SAMSUNG ELECTRONICS

USER'S MANUAL

S5PC100

June, 2009

REV 1.01

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product.

S5PC100 RISC Microprocessor
User's Manual, Revision 1.01
Publication Number:

Samsung Electronics' microcontroller business has been awarded full ISO-14001 certification (BSI Certificate No. FM24653). All semiconductor products are designed and manufactured in accordance with the highest quality standards and objectives.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea
C.P.O. Box #37, Suwon 446-711

TEL: (82)-(31)-209-2659
FAX: (82)-(31)-209-6285

Home Page: http://www.samsungsemi.com
Printed in the Republic of Korea

E-Mail: mobilesol.cs@samsung.com

Revision History

Revision No.	Description	Author(s)	Date
0.00	Initial Draft (Preliminary spec)	AP design	November, 2008
0.10	Following chapters are added - Electrical Data, Mechanical Data Following chapters are updated - Overview, Memory map, Chip ID, GPIO, Clock controller, Power management, Booting sequence, DRAM controller, SROM controller, OneNAND controller, NAND Flash controller, DMA controller, System timer, MIPI DSIM, MIPI CSIS, USB HOST controller, USB2.0 HS OTG, Modem interface, SD/MMC controller, Display controller, Camera interface, JPEG, FIMG-3DSE, TV & Video DAC, Video Processor, Mixer, HDMI, Multi format codec, Audio sub system, ADC & Touch screen interface, Keypad interface, Security system, Advanced crypto engine	AP design	January, 2009
0.20	Unused pin are renamed.	AP Evaluation	February, 2009
1.00	Public draft	AP Evaluation	May, 2009
1.01	Some chapters are updated	AP Evaluation	June, 2009

1.1 PRODUCT OVERVIEW

1 ARCHITECTURAL OVERVIEW

S5PC100 is a 32-bit RISC cost-effective, low power, high performance microprocessor solution for mobile phones and general applications, and integrates an ARM CortexTM-A8 which implements the ARM architecture V7-A with supporting numerous peripherals.

To provide optimized Hardware (H/W) performance for the 3G and 3.5G communication services, S5PC100 adopts 64-bit internal bus architecture and includes many powerful hardware accelerators for tasks such as motion video processing, display control and scaling. Integrated Multi Format Codec (MFC) supports encoding and decoding of MPEG-1/2/4, H.263, H.264 and decoding of VC1, Divx. This Hardware accelerators support real-time video conferencing and Analog TV out, HDMI for NTSC and PAL mode

The S5PC100 has an optimized interface to external memory capable of sustaining the demanding memory bandwidths required in high-end communication services. The memory system has Flash/ ROM external memory ports for parallel access and DRAM port for high bandwidth. DRAM port can be configured to support mobile DDR, DDR2 or LPDDR2.

Flash/ROM Port supports NAND Flash, NOR-Flash, OneNAND and ROM type external memory.

To reduce total system cost and enhance overall functionality, S5PC100 includes many hardware peripherals such as TFT 24-bit true color LCD controller, Camera Interface, MIPI DSI, CSI-2 and HSI, System Manager for power management, CF+, ATA I/F, 4-channel UART, 24-channel DMA, 4-channel Timers, General I/O Ports, 3-ch IIS, 1-ch S/PDIF, 2-ch CAN bus, IIC-BUS interface, 3-ch HS-SPI, USB Host v1.1, USB OTG v2.0 operating at high speed (480Mbps), SD Host & High Speed Multi-Media Card Interface and PLLs for clock generation.

Package on Package (POP) option with MCP is available for small form factor applications.

Salient features of S5PC100 are summarized below:

2 BLOCK DIAGRAM

This section summarizes the features of the S5PC100. Figure 1.1- **1** shows an overall block diagram of the S5PC100.

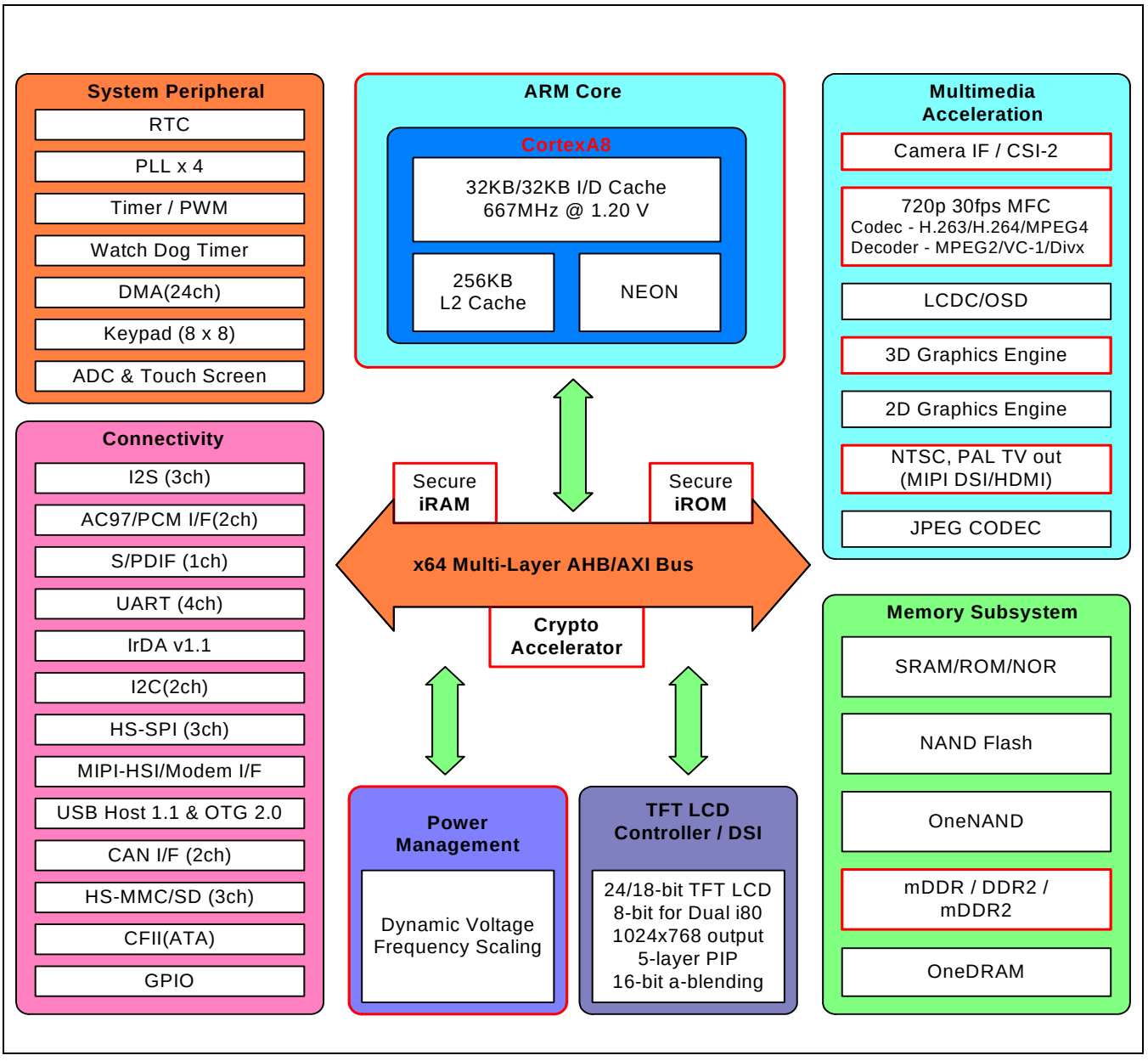


Figure 1.1- 1 S5PC100 Block Diagram

3 FEATURES

- ARM Cortex™-A8 based CPU Subsystem with NEON
  - ♦ 32/32KB I/D Cache, 256KB L2 Cache
  - ♦ Operating frequency up to 667MHz at 1.2V.
- 64-bit Multi-layer bus architecture
  - ♦ D0_BUS domain for ARM Cortex™-A8, Memory subsystem, Debugger, VIC, DMA, 2D, Security subsystem and CF controller, which shares pins with static memory controllers.
  - ♦ D1_BUS domain mainly for Multimedia IPs and for other peripherals
  - ♦ D2_BUS domain for low-power audio play
- Advanced power management for mobile applications
- Interal ROM for secure booting and internal RAM for security function
- 8-bit ITU 601/656 Camera Interface up to 8M pixel for scaled and 64M pixel for un-scaled resolution
- Multi Format Codec provides encoding and decoding of MPEG-4/H.263/H.264 up to 720p@30fps and decoding of MPEG-2/VC1/Divx video up to 720p@30fps.
- JPEG codec support up to 0.5Mpixels/MHz
- 3D Graphics Acceleration with programmable shader
- 2D Graphics Acceleration with BitBlit and Rotation
- 1/2/4/8 bpp Palletized or 8/16/24bpp Non-Palletized Color-TFT support up to 2048x2048 resolution.
- TV-out and HDMI interface support for NTSC and PAL mode with image enhancer
- MIPI-HSI, MIPI-DSI and MIPI-CSI interface
- AC-97 audio codec interface and PCM serial audio interface
- 3-channel 24-bit I2S interface
- 1-channel TX only S/PDIF interface support for digital audio
- 2-channel I2C interface
- 3-channel SPI interface
- 4-channel UART including 4Mbps port for Bluetooth 2.0
- Dedicated IrDA port for SIR/MIR/FIR
- USB 2.0 OTG supporting high speed (480Mbps, on-chip transceiver)
- USB 1.1 Host.
- Asynchronous direct Modem Interface.
- Channel SD/SDIO/HS-MMC interface.
- CF version 3.0 interface.
- 24-channel DMA controller (8 channels for Memory-to-memory DMA, 16 channels for Peripheral DMA)
- 8x8 key matrix interface.
- 10-channel 12-bit multiplexed ADC
- 2-channel CAN interface

- Configurable GPIOs
- Real time clock, PLL, timer with PWM and watch dog timer.
- System timer for variable tick.
- Memory Subsystem
  - SRAM/ROM/NOR/NAND Interface with x8 or x16 data bus
  - Muxed OneNAND Interface with x16 data bus
  - Mobile DDR Interface with x32 data bus (266~333Mbps/pin)
  - DDR2 interface with x32 data bus (333Mbps/pin)
  - Mobile DDR2 interface (333Mbps/pin)

3.1 MICROPROCESSOR

- The ARM Cortex™-A8 processor is the first application processor based on the ARMv7 architecture.
- With the ability to scale in speed from 600MHz to greater than 1GHz, the ARM Cortex™-A8 processor meets the requirements for power-optimized mobile devices needing operation in less than 300mW; and performance-optimized consumer applications requiring 2000 Dhrystone MIPS.
- ARM's first superscalar processor featuring technology for enhanced code density and performance, NEONTM technology for multimedia and signal processing, and Jazelle® RCT technology for efficient support of ahead-of-time and just-in-time compilation of Java and other byte code languages.
- ARM Cortex™-A8 Features
  - Thumb-2 technology for greater performance, energy efficiency, and code density
  - NEON™ signal processing extensions
  - Jazelle RCT Java-acceleration technology
  - TrustZone technology for secure transactions and DRM
  - 13-stage main integer pipeline
  - 10-stage NEON™ media pipeline
  - Integrated L2 Cache using standard compiled RAMs
  - Optimized L1 caches for performance and power

3.2 MEMORY SUBSYSTEM

- High bandwidth Memory Matrix subsystem
- Two independent external memory ports (1 x16 Static Hybrid Memory port and 1 x32 DRAM port)
- Matrix architecture increases overall bandwidth with the simultaneous access capability
  - ♦ SRAM/ROM/NOR Interface
    - ∗ x8 or x16 data bus
    - ∗ Address range support: 21-bit
    - ∗ Supports byte and half-word access
  - ♦ NAND Interface
    - ∗ Support industry standard NAND interface
    - ∗ x8 data bus
    - ∗ Density support: up to 32-Gb
  - ♦ Muxed OneNAND Interface
    - ∗ x16 data bus
    - ∗ Supports byte and half-word access
    - ∗ Supports 2KB page mode
    - ∗ Supports FlexOneNAND (4KB page mode)
  - ♦ Mobile DDR Interface
    - ∗ x32 data bus with 333Mbps/pin double data rate (DDR)
    - ∗ 1.8V interface voltage
    - ∗ Density support: up to 1-Gb per 1 CS (support upto 2 CS)
  - ♦ DDR2 Interface
    - ∗ x32 data bus with 333Mbps/pin double data rate (DDR)
    - ∗ 1.8V interface voltage
    - ∗ Density support: up to 512Mb per 1 CS (support upto 2 CS, when 4bank DDR2)
    - ∗ Density support: up to 4-Gb per 1 CS (support upto 1 CS, when 8 bank DDR2)
  - ♦ Mobile PDDR2 interface
    - ∗ x32 data bus with up to 333Mbps/pin
    - ∗ 1.2V interface voltage
    - ∗ Density support: up to 1-Gb per 1 CS (support upto 2 CS)

3.3 MULTIMEDIA ACCELERATION

- Camera Interface
  - Multiple input support
    - ITU-R BT 601/656 mode
    - DMA (AXI 64bit interface) mode
    - MIPI (CSI) mode
  - Multiple output support
    - DMA (AXI 64bit interface) mode
    - Direct FIFO mode
  - DZI (Digital Zoom In) capability
  - Multiple camera input support
  - Programmable polarity of video sync signals
  - Max. 8192 x 8192 pixels input support
  - Image mirror and rotation (X-axis mirror, Y-axis mirror, 90°, 180° and 270° rotation)
  - Various image format generation
  - Capture frame control support
  - Image effect support

- Multi-Format video Codec (MFC)
  - Real-time Video Encoding & decoding of MPEG-4/H.263/H.264 and decoding of MPEG-2/WMV9/Divx/Xvid.
  - MPEG4 up to ASP: 720p@30fps(1280x720)
  - H.263 P3: 720p@30fps (1280x720)
  - H.264 up to HP: 720p@30fps (1280x720)
  - VC1 Decoding: 720p@30fps (1280x720)
  - Resolutions: upto 1280x720(720p, D1, VGA, QVGA, CIF, QCIF)
  - Minimum size: 32x16, 16x32 for decoder, 32x32 for encoder
  - Supports single stream 720p @ 30fps encoding/decoding.
  - Time-multiplexed multi-stream encoding/decoding with fine-granular context switching.
  - Supported chroma format: encoding 4:2:0 and decoding: 4:2:0, 4:0:0 8 bit per sample.
  - Chrominance Interleaving.
  - Supports CABAC/CABAD in H.264
  - FMO and ASO not supported in H.264 decoding.
  - Supports only one rectangular visual object in MPEG-4 encoding/decoding.
  - Only forward reversible VLC (RVLC) is supported in MPEG-4 decoding.
  - MPEG-4 post-processing by re-using H.264 in-loop filter.
  - Supports only stationary warping function in GMC.

- ♦ Supports error resilience tool in MPEG-4 decoding.
- ♦ Error detection and concealment in decoding.
- ♦ Up to quarter-pel search for H.264
- ♦ Up to half-pel search for MPEG4
- ♦ Motion estimation search range: [+/-64, +/-32]
- ♦ All variable block sizes are supported. But 8x4, 4x8, 4x4 block sizes are not supported in encoding.
- ♦ Supports only DC prediction in MPEG-4 encoding.
- ♦ Rate control: CBR (Constant Bit-Rate) and VBR (Variable Bit-Rate)
- ♦ Frame level (H.264/MPEG4/H.263) and macroblock level (H.264) rate control can be enabled or disabled selectively.
- ♦ Progressive encoding only support.
- ♦ Non paired field mode is not supported.
- ♦ Start code must be included in the stream at decoding.

- JPEG Codec
  - ♦ Compression/decompression up to 65536x65536.
  - ♦ Support format of compression

    * Input raw image: YCbCr4:2:2 or RGB565

    * Output JPEG file: Baseline JPEG of YCbCr4:2:2 or YCbCr4:2:0

  - ♦ Support format of decompression (Refer to chapter 9.4)

    * Input JPEG file: Baseline JPEG of YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0, gray

    * Output raw image: YCbCr4:2:2 or YCbCr4:2:0

  - ♦ Support general-purpose color-space converter.

- 3D Graphic Engine
  - ♦ 5M triangles/s rendering performance
  - ♦ Max. 200M pixels/s fill-rates
  - ♦ Programmable Shader Model 3.0 support
  - ♦ 128-bit (32-bit x4) Floating-point Vertex Shader
  - ♦ 128-bit (32-bit x4) Floating-point two Fragment Shaders
  - ♦ Texture format: 1/2/4/8/16/32-bpp RGB, YUV422, S3TC Compressed
  - ♦ API support: OpenGL ES 1.1&2.0, D3D Mobile
  - ♦ Maximum 4K x 4K frame-buffer (16/32-bpp)
  - ♦ 32-bit depth buffer (8-bit stencil/24-bit Z)
  - ♦ Host Interface: 32-bit AHB
  - ♦ Memory Interface: three 64-bit AXI channels

- 2D Graphic Accelerator
  - Primitives: BitBlt and rotation, Color expansion, Line/Point Drawing
  - Per-pixel Operations: 256 3-operand ROP, Alpha blending, Window clipping
  - Data format: 16/24/32-bpp color format, YUV422 2plane
  - 64-bit AXI interface
- Analog TV interface
  - Out video format: NTSC-M/NTSC-J/NTSC4.43/PAL-B, D, G, H, I/PAL-M/PAL-N/PAL-Nc/PAL-60 compliant
  - Support input format: ITU-R BT.601 (YCbCr 4 :4 :4)
  - Support 480i/p and 576i resolution
  - Support Composite/S-Video/Component interface
- Digital TV Interface
  - High Definition Multimedia Interface (HDMI) 1.2
  - Support up to 720p 60Hz and 8-channel/112kHz/24-bit audio
  - Support for 480p, 576p, 720p, 1080i (cannot support 480i)
  - Support for HDCP v1.1
- Rotator
  - Supported image format: YCbCr422 (interleave), YCbCr420(non-interleave), RGB565 and RGB888(unpacked)
  - Supported rotate degree: 90, 180, 270, flip vertical and flip horizontal
- Video processor
  - Support BOB / 2D-IPC mode
  - Produce YCbCr 4:4:4 outputs to help MIXER to blend video and graphics
  - 1/4X to 16X vertical scaling with 4-tap/16-phase poly-phase filter
  - 1/4X to 16X horizontal scaling with 8-tap/16-phase poly-phase filter
  - Support Pan&Scan, Letterbox, and NTSC/PAL conversion using scaling
  - Flexible scaled video positioning within display area
  - Support 1/16 pixel resolution Pan&Scan mode
  - Flexible post video processing

    * Color saturation, Brightness/Contrast enhancement, Edge enhancement

    * Color space conversion between BT.601 and BT.709
  - Video input source size upto 1280x720
- Video Mixer

- Support overlapping & blending input video & graphic layers
- Support 480i/p, 576i/p, 720p and 1080i display size
- Support 4 layers (1 video layer, 2 graphic layer, 1 background layer)

3.4 AUDIO SUBSYSTEM

- Audio processing is progressed by NEON™ in ARM Cortex™-A8 CPU
- Low power audio subsystem
  - 5.1ch I2S with 32-bit-width 64-depth FIFO
  - 128KB audio play output buffer
  - HW mixer mixes primary sound and secondary sound.

3.5 SECURITY SUBSYSTEM

- On-Chip secure boot ROM
  - 32KB secure boot ROM for secure boot
- On-Chip secure RAM
  - 64KB secure RAM for security function
- H/W Crypto Accelerator
  - Securely integrated DES/TDES, AES, SHA-1, PRNG and PKA
  - Access control (Security Domain Manager with the ARM TrustZone HW)
  - Enabling enhanced secure platform for separate (secure/non-secure) execution environment for security sensitive application
- Secure JTAG
  - Authentication of JTAG user
  - Access control in JTAG mode

3.6 DISPLAY CONTROLLER

- TFT-LCD Interface
  - Support 24/18/16-bpp parallel RGB Interface LCD
  - Support 8/6 bpp serial RGB Interface
  - Support Dual i80 Interface LCD
  - 1/2/4/8bpp Palletized or 8/16/24-bpp Non-Palletized Color-TFT support
  - Typical actual screen size: 1024x768, 800x480, 640x480, 320x240, 160x160, and others
  - Virtual image up to 16M pixel (4K pixel x4K pixel)
  - Support 5 Window Layer for PIP or OSD
  - Real-time overlay plane multiplexing
  - Programmable OSD window positioning

- 16-level alpha blending
- ITU-BT601/656 format output

3.7 CONNECTIVITY

- PCM Audio Interface
  - ♦ 16-bit mono audio I/F
  - ♦ Master mode only
  - ♦ Support 2 port PCM interface
- AC97 Audio Interface
  - ♦ Independent channels for stereo PCM In, stereo PCM Out, mono MIC In.
  - ♦ 16-bit stereo (2-channel) audio.
  - ♦ Variable sampling rate AC97 Codec interface (48kHz and below)
  - ♦ Support AC97 Full Specification
- SPDIF Interface (TX only)
  - ♦ Linear PCM up to 24-bit per sample support
  - ♦ Non-Linear PCM formats such as AC3, MPEG1 and MPEG2 support
  - ♦ 2x24-bit buffers, which is alternately filled with data
- I2S Bus Interface
  - ♦ 2-ch IIS-bus for the audio-codec interface with DMA-based operation
  - ♦ Serial, 8/16/24-bit per channel data transfers
  - ♦ Supports IIS, MSB-justified and LSB-justified data format
  - ♦ Support PCM 5.1 channel
  - ♦ Various bit clock frequency and codec clock frequency support
    - ∗ 16, 24, 32, 48 fs of bit clock frequency
    - ∗ 256, 384, 512, 768 fs of codec clock
  - ♦ Support 1 5.1ch I2S (in Audio Subsystem) and 2 2ch I2S
- Modem Interface
  - ♦ Asynchronous direct/indirect 16-bit SRAM-style interface
  - ♦ On-chip 16KB dual-ported SRAM buffer for direct interface
- I2C Bus Interface
  - ♦ 2-ch Multi-Master IIC-Bus
  - ♦ Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbit/s in the standard mode
  - ♦ Up to 400 Kbit/s in the fast mode
- MIPI HSI Interface
  - ♦ MIPI Standard Draft Compliant

- High speed synchronous serial interface

- CompactFlash Controller
  - Supports Compact Flash Specification Revision 3.0
  - Compatible with the ATA/ATAPI-6 standard

- IrDA
  - Dedicated IrDA for v1.1 (1.152Mpbs and 4Mpbs)
  - SIR(111.5kbps) mode is supported by the URAT IrDA 1.0 block
  - Internal 64-byte Tx/Rx FIFO

- UART
  - 4-channel UART with DMA-based or interrupt-based operation
  - Supports 5-bit, 6-bit, 7-bit, or 8-bit serial data transmit/receive
  - Supports external clock for the UART operation (UCLK)
  - Programmable baud rate
  - Supports IrDA 1.0 SIR (115.2Kbps) mode
  - Loop back mode for testing
  - Non-integer clock divides in Baud clock generation (BRM)

- USB OTG 2.0
  - Complies with the USB OTG 2.0
  - Supports high speed up to 480Mbps
  - On-chip USB transceiver

- USB Host 1.1
  - Complies with the USB Host 1.1
  - Supports full speed up to 12Mbps
  - On-chip USB transceiver

- CAN Interface
  - Supports CAN protocol version 2.0 part A and B
  - Bit rates up to 1 Mbit/s
  - Support Programmable FIFO mode and programmable loop-back mode for self-test
  - Support Maskable interrupt
  - Support DAR mode for time triggered CAN applications
  - 8-bit non-multiplex Motorola HC08 compatible module interface
  - Support two 16-bit module interfaces to the AMBA APB bus from ARM

- HS-MMC/SDIO Interface
  - Multimedia Card Protocol version 4.0 compatible (HS-MMC)
  - SD Memory Card Protocol version 2.0 compatible
  - DMA based or Interrupt based operation
  - 128 word FIFO for Tx/Rx
  - 3-channel HS-MMC or 3-channel SDIO

- SPI Interface
  - 3-ch Serial Peripheral Interface Protocol version 2.11 compatible
  - 16 Byte (SPI) and 64 Byte (HS-SPI) FIFO for RX/TX
  - DMA-based or interrupt-based operation

- GPIO
  - Controls 173 external interrupts
  - Support 203 multi-functional input/output ports
  - GPA0: 8 in/out port – 2xUART with flow control
  - GPA1: 5 in/out port – 2xUART without flow control or 1xUART with flow control, 1x IrDA
  - GPB: 8 in/out port – 2x SPI
  - GPC: 5 in/out port – I2S, PCM, AC97
  - GPD: 7 in/out port – 2xI2C, PWM, External DMA request, SPDIF
  - GPE0,1: 14 in/out port – Camera I/F 0, MMC channel1(GPE0 support only 4-bit mode MMC, if 8-bit mode is needed, you can use GPE1 for another 4-bit data channel)
  - GPF0,1,2,3: 28 in/out port – LCD I/F
  - GPG0,1,2,3: 25 in/out port – 3xMMC channel(channel 0 support 4-bit and 8-bit mode, but channel 1, channel 2 support only 4-bit mode), SPI, I2S, PCM, SPDIF
  - GPH0,1,2,3: 32 in/out port – CAM IF channel, Key pad, External Wake-up(up-to 32-bit)
  - GPI: 8 in/out port – Booting option, SPW IF
  - GPJ0,1,2,3,4: 33 in/out port – Modem IF, HSI, ATA, LCD IF1
  - GPK0,1,2,3: 30 in/out port – SROM, NF, CF, OneNAND
  - GPL: 37 in/out memory port – EBI
  - MP1: 71 in/out memory port – DRAM

3.8 SYSTEM PERIPHERAL

- Real Time Clock
  - Full clock features: sec, min, hour, date, day, month, year
  - 32.768kHz operation
  - Alarm interrupt
  - Time-tick interrupt
- PLL
  - Four on-chip PLLs, APLL/MPLL/EPLL/HPLL
  - APLL dedicates to ARM core
  - MPLL generates a system bus clock and several special clocks
  - EPLL generates several special clocks
  - HPLL generates clocks for the HDMI interface
- Keypad
  - 8x8 Key Matrix support
  - Provides internal de-bounce filter
- Timer with Pulse Width Modulation
  - 5-ch 32-bit internal timer with interrupt-based operation
  - 3-ch 32-bit Timer with PWM
  - Programmable duty cycle, frequency, and polarity
  - Dead-zone generation
  - Support external clock source
- System timer
  - Accurate timer providing exact 1ms tick at any power mode except sleep.
  - Changeable interrupt interval without stopping reference tick timer
- DMA
  - Micro-code programming based DMA
  - The specific instruction set provides flexibility for programming DMA transfers
  - Linked list DMA function is supported.
  - 3 Enhanced DMA embedded. 8 channels supported per each DMA so then totally 24 channels is supported
  - 1 Memory to memory type optimized DMA + 2 Peripheral to memory type optimized DMA
  - M2M DMA supports up to 16burst, P2M DMA supports up to 8burst.

- A/D Converter and Touch Screen Interface
  - 10-ch multiplexed ADC
  - Max. 500Ksamples/sec and 12-bit resolution
- Watch Dog Timer
  - 16-bit watch dog timer
- Vectored Interrupt Controller
  - Multiple interrupt request inputs, one for each interrupt source, and one interrupt request output for the processor interrupt request input
  - Software can mask out particular interrupt requests
  - Prioritization of interrupt sources for interrupt nesting
- Power Management
  - Clock-off control for individual components
  - Various low-power modes are available such as Idle, Stop, Deep Idle, Deep Stop and Sleep mode
  - Sleep mode's wake up sources are external interrupts, RTC alarm, Tick timer and the key interface.
  - Stop and Deep Stop mode's wake up sources are MMC, Touch screen interface, Modem interface, MIPI HSI and the system timer as well all the wake up sources of Sleep mode.
  - Deep Idle mode's wake up sources are 5.1ch I2S as well all the wake up source of Stop mode.
- System operating clock generation
  - Four on-chip PLLs, APLL, MPLL, EPLL & HPLL
  - APLL is used for ARM Cortex^{TM}-A8 and D0_BUS domain (i.e., ARMCLK, HCLKD0, PCLKD0, HCLKD0_SECSS, and SCLK_ONENAND).
  - MPLL and EPLL are used for D1_BUS domain (HCLKD1 and PCLKD1) and other peripheral clocks (i.e., audio IPs, SPI, and etc.).
  - HPLL is used for a HD video clock (i.e., SCLK_HDMI).

3.9 ELECTRICAL CHARACTERISTICS

- Operating Conditions
  - Supply Voltage for Logic Core: VDD_INT 1.2V ±5%, VDD_ARM depends on operating frequency
  - External Memory Interface: 1.2/1.8V
  - External I/O Interface: 1.8/2.5/3.3V
- Operational Frequency
  - 667MHz@ ARM/Internal 1.2V
  - 833MHz@ ARM1.33V/Internal 1.25V

NOTES: 1.S5PC100 has three system clock domains called D0, D1 and D2. D0 domain is for CPU system, while D1 for multimedia, D2 for low power audio. Nominal 133/166MHz represents that D1 system frequency is 133MHz & D0 system frequency is 166MHz at 1.2V power level.
2. C100 supports only sync mode between CPU and D0 system.
3. The voltage of the external Memory & I/O interface depends on the attached device specification. Therefore, the PMIC should supply an appropriate voltage to the interface.

1.2 MEMORY MAP

1 MEMORY ADDRESS MAP

Start Addr	Limit Addr	Size	Usage	Note
0x0000_0000	0x0008_0000	32KB	IROM	Mirrored Region of 0xD000_0000~ 0xD800_0000
0x0002_0000	0x0003_8000	96KB	IRAM	
0x2000_0000	0x6000_0000	1GB	DRAM	
0x8000_0000	0x8800_0000	128MB	SMC Bank 0	
0x8800_0000	0x9000_0000	128MB	SMC Bank 1	
0x9000_0000	0x9800_0000	128MB	SMC Bank 2	
0x9800_0000	0xA000_0000	128MB	SMC Bank 3	
0xA000_0000	0xA800_0000	128MB	SMC Bank 4	
0xA800_0000	0xB000_0000	128MB	SMC Bank 5	
0xB000_0000	0xC000_0000	256MB	OneNAND Controller	
0xC000_0000	0xC002_0000	128KB	MP3_SRAM	Refer to "10.1 Audio subsystem"
0xD000_0000	0xD000_8000	32KB	IROM	
0xD002_0000	0xD003_8000	96KB	IRAM	
0xD800_0000	0xE000_0000	128MB	DMZ ROM	
0xE000_0000	0x0000_0000	512MB	SFR Region	

Start Addr	Limit Addr	Size	Usage
0x0002_0000	0x0002_1000	4KB	IROM's Stack
0x0002_1000	0x0002_4000	12KB	User specific purpose
0x0002_4000	0x0003_4000	64KB	Secure Domain Manager
0x0003_4000	0x0003_8000	16KB	BL1

NOTE: TZPCR0SIZE[5:0](TZPC0); (in TZPC SFR)
- 4KByte chunks
- Recommended value; 6'b00_0000 ~ 6'b10_0000
  * if (TZPCR0SIZE[5](TZPC0) == 1'b1), the full address range in IRAM is configured as secure.
  * if (TZPCR0SIZE(TZPC0) == 6'b00_0000), there is non-secure region in IRAM. (0kB)
  * if (TZPCR0SIZE(TZPC0) == 6'b00_0001), minimum secure region size (4kB)
  * if (TZPCR0SIZE(TZPC0) == 6'b01_0000), the 64KB from IRAM start address is secure region.
- iROM is always secure area

2 SPECIAL FUNCTION REGISTER MAP

1. System		0xE1F0_0000	2. Bus architecture		0xE3F0_0000	3.Interrupt		0xE5F0_0000
		0xE1E0_0000			0xE3E0_0000			0xE5E0_0000
		0xE1D0_0000			0xE3D0_0000			0xE5D0_0000
		0xE1C0_0000			0xE3C0_0000			0xE5C0_0000
		0xE1B0_0000			0xE3B0_0000			0xE5B0_0000
		0xE1A0_0000			0xE3A0_0000			0xE5A0_0000
		0xE190_0000			0xE390_0000			0xE590_0000
		0xE180_0000		TZPC0	0xE380_0000			0xE580_0000
		0xE170_0000			0xE370_0000			0xE570_0000
		0xE160_0000			0xE360_0000			0xE560_0000
		0xE150_0000			0xE350_0000			0xE550_0000
		0xE140_0000		Async bridge	0xE340_0000			0xE540_0000
		0xE130_0000			0xE330_0000			0xE530_0000
		0xE120_0000			0xE320_0000		TZIC2	0xE520_0000
	IEM_IEC	0xE110_0000			0xE310_0000		TZIC1	0xE510_0000
	IEM_APC	0xE100_0000		AXI_D0 bus	0xE300_0000		TZIC0	0xE500_0000
		0xE0F0_0000			0xE2F0_0000			0xE4F0_0000
		0xE0E0_0000			0xE2E0_0000			0xE4E0_0000
		0xE0D0_0000			0xE2D0_0000			0xE4D0_0000
		0xE0C0_0000			0xE2C0_0000			0xE4C0_0000
		0xE0B0_0000			0xE2B0_0000			0xE4B0_0000
		0xE0A0_0000			0xE2A0_0000			0xE4A0_0000
		0xE090_0000		TZPC2	0xE290_0000			0xE490_0000
		0xE080_0000		TZPC1	0xE280_0000			0xE480_0000
		0xE070_0000			0xE270_0000			0xE470_0000
		0xE060_0000			0xE260_0000			0xE460_0000
		0xE050_0000			0xE250_0000			0xE450_0000
		0xE040_0000			0xE240_0000			0xE440_0000
	GPIO	0xE030_0000			0xE230_0000			0xE430_0000
	System control (clock, power)	0xE020_0000			0xE220_0000		VIC2	0xE420_0000
		0xE010_0000			0xE210_0000		VIC1	0xE410_0000
	Chip ID	0xE000_0000			0xE200_0000		VIC0	0xE400_0000

Figure 1.2- 1 SFR Map - 1

4. Memory		0xE7F0_0000	5. DMA		0xE9F0_0000	6. Timer		0xEBF0_0000
		0xE7E0_0000			0xE9E0_0000			0xEBE0_0000
		0xE7D0_0000			0xE9D0_0000			0xEBD0_0000
		0xE7C0_0000			0xE9C0_0000			0xEBC0_0000
		0xE7B0_0000			0xE9B0_0000			0xEBB0_0000
		0xE7A0_0000			0xE9A0_0000			0xEBA0_0000
		0xE790_0000			0xE990_0000			0xEB90_0000
	CF controller	0xE780_0000			0xE980_0000			0xEB80_0000
		0xE770_0000			0xE970_0000			0xEB70_0000
		0xE760_0000			0xE960_0000			0xEB60_0000
		0xE750_0000			0xE950_0000			0xEB50_0000
		0xE740_0000			0xE940_0000			0xEB40_0000
		0xE730_0000			0xE930_0000			0xEB30_0000
	NAND controller	0xE720_0000		PDMA1	0xE920_0000			0xEB20_0000
	OneNAND controller	0xE710_0000			0xE910_0000			0xEB10_0000
	SROM controller	0xE700_0000		PDMA0	0xE900_0000			0xEB00_0000
		0xE6F0_0000			0xE8F0_0000			0xEAF0_0000
		0xE6E0_0000			0xE8E0_0000			0xEAE0_0000
		0xE6D0_0000			0xE8D0_0000			0xEAD0_0000
		0xE6C0_0000			0xE8C0_0000			0xEAC0_0000
		0xE6B0_0000			0xE8B0_0000			0xEAB0_0000
		0xE6A0_0000			0xE8A0_0000			0xEAA0_0000
		0xE690_0000			0xE890_0000			0xEA90_0000
		0xE680_0000			0xE880_0000			0xEA80_0000
		0xE670_0000			0xE870_0000			0xEA70_0000
		0xE660_0000			0xE860_0000			0xEA60_0000
		0xE650_0000			0xE850_0000			0xEA50_0000
		0xE640_0000			0xE840_0000			0xEA40_0000
		0xE630_0000			0xE830_0000		Real time clock	0xEA30_0000
		0xE620_0000			0xE820_0000		Watchdog timer	0xEA20_0000
		0xE610_0000		MDMA	0xE810_0000		System timer	0xEA10_0000
	DRAM controller	0xE600_0000			0xE800_0000		PWM timer	0xEA00_0000

Figure 1.2- 2 SFR Map - 2

7. Connectivity/storage		0xEDF0_0000	8. Multi Media1		0xEFF0_0000	9. Multi Media2		0xF1F0_0000
		0xEDE0_0000			0xEFE0_0000			0xF1E0_0000
		0xEDD0_0000			0xEFD0_0000			0xF1D0_0000
		0xEDC0_0000			0xEFC0_0000			0xF1C0_0000
		0xEDB0_0000			0xEFB0_0000			0xF1B0_0000
	HS_SD/MMC2	0xEDA0_0000			0xEFA0_0000			0xF1A0_0000
	HS_SD/MMC1	0xED90_0000			0xEF90_0000			0xF190_0000
	HS_SD/MMC0	0xED80_0000			0xEF80_0000			0xF180_0000
		0xED70_0000			0xEF70_0000			0xF170_0000
		0xED60_0000			0xEF60_0000			0xF160_0000
	Modem interface	0xED50_0000			0xEF50_0000			0xF150_0000
	USB1.1 Host	0xED40_0000			0xEF40_0000			0xF140_0000
	USB2.0 HS OTG PHY	0xED30_0000			0xEF30_0000			0xF130_0000
	USB2.0 HS OTG LINK	0xED20_0000			0xEF20_0000			0xF120_0000
		0xED10_0000			0xEF10_0000			0xF110_0000
		0xED00_0000		FIMG-3D	0xEF00_0000		MFC	0xF100_0000
		0xECF0_0000			0xEEF0_0000			0xF0F0_0000
		0xECE0_0000			0xEEE0_0000			0xF0E0_0000
		0xECD0_0000			0xEED0_0000			0xF0D0_0000
	MIPI CSIS	0xECC0_0000			0xEEC0_0000			0xF0C0_0000
	MIPI DSIM	0xECB0_0000			0xEEB0_0000			0xF0B0_0000
	MIPI HSI RX	0xECA0_0000			0xEEA0_0000			0xF0A0_0000
	MIPI HSI TX	0xEC90_0000			0xEE90_0000			0xF090_0000
	CAN1	0xEC80_0000		FIMG-2D	0xEE80_0000			0xF080_0000
	CAN0	0xEC70_0000			0xEE70_0000			0xF070_0000
	IrDA	0xEC60_0000			0xEE60_0000			0xF060_0000
	SPI2	0xEC50_0000		JPEG	0xEE50_0000			0xF050_0000
	SPI1	0xEC40_0000		Camera interface2	0xEE40_0000			0xF040_0000
	SPI0	0xEC30_0000		Camera interface1	0xEE30_0000		HDMI	0xF030_0000
	HDMI_IIC	0xEC20_0000		Camera interface0	0xEE20_0000		Mixer	0xF020_0000
	IIC	0xEC10_0000		Image rotator	0xEE10_0000		Video processor	0xF010_0000
	UART	0xEC00_0000		Display controller	0xEE00_0000		TV out	0xF000_0000

Figure 1.2- 3 SFR Map - 3

10. Audio/Etc		0xF3F0_0000	11. Security		0xF5F0_0000	12. Debugging		0xF7F0_0000
		0xF3E0_0000			0xF5E0_0000			0xF7E0_0000
		0xF3D0_0000			0xF5D0_0000			0xF7D0_0000
		0xF3C0_0000			0xF5C0_0000			0xF7C0_0000
		0xF3B0_0000			0xF5B0_0000			0xF7B0_0000
		0xF3A0_0000			0xF5A0_0000			0xF7A0_0000
		0xF390_0000			0xF590_0000			0xF790_0000
	Cellguide	0xF380_0000			0xF580_0000			0xF780_0000
		0xF370_0000			0xF570_0000			0xF770_0000
		0xF360_0000			0xF560_0000			0xF760_0000
		0xF350_0000			0xF550_0000			0xF750_0000
		0xF340_0000			0xF540_0000			0xF740_0000
		0xF330_0000			0xF530_0000			0xF730_0000
		0xF320_0000			0xF520_0000			0xF720_0000
	Keypad interface	0xF310_0000		Secure domain manager	0xF510_0000			0xF710_0000
	ADC & Touch screen	0xF300_0000		SECKEY	0xF500_0000			0xF700_0000
		0xF2F0_0000			0xF4F0_0000			0xF6F0_0000
		0xF2E0_0000			0xF4E0_0000			0xF6E0_0000
		0xF2D0_0000			0xF4D0_0000			0xF6D0_0000
		0xF2C0_0000			0xF4C0_0000			0xF6C0_0000
		0xF2B0_0000			0xF4B0_0000			0xF6B0_0000
		0xF2A0_0000			0xF4A0_0000			0xF6A0_0000
		0xF290_0000			0xF490_0000			0xF690_0000
		0xF280_0000			0xF480_0000			0xF680_0000
		0xF270_0000			0xF470_0000			0xF670_0000
	SPDIF	0xF260_0000			0xF460_0000			0xF660_0000
	PCM1	0xF250_0000			0xF450_0000			0xF650_0000
	PCM0	0xF240_0000			0xF440_0000			0xF640_0000
	AC97	0xF230_0000		Advanced crypto engine	0xF430_0000			0xF630_0000
	I2S v3.2 1	0xF220_0000			0xF420_0000			0xF620_0000
	I2S v3.2 0	0xF210_0000			0xF410_0000			0xF610_0000
	I2S v5.0	0xF200_0000			0xF400_0000		Coresight	0xF600_0000

Figure 1.2- 4 SFR Map - 4

1.3 BALL MAP & SIZE

1 PIN ASSIGNMENT

1.1 PIN ASSIGNMENT DIAGRAM - 580-BALL FCFBGA (POP)

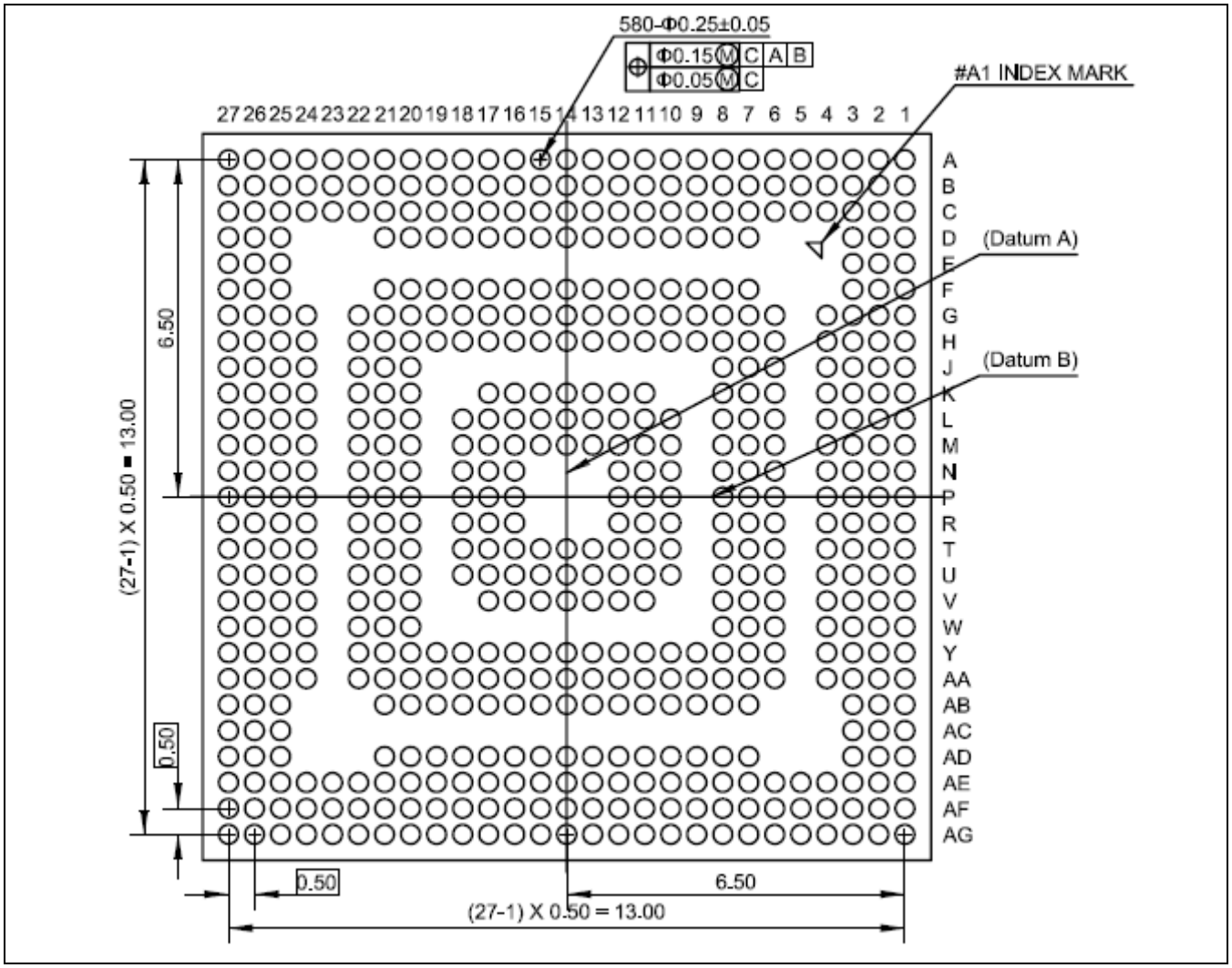


Figure 1.3- 1 S5PC100 Pin Assignment (580-FCFBGA) Bottom View

1.2 PIN NUMBER ORDER

Table 1.3-1 S5PC100 580 FCFBGA Pin Assignment – Pin Number Order (1/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
A1	VSS	B14	Xm0ADDR[4]	C27	XpwmTOUT[2]	F16	Xm0ADDR[0]
A2	VSS	B15	Xm0CFOEn	D1	Xm0WEn	F17	Xmmc0DATA[2]
A3	Xm0DATA[1]	B16	Xm0ADDR[19]	D2	Xm0FALE	F18	Xmmc1DATA[3]
A4	Xm0DATA[2]	B17	VDDQ_B	D3	Xm0DATA[5]	F19	XpwmTOUT[1]
A5	VDDQ_B	B18	Xm0IORDn	D7	Xm0DATA[13]	F20	Xmmc0DATA[1]
A6	Xm0DATA[6]	B19	XspiCSn[1]	D8	Xm0ADDR[11]	F21	VDD_ARM
A7	Xm0DATA[10]	B20	XjTMS	D9	VSS	F25	VSS
A8	VDD_DRAM	B21	Xi2c0SCL	D10	Xm0ADDR[6]	F26	XspiCSn[0]
A9	Xm0CSn[2]	B22	XpwmTOUT[0]	D11	Xm0ADDR[10]	F27	Xi2c1SDA
A10	Xm0ADDR[13]	B23	Xmmc1DATA[2]	D12	Xm0CFWEn	G1	VDDQ_B
A11	Xm0FRnB[1]	B24	XuRXD[0]	D13	Xm0ADDR[8]	G2	VDDQ_B
A12	VDDQ_B	B25	VDD_DRAM	D14	Xm0RESET	G3	Xm0OEn
A13	Xm0ADDR[18]	B26	VSS	D15	VSS	G4	XvVD[23]
A14	Xm0REG	B27	VSS	D16	Xm0DATA_RDn	G6	VSS
A15	XefVGATE_0	C1	Xm0FRnB[0]	D17	Xmmc0DATA[5]	G7	VDDQ_M0
A16	Xm0FREn	C2	Xm0FCLE	D18	Xmmc0DATA[7]	G8	Xm0ADDR[9]
A17	VDDQ_B	C3	VSS	D19	Xi2c0SDA	G9	Xm0FRnB[2]
A18	Xm0INPACKn	C4	Xm0DATA[3]	D20	Xmmc1DATA[0]	G10	Xm0WAITn
A19	N.C	C5	Xm0DATA[4]	D21	Xmmc1CLK	G11	Xm0ADDR[2]
A20	XjTCK	C6	Xm0DATA[8]	D25	XuTXD[2]	G12	VDDQ_DDR
A21	XjTDO	C7	Xm0DATA[12]	D26	XspiMOSI[0]	G13	VDDQ_DDR
A22	XuRXD[2]	C8	Xm0DATA[15]	D27	XuRXD[1]	G14	VSS
A23	XuTXD[1]	C9	VSS	E1	XvVD[9]	G15	VSS
A24	Xmmc0CMD	C10	Xm0ADDR[5]	E2	Xm0CSn[4]	G16	Xm0ADDR[7]
A25	XuRXD[3]	C11	Xm0IOWRn	E3	Xm0FWEn	G17	Xmmc0DATA[3]
A26	VSS	C12	Xm0ADDR[16]	E25	XuCTSn[0]	G18	Xmmc1CDn
A27	VSS	C13	Xm0ADDR[20]	E26	XjTRSTn	G19	Xmmc0DATA[0]
B1	VSS	C14	Xm0CDn	E27	VDD_DRAM	G20	VDD_ARM
B2	VSS	C15	VSS	F1	XvVD[4]	G21	VDD_ARM
B3	Xm0DATA[0]	C16	Xm0ADDR[1]	F2	XvVD[15]	G22	XuRTSn[0]
B4	Xm0DATA[7]	C17	XnRESET	F3	XvVD[22]	G24	XspiCLK[0]
B5	VDDQ_B	C18	Xmmc0CLK	F7	VDDQ_M0	G25	XDDR2SEL
B6	Xm0DATA[9]	C19	XspiMOSI[1]	F8	VDD_INT	G26	XjTDI
B7	Xm0DATA[11]	C20	XspiMISO[1]	F9	Xm0ADDR[15]	G27	XuCTSn[1]
B8	Xm0DATA[14]	C21	VSS	F10	Xm0INTRQ	H1	XvVD[3]
B9	Xm0BEn[0]	C22	Xmmc0DATA[6]	F11	Xm0FRnB[3]	H2	XvVD[8]
B10	Xm0IORDY	C23	XuTXD[0]	F12	Xm0ADDR[14]	H3	Xm0CSn[5]
B11	Xm0CSn[3]	C24	XuRTSn[1]	F13	Xm0ADDR[3]	H4	XvVD[1]
B12	VDDQ_B	C25	VSS	F14	VSS	H6	VDDQ_M0
B13	Xm0CSn[1]	C26	XuTXD[3]	F15	Xm0ADDR[17]	H7	Xm0ADDR[12]

Table 1.1-2 S5PC100 580 FCFBGA Pin Assignment – Pin Number Order (2/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
H8	XvVD[18]	K11	XciDATA[7]	M4	XvVCLK	P1	VCCQ_O
H9	XefFSOURCE_0	K12	XvVD[12]	M6	XciDATA[6]	P2	Xmmc2CLK
H10	VSS	K13	XvVD[16]	M7	XvVDEN	P3	Xmmc2DATA[2]
H11	VDDQ_DDR	K14	XvVD[11]	M8	VDDQ_LCD	P4	Xi2s0SDO[1]
H12	VDDQ_DDR	K15	VDD_INT	M10	XvVD[20]	P6	Xi2s0LRCK
H13	VDDQ_DDR	K16	VDD_ARM	M11	XvVD[5]	P7	Xi2s0SDO[0]
H14	VDD_INT	K17	VDD_ARM	M12	XvVD[19]	P8	Xmmc2CDn
H15	VDD_INT	K20	VDD_ARM	M13	VSS	P10	XciPCLK
H16	VDD_INT	K21	VDD_ARM	M14	VSS	P11	XciDATA[0]
H17	Xmmc0DATA[4]	K22	VSS	M15	VSS	P12	VSS
H18	VDD_ARM	K24	VSS	M16	VSS	P16	VSS
H19	VDD_ARM	K25	VSS	M17	VSS	P17	VSS
H20	VDD_ARM	K26	XnBATF	M18	VDD_ARM	P18	XEINT[7]
H21	VDD_ARM	K27	POP_DATA[1]	M20	VDDQ_EXT	P20	XEINT[22]
H22	VSS	L1	VCC_O	M21	Xmmc1CMD	P21	XEINT[2]
H24	XNFMOD[0]	L2	XciDATA[3]	M22	VSS	P22	XEINT[17]
H25	XuCLK	L3	XvHSYNC	M24	XEINT[1]	P24	XEINT[8]
H26	XspiMISO[0]	L4	VDDQ_CI	M25	POP_DATA[2]	P25	XEINT[20]
H27	XjDBGSEL	L6	XvVD[21]	M26	VDD_DRAM	P26	VSS
J1	VDD_DRAM	L7	XvVD[10]	M27	POP_DATA[4]	P27	VSS
J2	XiemSPWI	L8	XvVD[13]	N1	Xmmc2DATA[0]	R1	Xi2s1SCLK
J3	VSS	L10	Xm0BEn[1]	N2	Xmmc2DATA[1]	R2	VSS
J4	VSS	L11	VDD_INT	N3	XciCLKenb	R3	Xi2s0CDCLK
J6	XvVD[2]	L12	XvVD[0]	N4	XciVSYNC	R4	Xi2s0SDO[2]
J7	XvVD[7]	L13	VDD_INT	N6	VDDQ_LCD	R6	Xi2s1SDI
J8	XvVD[17]	L14	VDD_INT	N7	VSS	R7	VDDQ_MMC
J20	VDD_ARM	L15	VDD_INT	N8	XciDATA[1]	R8	Xmmc2DATA[3]
J21	VDD_ARM	L16	VSS	N10	XciHREF	R10	Xi2s0SDI
J22	VSS	L17	VDD_ARM	N11	XciRESET	R11	Xmmc2CMD
J24	XXTO	L18	VDD_ARM	N12	VSS	R12	VSS
J25	XXTI	L20	Xmmc1DATA[1]	N16	VSS	R16	VSS
J26	VDD_ALIVE	L21	VDD_ARM	N17	VSS	R17	XEINT[25]
J27	POP_INTB_B	L22	VSS	N18	XEINT[16]	R18	XEINT[24]
K1	XciDATA[4]	L24	XspiCLK[1]	N20	XEINT[6]	R20	XEINT[5]
K2	POP_CEB	L25	XPWRRGTON	N21	Xmmc0CDn	R21	VDDQ_EXT
K3	XvVSYNC	L26	VDDQ_SYS0	N22	Xi2c1SCL	R22	XEINT[3]
K4	XciFIELD	L27	N.C	N24	XEINT[18]	R24	XEINT[21]
K6	XvVD[14]	M1	XiemSCLK	N25	XEINT[0]	R25	POP_DATA[3]
K7	XvVD[6]	M2	XciDATA[2]	N26	POP_DATA[0]	R26	VDDQ_A
K8	Xm0CSn[0]	M3	XciDATA[5]	N27	POP_DATA[7]	R27	VDDQ_A

Table 1.1-3 S5PC100 580 FCFBGA Pin Assignment – Pin Number Order (3/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
T1	Xi2s1LRCK	U21	XEINT[13]	W27	VDDQ_A	AA16	VDD_EPLL
T2	Xi2s1CDCLK	U22	XEINT[12]	Y1	XmsmDATA[8]	AA17	VDD_MPLL
T3	Xi2s0SCLK	U24	XEINT[26]	Y2	XmsmDATA[3]	AA18	VSS_APLL
T4	Xi2s1SDO	U25	XNFMOD[1]	Y3	XmsmDATA[5]	AA19	VSS_UOTG
T6	XmsmDATA[9]	U26	POP_DATA[6]	Y4	XmsmDATA[0]	AA20	XusbXTO
T7	XmsmWEn	U27	POP_DATA[8]	Y6	VDDQ_MSM	AA21	XadcAIN[3]
T8	XmsmADDR[8]	V1	VSS	Y7	XmsmADDR[6]	AA22	XadcAIN[6]
T10	XmsmADDR[1]	V2	VSS	Y8	XmsmADDR[10]	AA24	XadcAIN[0]
T11	XCLKOUT	V3	VSS	Y9	XNFMOD[2]	AA25	XadcAIN[4]
T12	VDD_INT	V4	XmsmDATA[4]	Y10	VDD12_HDMI	AA26	XadcAIN[8]
T13	VSS	V6	XmsmDATA[6]	Y11	VSS	AA27	POP_DATA[12]
T14	VSS	V7	XmsmADDR[3]	Y12	VDD_INT	AB1	XmsmDATA[2]
T15	VSS	V8	XmsmRn	Y13	VDD18_MIPI_PLL	AB2	VSS_DAC_A
T16	XEINT[10]	V11	XmsmADDR[11]	Y14	XmipiReg_cap	AB3	XdacIREF
T17	XEINT[30]	V12	VDDQ_SYS2	Y15	VDDQ18_MIPI	AB7	VDD30_DAC_D
T18	XNFMOD[4]	V13	VDD_INT	Y16	VDDQ_UHOST	AB8	VSS_HDMI
T20	VSS	V14	VDD_INT	Y17	VSS_MPLL	AB9	VSS_HDMI
T21	XEINT[23]	V15	XusbDRVVBUS	Y18	VDD33_UOTG	AB10	XhdmiREXT
T22	XEINT[27]	V16	VSS	Y19	XOM[1]	AB11	VSS_MIPI_PLL18
T24	XEINT[4]	V17	VDDQ_SYS5	Y20	XEINT[11]	AB12	VSS_MIPI
T25	XEINT[19]	V20	XEINT[14]	Y21	VSS	AB13	VSS_MIPI
T26	POP_DATA[5]	V21	XEINT[29]	Y22	XOM[4]	AB14	VSS_USBHOST
T27	POP_DM[0]	V22	XEINT[28]	Y24	VSS_ADC	AB15	XuhDN
U1	VCCQ_O	V24	XEINT[31]	Y25	VSS	AB16	VSS_EPLL
U2	VCCQ_O	V25	VDD_RTC	Y26	VSS	AB17	VDD_APLL
U3	VDDQ_AUD	V26	POP_DATA[10]	Y27	POP_DATA[13]	AB18	XusbID
U4	XmsmDATA[14]	V27	POP_DATA[11]	AA1	XmsmADDR[12]	AB19	XusbREXT
U6	XmsmDATA[13]	W1	XmsmDATA[7]	AA2	XdacOUT[0]	AB20	XusbXTI
U7	XmsmDATA[12]	W2	XmsmDATA[15]	AA3	XdacVREF	AB21	XusbVBUS
U8	XmsmADDR[9]	W3	XmsmDATA[11]	AA4	XdacOUT[1]	AB25	VDD33_ADC
U10	XmsmADDR[2]	W4	XmsmDATA[10]	AA6	XdacCOMP	AB26	XadcAIN[9]
U11	XnRSTOUT	W6	VDDQ_MSM	AA7	VSS_DAC_D	AB27	POP_DATA[15]
U12	VDD_INT	W7	XmsmDATA[1]	AA8	XmsmADDR[5]	AC1	VDD30_DAC_A
U13	VDD_INT	W8	XmsmIRQn	AA9	XNFMOD[3]	AC2	XdacOUT[2]
U14	VDD_INT	W20	XOM[0]	AA10	X27mXTO	AC3	VSS
U15	VDD_INT	W21	XOM[2]	AA11	X27mXTI	AC25	XadcAIN[7]
U16	VSS	W22	XOM[3]	AA12	VSS_HPLL	AC26	VDDQ_A
U17	XEINT[15]	W24	XrtcXTI	AA13	VDD12_MIPI	AC27	POP_DATA[14]
U18	VDDQ_CAN	W25	XrtcXTO	AA14	VDD12_MIPI	AD1	XmsmADDR[7]
U20	XEINT[9]	W26	POP_DATA[9]	AA15	XuhDP	AD2	VCC_O

Table 1.1-4 S5PC100 580 FCFBGA Pin Assignment – Pin Number Order (4/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
AD3	XmsmCSn	AE18	POP_ADDR[10]	AF27	VSS		
AD7	VDD12_HDMI	AE19	POP_BA[1]	AG1	VSS		
AD8	XmipiDP[0]	AE20	VDD12_UOTG	AG2	VSS		
AD9	XmipiDN[1]	AE21	VSS	AG3	XhdmiTX1N		
AD10	XmipiDN[3]	AE22	XadcVref	AG4	VDDQ_A		
AD11	VDDQ_A	AE23	VDD_DRAM	AG5	XhdmiTX0N		
AD12	VDD_HPLL	AE24	XadcAIN[2]	AG6	XhdmiTXCN		
AD13	VDDQ_A	AE25	XadcAIN[5]	AG7	XmipiTXCP		
AD14	POP_DATA[24]	AE26	POP_ADDR[5]	AG8	XmipiDN[2]		
AD15	VSS	AE27	POP_ADDR[9]	AG9	XmipiDP[4]		
AD16	POP_DATA[21]	AF1	VSS	AG10	N.C		
AD17	POP_ADDR[1]	AF2	VDD_DRAM	AG11	XmipiRXCP		
AD18	POP_BA[0]	AF3	XhdmiTX1P	AG12	POP_DATA[29]		
AD19	XusbDP	AF4	VSS	AG13	POP_DATA[28]		
AD20	XusbDM	AF5	XhdmiTX0P	AG14	POP_DATA[31]		
AD21	VSS	AF6	XhdmiTXCP	AG15	POP_DATA[30]		
AD25	XadcAIN[1]	AF7	XmipiTXCN	AG16	VDD_DRAM		
AD26	POP_DM[1]	AF8	XmipiDP[2]	AG17	POP_DATA[16]		
AD27	POP_INTB_A	AF9	XmipiDN[4]	AG18	POP_DATA[17]		
AE1	XhdmiTX2P	AF10	N.C	AG19	POP_ADDR[0]		
AE2	XhdmiTX2N	AF11	XmipiRXCN	AG20	POP_RASN		
AE3	XmsmADDR[0]	AF12	POP_DATA[26]	AG21	POP_CLK		
AE4	VSS	AF13	POP_DATA[27]	AG22	POP_CKE		
AE5	XmsmADDR[4]	AF14	POP_DATA[18]	AG23	POP_ADDR[6]		
AE6	XNFMOD[5]	AF15	POP_DATA[19]	AG24	POP_ADDR[7]		
AE7	VDDQ_A	AF16	POP_DATA[20]	AG25	POP_ADDR[4]		
AE8	XmipiDN[0]	AF17	POP_DM[2]	AG26	VSS		
AE9	XmipiDP[1]	AF18	POP_ADDR[3]	AG27	VSS		
AE10	XmipiDP[3]	AF19	POP_CASN				
AE11	VSS	AF20	POP_CSN				
AE12	POP_DATA[25]	AF21	VSS_UOTG				
AE13	POP_DM[3]	AF22	POP_WEN				
AE14	POP_DATA[23]	AF23	POP_ADDR[12]				
AE15	VSS	AF24	POP_ADDR[11]				
AE16	POP_DATA[22]	AF25	POP_ADDR[8]				
AE17	POP_ADDR[2]	AF26	VSS				

1.3 POWER AND GROUND PIN ASSIGNMENT

TABLE 1.1-5 S5PC100 POWER PIN TO BALL ASSIGNMENT

Power Group	Pin Name	Ball	Description
MCP DDR	VDDQ_A	R26, R27, W27, AC26, AD11, AD13, AE7, AG4	OneDRAM A port I/O
	VDDQ_B	A5, A12, A17, B5, B12, B17, G1, G2	OneDRAM B port and mDDR I/O
	VDD_DRAM	A8, B25, E27, J1, M26, AE23, AF2, AG16	OneDRAM and mDDR core
MCP OneNAND	VCCQ_O	P1, U1, U2	OneNAND I/O
	VCC_O	L1, AD2	OneNAND core
Digital I/O	VDDQ_DDR	G12, G13, H11, H12, H13	DRAM I/O
	VDDQ_M0	F7, G7, H6	EBI I/O
	VDDQ_LCD	M8, N6	LCD I/O
	VDDQ_CI	L4	CAMIF I/O
	VDDQ_MMC	R7	MMC2 I/O
	VDDQ_AUD	U3	Audio I/O
	VDDQ_MSM	W6, Y6	Host IF I/O
	VDDQ_SYS0	L26	GPH, Main xtal I/O
	VDDQ_SYS2	V12	X-tal I/O
	VDDQ_SYS5	V17	Special clock I/O
	VDDQ_CAN	U18	CAN I/O
	VDDQ_EXT	M20, R21	JTAG I/O, MMC I/O
	VDDQ_UHOST	Y16	USB Host I/O
Internal Logic	VDD_RTC	V25	RTC
	VDD_INT	F8, H14, H15, H16, K15, L11, L13, L14, L15, T12, U12, U13, U14, U15, V13, V14, Y12	Internal logic
	VDD_ALIVE	J26	Alive logic
	VDD_ARM	F21, G20, G21, H18, H19, H20, H21, J20, J21, K16, K17, K20, K21, L17, L18, L21, M18	Cortex-A8 core
Analog & High Speed	VDD30_DAC_A	AC1	DAC analog part
	VDD30_DAC_D	AB7	DAC digital part
	VDD12_HDMI	Y10, AD7	HDMI Phy
	VDD18_MIPI	Y15	MIPI
	VDD12_MIPI	AA13, AA14	MIPI
	VDD18_MIPI_PLL	Y13	MIPI
	VDD_APLL	AB17	APLL

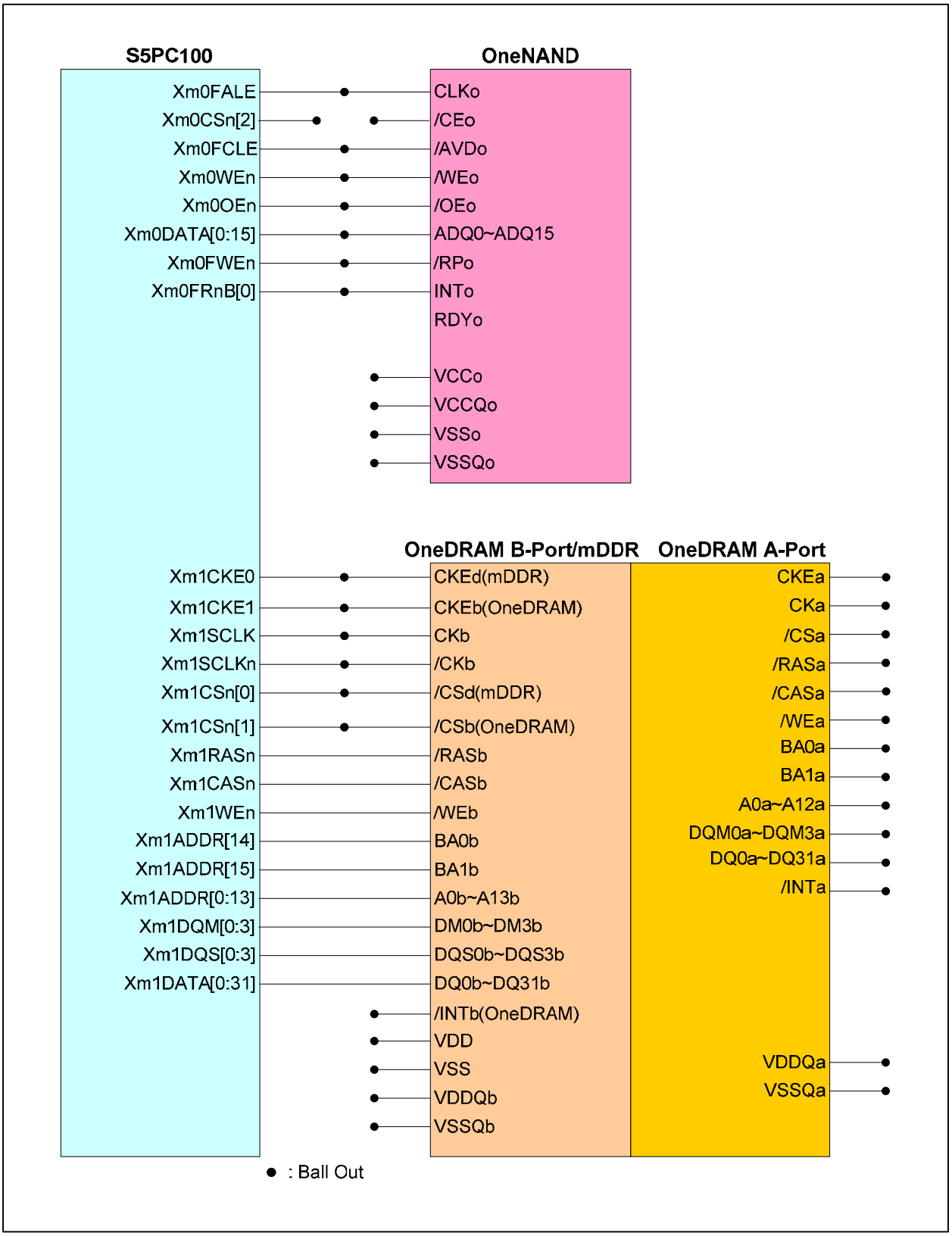
Power Group	Pin Name	Ball	Description
	VDD_MPLL	AA17	MPLL
	VDD_EPLL	AA16	EPLL
	VDD_HPLL	AD12	HPLL
	VDD33_UOTG	Y18	USB OTG Phy
	VDD12_UOTG	AE20	USB OTG Phy
	VDD33_ADC	AB25	ADC

Table 1.1-6 S5PC100 Ground Pin to Ball Assignment

Power Group	Pin Name	Ball
Digital I/O	VSSQ_DDR	A1, A2, A26, A27, B1, B2, B26, B27, C3, C9, C15, C21, C25, D9, D15, F14, F25, G6, G14, G15, H10, H22, J3, J4, J22, K22, K24, K25, L16, L22, M13, M14, M15, M16, M17, M22, N7, N12, N16, N17, P12, P16, P17, P26, P27, R2, R12, R16, T13, T14, T15, T20, U16, V1, V2, V3, V16, Y11, Y21, Y25, Y26, AB14, AC3, AD15, AD21, AE4, AE11, AE15, AE21, AF1, AF4, AF26, AF27, AG1, AG2, AG26, AG27
	VSSQ_M0	
	VSSQ_LCD	
	VSSQ_CI	
	VSSQ_MMC	
	VSSQ_AUD	
	VSSQ_MSM	
	VSSQ_SYS0	
	VSSQ_SYS2	
	VSSQ_SYS5	
	VSSQ_CG	
	VSSQ_CAN	
	VSSQ_EXT	
	VSSQ_UH	
Internal Logic	VSS_INT	
	VSS_ALIVE	
	VSS_ARM	
Analog/HSI	VSS30_DAC_A	AB2
	VSS30_DAC_D	AA7
	VSS12_HDMI	AB8, AB9
	VSS12_MIPI	AB12, AB13
	VSS18_MIPI_PLL	AB11
	VSS_APLL	AA18
	VSS_MPLL	Y17
	VSS_EPLL	AB16

	VSS_HPLL	AA12
	VSSQ_UOTG	AA19
	VSS_UOTG	AF21
	VSS_ADC	Y24

1.4 MCP CONNECTION



2 PACKAGE DIMENSION

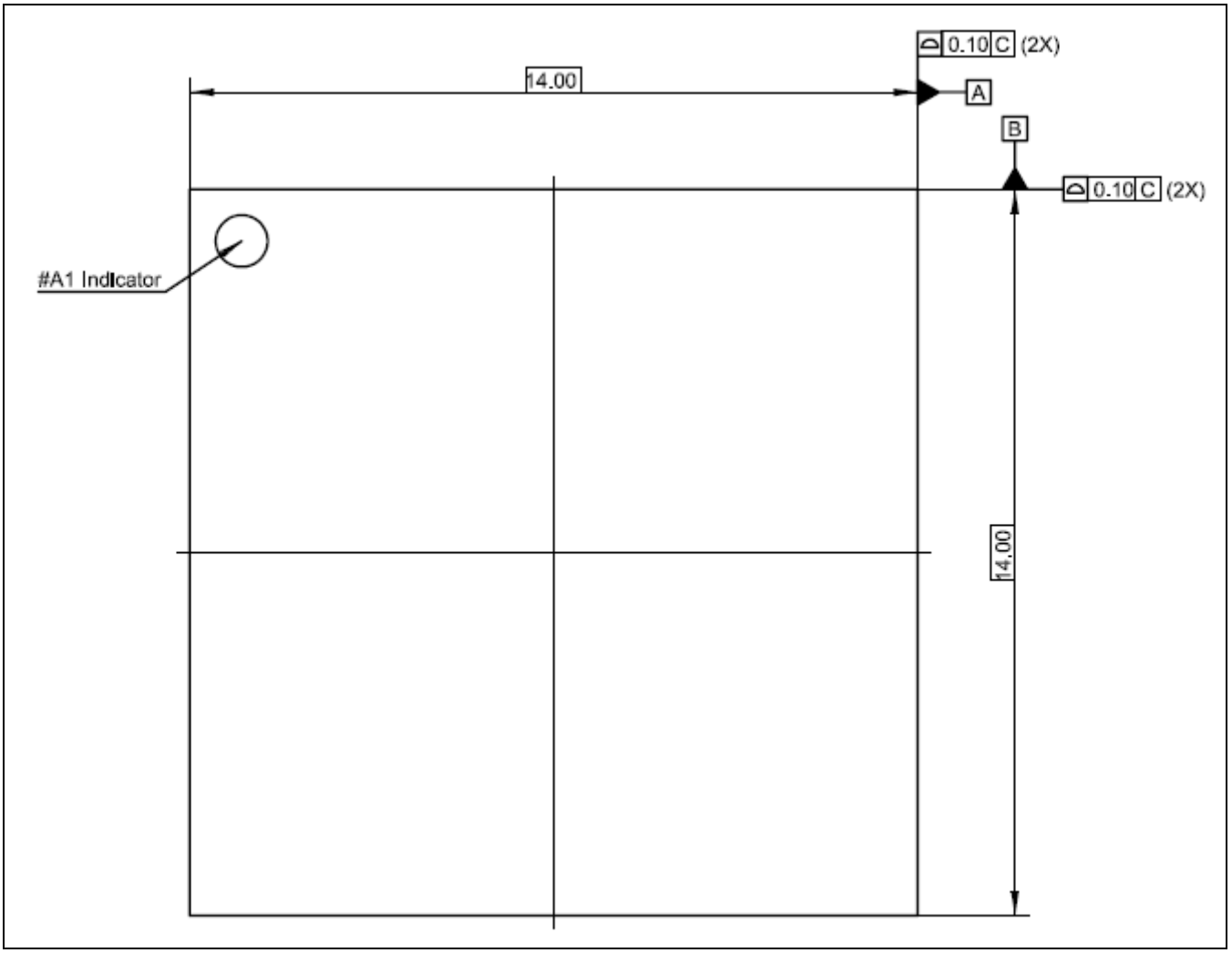


Figure 12.2- 1 S5PC100 Package Dimension (580-FCFBGA)- Top View

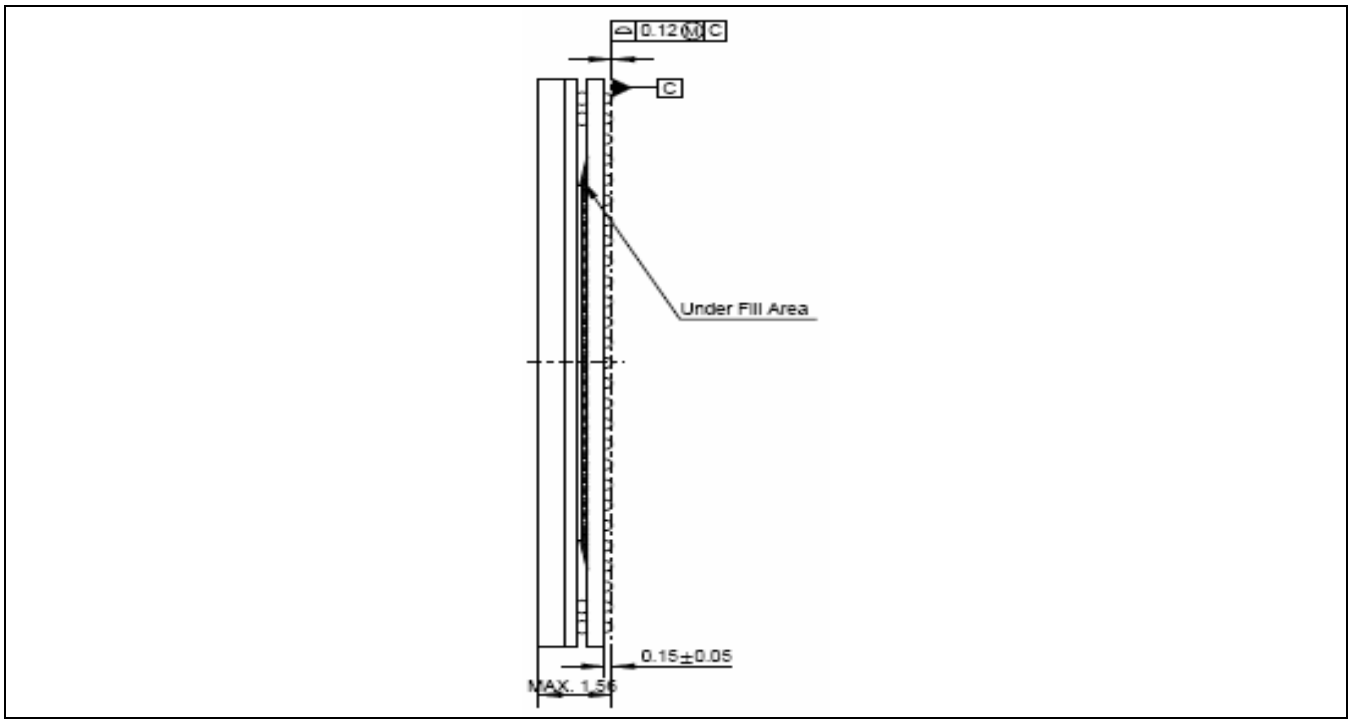


Figure 12.2- 2 S5PC100 Package Dimension (580-FCFBGA) – Side View

1.3 BALL MAP & SIZE

1 PIN ASSIGNMENT

1.1 PIN ASSIGNMENT DIAGRAM - 521-BALL FCFBGA (SINGLE CHIP)

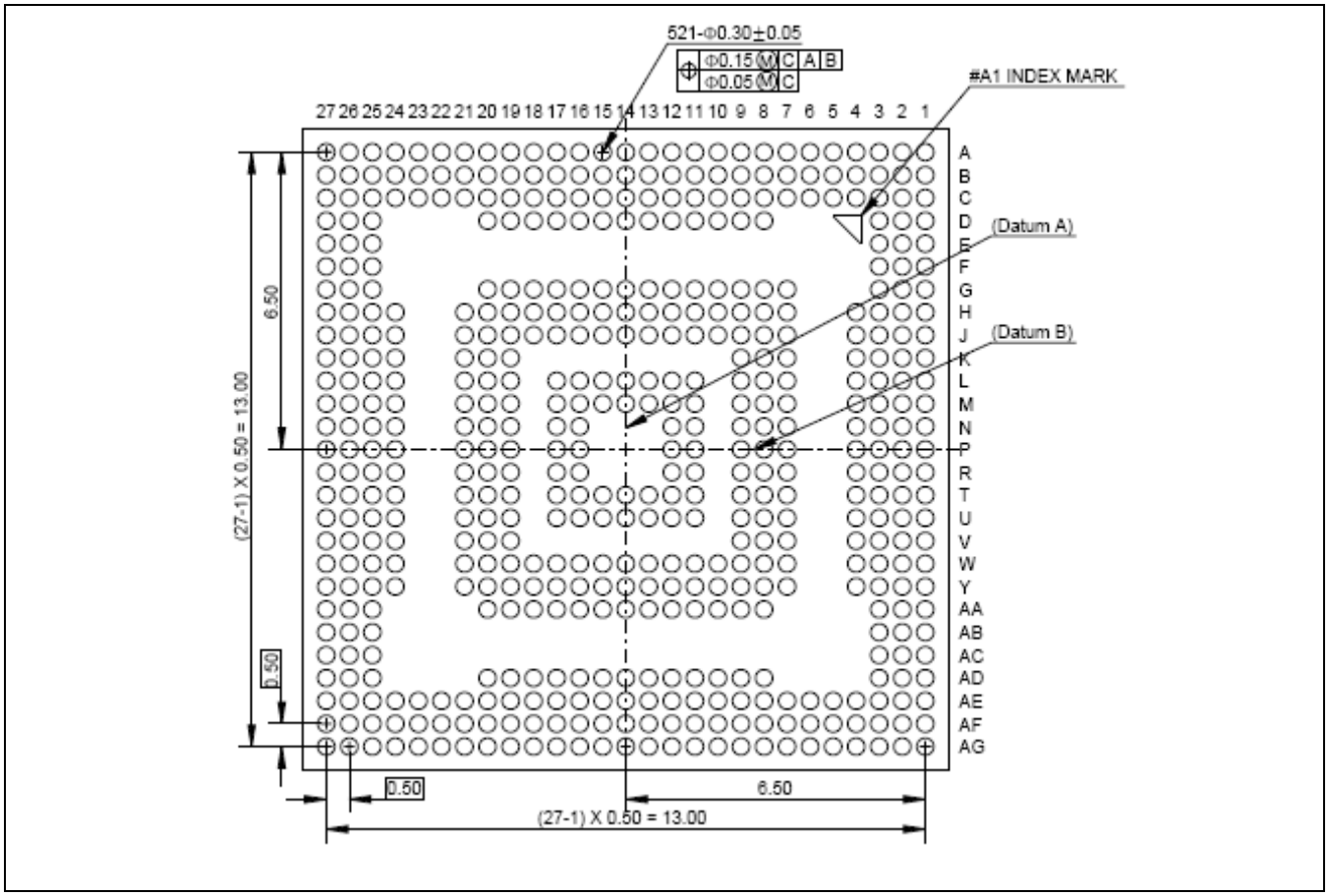


Figure 1.3 - 1 S5PC100 Pin Assignment (521-FCFBGA) Bottom View

1.2 PIN NUMBER ORDER

Table 1.3-1 S5PC100 521 FCFBGA Pin Assignment – Pin Number Order (1/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
A1	VSS	B14	Xm1DATA[16]	C27	Xm1DATA[4]	G12	Xm1ADDR[12]
A2	VSS	B15	Xm1ADDR[5]	D1	Xm0ADDR[20]	G13	Xm1ADDR[14]
A3	Xm0DATA[7]	B16	Xm1ADDR[9]	D2	Xm0BEn[0]	G14	Xm1ADDR[0]
A4	Xm0DATA[10]	B17	Xm1ADDR[7]	D3	VSS	G15	Xm1RASn
A5	Xm0DATA[2]	B18	Xm1NSCLK	D8	VDD_INT	G16	Xm1ADDR[10]
A6	Xm0DATA[3]	B19	Xm1WEn	D9	Xm0REG	G17	VDD_ARM
A7	Xm0DATA[8]	B20	Xm1ADDR[3]	D10	Xm1DATA[24]	G18	VDD_ARM
A8	Xm0DATA[0]	B21	Xm1DQM[1]	D11	Xm1DATA[26]	G19	VDD_ARM
A9	Xm0FRnB[1]	B22	Xm1DATA[14]	D12	Xm1DATA[25]	G20	VDD_ARM
A10	Xm0IORDY	B23	Xm1DATA[10]	D13	Xm1DATA[22]	G25	XpwmTOUT[0]
A11	Xm1DQS[3]	B24	Xm1DATA[9]	D14	Xm1DATA[19]	G26	XuRXD[0]
A12	Xm1DQSn[3]	B25	Xm1DQS[0]	D15	VDD_INT	G27	XuRXD[3]
A13	Xm1DATA[23]	B26	Xm1DQSn[0]	D16	VSS	H1	Xm0ADDR[9]
A14	Xm1DQM[2]	B27	VSS	D17	Xm1CSn[1]	H2	Xm0ADDR[5]
A15	Xm1DQS[2]	C1	Xm0WEn	D18	Xm1ADDR[15]	H3	Xm0DATA[6]
A16	Xm1DQSn[2]	C2	Xm0FALE	D19	VSS	H4	Xm0FRnB[0]
A17	Xm1ADDR[4]	C3	Xm0DATA[14]	D20	VDDQ_DDR	H7	XefFSOURCE_0
A18	Xm1ADDR[13]	C4	Xm0DATA[12]	D25	Xm1DQM[0]	H8	Xm0RESET
A19	Xm1ADDR[8]	C5	Xm0DATA[11]	D26	Xm1DATA[2]	H9	Xm0CDn
A20	Xm1DATA[13]	C6	Xm0DATA[15]	D27	Xm1DATA[3]	H10	Xm0BEn[1]
A21	Xm1DATA[12]	C7	Xm0FRnB[2]	E1	Xm0ADDR[0]	H11	Xm1ADDR[11]
A22	Xm1DATA[15]	C8	VSS	E2	Xm0ADDR[11]	H12	Xm1CKE[1]
A23	Xm1DQS[1]	C9	Xm0IOWRn	E3	VDDQ_M0	H13	Xm1CKE[0]
A24	Xm1DQSn[1]	C10	Xm1DATA[31]	E25	XpwmTOUT[1]	H14	Xm1CASn
A25	Xm1DATA[7]	C11	Xm1DATA[28]	E26	Xm1DATA[1]	H15	VDDQ_DDR
A26	VSS	C12	Xm1DQM[3]	E27	Xm1DATA[0]	H16	VSS
A27	VSS	C13	Xm1DATA[17]	F1	Xm0ADDR[12]	H17	VSS
B1	VSS	C14	Xm1DATA[20]	F2	Xm0ADDR[4]	H18	VSS
B2	Xm0DATA[4]	C15	Xm1DATA[18]	F3	Xm0ADDR[18]	H19	VSS
B3	Xm0DATA[9]	C16	VDDQ_DDR	F25	Xi2c0SCL	H20	VSS
B4	Xm0DATA_RDn	C17	Xm1SCLK	F26	XuRTSn[0]	H21	VDD_ARM
B5	Xm0DATA[13]	C18	Xm1CSn[0]	F27	XuRTSn[1]	H24	XuTXD[0]
B6	Xm0DATA[1]	C19	Xm1ADDR[1]	G1	Xm0CSn[3]	H25	XuRXD[2]
B7	Xm0FRnB[3]	C20	Xm1DATA[11]	G2	Xm0ADDR[13]	H26	XuRXD[1]
B8	Xm0CSn[2]	C21	VDDQ_DDR	G3	Xm0ADDR[10]	H27	XuTXD[2]
B9	VDDQ_M0	C22	VSS	G7	XefVGATE_0	J1	Xm0DATA[5]
B10	Xm1DATA[30]	C23	Xm1ADDR[2]	G8	Xm0CFWEn	J2	Xm0ADDR[1]
B11	Xm1DATA[29]	C24	Xm1DATA[8]	G9	Xm0CFOEn	J3	Xm0ADDR[2]
B12	Xm1DATA[27]	C25	Xm1DATA[6]	G10	Xm0INPACKn	J4	Xm0ADDR[16]
B13	Xm1DATA[21]	C26	Xm1DATA[5]	G11	Xm1ADDR[6]	J7	Xm0INTRQ

Table 1.3-2 S5PC100 521 FCFBGA Pin Assignment – Pin Number Order (2/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
J8	Xm0FCLE	L12	VSS	N9	XciRESET	R17	VDD_INT
J9	Xm0FWEn	L13	VSS	N11	Xm0ADDR[3]	R19	XEINT[7]
J10	Xm0FREn	L14	VSS	N12	VDD_INT	R20	XEINT[6]
J11	VDDQ_DDR	L15	VSS	N16	VSS	R21	Xmmc0DATA[7]
J12	Xm0OEn	L16	VDD_ARM	N17	VDDQ_EXT	R24	Xmmc1CDn
J13	Xm0WAITn	L17	VDD_ARM	N19	XjTMS	R25	Xmmc1DATA[1]
J14	Xm0IORDn	L19	Xmmc0CLK	N20	XspiMISO[1]	R26	VDDQ_EXT
J15	VSS	L20	XspiCSn[1]	N21	Xmmc0DATA[2]	R27	XDDR2SEL
J16	VDD_ARM	L21	XjTDO	N24	Xmmc0DATA[0]	T1	XciDATA[4]
J17	VDD_ARM	L24	XpwmTOUT[2]	N25	Xmmc1CMD	T2	XciFIELD
J18	VDD_ARM	L25	XuCTSn[1]	N26	XjTRSTn	T3	XvVDEN
J19	VDD_ARM	L26	Xi2c0SDA	N27	Xmmc0CDn	T4	XvVCLK
J20	VSS	L27	XspiCLK[1]	P1	XvVD[15]	T7	XiemSPWI
J21	VDD_ARM	M1	XvVD[1]	P2	XvVD[9]	T8	XciVSYNC
J24	XuTXD[1]	M2	XvVD[10]	P3	XvVD[8]	T9	XusbDRVVBUS
J25	XuTXD[3]	M3	XvVD[2]	P4	XvVD[4]	T11	XvVD[0]
J26	XuCLK	M4	Xm0CSn[5]	P7	XvVD[13]	T12	VDDQ_MMC
J27	XspiCSn[0]	M7	XvVD[17]	P8	XciDATA[0]	T13	VSS
K1	XvVD[21]	M8	XvVD[12]	P9	VDDQ_LCD	T14	VSS
K2	Xm0ADDR[6]	M9	XvVD[20]	P11	XvVD[16]	T15	VSS
K3	VDDQ_M0	M11	VDD_INT	P12	XvVD[11]	T16	VSS
K4	VSS	M12	VDD_INT	P16	VSS	T17	VDD_INT
K7	Xm0ADDR[7]	M13	VDD_INT	P17	VDD_INT	T19	XEINT[24]
K8	Xm0ADDR[17]	M14	VDD_INT	P19	XjTCK	T20	XnRESET
K9	Xm0ADDR[19]	M15	VDD_INT	P20	XEINT[16]	T21	XEINT[5]
K19	VDD_ARM	M16	VSS	P21	Xmmc0DATA[1]	T24	XEINT[3]
K20	VSS	M17	VDD_ARM	P24	Xmmc1DATA[3]	T25	XEINT[22]
K21	Xmmc0DATA[5]	M19	Xmmc0DATA[4]	P25	Xmmc1DATA[0]	T26	VSS
K24	Xi2c1SCL	M20	Xmmc0DATA[3]	P26	Xmmc1CLK	T27	XjTDI
K25	XuCTSn[0]	M21	XspiMOSI[1]	P27	Xmmc1DATA[2]	U1	XciDATA[2]
K26	Xi2c1SDA	M24	XspiMOSI[0]	R1	XvVD[3]	U2	VDDQ_CI
K27	XspiCLK[0]	M25	XspiMISO[0]	R2	XvVSYNC	U3	XciPCLK
L1	XvVD[7]	M26	Xmmc0DATA[6]	R3	XvHSYNC	U4	XciCLKenb
L2	XvVD[14]	M27	Xmmc0CMD	R4	XciDATA[5]	U7	Xi2s0LRCK
L3	Xm0CSn[1]	N1	Xm0CSn[4]	R7	Xi2s0SDO[0]	U8	Xi2s1SDO
L4	Xm0CSn[0]	N2	XvVD[23]	R8	XciHREF	U9	XciDATA[7]
L7	Xm0ADDR[8]	N3	XvVD[22]	R9	VDD_INT	U11	XmsmADDR[9]
L8	Xm0ADDR[15]	N4	VDDQ_LCD	R11	XciDATA[6]	U12	Xi2s0SDI
L9	Xm0ADDR[14]	N7	XvVD[18]	R12	XvVD[5]	U13	XvVD[19]
L11	VSS	N8	XvVD[6]	R16	VSS	U14	Xmmc2DATA[3]

Table 1.3-3 S5PC100 521 FCFBGA Pin Assignment – Pin Number Order (3/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
U15	XnRSTOUT	W19	XEINT[28]	AA15	XEINT[23]	AE1	XdacCOMP
U16	N.C	W20	XOM[0]	AA16	VDDQ_SYS2	AE2	VSS30_DAC_A
U17	VDDQ_SYS5	W21	XEINT[21]	AA17	XEINT[31]	AE3	XdacIREF
U19	XNFMOD[1]	W24	XEINT[20]	AA18	N.C	AE4	VDD30_DAC_A
U20	VDDQ_CAN	W25	VDDQ_SYS0	AA19	XEINT[12]	AE5	XmsmADDR[6]
U21	XEINT[2]	W26	XEINT[19]	AA20	XrtcXTO	AE6	VDD_MSM
U24	XPWRRGTON	W27	XEINT[26]	AA25	VDD_RTC	AE7	XNFMOD[5]
U25	XEINT[18]	Y1	Xi2s0CDCLK	AA26	XOM[3]	AE8	XNFMOD[3]
U26	XEINT[17]	Y2	Xi2s1SCLK	AA27	XXTO	AE9	VDD12_HDMI
U27	XEINT[27]	Y3	Xmmc2DATA[2]	AB1	Xi2s0SCLK	AE10	XhdmiREXT
V1	XciDATA[1]	Y4	VSS	AB2	Xi2s1LRCK	AE11	VDD_HPLL
V2	XciDATA[3]	Y7	XmsmDATA[0]	AB3	XmsmDATA[14]	AE12	MIPI_PLLVSS18
V3	XiemSCLK	Y8	XmsmDATA[9]	AB25	XadcAIN[7]	AE13	VDD12_MIPI
V4	Xmmc2DATA[1]	Y9	XmsmDATA[7]	AB26	VSS_ADC	AE14	VDD12_MIPI
V7	VDD_MSM	Y10	XmsmADDR[5]	AB27	XadcAIN[6]	AE15	MIPI_VSS
V8	Xi2s1SDI	Y11	XNFMOD[2]	AC1	XmsmDATA[10]	AE16	XmipiDN[3]
V9	XmsmDATA[12]	Y12	XmsmRn	AC2	XmsmDATA[15]	AE17	XmipiDP[3]
V19	XEINT[8]	Y13	XmsmADDR[2]	AC3	XmsmDATA[11]	AE18	VSS_EPLL
V20	XEINT[9]	Y14	XmsmADDR[8]	AC25	XadcAIN[8]	AE19	XuhDP
V21	XOM[2]	Y15	XEINT[30]	AC26	VDD33_ADC	AE20	XuhDN
V24	XEINT[1]	Y16	XEINT[10]	AC27	XadcAIN[9]	AE21	VDD_INT
V25	XEINT[0]	Y17	XEINT[13]	AD1	XmsmDATA[8]	AE22	VDD33_UOTG
V26	XjDBGSEL	Y18	XEINT[11]	AD2	XmsmDATA[3]	AE23	XusbVBUS
V27	XEINT[4]	Y19	XOM[1]	AD3	XmsmADDR[12]	AE24	XadcAIN[0]
W1	Xi2s0SDO[2]	Y20	XnBATF	AD8	VSS12_HDMI	AE25	XadcVref
W2	Xi2s0SDO[1]	Y21	XrtcXTI	AD9	VDD12_HDMI	AE26	XadcAIN[1]
W3	Xmmc2CDn	Y24	XOM[4]	AD10	VSS	AE27	XadcAIN[2]
W4	Xi2s1CDCLK	Y25	VDD_ALIVE	AD11	VDD_INT	AF1	VSS
W7	XmsmDATA[5]	Y26	XNFMOD[0]	AD12	MIPI_VSS	AF2	XdacOUT[0]
W8	XmsmDATA[13]	Y27	XXTI	AD13	N.C	AF3	XdacVREF
W9	XmsmDATA[6]	AA1	Xmmc2DATA[0]	AD14	N.C	AF4	VDD30_DAC_D
W10	XmsmDATA[4]	AA2	Xmmc2CLK	AD15	VDD_APLL	AF5	XmsmADDR[7]
W11	XmsmIRQn	AA3	VDDQ_AUD	AD16	VSS_APLL	AF6	XmsmADDR[0]
W12	XmsmWEn	AA8	XmsmDATA[2]	AD17	VDD_EPLL	AF7	XmsmCSn
W13	XmsmADDR[3]	AA9	XmsmADDR[10]	AD18	VDD_MPLL	AF8	XNFMOD[4]
W14	Xmmc2CMD	AA10	XmsmADDR[11]	AD19	VSS_MPLL	AF9	X27mXTO
W15	XEINT[25]	AA11	XmsmDATA[1]	AD20	VSS	AF10	X27mXTI
W16	XEINT[15]	AA12	XmsmADDR[4]	AD25	XadcAIN[3]	AF11	VSS12_HDMI
W17	XEINT[14]	AA13	XCLKOUT	AD26	XadcAIN[4]	AF12	VSS_HPLL
W18	XEINT[29]	AA14	XmsmADDR[1]	AD27	XadcAIN[5]	AF13	VDD18_MIPI_PLL

NOTE: U16 pin is for test purpose only. Should be pulled up.
NOTE: AD13, AD14 pins are for test purpose only. Do not connect.

Table 1.1-4 S5PC100 521 FCFBGA Pin Assignment – Pin Number Order (4/4)

Ball	Pin Name	Ball	Pin Name	Ball	Pin Name	Ball	Pin Name
AF14	VDD18_MIPI	AG2	VSS	AG17	XmipiTXCN		
AF15	XmipiDN[1]	AG3	XdacOUT[1]	AG18	XmipiTXCP		
AF16	XmipiDP[1]	AG4	XdacOUT[2]	AG19	XmipiRXCN		
AF17	XmipiDN[2]	AG5	VSS30_DAC_D	AG20	XmipiRXCP		
AF18	XmipiDP[2]	AG6	XhdmiTX2N	AG21	VSS_UHOST		
AF19	XmipiDN[4]	AG7	XhdmiTX2P	AG22	VDD_UHOST		
AF20	XmipiDP[4]	AG8	XhdmiTX1N	AG23	XusbDP		
AF21	XusbID	AG9	XhdmiTX1P	AG24	XusbDM		
AF22	VDD12_UOTG	AG10	XhdmiTX0N	AG25	VSS		
AF23	VSS12_UOTG	AG11	XhdmiTX0P	AG26	VSS		
AF24	XusbREXT	AG12	XhdmiTXCN	AG27	VSS		
AF25	XusbXTO	AG13	XhdmiTXCP				
AF26	XusbXTI	AG14	XmipiReg_cap				
AF27	VSS	AG15	XmipiDN[0]				
AG1	VSS	AG16	XmipiDP[0]				

1.3 POWER PINS

Table 1.1-5 S5PC100 Power Pin to Ball Assignment (1/2)

Power Group	Pin Name	Ball	Description
Digital I/O	VDDQ_DDR	C16, C21, D20, H15, J11	DRAM I/O
	VDDQ_M0	B9, E3, K3	Memory(EBI) I/O
	VDDQ_LCD	N4, P9	LCD I/O
	VDDQ_CI	U2	CAMIF I/O
	VDDQ_MMC	T12	MMC2 I/O
	VDDQ_AUD	AA3	Audio I/O
	VDDQ_MSM	AE6, V7	Host IF I/O
	VDDQ_SYS0	W25	GPH, Main xtal I/O
	VDDQ_SYS2	AA17	X-tal I/O
	VDDQ_SYS5	U17	Special clock I/O
	VDDQ_CAN	U20	CAN I/O
	VDDQ_EXT	N17, R26	JTAG I/O, MMC0,1 I/O
	VDDQ_UHOST	AG22	USB Host I/O
Internal Logic	VDD_RTC	AA25	RTC
	VDD_INT	AD11, AE21, D15, D8, M11, M12, M13, M14,M15, N12, P17, R17, R9, T17	Internal logic
	VDD_ALIVE	Y25	Alive logic
	VDD_ARM	G17, G18, G19, G20, H21, J16, J17, J18, J19, J21, K19, L16, L17, M17	Cortex-A8 core
Analog & High Speed	VDD30_DAC_A	AE4	DAC analog part
	VDD30_DAC_D	AF4	DAC digital part
	VDD12_HDMI	AD9, AE9	HDMI Phy
	VDD18_MIPI	AF14	MIPI
	VDD12_MIPI	AE13, AE14	MIPI
	VDD18_MIPI_PLL	AF13	MIPI
	VDD_APLL	AD15	APLL
	VDD_MPLL	AD18	MPLL
	VDD_EPLL	AD17	EPLL
	VDD_HPLL	AE11	HPLL
	VDD33_UOTG	AE22	USB OTG Phy
	VDD12_UOTG	AF22	USB OTG Phy
	VDD33_ADC	AC26	ADC

Table 1.1-6 S5PC100 Power Pin to Ball Assignment (2/2)

Power Group	Pin Name	Ball
Digital I/O	VSSQ_DDR	A1, A2, A26, A27, AD10, AD20, AF1, AF27, AG1, AG2, AG26, AG27, B1, B27, C22, C8, D16, D19, D3, H16, H17, H18, H19, H20, J15, J20, K20, K4, L11, L12, L13, L14, L15, M16, N16, P16, R16, T13, T14, T15, T16, T26, Y4
	VSSQ_M0	
	VSSQ_LCD	
	VSSQ_CI	
	VSSQ_MMC	
	VSSQ_AUD	
	VSSQ_MSM	
	VSSQ_SYS0	
	VSSQ_SYS2	
	VSSQ_SYS5	
	VSSQ_CG	
	VSSQ_CAN	
	VSSQ_EXT	
	VSSQ_UHOST	
Internal Logic	VSS_INT	
	VSS_ALIVE	
	VSS_ARM	
Analog/HSI	VSS30_DAC_A	AE2
	VSS30_DAC_D	AG5
	VSS12_HDMI	AD8, AF11
	VSS12_MIPI	AD12, AE15
	VSS18_MIPI_PLL	AE12
	VSS_APLL	AD16
	VSS_MPLL	AD19
	VSS_EPLL	AE18
	VSS_HPLL	AF12
	VSS33_UOTG	AG25
	VSS12_UOTG	AF23
	VSS33_ADC	AB26

2 PACKAGE DIMENSION

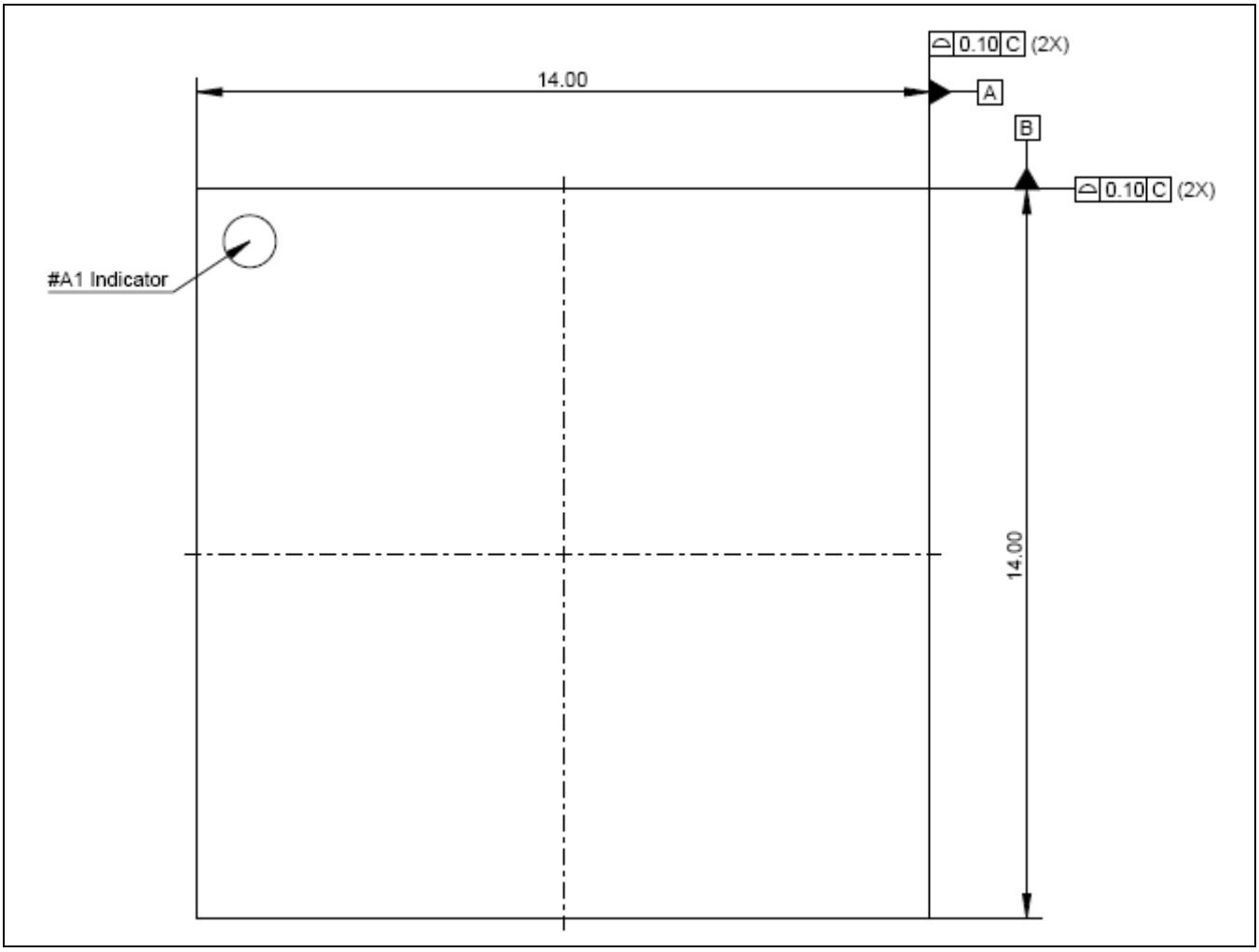


Figure 1.3- 1 S5PC100 Package Dimension (521-FCFBGA) – Top View

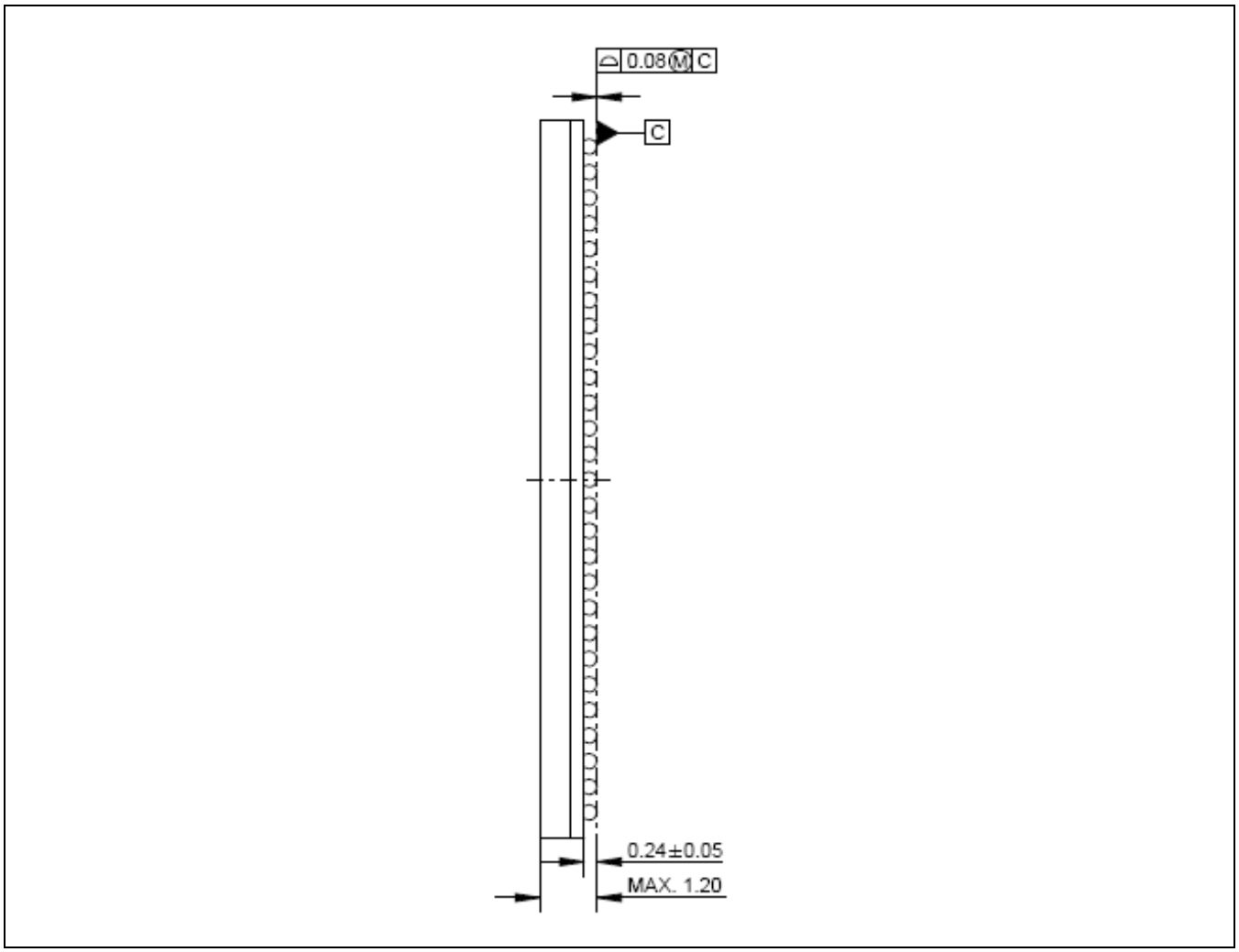


Figure 1.3- 2 S5PC100 Package Dimension (521-FCFBGA) – Side View

2.1 IROM CODE

1 OVERVIEW

The booting device can one of the followings,

- NAND Flash
- OneNAND
- MMC or SD Memory including Movi-NAND and iNAND
- USB

If there are more than one instance of the controller of the booting device, only the first instance of the controller is used for booting.

S5PC100 has 32KB ROM (iROM) and 96KB SRAM (iRAM) inside the chip. At the system reset, the program execution starts at iROM.

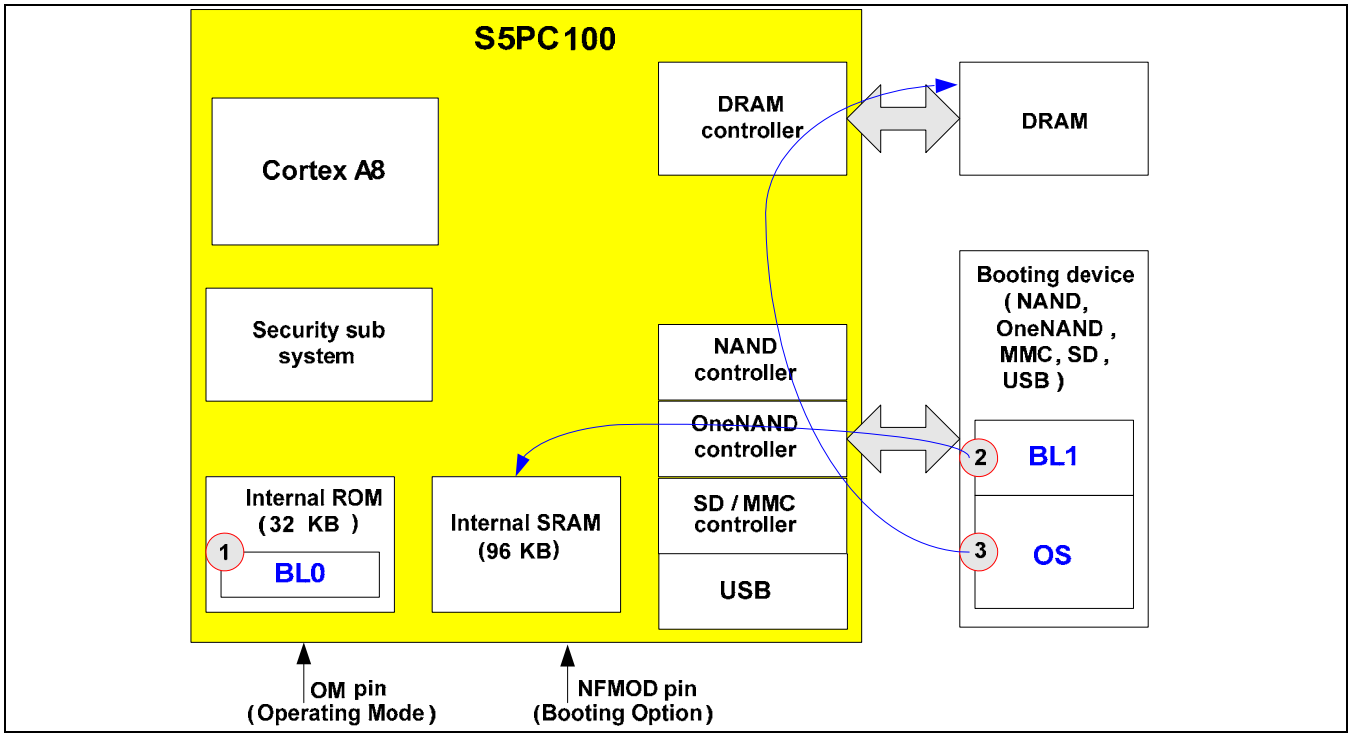


Figure 2.6-1 Booting Block Diagram

The boot loader is divided into the BL0(1st boot loader) and the BL1(2nd boot loader).

- BL0 which is placed in iROM loads BL1 from the booting device to iRAM. The booting device is determined by the OM and NFMOD pins. In addition, according to the secure boot key values, BL0 checks the integrity of BL1.
- BL1 should initializes the DRAM controller. After initializing the DRAM controller, it loads the OS image from the booting device to DRAM. And, according to the secure boot key values, BL1 checks the integrity of the OS image.

The operation on the booting time is presented in Figure 2.6-1.

2 FUNCTIONAL DESCRIPTION

2.1 RESET STATUS

There are several scenarios for system reset such as hardware reset, watchdog reset, software reset, and wake up from power down modes. For each scenario, the 'must' functions are summarized as Table 2.6-1.

Table 2.6-1 Functions on BL0

	PLL on	BL1 loading
Reset (XnRESET, Watchdog, Software)	O	O
Wakeup from ESLEEP	O	O
Wakeup from SLEEP	O	O
Wakeup from DEEP_STOP (Top-domain on)	X	X
Wakeup from DEEP_STOP (Top-domain off with retention)	X	X
Wakeup from DEEP_STOP (Top-domain off)	O	O
Wakeup from DEEP_IDLE (Top-domain on)	X	X
Wakeup from DEEP_IDLE (Top-domain off with retention)	X	X
Wakeup from DEEP_IDLE (Top-domain off)	O	O

On Hardware reset and watchdog reset, the system has to do full booting including BL1 and OS image loading. On ESLEEP case, the set product does not ensure that DRAM memory's contents are preserved. So, full booting is also required. These reset status which requires full booting is classified as reset group0.

Because SLEEP mode is different from ESLEEP in that DRAM memory's contents are preserved, it does not need an OS image loading to DRAM. However, SoC internal power is not supplied during SLEEP mode, all contents in internal SRAM is not preserved. So, BL1 should be loaded again. This reset status is classified as reset group1.

On SW reset, the contents of both internal SRAM and external DRAM are preserved. So, any boot loader loading is not required at all. Although the top block's power is gated in DEEP_STOP and DEEP_IDLE mode, If the internal SRAM is retention so that boot loader re-loading is not necessary. But internal SRAM is off then iROM reload BL1. Wake up from DEEP_STOP and wake up from DEEP_IDLE statuses are classified as reset group2.

When system enters into all power down modes except ESLEEP, current system status should be saved to safe memory region such as DRAM so that the system continues processing seamlessly after waking up from power

down modes. Because the system is entered into ESLEEP mode on the emergency case such as battery fault, there is no safe memory space on the whole system. So, any status information will not be saved to DRAM before entering ESLEEP mode.

Finally, the restoring pervious state function is required on wake up from SLEEP, wake up from DEEP_STOP, and wake up from DEEP_IDLE.

2.2 FUNCTIONAL SEQUENCE

Full booting sequence in BL0 is as follows,

1. Initialize the PLL & Clock setting with fixed value
2. Initialize the stack and heap region.
3. Initialize the Instruction Cache controller.
4. Load BL1 from the booting device to iRAM.*
5. If secure booting is enabled, execute integrity check.
6. If integrity check passes, then jump to 0x34010.(First 4word is reserved)
7. If integrity check fails, then it stops.

* NOTE1. In case of SD/MMC, iROM code load 9KB at 0x34000 from end of memory device.

* NOTE2. In case of OneNAND and NAND, iROM code load 16KB at 0x34000 from the beginning(Block 0) of memory device block.

* NOTE3. Bad Block Information is in 6th byte of Spare Area(Block#0) in case of 512 byte Page NAND device. And the rest of NAND device(2KB page size) has Bad Block Information in the first byte of Spare Area.

2.3 CLOCK SETTING

The setting values for PLL's and clock dividers are as follows.

- APLL: M=400, P=4, S=1
- MPLL: M=106, P=4, S=2
- EPLL: M=110, P=4,S=4

- DIV_{APLL} = 1 (1/2 divider)
- DIV_{ARM} = 0 (1/1 divider)
- DIV_{D0_BUS} = 2 (1/3 divider)
- DIV_{D1_BUS} = 0 (1/1 divider)
- DIV_{MPLL} = 0 (1/1 divider)
- DIV_{MPLL2} = 0 (1/1 divider)

Table 2.6-2 shows the clock frequencies after the initialization of the PLL's in BL0.

Table 2.6-2 BL0's Clock Speed

X-tal(MHz)	ARM Clock (MHz)	HCLK_D0(MHz)	HCLK_D1(MHz)	EPLL Clock (MHz)

12 Mhz	300.0 Mhz	100.0 Mhz	79.5 Mhz	20.6 Mhz

If there occurs an error in BL0, BL0 writes the error code at the Inform register (0xE010841C) and go to the USB booting. The error codes are as follows.

Error Code	Description
1	OM[2:1] is not 0x0(NAND), 0x1(OneNAND), 0x2(MoviNAND)
2	AES Decrytion of BL1 failed
3	Integrity of BL1failed
4	Unsupported NAND
5	Loading BL1 from NAND failed
6	Loading BL1 from OneNAND failed
7	Loading BL1 from MoviNAND failed

2.4 SECURE BOOTING

The operational sequence in BL0 is as follows,

1. Check the integrity of the RSA public key on the booting device with the hash value of the RSA key, which is stored on the e-fuse ROM. 128 bits of e-fuse are used for this integrity check.
2. Load BL1 to iRAM
3. Check the integrity of BL1 with the RSA public key

Figure 2.6- 2 shows the above sequence.

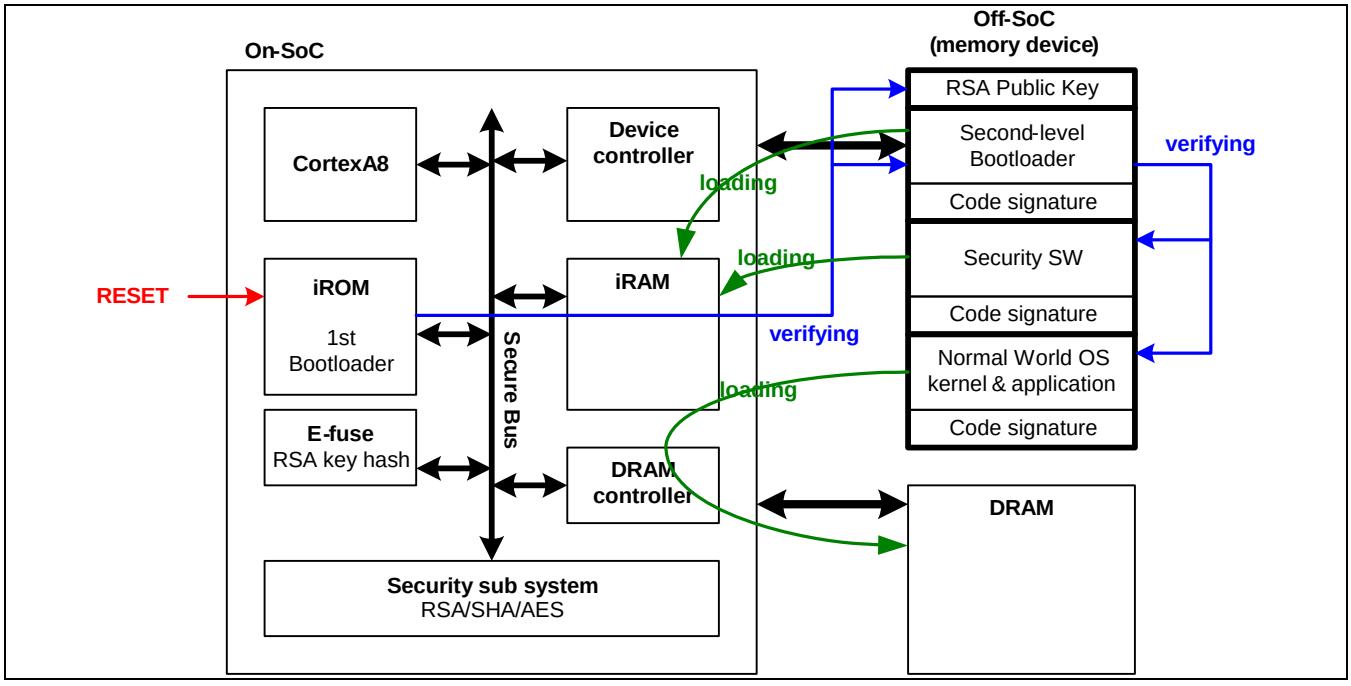


Figure 2.6- 2 Secure Booting Diagram

3 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
OM[4:0]	Input	OM[4]: 0 = normal mode, 1 = test mode OM[3]: 0 = 1st boot loader in iROM , 1 = reserved OM[2:1] : 00 = 2nd boot loader in NAND flash 01 = 2nd boot loader in OneNAND 10 = 2nd boot loader in MMC 11 = Reserved OM[0] : APLL/MPLL input selection 0 = XXTI 1 = XusbXTI	XOM [4:0]	Dedicated
NFMOD [5:0]	Input	NFMOD[5] : 0 = 2nd booting from the device selected by OM[2:1] 1 = 2nd booting from USB When OM[2:1] = 00 (2nd boot loader in NAND Flash), NFMOD[1:0]: 00 = Small Block (512page), 10 = Large Block (2048page), 11 = Dlarge Block (4096page) NFMOD[2]: 0 = 3 addr cycles (for small block) or 4 cycles (for large block) 1 = 4 addr cycles (for small block) or 5 cycles (for large block)	XNFMOD [5:0]	Muxed

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
PRO_ID	0xE000_0000	R	Product information (ID, package, revision)	0x43100???
OMR	0xE000_0004	R	Booting information (Booting modes)	0x0003????

4.1 PRODUCT ID REGISTER (PRO_ID, R, ADDRESS = 0XE000_0000)

Field	Bit	Description	Reset Value
Product ID	[31:12]	Product ID The product ID allocated to S5PC100 is "0x43100"	0x43100
Reserved	[11:10]	Should be zero	0x0
Package mode	[9:8]	Package mode 2'0X : discrete 2'1X : POP	Device dependent
Main revision number	[7:4]	Main revision number*	0xF**
Sub revision number	[3:0]	Sub revision number*	0xF**

NOTE: *: The sub revision number represents the revision number which does not need to change the SW, while the main revision number represents the revision number need to change the SW..

NOTE: **: PRO_ID register [7:0] depends on the e-fuse ROM value. Although the reset value of PRO_ID register [7:0] is 0xFF, as power on sequence is progressing, the e-fuse ROM values are loaded to the registers. Hence, SW cannot read register reset value itself, but it can only read the loaded current e-fuse ROM values. An the e-fuse ROM has main & sub revision numbers.

4.2 OPERATING MODE REGISTER (OMR, R, ADDRESS = 0XE000_0004)

Field	Bit	Description	Reset Value
Reserved	[31:18]	Read as zero	0x0
Reserved	[17]	Read as one	1
Reserved	[16]	Read as one	1
Reserved	[15:14]	Read as zero	0x0
NFMOD	[13:8]	The value of the XNFMOD[5:0] pad	Pin value
Reserved	[7:5]	Read as zero	0x0
OM	[4:0]	The value of the XOM[4:0] pad	Pin value

2.2 PAD CONTROL

1 OVERVIEW

S5PC100 includes 203 multi-functional input/ output pins and 108 memory pins. There are 29 general groups and 2 memory groups as listed below:

- GPA0: 8 in/out pin – 2xUART with flow control
- GPA1: 5 in/out pin – 2xUART without flow control or 1xUART with flow control, 1x IrDA
- GPB: 8 in/out pin – 2x SPI
- GPC: 5 in/out pin – I2S, PCM, AC97
- GPD: 7 in/out pin – 2xI2C, PWM, External DMA request, SPDIF
- GPE0,1: 14 in/out pin – Camera I/F 0, MMC channel1(GPE0 supports 4-bit mode MMC, if 8-bit mode is needed, use GPE1 for another 4-bit data channel)
- GPF0,1,2,3: 28 in/out pin – LCD I/F
- GPG0,1,2,3: 25 in/out pin – 3xMMC channel (Channel 0 supports 4-bit and 8-bit mode, but channel 1, channel 2 supports only 4-bit mode), SPI, I2S, PCM, SPDIF
- GPH0,1,2,3: 32 in/out pin – CAM IF channel, Key pad, External Wake-up (up-to 32-bit)
- GPI: 8 in/out pin – Booting option, PWI
- GPJ0,1,2,3,4: 33 in/out pin – Modem IF, HSI, ATA
- GPK0,1,2,3: 30 in/out pin – Control signals of EBI(SMC, NF, CF, OneNAND)
- GPL: 37 in/out memory pin – EBI(refer chapter 5,6 for detail description of EBI configuration) & GPIO
- MP1: 71 DRAM pin
- ETC0,1,2,3 : XjTRSTn, XjTMS, XjTDI, XI2S0, XusbXTI/XusbXTO, XrtcXTI/XrtcXTO …

2 FEATURES

The GPIO features include:

- Controls 173 GPIO Interrupts, 141 Non wake-up Interrupts and 32 Wake-up Interrupts
- 130 multi-functional input / output ports
- Controls pin states in Sleep Mode except GPH0, GPH1, GPH2, and GPH3

3 DRIVE STRENGTH

Output driver are subdivided into Type A, Type B, Type OSC_A and Type OSC_B.

3.1 OUTPUT DRIVER TYPES

Driver Types	PAD
A	GPA0, GPA1, GPB, GPC, GPD, GPE0, GPE1, GPF0, GPF1, GPF2, GPF3, GPG0, GPG2, GPG3, GPH0, GPH1, GPH2, GPH3, GPI, GPJ0, GPJ1, GPJ2, GPJ3, GPJ4, GPK0, GPK1, GPK2, GPK3
B	MP0
OSC_A	ETC4[5](XrtcXTI/XrtcXTO)
OSC_B	ETC4[0](X27mXTI/X27mXTO), ETC4[2](XXTI/XXTO), ETC4[4](XusbXTI/XusbXTO)

3.2 DC CURRENTS OF OUTPUT DRIVER TYPE A, B, OSC_A AND OSC_B (VDD=3.3V±0.3V)

Driver Type		VDD=3.00V T=125℃	VDD=3.30V T=25℃	VDD=3.60V T=-40℃
A	1X	1.7mA	2.6mA	3.4mA
	2X	3.4mA	5.2mA	6.9mA
	3X	5.2mA	7.8mA	10.4mA
	4X	6.9mA	10.5mA	13.9mA
B	1X	3.4mA	5.2mA	6.9mA
	2X	6.9mA	10.5mA	13.9mA
	3X	10.4mA	15.7mA	20.9mA
	4X	13.9mA	21.0mA	27.9mA
OSC_A	1X	3.9uA	6.6uA	9.7uA
	2X	7.8uA	13.0uA	19.0uA
OSC_B	1X	77uA	110uA	130uA
	2X	0.93mA	1.38mA	1.82mA
	3X	1.9mA	2.8mA	3.7mA
	4X	5.8mA	8.5mA	11.2mA

3.3 DC CURRENTS OF OUTPUT DRIVER TYPE A, B, OSC_A AND OSC_B (VDD=2.5V±0.2V)

Driver Type		VDD=2.30V T=125℃	VDD=2.50V T=25℃	VDD=2.70V T=-40℃
A	1X	1.4mA	2.2mA	3.1mA
	2X	2.8mA	4.4mA	6.2mA
	3X	4.2mA	6.6mA	9.3mA
	4X	5.7mA	8.9mA	12.4mA
B	1X	2.8mA	4.4mA	6.2mA
	2X	5.7mA	8.9mA	12.4mA
	3X	8.5mA	13.3mA	18.6mA
	4X	11.4mA	17.8mA	24.8mA
OSC_A	1X	3.1uA	5.4uA	8.5uA
	2X	6.2uA	11.0uA	17.0uA
OSC_B	1X	65uA	95uA	120uA
	2X	0.76mA	1.18mA	1.63mA
	3X	1.6mA	2.4mA	3.3mA
	4X	4.7mA	7.3mA	10mA

3.4 DC CURRENTS OF OUTPUT DRIVER TYPE A, B, OSC_A AND OSC_B (VDD=1.8V±0.15V)

Driver Type		VDD=1.65V T=125℃	VDD=1.80V T=25℃	VDD=1.95V T=-40℃
A	1X	1.1mA	1.6mA	2.4mA
	2X	2.0mA	3.3mA	4.8mA
	3X	3.0mA	4.9mA	7.3mA
	4X	4.0mA	6.6mA	9.7mA
B	1X	2.0mA	3.3mA	4.8mA
	2X	4.0mA	6.6mA	9.7mA
	3X	6.0mA	9.9mA	14.6mA
	4X	8.1mA	13.2mA	19.4mA
OSC_A	1X	2uA	3.7uA	6.2uA
	2X	4.1uA	7.5uA	12.0uA
OSC_B	1X	48uA	75uA	100uA
	2X	0.55mA	0.89mA	1.29mA
	3X	1.1mA	1.8mA	2.6mA
	4X	3.4mA	5.4mA	7.9mA

4 I/O DESCRIPTION

GPIO consists of two part, alive-part and off-part. In Alive-part power is supplied on sleep mode, but in off-part it is not the same. Therefore, the registers in alive-part keep their values during sleep mode.

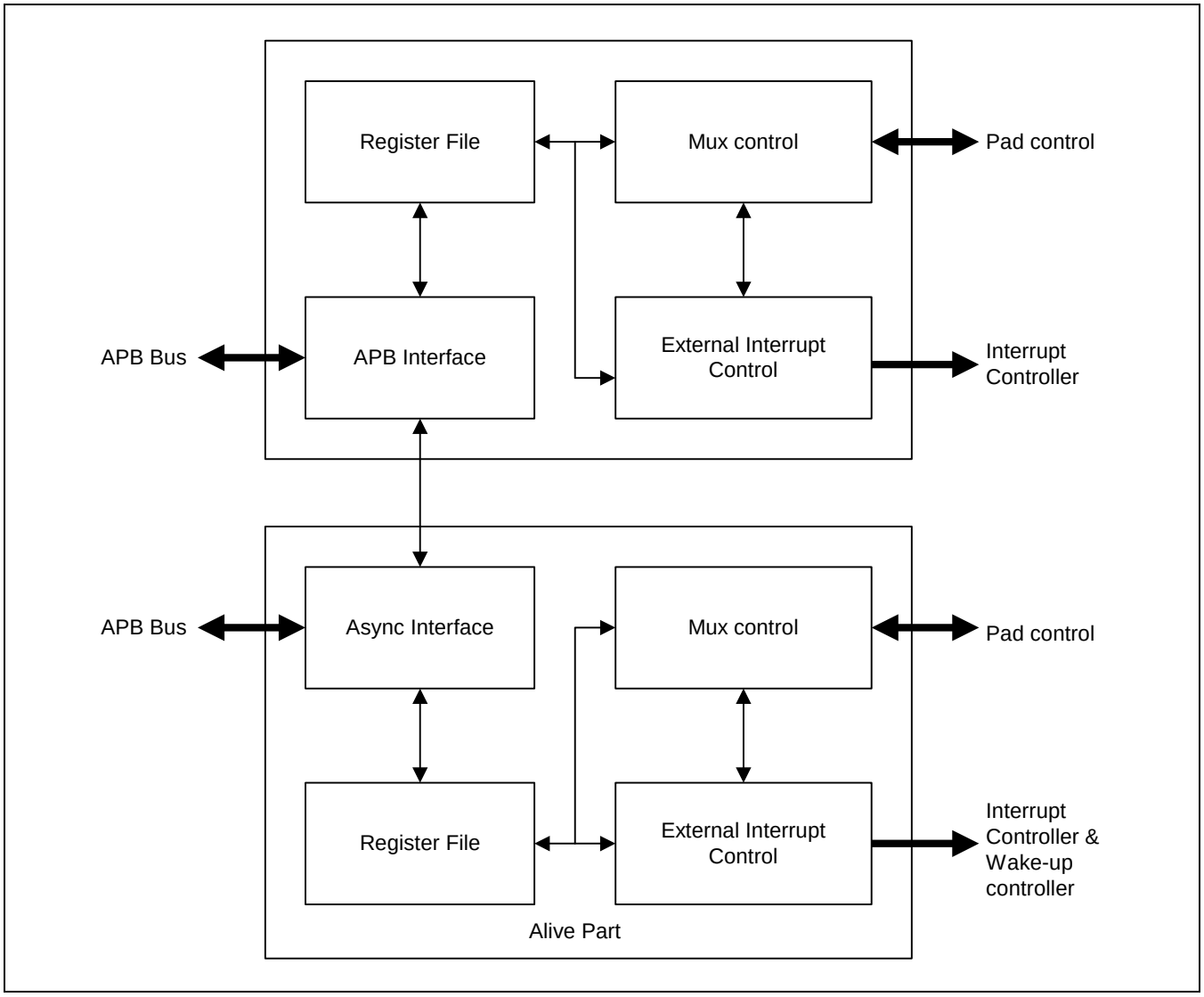


Figure 2.2-1 GPIO Block Diagram

4.1 PIN SUMMARY

Type	Function Description
A1	Control at power down mode is possible, power down mode is release by IO_RET_RELEASE[31]
A2	Control at power down mode is possible, power down mode is released automatically
B	Alive I/O
C	On power-down, they are set to be input without pull-up/down. On wakeup, they are set to the reset value
E1	The power-down state is "Previous State". On wakeup, S/W should release the power-down state
H	High-Z

The values of the Pull & I/O column in the below table are the state at the reset. The meaning of the values of the VDD column in the below table are described in the above table

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XuRXD[0]	GPA0[0] / UART0_RXD	PD	I	A1	VDDQ_EXT
XuTXD[0]	GPA0[1] / UART0_TXD	PD	I	A1	VDDQ_EXT
XuCTSn[0]	GPA0[2] / UART0_CTSn	PD	I	A1	VDDQ_EXT
XuRTSn[0]	GPA0[3] / UART0_RTSn	PD	I	A1	VDDQ_EXT
XuRXD[1]	GPA0[4] / UART1_RXD	PD	I	A1	VDDQ_EXT
XuTXD[1]	GPA0[5] / UART1_TXD	PD	I	A1	VDDQ_EXT
XuCTSn[1]	GPA0[6] / UART1_CTSn	PD	I	A1	VDDQ_EXT
XuRTSn[1]	GPA0[7] / UART1_RTSn	PD	I	A1	VDDQ_EXT
XuRXD[2]	GPA1[0] / UART2_RXD	PD	I	A1	VDDQ_EXT
XuTXD[2]	GPA1[1] / UART2_TXD	PD	I	A1	VDDQ_EXT
XuRXD[3]	GPA1[2] / UART3_RXD / UART2_CTSn / IrDA_RXD	PD	I	A1	VDDQ_EXT
XuTXD[3]	GPA1[3] / UART3_TXD / UART2_RTSn / IrDA_TXD	PD	I	A1	VDDQ_EXT
XuCLK	GPA1[4] / UARTCLK / IrDA_SDBW	PD	I	A1	VDDQ_EXT
XspiMISO[0]	GPB[0] / SPI0_MISO	PD	I	A1	VDDQ_EXT
XspiCLK[0]	GPB[1] / SPI0_CLK	PD	I	A1	VDDQ_EXT
XspiMOSI[0]	GPB[2] / SPI0_MOSI	PD	I	A1	VDDQ_EXT
XspiCSn[0]	GPB[3] / SPI0_nCS	PD	I	A1	VDDQ_EXT
XspiMISO[1]	GPB[4] / SPI1_MISO	PD	I	A1	VDDQ_EXT
XspiCLK[1]	GPB[5] / SPI1_CLK	PD	I	A1	VDDQ_EXT
XspiMOSI[1]	GPB[6] / SPI1_MOSI	PD	I	A1	VDDQ_EXT
XspiCSn[1]	GPB[7] / SPI1_nCS	PD	I	A1	VDDQ_EXT
XI2S1SCLK	GPC[0] / I2S1_SCLK / PCM1_SCLK / AC97_BITCLK	PD	I	A1	VDDQ_AUD
XI2S1CDCLK	GPC[1] / I2S1_CDCLK / PCM1_EXTCLK / AC97_RESETn	PD	I	A1	VDDQ_AUD
XI2S1LRCK	GPC[2] / I2S1_LRCK / PCM1_FSYNC / AC97_SYNC	PD	I	A1	VDDQ_AUD
XI2S1SDI	GPC[3] / I2S1_SDI / PCM1_SIN / AC97_SDI	PD	I	A1	VDDQ_AUD
XI2S1SDO	GPC[4] / I2S1_SDO / PCM1_SOUT / AC97_SDO	PD	I	A1	VDDQ_AUD

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XpwmTOUT[0]	GPD[0] / TOUT0 / PWM_TCLK	PD	I	A1	VDDQ_EXT
XpwmTOUT[1]	GPD[1] / TOUT1 / EX_DMA_REQn	PD	I	A1	VDDQ_EXT
XpwmTOUT[2]	GPD[2] / TOUT2 / EX_DMA_ACKn	PD	I	A1	VDDQ_EXT
XI2C0SDA	GPD[3] / I2C0_SDA	PD	I	A1	VDDQ_EXT
XI2C0SCL	GPD[4] / I2C0_SCL	PD	I	A1	VDDQ_EXT
XI2C1SDA	GPD[5] / I2C1_SDA / SPDIF_0_OUT	PD	I	A1	VDDQ_EXT
XI2C1SCL	GPD[6] / I2C1_SCL / SPDIF_EXTCLK	PD	I	A1	VDDQ_EXT
XciPCLK	GPE0[0] / CAM_A_PCLK / SD1_CLK	PD	I	A1	VDDQ_CI
XciVSYNC	GPE0[1] / CAM_A_VSYNC / SD1_CDn	PD	I	A1	VDDQ_CI
XciHREF	GPE0[2] / CAM_A_HREF / SD1_CMD	PD	I	A1	VDDQ_CI
XciD[0]	GPE0[3] / CAM_A_D[0] / SD1_D[0]	PD	I	A1	VDDQ_CI
XciD[1]	GPE0[4] / CAM_A_D[1] / SD1_D[1]	PD	I	A1	VDDQ_CI
XciD[2]	GPE0[5] / CAM_A_D[2] / SD1_D[2]	PD	I	A1	VDDQ_CI
XciD[3]	GPE0[6] / CAM_A_D[3] / SD1_D[3]	PD	I	A1	VDDQ_CI
XciD[4]	GPE0[7] / CAM_A_D[4] / SD1_D[4]	PD	I	A1	VDDQ_CI
XciD[5]	GPE1[0] / CAM_A_D[5] / SD1_D[5]	PD	I	A1	VDDQ_CI
XciD[6]	GPE1[1] / CAM_A_D[6] / SD1_D[6]	PD	I	A1	VDDQ_CI
XciD[7]	GPE1[2] / CAM_A_D[7] / SD1_D[7]	PD	I	A1	VDDQ_CI
XciCLKenb	GPE1[3] / CAM_A_CLKOUT	PD	I	A1	VDDQ_CI
XciRESET	GPE1[4] / CAM_A_RESET	PD	I	A1	VDDQ_CI
XciFIELD	GPE1[5] / CAM_A_FIELD	PD	I	A1	VDDQ_CI
XvHSYNC	GPF0[0] / LCD_HSYNC / SYS_CS0 / VEN_HSYNC	PD	I	A1	VDDQ_LCD
XvVSYNC	GPF0[1] / LCD_VSYNC / SYS_CS1 / VEN_VSYNC	PD	I	A1	VDDQ_LCD
XvVDEN	GPF0[2] / LCD_VDEN / SYS_RS / VEN_HREF	PD	I	A1	VDDQ_LCD
XvVCLK	GPF0[3] / LCD_VCLK / SYS_WE / V601_CLK	PD	I	A1	VDDQ_LCD
XvVD[0]	GPF0[4] / LCD_VD[0] / SYS_VD[0] / VEN_D[0]	PD	I	A1	VDDQ_LCD
XvVD[1]	GPF0[5] / LCD_VD[1] / SYS_VD[1] / VEN_D[1]	PD	I	A1	VDDQ_LCD
XvVD[2]	GPF0[6] / LCD_VD[2] / SYS_VD[2] / VEN_D[2]	PD	I	A1	VDDQ_LCD
XvVD[3]	GPF0[7] / LCD_VD[3] / SYS_VD[3] / VEN_D[3]	PD	I	A1	VDDQ_LCD
XvVD[4]	GPF1[0] / LCD_VD[4] / SYS_VD[4] / VEN_D[4]	PD	I	A1	VDDQ_LCD
XvVD[5]	GPF1[2] / LCD_VD[5] / SYS_VD[5] / VEN_D[5]	PD	I	A1	VDDQ_LCD
XvVD[6]	GPF1[2] / LCD_VD[6] / SYS_VD[6] / VEN_D[6]	PD	I	A1	VDDQ_LCD
XvVD[7]	GPF1[3] / LCD_VD[7] / SYS_VD[7] / VEN_D[7]	PD	I	A1	VDDQ_LCD
XvVD[8]	GPF1[4] / LCD_VD[8] / SYS_VD[8] / V656_D[0]	PD	I	A1	VDDQ_LCD
XvVD[9]	GPF1[5] / LCD_VD[9] / SYS_VD[9] / V656_D[1]	PD	I	A1	VDDQ_LCD
XvVD[10]	GPF1[6] / LCD_VD[10] / SYS_VD[10] / V656_D[2]	PD	I	A1	VDDQ_LCD

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XvVD[11]	GPF1[7] / LCD_VD[11] / SYS_VD[11] / V656_D[3]	PD	I	A1	VDDQ_LCD
XvVD[12]	GPF2[0] / LCD_VD[12] / SYS_VD[12] / V656_D[4]	PD	I	A1	VDDQ_LCD
XvVD[13]	GPF2[1] / LCD_VD[13] / SYS_VD[13] / V656_D[5]	PD	I	A1	VDDQ_LCD
XvVD[14]	GPF2[2] / LCD_VD[14] / SYS_VD[14] / V656_D[6]	PD	I	A1	VDDQ_LCD
XvVD[15]	GPF2[3] / LCD_VD[15] / SYS_VD[15] / V656_D[7]	PD	I	A1	VDDQ_LCD
XvVD[16]	GPF2[4] / LCD_VD[16] / SYS_VD[16]	PD	I	A1	VDDQ_LCD
XvVD[17]	GPF2[5] / LCD_VD[17] / SYS_VD[17]	PD	I	A1	VDDQ_LCD
XvVD[18]	GPF2[6] / LCD_VD[18]	PD	I	A1	VDDQ_LCD
XvVD[19]	GPF2[7] / LCD_VD[19]	PD	I	A1	VDDQ_LCD
XvVD[20]	GPF3[0] / LCD_VD[20]	PD	I	A1	VDDQ_LCD
XvVD[21]	GPF3[1] / LCD_VD[21]	PD	I	A1	VDDQ_LCD
XvVD[22]	GPF3[2] / LCD_VD[22] / VSYNC_LDI / V656_CLK	PD	I	A1	VDDQ_LCD
XvVD[23]	GPF3[3] / LCD_VD[23] / SYS_OE / VEN_FIELD	PD	I	A1	VDDQ_LCD
Xmmc0CLK	GPG0[0] / SD0_CLK	PD	I	A1	VDDQ_EXT
Xmmc0CMD	GPG0[1] / SD0_CMD	PD	I	A1	VDDQ_EXT
Xmmc0D[0]	GPG0[2] / SD0_D[0]	PD	I	A1	VDDQ_EXT
Xmmc0D[1]	GPG0[3] / SD0_D[1]	PD	I	A1	VDDQ_EXT
Xmmc0D[2]	GPG0[4] / SD0_D[2]	PD	I	A1	VDDQ_EXT
Xmmc0D[3]	GPG0[5] / SD0_D[3]	PD	I	A1	VDDQ_EXT
Xmmc0D[4]	GPG0[6] / SD0_D[4]	PD	I	A1	VDDQ_EXT
Xmmc0D[5]	GPG0[7] / SD0_D[5]	PD	I	A1	VDDQ_EXT
Xmmc0D[6]	GPG1[0] / SD0_D[6]	PD	I	A1	VDDQ_EXT
Xmmc0D[7]	GPG1[1] / SD0_D[7]	PD	I	A1	VDDQ_EXT
Xmmc0CDn	GPG1[2] / SD0_CDn	PD	I	A1	VDDQ_EXT
Xmmc1CLK	GPG2[0] / SD1_CLK	PD	I	A1	VDDQ_EXT
Xmmc1CMD	GPG2[1] / SD1_CMD	PD	I	A1	VDDQ_EXT
Xmmc1D[0]	GPG2[2] / SD1_D[0]	PD	I	A1	VDDQ_EXT
Xmmc1D[1]	GPG2[3] / SD1_D[1]	PD	I	A1	VDDQ_EXT
Xmmc1D[2]	GPG2[4] / SD1_D[2]	PD	I	A1	VDDQ_EXT
Xmmc1D[3]	GPG2[5] / SD1_D[3]	PD	I	A1	VDDQ_EXT
Xmmc1CDn	GPG2[6] / SD1_CDn	PD	I	A1	VDDQ_EXT
Xmmc2CLK	GPG3[0] / SD2_CLK / SPI2_CLK / I2S2_SCLK / PCM0_SCLK	PD	I	A1	VDDQ_MMC
Xmmc2CMD	GPG3[1] / SD2_CMD / SPI2_nSS / I2S2_CDCLK / PCM0_EXTCLK	PD	I	A1	VDDQ_MMC
Xmmc2DATA[0]	GPG3[2] / SD2_D[0] / SPI2_MISO / I2S2_LRCK / PCM0_FSYNC	PD	I	A1	VDDQ_MMC
Xmmc2DATA[1]	GPG3[3] / SD2_D[1] / SPI2_MOSI / I2S2_SDI / PCM0_SIN	PD	I	A1	VDDQ_MMC
Xmmc2DATA[2]	GPG3[4] / SD2_D[2] / I2S2_SDO / PCM0_SOUT	PD	I	A1	VDDQ_MMC

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
Xmmc2DATA[3]	GPG3[5] / SD2_D[3] / SPDIF_OUT	PD	I	A1	VDDQ_MMC
Xmmc2CDn	GPG3[6] / SD2_CDn / SPDIF_EXTCLK	PD	I	A1	VDDQ_MMC
XEINT[0]	GPH0[0] / WKUP_INT[0]	PD	I	B	VDDQ_SYS0
XEINT[1]	GPH0[1] / WKUP_INT[1]	PD	I	B	VDDQ_SYS0
XEINT[2]	GPH0[2] / WKUP_INT[2]	PD	I	B	VDDQ_SYS0
XEINT[3]	GPH0[3] / WKUP_INT[3]	PD	I	B	VDDQ_SYS0
XEINT[4]	GPH0[4] / WKUP_INT[4]	PD	I	B	VDDQ_SYS0
XEINT[5]	GPH0[5] / WKUP_INT[5]	PD	I	B	VDDQ_SYS0
XEINT[6]	GPH0[6] / WKUP_INT[6]	PD	I	B	VDDQ_SYS0
XEINT[7]	GPH0[7] / WKUP_INT[7]	PD	I	B	VDDQ_SYS0
XEINT[8]	GPH1[0] / WKUP_INT[8]	PD	I	B	VDDQ_SYS5
XEINT[9]	GPH1[1] / WKUP_INT[9]	PD	I	B	VDDQ_SYS5
XEINT[10]	GPH1[2] / WKUP_INT[10] / CG_REALIN	PD	I	B	VDDQ_SYS5
XEINT[11]	GPH1[3] / WKUP_INT[11] / CG_IMGIN	PD	I	B	VDDQ_SYS5
XEINT[12]	GPH1[4] / WKUP_INT[12] / CG_GPO[0]	PD	I	B	VDDQ_SYS5
XEINT[13]	GPH1[5] / WKUP_INT[13] / CG_GPO[1]	PD	I	B	VDDQ_SYS5
XEINT[14]	GPH1[6] / WKUP_INT[14] / CG_GPO[2]	PD	I	B	VDDQ_SYS5
XEINT[15]	GPH1[7] / WKUP_INT[15] / CG_GPO[3]	PD	I	B	VDDQ_SYS5
XEINT[16]	GPH2[0] / WKUP_INT[16] / KP_COL[0] / CAM_B_D[0]	PD	I	B	VDDQ_SYS0
XEINT[17]	GPH2[1] / WKUP_INT[17] / KP_COL[1] / CAM_B_D[1]	PD	I	B	VDDQ_SYS0
XEINT[18]	GPH2[2] / WKUP_INT[18] / KP_COL[2] / CAM_B_D[2]	PD	I	B	VDDQ_SYS0
XEINT[19]	GPH2[3] / WKUP_INT[19] / KP_COL[3] / CAM_B_D[3]	PD	I	B	VDDQ_SYS0
XEINT[20]	GPH2[4] / WKUP_INT[20] / KP_COL[4] / CAM_B_D[4]	PD	I	B	VDDQ_SYS0
XEINT[21]	GPH2[5] / WKUP_INT[21] / KP_COL[5] / CAM_B_D[5]	PD	I	B	VDDQ_SYS0
XEINT[22]	GPH2[6] / WKUP_INT[22] / KP_COL[6] / CAM_B_D[6]	PD	I	B	VDDQ_SYS0
XEINT[23]	GPH2[7] / WKUP_INT[23] / KP_COL[7] / CAM_B_D[7]	PD	I	B	VDDQ_SYS0
XEINT[24]	GPH3[0] / WKUP_INT[24] / KP_ROW[0] / CAM_B_PCLK	PD	I	B	VDDQ_SYS0
XEINT[25]	GPH3[1] / WKUP_INT[25] / KP_ROW[1] / CAM_B_VSYNC	PD	I	B	VDDQ_SYS0
XEINT[26]	GPH3[2] / WKUP_INT[26] / KP_ROW[2] / CAM_B_HREF	PD	I	B	VDDQ_SYS0
XEINT[27]	GPH3[3] / WKUP_INT[27] / KP_ROW[3] / CAM_B_FIELD	PD	I	B	VDDQ_SYS0
XEINT[28]	GPH3[4] / WKUP_INT[28] / KP_ROW[4] / CAN0_TX	PD	I	B	VDDQ_CAN
XEINT[29]	GPH3[5] / WKUP_INT[29] / KP_ROW[5] / CAN0_RX	PD	I	B	VDDQ_CAN
XEINT[30]	GPH3[6] / WKUP_INT[30] / KP_ROW[6] / CAN1_TX	PD	I	B	VDDQ_CAN
XEINT[31]	GPH3[7] / WKUP_INT[31] / KP_ROW[7] / CAN1_RX	PD	I	B	VDDQ_CAN
XiemSCLK	GPI[0] / IEM_SCLK	PD	I	A1	VDDQ_MMC
XiemSPWI	GPI[1] / IEM_SPWI	PD	I	A1	VDDQ_MMC

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XNFMOD[0]	GPI[2] / NFMOD[0]	-	I	C	VDDQ_SYS0
XNFMOD[1]	GPI[3] / NFMOD[1]	-	I	C	VDDQ_SYS0
XNFMOD[2]	GPI[4] / NFMOD[2]	PD	I	A1	VDDQ_SYS2
XNFMOD[3]	GPI[5] / NFMOD[3]	PD	I	A1	VDDQ_SYS2
XNFMOD[4]	GPI[6] / NFMOD[4]	PD	I	A1	VDDQ_SYS2
XNFMOD[5]	GPI[7] / NFMOD[5]	PD	I	A1	VDDQ_SYS2
XmsmADDR[0]	GPJ0[0] / MSM_A[0] / HSI_TXD / CF_A[0]	PD	I	A1	VDDQ_MSM
XmsmADDR[1]	GPJ0[1] / MSM_A[1] / HSI_TX_FLAG / CF_A[1]	PD	I	A1	VDDQ_MSM
XmsmADDR[2]	GPJ0[2] / MSM_A[2] / HSI_TX_WAKE / CF_A[2]	PD	I	A1	VDDQ_MSM
XmsmADDR[3]	GPJ0[3] / MSM_A[3] / HSI_TX_READY / CF_IORDY	PD	I	A1	VDDQ_MSM
XmsmADDR[4]	GPJ0[4] / MSM_A[4] / HSI_RXD / CF_INTRQ	PD	I	A1	VDDQ_MSM
XmsmADDR[5]	GPJ0[5] / MSM_A[5] / HSI_RX_FLAG / CF_INPACKn	PD	I	A1	VDDQ_MSM
XmsmADDR[6]	GPJ0[6] / MSM_A[6] / HSI_RX_WAKE/ CF_RESET	PD	I	A1	VDDQ_MSM
XmsmADDR[7]	GPJ0[7] / MSM_A[7] / HSI_RX_READY/ CF_REG	PD	I	A1	VDDQ_MSM
XmsmADDR[8]	GPJ1[0] / MSM_A[8]	PD	I	A1	VDDQ_MSM
XmsmADDR[9]	GPJ1[1] / MSM_A[9]	PD	I	A1	VDDQ_MSM
XmsmADDR[10]	GPJ1[2] / MSM_A[10]	PD	I	A1	VDDQ_MSM
XmsmADDR[11]	GPJ1[3] / MSM_A[11]	PD	I	A1	VDDQ_MSM
XmsmADDR[12]	GPJ1[4] / MSM_A[12]	PD	I	A1	VDDQ_MSM
XmsmDATA[0]	GPJ2[0] / MSM_D[0] / CF_D[0]	PD	I	A1	VDDQ_MSM
XmsmDATA[1]	GPJ2[1] / MSM_D[1] / CF_D[1]	PD	I	A1	VDDQ_MSM
XmsmDATA[2]	GPJ2[2] / MSM_D[2] / CF_D[2]	PD	I	A1	VDDQ_MSM
XmsmDATA[3]	GPJ2[3] / MSM_D[3] / CF_D[3]	PD	I	A1	VDDQ_MSM
XmsmDATA[4]	GPJ2[4] / MSM_D[4] / CF_D[4]	PD	I	A1	VDDQ_MSM
XmsmDATA[5]	GPJ2[5] / MSM_D[5] / CF_D[5]	PD	I	A1	VDDQ_MSM
XmsmDATA[6]	GPJ2[6] / MSM_D[6] / CF_D[6]	PD	I	A1	VDDQ_MSM
XmsmDATA[7]	GPJ2[7] / MSM_D[7] / CF_D[7]	PD	I	A1	VDDQ_MSM
XmsmDATA[8]	GPJ3[0] / MSM_D[8] / CF_D[8]	PD	I	A1	VDDQ_MSM
XmsmDATA[9]	GPJ3[1] / MSM_D[9] / CF_D[9]	PD	I	A1	VDDQ_MSM
XmsmDATA[10]	GPJ3[2] / MSM_D[10] / F_D[10]	PD	I	A1	VDDQ_MSM
XmsmDATA[11]	GPJ3[3] / MSM_D[11] / F_D[11]	PD	I	A1	VDDQ_MSM
XmsmDATA[12]	GPJ3[4] / MSM_D[12] / CF_D[12]	PD	I	A1	VDDQ_MSM
XmsmDATA[13]	GPJ3[5] / MSM_D[13] / CF_D[13]	PD	I	A1	VDDQ_MSM
XmsmDATA[14]	GPJ3[6] / MSM_D[14] / CF_D[14]	PD	I	A1	VDDQ_MSM
XmsmDATA[15]	GPJ3[7] / MSM_D[15] / CF_D[15]	PD	I	A1	VDDQ_MSM
XmsmCSn	GPJ4[0] / MSM_CSn / CF_nCS[0]	PD	I	A1	VDDQ_MSM

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XmsmWEn	GPJ4[1] / MSM_WEn / CF_nCS[1]	PD	I	A1	VDDQ_MSM
XmsmRn	GPJ4[2] / MSM_REn / CF_IORDn	PD	I	A1	VDDQ_MSM
XmsmIRQn	GPJ4[3] / MSM_IRQn / CF_IOWRn	PD	I	A1	VDDQ_MSM
Xm0CSn[0]	GPK0[0] / SMC_nCS[0]	-	OH	A2	VDDQ_M0
Xm0CSn[1]	GPK0[1] / SMC_nCS[1]	-	OH	A2	VDDQ_M0
Xm0CSn[2]	GPK0[2] / SMC_nCS[2] / NF_nCS[0] / OND_nCS[0]	-	OH	A2	VDDQ_M0
Xm0CSn[3]	GPK0[3] / SMC_nCS[3] / NF_nCS[1] / OND_nCS[1]	-	OH	A2	VDDQ_M0
Xm0CSn[4]	GPK0[4] / SMC_nCS[4] / NF_nCS[2] / CF_nCS[0]	-	OH	A2	VDDQ_M0
Xm0CSn[5]	GPK0[5] / SMC_nCS[5] / NF_nCS[3] / CF_nCS[1]	-	OH	A2	VDDQ_M0
Xm0OEn	GPK0[6] / EBI_OEn	-	OH	A2	VDDQ_M0
Xm0WEn	GPK0[7] / EBI_WEn	-	OH	A2	VDDQ_M0
Xm0BEn[0]	GPK1[0] / EBI_BEn[0]	-	OH	A2	VDDQ_M0
Xm0BEn[1]	GPK1[1] / EBI_BEn[1]	-	OH	A2	VDDQ_M0
Xm0WAITn	GPK1[2] / SMC_WAITn	PU	I	A2	VDDQ_M0
Xm0DATA_RDn	GPK1[3] / EBI_DATA_RDn	-	OH	A2	VDDQ_M0
Xm0CFOEn	GPK1[4] / CF_OEn	-	OH	A2	VDDQ_M0
Xm0CFWEn	GPK1[5] / CF_WEn	-	OH	A2	VDDQ_M0
Xm0FCLE	GPK2[0] / NF_CLE / OND_AVALID	-	OL	A2	VDDQ_M0
Xm0FALE	GPK2[1] / NF_ALE / OND_SMCLK	-	OL	A2	VDDQ_M0
Xm0FWEn	GPK2[2] / NF_WEn / OND_RPn	-	OH	A2	VDDQ_M0
Xm0FREn	GPK2[3] / NF_REn	-	OH	A2	VDDQ_M0
Xm0FRnB[0]	GPK2[4] / NF_RnB[0] / OND_INT[0]	PD	I	A2	VDDQ_M0
Xm0FRnB[1]	GPK2[5] / NF_RnB[1] / OND_INT[1]	PD	I	A2	VDDQ_M0
Xm0FRnB[2]	GPK2[6] / NF_RnB[2]	PU	I	A2	VDDQ_M0
Xm0FRnB[3]	GPK2[7] / NF_RnB[3]	PU	I	A2	VDDQ_M0
Xm0IORDY	GPK3[0] / CF_IORDY	PD	I	A2	VDDQ_M0
Xm0INTRQ	GPK3[1] / CF_INTRQ	PD	I	A2	VDDQ_M0
Xm0RESET	GPK3[2] / CF_RESET	-	OL	A2	VDDQ_M0
Xm0INPACKn	GPK3[3] / CF_INPACKn	PU	I	A2	VDDQ_M0
Xm0REG	GPK3[4] / CF_REG	-	OL	A2	VDDQ_M0
Xm0CDn	GPK3[5] / CF_CDn	PU	I	A2	VDDQ_M0
Xm0IORDn	GPK3[6] / CF_IORDn	-	OH	A2	VDDQ_M0
Xm0IOWRn	GPK3[7] / CF_IOWRn	-	OH	A2	VDDQ_M0
Xm0ADDR[0]	GPL0[0] / EBI_A[1]	-	OL	A2	VDDQ_M0
Xm0ADDR[1]	GPL0[1] / EBI_A[1]	-	OL	A2	VDDQ_M0
Xm0ADDR[2]	GPL0[2] / EBI_A[2]	-	OL	A2	VDDQ_M0

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
Xm0ADDR[3]	GPL0[3] / EBI_A[3]	-	OL	A2	VDDQ_M0
Xm0ADDR[4]	GPL0[4] / EBI_A[4]	-	OL	A2	VDDQ_M0
Xm0ADDR[5]	GPL0[5] / EBI_A[5]	-	OL	A2	VDDQ_M0
Xm0ADDR[6]	GPL0[6] / EBI_A[6]	-	OL	A2	VDDQ_M0
Xm0ADDR[7]	GPL0[7] / EBI_A[7]	-	OL	A2	VDDQ_M0
Xm0ADDR[8]	GPL1[0] / EBI_A[8]	-	OL	A2	VDDQ_M0
Xm0ADDR[9]	GPL1[1] / EBI_A[9]	-	OL	A2	VDDQ_M0
Xm0ADDR[10]	GPL1[2] / EBI_A[10]	-	OL	A2	VDDQ_M0
Xm0ADDR[11]	GPL1[3] / EBI_A[11]	-	OL	A2	VDDQ_M0
Xm0ADDR[12]	GPL1[4] / EBI_A[12]	-	OL	A2	VDDQ_M0
Xm0ADDR[13]	GPL1[5] / EBI_A[13]	-	OL	A2	VDDQ_M0
Xm0ADDR[14]	GPL1[6] / EBI_A[14]	-	OL	A2	VDDQ_M0
Xm0ADDR[15]	GPL1[7] / EBI_A[15]	-	OL	A2	VDDQ_M0
Xm0ADDR[16]	GPL2[0] / EBI_A[16]	-	OL	A2	VDDQ_M0
Xm0ADDR[17]	GPL2[1] / EBI_A[17]	-	OL	A2	VDDQ_M0
Xm0ADDR[18]	GPL2[2] / EBI_A[18]	-	OL	A2	VDDQ_M0
Xm0ADDR[19]	GPL2[3] / EBI_A[19]	-	OL	A2	VDDQ_M0
Xm0ADDR[20]	GPL2[4] / EBI_A[20]	-	OL	A2	VDDQ_M0
Xm0DATA[0]	GPL2[5] / EBI_D[0]	-	I	A2	VDDQ_M0
Xm0DATA[1]	GPL2[6] / EBI_D[1]	-	I	A2	VDDQ_M0
Xm0DATA[2]	GPL2[7] / EBI_D[2]	-	I	A2	VDDQ_M0
Xm0DATA[3]	GPL3[0] / EBI_D[3]	-	I	A2	VDDQ_M0
Xm0DATA[4]	GPL3[1] / EBI_D[4]	-	I	A2	VDDQ_M0
Xm0DATA[5]	GPL3[2] / EBI_D[5]	-	I	A2	VDDQ_M0
Xm0DATA[6]	GPL3[3] / EBI_D[6]	-	I	A2	VDDQ_M0
Xm0DATA[7]	GPL3[4] / EBI_D[7]	-	I	A2	VDDQ_M0
Xm0DATA[8]	GPL3[5] / EBI_D[8]	-	I	A2	VDDQ_M0
Xm0DATA[9]	GPL3[6] / EBI_D[9]	-	I	A2	VDDQ_M0
Xm0DATA[10]	GPL3[7] / EBI_D[10]	-	I	A2	VDDQ_M0
Xm0DATA[11]	GPL4[0] / EBI_D[11]	-	I	A2	VDDQ_M0
Xm0DATA[12]	GPL4[1] / EBI_D[12]	-	I	A2	VDDQ_M0
Xm0DATA[13]	GPL4[2] / EBI_D[13]	-	I	A2	VDDQ_M0
Xm0DATA[14]	GPL4[3] / EBI_D[14]	-	I	A2	VDDQ_M0
Xm0DATA[15]	GPL4[4] / EBI_D[15]	-	I	A2	VDDQ_M0
XefFSOURCE_0	efrom_fsource_0	-	I	-	VDDQ_M0
XefFVGATE_0	efrom_fvgate_0	-	I	-	VDDQ_M0

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XDDR2_SEL	DDR2SEL	PD	I	E1	VDDQ_EXT
XjTRSTn	TRSTn	PU	I	E1	VDDQ_EXT
XjTMS	TMS	PU	I	E1	VDDQ_EXT
XjTCK	TCK	PU	I	E1	VDDQ_EXT
XjTDO	TDO	-	OL	E1	VDDQ_EXT
XjTDI	TDI	PU	I	E1	VDDQ_EXT
XjDBGSEL	DBGSEL	PD	I	E1	VDDQ_EXT
XuhDP	UHOST_DP	-	AI	-	VDD_UH
XuhDN	UHOST_DN	-	AI	-	VDD_UH
XPWRRGTON	PWRRGTON	-	OL	B	VDDQ_SYS0
XnRESET	nRESET	-	I	B	VDDQ_SYS0
XnWRESET	nWRESET	-	I	B	VDDQ_SYS0
XnBATF	nBATF	-	I	B	VDDQ_SYS0
XOM[0]	OM[0]	-	I	B	VDDQ_SYS0
XOM[1]	OM[1]	-	I	B	VDDQ_SYS0
XOM[2]	OM[2]	-	I	B	VDDQ_SYS0
XOM[3]	OM[3]	-	I	B	VDDQ_SYS0
XOM[4]	OM[4]	-	I	B	VDDQ_SYS0
XXTI	XTI	-	I	-	VDDQ_SYS0
XXTO	XTO	-	OL	-	VDDQ_SYS0
XrtcXTI	RTC_XTI	-	I	-	VDD_RTC
XrtcXTO	RTC_XXTO	-	OL	-	VDD_RTC
XusbXTI	USB_XTI	-	AI	-	VDDQ_SYS2
XusbXTO	USB_XTO	-	AO	-	VDDQ_SYS2
X27mXTI	27M_XTI	-	I	-	VDDQ_SYS2
X27mXTO	27M_XTO	-	OL	-	VDDQ_SYS2
XCLKOUT	CLKOUT	-	OL	E1	VDDQ_SYS2
XnRSTOUT	nRSTOUT	-	OH	-	VDDQ_SYS2
XadcAIN[9]	ADC_IN[9]	-	I	H	VDD_ADC
XadcAIN[8]	ADC_IN[8]	-	I	H	VDD_ADC
XadcAIN[7]	ADC_IN[7]	-	I	H	VDD_ADC
XadcAIN[6]	ADC_IN[6]	-	I	H	VDD_ADC
XadcAIN[5]	ADC_IN[5]	-	I	-	VDD_ADC
XadcAIN[4]	ADC_IN[4]	-	I	-	VDD_ADC
XadcAIN[3]	ADC_IN[3]	-	I	-	VDD_ADC
XadcAIN[2]	ADC_IN[2]	-	I	-	VDD_ADC

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XadcAIN[1]	ADC_IN[1]	-	I	-	VDD_ADC
XadcAIN[0]	ADC_IN[0]	-	I	-	VDD_ADC
XadcVref	ADC_VREF	-	I	-	VDD_ADC
XusbDRVVBUS	USB_DRVVBUS	PD	I	-	VDDQ_USB
XusbDM	USB_DM	-	AI	-	VDDQ_USB
XusbREXT	USB_REXT	-	AI	-	VDDQ_USB
XusbDP	USB_DP	-	AI	-	VDDQ_USB
XusbVBUS	USB_VBUS	-	AI	-	VDDQ_USB
XusbID	USB_ID	-	AI	-	VDDQ_USB
XmipiDP[5]	MIPI_DP[5]	-	I	-	VDD_MIPI
XmipiDN[5]	MIPI_DN[5]	-	I	-	VDD_MIPI
XmipiDP[4]	MIPI_DP[4]	-	I	-	VDD_MIPI
XmipiDN[4]	MIPI_DN[4]	-	I	-	VDD_MIPI
XmipiRXCP	MIPI_RXCP	-	OH	-	VDD_MIPI
XmipiRXCN	MIPI_RXCN	-	OH	-	VDD_MIPI
XmipiDP[3]	MIPI_DP[3]	-	I	-	VDD_MIPI
XmipiDN[3]	MIPI_DN[3]	-	I	-	VDD_MIPI
XmipiDP[2]	MIPI_DP[2]	-	OH	-	VDD_MIPI
XmipiDPN[2]	MIPI_DN[2]	-	OH	-	VDD_MIPI
XmipiDP[1]	MIPI_DP[1]	-	OH	-	VDD_MIPI
XmipiDN[1]	MIPI_DN[1]	-	OH	-	VDD_MIPI
XmipiReg_cap	MIPI_Reg_cap	-	I	-	VDD_MIPI
XmipiTXCP	MIPI_TXCP	-	OH	-	VDD_MIPI
XmipiTXCN	MIPI_TXCN	-	OH	-	VDD_MIPI
XmipiDP[0]	MIPI_DP[0]	-	OH	-	VDD_MIPI
XmipiDN[0]	MIPI_DN[0]	-	OH	-	VDD_MIPI
XhdmiREXT	HDMI_REXT	-	I	-	VDD_HDMI
XhdmiTXCP	HDMI_TXCP	-	OH	-	VDD_HDMI
XhdmiTx0N	HDMI_Tx0N	-	OH	-	VDD_HDMI
XhdmiTX0P	HDMI_TX0P	-	OH	-	VDD_HDMI
XhdmiTX1N	HDMI_TX1N	-	OH	-	VDD_HDMI
XhdmiTX1P	HDMI_TX1P	-	OH	-	VDD_HDMI
XhdmiTX2N	HDMI_TX2N	-	OH	-	VDD_HDMI
XhdmiTX2P	HDMI_TX2P	-	OH	-	VDD_HDMI
XyyTCLK	CG_CLK	-	I	-	VDDQ_SYS5
XdacOUT[1]	DAC_OUT[1]	-	OH	-	VDDQ_MSM

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
XdacOUT[2]	DAC_OUT[2]	-	OH	-	VDDQ_MSM
XdacIREF	DAC_IREF	-	I	-	VDDQ_MSM
XdacVREF	DAC_VREF	-	I	-	VDDQ_MSM
XdacCOMP	DAC_COMP	-	OH	-	VDDQ_MSM
XdacOUT[0]	DAC_OUT[0]	-	OH	-	VDDQ_MSM
XI2S0LRCK	I2S0_LRCK	-	OL	E1	VDDQ_AUD
XI2S0CDCLK	I2S0_CDCLK	-	OL	E1	VDDQ_AUD
XI2S0SCLK	I2S0_SCLK	-	OL	E1	VDDQ_AUD
XI2S0SDI	I2S 0_SDI	PD	I	E1	VDDQ_AUD
XI2S0SDO[0]	I2S0_SDO[0]	-	OL	E1	VDDQ_AUD
XI2S0SDO[1]	I2S0_SDO[1]	-	OL	E1	VDDQ_AUD
XI2S0SDO[2]	I2S0_SDO[2]	-	OL	E1	VDDQ_AUD
Xm1CKE[0]	DDR_CKE[0]	-	OH	A2	VDDQ_DDR
Xm1CKE[1]	DDR_CKE[1]	-	OH	A2	VDDQ_DDR
Xm1SCLK	DDR_SCLK	-	OL	A2	VDDQ_DDR
Xm1nSCLK	DDR_SCLKn	-	OH	A2	VDDQ_DDR
Xm1CSn[0]	DDR_CSn[0]	-	OH	A2	VDDQ_DDR
Xm1CSn[1]	DDR_CSn[1]	-	OH	A2	VDDQ_DDR
Xm1ADDR[0]	DDR_A[0]	-	OL	A2	VDDQ_DDR
Xm1ADDR[1]	DDR_A[1]	-	OL	A2	VDDQ_DDR
Xm1ADDR[2]	DDR_A[2]	-	OL	A2	VDDQ_DDR
Xm1ADDR[3]	DDR_A[3]	-	OL	A2	VDDQ_DDR
Xm1ADDR[4]	DDR_A[4]	-	OL	A2	VDDQ_DDR
Xm1ADDR[5]	DDR_A[5]	-	OL	A2	VDDQ_DDR
Xm1ADDR[6]	DDR_A[6]	-	OL	A2	VDDQ_DDR
Xm1ADDR[7]	DDR_A[7]	-	OL	A2	VDDQ_DDR
Xm1ADDR[8]	DDR_A[8]	-	OL	A2	VDDQ_DDR
Xm1ADDR[9]	DDR_A[9]	-	OL	A2	VDDQ_DDR
Xm1ADDR[10]	DDR_A[10]	-	OL	A2	VDDQ_DDR
Xm1ADDR[11]	DDR_A[11]	-	OL	A2	VDDQ_DDR
Xm1ADDR[12]	DDR_A[12]	-	OL	A2	VDDQ_DDR
Xm1ADDR[13]	DDR_A[13]	-	OL	A2	VDDQ_DDR
Xm1ADDR[14]	DDR_A[14]	-	OL	A2	VDDQ_DDR
Xm1ADDR[15]	DDR_A[15]	-	OL	A2	VDDQ_DDR
Xm1RASn	DDR_RASn	-	OH	A2	VDDQ_DDR
Xm1CASn	DDR_CASn	-	OH	A2	VDDQ_DDR

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
Xm1WEn	DDR_WEn	-	OH	A2	VDDQ_DDR
Xm1DATA[0]	DDR_D[0]	-	I	A2	VDDQ_DDR
Xm1DATA[1]	DDR_D[1]	-	I	A2	VDDQ_DDR
Xm1DATA[2]	DDR_D[2]	-	I	A2	VDDQ_DDR
Xm1DATA[3]	DDR_D[3]	-	I	A2	VDDQ_DDR
Xm1DATA[4]	DDR_D[4]	-	I	A2	VDDQ_DDR
Xm1DATA[5]	DDR_D[5]	-	I	A2	VDDQ_DDR
Xm1DATA[6]	DDR_D[6]	-	I	A2	VDDQ_DDR
Xm1DATA[7]	DDR_D[7]	-	I	A2	VDDQ_DDR
Xm1DATA[8]	DDR_D[8]	-	I	A2	VDDQ_DDR
Xm1DATA[9]	DDR_D[9]	-	I	A2	VDDQ_DDR
Xm1DATA[10]	DDR_D[10]	-	I	A2	VDDQ_DDR
Xm1DATA[11]	DDR_D[11]	-	I	A2	VDDQ_DDR
Xm1DATA[12]	DDR_D[12]	-	I	A2	VDDQ_DDR
Xm1DATA[13]	DDR_D[13]	-	I	A2	VDDQ_DDR
Xm1DATA[14]	DDR_D[14]	-	I	A2	VDDQ_DDR
Xm1DATA[15]	DDR_D[15]	-	I	A2	VDDQ_DDR
Xm1DATA[16]	DDR_D[16]	-	I	A2	VDDQ_DDR
Xm1DATA[17]	DDR_D[17]	-	I	A2	VDDQ_DDR
Xm1DATA[18]	DDR_D[18]	-	I	A2	VDDQ_DDR
Xm1DATA[19]	DDR_D[19]	-	I	A2	VDDQ_DDR
Xm1DATA[20]	DDR_D[20]	-	I	A2	VDDQ_DDR
Xm1DATA[21]	DDR_D[21]	-	I	A2	VDDQ_DDR
Xm1DATA[22]	DDR_D[22]	-	I	A2	VDDQ_DDR
Xm1DATA[23]	DDR_D[23]	-	I	A2	VDDQ_DDR
Xm1DATA[24]	DDR_D[24]	-	I	A2	VDDQ_DDR
Xm1DATA[25]	DDR_D[25]	-	I	A2	VDDQ_DDR
Xm1DATA[26]	DDR_D[26]	-	I	A2	VDDQ_DDR
Xm1DATA[27]	DDR_D[27]	-	I	A2	VDDQ_DDR
Xm1DATA[28]	DDR_D[28]	-	I	A2	VDDQ_DDR
Xm1DATA[29]	DDR_D[29]	-	I	A2	VDDQ_DDR
Xm1DATA[30]	DDR_D[30]	-	I	A2	VDDQ_DDR
Xm1DATA[31]	DDR_D[31]	-	I	A2	VDDQ_DDR
Xm1DATAQM[0]	DDR_DQM[0]	-	OL	A2	VDDQ_DDR
Xm1DATAQM[1]	DDR_DQM[1]	-	OL	A2	VDDQ_DDR
Xm1DATAQM[2]	DDR_DQM[2]	-	OL	A2	VDDQ_DDR

Pad Name	Function Singnals	Pull	I/O	PDN	VDD
Xm1DATAQM[3]	DDR_DQM[3]	-	OL	A2	VDDQ_DDR
Xm1DATAQS[0]	DDR_DQS[0]	-	I	A2	VDDQ_DDR
Xm1DATAQS[1]	DDR_DQS[1]	-	I	A2	VDDQ_DDR
Xm1DATAQS[2]	DDR_DQS[2]	-	I	A2	VDDQ_DDR
Xm1DATAQS[3]	DDR_DQS[3]	-	I	A2	VDDQ_DDR
Xm1DATAQSn[0]	DDR_DQSn[0]	-	I	A2	VDDQ_DDR
Xm1DATAQSn[1]	DDR_DQSn[1]	-	I	A2	VDDQ_DDR
Xm1DATAQSn[2]	DDR_DQSn[2]	-	I	A2	VDDQ_DDR
Xm1DATAQSn[3]	DDR_DQSn[3]	-	I	A2	VDDQ_DDR

5 REGISTER DESCRIPTION

GPxCON, GPxDAT, GPxPULL, GPxDRV work in normal mode, GPxPDNCON, GPxPDNPULL work in power down mode(DEEP-IDLE and Top-off, DEEP-STOP and Top-off, SLEEP mode).

If, S5PC100 enter the power down mode, all configurations and Pull-down controls are selected by power down registers

Register	Address	R/W	Description	Reset Value
GPA0CON	0xE030_0000	R/W	Port Group GPA0 Configuration Register	0x00000000
GPA0DAT	0xE030_0004	R/W	Port Group GPA0 Data Register	-
GPA0PULL	0xE030_0008	R/W	Port Group GPA0 Pull-up/down Register	0x5555
GPA0DRV	0xE030_000C	R/W	Port Group GPA0 Drive strength control Register	0x0000
GPA0PDNCON	0xE030_0010	R/W	Port Group GPA0 Power down mode Configuration Register	0x00
GPA0PDNPULL	0xE030_0014	R/W	Port Group GPA0 Power down mode Pull-up/down Register	0x00
GPA1CON	0xE030_0020	R/W	Port Group GPA1 Configuration Register	0x00000000
GPA1DAT	0xE030_0024	R/W	Port Group GPA1 Data Register	-
GPA1PULL	0xE030_0028	R/W	Port Group GPA1 Pull-up/down Register	0x0155
GPA1DRV	0xE030_002C	R/W	Port Group GPA1 Drive strength control Register	0x0000
GPA1PDNCON	0xE030_0030	R/W	Port Group GPA1 Power down mode Configuration Register	0x00
GPA1PDNPULL	0xE030_0034	R/W	Port Group GPA1 Power down mode Pull-up/down Register	0x00
GPBCON	0xE030_0040	R/W	Port Group GPB Configuration Register	0x00000000
GPBDAT	0xE030_0044	R/W	Port Group GPB Data Register	-
GPBPULL	0xE030_0048	R/W	Port Group GPB Pull-up/down Register	0x5555
GPBDRV	0xE030_004C	R/W	Port Group GPB Drive strength control Register	0x0000
GPBPDNCON	0xE030_0050	R/W	Port Group GPB power down mode configuration register	0x00
GPBPDNPULL	0xE030_0054	R/W	Port Group GPB power down mode pull-up/down register	0x00
GPCCON	0xE030_0060	R/W	Port Group GPC Configuration Register	0x00000000
GPCDAT	0xE030_0064	R/W	Port Group GPC Data Register	-
GPCPULL	0xE030_0068	R/W	Port Group GPC Pull-up/down Register	0x0155
GPCDRV	0xE030_006C	R/W	Port Group GPC Drive strength control Register	0x0000
GPCPDNCON	0xE030_0070	R/W	Port group GPC power down mode configuration register	0x00
GPCPDNPULL	0xE030_0074	R/W	Port group GPC power down mode pull-up/down register	0x00
GPDCON	0xE030_0080	R/W	Port Group GPD Configuration Register	0x00000000

Register	Address	R/W	Description	Reset Value
GPDDAT	0xE030_0084	R/W	Port Group GPD Data Register	-
GPDPULL	0xE030_0088	R/W	Port Group GPD Pull-up/down Register	0x1555
GPDDRV	0xE030_008C	R/W	Port Group GPD Drive strength control Register	0x0000
GPDPDNCON	0xE030_0090	R/W	Port Group GPD Power down mode Configuration Register	0x00
GPDPDNPULL	0xE030_0094	R/W	Port Group GPD Power down mode Pull-up/down Register	0x00
GPE0CON	0xE030_00A0	R/W	Port Group GPE0 Configuration Register	0x00000000
GPE0DAT	0xE030_00A4	R/W	Port Group GPE0 Data Register	-
GPE0PULL	0xE030_00A8	R/W	Port Group GPE0 Pull-up/down Register	0x5555
GPE0DRV	0xE030_00AC	R/W	Port Group GPE0 Drive strength control Register	0x0000
GPE0PDNCON	0xE03000B0	R/W	Port Group GPE0 Power down mode Configuration Register	0x00
GPE0PDNPULL	0xE030_00B4	R/W	Port Group GPE0 Power down mode Pull-up/down Register	0x00
GPE1CON	0xE030_00C0	R/W	Port Group GPE1 Configuration Register	0x00000000
GPE1DAT	0xE030_00C4	R/W	Port Group GPE1 Data Register	-
GPE1PULL	0xE030_00C8	R/W	Port Group GPE1 Pull-up/down Register	0x0555
GPE1DRV	0xE030_00CC	R/W	Port Group GPE1 Drive strength control Register	0x0000
GPE1PDNCON	0xE030_00D0	R/W	Port Group GPE1 Power down mode Configuration Register	0x00
GPE1PDNPULL	0xE030_00D4	R/W	Port Group GPE1 Power down mode Pull-up/down Register	0x00
GPF0CON	0xE030_00E0	R/W	Port Group GPF0 Configuration Register	0x00000000
GPF0DAT	0xE030_00E4	R/W	Port Group GPF0 Data Register	Don't care
GPF0PULL	0xE030_00E8	R/W	Port Group GPF0 Pull-up/down Register	0x5555
GPF0DRV	0xE030_00EC	R/W	Port Group GPF0 Drive strength control Register	0x0000
GPF0PDNCON	0xE030_00F0	R/W	Port Group GPF0 Power down mode Configuration Register	0x00
GPF0PDNPULL	0xE030_00F4	R/W	Port Group GPF0 Power down mode Pull-up/down Register	0x00
GPF1CON	0xE030_0100	R/W	Port Group GPF1 Configuration Register	0x00000000
GPF1DAT	0xE030_0104	R/W	Port Group GPF1 Data Register	-
GPF1PULL	0xE030_0108	R/W	Port Group GPF1 Pull-up/down Register	0x5555
GPF1DRV	0xE030_010C	R/W	Port Group GPF1 Drive strength control Register	0x0000
GPF1PDNCON	0xE030_0110	R/W	Port Group GPF1 Power down mode Configuration Register	0x00
GPF1PDNPULL	0xE030_0114	R/W	Port Group GPF1 Power down mode Pull-up/down Register	0x00

Register	Address	R/W	Description	Reset Value
GPF2CON	0xE030_0120	R/W	Port Group GPF2 Configuration Register	0x00000000
GPF2DAT	0xE030_0124	R/W	Port Group GPF2 Data Register	-
GPF2PULL	0xE030_0128	R/W	Port Group GPF2 Pull-up/down Register	0x5555
GPF2DRV	0xE030_012C	R/W	Port Group GPF2 Drive strength control Register	0x0000
GPF2PDNCON	0xE030_0130	R/W	Port Group GPF2 Power down mode Configuration Register	0x00
GPF2PDNPULL	0xE030_0134	R/W	Port Group GPF2 Power down mode Pull-up/down Register	0x00
GPF3CON	0xE030_0140	R/W	Port Group GPF3 Configuration Register	0x00000000
GPF3DAT	0xE030_0144	R/W	Port Group GPF3 Data Register	-
GPF3PUD	0xE030_0148	R/W	Port Group GPF3 Pull-up/down Register	0x0055
GPF3DRV	0xE030_014C	R/W	Port Group GPF3 Drive strength control Register	0x0000
GPF3PDNCON	0xE030_0150	R/W	Port Group GPF3 Power down mode Configuration Register	0x00
GPF3PDNPULL	0xE030_0154	R/W	Port Group GPF3 Power down mode Pull-up/down Register	0x00
GPG0CON	0xE030_0160	R/W	Port Group GPG0 Configuration Register	0x00000000
GPG0DAT	0xE030_0164	R/W	Port Group GPG0 Data Register	-
GPG0PULL	0xE030_0168	R/W	Port Group GPG0 Pull-up/down Register	0x5555
GPG0DRV	0xE030_016C	R/W	Port Group GPG0 Drive strength control Register	0x0000
GPG0PDNCON	0xE030_0170	R/W	Port Group GPG0 Power down mode Configuration Register	0x00
GPG0PDNPULL	0xE030_0174	R/W	Port Group GPG0 Power down mode Pull-up/down Register	0x00
GPG1CON	0xE030_0180	R/W	Port Group GPG1 Configuration Register	0x00000000
GPG1DAT	0xE030_0184	R/W	Port Group GPG1 Data Register	-
GPG1PULL	0xE030_0188	R/W	Port Group GPG1 Pull-up/down Register	0x0015
GPG1DRV	0xE030_018C	R/W	Port Group GPG1 Drive strength control Register	0x0000
GPG1PDNCON	0xE030_0190	R/W	Port Group GPG1 Power down mode Configuration Register	0x00
GPG1PDNPULL	0xE030_0194	R/W	Port Group GPG1 Power down mode Pull-up/down Register	0x00
GPG2CON	0xE030_01A0	R/W	Port Group GPG2 Configuration Register	0x00000000
GPG2DAT	0xE030_01A4	R/W	Port Group GPG2 Data Register	-
GPG2PULL	0xE030_01A8	R/W	Port Group GPG2 Pull-up/down Register	0x1555
GPG2DRV	0xE030_01AC	R/W	Port Group GPG2 Drive strength control Register	0x0000
GPG2PDNCON	0xE030_01B0	R/W	Port Group GPG2 Power down mode Configuration Register	0x00
GPG2PDNPULL	0xE030_01B4	R/W	Port Group GPG2 Power down mode Pull-	0x00

Register	Address	R/W	Description	Reset Value
			up/down Register	
GPG3CON	0xE030_01C0	R/W	Port Group GPG3 Configuration Register	0x00000000
GPG3DAT	0xE030_01C4	R/W	Port Group GPG3 Data Register	-
GPG3PUD	0xE030_01C8	R/W	Port Group GPG3 Pull-up/down Register	0x1555
GPG3DRV	0xE030_01CC	R/W	Port Group GPG3 Drive strength control Register	0x0000
GPG3PDNCON	0xE030_01D0	R/W	Port Group GPG3 Power down mode Configuration Register	0x00
GPG3PDNPULL	0xE030_01D4	R/W	Port Group GPG3 Power down mode Pull-up/down Register	0x00
GPH0CON	0xE030_0C00	R/W	Port Group GPH0 Configuration Register	0x00000000
GPH0DAT	0xE030_0C04	R/W	Port Group GPH0 Data Register	-
GPH0PULL	0xE030_0C08	R/W	Port Group GPH0 Pull-up/down Register	0x5555
GPH0DRV	0xE030_0C0C	R/W	Port Group GPH0 Drive strength control Register	0x0000
GPH1CON	0xE030_0C20	R/W	Port Group GPH1 Configuration Register	0x00000000
GPH1DAT	0xE030_0C24	R/W	Port Group GPH1 Data Register	-
GPH1PULL	0xE030_0C28	R/W	Port Group GPH1 Pull-up/down Register	0x5555
GPH1DRV	0xE030_0C2C	R/W	Port Group GPH1 Drive strength control Register	0x0000
GPH2CON	0xE030_0C40	R/W	Port Group GPH2 Configuration Register	0x00000000
GPH2DAT	0xE030_0C44	R/W	Port Group GPH2 Data Register	-
GPH2PULL	0xE030_0C48	R/W	Port Group GPH2 Pull-up/down Register	0x5555
GPH2DRV	0xE030_0C4C	R/W	Port Group GPH2 Drive strength control Register	0x0000
GPH3CON	0xE030_0C60	R/W	Port Group GPH3 Configuration Register	0x00000000
GPH3DAT	0xE030_0C64	R/W	Port Group GPH3 Data Register	-
GPH3PUD	0xE030_0C68	R/W	Port Group GPH3 Pull-up/down Register	0x5555
GPH3DRV	0xE030_0C6C	R/W	Port Group GPH3 Drive strength control Register	0x0000
GPICON	0xE030_01E0	R/W	Port Group GPI Configuration Register	0x22222200
GPIDAT	0xE030_01E4	R/W	Port Group GPI Data Register	-
GPIPUD	0xE030_01E8	R/W	Port Group GPI Pull-up/down Register	0x5505
GPIDRV	0xE030_01EC	R/W	Port Group GPI Drive strength control Register	0x0000
GPIPDNCON	0xE030_01F0	R/W	Port Group GPI Power down mode Configuration Register	0x00
GPIPUDPDN	0xE030_01F4	R/W	Port Group GPI Power down mode Pull-up/down Register	0x00
GPJ0CON	0xE030_0200	R/W	Port Group GPJ0 Configuration Register	0x00000000
GPJ0DAT	0xE030_0204	R/W	Port Group GPJ0 Data Register	-
GPJ0PULL	0xE030_0208	R/W	Port Group GPJ0 Pull-up/down Register	0x5555
GPJ0DRV	0xE030_020C	R/W	Port Group GPJ0 Drive strength control Register	0x0000

Register	Address	R/W	Description	Reset Value
GPJ0PDNCON	0xE030_0210	R/W	Port Group GPJ0 Power down mode Configuration Register	0x00
GPJ0PDNPULL	0xE030_0214	R/W	Port Group GPJ0 Power down mode Pull-up/down Register	0x00
GPJ1CON	0xE030_0220	R/W	Port Group GPJ1 Configuration Register	0x00000000
GPJ1DAT	0xE030_0224	R/W	Port Group GPJ1 Data Register	-
GPJ1PULL	0xE030_0228	R/W	Port Group GPJ1 Pull-up/down Register	0x0155
GPJ1DRV	0xE030_022C	R/W	Port Group GPJ1 Drive strength control Register	0x0000
GPJ1PDNCON	0xE030_0230	R/W	Port Group GPJ1 Power down mode Configuration Register	0x00
GPJ1PDNPULL	0xE030_0234	R/W	Port Group GPJ1 Power down mode Pull-up/down Register	0x00
GPJ2CON	0xE030_0240	R/W	Port Group GPJ2 Configuration Register	0x00000000
GPJ2DAT	0xE030_0244	R/W	Port Group GPJ2 Data Register	-
GPJ2PULL	0xE030_0248	R/W	Port Group GPJ2 Pull-up/down Register	0x5555
GPJ2DRV	0xE030_024C	R/W	Port Group GPJ2 Drive strength control Register	0x0000
GPJ2PDNCON	0xE030_0250	R/W	Port Group GPJ2 power down mode configuration register	0x00
GPJ2PDNPULL	0xE030_0254	R/W	Port Group GPJ2 power down mode pull-up/down register	0x00
GPJ3CON	0xE030_0260	R/W	Port Group GPJ3 Configuration Register	0x00000000
GPJ3DAT	0xE030_0264	R/W	Port Group GPJ3 Data Register	-
GPJ3PUD	0xE030_0268	R/W	Port Group GPJ3 Pull-up/down Register	0x5555
GPJ3DRV	0xE030_026C	R/W	Port Group GPJ3 Drive strength control Register	0x0000
GPJ3PDNCON	0xE030_0270	R/W	Port Group GPJ3 Power down mode Configuration Register	0x00
GPJ3PDNPULL	0xE030_0274	R/W	Port Group GPJ3 Power down mode Pull-up/down Register	0x00
GPJ4CON	0xE030_0280	R/W	Port Group GPJ4 Configuration Register	0x00000000
GPJ4DAT	0xE030_0284	R/W	Port Group GPJ4 Data Register	-
GPJ4PULL	0xE030_0288	R/W	Port Group GPJ4 Pull-up/down Register	0x0055
GPJ4DRV	0xE030_028C	R/W	Port Group GPJ4 Drive strength control Register	0x0000
GPJ4PDNCON	0xE030_0290	R/W	Port Group GPJ4 Power down mode Configuration Register	0x00
GPJ4PDNPULL	0xE030_0294	R/W	Port Group GPJ4 Power down mode Pull-up/down Register	0x00
GPK0CON	0xE030_02A0	R/W	Port Group GPK0 Configuration Register	0x22443322
GPK0DAT	0xE030_02A4	R/W	Port Group GPK0 Data Register	-
GPK0PULL	0xE030_02A8	R/W	Port Group GPK0 Pull-up/down Register	0x0000

Register	Address	R/W	Description	Reset Value
GPK0DRV	0xE030_02AC	R/W	Port Group GPK0 Drive strength control Register	0xAAAA
GPK0PDNCON	0xE030_02B0	R/W	Port Group GPK0 Power down mode Configuration Register	0x00
GPK0PDNPULL	0xE030_02B4	R/W	Port Group GPK0 Power down mode Pull-up/down Register	0x00
GPK1CON	0xE030_02C0	R/W	Port Group GPK1 Configuration Register	0x222222
GPK1DAT	0xE030_02C4	R/W	Port Group GPK1 Data Register	-
GPK1PULL	0xE030_02C8	R/W	Port Group GPK1 Pull-up/down Register	0x0020
GPK1DRV	0xE030_02CC	R/W	Port Group GPK1 Drive strength control Register	0x0AAA
GPK1PDNCON	0xE030_02D0	R/W	Port Group GPK1 Power down mode Configuration Register	0x00
GPK1PDNPULL	0xE030_02D4	R/W	Port Group GPK1 Power down mode Pull-up/down Register	0x00
GPK2CON	0xE030_02E0	R/W	Port Group GPK2 Configuration Register	0x22223333
GPK2DAT	0xE030_02E4	R/W	Port Group GPK2 Data Register	-
GPK2PULL	0xE030_02E8	R/W	Port Group GPK2 Pull-up/down Register	0xA500
GPK2DRV	0xE030_02EC	R/W	Port Group GPK2 Drive strength control Register	0xAAAA
GPK2PDNCON	0xE030_02F0	R/W	Port Group GPK2 Power down mode Configuration Register	0x00
GPK2PDNPULL	0xE030_02F4	R/W	Port Group GPK2 Power down mode Pull-up/down Register	0x00
GPK3CON	0xE030_0300	R/W	Port Group GPK3 Configuration Register	0x22222222
GPK3DAT	0xE030_0304	R/W	Port Group GPK3 Data Register	-
GPK3PUD	0xE030_0308	R/W	Port Group GPK3 Pull-up/down Register	0x0885
GPK3DRV	0xE030_030C	R/W	Port Group GPK3 Drive strength control Register	0xAAAA
GPK3PDNCON	0xE030_0310	R/W	Port Group GPK3 Power down mode Configuration Register	0x00
GPK3PDNPULL	0xE030_0314	R/W	Port Group GPK3 Power down mode Pull-up/down Register	0x00
GPL0CON	0xE030_0320	R/W	Port Group GPL0 Configuration Register	0x22222222
GPL0DAT	0xE030_0324	R/W	Port Group GPL0 Data Register	-
GPL0PULL	0xE030_0328	R/W	Port Group GPL0 Pull-up/down Register	0x0000
GPL0DRV	0xE030_032C	R/W	Port Group GPL0 Drive strength control Register	0xAAAA
GPL0PDNCON	0xE030_0330	R/W	Port Group GPL0 Power down mode Configuration Register	0x00
GPL0PDNPULL	0xE030_0334	R/W	Port Group GPL0 Power down mode Pull-up/down Register	0x00
GPL1CON	0xE030_0340	R/W	Port Group GPL1 Configuration Register	0x22222222
GPL1DAT	0xE030_0344	R/W	Port Group GPL1 Data Register	-

Register	Address	R/W	Description	Reset Value
GPL1PULL	0xE030_0348	R/W	Port Group GPL1 Pull-up/down Register	0x0000
GPL1DRV	0xE030_034C	R/W	Port Group GPL1 Drive strength control Register	0xAAAA
GPL1PDNCON	0xE030_0350	R/W	Port Group GPL1 Power down mode Configuration Register	0x00
GPL1PDNPULL	0xE030_0354	R/W	Port Group GPL1 Power down mode Pull-up/down Register	0x00
GPL2CON	0xE030_0360	R/W	Port Group GPL2 Configuration Register	0x22222222
GPL2DAT	0xE030_0364	R/W	Port Group GPL2 Data Register	-
GPL2PULL	0xE030_0368	R/W	Port Group GPL2 Pull-up/down Register	0x0000
GPL2DRV	0xE030_036C	R/W	Port Group GPL2 Drive strength control Register	0xAAAA
GPL2PDNCON	0xE030_0370	R/W	Port Group GPL2 Power down mode Configuration Register	0x00
GPL2PDNPULL	0xE030_0374	R/W	Port Group GPL2 Power down mode Pull-up/down Register	0x00
GPL3CON	0xE030_0380	R/W	Port Group GPL3 Configuration Register	0x22222222
GPL3DAT	0xE030_0384	R/W	Port Group GPL3 Data Register	-
GPL3PUD	0xE030_0388	R/W	Port Group GPL3 Pull-up/down Register	0x0000
GPL3DRV	0xE030_038C	R/W	Port Group GPL3 Drive strength control Register	0xAAAA
GPL3PDNCON	0xE030_0390	R/W	Port Group GPL3 Power down mode Configuration Register	0x00
GPL3PDNPULL	0xE030_0394	R/W	Port Group GPL3 Power down mode Pull-up/down Register	0x00
GPL4CON	0xE030_03A0	R/W	Port Group GPL4 Configuration Register	0x22222
GPL4DAT	0xE030_03A4	R/W	Port Group GPL4 Data Register	-
GPL4PULL	0xE030_03A8	R/W	Port Group GPL4 Pull-up/down Register	0x0000
GPL4DRV	0xE030_03AC	R/W	Port Group GPL4 Drive strength control Register	0x02AA
GPL4PDNCON	0xE030_03B0	R/W	Port Group GPL4 Power down mode Configuration Register	0x00
GPL4PDNPULL	0xE030_03B4	R/W	Port Group GPL4 Power down mode Pull-up/down Register	0x00
MP1_0DRV	0xE030_03CC	R/W	Port Group MP1_0 Drive strength control Register	0xAAAA
MP1_1DRV	0xE030_03EC	R/W	Port Group MP1_1 Drive strength control Register	0xAAAA
MP1_2DRV	0xE030_040C	R/W	Port Group MP1_2 Drive strength control Register	0xAAAA
MP1_3DRV	0xE030_042C	R/W	Port Group MP1_3 Drive strength control Register	0xAAAA
MP1_4DRV	0xE030_044C	R/W	Port Group MP1_4 Drive strength control Register	0xAAAA
MP1_5DRV	0xE030_046C	R/W	Port Group MP1_5 Drive strength control Register	0xAAAA
MP1_6DRV	0xE030_048C	R/W	Port Group MP1_6 Drive strength control Register	0xAAAA
MP1_7DRV	0xE030_04AC	R/W	Port Group MP1_7 Drive strength control Register	0xAAAA
MP1_8DRV	0xE030_04CC	R/W	Port Group MP1_8 Drive strength control Register	0x2AAA

Register	Address	R/W	Description	Reset Value
ETC0PULL	0xE030_04E8	R/W	Port Group ETC0 Pull-up/down Register	0x04AA
ETC0DRV	0xE030_04EC	R/W	Port Group ETC0 Drive strength control Register	0x0000
ETC1PULL	0xE030_0508	R/W	Port Group ETC1 Pull-up/down Register	0x8555
ETC1DRV	0xE030_050C	R/W	Port Group ETC1 Drive strength control Register	0x0000
ETC2PULL	0xE030_0528	R/W	Port Group ETC2 Pull-up/down Register	0x0082
ETC2DRV	0xE030_052C	R/W	Port Group ETC2 Drive strength control Register	0x0000
ETC3PUD	0xE030_0548	R/W	Port Group ETC3 Pull-up/down Register	0x0040
ETC3DRV	0xE030_054C	R/W	Port Group ETC3 Drive strength control Register	0x0000
ETC4DRV	0xE030_056C	R/W	Port Group ETC4 Drive strength control Register	0x0000

Register	Address	R/W	Description	Reset Value
NWU_INT0_CON	0xE030_0700	R/W	Non wake-up Interrupt 0 Configuration Register	0x0
NWU_INT1_CON	0xE030_0704	R/W	Non wake-up Interrupt 1 Configuration Register	0x0
NWU_INT2_CON	0xE030_0708	R/W	Non wake-up Interrupt 2 Configuration Register	0x0
NWU_INT3_CON	0xE030_070C	R/W	Non wake-up Interrupt 3 Configuration Register	0x0
NWU_INT4_CON	0xE030_0710	R/W	Non wake-up Interrupt 4 Configuration Register	0x0
NWU_INT5_CON	0xE030_0714	R/W	Non wake-up Interrupt 5 Configuration Register	0x0
NWU_INT6_CON	0xE030_0718	R/W	Non wake-up Interrupt 6 Configuration Register	0x0
NWU_INT7_CON	0xE030_071C	R/W	Non wake-up Interrupt 7 Configuration Register	0x0
NWU_INT8_CON	0xE030_0720	R/W	Non wake-up Interrupt 8 Configuration Register	0x0
NWU_INT9_CON	0xE030_0724	R/W	Non wake-up Interrupt 9 Configuration Register	0x0
NWU_INT10_CON	0xE030_0728	R/W	Non wake-up Interrupt 10 Configuration Register	0x0
NWU_INT11_CON	0xE030_072C	R/W	Non wake-up Interrupt 11 Configuration Register	0x0
NWU_INT12_CON	0xE030_0730	R/W	Non wake-up Interrupt 12 Configuration Register	0x0
NWU_INT13_CON	0xE030_0734	R/W	Non wake-up Interrupt 13 Configuration Register	0x0
NWU_INT14_CON	0xE030_0738	R/W	Non wake-up Interrupt 14 Configuration Register	0x0
NWU_INT15_CON	0xE030_073C	R/W	Non wake-up Interrupt 15 Configuration Register	0x0
NWU_INT16_CON	0xE030_0740	R/W	Non wake-up Interrupt 16 Configuration Register	0x0
NWU_INT17_CON	0xE030_0744	R/W	Non wake-up Interrupt 17 Configuration Register	0x0
NWU_INT18_CON	0xE030_0748	R/W	Non wake-up Interrupt 18 Configuration Register	0x0
NWU_INT19_CON	0xE030_074C	R/W	Non wake-up Interrupt 19 Configuration Register	0x0
NWU_INT20_CON	0xE030_0750	R/W	Non wake-up Interrupt 20 Configuration Register	0x0
NWU_INT0_FLTCON0	0xE030_0800	R/W	Non wake-up Interrupt 0 Filter Configuration Register 0	0x0
NWU_INT0_FLTCON1	0xE030_0804	R/W	Non wake-up Interrupt 0 Filter Configuration Register 1	0x0
NWU_INT1_FLTCON0	0xE030_0808	R/W	Non wake-up Interrupt 1 Filter Configuration Register 0	0x0

Register	Address	R/W	Description	Reset Value
NWU_INT1_FLTCON1	0xE030_080C	R/W	Non wake-up Interrupt 1 Filter Configuration Register 1	0x0
NWU_INT2_FLTCON0	0xE030_0810	R/W	Non wake-up Interrupt 2 filter configuration register 0	0x0
NWU_INT3_FLTCON1	0xE030_0814	R/W	Non wake-up Interrupt 2 filter configuration register 1	0x0
NWU_INT3_FLTCON0	0xE030_0818	R/W	Non wake-up Interrupt 3 filter configuration register 0	0x0
NWU_INT3_FLTCON1	0xE030_081C	R/W	Non wake-up Interrupt 3 filter configuration register 1	0x0
NWU_INT4_FLTCON0	0xE030_0820	R/W	Non wake-up Interrupt 4 filter configuration register 0	0x0
NWU_INT4_FLTCON1	0xE030_0824	R/W	Non wake-up Interrupt 4 filter configuration register 1	0x0
NWU_INT5_FLTCON0	0xE030_0828	R/W	Non wake-up Interrupt 5 filter configuration register 0	0x0
NWU_INT5_FLTCON1	0xE030_082C	R/W	Non wake-up Interrupt 5 filter configuration register 1	0x0
NWU_INT6_FLTCON0	0xE030_0830	R/W	Non wake-up Interrupt 6 filter configuration register 0	0x0
NWU_INT6_FLTCON1	0xE030_0834	R/W	Non wake-up Interrupt 6 filter configuration register 1	0x0
NWU_INT7_FLTCON0	0xE030_0838	R/W	Non wake-up Interrupt 7 filter configuration register 0	0x0
NWU_INT7_FLTCON1	0xE030_083C	R/W	Non wake-up Interrupt 7 filter configuration register 1	0x0
NWU_INT8_FLTCON0	0xE030_0840	R/W	Non wake-up Interrupt 8 filter configuration register 0	0x0
NWU_INT8_FLTCON1	0xE030_0844	R/W	Non wake-up Interrupt 8 filter configuration register 1	0x0
NWU_INT9_FLTCON0	0xE030_0848	R/W	Non wake-up Interrupt 9 filter configuration register 0	0x0
NWU_INT9_FLTCON1	0xE030_084C	R/W	Non wake-up Interrupt 9 filter configuration register 1	0x0
NWU_INT10_FLTCON0	0xE030_0850	R/W	Non wake-up Interrupt 10 filter configuration register 0	0x0
NWU_INT11_FLTCON0	0xE030_0858	R/W	Non wake-up Interrupt 11 filter configuration register 0	0x0
NWU_INT11_FLTCON1	0xE030_085C	R/W	Non wake-up Interrupt 11 filter configuration register 1	0x0
NWU_INT12_FLTCON0	0xE030_0860	R/W	Non wake-up Interrupt 12 filter configuration register 0	0x0
NWU_INT13_FLTCON0	0xE030_0868	R/W	Non wake-up Interrupt 13 filter configuration register 0	0x0
NWU_INT13_FLTCON1	0xE030_086C	R/W	Non wake-up Interrupt 13 filter configuration register 1	0x0
NWU_INT14_FLTCON0	0xE030_0870	R/W	Non wake-up Interrupt 14 filter configuration register 0	0x0
NWU_INT14_FLTCON1	0xE030_0874	R/W	Non wake-up Interrupt 14 filter configuration register 1	0x0
NWU_INT15_FLTCON0	0xE030_0878	R/W	Non wake-up Interrupt 15 Filter Configuration Register 0	0x0
NWU_INT15_FLTCON1	0xE030_087C	R/W	Non wake-up Interrupt 15 Filter Configuration Register 1	0x0
NWU_INT16_FLTCON0	0xE030_0880	R/W	Non wake-up Interrupt 16 Filter Configuration Register 0	0x0
NWU_INT16_FLTCON1	0xE030_0884	R/W	Non wake-up Interrupt 16 Filter Configuration Register 1	0x0
NWU_INT17_FLTCON0	0xE030_0888	R/W	Non wake-up Interrupt 17 Filter Configuration Register 0	0x0
NWU_INT17_FLTCON1	0xE030_088C	R/W	Non wake-up Interrupt 17 Filter Configuration Register 1	0x0
NWU_INT18_FLTCON0	0xE030_0890	R/W	Non wake-up Interrupt 18 Filter Configuration Register 0	0x0
NWU_INT18_FLTCON1	0xE030_0894	R/W	Non wake-up Interrupt 18 Filter Configuration Register 1	0x0
NWU_INT19_FLTCON0	0xE030_0898	R/W	Non wake-up Interrupt 19 Filter Configuration Register 0	0x0
NWU_INT19_FLTCON1	0xE030_089C	R/W	Non wake-up Interrupt 19 Filter Configuration Register 1	0x0
NWU_INT20_FLTCON0	0xE030_08A0	R/W	Non wake-up Interrupt 20 Filter Configuration Register 0	0x0

Register	Address	R/W	Description	Reset Value
NWU_INT0_MASK	0xE030_0900	R/W	Non wake-up Interrupt 0 Mask Register	0xff
NWU_INT1_MASK	0xE030_0904	R/W	Non wake-up Interrupt 1 Mask Register	0x1f
NWU_INT2_MASK	0xE030_0908	R/W	Non wake-up Interrupt 2 Mask Register	0xff
NWU_INT3_MASK	0xE030_090C	R/W	Non wake-up Interrupt 3 Mask Register	0x1f
NWU_INT4_MASK	0xE030_0910	R/W	Non wake-up Interrupt 4 Mask Register	0x7f
NWU_INT5_MASK	0xE030_0914	R/W	Non wake-up Interrupt 5 Mask Register	0xff
NWU_INT6_MASK	0xE030_0918	R/W	Non wake-up Interrupt 6 Mask Register	0x3f
NWU_INT7_MASK	0xE030_091C	R/W	Non wake-up Interrupt 7 Mask Register	0xff
NWU_INT8_MASK	0xE030_0920	R/W	Non wake-up Interrupt 8 Mask Register	0xff
NWU_INT9_MASK	0xE030_0924	R/W	Non wake-up Interrupt 9 Mask Register	0xff
NWU_INT10_MASK	0xE030_0928	R/W	Non wake-up Interrupt 10 Mask Register	0x0f
NWU_INT11_MASK	0xE030_092C	R/W	Non wake-up Interrupt 11 Mask Register	0xff
NWU_INT12_MASK	0xE030_0930	R/W	Non wake-up Interrupt 12 Mask Register	0x07
NWU_INT13_MASK	0xE030_0934	R/W	Non wake-up Interrupt 13 Mask Register	0x7f
NWU_INT14_MASK	0xE030_0938	R/W	Non wake-up Interrupt 14 Mask Register	0x7f
NWU_INT15_MASK	0xE030_093C	R/W	Non wake-up Interrupt 15 Mask Register	0xff
NWU_INT16_MASK	0xE030_0940	R/W	Non wake-up Interrupt 16 Mask Register	0xff
NWU_INT17_MASK	0xE030_0944	R/W	Non wake-up Interrupt 17 Mask Register	0x1f
NWU_INT18_MASK	0xE030_0948	R/W	Non wake-up Interrupt 18 Mask Register	0xff
NWU_INT19_MASK	0xE030_094C	R/W	Non wake-up Interrupt 19 Mask Register	0xff
NWU_INT20_MASK	0xE030_0950	R/W	Non wake-up Interrupt 20 Mask Register	0x0f
NWU_INT0_PEND	0xE030_0A00	R/W	Non wake-up Interrupt 0 Pending Register	0x0
NWU_INT1_PEND	0xE030_0A04	R/W	Non wake-up Interrupt 1 Pending Register	0x0
NWU_INT2_PEND	0xE030_0A08	R/W	Non wake-up Interrupt 2 Pending Register	0x0
NWU_INT3_PEND	0xE030_0A0C	R/W	Non wake-up Interrupt 3 Pending Register	0x0
NWU_INT4_PEND	0xE030_0A10	R/W	Non wake-up Interrupt 4 Pending Register	0x0
NWU_INT5_PEND	0xE030_0A14	R/W	Non wake-up Interrupt 5 Pending Register	0x0
NWU_INT6_PEND	0xE030_0A18	R/W	Non wake-up Interrupt 6 Pending Register	0x0
NWU_INT7_PEND	0xE030_0A1C	R/W	Non wake-up Interrupt 7 Pending Register	0x0
NWU_INT8_PEND	0xE030_0A20	R/W	Non wake-up Interrupt 8 Pending Register	0x0
NWU_INT9_PEND	0xE030_0A24	R/W	Non wake-up Interrupt 9 Pending Register	0x0
NWU_INT10_PEND	0xE030_0A28	R/W	Non wake-up Interrupt 10 Pending Register	0x0
NWU_INT11_PEND	0xE030_0A2C	R/W	Non wake-up Interrupt 11 Pending Register	0x0
NWU_INT12_PEND	0xE030_0A30	R/W	Non wake-up Interrupt 12 Pending Register	0x0
NWU_INT13_PEND	0xE030_0A34	R/W	Non wake-up Interrupt 13 Pending Register	0x0
NWU_INT14_PEND	0xE030_0A38	R/W	Non wake-up Interrupt 14 Pending Register	0x0

Register	Address	R/W	Description	Reset Value
NWU_INT15_PEND	0xE030_0A3C	R/W	Non wake-up Interrupt 15 Pending Register	0x0
NWU_INT16_PEND	0xE030_0A40	R/W	Non wake-up Interrupt 16 Pending Register	0x0
NWU_INT17_PEND	0xE030_0A44	R/W	Non wake-up Interrupt 17 Pending Register	0x0
NWU_INT18_PEND	0xE030_0A48	R/W	Non wake-up Interrupt 18 Pending Register	0x0
NWU_INT19_PEND	0xE030_0A4C	R/W	Non wake-up Interrupt 19 Pending Register	0x0
NWU_INT20_PEND	0xE030_0A50	R/W	Non wake-up Interrupt 20 Pending Register	0x0
NWU_INT_GRPPRI	0xE030_0B00	R/W	Non wake-up Interrupt Group Priority Control Register	0x0
NWU_INT_PRIORITY	0xE030_0B04	R/W	Non wake-up Interrupt Priority Control Register	0x00
NWU_INT_SERVICE	0xE030_0B08	R	Current Service Register	0x00
NWU_INT_GRPFIXPRI	0xE030_0B10	R/W	Non wake-up Interrupt Group Fixed Priority Control Register	0x00
NWU_INT0_FIXPRI	0xE030_0B14	R/W	Non wake-up Interrupt 0 Fixed Priority Control Register	0x00
NWU_INT1_FIXPRI	0xE030_0B18	R/W	Non wake-up Interrupt 1 Fixed Priority Control Register	0x00
NWU_INT2_FIXPRI	0xE030_0B1C	R/W	Non wake-up Interrupt 2 Fixed Priority Control Register	0x00
NWU_INT3_FIXPRI	0xE030_0B20	R/W	Non wake-up Interrupt 3 Fixed Priority Control Register	0x00
NWU_INT4_FIXPRI	0xE030_0B24	R/W	Non wake-up Interrupt 4 Fixed Priority Control Register	0x00
NWU_INT5_FIXPRI	0xE030_0B28	R/W	Non wake-up Interrupt 5 Fixed Priority Control Register	0x00
NWU_INT6_FIXPRI	0xE030_0B2C	R/W	Non wake-up Interrupt 6 Fixed Priority Control Register	0x00
NWU_INT7_FIXPRI	0xE030_0B30	R/W	Non wake-up Interrupt 7 Fixed Priority Control Register	0x00
NWU_INT8_FIXPRI	0xE030_0B34	R/W	Non wake-up Interrupt 8 Fixed Priority Control Register	0x00
NWU_INT9_FIXPRI	0xE030_0B38	R/W	Non wake-up Interrupt 9 Fixed Priority Control Register	0x00
NWU_INT10_FIXPRI	0xE030_0B3C	R/W	Non wake-up Interrupt 10 Fixed Priority Control Register	0x00
NWU_INT11_FIXPRI	0xE030_0B40	R/W	Non wake-up Interrupt 11 Fixed Priority Control Register	0x00
NWU_INT12_FIXPRI	0xE030_0B44	R/W	Non wake-up Interrupt 12 Fixed Priority Control Register	0x00
NWU_INT13_FIXPRI	0xE030_0B48	R/W	Non wake-up Interrupt 13 Fixed Priority Control Register	0x00
NWU_INT14_FIXPRI	0xE030_0B4C	R/W	Non wake-up Interrupt 14 Fixed Priority Control Register	0x00
NWU_INT15_FIXPRI	0xE030_0B50	R/W	Non wake-up Interrupt 15 Fixed Priority Control Register	0x00
NWU_INT16_FIXPRI	0xE030_0B54	R/W	Non wake-up Interrupt 16 Fixed Priority Control Register	0x00
NWU_INT17_FIXPRI	0xE030_0B58	R/W	Non wake-up Interrupt 17 Fixed Priority Control Register	0x00
NWU_INT18_FIXPRI	0xE030_0B5C	R/W	Non wake-up Interrupt 18 Fixed Priority Control Register	0x00
NWU_INT19_FIXPRI	0xE030_0B60	R/W	Non wake-up Interrupt 19 Fixed Priority Control Register	0x00
NWU_INT20_FIXPRI	0xE030_0B64	R/W	Non wake-up Interrupt 20 Fixed Priority Control Register	0x00

Register	Address	R/W	Description	Reset Value
UHOST	0xE030_0B68	R/W	USB Host 1.1 Pad Configure Register	0

Register	Address	R/W	Description	Reset Value
WKUP_INT_CON0_7	0xE030_0E00	R/W	Wake-up Interrupt Configuration Register0_7	0x0
WKUP_INT_CON8_15	0xE030_0E04	R/W	Wake-up Interrupt Configuration Register8_15	0x0
WKUP_INT_CON16_23	0xE030_0E08	R/W	Wake-up Interrupt Configuration Register16_23	0x0
WKUP_INT_CON24_31	0xE030_0E0C	R/W	Wake-up Interrupt Configuration Register24_31	0x0
WKUP_INT_FLTCON0_3	0xE030_0E80	R/W	Wake-up Interrupt Filter Configuration Register 0_3	0x0
WKUP_INT_FLTCON4_7	0xE030_0E84	R/W	Wake-up Interrupt Filter Configuration Register 4_7	0x0
WKUP_INT_FLTCON8_11	0xE030_0E88	R/W	Wake-up Interrupt Filter Configuration Register 8_11	0x0
WKUP_INT_FLTCON12_15	0xE030_0E8C	R/W	Wake-up Interrupt Filter Configuration Register 12_15	0x0
WKUP_INT_FLTCON16_19	0xE030_0E90	R/W	Wake-up Interrupt Filter Configuration Register 16_19	0x0
WKUP_INT_FLTCON20_23	0xE030_0E94	R/W	Wake-up Interrupt Filter Configuration Register 20_23	0x0
WKUP_INT_FLTCON24_27	0xE030_0E98	R/W	Wake-up Interrupt Filter Configuration Register 24_27	0x0
WKUP_INT_FLTCON28_31	0xE030_0E9C	R/W	Wake-up Interrupt Filter Configuration Register 28_31	0x0
WKUP_INT_MASK0_7	0xE030_0F00	R/W	Wake-up Interrupt Mask Register0_7	0xff
WKUP_INT_MASK8_15	0xE030_0F04	R/W	Wake-up Interrupt Mask Register8_15	0xff
WKUP_INT_MASK16_23	0xE030_0F08	R/W	Wake-up Interrupt Mask Register16_23	0xff
WKUP_INT_MASK24_31	0xE030_0F0C	R/W	Wake-up Interrupt Mask Register24_31	0xff
WKUP_INT_PEND0_7	0xE030_0F40	R/W	Wake-up Interrupt Pending Register0_7	0x0
WKUP_INT_PEND8_15	0xE030_0F44	R/W	Wake-up Interrupt Pending Register8_15	0x0
WKUP_INT_PEND16_23	0xE030_0F48	R/W	Wake-up Interrupt Pending Register16_23	0x0
WKUP_INT_PEND24_31	0xE030_0F4C	R/W	Wake-up Interrupt Pending Register24_31	0x0

Register	Address	R/W	Description	Reset Value
PDNEN	0xE030_0F80	R/W	Power down mode Pad Configure Register	0

5.1 PORT CONFIGURATION REGISTERS

5.1.1 Port Group GPA0 Configuration Register (GPA0CON, R/W, Address = 0xE030_0000)

Field	Bit	Description	Reset Value
GPA0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = UART_0_RXD, 1111 = NWU_INT0[0]	0000
GPA0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = UART_0_TXD, 1111 = NWU_INT0[1]	0000
GPA0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = UART_0_CTSn, 1111 = NWU_INT0[2]	0000
GPA0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = UART_0_RTSn, 1111 = NWU_INT0[3]	0000
GPA0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = UART_1_RXD, 1111 = NWU_INT0[4]	0000
GPA0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = UART_1_TXD, 1111 = NWU_INT0[5]	0000
GPA0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = UART_1_CTSn, 1111 = NWU_INT0[6]	0000
GPA0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = UART_1_RTSn, 1111 = NWU_INT0[7]	0000

5.1.2 Port Group GPA1 Configuration Register (GPA1CON, R/W, Address = 0xE030_0020)

Field	Bit	Description	Reset Value
GPA1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = UART_2_RXD, 1111 = NWU_INT1[0]	0000
GPA1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = UART_2_TXD, 1111 = NWU_INT1[1]	0000
GPA1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = UART_3_RXD, 0011 = UART_2_CTSn, 0100 = IrDA_RXD, 1111 = NWU_INT1[2]	0000
GPA1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = UART_3_TXD, 0011 = UART_2_RTSn, 0100 = IrDA_TXD, 1111 = NWU_INT1[3]	0000
GPA1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = UARTCLK 0100 = IrDA_SDBW, 1111 = NWU_INT1[4]	0000

5.1.3 Port Group GPB Configuration Register (GPBCON, R/W, Address = 0xE030_0040)

Field	Bit	Description	Reset Value
GPBCON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = SPI_0_MISO, 1111 = NWU_INT2[0]	0000
GPBCON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = SPI_0_CLK, 1111 = NWU_INT2[1]	0000

GPBCON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SPI_0_MOSI, 1111 = NWU_INT2[2]	0000
GPBCON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = SPI_0_nSS, 1111 = NWU_INT2[3]	0000
GPBCON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = SPI_1_MISO, 1111 = NWU_INT2[4]	0000
GPBCON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = SPI_1_CLK, 1111 = NWU_INT2[5]	0000
GPBCON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = SPI_1_MOSI, 1111 = NWU_INT2[6]	0000
GPBCON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = SPI_1_nSS, 1111 = NWU_INT2[7]	0000

5.1.4 Port Group GPC Configuration Register (GPCCON, R/W, Address = 0xE030_0060)

Field	Bit	Description	Reset Value
GPCCON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = I2S1_SCLK, 0011 = PCM_1_SCLK, 0100 = AC97BITCLK, 1111 = NWU_INT3[0]	0000
GPCCON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = I2S1_CDCLK, 0011 = PCM_1_EXTCLK, 0100 = AC97RESETn, 1111 = NWU_INT3[1]	0000
GPCCON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = I2S1_LRCK, 0011 = PCM_1_FSYNC, 0100 = AC97SYNC, 1111 = NWU_INT3[2]	0000
GPCCON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = I2S1_SDI, 0011 = PCM_1_SIN, 0100 = AC97SDI, 1111 = NWU_INT3[3]	0000
GPCCON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = I2S1_SDO, 0011 = PCM_1_SOUT, 0100 = AC97SDO, 1111 = NWU_INT3[4]	0000

5.1.5 Port Group GPD Configuration Register (GPDCON, R/W, Address = 0xE030_0080)

Field	Bit	Description	Reset Value
GPDCON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = TOUT_0, 0011 = PWM_TCLK, 1111 = NWU_INT4[0]	0000
GPDCON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = TOUT_1, 0011 = EX_DMA_REQn, 1111 = NWU_INT4[1]	0000
GPDCON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = TOUT_2, 0011 = EX_DMA_ACKn, 1111 = NWU_INT4[2]	0000
GPDCON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = I2C0_SDA 1111 = NWU_INT4[3]	0000
GPDCON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = I2C0_SCL 1111 = NWU_INT4[4]	0000

GPDCON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = I2C1_SDA , 0011 = SPDIF_0_OUT, 1111 = NWU_INT4[5]	0000
GPDCON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = I2C1_SCL, 0011 = SPDIF_EXTCLK, 1111 = NWU_INT4[6]	0000

5.1.6 Port Group GPE0 Configuration Register (GPE0CON, R/W, Address = 0xE030_00A0)

Field	Bit	Description	Reset Value
GPE0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = CAM_A_PCLK, 0011 = SD_1_CLK, 1111 = NWU_INT5[0]	0000
GPE0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = CAM_A_VSYNC, 0011 = SD_1_CDn, 1111 = NWU_INT5[1]	0000
GPE0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = CAM_A_HREF, 0011 = SD_1_CMD, 1111 = NWU_INT5[2]	0000
GPE0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[0], 0011 = SD_1_D[0], 1111 = NWU_INT5[3]	0000
GPE0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[1], 0011 = SD_1_D[1], 1111 = NWU_INT5[4]	0000
GPE0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[2], 0011 = SD_1_D[2], 1111 = NWU_INT5[5]	0000
GPE0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[3], 0011 = SD_1_D[3], 1111 = NWU_INT5[6]	0000
GPE0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[4], 0011 = SD_1_D[4], 1111 = NWU_INT5[7]	0000

5.1.7 Port Group GPE1 Configuration Register (GPE1CON, R/W, Address = 0xE030_00C0)

Field	Bit	Description	Reset Value
GPE1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[5], 0011 = SD_1_D[5], 1111 = NWU_INT6[0]	0000
GPE1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[6], 0011 = SD_1_D[6], 1111 = NWU_INT6[1]	0000
GPE1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = CAM_A_D[7], 0011 = SD_1_D[7], 1111 = NWU_INT6[2]	0000
GPE1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = CAM_A_CLKOUT, 1111 = NWU_INT6[3]	0000
GPE1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = CAM_A_RESET, 1111 = NWU_INT6[4]	0000
GPE1CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = CAM_A_FIELD, 1111 = NWU_INT6[5]	0000

5.1.8 Port Group GPF0 Configuration Register (GPF0CON, R/W, Address = 0xE030_00E0)

Field	Bit	Description	Reset Value
GPF0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = LCD_HSYNC, 0011 = SYS_CS0, 0100 = VEN_HSYNC 1111 = NWU_INT7[0]	0000
GPF0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = LCD_VSYNC, 0011 = SYS_CS1, 0100 = VEN_VSYNC, 1111 = NWU_INT7[1]	0000
GPF0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = LCD_VDEN, 0011 = SYS_RS, 0100 = VEN_HREF, 1111 = NWU_INT7[2]	0000
GPF0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = LCD_VCLK, 0011 = SYS_WE, 0100 = V601_CLK, 1111 = NWU_INT7[3]	0000
GPF0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = LCD_VD[0], 0011 = SYS_VD[0] , 0100 = VEN_D[0], 1111 = NWU_INT7[4]	0000
GPF0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = LCD_VD[1], 0011 = SYS_VD[1] , 0100 = VEN_D[1], 1111 = NWU_INT7[5]	0000
GPF0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = LCD_VD[2], 0011 = SYS_VD[2] , 0100 = VEN_D[2], 1111 = NWU_INT7[6]	0000
GPF0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = LCD_VD[3] 0011 = SYS_VD[3] , 0100 = VEN_D[3], 1111 = NWU_INT7[7]	0000

5.1.9 Port Group GPF1 Configuration Register (GPF1CON, R/W, Address = 0xE030_0100)

Field	Bit	Description	Reset Value
GPF1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = LCD_VD[4], 0011 = SYS_VD[4], 0100 = VEN_D[4], 1111 = NWU_INT8[0]	0000
GPF1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = LCD_VD[5], 0011 = SYS_VD[5], 0100 = VEN_D[5], 1111 = NWU_INT8[1]	0000
GPF1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = LCD_VD[6], 0011 = SYS_VD[6], 0100 = VEN_D[6], 1111 = NWU_INT8[2]	0000
GPF1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = LCD_VD[7], 0011 = SYS_VD[7], 0100 = VEN_D[7], 1111 = NWU_INT8[3]	0000
GPF1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = LCD_VD[8], 0011 = SYS_VD[8], 0100 = V656_D[0], 1111 = NWU_INT8[4]	0000

GPF1CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = LCD_VD[9], 0011 = SYS_VD[9], 0100 = V656_D[1], 1111 = NWU_INT8[5]	0000
GPF1CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = LCD_VD[10], 0011 = SYS_VD[10], 0100 = V656_D[2], 1111 = NWU_INT8[6]	0000
GPF1CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = LCD_VD[11], 0011 = SYS_VD[11], 0100 = V656_D[3], 1111 = NWU_INT8[7]	0000

5.1.10 Port Group GPF2 Configuration Register (GPF2CON , R/W, Address = 0xE030_0120)

Field	Bit	Description	Reset Value
GPF2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = LCD_VD[12], 0011 = SYS_VD[12], 0100 = V656_D[4], 1111 = NWU_INT9[0]	0000
GPF2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = LCD_VD[13], 0011 = SYS_VD[13], 0100 = V656_D[5], 1111 = NWU_INT9[1]	0000
GPF2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = LCD_VD[14], 0011 = SYS_VD[14], 0100 = V656_D[6], 1111 = NWU_INT9[2]	0000
GPF2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = LCD_VD[15], 0011 = SYS_VD[15], 0100 = V656_D[7], 1111 = NWU_INT9[3]	0000
GPF2CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = LCD_VD[16], 0011 = SYS_VD[16], 1111 = NWU_INT9[4]	0000
GPF2CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = LCD_VD[17], 0011 = SYS_VD[17], 1111 = NWU_INT9[5]	0000
GPF2CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = LCD_VD[18], 1111 = NWU_INT9[6]	0000
GPF2CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = LCD_VD[19], 1111 = NWU_INT9[7]	0000

5.1.11 Port Group GPF3 Configuration Register (GPF3CON, R/W, Address = 0xE030_0140)

Field	Bit	Description	Reset Value
GPF3CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = LCD_VD[20], 1111 = NWU_INT10[0]	0000
GPF3CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = LCD_VD[21], 1111 = NWU_INT10[1]	0000
GPF3CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = LCD_VD[22], 0011 = VSYNC_LDI, 0100 = V656_CLK,	0000

		1111 = NWU_INT10[2]	
GPF3CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = LCD_VD[23], 0011 = SYS_OE, 0100 = VEN_FIELD, 1111 = NWU_INT10[3]	0000

5.1.12 Port Group GPG0 Configuration Register (GPG0CON, R/W, Address = 0xE030_0160)

Field	Bit	Description	Reset Value
GPG0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = SD_0_CLK, 1111 = NWU_INT11[0]	0000
GPG0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = SD_0_CMD, 1111 = NWU_INT11[1]	0000
GPG0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SD_0_D[0], 1111 = NWU_INT11[2]	0000
GPG0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = SD_0_D[1], 1111 = NWU_INT11[3]	0000
GPG0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = SD_0_D[2], 1111 = NWU_INT11[4]	0000
GPG0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = SD_0_D[3], 1111 = NWU_INT11[5]	0000
GPG0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = SD_0_D[4], 1111 = NWU_INT11[6]	0000
GPG0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = SD_0_D[5], 1111 = NWU_INT11[7]	0000

5.1.13 Port Group GPG1 Configuration Register (GPG1CON, R/W, Address = 0xE030_0180)

Field	Bit	Description	Reset Value
GPG1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = SD_0_D[6], 1111 = NWU_INT12[0]	0000
GPG1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = SD_0_D[7], 1111 = NWU_INT12[1]	0000
GPG1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SD_0_CDn, 1111 = NWU_INT12[2]	0000

5.1.14 Port Group GPG2 Configuration Register (GPG2CON, R/W, Address = 0xE030_01A0)

Field	Bit	Description	Reset Value
GPG2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = SD_1_CLK, 1111 = NWU_INT13[0]	0000
GPG2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = SD_1_CMD, 1111 = NWU_INT13[1]	0000
GPG2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SD_1_D[0], 1111 = NWU_INT13[2]	0000

GPG2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = SD_1_D[1], 1111 = NWU_INT13[3]	0000
GPG2CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = SD_1_D[2], 1111 = NWU_INT13[4]	0000
GPG2CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = SD_1_D[3], 1111 = NWU_INT13[5]	0000
GPG2CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = SD_1_CDn, 1111 = NWU_INT13[6]	0000

5.1.15 Port Group GPG3 Configuration Register (GPG3CON, R/W, Address = 0xE030_01C0)

Field	Bit	Description	Reset Value
GPG3CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = SD_2_CLK, 0011 = SPI_2_CLK, 0100 = I2S2_SCLK, 0101 = PCM_0_SCLK, 1111 = NWU_INT14[0]	0000
GPG3CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = SD_2_CMD, 0011 = SPI_2_nSS, 0100 = I2S2_CDCLK, 0101 = PCM_0_EXTCLK, 1111 = NWU_INT14[1]	0000
GPG3CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SD_2_D[0], 0011 = SPI_2_MISO, 0100 = I2S2_LRCK, 0101 = PCM_0_FSYNC, 1111 = NWU_INT14[2]	0000
GPG3CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = SD_2_D[1], 0011 = SPI_2_MOSI, 0100 = I2S2_SDI, 0101 = PCM_0_SIN, 1111 = NWU_INT14[3]	0000
GPG3CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = SD_2_D[2], 0011 = Reserved, 0100 = I2S2_SDO, 0101 = PCM_0_SOUT, 1111 = NWU_INT14[4]	0000
GPG3CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = SD_2_D[3], 0011 = Reserved, 0100 = Reserved, 0101 = SPDIF_0_OUT, 1111 = NWU_INT14[5]	0000
GPG3CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = SD_2_CDn, 0011 = Reserved, 0100 = Reserved, 0101 = SPDIF_EXTCLK, 1111 = NWU_INT14[6]	0000

5.1.16 Port Group GPH0 Configuration Register (GPH0CON, R/W, Address = 0xE030_0C00)

Field	Bit	Description	Reset Value
GPH0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[7]	0000
GPH0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[6]	0000
GPH0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[5]	0000
GPH0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[4]	0000
GPH0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[3]	0000
GPH0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[2]	0000
GPH0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[1]	0000

GPH0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = WKUP_INT[0]	0000

NOTE: WKUP_INT[x] are used for wake-up source in Power down and Idle mode, in Normal mode,these are same as NWU_INTx

5.1.17 Port Group GPH1 Configuration Register (GPH1CON, R/W, Address = 0xE030_0C20)

GPH1CON	Bit	Description	Reset Value
GPH1CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[15], 0011 = CG_GPO[3]	0000
GPH1CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[14], 0011 = CG_GPO[2]	0000
GPH1CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[13], 0011 = CG_GPO[1]	0000
GPH1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[12], 0011 = CG_GPO[0]	0000
GPH1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[11], 0011 = CG_IMGIN	0000
GPH1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[10], 0011 = CG_REALIN	0000
GPH1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[9]	0000
GPH1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[8]	0000

Note: WAKEUP_INT[x] are used for wake-up source in Power down and Idle mode, in Normal mode, these are same as NWU_INTx

5.1.18 Port Group GPH2 Configuration Register (GPH2CON, R/W, Address = 0xE030_0C40)

GPH2CON	Bit	Description	Reset Value
GPH2CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[23] , 0011 = KEYPAD_COL[7], 0100 = CAM_B_D[7]	0000
GPH2CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[22], 0011 = KEYPAD_COL[6], 0100 = CAM_B_D[6]	0000
GPH2CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[21], 0011 = KEYPAD_COL[5], 0100 = CAM_B_D[5]	0000
GPH2CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[20], 0011 = KEYPAD_COL[4], 0100 = CAM_B_D[4]	0000
GPH2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[19] , 0011 = KEYPAD_COL[3], 0100 = CAM_B_D[3]	0000
GPH2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[18] , 0011 = KEYPAD_COL[2], 0100 = CAM_B_D[2]	0000
GPH2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[17] , 0011 = KEYPAD_COL[1], 0100 = CAM_B_D[1]	0000

GPH2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[16] , 0011 = KEYPAD_COL[0], 0100 = CAM_B_D[0]	0000

Note: WAKEUP_INT[x] are used for wake-up source in Power down and Idle mode, in Normal mode, these are same as NWU_INTx

5.1.19 Port Group GPH3 Configuration Register (GPH3CON, R/W, Address = 0xE030_0C60)

GPH3CON	Bit	Description	Reset Value
GPH3CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[31] , 0011 = KEYPAD_ROW[7], 0100 = CAN1_RX	0000
GPH3CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[30] , 0011 = KEYPAD_ROW[6], 0100 = CAN1_TX	0000
GPH3CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[29] , 0011 = KEYPAD_ROW[5], 0100 = CAN0_RX	0000
GPH3CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[28] , 0011 = KEYPAD_ROW[4], 0100 = CAN0_TX	0000
GPH3CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[27] , 0011 = KEYPAD_ROW[3], 0100 = CAM_B_FIELD	0000
GPH3CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[26] , 0011 = KEYPAD_ROW[2], 0100 = CAM_B_HREF	0000
GPH3CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[25] , 0011 = KEYPAD_ROW[1], 0100 = CAM_B_PVSYNC	0000
GPH3CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = WAKEUP_INT[24] , 0011 = KEYPAD_ROW[0], 0100 = CAM_B_PCLK	0000

Note: WAKEUP_INT[x] are used for wake-up source in Power down and Idle mode, in Normal mode, these are same as NWU_INTx

5.1.20 Port Group GPI Configuration Register (GPICON, R/W, Address = 0xE030_01E0)

Field	Bit	Description	Reset Value
GPICON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = IEM_SCLK, 1111 = NWU_INT15[0]	0000
GPICON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = IEM_SPWI, 1111 = NWU_INT15[1]	0000
GPICON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = BOOT_OPT[0], 1111 = NWU_INT15[2]	0x2
GPICON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = BOOT_OPT[1], 1111 = NWU_INT15[3]	0x2
GPICON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = BOOT_OPT[2], 1111 = NWU_INT15[4]	0x2
GPICON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = BOOT_OPT[3], 1111 = NWU_INT15[5]	0x2
GPICON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = BOOT_OPT[4],	0x2

		1111 = NWU_INT15[6]	
GPICON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = BOOT_OPT[5], 1111 = NWU_INT15[7]	0x2

5.1.21 Port Group GPJ0 Configuration Register (GPJ0CON, R/W, Address = 0xE030_0200)

Field	Bit	Description	Reset Value
GPJ0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = MSM_A[0], 0011 = HSI_TXD, 0100 = CF_A[0], 1111 = NWU_INT16[0]	0000
GPJ0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = MSM_A[1], 0011 = HSI_TX_FLAG, 0100 = CF_A[1], 1111 = NWU_INT16[1]	0000
GPJ0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = MSM_A[2], 0011 = HSI_TX_WAKE, 0100 = CF_A[2], 1111 = NWU_INT16[2]	0000
GPJ0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = MSM_A[3], 0011 = HSI_TX_READY, 0100 = CF_IORDY, 1111 = NWU_INT16[3]	0000
GPJ0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = MSM_A[4], 0011 = HSI_RXD, 0100 = CF_INTRQ, 1111 = NWU_INT16[4]	0000
GPJ0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = MSM_A[5], 0011 = HSI_RX_FLAG , 0100 = CF_INPACKn, 1111 = NWU_INT16[5]	0000
GPJ0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = MSM_A[6], 0011 = HSI_RX_WAKE, 0100 = CF_RESET, 1111 = NWU_INT16[6]	0000
GPJ0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = MSM_A[7], 0011 = HSI_RX_READY, 0100 = CF_REG, 1111 = NWU_INT16[7]	0000

5.1.22 Port Group GPJ1 Configuration Register (GPJ1CON, R/W, Address = 0xE030_0220)

Field	Bit	Description	Reset Value
GPJ1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = MSM_A[8], 1111 = NWU_INT17[0]	0000
GPJ1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = MSM_A[9], 1111 = NWU_INT17[1]	0000
GPJ1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = MSM_A[10], 1111 = NWU_INT17[2]	0000
GPJ1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = MSM_A[11], 1111 = NWU_INT17[3]	0000
GPJ1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = MSM_A[12], 1111 = NWU_INT17[4]	0000

5.1.23 Port Group GPJ2 Configuration Register (GPJ2CON, R/W, Address = 0xE030_0240)

Field	Bit	Description	Reset Value
GPJ2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = MSM_D[0], 0011 = Reserved, 0100 = CF_D[0] 1111 = NWU_INT18[0]	0000
GPJ2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = MSM_D[1] , 0011 = Reserved, 0100 = CF_D[1] , 1111 = NWU_INT18[1]	0000
GPJ2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = MSM_D[2] , 0011 = Reserved, 0100 = CF_D[2] , 1111 = NWU_INT18[2]	0000
GPJ2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = MSM_D[3] , 0011 = Reserved, 0100 = CF_D[3] , 1111 = NWU_INT18[3]	0000
GPJ2CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = MSM_D[4], 0011 = Reserved, 0100 = CF_D[4], 1111 = NWU_INT18[4]	0000
GPJ2CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = MSM_D[5], 0011 = Reserved, 0100 = CF_D[5], 1111 = NWU_INT18[5]	0000
GPJ2CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = MSM_D[6] , 0011 = Reserved, 0100 = CF_D[6], 1111 = NWU_INT18[6]	0000
GPJ2CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = MSM_D[7] , 0011 = Reserved, 0100 = CF_D[7], 1111 = NWU_INT18[7]	0000

5.1.24 Port Group GPJ3 Configuration Register (GPJ3CON, R/W, Address = 0xE030_0260)

Field	Bit	Description	Reset Value
GPJ3CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = MSM_D[8] 0011 = Reserved, 0100 = CF_D[8], 1111 = NWU_INT19[0]	0000
GPJ3CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = MSM_D[9] 0011 = Reserved, 0100 = CF_D[9], 1111 = NWU_INT19[1]	0000
GPJ3CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = MSM_D[10] 0011 = Reserved, 0100 = CF_D[10], 1111 = NWU_INT19[2]	0000
GPJ3CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = MSM_D[11] 0011 = Reserved, 0100 = CF_D[11], 1111 = NWU_INT19[3]	0000
GPJ3CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = MSM_D[12] 0011 = Reserved, 0100 = CF_D[12], 1111 = NWU_INT19[4]	0000

Field	Bit	Description	Reset Value
GPJ3CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = MSM_D[13] 0011 = Reserved, 0100 = CF_D[13], 1111 = NWU_INT19[5]	0000
GPJ3CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = MSM_D[14] 0011 = Reserved, 0100 = CF_D[14], 1111 = NWU_INT19[6]	0000
GPJ3CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = MSM_D[15] 0011 = Reserved, 0100 = CF_D[15], 1111 = NWU_INT19[7]	0000

5.1.25 Port Group GPJ4 Configuration Register (GPJ4CON, R/W, Address = 0xE030_0280)

Field	Bit	Description	Reset Value
GPJ4CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = MSM_CSn, 0011 = Reserved, 0100 = CF_nCS[0], 1111 = NWU_INT20[0]	0000
GPJ4CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = MSM_WEn, 0011 = Reserved, 0100 = CF_nCS[1], 1111 = NWU_INT20[1]	0000
GPJ4CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = MSM_Rn, 0011 = Reserved, 0100 = CF_IORD_CFn, 1111 = NWU_INT20[2]	0000
GPJ4CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = MSM_IRQn, 0011 = Reserved, 0100 = CF_IOWR_CFn, 1111 = NWU_INT20[3]	0000

5.1.26 Port Group GPK0 Configuration Register (GPK0CON, R/W, Address = 0xE030_02A0)

Field	Bit	Description	Reset Value
GPK0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = SMC_CSn[0]	0010
GPK0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = SMC_CSn[1]	0010
GPK0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SMC_CSn[2] , 0011 = NFCSn[0] , 0100 = Reserved, 0101 = OND_CSn[0]	0101
GPK0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = SMC_CSn[3] , 0011 = NFCSn[1], 0100 = Reserved, 0101 = OND_CSn[1]	0101
GPK0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = SMC_CSn[4] , 0011 = NFCSn[2], 0100 = CF_CSn[0]	0100
GPK0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = SMC_CSn[5] , 0011 = NFCSn[3], 0100 = CF_CSn[1]	0100
GPK0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = EBI_OEn	0010
GPK0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = EBI_WEn	0010

5.1.27 Port Group GPK1 Configuration Register (GPK1CON, R/W, Address = 0xE030_02C0)

Field	Bit	Description	Reset Value
GPK1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = EBI_Ben[0]	0010
GPK1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = EBI_Ben[1]	0010
GPK1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = SMC_WAITn	0010
GPK1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = EBI_DATA_RDn	0010
GPK1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = CF_OEn	0010
GPK1CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = CF_WEn	0010

5.1.28 Port Group GPK2 Configuration Register (GPK2CON, R/W, Address = 0xE030_02E0)

Field	Bit	Description	Reset Value
GPK2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = Reserved, 0011 = NF_CLE, 0100= Reserved, 0101 = OND_ADDRVALID	0101
GPK2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = Reserved, 0011 = NF_ALE, 0011 ~ 0100= Reserved, 0101 = OND_SMCLK	0101
GPK2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = Reserved, 0011 = NF_FWEn, 0100= Reserved, 0101 = OND_RPn	0101
GPK2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = Reserved, 0011 = NF_FREn	0011
GPK2CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = NF_RnB[0] , 0011 ~ 0100= Reserved, 0101 = OND_INT[0]	0101
GPK2CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = NF_RnB[1] , 0011 ~ 0100= Reserved, 0101 = OND_INT[1]	0101
GPK2CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = NF_RnB[2]	0011
GPK2CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = NF_RnB[3]	0011

5.1.29 Port Group GPK3 Configuration Register (GPK3CON, R/W, Address = 0xE030_0300)

Field	Bit	Description	Reset Value
GPK3CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = CF_IORDY	0010
GPK3CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = CF_INTRQ	0010
GPK3CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = CF_RESET	0010
GPK3CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = CF_INPACKn	0010
GPK3CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = CF_REG	0010
GPK3CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = CF_CDn	0010
GPK3CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = CF_IORD_CFn	0010
GPK3CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = CF_IOWR_CFn	0010

5.1.30 Port Group GPL0 Configuration Register (GPL0CON, R/W, Address = 0xE030_0320)

Field	Bit	Description	Reset Value
GPL0CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = EBI_A[0]	0010
GPL0CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = EBI_A[1]	0010
GPL0CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = EBI_A[2]	0010
GPL0CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = EBI_A[3]	0010
GPL0CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = EBI_A[4]	0010
GPL0CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = EBI_A[5]	0010
GPL0CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = EBI_A[6]	0010
GPL0CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = EBI_A[7]	0010

5.1.31 Port Group GPL1 Configuration Register (GPL1CON, R/W, Address = 0xE030_0340)

Field	Bit	Description	Reset Value
GPL1CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = EBI_A[8]	0010
GPL1CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = EBI_A[9]	0010
GPL1CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = EBI_A[10]	0010
GPL1CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = EBI_A[11]	0010
GPL1CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = EBI_A[12]	0010
GPL1CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = EBI_A[13]	0010
GPL1CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = EBI_A[14]	0010
GPL1CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = EBI_A[15]	0010

5.1.32 Port Group GPL2 Configuration Register (GPL2CON, R/W, Address = 0xE030_0360)

Field	Bit	Description	Reset Value
GPL2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = EBI_A[16]	0010
GPL2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = EBI_A[17]	0010
GPL2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = EBI_A[18]	0010
GPL2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = EBI_A[19]	0010
GPL2CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = EBI_A[20]	0010
GPL2CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = EBI_D[0]	0010
GPL2CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = EBI_D[1]	0010
GPL2CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = EBI_D[2]	0010

5.1.33 Port Group GPL3 Configuration Register (GPL3CON, R/W, Address = 0xE030_0380)

Field	Bit	Description	Reset Value
GPL3CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = EBI_D[3]	0010

GPL3CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = EBI_D[4]	0010
GPL3CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = EBI_D[5]	0010
GPL3CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = EBI_D[6]	0010
GPL3CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = EBI_D[7]	0010
GPL3CON[5]	[23:20]	0000 = Input, 0001 = Output, 0010 = EBI_D[8]	0010
GPL3CON[6]	[27:24]	0000 = Input, 0001 = Output, 0010 = EBI_D[9]	0010
GPL3CON[7]	[31:28]	0000 = Input, 0001 = Output, 0010 = EBI_D[10]	0010

5.1.34 Port Group GPL4 Configuration Register (GPL4CON, R/W, Address = 0xE030_03A0)

Field	Bit	Description	Reset Value
GPL4CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = EBI_D[11]	0010
GPL4CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = EBI_D[12]	0010
GPL4CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = EBI_D[13]	0010
GPL4CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = EBI_D[14]	0010
GPL4CON[4]	[19:16]	0000 = Input, 0001 = Output, 0010 = EBI_D[15]	0010

5.2 PORT DATA & PULLING REGISTER

5.2.1 Port Data Registers

- GPA0DAT, R/W, Address = 0xE030_0004
- GPA1DAT, R/W, Address = 0xE030_0024
- GPBDAT, R/W, Address = 0xE030_0044
- GPCDAT, R/W, Address = 0xE030_0064
- GPDDAT, R/W, Address = 0xE030_0084
- GPE0DAT, R/W, Address = 0xE030_00A4
- GPE1DAT, R/W, Address = 0xE030_00C4
- GPF0DAT, R/W, Address = 0xE030_00E4
- GPF1DAT, R/W, Address = 0xE030_0104
- GPF2DAT, R/W, Address = 0xE030_0124
- GPF3DAT, R/W, Address = 0xE030_0144
- GPG0DAT, R/W, Address = 0xE030_0164
- GPG1DAT, R/W, Address = 0xE030_0184
- GPG2DAT, R/W, Address = 0xE030_01A4
- GPG3DAT, R/W, Address = 0xE030_01C4
- GPH0DAT, R/W, Address = 0xE030_0C04.
- GPH1DAT, R/W, Address = 0xE030_0C24.
- GPH2DAT, R/W, Address = 0xE030_0C44.

- GPH3DAT, R/W, Address = 0xE030_0C64.
- GPIDAT, R/W, Address = 0xE030_01E4
- GPJ0DAT, R/W, Address = 0xE030_0204
- GPJ1DAT, R/W, Address = 0xE030_0224
- GPJ2DAT, R/W, Address = 0xE030_0244
- GPJ3DAT, R/W, Address = 0xE030_0264
- GPJ4DAT, R/W, Address = 0xE030_0284
- GPK0DAT, R/W, Address = 0xE030_02A4
- GPK1DAT, R/W, Address = 0xE030_02C4
- GPK2DAT, R/W, Address = 0xE030_02E4
- GPK3DAT, R/W, Address = 0xE030_0304
- GPL0DAT, R/W, Address = 0xE030_0324
- GPL1DAT, R/W, Address = 0xE030_0344
- GPL2DAT, R/W, Address = 0xE030_0364
- GPL3DAT, R/W, Address = 0xE030_0384
- GPL4DAT, R/W, Address = 0xE030_03A4

Field	Bit	Description	Reset Value
DAT[n] (n=0~7)	[n]	If the bit is configured as input, it represents the pin state. If the bit is configured as output, the pin state is the same as the value of the bit. If the port is configured as functional pin, an undefined value is read.	-

5.2.2 Port Pull-up/down Register

- GPA0PULL, R/W, Address = 0xE030_0008.
- GPA1PULL, R/W, Address = 0xE030_0028.
- GPBPULL, R/W, Address = 0xE030_0048.
- GPCPULL, R/W, Address = 0xE030_0068.
- GPDPULL, R/W, Address = 0xE030_0088.
- GPE0PULL, R/W, Address = 0xE030_00A8.
- GPE1PULL, R/W, Address = 0xE030_00C8.
- GPF0PULL, R/W, Address = 0xE030_00E8.
- GPF1PULL, R/W, Address = 0xE030_0108.
- GPF2PULL, R/W, Address = 0xE030_0128.
- GPF3PUD, R/W, Address = 0xE030_0148.
- GPG0PULL, R/W, Address = 0xE030_0168.
- GPG1PULL, R/W, Address = 0xE030_0188.
- GPG2PULL, R/W, Address = 0xE030_01A8.

- GPG3PUD, R/W, Address = 0xE030_01C8.
- GPH0PULL, R/W, Address = 0xE030_0C08.
- GPH1PULL, R/W, Address = 0xE030_0C28.
- GPH2PULL, R/W, Address = 0xE030_0C48.
- GPH3PULL, R/W, Address = 0xE030_0C68.
- GPIPUD, R/W, Address = 0xE030_01E8.
- GPJ0PULL, R/W, Address = 0xE030_0208.
- GPJ1PULL, R/W, Address = 0xE030_0228.
- GPJ2PULL, R/W, Address = 0xE030_0248.
- GPJ3PUD, R/W, Address = 0xE030_0268.
- GPJ4PULL, R/W, Address = 0xE030_0288.
- GPK0PULL, R/W, Address = 0xE030_02A8.
- GPK1PULL, R/W, Address = 0xE030_02C8.
- GPK2PULL, R/W, Address = 0xE030_02E8.
- GPK3PUD, R/W, Address = 0xE030_0308.
- GPL0PULL, R/W, Address = 0xE030_0328.
- GPL1PULL, R/W, Address = 0xE030_0348.
- GPL2PULL, R/W, Address = 0xE030_0368.
- GPL3PUD, R/W, Address = 0xE030_0388.
- GPL4PULL, R/W, Address = 0xE030_03A8.

Field	Bit	Description	Reset Value
PUD[n] (n=0~7)	[2n+1:2n]	00 = Disables Pull-up/down 01 = Enables Pull-down 10 = Enables Pull-up 11 = Reserved	Refer to the Pin Summary table

5.2.3 Port Drive Strength Control Register

- GPA0DRV, R/W, Address = 0xE030_000C.
- GPA1DRV, R/W, Address = 0xE030_002C.
- GPBDRV, R/W, Address = 0xE030_004C.
- GPCDRV, R/W, Address = 0xE030_006C.
- GPDDRV, R/W, Address = 0xE030_008C.
- GPE0DRV, R/W, Address = 0xE030_00AC.
- GPE1DRV, R/W, Address = 0xE030_00CC.
- GPF0DRV, R/W, Address = 0xE030_00EC.
- GPF1DRV, R/W, Address = 0xE030_010C.
- GPF2DRV, R/W, Address = 0xE030_012C.

- GPF3DRV, R/W, Address = 0xE030_014C.
- GPG0DRV, R/W, Address = 0xE030_016C.
- GPG1DRV, R/W, Address = 0xE030_018C.
- GPG2DRV, R/W, Address = 0xE030_01AC.
- GPG3DRV, R/W, Address = 0xE030_01CC.
- GPH0DRV, R/W, Address = 0xE030_0C0C.
- GPH1DRV, R/W, Address = 0xE030_0C2C.
- GPH2DRV, R/W, Address = 0xE030_0C4C.
- GPH3DRV, R/W, Address = 0xE030_0C6C.
- GPIDRV, R/W, Address = 0xE030_01EC.
- GPJ0DRV, R/W, Address = 0xE030_020C.
- GPJ1DRV, R/W, Address = 0xE030_022C.
- GPJ2DRV, R/W, Address = 0xE030_024C.
- GPJ3DRV, R/W, Address = 0xE030_026C.
- GPJ4DRV, R/W, Address = 0xE030_028C.
- GPK0DRV, R/W, Address = 0xE030_02AC.
- GPK1DRV, R/W, Address = 0xE030_02CC.
- GPK2DRV, R/W, Address = 0xE030_02EC.
- GPK3DRV, R/W, Address = 0xE030_030C.
- GPL0DRV, R/W, Address = 0xE030_032C.
- GPL1DRV, R/W, Address = 0xE030_034C.
- GPL2DRV, R/W, Address = 0xE030_036C.
- GPL3DRV, R/W, Address = 0xE030_038C.
- GPL4DRV, R/W, Address = 0xE030_03AC.

Field	Bit	Description	Reset Value
DRV[n] (n=0~7)	[2n+1:2n]	Output Driver strength control register. These registers are used in both Normal mode and Power-Down mode 00 = 1x, 01 = 2x, 10 = 3x, 11 = 4x	Refer to the Pin Summary table

5.2.4 Port Power Down Mode Configuration Register

- GPA0PDNCON, R/W, Address = 0xE030_0010.
- GPA1PDNCON, R/W, Address = 0xE030_0030.
- GPBPDNCON, R/W, Address = 0xE030_0050.
- GPCPDNCON, R/W, Address = 0xE030_0070.
- GPDPDNCON, R/W, Address = 0xE030_0090.
- GPE0PDNCON, R/W, Address = 0xE030_00B0.
- GPE1PDNCON, R/W, Address = 0xE030_00D0.

- GPF0PDNCON, R/W, Address = 0xE030_00F0.
- GPF1PDNCON, R/W, Address = 0xE030_0110.
- GPF2PDNCON, R/W, Address = 0xE030_0130.
- GPF3PDNCON, R/W, Address = 0xE030_0150.
- GPG0PDNCON, R/W, Address = 0xE030_0170.
- GPG1PDNCON, R/W, Address = 0xE030_0190.
- GPG2PDNCON, R/W, Address = 0xE030_01B0.
- GPG3PDNCON, R/W, Address = 0xE030_01D0.
- GPIPDNCON, R/W, Address = 0xE030_01F0.
- GPJ0PDNCON, R/W, Address = 0xE030_0210.
- GPJ1PDNCON, R/W, Address = 0xE030_0230.
- GPJ2PDNCON, R/W, Address = 0xE030_0250.
- GPJ3PDNCON, R/W, Address = 0xE030_0270.
- GPJ4PDNCON, R/W, Address = 0xE030_0290.
- GPK0PDNCON, R/W, Address = 0xE030_02B0.
- GPK1PDNCON, R/W, Address = 0xE030_02E0.
- GPK2PDNCON, R/W, Address = 0xE030_02F0.
- GPK3PDNCON, R/W, Address = 0xE030_0310.
- GPL0PDNCON, R/W, Address = 0xE030_0330.
- GPL1PDNCON, R/W, Address = 0xE030_0350.
- GPL2PDNCON, R/W, Address = 0xE030_0370.
- GPL3PDNCON, R/W, Address = 0xE030_0390.
- GPL4PDNCON, R/W, Address = 0xE030_03B0.

Field	Bit	Description	Reset Value
PDNCON[n] (n=0~7)	[2n+1:2n]	00 = Output 0 01 = Output 1 10 = Input 11 = Previous state	Refer to the Pin Summary table

5.2.5 Port Power Down Mode Pull-up/down Register

- GPA0PDNPULL, R/W, Address = 0xE030_0014.
- GPA1PDNPULL, R/W, Address = 0xE030_0034.
- GPBPDNPULL, R/W, Address = 0xE030_0054.
- GPCPDNPULL, R/W, Address = 0xE030_0074.
- GPDPDNPULL, R/W, Address = 0xE030_0094.
- GPE0PDNPULL, R/W, Address = 0xE030_00B4.
- GPE1PDNPULL, R/W, Address = 0xE030_00D4.

- GPF0PDNPULL, R/W, Address = 0xE030_00F4.
- GPF1PDNPULL, R/W, Address = 0xE030_0114.
- GPF2PDNPULL, R/W, Address = 0xE030_0134.
- GPF3PDNPULL, R/W, Address = 0xE030_0154.
- GPG0PDNPULL, R/W, Address = 0xE030_0174.
- GPG1PDNPULL, R/W, Address = 0xE030_0194.
- GPG2PDNPULL, R/W, Address = 0xE030_01B4.
- GPG3PDNPULL, R/W, Address = 0xE030_01D4.
- GPIPUDPDN, R/W, Address = 0xE030_01F4.
- GPJ0PDNPULL, R/W, Address = 0xE030_0214.
- GPJ1PDNPULL, R/W, Address = 0xE030_0234.
- GPJ2PDNPULL, R/W, Address = 0xE030_0254.
- GPJ3PDNPULL, R/W, Address = 0xE030_0274.
- GPJ4PDNPULL, R/W, Address = 0xE030_0294.
- GPK0PDNPULL, R/W, Address = 0xE030_02B4.
- GPK1PDNPULL, R/W, Address = 0xE030_02E4.
- GPK2PDNPULL, R/W, Address = 0xE030_02F4.
- GPK3PDNPULL, R/W, Address = 0xE030_0314.
- GPL0PDNPULL, R/W, Address = 0xE030_0334.
- GPL1PDNPULL, R/W, Address = 0xE030_0354.
- GPL2PDNPULL, R/W, Address = 0xE030_0374.
- GPL3PDNPULL, R/W, Address = 0xE030_0394.
- GPL4PDNPULL, R/W, Address = 0xE030_03B4.

Field	Bit	Description	Reset Value
PDNPULL[n] (n=0~7)	[2n+1:2n]	00 = Disables Pull-up/down 01 = Enables Pull-down 10 = Pull-up enabled 11 = Reserved	Refer to the Pin Summary table

5.3 PORT GROUP MP REGISTERS

Port Group	Signal	IO	Port Group	Signal	IO
MP1_0[0]	DDR_CKE[0]	O	MP1_4[2]	DDR_D[9]	IO
MP1_0[1]	DDR_CKE[1]	O	MP1_4[3]	DDR_D[10]	IO
MP1_0[2]	DDR_SCLK	O	MP1_4[4]	DDR_D[11]	IO
MP1_0[3]	DDR_nSCLK	O	MP1_4[5]	DDR_D[12]	IO
MP1_0[4]	DDR_CSn[0]	O	MP1_4[6]	DDR_D[13]	IO
MP1_0[5]	DDR_CSn[1]	O	MP1_4[7]	DDR_D[14]	IO
MP1_0[6]	DDR_A[0]	O	MP1_5[0]	DDR_D[15]	IO
MP1_0[7]	DDR_A[1]	O	MP1_5[1]	DDR_D[16]	IO
MP1_1[0]	DDR_A[2]	O	MP1_5[2]	DDR_D[17]	IO
MP1_1[1]	DDR_A[3]	O	MP1_5[3]	DDR_D[18]	IO
MP1_1[2]	DDR_A[4]	O	MP1_5[4]	DDR_D[19]	IO
MP1_1[3]	DDR_A[5]	O	MP1_5[5]	DDR_D[20]	IO
MP1_1[4]	DDR_A[6]	O	MP1_5[6]	DDR_D[21]	IO
MP1_1[5]	DDR_A[7]	O	MP1_5[7]	DDR_D[22]	IO
MP1_1[6]	DDR_A[8]	O	MP1_6[0]	DDR_D[23]	IO
MP1_1[7]	DDR_A[9]	O	MP1_6[1]	DDR_D[24]	IO
MP1_2[0]	DDR_A[10]	O	MP1_6[2]	DDR_D[25]	IO
MP1_2[1]	DDR_A[11]	O	MP1_6[3]	DDR_D[26]	IO
MP1_2[2]	DDR_A[12]	O	MP1_6[4]	DDR_D[27]	IO
MP1_2[3]	DDR_A[13]	O	MP1_6[5]	DDR_D[28]	IO
MP1_2[4]	DDR_A[14]	O	MP1_6[6]	DDR_D[29]	IO
MP1_2[5]	DDR_A[15]	O	MP1_6[7]	DDR_D[30]	IO
MP1_2[6]	DDR_RASn	O	MP1_7[0]	DDR_D[31]	IO
MP1_2[7]	DDR_CASn	O	MP1_7[1]	DDR_DQM[0]	O
MP1_3[0]	DDR_WEn	O	MP1_7[2]	DDR_DQM[1]	O
MP1_3[1]	DDR_D[0]	IO	MP1_7[3]	DDR_DQM[2]	O
MP1_3[2]	DDR_D[1]	IO	MP1_7[4]	DDR_DQM[3]	O
MP1_3[3]	DDR_D[2]	IO	MP1_7[5]	DDR_DQS[0]	O
MP1_3[4]	DDR_D[3]	IO	MP1_7[6]	DDR_DQS[1]	O
MP1_3[5]	DDR_D[4]	IO	MP1_7[7]	DDR_DQS[2]	O
MP1_3[6]	DDR_D[5]	IO	MP1_8[0]	DDR_DQS[3]	O

Port Group	Signal	IO	Port Group	Signal	IO
MP1_3[7]	DDR_D[6]	IO	MP1_8[1]	LD2_DQSn[0]	O
MP1_4[0]	DDR_D[7]	IO	MP1_8[2]	LD2_DQSn[1]	O
MP1_4[1]	DDR_D[8]	IO	MP1_8[3]	LD2_DQSn[2]	O
			MP1_8[4]	LD2_DQSn[3]	O

5.3.1 Port Group MP1_X Drive Strength Control Register

- MP1_0DRV, R/W, Address = 0xE030_03CC.
- MP1_1DRV, R/W, Address = 0xE030_03EC.
- MP1_2DRV, R/W, Address = 0xE030_040C.
- MP1_3DRV, R/W, Address = 0xE030_042C.
- MP1_4DRV, R/W, Address = 0xE030_044C.
- MP1_5DRV, R/W, Address = 0xE030_046C.
- MP1_6DRV, R/W, Address = 0xE030_048C.
- MP1_7DRV, R/W, Address = 0xE030_04AC.
- MP1_8DRV, R/W, Address = 0xE030_04CC (Reset Value = 0x2AAA)

Field	Bit	Description	Reset Value
DRV[n] (n=0~7)	[2n+1:2n]	In case of VDD_DDR = 1.2V 00 = 2mA, 01 = 4mA, 10 = 6mA, 11 = 8mA In case of VDD_DDR = 1.8V 00 = 4mA, 01 = 8mA, 10 = 12mA, 11 = 16mA	0xAAAA

5.4 PORT GROUP ETC CONTROL REGISTER

Port Group ETCx controls 5 ports

5.4.1 Port Group ETC0 Pull-up/down Register (ETC0PULL, R/W, Address = 0xE030_04E8)

00 = Disables Pull-up/down, 01 = Enables Pull-down, 10 = Enables Pull-up

Field	Bit	Description	Reset Value
Reserved	[3:0]	Reserved	0xA
ETC0PULL[2]	[5:4]	XjTCK pulling control register	0x2
Reserved	[7:6]	Reserved	0x2
ETC0PULL[4]	[9:8]	XjTDO pulling control register	0x0
ETC0PULL[5]	[11:10]	XjDBGSEL pulling control register	0x1

NOTE: XjTRSTn, XjTMS, XjTDI, XnRESET, XnWRESET, XnBATF, XOM[4:0] are fixed to pull-up

5.4.2 Port Group ETC0 Drive strength control Register (ETC0DRV, R/W, Address = 0xE030_04EC)

00=1x, 01=2x, 10=3x, 11=4x.

Field	Bit	Description	Reset Value
ETC0DRV[0]	[1:0]	XjTRSTn Drive strength control register	0x0
ETC0DRV[1]	[3:2]	XjTMS Drive strength control register	0x0
ETC0DRV[2]	[5:4]	XjTCK Drive strength control register	0x0
ETC0DRV[3]	[7:6]	XjTDI Drive strength control register	0x0
ETC0DRV[4]	[9:8]	XjTDO Drive strength control register	0x0
ETC0DRV[5]	[11:10]	XjDBGSEL Drive strength control register	0x0

5.4.3 Port Group ETC1 Pull-up/down Register (ETC1PULL, R/W, Address = 0xE030_0508)

00 = Disables Pull-up/down, 01 = Enables Pull-down, 10 = Enables Pull-up

Field	Bit	Description	Reset Value
Reserved	[9:0]	Reserved	0x155
ETC1PULL[5]	[11:10]	XDDR2_SEL pulling control register	0x1
Reserved	[15:12]	Reserved	0x8

5.4.4 Port Group ETC1 Drive strength control Register (ETC1DRV, R/W, Address = 0xE030_050C)

00=1x, 01=2x, 10=3x, 11=4x.

Field	Bit	Description	Reset Value
Reseved	[9:0]	Reserved	0x000
ETC1DRV[5]	[11:10]	XDDR2_SEL pin Drive strength control register	0x0

5.4.5 Port Group ETC2 Pull-up/down Register (ETC2PULL, R/W, Address = 0xE030_0528)

00 = Disables Pull-up/down, 01 = Enables Pull-down, 10 = Enables Pull-up

Field	Bit	Description	Reset Value
Reserved	[3:0]	Reserved	0x2
ETC2PULL[2]	[5:4]	XnRSTOUT pin Pull-down control register	0x0
Reserved	[7:6]	Reserved	0x2

5.4.6 Port Group ETC2 Drive Strength Control Register (ETC2DRV, R/W, Address = 0xE030_052C)

00=1x, 01=2x, 10=3x, 11=4x.

Field	Bit	Description	Reset Value
Reserved	[3:0]	Reserved	0x0
ETC2DRV[3]	[5:4]	XnRSTOUT pin Drive strength control register	0x0

Reserved	[7:6]	Reserved	0x0

5.4.7 Port Group ETC3 Pull-up/down Register (ETC3PUD, R/W, Address = 0xE030_0548)

00 = Disables Pull-up/down, 01 = Enables Pull-down, 10 = Enables Pull-up

Field	Bit	Description	Reset Value
ETC3PUD[0]	[1:0]	XI2S0LRCK pin Pull-down control register	0x0
ETC3PUD[1]	[3:2]	XI2S0CDCLK pin Pull-down control register	0x0
ETC3PUD[2]	[5:4]	XI2S0SCLK pin Pull-down control register	0x0
ETC3PUD[3]	[7:6]	XI2S0SDI pin Pull-down control register	0x1
ETC3PUD[4]	[9:8]	XI2S0SDO[0] pin Pull-down control register	0x0
ETC3PUD[5]	[11:10]	XI2S0SDO[1] pin Pull-down control register	0x0
ETC3PUD[6]	[13:12]	XI2S0SDO[2] pin Pull-down control register	0x0

5.4.8 Port Group ETC3 Drive Strength Control Register (ETC3DRV, R/W, Address = 0xE030_054C)

00=1x, 01=2x, 10=3x, 11=4x.

Field	Bit	Description	Reset Value
ETC3DRV[0]	[1:0]	XI2S0LRCK pin Drive strength control register	0x0
ETC3DRV[1]	[3:2]	XI2S0CDCLK pin Drive strength control register	0x0
ETC3DRV[2]	[5:4]	XI2S0SCLK pin Drive strength control register	0x0
ETC3DRV[3]	[7:6]	XI2S0SDI pin Drive strength control register	0x0
ETC3DRV[4]	[9:8]	XI2S0SDO[0] pin Drive strength control register	0x0
ETC3DRV[5]	[11:10]	XI2S0SDO[1] pin Drive strength control register	0x0
ETC3DRV[6]	[13:12]	XI2S0SDO[2] pin Drive strength control register	0x0

5.4.9 Port Group ETC4 Drive strength control Register (ETC4DRV, R/W, Address = 0xE030_056C)

00=1x, 01=2x, 10=3x, 11=4x.

Field	Bit	Description	Reset Value
ETC4DRV[0]	[1:0]	X27mXTI/X27mXTo pin Drive strength control register	0x0
Reserved	[3:2]	Reserved	0x0
ETC4DRV[2]	[5:4]	XXTI/XXTO pin Drive strength control register	0x0
Reserved	[7:6]	Reserved	0x0
ETC4DRV[4]	[9:8]	XusbXTI/XusbXTO pin Drive strength control register	0x0
ETC4DRV[5]	[10]	XrtcXTI/XrtcXTO pin Drive strength control register 0 = 1x, 1 = 2x	0x0
Reserved	[13:11]	Reserved	0x0

5.5 NON WAKE-UP INTERRUPT CONTROL REGISTERS

GPIO interrupt consist of Non wake-up interrupt(NWU_INT) and Wake-up interrupt(WKUP). Non Wake-up Interrupt is consists of 21 groups numbered from 0 to 20. Wake-up Interrupt consists of 32.

In interrupt function, understanding the filter operation is essential. S5PC100 uses two types of filters to detect interrupt. They are delay filter and digital filter.

Delay filter uses delay cell. In case clock is not serviced, delay filter must be selected in alive area. Delay filter enables interrupt to be detected after 35ns from the time when the interrupt is occurred.

Digital filter means that all interrupt counts are based on clock. So, this filter can be used in clock-supported area.(off area and alive area) When digital filter is selected, filtering width should be set. Digital filter can detect interrupt per every clock count as many as filtering width. Filtering width is 6-bit in alive area and is 7-bit in off area.

When you use interrupt function, you set either delay filter or digital filter enabled in order to detect interrupt. If filter is disabled, there is strong probability that system detect all interrupt from successive interrupts.(Some interrupt detection will be missed.)

Non wake-up interrupt cannot use for wake-up source. If you need wake-up interrupt source, you can use Wake-up interrupt and you need to set delay filter for the Wake-up interrupt.

5.5.1 Non wake-up Interrupt 0 Configuration Register (NWU_INT0_CON, R/W, Address = 0xE030_0700)

Field	Bit	Description	Reset Value
NWU_INT0_CON[n]	[n*4+2:n*4]	Sets the signaling method for NWU_INT0[n] (n=0,1,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.2 Non wake-up Interrupt 1 Configuration Register (NWU_INT1_CON, R/W, Address = 0xE030_0704)

NWU_INT1_CON	Bit	Description	Reset Value
NWU_INT1_CON[n]	[n*4+2:n*4]	Sets the signaling method for NWU_INT1[n] (n=0,…,4) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.3 Non wake-up Interrupt 2 Configuration Register (NWU_INT2_CON, R/W, Address = 0xE030_0708)

Field	Bit	Description	Reset Value
NWU_INT2_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT2[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.4 Non wake-up Interrupt 3 Configuration Register (NWU_INT3_CON, R/W, Address = 0xE030_070C)

Field	Bit	Description	Reset Value
NWU_INT3_CON[4]	[n*4+2:n*4]	Sets the signaling method for NWU_INT3[n] (n=0,…,4) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.5 Non wake-up Interrupt 4 Configuration Register (NWU_INT4_CON, R/W, Address = 0xE030_0710)

Field	Bit	Description	Reset Value
NWU_INT4_CON[6]	[n*4+2:n*4]	Sets the signaling method for NWU_INT4[n] (n=0,…,6) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.6 Non wake-up Interrupt 5 Configuration Register (NWU_INT5_CON, R/W, Address = 0xE030_0714)

Field	Bit	Description	Reset Value
NWU_INT5_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT5[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.7 Non wake-up Interrupt 6 Configuration Register (NWU_INT6_CON, R/W, Address = 0xE030_0718)

Field	Bit	Description	Reset Value
NWU_INT6_CON[5]	[n*4+2:n*4]	Sets the signaling method for NWU_INT6[n] (n=0,…,5) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.8 Non wake-up Interrupt 7 Configuration Register (NWU_INT7_CON, R/W, Address = 0xE030_071C)

Field	Bit	Description	Reset Value
NWU_INT7_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT7[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.9 Non wake-up Interrupt 8 Configuration Register (NWU_INT8_CON, R/W, Address = 0xE030_0720)

Field	Bit	Description	Reset Value
NWU_INT8_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT8[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.10 Non wake-up Interrupt 9 Configuration Register (NWU_INT9_CON, R/W, Address = 0xE030_0724)

Field	Bit	Description	Reset Value
NWU_INT9_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT9[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.11 Non wake-up Interrupt 10 Configuration Register (NWU_INT10_CON, R/W, Address = 0xE030_0728)

Field	Bit	Description	Reset Value
NWU_INT10_CON[3]	[n*4+2:n*4]	Sets the signaling method for NWU_INT10[3] (n=0,…,3) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.12 Non wake-up Interrupt 11 Configuration Register (NWU_INT11_CON, R/W, Address = 0xE030_072C)

Field	Bit	Description	Reset Value
NWU_INT11_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT11[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.13 Non wake-up Interrupt 12 Configuration Register (NWU_INT12_CON, R/W, Address = 0xE030_0730)

Field	Bit	Description	Reset Value
NWU_INT12_CON[2]	[n*4+2:n*4]	Sets the signaling method for NWU_INT12[n] (n=0,…,2) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.14 Non wake-up Interrupt 13 Configuration Register (NWU_INT13_CON, R/W, Address = 0xE030_0734)

Field	Bit	Description	Reset Value
NWU_INT13_CON[6]	[n*4+2:n*4]	Sets the signaling method for NWU_INT13[n] (n=0,…,6) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.15 Non wake-up Interrupt 14 Configuration Register (NWU_INT14_CON, R/W, Address = 0xE030_0738)

Field	Bit	Description	Reset Value
NWU_INT14_CON[6]	[n*4+2:n*4]	Sets the signaling method for NWU_INT14[n] (n=0,…,6) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.16 Non wake-up Interrupt 15 Configuration Register (NWU_INT15_CON, R/W, Address = 0xE030_073C)

Field	Bit	Description	Reset Value
NWU_INT15_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT15[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.17 Non wake-up Interrupt 16 Configuration Register (NWU_INT16_CON, R/W, Address = 0xE030_0740)

Field	Bit	Description	Reset Value
NWU_INT16_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT16[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.18 Non wake-up Interrupt 17 Configuration Register (NWU_INT17_CON, R/W, Address = 0xE030_0744)

Field	Bit	Description	Reset Value
NWU_INT17_CON[4]	[n*4+2:n*4]	Sets the signaling method for NWU_INT17[n] (n=0,…,4) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.19 Non wake-up Interrupt 18 Configuration Register (NWU_INT18_CON, R/W, Address = 0xE030_0748)

Field	Bit	Description	Reset Value
NWU_INT18_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT18[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.20 Non wake-up Interrupt 19 Configuration Register (NWU_INT19_CON, R/W, Address = 0xE030_074C)

Field	Bit	Description	Reset Value
NWU_INT19_CON[7]	[n*4+2:n*4]	Sets the signaling method for NWU_INT19[n] (n=0,…,7) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.21 Non wake-up Interrupt 20 Configuration Register (NWU_INT20_CON, R/W, Address = 0xE030_0750)

Field	Bit	Description	Reset Value
NWU_INT20_CON[3]	[n*4+2:n*4]	Sets the signaling method for NWU_INT20[n] (n=0,…,3) 000 = Low level, 001 = High level, 010 = Falling edge, 011 = Rising edge, 100 = Both edge	000

5.5.22 Non wake-up Interrupt 0 Filter Configuration Register 0 (NWU_INT0_FLTCON0, R/W, Address = 0xE030_0800)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN0[3]	[31]	Digital Filter Enable for NWU_INT0[3]	0
NWU_INT0[3]	[30:24]	Filtering width for NWU_INT0[3]	000
FLTEN0[2]	[23]	Digital Filter Enable for NWU_INT0[2]	0
NWU_INT0[2]	[22:16]	Filtering width for NWU_INT0[2]	000
FLTEN0[1]	[15]	Digital Filter Enable for NWU_INT0[1]	0
NWU_INT0[1]	[14:8]	Filtering width for NWU_INT0[1]	000
FLTEN0[0]	[7]	Digital Filter Enable for NWU_INT0[0]	0
NWU_INT0[0]	[6:0]	Filtering width for NWU_INT0[0]	000

5.5.23 Non wake-up Interrupt 0 Filter Configuration Register 1 (NWU_INT0_FLTCON1, R/W, Address = 0xE030_0804)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN0[7]	[31]	Digital Filter Enable for NWU_INT0[7]	0
NWU_INT0[7]	[30:24]	Filtering width for NWU_INT0[7]	000
FLTEN0[6]	[23]	Digital Filter Enable for NWU_INT0[6]	0
NWU_INT0[6]	[22:16]	Filtering width for NWU_INT0[6]	000
FLTEN0[5]	[15]	Digital Filter Enable for NWU_INT0[5]	0
NWU_INT0[5]	[14:8]	Filtering width for NWU_INT0[5]	000
FLTEN0[4]	[7]	Digital Filter Enable for NWU_INT0[4]	0
NWU_INT0[4]	[6:0]	Filtering width for NWU_INT0[4]	000

5.5.24 Non wake-up Interrupt 1 Filter Configuration Register 0 (NWU_INT1_FLTCON0, R/W, Address = 0xE030_0808)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN1[3]	[31]	Digital Filter Enable for NWU_INT1[3]	0
NWU_INT1[3]	[30:24]	Filtering width for NWU_INT1[3]	000
FLTEN1[2]	[23]	Digital Filter Enable for NWU_INT1[2]	0
NWU_INT1[2]	[22:16]	Filtering width for NWU_INT1[2]	000
FLTEN1[1]	[15]	Digital Filter Enable for NWU_INT1[1]	0
NWU_INT1[1]	[14:8]	Filtering width for NWU_INT1[1]	000

FLTEN1[0]	[7]	Digital Filter Enable for NWU_INT1[0]	0
NWU_INT1[0]	[6:0]	Filtering width for NWU_INT1[0]	000

5.5.25 Non wake-up Interrupt 1 Filter Configuration Register 1 (NWU_INT1_FLTCON1, R/W, Address = 0xE030_080C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
FLTEN1[4]	[7]	Digital Filter Enable for NWU_INT1[4]	0
NWU_INT1[4]	[6:0]	Filtering width for NWU_INT1[4]	000

5.5.26 Non wake-up Interrupt 2 filter configuration register 0 (NWU_INT2_FLTCON0, R/W, Address = 0xE030_0810)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN2[3]	[31]	Digital Filter Enable for NWU_INT2[3]	0
NWU_INT2[3]	[30:24]	Filtering width for NWU_INT2[3]	000
FLTEN2[2]	[23]	Digital Filter Enable for NWU_INT2[2]	0
NWU_INT2[2]	[22:16]	Filtering width for NWU_INT2[2]	000
FLTEN2[1]	[15]	Digital Filter Enable for NWU_INT2[1]	0
NWU_INT2[1]	[14:8]	Filtering width for NWU_INT2[1]	000
FLTEN2[0]	[7]	Digital Filter Enable for NWU_INT2[0]	0
NWU_INT2[0]	[6:0]	Filtering width for NWU_INT2[0]	000

5.5.27 Non wake-up Interrupt 2 Filter Configuration Register 1 (NWU_INT2_FLTCON1, R/W, Address = 0xE030_0814)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN2[7]	[31]	Digital Filter Enable for NWU_INT2[7]	0
NWU_INT2[7]	[30:24]	Filtering width for NWU_INT2[7]	000
FLTEN2[6]	[23]	Digital Filter Enable for NWU_INT2[6]	0
NWU_INT2[6]	[22:16]	Filtering width for NWU_INT2[6]	000
FLTEN2[5]	[15]	Digital Filter Enable for NWU_INT2[5]	0
NWU_INT2[5]	[14:8]	Filtering width for NWU_INT2[5]	000
FLTEN2[4]	[7]	Digital Filter Enable for NWU_INT2[4]	0
NWU_INT2[4]	[6:0]	Filtering width for NWU_INT2[4]	000

5.5.28 Non wake-up Interrupt 3 Filter Configuration Register 0 (NWU_INT3_FLTCON0, R/W, Address = 0xE030_0818)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN3[3]	[31]	Digital Filter Enable for NWU_INT3[3]	0
NWU_INT3[3]	[30:24]	Filtering width for NWU_INT3[3]	000
FLTEN3[2]	[23]	Digital Filter Enable for NWU_INT3[2]	0
NWU_INT3[2]	[22:16]	Filtering width for NWU_INT3[2]	000
FLTEN3[1]	[15]	Digital Filter Enable for NWU_INT3[1]	0
NWU_INT3[1]	[14:8]	Filtering width for NWU_INT3[1]	000
FLTEN3[0]	[7]	Digital Filter Enable for NWU_INT3[0]	0
NWU_INT3[0]	[6:0]	Filtering width for NWU_INT3[0]	000

5.5.29 Non wake-up Interrupt 3 Filter Configuration Register 1 (NWU_INT3_FLTCON1, R/W, Address = 0xE030_081C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
FLTEN3[4]	[7]	Digital Filter Enable for NWU_INT3[4]	0
NWU_INT3[4]	[6:0]	Filtering width for NWU_INT3[4]	000

5.5.30 Non wake-up Interrupt 4 Filter Configuration Register 0 (NWU_INT4_FLTCON0, R/W, Address = 0xE030_0820)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN4[3]	[31]	Digital Filter Enable for NWU_INT4[3]	0
NWU_INT4[3]	[30:24]	Filtering width for NWU_INT4[3]	000
FLTEN4[2]	[23]	Digital Filter Enable for NWU_INT4[2]	0
NWU_INT4[2]	[22:16]	Filtering width for NWU_INT4[2]	000
FLTEN4[1]	[15]	Digital Filter Enable for NWU_INT4[1]	0
NWU_INT4[1]	[14:8]	Filtering width for NWU_INT4[1]	000
FLTEN4[0]	[7]	Digital Filter Enable for NWU_INT4[0]	0
NWU_INT4[0]	[6:0]	Filtering width for NWU_INT4[0]	000

5.5.31 Non wake-up Interrupt 4 Filter Configuration Register 1 (NWU_INT4_FLTCON1, R/W, Address = 0xE030_0824)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	000
FLTEN4[6]	[23]	Digital Filter Enable for NWU_INT4[6]	0
NWU_INT4[6]	[22:16]	Filtering width for NWU_INT4[6]	000
FLTEN4[5]	[15]	Digital Filter Enable for NWU_INT4[5]	0
NWU_INT4[5]	[14:8]	Filtering width for NWU_INT4[5]	000
FLTEN4[4]	[7]	Digital Filter Enable for NWU_INT4[4]	0
NWU_INT4[4]	[6:0]	Filtering width for NWU_INT4[4]	000

5.5.32 Non wake-up Interrupt 5 Filter Configuration Register 0 (NWU_INT5_FLTCON0, R/W, Address = 0xE030_0828)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN5[3]	[31]	Digital Filter Enable for NWU_INT5[3]	0
NWU_INT5[3]	[30:24]	Filtering width for NWU_INT5[3]	000
FLTEN5[2]	[23]	Digital Filter Enable for NWU_INT5[2]	0
NWU_INT5[2]	[22:16]	Filtering width for NWU_INT5[2]	000
FLTEN5[1]	[15]	Digital Filter Enable for NWU_INT5[1]	0
NWU_INT5[1]	[14:8]	Filtering width for NWU_INT5[1]	000
FLTEN5[0]	[7]	Digital Filter Enable for NWU_INT5[0]	0
NWU_INT5[0]	[6:0]	Filtering width for NWU_INT5[0]	000

5.5.33 Non wake-up Interrupt 5 Filter Configuration Register 1 (NWU_INT5_FLTCON1, R/W, 0xE030_082C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN5[7]	[31]	Digital Filter Enable for NWU_INT5[7]	0
NWU_INT5[7]	[30:24]	Filtering width for NWU_INT5[7]	000
FLTEN5[6]	[23]	Digital Filter Enable for NWU_INT5[6]	0
NWU_INT5[6]	[22:16]	Filtering width for NWU_INT5[6]	000
FLTEN5[5]	[15]	Digital Filter Enable for NWU_INT5[5]	0
NWU_INT5[5]	[13:8]	Filtering width for NWU_INT5[5]	000
FLTEN5[4]	[7]	Digital Filter Enable for NWU_INT5[4]	0
NWU_INT5[4]	[5:0]	Filtering width for NWU_INT5[4]	000

5.5.34 Non wake-up Interrupt 6 Filter Configuration Register 0 (NWU_INT6_FLTCON0, R/W, Address = 0xE030_0830)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN6[3]	[31]	Digital Filter Enable for NWU_INT6[3]	0
NWU_INT6[3]	[30:24]	Filtering width for NWU_INT6[3]	000
FLTEN6[2]	[23]	Digital Filter Enable for NWU_INT6[2]	0
NWU_INT6[2]	[22:16]	Filtering width for NWU_INT6[2]	000
FLTEN6[1]	[15]	Digital Filter Enable for NWU_INT6[1]	0
NWU_INT6[1]	[14:8]	Filtering width for NWU_INT6[1]	000
FLTEN6[0]	[7]	Digital Filter Enable for NWU_INT6[0]	0
NWU_INT6[0]	[6:0]	Filtering width for NWU_INT6[0]	000

5.5.35 Non wake-up Interrupt 6 Filter Configuration Register 1 (NWU_INT6_FLTCON1, R/W, Address = 0xE030_0834)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	000
FLTEN6[5]	[15]	Digital Filter Enable for NWU_INT6[5]	0
NWU_INT6[5]	[14:8]	Filtering width for NWU_INT6[5]	000
FLTEN6[4]	[7]	Digital Filter Enable for NWU_INT6[4]	0
NWU_INT6[4]	[6:0]	Filtering width for NWU_INT6[4]	000

5.5.36 Non wake-up Interrupt 7 Filter Configuration Register 0 (NWU_INT7_FLTCON0, R/W, Address = 0xE030_0838)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN7[3]	[31]	Digital Filter Enable for NWU_INT7[3]	0
NWU_INT7[3]	[30:24]	Filtering width for NWU_INT7[3]	000
FLTEN7[2]	[23]	Digital Filter Enable for NWU_INT7[2]	0
NWU_INT7[2]	[22:16]	Filtering width for NWU_INT7[2]	000
FLTEN7[1]	[15]	Digital Filter Enable for NWU_INT7[1]	0
NWU_INT7[1]	[14:8]	Filtering width for NWU_INT7[1]	000
FLTEN7[0]	[7]	Digital Filter Enable for NWU_INT7[0]	0
NWU_INT7[0]	[6:0]	Filtering width for NWU_INT7[0]	000

5.5.37 Non wake-up Interrupt 7 Filter Configuration Register 1 (NWU_INT7_FLTCON1, R/W, Address = 0xE030_083C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN7[7]	[31]	Digital Filter Enable for NWU_INT7[7]	0
NWU_INT7[7]	[30:24]	Filtering width for NWU_INT7[7]	000
FLTEN7[6]	[23]	Digital Filter Enable for NWU_INT7[6]	0
NWU_INT7[6]	[22:16]	Filtering width for NWU_INT7[6]	000
FLTEN7[5]	[15]	Digital Filter Enable for NWU_INT7[5]	0
NWU_INT7[5]	[14:8]	Filtering width for NWU_INT7[5]	000
FLTEN7[4]	[7]	Digital Filter Enable for NWU_INT7[4]	0
NWU_INT7[4]	[6:0]	Filtering width for NWU_INT7[4]	000

5.5.38 Non wake-up Interrupt 8 Filter Configuration Register 0 (NWU_INT8_FLTCON0, R/W, Address = 0xE030_0840)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN8[3]	[31]	Digital Filter Enable for NWU_INT8[3]	0
NWU_INT8[3]	[30:24]	Filtering width for NWU_INT8[3]	000
FLTEN8[2]	[23]	Digital Filter Enable for NWU_INT8[2]	0
NWU_INT8[2]	[22:16]	Filtering width for NWU_INT8[2]	000
FLTEN8[1]	[15]	Digital Filter Enable for NWU_INT8[1]	0
NWU_INT8[1]	[14:8]	Filtering width for NWU_INT8[1]	000
FLTEN8[0]	[7]	Digital Filter Enable for NWU_INT8[0]	0
NWU_INT8[0]	[6:0]	Filtering width for NWU_INT8[0]	000

5.5.39 Non wake-up Interrupt 8 Filter Configuration Register 1 (NWU_INT8_FLTCON1, R/W, Address = 0xE030_0844)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN8[7]	[31]	Digital Filter Enable for NWU_INT8[7]	0
NWU_INT8[7]	[30:24]	Filtering width for NWU_INT8[7]	000
FLTEN8[6]	[23]	Digital Filter Enable for NWU_INT8[6]	0
NWU_INT8[6]	[22:16]	Filtering width for NWU_INT8[6]	000
FLTEN8[5]	[15]	Digital Filter Enable for NWU_INT8[5]	0
NWU_INT8[5]	[14:8]	Filtering width for NWU_INT8[5]	000

FLTEN8[4]	[7]	Digital Filter Enable for NWU_INT8[4]	0
NWU_INT8[4]	[6:0]	Filtering width for NWU_INT8[4]	000

5.5.40 Non wake-up Interrupt 9 Filter Configuration Register 0 (NWU_INT9_FLTCON0, R/W, Address = 0xE030_0848)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN9[3]	[31]	Digital Filter Enable for NWU_INT9[3]	0
NWU_INT9[3]	[30:24]	Filtering width for NWU_INT9[3]	000
FLTEN9[2]	[23]	Digital Filter Enable for NWU_INT9[2]	0
NWU_INT9[2]	[22:16]	Filtering width for NWU_INT9[2]	000
FLTEN9[1]	[15]	Digital Filter Enable for NWU_INT9[1]	0
NWU_INT9[1]	[14:8]	Filtering width for NWU_INT9[1]	000
FLTEN9[0]	[7]	Digital Filter Enable for NWU_INT9[0]	0
NWU_INT9[0]	[6:0]	Filtering width for NWU_INT9[0]	000

5.5.41 Non wake-up Interrupt 9 Filter Configuration Register 1 (NWU_INT9_FLTCON1, R/W, Address = 0xE030_084C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN9[7]	[31]	Digital Filter Enable for NWU_INT9[7]	0
NWU_INT9[7]	[30:24]	Filtering width for NWU_INT9[7]	000
FLTEN9[6]	[23]	Digital Filter Enable for NWU_INT9[6]	0
NWU_INT9[6]	[22:16]	Filtering width for NWU_INT9[6]	000
FLTEN9[5]	[15]	Digital Filter Enable for NWU_INT9[5]	0
NWU_INT9[5]	[14:8]	Filtering width for NWU_INT9[5]	000
FLTEN9[4]	[7]	Digital Filter Enable for NWU_INT9[4]	0
NWU_INT9[4]	[56:0]	Filtering width for NWU_INT9[4]	000

5.5.42 Non wake-up Interrupt 10 Filter Configuration Register 0 (NWU_INT10_FLTCON0, R/W, Address = 0xE030_0850)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN10[3]	[31]	Digital Filter Enable for NWU_INT10[3]	0
NWU_INT10[3]	[30:24]	Filtering width for NWU_INT10[3]	000
FLTEN10[2]	[23]	Digital Filter Enable for NWU_INT10[2]	0

NWU_INT10[2]	[22:16]	Filtering width for NWU_INT10[2]	000
FLTEN10[1]	[15]	Digital Filter Enable for NWU_INT10[1]	0
NWU_INT10[1]	[14:8]	Filtering width for NWU_INT10[1]	000
FLTEN10[0]	[7]	Digital Filter Enable for NWU_INT10[0]	0
NWU_INT10[0]	[6:0]	Filtering width for NWU_INT10[0]	000

5.5.43 Non wake-up Interrupt 11 Filter Configuration Register 0 (NWU_INT11_FLTCON0, R/W, Address = 0xE030_0858)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN11[3]	[31]	Digital Filter Enable for NWU_INT11[3]	0
NWU_INT11[3]	[30:24]	Filtering width for NWU_INT11[3]	000
FLTEN11[2]	[23]	Digital Filter Enable for NWU_INT11[2]	0
NWU_INT11[2]	[22:16]	Filtering width for NWU_INT11[2]	000
FLTEN11[1]	[15]	Digital Filter Enable for NWU_INT11[1]	0
NWU_INT11[1]	[14:8]	Filtering width for NWU_INT11[1]	000
FLTEN11[0]	[7]	Digital Filter Enable for NWU_INT11[0]	0
NWU_INT11[0]	[6:0]	Filtering width for NWU_INT11[0]	000

5.5.44 Non wake-up Interrupt 11 Filter Configuration Register 1 (NWU_INT11_FLTCON1, R/W, Address = 0xE030_085C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN11[7]	[31]	Digital Filter Enable for NWU_INT11[7]	0
NWU_INT11[7]	[30:24]	Filtering width for NWU_INT11[7]	000
FLTEN11[6]	[23]	Digital Filter Enable for NWU_INT11[6]	0
NWU_INT11[6]	[22:16]	Filtering width for NWU_INT11[6]	000
FLTEN11[5]	[15]	Digital Filter Enable for NWU_INT11[5]	0
NWU_INT11[5]	[14:8]	Filtering width for NWU_INT11[5]	000
FLTEN11[4]	[7]	Digital Filter Enable for NWU_INT11[4]	0
NWU_INT11[4]	[6:0]	Filtering width for NWU_INT11[4]	000

5.5.45 Non wake-up Interrupt 12 Filter Configuration Register 0 (NWU_INT12_FLTCON0, R/W, Address = 0xE030_0860)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	000
FLTEN12[2]	[23]	Digital Filter Enable for NWU_INT12[2]	0
NWU_INT12[2]	[22:16]	Filtering width for NWU_INT12[2]	000
FLTEN12[1]	[15]	Digital Filter Enable for NWU_INT12[1]	0
NWU_INT12[1]	[14:8]	Filtering width for NWU_INT12[1]	000
FLTEN12[0]	[7]	Digital Filter Enable for NWU_INT12[0]	0
NWU_INT12[0]	[6:0]	Filtering width for NWU_INT12[0]	000

5.5.46 Non wake-up Interrupt 13 Filter Configuration Register 0 (NWU_INT13_FLTCON0, R/W, Address = 0xE030_0868)

Field	Bit	Description	Reset Value
FLTEN13[3]	[31]	Digital Filter Enable for NWU_INT13[3]	0
NWU_INT13[3]	[30:24]	Filtering width for NWU_INT13[3]	000
FLTEN13[2]	[23]	Digital Filter Enable for NWU_INT13[2]	0
NWU_INT13[2]	[22:16]	Filtering width for NWU_INT13[2]	000
FLTEN13[1]	[15]	Digital Filter Enable for NWU_INT13[1]	0
NWU_INT13[1]	[14:8]	Filtering width for NWU_INT13[1]	000
FLTEN13[0]	[7]	Digital Filter Enable for NWU_INT13[0]	0
NWU_INT13[0]	[6:0]	Filtering width for NWU_INT13[0]	000

5.5.47 Non wake-up Interrupt 13 Filter Configuration Register 1 (NWU_INT13_FLTCON1, R/W, Address = 0xE030_086C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	000
FLTEN13[6]	[23]	Digital Filter Enable for NWU_INT13[6]	0
NWU_INT13[6]	[22:16]	Filtering width for NWU_INT13[6]	000
FLTEN13[5]	[15]	Digital Filter Enable for NWU_INT13[5]	0
NWU_INT13[5]	[14:8]	Filtering width for NWU_INT13[5]	000
FLTEN13[4]	[7]	Digital Filter Enable for NWU_INT13[4]	0
NWU_INT13[4]	[6:0]	Filtering width for NWU_INT13[4]	000

5.5.48 Non wake-up Interrupt 14 Filter Configuration Register 0 (NWU_INT14_FLTCON0, R/W, Address = 0xE030_0870)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN14[3]	[31]	Digital Filter Enable for NWU_INT14[3]	0
NWU_INT14[3]	[30:24]	Filtering width for NWU_INT14[3]	000
FLTEN14[2]	[23]	Digital Filter Enable for NWU_INT14[2]	0
NWU_INT14[2]	[22:16]	Filtering width for NWU_INT14[2]	000
FLTEN14[1]	[15]	Digital Filter Enable for NWU_INT14[1]	0
NWU_INT14[1]	[14:8]	Filtering width for NWU_INT14[1]	000
FLTEN14[0]	[7]	Digital Filter Enable for NWU_INT14[0]	0
NWU_INT14[0]	[6:0]	Filtering width for NWU_INT14[0]	000

5.5.49 Non wake-up Interrupt 14 Filter Configuration Register 1 (NWU_INT14_FLTCON1, R/W, Address = 0xE030_0874)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	000
FLTEN14[6]	[23]	Digital Filter Enable for NWU_INT14[6]	0
NWU_INT14[6]	[22:16]	Filtering width for NWU_INT14[6]	000
FLTEN14[5]	[15]	Digital Filter Enable for NWU_INT14[5]	0
NWU_INT14[5]	[14:8]	Filtering width for NWU_INT14[5]	000
FLTEN14[4]	[7]	Digital Filter Enable for NWU_INT14[4]	0
NWU_INT14[4]	[6:0]	Filtering width for NWU_INT14[4]	000

5.5.50 Non wake-up Interrupt 15 Filter Configuration Register 0 (NWU_INT15_FLTCON0, R/W, Address = 0xE030_0878)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN15[3]	[31]	Digital Filter Enable for NWU_INT15[3]	0
NWU_INT15[3]	[30:24]	Filtering width for NWU_INT15[3]	000
FLTEN15[2]	[23]	Digital Filter Enable for NWU_INT15[2]	0
NWU_INT15[2]	[22:16]	Filtering width for NWU_INT15[2]	000
FLTEN15[1]	[15]	Digital Filter Enable for NWU_INT15[1]	0
NWU_INT15[1]	[14:8]	Filtering width for NWU_INT15[1]	000
FLTEN15[0]	[7]	Digital Filter Enable for NWU_INT15[0]	0
NWU_INT15[0]	[6:0]	Filtering width for NWU_INT15[0]	000

5.5.51 Non wake-up Interrupt 15 Filter Configuration Register 1 (NWU_INT15_FLTCON1, R/W, Address = 0xE030_087C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN15[7]	[31]	Digital Filter Enable for NWU_INT15[7]	0
NWU_INT15[7]	[30:24]	Filtering width for NWU_INT15[7]	000
FLTEN15[6]	[23]	Digital Filter Enable for NWU_INT15[6]	0
NWU_INT15[6]	[22:16]	Filtering width for NWU_INT15[6]	000
FLTEN15[5]	[15]	Digital Filter Enable for NWU_INT15[5]	0
NWU_INT15[5]	[14:8]	Filtering width for NWU_INT15[5]	000
FLTEN15[4]	[7]	Digital Filter Enable for NWU_INT15[4]	0
NWU_INT15[4]	[6:0]	Filtering width for NWU_INT15[4]	000

5.5.52 Non wake-up Interrupt 16 Filter Configuration Register 0 (NWU_INT16_FLTCON0, R/W, Address = 0xE030_0880)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN16[3]	[31]	Digital Filter Enable for NWU_INT16[3]	0
NWU_INT16[3]	[30:24]	Filtering width for NWU_INT16[3]	000
FLTEN16[2]	[23]	Digital Filter Enable for NWU_INT16[2]	0
NWU_INT16[2]	[22:16]	Filtering width for NWU_INT16[2]	000
FLTEN16[1]	[15]	Digital Filter Enable for NWU_INT16[1]	0
NWU_INT16[1]	[14:8]	Filtering width for NWU_INT16[1]	000
FLTEN16[0]	[7]	Digital Filter Enable for NWU_INT16[0]	0
NWU_INT16[0]	[6:0]	Filtering width for NWU_INT16[0]	000

5.5.53 Non wake-up Interrupt 16 Filter Configuration Register 1 (NWU_INT16_FLTCON1, R/W, Address = 0xE030_0884)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN16[7]	[31]	Digital Filter Enable for NWU_INT16[7]	0
NWU_INT16[7]	[30:24]	Filtering width for NWU_INT16[7]	000
FLTEN16[6]	[23]	Digital Filter Enable for NWU_INT16[6]	0
NWU_INT16[6]	[22:16]	Filtering width for NWU_INT16[6]	000
FLTEN16[5]	[15]	Digital Filter Enable for NWU_INT16[5]	0
NWU_INT16[5]	[14:8]	Filtering width for NWU_INT16[5]	000

FLTEN16[4]	[7]	Digital Filter Enable for NWU_INT16[4]	0
NWU_INT16[4]	[6:0]	Filtering width for NWU_INT16[4]	000

5.5.54 Non wake-up Interrupt 17 Filter Configuration Register 0 (NWU_INT17_FLTCON0, R/W, Address = 0xE030_0888)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN17[3]	[31]	Digital Filter Enable for NWU_INT17[3]	0
NWU_INT17[3]	[30:24]	Filtering width for NWU_INT17[3]	000
FLTEN17[2]	[23]	Digital Filter Enable for NWU_INT17[2]	0
NWU_INT17[2]	[22:16]	Filtering width for NWU_INT17[2]	000
FLTEN17[1]	[15]	Digital Filter Enable for NWU_INT17[1]	0
NWU_INT17[1]	[14:8]	Filtering width for NWU_INT17[1]	000
FLTEN17[0]	[7]	Digital Filter Enable for NWU_INT17[0]	0
NWU_INT17[0]	[6:0]	Filtering width for NWU_INT17[0]	000

5.5.55 Non wake-up Interrupt 17 Filter Configuration Register 1 (NWU_INT17_FLTCON1, R/W, Address = 0xE030_088C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	000
FLTEN17[4]	[7]	Digital Filter Enable for NWU_INT17[4]	0
NWU_INT17[4]	[6:0]	Filtering width for NWU_INT17[4]	000

5.5.56 Non wake-up Interrupt 18 Filter Configuration Register 0 (NWU_INT18_FLTCON0, R/W, Address = 0xE030_0890)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN18[3]	[31]	Digital Filter Enable for NWU_INT18[3]	0
NWU_INT18[3]	[30:24]	Filtering width for NWU_INT18[3]	000
FLTEN18[2]	[23]	Digital Filter Enable for NWU_INT18[2]	0
NWU_INT18[2]	[22:16]	Filtering width for NWU_INT18[2]	000
FLTEN18[1]	[15]	Digital Filter Enable for NWU_INT18[1]	0
NWU_INT18[1]	[14:8]	Filtering width for NWU_INT18[1]	000
FLTEN18[0]	[7]	Digital Filter Enable for NWU_INT18[0]	0
NWU_INT18[0]	[6:0]	Filtering width for NWU_INT18[0]	000

5.5.57 Non wake-up Interrupt 18 Filter Configuration Register 1 (NWU_INT18_FLTCON1, R/W, Address = 0xE030_0894)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN18[7]	[31]	Digital Filter Enable for NWU_INT18[7]	0
NWU_INT18[7]	[30:24]	Filtering width for NWU_INT18[7]	000
FLTEN18[6]	[23]	Digital Filter Enable for NWU_INT18[6]	0
NWU_INT18[6]	[22:16]	Filtering width for NWU_INT18[6]	000
FLTEN18[5]	[15]	Digital Filter Enable for NWU_INT18[5]	0
NWU_INT18[5]	[14:8]	Filtering width for NWU_INT18[5]	000
FLTEN18[4]	[7]	Digital Filter Enable for NWU_INT18[4]	0
NWU_INT18[4]	[6:0]	Filtering width for NWU_INT18[4]	000

5.5.58 Non wake-up Interrupt 19 Filter Configuration Register 0 (NWU_INT19_FLTCON0, R/W, Address = 0xE030_0898)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN19[3]	[31]	Digital Filter Enable for NWU_INT19[3]	0
NWU_INT19[3]	[30:24]	Filtering width for NWU_INT19[3]	000
FLTEN19[2]	[23]	Digital Filter Enable for NWU_INT19[2]	0
NWU_INT19[2]	[22:16]	Filtering width for NWU_INT19[2]	000
FLTEN19[1]	[15]	Digital Filter Enable for NWU_INT19[1]	0
NWU_INT19[1]	[14:8]	Filtering width for NWU_INT19[1]	000
FLTEN19[0]	[7]	Digital Filter Enable for NWU_INT19[0]	0
NWU_INT19[0]	[6:0]	Filtering width for NWU_INT19[0]	000

5.5.59 Non wake-up Interrupt 19 Filter Configuration Register 1 (NWU_INT19_FLTCON1, R/W, Address = 0xE030_089C)

The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN19[7]	[31]	Digital Filter Enable for NWU_INT19[7]	0
NWU_INT19[7]	[30:24]	Filtering width for NWU_INT19[7]	000
FLTEN19[6]	[23]	Digital Filter Enable for NWU_INT19[6]	0
NWU_INT19[6]	[22:16]	Filtering width for NWU_INT19[6]	000
FLTEN19[5]	[15]	Digital Filter Enable for NWU_INT19[5]	0

NWU_INT19[5]	[14:8]	Filtering width for NWU_INT19[5]	000
FLTEN19[4]	[7]	Digital Filter Enable for NWU_INT19[4]	0
NWU_INT19[4]	[6:0]	Filtering width for NWU_INT19[4]	000

5.5.60 Non wake-up Interrupt 20 Filter Configuration Register 0 (NWU_INT20_FLTCON0, R/W, Address = 0xE030_08A0) The Digital Filter Enable field means that 0 = disables, 1 = enables.

Field	Bit	Description	Reset Value
FLTEN20[3]	[31]	Digital Filter Enable for NWU_INT20[3]	0
NWU_INT20[3]	[30:24]	Filtering width for NWU_INT20[3]	000
FLTEN20[2]	[23]	Digital Filter Enable for NWU_INT20[2]	0
NWU_INT20[2]	[22:16]	Filtering width for NWU_INT20[2]	000
FLTEN20[1]	[15]	Digital Filter Enable for NWU_INT20[1]	0
NWU_INT20[1]	[14:8]	Filtering width for NWU_INT20[1]	000
FLTEN20[0]	[7]	Digital Filter Enable for NWU_INT20[0]	0
NWU_INT20[0]	[6:0]	Filtering width for NWU_INT20[0]	000

5.5.61 Non wake-up Interrupt 0 Mask Register (NWU_INT0_MASK, R/W, Address = 0xE030_0900)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT0_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.62 Non wake-up Interrupt 1 Mask Register (NWU_INT1_MASK, R/W, Address = 0xE030_0904)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
NWU_INT1_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,4)	1

5.5.63 Non wake-up Interrupt 2 Mask Register (NWU_INT2_MASK, R/W, Address = 0xE030_0908)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT2_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.64 Non wake-up Interrupt 3 Mask Register (NWU_INT3_MASK, R/W, Address = 0xE030_090C)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
NWU_INT3_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,4)	1

5.5.65 Non wake-up Interrupt 4 Mask Register (NWU_INT4_MASK, R/W, Address = 0xE030_0910)

Field	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
NWU_INT4_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,6)	1

5.5.66 Non wake-up Interrupt 5 Mask Register (NWU_INT5_MASK, R/W, Address = 0xE030_0914)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT5_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.67 Non wake-up Interrupt 6 Mask Register (NWU_INT6_MASK, R/W, Address = 0xE030_0918)

Field	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0
NWU_INT6_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,5)	1

5.5.68 Non wake-up Interrupt 7 Mask Register (NWU_INT7_MASK, R/W, Address = 0xE030_091C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT7_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.69 Non wake-up Interrupt 8 Mask Register (NWU_INT8_MASK, R/W, Address = 0xE030_0920)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT8_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.70 Non wake-up Interrupt 9 Mask Register (NWU_INT9_MASK, R/W, Address = 0xE030_0924)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT9_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.71 Non wake-up Interrupt 10 Mask Register (NWU_INT10_MASK, R/W, Address = 0xE030_0928)

Field	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
NWU_INT10_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,3)	1

5.5.72 Non wake-up Interrupt 11 Mask Register (NWU_INT11_MASK, R/W, Address = 0xE030_092C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT11_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.73 Non wake-up Interrupt 12 Mask Register (NWU_INT12_MASK, R/W, Address = 0xE030_0930)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
NWU_INT12_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,2)	1

5.5.74 Non wake-up Interrupt 13 Mask Register (NWU_INT13_MASK, R/W, Address = 0xE030_0934)

Field	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
NWU_INT13_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,6)	1

5.5.75 Non wake-up Interrupt 14 Mask Register (NWU_INT14_MASK, R/W, Address = 0xE030_0938)

Field	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
NWU_INT14_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,6)	1

5.5.76 Non wake-up Interrupt 15 Mask Register (NWU_INT15_MASK, R/W, Address = 0xE030_093C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT15_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.77 Non wake-up Interrupt 16 Mask Register (NWU_INT16_MASK, R/W, Address = 0xE030_0940)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT16_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.78 Non wake-up Interrupt 17 Mask Register (NWU_INT17_MASK, R/W, Address = 0xE030_0944)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
NWU_INT17_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,4)	1

5.5.79 Non wake-up Interrupt 18Mask Register (NWU_INT18_MASK, R/W, Address = 0xE030_0948)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT18_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.80 Non wake-up Interrupt 19 Mask Register (NWU_INT19_MASK, R/W, Address = 0xE030_094C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT19_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,7)	1

5.5.81 Non wake-up Interrupt 20 Mask Register (NWU_INT20_MASK, R/W, Address = 0xE030_0950)

Field	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
NWU_INT20_MASK[n]	[n]	0 = Enabled, 1 = Masked (n=0,..,3)	1

5.5.82 Non wake-up Interrupt 0 Pending Register (NWU_INT0_PEND, R/W, Address = 0xE030_0A00)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT0_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.83

5.5.84 Non wake-up Interrupt 1 Pending Register (NWU_INT1_PEND, R/W, Address = 0xE030_0A04)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
NWU_INT1_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,4)	0

5.5.85 Non wake-up Interrupt 2 Pending Register (NWU_INT2_PEND, R/W, Address = 0xE030_0A08)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT2_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.86 Non wake-up Interrupt 3 Pending Register (NWU_INT3_PEND, R/W, Address = 0xE030_0A0C)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
NWU_INT3_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,4)	0

5.5.87 Non wake-up Interrupt 4 Pending Register (NWU_INT4_PEND, R/W, Address = 0xE030_0A10)

Field	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
NWU_INT4_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,6)	0

5.5.88 Non wake-up Interrupt 5 Pending Register (NWU_INT5_PEND, R/W, Address = 0xE030_0A14)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT5_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.89 Non wake-up Interrupt 6 Pending Register (NWU_INT6_PEND, R/W, Address = 0xE030_0A18)

Field	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0
NWU_INT6_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,5)	0

5.5.90 Non wake-up Interrupt 7 Pending Register (NWU_INT7_PEND, R/W, Address = 0xE030_0A1C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT7_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.91 Non wake-up Interrupt 8 Pending Register (NWU_INT8_PEND, R/W, Address = 0xE030_0A20)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT8_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.92 Non wake-up Interrupt 9 Pending Register (NWU_INT9_PEND, R/W, Address = 0xE030_0A24)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT9_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.93 Non wake-up Interrupt 10 Pending Register (NWU_INT10_PEND, R/W, Address = 0xE030_0A28)

Field	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
NWU_INT10_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,3)	0

5.5.94 Non wake-up Interrupt 11 Pending Register (NWU_INT11_PEND, R/W, Address = 0xE030_0A2C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT11_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.95 Non wake-up Interrupt 12 Pending Register (NWU_INT12_PEND, R/W, Address = 0xE030_0A30)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
NWU_INT12_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,2)	0

5.5.96 Non wake-up Interrupt 13 Pending Register (NWU_INT13_PEND, R/W, Address = 0xE030_0A34)

Field	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
NWU_INT13_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,6)	0

5.5.97 Non wake-up Interrupt 14 Pending Register (NWU_INT14_PEND, R/W, Address = 0xE030_0A38)

Field	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
NWU_INT14_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,6)	0

5.5.98 Non wake-up Interrupt 15 Pending Register (NWU_INT15_PEND, R/W, Address = 0xE030_0A3C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT15_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.99 Non wake-up Interrupt 16 Pending Register (NWU_INT16_PEND, R/W, Address = 0xE030_0A40)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT16_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.100 Non wake-up Interrupt 17 Pending Register (NWU_INT17_PEND, R/W, Address = 0xE030_0A44)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
NWU_INT17_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.101 Non wake-up Interrupt 18 Pending Register (NWU_INT18_PEND, R/W, Address = 0xE030_0A48)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT18_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.102 Non wake-up Interrupt 19 Pending Register (NWU_INT19_PEND, R/W, Address = 0xE030_0A4C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
NWU_INT19_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,7)	0

5.5.103 Non wake-up Interrupt 20 Pending Register (NWU_INT20_PEND, R/W, Address = 0xE030_0A50)

Field	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
NWU_INT20_PEND[n]	[n]	1 = Interrupt occurred (n=0,…,3)	0

5.5.104 Non wake-up Interrupt Group Priority Control Register (NWU_INT_GRPPRI, R/W, Address = 0xE030_0B00)

Field	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
NWU_INT_GRPPRI	[0]	NWU_INT groups priority rotate enable 0: Does not rotate (Fixed) 1: Enables rotate	0

5.5.105 Non wake-up Interrupt Priority Control Register (NWU_INT_PRIORITY, R/W, Address = 0xE030_0B04)

0: Does not rotate(Fixed), 1: Enables rotate

Field	Bit	Description	Reset Value
Reserved	[31:21]	Reserved	0
NWU_INT20_PRI	[20]	NWU_INT group 20 priority rotate enable	0
NWU_INT19_PRI	[19]	NWU_INT group 19 priority rotate enable	0
NWU_INT18_PRI	[18]	NWU_INT group 18 priority rotate enable	0
NWU_INT17_PRI	[17]	NWU_INT group 17 priority rotate enable	0
NWU_INT16_PRI	[16]	NWU_INT group 16 priority rotate enable	0
NWU_INT15_PRI	[15]	NWU_INT group 15 priority rotate enable	0
NWU_INT14_PRI	[14]	NWU_INT group 14 priority rotate enable	0
NWU_INT13_PRI	[13]	NWU_INT group 13 priority rotate enable	0
NWU_INT12_PRI	[12]	NWU_INT group 12 priority rotate enable	0
NWU_INT11_PRI	[11]	NWU_INT group 11 priority rotate enable	0
NWU_INT10_PRI	[10]	NWU_INT group 10 priority rotate enable	0
NWU_INT9_PRI	[9]	NWU_INT group 9 priority rotate enable	0
NWU_INT8_PRI	[8]	NWU_INT group 8 priority rotate enable	0
NWU_INT7_PRI	[7]	NWU_INT group 7 priority rotate enable	0
NWU_INT6_PRI	[6]	NWU_INT group 6 priority rotate enable	0
NWU_INT5_PRI	[5]	NWU_INT group 5 priority rotate enable	0
NWU_INT4_PRI	[4]	NWU_INT group 4 priority rotate enable	0
NWU_INT3_PRI	[3]	NWU_INT group 3 priority rotate enable	0
NWU_INT2_PRI	[2]	NWU_INT group 2 priority rotate enable	0
NWU_INT1_PRI	[1]	NWU_INT group 1 priority rotate enable	0
NWU_INT0_PRI	[0]	NWU_INT group 0 priority rotate enable	0

5.5.106 Current Service Register (NWU_INT_SERVICE, R, Address = 0xE030_0B08)

Current Service Register presents which interrupt should be serviced. The bit value describes the group number and interrupts number. This value is decided by the PRIORITY register and valid if nIRQ is generated.

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
SVC_Group_Num	[7:3]	NWU_INT Service group number Group Num / Group Index 1 / NWU_EINT0 2 / NWU_EINT9 3 / NWU_EINT10 4 / NWU_EINT11	0

		5	NWU_EINT12		
		6	NWU_EINT13		
		7	NWU_EINT14		
		8	NWU_EINT1		
		9	NWU_EINT15		
		10	NWU_EINT16		
		11	NWU_EINT17		
		12	NWU_EINT18		
		13	NWU_EINT19		
		14	NWU_EINT20		
		15	NWU_EINT2		
		16	NWU_EINT3		
		17	NWU_EINT4		
		18	NWU_EINT5		
		19	NWU_EINT6		
		20	NWU_EINT7		
		21	NWU_EINT8		
SVC_Num	[2:0]	Interrupt number to be serviced			0

5.5.107 Non wake-up Interrupt Group Fixed Priority Control Register (NWU_INT_GRPFIXPRI, R/W, Address = 0xE030_0B10)

Field	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
Highest_GRP_NUM	[4:0]	Group number of the highest priority if fixed group priority mode: 1~25	0

5.5.108 Non wake-up Interrupt 0 Fixed Priority Control Register (NWU_INT0_FIXPRI, R/W, Address = 0xE030_0B14)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 0 if fixed priority mode: 0~7	0

5.5.109 Non wake-up Interrupt 1 Fixed Priority Control Register (NWU_INT1_FIXPRI, R/W, Address = 0xE030_0B18)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0

Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 1 if fixed priority mode: 0~7	0

5.5.110 Non wake-up Interrupt 2 Fixed Priority Control Register (NWU_INT2_FIXPRI, R/W, Address = 0xE030_0B1C)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 2 if fixed priority mode: 0~7	0

5.5.111 Non wake-up Interrupt 3 Fixed Priority Control Register (NWU_INT3_FIXPRI, R/W, Address = 0xE030_0B20)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 3 if fixed priority mode: 0~7	0

5.5.112 Non wake-up Interrupt 4 Fixed Priority Control Register (NWU_INT4_FIXPRI, R/W, Address = 0xE030_0B24)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 4 if fixed priority mode: 0~7	0

5.5.113 Non wake-up Interrupt 5 Fixed Priority Control Register (NWU_INT5_FIXPRI, R/W, Address = 0xE030_0B28)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 5 if fixed priority mode: 0~7	0

5.5.114 Non wake-up Interrupt 6 Fixed Priority Control Register (NWU_INT6_FIXPRI, R/W, Address = 0xE030_0B2C)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 6 if fixed priority mode: 0~7	0

5.5.115 Non wake-up Interrupt 7 Fixed Priority Control Register (NWU_INT7_FIXPRI, R/W, Address = 0xE030_0B30)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 7 if fixed priority mode: 0~7	0

5.5.116 Non wake-up Interrupt 8 Fixed Priority Control Register (NWU_INT8_FIXPRI, R/W, Address = 0xE030_0B34)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 8 if fixed priority mode: 0~7	0

5.5.117 Non wake-up Interrupt 9 Fixed Priority Control Register (NWU_INT9_FIXPRI, R/W, Address = 0xE030_0B38)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 9 if fixed priority mode: 0~7	0

5.5.118 Non wake-up Interrupt 10 Fixed Priority Control Register (NWU_INT10_FIXPRI, R/W, Address = 0xE030_0B3C)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 10 if fixed priority mode: 0~7	0

5.5.119 Non wake-up Interrupt 11 Fixed Priority Control Register (NWU_INT11_FIXPRI, R/W, Address = 0xE030_0B40)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 11 if fixed priority mode: 0~7	0

5.5.120 Non wake-up Interrupt 12 Fixed Priority Control Register (NWU_INT12_FIXPRI, R/W, Address = 0xE030_0B44)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 12 if fixed priority mode: 0~7	0

5.5.121 Non wake-up Interrupt 13 Fixed Priority Control Register (NWU_INT13_FIXPRI, R/W, Address = 0xE030_0B48)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 13 if fixed priority mode: 0~7	0

5.5.122 Non wake-up Interrupt 14 Fixed Priority Control Register (NWU_INT14_FIXPRI, R/W, Address = 0xE030_0B4C)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 14 if fixed priority mode: 0~7	0

5.5.123 Non wake-up Interrupt 15 Fixed Priority Control Register (NWU_INT15_FIXPRI, R/W, Address = 0xE030_0B50)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 15 if fixed priority mode: 0~7	0

5.5.124 Non wake-up Interrupt 16 Fixed Priority Control Register (NWU_INT16_FIXPRI, R/W, Address = 0xE030_0B54)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 16 if fixed priority mode: 0~7	0

5.5.125 Non wake-up Interrupt 17 Fixed Priority Control Register (NWU_INT17_FIXPRI, R/W, Address = 0xE030_0B58)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 17 if fixed priority mode: 0~7	0

5.5.126 Non wake-up Interrupt 18 Fixed Priority Control Register (NWU_INT18_FIXPRI, R/W, Address = 0xE030_0B5C)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 18 if fixed priority mode: 0~7	0

5.5.127 Non wake-up Interrupt 19 Fixed Priority Control Register (NWU_INT19_FIXPRI, R/W, Address = 0xE030_0B60)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 19 if fixed priority mode: 0~7	0

5.5.128 Non wake-up Interrupt 20 Fixed Priority Control Register (NWU_INT20FIXPRI, R/W, Address = 0xE030_0B64)

Field	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Highest_EINT_NUM	[2:0]	Interrupt number of the highest priority in Non wake-up Interrupt Group 20 if fixed priority mode: 0~7	0

5.6 WAKE-UP INTERRUPT CONTROL REGISTERS

Wake-up Interrupt consists of 32.(GPH0 ~ GPH3). These are used for wake-up source in Power down mode and, idle mode, all interrupts can be wake-up source, the other groups of Wake-up interrupts also can be the wake-up sources.

If these wake-up interrupts are used in normal mode, it works as Non wake-up Interrupt

The following table is the list of Wake-up interrupt control registers.

5.6.1 Wake-up Interrupt Configuration Register0_7 (WKUP_INT_CON0_7, R/W, Address = 0xE030_0E00)

3 bits control 1 pin of Wake-up interrupt as follows,

000 = Low level 001 = High level
010 = Falling edge triggered 011 = Rising edge triggered
100 = Both edge triggered 101 ~ 111 = Reserved

Field	Bit	Description	Reset Value
Reserved	[31]	Reserved	0
WKUP_INT_CON0_7[7]	[30:28]	Sets the signaling method for WKUP_INT[7]	000
Reserved	[27]	Reserved	0
WKUP_INT_CON0_7[6]	[26:24]	Sets the signaling method for WKUP_INT[6]	000
Reserved	[23]	Reserved	0
WKUP_INT_CON0_7[5]	[22:20]	Sets the signaling method for WKUP_INT[5]	000
Reserved	[19]	Reserved	0
WKUP_INT_CON0_7[4]	[18:16]	Sets the signaling method for WKUP_INT[4]	000
Reserved	[15]	Reserved	0
WKUP_INT_CON0_7[3]	[14:12]	Sets the signaling method for WKUP_INT[3]	000
Reserved	[11]	Reserved	0
WKUP_INT_CON0_7[2]	[10:8]	Sets the signaling method for WKUP_INT[2]	000
Reserved	[7]	Reserved	0
WKUP_INT_CON0_7[1]	[6:4]	Sets the signaling method for WKUP_INT[1]	000
Reserved	[3]	Reserved	0
WKUP_INT_CON0_7[0]	[2:0]	Sets the signaling method for WKUP_INT[0]	000

5.6.2 Wake-up Interrupt Configuration Register8_15 (WKUP_INT_CON8_15, R/W, Address = 0xE030_0E04)

3 bits control 1 pin of Wake-up interrupt as follows,

000 = Low level 001 = High level
010 = Falling edge triggered 011 = Rising edge triggered
100 = Both edge triggered 101 ~ 111 = Reserved

Field	Bit	Description	Reset Value
Reserved	[31]	Reserved	0
WKUP_INT_CON8_15[7]	[30:28]	Sets the signaling method for WKUP_INT[15]	000
Reserved	[27]	Reserved	0
WKUP_INT_CON8_15[6]	[26:24]	Sets the signaling method for WKUP_INT[14]	000
Reserved	[23]	Reserved	0
WKUP_INT_CON8_15[5]	[22:20]	Sets the signaling method for WKUP_INT[13]	000
Reserved	[19]	Reserved	0
WKUP_INT_CON8_15[4]	[18:16]	Sets the signaling method for WKUP_INT[12]	000
Reserved	[15]	Reserved	0
WKUP_INT_CON8_15[3]	[14:12]	Sets the signaling method for WKUP_INT[11]	000
Reserved	[11]	Reserved	0
WKUP_INT_CON8_15[2]	[10:8]	Sets the signaling method for WKUP_INT[10]	000
Reserved	[7]	Reserved	0
WKUP_INT_CON8_15[1]	[6:4]	Setting the signaling method for WKUP_INT[9]	000
Reserved	[3]	Reserved	0
WKUP_INT_CON8_15[0]	[2:0]	Sets the signaling method for WKUP_INT[8]	000

5.6.3 Wake-up Interrupt Configuration Register16_23 (WKUP_INT_CON16_23, R/W, Address = 0xE030_0E08)

3 bits control 1 pin of Wake-up interrupt as follows,

000 = Low level 001 = High level
010 = Falling edge triggered 011 = Rising edge triggered
100 = Both edge triggered 101 ~ 111 = Reserved

Field	Bit	Description	Reset Value
Reserved	[31]	Reserved	0
WKUP_INT_CON16_23[7]	[30:28]	Sets the signaling method for WKUP_INT[23]	000
Reserved	[27]	Reserved	0
WKUP_INT_CON16_23[6]	[26:24]	Sets the signaling method for WKUP_INT[22]	000
Reserved	[23]	Reserved	0
WKUP_INT_CON16_23[5]	[22:20]	Sets the signaling method for WKUP_INT[21]	000
Reserved	[19]	Reserved	0
WKUP_INT_CON16_23[4]	[18:16]	Sets the signaling method for WKUP_INT[20]	000
Reserved	[15]	Reserved	0
WKUP_INT_CON16_23[3]	[14:12]	Sets the signaling method for WKUP_INT[19]	000
Reserved	[11]	Reserved	0
WKUP_INT_CON16_23[2]	[10:8]	Sets the signaling method for WKUP_INT[18]	000
Reserved	[7]	Reserved	0
WKUP_INT_CON16_23[1]	[6:4]	Sets the signaling method for WKUP_INT[17]	000
Reserved	[3]	Reserved	0
WKUP_INT_CON16_23[0]	[2:0]	Sets the signaling method for WKUP_INT[16]	000

5.6.4 Wake-up Interrupt Configuration Register24_31 (WKUP_INT_CON24_31, R/W, Address = 0xE030_0E0C)

3 bits control 1 pin of Wake-up interrupt as follows,

000 = Low level 001 = High level
010 = Falling edge triggered 011 = Rising edge triggered
100 = Both edge triggered 101 ~ 111 = Reserved

Field	Bit	Description	Reset Value
Reserved	[31]	Reserved	0
WKUP_INT_CON24_31[7]	[30:28]	Sets the signaling method for WKUP_INT[31]	000
Reserved	[27]	Reserved	0
WKUP_INT_CON24_31[6]	[26:24]	Sets the signaling method for WKUP_INT[30]	000
Reserved	[23]	Reserved	0
WKUP_INT_CON24_31[5]	[22:20]	Sets the signaling method for WKUP_INT[29]	000
Reserved	[19]	Reserved	0
WKUP_INT_CON24_31[4]	[18:16]	Sets the signaling method for WKUP_INT[28]	000
Reserved	[15]	Reserved	0
WKUP_INT_CON24_31[3]	[14:12]	Sets the signaling method for WKUP_INT[27]	000
Reserved	[11]	Reserved	0
WKUP_INT_CON24_31[2]	[10:8]	Sets the signaling method for WKUP_INT[26]	000
Reserved	[7]	Reserved	0
WKUP_INT_CON24_31[1]	[6:4]	Sets the signaling method for WKUP_INT[25]	000
Reserved	[3]	Reserved	0
WKUP_INT_CON24_31[0]	[2:0]	Sets the signaling method for WKUP_INT[24]	000

5.6.5 Wake-up Interrupt Filter Configuration Register0_3 (WKUP_INT_FLTCON0_3, R/W, Address = 0xE030_0E80)

Field	Bit	Description	Reset Value
FLTEN0_3[3]	[31]	Filter Enable for WKUP_INT[3] 0 = disables 1 = enables	0
FLTSEL0_3[3]	[30]	Filter Selection for WKUP_INT[3] 0 = delay filter 1 = digital filter(clock count)	0
EINT0_3[3]	[29:24]	Filtering width for WKUP_INT[3] This value is valid if FLTSEL0 is 1.	000
FLTEN0_3[2]	[23]	Filter Enable for WKUP_INT[2] 0 = disables 1 = enables	0
FLTSEL0_3[2]	[22]	Filter Selection for WKUP_INT[2] 0 = delay filter 1 = digital filter(clock count)	0
EINT0_3[2]	[21:16]	Filtering width for WKUP_INT[2] This value is valid if FLTSEL0 is 1.	000
FLTEN0_3[1]	[15]	Filter Enable for WKUP_INT[1] 0 = disables 1 = enables	0
FLTSEL0_3[1]	[14]	Filter Selection for WKUP_INT[1] 0 = delay filter 1 = digital filter(clock count)	0
EINT0_3[1]	[13:8]	Filtering width for WKUP_INT[1] This value is valid if FLTSEL0 is 1.	000
FLTEN0_3[0]	[7]	Filter Enable for WKUP_INT[0] 0 = disables 1 = enables	0
FLTSEL0_3[0]	[6]	Filter Selection for WKUP_INT[0] 0 = delay filter 1 = digital filter(clock count)	0
EINT0_3[0]	[5:0]	Filtering width for WKUP_INT[0] This value is valid if FLTSEL0 is 1.	000

5.6.6 Wake-up Interrupt Filter Configuration Register4_7 (WKUP_INT_FLTCON4_7, R/W, Address = 0xE030_0E84)

Field	Bit	Description	Reset Value
FLTEN4_7[7]	[31]	Filter Enable for WKUP_INT[7] 0 = disables 1 = enables	0
FLTSEL4_7[7]	[30]	Filter Selection for WKUP_INT[7] 0 = delay filter 1 = digital filter(clock count)	0
EINT4_7[7]	[29:24]	Filtering width for WKUP_INT[7] This value is valid if FLTSEL0 is 1.	000
FLTEN4_7[6]	[23]	Filter Enable for WKUP_INT[6] 0 = disables 1 = enables	0
FLTSEL4_7[6]	[22]	Filter Selection for WKUP_INT[6] 0 = delay filter 1 = digital filter(clock count)	0
EINT4_7[6]	[21:16]	Filtering width for WKUP_INT[6] This value is valid if FLTSEL0 is 1.	000
FLTEN4_7[5]	[15]	Filter Enable for WKUP_INT[5] 0 = disables 1 = enables	0
FLTSEL4_7[5]	[14]	Filter Selection for WKUP_INT[5] 0 = delay filter 1 = digital filter(clock count)	0
EINT4_7[5]	[13:8]	Filtering width for WKUP_INT[5] This value is valid if FLTSEL0 is 1.	000
FLTEN4_7[4]	[7]	Filter Enable for WKUP_INT[4] 0 = disables 1 = enables	0
FLTSEL4_7[4]	[6]	Filter Selection for WKUP_INT[4] 0 = delay filter 1 = digital filter(clock count)	0
EINT4_7[4]	[5:0]	Filtering width for WKUP_INT[4] This value is valid if FLTSEL0 is 1.	000

5.6.7 Wake-up Interrupt Filter Configuration Register8_11 (WKUP_INT_FLTCON8_11, R/W, Address = 0xE030_0E88)

Field	Bit	Description	Reset Value
FLTEN8_11[3]	[31]	Filter Enable for WKUP_INT[11] 0 = disables 1 = enables	0
FLTSEL8_11[3]	[30]	Filter Selection for WKUP_INT[11] 0 = delay filter 1 = digital filter(clock count)	0
EINT8_11[3]	[29:24]	Filtering width for WKUP_INT[11] This value is valid if FLTSEL1 is 1.	000
FLTEN8_11[2]	[23]	Filter Enable for WKUP_INT[10] 0 = disables 1 = enables	0
FLTSEL8_11[2]	[22]	Filter Selection for WKUP_INT[10] 0 = delay filter 1 = digital filter(clock count)	0
EINT8_11[2]	[21:16]	Filtering width for WKUP_INT[10] This value is valid if FLTSEL1 is 1.	000
FLTEN8_11[1]	[15]	Filter Enable for WKUP_INT[9] 0 = disables 1 = enables	0
FLTSEL8_11[1]	[14]	Filter Selection for WKUP_INT[9] 0 = delay filter 1 = digital filter(clock count)	0
EINT8_11[1]	[13:8]	Filtering width for WKUP_INT[9] This value is valid if FLTSEL1 is 1.	000
FLTEN8_11[0]	[7]	Filter Enable for WKUP_INT[8] 0 = disables 1 = enables	0
FLTSEL8_11[0]	[6]	Filter Selection for WKUP_INT[8] 0 = delay filter 1 = digital filter(clock count)	0
EINT8_11[0]	[5:0]	Filtering width for WKUP_INT[8] This value is valid if FLTSEL1 is 1.	000

5.6.8 Wake-up Interrupt Filter Configuration Register12_15 (WKUP_INT_FLTCON12_15, R/W, Address = 0xE030_0E8C)

Field	Bit	Description	Reset Value
FLTEN12_15[7]	[31]	Filter Enable for WKUP_INT[15] 0 = disables 1 = enables	0
FLTSEL12_15[7]	[30]	Filter Selection for WKUP_INT[15] 0 = delay filter 1 = digital filter(clock count)	0
EINT12_15[7]	[29:24]	Filtering width for WKUP_INT[15] This value is valid if FLTSEL1 is1.	000
FLTEN12_15[6]	[23]	Filter Enable for WKUP_INT[14] 0 = disables 1 = enables	0
FLTSEL12_15[6]	[22]	Filter Selection for WKUP_INT[14] 0 = delay filter 1 = digital filter(clock count)	0
EINT12_15[6]	[21:16]	Filtering width for WKUP_INT[14] This value is valid if FLTSEL1 is1.	000
FLTEN12_15[5]	[15]	Filter Enable for WKUP_INT[13] 0 = disables 1 = enables	0
FLTSEL12_15[5]	[14]	Filter Selection for WKUP_INT[13] 0 = delay filter 1 = digital filter(clock count)	0
EINT12_15[5]	[13:8]	Filtering width for WKUP_INT[13] This value is valid if FLTSEL1 is1.	000
FLTEN12_15[4]	[7]	Filter Enable for WKUP_INT[12] 0 = disables 1 = enables	0
FLTSEL12_15[4]	[6]	Filter Selection for WKUP_INT[12] 0 = delay filter 1 = digital filter(clock count)	0
EINT12_15[4]	[5:0]	Filtering width for WKUP_INT[12] This value is valid if FLTSEL1 is1.	000

5.6.9 Wake-up Interrupt Filter Configuration Register16_19 (WKUP_INT_FLTCON16_19, R/W, Address = 0xE030_0E90)

Field	Bit	Description	Reset Value
FLTEN16_19[3]	[31]	Filter Enable for WKUP_INT[19] 0 = disables 1 = enables	0
FLTSEL16_19[3]	[30]	Filter Selection for WKUP_INT[19] 0 = delay filter 1 = digital filter(clock count)	0
EINT16_19[3]	[29:24]	Filtering width for WKUP_INT[19] This value is valid if FLTSEL2 is1.	000
FLTEN16_19[2]	[23]	Filter Enable for WKUP_INT[18] 0 = disables 1 = enables	0
FLTSEL16_19[2]	[22]	Filter Selection for WKUP_INT[18] 0 = delay filter 1 = digital filter(clock count)	0
EINT16_19[2]	[21:16]	Filtering width for WKUP_INT[18] This value is valid if FLTSEL2 is1.	000
FLTEN16_19[1]	[15]	Filter Enable for WKUP_INT[17] 0 = disables 1 = enables	0
FLTSEL16_19[1]	[14]	Filter Selection for WKUP_INT[17] 0 = delay filter 1 = digital filter(clock count)	0
EINT16_19[1]	[13:8]	Filtering width for WKUP_INT[17] This value is valid if FLTSEL2 is1.	000
FLTEN16_19[0]	[7]	Filter Enable for WKUP_INT[16] 0 = disables 1 = enables	0
FLTSEL16_19[0]	[6]	Filter Selection for WKUP_INT[16] 0 = delay filter 1 = digital filter(clock count)	0
EINT16_19[0]	[5:0]	Filtering width for WKUP_INT[16] This value is valid if FLTSEL2 is1.	000

5.6.10 Wake-up Interrupt Filter Configuration Register20_23 (WKUP_INT_FLTCON20_23, R/W, Address = 0xE030_0E94)

Field	Bit	Description	Reset Value
FLTEN20_23[7]	[31]	Filter Enable for WKUP_INT[23] 0 = disables 1 = enables	0
FLTSEL20_23[7]	[30]	Filter Selection for WKUP_INT[23] 0 = delay filter 1 = digital filter(clock count)	0
EINT20_23[7]	[29:24]	Filtering width for WKUP_INT[23] This value is valid if FLTSEL2 is1.	000
FLTEN20_23[6]	[23]	Filter Enable for WKUP_INT[22] 0 = disables 1 = enables	0
FLTSEL20_23[6]	[22]	Filter Selection for WKUP_INT[22] 0 = delay filter 1 = digital filter(clock count)	0
EINT20_23[6]	[21:16]	Filtering width for WKUP_INT[22] This value is valid if FLTSEL2 is1.	000
FLTEN20_23[5]	[15]	Filter Enable for WKUP_INT[21] 0 = disables 1 = enables	0
FLTSEL20_23[5]	[14]	Filter Selection for WKUP_INT[21] 0 = delay filter 1 = digital filter(clock count)	0
EINT20_23[5]	[13:8]	Filtering width for WKUP_INT[21] This value is valid if FLTSEL2 is1.	000
FLTEN20_23[4]	[7]	Filter Enable for WKUP_INT[20] 0 = disables 1 = enables	0
FLTSEL20_23[4]	[6]	Filter Selection for WKUP_INT[20] 0 = delay filter 1 = digital filter(clock count)	0
EINT20_23[4]	[5:0]	Filtering width for WKUP_INT[20] This value is valid if FLTSEL2 is1.	000

5.6.11 Wake-up Interrupt Filter Configuration Register24_27 (WKUP_INT_FLTCON24_27, R/W, Address = 0xE030_0E98)

Field	Bit	Description	Reset Value
FLTEN24_27[3]	[31]	Filter Enable for WKUP_INT[27] 0 = disables 1 = enables	0
FLTSEL24_27[3]	[30]	Filter Selection for WKUP_INT[27] 0 = delay filter 1 = digital filter(clock count)	0
EINT24_27[3]	[29:24]	Filtering width for WKUP_INT[27] This value is valid if FLTSEL3 is1.	000
FLTEN24_27[2]	[23]	Filter Enable for WKUP_INT[26] 0 = disables 1 = enables	0
FLTSEL24_27[2]	[22]	Filter Selection for WKUP_INT[26] 0 = delay filter 1 = digital filter(clock count)	0
EINT24_27[2]	[21:16]	Filtering width for WKUP_INT[26] This value is valid if FLTSEL3 is1.	000
FLTEN24_27[1]	[15]	Filter Enable for WKUP_INT[25] 0 = disables 1 = enables	0
FLTSEL24_27[1]	[14]	Filter Selection for WKUP_INT[25] 0 = delay filter 1 = digital filter(clock count)	0
EINT24_27[1]	[13:8]	Filtering width for WKUP_INT[25] This value is valid if FLTSEL3 is1.	000
FLTEN24_27[0]	[7]	Filter Enable for WKUP_INT[24] 0 = disables 1 = enables	0
FLTSEL24_27[0]	[6]	Filter Selection for WKUP_INT[24] 0 = delay filter 1 = digital filter(clock count)	0
EINT24_27[0]	[5:0]	Filtering width for WKUP_INT[24] This value is valid if FLTSEL3 is1.	000

5.6.12 Wake-up Interrupt Filter Configuration Register28_31 (WKUP_INT_FLTCON28_31, R/W, Address = 0xE030_0E9C)

Field	Bit	Description	Reset Value
FLTEN28_31[7]	[31]	Filter Enable for WKUP_INT[31] 0 = disables 1 = enables	0
FLTSEL28_31[7]	[30]	Filter Selection for WKUP_INT[31] 0 = delay filter 1 = digital filter(clock count)	0
EINT28_31[7]	[29:24]	Filtering width for WKUP_INT[31] This value is valid if FLTSEL3 is1.	000
FLTEN28_31[6]	[23]	Filter Enable for WKUP_INT[30] 0 = disables 1 = enables	0
FLTSEL28_31[6]	[22]	Filter Selection for WKUP_INT[30] 0 = delay filter 1 = digital filter(clock count)	0
EINT28_31[6]	[21:16]	Filtering width for WKUP_INT[30] This value is valid if FLTSEL3 is1.	000
FLTEN28_31[5]	[15]	Filter Enable for WKUP_INT[29] 0 = disables 1 = enables	0
FLTSEL28_31[5]	[14]	Filter Selection for WKUP_INT[29] 0 = delay filter 1 = digital filter(clock count)	0
EINT28_31[5]	[13:8]	Filtering width for WKUP_INT[29] This value is valid if FLTSEL3 is1.	000
FLTEN28_31[4]	[7]	Filter Enable for WKUP_INT[28] 0 = disables 1 = enables	0
FLTSEL28_31[4]	[6]	Filter Selection for WKUP_INT[28] 0 = delay filter 1 = digital filter(clock count)	0
EINT28_31[4]	[5:0]	Filtering width for WKUP_INT[28] This value is valid if FLTSEL3 is1.	000

5.6.13 Wake-up Interrupt Mask Register0_7 (WKUP_INT_MASK0_7, R/W, Address = 0xE030_0F00)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_MASK0_7[n]	[n]	Mask register for WKUP_INT[n] 0 = Enabled,1 = Masked (n=0~7)	1

5.6.14 Wake-up Interrupt Mask Register8_15 (WKUP_INT_MASK8_15, R/W, Address = 0xE030_0F04)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_MASK8_15[n]	[n]	Mask register for WKUP_INT[n+8] 0 = Enabled,1 = Masked (n=0~7)	1

5.6.15 Wake-up Interrupt Mask Register16_23 (WKUP_INT_MASK16_23, R/W, Address = 0xE030_0F08)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_MASK16_23[n]	[n]	Mask register for WKUP_INT[n+16] 0 = Enabled,1 = Masked (n=0~7)	1

5.6.16 Wake-up Interrupt Mask Register24_31 (WKUP_INT_MASK24_31, R/W, Address = 0xE030_0F0C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_MASK24_31[n]	[n]	Mask register for WKUP_INT[n+24] 0 = Enabled,1 = Masked (n=0~7)	1

5.6.17 Wake-up Interrupt Pending Register0_7 (WKUP_INT_PEND0_7, R/W, Address = 0xE030_0F40)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_PEND0_7[n]	[n]	This bit is set if WKUP_INT [n] is pending. Writing '1' makes this bit clear. (n=0~7)	0

Note. Even though WKUP_INT_MASK0_7[n] is masked, this register can be pended.

5.6.18 Wake-up Interrupt Pending Register8_15 (WKUP_INT_PEND8_15, R/W, Address = 0xE030_0F44)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_PEND8_15[n]	[n]	This bit is set if WKUP_INT [n+8] is pending. Writing '1' makes this bit clear. (n=0~7)	0

Note. Even though WKUP_INT_MASK8_15[n] is masked, this register can be pended.

5.6.19 Wake-up Interrupt Pending Register16_23 (WKUP_INT_PEND16_23, R/W, Address = 0xE030_0F48)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_PEND16_23[n]	[n]	This bit is set if WKUP_INT [n+16] is pending. Writing '1' makes this bit clear. (n=0~7)	0

Note. Even though WKUP_INT_MASK16_23[n] is masked, this register can be pended.

5.6.20 Wake-up Interrupt Pending Register24_31 (WKUP_INT_PEND24_31, R/W, Address = 0xE030_0F4C)

Field	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WKUP_INT_PEND24_31[n]	[n]	This bit is set if WKUP_INT [n+24] is pending. Writing '1' makes this bit clear.(n=0~7)	0

Note. Even though WKUP_INT_MASK24_31[n] is masked, this register can be pended.

5.7 EXTERN PIN CONFIGURATION REGISTERS IN POWER DOWN MODE

This registers keep their values during power down mode

5.7.1 Power Down Mode Pad Configure Register (PDNEN, R/W, Address = 0xE030_0F80)

Field	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
PDNEN_CFG	[1]	0 = Automatically by power down mode 1= by PDNEN bit	0
PDNEN	[0]	Power down mode pad state enable register. If this bit is set to '1', external pins are controlled by power down mode control register such as GPA0PDNCON, GPA0PDNPULL 1 = PADs Controlled by Power Down mode control registers 0 = PADs Controlled by normal mode This bit automatically set to '1' if system enters into Power down mode and write '0' to this bit or cold reset to clear. After wake up from Power down mode, this bit maintains value '1' until writing '0'	0

5.8 USB HOST 1.1 PIN CONTROL REGISTER

5.8.1 USB Host 1.1 Pad Configure Register (UHOST, R/W, Address = 0xE030_B68)

Field	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	0
USB_DMPD	[4]	USB DM Pull-down(3) control 0 = disables 1 = enables	0
USB_DPPD	[3]	USB DP Pull-down(3) control 0 = disables 1 = enables	0
USB_PUSW2	[2]	USB Pull-up(2) switch2 control 0 = off 1 = on	0
USB_PUSW1	[1]	USB Pull-up(1) switch1 control 0 = off 1 = on	0
USB_SUSPND	[0]	Make USB Tranceiver PAD to enter suspend mode. 0 = Normal mode 1 = Suspend mode	0

NOTES:

1. Pull-up resistance is 1.2kohm. Refer to figure 2.2-2
2. Pull-up resistance is 0.5kohm. Refer to figure 2.2-2.
3. Pull-down resistance is 20kohm. Refer to figure 2.2-2.

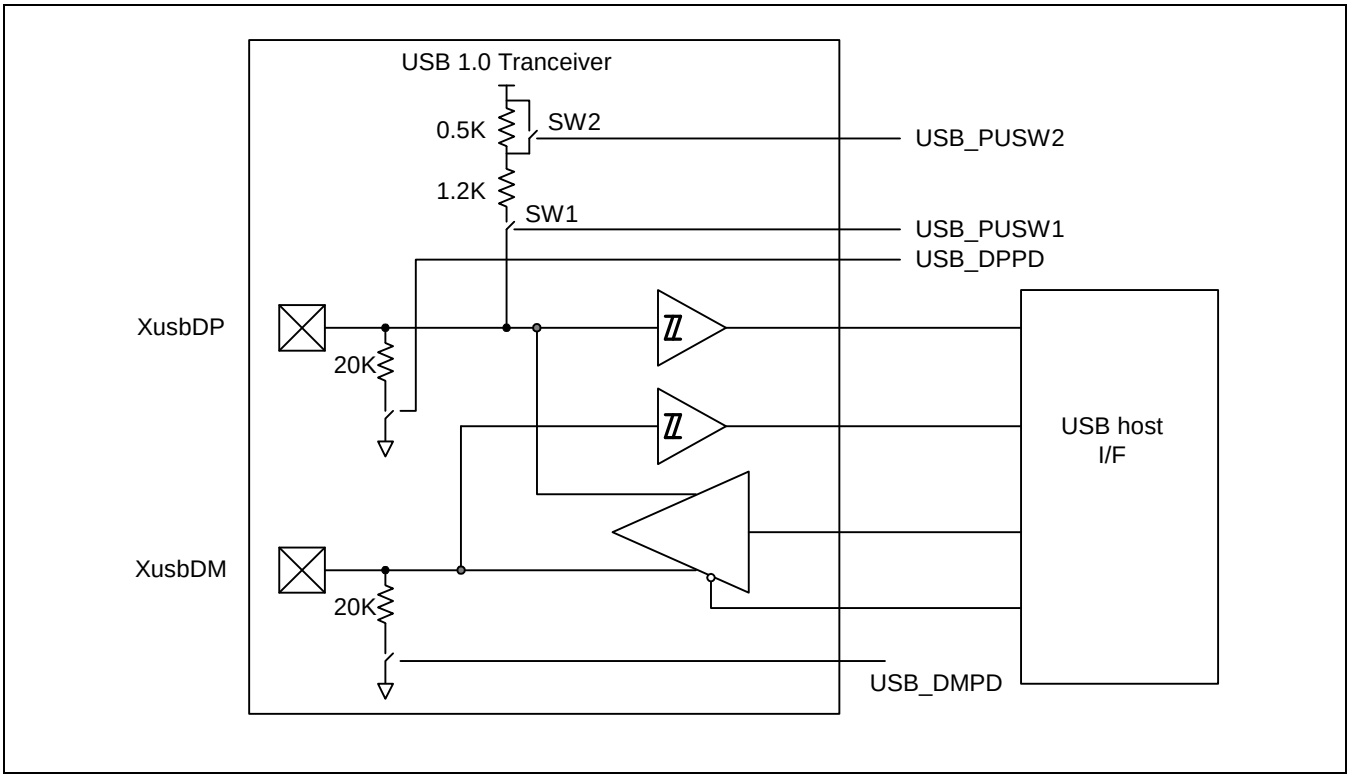


Figure 2.2-2 USB Transceiver Block Diagram

2.3 CLOCK CONTROLLER

1 CLOCK DOMAIN

S5PC100 consists of three bus parts. First part consists of Cortex A8, D0_BUS and D0_BUS-attached modules. Cortex A8 supports **only synchronous mode** so that Cortex A8 and D0_BUS must operate synchronously. Second part consists of D1_BUS and D1_BUS-attached modules. Final part, D2 domain, is for low-power audio play.

D0 domain operates at up-to 166MHz clock, D1 domain operates at up-to 133MHz (D1 domain has many multimedia IPs which can be synthesized under 133MHz.). D2 domain operates at up-to 80MHz. All 3 parts communicate asynchronously.

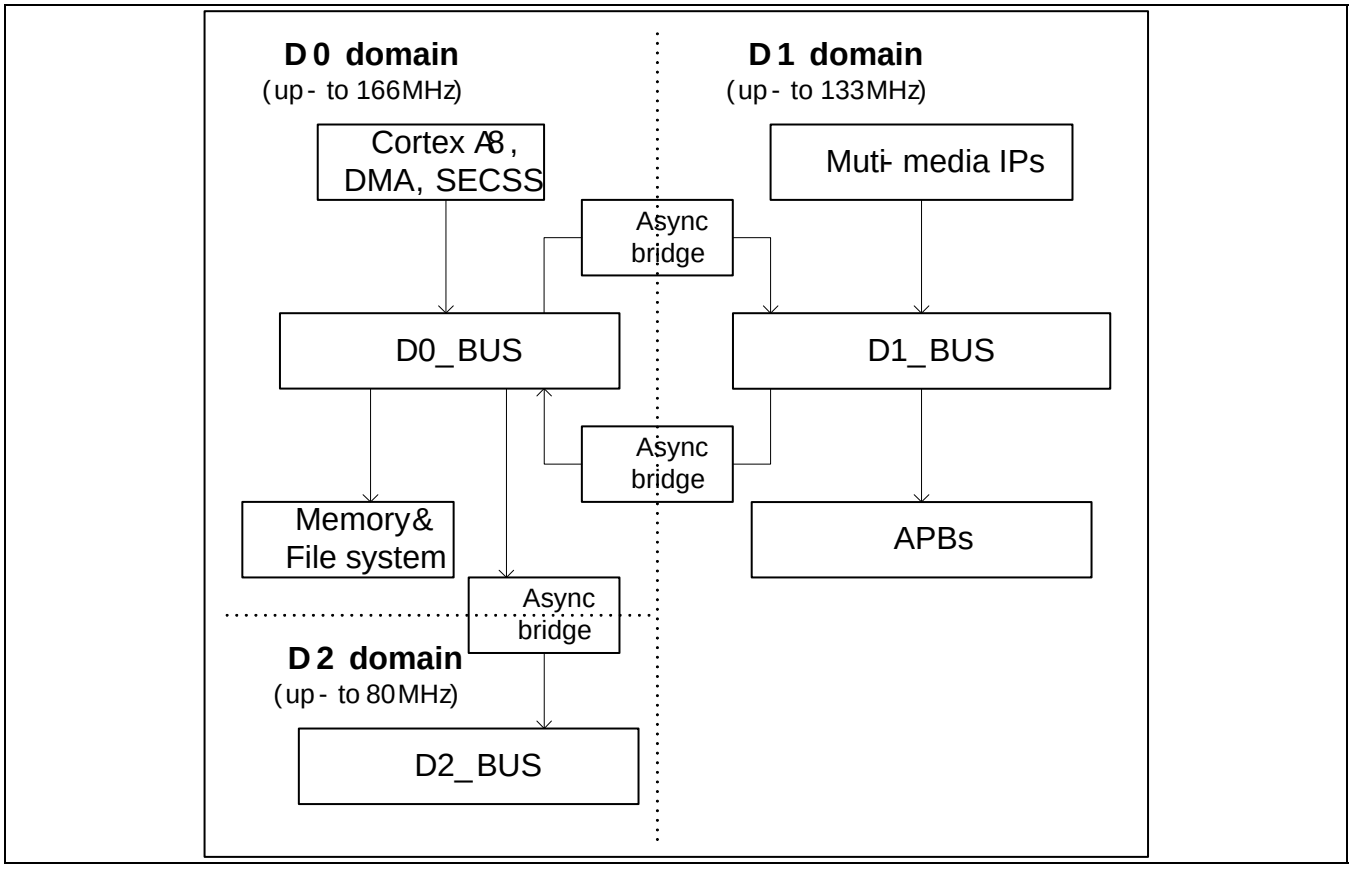


Figure 2.3-1 S5PC100 Clock Domain

2 CLOCK DECLARATION

The clocks in S5C100 are classified into following:

- Clocks from clock pads, i.e., XusbXTI, XXTI, XXTI27, and XrtcXTI
- Clocks from a Clock Management Unit (CMU), e.g., HCLKD0, HCLKD1, and etc
- Clocks from USB OTG PHY (refer 7.8)
- Clocks from GPIO pads, e.g., XciPCLK, XpwmECLK, and etc

2.1 CLOCKS FROM CLOCK PADS

The following clocks are provided from clock pads. Crystal clock pads can be disabled.

- **USB_XI:** Clock from a crystal pad with XusbXTI and XusbXTO pins. This clock is supplied to (A, E, M) PLLs as well as an USB PHY. For more information on USB PHY clock, refer to "Section 7.8". Input frequency range: 12 ~ 48MHz for supplying clock to USB PHY
- **OSC_IN:** Clock from a crystal pad with XXTI and XXTO pins. If USB is not used at commercial set, CMU and PLL use this clock to generate clocks to modules. Input frequency range: 12 ~ 20MHz
- **OSC27_IN:** Cock from 27MHz a crystal pad with XXTI27 and XXTO27 pins. Clock doubler in CMU, HPLL, and MIPI D-PHY use this clock. Clock doubler makes 54MHz clock (named, VCLK_54) for SD TV out. HPLL usually makes 74.25 MHz for HD @60fps, and 74.176 MHz for HD @59.94fps. A PLL in MIPI D-PHY also uses this clock to make MIPI clock (up-to 1GHz).
- **RTC_XT:** Clock from a 32.768 KHz crystal pad with XrtcXTI and XrtcXTO pins. RTC can use this clock as the source of a real time clock.

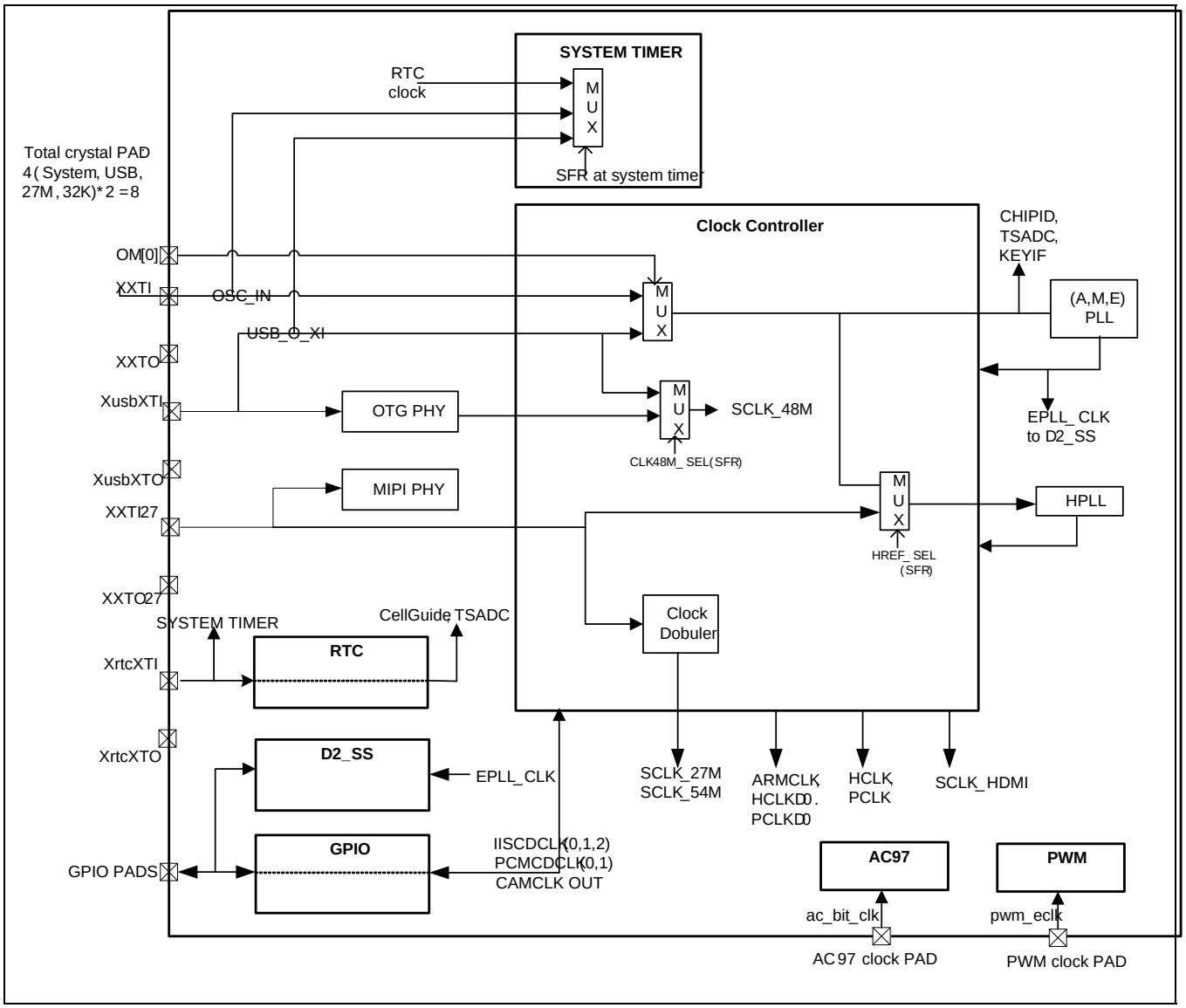


Figure 2.3-2 S5PC100 Top-Level Clocks

- XXTI and XXTO use wide-range OSC pads.
- **XusbXTI and XusbXTO use wide-range OSC pads.**
- XXTI27 and XXTO27 use wide-range OSC pads.
- XrtcXTI and XrtcXTO use OSC pads for RTC.
- ARMCLK means clock for Cortex A8.
- HCLKD0 means clock for D0_BUS and D0_BUS-attached modules.
- PCLKD0 means clock for APB modules at D0 domain.
- HCLKD1 means clock for D1_BUS and D1_BUS-attached modules.
- PCLKD1 means clock for APB modules at D1 domain.
- HCLKD2 means clock for D2_BUS and D2_BUS-attached modules.
- PCLKD2 means clock for APB modules at D2 domain.
- SCLK_HDMI means clock to HDMI, and MIXER.
- SCLK_48M means 48MHz clock to SPI, MMC, USB HOST1.1 and IrDA.
- SCLK_27M means 27MHz clock to MIXER, MIPI D-PHY, SPI, and MMC.
- SCLK_54M means 54MHz clock to VDAC, TV, MIXER, LCD, and FIMC.

2.2 CLOCKS FROM CMU

CMU generates internal clocks with a various intermediate frequencies using clocks from the clock pads (i.e., XXTI, XXTI27, XrtcXTI, and XusbXTI), four PLLs (i.e., APLL, MPLL, EPLL, and HPLL), and USB_OTG PHY clock. Some of them are selected, pre-scaled, and provided to the corresponding modules. Products of AP typically use 12MHz as an input clock source of APLL, MPLL and EPLL. Samsung recommend 27MHz for input clock of HPLL.

The following components are used to generate internal clocks:

- APLL uses SRCLK as input to generate 50MHz ~ 2GHz
- MPLL uses SRCLK as input to generate 10MHz ~ 600MHz.
- EPLL uses SRCLK as input to generate 10MHz~ 600MHz. This is mainly used for audio clock.
- HPLL uses OSC27_IN as input to generate 10MHz ~ 600MHz clock, especially, this PLL must generate clean clock for making 74.176MHz and 74.25MHz (HDMI clocks).
- **Clock Doubler** uses OSC27_IN to generate 54 MHz clock for SD TV out.
- USB OTG PHY uses XusbXTI to generate 30 and 48MHz.

In typical S5PC100 applications:

- APLL is used for Cortex A8 and D0_BUS domain (i.e., ARMCLK, HCLKD0, PCLKD0, HCLKD0_SECSS, and SCLK_ONENAND).
- MPLL is used for D1_BUS domain clock (HCLKD1 and PCLKD1) and other peripheral clocks (i.e., audio IPs, SPI, and etc.).
- EPLL is used for peripheral clocks (i.e., audio IPs, SPI, and etc.).
- HPLL is mainly used for a HD video clock (i.e., SCLK_HDMI).

Clock controller allows the bypassing of PLLs for slow clock and the connection/ disconnection of the clock to each block (clock gating) by software, which results in power reduction.

3 CLOCK RELATIONSHIP

Clocks have the following relationship:

- D0 bus domain
  - ♦ freq(APLLCLK) = n x freq(ARMCLK), where n = 2 ~ 8, 10, 12, 14, 16
    - – Two cascaded divider = DIV_APLL (divide up-to 2) & DIV_ARM (divide up-to 8)
  - ♦ freq(ARMCLK) = n x freq(HCLKD0), where n = 1 ~ 8
  - ♦ freq(PCLKD0) = freq(HCLKD0) / n, where n = 1 ~ 8
  - ♦ freq(HCLKD0_SECSS) = freq(HCLKD0) / n, where n = 1 ~ 8
  - ♦ freq(SCLK_ONENAND) = freq(HCLKD0) / n, where n = 1 ~ 4 (sync with D0 bus mode)
  - ♦ freq(SCLK_ONENAND2) = freq(ONENANDCLK) / 2 (sync with D0 bus mode)
- D1 bus domain
  - ♦ freq(PCLKD1) = freq(HCLKD1) / n, where n = 1 ~ 8
- D2 bus domain
  - ♦ HCLKD2 and PCLKD2 have no relationship.

The following values are for the high-performance operation:

- freq(ARMCLK) = 667 MHz
- freq(HCLKD0) = 166 MHz
- freq(PCLKD0) = 83 MHz
- freq(HCLKSECSS) = 83 MHz
- freq(ONENANDCLK) = 166 MHz (sync mode), 133 MHz (async mode)
- freq(ONENANDCLK_2) = 83 MHz (sync mode), 66.5MHz (async mode)
- freq(HCLKD1) = 133 MHz
- freq(PCLKD1) = 66.5 MHz
- freq(HCLKD2) = 83 MHz
- freq(PCLKD2) = 66 MHz

4 PLL

For PLL's, the following three types are used:

- pll6522x for APLL
- pll6545a for MPLL
- pll6545a for EPLL
- pll6545a for HPLL

Above listed PLL's is integer PLLs.
APLL is especially for D0 clock domain, which has ARM. It can make 50MHz ~ 2GHz, 45:55 duty frequency.
MPLL is especially for D1 clock domain. It supplies clock, 10MHz ~ 600MHz and 40:60 duty, to D1 bus and some peripherals like SPI, HSMMC and etc.
EPLL is mainly used to generate audio clock.
HPLL is mainly used to generate HDMI clock, 74.176MHz and 74.25MHz.

4.1 EXAMPLE PLL PMS VALUE FOR PLL6522X (APLL)

Table 2.3-1. APLL PMS Value

	FIN(MHz)	Target FOUT(MHz)	P	M	S	VCOOUT	FOUT(MHz)
133 MHz	12	133	3	266	3	1064	133.00
	12	267	3	267	2	1068	267.00
	12	400	3	400	2	1600	400.00
	12	533	4	355	1	1065	532.50
	12	667	3	333	1	1332	666.00
	12	800	3	400	1	1600	800.00
	12	933	4	622	1	1866	933.00
	12	1066	3	267	0	1068	1068.00
	12	1200	3	300	0	1200	1200.00
	12	1333	4	445	0	1335	1335.00
	12	1600	3	400	0	1600	1600.00
166 MHz	12	167	3	334	3	1336	167.00
	12	333	4	444	2	1332	333.00
	12	500	3	500	2	2000	500.00
	12	666	3	333	1	1332	666.00
	12	833	3	417	1	1668	834.00
	12	1000	3	500	1	2000	1000.00
	12	1166	4	389	0	1167	1167.00
	12	1333	4	445	0	1335	1335.00
	12	1666	3	417	0	1668	1668.00

4.2 EXAMPLE PLL PMS VALUE FOR PLL6545A (MPLL)

Table 2.3-2. MPLL PMS Value

	FIN(MHz)	Target FOUT(MHz)	P	M	S	FOUT
133 MHz	12	67	3	134	3	67
	12	133	3	133	2	133
	12	200	3	100	1	200
	12	267	2	89	1	267
	12	333	4	111	0	333
	12	400	3	100	0	400
166 MHz	12	83	3	83	2	83
	12	167	6	167	1	167
	12	250	3	125	1	250
	12	333	4	111	0	333
	12	417	4	139	0	417
	12	500	3	125	0	500

4.3 EXAMPLE PLL PMS VALUE FOR PLL6545A (EPLL)

Table 2.3-3. EPLL PMS Value

	FIN(MHz)	Target FOUT(MHz)	P	M	S	FOUT
48M	12	48.0000	3	96	3	48
	12	96.0000	3	96	2	96
	12	144.0000	3	144	2	144
	12	192.0000	3	96	1	192
Audio Clock (for 32K, 44.1K,48K)	12	32.7680	3	131	4	32.75
	12	45.1580	3	90	3	45
	12	49.1520	4	131	3	49.125
	12	67.7376	5	226	3	67.8
	12	73.7280	5	246	3	73.8

4.4 EXAMPLE PLL PMS VALUE FOR PLL6545A (HPLL)

Table 2.3-4. HPLL PMS Value

	FIN(MHz)	Target FOUT(MHz)	P	M	S	FOUT
Video Clock (for 74.25M or 74.176M)	27	54.0000	6	96	3	54
	27	108.0000	6	96	2	108
	27	74.2500	6	132	3	74.25
	27	148.5000	6	132	2	148.5
	27	222.7500	6	99	1	222.75
	27	297.0000	6	132	1	297
	27	371.2500	8	110	0	371.25
	27	445.5000	6	99	0	445.5
	27	74.1758	6	132	3	74.25
	27	148.3516	6	132	2	148.5
	27	222.5275	6	99	1	222.75
	27	296.7033	6	132	1	297
	27	370.8791	11	151	0	370.636
	27	445.0549	6	99	0	445.5
	27	519.2308	9	173	0	519
	27	146.2500	6	130	2	146.25
	27	147.7500	9	197	2	147.25

5 CLOCK GENERATION

Figure 2.3-3 shows a block diagram of the clock generation logic. An external crystal clock is connected to the oscillation amplifier, and the PLL converts the low input frequency into a high-frequency clock required by S5PC100. The clock generator block has a built-in logic to stabilize the clock frequency after each system reset since it takes time before stabilized.

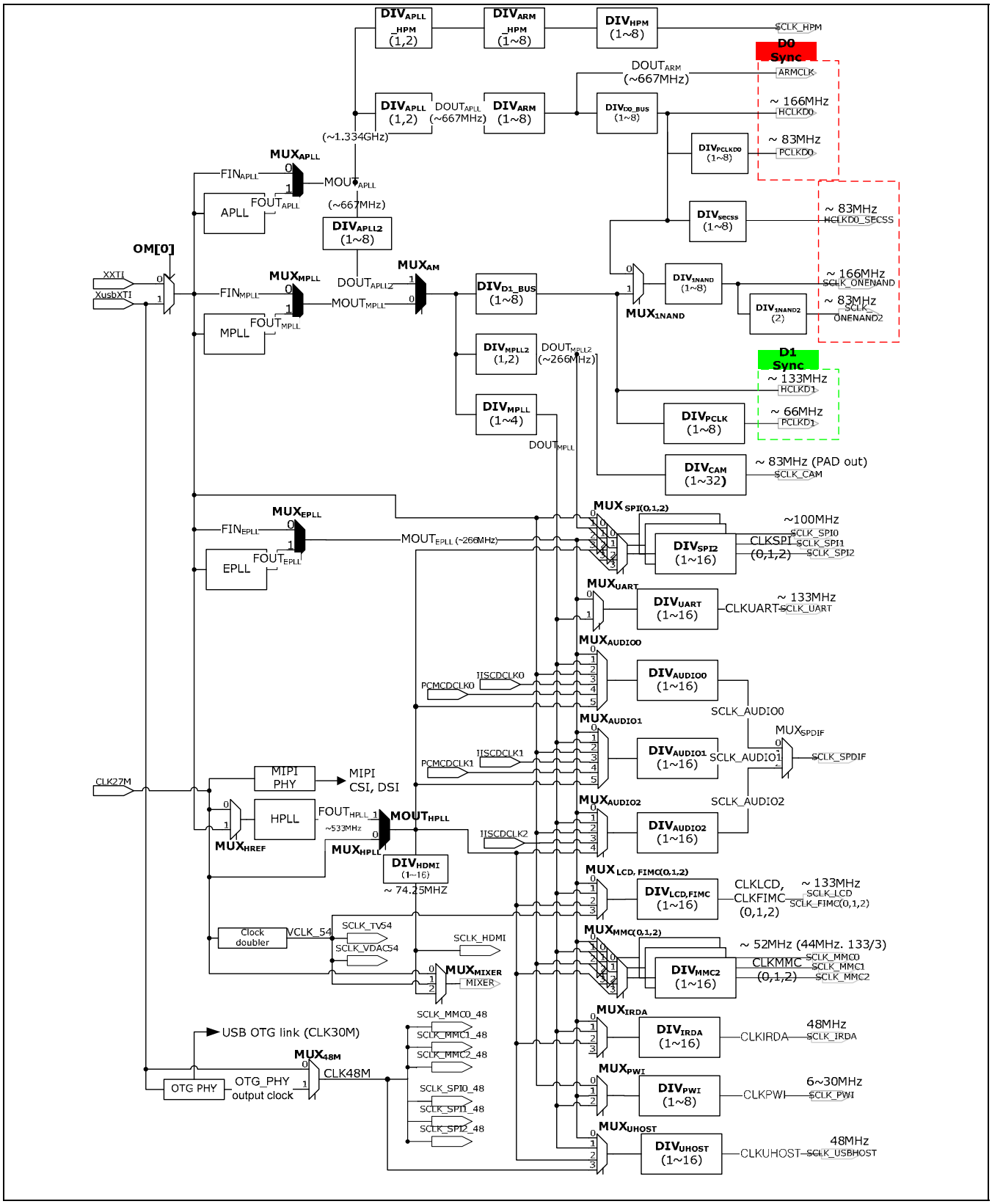


Figure 2.3-3 S5PC100 Clock Generation Circuit1

To run HDMI, set MIXER source clock as $MOUT_{HPLL}$.
DIV_{APLL2} can handle up-to 667MHz input clock. If $DOUT_{APLL2}$ path is used, APLL output clock must be lower than 667MHz.
If select signal of MUX_{48M} is 0, output clock of it may not be 48MHz. XusbXTI clock is bypassed.

Refer Power Management user's manual to use HPM path (DIV_{APLL_HPM}, DIV_{ARM_HPM}, DIV_{HPM}).
Related SFRs are CLKDIV_IEM_L1 ~ CLKDIV_IEM_L8.

Audio_CMU is an additional clock generation circuit in S5PC100. This resides at D2 sub system for low-power MP3 play. For more information refer to "Section 7.2".

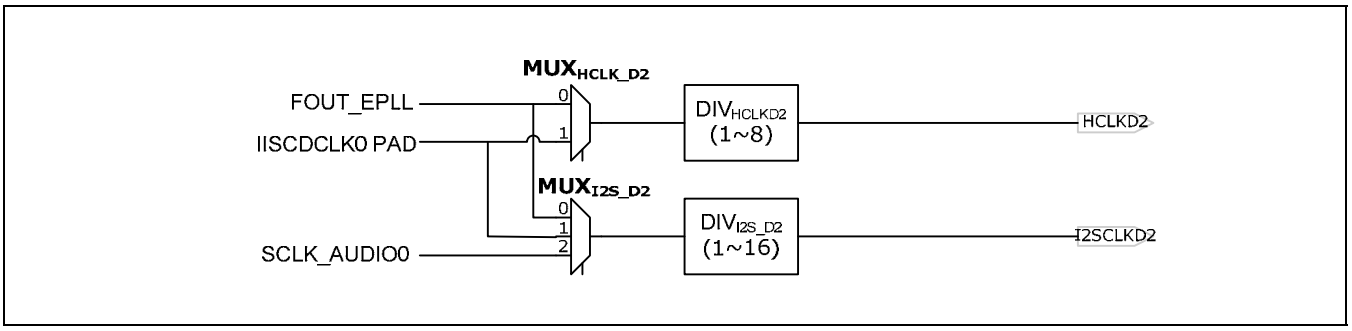


Figure 2.3-4 S5PC100 Clock Generation Circuit2

Table 2.3-5. Bus Clock Domain for Each Module

Bus clock domain	Module
D0	ARM, DRAMC, VIC0, VIC1, VIC2, TZIC0, TZIC1, TZIC2, M2M DMA, G2D, CFCON, CS(Core-sight)System, SEC(Security)SS (AES, DES/TDES, SHA1/PRNG, Secure JTAG, and PKA), IntMem Con, SROMC, ONENANDC, NFCON, EBI, TZPC0, CHIPID
D1	Peripheral DMA, SDMMC0, SDMMC1, SDMMC2, USBHOST1.1, USB OTG, MODEMIF, LCD sub-system (CAMIF0, CAMIF1, CAMIF2, MIPI CSI, MIPI DSI, LCD controller, JPEG, and rotator), TV sub-system (HDMI, SDTV out, VP, and mixer), MFC (Multi-Function Codec), G3D, GPIO, TZPC1 and TZPC2, Clock Management Unit, PWM, System Timer, WDT, RTC, UART, SPI0, SPI1, SPI2, IRDA, I2C0, HDMI_I2C, CAN0, CAN1, MIPI_CSI, MIPI_DSI, MIPI_HSI, I2S1 and I2S2, SPDIF, PCM0, PCM1, AC97, KEYIF, TSADC
D2	I2S0, Large SRAM

PMU (Power Management Unit) is operated by XXTI or XusbXTI (depends on OM[0] pin).

6 CLOCK CONFIGURATION PROCEDURE

Some basic SFR configuration flows are listed below.

1. Turn on a PLL

   (A,M,E,H)PLL_CON[31] = 1; // Power on a PLL (Refer (A, M, E, H)PLL_CON SFR)

   wait_lock_time; **// If you want to wait until PLL clock is out (locking time passed),**

   // check assertion of (A. M, E, H)PLL_CON[30].

   (A, M, E, H)PLL_SEL = 1; // Select the PLL output clock.

   (Refer 0, 4, 8, 12th bit of CLK_SRC0 SFR)

2. Turn off a PLL

   (A,M,E,H)PLL_SEL = 0; // De-select the output of a PLL

   (A,M,E,H)PLL_CON[31] = 0; // Power off the PLL

Must follow above procedure 1, 2

3. Change PLL's PMS values

   Set PMS values; // Set PDIV, MDIV, and SDIV values

   (Refer (A,M,E,H)PLL_CON SFR)

4. Change the system clock divider values

   CLK_DIV0 [31:0] = target value0;

5. Change the divider values for special clocks

   CLK_DIV1 [31:0] = target value1;

   CLK_DIV2 [31:0] = target value2;

Can change PMS value (procedure 3), dividing value (procedure 4, 5) freely without stopping clock.

6.1 CLOCK GATING

S5PC100 can disable the clock operation of each IP if it is not required. This reduces dynamic power.

7 DETAILED CLOCK DESCRIPTION

7.1 CPU AND BUS CLOCK

Cortex A8 supports only **synchronous** mode between CPU and bus (D0 Sync at Figure 2.3-5). D0 and D1 are asynchronous to each other.

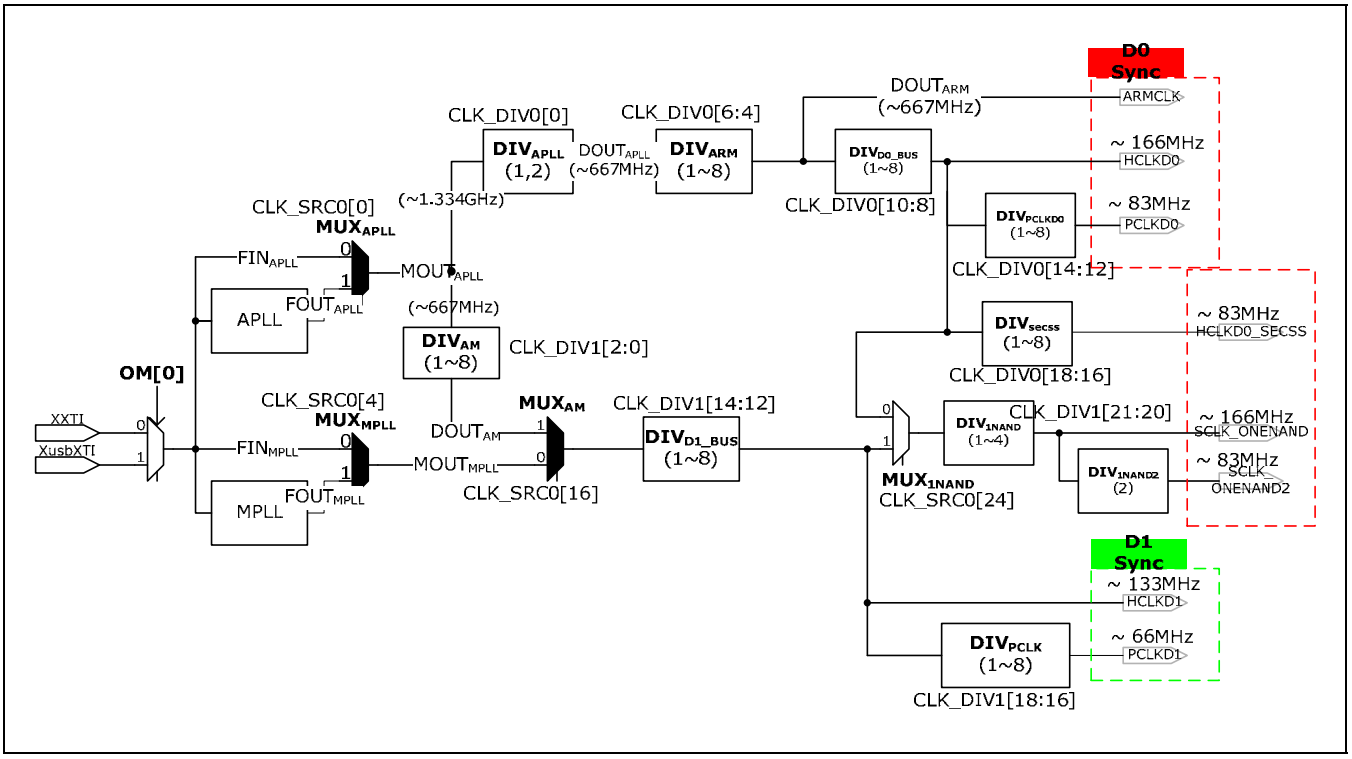


Figure 2.3-5 CPU and Bus Clock

Some clocks can be gated by using SFR.

Table 2.3-6. CPU and Bus Clock

Clock	Related SFR
HCLKD0, PCLKD0	CLK_GATE_D0_0, CLK_GATE_D0_1, CLK_GATE_D0_2
HCLKD0_SECSS	CLK_GATE_D0_0[5]
SCLK_ONENAND, SCLK_ONENAND2	CLK_GATE_SCLK_0[2]
HCLK	CLK_GATE_D1_0, CLK_GATE_D1_1, CLK_GATE_D1_2
PCLK	CLK_GATE_D1_3, CLK_GATE_D1_4, CLK_GATE_D1_5

Caution) ARM clock setting

Clock to ARM (ARMCLK) must be divided when using APLL. It means that at least one of the ARM clock dividers (DIV_{APLL} and DIV_{ARM}) must be set to more than 1 (Possible ARM clock division value = 2, 3, ~ 8, 10, 12, 14, 16).

As shown in Figure 2.3-5, DIV_{APLL} divides up-to 1.334GHz input, DIV_{ARM} and DIV_{APLL2} divides up-to 667MHz input, please consider upper-limit of dividers when setting. If PLL clock is more than 667MHz, you cannot use $DOUT_{AM}$ path. Use $MOUT_{MPLL}$ path for MUX_{AM}.

7.2 LOW POWER MUSIC-PLAY CLOCK

If music is played at D2 clock domain, Audio_CMU module in D2_SS supplies clock for AHB and 5.1ch I2S.

Clock sources of Audio_CMU are EPLL output clock, clock from PAD, and SCLK_AUDIO0 (Refer to "Section 7.10"). Clock source of I2S is dependent to I2S mode. If S5PC100 is operated as I2S master mode, EPLL supplies I2S clock. If S5PC100 is operated as I2S slave mode, external I2S master supplies I2S clock. Bus clocks, HCLKD2 and PCLKD2, are asynchronous to I2S clock. So you can freely select clock source for bus.

When LP (Low-Power) Audio mode with 'top off' is used, SCLK_AUDIO0 must not be selected for I2SCLKD2. SCLK_AUDIO0 is turned off at 'top off' mode.

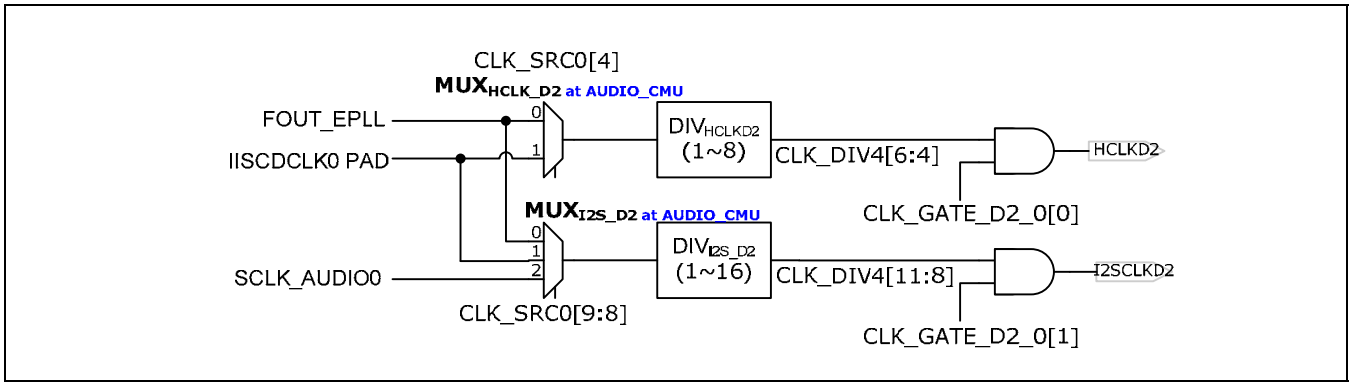


Figure 2.3-6 Audio_CMU in D2_SS

7.3 SPECIAL CLOCK TABLE

The Table 2.3-7 summarizes special clocks in S5PC100.

Table 2.3-7 Special Clock in S5PC100

Name	Description	Input range	Source
SCLK_ONENAND, SCLK_ONENAND_2	OneNAND controller operation clock	~88MHz (ONENAND_2) ~166MHz (ONENAND)	(A, M)PLL
SCLK_IRDA	IrDA operation clock	48MHz fix (+-960Hz margin)	(E,M,H)PLL, USB_48
SCLK_TVENC	SD-video pixel clock	54MHz fix	Clock doubler (27MHz x2)
SCLK_VDAC	SD-video pixel clock	54MHz fix	Clock doubler (27MHz x2)
SCLK_LCD	DISPLAY clock	~133MHz	(E,M,H)PLL, VCLK_54
SCLK_FIMC0, 1, 2	POST function clock	~133MHz	(E,M,H)PLL, VCLK_54
SCLK_HDMI	HD-video pixel clock	27MHz, 74.176MHz, 74.25MHz fix	PAD_27, HPLL
SCLK_AUDIO0	I2S0, PCM0-operation reference clock	2.048MHz ~ 73.7280MHz	CPAD, (E,M,H)PLL, Audio PAD
SCLK_AUDIO1	I2S1, PCM1-operation reference clock	Same as above	Same as above
SCLK_AUDIO2	I2S2-operation clock	Same as above	Same as above
SCLK_SPDIF	SPDIF-operation clock	Same as above	SCLK_AUDIO(0,1,2)
SCLK_SPIx SCLK_SPIx_48	SPI-operation reference clock	~50MHz (100MHz/2) (SPI divides input clock.)	CPAD, (E,M,H)PLL (SCLK_SPIx) / USB_48 (SCLK_SPIx_48)
Uart_eclk SCLK_UART	UART-operation reference clock	~133MHz	From UART PAD / (E,M,H)PLL (SCLK_UART)
SCLK_MMC(0,1) SCLK_MMC(0,1)_48	SDMMC-operation reference clock	~52MHz	(E,M,H)PLL, C_PAD / USB_48 (SCLK_MMCX_48)
SCLK_UHOST1.1	USB HOST-operation reference clock	48MHz	(E,M,H)PLL, USB_48 (OTG PHY)
SCLK_UOTG	USB OTG-operation reference clock	30MHz	USB_48 (OTG PHY)
CAM_PCLK / SCLK_LCD	FIMC-operation clocks (Pixel clock from CIS/ Shared clock with LCD)	~83MHz (CAM_PCLK) ~133MHz (SCLK_LCD)	Direct from CAM clock PAD (CAM_PCLK) / Clock controller (SCLK_LCD)
SCLK_CAM	Clock to CAM	~83MHz	(A, M)PLL
SCLK_MIPI_PHY	Clock for MIPI D-PHY	27MHz	PAD_27
ac_bit_clk	AC97 codec clock (ext chip generates)	12.288MHz fix	Direct from AC97 clock PAD to AC97 IP
pwm_eclk	Auxiliary clock	~70MHz	Direct from PWM clock PAD to PWM IP

CPAD means external crystal or oscillator input for system PLLs or USB OTG PHY.

PAD_27 means 27MHz crystal pad for TV-out.
VCLK_54 is generated from clock-doubler.

Audio pad means I2S and PCM codec clock.

USB_48 means 48MHz clock from USB OTG PHY.
CAM means Camera including CIS or CCD.

7.4 MEMORY CLOCK

Memory subsystem needs special clocks just for an OneNAND-controller (SCLK_ONENAND, SCLK_ONENAND2). SCLK_ONENAND and SCLK_ONAND2 are OneNAND-operational clocks. These special clocks are synchronous to HCLKD0 when clock source of SCLK_ONENAND and SCLK_ONENAND2 is HCLKD0. If selected pin of MUX1nand is 0, OneNAND operates HCLKD0-synchronously. If pin is 1, OneNAND operates HCLKD0-asynchronously.

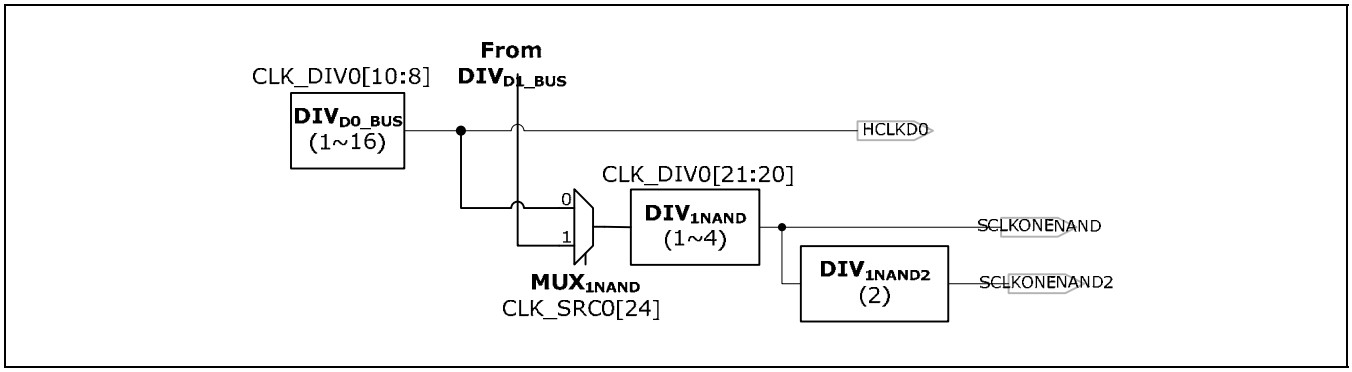


Figure 2.3-7 Clock Subsystem for ONENAND

7.5 TV CLOCK

S5PC100 gets 27MHz-clock source to make 54MHz-clock for SDTV out. Internal clock-doubler doubles 27MHz-clock, and then passes 54MHz-clock to VDAC and TV out module.

HDMI needs 27, 74.176, and 74.25 MHz clock. 74.176 and 74.25MHz are usually made by HPLL (HDMI PLL).

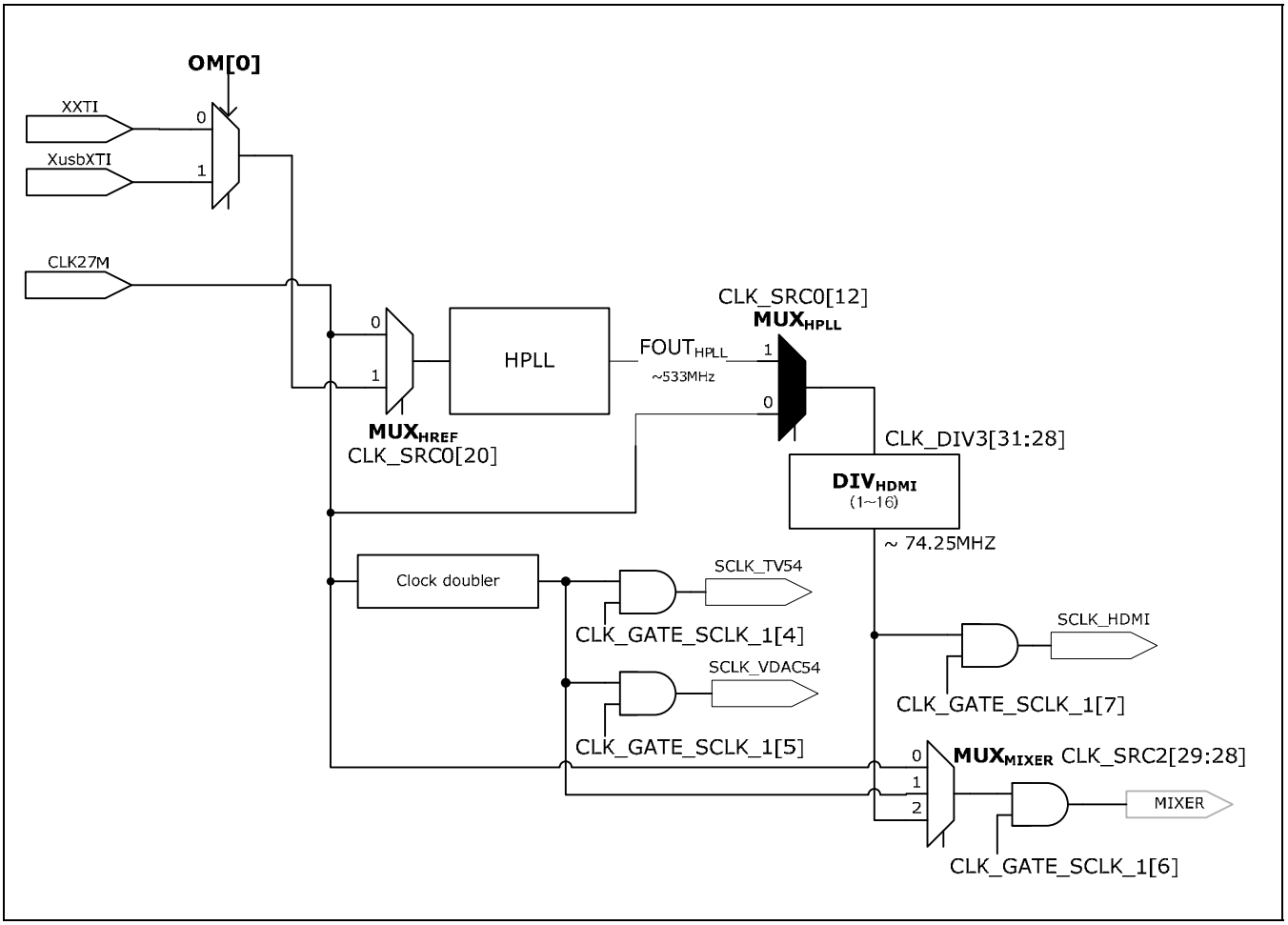
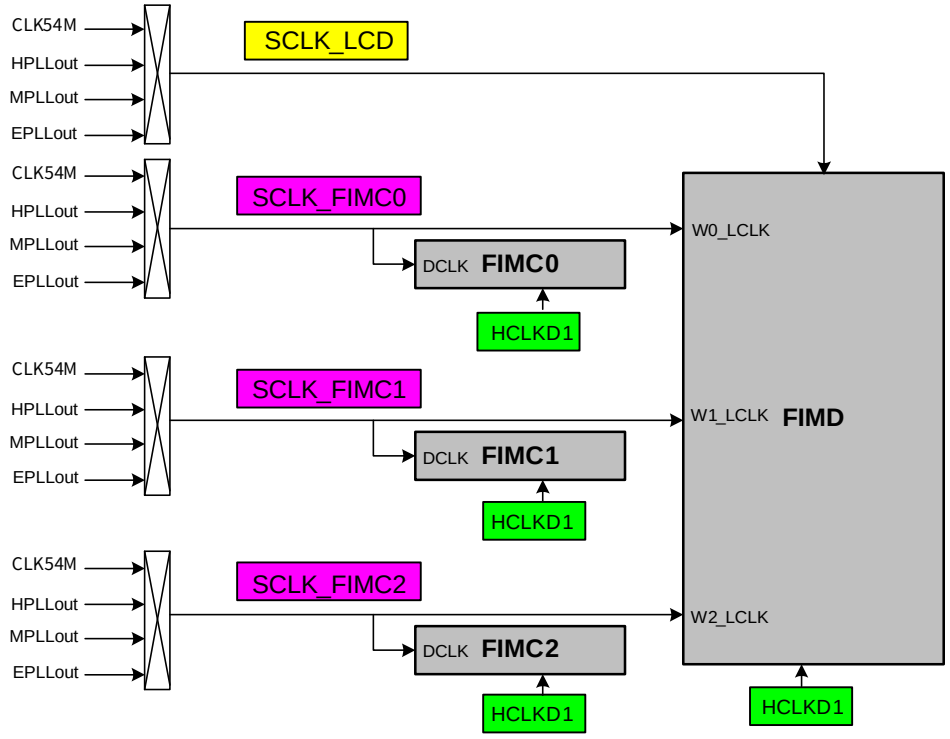


Figure 2.3-8 TV Clock

7.6 FIMD AND FIMC



FIMC DCLK is not used in DMA write mode

Figure 2.3-9 FIMD and FIMC clock

FIMD (LCD controller) and three FIMCs (Camera interface) have many clock sources. Figure 2.3-109 shows overall clock path and registers to set. SCLK_FIMCn is the local interface clock between FIMCn and FIMD.

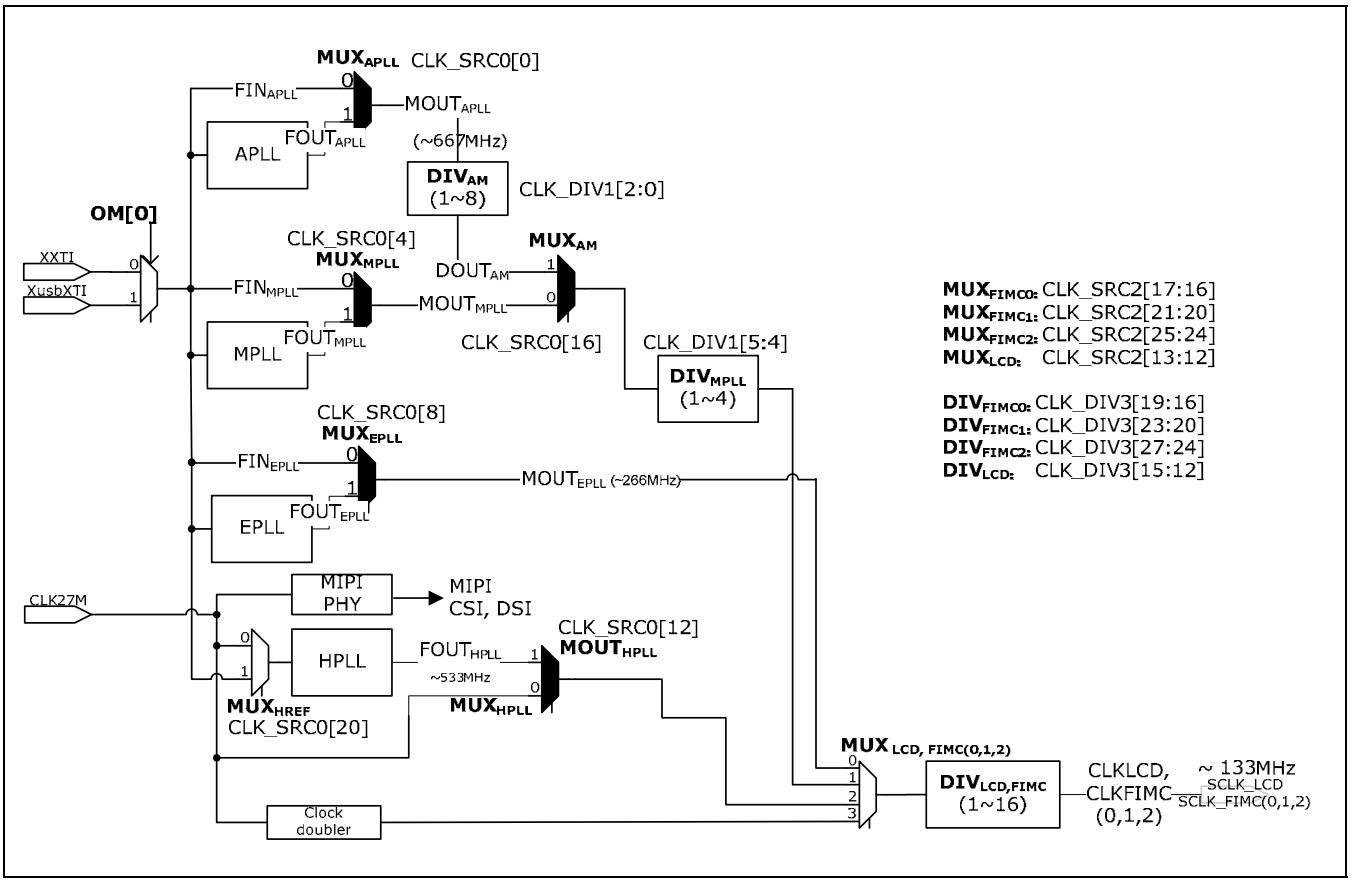


Figure 2.3-10 LCD Subsystem Clock (for Setting)

Use SFR to gate Special clocks.

Table 2.3-8. LCD and FIMC Clock Gating SFR

Clock	Related SFR
SCLK_FIMC0	CLK_GATE_SCLK_1[1]
SCLK_FIMC1	CLK_GATE_SCLK_1[2]
SCLK_FIMC2	CLK_GATE_SCLK_1[3]
SCLK_LCD	CLK_GATE_SCLK_1[0]

7.7 SECURE SUB-SYSTEM CLOCK

Modules in Secure Sub-System Clock (SECSS) have long combinational paths because secure algorithm needs **intensive computation**. Those modules use divided clock (HCLKD0_SECSS) derived from HCLKD0 for that long path. HCLKD0_SECSS must not be higher than 83MHz.
AES, DES/TDES, SHA1/PRNG, Secure JTAG, PKA are modules in SECSS.

7.8 USB PHY AND 48MHZ CLOCK

S5PC100 has an USB PHY for USB OTG. It has an internal PLL to generate clock for USB-OTG operation. Internal PLL must get definite clock-frequency (OSC 12, 24, and 48MHz / crystal 12MHz) through XusbXTI and XusbXTO.

An OTG PHY generates 48 MHz clock. SDMMC, SPI need 48MHz clock and **IrDA, USB HOST1.1 needs exact 48MHz** clock. S5PC100 supplies this clock from XusbXTI (when using 48MHz crystal or OSC) or output of OTG PHY or PLL.

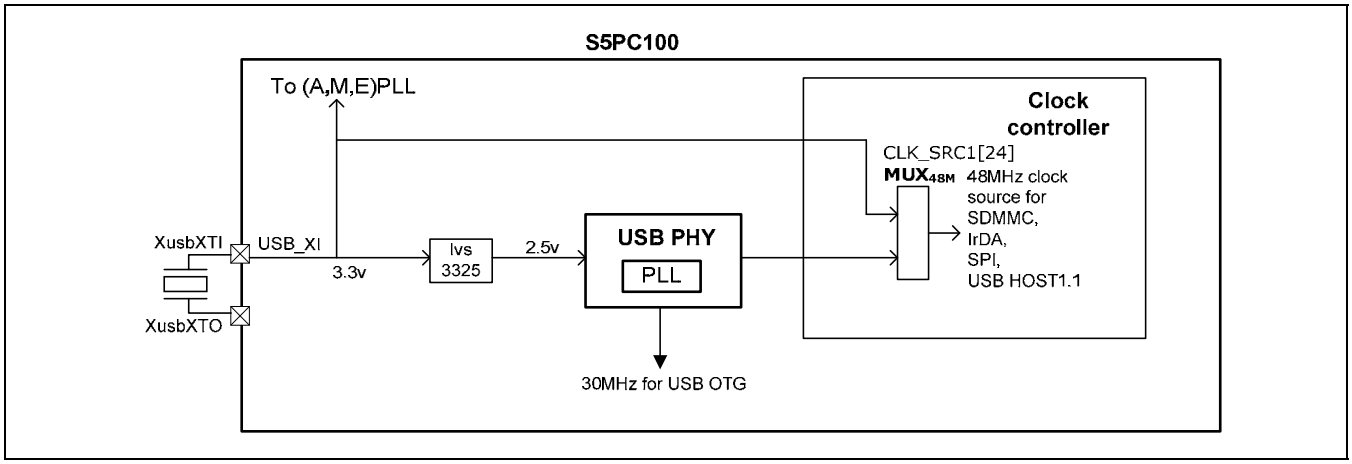


Figure 2.3-11 USB PHY and 48MHz clock

7.9 D2_SS CLOCK DOMAIN

For more information on Audio CMU in Figure 2.3-12, refer to "Section 7.2".

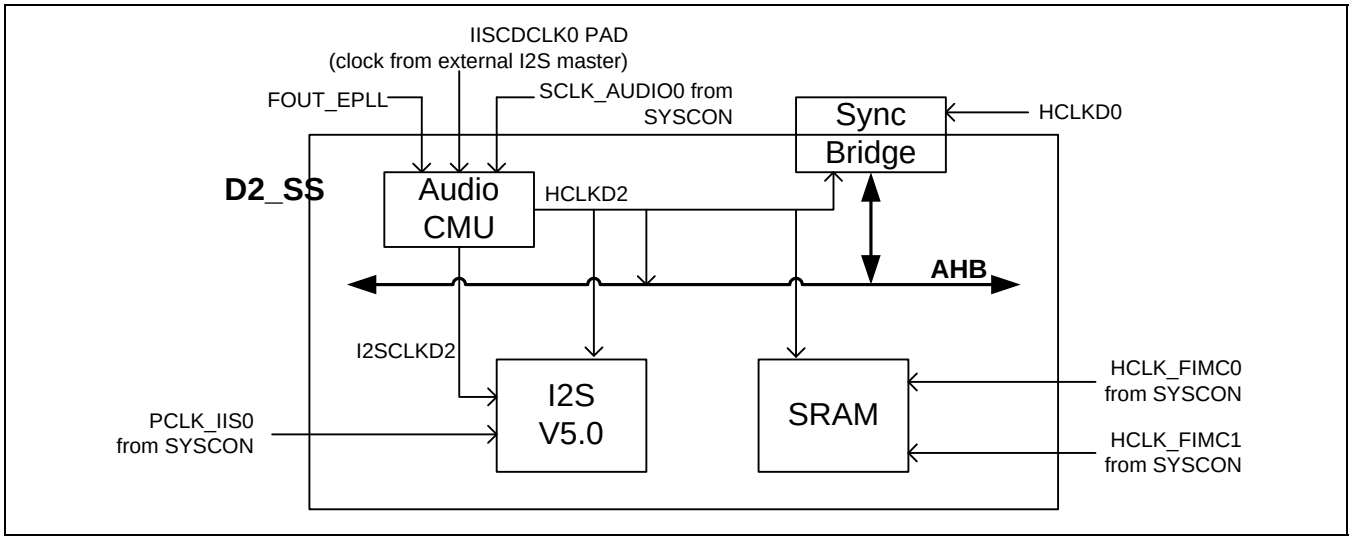


Figure 2.3-12 D2_SS Clock Domain

7.10 AUDIO CLOCK

S5PC100 has three I2S (one for 5.1ch), two PCM, one AC97, and one SPDIF. Clock for those modules can be PLL output clock or input clock from PAD.

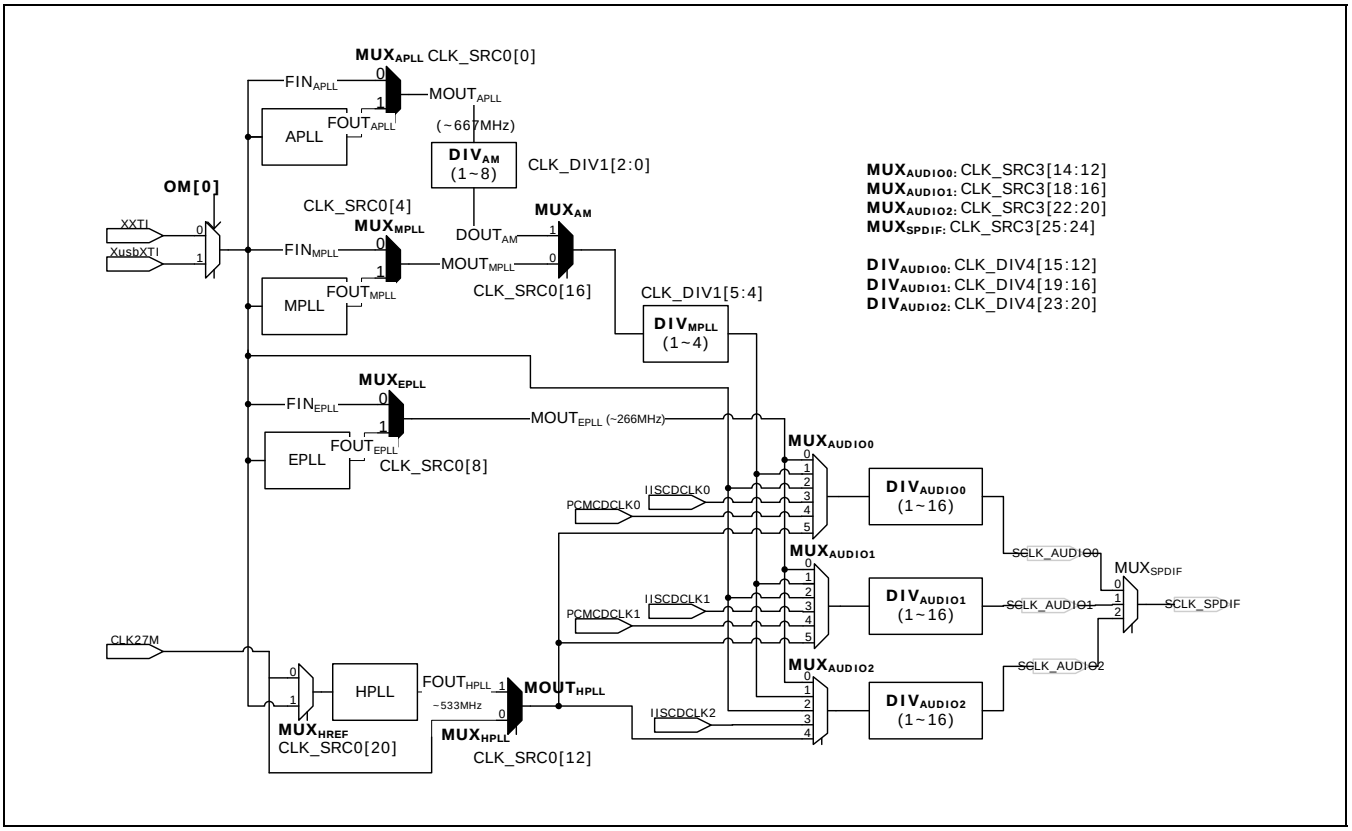


Figure 2.3-13 Clock source for audio peripherals

SCLK_AUDIO0 for I2S0 (5.1ch I2S in D2_SS, also refer Figure 2.3-12), PCM0

SCLK_AUDIO1 for I2S1, PCM1

SCLK_AUDIO2 for I2S2

SCLK_SPDIF for SPDIF

Special clock for AC97 is made by external codec chip. It is passed through GPIO PAD.

Special clocks can be gated by using SFR.

Table 2.3-9. Audio Clock Gating SFR

Clock	Related SFR
SCLK_AUDIO0	CLK_GATE_SCLK_1[8]
SCLK_AUDIO1	CLK_GATE_SCLK_1[9]
SCLK_AUDIO2	CLK_GATE_SCLK_1[10]
SCLK_SPDIF	CLK_GATE_SCLK_1[11]

7.11 MIPI CLOCK

MIPI D-PHY has an internal PLL which generates MIPI clock for high-speed transmission with meeting tight jitter spec. Input clock for that PLL is TV reference clock (27MHz).

7.12 PERIPHERAL

S5PC100x has many kinds of peripherals to communicate with external device, such as USB, MMC, UART, and etc. Figure 2.3-14 shows clock control path of peripherals.

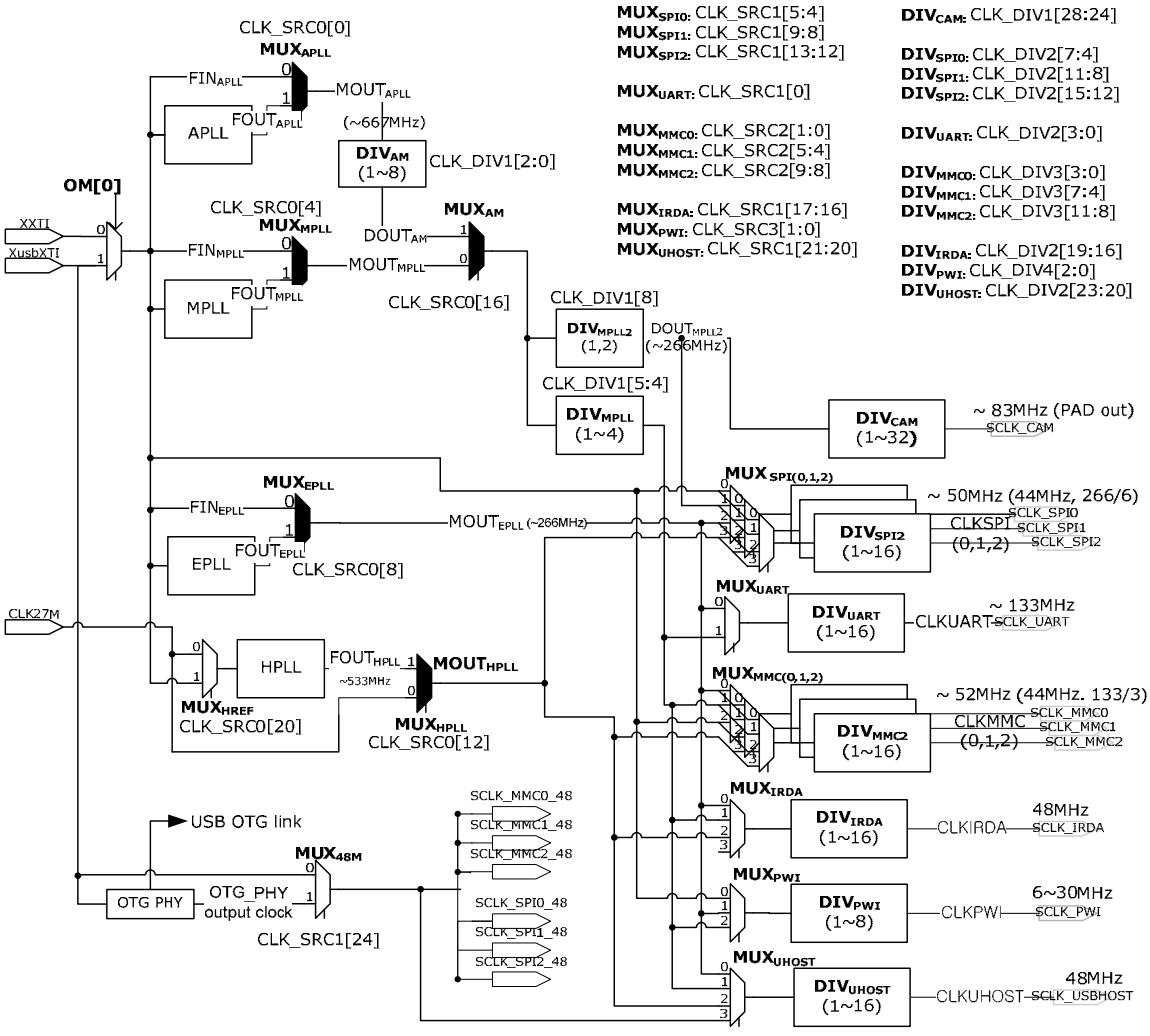


Figure 2.3-14 Peripheral Clock

Use SFR to gate Special clocks.

Table 2.3-10. Peripheral Clock Gating SFR

Clock	Related SFR
SCLK_CAM	CLK_GATE_SCLK_0[12]
SCLK_SPI0, SCLK_SPI1, SCLK_SPI2	CLK_GATE_SCLK_0[4], CLK_GATE_SCLK_0[5], CLK_GATE_SCLK_0[6]
SCLK_UART	CLK_GATE_SCLK_0[3]
SCLK_MMC0, SCLK_MMC1, SCLK_MMC2	CLK_GATE_SCLK_0[12], CLK_GATE_SCLK_0[13], CLK_GATE_SCLK_0[14]
SCLK_IRDA	CLK_GATE_SCLK_0[10]
SCLK_PWI	CLK_GATE_SCLK_0[1]
SCLK_USBHOST	CLK_GATE_SCLK_0[11]
SCLK_MMC0_48, SCLK_MMC1_48, SCLK_MMC2_48	CLK_GATE_SCLK_0[15], CLK_GATE_SCLK_0[16], CLK_GATE_SCLK_0[17]
SCLK_SPI0_48, SCLK_SPI1_48, SCLK_SPI2_48	CLK_GATE_SCLK_0[7], CLK_GATE_SCLK_0[8], CLK_GATE_SCLK_0[9]

8 I/O DESCRIPTION

Funtion Signal	I/O	Description	Pad	Type
CLKOUT	Out	Clock monitoring pad	XCLKOUT	Dedicated
XTI & XTO	Input & Output	Oscillator pads for system clock	XXTI & XXTO	Dedicated
USB_XTI & USB_XTO	Input & Output	Oscillator pads for USB clock or system clock	XusbXTI & XusbXTO	Dedicated
27M_XTI & 27M_XTO	Input & Output	Oscillator pads for TV clock	X27mXTI & X27mXTO	Dedicated

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals. When a clock pad is not used, the input clock pad should be connected to GND.

8.1 DRIVER STRENGTH OF OSCILLATOR PAD

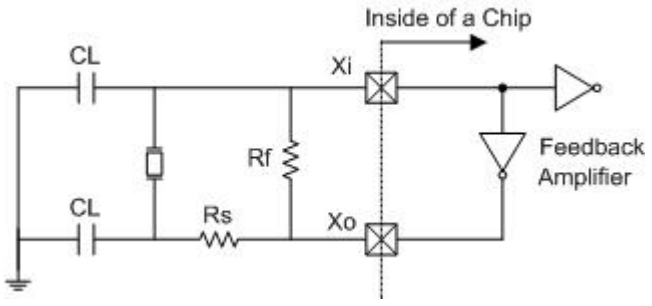
Driver strength of oscillation pads (XT, XT_USB, and XT_27M) must be set rightly, according to input frequency.

DRV value	Output	Freq
2'b00	Driver X1	32kHz ~ 1MHz
2'b01	Driver X2	1MHz ~ 15MHz
2'b10	Driver X3	15MHz ~ 30MHz
2'b11	Driver X4	30MHz ~ 50MHz

To get more information, refer to "ETC4DRV SFR in 02.02 S5PC100_GPIO".

8.2 CRYSTAL OSCILLATOR DESIGN CONSIDERATION

The external components commonly used for the oscillator circuit are a resonator (quarts crystal), two capacitors CL and two resistors Rf and Rs, as shown in the below figure.



XXTI needs 12MHz crystal oscillator. XusbXTI also needs 12MHz crystal oscillator to be used as main system clock by the setting of OM[0]=1. XXTI27 needs 27MHz crystal oscillator.

For 12MHz or 27MHz crystal oscillator, the following values of external components are recommended;

Rs=0ohm, Rf=1Mohm, CL=10~35pF

9 REGISTER DESCRIPTION

System controller controls PLL, clock generator, the power management part, and other system dependent part. This section describes how to control these parts using Special Functional Register (SFR) within the system controller.

9.1 (CLOCK) REGISTER MAP – CLOCK CONTROLLER

Register	Address	R/W	Description	Reset Value
1.1 PLL lock	**0xE010_0000**			
APLL_LOCK	0xE010_0000	R/W	Control PLL locking period for APLL	0x0000_FFFF
MPLL_LOCK	0xE010_0004	R/W	Control PLL locking period for MPLL	0x0000_FFFF
EPLL_LOCK	0xE010_0008	R/W	Control PLL locking period for EPLL	0x0000_FFFF
HPLL_LOCK	0xE010_000C	R/W	Control PLL locking period for HPLL	0x0000_FFFF
Reserved	0xE010_0010 ~ 0xE010_00FC		Reserved	0x0000_0000
1.2 PLL control	**0xE010_0100**			
APLL_CON	0xE010_0100	R/W	Control PLL output frequency for APLL	0x0190_0302
MPLL_CON	0xE010_0104	R/W	Control PLL output frequency for MPLL	0x0085_0302
EPLL_CON	0xE010_0108	R/W	Control PLL output frequency for EPLL	0x0085_0302
HPLL_CON	0xE010_010C	R/W	Control PLL output frequency for HPLL	0x0085_0302
Reserved	0xE010_0110 ~ 0xE010_01FC		Reserved	0x0000_0000
1.3 Clock source	**0xE010_0200**			
CLK_SRC0	0xE010_0200	R/W	Select clock source 0 (Main)	0x0000_0000
CLK_SRC1	0xE010_0204	R/W	Select clock source 1 (Connectivity)	0x0000_0000
CLK_SRC2	0xE010_0208	R/W	Select clock source 2 (Multimedia & Connectivity)	0x0000_0000
CLK_SRC3	0xE010_020C	R/W	Select clock source 3 (Audio & Others)	0x0000_0000
Reserved	0xE010_0210 ~ 0xE010_02FC		Reserved	0x0000_0000
1.4 Clock divider	**0xE010_0300**			
CLK_DIV0	0xE010_0300	R/W	Set clock divider ratio 0 (Main D0 domain)	0x0001_1000
CLK_DIV1	0xE010_0304	R/W	Set clock divider ratio 1 (Main D1 domain)	0x0001_0000
CLK_DIV2	0xE010_0308	R/W	Set clock divider ratio 2 (Connectivity)	0x0000_0000
CLK_DIV3	0xE010_030C	R/W	Set clock divider ratio 3 (Multimedia & Connectivity)	0x0000_0000
CLK_DIV4	0xE010_0310	R/W	Set clock divider ratio 4 (Audio & Others)	0x0000_0000
Reserved	0xE010_0318 ~		Reserved	0x0000_0000

Register	Address	R/W	Description	Reset Value
	0xE010_03FC			
1.5 Clock output	**0xE010_0400**			
CLK_OUT	0xE010_0400	R/W	Select clock output	0x0000_0000
Reserved	0xE010_0404 ~ 0xE010_04FC		Reserved	0x0000_0000
1.6 Clock gating	**0xE010_0500**			
CLK_GATE_D0_0	0xE010_0500	R/W	Control HCLKD0 / PCLKD0 clock gating 0 (System1)	0x0000_007F
CLK_GATE_D0_1	0xE010_0504	R/W	Control HCLKD0 / PCLKD0 clock gating 1 (Memory)	0x0000_003F
CLK_GATE_D0_2	0xE010_0508	R/W	Control PCLKD0 clock gating 2 (System2)	0x0000_0007
Reserved	0xE010_050C ~ 0xE010_051C		Reserved	0x0000_0000
CLK_GATE_D1_0	0xE010_0520	R/W	Control HCLKD1 / PCLKD1 clock gating 0 (File)	0x0000_00FF
CLK_GATE_D1_1	0xE010_0524	R/W	Control HCLKD1 / PCLKD1 clock gating 1 (Multimedia1)	0x0000_01FF
CLK_GATE_D1_2	0xE010_0528	R/W	Control HCLKD1 / PCLKD1 clock gating 2 (Multimedia2)	0x0000_001F
CLK_GATE_D1_3	0xE010_052C	R/W	Control HCLKD1 / PCLKD1 clock gating 1 (System)	0x0000_03FF
CLK_GATE_D1_4	0xE010_0530	R/W	Control HCLKD1 / PCLKD1 clock gating 2 (Connectivity)	0x0000_3FFF
CLK_GATE_D1_5	0xE010_0534	R/W	Control HCLKD1 / PCLKD1 clock gating 2 (Audio)	0x0000_03FF
Reserved	0xE010_0538 ~ 0xE010_053C		Reserved	0x0000_0000
CLK_GATE_D2_0	0xE010_0540	R/W	Control HCLKD2 clock gating 0 (Audio)	0x0000_0000
Reserved	0xE010_0544 ~ 0xE010_055C		Reserved	0x0000_0000
CLK_GATE_SCLK_0	0xE010_0560	R/W	Control SCLK clock gating (System)	0x0003_FFFF
CLK_GATE_SCLK_1	0xE010_0564	R/W	Control SCLK clock gating (Multimedia)	0x0000_1FFF
Reserved	0xE010_0568 ~ 0xE010_7FFC		Reserved	0x0000_0000

9.2 (OHTERS) REGISTER MAP – CLOCK CONTROLLER

Register	Address	R/W	Description	Reset Value
2.1 SW RESET registers	**0xE020_0000**			
SWRESET	0xE020_0000	R/W	Generate software reset	0x0000_0000
Reserved	0xE020_0004	R/W	Reserved	0x0000_0000
ONENAND_SWRESET	0xE020_0008	R/W	OneNAND controller setting	0x0000_0000
Reserved	0xE020_000C ~ 0xE020_00FC	R/W	Reserved	0x0000_0000
2.2 CPU registers	**0xE020_0100**			
GENERAL_CTRL	0xE020_0100	R/W	General Control Register	0x0000_0000
GENERAL_STATUS	0xE020_0104	R	General Status Register	0x0000_0000
Reserved	0xE020_0108 ~ 0xE020_01FC		Reserved	0x0000_0000
2.3 Memory registers	**0xE020_0200**			
MEM_SYS_CFG	0xE020_0200	R/W	ENDIAN & EBI configuration	0x0000_0000
Reserved	0xE020_0204 ~ 0xE020_02FC		Reserved	0x0000_0000
2.4 Multimedia registers	**0xE020_0300**			
CAM_MUX_SEL	0xE020_0300	R/W	Camera mapping to FIMC selection	0x0000_0000
MIXER_OUT_SEL	0xE020_0304	R/W	Video Mixer output to TVENC / HDMI selection	0x0000_0000
LPMP3_MODE_SEL	0xE020_0308	R/W	Low power MP3 mode selection	0x0000_0000
Reserved	0xE020_030C ~ 0xE020_03FC	R/W	Reserved	0x0000_0000
2.5 Test registers	**0xE020_0400**			
MIPI_PHY_CON0	0xE020_0400	R/W	MIPI D-PHY control register0	0x0000_0000
Reserved	0xE020_0404 ~ 0xE020_0410		Reserved	0x0000_0000
MIPI_PHY_CON1	0xE020_0414	R/W	MIPI D-PHY control register1	0x0000_0000
Reserved	0xE020_0418 ~ 0xE020_041C	R/W	Reserved	0x0000_0000
HDMI_PHY_CON0	0xE020_0420		HDMI PHY control register0	0x0000_0000
Reserved	0xE020_0424 ~ 0xE02F_FFFC		Reserved	0x0000_0000

SFRs consist of three parts. The SFRs, whose address are 0xE010_0XXX, controls clock-related logics. They control the output frequency of three PLLs, clock source selection, clock divider ratio and clock gating. The SFRs, whose address is 0xE010_8XXX, control the power management block. The SFRs, whose address is 0xE02_0XX, has some miscellaneous information.

10 INDIVIDUAL REGISTER DESCRIPTIONS

10.1 PLL CONTROL REGISTERS

S5PC100 has four internal PLLs, which are APLL, MPLL, EPLL, and HPLL. They are controlled by the following eight special registers.

10.1.1 PLL Lock

- Control PLL locking period for APLL (APLL_LOCK, R/W, Address = 0xE010_0000)
- Control PLL locking period for MPLL (MPLL_LOCK, R/W, Address = 0xE010_0004)
- Control PLL locking period for EPLL (EPLL_LOCK, R/W, Address = 0xE010_0008)
- Control PLL locking period for HPLL (HPLL_LOCK, R/W, Address = 0xE010_000C)
- RESERVED (RESERVED, Address = 0xE010_0010 ~ 0xE010_00FC)

A PLL requires locking period if input frequency is changed or frequency division (multiplication) values are changed. PLL_LOCK register specifies this locking period, which is based on PLL's source clock. During this period, PLL output will be masked with '0'.

APLL_LOCK / MPLL_LOCK / EPLL_LOCK / HPLL_LOCK	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0x0000
PLL_LOCKTIME	[15:0]	Required input clock count needed to generate a stable PLL output	0xFFFF

NOTE:
Max lock time in A/M/E/HPLL is 300usec. The value of PLL_LOCKTIME is based on input clock count. For example, when input clock frequency is 12MHz and lock time is 300us, the value of PLL_LOCKTIME will be 3600(=0xE10).

10.1.2 Control PLL Output Frequency for APLL (APLL_CON, R/W, Address = 0xE010_0100)

PLL_CON register controls the operation of each PLL. If ENABLE bit is set, the corresponding PLL generates output after PLL locking period. The output frequency of PLL is controlled by the MDIV, PDIV, and SDIV values.

APLL_CON	Bit	Description	Reset Value
ENABLE	[31]	PLL enable control (0: Disables, 1: Enables)	0
LOCKED	[30]	PLL is locked after locking time (locking time is set at PLL_LOCK SFR)	0
Reserved	[29:26]	Reserved	0
MDIV	[25:16]	PLL M divide value	0x190
Reserved	[15:14]	Reserved	0
PDIV	[13:8]	PLL P divide value	0x3
Reserved	[7:3]	Reserved	0
SDIV	[2:0]	PLL S divide value	0x2

The reset value of APLL_CON generates 400MHz output clock respectively, if the input clock frequency is 12MHz.

NOTE:

The output frequency is calculated by the following equation:

$F_{OUT} = MDIV \times F_{IN} / (PDIV \times 2^{SDIV})$

where, MDIV, PDIV, SDIV for APLL and MPLL must meet the following conditions :

PDIV: $1 \le PDIV \le 63$

MDIV: $64 \le MDIV \le 1023$

SDIV: $0 \le SDIV \le 5$

F_{ref} $(=F_{IN} / PDIV)$: $3MHz \le F_{ref} \le 6MHz$

F_{VCO} $(=MDIV \times F_{IN} / PDIV)$: $1000MHz \le F_{VCO} \le 2000MHz$

F_{OUT}: $50MHz \le F_{VCO} \le 2000MHz$

Ex) For FOUT=500MHz, P=3, M=500, S=2

10.1.3 Control PLL Output Frequency

- Control PLL output frequency for MPLL (MPLL_CON, R/W, Address = 0xE010_0104)
- Control PLL output frequency for EPLL (EPLL_CON, R/W, Address = 0xE010_0108)
- Control PLL output frequency for HPLL (HPLL_CON, R/W, Address = 0xE010_010C)
- RESERVED (RESERVED, Address = 0xE010_0110, Address = 0xE010_01FC)

MPLL_CON / HPLL_CON / EPLL_CON	Bit	Description	Reset Value
ENABLE	[31]	PLL enable control (0: Disables, 1: Enables)	0
LOCKED	[30]	PLL is locked after locking time (locking time is set at PLL_LOCK SFR)	0
RESERVED	[29:24]	Do not change	0
MDIV	[23:16]	PLL M divide value	0x85
Reserved	[15:14]	Reserved	0
PDIV	[13:8]	PLL P divide value	0x3
Reserved	[7:3]	Reserved	0
SDIV	[2:0]	PLL S divide value	0x2

The reset value of MPLL_CON, EPLL_CON, and HPLL_CON generates 133MHz output clock respectively, if the input clock frequency is 12MHz.

NOTE:

The output frequency is calculated by the following equation:

$F_{OUT} = MDIV \times F_{IN} / (PDIV \times 2^{SDIV})$

where, MDIV, PDIV, SDIV for PLLs must meet the following conditions:

PDIV: $1 \le PDIV \le 63$

MDIV: $16 \le MDIV \le 255$

SDIV: $0 \le SDIV \le 5$

F_{ref} ($=F_{IN} / PDIV$): $2MHz \le F_{ref} \le 6MHz$

F_{VCO} ($=MDIV \times F_{IN} / PDIV$): $300MHz \le F_{VCO} \le 600MHz$

F_{OUT} : $10MHz \le F_{OUT} \le 600MHz$

Ex) For F_{OUT}=266MHz, P=3, M=133, S=1

10.2 CLOCK SOURCE CONTROL REGISTER

S5PC100 has many clock sources, which include four PLL outputs, the external oscillator, the external clock, and other clock sources from GPIO. CLK_SRC0, 1, 2, and 3 registers control the source clock of each clock divider.

10.2.1 Select Clock Source 0 (Main) (CLK_SRC0, R/W, Address = 0xE010_0200)

Main clock source register

CLK_SRC0	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	
ONENAND_SEL	[24]	Control MUX_{1NAND} (0:HCLKD0, 1:HCLKD1)	0x0
Reserved	[23:21]	Reserved	
HREF_SEL	[20]	Control MUX_{HREF} (0:FIN_{27M}, 1:SRCLK)	0
Reserved	[19:17]	Reserved	
AMMUX_SEL	[16]	Control MUX_{AM} (0:$MOUT_{MPLL}$, 1:$DOUT_{APLL2}$)	0x0
Reserved	[15:13]	Reserved	
HPLLSEL	[12]	Control MUX_{HPLL} (0:CLK27M, 1: $FOUT_{HPLL}$)	0x0
Reserved	[11:9]	Reserved	
EPLL_SEL	[8]	Control MUX_{EPLL} (0:FIN_{EPLL}, 1:$FOUT_{EPLL}$)	0
Reserved	[7:5]	Reserved	
MPLL_SEL	[4]	Control MUX_{MPLL} (0:FIN_{MPLL}, 1:$FOUT_{MPLL}$)	0
Reserved	[3:1]	Reserved	
APLL_SEL	[0]	Control MUX_{APLL} (0:FIN_{APLL}, 1:$FOUT_{APLL}$)	0

10.2.2 Select Clock Source 1 (Connectivity) (CLK_SRC1, R/W, Address = 0xE010_0204)

Connectivity clock source register

CLK_SRC1	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	
CLK48M_SEL	[24]	Control MUX_{48M}, which is the source clock of 48M (0:XusbXTI, 1:OTG_PHY output clock)	0x0
Reserved	[23:22]	Reserved	
UHOST1.1_SEL	[21:20]	Control $MUX_{UHOST,}$ which is the source clock of UHOST1.1 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: $MOUT_{HPLL}$, 11: 48MHz)	0x0
Reserved	[19:18]	Reserved	
IRDA_SEL	[17:16]	Control MUX_{IRDA}, which is the source clock of IRDA (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: $MOUT_{HPLL}$, 11: 48MHz)	0x0
Reserved	[15:14]	Reserved	
SPI2_SEL	[13:12]	Control MUX_{SPI2}, which is the source clock of SPI2 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL2}$, 10: FIN_{EPLL}, 11: $MOUT_{HPLL}$)	0x0
Reserved	[11:10]	Reserved	
SPI1_SEL	[9:8]	Control MUX_{SPI1}, which is the source clock of SPI1 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL2}$, 10: FIN_{EPLL}, 11: $MOUT_{HPLL}$)	0x0
Reserved	[7:6]	Reserved	
SPI0_SEL	[5:4]	Control MUX_{SPI0}, which is the source clock of SPI0 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL2}$, 10: FIN_{EPLL}, 11: $MOUT_{HPLL}$)	0x0
Reserved	[3:1]	Reserved	
UART_SEL	[0]	Control MUX_{UART}, which is the source clock of UART (0:$MOUT_{EPLL}$, 1: $DOUT_{MPLL}$)	0

10.2.3 Select Clock Source 2 (Multimedia and Connectivity) (CLK_SRC2, R/W, Address = 0xE010_0208)

Multimedia and Connectivity clock source register.

CLK_SRC2	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	
MIXER_SEL	[29:28]	Control MUX_{MIXER}, which is the source clock of MIXER (00:CLK27M, 01: VCLK_54, 10: $MOUT_{HPLL}$) To run HDMI, set MIXER clock source as 10 ($MOUT_{HPLL}$)	0x0
Reserved	[27:26]	Reserved	
FIMC2_SEL	[25:24]	Control MUX_{FIMC2}, which is the source clock of FIMC2 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: $MOUT_{HPLL,}$ 11: VCLK_54)	0x0
Reserved	[23:22]	Reserved	
FIMC1_SEL	[21:20]	Control MUX_{FIMC1}, which is the source clock of FIMC1 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: $MOUT_{HPLL,}$ 11: VCLK_54)	0x0
Reserved	[19:18]	Reserved	
FIMC0_SEL	[17:16]	Control MUX_{FIMC0}, which is the source clock of FIMC0 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: $MOUT_{HPLL,}$ 11: VCLK_54)	0x0
Reserved	[15:14]	Reserved	
LCD_SEL	[13:12]	Control MUX_{LCD}, which is the source clock of LCD (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: $MOUT_{HPLL,}$ 11: VCLK_54)	0x0
Reserved	[11:10]	Reserved	
MMC2_SEL	[9:8]	Control MUX_{MMC2}, which is the source clock of MMC2 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL}, 11: $MOUT_{HPLL}$)	0x0
Reserved	[7:6]	Reserved	
MMC1_SEL	[5:4]	Control MUX_{MMC1}, which is the source clock of MMC2 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL}, 11: $MOUT_{HPLL}$)	0x0
Reserved	[3:2]	Reserved	
MMC0_SEL	[1:0]	Control MUX_{MMC0}, which is the source clock of MMC2 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL})	0x0

Restriction: 54MHz (VCLK_54) clock is used for FIMCs and LCD, pass clock at least one among SCLK_HDMI, SCLK_MIXER, and SCLK_VDAC54. (Refer CLK_GATE_SCLK_1 SFR)

10.2.4 Select Clock Source 3 (Audio and Others) (CLK_SRC3, R/W, Address = 0xE010_020C)

Audio and Others clock source register

CLK_SRC3	Bit	Description	Reset Value
Reserved	[31:26]	Reserved	
SPDIF_SEL	[25:24]	Control MUX_{SPDIF} (00:CLKAUDIO0, 01: CLKAUDIO1, 10:CLKAUDIO2)	0x0
Reserved	[23]	Reserved	
AUDIO2_SEL	[22:20]	Control MUX_{AUDIO2}, which is the source clock of I2S2 (000:$MOUT_{EPLL}$, 001:$DOUT_{MPLL}$, 010: FIN_{EPLL},011: I2SCDCLK2 100: $MOUT_{HPLL}$)	0x0
Reserved	[19]	Reserved	
AUDIO1_SEL	[18:16]	Control MUX_{AUDIO1}, which is the source clock of I2S1, PCM1 (000:$MOUT_{EPLL}$, 001:$DOUT_{MPLL}$, 010:FIN_{EPLL},011: I2SCDCLK1, 100: PCMCDCLK1, 101: $MOUT_{HPLL}$)	0x0
Reserved	[15]	Reserved	
AUDIO0_SEL	[14:12]	Control MUX_{AUDIO0}, which is the source clock of I2S0, PCM0 (000:$MOUT_{EPLL}$, 001:$DOUT_{MPLL}$, 010:FIN_{EPLL},011: I2SCDCLK0, 100: PCMCDCLK0, 101: $MOUT_{HPLL}$)	0x0
Reserved	[11:10]	Reserved	
I2S_D2_SEL	[9:8]	Control MUX_{I2S_D2} (00: $FOUT_{EPLL}$, 01: I2SCDCLK0 , 10:SCLK_AUDIO0) When LP (Low-Power) Audio mode with 'top off' is used, only $FOUT_{EPLL}$ and I2SCDCLK0 can be selected. Refer 7.2 Low Power Music-Play Clock.	0x0
Reserved	[7:5]	Reserved	
HCLK_D2_SEL	[4]	Control MUX_{HCLK_D2} (0: $FOUT_{EPLL}$, 1:I2SCDCLK0)	0x0
Reserved	[3:2]	Reserved	
PWI_SEL	[1:0]	Control MUX_{PWI}, which is the source clock of PWI (00:SRCLK, 01:$MOUT_{EPLL}$, 10: $DOUT_{MPLL}$)	0x0

10.3 CLOCK DIVIDER CONTROL REGISTER

S5PC100 has several clock dividers to support various operating clock frequency. CLK_DIV0, CLK_DIV1, CLK_DIV2, CLK_DIV3, CLK_DIV4, and CLK_DIV_5 controls clock divider ratio.

There are operating frequency limitations. The maximum operating frequency of $DOUT_{MPLL,}$ $DOUT_{MPLL2,}$ $DOUT_{D1\_BUS}$, HCLKD1, and PCLKD1 are 266MHz, 266MHz, 133MHz, 133MHz, and 66MHz respectively. These operating clock conditions must be met through CLK_DIVX configuration.

10.3.1 Set Clock Divider Ratio 0 (Main D0 domain) (CLK_DIV0, R/W, Address = 0xE010_0300)

Main D0 domain divider

CLK_DIV0	Bit	Description	Reset Value
Reserved	[31:19]	Reserved	-
SECSS_RATIO	[18:16]	DIV_{SECSS} clock divider ratio, HCLKD0_SECSS = $DOUT_{D0\_BUS}$ / RATIO (*RATIO = SECSS_RATIO + 1*) SECSS (Secure Sub-System) operating clock cannot exceed 83MHz.	0x1
Reserved	[15]	Reserved	-
PCLKD0_RATIO	[14:12]	DIV_{PCLKD0} clock divider ratio, PCLKD0 = HCLKD0 / RATIO (*RATIO = PCLKD0_RATIO + 1*)	0x1
Reserved	[11]	Reserved	-
DO_BUS_RATIO	[10:8]	DIV_{D0_BUS} clock divider ratio, HCLKD0 = $DOUT_{ARM}$ / RATIO (*RATIO = D0_BUS_RATIO + 1*)	0x0
RESERVED	[7]	RESERVED	-
ARM_RATIO	[6:4]	DIV_{ARM} clock divider ratio, ARMCLK = $DOUT_{APLL}$ / RATIO (*RATIO = ARM_RATIO + 1*)	0x0
Reserved	[3:1]	Reserved	-
APLL_RATIO	[0]	DIV_{APLL} clock divider ratio, $DOUT_{APLL}$ = $MOUT_{APLL}$ / RATIO (*RATIO = APLL_RATIO + 1*)	0x0

At least one of the ARM clock dividers (DIV_{APLL} and DIV_{ARM}) must be set to more than 1. For more information refer to "Section 7.1".

10.3.2 Set Clock Divider Ratio 1 (Main D1 domain) (CLK_DIV1, R/W, Address = 0xE010_0304)

Main D1 domain divider

CLK_DIV1	Bit	Description	Reset Value
Reserved	[31:29]	Reserved	-
CAM_RATIO	[28:24]	DIV_{CAM} clock divider ratio, CLKCAM = $DOUT_{MPLL2}$ / RATIO (*RATIO = CAM_RATIO + 1*)	0x0
Reserved	[23:22]	Reserved	-
ONENAND_RATIO	[21:20]	DIV_{1NAND} clock divider ratio, CLKONENAND = $DOUT_{D1\_BUS}$ / RATIO (*RATIO = ONENAND_RATIO + 1*)	0x0
Reserved	[19]	Reserved	-
PCLKD1_RATIO	[18:16]	DIV_{PCLK} clock divider ratio, PCLK = $DOUT_{D1\_BUS}$ / RATIO (*RATIO = PCLK_RATIO + 1*)	0x1
Reserved	[15]	Reserved	-
D1_BUS_RATIO	[14:12]	DIV_{D1_BUS} clock divider ratio, $DOUT_{D1\_BUS}$= $MOUT_{AMPLL}$ / RATIO (*RATIO = D1_BUS_RATIO + 1*)	0x0
Reserved	[11:9]	Reserved	-
MPLL2_RATIO	[8]	DIV_{MPLL2} clock divider ratio, $DOUT_{MPLL2}$ = $MOUT_{AMPLL}$ / RATIO (*RATIO = MPLL2_RATIO + 1*)	0x0
Reserved	[7:6]	Reserved	-
MPLL_RATIO	[5:4]	DIV_{MPLL} clock divider ratio, $DOUT_{MPLL}$ = $MOUT_{AMPLL}$ / RATIO (*RATIO = MPLL_RATIO + 1*)	0x0
Reserved	[3]	Reserved	-
APLL2_RATIO	[2:0]	DIV_{APLL2} clock divider ratio, $DOUT_{APLL2}$ = $MOUT_{APLL}$ / RATIO (*RATIO = APLL2_RATIO + 1*)	0x0

10.3.3 Set Clock Divider Ratio 2 (Connectivity) (CLK_DIV2, R/W, Address = 0xE010_0308)

Connectivity divider

CLK_DIV2	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	-
UHOST1.1 _RATIO	[23:20]	DIV_{UHOST} clock divider ratio, CLKUHOST = UHOSTMUX/ RATIO (*RATIO =UHOST_RATIO + 1*)	0x0
IRDA_RATIO	[19:16]	DIV_{IRDA} clock divider ratio, CLKIRDA = $CLKIRDA_{IN}$ / RATIO (*RATIO =IRDA_RATIO + 1*)	0x0
SPI2_RATIO	[15:12]	DIV_{SPI2} clock divider ratio, CLKSPI2 = $CLKSPI2_{IN}$ / RATIO (*RATIO = SPI2_RATIO + 1*)	0x0
SPI1_RATIO	[11:8]	DIV_{SPI1} clock divider ratio, CLKSPI1 = $CLKSPI1_{IN}$ / RATIO (*RATIO = SPI1_RATIO + 1*)	0x0
SPI0_RATIO	[7:4]	DIV_{SPI0} clock divider ratio, CLKSPI0 = $CLKSPI0_{IN}$ / RATIO (*RATIO = SPI0_RATIO + 1*)	0x0
UART_RATIO	[3:0]	DIV_{UART} clock divider ratio, CLKUART = $CLKUART_{IN}$ / RATIO (*RATIO = UART_RATIO + 1*)	0x0

10.3.4 Set Clock Divider Ratio 3 (Multimedia and Connectivity) (CLK_DIV3, R/W, Address = 0xE010_030C)

Multimedia and connectivity divider

CLK_DIV3	Bit	Description	Reset Value
HDMI_RATIO	[31:28]	DIV_{HDMI} clock divider ratio, CLKHDMI = $CLKHDMI_{IN}$ / RATIO (*RATIO = HDMI_RATIO + 1*)	0x0
FIMC2_RATIO	[27:24]	DIV_{FIMC2} clock divider ratio, CLKFIMC2 = $CLKFIMC2_{IN}$ / RATIO (*RATIO = FIMC2_RATIO + 1*)	0x0
FIMC1_RATIO	[23:20]	DIV_{FIMC1} clock divider ratio, CLKFIMC1 = $CLKFIMC1_{IN}$ / RATIO (*RATIO = FIMC1_RATIO + 1*)	0x0
FIMC0_RATIO	[19:16]	DIV_{FIMC0} clock divider ratio, CLKFIMC0 = $CLKFIMC0_{IN}$ / RATIO (*RATIO = FIMC0_RATIO + 1*)	0x0
LCD_RATIO	[15:12]	DIV_{LCD} clock divider ratio, CLKLCD = $CLKLCD_{IN}$ / RATIO (*RATIO = LCD_RATIO + 1*)	0x0
MMC2_RATIO	[11:8]	DIV_{MMC2} clock divider ratio, CLKMMC2 = $CLKMMC2_{IN}$ / RATIO (*RATIO = MMC2_RATIO + 1*)	0x0
MMC1_RATIO	[7:4]	DIV_{MMC1} clock divider ratio, CLKMMC1 = $CLKMMC1_{IN}$ / RATIO (*RATIO = MMC1_RATIO + 1*)	0x0
MMC0_RATIO	[3:0]	DIV_{MMC0} clock divider ratio, CLKMMC0 = $CLKMMC0_{IN}$ / RATIO (*RATIO = MMC0_RATIO + 1*)	0x0

10.3.5 Set Clock Divider Ratio 4 (Audio and Others) (CLK_DIV4, R/W, Address = 0xE010_0310)

Audio and Others divider

CLK_DIV4	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x0
AUDIO2_RATIO	[23:20]	DIV_{AUDIO2} clock divider ratio, $CLKAUDIO2 = CLKAUDIO2_{IN}$ / RATIO (*RATIO = AUDIO2_RATIO + 1*)	0x0
AUDIO1_RATIO	[19:16]	DIV_{AUDIO1} clock divider ratio, $CLKAUDIO1 = CLKAUDIO1_{IN}$ / RATIO (*RATIO = AUDIO1_RATIO + 1*)	0x0
AUDIO0_RATIO	[15:12]	DIV_{AUDIO0} clock divider ratio, $CLKAUDIO0 = CLKAUDIO0_{IN}$ / RATIO (*RATIO = AUDIO0_RATIO + 1*)	0x0
I2S_D2_RATIO	[11:8]	DIV_{I2S_D2} clock divider ratio, CLKI2SD2 = I2SD2MUX/ RATIO (*RATIO =I2S_D2_RATIO + 1*)	0x0
Reserved	[7]	Reserved	0x0
HCLK_D2_RATIO	[6:4]	DIV_{HCLK_D2} clock divider ratio, HCLKD2 = HCLKD2MUX/ RATIO (*RATIO = HCLK_D2_RATIO + 1*)	0x0
Reserved	[3]	Reserved	0x0
PWI_RATIO	[2:0]	DIV_{PWI} clock divider ratio, $CLKPWI = CLKPWI_{IN}$ / RATIO (*RATIO = PWI_RATIO + 1*)	0x0

10.4 CLOCK OUTPUT CONFIGURATION REGISTER

Internal clocks can be monitored through XCLKOUT PAD. CLK_OUT register selects an internal clock from PLL outputs, ARMCLK, HCLKD0, HCLKD1, PCLKD0, PCLKD1, 30MHz, 48MHz, 27MHz, 54MHz, RTC, and TICK. It also divides the selected clock. This is just for debugging. Do not supply this to other components as clock.

10.4.1 Select Clock Output (CLK_OUT, R/W, Address = 0xE010_0400)

CLK_OUT	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x000
DIVVAL	[23:20]	Divide ratio (Divide ratio = DIVVAL + 1)	0x0
Reserved	[19:17]	Reserved	0x000
CLKSEL	[16:12]	00000 = $FOUT_{APLL}$/4 00001 = $FOUT_{MPLL}$ 00010 = $FOUT_{EPLL}$ 00011 = $FOUT_{HPLL}$ 00100 = HCLKD1 00101 = HCLKD0 00110 = PCLKD1 00111 = PCLKD0 01000 = HCLKD2 01001 = ARMCLK/4 01010 = RTC clock from PAD 01011 = RTC tick 01100 = CLK48M 01101 = CLK27M 01110 = CLK30M 01111 = VCLK_54 10000 = DOUT	0x0
DCLKCMP	[11:8]	This field changes the clock duty of DCLK. Thus, it must be smaller than DCLKDIV. It is valid only if CLKSEL is 5'b10000. If the DCLKDIV is n, low level duration is (n+1). High level duration is ((DCLKDIV + 1) – (n+1))	0x0
DCLKDIV	[7:4]	DCLK divide value DCLK frequency = source clock / (DCLKDIV + 1)	0x0
Reserved	[3:2]	Reserved	0x0
DCLKSEL	[1]	Select DCLK source clock (0: PCLK, 1: 48MHz)	0
DCLKEN	[0]	Enable DCLK (0:Disables, 1:Enables)	0

Clocks in CLKSEL can be referred at Figure 2.3-3.

CLKOUT frequency = CLKIN (selected by CLKSEL) frequency / (DIVVAL+1)

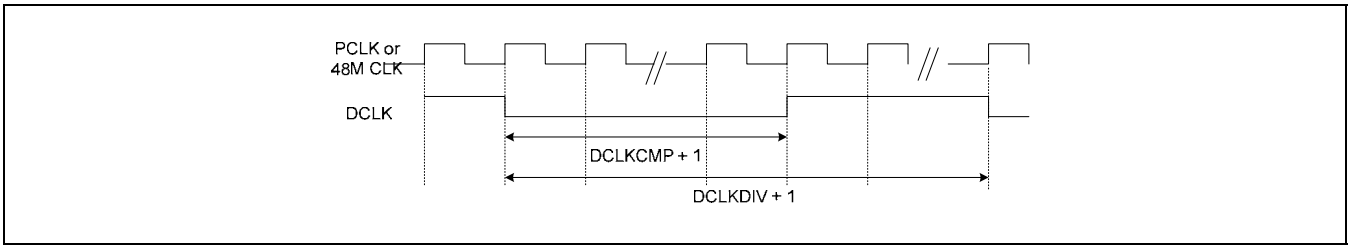


Figure 2.3-15 CLKOUT Waveform with DCLK Divider

10.5 CLOCK GATING CONTROL REGISTER

The following three registers controls clock disable/ enable operation.

10.5.1 Control HCLKD0 / PCLKD0 Clock Gating 0 (System1) (CLK_GATE_D0_0, R/W, Address = 0xE010_0500)

D0 domain system1 clock gating

CLK_GATE_D0_0	Bit	Description	Reset Value
Reserved	[31 : 7]	Reserved	0
CLK_CSSYS	[6]	Gating HCLK & PCLK for CSSYS (0: Mask, 1: Pass)	1
CLK_SECSS	[5]	Gating HCLK (0: Mask, 1: Pass)	1
CLK_G2D	[4]	Gating HCLK for G2D (0: Mask, 1: Pass)	1
CLK_MDMA	[3]	Gating HCLK & PCLKEN for MDMA (0: Mask, 1: Pass)	1
CLK_CFCON	[2]	Gating HCLK for CFCON (0: Mask, 1: Pass)	1
CLK_TZIC	[1]	Gating HCLK for trust interrupt controller (0: Mask, 1: Pass)	1
CLK_INTC	[0]	Gating HCLK for vectored interrupt controller (0: Mask, 1: Pass)	1

10.5.2 Control HCLKD0 / PCLKD0 Clock Gating 1 (Memory) (CLK_GATE_D0_1, R/W, Address = 0xE010_0504)

D0 domain memory clock gating

CLK_GATE_D0_1	Bit	Description	Reset Value
Reserved	[31 : 6]	Reserved	0
CLK_EBI	[5]	Gating HCLK for EBI (0: Mask, 1: Pass)	1
CLK_INTMEM	[4]	Gating HCLK for INTMEM (0: Mask, 1: Pass)	1
CLK_NFCON	[3]	Gating HCLK for NFCON (0: Mask, 1: Pass)	1
CLK_ONENANDC	[2]	Gating HCLK for ONENANDC (0: Mask, 1: Pass)	1
CLK_SROMC	[1]	Gating HCLK for SROMC (0: Mask, 1: Pass)	1
CLK_DMC	[0]	Gating HCLK & PCLK for DMC0 (0: Mask, 1: Pass)	1

10.5.3 Control PCLKD0 Clock Gating 2 (System2) (CLK_GATE_D0_2, R/W, Address = 0xE010_0508)

D0 domain System2 clock gating

CLK_GATE_D0_2	Bit	Description	Reset Value
Reserved	[31 : 3]	Reserved	0
CLK_SDM	[2]	Gating HCLK & PCLKEN for SDM (0: Mask, 1: Pass)	1
CLK_SECKEY	[1]	Gating PCLK for security key (0: Mask, 1: Pass)	1
Reserved	[0]	Do not change this value	1

10.5.4 Reserved (Reserved, Address = 0xE010_050C, 0xE010_051C)

10.5.5 Control HCLKD1 / PCLKD1 Clock Gating 0 (File) (CLK_GATE_D1_0, R/W, Address = 0xE010_0520)

D1 domain File system clock gating

CLK_GATE_D1_0	Bit	Description	Reset Value
Reserved	[31 : 8]	Reserved	0
CLK_HSMMC2	[7]	Gating HCLK for HSMMC2 (0: Mask, 1: Pass)	1
CLK_HSMMC1	[6]	Gating HCLK for HSMMC1 (0: Mask, 1: Pass)	1
CLK_HSMMC0	[5]	Gating HCLK for HSMMC0 (0: Mask, 1: Pass)	1
CLK_MODEMIF	[4]	Gating HCLK for MODEM interface (0: Mask, 1: Pass)	1
CLK_USBOTG	[3]	Gating HCLK for USB OTG (0: Mask, 1: Pass)	1
CLK_USBHOST	[2]	Gating HCLK for UHOST 1.1 (0: Mask, 1: Pass)	1
CLK_PDMA1	[1]	Gating HCLK & PCLKEN for PDMA1 (0: Mask, 1: Pass)	1
CLK_PDMA0	[0]	Gating HCLK & PCLKEN for PDMA0 (0: Mask, 1: Pass)	1

10.5.6 Control HCLKD1 / PCLKD1 Clock Gating 1 (Multimedia1) (CLK_GATE_D1_1, R/W, Address = 0xE010_0524)

D1 domain Multimedia1 system clock gating

CLK_GATE_D1_1	Bit	Description	Reset Value
Reserved	[31 : 9]	Reserved	0
CLK_G3D	[8]	Gating HCLK for G3D (0: Mask, 1: Ppass)	1
CLK_CSI	[7]	Gating HCLK for MIPI_CSI (0: Mask, 1: Pass)	1
CLK_DSI	[6]	Gating PCLK for MIPI_DSI (0: Mask, 1: Pass)	1
CLK_JPEG	[5]	Gating HCLK for JPEG (0: Mask, 1: Pass)	1
CLK_FIMC2	[4]	Gating HCLK for FIMC2 (0: Mask, 1: Pass)	1
CLK_FIMC1	[3]	Gating HCLK for FIMC1 (0: Mask, 1: Pass)	1
CLK_FIMC0	[2]	Gating HCLK for FIMC0 (0: Mask, 1: Pass)	1
CLK_ROTATOR	[1]	Gating HCLK for rotator (0: Mask, 1: Pass)	1
CLK_LCDCON	[0]	Gating HCLK for LCD controller (0: Mask, 1: pass)	1

10.5.7 Control HCLKD1 / PCLKD1 Clock Gating 2 (Multimedia2) (CLK_GATE_D1_2, R/W, Address = 0xE010_0528)

D1 domain Multimedia2 system clock gating

CLK_GATE_D1_2	Bit	Description	Reset Value
Reserved	[31 : 5]	Reserved	0
CLK_MFC	[4]	Gating HCLK & PCLK for MFC (0: Mask, 1: Pass)	1
CLK_HDMI	[3]	Gating HCLK for HDMI (0: Mask, 1: Pass)	1
CLK_MIXER	[2]	Gating HCLK for MIXER (0: Mask, 1: Pass)	1
CLK_VP	[1]	Gating HCLK for VP (0: Mask, 1: Pass)	1
CLK_TV	[0]	Gating HCLK for TV encoder (0: Mask, 1: Pass)	1

10.5.8 Control HCLKD1 / PCLKD1 Clock Gating 1 (System) (CLK_GATE_D1_3, R/W, Address = 0xE010_052C)

D1 domain system control clock gating

CLK_GATE_D1_3	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0
CLK_RTC	[9]	Gating PCLK for RTC (0: Mask, 1: Pass)	1
CLK_WDT	[8]	Gating PCLK for watch dog timer (0: Mask, 1: Pass)	1
CLK_SYSTIMER	[7]	Gating PCLK for system timer (0: Mask, 1: Pass)	1
CLK_PWM	[6]	Gating PCLK for PWM (0: Mask, 1: Pass)	1
Reserved	[5]	Do not change this value	1
Reserved	[4]	Do not change this value	1
CLK_IEC	[3]	Gating PCLK for IEM_IEC (0: Mask, 1: Pass)	1
CLK_APC	[2]	Gating PCLK for IEM_APC (0: Mask, 1: Pass)	1
CLK_GPIO	[1]	Gating PCLK for GPIO (0: Mask, 1: Pass)	1
CLK_CHIPID	[0]	Gating PCLK for chip ID (0: Mask, 1: Pass)	1

10.5.9 Control HCLKD1 / PCLKD1 Clock Gating 2 (Connectivity) (CLK_GATE_D1_4, R/W, Address = 0xE010_0530)

D1 domain connectivity system clock gating

CLK_GATE_D1_4	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0
CLK_HSIRX	[13]	Gating PCLK for MIPI_HSI receiver (0: mask, 1: pass)	1
CLK_HSITX	[12]	Gating PCLK for MIPI_HSI transmitter (0: mask, 1: pass)	1
CLK_CCAN1	[11]	Gating PCLK for CCAN1 (0: mask, 1: pass)	1
CLK_CCAN0	[10]	Gating PCLK for CCAN0 (0: mask, 1: pass)	1
CLK_IRDA	[9]	Gating PCLK for IRDA (0: mask, 1: pass)	1
CLK_SPI2	[8]	Gating PCLK for SPI2 (0: mask, 1: pass)	1
CLK_SPI1	[7]	Gating PCLK for SPI1 (0: mask, 1: pass)	1
CLK_SPI0	[6]	Gating PCLK for SPI0 (0: mask, 1: pass)	1
CLK_HDMI_IIC	[5]	Gating PCLK for IIC for HDMI (0: mask, 1: pass)	1
CLK_IIC	[4]	Gating PCLK for IIC (0: mask, 1: pass)	1
CLK_UART3	[3]	Gating PCLK for UART3 (0: mask, 1: pass)	1
CLK_UART2	[2]	Gating PCLK for UART2 (0: mask, 1: pass)	1
CLK_UART1	[1]	Gating PCLK for UART1 (0: mask, 1: pass)	1
CLK_UART0	[0]	Gating PCLK for UART0 (0: mask, 1: pass)	1

10.5.10 Control HCLKD1 / PCLKD1 Clock Gating 2 (Audio) (CLK_GATE_D1_5, R/W, Address = 0xE010_0534)

D1 domain audio system clock gating

CLK_GATE_D1_5	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0
Reserved	[9]	Reserved	1
CLK_KEYIF	[8]	Gating PCLK for Key IF (0: mask, 1: pass)	1
CLK_TSADC	[7]	Gating PCLK for touch screen ADC (0: mask, 1: pass)	1
CLK_SPDIF	[6]	Gating PCLK for SPDIF (0: mask, 1: pass)	1
CLK_PCM1	[5]	Gating PCLK for PCM1 (0: mask, 1: pass)	1
CLK_PCM0	[4]	Gating PCLK for PCM0 (0: mask, 1: pass)	1
CLK_AC97	[3]	Gating PCLK for AC97 (0: mask, 1: pass)	1
CLK_I2S2	[2]	Gating PCLK for I2S2 (0: mask, 1: pass)	1
CLK_I2S1	[1]	Gating PCLK for I2S1 (0: mask, 1: pass)	1
CLK_I2S0	[0]	Gating PCLK for I2S0 in D2_SS (0: mask, 1: pass)	1

10.5.11 Reserved (Reserved, Address = 0xE010_0538, 0xE010_053C)

10.5.12 Control HCLKD2 Clock Gating 0 (Audio) (CLK_GATE_D2_0, R/W, Address = 0xE010_0540)

D2 domain audio system clock gating

CLK_GATE_D2_0	Bit	Description	Reset Value
Reserved	[31 : 2]	Reserved	0
CLK_I2SCLKD2	[1]	Gating I2SCLK for D2 (0: mask, 1: pass)	0
CLK_HCLKD2	[0]	Gating HCLK for D2 (0: mask, 1: pass)	0

10.5.13 Reserved (Reserved, Address = 0xE010_0544, 0xE010_055C)

10.5.14 Control SCLK Clock Gating (System) (CLK_GATE_SCLK_0, R/W, Address = 0xE010_0560)

System Special clock gating

CLK_GATE_SCLK_0	Bit	Description	Reset Value
Reserved	[31:18]	Reserved	0
SCLK_MMC2_48	[17]	Gating special clock 48MHz for MMC2 (0: mask, 1: pass)	1
SCLK_MMC1_48	[16]	Gating special clock 48MHz for MMC1 (0: mask, 1: pass)	1
SCLK_MMC0_48	[15]	Gating special clock 48MHz for MMC0 (0: mask, 1: pass)	1
SCLK_MMC2	[14]	Gating special clock for MMC2 (0: mask, 1: pass)	1
SCLK_MMC1	[13]	Gating special clock for MMC1 (0: mask, 1: pass)	1
SCLK_MMC0	[12]	Gating special clock for MMC0 (0: mask, 1: pass)	1
SCLK_USBHOST	[11]	Gating special clock for UHOST1.1 (0: mask, 1: pass)	1
SCLK_IRDA	[10]	Gating special clock for IRDA (0: mask, 1: pass)	1
SCLK_SPI2_48	[9]	Gating special clock 48MHz for SPI2 (0: mask, 1: pass)	1
SCLK_SPI1_48	[8]	Gating special clock 48MHz for SPI1 (0: mask, 1: pass)	1
SCLK_SPI0_48	[7]	Gating special clock 48MHz for SPI0 (0: mask, 1: pass)	1
SCLK_SPI2	[6]	Gating special clock for SPI2 (0: mask, 1: pass)	1
SCLK_SPI1	[5]	Gating special clock for SPI1 (0: mask, 1: pass)	1
SCLK_SPI0	[4]	Gating special clock for SPI0 (0: mask, 1: pass)	1
SCLK_UART	[3]	Gating special clock for UART0~3 (0: mask, 1: pass)	1
SCLK_ONENAND	[2]	Gating special clock for ONENAND (0: mask, 1: pass)	0
SCLK_PWI	[1]	Gating special clock for PWI (0: mask, 1: pass)	1
SCLK_HPM	[0]	Gating special clock for HPM (0: mask, 1: pass)	1

10.5.15 Control SCLK Clock Gating (Multimedia) (CLK_GATE_SCLK_1, R/W, Address = 0xE010_0564)

Multimedia Special clock gating

CLK_GATE_SCLK_1	Bit	Description	Reset Value
Reserved	[31:13]	Reserved	0
SCLK_CAM	[12]	Gating special clock for Camera (0: mask, 1: pass)	1
SCLK_SPDIF	[11]	Gating special clock for SPDIF (0: mask, 1: pass)	1
SCLK_AUDIO2	[10]	Gating special clock for I2S2 (0: mask, 1: pass)	1
SCLK_AUDIO1	[9]	Gating special clock for PCM1, I2S1 (0: mask, 1: pass)	1
SCLK_AUDIO0	[8]	Gating special clock for PCM0, I2S0 (0: mask, 1: pass)	1
SCLK_HDMI	[7]	Gating special clock for HDMI (0: mask, 1: pass)	1
SCLK_MIXER	[6]	Gating special clock for MIXER (0: mask, 1: pass)	1
SCLK_VDAC54	[5]	Gating special clock for VDAC (0: mask, 1: pass)	1
SCLK_TV54	[4]	Gating special clock for TV encoder (0: mask, 1: pass)	1
SCLK_FIMC2	[3]	Gating special clock for FIMC2 (0: mask, 1: pass)	1
SCLK_FIMC1	[2]	Gating special clock for FIMC1 (0: mask, 1: pass)	1
SCLK_FIMC0	[1]	Gating special clock for FIMC0 (0: mask, 1: pass)	1
SCLK_LCD	[0]	Gating special clock for LCD controller (0: mask, 1: pass)	1

10.6 OTHER SFRS

10.6.1 Generate Software Reset (SWRESET, R/W, Address = 0xE020_0000)

SW RESET Registers

SWRESET	Bit	Description	Reset Value
Reserved	[31:18]	Reserved	0x0
SWRESET	[15:0]	Generates software reset if the value is 0xC100	0x0000

10.6.2 OneNAND Controller Setting (ONENAND_SWRESET, R/W, Address = 0xE020_0008)

ONENAND_SWRESET	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x0
ONENAND_SWRESET	[0]	ONENAND SW Reset (0 : Reset, 1: Normal mode)	0x0

10.6.3 General Control Register (GENERAL_CTRL, R/W, Address = 0xE020_0100)

GENERAL_CTRL	Bit	Description	Reset Value
Reserved	[31:0]	Do not change this value	32'b0

10.6.4 General Status Register (GENERAL_STATUS, R, Address = 0xE020_0104)

GENERAL_STATUS	Bit	Description	Reset Value
Reserved	[31:14]	Do not change this value	18'b0
TRIGGER	[13]	ETM trigger flag 0 = Trigger does not occur 1 = Trigger occurs	1'b0
Reserved	[12:2]	Do not change this value	11'b0
STANDBYWFI	[1]	Standby mode flag generated by WFI operation 0 = Processor not in standby mode 1 = Processor in standby mode	1'b0
Reserved	[0]	Do not change this value	1'b0

10.6.5 ENDIAN & EBI Configuration (MEM_SYS_CFG, R/W, Address = 0xE020_0200)

MEM_SYS_CFG	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
EBI_PRI	[8]	Set EBI priority scheme. 0 = Fixed priority scheme. 1 = Circular priority scheme.	0
Reserved	[7:6]	Reserved	
EBI_FIX_PRI	[5:4]	Set EBI fixed priority setting. 0 = SROMC - OneNANDC – NFCON – CFCON 1 = OneNANDC – SROMC – NFCON – CFCON 2 = NFCON – SROMC - OneNANDC – CFCON 3 = CFCON – SROMC - OneNANDC – NFCON	0
Reserved	[3:1]	Reserved	
ENDIAN	[0]	Set NFCON and SROMC endian control 0 = Little endian 1 = Big endian	0

10.6.6 Camera Mapping to FIMC Selection (CAM_MUX_SEL, R/W, Address = 0xE020_0300)

CAM_MUX_SEL	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0x0
CAM_MUX_SEL	[1:0]	Camera interface demuxing selection (0 : FIMC0, 1 : FIMC1, 2 : FIMC2)	0x00

10.6.7 Video Mixer Output to TVENC / HDMI Selection (MIXER_OUT_SEL, R/W, Address = 0xE020_0304)

MIXER_OUT_SEL	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x0
MIXER_OUT_SEL	[0]	Video Mixer output demuxing selection (0 : TVENC, 1 : HDMI)	0x0

10.6.8 Low Power MP3 Mode Selection (LPMP3_MODE_SEL, R/W, Address = 0xE020_0308)

LPMP3_MODE_SEL	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0x0
I2S0_FIFO_CLK_SEL	[1]	I2S0 in D2_SS operation clock selection (0 = I2S0 bit clock from AUDIO CMU (in D2_SS), 1 : PCLK from D1_SS)	0x0
LPMP3_MODE_SEL	[0]	Rotator memory demuxing selection (0 = FIMC, 1 : LPAUDIO)	0x0

10.6.9 MIPI D-PHY Control Register0 (MIPI_PHY_CON0, R/W, Address = 0xE020_0400)

MIPI_PHY_CON0	Bit	Description	Reset Value
Reserved	[31:2]	Do not change this value	0x0
M_RESETN	[1]	Reset DSIM part of MIPI_PHY This bit should be 1 before enabling DSIM	0x0
S_RESETN	[0]	Reset CSIS part of MIPI_PHY This bit should be 1 before enabling CSIS	0x0

10.6.10 MIPI D-PHY Control Register1 (MIPI_PHY_CON1, R/W, Address = 0xE020_0414)

MIPI_PHY_CON1	Bit	Description	Reset Value
PLL_BYPASS_SEL	[31]	Master PPI I/F signal. If it is high, the clock signal from M_EXTSERCLK is selected for serial clock for the Master lane instead of Master PLL. When it is low (default), Master PLL output is selected for serial clock. This is used for MIPI DSI.	0x0
Reserved	[30:28]	Reserved	0x0
FORCE_SWAP_EN	[27]	Force DP/DN swap enable 0: Do not use swap. 1: Enable swap feature according to M_DPDN_SWAP & S_DPDN_SWAP.	0x0
Reserved	[26]	Do not change this value	0x0
M_DPDN_SWAP	[25]	This is to prevent crossing PCB trace lines of DP/DN if the master lane is connected to a slave lane on PCB board. If it is set to HIGH, all the Master's DP/DN signals are swapped. This is used for MIPI DSI.	0x0
S_DPDN_SWAP	[24]	Slave PPI I/F signal. This is to prevent crossing PCB trace lines of DP/DN if the slave lane is connected to a master lane on PCB board. If set to HIGH, all the Slave's DP/DN signals are swapped. This is used for MIPI CSI.	0x0
Reserved	[23:0]	Do not change this value	0x0

10.6.11 HDMI PHY Control Register0 (HDMI_PHY_CON0, R/W, Address = 0xE020_0420)

HDMI_PHY_CON0	Bit	Description	Reset Value
Reserved	[31:0]	Do not change this value	0x0

2.4 POWER MANAGEMENT

1 OVERVIEW

The mobile application processor such as S5PC100 needs to consume low power since a mobile product has a small battery and limited power capacity.

The purpose of power management is to provide various power saving methods to S5PC100 in consuming power as low as possible under specific application scenarios.

Power management scheme in S5PC100 provides six system power modes, namely: NORMAL, IDLE, DEEP-IDLE, STOP, DEEP-STOP, and SLEEP.

The brief description of each power mode is as follows:

- **NORMAL:** The CPU core is running. Application programs can run normally.
- **IDLE:** The clock of the CPU core is turned off, but the remaining parts are running normally.
- **DEEP-IDLE:** The CPU core is power-off. The remaining parts can be power-on or power-off.
  1) Top domain power on: Sub and Audio domains have the same states as in NORMAL mode.
  2) Top domain power off: Sub domains become power-off, but Audio domain is running for low-power MP3 playback.
- **STOP:** All the blocks, including the CPU core, are clock-off except the RTC and Alive domain. The state of sub-power-domain power-off domain in NORMAL mode is still power-off in STOP mode. (The Stop mode should be used for test-purpose only.)
- **DEEP-STOP:** CPU core is power-off, and the remaining parts of the chip except RTC and ALIVE can have different configurations according to Top domain status.
  1) Top domain power on (clock off): Sub and Audio domains are power-on. (This feature should be used for test-purpose only.)
  2) Top domain power off: Sub domains are power-off.
- **SLEEP:** The main power is externally turned off using regulators or PMICs (Power Management IC) except the ALIVE domain and I/O pads. Note that separate RTC power to RTC and external power to I/O pad is still on. If wakeup event occurs, wakeup reset generated by the Power Management Unit initializes S5PC100 to the reset states except the ALIVE domain.

Note that 'DEEP' means CPU core is power-off, therefore CPU cores leakage power is minimized in those power modes (i.e. DEEP-IDLE and DEEP-STOP).

The above description about power mode is provided in view of internal digital logic. The power modes of Cortex-

A8, SRAM and other hard macros are controlled as explained in "Section 6 HARD MACRO POWER CONTROL". Power Management controls the power mode of SRAM, ROM, PLL, TS ADC and I/O. The power mode of hard macros except SRAM, ROM, PLL, TS ADC and I/O should be controlled by corresponding control module. Note that in addition to Power Management Unit (PMU below), Clock Controller (CLKCON below) also controls the PLL. According to application scenario, above mentioned proper system power mode should be applied, and hard macro should be controlled.

2 FUNCTION DESCRIPTION

There are two types of power consumption namely static and dynamic.. Static power is consumed if power to a circuit is supplied and there is no active operation in the circuit. Dynamic power is consumed if the signal to a circuit is toggling, and there is some active operation in the circuit. The static power consumption is due to leakage current in the process while the dynamic power consumption is due to charging and discharging of capacitors. The dynamic power consumption depends on operating voltage, operating frequency, and toggling ratios of the logic gate.

Various power-saving techniques have been developed and they are listed in Table 2.4-1.

Table 2.4-1 Comparison of Power Saving Techniques

Power saving techniques	Result	Clock	Power	State Retention	
				Normal F/F	Retention F/F
Frequency scaling	Reduce dynamic power	Enable	Supplied	Keep state	
Clock off	Minimize dynamic power	Disable	Supplied	Keep state	
Power off	Minimize leakage power	Disable	Not supplied (internally off)	Lose state	Keep state
External power off	Nearly zero power	Disable	Not supplied (externally off)	Lose state	

Frequency scaling means that the frequency of clock to a specific Intellectual Property (IP) module is lowered if the module is not required to run fast. Frequency scaling reduces dynamic power.

Clock off means that clock gating cells in Clock Controller is used to disable clock to a specific IP module. These clock gating cells are controlled by setting registers (CLK_GATE_D0_0-2, CLK_GATE_D1_0-5, and CLK_GATE_SCLK_0-1) in Clock Controller. If clock off is applied, power to logic gate is still supplied, and therefore the states of Normal F/F (Flip-Flop) and Retention F/F are maintained. Retention F/F is developed to meet special purpose to keep its state although power is not supplied due to power off.

Power off means that current path to a specific power domain (a group of IP modules) is internally disconnected using switch cells in that power domain. Therefore power to that domain is not supplied.

The switch cells are controlled by setting registers (NORMAL_CFG, and STOP_CFG) in PMU. Note that external power to S5PC100 is not off. Therefore, there can be two power off techniques as listed below.

- Power off without state retention
  - ♦ Normal F/F is used. ♦ Wakeup reset is necessary.

- Power off with state retention
  - ♦ Retention F/F is used. ♦ Wakeup reset is not necessary.

The power off does not preserve the state of normal F/F in the power-down domain. A power domain except TOP domain has only normal F/F, and does not have retention F/F. Therefore, sub-domain, power domain except TOP domain, does not preserve the state of F/F when the sub-domain is power-off. If a sub-domain is powered up again, a wakeup reset is invoked for the modules in the sub-domain.

But, top domain has both normal F/F and retention F/F, and each module in top domain has only normal F/F or retention F/F. It is not the case that a module has both normal F/F and retention F/F. Therefore, the module that

has retention F/F in top domain keeps the state of F/F if top domain is power-off, but the module that has normal F/F in top domain loses the state of F/F if the top domain is power-off. If top domain is powered up again, a wakeup reset is not invoked for the modules that have retention F/F, but a wake up reset is invoked for the modules that have normal F/F.

The retention modules in TOP domain are listed in Table 2.4-4.

The power-up takes time to stabilize the internal logic gates and memory after power is supplied again. The power-up time is necessary because a simultaneous power-up of all logic gates and all memories is not allowed since simultaneous power-up drains a large amount of current and may cause system malfunction.

There are two wakeup techniques in logic part. One technique is applied to TOP domain and the other one is applied to SUB domains except System Timer domain. Note that System Timer is always ON except SLEEP mode, since internal switch is not OFF in DEEP-IDLE (top domain off) and DEEP-STOP (top domain off).

First technique:

The logic gates in TOP domain are powered on in two steps, and then memories are powered on one by one memory group. Two steps means that power to about 10% of all switches are supplied first, and the power of remaining 90% switches is supplied after some time. This is automatically done by hardware logic. The power-up time is composed of the logic power-up time and the memory power-up time. These are determined by the oscillator frequency, the number of memories and the values set in the OSC_FREQ and INTERNAL_PWR_STABLE registers. The count values vary depending on the size of the logic gates and the number of memories.

Second technique:

The logic gates and memory in SUB domains except System Timer domain are turned ON at the same time. But to prevent wakeup noise from occurring, the power to switches is supplied in two steps in similar with first technique. Power to about 10% switches of all is supplied first, and the power of remaining 90% switched is supplied after some time goes. This is automatically done by hardware logic. The power-up time is determined by the oscillator frequency, the number of memories and the values set in the OSC_FREQ and INTERNAL_PWR_STABLE registers. The count values vary depending on the size of the logic gates and the number of memories.

External power off means that power to S5PC100 is externally off using regulator or PMIC (Power Management IC). PMU in S5PC100 generates power control signal to regulator or PMIC. If external power off is applied, the states of Normal F/F and Retention F/F are lost. Therefore, if you want to save some important data, you should move those data to external memory, and restore those data when wakeup event occurs.

To reduce the dynamic power consumption, S5PC100 uses clock off and frequency scaling. IP module basis disables clocks in S5PC100. Reduce the clock frequency if the system is not required to operate at the maximum frequency.

To reduce power consumption further in the system level, S5PC100 include following features.

- Make DRAM enter self-refresh mode when S5PC100 enters STOP, DEEP-STOP and SLEEP mode.

To reduce the static current, S5PC100 supports block-based power off. In specific applications, a certain group of IP modules are not required to run and therefore do not need to be powered on. For example, in case of MP3 playback, Multi-Format Codec (MFC), Video modules (Camera interface, JPEG, Video processor, Mixer, etc.), and 3D graphics core do not need to operate and can be power-off for the minimum static power consumption. S5PC100 internal modules are grouped into ten power domains based on their functions as shown in Table 2.4-2

and seven power domains except System Timer, ALIVE and RTC can be power-off by turning off switch cells which connects the current path from VDD to GND.

How to control the power mode of Cortex-A8, SRAM, and other hard macros included in each power domain is summarized in Table 2.4-8 and the detail description is given in Section 6 HARD MACRO POWER CONTROL.

Each hard macro has its own power-down mode. The power mode of SRAM, ROM, PLL, TS ADC and I/O are controlled by power management. The power mode of hard macros except SRAM, ROM, PLL, TS ADC and I/O should be controlled by corresponding control module. Note that in addition to PMU, CLKCON also controls the PLL.

Table 2.4-2 S5PC100 Power Domains

	Power Domain	Included Modules
1	CPU	Cortex-A8 Core, L2 Cache, ETM , NEON
2	MFC	MFC
3	G3D	G3D
4	Audio Sub-system	Audio related modules: I2S0, SRAM (128 KB)
5	LCD Sub-system	CAMIF, DISPCON, G2D, JPEG, MIPI_CSI, MIPI_DSI, ROTATOR
6	TV Sub-system	HDMI, MIXER, TVEncoder, VP
7	System Timer	System Timer
8	TOP	Others (AXI bridge, Memory controller, GPIO_OFF, peripherals, etc.)
9	ALIVE	Power Management Unit, GPIO_ALIVE, Wakeup Logic
10	RTC	RTC

3 SYSTEM POWER MODE

3.1 OVERVIEW

According to the power saving schemes and features explained in the "Section 2 FUNCTION DESCRIPTION", S5PC100 provides six power modes: NORMAL, IDLE, DEEP-IDLE, STOP, DEEP-STOP, and SLEEP.

Power modes are summarized in Table 2.4-3.

In NORMAL mode, use module-based clock off, block-based power off and frequency scaling to reduce power consumption.
Based on operating scenario clock off reduces dynamic power consumption by disabling clock input to specific module. Clock off can be done in module-by-module basis.
Power off reduces static power consumption of a block or power domain (a group of modules) by disconnecting a leakage-current path. Power off can be done in block-by-block basis.
Frequency scaling reduces dynamic power consumption by lowering operating frequency.

In IDLE mode, CPU clock is disabled internally by entering Standby mode of Cortex-A8. CPU enters into Standby mode by performing WFI instruction. In this mode, Cortex-A8 core is not running, therefore dynamic power of CPU is reduced. The remaining parts of the chip keep their states in NORMAL mode. That is, clock-off modules are still clock-off, and power-off domains are still power-off.

In DEEP-IDLE mode, Cortex-A8 core is power-off rather than clock-off. In DEEP-IDLE mode, CPU cores leakage power is minimized. There are two options in DEEP-IDLE mode.
One option is that the remaining parts of the chip keep their states in NORMAL mode.
Second option is that for low-power MP3 playback, Top domain is power-off and Sub-domain except Audio domain is power-off whereas Audio domain is still power on.
Set TOP_LOGIC_ON_DIDLE field of PWR_CFG register in PMU to select the above mentioned options, i.e., TOP domain can either power-on or power-off by the setting of TOP_LOGIC_ON_DIDLE field of PWR_CFG register before it enters into IDLE mode.

- TOP_LOGIC_ON_DIDLE = 1'b1, Top domain, Sub-domain, and Audio domains keep their states in NORMAL mode.
- TOP_LOGIC_ON_DIDLE = 1'b0, Top domain and Sub-domain are power-off, but Audio domain is still running.

As you can see, 'DEEP' means that Cortex-A8 Core is power-off.

In STOP mode, clocks to all modules except RTC are disabled, PLLs are disabled, and unnecessary oscillators are selectively disabled to minimize dynamic power consumption. In this mode, Cortex-A8 Core enters into Standby mode.

In DEEP-STOP mode, Cortex-A8 Core is power-off rather than clock-off as in STOP mode, and the remaining parts of the chip are power-off except TOP, RTC, and ALIVE module. But TOP domain can be either power-on or power-off by the setting of TOP_LOGIC_ON field of STOP_CFG register before it enters into DEEP-STOP mode. Cortex-A8 L2 cache is powered on for memory retention, or power-off to save power.

- TOP_LOGIC_ON = 1'b1, Top domain, Sub-domain, and Audio domains are clock-off.
- TOP_LOGIC_ON = 1'b0, Top domain, Sub-domain, and Audio domains are power-off.

Table 2.4-3 Power Mode Summary

Power Mode		NORMAL	IDLE	DEEP-IDLE[5]	STOP	DEEP-STOP	SLEEP
Cortex-A8	Core	Run with IEM[1]	Standby	Power off	Standby	Power off	External power off
	L2 Cache	Run with IEM	Standby	Retention / Power off	Standby	Retention / Power off	External power off
Logic	SUB[2]	Power on / Clock off / Power off	KEEP power state in NORMAL mode	KEEP power state in NORMAL mode / Power off[5]	Clock off / Power off	Clock off / Power off	External power off
	Audio domain[3]	Power on / Clock off / Power off	KEEP power state in NORMAL mode	KEEP power state in NORMAL mode	Clock off	Clock off / Power off	External power off
	TOP[4]	Power on / Clock off /	KEEP power state in NORMAL mode	KEEP power state in NORMAL mode / Power off[5]	Clock off	Clock off / Power off	External power off
ALIVE		Power on					
PLLs		Selectively Disabled	Selectively Disabled		Disabled		External power off
OSCs (XXTI27, XusbXTI)		Selectively Disabled	Selectively Disabled		Selectively Disabled		Selectively Disabled
Typical wakeup time[6]	XXTI enabled	XXTI 10 cycles[7]	XXTI 6 cycles	XXTI 1424 cycles[8]	XXTI 30 cycles + 300 us[9]	XXTI 926 cycles + 300 us or XXTI 1688 cycles[10]	XXTI 138 cycles + 5ms[11]
	XXTI disabled	N.A	N.A	N.A	1.3 ms[12]	XXTI 896 cycles + 1.3ms or XXTI 1688 cycles + 1ms[13]	

1) IEM means Intelligent Energy Management introduced by ARM. IEM is explained in detail separately in IEM related TRM (Technical Reference Manual).
2) SUB means power domain of Row 2, 3, 5, and 6 in Table 2.4-2.
3) Audio domain means power domain of Row 4 in Table 2.4-2.
4) TOP means power domain of Row 8 in Table 2.4-2.
5) There is second option in DEEP-IDLE mode for low-power MP3 playback, i.e., Top domain and Sub-domain is power-off, but Audio domain is still power on.
6) This time is measured from wakeup event assertion to ARM reset release (DEEP-IDLE, DEEP-STOP, and SLEEP mode) or ARM clock supply (IDLE and STOP mode). That is, ARM runs the next instruction this time after wakeup event is asserted. Restored time is not included. Those saved data in external memory should be restored after this time. All values are measured assuming external 12MHz oscillator.
For more information refer to Section 8 WAKEUP TIMING DIAGRAM.

7) Wake-up time in this case means the time to power-up a power domain.
8) until ARM reset is released
9) until ARM clock is supplied, 300us for APLL locking time
10) until ARM reset is released. If top domain on, wakeup time is XXTI 926 cycles added by APLL locking time (300us). If top domain off, wakeup time is XXTI 1688 cycles.
11) 5ms for regulator on + XXTI 138 cycles for ARM reset release. 5ms (regulator on-time) can be changed according to the type of regulator or PMIC (power management IC).
12) until ARM clock is supplied, 1ms for oscillator stabilization time + 300 us for APLL locking time
13) until ARM reset is released. If top domain on, wakeup time is XXTI 896 cycles added by APLL locking time (300us) and oscillator stabilization time (1ms). If top domain off, wakeup time is XXTI 1688 cycles added by oscillator stabilization time (1ms).

In SLEEP mode, power for all domains except ALIVE domain is not supplied since external power source is off by regulator or PMIC, and all PLLs and unnecessary oscillators are disabled. Static power consumption is very small in SLEEP mode. The only leakage power source is due to power supplied to ALIVE domain.

In STOP and SLEEP mode, PLL is disabled by hardware and OSCs are selectively disabled by setting OSC_EN field of STOP_CFG and SLEEP_CFG register in PMU.

3.2 NORMAL MODE

In NORMAL mode, clock off, power off, and frequency scaling are used for power saving.

Clock off is done in module-by-module basis. Disable the clock of one or more modules by setting the corresponding bits in the clock on/off control registers (CLK_GATE_D0_0-2, CLK_GATE_D1_0-5, and CLK_GATE_SCLK_0-1) in the Clock Controller module.

Frequency scaling is done in PLL-by-PLL basis. Lower the operating frequency of the modules by changing the PLL P/M/S values. Changing a P/M/S value causes PLL lock operation which takes time 300us. S5PC100 stops its operation during the PLL lock period since the PLL output clock is masked. Refer to Chapter 2.3 Clock Strategy to know how to change P/M/S value, and related clock divider value.

Power off is done in block-by-block basis. You can do power off one or more blocks by setting the corresponding bits in NORMAL_CFG register. The IP blocks which are power-off in NORMAL mode are, MFC, G3D, LCD sub-system, and TV sub-system (Refer to Table 2.4-2).

The power domain can also be powered on by setting the corresponding bit in NORMAL_CFG register. You can power on or power-off multiple power domains at the same time by changing multiple bits in the NORMAL_CFG registers. You should not initiate power on (or power off) before power off (or power on) is finished.

Power off status of each power domain is found in the BLK_PWR_STAT register. BLK_PWR_STAT is not updated until the power-up or power-down process is completed. NORMAL_CFG and BLK_PWR_STAT have different values while the power-up or power-down procedure is going on, and has the same value after the power state change is completed. Look up the BLK_PWR_STAT register value to check the completion of power off.

If you want to power on or off each power domain(Sub/Audio), you should first write the corresponding bits of NORMAL_CFG, and then you should check the power on/off status by reading the corresponding bits of BLK_PWR_STAT register. After checking the power on/off status, you should proceed to the next action.

Note that Cortex-A8 is always ON in NORMAL mode.

3.3 IDLE MODE

If Cortex-A8 is not required to operate, the clock for Cortex-A8 can be disabled internally in order to save the dynamic power consumption. This is done by executing a 'Wait For Interrupt' instruction. Except the state of Cortex-A8 core, the remaining parts of the chip, keep their operating states in NORMAL, i.e., the running modules are still running, clock-gated modules are still clock-gated, and powered-off modules are still power-off.

To enter IDLE mode:

1. Set CFG_STANDBYWFI field of PWR_CFG to 2'b01.
2. Set DEEP field register in CFG_DEEP_IDLE field of PWR_CFG to 1'b0.
3. Set PMU_INT_DISABLE bit of OTHERS register to 1'b1 to prevent interrupts from occurring while entering IDLE mode.
4. Execute 'Wait For Interrupt' instruction (WFI).

To exit IDLE mode:

1. Various types of wakeup sources are used. Wakeup sources referred in Section 5 WAKEUP SOURCES.

3.4 DEEP-IDLE MODE

If Cortex-A8 is not required to operate, and for further reduction of CPU power, the power to Cortex-A8 core is off internally in order to save the dynamic power consumption. Set registers (PWR_CFG) in PMU and execute a Wait for Interrupt instruction.

There are two options in DEEP-IDLE mode.
First option is that the remaining parts of the chip, keep their states in NORMAL mode.
Second option for low-power MP3 playback, i.e., Top domain and Sub-domain is also power off, but Audio domain is still power on.
Set TOP_LOGIC_ON_DIDLE field of PWR_CFG register in PMU to select this options, i.e., TOP domain can be either power-on or power-off by the setting of TOP_LOGIC_ON_DIDLE field of PWR_CFG register before the entry into IDLE mode.

- TOP_LOGIC_ON_DIDLE = 1'b1, Top domain (logic), Sub-domain (logic), and Audio domain (logic) keep their states in NORMAL mode.
- TOP_LOGIC_ON_DIDLE = 1'b0, Top domain (logic) and Sub-domain (logic) is also power-off, but Audio domain (logic) is still power on.

If TOP_LOGIC_ON_DIDLE = 1'b0, some IP modules in Top domain loses their states in F/F, and the other IP modules keep their states in F/F after wakeup. Table 2.4-4 shows IP modules in Top domain that have retention F/Fs, and those modules keep their states in F/F after wakeup.

However, IP modules in Top domain listed in Table 2.4-5 do not keep their states in F/F in DEEP-IDLE mode (Top domain off), therefore the states of F/F should be saved into external memory, and be restored after wakeup.

Note that IP modules in Sub/Audio domains do not keep their states in F/F and SRAM when the power domains become power-off.

In Deep-Idle mode OSC is always ON.

Table 2.4-4 IP Modules in Top Domain Having Retention F/Fs

Retention Module(1)	SDMMC, USB Host 1.1, USB OTG link, CFCON, UART, I2S, SPI, PCM, AC97, IrDA, System Timer

1) The state of SFR in Retention module is kept in top domain off.

To enter DEEP-IDLE mode:

1. Set CFG_STANDBYWFI field of PWR_CFG to 2'b01.
2. Set DEEP field register in CFG_DEEP_IDLE field of PWR_CFG to 1'b1.
3. Set PMU_INT_DISABLE bit of OTHERS register to 1'b1 to prevent interrupts from occurring while entering DEEP-IDLE mode.
4. Execute Wait for Interrupt instruction (WFI).

PMU performs the following sequence on entering the DEEP-IDLE mode (TOP_LOGIC_ON_DIDLE = 1'b0). This sequence is automatically done by hardware logic.

1. Completes all active bus transactions.
2. Completes all active memory controller transactions.
3. Initiates external DRAM to enter self-refresh mode (to preserve DRAM contents).
4. Mask clock input using internal signal in PMU.
5. Disables all PLLs except EPLL if Top domain is power-off.
6. Selectively disables OSCs except 32.768kHz.

To exit DEEP-IDLE mode:

1. Various types of wakeup sources are used. Wakeup sources referred in Section 5 WAKEUP SOURCES.

Then PMU performs the following sequence to exit DEEP-IDLE mode (TOP_LOGIC_ON_DIDLE = 1'b0).

1. Enables the OSC pads if disabled and waits for the OSC stabilization (around 1ms).
2. Unmasks clock input to clock-on blocks.
3. S/W sets system initialization including GPIO register setting since normal F/F lost information due to power-gating.
4. S/W sets IO_RET_RELEASE bit of OTHERS register to 1'b1 to release retention for I/O pad.
5. S/W sets PLL initial setting (P/M/S value).
6. S/W sets EPLL_RET_RELEASE bit of OTHERS register to 1'b1 to release retention for EPLL input ports. (Refer to 2.4.3.4.1 EPLL input retention and EPLL_RET_RELEASE)

7. S/W sets to Enable the PLLs and wait for locking (about 300us).

8. S/W sets initialization configurations in LPCON (DRAM controller) to access to/from DRAM.

Table 2.4-5 IP Modules in Top Domain Having Normal F/Fs

Normal Module(1)	Base address	Normal Module(1)	Base address
Chip ID / OM	0xE000_0000	Nand Flash Controller	0xE720_0000
Clock Controller	0xE010_0000	MDMA	0xE800_0000
	0xE020_0000	Secure MDMA	0xE810_0000
GPIO	0xE030_0000	PDMA0	0xE900_0000
IEM_APC	0xE100_0000	Secure PDMA0	0xE910_0000
IEM_IEC	0xE110_0000	PDMA1	0xE920_0000
APB_D1	0xE200_0000	Secure PDMA1	0xE930_0000
APB_B1	0xE200_0000	PWM Timer	0xEA00_0000
APB_C1	0xE220_0000	Watchdog Timer	0xEA20_0000
APB_LCD	0xE230_0000	RTC_APBIF	0xEA30_0000
APB_TV	0xE240_0000	I2C	0xEC10_0000
APB_MFC	0xE250_0000	HDMI_I2C	0xEC20_0000
TZPC1	0xE280_0000	CAN0	0xEC70_0000
TZPC2	0xE290_0000	CAN1	0xEC80_0000
APB_D0	0xE300_0000	MIPI HSI TX	0xEC90_0000
APB_B0	0xE310_0000	MIPI HSI RX	0xECA0_0000
APB_C0	0xE320_0000	MIPI DSI	0xECB0_0000
APB_MEM	0xE330_0000	MIPI CSI	0xECC0_0000
APB_ASYNCBR	0xE340_0000	Modem Interface	0xED50_0000
TZPC0	0xE380_0000	SPDIF	0xF260_0000
VIC0	0xE400_0000	ADC & Touch	0xF300_0000
VIC1	0xE410_0000	Key Pad	0xF310_0000
VIC2	0xE420_0000	AHB_RX	0xF400_0000
TZIC0	0xE500_0000	AHB_TX	0xF410_0000
TZIC1	0xE510_0000	DMA_RX	0xF420_0000
TZIC2	0xE520_0000	DMA_TX	0xF430_0000
APB_DMC	0xE600_0000	Security Key	0xF500_0000
SROM Controller	0xE700_0000	SDM	0xF510_0000
OneNAND Ccontroller	0xE710_0000	Coresight	0xF600_0000

1) The state of SFR in Normal module is not kept in top domain off. Therefore, you should save the state in external DRAM before entry to power down mode (top off) if necessary.

3.4.1 EPLL Input Retention and EPLL_RET_RELEASE

In DEEP-IDLE / DEEP-STOP mode (TOP domain off), P/M/S value to EPLL should be kept since CLKCON module is power-off. Therefore, EPLL_RET signal is asserted for retention circuit to keep these values. (See

Figure 2.4-1 EPLL operation diagram).

After wakeup from DEEP-IDLE / DEEP-STOP mode, P/M/S setting to EPLL_CON(@0xE010_0108) should be done, and EPLL_EN should be enabled by setting ENABLE[31] to 1'b1 in that register. After these setting is done, EPLL_RET should be released by setting EPLL_RET_RELEASE[30] to 1'b1 in OTHERS register.

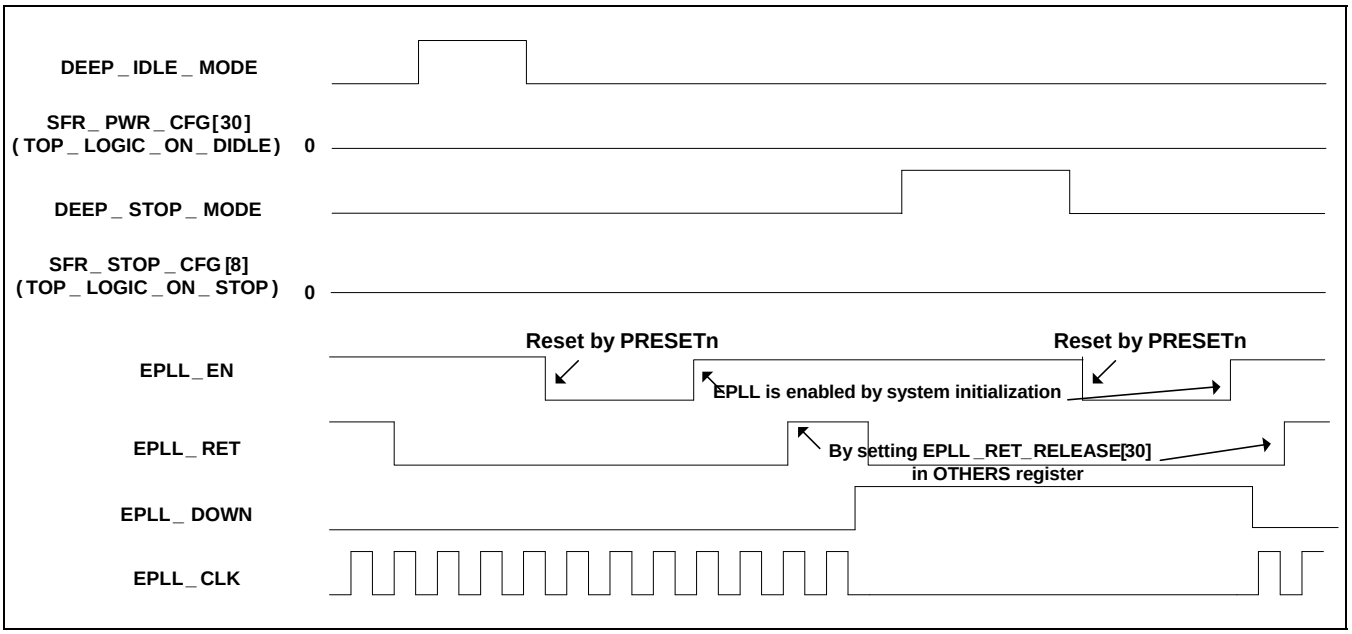


Figure 2.4-1 EPLL Operation Diagram

3.5 STOP MODE

In STOP mode, clock to modules except RTC, and ALIVE module, is disabled, PLLs are disabled, and unnecessary oscillators are selectively disabled so that dynamic power consumption is minimized. In this mode, Cortex-A8 Core enters into Standby mode. Therefore, current application program that was running in NORMAL mode, stops in STOP mode and waits for wakeup event to resume.

To enter STOP mode:

1. Set CFG_STANDBYWFI field of PWR_CFG to 2'b10.
2. Set PMU_INT_DISABLE bit of OTHERS register to 1'b1 to prevent interrupts from occurring while entering STOP mode.
3. Execute Wait For Interrupt instruction (WFI).

PMU performs the following sequence to enter STOP mode.

1. Completes all active bus transactions.
2. Completes all active memory controller transactions.
3. Initiates external DRAM to enter self-refresh mode (to preserve DRAM contents).
4. Mask input clock using internal signal in PMU.
5. Disables all PLLs.
6. Selectively disables OSCs except 32.768kHz.

To exit STOP mode:

- Various types of wakeup sources are used. Wakeup sources referred in Section 5 WAKEUP SOURCES.

Then PMU performs the following sequence to exit STOP mode.

1. Enables the OSC pads if disabled and waits for the OSC stabilization (around 1ms).
2. Enables the PLLs and waits for locking (300us).
3. Unmasks clock input to clock-on blocks.
4. Make DRAMs exit self-refresh mode.

OSC stabilization time is determined by the external clock frequency and the counter value specified in the OSC_STABLE register. PLL locking time is set in PLL_LOCK registers.

3.6 DEEP-STOP MODE

In DEEP-STOP mode, Cortex-A8 Core is power-off rather than clock-gated, and the remaining parts of the chip are power-off except TOP, RTC, and ALIVE module. The TOP domain is either power-on or power-off by the setting of TOP_LOGIC_ON field of STOP_CFG register before it enters into DEEP-STOP mode. Cortex-A8 L2 cache can be powered ON for memory retention, or power-off to save power.

- TOP_LOGIC_ON = 1'b1, Top domain is power on.
- TOP_LOGIC_ON = 1'b0, Top domain is power-off.

If TOP_LOGIC_ON_DIDLE = 1'b0, some IP modules in Top domain loses their states in F/F, and the other IP modules keeps their states in F/F after wakeup. IP modules in Top domain shown in Table 2.4-4 have retention F/Fs, and those modules will keep their states in F/F after wakeup. Therefore, IP modules that do not keep their states in F/F in DEEP-STOP mode (Top domain off), the states of F/F should be saved into external memory, and be restored after wakeup.

To enter DEEP-STOP mode:

1. Set ARM_LOGIC_ON field of STOP_CFG register to 1'b0.
2. Set CFG_STANDBYWFI field of PWR_CFG to 2'b10.
3. Set PMU_INT_DISABLE bit of OTHERS register to 1'b1 to prevent interrupts from occurring while entering DEEP-STOP mode.
4. Execute Wait For Interrupt instruction (WFI).

PMU performs the following sequence to enter DEEP-STOP mode.

1. Completes all active bus transactions.
2. Completes all active memory controller transactions.
3. Initiates external DRAM enter self-refresh mode (to preserve DRAM contents).
4. Masks input clock using internal signal in PMU.
5. Disables all PLLs.
6. Selectively disables OSCs except 32.768kHz.

To exit DEEP-STOP mode:

- Various types of wakeup sources are used. Wakeup sources referred in Section 5 WAKEUP SOURCES.

Then PMU performs the following sequence to exit DEEP-STOP mode (TOP_LOGIC_ON = 1'b1).

1. Enables the OSC pads if disabled and wait for the OSC stabilization (around 1ms).
2. Enables the PLLs and wait for locking (300us).
3. Unmasks clock input to clock-on blocks.
4. Initiates DRAMs to exit self-refresh mode.

PMU performs the following sequence to exit DEEP-STOP mode (TOP_LOGIC_ON = 1'b0).

1. Enables the OSC pads if disabled and wait for the OSC stabilization (around 1ms).
2. Unmasks clock input to clock-on blocks
3. S/W sets system initialization including GPIO register setting since normal F/F lost information due to power-gating.
4. S/W sets IO_RET_RELEASE bit of OTHERS register to 1'b1 to release retention for I/O pad.
5. S/W sets PLL initial setting (P/M/S value).
6. S/W sets to Enable the PLLs and wait for locking (about 300us).
7. S/W sets initialization configurations in LPCON (DRAM controller) to access to/from DRAM.

3.7 SLEEP MODE

In SLEEP mode, all power domains are powered down except for ALIVE and RTC, all PLLs are disabled, and the oscillators (OSCs) except that for the RTC are selectively disabled. Since the external regulator becomes off using control signal from S5PC100, you should consider waiting time for the regulator stabilization in the SLEEP mode wake-up using PWR_STABLE register.

To enter SLEEP mode:

1. Set NORMAL_CFG[5] to 1'b0 so that Audio domain becomes power-off to prevent current leakage from occurring in I2S0 related I/O pad (Xi2s0CDCLK, Xi2s0LRCK, Xi2s0SCLK, Xi2s0SDI, Xi2s0SDO[2:0]).
2. Check that BLK_PWR_STAT[5] becomes 1'b0 to confirm Audio domain is power-off.
3. Set CFG_STANDBYWFI field of PWR_CFG to 2'b11.
4. Set PMU_INT_DISABLE bit of OTHERS register to 1'b1 to prevent interrupts from occurring while entering SLEEP mode.
5. Execute Wait For Interrupt instruction (WFI).

Then PMU performs the following sequence to enter SLEEP mode.

1. Completes all active bus transactions.
2. Completes all active memory controller transactions.
3. Initiates the external DRAM enter self-refresh mode (to preserve DRAM contents).
4. Power down all power domains except the already power-down domains
5. Disables all PLLs.
6. Selectively disables OSCs except 32.768kHz.
7. XPWRRGTON becomes low to power off external voltage regulator.

To exit SLEEP mode:

- Various types of wakeup sources are used. Wakeup sources referred in Section 5 WAKEUP SOURCES.

Then PMU performs the following sequence to exit SLEEP mode.

1. Asserts wake-up reset to low.
2. XPWRRGTON becomes high to power on external voltage regulator.
3. Waits for voltage regulator to be stable.
4. Enables the OSC pads if disabled and waits for the OSC stabilization (around 1ms.)
5. Power up all power domains except power domains which was already in power-off state before entering the SLEEP Mode.
6. Release wake-up reset.

After SLEEP mode wakeup, s/w should perform the following sequence:
1. S/W sets system initialization including GPIO register setting since normal F/F lost information due to power-gating.
2. S/W sets IO_RET_RELEASE bit of OTHERS register to 1'b1 to release retention for I/O pad.
3. S/W sets PLL initial setting (P/M/S value).
4. S/W sets to Enable the PLLs and wait for locking (about 300us).
5. S/W sets initialization configurations in LPCON (DRAM controller) to access to/from DRAM.

Since all modules are powered off and their states are not preserved in SLEEP mode, you must save and restore necessary state information before and after SLEEP mode.

Note that after wakeup from SLEEP mode, S/W should IO_RET_RELEASE bit of OTHERS register to 1'b1 to release retention for I/O pad after setting GPIO registers, and NORMAL_CFG[5] to 1'b1 to access to/from Audio domain.

4 SYSTEM POWER MODE TRANSITION

Figure 2.4-2 shows the power mode transition diagram.

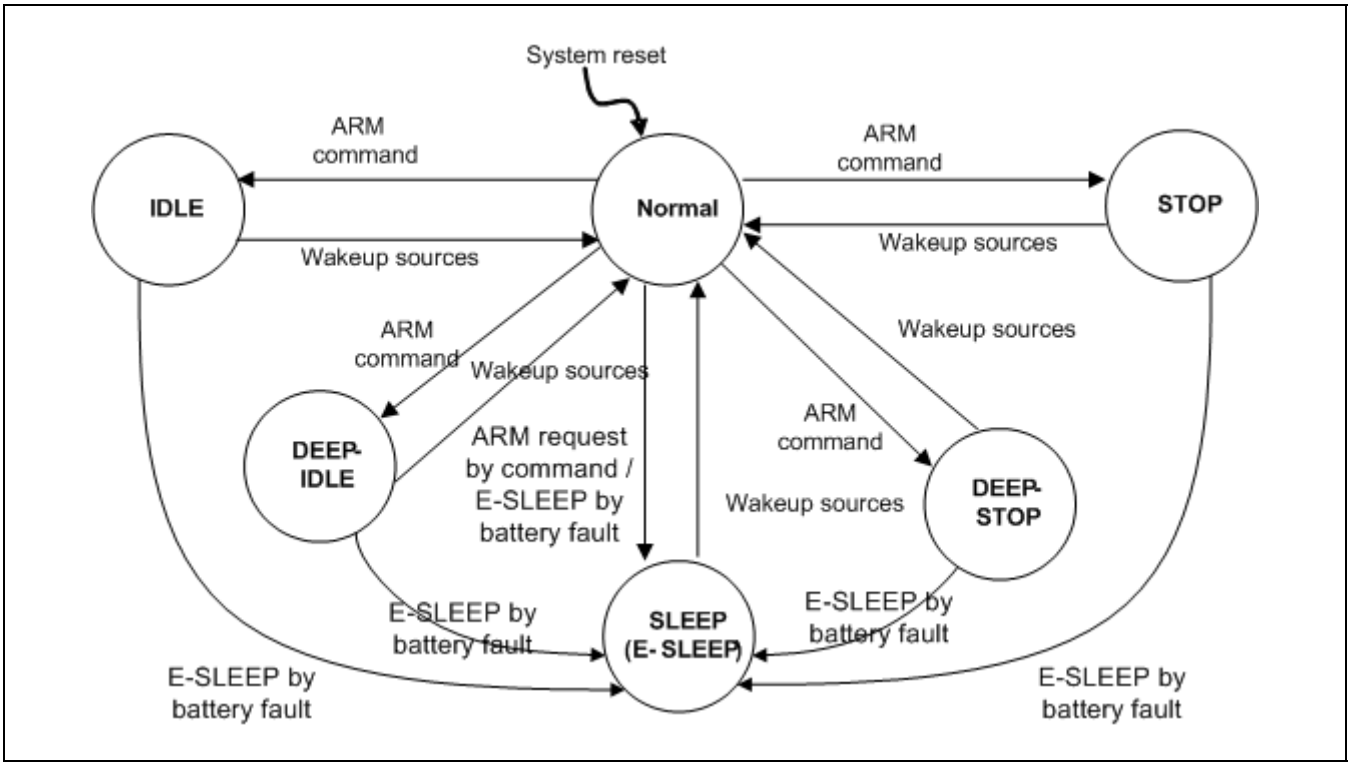


Figure 2.4-2 Power Mode Transition Diagram

The wakeup sources described in Figure 2.4-2 are summarized in Table 2.4-6.

4.1 TRANSITION ENTERING / EXITING CONDITION

Table 2.4-6 shows the Power Saving mode state and Entering or Exiting condition. As you can see, the entering conditions are set by the main ARM CPU.

Table 2.4-6 Power Saving Mode Entering/Exiting Condition

Power Saving Mode	Enter	Exit
Clock Off in NORMAL	Set the Clock-disable Bit for each IP block by s/w	Clear the Clock-disable Bit for each IP block by s/w
IDLE	1. CFG_STANDBYWFI in PWR_CFG = 2'b01 2. CFG_DEEP_IDLE of PWR_CFG = 1'b0. 3. PMU_INT_DISABLE in OTHERS = 1'b1 4. ARM Command (WFI)	1. All interrupt sources
DEEP-IDLE (Top domain on)	1. CFG_STANDBYWFI in PWR_CFG = 2'b01 2. CFG_DEEP_IDLE in PWR_CFG = 1'b1. 3. TOP_LOGIC_ON_DIDLE in PWR_CFG = 1'b1 4. PMU_INT_DISABLE in OTHERS = 1'b1 5. ARM Command (WFI)	
DEEP-IDLE (Top domain off)	1. CFG_STANDBYWFI in PWR_CFG = 2'b01 2. CFG_DEEP_IDLE in PWR_CFG = 1'b1 3. TOP_LOGIC_ON_DIDLE in PWR_CFG = 1'b0 4. PMU_INT_DISABLE in OTHERS = 1'b1 5. ARM Command (WFI)	1. nBATF (1) 2. External Interrupt 3. RTC Alarm 4. RTC TICK 5. Key Pad Press event 6. HSI interface (RX) 7. Modem I/F Wake-up event 8. MMC0~2 9. Touch Screen Pen-down event 10. I2S in audio sub-domain wake-up event 11. System Timer event
STOP	1. CFG_STANDBYWFI in PWR_CFG = 2'b10 2. PMU_INT_DISABLE in OTHERS = 1'b1 3. ARM Command (WFI)	1. nBATF(1) 2. External Interrupt 3. RTC Alarm 4. RTC TICK 5. Key Pad Press event 6. HSI interface (RX) 7. Modem I/F Wake-up event 8. MMC0~2 9. Touch Screen Pen-down event 10. System Timer event
DEEP-STOP	1. CFG_STANDBYWFI in PWR_CFG = 2'b10 2. ARM_LOGIC_ON in STOP_CFG = 1'b0 3. PMU_INT_DISABLE in OTHERS = 1'b1 4. ARM Command (WFI)	

Power Saving Mode	Enter	Exit
SLEEP	1. CFG_STANDBYWFI in PWR_CFG = 2'b11 2. PMU_INT_DISABLE in OTHERS = 1'b1 3. ARM Command (WFI)	1. nBATF(1) 2. External Interrupt 3. RTC Alarm 4. RTC TICK 5. Key Pad Press event
E-SLEEP(2) (Emergency SLEEP)	1. BATF_WAKEUP_MASK in PWR_CFG = 1'b0 2. CFG_BATFLT = 2'b11 3. Battery fault (XnBATF = 0)	1. External Interrupt 2. RTC Alarm 3. RTC TICK 4. Key Pad Press event

1. nBATF: Low Battery Interrupt from XnBATF pad.
2. Basically, E-SLEEP mode is the same as SLEEP mode, but we distinguish E-SLEEP mode from SLEEP mode in that entering condition and wakeup sources are different.
   The Wake-up Event Source for E-SLEEP mode can be optionally limited to be just none, or it can be any of those listed in the above table. See PWR_CFG register's CFG_BATF_WKUP field for more details.

Power mode exit condition is met if one of various wakeup sources occurs. For detailed descriptions of wakeup sources refer to Section 5. WAKEUP SOURCES.

5 WAKEUP SOURCES

Table 2.4-7 Summary of Power Mode Wakeup Sources

<table>
<tr><th colspan="3">Power mode</th><th>Wakeup sources</th></tr>
<tr><td colspan="3">IDLE</td><td>IRQ (signal from ARM to VIC)</td></tr>
<tr><td rowspan="11">DEEP-IDLE</td><td colspan="2"></td><td>I2S (in audio domain)</td></tr>
<tr><td rowspan="10">STOP / DEEP-STOP</td><td></td><td>MMC0, MMC1, MMC2</td></tr>
<tr><td></td><td>TS_PENDN</td></tr>
<tr><td></td><td>MODEM_IF</td></tr>
<tr><td></td><td>HSI</td></tr>
<tr><td></td><td>System Timer</td></tr>
<tr><td rowspan="5">SLEEP</td><td>External interrupt sources(1)</td></tr>
<tr><td>RTC Alarm</td></tr>
<tr><td>TICK</td></tr>
<tr><td>Key</td></tr>
<tr><td>Battery Fault</td></tr>
</table>

(1) If an external interrupt source is used as a wakeup source, the corresponding wakeup mask bit should be unmasked in EINT_WAKEUP_MASK in Chapter 2.4 Power Management and WKUP_INT function should be selected in GPH[0~3]CON and delay filter should be enabled in WKUP_INT[0~3]_FLTCON[0~1] in Chapter 2.2 Pad Control.

5.1 INTERNAL INTERRUPTS

All internal interrupts are wakeup sources in IDLE or DEEP_IDLE (Top domain on) mode.

5.2 EXTERNAL INTERRUPTS

External interrupts are the common wake-up source of all power down modes. The logic for external interrupt configuration such as polarity, edge/level sensitivity, and masking resides in the GPIO and can be modified through GPIO register setting before entering one of power down modes mentioned above. For more details, refer to 2.2 GPIO manual.

The external interrupt handling logic holds the external interrupt information until Cortex-A8 clears it. It allows Cortex-A8 to handle the external interrupt after wake up.

5.3 MODEM_IF

The Base-band Modem Chip can wakeup Application Processor such as S5PC100 in power down mode as specified in Table 2.4-7. For more information of detailed wakeup sequence, refer to Chapter 8.11 MODEM INTERFACE manual.

5.4 RTC ALARM

The Real Time Clock (RTC) has 32-bit counter used to wake up the system after the specified time. If the timer alarm happens, PMU wakes up the system and sets the RTL_ALARM_WAKEUP field of WAKEUP_STAT register to 1. After the wake-up, Cortex-A8 refers the WAKEUP_STAT register to find out what caused the wake-up.

5.5 TICK

The Real Time Clock (RTC) has tick counter function to be used for wakeup source from power down mode as specified in Table 2.4-7. In NORMAL mode, RTC time tick may be used for real time operating system (RTOS) kernel time tick.

5.6 SYSTEM TIMER

System Timer is newly introduced module in S5PC100 to supplement the disadvantage of the existing PWM timer.

In IDLE mode, the existing PWM timer generates interrupts at fixed interval using auto-reload function, so it wakes up the chip too often. If the PWM timer generates an interrupt at various interval using one-shot mode, but it accumulates the timing inaccuracy since it cannot count time during manual setting, and therefore it cannot generate accurate 1ms timing tick to be used for OS operation.

In DEEP-IDLE, STOP, and DEEP-STOP mode, there is no system clock if Top domain is power-off. Therefore, RTC is used for generating timing tick instead of PWM timer, but by using this clock, timing count is not controlled to meet exact 1ms OS time tick since RTC clock does not have high resolution.

On the other hand, System timer has the function of generating interrupts at various interval, and do not require manual setting, so it does not wakes up the chip too often, and provides accurate 1ms timing ticks. System timer uses one of an external crystal clock , RTC clock and the generated clock from CLKCON as clock input.

For System Timer to operate in DEEP-IDLE, STOP, and DEEP-STOP mode, power to System Timer is on. The wakeup event from System Timer will wake up S5PC100 from DEEP-IDLE, STOP, and DEEP-STOP mode.

6 HARD MACRO POWER CONTROL

Table 2.4-8 shows the hard macro power control summary.

Table 2.4-8 S5PC100 Hard Macro Power Control

	Hard macro	Controlled by	NORMAL(1)	IDLE/ DEEP-IDLE(2)	DEEP-IDLE(3) / STOP / DEEP-STOP	SLEEP
1	SRAM	SRAM control logic / PMU	Run / Stand-by	Keep power state in NORMAL	Stand-by Retention / Power-down	External power off
2	ROM	ROM control logic / PMU	Run / Stand-by	Keep power state in NORMAL	Stand-by	External power off
3	USB OTG phy(6)	USB OTG link	Run / IDLE / Suspend	Keep power state in NORMAL	Suspend	External power off
4	HDMI phy(6)	HDMI link	Run / Power-down	Keep power state in NORMAL	Power-down	External power off
5	MIPI D-phy(4) (6)	MIPI link	Run / LP / ULPS	Keep power state in NORMAL	LP / ULPS	External power off
6	PLL(6)	PMU / CLKCON	Run / Power-down	Keep power state in NORMAL	Power-down	External power off
7	DAC(6)	TV Encoder logic	Run / Power-down	Keep power state in NORMAL	Power-down	External power off
8	TS ADC(6)	PMU	Run	Stand-by	Stand-by	External power off / Internal power off
9	Digital I/O	PMU	Power-on	Power-on	Power-on	Power-on

1. External power to USB OTG phy, HDMI phy, MIPI D-phy and DAC can be off to prevent current leakage if they are not used in normal mode. In this case, you should set the following bits in OTHERS register to 1'b0: MIPI_DPHY_EN[28], DAC_EN[26], and USB_SIG_MASK[16]. But External power to PLL and TS ADC should be supplied even if they are not used in normal mode.
2. Top domain on
3. Top domain off
4. MIPI D-phy includes TX phy for DSI and RX phy for CSI.
5. For more information refer to "Section 6.9 Digital I/O".
6. Analog I/O is used to access to/from hard macros in S5PC100, and analog I/O power is supplied by separate analog I/O power pin.

6.1 SRAM

SRAM in Top domain has four power modes: Run, Stand-by, Retention, and Power-down mode.

- In Run mode, read and write access to SRAM are performed normally.
- In Stand-by mode, SRAM chip select is deactivated, so that there is no read and write access.
- In Retention mode, power is provided to only core of SRAM, and power to peripheral circuitry is off internally.
- In Power-down mode, all power to core and peripheral circuitry is off.

The list of SRAM in Top domain is as follows: CCAN, IRDA, MODEM IF, USB_OTG, SDMMC, CSSYS, SECSS, internal RAM 0, 1, and 2. (Refer to STOP_MEM_CFG register in Section 10 REGISTER DESCRIPTION)

In NORMAL mode, run, or stand-by mode is used.

Run mode is used if there is read and write access, while stand-by mode is used if there is no read and write access. The change between these two modes is done by corresponding SRAM control logic.

In IDLE mode, and DEEP-IDLE mode (top domain on), SRAM keeps its operation or power state in NORMAL.
In DEEP-IDLE mode (top domain is off), SRAM in TOP module enters stand-by, retention, or power-down mode. Before entry to this mode, you must set the TOP_MEMORY_ON_DIDLE and TOP_MEMORY_RET_ON_DIDLE field PWR_CFG in PMU.
In STOP mode and DEEP-STOP mode, stand-by, retention, and power-down mode can be entered.
Before entry to STOP mode, you must set the TOP_MEMORY_ON and TOP_MEMORY_RET field of STOP_CFG register in PMU to determine which power mode SRAM enters during STOP mode. And each SRAM can be separately set in STOP_MEM_CFG register.
In SLEEP mode, power to SRAM is off, so the data in SRAM is lost. Power mode of SRAM in SLEEP mode has no meaning.

6.2 ROM

ROM has two power modes: Run, and Stand-by mode.

- In Run mode, read access to ROM is performed normally.
- In Stand-by mode, chip selection to ROM is deactivated, so that there is no read access.

In NORMAL mode, both power modes are used. Run mode is used if there is read access, while Standby-mode is used if there is no read access. The change between these two modes is done by ROM control logic.

In IDLE mode and DEEP-IDLE mode (top domain on), ROM keeps its operation or power state in NORMAL.

To enter DEEP-IDLE mode (top domain is off), STOP mode and DEEP-STOP mode, s/w should not access ROM so that ROM enter Stand-by mode.

In SLEEP mode, power to ROM is off. Power mode of ROM in SLEEP mode has no meaning.

6.3 USB OTG PHY

Recommended power up sequence is shown in Figure 2.4-3.

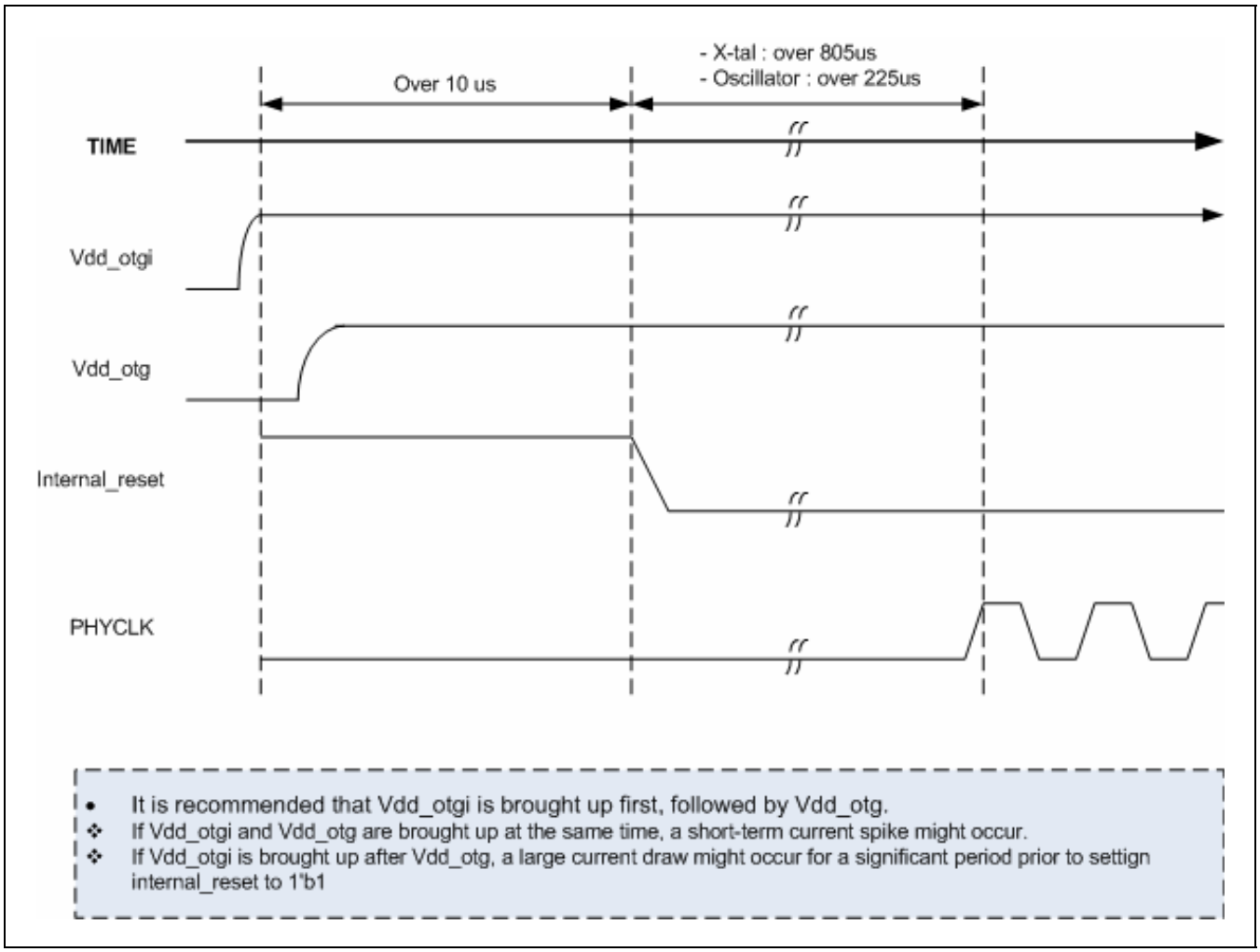


Figure 2.4-3 Power Up Sequence of USB OTG Phy

USB OTG phy has three power modes: Run, IDLE, and Suspend mode.

- In Run mode, USB OTG phy sends and receives data normally.
- In IDLE mode, there is no data transaction to and from USB OTG phy. But, the clock is still supplied to USB OTG phy.
- In Suspend mode, USB OTG phy clock is off to save power.

In NORMAL mode, all the three power modes can be used.

If USB OTG phy is in use, then it is in Run mode if there is data transaction, and in IDLE mode if there is no data transaction.
The change between these two modes is done by USB OTG link.

If USB OTG phy is not in use, then it enters into Suspend mode.
You should follow power down sequence for USB OTG link for Entry to, and Exit from Suspend mode.

In IDLE mode, and DEEP-IDLE mode (top domain on), USB OTG phy keeps its operation or power state in NORMAL.

Before entry to DEEP-IDLE mode (top domain off), STOP, and DEEP-STOP, it is recommended that USB OTG phy enters into Suspend mode.

In SLEEP mode, external power to USB OTG phy should be off. Power down mode of USB OTG phy in SLEEP mode has no meaning.

6.4 HDMI PHY

HDMI phy has two power modes: Run and Power-down mode

- In Run mode, HDMI phy sends and receives data normally.
- In Power-down mode, all power to HDMI phy is off internally.

In NORMAL mode, both the power modes can be used.

If HDMI phy is in use, then it is in Run mode.
If HDMI phy is not in use, it can enter into Power-down mode to save static power by setting register in HDMI link.

In IDLE mode and DEEP-IDLE mode (top domain on), HDMI phy keeps its operation or power state in NORMAL.

Before entry to DEEP-IDLE mode (top domain off), STOP, and DEEP-STOP mode, it is recommended that HDMI phy enters into Power-down mode.

In SLEEP mode, external power to HDMI phy should be off. Power down mode of HDMI phy in SLEEP mode has no meaning.

6.5 MIPI D-PHY

MIPI D-phy includes TX phy for DSI and RX phy for CSI.

MIPI D-phy has three power modes: Run, LP, and ULPS mode

- In Run mode, MIPI D-phy sends and receives data normally.
- In LP and ULPS mode, all power MIPI D-phy is off internally.

In NORMAL mode, all the three power modes can be used.

If MIPI D-phy is in use, then it is in Run mode.
If MIPI D-phy is not in use, it enters into LP or ULPS mode to save static power by setting register in MIPI link.

In IDLE mode and DEEP-IDLE mode (top domain on), MIPI D-phy keeps its operation or power state in NORMAL.

Before entry to DEEP-IDLE mode (top domain off), STOP, and DEEP-STOP mode, it is recommended that MIPI D-phy enters into LP or ULPS mode.

In SLEEP mode, external power to MIPI D-phy should be off. Power down mode of MIPI D-phy in SLEEP mode has no meaning.

6.6 PLL

PLL has two power modes: Run and Power-down mode

- In Run mode, PLL sends and receives data normally.
- In Power-down mode, all power to PLL is off internally.

In NORMAL mode, both power modes are used.

If PLL is in use, then it is in Run mode.
If PLL is not in use, it enters into Power-down mode to save static power by setting register (APLLCON, MPLLCON, EPLLCON, and HPLLCON) in CLKCON.

In IDLE mode, and DEEP-IDLE mode (top domain on), PLL keeps its operation or power state in NORMAL.

In DEEP-IDLE mode (top domain off), APLL, MPLL, and HPLL are powered down automatically by PMU Note that EPLL is still powered on in this mode for providing proper operating clock to Audio sub-domain.
In STOP mode, all PLLs are powered down automatically by PMU.

In SLEEP mode, external power to PLL is off with internal logic power. Therefore, power down mode of PLL in SLEEP mode has no meaning.

6.6.1 Status of PLL after Wake-Up Event

The behaviors of PLLs (APLL, MPLL, HPLL, EPLL) after wakeup from various power down modes are different. Table 2.4-9 shows the states of PLLs and HCLK after wake-up from the power-saving modes.

During IDLE mode and DEEP-IDLE mode (top domain on), PLLs (APLL, MPLL, HPLL) are running normally by the same configurations as those before power down modes. Therefore after wakeup, PLLs (APLL, MPLL, HPLL) are also running unchanged.

During STOP mode and DEEP-STOP mode (top domain on), PLLs (APLL, MPLL, HPLL) turned off for power saving. After wakeup, PLLs (APLL, MPLL, HPLL) are automatically turned on by hardware logic to the same configurations, and HCLK becomes PLL clock output after PLL locking time.

During DEEP-IDLE mode (top domain off), DEEP-STOP mode (top domain off) and SLEEP mode, PLLs (APLL, MPLL, HPLL) turned off for power saving. After wakeup, PLLs (APLL, MPLL, HPLL) should be turned on by s/w since PLL configurations are lost due to power off of top domain. Therefore, HCLK becomes PLL reference clock after wakeup.

The behavior of EPLL in DEEP-IDLE (top domain off) is different from those of the other PLLs (APLL, MPLL, HPLL). In DEEP-IDLE (top domain off), EPLL is used to generate clock to Audio domain for low-power MP3 audio playback, therefore EPLL is running by the same configurations as those before power down mode.

The behavior of EPLL in the other power down modes, is the same as the other PLLs (APLL, MPLL, HPLL).

The initial-state of S5PC100 after wake-up from the SLEEP mode is almost the same as the Power-On-Reset state except for the contents of the external DRAM is preserved.

On the other hand, S5PC100 automatically recovers the previous working state after wake-up from the STOP mode, and DEEP-STOP (top domain on).

Finally, after wake-up from the DEEP-IDLE (top domain off), DEEP-STOP (top domain off) and SLEEP mode, s/w should recover the previous working state by using preserved information.

Table 2.4-9 Status of PLLs and ARMCLK/HCLK after Wake-Up

Mode before wake-up	APLL/MPLL/HPLL on/off after wake up	EPLL on/off after wakeup	HCLK after wakeup
IDLE	unchanged	unchanged	PLL Output
DEEP-IDLE (Top domain on)	unchanged	unchanged	PLL Output
DEEP-IDLE (Top domain off)	off → off	unchanged	PLL reference clock
STOP	off → on (by H/W)	off → on (by H/W)	PLL Output
DEEP-STOP (Top domain on)	off → on (by H/W)	off → on (by H/W)	PLL Output
DEEP-STOP (Top domain off)	off → off	off → off	PLL reference clock
SLEEP	off → off	off → off	PLL reference clock

6.7 DAC

DAC has two power modes: Run and Power-down mode

- In Run mode, DAC sends and receives data normally.
- In Power-down mode, all power to DAC is off internally.

In NORMAL mode, both power modes can be used.
If DAC is in use, then it is in Run mode.
If DAC is not in use, it enters into Power-down mode to save static power by setting register in TVOUT logic.

In IDLE mode and DEEP-IDLE mode (top domain on), DAC keeps its operation or power state in NORMAL.

Before it enters to DEEP-IDLE mode (top domain off), STOP and DEEP-STOP, it is recommended that DAC enters into Power-down mode.

Before it enters to SLEEP mode, power to DAC should be off to prevent leakage current.

6.8 TS ADC

In NORMAL mode, TS ADC is in Run mode. External power to TS ADC has to be supplied even if they are not used in normal mode.
In IDLE mode, DEEP-IDLE mode, STOP and DEEP-STOP, TS ADC is in Stand-by mode, and therefore it is normally used since TS_PENDN signal from TS ADC is used to wakeup source in these power mode.

In SLEEP mode, external power to TS ADC can be off. Power mode of TS ADC in SLEEP mode has no meaning.

6.9 DIGITAL I/O

I/Os used in S5PC100 are divided into two groups, i.e., digital I/Os and analog I/Os, and digital I/Os are also divided into normal I/Os and alive I/Os.

All digital I/Os have internal power (1.2V) and I/O power (1.8~3.3V). Internal power is supplied separately for normal I/Os and alive I/Os. That is, internal power for alive I/Os should be always supplied, but internal power for normal I/Os should be always supplied except SLEEP mode. I/O power should be always supplied in any power mode.

Alive I/Os and normal I/Os should be always supplied. Alive I/Os are listed in Table 2.4-10.

In DEEP-IDLE mode (top domain off), normal I/Os except I2S related I/O do not work since top domain is power-off.
In DEEP-STOP mode (top domain off), all normal I/Os do not work.
In SLEEP mode, normal I/O does not work since internal power to normal I/O is off, but alive I/O always works since power to alive I/O is always on. Nontheless, I/O power to normal I/O should be supplied such as VDDQ_DDR, VDDQ_M0, VDDQ_LCD, VDDQ_CI, VDDQ_MMC, VDDQ_AUD, VDDQ_MSM, VDDQ_SYS0, VDDQ_SYS2, VDDQ_SYS5, VDDQ_CAN, VDDQ_EXT, VDDQ_RTC, VDDQ_UH.

Table 2.4-10 List of Alive I/O

ALIVE I/O	XEINT[31:0], XPWRRGTON, XOM[4:0], XnRESET, XnWRESET, XnBATF

6.9.1 OUTPUT PORT State in DEEP-IDLE, DEEP-STOP and SLEEP mode

In DEEP-IDLE mode (top domain off) and DEEP-STOP mode (top domain off), the output port of normal I/O keeps it's driving value before it enters DEEP-IDLE/DEEP-STOP mode. Normal I/O has output retention function, and it keeps its driving value by using latch. The retention control signal to input port (CLTCH, CPGI) of normal I/O is generated by PMU when entering DEEP-IDLE/DEEP-STOP mode.

Alive I/O also keeps its driving value from power-off region before it enters DEEP-IDLE/DEEP-STOP mode. PMU generates retention control signal (CPGI).

In SLEEP mode, internal power to normal I/O is off, and I/O power to normal I/O is still on.

Alive I/O changes its output path from Normal path (power-off region) to ALIVE path (ALIVE module). ALIVE module drives output value of alive I/O in SLEEP mode. Read value from alive I/O goes to ALIVE module. This read values acts as wakeup source in SLEEP mode.

6.9.2 Retention I/O, IO_RET_RELEASE, and SDMMC_IO_RET_RELEASE

In DEEP-IDLE mode (top domain off), DEEP-STOP mode (top domain off) and SLEEP mode, GPIO setting to normal I/O is lost due to power off. (See Figure 2.4-4) Therefore, these setting should be saved before power to top domain is off (nSCALL_BLK_TOP = 1'b0), and restored after wakeup from power down mode if necessary.

Before entry to DEEP-IDLE (top off), DEEP-STOP (top off), and SLEEP mode, I/O setting is switched from GPIO normal mode configuration register (GP*CON) to GPIO power down mode configuration register (GP*PDNCON), and after wakeup from power down mode, I/O setting is switched from GPIO power down mode configuration register (GP*PDNCON) to GPIO normal mode configuration register (GP*CON).

Four retention signals such as RET_EN0, RET_EN1, RET_LPA, and RET_SDMMC) are asserted to keep their setting after switch from GP*CON to GP*PDNCON before entry to above power down modes, and released for I/Os to be used in Normal mode after switching from GP*PDNCON to GP*CON. (See Figure 2.4-4).

In Figure 2.4-4, XnRSTOUT is asserted during above power down modes, and released after wakeup from those power down modes.

The mapping between these retention signals and normal I/O is shown in Table 2.4-11.

After wakeup from power down mode, RET_EN0 signal is released automatically by hardware to access to/from memory. GPIO setting should be done before the other retention signals are released.

RET_EN1 and RET_LPA are released by setting IO_RET_RELEASE[31] in OTHERS register to 1'b1, and RET_SDMMC are released by setting SDMMC_IO_RET_RELEASE[22] in OTHERS register to 1'b1

Table 2.4-11 Retention control signals and Related Digital I/O(1)

Retention Control Signal	Released by	Related Digital I/O
RET_EN0	Hardware Logic(2)	M0 ports (Xm0*), M1 ports (Xm1*), XnRSTOUT
RET_EN1	Set IO_RET _RELEASE[31] to 1'b1	XXTI27,XXTO27, JTAG (Xj*), UART (Xu*), SPI (Xspi*), I2S1 (Xi2s1*), PWM (XpwmTOUT), I2C (Xi2c*), Camera Interface (Xci*), LCD (Xv*), XPKG_MODE[1:0], XCLKOUT, IEM (Xiem*), MODEM_IF (Xmsm*), XDDR2SEL, MMC2 ports (Xmmc2*), XNFMOD[5:2]
RET_LPA		Xi2s0CDCLK, Xi2s0LRCK, Xi2s0SCLK, Xi2s0SDI, Xi2s0SDO[2:0]
RET_SDMMC	Set SDMMC_IO_RET _RELEASE[22] to 1'b1	MMC0, 1 ports (Xmmc0*, Xmmc1*)

1. The following I/Os have different behavior from above retention I/Os.
   1) Digital I/O, Alive I/O, controlled by alive register : XEINT[31:0]
      This I/O is controlled by alive register, and therefore its setting is kept during and after power down mode.
   2) Digital I/O, Alive I/O, controlled by off register : XNFMOD[1:0]
      This I/O is controlled by off register, and therefore its setting should be written into corresponding register after wakeup reset. Its state during power down mode is different according to top domain on/off.
      - DEEP-IDLE (top on), STOP, DEEP-STOP (top on) : keep its value in Normal mode
      - DEEP-IDLE (top off), DEEP-STOP (top off), SLEEP : input mode (no pull up/down)
   3) Digital I/O, Alive I/O, dedicated : XOM[4:0], XPWRGTON, XnBATF, XnRESET, XXTI(XXTO),

XusbXTI(XusbXTO)
This I/O is dedicated pin, and therefore its value is kept during power down mode.

4) Analog I/O, dedicated : Xadc*, Xdac*, Xhdmi*, Xmipi*, Xusb*, Xuh*, Xef*, Xrtc*, Xcg*
This I/O is dedicated pin, and therefore its value is kept during power down mode as long as analog power is supplied.

2. RET_EN0 is automatically released by hardware logic. The release is done 1 XTI (main clock) cycle after wakeup reset to top domain is released.

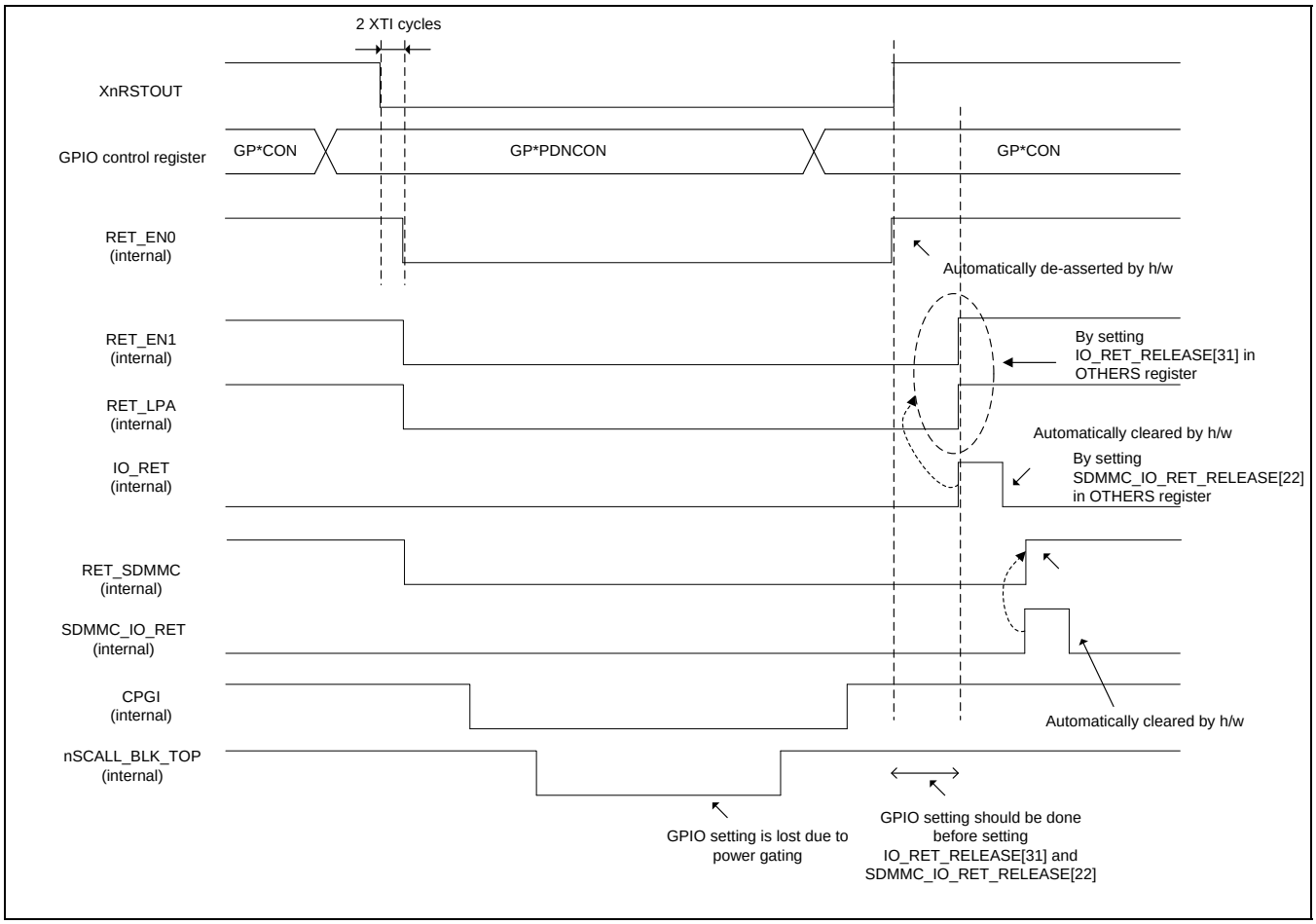


Figure 2.4-4 Retention I/O Control Timing Diagram

7 POWER ON SEQUENCE AND RESET CONTROL

S5PC100 has power-on sequence and four types of resets and reset generator places the system into one of four reset states.

- **Hardware Reset –** If XnRESET is driven to low it generates hardware reset. It is an uncompromised, ungated, total and complete reset that is used if you want to drive S5PC100 to a known initial state due to various reasons.
- **Software Reset –** Set special control register to reset signal.
- **Watchdog Reset –** Reset signal by watchdog timer
- **Wakeup Reset –** Generates Reset signal if a module that has normal F/Fs is powered down, and the module is powered up again by wakeup events; but in sleep mode, wakeup reset is generated to all modules that were powered off regardless of normal F/F or retention F/F.

7.1 POWER ON SEQUENCE

Power-on sequence starts progress after power is supplied to the S5PC100.

Figure 2.4-5 shows the clock behavior during the power-on sequence. The crystal oscillator begins oscillation within about 2~3 milliseconds after the power supply supplies enough power-level to the S5PC100. Internal PLLs become disabled after power turns on. XnRESET signal should be released after the fully settle-down of the power supply-level. For the proper system operation, the S5PC100 requires a hazard-free system clock (ARMCLK, HCLK and PCLK) when the system reset is released (XnRESET). However, since PLLs are disabled, Fin (the direct external oscillator clock) is fed directly to HCLK instead of the MPLL_CLK (PLL output) before the S/W configures the MPLLCON register to enable the operation of PLLs. If new P/M/S values are required, the S/W configures P/M/S field first, and the PLL_EN field later.

The PLL begins the lockup sequence toward the new frequency only after the S/W configures the PLL with a new frequency-value. HCLK is configured to be PLL output (MPLL_CLK) immediately after lock time.

Caution: Power regulator for system must be stable prior to the release of XnRESET. Otherwise, it damages to S5PC100 and its operation is not guaranteed.

S5PC100 has four PLLs namely: APLL, MPLL, EPLL, HPLL.

1. APLL: Used to generate ARM clock
2. MPLL: Used to generate system bus clock and several special clocks
3. EPLL: Used to generate several special clocks
4. HPLL: Used to generate HDMI phy clock

Clock timing constraints are summarized in Table 2.4-13.

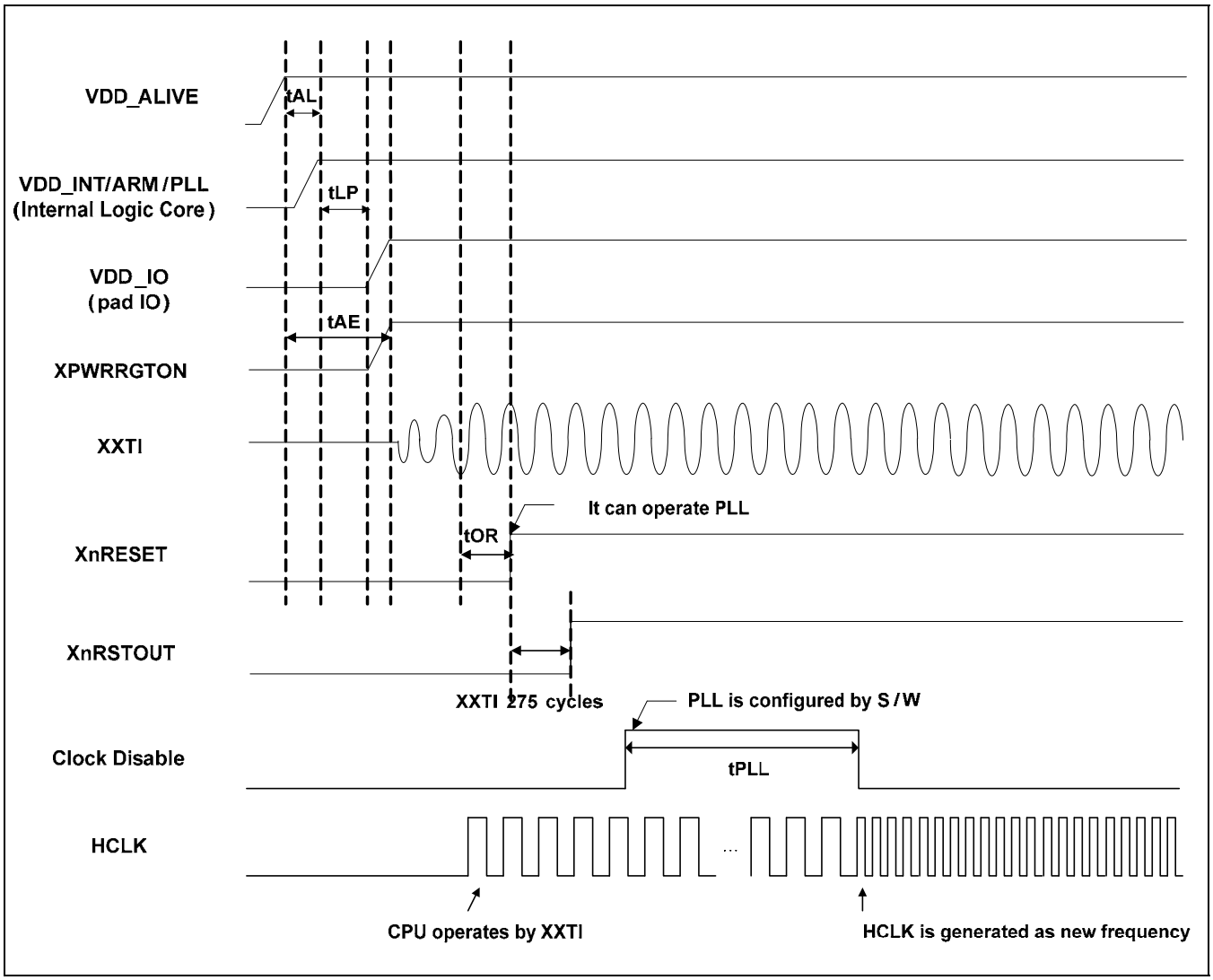


Figure 2.4-5 Power-On Sequence

NOTE: VDD_IO contains VDDQ_DDR, VDDQ_M0, VDDQ_LCD, VDDQ_CI, VDDQ_MMC, VDDQ_AUD, VDDQ_MSM, VDDQ_SYS0, VDDQ_SYS2, VDDQ_SYS5, VDDQ_CAN, VDDQ_EXT, and VDDQ_UHOST.

Table 2.4-12 Power on Timing Specifications

Symbol	Description	Min	Typical	Max	Units
tAL	VDD_ALIVE to VDD_INT/ARM/PLL	1			us
tLP	VDD_INT/ARM/PLL to VDD_IO	1			ns
tOSC	VDD_INT/VDD_ARM to Oscillator stabilization	10			cycle(1)
tOR	Oscillator stabilization to nRESET & nTRST high	1			us
tPLL	PLL locking time			300	us

NOTE : 1. The user should be aware that the crystal oscillator settle-down time is not explicitly added by the hardware during the power-on sequence. The S5PC100 assumes that the crystal oscillation is settled during the power-supply settle-down period. However, to ensure the proper operation during wake-up from the STOP

mode, the S5PC100 explicitly adds the crystal oscillator settle-down time (the wait-time is programmed using OSC_STABLE registers) after wake-up from the STOP mode.

Table 2.4-13 Clock Timing Constants

(VDD_INT= 1.2V± 0.05V, TA = -40 to 85°C, VDD_SYS0 = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min	Typ	Max	Unit
VDD_IO to VDD_ALIVE	tOA	0			ms
VDD_ALIVE to VDD_INT/VDD_ARM	tAI	1			us
VDD_ARM to PWR_EN(PWRRGTON)	tAE	1		10	ns
VDD_INT/VDD_ARM to Oscillator stabilization	tOSC	10			cycle
Oscillator stabilization to nRESET & nTRST high	tOR	1			us
External clock input high level pulse width	$t_{EXTHIGH}$	25		-	ns
External clock to HCLK (without PLL)	t_{EX2HC}	5		10	ns
HCLK (internal) to CLKOUT	t_{HC2CK}	4		10	ns
HCLK (internal) to SCLK	$t_{HC2SCLK}$	2		8	ns
Reset assert time after clock stabilization	t_{RESW}	4		-	XTIpll or EXTCLK
APLL, MPLL Lock Time	t_{PLL}	-		300	us
EPLL Lock Time		-		300	us
Sleep mode return oscillation setting time. (2)	t_{OSC2}	2^4		2^{19}	XTIpll or EXTCLK
The interval before CPU runs after nRESET is released.	$t_{RST2RUN}$	5		-	XTIpll or EXTCLK

NOTE : 2. Sleep mode return oscillation setting time depends on the value in the PWR_STABLE register.

7.2 HARDWARE RESET

Hardware reset is asserted if the XnRESET pin is driven to low, and all units in the system (except RTC function module) are reset to known states.

During the hardware reset, the following actions occur:

- All internal registers and Cortex-A8 go into their pre-defined reset state.
- All pins get their reset state, and XnBATF pin is ignored.
- The XnRSTOUT pin is asserted when XnRESET is driven.

Hardware reset is asserted when an external source drives the XnRESET input pin low. XnRESET is non-maskable, and therefore is always applicable. Upon assertion of XnRESET, S5PC100 enters into reset state regardless of the previous state. For hardware reset to be asserted actually, XnRESET must be held long enough to allow internal stabilization and propagation of the reset state.

Table 2.4-14 lists up pin names that should be supplied during reset.

Table 2.4-14 List of Power Pin Names supplied during reset

Power Pin Names supplied during reset
VDD_MCP, VDDQ_DDR, VDD_MO, VDDQ_LCD, VDDQ_CI, VDDQ_MMC, VDDQ_AUD, VDDQ_MSM, VDDQ_SYS0, VDDQ_SYS2, VDDQ_SYS5, VDDQ_CAN, VDDQ_EXT, VDDQ_UH, VDDINT, VDD_ALIVE, VDD_ARM, VDD_ADC

Manual reset input timing for hareware reset is shown in Figure 2.4-6.

EXTCLK means XXTI input (XOM[0]=0) or XusbXTI input (XOM[0]=1). Refer to Table 2.4-13 for t_{RESW} parameter.

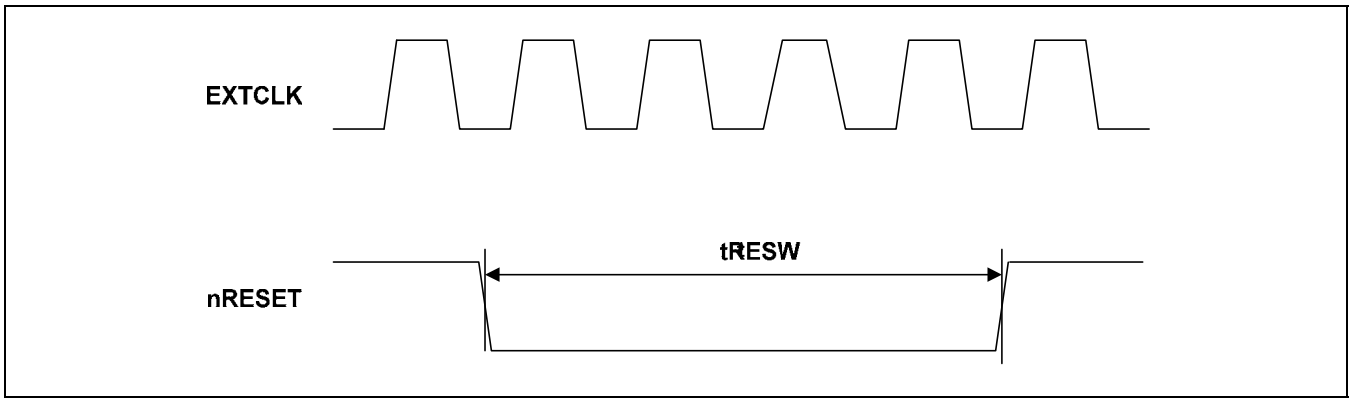


Figure 2.4- 6 Manual Reset Input Timing

7.3 SOFTWARE RESET

Software reset is asserted when S/W writes "0xC100" to SWRESET register (@0xE020_0000) in NORMAL mode.

During the software reset, the following actions occur:

- All units (except some registers listed in Table 2.4-15) go into their pre-defined reset state.
- All pins get their reset state, and XnBATF pin is ignored.
- The XnRSTOUT pin is asserted during software reset.

If Software reset is asserted the following sequence occurs.

1. PMU requests AXI masters and AHB-to-AXI bridges to finish current transactions.
2. Bus controller send acknowledge to PMU after completed bus transactions.
3. PMU request memory controller to enter self refresh mode.
4. PMU waits for self refresh acknowledge from memory controller.
5. Internal reset signals and XnRSTOUT are asserted and reset counter is activated.
6. Reset counter is expired then; internal reset signals and XnRSTOUT are released.

Timing diagram for software reset is shown in Figure 2.4-7.

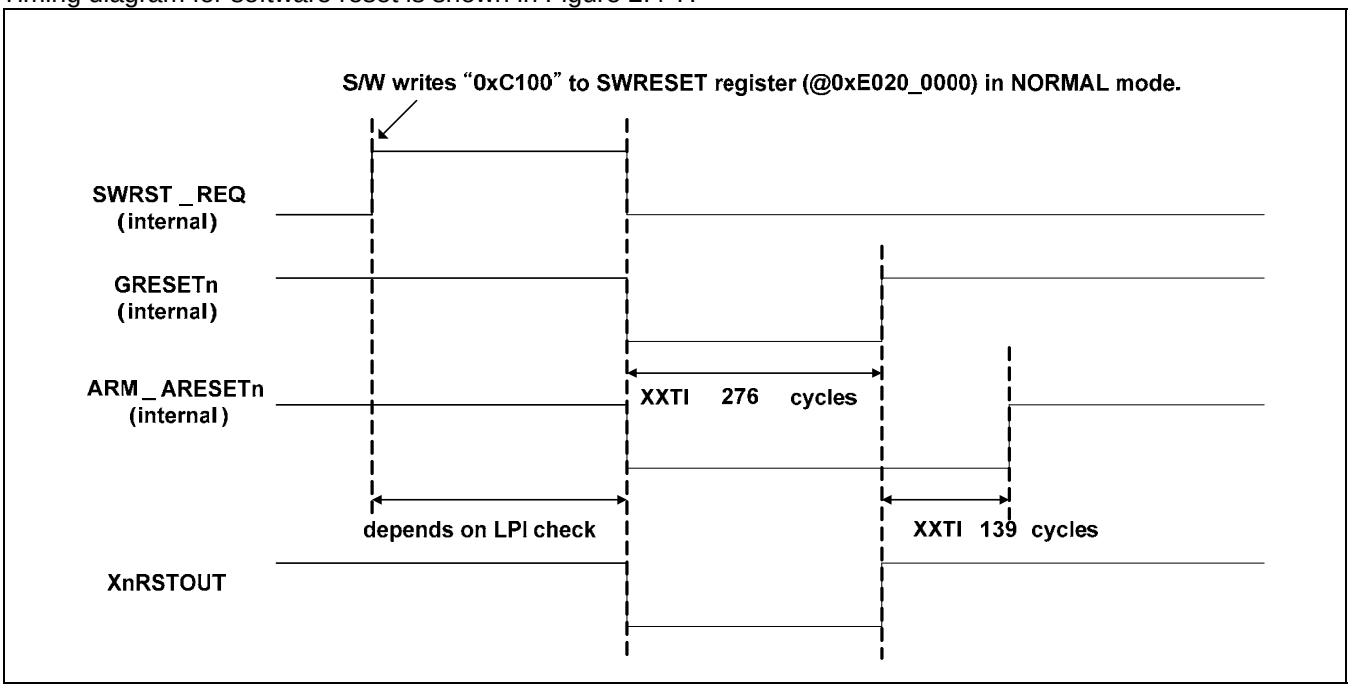


Figure 2.4-7 Software Reset Timing Diagram

7.4 WATCHDOG RESET

Watchdog reset is asserted if software fails to prevent the watchdog timer from timing out. In watchdog reset all units in S5PC100 (except some registers listed in Table 2.4-15) are reset to their predefined reset states. The behavior after Watchdog reset is asserted, is the same as Hardware reset case (Refer to "Section 7.1 HARDWARE RESET").

During the watchdog reset, the following actions occur:

- All units (except some registers listed in Table 2.4-15) go into their pre-defined reset state.
- All pins get their reset state, and XnBATF pin is ignored.
- The XnRSTOUT pin is asserted during watchdog reset.

Watchdog reset is activated in NORMAL / IDLE / DEEP-IDLE (Top domain on) mode because watchdog timer expires with clock.

Watchdog reset is asserted when watchdog timer and reset are enabled (WTCON[5] = 1, WTCON[0]=1) and watchdog timer is expired.

Watchdog reset is asserted then, the following sequence occurs. :

1. Watchdog Timer (WDT) generates time-out signal.
2. PMU invokes reset signals and initializes internal IPs, and XnRSTOUT are asserted and reset counter is activated.
3. Reset counter is expired then; internal reset signals and XnRSTOUT are released.

Timing diagram for watchdog reset is shown in Figure 2.4-8.

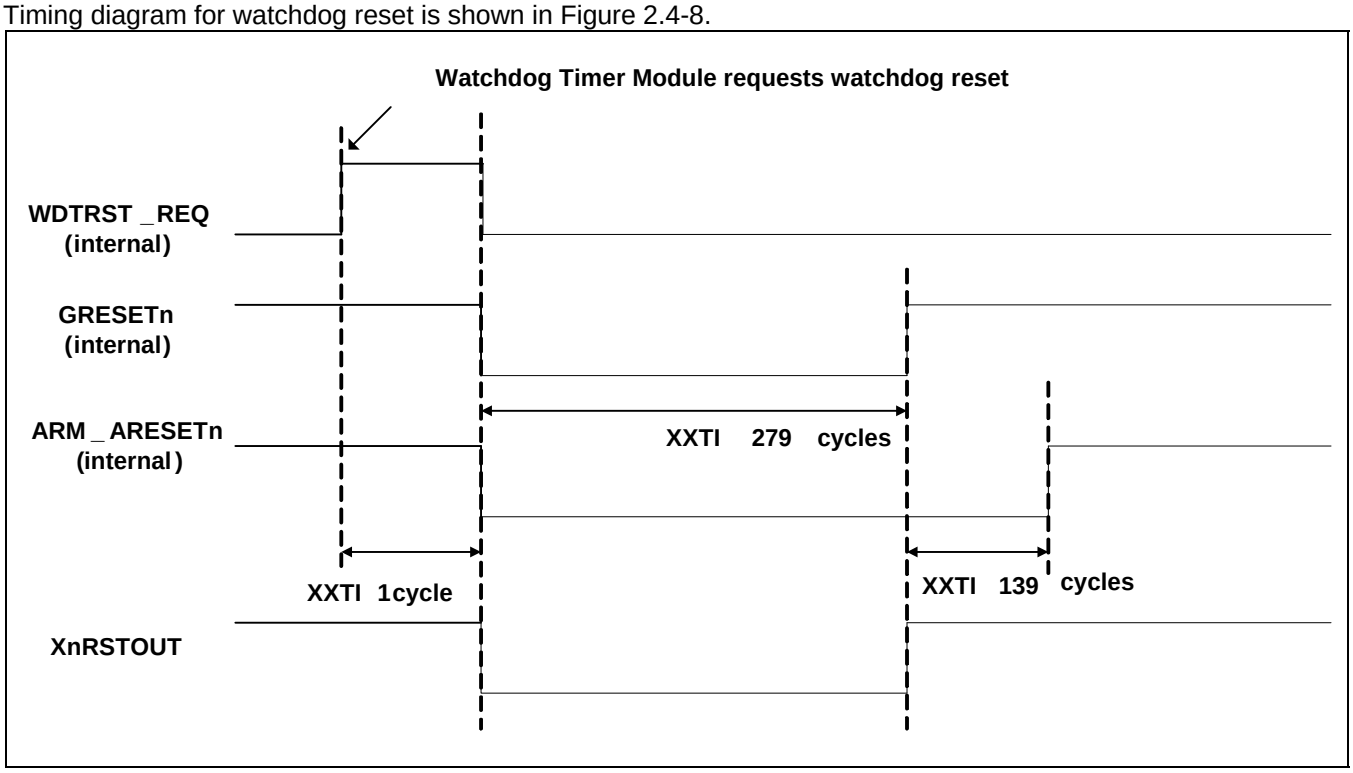


Figure 2.4-8 Watchdog Reset Timing Diagram

7.5 WAKEUP RESET

Wakeup reset is asserted when a module that has normal F/Fs is powered down, and the module is powered up again by wakeup events. Note that if the module has only retention F/Fs, wakeup reset is not asserted. But in sleep mode, wakeup reset is generated to all modules that were powered off regardless of normal F/F or retention F/F.

Therefore, wakeup reset is asserted in NORMAL, DEEP-IDLE, DEEP-STOP, and SLEEP mode.

In NORMAL mode, if a sub-domain is powered down, and the sub-domain is powered up again, wakeup reset is asserted to the sub-domain.
In DEEP-IDLE and DEEP-STOP mode, wakeup reset is asserted to Cortex-A8, since Cortex-A8 is powered up again when wakeup event occurs in these power modes. Wakeup reset is also asserted to a top domain if it was power-off (i.e., SFR_PWR_CFG[30] = 1'b0 for DEEP-IDLE mode / SFR_STOP_CFG[8] = 1'b0), and a sub-domain that becomes power on after exiting from DEEP-IDLE and DEEP-STOP mode.

Finally, wakeup reset is asserted when the system is waked up from sleep mode by wakeup event.

The exact timing diagram for wakeup reset is shown in Figure 2.4-11 for DEEP-IDLE mode (top domain off), Figure 2.4-14 for DEEP-STOP mode (top domain off), and Figure 2.4-15 for SLEEP mode.

In these cases, XnRSTOUT is asserted when SFR_OTHER [1:0] field is set to 2'b10.

Register initialization due to various resets, is shown in Table 2.4-15.

Table 2.4-15 Register Initialization Due to Various Resets

Block	Registers	Hardware (XnRESET)	Watchdog	Wakeup from SLEEP, DEEP IDLE(Top Off), DEEP STOP(Top Off)	Software
PMU	RST_STAT	X	X	X	O
PMU	INFORM4-7[1]	X	X	X	X
PMU	PWR_CFG, EINT_WAKEUP_MASK, NORMAL_CFG, STOP_CFG, SLEEP_CFG, OSC_FREQ, OSC_STABLE, PWR_STABLE, INTERNAL_PWR_STABLE, CLAMP_STABLE, OTHERS, WAKEUP_STAT, BLK_PWR_STAT, INFORM0-3, DCGIDX_MAP0-2, DCGPERF_MAP0-1, DVCIDX_MAP, FREQ_CPU, FREQ_DPM, APLL_CON_L1-8, CLKDIV_IEM_L1-8, IEM_HPMCLK_DIV, IEM_CONTROL	X	X	O	O
RTC_AHBIF	RTCCON, TICCNT, RTCALM, ALMSEC, ALMMIN, ALMHOUR, ALMDAY, ALMMON, ALMYEAR, RTCRST	X	X	O	O

GPIO	GPH0_CON, GPH1_CON, GPH2_CON, GPH3_CON, GPH0_PAD_OUT, GPH1_PAD_OUT, GPH2_PAD_OUT, GPH3_PAD_OUT, GPH0_PAD_EN, GPH1_PAD_EN, GPH2_PAD_EN, GPH3_PAD_EN, GPH0_PAD_CPU, GPH1_PAD_CPU, GPH2_PAD_CPU, GPH3_PAD_CPU, GPH0_PAD_CPD, GPH1_PAD_CPD, GPH2_PAD_CPD, GPH3_PAD_CPD, GPH0_PAD_DRV, GPH1_PAD_DRV, GPH2_PAD_DRV, GPH3_PAD_DRV	O	X	O	O
Others	-	O	O	O	O

1. Unlike INFORM0~3 registers, INFORM4-7 registers keep their values as long as alive power is supplied.

8 WAKEUP TIMING DIAGRAM

In this section, wakeup timing diagrams from various power-down modes are described.

8.1 IDLE MODE WAKEUP

Figure 2.4-9 shows wakeup timing from IDLE mode.

In IDLE mode, ARMCLK is disabled for reducing dynamic power in ARM CPU. When wakeup event is asserted, ARMCLK_OFF signal is released and then ARMCLK is supplied for ARM CPU within XXTI 6 cycles. After ARMCLK is supplied, CPU operation begins normally.

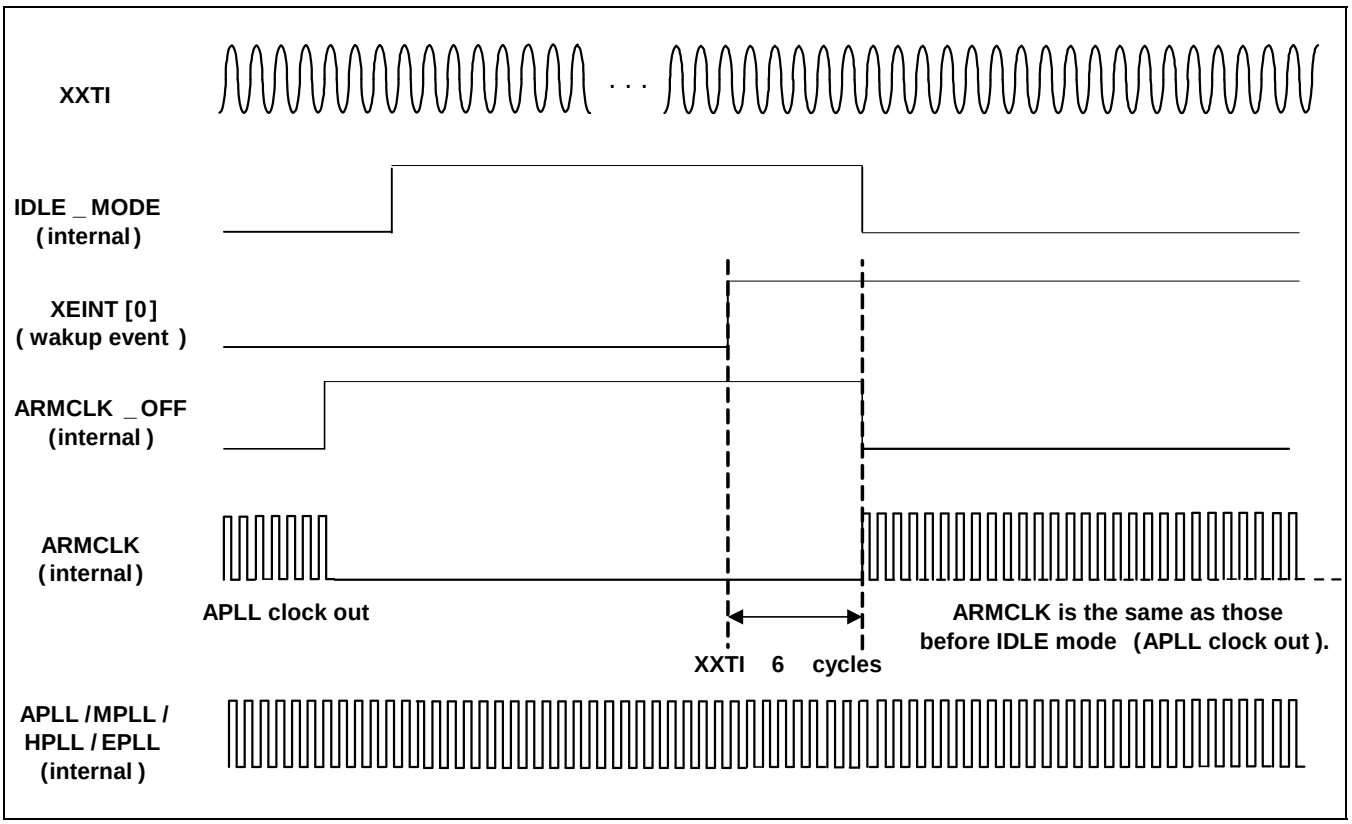


Figure 2.4-9 IDLE Mode Wakeup Timing

8.2 DEEP-IDLE MODE WAKEUP

Wakeup timing in DEEP-IDLE mode (Top domain on) is shown in Figure 2.4-10, and wakeup timing in DEEP-IDLE mode (Top domain off) is shown in Figure 2.4-11.

Note that in Figure 2.4-10, ARMCLK is the same as those before DEEP-IDLE mode, but in Figure 2.4-11 ARMCLK is the same as XXTI after wakeup and APLL should be enabled to supply high-frequency for ARM CPU. ARM_ARESETn is released within XXTI 1424 cycles after wakeup source is asserted.

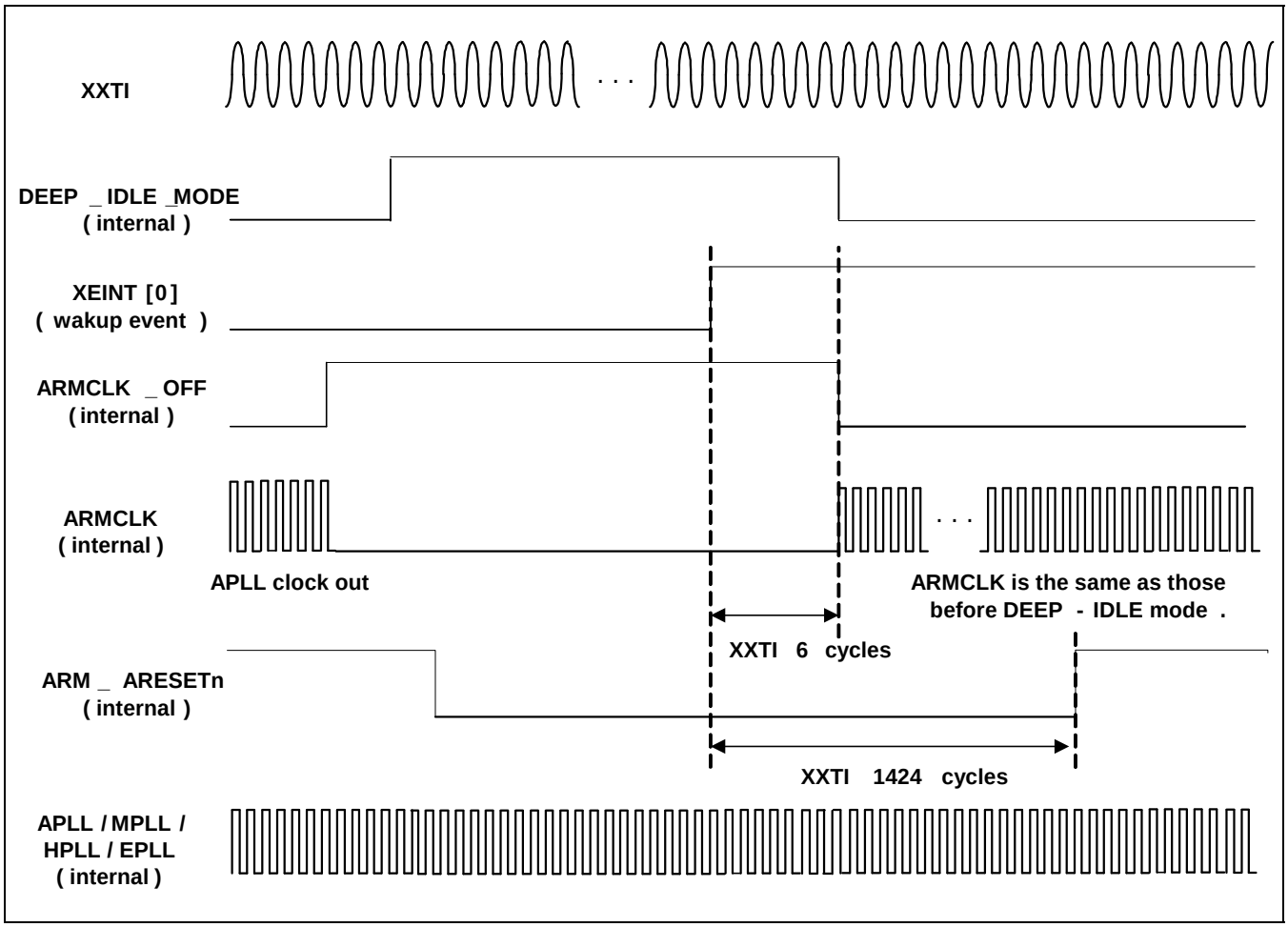


Figure 2.4-10 DEEP-IDLE Mode (Top Domain on) Wakeup Timing

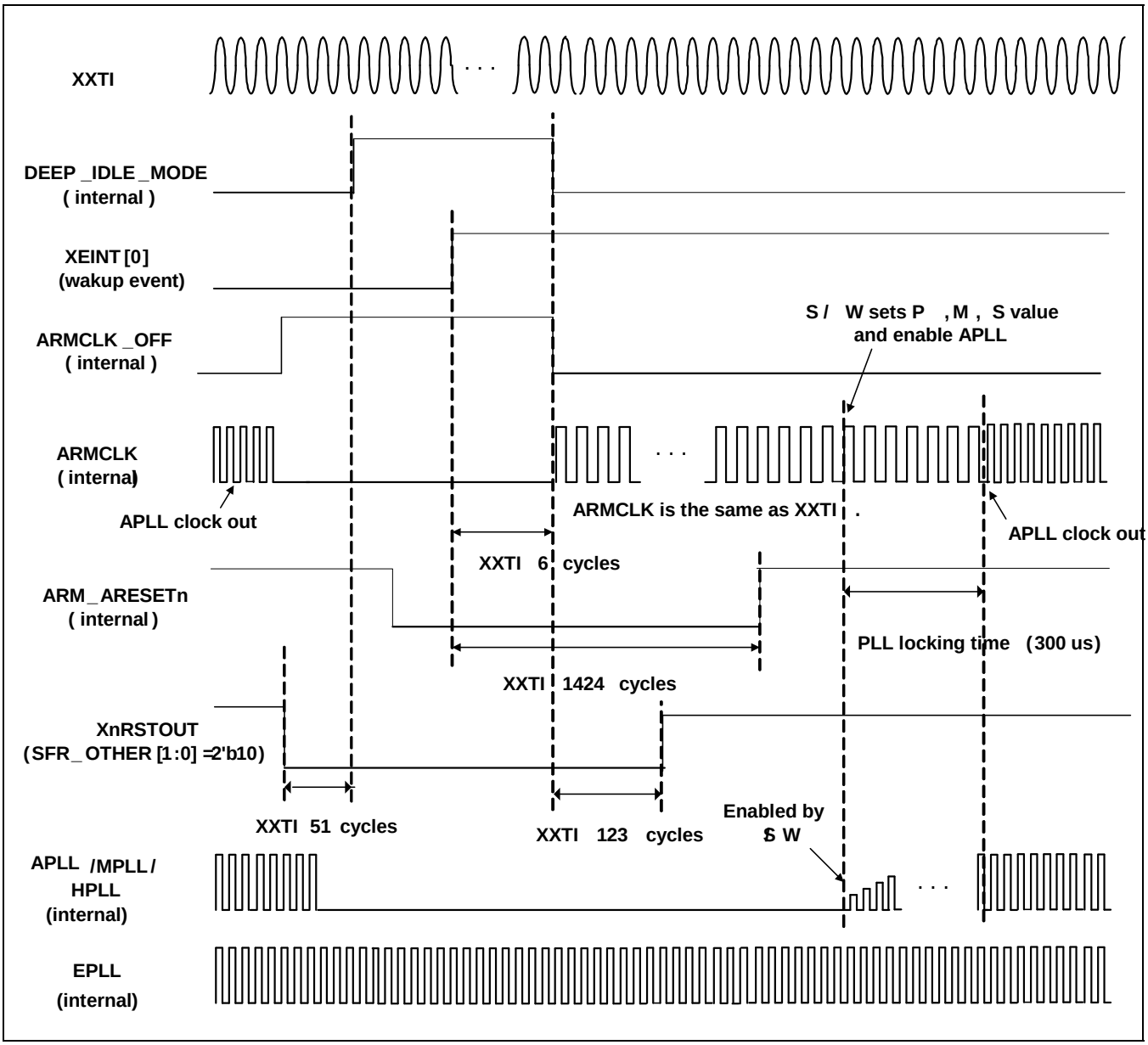


Figure 2.4-11 DEEP-IDLE Mode (Top Domain Off) Wakeup Timing

8.3 STOP MODE WAKEUP

Figure 2.4-12 shows wakeup timing from STOP mode.

When wakeup source is asserted, ARMCLK is supplied for ARM CPU within some interval (refer to comment on Figure 2.4-12). ARMCLK is the same as those before STOP mode. After ARMCLK is supplied, CPU operation begins normally.

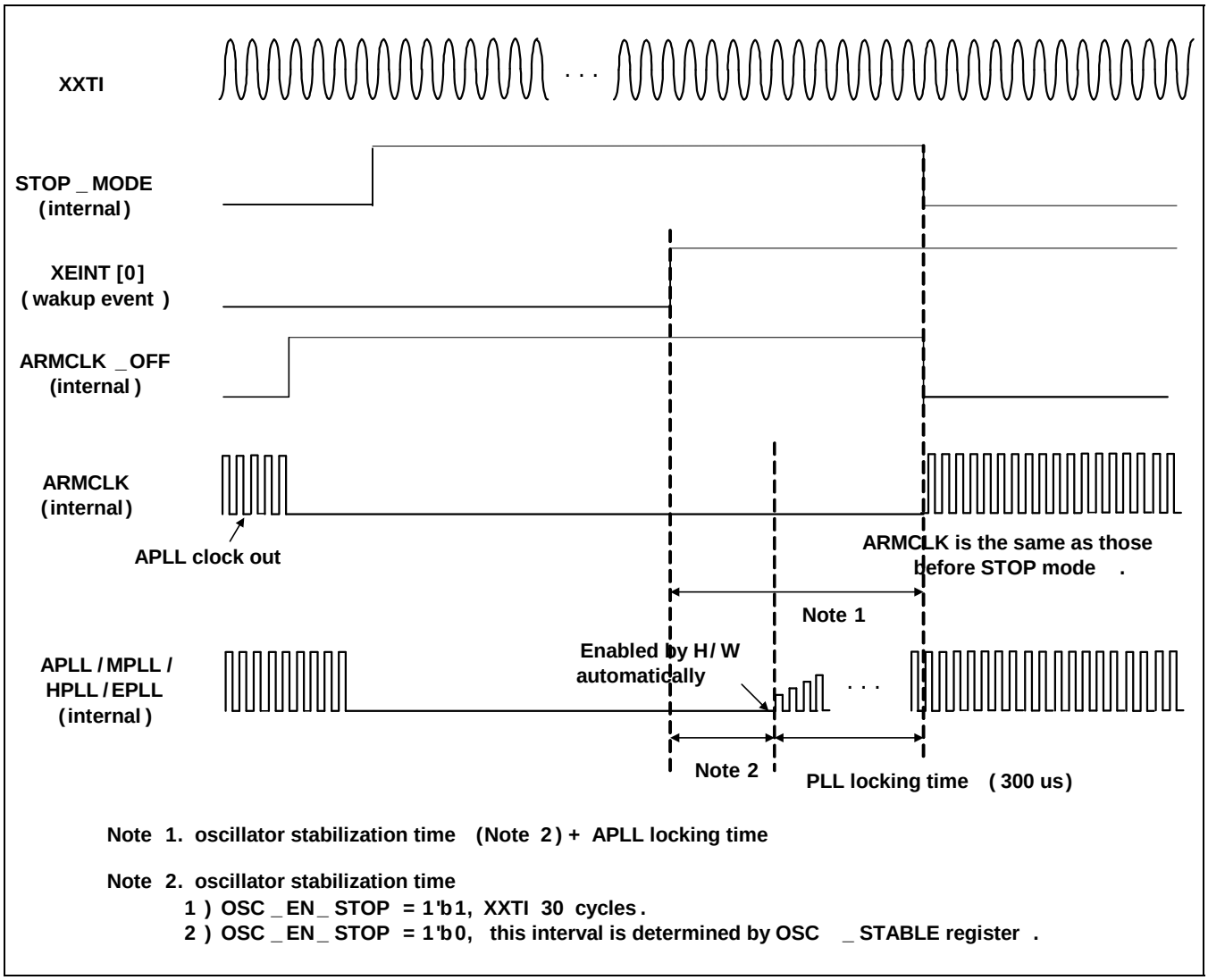


Figure 2.4-12 STOP Mode Wakeup Timing

8.4 DEEP-STOP MODE WAKEUP

Wakeup timing in DEEP-STOP mode (Top domain on) is shown in Figure 2.4-13.

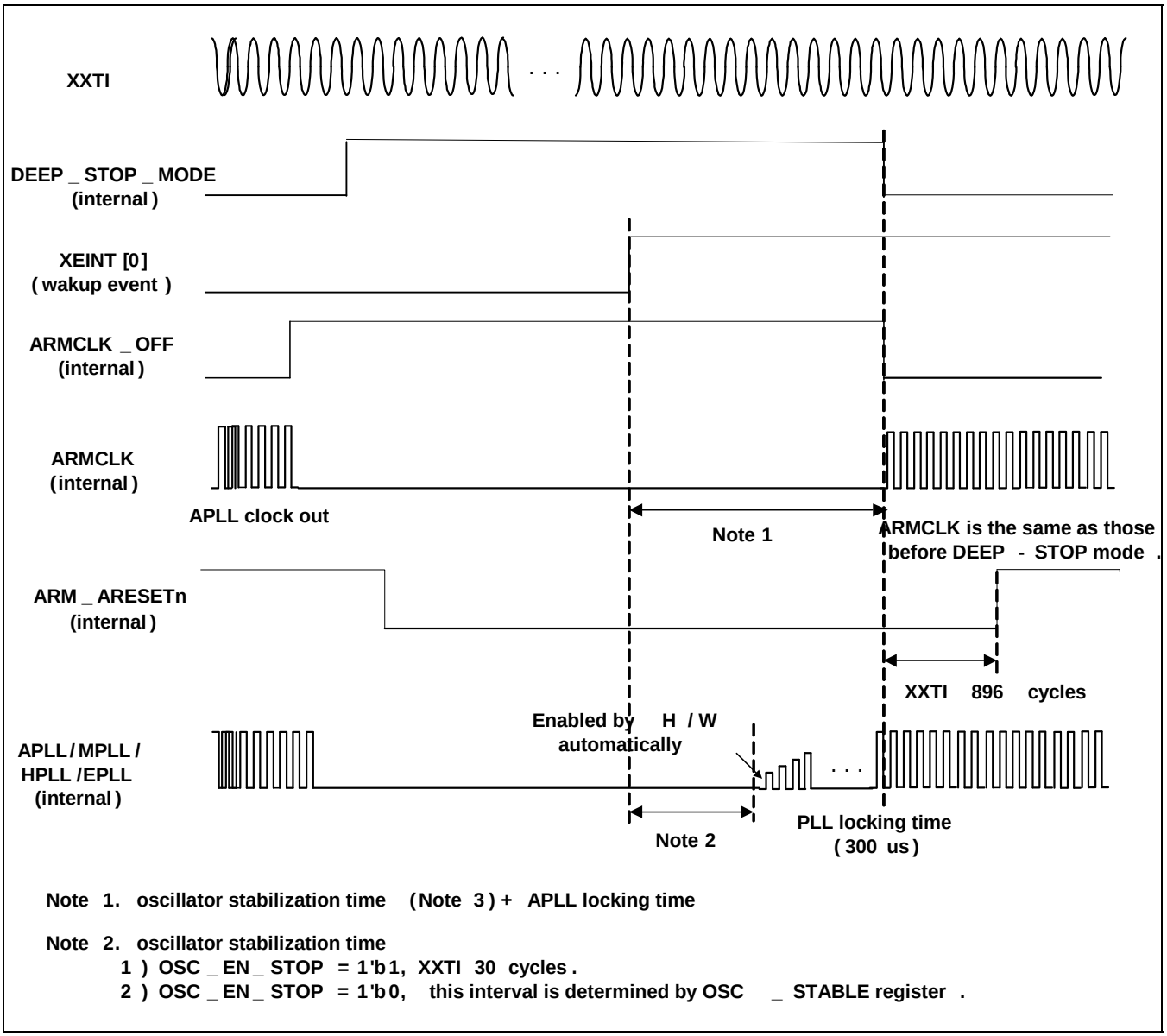


Figure 2.4-13 DEEP-STOP Mode (TOP Domain on) Wakeup Timing

When wakeup event is asserted, ARMCLK is supplied for ARM CPU within some interval (refer to comment on Figure 2.4-13) and ARM_ARESETn is released within XXTI 896 cycles after ARMCLK is supplied in case of OSC_EN_STOP bit = 1'b1.

Wakeup timing in DEEP-STOP mode (Top domain off) is shown in Figure 2.4-14.

Note that in Figure 2.4-13, ARMCLK is the same as those before DEEP-STOP mode, but in Figure 2.4-14 ARMCLK is the same as XXTI after wakeup and APLL should be enabled to supply high-frequency for ARM CPU. After ARM_ARESETn is released, CPU operation begins normally.

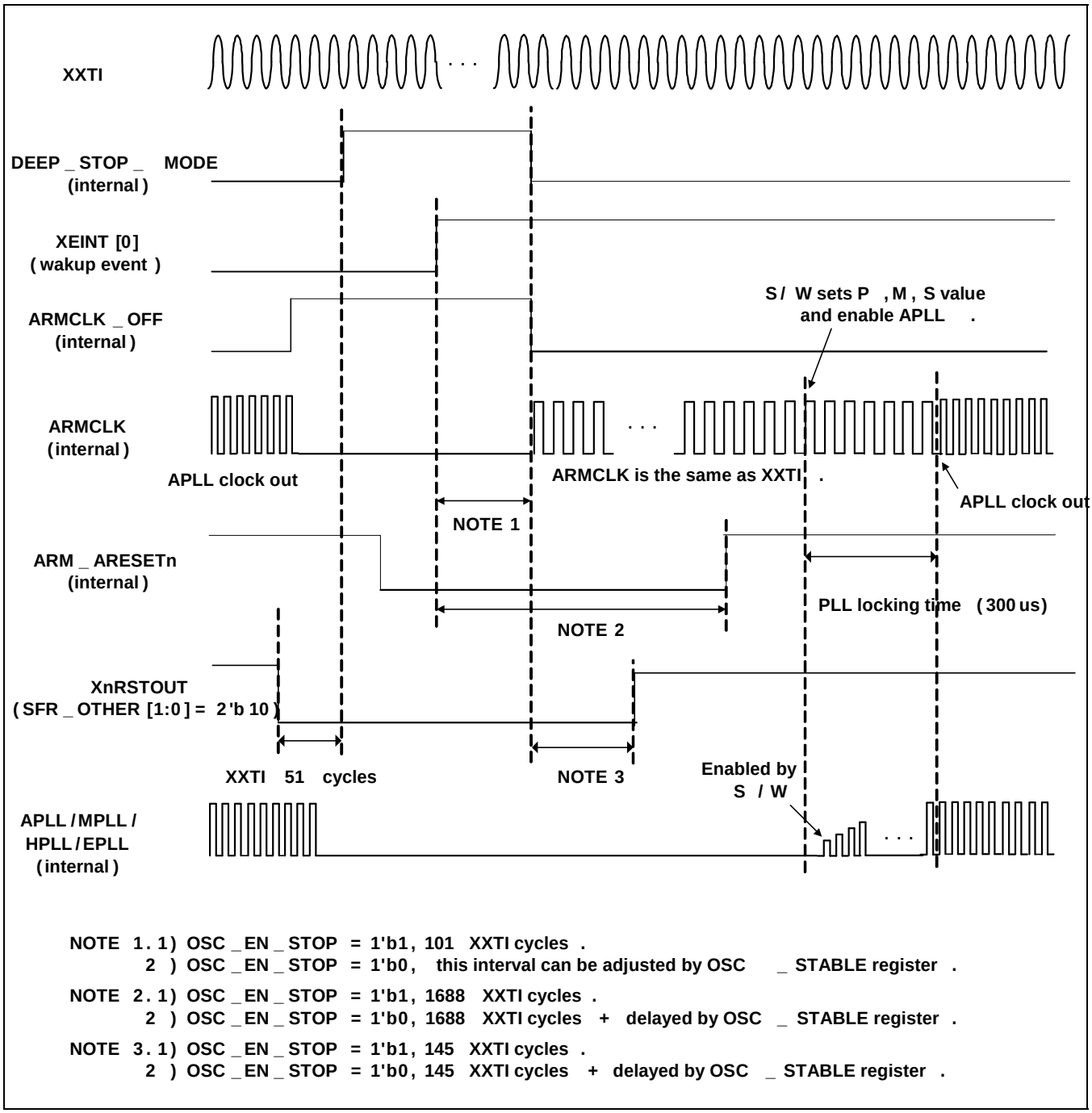


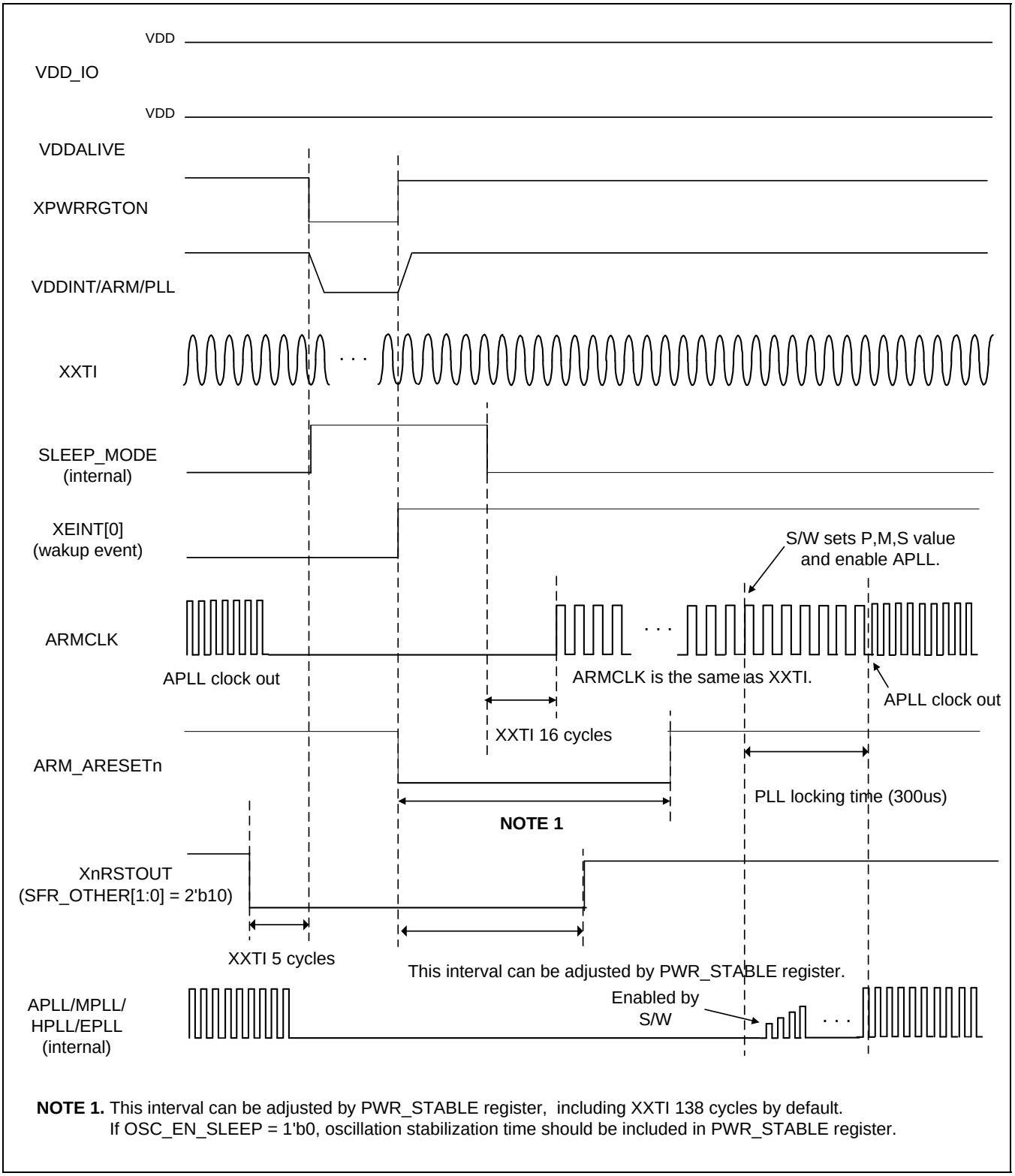
Figure 2.4-14 DEEP-STOP Mode (TOP Domain Off) Wakeup Timing

8.5 SLEEP MODE WAKEUP

Figure 2.4-5 shows wakeup timing from SLEEP mode.

In SLEEP mode, XPWRRGTON becomes low and then external power (VDDINT/ARM/PLL) is off to minimize leakage power of S5PC100. When SLEEP_WAKEUP is asserted, XPWRRGTON becomes high and then external power is on. ARMCLK is supplied for ARM and ARM_ARESETn is released within some interval (refer to

comment in Figure 2.4-5). After ARM_ARESETn is released, CPU operation begins normally.



NOTE 1. This interval can be adjusted by PWR_STABLE register, including XXTI 138 cycles by default.
If OSC_EN_SLEEP = 1'b0, oscillation stabilization time should be included in PWR_STABLE register.

Figure 2.4-15 SLEEP Mode Wakeup Timing

Oscillation setting time from sleep mode return is shown in Figure 2.4-16. Main clock can be supplied by XXTI input (XOM[0]=0) or XusbXTI input (XOM[0]=1), and oscillator (EXTCLK) or crystal (OSC) can be used for main clock.

Note that when crystal is used for main clock, oscillator stabilization time for XXTI or XusbXTI pad should be considered by using PWR_STABLE register in case that oscillation pad is disabled in SLEEP mode.

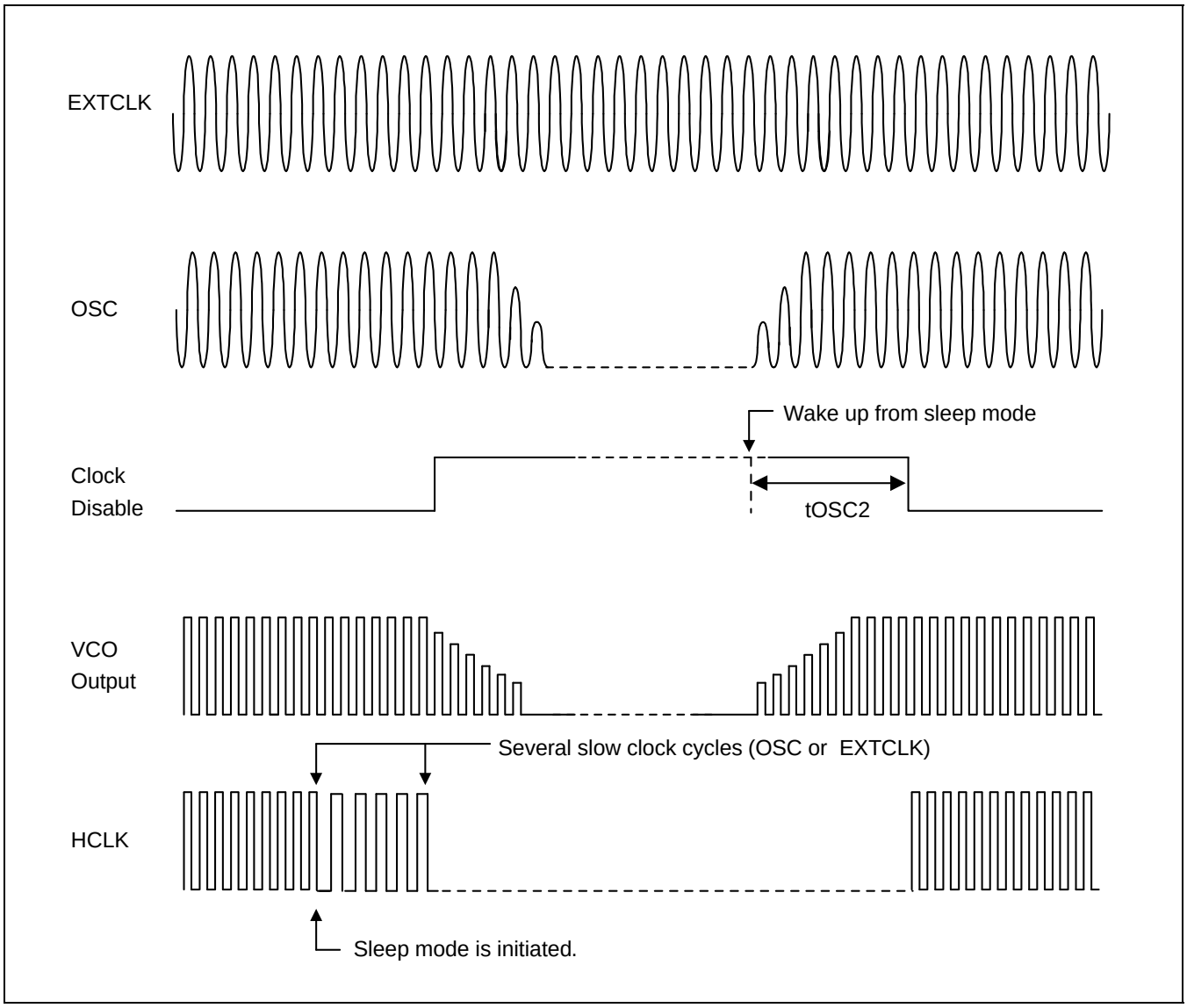


Figure 2.4-16 Sleep Mode Return Oscillation Setting Timing

9 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type(1)
PWRRGTON	Output	External Power Regulator (or PMIC) On	XPWRRGTON	Dedicated
nRSTOUT	Output	Internal Reset Out	XnRSTOUT(2)	Dedicated
nBATF	Input	Battery Fault	XnBATF	Dedicated
nRESET	Input	External Hardware Reset	XnRESET	Dedicated
nWRESET	Input	External Warm Reset	XnWRESET	Dedicated
XTI	Input	Oscillator input for Main clock	XXTI	Dedicated
27M_XTI	Input	Oscillator input for 27M clock	X27mXTI	Dedicated
USB_XTI	Input	Oscillator input for USB clock or Main clock	XusbXTI	Dedicated

1. Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals

2. XnRSTOUT is asserted when hardware reset, software reset, watchdog reset, and wakeup reset in DEEP-IDLE (top off)/ DEEP-STOP (top off)/SLEEP are asserted. The exact timing is referred to corresponding figure in Section 7 RESET CONTROL.

3. XnWRESET is not used, and therefore XnWRESET should be connected to VDD.

10 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
PWR_CFG	0xE010_8000	R/W	Configure power manager	0x2900_0001
EINT_WAKEUP_MASK	0xE010_8004	R/W	Configure EINT(external interrupt) mask	0x0000_0000
Reserved	0xE010_8008	-	DO NOT CHANGE	0x0000_0000
Reserved	0xE010_800C	-	Reserved	0x0000_0000
NORMAL_CFG	0xE010_8010	R/W	Configure power manager at NORMAL mode	0x00FF_FFFF
STOP_CFG	0xE010_8014	R/W	Configure power manager at STOP mode	0x2012_0100
SLEEP_CFG	0xE010_8018	R/W	Configure power manager at SLEEP mode	0x0000_0000
STOP_MEM_CFG	0xE010_801C	R/W	Configure memory power at STOP mode	0x0000_07FF
Reserved	0xE010_8020-0xE010_80FC	-	Reserved	-
OSC_FREQ	0xE010_8100	R/W	Oscillator frequency scale counter	0x0000_000F
OSC_STABLE	0xE010_8104	R/W	Oscillator pad stable counter	0x0000_0001
PWR_STABLE	0xE010_8108	R/W	Power stable counter	0x0000_0001
Reserved	0xE010_810C	-	DO NOT CHANGE	0x0000_0001
INTERNAL_PWR_STABLE	0xE010_8110	R/W	MTC stable counter	0xFFFF_FFFF
CLAMP_STABLE	0xE010_8114	R/W	Cortex-A8 CLAMP stable counter	0x03FF_03FF
Reserved	0xE010_8118	R/W	DO NOT CHANGE	0x03FF_03FF

Register	Address	R/W	Description	Reset Value
Reserved	0xE010_811C-0xE010_81FC	-	Reserved	-
OTHERS	0xE010_8200	R/W	Miscellaneous control register	0x0000_001E
Reserved	0xE010_8204-0xE010_82FC	-	Reserved	-
RST_STAT	0xE010_8300	R	Reset status register	0x0000_0001
WAKEUP_STAT	0xE010_8304	R/W	Wakeup status registers	0x0000_0000
BLK_PWR_STAT	0xE010_8308	R	Block power status register	0x0000_003F
Reserved	0xE010_830C-0xE010_83FC	-	Reserved	-
INFORM0	0xE010_8400	R/W	Information register0	0x0000_0000
INFORM1	0xE010_8404	R/W	Information register1	0x0000_0000
INFORM2	0xE010_8408	R/W	Information register2	0x0000_0000
INFORM3	0xE010_840C	R/W	Information register3	0x0000_0000
INFORM4	0xE010_8410	R/W	Information register4	-
INFORM5	0xE010_8414	R/W	Information register5	-
INFORM6	0xE010_8418	R/W	Information register6	-
INFORM7	0xE010_841C	R/W	Information register7	-
Reserved	0xE010_8420-0xE010_84FC	-	Reserved	-
DCGIDX_MAP0	0xE010_8500	R/W	DCG Index Map 0	0xFFFF_FFFF
DCGIDX_MAP1	0xE010_8504	R/W	DCG Index Map 1	0xFFFF_FFFF
DCGIDX_MAP2	0xE010_8508	R/W	DCG Index Map 2	0xFFFF_FFFF
DCGPERF_MAP0	0xE010_850C	R/W	DCG Performance Map 0	0xFFFF_FFFF
DCGPERF_MAP1	0xE010_8510	R/W	DCG Performance Map 1	0xFFFF_FFFF
DVCIDX_MAP	0xE010_8514	R/W	DVC Index Map	0x00FF_FFFF
FREQ_CPU	0xE010_8518	R/W	Maximum Frequency of CPU	0x0000_0000
FREQ_DPM	0xE010_851C	R/W	Frequency of DPM Accumulators	0x0000_0000
DVSEMCLK_EN	0xE010_8520	R/W	DVS Emulation Clock Enable	0x0000_0000
Reserved	0xE010_8524 - 0xE010_85FC	-	Reserved	-
APLL_CON_L8	0xE010_8600	R/W	ARM PLL Control (Performance Level : 8)	0x0190_0302
APLL_CON_L7	0xE010_8604	R/W	ARM PLL Control (Performance Level : 7)	0x0190_0302
APLL_CON_L6	0xE010_8608	R/W	ARM PLL Control (Performance Level : 6)	0x0190_0302
APLL_CON_L5	0xE010_860C	R/W	ARM PLL Control (Performance Level : 5)	0x0190_0302
APLL_CON_L4	0xE010_8610	R/W	ARM PLL Control (Performance Level : 4)	0x0190_0302
APLL_CON_L3	0xE010_8614	R/W	ARM PLL Control (Performance Level : 3)	0x0190_0302
APLL_CON_L2	0xE010_8618	R/W	ARM PLL Control (Performance Level: 2)	0x0190_0302

Register	Address	R/W	Description	Reset Value
APLL_CON_L1	0xE010_861C	R/W	ARM PLL Control (Performance Level : 1)	0x0190_0302
IEM_CONTROL	0xE010_8620	R/W	IEM Control Register	0x0000_0000
Reserved	0xE010_8624- 0xE010_86FC	-	Reserved	-
CLKDIV_IEM_L8	0xE010_8700	R/W	Clock Divider for IEM(Performance Level : 8)	0x0000_0002
CLKDIV_IEM_L7	0xE010_8704	R/W	Clock Divider for IEM(Performance Level : 7)	0x0000_0002
CLKDIV_IEM_L6	0xE010_8708	R/W	Clock Divider for IEM(Performance Level : 6)	0x0000_0002
CLKDIV_IEM_L5	0xE010_870C	R/W	Clock Divider for IEM(Performance Level : 5)	0x0000_0002
CLKDIV_IEM_L4	0xE010_8710	R/W	Clock Divider for IEM(Performance Level : 4)	0x0000_0002
CLKDIV_IEM_L3	0xE010_8714	R/W	Clock Divider for IEM(Performance Level : 3)	0x0000_0002
CLKDIV_IEM_L2	0xE010_8718	R/W	Clock Divider for IEM(Performance Level : 2)	0x0000_0002
CLKDIV_IEM_L1	0xE010_871C	R/W	Clock Divider for IEM(Performance Level : 1)	0x0000_0002
Reserved	0xE010_8720	-	DO NOT CHANGE	0x0000_0002
IEM_HPMCLK_DIV	0xE010_8724	R	IEM HPMCLK Divider Value	0x0000_0000

10.1 POWER MODE CONTROL REGISTER

10.1.1 Configure Power Manager (PWR_CFG, R/W, Address = 0xE010_8000)

PWR_CFG	Bit	Description	Reset Value
CFG_DEEP_IDLE	[31]	Configures DEEP setting for Cortex-A8 Core. (applied only to IDLE mode) 0 = No DEEP (Cortex-A8 core power on) 1= DEEP (Cortex-A8 core power off)	0
TOP_LOGIC_ON_DIDLE(1)	[30]	Configures Top domain state in DEEP-IDLE mode. 0 = SRPG (State-Retention Power Gating) 1 = ON	0
ARM_L2CACHE _RET_DIDLE	[29]	Configures ARM L2Cache Retention in DEEP-IDLE mode. 0 = OFF 1 = RETENTION	1
TOP_MEMORY_ON _DIDLE(1)	[28]	Configures TOP Memory On in DEEP-IDLE mode. 0 = LP mode(RET or OFF), 1 = ON	0
TOP_MEMORY_RET _DIDLE(1)	[27]	Configures TOP Memory Retention in DEEP-IDLE mode. 0 = OFF, 1 = RET	1
Reserved	[26:25]	DO NOT CHANGE	00
OSCUSB_EN	[24]	Controls USB X-tal oscillator pad (XusbXTI) in NORMAL mode. The state is kept in power down modes. **Note : Do not disable if XusbXTI is used main clock source.** 0 = Disable 1 = Enable	1
Reserved	[23:19]	DO NOT CHANGE	0x0
ST_WAKEUP_MASK	[18]	System Timer (ST) wake-up mask 0 = ST is used as a wakeup source 1 = ST cannot be used as a wakeup source	0
I2S_WAKEUP_MASK	[17]	I2S wake-up mask 0 = I2S is used as a wakeup source 1 = I2S cannot be used as a wakeup source	0
MMC2_WAKEUP_MASK	[16]	MMC2 wake-up mask 0 = MMC2 is used as a wakeup source 1 = MMC2 cannot be used as a wakeup source	0
MMC1_WAKEUP_MASK	[15]	MMC1 wake-up mask 0 = MMC1 is used as a wakeup source 1 = MMC1 cannot be used as a wakeup source	0
MMC0_WAKEUP_MASK	[14]	MMC0 wake-up mask 0 = MMC0 is used as a wakeup source 1 = MMC0 cannot be used as a wakeup source	0

PWR_CFG	Bit	Description	Reset Value
HSI_WAKEUP_MASK	[13]	HSI_RX wake-up mask 0 = HSI_RX is used as a wakeup source 1 = HSI_RX cannot be used as a wakeup source	0
TS_WAKEUP_MASK	[12]	Touch screen Pen down (TS_PENDN) wake-up mask 0 = TS_PENDN is used as a wakeup source 1 = TS_PENDN cannot be used as a wakeup source	0
TICK_WAKEUP_MASK	[11]	RTC TICK wake-up mask 0 = RTC TICK is used as a wakeup source 1 = RTC TICK cannot be used as a wakeup source	0
ALARM_WAKEUP _MASK	[10]	RTC alarm wake-up mask 0=RTC alarm is used as a wakeup source 1= RTC alarm cannot be used as a wakeup source	0
MODEM_WAKEUP _MASK	[9]	MODEM wake-up mask 0 = MODEM is used as a wakeup source 1 = MODEM cannot be used as a wakeup source	0
KEY_WAKEUP_MASK	[8]	Key pad wake-up mask 0 = Key Pad is used as a wakeup source 1 = Key Pad cannot be used as a wakeup source	0
BATF_WAKEUP _MASK	[7]	BATF wake-up source 0 = Disables 1 = Use as a wakeup source and only use if CFG_BATFLT field has 2'b01.	0
CFG_STANDBYWFI	[6:5]	Configures CORTEX-A8 STADNBYWFI Determines the actions taken if the STANDBYWFI signal is activated by the CORTEX-A8 00 = Enter IDLE mode 01 = Enters DEEP IDLE mode 10 = Enters STOP/DEEP STOP mode 11 = Enters SLEEP mode	0x0
CFG_BATFLT	[4:3]	Configure BATF Determines what actions are taken BATF condition occurs. BATF condition occurs when the nBAT_FLT input signal is activated (goes LOW) CFG_BATFLT 00 = ignore 01 = generate interrupt 10 = ignore 11 = BATFLT causes entry into E-SLEEP E-SLEEP is a special case of the SLEEP mode which is entered when the BATFLT is activated E-SLEEP configuration support requires: CFG_BATFLT = = 2'b11	0x0

PWR_CFG	Bit	Description	Reset Value
		If the configuration supports E-SLEEP, the system will entry E-SLEEP mode when nBATF is asserted (nBATF == 0).	
CFG_BATF_WKUP	[2]	Configure wakeup source from E-SLEEP mode 0 = No wakeup source 1 = use all SLEEP wakeup sources except nBATF Wakeup from E-SLEEP requires nBATF condition to have been cleared (nBATF == 1). CFG_BATF_WKUP is a don't care if the system is not in E-SLEEP.	0
Reserved	[1]	DO NOT CHANGE	0
OSC27_EN	[0]	Control 27MHz X-tal oscillator pad (XXTI27) in NORMAL mode. The state is kept in power down modes. 0 = Disable 1 = Enable	1

1. These three bits are configured in DEEP-IDLE mode (CFG_DEEP_IDLE = 1'b1) as follows:

	TOP_LOGIC_ON_DIDLE = 1'b1	
	TOP_MEMORY_RET_DIDLE = 1'b0	TOP_MEMORY_RET_DIDLE = 1'b1
TOP_MEMORY_ON_DIDLE = 1'b0	TOP memory power down	TOP memory retention
TOP_MEMORY_ON_DIDLE = 1'b1		TOP memory on

	TOP_LOGIC_ON_DIDLE = 1'b0	
	TOP_MEMORY_RET_DIDLE = 1'b0	TOP_MEMORY_RET_DIDLE = 1'b1
TOP_MEMORY_ON_DIDLE = 1'b0	TOP memory power down	TOP memory retention
TOP_MEMORY_ON_DIDLE = 1'b1		

NOTE: If CFG_DEEP_IDLE = 1'b0, TOP memory is always on regardless of TOP_LOGIC_ON_DIDLE, TOP_MEMORY_ON_DIDLE, and TOP_MEMORY_RET_DIDLE.

10.1.2 Configure EINT[31:0] Wakeup Mask (EINT_WAKEUP_MASK, R/W, Address = 0xE010_8004)

EINT_WAKEUP_MASK	Bit	Description	Reset Value
EINT_WAKEUP_MASK	[31:0]	External interrupt wake-up source masking selection in the corresponding bit for EINT[31:0] 0 – Unmask 1 - Mask	0x0

10.1.3 Configure Power Manager at Normal Mode (NORMAL_CFG, R/W, Address = 0xE010_8010)

NORMAL_CFG	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
Reserved	[23:8]	DO NOT CHANGE	0xFFFF
Reserved	[7:6]	DO NOT CHANGE	0x3
AUDIO	[5]	Audio Sub-system : I2S0, SRAM (128 KB) 0 = LP mode(OFF), 1 = active mode(ON)	1
TV	[4]	TV Sub-system : HDMI, MIXER, VP, TVEncoder 0 = LP mode(OFF), 1 = active mode(ON)	1
LCD	[3]	LCD Sub-system : CAMIF, DISPCON, G2D, JPEG, MIPI_CSI, MIPI_DSI, ROTATOR 0 = LP mode(OFF), 1 = Active mode(ON)	1
G3D	[2]	0 = LP mode(OFF), 1 = Active mode(ON)	1
MFC	[1]	0 = LP mode(OFF), 1 = Active mode(ON)	1
Reserved	[0]	DO NOT CHANGE	1

Note. When NORMAL_CFG[3](LCD) is set to 0, LPMP3_MODE_SEL(bit[0]) in LPMP3_MODE_SEL in Chapter 2.3. Clock Controller is recommended to be set to 1 in order to minimize the internal current.

10.1.4 Configure Power Manager at STOP mode (STOP_CFG, R/W, Address = 0xE010_8014)

STOP_CFG	Bit	Description	Reset Value
TOP_MEMORY_ON (1)	[31]	If TOP_LOGIC_ON = 0, you should set this bit to 0. 0 = RET or OFF, 1 = ON	0
Reserved	[30]	DO NOT CHANGE	0
ARM_L2CACHE_RET(2)	[29]	0 = No Retention(On or OFF), 1 = Retention	1
Reserved	[28:21]	DO NOT CHANGE	0x0
Reserved	[20]	DO NOT CHANGE	1
Reserved	[19:18]	DO NOT CHANGE	0
ARM_LOGIC_ON(3)	[17]	0 = LP mode(OFF), 1 = Active mode(ON)	1
Reserved	[16:9]	DO NOT CHANGE	0x0
TOP_LOGIC_ON	[8]	This bit has no effect if ARM_LOGIC_ON = 1. 0 = LP mode(Retention), 1 = Active mode(ON)	1
Reserved	[7:1]	DO NOT CHANGE	0x0
OSC_EN_STOP (4)	[0]	Control all X-tal oscillator pads, i.e., main X-tal oscillator(XXTI), 27M X-tal oscillator(XXTI27), and USB X-tal oscillator(XusbXTI) in STOP mode 0 = Disable 1 = Enable	0

1. Top Memory On bit must be applied with Top domain On configuration.
   Top memory includes all SRAM inside IP module of Top domain.
   If TOP_MEMORY_ON = 0, then you should determine power mode (retention or off) of each TOP memory by setting STOP_MEM_CFG register before entering STOP/DEEP-STOP mode.)
2. ARM_L2CACHE_RET field is related ARM_LOGIC_ON as follows.)

	ARM_LOGIC_ON = 1'b0	ARM_LOGIC_ON = 1'b1
ARM_L2CACHE_RET = 1'b0	ARM L2CACHE OFF	ARM L2CACHE ON
ARM_L2CACHE_RET = 1'b1	ARM L2CACHE RETENTION	N.A

3. If ARM_LOGIC_ON[17] = 1'b1, it is called STOP mode, and if ARM_LOGIC_ON[17] = 1'b0, it is called DEEP-STOP mode. The available configurations of other bits of STOP_CFG register in each mode are shown below:
   Note: [n]-bit offset of corresponding field in STOP_CFG register, TL-Top domain, TM-Top memory, AL-ARM Logic, AM-ARM L2Cache, O-available configuration, X-unavailable configuration, ()-the constraint or reason why that configuration is not supported

   (1) STOP_mode : ARM_LOGIC_ON[17] = 1'b1

TL[8]	TM[31]	AL[17]	AM[29]	Result
0	0	0	0	X (TL, AL should be 1)
0	0	0	1	X (TL, AL should be 1)
0	0	1	0	X (TL, AL should be 1)
0	0	1	1	X (TL, AL should be 1)
0	1	0	0	X (TL, AL should be 1)
0	1	0	1	X (TL, AL should be 1)
0	1	1	0	X (TL, AL should be 1)
0	1	1	1	X (TL, AL should be 1)
1	0	0	0	X (TL, AL should be 1)
1	0	0	1	X (TL, AL should be 1)
1	0	1	0	O
1	0	1	1	X (AL=1 and AM=1)
1	1	0	0	X (TL,AL should be 1)
1	1	0	1	X (TL,AL should be 1)
1	1	1	0	O
1	1	1	1	X (AL=1 and AM=1)

(2) DEEP-STOP_mode : ARM_LOGIC_ON[17] = 1'b0

TL[8]	TM[31]	AL[17]	AM[29]	Result
0	0	0	0	O
0	0	0	1	O
0	0	1	0	X (AL=1, AL should be 0)
0	0	1	1	X (AL=1, AL should be 0)
0	1	0	0	X (TL=0 and TM=1)
0	1	0	1	X (TL=0 and TM=1)
0	1	1	0	X (AL=1, AL should be 0)
0	1	1	1	X (AL=1, AL should be 0)
1	0	0	0	O
1	0	0	1	O
1	0	1	0	X (AL=1, AL should be 0)
1	0	1	1	X (AL=1, AL should be 0)
1	1	0	0	O
1	1	0	1	O
1	1	1	0	X (AL=1, AL should be 0)
1	1	1	1	X (AL=1, AL should be 0)

4. 27M Oscillator (XXTI27), and USB Oscillator (XusbXTI) Enable is also controlled by this bit. The relationships among OSCUSB_EN(PWR_CFG[24]), OSC27_EN(PWR_CFG[0]), OSC_EN STOP_CFG[0]), and OSC_EN(SLEEP_CFG[0], are summarized as follows.)

(1) XXTI: This pad is not affected by OSCUSB_EN(PWR_CFG[24]) and OSC27_EN(PWR_CFG[0]).

	OSC_EN_STOP (STOP_CFG[0]) = 1'b0		OSC_EN_STOP (STOP_CFG[0]) = 1'b1	
	OSC_EN_SLEEP (SLEEP_CFG[0]) = 1'b0	OSC_EN_SLEEP (SLEEP_CFG[0]) = 1'b1	OSC_EN_SLEEP (SLEEP_CFG[0]) = 1'b0	OSC_EN_SLEEP (SLEEP_CFG[0]) = 1'b1
NORMAL mode	Enable			
IDLE and DEEP-IDLE mode	Enable			
STOP and DEEP-STOP mode	Disable		Enable	
SLEEP mode	Disable	Enable	Disable	Enable

(2) XXTI27: This pad is not affected by OSCUSB_EN(PWR_CFG[24]).

	OSC27_EN (PWR_CFG[0]) = 1'b0				OSC27_EN (PWR_CFG[0]) = 1'b1			
	OSC_EN_STOP = 1'b0		OSC_EN_STOP = 1'b1		OSC_EN_STOP = 1'b0		OSC_EN_STOP = 1'b1	
	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1
NORMAL mode	Disable				Enable			
IDLE and DEEP-IDLE mode	Disable				Enable			
STOP and DEEP-STOP mode	Disable				Disable		Enable	
SLEEP mode	Disable				Disable	Enable	Disable	Enable

(3) XusbXTI: This pad is not affected by OSC27_EN(PWR_CFG[0]).

	OSCUSB_EN (PWR_CFG[24]) = 1'b0				OSCUSB_EN (PWR_CFG[24]) = 1'b1			
	OSC_EN_STOP = 1'b0		OSC_EN_STOP = 1'b1		OSC_EN_STOP = 1'b0		OSC_EN_STOP = 1'b1	
	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0	OSC_EN _SLEEP = 1'b1	OSC_EN _SLEEP = 1'b0
NORMAL mode	Disable				Enable			
IDLE and DEEP-IDLE mode	Disable				Enable			
STOP and DEEP-STOP mode	Disable				Disable		Enable	
SLEEP mode	Disable				Disable	Enable	Disable	Enable

10.1.5 Configure Power Manager at SLEEP mode (SLEEP_CFG, R/W, Address = 0xE010_8018)

SLEEP_CFG	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x0
OSC_EN_SLEEP	[0]	Control all X-tal oscillator pads, i.e., main X-tal oscillator (XXTI), 27M X-tal oscillator (XXTI27), and USB X-tal oscillator (XusbXTI) in SLEEP mode. 0 = Disables 1 = Enables	0x0

10.1.6 Configure Memory Power at STOP mode (STOP_MEM_CFG, R/W, Address = 0xE010_801C)

STOP_MEM_CFG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
CCAN	[10]	CCAN Block Memory control (0 = Off, 1 = Retention)	1
IRDA	[9]	IRDA Block Memory control (0 = Off, 1 = Retention)	1
MODEMIF	[8]	MODEM IF Block Memory control (0 = Off, 1 = Retention)	1
USBOTG	[7]	USB_OTG Block Memory control (0 = Off, 1 = Retention)	1
SDMMC	[6]	SDMMC Memory control (0 = Off, 1 = Retention)	1
CSSYS	[5]	CSSYS Memory control (0 = Off, 1 = Retention)	1
SECSS	[4]	SECSS control (0 = Off, 1 = Retention)	1
IRAMC2	[3]	IRAMC2 (32KB) memory control (0 = Off, 1 = Retention)	1
IRAMC1	[2]	IRAMC1 (32KB) memory control (0 = Off, 1 = Retention)	1
IRAMC0	[1]	IRAMC0 (32KB) memory control (0 = Off, 1 = Retention)	1
Reserved	[0]	DO NOT CHANGE	1

Each bit is available if TOP_MEMORY_ON field of STOP_CFG is 0. Therefore, if TOP_MEMORY_ON filed of STOP_CFG is 1, then each bit has no effect.
Memory corresponding to each bit, means SRAM inside corresponding IP module of Top domain, and used by that module.

10.2 SYSTEM STABILIZATION COUNTER

10.2.1 Oscillator Frequency Scale Counter (OSC_FREQ, R/W, Address = 0xE010_8100)

OSC_FREQ	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0x0
OSC_FREQ_VALUE	[3:0]	Oscillator frequency scale counter (OSC_FREQ_VALUE x *oscillator_period* > 200ns)	0xF

OSC_FREQ register is applied to control timing in power up/down sequence in the following cases :
1) power off of sub power-domain (MFC, G3D, LCD, TV) in NORMAL mode.
2) top domain on option in DEEP-IDLE mode
3) top domain off option in DEEP-STOP mode

10.2.2 Oscillation Pad Stable Counter (OSC_STABLE, R/W, Address = 0xE010_8104)

OSC_STABLE	Bit	Description	Reset Value
Reserved	[31:20]	Reserved	0x0
OSC_CNT_VALUE	[19:4]	Mapping to counter value 19 to 4 when STABLE COUNTER Type is 1 (OSC_CNT_VALUE[3:0] should stay at reset values)	
	[3:0]	Oscillation pad stable counter value (Exponential Scale) Value Cycles Value Cycles 0x0 2^4 0x1 2^9 0x2 2^{10} 0x3 2^{11} 0x4 2^{12} 0x5 2^{13} 0x6 2^{14} 0x7 2^{15} 0x8 2^{16} Others 2^4	0x1

If main oscillation pad (XXTI) is enabled from disabled state, it takes some amount of time for the pad to generate stable clock. OSC_STABLE register is applied to wait for this oscillation stabilization time when OSC_EN[0] of STOP_CFG is 1'b0 in STOP and DEEP-STOP mode.
Oscillation stabilization time is calculated as follows :

Oscillation stabilization time = OSC_Stabilization_Counter_Value x Ocillator_period (XTI)

where OSC_Stabilizaton_Counter_Value can be set by two ways as below :
you can select one by using STABLE_COUNTER TYPE[23] in OTHERS register.
1) Exponential Scale : The cycle value mapped by OSC_CNT_VALUE[3:0] is used.
e.g., if OSC_CNT_VALUE[3:0] = 0x1, then OSC_Stabilization_Counter_Value = 2^9 (512).
2) Set by SFR : OSC_Stabilization_Coutner_Value = OSC_CNT_VALUE[19:4]. And OSC_CNT_VALUE[3:0] should be kept as the reset value.

10.2.3 Power Stable Counter (PWR_STABLE, R/W, Address = 0xE010_8108)

PWR_STABLE	Bit	Description	Reset Value
Reserved	[31:20]	Reserved	0x0
PWR_CNT_VALUE	[19:4]	Mapping to counter value 19 to 4 when STABLE COUNTER Type is 1 (PWR_CNT_VALUE[3:0] should stay at reset values)	
	[3:0]	Power stable counter value (Exponential Scale) Value Cycles Value Cycles 0x0 2^4 0x1 2^{12} 0x2 2^{13} 0x3 2^{14} 0x4 2^{15} 0x5 2^{16} 0x6 2^{17} 0x7 2^{18} 0x8 2^{19} Others 2^4	0x1

If external power is supplied from off state, it takes some amount of time for the external regulator or PMIC to supply stable power. PWR_STABLE register is applied to wait for this power stabilization time when S5PC100 is exit from SLEEP mode.
Note that In SLEEP mode, oscillator stabilization time is included in PWR_STABLE register. Therefore, when OSC_EN (SLEEP_CFG[0]) is set to 1'b0, you should set PWR_STABLE register to proper value including oscillator stabilization time.
Power stabilization time is calculated as follows:

Power stabilization time = PWR_Stabilization_Counter_Value x Ocillator_period (XTI)

where PWR_Stabilizaton_Counter_Value can be set by two ways as below :
you can select one by using STABLE_COUNTER TYPE[23] in OTHERS register.
1) Exponential Scale : The cycle value mapped by PWR_CNT_VALUE[3:0] is used.
e.g., if PWR_CNT_VALUE[3:0] = 0x1, then PWR_Stabilization_Counter_Value = 2^{12} (4096).

2) Set by SFR : PWR_Stabilization_Coutner_Value = PWR_CNT_VALUE[19:4]. And PWR_CNT_VALUE[3:0] should be kept as the reset value.

10.2.4 MTC Stable Counter (INTERNAL_PWR_STABLE, R/W, Address = 0xE010_8110)

INTERNAL_PWR_STABLE	Bit	Description	Reset Value
Reserved	[31:24]	DO NOT CHANGE	0xFF
AUDIO	[23:20]	Internal power stabilization counter for domain AUDIO	0xF
TV	[19:16]	Internal power stabilization counter for domain TV	0xF
LCD	[15:12]	Internal power stabilization counter for domain LCD	0xF
G3D	[11:8]	Internal power stabilization counter for domain G3D	0xF
MFC	[7:4]	Internal power stabilization counter for domain MFC	0xF
TOP	[3:0]	Internal power stabilization counter for domain TOP	0xF

* : INTERNAL_PWR_STABLE cannot have 0 value.
INTERNAL_PWR_STABLE represents the number of external oscillator (or clock) cycles. If a sub-domain returns from Low power mode to normal operation mode, the internal power stabilization time is required. This period must be larger than 200nsec and it is estimated using INTERNAL_PWR_STABLE value and OSC_FREQ registers.

10.2.5 CorTex-A8 CLAMP Stable Counter (CLAMP_STABLE, R/W, Address = 0xE010_8114)

CLAMP_STABLE	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
Reserved	[27:26]	Reserved	00
CLAMP_OFF_VALUE	[25:16]	Clamp off counter value. This value is used in power-up sequence. This field is set to as follows: (CLAMP_OFF_VALUE x *oscillator_period (XTI)* > 33us)	0x3FF
Reserved	[15:12]	Reserved	0x0
Reserved	[11:10]	Reserved	00
CLAMP_ON_VALUE	[9:0]	Clamp on counter value. This value is used in power-down sequence. **Note:** You should set this field to 0x1 before entering any power down modes.	0x3FF

CLAMP_STABLE is used for control timing between CLAMP_ON/OFF and POWER_ON/OFF in Core, L2Cache, ETM, and NEON of Cortex-A8 in DEEP-IDLE, DEEP-STOP, and SLEEP mode. Before entering above power down modes, you should set the recommended values in both CLAMP_OFF_VALUE and CLAMP_ON_VALUE.

10.3 MISCELLANEOUS CONTROL REGISTER

10.3.1 Miscellaneous Control Register (OTHERS, R/W, Address = 0xE010_8200)

OTHERS	Bit	Description	Reset Value
IO_RET_RELEASE	[31]	IO_RET_RELEASE. IO_RET is retention control signal to I/O pad except SDMMC I/O. SDMMC I/O should be controlled separately (Refer to SDMMC_IO_RET_RELEASE[22] field in OTHERS register) Set this bit to 1 to release IO_RET. After IO_RET is released, this bit will be cleared to 0. **Usage:** After wakeup from DEEP-IDLE (Top domain off), DEEP-STOP (Top domain off) and SLEEP mode, you should first set GPIO configurations as the same ones before those power down mode since GPIO configurations are not kept in those power down mode. Then, you should set this bit to 1 so that I/O pads is used normally. 0 = auto clear by h/w, 1 = de-assert RET_EN1 and RET_LPA	0
EPLL_RET_RELEASE	[30]	EPLL_RET_RELEASE. EPLL_RET is retention control signal to EPLL. Set this bit to 1 to release EPLL_RET. After EPLL_RET is released, this bit will be cleared to 0. **Usage:** After wakeup from DEEP-IDLE / DEEP-STOP (Top domain off), you should first set PLL configurations as the same ones before those power down mode since PLL configurations are not	0

OTHERS	Bit	Description	Reset Value
		kept in those power down mode. Then, you should set this bit to 1 so that EPLL can be used normally. 0 = auto clear by h/w, 1 = de-assert EPLL_RET	
Reserved	[29]	DO NOT CHANGE	0
MIPI_DPHY_EN(1)	[28]	MIPI_DPHY Enable selection. This bit must be set to 1 at the system initialization step before MIPI_DPHY operation begins. **Caution: If MIPI_DPHY is not used in your system, do not touch this bit.** 0 = Disables 1 = Enables	0
Reserved	[27]	DO NOT CHANGE	0
DAC_EN(1)	[26]	Digital-to-Analog Converter (DAC) IP Enable selection. This bit must be set to 1 at the system initialization step before DAC operation begins. **Caution: If DAC is not used in your system, do not touch this bit.** 0 = Disables 1 = Enables	0
L2RSTDISABLE	[25]	Monitors the L2 hardware reset disable bit 0 = L2 valid RAM contents are reset by hardware 1 = L2 valid RAM contents are not reset by hardware	0
PMU_INT_DISABLE	[24]	Interrupt disabling by PMU. Auto-clear by H/W **Usage:** Before entry to power down mode, you should set this bit to 1 before issuing WFI command. Interrupt to Cortex-A8 after issuing WFI command should be masked since interrupt after issuing WFI command malfunctions due to conflict with power down process performed by H/W. 0 = Enables, 1 = Disables	0
STABLE_COUNTER TYPE	[23]	Indicate OSC_STABLE, PWR_STABLE counter type 0 = Exponential Scale, 1 = Set by SFR	0
SDMMC_IO_RET_RELEASE	[22]	SDMMC_IO_RET_RELEASE. SDMMC_IO_RET is retention control signal to SDMMC I/O pad. If you want to release SDMMC_IO_RET, set this bit to 1. After SDMMC_IO_RET is released, this bit will be cleared to 0. **Usage:** After wakeup from DEEP-IDLE (Top domain off), DEEP-STOP (Top domain off) and SLEEP mode, you should first set GPIO configurations as the same ones before those power down mode since GPIO configurations are not kept in those power down mode. And then, you should set this bit	0

OTHERS	Bit	Description	Reset Value
		to 1 so that SDMMC I/O pads can be used normally. 0 = auto clear by h/w, 1 = de-assert SDMMC_IO_RET	
Reserved	[21:17]	DO NOT CHANGE	0x0000
USB_SIG_MASK(1)	[16]	The role of this bit is to bypass or block the signals transferred from USB OTG PHY to internal logic. In order to start USB transaction, USB_SIG_MASK is set to 1 and then USB_PHY initialization sequence begins. **Caution: If USB_PHY is not used in your system, this bit is set to 0.** 0 = Mask 1 = Unmask	0
Reserved	[15:14]	DO NOT CHANGE	00
CLEAR_DBGACK	[13]	Clear DBGACK signal if this field has 1. Cortex-A8 asserts DBGACK signal to indicate the system has entered Debug state. If DBGACK is asserted, this state is stored in PMU until software clears it using this field.	0
CLEAR_BATF_INT(2)	[12]	If this bit is set it clears interrupt caused by battery fault when this bit is set.	0
Reserved	[11:5]	DO NOT CHANGE	0x0
Reserved	[4]	DO NOT CHANGE	1
Reserved	[3:2]	DO NOT CHANGE	0x3
XnRSTOUT_TYPE	[1:0]	These bits determines the type of XnRSTOUT asserted in DEEP-IDLE (top block off) / DEEP-STOP (top block off) / SLEEP mode. **Note: XnRSTOUT is asserted when TOP domain is power-off.** 00 = low 01 = high 10 = Auto nRESET 11 = reserved	0x2

1. These bits are necessary to prevent leakage current when the corresponding IP are not used in system.
2. Self-reset bit after SFR setting

10.4 STATUS REGISTER

10.4.1 Reset Status Register (RST_STAT, R, Address = 0xE010_8300)

RST_STAT	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x0
DIDLE_WAKEUP	[7]	ARM Reset from Deep IDLE	0
DSTOP_WAKEUP	[6]	ARM Reset from Deep STOP	0
SWRESET	[5]	Software reset by SWRESET	0
ESLEEP_WAKEUP	[4]	Reset by E-SLEEP mode wake-up	0
SLEEP_WAKEUP	[3]	Reset by SLEEP mode wake-up	0
nWDTRESET	[2]	Watch dog timer reset by WDTRST	0
Reserved	[1]	Reserved	0
nRESET	[0]	External reset by XnRESET	1

10.4.2 Wake-up Status Register (WAKEUP_STAT, R/W, Address = 0xE010_8304)

WAKEUP_STAT	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0x0
ST	[13]	Wake-up by System Timer. Write 1 to clear.	0
I2S	[12]	Wake-up by I2S in Audio Block. Write 1 to clear.	0
MMC2	[11]	Wake-up by MMC2. Write 1 to clear.	0
MMC1	[10]	Wake-up by MMC1. Write 1 to clear.	0
MMC0	[9]	Wake-up by MMC0. Write 1 to clear.	0
HSI	[8]	Wake-up by HSI. Write 1 to clear.	0
Reserved	[7]	DO NOT CHANGE	0
BATFLT	[6]	Wake-up by battery fault. Write 1 to clear.	0
MODEM	[5]	Wake-up by MODEM. Write 1 to clear.	0
KEY	[4]	Wake-up by Key PAD. Write 1 to clear.	0
TS	[3]	Wake-up by touch screen. Write 1 to clear.	0
RTC_TICK	[2]	Wake-up by tick interrupt. Write 1 to clear.	0
RTC_ALARM	[1]	Wake-up by RTC alarm. Write 1 to clear.	0
EINT	[0]	Wake-up by external interrupts. Write 1 to clear.	0

10.4.3 Block Power Status Register (BLK_PWR_STAT, R, Address = 0xE010_8308)

BLK_PWR_STAT	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0x0
AUDIO	[5]	Block AUDIO power ready	1
TV	[4]	Block TV power ready	1

BLK_PWR_STAT	Bit	Description	Reset Value
LCD	[3]	Block LCD power ready	1
G3D	[2]	Block G3D power ready	1
MFC	[1]	Block MFC power ready	1
Reserved	[0]	Reserved	1

10.5 INFORMATION REGISTER

- Information Register 0 (INFORM0, R/W, Address = 0xE010_8400)
- Information Register 1 (INFORM1, R/W, Address = 0xE010_8404)
- Information Register 2 (INFORM2, R/W, Address = 0xE010_8408)
- Information Register 3 (INFORM3, R/W, Address = 0xE010_840C)

Field	Bit	Description	Reset Value
INFORM	[31:0]	INFORM0~3 registers are cleared by asserting XnRESET pin.	0x0000_0000

- Information Register 4 (INFORM4, R/W, Address = 0xE010_8410)
- Information Register 5 (INFORM5, R/W, Address = 0xE010_8414)
- Information Register 6 (INFORM6, R/W, Address = 0xE010_8418)
- Information Register 7 (INFORM7, R/W, Address = 0xE010_841C)

Field	Bit	Description	Reset Value
INFORM	[31:0]	INFORM4~7 registers are not cleared when the XnRESET pin is asserted	-

NOTE:
INFORM4~7 registers are not cleared when the XnRESET pin is asserted.

10.5.1 DCGIDX_MAP0 Register (DCGIDX_MAP0, R/W, Address = 0xE010_8500)

This register is related to IECCFGDCGIDXMAP[31:0] of IEM_IEC input port.

DCGIDX_MAP0	Bit	Description	Reset Value
dcgidx_map0	[31:0]	IEC configuration for DCG index map [31:0]	0xFFFF_FFFF

dcgidx_map0[31:0] is mapped to IECCFGDCGIDXMAP[31:0] of IEM_IEC input port.

10.5.2 DCGIDX_MAP1 Register (DCGIDX_MAP1, R/W, Address = 0xE010_8504)

This register is related to IECCFGDCGIDXMAP[63:32] of IEM_IEC input port.

DCGIDX_MAP1	Bit	Description	Reset Value
dcgidx_map1	[31:0]	IEC configuration for DCG index map [63:32]	0xFFFF_FFFF

dcgidx_map1[31:0] is mapped to IECCFGDCGIDXMAP[63:32] of IEM_IEC input port.

10.5.3 DCGIDX_MAP2 Register (DCGIDX_MAP2, R/W, Address = 0xE010_8508)

This register is related to IECCFGDCGIDXMAP[95:64] of IEM_IEC input port.

DCGIDX_MAP2	Bit	Description	Reset Value
dcgidx_map2	[31:0]	IEC configuration for DCG index map [95:64]	0xFFFF_FFFF

dcgidx_map2[31:0] is mapped to IECCFGDCGIDXMAP[95:64] of IEM_IEC input port.

10.5.4 DCGPERF_MAP0 Register (DCGPERF_MAP0, R/W, Address = 0xE010_850C)

This register is related to IECCFGDCGPERFMAP[31:0] of IEM_IEC input port.

DCGPERF_MAP0	Bit	Description	Reset Value
dcgperf_map0	[31:0]	IEC configuration for DCG performance map [31:0]	0xFFFF_FFFF

dcgperf_map0[31:0] is mapped to IECCFGDCGPERFMAP[31:0] of IEM_IEC input port.

10.5.5 DCGPERF_MAP1 Register (DCGPERF_MAP1, R/W, Address = 0xE010_8510)

This register is related to IECCFGDCGPERFMAP[63:32] of IEM_IEC input port.

DCGPERF_MAP1	Bit	Description	Reset Value
dcgperf_map1	[31:0]	IEC configuration for DCG performance map [63:32]	0xFFFF_FFFF

dcgperf_map1[31:0] is mapped to IECCFGDCGPERFMAP[63:32] of IEM_IEC input port

10.5.6 DVCIDX_MAP Register (DVCIDX_MAP, R/W, Address = 0xE010_8514)

The register correspond to IECCFGDVCIDXMAP[23:0] of IEM_IEC input port.

DVCIDX_MAP	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
dvcidx_map	[23:0]	IEC configuration for DVC index map [23:0]	0xFF_FFFF

dvcidx_map1[23:0] is mapped to IECCFGDVCIDXMAP[23:0] of IEM_IEC input port

10.5.7 FREQ_CPU Register (FREQ_CPU, R/W, Address = 0xE010_8518)

The register related to IECCFGFREQCPU[23:0] of IEM_IEC input port.

FREQ_CPU	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
freq_cpu	[23:0]	Maximum frequency of CPU in kHz	0x00_0000

freq_cpu[23:0] is the maximum processor clock frequency in kHz, and gives the clock frequency of the processor in kHz. Table 2.4-16 lists three examples.

Table 2.4-16 IECCFGFREQCPU Examples

IECCFGFREQCPU[23:0]	Verilog expression	Processor frequency
0x004E20	24'd020_000	20000kHz = 20MHz
0x03A980	24'd240_000	240000kHz = 240MHz
0x0003E8	24'd001_000	1000kHz = 1MHz

10.5.8 FREQ_DPM Register (FREQ_DPM, R/W, Address = 0xE010_851C)

The register related to IECCFGFREQDPM[23:0] of IEM_IEC input port.

FREQ_DPM	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
freq_dpm	[23:0]	Frequency of DPM Accumulators	0x00_0000

freq_dpm[23:0] is the DPM frequency in kHz, and gives the rate that the DPM is accumulating in kHz. Table 2.4-17 lists three examples.

Table 2.4-17 IECCFGFREQDPM Examples

IECCFGFREQDPM[23:0]	Verilog expression	Processor frequency
0x004E20	24'd020_000	20000kHz = 20MHz
0x002710	24'd010_000	10000kHz = 10MHz
0x0003E8	24'd001_000	1000kHz = 1MHz

10.5.9 DVSEMCLK_EN Register (DVSEMCLK_EN, R/W, Address = 0xE010_8520)

The register related to IECDVSEMCLKEN of IEM_IEC input port.

DVSEMCLK_EN	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0x0000_000
Reserved	[3:2]	Reserved	00
dpmclk_en	[1]	DPM Clock Enable	0
dvsemclk_en	[0]	DVS Emulation Clock Enable	0

dvsemclken means that DPM Clock is enabled for advancing the PWM frame time slots when in DVS emulation mode. This signal must be pulsed at a frequency of 1MHz.

10.5.10 APLL_CON_L8 Register (APLL_CON_L8, R/W, Address = 0xE010_8600)

The register configures P/M/S values for ARM PLL at IEM performance level of 8.

APLL_CON_L8	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
Reserved	[27:26]	Reserved	00
apll_mval_l8	[25:16]	M[9:0] value for ARM PLL at IEM performance level of 8	0x190
Reserved	[15:14]	Reserved	00
apll_pval_l8	[13:8]	P[5:0] value for ARM PLL at IEM performance level of 8t	0x03
Reserved	[7:4]	Reserved	0x0
Reserved	[3]	Reserved	0
apll_sval_l8	[2:0]	S[2:0] value for ARM PLL at IEM performance level of 8	0x2

Use following equation to calculate frequency of PLL output clock:

Fout (Hz) = (m * Fin) / (p * 2^S)
where m = (MDIV + 8), p = (PDIV + 2), s = SDIV

apll_mval_l8[9:0] is equal to MDIV[9:0] , apll_pval_l8 is equal to PDIV[5:0], and apll_sval_l8 is equal to SDIV[2:0].

10.5.11 APLL_CON_L1-7 Register

- ARM PLL Control (Performance Level -7) (APLL_CON_L7, R/W, Address = 0xE010_8604)
- ARM PLL Control (Performance Level -6) (APLL_CON_L6, R/W, Address = 0xE010_8608)
- ARM PLL Control (Performance Level -5) (APLL_CON_L5, R/W, Address = 0xE010_860C)
- ARM PLL Control (Performance Level -4) (APLL_CON_L4, R/W, Address = 0xE010_8610)
- ARM PLL Control (Performance Level -3) (APLL_CON_L3, R/W, Address = 0xE010_8614)
- ARM PLL Control (Performance Level -2) (APLL_CON_L2, R/W, Address = 0xE010_8618)
- ARM PLL Control (Performance Level -1) (APLL_CON_L1, R/W, Address = 0xE010_861C)

Each register of APLL_CON_L1-7 configures P/M/S values for ARM PLL at IEM performance level from 1 to 7. The description of each field is the same as that of APLL_CON_L8 register.

APLL_CON_L1-7	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
Reserved	[27:26]	Reserved	00
apll_mval_l1-7	[25:16]	M[9:0] value for ARM PLL at IEM performance level from 1 to 7	0x190
Reserved	[15:14]	Reserved	00
apll_pval_l1-7	[13:8]	P[5:0] value for ARM PLL at IEM performance level from 1 to 7	0x03
Reserved	[7:4]	Reserved	0x0
Reserved	[3]	Reserved	0
apll_sval_l1-7	[2:0]	S[2:0] value for ARM PLL at IEM performance level from 1 to 7	0x2

10.5.12 IEM_CONTROL Registerbn (IEM_CONTROL, R/W, Address = 0xE010_8620)

The register controls IEM global function.

IEM_CONTROL	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0x0
Reserved	[27:26]	Reserved	00
Reserved	[25:24]	DO NOT CHANGE	00
iem_prev_target	[23:16]	IEM previous target performance level setting. You can set one of the following values: 0x01, 0x03, 0x07, 0x0f, 0x1f, 0x3f, 0x7f and 0xff	0x00
Reserved	[15:14]	Reserved	00
Reserved	[13:12]	DO NOT CHANGE	00
Reserved	[11:9]	DO NOT CHANGE	000
Iem_max_perf_con	[8]	This register value is transferred as IEC_MAXPERF signal to IEM_IEC module. You should not set this register to 1'b1 if iem_enable field (below) = 1'b0. 0 = Set IEC_MAXPERF to 1'b0 1 = Set IEC_MAXPERF to 1'b1	0x0
Reserved	[7:4]	Reserved	0x0
Reserved	[3]	Reserved	0
Reserved	[2:1]	DO NOT CHANGE	00
iem_enable	[0]	IEM function enable bit 0 = Disables IEM function , 1 = Enables IEM function	0

10.5.13 CLKDIV_IEM_L8 Register (CLKDIV_IEM_L8, R/W, Address = 0xE010_8700)

The register configures clock divider values for ARM and HPM clocks at IEM performance level of 8.

CLKDIV_IEM_L8	Bit	Description	Reset Value
Reserved	[31:16]	DO NOT CHANGE	0x0000
Reserved	[15:13]	DO NOT CHANGE	000
pll_div_val_l8	[12]	This field is used to set dividing value in both DIV_{APLL} and DIV_{APLL_HPM}. DIV_{APLL} clock divider ratio, $DOUT_{APLL} = MOUT_{APLL}$ / RATIO (*RATIO = pll_div_val_l8 + 1*), and DIV_{APLL_HPM} clock divider ratio, $DOUT_{APLL\_HPM} = MOUT_{APLL}$ / RATIO (*RATIO = pll_div_val_l8 + 1*) at IEM performance level of 8	0
Reserved	[11]	DO NOT CHANGE	0
hpm_div_val_l8	[10:8]	This field is used only in closed-loop mode (by using HPM). DIV_{HPM} clock divider ratio, SCLK_HPM = $DOUT_{ARM\_HPM}$ / RATIO (*RATIO = hpm_div_val_l8 + 1*) at IEM performance level of 8	000

CLKDIV_IEM_L8	Bit	Description	Reset Value
Reserved	[7]	DO NOT CHANGE	0
arm_div_val_l8	[6:4]	This field is used to set dividing value in both DIV_{ARM} and DIV_{ARM_HPM}. DIV_{ARM} clock divider ratio, $DOUT_{ARM} = DOUT_{APLL}$ / RATIO (*RATIO = arm_div_val_l8 + 1*), and DIV_{ARM_HPM} clock divider ratio, $DOUT_{ARM\_HPM} = DOUT_{APLL\_HPM}$ / RATIO (*RATIO = arm_div_val_l8 + 1*), at IEM performance level of 8	000
Reserved	[3]	DO NOT CHANGE	0
bus_div_val_l8	[2:0]	DIV_{D0_BUS} clock divider ratio, HCLKD0 = $DOUT_{ARM}$ / RATIO (*RATIO = bus_div_val_l8 + 1*) at IEM performance level of 8	010

DIV_{APLL}, DIV_{APLL_HPM}, DIV_{ARM} , $DIV_{ARM\_HPM,}$ DIV_{HPM}, and DIV_{D0_BUS} clock divider are shown in Fig 2.3-3. (Refer to 2.3 CLOCK STRATEGY)

10.5.14 CLKDIV_IEM_L1-7 Register

- Clock Divider for IEM (Performance Level -7) (CLKDIV_IEM_L7, R/W, Address = 0xE010_8704)
- Clock Divider for IEM (Performance Level -6) (CLKDIV_IEM_L6, R/W, Address = 0xE010_8708)
- Clock Divider for IEM (Performance Level -5) (CLKDIV_IEM_L5, R/W, Address = 0xE010_870C)
- Clock Divider for IEM (Performance Level -4) (CLKDIV_IEM_L4, R/W, Address = 0xE010_8710)
- Clock Divider for IEM (Performance Level -3) (CLKDIV_IEM_L3, R/W, Address = 0xE010_8714)
- Clock Divider for IEM (Performance Level -2) (CLKDIV_IEM_L2, R/W, Address = 0xE010_8718)
- Clock Divider for IEM (Performance Level -1) (CLKDIV_IEM_L1, R/W, Address = 0xE010_871C)

Each register of CLKDIV_IEM_L1-7 configures clock divider values for ARM and HPM clocks at IEM performance level from 1 to 7. The description of each field is the same as that of CLKDIV_IEM_L8 register.

CLKDIV_IEM_L1-7	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0x0000
Reserved	[15:13]	Reserved	000
pll_div_val_l1-7	[12]	This field is used to set dividing value in both DIV_{APLL} and DIV_{APLL_HPM}. DIV_{APLL} clock divider ratio, $DOUT_{APLL} = MOUT_{APLL}$ / RATIO (*RATIO = pll_div_val_l1-7 + 1*), and DIV_{APLL_HPM} clock divider ratio, $DOUT_{APLL\_HPM} = MOUT_{APLL}$ / RATIO (*RATIO = pll_div_val_l1-7 + 1*) at IEM performance level from 1 to 7	0
Reserved	[11]	DO NOT CHANGE	0
hpm_div_val_l1-7	[10:8]	This field is used only in closed-loop mode (by using HPM). DIV_{HPM} clock divider ratio, SCLK_HPM = $DOUT_{ARM\_HPM}$ / RATIO (*RATIO = hpm_div_val_l1-7 + 1*) at IEM performance level from 1 to 7	000
Reserved	[7]	DO NOT CHANGE	0
arm_div_val_l1-7	[6:4]	This field is used to set dividing value in both DIV_{ARM} and DIV_{ARM_HPM}. DIV_{ARM} clock divider ratio, $DOUT_{ARM} = DOUT_{APLL}$ / RATIO (*RATIO = arm_div_val_l1-7 + 1*), and DIV_{ARM_HPM} clock divider ratio, $DOUT_{ARM\_HPM} = DOUT_{APLL\_HPM}$ / RATIO (*RATIO = arm_div_val_l1-7 + 1*), at IEM performance level from 1 to 7	000
Reserved	[3]	DO NOT CHANGE	0
bus_div_val_l1-7	[2:0]	DIV_{D0_BUS} clock divider ratio, HCLKD0 = $DOUT_{ARM}$ / RATIO (*RATIO = bus_div_val_l1-7 + 1*) at IEM performance level from 1 to 7	010

10.5.15 IEM_HPMCLK_DIV Register (IEM_HPMCLK_DIV, R, Address = 0xE010_8724)

The register reads the current IEM HPMCLK divider value.

IEM_HPMCLK_DIV	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0x0000_000
Reserved	[3]	Reserved	0
iem_hpmclk_div	[2:0]	IEM HPMCLK Divider Value	000

11 POWER MODE APPLICATION FOR SCENARIO

In this section, we describe how to apply system power mode for a specific operation scenario to consume as the minimum power as possible, for easy understanding. We consider a several operation scenario: Audio playback, Video playback, and Camcorder Recording.

Note that this is only an example for understanding, therefore you can apply system power mode in a different way if you think it is better approach.

In this description, we mention the optimal configurations in NORMAL, IDLE, and DEEP-IDLE mode. You can also use STOP, DEEP-STOP, and SLEEP mode if necessary.

11.1 AUDIO PLAYBACK

Power Mode	NORMAL	DEEP-IDLE
CPU	RUN	L2RENTION / POWER-OFF
MFC	Power-gating	←
G3D	Power-gating	←
Audio Sub-system	Power on	←
LCD Sub-system	Power on / Power-gating	←
TV Sub-system	Power-gating	←
TOP	Power on	Power-gating
ALIVE	Power on	←

← : Keep states (Run or power-gating) in NORMAL mode

Low-power MP3 playback application can be run according to the following sequence.

1. Loading MP3 file from NAND to input buffer in external DRAM.
2. Memory-copy data to input buffer**.**
3. Cache line fill for MP3 code and data.
4. Read input stream from input buffer in DRAM.
5. MP3 decoding (Include parsing).
6. Write output data to output buffer in DRAM.
7. Memory-copy output data from output buffer in DRAM to output buffer in SRAM located in Audio sub-system.
8. Enter DEEP-IDLE mode.
9. Audio codec out through I2S0 (located in Audio sub-system).
10. Exit DEEP-IDLE mode by wakeup event from I2S0 output buffer empty.
11. If file buffer is empty, go to STEP 1.
12. Else if input buffer is empty, go to STEP2.
13. Else go to STEP 4.

In STEP 8, the following setting should be made before entering DEEP-IDLE mode.

1. I2S_FIFO_CLK_SEL field of LPMP3_MODE_SEL register in CLKCON @0xE020_0308 should be set to 1'b1.
2. OP_CLK field of IISMODE register in I2S0 @0x9E20_2004 should be set to one of 00, 01, and 10.

11.2 VIDEO PLAYBACK

Power Mode	NORMAL	IDLE	DEEP-IDLE
CPU	RUN	STANDBY	L2RETENTION / POWER-OFF
MFC	Power on	←	←
G3D	Power on / Power-gating	←	←
Audio Sub-system	Power on	←	←
LCD Sub-system	Power on	←	←
TV Sub-system	Power on / Power-gating	←	←
TOP	Power on	←	←
ALIVE	Power on	←	←

← : Keep states (Run or power-gating) in NORMAL mode

11.3 CAMCORDER RECODING

Power Mode	NORMAL	IDLE	DEEP-IDLE
CPU	RUN	STANDBY	L2RETENTION / POWER-OFF
MFC	Power on	←	←
G3D	Power-gating	←	←
Audio Sub-system	Power on	←	←
LCD Sub-system	Power on / Power-gating	←	←
TV Sub-system	Power-gating	←	←
TOP	Power on	←	←
ALIVE	Power on	←	←

← : Keep states (Run or power-gating) in NORMAL mode

12 IEM CLOCK CHANGE SCHEME

In this section, we describe how to change HPM and ARM clocks if you want to change IEM performance.

IEM supports eight performance levels. It depends on system and application how many performance levels you use.

To implement clock change scheme, PMU in S5PC100 has 22 registers (6 registers for IEM configurations, 8 registers for PLL P/M/S value, and 8 register for clock dividers). Among 22 registers, 9 registers for IEM configurations are common to all performance levels, and two registers (one for PLL P/M/S value, the other for clock divider) are assigned to each performance level. (See Table 2.4-58)

Additionally, there are three registers for special clock management, namely, FREQ_CPU, FREQ_DPM, and DVSEMCLK_EN. The detailed descriptions are given in next section.

Table 2.4-58 Registers for IEM Clock Change

Performance Level	IEM configuration	PLL P/M/S value	Clock Divider value
8 level	DCGIDX_MAP0 DCGIDX_MAP1 DCGIDX_MAP2 DCGPERF_MAP0 DCGPERF_MAP1 DVCIDX_MAP	APLL_CON_L8	CLKDIV_IEM_L8
7 level		APLL_CON_L7	CLKDIV_IEM_L7
6 level		APLL_CON_L6	CLKDIV_IEM_L6
5 level		APLL_CON_L5	CLKDIV_IEM_L5
4 level		APLL_CON_L4	CLKDIV_IEM_L4
3 level		APLL_CON_L3	CLKDIV_IEM_L3
2 level		APLL_CON_L2	CLKDIV_IEM_L2
1 level		APLL_CON_L1	CLKDIV_IEM_L1

In stage of chip initialization, the above registers are set to defined values.

If we want to change IEM performance, we need to set only new target performance level to IEM_ IEC register, then the appropriate value corresponding to target performance level are used to change current target level.

0 Performance level is defined to clock stopped state, so if you change IEM performance level to 0 level, you should use IDLE mode of PMU, and WFI mode of Cortex-A8.

2.5 INTELLIGENT ENERGY MANAGEMENT

1 OVERVIEW

The IEM solution is designed primarily for battery powered equipment, where the requirement to have as long a battery life as possible is paramount. The IEM solution is ideal for portable applications, for example smartphones, feature phones, Personal Digital Assistants (PDA), hand held games consoles and portable media players.

Figure 2.5-1 shows a high-level block diagram of a complete IEM solution.

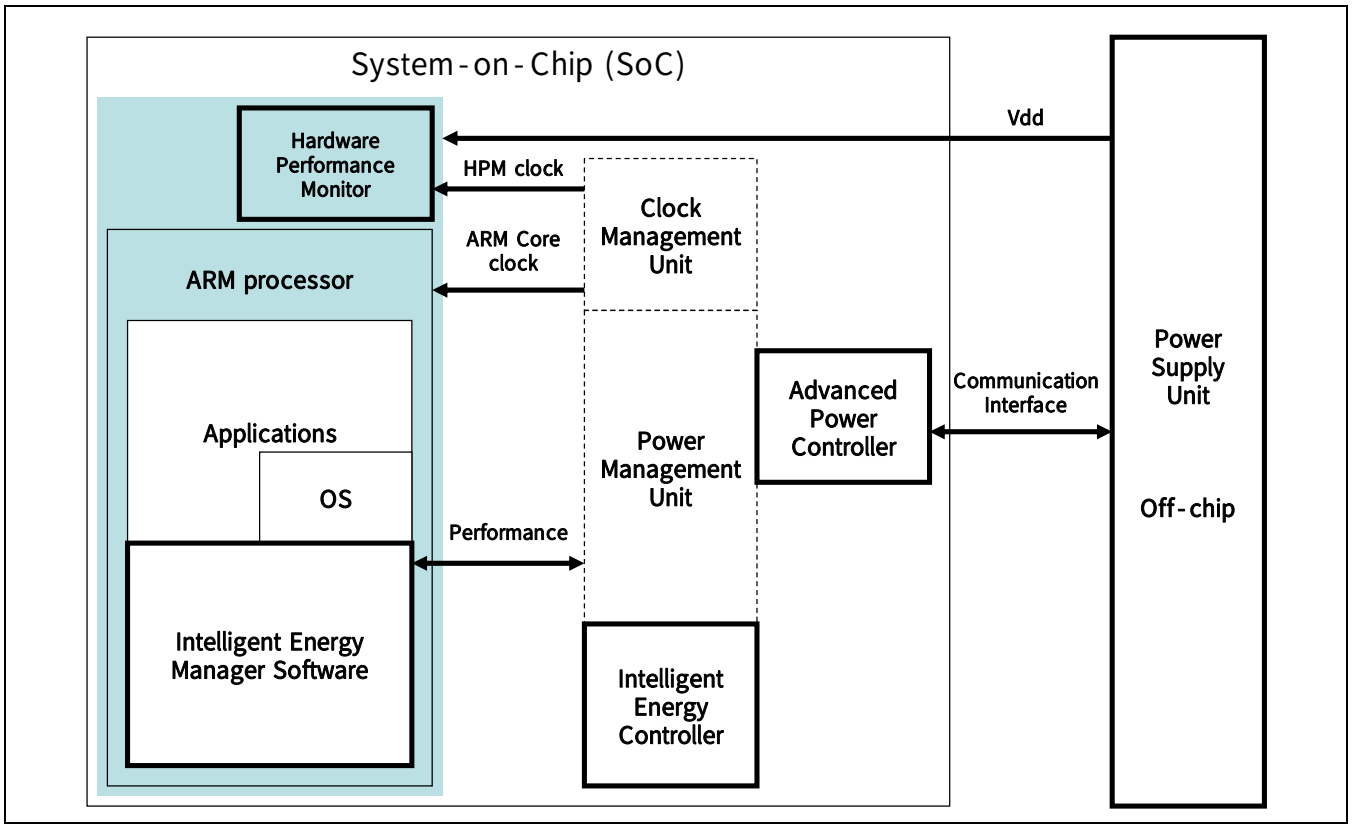


Figure 2.5-1 Intelligent Energy Manager Solution

An IEM system consists of the following:

* An OS modified to cooperate with the IEM software. (An IEM-enabled OS)
* The IEM software, ported to the platform that you are using.
* Performance scaling hardware and appropriate drivers for that hardware.

These parts of the system cooperate with each other to optimize power consumption, without compromising on performance or responsiveness.

Simply, an IEM system works as follows:

- When the IEM software starts, it registers some kernel hooks with the OS.
- The OS uses these hooks to call the IEM software. It does so whenever a system event occurs that might influence the optimum performance level.
- The IEM software records information about the events that occur, and the tasks related to them.
- The policies that are a part of the IEM software analyze this information to determine the optimum performance level.
- Whenever the optimum performance level changes, the IEM software uses the performance scaling hardware to set the new level.

1.1 FEATURE

- Up to eight energy level control
- Up to eight frequency level control
- Up to eight voltage level control
- Low power mode support

1.2 BLOCK DIAGRAM

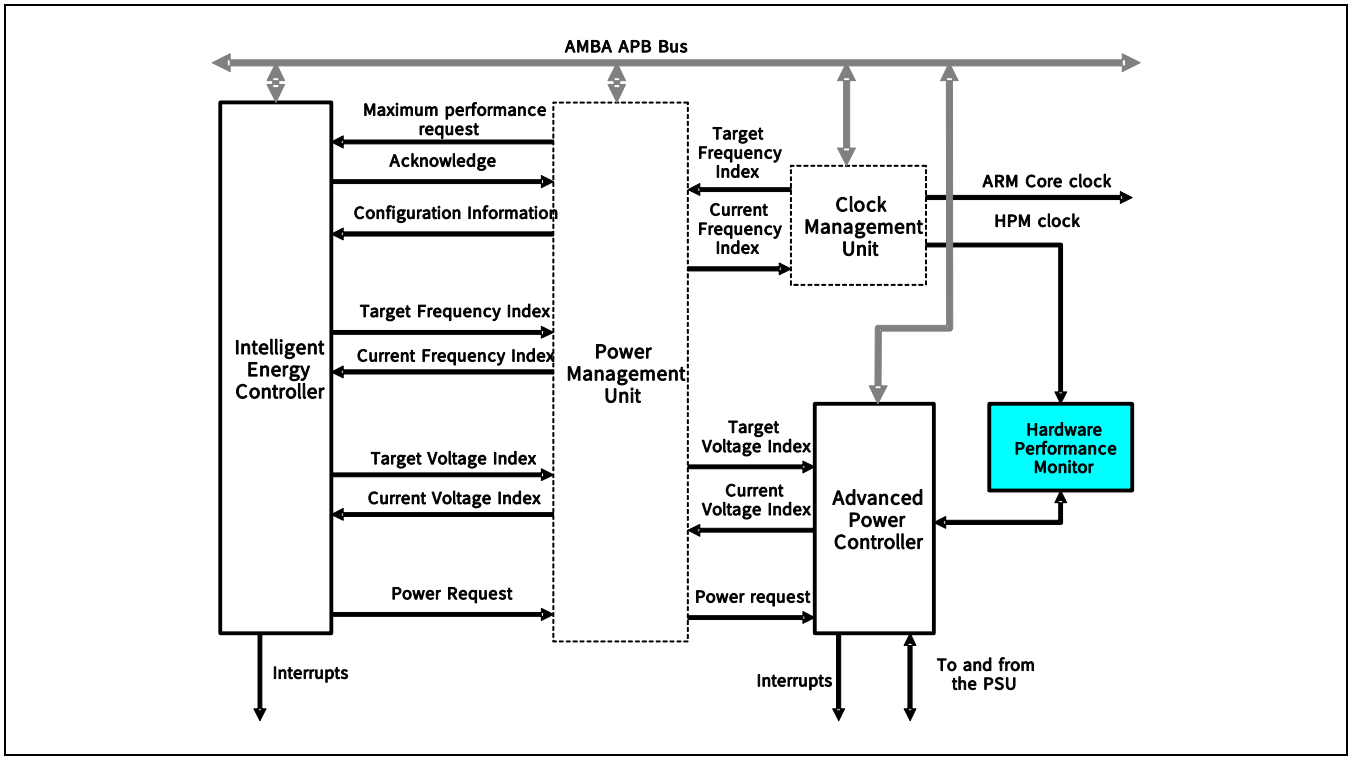


Figure 2.5-2 IEM Block Diagram

2 FUNCTIONAL DESCRIPTIONS

To support IEM, S5PC100 includes special IPs as below:

- Power Management Unit supporting IEM
- Intelligent Energy Controller
- Clock Management Unit supporting Dynamic Clock Generation
  - ♦ Clock Management Unit in System Controller acts as Dynamic Clock Generator
- Advanced Power Controller supporting Dynamic Voltage Control
  - ♦ APC1 acts as Dynamic Voltage Controller
- The Power Supply Unit supporting Dynamic Voltage Scaling
  - ♦ Power Supply Unit is the only off-chip component.
- Hardware Performance Monitor
  - ♦ This is optional and only required for a closed loop system

Figure 2.5-2 IEM Block Diagram shows the on-chip IEM components required for a complete solution of IEM and how they connect together briefly.

2.1 IEM SYSTEM COMPONENTS

2.1.1 Intelligent Energy Controller

The Intelligent Energy Controller (IEC) from ARM is designed for reuse in a wide variety of AMBA based designs and has a standard APB slave interface for programming the registers. This provides an Applications Programming Interface (API) for the IEM software. The IEC connects through defined interfaces to SoC-specific components such as the APC1.

The IEC uses prediction performance level requests from the IEM software. The performance setting is communicated to the IEC so that the System-on-Chip specific and product platform scaling hardware can be controlled to bring the system to that performance point. Battery life is extended by lowering the operating frequency and voltage of SoC components, such as the processor, and consequently reducing energy consumption.

The IEC provides an abstracted view of the SoC-specific performance scaling hardware. It is responsible for translating the performance prediction made by the IEM software (0-100% of maximum performance) into an appropriate performance point at which the system runs and then controlling the scaling hardware to achieve operation at that target point. This is achieved through passing a target performance request to the CMU and APC1.

The IEC also measures the work done in the system to ensure that the software deadlines are not going to be missed. Additionally, the IEC supports a maximum performance hardware request feature. The IEC is designed to map to an implementation-defined set of index levels. You must configure the IEC to define the CMU frequencies and APC1 voltage levels that can be selected. These frequencies and voltages depend on the capabilities of the dynamic or adaptive power supply technology to support multiple operating performance points.

The IEC interfaces to the CMU and APC1 blocks via PMU through a thermometer encoded interface protocol, that indicates to the IEC the current performance level. This protocol is specified to support interfacing across asynchronous clock domains between high-speed PLL and clock-generator and low-speed voltage scaling

hardware. The IEC provides an encoded performance index to SoC specific CMU and APC1 blocks.

The IEC also includes a Design for Test (DFT) interface. This enables easier control over the scaling hardware during production testing of the SoC device.

The IEC is an AMBA compliant, SoC peripheral that is developed, tested, and licensed by ARM Limited. The IEC features are as follows:

- AMBA APB compliant.
- Defined interfaces between the IEC and CMU/APC1 via PMU that are necessary for a complete energy management solution.
- An abstract interface to the underlying system-specific clock multiplexing and dynamic voltage or power control. This is through mapping to an implementation-defined set of index levels:
  - ♦ that correspond with the CMU frequencies that can be selected, and
  - ♦ that enable the voltage steps for the corresponding dynamic or adaptive power supply technology and consequently supports multiple operating performance points.
- An encoded interface protocol that provides a performance index to SoC specific CMU and APC1 blocks.
- Dynamic Voltage Scaling (DVS) emulation support enables a run fast then idle mode of operation.
- An API interface for efficient control and monitoring:
  - ♦ implementation-independent fractional performance setting interface to support performance prediction algorithms without hard-coded frequencies.
  - ♦ implementation-independent interrogation of performance-level quantization mapping levels to enable performance prediction software to adapt to the processor clock frequencies provided.
  - ♦ SoC-specific configuration interrogation, consisting of processor and IEC clock frequencies in kHz, and performance level mapping provided by the SoC specific CMU.
- Support for maximum performance signaling for real time subsystems that enables:
  - ♦ the maximum performance level to be requested regardless of the current programmed target performance level.
  - ♦ you to decide the events that activate this mode.
- Monitoring for IEM-specific algorithms, through a multi-channel interface designed to support automatic accumulation of system metrics.
- Support for synchronization handshaking with synchronous and asynchronous bridges to control entry and exit from maximum performance mode.
- Test registers for use in block and system level integration testing.
- System level integration testing using externally applied integration vectors.
- Debug mode for testing clock generation with maximum voltage.
- ID support registers for porting software driver compliance.
- DFT interface to control the target index outputs during SoC DFT.

2.1.2 Advanced Power Controller

In S5PC100X, Advanced Power Controller (APC1) from National Semiconductor is used for Dynamic Voltage Control.

The APC1 is an advanced power controller designed for reuse in the AMBA-based designs with a standard APB slave interface for programming registers. Based on the requested performance requirements from the CMU, the APC1 dynamically controls the EMU so that the voltage level provided to the SoC is sufficient enough for the required performance level. This is the minimum voltage for the best power saving. A thermometer-encoded interface is used for the APC1 to receive target performance level requirements, and to send out current performance level updates indicating voltage readiness.

Together with the HPM, the APC1 tracks the system timing in real time, and sends out voltage commands to the EMU to request the adjustment of voltage level. The flowchart in Figure 2.5-4 shows how the adaptive voltage control is processed to find optimum voltage level.

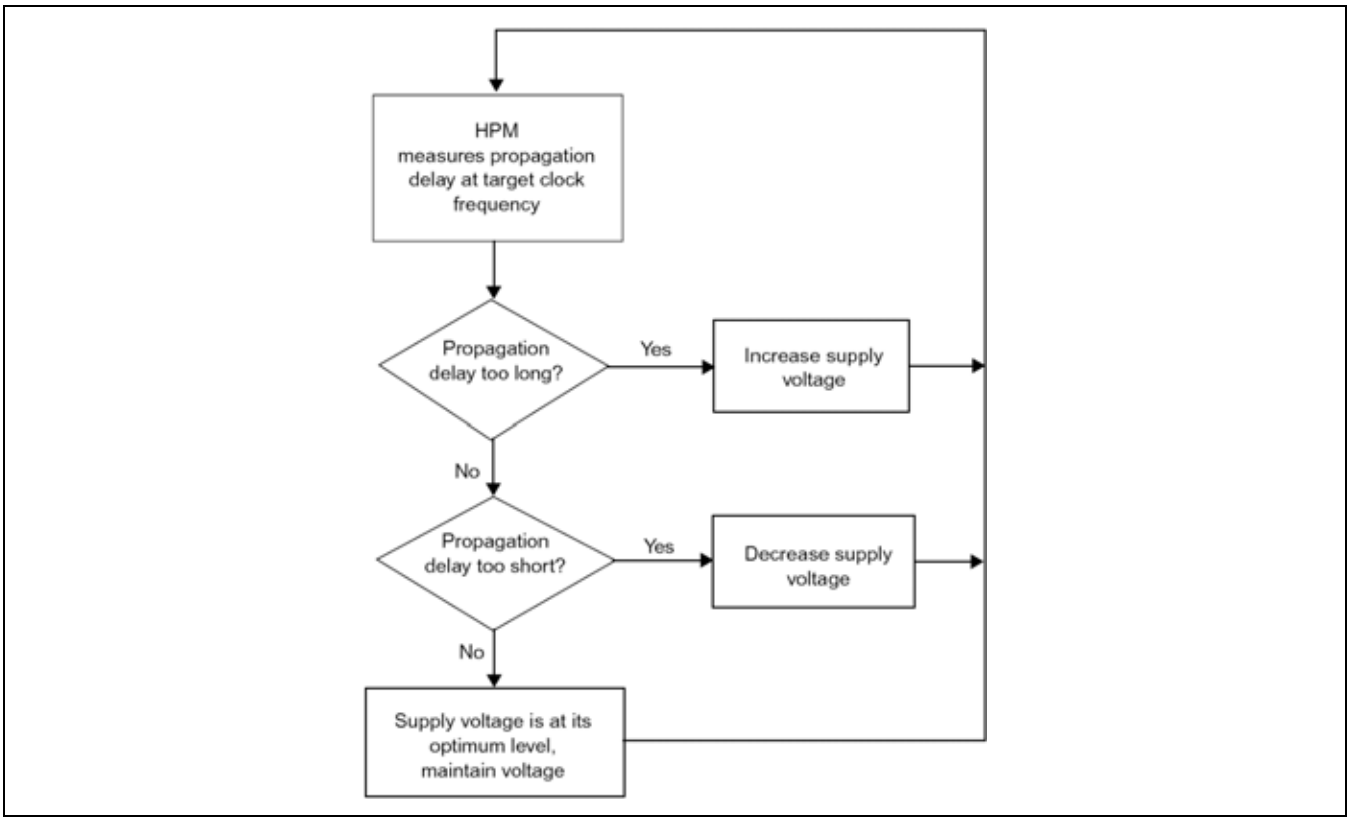


Figure 2.5-3 PowerWise Performance Tracking and Voltage Adjustment

If you require the open-loop Dynamic Voltage Scaling (DVS) voltage control you can use a built-in voltage table to request the EMU voltage level corresponding to the target performance level.

The APC1 is an AMBA APB-based SoC peripheral. The features are as follows:

- AMBA APB interface for programming the registers
- PowerWise Interface™ (PWI) Rev 1.0 compliant master for controlling an external PWI-compliant power supply

- supports the closed-loop AVS voltage control in conjunction with the HPM
- voltage table to support the open-loop DVS
- supports thermometer-encoded interface for a target performance level request and a current performance level update
- parameterized design supports up to eight performance levels
- sleep mode (retention level) power-down support
- revision identification register for porting software driver compliance
- DFT-ready for SCAN-based ATPG.

The APC1 receives the required target performance request from the IEC via PMU. This performance request is then translated to a voltage level that is communicated to the PSU through an interface such as the. The PWI has been developed jointly by ARM and National Semiconductor to provide a high-speed, low-power control interface between an IEM-enabled SoC and an external power supply unit.

For an open loop system, the APC1 can either:

- wait a programmed time that is dependent on the response time of the PSU, before signaling to the CMU that the target performance can be achieved
- interrogate the PSU through the PWI for a VDD_OK signal indication.

If the PSU provides intermediate stable voltage level indication, then the APC1 can also determine this through the PWI.

2.1.3 Hardware Performance Monitor

The Hardware Performance Monitor (HPM) is designed for reuse and easy implementation. Although it is a separate entity in physical partition, the HPM is an integral part of the APC1 for an AVS power management system. The HPM is not a memory mapped device. An HPM is required for closed loop control, but not for an open loop control system.

The HPM tracks the system delay. The output of the HPM is a function of voltage level and the HPM clock. As shown in Figure 2.5-1, the HPM is embedded in the ARM Core voltage domain that is AVS controlled. It receives the clock from the CMU, and outputs are connected to the APC1. It translates voltage level into system delay information. The system delay information is then used by the APC1 to determine the optimum voltage level for the target performance requirement.

To be short, the CMU supplies the target frequency required by the IEM software for that voltage domain, and the HPM informs the APC1 when this target frequency is detected.

The HPM design is structurally coded in the synthesizable RTL to facilitate ease of place and route. This is required to optimize the accuracy of the system delay tracking.

The HPM features are as follows:

- configurable for a different target frequency
- low power consumption overhead
- low area overhead
- DFT-ready for SCAN based ATPG

2.1.4 Power Management Unit

In S5PC100X, the Power Management Unit (PMU) supports IEM features. The PMU provides the IEC with configuration information, for example:

- fractional index map, indicating the fractional levels supported
- performance map, providing the mapping of the performance levels onto the clock frequencies supported by the CMU
- maximum processor performance.

2.1.5 Clock Management Unit

In S5PC100X, Clock Management Unit (CMU) in System Controller supports Dynamic Clock Generation.

The CMU receives target performance requests from the IEC, via Power Management Unit (PMU). It generates the necessary clocks for the CPU, for example:

- processor clock
- peripheral clocks
- AMBA clock.

Additionally, for a more efficient design, the CMU must be capable of generating the different performance levels as indicated by the IEC. The CMU can also be a memory mapped AMBA peripheral and can contain both control and status registers.

The design of the CMU must meet the requirements set by the IEC and the Advanced Power Controller (APC1). These constraints are necessary to ensure optimum and correct performance of the Hardware Performance Monitor (HPM).

2.1.6 Power Supply Unit supporting Dynamic Voltage Scaling

The Power Supply Unit (PSU) is the only off-chip component. The PSU provides the requested voltage to the SoC. It interfaces to the DVC through an interface such as the PWI. It ensures that the voltage targets specified by the DVC are provided to the SoC.

2.2 IEM SYSTEM OPERATION

Loading and starting the software

At an appropriate stage of system boot-up, the OS loads and initializes the modules that contain the IEM software:

- on most platforms, the module loader automatically runs the initialization code for a module (if any)
- otherwise, the OS (or a driver) must call the initialization code itself.

This initialization code performs much of the set up for the IEM software. For example:

- the code in the IEM HAL sets up and configures the performance scaling hardware
- the code in the Control component loads the Comms driver that it uses to communicate with the IEM kernel.

The OS then configures the IEM kernel by issuing commands to the Control component. The Control component encodes these commands as messages, and uses the Comms driver to send them to the IEM kernel. These control messages:

- start the policies, so that they are ready to use
- optionally:
  - ♦ configure the IEM activities that are traced
  - ♦ enable tracing.

Finally, the OS issues a command to start the IEM kernel. When the IEM kernel receives the corresponding control message, it:

1. Allocates memory for the event queue, and initializes it.
2. Allocates memory for the IEM blocks, and initializes them.
3. Registers the kernel hooks that the OS calls whenever a system event occurs.

2.2.1 Handling system events

When an event occurs that might influence the optimum performance level, the OS calls the appropriate kernel hook in the IEM kernel:

- The New Task hook is called whenever a new task is created. This hook generates a New Task system event for the new task that has just been created.
- The Exit hook is called whenever a task is about to exit. This hook generates a Task Exit system event for the exiting task.
- The Task Switch hook is called whenever the OS switches from one task to another. This hook generates two system events:
  - ♦ a Task Schedule Out system event for the previous task that has just been switched out
  - ♦ a Task Schedule In system event for the next task that is being switched in.
- The User Input hook is called whenever a task receives user input. This hook generates a User Input system event for the task that is receiving input.

When a kernel hook generates an system event, it determines whether any event handlers recognize the system event. If so, it:

- creates a structure describing the system event
- ensures that there is an IEM block describing the corresponding task
- runs the fast event handlers to process the system event.

The kernel hook then determines whether any standard event handlers recognize the system event. If so, the kernel hook adds the event to the event queue, for subsequent processing by the standard event handlers.

The kernel hook finally ensures that, if there are any system events in the event queue, the standard event handlers run within a given period.

2.2.2 Running the fast event handlers

The fast event handlers are run from the kernel hooks whenever a system event occurs.

For each policy, the IEM kernel determines whether its fast event handler recognizes the system event. If so, the IEM kernel runs the fast event handler, passing it pointers to the IEM kernel data structures that include:

- the system event structure describing the event
- the IEM block describing the task that triggered the system event.

The fast event handler then processes the event. Typical uses of the fast event handler include:

- Recognizing a task that requires an immediate change in performance level, and requesting that performance level. The fast event handler might recognize:
  - ♦ a specific task, such as a movie player
  - ♦ a type of task, such as real-time tasks, or tasks that are receiving user input.

If necessary, the fast event handler can get further information about the task by making calls to the OS layer API.

- Storing policy-specific information about the current state of the task or the system, for later processing by the standard event handler of the same policy. The fast event handler might get this information by making calls to the IEM HAL or OS layer APIs. It typically stores this information in arrays of memory that are allocated by the initialization function of the policy.

When the fast event handlers have been run, the IEM kernel then combines any performance requests that the fast event handlers are making, and sets the resulting performance level using the IEM HAL.

2.2.3 Running the standard event handlers

The standard event handlers are run periodically by the IEM kernel.

When the IEM kernel determines that it must run the standard event handlers, there are typically a number of outstanding system events in the event queue, that have not yet been processed by the standard event handlers. Starting with the oldest event, the IEM kernel processes each event in turn by enabling pre-emption, and then running the standard event handlers.

The standard event handlers are run in a very similar way to the fast event handlers. For each policy, the IEM kernel determines whether its standard event handler recognizes the system event. If so, the IEM kernel runs the standard event handler, passing it pointers to the IEM kernel data structures that include:

- the system event structure describing the event
- the IEM block describing the task that triggered the system event.

The standard event handler then processes the event, analyzing the data in the IEM kernel data structures and any data that was stored by the fast event handler to determine the optimum performance level. The analysis that the standard event handler performs is usually very different to that performed by the fast event handler of the same policy. This is because the standard event handler is working on historical data. Also, the standard event handler is pre-emptable, and so can spend longer analyzing the data without impacting system responsiveness. It can therefore use more complex algorithms, such as decaying weighted averages.

When the final outstanding event in the queue is processed, the standard event handlers can request a performance level. The IEM kernel then combines any performance requests that the standard event handlers are making, and sets the resulting performance level using the IEM HAL.

3 IEM IMPLEMENTATION AND DRIVER SETTING

3.1 DEFINITION OF PERFORMANCE

The maximum frequency of APLL is 1.6GHz. The expected frequency range of ARM Core is from 166MHz to 800MHz. AXI_D0 bus, which is connected to ARM Core, works at 166MHz.

In S5PC100X, CMU only uses clock divider to change performance. If you want to use PLL clock change, you should change PLL setting for that. With this specification, we should consider about number of the frequency levels as well as the resolution of each frequency level.

There are divider values for ARM Core clock, AXI_D0 bus AXI clock and HPM clock when PLL output is 1600MHz in table below.

Table 2.5- 1 Example Divider Values for 1600MHz PLL Output

PLL Output (MHz)	ARM Core Clock Frequency (MHz)	HPM Clock Frequency (MHz)	AXI Bus Clock Frequency (MHz)	ARM Clock Ratio (f_{PLL}/f_{ARM})	HPM Clock Ratio (f_{ARM}/f_{HPM})	AXI Bus Clock Ratio (f_{PLL}/f_{AXI})	Performance mapping (%)
1600	800.0	100.0	160.0	2	8	10	100.0
	533.3	66.7		3			66.7
	400.0	50.0		4			50.0
	320.0	40.0		5			40.0
	266.7	33.3		6			33.3
	228.6	28.6		7			28.6
	200.0	25.0		8			25.0
	177.8	22.2		9			22.2
	160.0	20.0		10			20.0

There are divider values for ARM Core clock, AXI_D0 bus AXI clock and HPM clock when PLL output is 833MHz in table below.

Table 2.5- 2 Example Divider Values for 833MHz PLL Output

PLL Output (MHz)	ARM Core Clock Frequency (MHz)	HPM Clock Frequency (MHz)	AXI Bus Clock Frequency (MHz)	ARM Clock Ratio (f_{PLL}/f_{ARM})	HPM Clock Ratio (f_{ARM}/f_{HPM})	AXI Bus Clock Ratio (f_{PLL}/f_{AXI})	Performance mapping (%)
833	833.0	104.1	166.6	2	8	5	100.0
	416.5	52.1		3			50.0
	277.7	34.7		4			33.3
	208.3	26.0		5			25.0
	166.6	20.8		6			20.0

If you want to add more performance level above 50%, you should put PLL change scheme to CMU.

3.2 HPM STRUCTURE AND CLOSED-LOOP BEHAVIOR

When IEM works with closed-loop, HPM and APC1 work like as Figure 2.5-4 and Figure 2.5-5.

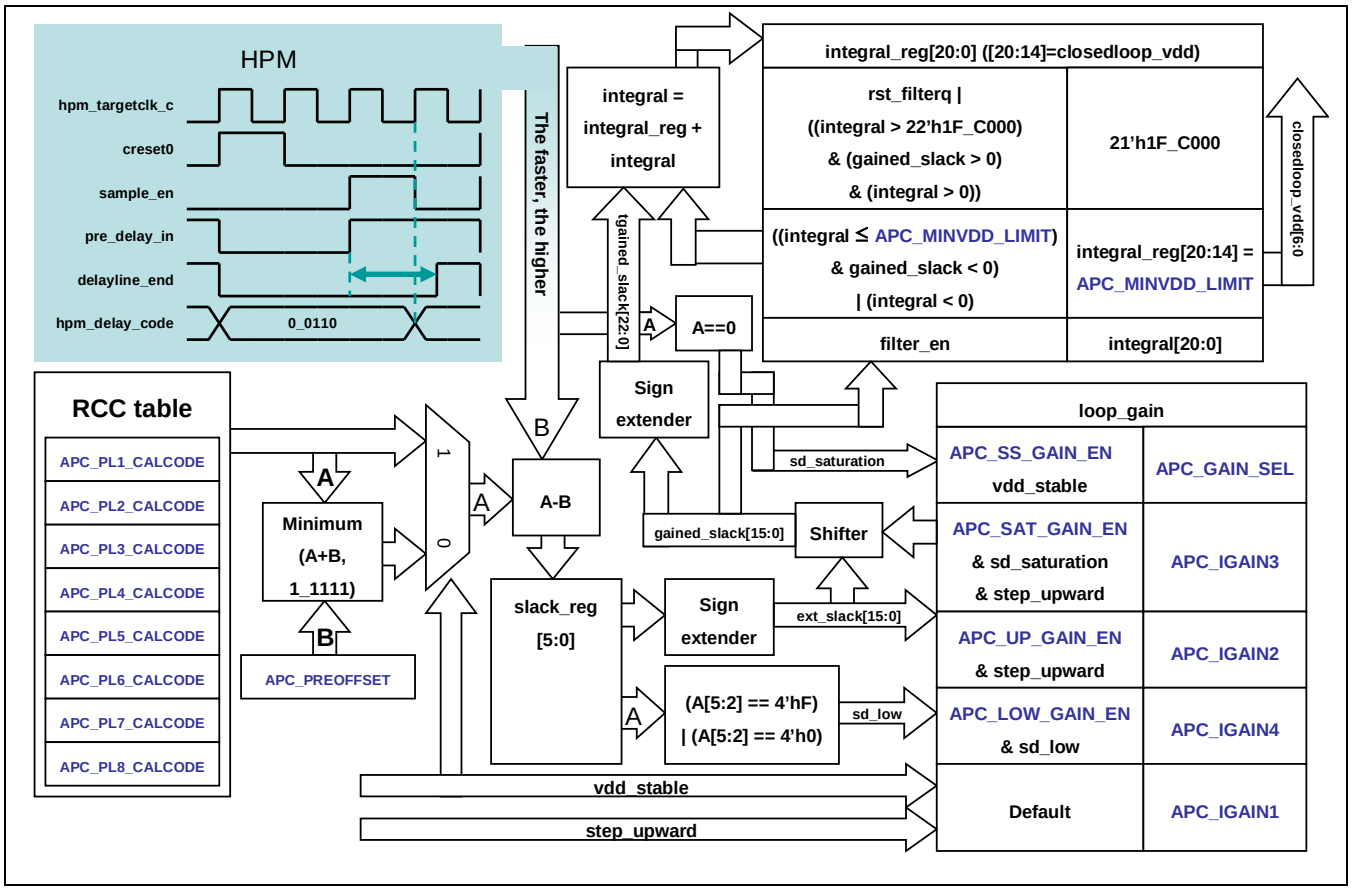


Figure 2.5-4 IEM Closed-loop Voltage Generation Flow in HPM and APC1

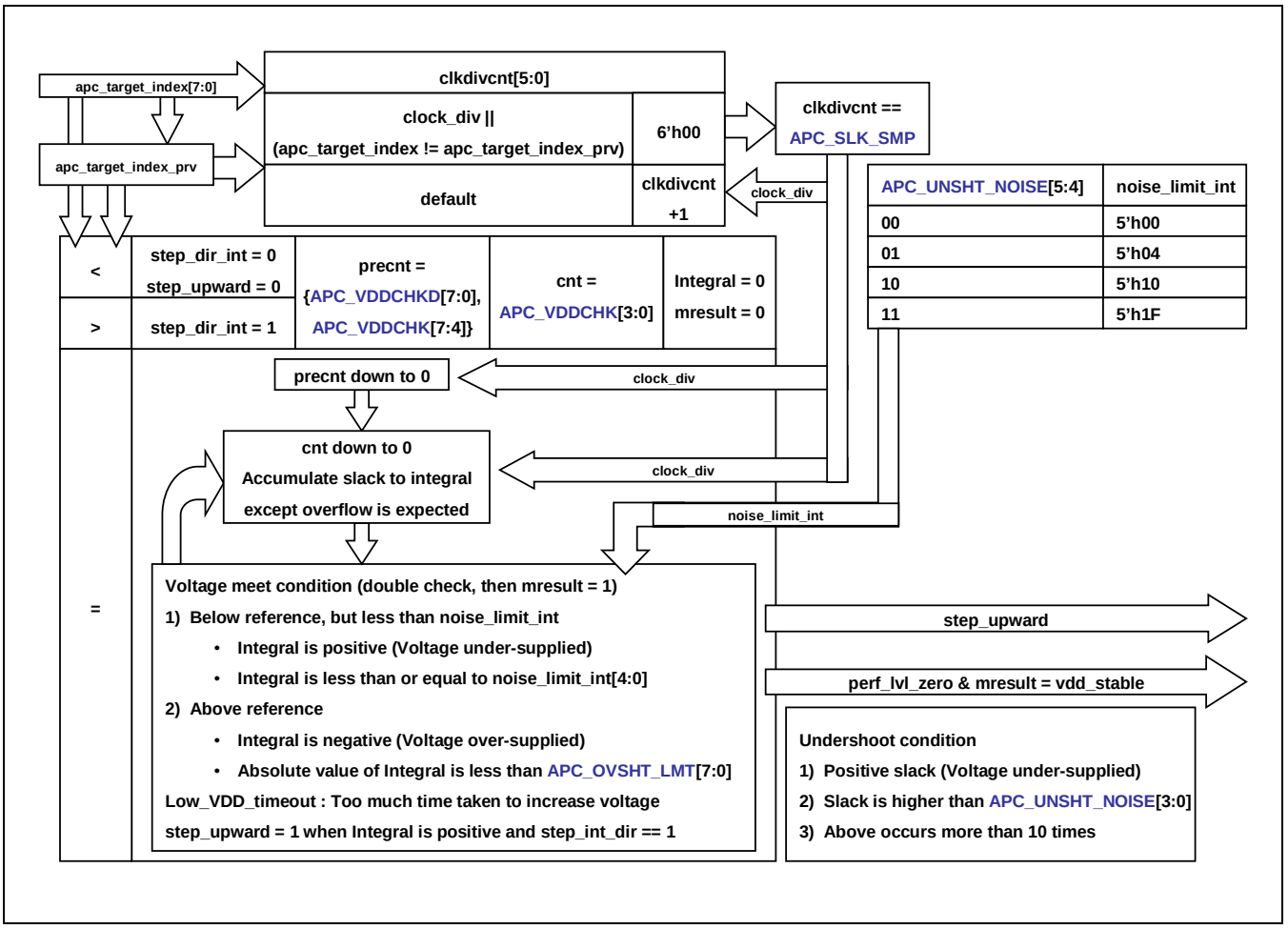


Figure 2.5-5 IEM Closed-loop Control Flow in APC1

3.2.1 HPM Delay

Critical path delay of ARM Core in S5PC100X is about 1.70ns in the worst condition. Delay of NOR2X1 cell is about 0.04609ns. One delay tap has four NOR2X1 cells and each delay tap gives 0.184ns delay. Delay tap structure is as shown in Figure 2.5-6.

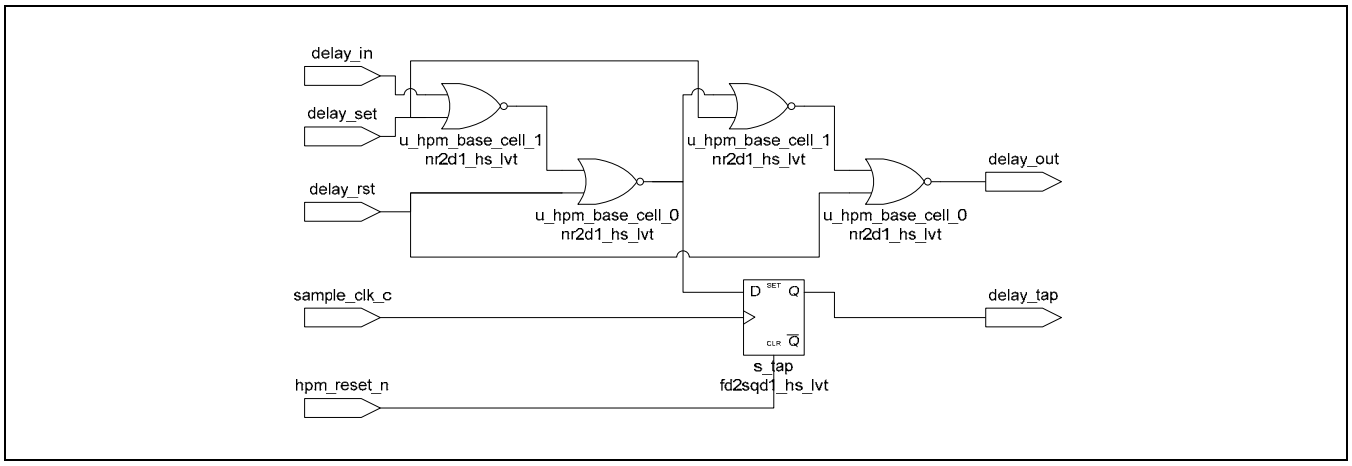


Figure 2.5-6 HPM Delay Tap Structure in S5PC100X

HPM has a predelay module that includes 32 delay tap-like delay elements and a delayline module that includes 32 delay taps. To correlate with ARM core, 14-th tap should be selected with setting predelay_sel[2:0] of HPM 3'b000 when HPM clock ratio is equal to 1.

3.2.2 Calibration Code for Closed-loop

In closed-loop mode, Calibration codes are used to control voltage level, while voltage values in open-loop mode. Calibration code stands for critical path delay of ARM core. In S5PC100X, 14-th tap output of HPM has the nearly same delay to the critical path of ARM core (when HPM clock ratio is equal to 1), which can be encoded to the delay code 5'hE.

3.3 INITIALIZATION SEQUENCE

1). Initialize the index map & all other IEM & APC mapping values.

2) If IEM will use 'overdrive' level, then programs 'Max performance mapping index value' in IECDPCCR register with proper values (smaller than 3'b111)

3) Enables voltage scaling feature in the APC by setting 'APC_VDD_UD' bit in APC_CONTROL register as "1"

4) If IEM will use closed loop mode, then programs 'APC_HPM_EN' bit & 'APC_LOOP_MODE' bit in APC_CONTROL register as "1"

5) Start IEM HW by setting 'iem_enable' bit in IEM_CONTROL register in power management unit as "1"

6) Start IEM control by setting "iec_enable" bit in IECDPCCR register as "1"

7) Now, the system is under the IEM control.

3.4 PWI (POWER WISE INTERFACE)

The PWI master drives the PWI clock line. Although a pull-down resistor is connected between the SCLK and GND-voltage on the slave, the PWI master drives the clock signal actively high and low.

The SCLK frequency range is 0 MHz ~ 15 MHz. The clock runs only when data is being transferred. Otherwise the SCLK signal line is at logic low voltage. Minimum pulse width of the clock signal is 26ns.

The PWI data-line is bi-directional. Data is written on the falling edge of the SCLK and read on the rising edge of the SCLK.

4 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	type
IEM_SCLK	Bidirectional	PWI clock	XiemSCLK	dedicated
IEM_SPWI	Bidirectional	PWI serial data	XiemSPWI	dedicated

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

5 REGISTER DESCRIPTION

5.1 REGISTER SUMMARY

Table 2.5-3 Register Summary of IEC

Register	Address	R/W	Description	Reset value
IECDPCCR	0xE110_0000	R/W	DPC Control Register	0x000000E0
IECDVSEMSTR	0xE110_0004	R/W	DVS Emulation Slot Time Register	0x63
IECDPCTGTPERF	0xE110_0008	WO	DPC Target Performance Register	0x80
IECDPCCRNTPERF	0xE110_000C	RO	DPC Current Performance Register	System Dependent
IECIMSC	0xE110_0010	R/W	Interrupt Mask Set and Clear Register	0x3
IECRIS	0xE110_0014	RO	Raw Interrupt Status Register	0x0
IECMIS	0xE110_0018	RO	Masked Interrupt Status Register	0x0
IECICR	0xE110_001C	WO	Interrupt Clear Register	0x0
IECCFGCPUFREQ	0xE110_0020	RO	Configured CPU Frequency Register	From PMU
IECDPMFREQ	0xE110_0024	RO	DPM Frequency Register	From PMU
IECCFGDCGIDXMAP00	0xE110_0040	RO	Configuration Fractional Index Map0	From PMU
IECCFGDCGIDXMAP32	0xE110_0044	RO	Configuration Fractional Index Map32	From PMU
IECCFGDCGIDXMAP64	0xE110_0048	RO	Configuration Fractional Index Map64	From PMU
IECCFGDVCIDXMAP	0xE110_004C	RO	Configuration DVC Index Map Register	From PMU
IECCFGDCGPERFMAP0	0xE110_0060	RO	Configuration Performance Map 0	From PMU
IECCFGDCGPERFMAP4	0xE110_0064	RO	Configuration Performance Map 4	From PMU
IECDPMCR	0xE110_0100	R/W	DPM Command Register	0x000
IECDPM2RATE	0xE110_0108	R/W	DPM Channel 2 Rate Register	0x80
IECDPM3RATE	0xE110_010C	R/W	DPM Channel 3 Rate Register	0x80
IECDPMILO	0xE110_0180	RO	DPM Channel 1 Low Register	0x00000000
IECDPM1H1	0xE110_0184	RO	DPM Channel 1 High Register	0x00000000
IECDPM2LO	0xE110_0188	RO	DPM Channel 2 Low Register	0x00000000
IECDPM2HI	0xE110_018C	RO	DPM Channel 2 High Register	0x00000000
IECDPM3LO	0xE110_0190	RO	DPM Channel 3 Low Register	0x00000000
IECDPM3HI	0xE110_0194	RO	DPM Channel 3 High Register	0x00000000
IECPeriphID4	0xE110_0FD0	RO	Peripheral Identification Register 4	0x03
IECPeriphID5	0xE110_0FD4	RO	Peripheral Identification Register 5	0x08
IECPeriphID6	0xE110_0FD8	RO	Peripheral Identification Register 6	Reserved

Register	Address	R/W	Description	Reset value
IECPeriphID7	0xE110_0FDC	RO	Peripheral Identification Register 7	Reserved
IECPeriphID0	0xE110_0FE0	RO	Peripheral Identification Register 0	0x50
IECPeriphID1	0xE110_0FE4	RO	Peripheral Identification Register 1	0x17
IECPeriphID2	0xE110_0FE8	RO	Peripheral Identification Register 2	0x04
IECPeriphID3	0xE110_0FEC	RO	Peripheral Identification Register 3	0x08
IECID0	0xE110_0FF0	RO	IEC Identification Register 0	0x0D
IECID1	0xE110_0FF4	RO	IEC Identification Register 1	0xF0
IECID2	0xE110_0FF8	RO	IEC Identification Register 2	0x05
IECID3	0xE110_0FFC	RO	IEC Identification Register 3	0xB1

Table 2.5-4 Register Summary of APC

Register	Address	R/W	Description	Reset value
APC_PWICMD	0xE100_0000	R/W	PWI Command Register	0x00
APC_PWIDATAWR	0xE100_0004	R/W	PWI Write Data Register	0x00
APC_PWIDATARD	0xE100_0008	RO	PWI Read Data Register	0x00
APC_CONTROL	0xE100_0010	R/W	APC Control Register	0x00
APC_STATUS	0xE100_0014	RO	APC Status Register	0x00
APC_MINVDD_LIMIT	0xE100_0018	R/W	Minimum Limit Register	0x00
APC_VDDCHK	0xE100_001C	R/W	VDD Check Register	0x00
APC_VDDCHKD	0xE100_0020	R/W	VDD Delay Time Register	0x00
APC_PREDLYSEL	0xE100_0024	R/W	VDD Pre-delay Select Register	0x07
APC_IMASK	0xE100_0028	R/W	APC Interrupt Mask Register	0x00
APC_ISTATUS	0xE100_002C	RO	APC Interrupt Status Register	0x00
APC_ICLEAR	0xE100_0030	WO	APC Interrupt Clear Register	0x00
APC_UNSH_NOISE	0xE100_0034	R/W	APC Undershoot Threshold and Noise Limit Register	0x00
APC_WKUP_DLY	0xE100_0038	R/W	Wakeup Delay Register	0x00
APC_SLK_SMP	0xE100_003C	R/W	Slack Sample Count Register	0x00
APC_CLKDIV_PWICLK	0xE100_0040	R/W	PWI Clock Division Register	0x00
APC_OVSHT_LMT	0xE100_0050	R/W	APC Overshoot Limit Register	0x00
APC_CLP_CTRL	0xE100_0054	R/W	APC Closed-loop Control	0x00
APC_SS_SRATE	0xE100_0058	R/W	APC Steady State Slew Rate Register	0x00
APC_IGAIN4	0xE100_005C	R/W	Integrator's Gain 4 Register	0x00
APC_IGAIN1	0xE100_0060	R/W	Integrator's Gain 1 Register	0x00
APC_IGAIN2	0xE100_0064	R/W	Integrator's Gain 2 Register	0x00
APC_IGAIN3	0xE100_0068	R/W	Integrator's Gain 3 Register	0x00
APC_PL1_CALCODE	0xE100_0080	R/W	Calibration Code 1 Register	0x1F
APC_PL2_CALCODE	0xE100_0084	R/W	Calibration Code 2 Register	0x1F
APC_PL3_CALCODE	0xE100_0088	R/W	Calibration Code 3 Register	0x1F
APC_PL4_CALCODE	0xE100_008C	R/W	Calibration Code 4 Register	0x1F
APC_PL5_CALCODE	0xE100_0090	R/W	Calibration Code 5 Register	0x1F
APC_PL6_CALCODE	0xE100_0094	R/W	Calibration Code 6 Register	0x1F
APC_PL7_CALCODE	0xE100_0098	R/W	Calibration Code 7 Register	0x1F
APC_PL8_CALCODE	0xE100_009C	R/W	Calibration Code 8 Register	0x1F
APC_PL1_COREVDD	0xE100_00A0	R/W	Open-loop VDD Core Register 1	0x7F
APC_PL2_COREVDD	0xE100_00A4	R/W	Open-loop VDD Core Register 2	0x7F
APC_PL3_COREVDD	0xE100_00A8	R/W	Open-loop VDD Core Register 3	0x7F

Register	Address	R/W	Description	Reset value
APC_PL4_COREVDD	0xE100_00AC	R/W	Open-loop VDD Core Register 4	0x7F
APC_PL5_COREVDD	0xE100_00B0	R/W	Open-loop VDD Core Register 5	0x7F
APC_PL6_COREVDD	0xE100_00B4	R/W	Open-loop VDD Core Register 6	0x7F
APC_PL7_COREVDD	0xE100_00B8	R/W	Open-loop VDD Core Register 7	0x7F
APC_PL8_COREVDD	0xE100_00BC	R/W	Open-loop VDD Core Register 8	0x7F
APC_RET_VDD	0xE100_00C0	R/W	Retention VDD Register	0x00
APC_ITSTOP3	0xE100_00C4	R/W	Integration Test Output Read or Set Register 3	0x00
APC_DBG_DLYCODE	0xE100_00E0	RO	Debug Performance Register	0x00
APC_REV	0xE100_00FC	RO	Revision Number Register	0x01

NOTE: All registers of IEM interface are accessible by word unit with STR/LDR instructions.

5.2 IEC RELATED REGISTERS

5.2.1 DPC Control Register (IECDPCCR, R/W, Address = 0xE110_0000)

IECDPCCR	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	0
Max Performance mapping index value	[7:5]	When IECMAXPERF goes high, the IEC requests maximum performance level which is decided by this register value. The reset value is 3'b111 which is literally max performance. However, if 3'b111 performance level needs overdrive, it is not desirable to overdrive SoC on every interrupt (MAXPERF case). In that case, software programs this register as lower value than the value needs overdrive.	0x7
Synchronous Mode Handshaking Enable	[4]	Enable/disable the use of the synchronous mode handshaking control signals. 0 = Synchronous mode handshaking disabled, also the reset value 1 = Synchronous mode handshaking enabled When this bit is set, the synchronous mode handshaking signals are used to control entry and exit from the maximum performance mode. When this bit is cleared, the handshaking signals are not used.	0
IEC Software Debug Emulation	[3]	Control to debug performance scaling. 0 = IEC performance scaling software debug disabled, also the reset value 1 = IEC performance scaling software debug enabled When this bit is seta, the performance level driven out of the **IECTGTDVCIDX** is set to maximum regardless of the software request. The performance level changes are only visible on **IECTGTDCGIDX**.	0
IEC Max Perf Enable	[2]	Enable/disable maximum performance mode override. 0 = IEC maximum performance mode disabled, also the reset value 1 = IEC maximum performance mode enabled. When this bit is set, the maximum performance mode is enabled and therefore whenever **IECMAXPERF** goes high, the IEC requests maximum performance level regardless of the current software request.	0
IEC PWM DVS En	[1]	Enable/disable the IEC PWM DVS mode. 0 = IEC PWM DVS mode disabled, also the reset value 1 = IEC PWM DVS mode enabled. When this bit is set, the IEC requests power through the **IECPWRREQ** output. The target performance index outputs are set to either maximum or minimum depending on the PWM state.	0
IEC Enable	[0]	Controls enabling and disabling of the IEC. 0 = IEC Disabled, also the reset value 1 = IEC Enabled When this bit is set, the IEC is enabled for performance scaling. When the bit is cleared, the IEC always requests	0

		maximum performance.	

5.2.2 DVS Emulation Slot Time Register (IECDVSEMSTR, R/W, Address = 0xE110_0004)

IECDVSEMSTR	Bits	Description	Reset Value
Reserved	[31:10]	Reserved, read undefined, do not modify.	0
Slot time	[9:0]	The time in µs for each slot of a PWM frame. This is reset to 0x63. For example, if you want each time slot to be 100µs in length, then the slot time bits must be programmed with 0x63. Similarly, if you want each time slot to be 200µs in length, then the slot time bits must be programmed with 0xC7a.	0x63

5.2.3 DPC Target Performance Register (IECDPCTGTPERF, W, Address = 0xE110_0008)

IECDPCTGTPERF	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	0
IECDPCTGTPERF	[7:0]	Sets the target fractional performance level. At system reset, the value 0x80 (100%).	0x80

NOTE: This register is read as 0x00.

5.2.4 DPC Current Performance Register (IECDPCCRNTPERF, R, Address = 0xE110_000C)

IECDPCCRNTPERF	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	0
IECDPCCRNTPERF	[7:0]	Returns the current performance level as indicated to the IEC by the DCG on the IECCRNTDCGIDX inputs.	System Dependent

5.2.5 Interrupt Mask Set and Clear Register(IECIMSC, R/W, Address = 0xE110_0010)

IECIMSC	Bits	Description	Reset Value
Reserved	[31:2]	Reserved, read undefined, do not modify.	0
CPU Sleep Interrupt Mask (CSIM)	[1]	On a read, the current mask of the CSIM is returned. On a write of 1, the mask of CSIM interrupt is set. A write of 0 clears the mask. The reset value is 1.	1
CPU Wake-up Interrupt Mask (CWIM)	[0]	On a read, the current mask of the CWIM is returned. On a write of 1, the mask of CWIM interrupt is set. A write of 0 clears the mask. The reset value is 1.	1

5.2.6 Raw Interrupt Status Register (IECRIS, R, Address = 0xE110_0014)

IECRIS	Bits	Description	Reset Value
Reserved	[31:2]	Reserved, read undefined, do not modify.	0
CPU Sleep Interrupt Status (CSRIS)	[1]	Returns the raw interrupt state prior to masking of the IECCPUSLPINT interrupt. The reset value is 0.	0
CPU Wake-up Interrupt Status (CWRIS)	[0]	Returns the raw interrupt state prior to masking of the IECCPUWUINT interrupt. The reset value is 0.	0

5.2.7 Interrupt Masked Interrupt Status Register (IECMIS, R, Address = 0xE110_0018)

IECMIS	Bits	Description	Reset Value
Reserved	[31:2]	Reserved, read undefined, do not modify.	0
CPU Sleep Masked Interrupt Status (CSMIS)	[1]	Gives the masked interrupt state (after masking) of the IECCPUSLPINT interrupt. The reset value is 0.	0
CPU Wake-up Masked Interrupt Status (CWMIS)	[0]	Gives the masked interrupt state (after masking) of the IECCPUWUINT interrupt. The reset value is 0.	0

5.2.8 Interrupt Clear Register (IECICR, W, Address = 0xE110_001C)

IECICR	Bits	Description	Reset Value
Reserved	[31:2]	Reserved, read undefined, do not modify.	0
CPU Sleep Interrupt Clear (CSIC)	[1]	Clears the IECCPUSLPINT interrupt. The reset value is 0.	0
CPU Wake-up Interrupt Clear (CWIC)	[0]	Clears the IECCPUWUINT interrupt. The reset value is 0.	0

5.2.9 Configured CPU Frequency Register (IECCFGCPUFREQ, R, Address = 0xE110_0020)

IECCFGCPUFREQ	Bits	Description	Reset Value
Reserved	[31:24]	Reserved, read undefined, do not modify.	0
Configured CPU Frequency (CFGCPUF)	[23:0]	The configured CPU frequency in kHz.	From PMU

5.2.10 DPM Frequency Register (IECDPMFREQ, R, Address = 0xE110_0024)

IECDPMFREQ	Bits	Description	Reset Value
Reserved	[31:24]	Reserved, read undefined, do not modify.	0
DPM Frequency (DPMF)	[23:0]	The DPM frequency in kHz.	From PMU

5.2.11 Configuration Fractional Index Map00 Register (IECCFGDCGIDXMAP00, R, Address = 0xE110_0040)

IECCFGDCGIDXMAP00	Bits	Description	Reset Value
IECCFGDCGIDXMAP00	[31:0]	State of IECCFGDCGIDXMAP [31:0]	From PMU

5.2.12 Configuration Fractional Index Map32 Register (IECCFGDCGIDXMAP32, R, Address = 0xE110_0044)

IECCFGDCGIDXMAP32	Bits	Description	Reset Value
IECCFGDCGIDXMAP32	[31:0]	State of IECCFGDCGIDXMAP [63:32]	From PMU

5.2.13 Configuration Fractional Index Map32 Register (IECCFGDCGIDXMAP64, R, Address = 0xE110_0048)

IECCFGDCGIDXMAP64	Bits	Description	Reset Value
IECCFGDCGIDXMAP64	[31:0]	State of IECCFGDCGIDXMAP [95:64]	From PMU

5.2.14 Configuration DVC Index Map Register (IECCFRDVCIDXMAP, R, Address = 0xE110_004C)

IECCFGDVCIDXMAP	Bits	Description	Reset Value
	[31:24]	Reserved, read undefined, do not modify.	0
IECCFGDVCIDXMAP	[23:0]	State of IECCFGDVCIDXMAP [23:0]	From PMU

5.2.15 Configuration Performance Map Register0 (IECCFGDCGPERFMAP0, R, Address = 0xE110_0060)

IECCFGDCGPERFMAP0	Bits	Description	Reset Value
IECCFGDCGPERFMAP0	[31:0]	State of IECCFGDCGPERFMAP [31:0]	From PMU

5.2.16 Configuration Performance Map Register4 (IECCFGDCGPERFMAP4, R, Address = 0xE110_0064)

IECCFGDCGPERFMAP4	Bits	Description	Reset Value
IECCFGDCGPERFMAP4	[31:0]	State of IECCFGDCGPERFMAP [63:32]	From PMU

5.2.17 DPM Command Register (IECDPMCR, R/W, Address = 0xE110_0100)

IECDPMCR	Bits	Description	Reset Value
Reserved	[31:12]	Reserved, read undefined, do not modify.	0
DPMCH3CMD	[11:8]	DPM Channel 3 command.	0
DPMCH2CMD	[7:4]	DPM Channel 2 command.	0
DPMCH1CMD	[3:0]	DPM Channel 1 command.	0

DPMCHxCMD	Bits	Description	Reset Value
b'0000	Freeze	The channel is frozen and stops accumulating. This is also the reset value.	0
b'0001	Reset	The channel is reset to zero.	0
b'0010	Accumulate	The channel starts accumulating.	0

5.2.18 DPM Channel Rate Registers (IECDPM2RATE, R/W, Address = 0xE110_0108)

IECDPM2RATE	Bits	Description	Reset Value
	[31:8]	Reserved, read undefined, do not modify.	0
IECDPM2RATE	[7:0]	The fractional rate that DPM channel 2 counts. The reset value of this register is 0x80, that is, 100%.	0x80

5.2.19 DPM Channel Rate Registers (IECDPM3RATE, R/W, Address = 0xE110_010C)

IECDPM3RATE	Bits	Description	Reset Value
	[31:8]	Reserved, read undefined, do not modify.	0
IECDPM3RATE	[7:0]	The fractional rate that DPM channel 3 counts. The reset value of this register is 0x80, that is, 100%.	0x80

5.2.20 DPM Channel Registers (IECDPM1LO, R, Address = 0xE1100_0180)

IECDPM1LO	Bits	Description	Reset Value
IECDPM1LO	[31:0]	Low 32-bit of DPM channel 1. The reset value is 0x00000000.	0x00000000.

5.2.21 DPM Channel Registers (IECDPM1HI, R, Address = 0xE1100_0184)

IECDPM1HI	Bits	Description	Reset Value
IECDPM1HI	[31:0]	High 32-bit of DPM channel 1. The reset value is 0x00000000.	0x00000000

5.2.22 DPM Channel Registers (IECDPM2LO, R, Address = 0xE1100_0188)

IECDPM2LO	Bits	Description	Reset Value
IECDPM2LO	[31:0]	Low 32-bit of DPM channel 2. The reset value is 0x00000000.	0x00000000

5.2.23 DPM Channel Registers (IECDPM2HI, R, Address = 0xE1100_018C)

IECDPM2HI	Bits	Description	Reset Value
IECDPM2HI	[31:0]	High 32-bits of DPM channel 2. The reset value is 0x00000000.	0x00000000.

5.2.24 DPM Channel Registers (IECDPM3LO, R, Address = 0xE1100_0190)

IECDPM3LO	Bits	Description	Reset Value
IECDPM3LO	[31:0]	Low 32-bits of DPM channel 3. The reset value is 0x00000000.	0x00000000.

5.2.25 DPM Channel Registers (IECDPM3HI, R, Address = 0xE1100_0194)

IECDPM3HI	Bits	Description	Reset Value
IECDPM3HI	[31:0]	High 32-bits of DPM channel 3. The reset value is 0x00000000.	0x00000000

5.2.26 Peripheral Identification Register 0 (IECPeriphID0, R, Address = 0xE110_0FE0)

IECPeriphID0	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Partnumber0	[7:0]	These bits read back as 0x50	0x50

5.2.27 Peripheral Identification Register 1 (IECPeriphID1, R, Address = 0xE110_0FE4)

IECPeriphID1	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Partnumber1	[7:0]	These bits read back as 0x07	0x07

5.2.28 Peripheral Identification Register 2 (IECPeriphID2, R, Address = 0xE110_0FE8)

IECPeriphID2	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Revision	[7:4]	These bits read back as 0x0	0x0
Designer1	[3:0]	These bits read back as 0x04	0x04

5.2.29 Peripheral Identification Register 3 (IECPeriphID3, R, Address = 0xE110_0FEC)

IECPeriphID3	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Configuration 1	[7:0]	Number of DPC levels. These bits read back as 0x08.	0x08

5.2.30 Peripheral Identification Register 4 (IECPeriphID4, R, Address = 0xE110_0FD0)

IECPeriphID4	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Reserved	[7:3]	Reserved	X
Configuration 2	[2:0]	Number of DPM channels. These bits are read back as 0x3.	0x3

5.2.31 Peripheral Identification Register 5 (IECPeriphID5, R, Address = 0xE110_0FD4)

IECPeriphID5	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Configuration 3	[7:0]	Number of DVS slots in a frame. These bits read back as 0x08.	0x08

5.2.32 Peripheral Identification Register 6 (IECPeriphID6, R, Address = 0xE110_0FD8)

IECPeriphID6	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Configuration 4	[7:0]	These bits are all reserved	Reserved

5.2.33 Peripheral Identification Register 7 (IECPeriphID7, R, Address = 0xE110_0FDC)

IECPeriphID7	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
Configuration 5	[7:0]	These bits are all reserved	Reserved

5.2.34 IEC Identification Register 0 (IECID0, R, Address = 0xE110_0FF0)

IECID0	Bits	Description	Reset Value
	[31:8]	Reserved, read undefined, do not modify.	X
IECID0	[7:0]	These bits read back as 0x0D	0x0D

5.2.35 IEC Identification Register 1 (IECID1, R, Address = 0xE110_0FF4)

IECID1	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
IECID1	[7:0]	These bits read back as 0xF0	0xF0

5.2.36 IEC Identification Register 2 (IECID2, R, Address = 0xE110_0FF8)

IECID2	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
IECID2	[7:0]	These bits read back as 0x05	0x05

5.2.37 IEC Identification Register 3 (IECID3, R, Address = 0xE110_0FFC)

IECID3	Bits	Description	Reset Value
Reserved	[31:8]	Reserved, read undefined, do not modify.	X
IECID3	[7:0]	These bits read back as 0xB1	0xB1

5.3 APC1 RELATED REGISTERS

5.3.1 PWI Command Register (APC_PWICMD, R/W, Address = 0xE100_0000)

APC_PWICMD	Bits	Description	Reset Value
PWI Slave Register Address	[7:4]	PWI slave Register address of the read and write register.	0x0
PWI Slave Command	[3:0]	PWI slave command: 4'b0000 = reset 4'b0001= authenticate 4'b0010 = register read 4'b0011= register write 4'b0100= wakeup 4'b0101= sleep 4'b0110= shutdown 4'b1001= synchronize. Unused command patterns result in a No OPeration (NOP) at the PWI interface.	0x0

5.3.2 PWI Write Data Register (APC_PWIDATAWR, R/W, Address = 0xE100_0004)

APC_PWIDATAWR	Bits	Description	Reset Value
PWI Slave Write Data	[7:0]	Data is written to the PWI slave.	0x00

5.3.3 PWI Read Data Register (APC_PWIDATARD, R, Address = 0xE100_0008)

APC_PWIDATARD	Bits	Description	Reset Value
PWI Slave Read Data	[7:0]	Data is read from the PWI slave.	0x00

5.3.4 APC Control Register (APC_CONTROL, R/W, Address = 0xE100_0010)

APC_CONTROL	Bits	Description	Reset Value
APC_HPM_AUTH_SET	[7]	HPM is set to the ring oscillator mode for a random PC used in the authentication sequence.	0
APC_PWRSV_EN	[6]	Enables the power save mode. On setting this bit: * the apc_refclk_req signal is deasserted when the apc_refclk_c clock signal is not required * the CMU can gate off the clock signal to save the power when the apc_refclk_req signal is deasserted.	0
APC_MULTICAL_EN	[5]	Enables the multiple RCC mode. Default, the APC1 is in the single RCC mode.	0
APC_HPM_EN	[4]	Enables the HPM. Default, the HPM is disabled and the PC is zero.	0
APC_PWRDN_EN	[3]	Enables the PWI sleep and wakeup command functions. Default, this feature is disabled.	0
Reserved	[2]	Read undefined. Write as zero.	0
APC_LOOP_MODE	[1]	Enable bit for the closed-loop or the open-loop mode: * defaults to the open-loop mode * setting this bit enables the closed-loop mode. The voltage scaling in the open-loop or the closed-loop mode is enabled only after setting the APC_VDD_UD bit of the APC_CONTROL Register.	0
APC_VDD_UD	[0]	Enables voltage scaling feature in the APC1: * defaults to the fixed voltage mode and the core voltage is set to the maximum value * for the closed-loop and the open-loop modes this bit must be enabled.	0

5.3.5 APC Status Register (APC_STATUS, R, Address = 0xE100_0014)

APC_STATUS	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined.	0
AUTH_DONE	[3]	Authentication procedure is completed.	0
PWI_BUSY	[2]	Bit is set on initiating a PWI command and is cleared when the command sequence is completed.	0
POWERWISE_VERIFIED	[1]	Bit is set on a successful PowerWise capable power supply authentication.	0
VDDOK	[0]	Vdd level is suitable for the current target frequency. This bit is set in the closed-loop mode.	0

5.3.6 Minimum Limit Register (APC_MINVDD_LIMIT, R/W, Address = 0xE100_0018)

APC_MINVDD_LIMIT	Bits	Description	Reset Value
Reserved	[7]	Read undefined. Write as zero.	0
Minimum core voltage	[6:0]	Minimum SoC operating core voltage.	0x00

5.3.7 VDD Check Register (APC_VDDCHK, R/W, Address = 0xE100_001C)

APC_VDDCHK	Bits	Description	Reset Value
vddchkd[3:0]	[7:4]	The upper nibble of this register holds the four LSBs of the 12-bit vddchkd counter.	0x0
vddchk[3:0]	[3:0]	Evaluation time period during the integration of the slack in the closed	0x0

5.3.8 VDD Delay Time Register (APC_VDDCHKD, R/W, Address = 0xE100_0020)

APC_VDDCHKD	Bits	Description	Reset Value
vddchkd[11:4]	[7:0]	Holds the upper eight bits of the 12-bit vddchkd counter.	0x00

5.3.9 VDD Pre-delay Select Register (APC_PREDYSEL, R/W, Address = 0xE100_0024)

APC_PREDYSEL	Bits	Description	Reset Value
Reserved	[7:3]	Read undefined. Write as zero.	0
Pre-delay	[2:0]	Selects the predelay value for the HPM.	0x7

5.3.10 APC Interrupt Mask Register (APC_IMASK, R/W, Address = 0xE100_0028)

APC_IMASK	Bits	Description	Reset Value
Reserved	[7]	Read undefined. Write as zero.	0
APB Write Discard	[6]	The APB write is discarded.	0
PWI Transaction Done	[5]	The PWI transaction is completed.	0
Error Detected in PWI	[4]	Error is detected in the PWI response frame.	0
No PWI Slave Response	[3]	No response frame detected on the PWI interface.	0
Output Voltage Clamped	[2]	The output voltage is clamped to the minimum voltage limit or to the zero voltage.	0
Low VDD Timeout	[1]	Vdd has not reached the optimum voltage in the closed-loop mode for the voltage upward slew within the hardware defined time period.	0
Undershoot Interrupt	[0]	Undershoot interrupt.	0

5.3.11 APC Interrupt Status Register (APC_ISTATUS, R, Address = 0xE100_002C)

APC_ISTATUS	Bits	Description	Reset Value
Reserved	[7]	Read undefined.	0
APB Write Discard	[6]	When the PWI command is active in the APC1, the new PWI commands issued by the host are discarded. This discarded status is reflected in this bit.	0
PWI Transaction Done	[5]	Bit is set when the APC1 completes the host issued PWI command. Software has to check this bit as well as the APC_STATUS.PWI_BUSY bit to confirm the completion of the command.	0
Error Detected in PWI	[4]	Bit is set on an error response from the PWI slave for the host issued as well as the APC1 issued PWI commands.	0
No PWI Slave Response	[3]	Bit is set for no response from the PWI slave for the host issued as well as the APC1 issued commands.	0
Output Voltage Clamped	[2]	This bit is set when the output voltage is clamped to the minimum limit or to the zero voltage.	0
Low VDD Timeout	[1]	During upward voltage slew, this bit is set in the closed-loop mode indicating that the dynamic compensator is not able to increase the voltage to the required level for the new higher performance level within the maximum time period set by the hardware.	0
Undershoot Interrupt	[0]	In the closed-loop AVS operation for a performance level change after reaching the optimum voltage the APC1 asserts an interrupt if the voltage correction continues and results in a slack error (+ve) which is more than the undershoot_limit value programmed in the APC_UNSHT_NOISE Register for nine consecutive samples.	0

5.3.12 APC Interrupt Clear Register (APC_ICLEAR, W, Address = 0xE100_0030)

APC_ICLEAR	Bits	Description	Reset Value
Reserved	[7]	Undefined. Write as zero.	0
APB Write Discard	[6]	The APB write is discarded.	0
PWI Transaction Done	[5]	The PWI transaction is completed.	0
Error Detected in PWI	[4]	Error is detected in PWI response frame.	0
No PWI Slave Response	[3]	No response frame is detected on PWI interface.	0
Output Voltage Clamped	[2]	The output voltage is clamped to minimum limit or zero voltage.	0
Low VDD Timeout	[1]	In the closed-loop mode, Vdd has not reached the target voltage in the programmed time period for the upward voltage slew.	0
Undershoot Interrupt	[0]	Undershoot interrupt.	0

5.3.13 APC Undershoot Threshold and Noise Limit Register (APC_UNSHT_NOISE, R/W, Address = 0xE100_0034)

APC_UNSHT_NOISE	Bits	Description	Reset Value
Reserved	[7:6]	Read undefined. Write as zero.	0
Noise Limit for VDDOK	[5:4]	Noise limit for the VDDOK generation due to the power supply regulation errors. Provides the acceptable integrated eHPM (+ve) below the RCC value for updating the performance level in the closed-loop mode.	0x0
Undershoot Threshold Level	[3:0]	This is the threshold level for the detection of voltage undershoot interrupt on the voltage slew. The value programmed is the amount of eHPM (+ve) allowed after reaching the optimum core voltage for the safe SoC operation.	0x0

APC_UNSHT_NOISE[5:4]	Minimum accumulated eHPM
00	0
01	4
10	16
11	31

5.3.14 Wakeup Delay Register (APC_WKUP_DLY, R/W, Address = 0xE100_0038)

APC_WKUP_DLY	Bits	Description	Reset Value
Wakeup Delay	[7:0]	Count for the wakeup delay.	0x00

5.3.15 Slack Sample Count Register (APC_SLK_SMP, R/W, Address = 0xE100_003C)

APC_SLK_SMP	Bits	Description	Reset Value
Reserved	[7:6]	Read undefined. Write as zero.	0
Slack Sample Count	[5:0]	The time period for each count in the vddchkd and the vddchk counters during the performance level change: * set to 0x1D for 2µs when theapc_refclk_c clock is 15MHZ * set to 0x3B for 2µs when the apc_refclk_c clock is 30MHZ.	0x00

5.3.16 PWI Clock Division Register (APC_CLKDIV_PWICLK, R/W, Address = 0xE100_0040)

APC_CLKDIV_PWICLK	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined. Write as zero.	0
Programmable Clock Division	[3:0]	Programmable division to theapc_refclk_c clock frequency for the PWI clock. The clock division is equal to 2 * (APC_CLKDIV_PWICLK + 1).	0x0

5.3.17 APC Overshoot Limit Register (APC_OVSHT_LMT, R/W, Address = 0xE100_0050)

APC_OVSHT_LMT	Bits	Description	Reset Value
overshoot limit	[7:0]	Overshoot limit during the voltage slew in the closed-loop mode.	0x00

5.3.18 APC Closed-loop Control Register (APC_CLP_CTRL, R/W, Address = 0xE100_0054)

APC_CLP_CTRL	Bits	Description	Reset Value
Reserved	[7:4]	Undefined. Write as zero.	0
APC_SS_GAIN_EN	[3]	Enables steady state gain term.	0
APC_UP_GAIN_EN	[2]	Enables the APC_GAIN2 term for the dynamic compensator. This gain term is selected during voltage upward slew.	0
APC_LOW_GAIN_EN	[1]	Enables the APC_GAIN4 term for the dynamic compensator. This gain term is selected when the slack or eHPM value is between +3 to -3.	0
APC_SAT_GAIN_EN	[0]	Enables the APC_GAIN3 term for the dynamic compensator. This gain term is selected when the PC value is saturated and the voltage is stepping up. This gain term has higher priority over the gain term 2.	0

5.3.19 APC Steady State Slew Rate Register (APC_SS_SRATE, R/W, Address = 0xE100_0058)

APC_SS_SRATE	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined.	0
APC_SS_SMP_RATE	[3:2]	00 Sample the eHPM every 32 **apc_refclk_c** cycles. 01 Sample the eHPM every 16 **apc_refclk_c** cycles. 10 Sample the eHPM every 8 **apc_refclk_c** cycles. 11 Sample the eHPM every **apc_refclk_c** cycle.	0x0
APC_GAIN_SEL	[1:0]	00 Gain term value of 0 in steady state mode. 01 Gain term value of 1 in steady state mode. 10 Gain term value of 2 in steady state mode. 11 Gain term value of 3 in steady state mode.	0x0

5.3.20 Integrator's Gain Registers (APC_IGAIN1, R/W, Address = 0xE100_0060)

APC_IGAIN1	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined. Write as zero.	0
Gain 1	[3:0]	Default gain term for the dynamic compensator. The programmable values for this gain term are one to ten. Rest of the values are treated as zero in the closed-loop AVS operations.	0x0

5.3.21 Integrator's Gain Registers (APC_IGAIN2, R/W, Address = 0xE100_0064)

APC_IGAIN2	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined. Write as zero.	0
Gain 2	[3:0]	Gain term for the upward voltage slew when enabled in the APC_CLP_CTRL Register. The programmable values for this gain term are one to ten. Rest of the values are treated as zero in the closed-loop AVS operations.	0x0

5.3.22 Integrator's Gain Registers (APC_IGAIN3, R/W, Address = 0xE100_0068)

APC_IGAIN3	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined. Write as zero.	0
Gain 3	[3:0]	Dynamic compensator uses this gain term for the saturated HPM output when enabled. The programmable values for this gain term are one to ten. Rest of the values are treated as zero in the closed-loop AVS operations.	0x0

5.3.23 Integrator's Gain Registers (APC_IGAIN4, R/W, Address = 0xE100_006C)

APC_IGAIN4	Bits	Description	Reset Value
Reserved	[7:4]	Read undefined. Write as zero.	0
Gain 4	[3:0]	Selected by the dynamic compensator when enabled in the APC_CL_CTRL Register for the low slack values. The programmable values for this gain term are one to ten. Rest of the values are treated as zero in the closed-loop AVS operations.	0x0

5.3.24 Voltage Information Registers

APC1 has two types of voltage information registers. Ones are for closed-loop control, and the others are for open-loop control. Registers for closed-loop control give delay information, while registers for open-loop control give direct voltage information. There is a register containing the retention voltage level for the performance level zero.

- **Calibration Code Registers (APC_PL1_CALCODE, R/W, Address = 0xE100_0080)**
- **Calibration Code Registers (APC_PL2_CALCODE, R/W, Address = 0xE100_0084)**
- **Calibration Code Registers (APC_PL3_CALCODE, R/W, Address = 0xE100_0088)**
- **Calibration Code Registers (APC_PL4_CALCODE, R/W, Address = 0xE100_008C)**
- **Calibration Code Registers (APC_PL5_CALCODE, R/W, Address = 0xE100_0090)**
- **Calibration Code Registers (APC_PL6_CALCODE, R/W, Address = 0xE100_0094)**
- **Calibration Code Registers (APC_PL7_CALCODE, R/W, Address = 0xE100_0098)**
- **Calibration Code Registers (APC_PL8_CALCODE, R/W, Address = 0xE100_009C)**

The Calibration Code Registers are eight, 5-bit registers. Their names are **APC_PL1_CALCODE ~ APC_PL8_CALCODE**. They give delay information target for closed-loop operation.

APC_PL*_CALCODE	Bits	Description	Reset Value
Reserved	[7:5]	Read undefined. Write as zero.	X
Reference Calibrated Code 1	[4:0]	The RCC for performance level *	0x1F

- **Open-loop VDD Core Registers (APC_PL1_COREVDD, R/W, Address = 0xE100_00A0)**
- **Open-loop VDD Core Registers (APC_PL2_COREVDD, R/W, Address = 0xE100_00A4)**
- **Open-loop VDD Core Registers (APC_PL3_COREVDD, R/W, Address = 0xE100_00A8)**
- **Open-loop VDD Core Registers (APC_PL4_COREVDD, R/W, Address = 0xE100_00AC)**
- **Open-loop VDD Core Registers (APC_PL5_COREVDD, R/W, Address = 0xE100_00B0)**
- **Open-loop VDD Core Registers (APC_PL6_COREVDD, R/W, Address = 0xE100_00B4)**
- **Open-loop VDD Core Registers (APC_PL7_COREVDD, R/W, Address = 0xE100_00B8)**
- **Open-loop VDD Core Registers (APC_PL8_COREVDD, R/W, Address = 0xE100_00BC)**

The Open-loop VDD Core Registers are eight, 7-bit registers. Their names are **APC_PL1_COREVDD** ~ **APC_PL8_COREVDD**. They give direct voltage information for open-loop operation.

APC_PL*_COREVDD	Bits	Description	Reset Value
Reserved	[7]	Read undefined. Write as zero.	X
OL_VDD1	[6:0]	The voltage value for the performance level * in the open-loop mode.	0x7F

5.3.25 Retention VDD Registers (APC_RET_VDD, R/W, Address = 0xE100_00C0)

APC_RET_VDD	Bits	Description	Reset Value
Reserved	[7]	Read undefined. Write as zero.	0
Retention VDD	[6:0]	The retention voltage level for performance level zero.	0x00

5.3.26 Debug Performance Registers (APC_DBG_DLYCODE, R, Address = 0xE100_00E0)

APC_DBG_DLYCODE	Bits	Description	Reset Value
Reserved	[7:5]	Read undefined.	0
Performance Code	[4:0]	The PC of the HPM.	0x00

5.3.27 Revision Number Registers (APC_REV, R, Address = 0xE100_00FC)

APC_REV	Bits	Description	Reset Value
Revision Number	[7:0]	Holds the APC1 revision number.	0x01

3.1 BUS CONFIGURATION

1 BUS CONFIGURATION OVERVIEW

Figure 3.1-1 shows the bus architecture of S5PC100. This chapter describes the feature & programmer's model of AXI interconnects used in this product.

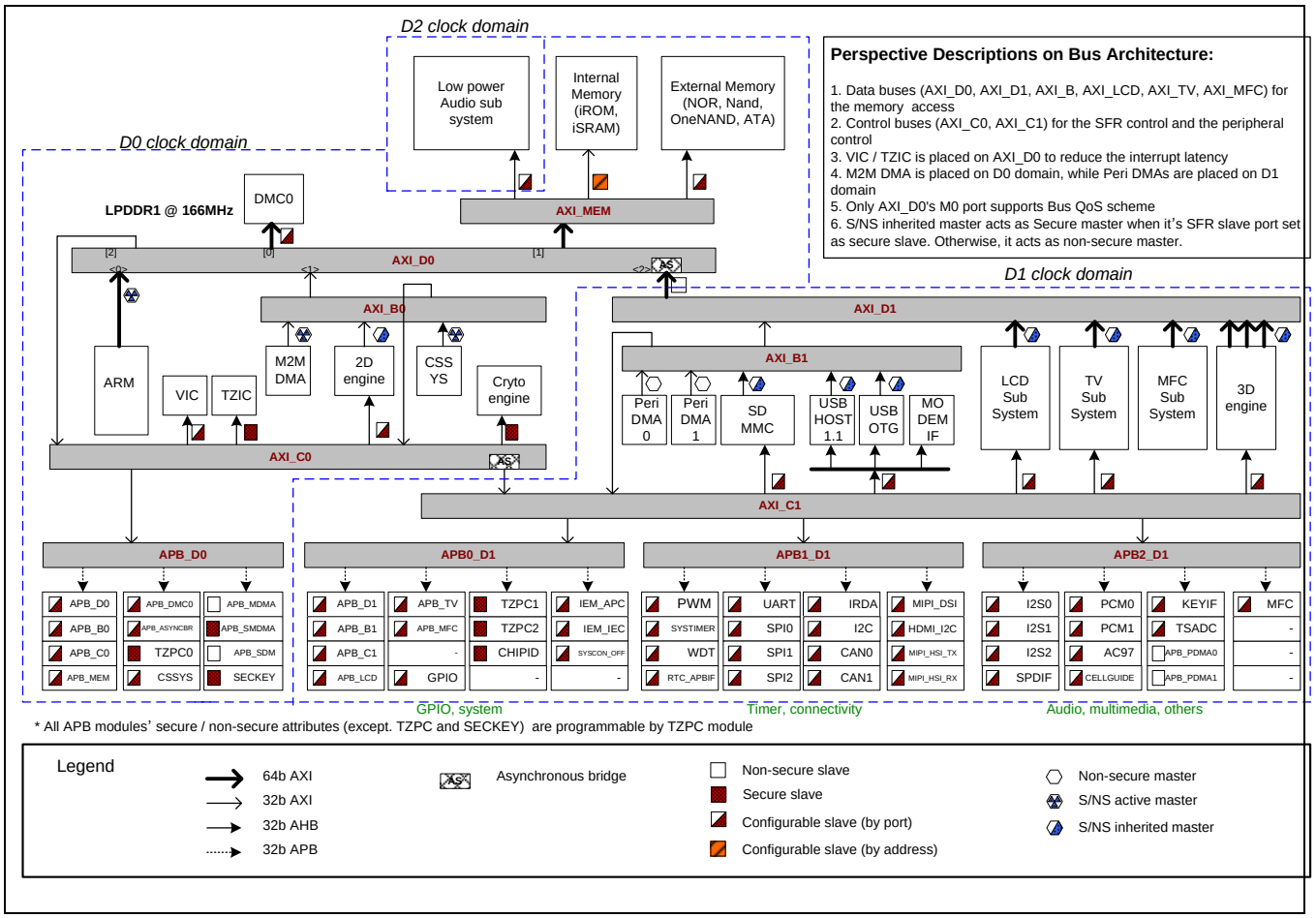


Figure 3.1-1 S5PC100 Bus Architecture

In S5PC100, the QoS scheme will be used for restricting the transactions of the background operation when there is heavy real time traffic. The masters attached to AXI_B0 are classified as background operation which requires a high bandwidth but does not require to be real time traffic. The QoS scheme is meaningless to the master interface which is connected to the slave whose combined acceptance capability is only "1". Therefore, the QoS feature is only applied to the M0 port of AXI_D0 interconnect in S5PC100.

The fixed round robin arbitration was proposed to allocate the proper bandwidth to each slave interface. Therefore, this arbitration is beneficial to the multimedia master bus such as AXI_D1 which requires correct bandwidth allocation. The fixed round robin arbitration scheme is only applied to AXI_D1.

2 PROGRAMMABLE FEATURES OF AXI INTERCONNECT

2.1 PROGRAMMABLE QUALITY OF SERVICE

The QoS scheme works by tracking the number of outstanding transactions, and when a specified number is reached, it only permits transactions from particular, specified masters.

The QoS scheme only provides support for slaves that have a combined acceptance capability, such as the PrimeCell Dynamic Memory Controller (PL340).

The QoS scheme has no effect until the AXI bus matrix calculates that, at a particular MI, there are a number of outstanding transactions equal to the value stored in the QoS tidemark register. It then accepts transactions only from slave ports specified in the QoS access control register. This restriction remains until the number of outstanding transactions is again less than the value stored in the QoS tidemark register.

Figure 3.1-2 shows the implementation for an interconnect that supports two masters and one slaves.

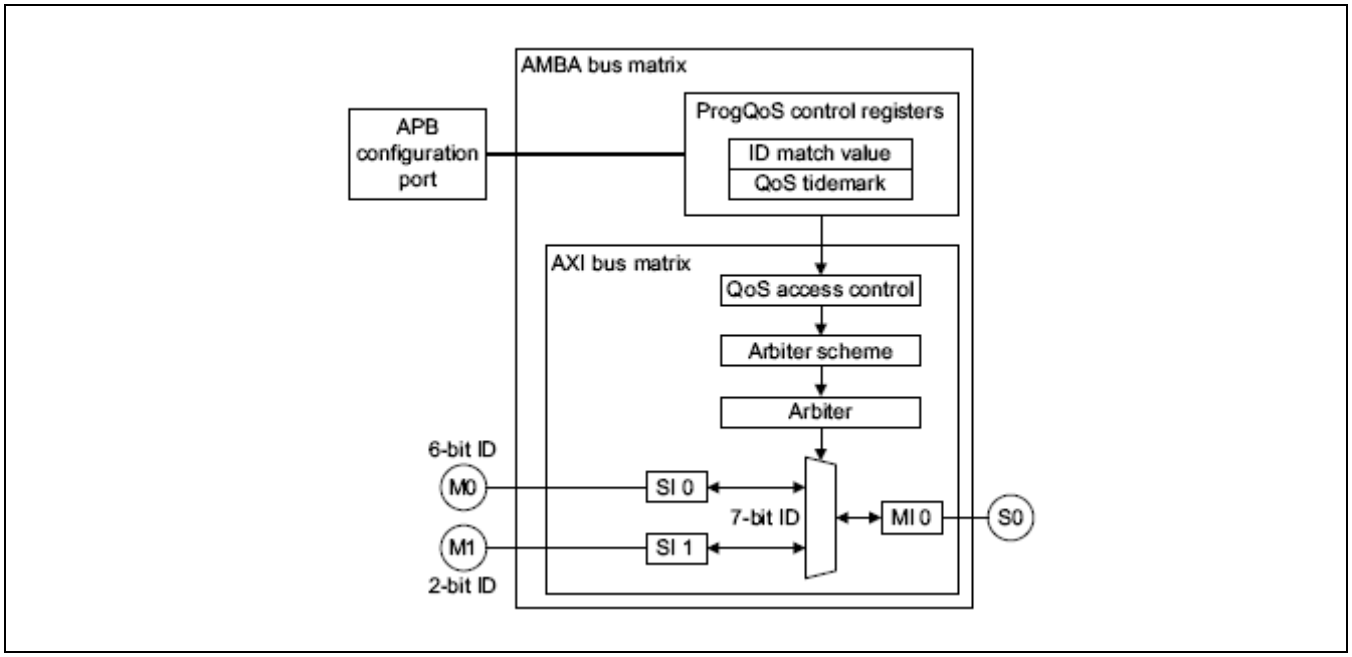


Figure 3.1-2 Example of ProgQoS Control Registers for 2-1 Interconnect

NOTES

1. When there is only one master, the QoS logic is removed as an optimization. However, the APB configuration interface enables you to program QoS parameters, but they have no effect.

2. The QoS feature is only applied to M0 port of AXI_D0 interconnect.

2.2 ARBITRATION SCHEME

In the AXI interconnect, you can configure each MI separately to have an arbitration scheme that is either:

- a non-programmable RR scheme
- a programmable LRG scheme.

The AW and AR channels have separate arbiters and can be programmed, if applicable, and interrogated separately through the APB programming interface, but both AW and AR channels are configured identically. Because the AW and AR channels are arbitrated separately, an MI can permit simultaneous read and write transactions from different SI's.

The arbitration mechanism registers the arbitration decision for use in the subsequent cycle. An arbitration decision taken in the current cycle does not affect the current cycle.

If no SI's are active, the arbiter adopts default arbitration, that is, the highest priority SI. If this occurs and then the highest priority interface becomes active in the same cycle as, or before any other SI, then this does not constitute a grant to an active SI and the arbitration scheme does not change its state.

If a QoS provision is enabled and active, only a subset of SI's is permitted to win arbitration, and it cannot be guaranteed that the default arbitration is among these. In these circumstances, no transaction is permitted to use the default arbitration, and arbitration must occur when there is an active SI.

2.3 ROUND-ROBIN (RR) SCHEME

In the RR scheme, you can select the following design time:

- The number of slots that are used
- The SI to which they are allocated
- The order of slots.

There must be at least one slot per connected SI and there can be up to 32 slots. By allocating multiple slots for a SI, you can allocate access to the slave, on average, in proportion to the number of slots. If the slots are appropriately ordered, this can also reduce the maximum time before a grant is guaranteed. The SI associated with a slot can be interrogated from the APB programming interface, but it cannot be changed.

Whenever arbitration is granted to an active SI, the slots are rotated so that the slot currently in the highest priority position becomes the lowest, and all other slots move to a higher priority but maintain their relative order, as shown in Figure 3.1-3. This means that if an SI is the highest priority active SI, but is not the highest priority interface, then it continues to win the arbitration until it becomes the highest priority interface, and then the lowest priority interface subsequently.

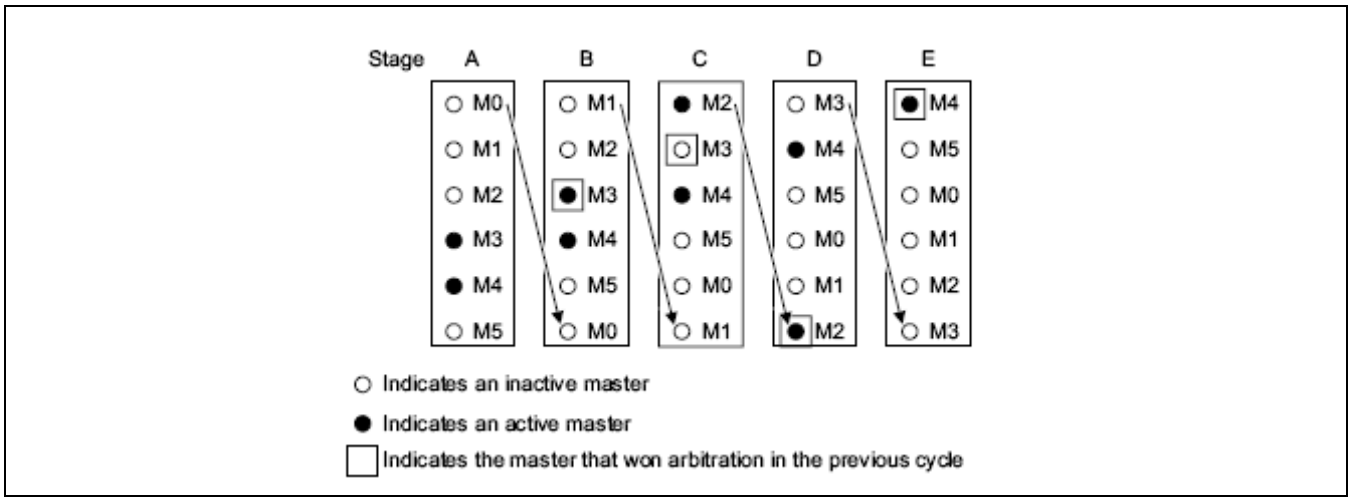


Figure 3.1-3 Example Operation of RR Rbitration Scheme

Because the arbitration value is registered, the arbitration decision made in this cycle is used in the next cycle. This means that if the SI that currently holds the arbitration is still the highest priority active SI in this cycle, wins the arbitration again regardless of whether or not it is active in the next cycle as shown by the status of M3 in stages A, B, and C of Figure 3.1-3.

NOTE

The fixed round robin arbitration scheme is only applied to M0 port of AXI_D1.

2.3.1 Least Recently Granted (LRG) Scheme

In the LRG scheme, each connected SI has a single slot associated with it, but each interface also has a priority value. This priority value, whose post-reset value can be configured at design time and programmed or interrogated through the APB programming interface, can make the arbiter behave as:

- A pure LRG scheme
- A fixed priority encoder
- A combination of the two.

All masters with the same priority form a priority group. As a result of arbitration, a master can move within its priority group but cannot leave its group, and no new masters can join the group.

Arbitration is granted to the highest priority group from which a member is trying to win access, and within that group, to the highest master at that time. When master wins arbitration, it is relegated to the bottom of its group to ensure that is cannot prevent other masters in its group from accessing the slave.

If you configure all master priorities to different levels, the arbiter implements a fixed priority scheme. This occurs because in this case, each master is in a group of its own, and therefore, masters maintain their ordering.

If all master priorities are the same, then an LRG scheme is implemented. The reason that it behaves as an LRG is because the process of relegating the master that was last granted access, to the bottom of its group, results in the masters being ordered from the LRG master at the top, to the *Most Recently Granted* (MRG) at the bottom.

The LRG and fixed priority modes concurrently exist when the master priority value registers are programmed with a combination of identical and unique values. You can mix priority groups that contain one member with priority groups that contain more than one member in an arbitrary manner. The arbiter places no restriction on the number of groups or their membership.

Figure 3.1-4 shows the movement of masters within their priority groups.

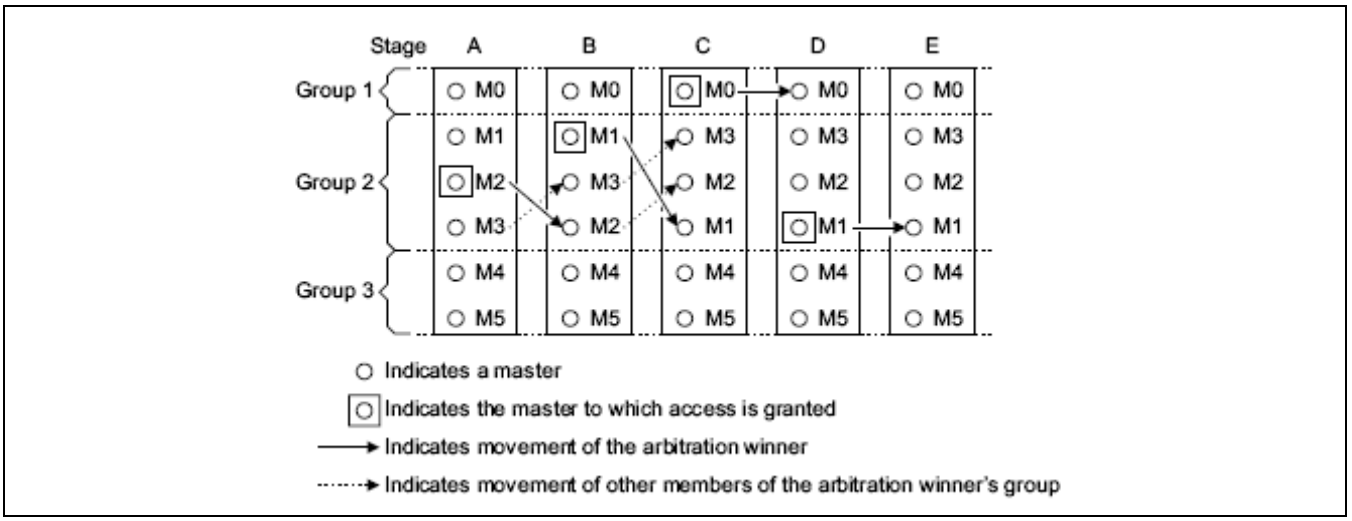


Figure 3.1-4 Example Operation of LRG Arbitration Scheme

3 REGISTER DESCRIPTION

3.1 AXI_D0 REGISTER MAPS

Register	Address	R/W	Description	Reset Value
QOSTM_M0	0xE300_0400	R/W	QoS Tidemark for MI 0	0x00000000
QOSAC_M0	0xE300_0404	R/W	Qos Access control for MI 0	0x00000000
R_AR_M0	0xE300_0408	R/W	AR Channel Arbitration value for MI 0	0x00000000
W_AR_M0	0xE300_040C	R/W	AW Channel Arbitration value for MI 0	0x00000000
R_AR_M1	0xE300_0428	R/W	AR Channel Arbitration value for MI 1	0x00000000
W_AR_M1	0xE300_042C	R/W	AW Channel Arbitration value for MI 1	0x00000000
R_AR_M2	0xE300_0448	R/W	AR Channel Arbitration value for MI 2	0x00000000
W_AR_M2	0xE300_044C	R/W	AW Channel Arbitration value for MI 2	0x00000000
CFGR0	0xE300_0FC0	R	Primecell Configuration register 0	0x00000003
CFGR1	0xE300_0FC4	R	Primecell Configuration register 1	0x00000003
CFGR2	0xE300_0FC8	R	Primecell Configuration register 0	0x00000000
CFGR3	0xE300_0FCC	R	Primecell Configuration register 1	0x00000000
PERPHID0	0xE300_0FE0	R	Primecell Peripheral register 0	0x01
PERPHID1	0xE300_0FE4	R	Primecell Peripheral register 1	0x13
PERPHID2	0xE300_0FE8	R	Primecell Peripheral register 2	0x24
PERPHID3	0xE300_0FEC	R	Primecell Peripheral register 3	0x00
PRIMEID0	0xE300_0FF0	R	Primcell ID register 0	0x0D
PRIMEID1	0xE300_0FF4	R	Primcell ID register 1	0xF0
PRIMEID2	0xE300_0FF8	R	Primcell ID register 2	0x05
PRIMEID3	0xE300_0FFC	R	Primcell ID register 3	0xB1

3.2 AXI_D0 REGISTER DESCRIPTION

3.2.1 QoS Tidemark Registers for M0 (QOSTM_M0, R/W, Address = 0xE300_0400)

QOSTM_M0	Bit	Description	Reset Value
Reserved -	[31:6]	Reserved	-
QoS Tidemark	[5:0]	The number of outstanding transactions that are permitted before the QoS scheme becomes active. If a value is written to this register that is larger than the combined acceptance capability of the attached slave, then the QoS scheme never becomes active for this MI. If a value of 0 is written to this register, then the QoS scheme is turned off for this MI. This behavior ensures that it is impossible to block all transactions completely by accidental mis-programming.	0x0

3.2.2 QoS Access Control Registers for M0 (QOSAC_M0, R/W, Address = 0xE300_0404)

QOSAC_M0	Bit	Description	Reset Value
Reserved -	[31:3]	Reserved	-
QoS Access Control	[2:0]	A 1 in any bit of this register indicates that the SI corresponding to the bit position is permitted to use the reserved slots of the connected combined acceptance capability of the slaves. Changes to these values occur on the first possible arbitration time after they are written.	0x0

3.2.3 Channel Arbitration Registers (M0/M1/M2, W, Address = 0xE300_0408, 0xE300_040C, 0xE300_0428, 0xE300_042C, 0xE300_0448, 0xE300_044C)

- Writes data case

M0 / M1 / M2	Bit	Description	Reset Value
Slot	[31:24]	The slot for which the data is to be written	-
Reserved	[23:16]	Reserved	-
Priority	[15:8]	The priority value	-
SI Number	[7:0]	The SI number for which the priority value applies	-

NOTE: It is important to ensure that a value is written only to the slot that already contains the priority value for the SI whose priority you want to modify - writes are ignored if this is not the case. This behavior is required because the arbitration system must maintain exactly one slot for each SI for correct operation.

You cannot program the RR scheme; therefore writes are completed but are ignored.

- Reads data case (W)
  Slot number writing before read

M0 / M1 / M2	Bit	Description	Reset Value
Special Format	[31:8]	0xFF0000 indicates its slot number writing function	-
Slot	[7:0]	The slot for which the data is to be read	-

- Read arbitration data (R)

M0 / M1 / M2	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	-
Arbitration Data	[7:0]	The format of the returned data depends on the arbitration scheme. The RR scheme returns the SI number that occupies that slot in the LSB of the read data. The LRG scheme returns the priority and SI number in the same positions as they occupy in write data.	0x0

NOTES

All other AXI interconntects are not need to be programmed by user.

3.2 CORESIGHT

1 CORESIGHT SYSTEM OVERVIEW

1.1 ABOUT CORESIGHT SYSTEMS GENERALS

CoreSight systems provide all the infrastructure you require to debug, monitor, and optimize the performance of a complete System on Chip (SoC) design.

There are historically three main ways of debugging an ARM processor based SoC:

- Conventional JTAG debug. This is invasive debug with the core halted using:
  √ breakpoints and watchpoints to halt the core on specific activity
  √ a debug connection to examine and modify registers and memory and provide single-step execution.

- Conventional monitor debug. This is invasive debug with the core running using a debug monitor that resides in memory.

- Trace. This is non-invasive debug with the core running at full speed using:
  √ collection of information on instruction execution and data transfers
  √ delivery off-chip in real-time
  √ tools to merge data with source code on a development workstation for later analysis.

1.2 FEATURES

1.2.1 Debug Access

You gain debug access in CoreSight systems through the Debug Access Port (DAP) that provides:

- real-time access to physical memory without halting the core and without any target resident code
- debug control and access to all status registers

The same mechanism provides fast access for downloading code at the start of the debug session. This is faster than the traditional JTAG mechanism that uses the ARM core to write data to memory. You can still use the ARM core to write data to virtual memory and to ease migration when the debugger does not support this approach.

Figure 3.2-1 shows an example system with debug components and a DAP in a SoC design.

The DAP provides the following advantages for multi-core SoC designs:

- There is no requirement to run at the lowest common speed. A slow or powered down component has no effect on access to other components. This means that power management has minimal impact on debug.
- The speed of access is not affected by the number of devices in the system. You have direct access to individual devices.
- You can add third party debug components with the *Advanced Microcontroller Bus Architecture* (AMBA) debug bus interface, AMBA 3 *Advanced Peripheral Bus* (APB), that provides internal and external access to the component.
- More than one core can control debug functionality, rather than restricting this to the core being debugged. One core can debug another. In particular this enables a multi-core SoC when used as a single core platform to have complex on-chip debug and analysis features. You could use this, for example, during application development.

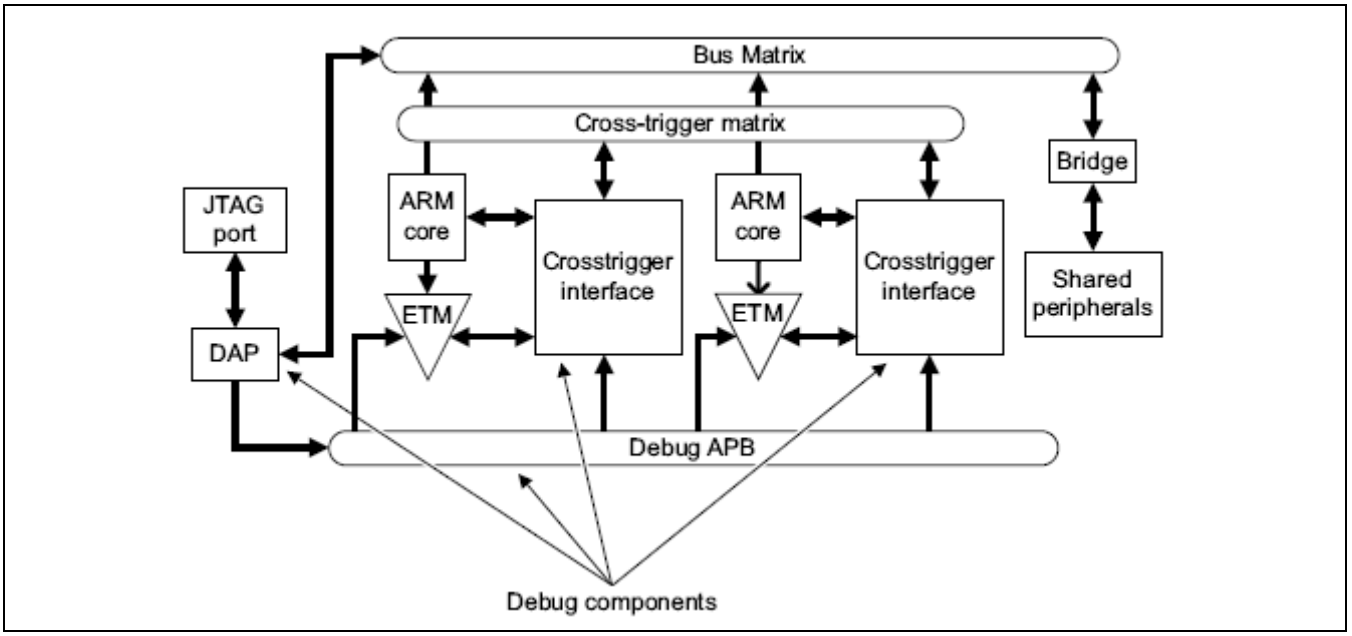


Figure 3.2-1 DAP Connections Inside a SoC

1.2.2 Cross Triggering

The Embedded Cross Trigger (ECT), comprising of the Cross Trigger Interface (CTI) and Cross Trigger Matrix (CTM), provides a standard interconnect mechanism to pass debug or profiling events around the SoC.

The ECT provides you with a standard mechanism to connect different signal types. A set of standard triggers for cores are predefined and you can add triggers for third party cores.

The ECT enables tool developers to supply a standard control dialog so that software programmers can connect trigger events.

1.2.3 Trace

The CoreSight Design Kit provides components that support a standard infrastructure for the capture and transmission of trace data, combination of multiple data streams by funneling together, and then output of data to a trace port or storing in an on-chip buffer.

1.3 CORESIGHT SYSTEM IN S5PC100

S5PC100 is single processor system with CortexA8 core. And it's main bus system is based on AMBA3 AXI interconnects. And it does not support Serial Wire debug port protocol.

The configuration of debugging system is as shown in Figure 3.2-2.

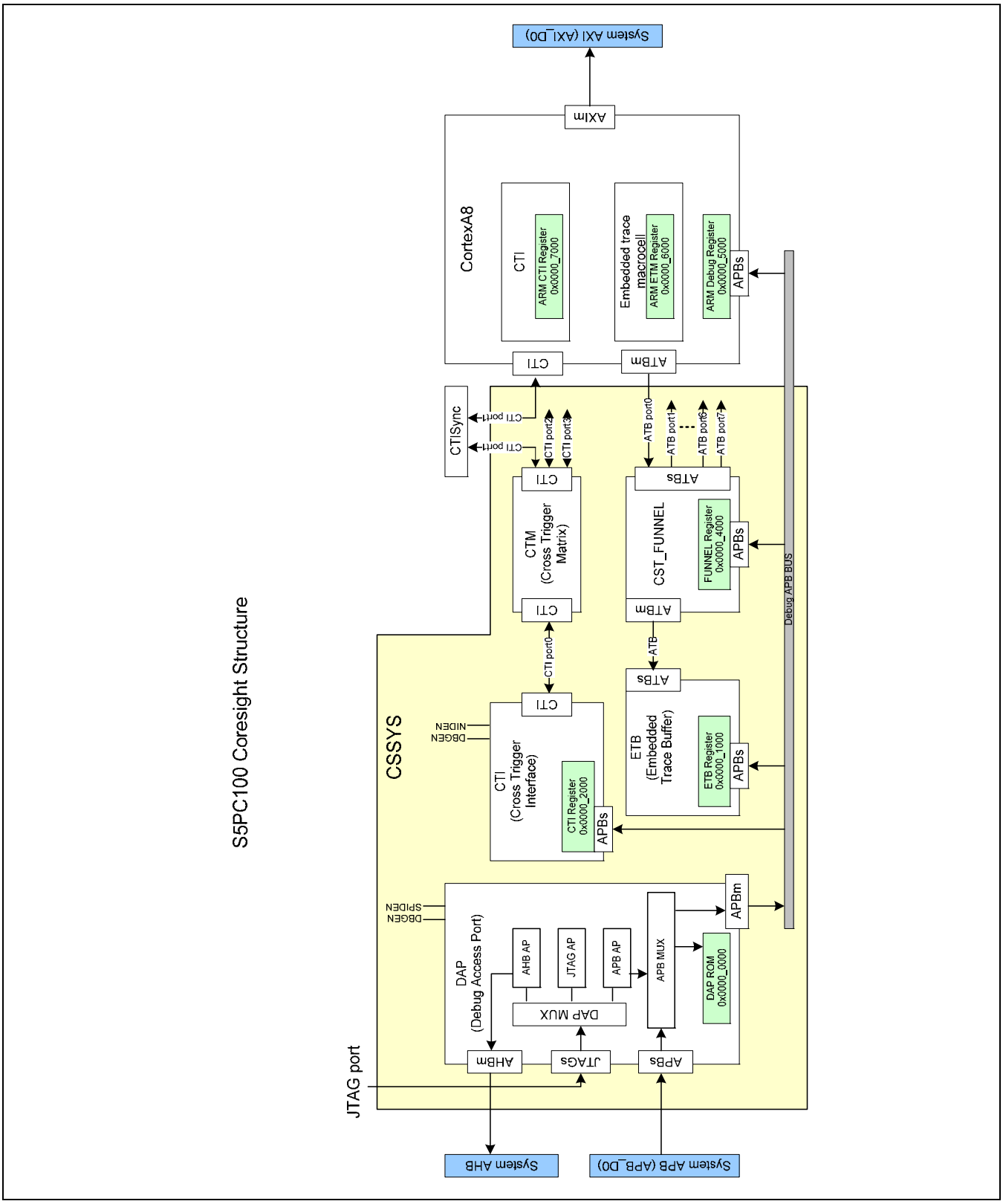


Figure 3.2-2 S5PC100 Coresight Structure

NOTE

Although Coresight's registers can be accessed through system APB bus as well as JTAG port, the address map of those registers are observed differently. While the memory map for JTAG port is same as shown in Figure 3.2-2, the memory map for system view is same as the memory map for JTAG port + system register offset. The debugger register map of S5PC100 is summarized in Figure 3.2-3.

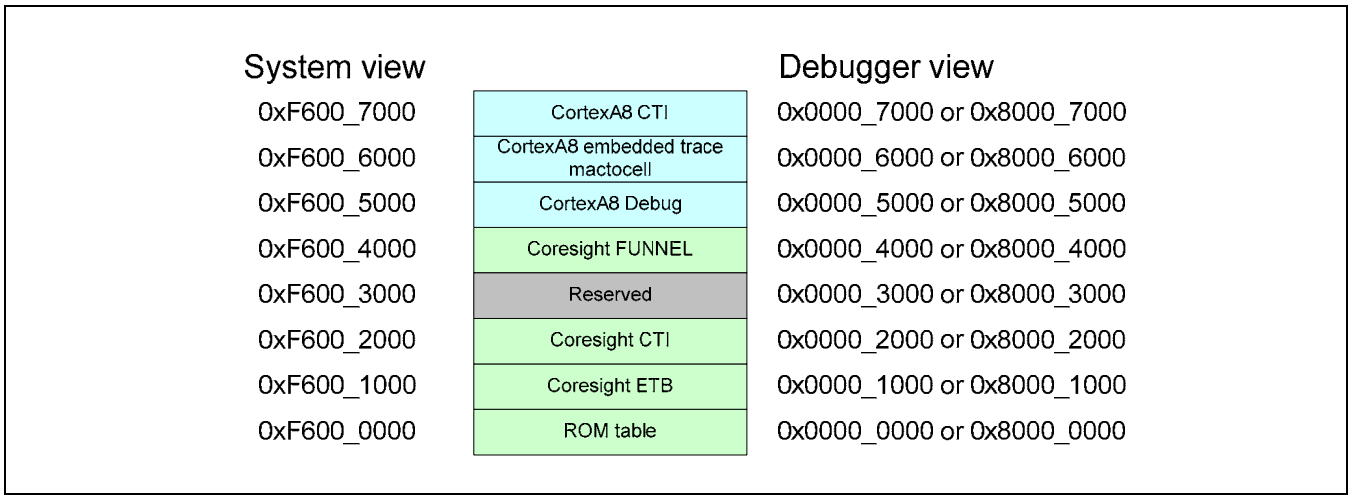


Figure 3.2-3 Debugger Register Map of S5PC100

The more detail information of debugger register will be handled in programmers model part.

1.3.1 Authentication for Secure JTAG Operation

S5PC100 supports Secure JTAG by using authentication signal of cortexA8 & coresight system.

And the Secure JTAG mode can be set by programming 5th e-fuse block.

Though total number of bits of 5th e-fuse rom is 96bit, only 82bits are used for SecureJTAG function.

- [79:0]: Secure JTAG hash key
- [80]: Secure JTAG lock on – 0: non-protection, 1: protected by secure JTAG
- [81]: Secure access type after authentication – 0: secure access, 1: non-secure access

Before authentication, the debugger should access to internal authentication module in security sub system. In S5PC100, the AHB AP of coresight provides path for internal module to debugger as depicted in Figure 3.2-4.

If Secure JTAG lock on bit is programmed as "1", the authentication signals such as DBGEN, NIDEN, SPIDEN, SPNIDEN are all "0" before passing authentication

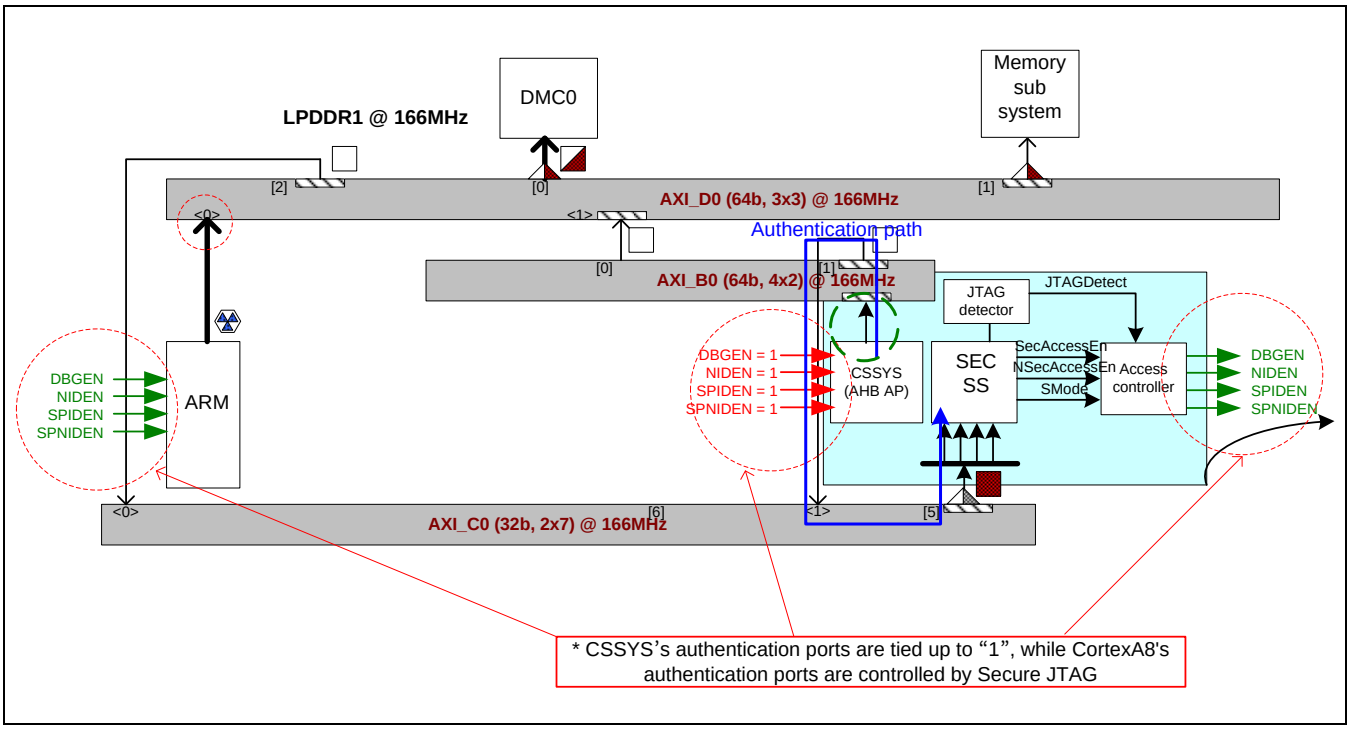


Figure 3.2-4 Secure JTAG Scheme in S5PC100

At that time, because the AHB AP's address ports are muxed with tied value for the address of authentication module, the debugger can access the authentication module in security sub system through the AHB AP of coresight.

By writing the passwords in predefined sequence, the authentication can be done. After authentication, the authentication signals are selectively asserted as defined in Table 3.2-1.

Table 3.2-1 Authentication Signal Rule

Mode	Secure JTAG lock on	Secure Access type	DBGEN	NIDEN	SPIDEN	SPNIDEN
JTAG & non-authenticated	1	dont care	1'b0	1'b0	1'b0	1'b0
JTAG & authenticated as non-secure non-invasive	1	1	1'b0	1'b1	1'b0	1'b0
JTAG & authenticated as non-secure invasive	1	1	1'b1	1'b1	1'b0	1'b0
JTAG & authenticated as secure non-invasive	1	0	1'b0	1'b1	1'b1	1'b1
JTAG & authenticated as secure invasive	1	0	1'b1	1'b1	1'b1	1'b1
non-protected mode non-secure invasive	0	1	1'b1	1'b1	1'b0	1'b0
non-protected mode secure invasive	0	0	1'b1	1'b1	1'b1	1'b1

And the AHB AP's address mux is controlled by authentication signals as well shown in Figure 3.2-5.

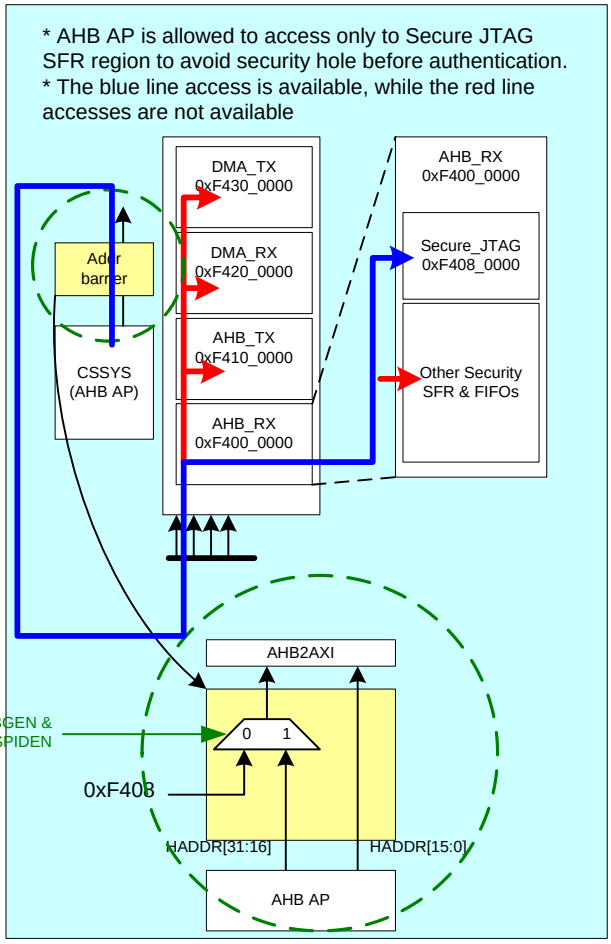


Figure 3.2-5 AHB AP Mux Control by Authentication Signal

The authentication sequence script and the hash key generation program will be provided to the customer.

Further information about Secure JTAG is handled in chapter 11.2 Security Sub system.

2 DEBUG ACCESS PORT

2.1 ABOUT DEBUG ACCESS PORT

The Debug Access Port (DAP) is a implementation of an ARM Debug Interface version 5 (ADIv5) comprising a number of components supplied in a single configuration. All the supplied components fit into the various architectural components for Debug Ports (DPs), which are used to access the DAP from an external debugger and Access Ports (APs), to access on-chip system resources.

The debug port and access ports together are referred to as the DAP.

The DAP provides real-time access for the debugger without halting the core to:

- AMBA system memory and peripheral registers
- All debug configuration registers.

The DAP also provides debugger access to JTAG scan chains of system components, for example non-CoreSight compliant processors. Figure 3.2-6 shows the top-level view of the functional blocks of the DAP.

The DAP enables debug access to the complete SoC using a number of master ports.

Access to the CoreSight Debug Advanced Peripheral Bus (APB) is enabled through the APB Access Port (APB-AP) and APB Multiplexer (APB-MUX), and system access through the Advanced High-performance Bus Access Port (AHB-AP).

The DAP comprises the following interface blocks:

- External debug access using the JTAG Debug Port.
  √ External JTAG access using the JTAG Debug Port (JTAG-DP).
- System access using:
  √ AHB-AP
  √ APB-AP
  √ JTAG-AP
  √ DAPBUS exported interface.
- An APB multiplexor enables system access to CoreSight components connected to the Debug APB.
- The ROM table provides a list of memory locations of CoreSight components connected to the Debug APB. This is visible from both tools and system access.

There are three access ports supplied in the DAP, and it is possible to connect a fourth access port externally.

The supplied access ports within this release are:

- AHB-AP for connection to the main system bus
- APB-AP to enable direct connection to the dedicated Debug Bus
- JTAG-AP to control up to eight scan chains.

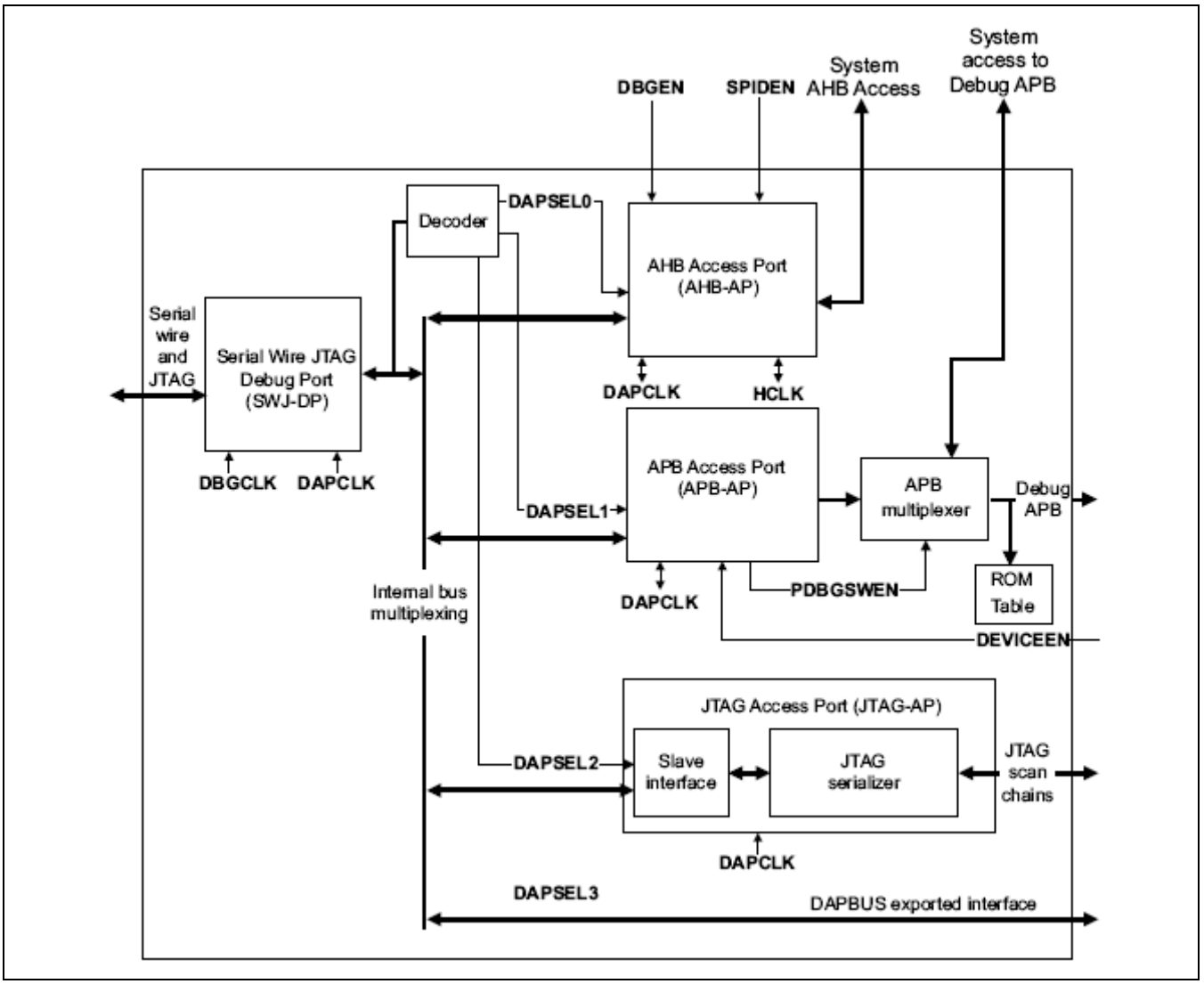


Figure 3.2-6 Structure of the Coresight DAP Components

3 ETB

3.1 ABOUT THE ETB

The ETB provides on-chip storage of trace data using 32-bit RAM. Figure 3.2-7 shows the main ETB blocks. The ETB accepts trace data from CoreSight trace source components through an AMBA Trace Bus (ATB).

The ETB contains the following blocks:

- Formatter - Inserts source ID signals into the data packet stream so that trace data can be re-associated with its trace source after the data is read back out of the ETB.
- Control - Control registers for trace capture and flushing.
- APB interface - Read, write, and data pointers provide access to ETB registers. In addition, the APB interface supports wait states through the use of a PREADYDBG signal output by the ETB. The APB interface is synchronous to the ATB domain.
- Register bank - Contains the management, control, and status registers for triggers, flushing behavior, and external control.
- Trace RAM interface - Controls reads and writes to the Trace RAM.
- Memory BIST interface - Provides test access to the Trace RAM.

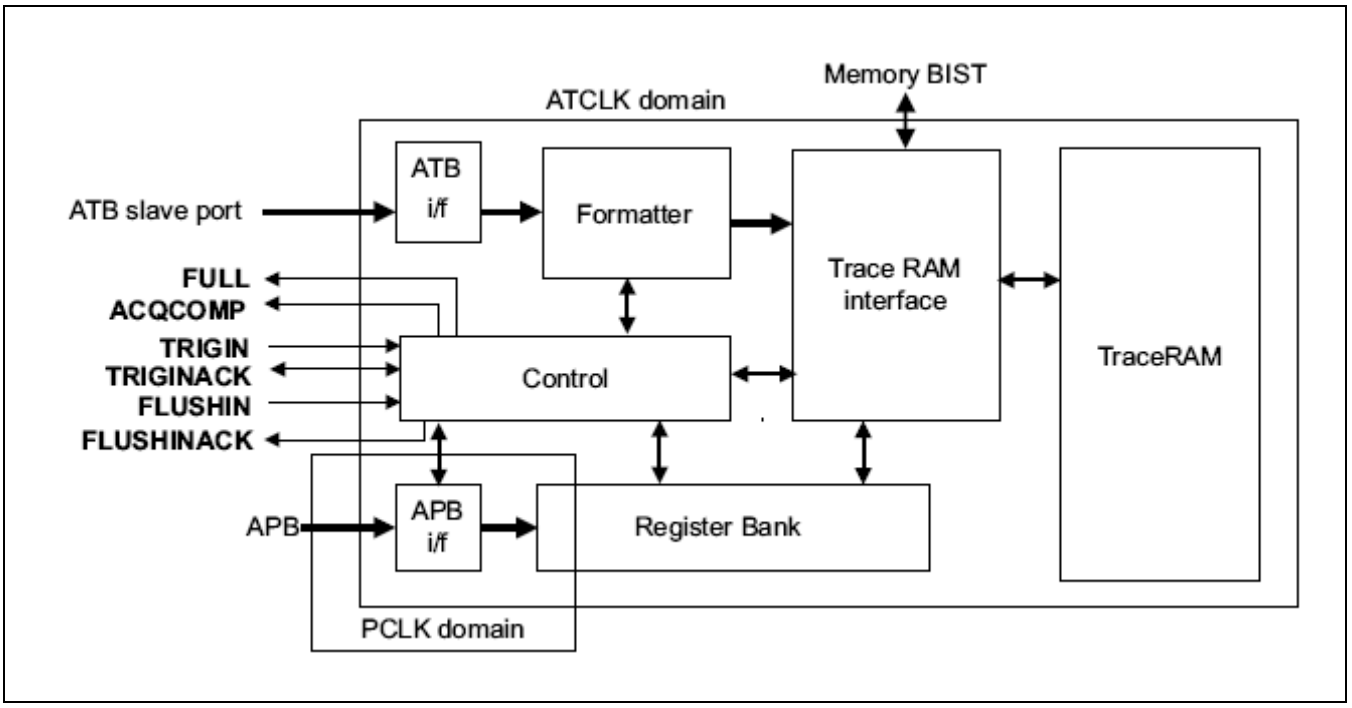


Figure 3.2-7 ETB Block Diagram

4 ECT (CTI + CTM)

4.1 ABOUT THE ECT

The ECT provides an interface to the debug system as shown in Figure 3.2-8. This enables ARM subsystems to interact, that is cross trigger, with each other. The debug system enables debug support for multiple cores, together with cross triggering between the cores and their respective internal embedded trace macrocells.

The main function of the ECT (CTI and CTM) is to pass debug events from one processor to another. For example, the ECT can communicate debug state information from one core to another, so that program execution on both processors can be stopped at the same time if required.

- Cross Trigger Interface (CTI)
  The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the ECT as channel events. When the CTI receives a channel event it maps this onto a trigger output. This enables subsystems to cross trigger with each other. The receiving and transmitting of triggers is performed through the trigger interface.
- Cross Trigger Matrix (CTM)
  This block controls the distribution of channel events. It provides Channel Interfaces (CIs) for connection to either CTIs or CTMs. This enables multiple CTIs to be linked together.

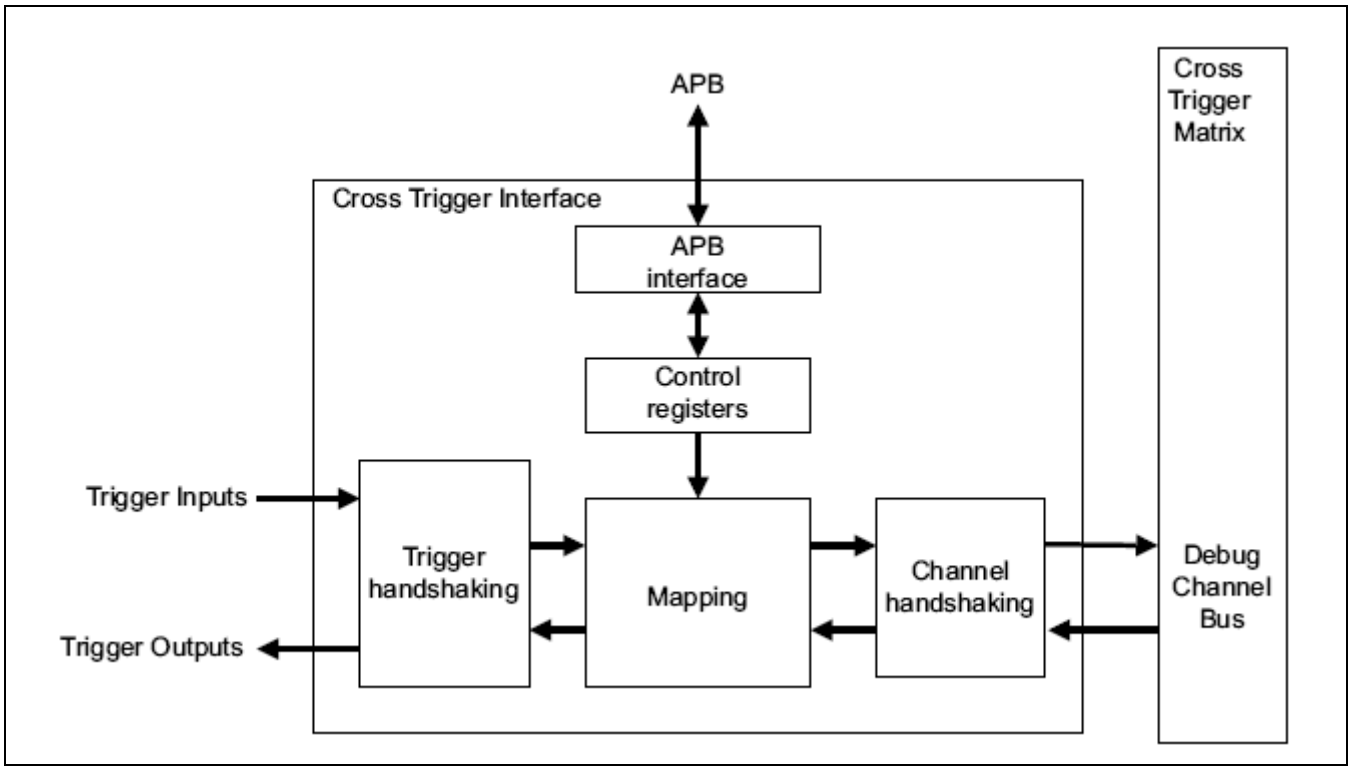


Figure 3.2-8 Coresight CTI and CTM Block Diagram

5 I/O DESCRIPTION

Signal	I/O	Description	Pad	Type
nTRST	Input	TAP reset	XjTRSTn	Dedicated
TCK	Input	TAP clock	XjTCK	Dedicated
TMS	Input	TAP test mode selection	XjTMS	Dedicated
TDI	Input	TAP data in	XjTDI	Dedicated
TDO	Output	TAP data out	XjTDO	Dedicated

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

3.3 ACCESS CONTROLLER (TZPC)

1 OVERVIEW

The TrustZone Protection Controller (TZPC), TZProtCtrl, is an AMBA-compliant, tested, and licensed by ARM Limited. The TZPC provides a software interface to the protection bits in a secure system in a TrustZone design. It provides system flexibility that enables to configure different areas of memory as secure or non-secure.

The S5PC100 comprises of three TZPC.

1.1 FEATURES

1. Protection bits: This enables you to program maximum 32 areas of memory as secure or non-secure
2. Secure region bits: This enables you to split an area of internal RAM into both secure and non-secure regions
3. The Access Controller includes AMBA APB system interface

1.2 BLOCK DIAGRAM

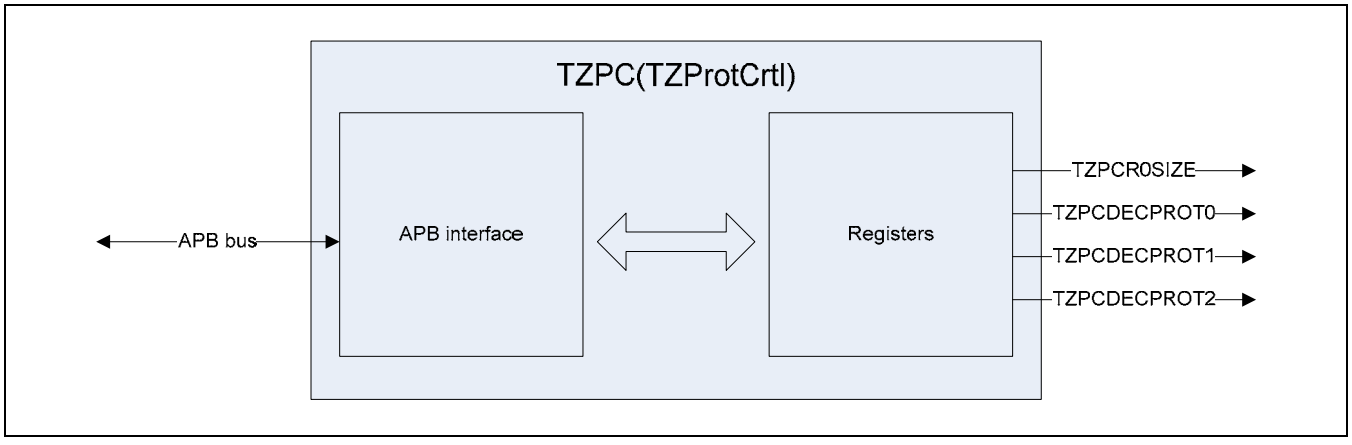


Figure 3.3-1 Block Diagram

2 FUNCTIONAL DESCRIPTION

The TZPC provides a software interface to set up memory areas as secure or non-secure.

The two ways to set up memory area as secure or non-secure is as follows:

- Programmable protection bits that can be allocated to memory area as determined by the external decoder.
- Programmable region size value for use by an AXI TrustZone Memory Adapter (TZMA). You can use this to split the RAM into two regions:
  - One secure
  - One non-secure.

This enables the best use of memory and other system resources. It is assumed that the specific secure and non-secure requirements for an application are determined during:

- BOOT-UP
- OS or Secure kernel port development work.

This means that the secure and non-secure memory partitioning is not expected to change dynamically during normal software operation because it is fixed at compile time and is only configured once during system boot-up. Ensure that this boot-up is always made in secure-state to guarantee full security protection.

3 TZPC CONFIGURATION

The following table(Table 3.3-1) shows the configurable region.

TZPC0 is configurable of TZPCDECPROT and TZPCR0SIZE.

TZPC1 and TZPC2 are configurable of TPCDECPROT.

Table 3.3-1 TZPC Table

Register	Bit	TZPC0	TZPC1	TZPC2
		Module Name	Module Name	Module Name
TZPCDECPROT0	[0]	ASYNC_APBIF	GPIO	I2S0
	[1]	G2D/CFCON	IEM_APC	I2S1
	[2]	CSSYS	IEM_IEC	I2S2
	[3]	MEMSYS(Data)	SYSCON_OFF	SPDIF
	[4]	AUDIO_SS	PWM	PCM0
	[5]	DMC	SYSTIMER	PCM1
	[6]	VIC	WDT	AC97
	[7]	MEMSYS(SFR)	RTC_APBIF	CELLGUIDE
TZPCDECPROT1	[0]	-	UART	KEYIF
	[1]	-	SPI0	TSADC
	[2]	-	SPI1	SDMMC
	[3]	-	SPI2	USB
	[4]	DMC(SFR)	IrDA	LCD
	[5]	-	I2C	TV
	[6]	-	CAN0	MFC
	[7]	-	CAN1	G3D
TZPCDECPROT2	[0]	-	MIPI_DSI	-
	[1]	-	HDMI_I2C	-
	[2]	-	MIPI_HSI_TX	-
	[3]	-	MIPI_HSI_RX	-
	[4]	-	-	-
	[5]	-	-	-
	[6]	-	-	-
	[7]	-	-	-
TZPCR0SIZE		IntMEM	-	-

The following table(Table 3.3-2) shows the fixed region with secure.

Table 3.3-2 Secure Region

Module Name	Secure
TZPC0	Secure
TZPC1	Secure
TZPC2	Secure
CHIPID	Secure
SECKEY	Secure
MEM_SDMA	Secure
TZIC	Secure
SECSS	Secure

If non-secure master accesses to secure slave area, DECERR occurs.

Table 3.3-3 TZPC Transfer Attribite

Master Attribute	Transfer Attribute	Slave/Area Attribite	Response
Secure Master	Secure Transfer	Secure Slave / Area	OK
	Secure Transfer	Non-Secure Slave / Area	OK
	Non-Secure Transfer	Secure Slave / Area	DECERR
	Non-Secure Transfer	Non-Secure Slave / Area	OK

4 REGISTER DISCRIPTION

4.1 SUMMARY OF REGISTERS

Register	Address	R/W	Description	Reset Value
TZPC0				
TZPCR0SIZE	0xE380_0000	R/W	Secure RAM Region Size Register	0x000003FF
TZPCDECPROT0Stat	0xE380_0800	R	Decode Protection 0 Status Register	0x00000000
TZPCDECPROT0Set	0xE380_0804	W	Decode Protection 0 Set Register	–
TZPCDECPROT0Clr	0xE380_0808	W	Decode Protection 0 Clear Register	–
TZPCDECPROT1Stat	0xE380_080C	R	Decode Protection 1 Status Register	0x00000000
TZPCDECPROT1Set	0xE380_0810	W	Decode Protection 1 Set Register	–
TZPCDECPROT1Clr	0xE380_0814	W	Decode Protection 1 Clear Register	–
TZPCDECPROT2Stat	0xE380_0818	R	Not used	0x00000000
TZPCDECPROT2Set	0xE380_081C	W	Not used	–
TZPCDECPROT2Clr	0xE380_0820	W	Not used	–
TZPCPERIPHID0	0xE380_0FE0	R	TZPC Peripheral Identification Register 0	0x00000070
TZPCPERIPHID1	0xE380_0FE4	R	TZPC Peripheral Identification Register 1	0x00000018
TZPCPERIPHID2	0xE380_0FE8	R	TZPC Peripheral Identification Register 2	0x00000000
TZPCPCELLID0	0xE380_0FF0	R	TZPC Identification Register 0	0x0000000D
TZPCPCELLID1	0xE380_0FF4	R	TZPC Identification Register 1	0x000000F0
TZPCPCELLID2	0xE380_0FF8	R	TZPC Identification Register 2	0x00000005
TZPC1				
TZPCR0SIZE	0xE280_0000	R/W	Not used	0x00000200
TZPCDECPROT0Stat	0xE280_0800	R	Decode Protection 0 Status Register	0x00000000
TZPCDECPROT0Set	0xE280_0804	W	Decode Protection 0 Set Register	–
TZPCDECPROT0Clr	0xE280_0808	W	Decode Protection 0 Clear Register	–
TZPCDECPROT1Stat	0xE280_080C	R	Decode Protection 1 Status Register	0x00000000
TZPCDECPROT1Set	0xE280_0810	W	Decode Protection 1 Set Register	–
TZPCDECPROT1Clr	0xE280_0814	W	Decode Protection 1 Clear Register	–
TZPCDECPROT2Stat	0xE280_0818	R	Decode Protection 2 Status Register	0x00000000
TZPCDECPROT2Set	0xE280_081C	W	Decode Protection 2 Set Register	–
TZPCDECPROT2Clr	0xE280_0820	W	Decode Protection 2 Clear Register	–
TZPCPERIPHID0	0xE280_0FE0	R	TZPC Peripheral Identification Register 0	0x00000070
TZPCPERIPHID1	0xE280_0FE4	R	TZPC Peripheral Identification Register 1	0x00000018
TZPCPERIPHID2	0xE280_0FE8	R	TZPC Peripheral Identification Register 2	0x00000000
TZPCPCELLID0	0xE280_0FF0	R	TZPC Identification Register 0	0x0000000D
TZPCPCELLID1	0xE280_0FF4	R	TZPC Identification Register 1	0x000000F0
TZPCPCELLID2	0xE280_0FF8	R	TZPC Identification Register 2	0x00000005

TZPC2				
TZPCR0SIZE	0xE290_0000	R/W	Not used	0x00000200
TZPCDECPROT0Stat	0xE290_0800	R	Decode Protection 0 Status Register	0x00000000
TZPCDECPROT0Set	0xE290_0804	W	Decode Protection 0 Set Register	–
TZPCDECPROT0Clr	0xE290_0808	W	Decode Protection 0 Clear Register	–
TZPCDECPROT1Stat	0xE290_080C	R	Decode Protection 1 Status Register	0x00000000
TZPCDECPROT1Set	0xE290_0810	W	Decode Protection 1 Set Register	–
TZPCDECPROT1Clr	0xE290_0814	W	Decode Protection 1 Clear Register	–
TZPCDECPROT2Stat	0xE290_0818	R	Not used	0x00000000
TZPCDECPROT2Set	0xE290_081C	W	Not used	–
TZPCDECPROT2Clr	0xE290_0820	W	Not used	–
TZPCPERIPHID0	0xE290_0FE0	R	TZPC Peripheral Identification Register 0	0x00000070
TZPCPERIPHID1	0xE290_0FE4	R	TZPC Peripheral Identification Register 1	0x00000018
TZPCPERIPHID2	0xE290_0FE8	R	TZPC Peripheral Identification Register 2	0x00000000
TZPCPCELLID0	0xE290_0FF0	R	TZPC Identification Register 0	0x0000000D
TZPCPCELLID1	0xE290_0FF4	R	TZPC Identification Register 1	0x000000F0
TZPCPCELLID2	0xE290_0FF8	R	TZPC Identification Register 2	0x00000005

4.2 SECURE RAM REGION SIZE REGISTER

- TZPCR0SIZE(TZPC0), RW, Address = 0xE380_0000

TZPCR0SIZE	Bit	Description	Reset Value
Reserved	[31:10]	Read undefined. Write as zero.	0
R0Size	[9:0]	Secure RAM region size in 4KB steps. 0x00000000 = no secure region 0x00000001 = 4KB secure region 0x00000002 = 8KB secure region … 0x000001FF = 2044KB secure region 0x00000200 or above sets the entire RAM to secure regardless of size	0x3FF

4.3 DECODE PROTECTION 0-3 STATUS REGISTERS

- TZPCDECPROTxSTAT(TZPC0), R, Address = 0xE380_0800, 0xE380_080C
- TZPCDECPROTxSTAT(TZPC1), R, Address = 0xE280_0800, 0xE280_080C, 0xE280_0818
- TZPCDECPROTxSTAT(TZPC2), R, Address = 0xE290_0800, 0xE290_080C

TXPCDECPROTxStat	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined.	0
DECPROTxStat	[7:0]	Show the status of the decode protection output: 0 = Decode region corresponding to the bit is secure 1 = Decode region corresponding to the bit is non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	0x000

4.4 DECODE PROTECTION 0-2 SET REGISTERS

- TZPCDECPROTxSet(TZPC0), W, Address = 0xE380_0804, 0xE380_0810
- TZPCDECPROTxSet(TZPC1), W, Address = 0xE280_0804, 0xE280_0810, 0xE280_081C
- TZPCDECPROTxSet(TZPC2), W, Address = 0xE290_0804, 0xE290_0810

TXPCDECPROTxSet	Bit	Description	Reset Value
Reserved	[31:8]	Write as zero.	–
DECPROTxSet	[7:0]	Sets the corresponding decode protection output: 0 = No effect 1 = Set decode region to non-secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

4.5 DECODE PROTECTION 0-2 CLEAR REGISTERS

- TZPCDECPROTxClr(TZPC0), W, Address = 0xE380_0808, 0xE380_081C
- TZPCDECPROTxClr(TZPC1), W, Address = 0xE280_0808, 0xE280_081C, 0xE280_0820
- TZPCDECPROTxClr(TZPC2), W, Address = 0xE290_0808, 0xE290_081C

TXPCDECPROTxClr	Bit	Description	Reset Value
Reserved	[31:8]	Write as zero.	–
DECPROTxClr	[7:0]	Clears the corresponding decode protection output: 0 = No effect 1 = Set decode region to secure There is one bit of the register for each protection output, eight outputs are implemented as standard.	–

4.6 TZPC PERIPHERAL IDENTIFICATION REGISTER 0 (TZPCPERIPHID0, R, ADDRESS = 0XE380_0FE0, 0XE280_0FE0, 0XE290_0FE0)

TZPCPERIPHID0	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined	0
Partnumber0	[7:0]	These bits read back as 0x70	0x70

4.7 TZPC PERIPHERAL IDENTIFICATION REGISTER 1 (TZPCPERIPHID1, R, ADDRESS = 0XE380_0FE4, 0XE280_0FE4, 0XE290_0FE4)

TZPCPERIPHID1	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined	0
Designer0	[7:4]	These bits read back as 0x1	0x1
Partnumber1	[3:0]	These bits read back as 0x8	0x8

4.8 TZPC PERIPHERAL IDENTIFICATION REGISTER 2 (TZPCPERIPHID2, R, ADDRESS = 0XE380_0FE8, 0XE280_0FE8, 0XE290_0FE8)

TZPCPERIPHID2	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined	0
Revision	[7:4]	These bits read back as the revision number which can be 0-15	0x0
Designer1	[3:0]	These bits read back as 0x4	0x4

4.9 IDENTIFICATION REGISTER 0 (TZPCPCELLID0, R, ADDRESS = 0XE380_0FF0, 0XE280_0FF0, 0XE290_0FF0)

TZPCPCELLID0	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined	0
TZPCPCELLID0	[7:0]	These bits read back as 0x0D	0x0D

4.10 IDENTIFICATION REGISTER 1 (TZPCPCELLID1, R, ADDRESS = 0XE380_0FF4, 0XE280_0FF4, 0XE290_0FF4)

TZPCPCELLID1	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined	0
TZPCPCELLID1	[7:0]	These bits read back as 0xF0	0xF0

4.11 IDENTIFICATION REGISTER 2 (TZPCPCELLID2, R, ADDRESS = 0XE380_0FF8, 0XE280_0FF8, 0XE290_0FF8)

TZPCPCELLID2	Bit	Description	Reset Value
Reserved	[31:8]	Read undefined	0
TZPCPCELLID2	[7:0]	These bits read back as 0x05	0x05

3.4 ASYNC BRIDGE

1 ASYNC BRIDGE OVERVIEW

1.1 ABOUT THE AXI ASYNCHRONOUS BRIDGE

The AXI asynchronous bridge (AsyncAxi) enables two AXI clock domains to communicate. Figure 3.4-1 shows AsyncAxi with data being transferred between two AXI clock domains.

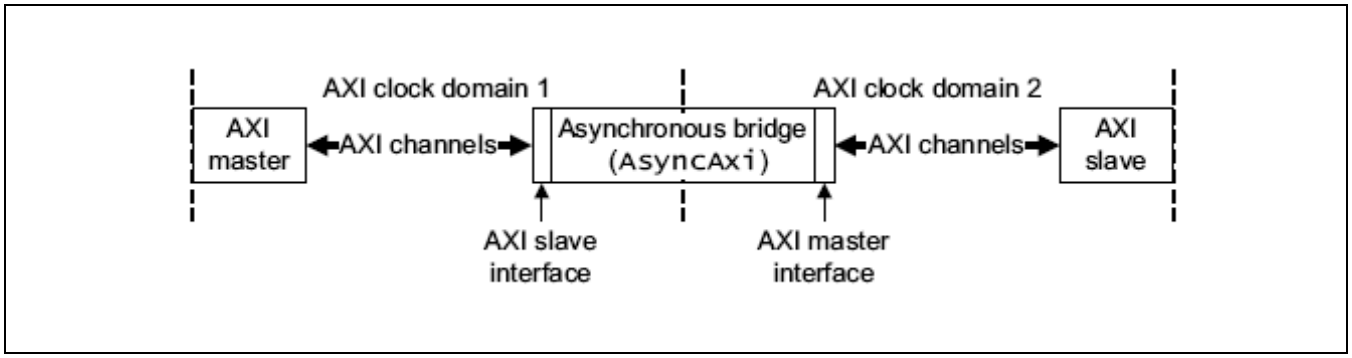


Figure 3.4-1 Asynchronous Bridge Block Diagram

The bridge provides buffered synchronization of the AXI channels:

AW Write Address Channel.

W Write Data Channel.

B Write Response Channel.

AR Read Address Channel.

R Read Data Channel.

The major features of the bridge include:

- Single independent AXI master and AXI slave interfaces
- All AXI channels are buffered independently
- Configurable FIFO buffer depth for each AXI channel
- Dynamic Synchronous Bypass mode. (S5PC100 does not use Bypass mode)

1.2 BRIEF DESCRIPTION OF FUNCTION

At the top level, the bridge comprises:

- Five FIFOs, one for each AXI channel
- Logic to manage transition (to and from Synchronous Bypass mode).

There is no requirement for the bridge to examine any part of the data in an AXI channel. Therefore it can be treated as unstructured and consider each channel as valid, ready, and payload data.

Figure 3.4-2 shows a block diagram of the major internal component blocks. The clock, reset, and scan pins are omitted for clarity.

Each channel FIFO is divided into two halves that correspond to the source of the clock for the components in them.

ACLKS Connects to the following:

- Write address channel FIFO write half
- Write data channel FIFO write half
- Read address channel FIFO write half
- Write response channel FIFO read half
- Read data channel FIFO read half.

ACLKM Connects to the following:

- Write address channel FIFO read half
- Write data channel FIFO read half
- Read address channel FIFO read half
- Write response channel FIFO write half
- Read data channel FIFO write half.

Two FIFOs operate in the opposite direction to the other three but all five behave in the same way with respect to their read and write halves.

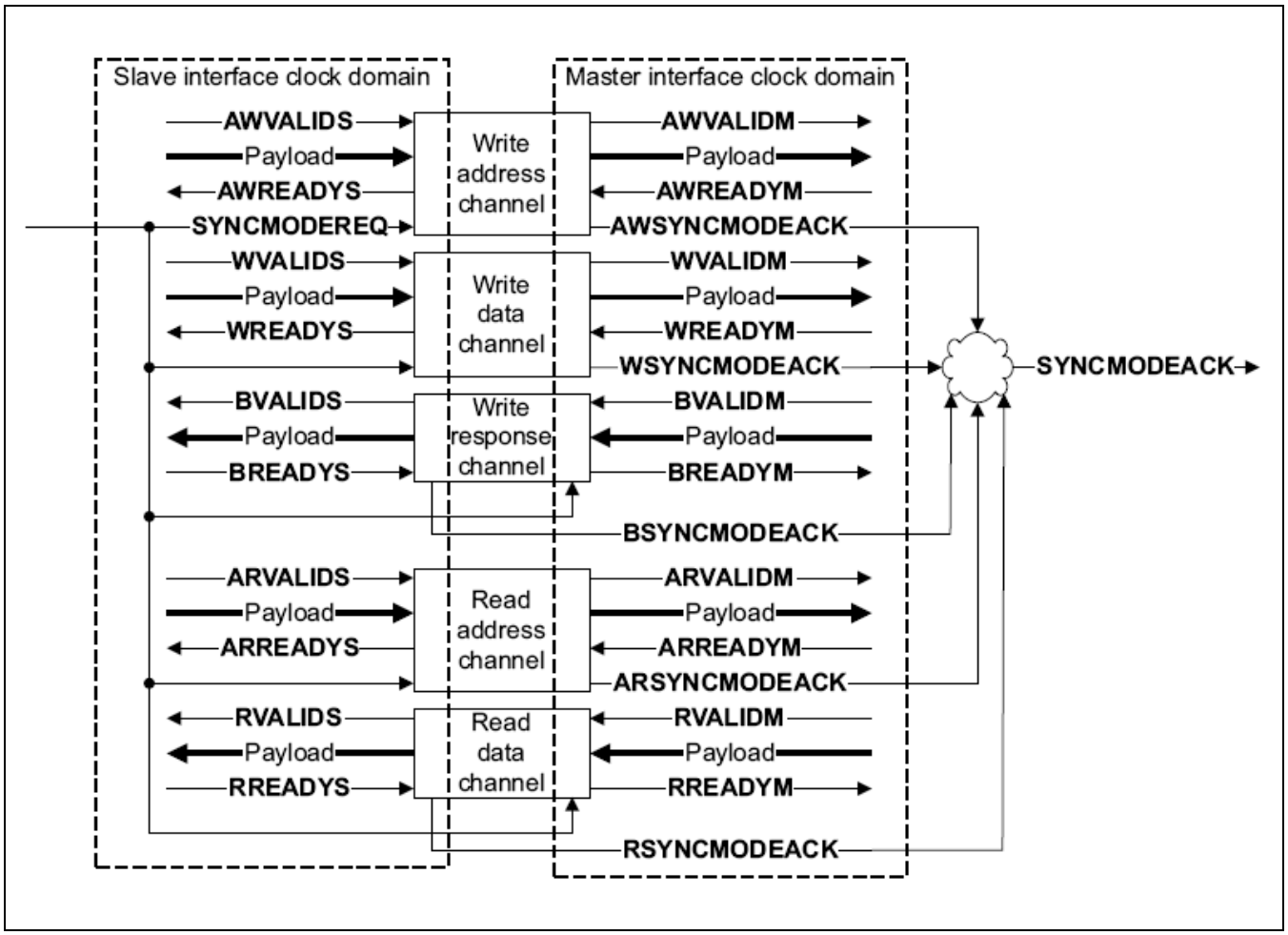


Figure 3.4-2 Asynchronous Bridge Components

1.3 OPERATION OF ASYNCHRONOUS BRIDGE

1.3.1 Normal Operation

Almost all functionality is provided by the five logically identical FIFOs and this section describes their operation. Figure 3.4-3 shows the internal structure of a FIFO.

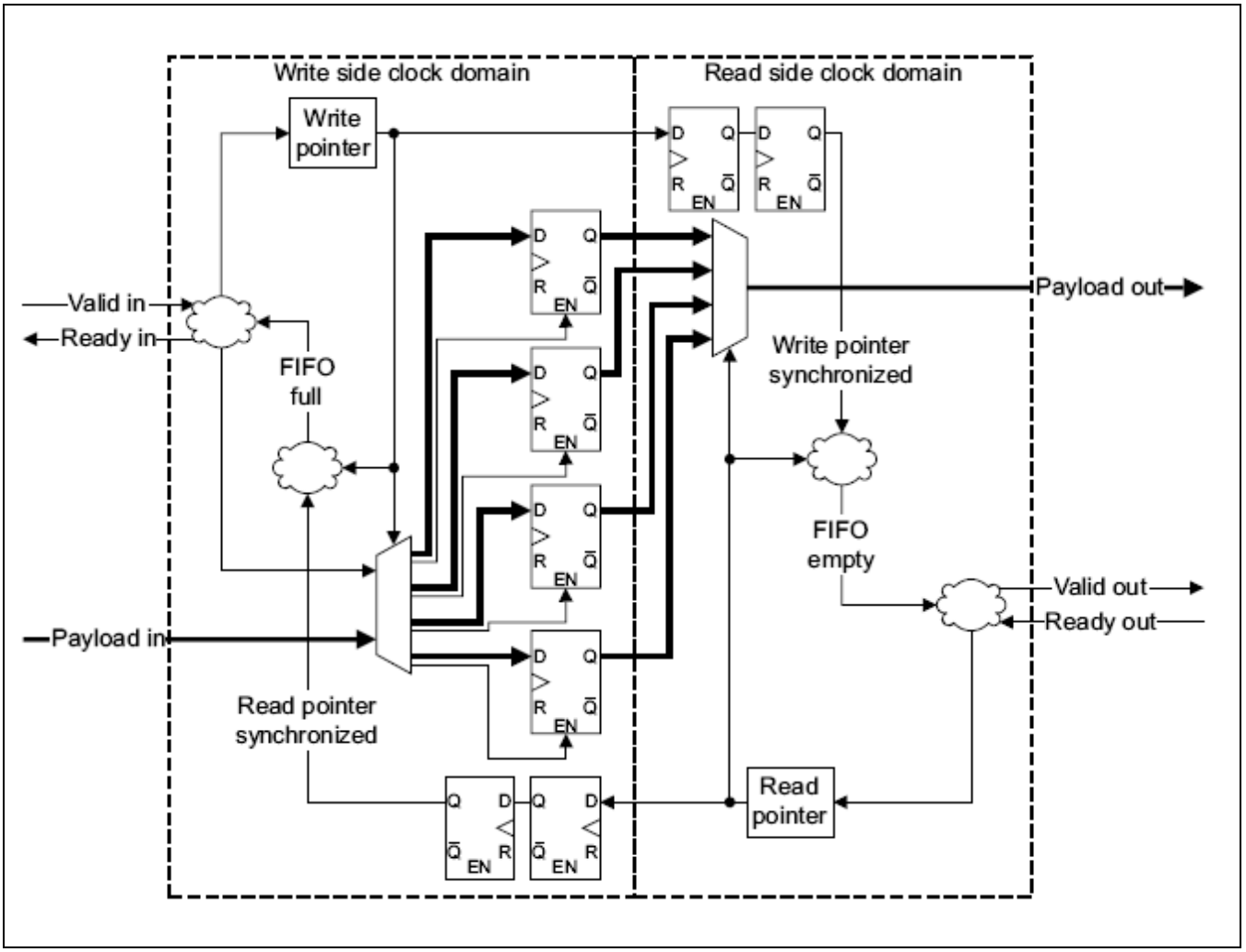


Figure 3.4-3 Internal Structure of a Four-Place Asynchronous FIFO

The asynchronous FIFOs are implemented as an array of data storage elements in parallel. The write enables and read multiplexing are controlled by separate circular counters that operate in each clock domain. The data storage elements provide the mechanism for data to cross the timing boundary.

The read counter is made available to the write process and the write counter is made available to the read process. This determines when the FIFO is either:

- Full, and cannot be written to
- Empty, and cannot be read from.

Each counter must be synchronized to the other clock domain because the counters are updated in different timing domains. The counters use a Gray code to ensure the safety of operation of the FIFOs, therefore the synchronization always captures either:

- Old value of the counter
- New value of the counter.

It cannot capture an intermediate state because only one bit of the counter changes between adjacent Gray code values. This ensures that if the old value is captured then the FIFO appears:

- on the read side, to be emptier than it is, the write side seems to have written less than it has
- on the write side, to be fuller than it is, the read side seems to have read less than it has.

In both cases, this indicates right-side failure that does not enable overflow or underflow. Subsequent synchronization captures the new value. Figure 3.4-4 shows timing for a single place FIFO.

Each channel exhibits latency between two and three read-clock cycles from the write-clock edge on which the data is accepted, and the earliest read-clock edge at which the data can be read. The latency experienced by any particular transfer depends on the occupancy of the FIFO with the minimum only guaranteed when starting with the FIFO in an empty state. The range of two to three cycles is because of the following:

- The write-clock edge that is the start of the latency interval can occur anywhere in the read-clock cycle
- The possibility of metastability in the synchronization process between the timing domains.

Sustained throughput in this mode cannot exceed that of the slower side. Based on the amount of buffering, the short term throughput can appear significantly higher than the sustained rate.

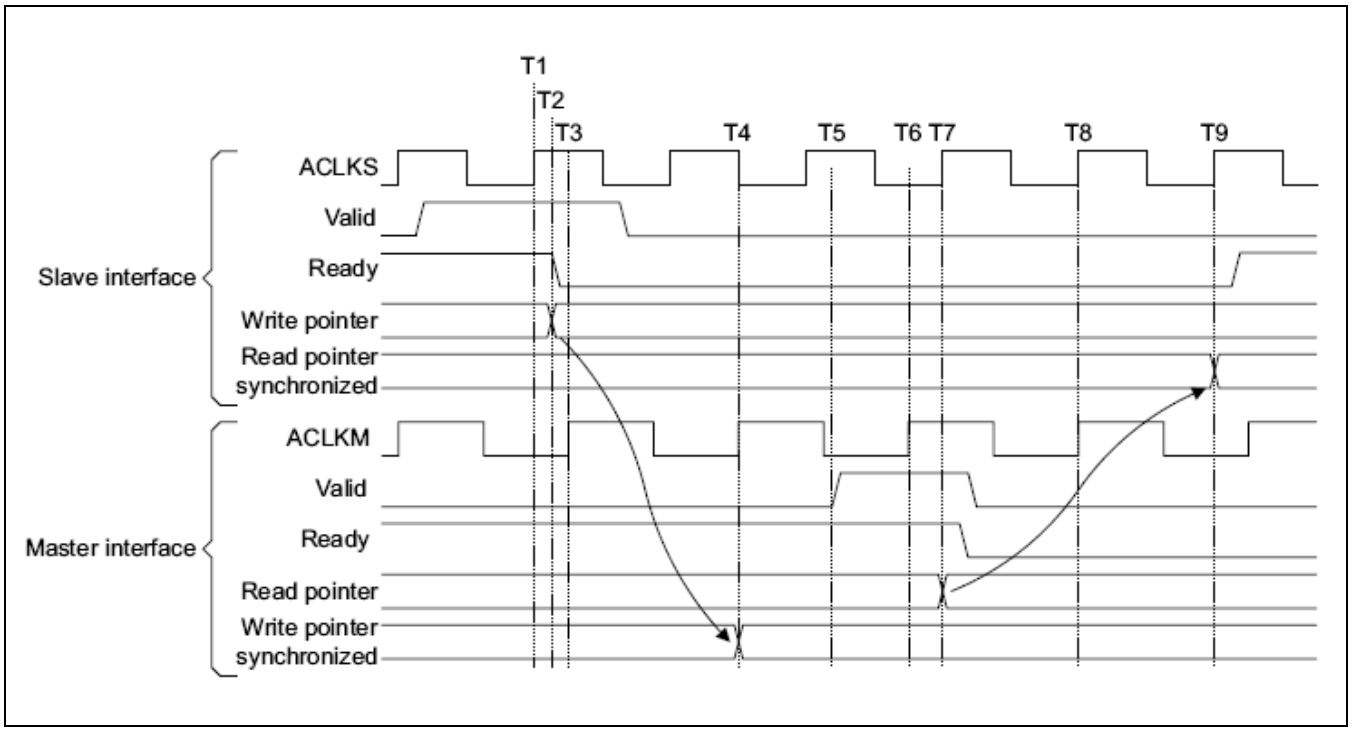


Figure 3.4-4 Single Place FIFO Timing

T1 Data is written into the FIFO.

T2 The write pointer update occurs.

T3 The first synchronization stage of this value to the read side occurs because enough time has elapsed since the write pointer updated.

T4 The second synchronization stage occurs and enables synchronized pointer visible to the read side.

T5 The read side asserts its valid signal because it has compared the read and synchronized write pointers and they are different.

T6 The data is read.

T7 The read pointer update occurs. The old or new value being captured is unpredictable because this value is changing on the ACLKS edge, when the synchronizer is sampling. In this case the old value is captured.

T8 The read pointer starts synchronizing to the write side.

T9 Synchronization is complete.

After T9 The read side can again indicate that it is ready to accept data.

1.3.2 Gray Code

The Gray code used for an N-entry FIFO has 2N values in its sequence, and the order of the values is chosen so that each value, when XORed with the value N places away from it, gives the same result and does not give that result when XORed with any other value in the sequence. This enables the fullness and emptiness to be determined by comparing the two counters:

- Emptiness is shown by the read and write counters being the same.
- Fullness is shown by the result of XORing the read and write counters being the first value in the second half of the sequence, at position N.

The required Gray code can be calculated by an algorithm:

- The number of bits required in the Gray code, k, can be calculated as:

  k = ceil(log2(N)) +1

  Where ceil represents a function that rounds its argument up to the nearest integer.

- The first N values, positions 0 to N-1 in the sequence, are derived from the binary value of their position in the sequence: Bits of the binary value B[k-1:0] is mapped to the bits of Gray coded value G[k-1:0] using the following rules:

  G[k-1] = B[k-1] (== 0)

  G[i] = B[i] XOR B[i+1]

  where 0<=i<k-1

- The second half of the sequence, positions N to 2N-1, is found by taking the value at position N-1, inverting its most significant bit and then XORing that with each value in the first half of the sequence. The first value in the second half of the sequence is the fullness value for the sequence. Figure 3.4-5 shows a typical sequence of length 6.

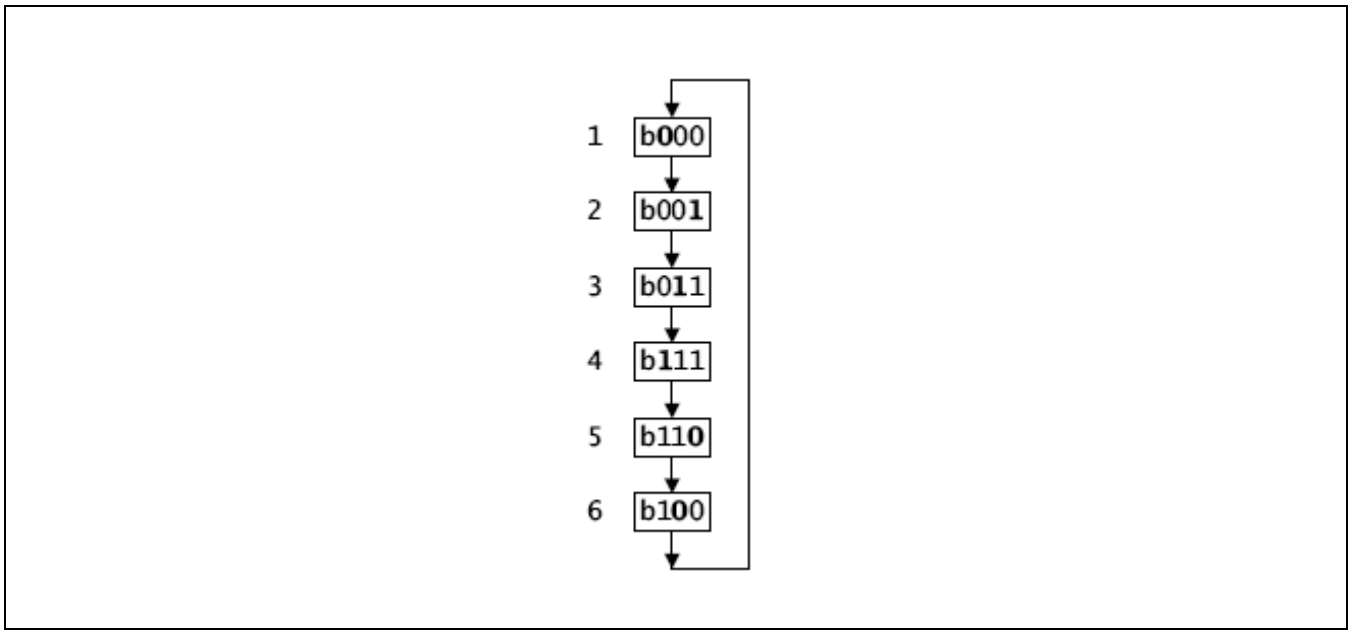


Figure 3.4-5 Example Gray Code Sequence of Length 6 for 3-Entry FIFO

This results in a sequence where the bits order change between values M and M+1 is the same as the order that they change between M+N and M+N+1. For example, in a sequence of 6 values and 3 bits, the order in which bits change is 0, 1, 2, 0, 1, 2.

2 THE PERFORMANCE IMPROVEMENT OF ASYNC BRIDGE IN S5PC100

2.1 LATENCY REDUCTION

The most outstanding architecture of S5PC100 is asynchronous clock scheme between D0 domain and D1 domain as shown in Figure 3.4-6. D0 domain is composed of high-speed memory system and the CPU system, while D1 domain is composed of multimedia systems including MFC, LCD, Camera interface and TV system whose performance largely depends on memory bandwidth.

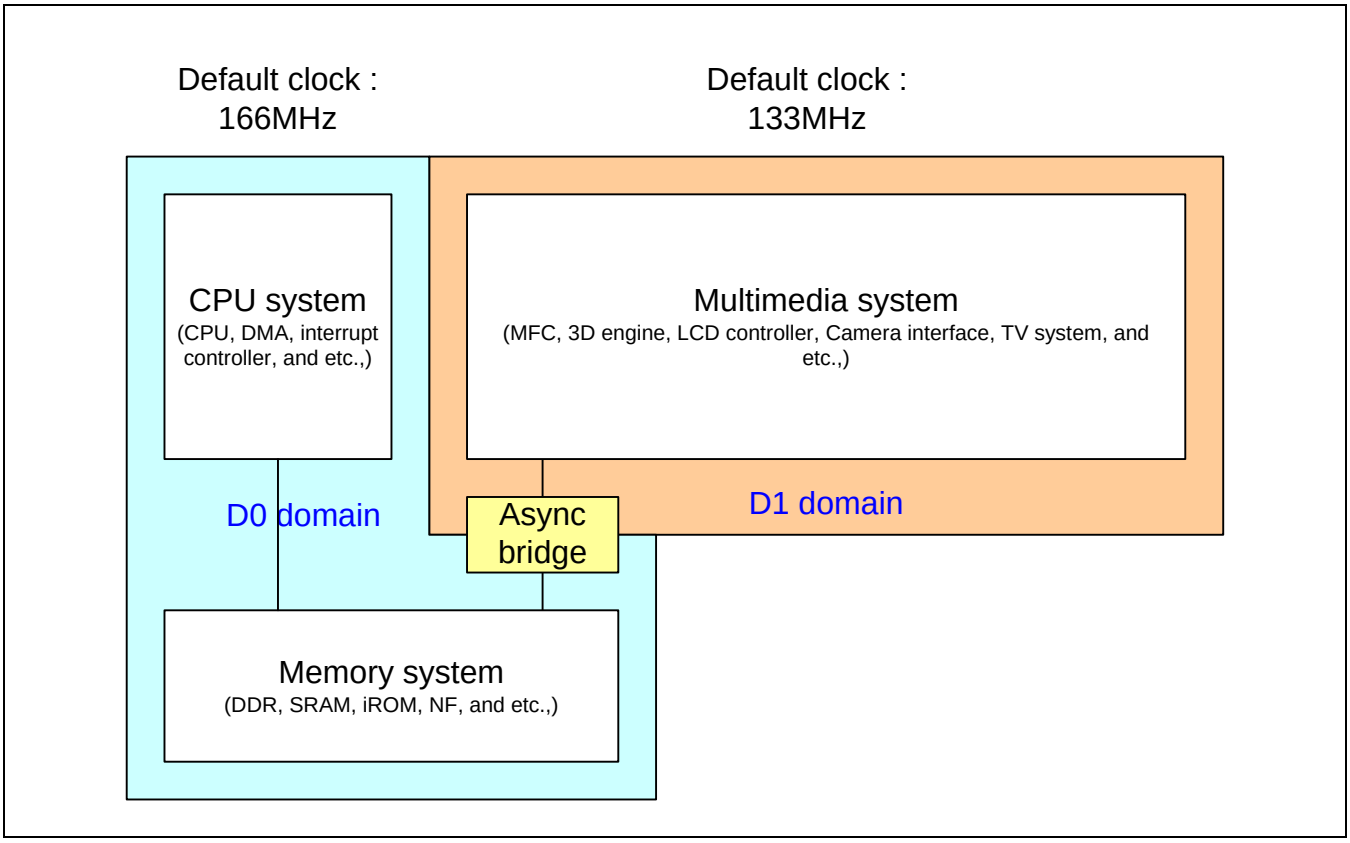


Figure 3.4-6 Simplified S5PC100 System bus Structure Focused on Clock

Although, the most masters of D1 domain's performance depend not on memory latency but on memory bandwidth, the 3D engine has cache interface whose performance is affected by memory latency. The MFC has limited ability of hiding memory latency. Therefore, the less latency Async Bridge yields, the better performance D1 domain can be obtained.

S5PC100 improved general asynchronous bridge scheme by adding path which has 1/2 clock cycle synchronizer as shown in Figure 3.4-7.

When the meta-stability is removed in 1/2 cycle, you can program Async Bridge to use 1/2 cycle synchronizer, otherwise, Async Bridge uses 1 cycle synchronizer as default. The change of synchronizer path can be executed without stopping the Async Bridge operation.

In case of 1 cycle synchronizer, the Async Bridge penalty is 2.5 cycle at master domain clock + 2.5 cycle at slave domain clock. By applying 1/2 cycle synchronizer, the Async Bridge penalty can be reduced by 0.5 cycle at each clock domain.

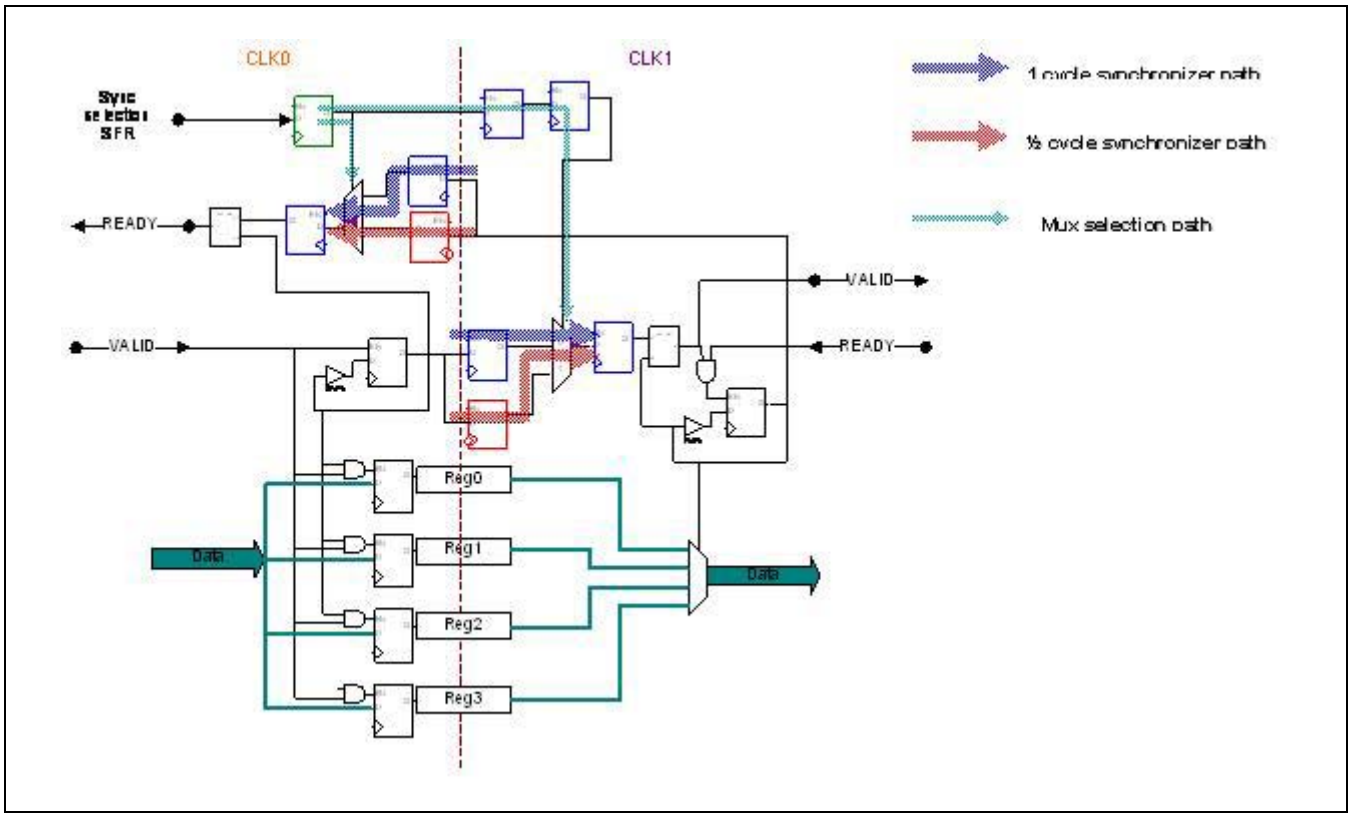


Figure 3.4-7 Dynamically Programmable Synchronizer Async Bridge Scheme

For example, let D1 domain has 133MHz clock, and D0 domain has 166MHz clock. With 1 cycle synchronizer, the initial latency of DRAM from the master of D1 domain is about 15.12 cycle at D1 clock, while the latency is about 14.22 cycle by adopting 1/2 cycle synchronizer.

In S5PC100 system, the 7~8% degradation of 3D engine is observed in the experiment which inserts the register slice between 3D engine and the DRAM (The full register slice increases total 2 cycles latency).

Hence, we expect that using 1/2 cycle synchronizer enhances the 3D performance by 3~4%.

2.2 BANDWIDTH IMPROVEMENT

When the read data transaction is distributed evenly, there will be no penalty in D0 domain with any Async Bridge in case of the dram utilization is smaller then the clock speed rate between D1/D0.

However, the actual transactions are distributed in burst; you must consider the architecture of Async Bridge.

2.2.1 Read Performance

When the master in slow domain burst reads slave in fast domain, the read FIFO of Async Bridge may overflow. It causes read data channel stall, and the stalling of data channel creates bubbles in data bus.

Because the read data bubbles are generated after DRAM controller operation, there is no bus component that can remove these bubbles except Async Bridge.

Fortunately, these read data bubbles can be reduced by increasing read data FIFO size of Async Bridge. In S5PC100, the read data FIFO size is increased to 32. We assumed the worst case is the 4 successive 16burst is one bunch of transaction. And the 4 bunch of transaction are serviced with the 10 cycle interval between the bunch of transactions. Figure 3.4-8 shows the timing diagram of upper situation to help the decision logic of read data FIFO size.

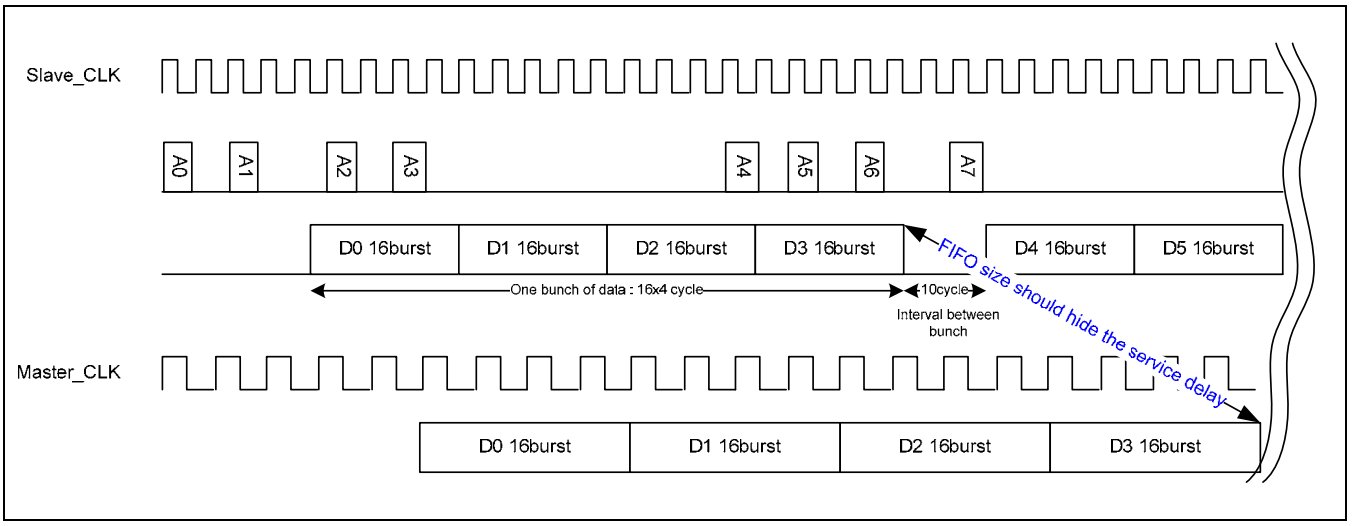


Figure 3.4-8 Example Timing Diagram of Async Bridge for Deciding the Read Data FIFO Size

2.2.2 Write Performance

When the master in slow domain burst writes slave in fast domain, the write data FIFO of Async Bridge may underflow. It causes the fast bus stalling to wait for supplying of slow master's data, and the stalling of data channel makes bubbles in write data bus. Unfortunately, the write data bubbles cannot be removed by any kind of DRAM controller. Increasing the write data FIFO size is also useless in this case.

There might be two types of bubbles caused by Async Bridge.

First, the difference of clock speed of master and slaves makes bubbles in write data channel itself. The master writes down the write data to the write FIFO of Async Bridge in low speed as long as FIFO is not full. The slave can fetch the write data from write FIFO of Async Bridge in relative high speed as long as WREADY is asserted. However, the slave's fetching speed is faster than the master's filling speed, after fetching several beat of data, the slave should wait for the master's writing down. It causes the bubbles in the write data channel.

In S5PC100, in order to avoid the bubble condition, WREADY of Async Bridge's slave part is delayed for proper cycles according to the clock speed rate of master and slave as shown in Figure 3.4-9.

As shown in Figure 3.4-9, although we delayed the WREADY signal, and it causes delayed reception of write data to slave, the last write data is not delayed at all. That means, there is no latency penalty with bubble squeezing scheme.

The size of write data FIFO is also increased to 16 in order to accumulate the write data without FIFO overflow, when the slave part of Async Bridge wait for delayed WREADY signal.

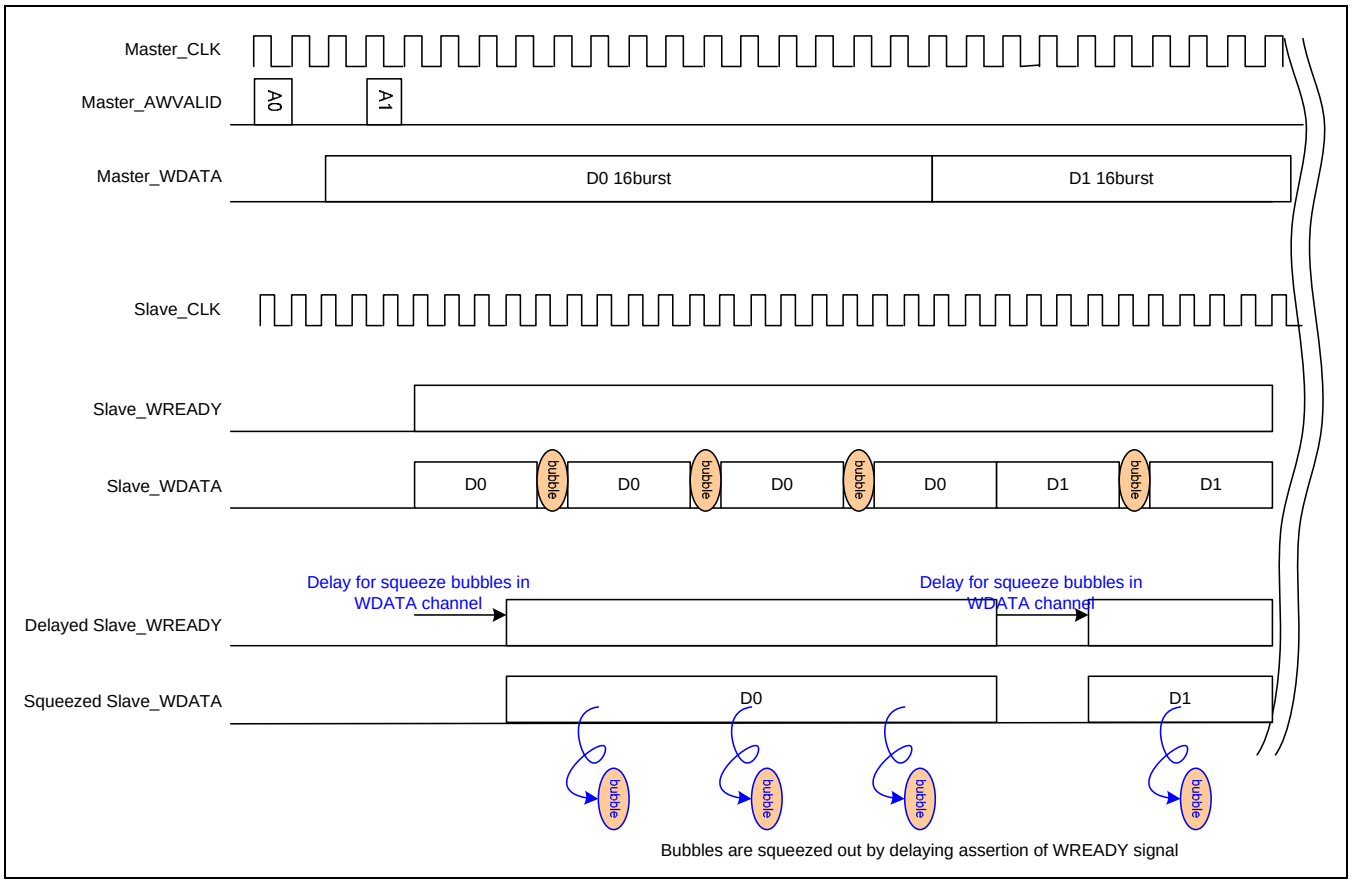


Figure 3.4-9 Write Data Bubble Squeezing Scheme in Async Bridge

Second, the big bubble between the write address channel and the write data channel should be investigated.

According to the AXI bus specification, the master issues the data of write transactions in the same order in which it issues the transaction addresses. Therefore, If first arbitrated address channel's master is slower than the second arbitrated address channel's master, the second arbitrated master should wait for all write data transaction of slow master, although the second master can finish transaction of write data before first master's starting of write data transaction. This is a big bubble.

The bubble generation mechanism and its solution are depicted in Figure 3.4-10. The basic idea is very simple. The AWVALID assertion of slave part of Async Bridge is delayed as like delaying WREADY in first write enhancement scheme.

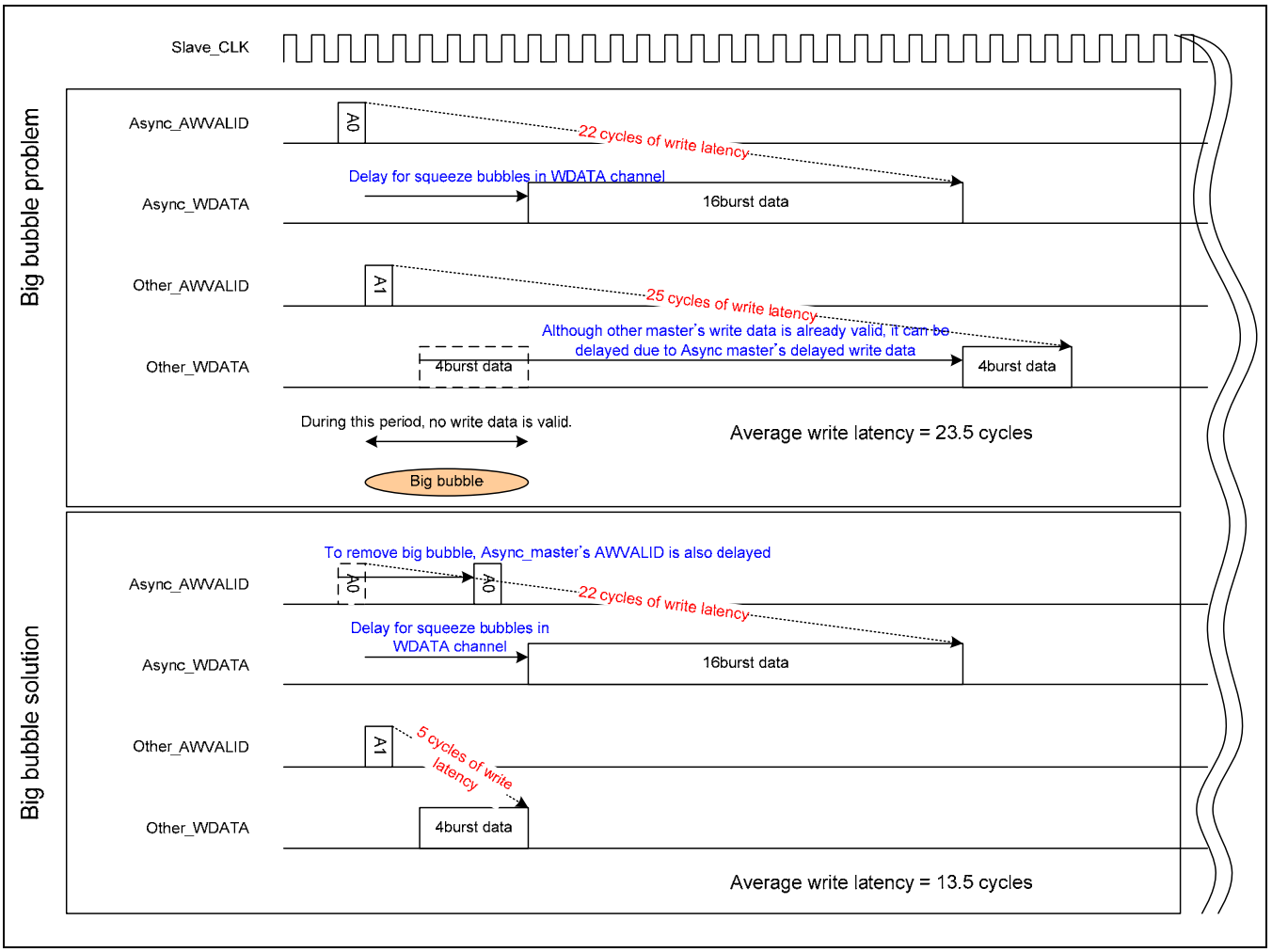


Figure 3.4-10 Big Bubble Between AW Channel and W Channel and its Solution in Async Bridge

The delay cycle of AW channel and W channel depends on both the clock speed rate of fast and slow clock domain and the burst size of transaction. Table 3.4-1 shows the required delay cycle for various burst transactions and the fast and slow clock rate. For example, if the D0 clock speed is 166MHz, and the D1 clock speed is 133MHz, the Async_index should be set as "2" to enhance write performance of Async Bridge. The Async Bridge automatically delays the AW channel and W channel to appropriate cycle for burst size of transactions.

If the AXI interconnect's arbitration scheme is priority base and Async Bridge has highest priority, Async Bridge is handled as a default master. In that case, the AXI interconnect can give WREADY signal to default master 1 cycle earlier than to other masters. If Async Bridge is default master, AWVALID signal should be delayed 1 more cycle to increase bandwidth capacity of write data channel.

Table 3.4-1 The Delay Cycle for Various Burst Transactions and the Clock Combinations

Clock & Index				Bubble Size				Delay Cycle			
D1/D0 Clock Rate	Master Clock	Slave Clock	Async Index	4	8	12	16	4	8	12	16
200:166	200.0	166.7	1	-1.4	-0.6	0.2	1	0	0	1	1
166:133	166.7	133.3	2	-1.3	-0.3	0.7	1.8	0	0	1	2
133:100	133.3	100.0	3	-1	0.3	1.7	3	0	1	2	3
100:66	100.0	66.7	4	-0.5	1.5	3.5	5.5	0	2	4	6
66:33	66.7	33.3	5	1	5	9	13	1	5	9	13

3 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
HALFSYNC	0xE340_0000	R/W	1/2 cycle synchronizer selection	0x00
ASYNC_INDEX	0xE340_0004	R/W	Async index register for write enhancing	0x00
DMASTER	0xE340_0008	R/W	Optimizing delay for default master	0x00

NOTE: If async bridge has highest priority in AXI interconnect, then async bridge is handled as a default master.

3.1 HALFSYNC REGISTER (HALFSYNC, R/W, 0XE340_0000)

HALFSYNC	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
HALFSYNC SEL	[0]	0 = Using 1 cycle synchronizer 1 = Using 1/2 cycle synchronizer	0

3.2 ASYNCINDEX REGISTER (ASYNCINDEX, R/W, 0XE340_0004)

ASYNCINDEX	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
ASYNC_INDEX	[2:0]	0 = D0 clock : D1 clock = 1: 1 & W channel BW improve off 1 = D0 clock : D1 clock = 6: 5 & W channel BW improve on 2 = D0 clock : D1 clock = 5: 4 & W channel BW improve on 3 = D0 clock : D1 clock = 4: 3 & W channel BW improve on 4 = D0 clock : D1 clock = 3: 2 & W channel BW improve on 5 = D0 clock : D1 clock = 2: 1 & W channel BW improve on	0x0

3.3 DMASTER REGISTER (DMASTER, R/W, 0XE340_0008)

Name	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
DEFAULT_MASTER	[0]	0 = Optimizing AW channel delay to non-default master 1 = Optimizing AW channel delay to default master	0x0

NOTE: All SFR can be programmed at anytime

4.1 VECTORED INTERRUPT CONTROLLER

1 OVERVIEW

The interrupt controller in S5PC100 is composed of 3 VIC's (Vectored Interrupt Controller, ARM PrimeCell PL192) and 3 TZIC's (TrustZone Interrupt Controller, SP890).

Three TZIC's and three VIC's are daisy-chained to support up to 94 interrupt sources. The TZIC provides a software interface to the secure interrupt system in a TrustZone design. It provides secure control of the nFIQ interrupt and masks the interrupt source(s) from the interrupt controller on the non-secure side of the system (VIC). Use the latter to generate nIRQ signal.

To generate nFIQ from the non-secure interrupt sources, the TZIC0 takes the nNSFIQIN signal from the non-secure interrupt controller.

1.1 FEATURES

- Supports 94 vectored IRQ interrupts
- Fixed hardware interrupts priority levels
- Programmable interrupt priority levels
- Supports Hardware interrupt priority level masking
- Programmable interrupt priority level masking
- Generates IRQ and FIQ
- Generates Software interrupt
- Test registers
- Raw interrupt status
- Interrupt request status
- Supports Privileged mode for restricted access

1.2 BLOCK DIAGRAM

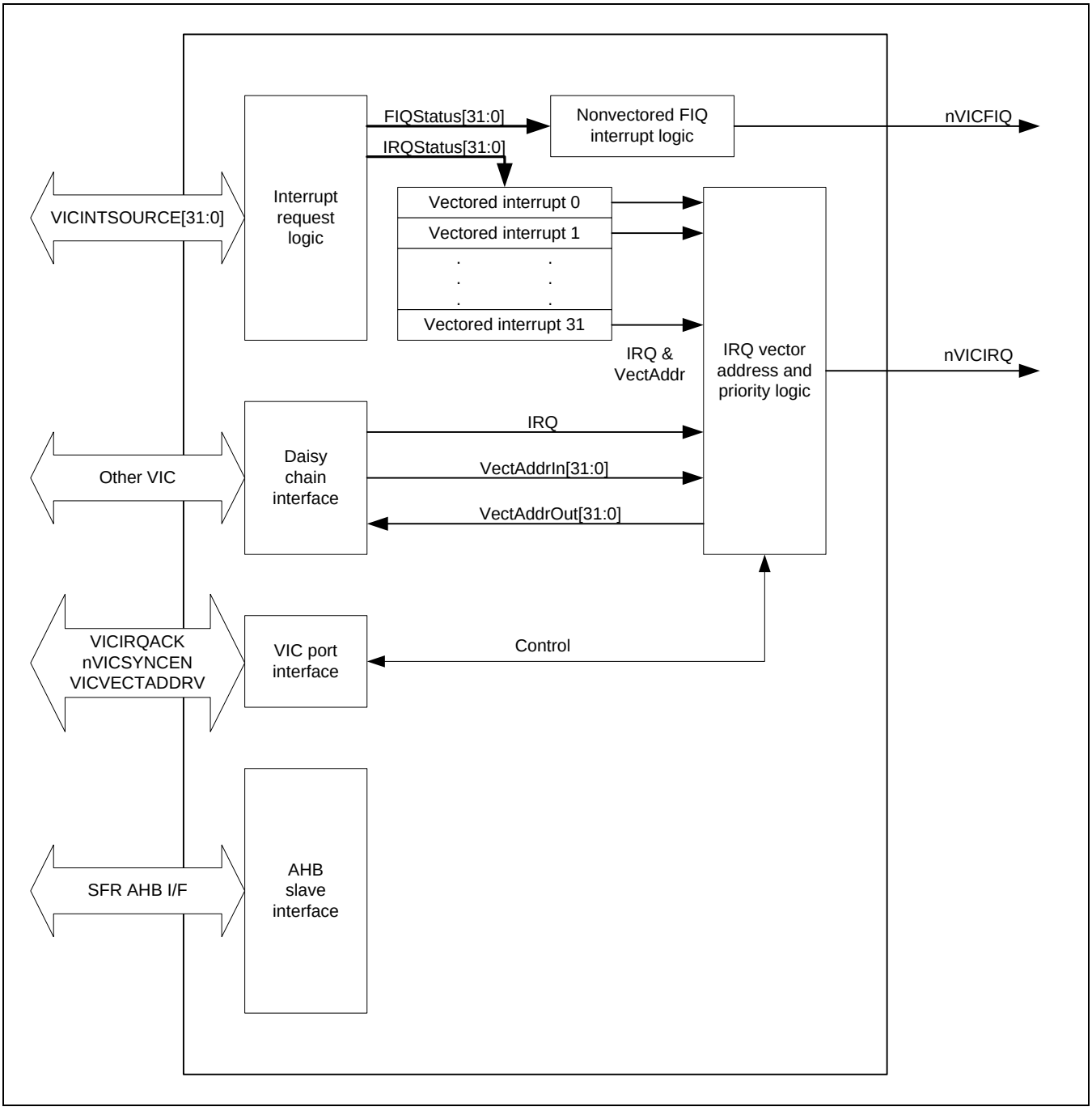


Figure 4.1-1 VICn Diagram

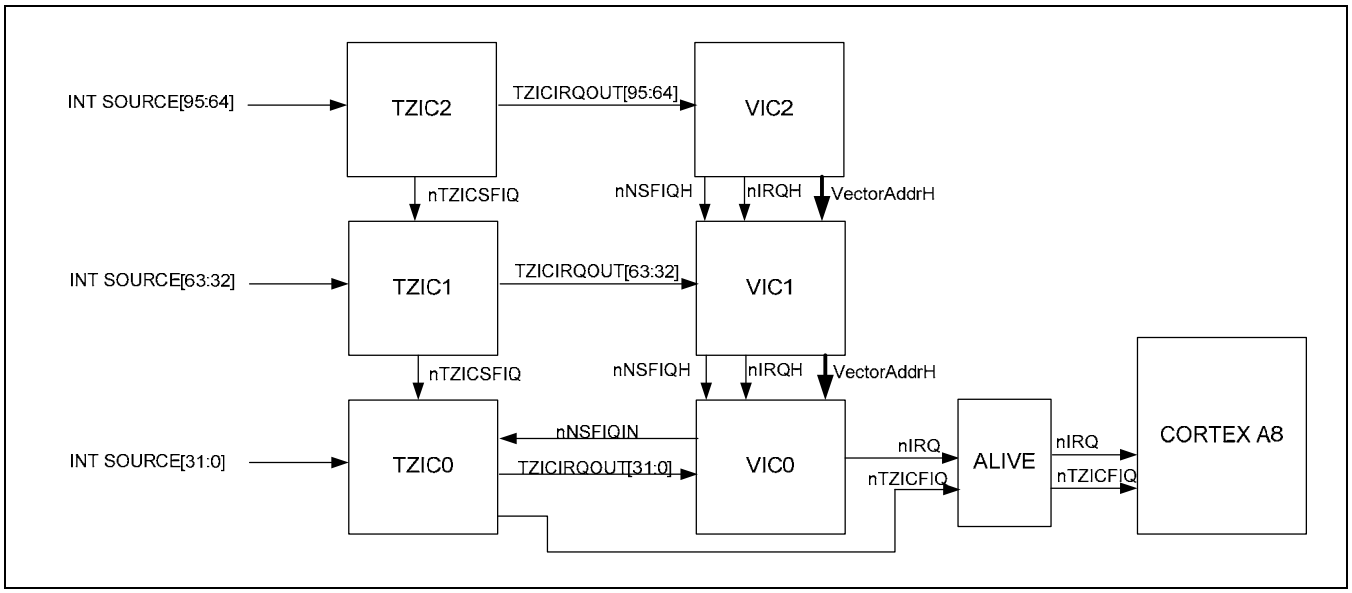


Figure 4.1-2 VIC Daisy Chain

2 INTERRUPT SOURCE

The S5PC100 supports interrupt sources as shown in the Table below.

Module	No	INT Request	Description
VIC2 Multimedia, Audio, Security, Etc.,	95	**SDM_FIQ** (security)	SDM Security Violation Detect FIQ Interrupt
	94	**SDM_IRQ** (security)	SDM Security Violation Detect IRQ Interrupt
	93	**SEC_TX** (security)	Crypto Engine's TX FIFO Interrupt
	92	**SEC_RX** (security)	Crypto Engine's RX FIFO Interrupt
	91	**SEC** (security)	Crypto Engine Interrupt
	90	**Reserved**	
	89	**KEYPAD**	Keypad Interrupt
	88	**PENDN**	TSADC Pen Down Interrupt
	87	**ADC**	TSADC EOC (End of conversion) Interrupt
	86	**SPDIF**	SPDIF Interrupt
	85	**PCM1**	PCM1 Interrupt
	84	**PCM0**	PCM0 Interrupt
	83	**AC97**	AC97 Interrupt
	82	**I2S2**	I2S 2 Interrupt
	81	**I2S1**	I2S 1 Interrupt
	80	**I2S0**	I2S 0 Interrupt
	79	**TVENC**	TV Encoder Interrupt
	78	**MFC**	MFC Interrupt
	77	**I2C1 (for HDMI)**	I2C1 Interrupt
	76	**HDMI**	HDMI Interrupt
	75	**Mixer**	Mixer Interrupt
	74	**3D**	3D Graphic Controller Interrupt
	73	**2D**	2D Interrupt
	72	**JPEG**	JPEG Interrupt
	71	**FIMC2**	Camera Controller 2 Interrupt
	70	**FIMC1**	Camera Controller 1 Interrupt
	69	**FIMC0**	Camera Controller 0 Interrupt
	68	**ROTATOR**	Rotator Interrupt
	67	**LCD[3]**	LCD Controller Interrupt 3
	66	**LCD[2]**	LCD Controller Interrupt 2
	65	**LCD[1]**	LCD Controller Interrupt 1
	64	**LCD[0]**	LCD Controller Interrupt 0

Module	No	INT Request	Description
VIC1 ARM, power, memory, Connectivity, Storage	63	**Reserved**	
	62	**MIPI_DSI**	MIPI DSI Interrupt
	61	**MIPI_CSI**	MIPI CSI Interrupt
	60	**HSMMC2**	HSMMC2 Interrupt
	59	**HSMMC1**	HSMMC1 Interrupt
	58	**HSMMC0**	HSMMC0 Interrupt
	57	**MODEM**	Modem I/F Interrupt
	56	**OTG** (usb)	USB OTG Interrupt
	55	**UHOST** (usb)	USB Host Interrupt
	54	**HSItx**	HSI Tx Interrupt
	53	**HSIrx**	HSI Rx Interrupt
	52	**CAN1**	CAN 1 Interrupt
	51	**CAN0**	CAN 0 Interrupt
	50	**IrDA**	IrDA Interrupt
	49	**SPI2**	SPI2 Interrupt
	48	**SPI1**	SPI1 Interrupt
	47	**SPI0**	SPI0 Interrupt
	46	**I2C0**	I2C0 Interrupt
	45	**UART3**	UART3 Interrupt
	44	**UART2**	UART2 Interrupt
	43	**UART1**	UART1 Interrupt
	42	**UART0**	UART0 Interrupt
	41	**CFC**	CFCON Interrupt
	40	**NFC**	NFCON Interrupt
	39	**ONENAND**	OneNAND Interrupt
	38	**IEM_IEC**	IEM_IEC Interrupt
	37	**IEM_APC**	IEM_APC Interrupt
	36	**CORTEX4**	nCTIIRQ Interrupt
	35	**CORTEX3**	nDMAEXTERIRQ Interrupt
	34	**CORTEX2**	nDMAIRQ Interrupt
	33	**CORTEX1**	nDMASIRQ Interrupt
	32	**CORTEX0**	nPMUIRQ Interrupt

Module	No	INT Request	Description
VIC0 System, DMA, Timer	31	**Reserved**	
	30	**GPIOINT**	All other GPIO Interrupt (ORed)
	29	**RTC_TIC**	RTC TIC Interrupt
	28	**RTC_ALARM**	RTC Alarm Interrupt
	27	**WDT**	Watchdog Timer Interrupt
	26	**System Timer**	System Timer Interrupt
	25	**TIMER4**	Timer 4 Interrupt
	24	**TIMER3**	Timer 3 Interrupt
	23	**TIMER2**	Timer 2 Interrupt
	22	**TIMER1**	Timer 1 Interrupt
	21	**TIMER0**	Timer 0 Interrupt
	20	**PDMA1**	Peri DMA Interrupt
	19	**PDMA0**	Peri DMA Interrupt
	18	**MDMA**	M2M DMA Interrupt
	17	**BATF**	Battery Fault Interrupt
	16	**EINT 16_31**	EXT_INT[16] ~ [31]
	15	**EINT15**	EXT_INT[15]
	14	**EINT14**	EXT_INT[14]
	13	**EINT13**	EXT_INT[13]
	12	**EINT12**	EXT_INT[12]
	11	**EINT11**	EXT_INT[11]
	10	**EINT10**	EXT_INT[10]
	9	**EINT9**	EXT_INT[9]
	8	**EINT8**	EXT_INT[8]
	7	**EINT7**	EXT_INT[7]
	6	**EINT6**	EXT_INT[6]
	5	**EINT5**	EXT_INT[5]
	4	**EINT4**	EXT_INT[4]
	3	**EINT3**	EXT_INT[3]
	2	**EINT2**	EXT_INT[2]
	1	**EINT1**	EXT_INT[1]
	0	**EINT0**	EXT_INT[0]

3 FUNCTIONAL DESCRIPTION

When user clears interrupt pending, user must write 0 to all the VICADDRESS registers (VIC0ADDRESS, VIC1ADDRESS, and VIC2ADDRESS).

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
VIC0IRQSTATUS	0xE400_0000	R	IRQ Status Register	0x00000000
VIC0FIQSTATUS	0xE400_0004	R	FIQ Status Register	0x00000000
VIC0RAWINTR	0xE400_0008	R	Raw Interrupt Status Register	-
VIC0INTSELECT	0xE400_000C	R/W	Interrupt Select Register	0x00000000
VIC0INTENABLE	0xE400_0010	R/W	Interrupt Enable Register	0x00000000
VIC0INTENCLEAR	0xE400_0014	W	Interrupt Enable Clear Register	-
VIC0SOFTINT	0xE400_0018	R/W	Software Interrupt Register	0x00000000
VIC0SOFTINTCLEAR	0xE400_001C	W	Software Interrupt Clear Register	-
VIC0PROTECTION	0xE400_0020	R/W	Protection Enable Register	0x0
VIC0SWPRIORITYMASK	0xE400_0024	R/W	Software Priority Mask Register	0xFFFF
VIC0PRIORITYDAISY	0xE400_0028	R/W	Vector Priority Register for Daisy Chain	0xF
VIC0VECTADDR0	0xE400_0100	R/W	Vector Address 0 Register	0x00000000
VIC0VECTADDR1	0xE400_0104	R/W	Vector Address 1 Register	0x00000000
VIC0VECTADDR2	0xE400_0108	R/W	Vector Address 2 Register	0x00000000
VIC0VECTADDR3	0xE400_010C	R/W	Vector Address 3 Register	0x00000000
VIC0VECTADDR4	0xE400_0110	R/W	Vector Address 4 Register	0x00000000
VIC0VECTADDR5	0xE400_0114	R/W	Vector Address 5 Register	0x00000000
VIC0VECTADDR6	0xE400_0118	R/W	Vector Address 6 Register	0x00000000
VIC0VECTADDR7	0xE400_011C	R/W	Vector Address 7 Register	0x00000000
VIC0VECTADDR8	0xE400_0120	R/W	Vector Address 8 Register	0x00000000
VIC0VECTADDR9	0xE400_0124	R/W	Vector Address 9 Register	0x00000000
VIC0VECTADDR10	0xE400_0128	R/W	Vector Address 10 Register	0x00000000
VIC0VECTADDR11	0xE400_012C	R/W	Vector Address 11 Register	0x00000000
VIC0VECTADDR12	0xE400_0130	R/W	Vector Address 12 Register	0x00000000
VIC0VECTADDR13	0xE400_0134	R/W	Vector Address 13 Register	0x00000000
VIC0VECTADDR14	0xE400_0138	R/W	Vector Address 14 Register	0x00000000
VIC0VECTADDR15	0xE400_013C	R/W	Vector Address 15 Register	0x00000000
VIC0VECTADDR16	0xE400_0140	R/W	Vector Address 16 Register	0x00000000
VIC0VECTADDR17	0xE400_0144	R/W	Vector Address 17 Register	0x00000000
VIC0VECTADDR18	0xE400_0148	R/W	Vector Address 18 Register	0x00000000
VIC0VECTADDR19	0xE400_014C	R/W	Vector Address 19 Register	0x00000000
VIC0VECTADDR20	0xE400_0150	R/W	Vector Address 20 Register	0x00000000
VIC0VECTADDR21	0xE400_0154	R/W	Vector Address 21 Register	0x00000000
VIC0VECTADDR22	0xE400_0158	R/W	Vector Address 22 Register	0x00000000

Register	Address	R/W	Description	Reset Value
VIC0VECTADDR23	0xE400_015C	R/W	Vector Address 23 Register	0x00000000
VIC0VECTADDR24	0xE400_0160	R/W	Vector Address 24 Register	0x00000000
VIC0VECTADDR25	0xE400_0164	R/W	Vector Address 25 Register	0x00000000
VIC0VECTADDR26	0xE400_0168	R/W	Vector Address 26 Register	0x00000000
VIC0VECTADDR27	0xE400_016C	R/W	Vector Address 27 Register	0x00000000
VIC0VECTADDR28	0xE400_0170	R/W	Vector Address 28 Register	0x00000000
VIC0VECTADDR29	0xE400_0174	R/W	Vector Address 29 Register	0x00000000
VIC0VECTADDR30	0xE400_0178	R/W	Vector Address 30 Register	0x00000000
VIC0VECTADDR31	0xE400_017C	R/W	Vector Address 31 Register	0x00000000
VIC0VECPRIORITY0	0xE400_0200	R/W	Vector Priority 0 Register	0xF
VIC0VECTPRIORITY1	0xE400_0204	R/W	Vector Priority 1 Register	0xF
VIC0VECTPRIORITY2	0xE400_0208	R/W	Vector Priority 2 Register	0xF
VIC0VECTPRIORITY3	0xE400_020C	R/W	Vector Priority 3 Register	0xF
VIC0VECTPRIORITY4	0xE400_0210	R/W	Vector Priority 4 Register	0xF
VIC0VECTPRIORITY5	0xE400_0214	R/W	Vector Priority 5 Register	0xF
VIC0VECTPRIORITY6	0xE400_0218	R/W	Vector Priority 6 Register	0xF
VIC0VECTPRIORITY7	0xE400_021C	R/W	Vector Priority 7 Register	0xF
VIC0VECTPRIORITY8	0xE400_0220	R/W	Vector Priority 8 Register	0xF
VIC0VECTPRIORITY9	0xE400_0224	R/W	Vector Priority 9 Register	0xF
VIC0VECTPRIORITY10	0xE400_0228	R/W	Vector Priority 10 Register	0xF
VIC0VECTPRIORITY11	0xE400_022C	R/W	Vector Priority 11 Register	0xF
VIC0VECTPRIORITY12	0xE400_0230	R/W	Vector Priority 12 Register	0xF
VIC0VECTPRIORITY13	0xE400_0234	R/W	Vector Priority 13 Register	0xF
VIC0VECTPRIORITY14	0xE400_0238	R/W	Vector Priority 14 Register	0xF
VIC0VECTPRIORITY15	0xE400_023C	R/W	Vector Priority 15 Register	0xF
VIC0VECTPRIORITY16	0xE400_0240	R/W	Vector Priority 16 Register	0xF
VIC0VECTPRIORITY17	0xE400_0244	R/W	Vector Priority 17 Register	0xF
VIC0VECTPRIORITY18	0xE400_0248	R/W	Vector Priority 18 Register	0xF
VIC0VECTPRIORITY19	0xE400_024C	R/W	Vector Priority 19 Register	0xF
VIC0VECTPRIORITY20	0xE400_0250	R/W	Vector Priority 20 Register	0xF
VIC0VECTPRIORITY21	0xE400_0254	R/W	Vector Priority 21 Register	0xF
VIC0VECTPRIORITY22	0xE400_0258	R/W	Vector Priority 22 Register	0xF
VIC0VECTPRIORITY23	0xE400_025C	R/W	Vector Priority 23 Register	0xF
VIC0VECTPRIORITY24	0xE400_0260	R/W	Vector Priority 24 Register	0xF
VIC0VECTPRIORITY25	0xE400_0264	R/W	Vector Priority 25 Register	0xF

Register	Address	R/W	Description	Reset Value
VIC0VECTPRIORITY26	0xE400_0268	R/W	Vector Priority 26 Register	0xF
VIC0VECTPRIORITY27	0xE400_026C	R/W	Vector Priority 27 Register	0xF
VIC0VECTPRIORITY28	0xE400_0270	R/W	Vector Priority 28 Register	0xF
VIC0VECTPRIORITY29	0xE400_0274	R/W	Vector Priority 29 Register	0xF
VIC0VECTPRIORITY30	0xE400_0278	R/W	Vector Priority 30 Register	0xF
VIC0VECTPRIORITY31	0xE400_027C	R/W	Vector Priority 31 Register	0xF
VIC0ADDRESS	0xE400_0F00	R/W	Vector Address Register	0x00000000
VIC0PERIPHID0	0xE400_0FE0	R	Peripheral Identification Register bit 7:0	0x92
VIC0PERIPHID1	0xE400_0FE4	R	Peripheral Identification Register bit 15:9	0x11
VIC0PERIPHID2	0xE400_0FE8	R	Peripheral Identification Register bit 23:16	0x04
VIC0PERIPHID3	0xE400_0FEC	R	Peripheral Identification Register bit 31:24	0x00
VIC0PCELLID0	0xE400_0FF0	R	PrimeCell Identification Register bit 7:0	0x0D
VIC0PCELLID1	0xE400_0FF4	R	PrimeCell Identification Register bit 15:9	0xF0
VIC0PCELLID2	0xE400_0FF8	R	PrimeCell Identification Register bit 23:16	0x05
VIC0PCELLID3	0xE400_0FFC	R	PrimeCell Identification Register bit 31:24	0xB1
VIC1IRQSTATUS	0xE410_0000	R	IRQ Status Register	0x00000000
VIC1FIQSTATUS	0xE410_0004	R	FIQ Status Register	0x00000000
VIC1RAWINTR	0xE410_0008	R	Raw Interrupt Status Register	-
VIC1INTSELECT	0xE410_000C	R/W	Interrupt Select Register	0x00000000
VIC1INTENABLE	0xE410_0010	R/W	Interrupt Enable Register	0x00000000
VIC1INTENCLEAR	0xE410_0014	W	Interrupt Enable Clear Register	-
VIC1SOFTINT	0xE410_0018	R/W	Software Interrupt Register	0x00000000
VIC1SOFTINTCLEAR	0xE410_001C	W	Software Interrupt Clear Register	-
VIC1PROTECTION	0xE410_0020	R/W	Protection Enable Register	0x0
VIC1SWPRIORITYMASK	0xE410_0024	R/W	Software Priority Mask Register	0xFFFF
VIC1PRIORITYDAISY	0xE410_0028	R/W	Vector Priority Register for Daisy Chain	0xF
VIC1VECTADDR0	0xE410_0100	R/W	Vector Address 0 Register	0x00000000
VIC1VECTADDR1	0xE410_0104	R/W	Vector Address 1 Register	0x00000000
VIC1VECTADDR2	0xE410_0108	R/W	Vector Address 2 Register	0x00000000
VIC1VECTADDR3	0xE410_010C	R/W	Vector Address 3 Register	0x00000000
VIC1VECTADDR4	0xE410_0110	R/W	Vector Address 4 Register	0x00000000
VIC1VECTADDR5	0xE410_0114	R/W	Vector Address 5 Register	0x00000000
VIC1VECTADDR6	0xE410_0118	R/W	Vector Address 6 Register	0x00000000
VIC1VECTADDR7	0xE410_011C	R/W	Vector Address 7 Register	0x00000000

Register	Address	R/W	Description	Reset Value
VIC1VECTADDR8	0xE410_0120	R/W	Vector Address 8 Register	0x00000000
VIC1VECTADDR9	0xE410_0124	R/W	Vector Address 9 Register	0x00000000
VIC1VECTADDR10	0xE410_0128	R/W	Vector Address 10 Register	0x00000000
VIC1VECTADDR11	0xE410_012C	R/W	Vector Address 11 Register	0x00000000
VIC1VECTADDR12	0xE410_0130	R/W	Vector Address 12 Register	0x00000000
VIC1VECTADDR13	0xE410_0134	R/W	Vector Address 13 Register	0x00000000
VIC1VECTADDR14	0xE410_0138	R/W	Vector Address 14 Register	0x00000000
VIC1VECTADDR15	0xE410_013C	R/W	Vector Address 15 Register	0x00000000
VIC1VECTADDR16	0xE410_0140	R/W	Vector Address 16 Register	0x00000000
VIC1VECTADDR17	0xE410_0144	R/W	Vector Address 17 Register	0x00000000
VIC1VECTADDR18	0xE410_0148	R/W	Vector Address 18 Register	0x00000000
VIC1VECTADDR19	0xE410_014C	R/W	Vector Address 19 Register	0x00000000
VIC1VECTADDR20	0xE410_0150	R/W	Vector Address 20 Register	0x00000000
VIC1VECTADDR21	0xE410_0154	R/W	Vector Address 21 Register	0x00000000
VIC1VECTADDR22	0xE410_0158	R/W	Vector Address 22 Register	0x00000000
VIC1VECTADDR23	0xE410_015C	R/W	Vector Address 23 Register	0x00000000
VIC1VECTADDR24	0xE410_0160	R/W	Vector Address 24 Register	0x00000000
VIC1VECTADDR25	0xE410_0164	R/W	Vector Address 25 Register	0x00000000
VIC1VECTADDR26	0xE410_0168	R/W	Vector Address 26 Register	0x00000000
VIC1VECTADDR27	0xE410_016C	R/W	Vector Address 27 Register	0x00000000
VIC1VECTADDR28	0xE410_0170	R/W	Vector Address 28 Register	0x00000000
VIC1VECTADDR29	0xE410_0174	R/W	Vector Address 29 Register	0x00000000
VIC1VECTADDR30	0xE410_0178	R/W	Vector Address 30 Register	0x00000000
VIC1VECTADDR31	0xE410_017C	R/W	Vector Address 31 Register	0x00000000
VIC1VECPRIORITY0	0xE410_0200	R/W	Vector Priority 0 Register	0xF
VIC1VECTPRIORITY1	0xE410_0204	R/W	Vector Priority 1 Register	0xF
VIC1VECTPRIORITY2	0xE410_0208	R/W	Vector Priority 2 Register	0xF
VIC1VECTPRIORITY3	0xE410_020C	R/W	Vector Priority 3 Register	0xF
VIC1VECTPRIORITY4	0xE410_0210	R/W	Vector Priority 4 Register	0xF
VIC1VECTPRIORITY5	0xE410_0214	R/W	Vector Priority 5 Register	0xF
VIC1VECTPRIORITY6	0xE410_0218	R/W	Vector Priority 6 Register	0xF
VIC1VECTPRIORITY7	0xE410_021C	R/W	Vector Priority 7 Register	0xF
VIC1VECTPRIORITY8	0xE410_0220	R/W	Vector Priority 8 Register	0xF
VIC1VECTPRIORITY9	0xE410_0224	R/W	Vector Priority 9 Register	0xF
VIC1VECTPRIORITY10	0xE410_0228	R/W	Vector Priority 10 Register	0xF

Register	Address	R/W	Description	Reset Value
VIC1VECTPRIORITY11	0xE410_022C	R/W	Vector Priority 11 Register	0xF
VIC1VECTPRIORITY12	0xE410_0230	R/W	Vector Priority 12 Register	0xF
VIC1VECTPRIORITY13	0xE410_0234	R/W	Vector Priority 13 Register	0xF
VIC1VECTPRIORITY14	0xE410_0238	R/W	Vector Priority 14 Register	0xF
VIC1VECTPRIORITY15	0xE410_023C	R/W	Vector Priority 15 Register	0xF
VIC1VECTPRIORITY16	0xE410_0240	R/W	Vector Priority 16 Register	0xF
VIC1VECTPRIORITY17	0xE410_0244	R/W	Vector Priority 17 Register	0xF
VIC1VECTPRIORITY18	0xE410_0248	R/W	Vector Priority 18 Register	0xF
VIC1VECTPRIORITY19	0xE410_024C	R/W	Vector Priority 19 Register	0xF
VIC1VECTPRIORITY20	0xE410_0250	R/W	Vector Priority 20 Register	0xF
VIC1VECTPRIORITY21	0xE410_0254	R/W	Vector Priority 21 Register	0xF
VIC1VECTPRIORITY22	0xE410_0258	R/W	Vector Priority 22 Register	0xF
VIC1VECTPRIORITY23	0xE410_025C	R/W	Vector Priority 23 Register	0xF
VIC1VECTPRIORITY24	0xE410_0260	R/W	Vector Priority 24 Register	0xF
VIC1VECTPRIORITY25	0xE410_0264	R/W	Vector Priority 25 Register	0xF
VIC1VECTPRIORITY26	0xE410_0268	R/W	Vector Priority 26 Register	0xF
VIC1VECTPRIORITY27	0xE410_026C	R/W	Vector Priority 27 Register	0xF
VIC1VECTPRIORITY28	0xE410_0270	R/W	Vector Priority 28 Register	0xF
VIC1VECTPRIORITY29	0xE410_0274	R/W	Vector Priority 29 Register	0xF
VIC1VECTPRIORITY30	0xE410_0278	R/W	Vector Priority 30 Register	0xF
VIC1VECTPRIORITY31	0xE410_027C	R/W	Vector Priority 31 Register	0xF
VIC1ADDRESS	0xE410_0F00	R/W	Vector Address Register	0x00000000
VIC1PERIPHID0	0xE410_0FE0	R	Peripheral Identification Register bit 7:0	0x92
VIC1PERIPHID1	0xE410_0FE4	R	Peripheral Identification Register bit 15:9	0x11
VIC1PERIPHID2	0xE410_0FE8	R	Peripheral Identification Register bit 23:16	0x04
VIC1PERIPHID3	0xE410_0FEC	R	Peripheral Identification Register bit 31:24	0x00
VIC1PCELLID0	0xE410_0FF0	R	PrimeCell Identification Register bit 7:0	0x0D
VIC1PCELLID1	0xE410_0FF4	R	PrimeCell Identification Register bit 15:9	0xF0
VIC1PCELLID2	0xE410_0FF8	R	PrimeCell Identification Register bit 23:16	0x05
VIC1PCELLID3	0xE410_0FFC	R	PrimeCell Identification Register bit 31:24	0xB1
VIC2IRQSTATUS	0xE420_0000	R	IRQ Status Register	0x00000000
VIC2FIQSTATUS	0xE420_0004	R	FIQ Status Register	0x00000000
VIC2RAWINTR	0xE420_0008	R	Raw Interrupt Status Register	-

Register	Address	R/W	Description	Reset Value
VIC2INTSELECT	0xE420_000C	R/W	Interrupt Select Register	0x00000000
VIC2INTENABLE	0xE420_0010	R/W	Interrupt Enable Register	0x00000000
VIC2INTENCLEAR	0xE420_0014	W	Interrupt Enable Clear Register	-
VIC2SOFTINT	0xE420_0018	R/W	Software Interrupt Register	0x00000000
VIC2SOFTINTCLEAR	0xE420_001C	W	Software Interrupt Clear Register	-
VIC2PROTECTION	0xE420_0020	R/W	Protection Enable Register	0x0
VIC2SWPRIORITYMASK	0xE420_0024	R/W	Software Priority Mask Register	0xFFFF
VIC2PRIORITYDAISY	0xE420_0028	R/W	Vector Priority Register for Daisy Chain	0xF
VIC2VECTADDR0	0xE420_0100	R/W	Vector Address 0 Register	0x00000000
VIC2VECTADDR1	0xE420_0104	R/W	Vector Address 1 Register	0x00000000
VIC2VECTADDR2	0xE420_0108	R/W	Vector Address 2 Register	0x00000000
VIC2VECTADDR3	0xE420_010C	R/W	Vector Address 3 Register	0x00000000
VIC2VECTADDR4	0xE420_0110	R/W	Vector Address 4 Register	0x00000000
VIC2VECTADDR5	0xE420_0114	R/W	Vector Address 5 Register	0x00000000
VIC2VECTADDR6	0xE420_0118	R/W	Vector Address 6 Register	0x00000000
VIC2VECTADDR7	0xE420_011C	R/W	Vector Address 7 Register	0x00000000
VIC2VECTADDR8	0xE420_0120	R/W	Vector Address 8 Register	0x00000000
VIC2VECTADDR9	0xE420_0124	R/W	Vector Address 9 Register	0x00000000
VIC2VECTADDR10	0xE420_0128	R/W	Vector Address 10 Register	0x00000000
VIC2VECTADDR11	0xE420_012C	R/W	Vector Address 11 Register	0x00000000
VIC2VECTADDR12	0xE420_0130	R/W	Vector Address 12 Register	0x00000000
VIC2VECTADDR13	0xE420_0134	R/W	Vector Address 13 Register	0x00000000
VIC2VECTADDR14	0xE420_0138	R/W	Vector Address 14 Register	0x00000000
VIC2VECTADDR15	0xE420_013C	R/W	Vector Address 15 Register	0x00000000
VIC2VECTADDR16	0xE420_0140	R/W	Vector Address 16 Register	0x00000000
VIC2VECTADDR17	0xE420_0144	R/W	Vector Address 17 Register	0x00000000
VIC2VECTADDR18	0xE420_0148	R/W	Vector Address 18 Register	0x00000000
VIC2VECTADDR19	0xE420_014C	R/W	Vector Address 19 Register	0x00000000
VIC2VECTADDR20	0xE420_0150	R/W	Vector Address 20 Register	0x00000000
VIC2VECTADDR21	0xE420_0154	R/W	Vector Address 21 Register	0x00000000
VIC2VECTADDR22	0xE420_0158	R/W	Vector Address 22 Register	0x00000000
VIC2VECTADDR23	0xE420_015C	R/W	Vector Address 23 Register	0x00000000
VIC2VECTADDR24	0xE420_0160	R/W	Vector Address 24 Register	0x00000000
VIC2VECTADDR25	0xE420_0164	R/W	Vector Address 25 Register	0x00000000
VIC2VECTADDR26	0xE420_0168	R/W	Vector Address 26 Register	0x00000000

Register	Address	R/W	Description	Reset Value
VIC2VECTADDR27	0xE420_016C	R/W	Vector Address 27 Register	0x00000000
VIC2VECTADDR28	0xE420_0170	R/W	Vector Address 28 Register	0x00000000
VIC2VECTADDR29	0xE420_0174	R/W	Vector Address 29 Register	0x00000000
VIC2VECTADDR30	0xE420_0178	R/W	Vector Address 30 Register	0x00000000
VIC2VECTADDR31	0xE420_017C	R/W	Vector Address 31 Register	0x00000000
VIC2VECPRIORITY0	0xE420_0200	R/W	Vector Priority 0 Register	0xF
VIC2VECTPRIORITY1	0xE420_0204	R/W	Vector Priority 1 Register	0xF
VIC2VECTPRIORITY2	0xE420_0208	R/W	Vector Priority 2 Register	0xF
VIC2VECTPRIORITY3	0xE420_020C	R/W	Vector Priority 3 Register	0xF
VIC2VECTPRIORITY4	0xE420_0210	R/W	Vector Priority 4 Register	0xF
VIC2VECTPRIORITY5	0xE420_0214	R/W	Vector Priority 5 Register	0xF
VIC2VECTPRIORITY6	0xE420_0218	R/W	Vector Priority 6 Register	0xF
VIC2VECTPRIORITY7	0xE420_021C	R/W	Vector Priority 7 Register	0xF
VIC2VECTPRIORITY8	0xE420_0220	R/W	Vector Priority 8 Register	0xF
VIC2VECTPRIORITY9	0xE420_0224	R/W	Vector Priority 9 Register	0xF
VIC2VECTPRIORITY10	0xE420_0228	R/W	Vector Priority 10 Register	0xF
VIC2VECTPRIORITY11	0xE420_022C	R/W	Vector Priority 11 Register	0xF
VIC2VECTPRIORITY12	0xE420_0230	R/W	Vector Priority 12 Register	0xF
VIC2VECTPRIORITY13	0xE420_0234	R/W	Vector Priority 13 Register	0xF
VIC2VECTPRIORITY14	0xE420_0238	R/W	Vector Priority 14 Register	0xF
VIC2VECTPRIORITY15	0xE420_023C	R/W	Vector Priority 15 Register	0xF
VIC2VECTPRIORITY16	0xE420_0240	R/W	Vector Priority 16 Register	0xF
VIC2VECTPRIORITY17	0xE420_0244	R/W	Vector Priority 17 Register	0xF
VIC2VECTPRIORITY18	0xE420_0248	R/W	Vector Priority 18 Register	0xF
VIC2VECTPRIORITY19	0xE420_024C	R/W	Vector Priority 19 Register	0xF
VIC2VECTPRIORITY20	0xE420_0250	R/W	Vector Priority 20 Register	0xF
VIC2VECTPRIORITY21	0xE420_0254	R/W	Vector Priority 21 Register	0xF
VIC2VECTPRIORITY22	0xE420_0258	R/W	Vector Priority 22 Register	0xF
VIC2VECTPRIORITY23	0xE420_025C	R/W	Vector Priority 23 Register	0xF
VIC2VECTPRIORITY24	0xE420_0260	R/W	Vector Priority 24 Register	0xF
VIC2VECTPRIORITY25	0xE420_0264	R/W	Vector Priority 25 Register	0xF
VIC2VECTPRIORITY26	0xE420_0268	R/W	Vector Priority 26 Register	0xF
VIC2VECTPRIORITY27	0xE420_026C	R/W	Vector Priority 27 Register	0xF
VIC2VECTPRIORITY28	0xE420_0270	R/W	Vector Priority 28 Register	0xF
VIC2VECTPRIORITY29	0xE420_0274	R/W	Vector Priority 29 Register	0xF

Register	Address	R/W	Description	Reset Value
VIC2VECTPRIORITY30	0xE420_0278	R/W	Vector Priority 30 Register	0xF
VIC2VECTPRIORITY31	0xE420_027C	R/W	Vector Priority 31 Register	0xF
VIC2ADDRESS	0xE420_0F00	R/W	Vector Address Register	0x00000000
VIC2PERIPHID0	0xE420_0FE0	R	Peripheral Identification Register bit 7:0	0x92
VIC2PERIPHID1	0xE420_0FE4	R	Peripheral Identification Register bit 15:9	0x11
VIC2PERIPHID2	0xE420_0FE8	R	Peripheral Identification Register bit 23:16	0x04
VIC2PERIPHID3	0xE420_0FEC	R	Peripheral Identification Register bit 31:24	0x00
VIC2PCELLID0	0xE420_0FF0	R	PrimeCell Identification Register bit 7:0	0x0D
VIC2PCELLID1	0xE420_0FF4	R	PrimeCell Identification Register bit 15:9	0xF0
VIC2PCELLID2	0xE420_0FF8	R	PrimeCell Identification Register bit 23:16	0x05
VIC2PCELLID3	0xE420_0FFC	R	PrimeCell Identification Register bit 31:24	0xB1

Register	Address	R/W	Description	Reset Value
TZIC0FIQStatus	0xE500_0000	R	FIQ Status Register	0x00000000
TZIC0RawIntr	0xE500_0004	R	Raw Interrupt Status Register	-
TZIC0IntSelect	0xE500_0008	R/W	Interrupt Select Register	0x00000000
TZIC0FIQEnable	0xE500_000C	R/W	FIQ Enable Register	0x00000000
TZIC0FIQENClear	0xE500_0010	W	FIQ Enable Clear Register	-
TZIC0FIQBypass	0xE500_0014	R/W	FIQ Bypass Register	0x00000000
TZIC0Protection	0xE500_0018	R/W	Protection Register	0x00000000
TZIC0Lock	0xE500_001C	W	Lock Enable Register	-
TZIC0LockStatus	0xE500_0020	R	Lock Status Register	0x00000001
TZIC0PeriphID0	0xE500_0FE0	R	Peripheral Identification Registers	0x00000090
TZIC0PeriphID1	0xE500_0FE4	R		0x00000018
TZIC0PeriphID2	0xE500_0FE8	R		0x00000004
TZIC0PeriphID3	0xE500_0FEC	R		0x00000000
TZIC0PCellID0	0xE500_0FF0	R	Identification Registers	0x0000000D
TZIC0PCellID1	0xE500_0FF4	R		0x000000F0
TZIC0PCellID2	0xE500_0FF8	R		0x00000005
TZIC0PCellID3	0xE500_0FFC	R		0x000000B1
TZIC1FIQStatus	0xE510_0000	R	FIQ Status Register	0x00000000
TZIC1RawIntr	0xE510_0004	R	Raw Interrupt Status Register	-
TZIC1IntSelect	0xE510_0008	R/W	Interrupt Select Register	0x00000000
TZIC1FIQEnable	0xE510_000C	R/W	FIQ Enable Register	0x00000000
TZIC1FIQENClear	0xE510_0010	W	FIQ Enable Clear Register	-
TZIC1FIQBypass	0xE510_0014	R/W	FIQ Bypass Register	0x00000000
TZIC1Protection	0xE510_0018	R/W	Protection Register	0x00000000
TZIC1Lock	0xE510_001C	W	Lock Enable Register	-
TZIC1LockStatus	0xE510_0020	R	Lock Status Register	0x00000001
TZIC1PeriphID0	0xE510_0FE0	R	Peripheral Identification Registers	0x00000090
TZIC1PeriphID1	0xE510_0FE4	R		0x00000018
TZIC1PeriphID2	0xE510_0FE8	R		0x00000004
TZIC1PeriphID3	0xE510_0FEC	R		0x00000000
TZIC1PCellID0	0xE510_0FF0	R	Identification Registers	0x0000000D
TZIC1PCellID1	0xE510_0FF4	R		0x000000F0
TZIC1PCellID2	0xE510_0FF8	R		0x00000005
TZIC1PCellID3	0xE510_0FFC	R		0x000000B1
TZIC2FIQStatus	0xE520_0000	R	FIQ Status Register	0x00000000

Register	Address	R/W	Description	Reset Value
TZIC2RawIntr	0xE520_0004	R	Raw Interrupt Status Register	-
TZIC2IntSelect	0xE520_0008	R/W	Interrupt Select Register	0x00000000
TZIC2FIQEnable	0xE520_000C	R/W	FIQ Enable Register	0x00000000
TZIC2FIQENClear	0xE520_0010	W	FIQ Enable Clear Register	-
TZIC2FIQBypass	0xE520_0014	R/W	FIQ Bypass Register	0x00000000
TZIC2Protection	0xE520_0018	R/W	Protection Register	0x00000000
TZIC2Lock	0xE520_001C	W	Lock Enable Register	-
TZIC2LockStatus	0xE520_0020	R	Lock Status Register	0x00000001
TZIC2PeriphID0	0xE520_0FE0	R	Peripheral Identification Registers	0x00000090
TZIC2PeriphID1	0xE520_0FE4	R		0x00000018
TZIC2PeriphID2	0xE520_0FE8	R		0x00000004
TZIC2PeriphID3	0xE520_0FEC	R		0x00000000
TZIC2PCellID0	0xE520_0FF0	R	Identification Registers	0x0000000D
TZIC2PCellID1	0xE520_0FF4	R		0x000000F0
TZIC2PCellID2	0xE520_0FF8	R		0x00000005
TZIC2PCellID3	0xE520_0FFC	R		0x000000B1

4.1 IRQ STATUS REGISTER (VICIRQSTATUS, R, ADDRESS=0XE400_0000, 0XE410_0000, 0XE420_0000)

VICIRQSTATUS	Bit	Description	Reset Value
IRQStatus	[31:0]	Shows the status of the interrupts after masking by the VICINTENABLE and VICINTSELECT Registers: 0 = Interrupt is inactive 1 = Interrupt is active. There is one bit of the register for each interrupt source.	0x00000000

4.2 FIQ STATUS REGISTER (VICFIQSTATUS, R, ADDRESS=0XE400_0004, 0XE410_0004, 0XE420_0004)

VICFIQSTATUS	Bit	Description	Reset Value
FIQStatus	[31:0]	Shows the status of the FIQ interrupts after masking by the VICINTENABLE and VICINTSELECT Registers: 0 = Interrupt is inactive 1 = Interrupt is active. There is one bit of the register for each interrupt source.	0x00000000

4.3 RAW INTERRUPT STATUS REGISTER (VICRAWINTR, R, ADDRESS=0XE400_0008, 0XE410_0008, 0XE420_0008)

VICRAWINTR	Bit	Description	Reset Value
RawInterrupt	[31:0]	Shows the status of the FIQ interrupts before masking by the VICINTENABLE and VICINTSELECT Registers: 0 = Interrupt is inactive before masking 1 = Interrupt is active before masking Because this register provides a direct view of the raw interrupt inputs, the reset value is unknown. There is one bit of the register for each interrupt source.	-

4.4 INTERRUPT SELECT REGISTER
(VICINTSELECT, R/W, ADDRESS=0XE400_000C, 0XE410_000C, 0XE420_000C)

VICINTSELECT	Bit	Description	Reset Value
IntSelect	[31:0]	Selects interrupt type for interrupt request: 0 = IRQ interrupt 1 = FIQ interrupt There is one bit of the register for each interrupt source.	0x00000000

4.5 INTERRUPT ENABLE REGISTER
(VICINTENABLE, R/W, ADDRESS=0XE400_0010, 0XE410_0010, 0XE420_0010)

VICINTENABLE	Bit	Description	Reset Value
IntEnable	[31:0]	Enables the interrupt request lines, which allows the interrupts to reach the processor. Read: 0 = Disables Interrupt 1 = Enables Interrupt Use this register to enable interrupt. The VICINTENCLEAR Register must be used to disable the interrupt enable. Write: 0 = No effect 1 = Enables Interrupt. On reset, all interrupts are disabled. There is one bit of the register for each interrupt source.	0x00000000

4.6 INTERRUPT ENABLE CLEAR
(VICINTENCLEAR, W, ADDRESS=0XE400_0014, 0XE410_0014, 0XE420_0014)

VICINTENCLEAR	Bit	Description	Reset Value
IntEnable Clear	[31:0]	Clears corresponding bits in the VICINTENABLE Register: 0 = No effect 1 = Disables Interrupt in VICINTENABLE Register. There is one bit of the register for each interrupt source.	-

4.7 SOFTWARE INTERRUPT REGISTER (VICSOFTINT, R/W, ADDRESS=0XE400_0018, 0XE410_0018, 0XE420_0018)

VICSOFTINT	Bit	Description	Reset Value
SoftInt	[31:0]	Setting a bit HIGH generates a software interrupt for the selected source before interrupt masking. Read: 0 = Software interrupt inactive 1 = Software interrupt active Write: 0 = No effect 1 = Enables Software interrupt There is one bit of the register for each interrupt source.	0x00000000

4.8 SOFTWARE INTERRUPT CLEAR REGISTER (VICSOFTINTCLEAR, W, ADDRESS=0XE400_001C, 0XE410_001C, 0XE420_001C)

VICSOFTINTCLEAR	Bit	Description	Reset Value
SoftIntClear	[31:0]	Clears corresponding bits in the VICSOFTINT Register: 0 = No effect 1 = Disables Software interrupt in the VICSOFTINT Register. There is one bit of the register for each interrupt source.	-

4.9 PROTECTION ENABLE REGISTER (VICPROTECTION, R/W, ADDRESS=0XE400_0020, 0XE410_0020, 0XE420_0020)

VICPROTECTION	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as 0, do not modify.	0x0
Protection	[0]	Enables or disables protected register access: 0 = Disables Protection mode 1 = Enables Protection mode. If enabled, only privileged mode accesses (reads and writes) can access the interrupt controller registers, that is, if HPROT[1] is set HIGH for the current transfer. If disabled, both user mode and privileged mode can access the registers. This register can only be accessed in privileged mode, even if protection mode is disabled.	0x0

4.10 VECTOR ADDRESS REGISTER (VICADDRESS, R/W, ADDRESS=0XE400_0F00, 0XE410_0F00, 0XE420_0F00)

VICADDRESS	Bit	Description	Reset Value
VectAddr	[31:0]	Contains the address of the currently active ISR, with reset value 0x00000000. A read of this register returns the address of the ISR and sets the current interrupt as being serviced. A read must be performed while there is an active interrupt. A write of any value to this register clears the current interrupt. A write must only be performed at the end of an interrupt service routine.	0x00000000

4.11 SOFTWARE PRIORITY MASK REGISTER (VICSWPRIORITYMASK, R/W, ADDRESS=0XE400_0024, 0XE410_0024, 0XE420_0024)

VICSWPRIORITYMASK	Bit	Description	Reset Value
Reserved	[31:16]	Reserved, read as 0, do not modify	0x0
SWPriorityMask	[15:0]	Controls software masking of the 16 interrupt priority levels: 0 = Interrupt priority level is masked 1 = Interrupt priority level is not masked Each bit of the register is applied to each of the 16 interrupt priority levels.	0xFFFF

4.12 VECTOR ADDRESS REGISTERS (VICVECTADDR[0-31], R/W, ADDRESS=0XE400_0100~017C, 0XE410_0100~017C, 0XE420_0100~017C)

VICVECTADDR[0-31]	Bit	Description	Reset Value
VectorAddr 0-31	[31:0]	Contains ISR vector addresses.	0x00000000

4.13 VECTOR PRIORITY REGISTERS (VICVECTPRIORITY[0-31] AND VICVECTPRIORITYDAISY, R/W, ADDRESS=0XE400_0200~027C, 0XE410_0200~027C, 0XE420_0200~027C)

VICVECTPRIORITY[0-31] and VICVECTPRIORITYDAISY	Bit	Description	Reset Value
Reserved	[31:4]	Reserved, read as 0, do not modify.	0x0
VectPriority	[3:0]	Selects vectored interrupt priority level. You can select any of the 16 vectored interrupt priority levels by programming the register with the hexadecimal value of the priority level required, from 0-15.	0xF

4.14 VICPERIPHID0 REGISTER
(VICPERIPHID0, R, ADDRESS=0XE400_0FE0, 0XE410_0FE0, 0XE420_0FE0)

VICPERIPHID0	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
Partnumber0	[7:0]	These bits read back as 0x92	0x92

4.15 VICPERIPHID1 REGISTER
(VICPERIPHID1, R, ADDRESS=0XE400_0FE4, 0XE410_0FE4, 0XE420_0FE4)

VICPERIPHID1	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
Designer0	[7:4]	These bits read back as 0x1.	0x1
Partnumber1	[3:0]	These bits read back as 0x1.	0x1

4.16 VICPERIPHID2 REGISTER
(VICPERIPHID2, R, ADDRESS=0XE400_0FE8, 0XE410_0FE8, 0XE420_0FE8)

VICPERIPHID2	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
Revision	[7:4]	These bits read back as the revision number, which can be between 0 and 15.	0x0
Designer1	[3:0]	These bits read back as 0x4.	0x4

4.17 VICPERIPHID3 REGISTER
(VICPERIPHID3, R, ADDRESS=0XE400_0FEC, 0XE410_0FEC, 0XE420_0FEC)

VICPERIPHID3	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
Configuration	[7:2]	These bits read back as 0x0.	0x0
Configuration	[1:0]	Indicates the number of interrupts supported: 00 = 32 (default) 01 = 64 10 = 128 11 = 256	0x0

4.18 VICPCELLID0 REGISTER (VICPCELLID0, R, ADDRESS=0XE400_0FF0, 0XE410_0FF0, 0XE420_0FF0)

VICPCELLID0	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
VICPCellID0	[7:0]	These bits read back as 0x0D.	0x0D

4.19 VICPCELLID1 REGISTER (VICPCELLID1, R, ADDRESS=0XE400_0FF4, 0XE410_0FF4, 0XE420_0FF4)

VICPCELLID1	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
VICPCellID1	[7:0]	These bits read back as 0xF0.	0xF0

4.20 VICPCELLID2 REGISTER (VICPCELLID2, R, ADDRESS=0XE400_0FF8, 0XE410_0FF8, 0XE420_0FF8)

VICPCELLID2	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
VICPCellID2	[7:0]	These bits read back as 0x05.	0x05

4.21 VICPCELLID3 REGISTER (VICPCELLID3, R, ADDRESS=0XE400_0FFC, 0XE410_0FFC, 0XE420_0FFC)

VICPCELLID3	Bit	Description	Reset Value
-	[31:8]	Reserved, read as 0, do not modify.	0x0
VICPCellID3	[7:0]	These bits read back as 0xB1.	0xB1

4.22 FIQ STATUS REGISTER (TZICFIQSTATUS, R, ADDRESS=0XE500_0000, 0XE510_0000, 0XE520_0000)

TZICFIQStatus	Bit	Description	Reset Value
FIQStatus	[31:0]	Shows the status of the interrupts after masking by the TZICFIQIntEnable and TZICFIQIntEnClear Registers. A HIGH bit indicates that the interrupt is active, and generates an nFIQ interrupt to the processor.	0x00000000

4.23 RAW INTERRUPT STATUS REGISTER (TZICRAWINTR, R, ADDRESS=0XE500_0004, 0XE510_0004, 0XE520_0004)

TZICRawIntr	Bit	Description	Reset Value
RawIntr	[31:0]	Shows the status of the interrupts before masking by the TZICFIQIntEnable and TZICFIQIntEnClear Registers. A HIGH bit indicates that the interrupt is active before masking.	-

4.24 INTERRUPT SELECT REGISTER (TZICINTSELECT, R/W, ADDRESS=0XE500_0008, 0XE510_0008, 0XE520_0008)

TZICRawIntr	Bit	Description	Reset Value
IntSelect	[31:0]	Selects whether the interrupt source generates an FIQ interrupt or passes straight through to **TZICIRQOUT**. 0 = Interrupt passes through to **TZICIRQOUT** 1 = Interrupt is available for FIQ generation	0x00000000

4.25 FIQ ENABLE REGISTER (TZICFIQENABLE, R/W, ADDRESS=0XE500_000C, 0XE510_000C, 0XE520_000C)

TZICFIQEnable	Bit	Description	Reset Value
FIQEnable	[31:0]	Enables the FIQ-selected interrupt lines, allowing the interrupts to reach the processor. Read: 0 = Disables Interrupt 1 = Enables Interrupt. To enable the interrupt use this register. You must use the TZICFIQEnClear Register to disable the interrupt enable. Write: 0 = No effect 1 = Enables Interrupt. If Reset it disables all interrupts. There is 1 bit of the register for each interrupt source.	0x00000000

4.26 FIQ ENABLE CLEAR REGISTER (TZICFIQENCLEAR, W, ADDRESS=0XE500_0010, 0XE510_0010, 0XE520_0010)

TZICFIQENClear	Bit	Description	Reset Value
FIQEnClear	[31:0]	Clears bits in the TZICFIQEnable Register. Writing a HIGH clears the corresponding bit in the TZICFIQEnable Register. Writing a LOW has no effect.	-

4.27 FIQ BYPASS REGISTER (TZICFIQBYPASS, R/W, ADDRESS=0XE500_0014, 0XE510_0014, 0XE520_0014)

TZICFIQBypass	Bit	Description	Reset Value
-	[31:1]	Read undefined. Write as 0.	0x0
FIQBypass	[0]	Enables **nNSFIQIN** to route directly to nFIQ. 0 = No Bypass 1 = Bypass.	0x0

4.28 PROTECTION REGISTER (TZICPROTECTION, R/W, ADDRESS=0XE500_0018, 0XE510_0018, 0XE520_0018)

TZICProtection	Bit	Description	Reset Value
-	[31:1]	Read undefined. Write as 0.	0x0
Protection	[0]	Enables or disables protected register access: 0 = Disables Protection mode 1 = Enables Protection mode. If enabled, you can only make privileged mode access (reads and writes) to the TZIC. This register is accessed in privileged mode, even if protection mode is disabled.	0x0

4.29 LOCK ENABLE REGISTER (TZICLOCK, W, ADDRESS=0XE500_001C, 0XE510_001C, 0XE520_001C)

TZICLock	Bit	Description	Reset Value
Lock	[31:0]	To enable access to the other registers in the TZIC, you must write the correct access code of 0x0ACCE550 to this register. To disable access to the other TZIC registers, you must write any other value except 0x0ACCE550 to this register.	-

4.30 LOCK STATUS REGISTER (TZICLOCKSTATUS, R, ADDRESS=0XE500_0020, 0XE510_0020, 0XE520_0020)

TZICLockStatus	Bit	Description	Reset Value
-	[31:1]	Read undefined.	0x0
Locked	[0]	Shows the locked status of the TZIC: 0 = Access to the TZIC is not locked 1 = Access to the TZIC is locked Use TZICLock Register to unlock the access	0x1

4.31 PERIPHERAL IDENTIFICATION REGISTER (TZICPERIPHID0, R, ADDRESS=0XE500_0FE0, 0XE510_0FE0, 0XE520_0FE0)

TZICPeriphID0	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
Partnumber0	[7:0]	These bits read back as 0x90	0x90

4.32 PERIPHERAL IDENTIFICATION REGISTER (TZICPERIPHID1, R, ADDRESS=0XE500_0FE4, 0XE510_0FE4, 0XE520_0FE4)

TZICPeriphID1	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
Designer0	[7:4]	These bits read back as 0x1	0x1
Partnumber1	[3:0]	These bits read back as 0x8	0x8

4.33 PERIPHERAL IDENTIFICATION REGISTER (TZICPERIPHID2, R, ADDRESS=0XE500_0FE8, 0XE510_0FE8, 0XE520_0FE8)

TZICPeriphID2	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
Revision	[7:4]	These bits read back as the revision number and can be between 0 and 15	0x0
Designer1	[3:0]	These bits read back as 0x4	0x4

4.34 PERIPHERAL IDENTIFICATION REGISTER (TZICPERIPHID3, R, ADDRESS=0XE500_0FEC, 0XE510_0FEC, 0XE520_0FEC)

TZICPeriphID3	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
Configuration	[7:0]	These bits read back as 0x00	0x0

4.35 IDENTIFICATION REGISTER (TZICPCELLID0, R, ADDRESS=0XE500_0FF0, 0XE510_0FF0, 0XE520_0FF0)

TZICPCellID0	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
TZICPCellID0	[7:0]	These bits read back as 0x0D	0x0D

4.36 IDENTIFICATION REGISTER (TZICPCELLID1, R, ADDRESS=0XE500_0FF4, 0XE510_0FF4, 0XE520_0FF4)

TZICPCellID1	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
TZICPCellID1	[7:0]	These bits read back as 0xF0	0xF0

4.37 IDENTIFICATION REGISTER (TZICPCELLID2, R, ADDRESS=0XE500_0FF8, 0XE510_0FF8, 0XE520_0FF8)

TZICPCellID2	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
TZICPCellID2	[7:0]	These bits read back as 0x05	0x05

4.38 IDENTIFICATION REGISTER (TZICPCELLID3 REGISTER, R, ADDRESS=0XE500_0FFC, 0XE510_0FFC, 0XE520_0FFC)

TZICPCellID3	Bit	Description	Reset Value
-	[31:8]	Read undefined	0x0
TZICPCellID3	[7:0]	These bits read back as 0xB1	0xB1

5.1 DRAM CONTROLLER

1 OVERVIEW

1.1 FEATURES

- Compatible with JEDEC specification of DDR2, mDDR and LPDDR2 SDRAM
- 32bit data width only
- mDDR: up to 256Mbyte[1)] per 1 nCS (up to 2 nCS's)
- DDR2: up to 1Gbyte (8-bank) per 1 nCS (up to 1 nCS)
- LPDDR2: up to 256Mbyte[1)] per 1 nCS (up to 2 nCS's)
- Optimized pipeline stage for low latency
- QoS scheme to ensure low latency for some applications
- An advanced scheduler which enables efficient out-of order operations
- Excellent chip/bank interleaving and memory interrupting
- Adapts to various low power schemes to reduce the dynamic and static current of memory
- Outstanding exclusive accesses
- Bank selective precharge policy
- 1:1 synchronous operation between AXI bus and memory interface

Note 1: Some DRAM's which are not compatible with the JEDEC specification can not be used as big as lised in the context.

1.2 BLOCK DIAGRAM

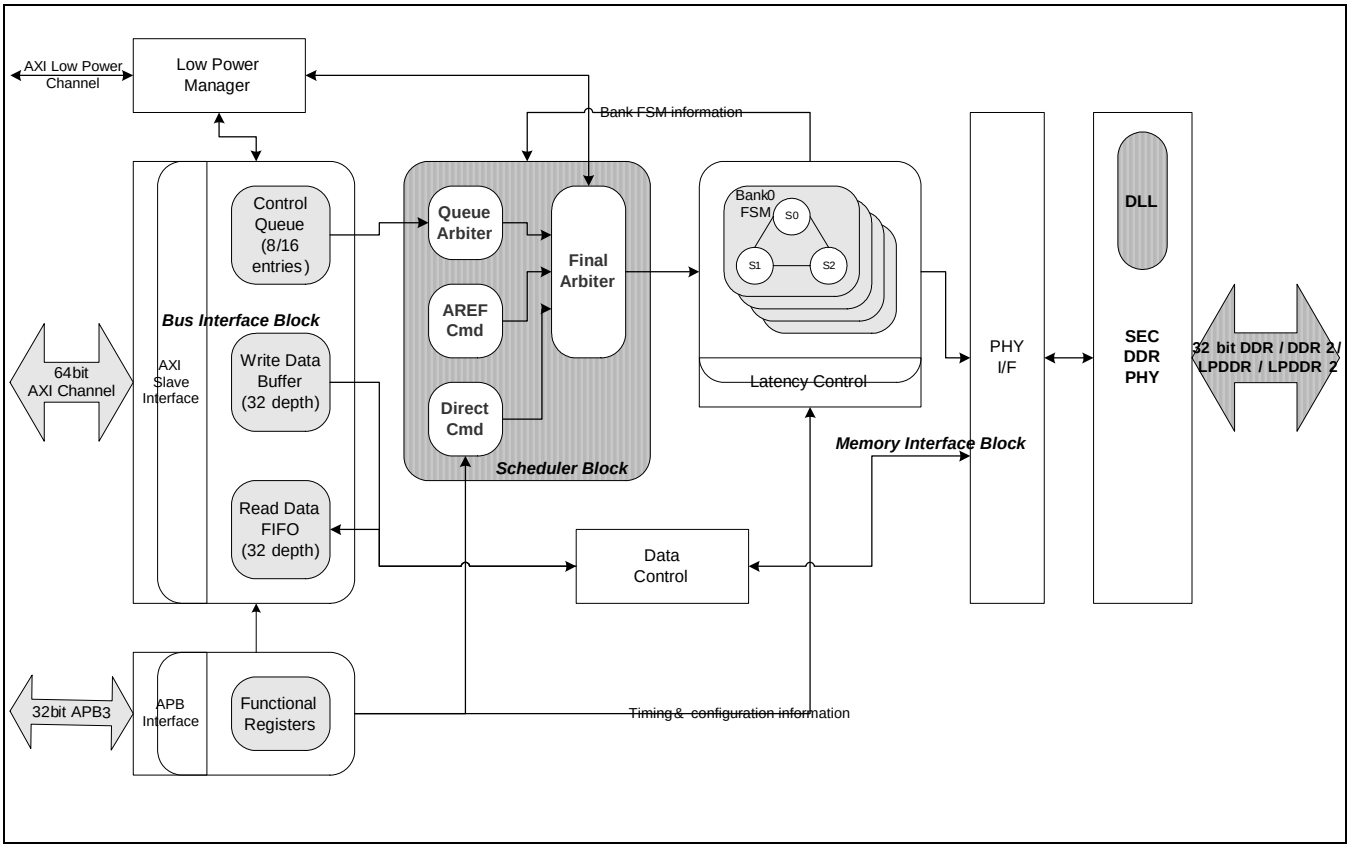


Figure 5.1-1 Overall Block Diagram

Figure 5.1-1 shows the overall block diagram of the controller. The block diagram shows the bus interface block, scheduler block, and memory interface block, which connects and interfaces with the SEC DDR PHY.

The bus interface block saves the bus transactions for memory access that come from the AXI slave port to the command queue. Additionally it saves the write data to the write buffer or sends the read data to the Master via the AXI bus. It also acts as a read FIFO if AXI Master is not ready and has an APB interface for special function registers/ direct commands and an AXI low power channel interface.

The Scheduler block uses the memory bank Finite State Machine (FSM) information to arbitrate the bus transactions in the command queues and transforms the commands into a memory command type, which is sent to the Memory interface block. It also controls the write and read data flow between the memory and the AXI bus.

The Memory interface block updates each memory bank state according to the memory command coming from the scheduler and sends the bank state back to the scheduler. It creates a memory command depending on the memory latency and sends the command to the SEC DDR PHY via the PHY interface.

2 FUNCTIONAL DESCRIPTION

2.1 INITIALIZATION

An Initialization procedure consists of PHY DLL initialization, setting controller register and memory initialization. For memory initialization, please refer to JEDEC specifications and data sheets of memory devices. According to the memory types, initialization sequences are as follows.

2.1.1 mDDR

1. To provide stable power for controller and memory device, the controller must assert and hold CKE to a logic high level. Then apply stable clock. **Note:** XDDR2SEL should be Low level to hold CKE to high.
2. Set the **PhyControl0.ctrl_start_point** and **PhyControl0.ctrl_inc** bit-fields to correct value according to clock frequency. Set the **PhyControl0.ctrl_dll_on** bit-field to '1' to activate the PHY DLL.
3. DQS Cleaning: Set the **PhyControl1.ctrl_shiftc** and **PhyControl1.ctrl_offsetc** bit-fields to the correct value according to clock frequency and memory tAC parameters.
4. Set the **PhyControl0.ctrl_start** bit-field to '1'.
5. Set the **ConControl**. At this moment, an auto refresh counter should be off.
6. Set the **MemControl**. At this moment, all power down modes should be off.
7. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
8. Set the **PrechConfig** and **PwrdnConfig** registers.
9. Set the **TimingAref, TimingRow, TimingData** and **TimingPower** registers according to memory AC parameters.
10. If QoS scheme is required, set the **QosControl0~7** and **QosConfig0~7** registers.
11. You must wait for the **PhyStatus0.ctrl_locked** bit-fields to change to '1'. Check whether PHY DLL is locked.
12. PHY DLL compensates the changes of delay amount caused by Process, Voltage and Temperature (PVT) variation during memory operation. Therefore, it should not be off for reliable operation. It can be off except runs at low frequency. If off mode is used, set the **PhyControl0.ctrl_force** bit-field to the correct value according to the **PhyStatus0.ctrl_lock_value[9:2]** bit-field for fix delay amount. Clear the **PhyControl0.ctrl_dll_on** bit-field to turn off PHY DLL.
13. Confirm whether stable clock issues minimum 200us after power on
14. Issue a ***PALL*** command using the **DirectCmd** register.
15. Issue two ***Auto Refresh*** commands using the **DirectCmd** register.
16. Issue a ***MRS*** command using the **DirectCmd** register to program the operating parameters.
17. Issue an ***EMRS*** command using the **DirectCmd** register to program the operating parameters.
18. If there are two external memory chips, perform steps 14~17 for chip1 memory device.
19. Set the **ConControl to turn on an auto refresh counter.**
20. If power down modes are required, set the **MemControl** registers.

2.1.2 mDDR2

1. To provide stable power for controller and memory device, the controller must assert and hold CKE to a logic low level. Then apply stable clock. **Note:** XDDR2SEL should be High level to hold CKE to low.
2. Set the **PhyControl0.ctrl_start_point** and **PhyControl0.ctrl_inc** bit-fields to correct value according to clock frequency. Set the **PhyControl0.ctrl_dll_on** bit-field to '1' to activate the PHY DLL.
3. DQS Cleaning: Set the **PhyControl1.ctrl_shiftc** and **PhyControl1.ctrl_offsetc** bit-fields to correct value according to clock frequency and memory tAC parameters.
4. Set the **PhyControl0.ctrl_start** bit-field to '1'.
5. Set the **ConControl**. At this moment, an auto refresh counter should be off.
6. Set the **MemControl**. At this moment, all power down modes should be off.
7. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
8. Set the **PrechConfig** and **PwrdnConfig** registers.
9. Set the **TimingAref, TimingRow, TimingData** and **TimingPower** registers according to memory AC parameters.
10. If QoS scheme is required, set the **QosControl0~7** and **QosConfig0~7** registers.
11. You must wait for the **PhyStatus0.ctrl_locked** bit-fields to change to '1'. Check whether PHY DLL is locked.
12. PHY DLL compensates the changes of delay amount caused by Process, Voltage and Temperature (PVT) variation during memory operation. Therefore, it should not be off for reliable operation. It can be off except runs at low frequency. If off mode is used, set the **PhyControl0.ctrl_force** bit-field to correct value according to the **PhyStatus0.ctrl_lock_value[9:2]** bit-field for fix delay amount. Clear the **PhyControl0.ctrl_dll_on** bit-field to turn off PHY DLL.
13. Set the **PhyControl1.fp_resync** bit-field to '1' to update DLL information.
14. Confirm that CKE has been as a logic low level at least 100ns after power on.
15. Issue a ***NOP*** command using the **DirectCmd** register to assert and to hold CKE to a logic high level.
16. Wait for at least 200us.
17. Issue a ***MRS*** command using the **DirectCmd** register to reset memory device and program the operating parameters.
18. Wait for minimum 1us.
19. Issue a ***MRR*** command using the **DirectCmd** register to poll the DAI bit of the **MRStatus** register to know whether Device Auto-Initialization is completed or not.
20. If there are two external memory chips, perform steps 15 ~ 19 for chip1 memory device.
21. Set the **ConControl** to turn on an auto refresh counter.
22. If power down modes is needed, set the **MemControl** register.

2.1.3 DDR2

1. To provide stable power for controller and memory device, the controller must assert and hold CKE to a logic low level. Then apply stable clock. **Note:** XDDR2SEL should be High level to hold CKE to low.
2. Set the **PhyControl0.ctrl_start_point** and **PhyControl0.ctrl_inc** bit-fields to correct value according to clock frequency. Set the **PhyControl0.ctrl_dll_on** bit-field to '1' to turn on the PHY DLL.
3. DQS Cleaning: Set the **PhyControl1.ctrl_shiftc** and **PhyControl1.ctrl_offsetc** bit-fields to correct value according to clock frequency and memory tAC parameters.
4. Set the **PhyControl0.ctrl_start** bit-field to '1'.
5. Set the **ConControl**. At this moment, an auto refresh counter should be off.
6. Set the **MemControl**. At this moment, all power down modes should be off.
7. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
8. Set the **PrechConfig** and **PwrdnConfig** registers.
9. Set the **TimingAref, TimingRow, TimingData** and **TimingPower** registers according to memory AC parameters.
10. If QoS scheme is required, set the **QosControl0~7** and **QosConfig0~7** registers.
11. You must wait for the **PhyStatus0.ctrl_locked** bit-fields to change to '1'. Check whether PHY DLL is locked.
12. PHY DLL compensates the changes of delay amount caused by Process, Voltage and Temperature (PVT) variation during memory operation. Therefore, it should not be off for reliable operation. It can be off except runs at low frequency. If off mode is used, set the **PhyControl0.ctrl_force** bit-field to correct value according to the **PhyStatus0.ctrl_lock_value[9:2]** bit-field for fix delay amount. Clear the **PhyControl0.ctrl_dll_on** bit-field to turn off PHY DLL.
13. Confirm whether stable clock is issued minimum 200us after power on
14. Issue a ***NOP*** command using the **DirectCmd** register to assert and to hold CKE to a logic high level.
15. You must wait for minimum 400ns.
16. Issue a ***PALL*** command using the **DirectCmd** register.
17. Issue an ***EMRS2*** command using the **DirectCmd** register to program the operating parameters.
18. Issue an ***EMRS3*** command using the **DirectCmd** register to program the operating parameters.
19. Issue an ***EMRS*** command using the **DirectCmd** register to enable the memory DLLs.
20. Issue a ***MRS*** command using the **DirectCmd** register to reset the memory DLL.
21. Issue a ***PALL*** command using the **DirectCmd** register.
22. Issues two ***Auto Refresh*** commands using the **DirectCmd** register.
23. Issues a ***MRS*** command using the **DirectCmd** register to program the operating parameters without resetting the memory DLL.
24. Wait for minimum 200 clock cycles.
25. Issue an ***EMRS*** command using the **DirectCmd** register to program the operating parameters. If OCD calibration is not used, issue an ***EMRS*** command to set OCD Calibration Default. After that, issue an ***EMRS*** command to exit OCD Calibration Mode and to program the operating parameters.
26. If there are two external memory chips, perform steps 14~25 for chip1 memory device.
27. Set the **ConControl** to turn on an auto refresh counter.
28. If power down modes are needed, set the **MemControl** register.

2.2 ADDRESS MAPPING

The controller modifies the address of the bus transaction coming from the AXI slave port into a memory address - chip select, bank address, row address, column address and memory data width.

To map chip select0 of the memory device to a specific area of the address map, the **chip_base** and **chip_mask** bit-fields of the **MemConfig0** register needs to be set (Refer to Register Descriptions). If chip1 of the memory device exists, the **MemConfig1** register must also be set.

Then, the AXI address requested by the AXI Master is divided into AXI base address and AXI offset address. The AXI base address activates the appropriate memory chip select and the AXI offset address is mapped to a memory address according to the bank, row, column number, and data width set by the **MemConfig** register.

There are two ways to map the AXI offset address as shown below: 1) Linear mapping 2) Interleaved mapping.

2.2.1 Linear Mapping

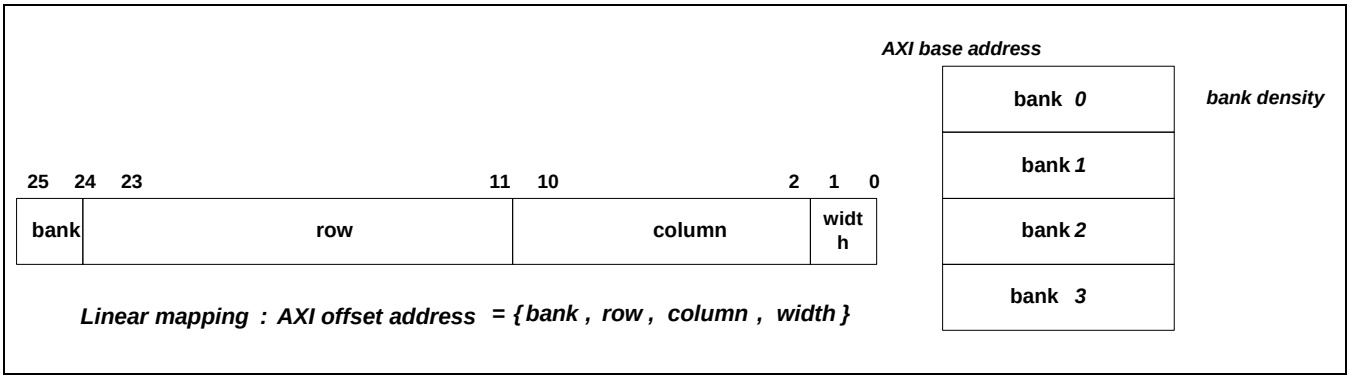


Figure 5.1-2 Linear Address Mapping

As shown in Figure 5.1-2, the linear mapping method maps the AXI address in the order of bank, row, column and width. Since the bank address does not change for at least one bank size, applications that use linear address mapping have a high possibility to access the same bank.

2.2.2 Interleaved Mapping

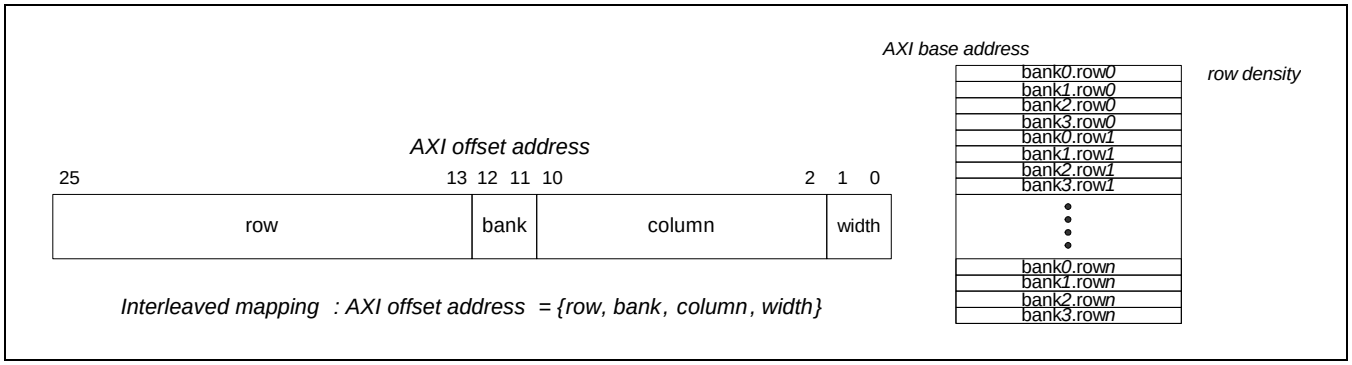


Figure 5.1-3 Interleaved Address Mapping

As shown in Figure 5.1-3, the interleaved mapping method maps the AXI address in the order of row, bank, column and width. The difference between the linear mapping method and the interleaved method is that the bank and row order is different. For accesses beyond a row size, interleaved mapping accesses a different bank. Therefore, applications that use interleaved mapping access numerous banks. It causes better performance but more power consumption.

2.3 LOW POWER OPERATION

The controller executes a low power memory operation in five ways as described below. Each feature is independent of each other and executed at the same time.

2.3.1 AXI low Power Channel

The controller has an AXI low power channel interface to communicate with low power management units such as the system controller, which makes the memory device go into self-refresh mode. To request through the AXI low power channel, refer the **chip1_empty** and **chip0_empty** bit-fields of **ConControl** register to check if the command queue is currently empty.

2.3.2 Dynamic Power Down

The SDRAM device has an active/ precharge power down mode. This mode is entered if CKE becomes LOW. To enter active power down mode minimum one row of a bank must be open. To enter precharge power down mode CKE must be low.

If no AXI transaction enters the controller and the command queue becomes empty for a specific number of cycles (**PwrdnConfig.dpwrdn_cyc** bit-field), the controller changes the memory device's state to active/ precharge power down automatically. Then, there are two ways to enter the active/ precharge power down state and It is selected by **MEMCONTROL.dpwrdn_type** bit.

1. Active/ precharge power down mode: Enter power down without considering whether there is a row open or not,
2. Force precharge power down mode: Enter power down after closing all banks.

If a new AXI transaction enters the controller, the controller automatically wakes up the memory device from power down state and executes in a normal operation state.

2.3.3 Dynamic Self Refresh

Similar to the dynamic power down feature (Refer to Section 2.3.2 Dynamic Power Down), if the command queue is empty for a specific amount of cycles (**PwrdnConfig.dsref_cyc** bit-field**)**, the memory device enters self-refresh mode. Since exiting power down mode requires many cycles, we recommend to choose a greater cycle size for dynamic self-refresh entry than dynamic power down.

2.3.4 Clock Stop

To reduce the I/O power of the memory device and the controller, it is possible to stop the clock if the LPDDR / LPDDR2 is in idle mode, or self refresh mode and DDR2 is in self refresh mode. If this feature is enabled, the controller automatically executes the clock stop feature.

2.3.5 Direct Command

Use the direct command feature to send a command to the memory device through the APB3 port. This way, you force the memory device to enter active/ precharge power down, self-refresh or deep power down mode

2.4 PRECHARGE POLICY

There are two ways for the controller to decide precharge policy.

2.4.1 Bank Selective Precharge Policy

Since applications have different page policy preferences, it is hard for the engineer to decide whether to use open page policy, or close page (auto precharge) policy. Instead of applying the page policy to all of the banks, the bank selective precharge policy allows the user to choose a precharge policy for each bank Rrefer to **PrechConfig.chip1_policy**). This way, you assign certain applications to a bank that uses an open page policy, and other applications to a bank that uses a close page (auto precharge) policy.

Open Page Policy: After a READ or WRITE, the accessed row is left open.

Close Page (Auto Precharge) Policy: Right after a READ or WRITE command, the controller issues an auto precharge to the bank.

2.4.2 Timeout Precharge

If a certain bank uses an open page policy, the row is left open after a data access. If this happens and the bank that is left open is not scheduled for a specific number of cycles (**PrechConfig.tp_cnt** bit-field) the controller automatically issues a precharge command to close the bank.

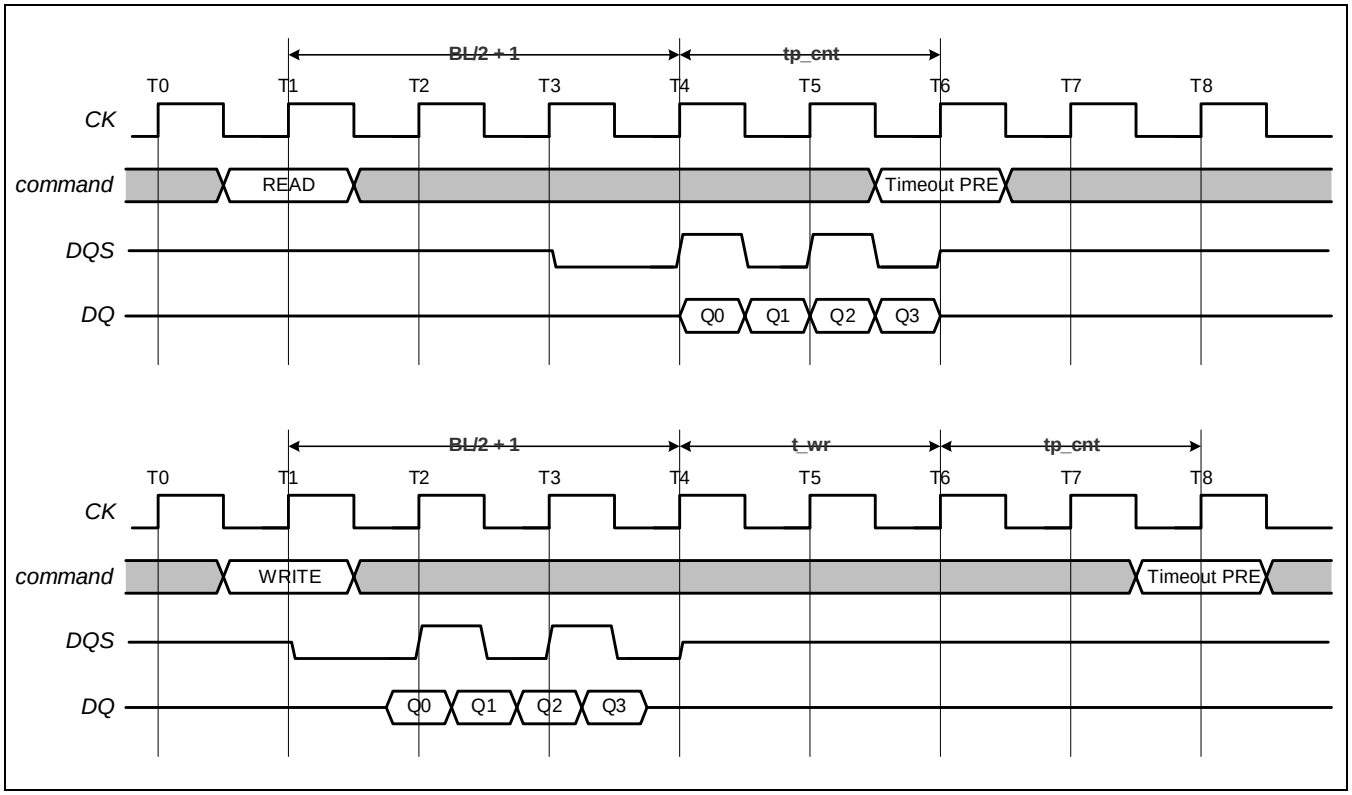


Figure 5.1-4 Timing Diagram of Timeout Precharge

2.5 QUALITY OF SERVICE

QoS is defined for the CONTROLLER as a method to increase the arbitration priority of a master that requires low latency read data. The QoS is determined if the control queue (Refer to Figure 5.1-1) receives the command through the AXI bus and the QoS count starts depreciating at this moment. If the count reaches zero, this command becomes the highest priority among the other commands that are in the control queue.

There are two types of QoS.

- qos_cnt
- default_qos

2.5.1 qos_cnt

There are 8 configurable QoSControls applicable to Read data transfers, which have independent qos_masks that mask the ARID from one bit up to the ARID width. All 8 QoSControls are either enabled or disabled,

1. If the command is received via the AXI bus, the ARID is masked by the qos_masks (**QoSConfig**(**n**)**.qos_mask**) from the 8 QoSControls that are enabled.
2. The masked results are then compared to the qos_ids (**QoSConfig**(**n**)**.qos_id**). If any one of the result are equal, the qos_cnt (**QoSControl**(**n**).**qos_cnt**) value is applied to the command and saved in the control queue.

2.5.2 Default QoS

If qos_cnt is not applied to the command, a default QoS counter (Is set to a different value by modifying **ConControl.timeout_cnt**) is applied to the command. (Default QoS counter is applied to both read and write)

2.6 READ DATA CAPTURE

A memory device that receives a read command sends the data to the controller after a read latency (i.e. CAS latency). After clearing the DQS, the PHY uses the PHY DLL to phase shift the DQS 90 degrees. Using the shifted DQS, the PHY samples the read data and saves the data into the read data input FIFO, which is located inside the PHY. Then, the controller fetches the data from the PHY while considering the read latency and the read fetch delay, and then sends it to the AXI read channel. The following figures show the read data capture process's timing diagram for each memory type.

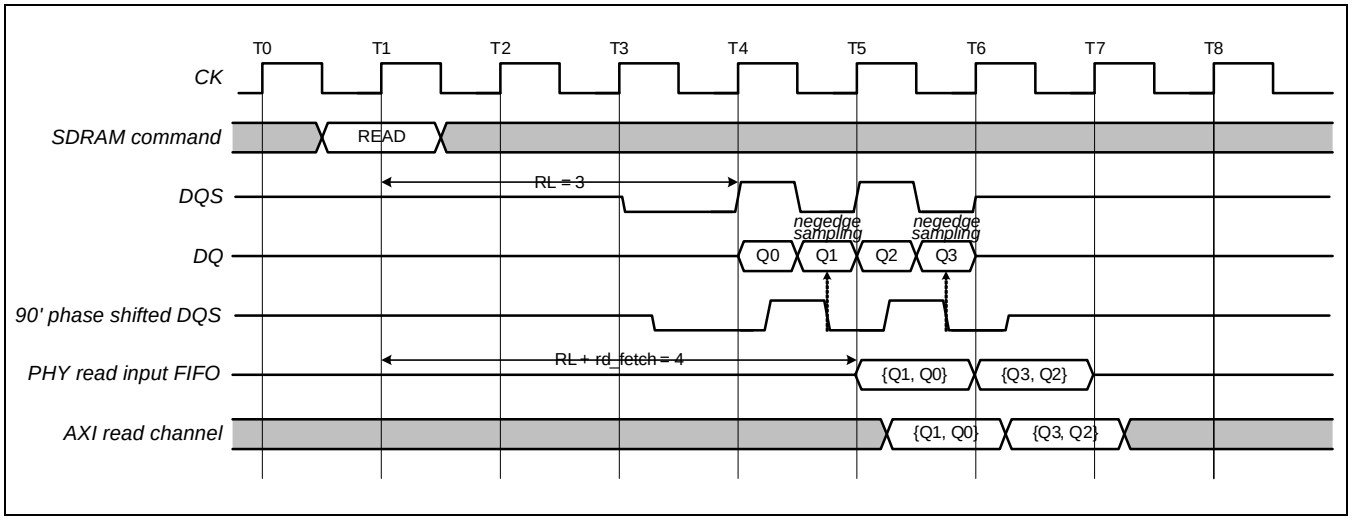


Figure 5.1-5 Timing Diagram of Read Data Capture (DDR2, zero delay, RL=3, rd_fetch=1)

Figure 5.1-5 is for DDR2 having an internal DLL. An internal DLL exists which allows it to send the data after an exact amount of read latency. If we assume there are minimal or no board/ PHY input delay, if sampling the negedge (Q1, Q3 sampling), since the data gets saved into the PHY read data input FIFO, the controller sends the read data to the AXI read channel in 'read latency + 1(read fetch)' cycles. The read fetch cycle is set using the **ConControl.rd_fetch** bit-field.

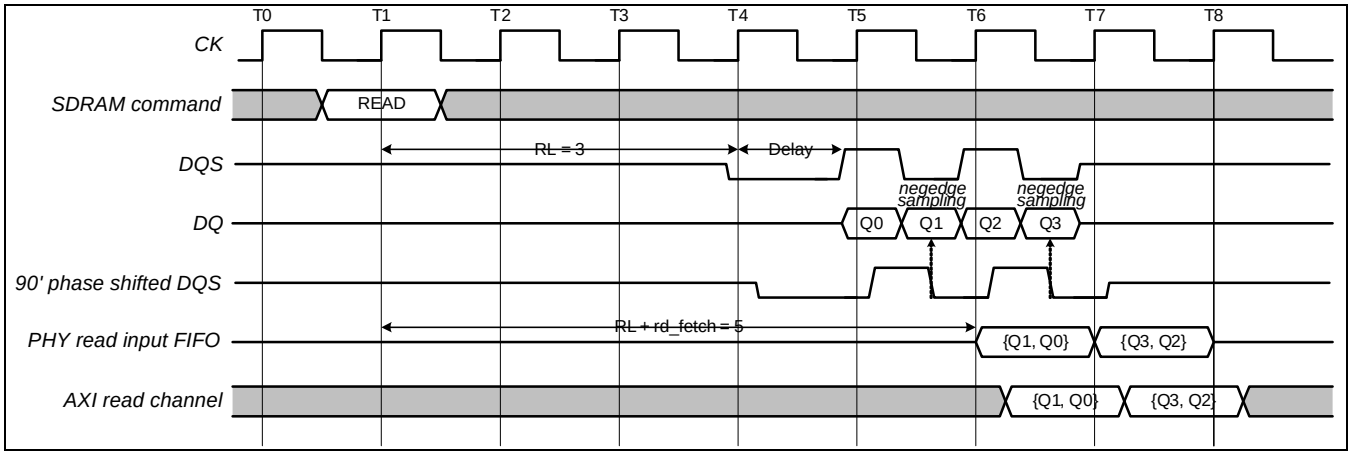


Figure 5.1-6 Timing Diagram of Read Data Capture (DDR2, non-zero delay, RL=3, rd_fetch=2)

Figure 5.1-6 is different from Figure 5.1-5 because a delay exists. Negedge sampling happens at T5 and T6, which is one cycle slower than T4/T5 shown in Figure 5.1-4. Therefore, the read fetch cycle should be set to two since the sampled read data is saved into the read input FIFO slower.

To calculate the DDR2 rd_fetch value:

rd_fetch DDR2) = INT((Delay + 0.5T + 0.25T)/T) = INT(Delay/T + 0.75),

Delay: board delay + PHY input/output delay, T: clock period, INT(x): the rounded-up integer value of x

Therefore, rd_fetch must have minimum one value.

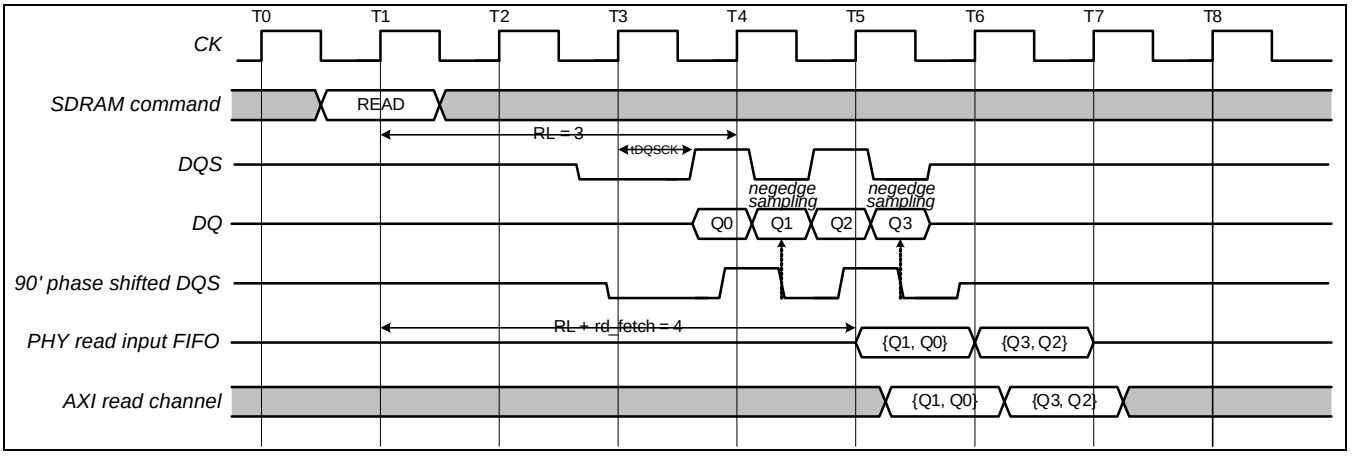


Figure 5.1-7 Timing Diagram of Read Data Capture (LPDDR/LPDDR2, zero delay, RL=3, rd_fetch=1)

An LPDDR/LPDDR2 does not have an internal DLL. Without an internal DLL as you may see in Figure 5.1-7, the data is sent out after tDQSCK before the read latency is over. Even if we assume zero delay, since tDQSCK becomes relatively large in high frequencies, the read fetch cycle should be set to one.

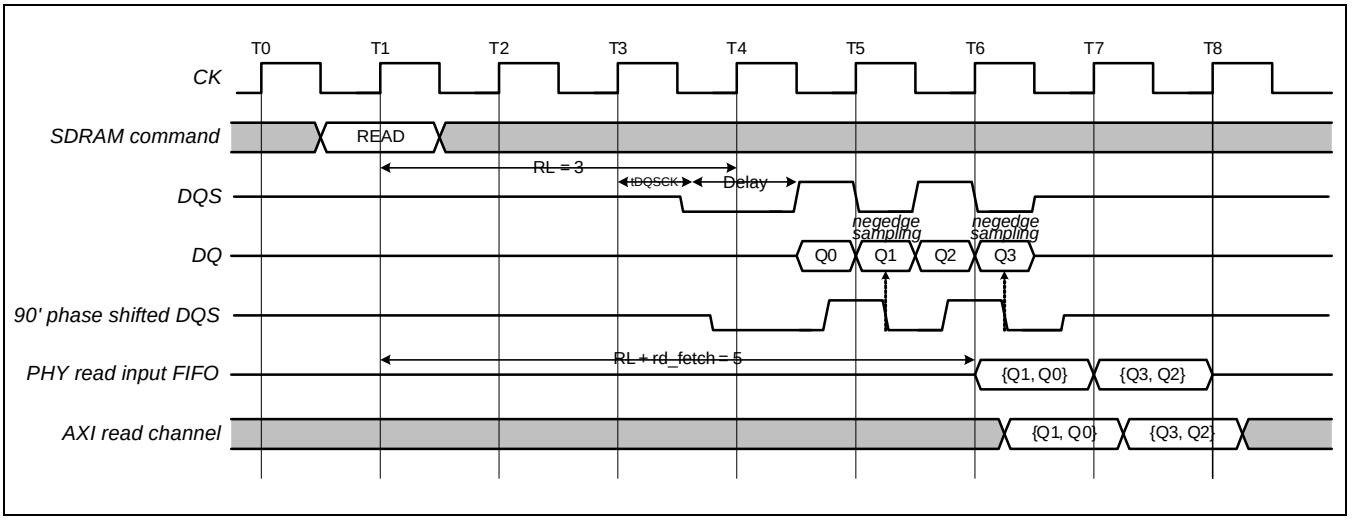


Figure 5.1-8 Timing Diagram of Read Data Capture (LPDDR/LPDDR2, non-zero delay, RL=3, rd_fetch=2)

If a delay exists such as Figure 5.1-8, a bigger value should be assigned to rd_fetch.

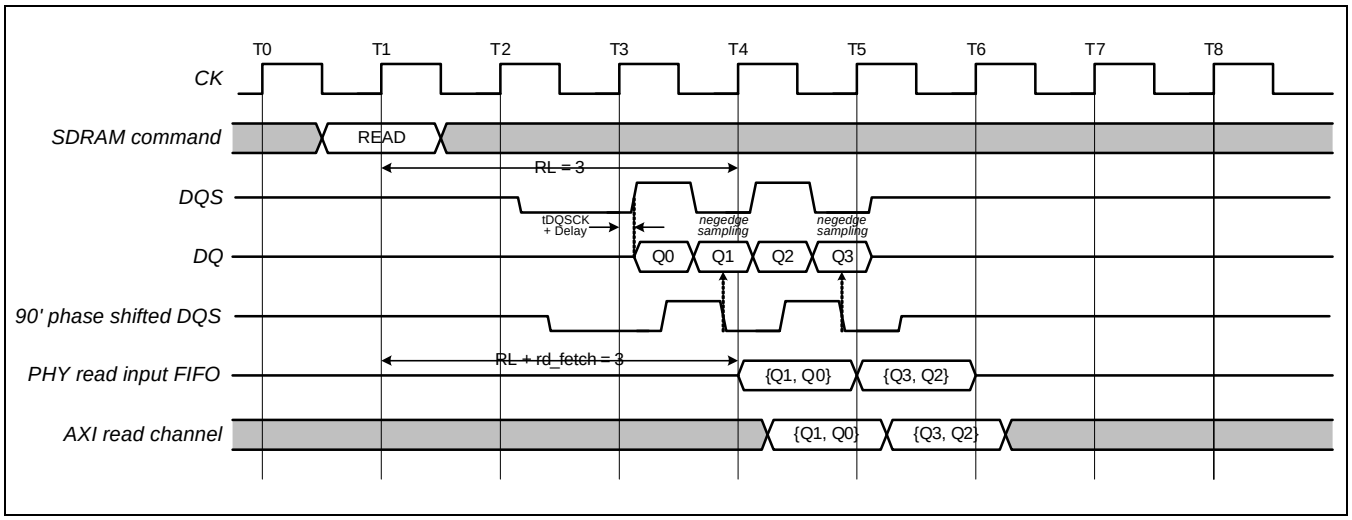


Figure 5.1-9 Timing Diagram of Read Data Capture (LPDDR/LPDDR2, low frequency, RL=3, rd_fetch=0)

tDQSCK + Delay is relatively small compared to the clock period during low frequencies as shown in Figure 5.1-9. In this situation, negedge sampling happens before read latency and therefore read fetch is set to zero.

To calculate the LPDDR/LPDDR2 rd_fetch value:

rd_fetch (LPDDR/LPDDR2) = INT((-1 + Delay + 0.5T + 0.25T)/T) = INT(Delay/T - 0.25),

Delay: board delay + PHY input delay, T: clock period, INT(x): the rounded-up integer value of x

Therefore, if the value of Delay/T is less than 0.25, rd_fetch is set to zero

3 I/O DISCRIPTION

Function Signal	I/O	Description	PAD	Type
DDR2SEL	I	Memory Type Selection (0; LPDDR1, 1: DDR2, LPDDR2)	XDDR2SEL	dedicated
DDR_CSn[1:0]	O	Memory Chip Select	Xm1CSn[1:0]	dedicated
DDR_SCLK	O	Memory Clock	Xm1SCLK	dedicated
DDR_nSCLK	O	Memory Negative Clock	Xm1nSCLK	dedicated
DDR_RASn	O	Row Address Selection	Xm1RASn	dedicated
DDR_CASn	O	Column Address Selection	Xm1CASn	dedicated
DDR_WEn	O	Write Enable	Xm1WEn	dedicated
DDR_A[31:0]	I/O	Memory Address Bus	Xm1ADDR[31:0]	dedicated
DDR_D[31:0]	I/O	Memory Data Bus	Xm1DATA[31:0]	dedicated
DDR_DQM[3:0]	O	Write Masking Per Byte	Xm1DQM[3:0]	dedicated
DDR_DQS[3:0]	I/O	Data Strobe Signal Per Byte	Xm1DQS[3:0]	dedicated
DDR_DQSn[3:0]	I/O	Data Strobe Negative Signal Per Byte	Xm1DQSn[3:0]	dedicated
DDR_CKE[1:0]	O	Memory Address, Bank Address, CS, CKE signals	* refer to 3.1 table	dedicated

3.1 PAD MUX FOR DRAM TYPE

PAD Name	mDDR			LPDDR2	DDR2				
	64MB	128MB	256MB	256MB	64MB	128MB	256MB	512MB	1GB
Xm1ADDR[0]	ADDR_0	ADDR_0	ADDR_0	CA_0	ADDR_0	ADDR_0	ADDR_0	ADDR_0	ADDR_0
Xm1ADDR[1]	ADDR_1	ADDR_1	ADDR_1	CA_1	ADDR_1	ADDR_1	ADDR_1	ADDR_1	ADDR_1
Xm1ADDR[2]	ADDR_2	ADDR_2	ADDR_2	CA_2	ADDR_2	ADDR_2	ADDR_2	ADDR_2	ADDR_2
Xm1ADDR[3]	ADDR_3	ADDR_3	ADDR_3	CA_3	ADDR_3	ADDR_3	ADDR_3	ADDR_3	ADDR_3
Xm1ADDR[4]	ADDR_4	ADDR_4	ADDR_4	CA_4	ADDR_4	ADDR_4	ADDR_4	ADDR_4	ADDR_4
Xm1ADDR[5]	ADDR_5	ADDR_5	ADDR_5	CA_5	ADDR_5	ADDR_5	ADDR_5	ADDR_5	ADDR_5
Xm1ADDR[6]	ADDR_6	ADDR_6	ADDR_6	CA_6	ADDR_6	ADDR_6	ADDR_6	ADDR_6	ADDR_6
Xm1ADDR[7]	ADDR_7	ADDR_7	ADDR_7	CA_7	ADDR_7	ADDR_7	ADDR_7	ADDR_7	ADDR_7
Xm1ADDR[8]	ADDR_8	ADDR_8	ADDR_8	CA_8	ADDR_8	ADDR_8	ADDR_8	ADDR_8	ADDR_8
Xm1ADDR[9]	ADDR_9	ADDR_9	ADDR_9	CA_9	ADDR_9	ADDR_9	ADDR_9	ADDR_9	ADDR_9
Xm1ADDR[10]	ADDR_10	ADDR_10	ADDR_10		ADDR_10	ADDR_10	ADDR_10	ADDR_10	ADDR_10
Xm1ADDR[11]	ADDR_11	ADDR_11	ADDR_11		ADDR_11	ADDR_11	ADDR_11	ADDR_11	ADDR_11
Xm1ADDR[12]		ADDR_12	ADDR_12		ADDR_12	ADDR_12	ADDR_12	ADDR_12	ADDR_12
Xm1ADDR[13]			ADDR_13					ADDR_13	ADDR_13
Xm1ADDR[14]	BA_0	BA_0	BA_0		BA_0	BA_0	BA_0	BA_0	BA_0
Xm1ADDR[15]	BA_1	BA_1	BA_1		BA_1	BA_1	BA_1	BA_1	BA_1
Xm1CSn[1]	CS_1	CS_1	CS_1	CS_1	CS_1	CS_1	BA_2[1)]	BA_2[1)]	BA_2[1)]
Xm1CSn[0]	CS_0	CS_0	CS_0	CS_2	CS_0	CS_0	CS_0	CS_0	CS_0
Xm1CKE[1]	CKE_1	CKE_1	CKE_1	CKE_1	CKE_1	CKE_1			ADDR_14[2)]
Xm1CKE[0]	CKE_0	CKE_0	CKE_0	CKE_0	CKE_0	CKE_0	CKE_0	CKE_0	CKE_0

NOTES:

1. If Number of Banks (**MEMCONFIGn.chip_bank**) is set 8 banks, Xm1CSn[1] is BA[2] bit.
2. If Number of Row Address Bits(**MEMCONFIGn.chip_row**) is set 15bits, Xm1CKE[1] is Addr[14].

Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

4 REGISTER DESCRIPTION

4.1 REGISTER OVERVIEW

Register	Address	R/W	Description	Reset Value
CONCONTROL	0xE600_0000	R/W	Controller Control Register	0x0FFF1310
MEMCONTROL	0xE600_0004	R/W	Memory Control Register	0x00202100
MEMCONFIG0	0xE600_0008	R/W	Memory Chip0 Configuration Register	0x20F80312
MEMCONFIG1	0xE600_000C	R/W	Memory Chip1 Configuration Register	0x28F80312
DIRECTCMD	0xE600_0010	R/W	Memory Direct Command Register	0x00000000
PRECHCONFIG	0xE600_0014	R/W	Precharge Policy Configuration Register	0xFF000000
PHYCONTROL0	0xE600_0018	R/W	PHY Control0 Register	0x00000000
PHYCONTROL1	0xE600_001C	R/W	PHY Control1 Register	0x00000040
PHYCONTROL2	0xE600_0020	R/W	PHY Control2 Register	0x00000000
PWRDNCONFIG	0xE600_0028	R/W	Dynamic Power Down Configuration Register	0xFFFF00FF
TIMINGAREF	0xE600_0030	R/W	AC Timing Register for SDRAM Auto Refresh	0x0000040E
TIMINGROW	0xE600_0034	R/W	AC Timing Register for SDRAM Row	0x0F233286
TIMINGDATA	0xE600_0038	R/W	AC Timing Register for SDRAM Data	0x12130204
TIMINGPOWER	0xE600_003C	R/W	AC Timing Register for Power Mode of SDRAM	0x0E1B0422
PHYSTATUS0	0xE600_0040	R	PHY Status Register 0	0x0000000X
PHYSTATUS1	0xE600_0044	R	PHY Status Register 1	0x00000000
CHIP0STATUS	0xE600_0048	R	Memory Chip0 Status Register	0x00000000
CHIP1STATUS	0xE600_004C	R	Memory Chip1 Status Register	0x00000000
AREFSTATUS	0xE600_0050	R	Counter Status Register for Auto Refresh	0x0000FFFF
MRSTATUS	0xE600_0054	R	Memory Mode Registers Status Register	0x00000000
PHYTEST0	0xE600_0058	R/W	PHY Test Register 0	0x00000000
PHYTEST1	0xE600_005C	R	PHY Test Register 1	0x00000000
QOSCONTROL0	0xE600_0060	R/W	Quality of Service Control Register 0	0x00000000
QOSCONFIG0	0xE600_0064	R/W	Quality of Service Configuration Register 0	0x00000000
QOSCONTROL1	0xE600_0068	R/W	Quality of Service Control Register 1	0x00000000
QOSCONFIG1	0xE600_006C	R/W	Quality of Service Configuration Register 1	0x00000000
QOSCONTROL2	0xE600_0070	R/W	Quality of Service Control Register 2	0x00000000
QOSCONFIG2	0xE600_0074	R/W	Quality of Service Configuration Register 2	0x00000000
QOSCONTROL3	0xE600_0078	R/W	Quality of Service Control Register 3	0x00000000
QOSCONFIG3	0xE600_007C	R/W	Quality of Service Configuration Register 3	0x00000000
QOSCONTROL4	0xE600_0080	R/W	Quality of Service Control Register 4	0x00000000
QOSCONFIG4	0xE600_0084	R/W	Quality of Service Configuration Register 4	0x00000000
QOSCONTROL5	0xE600_0088	R/W	Quality of Service Control Register 5	0x00000000
QOSCONFIG5	0xE600_008C	R/W	Quality of Service Configuration Register 5	0x00000000
QOSCONTROL6	0xE600_0090	R/W	Quality of Service Control Register 6	0x00000000
QOSCONFIG6	0xE600_0094	R/W	Quality of Service Configuration Register 6	0x00000000
QOSCONTROL7	0xE600_0098	R/W	Quality of Service Control Register 7	0x00000000
QOSCONFIG7	0xE600_009C	R/W	Quality of Service Configuration Register 7	0x00000000

4.2 DETAILED DESCRIPTION

4.2.1 Controller Control Register (ConControl, R/W, Address=0xE600_0000)

CONCONTROL	Bit	Description	R/W	Reset Value
Reserved	[31:28]	Should be zero		0x0
timeout_cnt	[27:16]	**Default Timeout Cycles** 0xn = n aclk cycles (aclk: AXI clock) This counter prevents transactions in command queue from starvation. This counter starts if a new AXI transaction comes into a queue. If the counter becomes zero, the corresponding transaction becomes the highest priority command of all the transactions in the command queue. This is a default timeout counter and overridden by the QoS counter if the ARID matched with the QoS ID comes into the command queue. Refer to "Section 2.5 Quality of Service".	R/W	0xFFF
rd_fetch	[15:12]	**Read Data Fetch Cycles** 0xn = n mclk cycles (mclk: Memory clock) This register is for the unpredictable latency of read data coming from memory devices by tDQSCK variation or the board flying time. The read fetch delay of PHY read FIFO must be controlled by this parameter. The controller will fetch read data from PHY after read_latency + n mclk cycles. Refer to "Section 2.56 Read Data Capture".	R/W	0x1
Reserved	[11]	Should be zero		0x0
dq_swap	[10]	**DQ Swap** 0x0 = Disable, 0x1 = Enable, If enabled, the controller reverses the bit order of memory data pins. (For example, DQ[31] <-> DQ[0], DQ[30] <-> DQ[1])	R/W	0x0
chip1_empty	[9]	**Command Queue Status of Chip1** 0x0 = Not Empty, 0x1 = Empty There is no AXI transaction corresponding to chip1 memory in the command queue entries	R	0x1
chip0_empty	[8]	**Command Queue Status of Chip0** 0x0 = Not Empty, 0x1 = Empty There is no AXI transaction corresponding to chip0 memory in the command queue entries	R	0x1
drv_en	[7]	**PHY Driving** 0x0 = Disable, 0x1 = Enable During the high-Z state of the memory bidirectional pins, PHY drives these pins with the zeros or pull down these pins for preventing current leakage. Set **PhyControl1.drv_type** register to select driving type.	R/W	0x0

CONCONTROL	Bit	Description	R/W	Reset Value
ctc_rtr_gap_en	[6]	**Read Cycle Gap for Two Different Chips** 0x0 = Disable, 0x1 = Enable To prevent collision between reads from two different memory devices, a one-cycle gap is required. Enable this register to insert the gap automatically for continuous reads from two different memory devices.	R/W	0x0
aref_en	[5]	**Auto Refresh Counter** 0x0 = Disable, 0x1 = Enable Enable this to decrease the auto refresh counter by 1 at the rising edge of the mclk.	R/W	0x0
out_of	[4]	**Out of Order Scheduling** 0x0 = Disable, 0x1 = Enable The embedded scheduler enables out-of order operation to improve SDRAM utilization	R/W	0x1
clk_ratio	[3:1]	**Clock Ratio of Bus Clock to Memory Clock** 0x0 = freq.(aclk) : freq.(mclk) = 1 : 1, 0x1 ~ 0x7 = Reserved	R/W	0x0
Reserved	[0]	Should be zero		0x0

4.2.2 Memory Control Register (MemControl, R/W, Address=0xE600_0004)

MEMCONTROL	Bit	Description	R/W	Reset Value
Reserved	[31:23]	Should be zero		0x0
bl	[22:20]	**Memory Burst Length** 0x0 = Reserved, 0x1 = 2, 0x2 = 4, 0x3 = 8, 0x4 = 16, 0x5 ~ 0x7 = Reserved In case of DDR2/LPDDR2, the controller only supports burst length 4.	R/W	0x2
num_chip	[19:16]	**Number of Memory chips** 0x0 = 1 chip, 0x1 = 2 chips, 0x2 ~ 0xf = Reserved	R/W	0x0
mem_width	[15:12]	**Width of Memory Data Bus** 0x0 ~ 0x1 = Reserved, 0x2 = 32-bit, 0x3 ~ 0xf = Reserved	R/W	0x2
mem_type	[11:8]	**Type of Memory** 0x0 = Reserved, 0x1 = LPDDR 0x2 = LPDDR2, 0x3 = Reserved 0x4 = DDR2, 0x5 ~ 0xf = Reserved	R/W	0x1
add_lat_pall	[7:6]	**Additional Latency for PALL** 0x0 = 0 cycle, 0x1 = 1 cycle 0x2 = 2 cycle, 0x3 = 3 cycle If all banks precharge command is issued, the latency of precharging will be tRP + add_lat_pall	R/W	0x0
dsref_en	[5]	**Dynamic Self Refresh** 0x0 = Disable, 0x1 = Enable Refer to "Section 2.3.3. Dynamic Self Refresh"	R/W	0x0
tp_en	[4]	**Timeout Precharge** 0x0 = Disable, 0x1 = Enable If tp_en is enabled it automatically precharges an open bank after a specified amount of mclk cycles (if no access has been made in between the cycles) in an open page policy. If **PrechConfig.tp_cnt** bit-field is set, it specifies the amount of	R/W	0x0

MEMCONTROL	Bit	Description	R/W	Reset Value
		mclk cycles to wait until timeout precharge precharges the open bank. Refer to "Section 2.4.2. Timeout Precharge".		
dpwrdn_type	[3:2]	**Type of Dynamic Power Down** 0x0 = Active/ Precharge power down, 0x1 = Force precharge power down 0x2 ~ 0x3 : Reserved Refer to "Section 2.3.2. Dynamic Power Down".	R/W	0x0
dpwrdn_en	[1]	**Dynamic Power Down** 0x0 = Disable, 0x1 = Enable	R/W	0x0
clk_stop_en	[0]	**Dynamic Clock Control** 0x0 = Always running, 0x1 = Stops during idle periods Refer to "Section 2.3.4. Clock Stop".	R/W	0x0

4.2.3 Memory Chip0 Configuration Register (MemConfig0, R/W, Address=0xE600_0008)

MEMCONFIG0	Bit	Description	R/W	Reset Value
chip_base	[31:24]	**AXI Base Address** AXI base address [31:24] = chip_base, For example, if chip_base = 0x20, then AXI base address of memory chip0 becomes 0x2000_0000.	R/W	0x20
chip_mask	[23:16]	**AXI Base Address Mask** Upper address bit mask to determine AXI offset address of memory chip0. 0 = Corresponding address bit is not to be used for comparison 1 = Corresponding address bit is to be used for comparison This bit field is used to check whether 'accessed address' & 'mask address' is equal to 'base address'. For example, if AXI base address of memory chip0 is 0x2000_0000 and AXI Base address mask is 0xF8, then memory chip0 has an address range of 0x2000_0000 ~ 0x27FF_FFFF.	R/W	0xF8
chip_map	[15:12]	**Address Mapping Method (AXI to Memory)** 0x0 = Linear ({bank, row, column, width}), 0x1 = Interleaved ({row, bank, column, width}), 0x2 ~ 0xf = Reserved	R/W	0x0
chip_col	[11:8]	**Number of Column Address Bits** 0x0 = 7 bits, 0x1 = 8 bits, 0x2 = 9 bits, 0x3 = 10 bits, 0x4 ~ 0xf = Reserved	R/W	0x3
chip_row	[7:4]	**Number of Row Address Bits** 0x0 = 12 bits, 0x1 = 13 bits, 0x2 = 14 bits, 0x3 = 15 bits, 0x4 ~ 0xf = Reserved	R/W	0x1
chip_bank	[3:0]	**Number of Banks** 0x0 = 1 bank, 0x1 = 2 banks, 0x2 = 4 banks, 0x3 = 8 banks, 0x4 ~ 0xf = Reserved	R/W	0x2

4.2.4 Memory Chip1 Configuration Register (MemConfig1, R/W, Address=0xE600_000C)

MEMCONFIG1	Bit	Description	R/W	Reset Value
chip_base	[31:24]	**AXI Base Address** AXI base address [31:24] = chip_base, For example, if chip_base = 0x28, then AXI base address of chip1 becomes 0x2800_0000.	R/W	0x28
chip_mask	[23:16]	**AXI Base Address Mask** Upper address bit mask to determine AXI offset address of memory chip1. 0 = Corresponding address bit is not to be used for comparison 1 = Corresponding address bit is to be used for comparison This bit field is used to check whether 'accessed address' & 'mask address' is equal to 'base address'. For example, if chip_mask = 0xF8, then AXI offset address becomes 0x0000_0000 ~ 0x07FF_FFFF. If AXI base address of memory chip1 is 0x2800_0000, then memory chip1 has an address range of 0x2800_0000 ~ 0x2FFF_FFFF.	R/W	0xF8
chip_map	[15:12]	**Address Mapping Method (AXI to memory)** 0x0 = Linear ({bank, row, column, width}), 0x1 = Interleaved ({row, bank, column, width}), 0x2 ~ 0xf = Reserved	R/W	0x0
chip_col	[11:8]	**Number of Column Address Bits** 0x0 = 7 bits, 0x1 = 8 bits, 0x2 = 9 bits, 0x3 = 10 bits, 0x4 ~ 0xf = Reserved	R/W	0x3
chip_row	[7:4]	**Number of Row Address Bits** 0x0 = 12 bits, 0x1 = 13 bits, 0x2 = 14 bits, 0x3 = 15 bits, 0x4 ~ 0xf = Reserved	R/W	0x1
chip_bank	[3:0]	**Number of Banks** 0x0 = 1 bank, 0x1 = 2 banks, 0x2 = 4 banks, 0x3 = 8 banks, 0x4 ~ 0xf = Reserved	R/W	0x2

4.2.5 Memory Direct Command Register (DirectCmd, R/W, Address=0xE600_0010)

DIRECTCMD	Bit	Description	R/W	Reset Value
Reserved	[31:28]	Should be zero.		0x0
cmd_type	[27:24]	**Type of Direct Command** 0x0 = MRS/EMRS (mode register setting), 0x1 = PALL (all banks precharge), 0x2 = PRE (per bank precharge), 0x3 = DPD (deep power down), 0x4 = REFS (self refresh), 0x5 = REFA (auto refresh), 0x6 = CKEL (active/ precharge power down), 0x7 = NOP (exit from active/ precharge power down or deep power down, 0x8 = REFSX (exit from self refresh) 0x9 = MRR (mode register reading), 0xa ~ 0xf = Reserved If a direct command is issued, AXI masters must not access memory. It is strongly recommended to check the command queue's state by **Concontrol.chip0/1_empty** before issuing a direct command You must disable dynamic power down, dynamic self refresh and force precharge function (**MemControl** register). MRS/EMRS and MRR commands should be issued if all banks are in idle state. If MRS/EMRS and MRR is issued to LPDDR2, the CA pins must be mapped as follows. MA[7:0] = {cmd_addr[1:0], cmd_bank[2:0], cmd_addr[12:10]}, OP[7:0] = cmd_addr[9:2]	R/W	0x0
Reserved	[23:21]	Should be zero.		0x0
cmd_chip	[20]	**Chip Number to send the direct command to** 0 = Chip 0 1 = Chip 1	R/W	0x0
Reserved	[19]	Should be zero.		0x0
cmd_bank	[18:16]	**Related Bank Address when issuing a direct command** To send a direct command to a chip, additional information such as the bank address is required. This register is used in such situations.	R/W	0x0
Reserved	[15]	Should be zero.		0x0
cmd_addr	[14:0]	**Related Address Value when issuing a direct command** To send a direct command to a chip, additional information such as the address is required. This register is used in such situations.	R/W	0x0

4.2.6 Precharge Policy Configuration Register (PrechConfig, R/W, Address=0xE600_0014)

PRECHCONFIG	Bit	Description	R/W	Reset Value
tp_cnt	[31:24]	**Timeout Precharge Cycles** 0xn = n mclk cycles, If the timeout precharge function (**MemControl.tp_en**) is enabled and the timeout precharge counter becomes zero, the controller forces the activated memory bank into the precharged state. Refer to "Section 2.4.2. Timeout Precharge".	R/W	0xFF
Reserved	[23:16]	Should be zero		0x0
chip1_policy	[15:8]	**Memory Chip1 Precharge Bank Selective Policy** 0x0 = Open page policy, 0x1 = Close page (auto precharge) policy chip1_policy[n], n is the bank number of chip1. Open Page Policy: After a READ or WRITE, the row accessed before is left open. Close Page (Auto Precharge) Policy: Right after a READ or WRITE command, memory devices automatically precharges the bank. This is a bank selective precharge policy. For example, if chip1_policy[2] is 0x0, bank2 of chip1 has an open page policy and if chip1_policy[6] is 0x1, bank6 of chip1 has a close page policy. Refer to "Section 2.4.1. Bank Selective Precharge Policy".	R/W	0x0
chip0_policy	[7:0]	**Memory chip0 Precharge Bank Selective Policy** 0x0 = open page policy, 0x1 = close page (auto precharge) policy Chip0_policy[n], n is the bank number of chip0. This is for memory chip0.	R/W	0x0

4.2.7 PHY Control0 Register (PhyControl0, R/W, Address=0xE600_0018)

PHYCONTROL0	Bit	Description	R/W	Reset Value
ctrl_force	[31:24]	**DLL Force Delay** This field is used instead of ctrl_lock_value[9:2] from the DLL when ctrl_dll_on is LOW. (i.e. If the DLL is off, this field is used to generate 270' clock and shift DQS by 90'.)	R/W	0x0
ctrl_inc	[23:16]	**DLL Delay Increment** Increase the amount of start point This value should be 0x10	R/W	0x0
ctrl_start_point	[15:8]	**DLL Lock Start Point** Initial DLL lock start point. This is the number of delay cells and is the start point where "DLL" start tracing to lock. Calculates Initial delay time by multiplying the unit delay of delay cell and this value. This value should be 0x10	R/W	0x0
ctrl_fnc_fb	[7:5]	**Function Feedback Test** 0x0 = Normal Operation Mode, 0x1 = Reserved, 0x2 = External FNC feedback test mode, 0x3 = Internal FNC feedback test mode, 0x4 = Board PHY external read feedback, 0x5 = Board PHY internal read feedback, 0x6 = Board PHY internal write feedback, 0x7 = Reserved	R/W	0x0
Reserved	[4]	**Should be zero**		
ctrl_dfdqs	[3]	**Differential DQS** If enabled, PHY generates differential DQS out signals for write command and receives differential DQS input signals for read command. This function is used in case of DDR2/LPDDR2.	R/W	0x0
ctrl_half	[2]	**DLL Low Speed** HIGH active signal to turn on the low speed mode for DLL. If this bit is set, DLL runs at low speed (80MHz ~ 100MHz)	R/W	0x0
ctrl_dll_on	[1]	**DLL On** HIGH active start signal to turn on the DLL. This signal should be kept HIGH for normal operation. If this signal becomes LOW, DLL is turned off and ctrl_clock and ctrl_flock become HIGH. This bit should be kept set before ctrl_start is set to turn on the DLL	R/W	0x0
ctrl_start	[0]	**DLL Start** HIGH active start signal to make the DLL run and lock. This signal should be kept HIGH during normal operation. If this signal becomes LOW, DLL stops running. To re-run DLL, make this signal HIGH again. In the case of re-running, DLL loses previous lock information. Before ctrl_start is set, make sure that ctrl_dll_on is HIGH.	R/W	0x0

NOTE

PHY DLL Lock Procedure.

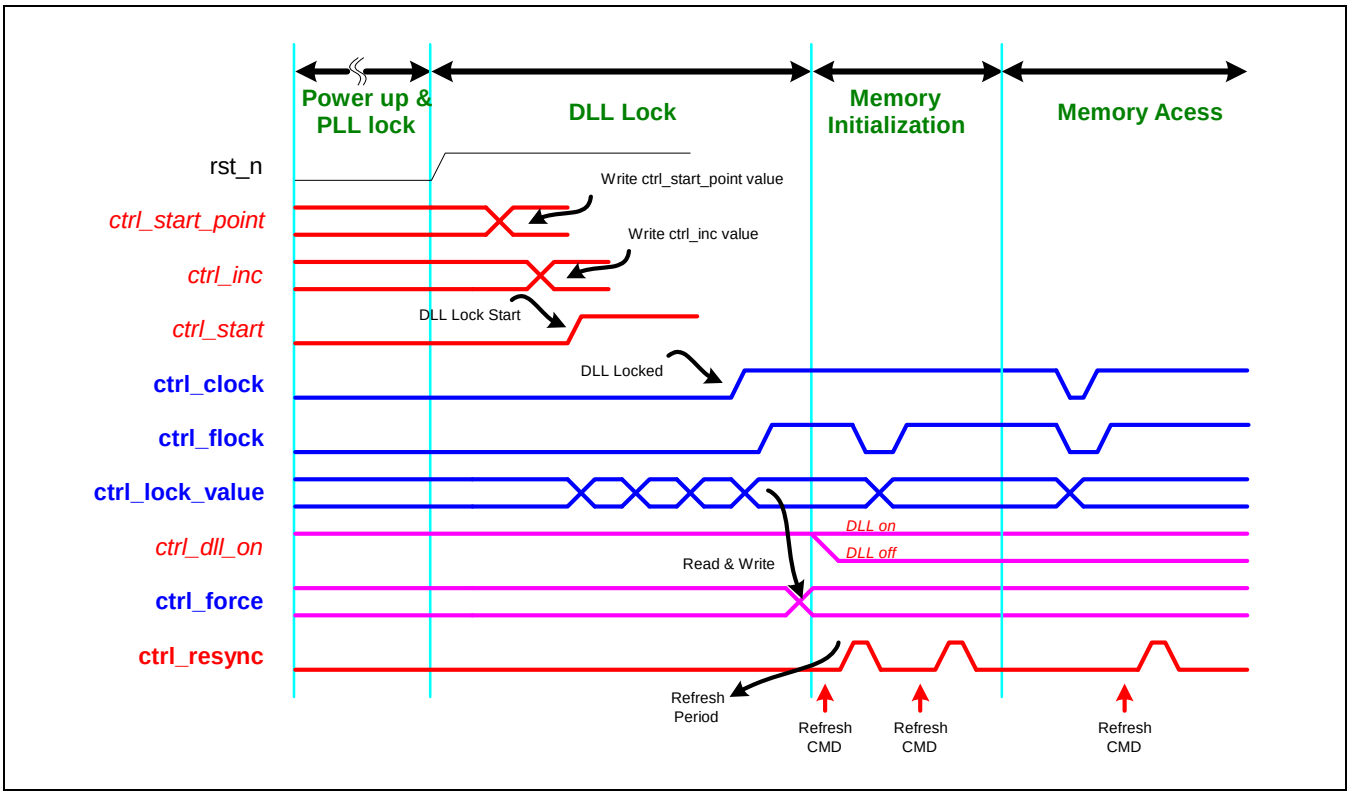


Figure 5.1-10 DLL Lock Procedure

The case of frequency scaling, DLL should be turned-off during the clock change time.

4.2.8 PHY Control1 Register (PhyControl1, R/W, Address=0xE600_001C)

PHYCONTROL1	Bit	Description	R/W	Reset Value
dqs_delay	[31:28]	**Delay Cycles for DQS Cleaning**	R/W	0x0
Reserved	[27:23]	Should be zero		0x0
ctrl_offsetd	[22:16]	This field is for debug purpose. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. offset amount for 270' clock generation ctrl_offsetd[6] = 1 : (tFS : fine step delay) 270' delay amount - ctrl_offsetd[5:0] **x tFS** ctrl_offsetd[6] = 0 : 270' delay amount + ctrl_offsetd[5:0] **x tFS**	R/W	0x0
drv_type	[15]	**Driving Type of Bidirectional Pins in Idle State** 0x0 = Drive all to zeros, 0x1 = Pull down all If CAS or read data latency is 2, this register must not set be to 0x0.		0x0
ctrl_offsetc	[14:8]	**Delay Offset for DQS Cleaning** Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. ctrl_offsetc[6] = 1 : (tFS : fine step delay) GATEout delay amount - ctrl_offsetc[5:0] **x tFS** ctrl_offsetc[6] = 0 : GATEout delay amount + ctrl_offsetc[5:0] **x tFS**	R/W	0x0
ctrl_ref	[7:4]	**Reference Count for DLL Lock Confirmation** This field determines the period of time when ctrl_locked is cleard. 0x0000 : Don't use. 0x0001 : ctrl_flock is de-asserted during 6 clock cycles, ctrl_locked is de-asserted. 0x0010 : ctrl_flock is de-asserted during 9 clock cycles, ctrl_locked is de-asserted. ~ 0x1111 : ctrl_flock is de-asserted during 48 clock cycles, ctrl_locked is de-asserted.	R/W	0x4
fp_resync	[3]	**Force DLL Resynchronization**	R/W	0x0

ctrl_shiftc	[2:0]	**Phase Delay for DQS Cleaning** GATEout signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. 0x000 = T/128 (2.8125' shift), 0x001 = T/64 (5.625' shift), 0x010 = T/32 (11.25' shift), 0x011 = T/16 (22.5' shift), 0x100 = T/8 (45' shift), 0x101 = T/4 (90' shift), 0x110 = T/2 (180' shift), 0x111 = T (360' shift) Recommended values according to memory type : 0x100 when LPDDR/LPDDR2, 0x110 when DDR2	R/W	0x0

NOTE : DQS CLEANING SCHEME

Use DQS cleaning to remove high-Z state of DQS.

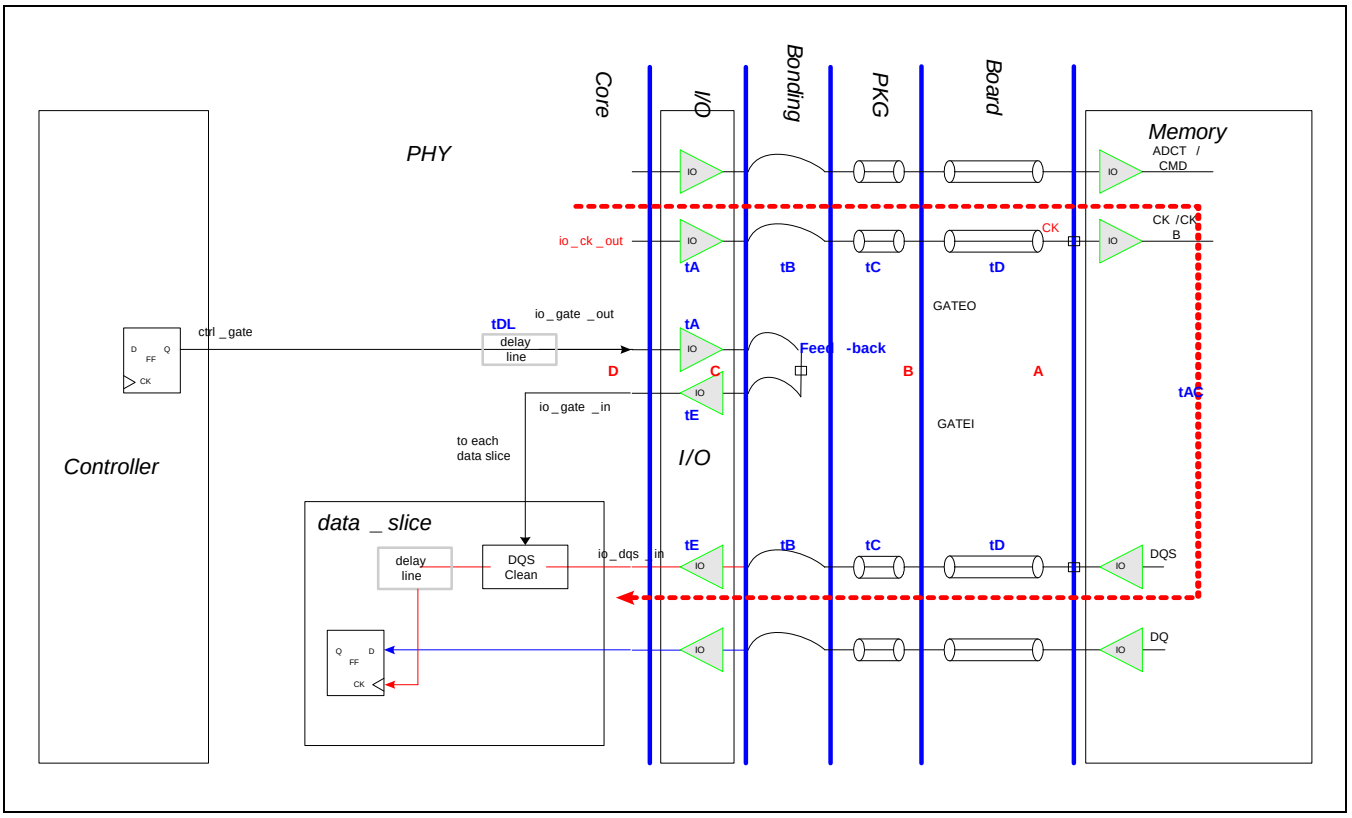


Figure 5.1-11 Board Level Connection Diagram for DQS Cleaning.

tA: I/O output delay tB: Package bonding wire delay tC: Package board delay
tD: Board trace delay tE: I/O input delay tDL: delay line delay
tAC: minimum CK-to-DQS timing of LPDDR/DDR2 memory spec. (LPDDR (1ns, DDR2 (0.5tCK)

tFS : Fine step delay in DLL, From PhyStatus0.ctrl_lock_value[9:0], tFS can beis calculated.

- If ctrl_half = 0, tFS = tCK / ctrl_lock_value[9:0].
- If ctrl_half = 1, tFS = tCK*0.5 / ctrl_lock_value[9:0]

ctrl_shiftc controls PVT-independent delay amount(tF) and **ctrl_offsetc** controls PVT-dependent delay amount(tV).

Delay line programming value; tDL ≈ tAC + 2*(tB+tC+tD).

tDL = tF (ctrl_shiftc[2:0]) + tV (ctrl_offsetc[6:0])

If ctrl_shiftc[2:0] is 3'b100, tF is Tperiod/8 ≈ 0.9375ns. (If tCK is 7.5ns)

If ctrl_offsetc[6:0] is 7'b00010_00, tV is 0.320ns(40ps * 8) @ worst case (if tFS = 40ps)

Therefore tDL = tF + tV = 0.9375ns + 0.320ns = 1.2575ns

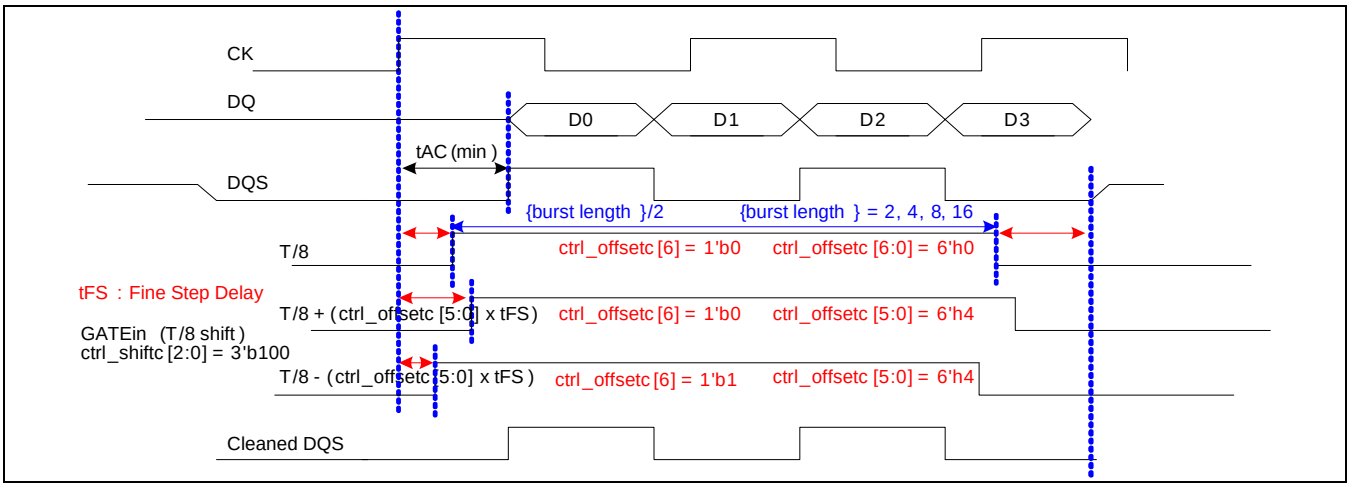


Figure 5.1-12 DQS Cleaning for LPDDR if tAC min

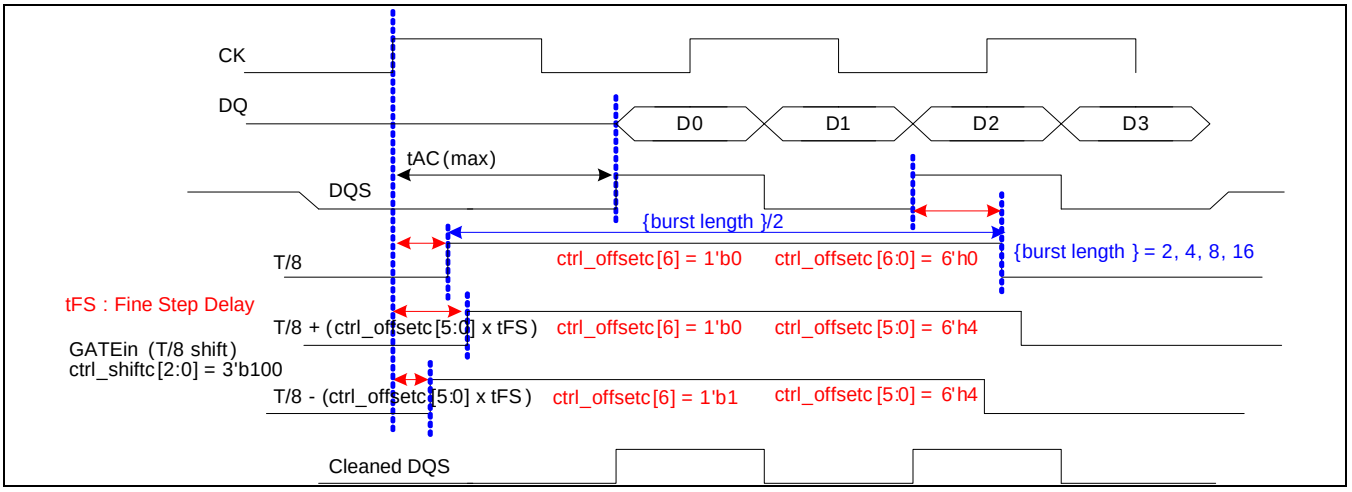


Figure 5.1-13 DQS Cleaning for LPDDR if tAC max

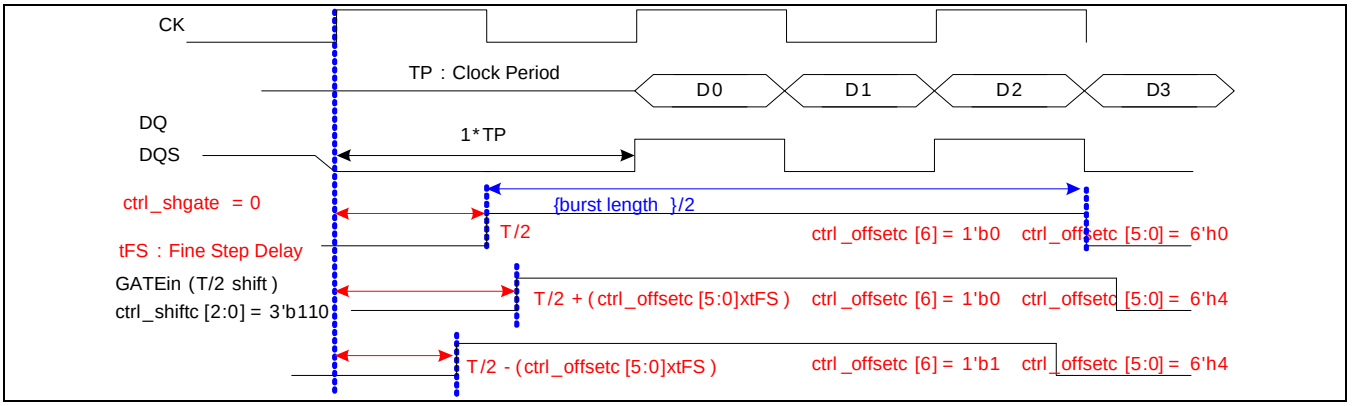


Figure 5.1-14 DQS Cleaning for DDR2

4.2.9 PHY Control2 Register (PhyControl2, R/W, Address=0xE600_0020)

PHYCONTROL2	Bit	Description	R/W	Reset Value
Reserved	[31]	Should be zero		0x0
ctrl_offset3	[30:24]	This field is for debug purpose. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. rd_slice_3 offset amount : ctrl_offset3[6] = 1 : (tFS : fine step delay) 90' delay amount - ctrl_offset0[5:0] **x tFS** ctrl_offset3[6] = 0 : 90' delay amount + ctrl_offset0[5:0] **x tFS**	R/W	0x0
Reserved	[23]	Should be zero		0x0
ctrl_offset2	[22:16]	This field is for debug purpose. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. rd_slice_2 offset amount : ctrl_offset2[6] = 1 : (tFS : fine step delay) 90' delay amount - ctrl_offset0[5:0] **x tFS** ctrl_offset2[6] = 0 : 90' delay amount + ctrl_offset0[5:0] **x tFS**	R/W	0x0
Reserved	[15]	Should be zero		0x0
ctrl_offset1	[14:8]	This field is for debug purpose. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. rd_slice_1 offset amount : ctrl_offset1[6] = 1 : (tFS : fine step delay) 90' delay amount - ctrl_offset0[5:0] **x tFS** ctrl_offset1[6] = 0 : 90' delay amount + ctrl_offset0[5:0] **x tFS**	R/W	0x0
Reserved	[7]	Should be zero		0x0
ctrl_offset0	[6:0]	This field is for debug purpose. If this field is fixed, this should not be changed during operation. This value is valid after ctrl_resync becomes HIGH and LOW. rd_slice_0 offset amount : ctrl_offset0[6] = 1 : (tFS : fine step delay) 90' delay amount - ctrl_offset0[5:0] **x tFS** ctrl_offset0[6] = 0 : 90' delay amount + ctrl_offset0[5:0] **x tFS**	R/W	0x0

4.2.10 Dynamic Power Down Configuration Register (PwrdnConfig, R/W, Address=0xE600_0028)

PWRDNCONFIG	Bit	Description	R/W	Reset Value
dsref_cyc	[31:16]	**Number of Cycles for Dynamic Self Refresh Entry** 0xn = n aclk cycles, If the command queue is empty for n+1 cycles, the controller forces the memory device into self refresh state. Refer to "Section 2.3.3. Dynamic Self Refresh".	R/W	0xFFFF
Reserved	[15:8]	Should be zero		0x0
dpwrdn_cyc	[7:0]	**Number of Cycles for Dynamic Power Down Entry** 0xn = n aclk cycles, If the command queue is empty for n+1 cycles, the controller forces the memory device into active/ precharge power down state. Refer to "Section 2.3.2. Dynamic Power Down".	R/W	0xFF

4.2.11 AC Timing Register for Auto Refresh of memory (TimingAref, R/W, Address=0xE600_0030)

TIMINGAREF	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Should be zero		0x0
t_refi	[15:0]	**Average Periodic Refresh Interval** Should be minimum. memory tREFI (all bank) < t_refi * T(mclk), For example, for the all bank refresh period of 7.8us, and an mclk frequency of 133MHz, the following value should be programmed : 7.8 us * 133 MHz = 1038	R/W	0x40E

4.2.12 AC Timing Register for the Row of memory (TimingRow, R/W, Address=0xE600_0034)

TIMINGROW	Bit	Description	R/W	Reset Value
t_rfc	[31:24]	**Auto refresh to Active / Auto refresh command period, in cycles** t_rft * T(mclk) should be greater than or equal to the minimum value of memory tRFC.	R/W	0xF
t_rrd	[23:20]	**Active bank A to Active bank B delay, in cycles** t_rrd * T(mclk) should be greater than or equal to the minimum value of memory tRRD.	R/W	0x2
t_rp	[19:16]	**Precharge command period, in cycles** t_rp * T(mclk) should be greater than or equal to the minimum value of memory tRP.	R/W	0x3
t_rcd	[15:12]	**Active to Read or Write delay, in cycles** t_rcd * T(mclk) should be greater than or equal to the minimum value of memory tRCD	R/W	0x3
t_rc	[11:6]	**Active to Active period, in cycles** t_rc * T(mclk) should be greater than or equal to the minimum value of memory tRC.	R/W	0xA
t_ras	[5:0]	**Active to Precharge command period, in cycles** t_ras * T(mclk) should be greater than or equal to the minimum value of memory tRAS.	R/W	0x6

4.2.13 AC Timing Register for the Data of memory (TimingData, R/W, Address=0xE600_0038)

TIMINGDATA	Bit	Description	R/W	Reset Value
t_wtr	[31:28]	**Internal write to Read command delay, in cycles** t_wtr * T(mclk) should be greater than or equal to the minimum value of memory tWTR. t_wtr must be 0x1 in case of JEDEC LPDDR.	R/W	0x1
t_wr	[27:24]	**Write recovery time, in cycles** t_wr * T(mclk) should be greater than or equal to the minimum value of memory tWR	R/W	0x2
t_rtp	[23:20]	**Internal read to Precharge command delay, in cycles** t_rtp * T(mclk) should be greater than or equal to the minimum value of memory tRTP. t_rtp must be 0x1 in case of JEDEC LPDDR.	R/W	0x1
cl	[19:16]	**CAS Latency (for LPDDR/DDR/DDR2), in cycles** cl should be greater than or equal to the minimum value of memory CL.	R/W	0x3
Reserved	[15:12]	Should be zero		0x0
wl	[11:8]	**Write data latency (for only LPDDR2), in cycles** wl should be greater than or equal to the minimum value of memory WL	R/W	0x2
Reserved	[7:4]	Should be zero		0x0
rl	[3:0]	**Read data latency (for only LPDDR2), in cycles** rl should be greater than or equal to the minimum value of memory RL	R/W	0x4

* tDAL (Auto precharge write recovery + precharge time) = t_wr + t_rp (automatically calculated)

4.2.14 AC Timing Register for the Power mode of Memory (TimingPower, R/W, Address=0xE600_003C)

TIMINGPOWER	Bit	Description	R/W	Reset Value
Reserved	[31:30]	Should be zero		0x0
t_faw	[29:24]	**Four Active Window** t_faw * T(mclk) should be greater than or equal to the minimum value of memory tFAW	R/W	0xE
t_xsr	[23:16]	**Self refresh exit power down to next valid command delay, in cycles** t_xsr * T(mclk) should be greater than or equal to the minimum value of memory tXSR. In case of DDR/DDR2, this value should be greater than or equal to the minimum value of memory tXSRD.	R/W	0x1B
t_xp	[15:8]	**Exit power down to next valid command delay, in cycles** t_xp * T(mclk) should be greater than or equal to the minimum value of memory tXP	R/W	0x4
t_cke	[7:4]	**CKE minimum pulse width (minimum power down mode duration), in cycles** t_cke should be greater than or equal to the minimum value of memory tCKE	R/W	0x2
t_mrd	[3:0]	**Mode Register Set command period, in cycles** t_mrd should be greater than or equal to the minimum value of memory tMRD.	R/W	0x2

4.2.15 PHY Status Register 0 (PhyStatus0, R, Address=0xE600_0040)

PHYSTATUS0	Bit	Description	R/W	Reset Value
Reserved	[31:14]	Should be zero		0x0
ctrl_lock_value	[13:4]	**Locked Delay** Locked delay line encoding value ctrl_lock_value[9:2]: number of delay cells for coarse lock ctrl_lock_value[1:0]: control value for fine lock	R	0x0
Reserved	[3]	Should be zero		0x0
ctrl_locked	[2]	**DLL Lock** 0 = Unlocks DLL, 1 = Locks DLL	R	0x0
ctrl_flock	[1]	**Fine Lock Information** It is indicated that DLL is locked with fine resolution, "phase offset error" is less than 80ps.	R	0xX
ctrl_clock	[0]	**Coarse Lock Information** It is indicated that DLL changes step delays of the "delay line" and "phase offset error" is less than 160ps.	R	0xX

4.2.16 PHY Status Register 1 (PhyStatus1, R, Address=0xE600_0044)

PHYSTATUS1	Bit	Description	R/W	Reset Value
Reserved	[31:0]		R	0x0

4.2.17 Memory Chip0 Status Register (Chip0Status, R, Address=0xE600_0048)

CHIP0STATUS	Bit	Description	R/W	Reset Value
bank7_state	[31:28]	**The current state of bank 7 of memory chip0**	R	0x0
bank6_state	[27:24]	**The current state of bank 6 of memory chip0**	R	0x0
bank5_state	[23:20]	**The current state of bank 5 of memory chip0**	R	0x0
bank4_state	[19:16]	**The current state of bank 4 of memory chip0**	R	0x0
bank3_state	[15:12]	**The current state of bank 3 of memory chip0**	R	0x0
bank2_state	[11:8]	**The current state of bank 2 of memory chip0**	R	0x0
bank1_state	[7:4]	**The current state of bank 1 of memory chip0**	R	0x0
bank0_state	[3:0]	**The current state of bank 0 of memory chip0** 0x0 = Idle (precharged), 0x1 = MRS/EMRS, 0x2 = Deep power down, 0x3 = Self refresh, 0x4 = Auto refresh, 0x5 = Precharge power down, 0x6 = Row active, 0x7 = Active power down, 0x8 = Write, 0x9 = Write with auto precharge, 0xA = Read, 0xB = Read with auto precharge, 0xC = Burst stop, 0xD = Precharging, 0xE = MRR, 0xF = Reserved	R	0x0

4.2.18 Memory Chip1 Status Register (Chip1Status, R, Address=0xE600_004C)

CHIP1STATUS	Bit	Description	R/W	Reset Value
bank7_state	[31:28]	**The current state of bank 7 of SDRAM chip1**	R	0x0
bank6_state	[27:24]	**The current state of bank 6 of SDRAM chip1**	R	0x0
bank5_state	[23:20]	**The current state of bank 5 of SDRAM chip1**	R	0x0
bank4_state	[19:16]	**The current state of bank 4 of SDRAM chip1**	R	0x0
bank3_state	[15:12]	**The current state of bank 3 of SDRAM chip1**	R	0x0
bank2_state	[11:8]	**The current state of bank 2 of SDRAM chip1**	R	0x0
bank1_state	[7:4]	**The current state of bank 1 of SDRAM chip1**	R	0x0
bank0_state	[3:0]	**The current state of bank 0 of SDRAM chip1** 0x0 = Idle (precharged), 0x1 = MRS/EMRS, 0x2 = Deep power down, 0x3 = Self refresh, 0x4 = Auto refresh, 0x5 = Precharge power down, 0x6 = Row active, 0x7 = Active power down, 0x8 = Write, 0x9 = Write with auto precharge, 0xA = Read, 0xB = Read with auto precharge, 0xC = Burst stop, 0xD = Precharging, 0xE = MRR, 0xF = Reserved	R	0x0

4.2.19 Counter Status Register for the Auto Refresh (ArefStatus, R, Address=0xE600_0050)

AREFSTATUS	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Should be zero		0x0
aref_cnt	[15:0]	**Current Value of Auto Refresh Counter** Shows the current value of all bank auto refresh counter. This is updated if a new t_refi is programmed into the **TimingAref** register and decreases by 1 at the rising edge of mclk. An all bank auto refresh command is issued to memory device and this counter is reloaded with **TimingAref.t_ref** if this becomes zero.	R	0xFFFF

4.2.20 Memory Mode Registers Status Register (MrStatus, R, Address=0xE600_0054)

MRSTATUS	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Should be zero		0x0
mr_status	[7:0]	**Mode Registers Status**	R	0x0

4.2.21 PHY Test Register 0 (PhyTest0, R/W, Address=0xE600_0058)

PHYTEST0	Bit	Description	R/W	Reset Value
ctrl_fb_cnt4	[31:24]	**Count Value for Control Channel**	R	0x0
Reserved	[23:21]	Should be zero		0x0
ctrl_fb_oky	[20:16]	ctrl_fb_okay[4] : okay for control, ctrl_fb_okay[3:0] : okay for data	R	0x0
Reserved	[15:13]	Should be zero		0x0
ctrl_fb_err	[12:8]	ctrl_fb_err[4] : Error for control, ctrl_fb_err[3:0] : Error for data	R	0x0
Reserved	[7:5]	Should be zero		0x0
ctrl_fb_start	[4:0]	ctrl_fb_start[4] : Start for control, ctrl_fb_start[3:0] : Start for data	R/W	0x0

4.2.22 PHY Test Register 1 (PhyTest1, R, Address=0xE600_005C)

PHYTEST1	Bit	Description	R/W	Reset Value
ctrl_fb_cnt3	[31:24]	**Count value for data3 channel**	R	0x0
ctrl_fb_cnt2	[23:16]	**Count value for data2 channel**	R	0x0
ctrl_fb_cnt1	[15:8]	**Count value for data1 channel**	R	0x0
ctrl_fb_cnt0	[7:0]	**Count value for data0 channel**	R	0x0

4.2.23 Quality of Service Control Register n (QosControl n, R/W, Address=0xE600_0060 + 8n (n=0~7, integer)

QOSCONTROLn	Bit	Description	R/W	Reset Value
Reserved	[31:28]	Should be zero		0x0
qos_cnt	[27:16]	**QoS Cycles** 0xn = n aclk cycles The matched ARID uses this value for its timeout counters instead of **ConControl.timeout_cnt**.	R/W	0x0
qos_en	[0]	**QoS Enable** 0x0 = Disable, 0x1 = Enable If this function is enabled, its timeout counter works and the ARID is masked with **QoSConfig.qos_mask** and compared with **QoSConfig.qos_id**	R/W	0x0

4.2.24 Quality of Service Configuration Register n (QosConfig n, R/W, Address=0xE600_0064 + 8n (n=0~7, integer)

QOSCONFIGn	Bit	Description	R/W	Reset Value
qos_mask	[31:16]	**QoS Mask Bits** This is used to mask the incoming ARID to compare with the **qos_id**. For example, to have 0b00110XX000 IDs the same QoS, the 4th and 5th bits must be masked. Therefore, **qos_mask** would be 0b1111100111.	R/W	0x0
qos_id	[15:0]	**QoS ID** This is used to compare with the masked ARID to check whether its timeout counter should be used for QoS. After applying the qos_mask to these ARID, it is compared with **qos_id**. The **qos_id** must be 0b001100_0000 using the example above. Comparing the masked ID, if the result is equal to the **qos_id**, then the **QoSControl0.qos_cnt** is applied to this ARID transaction for timeout. Don't care bits must be assigned zeros.	R/W	0x0

Table 5.1-1 Master Transaction ID in S5PC100 (1/2)

Transaction Master	R/W	Transaction ID	Description
ARM	R/W	12'b0000_0000_0000	Integer data & NEON - Noncacheable or strongly ordered read/write
		12'b0000_0000_0100	Integer data & NEON - Shared device read/ write
		12'b0000_0000_1000	Integer data & NEON - Cacheable non-burst write
		12'b0000_0000_1100	Integer data & NEON - Nonshared device read/ write
		12'b0000_0001_0000	Instruction fetch - Noncacheable or strongly ordered
		12'b0000_0001_0100	Instruction fetch - Shared or nonshared device
		12'b0000_0001_1000	Table walk
		12'b0000_0001_1100	Reserved
		12'b0000_0010_0000	Cacheable linefill - All except linefill into data cache including PLE Eviction including PLE
		12'b0000_0010_0100	
		12'b0000_0010_1000	
		12'b0000_0010_1100	
		12'b0000_0011_0000	Reserved
		12'b0000_0011_0100	Reserved
		12'b0000_0011_1000	Cacheable linefill - Linefill into data cache
		12'b0000_0011_1100	Cacheable linefill - Linefill into instruction cache (L2 Noncacheable)
MDMA	R/W	12'b0000_0000_0001	DMA channel0 thread
		12'b0000_0001_0001	DMA channel1 thread
		12'b0000_0010_0001	DMA channel2 thread
		12'b0000_0011_0001	DMA channel3 thread
		12'b0000_0100_0001	DMA channel4 thread
		12'b0000_0101_0001	DMA channel5 thread
		12'b0000_0110_0001	DMA channel6 thread
		12'b0000_0111_0001	DMA channel7 thread
		12'b0000_1000_0001	DMA manager thread
G2D	R/W	12'b0000_0000_0101	G2D
CFCON	R/W	12'b0000_0000_1001	CFCON
CSSYS	R/W	12'b0000_0000_1101	Coresight AHB access port
FIMC0	R/W	12'b0000_0000_0010	Camera IF 0
FIMC1	R/W	12'b0000_0010_0010	Camera IF 1
FIMC2	R/W	12'b0000_0100_0010	Camera IF 2
JPEG	R/W	12'b0000_0110_0010	JPEG
Rotator	R/W	12'b0000_1000_0010	Rotator
FIMD0	R	12'b0000_1010_0010	window 0
	R	12'b0001_1010_0010	window 4
FIMD1	R	12'b0000_1100_0010	window 1
	R	12'b0001_1100_0010	window 2
	R	12'b0010_1100_0010	window 3

Table 5.1-1 Master Transaction ID in S5PC100 (2/2) (Continued)

Transaction Master	R/W	Transaction ID	Description
VP	R	12'b0000_0000_0110	VP Y line buffer 0
		12'b0000_0100_0110	VP Y line buffer 1
		12'b0000_1000_0110	VP Y line buffer 2
		12'b0000_1100_0110	VP Y line buffer 3
		12'b0001_0100_0110	VP CbCr line buffer 0
		12'b0001_1000_0110	VP CbCr line buffer 1
Mixer	R	12'b0000_0010_0110	Mixer window0
		12'b0000_0110_0110	Mixer window1
MFC0	R/W	12'b0000_0000_1010	MFC video signal processing
MFC1	R	12'b0000_0010_1010	MFC memory pool
MFC2	R/W	12'b0000_0100_1010	MFC prediction
MFC3	R/W	12'b0000_0110_1010	MFC de-blocking filter
MMC0	R/W	12'b0000_0000_1110	SDMMC 0
MMC1	R/W	12'b0000_0010_1110	SDMMC 1
MMC2	R/W	12'b0000_0100_1110	SDMMC 2
USBHost0	R/W	12'b0000_0110_1110	USBHost 0
USBHost1	R/W	12'b0000_1000_1110	USBHost 1
USBOTG	R/W	12'b0000_1010_1110	USBOTG
PDMA0	R/W	12'b0000_1100_1110	DMA channel0 thread
		12'b0001_1100_1110	DMA channel1 thread
		12'b0010_1100_1110	DMA channel2 thread
		12'b0011_1100_1110	DMA channel3 thread
		12'b0100_1100_1110	DMA channel4 thread
		12'b0101_1100_1110	DMA channel5 thread
		12'b0110_1100_1110	DMA channel6 thread
		12'b0111_1100_1110	DMA channel7 thread
		12'b1000_1100_1110	DMA manager thread
PDMA1	R/W	12'b0000_1110_1110	DMA channel0 thread
		12'b0001_1110_1110	DMA channel1 thread
		12'b0010_1110_1110	DMA channel2 thread
		12'b0011_1110_1110	DMA channel3 thread
		12'b0100_1110_1110	DMA channel4 thread
		12'b0101_1110_1110	DMA channel5 thread
		12'b0110_1110_1110	DMA channel6 thread
		12'b0111_1110_1110	DMA channel7 thread
		12'b1000_1110_1110	DMA manager thread
3D0	R	12'b0000_0001_0010	G3D vertex texture L1 cache
		12'b0000_0011_0010	G3D Texture L2 cache
3D1	R/W	12'b0000_0001_0110	G3D Pixel cache 0
		12'b0000_0011_0110	G3D Z buffer/Stencil cache 0
3D2	R/W	12'b0000_0001_1010	G3D Pixel cache 1
		12'b0000_0011_1010	G3D Z buffer/Stencil cache 1

5.2 STATIC MEMORY CONTROLLER

1 OVERVIEW

The S5PC100 Static Memroy Controller supports external 8/ 16-bit NOR Flash/ PROM/ SRAM memory. From now on, we refer this controller as SMC.

SMC supports 6-bank memory (Maximum 128MB).

1.1 FEATURE

- Supports SRAM, various ROMs and NOR flash memory
- Supports only 8 or 16-bit data bus
- Address space: Maximum 4MB per Bank
- Supports 6 banks.
- Fixed memory bank start address
- External wait to extend the bus cycle
- Support byte and half-word access for external memory

1.2 BLOCK DIAGRAM

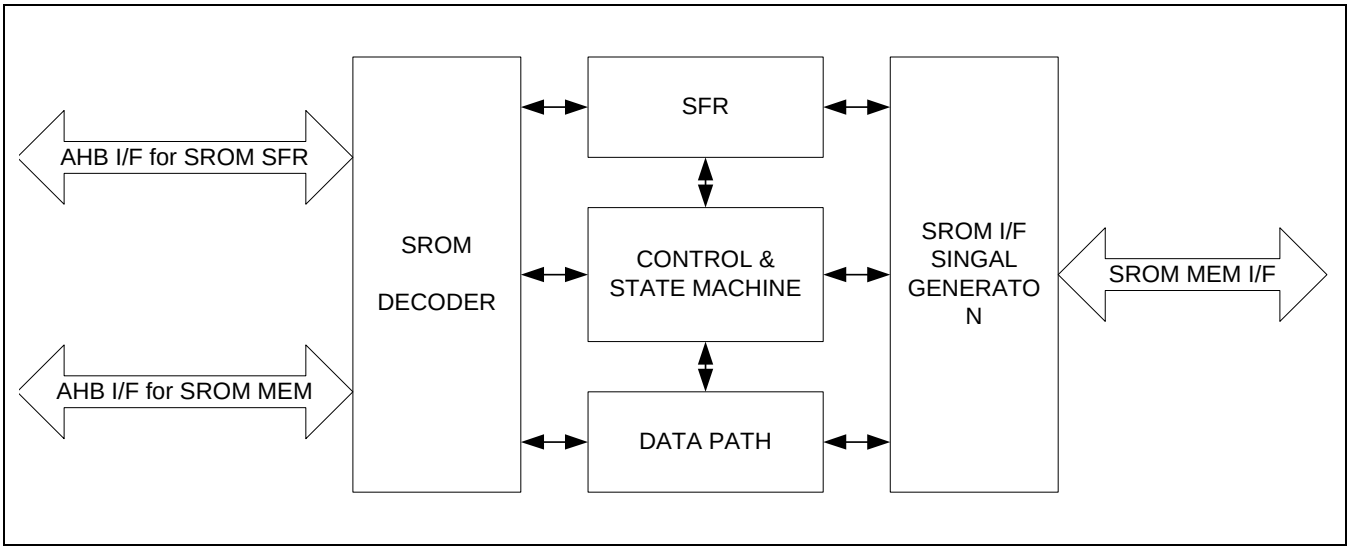


Figure 5.2-1 SMC Block Diagram

2 FUNCTIONAL DESCRIPTION

SMC Controller support SMC interface for Bank0 to Bank5.

2.1 NWAIT PIN OPERATION

If the WAIT corresponding to each memory bank is enabled, the nOE duration should be prolonged by the external nWAIT pin while the memory bank is active. nWAIT is checked from tacc-1. nOE will be deasserted at the next clock after sampling nWAIT is high. The nWE signal have the same relation with nOE.

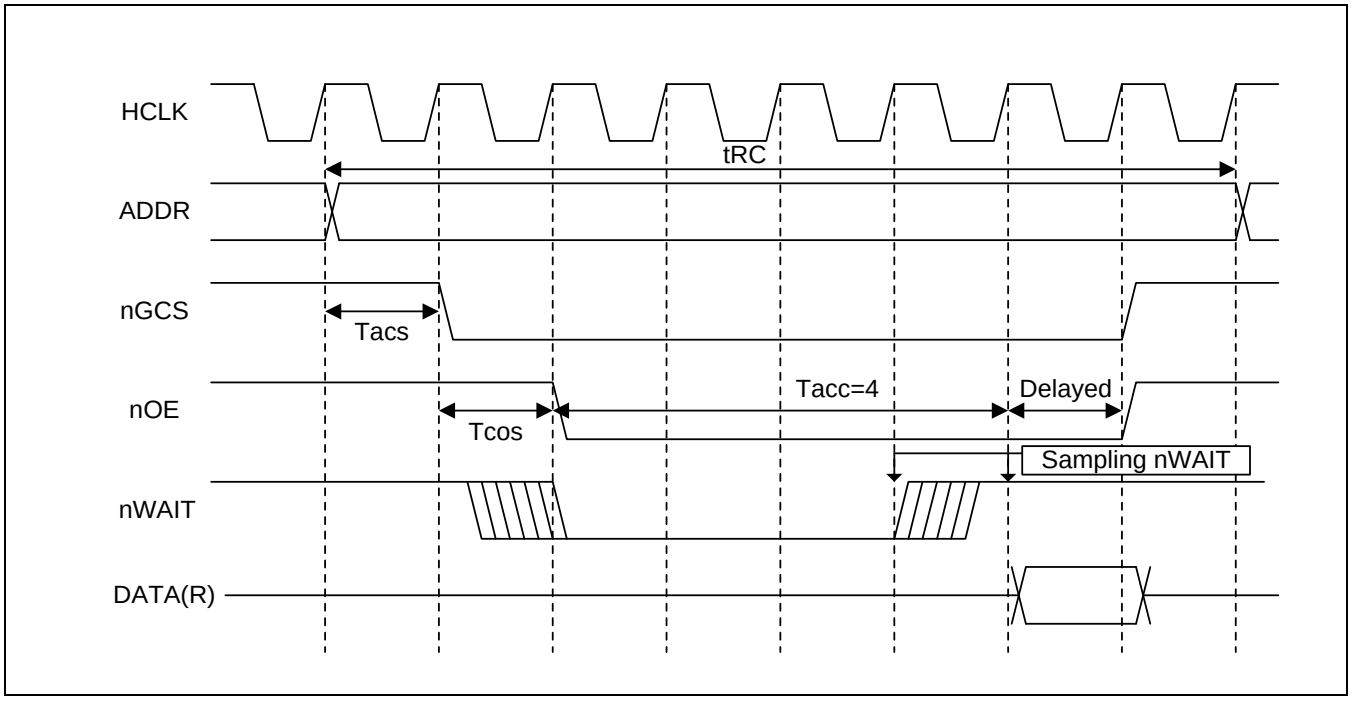


Figure 5.2-2 SMC Controller nWAIT Timing Diagram

2.2 PROGRAMMABLE ACCESS CYCLE

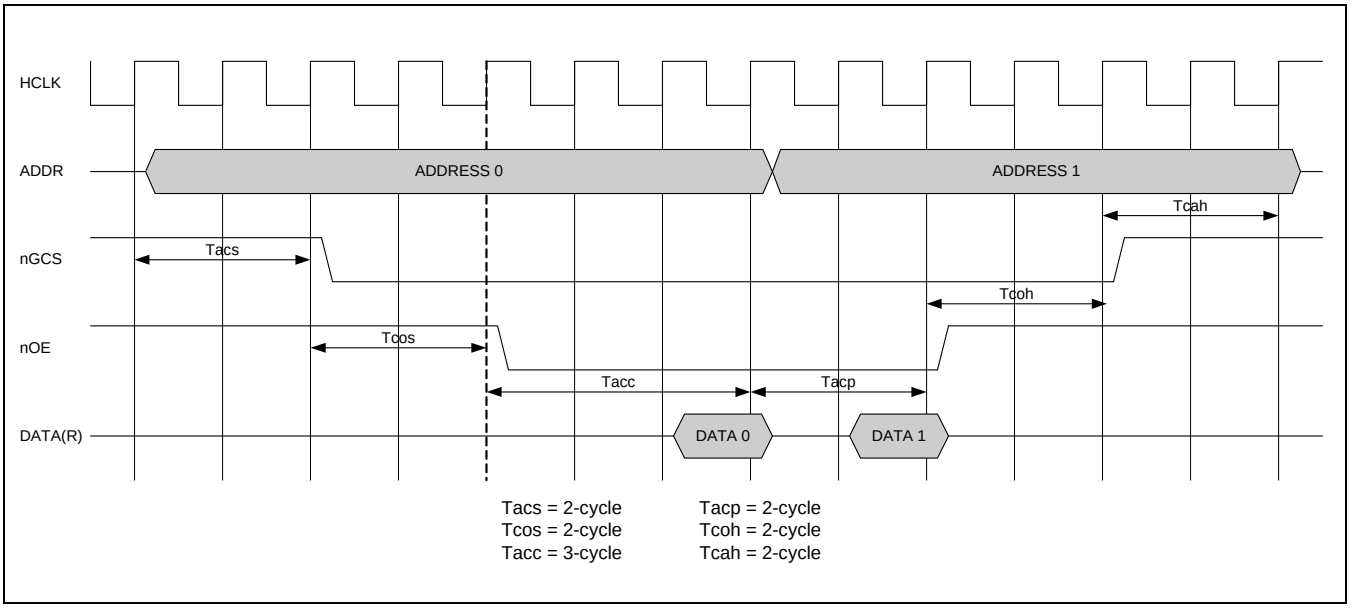


Figure 5.2-3 SMC Controller Read Timing Diagram

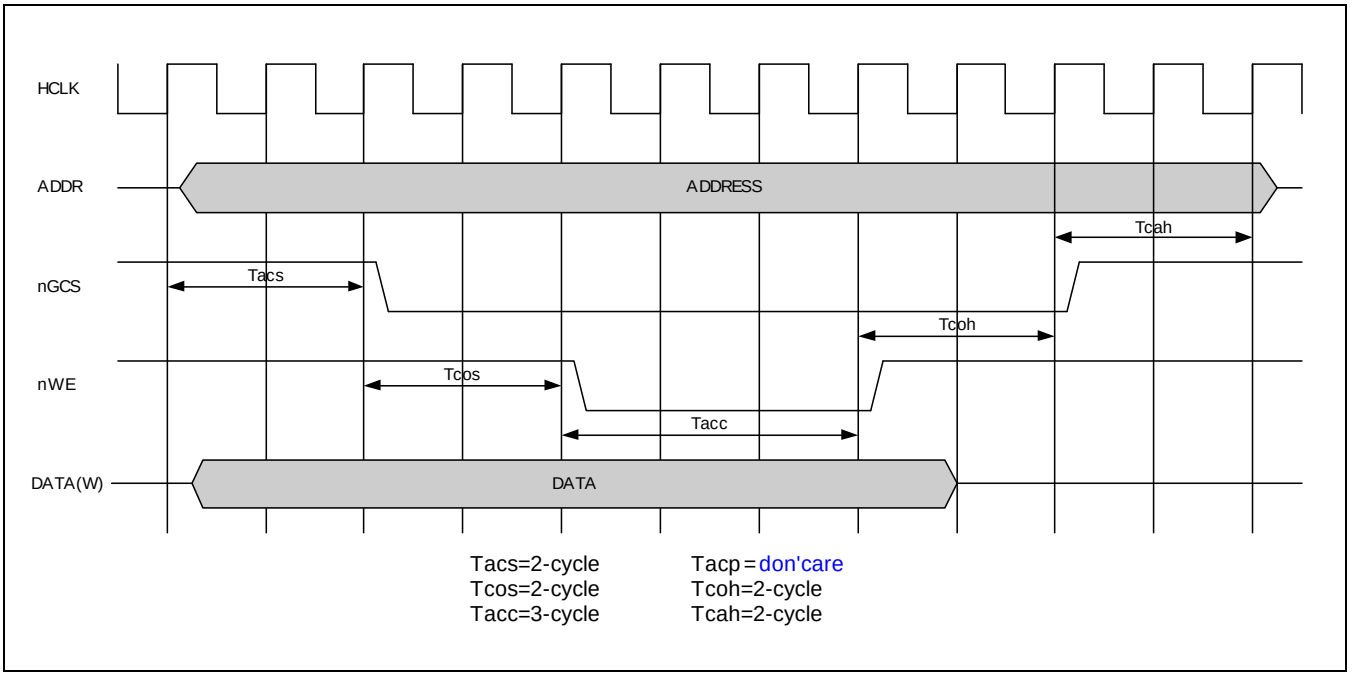


Figure 5.2-4 SMC Controller Write Timing Diagram

3 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
SMC_nCS[5:0]	Output	Bank Selection Signal	Xm0CSn[5:0]	muxed
SMC_ADDR[20:0]	Output	Address bus	Xm0ADDR[20:0]	muxed
SMC_OEn	Output	Output Enable	Xm0OEn	muxed
SMC_WEn	Output	Write Enable	Xm0WEn	muxed
SMC_Ben [1:0]	Output	Byte write Enable/ Byte Enable	Xm0BEn	muxed
SMC_DATA[15:0]	In/Out	Data bus	Xm0DATA[15:0]	muxed
SMC_WAITn	Input	Wait input	Xm0WAITn	muxed

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
SMC_BW	0xE700_0000	R/W	SMC Bus Width & Wait control	0x00000009
SMC_BC0	0xE700_0004	R/W	SMC Bank0 Control Register	0x000F0000
SMC_BC1	0xE700_0008	R/W	SMC Bank1 Control Register	0x000F0000
SMC_BC2	0xE700_000C	R/W	SMC Bank2 Control Register	0x000F0000
SMC_BC3	0xE700_0010	R/W	SMC Bank3 Control Register	0x000F0000
SMC_BC4	0xE700_0014	R/W	SMC Bank4 Control Register	0x000F0000
SMC_BC5	0xE700_0018	R/W	SMC Bank5 Control Register	0x000F0000

4.1 SMC BUS WIDTH & WAIT CONTRL REGISTER (SMC_BW, RW, ADDRESS=0XE700_0000)

Field	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
ByteEnable5	[23]	nWBE / nBE(for UB/LB) control for Memory Bank5 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]	0
WaitEnable5	[22]	Enables Wait for Memory Bank5 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode5	[21]	Select SMC ADDR Base for Memory Bank5 0 = SMC_ADDR is Half-word base address. (SRAM_ADDR[20:0] ← HADDR[21:1]) 1 = SMC_ADDR is byte base address (SMC_ADDR[20:0] ← HADDR[20:0]) **Note:** If DataWidth0 is "0", SMC_ADDR is byte base address. (Ignored this bit.)	0
DataWidth5	[20]	Data bus width control for Memory Bank5 0 = 8-bit 1 = 16-bit	0
ByteEnable4	[19]	nWBE / nBE(for UB/LB) control for Memory Bank4 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]	0
WaitEnable4	[18]	Enables Wait for Memory Bank4 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode4	[17]	Select SMC ADDR Base for Memory Bank4 0 = SMC_ADDR is Half-word base address. (SRAM_ADDR[20:0] ← HADDR[21:1]) 1 = SMC_ADDR is byte base address (SMC_ADDR[20:0] ← HADDR[20:0]) **Note:** If DataWidth0 is "0", SMC_ADDR is byte base address. (Ignored this bit.)	0
DataWidth4	[16]	Data bus width control for Memory Bank4 0 = 8-bit 1 = 16-bit	0
ByteEnable3	[15]	nWBE / nBE(for UB/LB) control for Memory Bank3 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]	0
WaitEnable3	[14]	Wait enable control for Memory Bank3 0 = Disables WAIT 1 = Enables WAIT	0

Field	Bit	Description	Reset Value
AddrMode3	[13]	Select SMC ADDR Base for Memory Bank3 0 = SMC_ADDR is Half-word base address. (SRAM_ADDR[20:0] ← HADDR[21:1]) 1 = SMC_ADDR is byte base address (SMC_ADDR[20:0] ← HADDR[20:0]) **Note:** If DataWidth0 is "0", SMC_ADDR is byte base address. (Ignored this bit.)	0
DataWidth3	[12]	Data bus width control for Memory Bank3 0 = 8-bit 1 = 16-bit	0
ByteEnable2	[11]	nWBE / nBE(for UB/LB) control for Memory Bank2 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]	0
WaitEnable2	[10]	Wait enable control for Memory Bank2 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode2	[9]	Select SMC ADDR Base for Memory Bank2 0 = SMC_ADDR is Half-word base address. (SRAM_ADDR[20:0] ← HADDR[21:1]) 1 = SMC_ADDR is byte base address (SMC_ADDR[20:0] ← HADDR[20:0]) **Note:** If DataWidth0 is "0", SMC_ADDR is byte base address. (Ignored this bit.)	0
DataWidth2	[8]	Data bus width control for Memory Bank2 0 = 8-bit 1 = 16-bit	0
ByteEnable1	[7]	nWBE / nBE(for UB/LB) control for Memory Bank1 0 = Not using UB/LB (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = Using UB/LB (XrnWBE[1:0] is dedicated nBE[1:0]	0
WaitEnable1	[6]	Wait enable control for Memory Bank1 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode1	[5]	Select SMC ADDR Base for Memory Bank1 0 = SMC_ADDR is Half-word base address. (SRAM_ADDR[20:0] ← HADDR[21:1]) 1 = SMC_ADDR is byte base address (SMC_ADDR[20:0] ← HADDR[20:0]) **Note:** If DataWidth0 is "0", SMC_ADDR is byte base address. (Ignored this bit.)	0
DataWidth1	[4]	Data bus width control for Memory Bank1 0 = 8-bit 1 = 16-bit	0

Field	Bit	Description	Reset Value
ByteEnable0	[3]	nWBE / nBE(for UB/LB) control for Memory Bank0 0 = UB/LB used (XrnWBE[1:0] is dedicated nWBE[1:0]) 1 = UB/LB not used (XrnWBE[1:0] is dedicated nBE[1:0]	1
WaitEnable0	[2]	Wait enable control for Memory Bank0 0 = Disables WAIT 1 = Enables WAIT	0
AddrMode0	[1]	Select SMC ADDR Base for Memory Bank0 0 = SMC_ADDR is Half-word base address. (SRAM_ADDR[20:0] ← HADDR[21:1]) 1 = SMC_ADDR is byte base address (SMC_ADDR[20:0] ← HADDR[20:0]) **Note:** If DataWidth0 is "0", SMC_ADDR is byte base address. (This bit is ignored.)	0
DataWidth0	[0]	Data bus width control for Memory Bank0 0 = 8-bit 1 = 16-bit	1

4.2 SMC BANK CONTROL REGISTER (SMC_BC: XRCSN0 ~ XRCSN2)

Register	Address	R/W	Description	Reset Value
SMC_BC0	0xE700_0004	R/W	SMC Bank0 Control Register	0x000F0000
SMC_BC1	0xE700_0008	R/W	SMC Bank1 Control Register	0x000F0000
SMC_BC2	0xE700_000C	R/W	SMC Bank2 Control Register	0x000F0000
SMC_BC3	0xE700_0010	R/W	SMC Bank3 Control Register	0x000F0000
SMC_BC4	0xE700_0014	R/W	SMC Bank4 Control Register	0x000F0000
SMC_BC5	0xE700_0018	R/W	SMC Bank5 Control Register	0x000F0000

Field	Bit	Description	Reset Value
Tacs	[31:28]	Address set-up before nGCS 0000 = 0 clock 0001 = 1 clocks 0010 = 2 clocks 0011 = 3 clocks …… 1100 = 12 clocks 1101 = 13 clocks 1110 = 14 clocks 1111 = 15 clocks **Note:** More 1~2 cycles according to bus i/f status	0000
Tcos	[27:24]	Chip selection set-up before nOE 0000 = 0 clock 0001 = 1 clocks 0010 = 2 clocks 0011 = 3 clocks …… 1100 = 12 clocks 1101 = 13 clocks 1110 = 14 clocks 1111 = 15 clocks	0000
Reserved	[23:21]	Reserved	000
Tacc	[20:16]	Access cycle 00000 = 1 clock 00001 = 2 clocks 00010 = 3 clocks 00011 = 4 clocks …… 11100 = 29 clocks 11101 = 30 clocks 11110 = 31 clocks 11111 = 32 clocks	01111
Tcoh	[15:12]	Chip selection hold on nOE 0000 = 0 clock 0001 = 1 clocks 0010 = 2 clocks 0011 = 3 clocks …… 1100 = 12 clocks 1101 = 13 clocks 1110 = 14 clocks 1111 = 15 clocks	0000

Field	Bit	Description	Reset Value
Tcah	[11:8]	Address holding time after nGCSn 0000 = 0 clock 0001 = 1 clocks 0010 = 2 clocks 0011 = 3 clocks …… 1100 = 12 clocks 1101 = 13 clocks 1110 = 14 clocks 1111 = 15 clocks **Note:** More 1~2 cycles according to bus i/f status	0000
Tacp	[7:4]	Succesive access cycle 0000 = 0 clock 0001 = 1 clocks 0010 = 2 clocks 0011 = 3 clocks …… 1100 = 12 clocks 1101 = 13 clocks 1110 = 14 clocks 1111 = 15 clocks	0000
Reserved	[3:2]	Reserved	
Reserved	[1:0]	Should be zero.	00

5.3 ONENAND CONTROLLER

1 OVERVIEW

S5PC100 supports external 16-bit bus for both asynchronous and synchronous OneNAND external memory via shared memory port 0. The controller is a complete embedded controller that interfaces directly to Input/ Output (I/O) drivers in the I/O pad ring that connect to OneNAND Flash memory devices.

1.1 FEATURE

- Supports maximum 2 chip selects that should be same type device.
- Supports asynchronous / synchronous muxed OneNAND memory
- Supports 16-bit wide external memory data paths
- Supports Data Buffering: Required to achieve maximum performance
- Asynchronous FIFOs between the controller and the system interface for speed matching
- Read and write protocols implemented with bus interface read and write commands.
- Erase commands implemented through address mapping.
- Programmable-sized burst transfers.
- Support for multiple memory devices in the OneNAND family with a single bus interface protocol.
- Supported transfer types are INCRx transactions of one word and SINGLE word transactions for the register slave. Transfers of a size less than width are not supported.
- Supports 4-way and 2-way interleaving operation for Program and Read operations
- Provision for software reset in which the controller is reset, but register values are retained
- Supports write-synchronous mode if OneNAND device ID is 0x0040, 0x0058, 0x0050, 0x0068, 0x0060, and 0x0078

1.2 CONTROLLER USAGE EXPECTATIONS

The controller is designed with the following expectations:

- If the controller is in "Spare area" mode or "Main + Spare area" mode (set through the programming of the *TRANS_SPARE* parameter and the command type used), programming the memory burst length value to 32 causes an unsupported command interrupt.
- If the controller is programmed with 32-word burst length, any attempt to change the controller from Main area mode triggers an interrupt and the change is ignored.
- AMBA transactions should be of the same burst type throughout a page. During page transfers, the entire page must be transferred to memory or to the AMBA bus before the dataram buffer is released for another transaction.
- Register accesses must use a transfer type (HBURST) of SINGLE.
- Data access must use a transfer size (HSIZE) of a WORD.
- SPLIT and RETRY are not supported.
- Early burst termination is not supported.
- Limited support for unspecified length transfers.

2 FUNCTIONAL DESCRIPTION

2.1 BLOCK DIAGRAM

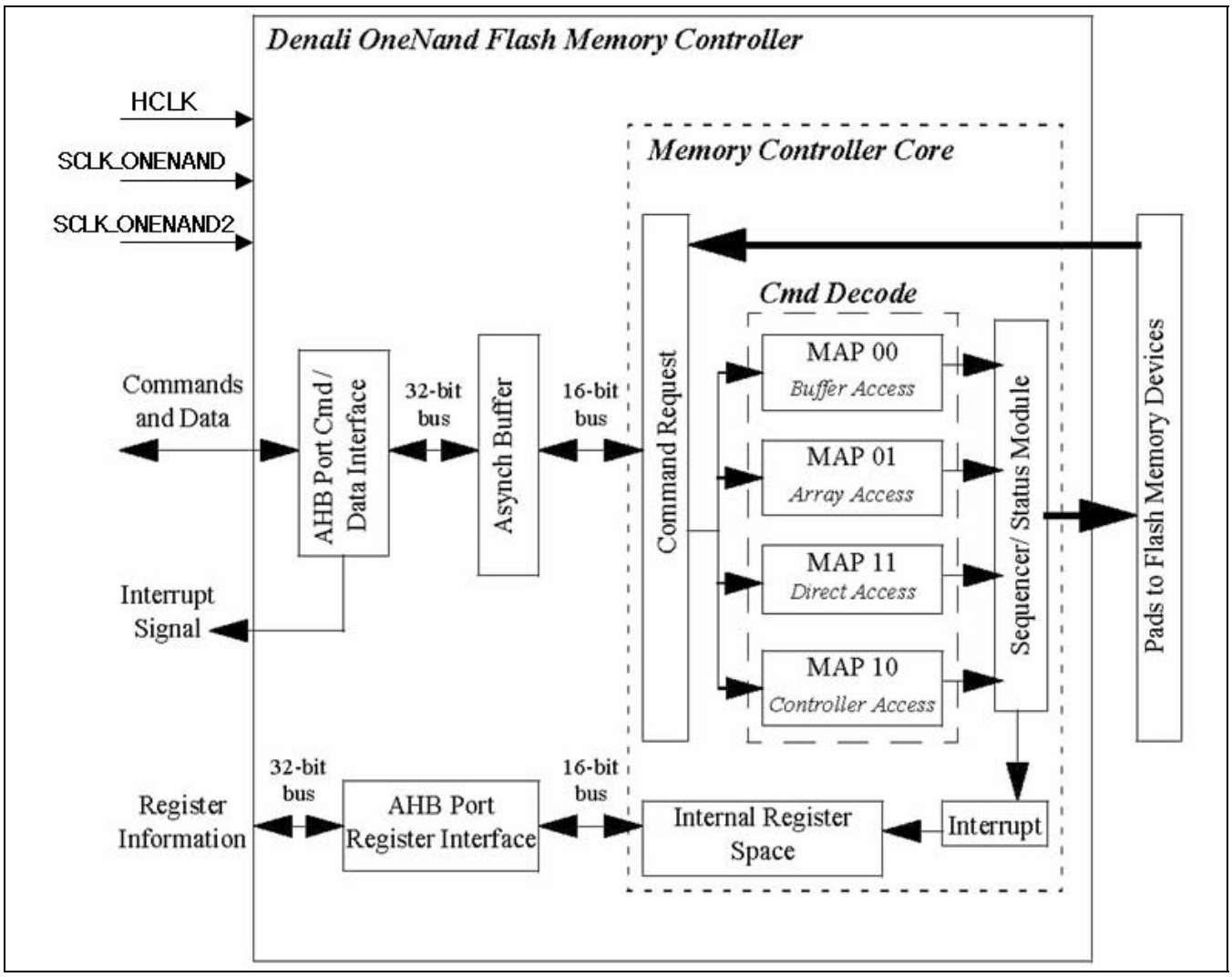


Figure 5.3-1 OneNAND Controller Block Diagram

2.2 CLOCK CONTROL

The controller has three clock source inputs. Bus system interface gets AHB bus clock, HCLK. Flash controller core gets two flash clocks, SCLK_ONENAND and SCLK_ONENAND2. Frequency of SCLK_ONENAND should be double of SCLK_ONENAND2, which is supplied to OneNAND flash memory.

Set the frequency ratio in the SFR of the System Controller. Refer to "*Chapter 2.3 Clock Controller"* for clock ratio settings.

If you change the clock frequency ratio, you should follow the below procedure:

1. Ensure that there are no memory transfers.
2. Switch the clock ratio in the SFR of Clock Controller.
3. Write to the Clock Ratio Register.
4. Start the memory accesses.

2.3 ACCESS CLOCKS

The value programmed into the *access_clocks* register is used to determine the length of time, in clock cycles, that the controller must wait before capturing data from the flash devices on reads, or before sending data to the flash devices on writes. Access clocks are used for asynchronous operations.

The value should be based on the flash clock period, based on the following formula:

$$\text{ACCESS\_CLOCKS} = int\left(\frac{35\ \text{ns}}{\text{FLASH CLOCK PERIOD in ns}} + 1\right)$$

As an example, if the flash device is running at 83 MHz, the clock period would be 12.048 ns, then the formula would result to int(35/12.048) + 1 = 3.905, therefore *access_clocks* should be programmed to 3.

2.4 INITIALIZATION PROTOCOL

1) Power On

The controller is designed such that it requires a sequence for correct operation after all power to the S5PC100 and to the memory devices is stable. The controller does not include circuitry to control the activation of power and ground to the system. Once the power to the memory devices and the S5PC100 is stable, the controller must be initialized.

The memory device is stable after 470us later than POR activation.

When the system is reset and enable special clock(CLK_GATE_SCLK_0[2]) and count done *Cold_Reset_Delay(refer to 4.1.33)*, the controller automatically reads certain registers in the flash memory device and stores the values into the controller registers. If you want to skip Cold_Reset_Delay counting, Write "0x00" to Cold_Reset_Delay SFR. To configure the system to use the flash memory devices, you must read the following controller registers:

- manufacturer_id
- device_id
- version_id
- data_buffer_size
- boot_buffer_size
- amount_of_buffers
- technology

2) Auto Configuration

After performing the power on sequence and enable special clock(CLK_GATE_SCLK_0[2]), the controller automatically configures itself to work with the OneNAND Flash memory devices that are connected by populating the following registers:

- ddp_support
- cache_read_support
- sync_write
- fba_width
- fpa_width
- fsa_width
- device_page_size
- single_page_buffer

If any of these parameters are incorrectly decoded for a new device, they are overwritten with the correct values by software.

The register superload_support and cache_program_support must be set before issuing superload and cache program command.

2.5 ADDRESS MAPPING

There are four kinds of interfaces supported by the controller. These interfaces are selected through the value of bits 27 and 26 of the incoming address. The interface mapping determines the way how the lower 26 bits of the address bus are used.

addr[27:26]	Interface	Description
00	MAP00	Boot read or buffer read/Write during RMW operations.
01	MAP01	Array read/write
10	MAP10	Commands
11	MAP11	Direct access

2.5.1 MAP00

The MAP00 interface are used to communicate with either of the dataram buffers or the boot buffer. This is considered an "execute in place" (XIP) buffer, which means that the selection of the buffer is invisible to the command requestor. The starting address for the buffer always be 0x0, and the maximum address is based on the *BOOT_BUF_SIZE* register (for normal operation) or the *DATA_BUF_SIZE* register (for read/modify/write operations.)

The value of the address bits [16:0] (addr[0] should be zero) is the address of the buffer.

If the controller is initialized, the controller accesses the read-only section of flash memory of boot code. This code is read and used to configure for the specific memory device.

During normal operation, the XIP buffer points to the boot buffer and the user may only read the data. However, if following a read/modify/write sequence, the XIP buffer points to the dataram0 buffer and you modify any word of data through read or write MAP00 commands.

If a MAP00 command is being used and the controller is not in the middle of a read/modify/write sequence, only AHB read transactions are appropriate. If a MAP00 command is received with an AHB write in this case, an interrupt is triggered and the command is ignored.

2.5.2 MAP01.

The MAP01 interface is used for block data transfers to and from the memory device to a specific sector of the flash array. Since the controller only supports page addresses (FSA must be set to 0), an entire page must be read or written at a time. The actual number of commands used depends on the size of the data to be transferred. Even if multiple commands are required, the same address should be used until the entire block is transferred. The write data should be written from the start of the block to the end of the block. If the controller receives the final word of the transfer, it issues commands necessary to program the data into the array.

The meaning of the address bits [25:0] can be found in Table 5.2-1.

MAP01 command allows the host device to read or write specific pages of the flash memory. Byte enables will be ignored for writes and reads, and the controller always writes an entire page. The write data must be written from the start of the page to the end of the page. Once the last word of the page is received, the controller issues the commands necessary to program the data into the array.

If a read or write command is received during an ongoing erase operation, as long as the read/ write is not to an address in the erase area, the erase operation will be suspended (if possible) and the read/ write is completed. At the end of the read/ write, the erase operation resumes. If the device is not able to suspend the erase, or if the

address of the read/ write is in the area of the part that is being erased, the controller holds off the read/ write until the erase is completed.

OneNAND Flash memory devices contain ECC circuitry and data is checked against the expected check code on read commands. If an ECC error occurs during a read, the data is returned on the data bus, but the controller marks the error by setting the Ld_Fail_Ecc_Err bit in the *int_error_status* register and save error information in the *ecc_error_status* register. In addition, the block and page of the failing read is saved in the *ERR_PAGE_ADDR* and *ERR_PAGE_ADDR_1* registers.

In addition to array access, MAP01 commands are also used to read and write alternate areas of the flash device including OTP, boot and spare areas if these areas are selected through MAP10 commands.

2.5.3 MAP10.

The MAP10 interface is used to operate special functions of the memory device. For writes, the data is not transferred to the memory device, but is used to manipulate the controller. For reads, the data is not information from the memory device, but indicates status of the controller.

There are several operations executed through the MAP10 commands, such as erase, lock, unlock, lock-tight, copy back, OTP access, spare area access, verify read, pipeline read, pipeline write and read/ modify/ write. The lower two bytes of the address bus determines the type of MAP10 command.

2.5.4 The address bits [15:0] Identify the command type. MAP11

The MAP11 interface is a direct interface with the memory device. Through this interface, you have direct control of the address specified without any translation through the controller. The address MAPused is dependent on the memory device being used. Data written or read is written to or read from the memory device itself. This command type is used to debug or test the memory device.

MAP11 and MAP10 can not share INT pin. If the system use MAP11 and MAP10 in same time, replace external interrupt instead of INT pin.

Table 5.3-1 summarizes the MAP10 commands.

2.5.5 MAP11

The MAP11 interface is a direct interface with the memory device. Through this interface, you have direct control of the address specified without any translation through the controller. The address MAPused is dependent on the memory device being used. Data written or read is written to or read from the memory device itself. This command type is used to debug or test the memory device.

MAP11 and MAP10 can not share INT pin. If the system use MAP11 and MAP10 in same time, replace external interrupt instead of INT pin.

Table 5.3-1 MAP10 Commands

Cmd Type	Cmd Value[15:0]	Function
Write	0x0000	Save erase status to controller
Read	-	Send current or last erase status to AHB interface
Write	0x0001	Save block address for multi-block erase
Write	0x0003	Save block address for single-block or the final block of a multi-block erase and initiate the erase
Write	0x0008	Set unlock start address
Write	0x0009	Set unlock end address and initiate the unlock
Write	0x000a	Set lock start address
Write	0x000b	Set lock end address and initiate the lock
Write	0x000c	Set lock-tight start address
Write	0x000d	Set lock-tight end address initiate the lock
Write	0x000e	Unlock the entire memory array
Write	0x0010	Load page to the RMW buffer
Write	0x0011	Write RMW buffer to memory
Write	0x0012	Set up for OTP access
Write	0x0013	Set up for spare area access only
Write	0x0014	Set up for main area access only
Write	0x0015	Verify erase block
Write	0x0016	Set up for main and spare area access
Write	0x1000	Set copy source address
Write	0x20PP	Set copy destination address and initiate copy of PP pages
Write	0x4WPP	Set up a pipeline read or pipeline write of PP pages W = 0 for pipeline read W = 1 for pipeline 2X write W = 2 for pipeline 2X read W = 3 for 2-way pipeline read W = 4 for 4-way pipeline read W = 5 for Reserved W = 6 for 2-way pipeline write W = 7 for 4-way pipeline write W = 8 for Reserved

Table 5.3-2 Address Mapping

MAP	Addr	MAP00	MAP11	MAP01								
device ID;				b0000	b0001	b0010	b0011	b0011	b0100	b0100	b0101	b0101
Density				128Mb	256Mb	512Mb	1Gb(D)	1Gb	2Gb(D)	2Gb	4Gb(D)	4Gb
CMD_MAP	27	0	1	0	0	0	0	0	0	0	0	0
	26	0	1	1	1	1	1	1	1	1	1	1
Address	25	resv.	resv.	resv.	resv.	resv.	resv.	resv.	resv	resv	CS	CS
	24	resv.	resv.	resv.	resv.	resv.	resv	resv	CS	CS	DFS_DBS	FBA[11]
	23	resv.	resv.	resv.	resv.	resv	CS	CS	DFS_DBS	FBA[10]	FBA[10]	FBA[10]
	22	resv.	resv.	resv.	resv	CS	DFS_DBS	FBA[9]	FBA[9]	FBA[9]	FBA[9]	FBA[9]
	21	resv.	resv.	resv	CS	FBA[8]	FBA[8]	FBA[8]	FBA[8]	FBA[8]	FBA[8]	FBA[8]
	20	resv.	resv.	CS	FBA[8]	FBA[7]	FBA[7]	FBA[7]	FBA[7]	FBA[7]	FBA[7]	FBA[7]
	19	resv.	resv.	FBA[7]	FBA[7]	FBA[6]	FBA[6]	FBA[6]	FBA[6]	FBA[6]	FBA[6]	FBA[6]
	18	resv	CS	FBA[6]	FBA[6]	FBA[5]	FBA[5]	FBA[5]	FBA[5]	FBA[5]	FBA[5]	FBA[5]
	17	CS	ADDR[15]	FBA[5]	FBA[5]	FBA[4]	FBA[4]	FBA[4]	FBA[4]	FBA[4]	FBA[4]	FBA[4]
	16	ADDR[15]	ADDR[14]	FBA[4]	FBA[4]	FBA[3]	FBA[3]	FBA[3]	FBA[3]	FBA[3]	FBA[3]	FBA[3]
	15	ADDR[14]	ADDR[13]	FBA[3]	FBA[3]	FBA[2]	FBA[2]	FBA[2]	FBA[2]	FBA[2]	FBA[2]	FBA[2]
	14	ADDR[13]	ADDR[12]	FBA[2]	FBA[2]	FBA[1]	FBA[1]	FBA[1]	FBA[1]	FBA[1]	FBA[1]	FBA[1]
	13	ADDR[12]	ADDR[11]	FBA[1]	FBA[1]	FBA[0]	FBA[0]	FBA[0]	FBA[0]	FBA[0]	FBA[0]	FBA[0]
	12	ADDR[11]	ADDR[10]	FBA[0]	FBA[0]	FPA[5]	FPA[5]	FPA[5]	FPA[5]	FPA[5]	FPA[5]	FPA[5]
	11	ADDR[10]	ADDR[9]	FPA[5]	FPA[5]	FPA[4]	FPA[4]	FPA[4]	FPA[4]	FPA[4]	FPA[4]	FPA[4]
	10	ADDR[9]	ADDR[8]	FPA[4]	FPA[4]	FPA[3]	FPA[3]	FPA[3]	FPA[3]	FPA[3]	FPA[3]	FPA[3]
	9	ADDR[8]	ADDR[7]	FPA[3]	FPA[3]	FPA[2]	FPA[2]	FPA[2]	FPA[2]	FPA[2]	FPA[2]	FPA[2]
	8	ADDR[7]	ADDR[6]	FPA[2]	FPA[2]	FPA[1]	FPA[1]	FPA[1]	FPA[1]	FPA[1]	FPA[1]	FPA[1]
	7	ADDR[6]	ADDR[5]	FPA[1]	FPA[1]	FPA[0]	FPA[0]	FPA[0]	FPA[0]	FPA[0]	FPA[0]	FPA[0]
	6	ADDR[5]	ADDR[4]	FPA[0]	FPA[0]	FSA[1]	FSA[1]	FSA[1]	FSA[1]	FSA[1]	FSA[1]	FSA[1]
	5	ADDR[4]	ADDR[3]	FSA[0]	FSA[0]	FSA[0]	FSA[0]	FSA[0]	FSA[0]	FSA[0]	FSA[0]	FSA[0]
	4	ADDR[3]	ADDR[2]	offset[4]	offset	offset	offset	offset	offset	offset	offset	offset
	3	ADDR[2]	ADDR[1]	offset[3]	offset	offset	offset	offset	offset	offset	offset	offset
	2	ADDR[1]	ADDR[0]	offset[2]	offset	offset	offset	offset	offset	offset	offset	offset
	1	ADDR[0]	BYTE = 0	offset[1]	offset	offset	offset	offset	offset	offset	offset	offset
	0	BYTE = 0	BYTE = 0	offset[0]	offset	offset	offset	offset	offset	offset	offset	offset

NOTE1. Offset is configured by offset_addr Register (SFR, 0x0380). If offset_addr is 5'h5, FSA[0] bit is allocated at HADDR[5] in MAP01/10 case.

NOTE2. FBA: Flash Block Address(Max 4096blocks per device), FPA: Flash Page Address(Max 64pages per block), FSA: Flash sector Address

NOTE3. In the MAP00, ADDR[N] mean device bufferRAM address. In the MAP11, ADDR[N] mead devcie register address.

2.6 WATCHDOG TIMER

The watchdog state machine monitors the various state machines of the controller. If the watchdog state machine detects a potential hang in the controller or is waiting for some information from the OneNAND Flash memory device, it is send out a watchdog interrupt if the watchdog timer expires. This does not trigger a state machine change, and the OneNAND Flash memory device does not reset; software is expected to handle this condition and take the necessary action. The software changes the watchdog reset values by writing to the *WTCHDG_RST_L* and/or *WTCHDG_RST_H* registers.

2.7 ERROR HANDLING

If an error is detected, the controller updates the *int_error_status* register with the corresponding error information and sends out an interrupt. In addition, the *ERR_PAGE_ADDR* and *ERR_PAGE_ADDR_1* registers are updated with the chip number, die number, block address and page address relating to the error situation. These registers holds this information until they are read and the host clears the error bit in the *int_error_status* register.

If an error is detected in a pipelined operation, only the error information is updated and an interrupt is sent. The operation will not halt and continues until the operation is complete. Similarly, the error information is held until the error bit in the *int_error_status* register is cleared.

3 I/O DESCRIPTION

Function Signal	I/O	Description	PAD	TYPE
OND_DATA[15:0]	I/O	Data Bus outputs address during memory read/ write address phase, inputs data during memory read data phase and outputs data during memory write data phase.	Xm0DATA[15:0]	muxed
OND_CSn[1:0]	O	Chip Selects are activated if the address of a memory is within the address region of each bank. Xm0CSn[3:2] is assigned to either SROMC or OneNAND controller by System Controller SFR setting. Active LOW.	Xm0CSn[3:2]	muxed
OND_WEn	O	Write Enable indicates that the current bus cycle is a write cycle. Active LOW.	Xm0WEn	muxed
OND_OEn	O	Output Enable indicates that the current bus cycle is a read cycle. Active LOW.	Xm0OEn	muxed
OND_INT[1:0]	I	Interrupt inputs from OneNAND memory Bank 0, 1. If OneNAND memory is not used, these signals should be tied to zero.	Xm0FRnB[1:0]	muxed
OND_AVALID	O	Address valid output. In the POP products, address and data are multiplexed. Xm0ADDRVALID indicate if the bus is used for address. Active LOW.	Xm0FCLE	muxed
OND_PRn	O	System reset output for OneNAND memory. Active LOW.	Xm0FWEn	muxed
OND_SMCLK	O	Static memory clock for synchronous static memory devices.	Xm0FALE	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
MEM_CFG	0xE710_0000	R/W	Memory Device Configuration Register	Device Dependent
BURST_LEN	0xE710_0010	R	Burst Length Register	Device Dependent
MEM_RESET	0xE710_0020	R/W	Memory Reset Register	0x0000
INT_ERR_STAT	0xE710_0030	R	Interrupt Error Status Register	0x0000
INT_ERR_MASK	0xE710_0040	R/W	Interrupt Error Mask Register	0x0000
INT_ERR_ACK	0xE710_0050	R/W	Interrupt Error Acknowledge Register	0x0000
ECC_ERR_STAT_1	0xE710_0060	R	ECC Error Status Register	0x0000
MANUFACT_ID	0xE710_0070	R	Manufacturer ID Register	Device Dependent
DEVICE_ID	0xE710_0080	R	Device ID Register	Device Dependent
DATA_BUF_SIZE	0xE710_0090	R	Data Buffer Size Register	Device Dependent
BOOT_BUF_SIZE	0xE710_00A0	R	Boot Buffer Size Register	Device Dependent
BUF_AMOUNT	0xE710_00B0	R	Amount of Buffer Register	Device Dependent
TECH	0xE710_00C0	R	Technology Register	Device Dependent
FBA_WIDTH	0xE710_00D0	R/W	FBA Width Register	Device Dependent
FPA_WIDTH	0xE710_00E0	R/W	FPA Width Register	Device Dependent
FSA_WIDTH	0xE710_00F0	R/W	FSA Width Register	Device Dependent
REVISION	0xE710_0100	R	Revision Register	0x09
Reserved	0xE710_0110	R	Reserved	0x0000
Reserved	0xE710_0120	R	Reserved	0x0000
SYNC_MODE	0xE710_0130	R	Synchronous Mode Register	0x0000
TRANS_SPARE	0xE710_0140	R/W	Transfer Size Register	0x0000
Reserved	0xE710_0150	R/W	Reserved	0x0000
Reserved	0xE710_0160	R/W	Reserved	0x0000
PAGE_CNT	0xE710_0170	R	Page Count Register	0x0000
ERR_PAGE_ADDR	0xE710_0180	R	Error Page Address Register	0x0000
BURST_RD_LAT	0xE710_0190	R	Burst Read Latency Register	0x0007
INT_PIN_ENABLE	0xE710_01A0	R/W	Interrupt Pin Enable Register	0x0000
INT_MON_CYC	0xE710_01B0	R/W	Interrupt Monitor Cycle Count Register	0x01F4
ACC_CLOCK	0xE710_01C0	R/W	Access Clock Register	0x0003
reserved	0xE710_01D0	R		0x0000
ERR_BLK_ADDR	0xE710_01E0	R	Error Block Address Register	0x0000
FLASH_VER_ID	0xE71001F0	R	Flash Version ID Register	Device Dependent
Reserved	0xE7100200	R	Reserved	0x0000
Reserved	0xE7100210	R	Reserved	0x0000
BANK_EN	0xE7100220	R/W	Enable banks selection Register	0x0120
reserved	0xE7100230	R		0x0000

Register	Address	R/W	Description	Reset Value
Reserved	0xE710_0240	R	Reserved	0x0000
Reserved	0xE710_0250	R	Reserved	0x0000
WTCHDG_RST_L	0xE710_0260	R/W	Programmable Watchdog Delay Counter Register	0x21c0
WTCHDG_RST_H	0xE710_0270	R/W	Programmable Watchdog Delay Counter Register	0x000d
SYNC_WRITE	0xE710_0280	R	Synchronous Write Enable Register	Device Dependent
CACHE_READ	0xE710_0290	R/W	Cache Read Enable Register	Device Dependent
COLD_RST_DLY	0xE710_02A0	R/W	Cold Reset Delay Counter Register	0x3355
DDP_DEVICE	0xE710_02B0	R/W	DDP Device Indicator Register	Device Dependent
MULTI_PLANE	0xE710_02C0	R/W	Multi-plane Enable Register	0x0001
MEM_CNT	0xE710_02D0	R/W	Number of Transferred Pages Register	0x0000
TRANS_MODE	0xE710_02E0	R	Indicates Transfer Area Register	0x0000
DEV_STAT	0xE710_02F0	R	Device Status Register	0x0000
ECC_ERR_STAT_2	0xE710_0300	R	ECC Error Status Register	0x0000
ECC_ERR_STAT_3	0xE710_0310	R	ECC Error Status Register	0x0000
ECC_ERR_STAT_4	0xE710_0320	R	ECC Error Status Register	0x0000
EFCT_BUF_CNT	0xE710_0330	R	Effective Number of Dataram Buffers Register	0x0000
DEV_PAGE_SIZE	0xE710_0340	R/W	Device Page Size Register	Device Dependent
SUPERLOAD_EN	0xE710_0350	R/W	Superload Support Enable Register	0x0000
CACHE_PRG_EN	0xE710_0360	R/W	Cache Program Support Enable Register	0x0000
SINGLE_PAGE_BUF	0xE710_0370	R	Indicate Single Page Buffer Size Register	0x0000
OFFSET_ADDR	0xE710_0380	R/W	Number of Address Offset in MAP01/10 Cmd.	0x0005
INT_MON_STATUS	0xE710_0390	R	Initialize & Reset Status Monitor Register	0x0000

4.1 MEMORY DEVICE CONFIGURATION REGISTER (MEM_CFG, R/W, ADDRESS = 0XE710_0000)

MEM_CFG0	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This data is used to configure the flash device for the hardware and software environment and may include burst length, read latency, transfer mode, ECC configuration, polarity levels, etc. The each bits filed are same as System Configuration 1 register (F221h) of memory device. This register is set by software during initialization.	Device Dependent

4.2 BURST LENGTH REGISTER (BURST_LEN, R, ADDRESS = 0XE710_0010)

BURST_LEN	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	
Burst_Length	[5:0]	Specifies the decoded numerical value of the burst length (word count) written in the System Configuration 1 register (F221) of memory device. Read-Only.	Device Dependent

4.3 MEMORY RESET REGISTER (MEM_RESET, R/W, ADDRESS = 0XE710_0020)

MEM_RESET	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	
Reset_Code	[2:0]	Sets the reset code. This register is reset to 0x0 after the reset sequence is complete. This register is controlled through software. 001 = Warm Reset 010 = Core Reset 011 = Hot Reset All other settings are Reserved	000

4.4 INTERRUPT ERROR STATUS REGISTER (INT_ERR_STAT, R/W, ADDRESS = 0XE710_0030)

INT_ERR_STAT	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	
Cache_Op_Err	[13]	An error occurred during a cache read or write setup or operation. If the host does not send an appropriate sequence of MAP01 following a MAP10 pipeline command, then the controller issues a Cache_Op_Err. As an example, a MAP10 command is issued for an 8-way interleaved read of 10 pages starting from address A. If the next command is not a MAP01 command to the same address, or if the order of MAP01 commands is inaccurate, then the Controller issues this interrupt, send a core reset to the device, clears all the pipeline registers and service the request as a normal read. A similar example would hold for a pipelined write command.	0
Rst_Cmp	[12]	The controller has completed its reset and auto- initialization process. (After auto-initialize complete and available OneNAND Device, this bit is "1")	0
Reserved	[11]	Reserved	0
INT_act	[10]	The memory device's INT pin is actively transitioning.	0
Unsup_Cmd	[9]	An unsupported command was received. This interrupt is set if an invalid command is received, or if a command sequence is broken.	0
Locked_Blk	[8]	The address to program or erase is in a protected block.	0
Blk_RW_Cmp	[7]	This interrupt indicates one of the following have occurred: - A copyback operation is complete. - A pipeline transaction is complete. - A lock, lock-tight or unlock command to a range of address is complete. - An Erase Verify operation is complete.	0
Ers_Cmp	[6]	The erase operation is complete. This interrupt is automatically reset at the beginning of an erase operation. Default is 0.	0
Pgm_Cmp	[5]	The program operation is complete. This interrupt is automatically reset at the beginning of a program operation. Default is 0.	0
Load_Cmp	[4]	The load operation is complete. Default is 0.	0
Ers_Fail	[3]	The erase operation was unsuccessful.	0
Pgm_Fail	[2]	The program operation was unsuccessful.	0
Int_T0	[1]	Interrupt time-out.	0
Ld_Fail_Ecc_Err	[0]	Dual purpose interrupt bit. The load operation was unsuccessful or there was an ECC error.	0

4.5 INTERRUPT ERROR MASK REGISTER (INT_ERR_MASK, R/W, ADDRESS = 0XE710_0040)

INT_ERR_MASK	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	
Cache_Op_Err	[13]	Mask bits that correspond to the bits in the int_error_status register. Setting this bit allows the controller to issue this type of interrupt. Set through software. **Note:** Interrupts are only sent to the host if an interrupt is set in the int_error_status register, the corresponding bit is set in this register, and the INT_PIN_ENABLE register is set.	0
Rst_Cmp	[12]		0
Reserved	[11]		0
INT_act	[10]		0
Unsup_Cmd	[9]		0
Locked_Blk	[8]		0
Blk_RW_Cmp	[7]		0
Ers_Cmp	[6]		0
Pgm_Cmp	[5]		0
Load_Cmp	[4]		0
Ers_Fail	[3]		0
Pgm_Fail	[2]		0
Int_TO	[1]		0
Ld_Fail_Ecc_Err	[0]		0

4.6 INTERRUPT ERROR ACKNOWLEDGE REGISTER (INT_ERR_ACK, R/W ADDRESS = 0XE710_0050)

INT_ERR_ACK	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	
Cache_Op_Err	[13]	Acknowledge bits that correspond to the bits in the int_error_status register. Setting this bit resets or acknowledges the associated interrupt. Set by software.	0
Rst_Cmp	[12]		0
Reserved	[11]		0
INT_act	[10]		0
Unsup_Cmd	[9]		0
Locked_Blk	[8]		0
Blk_RW_Cmp	[7]		0
Ers_Cmp	[6]		0
Pgm_Cmp	[5]		0
Load_Cmp	[4]		0
Ers_Fail	[3]		0
Pgm_Fail	[2]		0
Int_TO	[1]		0
Ld_Fail_Ecc_Err	[0]		0

4.7 ECC ERROR STATUS REGISTER (ECC_ERR_STAT_1, R, ADDRESS = 0XE710_0060)

ECC_ERR_STAT_1	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This data is used to report ECC error information.	Device Dependent

4.8 MANUFACTURER ID REGISTER (MANUFACT_ID, R, ADDRESS = 0XE710_0070)

MANUFACT_ID	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.9 DEVICE ID REGISTER (DEVICE_ID, R, ADDRESS = 0XE710_0080)

DEVICE_ID	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.10 DATA BUFFER SIZE REGISTER (DATA_BUF_SIZE, R, ADDRESS = 0XE710_0090)

DATA_BUF_SIZE	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.11 BOOT BUFFER SIZE REGISTER (BOOT_BUF_SIZE, R, ADDRESS = 0XE710_00A0)

BOOT_BUF_SIZE	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.12 AMOUNT OF BUFFER REGISTER (BUF_AMOUNT, R, ADDRESS = 0XE710_00B0)

BUF_AMOUNT	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.13 TECHNOLOGY REGISTER (TECH, R, ADDRESS = 0XE710_00C0)

TECH	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.14 FBA WIDTH REGISTER (FBA_WIDTH, R/W, ADDRESS = 0XE710_00D0)

FBA_WIDTH	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	
FBA	[4:0]	Sets the number of bits is used to represent the number of blocks. The default value is 0x0A. Set by software during initialization.	Device Dependent

4.15 FPA WIDTH REGISTER (FPA_WIDTH, R/W, ADDRESS = 0XE710_00E0)

FPA_WIDTH	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	
FPA	[4:0]	Sets the number of bits used to represent the number of pages. The default value is 6. Set by software during initialization.	Device Dependent

4.16 FSA WIDTH REGISTER (FSA_WIDTH, R/W, ADDRESS = 0XE710_00F0)

FSA_WIDTH	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	
FSA	[2:0]	Sets the number of bits used to represent the number of sectors. The default value is 0x2. Set by software during initialization.	Device Dependent

4.17 REVISION REGISTER (REVISION, R, ADDRESS = 0XE710_0100)

Revision	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Revision	[15:0]	Holds the controller revision number. Default value is 0x1. Read-Only.	0x09

4.18 SYNCHRONOUS MODE REGISTER (SYNC_MODE, R, ADDRESS = 0XE710_0130)

SYNC_MODE	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	
RM	[1]	Sets the transfer mode for read operations as synchronous or asynchronous. Default value is 0x0. This value is copied from the memory_configuration register [15]. Read-Only. 0 = Asynchronous Mode 1 = Activate Synchronous Mode	0
WM	[0]	Sets the transfer mode for write operations as synchronous or asynchronous. Default value is 0x0. Set by software during initialization. This value is copied from the memory_configuration register [1]. Read-Only. 0 = Asynchronous Mode 1 = Activate Synchronous Mode	0

4.19 TRANSFER SIZE REGISTER (TRANS_SPARE, R/W, ADDRESS = 0XE710_0140)

TRANS_SPARE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
Tsfr_Spare	[0]	If this bit is set, the data in the spare area of memory is transferred to the asynchronous FIFO of the controller along with the main data. Size of the spare area is part dependent depending on the size of the sectors. 0 = Transfers main area or both main and spare areas, 1 = Increase transfer size. Always transfers main area and spare area. The main data area for the page is transferred first and then the spare area. If burst length is 3'b100, this bit dose not set.	0

4.20 PAGE COUNT REGISTER (PAGE_CNT, R, ADDRESS = 0XE710_0170)

PAGE_CNT	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	
Page_Count	[7:0]	Holds the page count of the multi-page command currently being executed. Read-Only.	0

4.21 ERROR PAGE ADDRESS REGISTER (ERR_PAGE_ADDR, R, ADDRESS = 0XE710_0180)

ERR_PAGE_ADDR	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
CS	[15:14]	Chip Select.	0
DFS_DBS	[13]	DFS/DBS Bit.	0
FPA_Fail	[5:0]	After a program, load or erase error interrupt, this register will hold the page address of the failing operation. Read-Only.	0

4.22 BURST READ LATENCY REGISTER (BURST_RD_LAT, R, ADDRESS = 0XE710_0190)

BURST_RD_LAT	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	
Burst_Rd_ Latency	[2:0]	Sets the burst read latency in cycles. The default value is 0x7 This value is copied from the memory_configuration register [14:12]. Read-Only.	0x7

4.23 INTERRUPT PIN ENABLE REGISTER (INT_PIN_ENABLE, R/W, ADDRESS = 0XE710_01A0)

INT_PIN_ENABLE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
INT	[0]	Interrupt Pin Enable. Enables interrupt information to be transmitted to the host. An interrupt is only sent to the host if a bit is set in the int_error_status register, the corresponding bit is set in the INT_ERR_MASK register AND this bit is set. 0 = Does not report interrupts 1 = Send interrupt information to the host.	0

4.24 INTERRUPT MONITOR CYCLE COUNT REGISTER (INT_MON_CYC, R/W, ADDRESS = 0XE710_01B0)

INT_MON_CYC	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	
Int_Mon_Cyc_ Cnt	[11:0]	Sets the number of cycles in between checks of the int_error_status register and the memory device's status register. This register is used if the Flash configuration register bit IOBE is clear.	0x01F4

4.25 ACCESS CLOCK REGISTER (ACC_CLOCK, R/W, ADDRESS = 0XE710_01C0)

ACC_CLOCK	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	
Access_Clocks	[2:0]	Sets the number of cycles required to cover the access time of the Flash memory device. Follows the formula *(35ns / ClkPeriod) + 1*	0x3

2X Clock	Flash Clock	Access_Clocks
166MHz	83MHz	3
134	67	3
100	50	2
60	30	2

4.26 ERROR BLOCK ADDRESS REGISTER (ERR_BLK_ADDR, R, ADDRESS = 0XE710_01E0)

ERR_PAGE_ADDR_1	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	
FBA_Fail	[11:0]	After a program, load or erase error interrupt, this register holds the block address of the failing operation. Read-Only.	0

4.27 FLASH VERSION ID REGISTER (FLASH_VER_ID, R, ADDRESS = 0XE710_01F0)

FLASH_VER_ID	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This register is set by the controller after reset. Read-Only.	Device Dependent

4.28 BANK_EN (BANK_EN, R/W, ADDRESS = 0XE710_0220)

BANK_EN	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	
EBI_REQ_dly	[9:7]	Defines the value for EBI BUS release delay. After transfer is complete, OneNAND Controller waits for this number of cycles before releasing EBI request.	3'b010
EBI_RDY_dly	[6:4]	Defines the value for EBI BUS release delay. After transfer is complete, OneNAND Controller waits for this number of cycles before release EBI request.	3'b010
Reserved	[3:2]	Reserved	2'b0
bank_en	[1:0]	Selects the enabled banks. 00 = Enables Bank 0 01 = Enables Bank 0 and 1 10/11 = Reserved	2'b00

4.29 WATCHDOG RESET LOW REGISTER (WTCHDG_RST_L, R/W, ADDRESS = 0XE710_0260)

WTCHDG_RST_L	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Wtchdog_Rst_ Low	[15:0]	Specifies the least-significant counter value for the programmable watchdog delay.	0x21c0

4.30

4.31 WATCHDOG RESET HIGH REGISTER (WTCHDG_RST_H, R/W, ADDRESS = 0XE710_0270)

WTCHDG_RST_H	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	
Wtchdog_Rst_ High	[3:0]	Specifies the most-significant counter value for the programmable watchdog delay.	0xd

4.32 SYNCHRONOUS WRITE ENABLE REGISTER (SYNC_WRITE, R, ADDRESS = 0XE710_0280)

SYNC_WRITE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
Sync_Wr	[0]	Synchronous write enable. Set by software during initialization. If this bit is not set, the controller assumes that the device connected does not support synchronous writes. Therefore, if this bit is clear but the host tries to program the configuration register of the flash device with the sync_write bit enabled to "1", the controller masks the value of the sync_write bit prior to sending the configuration register information to the OneNAND Flash memory device. In this setting, any writes to the OneNAND Flash memory device are sent in asynchronous mode. If this bit is set, the controller writes configuration register information as programmed. Therefore, if the host enables the "sync_write" bit in configuration register, the controller sends all future writes to the device as synchronous writes. 0 = Not supported. The "Sync_Write" bit of the configuration register is masked. All writes are in asynchronous mode. 1 = Supported. If the "Sync_Write" bit of the configuration register is set, all writes is performed as synchronous writes.	Device Dependent

4.33 CACHE READ ENABLE REGISTER (CACHE_READ, R/W, ADDRESS = 0XE710_0290)

CACHE_READ	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
Cache_Rd	[0]	Cache read enable. Set by software during initialization. If this bit is clear, the controller assumes that the OneNAND Flash memory device does not support cache read/ write operations and ignores any MAP10 pipelined commands. The subsequent MAP01 operations are treated as normal reads and writes. If this bit is set, the controller accepts all MAP10 pipelined commands and process them as expected. 0 = Not supported 1 = Supported	Device Dependent

4.34 COLD RESET DELAY ENABLE REGISTER (COLD_RST_DLY, R/W, ADDRESS = 0XE710_02A0)

COLD_RST_DLY	Bit	Description	Reset Value
Reserved	[31:15]	Reserved	
Cold_Reset_Delay	[14:0]	Defines the value for cold reset delay. After a cold reset, the The controller waits for this number of cycles before reading the device registers.	0x3355

4.35 DDP DEVICE INDICATOR ENABLE REGISTER (DDP_DEVICE, R/W, ADDRESS = 0XE710_02B0)

DDP_DEVICE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
DDP_Device	[0]	DDP device indicator. Set by software during initialization. 0 = Non-DDP part 1 = DDP part	Device Dependent

4.36 MULTI-PLANE ENABLE REGISTER (MULTI_PLANE, R/W, ADDRESS = 0XE710_02C0)

MULTI_PLANE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
MP_Device	[0]	Enables Multi-plane. Set by software during initialization. 0 = Not a Multi-plane device 1 = Multi-plane device	0x1

4.37 MEMORY COUNTER REGISTER (MEM_CNT, R/W, ADDRESS = 0XE710_02D0)

MEM_CNT	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory_Counter	[15:0]	Indicates the number of pages that have been transferred. A free-running counter controls this register's value.	0

4.38 TRANSFER_MODE (TRANS_MODE, R, ADDRESS = 0XE710_02E0)

TRANS_MODE	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	
Tsfr_Mode	[1:0]	Indicates transfer area of the memory. Read-Only. 00 = Main Area Only 01 = Spare Area Only 10 = Main + Spare Area 11 = Reserved	0

4.39 DEVICE STATUS REGISTER (DEV_STAT, R, ADDRESS = 0XE710_02F0)

DEV_STAT	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Status	[15:0]	Holds the status register information from the device that was last read. Read-Only.	0

4.40 ECC ERROR STATUS REGISTER 2 (ECC_ERR_STAT_2, R, ADDRESS = 0XE710_0300)

ECC_ERR_STAT_2	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This data is used to report ECC error information.	0

4.41 ECC ERROR STATUS REGISTER 3 (ECC_ERR_STAT_3, R, ADDRESS = 0XE710_0310)

ECC_ERR_STAT_3	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This data is used to report ECC error information.	0

4.42 ECC ERROR STATUS REGISTER 4 (ECC_ERR_STAT_4, R, ADDRESS = 0XE710_0320)

ECC_ERR_STAT_4	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	
Memory Device Dependent	[15:0]	The value programmed depends on the actual memory device being used. This data is used to report ECC error information.	0

4.43 EFFECTIVE BUFFER COUNT REGISTER (EFCT_BUF_CNT, R, ADDRESS = 0XE710_0330)

EFCT_BUF_CNT	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	
Effective Buffer Count	[1:0]	Effective number of dataram buffers that holds a single page of data from the memory device (DATA_BUF_SIZE divided by DEV_PAGE_SIZE). 00 = 1 buffer 01 = 2 buffers 10 = 4 buffers 11 = 8 buffers	0

4.44 DEVICE PAGE SIZE ENABEL REGISTER (DEV_PAGE_SIZE, R/W, ADDRESS = 0XE710_0340)

DEV_PAGE_SIZE	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	
Device Page Size	[1:0]	Size of a single page of the device. 00 = 1 KB; 01 = 2 KB; 10 = 4 KB; 11 = 8 KB	0

4.45 SUPERLOAD SUPPORT REGISTER (SUPERLOAD_EN, R/W, ADDRESS = 0XE710_0350)

SUPERLOAD_EN	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
Superload Supported	[0]	0 = Not Supported 1 = Supported	0

4.46 CACHE PROGRAM SUPPORT REGISTER (CACHE_PRG_EN, R/W, ADDRESS = 0XE710_0360)

CACHE_PRG_EN	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
Cache Program Supported	[0]	0 = Not Supported 1 = Supported	0

4.47 SINGLE PAGE BUFFER REGISTER (SINGLE_PAGE_BUF, R, ADDRESS = 0XE710_0370)

SINGLE_PAGE_BUF	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
Single Page Buffer	[0]	0 = Device page size not equal to total data buffer size 1 = Device page size equal to total data buffer size	0

4.48 OFFSET ADDRESS (OFFSET_ADDR, R/W, ADDRESS = 0XE710_0380)

OFFSET_ADDR	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	
sector_addr_en	[4]	Enables sector address in MAP01/10 1 = FSA in HADDR is means. (DEBUG only) 0 = FSA is always 0 padding. **Note:** In normal operation, this bit should be "0".	0x0
offset_addr	[3:0]	Sets the number of bits that represents the number of address offset. This Address offset is only means in MAP01 & MAP10 operations.	0x5

4.49 INT MON STATUS (INT_MON_STATUS, R, ADDRESS = 0XE710_0390)

INT_MON_STAT	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	
int_mon_status	[5:0]	6'h0; RESET WAIT 6'h1; IDLE 6'h2 … 6'h35 are dependent on H/W.	0x1

5.4 NAND FLASH CONTROLLER

1 OVERVIEW

Recently NOR Flash memory price has increased and price for DRAM and a NAND Flash memory is moderately placed. This is the reason why customer prefer to execute the boot code on a NAND Flash and execute the main code on DRAM.

S5PC100 boot code is executed on an external NAND Flash memory. The boot code copies NAND Flash content to DRAM. The hardware ECC is used to check data validity of the NAND Flash. After the NAND Fash content is copied to DRAM, main program is executed on DRAM.

2 FEATURES

- NAND Flash memory I/F: Supports 512Bytes, 2KB and 4KB Page.
- Software mode: User can directly access NAND flash memory, for example this feature is used in read/erase/program NAND Flash memory.
- Interface: 8-bit NAND Flash memory interface bus.
- Generates, detects and indicates Hardware ECC (Software correction).
- Supports both SLC and MLC NAND Flash memory
- SFR I/F: Supports Byte/half word/word access to Data and ECC Data register, and Word access to other registers

2.1 BLOCK DIAGRAM

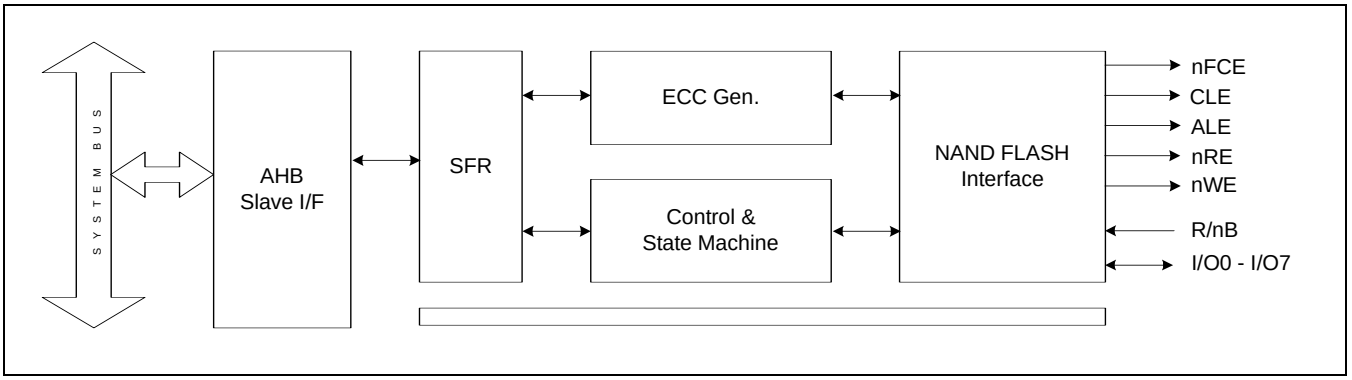


Figure 5.4-1 NAND Flash Controller Block Diagram

2.2 NAND FLASH MEMORY TIMING

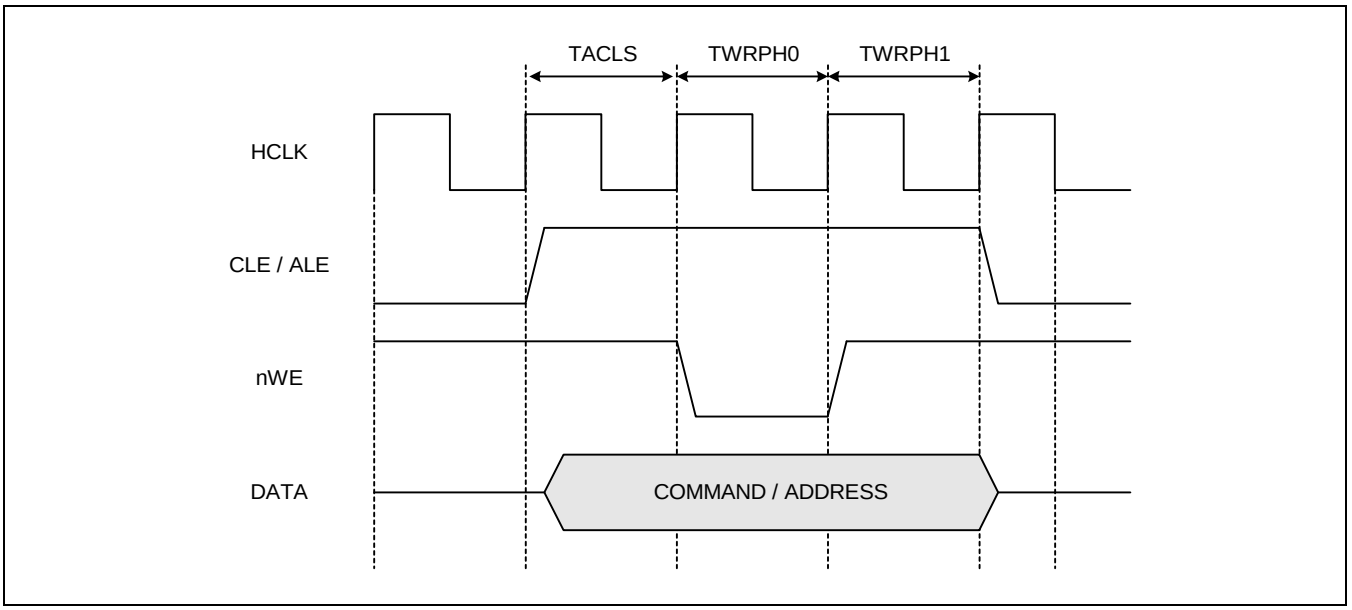


Figure 5.4-2 CLE & ALE Timing (TACLS=1, TWRPH0=0, TWRPH1=0)

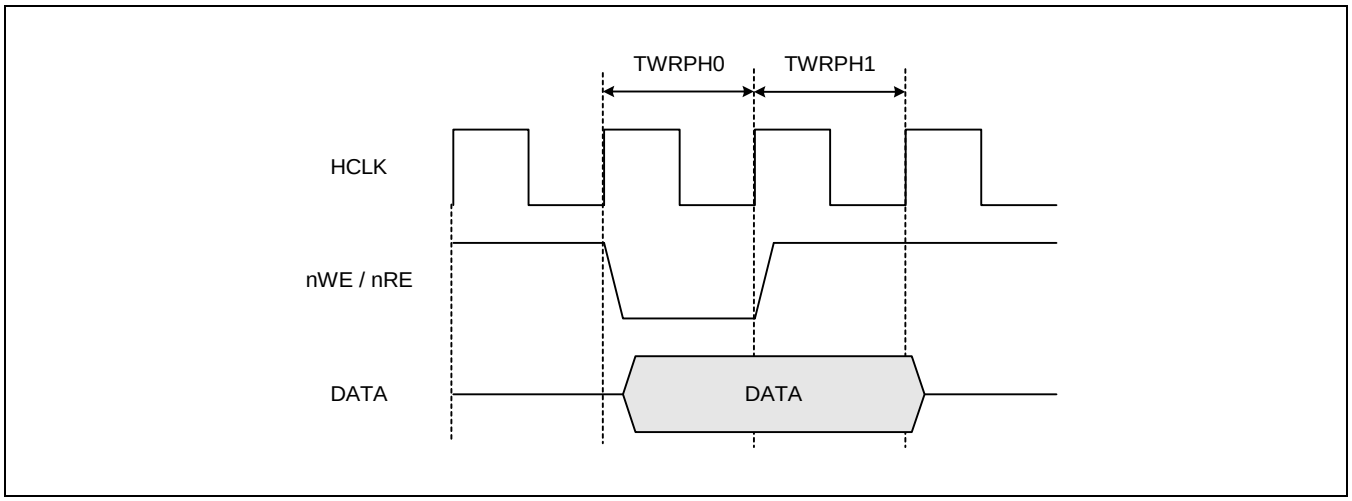


Figure 5.4-3 nWE & nRE Timing (TWRPH0=0, TWRPH1=0)

3 SOFTWARE MODE

S5PC100 only supports software mode access. Use this mode to access NAND flash memory completely. The NAND Flash Controller supports direct access interface with the NAND flash memory.

- Writing to the command register (**NFCMMD**) = the NAND Flash Memory command cycle
- Writing to the address register (**NFADDR**) = the NAND Flash Memory address cycle
- Writing to the data register (**NFDATA**) = write data to the NAND Flash Memory (write cycle)
- Reading from the data register (**NFDATA**) = read data from the NAND Flash Memory (read cycle)
- Reading main ECC registers (**NFMECCD0/NFMECCD1**) and Spare ECC registers (**NFSECCD**) = read data from the NAND Flash Memory

NOTE

In the software mode, you have to check the RnB status input pin by using polling or interrupt.

4 DATA REGISTER CONFIGURATION

1) 8-bit NAND Flash Memory Interface

A. Word Access

Register	Endian	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Little	4th I/O[7:0]	3rd I/O[7:0]	2nd I/O[7:0]	1st I/O[7:0]

B. Half-word Access

Register	Endian	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Little	Invalid Value	Invalid Value	2nd I/O[7:0]	1st I/O[7:0]

C. Byte Access

Register	Endian	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Little	Invalid Value	Invalid Value	Invalid Value	1st I/O[7:0]

5 1-BIT/ 4-BIT/ 8-BIT ECC (ERROR CORRECTION CODE)

NAND Flash controller has four Error Correction Code (ECC) modules for 1-bit ECC, one ECC module for 4-bit ECC and one for 8-bit ECC.

For 1-bit ECC, NAND Flash controller has 4 ECC modules. Some 1-bit ECC modules are used to generate (up to) 2048 bytes ECC parity code and the others are used to generate (up to) 4 bytes ECC Parity code.

NAND Flash controller has one ECC module for 4-bit ECC and one ECC module for 8-bit ECC. They are used to generate 512 or 24 bytes ECC parity code. 4-bit/ 8-bit ECC module generates parity code for each 512 byte. However, 1-bit ECC modules generate parity code per byte lane separately.

Following are the ECC parity code and two tables are 1-bit ECC.

- 28-bit ECC Parity Code = 22-bit Line parity + 6bit Column Parity
- 10-bit ECC Parity Code = 4-bit Line parity + 6bit Column Parity

5.1 2048 BYTE 1-BIT ECC PARITY CODE ASSIGNMENT TABLE

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
MECCn_0	~P64	~P64'	~P32	~P32'	~P16	~P16'	~P8	~P8'
MECCn_1	~P1024	~P1024'	~P512	~P512'	~P256	~P256'	~P128	~P128'
MECCn_2	~P4	~P4'	~P2	~P2'	~P1	~P1'	~P2048	~P2048'
MECCn_3	1	1	1	1	~P8192	~P8192'	~P4096	~P4096'

5.2 4 BYTE 1-BIT ECC PARITY CODE ASSIGNMENT TABLE

	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
SECCn_0	~P2	~P2'	~P1	~P1'	~P16	~P16'	~P8	~P8'
SECCn_1	1	1	1	1	1	1	~P4	~P4'

5.3 1-BIT ECC MODULE FEATURES

Generation of 1-bit ECC is controlled by the ECC Lock (MainECCLock, SpareECCLock) bit of the Control register. If ECCLock is Low, H/W ECC modules generate ECC codes.

1-bit ECC Register Configuration

Following tables shows the configuration of 1-bit ECC value read from spare area of external NAND Flash memory. To compare ECC parity code generated by the H/W modules, the format of ECC read from memory is important.

NOTE

4-bit/ 8-bit ECC decoding scheme is different from 1bit ECC.

1) 8-bit NAND Flash Memory Interface

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFMECCD0	4th ECC for I/O[7:0]	3rd ECC for I/O[7:0]	2nd ECC for I/O[7:0]	1st ECC for I/O[7:0]
NFMECCD1	Not used			

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFSECCD	Not used		2nd ECC for I/O[7:0]	1st ECC for I/O[7:0]

5.4 1-BIT ECC PROGRAMMING GUIDE

1. To use SLC ECC in software mode, reset the ECCType to '0' (enable SLC ECC)'. ECC module generates ECC parity code for all read/ write data if MainECCLock (NFCON[7]) and SpareECCLock (NFCON[6]) are unlocked('0'). You must reset ECC value by writing the InitMECC (NFCONT[5]) and InitSECC (NFCON[4]) bit as '1' and clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before read or write data. MainECCLock (NFCONT[7]) and SpareECCLock(NFCONT[6]) bit controls the generation of ECC parity code.
2. The ECC module generates ECC parity code on register NFMECC0/1 whenever data is read or written.
3. After you complete read or write one page (does not include spare area data), Set the MainECCLock bit to '1' (Lock). ECC Parity code is locked and the value of the ECC status register does not change.
4. To generate spare area ECC parity code, Clear SpareECCLock (NFCONT[6]) bit to '0' (Unlock).
5. The spare area ECC module generates ECC parity code on register NFSECC whenever data is read or written.
6. After you complete read or write spare area, set the SpareECCLock bit to '1' (Lock). ECC Parity code is locked and the value of the ECC status register does not change.
7. From now on, you can use these values to record to the spare area or check the bit error.
8. For example, to check the bit error of main data area on page read operation, you must move the ECC parity codes (is stored to spare area) to NFMECCD0 and NFMECCD1 after ECC codes for main data area is generated. From this point, the NFECCERR0 and NFECCERR1 have the valid error status values.

NOTE

NFSECCD is for ECC in the spare area (Usually, the user writes the ECC value generated from main data area to Spare area and value is same as NFMECC0/1) and it is generated from the main data area.

5.5 4-BIT ECC PROGRAMMING GUIDE (ENCODING)

1. To use 4-bit ECC in software mode, set the MsgLength to 0 (512-byte message length) and set the ECCType to "10" (enable 4bit ECC). The ECC module generates ECC parity code for 512-byte write data. You must reset the ECC value by writing the InitMECC (NFCONT[5]) bit as '1' and clear the MainECCLock (NFCONT[7]) bit to '0' (Unlock) before write data.
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity.
2. The 4-bit ECC modules generates ECC parity code internally whenever data is written.
3. After writing 512-byte data (not include spare area data), the parity codes are automatically updated to NFMECC0, NFMECC1 register. Use 512-byte NAND Flash memory to program these values to spare area. However, if you use NAND Flash memory more than 512-byte page, you can't program immediately. In this case, copy these parity codes to other memory like DRAM. After main data is written; write the copied ECC values to spare area.
   The parity codes have self-correctable information.
4. To generate spare area ECC parity code, set the MsgLength to 1 (24-byte message length), and set the ECCType to "10" (enable 4bit ECC). The ECC module generates ECC parity code for 24-byte write data. Therefore reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' and clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before write data.
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity.
5. The 4-bit ECC module generates ECC parity code internally whenever data is written.
6. If you write 24-byte meta or extra data, the parity codes are automatically updated to NFMECC0 and NFMECC1 register. Program these parity codes to spare area.
   The parity codes have self-correctable information.

5.6 4-BIT ECC PROGRAMMING GUIDE (DECODING)

1. To use 4-bit ECC in software mode, set the MsgLength to 0 (512-byte message length) and set the ECCType to "10" (enable 4bit ECC). ECC module generates ECC parity code for 512-byte read data. Therefore write the InitMECC (NFCONT[5]) bit as '1' to reset ECC value and clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before read data.
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity code
2. The 4-bit ECC modules generates ECC parity code internally whenever data is read.
3. After 512-byte (does not include spare area data) is read, you must read parity codes. MLC ECC module needs parity codes to detect whether error bits are or not. Therefore read ECC parity code right after read 512-byte. Once ECC parity code is read, 4-bit ECC engine starts to search any error internally. 4-bit ECC error searching engine need minimum 155 cycles to find any error. During this time, continue read main data from external NAND Flash memory. ECCDecDone(NFSTAT[6]) is used to check whether ECC decoding is complete.
4. If ECCDecDone (NFSTAT[6]) is set ('1'), NFECCERR0 indicates whether error bit exist or not. If any error exists, fix it by referring NFECCERR0/1 and NFMLCBITPT register.
5. If there is more main data to be read, continue from step 1 for remaining data.
6. For meta data error check, set the MsgLength to 1(24-byte message length) and set the ECCType to '1' (enable 4-bit ECC). ECC module generates ECC parity code for 512-byte read data. Therefore reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' and clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before read data.
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity code.
7. The 4-bit ECC modules generates ECC parity code internally whenever data is read.
8. After 512-byte (not include spare area data) is read, you must read parity codes. 4-bit ECC module needs parity codes to detect whether error bits are or not. Therefore read ECC parity codes right after read 512-byte. Once ECC parity code is read, 4-bit ECC engine starts to search any error internally. 4-bit ECC error searching engine need minimum 155 cycles to find any error. During this time, continue read main data from external NAND Flash memory. The ECCDecDone(NFSTAT[6]) is used to check whether ECC decoding is complete.
9. If ECCDecDone (NFSTAT[6]) is set ('1'), NFECCERR0 indicates whether error bit exist or not. If any error exists, fix it by referring NFECCERR0/1 and NFMLCBITPT register.

5.7 8-BIT ECC PROGRAMMING GUIDE (ENCODING)

1. To use 8-bit ECC in software mode, set the MsgLength to 0 (512-byte message length) and set the ECCType to "01" (enable 8-bit ECC). The ECC module generates ECC parity code for 512-byte write data. Therefore reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' after clearing the MainECCLock (NFCONT[7]) bit to '0'(Unlock) .
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity code.

   Note: In 8-bit ECC, MainECCLock should be cleared before InitMECC.

2. The 8-bit ECC module generates ECC parity code internally whenever data is written.
3. After 512-byte data (not include spare area data) is written, the parity codes are automatically updated to NF8MECC0, NFMECC1, NF8MECC2, and NF8MECC3 registers. Use 512-byte NAND flash memory to program these values to spare area. However, if more than 512-byte page NAND flash memory is used, s/w can't program immediately. In this case, you must copy these parity codes to other memory like DRAM. After writing all main data, write the copied ECC values to spare area.
   The parity codes have self-correctable information include parity code itself.
4. To generate spare area ECC parity code, set the MsgLength to 1(24-byte message length), and set the ECCType to "01"(enable 8bit ECC). The ECC module generates ECC parity code for 24-byte write data. Therefore reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' after clearing the MainECCLock (NFCONT[7]) bit to '0'(Unlock.
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity code.

   Note: In 8-bit ECC, MainECCLock should be cleared before InitMECC.

5. The 8-bit ECC module generates ECC parity code internally whenever data is written.
6. If you finish writing 24-byte meta or extra data, the parity codes are automatically updated to NF8MECC0, NFMECC1, NF8MECC2, and NF8MECC3 registers. Program these parity codes to spare area.The parity codes have self-correctable information including parity code itself.

5.8 8-BIT ECC PROGRAMMING GUIDE (DECODING)

1. To use 8-bit ECC in software mode, set the MsgLength to 0 (512-byte message length) and set the ECCType to "01" (enable 8bit ECC). The ECC module generates ECC parity code for 512-byte read data. Therefore reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' after clearing the MainECCLock (NFCONT[7]) bit to '0'(Unlock.
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity code.

   Note: In 8-bit ECC, MainECCLock should be cleared before InitMECC

2. The MLC ECC module generates ECC parity code internally whenever data is read.
3. After 512-byte (does not include spare area data) data is read, set the MainECCLock (NFCONT[7]) bit to '1'(Lock) and read parity codes. The 8-bit ECC module needs parity codes to detect whether error bits are generated or not. Therefore read ECC parity code right after 512-byte is read. Once ECC parity code is read, MLC ECC engine start to search internal error. 8-bit ECC error searching engine need minimum 372 cycles to find any error. During this time, continue read main data from external NAND Flash memory.
   ECCDecDone(NFSTAT[6]) is used to check whether ECC decoding is complete.
4. If ECCDecDone (NFSTAT[6]) is set ('1'), NF8ECCERR0 indicates whether error bit exist or not. If any error exists, fix it by referring NF8ECCERR0/1/2 and NFMLC8BITPT0/1 registers.
5. If more main data has to be read, continue from step 1.
6. For meta data error check, set the MsgLength to 1(24-byte message length) and set the ECCType to "01" (enable 8-bit ECC). The ECC module generates ECC parity code for 512-byte read data. Therefore reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' after clearing the MainECCLock (NFCONT[7]) bit to '0'(Unlock).
   MainECCLock (NFCONT[7]) bit controls the generation of ECC parity code.
7. The 8-bit ECC module generates ECC parity code internally whenever data is read.
8. After 512-byte data (not include spare area data) is read, set the MainECCLock (NFCONT[7]) bit to '1'(Lock) and read parity codes. The MLC ECC module needs parity codes to detect whether error bits are generated or not. Therefore read ECC parity codes right after read 512-byte. Once ECC parity code is read, 8-bit ECC engine starts to search internal error. 8-bit ECC error searching engine need minimum 372 cycles to find any error. During this time, continue read main data from external NAND Flash memory.
   ECCDecDone(NFSTAT[6]) is used to check whether ECC decoding is complete.
9. If ECCDecDone (NFSTAT[6]) is set ('1'), NF8ECCERR0 indicates whether error bit exist or not. If error exists, fix it by referring NF8ECCERR0/1/2 and NF8MLCBITPT registers.

6 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
NF_DATA[15:0]	Input / Output	Address/ Data Bus	Xm0DATA	muxed
NF_RnB[3:0]	Input	Ready and Busy	Xm0FRnB	muxed
NF_CLE	Output	Command Latch Enable	Xm0FCLE	muxed
NF_ALE	Output	Address Latch Enable	Xm0FALE	muxed
NF_nCS[3:0]	Output	Chip Enable	Xm0CSn	muxed
NF_REn	Output	Read Enable	Xm0FREn	muxed
NF_WEn	Output	Write Enable	Xm0FWEn	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

7 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
NFCONF	0xE720_0000	R/W	Configuration Register	0xX000100X
NFCONT	0xE720_0004	R/W	Control Register	0x000100C6
NFCMMD	0xE720_0008	R/W	Command Register	0x00000000
NFADDR	0xE720_000C	R/W	Address Register	0x00000000
NFDATA	0xE720_0010	R/W	Data Register	0x00000000
NFMECCD0	0xE720_0014	R/W	1st and 2nd main ECC Data Register	0x00000000
NFMECCD1	0xE720_0018	R/W	3rd and 4th main ECC Data Register	0x00000000
NFSECCD	0xE720_001C	R/W	Spare ECC Read Register	0x00000000
NFSBLK	0xE720_0020	R/W	Programmable Start Block Address Register	0x00000000
NFEBLK	0xE720_0024	R/W	Programmable End Block Address Register	0x00000000
NFSTAT	0xE720_0028	R/W	NAND Status Register	0x0080001D
NFECCERR0	0xE720_002C	R	ECC Error Status0 Register	0x007FFFFA
NFECCERR1	0xE720_0030	R	ECC Error Status1 Register	0x00000000
NFMECC0	0xE720_0034	R	Generated ECC Status0 Register	0xFFFFFFFF
NFMECC1	0xE720_0038	R	Generated ECC Status1 Register	0xFFFFFFFF
NFSECC	0xE720_003C	R	Generated Spare Area ECC Status Register	0x03FF03FF
NFMLCBITPT	0xE720_0040	R	4-bit ECC Error Bit Pattern Register	0x00000000
NF8ECCERR0	0xE720_0044	R	8bit ECC Error Status0 Register	0x40000000
NF8ECCERR1	0xE720_0048	R	8bit ECC Error Status1 Register	0x00000000
NF8ECCERR2	0xE720_004C	R	8bit ECC Error Status2 Register	0x00000000
NFM8ECC0	0xE720_0050	R	Generated ECC Status0 Register	0x00000000
NFM8ECC1	0xE720_0054	R	Generated ECC Status0 Register	0x00000000
NFM8ECC2	0xE720_0058	R	Generated ECC Status0 Register	0x00000000
NFM8ECC3	0xE720_005C	R	Generated ECC Status0 Register	0x00000000
NFMLC8BITPT0	0xE720_0060	R	4-bit ECC Error Bit Pattern Register	0x00000000
NFMLC8BITPT1	0xE720_0064	R	4-bit ECC Error Bit Pattern Register	0x00000000
NFACTADJ	0xE720_0068	R/W	NFCON Access Timing Adjustment Register	0x00000000

7.1 NAND FLASH CONFIGURATION REGISTER (NFCONF, R/W, ADDRESS = 0XE720_0000)

NFCONF	Bit	Description	Reset Value
Reserved	[31]	Reserved	0
MLCClkCtrl	[30]	Clock control for 4-bit ECC & 8-bit ECC engine. 0: Recommended when system clock is more than 66MHz. 1: Recommended when system clock is less than 66MHz	0
Reserved	[30:26]	Reserved	000
MsgLength	[25]	0 = 512 byte Message Length 1 = 24 byte Message Length	0
ECCType	[24:23]	This bit indicates what kind of ECC should be used. 00 = 1-bit ECC 10 = 4-bit ECC 01 = 8-bit ECC	0
Reserved	[22:15]	Reserved	000000000
TACLS	[14:12]	CLE & ALE duration setting value (0~7) Duration = HCLK x TACLS	001
Reserved	[11]	Reserved	0
TWRPH0	[10:8]	TWRPH0 duration setting value (0~7) Duration = HCLK x (TWRPH0 + 1)	000
Reserved	[7]	Reserved	0
TWRPH1	[6:4]	TWRPH1 duration setting value (0~7) Duration = HCLK x (TWRPH1 + 1)	000
MLCFlash	[3]	This bit indicates type of NAND Flash memory used. 0 = SLC NAND Flash 1 = MLC NAND Flash	0
PageSize	[2]	This bit indicates the page size of NAND Flash Memory, If MLCFlash is 0, the value of PageSize is following: 0 = 2048 Bytes/page 1 = 512 Bytes/page If MLCFlash is 1, the value of PageSize is following: 0 = 4096 Bytes/page 1 = 2048 Bytes/page	0
AddrCycle	[1]	This bit indicates the number of Address cycle of NAND Flash memory. If Page Size is 512 Bytes, 0 = 3 address cycle 1 = 4 address cycle If page size is 2K or 4K, 0 = 4 address cycle 1 = 5 address cycle	0
Reserved	[0]	Reserved	0

7.2 CONTROL REGISTER (NFCONT, R/W, ADDRESS = 0XE720_0004)

NFCONT	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
Reg_nCE3	[23]	NAND Flash Memory nRCS[3] signal control 0 = Force nRCS[3] to low (Enable chip select) 1 = Force nRCS[3] to High (Disable chip select)	1
Reg_nCE2	[22]	NAND Flash Memory nRCS[2] signal control 0 = Force nRCS[2] to low (Enable chip select) 1 = Force nRCS[2] to High (Disable chip select)	1
Reserved	[21:19]	Reserved	0
MLCEccDirection	[18]	4-bit, 8-bitECC encoding/ decoding control 0 = Decoding 4-bit, 8bit ECC, It is used for page read 1 = Encoding 4-bit, 8-bit ECC, It is be used for page program	0
LockTight	[17]	Lock-tight configuration 0 = Disable lock-tight 1 = Enable lock-tight, If this bit is set to 1, you cannot clear this bit. Only when reset or wake up from sleep mode can clear this bit. (It is impossible to clear LockTight by software). If this bit is set to 1, the location excluding the area between NFSBLK (0xE720_0020) and NFEBLK (0xE720_0024)-1 is locked, write or erase to the area is invalid and only read access is possible. If you try to write or erase locked area, the illegal access occurs (NFSTAT [3] bit will be set). If the NFSBLK and NFEBLK are same, entire area is locked.	0
LOCK	[16]	Soft Lock configuration 0 = Disables lock 1 = Enables lock Soft lock area is modified any time by software. If this bit is set to 1, the location excluding the area between NFSBLK (0xE720_0020) and NFEBLK (0xE720_0024)-1 is locked, write or erase to the area is invalid and only read access is possible. If you try to write or erase locked area, the illegal access occurs (NFSTAT [3] bit will be set). If the NFSBLK and NFEBLK are same, entire area is locked.	1
Reserved	[15:14]	Reserved	00
EnbMLCEncInt	[13]	4-bit, 8-bit ECC encoding completion interrupt control 0 = Disables interrupt 1 = Enables interrupt	0
EnbMLCDecInt	[12]	4-bit, 8-bit ECC decoding completion interrupt control 0 = Disables interrupt	0

NFCONT	Bit	Description	Reset Value
		1 = Enables interrupt	
MLCStop	[11]	8-bit ECC encoding/ decoding operation initialization	0
EnbIllegalAccINT	[10]	Illegal access interrupt control 0 = Disables interrupt 1 = Enables interrupt Illegal access interrupt occurs if CPU tries to program or erase locking area (the area setting in NFSBLK (0xE720_0020) to NFEBLK (0xE720_0024)-1.	0
EnbRnBINT	[9]	RnB status input signal transition interrupt control 0 = Disables RnB interrupt 1 = Enables RnB interrupt	0
RnB_TransMode	[8]	RnB transition detection configuration 0 = Detects rising edge 1 = Detects falling edge	0
MECCLock	[7]	Lock Main area ECC generation 0 = Unlocks Main area ECC 1 = Locks Main area ECC Main area ECC status register is NFMECC0/NFMECC1(0xE720_0034/0xE720_0038),	1
SECCLock	[6]	Lock Spare area ECC generation. 0 = Unlocks Spare ECC 1 = Locks Spare ECC Spare area ECC status register is NFSECC(0xE720_003C),	1
InitMECC	[5]	1 = Initialize main area ECC decoder/ encoder (write-only)	0
InitSECC	[4]	1 = Initialize spare area ECC decoder/ encoder (write-only)	0
HW_nCE	[3]	Reserved (HW_nCE)	0
Reg_nCE1	[2]	NAND Flash Memory nRCS[1] signal control	1
Reg_nCE0	[1]	NAND Flash Memory nRCS[0] signal control 0 = Force nRCS[0] to low (Enables chip select) 1 = Force nRCS[0] to High (Disables chip select)	1
MODE	[0]	NAND Flash controller operating mode 0 = Disables NAND Flash Controller 1 = Enables NAND Flash Controller	0

7.3 COMMAND REGISTER (NFCMMD, R/W, ADDRESS = 0XE720_0008)

NFCMMD	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x000000
REG_CMMD	[7:0]	NAND Flash memory command value	0x00

7.4 ADDRESS REGISTER (NFADDR, R/W, ADDRESS = 0XE720_000C)

NFADDR	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x000000
REG_ADDR	[7:0]	NAND Flash memory address value	0x00

7.5 DATA REGISTER (NFDATA, R/W, ADDRESS = 0XE720_0010)

NFDATA	Bit	Description	Reset Value
NFDATA	[31:0]	NAND Flash read/ program data value for I/O **Note:** Refer to DATA REGISTER CONFIGURATION in page 6-5.	0x00000000

7.6 MAIN DATA AREA ECC REGISTER (NFMECCD0, R/W, ADDRESS = 0XE720_0014)

NFMECCD0	Bit	Description	Reset Value
ECCData1_1	[31:24]	2nd ECC for I/O[15:8]	0x00
ECCData1_0 (ECC1)	[23:16]	2nd ECC for I/O[7:0]	0x00
ECCData0_1	[15:8]	1st ECC for I/O[15:8]	0x00
ECCData0_0 (ECC0)	[7:0]	1st ECC for I/O[7:0]	0x00

NOTE: Only word access is valid.

7.7 MAIN DATA AREA ECC REGISTER (NFMECCD0, R/W, ADDRESS = 0XE720_0018)

NFMECCD1	Bit	Description	Reset Value
ECCData3_1	[31:24]	4th ECC for I/O[15:8]	0x00
ECCData3_0 (ECC3)	[23:16]	4th ECC for I/O[7:0]	0x00
ECCData2_1	[15:8]	3rd ECC for I/O[15:8]	0x00
ECCData2_0 (ECC2)	[7:0]	3rd ECC for I/O[7:0]	0x00

NOTE: Only word access is valid.

7.8 SPARE AREA ECC REGISTER (NFSECCD, R/W, ADDRESS = 0XE720_001C)

NFSECCD	Bit	Description	Reset Value
SECCData1_1	[31:24]	2nd ECC for I/O[15:8]	0x00
SECCData1_0	[23:16]	2nd ECC for I/O[7:0]	0x00
SECCData0_1	[15:8]	1st ECC for I/O[15:8]	0x00
SECCData0_0	[7:0]	1st ECC for I/O[7:0]	0x00

NOTE: Only word access is valid.

7.9 PROGRMMABLE BLOCK ADDRESS REGISTER (NFSBLK, R/W, ADDRESS = 0XE720_0020)

NFSBLK	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
SBLK_ADDR2	[23:16]	The 3rd block address of the block erase operation	0x00
SBLK_ADDR1	[15:8]	The 2nd block address of the block erase operation	0x00
SBLK_ADDR0	[7:0]	The 1st block address of the block erase operation (Only bit [7:5] are valid)	0x00

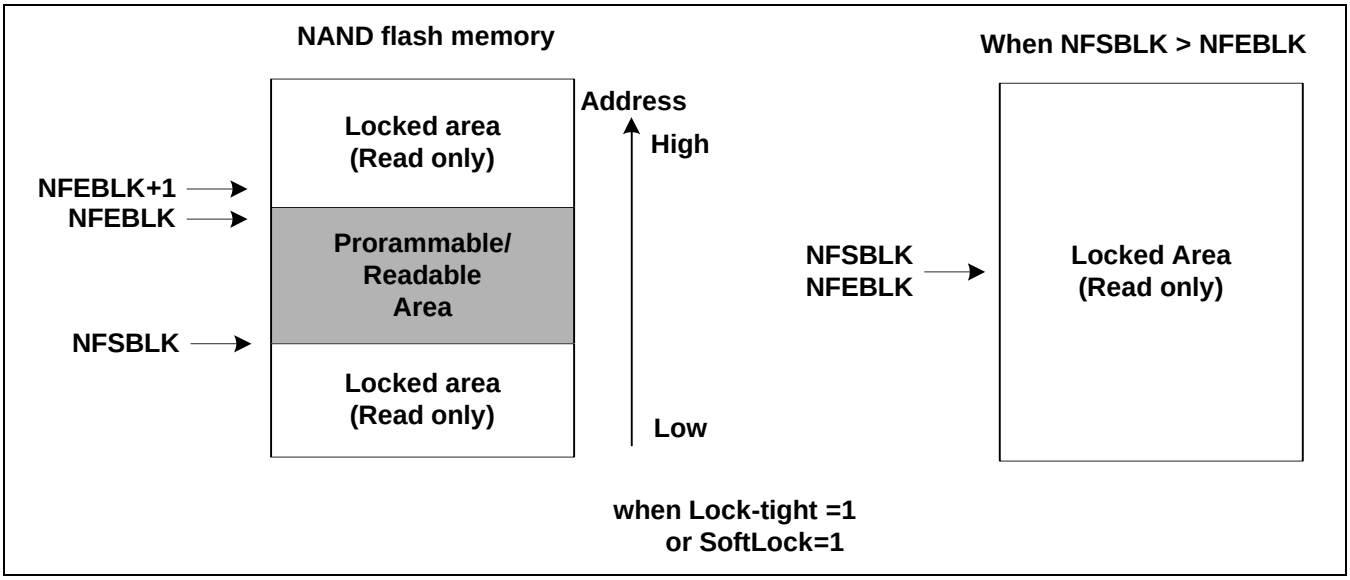
NOTE: Advance Flash's block Address start from 3-address cycle. Therefore block address register needs 3-bytes.

7.10 PROGRMMABLE BLOCK ADDRESS REGISTER (NFEBLK, R/W, ADDRESS = 0XE720_0024)

NFEBLK	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
EBLK_ADDR2	[23:16]	The 3rd block address of the block erase operation	0x00
EBLK_ADDR1	[15:8]	The 2nd block address of the block erase operation	0x00
EBLK_ADDR0	[7:0]	The 1st block address of the block erase operation (Only bit [7:5] are valid)	0x00

NOTE: Advance Flash's block Address start from 3-address cycle. Therefore block address register needs 3-bytes.

The NFSLK and NFEBLK is changed while Soft lock bit (NFCONT[16]) is enabled. They cannot be changed if Lock-tight bit(NFCONT[17]) is set.



7.11 NFCON STATUS REGISTER (NFSTAT, R/W, ADDRESS = 0XE720_0028)

NFSTAT	Bit	Description	Reset Value
Flash_RnB_GRP	[31:28]	The status of RnB[3:0] input pin. 0 = NAND Flash memory busy 1 = NAND Flash memory ready to operate	0x0
RnB_TransDetect _GRP	[27:24]	If RnB[3:0] low to high transition is occurs, this field is set. To clear this write '1'. 0 = RnB transition is not detected 1 = RnB transition is detected Transition configuration is set in RnB_TransMode(NFCONT[8]).	
Reserved	[23:12]	Reserved	0x800
Flash_nCE[3:0] (Read-only)	[11:8]	The status of nCE[3:0] output pin	0x0
MLCEncodeDone	[7]	If 4-bit ECC or 8-bit ECC encodng is finished, this field is set and issues interrupt. The NFMLCECC0 and NFMLCECC1 have valid values. To clear this, write '1'. 1 = 4-bit ECC or 8-bit ECC encoding is completed	0
MLCDecodeDone	[6]	If 4-bit ECC or 8-bit ECC decoding is finished, this field set and issues interrupt. The NFMLCBITPT, NFMLCL0 and NFMLCEL1 have valid values. To clear this, write '1'. 1 = 4-bit ECC or 8-bit ECC decoding is completed	0
IllegalAccess	[5]	Once Soft Lock or Lock-tight is enabled, The illegal access (program, erase) to the memory set this bit. 0 = illegal access is not detected 1 = illegal access is detected To clear this value, write 1 to this bit.	0
RnB_TransDetect	[4]	If RnB low to high transition occurs, this field is set and issues interrupt. To clear this write '1'. 0 = RnB transition is not detected 1 = RnB transition is detected Transition configuration is set in RnB_TransMode(NFCONT[8]).	1
Flash_nCE[1] (Read-only)	[3]	The status of nCE[1] output pin	1
Flash_nCE[0] (Read-only)	[2]	The status of nCE[0] output pin	1
Reserved	[1]	Reserved	0
Flash_RnB (Read-only)	[0]	The status of RnB input pin. 0 = NAND Flash memory busy 1 = NAND Flash memory ready to operate	1

7.12 ECC0/1 ERROR STATUS REGISTER (NFECCERR0, R, ADDRESS = 0XE720_002C)

- **If ECCType is 1bit ECC**

NFECCERR0	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0x00
ECC0SDataAddr	[24:21]	In spare area, Indicates which number data has error	0011
ECC0SBitAddr	[20:18]	In spare area, Indicates which bit has error	111
ECC0DataAddr	[17:7]	In main data area, Indicates which number data has error	0x7FF
ECC0BitAddr	[6:4]	In main data area, Indicates which bit has error	111
ECC0SprErrNo	[3:2]	Indicates whether spare area bit fail error occurred 00 = No Error 01 = 1-bit error(correctable) 10 = Multiple error 11 = ECC area error	10
ECC0MainErrNo	[1:0]	Indicates whether main data area bit fail error occurred 00 = No Error 01 = 1-bit error(correctable) 10 = Multiple error 11 = ECC area error	10

NOTE: The above values are valid if both ECC register and ECC status register have valid value.

- **If ECCType is 4bit ECC**

NFECCERR0	Bit	Description	Reset Value
MLCECCBusy	[31]	Indicates that the 4-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy	0
MLCECCReady	[30]	ECC Ready bit	1
MLCFreePage	[29]	Indicates that the page data read from NAND flash has all 'FF' value.	0
MLCECCError	[28:26]	4-bit ECC decoding result 000 = No error 001 = 1-bit error 010 = 2-bit error 011 = 3-bit error 100 = 4-bit error 101 = Uncorrectable 11x = reserved	000
MLCErrLocation2	[25:16]	Error byte location of 2nd bit error	0x000
Reserved	[15:10]	Reserved	0x00
MLCErrLocation1	[9:0]	Error byte location of 1st bit error	0x000

NOTE: These values are updated if ECCDecodeDone (NFSTAT[6]) is set ('1').

7.13 ECC0/1 ERROR STATUS REGISTER (NFECCERR1, R, 0XE720_0030)

- **If ECCType is 1bit ECC**

NFECCERR1	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0x00
ECC1SDataAddr	[24:21]	In spare area, Indicates which number data has error	0000
ECC1SBitAddr	[20:18]	In spare area, Indicates which bit has error	000
ECC1DataAddr	[17:7]	In main data area, Indicates which number data has error	0x000
ECC1BitAddr	[6:4]	In main data area, Indicates which bit has error	000
ECC1SprErrNo	[3:2]	Indicates whether spare area bit fail error occurred 00 = No Error 01 = 1-bit error(correctable) 10 = Multiple error 11 = ECC area error	00
ECC1MainErrNo	[1:0]	Indicates whether main data area bit fail error occurred 00 = No Error 01 = 1-bit error(correctable) 10 = Multiple error 11 = ECC area error	00

NOTE: The above values are valid if both ECC register and ECC status register have valid value.

- **If ECCType is 4bit ECC**

NFECCERR1	Bit	Description	Reset Value
Reserved	[31:26]	Reserved	0x00
MLCErrLocation4	[25:16]	Error byte location of 4^{th} bit error	0x00
Reserved	[15:10]	Reserved	0x00
MLCErrLocation3	[9:0]	Error byte location of 3^{rd} bit error	0x000

NOTE: These values are updated if ECCDecodeDone (NFSTAT[6]) is set ('1').

7.14 MAIN DATA AREA ECC0 STATUS REGISTER (NFMECC0, R, ADDRESS = 0XE720_0034)

- **If ECCType is 1 bit ECC.**

NFMECC0	Bit	Description	Reset Value
MECC0_3	[31:24]	ECC3 for data[7:0]	0xFF
MECC0_2	[23:16]	ECC2 for data[7:0]	0xFF
MECC0_1	[15:8]	ECC1 for data[7:0]	0xFF
MECC0_0	[7:0]	ECC0 for data[7:0]	0xFF

NOTE: The NAND flash controller generate NFMECC0/1 if read or write main area data while the MainECCLock(NFCONT[7]) bit is '0'(Unlock).

- **If ECCType is 4 bit ECC.**

NFMECC0	Bit	Description	Reset Value
4th Parity	[31:24]	4th Check Parity generated from main area (512-byte)	0x00
3rd Parity	[23:16]	3rd Check Parity generated from main area (512-byte)	0x00
2nd Parity	[15:8]	2nd Check Parity generated from main area (512-byte)	0x00
1st Parity	[7:0]	1st Check Parity generated from main area (512-byte)	0x00

NOTE: The NAND flash controller generate these ECC parity codes if write main area data while the MainECCLock (NFCON[7]) bit is '0'(unlock).

7.15 MAIN DATA AREA ECC0 STATUS REGISTER (NFMECC1, R, ADDRESS = 0XE720_0038)

- **If ECCType is 1 bit ECC.**

NFMECC1	Bit	Description	Reset Value
MECC1_3	[31:24]	ECC3 data[15:8]	0xFF
MECC1_2	[23:16]	ECC2 data[15:8]	0xFF
MECC1_1	[15:8]	ECC1 data[15:8]	0xFF
MECC1_0	[7:0]	ECC0 data[15:8]	0xFF

NOTE: The NAND flash controller generates NFMECC0/1 if read or write main area data while the MainECCLock(NFCONT[7]) bit is '0'(Unlock).

- **If ECCType is 4 bit ECC.**

NFMECC1	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
7^{th} Parity	[23:16]	7^{th} Check Parity generated from main area (512-byte)	0x00
6^{th} Parity	[15:8]	6^{th} Check Parity generated from main area (512-byte)	0x00
5^{th} Parity	[7:0]	5^{th} Check Parity generated from main area (512-byte)	0x00

NOTE: The NAND flash controller generates these ECC parity codes if write main area data while the MainECCLock (NFCON[7]) bit is '0'(unlock).

7.16 SPARE AREA ECC STATUS REGISTER (NFSECC, R, ADDRESS = 0XE720_003C)

NFSECC	Bit	Description	Reset Value
SECC1_1	[31:24]	Spare area ECC1 Status for I/O[15:8]	0x03
SECC1_0	[23:16]	Spare area ECC0 Status for I/O[15:8]	0xFF
SECC0_1	[15:8]	Spare area ECC1 Status for I/O[7:0]	0x03
SECC0_0	[7:0]	Spare area ECC0 Status for I/O[7:0]	0xFF

NOTE: The NAND flash controller generates NFSECC if read or write spare area data while the SpareECCLock(NFCONT[6]) bit is '0'(Unlock).

7.17 MLC 4-BIT ECC ERROR PATTEN REGISER (NFMLCBITPT, R, ADDRESS = 0XE720_0040)

NFMLCBITPT	Bit	Description	Reset Value
4^{th} Error bit pattern	[31:24]	4^{th} Error bit pattern	0x00
3^{rd} Error bit pattern	[23:16]	3^{rd} Error bit pattern	0x00
2^{nd} Error bit pattern	[15:8]	2^{nd} Error bit pattern	0x00
1^{st} Error bit pattern	[7:0]	1^{st} Error bit pattern	0x00

7.18 ECC0/1/2 FOR 8BIT ECC STATUS REGISTER (NFECCERR0, R, ADDRESS = 0XE720_0044)

NFECCERR0	Bit	Description	Reset Value
MLC8ECCBusy	[31]	Indicates that the 8-bit ECC decoding engine is searching whether a error exists or not 0 = Idle 1 = Busy	0
MLC8ECCReady	[30]	ECC Ready bit	1
MLC8FreePage	[29]	Inidicates that the page data read from NAND flash has all 'FF' value.	0
MLC8ECCError	[28:25]	8-bit ECC decoding result 0000 = No error 0001 = 1-bit error 0010 = 2-bit error 0011 = 3-bit error 0100 = 4-bit error 0101 = 5-bit error 0110 = 6-bit error 0111 = 7-bit error 1000 = 8-bit error 1001 = Uncorrectable 1010 ~1111 = reserved	0000
MLC8ErrLocation2	[24:15]	Error byte location of 2nd bit error	0x000
Reserved	[14:10]	Reserved	0x00
MLC8ErrLocation1	[9:0]	Error byte location of 1st bit error	0x000

NOTE: These values are updated if ECCDecodeDone (NFSTAT[6]) is set ('1').

7.19 ECC0/1/2 FOR 8BIT ECC STATUS REGISTER (NFECCERR1, R, ADDRESS = 0XE720_0048)

NFECCERR1	Bit	Description	Reset Value
MLCErrLocation5	[31:22]	Error byte location of 5th bit error	0x000
Reserved	[21]	Reserved	0
MLCErrLocation4	[20:11]	Error byte location of 4th bit error	0x000
Reserved	[10]	Reserved	0
MLCErrLocation3	[9:0]	Error byte location of 3rd bit error	0x000

NOTE: These values are updated if ECCDecodeDone (NFSTAT[6]) is set ('1').

7.20 ECC0/1/2 FOR 8BIT ECC STATUS REGISTER (NFECCERR1, R, ADDRESS = 0XE720_004C)

NFECCERR1	Bit	Description	Reset Value
MLCErrLocation8	[31:22]	Error byte location of 8th bit error	0x000
Reserved	[21]	Reserved	0
MLCErrLocation7	[20:11]	Error byte location of 7th bit error	0x000
Reserved	[10]	Reserved	0
MLCErrLocation6	[9:0]	Error byte location of 6th bit error	0x000

NOTE: These values are updated if ECCDecodeDone (NFSTAT[6]) is set ('1').

7.21 MAIN DATA AREA ECC0 STATUS REGISTER (NFM8ECC0, R, ADDRESS = 0XE720_0050)

NFM8ECC0	Bit	Description	Reset Value
4th Parity	[31:24]	4th Check Parity generated from main area (512-byte)	0x00
3rd Parity	[23:16]	3rd Check Parity generated from main area (512-byte)	0x00
2nd Parity	[15:8]	2nd Check Parity generated from main area (512-byte)	0x00
1st Parity	[7:0]	1st Check Parity generated from main area (512-byte)	0x00

7.22 MAIN DATA AREA ECC0 STATUS REGISTER (NFM8ECC1, R, ADDRESS = 0XE720_0054)

NFM8ECC1	Bit	Description	Reset Value
8th Parity	[31:24]	8th Check Parity generated from main area (512-byte)	0x00
7th Parity	[23:16]	7th Check Parity generated from main area (512-byte)	0x00
6th Parity	[15:8]	6th Check Parity generated from main area (512-byte)	0x00
5th Parity	[7:0]	5th Check Parity generated from main area (512-byte)	0x00

7.23 MAIN DATA AREA ECC0 STATUS REGISTER (NFM8ECC2, R, ADDRESS = 0XE720_0058)

NFM8ECC2	Bit	Description	Reset Value
12th Parity	[31:24]	12th Check Parity generated from main area (512-byte)	0x00
11th Parity	[23:16]	11th Check Parity generated from main area (512-byte)	0x00
10th Parity	[15:8]	10th Check Parity generated from main area (512-byte)	0x00
9th Parity	[7:0]	9th Check Parity generated from main area (512-byte)	0x00

7.24 MAIN DATA AREA ECC0 STATUS REGISTER (NFM8ECC0, R, ADDRESS = 0XE720_005C)

NFM8ECC3	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x000000
13th Parity	[7:0]	13th Check Parity generated from main area (512-byte)	0x00

NOTE: The NAND flash controller generates these ECC parity codes if write main area data while the MainECCLock (NFCON[7]) bit is '0'(unlock).

7.25 MLC 8-BIT ECC ERROR PATTEN REGISTER (NFMLC8BITPT0, R, ADDRESS = 0XE720_0060)

NFMLC8BITPT0	Bit	Description	Reset Value
4th Error bit pattern	[31:24]	4th Error bit pattern	0x00
3rd Error bit pattern	[23:16]	3rd Error bit pattern	0x00
2nd Error bit pattern	[15:8]	2nd Error bit pattern	0x00
1st Error bit pattern	[7:0]	1st Error bit pattern	0x00

7.26 MLC 8-BIT ECC ERROR PATTEN REGISTER (NFMLC8BITPT1, R, ADDRESS = 0XE720_0064)

NFMLC8BITPT1	Bit	Description	Reset Value
8th Error bit pattern	[31:24]	8th Error bit pattern	0x00
7th Error bit pattern	[23:16]	7th Error bit pattern	0x00
6th Error bit pattern	[15:8]	6th Error bit pattern	0x00
5th Error bit pattern	[7:0]	5th Error bit pattern	0x00

7.27 NFCON ACCESS TIMING ADJUSTMENT REGISTER (NFACTADJ, R/W, ADDRESS = 0XE720_0068)

NFACTADJ	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x00
NFACTADJ	[0]	0 : HCLK =< 133MHz 1 : HCLK => 166MHz	0x0

5.5 COMPACT FLASH CONTROLLER

1 OVERVIEW

A Compact Flash Controller (CFC) connects seamlessly to the AHB Bus as a Bus slave and AHB Master. The CFC Subsystem recognizes AHB Bus transactions that target the compact flash card. The Master Interface initiates compact flash card requests to the CFC block requester interface . The Slave interface responds to the requests initiated. To complete the AHB Bus transaction, the CFC drives the appropriate AHB response onto the AHB Bus.

The CFC manages all data transfer operations in and out of CompactFlash memory. CFC is compatible with CF standard specification Revision 3.0. It is also compatible with AMBA 2.0 specification. The CFC supports PC card memory/ IO mode & True-IDE mode. CFC operates in one mode at a time. Default mode is PC card mode. The CFC has a top level SFR that includes a mode select bit.

1.1 FEATURES

The Features supported by the CFC are:

- CompactFlash Specification Revision 3.0
- AMBA specification AHB (2.0)
- PC card memory, PC card I/O, and true IDE modes.
- Support PIO 0~4, MDMA 0~2, UDMA 0~4 modes.
- 8-bit or 16-bit data bus
- PC card controller has 5 word-sized Special Function Registers.
- ATAPI controller has 32 word-sized Special Function Registers.
- ATAPI controller is compatible with the ATA/ATAPI-6 standard.
- AHB master (DMA controller) supports 8 burst and word size transfer.

1.2 BLOCK DIAGRAM

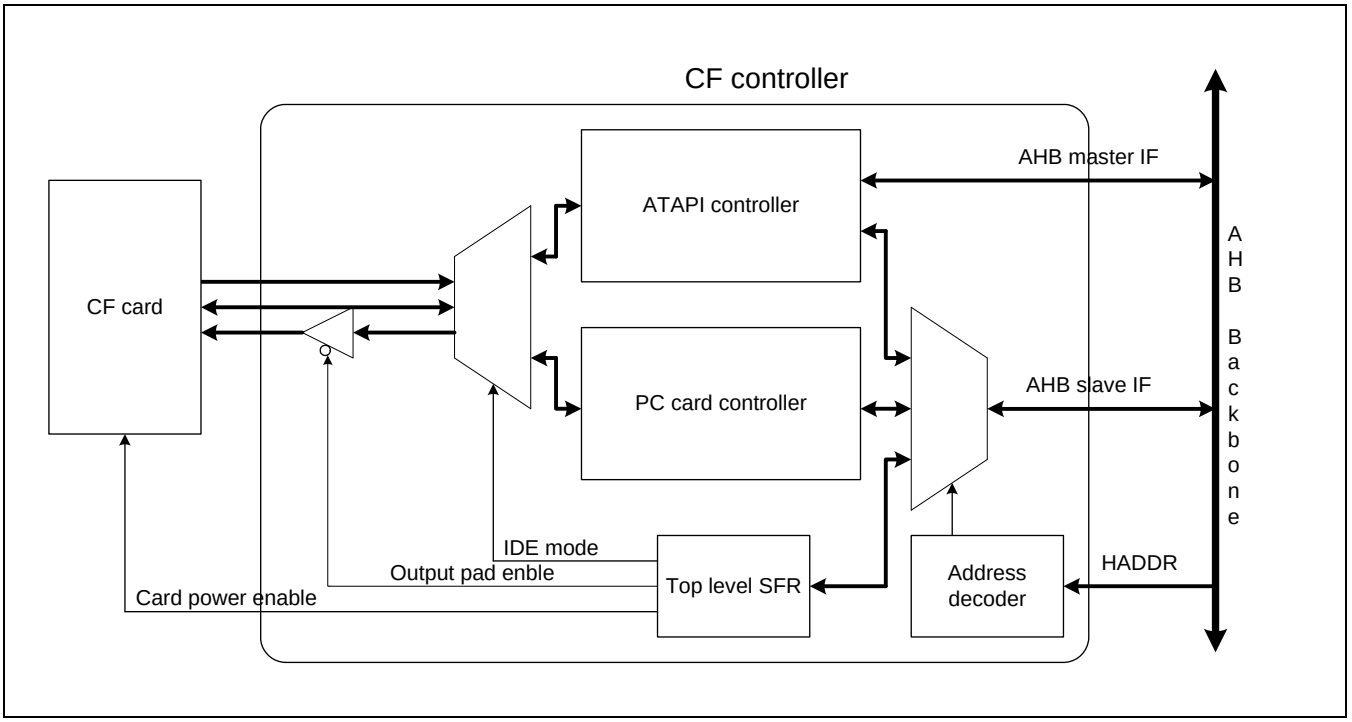


Figure 5.5-1 CFC Block Diagram

1.3 FUNCTIONAL DESCRIPTION

PC Card using memory mode has two distinct memory spaces namely: Attribute memory and Common memory. The Attribute memory holds descriptive information and configuration registers (Card Information Structure CIS). CIS informs about the type of card inserted and is used to configure system to recognize different types of cards and load the correct drivers. This register has status change indication and reporting, 8/16 bits I/O mode selection, interrupt pending status and so on.

The Common memory includes bulk storage of a memory card or device buffers in case of I/O cards. Common memory accesses are 8-bit or 16-bit wide. The task file registers control the ATA disk drive. The task file registers are mapped into common memory space (Task file registers have commands used to control all ATA/IDE drives).

The PC Card in I/O mode has transfers that are 8-bit or 16-bit wide. In PC Card I/O mode, the task file registers are mapped into I/O address space. The value in card option register in attribute mode determines whether the task file registers mapped to common memory or I/O space.

The PC card memory mode use nWE_CF (write enable strobe) and nOE_CF (output enable strobe) to access memory locations. PC card distinguishes between attribute memory and common memory by the signal nREG_CF. If nREG_CF is high, common memory is accessed. If this signal goes low, it access attribute memory. The PC card I/O mode use nIOWR_CF and nIORD_CF to access I/O locations (Refer to Table 5.5-2).

Table 5.5-1 Control Signaling Each Transaction Type

Transaction Type	nIORD	nIOWR	nOE	nWE	nREG
I/O Read	0	1	1	1	0
I/O Write	1	0	1	1	0
Attribute Memory Read	1	1	0	1	0
Attribute Memory Write	1	1	1	0	0
Common Memory Read	1	1	0	1	1
Common Memory Write	1	1	1	0	1

The PC card mode has two half-word (16-bits) write buffers and 4 half-word (16-bits) read buffers. The PC card mode has 5 word-sized (32 bits) Special function Registers. Three timing configuration registers are available for attribute memory, common memory and I/O interface. There is one status and control configuration register, and one interrupt source and mask register.

The CFC is configured to True IDE mode, when the nOE_CF signal is grounded. The ATAPI controller is compatible with the ATA/ATAPI-6 standard. This mode allows I/O operations to the task file and data registers. It has access to one FIFO that is 16X32-bit. The ATAPI controller has internal DMA controller for data transfer between ATA device and memory. The ATAPI controller has 32 word-sized (32-bits) Special Function Registers.

1.4 PC CARD MODE TIMING DIAGRAM

The Figure 5.5-6, Figure 5.5-7, Figure 5.5-8 and Figure 5.5-9 shows the timing diagrams in the PC Card mode.

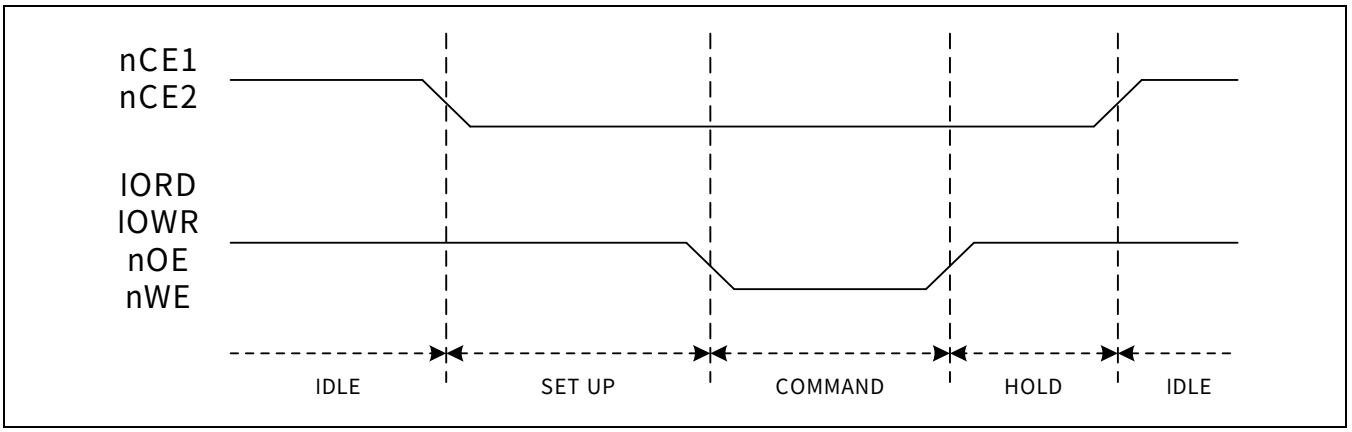


Figure 5.5-2 PC Card State Definition

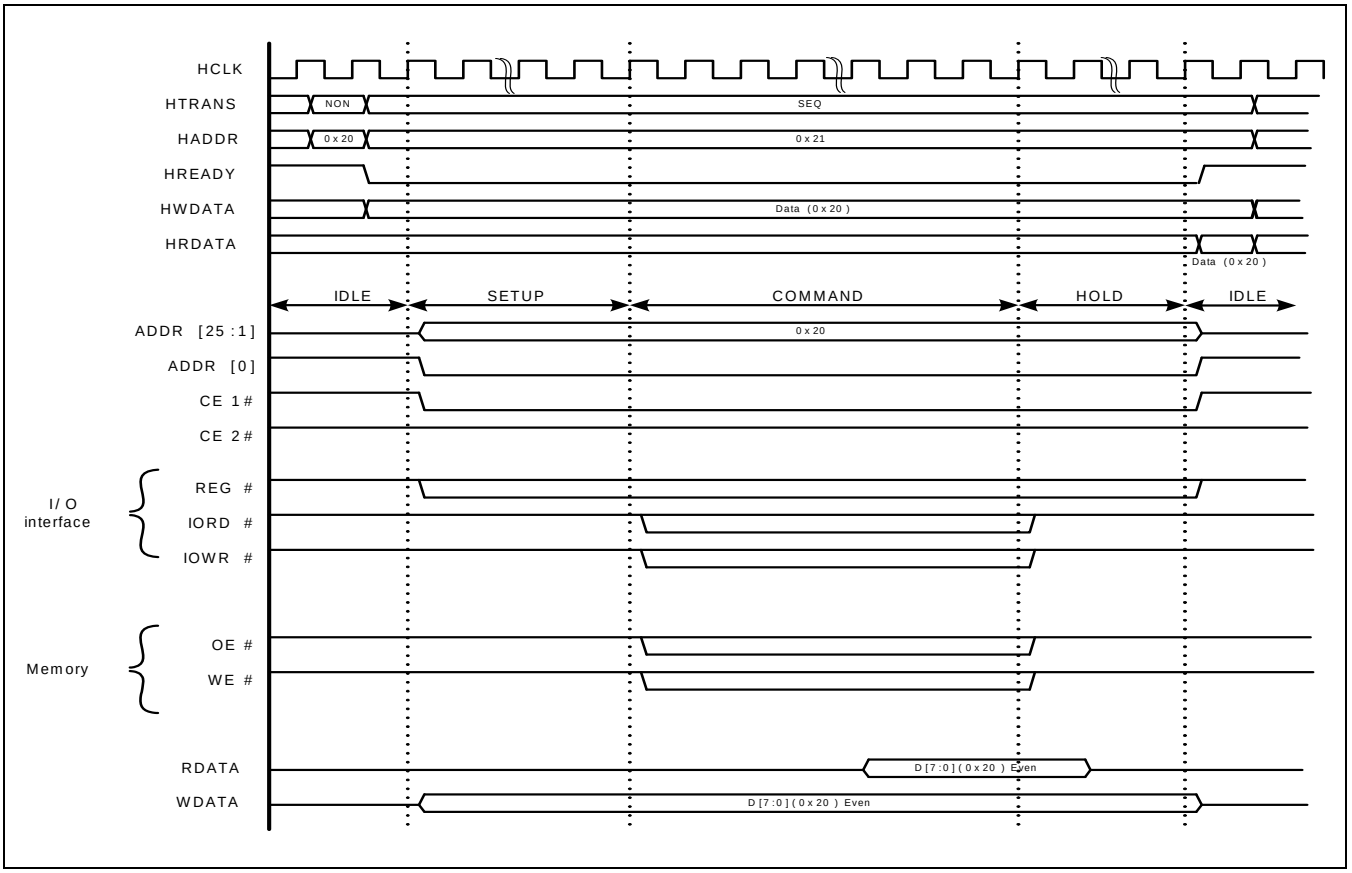


Figure 5.5-3 PC Card 8-bit Transfer (8-bit PC Card)

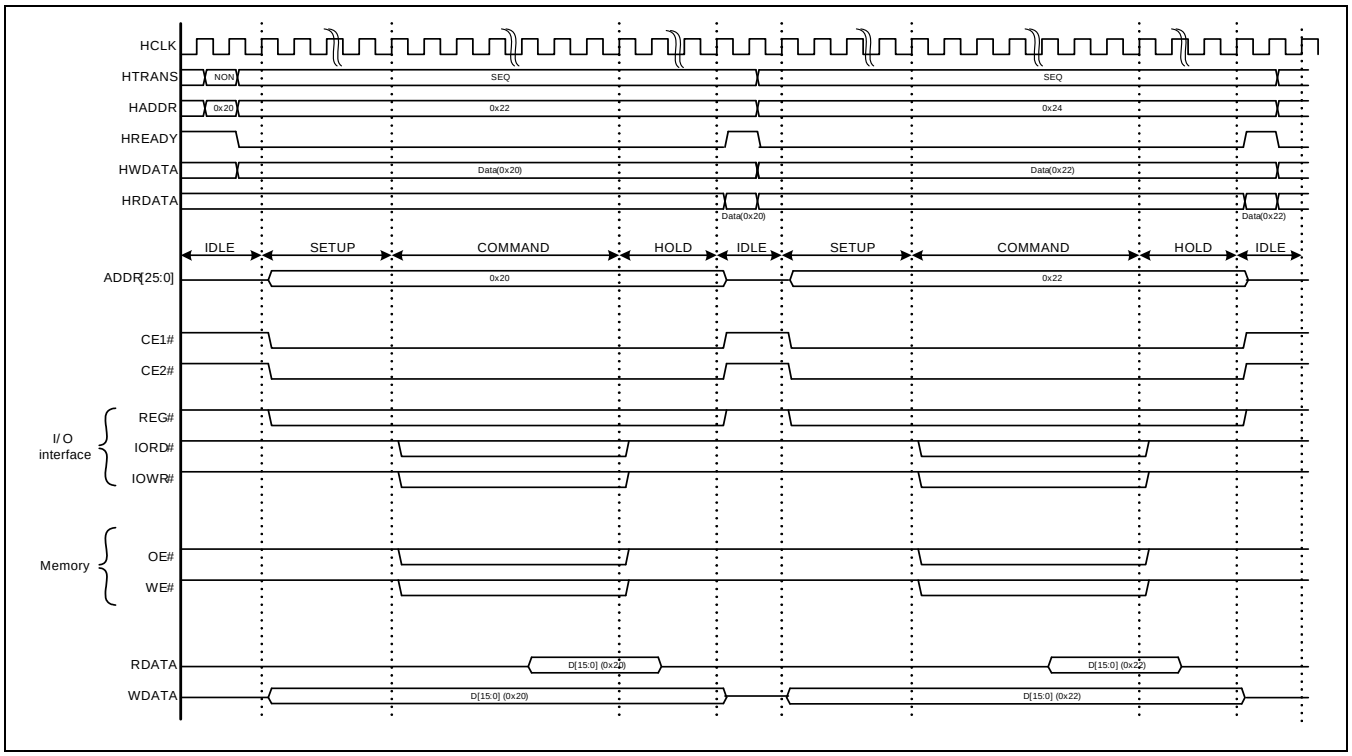


Figure 5.5-4 PC Card 16-bit Transfer (16-bit PC Card)

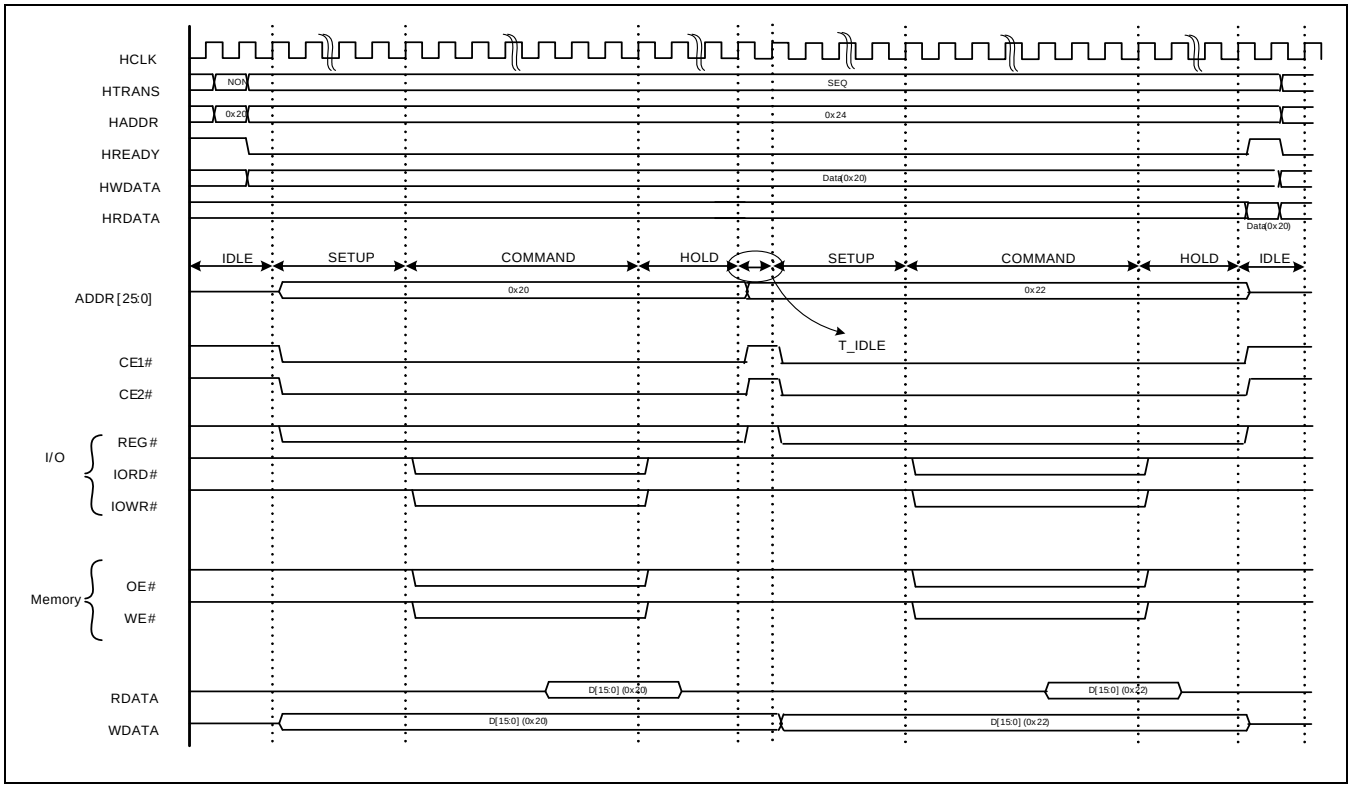


Figure 5.5-5 PC Card 32-bit Transfer (16-bit PC Card)

The PC card mode operation has four states namely: IDLE, SETUP, COMMAND, and HOLD. The timing for various states is set in special function registers: PCCARD_ATTR, PCCARD_I/O AND PCCARD_COMM. Each register contains timing for SETUP, COMMAND and HOLD state for the attribute, I/O or common memory.

Each Function Mode Control Signaling is described in Table 5.5-2 below.

Table 5.5-2 Addressing Mode & Valid Data

Addressing mode	nCE1	nCE2	A[0]	D[15:8]	D[7:0]
No access (standby mode)	1	1	X	High-Z	High-Z
8/16-bit mode (even byte)	0	1	0	High-Z	Even byte
8-bit mode (odd byte)	0	1	1	High-Z	Odd byte
16-bit mode (odd byte only)	1	0	1	Odd byte	High-Z
16-bit mode (even & odd byte)	0	0	0	Odd byte	Even byte

1.5 TRUE IDE MODE PIO/ PDMA TIMING DIAGRAM

The PIO transfer protocol supports 8-bit register access in driver and 16-bit PIO data access. Both hosts and devices support ATA_IORDY if PIO mode 3 or 4 is the currently selected mode of operation. The Figure 5.5-6 defines the relationships between host and device interface signals for data and registers transfer. The table 5.5-3 describes the timing parameters of PIO modes.

The host cannot have a 10Kohm pull-down resistor and pull-up resistor on DD[7] to allow a host to recognize the absence of a device at power-up so that a host shall detect BSY as being cleared when attempting to read the Status register of a device that is not present.

The Figure 5.5-6 shows the timing cycle of the true IDE PIO mode if ATA controller is in the ATA_TRANS state. The figure indicates various timing parameters. Timing 't1' indicates the time between address valid and IORD/IOWR asserted. Timing 't2' indicates the time for which IORD/ IOWR is asserted. The ATA state transfer in PDMA class follows similar timing.

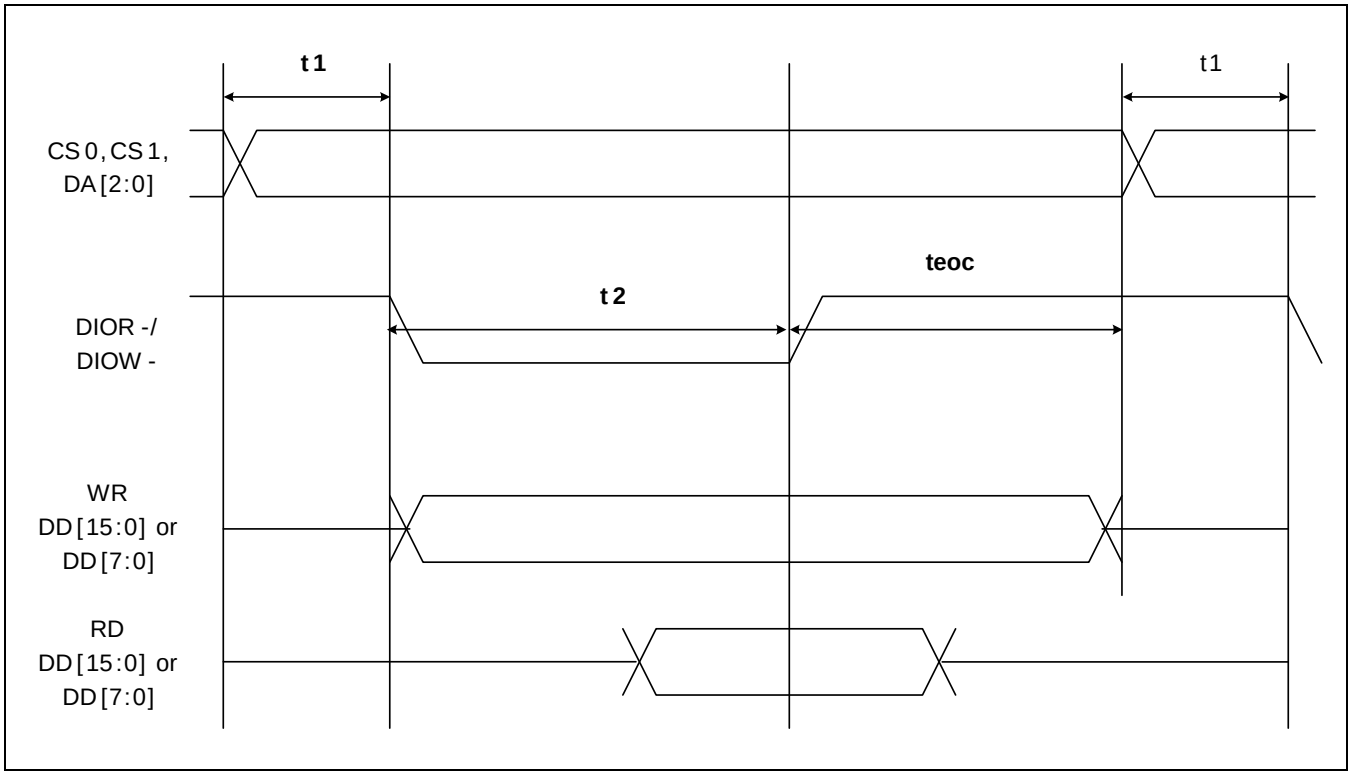


Figure 5.5-6 PIO Mode Waveform

Table 5.5-3 Timing Parameter Each PIO Mode

Register Transfer	MODE0	MODE1	MODE2	MODE3	MODE4
t1	(70, --)	(50, --)	(30, --)	(30, --)	(25, --)
t2	(290, --)	(290, --)	(290, --)	(80, --)	(70, --)
tEOC	(240, --)	(43, --)	(10, --)	(70, --)	(25, --)
t1+t2+tEOC	(600, --)	(383, --)	(330, --)	(180, --)	(120, --)
Data Transfer	**MODE0**	**MODE1**	**MODE2**	**MODE3**	**MODE4**
t1	(70, --)	(50, --)	(30, --)	(30, --)	(25, --)
t2	(165, --)	(125, --)	(100, --)	(80, --)	(70, --)
tEOC	(365, --)	(208, --)	(110, --)	(70, --)	(25, --)
t1+t2+tEOC	(600, --)	(383, --)	(240, --)	(180, --)	(120, --)

unit "ns"

1.5.1 ATA_PIO_TIME Register Setting Example (In case of Data Transfer)

The "t1" minimum time is 70ns in the system clock of 100MHz (10ns). It gives 7; "t1" divided by 10ns. This case has no residual, therefore pio_t1[3:0] assigns 6 which is 7 minus 1. If it has residual, assign the quotient at pio_t1[3:0].

ATA_PIO_TIME (Tpara) = PIO mode (Minimum, Maximum)/ system clock – 1

tPIO0 (Timing Parameter of PIO Mode 0 in case of Register Transfer) : 32'h000_17_1c_6

t1: 70/10 = 7 pio_t1 value = 7- 1 = 6 pio_t1[3:0] : 0x6

t2: 290/10 = 29 pio_t2 value = 29 - 1 = 28 pio_t2[11 :4] : 0x1c

teoc: 240/10 = 24 pio_teoc value = 24 - 1 = 23 pio_teoc[19:12] : 0x17

Steps for ATAPI PIO protocol:

- Initialization: Address (ATA_DA) and chip selection (ATA_CS0n/CS1n) setup if the read/ write strobe (ATA_DIORn/DIOWn) is deasserted.
- Data Read/ Write setup: Both Driver and Host shall do that the data on the data line for setup.
- Data Read/ Write hold: Both Driver and Host must remain stable after read/ write strobe signal is asserted (active low).

1.6 FLOWCHART FOR PIO READ / WRITE

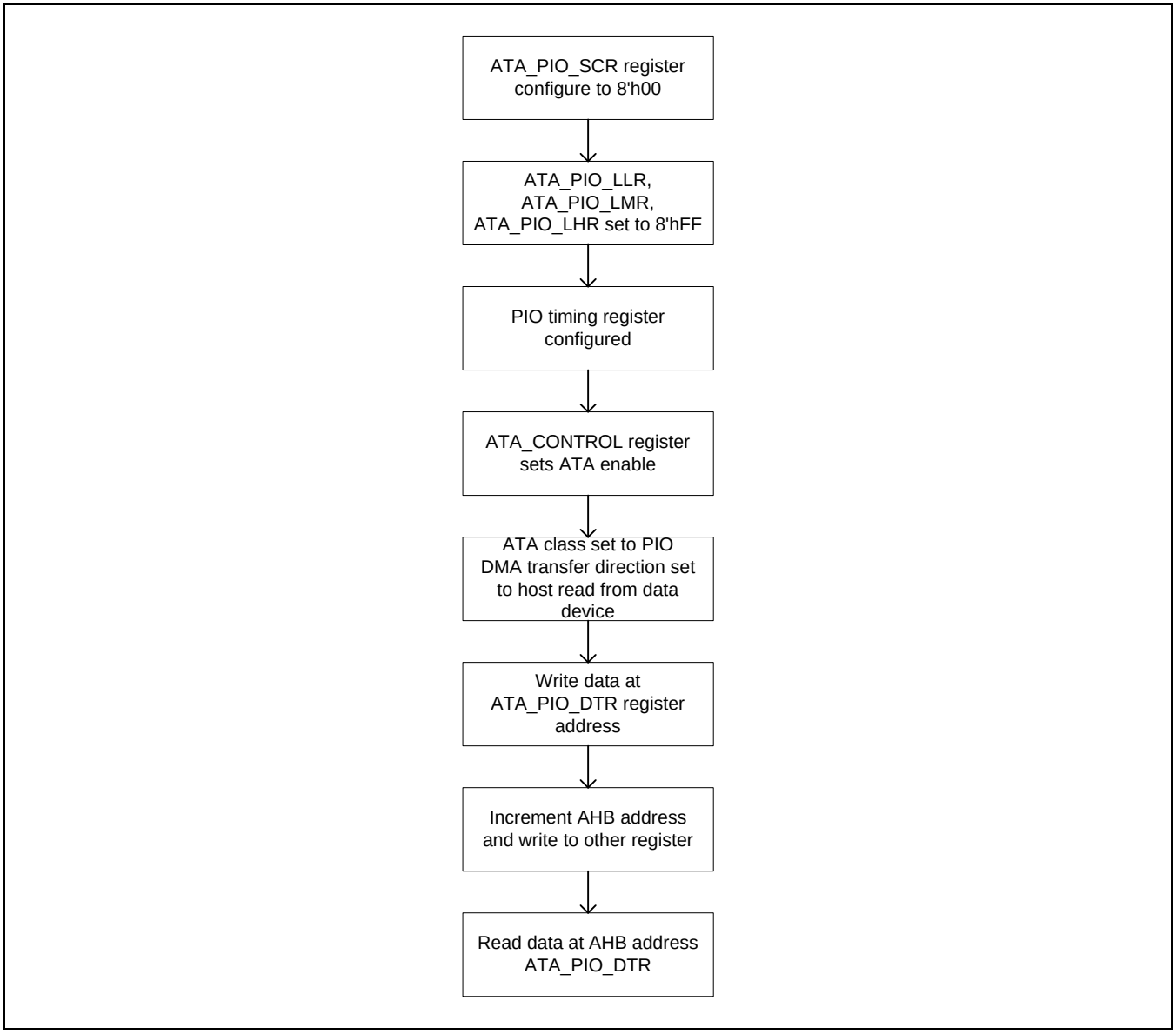


Figure 5.5-7 Flowchart for Read / Write in PIO Class

1.7 TRUE IDE MDMA MODE TIMING DIAGRAM

The ATAPI MDMA streams data continuously across the ATA interface between the host and the target device. This transfer class allows either the driver or host to pause or terminate the data flow. To support various transfer speed classes, the CPU programs appropriate timing parameters. The ATA_CS0n and CS1n are inactive during MDMA transfer. The ATA Host controller is always the master in the MDMA transfer classes. The MDMA has three transfer modes (Mode 0 ~ 2). The fastest mode is mode 2.

The Figure 5.5-8 defines the relationships between host and device interface signals for data transfer. The Table 5.5-4 describes the timing parameters of MDMA read and write transfer.

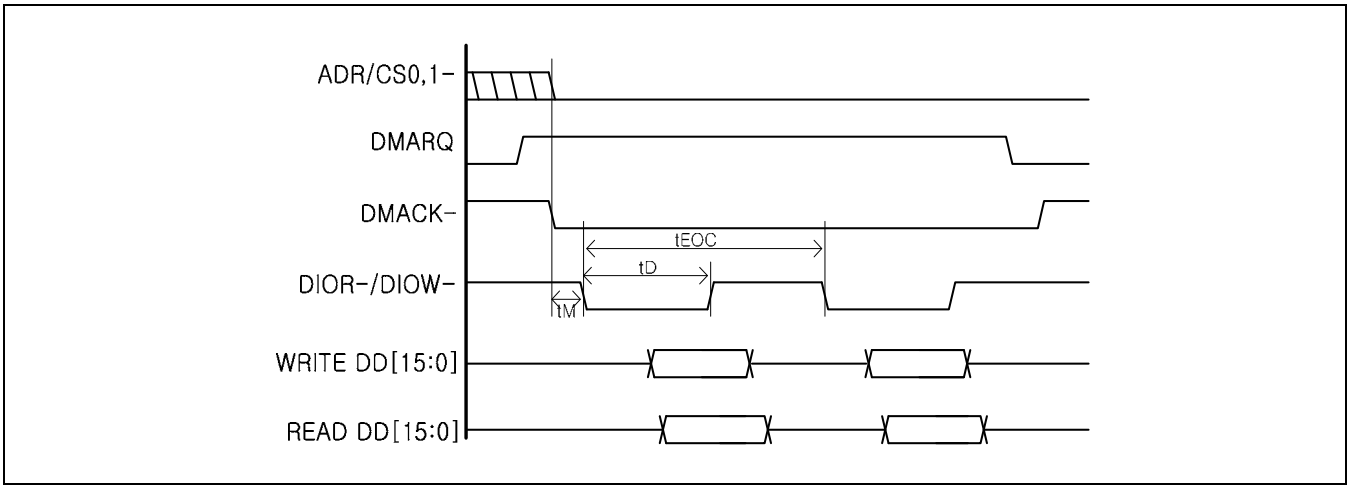


Figure 5.5-8 MDMA Timing Diagram

Table 5.5-4 MDMA Timing Parameters

	MODE0	MODE1	MODE2
tm	(50, --)	(30, --)	(25, --)
td	(215, --)	(80, --)	(70, --)
tEOC	(265, --)	(70, --)	(50, --)
td+tEOC	(480, --)	(150, --)	(120, --)

unit: ns

1.7.1 ATA_MDMA_TIME Register Setting Example

The "td" minimum time is 215ns in the system clock is 100MHz (10ns). It gives 21.5; "td" divided by 10ns. This case has residual, assigning quotient (21) to the dma_td[3:0]. If it has no residual, assign the quotient minus 1 at dma_td[3:0].

tMDMA0 (Timing Parameter of MDMA Mode 0)		: 32'h000_1a_15_4	
tm: 50/10 = 5	dma_td value = 5 - 1 = 4	dma_tm[3 :0]	: 0x4
td: 215/10 = 21.5	dma_td value = 21	dma_td[11:4]	: 0x15
teoc: 265/10 = 26.5	dma_teoc value = 26	dma_teoc[19:12]	: 0x1a

Steps for ATAPI MDMA transfer protocol (To write and read transfer):

Steps to Write Protocol:

1. Wait for the driver to activate ATA_DMARQ.
2. Activate ATA_DMACKn, deactivate ATA_CS0n/CS1n, and set time to 0.
3. Activate ATA_DIOWn at time tM.
4. Drive 16-bit data on the lines at time tD.
5. Deactivate ATA_DIOWn after tD.
6. If ATA_DMARQ is still active, repeat step 3 to 6 for another word, and deactivate ATA_DMACKn at time tM.

Steps to Read Protocol:

1. Wait for the driver to activate ATA_DMARQ.
2. Activate ATA_DMACKn, deactivate ATA_CS0n/CS1n, and set time to 0.
3. Activate ATA_DIORn at time tM.
4. Deactivate ATA_DIORn and latch 16-bit data lines at time tD.
5. If ATA_DMARQ is still active, repeat step 3 to 5 for another word, and deactivate ATA_DMACKn at time tM.

1.8 TRUE IDE UDMA MODE TIMING DIAGRAM

The Ultra-DMA (UDMA) is a fast DMA protocol which supports six timing modes (mode 0 ~ 5). Mode 5 is the fastest; it operates at 100MHz. This ATAPI host controller supports to mode 4. It runs 66MHz. Both host and device driver perform CRC check during UDMA burst transfer. At the end of the burst, the host sends its CRC result to the device. If the CRC result does not match, the driver reports an error in the error register and asserts the ATA_INTRQ signal.

The following figures (figure 7, figure 8, figure 9 and figure 10) defines the relationships between host and device interface signals for UDMA data transfer.

The timing parameters involved are tACKENV, tRP, tSS, tDVS, tDVH.

- tACKENV indicates the setup and hold times of DMACK (Before assertion or negation) and envelope time (From DMACKn to STOP and HDMARDYn).
- tRP indicates Ready-to-pause time.
- tSS indicates time from STROBE edge to negation of DMARQ or assertion of STOP.
- tDVS is time for which data is valid until STROBE edge.
- tDVH is time from STROBE edge until data may become invalid.

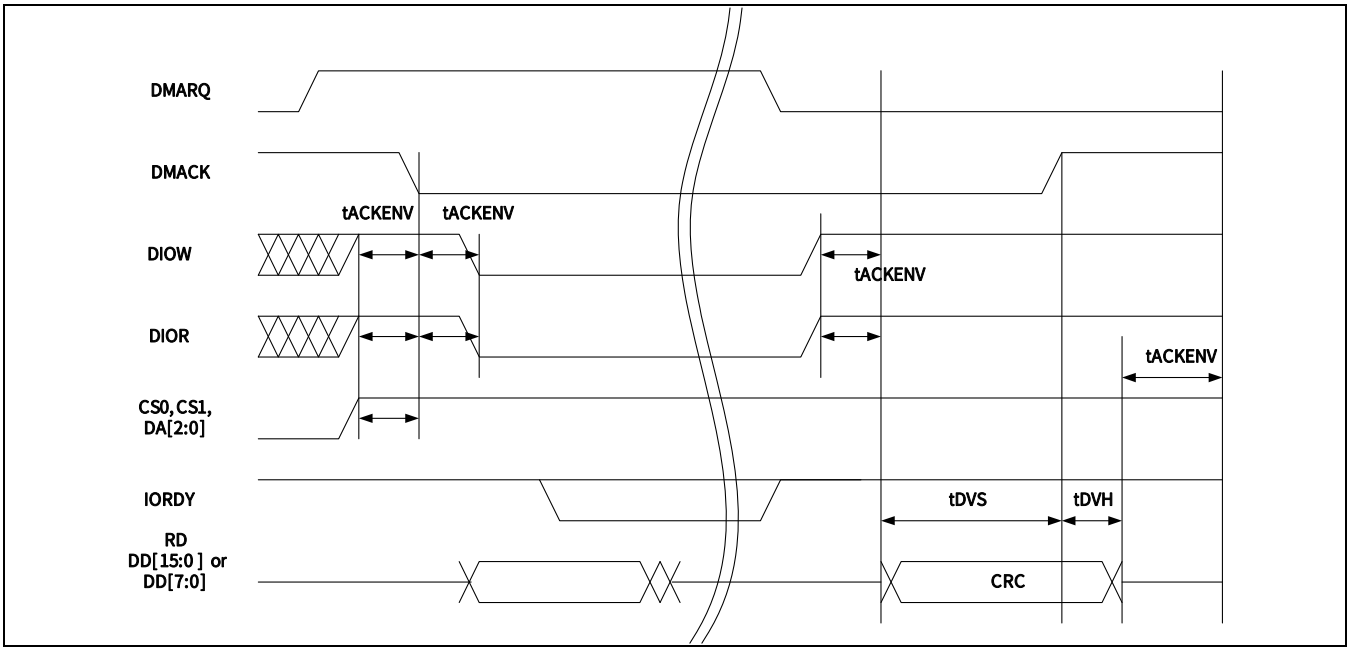


Figure 5.5-9 UDMA- In Operation (Terminated by Device)

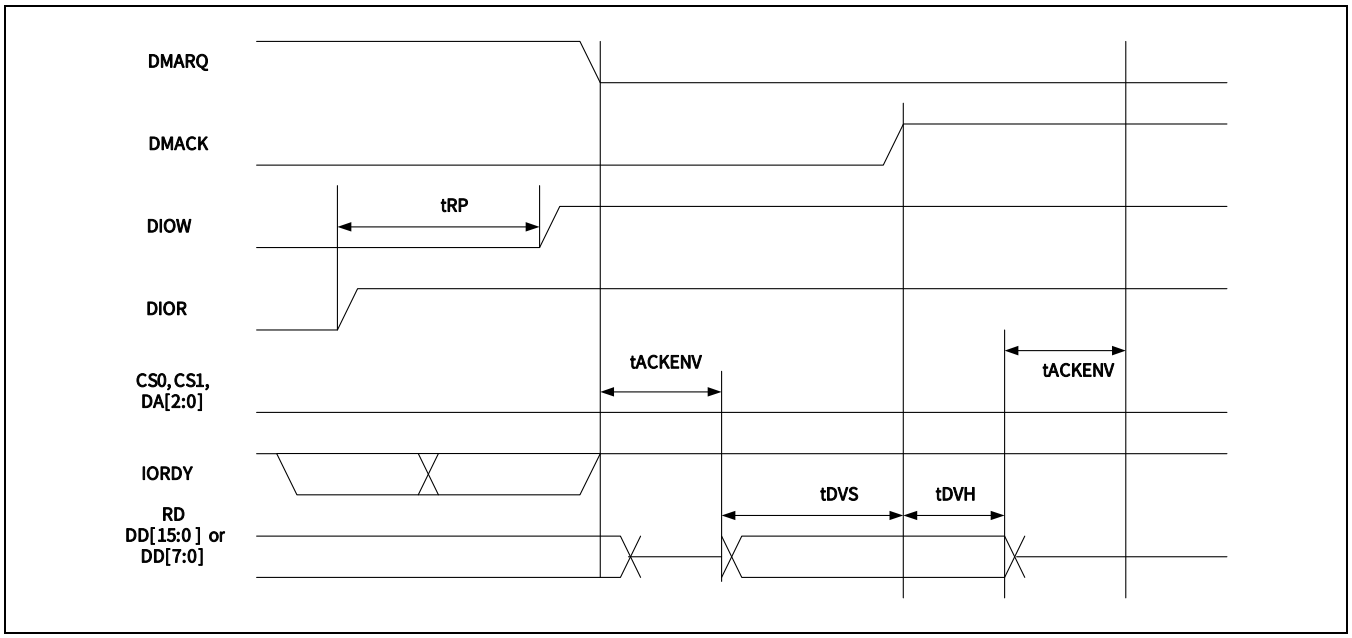


Figure 5.5-10 UDMA - In Operation (Terminated by Host)

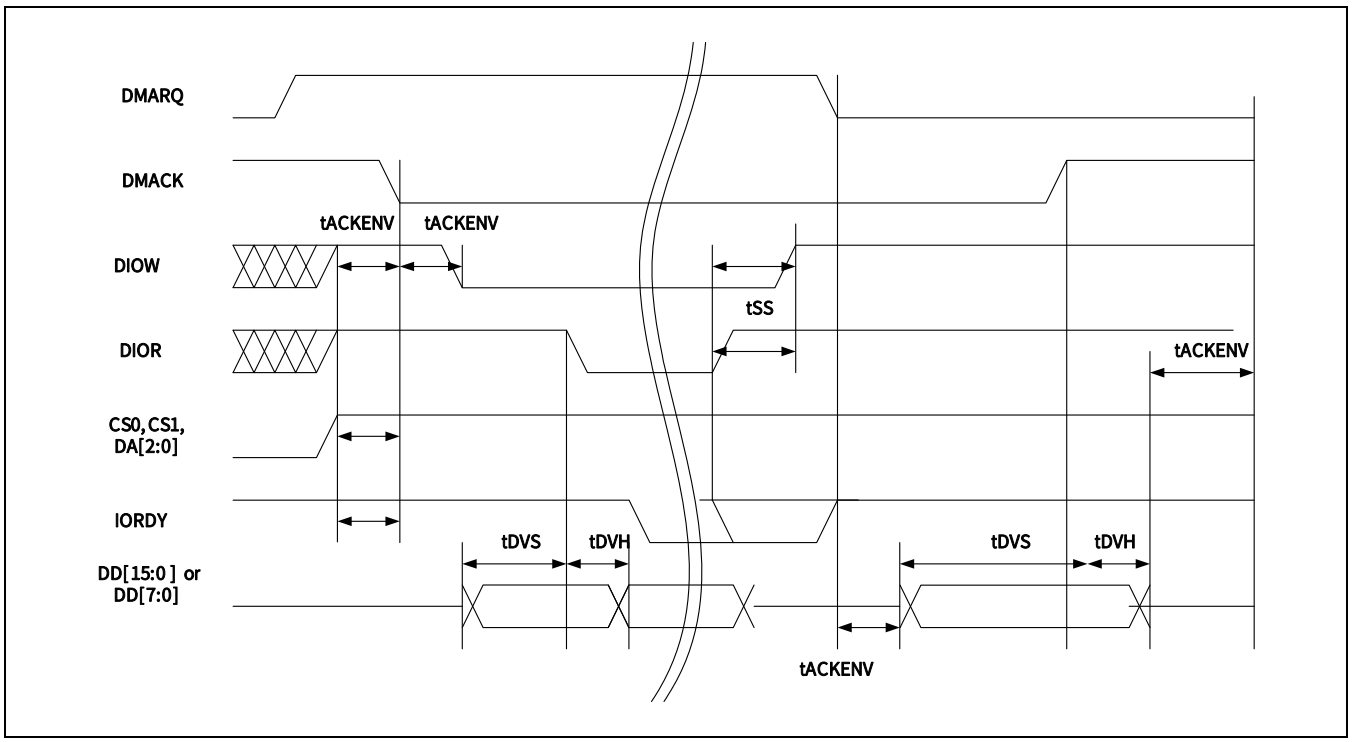


Figure 5.5-11 UDMA - Out Operation (Terminated by Device)

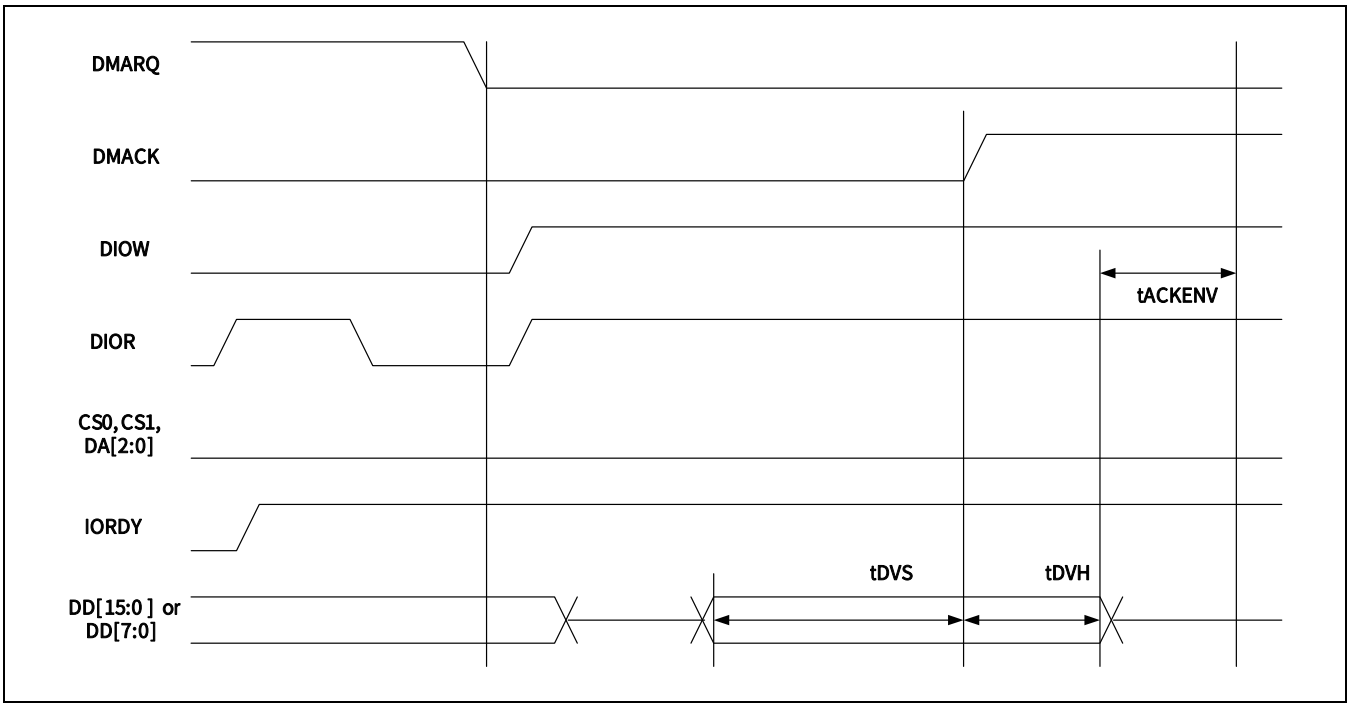


Figure 5.5-12 UDMA - Out Operation (Terminated by Host)

Table 5.5-5 Timing Parameter Each UDMA Mode

UDMA mode	UDMA 0	UDMA 1	UDMA 2	UDMA 3	UDMA 4
tACKENV	(20, 70)	(20, 70)	(20, 70)	(20, 55)	(20, 55)
tSS	(50, --)	(50, --)	(50, --)	(50, --)	(50, --)
tRP	(160, --)	(125, --)	(100, --)	(100, --)	(100, --)
tDVS	(70, --)	(48, --)	(31, --)	(20, --)	(6.7, --)
tDVH	(6.2, --) *(50)	(6.2, --) *(32)	(6.2, --) *(29)	(6.2, --) *(25)	(6.2, --) *(23.3)
tDVS+tDVH	(120, --)	(80, --)	(60, --)	(45, --)	(30, --)

unit: ns, *(50) is "(tDVS+tDVH)" – "tDVS" = 120 – 70 = 50

1.8.1 ATA_UDMA_TIME Register Setting Example

The "tackenv" minimum time is 20ns in the system clock of 100MHz (10ns). It gives 2; "tackenv" divided by 10ns. This case has no residual, therefore the udma_tackenv[3:0] assigns 1 which is 2 minus 1. If it has residual, assign the quotient at udma_tackenv[3:0].

ATA_UDMA_TIME (Tpara) = UDMA mode(min, max) / system clock – 1

tUDMA0(Timing Parameter of UDMA Mode 0) : 32'h04_06_0f_4_1

tackenv:	20/10 = 2	udma_tackenv value = 2 - 1 = 1	udma_tackenv[3 :0] : 0x1
tss:	50/10 = 5	udma_tss value = 5 - 1 = 4	udma_tss[7:4] : 0x4
trp:	160/10 = 16	udma_trp value = 16 -1 = 15	udma_trp[15:8] : 0x0f
tdvs:	70/10 = 7	udma_tdvs value = 7 - 1 =6	udma_tdvs[23:16] : 0x06
tdvh:	50/10 = 5	udma_tdvh value = 5 - 1 = 4	udma_tdvh[27:24]: 0x4

(tdvh minimum timing is 6.2ns, but the timing parameter sets 50ns since the tDVS and tDVH summation is 120ns)

The Table 5.5-6 shows True-IDE Mode Control Signaling:

Table 5.5-6 True-IDE Mode I/O Decoding

nCE2	nCE1	A2	A1	A0	nDMACK	nIORD=0	nIOWR=0	Note
1	0	0	0	0	1	PIO RD data	PIO WR data	8 or 16 bit
1	1	X	X	X	0	DMA RD data	DMA WR data	16bit
1	0	0	0	1	1	Error Register	Feature	8 bit
1	0	0	1	0	1	Sector Count	Sector Count	8 bit
1	0	0	1	1	1	Sector No.	Sector No.	8 bit
1	0	1	0	0	1	Cylinder Low	Cylinder Low	8 bit
1	0	1	0	1	1	Cylinder High	Cylinder High	8 bit
1	0	1	1	0	1	Select Card/Head	Select Card/Head	8 bit
1	0	1	1	1	1	Status	Command	8 bit
0	1	1	1	0	1	Alt Status	Device Control	8 bit

1.9 TRANSFER STATE ABORT

The PIO, PDMA, MDMA or UDMA checks for abort or stop transfer state after completing one full cycle of the finite state machine. The FSM transition from IDLE state happens if ATA transfer state is in ATA_TRANS. The FSM continues the cycle while the abort is asserted. The transfer in any class stays in IDLE after detecting ATA state in ATA_ABORT.

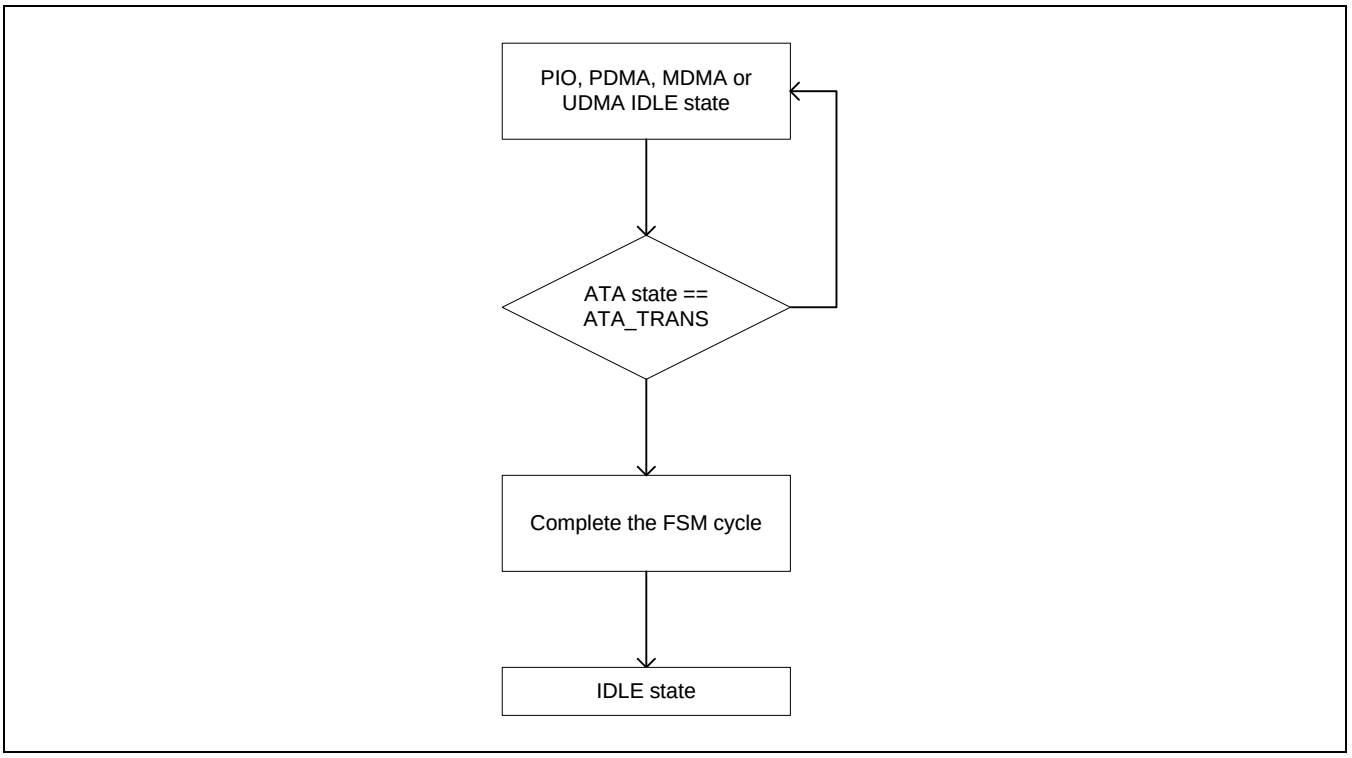


Figure 5.5-13 Flowchart for Abort in ATA Mode

1.10 EBI BACKOFF SUPPORTED PROTOCOL

In the UDMA/ MDMA Class (True IDE mode), Device I/F can used with EBI (External BUS Interface) MUX;

In this case, if the EBIBACKOFF signal is issued (by the higher priority controller), CFCON terminates the current transfer (unit is sector) as soon as possible.

1.10.1 Level 1

1. EBI issues EBI-BACKOFF signal, during operation with UDMA/MDMA Class.
2. Issues interrupt request to CPU (ebi_bf_rd_int or ebi_bf_wr_int)
3. CPU asserts ABORT command to ATA controller (Write SFR xfr_command in ATA_COMMAND).
4. ATA Controller releases EBI BUS and interrupt bit is cleared (Other device is used to EBI BUS).
5. Make software reset by ATA_SWRST to clear the leftover values of internal registers.
6. Re-sets transfer configuration and re-start remaining data transmission continuously.
   - Set 'ebi_bf_en' SFR in ATA_CFG_2.
   - Unmasking Interrupt bit (mask_ebi_bf_rd_int or mask_ebi_bf_wr_int).

1.10.2 Level 2

1. EBI issues EBI-BACKOFF signal, during write operation with UDMA/MDMA Class.
2. Complete operation for current transfer sector.
3. Aborts ATA I/F and releases EBI BUS (Other device is used to EBI BUS).
4. Issues interrupt request to CPU (ebi_abort_rd_int or ebi_abort_wr_int)
5. Check status register until transfer has done (ABORT command is applied).
   - Read & Wait until {ATA_STATUS[1:0] == 2'b00}.
6. Make software reset by ATA_SWRST to clear the leftover values of internal registers.
7. Re-sets transfer configuration and re-start remaining data transmission continuously.
   - Set 'ebi_bf_en' and 'ebi_abort_rd'(or 'ebi_abort_wr') SFR in ATA_CFG_2.
   - Unmasking Interrupt bit (mask_ebi_abort_rd_int or mask_ebi_abort_wr_int).

1.11 PROGRAMMER'S MODEL

For top-level control and configuration of the CFC, use following register:

1. MUX_REG: Set internal mode and card power enable

For configuration and status of PC Card mode, use following registers:

1. PCCARD_CONFIG&STATUS: Set configuration and read the status of the register
2. PCCARD_INTMASK&SRC: Interrupt source and interrupt mask register
3. PCCARD_ATTR: Set the card access timing
4. PCCARD_I/O: Set the card access timing
5. PCCARD_COMM: Set the card access timing

For configuration and status read of ATA mode, use following registers:

1. ATA_CONTROL: ATA enable and clock down status
2. ATA_STATUS: ATA status
3. ATA_COMMAND: ATA command
4. ATA_SWRST: ATA software reset
5. ATA_IRQ: ATA interrupt sources
6. ATA_IRQ_MASK: ATA interrupt mask
7. ATA_CFG: ATA configuration for ATA interface
8. ATA_CFG2: ATA configuration 2 for ATA interface
9. ATA_MDMA_TIME : ATA multi-word DMAtiming
10. ATA_PIO_TIME: ATA PIO time
11. ATA_UDMA_TIME: ATA UDMA timing
12. ATA_XFR_NUM: ATA transfer number
13. ATA_XFR_CNT: ATA current transfer count
14. ATA_TBUF_BASE: ATA start address of track buffer
15. ATA_TBUF_SIZE: ATA size of track buffer
16. ATA_SBUF_BASE: ATA start address of source buffer
17. ATA_SBUF_SIZE: ATA size of source buffer
18. ATA_CADR_TBUF: ATA current address of track buffer
19. ATA_CADR_SBUF: ATA current address of source buffer
20. ATA_PIO_DTR: ATA PIO device data register
21. ATA_PIO_FED: ATA PIO device feature/ error register
22. ATA_PIO_SCR: ATA PIO sector count register

23. ATA_PIO_LLR: ATA PIO device LBA low register
24. ATA_PIO_LMR: ATA PIO device LBA middle register
25. ATA_PIO_LHR: ATA PIO device LBA high register
26. ATA_PIO_DVR: ATA PIO device register
27. ATA_PIO_CSD: ATA PIO device command/ status register
28. ATA_PIO_DAD: ATA PIO device control/alternate status register
29. ATA_PIO_READY: ATA PIO data read/write ready
30. ATA_PIO_RDATA: ATA PIO read data from device data register
31. BUS_FIFO_STATUS: ATA internal AHB FIFO status
32. ATA_FIFO_STATUS: ATA internal ATA FIFO status

1.12 MEMORY MAP

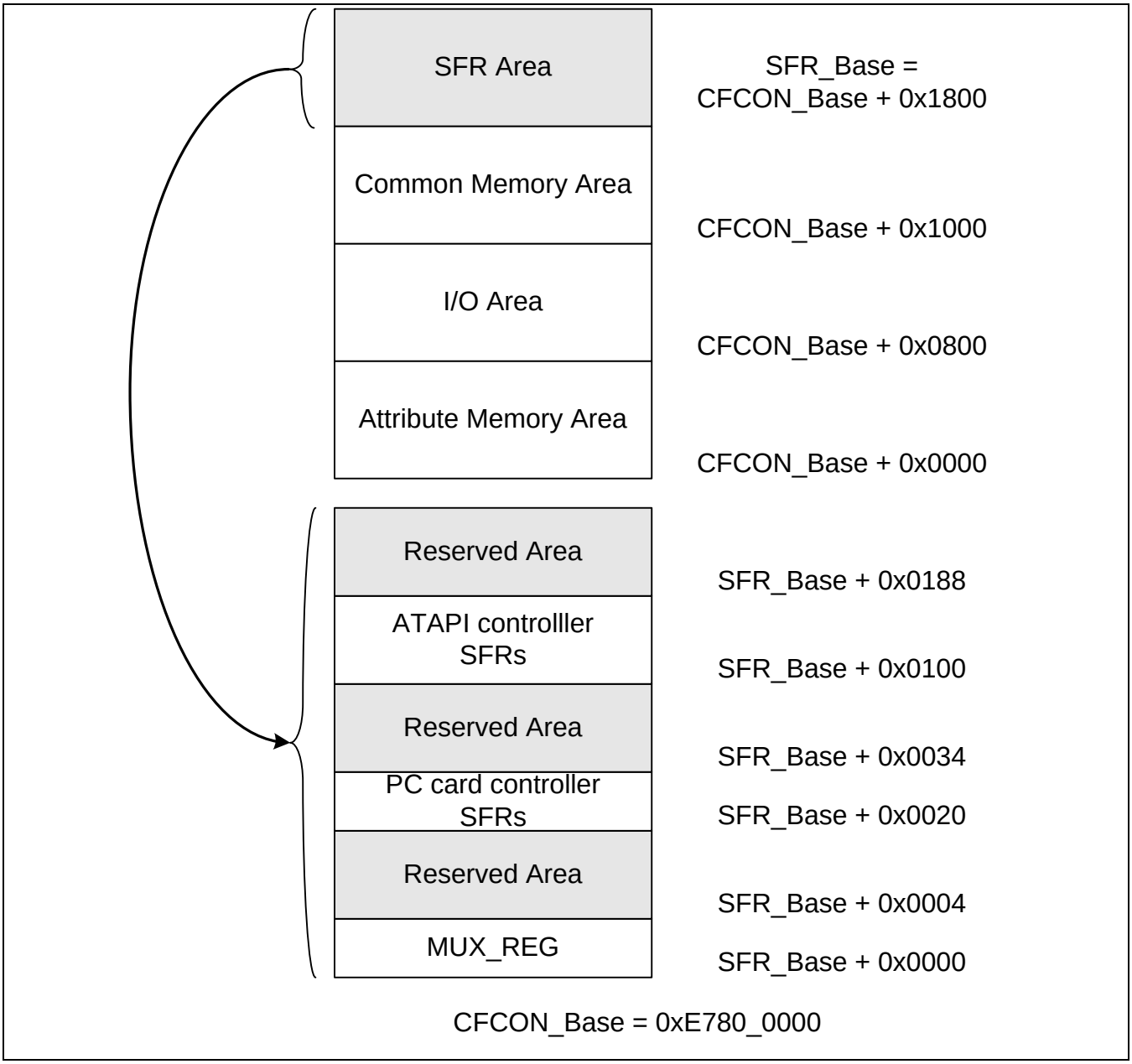


Figure 5.5-14 Memory Map Diagram

2 PROGRAMMING GUIDE

2.1 BASIC DRAWING FUNCTION

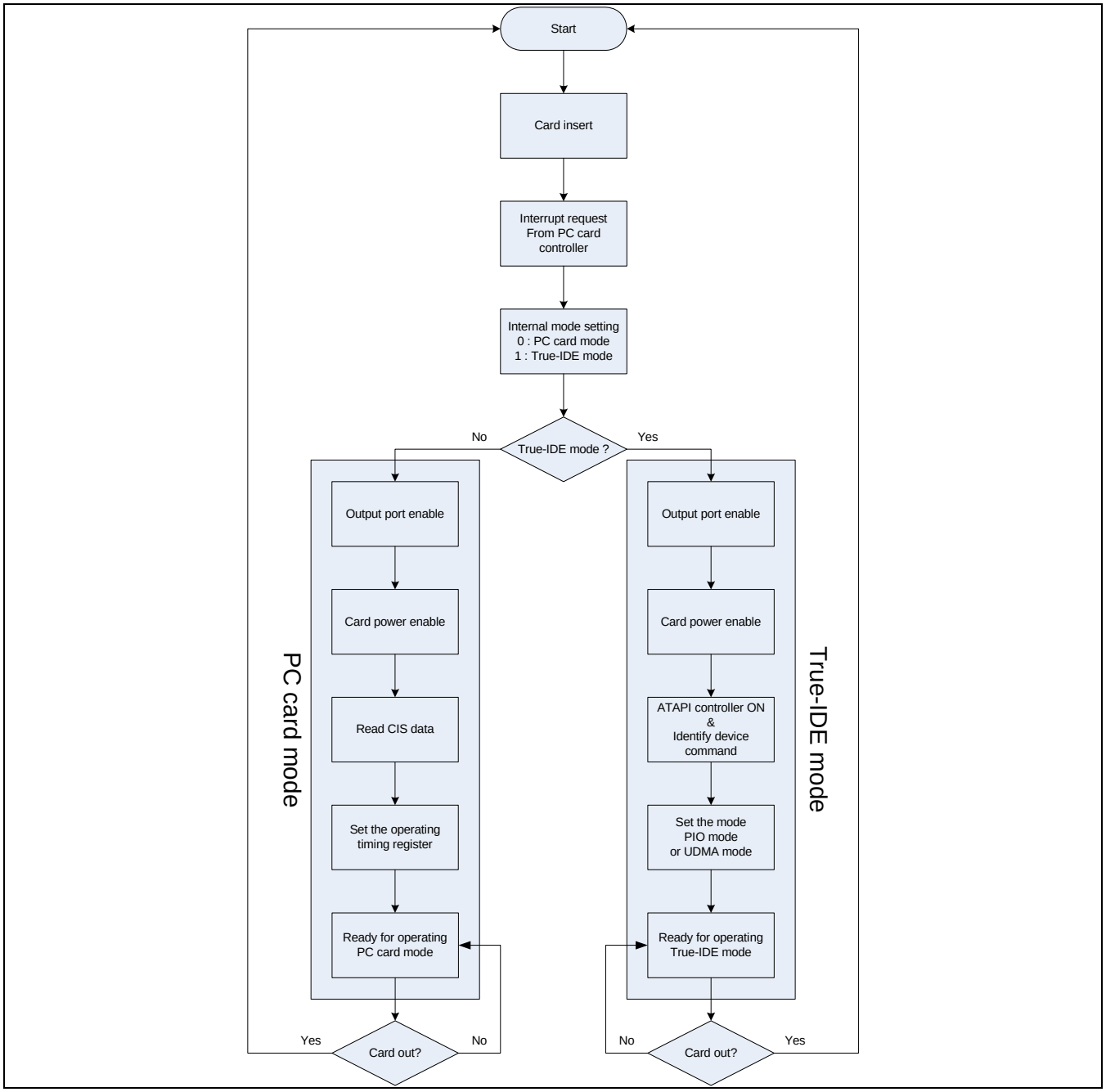


Figure 5.5-15 Basic Function Flow Chart

2.2 MISC

2.2.1 nCSEL

The nCSEL is card selection pin. This signal internally pulled-up in CF/CF+ card. This configures the device as a Master or a Slave if configured in the True-IDE mode. If this pin is grounded, this device is configured as a Master. If this pin is open, this device is configured as a Slave. This signal must be tied GND or VCC, not pulled-up/ pulled-down. If you pulled down this signal on board, nCSEL voltage level is not stable because this signal internally pulled-up in CF/CF+ card.

2.3 EXTERNAL PULL-UP/ PULL-DOWN

Table 5.5-7 Externally Pull-Up/Pull-Down Signals

Pull-up to VCC	Pull-down to GND
nIREQ (R ≥ 10KΩ)	nINPACK (R ≥ 5.6KΩ)
nWAIT (R ≥ 10KΩ)	
nIOIS16 (R ≥ 10KΩ)	
nCD (R ≥ 10KΩ)	

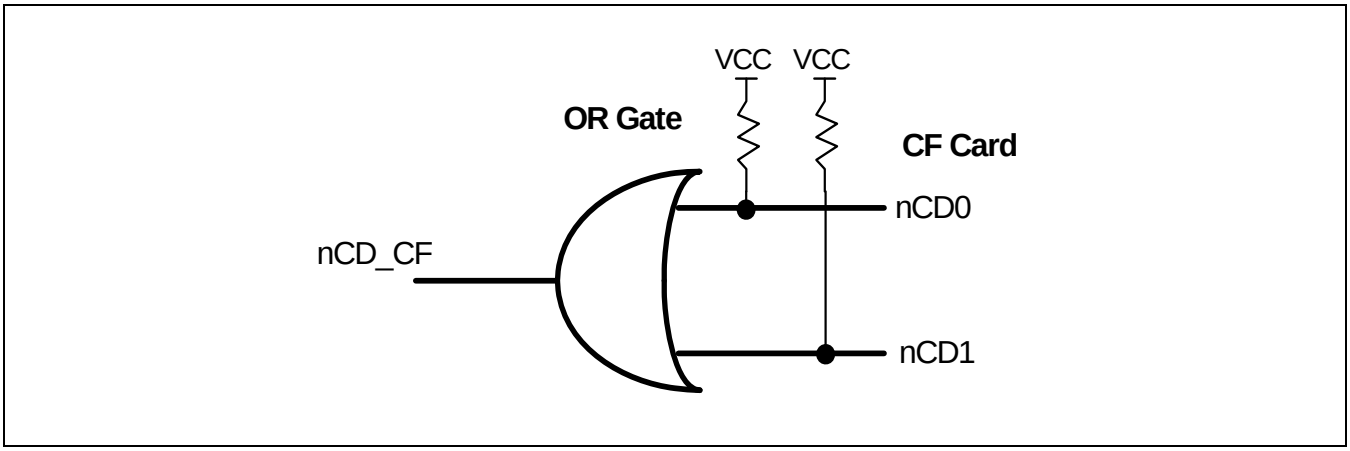


Figure 5.5-16 External (External of SOC) Glue Logic for Card Detection Pin

3 EXAMPLE CODES

This explains the basic steps followed in the programming of the PCCARD MODE and ATA MODE.

3.1 INITIALIZATION ROUTINE

PCCARD MODE:-

```
void activate_cfc_controller (void) {

      apbif_single_write   (CFC_PROGRAMMING, MUX_REG, 0xaaaa_aaa6); // Initialize the internal mode
Register
      apbif_single_write   (CFC_PROGRAMMING, PCCARD_CFNG_STATUS, 0xaaaa_00a0); // Initialize the
status Register
    apbif_single_write   (CFC_PROGRAMMING, PCCARD_INTMSK_SRC, 0xaaaa_a000);// Intialize the pccard
interrupt mask and source register

     apbif_single_write   (CFC_PROGRAMMING,MUX_REG ,  0xaaaa_aaa0 ); // configuring the internal mode
Register
     apbif_single_write   (CFC_PROGRAMMING, PCCARD_CFNG_STATUS, 0xaaaa_00a0); // configuring the
status register
      apbif_single_write   (CFC_PROGRAMMING, PCCARD_INTMSK_SRC, 0xaaaa_af00); // configuring the
pccard interrupt mask and source register

       };
```

ATA MODE:-

```
      apbif_single_write   (CFC_PROGRAMMING, MUX_REG, 0xaaaa_aaa7); // Initialize the internal mode
Register
      apbif_single_write   (CFC_PROGRAMMING, ATA_CFG , 0xaaaa_0000 );// Initializing the ata configuration
Register
     apbif_single_write   (CFC_PROGRAMMING, ATA_IRQ ,  0xaaaa_aa00 ); // initializing interrupt register
     apbif_single_write   (CFC_PROGRAMMING, ATA_IRQ_MASK, 0xaaaa_aa00); // initializing interrupt mask
register
      apbif_single_write   (CFC_PROGRAMMING, MUX_REG, 0xaaaa_aaa1); // configuring the internal mode
register
     apbif_single_write   (CFC_PROGRAMMING, ATA_CFG ,  0xaaaa_0000 );// configuring the  ata configuration
register
     apbif_single_write   (CFC_PROGRAMMING, ATA_IRQ, 0xaaaa_aa00); // configuring interrupt register
     apbif_single_write   (CFC_PROGRAMMING, ATA_IRQ_MASK, 0xaaaa_aaff); // configuring interrupt mask
register


       };
```

3.2 INTERRUPT SERVICE ROUTINE

PCCARD MODE:-

```
Void interrupt_service_routine ( void ) {
    //Poll for the INTSRC_ERR_N bit to be set to 0 in the PCCARD_INTMSK_SRC register
    wait_for_cfc_intstat();
    //Read the INTSRC_ERR_N bit of the PCCARD_INTMSK_SRC  register
    apbif_single_read (SEND_FROM_CFC, PCCARD_INTMSK_SRC);
    //Write '1' to the INTSRC_ERR_N bit of the PCCARD_INTMSK_SRC register (this clears the interrupt)
    apbif_single_write (SEND_TO_CFC, PCCARD_INTMSK_SRC, 0Xaaaa_aff4);

   };
Void interrupt_service_routine ( void ) {
    //Poll for the INTSRC_IREQ bit to be set to 0 in the PCCARD_INTMSK_SRC register
    wait_for_cfc_intstat();
    //Read the INTSRC_IREQ bit of the PCCARD_INTMSK_SRC register
    apbif_single_read (SEND_FROM_CFC, PCCARD_INTMSK_SRC);
    //Write '1' to the INTSRC_IREQ bit of the PCCARD_INTMSK_SRC register (this clears the interrupt)
    apbif_single_write (SEND_TO_CFC, PCCARD_INTMSK_SRC, 0Xaaaa_aff2);

   };
Void interrupt_service_routine ( void ) {
    //Poll for the INTSRC_CD bit to be set to 0 in the PCCARD_INTMSK_SRC register
    wait_for_cfc_intstat();
    //Read the INTSRC_CD bit of the PCCARD_INTMSK_SRC register
    apbif_single_read (SEND_FROM_CFC, PCCARD_INTMSK_SRC);
    //Write '1' to the INTSRC_CD bit of the PCCARD_INTMSK_SRC register (this clears the interrupt)
    apbif_single_write (SEND_TO_CFC, PCCARD_INTMSK_SRC, 0Xaaaa_aff1);

   };
```

ATA MODE:-

```
Void interrupt_service_routine ( void ) {
    //Poll for the sbuf_empty_int  bit to be set to 0 in the ATA_IRQ register
    wait_for_cfc_intstat();
    //Read the sbuf_empty_int bit of the ATA_IRQ register
    apbif_single_read (SEND_FROM_CFC, ATA_IRQ);
    //Write '1' to the sbuf_empty_int  bit of the ATA_IRQ register (this clears the interrupt)
    apbif_single_write (SEND_TO_CFC, ATA_IRQ, 0Xaaaa_afe0);

   };
Void interrupt_service_routine ( void ) {
    //Poll for the tbuf_full_int  bit to be set to 0 in the ATA_IRQ register
    wait_for_cfc_intstat();
    //Read the tbuf_full_int bit of the ATA_IRQ register
    apbif_single_read (SEND_FROM_CFC, ATA_IRQ);
    //Write '1' to the tbuf_full_int  bit of the ATA_IRQ register (this clears the interrupt)
```

```
 apbif_single_write (SEND_TO_CFC, ATA_IRQ, 0Xaaaa_aff8);

};
Void interrupt_service_routine ( void ) {
 //Poll for the atadev_irq_int  bit to be set to 0 in the ATA_IRQ register
 wait_for_cfc_intstat();
 //Read the atadev_irq_int bit of the ATA_IRQ register
 apbif_single_read (SEND_FROM_CFC, ATA_IRQ);
 //Write '1' to the atadev_irq_int  bit of the ATA_IRQ register (this clears the interrupt)
 apbif_single_write (SEND_TO_CFC, ATA_IRQ, 0Xaaaa_aff4);

};
Void interrupt_service_routine ( void ) {
 //Poll for the udma_hold_int  bit to be set to 0 in the ATA_IRQ register
 wait_for_cfc_intstat();
 //Read the udma_hold_int bit of the ATA_IRQ register
 apbif_single_read (SEND_FROM_CFC, ATA_IRQ);
 //Write '1' to the udma_hold_int  bit of the ATA_IRQ register (this clears the interrupt)
 apbif_single_write (SEND_TO_CFC, ATA_IRQ, 0Xaaaa_aff2);

};


Void interrupt_service_routine ( void ) {
 //Poll for the xfr_done_int  bit to be set to 0 in the ATA_IRQ register
 wait_for_cfc_intstat();
 //Read the xfr_done_int bit of the ATA_IRQ register
 apbif_single_read (SEND_FROM_CFC, ATA_IRQ);
 //Write '1' to the xfr_done_int  bit of the ATA_IRQ register (this clears the interrupt)
 apbif_single_write (SEND_TO_CFC, ATA_IRQ, 0Xaaaa_aff1);

};
```

3.3 USEFUL ROUTINES

4 I/O DESCRIPTION

Funtion Signal	I/O	Description	Pad		Type
			EBI muxed	Indep. ATA	
CF_nCS[0]	O	Card Enable Strobe, muxed with CF_CSn_0	Xm0CSn[4]	XmsmCSn	muxed
CF_nCS[1]	O	Card Enable Strobe, muxed with CF_CSn_1	Xm0CSn[5]	XmsmWEn	muxed
CF_A [10:3]	O	CF Card Address	Xm0ADDR[10:3]		muxed
CF_A[2:0]	O	CF Card Address, ATA IF DA	Xm0ADDR[2:0]	XmsmADDR[2:0]	muxed
CF_D[15:0]	I/O	Read&Write Data Bus	Xm0DATA[15:0]	XmsmDATA[15:0]	muxed
CF_IORDY	I	Wait signal from CF Card, muxed with IORDY at ATA mode	Xm0IORDY	XmsmADDR[3]	muxed
CF_INTRQ	I	Interrupt request from CF Card, muxed with INTRQ at ATA mode	Xm0INTRQ	XmsmADDR[4]	muxed
CF_INPACKn	I	Input Acknowledge in I/O mode, muxed with DMARQ at ATA mode	Xm0INPACKn	XmsmADDR[5]	muxed
CF_RESET	O	CF Card Reset, muxed with ATA_RST at ATA mode.	Xm0RESET	XmsmADDR[6]	muxed
CF_REG	O	Register in CF Card Strobe, muxed with DMACKn at ATA mode.	Xm0REG	XmsmADDR[7]	muxed
CF_IORDn	O	IO Read Enable Strobe	Xm0IORDn	XmsmRn	muxed
CF_IOWRn	O	IO Write Enable Strobe	Xm0IOWRn	XmsmIRQn	muxed
CF_CDn	I	Card Detect Signals	Xm0CDn		muxed
CF_OEn	O	Output Enable Strobe	Xm0CFOEn		muxed
CF_WEn	O	Write Enable strobe	Xm0CFWEn		muxed

4.1 ATA 40 PIN CONNECTOR

Num	Name	Description	Function Signal
1	/RESET	Reset	CF_RESET
2	GND	Ground	-
3	DD7	Data 7	CF_D[7]
4	DD8	Data 8	CF_D[8]
5	DD6	Data 6	CF_D[6]
6	DD9	Data 9	CF_D[9]
7	DD5	Data 5	CF_D[5]
8	DD10	Data 10	CF_D[10]
9	DD4	Data 4	CF_D[4]
10	DD11	Data 11	CF_D[11]
11	DD3	Data 3	CF_D[3]
12	DD12	Data 12	CF_D[12]
13	DD2	Data 2	CF_D[2]
14	DD13	Data 13	CF_D[13]
15	DD1	Data 1	CF_D[1]
16	DD14	Data 14	CF_D[14]
17	DD0	Data 0	CF_D[0]
18	DD15	Data 15	CF_D[15]
19	GND	Ground	-
20	(keypin)	Key	-
21	DMARQ	DMA Request	CF_INPACKn
22	GND	Ground	-
23	DIOW	Write Strobe	CF_IOWRn
24	GND	Ground	-
25	DIOR	Read Strobe	CF_IORDn
26	GND	Ground	-
27	IORDY	IO Ready	CF_IORDY
28	CSEL	Cable Select (Low:master, Hi-Z:slave)	-
29	DMACK	DMA Acknowledge	CF_REG
30	GND	Ground	-
31	INTRQ	Interrupt Request	CF_INTRQ
32	N.C.	Not Connected	-
33	DA1	Address 1	CF_A[1]
34	PDIAG	Hi-Z:master only	-
35	DA0	Address 0	CF_A[0]
36	DA2	Address 2	CF_A[2]
37	CS0	Chip Select 0	CF_nCS[0]
38	CS1	Chip Select 1	CF_nCS[1]
39	DASP	Device Active(Hi-Z:master only)	-
40	GND	Ground	-

4.2 CF 50 PIN CONNECTOR

Num	Name	Description	Function Signal
1	GND	Ground	-
2	D03	Data 3	CF_D[3]
3	D04	Data 4	CF_D[4]
4	D05	Data 5	CF_D[5]
5	D06	Data 6	CF_D[6]
6	D07	Data 7	CF_D[7]
7	-CE1	Card Enable 1	CF_nCS[0]
8	A10	Address 10	CF_A[10]
9	-OE	Output Enable	CF_OEn
10	A09	Address 9	CF_A[9]
11	A08	Address 8	CF_A[8]
12	A07	Address 7	CF_A[7]
13	Vcc	Power	-
14	A06	Address 6	CF_A[6]
15	A05	Address 5	CF_A[5]
16	A04	Address 4	CF_A[4]
17	A03	Address 3	CF_A[3]
18	A02	Address 2	CF_A[2]
19	A01	Address 1	CF_A[1]
20	A00	Address 0	CF_A[0]
21	D00	Data 0	CF_D[0]
22	D01	Data 1	CF_D[1]
23	D02	Data 2	CF_D[2]
24	WP	Write Protect	-
25	CD2	Card Detect 2	CF_CDn
26	CD1	Card Detect 1	CF_CDn
27	D11	Data 11	CF_D[11]
28	D12	Data 12	CF_D[12]
29	D13	Data 13	CF_D[13]
30	D14	Data 14	CF_D[14]
31	D15	Data 15	CF_D[15]
32	-CE2	Card Enable 2	CF_nCS[1]
33	-VS1	Ground	-
34	-IORD	IO Read Strobe	CF_IORDn
35	-IOWR	IO Write Strobe	CF_IOWRn
36	-WE	Write Enable	CF_WEn
37	Ready	Ready	CF_INTRQ
38	Vcc	Power	-
39	-CSEL	Card Select	-
40	-VS2	Open	-
41	RESET	Reset	CF_RESET
42	-WAIT	IO Ready	CF_IORDY
43	-INPACK	Input Acknowledgement	CF_INPACKn
44	-REG	DMACK	CF_REG
45	BVD2	DASP	-
46	BVD1	PDIAG	-
47	DO8	Data 8	CF_D[8]

48	DO9	Data 9	CF_D[9]
49	D10	Data 10	CF_D[10]
50	GND	Ground	-

5 REGISTER DESCRIPTION

5.1 REGISTER OVERVIEW

Register	Address	R/W	Description	Reset Value
CF card host controller base address				
MUX_REG	0xE780_1800	R/W	Top control and configuration	0x00000002
Reserved	~ 0xE780_181C	-	Reserved area	
PC card controller base address				
PCCARD_ CNFG&STATUS	0xE780_1820	R/W	PC card configuration & status register	0x00000F0X
PCCARD_ INTMSK&SRC	0xE780_1824	R/W	PC card interrupt mask & source register	0x0000070X
PCCARD_ATTR	0xE780_1828	R/W	PC card attribute memory area operation timing configuration register	0x00031909
PCCARD_I/O	0xE780_182C	R/W	PC card I/O area operation timing configuration register	0x00031909
PCCARD_COMM	0xE780_1830	R/W	PC card common memory area operation timing configuration register	0x00031909
Reserved	~ 0xE780_18FC	R/W	Reserved area	
ATAPI controller base address				
ATA_CONTROL	0xE780_1900	R/W	ATA enable and clock down status	0x00000002
ATA_STATUS	0xE780_1904	R	ATA status	0x00000008
ATA_COMMAND	0xE780_1908	R/W	ATA command	0x00000000
ATA_SWRST	0xE780_190C	R/W	ATA software reset	0x00000000
ATA_IRQ	0xE780_1910	R/W	ATA interrupt sources	0x00000000
ATA_IRQ_MASK	0xE780_1914	R/W	ATA interrut mask	0x00000000
ATA_CFG	0xE780_1918	R/W	ATA configuration for ATA interface	0x80000000
ATA_CFG_2	0xE780_191C	R/W	2^{nd} ATA configuration for ATA interface with EBI	0x01000000
Reserved	0xE780_1920	R/W	Reserved	
Reserved	0xE780_1924	R/W	Reserved	
ATA_MDMA_TIME	0xE780_1928	R/W	ATA multi-word DMA timing	0x0002c238
ATA_PIO_TIME	0xE780_192C	R/W	ATA PIO timing	0x000272fa
ATA_UDMA_TIME	0xE780_1930	R/W	ATA UDMA timing	0x080b1a83
ATA_XFR_NUM	0xE780_1934	R/W	ATA transfer number	0x00000000
ATA_XFR_CNT	0xE780_1938	R	ATA current transfer count	0x00000000
ATA_TBUF_BASE	0xE780_193C	R/W	ATA start address of track buffer	0x00000000
ATA_TBUF_SIZE	0xE780_1940	R/W	ATA size of track buffer	0x00000000
ATA_SBUF_BASE	0xE780_1944	R/W	ATA start address of source buffer	0x00000000

Register	Address	R/W	Description	Reset Value
ATA_SBUF_SIZE	0xE780_1948	R/W	ATA size of source buffer	0x00000000
ATA_CADR_TBUF	0xE780_194C	R	ATA current write address of track buffer	0x00000000
ATA_CADR_SBUF	0xE780_1950	R	ATA current read address of source buffer	0x00000000
ATA_PIO_DTR	0xE780_1954	R/W	ATA PIO device data register	0x00000000
ATA_PIO_FED	0xE780_1958	R/W	ATA PIO device Feature/ Error register	0x00000000
ATA_PIO_SCR	0xE780_195C	R/W	ATA PIO sector count register	0x00000000
ATA_PIO_LLR	0xE780_1960	R/W	ATA PIO device LBA low register	0x00000000
ATA_PIO_LMR	0xE780_1964	R/W	ATA PIO device LBA middle register	0x00000000
ATA_PIO_LHR	0xE780_1968	R/W	ATA PIO device LBA high register	0x00000000
ATA_PIO_DVR	0xE780_196C	R/W	ATA PIO device register	0x00000000
ATA_PIO_CSD	0xE780_1970	R/W	ATA PIO device command/ status register	0x00000000
ATA_PIO_DAD	0xE780_1974	R/W	ATA PIO device control/ alternate status register	0x00000000
ATA_PIO_READY	0xE780_1978	R	ATA PIO data read/ write ready	0x00000003
ATA_PIO_RDATA	0xE780_197C	R	ATA PIO read data from device data register	0x00000000
BUS_FIFO_STATUS	0xE780_1980	R	ATA internal AHB FIFO status	0x00000000
ATA_FIFO_STATUS	0xE780_1984	R	ATA internal ATA FIFO status	0x00000000

5.2 DETAILED DESCRIPTION

5.2.1 Top Level Control and Configuration Register (MUX_REG, R/W, Address = 0xE780_1800)

MUX_REG	Bit	Description	R/W	Initial State
Reserved	[31:3]	Reserved	R	0x0
ATA_Indep_Port	[2]	Indep. ATA I/F Select. 0 = EBI Muxed port 1 = Indep. ATA I/F (Modem I/F)	R/W	0x0
Reserved	[1]	Fixed to 1	R/W	0x1
IDE_MODE	[0]	Internal operation mode select 0 = PC card mode 1 = True-IDE mode	R/W	0x0

5.2.2 PC Card Configuration and Status Register (PCCARD_CNFG&STATUS, R/W, Address = 0xE780_1820)

PCCARD_CNFG &STATUS	Bit	Description	R/W	Initial State
Reserved	[31:14]	Reserved	R	0x0
CARD_RESET	[13]	CF card reset in PC card mode 0 = No reset 1 = Reset	R/W	0x0
INT_SEL	[12]	Selects Card interrupt request type 0 = falling edge triggering 1 = level Low triggering	R/W	0x0
nWAIT_EN	[11]	Enables nWAIT (from CF card) 0 = Disables (always ready) 1 = Enables	R/W	0x1
DEVICE_ATT	[10]	Device type is 16-bits or 8-bits (Attribute memory area) 0 = 8-bit device 1 = 16-bit device	R/W	0x1
DEVICE_COMM	[9]	Device type is 16-bits or 8-bits (Common memory area) 0 = 8-bit device 1 = 16-bit device	R/W	0x1
DEVICE_IO	[8]	Device type is 16-bits or 8-bits (I/O area) 0 = 8-bit device 1 = 16-bit device	R/W	0x1
Reserved	[7:4]	Reserved	R	0x0
NOCARD_ERR	[3]	No card operation 0 = No error 1 = Error	R	0x0
nWAIT	[2]	nWAIT from CF card 0 = Wait 1 = Ready	R	0x1
nIREQ	[1]	Interrupt request from CF card 0 = Interrupt request 1 = No interrupt request	R	0x1
nCD	[0]	Detects Card 0 = Detects Card 1 = Does not detect card	R	0xX

5.2.3 Interrupt Source & Mask Register (PCCCARD_INTMASK&SRC, R/W, Address = 0xE780_1824)

PCCCARD_INTMASK &SRC	Bit	Description	R/W	Initial State
Reserved	[31:11]	Reserved	R	0x0
INTMSK_ERR_N	[10]	Interrupt mask bit of no card error 0 = Unmask 1 = Mask	R/W	0x1
INTMSK_IREQ	[9]	Interrupt mask bit of CF card interrupt request 0 = Unmask 1 = Mask	R/W	0x1
INTMSK_CD	[8]	Interrupt mask bit of CF card detect 0 = Unmask 1 = Mask	R/W	0x1
Reserved	[7:3]	Reserved	R/W	0x0
INTSRC_ERR_N	[2]	If host access no card in slot. CPU clears this interrupt by writing "1".	R/W	0x0
INTSRC_IREQ	[1]	If CF card interrupt request CPU clears this interrupt by writing "1".	R/W	0xX
INTSRC_CD	[0]	If CF card is detected in slot CPU clears this interrupt by writing "1".	R	0xX

5.2.4 Attribute Memory Card Access Timing (PCCARD_ATTR, R/W, Address = 0xE780_1828)

PCCARD_ATTR	Bit	Description	R/W	Reset Value
Reserved	[31:23]	Reserved	R	0x0
HOLD_ATTR	[22:16]	Hold state timing of attribute memory area	R/W	0x03
Reserved	[15]	Reserved	R	0x0
CMND_ATTR	[14:8]	Command state timing of attribute memory area	R/W	0x19
Reserved	[7]	Reserved	R	0x0
SETUP_ATTR	[6:0]	Setup state timing of attribute memory area	R/W	0x09

5.2.5 I/O Card Access Timing (PCCARD_I/O, R/W, Address = 0xE780_182C)

PCCARD_I/O	Bit	Description	R/W	Reset Value
Reserved	[31:23]	Reserved	R	0x0
HOLD_IO	[22:16]	Hold state timing of I/O area	R/W	0x03
Reserved	[15]	Reserved	R	0x0
CMND_IO	[14:8]	Command state timing of I/O area	R/W	0x19
Reserved	[7]	Reserved	R	0x0
SETUP_IO	[6:0]	Setup state timing of I/O area	R/W	0x09

5.2.6 Common Memory Card Access Timing (PCCARD_COMM, R/W, Address = 0xE780_1830)

PCCARD_COMM	Bit	Description	R/W	Reset Value
Reserved	[31:23]	Reserved	R	0x0
HOLD_COMM	[22:16]	Hold state timing of common memory area	R/W	0x03
Reserved	[15]	Reserved	R	0x0
CMND_COMM	[14:8]	Command state timing of common memory area	R/W	0x19
Reserved	[7]	Reserved	R	0x0
SETUP_COMM	[6:0]	Setup state timing of common memory area	R/W	0x09

5.2.7 ATA Control Register (ATA_CONTROL, R/W, Address = 0xE780_1900)

ATA_CONTROL	Bit	Description	R/W	Reset Value
Reserved	[31:2]	Reserved	R	0x0
clk_down_ready	[1]	Status for clock down This bit is asserted in idle state if ATA_CONTROL bit [0] is zero. 0 = Not ready for clock down 1 = Ready for clock down	R	0x1
ata_enable	[0]	Enables ATA 0 = ATA is disabled and preparation for clock down maybe in progress 1 = Enables ATA.	R/W	0x0

5.2.8 ATA Status Register (ATA_STATUS, R, Address = 0xE780_1904)

ATA_STATUS	Bit	Description	R/W	Reset Value
Reserved	[31:6]	Reserved	R	0x0
atadev_cblid	[5]	ATAPI cable identification	R	0x0
atadev_irq	[4]	ATAPI interrupt signal line	R	0x0
atadev_iordy	[3]	ATAPI iordy signal line	R	0x1
atadev_dmareq	[2]	ATAPI dmareq signal line	R	0x0
xfr_state	[1:0]	Transfer state 2'b00 = Idle state 2'b01 = Transfer state 2'b10 = Abort state 2'b11 = Wait for completion state	R	0x0

5.2.9 ATA Command Register (ATA_COMMAND, R/W, Address = 0xE780_1908)

ATA_COMMAND	Bit	Description	R/W	Reset Value
Reserved	[31:2]	Reserved	R	0x0
xfr_command	[1:0]	ATA transfer command Four command types (START, STOP, ABORT and CONTINUE) are supported for data transfer control. The "START" command starts data transfer. The "STOP" command pause transfer temporarily. The "CONTINUE" command is used after "STOP" command or internal state of "pause" if track buffer is full or UDMA hold state. The "ABORT" command terminates current data transfer sequences and make ATA host controller move to idle state. 00 = Stop command 01 = Start command (Available in idle state) 10 = Abort command 11 = Continue command (Available in transfer pause) ** After CPU ABORT commands, make a software reset by ATA_SWRST to clear the leftover values of internal registers.	R/W	0x0

If CPU wants to pause data transfer use STOP command. Issue a CONTINUE command to send data continuously.

The STOP command controls the ATA Device side signal but does not control DMA side. Namely, if the FIFO has data after STOP command, DMA operation progresses until the FIFO becomes empty at read operation. In case of write operation, the DMA acts the same way until the FIFO becomes full.

Use the ABORT command if the transmitting data has proven useless data or discontinues absurd state by error interrupt from device.

At that time, it clears all data in ATA Host controller (register, FIFO) and the transmission state machine goes to IDLE.

The Software Reset's meaning becomes clear all registers even though the ABORT command had been executed before doing configuration register set for next transmission. It is not mandatory.

5.2.10 ATA Software Reset (ATA_SWRST, R/W, Address = 0xE780_190C)

ATA_SWRST	Bit	Description	R/W	Reset Value
Reserved	[31:1]	Reserved	R	0x0
ata_swrst	[0]	Software reset for the ATAPI host 0 = No reset 1 = Resets device registers and all registers of ATAPI host controller except CPU interface registers. After software reset, to continue transfer, user must configure all registers of host controller and device registers.	R/W	0x0

5.2.11 ATA Interrupt Register (ATA_IRQ, R/W, Address = 0xE780_1910)

ATA_IRQ	Bit	Description	R/W	Reset Value
Reserved	[31:10]	Reserved	R	0x0
ebi_abort_rd_int	[9]	When ATAPI is aborted by EBI-BACKOFF signal, in case of read transfer (UDMA/MDMA class). CPU clears this interrupt by writing "1".	R/W	0x0
ebi_abort_wr_int	[8]	When ATAPI is aborted by EBI-BACKOFF signal, in case of write transfer (UDMA/MDMA class). CPU clears this interrupt by writing "1".	R/W	0x0
ebi_bf_rd_int	[7]	When EBI-BACKOFF signal is issued by EBI, In case of read transfer. If CFCON release the EBI BUS, this bit clears automatically.	R/W	0x0
ebi_bf_wr_int	[6]	When EBI-BACKOFF signal is issued by EBI, in case of write transfer. If CFCON release the EBI BUS, this bit clears automatically.	R/W	0x0
mdma_hold_int	[5]	If ATAPI device makes pending in MDMA class. CPU clears this interrupt by writing "1".	R/W	0x0
sbuf_empty_int	[4]	If source buffer is empty. CPU clears this interrupt by writing "1".	R/W	0x0
tbuf_full_int	[3]	If track buffer is half-full. CPU clears this interrupt by writing "1".	R/W	0x0
atadev_irq_int	[2]	If ATAPI device generates interrupt. CPU clears this interrupt by writing "1".	R/W	0x0
udma_hold_int	[1]	If ATAPI device makes early termination in UDMA class. CPU clears this interrupt by writing "1".	R/W	0x0
xfr_done_int	[0]	If all data transfers are complete. CPU clears this interrupt by writing "1".	R/W	0x0

5.2.12 ATA Interrupt Mask Register (ATA_IRQ_MASK, R/W, Address = 0xE780_1914)

ATA_IRQ_MASK	Bit	Description	R/W	Reset Value
Reserved	[31:10]	Reserved	R	0x0
mask_ebi_abort_rd_int	[9]	0 = Mask ebi_abort_rd_int; Disable 1 = Unmask ebi_abort_rd_int; Enable	R/W	0x0
mask_ebi_abort_wr_int	[8]	0 = Mask ebi_abort_wr_int; Disable 1 = Unmask ebi_abort_wr_int; Enable	R/W	0x0
mask_ebi_bf_rd_int	[7]	0 = Mask ebi_bf_rd_int; Disable 1 = Unmask ebi_bf_rd_int; Enable	R/W	0x0
mask_ebi_bf_wr_int	[6]	0 = Mask ebi_bf_wr_int; Disable 1 = Unmask ebi_bf_wr_int; Enable	R/W	0x0
mask_mdma_hold_int	[5]	0 = Mask mdma_hold_int; Disable 1 = Unmask mdma_hold_int; Enable	R/W	0x0
mask_sbut_empty_int	[4]	0 = Mask sbut_empty_int; Disable 1 = Unmask sbuf_empty_int; Enable	R/W	0x0
mask_tbuf_full_int	[3]	0 = Mask tbuf_full_int; Disable 1 = Unmask tbuf_full_int; Enable	R/W	0x0
mask_atadev_irq_int	[2]	0 = Mask atadev_irq_int; Disable 1 = Unmask ata_irq_int; Enable	R/W	0x0
mask_udma_hold_int	[1]	0 = Mask udma_hold_int; Disable 1 = Unmask udma_hold_int; Enable	R/W	0x0
mask_xfr_done_int	[0]	0 = Mask xfr_done_int; Disable 1 = Unmask xfr_done_int; Enable	R/W	0x0

5.2.13 ATA Configuration Register (ATA_CFG, R/W, Address = 0xE780_1918)

ATA_CFG	Bit	Description	R/W	Reset Value
Reserved	[31]	Reserved (This field should be 0x1)	R/W	0x1
Reserved	[30:13]	Reserved	R	0x0
dma_mode	[12]	Determines whether DMA is normal DMA 0 = ormal DMA mode 1 = Reserved.	R/W	0x0
Reserved	[11]	Reserved	R	
word_swap	[10]	Determines whether endian is little or big in AHB word data. (half word swapping) 0 = Little endian {byte3, byte2, byte1, byte0} 1 = Big endian {byte1, byte0, byte3, byte2}	R/W	0x0
udma_auto_mode	[9]	Determines whether to continue automatically in case of early termination in UDMA mode by Device. This bit should not be changed during runtime operation. 0 = Stays in pause state and wait for CPU's action. 1 = Continue automatically	R/W	0x0
Reserved	[8]	Reserved	R	0x0
Reserved	[7]	Reserved	R	0x0
byte_swap	[6]	Determines whether data endian is little or big in 16-bit data. 0 = Little endian (data[15:8], data[7:0]) 1 = Big endian (data[7:0], data[15:8])	R/W	0x0
atadev_irq_al	[5]	Device interrupt signal level 0 = Active high 1 = Active low	R/W	0x0
dma_dir	[4]	DMA transfer direction 0 = Host read data from device 1 = Host write data to device	R/W	0x0
ata_class	[3:2]	Selects ATA transfer class 2'b00 = Transfer class is PIO 2'b01 = Transfer class is PIO DMA 2'b10 = Transfer class is Multi-word DMA 2'b11 = Transfer class is UDMA	R/W	0x0
ata_iordy_en	[1]	Determines whether IORDY input extends data transfer. 0 = Disables IORDY (ignored) 1 = Enables IORDY (can extend)	R/W	0x0
ata_rst	[0]	ATAPI device reset by this host. 0 = No reset 1 = Reset	R/W	0x0

5.2.14 2nd ATA Configuration for ATA Interface with EBI (ATA_CFG_2, R/W, Address = 0xE780_191C)

ATA_CFG_2	Bit	Description	R/W	Reset Value
ebi_sc_end_value	[31:24]	When support EBI BACKOFF, This value is used checked that current transfer sector is done. (For DEBUGGING)	R/W	0x1
Reserved	[23:6]	Reserved	R	0x0
udma_cnt_sel	[5]	In case of UDMA class, Indicate sector_end 0 = Sector counter is 32'h0000_0002. 1 = Sector counter is 32'h0000_0000.(For DEBUGGING)	R/W	0x0
Reserved	[4]	Reserved	R	0x0
ebi_rd_abort_en	[3]	Asserted auto-abort command, when EBI-BACKOFF is issued, in case of read transfer (UDMA/MDMA class). This bit only means when ebi_bf_en is "1". **Note:** Steps to Auto ABORT.. 1. EBI issues EBI-BACKOFF. 2. Complete operation of current transfer sector. 3. Assert ABORT command by H/W. (Release EBI BUS) 4. Issues interrupt (ebi_abort_rd_int). 5. CPU checks ATA_XFR_CNT. 6. Make software reset by ATA_SWRST. 7. Re-Start for remaining transition by S/W	R/W	0x0
ebi_wr_abort_en	[2]	Asserted auto-abort command, when EBI-BACKOFF is issued, in case of write transfer (UDMA/MDMA class). This bit only means when ebi_bf_en is "1". **Note:** Steps to Auto ABORT. 1. EBI issues EBI-BACKOFF. 2. Complete operation of current transfer sector. 3. Assert ABORT command by H/W. (Release EBI BUS) 4. Issues interrupt (ebi_abort_wr_int). 5. CPU checks ATA_XFR_CNT and ATA_CADDR_TBUF. 6. Make software reset by ATA_SWRST. 7. Re-Start for remaining transition by S/W	R/W	0x0
Reserved	[1]	Reserved	R	0x0
ebi_bf_en	[0]	Supports EBI-BACKOFF in UDMA/MDMA class. 0 = Disables (EBI BUS is released by S/W) 1 = Enables;	R/W	0x0

5.2.15 ATA Multi_word DMA Timing (ATA_MDMA_TIME, R/W, Address = 0xE780_1928)

ATA_MDMA_TIME	Bit	Description	R/W	Reset Value
Reserved	[31:20]	Reserved	R	0x0
dma_teoc	[19:12]	DMA timing parameter, Teoc, end of cycle time	R/W	0x2C
dma_t2	[11:4]	DMA timing parameter, tD, DIOR/DIOWn pulse width	R/W	0x23
dma_t1	[3:0]	DMA timing parameter, tM, CS0,1n valid to DIOR/Wn	R/W	0x8

5.2.16 ATA PIO Time (ATA_PIO_TIME, R/W, Address = 0xE780_192C)

ATA_PIO_TIME	Bit	Description	R/W	Reset Value
Reserved	[31:20]	Reserved	R	0x0
pio_teoc	[19:12]	PIO timing parameter, teoc, end of cycle time It shall not have zero value.	R/W	0x27
pio_t2	[11:4]	PIO timing parameter, t2, DIOR/Wn pulse width It cannot have zero value.	R/W	0x2f
pio_t1	[3:0]	PIO timing parameter, t1, address valid to DIOR/Wn	R/W	0xa

5.2.17 ATA UDMA Time (ATA_UDMA_TIME, R/W, Address = 0xE780_1930)

ATA_UDMA_TIME	Bit	Description	R/W	Reset Value
Reserved	[31:28]	Reserved	R	0x0
udma_tdvh	[27:24]	UDMA timing parameter tDVH	R/W	0x8
udma_tdvs	[23:16]	UDMA timing parameter tDVS It cannot have zero value.	R/W	0x0b
udma_trp	[15:8]	UDMA timing parameter tRP	R/W	0x1a
udma_tss	[7:4]	UDMA timing parameter, tSS	R/W	0x8
udma_tackenv	[3:0]	UDMA timing parameter tENV (envelope time (From DMACKn to STOP and HDMARDYn), tACK (setup and hold time for DMACKn)	R/W	0x3

5.2.18 ATA Transfer Count Number (ATA_XFR_NUM, R/W, Address = 0xE780_1934)

ATA_XFR_NUM	Bit	Description	R/W	Reset Value
xfr_num	[31:1]	Data transfer number. To transfer 1-sector (512-byte), you should set 32'h1ff.	R/W	0x00000000
Reserved	[0]	Reserved	R	0x0

5.2.19 ATA Current Transfer Count (ATA_XFR_CNT, R, Address = 0xE780_1938)

ATA_XFR_CNT	Bit	Description	R/W	Reset Value
xfr_cnt	[31:1]	Current remaining transfer counter. This value counts down from ATA_XFR_NUM. It goes to zero if all pre-defined data are transferred. In case of read transfer, ATA_XFR_NUM decreases by 1(2-byte) . In case of write transfer, ATA_XFR_NUM decreases by 16(32-byte), because the AHB burst size is 8.	R	0x00000000
Reserved	[0]	Reserved	R	0x0

5.2.20 Start Address of the Track Buffer (ATA_TBUF_BASE, R/W, Address = 0xE780_193C)

ATA_TBUF_BASE	Bit	Description	R/W	Reset Value
track_buffer_base	[31:2]	Start address of track buffer (4 byte unit)	R/W	0x00000000
Reserved	[1:0]	Reserved	R	0x0

5.2.21 Size of the Track Buffer (ATA_TBUF_SIZE, R/W, Address = 0xE780_1940)

ATA_TBUF_SIZE	Bit	Description	R/W	Reset Value
track_buffer_size	[31:5]	Size of track buffer (32 byte unit) This should be set to "size_of_data_in_bytes – 1". For example, to transfer 1-sector (512-byte, 32'h200), you should set 32'h1FF (= 32'h200 – 1).	R/W	0x0000000
Reserved	[4:0]	Reserved	R	0x00

5.2.22 Start Address of the Source Buffer (ATA_SBUF_BASE, R/W, Address = 0xE780_1944)

ATA_SBUF_BASE	Bit	Description	R/W	Reset Value
src_buffer_base	[31:2]	Start address of source buffer (4byte unit)	R/W	0x00000000
Reserved	[1:0]	Reserved	R	0x0

5.2.23 Size of Source Buffer (ATA_SBUF_SIZE, R/W, Address = 0xE780_1948)

ATA_SBUF_SIZE	Bit	Description	R/W	Reset Value
src_buffer_size	[31:5]	Size of source buffer (32byte unit) This should be set to "size_of_data_in_bytes – 1". For example, to transfer 1-sector (512-byte, 32'h200), you should set 32'h1FF (= 32'h200 – 1).	R/W	0x0000000
Reserved	[4:0]	Reserved	R	0x00

5.2.24 Current Address of Track Buffer (ATA_CADDR_TBUF, R, Address = 0xE780_194C)

ATA_CADDR_TBUF	Bit	Description	R/W	Reset Value
track_buf_cur_adr	[31:2]	Current address of track buffer	R	0x00000000
Reserved	[1:0]	Reserve	R	0x0

5.2.25 Current Address of Source Buffer (ATA_CADDR_SBUF, R, Address = 0xE780_1950)

ATA_CADDR_SBUF	Bit	Description	R/W	Reset Value
source_buf_cur_adr	[31:2]	Current address of source buffer	R	0x00000000
Reserved	[1:0]	Reserved	R	0x0

5.2.26 ATA PIO Data Register (ATA_PIO_DTR, R/W, Address = 0xE780_1954)

ATA_PIO_DTR	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	R	0x0
pio_dev_dtr	[15:0]	16-bit PIO data register	R/W	0x0000

5.2.27 ATA PIO Device Feature/Error Register (ATA_PIO_FED, R/W, Address = 0xE780_1958)

ATA_PIO_FED	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_fed	[7:0]	8-bit PIO device feature/ error (command block) register	R/W	0x00

5.2.28 ATA PIO Device Sector Count Register (ATA_PIO_SCR, R/W, Address = 0xE780_195C)

ATA_PIO_SCR	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_scr	[7:0]	8-bit PIO device sector count (command block) register	R/W	0x00

5.2.29 ATA PIO Device LBA Low Register (ATA_PIO_LLR, R/W, Address = 0xE780_1960)

ATA_PIO_LLR	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_llr	[7:0]	8-bit PIO device LBA low (command block) register	R/W	0x00

5.2.30 ATA PIO Device LBA Middle Register (ATA_PIO_LMR, R/W, Address = 0xE780_1964)

ATA_PIO_LMR	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_lmr	[7:0]	8-bit PIO device LBA middle (command block) register	R/W	0x00

5.2.31 ATA PIO Device LBA High Register (ATA_PIO_LHR, R/W, Address = 0xE780_1968)

ATA_PIO_LHR	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_lhr	[7:0]	8-bit PIO LBA high (command block) register	R/W	0x00

5.2.32 ATA PIO Device Register (ATA_PIO_DVR, R/W, Address = 0xE780_196C)

ATA_PIO_DVR	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_dvr	[7:0]	8-bit PIO device (command block) register	R/W	0x00

5.2.33 ATA PIO Device Command Status Register (ATA_PIO_CSD, R/W, Address = 0xE780_1970)

ATA_PIO_CSD	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_csd	[7:0]	8-bit PIO device command/status (command block) register	R/W	0x00

5.2.34 ATA PIO Device Control/Alternate Status Register (ATA_PIO_DAD, R/W, Address = 0xE780_1974)

ATA_PIO_DAD	Bit	Description	R/W	Reset Value
Reserved	[31:8]	Reserved	R	0x0
pio_dev_dad	[7:0]	8-bit PIO device control/ alternate status (control block) register	R/W	0x00

5.2.35 ATA PIO Data Ready Register (ATA_PIO_READY, R, Address = 0xE780_1978)

ATA_PIO_READY	Bit	Description	R/W	Reset Value
Reserved	[31:2]	Reserved	R	0x0
dev_acc_ready	[1]	Indicates whether host can start access to device register 0 = Not ready to start access ATA device register 1 = Ready to start access ATA device register	R	0x1
pio_data_ready	[0]	Indicates whether data is valid in ATA_PIO_DATA register 0 = No valid data in ATA_PIO_DATA register 1 = Valid data in ATA_PIO_DATA register	R	0x1

5.2.36 ATA PIO Read Data Register (ATA_PIO_RDATA, R, Address = 0xE780_197C)

ATA_PIO_RDATA	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	R	0x0
pio_rdata	[15:0]	PIO read data register while HOST read from ATA device register	R	0x0000

5.2.37 AHB Bus FIFO Status Register (BUS_FIFO_STATUS, R, Address = 0xE780_1980)

BUS_FIFO_STATUS	Bit	Description	R/W	Reset Value
Reserved	[31:19]	Reserved	R	0x0
bus_state[2:0]	[18:16]	3'b000 = IDLE 3'b001 = BUSYW 3'b010 = PREP 3'b011 = BUSYR 3'b100 = PAUSER 3'b101 = PAUSEW 3'b110 = PAUSER2	R	0x00
Reserved	[15:14]	Reserved	R	0x0
bus_fifo_rdpnt	[13:8]	Bus FIFO read pointer	R	0x00
Reserved	[7:6]	Reserved	R	0x0
bus_fifo_wrpnt	[5:0]	Bus FIFO write pointer	R	0x00

5.2.38 ATA FIFO Status Register (ATA_FIFO_STATUS, R, Address = 0xE780_1984)

ATA_FIFO_STATUS	Bit	Description	R/W	Reset Value
Reserved	[31]	Reserved	R	0x0
ata_state	[30:28]	0 = ATA_IDLE 1 = ATA_TRANS 2 = ATA_PAUSE 3 = ATA_PAUSE2 4 = ATA_ABORT	R	0x0000
pio_state	[27:26]	2'b00 = IDLE 2'b01 = T1 2'b10 = T2 2'b11 = TEOC	R	0x0
pdma_state	[25:24]	2'b00 = IDLE 2'b01 = T1 2'b10 = T2 2'b11 = TEOC	R	0x0
Reserved	[23]	Reserved	R	0x0
dma_state[1:0]	[22:21]	0 = IDLE 1 = TD 2 = TM 3 = TEOC	R	0x00
udma_state[4:0]	[20:16]	5'b00000 = IDLE 5'b00001 = TMI 5'b00010 = CRCS 5'b00011 = CRCH 5'b00100 = END 5'b01000 = STOPW 5'b01001 = ACKW 5'b01010 = NSEQWS 5'b01011 = NSEQWH 5'b01100 = SEQWS 5'b01101 = SEQWH 5'b01110 = TSSW 5'b10000 = STOPR 5'b10001 = ACKR 5'b10010 = NSEQR 5'b10011 = SEQR 5'b10100 = TRPR 5'b10101 = STRPR	R	0x00
Reserved	[15:0]	Reserved	R	0x0

5.6 EXTERNAL BUS INTERFACE

1 OVERVIEW

S5PC100 External Bus Interface (EBI), as a peripheral, relies on the memory controllers to release their external requests for the external bus when they are idle. Because it has no knowledge of when a transfer starts or completes. It enables for one SROM controller, one OneNAND controller, one NAND Flash controller and one CF controller to share one external memory bus named as memory port 0.

2 FEATURES

S5PC100 EBI features include:

Memory Port 0 is shared using EBI.

*** Reference:** ARM PrimeCell External Bus Interface (PL220), ARM DDI 0249B.

- AMBA AXI 3.0 low power interface (CSYSREQ, CACTIVE, CSYSACK) to prevent memory controller from accessing memories.
- Share of pad interface used by 4 memory controllers (SROMC, OneNANDC, NFCON, CFCON).
- Pad interface ownership is determined by the priority which can be changed.
- The handshaking between the EBI and the memory controller consists of a three-wire interface, EBIREQ, EBIGNT, and EBIBACKOFF, all active High.
  EBIREQ signals are asserted by memory controllers to indicate that they require external bus access.
  The respective arbitrated EBIGNT is issued to the highest priority memory controller.
  EBIBACKOFF is output of the EBI to signal that the memory controller must complete the current transfer and release the bus.
- The EBI arbitration scheme keeps track of the memory controller that is currently granted and waits for the transaction from that memory controller to finish (EBIREQ taken Low by the memory controller) before it grants the next memory controller. If a higher priority memory controller requests the bus then EBIBACKOFF signal tells the currently granted memory controller to terminate the current transfer as soon as possible.
- Memory Subsystem gets booting method and CS selection information from the System Controller.
- nCS0 and nCS1 in memory port 0 are dedicated for SROMC.
- If NAND Flash or OneNAND is selected for boot device, nCS2 is used to access that boot media.
- EBIGNT is required to be deasserted one cycle after EBIREQ is deasserted in sync. mode
- EBIBACKOFF is required to be deasserted one cycle after EBIREQ is deasserted in sync. mode
- In case EBIREQ is deasserted because of higher priority EBIBACKOFF, EBIREQ signal must be low for at least one clock cycle in sync. mode
- EBI_REQ duration is not required at least 4 cycle from CSYSREQ to CACTIVE

3 BLOCK DIAGRAM

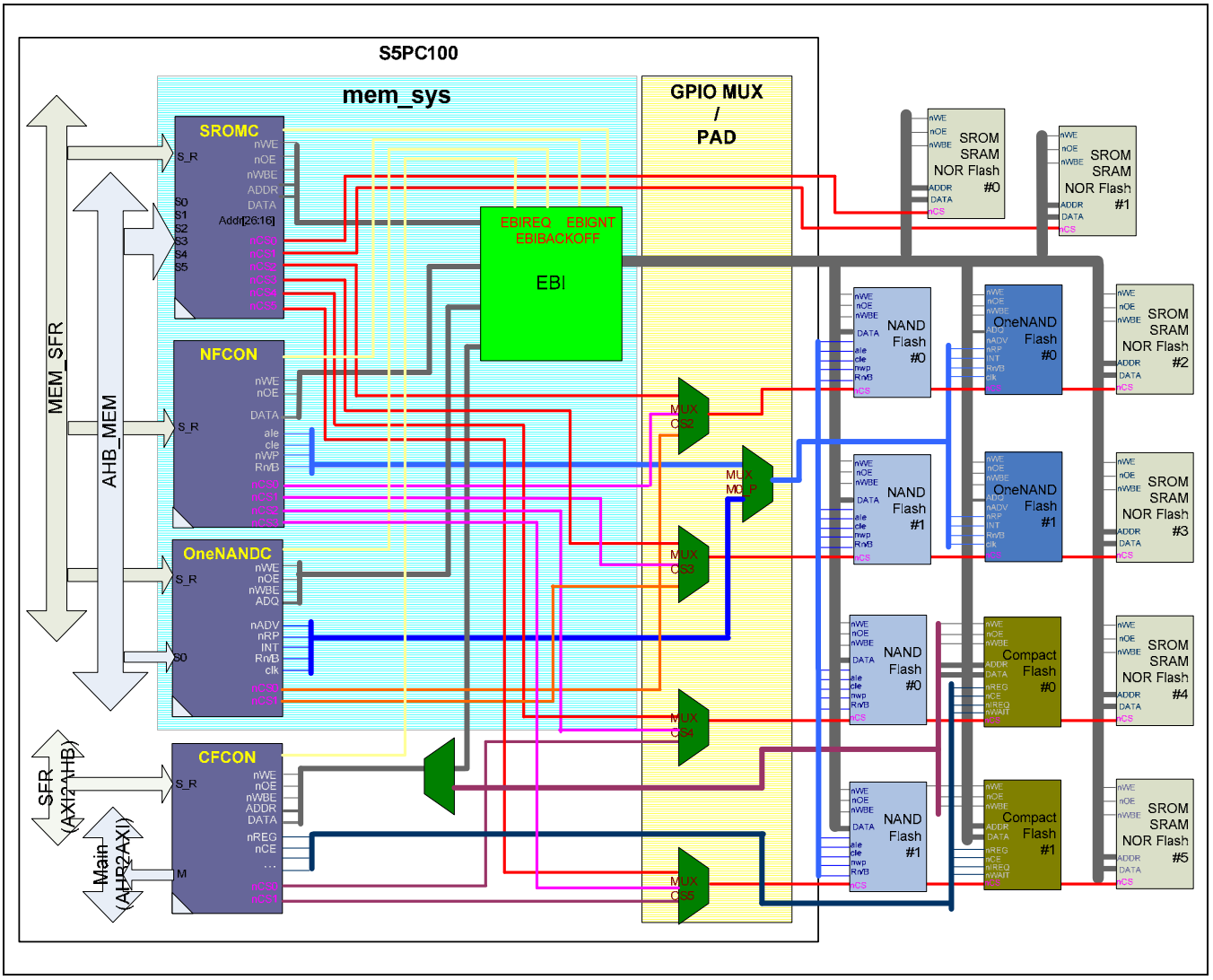


Figure 5.6-1 Memory Interface Through EBI

4 CLOCK SCHEME

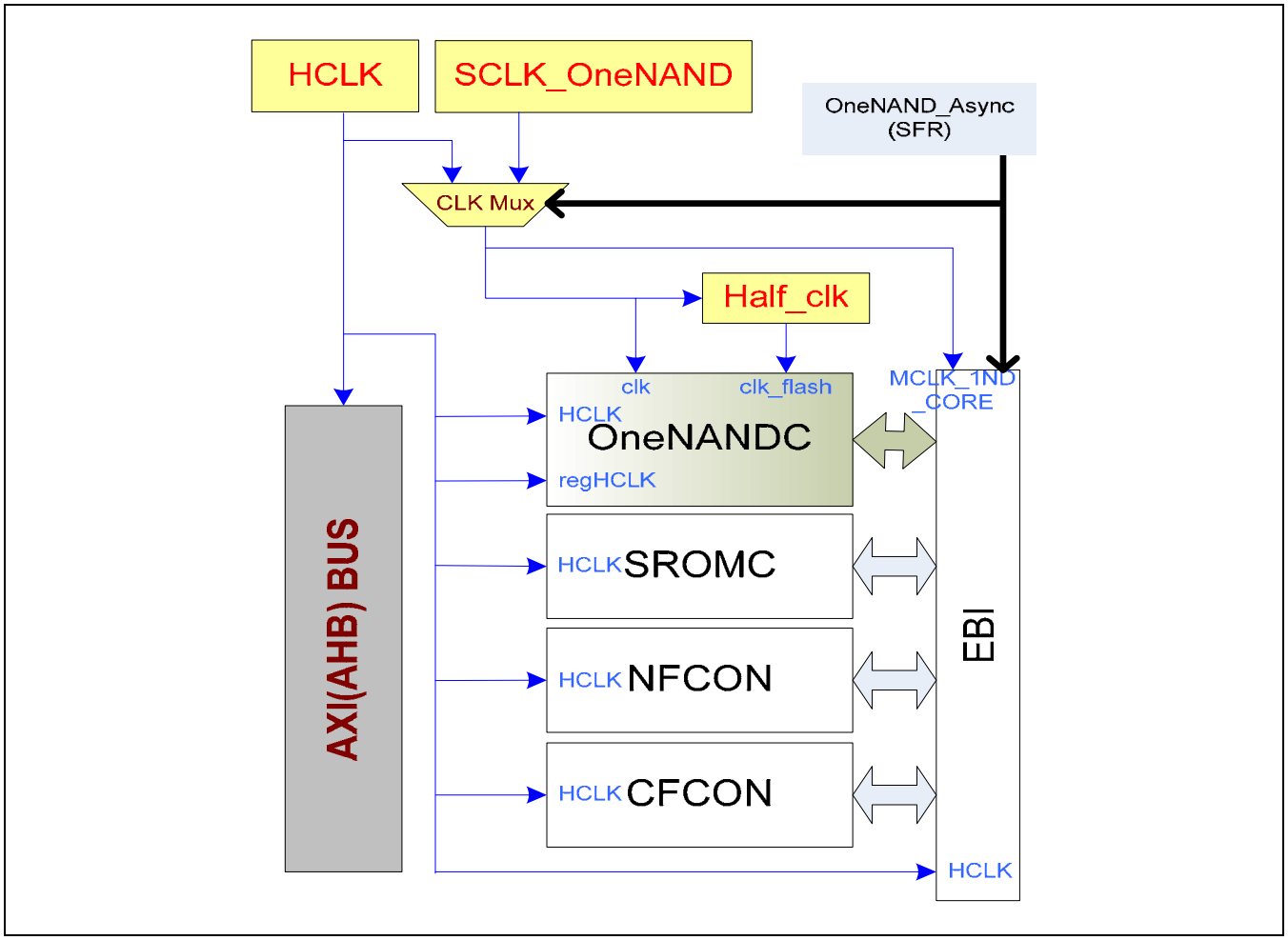


Figure 5.6-2 Clock Scheme of Memory Controllers and EBI

NOTE: The OneNAND Clock selection register name in Chapter 02.03. Clock Controller is OneNAND_SEL (OneNAND_Async). This register address is 0xE010_0200 (CLK_SRC0[24]).

5 FUNCTIONAL DESCRIPTION

Memory Subsystem can be configured by chapter 02.03. Clock controller. The memory configuration register name in System Controller is MEM_SYS_CFG. This register address is 0xE020_0200. Clock controller sends information based on fixed priority order. Fixed Priority Order is summarized in the table below:

EBI

- Priority Type
  - √ 0 = Fixed priority
  - √ 1 = Circular priority

Fixed Priority Order

CfgFixPriTyp[1:0]	1st	2nd	3rd	4th
0	SROMC	OneNANDC	NFCON	CFCON
1	OneNANDC	SROMC	NFCON	CFCON
2	NFCON	SROMC	OneNANDC	CFCON
3	CFCON	SROMC	OneNANDC	NFCON

6 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
EBI_DATA_RDn	Output	Bank Selection Signal	Xm0DATA_RDn	muxed

7 EBI I/O MUXING

Pad Name	Function Signal			
	SMC	NFCON	ONENANDC	CFCON
Xm0ADDR[20:0]	SMC_ADDR[20:0]			CF_A [10:0]
Xm0DATA[15:0]	SMC_DATA[15:0]	NF_DATA[15:0]	OND_DATA[15:0]	CF_D[15:0]
Xm0CSn[5:0]	SMC_nCS[5:0]	NF_nCS[3:0]	OND_CSn[1:0]	CF_nCS[1:0]
Xm0FRnB[1:0]		NF_RnB[3:0]	OND_INT[1:0]	
Xm0OEn	SMC_OEn		OND_OEn	
Xm0WEn	SMC_WEn		OND_WEn	
Xm0BEn[1:0]	SMC_Ben [1:0]			
Xm0WAITn	SMC_WAITn			
Xm0FCLE		NF_CLE	OND_AVALID	
Xm0FALE		NF_ALE	OND_SMCLK	
Xm0FREn		NF_REn		
Xm0FWEn		NF_WEn	OND_PRn	
Xm0IORDY				CF_IORDY
Xm0INTRQ				CF_INTRQ
Xm0INPACKn				CF_INPACKn
Xm0REG				CF_REG
Xm0IORDn				CF_IORDn
Xm0IOWRn				CF_IOWRn
Xm0CDn				CF_CDn
Xm0CFOEn				CF_OEn
Xm0CFWEn				CF_WEn
Xm0DATA_RDn	EBI_DATA_RDn			

6.1 DMA CONTROLLER

1 OVERVIEW

S5PC100 supports 2 Direct Memory Access (DMA) tops, one for Memory to Memory (M2M) transfer (DMA_mem), and the other one for Peripheral to memory transfer and vice-versa (DMA_peri). The M2M DMA top consists of a PL330 and some logics. The Peri DMA top consists of two PL330s, and dma_map.

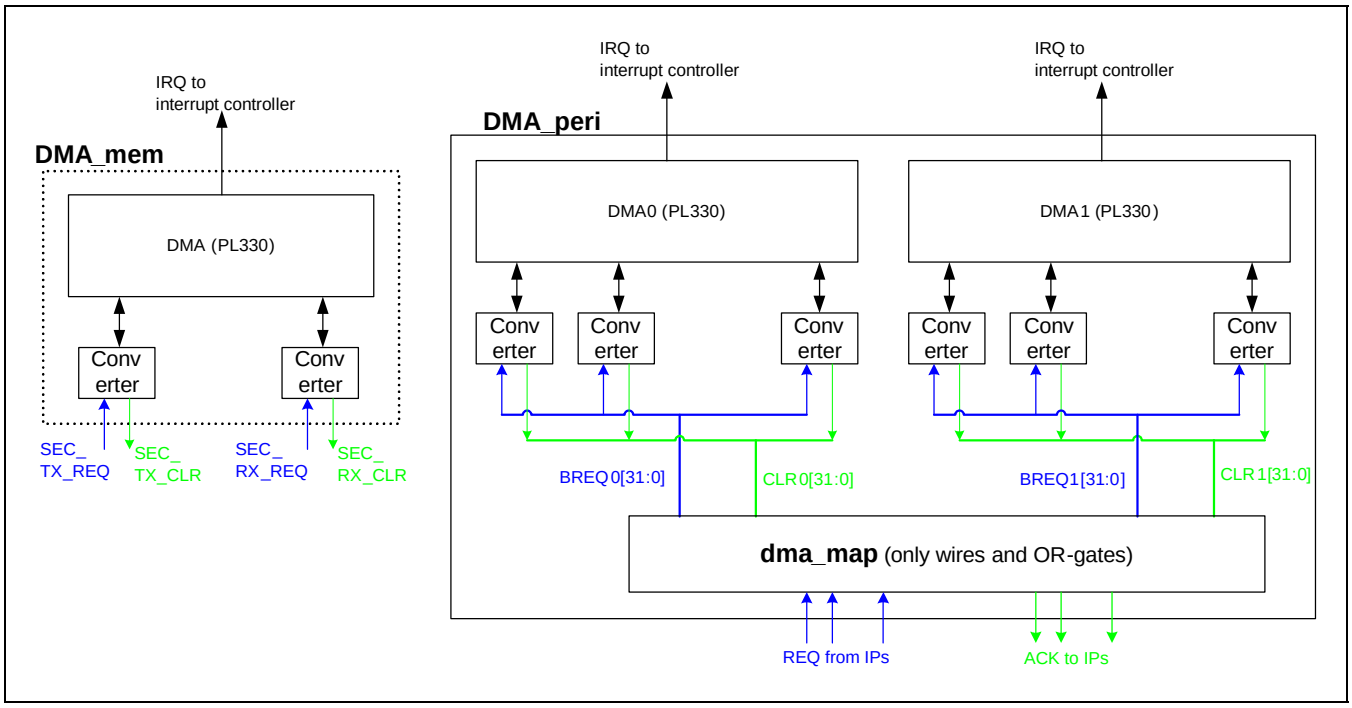


Figure 6.1-1 Two DMA Tops

All peripherals must be set as non-secure at TZPC module, because DMA_peri operates only as non-secure.

Bus interface of PL330 is AXI, so that DMA_mem is attached to AXI_B0, and DMA_peri is attached to AXI_B1 (For more information, refer **"03.01.Bus Configuration" chapter**).

2 FEATURES

Use the features listed in below table as reference for DMA and writing DMA assembly code.

	DMA_mem	DMA_peri
Supported Data Size	Up-to double word (64-bit)	Up-to word (32-bit)
Burst Size	Up-to 16 burst	Word transfer: Up-to 8 burst
		Byte or word transfer: Up-to 16 burst
Supported Channel	8 channels at the same time	8 channels at the same time

Although each DMA module has 32 interrupt sources, **only one interrupt is sent to VIC (Vectored Interrupt Controller) for each DMA. To see interrupt number of DMA, refer interrupt number table at "04.01.S5PC100_Interrupt Controller" chapter**. SW reads INTSTATUS (Interrupt Status) register for each module to keep track of occurred interrupt source.

Table 6.1-1 DMA Request Mapping Table

Module	No	DMA Request	Category	Service Module
Peri DMA1	31	Reserved		
	30	Reserved		
	29	MSM_REQ3	Others	by only DMA1
	28	MSM_REQ2		
	27	MSM_REQ1		
	26	MSM_REQ0		
	25	PCM1_TX	Audio and SPI	
	24	PCM1_RX		
	23	PCM0_TX		
	22	PCM0_RX		
	21	SPI2_TX		
	20	SPI2_RX		
	19	SPI1_TX		
	18	SPI1_RX		
	17	SPI0_TX		
	16	SPI0_RX		
	15	I2S2_TX		
	14	I2S2_RX		
	13	I2S1_TX		
	12	I2S1_RX		
	11	I2S0S_TX		
	10	I2S0_TX		

Module	No	DMA Request	Category	Service Module
	9	I2S0_RX		
	8	IrDA	System	
	7	UART3[1]		
	6	UART3[0]		
	5	UART2[1]		
	4	UART2[0]		
	3	UART1[1]		
	2	UART1[0]		
	1	UART0[1]		
	0	UART0[0]		
Peri DMA0	31	Reserved		
	30	Reserved		
	29	HSI_TX	Others	by only DMA0
	28	HSI_RX		
	27	SPDIF		
	26	PWM		
	25	EXTERNAL (GPIO)		
	24	AC_PCMout	Audio and SPI	
	23	AC_PCMin		
	22	AC_MICin		
	21	SPI2_TX		
	20	SPI2_RX		
	19	SPI1_TX		
	18	SPI1_RX		
	17	SPI0_TX		
	16	SPI0_RX		
	15	I2S2_TX		
	14	I2S2_RX		
	13	I2S1_TX		
	12	I2S1_RX		
	11	I2S0S_TX		
	10	I2S0_TX		
	9	I2S0_RX		
	8	IrDA	System	

Module	No	DMA Request	Category	Service Module
	7	UART3[1]		
	6	UART3[0]		
	5	UART2[1]		
	4	UART2[0]		
	3	UART1[1]		
	2	UART1[0]		
	1	UART0[1]		
	0	UART0[0]		
DMA_mem	1	SEC_TX	Security	by M2M DMA only
	0	SEC_RX		

3 REGISTER DESCRIPTION

Most of the SFRs are read-only. The main role of SFR is to check the PL330 status. There are many SFRs for PL330, therefore we have explained S5PC100-specific SFR and related summary. For more information refer to "Chapter 3 of PL330 TRM".

The Table 6.1-2 describes base address of each PL330 blocks.

3.1 REGISTER OVERVIEW

Table 6.1-2 DMA_mem Register Summary

Register	Address	R/W	Description	Reset Value
DS	0xE810_0000	R	DMA Status Register. For more information refer to "Page 3-11 of PL330 TRM"	0x0
DPC	0xE810_0004	R	DMA Program Counter Register. For more information refer to "Page 3-13 of PL330 TRM"	0x0
Reserved	0xE810_0008-0xE810_001C	-	Reserved	-
INTEN	0xE810_0020	R/W	Interrupt Enable Register. For more information refer to "Page 3-13 of PL330 TRM"	0x0
ES	0xE810_0024	R	Event Status Register. For more information refer to "Page 3-14 of PL330 TRM"	0x0
INTSTATUS	0xE810_0028	R	Interrupt Status Register. For more information refer to "Page 3-16 of PL330 TRM"	0x0
INTCLR	0xE810_002C	W	Interrupt Clear Register. For more information refer to "Page 3-17 of PL330 TRM"	0x0
FSM	0xE810_0030	R	Fault Status DMA Manager Register. For more information refer to "Page 3-18 of PL330 TRM"	0x0
FSC	0xE810_0034	R	Fault Status DMA Channel Register. For more information refer to "Page 3-19 of PL330 TRM"	0x0
FTM	0xE810_0038	R	Fault Type DMA Manager Register. For more information refer to "Page 3-20 of PL330 TRM"	0x0
Reserved	0xE810_003C	-	Reserved	-
FTC0	0xE810_0040	R	Fault type for DMA Channel 0	0x0
FTC1	0xE810_0044	R	Fault type for DMA Channel 1	0x0
FTC2	0xE810_0048	R	Fault type for DMA Channel 2	0x0
FTC3	0xE810_004C	R	Fault type for DMA Channel 3	0x0
FTC4	0xE810_0050	R	Fault type for DMA Channel 4	0x0
FTC5	0xE810_0054	R	Fault type for DMA Channel 5	0x0
FTC6	0xE810_0058	R	Fault type for DMA Channel 6	0x0
FTC7	0xE810_005C	R	Fault type for DMA Channel 7	0x0

Register	Address	R/W	Description	Reset Value
Reserved	0xE810_0060- 0xE810_00FC	-	Reserved	-
Channel Status Registers. For more information refer to "Page 3-24 of PL330 TRM"				
CS0	0xE810_0100	R	Channel Status for DMA Channel 0	0x0
CS1	0xE810_0108	R	Channel Status for DMA Channel 1	0x0
CS2	0xE810_0110	R	Channel Status for DMA Channel 2	0x0
CS3	0xE810_0118	R	Channel Status for DMA Channel 3	0x0
CS4	0xE810_0120	R	Channel Status for DMA Channel 4	0x0
CS5	0xE810_0128	R	Channel Status for DMA Channel 5	0x0
CS6	0xE810_0130	R	Channel Status for DMA Channel 6	0x0
CS7	0xE810_0138	R	Channel Status for DMA Channel 7	0x0
Channel Program Counter Registers. For more information refer to "Page 3-26 of PL330 TRM"				
CPC0	0xE810_0104	R	Channel PC for DMA Channel 0	0x0
CPC1	0xE810_010C	R	Channel PC for DMA Channel 1	0x0
CPC2	0xE810_0114	R	Channel PC for DMA Channel 2	0x0
CPC3	0xE810_011C	R	Channel PC for DMA Channel 3	0x0
CPC4	0xE810_0124	R	Channel PC for DMA Channel 4	0x0
CPC5	0xE810_012C	R	Channel PC for DMA Channel 5	0x0
CPC6	0xE810_0134	R	Channel PC for DMA Channel 6	0x0
CPC7	0xE810_013C	R	Channel PC for DMA Channel 7	0x0
Reserved	0xE810_0140- 0xE810_03FC	-	Reserved	-
Source Address Registers. For more information refer to "Page 3-27 of PL330 TRM"				
SA_0	0xE810_0400	R	Source Address for DMA Channel 0	0x0
SA_1	0xE810_0420	R	Source Address for DMA Channel 1	0x0
SA_2	0xE810_0440	R	Source Address for DMA Channel 2	0x0
SA_3	0xE810_0460	R	Source Address for DMA Channel 3	0x0
SA_4	0xE810_0480	R	Source Address for DMA Channel 4	0x0
SA_5	0xE810_04A0	R	Source Address for DMA Channel 5	0x0
SA_6	0xE810_04C0	R	Source Address for DMA Channel 6	0x0
SA_7	0xE810_04E0	R	Source Address for DMA Channel 7	0x0
Destination Address Registers. For more information refer to "Page 3-29 of PL330 TRM"				
DA_0	0xE810_0404	R	Destination Address for DMA Channel 0	0x0
DA_1	0xE810_0424	R	Destination Address for DMA Channel 1	0x0
DA_2	0xE810_0444	R	Destination Address for DMA Channel 2	0x0
DA_3	0xE810_0464	R	Destination Address for DMA Channel 3	0x0
DA_4	0xE810_0484	R	Destination Address for DMA Channel 4	0x0

Register	Address	R/W	Description	Reset Value
DA_5	0xE810_04A4	R	Destination Address for DMA Channel 5	0x0
DA_6	0xE810_04C4	R	Destination Address for DMA Channel 6	0x0
DA_7	0xE810_04E4	R	Destination Address for DMA Channel 7	0x0
Channel Control Registers. For more information refer to "Page 3-30 of PL330 TRM"				
CC_0	0xE810_0408	R	Channel Control for DMA Channel 0	0x0
CC_1	0xE810_0428	R	Channel Control for DMA Channel 1	0x0
CC_2	0xE810_0448	R	Channel Control for DMA Channel 2	0x0
CC_3	0xE810_0468	R	Channel Control for DMA Channel 3	0x0
CC_4	0xE810_0488	R	Channel Control for DMA Channel 4	0x0
CC_5	0xE810_04A8	R	Channel Control for DMA Channel 5	0x0
CC_6	0xE810_04C8	R	Channel Control for DMA Channel 6	0x0
CC_7	0xE810_04E8	R	Channel Control for DMA Channel 7	0x0
Loop Counter 0 Registers. For more information refer to "Page 3-35 of PL330 TRM"				
LC0_0	0xE810_040C	R	Loop Counter 0 for DMA Channel 0	0x0
LC0_1	0xE810_042C	R	Loop Counter 0 for DMA Channel 1	0x0
LC0_2	0xE810_044C	R	Loop Counter 0 for DMA Channel 2	0x0
LC0_3	0xE810_046C	R	Loop Counter 0 for DMA Channel 3	0x0
LC0_4	0xE810_048C	R	Loop Counter 0 for DMA Channel 4	0x0
LC0_5	0xE810_04AC	R	Loop Counter 0 for DMA Channel 5	0x0
LC0_6	0xE810_04CC	R	Loop Counter 0 for DMA Channel 6	0x0
LC0_7	0xE810_04EC	R	Loop Counter 0 for DMA Channel 7	0x0
Loop Counter 1 Registers. For more information refer to "Page 3-36 of PL330 TRM"				
LC1_0	0xE810_0410	R	Loop Counter 1 for DMA Channel 0	0x0
LC1_1	0xE810_0430	R	Loop Counter 1 for DMA Channel 1	0x0
LC1_2	0xE810_0450	R	Loop Counter 1 for DMA Channel 2	0x0
LC1_3	0xE810_0470	R	Loop Counter 1 for DMA Channel 3	0x0
LC1_4	0xE810_0490	R	Loop Counter 1 for DMA Channel 4	0x0
LC1_5	0xE810_04B0	R	Loop Counter 1 for DMA Channel 5	0x0
LC1_6	0xE810_04D0	R	Loop Counter 1 for DMA Channel 6	0x0
LC1_7	0xE810_04F0	R	Loop Counter 1 for DMA Channel 7	0x0
Reserved	0xE810_0414- 0xE810_041C	-	Reserved	-
Reserved	0xE810_0434- 0xE810_043C	-	Reserved	-
Reserved	0xE810_0454- 0xE810_045C	-	Reserved	-

Register	Address	R/W	Description	Reset Value
Reserved	0xE810_0474-0xE810_047C	-	Reserved	-
Reserved	0xE810_0494-0xE810_049C	-	Reserved	-
Reserved	0xE810_04B4-0xE810_04BC	-	Reserved	-
Reserved	0xE810_04D4-0xE810_04DC	-	Reserved	-
Reserved	0xE810_04F4-0xE810_0CFC	-	Reserved	-
DBGSTATUS	0xE810_0D00	R	Debug Status Register. For more information refer to "Page 3-37 of PL330 TRM"	0x0
DBGCMD	0xE810_0D04	W	Debug Command Register. For more information refer to "Page 3-37 of PL330 TRM"	-
DBGINST0	0xE810_0D08	W	Debug Instruction-0 Register. For more information refer to "Page 3-38 of PL330 TRM"	-
DBGINST1	0xE810_0D0C	W	Debug Instruction-1 Register. For more information refer to "Page 3-39 of PL330 TRM"	-
CR0	0xE810_0E00	R	Configuration Register 0. For more information refer to "Page 3-40 of PL330 TRM"	0x003E_1071
CR1	0xE810_0E04	R	Configuration Register 1. For more information refer to "Page 3-42 of PL330 TRM"	0x0000_0075
CR2	0xE810_0E08	R	Configuration Register 2. For more information refer to "Page 3-43 of PL330 TRM"	0x0
CR3	0xE810_0E0C	R	Configuration Register 3. For more information refer to "Page 3-44 of PL330 TRM"	0xFFFF_FFFF
CR4	0xE810_0E10	R	Configuration Register 4. For more information refer to "Page 3-45 of PL330 TRM"	0x0000_0003
CRDn	0xE810_0E14	R	Configuration Register Dn. For more information refer to "Page 3-46 of PL330 TRM"	0x01F7_3733
periph_id_n	0xE810_0FE0 - 0xE810_0FEC	R	Peripheral Identification Registers 0-3. For more information refer to "Page 3-48 of PL330 TRM"	Configuration-dependent
pcell_id_n	0xE810_0FF0 – 0xE810_0FFC	R	PrimeCell Identification Registers 0-3. For more information refer to "Page 3-50 of PL330 TRM"	Configuration-dependent

Table 6.1-3 DMA_peri0 Register Summary

Register	Address	R/W	Description	Reset Value
DS	0xE900_0000	R	DMA Status Register. For more information refer to "Page 3-11 of PL330 TRM"	0x0
DPC	0xE900_0004	R	DMA Program Counter Register. For more information refer to "Page 3-13 of PL330 TRM"	0x0
Reserved	0xE900_0008-0xE900_001C	-	Reserved	-
INTEN	0xE900_0020	R/W	Interrupt Enable Register. For more information refer to "Page 3-13 of PL330 TRM"	0x0
ES	0xE900_0024	R	Event Status Register. For more information refer to "Page 3-14 of PL330 TRM"	0x0
INTSTATUS	0xE900_0028	R	Interrupt Status Register. For more information refer to "Page 3-16 of PL330 TRM"	0x0
INTCLR	0xE900_002C	W	Interrupt Clear Register. For more information refer to "Page 3-17 of PL330 TRM"	0x0
FSM	0xE900_0030	R	Fault Status DMA Manager Register. For more information refer to "Page 3-18 of PL330 TRM"	0x0
FSC	0xE900_0034	R	Fault Status DMA Channel Register. For more information refer to "Page 3-19 of PL330 TRM"	0x0
FTM	0xE900_0038	R	Fault Type DMA Manager Register. For more information refer to "Page 3-20 of PL330 TRM"	0x0
Reserved	0xE900_003C	-	Reserved	-
FTC0	0xE900_0040	R	Fault Type for DMA Channel 0	0x0
FTC1	0xE900_0044	R	Fault Type for DMA Channel 1	0x0
FTC2	0xE900_0048	R	Fault Type for DMA Channel 2	0x0
FTC3	0xE900_004C	R	Fault Type for DMA Channel 3	0x0
FTC4	0xE900_0050	R	Fault Type for DMA Channel 4	0x0
FTC5	0xE900_0054	R	Fault Type for DMA Channel 5	0x0
FTC6	0xE900_0058	R	Fault Type for DMA Channel 6	0x0
FTC7	0xE900_005C	R	Fault Type for DMA Channel 7	0x0
Reserved	0xE900_0060-0xE900_00FC	-	Reserved	-
Channel Status Registers. For more information refer to "Page 3-24 of PL330 TRM"				
CS0	0xE900_0100	R	Channel Status for DMA Channel 0	0x0
CS1	0xE900_0108	R	Channel Status for DMA Channel 1	0x0
CS2	0xE900_0110	R	Channel Status for DMA Channel 2	0x0
CS3	0xE900_0118	R	Channel Status for DMA Channel 3	0x0
CS4	0xE900_0120	R	Channel Status for DMA Channel 4	0x0
CS5	0xE900_0128	R	Channel Status for DMA Channel 5	0x0

Register	Address	R/W	Description	Reset Value
CS6	0xE900_0130	R	Channel Status for DMA Channel 6	0x0
CS7	0xE900_0138	R	Channel Status for DMA Channel 7	0x0
Channel Program Counter Registers. For more information refer to "Page 3-26 of PL330 TRM"				
CPC0	0xE900_0104	R	Channel PC for DMA Channel 0	0x0
CPC1	0xE900_010C	R	Channel PC for DMA Channel 1	0x0
CPC2	0xE900_0114	R	Channel PC for DMA Channel 2	0x0
CPC3	0xE900_011C	R	Channel PC for DMA Channel 3	0x0
CPC4	0xE900_0124	R	Channel PC for DMA Channel 4	0x0
CPC5	0xE900_012C	R	Channel PC for DMA Channel 5	0x0
CPC6	0xE900_0134	R	Channel PC for DMA Channel 6	0x0
CPC7	0xE900_013C	R	Channel PC for DMA Channel 7	0x0
Reserved	0xE900_0140- 0xE900_03FC	-	Reserved	-
Source Address Registers. For more information refer to "Page 3-27 of PL330 TRM"				
SA_0	0xE900_0400	R	Source Address for DMA Channel 0	0x0
SA_1	0xE900_0420	R	Source Address for DMA Channel 1	0x0
SA_2	0xE900_0440	R	Source Address for DMA Channel 2	0x0
SA_3	0xE900_0460	R	Source Address for DMA Channel 3	0x0
SA_4	0xE900_0480	R	Source Address for DMA Channel 4	0x0
SA_5	0xE900_04A0	R	Source Address for DMA Channel 5	0x0
SA_6	0xE900_04C0	R	Source Address for DMA Channel 6	0x0
SA_7	0xE900_04E0	R	Source Address for DMA Channel 7	0x0
Destination Address Registers. For more information refer to "Page 3-29 of PL330 TRM"				
DA_0	0xE900_0404	R	Destination Address for DMA Channel 0	0x0
DA_1	0xE900_0424	R	Destination Address for DMA Channel 1	0x0
DA_2	0xE900_0444	R	Destination Address for DMA Channel 2	0x0
DA_3	0xE900_0464	R	Destination Address for DMA Channel 3	0x0
DA_4	0xE900_0484	R	Destination Address for DMA Channel 4	0x0
DA_5	0xE900_04A4	R	Destination Address for DMA Channel 5	0x0
DA_6	0xE900_04C4	R	Destination Address for DMA Channel 6	0x0
DA_7	0xE900_04E4	R	Destination Address for DMA Channel 7	0x0
Channel Control Registers. For more information refer to "Page 3-30 of PL330 TRM"				
CC_0	0xE900_0408	R	Channel Control for DMA Channel 0	0x0
CC_1	0xE900_0428	R	Channel Control for DMA Channel 1	0x0
CC_2	0xE900_0448	R	Channel Control for DMA Channel 2	0x0
CC_3	0xE900_0468	R	Channel Control for DMA Channel 3	0x0
CC_4	0xE900_0488	R	Channel Control for DMA Channel 4	0x0

Register	Address	R/W	Description	Reset Value
CC_5	0xE900_04A8	R	Channel Control for DMA Channel 5	0x0
CC_6	0xE900_04C8	R	Channel Control for DMA Channel 6	0x0
CC_7	0xE900_04E8	R	Channel Control for DMA Channel 7	0x0
Loop Counter 0 Registers. For more information refer to "Page 3-35 of PL330 TRM"				
LC0_0	0xE900_040C	R	Loop Counter 0 for DMA Channel 0	0x0
LC0_1	0xE900_042C	R	Loop Counter 0 for DMA Channel 1	0x0
LC0_2	0xE900_044C	R	Loop Counter 0 for DMA Channel 2	0x0
LC0_3	0xE900_046C	R	Loop Counter 0 for DMA Channel 3	0x0
LC0_4	0xE900_048C	R	Loop Counter 0 for DMA Channel 4	0x0
LC0_5	0xE900_04AC	R	Loop Counter 0 for DMA Channel 5	0x0
LC0_6	0xE900_04CC	R	Loop Counter 0 for DMA Channel 6	0x0
LC0_7	0xE900_04EC	R	Loop Counter 0 for DMA Channel 7	0x0
Loop Counter 1 Registers. For more information refer to "Page 3-36 of PL330 TRM"				
LC1_0	0xE900_0410	R	Loop Counter 1 for DMA Channel 0	0x0
LC1_1	0xE900_0430	R	Loop Counter 1 for DMA Channel 1	0x0
LC1_2	0xE900_0450	R	Loop Counter 1 for DMA Channel 2	0x0
LC1_3	0xE900_0470	R	Loop Counter 1 for DMA Channel 3	0x0
LC1_4	0xE900_0490	R	Loop Counter 1 for DMA Channel 4	0x0
LC1_5	0xE900_04B0	R	Loop Counter 1 for DMA Channel 5	0x0
LC1_6	0xE900_04D0	R	Loop Counter 1 for DMA Channel 6	0x0
LC1_7	0xE900_04F0	R	Loop Counter 1 for DMA Channel 7	0x0
Reserved	0xE900_0414- 0xE900_041C	-	Reserved	-
Reserved	0xE900_0434- 0xE900_043C	-	Reserved	-
Reserved	0xE900_0454- 0xE900_045C	-	Reserved	-
Reserved	0xE900_0474- 0xE900_047C	-	Reserved	-
Reserved	0xE900_0494- 0xE900_049C	-	Reserved	-
Reserved	0xE900_04B4- 0xE900_04BC	-	Reserved	-
Reserved	0xE900_04D4- 0xE900_04DC	-	Reserved	-
Reserved	0xE900_04F4- 0xE900_0CFC	-	Reserved	-
DBGSTATUS	0xE900_0D00	R	Debug Status Register on page 3-37 of TRM	0x0

Register	Address	R/W	Description	Reset Value
DBGCMD	0xE900_0D04	W	Debug Command Register. For more information refer to "Page 3-37 of PL330 TRM"	-
DBGINST0	0xE900_0D08	W	Debug Instruction-0 Register. For more information refer to "Page 3-38 of PL330 TRM"	-
DBGINST1	0xE900_0D0C	W	Debug Instruction-1 Register. For more information refer to "Page 3-39 of PL330 TRM"	-
CR0	0xE900_0E00	R	Configuration Register 0. For more information refer to "Page 3-40 of PL330 TRM"	0x003F_F075
CR1	0xE900_0E04	R	Configuration Register 1. For more information refer to "Page 3-42 of PL330 TRM"	0x0000_0074
CR2	0xE900_0E08	R	Configuration Register 2. For more information refer to "Page 3-43 of PL330 TRM"	0x0000_0000
CR3	0xE900_0E0C	R	Configuration Register 3. For more information refer to "Page 3-44 of PL330 TRM"	0xFFFF_FFFF
CR4	0xE900_0E10	R	Configuration Register 4. For more information refer to "Page 3-45 of PL330 TRM"	0xFFFF_FFFF
CRDn	0xE900_0E14	R	Configuration Register Dn. For more information refer to "Page 3-46 of PL330 TRM"	0x0077_3732
periph_id_n	0xE900_0FE0 - 0xE900_0FEC	R	Peripheral Identification Registers 0-3. For more information refer to "Page 3-48 of PL330 TRM"	Configuration-dependent
pcell_id_n	0xE900_0FF0 – 0xE900_0FFC	R	PrimeCell Identification Registers 0-3. For more information refer to "Page 3-50 of PL330 TRM"	Configuration-dependent

Table 6.1-4 DMA_peri1 Register Summary

Register	Address	R/W	Description	Reset Value
DS	0xE920_0000	R	DMA Status Register. For more information refer to "Page 3-11 of PL330 TRM"	0x0
DPC	0xE920_0004	R	DMA Program Counter Register. For more information refer to "Page 3-13 of PL330 TRM"	0x0
Reserved	0xE920_0008-0xE920_001C	-	Reserved	-
INTEN	0xE920_0020	R/W	Interrupt Enable Register. For more information refer to "Page 3-13 of PL330 TRM"	0x0
ES	0xE920_0024	R	Event Status Register. For more information refer to "Page 3-14 of PL330 TRM"	0x0
INTSTATUS	0xE920_0028	R	Interrupt Status Register. For more information refer to "Page 3-16 of PL330 TRM"	0x0
INTCLR	0xE920_002C	W	Interrupt Clear Register. For more information refer to "Page 3-17 of PL330 TRM"	0x0
FSM	0xE920_0030	R	Fault Status DMA Manager Register. For more information refer to "Page 3-18 of PL330 TRM"	0x0
FSC	0xE920_0034	R	Fault Status DMA Channel Register. For more information refer to "Page 3-19 of PL330 TRM"	0x0
FTM	0xE920_0038	R	Fault Type DMA Manager Register. For more information refer to "Page 3-20 of PL330 TRM"	0x0
Reserved	0xE920_003C	-	Reserved	-
FTC0	0xE920_0040	R	Fault Type for DMA Channel 0	0x0
FTC1	0xE920_0044	R	Fault Type for DMA Channel 1	0x0
FTC2	0xE920_0048	R	Fault Type for DMA Channel 2	0x0
FTC3	0xE920_004C	R	Fault Type for DMA Channel 3	0x0
FTC4	0xE920_0050	R	Fault Type for DMA Channel 4	0x0
FTC5	0xE920_0054	R	Fault Type for DMA Channel 5	0x0
FTC6	0xE920_0058	R	Fault Type for DMA Channel 6	0x0
FTC7	0xE920_005C	R	Fault Type for DMA Channel 7	0x0
Reserved	0xE920_0060-0xE920_00FC	-	Reserved	-
Channel Status Registers. For more information refer to "Page 3-24 of PL330 TRM"				
CS0	0xE920_0100	R	Channel Status for DMA Channel 0	0x0
CS1	0xE920_0108	R	Channel Status for DMA Channel 1	0x0
CS2	0xE920_0110	R	Channel Status for DMA Channel 2	0x0
CS3	0xE920_0118	R	Channel Status for DMA Channel 3	0x0
CS4	0xE920_0120	R	Channel Status for DMA Channel 4	0x0
CS5	0xE920_0128	R	Channel Status for DMA Channel 5	0x0

Register	Address	R/W	Description	Reset Value
CS6	0xE920_0130	R	Channel Status for DMA Channel 6	0x0
CS7	0xE920_0138	R	Channel Status for DMA Channel 7	0x0
Channel Program Counter Registers. For more information refer to "Page 3-26 of PL330 TRM"				
CPC0	0xE920_0104	R	Channel PC for DMA Channel 0	0x0
CPC1	0xE920_010C	R	Channel PC for DMA Channel 1	0x0
CPC2	0xE920_0114	R	Channel PC for DMA Channel 2	0x0
CPC3	0xE920_011C	R	Channel PC for DMA Channel 3	0x0
CPC4	0xE920_0124	R	Channel PC for DMA Channel 4	0x0
CPC5	0xE920_012C	R	Channel PC for DMA Channel 5	0x0
CPC6	0xE920_0134	R	Channel PC for DMA Channel 6	0x0
CPC7	0xE920_013C	R	Channel PC for DMA Channel 7	0x0
Reserved	0xE920_0140- 0xE920_03FC	-	Reserved	-
Source Address Registers. For more information refer to "Page 3-27 of PL330 TRM"				
SA_0	0xE920_0400	R	Source Address for DMA Channel 0	0x0
SA_1	0xE920_0420	R	Source Address for DMA Channel 1	0x0
SA_2	0xE920_0440	R	Source Address for DMA Channel 2	0x0
SA_3	0xE920_0460	R	Source Address for DMA Channel 3	0x0
SA_4	0xE920_0480	R	Source Address for DMA Channel 4	0x0
SA_5	0xE920_04A0	R	Source Address for DMA Channel 5	0x0
SA_6	0xE920_04C0	R	Source Address for DMA Channel 6	0x0
SA_7	0xE920_04E0	R	Source Address for DMA Channel 7	0x0
Destination Address Registers. For more information refer to "Page 3-29 of PL330 TRM"				
DA_0	0xE920_0404	R	Destination Address for DMA Channel 0	0x0
DA_1	0xE920_0424	R	Destination Address for DMA Channel 1	0x0
DA_2	0xE920_0444	R	Destination Address for DMA Channel 2	0x0
DA_3	0xE920_0464	R	Destination Address for DMA Channel 3	0x0
DA_4	0xE920_0484	R	Destination Address for DMA Channel 4	0x0
DA_5	0xE920_04A4	R	Destination Address for DMA Channel 5	0x0
DA_6	0xE920_04C4	R	Destination Address for DMA Channel 6	0x0
DA_7	0xE920_04E4	R	Destination Address for DMA Channel 7	0x0
Channel Control Registers. For more information refer to "Page 3-30 of PL330 TRM"				
CC_0	0xE920_0408	R	Channel Control for DMA Channel 0	0x0
CC_1	0xE920_0428	R	Channel Control for DMA Channel 1	0x0
CC_2	0xE920_0448	R	Channel Control for DMA Channel 2	0x0
CC_3	0xE920_0468	R	Channel Control for DMA Channel 3	0x0
CC_4	0xE920_0488	R	Channel Control for DMA Channel 4	0x0

Register	Address	R/W	Description	Reset Value
CC_5	0xE920_04A8	R	Channel Control for DMA Channel 5	0x0
CC_6	0xE920_04C8	R	Channel Control for DMA Channel 6	0x0
CC_7	0xE920_04E8	R	Channel Control for DMA Channel 7	0x0
Loop Counter 0 Registers. For more information refer to "Page 3-35 of PL330 TRM"				
LC0_0	0xE920_040C	R	Loop Counter 0 for DMA Channel 0	0x0
LC0_1	0xE920_042C	R	Loop Counter 0 for DMA Channel 1	0x0
LC0_2	0xE920_044C	R	Loop Counter 0 for DMA Channel 2	0x0
LC0_3	0xE920_046C	R	Loop Counter 0 for DMA Channel 3	0x0
LC0_4	0xE920_048C	R	Loop Counter 0 for DMA Channel 4	0x0
LC0_5	0xE920_04AC	R	Loop Counter 0 for DMA Channel 5	0x0
LC0_6	0xE920_04CC	R	Loop Counter 0 for DMA Channel 6	0x0
LC0_7	0xE920_04EC	R	Loop Counter 0 for DMA Channel 7	0x0
Loop Counter 1 Registers. For more information refer to "Page 3-36 of PL330 TRM"				
LC1_0	0xE920_0410	R	Loop Counter 1 for DMA Channel 0	0x0
LC1_1	0xE920_0430	R	Loop Counter 1 for DMA Channel 1	0x0
LC1_2	0xE920_0450	R	Loop Counter 1 for DMA Channel 2	0x0
LC1_3	0xE920_0470	R	Loop Counter 1 for DMA Channel 3	0x0
LC1_4	0xE920_0490	R	Loop Counter 1 for DMA Channel 4	0x0
LC1_5	0xE920_04B0	R	Loop Counter 1 for DMA Channel 5	0x0
LC1_6	0xE920_04D0	R	Loop Counter 1 for DMA Channel 6	0x0
LC1_7	0xE920_04F0	R	Loop Counter 1 for DMA Channel 7	0x0
Reserved	0xE920_0414- 0xE920_041C	-	Reserved	-
Reserved	0xE920_0434- 0xE920_043C	-	Reserved	-
Reserved	0xE920_0454- 0xE920_045C	-	Reserved	-
Reserved	0xE920_0474- 0xE920_047C	-	Reserved	-
Reserved	0xE920_0494- 0xE920_049C	-	Reserved	-
Reserved	0xE920_04B4- 0xE920_04BC	-	Reserved	-
Reserved	0xE920_04D4- 0xE920_04DC	-	Reserved	-
Reserved	0xE920_04F4- 0xE920_0CFC	-	Reserved	-

Register	Address	R/W	Description	Reset Value
DBGSTATUS	0xE920_0D00	R	Debug Status Register. For more information refer to "Page 3-37 of PL330 TRM"	0x0
DBGCMD	0xE920_0D04	W	Debug Command Register. For more information refer to "Page 3-37 of PL330 TRM"	-
DBGINST0	0xE920_0D08	W	Debug Instruction-0 Register. For more information refer to "Page 3-38 of PL330 TRM"	-
DBGINST1	0xE920_0D0C	W	Debug Instruction-1 Register. For more information refer to "Page 3-39 of PL330 TRM"	-
CR0	0xE920_0E00	R	Configuration Register 0. For more information refer to "Page 3-40 of PL330 TRM"	0x003F_F075
CR1	0xE920_0E04	R	Configuration Register 1. For more information refer to "Page 3-42 of PL330 TRM"	0x0000_0074
CR2	0xE920_0E08	R	Configuration Register 2. For more information refer to "Page 3-43 of PL330 TRM"	0x0000_0000
CR3	0xE920_0E0C	R	Configuration Register 3. For more information refer to "Page 3-44 of PL330 TRM"	0xFFFF_FFFF
CR4	0xE920_0E10	R	Configuration Register 4. For more information refer to "Page 3-45 of PL330 TRM"	0xFFFF_FFFF
CRDn	0xE920_0E14	R	Configuration Register Dn. For more information refer to "Page 3-46 of PL330 TRM"	0x0077_3732
periph_id_n	0xE920_0FE0-0xE920_0FEC	R	Peripheral Identification Registers 0-3. For more information refer to "Page 3-48 of PL330 TRM"	Configuration-dependent
pcell_id_n	0xE920_0FF0-0xE920_0FFC	R	PrimeCell Identification Registers 0-3. For more information refer to "Page 3-50 of PL330 TRM"	Configuration-dependent

Below SFR descriptions show just restricted and fixed part of some SFR. PL330 TRM shows detailed information of other parts and other SFRs.

3.2 DETAILED DESCRIPTION

3.2.1 Channel Control Register for DMA_mem (CC, R)

- CC_0, R, Address = 0xE810_0408
- CC_1, R, Address = 0xE810_0428
- CC_2, R, Address = 0xE810_0448
- CC_3, R, Address = 0xE810_0468
- CC_4, R, Address = 0xE810_0488
- CC_5, R, Address = 0xE810_04A8
- CC_6, R, Address = 0xE810_04C8
- CC_7, R, Address = 0xE810_04E8

CCn	Bit	Description	Reset Value
dst_burst_size	[17:15]	Programs the burst size that the DMAC uses when it writes the destination data b000 = 1 byte b001 = 2 bytes b010 = 4 bytes b011 = 8 bytes Other = Reserved	0
src_burst_size	[3:1]	Programs the burst size that the DMAC uses when it reads the source data b000 = 1 byte b001 = 2 bytes b010 = 4 bytes b011 = 8 bytes Other = Reserved	0

3.2.2 Channel Control Register for DMA_PERI(0,1) (CC, R)

- CC_0, R, Address = 0xE900_0408, 0xE920_0408
- CC_1, R, Address = 0xE900_0428, 0xE920_0428
- CC_2, R, Address = 0xE900_0448, 0xE920_0448
- CC_3, R, Address = 0xE900_0468, 0xE920_0468
- CC_4, R, Address = 0xE900_0488, 0xE920_0488
- CC_5, R, Address = 0xE900_04A8, 0xE920_04A8
- CC_6, R, Address = 0xE900_04C8, 0xE920_04C8
- CC_7, R, Address = 0xE900_04E8, 0xE920_04E8

CCn	Bit	Description	Reset Value
dst_burst_size	[17:15]	Programs the burst size that the DMAC uses when it writes the destination data b000 = 1 byte b001 = 2 bytes b010 = 4 bytes Other = Reserved	0
src_burst_size	[3:1]	Programs the burst size that the DMAC uses when it reads the source data b000 = 1 byte b001 = 2 bytes b010 = 4 bytes Other = Reserved	0

3.2.3 Configuration Register0 for DMA_PERI(0,1) (CR0, R)

- CR0 for DMA_PERI0, R, Address = 0xE900_0E00
- CR0 for DMA_PERI1, R, Address = 0xE920_0E00

CR0	Bit	Description	Reset Value
num_events	[21:17]	Number of interrupt outputs that the DMAC provides b11111 = 32 interrupt outputs, irq[31:0]	0x1F
num_periph_req	[16:12]	Number of peripheral request interfaces that the DMAC provides b11111 = 32 peripheral request interfaces	0x1F
num_chnls	[6:4]	Number of DMA channels that the DMAC supports b111 = 8 DMA channels	0x7
mgr_ns_at_rst	[2]	Indicates the status of the boot_manager_ns signal when the DMAC exited from reset 1 = boot_manager_ns was HIGH	1
boot_en	[1]	Indicates the status of the boot_from_pc signal when the DMAC exited from reset 0 = boot_from_pc was LOW	0
periph_req	[0]	Supports peripheral requests 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.	1

3.2.4 Configuration Register1 for DMA_PERI(0,1) (CR1, R)

- CR1 for DMA_PERI0, R, Address = 0xE900_0E04
- CR1 for DMA_PERI1, R, Address = 0xE920_0E04

CR1	Bit	Description	Reset Value
num_i-cache_lines	[7:4]	**Read value is always 7. It means DMA_PERI(0,1) has 8 i-cache lines**	0x7
i-cache_len	[2:0]	**Read value is always 4. It means the length of an i-cache line is 16 bytes.**	0x4

3.2.5 Configuration Register2 for DMA_PERI(0,1) (CR2, R)

- CR2 for DMA_PERI0, R, Address = 0xE900_0E08
- CR2 for DMA_PERI1, R, Address = 0xE920_0E08

CR2	Bit	Description	Reset Value
boot_addr	[31:0]	Provides the value of boot_addr[31:0] when the DMAC exited from reset 32'b0	0

3.2.6 Configuration Register3 for DMA_PERI(0,1) (CR3, R)

- CR3 for DMA_PERI0, R, Address = 0xE900_0E0C
- CR3 for DMA_PERI1, R, Address = 0xE920_0E0C

CR3	Bit	Description	Reset Value
INS	[31:0]	Provides the security state of the interrupt outputs Bit [N] = 1 Assigns irq[N] to the Non-secure state 32'hffff_ffff	0xFFFF_FFFF

3.2.7 Configuration Register4 for DMA_PERI(0,1) (CR4, R)

- CR4 for DMA_PERI0, R, Address = 0xE900_0E10
- CR4 for DMA_PERI1, R, Address = 0xE920_0E10

CR4	Bit	Description	Reset Value
PNS	[31:0]	Provides the security state of the peripheral request interfaces: Bit [*N*] = 1 Assigns peripheral request interface N to the Non-secure state. 32'hffff_ffff	0xFFFF_FFFF

3.2.8 Configuration Register DN for DMA_PERI(0,1) (CRdn, R)

- CRDn for DMA_PERI0, R, Address = 0xE900_0E14
- CRDn for DMA_PERI1, R, Address = 0xE920_0E14

CRDn	Bit	Description	Reset Value
data_buffer_dep	[29:20]	The number of lines that the data buffer contains b000000111 = 8 lines	0x7
rd_q_dep	[19:16]	The depth of the read queue b0111 = 8 lines	0x7
rd_cap	[14:12]	Read issuing capability that programs the number of outstanding read transactions b011 = 4	0x3
wr_q_dep	[11:8]	The depth of the write queue b0111 =8 lines	0x7
wr_cap	[6:4]	Write issuing capability that programs the number of outstanding write transactions b011 = 4	0x3
data_width	[2:0]	The data bus width of the AXI interface b010 = 32-bit	0x2

3.2.9 Configuration Register0 for DMA_mem (CR0, R, Address=0xE810_0E00)

CR0	Bit	Description	Reset Value
num_events	[21:17]	Number of interrupt outputs that the DMAC provides b11111 = 32 interrupt outputs, irq[31:0]	0x1F
num_periph_req	[16:12]	Number of peripheral request interfaces that the DMAC provides b00001 = 2 peripheral request interfaces	0x1
num_chnls	[6:4]	Number of DMA channels that the DMAC supports b111 = 8 DMA channels	7
mgr_ns_at_rst	[2]	Indicates the status of the boot_manager_ns signal when the DMAC exited from reset 0 = boot_manager_ns was LOW	0
boot_en	[1]	Indicates the status of the boot_from_pc signal when the DMAC exited from reset 0 = boot_from_pc was LOW	0
periph_req	[0]	Supports peripheral requests 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.	1

3.2.10 Configuration Register1 for DMA_MEM (CR1, R, Address=0xE810_0E04)

CR1	Bit	Description	Reset Value
num_i-cache_lines	[7:4]	**Read value is always 7. It means DMA_MEM has 8 i-cache lines.**	0x7
i-cache_len	[2:0]	**Read value is always 5. It means the length of an i-cache line is 32 bytes.**	0x5

3.2.11 Configuration Register2 for DMA_MEM (CR2, R, Address=0xE810_0E08)

CR2	Bit	Description	Reset Value
boot_addr	[31:0]	Provides the value of boot_addr[31:0] when the DMAC exited from reset 32'b0	0

3.2.12 Configuration Register3 for DMA_MEM (CR3, R, Address=0xE810_0E0C)

CR3	Bit	Description	Reset Value
INS	[31:0]	Provides the security state of the interrupt outputs Bit [N] = 1 Assigns irq[N] to the Non-secure state 32'hffff_ffff	0xFFFF_FFFF

3.2.13 Configuration Register4 for DMA_MEM (CR4, R, Address=0xE810_0E10)

CR4	Bit	Description	Reset Value
PNS	[31:0]	Provides the security state of the peripheral request interfaces: Bit [*N*] = 1 Assigns peripheral request interface N to the Non-secure state. b11	0x3

3.2.14 Configuration Register DN for DMA_MEM (CRdn, R, Address=0xE810_0E14)

CRDn	Bit	Description	Reset Value
data_buffer_dep	[29:20]	The number of lines that the data buffer contains b000011111 = 32 lines	0x1F
rd_q_dep	[19:16]	The depth of the read queue b0111 = 8 lines	0x7
rd_cap	[14:12]	Read issuing capability that programs the number of outstanding read transactions b011 = 4	0x3
wr_q_dep	[11:8]	The depth of the write queue b0111 =8 lines	0x7
wr_cap	[6:4]	Write issuing capability that programs the number of outstanding write transactions b011 = 4	0x3
data_width	[2:0]	The data bus width of the AXI interface b011 = 64-bit	0x3

4 INSTRUCTION

Table 6.1-5 Instruction Syntax Summary

Mnemonic	Instruction	Thread usage: M = DMA manager C = DMA channel		Description
DMAADDH	Add Halfword	-	C	See DMAADDH on page 4-5 of "PL330 TRM"
DMAEND	End	M	C	See DMAEND on page 4-5 of "PL330 TRM"
DMAFLUSHP	Flush and notify Peripheral	-	C	See DMAFLUSHP on page 4-6 of "PL330 TRM"
DMAGO	Go	M	-	See DMAGO on page 4-6 of "PL330 TRM"
DMALD	Load	-	C	See DMALD[S|B] on page 4-8 of "PL330 TRM"
DMALDP	Load Peripheral	-	C	See DMALDP<S|B> on page 4-9 of "PL330 TRM"
DMALP	Loop	-	C	See DMALP on page 4-10 of "PL330 TRM"
DMALPEND	Loop End	-	C	See DMALPEND[S|B] on page 4-11 of "PL330 TRM"
DMALPFE	Loop Forever	-	C	See DMALPFE on page 4-13 of "PL330 TRM"
DMAKILL	Kill	M	C	See DMAKILL on page 4-13 of "PL330 TRM"
DMAMOV	Move	-	C	See DMAMOV on page 4-14 of "PL330 TRM"
DMANOP	No operation	M	C	See DMANOP on page 4-16 of "PL330 TRM"
DMARMB	Read Memory Barrier	-	C	See DMARMB on page 4-16 of "PL330 TRM"
DMASEV	Send Event	M	C	See DMASEV on page 4-17 of "PL330 TRM"
DMAST	Store	-	C	See DMAST[S|B] on page 4-17 of "PL330 TRM"
DMASTP	Store and notify Peripheral	-	C	See DMASTP<S|B> on page 4-19 of "PL330 TRM"
DMASTZ	Store Zero	-	C	See DMASTZ on page 4-20 of "PL330 TRM"
DMAWFE	Wait For Event	M	C	See DMAWFE on page 4-20 of "PL330 TRM"
DMAWFP	Wait For Peripheral	-	C	See DMAWFP<S|B|P> on page 4-21 of "PL330 TRM"
DMAWMB	Write Memory Barrier	-	C	See DMAWMB on page 4-22 of "PL330 TRM"

Each PL330 has a manager thread and eight channel threads. A manager thread controls overall operation of DMAC, like initiating and killing channel. The channel thread actually does DMA operation.

4.1 KEY INSTRUCTION

To run channel thread, you must write assembly code. Compared to PC assembly, it is very simple. The description of key instruction is listed below. For full instruction set refer to "Chapter 4 of PL330 TRM".

4.1.1 DMAMOV

Move instructs the DMAC to move a 32-bit immediate into Source Address REG (SAR), Destination Address REG (DAR), and Channel Control REG (CCR).

1. SAR
   A. Example: DMAMOV SAR, 0x24000000
      √ 0x2400_0000 is source address of DMA operation

2. DAR
   A. Example: DMAMOV DAR, 0x24001000
      √ 0x2400_1000 is destination address of DMA operation

3. CCR
   A. Example: DMAMOV CCR, SB2 SS32 SP0 DB2 DS32 DP0
      √ Source: burst length is 2, 32-bit data width
      √ Destination: burst length is 2, 32-bit data width
      √ SP0 and DP0 means normal and secure. SP2 and DP2 means normal and non-secure.
   B. Refer to PL330 TRM, Page.4-25~4-26 for exact DMA setting you need, like burst length, bit-width, address increment, and etc.

DMALD, DMALDP

Load instructs the DMAC to perform a DMA load, using AXI transactions that the SAR and CCR specify. For example, if you define CCR as 32-bit, burst length 2, one DMALD generates a bus transaction of 32-bit and burst length2. DMALDP notifies the peripheral that the data transfer is complete.

DMAST, DMASTP

Store instructs the DMAC to transfer data from the FIFO to the location that the DAR specifies, using AXI transactions that the DAR and CCR specify. For example, if you define CCR as 32-bit, burst length 2, one DMAST generates a bus transaction of 32-bit and burst length2. DMASTP notifies the peripheral that the data transfer is complete.

DMASTZ

Store Zero instructs the DMAC to store zeros, using AXI transactions that the DAR and CCR specify. For example, if you define CCR as 32-bit, burst length 2, one DMASTZ generates a bus transaction of 32-bit and burst length2 with zeros at data bus.

DMALP, DMALPEND

"**DMALP lc0, 4** bla~bla~ **DMALPEND lc0"** loops "bla~bla~" 4 times. There are 2 loop counters, lc0 and lc1. You can use nested loop by 2 loop counters.

DMAWFP

This is used for peripheral DMA. Wait for Peripheral instructs the DMAC to halt execution of the thread until the specified peripheral signals a DMA request for that DMA channel.

DMAFLUSHP

This is used for peripheral DMA. Flush Peripheral clears the state in DMA that describes the contents of the peripheral and sends a message to the peripheral to resend its level status. **This instruction asserts DMAACK.** If you need DMAACK at certain point, place this to that point.

DMAEND

This instructs a channel to stop.

4.2 USAGE MODEL

PL330 needs its own binary.

1. Load DMA binary into memory.

2. Use DMA debug SFRs to start DMA controller, PL330.
   A. Using debug SFRs
      √ DBGCMD, DBGINST0, and DBGINST1 (all write-only)
      √ **Before writing above three, check DBGSTATUS is not busy.**
      √ For more information refer to "PL330 TRM, Page.3-37~3-40".
   B. DBGINST0 and DBGINST1 will have debug instructions.
      √ These SFRs can receive only three instructions, DMAGO, DMASEV, and DMAKILL.
      √ DMAGO starts a channel (For more information refer to "PL330 TRM, P.3-38~3.40 and P.4-6~4.8").
   C. DBGCMD executes the instruction **stored in DBGINST0 and 1 SFRs**.

Security Scheme

DMA_mem runs on both secure & non-secure mode and DMA_peri runs only on non-secure mode.

1. Channel thread
   A. DMA_mem: both secure (ns bit at DMAGO instruction as 0) and non-secure (ns bit at DMAGO instruction as 1)
   B. DMA_peri: just non-secure (ns bit at DMAGO instruction as 1)

2. ASM code
   A. For non-secure transaction,
      √ Use **SP2** and DP2 at DMAMOV instruction.
      √ APROT[1] will be 1'b1
   B. For secure transaction,
      √ Use **SP0** and **DP0** at DMAMOV instruction.
      √ APROT[1] will be 1'b0

Interrupts

The DMAC provides the **irq** signals to use as active-high level-sensitive interrupts to external CPUs. If you program the *Interrupt Enable Register* to generate an interrupt, after the DMAC executes DMASEV it sets the corresponding **irq** HIGH.

You can clear the interrupt by writing to the *Interrupt Clear Register*.

Following are the steps to control interrupt:

1. Setup the *Interrupt Enable Register* to generate interrupts.

- The interrupt enable register is a 32-bit register. Each bit of the *INTEN Register* controls if the DMAC signals an interrupt using the corresponding **irq**.

- Programs the appropriate bit to control how the DMAC responds if it executes DMASEV:
  - √ **Bit [*N*] = 0** If executing DMASEV for event *N* then the DMAC signals event *N* to all of the threads.
  - √ **Bit [*N*] = 1** If executing DMASEV for event *N* then the DMAC sets **irq[N]** HIGH.

2. Program assembly code to set the corresponding IRQ HIGH by executing DMASEV.

- **Use DMASEV instruction to signal an interrupt using one of the IRQ outputs.**

3. Clear the interrupt by writing to the *Interrupt Clear Register*

- Each bit in the *INTCLR Register* controls the clearing of an interrupt.
- Program to controls the clearing of the **irq** outputs:
  - √ **Bit [*N*] = 0** The status of **irq[N]** does not change.
  - √ **Bit [*N*] = 1** The DMAC sets **irq[N]** LOW.

Interrupt also occurs if **DMA is at fault status**.

Summary

1. You can configure the DMAC with up to eight DMA channels, with each channel being capable of supporting a single concurrent thread of DMA operation. In addition there is a single DMA manager thread to initialize the DMA channel thread.

2. Channel thread
   A. Each channel thread does actual DMA operation. You must make assembly code representing your intention. If you need a number of independent DMA channels, you must make a number of assembly codes for each.
   B. Assemble them, link them into one file, and load it into memory.

3. Start each channel using DBGCMD, DBGINST0, and DBGINST1 SFR.

7.1 PULSE WIDTH MODULATION TIMER

1 OVERVIEW

The S5PC100 has five 32-bit timers. These timers generate internal interrupts to the ARM subsystem. In addition, Timers 0, 1, and 2 include a Pulse Width Modulation (PWM) function which drives an external I/O signal. The PWM for timer 0 has an optional dead-zone generator capability to support a large current device. Timer 3 and 4 are internal timers without output pins.

The Timers are normally clocked off of a divided version of the APB-PCLK. Timers 0 and 1 share a programmable 8-bit prescaler that provides the first level of division for the PCLK. Timer 2, 3, and 4 share a different 8-bit prescaler. Each timer has its own, private clock-divider that provides a second level of clock division (prescaler divided by 2,4,8, or 16). Alternatively, the Timers can select a clock source from an external pin. Timers 0,1, 2, 3,and 4 selects the external clock (PWM_TCLK)..

Each timer has its own 32-bit down-counter which is driven by the timer clock. The down-counter is initially loaded from the Timer Count Buffer register (TCNTBn). If the down-counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation is complete. If the timer down-counter reaches zero, the value of corresponding TCNTBn automatically reloads into the down-counter to start a next cycle. However, if the timer stops, for example, by clearing the timer enable bit of TCONn during the timer running mode, the value of TCNTBn is not reloaded into the counter.

The Pulse Width Modulation function (PWM) uses the value of the TCMPBn register. The timer control logic changes the output level if down-counter value matches the value of the compare register in timer control logic. Therefore, the compare register determines the turn-on time (or turn-off time) of a PWM output.

The TCNTBn and TCMPBn registers are double buffered to allow the timer parameters to be updated in the middle of a cycle. The new values do not take effect until the current timer cycle completes.

The Figure 7.1-1 shows a simple example of a PWM cycle.

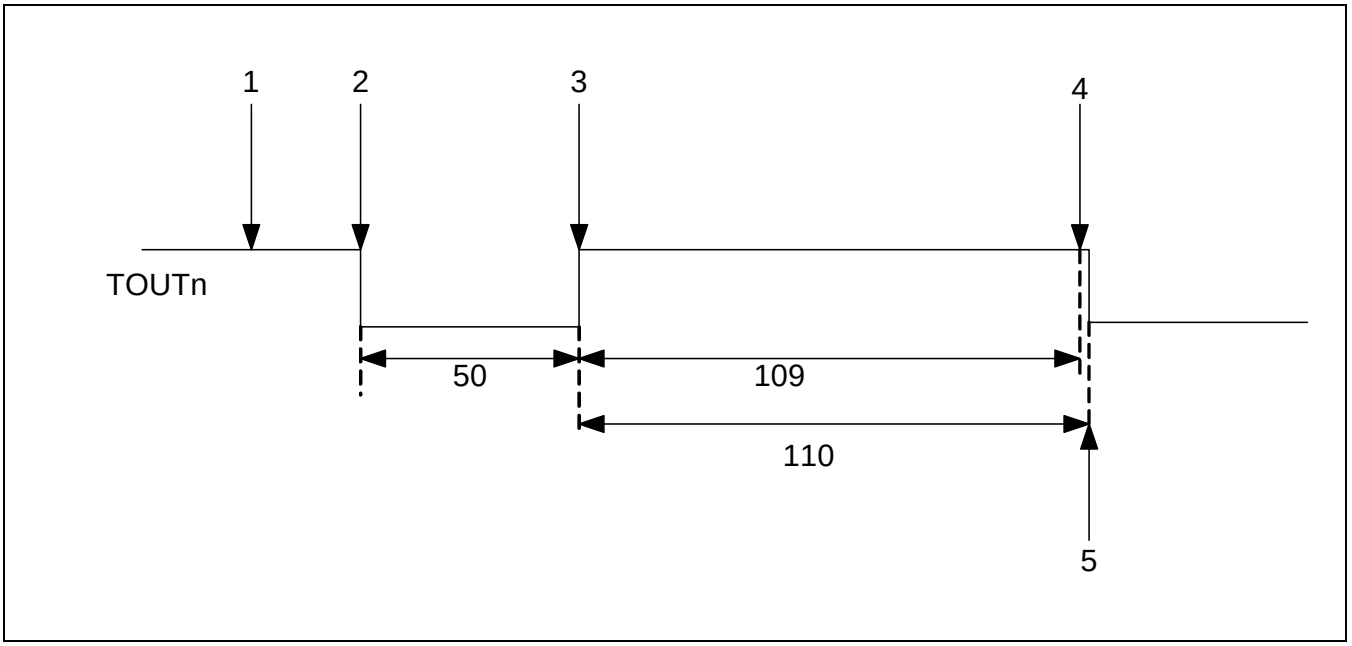


Figure 7.1-1 Simple Example of PWM Cycle

1. Initialize TCNTBn with 159(50+109) and TCMPBn with 109.
2. Start Timer: Set the start bit and manual update bit to off.
   The TCNTBn value of 159 is loaded into the down-counter, the output is driven low.
3. If down-counter counts down to value in the TCMPBn register 109, the output changes from low to high
4. If the down-counter reaches 0, it generates interrupt request.
5. The down-counter automatically reloads with TCNTBn. This restarts the cycle.

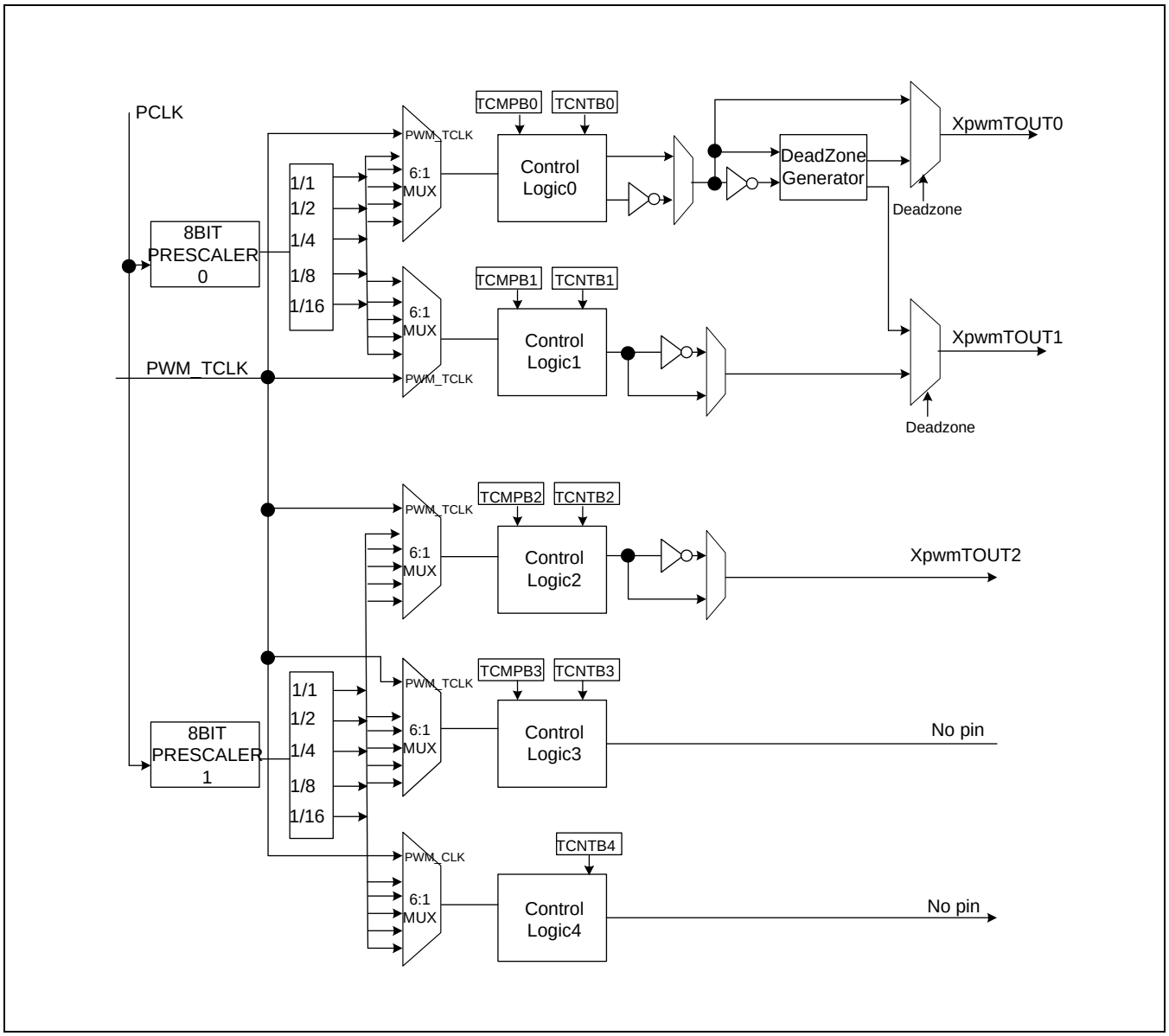


Figure 7.1-2 PWM TIMER Clock Tree Diagram

The Figure 7.1-2 shows the clock generation scheme for individual PWM Channels.

Each timers can generate level interrupts.

2 FEATURES

The Features supported by the PWM include:

- Five 32-bit Timers.
- Two 8-bit Clock Prescalers providing first level of division for the PCLK, Five Clock Dividers and Multiplexers providing second level of division for the Prescaler clock and one External Clock
- Programmable Clock Select Logic for individual PWM Channels.
- Four Independent PWM Channels with Programmable Duty Control and Polarity.
- Static Configuration: PWM is stopped
- Dynamic Configuration: PWM is running.
- Auto-Reload Mode and One-Shot Pulse Mode.
- One external input to start PWM.
- Dead Zone Generator on two PWM Outputs.
- Level Interrupt Generation.

The PWM has two operation modes:

- Auto-Reload Mode
  Continuous PWM pulses are generated based on programmed duty cycle and polarity.
- One-Shot Pulse Mode
  Only one PWM pulse is generated based on programmed duty cycle and polarity.

To control the functionality of PWM, 18 special function registers are provided. The PWM is a programmable output, dual clock input AMBA slave module and connects to the Advanced Peripheral Bus (APB). The 18 special function registers within PWM are accessed via APB transactions.

3 PWM OPERATION

3.1 PRESCALER & DIVIDER

An 8-bit prescaler and 3-bit divider makes the following output frequencies:

Table 7.1-1 Minimum and Maximum Resolution based on Prescaler and Clock Divider Values

4-bit Divider Settings	Minimum Resolution (prescaler value=1)	Maximum Resolution (prescaler value=255)	Maximum Interval (TCNTBn=4294967295)
1/1(PCLK=66MHz)	0.030us(33.0MHz)	3.879us(257.8KHz)	16659.27s
1/2 (PCLK=66MHz)	0.061us (16.5MHz)	7.758us (128.9KHz)	33318.53s
1/4 (PCLK=66MHz)	0.121us (8.25MHz)	15.515us (64.5KHz)	66637.07s
1/8 (PCLK=66MHz)	0.242us (4.13MHz)	31.03us (32.2KHz)	133274.14s
1/16 (PCLK=66MHz)	0.485us (2.06MHz)	62.061us (16.1KHz)	266548.27s

3.2 BASIC TIMER OPERATION

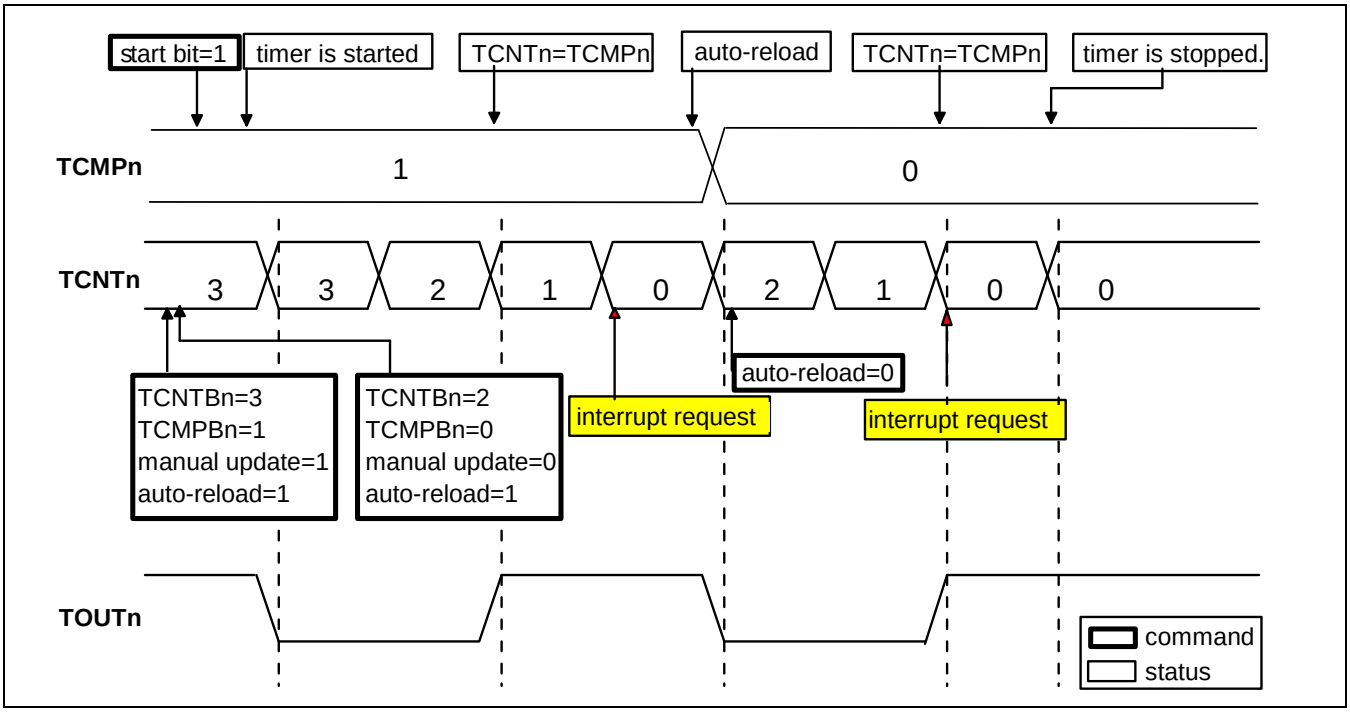


Figure 7.1-3 Timer Operations

A timer (except the timer channel 4) has TCNTBn, TCNTn, TCMPBn and TCMPn registers. TCNTBn and TCMPBn are loaded into TCNTn and TCMPn if the timer reaches 0. If TCNTn reaches 0, the interrupt request occurs if the interrupt is enabled (TCNTn and TCMPn are the names of the internal registers. The TCNTn register is read from the TCNTOn register).

If you want to generate interrupt at intervals 3cycle of XpwmTOUTn, set TCNTBn, TCMPBn and TCON register as shown in Figure 7.1-3.

Steps to generate interrupt:

1. Set TCNTBn=3 and TCMPBn=1.
2. Set auto-reload=1 and manual update=1.
   If manual update bit is 1, TCNTBn and TCMPBn value are loaded to TCNTn and TCMPn.
3. Set TCNTBn=2 and TCMPBn=0 for next operation.
4. Set auto-reload=1 and manual update=0.
   If you set manual update=1 at this time, TCNTn is changed to 2 and TCMP is changed to 0.
   Therefore, interrupt is generated at interval 2cycle instead of 3cycle.
   You must set auto-reload=1 automatically for next operation.
5. Set start = 1 for operation start and then TCNTn is down counting.
   If TCNTn is 0, interrupt is generated and if auto-reload is enable, TCNTn is loaded 2 (TCNTBn value) and TCMPn is loaded 0 (TCMPn value).
6. Before stop, TCNTn is down counting.

3.3 AUTO-RELOAD AND DOUBLE BUFFERING

The Timers have a double buffering feature, which changes the reload value for the next timer operation without stopping the current timer operation. Even though the new timer value is set, a current timer operation completes successfully.

The timer value is written into TCNTBn (Timer Count Buffer register) and the current counter value of the timer is read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value is not the current state of the counter but the reload value for the next timer duration.

The auto-reload is the operation, which copies the TCNTBn into TCNTn if TCNTn reaches 0. The value, written into TCNTBn, is loaded to TCNTn if the TCNTn reaches to 0 and auto-reload is enabled. If the TCNTn is 0 and the auto-reload bit is 0, the TCNTn does not operate further.

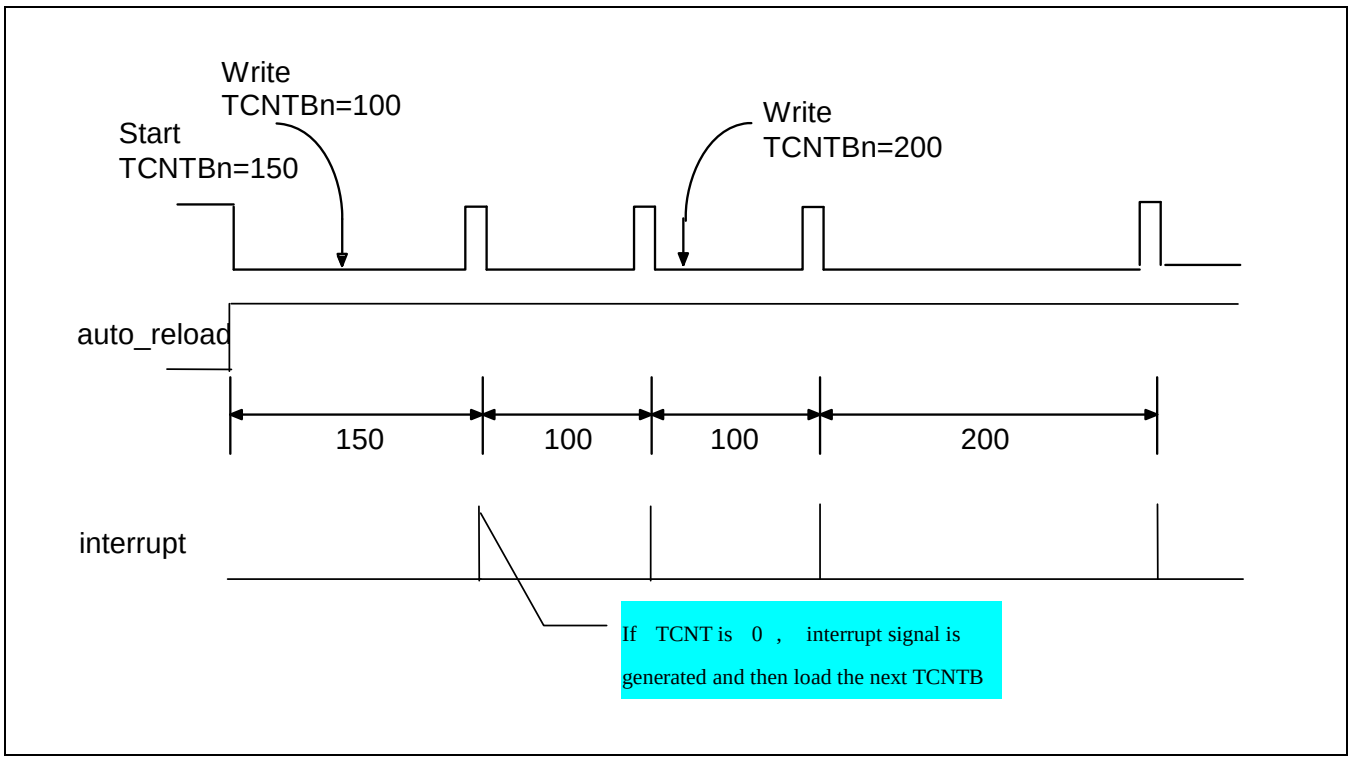


Figure 7.1-4 Example of Double Buffering Feature

3.4 TIMER OPERATION EXAMPLE

The result of the following procedure is shown in Figure 7.1-5.

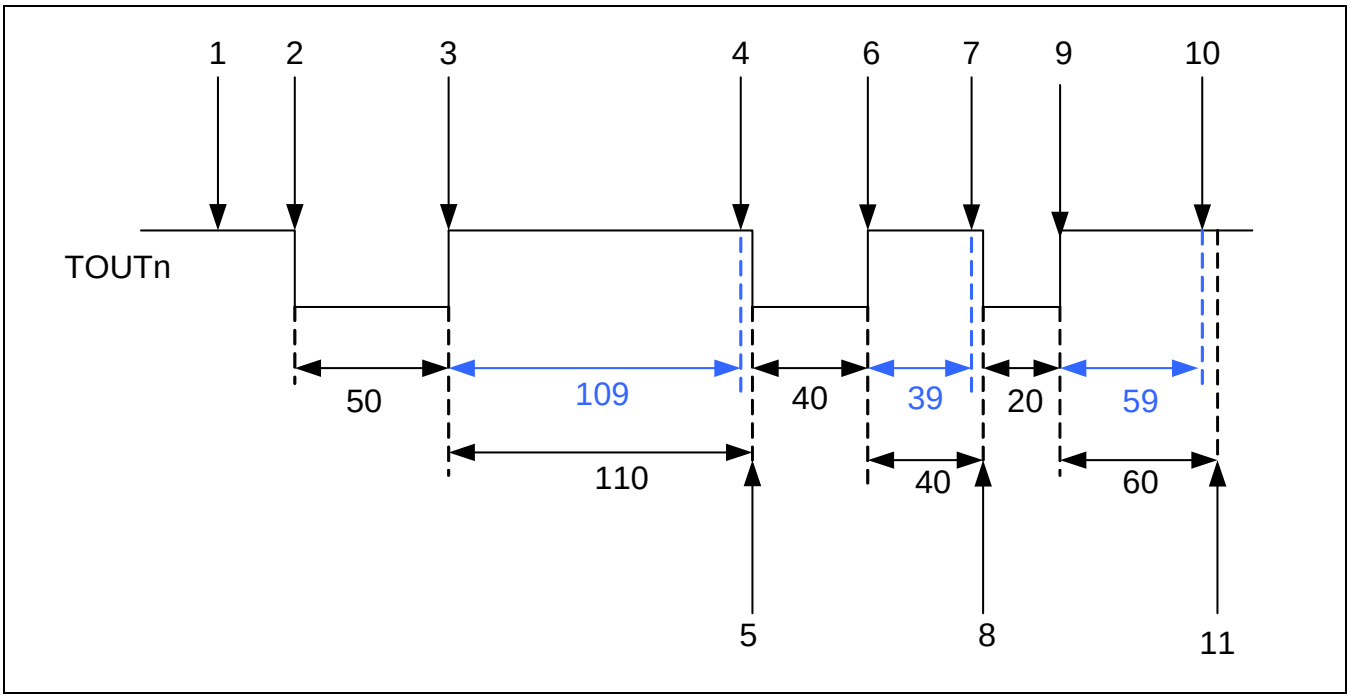


Figure 7.1-5 Example of a Timer Operation

1. Enable the auto-reload feature. Set the TCNTBn as 159(50+109) and TCMPBn as 109. Set the manual update bit on and set the manual update bit off. Set the inverter on/ off bit. The manual update bit sets TCNTn and TCMPn to the value of TCNTBn and TCMPBn.

   Set TCNTBn and TCMPBn as 79(40+39) and 39.
2. Start Timer: Set the start bit in TCON
3. If TCNTn and TCMPn has same value, the logic level of TOUTn is changed from low to high
4. As soon as TCNTn reaches to 0, it generates interrupt request.
5. TCNTn and TCMPn are reloaded automatically with TCNTBn and TCMPBn as (79(40+39)) and 39. In the ISR (Interrupt Service Routine), the TCNTBn and TCMPBn are set as 79(20+59) and 59.
6. If TCNTn and TCMPn has same value, the logic level of TOUTn is changed from low to high
7. As soon as TCNTn reaches to 0, it generates interrupt request.
8. TCNTn and TCMPn are reloaded automatically with TCNTBn,TCMPBn as (79(20+59)) and 59. In the ISR (Interrupt Service Routine), auto-reload and interrupt request are disabled to stop the timer.
9. If TCNTn and TCMPn has same value, the logic level of TOUTn is changed from low to high
10. Even if TCNTn reaches to 0, No interrupt request is generated.
11. TCNTn is not reloaded and the timer is stopped because auto-reload is disabled.

3.5 INITIALIZE TIMER (SETTING MANUAL-UP DATA AND INVERTER)

Because an auto-reload operation of the timer occurs when the down counter reaches to 0, a starting value of the TCNTn has to be defined by the user at first. In this case, the starting value must be loaded by manual update bit. The sequence of starting a timer is as follows:

1. Write the initial value into TCNTBn and TCMPBn.
2. Set the manual update bit and clear only manual update bit of the corresponding timer.
   Note: We recommend you to set the inverter on/off bit (whether inverter is used or not).
3. Set the start bit of corresponding timer to start the timer

3.6 PWM (PULSE WIDTH MODULATION)

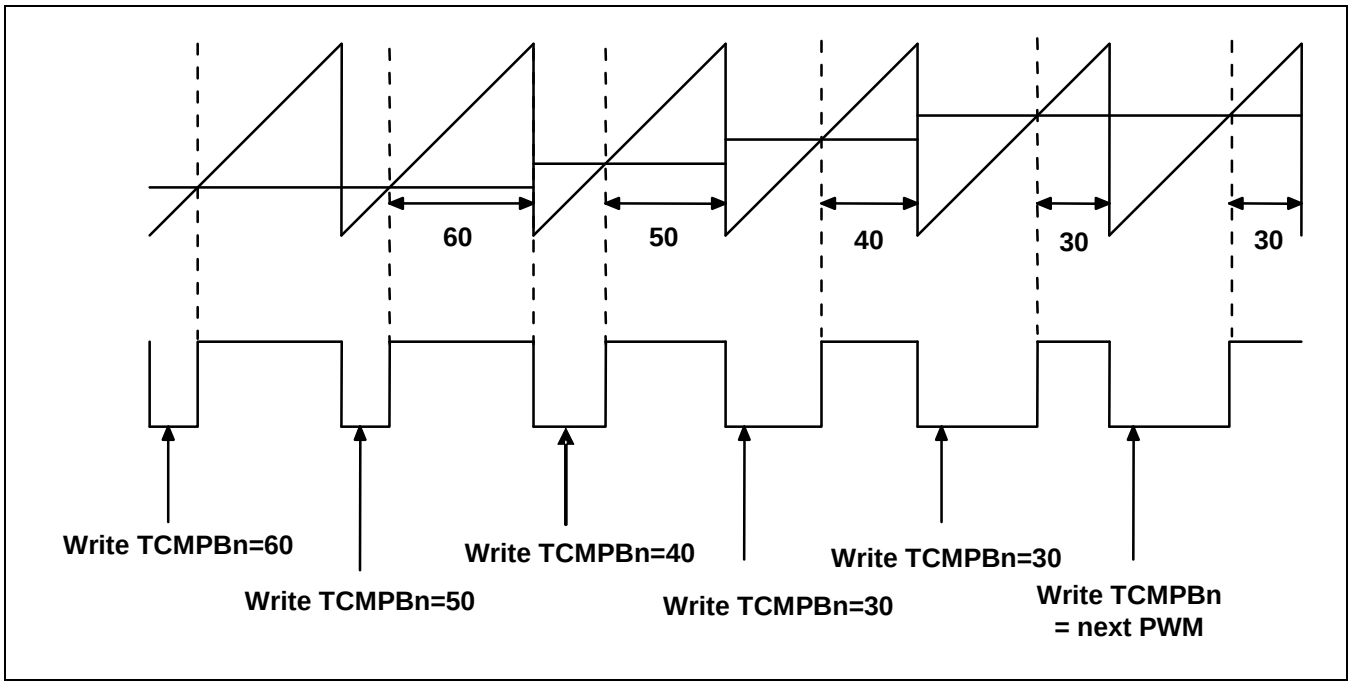


Figure 7.1-6 Example of PWM

Use TCMPBn to implement PWM feature. PWM frequency is determined by TCNTBn. A PWM value is determined by TCMPBn as shown in the Figure 7.1-6.

For higher PWM value, decrease TCMPBn value. For lower PWM value, increase TCMPBn value. If output inverter is enabled, the increment/ decrement may be opposite.

Because of double buffering feature, TCMPBn, for a next PWM cycle, is written by ISR at any point of current PWM cycle.

3.7 OUTPUT LEVEL CONTROL

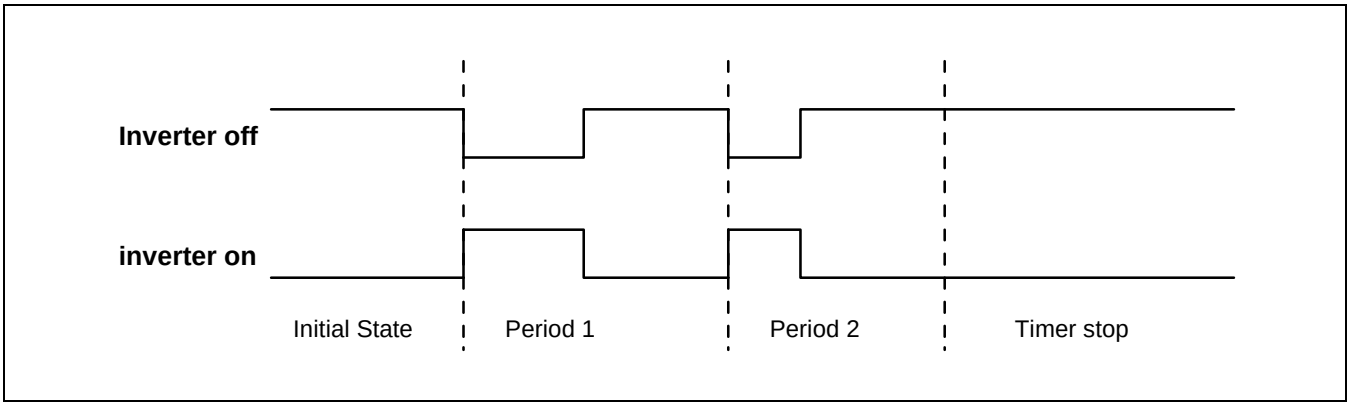


Figure 7.1-7 Inverter On/Off

The steps to maintain TOUT as high or low (Assume that inverter is off).

1. Turn off the auto-reload bit. Then, TOUTn goes to high level and the timer is stopped after TCNTn reaches to 0. This method is recommended.
2. Stop the timer by clearing the timer start/ stop bit to 0. If TCNTn <= TCMPn, the ouput level is *high*. If TCNTn >TCMPn, the output level is *low*
3. TOUTn is inverted by the inverter on/ off bit in TCON. The inverter removes the additional circuit to adjust the output level.

3.8 DEAD ZONE GENERATOR

The deadzone is for the PWM control of power devices. This feature is used to insert the time gap between a turn-off of a switching device and a turn on of the other switching device. This time gap prohibits the two switching device turning on simultaneously even for a very short time.

TOUT_0 is the PWM output. nTOUT_0 is the inversion of the TOUT_0. If the dead-zone is enabled, the output wave-form of TOUT_0 and nTOUT_0 is TOUT_0_DZ and nTOUT_0_DZ. TOUT0_DZ and nTOUT_0_DZ cannot be turned on simultaneously by the dead zone interval. For functional correctness, the deadzone length must be set smaller than compare counter value.

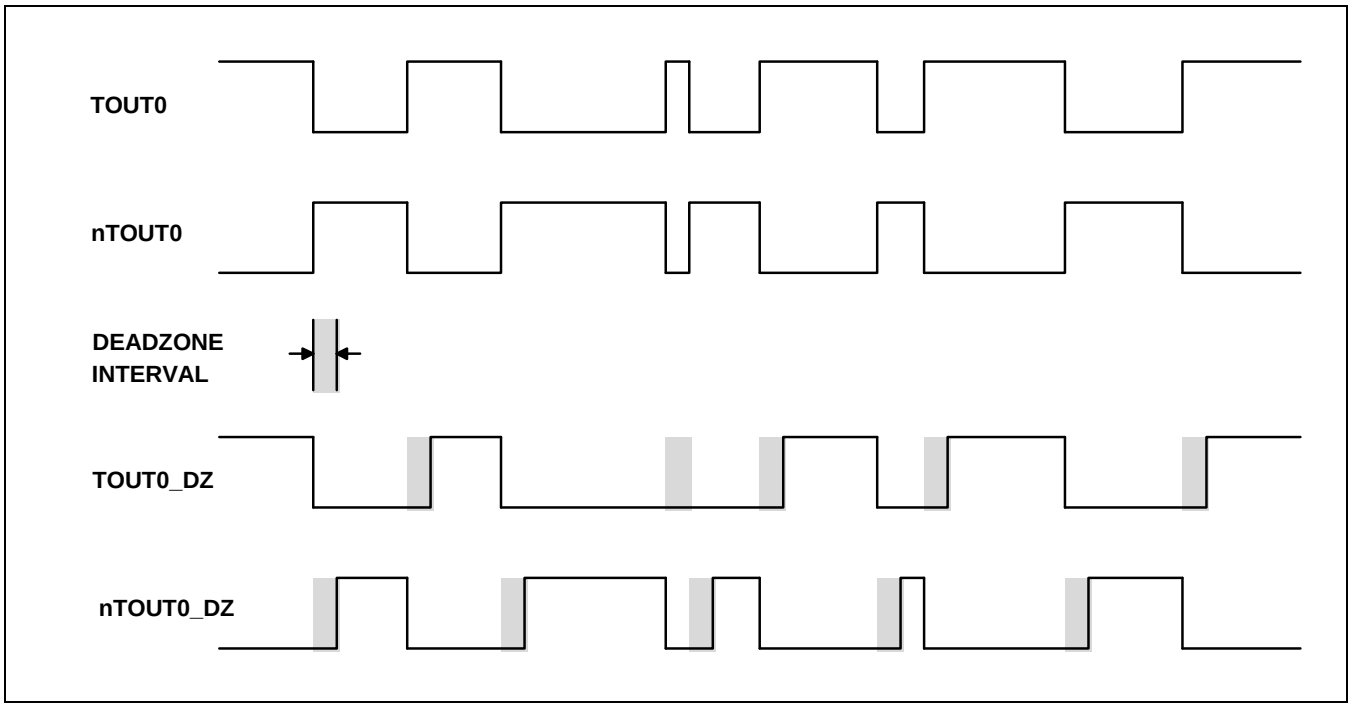


Figure 7.1-8 The Waveform when a Dead Zone Feature is Enabled.

4 I/O DESCRIPTION

Funtion Signal	I/O	Descrioption	Pad	Type
TOUT0	Output	PWMTIMER TOUT[0]	XpwmTOUT[0]	muxed
TOUT1	Output	PWMTIMER TOUT[1]	XpwmTOUT[1]	muxed
TOUT2	Output	PWMTIMER TOUT[2]	XpwmTOUT[2]	muxed
PWM_TCLK	Input	PWMTIMER External Clock	XpwmTOUT[0]	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

5 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
TCFG0	0xEA00_0000	R/W	Timer Configuration Register 0 that configures the two 8-bit Prescaler and DeadZone Length	0x0000_0101
TCFG1	0xEA00_0004	R/W	Timer Configuration Register 1 that controls 5 MUX Select Bit	0x0000_0000
TCON	0xEA00_0008	R/W	Timer Control Register	0x0000_0000
TCNTB0	0xEA00_000C	R/W	Timer 0 Count Buffer Register	0x0000_0000
TCMPB0	0xEA00_0010	R/W	Timer 0 Compare Buffer Register	0x0000_0000
TCNTO0	0xEA00_0014	R	Timer 0 Count Observation Register	0x0000_0000
TCNTB1	0xEA00_0018	R/W	Timer 1 Count Buffer Register	0x0000_0000
TCMPB1	0xEA00_001C	R/W	Timer 1 Compare Buffer Register	0x0000_0000
TCNTO1	0xEA00_0020	R	Timer 1 Count Observation Register	0x0000_0000
TCNTB2	0xEA00_0024	R/W	Timer 2 Count Buffer Register	0x0000_0000
TCMPB2	0xEA00_0028	R/W	Timer 2 Compare Buffer Register	0x0000_0000
TCNTO2	0xEA00_002C	R	Timer 2 Count Observation Register	0x0000_0000
TCNTB3	0xEA00_0030	R/W	Timer 3 Count Buffer Register	0x0000_0000
TCNTO3	0xEA00_0038	R	Timer 3 Count Observation Register	0x0000_0000
TCNTB4	0xEA00_003C	R/W	Timer 4 Count Buffer Register	0x0000_0000
TCNTO4	0xEA00_0040	R	Timer 4 Count Observation Register	0x0000_0000
TINT_CSTAT	0xEA00_0044	R/W	Timer Interrupt Control and Status Register	0x0000_0000

5.1 TIMER CONFIGURATION REGISTER (TCFG0, R/W, ADDRESS = 0XEA00_0000)

Timer Input Clock Frequency = PCLK / ({prescaler value + 1}) / {divider value}
{prescaler value} = 1~255
{divider value} = 1, 2, 4, 8, 16, TCLK
Dead zone length = 0~254

TCFG0	Bit	Description	Reset Value
Reserved	[31:24]	Reserved Bits	0x00
Dead zone length	[23:16]	Dead zone length	0x00
Prescaler 1	[15:8]	Prescaler 1 value for Timer 2, 3 and 4	0x01
Prescaler 0	[7:0]	Prescaler 0 value for timer 0 & 1	0x01

NOTE: If Dead Zone Length is set 'n', Real Dead Zone Length is 'n+1' (n=0~254).

5.2 TIMER CONFIGURATION REGISTER (TCFG1, R/W, ADDRESS = 0XEA00_0004)

TCFG1	Bit	Description	Reset Value
Reserved	[31:24]	Reserved Bits	0x00
Divider MUX4	[19:16]	Selects Mux input for PWM Timer 4 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 0101 = PWM_TCLK	0x00
Divider MUX3	[15:12]	Selects Mux input for PWM Timer 3 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 0101 = PWM_TCLK	0x00
Divider MUX2	[11:8]	Selects Mux input for PWM Timer 2 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 0101 = PWM_TCLK	0x00
Divider MUX1	[7:4]	Selects Mux input for PWM Timer 1 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 0101 = PWM_TCLK	0x00
Divider MUX0	[3:0]	Selects Mux input for PWM Timer 0 0000 = 1/1 0001 = 1/2 0010 = 1/4 0011 = 1/8 0100 = 1/16 0101 = PWM_TCLK	0x00

NOTE: If you use PWM_TCLK, duty of TOUT may show slight error. PWM_TCLK is sampled by PCLK in PWM module.
But PWM_TCLK and PCLK is asynchronous clock. Therefore PWM_TCLK is not sampled at exact time. This slight error can be reduced if PWM_TCLK is slower than PCLK. Therefore we recommend to use PWM_TCLK under 1MHz.
(For Example: If PCLK is 66MHz and PWM_TCLK is 1MHz, duty error is 1.5%. If PCLK is 66MHz and PWM_TCLK is 0.5MHz, duty error is 0.75%)

5.3 TIMER CONTROL REGISTER (CON, R/W, ADDRESS = 0XEA00_0008)

TCON	Bit	Description	Reset Value
Reserved	[31:23]	Reserved Bits	0x000
Timer 4 Auto Reload on/off	[22]	0 = One-Shot 1 = Interval Mode(Auto-Reload)	0x0
Timer 4 Manual Update	[21]	0 = No Operation 1 = Update TCNTB4	0x0
Timer 4 Start/Stop	[20]	0 = Stop 1 = Start Timer 4	0x0
Timer 3 Auto Reload on/off	[19]	0 = One-Shot 1 = Interval Mode(Auto-Reload)	0x0
Reserved	[18]	Reserved Bit	0x0
Timer 3 Manual Update	[17]	0 = No Operation 1 =Update TCNTB3	0x0
Timer 3 Start/Stop	[16]	0 = Stop 1 = Start Timer 3	0x0
Timer 2 Auto Reload on/off	[15]	0 = One-Shot 1 = Interval Mode(Auto-Reload)	0x0
Timer 2 Output Inverter on/off	[14]	0 = Inverter Off 1 = TOUT_2 Inverter-On	0x0
Timer 2 Manual Update	[13]	0 = No Operation 1 = Update TCNTB2,TCMPB2	0x0
Timer 2 Start/Stop	[12]	0 = Stop 1 = Start Timer 2	0x0
Timer 1 Auto Reload on/off	[11]	0 = One-Shot 1 = Interval Mode(Auto-Reload)	0x0
Timer 1 Output Inverter on/off	[10]	0 = Inverter Off 1 = TOUT_1 Inverter-On	0x0
Timer 1 Manual Update	[9]	0 = No Operation 1 = Update TCNTB1,TCMPB1	0x0
Timer 1 Start/Stop	[8]	0 = Stop 1 = Start Timer 1	0x0
Reserved	[7:5]	Reserved Bits	0x0
Dead Zone Enable/Disable	[4]	Dead Zone Generator Enable/Disable	0x0
Timer 0 Auto Reload on/off	[3]	0 = One-Shot 1 = Interval Mode(Auto-Reload)	0x0
Timer 0 Output Inverter on/off	[2]	0 = Inverter Off 1 = TOUT_0 Inverter-On	0x0
Timer 0 Manual Update	[1]	0 = No Operation 1 = Update TCNTB0,TCMPB0	0x0
Timer 0 Start/Stop	[0]	0 = Stop 1 = Start Timer 0	0x0

NOTE: Manual update bit must be 1'b0 before Start/Stop bit is 1'b1.

If Manual update bit is 1'b1 and Start/Stop bit is 1'b1, timer counter is not update by new value.

Timer counter value is last value.

5.4 TIMER0 COUNTER REGISTER (TCNTB0, R/W, ADDRESS = 0XEA00_000C)

TCNTB0	Bit	Description	Reset Value
Timer 0 Count Buffer	[31:0]	Timer 0 Count Buffer Register	0x0000_0000

5.5 TIMER0 COMPARE REGISTER (TCMPB0, R/W, ADDRESS = 0XEA00_0010)

TCMPB0	Bit	Description	Reset Value
Timer 0 Compare Buffer	[31:0]	Timer 0 Compare Buffer Register	0x0000_0000

5.6 TIMER0 OBSERVATION REGISTER (TCNTO0, R, ADDRESS = 0XEA00_0014)

TCNTO0	Bit	Description	Reset Value
Timer 0 Count Observation	[31:0]	Timer 0 Count Observation Register	0x0000_0000

NOTE: Counter observation time is dependent on PWM timer clock and PCLK.
When you observe counter value after PWM timer start, wait for delay time(Timer Clock period/2 + 2*PCLK period).

Example) if Prescaler is 4 and Divider mux is 1/2,

Timer clock period = (4+1)*2*PCLK

TCNTO observation delay time = (4+1)*2/2*PCLK+2*PCLK

5.7 TIMER1 COUNTER REGISTER (TCNTB1, R/W, ADDRESS = 0XEA00_0018)

TCNTB1	Bit	Description	Reset Value
Timer 1 Count Buffer	[31:0]	Timer 1 Count Buffer Register	0x0000_0000

5.8 TIMER1 COMPARE REGISTER (TCMPB1, R/W, ADDRESS = 0XEA00_001C)

TCMPB1	Bit	Description	Reset Value
Timer 1 Compare Buffer	[31:0]	Timer 1 Compare Buffer Register	0x0000_0000

5.9 TIMER1 OBSERVATION REGISTER (TCNTO1, R, ADDRESS = 0XEA00_0020)

TCNTO1	Bit	Description	Reset Value
Timer 1 Count Observation	[31:0]	Timer 1 Count Observation Register	0x0000_0000

NOTE: Counter observation time is dependent on PWM timer clock and PCLK.
When you observe counter value after PWM timer start, wait for delay time(Timer Clock period/2 + 2*PCLK period).

5.10 TIMER2 COUNTER REGISTER (TCNTB2, R/W, ADDRESS = 0XEA00_0024)

TCNTB2	Bit	Description	Reset Value
Timer 2 Count Buffer	[31:0]	Timer 2 Count Buffer Register	0x0000_0000

5.11 TIMER2 COMPARE REGISTER (TCMPB2, R/W, ADDRESS = 0XEA00_0028)

TCMPB2	Bit	Description	Reset Value
Timer 2 Compare Buffer	[31:0]	Timer 2 Compare Buffer Register	0x0000_0000

5.12 TIMER2 OBSERVATION REGISTER (TCNTO2, R, ADDRESS = 0XEA00_002C)

TCNTO2	Bit	Description	Reset Value
Timer 2 Count Observation	[31:0]	Timer 2 Count Observation Register	0x0000_0000

NOTE: Counter observation time is dependent on PWM timer clock and PCLK.
When you observe counter value after PWM timer start, wait for delay time(Timer Clock period/2 + 2*PCLK period).

5.13 TIMER3 COUNTER REGISTER (TCNTB3, R/W, ADDRESS = 0XEA00_0030)

TCNTB3	Bit	Description	Reset Value
Timer 3 Count Buffer	[31:0]	Timer 3 Count Buffer Register	0x0000_0000

5.14 TIMER3 OBSERVATION REGISTER (TCNTO3, R, ADDRESS = 0XEA00_0038)

TCNTO3	Bit	Description	Reset Value
Timer 3 Count Observation	[31:0]	Timer 3 Count Observation Register	0x0000_0000

NOTE: Counter observation time is dependent on PWM timer clock and PCLK.
When you observe counter value after PWM timer start, wait for delay time(Timer Clock period/2 + 2*PCLK period).

5.15 TIMER4 COUNTER REGISTER (TCNTB4, R/W, ADDRESS = 0XEA00_003C)

TCNTB4	Bit	Description	Reset Value
Timer 4 Count Buffer	[31:0]	Timer 4 Count Buffer Register	0x0000_0000

5.16 TIMER4 OBSERVATION REGISTER (TCNTO4, R, ADDRESS = 0XEA00_0040)

TCNTO4	Bit	Description	Reset Value
Timer 4 Count Observation	[31:0]	Timer 4 Count Observation Register	0x0000_0000

NOTE: Counter observation time is dependent on PWM timer clock and PCLK.
When you observe counter value after PWM timer start, wait for delay time(Timer Clock period/2 + 2*PCLK period).

5.17 INTERRUPT CONTROL AND STATUS REGISTER (TINT_CSTAT, R/W, ADDRESS = 0XEA00_0044)

TINT_CSTAT	Bit	Description	Reset Value
Reserved	[31:10]	Reserved Bits	0x00000
Timer 4 Interrupt Status	[9]	Timer 4 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 3 Interrupt Status	[8]	Timer 3 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 2 Interrupt Status	[7]	Timer 2 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 1 Interrupt Status	[6]	Timer 1 Interrupt Status Bit. Clears by writing `1' on this bit.	0x0
Timer 0 Interrupt Status	[5]	Timer 0 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 4 interrupt Enable	[4]	Enables Timer 4 Interrupt. 1 = Enabled 0 = Disabled	0x0
Timer 3 interrupt Enable	[3]	Enables Timer 3 Interrupt. 1 = Enabled 0 = Disabled	0x0
Timer 2 interrupt Enable	[2]	Enables Timer 2 Interrupt. 1 = Enabled 0 = Disabled	0x0
Timer 1 interrupt Enable	[1]	Enables Timer 1 Interrupt. 1 = Enabled 0 = Disabled	0x0
Timer 0 interrupt Enable	[0]	Enables Timer 0 Interrupt. 1 = Enabled 0 = Disabled	0x0

7.2 SYSTEM TIMER

1 OVERVIEW

System timer provides two distinctive features. First one is accurate timer which provides 1ms time tick at any power mode except sleep mode. Second one is changeable interrupt interval without stopping reference tick timer.

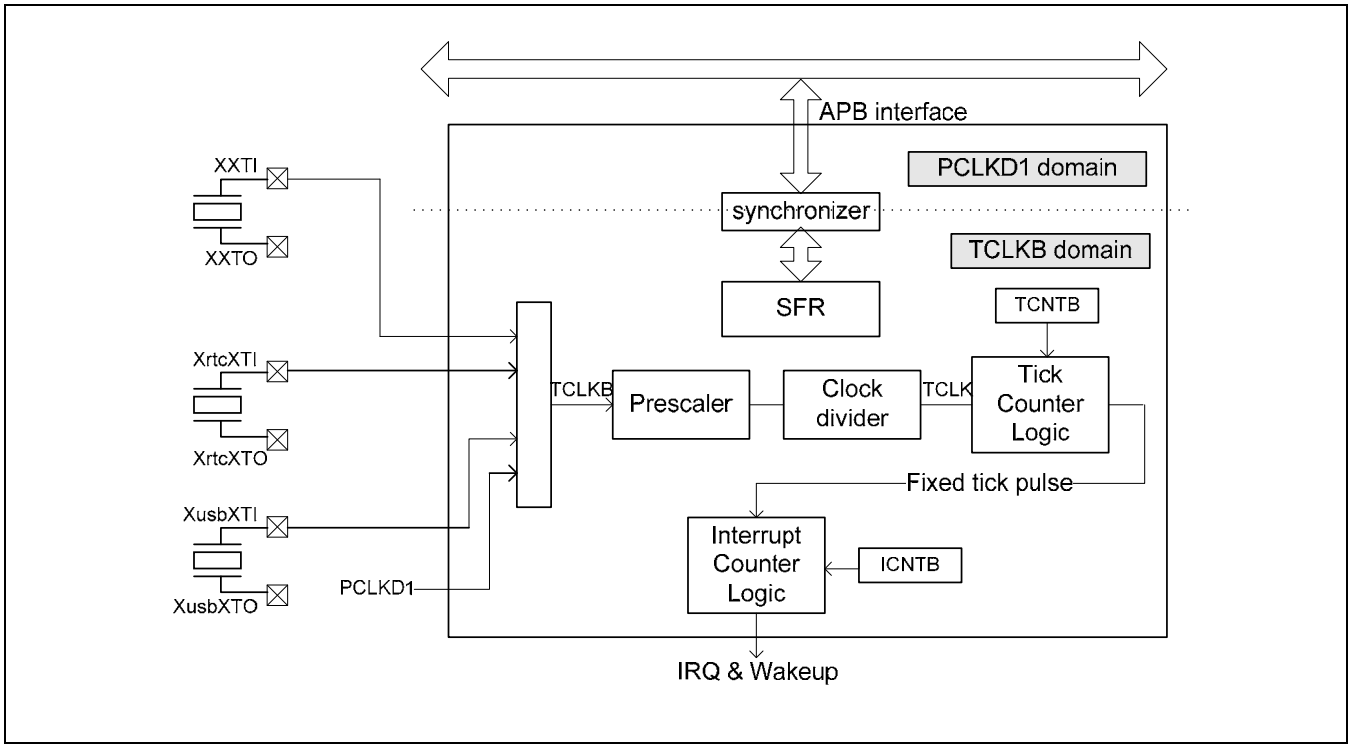


Figure 7.2-1 Overall System Timer Block Diagram

2 FEATURES

- Clock sources of ST: main OSC (XXTI), RTC OSC (XrtcXTI), USB OSC (XusbXTI), and PCLKD1
- Counter bit: 32-bit
- Changeable interrupt-interval without stopping reference tick timer
- Used at any power mode except sleep mode.

3 MEMORY MAP

Register	Address	R/W	Description	Reset Value
TCFG	EA10_0000	R/W	Configures 8-bit-Prescaler and Clock MUX	0x0000_0000
TCON	EA10_0004	R/W	Timer Control Register	0x0000_0000
TCNTB	EA10_0008	R/W	Tick Count Buffer Register	0x0000_0000
TCNTO	EA10_000C	R	Tick Count Observation Register	0x0000_0000
ICNTB	EA10_0010	R/W	Interrupt Count Buffer Register	0x0000_0000
ICNTO	EA10_0014	R	Interrupt Count Observation Register	0x0000_0000
INT_CSTAT	EA10_0018	R/W	Interrupt Control and Status Register	0x0000_0000

4 INTERNAL FUNCTION

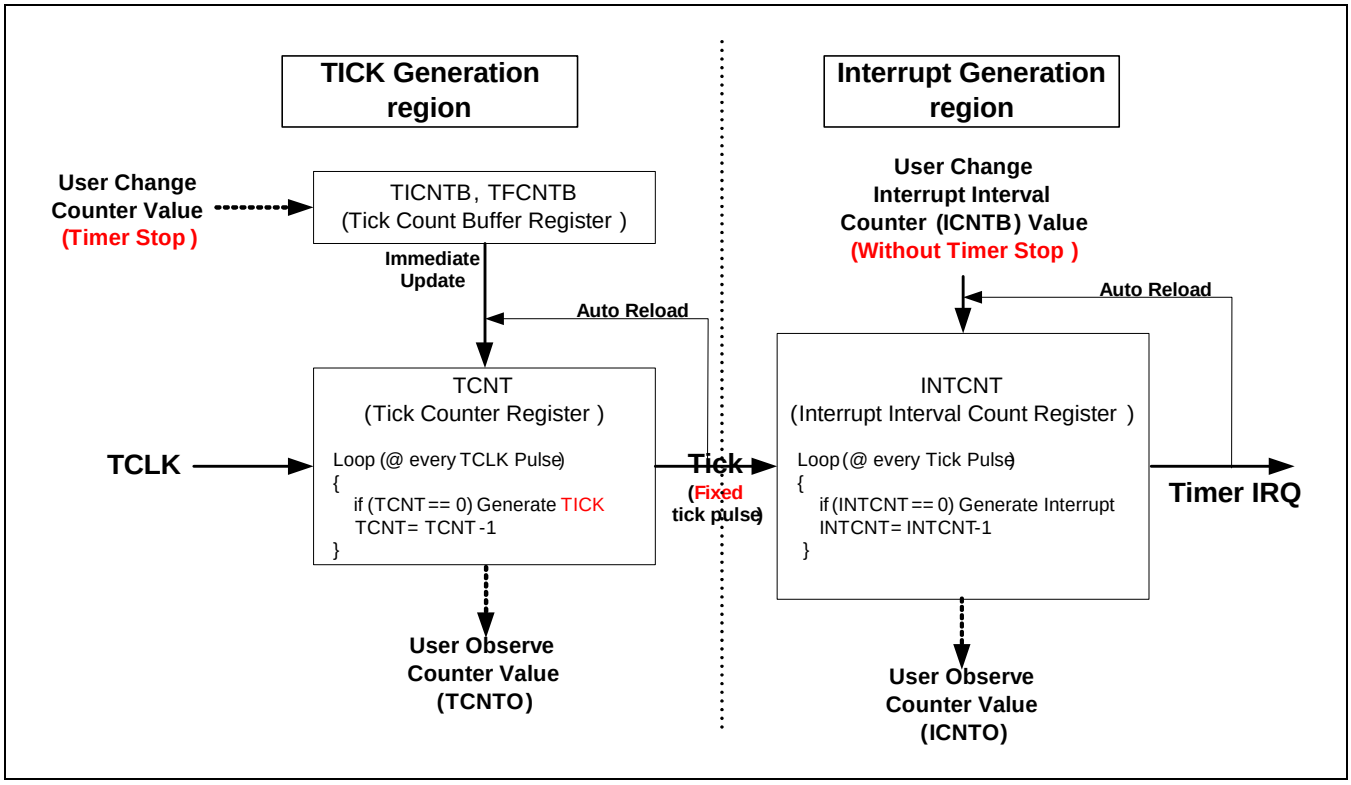


Figure 7.2-2 Two Separate Timers

There are two separate timers (One for tick generation and the other for interrupt generation) and there are two independent SFR sets and logic blocks for each (tick and interrupt) region. Each logic block operates separately. Therefore you can change interrupt interval independent to reference tick generation. This is very useful feature for some power-saving mode.

5 DETAILED OPEARATION

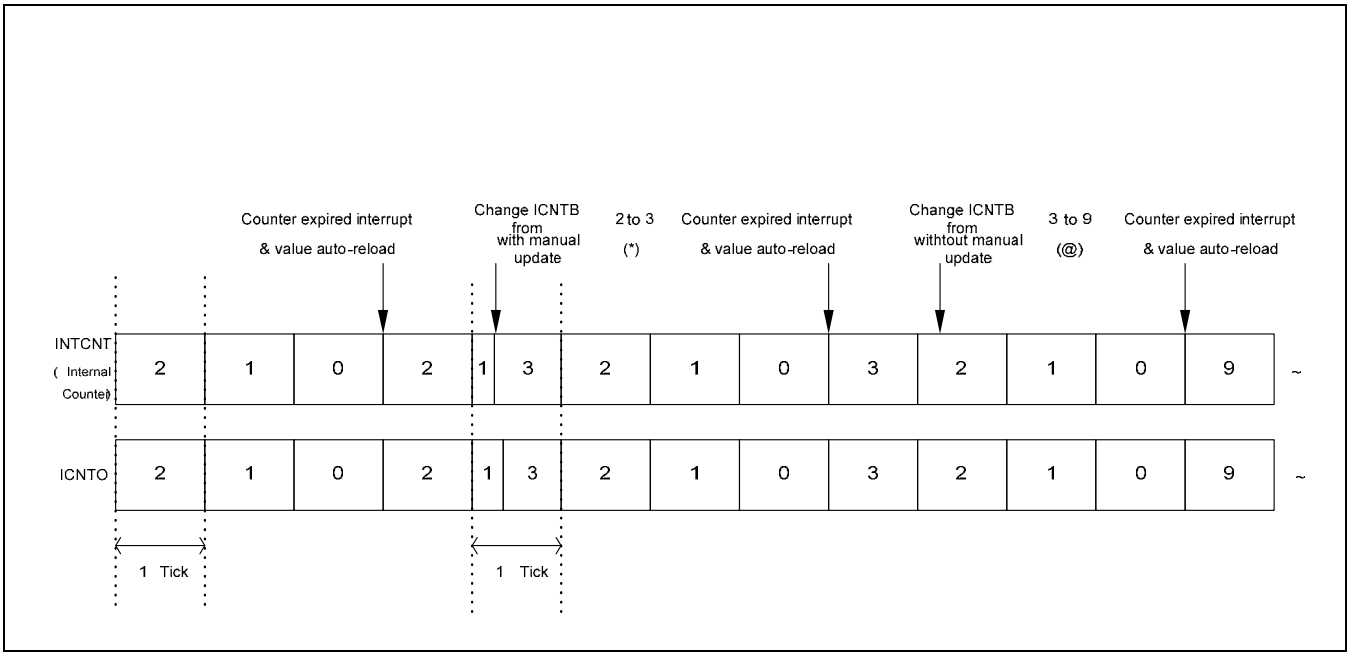


Figure 7.2-3 Timer Operation with Always on of Auto-reload

Generally tick interval is fixed after initial setting. And interrupt interval can be changed at run-time. Figure 7.2-3 shows detailed operation of interrupt counter and interrupt counter observation SFR with auto-reload. Each rectangular shows one tick time. Although interrupt interval is changed, that tick time is fixed. Because tick and interrupt counter are independent to each other.
Interrupt is asserted at INTCNT value is expired (INTCNT=0). SW can know elapsed time by reading ICNTO.

NOTE: When ICNTB is changed with interrupt manual update (TCON[4]) (* at Figure 7.2-3), the new changed value is applied to INTCNT (interrupt counter) at that time. When ICNTB is changed without interrupt manual update (TCON[4]) (@ at Figure 7.2-3), the new changed value is applied to INTCNT (interrupt counter) after counter is expired.
If auto reload is needed, Interrupt Manual Update (TCON[4]) must be de-asserted, before INTCNT value is expired (INTCNT=0).

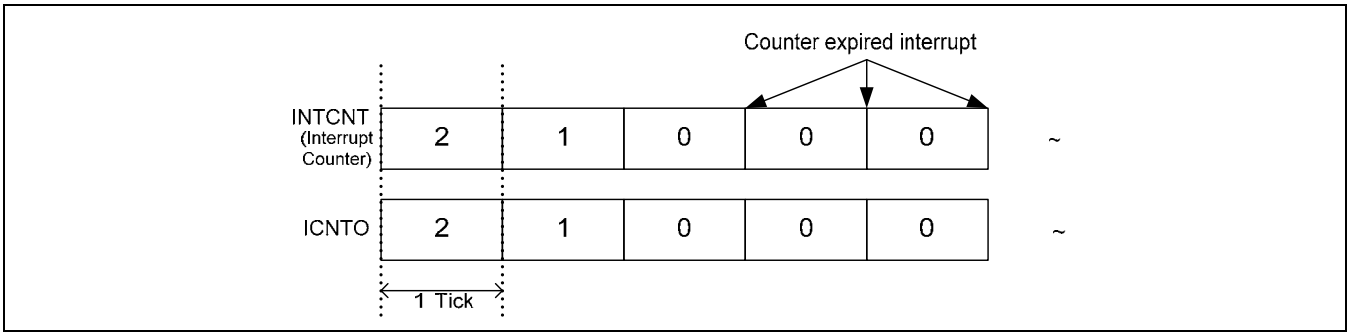


Figure 7.2-4 Timer Operation without Auto Reload (One-shot mode)

When Interrupt Manual Update (TCON[4]) is asserted off, after timer is expired, INTCNT (interrupt counter) is set as 0. And interrupt is occurred at every TICK.

6 USAGE MODEL

Follow the restrictions given below:

- TCLKB is equal to or slower than PCLKD1.
- Set count value as big as possible (To improve resolution).
- Prescaler, Divider Mux, TCLKB at TCFG must not be changed at run-time. If you want to change them, stop timers in advance and then change.

6.1 INTERRUPT

There are four kinds of interrupt source in system timer. The first one is interrupt counter expired, and the other three are SFR write status.

When you write value to TCON, TCNTB, and ICNTB SFR, the value may be not updated immediately to system timer-internal counter, because sometimes system timer uses RTC or XXTI clock for counter. They are slower than PCLK, operating clock of SFR. After those write status interrupts are asserted, SW can assure new value is really updated at internal counter. SW must wait until that time.

Note) TCON and ICNTB-writing interrupt can be disabled by SFR. But TCNTB-writing can't be disabled. When you write TCNTB SFR, that interrupt is always asserted.

If you don't want to assert interrupt, you can know update moment with continually reading INT_CSTAT[4:1] (polling method). In that case, you can selectively enable or disable each source of interrupt. Please refer TCFG[7:6] and INT_CSTAT[0].

6.2 COUNT VALUE UPDATE

Manual update (TCON[1] and TCON[4]) updates timer at that time. As described at 6.1, after write status interrupt is occurred, you can assure that internal counter is really updated.

Interval mode (TCON[2] and TCON[5]) makes timers to update value in ICNTB and TCNTB automatically, after timer counter is expired.

6.3 START TIMER

1. Set TCFG SFR to make appropriate TCLK.
   A. Select clock source.
   B. Set pre-scaler and divider value.
2. Set tick counter by writing to TCNTB SFR.
3. Wait until TCNTB write interrupt occurs, and write 1 to INT_CSTAT[2] to clear interrupt status bit.
4. Write TCON[1:0] = 2'b11 to update wanted value to internal timer counter at that time and start tick counter.
5. Wait until TCON write interrupt occurs, and write 1 to INT_CSTAT[4] to clear interrupt status bit.
6. If auto-reload mode is needed, TCON[1] must be de-asserted and write 1 to TCON[2].
7. Wait until TCON write interrupt occurs, and write 1 to INT_CSTAT[4] to clear interrupt status bit.
8. Set interrupt counter by writing to ICNTB SFR.
9. Wait until ICNTB write interrupt occurs, and write 1 to INT_CSTAT[3] to clear interrupt status bit.
10. Write TCON[4:3] = 2'b11 to update wanted value to internal interrupt counter at that time and start interrupt counter.
11. Wait until TCON write interrupt occurs, and write 1 to INT_CSTAT[4] to clear interrupt status bit.
12. If auto-reload mode is needed, TCON[4] must be de-asserted and write 1 to TCON[5]..
13. Wait until TCON write interrupt occurs, and write 1 to INT_CSTAT[4] to clear interrupt status bit.

6.4 STOP TIMER

1. Write INT_CSTAT[0] = 1'b0 to disable Interrupt counter expired (INTCNT=0) Interrupt.
2. Write TCON[3] = 1'b0 to stop internal interrupt counter.
3. Write TCON[0] = 1'b0 To stop internal timer counter.

6.5 CHANGE INTERVAL INTERRUPT AT RUN-TIME

To run system timer as figure 7.2-4,

1. Set new interrupt counter value by writing to ICNTB SFR.
2. Wait until ICNTB write interrupt occurs, and write 1 to INT_CSTAT[3] to clear interrupt status bit.
3. Write TCON [4] = 1'b1 to update new value to internal interrupt counter at that time.
4. Wait until TCON write interrupt occurs, and write 1 to INT_CSTAT[4] to clear interrupt status bit.

And the new value is updated after interrupt counter is expired.

7 REGISTER DESCRIPTION

7.1 REGISTER SUMMARY

Register	Address	R/W	Description	Reset Value
TCFG	EA10_0000	R/W	Configures 8-bit-Prescaler and Clock MUX	0x0000_0000
TCON	EA10_0004	R/W	Timer Control Register	0x0000_0000
TCNTB	EA10_0008	R/W	Tick Count Buffer Register	0x0000_0000
TCNTO	EA10_000C	R	Tick Count Observation Register	0x0000_0000
ICNTB	EA10_0010	R/W	Interrupt Count Buffer Register	0x0000_0000
ICNTO	EA10_0014	R	Interrupt Count Observation Register	0x0000_0000
INT_CSTAT	EA10_0018	R/W	Clears Interrupt	0x0000_0000

7.2 DETAILED DESCRIPTION

7.2.1 TIMER CONFIGURATION REGISTER (TCFG, R/W, Address = EA10_0000)

Timer Input Clock Frequency = TCLKB / ({prescaler value + 1}) / {divider value}

{prescaler value} = 1~63 / {divider value} = 1, 2, 4, 8, 16

TCFG	Bit	Description	Reset Value
TCLKB MUX	[13:12]	Selects clock input for TCLKB 00 = XXTI 01 = XrtcXTI 10 = XusbXTI 11 = PCLKD1 (APB) Usable clock source is restricted by power mode & oscillator pad configuration. XrtcXTI for any power mode & oscillator pad configuration. XXTI & XusbXTI for any power mode with turning on OSCs (XXTI and XusbXTI) PCLKD1 for just normal & idle mode To get detailed information, refer PMU (Power Management Unit) user's manual.	0x0
Divider MUX	[10:8]	Selects Mux input for Timer 000 = 1 001 = 1/2 010 = 1/4 011 = 1/8 100 = 1/16	0x0
IWIE	[7]	ICNTB write Interrupt Enable / 0: Disable, 1: Enable	0x0
TWIE	[6]	TCON Write Interrupt Enable / 0: Disable, 1: Enable	0x0
Prescaler	[5:0]	Prescaler value for timer	0x00

7.2.2 TIMER CONTROL REGISTER (TCON, R/W, Address = EA10_0004)

TCON	Bit	Description	Reset Value
Interrupt Auto Reload On/ Off	[5]	0 = No operation 1 = Interval mode (auto-reload)	0x0
Interrupt Manual Update	[4]	0 = No operation 1 = Update ICNTB and One-shot mode	0x0
Interrupt Start/ Stop	[3]	0 = Stop 1 = Start timer	0x0
Timer Auto Reload On/ Off	[2]	0 = No operation 1 = Interval mode (auto-reload)	0x0
Timer Manual Update	[1]	0 = No operation 1 = Update TCNTB and One-shot mode	0x0
Timer Start/ Stop	[0]	0 = Stop 1 = Start timer	0x0

If you want to use One-shot mode for interrupt, remain interrupt manual update bit as asserted (as '1'). In that case, interrupt occurs at every TICK after interrupt counter is reached to zero.
If you want to use interval mode for interrupt, de-assert (as '0') manual update bit after asserting it. In that case, the value in ICNTB is automatically reloaded to INTCNT counter when INTCNT is expired.
Either Manual Update or Auto Reload must be 1.

7.2.3 TICK COUNTER REGISTER (TCNTB, R/W, Address = EA10_0008)

Real Tick Counter Value = TCNTB+1, Do not use 0.

TCNTB	Bit	Description	Reset Value
Tick Count Buffer	[31:0]	Tick Count Buffer Register	0x0

7.2.4 TICK OBSERVATION REGISTER (TCNTO, R, Address = EA10_000C)

TCNTO	Bit	Description	Reset Value
Tick Count Observation	[31:0]	Tick Count Observation Register	0x0

7.2.5 INTERRUPT COUNTER REGISTER (ICNTB, R/W, Address = EA10_0010)

Real Interrupt Counter Value = ICNTB+1.
If ICNTB value is 0, interrupt occurs at every TICK.

ICNTB	Bit	Description	Reset Value
Interrupt Count Buffer	[31:0]	Interrupt Count Buffer Register	0x0

7.2.6 INTERRUPT OBSERVATION REGISTER (ICNTO, R, Address = EA10_0014)

ICNTO	Bit	Description	Reset Value
Interrupt Count Observation	[31:0]	Interrupt Count Observation Register	0x0

7.2.7 INTERRUPT CONTROL AND STATUS REGISTER (INT_CSTAT, R/W, Address = EA10_0018)

INT_CSTAT	Bit	Description	Reset Value
TCON Write Status	[4]	TCON Write Status Interrupt Bit. After internal counters are updated, this bit is asserted. Clear by writing '1' on this bit.	0x0
ICNTB Write Status	[3]	ICNTB Write Status Interrupt Bit. After internal counters are updated, this bit is asserted. Clear by writing '1' on this bit.	0x0
TCNTB Write Status	[2]	TCTNB Write Status Interrupt Bit. After internal counters are updated, this bit is asserted. Clear by writing '1' on this bit.	0x0
INTCNT counter expired Status	[1]	Interrupt counter expired (INTCNT=0) Interrupt Status Bit. When timer interrupt is occurred, this bit is asserted. Clear by writing '1' on this bit. **Clear delay: 7 cycles of TCLK plus 2 cycles of PCLK**	0x0
ICNTEIE	[0]	Interrupt counter expired (INTCNT=0) Interrupt Enables 1 = Enable 0 = Disable	0x0

7.3 WATCHDOG TIMER

1 OVERVIEW

The S5PC100 Watchdog Timer (WDT) is used to resume the controller operation if it is disturbed by malfunctions such as noise and system errors. It is used as a normal 16-bit interval timer to request interrupt service. The WDT generates the reset signal.

Difference between WDT and PWM timer is that WDT generates the reset signal.

2 FEATURES

- Supports Normal interval timer mode with interrupt request
- Activates Internal reset signal if the timer count value reaches 0 (Time-out).
- Supports Level-triggered Interrupt mechanism

3 FUNCTIONAL DESCRIPTION

3.1 WATCHDOG TIMER OPERATION

The Figure 7.3-1 shows the functional block diagram of the watchdog timer. The watchdog timer uses PCLK as its source clock. The PCLK frequency is prescaled to generate the corresponding watchdog timer clock, and the resulting frequency is divided again.

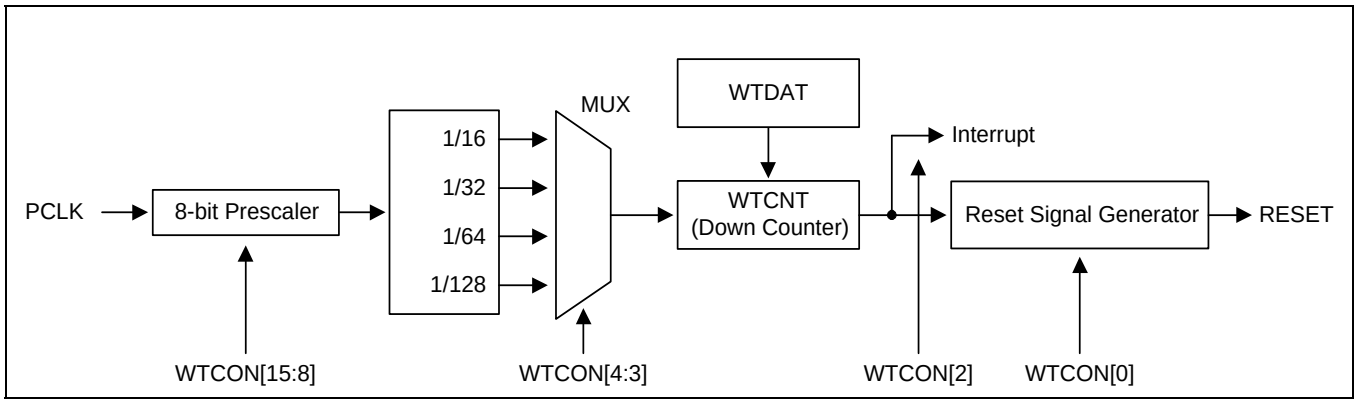


Figure 7.3-1 Watchdog Timer Block Diagram

The prescaler value and the frequency division factor are specified in the watchdog timer control (WTCON) register. Valid prescaler values range from 0 to 2^8-1. The frequency division factor can be selected as 16, 32, 64, or 128.

Use the following equation to calculate the watchdog timer clock frequency and the duration of each timer clock cycle:

t_watchdog = 1/(PCLK / (Prescaler value + 1) / Division_factor)

3.2 WTDAT & WTCNT

Once the watchdog timer is enabled, the value of watchdog timer data (WTDAT) register cannot be automatically reloaded into the timer counter (WTCNT). For this reason, an initial value must be written to the watchdog timer count (WTCNT) register, before the watchdog timer starts.

3.3 WDT START

To start WDT, set WTCON[0] and WTCON[5] as 1.

3.4 CONSIDERATION OF DEBUGGING ENVIRONMENT

If the S5PC100 is in debug mode using Embedded ICE, the watchdog timer must not operate.

The watchdog timer determines whether it is currently in the debug mode from the CPU core signal (DBGACK signal). Once the DBGACK signal is asserted, the reset output of the watchdog timer is not activated as the watchdog timer is expired.

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
WTCON	0xEA20_0000	R/W	Watchdog Timer Control Register	0x00008021
WTDAT	0xEA20_0004	R/W	Watchdog Timer Data Register	0x00008000
WTCNT	0xEA20_0008	R/W	Watchdog Timer Count Register	0x00008000
WTCLRINT	0xEA20_000C	W	Watchdog Timer Interrupt Clear Register	-

4.1 WATCHDOG TIMER CONTROL REGISTER (WTCON, R/W, ADDRESS = 0XEA20_0000)

The WTCON register allows you to enable/ disable the watchdog timer, select the clock signal from 4 different sources, enable/ disable interrupts, and enable/ disable the watchdog timer output.
The Watchdog timer is used to resume the S5PC100 restart on mal-function after its power on; if controller restart is not desired, the Watchdog timer should be disabled.

If you want to use the normal timer provided by the Watchdog timer, enable the interrupt and disable the Watchdog timer.

WTCON	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
Prescaler value	[15:8]	Prescaler value. The valid range is from 0 to (2^8-1).	0x80
Reserved	[7:6]	Reserved. These two bits must be 00 in normal operation.	00
Watchdog timer	[5]	Enable or disable Watchdog timer bit. 0 = Disable 1 = Enable	1
Clock select	[4:3]	Determines the clock division factor. 00 = 16 01 = 32 10 = 64 11 = 128	00
Interrupt generation	[2]	Enable or disable interrupt bit. 0 = Disable 1 = Enable	0
Reserved	[1]	Reserved. This bit must be 0 in normal operation.	0
Reset enable/disable	[0]	Enable or disable Watchdog timer output bit for reset signal. 1 = Asserts reset signal of the S5PC100 at watchdog time-out 0 = Disable the reset function of the watchdog timer.	1

4.2 WATCHDOG TIMER DATA REGISTER (WTDAT, R/W, ADDRESS = 0XEA20_0004)

The WTDAT register is used to specify the time-out duration. The content of WTDAT cannot be automatically loaded into the timer counter at initial watchdog timer operation. However, using 0x8000 (initial value) drives the first time-out. In this case, the value of WTDAT is automatically reloaded into WTCNT.

WTDAT	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
Count reload value	[15:0]	Watchdog timer count value for reload.	0x8000

4.3 WATCHDOG TIMER COUNT REGISTER (WTCNT, R/W, ADDRESS = 0XEA20_0008)

The WTCNT register contains the current count values for the watchdog timer during normal operation. Note that the content of the WTDAT register cannot be automatically loaded into the timer count register if the watchdog timer is enabled initially, therefore the WTCNT register must be set to an initial value before enabling it.

WTCNT	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
Count value	[15:0]	The current count value of the watchdog timer	0x8000

4.4 WATCHDOG TIMER INTERRUPT CLEAR REGISTER (WTCLRINT, W, ADDRESS = 0XEA20_000C)

The WTCLRINT register is used to clear the interrupt. Interrupt service routine is responsible for clearing the relevant interrupt after the interrupt service is complete. Writing any values on this register clears the interrupt. Reading this register is not allowed.

WTCLRINT	Bit	Description	Reset Value
Interrupt clear	[31:0]	Write any values clears the interrupt	-

7.4 REAL TIME CLOCK (RTC)

1 OVERVIEW

The Real Time Clock (RTC) unit can be operated by the backup battery while the system power is off. The data include the time by Second, Minute, Hour, Date, Day, Month, and Year. The RTC unit works with an external 32.768 kHz crystal and performs the alarm function.

2 FEATURES

- BCD Number: Second, Minute, Hour, Date, Day, Month, and Year.
- Leap Year Generator
- Alarm Function: Alarm-Interrupt or Wake-up from Power-off mode
- Tick Counter Function: Tick-Interrupt or Wake-up from Power-off mode.
- Year 2000 problem is removed.
- Independent Power Pin (RTCVDD)
- Supports millisecond tick time interrupt for RTOS kernel time tick.

3 REAL TIME CLOCK OPERATION DESCRIPTION

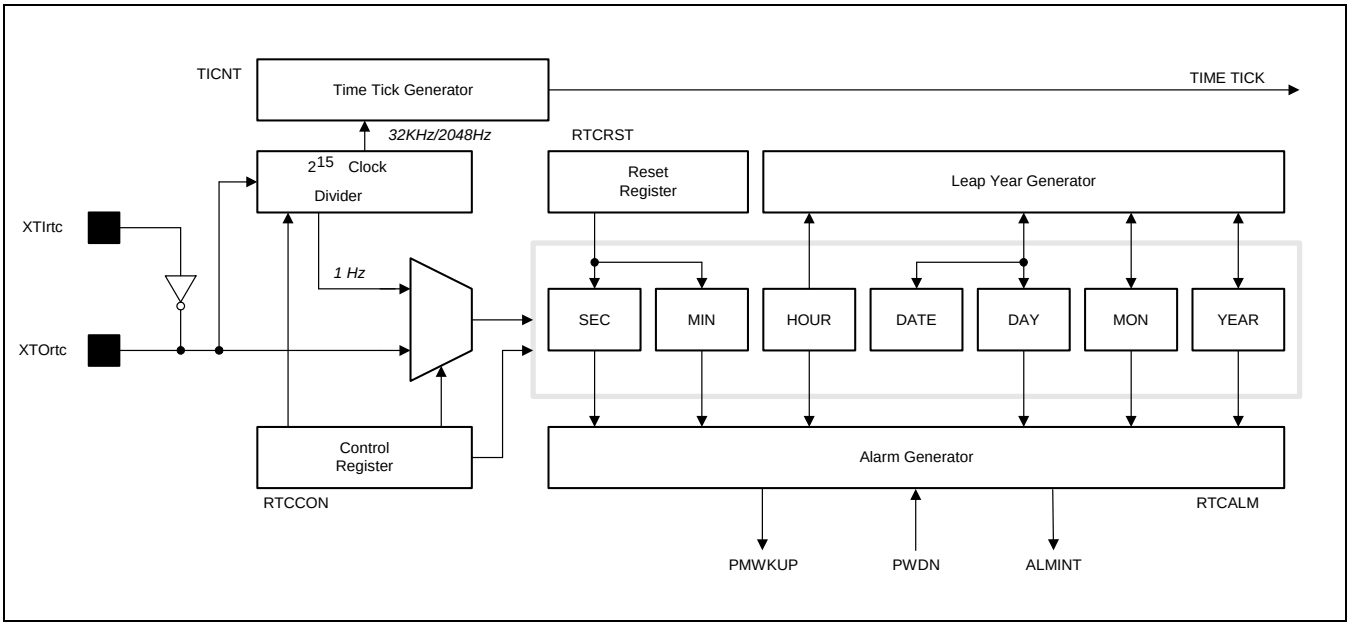


Figure 7.4-1 Real Time Clock Block Diagram

4 LEAP YEAR GENERATOR

The leap year generator determines the last date of each month out of 28, 29, 30, or 31, based on data from BCDDAY, BCDMON, and BCDYEAR. This block considers leap year in deciding the last date. An 8-bit counter represents 2 BCD digits, therefore it cannot decide whether "00" year (Year with its last two digits zeros) is a leap year or not. For example, it cannot discriminate between 1900 and 2000. To solve this problem, the RTC block in S5PC100 has hard-wired logic to support the leap year in 2000. Note 1900 is not leap year while 2000 is leap year. Therefore, two digits of 00 in S5PC100 denote 2000, not 1900.

5 READ / WRITE REGISTER

Set Bit 0 of the RTCCON register to high in order to write the BCD register in RTC block. To display the second, minute, hour, day, date, month, and year, the CPU should read the data in BCDSEC, BCDMIN, BCDHOUR, BCDDATE, BCDDAY, BCDMON, and BCDYEAR registers, respectively, in the RTC block. However, a one second deviation may exist because multiple registers are read. For example, if you read the registers from BCDYEAR to BCDMIN, the result is assumed to be 2059 (Year), 12 (Month), 31 (Date), 23 (Hour) and 59 (Minute). If you read the BCDSEC register and the value ranges from 1 to 59 (Second), there is no problem, but, if the value is 0 sec., the year, month, date, hour, and minute may be changed to 2060 (Year), 1 (Month), 1 (Date), 0 (Hour) and 0 (Minute) because of the one second deviation that was mentioned. In this case, you must re-read from BCDYEAR to BCDSEC if BCDSEC is zero.

6 BACKUP BATTERY OPERATION

The RTC logic can be driven by the backup battery, which supplies the power through the RTCVDD pin into the RTC block, even if the system power is off. If the system is off, the interfaces of the CPU and RTC logic should be blocked, and the backup battery only drives the oscillation circuit and the BCD counters to minimize power dissipation.

7 ALARM FUNCTION

The RTC generates an alarm signal at a specified time in the power-off mode or normal operation mode. In normal operation mode, the alarm interrupt (ALMINT) is activated. In the power-off mode, the power management wakeup (PMWKUP) signal is activated as well as the ALMINT. The RTC alarm register (RTCALM) determines the alarm enable/ disable status and the condition of the alarm time setting.

8 TICK TIME INTERRUPT

The RTC tick time is used for interrupt request. The TICNT register has an interrupt enable bit and the count value for the interrupt. The count value reaches '0' if the tick time interrupt occurs. Then the period of interrupt is as follows:

Period = (n+1) / (Tick clock source frequency) second (n= tick counter value)

Table 7.4-1 Tick Interrupt Resolution

Tick Counter Clock Source Selection	Tick Clock Source Frequency(Hz)	Clock Range (s)	Resolution (ms)
4'b0000	32768 (2^15)	0 ~ 131071	0.03
4'b0001	16384 (2^14)	0 ~ 262143	0.06
4'b0010	8192 (2^13)	0 ~ 524287	0.12
4'b0011	4096 (2^12)	0 ~ 1048575	0.24
4'b0100	2048 (2^11)	0 ~ 2097151	0.49
4'b0101	1024 (2^10)	0 ~ 4194303	0.97
4'b0110	512 (2^9)	0 ~ 8388607	1.95
4'b0111	256 (2^8)	0 ~ 16777215	3.90
4'b1000	128 (2^7)	0 ~ 33554431	7.81
4'b1001	64 (2^6)	0 ~ 67108863	15.62
4'b1010	32 (2^5)	0 ~ 134217727	31.25
4'b1011	16 (2^4)	0 ~ 268435455	62.50
4'b1100	8 (2^3)	0 ~ 536870911	125
4'b1101	4 (2^2)	0 ~ 1073741823	250
4'b1110	2	0 ~ 2147483647	500
4'b1111	1	0 ~ 4294967295	1000

NOTE: Tick time resolution is extended by selecting the appropriate tick time clock source.

This RTC time tick may be used for Real Time Operating System (RTOS) kernel time tick.

If time tick is generated by the RTC time tick, the time related function of RTOS always synchronized in real time.

9 32.768KHZ X-TAL CONNECTION EXAMPLE

The Figure 7.4-2 shows a circuit of the RTC unit oscillation at 32.768kHz.

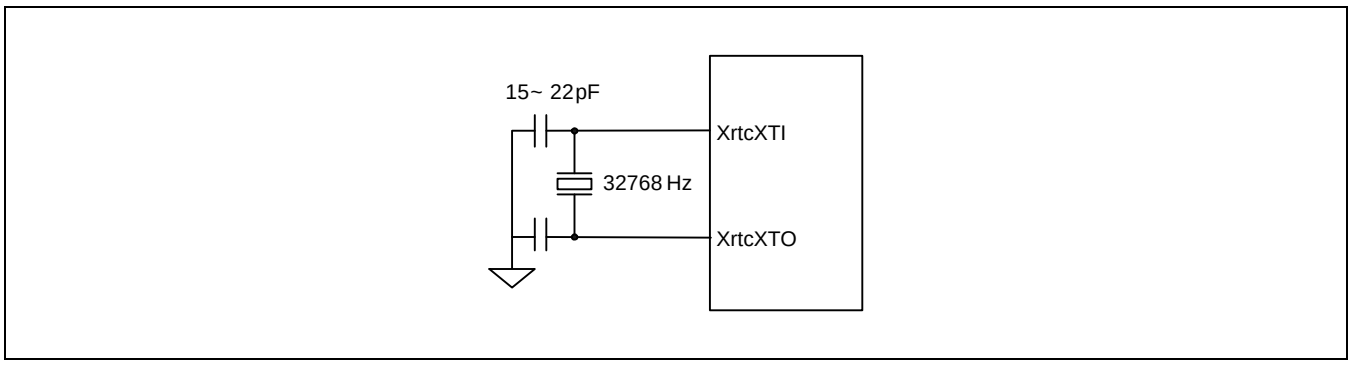


Figure 7.4-2 Main Oscillator Circuit Example

Note. The external components commonly used for the above circuit are a resonator, two capacitors(15~22 pF) and a resistor(>5MΩ) between XrtcXTI and XrtcXTO.

10 DRIVE STRENGTH CONTROL

For more information, refer to ETC4DRV(0xE030_056C) in Pad Control.doc

SFR Name	Field Value	Min	Type	Max	Unit
ETC4DRV	1	1.7	1.8	1.9	V
	0(default)	2.3	2.5~3.3	3.6	V

11 RTC START

To start RTC, set RTCCON[0] as 1.

12 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
RTC_XTI	Input	32.768 KHz RTC Oscillator Clock Input	XrtcXTI	Dedicated
RTC_XXTO	Output	32.768 KHz RTC Oscillator Clock output	XrtcXTO	Dedicated

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

13 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
INTP	0xEA30_0030	R/W	Interrupt Pending Register	0x00000000
RTCCON	0xEA30_0040	R/W	RTC Control Register	0x00000000
TICCNT	0xEA30_0044	R/W	Tick Time Count Register	0x00000000
RTCALM	0xEA30_0050	R/W	RTC Alarm Control Register	0x00000000
ALMSEC	0xEA30_0054	R/W	Alarm Second Data Register	0x00000000
ALMMIN	0xEA30_0058	R/W	Alarm Minute Data Register	0x00000000
ALMHOUR	0xEA30_005C	R/W	Alarm Hour Data Register	0x00000000
ALMDATE	0xEA30_0060	R/W	Alarm Date Data Register	0x00000000
ALMMON	0xEA30_0064	R/W	Alarm Month Data Register	0x00000000
ALMYEAR	0xEA30_0068	R/W	Alarm Year Data Register	0x00000000
BCDSEC	0xEA30_0070	R/W	BCD Second Register	Undefined
BCDMIN	0xEA30_0074	R/W	BCD Minute Register	Undefined
BCDHOUR	0xEA30_0078	R/W	BCD Hour Register	Undefined
BCDDATE	0xEA30_007C	R/W	BCD Date Register	Undefined
BCDDAY	0xEA30_0080	R/W	BCD Day Register	Undefined
BCDMON	0xEA30_0084	R/W	BCD Month Register	Undefined
BCDYEAR	0xEA30_0088	R/W	BCD Year Register	Undefined
CURTICCNT	0xEA30_0090	R	Current Tick Time Counter Register	0x00000000

13.1 INTERRUPT PENDING REGISTER (INTP, R/W, ADDRESS = 0XEA30_0030)

You can clear specific bits of INTP register by writing 1's to the bits that you want to clear regardless of RTCEN value.

INTP	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0
ALARM	[1]	Alarm interrupt pending bit 0 = No interrupt occurred 1 = Interrupt occurred	0
Time TIC	[0]	Time TIC interrupt pending bit 0 = No interrupt occurred 1 = Interrupt occurred.	0

13.2 REAL TIME CLOCK CONTROL REGISTER (RTCCON, R/W, ADDRESS = 0XEA30_0040)

The RTCCON register consists of 9 bits such as the RTCEN, which controls the read/ write enable of the BCD SEL, CNTSEL, CLKRST, TICCKSEL and TICEN for testing.

RTCEN bit controls all interfaces between the CPU and the RTC, therefore it should be set to 1 in an RTC control routine to enable data read/ write after a system reset.Before power off, the RTCEN bit should be cleared to 0 to prevent inadvertent writing into RTC registers.

RTCCON	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
TICEN	[8]	Tick timer enable 0 = Disables 1 = Enables	0
TICCKSEL	[7:4]	Tick timer sub clock selection. 4'b0000 = 32768 hz 4'b0001 = 16384 hz 4'b0010 = 8192 hz 4'b0011 = 4096 hz 4'b0100 = 2048 hz 4'b0101 =1024 hz 4'b0110 =512 hz 4'b0111 =256 hz 4'b1000 =128 hz 4'b1001 =64 hz 4'b1010 =32 hz 4'b1011 =16 hz 4'b1100 =8 hz 4'b1101 =4 hz 4'b1110 =2 hz 4'b1111 =1 hz	4'b0000
CLKRST	[3]	RTC clock count reset. 0 = No reset 1 = Reset	0
CNTSEL	[2]	BCD count select. 0 = Merge BCD counters 1 = Reserved (Separate BCD counters)	0
CLKSEL	[1]	BCD clock select. 0 = XTAL $1/2^{15}$ divided clock 1 = Reserved (XTAL clock only for test)	0
RTCEN	[0]	RTC control enable. 0 = Disables 1 = Enables **Note:** If RTCEN is enable, BCD time count setting, RTC clock counter reset and read operation is performed.	0

13.3 TICK TIME COUNT REGISTER (TICNT, R/W, ADDRESS = 0XEA30_0044)

TICNT	Bit	Description	Reset Value
TICK TIME COUNT	[31:0]	32-bit tick time count value This value must not be 0.	32'b0

13.4 RTC ALARM CONTROL REGISTER (RTCALM, R/W, ADDRESS = 0XEA30_0050)

The RTCALM register determines the alarm enable and the alarm time. Note that the RTCALM register generates the alarm signal through both ALMINT and PMWKUP in power down mode, but only through ALMINT in the normal operation mode.

RTCALM	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
ALMEN	[6]	Alarm global enable 0 = Disables 1 = Enables	0
YEAREN	[5]	Year alarm enable 0 = Disables 1 = Enables	0
MONEN	[4]	Month alarm enable 0 = Disables 1 = Enables	0
DATEEN	[3]	Date alarm enable 0 = Disables 1 = Enables	0
HOUREN	[2]	Hour alarm enable 0 = Disables 1 = Enables	0
MINEN	[1]	Minute alarm enable 0 = Disables 1 = Enables	0
SECEN	[0]	Second alarm enable 0 = Disables 1 = Enables	0

13.5 ALARM SECOND DATA REGISTER (ALMSEC, R/W, ADDRESS = 0XEA30_0054)

ALMSEC	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
SECDATA	[6:4]	BCD value for alarm second. 0 ~ 5	000
	[3:0]	0 ~ 9	0000

13.6 ALARM MIN DATA REGISTER (ALMMIN, R/W, ADDRESS = 0XEA30_0058)

ALMMIN	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
MINDATA	[6:4]	BCD value for alarm minute. 0 ~ 5	000
	[3:0]	0 ~ 9	0000

13.7 ALARM HOUR DATA REGISTER (ALMHOUR, R/W, ADDRESS = 0XEA30_005C)

ALMHOUR	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0
HOURDATA	[5:4]	BCD value for alarm hour. 0 ~ 2	00
	[3:0]	0 ~ 9	0000

13.8 ALARM DATE DATA REGISTER (ALMDATE, R/W, ADDRESS = 0XEA30_0060)

ALMDATE	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0
DATEDATA	[5:4]	BCD value for alarm date, from 0 to 28, 29, 30, 31. 0 ~ 3	00
	[3:0]	0 ~ 9	0000

13.9 ALARM MONTH DATA REGISTER (ALMMON, R/W, ADDRESS = 0XEA30_0064)

ALMMON	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
MONDATA	[4]	BCD value for alarm month. 0 ~ 1	0
	[3:0]	0 ~ 9	0000

13.10 ALARM YEAR DATA REGISTER (ALMYEAR, R/W, ADDRESS = 0XEA30_0068)

ALMYEAR	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
YEARDATA	[7:4]	BCD value for year. 0 ~ 9	0x0
	[3:0]	0 ~ 9	0x0

13.11 BCD SECOND REGISTER (BCDSEC, R/W, ADDRESS = 0XEA30_0070)

BCDSEC	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	-
SECDATA	[6:4]	BCD value for second. 0 ~ 5	-
	[3:0]	0 ~ 9	-

13.12 BCD MINUTE REGISTER (BCDMIN, R/W, ADDRESS = 0XEA30_0074)

BCDMIN	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	-
MINDATA	[6:4]	BCD value for minute. 0 ~ 5	-
	[3:0]	0 ~ 9	-

13.13 BCD HOUR REGISTER (BCDHOUR, R/W, ADDRESS = 0XEA30_0078)

BCDHOUR	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	-
HOURDATA	[5:4]	BCD value for hour. 0 ~ 2	-
	[3:0]	0 ~ 9	-

13.14 BCD DATE REGISTER (BCDDATE, R/W, ADDRESS = 0XEA30_007C)

BCDDAY	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	-
DATEDATA	[5:4]	BCD value for date. 0 ~ 3	-
	[3:0]	0 ~ 9	-

13.15 BCD DAY REGISTER (BCDDAY, R/W, ADDRESS = 0XEA30_0080)

BCDDAY	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	-
DAYDATA	[2:0]	BCD value for a day of the week. 1 ~ 7	-

13.16 BCD MONTH REGISTER (BCDMON, R/W, ADDRESS = 0XEA30_0084)

BCDMON	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	-
MONDATA	[4]	BCD value for month. 0 ~ 1	-
	[3:0]	0 ~ 9	-

13.17 BCD YEAR REGISTER (BCDYEAR, R/W, ADDRESS = 0XEA30_0088)

BCDYEAR	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	-
YEARDATA	[7:4]	BCD value for year. 0 ~ 9	-
	[3:0]	0 ~ 9	

13.18 TICK COUNTER REGISTER (CURTICCNT, R, ADDRESS = 0XEA30_0090)

CURTICCNT	Bit	Description	Reset Value
Tick counter observation	[31:0]	Current tick count value	-

8.1 UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER

1 OVERVIEW

The S5PC100 Universal Asynchronous Receiver and Transmitter (UART) provide four independent asynchronous serial Input/ Output (I/O) (SIO) ports. Both the ports operate in interrupt-based or DMA-based mode. In other words, UART generates an interrupt or a DMA request to transfer data between CPU and the UART. The UART ch0, ch2 supports bit rates up to 115.2K bps and UART ch1 and ch3 supports bit rates up to 3M bps. If an external device provides the UART with UCLK, then the UART operates at a higher speed. Each UART channel contains two 64-byte FIFOs to receive and transmit.

UART includes programmable baud rates, infrared (IR) transmit/receive, one or two stop bit insertion, 5-bit, 6-bit, 7-bit, or 8-bit data width and parity checking.

Each UART contains a baud-rate generator, a transmitter, a receiver and a control unit, as shown in Figure 8.1-1. The baud-rate generator is clocked by PCLK or UCLK. The transmitter and the receiver contain 64-byte FIFOs and data shifters. The data to be transmitted is written to FIFO, and copied to the transmit shifter. The data is then shifted out by the transmit data pin (TxDn). The received data is shifted from the receive data pin (RxDn), and copied to FIFO from the shifter.

2 FEATURES

- RxD0, TxD0, RxD1, TxD1, RxD2, TxD2, RxD3 and TxD3 with DMA-based or interrupt-based operation
- UART Ch 0, 1, 2 and 3 with IrDA 1.0 & 64-byte FIFO
- UART Ch 0, 1 and 2 with nRTS0, nCTS0, nRTS1, nCTS1, nCTS2 and nRTS2 for Auto Flow Control
- Supports high-speed operation in UART Ch1 and 3
- Supports handshake transmit/receive

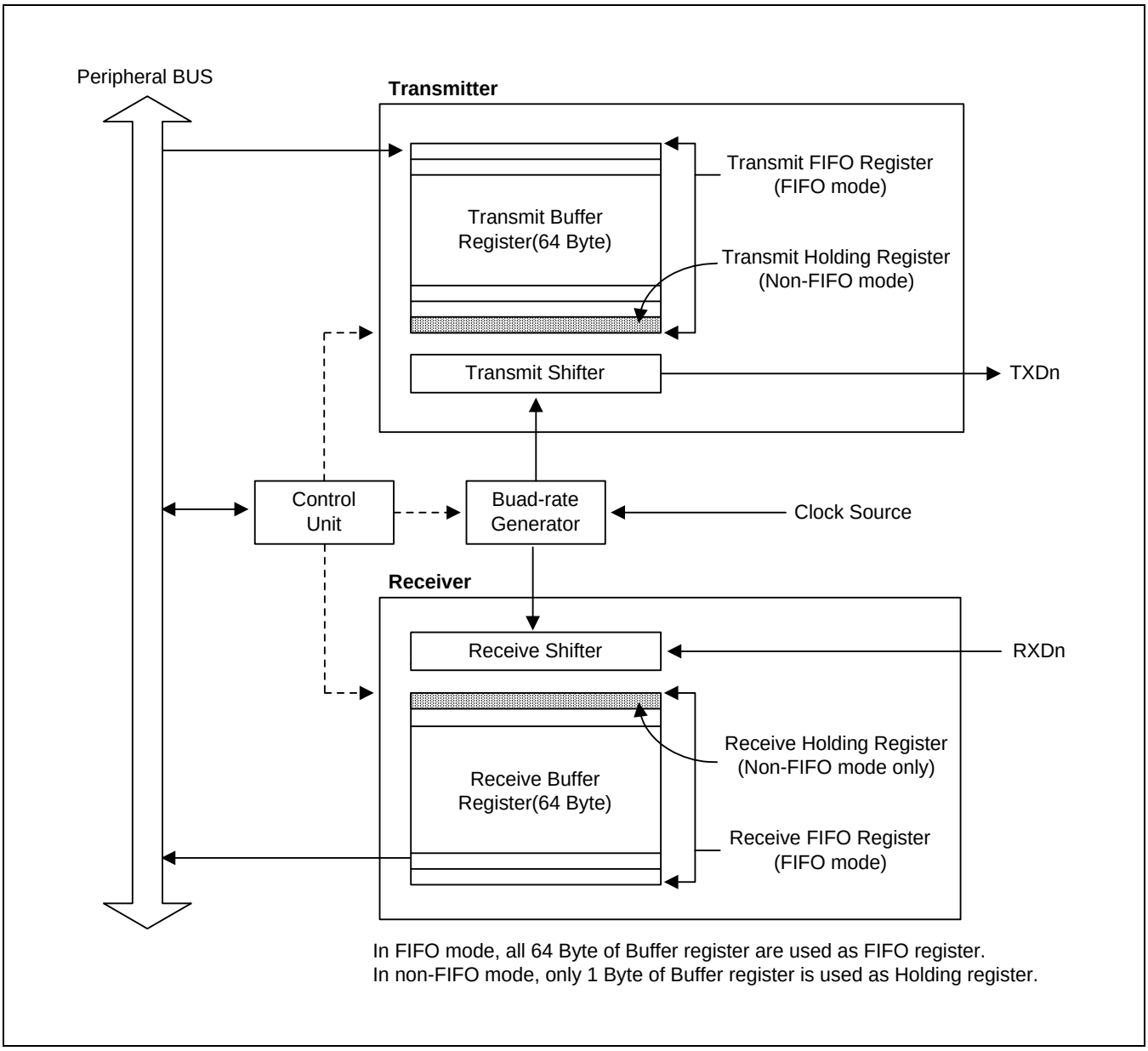


Figure 8.1-1 UART Block Diagram

3 DESCRIPTION

The following sections describe UART operations, including data transmission, data reception, interrupt generation, baud-rate generation, Loop-back mode, infrared mode, and auto flow control.

3.1 DATA TRANSMISSION

The data frame for transmission is programmable. It consists of a start bit, 5 to 8 data bits, an optional parity bit, and 1 to 2 stop bits, which are specified by the line control register (ULCONn). The transmitter can also produce the break condition, which forces the serial output to logic 0 state for one frame transmission time. This block transmits break signals after the present transmission word is transmitted completely. After the break signal transmission, it continuously transmits data into the Tx FIFO (Tx holding register in case of Non-FIFO mode).

3.2 DATA RECEPTION

Similar to data transmission, the data frame for reception is also programmable. It consists of a start bit, 5 to 8 data bits, an optional parity bit, and 1 to 2 stop bits in the line control register (ULCONn). The receiver detects overrun error, parity error, frame error and break condition, each of which sets an error flag.

- Overrun error indicates that new data has overwritten the old data before the old data has been read.
- Parity error indicates that the receiver has detected an unexpected parity condition.
- Frame error indicates that the received data does not have a valid stop bit.
- Break condition indicates that the RxDn input is held in the logic 0 state for more than one frame transmission time.

Receive time-out condition occurs if no data is received during the 3 word time (This interval follows the setting of Word Length bit) and the Rx FIFO is not empty in the FIFO mode.

3.3 AUTO FLOW CONTROL(AFC)

The S5PC100's UART0 and UART1 support auto flow control with nRTS and nCTS signals. UART2 supports auto flow control if TxD3 and RxD3 are set as nRTS2 and nCTS2 by GPA1CON(GPIO SFR). In case, it can be connected to external UARTs. To connect a UART to a Modem, disable auto flow control bit in UMCONn register and control the signal of nRTS by software.

In AFC, nRTS depends on the condition of the receiver and nCTS signals control the operation of the transmitter. The UART's transmitter transfers the data in FIFO if nCTS signals are activated (In AFC, nCTS means that other UART's FIFO is ready to receive data). Before UART receives the data, nRTS must be activated if its receive FIFO has more than 2-byte as spare. The nRTS must be inactivated if its receive FIFO has less than 1-byte as spare (In AFC, nRTS means that its own receive FIFO is ready to receive data).

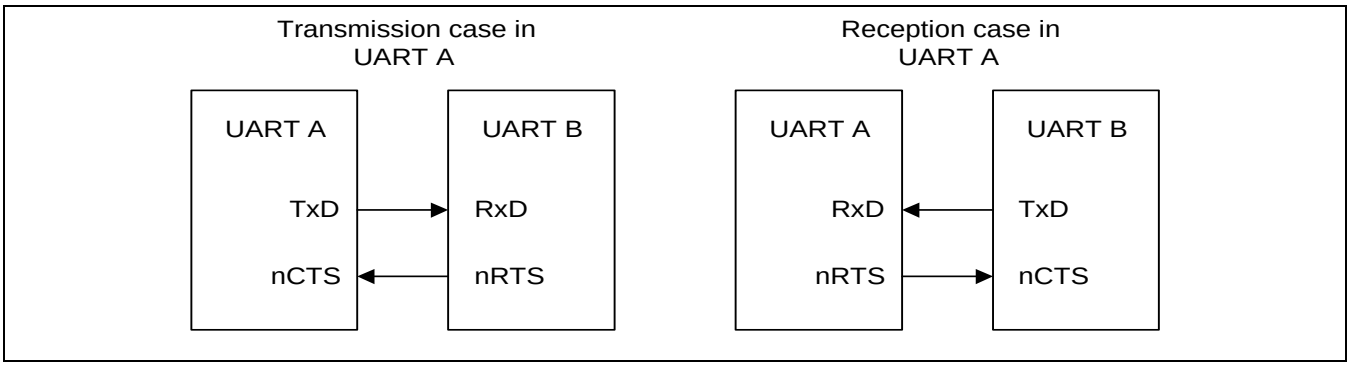


Figure 8.1-2 UART AFC interface

3.4 EXAMPLE OF NON AUTO-FLOW CONTROL (CONTROLLING NRTS AND NCTS BY SOFTWARE)

3.4.1 Rx operation with FIFO

1. Select transmit mode (Interrupt or DMA mode).
2. Check the value of Rx FIFO count in UFSTATn register. If the value is less than 16, you must set the value of UMCONn[0] to '1' (activate nRTS), and if it is equal or larger than 16 you must set the value to '0' (inactivate nRTS).
3. Repeat the Step 2.

3.4.2 Tx operation with FIFO

1. Select transmit mode (Interrupt or DMA mode).
2. Check the value of UMSTATn[0]. If the value is '1' (activate nCTS), you must write the data to Tx FIFO register.
3. Repeat the Step 2

3.5 RS-232C INTERFACE

To connect the UART to modem interface (instead of null modem), nRTS, nCTS, nDSR, nDTR, DCD and nRI signals are required. In this case, you can control these signals with general I/O ports by software because the AFC does not support the RS-232C interface.

3.6 INTERRUPT/DMA REQUEST GENERATION

Each UART of the S5PC100 has seven status (Tx/Rx/Error) signals, namely: Overrun error, Parity error, Frame error, Break, Receive buffer data ready, Transmit buffer empty, and Transmit shifter empty. These conditions are indicated by the corresponding UART status register (UTRSTATn/UERSTATn).

The Overrun Error, Parity Error, Frame Error and Break Condition are referred as the receive error status. If receive-error-status-interrupt-enable bit is set to 1 in the control register, UCONn, receive error status generates receive-error-status-interrupt. If a receive-error-status-interrupt-request is detected, you can identify the source of interrupt by reading the value of UERSTSTn.

If the receiver transfers the data of the receive shifter to the receive FIFO register in FIFO mode and the number of received data reaches Rx FIFO Trigger Level, Rx interrupt is generated, if Receive mode in control register (UCONn) is set to 1 (Interrupt request or polling mode).

In Non-FIFO mode, transferring the data of receive shifter to receive holding register causes Rx interrupt under the Interrupt request and polling mode.

If the transmitter transfers data from its transmit FIFO register to its transmit shifter and the number of data left in transmit FIFO reaches Tx FIFO Trigger Level, Tx interrupt is generated (provided Transmit mode in control register is selected as Interrupt request or polling mode). In Non-FIFO mode, transferring data from the transmit holding register to transmit shifter causes Tx interrupt under the Interrupt request and polling mode.

Remember that the Tx interrupt is always requested if the number of data in the transmit FIFO is smaller than the trigger level. This means that an interrupt is requested as soon as you enable the Tx interrupt unless you fill the Tx buffer prior to that. It is recommended to fill the Tx buffer first and then enable the Tx interrupt.

The interrupt controllers of S5PC100 are level-triggered type. You must set the interrupt type as 'Level' if you program the UART control registers.

If Receive mode and Transmit mode in control register are selected as DMAn request mode then DMAn request occurs instead of Rx or Tx interrupt in the above situation.

Table 8.1-1 Interrupts in Connection with FIFO

Type	FIFO Mode	Non-FIFO Mode
Rx interrupt	Generated if received data reaches the trigger level of received FIFO. Generated if the number of data in FIFO does not reaches Rx FIFO trigger Level and does not receive any data during 3 word time (receive time out). This interval follows the setting of Word Length bit.	Generated by receive holding register whenever receive buffer becomes full.
Tx interrupt	Generated if transmit data reaches the trigger level of transmit FIFO (Tx FIFO trigger Level).	Generated by transmit holding register whenever transmit buffer becomes empty.
Error interrupt	Generated if frame error, parity error, or break signal are detected. Generated if it gets to the top of the receive FIFO without reading out data in it (overrun error).	Generated by all errors. However if another error occurs at the same time, only one interrupt is generated.

3.7 UART ERROR STATUS FIFO

UART has the error status FIFO besides the Rx FIFO register. The error status FIFO indicates which data, among FIFO registers, is received with an error. An error interrupt is issued only if the data, which has an error, is ready to read out. To clear the error status FIFO, the URXHn with an error and UERSTATn must be read out.

For example, it is assumed that the UART Rx FIFO receives A, B, C, D, and E characters sequentially and the frame error occurs while receiving 'B', and the parity error occurs while receiving 'D'.

The actual UART receive error does not generate any error interrupt, since the character, which was received with an error, has not been read yet. The error interrupt occurs if the character is read out.

Time	Sequence Flow	Error Interrupt	Note
#0	If no character is read out	-	
#1	A, B, C, D, and E is received	-	
#2	After A is read out	Frame error (in B) interrupt occurs.	The 'B' has to be read out.
#3	After B is read out	-	
#4	After C is read out	Parity error (in D) interrupt occurs.	The 'D' has to be read out.
#5	After D is read out	-	
#6	After E is read out	-	

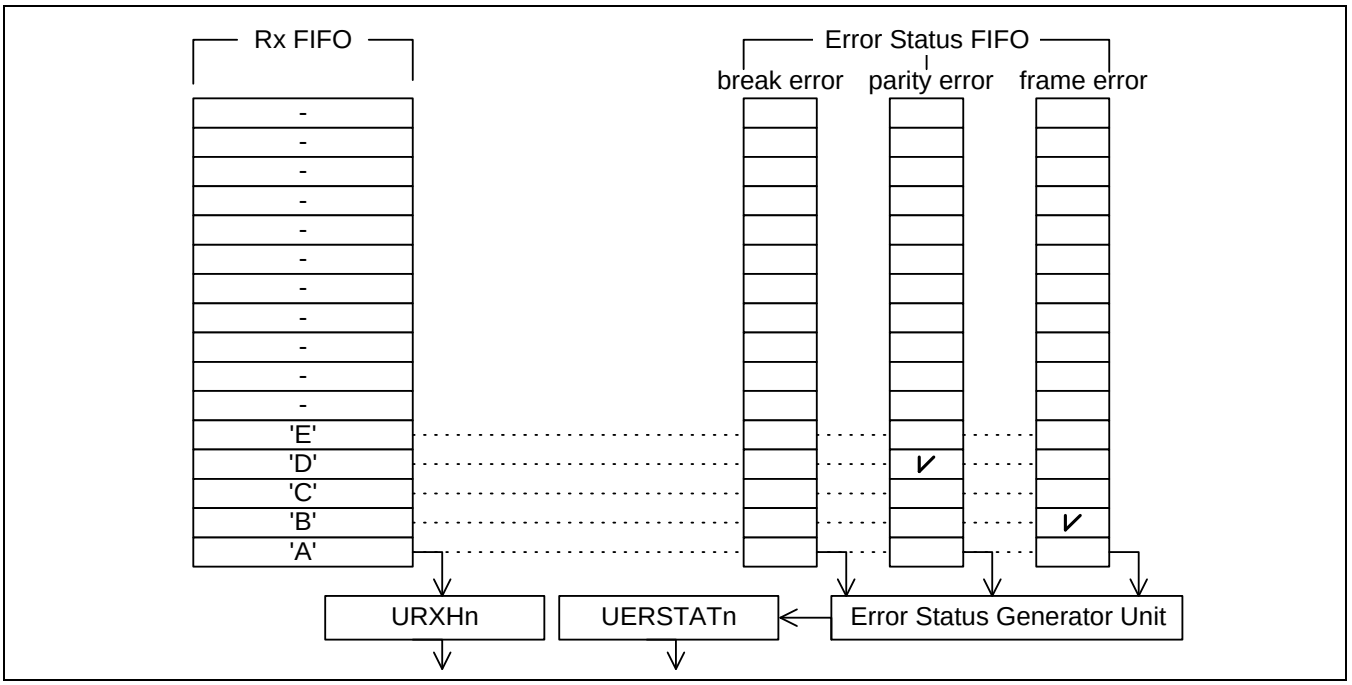


Figure 8.1-3 UART Receives the five Characters including two errors.

3.7.1 Infra-Red (IR) Mode

The S5PC100 UART block supports infra-red (IR) transmission and reception. It is selected by setting the Infra-red-mode bit in the UART line control register (ULCONn). Figure 8.1- illustrates how to implement the IR mode. In IR transmit mode, the transmit pulse comes out at a rate of 3/16, the normal serial transmit rate (If the transmit data bit is 0); In IR receive mode, the receiver must detect the 3/16 pulsed period to recognize a 0 value (Refer frame timing diagrams shown in Figure 8.1- and Figure 8.1-).

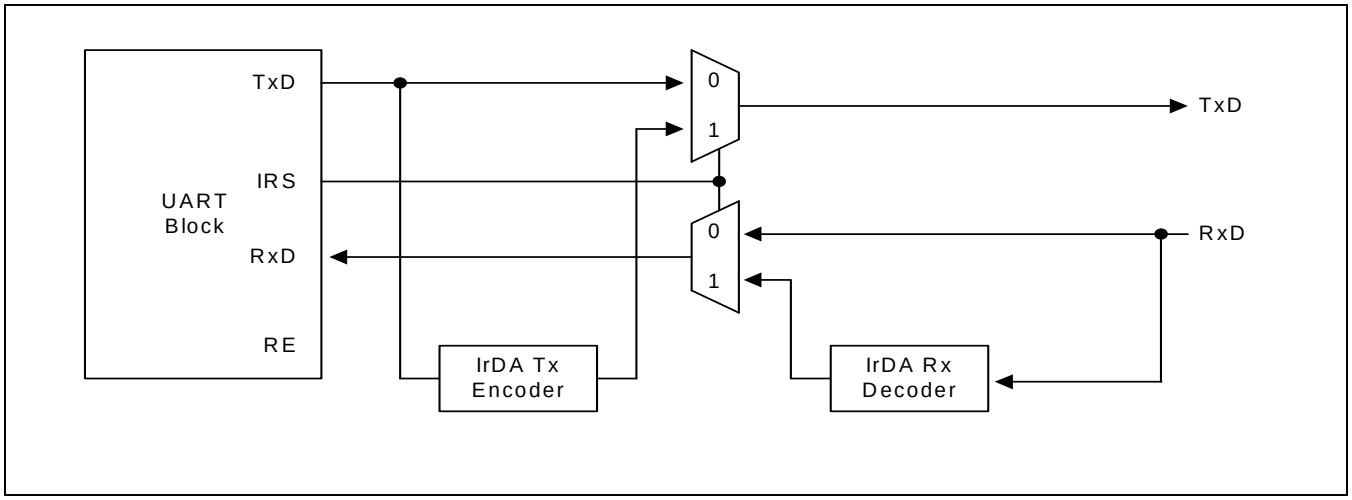


Figure 8.1-4 IrDA Function Block Diagram

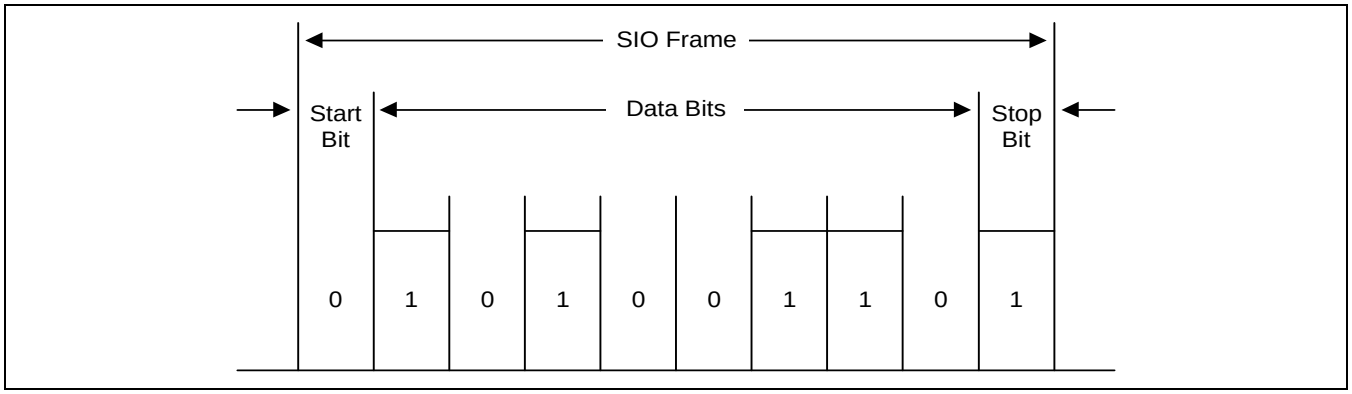


Figure 8.1-5 Serial I/O Frame Timing Diagram (Normal UART)

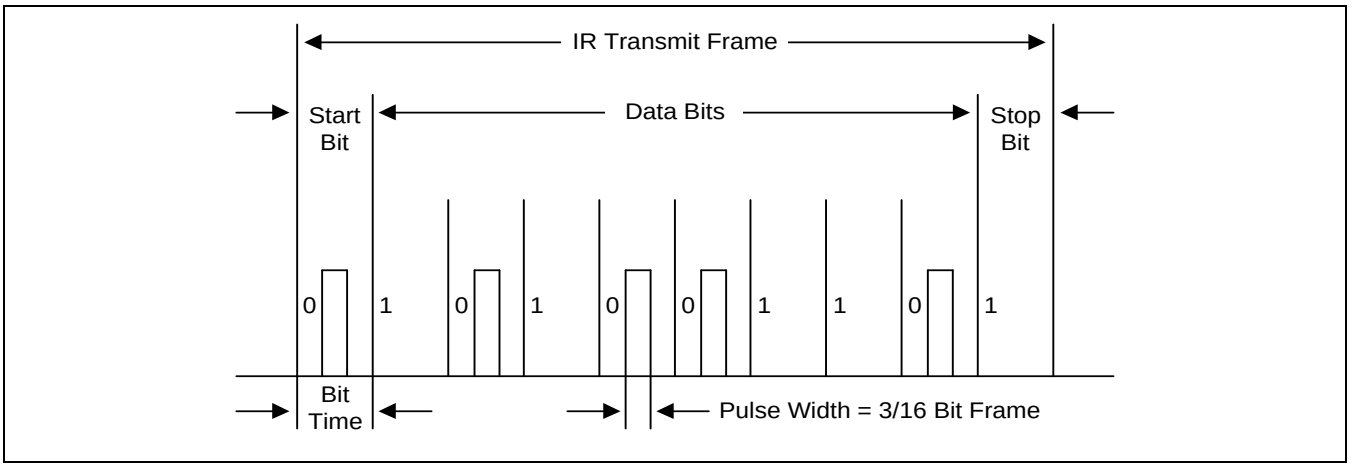


Figure 8.1-6 Infra-Red Transmit Mode Frame Timing Diagram

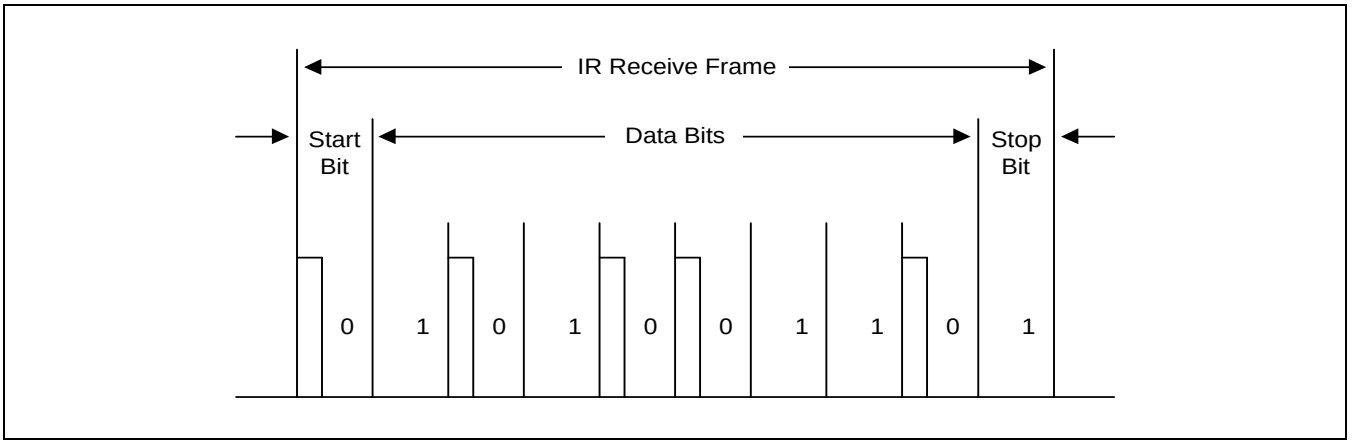


Figure 8.1-7 Infra-Red Receive Mode Frame Timing Diagram

4 UART INPUT CLOCK DIAGRAM DESCRIPTION

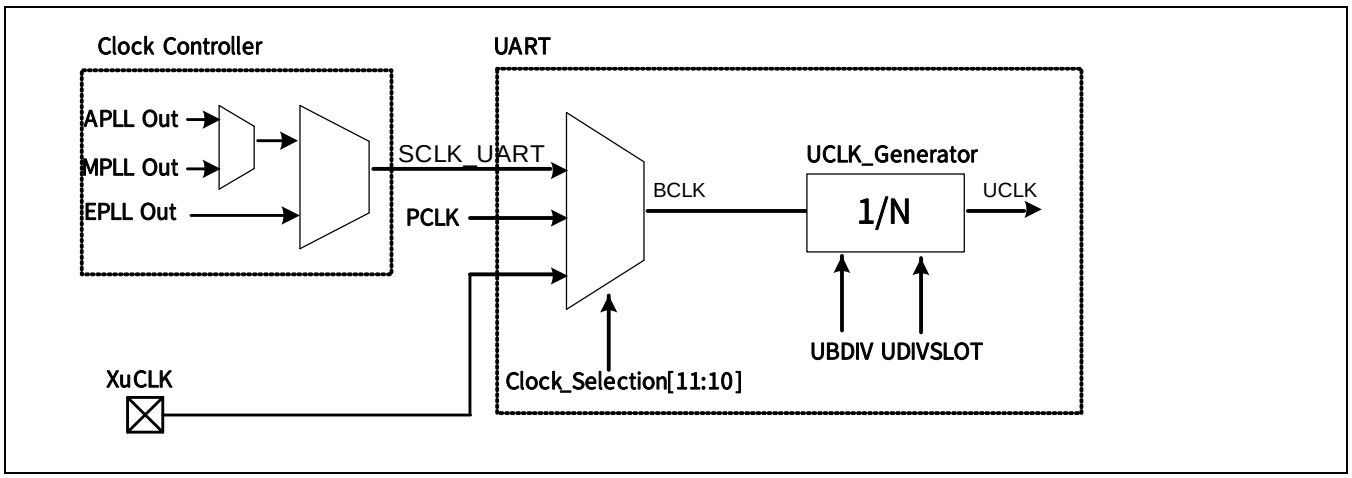


Figure 8.1-8 Input Clock Diagram for UART

S5PC100X provides UART with a variety of clock. As described in the Figure 8.1-, UART is able to select clock from PCLK, XuCLK which come from PAD or SCLK_UART which is from clock controller. We can also select SCLK_UART among PLLs. To select SCLK_UART, please refer to the Clock Controller.

4.1 SETTING SEQUENCE OF SPECIAL FUNCTION REGISTER

Special Function Register should be set as the following sequence.

1. Set Line control register(ULCON#) to set a frame format.
2. Set Control register(UCON#) without Transmit mode bits and Receive mode bits.
3. Set 1'b1 on TX FIFO Reset bit and RX FIFO Reset bit of FIFO control register(UFCON) to reset TX FIFO and RX FIFO.
4. Set FIFO control register(UFCON#) to set Triger Levels and Enable TX FIFO and RX FIFO
5. Set Modem control register(UMCON#).
6. Set Baud rate divisior register(UBRDIV#) and Dividing slot register(UDIVSLOT#) to set BAUD rate.

Choose CPU mode or DMA mode and then follow below sequence.

- In case of CPU mode

8. Set Transmit mode and Receive mode bits to 2'b01 of Control register(UCON#) to enable interrupt mode
9. Put FIFO.

- In case of DMA mode

8. Set Transmit mode and Receive mode bits of Control register(UCON#) to 2'b10 or 2'b11 to enable dma mode.
9. Turn on DMA

5 I/O DESCRIPTION

Function Signal	I/O	Descrioption	Pad	Type
UART0_RXD	Input	Receive Data for UART0	XuRXD[0]	muxed
UART0_TXD	Output	Transmit Data for UART0	XuTXD[0]	muxed
UART0_CTSn	Input	Clear to Send(active low) for UART0	XuCTSn[0]	muxed
UART0_RTSn	Output	Request to Send(active low) for UART0	XuRTSn[0]	muxed
UART1_RXD	Input	Receive Data for UART1	XuRXD[1]	muxed
UART1_TXD	Output	Transmit Data for UART1	XuTXD[1]	muxed
UART1_CTSn	Input	Clear to Send(active low) for UART1	XuCTSn[1]	muxed
UART1_RTSn	Output	Request to Send(active low) for UART1	XuRTSn[1]	muxed
UART2_RXD	Input	Receive Data for UART2	XuRXD[2]	muxed
UART2_TXD	Output	Transmit Data for UART2	XuTXD[2]	muxed
UART2_CTSn	Input	Clear to Send(active low) for UART2	XuRXD[3]	muxed
UART2_RTSn	Output	Request to Send(active low) for UART2	XuTXD[3]	muxed
UART3_RXD	Input	Receive Data for UART3	XuRXD[3]	muxed
UART3_TXD	Output	Transmit Data for UART3	XuTXD[3]	muxed
UARTCLK	Input	External UART clock	XuCLK	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals. UART external pads are shared with IrDA. In order to use these pads, GPIO must be set before the start of UART. Please refer to Chapter GPIO for exact settings

6 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
ULCON0	0xEC00_0000	R/W	UART Channel 0 Line Control Register	0x00
UCON0	0xEC00_0004	R/W	UART Channel 0 Control Register	0x00
UFCON0	0xEC00_0008	R/W	UART Channel 0 FIFO Control Register	0x0
UMCON0	0xEC00_000C	R/W	UART Channel 0 Modem Control Register	0x0
UTRSTAT0	0xEC00_0010	R	UART Channel 0 Tx/Rx Status Register	0x6
UERSTAT0	0xEC00_0014	R	UART Channel 0 Rx Error Status Register	0x0
UFSTAT0	0xEC00_0018	R	UART Channel 0 FIFO Status Register	0x00
UMSTAT0	0xEC00_001C	R	UART Channel 0 Modem Status Register	0x0
UTXH0	0xEC00_0020	W	UART Channel 0 Transmit Buffer Register	-
URXH0	0xEC00_0024	R	UART Channel 0 Receive Buffer Register	0x00
UBRDIV0	0xEC00_0028	R/W	UART Channel 0 Baud Rate Divisior Register	0x0000
UDIVSLOT0	0xEC00_002C	R/W	UART Channel 0 Dividing Slot Register	0x0000
UINTP0	0xEC00_0030	R/W	UART Channel 0 Interrupt Pending Register	0x0
UINTSP0	0xEC00_0034	R/W	UART Channel 0 Interrupt Source Pending Register	0x0
UINTM0	0xEC00_0038	R/W	UART Channel 0 Interrupt Mask Register	0x0
ULCON1	0xEC00_0400	R/W	UART Channel 1 Line Control Register	0x00
UCON1	0xEC00_0404	R/W	UART Channel 1 Control Register	0x00
UFCON1	0xEC00_0408	R/W	UART Channel 1 FIFO Control Register	0x0
UMCON1	0xEC00_040C	R/W	UART Channel 1 Modem Control Register	0x0
UTRSTAT1	0xEC00_0410	R	UART Channel 1 Tx/Rx Status Register	0x6
UERSTAT1	0xEC00_0414	R	UART Channel 1 Rx Error Status Register	0x0
UFSTAT1	0xEC00_0418	R	UART Channel 1 FIFO Status Register	0x00
UMSTAT1	0xEC00_041C	R	UART Channel 1 Modem Status Register	0x0
UTXH1	0xEC00_0420	W	UART Channel 1 Transmit Buffer Register	-
URXH1	0xEC00_0424	R	UART Channel 1 Receive Buffer Register	0x00
UBRDIV1	0xEC00_0428	R/W	UART Channel 1 Baud Rate Divisior Register	0x0000
UDIVSLOT1	0xEC00_042C	R/W	UART Channel 1 Dividing Slot Register	0x0000
UINTP1	0xEC00_0430	R/W	UART Channel 1 Interrupt Pending Register	0x0
UINTSP1	0xEC00_0434	R/W	UART Channel 1 Interrupt Source Pending Register	0x0
UINTM1	0xEC00_0438	R/W	UART Channel 1 Interrupt Mask Register	0x0

Register	Address	R/W	Description	Reset Value
ULCON2	0xEC00_0800	R/W	UART Channel 2 Line Control Register	0x00
UCON2	0xEC00_0804	R/W	UART Channel 2 Control Register	0x00
UFCON2	0xEC00_0808	R/W	UART Channel 2 FIFO Control Register	0x0
UMCON2	0xEC00_080C	R/W	UART Channel 2 Modem Control Register	0x0
UTRSTAT2	0xEC00_0810	R	UART Channel 2 Tx/Rx Status Register	0x6
UERSTAT2	0xEC00_0814	R	UART Channel 2 Rx Error Status Register	0x0
UFSTAT2	0xEC00_0818	R	UART Channel 2 FIFO Status Register	0x00
UMSTAT2	0xEC00_081C	R	UART Channel 2 Modem Status Register	0x0
UTXH2	0xEC00_0820	W	UART Channel 2 Transmit Buffer Register	-
URXH2	0xEC00_0824	R	UART Channel 2 Receive Buffer Register	0x00
UBRDIV2	0xEC00_0828	R/W	UART Channel 2 Baud Rate Divisior Register	0x0000
UDIVSLOT2	0xEC00_082C	R/W	UART Channel 2 Dividing Slot Register	0x0000
INTP2	0xEC00_0830	R/W	UART Channel 2 Interrupt Pending Register	0x0
UINTSP2	0xEC00_0834	R/W	UART Channel 2 Interrupt Source Pending Register	0x0
UINTM2	0xEC00_0838	R/W	UART Channel 2 Interrupt Mask Register	0x0
ULCON3	0xEC00_0C00	R/W	UART Channel 3 Line Control Register	0x00
UCON3	0xEC00_0C04	R/W	UART Channel 3 Control Register	0x00
UFCON3	0xEC00_0C08	R/W	UART Channel 3 FIFO Control Register	0x0
UMCON3	0xEC00_0C0C	R/W	UART Channel 3 Modem Control Register	0x0
UTRSTAT3	0xEC00_0C10	R	UART Channel 3 Tx/Rx Status Register	0x6
UERSTAT3	0xEC00_0C14	R	UART Channel 3 Rx Error Status Register	0x0
UFSTAT3	0xEC00_0C18	R	UART Channel 3 FIFO Status Register	0x00
UMSTAT3	0xEC00_0C1C	R	UART Channel 3 Modem Status Register	0x0
UTXH3	0xEC00_0C20	W	UART Channel 3 transmit Buffer Register	-
URXH3	0xEC00_0C24	R	UART Channel 3 receive Buffer Register	0x00
UBRDIV3	0xEC00_0C28	R/W	UART Channel 3 Baud Rate Divisior Register	0x0000
UDIVSLOT3	0xEC00_0C2C	R/W	UART Channel 3 Dividing Slot Register	0x0000
INTP3	0xEC00_0C30	R/W	UART Channel 3 Interrupt Pending Register	0x0
UINTSP3	0xEC00_0C34	R/W	UART Channel 3 Interrupt Source Pending Register	0x0
UINTM3	0xEC00_0C38	R/W	UART Channel 3 Interrupt Mask Register	0x0

6.1 UART LINE CONTROL REGISTER

- ULCON0, R/W, Address = 0xEC00_0000
- ULCON1, R/W, Address = 0xEC00_0400
- ULCON2, R/W, Address = 0xEC00_0800
- ULCON3, R/W, Address = 0xEC00_0C00

There are four UART line control registers namely ULCON0, ULCON1, ULCON2, and ULCON3 in the UART block.

ULCONn	Bit	Description	Reset Value
Reserved	[7]		0
Infrared Mode	[6]	Determines whether to use the Infrared mode. 0 = Normal mode operation 1 = Infrared Tx/Rx mode	0
Parity Mode	[5:3]	Specifies the type of parity generation to be performed and checking during UART transmit and receive operation. 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/ checked as 1 111 = Parity forced/ checked as 0	000
Number of Stop Bit	[2]	Specifies how many stop bits are used to signal end-of-frame signal. 0 = One stop bit per frame 1 = Two stop bit per frame	0
Word Length	[1:0]	Indicates the number of data bits to be transmitted or received per frame. 00 = 5-bit 01 = 6-bit 10 = 7-bit 11 = 8-bit	00

6.2 UART CONTROL REGISTER

- UCON0, R/W, Address = 0xEC00_0004
- UCON1, R/W, Address = 0xEC00_0404
- UCON2, R/W, Address = 0xEC00_0804
- UCON3, R/W, Address = 0xEC00_0C04

There are four UART control registers namely UCON0, UCON1, UCON2 and UCON3 in the UART block.

UCONn	Bit	Description	Reset Value
Clock Selection	[11:10]	Selects PCLK, UART_CLK (from PAD) or SCLK_UART (from Clock Controller) clock for the UART baud rate. x0 = PCLK: DIV_VAL[1)] = (PCLK / (bps x 16)) –1 01 = UART_CLK: DIV_VAL[1)] = (UART_CLK / (bps x 16)) –1 11 = SCLK_UART: DIV_VAL[1)] = (SCLK_UART / (bps x 16)) –1	0
Tx Interrupt Type	[9]	Interrupt request type.[2)] 0 = Pulse (Interrupt is requested as soon as the Tx buffer is empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.) 1 = Level (Interrupt is requested while Tx buffer is empty in Non-FIFO mode or reaches Tx FIFO Trigger Level in FIFO mode.)	0
Rx Interrupt Type	[8]	Interrupt request type. [2)] 0 = Pulse (Interrupt is requested the instant Rx buffer receives the data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.) 1 = Level (Interrupt is requested while Rx buffer is receiving data in Non-FIFO mode or reaches Rx FIFO Trigger Level in FIFO mode.)	0
Rx Time Out Enable	[7]	Enables/ Disables Rx time-out interrupts if UART FIFO is enabled. The interrupt is a receive interrupt. 0 = Disable 1 = Enable	0
Rx Error Status Interrupt Enable	[6]	Enables the UART to generate an interrupt upon an exception, such as a break, frame error, parity error, or overrun error during a receive operation. 0 = Do not generate receive error status interrupt. 1 = Generates receive error status interrupt.	0
Loop-back Mode	[5]	Setting loop-back bit to 1 trigger the UART to enter the loop-back mode. This mode is provided for test purposes only. 0 = Normal operation 1 = Loop-back mode	0
Send Break Signal	[4]	Setting this bit trigger the UART to send a break during 1 frame time. This bit is automatically cleared after sending the break signal. 0 = Normal transmit 1 = Send break signal	0
Transmit Mode	[3:2]	This bit determines which function is able to write Tx data to the UART transmit buffer register.	00

UCONn	Bit	Description	Reset Value
		00 = Disable 01 = Interrupt request or polling mode 10 = DMA request(request signal 0) 11 = DMA request(request signal 1)	
Receive Mode	[1:0]	This bit determines which function is able to read data from UART receive buffer register. 00 = Disable 01 = Interrupt request or polling mode 10 = DMA request(request signal 0) 11 = DMA request(request signal 1)	00

NOTE:

1. DIV_VAL = UBRDIVn + (num of 1's in UDIVSLOTn)/16. Refer to UART Buad Rate Configure Registers
2. RX interrupt type must be set to pulse for every transfer in S5PC100.
3. If the UART does not reach the FIFO trigger level and does not receive data during 3 word time in DMA receive mode with FIFO, the Rx interrupt is generated (receive time out), and you must check the FIFO status and read out the rest.
4. When you want to change UCLK_CLK to PCLK for UART baudrate , clock selection field must be set to 2'b00. But, when you want to change SCLK_UART to PCLK for UART baudrate , clock selection field must be set to 2'b10.

UART FIFO Control Register

- UFCON0, R/W, Address = 0xEC00_0008
- UFCON1, R/W, Address = 0xEC00_0408
- UFCON2, R/W, Address = 0xEC00_0808
- UFCON3, R/W, Address = 0xEC00_0C08

There are four UART FIFO control registers namely UFCON0, UFCON1, UFCON2 and UFCON3 in the UART block.

UFCONn	Bit	Description	Reset Value
Tx FIFO Trigger Level	[7:6]	Determines the trigger level of transmit FIFO. 00 = Empty 01 = 16-byte 10 = 32-byte 11 = 48-byte	00
Rx FIFO Trigger Level	[5:4]	Determines the trigger level of receive FIFO. 00 = 1-byte 01 = 8-byte 10 = 16-byte 11 = 32-byte	00
Reserved	[3]		0
Tx FIFO Reset	[2]	Auto-cleared after resetting FIFO 0 = Normal 1= Tx FIFO reset	0
Rx FIFO Reset	[1]	Auto-cleared after resetting FIFO 0 = Normal 1= Rx FIFO reset	0
FIFO Enable	[0]	0 = Disable 1 = Enable	0

NOTE: For using RX DMA in FIFO mode, Rx FIFO trigger level must be same value as DMA burst size. When using DMA single operation, RX FIFO trigger level must be set to 1-byte.

6.3 UART MODEM CONTROL REGISTER

- UMCON0, R/W, Address = 0xEC00_000C
- UMCON1, R/W, Address = 0xEC00_040C
- UMCON2, R/W, Address = 0xEC00_080C
- Reserved (Address = 0xEC00_0C0C)

There are three UART MODEM control registers namely UMCON0, UMCON1 and UMCON2 in the UART block.

UMCONn	Bit	Description	Reset Value
RTS trigger Level	[7:5]	If AFC bit is enabled, these bits determine when to inactivate nRTS signal. 000 = If RX FIFO contains 63 bytes. 001 = If RX FIFO contains 56 bytes. 010 = If RX FIFO contains 48 bytes. 011 = If RX FIFO contains 40 bytes. 100 = If RX FIFO contains 32 bytes. 101 = If RX FIFO contains 24 bytes. 110 = If RX FIFO contains 16 bytes. 111 = If RX FIFO contains 8 bytes.	000
Auto Flow Control (AFC)	[4]	0 = Disables 1 = Enables	0
Modem Interrupt Enable	[3]	0 = Disables 1 = Enables	0
Reserved	[2:1]	These bits must be 0	00
Request to Send	[0]	If AFC bit is enabled, this value will be ignored. In this case the S5PC100 controls nRTS automatically. If AFC bit is disabled, nRTS must be controlled by software. 0 = 'H' level (Inactivate nRTS) 1 = 'L' level (Activate nRTS)	0

NOTE: UART 2 supports AFC function, if nRxD3 and nTxD3 are set as nRTS2 and nCTS2 by GPA1CON. UART 3 does not support AFC function, because the S5PC100 has no nRTS3 and nCTS3.

6.4 UART TX/RX STATUS REGISTER

- UTRSTAT0, R, Address = 0xEC00_0010
- UTRSTAT1, R, Address = 0xEC00_0410
- UTRSTAT2, R, Address = 0xEC00_0810
- UTRSTAT3, R, Address = 0xEC00_0C10

There are four UART Tx/Rx status registers namely UTRSTAT0, UTRSTAT1, UTRSTAT2 and UTRSTAT3 in the UART block.

UTRSTATn	Bit	Description	Reset Value
Transmitter empty	[2]	This bit is automatically set to 1 if the transmit buffer register has no valid data to transmit, and the transmit shift register is empty. 0 = Not empty 1 = Transmitter (transmit buffer & shifter register) empty	1
Transmit buffer empty	[1]	This bit is automatically set to 1 if transmit buffer register is empty. 0 = Buffer register is not empty 1 = Buffer register is empty (In Non-FIFO mode, Interrupt or DMA is requested. In FIFO mode, Interrupt or DMA is requested, if Tx FIFO Trigger Level is set to 00 (Empty)) If UART uses FIFO, check Tx FIFO Count bits and Tx FIFO Full bit in UFSTAT register instead of this bit.	1
Receive buffer data ready	[0]	This bit is automatically set to 1 if receive buffer register contains valid data, received over the RXDn port. 0 = Buffer register is empty 1 = Buffer register has a received data (In Non-FIFO mode, Interrupt or DMA is requested) If UART uses the FIFO, check Rx FIFO Count bits and Rx FIFO Full bit in UFSTAT register instead of this bit.	0

6.5 UART ERROR STATUS REGISTER

- UERSTAT0, R, Address = 0xEC00_0014
- UERSTAT1, R, Address = 0xEC00_0414
- UERSTAT2, R, Address = 0xEC00_0814
- UERSTAT3, R, Address = 0xEC00_0C14

There are four UART Rx error status registers namely UERSTAT0, UERSTAT1, UERSTAT2 and UERSTAT3 in the UART block.

UERSTATn	Bit	Description	Reset Value
Break Detect	[3]	This bit is automatically set to 1 to indicate that a break signal has been received. 0 = No break received 1 = Break received (Interrupt is requested.)	0
Frame Error	[2]	This bit is automatically set to 1 if a frame error occurs during the receive operation. 0 = No frame error during receive 1 = Frame error (Interrupt is requested.)	0
Parity Error	[1]	This bit is automatically set to 1 if a parity error occurs during the receive operation. 0 = No parity error during receive 1 = Parity error (Interrupt is requested.)	0
Overrun Error	[0]	This bit is automatically set to 1 automatically if an overrun error occurs during the receive operation. 0 = No overrun error during receive 1 = Overrun error (Interrupt is requested.)	0

NOTE: These bits (UERSATn[3:0]) are automatically cleared to 0 if UART error status register is read.

6.6 UART FIFO STATUS REGISTER

- UFSTAT0, R, Address = 0xEC00_0018
- UFSTAT1, R, Address = 0xEC00_0418
- UFSTAT2, R, Address = 0xEC00_0818
- UFSTAT3, R, Address = 0xEC00_0C18

There are four UART FIFO status registers namely UFSTAT0, UFSTAT1, UFSTAT2 and UFSTAT3 in the UART block

UFSTATn	Bit	Description	Reset Value
Reserved	[15]		0
Tx FIFO Full	[14]	This bit is automatically set to 1 if the transmitted FIFO is full during transmit operation 0 = 0-byte ≤ Tx FIFO data ≤ 63-byte 1 = Full	0
Tx FIFO Count	[13:8]	Number of data in Tx FIFO	0
Reserved	[7]		0
Rx FIFO Full	[6]	This bit is automatically set to 1 if the received FIFO is full during receive operation 0 = 0-byte ≤ Rx FIFO data ≤ 63-byte 1 = Full	0
Rx FIFO Count	[5:0]	Number of data in Rx FIFO	0

6.7 UART MODEM STATUS REGISTER

- UMSTAT0, R, Address = 0xEC00_001C
- UMSTAT1, R, Address = 0xEC00_041C
- UMSTAT2, R, Address = 0xEC00_081C
- Reserved, Address = 0xEC00_0C1C

There are three UART modem status registers namely UMSTAT0, UMSTAT1 and UMSTAT2 in the UART block.

UMSTAT0	Bit	Description	Reset Value
Reserved	[7:5]	reserved	000
Delta CTS	[4]	This bit indicates that the nCTS input to the S5PC100 has changed its state since the last time it was read by CPU. (Refer Figure 8.1-) 0 = Has not changed 1 = Has changed	0
Reserved	[3:1]	Reserved	00
Clear to Send	[0]	0 = CTS signal is not activated (nCTS pin is high) 1 = CTS signal is activated (nCTS pin is low)	0

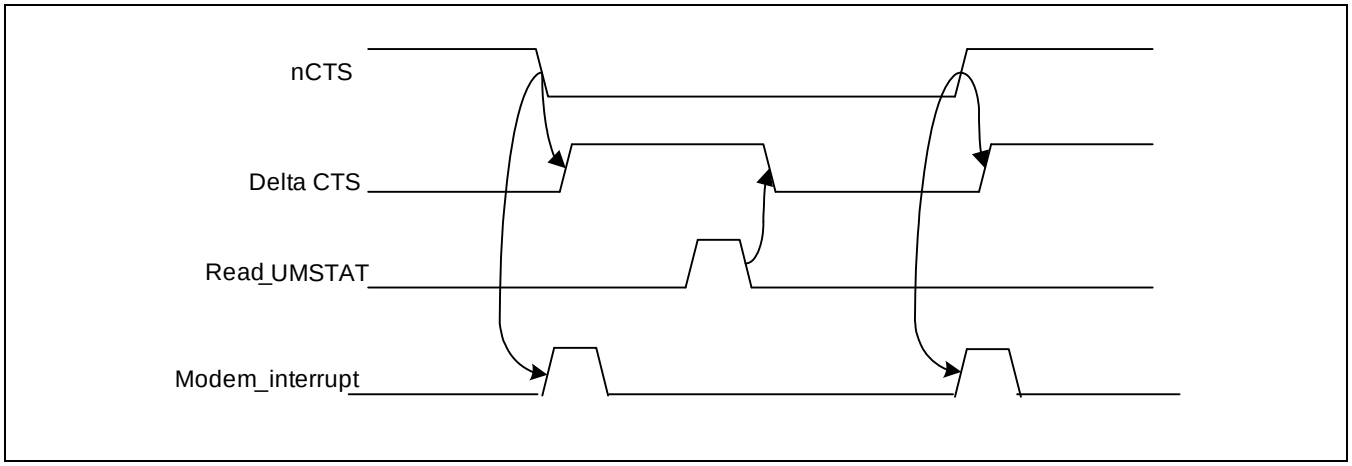


Figure 8.1-9 nCTS and Delta CTS Timing Diagram

6.8 UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

- UTXH0, W, Address = 0xEC00_0020
- UTXH1, W, Address = 0xEC00_0420
- UTXH2, W, Address = 0xEC00_0820
- UTXH3, W, Address = 0xEC00_0C20

There are four UART transmit buffer registers namely UTXH0, UTXH1, UTXH2 and UTXH3 in the UART block. UTXHn has an 8-bit data for transmission data.

UTXHn	Bit	Description	Reset Value
UTXHn	[7:0]	Transmit data for UARTn	-

6.9 UART RECIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

- URXH0, R, Address = 0xEC00_0024
- URXH1, R, Address = 0xEC00_0424
- URXH2, R, Address = 0xEC00_0824
- URXH3, R, Address = 0xEC00_0C24

There are four UART receive buffer registers namely URXH0, URXH1, URXH2 and URXH3 in the UART block. URXHn has an 8-bit data for received data.

URXHn	Bit	Description	Reset Value
URXHn	[7:0]	Receive data for UARTn	0x00

NOTE: If an overrun error occurs, the URXHn must be read. If not, the next received data makes an overrun error, even though the overrun bit of UERSTATn had been cleared.

6.10 UART CHANNEL BAUD RATE DIVISIOR REGISTER

- UBRDIV0, R/W, Address = 0xEC00_0028
- UBRDIV1, R/W, Address = 0xEC00_0428
- UBRDIV2, R/W, Address = 0xEC00_0828
- UBRDIV3, R/W, Address = 0xEC00_0C28

UBRDIV n	Bit	Description	Reset Value
UBRDIVn	[15:0]	Baud rate division value (When UART clock source is PCLK, UBRDIVn must be more than 0 (UBRDIVn >0))	-

NOTE: If UBRDIV value is 0, UART baudrate is not affected by UDIVSLOT value.

6.11 UART CHANNEL DIVIDING SLOT REGISTER

- UDIVSLOT0, R/W, Address = 0xEC00_002C
- UDIVSLOT1, R/W, Address = 0xEC00_042C
- UDIVSLOT2, R/W, Address = 0xEC00_082C
- UDIVSLOT3, R/W, Address = 0xEC00_0C2C

UDIVSLOT n	Bit	Description	Reset Value
UDIVSLOTn	[15:0]	Select the slot where clock generator divide clock source	-

NOTE:

1) UART Baud Rate Configuration

There are four UART baud rate divisor registers namely UBRDIV0, UBRDIV1, UBRDIV2 and UBRDIV3 in the UART block.

The value stored in the baud rate divisor register (UBRDIVn) and dividing slot register(UDIVSLOTn), are used to determine the serial Tx/Rx clock rate (baud rate) as follows:

DIV_VAL = UBRDIVn + (num of 1's in UDIVSLOTn)/16

DIV_VAL = (PCLK / (bps x 16)) –1

DIV_VAL = (UART_CLK / (bps x 16)) –1

or

DIV_VAL = (SCLK_UART / (bps x 16)) –1

Where, the divisor should be from 1 to (2^{16}-1).

Using UDIVSLOT, you can generate more accurate baud rate.

For example, if the baud-rate is 115200 bps and UART_CLK is 40 MHz, UBRDIVn and UDIVSLOTn are:

DIV_VAL = (40000000 / (115200 x 16)) -1
= 21.7 -1
= 20.7

UBRDIVn = 20 (integer part of DIV_VAL)

(num of 1's in UDIVSLOTn)/16 = 0.7

then, (num of 1's in UDIVSLOTn) = 11

so, UDIVSLOTn can be *16'b1110_1110_1110_1010* or *16'b0111_0111_0111_0101*, etc.

We recommend selecting UDIVSLOTn as described in the following table:

Num of 1's	UDIVSLOTn	Num of 1's	UDIVSLOTn
0	0x0000(0000_0000_0000_0000b)	8	0x5555(0101_0101_0101_0101b)
1	0x0080(0000_0000_0000_1000b)	9	0xD555(1101_0101_0101_0101b)
2	0x0808(0000_1000_0000_1000b)	10	0xD5D5(1101_0101_1101_0101b)
3	0x0888(0000_1000_1000_1000b)	11	0xDDD5(1101_1101_1101_0101b)
4	0x2222(0010_0010_0010_0010b)	12	0xDDDD(1101_1101_1101_1101b)
5	0x4924(0100_1001_0010_0100b)	13	0xDFDD(1101_1111_1101_1101b)
6	0x4A52(0100_1010_0101_0010b)	14	0xDFDF(1101_1111_1101_1111b)
7	0x54AA(0101_0100_1010_1010b)	15	0xFFDF(1111_1111_1101_1111b)

2) Baud Rate Error Tolerance

UART Frame error should be less than 1.87%(3/160)
tUPCLK = (UBRDIVn + 1) x 16 x 1Frame / (PCLK ,UART_CLK or SCLK_UART) tUPCLK: Real UART Clock
tEXTUARTCLK = 1Frame / baud-rate tEXTUARTCLK: Ideal UART Clock
UART error = (tUPCLK – tEXTUARTCLK) / tEXTUARTCLK x 100%

1Frame = start bit + data bit + parity bit + stop bit.

3) UART Clock and PCLK Relation

There is a constraint on the ratio of clock frequencies for PCLK to UARTCLK.
The frequency of UARTCLK must be no more than 5.5/3 times faster than the frequency of PCLK :

$F_{UARTCLK}$ <= 5.5/3 X F_{PCLK} **$F_{UARTCLK}$ = baudrate x 16**

This allows sufficient time to write the received data to the receive FIFO

6.12 UART INTERRUPT PENDING REGISTER

- UINTP0, R/W, Address = 0xEC00_0030
- UINTP1, R/W, Address = 0xEC00_0430
- UINTP2, R/W, Address = 0xEC00_0830
- UINTP3, R/W, Address = 0xEC00_0C30

Interrupt pending register contains the information of the interrupts that are generated.

UINTPn	Bit	Description	Reset Value
MODEM	[3]	Generates Modem interrupt.	0
TXD	[2]	Generates Transmit interrupt.	0
ERROR	[1]	Generates Error interrupt.	0
RXD	[0]	Generates Receive interrupt.	0

If one of above 4 bits is logical high('1') , each UART channel generate interrupt.

This register must be cleared in the interrupt service routine after clearing interrupt pending register in Interrupt Controller(INTC). Clear specific bits of UINTP register by writing 1's to the bits that you want to clear.

6.13 UART INTERRUPT SOURCE PENDING REGISTER

- UINTSP0, R/W, Address = 0xEC00_0034
- UINTSP1, R/W, Address = 0xEC00_0434
- UINTSP2, R/W, Address = 0xEC00_0834
- UINTSP3, R/W, Address = 0xEC00_0C34

Interrupt Source Pending Register contains the information of the interrupt that are generated regardless of the value of Interrupt Mask Register

UINTSPn	Bit	Description	Reset Value
MODEM	[3]	Generates Modem interrupt.	0
TXD	[2]	Generates Transmit interrupt.	0
ERROR	[1]	Generates Error interrupt.	0
RXD	[0]	Generates Receive interrupt.	0

6.14 UART INTERRUPT MASK REGISTER

- UINTM0, R/W, Address = 0xEC00_0038
- UINTM1, R/W, Address = 0xEC00_0438
- UINTM2, R/W, Address = 0xEC00_0838
- UINTM3, R/W, Address = 0xEC00_0C38

Interrupt mask register contains the information which interrupt source is masked. If a specific bit is set to 1, interrupt request signal to Interrupt Controller is not generated even though corresponding interrupt is generated. (**NOTE:** Even in such a case, the corresponding bit of UINTSPn register is set to 1). If the mask bit is 0, the interrupt requests are serviced from the corresponding interrupt source (**NOTE:** Even in such a case, the corresponding bit of UINTSPn register is set to 1). If the mask bit is 0, the interrupt requests are serviced.

UINTMn	Bit	Description	Reset Value
MODEM	[3]	Mask Modem interrupt.	0
TXD	[2]	Mask Transmit interrupt.	0
ERROR	[1]	Mask Error interrupt.	0
RXD	[0]	Mask Receive interrupt.	0

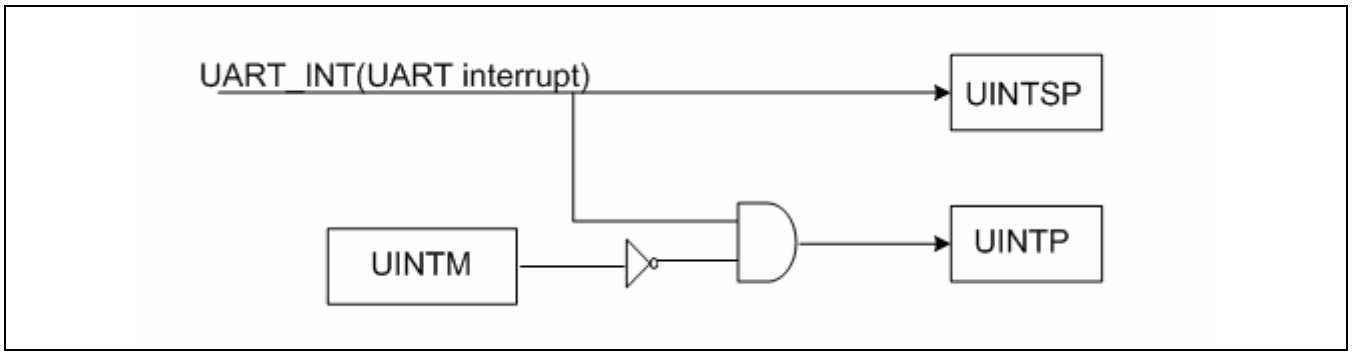


Figure 8.1-10 UINTSP, UINTP and UINTM block diagram

NOTES

8.2 I²C-BUS INTERFACE

1 OVERVIEW

The S5PC100 RISC microprocessor supports a multi-master I²C-bus serial interface. A dedicated Serial Data Line (SDA) and an Serial Clock Line (SCL) carry information between bus masters and peripheral devices which are connected to the I²C -bus. The SDA and SCL lines are bi-directional.

In the multi-master I²C-bus mode, multiple microprocessors receive or transmit serial data to or from slave devices. The master S5PC100 initiates and terminates a data transfer over the I²C-bus. The I²C-bus in the S5PC100 uses Standard bus arbitration procedure.

To control multi-master I²C-bus operations, values must be written to the following registers:

- Multi-master I²C-bus control register- I2CCON
- Multi-master I²C-bus control/status register- I2CSTAT
- Multi-master I²C-bus Tx/Rx data shift register- I2CDS
- Multi-master I²C-bus address register- I2CADD

If the I²C-bus is free, the SDA and SCL lines should be both at High level. A High-to-Low transition of SDA initiates a Start condition. A Low-to-High transition of SDA initiates a Stop condition while SCL remains steady at High Level.

The master device always generates Start and Stop conditions. A 7-bit address value in the first data byte, which is put onto the bus after the Start condition has been initiated, can determine the slave device which the bus master device has selected. The 8th bit determines the direction of the transfer (read or write).

Every data byte put onto the SDA line should be eight bits in total. There is no limit send or receive bytes during the bus transfer operation. Data is always sent from most-significant bit (MSB) first, and every byte should be immediately followed by acknowledge (ACK) bit.

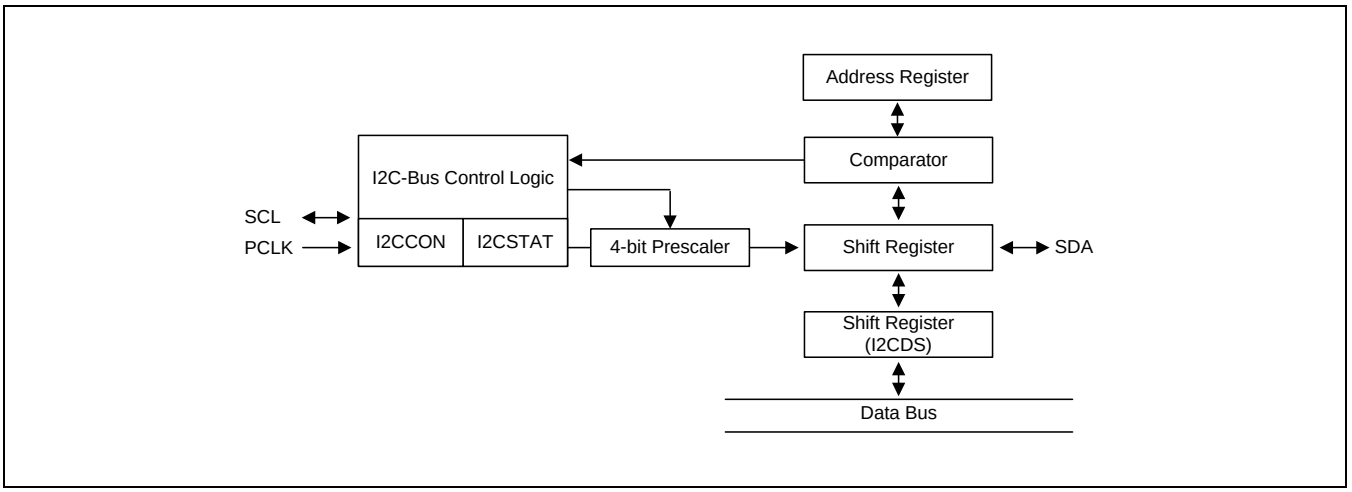


Figure 8.2-1 I²C-Bus Block Diagram

1.1 I²C-BUS INTERFACE

The S5PC100 I²C-bus interface has four operation modes:

- Master Transmitter Mode
- Master Receive Mode
- Slave Transmitter Mode
- Slave Receive Mode

Functional relationships among these operating modes are described below.

1.2 START AND STOP CONDITIONS

If the I²C-bus interface is inactive, it is usually in Slave mode. In other words, the interface should be in Slave mode before detecting a Start condition on the SDA line (a Start condition is initiated with a High-to-Low transition of the SDA line while the clock signal of SCL is High). If the interface state is changed to Master mode, a data transfer on the SDA line can be initiated and SCL signal generated.

A Start condition transfers one-byte serial data over the SDA line, and a Stop condition terminates the data transfer. A Stop condition is a Low-to-High transition of the SDA line while SCL is High. The master generates Start and Stop conditions. The I²C-bus gets busy if a Start condition is generated. A Stop condition frees the I²C-bus.

If a master initiates a Start condition, it should send a slave address to notify the slave device. One byte of address field consists of a 7-bit address and a 1-bit transfer direction indicator (Showing write or read).
If bit 8 is 0, it indicates a write operation (Transmit Operation); if bit 8 is 1, it indicates a request for data read (Receive Operation).

The master transmits Stop condition to complete the transfer operation. If the master wants to continue the data transmission to the bus, it should generate another Start condition as well as a slave address. In this way, the read-write operation is performed in various formats.

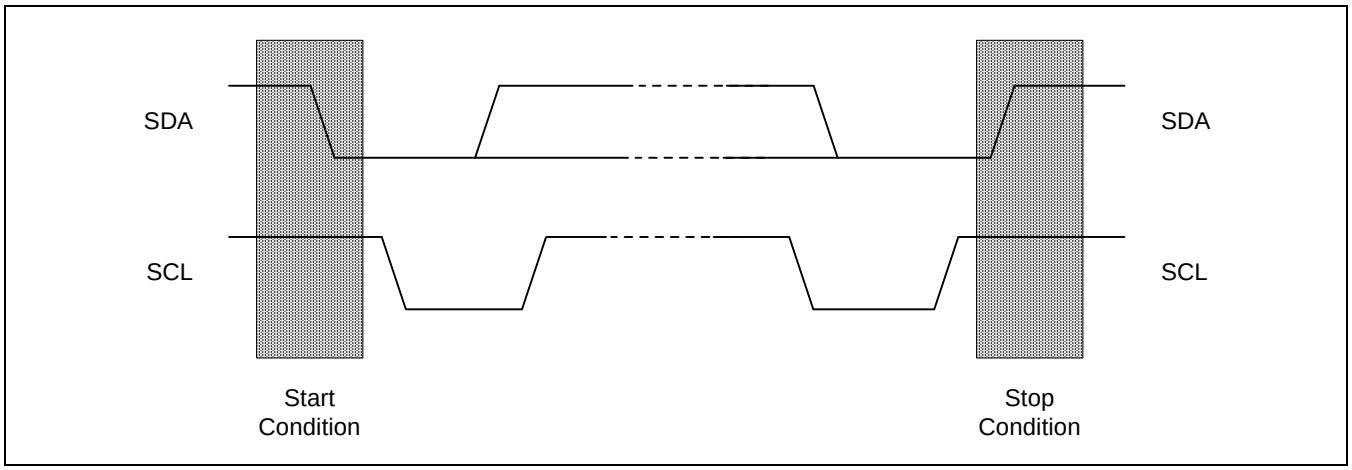


Figure 8.2-2 Start and Stop Condition

1.3 DATA TRANSFER FORMAT

Every byte placed on the SDA line should be eight bits in length. There is no limit to transmit bytes per transfer. The first byte following a Start condition should have the address field. If the I²C-bus is operating in Master mode, master transmit the address field. Each byte should be followed by an acknowledgement (ACK) bit. The MSB bit of the serial data and addresses are sent first.

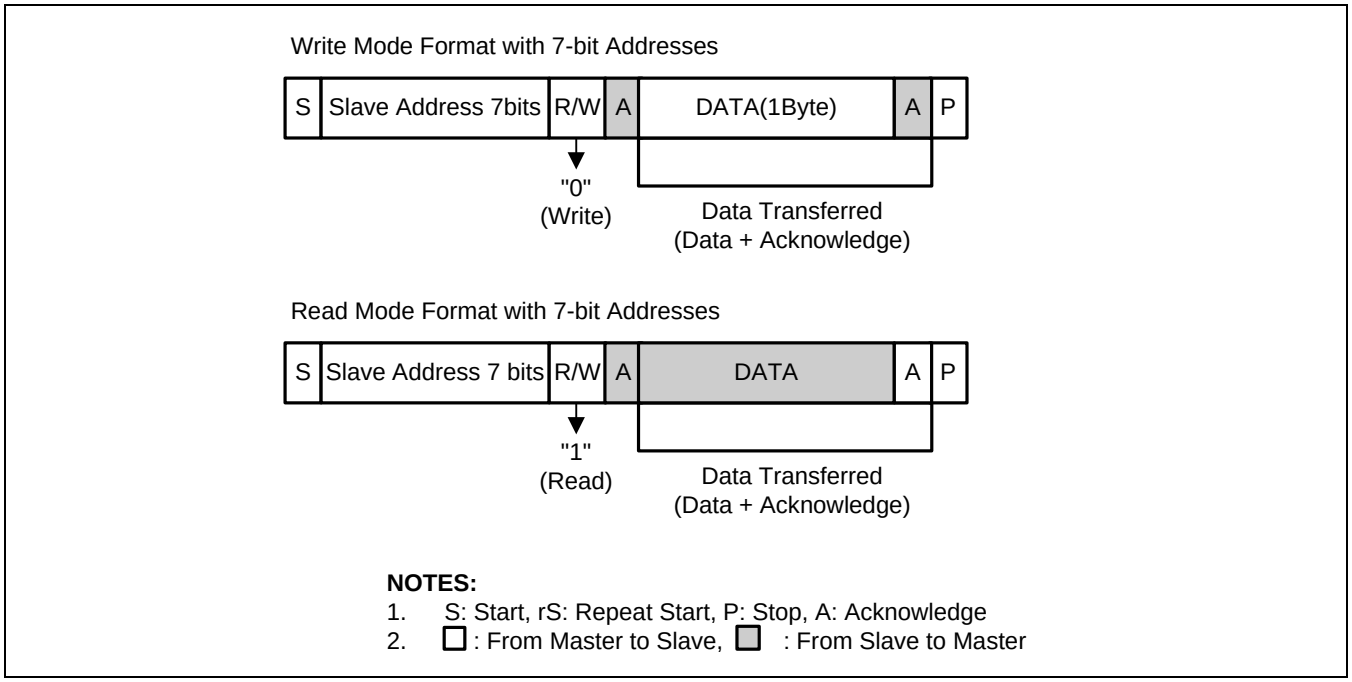


Figure 8.2-3 I²C-Bus Interface Data Format

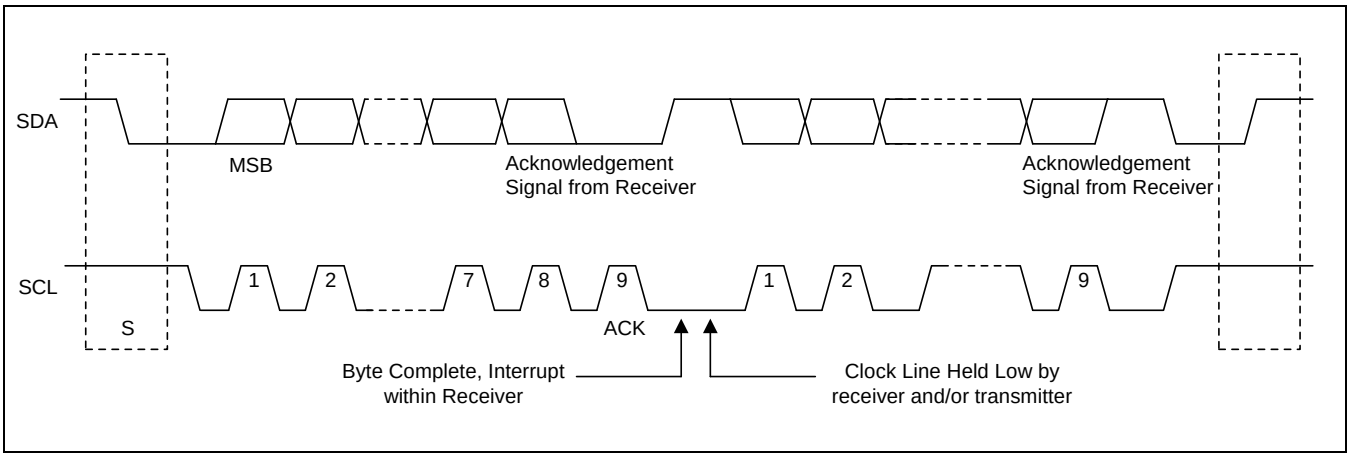


Figure 8.2-4 Data Transfer on the I²C-Bus

1.4 ACK SIGNAL TRANSMISSION

To complete a one-byte transfer operation, the receiver sends an ACK bit to the transmitter. The ACK pulse occurs at the ninth clock of the SCL line. Eight clocks are required for the one-byte data transfer. The master generates the clock pulse required to transmit the ACK bit.

The transmitter releases the SDA line by making the SDA line High if the ACK clock pulse is received. The receiver drives the SDA line Low during the ACK clock pulse so that the SDA keeps Low during the High period of the ninth SCL pulse.

The software (I2CSTAT) enables or disables ACK bit transmit function. However, the ACK pulse on the ninth clock of SCL is required to complete the one-byte data transfer operation.

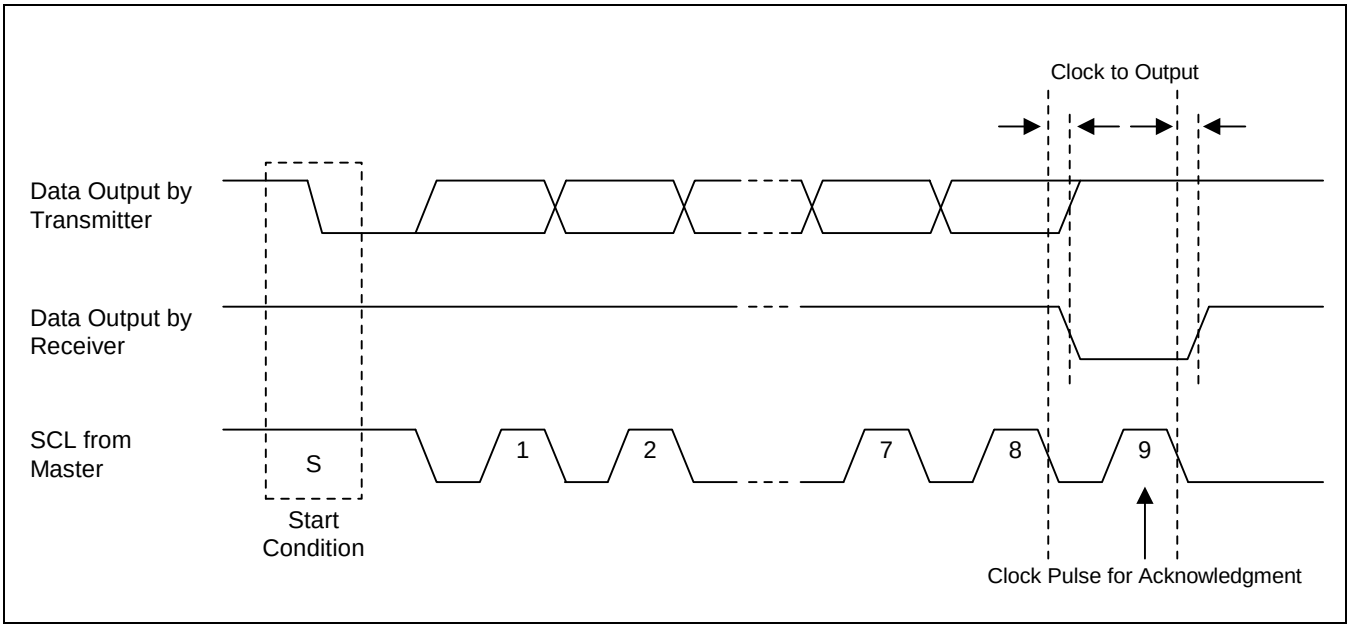


Figure 8.2-5 Acknowledge on the I²C-Bus

1.5 READ-WRITE OPERATION

In Transmitter mode, if the data is transferred, the I²C-bus interface waits until I²C-bus Data Shift (I2CDS) register receives a new data. Before the new data is written into the register, the SCL line is held low, and then released after it is written. The S5PC100 holds the interrupt to identify the completion of current data transfer. After the CPU receives the interrupt request, it writes a new data into the I2CDS register, again.

In Receive mode, if data is received, the I²C-bus interface waits until I2CDS register is read. Before the new data is read out, the SCL line is held low and then released after it is read. The S5PC100 holds the interrupt to identify the completion of the new data reception. After the CPU receives the interrupt request, it reads the data from the I2CDS register.

1.6 BUS ARBITRATION PROCEDURES

Arbitration takes place on the SDA line to prevent the contention on the bus between two masters. If a master with a SDA High level detects the other master with a SDA active Low level, it does not initiate a data transfer because the current level on the bus does not correspond to its own. The arbitration procedure extends until the SDA line turns High.

If the masters simultaneously lower the SDA line, each master evaluates whether the mastership is allocated itself or not. For the purpose of evaluation each master detects the address bits. While each master generates the slaver address, it detects the address bit on the SDA line because the SDA line is likely to get Low rather than to keep High. Assume that one master generates a Low as first address bit, while the other master is maintaining High. In this case, both masters detect Low on the bus because the Low status is superior to the High status in power. If this happens, Low (as the first bit of address) generating master gets the mastership while High (as the first bit of address) generating master withdraws the mastership. If both masters generate Low as the first bit of address, there is arbitration for the second address bit, again. This arbitration continues to the end of last address bit.

1.7 ABORT CONDITIONS

If a slave receiver cannot acknowledge the confirmation of the slave address, it holds the level of the SDA line High. In this case, the master generates a Stop condition and to abort the transfer.

If a master receiver is involved in the aborted transfer, it signals the end of the slave transmit operation by canceling the generation of an ACK after the last data byte received from the slave. The slave transmitter releases the SDA to allow a master to generate a Stop condition.

1.8 CONFIGURING I²C-BUS

To control the frequency of the serial clock (SCL), the 4-bit prescaler value is programmed in the I2CCON register. The I²C-bus interface address is stored in the I²C-bus address (I2CADD) register (By default, the I²C-bus interface address has an unknown value).

1.9 FLOWCHARTS OF OPERATIONS IN EACH MODE

The following steps must be executed before any I²C Tx/Rx operations.

1. Write own slave address on I2CADD register, if needed.
2. Set I2CCON register.
   a) Enable interrupt
   b) Define SCL period
3. Set I2CSTAT to enable Serial Output

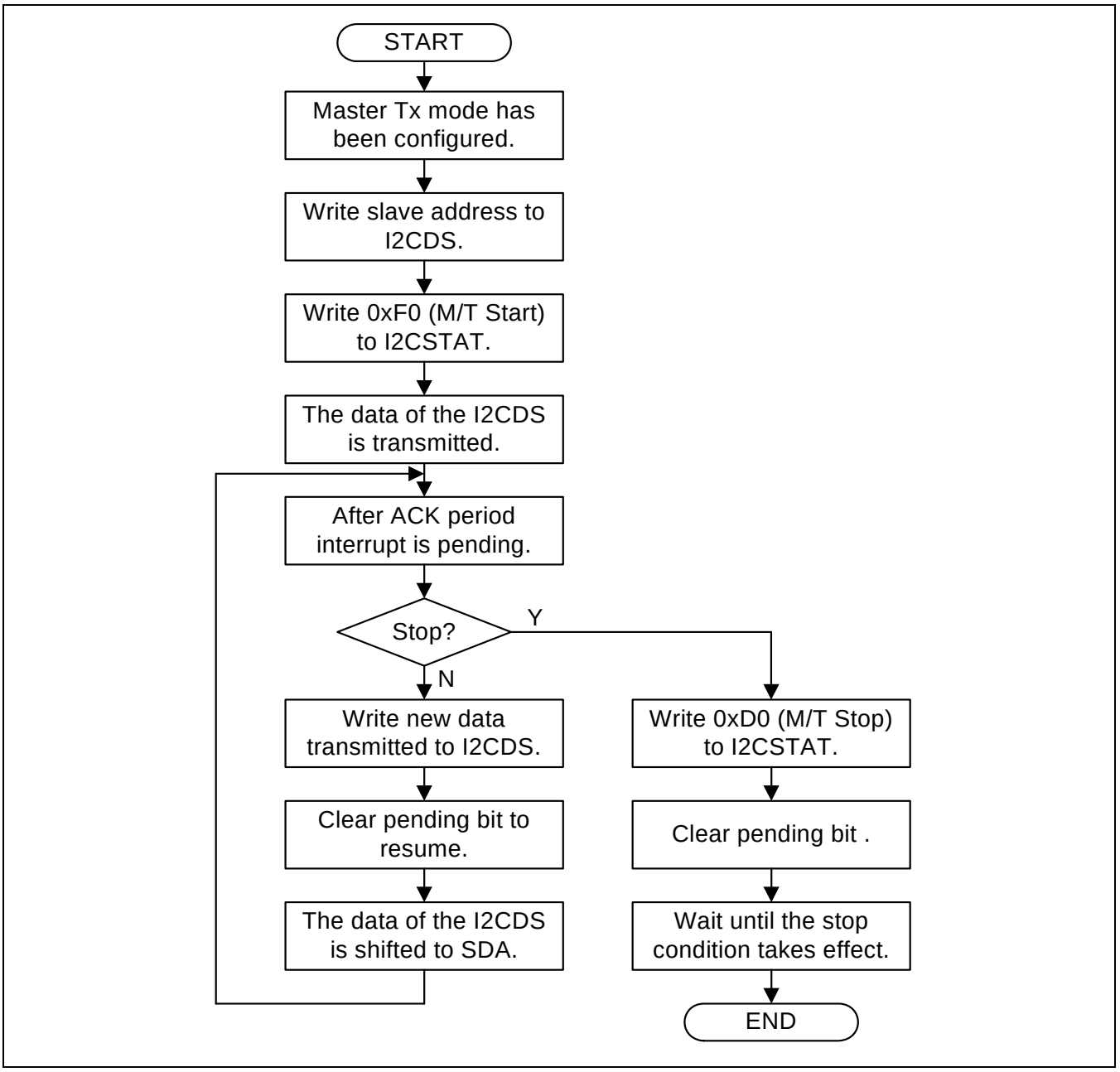


Figure 8.2-6 Operations for Master/Transmitter Mode

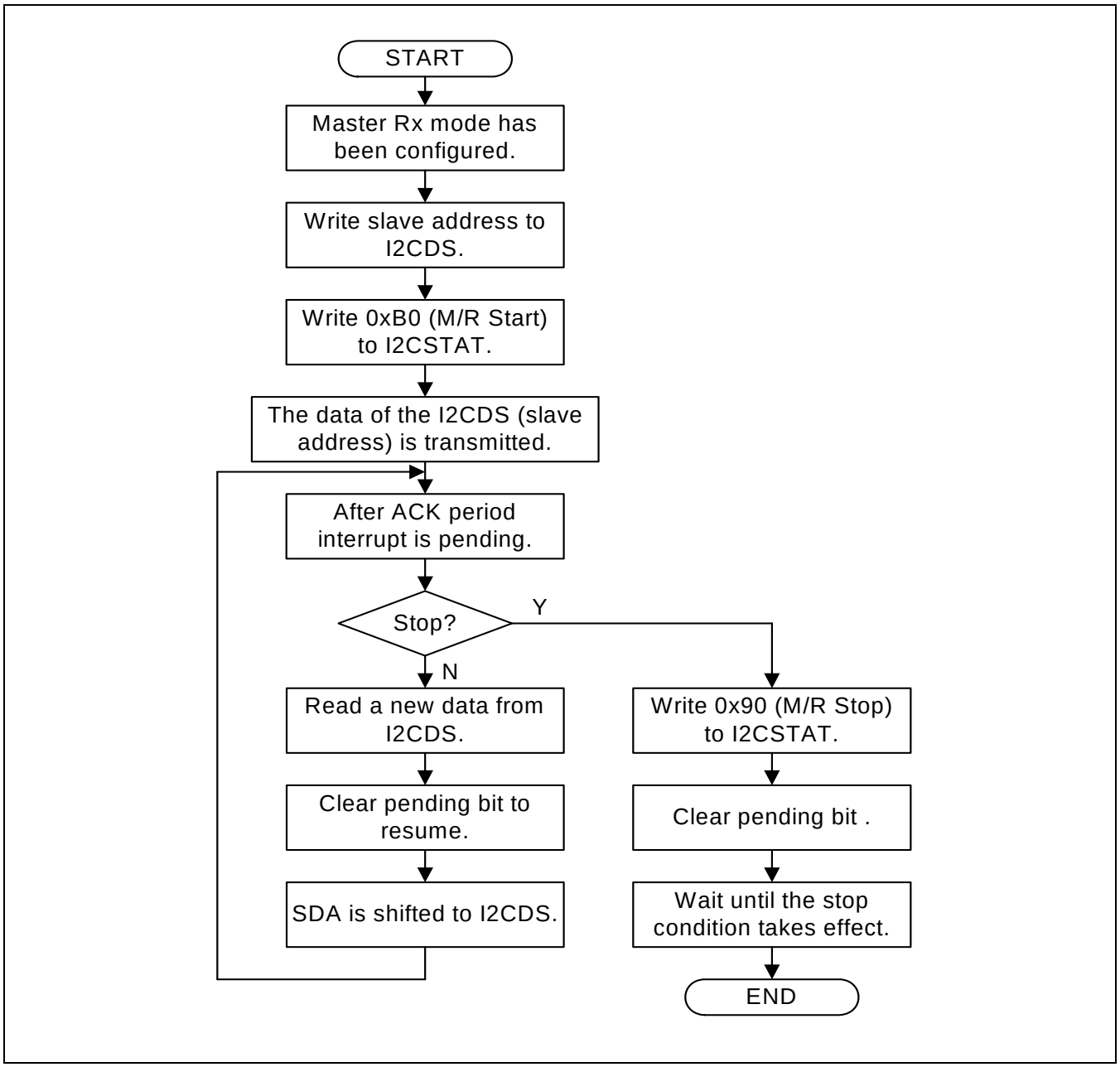


Figure 8.2-7 Operations for Master/Receiver Mode

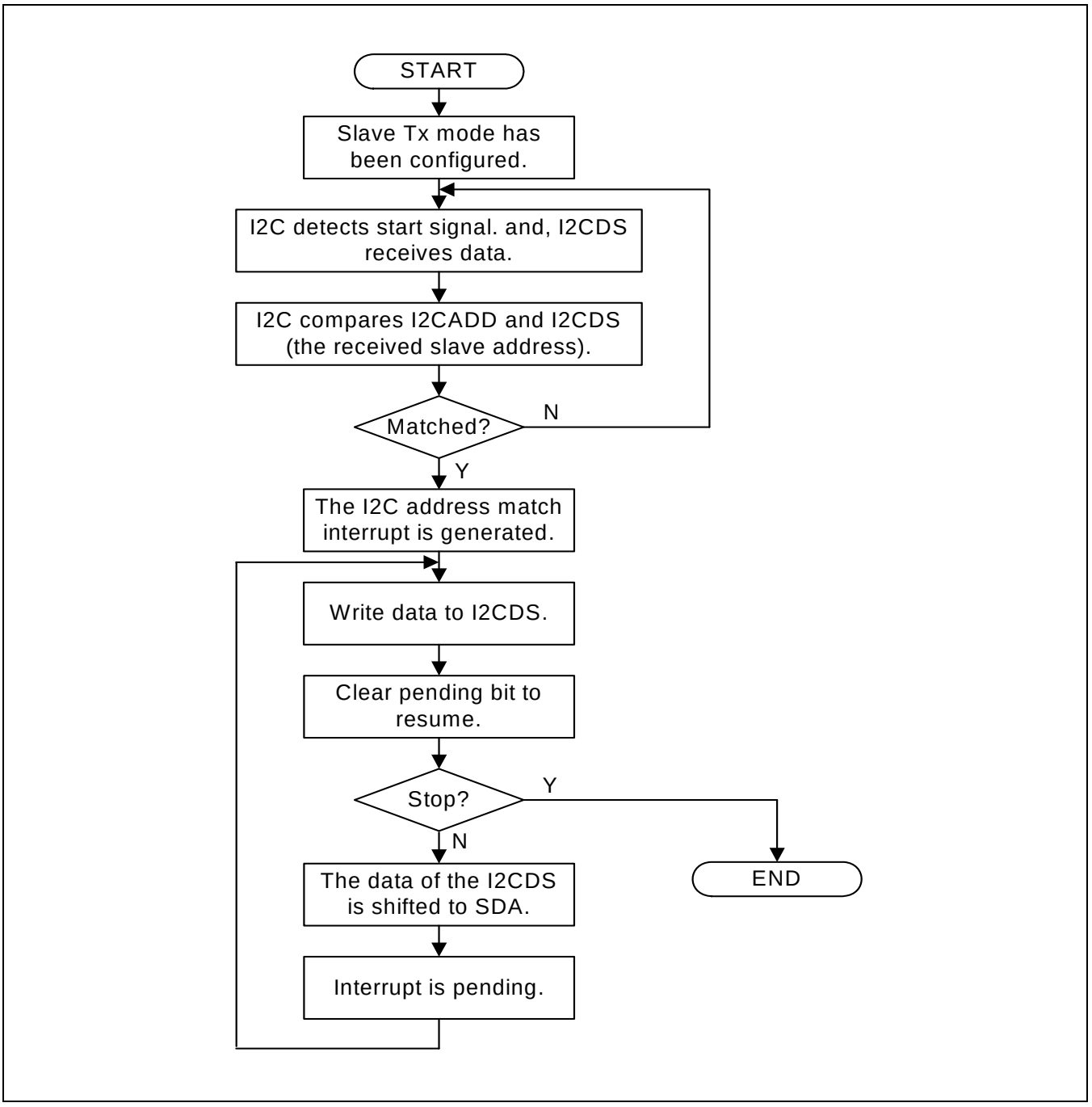


Figure 8.2-8 Operations for Slave/ Transmitter Mode

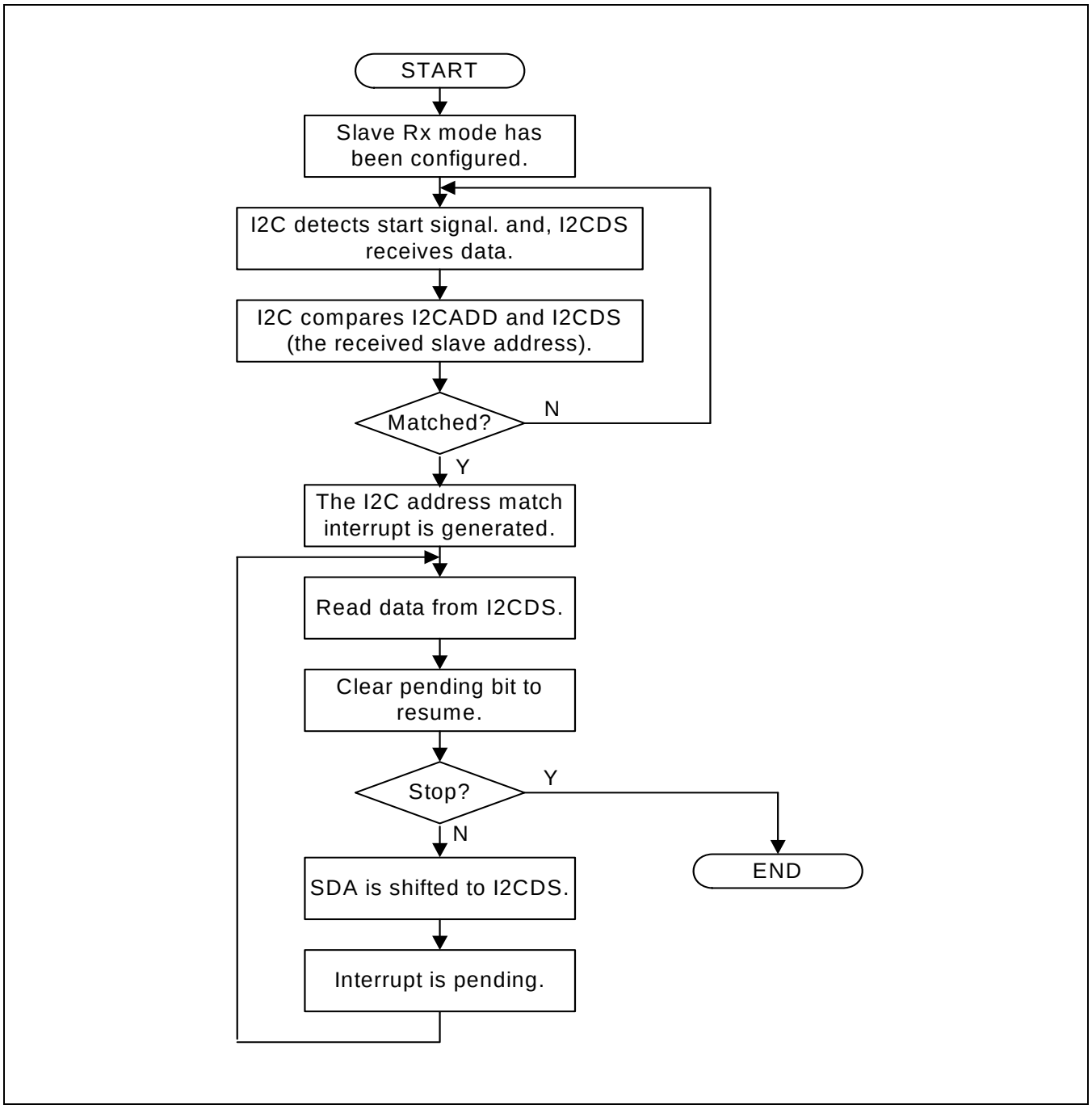


Figure 8.2-9 Operations for Slave/Receiver Mode

2 I/O DESCRIPTION

Function Signal	I/O	Descrioption	Pad	Type
I2C0_SCL	Input/Output	I²C-Bus Interface0 Serial Clock Line	Xi2c0SCL	muxed
I2C0_SDA	Input/Output	I²C-Bus Interface0 Serial Data Line	Xi2c0SDA	muxed
I2C1_SCL	Input/Output	I²C-BUS Interface1 Serial Clock Line	Xi2c1SCL	muxed
I2C1_SDA	Input/Output	I²C-BUS Interface1 Serial Data Line	Xi2c1SDA	muxed

3 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
I2CCON0	0xEC10_0000	R/W	I²C-Bus Interface0 control register	0x0X
I2CSTAT0	0xEC10_0004	R/W	I²C-Bus Interface0 control/status register	0x00
I2CADD0	0xEC10_0008	R/W	I²C-Bus Interface0address register	0xXX
I2CDS0	0xEC10_000C	R/W	I²C-Bus Interface0transmit/receive data shift register	0xXX
I2CLC0	0xEC10_0010	R/W	I²C-Bus Interface0multi-master line control register	0x00
I2CCON1	0xEC20_0000	R/W	I²C-Bus Interface1 control register	0x0X
I2CSTAT1	0xEC20_0004	R/W	I²C-Bus Interface1 control/status register	0x00
I2CADD1	0xEC20_0008	R/W	I²C-Bus Interface1 address register	0xXX
I2CDS1	0xEC20_000C	R/W	I²C-Bus Interface1 transmit/receive data shift register	0xXX
I2CLC1	0xEC20_0010	R/W	I²C-Bus Interface1 multi-master line control register	0x00

3.1 MULTI-MASTER I²C-BUS CONTROL REGISTER

- I2CCON0, R/W, Address = 0xEC10_0000
- I2CCON1, R/W, Address = 0xEC20_0000

I2CCON	Bit	Description	Reset Value
Acknowledge generation (1)	[7]	I²C-bus acknowledge enable bit. 0 = Disables 1 = Enables In Tx mode, the I2CSDA is free in the ACK time. In Rx mode, the I2CSDA is L in the ACK time.	0
Tx clock source selection	[6]	Source clock of I²C-bus transmit clock prescaler selection bit. 0 = I2CCLK = fPCLK /16 1= I2CCLK = fPCLK /512	0
Tx/Rx Interrupt (5)	[5]	I²C-Bus Tx/Rx interrupt enable/ disable bit. 0 = Disables, 1 = Enables	0
Interrupt pending flag (2) (3)	[4]	I²C-bus Tx/Rx interrupt pending flag. This bit cannot be written to 1. If this bit is read as 1, the I2CSCL is tied to L and the I²C is stopped. To resume the operation, clear this bit as 0. 0 = 1) No interrupt pending (If read). 2) Clear pending condition & Resume the operation (If write). 1 = 1) Interrupt is pending (If read) 2) N/A (If write)	0
Transmit clock value (4)	[3:0]	I²C-Bus transmit clock prescaler. I²C-Bus transmit clock frequency is determined by this 4-bit prescaler value, according to the following formula: Tx clock = I2CCLK/(I2CCON[3:0]+1).	Undefined

NOTES:

1. Interfacing with EEPROM, the ACK generation may be disabled before reading the last data in order to generate the STOP condition in Rx mode.
2. An I²C-bus interrupt occurs 1)if a 1-byte transmits or receive operation is complete. in other word, ack period is finished. 2) If a general call or a slave address match occurs, or 3) if bus arbitration fails.
3. To adjust the setup time of SDA before SCL rising edge, I2CDS has to be written before clearing the I²C interrupt pending bit.
4. I2CCLK is determined by I2CCON[6].
   Tx clock can vary by SCL transition time.
   If I2CCON[6]=0, I2CCON[3:0]=0x0 or 0x1 is not available.
5. If the I2CCON[5]=0, I2CCON[4] does not operate correctly.
   Therefore, It is recommended that you should set I2CCON[5]=1, although you does not use the I²C interrupt.

3.2 MULTI-MASTER I²C-BUS CONTROL/STATUS REGISTER

- I2CSTAT0, R/W, Address = 0xEC10_0004
- I2CSTAT1, R/W, Address = 0xEC20_0004

I2CSTAT	Bit	Description	Reset Value
Mode selection	[7:6]	I²C-bus master/ slave Tx/Rx mode select bits. 00 = Slave receive mode 01 = Slave transmit mode 10 = Master receive mode 11 = Master transmit mode	00
Busy signal status / START STOP condition	[5]	I²C-Bus busy signal status bit. 0 = read) Not busy (If read) write) STOP signal generation 1 = read) Busy (If read) write) START signal generation. The data in I2CDS is transferred automatically just after the start signal.	0
Serial output	[4]	I²C-bus data output enable/ disable bit. 0 = Disable Rx/Tx, 1 = Enable Rx/Tx	0
Arbitration status flag	[3]	I²C-bus arbitration procedure status flag bit. 0 = Bus arbitration successful 1 = Bus arbitration failed during serial I/O	0
Address-as-slave status flag	[2]	I²C-bus address-as-slave status flag bit. 0 = Cleared when START/STOP condition was detected 1 = Received slave address matches the address value in the I2CADD	0
Address zero status flag	[1]	I²C-bus address zero status flag bit. 0 = Cleared if START/ STOP condition is detected 1= Received slave address is 00000000b.	0
Last-received bit status flag	[0]	I²C-bus last-received bit status flag bit. 0 = Last-received bit is 0 (ACK was received). 1 = Last-received bit is 1 (ACK was not received).	0

3.3 MULTI-MASTER I²C-BUS ADDRESS REGISTER

- I2CADD0, R/W, Address = 0xEC10_0008
- I2CADD1, R/W, Address = 0xEC20_0008

I2CADD	Bit	Description	Reset Value
Slave address	[7:0]	7-bit slave address, latched from the I²C-bus. If serial output enable = 0 in the I2CSTAT, I2CADD is write-enabled. The I2CADD value is read any time, regardless of the current serial output enable bit (I2CSTAT) setting. Slave address : [7:1] Not mapped : [0]	Undefined

3.4 MULTI-MASTER I²C-BUS TRANSMIT/RECEIVE DATA SHIFT REGISTER

- I2CDS1, R/W, Address = 0xEC10_000C
- I2CDS1, R/W, Address = 0xEC20_000C

I2CDS	Bit	Description	Reset Value
Data shift	[7:0]	8-bit data shift register for I²C-bus Tx/Rx operation. If serial output enable = 1 in the I2CSTAT, I2CDS is write-enabled. The I2CDS value is read any time, regardless of the current serial output enable bit (I2CSTAT) setting.	Undefined

3.5 MULTI-MASTER I²C-BUS LINE CONTROL REGISTER

- I2CLC0, R/W, Address = 0xEC10_0014
- I2CLC1, R/W, Address = 0xEC20_0014

I2CLC	Bit	Description	Reset Value
Filter enable	[2]	I²C-bus filter enable bit. If SDA port is operating as input, this bit should be High. This filter prevents from occurred error by a glitch during double of PCLK time. 0 = Disables Filter 1 = Enables Filter	0
SDA output delay	[1:0]	I²C-Bus SDA line delay length selection bits. SDA line is delayed as following clock time(PCLK) 00 = 0 clocks 01 = 5 clocks 10 = 10 clocks 11 = 15 clocks	00

8.3 SPI CONTROLLER

1 OVERVIEW

The Serial Peripheral Interface (SPI) in S5PC100 transfers serial data using various peripherals. SPI includes two 8, 16, 32-bit shift registers for the purpose of transmission and receiving. During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). SPI supports the protocols for National Semiconductor Microwire and Motorola Serial Peripheral Interface.

2 FEATURES

- Full duplex
- 8/16/32-bit shift register for TX/RX
- 8-bit Prescaler Logic
- 3 clocks source
- Supports 8-bit/16-bit/32-bit bus interface
- Supports the Motorola SPI protocol and National Semiconductor Microwire
- Supports two independent transmit and receive FIFO, each 16 samples deep by 32-bit wide
- Master-mode and Slave-mode
- Receive-without-transmit operation
- Tx/Rx maximum frequency at up to 50MHz

2.1 OPERATION

The SPI transfers 1-bit serial data between S5PC100x and external device. The SPI in S5PC100x supports the CPU or DMA to transmit or receive FIFOs separately and to transfer data in both directions simultaneously. SPI has 2 channels, TX channel and RX channel. TX channel has the path from Tx FIFO to external device. RX channel has the path from external device to RX FIFO.

CPU (or DMA) must write data on the register SPI_TX_DATA, to write data in FIFO. Data on the register are automatically moved to Tx FIFOs. To read data from Rx FIFOs, CPU (or DMA) must access the register SPI_RX_DATA and data are automatically sent to the register SPI_RX_DATA.

2.1.1 Operation Mode

HS_SPI has 2 modes, master and slave mode. In master mode, HS_SPICLK is generated and transmitted to external device. XspiCS#, which is the signal to select slave, indicates data valid when it is low level. XspiCS# must be set low before packets are transmitted or received.

2.1.2 FIFO Access

The SPI in S5PC100x supports CPU access and DMA access to FIFOs. Data size of CPU access and DMA access to FIFOs are selected from 8-bit, 16-bit, or 32-bit data. If 8-bit data size is selected, valid bits are from 0 bit to 7 bit. CPU accesses are normally on and off by trigger threshold. This is user defined. The trigger level of each FIFO is set from 0byte to 64bytes. TxDMAOn or RxDMAOn bit of SPI_MODE_CFG register must be set to use DMA access. DMA access supports single transfer and 4-burst transfer. In TX FIFO, DMA request signal is high until that FIFO is full. In RX FIFO, DMA request signal is high if FIFO is not empty.

2.1.3 Trailing Bytes in the Rx FIFO

If the number of samples in Rx FIFO is less than threshold value in INT mode or DMA 4 burst mode and no additional data is received, the remaining bytes are called trailing bytes. To remove these bytes in RX FIFO, internal timer and interrupt signal are used. The value of internal timer is set up to 1024 clocks based on APB BUS clock. When timer value is to be zero, interrupt signal is occurred and CPU can remove trailing bytes in FIFO.

2.1.4 Packet Number Control

SPI controls the number of packets to be received in master mode. Set SFR (Packet_Count_reg) to receive any number of packets. SPI stops generating SPICLK if the number of packets is the same as what you had set. It is mandatory to follow software or hardware reset before this function is reloaded. (Software reset can clear all registers except special function registers, but hardware reset clears all registers.)

2.1.5 Chip Select Control

Chip select XspiCS is active low signal. In other words, a chip is selected when XspiCS input is 0.

XspiCS# can be selected auto control or manual control.

When you use manual control mode, AUTO_N_MANUAL must be cleared (Default value is 0). XspiCS# level is controlled by NSSOUT bit.

When you use audio control mode, AUTO_N_MANUAL must be set. XspiCS toggled between packet and packet automatically. Inactive period of XspiCS is controlled by NCS_TIME_COUNT. NSSOUT is not available at this time.

2.1.6 CLOCK CONTROL

Each SPI get three different clocks from Clock Controller. User can select one of them to make SPI clock by setting CLK_CFG register.

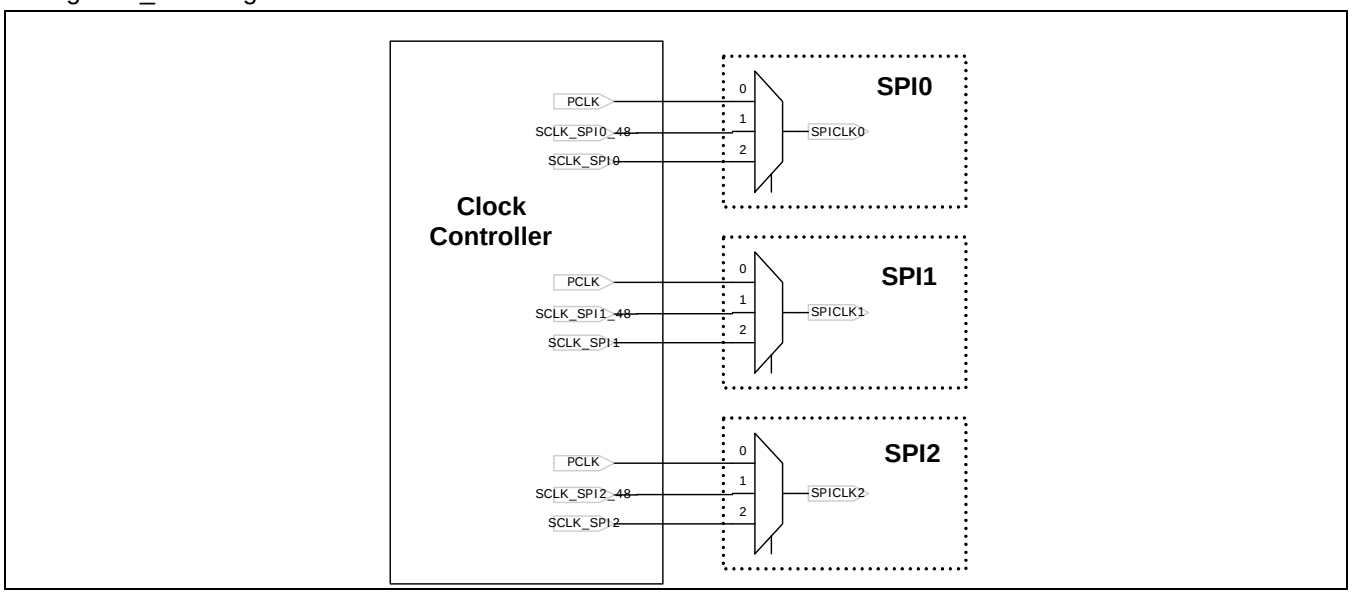


Figure 8.3-1 SPI CLOCK

2.1.7 SPI Transfer Format

The S5PC100 supports 4 different formats for data transfer. Figure 8.3-2 describes four waveforms for SPICLK.

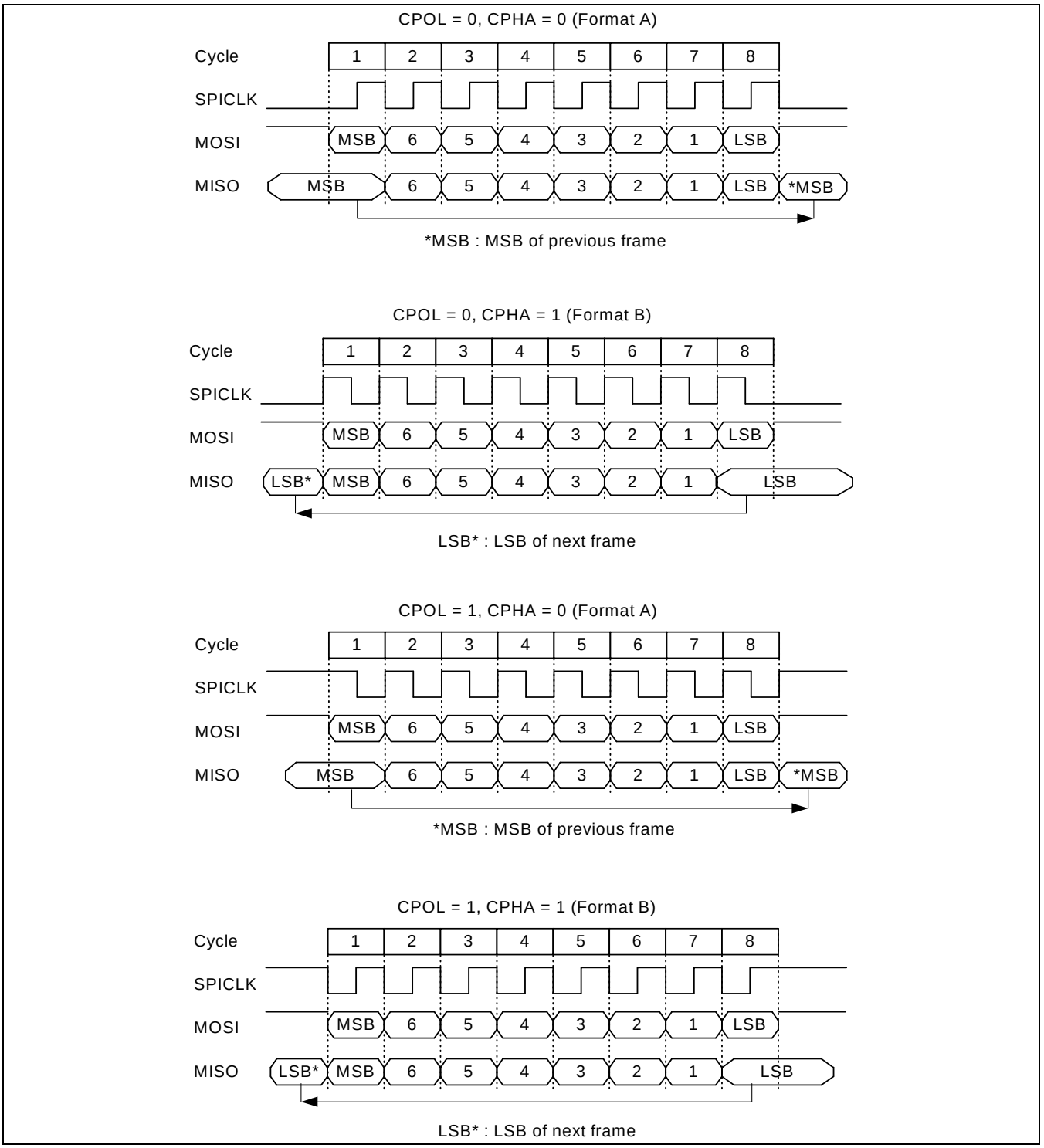


Figure 8.3-2 SPI Transfer Format

3 I/O DESCRIPTION

The following table lists the external signals between the SPI and external device. The unused SPI ports are used as General Purpose I/O ports. Refer Chapter "General Purpose I/O" for detail information.

Function Signal	I/O	Description	Pad	Type
SPI0_CLK SPI1_CLK SPI2_CLK	In/Out	XspiCLK is the serial clock used to control time of data transfer. Out, when used as master In, when used as slave	XspiCLK[0] XspiCLK[1] Xmmc2CLK	muxed
SPI0_MISO SPI1_MISO SPI2_MISO	In/Out	This port is the input port in Master mode. Input mode is used to get data from slave output port. Data are transmitted to master through this port in slave mode. Out, when used as slave In, when used as master	XspiMISO[0] XspiMISO[1] Xmmc2DATA[0]	muxed
SPI0_MOSI SPI1_MOSI SPI2_MOSI	In/Out	This port is the output port in Master mode. This port is used to transfer data from master output port. Data are received from master through this port in slave mode. Out, when used as master In, when used as slave	XspiMOSI[0] XspiMOSI[1] Xmmc2 DATA[1]	muxed
SPI0_nSS SPI1_nSS SPI2_nSS	In/Out	Slave selection signal. All data TX/RX sequences are executed if XspiCS is low. Out, when used as master In, when used as slave	XspiCSn[0] XspiCSn[1] Xmmc2CMD	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
CH_CFG(Ch0)	0xEC30_0000	R/W	SPI Configuration Register	0x40
CH_CFG(Ch1)	0xEC40_0000	R/W	SPI Configuration Register	0x40
CH_CFG(Ch2)	0xEC50_0000	R/W	SPI Configuration register	0x40
CLK_CFG(Ch0)	0xEC30_0004	R/W	Clock Configuration Register	0x0
CLK_CFG(Ch1)	0xEC40_0004	R/W	Clock Configuration Register	0x0
CLK_CFG(Ch2)	0xEC50_0004	R/W	Clock Configuration Register	0x0
MODE_CFG(Ch0)	0xEC30_0008	R/W	SPI FIFO Control Register	0x0
MODE_CFG(Ch1)	0xEC40_0008	R/W	SPI FIFO Control Register	0x0
MODE_CFG(Ch2)	0xEC50_0008	R/W	SPI FIFO Control Register	0x0
CS_REG (Ch0)	0xEC30_000C	R/W	Slave Selection Signal Control Signal	0x1
CS_REG (Ch1)	0xEC40_000C	R/W	Slave Selection Signal Control Signal	0x1
CS_REG (Ch2)	0xEC50_000C	R/W	Slave Selection Signal Control Signal	0x1
SPI_INT_EN(Ch0)	0xEC30_0010	R/W	SPI Interrupt Enable Register	0x0
SPI_INT_EN(Ch1)	0xEC40_0010	R/W	SPI Interrupt Enable Register	0x0
SPI_INT_EN(Ch2)	0xEC50_0010	R/W	SPI Interrupt Enable Register	0x0
SPI_STATUS(Ch0)	0xEC30_0014	R	SPI Status Register	0x0
SPI_STATUS(Ch1)	0xEC40_0014	R	SPI Status Register	0x0
SPI_STATUS(Ch2)	0xEC50_0014	R	SPI Status Register	0x0
SPI_TX_DATA(Ch0)	0xEC30_0018	W	SPI TX DATA Register	0x0
SPI_TX_DATA(Ch1)	0xEC40_0018	W	SPI TX DATA Register	0x0
SPI_TX_DATA(Ch2)	0xEC50_0018	W	SPI TX DATA Register	0x0
SPI_RX_DATA(Ch0)	0xEC30_001C	R	SPI RX DATA Register	0x0
SPI_RX_DATA(Ch1)	0xEC40_001C	R	SPI RX DATA Register	0x0
SPI_RX_DATA(Ch2)	0xEC50_001C	R	SPI RX DATA Register	0x0
PACKET_CNT_REG (Ch0)	0xEC30_0020	R/W	Packet Counter Register	0x0
PACKET_CNT_REG (Ch1)	0xEC40_0020	R/W	Packet Counter Register	0x0
PACKET_CNT_REG (Ch2)	0xEC50_0020	R/W	Packet Counter Register	0x0
PENDING_CLR_REG(Ch0)	0xEC30_0024	R/W	Status Pending Clear Register	0x0
PENDING_CLR_REG (Ch1)	0xEC40_0024	R/W	Status Pending Clear Register	0x0
PENDING_CLR_REG (Ch2)	0xEC50_0024	R/W	Status Pending Clear Register	0x0
SWAP_CFG(Ch0)	0xEC30_0028	R/W	SWAP Configuration register	0x0
SWAP_CFG (Ch1)	0xEC40_0028	R/W	SWAP Configuration Register	0x0
SWAP_CFG (Ch2)	0xEC50_0028	R/W	SWAP Configuration Register	0x0
FB_CLK_SEL (Ch0)	0xEC30_002C	R/W	FeedBack Clock Selection Register	0x3
FB_CLK_SEL (Ch1)	0xEC40_002C	R/W	FeedBack Clock Selection Register	0x3
FB_CLK_SEL (Ch2)	0xEC50_002C	R/W	FeedBack Clock Selection Register	0x3

4.1 PROGRAMMING GUIDE

Special Function Register should be set as the following sequence. (nCS manual mode)

1. Set Transfer Type (CPOL & CPHA set).
2. Set Clock configuration register.
3. Set SPI MODE configuration register.
4. Set SPI INT_EN register.
5. Set Packet count configuration register if necessary.
6. Set Tx or Rx Channel on.
7. Set nSSout low to start Tx or Rx operation.
   a. Set nSSout Bit to low, then start TX data writing.
   b. If auto chip selection bit is set, should not control nSSout.

4.2 DETAILED DESCRIPTION

4.2.1 SPI Configuration Register

- CH_CFG0, R/W, Address = 0xEC30_0000
- CH_CFG1, R/W, Address = 0xEC40_0000
- CH_CFG2, R/W, Address = 0xEC50_0000

CH_CFGn	Bit	Description	Reset value
HIGH_SPEED_EN	[6]	Slave TX output time control bit. If this bit is enabled in CPOL 0, Slave TX output time is reduced as much as half period of SPICLKout period. 0 = Disable 1 = Enable	1
SW_RST	[5]	Software reset. The following registers and bits are cleared by this bit. Rx/Tx FIFO Data, SPI_STATUS Once reset, this bit must be clear manually. 0 = Inactive 1= Active	0
SLAVE	[4]	Whether SPI Channel is Master or Slave 0 = Master 1 = Slave	0
CPOL	[3]	Determines whether active high or active low clock 0 = Active High 1 = Active Low	0
CPHA	[2]	Select one of the two fundamentally different transfer format 0 = Format A 1 = Format B	0
RX_CH_ON	[1]	SPI Rx Channel On 0 = Channel Off 1 = Channel On	0
TX_CH_ON	[0]	SPI Tx Channel On 0 = Channel Off 1 = Channel On	0

4.2.2 Clock Configuration Register

- CLK_CFG0, R/W, Address = 0xEC30_0004
- CLK_CFG1, R/W, Address = 0xEC40_0004
- CLK_CFG2, R/W, Address = 0xEC50_0004

CLK_CFGn	Bit	Description	Reset Value
SPI_CLKSEL	[10:9]	Clock source selection to generate SPI clock-out 00 = PCLK 01 = SCLK_SPI_48 10 = SCLK_SPI 11 = Reserved	0
ENCLK	[8]	Clock enable/ disable 0 = Disable 1 = Enable	0
SPI_SCALER	[7:0]	SPI clock-out division rate SPI clock-out = Clock source / (2 x (Prescaler value +1))	0

4.2.3 SPI FIFO Control Register

- MODE_CFG0, R/W, Address = 0xEC30_0008
- MODE_CFG1, R/W, Address = 0xEC40_0008
- MODE_CFG2, R/W, Address = 0xEC50_0008

MODE_CFGn	Bit	Description	Reset Value
CH_WIDTH	[30:29]	00 = Byte 01 = Halfword 10 = Word 11 = Reserved	0
TRAILING_CNT	[28:19]	Count value from writing the last data in RX FIFO to flush trailing bytes in FIFO	0
BUS_WIDTH	[18:17]	00 = Byte 01 = Halfword 10 = Word 11 = Reserved	0
RX_RDY_LVL	[16:11]	Rx FIFO trigger level in INT mode. Trigger level is from 0 to 63. The value means byte number in RX FIFO	0
TX_RDY_LVL	[10:5]	Tx FIFO trigger level in INT mode. Trigger level is from 0 to 63. The value means byte number in TX FIFO	0
Reserved	[4:3]	Reserved	-
RX_DMA_SW	[2]	Rx DMA mode enable/disable 0 = Disables DMA Mode 1 = Enables DMA Mode	0
TX_DMA_SW	[1]	Tx DMA mode on/off 0 = Disables DMA Mode 1 = Enables DMA Mode	0
DMA_TYPE	[0]	DMA transfer type, single or 4 busts. 0 = single 1 = 4 burst DMA transfer size must be set as the same size in SPI DMA.	0

** Channel Transfer size must be smaller than Bus Transfer size or the same as.

4.2.4 Slave Selection Signal Control Signal

- CS_REG0, R/W, Address = 0xEC30_000C
- CS_REG1, R/W, Address = 0xEC40_000C
- CS_REG2, R/W, Address = 0xEC50_000C

CS_REGn	Bit	Description	Reset Value
NCS_TIME_COUNT	[9:4]	NSSOUT inactive time = ((nCS_time_count+3)/2) x SPICLKout	0
Reserved	[3:2]	Reserved	-
AUTO_N_MANUAL	[1]	Chip select toggle manual or auto selection 0 = Manual 1 = Auto	0
NSSOUT	[0]	Slave selection signal (manual only) 0 = Active 1 = Inactive	1

4.2.5 SPI Interrupt Enable Register

- SPI_INT_EN0, R/W, Address = 0xEC30_0010
- SPI_INT_EN1, R/W, Address = 0xEC40_0010
- SPI_INT_EN2, R/W, Address = 0xEC50_0010

SPI_INT_ENn	Bit	Description	Reset Value
INT_EN_TRAILING	[6]	Interrupt Enable for trailing count to be 0 0 = Disable 1 = Enable	0
INT_EN_RX_OVERRUN	[5]	Interrupt Enable for RxOverrun 0 = Disable 1 = Enable	0
INT_EN_RX_UNDERRUN	[4]	Interrupt Enable for RxUnderrun 0 = Disable 1 = Enable	0
INT_EN_TX_OVERRUN	[3]	Interrupt Enable for TxOverrun 0 = Disable 1 = Enable	0
INT_EN_TX_UNDERRUN	[2]	Interrupt Enable for TxUnderrun. In slave mode, this bit must be clear first after turning on slave TX path. 0 = Disable 1 = Enable	0
INT_EN_RX_FIFO_RDY	[1]	Interrupt Enable for RxFifoRdy (INT mode) 0 = Disable 1 = Enable	0
INT_EN_TX_FIFO_RDY	[0]	Interrupt Enable for TxFifoRdy (INT mode) 0 = Disable 1 =Enable	0

4.2.6 SPI Status Register

- SPI_STATUS0, R, Address = 0xEC30_0014
- SPI_STATUS1, R, Address = 0xEC40_0014
- SPI_STATUS2, R, Address = 0xEC50_0014

SPI_STATUSn	Bit	Description	Reset Value
TX_DONE	[21]	Indication of transfer done in Shift register(master mode only) 0 = All case except blow case 1 = If Tx FIFO and shift register are empty	0
TRAILING_BYTE	[20]	Indication that trailing count is 0	0
RX_FIFO_LVL	[19:13]	Data level in RX FIFO 0 ~ 64 byte	0
TX_FIFO_LVL	[12:6]	Data level in TX FIFO 0 ~ 64 byte	0
RX_OVERRUN	[5]	Rx Fifo overrun error 0 = No Error, 1 = Overrun Error	0
RX_UNDERRUN	[4]	Rx Fifo underrun error 0 = No Error, 1 = Underrun Error	0
TX_OVERRUN	[3]	Tx Fifo overrun error 0 = No Error, 1 = Overrun Error	0
TX_UNDERRUN	[2]	Tx FIFO underrun error Tx FIFO underrun error is occurred if TX FIFO is empty in slave mode.(no empty state in slave Tx mode) 0 = No Error, 1 = Underrun Error	0
RX_FIFO_RDY	[1]	0 = Data in FIFO less than trigger level 1 = Data in FIFO more than trigger level	0
TX_FIFO_RDY	[0]	0 = Data in FIFO more than trigger level 1 = Data in FIFO less than trigger level	0

4.2.7 SPI TX Data Register

- SPI_TX_DATA, W, Address = 0xEC30_0018
- SPI_TX_DATA, W, Address = 0xEC40_0018
- SPI_TX_DATA, W, Address = 0xEC50_0018

SPI_TX_DATAn	Bit	Description	Reset Value
TX_DATA	[31:0]	This field contains the data to be transmitted over the SPI channel.	0

4.2.8 SPI RX Data Register

- SPI_RX_DATA0, R, Address = 0xEC30_001C
- SPI_RX_DATA1, R, Address = 0xEC40_001C
- SPI_RX_DATA2, R, Address = 0xEC50_001C

SPI_RX_DATAn	Bit	Description	Reset Value
RX_DATA	[31:0]	This field contains the data to be received over the SPI channel.	0

4.2.9 Packet Count Register

- PACKET_CNT_REG0, R/W, Address = 0xEC30_0020
- PACKET_CNT_REG1, R/W, Address = 0xEC40_0020
- PACKET_CNT_REG2, R/W, Address = 0xEC50_0020

PACKET_CNT_REGn	Bit	Description	Reset Value
PACKET_CNT_EN	[16]	Enable bit for packet count 0 = Disable 1 = Enable	0
COUNT_VALUE	[15:0]	Packet count value	0

4.2.10 Status Pending Clear Register

- PENDING_CLR_REG0, R/W, Address = 0xEC30_0024
- PENDING_CLR_REG1, R/W, Address = 0xEC40_0024
- PENDING_CLR_REG2, R/W, Address = 0xEC50_0024

PENDING_CLR_REGn	Bit	Description	Reset Value
TX_UNDERRUN_CLR	[4]	TX underrun pending clear bit 0 = Non-Clear 1 = Clear	0
TX_OVERRUN_CLR	[3]	TX overrun pending clear bit 0 = Non-Clear 1 = Clear	0
RX_UNDERRUN_CLR	[2]	RX underrun pending clear bit 0 = Non-clear 1 = Clear	0
RX_OVERRUN_CLR	[1]	RX overrun pending clear bit 0 = Non-Clear 1 = Clear	0
TRAILING_CLR	[0]	Trailing pending clear bit 0 = Non-Clear 1 = Clear	0

4.2.11 SWAP Config Register

- SWAP_CFG_REG0, R/W, Address = 0xEC30_0028
- SWAP_CFG_REG1, R/W, Address = 0xEC40_0028
- SWAP_CFG_REG2, R/W, Address = 0xEC50_0028

SWAP_CFGn	Bit	Description	Reset Value
RX_HWORD_SWAP	[7]	0 = Off 1 = Swap	0
RX_BYTE_SWAP	[6]	0 = Off 1 = Swap	0
RX_BIT_SWAP	[5]	0 = Off 1 = Swap	0
RX_SWAP_EN	[4]	Swap enable 0 = Normal 1 = Swap	0
TX_HWORD_SWAP	[3]	0 = Off 1= Swap	0
TX_BYTE_SWAP	[2]	0 = Off 1= Swap	0
TX_BIT_SWAP	[1]	0 = Off 1= Swap	0
TX_SWAP_EN	[0]	Swap enable 0 = Normal 1 = Swap	0

** Data size must be larger than swap size.

4.2.12 FeedBack Clock Selection (FB_CLK_SEL(Ch0), R/W, Address = 0xEC30_002C)

- FB_CLK_SEL_REG0, R/W, Address = 0xEC30_002C
- FB_CLK_SEL_REG1, R/W, Address = 0xEC40_002C
- FB_CLK_SEL_REG2, R/W, Address = 0xEC50_002C

FB_CLK_SELn	Bit	Description	Reset Value
FB_CLK_SEL	[1:0]	00 = 0nS additional delay 01 = 3nS additional delay 10 = 6nS additional delay 11 = 9nS additional delay * Delay based on typical condition.	0x3

4.2.13 PAD Driving Strength

PAD driving strength of SPI is controlled by setting drive strength control register in GPIO. SPI related SFR is GPBDRV (for SPI channel 0, 1) and GPG3DRV GPBDRV (for SPI channel 2).

8.4 INFRARED DATA ASSOCIATION CONTROLLER

1 OVERVIEW

The Samsung Infrared Data association (IrDA) Core is a wireless serial communication controller. It supports two different types of IrDA speed (MIR, FIR). This core transmits Infrared pulses at the high speed rate, 4 Mbps. It includes configurable FIFO feature to reduce the CPU burden. This makes it easy to adjust the internal FIFO sizes.

A user can program the core by accessing 16 internal registers. At the time of receiving the infrared pulses core detects three kinds of line errors namely:

- CRC-Error
- PHY-Error
- Payload Length Error

2 FEATURE

1. IrDA Specification compliant
   - Support IrDA 1.1 physical layer specification (4Mbps, 1.152Mpbs and 0.576Mbps)
2. Supports FIFO operation in the MIR and FIR mode
3. Configurable 64-byte FIFO size
4. Supports Back-to-Back Transactions
5. Supports software in selecting Temic-IBM or HP transceiver

3 BLOCK DIAGRAM

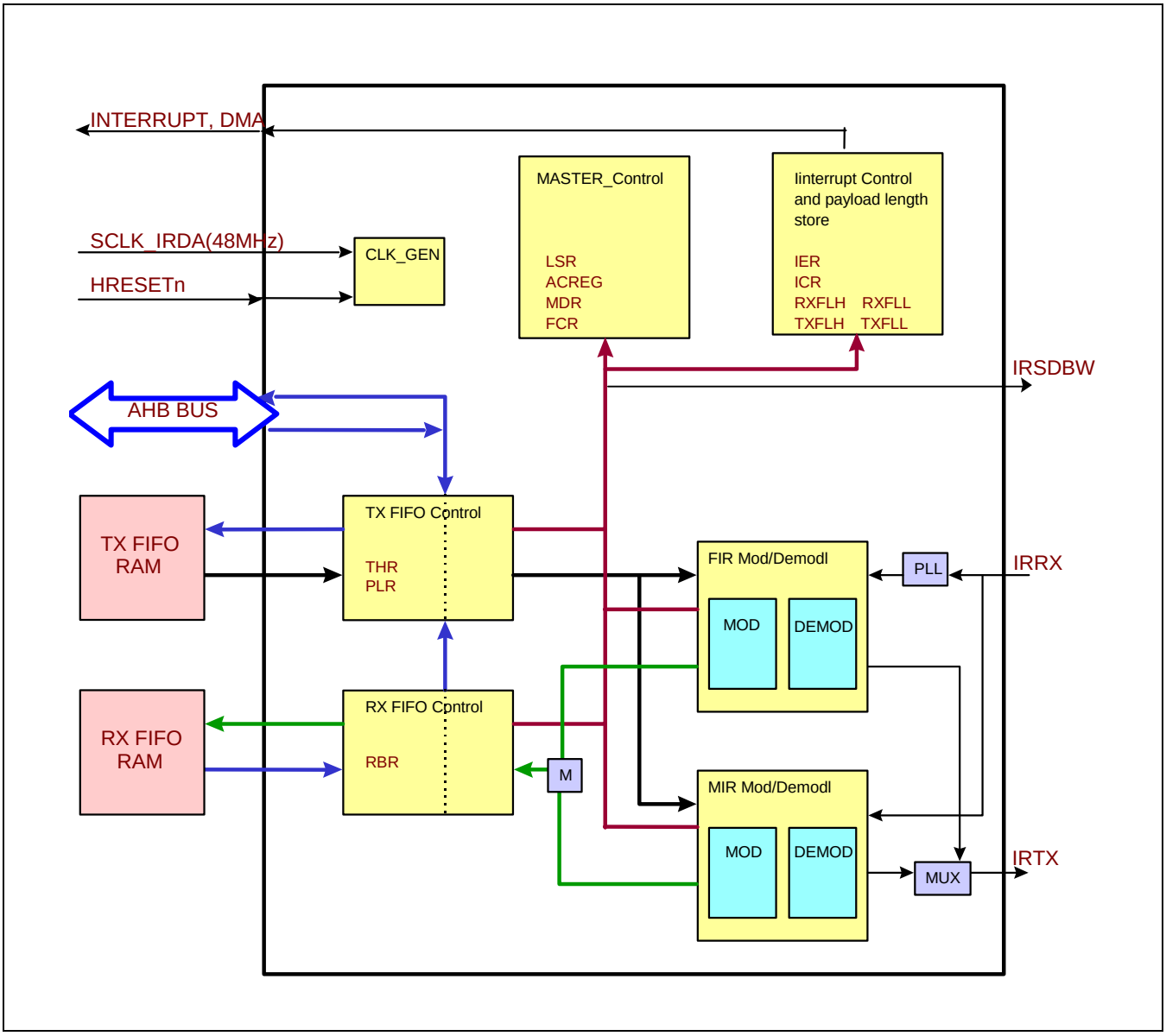


Figure 8.4-1 Block Diagram

4 FUNCTION DESCRIPTION

4.1 FAST-SPEED INFRARED (FIR) MODE (IRDA 1.1)

In this FIR mode, data communicates at the baud rate speed of 4 Mbps. In the data transmission mode, the core encodes the payload data into 4PPM format and attaches the Preamble, Start Flag, CRC-32, and Stop flag on the encoded payload and shifts them out serially. In data receive mode, the core works in reverse direction. First, if infrared pulse is detected, the core recovers receiver clock from the incoming data and removes the Preamble and Start Flag, then it extracts the payload from received 4PPM data until it meets the Stop Flag. The core detects three different kinds of errors which may occur in the middle of transmission. These are Phy-Error, Frame-Length Error and CRC error. The last one, CRC error is checked after the entire payload data is received. The micro-controller monitors the error status of the received frame by reading the Line Status Register (LSR) at the end of the frame receiving.

The below diagram shows the frame structure of the FIR data frame (The specific information of the each field is available in IrDA specification.)

Preamble	Start Flag	Link Layer Frame(Payload)	CRC32	Stop Flag

Preamble : 1000, 0000, 1010, 1000

Start Flag : 0000, 1100, 0000, 1100, 0110, 0000, 0110,0000

Stop Flag : 0000,1100, 0000, 1100, 0000, 0110, 0000, 0110

By programming the internal registers, the number of preambles is selected from 4 to 32.

* **NOTE:** 4 PPM Coding

Data Bit Pair(DBP)	4PPM Data Symbol(DD)
00	1000
01	0100
10	0010
11	0001

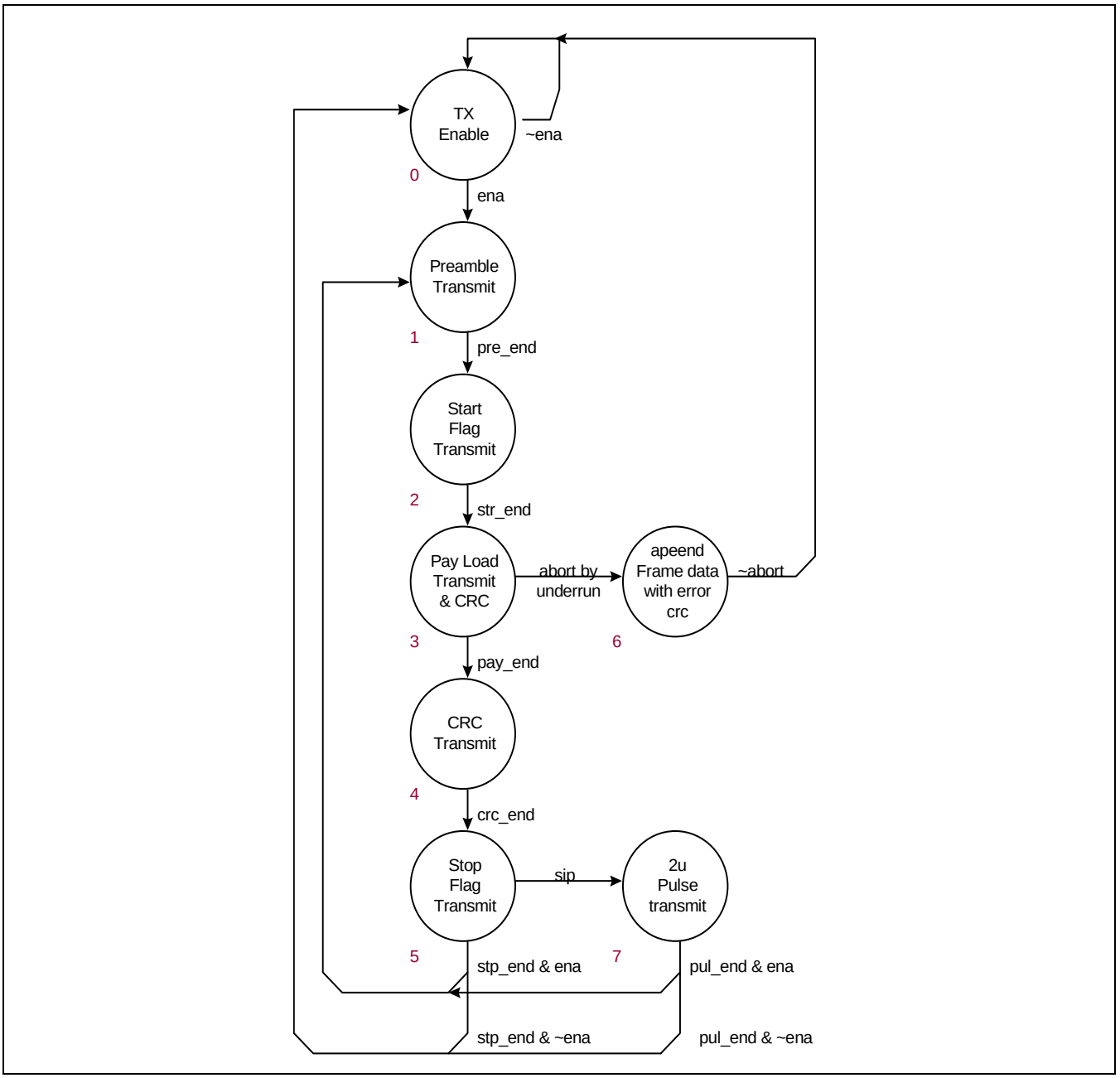


Figure 8.4-2 FIR Modulation Process

Figure 8.4-2 shows the FIR modulation state machine. The FIR transmission mode is selected by programming ACR register. If an underrun condition occurs, the state machine appends the payload with error CRC data and terminates the transmission.

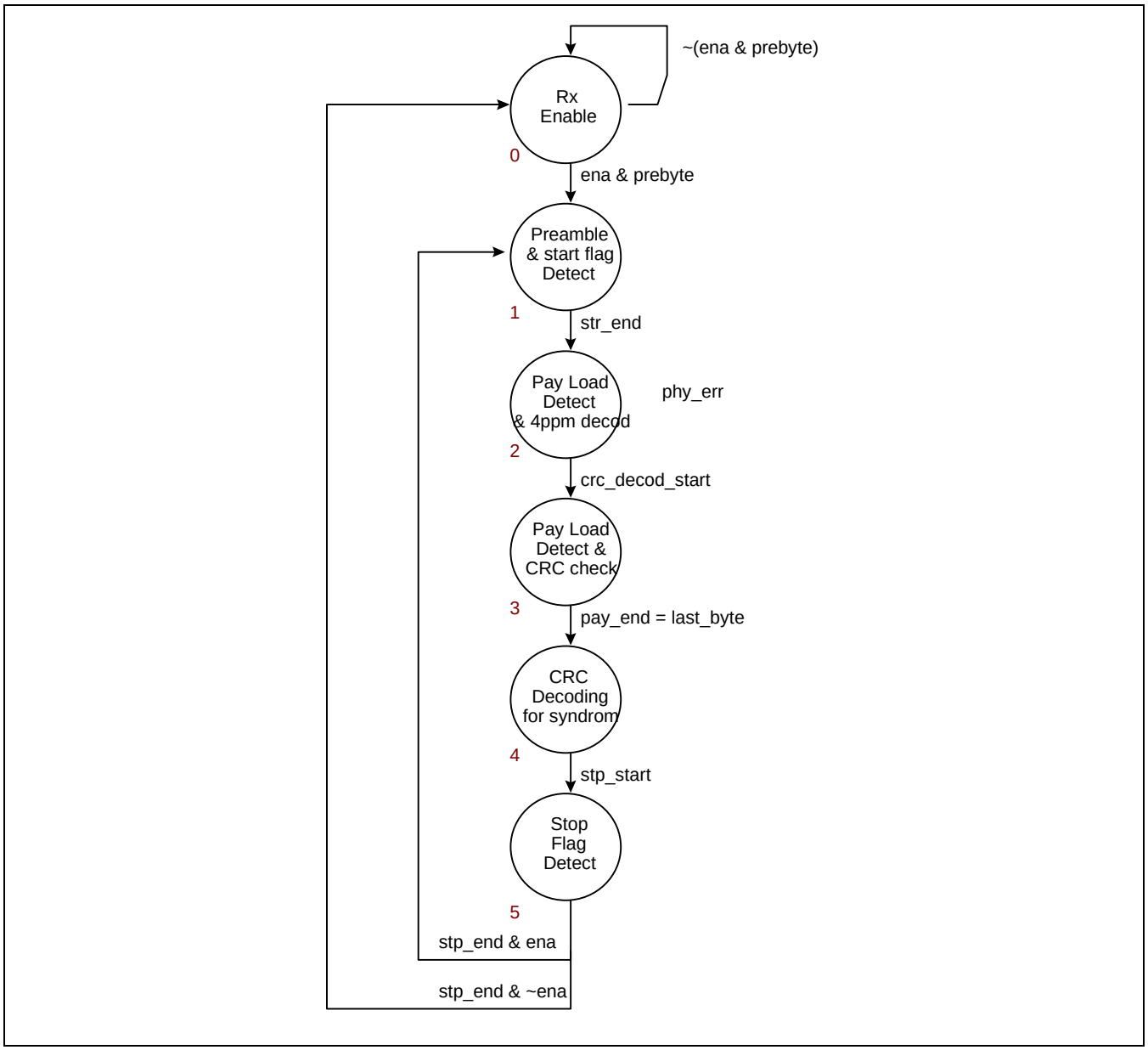


Figure 8.4-3 FIR Demodulation Process

Figure 8.4-3 shows FIR demodulation state machine. The state machine starts if ACR register bit 6 is set to logic high. The incoming data is depacketized by removing preamble and start flag and stop flag.

4PPM decoding and CRC decoding is also carried out.

4.2 MEDIUM-SPEED INFRARED (MIR) MODE (IRDA 1.1)

In MIR mode, data communicates at the speed of 1.152Mbps, and 0.576Mbps (half mode). The payload data is wrapped around by Start Flags, CRC-16, and Stop Flags. The Start Flag should be at least 2 bytes. To transmit and receive, the basic wrapping and de-wrapping processes are same as FIR mode, but, the MIR mode needs the bit-stuffing procedure. Bit stuffing in MIR mode have the core insert zero bit per every 5 consecutive ones in transmission mode. In receiving mode, the stuffed bit should be removed. Three different kinds of errors (CRC, PHY and frame length error) is reported to the microcontroller in receiving mode by reading the LSR register.

The diagram below shows the data structure of MIR frame.

STA	STA	Link layer frame (Payload)	CRC16	STO

STA : Beginning flag, 01111110 binary

CRC16 : CCITT 16-bit CRC

STO : Ending flag, 01111110 binary

The MIR pulse is modulated by 1/4 pulse format. The below diagram shows generation of pulse.

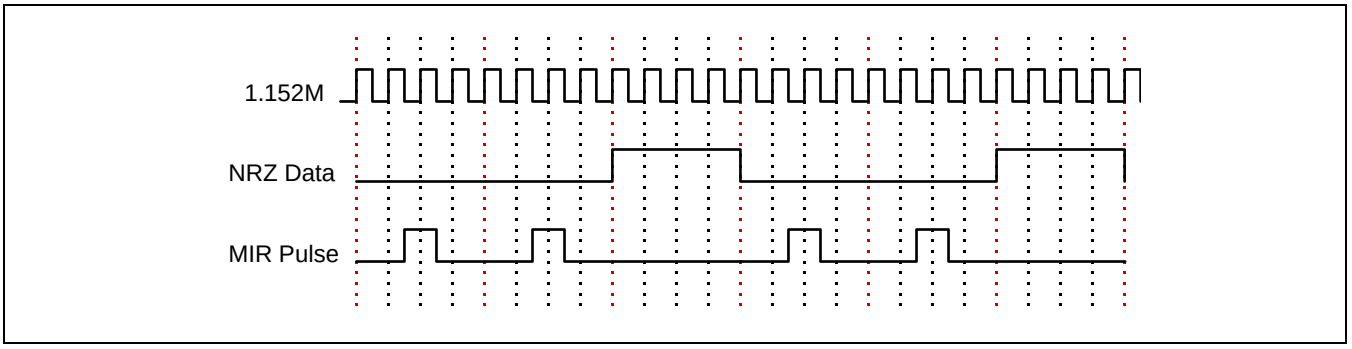


Figure 8.4-4 Pulse Modulation in MIR Mode

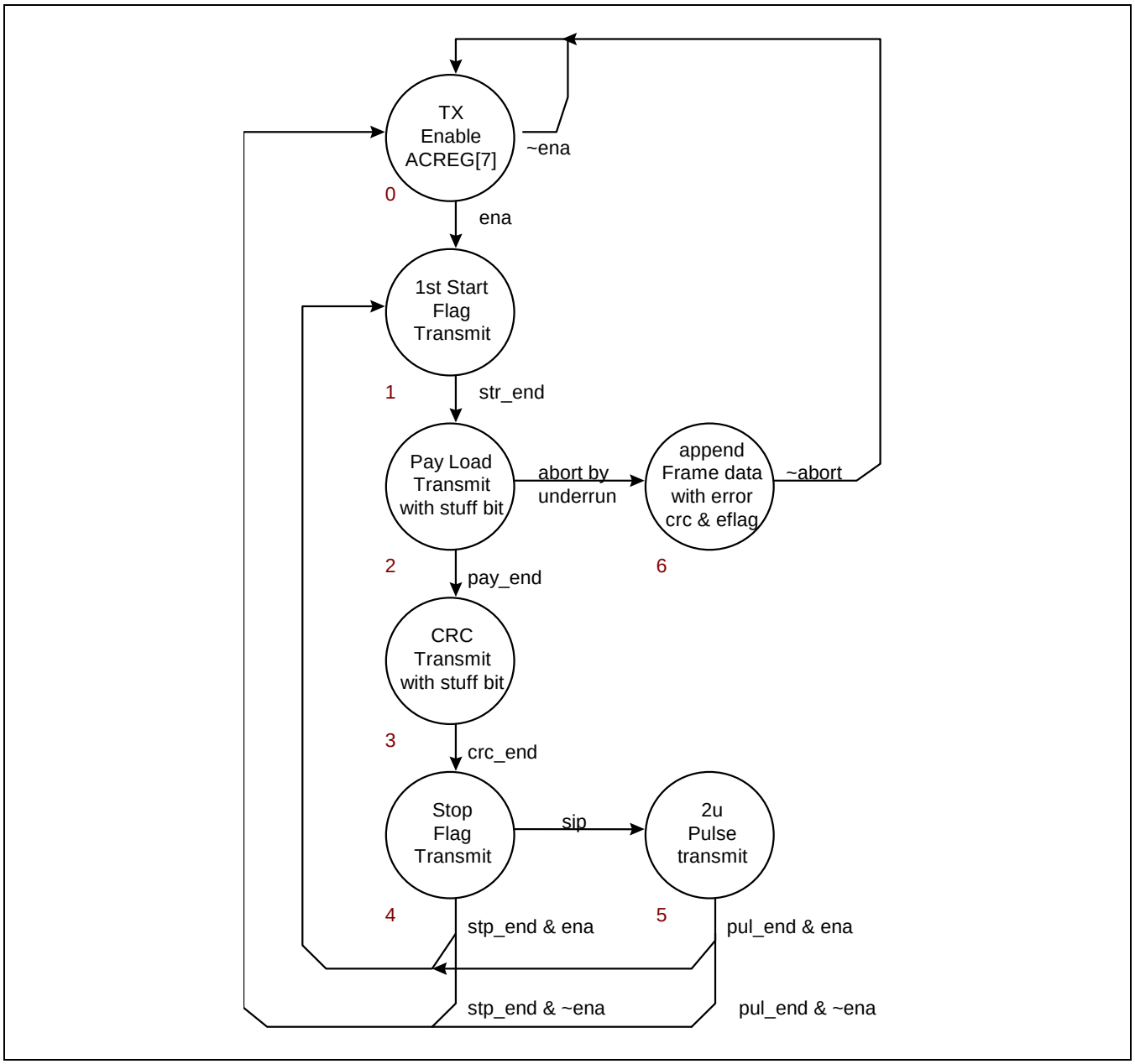


Figure 8.4-5 MIR Modulation Process.

Figure 8.4-5 shows MIR modulation state machine. This machine works very similarly with FIR modulation state machine. The major difference is that the MIR data transmission needs bit stuffing. After 5 consecutive 1, a zero data should be stuffed in MIR payload data. The state machine for this bit-stuffing is not presented here.

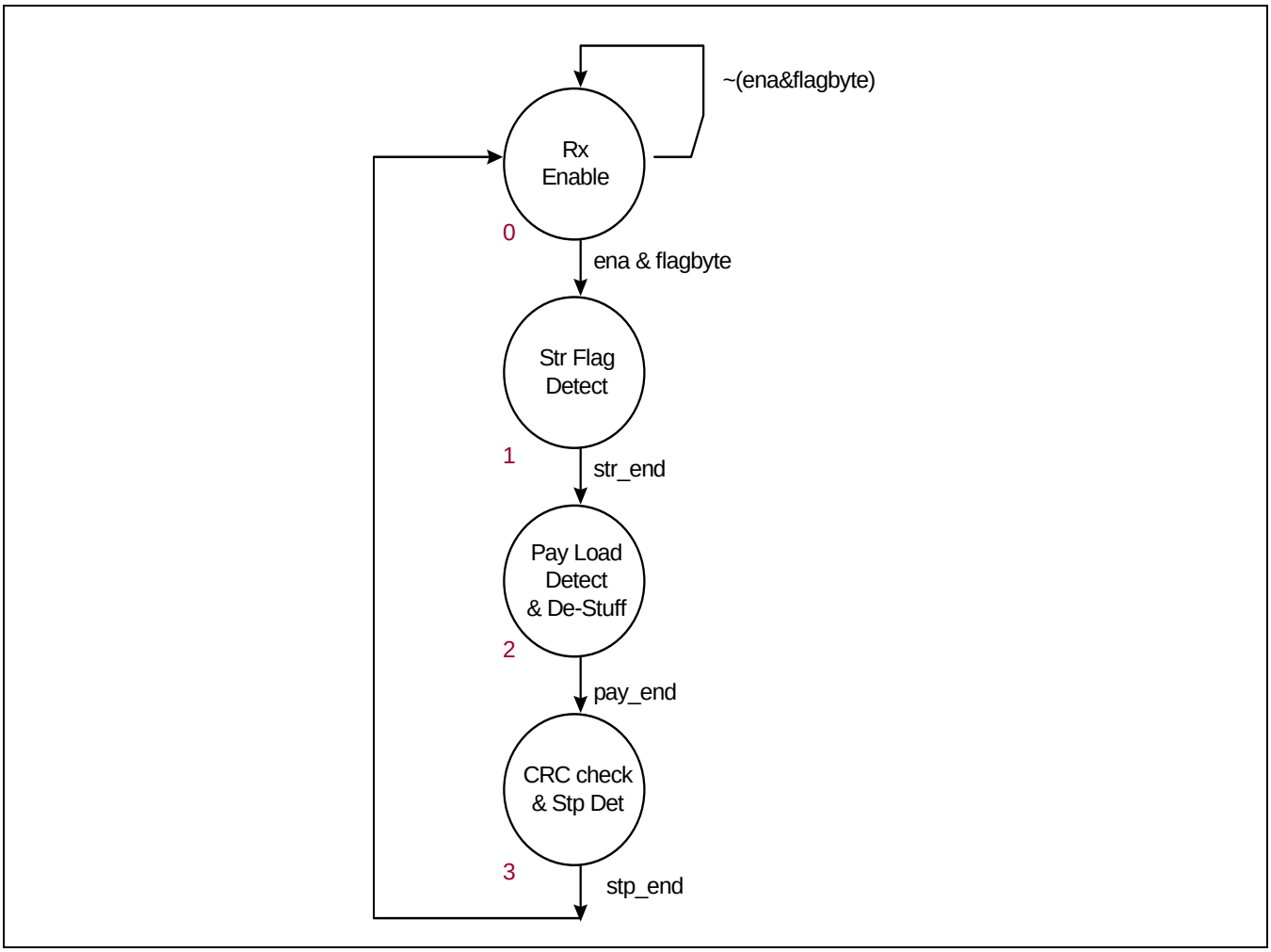


Figure 8.4-6 MIR Demodulation Process

Figure 8.4-6 shows the MIR demodulation state machine. Basically, it has similar structure with FIR demodulation state machine. Instead of having 4 PPM demodulation phase, it has the stage of removing stuffed bits from payload data stream. Since the MIR data stream does not have preamble data, the preamble/ start flag data detection stage in MIR demodulation is simplified to start flag detection state.

4.3 CORE INITIALIZATION PROCEDURE

4.3.1 MIR/FIR Mode Initialization Operation

1. Program the MDR register to select the MIR/ FIR mode.
2. Program the ACR register to select the transceiver type.
   - For the Temic-IBM type transceiver, program twice in ACR[0] = 1'b0 and ACR[0] = 1'b1.
   - For the HP type transceiver, program just once in ACR[0] = 1'b0 to FIR/MIR mode.
3. Program the PLR register to select the number of preamble or start flag, and TX threshold level.
4. Program the RXFLL and RXFLH register (maximum available receive bytes in frame).
5. Program the TXFLL and TXFLH register (transmit bytes in transmission frame).
6. Program the FCR registers (FIFO size and RX threshold level).
7. Program the IER registers (the types of interrupt).
8. Program the ACR registers (TX enable or RX enable).
9. Program the ICR register (interrupt enable).
10. Service Interrupt signal from the core.

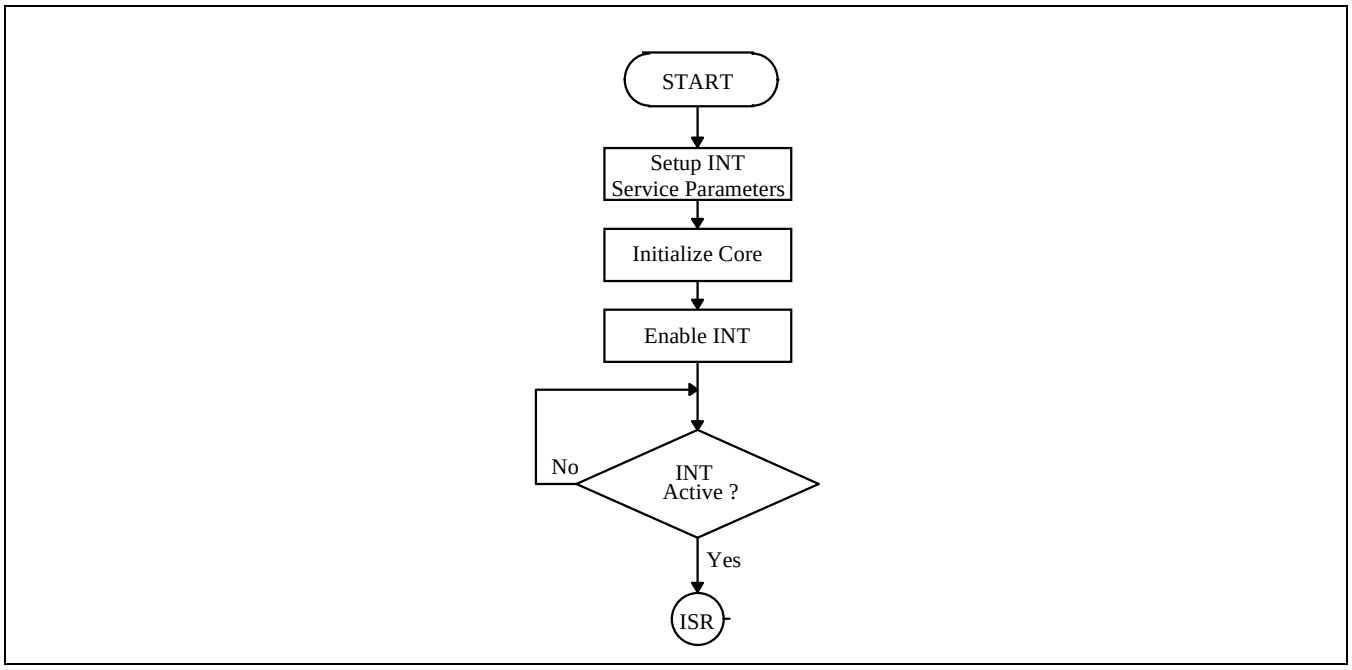


Figure 8.4-7 General Program Flowchart

5 I/O DESCRIPTION

Table 8.4-1 IrDA signals

Function Signal	I/O	Description	Pad	Type
IrDA_RXD	Input	IrDA Rx signal	XuTXD[3]	Muxed
IrDA_TXD	Output	IrDA Tx signal	XuRXD[3]	Muxed
IrDA_SDBW	Output	IrDA Transceiver control (Shutdown, Bandwidth)	XuCLK	Muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

6 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
IrDA _CNT	0xEC60_0000	R/W	IrDA Control Register	0x00
IrDA_MDR	0xEC60_0004	R/W	IrDA Mode Definition Register	0x00
IrDA_CNF	0xEC60_0008	R/W	IrDA Interrupt / DMA Configuration Register	0x00
IrDA _IER	0xEC60_000C	R/W	IrDA Interrupt Enable Register	0x00
IrDA _IIR	0xEC60_0010	R	IrDA Interrupt Identification Register	0x00
IrDA _LSR	0xEC60_0014	R	IrDA Line Status Register	0x83
IrDA _FCR	0xEC60_0018	R/W	IrDA FIFO Control Register	0x00
IrDA _PLR	0xEC60_001C	R/W	IrDA Preamble Length Register	0x12
IrDA_RBR IrDA_THR	0xEC60_0020	R/W	IrDA Receiver & Transmitter Buffer Register	0x00
IrDA _TXNO	0xEC60_0024	R	The total number of data bytes remained in Tx FIFO	0x00
IrDA _RxNO	0xEC60_0028	R	The total number of data bytes remained in Rx FIFO	0x00
IrDA _TXFLL	0xEC60_002C	R/W	IrDA Transmit Frame-Length Register Low	0x00
IrDA _TXFLH	0xEC60_0030	R/W	IrDA Transmit Frame-Length Register High	0x00
IrDA _RXFLL	0xEC60_0034	R/W	IrDA Receive Frame-Length Register Low	0x00
IrDA _RXFLH	0xEC60_0038	R/W	IrDA Receive Frame-Length Register High	0x00
IrDA _INTCLR	0xEC60_003C	W	IrDA Interrupt Clear Register	

6.1 IRDA CONTROL REGISTER(IRDA_CNT, R/W, ADDRESS = 0XEC60_0000)

IrDA _CNT	Bit	Description	Initial State
TX Enable	[7]	Enables TX. Bit 7 must be set to '1' to enable data transmission in MIR/FIR infrared modes.	0
RX Enable	[6]	Enables RX. Bit 6 must be set to '1' to enable data receive in all MIR/FIR infrared modes.	0
Core Loop	[5]	Core loop for software debugging. The IRRX port connects directly to the IRTX internally.	0
MIR Half Mode	[4]	MIR half mode. If bit 4 is set to a '1', the operating speed in the MIR mode changes from 1.152 Mbps to 0.576 Mbps.	0
Send IR Pulse	[3]	Sends 1.6-us IR pulse. If the IrDA_MDR[3] bit equals to a '1' and the CPU writes a '1' to this bit, the transmitting interface device sends a 1.6-us IR pulse at the end of the frame. Bit 3 is cleared automatically by the transmitting interface device at the end of 1.6-us IR pulse data transmission.	0
Reserved	[2]	-	0
Frame Abort	[1]	Frame abort. The CPU intentionally aborts data transmission of a frame by writing a '1' to bit 1. Neither the end flag nor the CRC bits are appended to the frame. The receiver finds the frame with the abort pattern in the MIR mode and a PHY-error in the FIR mode. The CPU must reset the TX FIFO and reset this bit by writing a '0' to bit '1' before next frame is transmitted.	0
SD/BW	[0]	This signal controls IrDA_SDBW output signal. It is used for controlling mode (shutdown, band width) of IrDA transceiver.	0

6.2 IRDA MODE DEFINITION REGISTER(IRDA_MDR, R/W, ADDRESS = 0XEC60_0004)

IrDA _MDR	Bit	Description	Initial State
Reserved	[7:5]	-	0
SIP Select	[4]	SIP select method. If this bit is set to '1' and the IrDA_CNT[3] is set to '1', the SIP pulse is appended at the end of FIR/MIR TX frame. Likewise, if this bit is set to a '0', SIP is generated at the end of the every FIR/MIR frames. If IrDA_CNT[3] is set to '0', setting this bit to '1' does not help to generate SIP. Along with IrDA_CNT[3] bit, the way of SIP generation is controlled.	0
Temic Select	[3]	Bit 3 is Temic transceiver select bit. If bit 3 is clear to "0", core automatically selects in Temic transceiver mode.	0
Reserved	[2:1]	-	00
Mode Select	[0]	Select the mode of operation as 0 : FIR Mode 1 : MIR Mode	0

6.3 IRDA INTERRUPT / DMA CONFIGURATION REGISTER (IRDA_CNF, R/W, ADDRESS = 0XEC60_0008)

IrDA _CNF	Bit	Description	Initial State
Reserved	[7:4]	-	0
DMA Enable	[3]	1 : DMA Enable	0
DMA Mode	[2]	0 : Tx DMA 1 : Rx DMA	0
Reserved	[1]	-	0
Interrupt Enable	[0]	The bit 0 enables Interrupt output signal.	0

6.4 IRDA INTERUPT ENALBLE REGISTER(IRDA_IER, R/W, ADDRESS = 0XEC60_000C)

IrDA_IER	Bit	Description	Initial State
Last Byte to Rx FIFO	[7]	Enables state indication interrupt. If Last byte write to RX FIFO.	0
Error Indication	[6]	Enables error status indication interrupt in data receiving mode.	0
Tx Underrun	[5]	Enables transmitter under-run interrupt.	0
Last Byte Detect	[4]	Detect stop-flag interrupt enable. If this bit is set to "1", an interrupt signal is activated if the last byte of the received data frame comes into the demodulation block and CRC decoding is finished.	0
Rx Overrun	[3]	Enables receiver over-run interrupt.	0
Last Byte Read from Rx FIFO	[2]	Bit 2 enables last byte from RX FIFO interrupt which is generated if the microcontroller reads the last byte of the frame from the RX FIFO.	0
Tx FIFO Below Threshold	[1]	Bit 1 enables a TX FIFO below threshold level interrupt if the available empty space in TX FIFO is over the threshold level.	0
Rx FIFO Over Threshold	[0]	Bit 0 enables received data in RX FIFO over threshold level interrupt if the RX FIFO is equal to or above the threshold level.	0

6.5 IRDA INTERUPT IDENTIFICATION REGISTER(IRDA_IIR, R, ADDRESS = 0XEC60_0010)

IrDA _IIR	Bit	Description	Initial State
Last Byte to Rx FIFO	[7]	Last byte write to RX FIFO interrupt pending. If the last payload byte of the frame is loaded into the RX FIFO, bit 7 is set to '1'. Bit 7 is set prior to bit 2. Bit 7 is cleared if it is read.	0
Error Indication	[6]	Receiver line error Indication. Bit 6 is set to a '1' if one of three possible errors occurs in the RX process. With the corresponding interrupt enable bit active, one of PHY, CRC and Frame length errors let this bit go active. Bit 6 is cleared if the source of the error is cleared.	0
Tx Underrun	[5]	Transmit under-run interrupt pending. If corresponding interrupt enable bit is active, bit 5 is set to '1' if an under-run occurs in TX FIFO. Bit 5 is cleared by serving the under-run.	0
Last Byte Detect	[4]	Detects last byte of a frame interrupt pending. If the corresponding interrupt enable bit is active, bit 4 is set to '1' if the demodulation block detects the last byte of a received frame and the CRC decoding is finished. Bit 4 is cleared if it is read.	0
Rx Overrun	[3]	RX FIFO over-run interrupt. If corresponding interrupt enable bit is set, bit 3 is active, bit 3 is set to '1' if an overrun occurs in the RX FIFO. Bit 3 is cleared by serving the over-run.	0
Last Byte Read from Rx FIFO	[2]	RX FIFO last byte read interrupt. If corresponding interrupt enable bit is active, it is set to '1' if the CPU reads the last byte of a frame from the RX FIFO. It is cleared if it is read.	0
Tx FIFO Below Threshold	[1]	TX FIFO below threshold interrupt pending. Bit 1 is set to '1' if the transmitter FIFO level is below its threshold level.	0
Rx FIFO Over Threshold	[0]	RX FIFO over threshold interrupt pending. Bit 0 is set to '1' if the receiver FIFO level is equal to or above its threshold level.	0

6.6 IRDA LINE STATUS REGISTER(IRDA_LSR, R, ADDRESS = 0XEC60_0014)

IrDA_LSR	Bit	Description	Initial State
Tx Empty	[7]	Transmitter empty. This bit is set to '1' if TX FIFO is empty and the transmitter front-end is idle.	1
Reserved	[6]	-	0
Received Last Byte from Rx FIFO	[5]	Last byte received from RX FIFO. It is set to a '1' if the microcontroller reads the last byte of a frame from the RX FIFO and cleared if the MCU reads the IrDA_LSR register.	0
Frame Length Error	[4]	Frame length error. It is set to '1' if a frame exceeding the maximum frame length predefined by IrDA_RXFLL and IrDA_RXFLH register is received. This bit is cleared if the microcontroller reads the IrDA_LSR register. If this error is detected, current frame reception is terminated. Data receiving is stopped until the next BOF is detected. Bit 4 is cleared to '0' if the IrDA_LSR register is read by the microcontroller.	0
PHY Error	[3]	PHY error. In FIR mode, It is set to a '1' if an illegal 4PPM symbol is received. In IrDA_MIR mode, if an abort pattern (more than 7 consecutive '1's) is received during reception, this bit is set to '1'. It is cleared if microcontroller reads the LSR register.	0
CRC Error	[2]	CRC error. Bit 2 is set to '1' if a bad IrDA_CRC is detected on data receive. It is cleared to '0' if microcontroller reads the LSR register.	0
Reserved	[1]	-	1
Rx FIFO Empty	[0]	RX FIFO empty. It indicates that the RX FIFO is empty. If the state of RX FIFO turns into empty, it is set to '1'. If the RX FIFO is not empty, it is set to '0'.	1

6.7 IRDA FIFO CONTROL REGISTER(IRDA_FCR, R/W, ADDRESS = 0XEC60_0018)

<table>
<tr><th>IrDA _FCR</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td>Rx FIFO Trigger Level Select</td><td>[7:6]</td><td>Receiver FIFO trigger level selection.
<table>
<tr><th>Bit 7</th><th>Bit 6</th><th>64-byte RX FIFO</th></tr>
<tr><td>0</td><td>0</td><td>01</td></tr>
<tr><td>0</td><td>1</td><td>16</td></tr>
<tr><td>1</td><td>0</td><td>32</td></tr>
<tr><td>1</td><td>1</td><td>56</td></tr>
</table></td><td>00</td></tr>
<tr><td>FIFO Size Select</td><td>[5]</td><td>Must set to '1', to use 64 bytes TX and RX FIFO.</td><td>0</td></tr>
<tr><td>TX FIFO Clear Notification</td><td>[4]</td><td>This bit will be activated if the FIFO clear is over. This bit is cleared by the CPU reads this register.</td><td>0</td></tr>
<tr><td>RX FIFO CLEAR NOTIFICATION</td><td>[3]</td><td>This bit will be activated if the FIFO clear is over. This bit is cleared by the CPU reads this register.</td><td>0</td></tr>
<tr><td>Tx FIFO Reset</td><td>[2]</td><td>TX FIFO reset. If set to '1', bit 2 clears all bytes in the transmitter FIFO and reset its counter to '0'. A '1' written to bit 2 is self-clearing.</td><td>0</td></tr>
<tr><td>Rx FIFO Reset</td><td>[1]</td><td>RX FIFO reset. If set to '1', bit 1 clears all bytes in the receiver FIFO and reset its counter to '0'. A '1' written to bit 1 is self clearing.</td><td>0</td></tr>
<tr><td>FIFO Enable</td><td>[0]</td><td>FIFO enable. If set to '1', bit 0 enables both the transmitter and receiver FIFOs. Bit 0 must be a '1' when setting other IrDA_FCR bits. Changing bit 0 clears the FIFO.</td><td>0</td></tr>
</table>

6.8 IRDA PREAMBLE LENGTH REGISTER(IRDA_PLR, R/W, ADDRESS = 0XEC60_001C)

<table>
<tr><th>REG_PLR</th><th>Bit</th><th>Description</th><th>Initial State</th></tr>
<tr><td>Preamble Length in FIR mode</td><td>[7:6]</td><td>These two bits decide preamble length to be transmitted at the beginning of each frame in FIR mode. The default value of PLR[7:6] = '00' which is equal to 16 preambles.
00 : 16 01 : 4
01: 8 11: 32</td><td>00</td></tr>
<tr><td>TX FIFO Trigger Level Select</td><td>[5:4]</td><td>Transceiver FIFO trigger level selection.
<table>
<tr><th>Bit 5</th><th>Bit 4</th><th>64-byte FIFO</th></tr>
<tr><td>0</td><td>0</td><td>Reserved</td></tr>
<tr><td>0</td><td>1</td><td>48</td></tr>
<tr><td>1</td><td>0</td><td>32</td></tr>
<tr><td>1</td><td>1</td><td>08</td></tr>
</table>
Note: Tx Trigger level value means how many data are empty.</td><td>01</td></tr>
<tr><td>Number of Start Flags in MIR Mode</td><td>[3:0]</td><td>Number of start flags in MIR mode. The number of start flags to be transmitted at the beginning of a frame is equal to the IrDA_PLR [3:0] value. The minimum value is 3.</td><td>0010</td></tr>
</table>

6.9 IRDA RECEIVER & TRANSMITTER BUFFER REGISTER(IRDA_RBR/IRDA_THR, R/W, ADDRESS = 0XEC60_0020)

IrDA _RBR	Bit	Description	Initial State
Rx/Tx Data	[7:0]	Received data (If read data) Data to transmit (If write data)	0x00

6.10 IRDA TOTAL NUMBER OF DATA BYTES REMAINED IN TX FIFO(IRDA_TXNO, R, ADDRESS = 0XEC60_0024)

IrDA _TxNO	Bit	Description	Initial State
Tx Data Total Number	[7:0]	The total number of data bytes remaining in Tx FIFO	0x00

6.11 IRDA TOTAL NUMBER OF DATA BYTES REMAINED IN RX FIFO(IRDA_RXNO, R, ADDRESS = 0XEC60_0028)

IrDA _RxNO	Bit	Description	Initial State
Rx data total number	[7:0]	The total number of data bytes remaining in Rx FIFO.	00

6.12 IRDA TRANSMIT FRAME-LENGTH REGISTER LOW(IRDA_TXFLL, R/W, ADDRESS = 0XEC60_002C)

IrDA _TXFLL	Bit	Description	Initial State
Tx frame length low	[7:0]	TXFLL stores the lower 8 bits of the byte number of the frame to be transmitted.	00

6.13 IRDA TRANSMIT FRAME-LENGTH REGISTER HIGH(IRDA_TXFLH, R/W, ADDRESS = 0XEC60_0030)

IrDA _TXFLH	Bit	Description	Initial State
Tx frame length high	[7:0]	TXFLH stores the upper 8 bits of the byte number of the frame to be transmitted.	00

6.14 IRDA RECEIVER FRAME-LENGTH REGISTER LOW(IRDA_RXFLL, R/W, ADDRESS = 0XEC60_0034)

IrDA _RXFLL	Bit	Description	Initial State
Rx frame length low	[7:0]	RXFLL stores the lower 8 bits of the maximum byte number of the frame to be received.	00

6.15 IRDA RECEIVER FRAME-LENGTH REGISTER HIGH(IRDA_RXFLH, R/W, ADDRESS = 0XEC60_0038)

IrDA _RXFLH	Bit	Description	Initial State
Reserved	[7:6]	-	00
Rx frame length high	[5:0]	TXFLL stores the upper 6 bits of the maximum byte number of the frame to be received.	

6.16 IRDA INTERRUPT CLEAR REGISTER (IRDA_INTCLR, W, ADDRESS = 0XEC60_003C)

IrDA _INTCLR	Bit	Description	Initial State
Interrupt Clear	[31:0]	Read undefined. Write any value to clear IrDA interrupt.	

8.5 C_CAN

1 OVERVIEW

The C_CAN is a CAN module integrated as stand-alone device or as part of an ASIC. It is described in VHDL on RTL level, prepared for synthesis. It consists of the components namely (see Figure 8.5-1) CAN Core, Message RAM, Message Handler, Control Registers, and Module Interface.

The CAN_Core performs communication according to the CAN protocol version 2.0 part A and B. The bit rate is programmed to values up to 1MBit/s. This depends on the technology used. To connect to the physical layer additional transceiver hardware is required.

To communicate on a CAN network, individual Message Objects are configured. The Message Objects and Identifier Masks for acceptance filtering of received messages are stored in the Message RAM.

The functions related to handling of messages are implemented in the Message Handler. The functions implemented are Acceptance Filtering, Transfer of messages between the CAN Core and the Message RAM, and Handling of transmission requests as well as the generation of the module interrupt.

The register set of the C_CAN is accessed directly by an external CPU via the module interface. These registers are used to control/ configure the CAN Core and the Message Handler and to access the Message RAM.

The Module Interfaces delivered with the C_CAN module can easily be replaced by a customized module interface adapted to the user requirement.

2 FEATURE

- Supports CAN protocol version 2.0 part A and B
- Bit rates up to 1 MBit/s
- 32 Message Objects
- Each Message Object has its own identifier mask
- Programmable FIFO mode (Concatenation of Message Objects)
- Maskable interrupt
- Disabled Automatic Retransmission mode for Time Triggered CAN applications
- Programmable loop-back mode for self-test operation
- 8-bit non-multiplex Motorola HC08 compatible module interface
- Two 16-bit module interfaces to the AMBA APB bus from ARM

3 BLOCK DIAGRAM

The design consists of the functional blocks described below (see Figure 8.5-1):

CAN Core

CAN Protocol Controller and Rx/Tx Shift Register for serial/ parallel conversion of messages.

Message RAM

Stores Message Objects and Identifier Masks.

Registers

All registers used to control and configure the C_CAN module.

Message Handler

State Machine that controls the data transfer between the Rx/Tx Shift Register of the CAN Core and the Message RAM as well as the generation of interrupts as programmed in the Control and Configuration Registers.

Module Interface

The C_CAN module is delivered with three different interfaces. An 8-bit interface for the Motorola HC08 controller and two 16-bit interfaces to the AMBA APB bus from ARM. This can easily be replaced by a user-defined module interface.

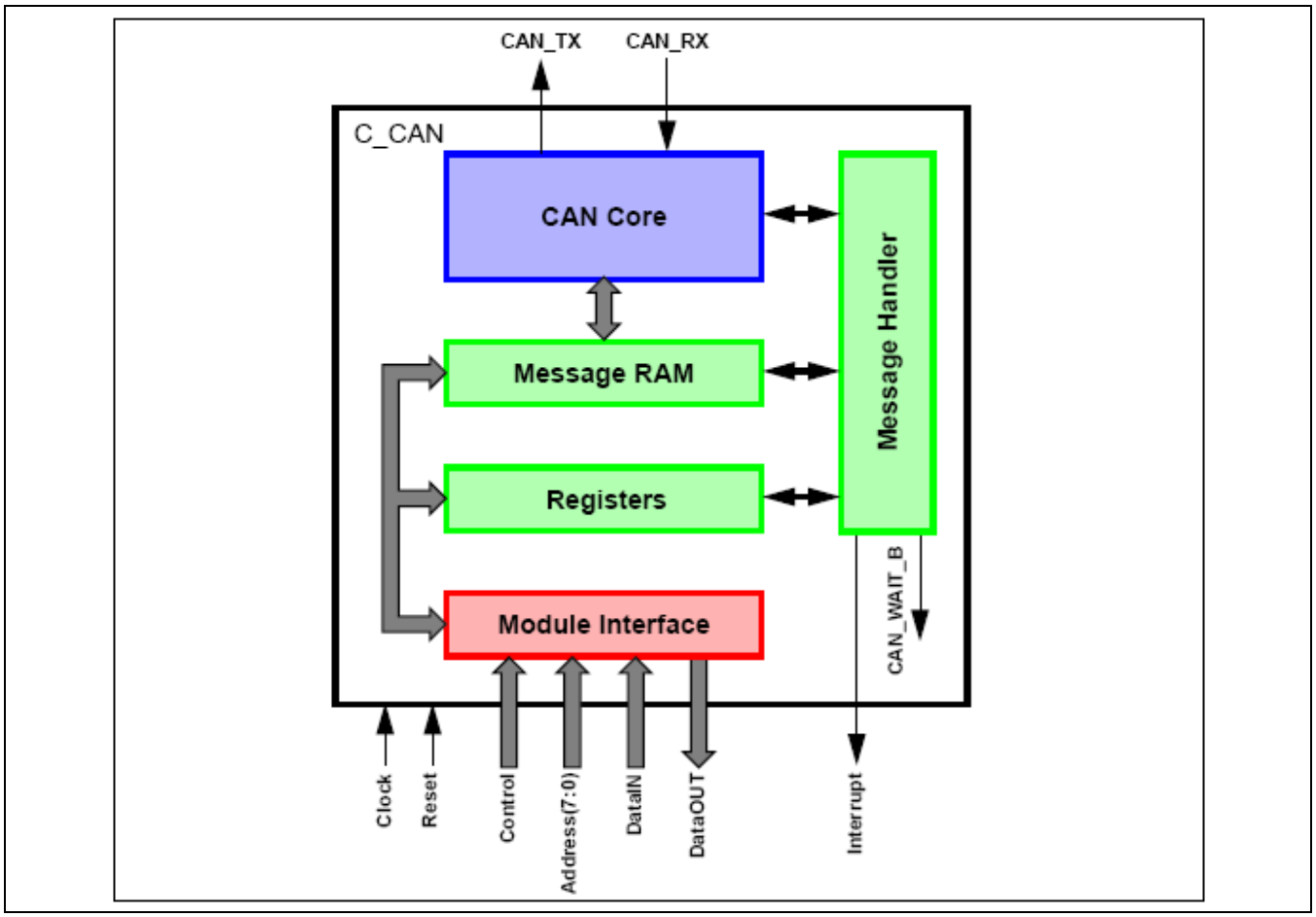


Figure 8.5-1 Block Diagram of the C_CAN

4 OPERATION MODES

4.1 SOFTWARE INITIALISATIOIN

The software initialization is started by setting the bit Init in the CAN Control Register, either by software or by a hardware reset, or by going Bus_Off.

While Init is set, all message transfer from and to the CAN bus is stopped. The status of the CAN bus output CAN_TX is recessive (HIGH). The counters of the EML are unchanged. Setting Init does not change any configuration register.

To initialize the CAN Controller, the CPU has to set up the Bit Timing Register and each Message Object. If a Message Object is not needed, it is sufficient to set it's MsgVal bit to not valid. Otherwise, the whole Message Object has to be initialized.

Access to the Bit Timing Register and to the BRP Extension Register for the configuration of the bit timing is enabled if both bits Init and CCE in the CAN Control Register are set.

Resetting Init (by CPU only) finishes the software initialization. Later the Bit Stream Processor BSP (Refer Section 4.10 on Page 34) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (º Bus Idle) before it can take part in bus activities and starts the message transfer.

The initialization of the Message Objects is independent of Init and is done on the fly, but the Message Objects should be configured to particular identifiers or set to not valid before the BSP starts the message transfer.

To change the configuration of a Message Object during normal operation, the CPU has to start by setting MsgVal to not valid. If the configuration is completed, MsgVal is set to valid again.

4.1.1 CAN Message Transfer

Once the C_CAN is initialized and Init is reset to 0, the C_CAN's CAN Core synchronizes itself to the CAN bus and starts the message transfer.

Received messages are stored into their appropriate Message Objects if they pass the Message Handler's acceptance filtering. The whole message including all arbitration bits, DLC and eight data bytes is stored into the Message Object. If the Identifier Mask is used, the arbitration bits which are masked to "don't care" may be overwritten in the Message Object.

The CPU may read or write each message any time via the Interface Registers, the Message Handler guarantees data consistency in case of concurrent accesses.

Messages to be transmitted are updated by the CPU. If a permanent Message Object (arbitration and control bits set up during configuration) exists for the message, only the data bytes are updated and then TxRqst bit with NewDat bit are set to start the transmission. If several transmit messages are assigned to the same Message Object (If the number of Message Objects is not sufficient), the whole Message Object has to be configured before the transmission of this message is requested.

The transmission of any number of Message Objects may be requested at the same time, they are transmitted subsequently according to their internal priority. Messages is updated or set to not valid any time, even if their requested transmission is still pending. The old data is discarded if a message is updated before its pending transmission has started.

Depending on the configuration of the Message Object, the transmission of a message may be requested autonomously by the reception of a remote frame with a matching identifier.

4.1.2 Disabled Automatic retransmission

According to the CAN Specification (see ISO11898, 6.3.3 Recovery Management), the C_CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed. By default, this means for automatic retransmission is enabled. It can be disabled to enable the C_CAN to work within a Time Triggered CAN (TTCAN, see ISO11898-1) environment.

The Disabled Automatic Retransmission mode is enabled by programming bit DAR in the CAN Control Register to one. In this operation mode the programmer has to consider the different behavior of bits TxRqst and NewDat in the Control Registers of the Message Buffers:

- If a transmission starts bit TxRqst of the respective Message Buffer is reset, while bit NewDat remains set.
- If the transmission completed successfully bit NewDat is reset. If a transmission failed (lost arbitration or error) bit NewDat remains set. To restart the transmission the CPU has to set TxRqst back to one.

4.1.3 Test Mode

Set Test bit to 1 in the CAN Control Register to enter Test Mode. In Test Mode the bits Tx1, Tx0, LBack, Silent and Basic in the Test Register are writable. Bit Rx monitors the state of pin CAN_RX and therefore is only readable. All Test Register functions are disabled if Test bit is reset to 0.

4.1.4 Silent Mode

The CAN Core is set in Silent Mode by programming the Test Register bit Silent to 1.

In Silent Mode, the C_CAN is able to receive valid data frames and valid remote frames, but it sends only recessive bits on the CAN bus and it cannot start a transmission. If the CAN Core is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN Core monitors this dominant bit, although the CAN bus may remain in recessive state. The Silent Mode is used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits (Acknowledge Bits, Error Frames). Figure 8.5-2 shows the connection of signals CAN_TX and CAN_RX to the CAN Core in Silent Mode.

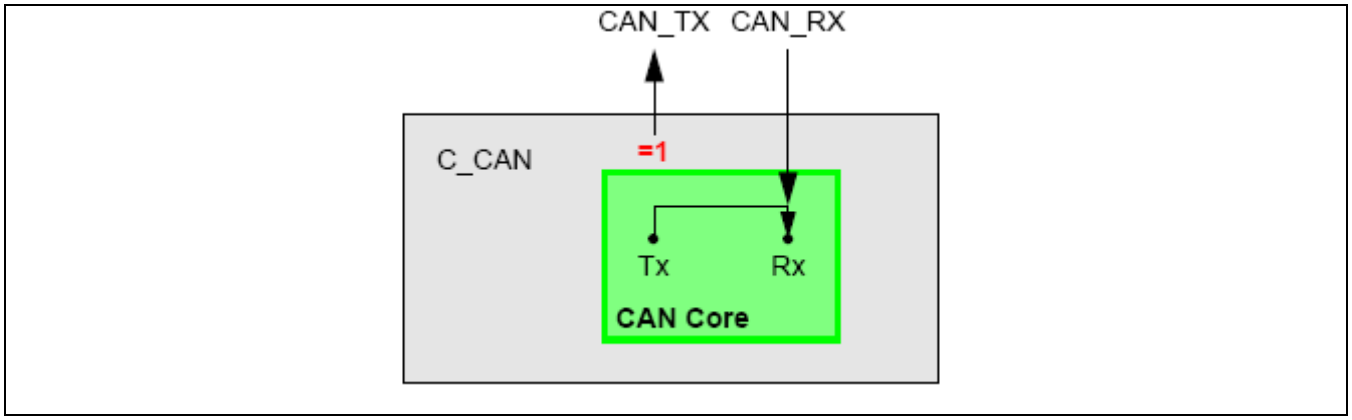


Figure 8.5-2 CAN Core in Silent Mode

4.1.5 Loop Back Mode

The CAN Core is set in Loop Back Mode by programming the Test Register bit LBack to 1. In Loop Back Mode, the CAN Core treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into a Receive Buffer. Figure 8.5-3 shows the connection of signals CAN_TX and CAN_RX to the CAN Core in Loop Back Mode.

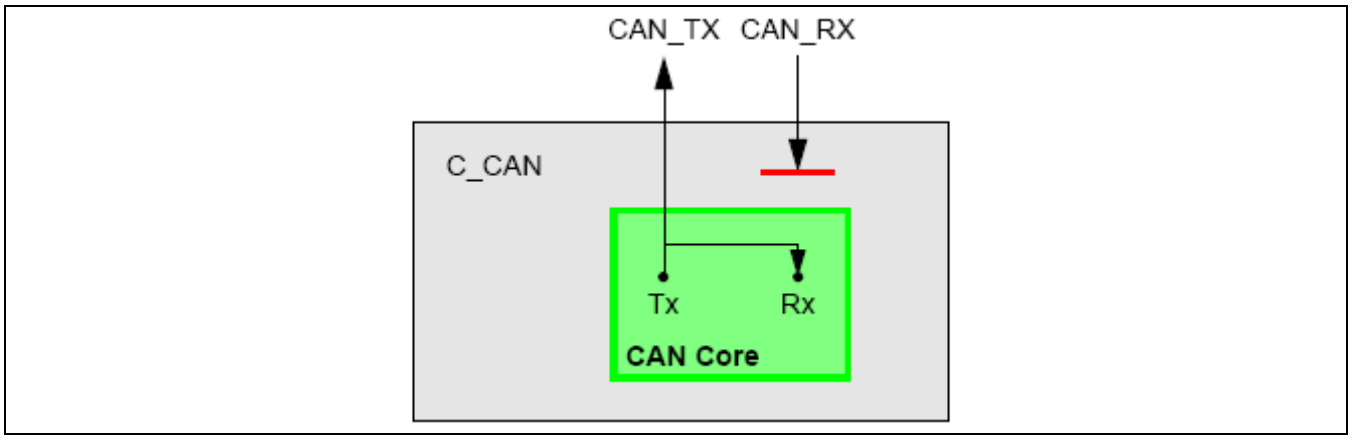


Figure 8.5-3 CAN Core in Loop Back Mode

4.1.6 Loop Back combined with Silent Mode

It is also possible to combine Loop Back Mode and Silent Mode by programming bits LBack and Silent to one at the same time. This mode is used for a "Hot Selftest", meaning the C_CAN is tested without affecting a running CAN system connected to the pins CAN_TX and CAN_RX. In this mode the CAN_RX pin is disconnected from the CAN Core and the CAN_TX pin is held recessive. Figure 8.5-4 shows the connection of signals CAN_TX and CAN_RX to the CAN Core in case of the combination of Loop Back Mode with Silent Mode.

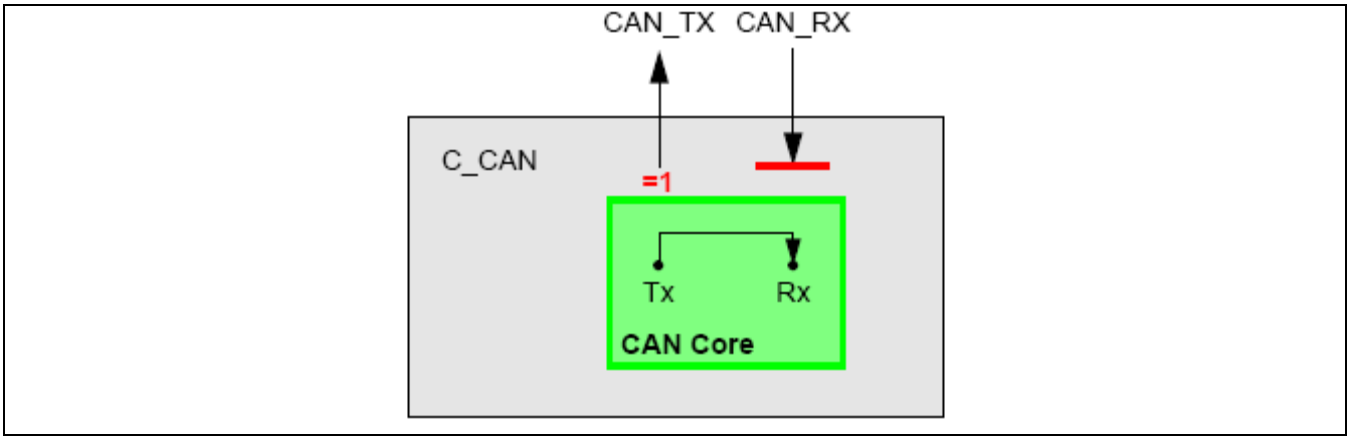


Figure 8.5-4 CAN Core in Loop Back Combined with Silent Mode

4.1.7 Basic Mode

The CAN Core is set in Basic Mode by programming the Test Register bit Basic to 1. In this mode the C_CAN module runs without the Message RAM.

The IF1 Registers are used as Transmit Buffer. The transmission of the contents of the IF1 Registers is requested by writing the Busy bit of the IF1 Command Request Register to '1'. The IF1 Registers are locked while the Busy bit is set. The Busy bit indicates that the transmission is pending.

As soon as the CAN bus is idle, the IF1 Registers are loaded into the shift register of the CAN Core and the transmission is started. If the transmission is complete, the Busy bit is reset and the locked IF1 Registers are released.

A pending transmission is aborted any time by resetting the Busy bit in the IF1 Command Request Register while the IF1 Registers are locked. If the CPU has reset the Busy bit, a possible retransmission in case of lost arbitration or in case of an error is disabled.

The IF2 Registers are used as Receive Buffer. After the reception of a message the contents of the shift register is stored into the IF2 Registers, without any acceptance filtering.

Additionally, the actual contents of the shift register are monitored during the message transfer. Each time a read Message Object is initiated by writing the Busy bit of the IF2 Command Request Register to '1', the contents of the shift register is stored into the IF2 Registers.

In Basic Mode the evaluation of all Message Object related control and status bits and of the control bits of the IFx Command Mask Registers is turned off. The message number of the Command request registers is not evaluated. The NewDat and MsgLst bits of the IF2 Message Control Register retain their function, DLC3-0 will show the received DLC, the other control bits will be read as '0'.

In Basic Mode the ready output CAN_WAIT_B is disabled (always '1').

4.1.8 Software control of pin CAN_TX

Four output functions are available for the CAN transmit pin CAN_TX. Additionally to its default function, the serial data output, drives the CAN Sample Point signal to monitor CAN_Core's bit timing and drives constant dominant or recessive values. The last two functions, combined with the readable CAN receive pin CAN_RX, are used to check the CAN bus' physical layer.

The output mode of pin CAN_TX is selected by programming the Test Register bits Tx1 and Tx0.

The three test functions for pin CAN_TX interfere with all CAN protocol functions. CAN_TX must be left in its default function if CAN message transfer or any of the test modes Loop Back Mode, Silent Mode, or Basic Mode are selected.

4.2 APB INTERFACE

There are two interfaces to the AMBA APB.

With APB1 interface the block CCAN_GENERIC is clocked by the APB system clock PCLK. For the APB2 interface a clock divider is inserted between APB2 interface and CCAN_GENERIC. Please refer to "AMBA Specification Rev. 2.0" for more details on the ARM AMBA bus.

The APB2 interface synchronizes write accesses from the PCLK domain to the CAN_CLK domain and read accesses from the CAN_CLK domain to the PCLK domain. It is intended for cases where the C_CAN module is connected to an AMBA APB bus which is clocked with a PCLK >> 8 MHz. To reduce the power consumption of the module and to be able to generate the lower baud rates on the CAN bus a clock divider was inserted. The clock divider can be set to values of 2^{DIV}, where DIV is a positive value. For the actual version of the APB2 interface it is set to 4 (2^2)

The structure of the APB2 interface is shown is Figure 8.5-5.

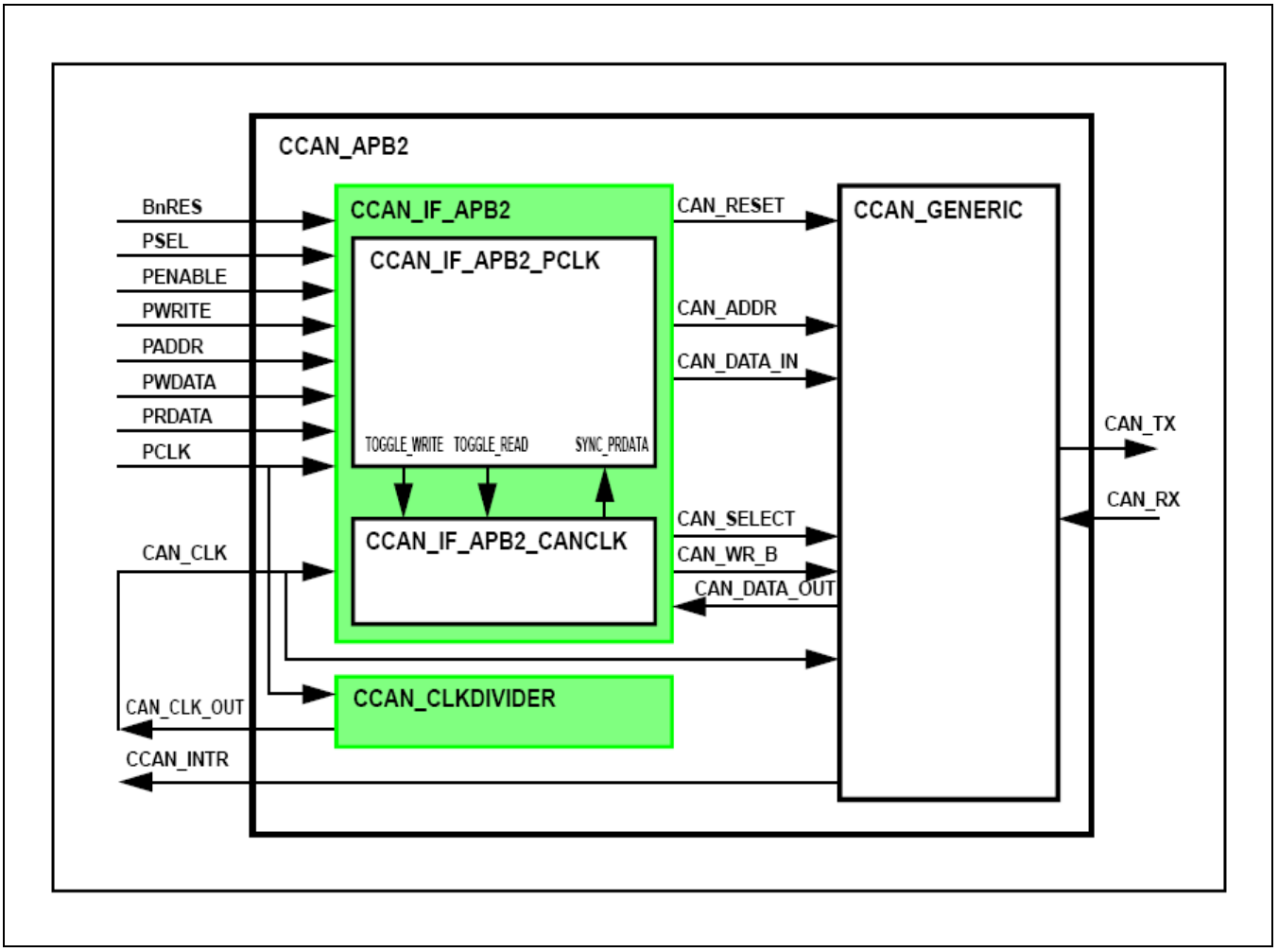


Figure 8.5-5 Structure of the C_CAN with APB2 Interface

If there is a write access, the registers CAN_ADDR and CAN_DATA_IN store the actual address PADDR and write data PWDATA clocked with PCLK. Additionally the flip-flop TOGGLE_WR is toggled.

The flips-flops TOGGLE_WR, WR_TOGGLE1 and WR_TOGGLE2 describe a shift register, where TOGGLE_WR is clocked by PCLK, WR_TOGGLE1 and WR_TOGGLE2 are clocked by CAN_CLK. The synchronized write signal SYNC_WRITE is generated by an EXOR of signals WRITE_TOGGLE1 and WRITE_TOGGLE2 (see Figure 8.5-6).

Two consecutive write accesses must have a minimum distance of eight PCLK periods (2 CAN_CLK periods, when frequency (PCLK) = 4 x frequency (CAN_CLK)).

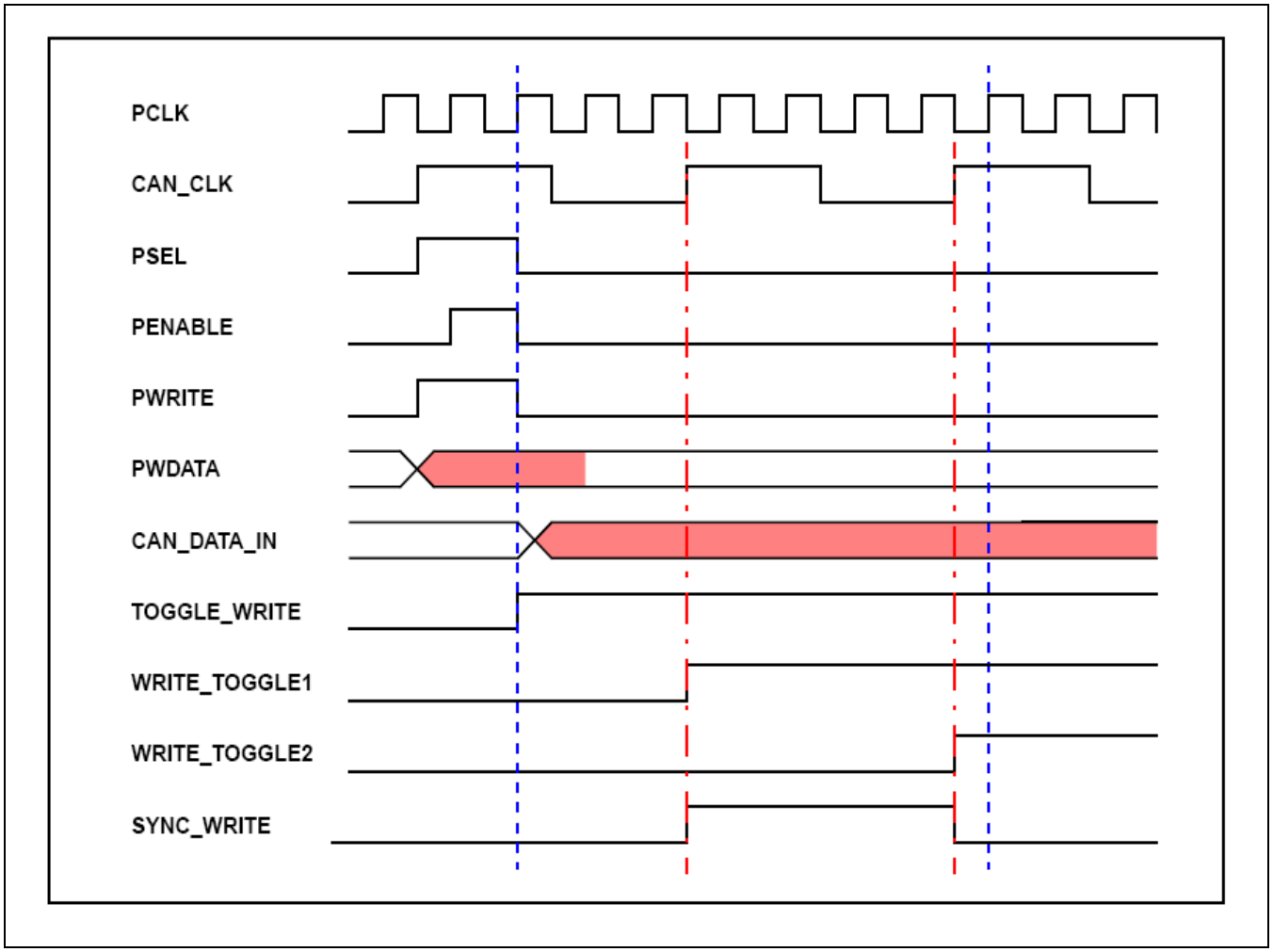


Figure 8.5-6 APB2 Write Timing

A read access to the C_CAN module with APB2 interface must be performed as "double read". With the first read, the requested data is written into the internal register (SYNC_PRDATA) clocked by CAN_CLK. With the second read the data output register PRDATA takes over the value of SYNC_PRDATA (see Figure 8.5-7).

The two consecutive read accesses must have a minimum distance of nine PCLK periods (when frequency (PCLK) = 4 x frequency (CAN_CLK)).

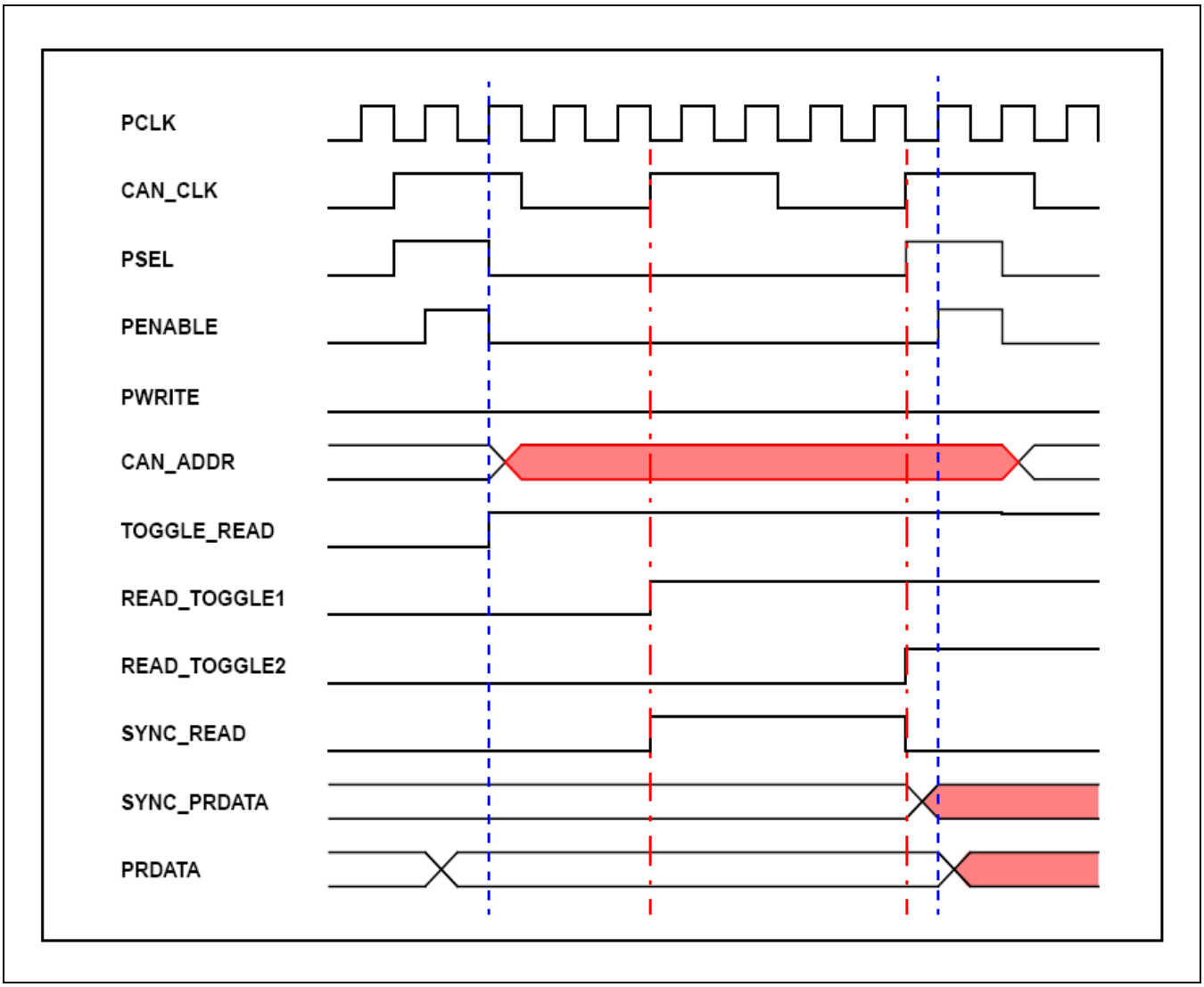


Figure 8.5-7 APB2 Read Timing

5 I/O DESCRIPTION

Table 8.5-1 CCAN Signals

Function Signal	I/O	Description	Pad	Type
CAN0_TX	Output	CCAN0 TX Data	XEINT[28]	Muxed
CAN0_RX	Input	CCAN0 RX Data	XEINT[29]	Muxed
CAN1_TX	Output	CCAN1 TX Data	XEINT[30]	Muxed
CAN1_RX	Input	CCAM1 RX Data	XEINT[31]	Muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

6 REGISTER DESCRIPTION

The C_CAN module allocates an address space of 256 bytes. The registers are organized as 16-bit registers, with the high byte at the odd address and the low byte at the even address.

The two sets of interface registers (IF1 and IF2) control the CPU access to the Message RAM. They buffer the data to be transferred to and from the RAM, avoiding conflicts between CPU accesses and message reception/ transmission.

Table 8.5-2 C_CAN Register Summary

Register	Address	R/W	Description	Reset Value
CCAN0				
CAN_CON	0xEC70_0000	R/W	CAN Control Register	0x0001
CAN_STS	0xEC70_0004	R/W	Status Register	0x0000
CAN_ERR	0xEC70_0008	R	Error Counter	0x0000
CAN_BT	0xEC70_000C	R/W	Bit Timing Register(Write enabled by CCE)	0x2301
CAN_INTR	0xEC70_0010	R	Interrupt Register	0x0000
CAN_TEST	0xEC70_0014	R/W	Test Register(Write enabled by Test)	0x0000
CAN_BRP	0xEC70_0018	R/W	BRP Extension Register(Write enabled by CCE)	0x0000
-	0xEC70_001C	-	- reserved	0x0000
CAN_IF1_CR	0xEC70_0020	R/W	IF1 Command Request	0x0001
CAN_IF1_CM	0xEC70_0024	R/W	IF1 Command Mask	0x0000
CAN_IF1_M1	0xEC70_0028	R/W	IF1 Mask1	0xFFFF
CAN_IF1_M2	0xEC70_002C	R/W	IF1 Mask2	0xFFFF
CAN_IF1_A1	0xEC70_0030	R/W	IF1 Arbitration 1	0x0000
CAN_IF1_A2	0xEC70_0034	R/W	IF1 Arbitration 2	0x0000
CAN_IF1_MC	0xEC70_0038	R/W	IF1 Message Control	0x0000
CAN_IF1_DA1	0xEC70_003C	R/W	IF1 Data A 1	0x0000
CAN_IF1_DA2	0xEC70_0040	R/W	IF1 Data A 2	0x0000
CAN_IF1_DB1	0xEC70_0044	R/W	IF1 Data B 1	0x0000
CAN_IF1_DB2	0xEC70_0048	R/W	IF1 Data B 2	0x0000
-	0xEC70_004C~ 0xEC70_007C	-	- reserved	0x0000
CAN_IF2_CR	0xEC70_0080	R/W	IF2 Command Request	0x0001
CAN_IF2_CM	0xEC70_0084	R/W	IF2 Command Mask	0x0000
CAN_IF2_M1	0xEC70_0x88	R/W	IF2 Mask1	0xFFFF
CAN_IF2_M2	0xEC70_008C	R/W	IF2 Mask2	0xFFFF
CAN_IF2_A1	0xEC70_0090	R/W	IF2 Arbitration 1	0x0000
CAN_IF2_A2	0xEC70_0094	R/W	IF2 Arbitration 2	0x0000
CAN_IF2_MC	0xEC70_0x98	R/W	IF2 Message Control	0x0000

Register	Address	R/W	Description	Reset Value
CAN_IF2_DA1	0xEC70_009C	R/W	IF2 Data A 1	0x0000
CAN_IF2_DA2	0xEC70_00A0	R/W	IF2 Data A 2	0x0000
CAN_IF2_DB1	0xEC70_00A4	R/W	IF2 Data B 1	0x0000
CAN_IF2_DB2	0xEC70_00A8	R/W	IF2 Data B 2	0x0000
-	0xEC70_00AC ~ 0xEC70_00FC	-	Reserved	0x0000
CAN_TrReq1	0xEC70_00100	R	Transmission Request 1	0x0000
CAN_TrReq2	0xEC70_00104	R	Transmission Request 2	0x0000
-	0xEC70_0108~ 0xEC70_011C	-	Reserved	0x0000
CAN_ND1	0xEC70_0120	R	New Data 1	0x0000
CAN_ND2	0xEC70_0124	R	New Data 2	0x0000
-	0xEC70_0128~ 0xEC70_013C	-	- Reserved	0x0000
CAN_IntPen1	0xEC70_0140	R	Interrupt Pending 1	0x0000
CAN_IntPen2	0xEC70_0144	R	Interrupt Pending 2	0x0000
-	0xEC70_0148~ 0xEC70_015C	-	- Reserved	0x0000
CAN_MesVal1	0xEC70_0160	R	Message Valid 1	0x0000
CAN_MesVal2	0xEC70_0164	R	Message Valid 2	0x0000
-	0xEC70_0168~ 0xEC70_017C	-	- Reserved	0x0000
CCAN1				
CAN_CON	0xEC80_0000	R/W	CAN Control Register	0x0001
CAN_STS	0xEC80_0004	R/W	Status Register	0x0000
CAN_ERR	0xEC80_0008	R	Error Counter	0x0000
CAN_BT	0xEC80_000C	R/W	Bit Timing Register(Write enabled by CCE)	0x2301
CAN_INTR	0xEC80_0010	R	Interrupt Register	0x0000
CAN_TEST	0xEC80_0014	R/W	Test Register(Write enabled by Test)	0x0000
CAN_BRP	0xEC80_0018	R/W	BRP Extension Register(Write enabled by CCE)	0x0000
-	0xEC80_001C	-	- Reserved	0x0000
CAN_IF1_CR	0xEC80_0020	R/W	IF1 Command Request	0x0001
CAN_IF1_CM	0xEC80_0024	R/W	IF1 Command Mask	0x0000
CAN_IF1_M1	0xEC80_0028	R/W	IF1 Mask1	0xFFFF
CAN_IF1_M2	0xEC80_002C	R/W	IF1 Mask2	0xFFFF
CAN_IF1_A1	0xEC80_0030	R/W	IF1 Arbitration 1	0x0000
CAN_IF1_A2	0xEC80_0034	R/W	IF1 Arbitration 2	0x0000

Register	Address	R/W	Description	Reset Value
CAN_IF1_MC	0xEC80_0038	R/W	IF1 Message Control	0x0000
CAN_IF1_DA1	0xEC80_003C	R/W	IF1 Data A 1	0x0000
CAN_IF1_DA2	0xEC80_0040	R/W	IF1 Data A 2	0x0000
CAN_IF1_DB1	0xEC80_0044	R/W	IF1 Data B 1	0x0000
CAN_IF1_DB2	0xEC80_0048	R/W	IF1 Data B 2	0x0000
-	0xEC80_004C~ 0xEC80_007C	-	- Reserved	0x0000
CAN_IF2_CR	0xEC80_0080	R/W	IF2 Command Request	0x0001
CAN_IF2_CM	0xEC80_0084	R/W	IF2 Command Mask	0x0000
CAN_IF2_M1	0xEC80_0x88	R/W	IF2 Mask1	0xFFFF
CAN_IF2_M2	0xEC80_008C	R/W	IF2 Mask2	0xFFFF
CAN_IF2_A1	0xEC80_0090	R/W	IF2 Arbitration 1	0x0000
CAN_IF2_A2	0xEC80_0094	R/W	IF2 Arbitration 2	0x0000
CAN_IF2_MC	0xEC80_0x98	R/W	IF2 Message Control	0x0000
CAN_IF2_DA1	0xEC80_009C	R/W	IF2 Data A 1	0x0000
CAN_IF2_DA2	0xEC80_00A0	R/W	IF2 Data A 2	0x0000
CAN_IF2_DB1	0xEC80_00A4	R/W	IF2 Data B 1	0x0000
CAN_IF2_DB2	0xEC80_00A8	R/W	IF2 Data B 2	0x0000
-	0xEC80_00AC ~ 0xEC80_00FC	-	- Reserved	0x0000
CAN_TrReq1	0xEC80_00100	R	Transmission Request 1	0x0000
CAN_TrReq2	0xEC80_00104	R	Transmission Request 2	0x0000
-	0xEC80_0108~ 0xEC80_011C	-	- Reserved	0x0000
CAN_ND1	0xEC80_0120	R	New Data 1	0x0000
CAN_ND2	0xEC80_0124	R	New Data 2	0x0000
-	0xEC80_0128~ 0xEC80_013C	-	- Reserved	0x0000
CAN_IntPen1	0xEC80_0140	R	Interrupt Pending 1	0x0000
CAN_IntPen2	0xEC80_0144	R	Interrupt Pending 2	0x0000
-	0xEC80_0148~ 0xEC80_015C	-	- Reserved	0x0000
CAN_MesVal1	0xEC80_0160	R	Message Valid 1	0x0000
CAN_MesVal2	0xEC80_0164	R	Message Valid 2	0x0000
-	0xEC80_0168~ 0xEC80_017C	-	- Reserved	0x0000

The two sets of Message Interface Registers – IF2 and IF2 – have identical functions.

Reserved bits are read as '0' except for IFx Mask 2 Register where they are read as '1'.

All reserved bits are read only.

6.1 HARDWARE RESET DESCRIPTION

After hardware reset, the registers of the C_CAN hold the values described in table 8.5-3.

Additionally the busoff state is reset and the output CAN_TX is set to recessive (HIGH). The value 0x0001 (Init = '1') in the CAN Control Register enables the software initialization. The C_CAN does not influence the CAN bus until the CPU resets Init to '0'.

The data stored in the Message RAM is not affected by a hardware reset. After power-on, the contents of the Message RAM are undefined.

6.2 DETAILED DESCRIPTION

These registers are related to the CAN protocol controller in the CAN Core. They control the operating modes and the configuration of the CAN bit timing and provide status information.

6.2.1 CAN Control Register

- CAN0_CON, R/W, Address = 0xEC70_0000
- CAN1_CON, R/W, Address = 0xEC80_0000

CANn_CON	Bit	Description	R/W	Reset Value
Reserved	[15:8]	Reserved	R	0
Test	[7]	Test Mode Enable 1 = Test Mode 0 = Normal Operation	R/W	0
CCE	[6]	Configuration Change Enable 1 = The CPU has write access to the Bit Timing Register (while Init = 1'b1) 0 = The CPU has no write access to the Bit Timing Register	R/W	0
DAR	[5]	Disable Automatic Retransmission 1 = Disables Automatic Retransmission 0 = Enables Automatic Retransmission of disturbed messages	R/W	0
Reserved	[4]	Reserved	R	0
EIE	[3]	Error Interrupt Enable 1 = Enabled – A change in the bits BOff or EWarn in the Status Register will generate an interrupt 0 = Disabled – No Error Status Interrupt is generated	R/W	0
SIE	[2]	Status Change Interrupt Enable 1= Enabled – An interrupt is generated if a message transfer is successfully completed or a CAN bus error is detected 0 = Disabled – No Status Change Interrupt is generated	R/W	0
IE	[1]	Module Interrupt Enable 1 = Enabled – Interrupts sets IRQ_B to LOW. IRQ_B remains LOW until all pending interrupts are processed 0 = Disabled – Module Interrupt IRQ_B is always HIGH	R/W	0
Init	[0]	Initialization 1 = Initialization is started 0 = Normal Operation	R/W	1

NOTE: The busoff recovery sequence (see CAN Specification Rev. 2.0) cannot be shortened by setting or resetting Init. If the device goes busoff, it sets Init of its own accord, stopping all bus activities. Once Init has been cleared by the CPU, the device waits for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operations. At the end of the busoff recovery sequence, the Error Management Counters will be reset.

During the waiting time after the resetting of Init, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the Status Register. This enables the CPU to readily check whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the proceeding of the busoff recovery sequence.

6.2.2 Status Register

- CAN0_STS, R/W, Address = 0xEC70_0004
- CAN1_STS, R/W, Address = 0xEC80_0004

CANn_STS	Bit	Description	R/W	Reset Value
Reserved	[15:8]	Reserved	R	0
BOff	[7]	Busoff Status 1 = The CAN module is in busoff state. 0 = The CAN module is not in busoff state.	R	0
EWarn	[6]	Warning Status 1 = At least one of the error counters in the EML has reached the error warning limit of 96. 0 = Both error counters are below the error warning limit of 96.	R	0
EPass	[5]	Error Passive 1 = The CAN Core is in the error passive state as defined in the CAN Specification. 0 = The CAN Core is in error active state.	R	0
RxOk	[4]	Received a Message Successfully 1= Since this bit was last reset (to zero) by the CPU, a message has been successfully received (independent of the result of acceptance filtering). 0 = Since this bit was last reset by the CPU, no message has been successfully received. This bit is never reset by the CAN Core	R/W	0
TxOk	[3]	Transmitted a Message Successfully 1: Since this bit was last reset by the CPU, a message has been successfully (error free and acknowledged by at least one other node) transmitted. 0: Since this bit was reset by the CPU, no message has been successfully transmitted. This bit is never reset by the CAN Core.	R/W	0
LEC	[2:0]	Last Error Code(Type of the last error to occur on the CAN bus) 000 = No Error 001 = Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 010 = Form Error :A fixed format part of a received frame has the wrong format 011 = AckError : The message this CAN Core transmitted was not acknowledged by another node	R/W	0

CANn_STS	Bit	Description	R/W	Reset Value
		100 = Bit1Error : During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant 101 = Bit0Error : During the transmission of a message (of acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored Bus value was recessive. During busoff recovery this status is set each time after a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the busoff recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 110 = CRCError: The CRC check sum is incorrect in the message received. The CRC received for an incoming message does not match with the calculated CRC for the received data 111 = unused : If the LEC show the value '7', no CAN bus event was detected since the CPU wrote this value to the LEC.		

6.2.3 Error Counter

- CAN0_ERR, R/W, Address = 0xEC70_0008
- CAN1_ERR, R/W, Address = 0xEC80_0008

CANn_ERR	Bit	Description	R/W	Reset Value
RP	[15]	Receive Error Passive 1 = The Receive Error Counter has reached the error passive level as defined in the CAN Specification 0 = The Receive Error Counter is below the error passive level	R	0
REC6-0	[14:8]	Receive Error Counter Actual state of the Receive Error Counter. Values between 0 and 127	R	0
TEC7-0	[7:0]	Transmit Error Counter Actual state of the Transmit Error Counter. Values between 0 and 255	R	0

6.2.4 Bit timing Register

- CAN0_BT, R/W, Address = 0xEC70_000C
- CAN1_BT, R/W, Address = 0xEC80_000C

CANn_BT	Bit	Description	R/W	Reset Value
Reserved	[15]	Reserved	R	0
TSeg2	[14:12]	The time segment after the sample point 0x0-0x7: valid values for TSeg2 are [0..7]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	R/W	0x2
TSeg1	[11:8]	The timing segment before the sample point 0x01-0x0F: valid values for TSeg1 are [1..15]. The actual interpretation by the hard ware of this value is such that one more than the value programmed here is used	R/W	0x3
SJW	[7:6]	(Re)Synchronisation Jump Width 0x0-0x3: Valid programmed values are 0-3. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	R/W	0
BRP	[5:0]	Baud Rate Prescaler 0x01-0x3F: The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is build up from a multiple of these quanta. Valid values for the Baud Rate Prescaler are [0..63]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	R/W	0x10

NOTE: With a module clock CAN_CLK of 8 MHz, the reset value of 0x2301 configures the C_CAN for a bit rate of 500 kBit/s. The registers are only writable if bits CCE and Init in the CAN Control Register are set.

6.2.5 Interrupt Register

- CAN0_INTR, R/W, Address = 0xEC70_0010
- CAN1_INTR, R/W, Address = 0xEC80_0010

CANn_INTR	Bit	Description	R/W	Reset Value
IntID15-0	[15:0]	Interrupt Identifier (the number here indicates the source of the interrupt) 0x0000: No interrupt is pending 0x0001-0x0020: Number of Message Object which caused the interrupt 0x0021-0x7FFF: unused 0x8000: Status Interrupt 0x8001-0xFFFF: unused	R	0

6.2.6 Test Register

- CAN0_TEST, R/W, Address = 0xEC70_0014
- CAN1_TEST, R/W, Address = 0xEC80_0014

CANn_TEST	Bit	Description	R/W	Reset Value
Reserved	[15:8]	Reserved	R	0
Rx	[7]	Monitors the actual value of the CAN_RX Pin 1 = The CAN bus is recessive (CAN_RX = '1') 0 = The CAN bus is dominant (CAN_RX = '0')	R	0
Tx1-0	[6:5]	Control of CAN_TX Pin 00 = Reset value, CAN_TX is controlled by the CAN Core 01 = Sample Point is monitored at CAN_TX pin 10 = CAN_TX pin drives a dominant ('0') value 11 = CAN_TX pin drives a recessive ('1') value	R/W	0
LBack	[4]	Loop Back Mode 1 = Enables Loop Back Mode 0 = Disables Loop Back Mode	R/W	0
Silent	[3]	Silent Mode 1 = The module is in Silent Mode 0 = Normal operation	R/W	0
Basic	[2]	Basic Mode 1 = IF2 Registers used as Tx Buffer, IF2 Registers used as Rx Buffer 0 = Basic Mode disabled	R/W	0
Reserved	[1:0]	Reserved	R	0

NOTE: Write access to the Test Register is enabled by setting bit Test in the CAN Control Register. The different test functions may be combined, but Tx1-0 ≠ "00" disturbs message transfer.

6.2.7 BRP Extension Register

- CAN0_BRP, R/W, Address = 0xEC70_0018
- CAN1_BRP, R/W, Address = 0xEC80_0018

CANn_BRP	Bit	Description	R/W	Reset Value
Reserved	[15:4]	Reserved	R	0
BRPE	[3:0]	Baud Rate Prescaler Extension 0x00-0x0F: By programming BREP the Baud Rate Prescaler is extended to values up to 1023. The actual interpretation by the hardware is that one more than the value programmed by BRPE (MSBs) and BRP(LSBs) is used.	R/W	0

6.2.8 Message Interface Register Sets

There are two sets of Interface Registers which are used to control the CPU access to the Message RAM. The Interface Registers avoid conflicts between CPU accesses to the Message RAM and CAN message reception and transmission by buffering the data to be transferred. A complete Message Object or parts of the Message Object may be transferred between the Message RAM and the IFx Message Buffer registers in one single transfer.

The function of the two interface register sets is identical (except for test mode Basic). One set of registers is used for data transfer to the Message RAM while the other set of registers is used for the data transfer from the Message RAM, allowing both processes to be interrupted by each other. Figure 6 shows an overview of the two Interface Register sets.

Each set of Interface Registers consists of Message Buffer Registers controlled by their own Command Registers. The Command Mask Register specifies the direction of the data transfer and which parts of a Message Object will be transferred. The Command Request Register is used to select a Message Object in the Message RAM as target or source for the transfer and to start the action specified in the Command Mask Register.

Base Offset	IF2 Register Set	Base Offset	IF2 Register Set
0x20	IF1 Command Request	0x80	IF2 Command Request
0x24	IF1 Command Mask	0x84	IF2 Command Mask
0x28	IF1 Mask1	0x88	IF2 Mask1
0x2C	IF1 Mask2	0x8C	IF2 Mask2
0x30	IF1 Arbitration 1	0x90	IF2 Arbitration 1
0x34	IF1 Arbitration 2	0x94	IF2 Arbitration 2
0x38	IF1 Message Control	0x98	IF2 Message Control
0x3C	IF1 Data A 1	0x9C	IF2 Data A 1
0x40	IF1 Data A 2	0xA0	IF2 Data A 2
0x44	IF1 Data B 1	0xA4	IF2 Data B 1
0x48	IF1 Data B 2	0xA8	IF2 Data B 2

6.2.9 IF1 Command Request Register

- CAN0_IF1, R/W, Address = 0xEC70_0020
- CAN1_IF1, R/W, Address = 0xEC80_0020

6.2.10 IF2 Command Request Register

- CAN0_IF2, R/W, Address = 0xEC70_0080
- CAN1_IF2, R/W, Address = 0xEC80_0080

CANn_IF1 CANn_IF2	Bit	Description	R/W	Reset Value
Busy	[15]	Busy Flag 1 = Set to one if written to the IFx Command Request Register 0 = Reset to zero if read/ write action is complete.	R	0
Reserved	[14:6]	Reserved	R	0
Message Number	[5:0]	0x01-0x20: Valid Message Number, the Message Object in the Message RAM is selected for data transfer 0x00: Not a valid Message Number, interpreted as 0x20 0x21-0x3F: Not a valid Message Number, interpreted as 0x01-0x1F	R/W	0x01

NOTE: If a Message Number that is not valid is written into the Command Request Register, the Message Number will be transformed into a valid value and that Message Object will be transferred.

6.2.11 IF1 Command Mask Register

- CAN0_IF1_CM, R/W, Address = 0xEC70_0024
- CAN1_IF1_CM, R/W, Address = 0xEC80_0024

6.2.12 IF2 Command Mask Register

- CAN0_IF2_CM, R/W, Address = 0xEC70_0084
- CAN1_IF2_CM, R/W, Address = 0xEC80_0084

CANn_IF1_CM CANn_IF2_CM	Bit	Description	R/W	Reset Value
Reserved	[15:8]	Reserved	R	0
WR/RD	[7]	Write / Read 1 = Write: Transfer data from the selected Message Buffer Registers to the Message Object addressed by the Command Request Register. 0 = Read: Transfer data from the Message Object addressed by the Command Request Register into the selected Message Buffer Register.	R/W	0

Mask	[6]	Access Mask Bits
		If (**Direction = Write**)
		1 = Transfer Identifier Mask + MDir + MXtd to Message Object
		0 = Mask bits unchanged
		If (**Direction = Read**)
		1 = Transfer Identifier Mask + MDir + MXtd to IFx Message Buffer Register
		0 = Mask bits unchanged
			R/W	0

Arb	[5]	Access Arbitration Bits
		If (**Direction = Write**)
		1 = Transfer Identifier + Dir + Xtd + MsgVal to message Object
		0 = Arbitration bits unchanged
		If (**Direction = Read**)
		1 = Transfer Identifier + Dir + Xtd + MsgVal to IFx Message Buffer Register
		0 = Arbitration bits unchanged
			R/W	0

Control	[4]	Access Control Bits
		If (**Direction = Write**)
		1 = Transfer Control Bits to message Object
		0 = Control bits unchanged
		If (**Direction = Read**)
		1= Transfer Control Bits to IFx Message Buffer Register
		0 = Control Bits unchanged
			R/W	0

ClrIntPnd	[3]	Clear Interrupt Pending Bit
		If (**Direction = Write**)
		This bit is ignored
		If (**Direction = Read**)
		1 = Clear IntPnd bit in the Message Object
		0 = IntPnd bit remains unchanged
			R/W	0

TxRqst/ NewDat	[2]	Access new Data Bit If (**Direction = Write**)	R/W	0

CANn_IF1_CM CANn_IF2_CM	Bit	Description	R/W	Reset Value
		1 = Set TxRqst bit 0 = TxRqst bit unchanged If (**Direction = Read**) 1 = Clear NewDat bit in the Message Object 0 = NewDat bit remains unchanged		

Data A	[1]	Access Data Bytes 0-3
		If (**Direction = Write**)
		1 = Transfer Data Bytes 0-3 to Message Object
		0 = Data Bytes 0-3 unchanged
		If (**Direction = Read**)
		1 = Transfer Data Bytes 0-3 to IFx Message Buffer Register
		0 = Data Bytes 0-3 unchanged
			R/W	0

Data B	[0]	Access Data Bytes 4-7
		If (**Direction = Write**)
		1 = Transfer Data Bytes 4-7 to Message Object
		0 = Data Bytes 4-7 unchanged
		If (**Direction = Read**)
		1 = Transfer Data Bytes 4-7 to IFx Message Buffer Register
		0 = Data Bytes 4-7 unchanged
			R/W	0

6.2.13 IF1 Mask 1 Register

- CAN0_IF1_M1, R/W, Address = 0xEC70_0028
- CAN1_IF1_M1, R/W, Address = 0xEC80_0028

6.2.14 IF2 Mask 1 Register

- CAN0_IF2_M1, R/W, Address = 0xEC70_0088
- CAN1_IF2_M1, R/W, Address = 0xEC80_0088

CANn_IF1_M1 CANn_IF2_M1	Bit	Description	R/W	Reset Value
Msk15-0	[15:0]	Identifier Mask 15-0 1 = The corresponding identifier bit is used for acceptance filtering. 0 The corresponding bit in the identifier of the message object cannot inhibit the match in the acceptance filtering.	R/W	0xFFFF

6.2.15 IF1 Mask 2 Register

- CAN0_IF1_M2, R/W, Address = 0xEC70_002C
- CAN1_IF1_M2, R/W, Address = 0xEC80_002C

6.2.16 IF2 Mask 2 Register

- CAN0_IF2_M2, R/W, Address = 0xEC70_008C
- CAN1_IF2_M2, R/W, Address = 0xEC80_008C

CANn_IF1_M2 CANn_IF2_M2	Bit	Description	R/W	Reset Value
MXtd	[15]	Mask Extended Identifier 1 = The extended identifier bit (IDE) is used for acceptance filtering. 0 = The extended identifier bit (IDE) has no effect on the acceptance filtering.	R/W	1
MDir	[14]	Mask Message Direction 1 = The message direction bit (Dir) is used for acceptance filtering. 0 = The message direction bit (Dir) has no effect on the acceptance filtering.	R/W	1
Reserved	[13]	Reserved	R	1
Msk28-16	[12:0]	Identifier Mask 28-16 1 = The corresponding identifier bit is used for acceptance filtering. 0 = The corresponding bit in the identifier of the message object cannot inhibit the match in the acceptance filtering.	R/W	0x1FFF

6.2.17 IF1 Arbitration 1 Register

- CAN0_IF1_A1, R/W, Address = 0xEC70_0030
- CAN1_IF1_A1, R/W, Address = 0xEC80_0030

6.2.18 IF2 Arbitration 1 Register

- CAN0_IF2_A1, R/W, Address = 0xEC70_0090
- CAN1_IF2_A1, R/W, Address = 0xEC80_0090

CANn_IF1_A1 CANn_IF2_A1	Bit	Description	R/W	Reset Value
ID15-0	[15:0]	Message Identifier 15-0	R/W	0

6.2.19 IF1 Arbitration 2 Register (CAN_IF1_A2, R/W, Address = 0xEC70_0034, 0xEC80_0034)

- CAN0_IF1_A2, R/W, Address = 0xEC70_0034
- CAN1_IF1_A2, R/W, Address = 0xEC80_0034

6.2.20 IF2 Arbitration 2 Register (CAN_IF2_A2, R/W, Address = 0xEC70_0094, 0xEC80_0094)

- CAN0_IF2_A2, R/W, Address = 0xEC70_0094
- CAN1_IF2_A2, R/W, Address = 0xEC80_0094

CANn_IF1_A2 CANn_IF2_A2	Bit	Description	R/W	Reset Value
MsgVal	[15]	Message Valid 1 = The Message Object is configured and should be considered by the Message Handler. 0 = The Message Object is ignored by the Message Handler.	R/W	0
Xtd	[14]	Extended Identifier 1 = The 29-bit("extended") Identifier is used for this Message Object. 0 = The 11-bit("standard") Identifier is used for this Message Object.	R/W	0
Dir	[13]	Message Direction 1 = Direction = transmit. On TxRqst, the respective Message Object is transmitted as a Data Frame. On reception of a Remote Frame with matching identifier, the TxRqst bit of this Message Object is set (if RmtEn=1'b1) 0 = Direction = receive. On TxRqst, a Remote Frame with the identifier of this Message Object is transmitted. On reception of a Data Frame with matching identifier, that message is stored in this Message Object	R/W	0
ID28-16	[12:0]	Message Identifier ID28-ID0: 29-bit Identifier ("Extended Frame") ID28-ID18:11-bit Identifier ("Standard Frame")	R/W	0

NOTES:

1. The CPU must reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN Control Register. This bit must also be reset before the identifier Id28-0, the control bits Xtd, Dir, or the Data Length Code DLC3-0 are modified, or if the Messages Object is no longer required.
2. If 11-bit ("standard") Identifiers are used for a Message Object, the identifiers of received Data Frames are written into bits ID28 to ID18. For acceptance filtering, only these bits together with mask bits Msk28 to Msk18 are considered.

6.2.21 IF1 Message Control Register (CAN_IF1_MC, R/W, Address = 0xEC70_0038, 0xEC80_0038)

- CAN0_IF1_MC, R/W, Address = 0xEC70_0038
- CAN1_IF1_MC, R/W, Address = 0xEC80_0038

6.2.22 IF2 Message Control Register (CAN_IF2_MC, R/W, Address = 0xEC70_0098, 0xEC80_0098)

- CAN0_IF2_MC, R/W, Address = 0xEC70_0098
- CAN1_IF2_MC, R/W, Address = 0xEC80_0098

CANn_IF1_MC CANn_IF2_MC	Bit	Description	R/W	Reset Value
NewDat	[15]	New Data 1 = The Message Handler or the CPU has written new data into the data portion of this Message Object. 0 = No new data has been written into the data portion of this Message Object by the Message Handler since last time this flag was cleared by the CPU.	R/W	0
MsgLst	[14]	Message Lost (only valid for Message Object with direction = receive) 1 = The Message Handler stores a new message into this object if NewDat was still set, the CPU has lost a message. 0 = No message lost since last time this bit was reset by the CPU.	R/W	00
IntPnd	[13]	Interrupt Pending 1 = This message object is the source of an interrupt. The Interrupt Identifier in the Interrupt Register will point to this message object if there is no other interrupt source with higher priority. 0 =This message object is not the source of an interrupt.	R/W	0
UMask	[12]	Use Acceptance Mask 1 = Use Mask (Msk28-0, MXtd and MDir) for acceptance filtering. 0 = Mask Ignored.	R/W	0
TxIE	[11]	Transmit Interrupt Enable 1 = IntPnd is set after a successful transmission of a frame. 0 = IntPnd is left unchanged after the successful transmission of a frame.	R/W	0
RxIE	[10]	Receive Interrupt Enable 1 = IntPnd is set after a successful reception of a frame. 0 = IntPnd is left unchanged after a successful reception of a frame.	R/W	0
RmtEn	[9]	Remote Enable 1 = At the reception of a Remote Frame, TxRqst is set	R/W	0

CANn_IF1_MC CANn_IF2_MC	Bit	Description	R/W	Reset Value
		0 = At the reception of a Remote Frame, TxRqst is left unchanged		
TxRqst	[8]	Transmit Request 1= The transmission of this Message Object is requested and is not yet done. 0 = This Message Object is not waiting for transmission.	R/W	0
EoB	[7]	End of Buffer 1 = Single Message Object or last Message Object of a FIFO Buffer. 0 = Message Object belongs to a FIFO Buffer and is not the last Message Object of that FIFO Buffer.	R/W	0
Reserved	[6:4]	Reserved	R	0
DLC3-0	[3:0]	Data Length Code Data 0: 1st data byte of a CAN Data Frame Data 1: 2nd data byte of a CAN Data Frame Data 2: 3rd data byte of a CAN Data Frame Data 3: 4th data byte of a CAN Data Frame Data 4: 5th data byte of a CAN Data Frame Data 5: 6th data byte of a CAN Data Frame Data 6: 7th data byte of a CAN Data Frame Data 7: 8th data byte of a CAN Data Frame	R/W	0

6.2.23 IF1 Message Data A1 (CAN_IF1_DA1, R/W, Address = 0xEC70_003C, 0xEC80_003C)

- CAN0_IF1_DA1, R/W, Address = 0xEC70_003C
- CAN1_IF1_DA1, R/W, Address = 0xEC80_003C

6.2.24 IF2 Message Data A1 (CAN_IF2_DA1, R/W, Address = 0xEC70_009C, 0xEC80_009C)

- CAN0_IF2_DA1, R/W, Address = 0xEC70_009C
- CAN1_IF2_DA1, R/W, Address = 0xEC80_009C

CANn_IF1_DA1 CANn_IF2_DA1	Bit	Description	R/W	Reset Value
Data1	[15:8]	Data1	R/W	0
Data0	[7:0]	Data0	R/W	0

6.2.25 IF1 Message Data A2 (CAN_IF1_DA2, R/W, Address = 0xEC70_0040, 0xEC80_0040)

- CAN0_IF1_A2, R/W, Address = 0xEC70_0040
- CAN1_IF1_A2, R/W, Address = 0xEC80_0040

6.2.26 IF2 Message Data A2 (CAN_IF2_DA2, R/W, Address = 0xEC70_00A0, 0xEC80_00A0)

- CAN0_IF2_A2, R/W, Address = 0xEC70_00A0
- CAN1_IF2_A2, R/W, Address = 0xEC80_00A0

CANn_IF1_A2 CANn_IF2_A2	Bit	Description	R/W	Reset Value
Data3	[15:8]	Data3	Rw	0
Data2	[7:0]	Data2	RW	0

6.2.27 IF1 Message Data B1 (CAN_IF1_DB1, R/W, Address = 0xEC70_0044, 0xEC80_0044)

- CAN0_IF1_DB1, R/W, Address = 0xEC70_0044
- CAN1_IF1_DB1, R/W, Address = 0xEC80_0044

6.2.28 IF2 Message Data B1 (CAN_IF2_DB1, R/W, Address = 00EC70_0xA4, 0xEC80_00A4)

- CAN0_IF2_DB1, R/W, Address = 0xEC70_00A4
- CAN1_IF2_DB1, R/W, Address = 0xEC80_00A4

CANn_IF1_DB1 CANn_IF2_DB1	Bit	Description	R/W	Reset Value
Data5	[15:8]	Data5	R/W	0
Data4	[7:0]	Data4	R/W	0

6.2.29 IF1 Message Data B2 (CAN_IF1_DB2, R/W, Address = 00EC70_0048, 0xEC80_0048)

- CAN0_IF1_DB2, R/W, Address = 0xEC70_0048
- CAN1_IF1_DB2, R/W, Address = 0xEC80_0048

6.2.30 IF2 Message Data B2 (CAN_IF2_DB2, R/W, Address = 00EC70_00A8, 0xEC80_00A8)

- CAN0_IF2_DB2, R/W, Address = 0xEC70_00A8
- CAN1_IF2_DB2, R/W, Address = 0xEC80_00A8

CANn_IF1_DB2 CANn_IF2_DB2	Bit	Description	R/W	Reset Value
Data7	[15:8]	Data7	RW	0
Data6	[7:0]	Data6	RW	0

6.2.31 Transmission Request Register 1 (CAN_TrReq1, R, Address = 00EC70_0100, 0xEC80_0100)

- CAN0_TrReq1, R/W, Address = 0xEC70_0100
- CAN1_TrReq1, R/W, Address = 0xEC80_0100

CANn_TrReq1	Bit	Description	R/W	Reset Value
TxRqst16-1	[15:0]	Transmission Request Bits (of all Message Objects) 1 = The transmission of this Message Object is requested and is not yet done. 0 = This Message Object is not waiting for transmission.	R	0x0000

6.2.32 Transmission Request Register 2 (CAN_TrReq2, R, Address = 00EC70_0104, 0xEC80_0104)

- CAN0_TrReq2, R/W, Address = 0xEC70_0104
- CAN1_TrReq2, R/W, Address = 0xEC80_0104

CANn_TrReq2	Bit	Description	R/W	Reset Value
TxRqst32-17	[15:0]	Transmission Request Bits (of all Message Objects) 1 = The transmission of this Message Object is requested and is not yet done. 0 = This Message Object is not waiting for transmission.	R	0

6.2.33 New Data Register 1 (CAN_ND1, R, Address = 00EC70_0120, 0xEC80_0120)

- CAN0_ND1, R/W, Address = 0xEC70_0120
- CAN1_ND1, R/W, Address = 0xEC80_0120

CANn_ND1	Bit	Description	R/W	Reset Value
NewDat16-1	[15:0]	New Data Bits (of all Message Objects) 1 = The Message Handler or the CPU has written new data into the data portion of this Message Object 0 = No new data has been written into the data portion of this Message Object by the Message Handler since last time this flag was cleared by the CPU.	R	0x0000

6.2.34 New Data Register 2 (CAN_ND2, R, Address = 00EC70_0124, 0xEC80_0124)

- CAN0_ND2, R/W, Address = 0xEC70_0124
- CAN1_ND2, R/W, Address = 0xEC80_0124

CANn_ND2	Bit	Description	R/W	Reset Value
NewDat32-17	[15:0]	New Data Bits (of all Message Objects) 1 = The Message Handler or the CPU has written new data into the data portion of this Message Object. 0 = No new data has been written into the data portion of this Message Object by the Message Handler since last time this flag was cleared by the CPU.	R	0

6.2.35 Interrupt Pending Register 1 (CAN_IntPen1, R, Address = 00EC70_0140, 0xEC80_0140)

- CAN0_IntPen1, R/W, Address = 0xEC70_0140
- CAN1_IntPen1, R/W, Address = 0xEC80_0140

CANn_IntPen1	Bit	Description	R/W	Reset Value
IntPnd16-1	[15:0]	Interrupt Pending Bits (of all Message Objects) 1 = This message object is the source of an interrupt. 0 = This message object is not the source of an interrupt.	R	0

6.2.36 Interrupt Pending Register 2 (CAN_IntPen2, R, Address = 00EC70_0144, 0xEC80_0144)

- CAN0_IntPen2, R/W, Address = 0xEC70_0144
- CAN1_IntPen2, R/W, Address = 0xEC80_0144

CANn_IntPen2	Bit	Description	R/W	Reset Value
IntPnd32-17	[15:0]	Interrupt Pending Bits (of all Message Objects) 1 = This message object is the source of an interrupt. 0 = This message object is not the source of an interrupt.	R	0

6.2.37 Message Valid Register 1 (CAN_MesVal1, R, Address = 00EC70_0160, 0xEC80_0160)

- CAN0_MesVal1, R/W, Address = 0xEC70_0160
- CAN1_MesVal1, R/W, Address = 0xEC80_0160

CANn_MesVal1	Bit	Description	R/W	Reset Value
MsgVal16-1	[15:0]	Message Valid Bits (of all Message Objects) 1 = This Message Object is configured and should be considered by the Message Handler. 0 = This Message Object is ignored by the Message Handler.	R	0

6.2.38 Message Valid Register 2 (CAN_MesVal2, R, Address = 00EC70_0164, 0xEC80_0164)

- CAN0_MesVal2, R/W, Address = 0xEC70_0164
- CAN1_MesVal2, R/W, Address = 0xEC80_0164

CANn_MesVal2	Bit	Description	R/W	Reset Value
MsgVal32-17	[15:0]	Message Valid Bits (of all Message Objects) 1 = This Message Object is configured and should be considered by the Message Handler. 0 = This Message Object is ignored by the Message Handler.	RO	0

8.6 MIPI HSI INTERFACE CONTROLLER

1 OVERVIEW

MIPI HSI is a high speed synchronous serial interface and is defined for communication between two ICs. The targeted scenario is an application IC and cellular modem IC communication. Data transaction model is peer-to-peer.

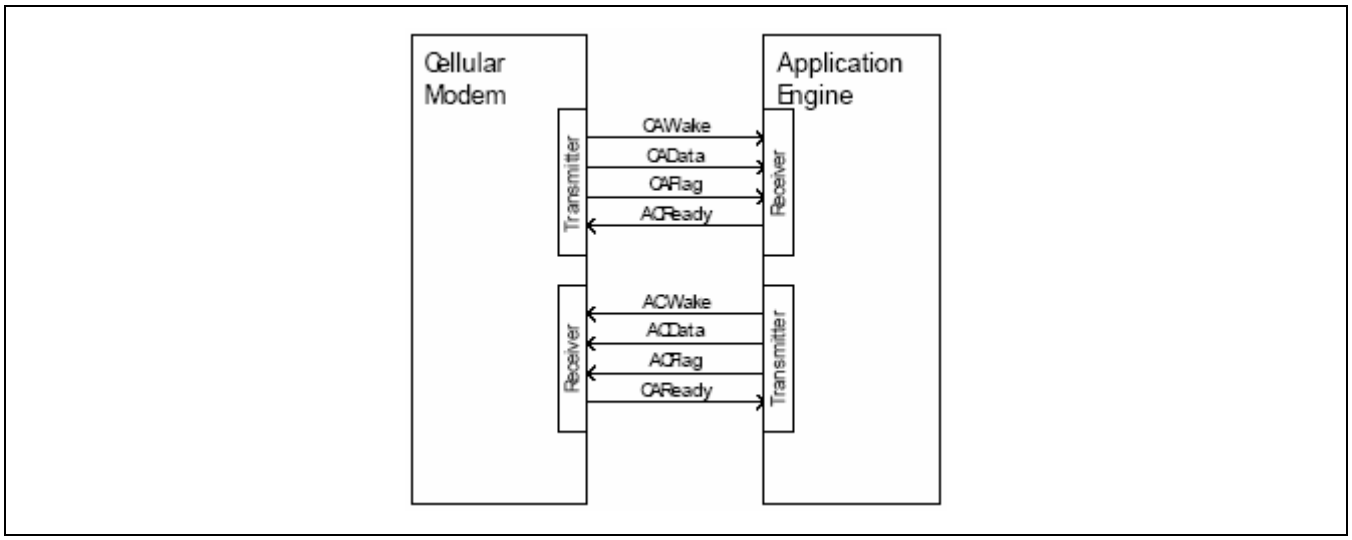


Figure 8.6-1 MIPI HSI Signal Definition

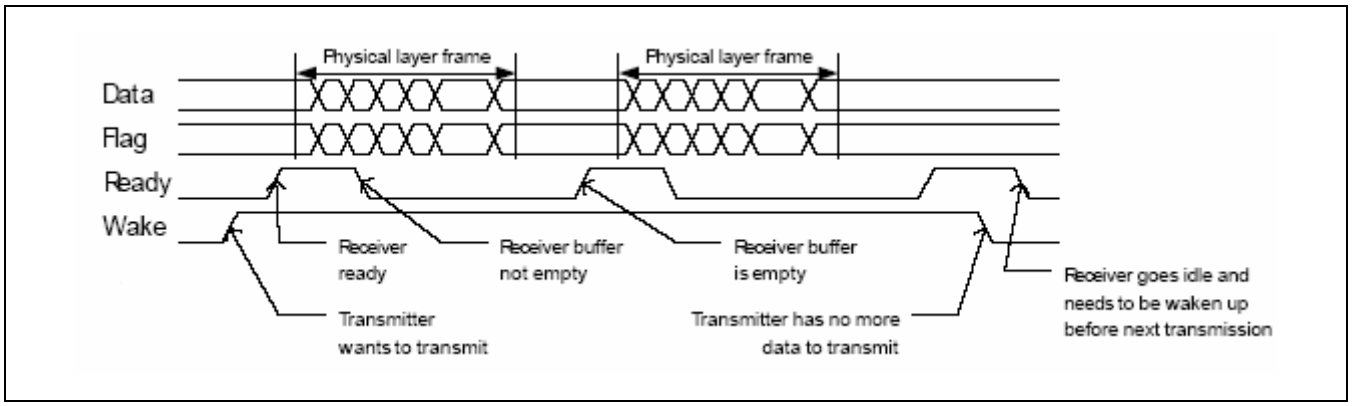


Figure 8.6-2 MIPI HSI Transmitting Example

2 FEATURES

The MIPI HSI Rx/ Tx controller features:

The MIPI HSI interface is uni-direction interface.

MIPI HSI maximum bandwidth is 200Mbps. The MIPI HSI Tx controller use PCLK for data transmitting.

Tx Module:

- Status Register
  - FIFO status (fifo full, fifo empty, fifo write point, fifo read point)
  - MIPI status (internal status: current status & next status)
- Configuration Register
  - Select Operation mode (Stream mode or Frame mode)
  - Fixed channel ID mode
  - Number of channel
  - Generated Error clear
  - TxHOLD state timer and enable
  - TxIDLE state timer and enable
  - TxREQ state timer and enable
- Interrupt Source Register
  - FIFO empty
  - Break frame transfer done
  - TxHOLD state timeout
  - TxIDLE state timeout
  - TxREQ state timeout
- Interrupt Mask Register
- Software Reset Register
- Channel ID Register
- Data Register
  - Tx FIFO input
  - Tx FIFO size (Flip-Flop FIFO, not memory)
    - 32-bit width X 32 depth (128-Byte)

Rx Module:

- Status Register
  - FIFO status (fifo full, fifo empty, fifo write point, fifo read point)
  - Sks..MIPI status (Internal status: Current status and Next status)
- Configuration Register 0
  - Select Operation mode (Stream mode or Frame mode)
  - Fixed channel ID mode
  - Number of channel
  - Generated Error clear
  - RxACK state timer and enable
  - Rx state timer
- Configuration Register 1
  - Rx FIFO clear
  - Rx FIFO timer and enable
- Interrupt Source Register
  - Rx FIFO full
  - Rx FIFO timeout
  - Data Receiving Done
  - Break frame received
  - Break frame receiving error
  - RxACK state timeout
  - Missed clock input
  - Added clock input
- Software Reset Register
- Channel ID Register
- Data Register
  - Rx FIFO input
  - Rx FIFO size (Flip-Flop FIFO, not memory)
    - 32-bit width X 64 depth (256-Byte)

3 BLOCK DIAGRAM

3.1 TOP-LEVEL BLOCK DIAGRAM

Rx module part and Tx module part basic architecture is similar.

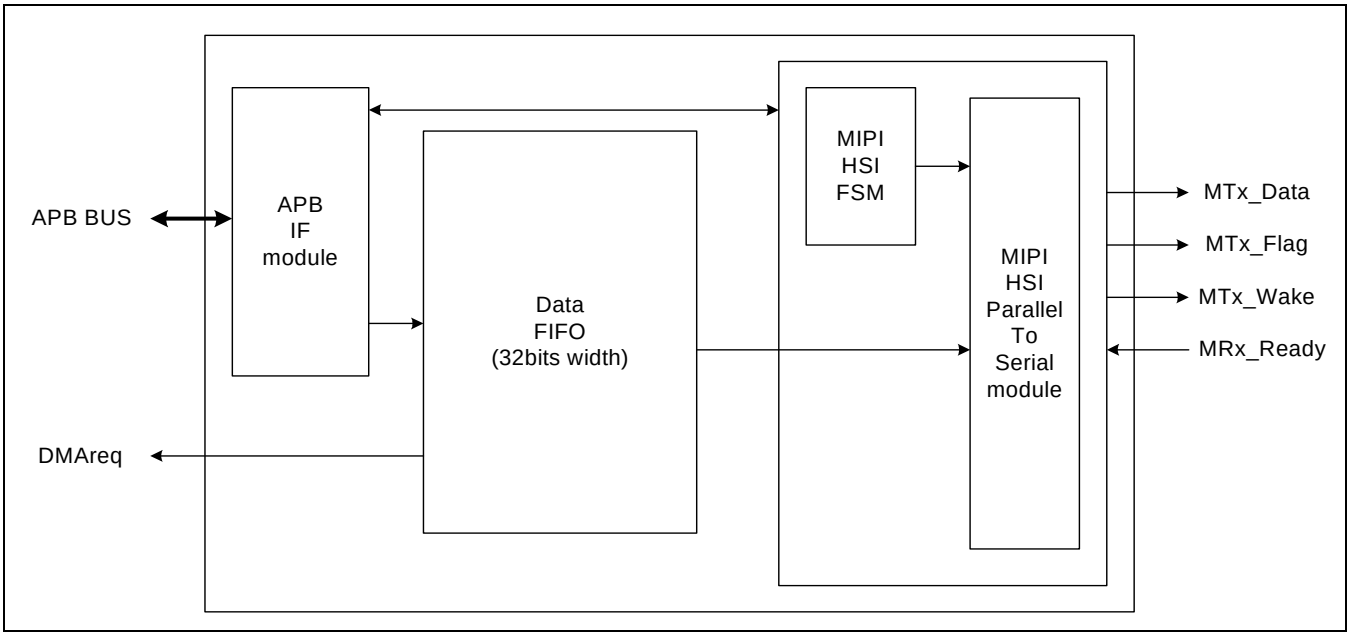


Figure 8.6-3 MIPI HSI Interface Controller Tx Module Top Block Diagram

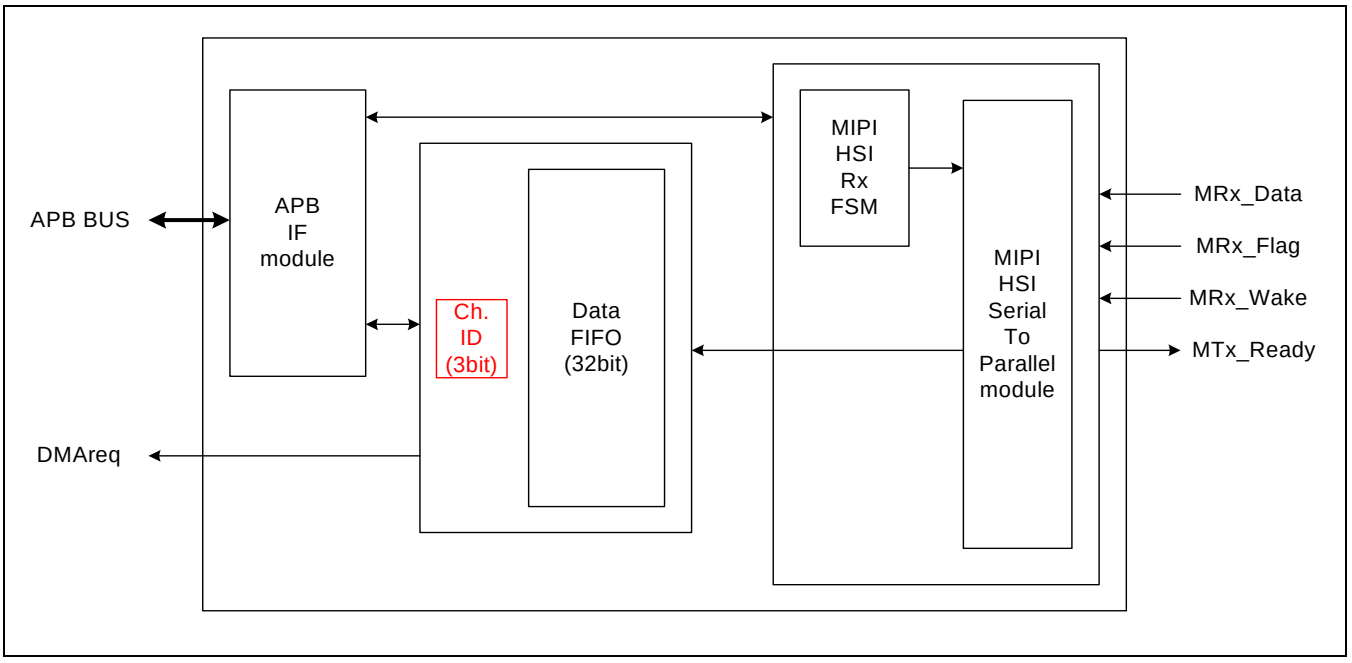


Figure 8.6-4 MIPI HSI Interface Controller Rx Module Top Block Diagram

3.1.1 Tx Module Part Parallel-to-Serial Block

A top-level block diagram of the PC card controller is shown below in Figure 8.6-5.

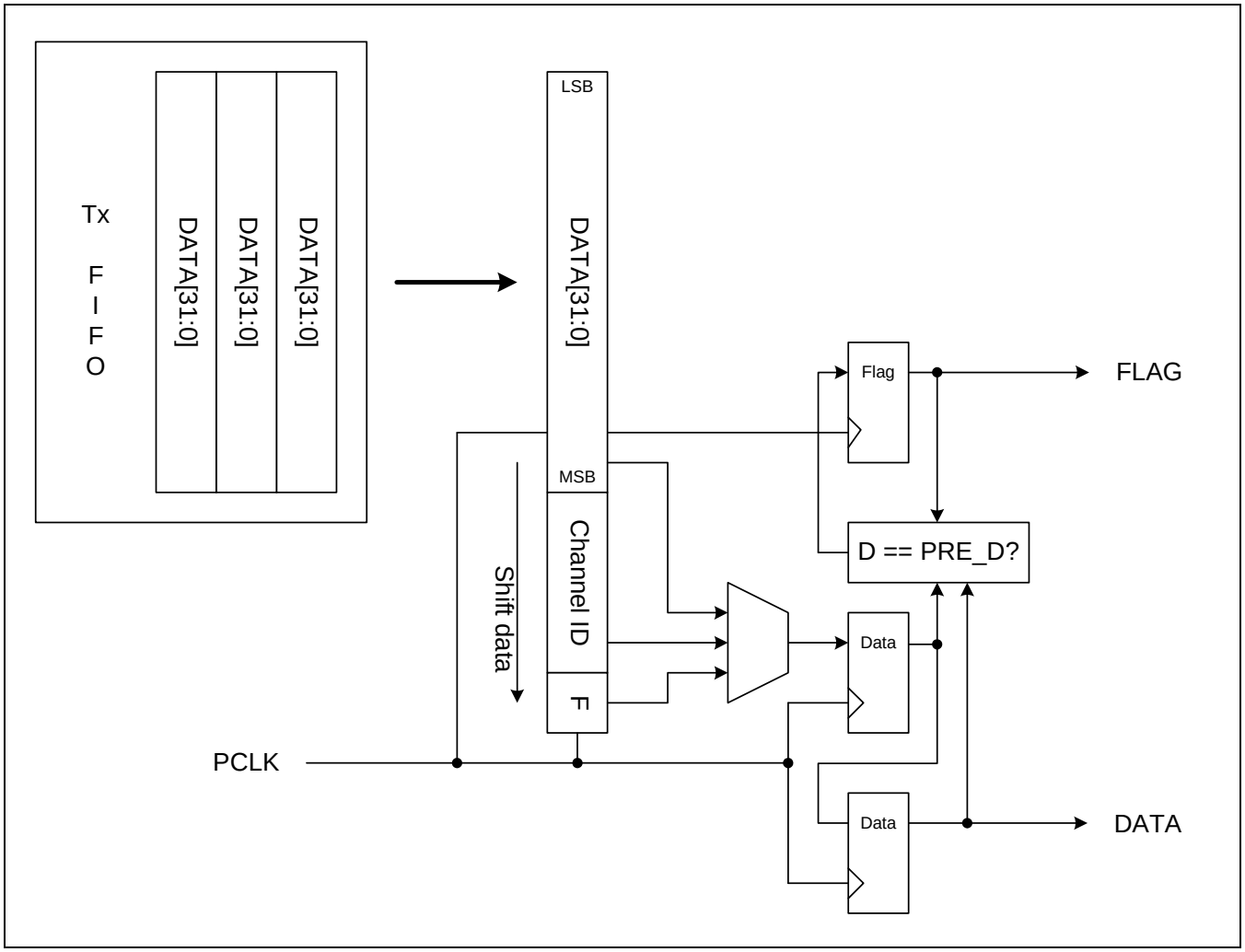


Figure 8.6-5 Parallel -to- Serial Block (Tx Module Part)

3.1.2 Rx Module Part Serial-to-Parallel Block

A top-level block diagram of the ATAPI controller is shown below in Figure 8.6-6.

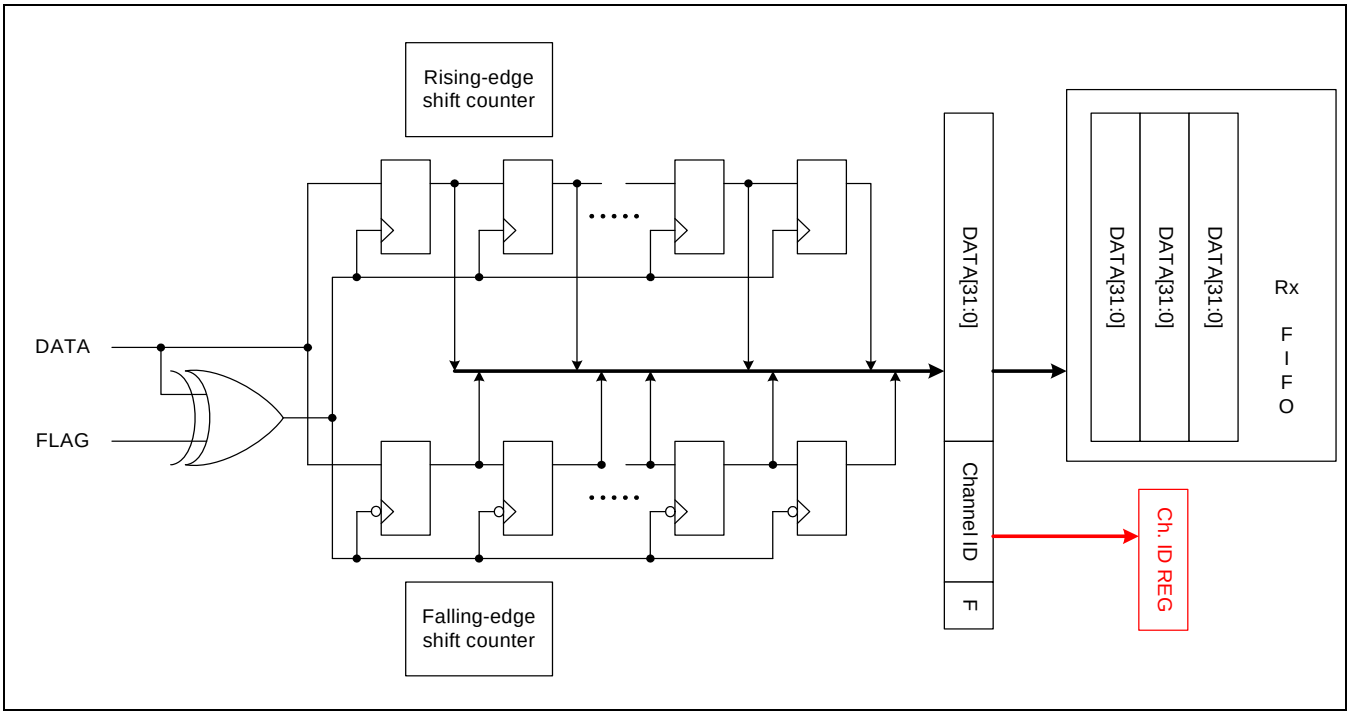


Figure 8.6-6 Serial-to-Parallel Block (Rx Module Part)

4 CLOCK SCHEME

4.1 TX MODULE PART PARALLEL-TO-SERIAL BLOCK

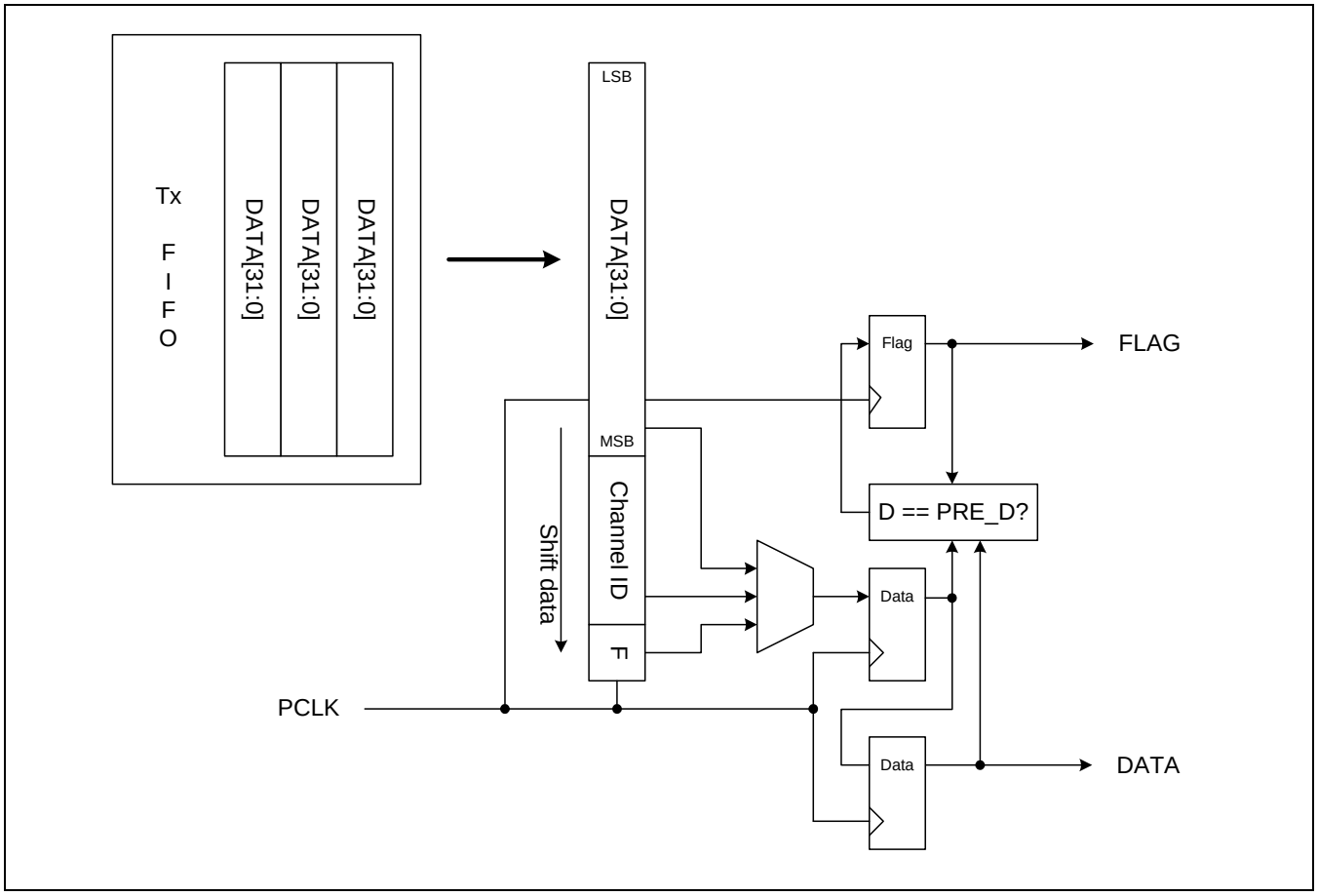


Figure 8.6-7 Parallel-to-Serial Block (Tx Module Part)

4.2 RX MODULE PART : SERIAL-TO-PARALLEL BLOCK

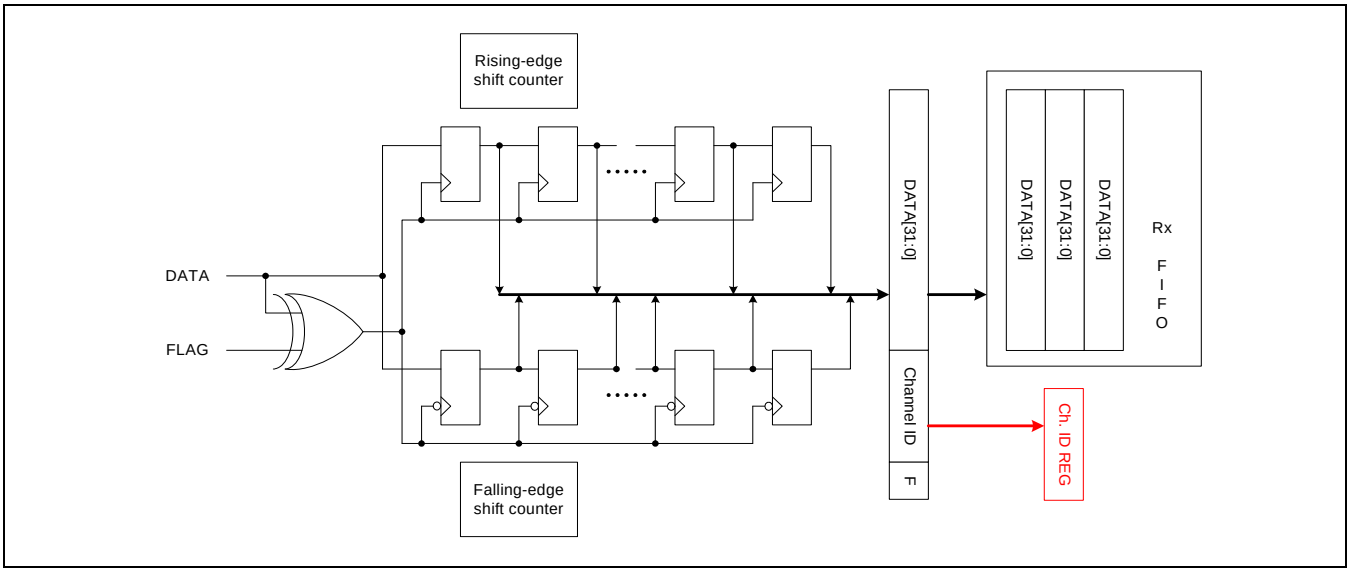


Figure 8.6-8 Serial-to-Parallel Block (Rx Module Part)

5 TIMING DIAGRAM

5.1 WAVEFORM

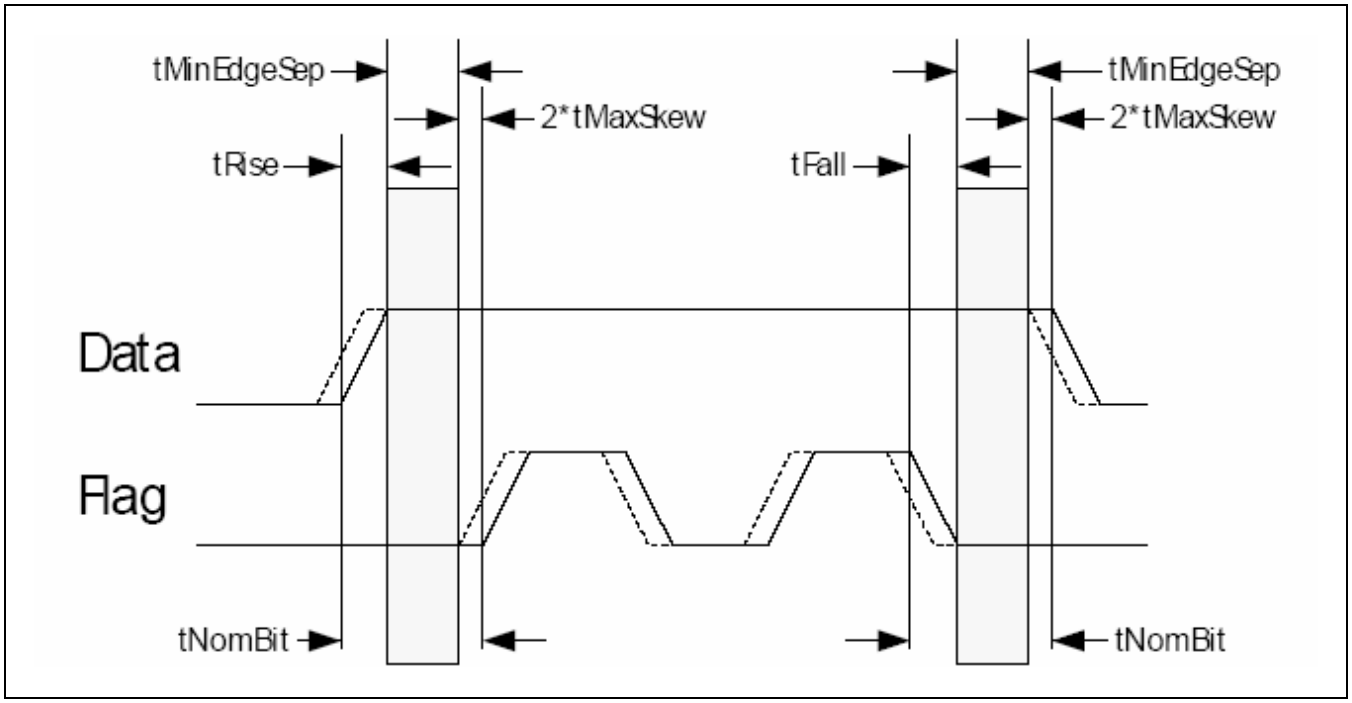


Figure 8.6-9 Waveform

5.2 SIGNAL TIMINGS

Table 8.6-1 Signal Timings

Parameter	Description	1 Mbit/s	100 Mbit/s	200 Mbit/s
TNomBit	Nominal bit time	1000 ns	10 ns	5 ns
TMinEdgeSep	Minimum allowed separation of DATA and FLAG signal transitions	500 ns	5 ns	2.5 ns
TMaxSkew	Maximum allowed time for combined skew and jitter	249 ns	1.5 ns	0.75 ns
tRise and tFall	Minimum allowed signal rise and fall time	2 ns	2 ns	1 ns

5.3 SINGLE/BURST CHANNEL ID MODE

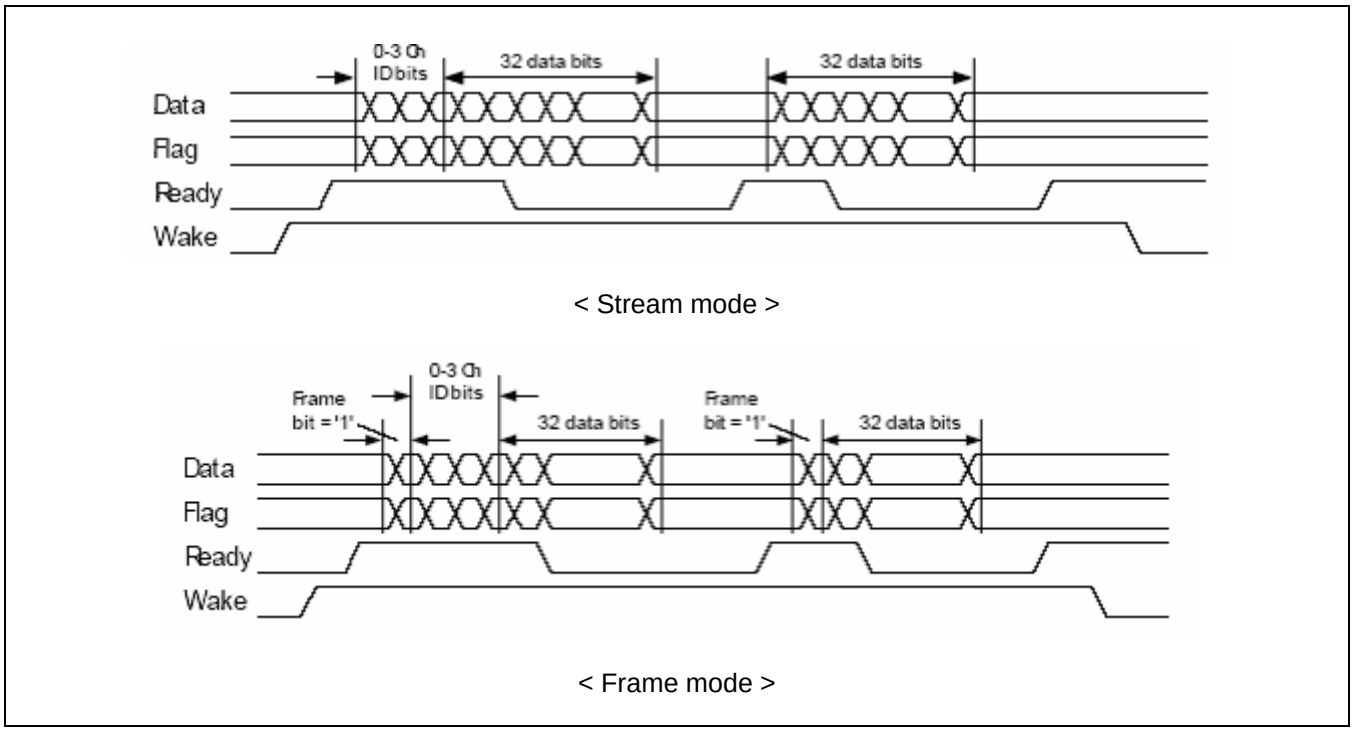


Figure 8.6-10 Example of Burst Channel ID Mode

If HSI is in Single channel ID mode, it attaches channel ID to each data whenever HSI transfers data. If HSI is in Burst channel ID mode, it attaches channel ID to beginning of data frame and transfers 32-bit data before it enters IDLE state. That is reason bandwidth of Burst channel ID mode is wider than that of Single channel ID mode.

5.4 STREAM MODE

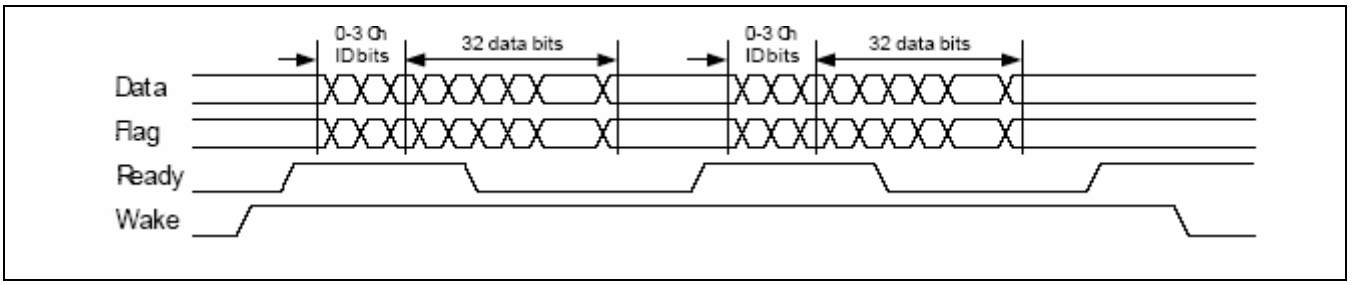


Figure 8.6-11 Example of Stream Mode

5.5 FRAME MODE

5.5.1 Normal Mode

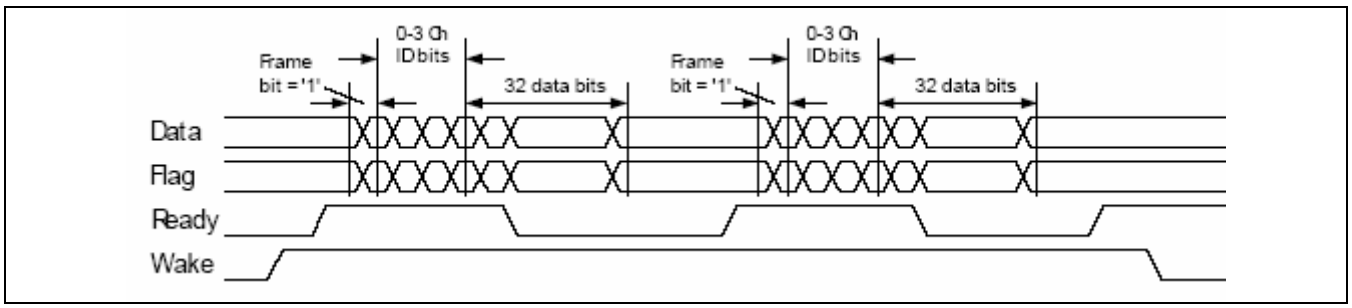


Figure 8.6-12 Example of Frame Mode (Normal Mode)

5.5.2 Break Frame

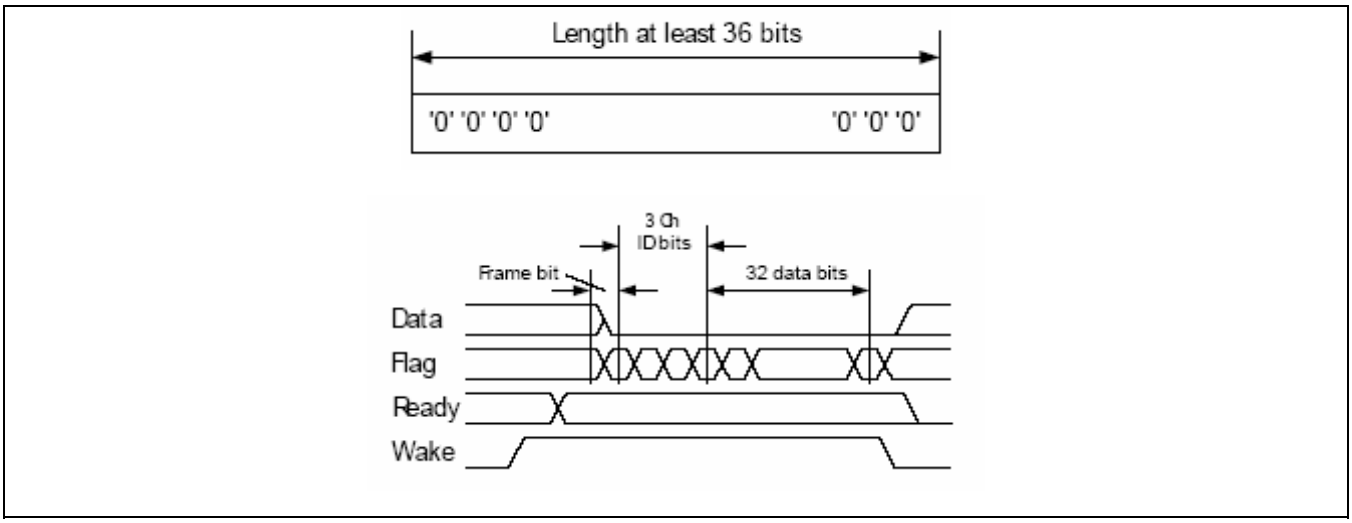


Figure 8.6-13 Break Frame

Flag line toggles until the transfer is end if Break frame transfers at least 36 zeros. Tx module transferring Break frame does not require to monitor the ready signal unlike normal mode.

6 I/O DESCRIPTION

Table 8.6-2 I/O Description

Function Signal	I/O	Description	Pad	Type
MIPI HSI interface Signals (Tx)				
HSI_TXD	O	MIPI HSI data line	XmsmA[0]	Muxed
HSI_TX_FLAG	O	MIPI HSI flag line	XmsmA[1]	Muxed
HSI_TX_WAKE	O	MIPI HSI wake up line to the other side Rx	XmsmA[2]	Muxed
HSI_TX_READY	I	MIPI HSI ready line from the other side Rx	XmsmA[3]	Muxed
MIPI HSI interface Signals (Rx)				
HSI_RXD	I	MIPI HSI data line	XmsmA[4]	Muxed
HSI_RX_FLAG	I	MIPI HSI flag line	XmsmA[5]	Muxed
HSI_RX_WAKE	I	MIPI HSI wake up line from the other side Tx	XmsmA[6]	Muxed
HSI_RX_READY	O	MIPI HSI ready line to the other side Tx	XmsmA[7]	Muxed

Note : Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals

7 REGISTER DESCRIPTION

7.1 REGISTER MAP TABLE

Table 8.6-3 Tx Controller Register Map Table

Register	Address	R/W	Description	Reset Value
STATUS_REG	0xEC90_0000	R	MIPI HSI Tx Controller Status Register	0x00010000
CONFIG_REG	0xEC90_0004	R/W	MIPI HSI Tx Controller Configuration Register	0xFFFFFF02
Reseved	0xEC90_0008	-	Reserved Register Area	0x00000000
INTSRC_REG	0xEC90_000C	R/W	MIPI HSI Tx Controller Interrupt Source Register	0x00000000
INTMSK_REG	0xEC90_0010	R/W	MIPI HSI Tx Controller Interrupt Mask Register	0x0000001F
SWRST_REG	0xEC90_0014	R/W	Tx Controller Software Reset	0x00000000
CHID_REG	0xEC90_0018	R/W	MIPI HSI Tx Controller Channel ID Register	0x00000000
DATA_REG	0xEC90_001C	W	MIPI HSI Tx Controller Data Register (FIFO input)	0x00000000

Table 8.6-4 Rx Controller Register Map Table

Register	Address	R/W	Description	Reset Value
STATUS_REG	0xECA0_0000	R	MIPI HSI Rx Controller Status Register	0x00010000
CONFIG0_REG	0xECA0_0004	R/W	MIPI HSI Rx Controller Configuration Register	0x0FFFFF02
CONFIG1_REG	0xECA0_0008	R/W	MIPI HSI Rx Controller Configuration Register	0x00FFFFFF
INTSRC_REG	0xECA0_000C	R/W	MIPI HSI Rx Controller Interrupt Source Register	0x00000000
INTMSK_REG	0xECA0_0010	R/W	MIPI HSI Rx Controller Interrupt Mask Register	0xFFFFFFFF
SWRST_REG	0xECA0_0014	R/W	Rx Controller Software Reset	0x00000000
CHID_REG	0xECA0_0018	R	MIPI HSI Rx Controller Channel ID Register	0x00000000
DATA_REG	0xECA0_ 001C	R	MIPI HSI Rx Controller Data Register (FIFO output)	0x00000000

8 INDIVIDUAL REGISTER DESCRIPTIONS (TX CONTROLLER)

8.1 MIPI HSI RX CONTROLLER STATUS REGISTER (STATUS_REG, R, ADDRESS = 0XEC90_0000)

STATUS_REG	Bit	Description	Reset Value
reserved	[31]	-	0x0
next_state	[30:28]	Next state*	0x0
reserved	[27]	-	0x0
current state	[26:24]	Current state*	0x0
reserved	[23:18]	-	0x00
FIFO_full	[17]	TxFIFO full 0 : FIFO not full 1 : FIFO full	0x0
FIFO_empty	[16]	TxFIFO empty 0 : FIFO not empty 1 : FIFO empty	0x1
reserved	[15:13]	-	0x0
tx_rd_point	[12:8]	TxFIFO read point	0x00
reserved	[7:5]	-	0x0
tx_wr_point	[4:0]	TxFIFO write point	0x00

- State register value

  000 : IDLE 001 : TxREQ

  010 : Tx 011 : TxHOLD

  100 : TxIDLE 101 : Reserved state

  110 : TxBRK 111 : TxERR

8.2 MIPI HSI TX CONTROLLER CONFIGURATION REGISTER(CONFIG_REG, R/W, ADDRESS = 0XEC90_0004)

CONFIG_REG	Bit	Description	Reset Value
TxHOLD time	[31:24]	TxHOLD state timer setting value	0xFF
TxIDLE time	[23:16]	TxIDLE state timer setting value	0xFF
TxREQ time	[15:8]	TxREQ state timer setting value	0xFF
TxHOLD time_en	[7]	Enable TxHOLD state timer 0 : Disables 1 : Enables	0x0
TxIDLE time_en	[6]	Enable TxIDLE state timer 0 : Disables 1 : Enables	0x0
TxREQ time_en	[5]	Enable TxREQ state timer 0 : Disables 1 : Enables	0x0
Err_clr	[4]	Clear generated Error 0 : Stay 1 : Clear	0x0
Num of CHID	[3:2]	Number of channel ID	0x0
Burst_mode	[1]	Fixed channel ID mode 0 : Burst channel ID mode 1 : Single channel ID mode	0x1
Frame_mode	[0]	Frame mode 0 : Stream Mode 1 : Frame Mode	0x0

8.3 MIPI HSI TX CONTROLLER INTERRUPT SOURCE REGISTER(INTSRC_REG, R/W, ADDRESS = 0XEC90_000C)

INTSRC_REG	Bits	Description	Reset Value
Reserved	[31:5]	-	0x0000001
TxH_timeout	[4]	TxHOLD state timeout interrupt (set '1' to clear)	0x0
TxI_timeout	[3]	TxIDLE state timeout interrupt (set '1' to clear)	0x0
TxR_timeout	[2]	TxREQ state timeout interrupt (set '1' to clear)	0x0
Brframe_end	[1]	Break frame transfer – done in Frame mode (set '1' to clear)	0x0
TxFIFO_empty	[0]	TxFIFO empty interrupt (set '1' to clear)	0x0(behave 0x1)

8.4 MIPI HSI TX CONTROLLER INTERRUPT MASK REGISTER (INTMSK_REG, R/W, ADDRESS = 0XEC90_0010)

INTMSK_REG	Bit	Description	Reset Value
DMA_req_en	[31]	Enables DMA request signal 0: Enables 1: Disables	0x1
Reserved	[30:5]	-	0x0000000
TxH_timeout_msk	[4]	TxHOLD state timeout interrupt mask 0 : Unmask 1 : Mask	0x1
TxI_timeout_msk	[3]	TxIDLE state timeout interrupt mask 0 : Unmask 1 : Mask	0x1
TxR_timeout_msk	[2]	TxREQ state timeout interrupt mask 0 : Unmask 1 : Mask	0x1
Brframe_end_msk	[1]	Break frame transfer–done mask (in Frame mode) 0 : Unmask 1 : Mask	0x1
TxFIFO_empty_msk	[0]	TxFIFO empty interrupt mask 0 : Unmask 1 : Mask	0x1

8.5 TX CONTROLLER SOFTWARE RESET (SWRST_REG, R/W, ADDRESS = 0XEC90_0014)

SWRST_REG	Bit	Description	Reset Value
Reserved	[31:1]	-	0x00000000
Sw_rst	[0]	Software reset 0 : Set 1 : Reset	0x0

8.6 MIPI HSI TX CONTROLLER CHANNEL ID REGISTER (CHID_REG, R/W, ADDRESS = 0XEC90_0018)

CHID_REG	Bit	Description	Reset Value
Break_frame	[31]	Break frame transfer in Frame mode In auto clear mode, this bit is automatically cleared. But the other mode, TxDATA send '0' stream during setting '1' at br_frame_clr bit.	0x0
Auto_clr	[30]	Break frame auto clear bit 0 : Auto clear & TxBRK state end 1 : Auto clear disable & TxBRK state continue	0x0
Br_frame_clr	[29]	Stop break frame continuing transfer	0x0
Reserved	[28:3]	-	0x0000000
CHID	[2:0]	Channel ID	0x0

8.7 MIPI HSI TX CONTROLLER DATA REGISTER (FIFO INPUT)(DATA_REG, W, ADDRESS = 0XEC90_001C)

DATA_REG	Bit	Description	Reset Value
TxFIFO in	[31:0]	TxFIFO data input for transmitting	0x0

8.8 MIPI HSI RX CONTROLLER STATUS REGISTER (STATUS_REG, R, ADDRESS = 0XECA0_0000)

STATUS_REG	Bit	Description	Reset Value
Reserved	[31]	-	0x0
Next_state	[30:28]	Next state*	0x0
Reserved	[27]	-	0x0
Curr_state	[26:24]	Current state*	0x0
Reserved	[23:19]	-	0x00
FIFO_timeout	[18]	RxFIFO read timeout 0 : In time 1 : Time out	0x0
FIFO_full	[17]	RxFIFO full 0 : FIFO not full 1: FIFO full	0x0
FIFO_empty	[16]	RxFIFO empty 0 : FIFO not empty 1: FIFO empty	0x1
Reserved	[15:14]	-	0x0
Rx_rd_point	[13:8]	RxFIFO read point	0x00
Reserved	[7:6]	-	0x0
Rx_wr_point	[5:0]	RxFIFO write point	0x00

- State register value

  000 : IDLE 001 : RxACK

  010 : Rx 011 : RxHOLD

  100 : RxBREAK 101 : Reserved state

  110 : RxRST 111 : RxERR

8.9 MIPI HSI RX CONTROLLER CONFIGURATION REGISTER (CONFIG0_REG, R/W , ADDRESS = 0XECA0_0004)

CONFIG0_REG	Bit	Description	Reset Value
Reserved	[31:30]	-	0x0
DREQ_thres_val	[29:28]	DMA request threshold value DMA request signal is active if valid data in FIFO is 0x00 : full 0x01 : more than 4word 0x10 : more than 8word 0x11 : more than 16word	0x00
Rx_state time	[27:16]	Rx state timer setting value	0xFFF
RxACK time	[15:8]	RxACK state timer setting value	0xFF
Reserved	[7]	-	0x0
RxACK time_en	[6]	Enable RxACK state timer 0 : Disables 1 :Enables	0x0
Break_clr	[5]	RxBREAK state clear bit 0 : Disables 1 : Enables	0x0
Err_clr	[4]	Generated Error clear	0x0
Num of CHID	[3:2]	Number of channel ID	0x0
Burst_mode	[1]	Fixed channel ID mode 0 : Burst channel ID mode 1 : Single channel ID mode	0x1
Frame_mode	[0]	Frame mode 0 : Stream mode 1 : Frame mode	0x0

8.10 MIPI HSI RX CONTROLLER CONFIGURATION REGISTER (CONFIG1_REG, R/W, ADDRESS = 0XECA0_0008)

CONFIG1_REG	Bit	Description	Reset Value
RxFIFO_clr	[31]	Break frame receiving timer setting value	0x0
Reserved	[30:28]	-	0x0
RxFIFO_timer_en	[27]	Enables RxFIFO timer	0x0
Reserved	[26:24]	-	0x0
RxFIFO_time	[23:0]	RxFIFO timer setting value	0xFFFFFF

8.11 MIPI HSI RX CONTROLLER INTERRUPT SOURCE REGISTER (INTSRC_REG, R/W, ADDRESS = 0XECA0_000C)

INTSRC_REG	Bit	Description	Reset Value
Reserved	[31:10]	-	0x000000
RxWakeup	[9]	Received Rx wake-up signal (set '1' for clearing)	0x0
Reserved	[8]	-	0x000000
Break_done	[7]	Received Break frame in Frame mode (set '1' to clear)	0x0
Added_clock	[6]	Added clock input (set '1' to clear)	0x0
Missed_clock	[5]	Missed clock input interrupt (set '1' to clear)	0x0
RxACK_timeout	[4]	RxACK state timeout interrupt (set '1' to clear)	0x0
Brframe_err	[3]	Received data is not break frame. (set '1' to clear)	0x0
RxDONE	[2]	Data receiving is Done. (set '1' to clear)	0x0
RxFIFO_timeout	[1]	RxFIFO timeout but RxFIFO is not empty. (set '1' to clear)	0x0
RxFIFO_full	[0]	RxFIFO full interrupt (set '1' to clear)	0x0

8.12 MIPI HSI RX CONTROLLER INTERRUPT MASK REGISTER (INTMSK_REG, R/W, ADDRESS = 0XECA0_0010)

INTMSK_REG	Bit	Description	Reset Value
DMA_req_en	[31]	Enable DMA request signal 0: Enables 1: Disables	0x1
Reserved	[30:10]	-	0x1fffff
RxWakeup_msk	[9]	Enables RX Wake-up Interrupt 0: Enables 1: Disables	0x1
wakeup_enn	[8]	Enables MIPI wake up 0 : Enables 1 : Disables	0x1
Break_done_msk	[7]	Break frame done interrupt mask 0 : Unmask 1 : Mask	0x1
Added_clock_msk	[6]	Added clock input interrupt mask 0 : Unmask 1 : Mask	0x1
Missed_clock_msk	[5]	Missed clock input interrupt mask 0 : Unmask 1 : Mask	0x1
RxACK_timeout_msk	[4]	RxACK state timeout interrupt mask 0 : Unmask 1 : Mask	0x1
Brframe_err_msk	[3]	Break frame err interrupt mask 0 : Unmask 1 : Mask	0x1
RxDONE_msk	[2]	RxDONE interrupt mask 0 : Unmask 1 : Mask	0x1
RxFIFO_timeout_msk	[1]	RxFIFO_timeout interrupt mask 0 : Unmask 1 : Mask	0x1
RxFIFO_full_msk	[0]	RxFIFO full interrupt mask 0 : Unmask 1 : Mask	0x1

8.13 RX CONTROLLER SOFTWARE RESET (SWRST_REG, R/W, ADDRESS = 0XECA0_0014)

SWRST_REG	Bit	Description	Reset Value
Reserved	[31:1]	-	0x00000000
Sw_rst	[0]	Software reset 0 : Set 1 : Reset	0x0

8.14 MIPI HSI RX CONTROLLER CHANNEL ID REGISTER (CHID_REG, R, ADDRESS = 0XECA0_0018)

CHID_REG	Bit	Description	Reset Value
Reserved	[31:3]	-	0x0000000
CURR_ID	[2:0]	Current Channel ID	0x0

8.15 MIPI HSI RX CONTROLLER DATA REGISTER (FIFO OUTPUT) (DATA_REG, R, ADDRESS = 0XECA0_ 001C)

DATA_REG	Bit	Description	Reset Value
RxFIFO out	[31:0]	RxFIFO data output	0x0

9 PROGRAMMING GUIDE

9.1 BASIC DRAWING FUNCTION

9.1.1 Tx Module Programming Guide Flow Chart

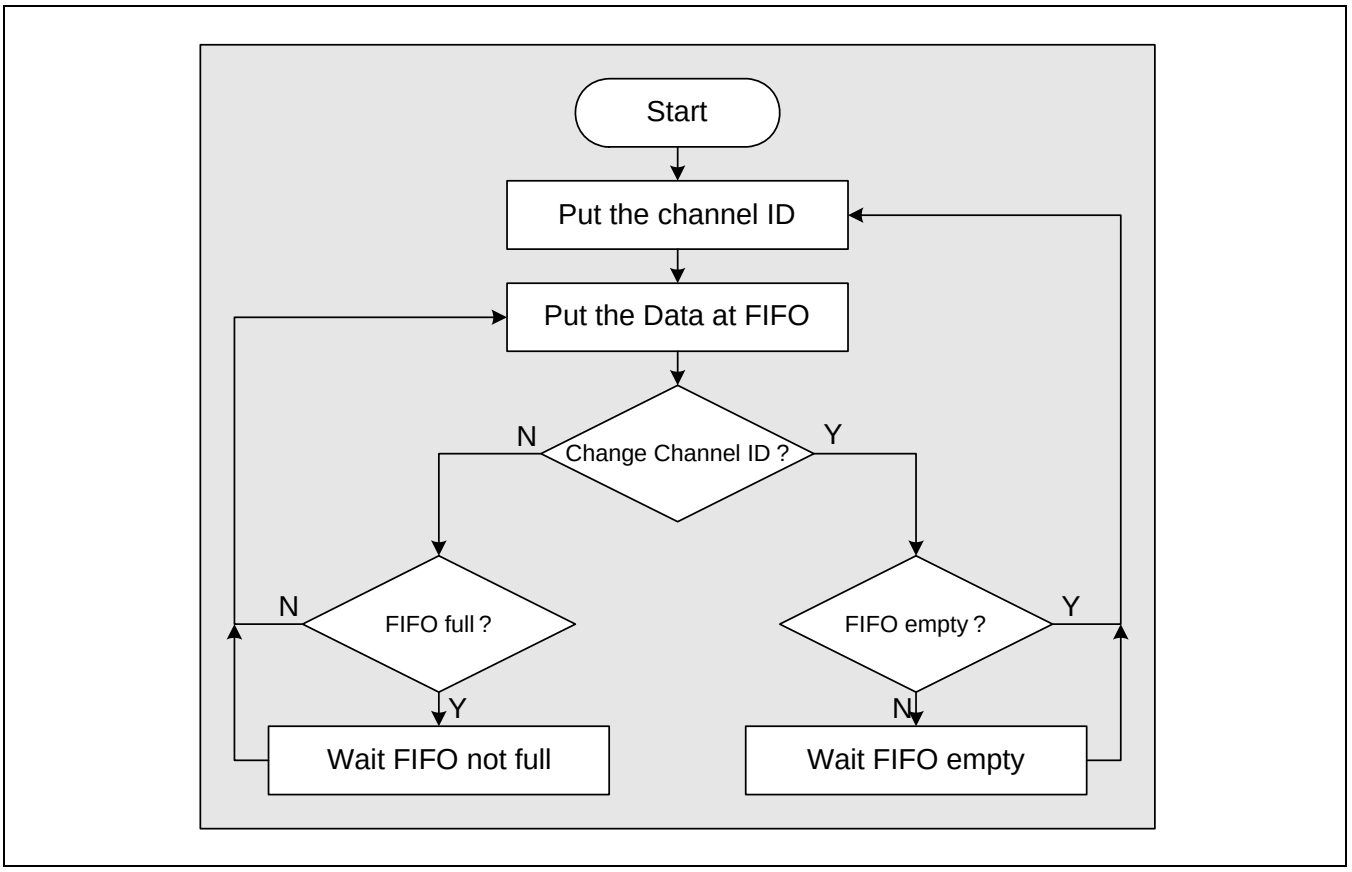


Figure 8.6-14 Basic Tx Module Programming Flow Chart

9.1.2 Rx Module Programming Guide Flow Chart

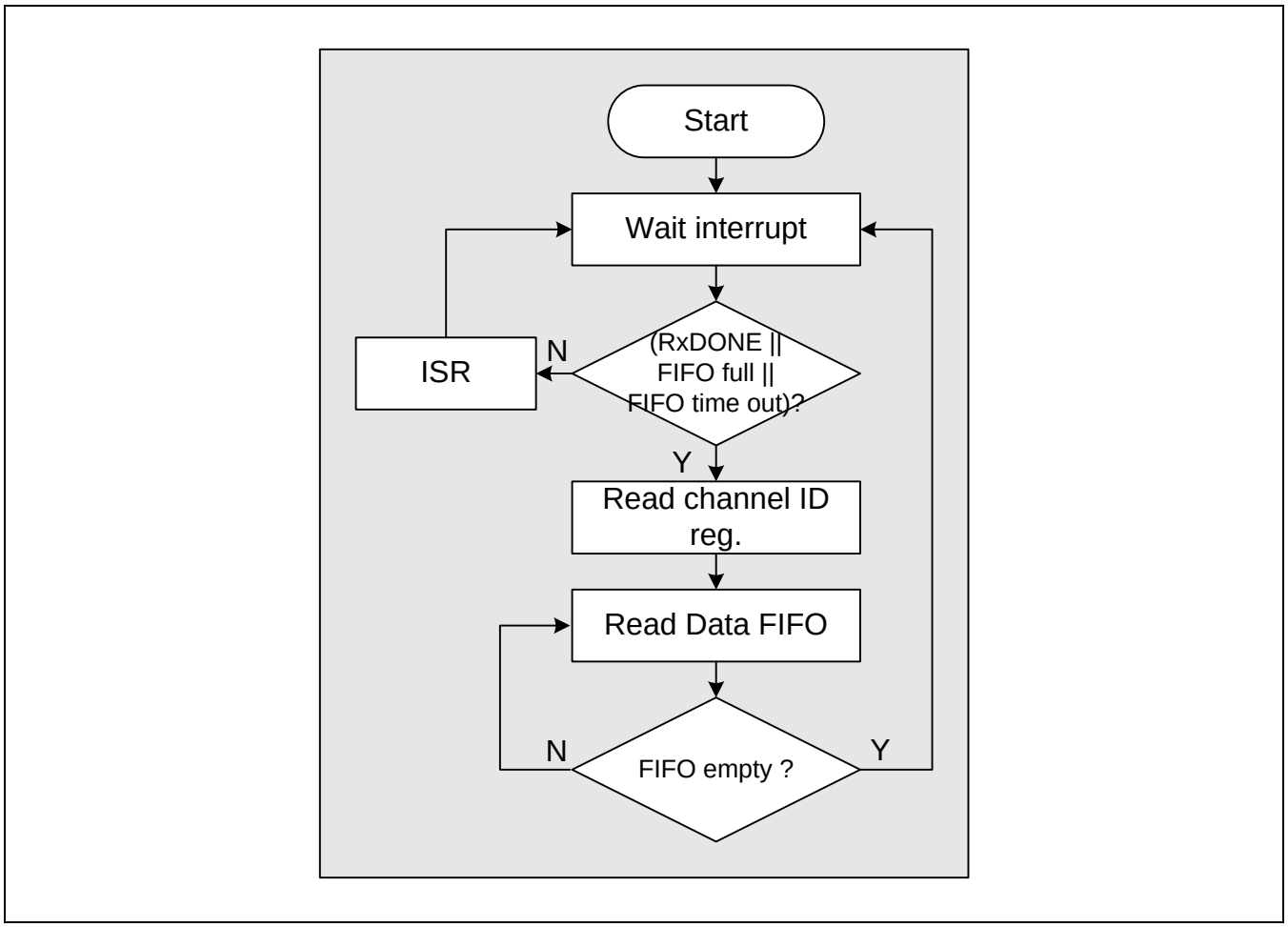


Figure 8.6-15 Basic Rx Module Programming Flow Chart

NOTES

8.7 MIPI DSIM

1 ARCHITECTURE

1.1 FEATURES

The MIPI DSIM features include:

- Compliant to MIPI DSI Standard Specification V1.01r02
  - ♦ Maximum resolution range: up to XGA (1028x768)
  - ♦ Supports 1, 2 or 3 data lanes.
  - ♦ Supports pixel format: 16bpp, 18bpp packed, 18bpp loosely packed (3 byte format), and 24bpp
- Interfaces
  - ♦ Compatible to Protocol-to-PHY Interface (PPI) in MIPI D-PHY Specification V0.86
  - ♦ RGB Interface for Video Image Input from display controller
  - ♦ I80 Interface for Command mode Image input from display controller
  - ♦ PMS control interface for PLL to configure byte clock frequency
- Prescaler to generate escape clock from byte clock

1.2 BLOCK DIAGRAM

1.2.1 Total System Block Diagram

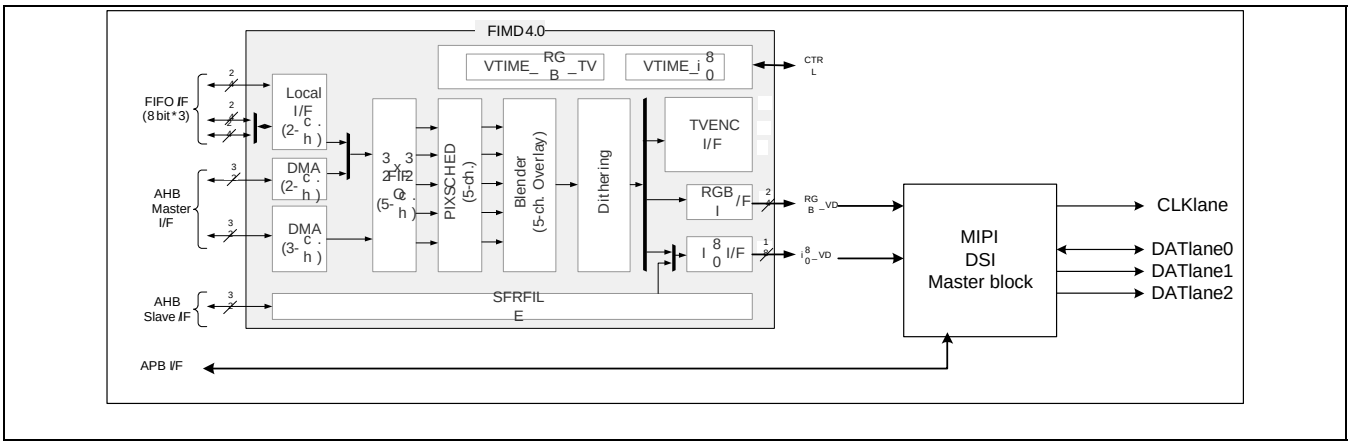


Figure 8.7-1 MIPI DSI System Block Diagram

1.2.2 MIPI DSI Master & D-PHY I/F Block Diagram

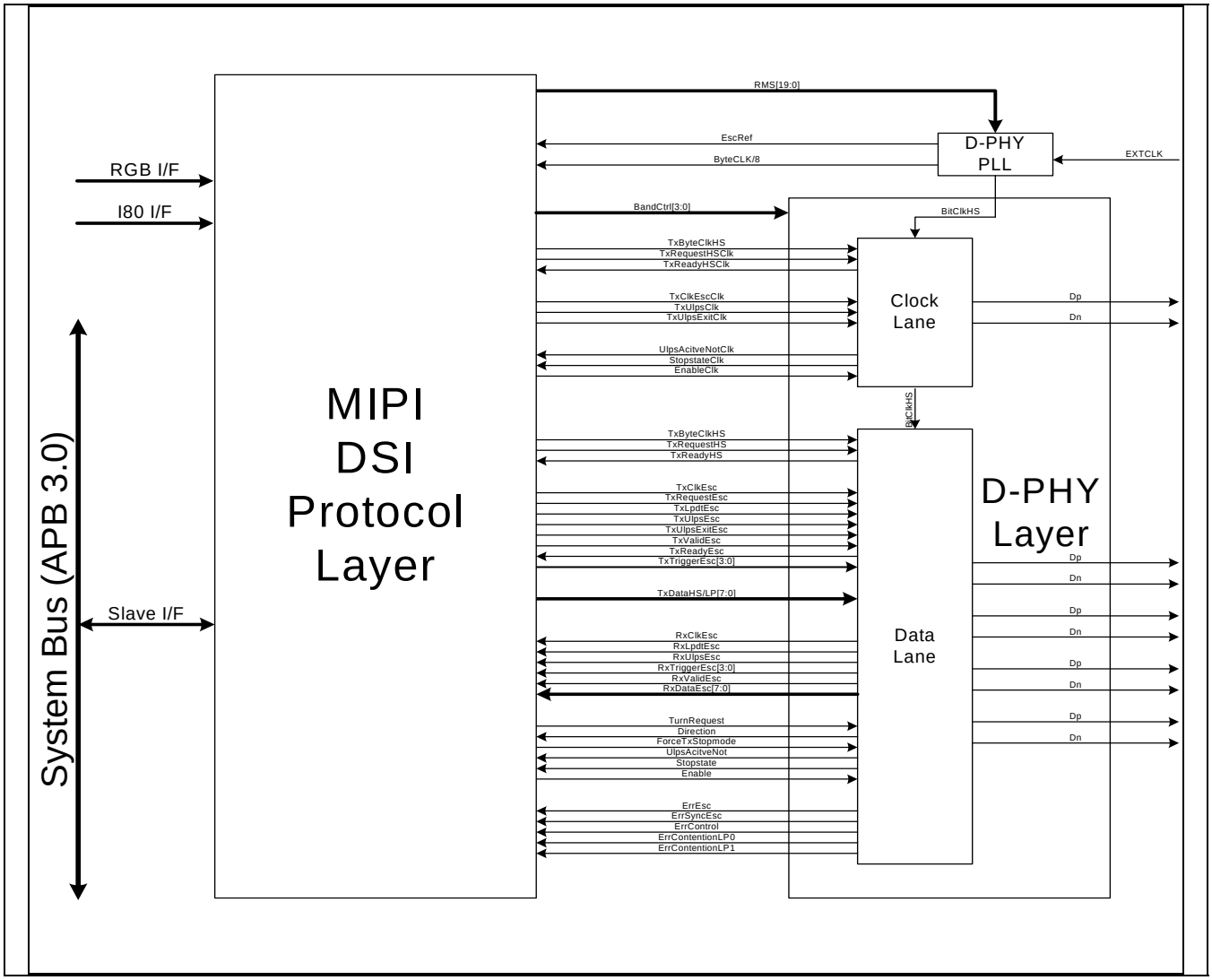


Figure 8.7-2 PPI I/F Block Diagram

1.2.3 Internal Primary FIFOs

There are some configurable-sized primary FIFOs. Table 8.7-1 describes these primary FIFOs.

Table 8.7-1 Internal Primary FIFO list

Port	FIFO type	Size	Description
Main display	Packet Header FIFO	3byte X 64 depth	Packet header FIFO for main display
	Payload FIFO	4byteX 1024 depth	Payload FIFO for main display image
Sub display for I80 I/F image data	Packet Header FIFO	3byte X 4 depth	Packet header FIFO for I80 I/F sub display
	Payload FIFO	4byte X 512 depth	Payload FIFO for I80 I/F sub display image
Command for I80 I/F command	Packet Header FIFO	3byte X 16 depth	Packet header FIFO for I80 I/F command packet
	Payload FIFO	4byte X 16 depth	Payload FIFO for I80 I/F command long packet payload
SFR for general packets	Packet Header FIFO	3byte X 16 depth	Packet header FIFO for general packet
	Payload FIFO	4byte X 512 depth	Payload FIFO for general long packet
RX FIFO	Packet header and Payload FIFO	4byte X 64 depth	Rx FIFO for LPDR. This FIFO is common for packet header and payload

1.2.4 Packet Header Arbitration

There are four-packet headers FIFOs for Tx: Main display, Sub display, I80 I/F command and SFR FIFO.
Main and sub display packet header FIFO transfers image data. I80 I/F command packet header FIFO transfers command packets. SFR packet header FIFO transfers command packets, sub display image data (in Video mode) and etc.
Packet header arbiter has "Fixed priority" algorithm. Priority order is main display, sub display, I80 I/F command and SFR packet header FIFO.
In Video mode, sub display and I80 I/F FIFO are not used. SFR packet header FIFO requests, if main display FIFO is empty (no request) in not-active image region.

1.2.5 RxFIFO Structure

RxFIFO is an SFR to read the packets received via Low Power Data Receiving. RxFIFO is an asynchronous FIFO with ByteClk and PCLK as input clock and output clock, respectively. Rx data is synchronized to RxClk. RXBUF has 4 Rx Byte buffer for aligning from byte to word.
The packet header of all packets stored in RXFIFO are word-aligned, i.e, the first byte of a packet is always stored in LSByte. For example, if a long packet has 7-byte payload, the last byte is filled with dummy byte and the next packet are stored in the next word as shown in Figure 8.7-. Note that CRC data is not stored in RXFIFO.

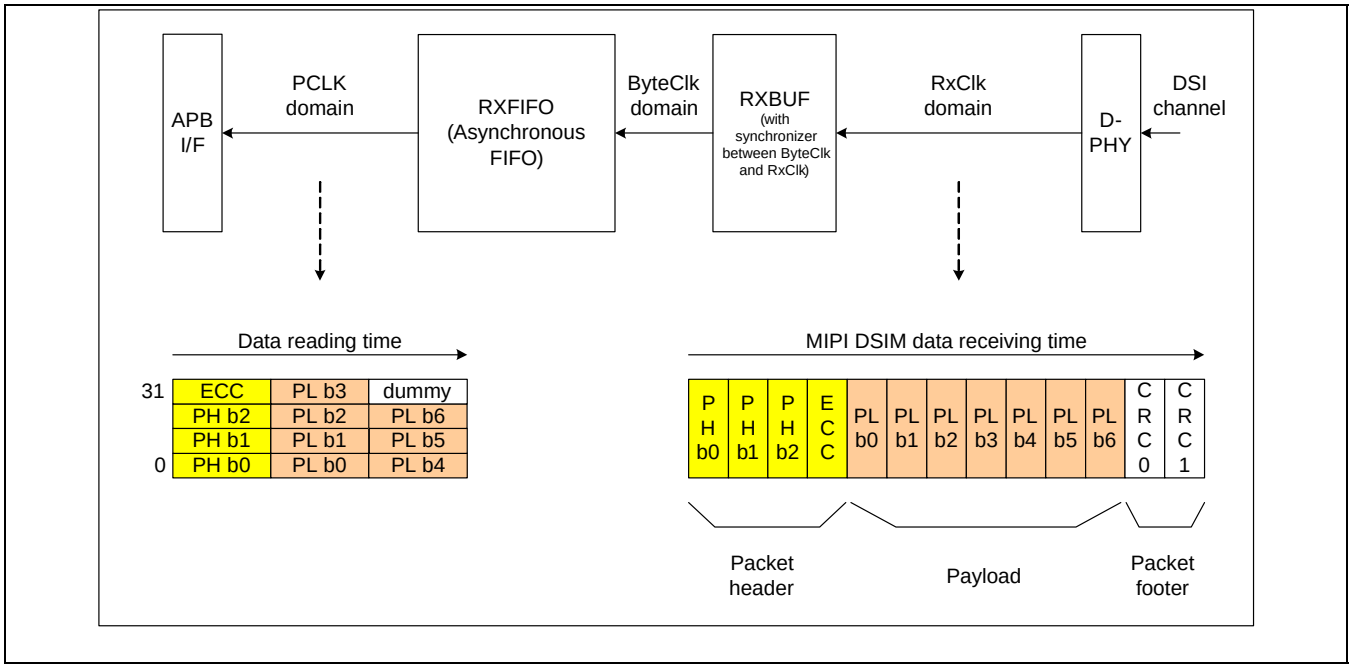


Figure 8.7-3 Rx Data Word Alignment

1.3 INTERFACES & PROTOCOL

1.3.1 Display Controller-to-DSI Conceptual Signal Converting Diagram in Video Mode

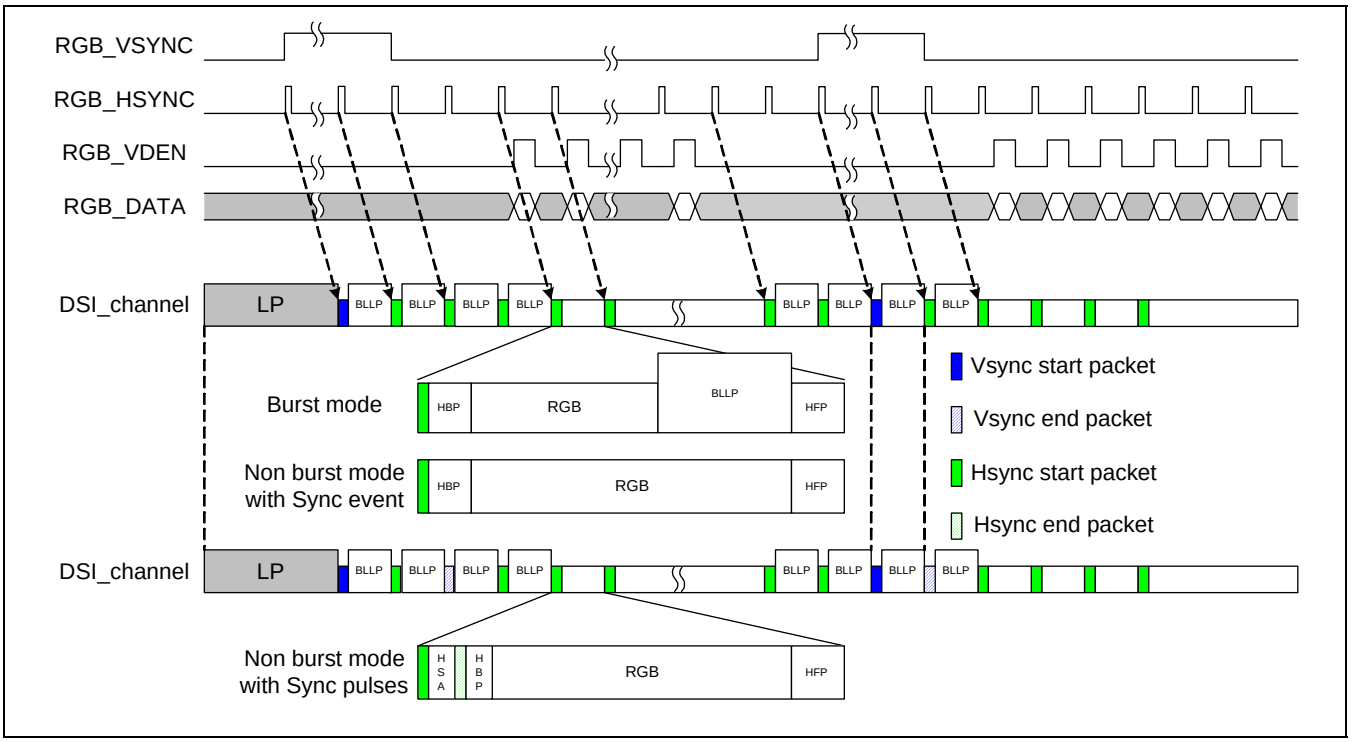


Figure 8.7-4 Signal Converting Diagram in Video Mode

1.3.2 Interface Timing and Protocol

1.3.2.1 Display Controller Interface

MIPI DSI Master has two-display controller interface: RGB I/F for main display and CPU I/F (I80 I/F) for main/sub display. Video mode uses RGB I/F,and Command mode uses CPU I/F.

The RGB image data is loaded on the data bus of RGB I/F and I80 I/F with the same order : RGB_VD[23:0] or SYS_VDOUT[23:0] is {R[7:0],G[7:0],B[7:0]}. Each byte aligns to the most significant bit. For instance, in 12-bit mode, only three 4-bit values, i.e. data[23:20], data[15:12], and data[7:4], are valid as R,G, and B each. The DSIM ignores rest of the bits.

1.3.2.2 RGB I/F

Vsync, Hsync and VDEN have to be active high signals.
Vsync and Hsync is pulse type that spends several video clocks.
RGB_VD[23:0] is {R[7:0],G[7:0],B[7:0]}.
All sync signals are synchronized to rising edge of RGB_VCLK.
Display controller has to send minimum 1 horizontal line length of Vsync pulse, V back porch, and V front porch.
Hsync pulse width should be longer than 1-byte clock cycle.

1.3.2.3 HSA Mode

HSA mode is Horizontal Sync Pulse areadisable mode.

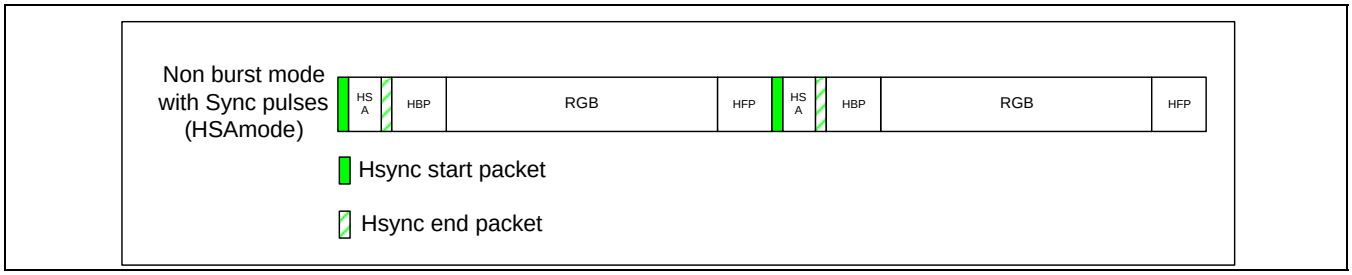


Figure 8.7-5 Block Timing Diagram of HSA Mode (HSA mode reset : DSIM_CONFIG[20] = 0)

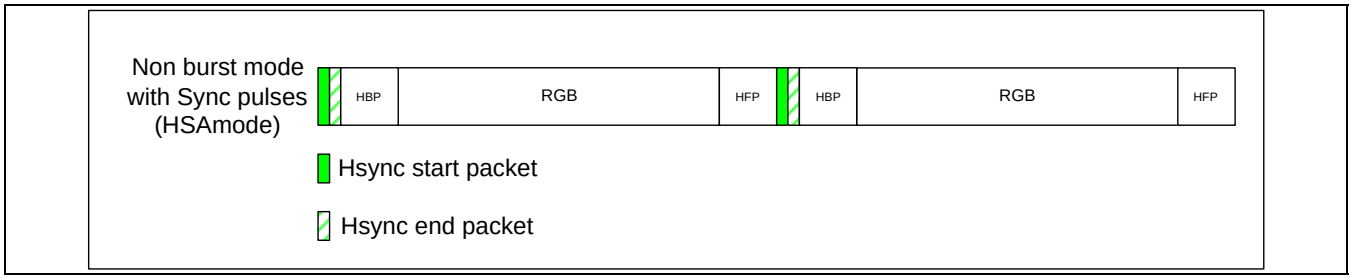


Figure 8.7-6 Block Timing Diagram of HSA mode (HSA mode set : DSIM_CONFIG[20] = 1)

1.3.2.4 HBP Mode

HBP mode is Horizontal Back Porch disable mode.

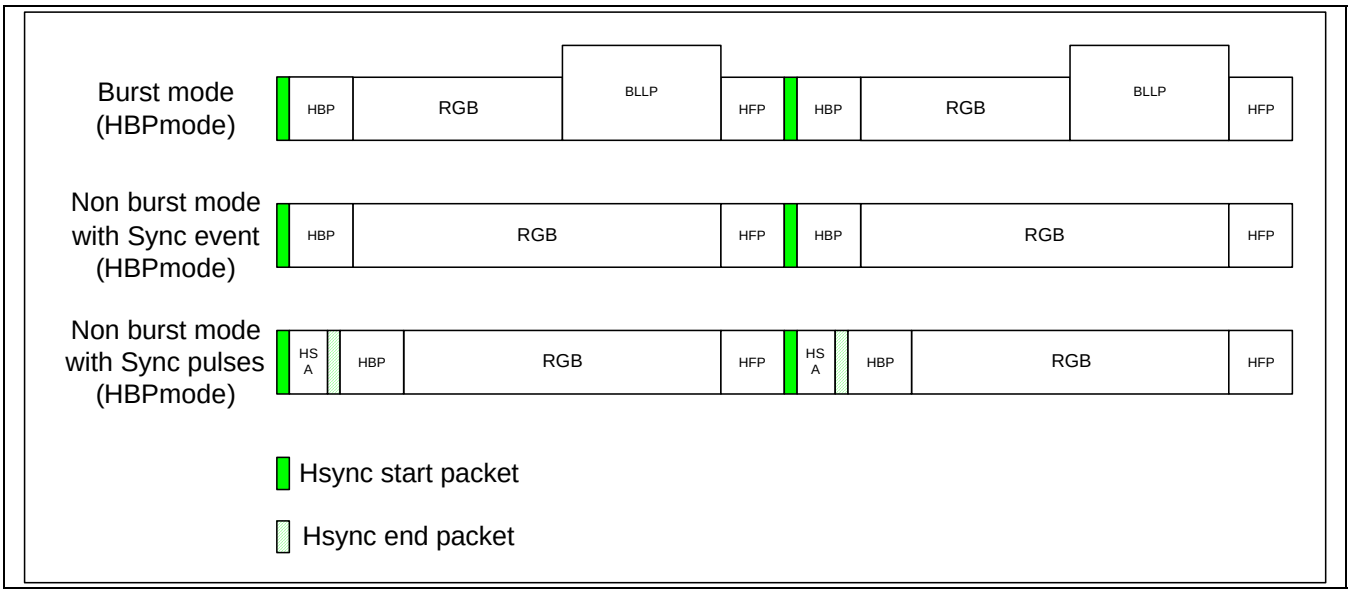


Figure 8.7-7 Block Timing Diagram of HBP mode (HBP Mode Reset : DSIM_CONFIG[21] = 0)

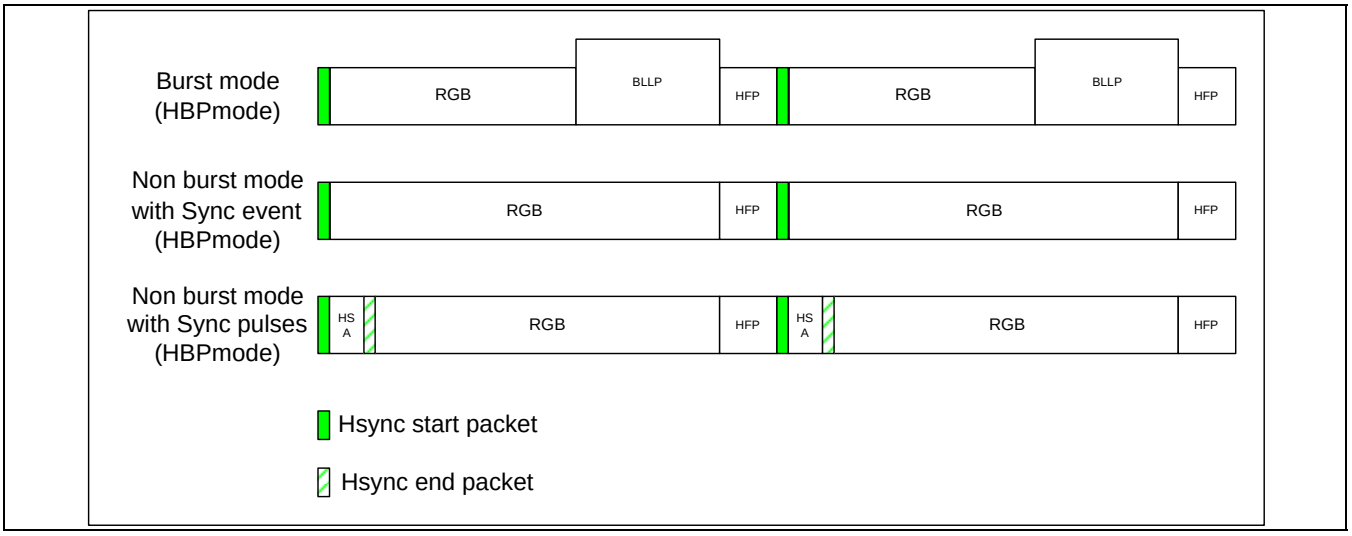


Figure 8.7-3 Block Timing Diagram of HBP mode (HBP Mode Set : DSIM_CONFIG[21] = 1)

1.3.2.5 HFP mode

HFP mode is Horizontal Front Porch disable mode.

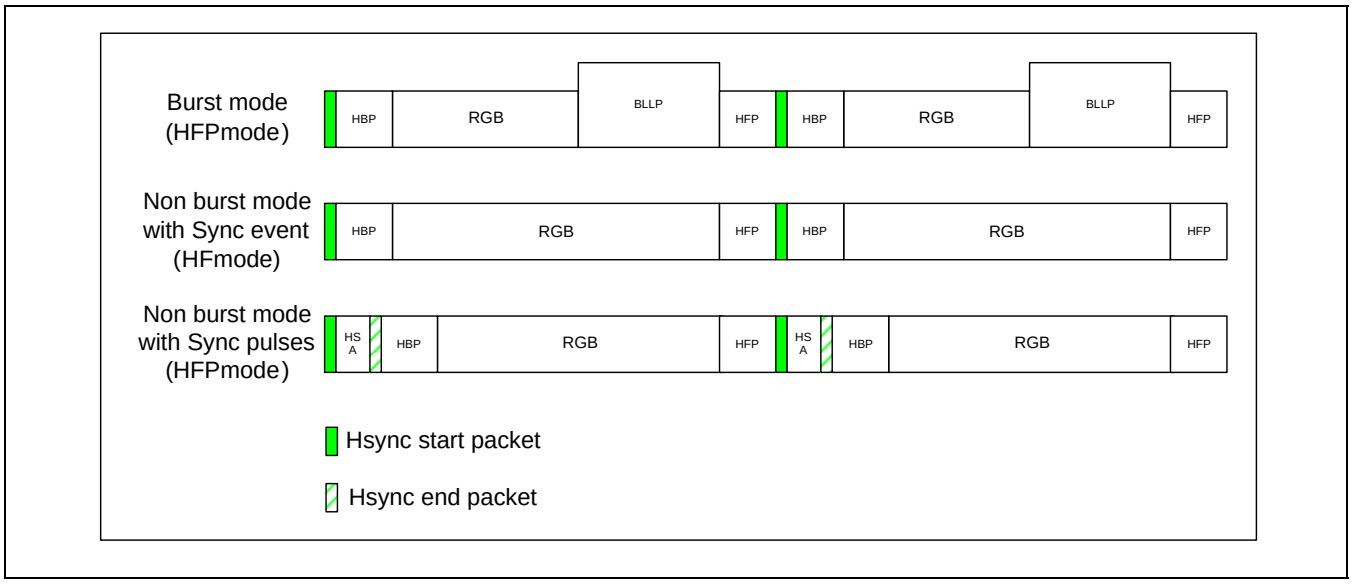


Figure 8.7-94 Block Timing Diagram of HFP mode (HFP Mode Reset : DSIM_CONFIG[22] = 0)

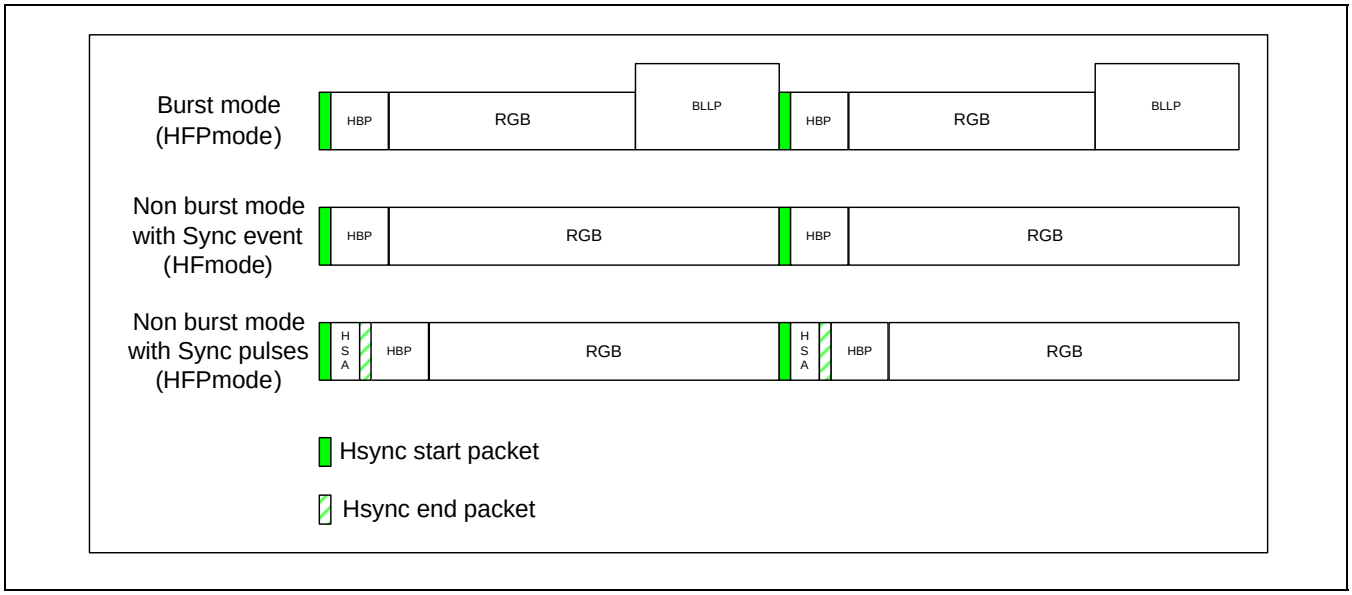


Figure 8.7-10 Block Timing Diagram of HFP mode (HFP Mode Set : DSIM_CONFIG[22] = 1)

1.3.2.6 HSE mode

HSE mode is Horizontal Sync End packet enable mode in Vsync pulse or Vporch area.

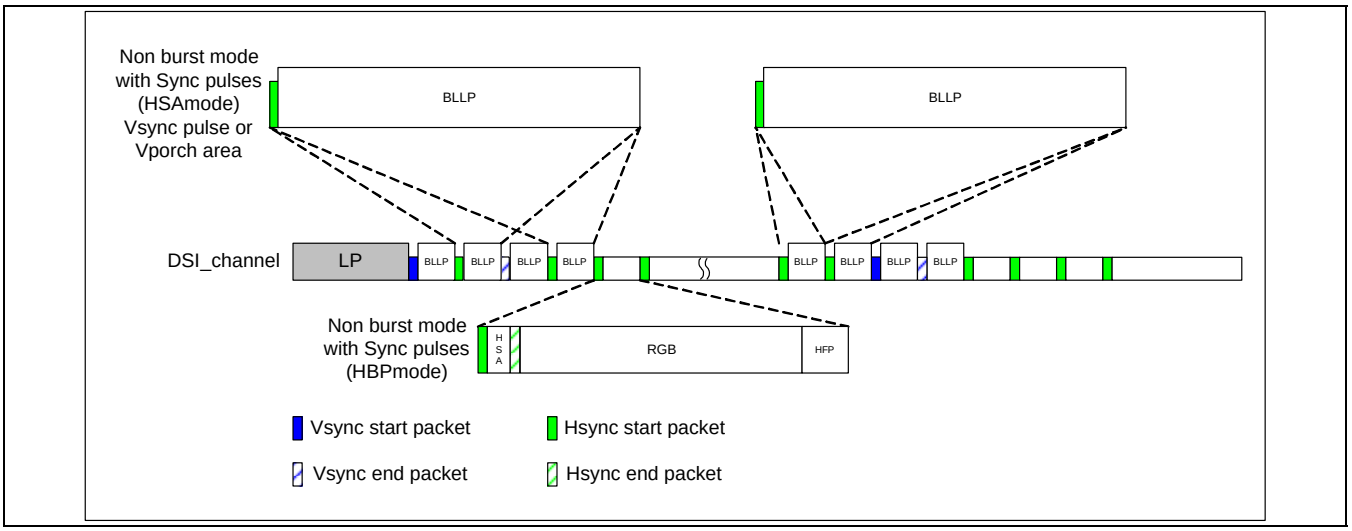


Figure 8.7-11 Block Timing Diagram of HSE Mode (HSE Mode Reset : DSIM_CONFIG[23] = 0)

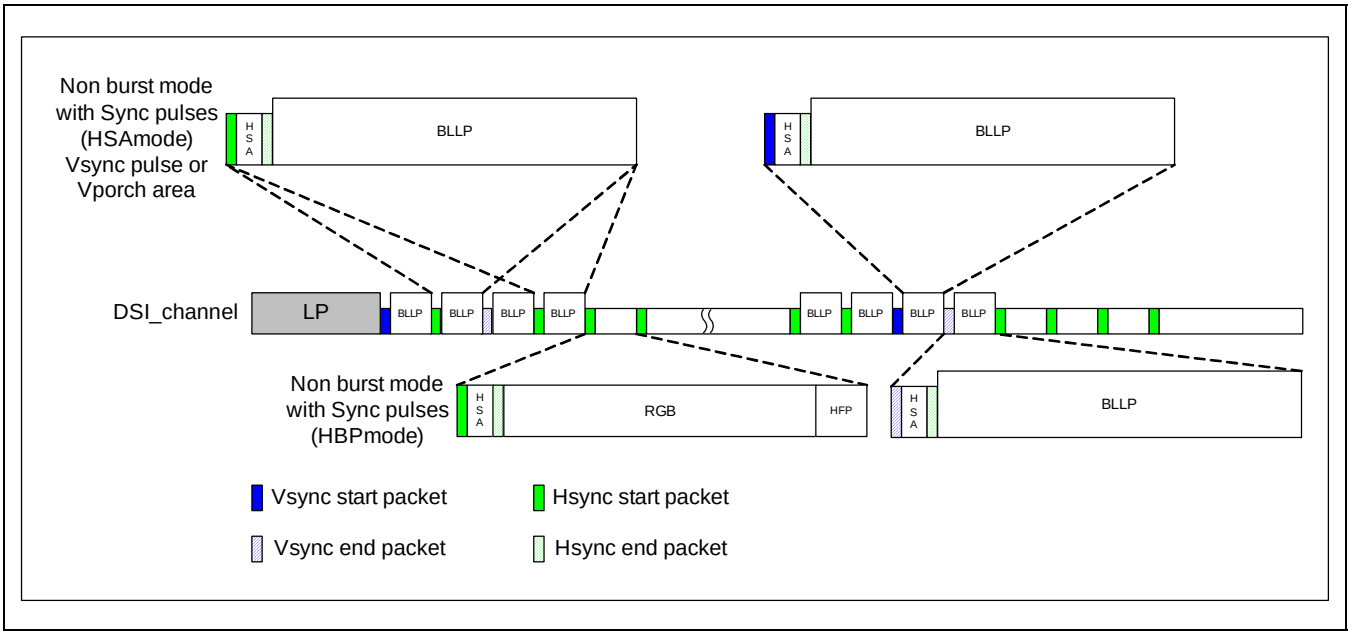


Figure 8.7-52 Block Timing Diagram of HSE Mode (HSE Mode Set : DSIM_CONFIG[23] = 1)

1.3.2.7 Transfer General Data in Video Mode

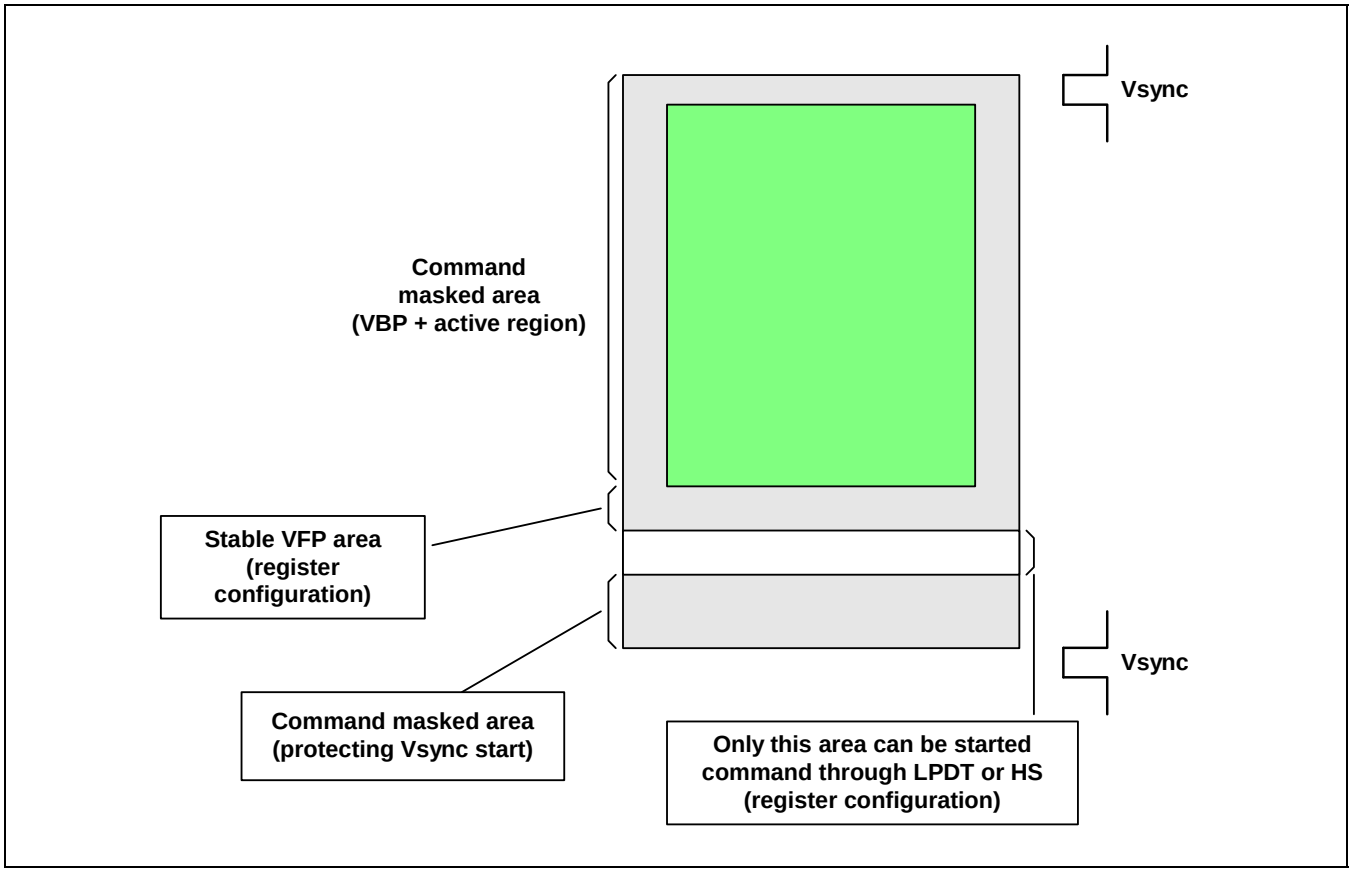


Figure 8.7-13 Stable VFP Area Before Command Transfer Allowing Area

MIPI DSIM Converts RGB I/F to Video mode

In Video mode, because Vsync and Hsync packet is very important for protecting image, MIPI DSIM allows several lines in VFP area for transferring general data transfer. In Figure 8.7-, Vertical front porch is separated into 3 areas: Stable VFP, Command allowed area, and command masked area.

Stable VFP is configurable by register. Configuration boundary is 11'h000 ~ 11'h7FFF in DSIM_MVPORCH.

Register configures the command allowed area. Configuration boundary is 4'h0 ~ 4'hF in DSIM_MVPORCH. Only this area is allowed command transferring start through HS mode or LPDT. In LPDT, data transferring spends long time about hundreds of us or more. In this time, Hsync packet does not come out because of LPDT long packet. MIPI DSIM has big size FIFO for this lost Hsync packet. After LPDT, MIPI DSIM transfers these Hsync packets at once through HS mode.

To protect Vsync, command masked area is masked area. This area is calculated by LPDT bandwidth. For example, If EscClk is 10MHz, Maximum long packet payload size is 1KB, and LPDT, LPDT transferring time is 824us (packet size : 1030byte, LPDT maximum bandwidth : 10Mbps). If 1 line time is 20us, line timing violation is occurred about 42 lines. Therefore, command masked area is larger than 42 + α. This 'α' is transferring time of violated Hsync packets

Display controller should be configured that VFP lines are sum of them – Stable VFP, Command allowed area, and command masked area.

1.3.2.8 I80 I/F

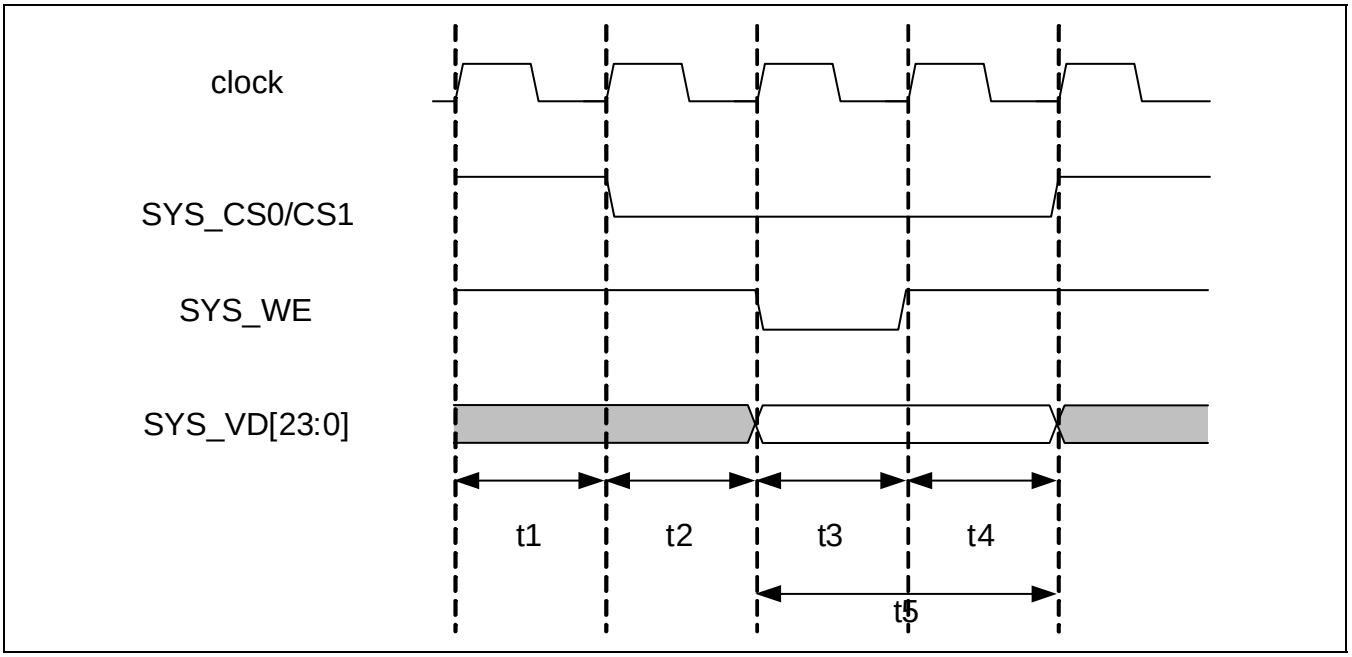


Figure 8.7-14 I80 I/F Timing Diagram

T1 ≥ 1 clock cycle.

T2 ≥ 0 clock cycle.

T3 ≥ 1 clock cycle.

T4 ≥ 1 clock cycle.

T5 ≥ 2 clock cycle.

T2+T3+T4 > 1 cycle of byte clock

A display controller generates these signals (SYS_CS0/CS1, SYS_WE, SYS_VD) with its internal clock. MIPI DSI master decodes the SYS_ADDR. Table 8.7-2 describes I80 I/F address map.

Table 8.7-2 I80 I/F Address Map

SYS_ADDR[1:0]	Description
2'b00	Image data
2'b01	Reserved
2'b10	Payload data
2'b11	Packet Header

Figure 8.7-15 shows how MIPI DSI Master packetizes the image data stream via i80 I/F in Command mode. MIPI DSI master packetizes the first line with DCS command "write_memory_start", and the other lines with DCS command "write_memory_continue".

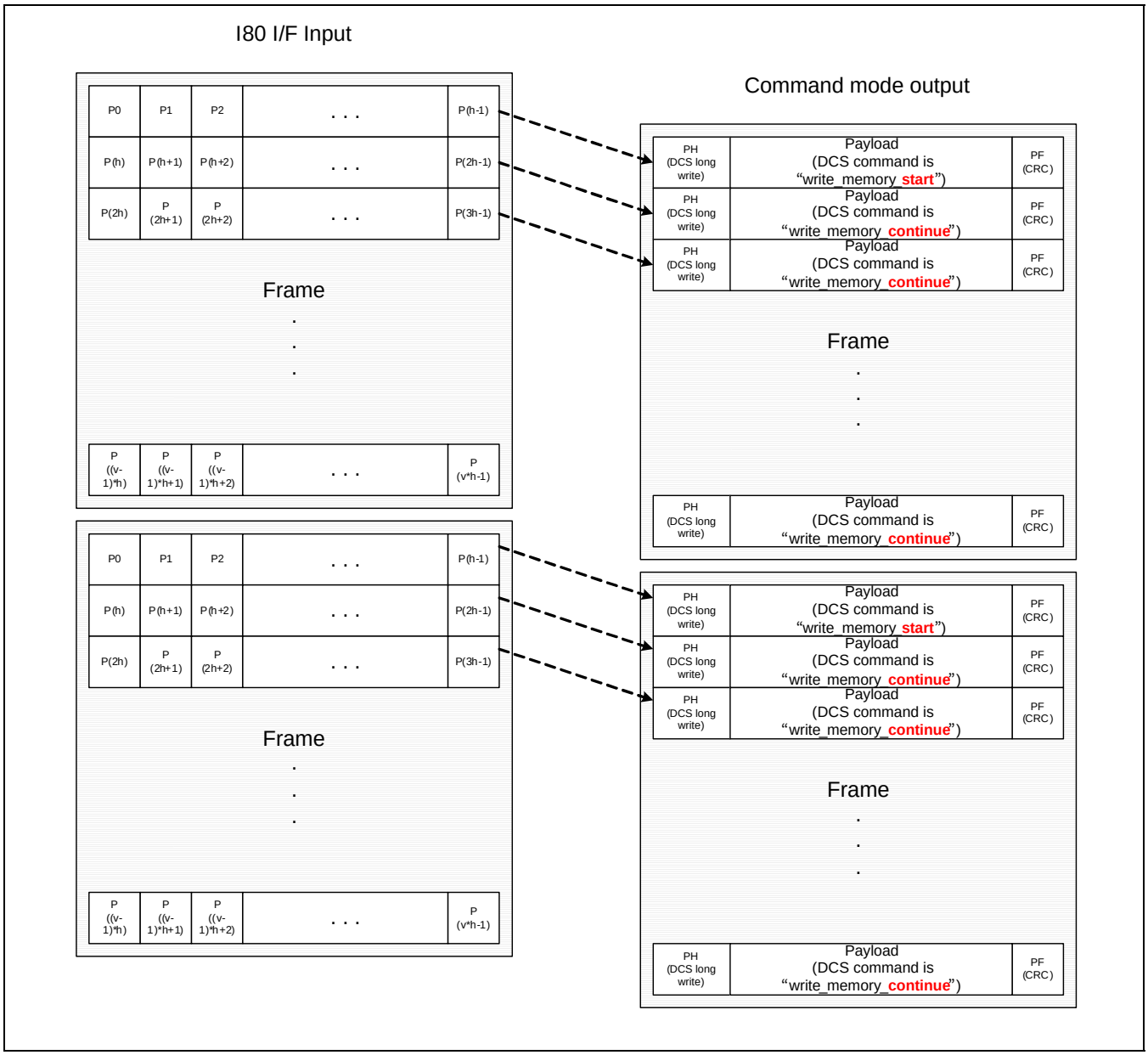


Figure 8.7-156 Packetizing for MIPI DSI Command Mode from I80 I/F

1.3.2.9 Relation between Input Transactions and DSI Transactions

Table 8.7-3 Relation between input transactions and DSI transactions

Input Interface	Input transaction	DSI Transaction
RGB	RGB transaction	RGB Packet 888,666,666(loosely packed), 565 should be specified via register configuration
I80	I80 Image Transaction	Data type is "DCS Long Write packet". (DCS command is "memory write start/continue".)
I80	I80 Command Transaction	Any DSI packets Bytes in I80 transaction should be the bytes in DSI packets.
SFR	Header and Payload FIFO access	Any DSI Packets Bytes in APB transaction should be the bytes in DSI packets.

1.4 CONFIGURATION

1.4.1 Video Mode vs. Command Mode

MIPI DSI Master Block supports two modes: Video mode and command mode.

1.5 DUAL DISPLAY VS. SINGLE DISPLAY

1.5.1 Dual Display

MIPI DSI Master supports dual display configuration only in command mode. That is, both main and sub display image should be transmitted via i80 interface.

1.5.2 Single Display

Use video mode or command mode for single display configuration.

1.6 PLL

MIPI DSI Master Block needs very high frequency (80MHz ~ 1GHz) clock generated by PLL to transmit Image data.
MIPI DSI Master has SFRs and corresponding interface signals to configure PLL. PLL is embedded in PHY module. Use other PLL in SoC as long as it meets the timing specification.

1.7 BUFFER

In MIPI DSI standard specification, DSI Master sends image stream in burst mode, where image stream transmits in higher speed along with high bit-clock frequency than its original speed of input stream. This mode allows the device to stay in stop state longer to reduce the power consumption. For this mode, MIPI DSI Master has a dual line buffer to store one complete line and send it faster at the next line time.

2 I/O DESCRIPTION

Table 8.7-4 MIPI-DPHY interface Slave Signal

Function Signal	I/O	Description	Pad	Type
MIPI_DP_0	B	DP signal for MIPI-DPHY Master data-lane 0	XmipiDP[0]	dedicated
MIPI_DN_0	B	DN signal for MIPI-DPHY Master data-lane 0	XmipiDN[0]	dedicated
MIPI_DP_1	B	DP signal for MIPI-DPHY Master data-lane 1	XmipiDP[1]	dedicated
MIPI_DN_1	B	DN signal for MIPI-DPHY Master data-lane 1	XmipiDN[1]	dedicated
MIPI_DP_2	B	DP signal for MIPI-DPHY Master data-lane 1	XmipiDP[2]	dedicated
MIPI_DN_2	B	DN signal for MIPI-DPHY Master data-lane 1	XmipiDN[2]	dedicated
MIPI_TXCP	B	DP signal for MIPI-DPHY Master clock-lane	XmipiTXCP	dedicated
MIPI_TXCN	B	DN signal for MIPI-DPHY Master clock-lane	XmipiTXCN	dedicated
MIPI _REG_CAP	B	Regulator capacitor connection. Connect a 2nF capacitor between this pin and GND	XmipiReg_CAP	dedicated

NOTES:
1. I/O direction. I: input, O: output, B: bi-direction
2. Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

3 REGISTER DESCRIPTION

3.1 REGISTER OVERVIEW

Register	Address	R/W	Description	Reset Value
DSIM_STATUS	0xECB0_0000	R	Status register	0x0010_010F
DSIM_SWRST	0xECB0_0x04	R/W	Software reset register	0x0000_0000
DSIM_CLKCTRL	0xECB0_0x08	R/W	Clock control register	0x0000_FFFF
DSIM_TIMEOUT	0xECB0_0x0C	R/W	Time Out register	0x00FF_FFFF
DSIM_CONFIG	0xECB0_0x10	R/W	Configuration register	0x0200_0000
DSIM_ESCMODE	0xECB0_0x14	R/W	Escape mode register	0x0000_0000
DSIM_MDRESOL	0xECB0_0x18	R/W	Main display Image resolution register	0x0300_0400
DSIM_MVPORCH	0xECB0_0x1C	R/W	Main display Vporch register	0xF000_0000
DSIM_MHPORCH	0xECB0_0x20	R/W	Main display Hporch register	0x0000_0000
DSIM_MSYNC	0xECB0_0x24	R/W	Main display Sync Area register	0x0000_0000
DSIM_SDRESOL	0xECB0_0x28	R/W	Sub display Image resolution register	0x0300_0400
DSIM_INTSRC	0xECB0_0x2C	R/W	Interrupt source register	0x0000_0000
DSIM_INTMSK	0xECB0_0x30	R/W	Interrupt mask register	0xB337_FFFF
DSIM_PKTHDR	0xECB0_0x34	W	Packet Header FIFO register	0x0000_0000
DSIM_PAYLOAD	0xECB0_0x38	W	Payload FIFO register	0x0000_0000
DSIM_RXFIFO	0xECB0_0x3C	R	Read FIFO register	0xXXXX_XXXX
DSIM_FIFOTHLD	0xECB0_0x40	R/W	FIFO threshold level register	0x0000_01FF
DSIM_FIFOCTRL	0xECB0_0x44	R	FIFO status and control register	0x0155_551F
DSIM_MEMACCHR	0xECB0_0x48	R/W	FIFO memory AC characteristic register	0x0000_4040
DSIM_PLLCTRL	0xECB0_0x4C	R/W	PLL control register	0x0000_0000
DSIM_PLLTMR	0xECB0_0x50	R/W	PLL timer register	0xFFFF_FFFF
DSIM_VERINFO	0xECB0_0x7C	R	DSIM IP version register	0x0000_0000

NOTE : M_RESETN at MIPI_PHY_CON0 (0xE020_0400) should be '1' before enabling DSIM.

3.2 DETAILED DECRIPTION

3.2.1 Status Register (DSIM_STATUS, R, Address = 0xECB0_0000)

This registers reads and checks status Internal & interface status.
FSM status, Line buffer status, current image line number, etc.

DSIM_STATUS	Bit	Description	R/W	Reset Value
PllStable	[31]	D-phy pll generates stable byteclk.	R	0
Reserved	[30:21]	Reserved	-	0
SwRstRls	[20]	Software reset status 0 = Reset state 1 = Release state	R	0
Reserved	[19:17]	Reserved	-	0
Direction	[16]	Data direction indicator 0 = Forward direction 1 = Backward direction	R	1
Reserved	[15:11]	Reserved	-	0
TxReadyHsClk	[10]	HS clock ready at Clock lane 0 = Not ready for transmitting HS data at clock lane. 1 = Ready for transmitting HS data at clock lane.	R	0
UlpsClk	[9]	ULPS indicator at clock lane 0 = No ULPS in clock lane 1 = ULSP in clock lane	R	1
StopstateClk	[8]	Stop state indicator at clock lane 0 = No Stop state in clock lane 1 = Stop state in clock lane	R	0
UlpsDat[3:0]	[7:4]	ULPS indicator at data lanes UlpsDat[0] : Data lane 0 UlpsDat[1] : Data lane 1 UlpsDat[2] : Data lane 2 0 = No ULPS in each data lane. 1 = ULPS in each data lane.	R	F
StopstateDat[3:0]	[3:0]	Stop state indicator at data lane StopstateDat[0] : Data lane 0 StopstateDat[1] : Data lane 1 StopstateDat[2] : Data lane 2 0 = No Stop state in each data lane 1 = Stop state in each data lane	R	0

3.2.2 Software Reset Register (DSIM_SWRST, R/W, Address = 0xECB0_0x04)

DSIM_SWRST	Bit	Description	R/W	Reset Value
Reserved	[31:17]	Reserved	-	-
FuncRst	[16]	Software reset (High active). "Software reset" reset all of FF in MIPI DSIM (except SFRs: STATUS, SWRST, CLKCTRL,TIMEOUT,CONFIG, ESCMODE*, MDRESOL, MDVPORCH, MHPORCH, MSYNC, INTMSK, SDRESOL, FIFOTHLD, FIFOCTRL**, MEMACCHR, PLLCTRL, PLLTMR, PHYACCHR,VERINFORM). 0 = Standby 1 = Reset * : ForceStopstate, CmdLpdt, TxLpdt, ** : nInitRx, nInitSfr, nInitI80, nInitSub, nInitMD	R/W	0
Reserved	[15:1]	Reserved	-	-
SwRst	[0]	Software reset (High active). "Software reset" reset all of FF in MIPI DSIM (except some SFRs: STATUS, SWRST, CLKCTRL, PLLCTRL, PLLTMR, and PHYTUNE). 0 = Standby 1 = Reset	R/W	0

3.2.3 Clock Control Register (DSIM_CLKCTRL, R/W, Address = 0xECB0_0x08)

DSIM_CLKCTRL	Bit	Description	R/W	Reset Value
TxRequestHsClk	[31]	HS clock request for HS transfer at clock lane (Turn on HS clock)	R/W	0
Reserved	[30:29]	Reserved	-	-
EscClkEn	[28]	Escape clock generating prescaler enabler 0 = Disables 1 = Enables	R/W	0
PLLBypass	[27]	Set PLLBypass signal connected to D-PHY module input to select for clock source bit. Refer to MIPI D-PHY specification. 0 = PLL output 1 = External Serial clock	R/W	0

DSIM_CLKCTRL	Bit	Description	R/W	Reset Value
ByteClkSrc	[26:25]	Byte clock source selection 00 = D-PHY PLL (default) 01 = External Bit clock source and ByteClk source by dividing by 4. 1X = External clock bypass to ByteClk If you want to change clock source, turn off D-PHY PLL before changing clock source. If bit[1] is 1'b1, this case is used that MIPI DSIM transfer data through LPDT mode only. In this case, it turns off PLL and cannot generate bit clock. If bit[1] is 1'b0 and bit[0] is 1'b1, External clock source is used bit clock and ByteClk source. Generates ByteClk by dividing 4 in MIPI DSIM clock generation module.	R/W	0
ByteClkEn	[24]	Byte clock enabler 0 = Disables 1 = Enables	R/W	0
LaneEscClkEn	[23:19]	Escape clock enabler for D-phy Lane LaneEscClkEn[0] = Clock lane LaneEscClkEn[1] = Data lane 0 LaneEscClkEn[2] = Data lane 1 LaneEscClkEn[3] = Data lane 2 0 = Disables 1 = Enables	R/W	0
Reserved	[18:16]	Reserved	-	-
EscPrescaler	[15:0]	Escape clock prescaler value. Escape clock frequency range is up to 20MHz. Note that the requirement for BTA is Host Escclk frequency is 66.7 ~ 150% of Peripheral escape clock frequency. EscClk = ByteClk / (2 * EscPrescaler)	R/W	0xFFFF

3.2.4 Time Out register (DSIM_TIMEOUT, R/W, Address = 0xECB0_0x0C)

DSIM_TIMEOUT	Bit	Description	R/W	Reset Value
Reserved	[31:24]	Reserved	-	-
BtaTout	[23:16]	Timer for BTA This register specifies time out from BTA request to change the direction with respect to Tx escape clock.	R/W	0xFF
LpdrTout	[15:0]	Timer for LP Rx mode timeout This register specifies time out on how long RxValid keeps deasserted after RxLpdt asserts with respect to Tx escape clock. RxValid is Rx data valid indicator. RxLpdt is an indicator that D-phy is under RxLpdt mode. RxValid and RxLpdt is signal from D-phy.	R/W	0xFFFF

3.2.5 Configuration register (DSIM_CONFIG , R/W, 0xECB0_0x10)

This register configures MIPI DSI master such as data lane number, input I/F, porch area, frame rate, BTA, LPDT, ULPS, etc.

DSIM_CONFIG	Bit	Description	R/W	Reset Value
Reserved	[31:29]	Reserved	-	-
TxTypeSfr	[28]	Packet Header data in SFR FIFO transmitting type 0 = If SFR packet header FIFO is not empty, 1 = If SFR Payload FIFO is filled over threshold level, Transmits Packet Header. Set threshold level via DSIM_FIFOTHLD register(Base_addr + 0x44)	R/W	0
SyncInform	[27]	Sync Pulse or Event mode select in Video mode 0 = Event mode (non burst, burst) 1 = Pulse mode (non burst only) In command mode, this bit is ignored.	R/W	0
BurstMode	[26]	Burst mode in Video mode In Non-burst mode, RGB data area is filled with RGB data and Null packets according to input bandwidth of RGB I/F. In Burst mode, RGB data area is filled with RGB data only. 0 = Non-burst mode 1 = Burst mode In command mode, this bit is ignored.	R/W	0
VideoMode	[25]	Display configuration 0 = Command mode 1 = Video mode	R/W	1

DSIM_CONFIG	Bit	Description	R/W	Reset Value
AutoMode	[24]	Auto vertical count mode In Video mode, Vertical line transition use line counter configured by VSA, VBP and Vertical resolution. If this bit is set to '1', line counter does not use VSA and VBP registers. 0 = Configuration mode 1 = Auto mode In command mode, this bit is ignored.	R/W	0
HseMode	[23]	In Vsync pulse and Vporch area, MIPI DSI master transfers only Hsync start packet to MIPI DSI slave at MIPI DSI spec 1.1r02. This bit is optional function to transfer Hsync end packet in Vsync pulse and Vporch area. 0 = Disables to transfer 1 = Enables to transfer In command mode, this bit is ignored.	R/W	0
HfpMode	[22]	HFP disable mode. If this bit set, DSI master ignores HFP area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.	R/W	0
HbpMode	[21]	HBP disable mode. If this bit set, DSI master ignores HBP area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.	R/W	0
HsaMode	[20]	HSA disable mode. If this bit set, DSI master ignores HSA area in Video mode. 0 = Enables 1 = Disables In command mode, this bit is ignored.	R/W	0
MainVc	[19:18]	Virtual channel number for main display	R/W	0
SubVc	[17:16]	Virtual channel number for sub display	R/W	0
Reserved	[15]	Reserved	-	-

DSIM_CONFIG	Bit	Description	R/W	Reset Value
MainPixFormat	[14:12]	Pixel stream format for main display 000 = 3bpp for Command mode only 001 = 8bpp for Command mode only 010 = 12bpp for Command mode only 011 = 16bpp for Command mode only 100 = 16-bit RGB (565) for Video mode only 101 = 18-bit RGB (666 : packed pixel stream) for Video mode only 110 = 18-bit RGB (666 : loosely packed pixel stream) for Common 111 = 24-bit RGB (888) for common	R/W	0
Reserved	[11]	Reserved	-	-
SubPixFormat	[10:8]	Pixel stream format for sub display 000 = 3bpp for Command mode only 001 = 8bpp for Command mode only 010 = 12bpp for Command mode only 011 = 16bpp for Command mode only 100 = 16-bit RGB (565) for Video mode only 101 = 18-bit RGB (666 : packed pixel stream) for Video mode only 110 = 18-bit RGB (666 : loosely packed pixel stream) for common 111 = 24-bit RGB (888) for Common	R/W	0
Reserved	[7]	Reserved	-	-
NumOfDatLane	[6:5]	Set the using data lane number 00 = Data lane 0 (1 data lane) 01 = Data lane 0 ~ 1 (2 data lanes) 10 = Data lane 0 ~ 2 (3 data lanes) 11 = Reserved	R/W	0
Reserved	[4]	Reserved	-	-
LaneEn[3:0]	[3:0]	Lane enabler. If Lane_EN is disabled, lane ignores input and drive initial value through output port. 0 = Lane is off. 1 = Lane is on. LaneEn[0] = clock lane enabler LaneEn[1] = data lane 0 enabler LaneEn[2] = data lane 1 enabler LaneEn[3] = data lane 2 enabler	R/W	0

3.2.6 Escape Mode Register (DSIM_ESCMODE , R/W, Address = 0xECB0_0x14)

This register configures MIPI DSI master.

DSIM_ESCMODE	Bit	Description	R/W	Reset Value
STOPstate_Cnt	[31:21]	After transmitting read packet or write "set_tear_on" command, BTA request to D-phy automatically. This counter value is the interval value between transmitting read packet (or write "set_tear_on" command) and BTA request. 11'h000 = 2 EscClk 11'h001 = 2 EscClk + 1 EscClk ~ 11'h3FF = 2 EscClk + 1023 EscClk	R/W	0
ForceStopstate	[20]	STOPstate forcing for D-PHY	R/W	0
Reserved	[19:17]	Reserved	-	-
ForceBta	[16]	Forcing Bus Turn Around Set '1', protocol layer request to D-PHY and MIPI DSI peripheral will be master after BTA sequence. This bit clears automatically at receiving BTA acknowledge from MIPI DSI peripheral.	R/W	0
Reserved	[15:8]	Reserved	-	-
CmdLpdt	[7]	LPDT transfers command in SFR FIFO. 0 = HS Mode 1 = LP Mode	R/W	0
TxLpdt	[6]	Data transmission in LP mode (all data transfer in LPDT) 0 = HS Mode 1 = LP Mode	R/W	0
Reserved	[5]	Reserved	-	-
TxTriggerRst	[4]	Remote reset trigger function. After Trigger operation, these bits will clear automatically	R/W	0
TxUlpsDat	[3]	ULPS request for data lane. Manually clear after ULPS Exit.	R/W	0
TxUlpsExit	[2]	ULPS Exit request for data lane. Manually clear after ULPS Exit.	R/W	0
TxUlpsClk	[1]	ULPS request for clock lane. Manually clear after ULPS Exit.	R/W	0
TxUlpsClkExit	[0]	ULPS Exit request for clock lane. Manually clear after ULPS Exit.	R/W	0

3.2.7 Main Display Image Resolution Register (DSIM_MDRESOL, R/W, Address = 0xECB0_0x18)

DSIM_MDRESOL	Bit	Description	R/W	Reset Value
MainStandby	[31]	Standby for receiving DISPCON output in Command mode after setting all configuration 0 = Not ready 1 = Stand by Standby should be set after configuration (resolution, reqtype, pixelform, etc.) is set for command mode. In Video mode, this bit is ignored.	R/W	0
Reserved	[30:27]	Reserved	-	-
MainVResol[10:0]	[26:16]	Vertical resolution (1 ~ 1024)	R/W	0x300
Reserved	[15:11]	Reserved	-	-
MainHResol[10:0]	[10:0]	Horizontal resolution (1 ~ 1024)	R/W	0x400

3.2.8 Main display Vporch register (DSIM_MVPORCH , R/W, Address = 0xECB0_0x1C)

DSIM_MVPORCH	Bit	Description	R/W	Reset Value
CmdAllow	[31:28]	Number of horizontal lines where command packet transmission is allowed after Stable VFP period. See Figure 8.7-.	R/W	0xF
Reserved	[27]	Reserved	-	-
StableVfp[10:0]	[26:16]	Number of horizontal lines where command packet transmission is not allowed after end of active region. See Figure 8.7-. * In Command mode, these bits are ignored.	R/W	0
Reserved	[15:11]	Reserved	-	-
MainVbp[10:0]	[10:0]	Vertical back porch width for Video mode (line count) In Command mode, these bits are ignored.	R/W	0

*Transfers command packets after Stable VFP area. Display controller VFP lines should be set based on sum of these values: Stable VFP, command allowing area and command masked area. See chapter 0 for transferring general data in Video mode.

3.2.9 Main display Hporch register (DSIM_MHPORCH, R/W, Address = 0xECB0_0x20)

DSIM_MHPORCH	Bit	Description	R/W	Reset Value
MainHfp[15:0]	[31:16]	Horizontal Front Porch width for Video mode HFP is specified using along blank packet. These bits are word counts for blank packet in HFP. In Command mode, these bits are ignored.	R/W	0
MainHbp[15:0]	[15:0]	Horizontal Back Porch width for Video mode HBP is specified using along blank packet. These bits are word counts for blank packet in HBP. In Command mode, these bits are ignored.	R/W	0

3.2.10 Main display Sync Area register (DSIM_MSYNC, R/W, Address = 0xECB0_0x24)

DSIM_MSYNC	Bit	Description	R/W	Reset Value
MainVsa[9:0]	[31:22]	Vertical Sync Pulse width for Video mode (line count) In command mode, these bits are ignored.	R/W	0
Reserved	[21:16]	Reserved	-	-
MianHsa[15:0]	[15:0]	Horizontal Sync Pulse width for Video mode HSA is specified using along blank packet. These bits are word counts for blank packet in HSA. In command mode, these bits are ignored.	R/W	0

3.2.11 Sub display Image resolution register (DSIM_SDRESOL, R/W, Address = 0xECB0_0x28)

DSIM_SDRESOL	Bit	Description	R/W	Reset Value
SubStandby	[31]	Standby for receiving DISPCON output in Command mode after setting all configuration 0 = Not ready 1 = Standby Standby should be set after configuration(resolution, reqtype, pixelform, and etc.) is set for command mode. In Video mode, this bit is ignored.	R/W	0
Reserved	[30:27]	Reserved	-	-
SubVResol[10:0]	[26:16]	Vertical resolution (1 ~ 1024)	R/W	0x300
Reserved	[15:11]	Reserved	-	-
SubHResol[10:0]	[10:0]	Horizontal resolution (1 ~ 1024)	R/W	0x400

3.2.12 Interrupt Source Register (DSIM_INTSRC, R/W, Address = 0xECB0_0x2C)

This register identifies interrupt sources.
Internal block error, data transmit interrupt, inter-layer(D_PHY) error, etc.
The bits are set even if they are masked off by DSIM_INTMSK_REG.
Write '1' to clear the Interrupt.

DSIM_INTSRC	Bit	Description	R/W	Reset Value
PllStable	[31]	Indicates that D-phy PLL is stable	R/W	0
SwRstRelease	[30]	Software reset is released	R/W	0
SFRFifoEmpty	[29]	SFR payload FIFO empty	R/W	0
SyncOverride	[28]	Indicates that other DSI command transfer has overridden Sync timing.	R/W	0
Reserved	[27:26]	Reserved	-	-
BusTurnOver	[25]	Indicates when Bus grant turns over from DSI slave to DSI master.	R/W	0
FrameDone	[24]	Indicates when MIPI DSIM transfers whole image frame ***NOTE:** When Hsync is not received during 2 line times, internal timer is time out and this bit is flagged.	R/W	0
Reserved	[23:22]	Reserved	-	-
LpdrTout	[21]	LP Rx timeout. See the time out register (0x10)	R/W	0
TaTout	[20]	Turn around Acknowledge Timeout. See the time out register (0x10)	R/W	0
Reserved	[19]	Reserved	-	-
RxDatDone	[18]	Data receiving complete	R/W	0
RxTE	[17]	TE Rx trigger received	R/W	0
RxAck	[16]	Ack Rx trigger received	R/W	0
ErrRxECC	[15]	ECC multi bit error in LPDR	R/W	0
ErrRxCRC	[14]	CRC error in LPDR	R/W	0
Reserved	[13]	Reserved	-	-
ErrEsc2	[12]	Escape mode entry error lane 2 (For more information refer to standard d-phy specification)	R/W	0
ErrEsc1	[11]	Escape mode entry error lane 1 (For more information refer to standard d-phy specification)	R/W	0
ErrEsc0	[10]	Escape mode entry error lane 0 (For more information refer to standard d-phy specification)	R/W	0
Reserved	[9]	Reserved	-	-
ErrSync2	[8]	LPDT Sync Error lane2 (For more information refer to standard d-phy specification)	R/W	0

DSIM_INTSRC	Bit	Description	R/W	Reset Value
ErrSync1	[7]	LPDT Sync Error lane1 (For more information refer to standard d-phy specification)	R/W	0
ErrSync0	[6]	LPDT Sync Error lane0 (For more information refer to standard d-phy specification)	R/W	0
Reserved	[5]	Reserved	-	-
ErrControl2	[4]	Control Error lane2 (For more information refer to standard d-phy specification)	R/W	0
ErrControl1	[3]	Control Error lane1 (For more information refer to standard d-phy specification)	R/W	0
ErrControl0	[2]	Control Error lane0 (For more information refer to standard d-phy specification)	R/W	0
ErrContentLP0	[1]	LP0 Contention Error (only lane0, because BTA is only occurred at lane0) (For more information refer to standard d-phy specification)	R/W	0
ErrContentLP1	[0]	LP1 Contention Error (only lane0, because BTA is only occurred at lane0) (For more information refer to standard d-phy specification)	R/W	0

3.2.13 Interrupt mask register (DSIM_INTMSK, R/W, Address = 0xECB0_0x30)

This register masks interrupt sources.

DSIM_INTMSK	Bit	Description	R/W	Reset Value
MskPllStable	[31]	Indicates that D-phy PLL is stable	-	-
MskSwRstRelease	[30]	Software reset is released	R/W	0
MskSFRFifoEmpty	[29]	SFR payload FIFO empty	R/W	1
MskSyncOverride	[28]	Indicates that a Sync timing has been overridden by other DSI command transfer.	R/W	1
Reserved	[27:26]	Reserved	-	-
MskBusTurnOver	[25]	Indicates when Bus grant turns over from DSI slave to DSI master.	R/W	1
MskFrameDone	[24]	Indicates when MIPI DSIM transfers whole image frame	R/W	1
Reserved	[23:22]	Reserved	-	-
MskLpdrTout	[21]	LP Rx timeout. See the time out register (0x10)	R/W	1
MskTaTout	[20]	Turn around Acknowledge Timeout. See the time out register (0x10)	R/W	1
Reserved	[19]	Reserved	-	-
MskRxDatDone	[18]	Data receiving complete	R/W	1
MskRxTE	[17]	TE Rx trigger received	R/W	1
MskRxAck	[16]	ACK Rx trigger received	R/W	1
MskRxECC	[15]	ECC multi bit error in LPDR	R/W	1
MskRxCRC	[14]	CRC error in LPDR	R/W	1
Reserved	[13]	Reserved	-	-
MskEsc2	[12]	Escape mode entry error lane2 (For more information refer to standard d-phy specification)	R/W	1
MskEsc1	[11]	Escape mode entry error lane1 (For more information refer to standard d-phy specification)	R/W	1
MskEsc0	[10]	Escape mode entry error lane0 (For more information refer to standard d-phy specification)	R/W	1
Reserved	[9]	Reserved	-	-
MskSync2	[8]	LPDT Sync Error lane2 (For more information refer to standard d-phy specification)	R/W	1
MskSync1	[7]	LPDT Sync Error lane1 (For more information refer to standard d-phy specification)	R/W	1
MskSync0	[6]	LPDT Sync Error lane0	R/W	1

DSIM_INTMSK	Bit	Description	R/W	Reset Value
		(For more information refer to standard d-phy specification)		
Reserved	[5]	Reserved	-	-
MskControl2	[4]	Control Error lane2 (For more information refer to standard d-phy specification)	R/W	1
MskControl1	[3]	Control Error lane1 (For more information refer to standard d-phy specification)	R/W	1
MskControl0	[2]	Control Error lane0 (For more information refer to standard d-phy specification)	R/W	1
MskContentLP0	[1]	LP0 Contention Error (For more information refer to standard d-phy specification)	R/W	1
MskContentLP1	[0]	LP1 Contention Error (For more information refer to standard d-phy specification)	R/W	1

3.2.14 Packet Header FIFO register (DSIM_PKTHDR, W, Address = 0xECB0_0x34)

This register is the FIFO for packet header to send DSI packets.

DSIM_PKTHDR	Bit	Description	R/W	Reset Value
Reserved	[31:24]	Reserved	-	-
PacketHeader	[23:0]	This register is to write the packet header of Tx packet. [7:0] = DI [15:8] = Dat0 (Word Count lower byte for long packet) [23:16] = Dat1 (Word Count upper byte for long packet)	W	0

3.2.15 Payload FIFO register (DSIM_PAYLOAD, W, Address = 0xECB0_0x38)

This register is the FIFO for payload to send DSI packets.

DSIM_PAYLOAD	Bit	Description	R/W	Reset Value
Payload	[31:0]	This register is to write the Payload of Tx packet.	W	0

3.2.16 Read FIFO register (DSIM_RXFIFO, R, Address = 0xECB0_0x3C)

This register is the gate of FIFO read

DSIM_RXFIFO	Bit	Description	R/W	Reset Value
RxDat	[31:0]	In Rx mode, user can read Rx data through this register. Note that the CRC in the packet is not stored in RxFIFO.	R	Unknown

3.2.17 FIFO threshold level register (DSIM_FIFOTHLD, R/W, Address = 0xECB0_0x40 : hidden)

DSIM_FIFOTHLD	Bit	Description	R/W	Reset Value
Reserved	[31:9]	Reserved	-	-
WFullLevelSfr	[8:0]	Almost full level of SFR payload FIFO	R/W	0x1FF

3.2.18 FIFO Status & Control Register (DSIM_FIFOCTRL, R, Address = 0xECB0_0x44 : hidden)

DSIM_FIFOCTRL	Bit	Description	R/W	Reset Value
Reserved	[31:26]	Reserved	-	-
FullRx	[25]	Rx FIFO full	R	0
EmptyRx	[24]	Rx FIFO empty	R	1
FullHSfr	[23]	SFR packet header FIFO full	R	0
EmptyHSfr	[22]	SFR packet header FIFO empty	R	1
FullLSfr	[21]	SFR payload FIFO full	R	0
EmptyLSfr	[20]	SFR payload FIFO empty	R	1
FullHI80	[19]	I80 packet header FIFO full	R	0
EmptyHI80	[18]	I80 packet header FIFO empty	R	1
FullLI80	[17]	I80 payload FIFO full	R	0
EmptyLI80	[16]	I80 payload FIFO empty	R	1
FullHSub	[15]	Sub display packet header FIFO full	R	0
EmptyHSub	[14]	Sub display packet header FIFO empty	R	1
FullLSub	[13]	Sub display payload FIFO full	R	0
EmptyLSub	[12]	Sub display payload FIFO empty	R	1
FullHMain	[11]	Main display packet header FIFO full	R	0
EmptyHMain	[10]	Main display packet header FIFO empty	R	1
FullLMain	[9]	Main display payload FIFO full	R	0
EmptyLMain	[8]	Main display payload FIFO empty	R	1
Reserved	[7:5]	Reserved	-	-
nInitRx	[4]	MD FIFO read point initialize	R/W	1
nInitSfr	[3]	SFR FIFO write point initialize	R/W	1
nInitI80	[2]	I80 FIFO write point initialize	R/W	1
nInitSub	[1]	SD FIFO write point initialize	R/W	1
nInitMain	[0]	MD FIFO write point initialize	R/W	1

3.2.19 FIFO Memory AC Characteristic Register (DSIM_MEMACCHR, R/W, Address = 0xECB0_0x48 : hidden)

DSIM_MEMACCHR	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	-	-
PGEN_SD	[15]	Sub display FIFO memory power gating	R/W	0
RETN_SD	[14]	Sub display FIFO memory Retention	R/W	1
EMAB_SD	[13:11]	Sub display FIFO memory B port margin adjustment	R/W	0
EMAA_SD	[10:8]	Sub display FIFO memory A port margin adjustment	R/W	0
PGEN_MD	[7]	Main display FIFO memory power gating	R/W	0
RETN_MD	[6]	Main display FIFO memory Retention	R/W	1
EMAB_MD	[5:3]	Main display FIFO memory B port margin adjustment	R/W	0
EMAA_MD	[2:0]	Main display FIFO memory A port margin adjustment	R/W	0

3.2.20 PLL Control Register (DSIM_PLLCTRL, R/W, Address = 0xECB0_0x4C)

This register configures PLL control, D-PHY, clock range indication, etc.

DSIM_PLLCTRL	Bit	Description	R/W	Reset Value
Reserved	[31:28]	Should be 0	-	-
FreqBand[3:0]	[27:24]	Bitclk frequency band indicator for D-PHY global timing	R/W	0
PllEn	[23]	Enables PLL	R/W	0
Reserved	[22:20]	Should be 0	-	-
PMS[19:1]	[19:1]	PLL PMS value	R/W	0
DpDnSwap	[0]	Set DpDnSwap signal connected to D-PHY module input signal to swap Dp and Dn channel of D-phy.	R/W	0

3.2.21 PLL Timer Register (DSIM_PLLTMR, R/W, Address = 0xECB0_0x50)

DSIM_PLLTMR	Bit	Description	R/W	Reset Value
PllTimer	[31:0]	PLL Timer for stability of the generated clock (System clock cycle base) If timer value goes to 0x00000000, clock stable bit of status and interrupt register is set.	R/W	0xFFFFFFFF

3.2.22 DSIM IP Version Register (DSIM_VERINFO, R, Address = 0xECB0_0x7C)

DSIM_VERINFO	Bit	Description	R/W	Reset Value
VerInform	[31:0]	MIPI DSIM IP version information	R	0

4 DPHY PLL CONTROL

4.1 PMS SETTING SAMPLE FOR MIPI PLL

4.1.1 PMS setting

Write a value to PMS field in DSIM_PLLCTRL (0xECB0_004C) to set PMS value for DPHY PLL. Each P, M and S resides in PMS [19:14] , PMS[13:4] and PMS[3:1] . Before Setting, PMS value please read the following:

1. Do not set the P or M value as zero, that is, setting the P(00000), M(0000000000) causes malfunction of the PLL.
2. Select M and P considering stability and VCO range
3. Keep range of Fin_pll from 2.5 MHz to 5 MHz, by P code setting

Table 8.7-5 PMS and frequency constraint

	Function	Value	Description
Fin	Fin	3~200MHz	Pll input frequency
Fin_pll	Fin_pll	2.5 ~ 5 MHz	Pfd input frequency
VCO_out	(M*Fin)/P	500 ~ 1000 MHz	VCO output frequency
Fout	(M*Fin)/(P*2^S)	16 ~ 1000 MHz	PLL output frequency
P[5:0]	P	1 ~ 63	PMS[19:14]
M[9:0]	M	100 ~ 1023	PMS[13:4]
S[2:0]	2^S	1, 2, 4, 8, 16, 32	PMS[3:1]

4.1.2 Sample for Fout 80 MHz

Followings are some example to set PMS value for Fout 80 MHz

	Case 1	Case 2
Fin	24MHz	4MHz
Fin_pll	4MHz	4MHz
P[5:0]	6	1
M[9:0]	160	160
S[2:0]	8	8
VCO_out	640MHz	640MHz
Fout	80MHz	80MHz

4.1.3 Sample for Fout 1000 MHz

Followings are some example to set PMS value for Fout 1000 MHz

	Case 1	Case 2
Fin	4MHz	80MHz
Fin_pll	4MHz	4MHz
P[5:0]	1	2
M[9:0]	250	250
S[2:0]	1	1
VCO_out	1000MHz	1000MHz
Fout	1000MHz	1000MHz

4.1.4 Sample for Fout 999 MHz

Followings are some example to set PMS value for Fout 999 MHz

	Case 1	Case 2
Fin	3MHz	81MHz
Fin_pll	3MHz	3MHz
P[5:0]	1	27
M[9:0]	333	333
S[2:0]	1	1
VCO_out	999MHz	999MHz
Fout	999MHz	999MHz

8.8 MIPI CSIS

1 OVERVIEW

MIPI CSIS overall features are summarized as follows.

2 FEATURE

- Compliant to MIPI CSI2 Standard Specification Version 1.0
  - ♦ Supports 1 or 2 data lanes
  - ♦ Supports 1 channels
  - ♦ Supports RAW8, RAW10, RAW12, Embedded byte-based long packet and YUV422 8-bit
  - ♦ "User defined Byte-based Data" is not supported
- Interfaces
  - ♦ Compatible to PPI (Protocol-to-PHY Interface) in MIPI D-PHY Specification Version 0.86
  - ♦ Supports AHB Slave I/F
  - ♦ Supports ISP Output
  - ♦ Supports Memory I/F (DPSRAM_2048X32) for embedded packet or generic short packet

3 BLOCK DIAGRAM

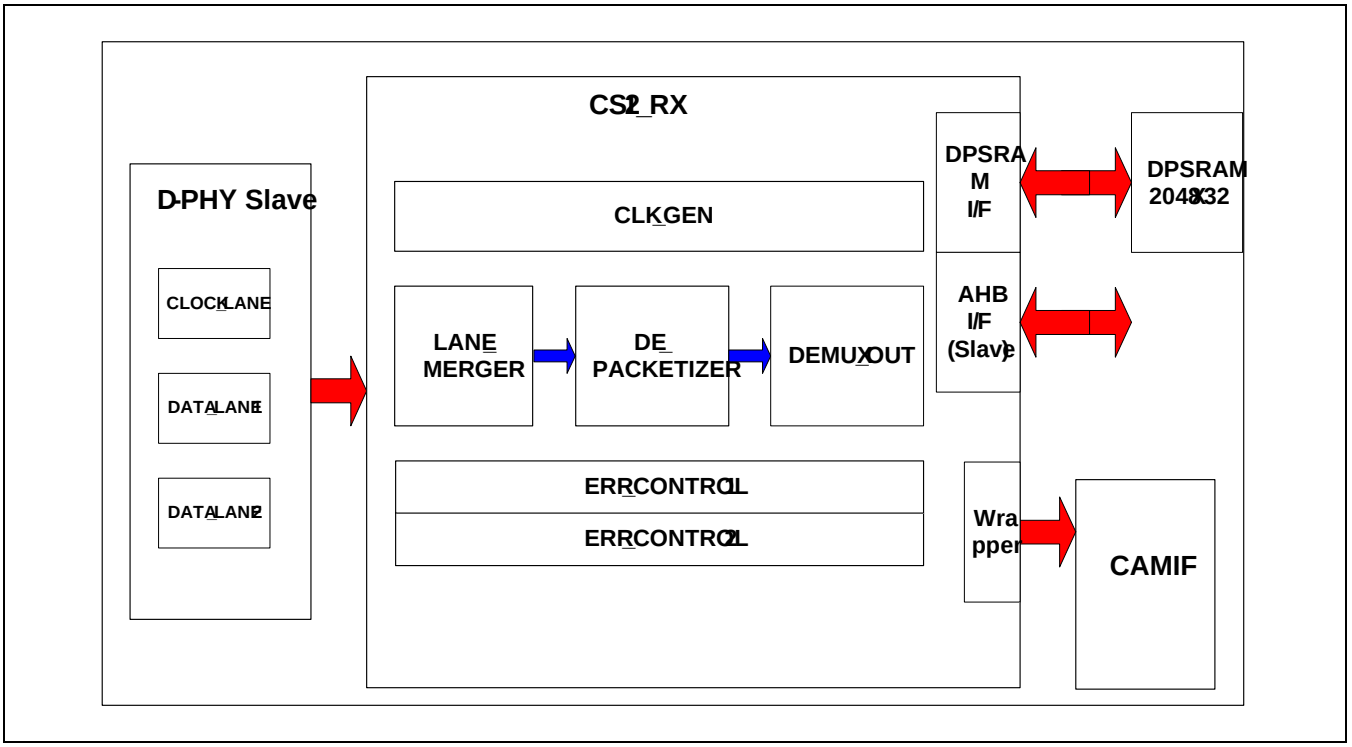


Figure 8.8-1 MIPI CSI System Block Diagram

4 INTERFACE & PROTOCOL

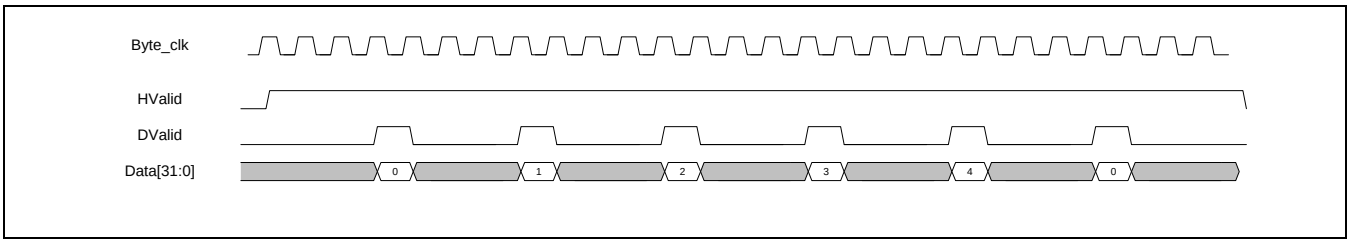


Figure 8.8-2 Waveform of Output Data

5 DATA FORMAT

Table 8.8-1 Data Size Constraint

FORMAT	Pixels	Bytes	Bits
YUV422 8-bit	2	4	32
RAW8	1	1	8
RAW10	4	5	40
RAW12	2	3	24

5.1 YUV422 8-BIT

YUV422 8-bit format data is stored as a UYVY sequence. This sequence is illustrated in Figure 8.8-3. Table 8.8-1 specifies the data size constraints for YUV422 8-bit format.

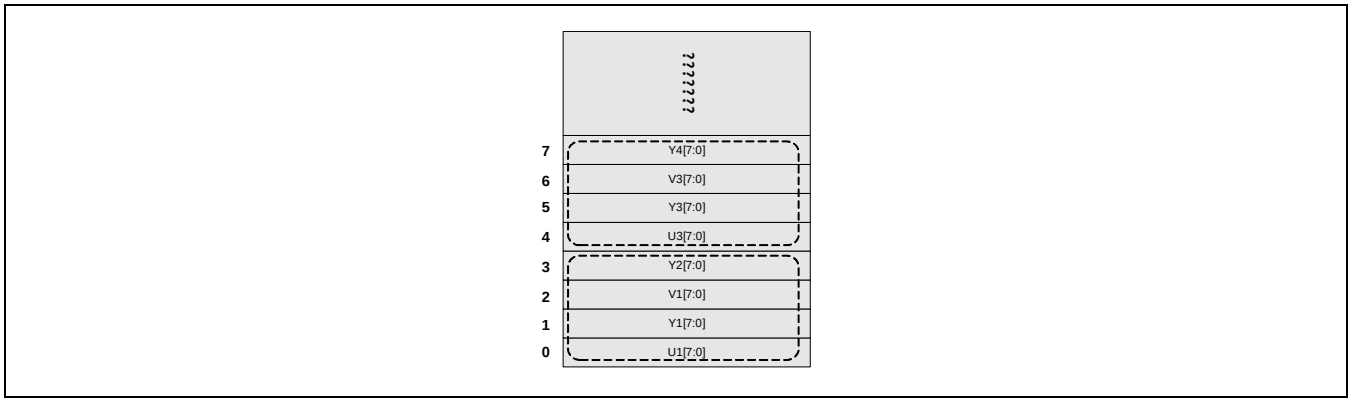


Figure 8.8-3 YUV422 Data Storing Order

5.2 RAW8

8-bit RAW data is stored as the sequence illustrated in Figure 8.8-3, Table 8.8-1 specifies the data size constraints for RAW8 format.

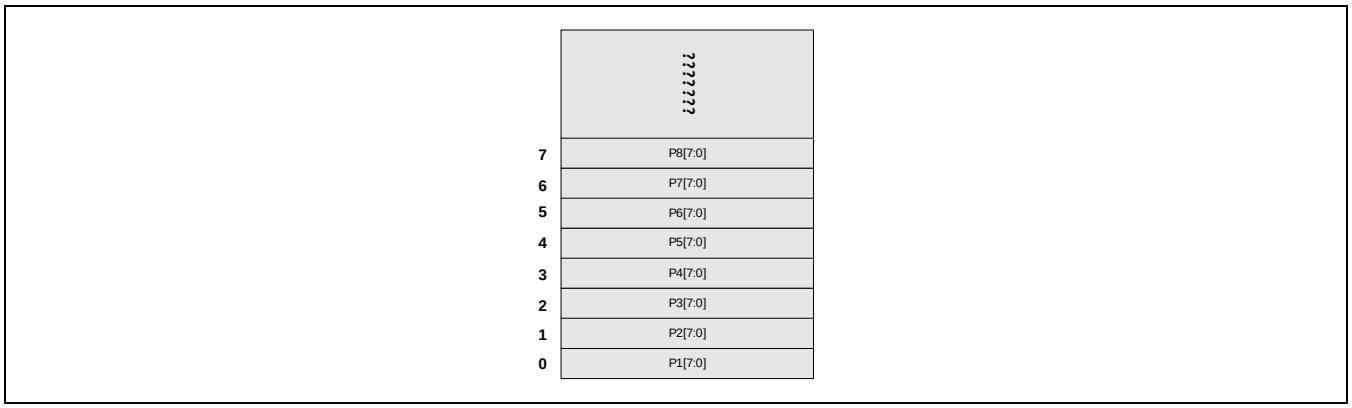


Figure 8.8-4 RAW8 Data Storing Order

5.3 RAW10

10-bit RAW data is stored as the sequence illustrated in Figure 8.8-4, Table 8.8-1 specifies the data size constraints for RAW10 format.

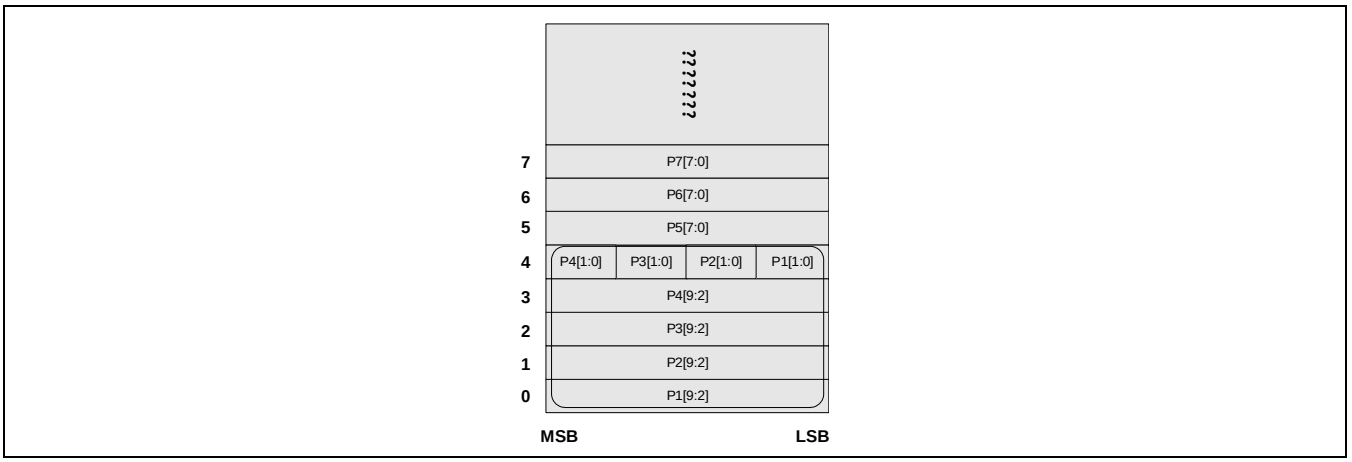


Figure 8.8-5 RAW10 Data Storing Order

5.4 RAW12

12-bit RAW data is stored as the sequence illustrated in Figure 8.8-5, Table 8.8-1 specifies the data size constraints for RAW12 format.

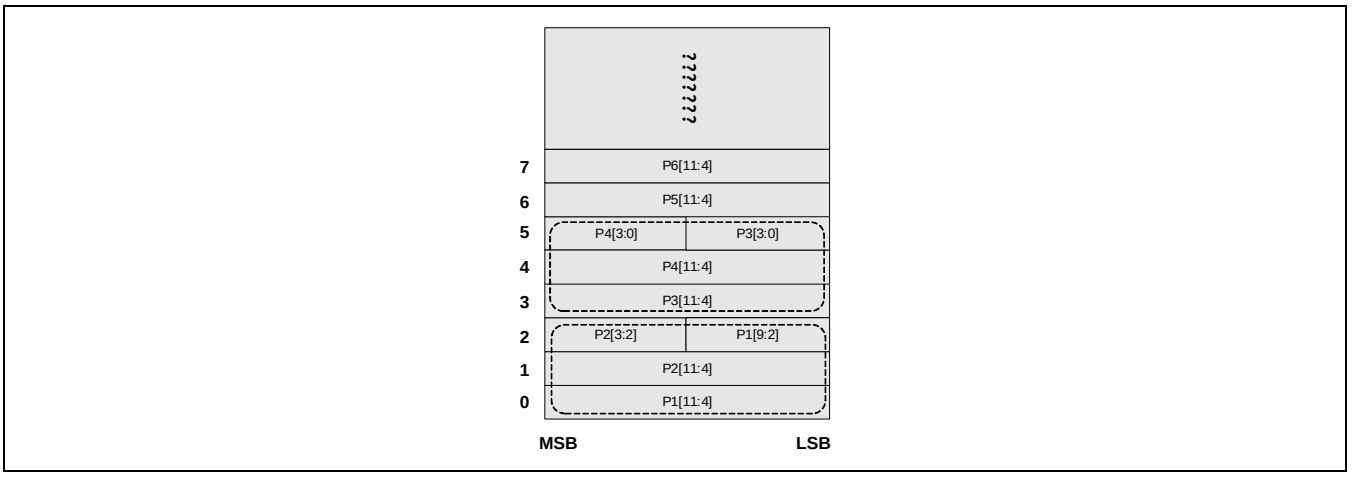


Figure 8.8-6 RAW12 Data Storing Order

6 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
MIPI_DP[3]	B	DP signal for MIPI-DPHY slave data-lane 0	XmipiDP[3]	Dedicated
MIPI_DN[3]	B	DN signal for MIPI-DPHY slave data-lane 0	XmipiDN[3]	Dedicated
MIPI_DP[4]	B	DP signal for MIPI-DPHY slave data-lane 1	XmipiDP[4]	Dedicated
MIPI_DP[4]	B	DN signal for MIPI-DPHY slave data-lane 1	XmipiDN[4]	Dedicated
MIPI_RXCP	B	DP signal for MIPI-DPHY slave clock-lane	XmipiRXCP	Dedicated
MIPI_RXCN	B	DN signal for MIPI-DPHY slave clock-lane	XmipiRXCN	Dedicated

NOTES:

1. I/O direction. I: input, O: output, B: bi-direction
2. Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

7 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
CSIS_CONTROL	0xECC0_0000	R/W	Control Register	0x0000_0000
CSIS_DPHYCTRL	0xECC0_0004	R/W	D-PHY Control Register	0x0000_0000
CSIS_CONFIG	0xECC0_0008	R/W	Configuration Register	0x0000_0000
CSIS_DPHYSTS	0xECC0_000C	R	D-PHY STOP State Register	0x0000_0031
CSIS_INTMSK	0xECC0_0010	R/W	Interrupt Mask Register	0x0000_0000
CSIS_INTSRC	0xECC0_0014	R/W	Interrupt Status Register	0x0000_0000
CSIS_PKTDATA	0xECC0_2000 ~ 0xECC0_3FFC	R	Packet Data (Embedded data or generic short packet data)	0xXXXX_XXXX

NOTE: S_RESETN at MIPI_PHY_CON0 (0xE020_0400) should be '1' before enabling CSIS.

7.1 CONTROL REGISTER (CSIS_CONTROL, R/W, ADDRESS = 0XECC0_0000)

CSIS_CONTROL	Bit	Description	Reset Value
DpDnSwap	[31]	Swapping Dp channel and Dn channel 0 = Default 1 = Swapping	0
Reserved	[30:5]	Reserved	0
SwRst	[4]	Software reset 0 = No reset 1 = Reset All writable registers in CSI2 go back to reset value. After this bit is active for 3 cycles, this bit is de-asserted automatically * Note: Almost MIPI CSI2 block uses "ByteClk" from D-phy. This "ByteClk" is not continuous clock. You must assert software reset if Camera module is turned off.	0
Reserved	[3:1]	Reserved	0
Enable	[0]	CSI2 system On/Off 0 = Off 1 = On If main host turn off CSI2 and CSI2 completes current bus transaction and then let CSI2_READY_CLK_DOWN be high. If this bit is low although CSI2 clock is alive, request from CSI2 is not serviced and kept waiting. Once main host disables CSI2, It should be reset either by software or by hardware before main host enables CSI2.	0

7.2 D-PHY CONTROL REGISTER (CSIS_DPHYCTRL, R/W, ADDRESS = 0XECC0_0004)

This register is used to control of D-phy.

CSIS_DPHYCTRL	Bit	Description	Reset Value
Reserved	[31:1]	Should not change the value	0
DPHYOn	[0]	Enables D-PHY Clock and Data lane 0 = Disable 1 = Enable	0

7.3 CONFIGURATION REGISTER (CSIS_CONFIG, R/W, ADDRESS = 0XECC0_0008)

CSIS_CONFIG	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0
NumOfDatLane	[1:0]	Number of data lane 00 = 1 Data Lane 01 = 2 Data Lane 10 ~ 11 : Reserved	0

7.4 DPHY STATE REGISTER (CSIS_DPHYSTS, R, ADDRESS = 0XECC0_000C)

CSIS_DPHYSTS	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0
StopStateDat	[5:4]	Data lane [1:0] is in Stop State [5] : Data Lane 1 [4] : Data Lane 0	3
Reserved	[3:1]	Reserved	0
StopStateClk	[0]	Clock lane is in Stop State	1

7.5 INTERRUPT MASK REGISTER (CSIS_INTMSK, R/W, ADDRESS = 0XECC0_0010)

This register is used to mask interrupt sources.

CSIS_INTMSK	Bit	Description	Reset Value
MSK_EvenBefore	[31]	Non Image data are received at Even frame and Before Image 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_EvenAfter	[30]	Non Image data are received at Even frame and After Image 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_OddBefore	[29]	Non Image data are received at Odd frame and Before Image 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_OddAfter	[28]	Non Image data are received at Odd frame and After Image 0 = Disables Interrupt 1 = Enables Interrupt	0
Reserved	[27:13]	Reserved	0
MSK_ERR _SOT_HS	[12]	Start of transmission error 0 = Disables Interrupt 1 = Enables Interrupt	0
Reserved	[11:9]	Reserved	0
MSK_ERR_ESC	[8]	Escape entry error 0 = Disables Interrupt 1 = Enables Interrupt	0
Reserved	[7:5]	Reserved	0
MSK_ERR_CTRL	[4]	Control error 0 = Disables Interrupt 1 = Enables Interrupt	0
Reserved	[3]	Reserved	0
MSK_ERR_ECC	[2]	ECC error 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_CRC	[1]	CRC error 0 = Disables Interrupt 1 = Enables Interrupt	0
MSK_ERR_ID	[0]	Unknown ID error 0 = Disables Interrupt 1 = Enables Interrupt	0

7.6 INTERRUPT SOURCE REGISTER (CSIS_INTSRC, R/W, ADDRESS = 0XECC0_0014)

This register is used to identify interrupt sources.

CSIS_INTSRC	Bit	Description	Reset Value
EvenBefore	[31]	Non image data are received at even frame and before image	0
EvenAfter	[30]	Non image data are received at even frame and after image	0
OddBefore	[29]	Non image data are received at odd frame and before image	0
OddAfter	[28]	Non image data are received at odd frame and after image	0
Reserved	[27:14]	Reserved	0
ERR_SOT_HS	[13:12]	Start of transmission error	0
Reserved	[11:10]	Reserved	0
ERR_ESC	[9:8]	Escape entry error	0
Reserved	[7:6]	Reserved	0
ERR_CTRL	[5:4]	Control error	0
Reserved	[3]	Reserved	0
ERR_ECC	[2]	ECC error	0
ERR_CRC	[1]	CRC error	0
ERR_ID	[0]	Unknown ID error - Writing '1' clears status bit - Writing '0' has no effect.	0

7.7 PACKET DATA REGISTER (CSIS_PKTDATA, R, ADDRESS = 0XECC0_2000 ~ 0XECC0_3FFC)

CSIS_PKTDATA	Bit	Description	Reset Value
PktData	[31:0]	Packet Data	Unknown

NOTES

8.9 USB HOST CONTROLLER

1 OVERVIEW

The Universal Serial Bus (USB) supports serial data exchanges between a host computer and a variety of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. Peripherals are attached, configured, used, and detached, while the host and other peripherals continue operation. Familiarity with the Universal Serial Bus Specification, Revision 1.11 and the OHCI specification2 are necessary to fully understand the material contained in this section. Refer to the Universal Serial Bus Specification Revision 1.1 and the Open HCI – Open Host controller Specification for USB for details of the interface operation.

2 FEATURES

S5PC100 supports 2-port USB host interface as follows:

- OHCI Revision 1.0 compatible
- USB Revision 1.1 compatible
- Two down stream ports
- Support both LowSpeed and FullSpeed USB devices

3 OPERATION

3.1 OPERATION

3.1.1 Basic Operations

A USB system consists of four main components: two of these, the client software and the host controller driver, are implemented in software; the other two areas (host controller and the device controller) are implemented in hardware. The host controller driver and the host controller work together to serially transfer data between a shared memory data structure and the USB controller. The USB host controller consists of an OHCI compliant core, a bus interface unit to connect it the system bus, two small FIFOs for buffering data in and out, 2 input pins, 2 output pins, and the transceivers located in the pad unit. The bus interface unit connects to the system bus for register access and for writing and reading of the FIFOs. The registers must be accessed as 32-bit entities on 32-bit aligned addresses.

The serial information transmitted and received by the USB host (USBH) contains layers of communications protocols, the most basic of which are fields. USBH fields include sync, packet identifier, address, Endpoint, Frame Number, data, and CRC fields. Fields are used to produce packets. Depending on the packet function, a different combination and number of fields can be used. Packet types include token, start of frame, data, and handshake. There are four data transfer types define in USB: bulk, control, interrupt, and isochronous. Packets are assembled into groups to produce frames. Data transfers are grouped into two categories: Periodic (Isochronous and Interrupt) and Non-periodic (Control and Bulk). Fields inside of the Endpoint Descriptor (ED) and Transfer Descriptor (TD) memory structures define what type of transfer is to take place.

There are two communication channels between the host controller and the host controller driver. The first channel uses a set of registers to control, status and list pointers. The second communication channel is the Host Controller Communication Area (HCCA). The HCCA contains pointers to the interrupt ED, done queue, and status information associated with start of frame processing. The USB host controller functions as a "smart" DMA operating on linked lists (EDs and TDs) created by the HCD and located in system memory. The HCD assigns an Endpoint Descriptor to each Endpoint in the system. The information in these descriptors include: maximum packet size, the Endpoint address, the speed of the Endpoint, and the direction of data flow. A queue of TDs is linked to the ED for a specific Endpoint. The TDs contain information on data toggle, shared memory buffer location and completion status codes. The HCD creates these ED lists and TD queues then passes control to the UHC for processing (by setting bits two through five of the UHCCON register). The HCD adds to the TD queues and the UHC removes from the queues by linking a finished TD with the Done Queue. The UHC updates fields (such as Current Buffer Pointer and Condition Code) in the TD in system memory space upon completion of a TD. Head pointers to the Bulk and Control ED lists are maintained in the UHC (UHCBHDED and UHCHDED registers). The HCD must initialize these registers. Figure 8.9-1 illustrates a typical list structure.

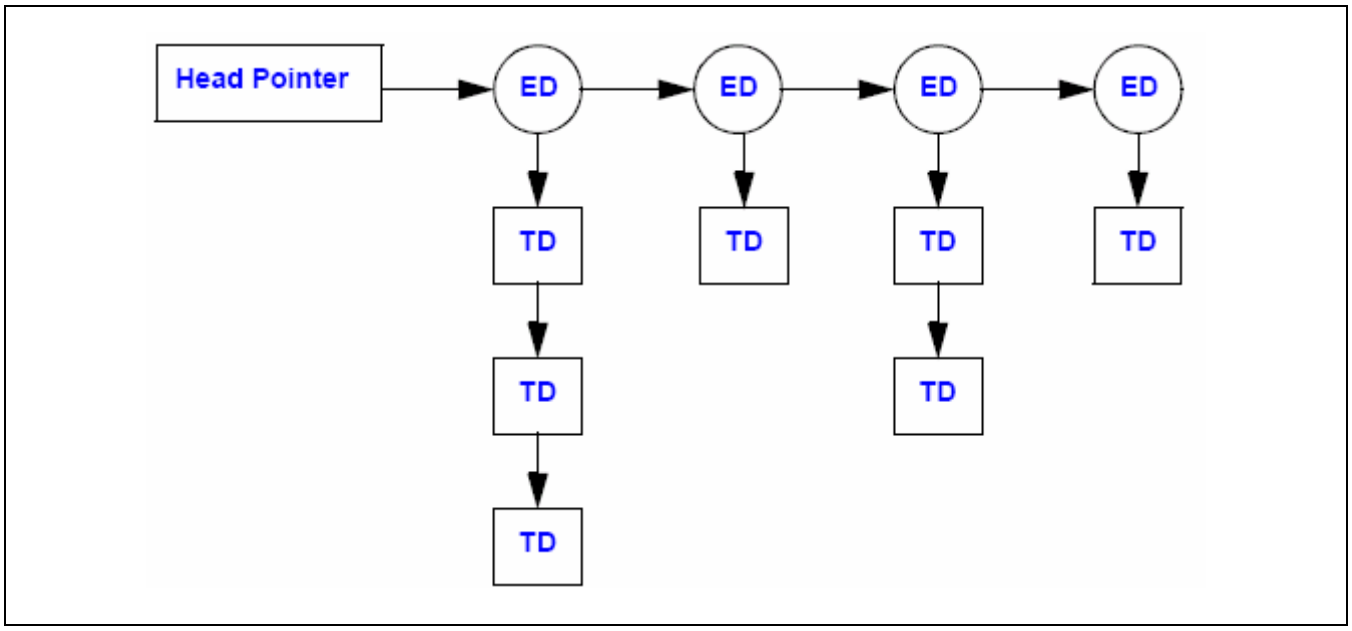


Figure 8.9-1 Typical List Structure

(Operational, resume, suspend, reset), the list processing pointers (the UHCBHDED and UHCHDED registers), list processing enables (bits 5-2 of the UHCCON register), and interrupt enables (in the UHCINTENB register).

NOTES:

1. Remote host wake from sleep is not absolutely USB 1.1 specification-compliant. The USB host controller may not be able to feed back resume signaling to downstream devices within 100 microseconds.

2. In isochronous transfer mode, the latency associated with using VLIO and PC Card memory accesses may violate the 10-µs time limit. As a result, the USB host controller sends a corrupted CRC (an OUT packet) or not issue an ACK; in the latter case, an interrupt occurs (if enabled) and the ISO packet is dropped by software.

3.1.2 USB Host Controller Block Diagram

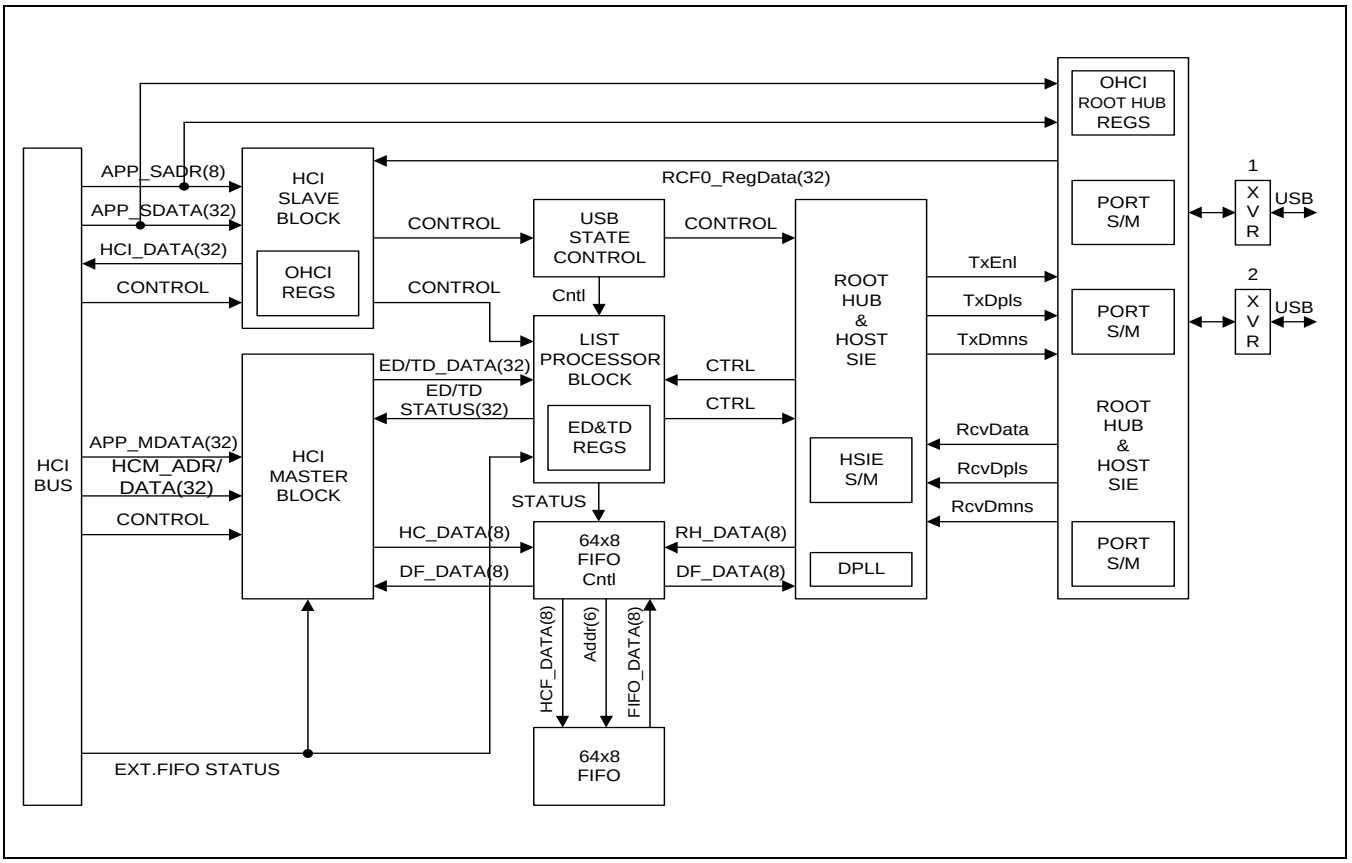


Figure 8.9-2 USB Host Controller Block Diagram

3.1.3 USB Block Diagram

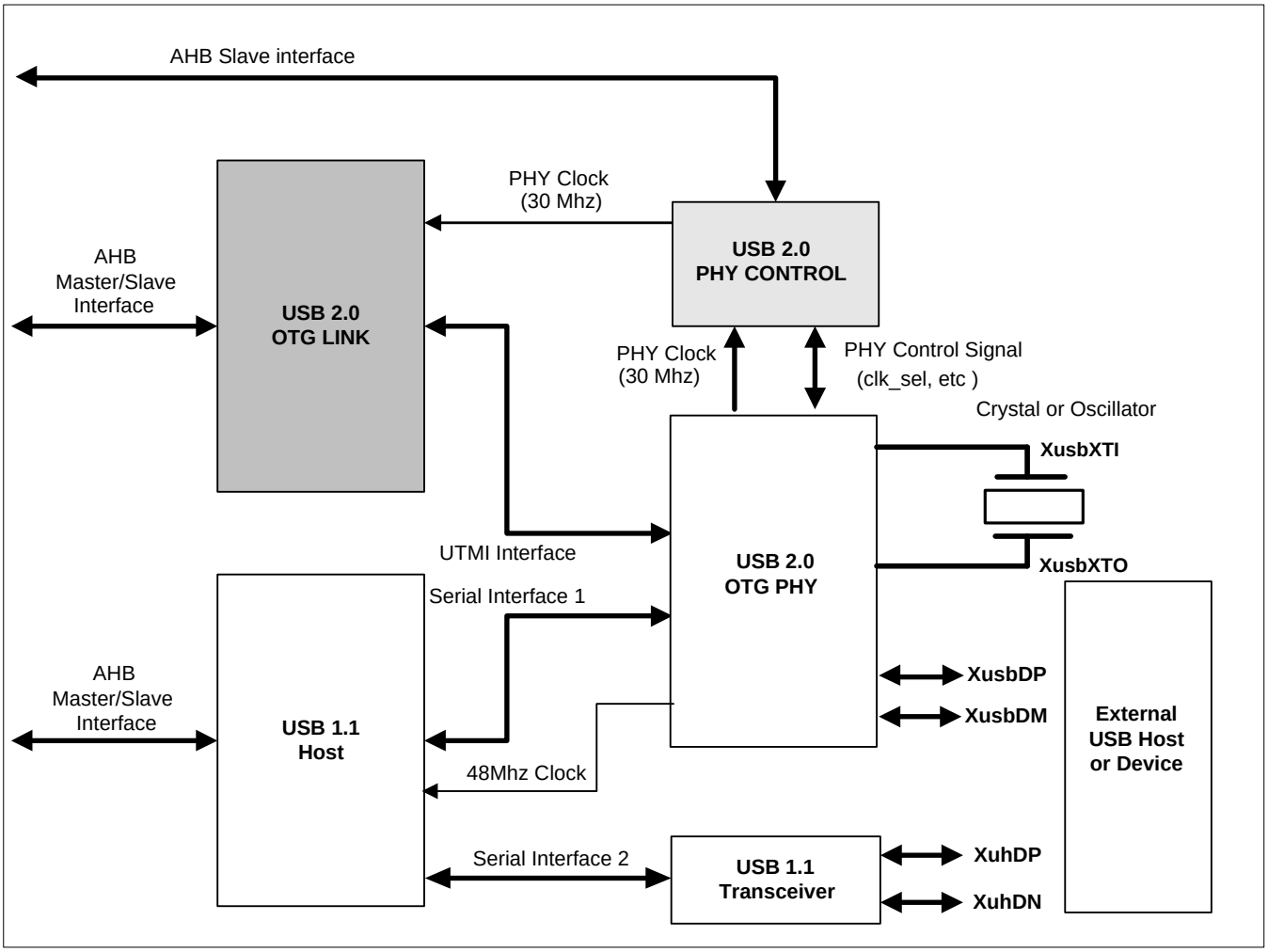


Figure 8.9-3 System Level Block Diagram

The S5PC100x USB system shown in Figure 8.9-3 is configured as following:

1. USB 1.1 Host 1 Port & USB 2.0 OTG 1 Port
2. **USB 1.1 Host 2 Ports**

Note: To enable Serial Interface 1 and use 2 ports of Host 1.1, set the OPHYCLK.serial_mode register bit to 1.

3.1.4 Interrupts

The USB host controller generates two interrupts to the interrupt controller.

- An OHCI USB interrupts (Generated from the Interrupt Status register)
- All other USB host controller event interrupts
  - Buffer-access interrupt
  - Remote wake-up interrupt
  - Port-resume signal interrupt
  - OHCI-initiated interface-clear signal interrupt (transfer abort)
  - USB port-power-over-current-exception interrupts

3.1.5 Power Management

The Open Host Controller Interface Specification for the USB defines a port power-switching mechanism.

- All ports are continuously powered, or the power is switched.
- Power switching is switched globally for all ports or individually switched.

The power switching is independent of the port state, such as its speed, or whether it is connected or enabled. Power-enable features operates only while the USB host clocks are running (the USB clocks must be running to disable or enable power-enable features). This USB host controller has the following features that support power-conservation features of the Open Host Controller Interface:

- USB clock stopping
- Port power-enable

3.1.6 USB Clock Stopping and Power Enable

The USB clocks can be stopped at any time; however, stopping the clocks is recommended if the USB is in the global-suspend state, which is reached if there has been no activity on the USB for more than 5 ms, and the host controller is in the suspend state. In the suspend state, the port (single-ended receivers on the USBH_P and USBH_N pads) is also in low-power mode, but is still able to detect the USB port resume condition and asynchronously generate an interrupt to the wake-up controller of the clock unit.

If a device is not connected to the USB port, the host controller cannot be suspended, and the port (PAD) is not in its low-power mode. To save power if a device is not connected to the USB port (PAD), the port power enable can be disabled before the USB clock is stopped (Refer to OHCI specification).

3.2 PROGRAMMING GUIDES

Programming the USB host controller is very similar to programming the USB host in S3C2443. The major exceptions are in the base address of the OHC registers.

3.2.1 USB Reset

The USB host controller is not fully reset following a S5PC100 processor reset. The full-chip processor reset leaves the USB force host controller reset bit set. To initialize the USB host controller, use the following sequence:

1. Start the USB clocks
2. Wait at least 10 µs
3. Force HC to Reset

The USB host controller is then operational.

3.2.2 USB Suspend

The USB specification defines a power-conservation mechanism called UsbSuspend. Devices and hubs is placed into a suspend state; and if required, the whole system is suspended (global suspend). A USB device enters the suspend state if it does not receive an SOF (keep awake) packet in greater than 3 ms. Software needs to explicitly invoke the suspend state; otherwise, devices automatically receive 1-ms "keep-awake" SOF packets. If there is no USB activity, the host invokes the global suspend state, which causes all connected devices and hubs to enter into their suspend states within 5 ms.

4 I/O DESCRIPTION

USB Host has two paths to transmit and receive data. First path is to use USB special pad and Second path is to use USB PHY.

4.1 USB PAD

Funtion Signal	I/O	Description	Pad	Type
UHOST_DP	Input/Output	USB DP bi –directional I/O PAD	XuhDP	Dedicated
UHOST_DN	Input/Output	USB DN bi –directional I/O PAD	XuhDN	Dedicated

4.2 USB PHY

Funtion Signal	I/O	Description	Pad	Type
USB_DP	Input/Output	Data Plus signal form the USB cable	XusbDP	Dedicated
USB_DM	Input/Output	Data Minus signal form the USB cable	XusbDM	Dedicated

5 REGISTER DESCRIPTIONS

5.1 REGISTER OVERVIEW

Register	Address	R/W	Description	Reset Value
UHCREV	0xED40_0000	R	USB HcRevision Register	
UHCCON	0xED40_0004	R/W	USB HcControl Register	
UHCCONSTAT	0xED40_0008		USB HcCommandStatus Register	
UHCINTSTAT	0xED40_000C		USB HcInterruptStatus Register	
UHCINTENB	0xED40_0010		USB HcInterruptEnable Register	
UHCINTDISB	0xED40_0014	R/W	USB HcInterruptDisable Register	
UHCHCCA	0xED40_0018		USB HcHCCA Register	
UHCPDCURRED	0xED40_001C	R	USB HcPeriodCurrentED Register	
UHCHDED	0xED40_0020		USB HcControlHeadED Register	
UHCCONCURRED	0xED40_0024		USB HcControlCurrentED Register	
UHCBHDED	0xED40_0028		USB HcBulkHeadED Register	
UHCBCURRED	0xED40_002C		USB HcBulkCurrentED Register	
UHCDHD	0xED40_0030	R	USB HcDoneHead Register	
UHCFMI	0xED40_0034	R/W	USB HcFmInterval Register	
UHCFMRM	0xED40_0038	R	USB HcFmRemaining Register	
UHCFMNUM	0xED40_003C	R	USB HcFmNumber Register	
UHCPST	0xED40_0040	R/W	USB HcPeriodicStart Register	
UHCLSTH	0xED40_0044	R/W	USB HcLSThreshold Register	
UHCRHDA	0xED40_0048		USB HcRhDesciptorA Register	
UHCRHDB	0xED40_004C		USB HcRhDescriptorB Register	
UHCRHSTAT	0xED40_0050		USB HcRhStatus Register	
UHCRHPSTAT1	0xED40_0054		USB HcRhPortStatus 1 Register	
UHCRHPSTAT2	0xED40_0058		USB HcRhPortStatus 2 Register	

5.2 DETAILED DESCRIPTION

5.2.1 USB HcRevision Register (HCREV, R, Address = 0xED40_0000)

This read-only register contains the binary-coded decimal (BCD) representation of the HCI specification version that the S5PC100 processor implements. The UHC complies with the *OHCI Revision. 1.0a* specification, therefore this register always has a value of 0x0000_0010. The register organization and individual bit definitions are shown in Table 8.9-2.

Table 8.9-1 UHCREV Bit Definitions

UHCREV	Bit	Description	R/W	Reset Value
Rev	[31:0]	OHCI Specification Revision Number This USB host is compliant to OHCI Revision. 1.0a, therefore this register reads 0x0000_0010.	R	

5.2.2 USB HcControl Register (UHCCON, R/W, Address = 0xED40_0004)

The USB Host Control (UHCCON, shown in Table 8.9-3) register defines the operating modes for the host controller. Most of the fields in this register are modified only by the host controller driver, except for the HostControllerFunctionalState and RemoteWakeupConnected fields.

This is a read/ write register. Ignore reads from reserved bits. Write 1'b0 to reserved bits.

Table 8.9-2 UHCCON Bit Definitions

UHCCON	Bit	Description	R/W	Reset Value
Reserved	[31:11]	Reserved	-	
RWE	[10]	RemoteWakeupEnable The host controller driver uses this bit to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. If this bit is set and the ResumeDetected bit in HcInterruptStatus (Refer to Section 20.8.4) is set, a remote wake-up is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt. 0 = Disable remote wake-up is signaling to the host system. 1 = Enable remote wake-up is signaling to the host system.	R/W	
RWC	[9]	RemoteWakeupConnected This bit indicates whether the UHC supports remote wake-up signaling. If remote wake-up is supported and used by the system, it is the responsibility of system firmware to set this bit after reset. The UHC clears the bit upon a hardware reset, but does not alter it upon a software reset 0 = UHC does not support remote wake-up signaling. 1 = UHC supports remote wake-up signaling.	R/W	
IR	[8]	InterruptRouting This bit determines the routing of interrupts generated by events registered in the UHC Interrupt Status register, UHCINTS (Refer to Section 20.8.4). If clear, all interrupts are routed to the normal host bus-interrupt mechanism. If set, interrupts are routed to the system management interrupt (SMI). The host controller driver clears this bit on a hardware reset, but it does not alter this bit on a software reset. The host controller driver uses this bit as a tag to indicate the ownership of UHC. This implementation of the OHCI host controller does not support SMI. Therefore, the host controller driver must never set this bit. 0 = All interrupts are routed to the normal host bus interrupt mechanism. 1 = Interrupts are routed to the SMI.	R/W	
HCFS	[7:6]	HostControllerFunctionalState This field of two bits displays the current functional state of the USB host controller. 0b00 = USBRESET 0b01 = USBRESUME 0b10 = USBOPERATIONAL 0b11 = USBSUSPEND A transition to USBOPERATIONAL from another state causes an SOF generation to begin 1 ms later. The host controller driver determines whether UHC has started sending SOFs by reading the StartofFrame field of HcInterruptStatus (Refer to Section 20.8.4). This field is changed by UHC if in the USBSUSPEND state. UHC moves from the USBSUSPEND state to the USBRESUME state after detecting the resume signaling from a downstream port.	R/W	

UHCCON	Bit	Description	R/W	Reset Value
		HC enters USBSUSPEND after software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the root hub and asserts subsequent reset signaling to downstream ports. The root hub is defined as the USB hub attached directly to the USB host controller (physically it is a part of the host). Root hubs can have from 1 to 15 downstream ports (this implementation in the PXA27x processor has two downstream ports). Refer to Section 6.2 of the OHCI Revision. 1.0a Specification.		
BLE	[5]	BulkListEnable The host controller driver sets this bit to enable the processing of the bulk list in the next frame. If cleared by the host controller driver, processing of the bulk list does not occur after the next SOF. UHC checks this bit whenever it needs to process the list. If disabled, the host controller driver modifies the list. If HcBulkCurrentED (Refer to Section 20.8.12) is pointing to an ED to be removed, the host controller driver must advance the pointer by updating HcBulkCurrentED before re-enabling list processing. 0 = Disables processing of the bulk list after the next SOF. 1 = Enables the processing of the bulk list in the next frame.	R/W	
CLE	[4]	ControlListEnable The host controller driver sets this bit to enable the processing of the control list in the next frame. If cleared by the host controller driver, processing of the control list does not occur after the next SOF. UHC must check this bit whenever it determines to process the list. If disabled, the host controller driver modifies the list. If HcControlCurrentED (Refer to Section 20.8.10) is pointing to an ED to be removed, the host controller driver must advance the pointer by updating HcControlCurrentED before re-enabling list processing. 0 = Disables processing of the control list after the next SOF. 1 = Enables the processing of the control list in the next frame.	R/W	
IE	[3]	IsochronousEnable The host controller driver uses this bit to enable/ disable processing of isochronous EDs. While processing the periodic list in a frame, UHC checks the status of this bit if it finds an isochronous ED (F = 1). If set (enabled), UHC continues processing the EDs. If cleared (disabled), UHC halts processing of the periodic list, and begins processing the bulk/ control lists. Setting this bit is guaranteed to take effect in the next frame (not the current frame). 0 = Disables processing of isochronous EDs. 1 = Enables processing of isochronous EDs.	R/W	
PLE	[2]	PeriodicListEnable This bit is set to enable the processing of the periodic list in the current frame. If cleared by the host controller driver,	R/W	

UHCCON	Bit	Description	R/W	Reset Value
		processing of the periodic list does not halt until after the next SOF. UHC must check this bit before it starts processing the list. 0 = Disables processing of the periodic list. 1 = Enables processing of the periodic list.		
CBSR	[1:0]	ControlBulkServiceRatio This bit specifies the service ratio between control and bulk EDs. Before processing any of the non-periodic lists, UHC must compare the ratio specified with its internal count on how many non-empty control EDs have been processed, in determining whether to continue serving another control ED or switching to bulk EDs. The internal count is retained if frame boundary is crossed. In case of reset, the host controller driver is responsible for restoring this value. The number of control EDs over bulk EDs served is as follows: 2'b00 = 1:1 2'b01 = 2:1 2'b10 = 3:1 2'b11 = 4:1	R/W	

5.2.3 USB HcCommandStatus Register (UHCCONSTAT, Address = 0xED40_0008)

The USB host controller uses the USB HOST Command Status (UHCCONSTAT, shown in Table 8.9-4) register to receive commands that the host controller driver issues, as well as reflecting the current status of the host controller. To the host controller driver, it appears to be a write-to-set register. Bits written as 1 become set in the register, while bits written as 0 remain unchanged in the register. In this way, the host controller driver issues multiple distinct commands to the host controller without concern for corrupting previously issued commands. The host controller driver has read access to all of these bits.

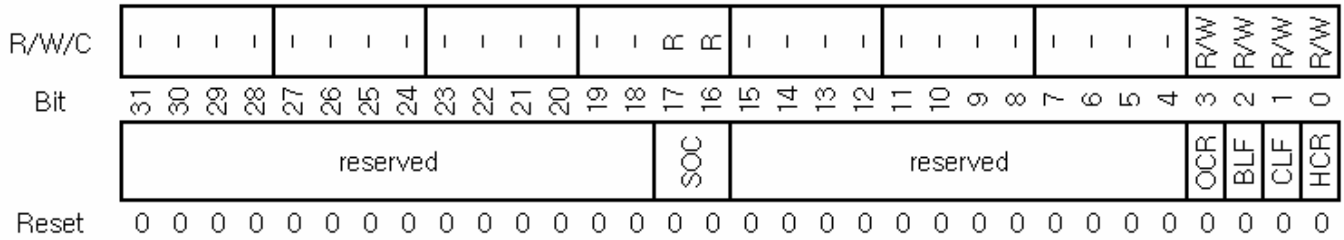


Table 8.9-3 UHCCONSTAT Bit Definitions

UHCCONSTAT	Bit	Description	R/W	Reset Value
Reserved	[31:18]	Reserved	-	
SOC	[17:16]	SchedulingOverrunCount The SchedulingOverrunCount field indicates the number of frames with which the host controller has detected the scheduling overrun error, which occurs if the periodic list does not complete before EOF. If a scheduling-overrun error is detected, the host controller increments the counter and sets the SchedulingOverrun field in the UHC Interrupt Status register, UHCINTS. These bits are incremented on each scheduling overrun error. It is initialized to 0b00 and wraps around at 0b11. This is incremented if a scheduling overrun is detected even if SchedulingOverrun in HcInterruptStatus (Refer to Section 20.8.4) has already been set. This bit field and the scheduling overrun interrupt are used by the host controller driver to monitor any persistent scheduling problems. These bits are incremented on each scheduling overrun error.	R	
Reserved	[15:4]	Reserved	-	
OCR	[3]	OwnershipChangeRequest The host controller driver sets this bit to request a change in UHC control If this bit is set, UHC sets OwnershipChange in HcInterruptStatus (Refer to Section 20.8.4). If read, this bit always returns 0. This implementation of the OHCI host does not support SMI. Therefore, software must never write 0b1 to this bit. 0 = No HCD request for a change in control of the UHC is pending. 1 = HCD is requesting a change change in UHC control	R/W	

UHCCONSTAT	Bit	Description	R/W	Reset Value
BLF	[2]	BulkListFilled This bit indicates whether there are any TDs on the bulk list. It is set by the host controller driver whenever it adds a TD to an ED in the bulk list. If UHC begins to process the head of the bulk list, it checks the BulkListFilled bit (BLF). As long as BulkListFilled is 0, UHC does not start processing the bulk list. If BulkListFilled is 1, UHC starts processing the bulk list and sets BLF to 0. If UHC finds a TD on the list, then UHC sets BulkListFilled to 1, causing the bulk list processing to continue. If no TD is found on the bulk list, and if the host controller driver does not set BulkListFilled, then BulkListFilled is still 0 if UHC completes processing the bulk list, and bulk list processing stops. 0 = Bulk list processing stops. 1 = HC starts processing the bulk list and sets BLF to 0.	R/W	
CLF	[1]	ControlListFilled This bit indicates whether there are any TDs on the control list. It is set by the host controller driver whenever it adds a TD to an ED in the control list. If UHC begins to process the head of the control list, it checks the control list filled (CLF) bit. As long as CLF is 0, UHC does not start processing the Control list. If CF is 1, UHC starts processing the control list and sets CLF to 0. If UHC finds a TD on the list, then UHC sets CLF to 1 causing the control list processing to continue. If no TD is found on the control list, and if the host controller driver does not set CLF, then it is still 0, if UHC completes processing the control list, and control list processing stops. 0 = UHC does not start processing the control list. 1 = UHC starts processing the control list and sets ControlListFilled to 0.	R/W	
HCR	[0]	HostControllerReset The host controller driver sets this bit to initiate a software reset of UHC. Regardless of the functional state of the UHC, it moves to the USBSUSPEND state in which most of the operational registers are reset. The exceptions to that are stated in the OHCI Spec (for example, the InterruptRouting field of the UHC Host Control register). While this bit is set, no USB host bus accesses are allowed (writes or reads of the OHCI registers). This bit is cleared by the UHC upon the completion of the reset operation (which is completed within 10 μs.). This bit, if set, does not cause a reset to the root hub and no subsequent reset signaling is asserted to its downstream ports. 0 = Controller is not in reset. 1 = Reset the host controller.	R/W	

5.2.4 USB HcInterruptStatus Register (UHCINTSTAT, Address = 0xED40_000C)

The USB INTerrupt STATus (UHCINTSTAT) register provides status on various events that cause hardware interrupts. If an event occurs, the host controller sets the corresponding bit in this register. If a bit is set, a hardware interrupt is generated if the interrupt is enabled in the HcInterruptEnable register and the MasterInterruptEnable bit is set. The host controller driver clears specific bits in this register by writing 1 to bit positions to be cleared. The host controller driver cannot set any of these bits. The host controller never clears these bits (except for during a system reset). Because of the difference in clock frequencies between the system bus and the USB host controller core, there is latency from the time a write occurs on the system bus until the actual register bit is cleared. The HCD needs to take into this delay into consideration.

*C; To clear this bit, write 1'b1 to it.

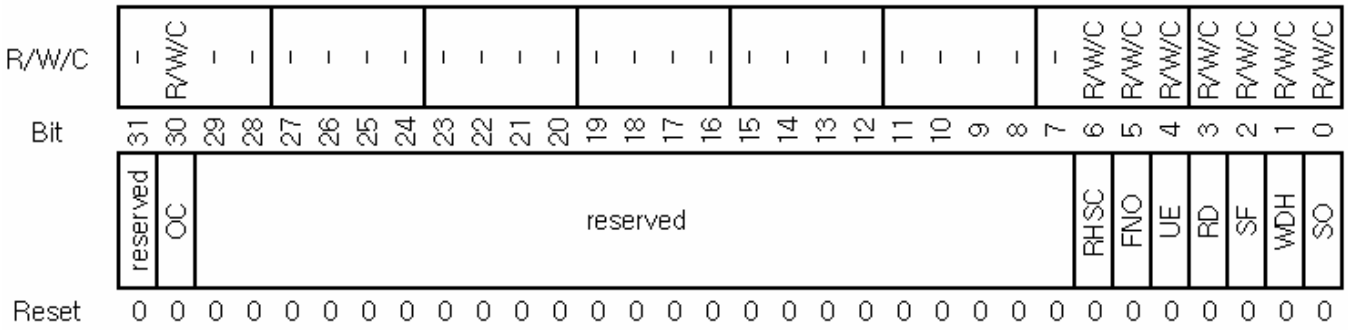


Table 8.9-4 UHCINTSTAT Bit Definitions

UHCINTSTAT	Bit	Description	R/W	Reset Value
Reserved	[31]	Reserved	-	
OC	[30]	OwnershipChange This bit is set by UHC if the host controller driver sets the OwnershipChangeRequest field in HcCommandStatus. However, this implementation of the OHCI host does not support SMI. Therefore, software must never write 1'b1 to the UHCCOMSTAT[OCR] bit. This bit is tied to 0 because the SMI pin is not implemented in this design.	R/WC	
Reserved	[29:7]	Reserved	-	
RHSC	[6]	RootHubStatusChange This bit is set if the content of HcRhStatus or the content of any of HcRhPortStatus for port 1 and 2 has changed. 0 = The contents of the above four registers has not changed. 1 = The contents of HcRhStatus or HcRhPortStatus[2:1] has changed.	R/WC	
FNO	[5]	FrameNumberOverflow This bit is set if the most significant bit (15) of HcFmNumber changes value, from 0 to 1 or from 1 to 0, and after HccaFrameNumber has been updated. 0 = Otherwise 1 = HcFmNumber (bit 15) changed in value, from 0 to 1 or from 1 to 0, or HccaFrameNumber has been updated	R/WC	

UHCINTSTAT	Bit	Description	R/W	Reset Value
UE	[4]	UnrecoverableError This bit is set if UHC detects a system error not related to USB. UHC must not proceed with any processing or signaling before the system error has been corrected. System errors such as incorrect buffer addressing offsets or condition codes (Refer to OHCI Revision 1.0a specification.). The HCD clears this bit by either writing a 1 to it, or by resetting the UHC OHCI core by generating a software reset (writing a 1 to the UHCCOMSTAT[HCR] bit). 0 = No system error is detected. 1 = UHC detected a system error not related to USB.	R/WC	
RD	[3]	ResumeDetected This bit is set if UHC detects that a device on the USB is asserting resume-signaling. It is the transition from no-resume-signaling to resume-signaling that sets this bit. This bit is not set if HCD sets the USBRESUME state. 0 = No device is asserting resume-signaling. 1 = A device on the USB is asserting resume-signaling.	R/WC	
SF	[2]	StartofFrame This bit is set by UHC start of each frame and after the update of HccaFrameNumber. UHC also generates a SOF token at the same time. 1 = A start of a frame or an update of HccaFrameNumber has occurred	R/WC	
WDH	[1]	WritebackDoneHead This bit is set immediately after the UHC has written to the value of the UHCDHead to an external the memory location referred to (by the OHCI spec) as HccaDoneHead. Additional updates of the HccaDoneHead do not occur until this bit has been cleared. HCD must clear this bit only after it has saved the content of HccaDoneHead. 1 = UHC has written HcDoneHead to HccaDoneHead.	R/WC	
SO	[0]	SchedulingOverrun This bit is set if the USB schedule for the current frame overruns, and after the update of HccaFrameNumber. A scheduling overrun also causes the SchedulingOverrunCount of UHCCOMS to be incremented To avoid this condition, take care to enable the periodic lists as early in the frame as possible. Processing of the periodic lists occurs with no TDs scheduled. If a list is enabled at the end of the frame, and the reading of the lists occurs beyond the frame boundary the SO bit is set to 0b1 and the SOC counter is incremented. 1 = The USB schedule for the current frame has overrun or the update of HccaFrameNumber has occurred.	R/WC	

5.2.5 USB HcInterruptEnable Register (UHCINTENB, R/W, Address = 0xED40_0010)

Each *enable* bit in the UHC Interrupt Enable (UHCINTENB) register corresponds to an associated *interrupt* bit in the UHCINTSTAT register. The UHC Interrupt Enable register controls which events generate OHCI hardware interrupt. If a bit is set in UHCINTSTAT, the UHC Interrupt Status register, the corresponding bit in the UHC Interrupt Enable register is set, and UHCINTENB[MIE], the MasterInterruptEnable bit is set, an OCHI interrupt request is sent to the processor interrupt controller. Writing a 1 to a bit in this register sets the corresponding bit, whereas writing a 0 to a bit in this register leaves the corresponding bit unchanged. On a read, the current value of this register is returned. To clear a bit in this register, write 1'b1 to the Interrupt Disable register (UHCINTDISB). The register organization and individual bit definitions are shown in

Table 8.9-6. All reserved bits are read as unknown values and must be written with only a 0. A question mark indicates the value is unknown at reset.

Table 8.9-5 UHCINTENB Bit Definitions

UHCINTENB	Bit	Description	R/W	Reset Value
MIE	[31]	Master Interrupt Enable 0 = Ignore 1 = Enables interrupt generation due to events specified in the other bits of this register.	R/W	
OC	[30]	Ownership Change 0 = Ignore 1 = Enables interrupt generation due to ownership change.	R/WS	
Reserved	[29:7]	Reserved	-	
RHSC	[6]	Root Hub Status Change 0 = Ignore 1 = Enables interrupt generation due to root hub status change.	R/WS	
FNO	[5]	Frame Number Overflow 0 = Ignore 1 = Enables interrupt generation due to frame number overflow.	R/WS	
UE	[4]	Unrecoverable Error 0 = Ignore 1 = Enables interrupt generation due to unrecoverable error.	R/WS	
RD	[3]	Resume Detect	R/WS	

UHCINTENB	Bit	Description	R/W	Reset Value
		0 = Ignore 1 = Enables interrupt generation due to resume detect.		
SF	[2]	Start of Frame 0 = Ignore 1 = Enables interrupt generation due to start of frame.	R/WS	
WDH	[1]	Writeback HcDoneHead 0 = Ignore 1 = Enables interrupt generation due to HcDoneHead writeback.	R/WS	
SO	[0]	Scheduling Overrun 0 = Ignore 1 = Enables interrupt generation due to scheduling overrun.	R/WS	

5.2.6 USB HcInterruptDisable Register (UHCINTDISB, R/W, Address = 0xED40_0014)

Each disable bit in the UHC Interrupt Disable (UHCINTDISB, shown in Table 8.9-7) register corresponds to an associated interrupt bit in UHCINTSTAT, the UHC Interrupt Status register. The UHC Interrupt Disable register is coupled with the UHC Interrupt Enable register. Thus, writing a 1 to a bit in this register clears the corresponding bit in the UHC Interrupt Enable register, whereas writing a 0 to a bit in this register leaves the corresponding bit in the UHC Interrupt Enable register unchanged. On a Read, the current value of the UHC Interrupt Enable register is returned.

Table 8.9-6 UHCINTDISB Bit Definitions

UHCINTDISB	Bit	Description	R/W	Reset Value
MIE	[31]	Master Interrupt Enable 0 = Ignore 1 = Disables interrupt generation due to events specified in the other bits of this register. This field is set after a hardware or software reset	R/W	
OC	[30]	Ownership Change 0 = Ignore 1 = Disables interrupt generation due to ownership change.	R/W	
Reserved	[29:7]	Reserved	-	
RHSC	[6]	Root Hub Status Change 0 = Ignore 1 = Disables interrupt generation due to root hub status change.	R/W	
FNO	[5]	Frame Number Overflow 0 = Ignore 1 = Disables interrupt generation due to frame number overflow.	R/W	
UE	[4]	Unrecoverable Error 0 = Ignore 1 = Disables interrupt generation due to unrecoverable error.	R/W	
RD	[3]	Resume Detect 0 = Ignore 1 = Disables interrupt generation due to resume detect.	R/W	
SF	[2]	Start of Frame 0 = Ignore 1 = Disables interrupt generation due to start of frame.	R/W	
WDH	[1]	Writeback HcDoneHead 0 = Ignore	R/W	

UHCINTDISB	Bit	Description	R/W	Reset Value
		1 = Disables interrupt generation due to HcDoneHead Writeback.		
SO	[0]	Scheduling Overrun 0 = Ignore 1 = Disables interrupt generation due to scheduling overrun.	R/W	

5.2.7 USB HcHCCA Register (UHCHCCA, Address = 0xED40_0018)

The UHCHCCA register contains the exact physical address of the host controller communication area.

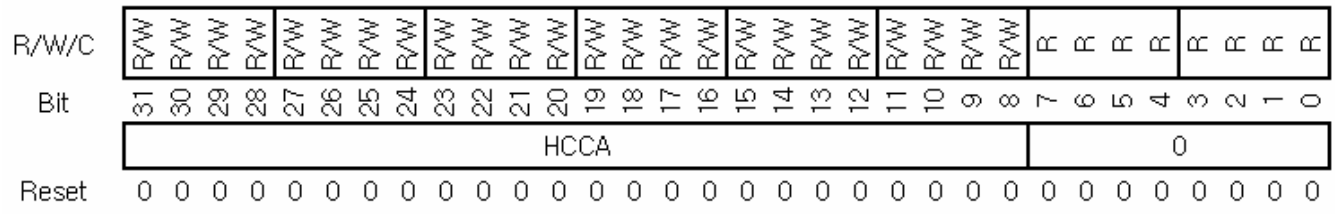


Table 8.9-7 UHCHCCA Bit Definitions

UHCHCCA	Bit	Description	R/W	Reset Value
HCCA	[31:8]	Host Controller Communication Area Base address of the host controller communication area. This register is a pointer to an area that holds the control structures and the interrupt table that both the host controller and the host controller driver access. The minimum alignment for the HCCA is 256 bytes; therefore, bits 0 through 7 of this register always return 0 if read. Refer to Chapter 4 of the OHCI Revision. 1.0a Specification.	R/W	
-	[7:0]	Fixed at 0	R	

5.2.8 USB HcPeriodCurrentED Register (UHCPDCURRED, R, Address = 0xED40_001C)

The UHCPDCURRED register contains the exact physical address of the current Isochronous or Interrupt Endpoint Descriptor. The register organization and individual bit definitions are shown in Table 8.9-9. The lower 4 bits are read as 0 and are unaffected by writes.

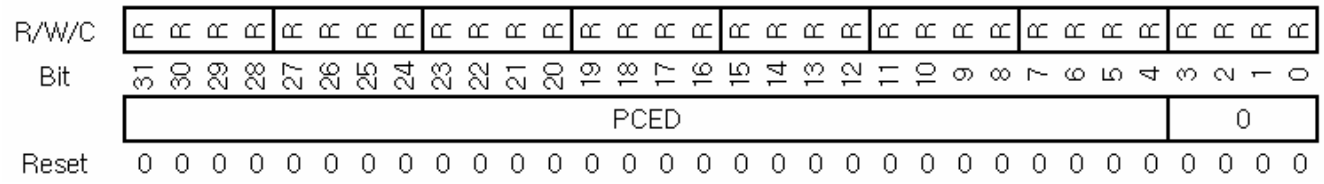


Table 8.9-8 UHCPDCURRED Bit Definitions

UHCPDCURRED	Bit	Description	R/W	Reset Value
PCED	[31:4]	PeriodCurrent Endpoint Descriptor This register is used by the UHC to point to the head of one Periodic list which is processed in the current frame. The content of this register is updated by the UHC after a Periodic ED has been processed. The HCD reads the content in determining which ED is currently being processed at the time of reading. This address is 32 byte aligned, therefore, the lower 4 bits are always 0	R	
-	[3:0]	Fixed at 0	R	

5.2.9 USB HcControlHeadED Register (UHCHDED, Address = 0xED40_0020)

The UHCHDED register contains the physical address of the current Endpoint Descriptor of the control list. The register organization and individual bit definitions are shown in Table 8.9-10. All reserved bits are read as 0 and are unaffected by writes.

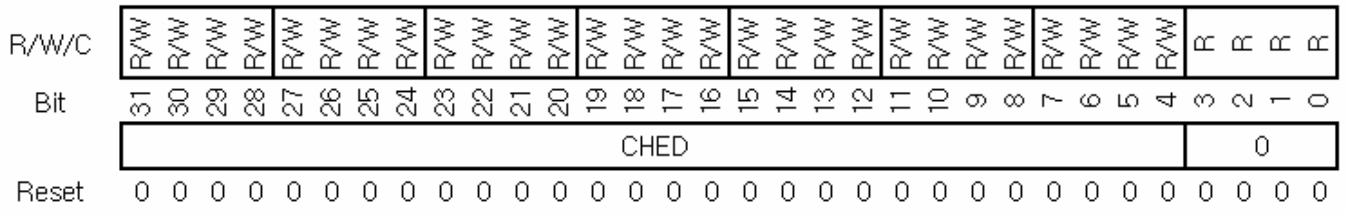


Table 8.9-9 UHCHDED Bit Definitions

UHCHDED	Bit	Description	R/W	Reset Value
CHED	[31:4]	ControlHead Endpoint Descriptor The UHC traverses the control list starting with the HcControlHeadED pointer (the UHCCHED register); the content is loaded from the HCCA during the initialization of the UHC. This address pointer is 32 byte aligned, therefore, the lower 4 bits are always 0.	R/W	
-	[3:0]	Fixed at 0	R	

5.2.10 USB HcControlCurrentED Register (UHCCONCURRED, Address = 0xED40_0024)

The UHCCONCURRED register contains the physical address of the current Endpoint Descriptor of the control list. The register organization and individual bit definitions are shown in Table 8.9-11. All reserved bits are read as 0 and are unaffected by writes.

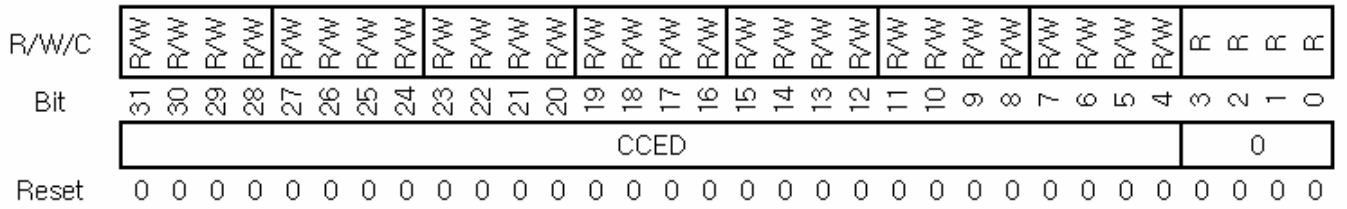


Table 8.9-10 UHCCONCURRED Bit Definitions

UHCCONCURRED	Bit	Description	R/W	Reset Value
CCED	[31:4]	ControlCurrent Enpoint Descriptor This pointer advances to the next ED after serving the present one. The UHC continues processing the list from where it left off in the last frame. If it reaches the end of the Control list, the UHC checks the ControlListFilled bit of in HcCommandStatus (UHCCOMS[CLF]). If set, it copies the content of HcControlHeadED (the UHCCHED register) to the UHC Control Current ED (this register) and clears the UHCCOMS[CLF] bit. If UHCCOMS[CLF] is not set, the UHC does nothing. The HCD is allowed to modify this register if the ControlListEnable bit of HcControl (UHCHCON[CLE]) is cleared. If UHCHCON[CLE] is set, the HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list. This pointer/address is 32-byte aligned, therefore, the lower 4 bits are always 0.	R/W	
-	[3:0]	Fixed at 0	R	

5.2.11 USB HcBulkHeadED Register (UHCBHDED, Address = 0xED40_0028)

The UHCBHDED register contains the physical address of the first Endpoint Descriptor of the Bulk list. The register organization and individual bit definitions are shown in Table 8.9-12. The lower 4 bits are read as 0 and are unaffected by Writes.

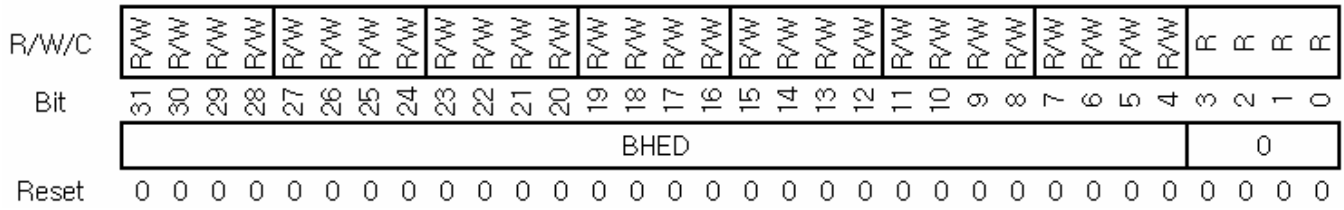


Table 8.9-11 UHCBHDED Bit Definitions

UHCBHDED	Bit	Description	R/W	Reset Value
BHED	[31:4]	BulkHead Endpoint Descriptor The UHC traverses the Bulk list starting with the HcBulkHeadED pointer (UHCBHED register, Refer to Section 20.8.11); the content is loaded from the HCCA during the initialization of the UHC. This pointer/address is 32 byte aligned, therefore, the lower 4 bits are always 0.	R/W	
-	[3:0]	Fixed at 0	R	

5.2.12 USB HcBulkCurrentED Register (UHCBCURRED, Address = 0xED40_002C)

The UHCBCURRED register contains the physical address of the current Endpoint of the Bulk list. The register organization and individual bit definitions are shown in Table 8.9-13. The lower 4 bits are read as 0 and are unaffected by Writes.

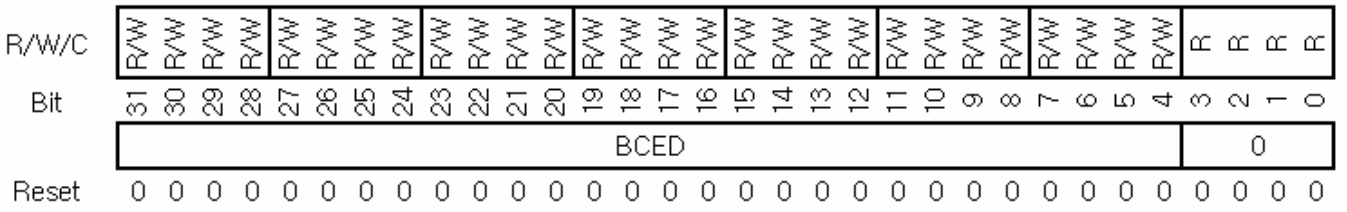


Table 8.9-12 UHCBCURRED Bit Definitions

UHCBCURRED	Bit	Description	R/W	Reset Value
BCED	[31:4]	Bulk Current Endpoint Descriptor This pointer advances to the next ED after the UHC has served the present one. The UHC continues processing the Bulk list from where it left off in the last frame. If it reaches the end of the Bulk list, the UHC checks the ControlListFilled bit of HcControl (UHCHCON[CLF]). If set, the UHC copies the content of the UHC Bulk Head ED register to UHC Bulk Current ED and clears the UHCHCON[CLF] bit. If the UHCHCON[CLF] bit is not set, the UHC does nothing. The HCD is only allowed to modify this register if the BulkListEnable bit of the UHC Control register (UHCHCON[BLE]) is cleared. If the UHCHCON[BLE] bit is set, the HCD only reads the instantaneous value of this register. This register is initially set to zero to indicate the end of the Bulk list. As the Bulk list is served in a round-robin fashion, the Endpoints are ordered according to their insertion to the list. This address is 32 byte aligned, therefore, the lower 4 bits are always 0	R/W	
-	[3:0]	Fixed at 0	R	

5.2.13 USB HcDoneHead Register (UHCDHD, R, Address = 0xED40_0030)

The UHCDHD register contains the physical address of the last completed Transfer Descriptor that was added to the Done queue. The register organization and individual bit definitions are shown in Table 8.9-14. The lower 4 bits are read as 0 and are unaffected by Writes.

Table 8.9-13 UHCDHD Bit Definitions

UHCDHD	Bit	Description	R/W	Reset Value
DHED	[31:4]	DoneHead If a TD is completed, the UHC writes the content of the UHC Done Head register to the NextTD field of the TD. The UHC then overwrites the content of the UHCDHead with the address of this TD. Whenever the UHC writes the content of this register to HCCA, this register is set to zero, and the UHC also sets the WritebackDoneHead bit of the UHC Interrupt Status register (UHCINTS[WBDH]). In normal operation, the host controller driver does not read this register as its content is periodically written to the HCCA. This address is 32 byte aligned, therefore, the lower 4 bits are always 0.	R	
-	[3:0]	Fixed at 0	R	

5.2.14 USB HcFmInterval Register (UHCFMI, R/W, Address = 0xED40_0034)

The UHC Frame Interval (UHCFMI) register contains a 14-bit value that indicates the bit-time interval in a frame (between two consecutive SOFs), and a 15-bit value indicating the full-speed, maximum packet size that the host controller transmit or receive without causing scheduling overrun. The host controller driver carries out a minor adjustment on the frame interval by writing a new value over the present one at each SOF. This mechanism provides the programmability necessary for the host controller to synchronize with an external clocking resource and to adjust any unknown local clock offset. A write to this register by the HCD takes affect in the next frame (the host controller delays updating its frame interval counter until the next SOF). The register organization and individual bit definitions are shown in Table 8.9-17 . All *reserved* bits are read as unknown values and must be written with only a 0. A question mark indicates the value is unknown at reset.

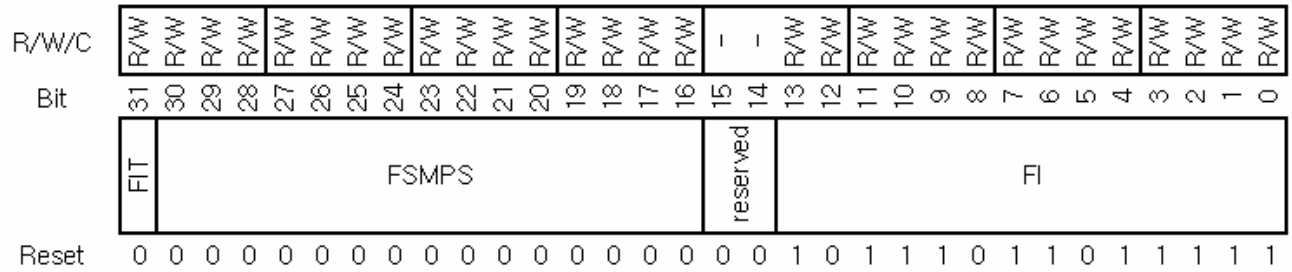


Table 8.9-14 UHCFMI Bit Definitions

UHCFMI	Bit	Description	R/W	Reset Value
FIT	[31]	FrameIntervalToggle HCD toggles this bit whenever it loads a new value to FrameInterval	R/W	
FSMPS	[30:16]	FSLargestDataPacket This field specifies a value that is loaded into the Largest Data Packet counter at the beginning of each frame. The counter value represents the largest amount of data in bits that is sent or received by the UHC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD.	R/W	
Reserved	[15:14]	Reserved	-	
FI	[13:0]	FrameInterval This specifies the interval between two consecutive SOFs in bit-times. The nominal value is set at 11,999. HCD must store the current value of this field before resetting UHC (by setting the HostControllerReset field of HcCommandStatus, Refer to Section 20.8.3) as this causes the UHC to reset this field to its nominal value. The HCD chooses to restore the stored value upon the completion of the reset sequence.	R/W	

5.2.15 USB HcFmRemaining Register (UHCFMRM, R, Address = 0xED40_0038)

The UHC Frame Remaining (UHCFMRM) register is a 14-bit down counter showing the bit-time remaining in the current frame.

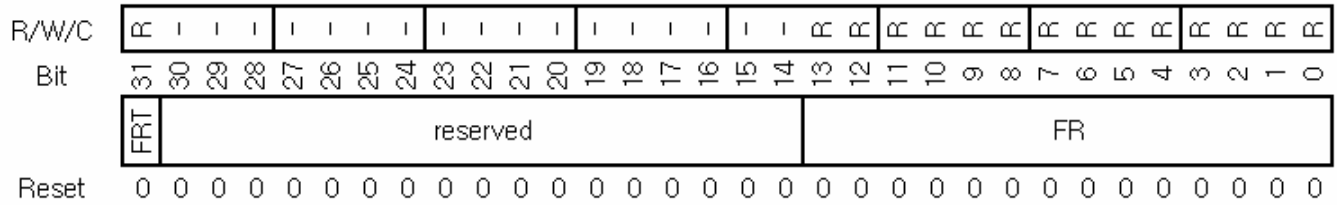


Table 8.9-15 UHCFMRM Bit Definitions

UHCFMRM	Bit	Description	R/W	Reset Value
FRT	[31]	Frame Remaining Toggle Loaded from the Frame Interval Toggle field of the UHC Frame Interval register (UHCFMI[FIT]) whenever the Frame Remaining value (UHCFMR[FR]) reaches 0. This bit is used by the HCD for the synchronization between frame interval and frame remaining.	R	
Reserved	[30:14]	Read as unknown and must be written as zero.	-	
FR	[13:0]	Frame Remaining This counter is decremented at each bit-time. If it reaches zero, it is reset by loading the frame interval value specified in the UHC Frame Interval register at the next bit-time boundary. If the USBOPERATIONAL state is entered, the UHC re-loads the content with the frame interval (UHCFMI[FI]) and uses the updated value from the next SOF.	R	

5.2.16 USB HcFmNumber Register (UHCFMNUM, R, Address = 0xED40_003C)

The UHC Frame Number (UHCFMNUM) register is a 16-bit counter which provides a timing reference.

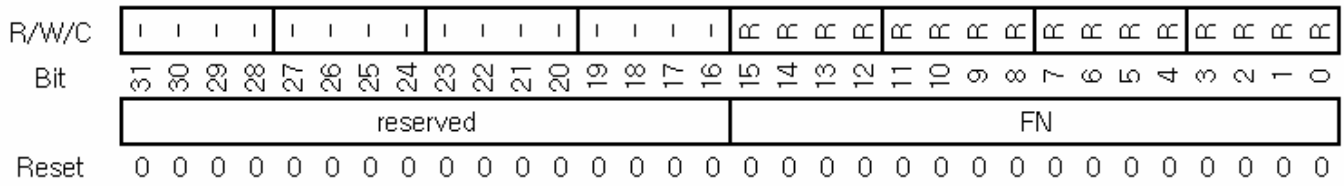


Table 8.9-16 UHCFMNUM Bit Definitions

UHCFMNUM	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Read as unknown and must be written as zero.	-	
FN	[15:0]	Frame Number This 16-bit counter provides a timing reference among events happening in the host controller and the host controller driver. The host controller driver uses the 16-bit value specified in this register and generate a 32-bit Frame Number without requiring frequent register access. This is incremented if UHCFR is re-loaded. It is rolled over to 0x0 after 0xFFFF. If the USBOPERATIONAL state is entered, this is incremented automatically. The content is written to the HCCA after the UHC has incremented the Frame Number at each frame boundary and sent a SOF but before the UHC reads the first ED in that Frame. After writing to the HCCA, the UHC sets the Start of Frame bit in the UHC Interrupt Status register (UHCINTS[SOF])	R	

5.2.17 USB HcPeriodicStart Register (UHCPST, R/W, Address = 0xED40_0040)

The UHC Periodic Start (UHCPST) register has a 14-bit programmable value that determines the earliest time the UHC must start processing the periodic list.

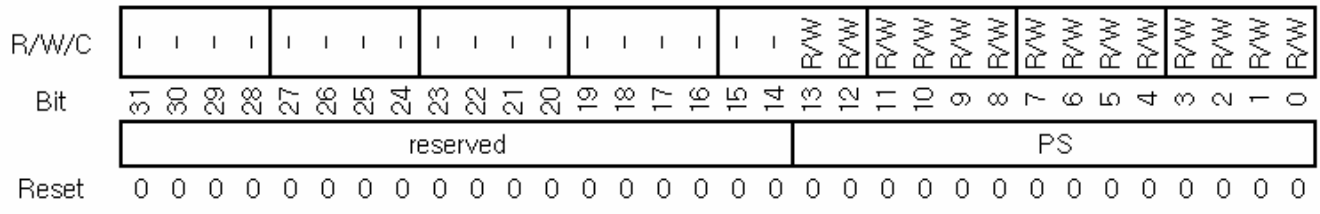


Table 8.9-17 UHCPST Bit Definitions

UHCPST	Bit	Description	R/W	Reset Value
Reserved	[31:14]	Read as unknown and must be written as zero.	-	
PS	[13:0]	PeriodicStart After a hardware reset, this field is cleared. This is then set by the HCD during the UHC initialization. The value is calculated roughly as 10% off from the UHC frame interval value. A typical value is 0x3E67. If UHC Frame Remaining (UHCFMR[FR]) value reaches the value specified in this register, processing of the periodic lists has priority over Control/Bulk processing. The UHC therefore starts processing the interrupt list after completing the current control or bulk transaction that is in progress	R/W	

5.2.18 USB HcLSThreshold Register (UHCLSTH, R/W, Address = 0xED40_0044)

The UHC Low-Speed Threshold register is used by the UHC to determine whether to commit to the transfer of a maximum of 8-byte LS packet before EOF. The register organization and individual bit definitions are shown in Table 8.9-20. All *reserved* bits are read as unknown values and must be written with only a 0. A question mark indicates the value is unknown at reset.

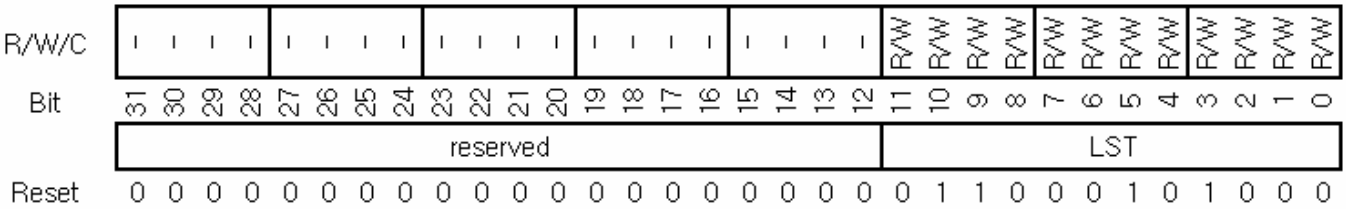


Table 8.9-18 UHCLSTH Bit Definitions

UHCLSTH	Bit	Description	R/W	Reset Value
Reserved	[31:12]	Read as unknown and must be written as zero.	-	
LST	[11:0]	LSThreshold This field contains an 12-bit value that is compared to the Frame Remaining field (UHCFMR[FR]) prior to initiating a low-speed transaction. The transaction is started only if the Frame Remaining value is greater than or equal to this field. The value is calculated by HCD with the consideration of transmission and setup overhead.	R/W	

5.2.19 USB HcRhDesciptorA Register (UHCRHDA, Address = 0xED40_0048)

The Root Hub Descriptor A register is the first register of two describing the characteristics of the root hub. The descriptor length, descriptor type, and hub controller current fields of the Hub Class descriptor are emulated by the HCD. All other fields are located in the Root Hub Descriptor A and Root Hub Descriptor B registers.

Table 8.9-19 UHCRHDA Bit Definitions

UHCRHDA	Bit	Description	R/W	Reset Value
POT PGT	[31:24]	PowerOnToPowerGoodTime The duration that the HCD must wait before accessing a powered on port (in 2-ms units)	R/W	
Reserved	[23:13]	Read as unknown and must be written as zero.	-	
NOCP	[12]	NoOverCurrentProtection This bit describes if over-current protection is supported. 0 = Over-current protection is supported. 1 = Over-current protection is not supported.	R/W	
OCPM	[11]	OverCurrentProtectionMode This bit describes how the over-current status for the root hub ports is reported. At reset, this field reflects the same mode as PowerSwitchingMode. This field is valid only if the No Over Current Protection field is cleared. 0 = Over-current status is reported collectively for all downstream ports (global). 1 = Over-current status is reported on a per-port basis.	R/W	
DT	[10]	DeviceType This bit specifies that the root hub is not a compound	R	

UHCRHDA	Bit	Description	R/W	Reset Value
		device. The root hub is not permitted to be a compound device. This field always reads 0. This bit always reads 0.		
NPS	[9]	NoPowerSwitching This bit specifies if power-switching is supported or if ports are always powered. If this bit is cleared, the PowerSwitchingMode (UHCRHDA[PSM]) bit specifies global or per-port switching (the ports are power-switched). If this bit is set, ports are always powered on if the UHC is powered on. 0 = Ports are power-switched. 1 = Ports are always powered on if the UHC is powered on.	R/W	
PSM	[8]	PowerSwitchingMode This bit specifies how the power switching of the root hub ports is controlled. This field is only valid if the NoPowerSwitching bit UHCRHDA[NPS] is clear. If UHCRHDA[PSM] is set, each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If the Port Power Control Mask bit (UHCRHDB[PPCM]) is set, the port responds only to port power commands (Set/ClearPortPower: to Set Port Power, write 0b1 to UHCRHPS1/2/3[PPS], to Clear Port Power, write 0b1 to UHCRHPS1/2/3[LDA]). If the port mask is cleared, the port is controlled only by the global power switch (to Set/ClearGlobalPower: write 1 to UHCRHDB[LPSC] to Set Global Power, write 1 to UHCRHDB[LPS] to Clear Global Power). If the UHCRHDA[PSM] bit is cleared, all ports are powered at the same time. 0 = All ports are powered at the same time. 1 = Each port is powered individually.	R/W	
NDP	[7:0]	NumberDownstreamPorts The actual number of downstream ports supported by the root hub is equal to NDP + 1. NOTE: Although the USB host controller supports three (3) downstream ports, this field always contains a value of 0x2.	R	

5.2.20 USB HcRhDescriptorB Register (UHCRHDB, Address = 0xED40_004C)

The UHC Root Hub Descriptor B register is the second register of two describing the characteristics of the root hub. These fields are written during initialization to correspond with the system implementation. Only bits 1, 2, 17, and 18 are writable; all other bits always read as 0.

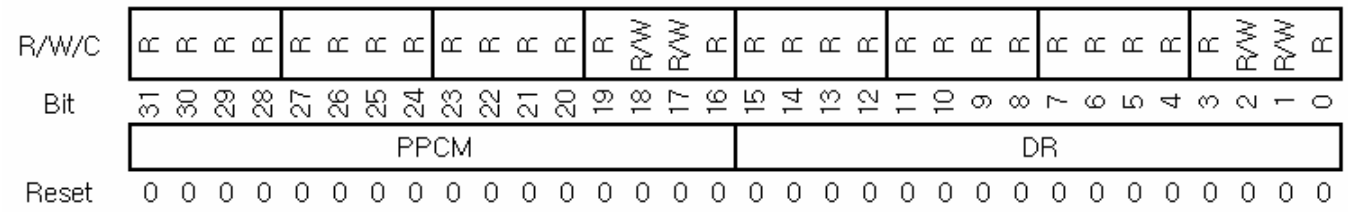


Table 8.9-20 UHCRHDB Bit Definitions

UHCRHDB	Bit	Description	R/W	Reset Value
PPCM	[31:16]	PortPowerControlMask Each bit in this field indicates if a port is affected by a global power control command when power switching mode (UHCRHDA[PSM]) is set. If this bit (UHCRHDB[PPCM]) is set, the port power state is affected only by perport power control (Set/Clear Port Power). If cleared, the port is controlled by the global power switch (Set/Clear Global Power). If the device is configured to global switching mode (UHCRHDA[PSM] = 0), this field is not valid. Bit 17 corresponds to ganged-power mask on port 1, and bit 18 corresponds to ganged-power mask on port 2, and bit 19 corresponds to ganged-power mask on port 3. Bits 31 to 20, and bit 16 are not used. bit 16: Unused, always reads as 0 bit 17: Ganged-power mask on port 1 bit 18: Ganged-power mask on port 2 bits 19 – 31: Unused, always reads as 0	R/W	
DR	[15:0]	DeviceNotRemovable Each bit is dedicated to a port of the root hub. If cleared, the attached device is removable; if set, it is not removable. Bit 1 refers to port 1, bit 2 refers to port 2, and bit 3 refers to port 3. Bits 15 – 4 and bit 0 are unused. If set, this bit forces the corresponding Port Status register current connection status bit (UHCRHPS1/2[CSC]) to 1. bit 0: Unused, always reads as 0 bit 1: 1 if device attached to port 1 is not removable bit 2: 1 if device attached to port 2 is not removable bits 3 – 15: Unused, always reads as 0	R/W	

5.2.21 USB HcRhStatus Register (UHCRHSTAT, Address = 0xED40_0050)

The UHC Root Hub Status register is divided into two parts. The lower word of a word represents the Hub Status field and the upper word represents the Hub Status Change field.

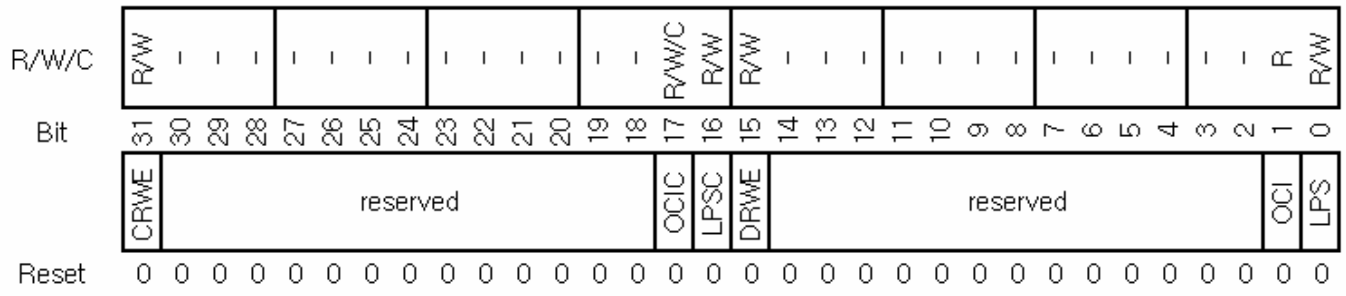


Table 8.9-21 UHCRHSTAT Bit Definitions

UHCRHSTAT	Bit	Description	R/W	Reset Value
CRWE	[31]	ClearRemoteWakeupEnable Writing a 1 clears the device remote wake-up enable (UHCRHS[DRWE]) bit. Writing a 0 has no effect. This bit always reads as 0.	R/W	
Reserved	[30:18]	Read as unknown and must be written as zero.	-	
OCIC	[17]	OverCurrentIndicatorChange This bit is set by hardware if a change has occurred to the OCI field of this register. The HCD clears this bit by writing a 1. Writing a 0 has no effect. 1 = A change has occurred to the OCI field of this register	R/WC	
LPSC	[16]	(read) LocalPowerStatusChange / (write) SetGlobalPower: wake-up The root hub does not support the local power status feature; thus, this bit is always read as 0. In global power mode (UHCRHDA[PSM] is clear), this bit is written to 1 to turn on power to all ports (set Goal Port Power). In per-port power mode, this bit sets PortPowerStatus (UHCRHPS1/2/3[PPS]) only on ports whose port power control mask bit (UHCRHDB[PPCM]) is not set. Writing a 0 has no effect. This bit is always read as 0.	R/W	
DRWE	[15]	(read) DeviceRemoteWakeupEnable / (write) SetRemoteWakeupEnable This bit enables a ConnectStatusChange bit (UHCRHPS1/2/3[CSC]) as a resume event, causing a USBSUSPEND-to-USBRESUME state transition and setting the resume-detected interrupt. If it reads 0, a ConnectStatusChange is not a remote wake-up event. If it is read as a 1, ConnectStatusChange is a remote wake-up event. Writing a 1 sets the device remote wake-up enable (this bit). Writing a 0 has no effect. 0 = ConnectStatusChange is not a remote wake-up	R/W	

UHCRHSTAT	Bit	Description	R/W	Reset Value
		event. 1 = ConnectStatusChange is a remote wake-up event.		
Reserved	[14:2]	Read as unknown and must be written as zero.	-	
OCI	[1]	OverCurrentIndicator This bit reports over-current conditions if the global reporting is implemented. If set, an over-current condition exists. If cleared, all power operations are normal. If per-port over-current protection is implemented, this bit is always 0	R	
LPS	[0]	(read) LocalPowerStatus/ (write) ClearGlobalPower The root hub does not support the local power-status feature; thus, this bit is always read as 0. In global power mode (If UHCRHDA[PSM] is clear), this bit is set to turn off power to all ports (clear PortPowerStatus). In per-port power mode, it clears PortPowerStatus only on ports whose port power control mask bit is not set. Clearing to 0 has no effect.	R/W	

5.2.22 USB HcRhPortStatus 1 Register (UHCRHPSTAT1, Address = 0xED40_0054)

The UHC Root Hub Port Status[2:1] registers control and report USB ports 1, and 2 events on a per-port basis. The lower word of UHCRHPS1/2 reflects the port status, whereas the upper word reflects the status change bits. Some status bits are implemented with special write behavior (refer to Table 8.9-23). If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port-status change must be postponed until the transaction completes. The register organization and individual bit definitions are shown in Table 8.9-23

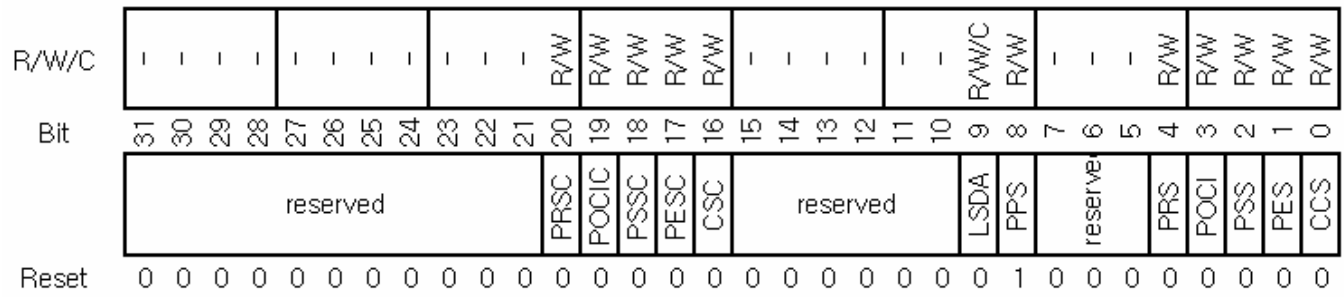


Table 8.9-22 UHCRHPSTAT1 Bit Definitions

UHCRHPSTAT1	Bit	Description	R/W	Reset Value
Reserved	[31:21]	Read as unknown and must be written as zero.	-	
PRSC	[20]	PortResetStatusChange This bit is set at the end of the 10-ms port reset signal. A 1 indicates that the port reset has completed. A 0 indicates that the port reset is not yet complete. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. 0 = Port reset is not complete. 1 = Port reset is complete.	R/W	
POCIC	[19]	PortOverCurrentIndicatorChange This bit is valid only if over-current conditions are reported on a per-port basis. This bit is set if root hub changes the port over-current indicator bit. The HCD writes a 1 to clear this bit. Writing a 0 has no effect 0 = No change in PortOverCurrentIndicator. 1 = PortOverCurrentIndicator has changed.	R/W	
PSSC	[18]	PortSuspendStatusChange This bit is set by the UHC if the full resume sequence is complete. This sequence includes the 20-ms resume pulse, LS EOP, and 3-ms re-synchronization delay. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. This bit is cleared if PortResetStatusChange is set. 0 = Resume is not complete. 1 = Resume completed.	R/W	
PESC	[17]	PortEnableStatusChange This bit is set if events (such as over-current condition, disconnect, switched-off power, or operational bus error-babble), cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a 1 to clear this bit. Writing a 0 has no effect 0 = No change in PortEnableStatus. 1 = Change in PortEnableStatus.	R/W	
CSC	[16]	ConnectStatusChange This bit is set by hardware whenever connect or disconnect event occurs. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared when a set-port-reset, set-port-enable, or set-port-suspend write occurs, this bit is set to force the driver to reevaluate the connection status since these writes must not occur if the port is disconnected. 0 = No change in CurrentConnectStatus. 1 = Change in CurrentConnectStatus. NOTE: If the device removable bit in the Root Hub	R/W	

UHCRHPSTAT1	Bit	Description	R/W	Reset Value
		Descriptor B register is set (indicating that this device is not removable), then the CSC bit is set only after a root hub reset to inform the system that the device is attached.		
Reserved	[15:10]	Read as unknown and must be written as zero.	-	
LSDA	[9]	(read) LowSpeedDeviceAttached / (write) ClearPortPower If read, this bit indicates the speed of the device attached to this port. If set, a low-speed device is attached to this port. If clear, a fullspeed device is attached to this port. This field is valid if the CurrentConnectStatus is set. If the HCD writes a 1 to this bit, the controller clears the PortPowerStatus bit. Writing a 0 has no effect. 0 = Full-speed device attached. 1 = Low-speed device attached.	R/WC	
PPS	[8]	(read) PortPowerStatus / (write) SetPortPower This bit reflects the PortPowerStatus, regardless of the type of powerswitching implemented. This bit is cleared if an over-current condition is detected. HCD sets this bit by writing Set Port Power (writing a 1 to this bit, UHCRHPS1/2/3[PPS]) or Set Global Power (UHCRHS[LPSC]). HCD clears this bit by writing Clear Port Power (writing a 1 to UHCRHPS1/2/3[LSDA]) or Clear Global Power (UHCRHS[LPS]). 0 = Port power is off. 1 = Port power is on. The HCD writes a 0b1 to set the PortPowerStatus bit. Writing 0b0 has no effect	R/W	
Reserved	[7:5]	Read as unknown and must be written as zero.	-	
PRS	[4]	(read) PortResetStatus / (write) SetPortReset If this bit is set by an HCD write to set port reset, port reset signaling is asserted. If reset is complete, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared. The HCD sets the port reset signaling by writing a 1 to this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set port-reset status, but instead sets connect-status-change. This informs the driver that it attempted to reset a disconnected port. 0 = Port reset signal is not active. 1 = Port reset signal is active.	R/W	
POCI	[3]	(read) PortOverCurrentIndicator / (write) ClearSuspendStatus If read, this bit is valid only when the root hub is configured in such a way that over-current conditions are reported on a per-port basis. If per-port over-current	R/W	

UHCRHPSTAT 1	Bit	Description	R/W	Reset Value
		reporting is not supported, this bit is forced to 0. If this bit is cleared, all power operations are normal for this port. If set, an over-current condition exists on this port. This bit always reflects the over-current input signal. The HCD writes a 1 to this bit to initiate a resume. Writing a 0 has no effect. A resume is initiated only if PortSuspendStatus is set. 0 = No over-current condition. 1 = Over-current condition detected.		
PSS	[2]	(read) PortSuspendStatus / (write) SetPortSuspend This bit indicates if the port is suspended or in the resume sequence (1). If it is a 0, the port is not suspended or in the resume sequence. It is set by a set suspend state write and cleared if PortSuspendStatus change is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared if PortResetStatus change is set at the end of the port reset, or if the UHC is placed in the USBRESUME state. If an upstream resume is in progress, it must propagate to the HC. The HCD sets the PortSuspendStatus bit by writing a 1 to this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port. 0 = Port is not suspended. 1 = Port is suspended.	R/W	
PES	[1]	(read) PortEnableStatus / (write) SetPortEnable This bit indicates whether the port is enabled (1) or disabled (0). The root hub clears this bit if an over-current condition, disconnect event, switched-off power, or operational bus error, such as babble, is detected. This change also causes the port enabled status change bit (UHCRHPS1/2/3[PESC] to be set. The HCD sets the PortEnableStatus bit, by writing a 1 to it, and clears this bit by writing a 1 to clear port enable (UHCRHPS1/2/3[CCS], bit 0 of this register). If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected port. Writing a 0 to the PortEnableStatus bit has no effect. This bit is also set, if not already, at the completion of a port reset when reset status change (UHCRHPS1/2/3[PRSC]) is set or port suspend if suspend status change (UHCRHPS1/2/3[PSSC]) is set	R/W	

UHCRHPSTAT 1	Bit	Description	R/W	Reset Value
		0 = Port is disabled. 1 = Port is enabled.		
CCS	[0]	(read) CurrentConnectStatus / (write) ClearPortEnable This bit reflects the current state of the downstream port. If a device is connected, this bit reads as 1, and if no device is connected, it is 0. The HCD writes a 1 to this bit to clear the PortEnableStatus bit. Writing a 0 has no effect. The CurrentConnectStatus is not affected by any write. NOTE: This bit is always read as 1 if the attached device is non-removable This bit reflects the current state of the downstream port. 0 = No device connected. 1 = Device connected.	R/W	

5.2.23 USB HcRhPortStatus 2 Register (UHCRHPSTAT2, Address = 0xED40_0058)

The UHC Root Hub Port Status[2:1] registers control and report USB ports 1 and 2 events on a per-port basis. The lower word of UHCRHPS1/2 reflects the port status, whereas the upper word reflects the status change bits. Some status bits are implemented with special write behavior (see Table 8.9-23). If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port-status change must be postponed until the transaction completes. The register organization and individual bit definitions are shown in Table 8.9-23.

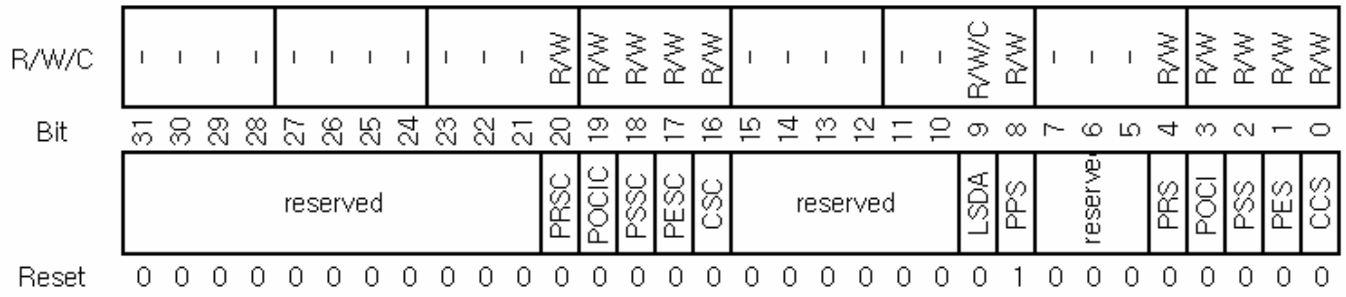


Table 8.9-23 UHCRHPSTAT2 Bit Definitions

UHCRHPSTAT2	Bit	Description	R/W	Reset Value
Reserved	[31:21]	Read as unknown and must be written as zero.	-	
PRSC	[20]	PortResetStatusChange This bit is set at the end of the 10-ms port reset signal. A 1 indicates that the port reset is complete. A 0 indicates that the port reset is not yet complete. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. 0 = Port reset is not complete. 1 = Port reset is complete.	R/W	
POCIC	[19]	PortOverCurrentIndicatorChange This bit is valid only if over-current conditions are reported on a per-port basis. This bit is set if root hub changes the port over-current indicator bit. The HCD writes a 1 to clear this bit. Writing a 0 has no effect 0 = No change in PortOverCurrentIndicator. 1 = PortOverCurrentIndicator has changed.	R/W	
PSSC	[18]	PortSuspendStatusChange This bit is set by the UHC if the full resume sequence has been complete. This sequence includes the 20-ms resume pulse, LS EOP, and 3-ms re-synchronization delay. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. This bit is also cleared if PortResetStatusChange is set. 0 = Resume is not complete. 1 = Resume complete.	R/W	
PESC	[17]	PortEnableStatusChange This bit is set if events (such as over-current condition, disconnect, switched-off power, or operational bus	R/W	

UHCRHPSTAT2	Bit	Description	R/W	Reset Value
		error-babble), cause the PortEnableStatus bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a 1 to clear this bit. Writing a 0 has no effect 0 = No change in PortEnableStatus. 0 = Change in PortEnableStatus.		
CSC	[16]	ConnectStatusChange This bit is set by hardware whenever connect or disconnect event occurs. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared when a set-port-reset, set-port-enable, or set-port-suspend write occurs, this bit is set to force the driver to reevaluate the connection status since these writes must not occur if the port is disconnected. 0 = No change in CurrentConnectStatus. 1 = Change in CurrentConnectStatus. NOTE: If the device removable bit in the Root Hub Descriptor B register is set (indicating that this device is not removable), then the CSC bit is set only after a root hub reset to inform the system that the device is attached.	R/W	
Reserved	[15:10]	Read as unknown and must be written as zero.	-	
LSDA	[9]	(read) LowSpeedDeviceAttached / (write) ClearPortPower If read, this bit indicates the speed of the device attached to this port. If set, a low-speed device is attached to this port. If clear, a fullspeed device is attached to this port. This field is valid only if the CurrentConnectStatus is set. If the HCD writes a 1 to this bit, the controller clears the PortPowerStatus bit. Writing a 0 has no effect. 0 = Full-speed device attached. 1 = Low-speed device attached.	R/WC	
PPS	[8]	(read) PortPowerStatus / (write) SetPortPower This bit reflects the PortPowerStatus, regardless of the type of powerswitching implemented. This bit is cleared if an over-current condition is detected. HCD sets this bit by writing Set Port Power (writing a 1 to this bit, UHCRHPS1/2/3[PPS]) or Set Global Power (UHCRHS[LPSC]). HCD clears this bit by writing Clear Port Power (writing a 1 to UHCRHPS1/2/3[LSDA]) or Clear Global Power (UHCRHS[LPS]). 0 = Port power is off. 1 = Port power is on. The HCD writes a 0b1 to set the PortPowerStatus bit. Writing 0b0 has no effect	R/W	
Reserved	[7:5]	Read as unknown and must be written as zero.	-	

UHCRHPSTAT2	Bit	Description	R/W	Reset Value
PRS	[4]	(read) PortResetStatus / (write) SetPortReset If this bit is set by an HCD write to set port reset, port reset signaling is asserted. If reset is completed, this bit is cleared when PortResetStatusChange is set. This bit cannot be set if CurrentConnectStatus is cleared. The HCD sets the port reset signaling by writing a 1 to this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set port-reset status, but instead sets connect-status-change. This informs the driver that it attempted to reset a disconnected port. 0 = Port reset signal is not active. 1 = Port reset signal is active.	R/W	
POCI	[3]	(read) PortOverCurrentIndicator / (write) ClearSuspendStatus If read, this bit is valid only when the root hub is configured in such a way that over-current conditions are reported on a per-port basis. If per-port over-current reporting is not supported, this bit is forced to 0. If this bit is cleared, all power operations are normal for this port. If set, an over-current condition exists on this port. This bit always reflects the over-current input signal. The HCD writes a 1 to this bit to initiate a resume. Writing a 0 has no effect. A resume is initiated only if PortSuspendStatus is set. 0 = No over-current condition. 1 = Over-current condition detected.	R/W	
PSS	[2]	(read) PortSuspendStatus / (write) SetPortSuspend This bit indicates if the port is suspended or in the resume sequence (1). If it is a 0, the port is not suspended or in the resume sequence. It is set by a set suspend state write and cleared if PortSuspendStatus change is set at the end of the resume interval. This bit cannot be set if CurrentConnectStatus is cleared. This bit is also cleared if PortResetStatus change is set at the end of the port reset, or if the UHC is placed in the USBRESUME state. If an upstream resume is in progress, it must propagate to the HC. The HCD sets the PortSuspendStatus bit by writing a 1 to this bit. Writing a 0 has no effect. If CurrentConnectStatus is cleared, this write does not set PortSuspendStatus; instead it sets ConnectStatusChange. This informs the driver that it attempted to suspend a disconnected port. 0 = Port is not suspended. 1 = Port is suspended.	R/W	
PES	[1]	(read) PortEnableStatus / (write) SetPortEnable This bit indicates whether the port is enabled (1) or disabled (0). The root hub clears this bit if an over-current condition, disconnect event, switched-off power, or operational bus error, such as babble, is detected.	R/W	

UHCRHPSTAT2	Bit	Description	R/W	Reset Value
		This change also causes the port enabled status change bit (UHCRHPS1/2/3[PESC] to be set. The HCD sets the PortEnableStatus bit, by writing a 1 to it, and clears this bit by writing a 1 to clear port enable (UHCRHPS1/2/3[CCS], bit 0 of this register). If CurrentConnectStatus is cleared, this write does not set PortEnableStatus, but instead sets ConnectStatusChange. This informs the driver that it attempted to enable a disconnected port. Writing a 0 to the PortEnableStatus bit has no effect. This bit is also set, if not already, at the completion of a port reset if reset status change (UHCRHPS1/2/3[PRSC]) is set or port suspend if suspend status change (UHCRHPS1/2/3[PSSC]) is set 0 = Port is disabled. 1 = Port is enabled.		
CCS	[0]	(read) CurrentConnectStatus / (write) ClearPortEnable This bit reflects the current state of the downstream port. If a device is connected, this bit reads as 1, and if no device is connected, it is 0. The HCD writes a 1 to this bit to clear the PortEnableStatus bit. Writing a 0 has no effect. The CurrentConnectStatus is not affected by any write. NOTE: This bit is always read as 1 when the attached device is nonremovable This bit reflects the current state of the downstream port. 0 = No device connected. 1 = Device connected.	R/W	

8.10 USB2.0 HS OTG

1 OVERVIEW

Samsung USB On-The-Go (OTG) is a Dual-Role Device (DRD) controller, which supports both device and host functions. It is fully compliant with the On-The-Go Supplement to the USB 2.0 Specification, Revision 1.0a. It supports high-speed (HS, 480-Mbps), full-speed (FS, 12-Mbps, Device only), and low-speed (LS, 1.5-Mbps, Host only) transfers. HS OTG can be configured as a Host-only or Device-only controller.

2 FEATURE

The USB2.0 HS OTG features include the following:

- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 1.0a)
- Operates in High-Speed (480 Mbps), Full-Speed (12 Mbps, Device only) and Low-Speed (1.5 Mbps, Host only) modes
- Supports UTMI+ Level 3 interface (Revision 1.0)
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports only 32-bit data on the AHB
- 1 Control Endpoint 0 for control transfer
- 15 Device Mode programmable Endpoints
  - ♦ Programmable endpoint type: Bulk, Isochronous, or Interrupt
  - ♦ Programmable IN/ OUT direction
- Supports 16 Host channels
- Supports packet-based, dynamic FIFO memory allocation of 6,144 depths (35-bit width)

3 BLOCK DIAGRAM

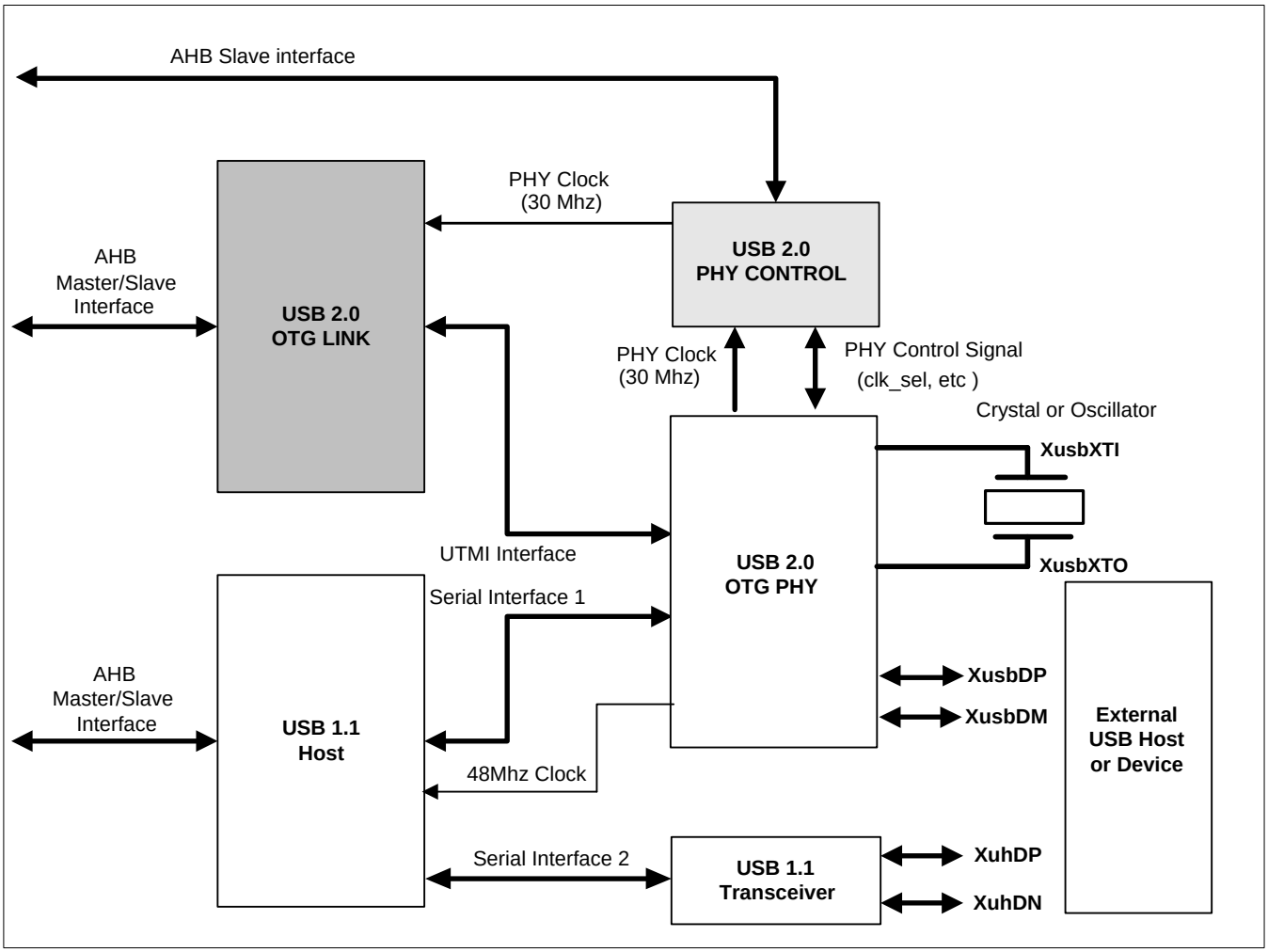


Figure 8.10-1 System Level Block Diagram

USB HS OTG controller is composed of two independent blocks namely, USB 2.0 OTG Link Core and USB 2.0 PHY Control. Each has an AHB Slave, which provides the microcontroller with read and write access to the Control and Status Registers (CSRs). The OTG Link has an AHB Master to enable the link to transfer data on the AHB.

The S5PC100x USB system shown in Figure 8.10-1 is configured as following:

1. USB 1.1 Host 1 Port & USB 2.0 OTG 1 Port
2. USB 1.1 Host 2 Ports

Note: To enable Serial Interface 1 and use 2 ports of Host 1.1, set the OPHYCLK.serial_mode register bit to 1.

4 MODES OF OPERATION

The application operates the Link either in DMA mode or in Slave mode. The application cannot operate the core using DMA and Slave modes simultaneously.

4.1 DMA MODE

USB OTG host uses the AHB Master interface to transmit packet data fetch (AHB to USB) and receive data update (USB to AHB). The AHB master uses the programmed DMA address (HCDMAn register in Host mode and DIEPDMAn/ DOEPDMAn register in Device mode) to access the data buffers.

4.2 SLAVE MODE

USB OTG can operate either in transaction-level operation or in pipelined transaction-level operation. The application handles one data packet at a time per channel / endpoint in transaction-level operations. In pipelined transaction-level operation, the application programs the OTG to perform multiple transactions. The advantage of pipelined operation is that the application is not interrupted on packet basis.

5 POWER MANAGEMENT UNIT SETTING

A register in Power Management Unit has to be set for USB to work appropriately. (**OTHERS = 0xE010_8200)**

OTHERS	Bit	Description	Reset Value
USB_SIG_MASK	[16]	The role of this bit is to bypass or block the signals transferred from USB OTG PHY to internal logic. In order to start USB transaction, This bit is set to 1 and then USB_PHY initialization sequence begins. **Caution: If USB_PHY is not used in your system, this bit is set to 0.** 0 = Mask 1 = Unmask	0

In order to start USB transaction, USB_SIG_MASK is set to 1'b1 and then USB OTG should be initialized. On the other hand, USB_SIG_MASK is set to 1'b0 to prevent the unwanted signal transfer from OTG PHY to internal logic in case USB OTG voltage sources of 1.2V and 3.3V are not supplied.

There are two ways to minimize the current consumption when USB OTG is not used. In case USB OTG voltage sources of 1.2V and 3.3V are provided, USB_SIG_MASK is set to 1'b1 and USB OTG should be initialized and then entered into standby state. In the other case, USB_SIG_MASK is set to 1'b0.

6 MEMORY MAP

6.1 OVERVIEW

The OTG PHY control registers based on address ED30_0000h must be accessed to control and observe the OTG PHY.

The OTG Link Core registers based on address ED20_0000h is classified as follows:

- Core Global Registers
- Host Mode Registers
  - Host Global Registers
  - Host Port CSRs
  - Host Channel-Specific Registers
- Device Mode Registers
  - Device Global Registers
  - Device Endpoint-Specific Registers

The Core Global and Host Port registers is accessed in both Host and Device modes. If the OTG Link is operating in either Device or Host mode, the application must not access registers from the other mode. If an illegal access occurs, a Mode Mismatch interrupt is generated and reflected in the Core Interrupt register. If the core switches from one mode to another, the registers in the new mode of operation must be reprogrammed as they would be after a power-on reset.

6.2 OTG LINK CSR MEMORY MAP

Figure 8.10-2 shows the OTG link CSR address map. Host and Device mode registers occupy different addresses. All registers are implemented in the AHB Clock domain.

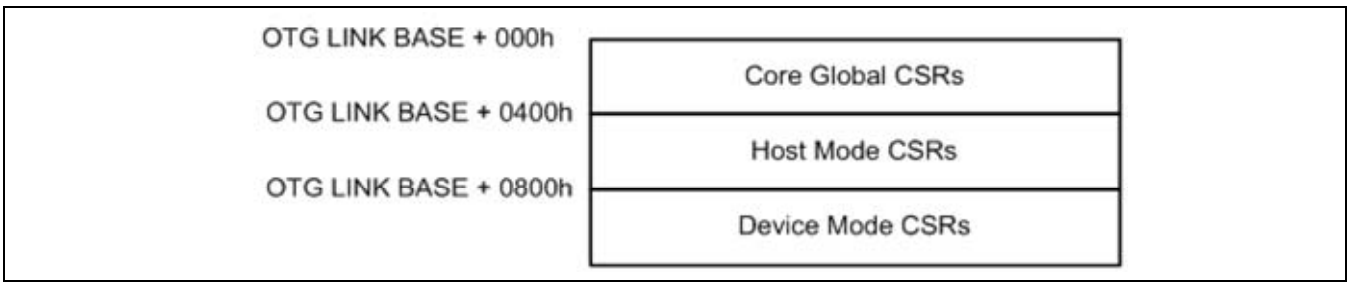


Figure 8.10-2 OTG Link CSR Memory Map

6.3 OTG FIFO ADDRESS MAPPING

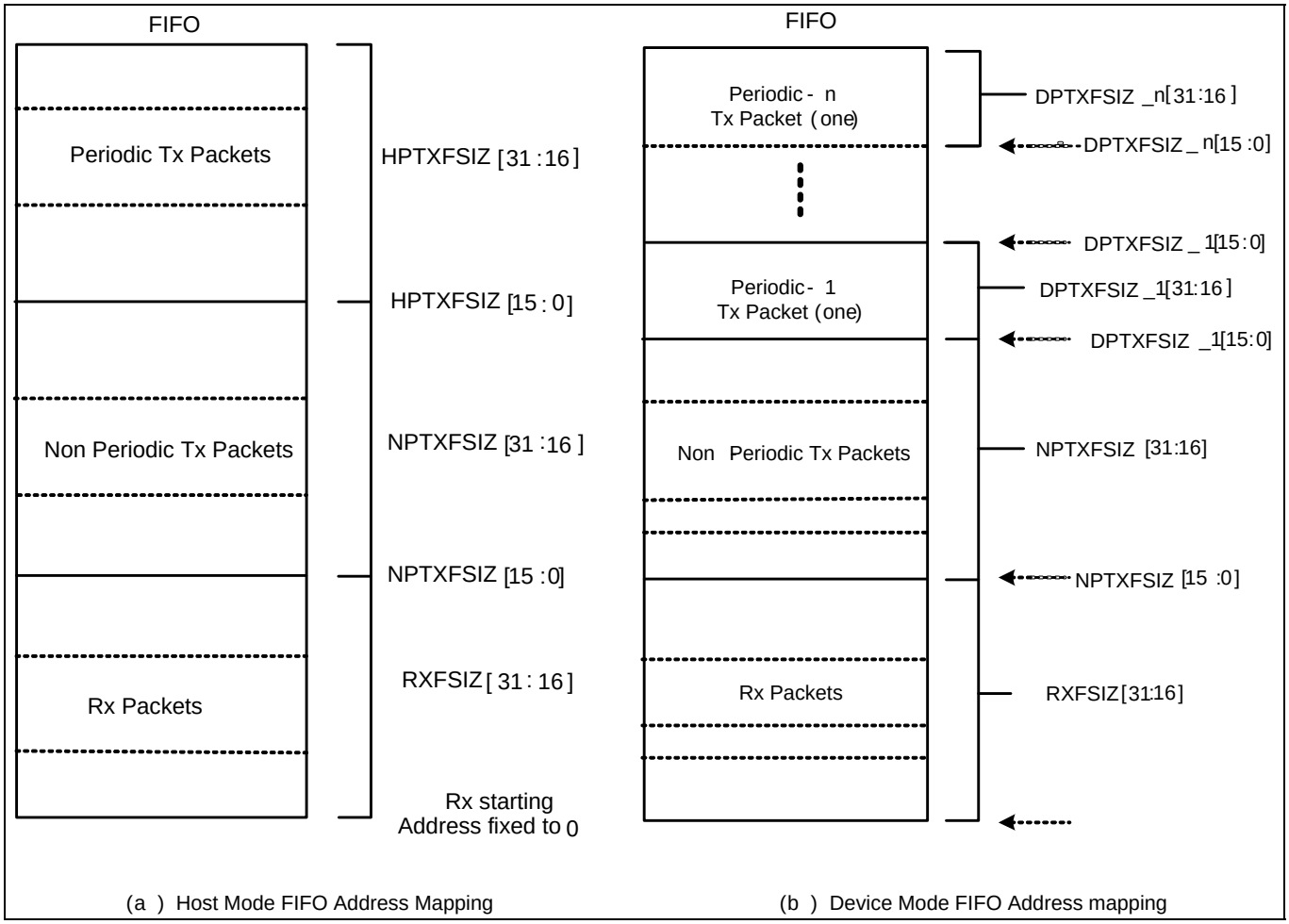


Figure 8.10-3 OTG FIFO Mapping

Figure 8.10-3 shows the OTG FIFO Address Mapping. The following registers must be programmed as follows;

In Host Mode

RXFSIZ[15:0] = OTG_RX_DFIFO_DEPTH
NPTXFSIZ[15:0] = OTG_RX_DFIFO_DEPTH
NPTXFSIZ[31:16] = OTG_TX_NPERIO_DFIFO_DEPTH
HPTXFSIZ[15:0] = RX_DFIFO_DEPTH + TX_NPERIO_DFIFO_DEPTH
HPTXFSIZ[31:16] = TX_PERIOD_DFIFO_DEPTH

In Device Mode

RXFSIZ[15:0] = OTG_RX_DFIFO_DEPTH
NPTXFSIZ[15:0] = OTG_RX_DFIFO_DEPTH
NPTXFSIZ[31:16] = OTG_TX_NPERIO_DFIFO_DEPTH
DPTXFSIZ_1[15:0] = OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH
DPTXFSIZ_1[31:16] = OTG_TX_DPERIO_DFIFO_DEPTH_1
DPTXFSIZ_2[15:0] = DPTXFSIZE_1[15:0] + OTG_TX_DPERIO_DFIFO_DEPTH_1
DPTXFSIZ_2[31:16] = OTG_TX_DPERIO_DFIFO_DEPTH2
…….
DPTXFSIZ_n[15:0] = DPTXFSIZE_n-1[15:0] + OTG_TX_DPERIO_DFIFO_DEPTH_n-1
DPTXFSIZ_n[31:16] = OTG_TX_DPERIO_DFIFO_DEPTHn

6.4 APPLICATION ACCESS TO THE CSRS

The Access column of each register description that follows specifies how the application and the core access the register fields of the CSRs. The following conventions are used.

Read Only	R	The application has permission to read the Register field. Writes to read-only fields have no effect.
Write Only	W	The application has permission to write in the Register field.
READ and Write	R/ W	The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0.
Read, Write, and Self Clear	R/ W_SC	Register field is read and written by the application (Read and Write), and is cleared to 1'b0 by the core (Self Clear). The conditions under which the core clears this field are explained in detail in the field's description.
Read, Write, Self Set, and Self Clear	R/ W_SS_SC	Register field is read and written by the application (Read and Write), set to 1'b1 by the core on certain USB events (Self Set), and cleared to 1'b0 by the core (Self Clear).
Read, Self set, and Write Clear	R/ SS_WC	Register field is read by the application (Read), set to 1'b1 by the core on certain internal or USB or AHB event (Self Set), and cleared to 1'b0 by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 has no effect on this field.
Read, Write Set, and Self Clear	R/ WS_SC	Register field is read by the application (Read), set to 1'b1 by the application with a register write of 1'b1 (Write Set), and is cleared to 1'b0 by the core (Self Clear). The application cannot clear this type of field, and a register write of 1'b0 to this bit has no effect on this field.
Read, Self set, and Self Clear or Write Clear	R/ SS_SC_WC	Register field is read by the application (Read), set to 1'b1 by the core on certain internal or USB or AHB events (Self Set), and cleared to 1'b0 either by the core itself (Self Clear) or by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 to this bit has no effect on this field.

7 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
USB_DP	Input/Output	Data Plus Signal from the USB Cable	Xusb DP	Dedicated
USB_DM	Input/Output	Data Minus Signal from the USB Cable	XusbDM	Dedicated
USB_XTI	Input	Crystal Oscillator XI	XusbXTI	Dedicated
USB_XTO	Output	Crystal Oscillator XO	XusbXTO	Dedicated
USB_REXT	Input/Output	Connection to the External 44.2 ohms register	XusbREXT	Dedicated
USB_VBUS(1)	Input/Output	USB 5V Power detection	XusbVBUS	Dedicated
USB_ID	Input/Output	This signal indicates whether the connected plug is mini-A or mini-B. Low : Mini-A connected High: Mini-B connected	XusbID	Dedicated
USB_DRVVBUS	Output	This controller signal enables or disables external charge pump.	XusbDRVVBUS	Dedicated

NOTE 1 : USB2.0 HS OTG module can operate in device mode when the voltage on USB_VBUS is valid. To be valid in device mode, the voltage on USB_VBUS is required to be between 4.75V and 5.25V.
In host mode, OTG makes USB_DRVVBUS low if the voltage on USB_VBUS is lower than the threshold voltage. The setting value in OPHYTUNE[16:14] can change the threshold voltage on USB_VBUS.

A register in GPIO has to be set for USB to work appropriately.

Register	Address	R/W	Description	Reset Value
ETC4DRV	0xE030056C	R/W	Port Group ETC4 Drive strength control Register	0x0000

To use crystal, The 8th and 9th of Port Group ETC4 Control register based on address 0xE030056C is guided to be set differently depending on the crystal input frequency of USB.

9^{th} bit	8^{th} bit	Output	comment
0	0	Driver X1	32KHz ~ 1MHz
0	1	Driver X2	1MHz ~ 15MHz
1	0	Driver X4	15MHz ~ 30MHz
1	1	Driver X8	30MHz ~ 50MHz

8 REGISTER DESCRIPTION

8.1 REGISTER OVERVIEW

Table 8.10-1 Register summary of HS OTG Controller

Register	Offset	R/W	Description	Reset Value
OTG PHY CONTROL REGISTERS				
OPHYPWR	0xED30_0000	R/W	OTG PHY Power Control Register	0x0000_0019
OPHYCLK	0xED30_0004	R/W	OTG PHY Clock Control Register	0x0000_0000
ORSTCON	0xED30_0008	R/W	OTG Reset Control Register	0x0000_0001
OPHYTUNE	0xED30_0020	R/W	OTG PHY Tuning Register	0x0027_1B93
OTG LINK CORE REGISTERS				
Core Global Registers				
GOTGCTL	0xED20_0000	R/W	OTG Control and Status Register	0x0001_0000
GOTGINT	0xED20_0004	R/W	OTG Interrupt Register	0x0000_0000
GAHBCFG	0xED20_0008	R/W	Core AHB Configuration Register	0x0000_0000
GUSBCFG	0xED20_000C	R/W	Core USB Configuration Register	0x0000_1400
GRSTCTL	0xED20_0010	R/W	Core Reset Register	0x8000_0000
GINTSTS	0xED20_0014	R/W	Core Interrupt Register	0x0400_1020
GINTMSK	0xED20_0018	R/W	Core Interrupt Mask Register	0x0000_0000
GRXSTSR	0xED20_001C	R	Receive Status Debug Read Register	-
GRXSTSP	0xED20_0020	R	Receive Status Read/Pop Register	-
GRXFSIZ	0xED20_0024	R/W	Receive FIFO Size Register	0x0000_1800
GNPTXFSIZ	0xED20_0028	R/W	Non-Periodic Transmit FIFO Size Register	0x1800_1800
GNPTXSTS	0xED20_002C	R	Non-Periodic Transmit FIFO/Queue Status Register	0x0008_1800
HPTXFSIZ	0xED20_0100	R/W	Host Periodic Transmit FIFO Size Register	0x0300_5A00
DPTXFSIZ1	0xED20_0104	R/W	Device Periodic Transmit FIFO-1 Size Register	0x0300_1000
DPTXFSIZ2	0xED20_0108	R/W	Device Periodic Transmit FIFO-2 Size Register	0x0300_3300
DPTXFSIZ3	0xED20_010C	R/W	Device Periodic Transmit FIFO-3 Size Register	0x0300_3600
DPTXFSIZ4	0xED20_0110	R/W	Device Periodic Transmit FIFO-4 Size Register	0x0300_3900
DPTXFSIZ5	0xED20_0114	R/W	Device Periodic Transmit FIFO-5 Size Register	0x0300_3C00
DPTXFSIZ6	0xED20_0118	R/W	Device Periodic Transmit FIFO-6 Size Register	0x0300_3F00
DPTXFSIZ7	0xED20_011C	R/W	Device Periodic Transmit FIFO-7 Size Register	0x0300_4200
DPTXFSIZ8	0xED20_0120	R/W	Device Periodic Transmit FIFO-8 Size Register	0x0300_4500
DPTXFSIZ9	0xED20_0124	R/W	Device Periodic Transmit FIFO-9 Size Register	0x0300_4800
DPTXFSIZ10	0xED20_0128	R/W	Device Periodic Transmit FIFO-10 Size Register	0x0300_4B00
DPTXFSIZ11	0xED20_012C	R/W	Device Periodic Transmit FIFO-11 Size Register	0x0300_4E00
DPTXFSIZ12	0xED20_0130	R/W	Device Periodic Transmit FIFO-12 Size Register	0x0300_5100

Register	Offset	R/W	Description	Reset Value
DPTXFSIZ13	0xED20_0134	R/W	Device Periodic Transmit FIFO-13 Size Register	0x0300_5400
DPTXFSIZ14	0xED20_0138	R/W	Device Periodic Transmit FIFO-14 Size Register	0x0300_5700
DPTXFSIZ15	0xED20_013C	R/W	Device Periodic Transmit FIFO-15 Size Register	0x0300_5A00
Host Mode Registers				
Host Global Registers				
HCFG	0xED20_0400	R/W	Host Configuration Register	0x0020_0000
HFIR	0xED20_0404	R/W	Host Frame Interval Register	0x0000_17D7
HFNUM	0xED20_0408	R	Host Frame Number/Frame Time Remaining Register	0x0000_0000
HPTXSTS	0xED20_0410	R	Host Periodic Transmit FIFO/Queue Status Register	0x0008_0100
HAINT	0xED20_0414	R	Host All Channels Interrupt Register	0x0000_0000
HAINTMSK	0xED20_0418	R/W	Host All Channels Interrupt Mask Register	0x0000_0000
Host Port Control and Status Registers				
HPRT	0xED20_0440	R/W	Host Port Control and Status Register	0x0000_0000
Host Channel-Specific Registers				
HCCHAR0	0xED20_0500	R/W	Host Channel 0 Characteristics Register	0x0000_0000
HCSPLT0	0xED20_0504	R/W	Host Channel 0 Spilt Control Register	0x0000_0000
HCINT0	0xED20_0508	R/W	Host Channel 0 Interrupt Register	0x0000_0000
HCINTMSK0	0xED20_050C	R/W	Host Channel 0 Interrupt Mask Register	0x0000_0000
HCTSIZ0	0xED20_0510	R/W	Host Channel 0 Transfer Size Register	0x0000_0000
HCDMA0	0xED20_0514	R/W	Host Channel 0 DMA Address Register	0x0000_0000
HCCHAR1	0xED20_0520	R/W	Host Channel 1 Characteristics Register	0x0000_0000
HCSPLT1	0xED20_0524	R/W	Host Channel 1 Spilt Control Register	0x0000_0000
HCINT1	0xED20_0528	R/W	Host Channel 1 Interrupt Register	0x0000_0000
HCINTMSK1	0xED20_052C	R/W	Host Channel 1 Interrupt Mask Register	0x0000_0000
HCTSIZ1	0xED20_0530	R/W	Host Channel 1 Transfer Size Register	0x0000_0000
HCDMA1	0xED20_0534	R/W	Host Channel 1 DMA Address Register	0x0000_0000
HCCHAR2	0xED20_0540	R/W	Host Channel 2 Characteristics Register	0x0000_0000
HCSPLT2	0xED20_0544	R/W	Host Channel 2 Spilt Control Register	0x0000_0000
HCINT2	0xED20_0548	R/W	Host Channel 2 Interrupt Register	0x0000_0000
HCINTMSK2	0xED20_054C	R/W	Host Channel 2 Interrupt Mask Register	0x0000_0000
HCTSIZ2	0xED20_0550	R/W	Host Channel 2 Transfer Size Register	0x0000_0000
HCDMA2	0xED20_0554	R/W	Host Channel 2 DMA Address Register	0x0000_0000
HCCHAR3	0xED20_0560	R/W	Host Channel 3 Characteristics Register	0x0000_0000
HCSPLT3	0xED20_0564	R/W	Host Channel 3 Spilt Control Register	0x0000_0000
HCINT3	0xED20_0568	R/W	Host Channel 3 Interrupt Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
HCINTMSK3	0xED20_056C	R/W	Host Channel 3 Interrupt Mask Register	0x0000_0000
HCTSIZ3	0xED20_0570	R/W	Host Channel 3 Transfer Size Register	0x0000_0000
HCDMA3	0xED20_0574	R/W	Host Channel 3 DMA Address Register	0x0000_0000
HCCHAR4	0xED20_0580	R/W	Host Channel 4 Characteristics Register	0x0000_0000
HCSPLT4	0xED20_0584	R/W	Host Channel 4 Spilt Control Register	0x0000_0000
HCINT4	0xED20_0588	R/W	Host Channel 4 Interrupt Register	0x0000_0000
HCINTMSK4	0xED20_058C	R/W	Host Channel 4 Interrupt Mask Register	0x0000_0000
HCTSIZ4	0xED20_0580	R/W	Host Channel 4 Transfer Size Register	0x0000_0000
HCDMA4	0xED20_0584	R/W	Host Channel 4 DMA Address Register	0x0000_0000
HCCHAR5	0xED20_05A0	R/W	Host Channel 5 Characteristics Register	0x0000_0000
HCSPLT5	0xED20_05A4	R/W	Host Channel 5 Spilt Control Register	0x0000_0000
HCINT5	0xED20_05A8	R/W	Host Channel 5 Interrupt Register	0x0000_0000
HCINTMSK5	0xED20_05AC	R/W	Host Channel 5 Interrupt Mask Register	0x0000_0000
HCTSIZ5	0xED20_05B0	R/W	Host Channel 5 Transfer Size Register	0x0000_0000
HCDMA5	0xED20_05B4	R/W	Host Channel 5 DMA Address Register	0x0000_0000
HCCHAR6	0xED20_05C0	R/W	Host Channel 6 Characteristics Register	0x0000_0000
HCSPLT6	0xED20_05C4	R/W	Host Channel 6 Spilt Control Register	0x0000_0000
HCINT6	0xED20_05C8	R/W	Host Channel 6 Interrupt Register	0x0000_0000
HCINTMSK6	0xED20_05CC	R/W	Host Channel 6 Interrupt Mask Register	0x0000_0000
HCTSIZ6	0xED20_05D0	R/W	Host Channel 6 Transfer Size Register	0x0000_0000
HCDMA6	0xED20_05D4	R/W	Host Channel 6 DMA Address Register	0x0000_0000
HCCHAR7	0xED20_05E0	R/W	Host Channel 7 Characteristics Register	0x0000_0000
HCSPLT7	0xED20_05E4	R/W	Host Channel 7 Spilt Control Register	0x0000_0000
HCINT7	0xED20_05E8	R/W	Host Channel 7 Interrupt Register	0x0000_0000
HCINTMSK7	0xED20_05EC	R/W	Host Channel 7 Interrupt Mask Register	0x0000_0000
HCTSIZ7	0xED20_05F0	R/W	Host Channel 7 Transfer Size Register	0x0000_0000
HCDMA7	0xED20_05F4	R/W	Host Channel 7 DMA Address Register	0x0000_0000
HCCHAR8	0xED20_0600	R/W	Host Channel 8 Characteristics Register	0x0000_0000
HCSPLT8	0xED20_0604	R/W	Host Channel 8 Spilt Control Register	0x0000_0000
HCINT8	0xED20_0608	R/W	Host Channel 8 Interrupt Register	0x0000_0000
HCINTMSK8	0xED20_060C	R/W	Host Channel 8 Interrupt Mask Register	0x0000_0000
HCTSIZ8	0xED20_0610	R/W	Host Channel 8 Transfer Size Register	0x0000_0000
HCDMA8	0xED20_0614	R/W	Host Channel 8 DMA Address Register	0x0000_0000
HCCHAR9	0xED20_0620	R/W	Host Channel 9 Characteristics Register	0x0000_0000
HCSPLT9	0xED20_0624	R/W	Host Channel 9 Spilt Control Register	0x0000_0000
HCINT9	0xED20_0628	R/W	Host Channel 9 Interrupt Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
HCINTMSK9	0xED20_062C	R/W	Host Channel 9 Interrupt Mask Register	0x0000_0000
HCTSIZ9	0xED20_0630	R/W	Host Channel 9 Transfer Size Register	0x0000_0000
HCDMA9	0xED20_0634	R/W	Host Channel 9 DMA Address Register	0x0000_0000
HCCHAR10	0xED20_0640	R/W	Host Channel 10 Characteristics Register	0x0000_0000
HCSPLT10	0xED20_0644	R/W	Host Channel 10 Spilt Control Register	0x0000_0000
HCINT10	0xED20_0648	R/W	Host Channel 10 Interrupt Register	0x0000_0000
HCINTMSK10	0xED20_064C	R/W	Host Channel 10 Interrupt Mask Register	0x0000_0000
HCTSIZ10	0xED20_0650	R/W	Host Channel 10 Transfer Size Register	0x0000_0000
HCDMA10	0xED20_0654	R/W	Host Channel 10 DMA Address Register	0x0000_0000
HCCHAR11	0xED20_0660	R/W	Host Channel 11 Characteristics Register	0x0000_0000
HCSPLT11	0xED20_0664	R/W	Host Channel 11 Spilt Control Register	0x0000_0000
HCINT11	0xED20_0668	R/W	Host Channel 11 Interrupt Register	0x0000_0000
HCINTMSK11	0xED20_066C	R/W	Host Channel 11 Interrupt Mask Register	0x0000_0000
HCTSIZ11	0xED20_0670	R/W	Host Channel 11 Transfer Size Register	0x0000_0000
HCDMA11	0xED20_0674	R/W	Host Channel 11 DMA Address Register	0x0000_0000
HCCHAR12	0xED20_0680	R/W	Host Channel 12 Characteristics Register	0x0000_0000
HCSPLT12	0xED20_0684	R/W	Host Channel 12 Spilt Control Register	0x0000_0000
HCINT12	0xED20_0688	R/W	Host Channel 12 Interrupt Register	0x0000_0000
HCINTMSK12	0xED20_068C	R/W	Host Channel 12 Interrupt Mask Register	0x0000_0000
HCTSIZ12	0xED20_0690	R/W	Host Channel 12 Transfer Size Register	0x0000_0000
HCDMA12	0xED20_0694	R/W	Host Channel 12 DMA Address Register	0x0000_0000
HCCHAR13	0xED20_06A0	R/W	Host Channel 13 Characteristics Register	0x0000_0000
HCSPLT13	0xED20_06A4	R/W	Host Channel 13 Spilt Control Register	0x0000_0000
HCINT13	0xED20_06A8	R/W	Host Channel 13 Interrupt Register	0x0000_0000
HCINTMSK13	0xED20_06AC	R/W	Host Channel 13 Interrupt Mask Register	0x0000_0000
HCTSIZ13	0xED20_06B0	R/W	Host Channel 13 Transfer Size Register	0x0000_0000
HCDMA13	0xED20_06B4	R/W	Host Channel 13 DMA Address Register	0x0000_0000
HCCHAR14	0xED20_06C0	R/W	Host Channel 14 Characteristics Register	0x0000_0000
HCSPLT14	0xED20_06C4	R/W	Host Channel 14 Spilt Control Register	0x0000_0000
HCINT14	0xED20_06C8	R/W	Host Channel 14 Interrupt Register	0x0000_0000
HCINTMSK14	0xED20_06CC	R/W	Host Channel 14 Interrupt Mask Register	0x0000_0000
HCTSIZ14	0xED20_06D0	R/W	Host Channel 14 Transfer Size Register	0x0000_0000
HCDMA14	0xED20_06D4	R/W	Host Channel 14 DMA Address Register	0x0000_0000
HCCHAR15	0xED20_06E0	R/W	Host Channel 15 Characteristics Register	0x0000_0000
HCSPLT15	0xED20_06E4	R/W	Host Channel 15 Spilt Control Register	0x0000_0000
HCINT15	0xED20_06E8	R/W	Host Channel 15 Interrupt Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
HCINTMSK15	0xED20_06EC	R/W	Host Channel 15 Interrupt Mask Register	0x0000_0000
HCTSIZ15	0xED20_06F0	R/W	Host Channel 15 Transfer Size Register	0x0000_0000
HCDMA15	0xED20_06F4	R/W	Host Channel 15 DMA Address Register	0x0000_0000
Device Mode Registers				
Device Global Registers				
DCFG	0xED20_0800	R/W	Device Configuration Register	0x0020_0000
DCTL	0xED20_0804	R/W	Device Control Register	0x0000_0000
DSTS	0xED20_0808	R	Device Status Register	0x0000_0002
DIEPMSK	0xED20_0810	R/W	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000
DOEPMSK	0xED20_0814	R/W	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000
DAINT	0xED20_0818	R	Device ALL Endpoints Interrupt Register	0x0000_0000
DAINTMSK	0xED20_081C	R/W	Device ALL Endpoints Interrupt Mask Register	0x0000_0000
DTKNQR1	0xED20_0820	R	Device IN Token Sequence Learning Queue Read Register 1	0x0000_0000
DTKNQR2	0xED20_0824	R	Device IN Token Sequence Learning Queue Read Register 2	0x0000_0000
DVBUSDIS	0xED20_0828	R/W	Device VBUS Discharge Time Register	0x0000_17D7
DVBUSPULSE	0xED20_082C	R/W	Device VBUS Pulsing Time Register	0x0000_05B8
DTKNQR3	0xED20_0830	R	Device IN Token Sequence Learning Queue Read Register 3	0x0000_0000
DTKNQR4	0xED20_0834	R	Device IN Token Sequence Learning Queue Read Register 4	0x0000_0000
Device Logical IN Endpoint-Specific Registers				
DIEPCTL0	0xED20_0900	R/W	Device Control IN Endpoint 0 Control Register	0x0000_8000
DIEPINT0	0xED20_0908	R/W	Device IN Endpoint 0 Interrupt Register	0x0000_0000
DIEPTSIZ0	0xED20_0910	R/W	Device IN Endpoint 0 Transfer Size Register	0x0000_0000
DIEPDMA0	0xED20_0914	R/W	Device IN Endpoint 0 DMA Address Register	0x0000_0000
DIEPCTL1	0xED20_0920	R/W	Device Control IN Endpoint 1 Control Register	0x0000_0000
DIEPINT1	0xED20_0928	R/W	Device IN Endpoint 1 Interrupt Register	0x0000_0080
DIEPTSIZ1	0xED20_0930	R/W	Device IN Endpoint 1 Transfer Size Register	0x0000_0000
DIEPDMA1	0xED20_0934	R/W	Device IN Endpoint 1 DMA Address Register	0x0000_0000
DIEPCTL2	0xED20_0940	R/W	Device Control IN Endpoint 2 Control Register	0x0000_0000
DIEPINT2	0xED20_0948	R/W	Device IN Endpoint 2 Interrupt Register	0x0000_0080
DIEPTSIZ2	0xED20_0950	R/W	Device IN Endpoint 2 Transfer Size Register	0x0000_0000
DIEPDMA2	0xED20_0954	R/W	Device IN Endpoint 2 DMA Address Register	0x0000_0000
DIEPCTL3	0xED20_0960	R/W	Device Control IN Endpoint 3 Control Register	0x0000_0000
DIEPINT3	0xED20_0968	R/W	Device IN Endpoint 3 Interrupt Register	0x0000_0080

Register	Offset	R/W	Description	Reset Value
DIEPTSIZ3	0xED20_0970	R/W	Device IN Endpoint 3 Transfer Size Register	0x0000_0000
DIEPDMA3	0xED20_0974	R/W	Device IN Endpoint 3 DMA Address Register	0x0000_0000
DIEPCTL4	0xED20_0980	R/W	Device Control IN Endpoint 0 Control Register	0x0000_0000
DIEPINT4	0xED20_0988	R/W	Device IN Endpoint 4 Interrupt Register	0x0000_0080
DIEPTSIZ4	0xED20_0990	R/W	Device IN Endpoint 4 Transfer Size Register	0x0000_0000
DIEPDMA4	0xED20_0994	R/W	Device IN Endpoint 4 DMA Address Register	0x0000_0000
DIEPCTL5	0xED20_09A0	R/W	Device Control IN Endpoint 5 Control Register	0x0000_0000
DIEPINT5	0xED20_09A8	R/W	Device IN Endpoint 5 Interrupt Register	0x0000_0080
DIEPTSIZ5	0xED20_09B0	R/W	Device IN Endpoint 5 Transfer Size Register	0x0000_0000
DIEPDMA5	0xED20_09B4	R/W	Device IN Endpoint 5 DMA Address Register	0x0000_0000
DIEPCTL6	0xED20_09C0	R/W	Device Control IN Endpoint 6 Control Register	0x0000_0000
DIEPINT6	0xED20_09C8	R/W	Device IN Endpoint 6 Interrupt Register	0x0000_0080
DIEPTSIZ6	0xED20_09D0	R/W	Device IN Endpoint 6 Transfer Size Register	0x0000_0000
DIEPDMA6	0xED20_09D4	R/W	Device IN Endpoint 6 DMA Address Register	0x0000_0000
DIEPCTL7	0xED20_09E0	R/W	Device Control IN Endpoint 7 Control Register	0x0000_0000
DIEPINT7	0xED20_09E8	R/W	Device IN Endpoint 7 Interrupt Register	0x0000_0080
DIEPTSIZ7	0xED20_09F0	R/W	Device IN Endpoint 7 Transfer Size Register	0x0000_0000
DIEPDMA7	0xED20_09F4	R/W	Device IN Endpoint 7 DMA Address Register	0x0000_0000
DIEPCTL8	0xED20_0A00	R/W	Device Control IN Endpoint 8 Control Register	0x0000_0000
DIEPINT8	0xED20_0A08	R/W	Device IN Endpoint 8 Interrupt Register	0x0000_0080
DIEPTSIZ8	0xED20_0A10	R/W	Device IN Endpoint 8 Transfer Size Register	0x0000_0000
DIEPDMA8	0xED20_0A14	R/W	Device IN Endpoint 8 DMA Address Register	0x0000_0000
DIEPCTL9	0xED20_0A20	R/W	Device Control IN Endpoint 9 Control Register	0x0000_0000
DIEPINT9	0xED20_0A28	R/W	Device IN Endpoint 9 Interrupt Register	0x0000_0080
DIEPTSIZ9	0xED20_0A30	R/W	Device IN Endpoint 9 Transfer Size Register	0x0000_0000
DIEPDMA9	0xED20_0A34	R/W	Device IN Endpoint 9 DMA Address Register	0x0000_0000
DIEPCTL10	0xED20_0A40	R/W	Device Control IN Endpoint 10 Control Register	0x0000_0000
DIEPINT10	0xED20_0A48	R/W	Device IN Endpoint 10 Interrupt Register	0x0000_0080
DIEPTSIZ10	0xED20_0A50	R/W	Device IN Endpoint 10 Transfer Size Register	0x0000_0000
DIEPDMA10	0xED20_0A54	R/W	Device IN Endpoint 10 DMA Address Register	0x0000_0000
DIEPCTL11	0xED20_0A60	R/W	Device Control IN Endpoint 11 Control Register	0x0000_0000
DIEPINT11	0xED20_0A68	R/W	Device IN Endpoint 11 Interrupt Register	0x0000_0080
DIEPTSIZ11	0xED20_0A70	R/W	Device IN Endpoint 11 Transfer Size Register	0x0000_0000
DIEPDMA11	0xED20_0A74	R/W	Device IN Endpoint 11 DMA Address Register	0x0000_0000
DIEPCTL12	0xED20_0A80	R/W	Device Control IN Endpoint 12 Control Register	0x0000_0000
DIEPINT12	0xED20_0A88	R/W	Device IN Endpoint 12 Interrupt Register	0x0000_0080

Register	Offset	R/W	Description	Reset Value
DIEPTSIZ12	0xED20_0A90	R/W	Device IN Endpoint 12 Transfer Size Register	0x0000_0000
DIEPDMA12	0xED20_0A94	R/W	Device IN Endpoint 12 DMA Address Register	0x0000_0000
DIEPCTL13	0xED20_0AA0	R/W	Device Control IN Endpoint 13 Control Register	0x0000_0000
DIEPINT13	0xED20_0AA8	R/W	Device IN Endpoint 13 Interrupt Register	0x0000_0080
DIEPTSIZ13	0xED20_0AB0	R/W	Device IN Endpoint 13 Transfer Size Register	0x0000_0000
DIEPDMA13	0xED20_0AB4	R/W	Device IN Endpoint 13 DMA Address Register	0x0000_0000
DIEPCTL14	0xED20_0AC0	R/W	Device Control IN Endpoint 14 Control Register	0x0000_0000
DIEPINT14	0xED20_0AC8	R/W	Device IN Endpoint 14 Interrupt Register	0x0000_0080
DIEPTSIZ14	0xED20_0AD0	R/W	Device IN Endpoint 14 Transfer Size Register	0x0000_0000
DIEPDMA14	0xED20_0AD4	R/W	Device IN Endpoint 14 DMA Address Register	0x0000_0000
DIEPCTL15	0xED20_0AE0	R/W	Device Control IN Endpoint 15 Control Register	0x0000_0000
DIEPINT15	0xED20_0AE8	R/W	Device IN Endpoint 15 Interrupt Register	0x0000_0080
DIEPTSIZ15	0xED20_0AF0	R/W	Device IN Endpoint 15 Transfer Size Register	0x0000_0000
DIEPDMA15	0xED20_0AF4	R/W	Device IN Endpoint 15 DMA Address Register	0x0000_0000
Device Logical OUT Endpoint-Specific Registers				
DOEPCTL0	0xED20_0B00	R/W	Device Control OUT Endpoint 0 Control Register	0x0000_8000
DOEPINT0	0xED20_0B08	R/W	Device OUT Endpoint 0 Interrupt Register	0x0000_0000
DOEPTSIZ0	0xED20_0B10	R/W	Device OUT Endpoint 0 Transfer Size Register	0x0000_0000
DOEPDMA0	0xED20_0B14	R/W	Device OUT Endpoint 0 DMA Address Register	0x0000_0000
DOEPCTL1	0xED20_0B20	R/W	Device Control OUT Endpoint 1 Control Register	0x0000_0000
DOEPINT1	0xED20_0B28	R/W	Device OUT Endpoint 1 Interrupt Register	0x0000_0000
DOEPTSIZ1	0xED20_0B30	R/W	Device OUT Endpoint 1 Transfer Size Register	0x0000_0000
DOEPDMA1	0xED20_0B34	R/W	Device OUT Endpoint 1 DMA Address Register	0x0000_0000
DOEPCTL2	0xED20_0B40	R/W	Device Control OUT Endpoint 2 Control Register	0x0000_0000
DOEPINT2	0xED20_0B48	R/W	Device OUT Endpoint 2 Interrupt Register	0x0000_0000
DOEPTSIZ2	0xED20_0B50	R/W	Device OUT Endpoint 2 Transfer Size Register	0x0000_0000
DOEPDMA2	0xED20_0B54	R/W	Device OUT Endpoint 2 DMA Address Register	0x0000_0000
DOEPCTL3	0xED20_0B60	R/W	Device Control OUT Endpoint 3 Control Register	0x0000_0000
DOEPINT3	0xED20_0B68	R/W	Device OUT Endpoint 3 Interrupt Register	0x0000_0000
DOEPTSIZ3	0xED20_0B70	R/W	Device OUT Endpoint 3 Transfer Size Register	0x0000_0000
DOEPDMA3	0xED20_0B74	R/W	Device OUT Endpoint 3 DMA Address Register	0x0000_0000
DOEPCTL4	0xED20_0B80	R/W	Device Control OUT Endpoint 4 Control Register	0x0000_0000
DOEPINT4	0xED20_0B88	R/W	Device OUT Endpoint 4 Interrupt Register	0x0000_0000
DOEPTSIZ4	0xED20_0B90	R/W	Device OUT Endpoint 4 Transfer Size Register	0x0000_0000
DOEPDMA4	0xED20_0B94	R/W	Device OUT Endpoint 4 DMA Address Register	0x0000_0000
DOEPCTL5	0xED20_0BA0	R/W	Device Control OUT Endpoint 5 Control Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
DOEPINT5	0xED20_0BA8	R/W	Device OUT Endpoint 5 Interrupt Register	0x0000_0000
DOEPTSIZ5	0xED20_0BB0	R/W	Device OUT Endpoint 5 Transfer Size Register	0x0000_0000
DOEPDMA5	0xED20_0BB4	R/W	Device OUT Endpoint 5 DMA Address Register	0x0000_0000
DOEPCTL6	0xED20_0BC0	R/W	Device Control OUT Endpoint 6 Control Register	0x0000_0000
DOEPINT6	0xED20_0BC8	R/W	Device OUT Endpoint 6 Interrupt Register	0x0000_0000
DOEPTSIZ6	0xED20_0BD0	R/W	Device OUT Endpoint 6 Transfer Size Register	0x0000_0000
DOEPDMA6	0xED20_0BD4	R/W	Device OUT Endpoint 6 DMA Address Register	0x0000_0000
DOEPCTL7	0xED20_0BE0	R/W	Device Control OUT Endpoint 7 Control Register	0x0000_0000
DOEPINT7	0xED20_0BE8	R/W	Device OUT Endpoint 7 Interrupt Register	0x0000_0000
DOEPTSIZ7	0xED20_0BF0	R/W	Device OUT Endpoint 7 Transfer Size Register	0x0000_0000
DOEPDMA7	0xED20_0BF4	R/W	Device OUT Endpoint 7 DMA Address Register	0x0000_0000
DOEPCTL8	0xED20_0C00	R/W	Device Control OUT Endpoint 8 Control Register	0x0000_0000
DOEPINT8	0xED20_0C08	R/W	Device OUT Endpoint 8 Interrupt Register	0x0000_0000
DOEPTSIZ8	0xED20_0C10	R/W	Device OUT Endpoint 8 Transfer Size Register	0x0000_0000
DOEPDMA8	0xED20_0C14	R/W	Device OUT Endpoint 8 DMA Address Register	0x0000_0000
DOEPCTL9	0xED20_0C20	R/W	Device Control OUT Endpoint 9 Control Register	0x0000_0000
DOEPINT9	0xED20_0C28	R/W	Device OUT Endpoint 9 Interrupt Register	0x0000_0000
DOEPTSIZ9	0xED20_0C30	R/W	Device OUT Endpoint 9 Transfer Size Register	0x0000_0000
DOEPDMA9	0xED20_0C34	R/W	Device OUT Endpoint 9 DMA Address Register	0x0000_0000
DOEPCTL10	0xED20_0C40	R/W	Device Control OUT Endpoint 10 Control Register	0x0000_0000
DOEPINT10	0xED20_0C48	R/W	Device OUT Endpoint 10 Interrupt Register	0x0000_0000
DOEPTSIZ10	0xED20_0C50	R/W	Device OUT Endpoint 10 Transfer Size Register	0x0000_0000
DOEPDMA10	0xED20_0C54	R/W	Device OUT Endpoint 10 DMA Address Register	0x0000_0000
DOEPCTL11	0xED20_0C60	R/W	Device Control OUT Endpoint 11 Control Register	0x0000_0000
DOEPINT11	0xED20_0C68	R/W	Device OUT Endpoint 11 Interrupt Register	0x0000_0000
DOEPTSIZ11	0xED20_0C70	R/W	Device OUT Endpoint 11 Transfer Size Register	0x0000_0000
DOEPDMA11	0xED20_0C74	R/W	Device OUT Endpoint 11 DMA Address Register	0x0000_0000
DOEPCTL12	0xED20_0C80	R/W	Device Control OUT Endpoint 12 Control Register	0x0000_0000
DOEPINT12	0xED20_0C88	R/W	Device OUT Endpoint 12 Interrupt Register	0x0000_0000
DOEPTSIZ12	0xED20_0C90	R/W	Device OUT Endpoint 12 Transfer Size Register	0x0000_0000
DOEPDMA12	0xED20_0C94	R/W	Device OUT Endpoint 12 DMA Address Register	0x0000_0000
DOEPCTL13	0xED20_0CA0	R/W	Device Control OUT Endpoint 13 Control Register	0x0000_0000
DOEPINT13	0xED20_0CA8	R/W	Device OUT Endpoint 13 Interrupt Register	0x0000_0000
DOEPTSIZ13	0xED20_0CB0	R/W	Device OUT Endpoint 13 Transfer Size Register	0x0000_0000
DOEPDMA13	0xED20_0CB4	R/W	Device OUT Endpoint 13 DMA Address Register	0x0000_0000
DOEPCTL14	0xED20_0CC0	R/W	Device Control OUT Endpoint 14 Control Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
DOEPINT14	0xED20_0CC8	R/W	Device OUT Endpoint 14 Interrupt Register	0x0000_0000
DOEPTSIZ14	0xED20_0CD0	R/W	Device OUT Endpoint 14 Transfer Size Register	0x0000_0000
DOEPDMA14	0xED20_0CD4	R/W	Device OUT Endpoint 14 DMA Address Register	0x0000_0000
DOEPCTL15	0xED20_0CE0	R/W	Device Control OUT Endpoint 15 Control Register	0x0000_0000
DOEPINT15	0xED20_0CE8	R/W	Device OUT Endpoint 15 Interrupt Register	0x0000_0000
DOEPTSIZ15	0xED20_0CF0	R/W	Device OUT Endpoint 15 Transfer Size Register	0x0000_0000
DOEPDMA15	0xED20_0CF4	R/W	Device OUT Endpoint 15 DMA Address Register	0x0000_0000
Power and Clock Gating Register				
PCGCCTL	0xED20_0E00	R/W	Power and Clock Gating Control Register	0x0000_0000

NOTE: All registers of HS OTG Controller are accessible by word unit with STR/ LDR instructions.

8.2 DETAILED DESCRIPTION

8.2.1 OTG PHY Power Control Register (OPHYPWR, R/W, address = 0xED30_0000)

OPHYPWR	Bit	Description	R/W	Reset Value
Reserved	[31:5]	-		27'h0
otg_disable	[4]	OTG block power down in PHY2.0 • 1'b0 : OTG block power up • 1'b1 : OTG block power down If the application does not use OTG functionality, set this input high to save power.		1'b1
Analog _powerdown	[3]	Analog block power down in PHY2.0 • 1'b0 : Analog block power up (Normal Operation) • 1'b1 : Analog block power down	R/W	1'b1
Reserved	[2:1]	-		2b'
force_suspend	[0]	Apply Suspend signal to save power • 1'b0 : Disables (Normal Operation) • 1'b1 : Enables	R/W	1'b1

8.2.2 OTG PHY Clock Control Register (OPHYCLK, R/W, address = 0xED30_0004)

OPHYCLK	Bit	Description	R/W	Reset Value
Reserved	[31:7]	-		25'h0
serial_mode	[6]	UTMI/ Serial Interface Select If this register is asserted, USB traffic flows through the serial interface. • 1'b0: Data on the D+ and D- lines is transmitted and received through the UTMI. • 1'b1: Data on the D+ and D- lines is transmitted and received through the USB1.1 Serial Interface.	R/W	1'b0
xo_ext_clk_enb	[5]	Reference Clock Select for XO Block • 1'b0: External crystal • 1'b1: External clock/ Oscillator	R/W	1'b0
common_on_n	[4]	Force XO, Bias, Bandgap, and PLL to Remain Powered During a Suspend This bit controls the power-down signals of sub-blocks in the Common block if the USB 2.0 OTG PHY is suspended. • 1'b0 : 48MHz clock on clk48m_ohci is available at all times, except in Suspend mode. • 1'b1 : 48MHz clock on clk48m_ohci is available at all times, even in Suspend mode.	R/W	1'b0
Reserved	[3]	-		1'b0
id_pullup	[2]	Analog ID Input Sample Enable • 1'b0 : id_dig disable. • 1'b1 : id_dig enable. (The id_dig output is valid, and within 20ms, id_dig must indicate the type of plug connected.)	R/W	1'b0
clk_sel	[1:0]	Reference Clock Frequency Select for PLL • 2'b00 : 48MHz • 2'b01 : Reserved • 2'b10 : 12MHz • 2'b11 : 24MHz	R/W	2'b00

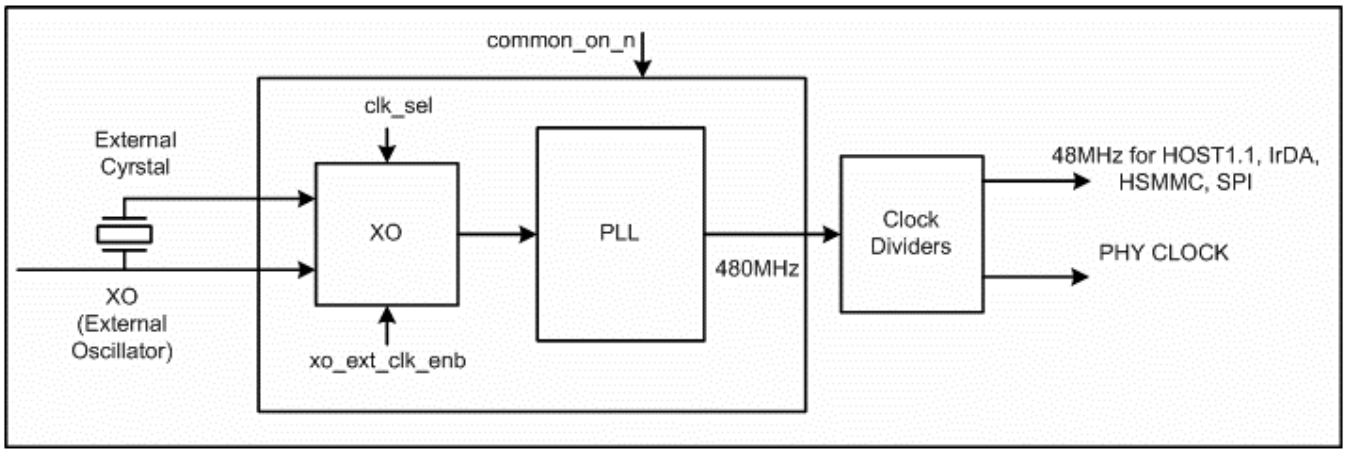


Figure 8.10-4 OTG PHY Clock Path

8.2.3 OTG Reset Control Register (ORSTCON, R/W, address = 0xED30_0008)

ORSTCON	Bit	Description	R/W	Reset Value
Reserved	[31:3]	-		29'h0
phylnk_sw_rst	[2]	OTG Link Core phy_clock domain S/W Reset	R/W	1'b0
link_sw_rst	[1]	OTG Link Core hclk domain S/W Reset	R/W	1'b0
phy_sw_rst	[0]	OTG PHY 2.0 S/W Reset The phy_sw_rst signal must be asserted for at least 10us	R/W	1'b1

8.2.4 PHY Tune Register(PHY_TUNE, R/W, address = 0xED30_0020)

OPHYTUNE	Bit	Description	R/W	Reset Value
Reserved	[31:21]	-		11'h1
txpreemphasistune	[20]	HS Transmitter Pre-Emphasis Enable. This signal enables or disables the pre-emphasis for a J-K or K-J state transition in HS mode. • 1 : Enables HS Transmitter pre-emphasis. • 0 : Disables HS Transmitter pre_empahsis.	R/W	1'b0
compdistune	[19:17]	Disconnect Threshold Adjustment. This bit field adjusts the voltage level for the threshold used to detect a disconnect event at the host. • 111 : +6% • 110 : +4.5% • 101 : +3% • 100 : +1.5% • 011 : Design default • 010 : -3% • 001 : -4% • 000 : -6%	R/W	3'b011
otgtune	[16:14]	VBUS Valid Threshold Adjustment. This bit field adjusts the voltage level for the VBUS Valid threshold. • 111 : +9% • 110 : +6% • 101 : +3% • 100 : Design default (4.75V) • 011 : -3% • 010 : -6% • 001 : -9% • 000 : -12%	R/W	3'b100
sqrxtune	[13:11]	Squelch Threshold Tune. This bit field adjusts the voltage level for the threshold used to detect valid high-speed data. • 111 : -20% • 110 : -15% • 101 : -10% • 100 : -5% • 011 : Design default • 010 : +5% • 001 : +10% • 000 : +15%	R/W	3'b011

OPHYTUNE	Bit	Description	R/W	Reset Value
txfslstune	[10:7]	FS/ LS Pull-Up Resistance Adjustment. This bit field adjusts the low-and full-speed pull-up resistance based on nominal power, voltage, and temperature. • 1111 : -2.5% • 0111 : Design default • 0011 : +2.5% • 0001 : +5% • 0000 : +7.5%	R/W	4'b0111
txrisetune	[6]	HS Transmitter Rise/ Fall Time Adjustment. This bit field adjusts the rise/ fall times of the high speed waveform. • 1 : -8% • 0 : Design default	R/W	1'b0
txhsxvtune	[5:4]	Transmitter High-Speed Crossover adjustment. This bit field adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. • 11 : The crossover voltage is increased by 15mV • 10 : The crossover voltage is increased by 30mV • 01 : Default setting • 00 : Reserved	R/W	2'b01
txvreftune	[3:0]	HS DC Voltage Level Adjustment. This bit field adjusts the voltage to which the high-speed DC level is tuned. • 1111 : +8.75% • 1110 : +7.5% • 1101 : +6.25% • 1100 : +5% • 1011 : +3.75% • 1010 : +2.5% • 1001 : +1.25% • 1000 : • 0111 : -1.25% • 0110 : -2.5% • 0101 : -3.75% • 0100 : -5% • 0011 : Design default • 0010 : -7.5% • 0001 : -8.75% • 0000 :	R/W	4'b0011

These registers are available in both Host and Device modes, and not required to be reprogrammed to switch between these modes.

8.2.5 OTG Control and Status Register (GOTGCTL, R/W, Address = 0xED20_0000)

The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.

GOTGCTL	Bit	Description	R/W	Reset Value
Reserved	[31:20]	-		12'h0
BSesVld	[19]	B-Session Valid Indicates the Device mode transceiver status. • 1'b0 : B-session is not valid • 1'b1 : B-session is valid	R	1'b0
ASesVld	[18]	A-Session Valid Indicates the Host mode transceiver status. • 1'b0 : A-session is not valid • 1'b1 : A-session is valid	R	1'b0
DbncTime	[17]	Long/ Short Debounce Time Indicates the Debounce time of a detected connection. • 1'b0 : Long Debounce time, used for physical connections • 1'b1 : Short Debounce time, used for soft connections	R	1'b0
ConIDSts	[16]	Connector ID Status Indicates the connector ID status. • 1'b0 : The OTG core is in A-device mode • 1'b1 : The OTG core is in B-device mode	R	1'b1
Reserved	[15:12]	-		4'h0
DevHNPEn	[11]	Device HNP Enable The application sets the bit if it successfully receives a SetFeature. • 1'b0 : HNP is not enabled in the application • 1'b1 : HNP is enabled in the application	R/W	1'b0
HstSet HNPEn	[10]	Host Set HNP Enable The application sets this bit if it has successfully enabled HNP on the connected device. • 1'b0 : Host Set HNP is not enabled • 1'b1 : Host Set HNP is enabled	R/W	1'b0
HNPReq	[9]	HNP Request The application sets this bit to initiate an HNP request to the connected USB host. The core clears this bit if the HstNegSucStsChng bit is cleared. • 1'b0 : No HNP request • 1'b1 : HNP request	R/W	1'b0

GOTGCTL	Bit	Description	R/W	Reset Value
HstNegScs	[8]	Host Negotiation Success The core sets this bit if host negotiation is successful. The core clears this bit if the HNP Request (HNPReq) bit in this register is set. • 1'b0 : Host negotiation failure • 1'b1 : Host negotiation success	R	1'b0
Reserved	[7:2]	-		6'h0
SesReq	[1]	Session Request The application sets this bit to initiate a session request on the USB. The core clears this bit if the HstNegSucStsChng bit is cleared. • 1'b0 : No session request • 1'b1 : Session request	R/W	1'b0
SesReqScs	[0]	Session Request Success The core sets this bit if a session request initiation is successful. • 1'b0 : Session request failure • 1'b1 : Session request success	R	1'b0

8.2.6 OTG Interrupt Register (GOTGINT, R/W, Address = 0xED20_0004)

The application reads this register at the time of OTG interrupt and clears the bits in this register to clear the OTG interrupt.

GOTGINT	Bit	Description	R/W	Reset Value
Reserved	[31:20]	-		12'h0
DbnceDone	[19]	Debounce Done The core sets this bit if the debounce is complete after the device connects. This bit is only valid if the HNP Capable or SRP Capable bit is set in the Core USB Configuration register.	R_SS_WC	1'b0
ADev TOUTChg	[18]	A-Device Timeout Change The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.	R_SS_WC	1'b0
HstNegDet	[17]	Host Negotiation Detected. The core sets this bit if it detects a host negotiation request on the USB.	R_SS_WC	1'b0
Reserved	[16:10]	-		7'h0
HstnegSuc StsChng	[9]	Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request.	R_SS_WC	1'b0
SesReq SucStsChng	[8]	Session Request Success Status Change The core sets this bit on the success or failure of a session request.	R_SS_WC	1'b0
Reserved	[7:3]	-		5'h0
SesEndDet	[2]	Session End Detected The core sets this bit if the b_valid signal is deasserted.	R_SS_WC	1'b0
Reserved	[1:0]	-		2'h0

8.2.7 OTG AHB Configuration Register (GAHBCFG, R/W, Address = 0xED20_0008)

This register configures the core after power-on or a change in mode of operation. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.

GAHBCFG	Bit	Description	R/W	Reset Value
Reserved	[31:9]	-		23'h0
PTxFEmpLvl	[8]	Periodic TxFIFO Empty Level Indicates if the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt registers (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. • 1'b0 : GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty • 1'b1 : GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	R/W	1'b0
NPTxFEmp Lvl	[7]	Non-Periodic TxFIFO Empty Level Indicates if the Non-Periodic TxFIFO Empty Interrupt bits in the Core Interrupt register (GINSTS.NPTxFEmp) is triggered. This bit is used only in Slave mode. • 1'b0 : GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty • 1'b1 : GINTSTS.NPTxFEmp interrupt indicates that the Non Periodic TxFIFO is completely empty	R/W	1'b0
Reserved	[6]	-		1'b0
DMAEn	[5]	DMA Enable • 1'b0 : Core operates in Slave mode • 1'b1 : Core operates in a DMA mode	R/W	1'b0
HBstLen	[4:1]	Burst Length/vType Internal DMA Mode – AHB Master burst type: • 4'b0000 : Single • 4'b0001 : INCR • 4'b0011 : INCR4 • 4'b0101 : INCR8 • 4'b0111 : INCR16 • Others : Reserved	R/W	4'b0
GlblIntrMsk	[0]	Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion to itself. • 1'b0 : Mask the interrupt assertion to the application • 1'b1 : Unmask the interrupt assertion to the application	R/W	1'b0

8.2.8 OTG USB Configuration Register (GUSBCFG, R/W, Address = 0xED20_000C)

This register configures the core after power-on or a changing to Host mode or Device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.

GUSBCFG	Bit	Description	R/W	Reset Value
Reserved	[31:16]	-		16'h0
PHY Low-Power Clock Select	[15]	PHY Low-Power Clock Select Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY usually operate on a 48-MHz clock to save power. • 1'b0 : 480-MHz Internal PLL clock • 1'b1 : 48-MHz External clock *__Note:__ This bit must be configured with OPHYPWR.pll_powerdown.		1'b0
Reserved	[14:10]	-		5'h5
HNPCap	[9]	HNP – Capable The application uses this bit to control the OTG cores's HNP capabilities. • 1'b0 : HNP capability is not enabled • 1'b1 : HNP capability is enabled	R/W	1'b0
SRPCap	[8]	SRP – Capable The application uses this bit to control the OTG core's SRP capabilities. • 1'b0 : SRP capability is not enabled • 1'b1 : SRP capability is enabled	R/W	1'b0
Reserved	[7:4]	-		4'h0
PHYIf	[3]	PHY Interface The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. Only 16-bit interface is supported. This bit must be set to 1. • 1'b0 : 8 bits • 1'b1 : 16 bits	R/W	1'b0
TOutCal	[2:0]	HS/ FS Timeout Calibration Set this bit to 3'h7.	R/W	3'h0

8.2.9 Core Reset Register (GRSTCTL, R/W, Address = 0xED20_0010)

The application uses this register to reset various hardware features inside the core.

GRSTCTL	Bit	Description	R/W	Reset Value
AHBIdle	[31]	AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.	R	1'b1
DMAReq	[30]	DMA Request Signal Indicates that the DMA request is in progress. Used for debug.	R	1'b0
Reserved	[29:11]	-		19'h0
TxFNum	[10:6]	TxFIFO Number This is the FIFO number. Use TxFIFO Flush bit to flush FIFO number. This field must not be changed until the core clears the TxFIFO Flush bit. • 5'h0 : Non-Periodic TxFIFO flush • 5'h1 : Periodic TxFIFO 1 flush in Device mode for Periodic TxFIFO flush in Host mode • 5'h2 : Periodic TxFIFO 2 flush in Device mode • • • • 5'hF : Periodic TxFIFO 15 flush in Device mode • 5'h10 : Flush all the Periodic and Non-Periodic TxFIFOs in the core	R/W	5'h0
TxFFlsh	[5]	TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot flush if the core is in the middle of a transaction. The application must only write this bit after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. The application must wait until the core clears this bit before performing any operations. This bit takes 8 clocks to clear.	R_WS_SC	1'b0
RxFFlsh	[4]	RxFIFO Flush The application flushes the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit takes 8 clocks to clear.	R_WS_SC	1'b0
INTknQFlsh	[3]	IN Token Sequence Learning Queue Flush The application writes this bit to flush the IN Token Sequence Learning Queue.	R_WS_SC	1'b0
FrmCntrRst	[2]	Host Frame Counter Reset The application writes this bit to reset the (micro) frame number counter inside the core. If the (micro) frame counter is reset, the subsequent SOF sent out by the core will have a (micro) frame number of 0.	R_WS_SC	1'b0

GRSTCTL	Bit	Description	R/W	Reset Value
HSftRst	[1]	HClk Soft Reset The application uses this bit to flush the control logic in the AHB Clock domain. Only AHB Clock Domain pipelines are reset. • FIFOs are not flushed with this bit. • All state machines in the AHB clock Domain are reset to IDLE state after terminating the transactions on the AHB, following the protocol. • Control bits in the CSRs that the AHB Clock domain state machines use are cleared. • Status mask bits generated by the AHB Clock domain state machine that control the interrupt status, are cleared to clear the interrupt. • Because interrupt status bits are not cleared, the application gets the status of any core events that occurred after this bit is set. This is a self-clearing bit that the core clears after all necessary logic is reset in the core. This may take several clocks, depending on the core's current state.	R_WS_SC	1'b0
CSftRst	[0]	Core Soft Reset Resets the hclk and phy_clock domains as follows: • Clears the interrupts and all the CSR registers except the following register bits: - HCFG.FSLSPclkSel - DCFG.DevSpd • All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. • Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which may take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain. Software must also check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and if you dynamically change the PHY selection bits in the USB configuration registers listed above. If you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.	R_WS_SC	1'b0

8.2.10 Core Interrupt Register (GINTSTS, R/W, Address = 0xED20_0014)

This register interrupts the application for system-level events in the current mode of operation (Device mode or Host mode).

GINTSTS	Bit	Description	R/W	Reset Value
WkUpInt	[31]	Resume/ Remote Wakeup Detected Interrupt In Device mode, this interrupt is asserted if a resume is detected on the USB. In Host mode, this interrupt is asserted if a remote wakeup is detected on the USB.	R_SS_WC	1'b0
SessReqInt	[30]	Session Request/ New Session Detected Interrupt In Host mode, this interrupt is asserted if a session request is detected from the device. In Device mode, this interrupt is asserted if the b_valid signal goes high.	R_SS_WC	1'b0
DisconnInt	[29]	Disconnect Detected Interrupt Asserted when a device disconnect is detected.	R_SS_WC	1'b0
ConIDSts Chng	[28]	Connector ID Status Change The core sets this bit if there is a change in connector ID status.	R_SS_WC	1'b0
Reserved	[27]	-		1'b0
PTxFEmp	[26]	Periodic TxFIFO Empty Asserted if the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register.	R	1'b1
HChInt	[25]	Host Channels Interrupt The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	R	1'b0
PrtInt	[24]	Host Port Interrupt The core sets this bit to indicate a change in port status of one of the OTG core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	R	1'b0
Reserved	[23]	-		1'b0

GINTSTS	Bit	Description	R/W	Reset Value
FetSusp	[22]	Data Fetch Suspended. This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: • Sets a global non-periodic IN NAK handshake • Disables In endpoints • Flushes the FIFO • Determines the token sequence from the IN Token Sequence Learning Queue • Re-enables the endpoints • Clears the global non-periodic IN NAK handshake If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token received: the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application checks the GINSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application masks the "IN token received when FIFO empty" interrupt if clearing a global IN NAK handshake.	R_SS_WC	1'b0
incompIP	[21]	Incomplete Periodic Transfer. In Host mode, the core sets this interrupt bit if there are incomplete periodic transactions still pending which are scheduled for the current microframe.	R_SS_WC	1'b0
incompl SOOUT		Incomplete Isochronous OUT Transfer. The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not complete in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.		
Incompl SOIN	[20]	Incomplete Isochronous IN Transfer. The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not complete in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	R_SS_WC	1'b0
OEPInt	[19]	OUT Endpoints Interrupt. The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	R	1'b0

GINTSTS	Bit	Description	R/W	Reset Value
IEPInt	[18]	IN Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.	R	1'b0
EPMis	[17]	Endpoint Mismatch Interrupt Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-Periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.	R_SS_WC	1'b0
Reserved	[16]	-		1'b0
EOPF	[15]	End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.	R_SS_WC	1'b0
ISOutDrop	[14]	Isochronous OUT Packet Dropped Interrupt The core sets this bit if it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	R_SS_WC	1'b0
EnumDone	[13]	Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.	R_SS_WC	1'b0
USBRst	[12]	USB Reset The core sets this bit to indicate that a reset is detected on the USB.	R_SS_WC	1'b1
USBSusp	[11]	USB Suspend The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state if there is no activity on the line_state signal for an extended period of time.	R_SS_WC	1'b0
ErlySusp	[10]	Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.	R_SS_WC	1'b0
Reserved	[9]	-		1'b0
Reserved	[8]	-		1'b0

GINTSTS	Bit	Description	R/W	Reset Value
GOUTNak Eff	[7]	Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit is cleared by writing the Clear Global OUT NAK bit in the Device Control register.	R	1'b0
GINNakEff	[6]	Global IN Non-Periodic NAK Effective Indicates that the Set Global Non-Periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit is cleared by clearing the Clear Global Non-Periodic IN NAK bit set by the application. This bit is cleared by clearing the Clear Global Non-Periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.	R	1'b0
NPTxFEmp	[5]	Non-Periodic TxFIFO Empty This interrupt is asserted if the Non-Periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-Periodic Transmit Request Queue. The half or completely empty status is determined by the Non-Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl).	R	1'b1
RxFLvl	[4]	RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.	R	1'b0
Sof	[3]	Start of (micro) Frame In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application reads the Device Status register to get the current (micro) frame number. This interrupt is seen if the core is operating at either HS or FS.	R_SS_WC	1'b0
OTGInt	[2]	OTG Interrupt The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.	R	1'b0
ModeMis	[1]	Mode Mismatch Interrupt The core sets this bit if the application is trying to access: • A Host mode register, if the core is operating in Device mode • A Device mode register, if the core is operating in Host mode	R_SS_WC	1'b0

GINTSTS	Bit	Description	R/W	Reset Value
CurMod	[0]	Current Mode Of Operation Indicates the current mode of operation. • 1'b0 : Device mode • 1'b1 : Host mode	R	1'b0

8.2.11 Core Interrupt Mask Register (GINTMSK, R/W, Address = 0xED20_0018)

This register works with the Core Interrupt register to interrupt the application. If an interrupt bit is masked, the interrupt associated with that bit will not be generated. However, the Core Interrupt (GINTSTS) register bit corresponding to that interrupt will still be set.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

GINTMSK	Bit	Description	R/W	Reset Value
WkUpIntMsk	[31]	Resume/ Remote Wakeup Detected Interrupt Mask	R/W	1'b0
SessReqIntMsk	[30]	Session Request/ New Session Detected Interrupt Mask	R/W	1'b0
DisconnIntMsk	[29]	Disconnect Detected Interrupt Mask	R/W	1'b0
ConIDStsChngMsk	[28]	Connector ID Status Change Mask	R/W	1'b0
Reserved	[27]	-		1'b0
PTxFEmpMsk	[26]	Periodic TxFIFO Empty Mask	R/W	1'b0
HChIntMsk	[25]	Host Channels Interrupt Mask	R/W	1'b0
PrtIntMsk	[24]	Host Port Interrupt Mask	R/W	1'b0
Reserved	[23]	-		1'b0
FetSuspMsk	[22]	Data Fetch Suspended Mask	R/W	1'b0
incomplPMsk	[21]	Incomplete Periodic Transfer Mask	R/W	1'b0
incompISOOUTMsk		Incomplete Isochronous OUT Transfer Mask		
incompISOINMsk	[20]	Incomplete Isochronous IN Transfer Mask	R/W	1'b0
OEPIntMsk	[19]	OUT Endpoints Interrupt Mask	R/W	1'b0
INEPIntMsk	[18]	IN Endpoints Interrupt Mask	R/W	1'b0
EPMisMsk	[17]	Endpoint Mismatch Interrupt Mask	R/W	1'b0
Reserved	[16]	-		1'b0
EOPFMsk	[15]	End of Periodic Frame Interrupt Mask	R/W	1'b0
ISOOutDropMsk	[14]	Isochronous OUT Packet Dropped Interrupt Mask	R/W	1'b0
EnumDoneMsk	[13]	Enumeration Done Mask	R/W	1'b0
USBRstMsk	[12]	USB Reset Mask	R/W	1'b0
USBSuspMsk	[11]	USB Suspend Mask	R/W	1'b0
ErlySuspMsk	[10]	Early Suspend Mask	R/W	1'b0
Reserved	[9]	-		1'b0
Reserved	[8]	-		1'b0
GOUTNakEffMsk	[7]	Global OUT NAK Effective Mask	R/W	1'b0
GINNakEffMsk	[6]	Global Non-Periodic IN NAK Effective Mask	R/W	1'b0
NPTxFEmpMsk	[5]	Non-Periodic TxFIFO Empty Mask	R/W	1'b0
RxFLvlMsk	[4]	Receive FIFO Non-Empty Mask	R/W	1'b0
SofMsk	[3]	Start of (micro)Frame Mask	R/W	1'b0

GINTMSK	Bit	Description	R/W	Reset Value
OTGIntMsk	[2]	OTG Interrupt Mask	R/W	1'b0
ModeMisMsk	[1]	Mode Mismatch Interrupt Mask	R/W	1'b0
Reserved	[0]	-		1'b0

8.2.12 Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

A read to the Receive Status Debug Read register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO.

The receive status contents must be interpreted differently in Host and Device modes. The core ignores the receive status pop/ read if the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO if the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

8.2.13 Host Mode Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP, R, Address = 0xED20_001C, 0xED20_0020)

GRXSTSR/ GRXSTSP	Bit	Description	R/W	Reset Value
Reserved	[31:21]	-		-
PktSts	[20:17]	Packet Status Indicates the status of the received packet. • 4'b0010 : IN data packet received • 4'b0011 : IN transfer completed (triggers an interrupt) • 4'b0101 : Data toggle error (triggers an interrupt) • 4'b0111 : Channel halted (triggers an interrupt) • others : Reserved	R	-
DPID	[16:15]	Data PID Indicates the Data PID of the received packet. • 2'b00 : DATA0 • 2'b10 : DATA1 • 2'b01 : DATA2 • 2'b11 : MDATA	R	-
BCnt	[14:4]	Byte Count Indicates the byte count of the received IN data packet.	R	-
ChNum	[3:0]	Channel number Indicates the channel number to which the current received packet belongs.	R	-

8.2.14 Device Mode Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP, R, Address = 0xED20_001C, 0xED20_0020)

GRXSTSR/ GRXSTSP	Bit	Description	R/W	Reset Value
Reserved	[31:25]	-		7'h3F
FN	[24:21]	Frame Number This is the least significant 4 bits of the (micro) frame number in which the packet is received on the USB. This field is supported if isochronous OUT endpoints are supported.	R	4'hF
PktSts	[20:17]	Packet Status Indicates the status of the received packet. • 4'b0001: Global OUT NAK (triggers an interrupt) • 4'b0010: OUT data packet received • 4'b0011: OUT transfer completed (triggers an interrupt) • 4'b0100: SETUP transaction completed (triggers an interrupt) • 4'b0110: SETUP data packet received • others: Reserved	R	4'b1111
DPID	[16:15]	Data PID Indicates the Data PID of the received OUT data packet. • 2'b00 : DATA0 • 2'b10 : DATA1 • 2'b01 : DATA2 • 2'b11 : MDATA	R	2'b11
BCnt	[14:4]	Byte Count Indicates the byte count of the received data packet.	R	11'h3FF
EPNum	[3:0]	Endpoint number Indicates the endpoint number to which the current received packet belongs.	R	4'hF

8.2.15 Receive FIFO Size Register (GRXFSIZ, R/W, Address = 0xED20_0024)

The application programs the RAM size that must be allocated to the RxFIFO.

GRXFSIZ	Bit	Description	R/W	Reset Value
Reserved	[31:16]	-		16'h0
RxFDep	[15:0]	RxFIFO Depth This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 6144 The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. A new value must be written to this field. Programmed values must not exceed the power-on value set.	R/W	16'h1800

8.2.16 Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ, R/W, Address = 0xED20_0028)

The application programs the RAM size and the memory start address for the Non-Periodic TxFIFO.

GNPTXFSIZ	Bit	Description	R/W	Reset Value
NPTxFDep	[31:16]	Non-Periodic TxFIFO Depth This value is in terms of 32-bit words. • Minimum value is 16 • Maximum value is 32768 The power-on reset value of this register is specified as the Largest Non-Periodic Tx Data FIFO Depth (6144). A new value must be written to this field. Programmed values must not exceed the power-on value set.	R/W	16'h1800
NPTxFStAddr	[15:0]	Non-Periodic Transmit Start Address This field contains the memory start address for Non-Periodic Transmit FIFO RAM. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth (6144). A new value must be written to this field. Programmed values must not exceed the power-on value set.	R/W	16'h1800

8.2.17 Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS, R, Address = 0xED20_002C)

This read-only register contains the free space information for the Non-Periodic TxFIFO and the Non-Periodic Transmit Request Queue.

GNPTXSTS	Bit	Description	R/W	Reset Value
Reserved	[31]	-		1'b0
NPTxQTop	[30:24]	Top of the Non-Periodic Transmit Request Queue. Entry in the Non-Periodic Tx Request Queue that is currently being processed by the MAC. • Bits[30:27] : Channel/ endpoint number • Bits[26:25] : • 2'b00 : IN/ OUT token • 2'b01 : Zero-length transmit packet (device IN/host OUT) • 2'b10 : PING/CSPLIT token • 2'b11 : Channel halt command • Bit[24] : Terminate (last entry for selected channel/ endpoint)	R	7'h0
NPTxQSpcAvail	[23:16]	Non-Periodic Transmit Request Queue Space Available. Indicates the amount of free space available in the Non-Periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. • 8'h0 : Non-Periodic Transmit Request Queue is full • 8'h1 : 1 location available • 8'h2 : 2 locations available • n : n locations available(0≤n≤8) • Others : Reserved	R	8'h08
NPTxFSpcAvail	[15:0]	Non-Periodic TxFIFO Space Available Indicates the amount of free space available in the Non-Periodic TxFIFO. Values are in terms of 32-bit words. • 16'h0 : Non-Periodic TxFIFO is full • 16'h1 : 1 word available • 16'h2 : 2 words available • 16'hn : n words available (where 0 ≤ n ≤ 32768) • 16'h8000 : 32768 words available • Others : Reserved	R	16'h1800

8.2.18 Host Periodic Transmit FIFO Size Register (HPTXFSIZ, R/W, Address = 0xED20_0100)

This register holds the size and the memory start address of the Periodic TxFIFO.

HPTXFSIZ	Bit	Description	R/W	Reset Value
PTxFSize	[31:16]	Host Periodic TxFIFO Depth This value is in terms of 32-bit words • Minimum value is 16 • Maximum value is 6144 A new value must be written to this field. Programmed values must not exceed the Maximum value.	R/W	16'h0300
PTxFStAddr	[15:0]	Host Periodic TxFIFO Start Address The power-on reset value of this register is sum of the Largest Rx Data FIFO Depth and Largest Non –Periodic Tx Data FIFO Depth specified. If you have programmed new values for the RxFIFO or Non-Periodic TxFIFO, write their sum in this field. Programmed values must not exceed the power-on value.	R/W	16'h5A00

8.2.19 Device Periodic Transmit FIFO-n Size Register (DPTXFSIZn, R/W, Address = 0xED20_0104 + (n-1)*04h)

FIFO_number: 1 ≤ n ≤ 15

This register holds the memory start address of each periodic TxFIFO to implement in Device mode. Each periodic FIFO holds the data for one periodic IN endpoint. This register is repeated for each periodic FIFO instantiated.

DPTXFSIZn	Bit	Description	R/W	Reset Value
DPTxFSize	[31:16]	Device Periodic TxFIFO Size This value is in terms of 32-bit words • Minimum value is 4 • Maximum value is 768 The power-on reset value of this register is the Largest Device Mode Periodic Tx Data FIFO-n Depth. Write a new value to this field.	R/W	n :1 (16'h300) n :2 (16'h300) n :3 (16'h300) n :4 (16'h300) n :5 (16'h300) n :6 (16'h300) n :7 (16'h300) n :8 (16'h300) n :9 (16'h300) n :10 (16'h300) n :11 (16'h300) n :12 (16'h300) n :13 (16'h300) n :14 (16'h300) n :15 (16'h300)
DPTxFStAddr	[15:0]	Device Periodic TxFIFO RAM Start Address Holds the start address in the RAM for this periodic FIFO. The power-on reset value of this register is sum of the Largest Rx Data FIFO Depth, Largest Non-Periodic Tx Data FIFO Depth, and all lower numbered Largest Device Mode Periodic Tx Data FIFOn Depth specified. If you have programmed new values for the RxFIFO, Non-Periodic TxFIFO, or device Periodic TxFIFOs, write their sum in this field. Programmed values must not exceed the power-on value set.	R/W	n :1 (16'h1000) n :2 (16'h3300) n :3 (16'h3600) n :4 (16'h3900) n :5 (16'h3C00) n :6 (16'h3F00) n :7 (16'h4200) n :8 (16'h4500) n :9 (16'h4800) n :10 (16'h4B00) n :11 (16'h4E00) n :12 (16'h5100) n :13 (16'h5400) n :14 (16'h5700) n :15 (16'h5A00)

HOST MODE REGISTERS

These registers affect the operation of the core in the Host mode. Host mode registers must not be accessed in Device mode, as the results are undefined. Host Mode registers are categorized as follows:

- Host Global registers
- Host Port Control and Status registers
- Host Channel-Specific registers

8.2.20 Host Configuration Register (HCFG, R/W, Address = 0xED20_0400)

This register configures the core after power-on. Do not make changes to this register after initializing the host.

HCFG	Bit	Description	R/W	Reset Value
Reserved	[31:3]	-		29'h0040000
FSLSSupp	[2]	FS- and LS- Only Support The application uses this bit to control the core's enumeration speed. Using this bit, the application makes the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. • 1'b0 : HS/FS/LS, based on the maximum speed supported by the connected device • 1'b1 : FS/LS -only, even if the connected device can support HS	R/W	1'b0
FSLSPclkSel	[1:0]	FS/ LS PHY Clock Select If the core is in FS Host mode • 2'b00 : PHY clock is 30/60 MHz • 2'b01 : PHY clock is 48 MHz • Others : Reserved If the core is in LS Host mode • 2'b00 : PHY clock is 30/60 MHz • 2'b01 : PHY clock is 48 MHz • 2'b10 : PHY clock is 6 MHz • 2'b11 : Reserved	R/W	2'b0

8.2.21 Host Frame Interval Register (HFIR, R/W, Address = 0xED20_0404)

This register stores the frame interval information for the current speed to which the core has enumerated

HFNUM	Bit	Description	R/W	Reset Value
Reserved	[31:16]	-		16'h0
FrInt	[15:0]	Frame Interval The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro- SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation if the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/ LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. • 125 µs * (PHY clock frequency for HS) • 1 ms * (PHY clock frequency for FS/LS)	R/W	16'h17D7

8.2.22 Host Frame Number/Frame Time Remaining Register (HFNUM, R, Address = 0xED20_0408)

This register indicates the current frame number. It also indicates the time remaining in the current frame.

HFNUM	Bit	Description	R/W	Reset Value
FrRem	[31:16]	Frame Time Remaining Indicates the amount of time remaining in the current microframe (HS) or frame (FS/ LS), in terms of PHY clocks. This field decrements on each PHY clock. If it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.	R	16'h0
FrNum	[15:0]	Frame Number This field increment if a new SOF is transmitted on the USB, and is reset to 0 if it reaches 16'h3FFF.	R	16'h0

8.2.23 Host Periodic Transmit FIFO/QUEUE Status Register (HPTXSTS, R, Address = 0xED20_0410)

This read-only register contains the free space information for the Periodic TxFIFO and the Periodic Transmit Request Queue.

HPTXSTS	Bit	Description	R/W	Reset Value
PTxQTop	[31:24]	Top of the Periodic Transmit Request Queue This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. • Bit [31] : Odd/Even (micro)frame - 1'b0 : send in even (micro)frame - 1'b1 : send in odd (micro)frame • Bits [30:27] : Channel/endpoint number • Bits [26:25] : Type -2'b00 : IN/OUT -2'b01 : Zero-length packet -2'b10 : CSPLIT -2'b11 : Disable channel command • Bit[24] : Terminate	R	8'h0
PTxQSpcAvail	[23:16]	Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. • 8'h0 : Periodic Transmit Request Queue is full • 8'h1 : 1 location available • 8'h2 : 2 location available • n : n locations available (0 ≤ n ≤ 8) • Others : Reserved	R	8'h8
PTxFSpcAvail	[15:0]	Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words • 16'h0: Periodic TxFIFO is full • 16'h1: 1 word available • 16'h2: 2 words available • n: n words available (0 ≤ n ≤ 8) • Others: Reserved	R	16'h0100

8.2.24 Host All Channels Interrupt Register (HAINT, R, Address = 0xED20_0414)

If a significant event occurs on a channel, the Host All Channels Interrupt register interrupts the application using the Host Channels Interrupt bit of the Core Interrupt register. There is one interrupt bit per channel, up to a maximum of 16 bits. Bits in this register are set and cleared if the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

HAINT	Bit	Description	R/W	Reset Value
Reserved	[31:16]	-		16'h0
HAINT	[15:0]	Channel Interrupts One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	R	16'h0

8.2.25 Host All Channels Interrupt Mask Register (HAINTMSK, R/W, Address = 0xED20_0418)

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application if an event occurs on a channel. There is one interrupt mask bit per channel, up to a maximum of 16 bits.

- Msk interrupt: 1'b0
- Unmask interrupt: 1'b0

HAINTMSK	Bit	Description	R/W	Reset Value
Reserved	[31:16]	-		16'h0
HAINTMsk	[15:0]	Channel Interrupt Mask One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	R/W	16'h0

8.2.26 Host Port Control and Status Register (HPRT, R/W, Address = 0xED20_0440)

This register is available in both Host and Device modes. Currently, the OTG Host supports only one port. A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, connect status, and test mode for each port. The R_SS_WC bits in this register triggers an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register. On a Port Interrupt, the application must read this register and clear the bit that caused the interrupt. For the R_SS_WC bits, the application must write a 1 to the bit to clear the interrupt.

HPRT	Bit	Description	R/W	Reset Value
Reserved	[31:19]	-		13'h0
PrtSpd	[18:17]	Port Speed Indicates the speed of the device attached to this port. • 2'b00 : High speed • 2'b01 : Full speed • 2'b10 : Low speed • 2'b11 : Reserved	R	2'b0
PrtTstCtl	[16:13]	Port Test Control The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. • 4'b0000 : Test mode disabled • 4'b0001 : Test_J mode • 4'b0010 : Test_K mode • 4'b0011 : Test_SE0_NAK mode • 4'b0100 : Test_Packet mode • 4'b0101 : Test_Force_Enable • Others : Reserved	R/W	4'h0
PrtPwr	[12]	Port Power The application uses this field to control power to this port, and the core clears this bit on an overcurrent condition. • 1'b0 : Power off • 1'b1 : Power on	R_W_SC	1'b0
PrtLnSts	[11:10]	Port Line Status Indicates the current logic level USB data lines • Bit [10] : Logic level of D– • Bit [11] : Logic level of D+	R	2'b0
Reserved	[9]	-		1'b0

HPRT	Bit	Description	R/W	Reset Value
PrtRst	[8]	Port Reset If the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. • 1'b0 : Port not in reset • 1'b1 : Port in reset The application must leave this bit set for at least a minimum duration mentioned below to start a reset on the port. The application can leave it set for another 10ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard. • High speed : 50 ms • Full speed/Low speed : 10ms	R/W	1'b0
prtSusp	[7]	Port Suspend The application sets this bit to put this port in Suspend mode. The core stops sending SOFs if this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register. • 1'b0 : Port not in Suspend mode • 1'b1 : Port in Suspend mode	R_WS_SC	1'b0
PrtRes	[6]	Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/ Remote Wakeup Detected Interrupt bit of the Core Interrupt register, the core starts driving resume signaling without application intervention and clears this bit if it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. • 1'b0 : No resume driven • 1'b1 : Resume driven	R_W_SS_SC	1'b0
PrtOvr CurrChng	[5]	Port Overcurrent Change The core sets this bit if the status of the Port Overcurrent Active bit (bit 4) in this register changes.	R_SS_WC	1'b0
PrtOvr CurrAct	[4]	Port Overcurrent Active Indicates the overcurrent condition of the port. • 1'b0 : No overcurrent condition • 1'b1 : Overcurrent condition	R	1'b0

HPRT	Bit	Description	R/W	Reset Value
PrtEnChng	[3]	Port Enable/Disable Change The core sets this bit if the status of the Port Enable bit [2] of this register changes.	R_SS_WC	1'b0
PrtEna	[2]	Port Enable A port is enabled by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It clears it to disable the port. This bit does not trigger any interrupt to the application. • 1'b0 : Port disabled • 1'b1 : Port enabled	R_SS_SC_WC	1'b0
PrtConn Det	[1]	Port Connect Detected The core sets this bit if a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register. The application must write a 1 to this bit to clear the interrupt.	R_SS_WC	1'b0
PrtConnSts	[0]	Port Connect Status • 1'b0 : No device is attached to the port • 1'b1 : A device is attached to the port	R	1'b0

8.2.27 Host Channel-N Characteristics Register (HCCHARn, R/W, Address = 0xED20_0500+n*20h)

Channel_number: 0 ≤ n ≤ 15

HCCHARn	Bit	Description	R/W	Reset Value
ChEna	[31]	Channel Enable This field is set by the application and cleared by the OTG host. • 1'b0 : Disables Channel • 1'b1 : Enables Channel	R_WS_SC	1'b0
ChDis	[30]	Channel Disable The application sets this bit to stop transmitting/ receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	R_WS_SC	1'b0
OddFrm	[29]	Odd Frame This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro) frame. This field is applicable for only periodic transactions. • 1'b0 : Even (micro)frame • 1'b1 : Odd (micro)frame	R/W	1'b0
DevAddr	[28:22]	Device Address This field selects the specific device serving as the data source or sink.	R/W	7'h0
MC/EC	[21:20]	Multi Count/Error Count If the Split Enable bit of the Host Channel-n Split Control register is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this endpoint. • 2'b00 : Reserved • 2'b01 : 1 transaction • 2'b10 : 2 transactions to be issued for this endpoint per microframe • 2'b11 : 3 transactions to be issued for this endpoint per microframe If HCSPLTn.SpltEna is set, this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.	R/W	2'b0
EPType	[19:18]	Endpoint Type Indicates the transfer type selected. • 2'b00 : Control • 2'b01 : Isochronous • 2'b10 : Bulk • 2'b11 : Interrupt	R/W	2'b0
LSpdDev	[17]	Low-Speed Device This field is set by the application to indicate that this channel is	R/W	1'b0

HCCHARn	Bit	Description	R/W	Reset Value
		communicating to a low-speed device.		
Reserved	[16]	-		1'b0
EPDir	[15]	Endpoint Direction Endpoint Type Indicates the transfer type selected. • 1'b0 : OUT • 1'b1 : IN	R/W	1'b0
EPNum	[14:11]	Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.	R/W	4'h0
MPS	[10:0]	Maximum Packet Size Indicates the maximum packet size of the associated endpoint.	R/W	11'h0

8.2.28 Host Channel-n Split Register (HCSPLTn, R/W, Address = 0xED20_0504+n*20h)

Channel_number : $0 \leq n \leq 15$

HCSPLTn	Bit	Description	R/W	Reset Value
SpltEna	[31]	Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.	R/W	1'b0
Reserved	[30:17]	-		14'h0
CompSplt	[16]	Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.	R/W	1'b0
XactPos	[15:14]	Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. • 2'b11: All. This is the entire data payload is of this transaction. • 2'b10: Begin. This is the first data payload of this transaction. • 2'b00: Mid. This is the middle payload of this transaction. • 2'b01: End. This is the last payload of this transaction.	R/W	2'h0
HubAddr	[13:7]	Hub Address This field holds the device address of the transaction translator's hub.	R/W	7'h0
PrtAddr	[6:0]	Port Address This field is the port number of the recipient transaction translator.	R/W	7'h0

8.2.29 Host Channel-n interrupt Register (HCINTn, R/W, Address = 0xED20_0508+n*20h)

Channel_number : $0 \leq n \leq 15$

This register indicates the status of a channel with respect to USB- and AHB-related events. The application must read this register if the Host Channels Interrupt bit of the Core Interrupt register is set. Before the application reads this register, it must first read the Host All Channels Interrupt register to get the exact channel number for the Host Channel-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the HAINT and GINTSTS registers.

HCINTn	Bit	Description	R/W	Reset Value
Reserved	[31:11]	-		21'h0
DataTglErr	[10]	Data Toggle Error	R_SS_WC	1'b0
FrmOvrun	[9]	Frame Overrun	R_SS_WC	1'b0
BblErr	[8]	Babble Error	R_SS_WC	1'b0
XactErr	[7]	Transaction Error	R_SS_WC	1'b0
NYET	[6]	NYET Response Received Interrupt	R_SS_WC	1'b0
ACK	[5]	ACK Response Received Interrupt	R_SS_WC	1'b0
NAK	[4]	NAK Response Received Interrupt	R_SS_WC	1'b0
STALL	[3]	STALL Response Received Interrupt	R_SS_WC	1'b0
AHBErr	[2]	AHB Error This is generated only in Internal DMA mode if there is an AHB error during AHB read/ writes. The application reads the corresponding channel's DMA address register to get the error address.	R_SS_WC	1'b0
ChHltd	[1]	Channel Halted Indicates the incomplete transfer either because of any USB transaction error or in response to disable request by the application.	R_SS_WC	1'b0
XferCompl	[0]	Transfer Completed Transfer completed normally without any errors.	R_SS_WC	1'b0

8.2.30 Host Channel-n interrupt Mask Register (HCINTMSKn, R/W, Address = 0xED20_050C+n*20h)

Channel_number : 0 ≤ n ≤ 15

This register reflects the mask for each channel status described in the previous section.

- Mask interrupt : 1'b0
- Unmask interrupt : 1'b1

HCINTMSKn	Bit	Description	R/W	Reset Value
Reserved	[31:11]	-		21'h0
DataTglErrMsk	[10]	Data Toggle Error Mask	R/W	1'b0
FrmOvrunMsk	[9]	Frame Overrun Mask	R/W	1'b0
BblErrMsk	[8]	Babble Error Mask	R/W	1'b0
XactErrMsk	[7]	Transaction Error Mask	R/W	1'b0
NyetMsk	[6]	NYET Response Received Interrupt Mask	R/W	1'b0
AckMsk	[5]	ACK Response Received Interrupt Mask	R/W	1'b0
NakMsk	[4]	NAK Response Received Interrupt Mask	R/W	1'b0
StallMsk	[3]	STALL Response Received Interrupt Mask	R/W	1'b0
AHBErrMsk	[2]	AHB Error Mask	R/W	1'b0
ChHltdMsk	[1]	Channel Halted Mask	R/W	1'b0
XferComplMsk	[0]	Transfer Completed Mask	R/W	1'b0

8.2.31 Host Channel-n Transfer Size Register (HCTSIZn, R/W, Address = 0xED20_0510+n*20h)

Channel_number : 0 ≤ n ≤ 15

HCTSIZn	Bit	Description	R/W	Reset Value
DoPng	[31]	Do Ping Setting this field to 1 directs the host to do PING protocol.	R/W	1'h0
Pid	[30:29]	PID The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. • 2'b00: DATA0 • 2'b01: DATA1 • 2'b10: DATA2 • 2'b11: MDATA (non-control)/ SETUP(control)	R/W	2'b0
PktCnt	[28:19]	Packet Count This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission or reception of an OUT/ IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion.	R/W	10'b0
XferSize	[18:0]	Transfer Size For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions.	R/W	19'b0

NOTE: Transfer Size for a Host Channel must equal [Packet Count * Max Packet Size] for accurate data transfer.

8.2.32 Host Channel-n DMA Address Register (HCDMAn, R/W, Address = 0xED20_0514+n*20h)

Channel_number: 0 ≤ n ≤ 15

This register is used by the OTG host in the internal DMA mode to maintain the buffer pointer for IN/ OUT transactions.

HCDMAn	Bit	Description	R/W	Reset Value
DMAAddr	[31:0]	DMA Address This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction.	R/W	32'h0

DEVICE MODE REGISTERS

These registers are visible only in Device mode and must not be accessed in Host mode, as the results are unknown. Some of them affect all the endpoints uniformly, while others affect only a specific endpoint. Device Mode registers fall into two categories:

- Device Global registers
- Device logical endpoint-specific registers

8.2.33 Device Configuration Register (DCFG, R/W, Address = 0xED20_0800)

This register configures the core after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

DCFG	Bit	Description	R/W	Reset Value
Reserved	[31:23]	-		9'h0
EPMisCnt	[22:18]	IN Endpoint Mismatch Count The application programs this field with a count that determines when the core generates an Endpoint Mismatch interrupt. The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or if the counter expires. The width of this counter depends on the depth of the Token Queue.	R/W	5'h8
Reserved	[17:13]	-		5'h0
PerFrInt	[12:11]	Periodic Frame Interval Indicates the time within a (micro) frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro) frame is complete. • 2'b00 : 80% of the (micro) frame interval • 2'b01 : 85% • 2'b10 : 90% • 2'b11 : 95%	R/W	2'h0
DevAddr	[10:4]	Device Address The application must program this field after every SetAddress control command.	R/W	7'h0
Reserved	[3]	-		1'b0
NZSts OUTHShk	[2]	Non-Zero-Length Status OUT Handshake The application uses this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. • 1'b0: Sends a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. • 1'b1: Sends the received OUT packet to the application and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.	R/W	1'b0

DCFG	Bit	Description	R/W	Reset Value
DevSpd	[1:0]	Device Speed. Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application supports. However the actual bus speed is determined only after the chirp sequence is complete, and is based on the speed of the USB host to which the core is connected. • 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) • 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) • 2'b10: Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6 MHz LS mode, you must do a soft reset. • 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz).	R/W	2'b0

8.2.34 Device Control Register (DCTL, R/W, Address = 0xED20_0804)

DCTL	Bit	Description	R/W	Reset Value
Reserved	[31:12]	-		20'h0
PWROnPrgDone	[11]	Power-On Programming Done The application uses this bit to indicate that register programming is complete after a wake-up from Power Down mode.	R/W	1'b0
CGOUTNak	[10]	Clear Global OUT NAK A write to this field clears the Global OUT NAK.	W	1'b0
SGOUTNak	[9]	Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set this bit after making sure that the Global OUT NAK Effective bit in Core Interrupt Register is cleared.	W	1'b0
CGNPInNAK	[8]	Clear Global Non-Periodic IN NAK A write to this field clears the Global Non-Periodic IN NAK.	W	1'b0
SGNPInNAK	[7]	Set Global Non-Periodic IN NAK A write to this field sets the Global Non-Periodic IN NAK. The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core sets this bit if a timeout condition is detected on a non-periodic endpoint. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register is cleared.	W	1'b0
TstCtl	[6:4]	Test Control • 3'b000 : Test mode disabled • 3'b001 : Test_J mode • 3'b010 : Test_K mode • 3'b011 : Test_SE0_NAK mode • 3'b100 : Test_Packet mode • 3'b101 : Test_Force_Enable • Others : Reserved	R/W	3'b0
GOUTNakSts	[3]	Global OUT NAK Status • 1'b0 : A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. • 1'b1 : No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	R	1'b0

DCTL	Bit	Description	R/W	Reset Value
GNPINNakSts	[2]	Global Non-Periodic IN NAK Status • 1'b0 : A handshake is sent based on the data availability in the transmit FIFO. • 1'b1 : A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.	R	1'b0
SftDiscon	[1]	Soft Disconnect The application uses this bit to signal the OTG core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device will not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. • 1'b0 : Normal operation. If this bit is cleared after a soft disconnect, the core drives the opmode signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. If the device is reconnected, the USB host restarts device enumeration. • 1'b1 : The core drives the opmode signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.	R/W	1'b0
RmtWkUpSig	[0]	Remote Wakeup Signaling If the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1-15ms after setting it.	R/W	1'b0

The following table lists the minimum duration under various conditions for which the SoftDisconnect bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add some extra delay to the specified minimum duration.

Operating Speed	Device state	Minimum Duration
High speed	Suspended	1ms + 2.5μs
High speed	Idle	3ms + 2.5 μs
High speed	Not Idle or Suspended (Performing transactions)	125 μs
Full speed/ Low speed	Suspended	1ms + 2.5 μs
Full speed/ Low speed	Idle	2.5 μs
Full speed/ Low speed	Not Idle or Suspended (Performing transactions)	2.5 μs

8.2.35 Device Status Register (DSTS, R, Address = 0xED20_0808)

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from Device ALL Interrupts (DAINT) register.

DSTS	Bit	Description	R/W	Reset Value
Reserved	[31:22]	-		10'h0
SOFFN	[21:8]	Frame or Microframe Number of the Received SOF If the core is operating at high speed; this field contains a microframe number. If the core is operating at full or low speed, this field contains a frame number.	R	14'h0
Reserved	[7:4]	-		4'h0
ErrticErr	[3]	Erratic Error The core sets this bit to report any erratic errors seen on the UTMI+. Due to erratic errors, the OTG core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register. If the early suspend is asserted due to an erratic error, the application performs a soft disconnect recover.	R	1'b0
EnumSpd	[2:1]	Enumerated Speed Indicates the speed at which the OTG core has come up after speed detection through a chirp sequence. • 2'b00: High speed (PHY clock is 30 MHz or 60 MHz) • 2'b01: Full speed (PHY clock is 30 MHz or 60 MHz) • 2'b10: Low speed (PHY clock is 6 MHz). • 2'b11: Full speed (PHY clock is 48 MHz). Low speed is not supported for devices using a UTMI+ PHY.	R	2'b01
SuspSts	[0]	Suspend Status In device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state if there is no activity on the line_state signal for an extended period of time. The core comes out of the suspend: • If there is any activity on the line_state signal • If the application writes to the Remote Wakeup Signaling bit in the Device Control register.	R	1'b0

8.2.36 Device IN Endpoint Common Interrupt Mask Register (DIEPMSK, R/W, Address = 0xED20_0810)

This register works with each of the Device IN Endpoint Interrupt registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register is masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

DIEPMSK	Bit	Description	R/W	Reset Value
Reserved	[31:7]	-		25'h0
INEPNakEffMsk	[6]	IN Endpoint NAK Effective Mask	R/W	1'b0
INTknEPMisMsk	[5]	IN Token received with EP Mismatch Mask	R/W	1'b0
INTknTXFEmpMsk	[4]	IN Token received with TxFIFO Empty mask	R/W	1'b0
TimeOUTMsk	[3]	Timeout Condition Mask	R/W	1'b0
AHBErrMsk	[2]	AHB Error Mask	R/W	1'b0
EPDisbldMsk	[1]	Endpoint Disabled Interrupt Mask	R/W	1'b0
XferComplMsk	[0]	Transfer Completed Interrupt Mask	R/W	1'b0

8.2.37 Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK, R/W, Address = 0xED20_0814)

This register works with each of the Device OUT Endpoint Interrupt registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupts for a specific status in the DOEPINTn register is masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

DOEPMSK	Bit	Description	R/W	Reset Value
Reserved	[31:7]	-		27'h0
Back2BackSETup	[6]	Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.	R/W	1'b0
Reserved	[5]	-		1'b0
OUTTknEPdisMsk	[4]	OUT Token Received When Endpoint Disabled Applies to control OUT endpoints only.	R/W	1'b0
SetUPMsk	[3]	SETUP Phase Done Mask Applies to control endpoints only.	R/W	1'b0
AHBErrMsk	[2]	AHB Error	R/W	1'b0
EPDisbldMsk	[1]	Endpoint Disabled Interrupt Mask	R/W	1'b0
XferComplMsk	[0]	Transfer Completed Interrupt Mask	R/W	1'b0

8.2.38 Device ALL Endpoints Interrupt Register (DAINT, R, Address = 0xED20_0818)

If a significant event occurs on an endpoint, a Device All Endpoints Interrupt register interrupts the application using the Device OUT Endpoints Interrupt bit or Device IN Endpoints Interrupt bit of the Core Interrupt register. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared if the application sets and clears bits in the corresponding Device Endpoint – n Interrupt register.

DAINT	Bit	Description	R/W	Reset Value
OutEPInt	[31:16]	OUT Endpoint Interrupt Bits One bit per OUT endpoint : Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	R	16'h0
InEpInt	[15:0]	IN Endpoint Interrupt Bits One bit per IN endpoint : Bit 0 for IN endpoint 0, bit 15 for endpoint 15	R	16'h0

8.2.39 Device ALL Endpoints Interrupt Mask Register (DAINTMSK, R/W, Address = 0xED20_081C)

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application if an event occurs on a device endpoint. However, the Device all Endpoints Interrupt register bit corresponding to that interrupt remains set.

- Mask interrupt : 1'b0
- Unmask interrupt : 1'b1

DAINTMSK	Bit	Description	R/W	Reset Value
OutEPMsk	[31:16]	OUT EP Interrupt Mask Bits One bit per OUT endpoint : Bit 16 for OUT EP 0, bit 31 for OUT EP 15	R/W	16'h0
InEpMsk	[15:0]	IN EP Interrupt Mask Bits One bit per IN endpoint : Bit 0 for IN EP 0, bit 15 for IN EP 15	R/W	16'h0

8.2.40 Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1, R, Address = 0xED20_0820)

The queue is 4 bits wide to store the endpoint number. A read from this register returns the first 5 endpoint entries of the IN Token Sequence Learning Queue. If the Queue is full, the new token is pushed into the queue and oldest token is discarded.

DTKNQR1	Bit	Description	R/W	Reset Value
EPTkn	[31:8]	Endpoint Token Four bits per token represent the endpoint number of the token : • Bits [31:28] : Endpoint number of Token 5 • Bits [27:24] : Endpoint number of Token 4 • • • • Bits [15:12] : Endpoint number of Token 1 • Bits [11:8] : Endpoint number of Token 0	R	24'h0
WrapBit	[7]	Wrap Bit This bit is set if the write pointer wraps. It is cleared if the learning queue is cleared.	R	1'b0
Reserved	[6:5]	-	R	2'h0
INTKnWPtr	[4:0]	IN Token QUEUE Write Pointer	R	5'h0

8.2.41 Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2, R, Address = 0xED20_0824)

Read from this register returns the next 8 endpoint entries of the learning queue.

DTKNQR2	Bit	Description	R/W	Reset Value
EPTkn	[31:0]	Endpoint Token Four bits per token represent the endpoint number of the token: • Bits [31:28] : Endpoint number of Token 13 • Bits [27:24] : Endpoint number of Token 12 • • • • Bits [7:4] : Endpoint number of Token 7 • Bits [3:0] : Endpoint number of Token 6	R	32'h0

8.2.42 Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3, R, Address = 0xED20_0830)

Read from this register returns the next 8 endpoint entries of the learning queue.

DTKNQR3	Bit	Description	R/W	Reset Value
EPTkn	[31:0]	Endpoint Token Four bits per token represent the endpoint number of the token: • Bits [31:28]: Endpoint number of Token 21 • Bits [27:24] : Endpoint number of Token 20 • • • • Bits [7:4] : Endpoint number of Token 15 • Bits [3:0] : Endpoint number of Token 14	R	32'h0

8.2.43 Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4, R, Address = 0xED20_0834)

Read from this register returns the next 8 endpoint entries of the learning queue.

DTKNQR4	Bit	Description	R/W	Reset Value
EPTkn	[31:0]	Endpoint Token Four bits per token represent the endpoint number of the token: • Bits [31:28] : Endpoint number of Token 29 • Bits [27:24] : Endpoint number of Token 28 • • • • Bits [7:4] : Endpoint number of Token 23 • Bits [3:0] : Endpoint number of Token 22	R	32'h0

8.2.44 Device VBUS Discharge Time Register (DVBUSDIS, R/W, Address = 0xED20_0828)

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

DVBUSDIS	Bit	Description	R/W	Reset Value
Reserved	[31:16]	-		16'h0
DVBUSDis	[15:0]	Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals : VBUS discharge time in PHY clocks /1,024	R/W	16'h17D7

8.2.45 Device VBUS Pulsing Time Register (DVBUSPULSE, R/W, Address = 0xED20_082C)

This register specifies the VBUS discharge time during SRP.

DVBUSPULSE	Bit	Description	R/W	Reset Value
Reserved	[31:12]	-		16'h0
DVBUSPulse	[11:0]	Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals : VBUS pulse time in PHY clocks /1,024	R/W	12'h5B8

8.2.46 Device Logical Endpoint-Specific Registers

A logical endpoint is unidirectional: it is either IN or OUT. To represent a bidirectional endpoint, two logical endpoints are required, one for the IN direction and the other for the OUT direction. This is also true for control endpoints. The registers and register fields described in this section may pertain to IN or OUT endpoints, or both, or specific endpoint types are noted.

8.2.47 Device Control IN Endpoint 0 Control Register (DIEPCTL0, R/W, Address = 0xED20_0900)

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

DIEPCTL0	Bit	Description	R/W	Reset Value
EPEna	[31]	Endpoint Enable Indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting any of the following interrupts on this endpoint. • Endpoint Disabled • Transfer Completed	R_WS_SC	1'b0
EPDis	[30]	Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	R_WS_SC	1'b0
Reserved	[29:28]	-		2'b0
SetNAK	[27]	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core also sets this bit for an endpoint after a SETUP packet is received on that endpoint.	W	1'b0
CNAK	[26]	Clear NAK A write to this bit clears the NAK bit for the endpoint.	W	1'b0
TxFNum	[25:22]	TxFIFO Number This value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO.	R	4'h0
Stall	[21]	STALL Handshake The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.	R_WS_SC	1'b0
Reserved	[20]	-		1'b0
EPType	[19:18]	Endpoint Type Hardcoded to 00 for control	R	2'h0

DIEPCTL0	Bit	Description	R/W	Reset Value
NAKsts	[17]	NAK Status Indicates the following: • 1'b0 : The core is transmitting non-NAK handshakes based on the FIFO status • 1'b1 : The core is transmitting NAK handshakes on this endpoint If this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	R	1'b0
Reserved	[16]	-		1'b0
USBActEP	[15]	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	R	1'b1
NextEp	[14:11]	Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core accesses this field, even if the Endpoint Enable bit is not set. This field is not valid in Slave mode operation.	R/W	4'b0
Reserved	[10:2]	-		9'h0
MPS	[1:0]	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. • 2'b00 : 64 bytes • 2'b01 : 32 bytes • 2'b10 : 16 bytes • 2'b11 : 8 bytes	R/W	2'h0

8.2.48 Device Control OUT Endpoint 0 Control Register (DOEPCTL0, R/W, Address =0xED20_0B00)

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

DOEPCTL0	Bit	Description	R/W	Reset Value
EPEna	[31]	Endpoint Enable Indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint. • SETUP Phase Done • Endpoint Disabled • Transfer Complete **Note:** In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.	R_WS_SC	1'b0
EPDis	[30]	Endpoint Disable The application cannot disable control OUT endpoint 0.	R	1'b0
Reserved	[29:28]	-		2'b0
SetNAK	[27]	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core sets this bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	W	1'b0
CNAK	[26]	Clear NAK A write to this bit clears the NAK bit for the endpoint.	W	1'b0
Reserved	[25:22]	-		4'h0
Stall	[21]	STALL Handshake The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	R_WS_SC	1'b0
Snp	[20]	Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	R/W	1'b0
EPType	[19:18]	Endpoint Type Hardcoded to 2'b00 for control.	R	2'h0

DOEPCTL0	Bit	Description	R/W	Reset Value
NAKsts	[17]	NAK Status Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status • 1'b1: The core is transmitting NAK handshakes on this endpoint If application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake	R	1'b0
Reserved	[16]	-		1'b0
USBActEP	[15]	USB Active Endpoint This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.	R	1'b1
Reserved	[14:2]	-		13'h0
MPS	[1:0]	Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. • 2'b00: 64 bytes • 2'b01 : 32 bytes • 2'b10 : 16 bytes • 2'b11 : 8 bytes	R	2'h0

8.2.49 Device Endpoint-N Control Register (DIEPCTLn/DOEPCTLn, R/W, Address = 0xED20_0900+ n*20h, 0xED20_0B00+ n*20h)

Endpoint_number :1 ≤ n ≤ 15

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

DIEPCTLn/ DOEPCTLn	Bit	Description	R/W	Reset Value
EPEna	[31]	Endpoint Enable Applies to IN and OUT endpoints. For IN endpoint, this bit indicates that data is ready to be transmitted on the endpoint. For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint : • SETUP Phase Done (OUT only) • Endpoint Disabled • Transfer Complete Transfer Completed **Note:** For control OUT endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.	R_WS _SC	1'b0
EPDis	[30]	Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/ receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	R_WS _SC	1'b0
SetD1PID	[29]	Set DATA1 PID Applies to interrupt/ bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.	W	1'b0
SetOddFr		Set Odd (micro) frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/ Odd (micro) frame field to odd (micro)frame.		
SetD0PID	[28]	Set DATA0 PID Applies to interrupt/ bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.	W	1'b0
SetEvenFr		Set Even (micro) frame Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/ Odd (micro) frame field to even (micro) frame.		
SNAK	[27]	Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application controls the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on	W	1'b0

DIEPCTLn/ DOEPCTLn	Bit	Description	R/W	Reset Value
		a Transfer Complete interrupt, or after a SETUP packet is received on that endpoint.		
CNAK	[26]	Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.	W	1'b0
TxFNum	[25:22]	TxFIFO Number Applies to IN endpoints only. Non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. • 4'h0 : Non-Periodic TxFIFO • Others : Specified Periodic TxFIFO number **Note:** An interrupt IN endpoint could be configured as a non-periodic endpoint for applications like mass storage.	R/W	4'h0
Stall	[21]	STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-Periodic In NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application clears this bit, never the core.	R/W	1'b0
		Applies to control endpoints only The application sets this bit, and the core clears it, if a SETUP token is received for this endpoint. If a NAK bit, Global Non-Periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	R_WS_SC	
Snp	[20]	Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	R/W	1'b0
EPType	[19:18]	Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. • 2'b00 : Control • 2'b01 : Isochronous • 2'b10 : Bulk • 2'b11 : Interrupt	R	2'h0
NAKsts	[17]	NAK Status Applies to IN and OUT endpoints. Indicates the following: • 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status • 1'b1: The core is transmitting NAK handshakes on this endpoint. If either the application or the core sets this bit:	R	1'b0

DIEPCTLn/ DOEPCTLn	Bit	Description	R/W	Reset Value
		• The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. • For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. • For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.		
DPID	[16]	Endpoint Data PID Applies to interrupt/ bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. Applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. • 1'b0 : DATA0 • 1'b1 : DATA1	R	1'b0
EO_FrNum		Even/ Odd (Micro) Frame Applies to isochronous IN and OUT endpoints only. Indicates the (micro) frame number in which the core transmits/ receives isochronous data for this endpoint. The application must program the even/ odd (micro) frame number in which it intends to transmit/ receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. • 1'b0 : Even (micro)frame • 1'b1 : Odd (micro)frame		
USBActEP	[15]	USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.	R_W_SC	1'b0
NextEp	[14:11]	Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core accesses this field, even if the Endpoint Enable bit is low. This field is not valid in Slave mode operation.	R/W	4'h0
MPS	[10:0]	Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.	R/W	11'h0

8.2.50 Device Endpoint-n Interrupt Register (DIEPINTn/DOEPINTn, R/W, Address = 0xED20_0908 +n*20h, 0xED20_0B08 +n*20h)

Endpoint_number : 0 ≤ n ≤ 15

This register indicates the status of an endpoint with respect to USB- and AHB-related events. The application must read this register if the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register is set. Before the application reads this register, it must first read the Device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the Device Endpoint-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINT and GINTSTS registers.

DIEPINTn/ DOEPINTn	Bit	Description	R/W	Reset Value
EPEna	[31:7]	Reserved		25'h1
INEPNakEff	[6]	IN Endpoint NAK Effective Applies to periodic IN endpoints only. Indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This bit is cleared if the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set. This interrupt does not necessarily mean that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.	R	1'b0
Back2Back SETup		Back-to-Back SETUP Packets Received Applies to Control OUT endpoints only. This bit indicates that core has received more than three back-to-back SETUP packets for this particular endpoint.	R/W	
INTknEPMis	[5]	IN Token Received with EP Mismatch Applies to periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. For OUT endpoints, this bit is reserved.	R_SS_WC	1'b0
INTknTXFEmp	[4]	IN Token Received When TxFIFO is Empty Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO was empty. This interrupt is asserted on the endpoint for which the IN token was received.	R_SS_WC	1'b0
OUTTknEPdis		OUT Token Received When Endpoint Disabled Applies only to control OUT endpoints. Indicates that an OUT token is received if the endpoint is not enabled. This interrupt is asserted on the endpoint for which the OUT token was received.		
TimeOUT	[3]	Timeout Condition Applies to non-isochronous IN endpoints only. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint.	R_SS_WC	1'b0

DIEPINTn/ DOEPINTn	Bit	Description	R/W	Reset Value
SetUp		SETUP Phase Done Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application decodes the received SETUP data packet.		
AHBErr	[2]	AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode if there is an AHB error during an AHB read/write. The application reads the corresponding endpoint DMA address register to get the error address.	R_SS_WC	1'b0
EPDisbld	[1]	Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.	R_SS_WC	1'b0
XferCompl	[0]	Transfer Completed Interrupt Applies to IN and OUT endpoints. Indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.	R_SS_WC	1'b0

8.2.51 Device Endpoint 0 Transfer Size Register (DIEPTSIZ0, R/W, Address = 0xED20_0910)

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the Device Control Endpoint 0 Control registers, the core modifies this register. The application reads this register after the core has cleared the Endpoint Enable bit.

DIEPTSIZ0	Bit	Description	R/W	Reset Value
Reserved	[31:21]	-		11'h0
PktCnt	[20:19]	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet is read from the TxFIFO.	R/W	2'b0
Reserved	[18:7]	-		12'h0
XferSize	[6:0]	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to the TxFIFO.	R/W	7'h0

8.2.52 Device OUT Endpoint 0 Transfer Size Register (DOEPTSIZ0, R/W, Address = 0xED20_0B10)

DOEPTSIZ0	Bit	Description	R/W	Reset Value
Reserved	[31]	-		1'b0
SUPCnt	[30:29]	SETUP Packet Count This field specifies the number of back-to-back SETUP data packets the endpoint can receive. • 2'b01 : 1 packet • 2'b10 : 2 packets • 2'b11 : 3 packets	R/W	2'h0
Reserved	[28:20]	-		9'h0
PktCnt	[19]	Packet Count This field is decremented to zero after a packet is written into the RxFIFO.	R/W	1'b0
Reserved	[18:7]	-		12'h0
XferSize	[6:0]	Transfer Size Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from RxFIFO and written to the external memory.	R/W	7'h0

8.2.53 Device Endpoint-N Transfer Size Register (DIEPTSIZn/DOEPTSIZn, R/W, Address = 0xED20_0910 +n*20h, 0xED20_0B10 +n*20h)

Endpoint_number: $1 \leq n \leq 15$

The application must modify this register before enabling the endpoint. Once the endpoint is enabled using endpoint Enable bit of the Device Endpoint-n Control registers, the core modifies this register. The application reads this register once the core has cleared the Endpoint Enable bit. This register is used only for endpoints other than Endpoint 0.

DIEPTSIZn/ DOEPTSIZn	Bit	Description	R/W	Reset Value
Reserved	[31]	-		1'b0
MC	[30:29]	Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. • 2'b01 : 1 packet • 2'b10 : 2 packets • 2'b11 : 3 packets	R/W	2'b0
		For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register.	R	
RxDPID		Received Data PID Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. • 2'b00 : DATA0 • 2'b01 : DATA1 • 2'b10 : DATA2 • 2'b11 : MDATA	R	
SUPCnt		SETUP Packet Count Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. • 2'b01: 1 packet • 2'b10: 2 packets • 2'b11: 3 packets	R/W	
PktCnt	[28:19]	Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. • IN Endpoints: This field is decremented every time a packet is read from the TxFIFO • OUT Endpoints: This field is decremented every time a packet is written to the RxFIFO	R/W	10'h0

XferSize	[18:0]	Transfer Size This field contains the transfer size in bytes for the current endpoint. The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size is set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. • IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. • OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	R/W	19'h0

NOTE: Transfer Size for a Device Endpoint must equal [Packet Count * Max Packet Size] for accurate data transfer.

8.2.54 Device Endpoint-N DMA Address (DIEPDMAn/DOEPDMAn, R/W, Address = 0xED20_0914 +n*20h, 0xED20_0B14 +n*20h)

Endpoint_number : $0 \leq n \leq 15$

The starting DMA address must be DWORD-aligned.

DIEPDMAn/ DOEPDMAn	Bit	Description	R/W	Reset Value
DMAAddr	[31:0]	DMA Address Holds the start address of the external memory for storing or fetching endpoint data. This register is incremented on every AHB transaction. **Note:** For control endpoints, this address stores control OUT data packets as well as SETUP transaction data packets. If multiple SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten.	R/W	32'h0

8.2.55 Power and Clock Gating control register (PCGCCTL, R/W, Address = 0xED20_0E00)

The application uses this register to control OTG's clock gating.

DIEPTSIZ0	Bit	Description	R/W	Reset Value
Reserved	[31:1]	-		31'h0
StopPclk	[0]	STOP Pclk The application sets this bit to stop the PHY clock if the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit if the USB is resumed or a new session starts.	R/W	1'b0

8.11 MODEM INTERFACE

1 OVERVIEW

This specification defines the interface between the Base-band Modem (like MSM) and the Application Processor for the data-exchange of these two devices (Refer Figure 8.11-1). For the data-exchange, the AP (Application Processor, S5PC100) has a dual-ported SRAM buffer (on-chip). The Modem chip accesses SRAM buffer using a typical asynchronous-SRAM interface.

The size of the SRAM buffer is 16 KB. For the buffer status and Interrupt Requests, this specification also specifies a few pre-defined special addresses.

The Modem chip writes data in the data buffer and writes interrupt control-data to the interrupt-port address for the interrupt request to the AP. The AP reads that data if an interrupt request is accepted and the interrupt is cleared if AP accesses the interrupt-port address. In the same manner, AP writes data in the data buffer and writes interrupt control-data to the interrupt-port address for interrupt request to the Modem chip.

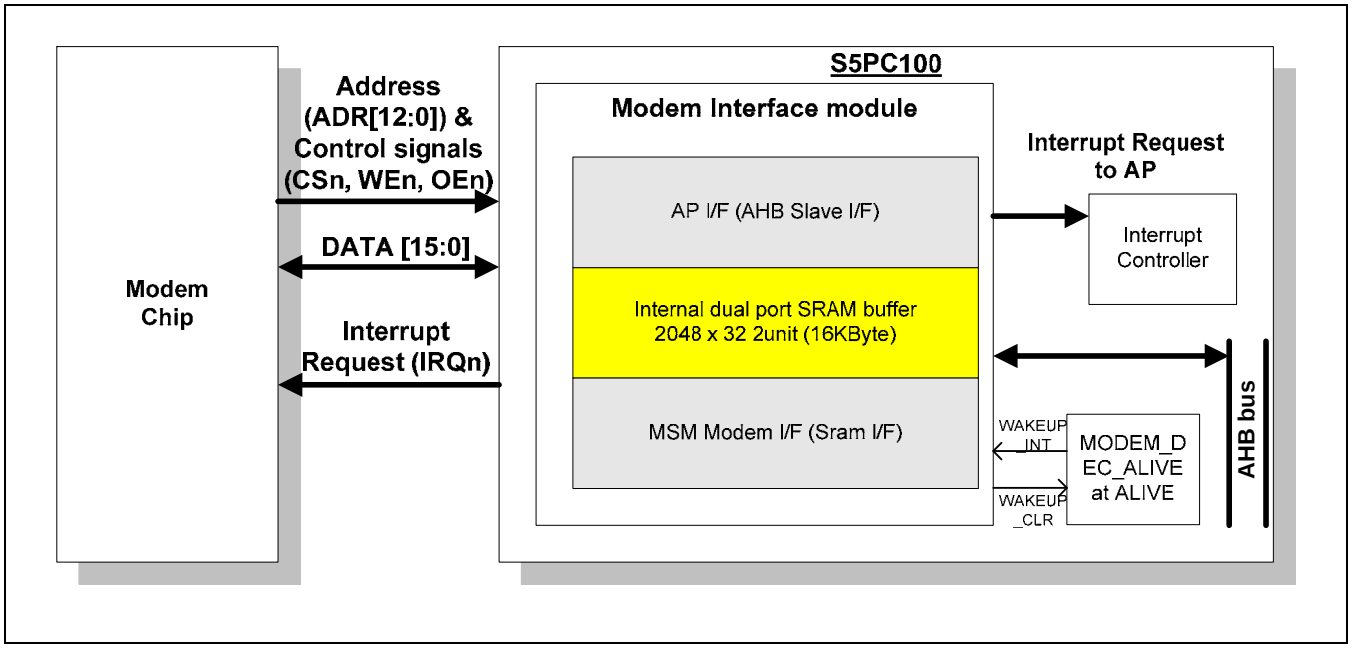


Figure 8.11-1 Interface with the Modem Chip and the MODEM I/F Block Diagram

2 FEATURES

- Asynchronous SRAM interface style interface
- Supports both Standard mode and Address Muxed mode
- Supports 16-bit parallel bus for data transfer
- Supports 16 KB internal dual-port SRAM buffer
- Supports Interrupt request for data exchange
- Programmable interrupt port address
- Supports DMA for data transfer without intervention of CPU

3 INTERRUPT PORTS

Interrupts are requested or cleared if the Modem chip or the AP accesses the interrupt-port (predefined special addresses). That special address is configured by the AP and the default address-map is shown in the Table 8.11-1.

Table 8.11-1 Interrupt Request and Clear Conditions

Interrupt	An Interrupt is requested, when	The Interrupt is cleared, when
To AP	Modem chip writes at least 1 to 0x1FFF through ADR.	AP writes at least 1 to MSMINTCLR register in MODEM IF (**NOTE 2**).
To Modem	AP writes 1 to 0xED50_3FFC through internal-chip AHB bus.	Modem chip writes 1 to the bits at 0x1FFE through ADR.

NOTE:

1. There are two address views for MODEMIF, one is MSM address (ADR) for MODEM chip, and the other one is AHB address for S5PC100. AHB address is twice of ADR. For example, 0x3FFC at AHB bus is 0x1FFE at ADR.
   Figure 8.11-2 helps you to understand it.
   This is default value. It can be set to other value by the SFR. (INT2AP and INT2MSM)
2. Modem interface block has one Interrupt Clear Registers; MSMINTCLR. Level type interrupt request is generated by modem interface block and is sustained until the AP clears the interrupt clear registers by writing any value to the registers.

Modem chip or AP (S5PC100) reads the data that indicates what event happens – data transfer requested, data transfer done, special command issued, etc. – from interrupt port address. That data format should be defined for communication between the modem chip and AP.

3.1 WAKEUP

* Standard mode

To assert wakeup at IDLE/ STOP mode, write **0x1100~0x117F** to ADR[12:0]
To clear wakeup signal, write **0x1180~0x11FF** to ADR[12:0]

* Address Muxed mode

To assert wakeup at IDLE/STOP mode, assert chip select (XmsmCsn) and address valid (XmsmADDR[4])
To clear wakeup signal, assert chip select (XmsmCsn) and address valid (XmsmADDR[4]) again

4 ADDRESS MAPPING

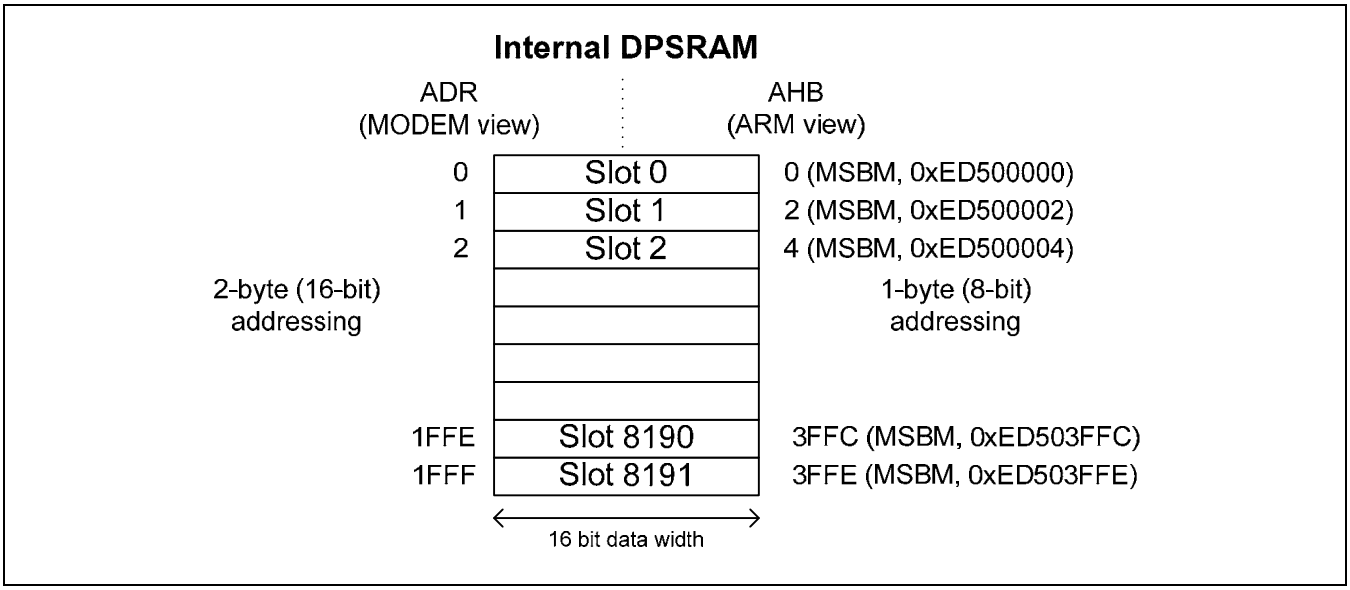


Figure 8.11-2 MODEM I/F Address Mapping

5 TIMING DIAGRAM

5.1 STANDARD MODE WRITE, READ TIMING

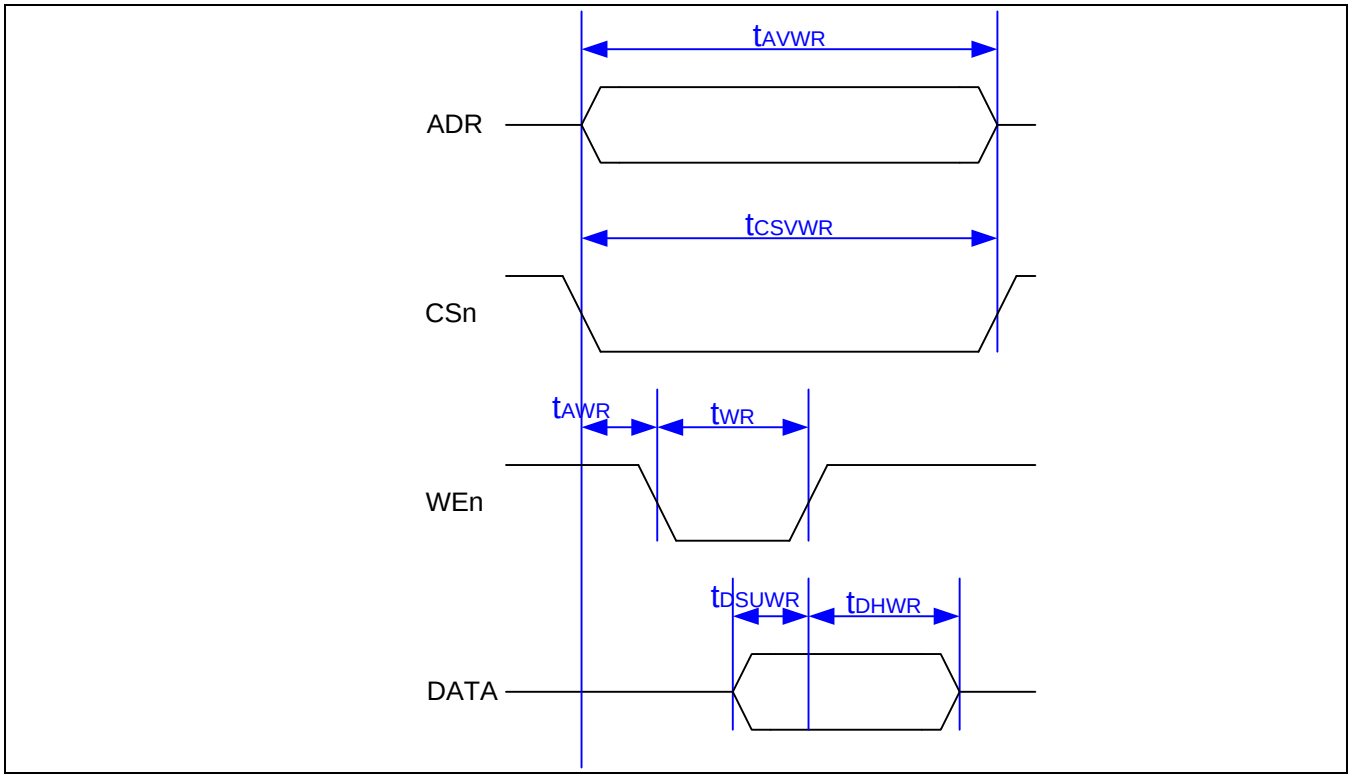


Figure 8.11-3 Modem Interface Write Timing Diagram (Standard Mode)

Table 8.11-2 Modem Interface Write Timing (Standard mode)

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVWR}	Address Valid to Address Invalid	16 ns	-	-
t_{CSVWR}	Chip Select Active	16 ns	-	-
t_{AWR}	Address Valid to Write Active	4 ns	-	-
t_{WR}	Write Active	8 ns	-	-
t_{DSUWR}	Write Data Setup	8 ns	-	-
t_{DHWR}	Write Data Hold	4 ns	-	-

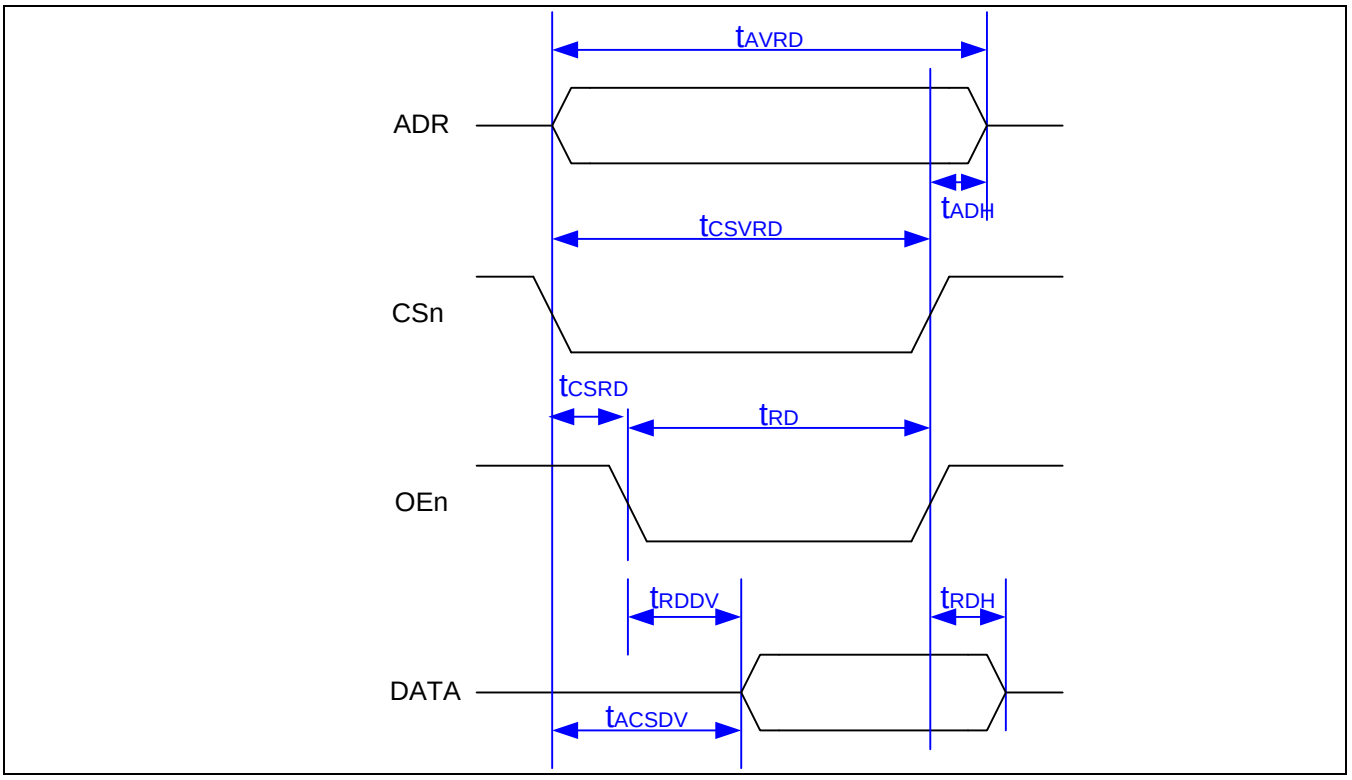


Figure 8.11-4 Modem Interface Read Timing Diagram (Standard Mode)

Table 8.11-3 Modem interface read timing (Standard Mode)

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVRD}	Address Valid to Address Invalid	50 ns	-	-
t_{ADH}	Address Hold	0 ns	-	-
t_{CSVRD}	Chip Select Active	50 ns	-	-
t_{CSRD}	Chip Select Active to Read Active	14 ns	-	-
t_{RD}	Read Active	36 ns	-	-
t_{RDDV}	Read Active to Data Valid	-	35 ns	-
t_{RDH}	Read Data Hold	6 ns	-	-
t_{ACSDV}	Address and Chip Select Active to Data Valid	-	49 ns	-

NOTE: Output load is 30pF at room temperature (25 Degree)

5.2 ADDRESS MUXED MODE WRITE, READ TIMING

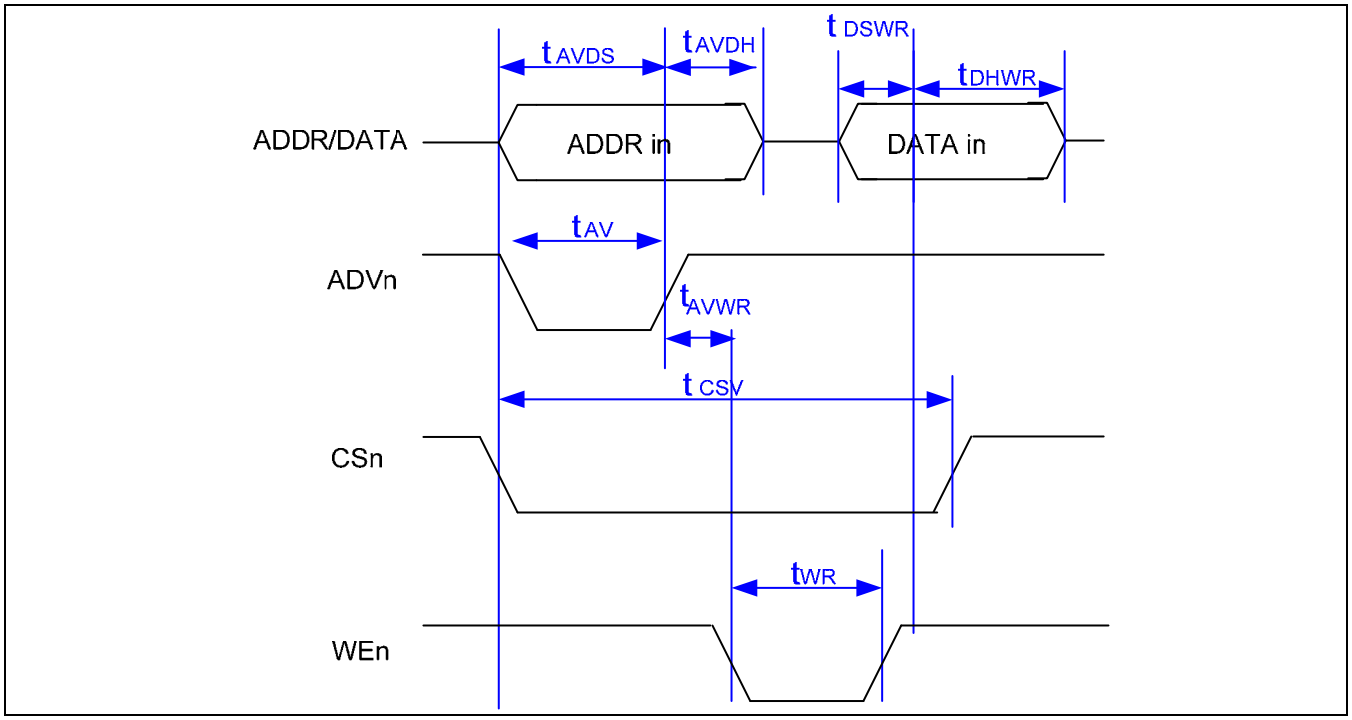


Figure 8.11-5 Modem Interface Write Timing Diagram (Address Muxed Mode)

Table 8.11-4 Modem Interface Write Timing (Address Muxed Mode)

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVDS}	Address Valid Setup	15 ns	-	-
t_{AVDH}	Address Valid Hold	5 ns	-	-
t_{AV}	Address Valid duration	15 ns	-	-
t_{AVWR}	Address Valid to Write Enable	0 ns	-	-
t_{CSV}	Chip Select duration	20 ns	-	-
t_{WR}	Write Enable duration	5 ns	-	-
t_{DSWR}	Write Data Setup	8 ns	-	-
t_{DHWR}	Write Data Hold	4 ns	-	-

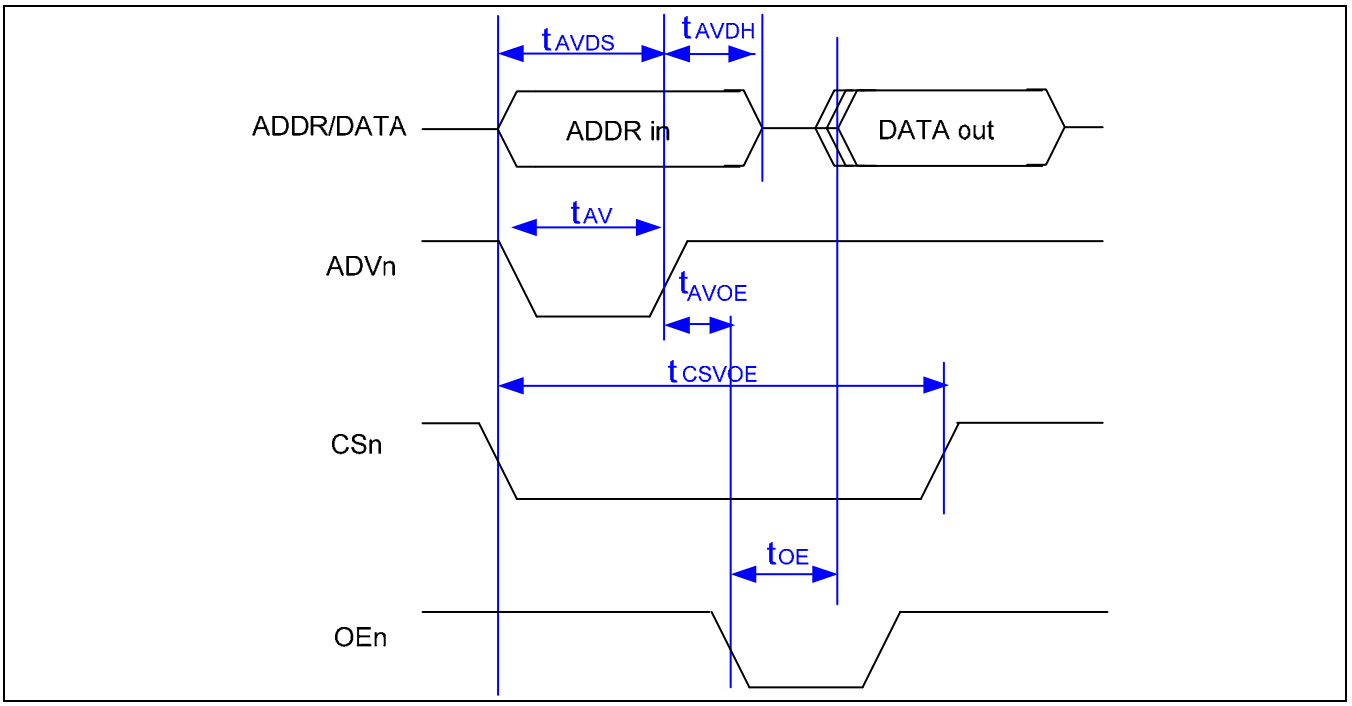


Figure 8.11-6 Modem Interface Read Timing Diagram (Address Muxed Mode)

Table 8.11-5 Modem interface read timing (Address Muxed Mode)

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVDS}	Address Valid Setup	15 ns	-	-
t_{AVDH}	Address Valid Hold	5 ns	-	-
t_{AV}	Address Valid duration	15 ns	-	-
t_{AVOE}	Address Valid to Read Enable	5 ns	-	-
t_{CSVOE}	Chip Select duration (Read mode)	45 ns	-	-
t_{OE}	Output Enable(Read Active) to Data Valid	-	35 ns	-

NOTE: Output load is 30pF at room temperature (25 Degree)

6 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
MSM_A[12:0]	Input	Address from MODEM Chip	XmsmADDR[12:0]	muxed
MSM_CSn	Input	Chip Select Signal from MODEM Chip	XmsmCSn	muxed
MSM_WEn	Input	Write Enable Signal from MODEM Chip	XmsmWEn	muxed
MSM_REn	Input	Read Enable Signal from MODEM Chip	XmsmRn	muxed
MSM_D[15:0]	In/Out	Data from/to MODEM Chip	XmsmDATA[15:0]	muxed
MSM_IRQn	Output	Interrupt Request to MODEM Chip	XmsmIRQn	muxed
ADVn	Input	Address Valid from MODEM Chip (Only Address Muxed mode)	XmsmADDR[4]	muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

7 SOFTWARE INTERFACE AND REGISTERS

This modem interface provides a generic data-exchange method. This interface does not implement any other complex features except for the interrupt-request/ clear such as automatic FIFO managements, etc. The software should be responsible for all the other required functionalities for the data exchange between a modem chip and the AP such as data exchange protocol, data buffer managements, etc.

8 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
MSBM	0xED50_0000 ~ 0xED50_3FFC	R/W	MODEM I/F SRAM Buffer Memory (AP side)	-

Register	Address	R/W	Description	Reset Value
INT2AP	0xED50_8000	R/W	Interrupt Request to AP Register	0x00003FFE
INT2MSM	0xED50_8004	R/W	Interrupt Request to MSM Modem Register	0x00003FFC
MIFCON	0xED50_8008	R/W	Modem Interface Control Register	0x00000008
MIFPCON	0xED50_800C	R/W	Modem Interface Port Control register	0x00000000
MSMINTCLR	0xED50_8010	W	MSM Modem Interface Pending Interrupt Request Clear	-
DMA_TX_ADR	0xED50_8014	R/W	DMA TX Request Address Register	0x17FE_13FE
DMA_RX_ADR	0xED50_8018	R/W	DMA RX Request Address Register	0x1FFE_1BFE

8.1 INTERRUPT REQUEST TO AP REGISTER (INT2AP, R/W, ADDRESS = 0XED50_8000)

INT2AP	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0
INT2AP_ADR	[13:0]	Modem interface requests the interrupt to AP if modem chip writes this address. This interrupt is cleared by the interrupt controller of AP and write access to the MSMINTCLR register.	0x3FFE

8.2 INTERRUPT REQUEST TO MODEM REGISTER (INT2MSM, R/W, ADDRESS = 0XED50_8004)

INT2MSM	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0
INT2MSM_ADR	[13:0]	Modem interface (in this module) requests the interrupt to modem chip if AP writes this address and clears the interrupt if modem chip write 1 to the corresponding bit field.	0x3FFC

NOTE: It is recommended that AP writes data with half-word access on the interrupt port because AP overwrites the data in INT2AP if there are INT2AP and INT2MSM sharing the same word.

8.3 MODEM INTERFACE CONTROL REGISTER (MIFCON, R/W, ADDRESS = 0XED50_8008)

MIFCON	Bit	Description	Reset Value
Reserved	[31:21]	Reserved	0
Fixed	[20]	**Shoud write as 1**	0
DMARXREQEN_1	[19]	Enables MSM Write DMA Request (RX 1) to AP (DMA Controller)	0
DMARXREQEN_0	[18]	Enables MSM Write DMA Request (RX 0) to AP (DMA Controller)	0
DMATXREQEN_1	[17]	Enables MSM Read DMA Request (TX 1) to AP (DMA Controller)	0
DMATXREQEN_0	[16]	Enables MSM Read DMA Request (TX 0) to AP (DMA Controller)	0
Reserved	[15:4]	Reserved	0
INT2MSMEN	[3]	Enables Interrupt to MSM (Modem) : MSM_nIRQ is interrupt signal enable. '0' = Disable, '1' = Enable	1
INT2APEN	[2]	Enables MSM (Modem) write interrupt to AP '0' = Disable, '1' = Enable	0
Reserved	[1]	Reserved	0
Fixed	[0]	Fixed to 0	0

8.4 MODEM INTERFACE PORT CONTROL REGISTER (MIFPCON, R/W, ADDRESS = 0XED50_800C)

MIFCON	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
ADM_MODE	[6]	Address Muxed mode selection '0' = Disable, '1' = Enable Note) XmsmADDR[4] is mapped to XmsmADVn	0
ADM_WAKE_EN	[5]	Address Muxed mode Wakeup Enable '0' = Disable, '1' = Enable	0
INT2M_LEVEL	[4]	Interrupt to MSM(Modem) Active High MSM_nIRQ interrupt signal make active high if this bit is set to High. '0' = Disable, '1' = Enable	0
Fixed	[3:0]	Fixed to 0	0

8.5 MSM INTERRUPT CLEAR REGISTER (MSMINTCLR, W, ADDRESS = 0XED50_8010)

MSMINTCLR	Bit	Description	Reset Value
MSMINTCLR	[31:0]	Write access to this register with any data will clear the interrupt pending register of MSM modem interface.	-

NOTE: The interrupt controllers of AP (S5PC100) receive level-triggered type interrupt requests. Therefore, interrupt requests from MSM interface are maintained until the interrupt service routine clears the interrupt pending register by writing any data into this register.

8.6 DMA REQUEST TX ADDRESS REGISTER USED AT DIRECT MODE (DMAREQ_TX_ADR, R/W, ADDRESS = 0XED50_8014)

INT2AP	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	0
DMA_TX_ADR_1	[29:16]	Modem interface requests the DMA to AP (DMA Controller) if modem chip **reads** this address. Source : DMA_MSM_Req[1]	0x17FE
Reserved	[15:14]	Reserved	0
DMA_TX_ADR_0	[13:0]	Modem interface requests the DMA to AP (DMA Controller) if modem chip **reads** this address. Source : DMA_MSM_Req[0]	0x13FE

8.7 DMA REQUEST RX ADDRESS REGISTER USED AT DIRECT MODE (DMAREQ_RX_ADR, R/W, ADDRESS = 0XED50_8018)

INT2AP	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	0
DMA_RX_ADR_1	[29:16]	Modem interface requests the DMA to AP (DMA Controller) if modem chip **writes** this address. Source : DMA_MSM_Req[3]	0x1FFE
Reserved	[15:14]	Reserved	0
DMA_RX_ADR_0	[13:0]	Modem interface requests the DMA to AP (DMA Controller) if modem chip **writes** this address. Source : DMA_MSM_Req[2]	0x1BFE

NOTES

8.12 SD/MMC CONTROLLER

This chapter describes the Secure Digital (SD/ SDIO), MultiMediaCard (MMC), CE-ATA host controller and related registers supported by S5PC100X RISC microprocessor.

1 OVERVIEW

The SD/ MMC host controller is a combo host for Secure Digital card and MultiMediaCard. This host is based on SD Association's (SDA) Host Standard Specification.

You can interface your system with SD card and MMC card. The performance of this host is very powerful, as clock rate is 48-MHz and access 8-bit data pin simultaneously.

2 FEATURES

The High-Speed MMC controller supports:

- SD Standard Host Specification Version 2.0 standard
- SD Memory Card Specification Version 2.0 / High Speed MMC Specification Version 4.2 standard
- SDIO Card Specification Version 1.0 standard
- 512 bytes FIFO for data Tx/ Rx
- CPU Interface and DMA data transfer mode
- 1-bit / 4-bit / 8-bit mode switch
- Auto CMD12
- Suspend/ Resume
- Read Wait operation
- Card Interrupt
- CE-ATA mode

3 BLOCK DIAGRAM

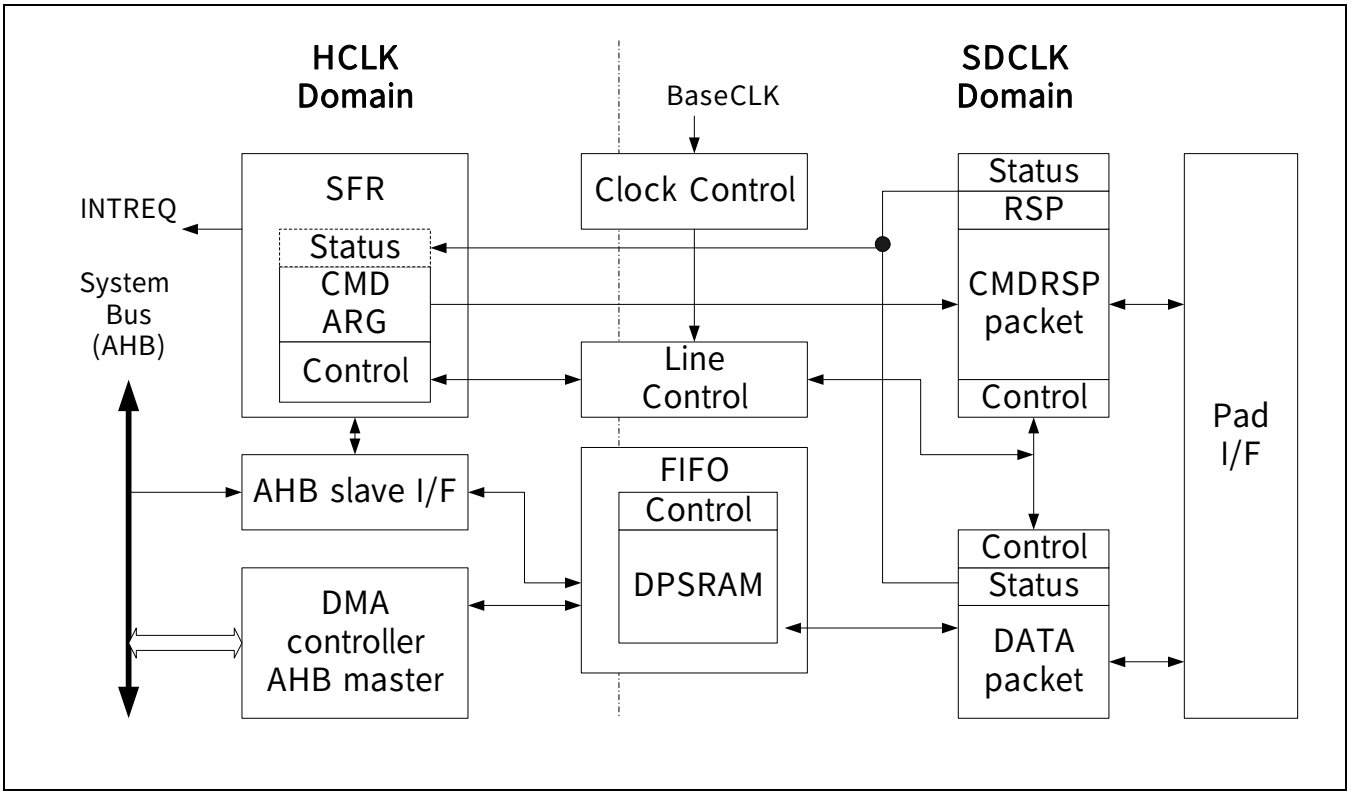


Figure 8.12-1 HSMMC Block Diagram

4 SEQUENCE

This section defines basic sequence flow chart divided into several sub sequences. "Wait for interrupts" is used in the flow chart. This means the Host Driver waits until specified interrupts are asserted. If already asserted, then follow the next step in the flow chart. Timeout checking required to detect no interrupt generated. This is not described in the flow chart.

4.1 SD CARD DETECTION SEQUENCE

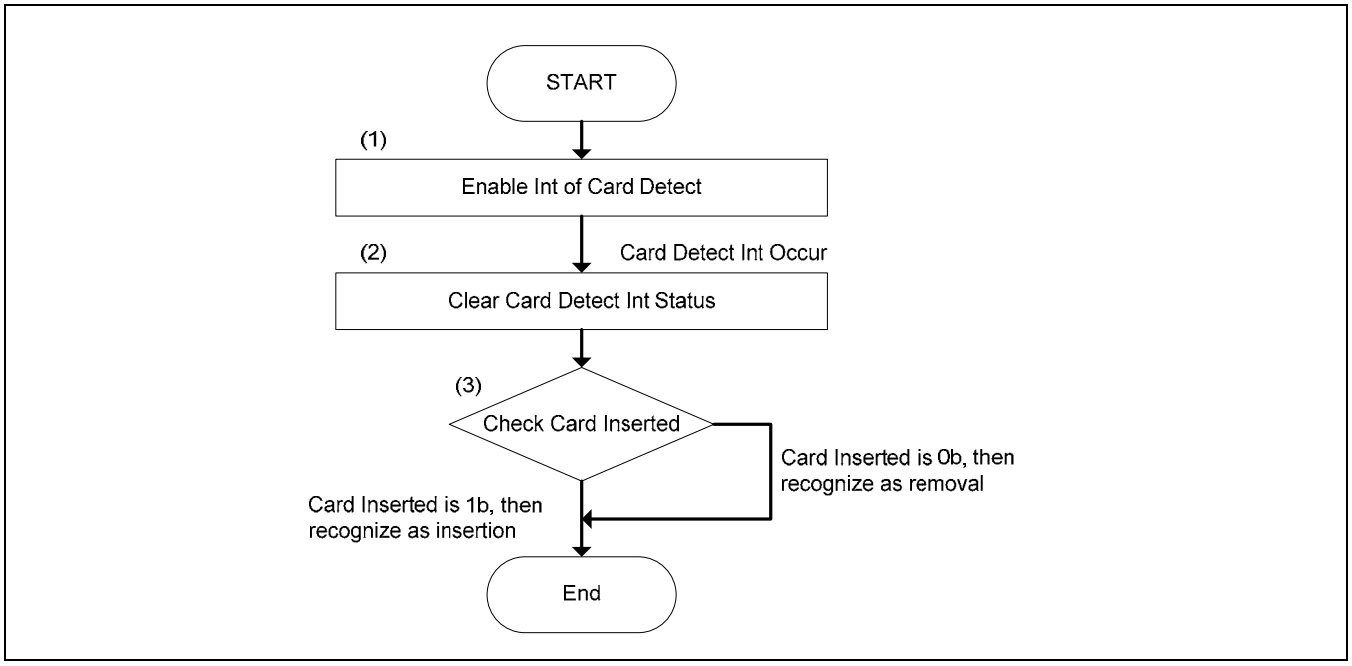


Figure 8.12-2 SD Card Detect Sequence

The flow chart to detect a SD card is shown in Figure 8.12-2. The steps shown in flowchart are explained below:

1. To enable interrupt for card detection, write 1 to the following bits:

   Card Insertion Status Enable (ENSTACARDNS) in the Normal Interrupt Status Enable register

   Card Insertion Signal Enable (ENSIGCARDNS) in the Normal Interrupt Signal Enable register

   Card Removal Status Enable (ENSTACARDREM) in the Normal Interrupt Status Enable register

   Card Removal Signal Enable (ENSIGCARDREM) in the Normal Interrupt Signal Enable register

2. If Host Driver detects the card insertion or removal, it clears the interrupt statuses. If Card Insertion interrupt (STACARDINS) is generated, write 1 to Card Insertion in the Normal Interrupt Status register. If Card Removal interrupt (STACARDREM) is generated, write 1 to Card Removal in the Normal Interrupt Status register.
3. Check Card Inserted in the Present State register. If Card Inserted (INSCARD) is 1, the Host Driver supplies power and clock to the SD card. If Card Inserted is 0, the other executing process of the Host Driver is closed.

4.2 SD CLOCK SUPPLY SEQUENCE

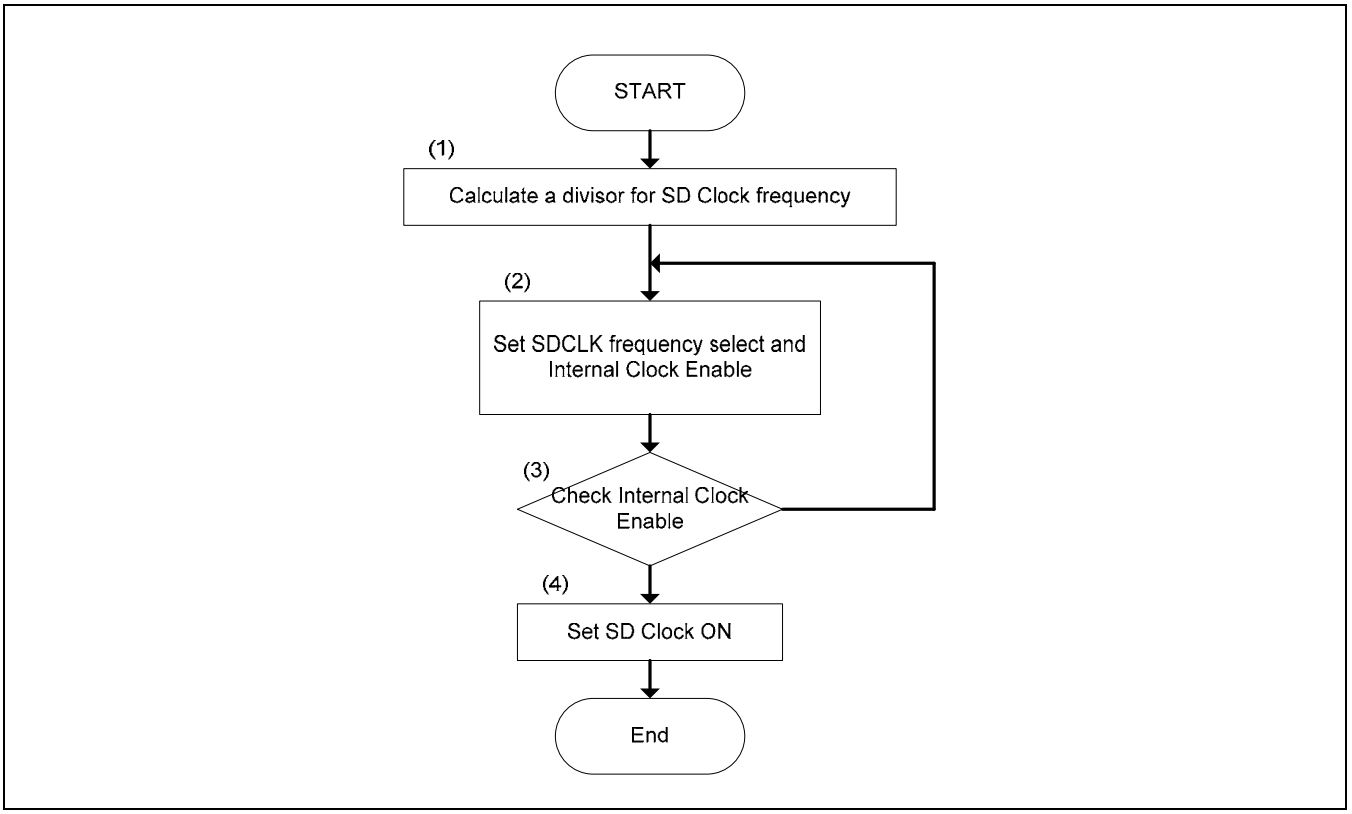


Figure 8.12-3 SD Clock Supply Sequence

The sequence to set SD Clock to a SD card is described in Figure 8.12-3. The clock is enabled before one of the following actions is taken.

a) Issuing a SD command

b) Detect an interrupt from a SD card in 4-bit mode.

The steps shown in Figure 8.12-3 are explained below:

1. Calculate a divisor to determine SD Clock frequency for SD Clock by reading Base Clock Frequency.
   Refer to clock control register.
2. Set Internal Clock Enable (ENINTCLK) and SDCLK Frequency Select in the Clock Control register in accordance with the calculated result of step (1).
3. Check Internal Clock Stable (STBLINTCLK) in the Clock Control register. Repeat this step until Clock Stable is 1.
4. Set SD Clock Enable (ENSDCLK) in the Clock Control register to 1. After ENSDCLK is set, the Host Controller starts SD Clock.

4.3 SD CLOCK STOP SEQUENCE

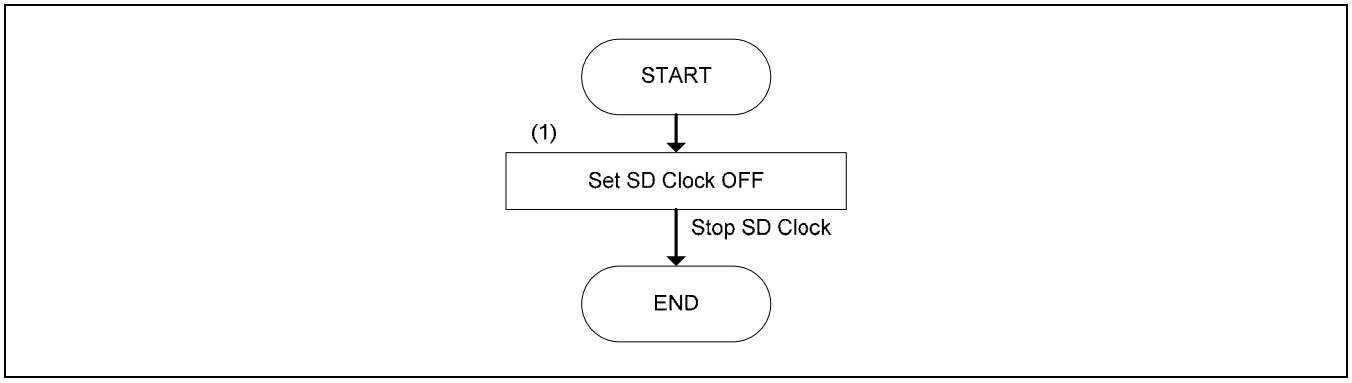


Figure 8.12-4 SD Clock Stop Sequence

The flow chart for stopping the SD Clock is shown in Figure 8.12-4. The Host Driver does not stop the SD Clock if a SD transaction takes place on the SD Bus -- namely, when either Command Inhibit (DAT) or Command Inhibit (CMD) in the Present State register is set to 1.

1. Set SD Clock Enable (ENSDCLK) in the Clock Control register to 0. After ENSDCLK is set, the Host Controller stops SD Clock.

4.4 SD CLOCK FREQUENCY CHANGE SEQUENCE

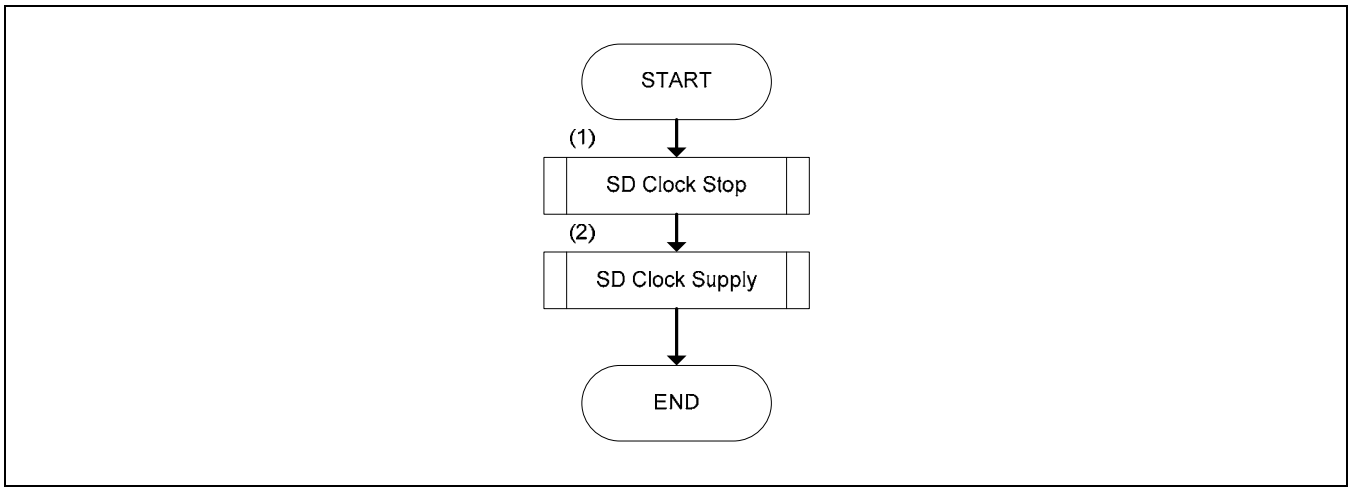


Figure 8.12-5 SD Clock Change Sequence

The sequence for changing SD Clock frequency is shown in Figure 8.12-5. If SD Clock is still off, skip step (1).

The steps shown in Figure 8.12-5 are explained below:

1. Perform SD Clock Stop Sequence. Refer to 8.12.4.2
2. Perform SD Clock Supply Sequence. Refer to 8.12.4.3

4.5 SD BUS POWER CONTROL SEQUENCE

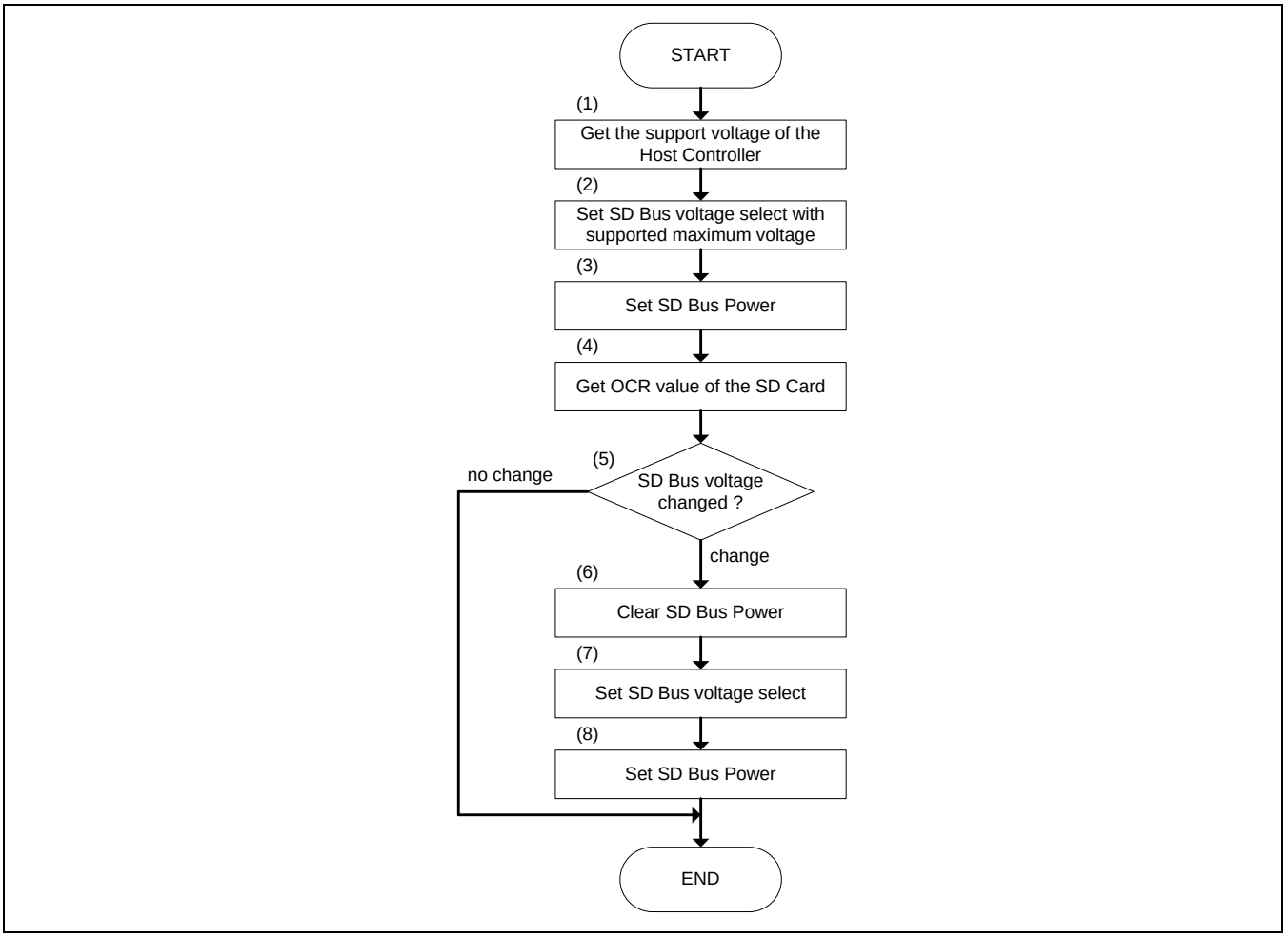


Figure 8.12-6 SD Bus Power Control Sequence

The sequence for controlling the SD Bus Power is shown in Figure 8.12-6 and steps are described below:

(1) Read the Capabilities register, to get the support voltage of the Host Controller.
(2) Set SD Bus Voltage Select in external power regulator (optional) with maximum voltage that the Host Controller supports.
(3) Set SD Bus Power (PWRON) in the Power Control register to 1.
(4) Get the OCR value of all function internal of SD card.
(5) Judge whether SD Bus voltage must be changed or not. If SD Bus voltage must be changed, continue with step (6). If SD Bus voltage is not to be changed, go to 'End'.
(6) Set SD Bus Power in the Power Control register to 0 for clearing this bit. The card requires voltage rising from 0 volt to detect it correctly. The Host Driver clears SD Bus Power before changing voltage by setting SD Bus Voltage Select.
(7) Set SD Bus Voltage Select (SELPWRLVL) in the Power Control register.
(8) Set SD Bus Power (PWRON) in the Power Control register to 1.

NOTE: Step (2) and step (3) can be executed at same time. Also, step (7) and step (8) can be executed at same time.

4.6 CHANGE BUS WIDTH SEQUENCE

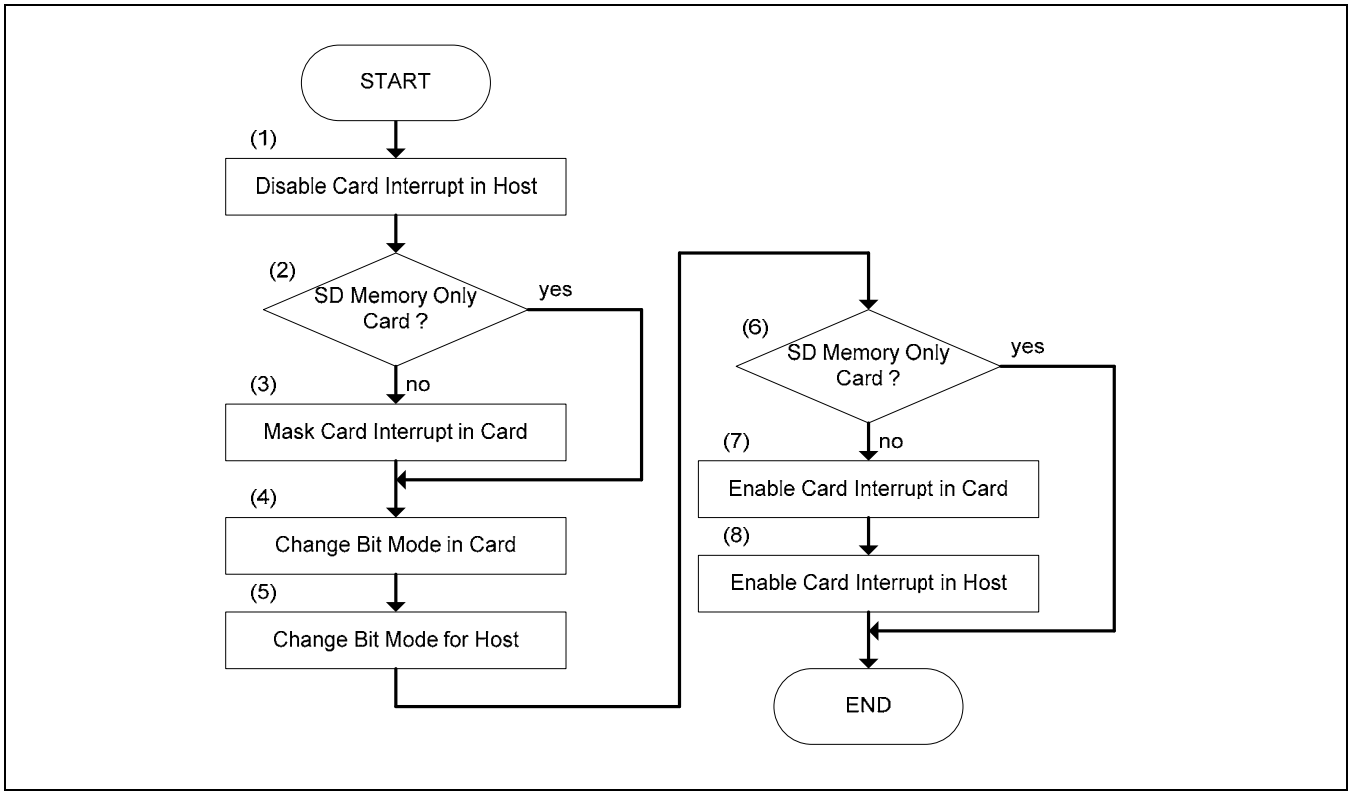


Figure 8.12-7 Change Bus Width Sequence

The sequence to change bit mode on SD Bus is shown in Figure 8.12-7 and steps are described below:

(1) Set Card Interrupt Status Enable (STACARDINT) in the Normal Interrupt Status Enable register to 0 to mask incorrect interrupts that may occur while changing the bus width.

(2) If SD memory card is used, go to step (4). In case of other card, go to step (3).

(3) Set "IENM" of the CCCR in a SDIO or SD combo card to 0 by CMD52.

(4) Change the bit mode for a SD card. Set Bus Width of Bus Interface Control register in CCCR to change SD memory card bus width by ACMD6(Set bus width) and SDIO card bus width.

(5) If you want to change to 4-bit mode, set Data Transfer Width(WIDE4) to 1 in the Host Control register. In another case (1-bit mode), set this bit to 0.

(6) If SD memory card is used, go to the 'End'. In case of other card, go to step (7).

(7) Set "IENM" of the CCCR in a SDIO or SD combo card to 1 by CMD52.

(8) Set Card Interrupt Status Enable to 1 in the Normal Interrupt Status Enable register.

4.7 TIMEOUT SETTING FOR DAT LINE

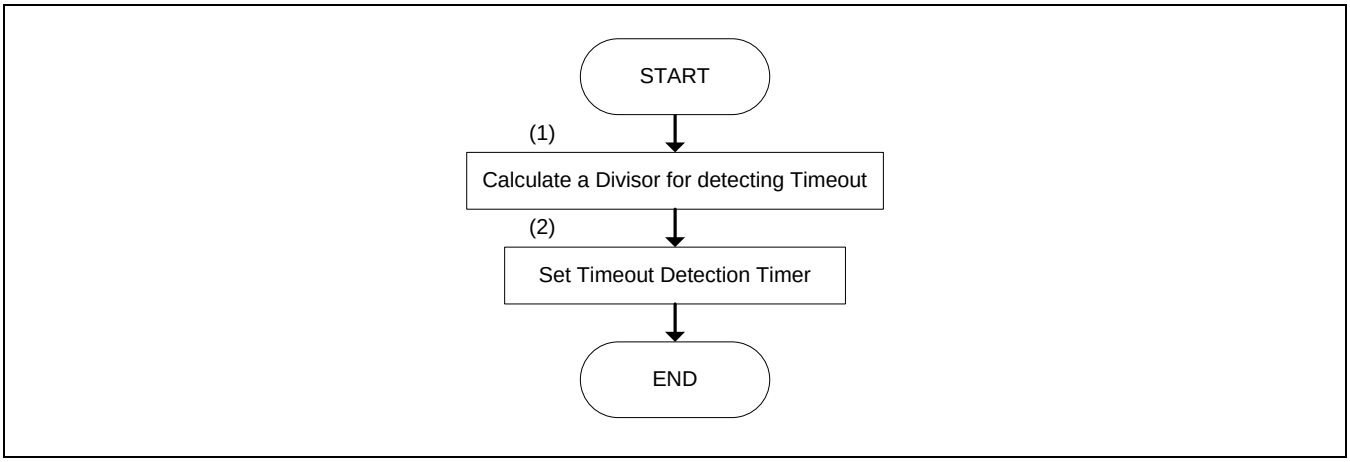


Figure 8.12-8 Timeout Setting Sequence

In order to detect timeout errors on DAT line, the Host Driver executes the following two steps before any SD transaction.

(1) To calculate a divisor for detecting timeout, refer to Timeout Control Register.

(2) Set Data Timeout Counter Value (TIMEOUTCON) in the Timeout Control register in accordance with the value of step (1).

4.8 SD TRANSACTION GENERATION

This section describes the sequence to generate and control various kinds of SD transactions. SD transactions are classified into three cases:

(1) Transactions that do not use the DAT line.

(2) Transactions that use the DAT line for the busy signal.

(3) Transactions that use the DAT line for transferring data.

In this specification the first and the second case's transactions are classified as "Transaction Control without Data Transfer using DAT Line", the third case's transaction is classified as "Transaction Control with Data Transfer using DAT Line".

Please refer to the specifications below for the detailed specifications on the SD Command itself:

- SD Memory Card Specification Part 1

PHYSICAL LAYER SPECIFICATION Version 1.01

- SD Card Specification PART E1

Secure Digital Input/Output (SDIO) Specification Version 1.00

4.9 SD COMMAND ISSUE SEQUENCE

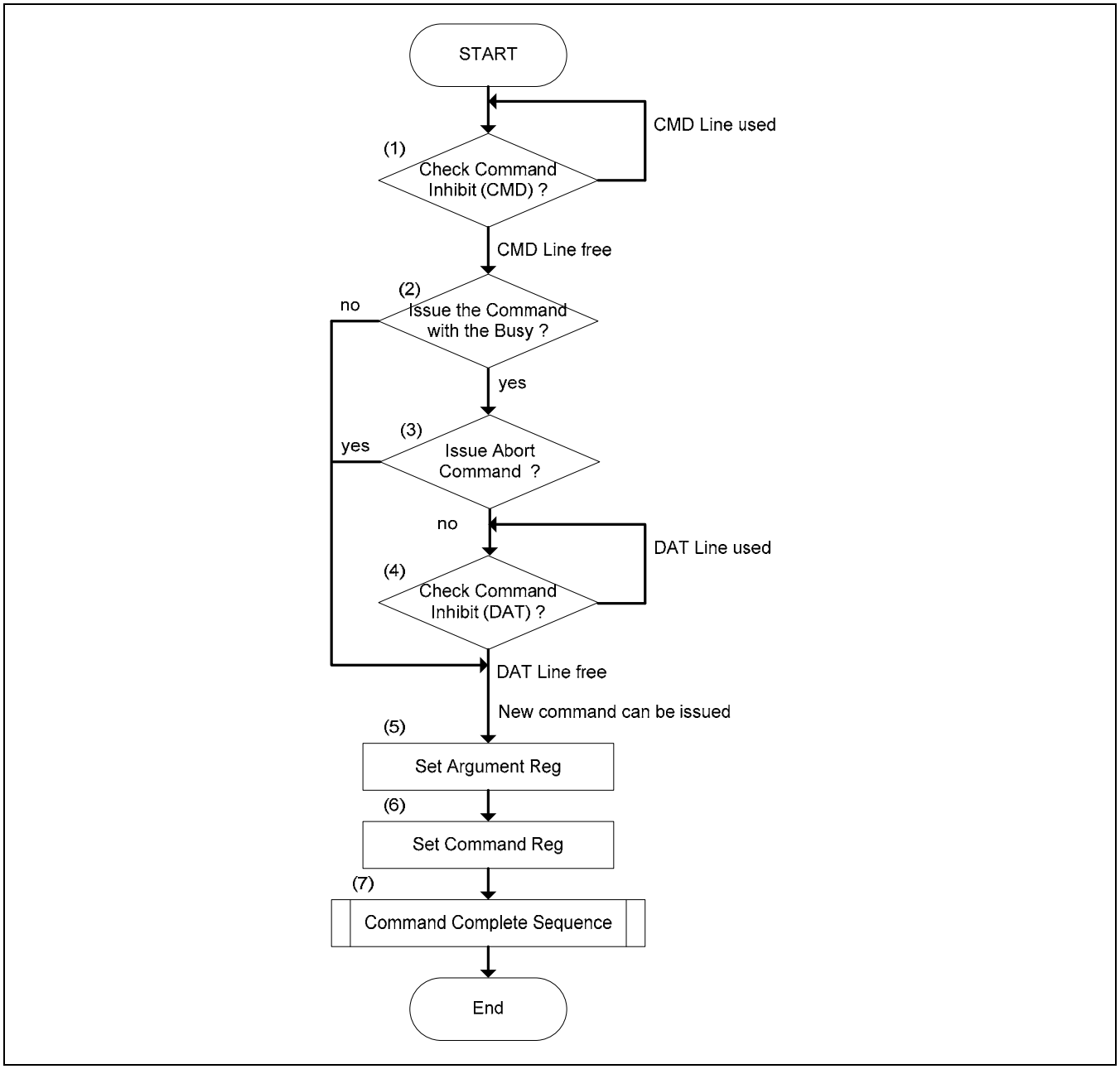


Figure 8.12-9 Timeout Setting Sequence

Take the following steps to set Timeout:

(1) Check Command Inhibit (CMD) in the Present State register. Repeat this step until Command Inhibit (CMD) is 0. If Command Inhibit (CMD) is 1, the Host Driver does not issue a SD Command.

(2) If the Host Driver issues a SD Command with busy signal, go to step (3). If without busy signal, go to step (5).

(3) If the Host Driver issues an abort command, go to step (5). If no abort command is issued, go to step (4).

(4) Check Command Inhibit (DAT) in the Present State register. Repeat this step until Command Inhibit (DAT) is 0.

(5) Set the value corresponding to the issued command in the Argument register.

(6) Set the value corresponding to the issued command in the Command register.

NOTE: If the upper byte is written in the Command register, it issues a SD command.

(7) Perform Command Complete Sequence

4.10 COMMAND COMPLETE SEQUENCE

The sequence for completing the SD Command is shown in Figure 8.12-10. The following errors might occur during this sequence: Command Index/ End bit/ CRC/ Timeout Error.

(1) Wait for the Command Complete Interrupt. If the Command Complete Interrupt occurs, go to step (2).

(2) Write 1 to Command Complete (STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.

(3) Read the Response register and get necessary information in accordance with the issued command.

(4) Judge whether the command uses the Transfer Complete Interrupt or not. If it uses Transfer Complete, proceed with step (5). If not, go to step (7).

(5) Wait for the Transfer Complete Interrupt. If the Transfer Complete Interrupt occurs, go to step (6).

(6) Write 1 to Transfer Complete (STATRANCMPLT) in the Normal Interrupt Status register to clear this bit.

(7) Check for errors in Response Data. If there is no error, proceed with step (8). If there is an error, go to step (9).

(8) Return Status of "No Error".

(9) Return Status of "Response Contents Error".

NOTES:

1. While waiting for the Transfer Complete interrupt, the Host Driver issues commands that do not use the busy signal.
2. The Host Driver judges the Auto CMD12 (Stop Command) complete by monitoring Transfer Complete.
3. If the last block of un-protected area is read using memory multiple blocks read command (CMD18), OUT_OF_RANGE error may occur even if the sequence is correct. The Host Driver must ignore it. This error appears in the response of Auto CMD12 or in the response of the next memory command.

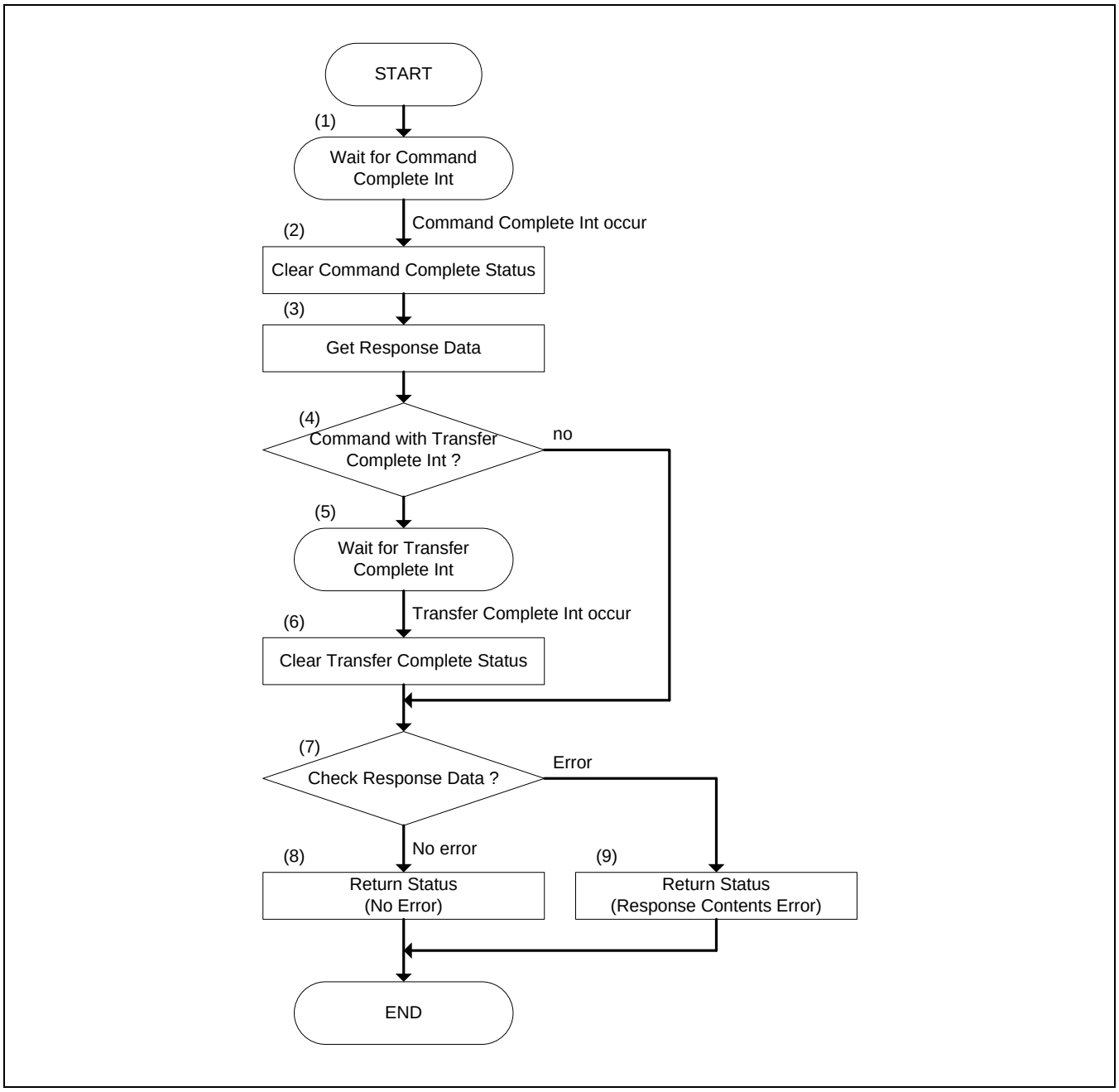


Figure 8.12-10 Command Complete Sequence

4.11 TRANSACTION CONTROL WITH DATA TRANSFER USING DAT LINE

Depending on whether DMA (optional) is used or not, there are two execution methods. The sequence without using DMA is shown in Figure 8.12-11 and the sequence using DMA is shown in Figure 8.12-12.

In addition, the sequences for SD transfers are basically classified according to how the number of blocks is specified. The three kinds of classification are as follows:

1) Single Block Transfer:

The number of blocks is specified to the Host Controller before the transfer. The number of blocks specified is always one.

2) Multiple Block Transfer:

The number of blocks is specified to the Host Controller before the transfer. The number of blocks specified is one or more.

3) Infinite Block Transfer:

The number of blocks is not specified to the Host Controller before the transfer. This transfer is continued until an abort transaction is executed. This abort transaction is performed by CMD12 (Stop Command) in the case of a SD memory card and by CMD52 (IO_RW_DIRECT) in the case of a SDIO card.

4.12 NOT USING DMA

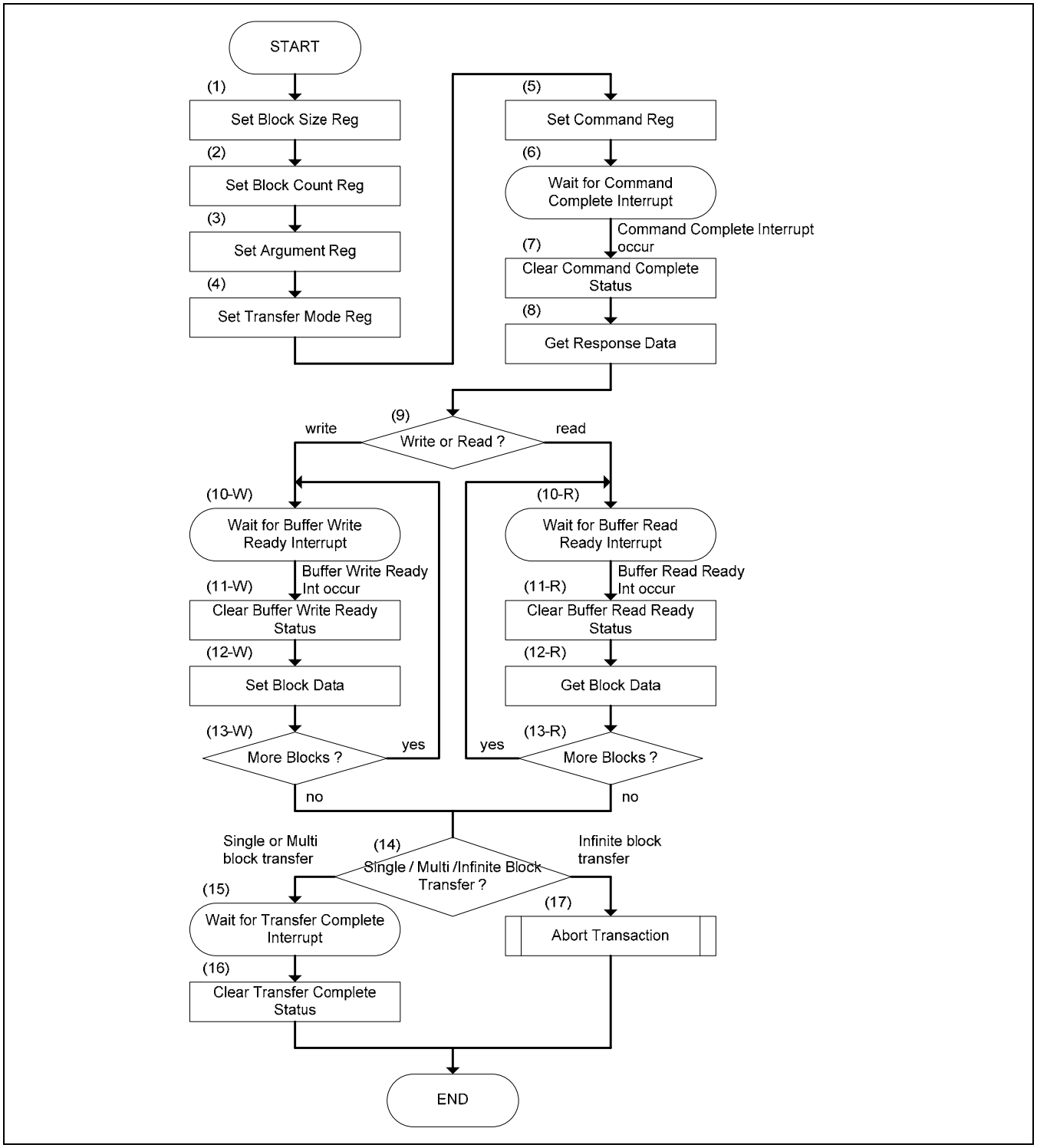


Figure 8.12-11 Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)

(1) Set the value corresponding to the executed data byte length of one block to Block Size register.

(2) Set the value corresponding to the executed data block count to Block Count register.

(3) Set the value corresponding to the issued command to Argument register.

(4) Set the value to Multi / Single Block Select and Block Count Enable. At this time, set the value corresponding to the issued command to Data Transfer Direction, Auto CMD12 Enable and DMA Enable.

(5) Set the value corresponding to the issued command to Command register.

NOTE: If the upper byte is written in the Command register, it issues a SD command

(6) Wait for the Command Complete Interrupt.

(7) Write 1 to the Command Complete (STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.

(8) Read Response register and get necessary information in accordance with the issued command.

(9) If this sequence is for write to a card, proceed to step (10-W). If read from a card, go to step (10-R).

(10-W) Wait for Buffer Write Ready Interrupt.

(11-W) Write 1 to the Buffer Write Ready (STABUFWTRDY) in the Normal Interrupt Status register to clear this bit.

(12-W) Write block data (in according to the number of bytes specified at the step (1)) to Buffer Data Port register.

(13-W) Repeat until all blocks are sent and then go to step (14).

(10-R) Wait for the Buffer Read Ready Interrupt.

(11-R) Write 1 to the Buffer Read Ready (STABUFRDRDY) in the Normal Interrupt Status register to clear this bit.

(12-R) Read block data (in according to the number of bytes specified at the step (1)) from the Buffer Data Port register.

(13-R) Repeat until all blocks are received and proceed to step (14).

(14) If this sequence is for Single or Multiple Block Transfer, proceed to step (15). In case of Infinite Block Transfer, go to step (17).

(15) Wait for Transfer Complete Interrupt.

(16) Write 1 to the Transfer Complete (STATRANCMPLT) in the Normal Interrupt Status register to clear this bit.

(17) Perform the sequence for Abort Transaction.

NOTE: Step (1) and Step (2) can be executed at same time. Step (4) and Step (5) can be executed at same time

4.13 USING DMA

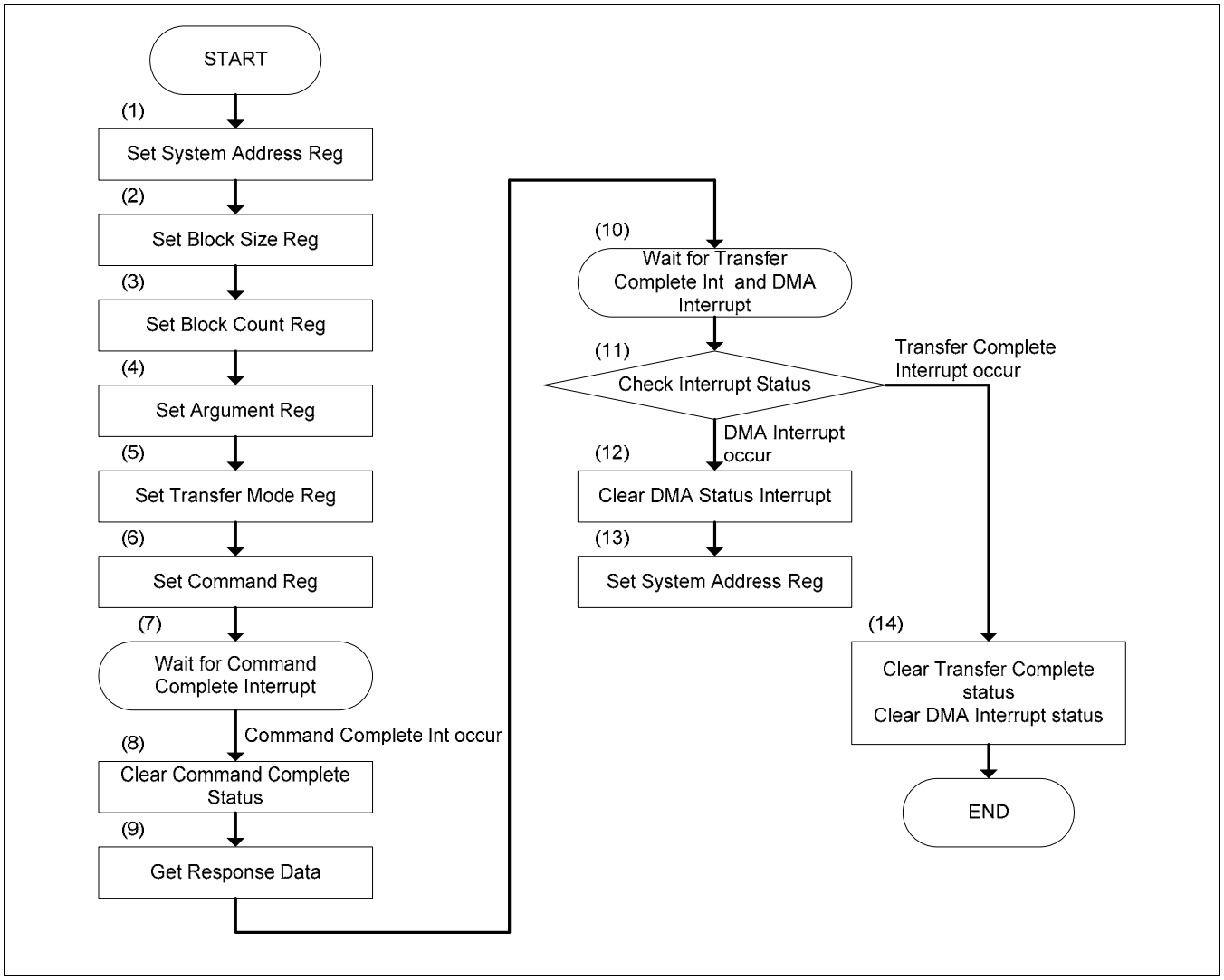


Figure 8.12-12 Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)

(1) Set the system address for DMA in the System Address register.

(2) Set the value corresponding to the executed data byte length of one block in the Block Size register.

(3) Set the value corresponding to the executed data block count in the Block Count register (BLKCNT).

(4) Set the value corresponding to the issued command in the Argument register (ARGUMENT).

(5) Set the values for Multi / Single Block Select and Block Count Enable.

At this time, set the value corresponding to the issued command for Data Transfer Direction, Auto CMD12 Enable and DMA Enable.

(6) Set the value corresponding to the issued command in the Command register (CMDREG).

NOTE: If the upper byte is written in the Command register, it issues a SD command and DMA is started.

(7) Wait for the Command Complete Interrupt.

(8) Write 1 to the Command Complete (STACMDCMPLT) in the Normal Interrupt Status register to clear this bit.

(9) Read Response register and get necessary information in accordance with the issued command.

(10) Wait for the Transfer Complete Interrupt and DMA Interrupt.

(11) If Transfer Complete (STATRANCMPLT) is set 1, go to Step (14) else if DMA Interrupt is set to 1; proceed to Step (12). Transfer Complete is higher priority than DMA Interrupt.

(12) Write 1 to the DMA Interrupt in the Normal Interrupt Status register to clear this bit.

(13) Set the next system address of the next data position to the System Address register and go to Step (10).

(14) Write 1 to the Transfer Complete and DMA Interrupt in the Normal Interrupt Status register to clear this bit.

NOTE: Step (2) and Step (3) can be executed simultaneously. Step (5) and Step (6) can also be executed simultaneously.

5 ABORT TRANSACTION

Abort transaction is performed by issuing CMD12 (Stop Command) for a SD memory card and by issuing CMD52 for a SDIO card. There are two cases where the Host Driver needs to do an Abort Transaction. The first case is if the Host Driver stops Infinite Block Transfers. The second case is if the Host Driver stops transfers while a Multiple Block Transfer is executing.
There are two ways to issue an Abort Command. The first is an asynchronous abort. The second is a synchronous abort. In an asynchronous abort sequence, the Host Driver issues an Abort Command at anytime unless Command Inhibit (CMD) in the Present State register is set to 1. In a synchronous abort, the Host Driver issues an Abort Command after the data transfer stopped by using Stop At Block Gap Request in the Block Gap Control register.

6 DMA TRANSACTION

DMA allows a peripheral to read and write memory without intervention from the CPU. DMA executes one SD command transaction. Host Controllers that support DMA supports both single block and multiple block transfers.

The System Address register points to the first data address, and data is then accessed sequentially from that address. Host Controller registers remains accessible for issuing non-DAT line commands during a DMA transfer. The result of a DMA transfer is same regardless of the system bus transaction method used. DMA does not support infinite transfers.

DMA transfers are stopped and restarted using control bits in the Block Gap Control register. If the Stop At Block Gap Request is set, DMA transfers is suspended. If the Continue Request is set or a Resume Command is issued, DMA continues to execute transfers. Refer to the Block Gap Control register for details. If SD Bus errors occur, SD Bus transfers and DMA transfers are stopped. Setting the Software Reset for DAT Line in the Software Reset register aborts DMA transfers.

7 ADMA(ADVANCED DMA)

In the SD Host Controller Standard Specification Version 2.00, new DMA transfer algorithm called ADMA (Advanced DMA) is defined. The DMA algorithm defined in the SD Host Controller Standard Specification Version 1.00 is called SDMA (Single Operation DMA). SDMA had disadvantage that **DMA Interrupt** generated at every page boundary disturbs CPU to reprogram the new system address. This SDMA algorithm forms a performance bottleneck by interruption at every page boundary. ADMA adopts scatter gather DMA algorithm so that higher data transfer speed is available. The Host Driver can program a list of data transfers between system memory and SD card to the Descriptor Table before executing ADMA. It enables ADMA to operate without interrupting the Host Driver. Furthermore, ADMA can support not only 32-bit system memory addressing but also 64-bit system memory addressing. The 32-bit system memory addressing uses lower 32-bit field of 64-bit address registers. Support of SDMA and ADMA are optional for the Host Controller. ADMA improves the restriction so that data of any location and any size can be transferred in system memory. The format of Descriptor Table is different between them. The Host Controller Specification Ver2.00 defines ADMA as standard ADMA.

7.1 BLOCK DIAGRAM OF ADMA

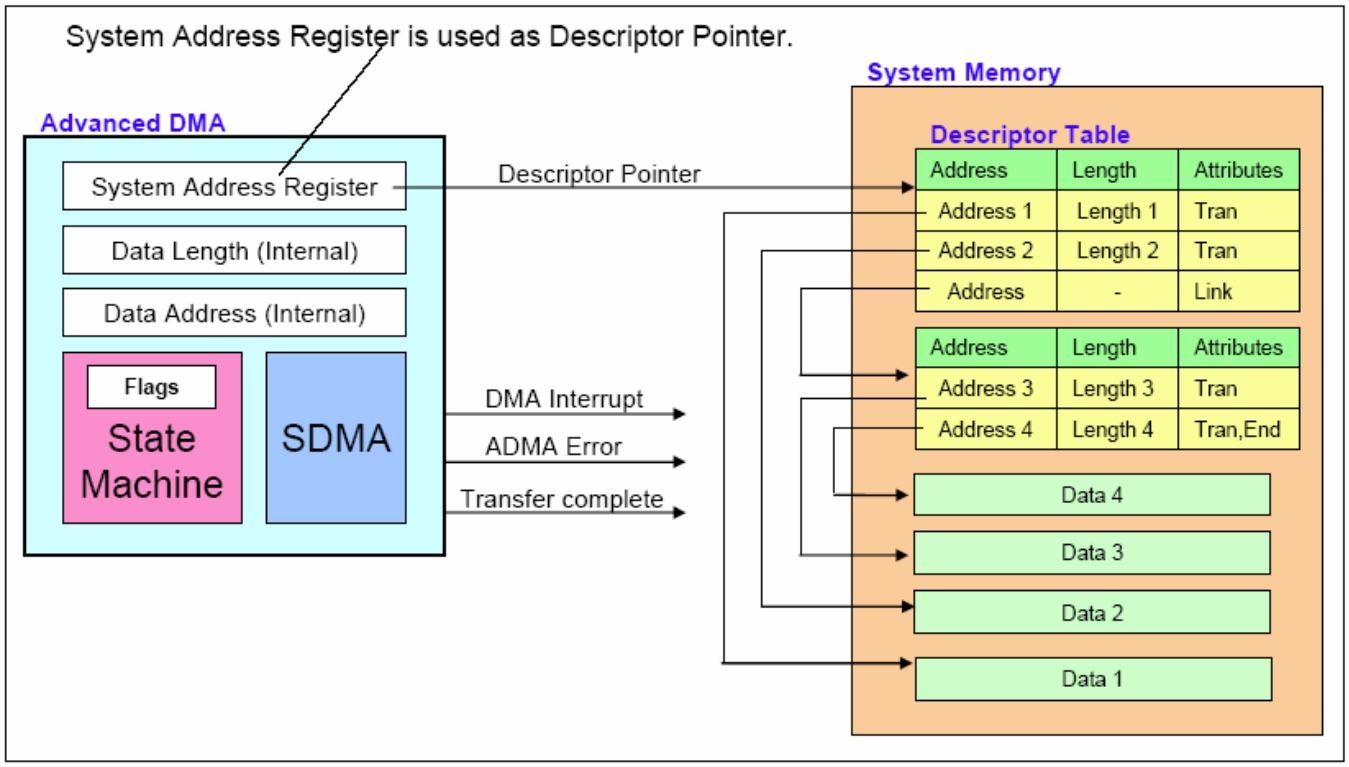


Figure 8.12- 13 Block Diagram of ADMA

Figure 8.12-13 shows block diagram of ADMA. The Descriptor Table is created in system memory by the Host Driver. 32-bit Address Descriptor Table is used for the system with 32-bit addressing and 64-bit Address Descriptor Table is used for the system with 64-bit addressing. Each descriptor line (one executable unit) consists with address, length and attribute field. The attribute specifies operation of the descriptor line. ADMA includes SDMA, State Machine and Registers circuits. ADMA does not use 32-bit SDMA System Address Register (offset 0) but uses the 64-bit Advanced DMA System Address register (offset 058h) for descriptor pointer. Writing Command register triggers off ADMA transfer. ADMA fetches one descriptor line and execute it. This procedure is repeated until end of descriptor is found (End=1 in attribute).

7.2 AN EXAMPLE OF ADMA PROGRAMMING

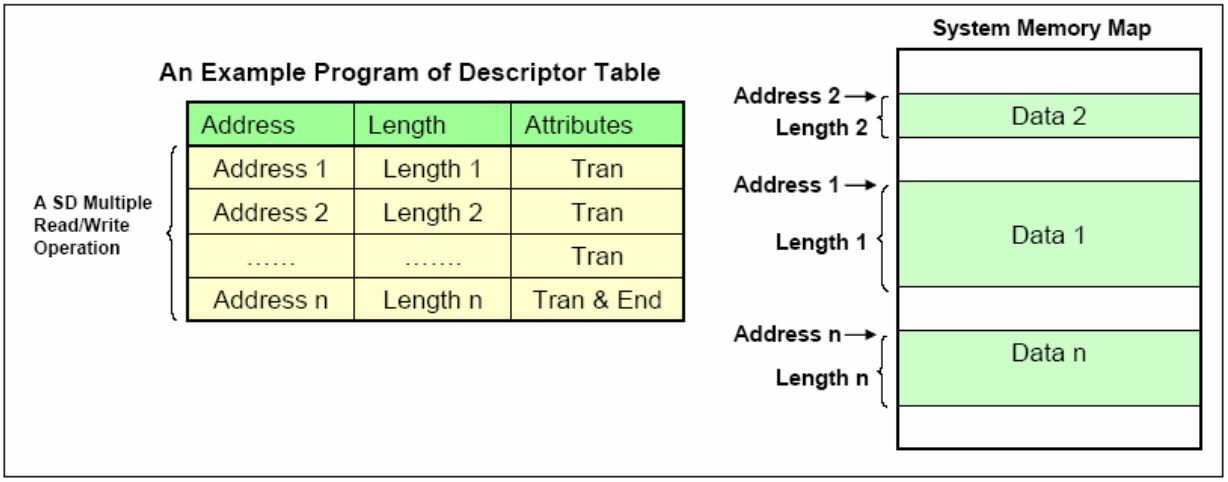


Figure 8.12- 14 An Example of ADMA Data Transfer

Figure 8.12-14 shows a typical ADMA descriptor program. The data area is sliced in various lengths and each slice is placed somewhere in system memory. The Host Driver describes the Descriptor Table with set of address, length and attributes. Each sliced data is transferred in turns as programmed in descriptor.

7.3 DATA ADDRESS AND DATA LENGTH REQUIREMENTS

There are 3 requirements to program descriptor. The minimum unit of address is 4byte.

The maximum data length of each descriptor line is less than 64KB.

Total Length = Length 1 + Length 2 + Length 3 + ... + Length n

= multiple of Block Size If total length of a descriptor were not multiple of block size, ADMA transfer might not be terminated. In this case, the transfer should be aborted by data timeout. Block Count register is defined as 16-bit register and it limits the maximum of 65535 blocks transfer. If ADMA operation is less than or equal 65535 blocks transfer, Block Count register can be used. In this case, total length of Descriptor Table shall be equivalent to multiply block size and block count. If ADMA operation is more than 65535 blocks transfer, Block Count register shall be disabled by setting 0 to Block Count Enable in the Transfer Mode Register. In this case, length of data transfer is not designated by block count but Descriptor Table. Therefore, the timing of detecting the last block on SD bus may be different and it affects the control of Read Transfer Active, Write Transfer Active and DAT line Active in the Present State register. In case of read operation, several blocks may be read more than required. The Host Driver shall ignore out of range error if the read operation is for the last block of memory area.

7.4 DESCRIPTOR TABLE

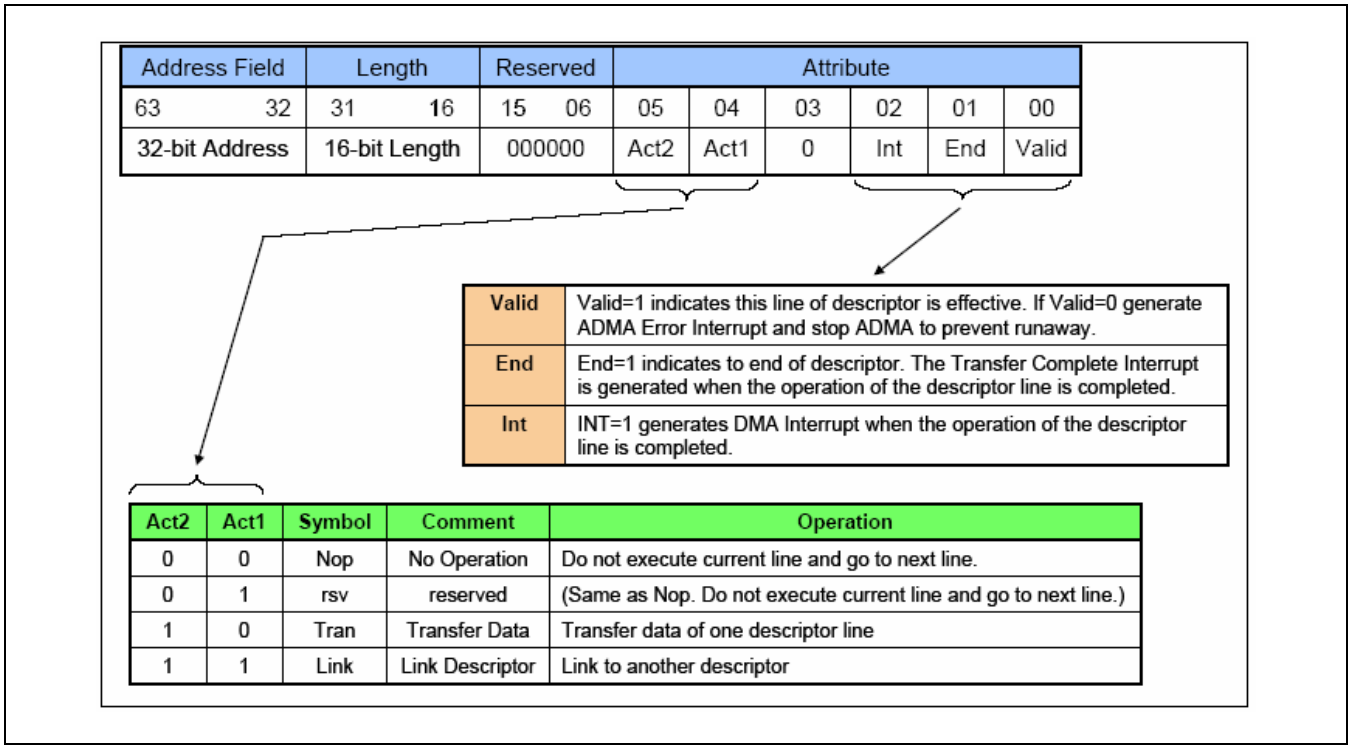


Figure 8.12- 15 32-bit Address Descriptor Table

Figure 8.12-15 shows the definition of 32-bit Address Descriptor Table. One descriptor line consumes 64-bit (8-byte) memory space. Attribute is used to control descriptor. 3 action symbols are specified. "Nop" operation skips current descriptor line and fetches next one. "Tran" operation transfers data designated by address and length field. "Link" operation is used to connect separated two descriptors. The address field of link points to next Descriptor Table. The combination of Act2=0 and Act1=1 is reserved and defined the same operation as Nop. A future version of controller may use this field and redefine a new operation. 32-bit address is stored in the lower 32-bit of 64-bit address registers. Address field shall be set on 32-bit boundary (Lower 2-bit is always set to 0) for 32-bit address descriptor table. Table 8.12-1 shows the definition of length field in the Descriptor Table.

Table 8.12- 1 ADMA Length Field

Length Field	Value of Length
0000h	65536 bytes
0001h	1 byte
0002h	2 bytes
……	……
FFFFh	65535 bytes

7.5 ADMA STATES

Figure 8.12-16 shows state diagram of ADMA. 4 states are defined; Fetch Descriptor state, Change Address state, Transfer Data state, and Stop ADMA state. Operation of each state is explained in Table 8.12-3.

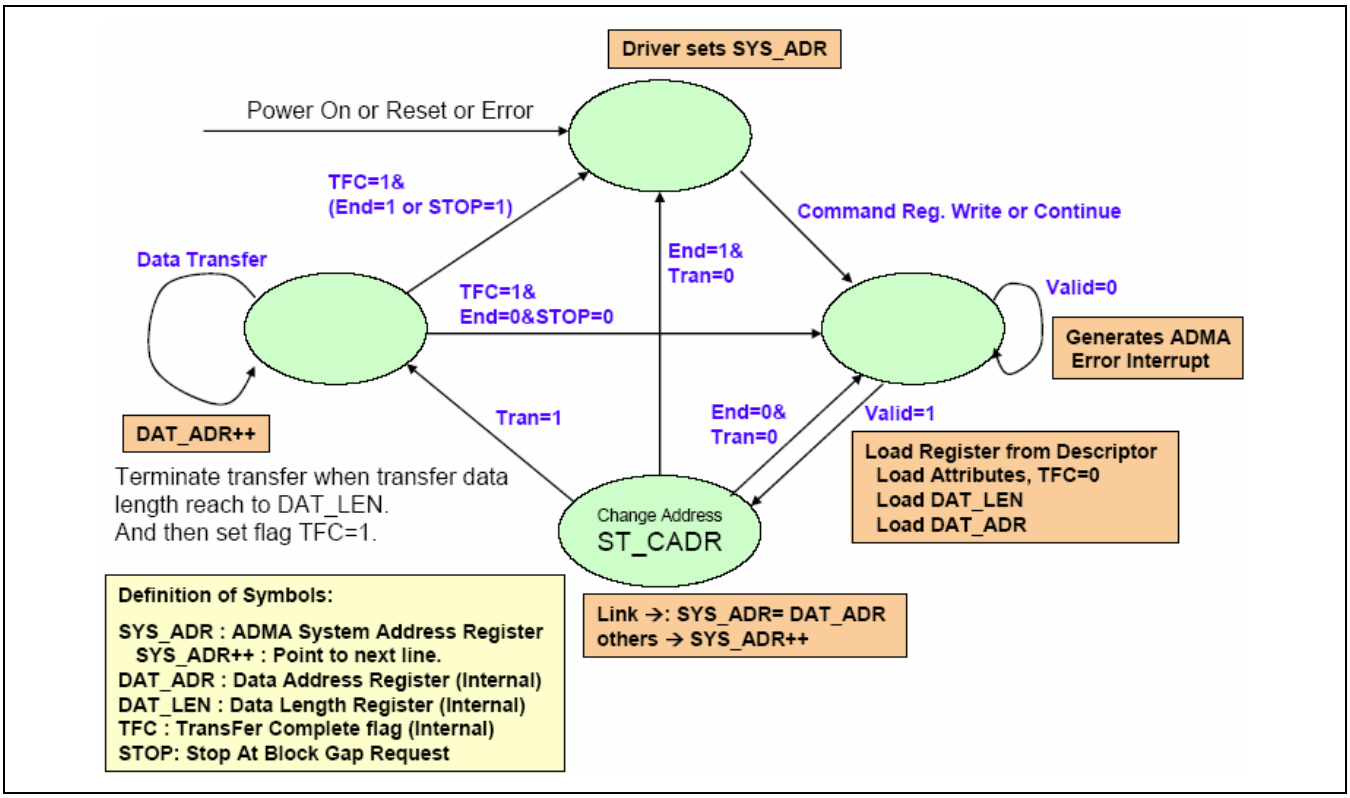


Figure 8.12- 16 State Diagram of the ADMA

Table 8.12- 2 ADMA States

State Name	Operation
ST_FDS (Fetch Descriptor)	ADMA fetches a descriptor line and set parameters in internal registers. Next go to ST_CADR state.
ST_CADR (Change Address)	Link operation loads another Descriptor address to ADMA System Address register. In other operations, ADMA System Address register is incremented to point next descriptor line. If End=0, go to ST_TFR state. ADMA shall not be stopped at this state even if some errors occur.
ST_TFR (Transfer Data)	Data transfer of one descriptor line is executed between system memory and SD card. If data transfer continues (End=0) go to ST_FDS state. If data transfer completes, go to ST_STOP state.
ST_STOP (Stop DMA)	ADMA stays in this state in following cases: (1) After Power on reset or software reset. (2) All descriptor data transfers are completed If a new ADMA operation is started by writing Command register, go to ST_FDS state.

ADMA does not support suspend / resume function but stop and continue are available. When the Stop At Block Gap Request in the Block Gap Control register is set during the ADMA operation, the Block Gap Event Interrupt is generated when ADMA is stopped at block gap. The Host Controller shall stop ADMA read operation by using

Read Wait or stopping SD Clock. While stopping ADMA, any SD commands cannot be issued. (In case of Host Controller version 1.00, the Stop At Block Gap Request can be set only when the card supports the Read Wait.)

Error occurrence during ADMA transfer may stops ADMA operation and generates ADMA Error Interrupt. The ADMA Error State field in the ADMA Error Status register holds state of ADMA stopped. The host driver can identify error descriptor location by following method. If ADMA stopped at ST_FDS state, the ADMA System Address Register points the error descriptor line. If ADMA stopped at ST_TFR or ST_STOP state, the ADMA System Address Register points the next location of error descriptor line. By this reason, ADMA2 shall not stop at ST_CADR state.

8 I/O DESCRIPTION

Function Signal	I/O	Descrioption	Pad	Type
SD_0_CLK	OUTPUT	Clock for HSMMC0	Xmmc0CLK	muxed
SD_0_CMD	IN/OUT	Command for HSMMC0	Xmmc0CMD	muxed
SD_0_D[0]	IN/OUT	Data for HSMMC0	Xmmc0DATA[0]	muxed
SD_0_D[1]	IN/OUT	Data for HSMMC0	Xmmc0DATA[1]	muxed
SD_0_D[2]	IN/OUT	Data for HSMMC0	Xmmc0DATA[2]	muxed
SD_0_D[3]	IN/OUT	Data for HSMMC0	Xmmc0DATA[3]	muxed
SD_0_D[4]	IN/OUT	Data for HSMMC0	Xmmc0DATA[4]	muxed
SD_0_D[5]	IN/OUT	Data for HSMMC0	Xmmc0DATA[5]	muxed
SD_0_D[6]	IN/OUT	Data for HSMMC0	Xmmc0DATA[6]	muxed
SD_0_D[7]	IN/OUT	Data for HSMMC0	Xmmc0DATA[7]	muxed
SD_0_CDn	INPUT	Card Detect for HSMMC0	Xmmc0CDn	muxed
SD_1_CLK	OUTPUT	Clock for HSMMC1	XciPCLK	muxed
SD_1_CMD	IN/OUT	Command for HSMMC1	XciHREF	muxed
SD_1_D[0]	IN/OUT	Data for HSMMC1	XciDATA[0]	muxed
SD_1_D[1]	IN/OUT	Data for HSMMC1	XciDATA[1]	muxed
SD_1_D[2]	IN/OUT	Data for HSMMC1	XciDATA[2]	muxed
SD_1_D[3]	IN/OUT	Data for HSMMC1	XciDATA[3]	muxed
SD_1_D[4]	IN/OUT	Data for HSMMC1	XciDATA[4]	muxed
SD_1_D[5]	IN/OUT	Data for HSMMC1	XciDATA[5]	muxed
SD_1_D[6]	IN/OUT	Data for HSMMC1	XciDATA[6]	muxed
SD_1_D[7]	IN/OUT	Data for HSMMC1	XciDATA[7]	muxed
SD_1_CDn	INPUT	Card Detect for HSMMC1	XciVSYNC	muxed
SD_1_CLK	OUTPUT	Clock for HSMMC1	Xmmc1CLK	muxed
SD_1_CMD	IN/OUT	Command for HSMMC1	Xmmc1CMD	muxed
SD_1_D[0]	IN/OUT	Data for HSMMC1	Xmmc1DATA[0]	muxed
SD_1_D[1]	IN/OUT	Data for HSMMC1	Xmmc1DATA[1]	muxed
SD_1_D[2]	IN/OUT	Data for HSMMC1	Xmmc1DATA[2]	muxed

Function Signal	I/O	Descrioption	Pad	Type
SD_1_D[3]	IN/OUT	Data for HSMMC1	Xmmc1DATA[3]	muxed
SD_1_CDn	INPUT	Card Detect for HSMMC1	Xmmc1CDn	muxed
SD_2_CLK	OUTPUT	Clock for HSMMC2	Xmmc2CLK	muxed
SD_2_CMD	IN/OUT	Command for HSMMC2	Xmmc2CMD	muxed
SD_2_D[0]	IN/OUT	Data for HSMMC2	Xmmc2DATA[0]	muxed
SD_2_D[1]	IN/OUT	Data for HSMMC2	Xmmc2DATA[1]	muxed
SD_2_D[2]	IN/OUT	Data for HSMMC2	Xmmc2DATA[2]	muxed
SD_2_D[3]	IN/OUT	Data for HSMMC2	Xmmc2DATA[3]	muxed
SD_2_CDn	INPUT	Card Detect for HSMMC2	Xmmc2CDn	muxed

NOTE: HSMMC external pads are shared with CAMIF or SPI. In order to use these pads for HSMMC, set the GPIO before the HSMMC started. Refer to the GPIO chapter for correct GPIO settings.

9 REGISTER DESCRIPTION

Configuration register fields are assigned to one of the attributes described below:

Register	Address	R/W	Description	Reset Value
SDMASYSAD0	0xED80_0000	R/W	SDMA System Address register (Channel 0)	0x0
BLKSIZE0	0xED80_0004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 0)	0x0
BLKCNT0	0xED80_0006	R/W	Blocks count for current transfer (channel 0)	0x0
ARGUMENT0	0xED80_0008	R/W	Command Argument Register (Channel 0)	0x0
TRNMOD0	0xED80_000C	R/W	Transfer Mode Setting Register (Channel 0)	0x0
CMDREG0	0xED80_000E	R/W	Command Register (Channel 0)	0x0
RSPREG0_0	0xED80_0010	ROC	Response Register 0 (Channel 0)	0x0
RSPREG1_0	0xED80_0014	ROC	Response Register 1 (Channel 0)	0x0
RSPREG2_0	0xED80_0018	ROC	Response Register 2 (Channel 0)	0x0
RSPREG3_0	0xED80_001C	ROC	Response Register 3 (Channel 0)	0x0
BDATA0	0xED80_0020	R/W	Buffer Data Register (Channel 0)	0x0
PRNSTS0	0xED80_0024	R/ROC	Present State Register (Channel 0)	0x000A0000
HOSTCTL0	0xED80_0028	R/W	Host Control Register (Channel 0)	0x0
PWRCON0	0xED80_0029	R/W	Power Control Register (Channel 0)	0x0
BLKGAP0	0xED80_002A	R/W	Block Gap Control Register (Channel 0)	0x0
WAKCON0	0xED80_002B	R/W	Wakeup Control Register (Channel 0)	0x0
CLKCON0	0xED80_002C	R/W	Clock Control Register (Channel 0)	0x0
TIMEOUTCON0	0xED80_002E	R/W	Timeout Control Register (Channel 0)	0x0
SWRST0	0xED80_002F	R/W	Software Reset Register (Channel 0)	0x0

Register	Address	R/W	Description	Reset Value
NORINTSTS0	0xED80_0030	ROC/ RW1C	Normal Interrupt Status Register (Channel 0)	0x0
ERRINTSTS0	0xED80_0032	ROC/ RW1C	Error Interrupt Status Register (Channel 0)	0x0
NORINTSTSEN0	0xED80_0034	R/W	Normal Interrupt Status Enable Register (Channel 0)	0x0
ERRINTSTSEN0	0xED80_0036	R/W	Error Interrupt Status Enable Register (Channel 0)	0x0
NORINTSIGEN0	0xED80_0038	R/W	Normal Interrupt Signal Enable Register (Channel 0)	0x0
ERRINTSIGEN0	0xED80_003A	R/W	Error Interrupt Signal Enable Register (Channel 0)	0x0
ACMD12ERRSTS0	0xED80_003C	ROC	Auto CMD12 error status register (channel 0)	0x0
CAPAREG0	0xED80_0040	HWInit	Capabilities Register (Channel 0)	0x05E80080
MAXCURR0	0xED80_0048	HWInit	Maximum Current Capabilities Register (Channel 0)	0x0
FEAER0	0xED80_0050	W	Force Event Auto CMD12 Error Interrupt Register (Channel 0)	0x0000
FEERR0	0xED80_0052	W	Force Event Error Interrupt Register Error Interrupt (Channel 0)	0x0000
ADMAERR0	0xED80_0054	R/W	ADMA Error Status Register (Channel 0)	0x00
ADMASYSADDR0	0xED80_0058	R/W	ADMA System Address Register (Channel 0)	0x00
CONTROL2_0	0xED80_0080	R/W	Control register 2 (Channel 0)	0x0
CONTROL3_0	0xED80_0084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 0)	0x7F5F3F1F
CONTROL4_0	0xED80_008C	R/W	Control register 4 (Channel 0)	0x0
HCVER0	0xED80_00FE	HWInit	Host Controller Version Register (Channel 0)	0x0401
SDMASYSAD1	0xED90_0000	R/W	SDMA System Address register (Channel 1)	0x0
BLKSIZE1	0xED90_0004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 1)	0x0
BLKCNT1	0xED90_0006	R/W	Blocks count for current transfer (channel 1)	0x0
ARGUMENT1	0xED90_0008	R/W	Command Argument Register (Channel 1)	0x0
TRNMOD1	0xED90_000C	R/W	Transfer Mode Setting Register (Channel 1)	0x0
CMDREG1	0xED90_000E	R/W	Command Register (Channel 1)	0x0
RSPREG0_1	0xED90_0010	ROC	Response Register 0 (Channel 1)	0x0
RSPREG1_1	0xED90_0014	ROC	Response Register 1 (Channel 1)	0x0
RSPREG2_1	0xED90_0018	ROC	Response Register 2 (Channel 1)	0x0
RSPREG3_1	0xED90_001C	ROC	Response Register 3 (Channel 1)	0x0

Register	Address	R/W	Description	Reset Value
BDATA1	0xED90_0020	R/W	Buffer Data Register (Channel 1)	0x0
PRNSTS1	0xED90_0024	R/ROC	Present State Register (Channel 1)	0x000A0000
HOSTCTL1	0xED90_0028	R/W	Host Control Register (Channel 1)	0x0
PWRCON1	0xED90_0029	R/W	Power Control Register (Channel 1)	0x0
BLKGAP1	0xED90_002A	R/W	Block Gap Control Register (Channel 1)	0x0
WAKCON1	0xED90_002B	R/W	Wakeup Control Register (Channel 1)	0x0
CLKCON1	0xED90_002C	R/W	Clock Control Register (Channel 1)	0x0
TIMEOUTCON1	0xED90_002E	R/W	Timeout Control Register (Channel 1)	0x0
SWRST1	0xED90_002F	R/W	Software Reset Register (Channel 1)	0x0
NORINTSTS1	0xED90_0030	ROC/ RW1C	Normal Interrupt Status Register (Channel 1)	0x0
ERRINTSTS1	0xED90_0032	ROC/ RW1C	Error Interrupt Status Register (Channel 1)	0x0
NORINTSTSEN1	0xED90_0034	R/W	Normal Interrupt Status Enable Register (Channel 1)	0x0
ERRINTSTSEN1	0xED90_0036	R/W	Error Interrupt Status Enable Register (Channel 1)	0x0
NORINTSIGEN1	0xED90_0038	R/W	Normal Interrupt Signal Enable Register (Channel 1)	0x0
ERRINTSIGEN1	0xED90_003A	R/W	Error Interrupt Signal Enable Register (Channel 1)	0x0
ACMD12ERRSTS1	0xED90_003C	ROC	Auto CMD12 error status register (channel 1)	0x0
CAPAREG1	0xED90_0040	HWInit	Capabilities Register (Channel 1)	0x05E80080
MAXCURR1	0xED90_0048	HWInit	Maximum Current Capabilities Register (Channel 1)	0x0
FEAER1	0xED90_0050	W	Force Event Auto CMD12 Error Interrupt Register (Channel 1)	0x0000
FEERR1	0xED90_0052	W	Force Event Error Interrupt Register Error Interrupt (Channel 1)	0x0000
ADMAERR1	0xED90_0054	R/W	ADMA Error Status Register (Channel 1)	0x00
ADMASYSADDR1	0xED90_0058	R/W	ADMA System Address Register (Channel 1)	0x00
CONTROL2_1	0xED90_0080	R/W	Control register 2 (Channel 1)	0x0
CONTROL3_1	0xED90_0084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 1)	0x7F5F3F1F
CONTROL4_1	0xED90_008C	R/W	Control register 4 (Channel 1)	0x0
HCVER1	0xED90_00FE	HWInit	Host Controller Version Register (Channel 1)	0x0401
SDMASYSAD2	0xEDA0_0000	R/W	SDMA System Address register (Channel 2)	0x0

Register	Address	R/W	Description	Reset Value
BLKSIZE2	0xEDA0_0004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 2)	0x0
BLKCNT2	0xEDA0_0006	R/W	Blocks count for current transfer (channel 2)	0x0
ARGUMENT2	0xEDA0_0008	R/W	Command Argument Register (Channel 2)	0x0
TRNMOD2	0xEDA0_000C	R/W	Transfer Mode Setting Register (Channel 2)	0x0
CMDREG2	0xEDA0_000E	R/W	Command Register (Channel 2)	0x0
RSPREG0_2	0xEDA0_0010	ROC	Response Register 0 (Channel 2)	0x0
RSPREG1_2	0xEDA0_0014	ROC	Response Register 1 (Channel 2)	0x0
RSPREG2_2	0xEDA0_0018	ROC	Response Register 2 (Channel 2)	0x0
RSPREG3_2	0xEDA0_001C	ROC	Response Register 3 (Channel 2)	0x0
BDATA2	0xEDA0_0020	R/W	Buffer Data Register (Channel 2)	0x0
PRNSTS2	0xEDA0_0024	R/ROC	Present State Register (Channel 2)	0x000A0000
HOSTCTL2	0xEDA0_0028	R/W	Host Control Register (Channel 2)	0x0
PWRCON2	0xEDA0_0029	R/W	Power Control Register (Channel 2)	0x0
BLKGAP2	0xEDA0_002A	R/W	Block Gap Control Register (Channel 2)	0x0
WAKCON2	0xEDA0_002B	R/W	Wakeup Control Register (Channel 2)	0x0
CLKCON2	0xEDA0_002C	R/W	Clock Control Register (Channel 2)	0x0
TIMEOUTCON2	0xEDA0_002E	R/W	Timeout Control Register (Channel 2)	0x0
SWRST2	0xEDA0_002F	R/W	Software Reset Register (Channel 2)	0x0
NORINTSTS2	0xEDA0_0030	ROC/ RW1C	Normal Interrupt Status Register (Channel 2)	0x0
ERRINTSTS2	0xEDA0_0032	ROC/ RW1C	Error Interrupt Status Register (Channel 2)	0x0
NORINTSTSEN2	0xEDA0_0034	R/W	Normal Interrupt Status Enable Register (Channel 2)	0x0
ERRINTSTSEN2	0xEDA0_0036	R/W	Error Interrupt Status Enable Register (Channel 2)	0x0
NORINTSIGEN2	0xEDA0_0038	R/W	Normal Interrupt Signal Enable Register (Channel 2)	0x0
ERRINTSIGEN2	0xEDA0_003A	R/W	Error Interrupt Signal Enable Register (Channel 2)	0x0
ACMD12ERRSTS2	0xEDA0_003C	ROC	Auto CMD12 error status register (channel 2)	0x0
CAPAREG2	0xEDA0_0040	HWInit	Capabilities Register (Channel 2)	0x05E80080
MAXCURR2	0xEDA0_0048	HWInit	Maximum Current Capabilities Register (Channel 2)	0x0
FEAER2	0xEDA0_0050	W	Force Event Auto CMD12 Error Interrupt	0x0000

Register	Address	R/W	Description	Reset Value
			Register (Channel 2)	
FEERR2	0xEDA0_0052	W	Force Event Error Interrupt Register Error Interrupt (Channel 2)	0x0000
ADMAERR2	0xEDA0_0054	R/W	ADMA Error Status Register (Channel 2)	0x00
ADMASYSADDR2	0xEDA0_0058	R/W	ADMA System Address Register (Channel 2)	0x00
CONTROL2_2	0xEDA0_0080	R/W	Control register 2 (Channel 2)	0x0
CONTROL3_2	0xEDA0_0084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 2)	0x7F5F3F1F
CONTROL4_2	0xEDA0_008C	R/W	Control register 4 (Channel 2)	0x0
HCVER2	0xEDA0_00FE	HWInit	Host Controller Version Register (Channel 2)	0x0401

9.1 SDMA SYSTEM ADDRESS REGISTER MA SYSTEM ADDRESS REGISTER SDMA SYSTEM ADDRESS REGISTER

SDMA System Address Register

- SDMASYSAD0, R/W, Address = 0xED80_0000
- SDMASYSAD1, R/W, Address = 0xED90_0000
- SDMASYSAD2, R/W, Address = 0xEDA0_0000

This register contains the physical system memory address used for DMA transfers.

SDMASYSAD	Bit	Description	Reset Value
SDMASYSAD	[31:0]	SDMA System Address This register contains the system memory address for a DMA transfer. If the Host Controller stops a DMA transfer, this register points to the system address of the next contiguous data position. It is accessed if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver initializes this register before starting a DMA transaction. After DMA has stopped, the next system address of the next contiguous data position is read from this register. The DMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the *Block Size* register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver set the next system address of the next data position to this register. If the most upper byte of this register (003h) is written, the Host Controller restarts the DMA transfer. If restarting DMA by the Resume command or by setting Continue Request in the *Block Gap Control* register, the Host Controller starts at the next contiguous address stored here in the *System Address* register.	0x00

9.2 BLOCK SIZE REGISTER

Host DMA Buffer Boundary and Transfer Block Size Register

- BLKSIZE0, R/W, Address = 0xED80_0004
- BLKSIZE1, R/W, Address = 0xED90_0004
- BLKSIZE2, R/W, Address = 0xEDA0_0004

This register is used to configure the number of bytes in a data block.

BLKSIZE	Bit	Description	Reset Value
	[15]	Reserved	0
BUF BOUND	[14:12]	**Host DMA Buffer Boundary** The large contiguous memory space may not be available in the virtual memory system. To perform long SDMA transfer, *System Address* register is updated at every system memory boundary during SDMA transfer. These bits specify the size of contiguous buffer in the system memory. The SDMA transfer waits at the every boundary specified by these fields and the Host Controller generates the **DMA Interrupt** to request the Host Driver to update the SDMA *System Address* register. At the end of transfer, the Host Controller issue or not issue **DMA Interrupt**. In particular, **DMA Interrupt** is not issued after **Transfer Complete Interrupt** is issued. If this register is set to 0 (buffer size = 4K bytes), lower 12-bit of byte address points data in the contiguous buffer and the upper 20-bit points the location of the buffer in the system memory. The DMA transfer stops if the Host Controller detects carry out of the address from bit 11 to 12. These bits are supported if the **SDMA Support** in the *Capabilities* register is set to 1 and this function is active if **DMA Enable** in the *Transfer Mode* register is set to 1. 000b = 4K bytes (Detects A11 carry out) 001b = 8K bytes (Detects A12 carry out) 010b = 16K Bytes (Detects A13 carry out) 011b = 32K Bytes (Detects A14 carry out) 100b = 64K bytes (Detects A15 carry out) 101b = 128K Bytes (Detects A16 carry out) 110b = 256K Bytes (Detects A17 carry out) 111b = 512K Bytes (Detects A18 carry out)	0

BLKSIZE	[11:0]	**Transfer Block Size** This register specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. Values ranging from 1 up to maximum buffer size are set. In case of memory, it is set up to 512 bytes. It is accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers return an invalid value, and write operations are ignored. 0200h = 512 Bytes 01FFh = 511 Bytes …… 0004h = 4 Bytes 0003h = 3 Bytes 0002h = 2 Bytes 0001h = 1 Byte 0000h = No data transfer	0

9.3 BLOCK COUNT REGISTER

Blocks Count for Current Transfer

- BLKCNT0, R/W, Address = 0xED80_0006
- BLKCNT1, R/W, Address = 0xED90_0006
- BLKCNT2, R/W, Address = 0xEDA0_0006

This register is used to configure the number of data blocks.

BLKCNT	Bit	Description	Reset Value
BLKCNT	[15:0]	Blocks Count For Current Transfer This register is enabled if Block Count Enable in the *Transfer Mode* register is set to 1 and is valid only for multiple block transfers. The Host Driver sets this register to a value between 1 and the maximum block count. The Host Controller decrements the block count after each block transfer and stops if the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register must be accessed if no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register returns an invalid value and write operations are ignored. If saving transfer context as a result of a Suspend command, the number of blocks yet to be transferred is determined by reading this register. If restoring transfer context prior to issuing a Resume command, the Host Driver restores the previously saved block count. FFFFh = 65535 blocks … … 0002h = 2 blocks 0001h = 1 block 0000h = Stop Count	0

9.4 ARGUMENT REGISTER

Command Argument Register

- ARGUMENT0, R/W, Address = 0xED80_0008
- ARGUMENT1, R/W, Address = 0xED90_0008
- ARGUMENT2, R/W, Address = 0xEDA0_0008

This register contains the SD Command Argument.

ARGUMENT	Bit	Description	Reset Value
ARGUMENT	[31:0]	Command Argument The SD Command Argument is specified as bit [39:8] of Command-Format in the SD Memory Card Physical Layer Specification.	0

9.5 TRANSFER MODE REGISTER

Transfer Mode Register Setting

- TRNMOD0, R/W, Address = 0xED80_000C
- TRNMOD1, R/W, Address = 0xED90_000C
- TRNMOD2, R/W, Address = 0xEDA0_000C

This register is used to control the data transfer operations. The Host Driver sets this register before issuing a command which transfers data (Refer to Data Present Select in the Command register), or before issuing a Resume command. The Host Driver saves the value of this register if data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, the Host Controller implements write protection for this register during data transactions. Writes to this register is ignored if the Command Inhibit (DAT) in Present State register is 1.

TRNMOD	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
CCSCON	[9:8]	Command Completion Signal Control '00' = No CCS Operation (Normal operation and No CE-ATA mode) '01' = Read or Write data transfer CCS enable (Only CE-ATA mode) '10' = Without data transfer CCS enable (Only CE-ATA mode) '11' = Abort Completion Signal (ACS) generation (Only CE-ATA mode)	0
Reserved	[7:6]	Reserved	0
MUL1SIN0	[5]	Multi/ Single Block Select This bit enables multiple block DAT line data transfers. For any other commands, this bit is set to 0. If this bit is 0, it is not mandatory to set the *Block Count* register. (Refer to the Table below "Determination of Transfer Type") 1 = Multiple Block 0 = Single Block	0

TRNMOD	Bit	Description	Reset Value
RD1WT0	[4]	Data Transfer Direction Select This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the SD Host Controller and it is set to 0 for all other commands. 1 = Read (Card to Host) 0 = Write (Host to Card)	0
Reserved	[3]	Reserved	0
ENACMD12	[2]	Auto CMD12 Enable Multiple block transfers for memory require CMD12 to stop the transaction. If this bit is set to 1 and last block transfer is complete, the Host Controller issues CMD12 automatically. The Host Driver does not set this bit to issue commands that do not require CMD12 to stop data transfer. 1 = Enable 0 = Disable	0
ENBLKCNT	[1]	Block Count Enable This bit is used to enable the *Block Count* register, which is only relevant for multiple block transfers. If this bit is 0, the *Block Count* register is disabled, which is useful in executing an infinite transfer. (Refer to the Table below "Determination of Transfer Type") 1 = Enable 0 = Disable	0
ENDMA	[0]	DMA Enable This bit enables DMA functionality. DMA is enabled if it is supported as indicated in the DMA Support in the *Capabilities* register. If DMA is not supported, this bit is meaningless and always read 0. If this bit is set to 1, a DMA operation begins if the Host Driver writes to the upper byte of *Command* register (00Fh). 1 = Enable 0 = Disable	0

Table below shows the summary of how register settings determine types of data transfer.

Determination of Transfer Type

Multi/Single Block Select	Block Count Enable	*Block Count*	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

NOTE: For CE-ATA access, (Auto) CMD12 must be issued after Command Completion Signal Disable.

9.6 COMMAND REGISTER

Command Register

- CMDREG0, R/W, Address = 0xED80_000E
- CMDREG1, R/W, Address = 0xED90_000E
- CMDREG2, R/W, Address = 0xEDA0_000E

This register contains the SD Command Argument.

The Host Driver checks the Command Inhibit (DAT) bit and Command Inhibit (CMD) bit in the Present State register before writing to this register. Writing to the upper byte of this register triggers SD command generation. The Host Driver is responsible to write this register because the Host Controller does not protect the writing if Command Inhibit (CMD) is set.

CMDREG	Bit	Description	Reset Value
Reserved	[15:14]	Reserved	
CMDIDX	[13:8]	Command Index These bits are set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.	
CMDTYP	[7:6]	Command Type There are three types of special commands: Suspend, Resume and Abort. These bits are set to 00b for all other commands. • Suspend Command If the Suspend command succeeds, the Host Controller assumes that the SD Bus has been released and it is possible to issue the next command, which uses the *DAT* line. The Host Controller de-asserts Read Wait for read transactions and stops checking busy for write transactions. The interrupt cycle starts, in 4-bit mode. If the Suspend command fails, the Host Controller maintains its current state, and the Host Driver restarts the transfer by setting Continue Request in the *Block Gap Control* register. • Resume Command The Host Driver re-starts the data transfer by restoring the registers in the range of 000-00Dh (Refer to Suspend and Resume mechanism). The Host Controller checks for busy before starting write transfers. • Abort Command If this command is set when executing a read transfer, the Host Controller stops reads to the buffer. If this command is set while executing a write transfer, the Host Controller stops driving the *DAT* line. After issuing the Abort command, the Host Driver must issue a software reset (Refer to Abort Transaction). 11b = Abort CMD12, CMD52 for writing "I/O Abort" in CCCR 10b = Resume CMD52 for writing "Function Select" in CCCR 01b = Suspend CMD52 for writing "Bus Suspend" in CCCR 00b = Normal Other commands	

CMDREG	Bit	Description	Reset Value
DATAPRNT	[5]	Data Present Select This bit is set to 1 to indicate that data is present and transferred using the *DAT* line. It is set to 0 for the following: (1) Commands using only *CMD* line (ex. CMD52). (2) Commands with no data transfer but using busy signal on *DAT*[0] line (R1b or R5b ex. CMD38) (3) Resume command 1 = Data Present 0 = No Data Present	
ENCMDIDX	[4]	Command Index Check Enable If this bit is set to 1, the Host Controller checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 1 = Enable 0 = Disable	
ENC MDCRC	[3]	Command CRC Check Enable If this bit is set to 1, the Host Controller checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. 1 = Enable 0 = Disable	
Reserved	[2]	Reserved	
RSPTYP	[1:0]	Response Type Select 00 = No Response 01 = Response Length 136 10 = Response Length 48 11 = Response Length 48 check Busy after response	

Relation Between Parameters and the Name of Response Type

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R6, R5
11	1	1	R1b, R5b

These bits determine Response types.

NOTES:

1. In the SDIO specification, response type notation of R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to specify the Host Controller checks busy after receiving response. For example, usually CMD52 is used as R5 but I/O abort command is used as R5b.
2. For CMD52 to read BS after writing "Bus Suspend," Command Type must be "Suspend" as well.

9.7 RESPONSE REGISTER

This register is used to store responses from SD cards.

- Response Register 0 (Channel 0) (RSPREG0_0, ROC, Address = 0xED80_0010)
- Response Register 1 (Channel 0) (RSPREG1_0, ROC, Address = 0xED80_0014)
- Response Register 2 (Channel 0) (RSPREG2_0, ROC, Address = 0xED80_0018)
- Response Register 3 (Channel 0) (RSPREG3_0, ROC, Address = 0xED80_001C)

- Response Register 0 (Channel 1) (RSPREG0_1, ROC, Address = 0xED90_0010)
- Response Register 1 (Channel 1) (RSPREG1_1, ROC, Address = 0xED90_0014)
- Response Register 2 (Channel 1) (RSPREG2_1, ROC, Address = 0xED90_0018)
- Response Register 3 (Channel 1) (RSPREG3_1, ROC, Address = 0xED90_001C)

- Response Register 0 (Channel 2) (RSPREG0_2, ROC, Address = 0xEDA0_0010)
- Response Register 1 (Channel 2) (RSPREG1_2, ROC, Address = 0xEDA0_0014)
- Response Register 2 (Channel 2) (RSPREG2_2, ROC, Address = 0xEDA0_0018)
- Response Register 3 (Channel 2) (RSPREG3_2, ROC, Address = 0xEDA0_001C)

RSPREG	Bit	Description	Reset Value
CMDRSP	[127:0]	Command Response The Table below describes the mapping of command responses from the SD Bus to this register for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit range within the *Response* register. 128-bit Response bit order : {RSPREG3, RSPREG2, RSPREG1, RSPREG0}	

Response Bit Definition for Each Response Type.

Kind of Response	Meaning of Response	Response Field	Response Register
R1, R1b (normal response)	Card Status	R [39:8]	REP [31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R [39:8]	REP [127:96]
R2 (CID, CSD register)	CID or CSD reg. incl.	R [127:8]	REP [119:0]
R3 (OCR register)	OCR register for memory	R [39:8]	REP [31:0]
R4 (OCR register)	OCR register for I/O etc	R [39:8]	REP [31:0]
R5,R5b	SDIO response	R [39:8]	REP [31:0]
R6 (Published RCA response)	New published RCA[31:16] etc	R [39:8]	REP [31:0]

The Response Field indicates bit positions of "Responses" defined in the PHYSICAL LAYER SPECIFICATION Version 1.01. The Table (above) shows that most responses with a length of 48 (R[47:0]) have 32 bits of the response data (R[39:8]) stored in the Response register at REP[31:0]. Responses of type R1b (Auto CMD12 responses) have response data bits R[39:8] stored in the Response register at REP[127:96]. Responses with length 136 (R[135:0]) have 120 bits of the response data (R[127:8]) stored in the Response register at REP[119:0].

To be able to read the response status efficiently, the Host Controller only stores part of the response data in the Response register. This enables the Host Driver to efficiently read 32 bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the Host Controller (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Command register) and generate an error interrupt if an error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the Host Controller checks R[47:1], and if the response length is 136 the Host Controller checks R[119:1].

Since the Host Controller may have a multiple block data DAT line transfer executing concurrently with a CMD_wo_DAT command, the Host Controller stores the Auto CMD12 response in the upper bits (REP[127:96]) of the Response register. The CMD_wo_DAT response is stored in REP[31:0]. This allows the Host Controller to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa.

If the Host Controller modifies part of the Response register, as shown in the Table above, it shall preserve the unmodified bits.

9.8 BUFFER DATA PORT REGISTER

Buffer Data Register

- BDATA0, R/W, Address = 0xED80_0020
- BDATA1, R/W, Address = 0xED90_0020
- BDATA2, R/W, Address = 0xEDA0_0020

32-bit data port register to access internal buffer.

BDATA	Bit	Description	Reset Value
BUFDAT	[31:0]	Buffer Data The Host Controller buffer is accessed through this 32-bit single port SRAM memory. Write and Read memories are separated.	Not fixed

NOTE: Detailed documents are to be copied from SD Host Standard Specification.

9.9 PRESENT STATE REGISTER

Present State Register

- PRNSTS0, R/ROC, Address = 0xED80_0024
- PRNSTS1, R/ROC, Address = 0xED90_0024
- PRNSTS2, R/ROC, Address = 0xEDA0_0024

This register contains the SD Command Argument.

PRNSTS	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0
PRNTCMD	[24]	CMD Line Signal Level (RO) This status is used to check the *CMD* line level to recover from errors, and for debugging. **Note:** *CMD* port is mapped to *SD0_CMD* pin	0
PRNTDAT	[23:20]	DAT[3:0] Line Signal Level (RO) This status is used to check the *DAT* line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from *DAT*[0]. D23 : DAT[3] D22 : DAT[2] D21 : DAT[1] D20 : DAT[0] **Note:** *DAT* port is mapped to *SD0_DAT* pin	Line State
Reserved -	[19]	Reserved	1
PRNTCD	[18]	Card Detect Pin Level (RO) This bit reflects the inverse value of the *SDCD#* pin. Debouncing is not performed on this bit. This bit is valid if Card State Stable is set to 1, but it is not guaranteed because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. 1 = Card present (*SDCD#*=0) 0 = No card present (*SDCD#*=1) **Note:** *SDCD#* port is mapped to *SD0_nCD* pin, SD2_nCD (Channel 2) port is fixed to LOW.	Line State
STBLCARD	[17]	Card State Stable (RO) This bit is used for testing. If this bit is 0, the Card Detect Pin Level is not stable. If this bit is set to 1, it means the Card Detect Pin Level is stable. No Card state is detected by this bit if set to 1 and Card Inserted is set to 0. The Software Reset For All in the Software Reset register does not affect this bit. 1 = No Card or Inserted 0 = Reset or Debouncing	1 (After Reset)
INSCARD	[16]	Card Inserted (RO) This bit indicates whether a card has been inserted. The Host Controller debounce this signal so that the Host Driver does not require to wait for it to stabilize. Changing from 0 to 1 generates a Card Insertion interrupt in the Normal Interrupt Status register and changing from 1 to 0 generates a Card Removal interrupt in the Normal Interrupt Status register. The Software Reset For All in the	0

PRNSTS	Bit	Description	Reset Value
		Software Reset register does not affect this bit. If a card is removed while its power is on and its clock is oscillating, the Host Controller clears SD Bus Power in the Power Control register and SD Clock Enable in the Clock Control register. If this bit is changed from 1 to 0, the Host Controller immediately stops driving CMD and DAT[3:0] (tri-state). In addition, the Host Driver must clear the Host Controller by the Software Reset For All in Software Reset register. The card detect is active regardless of the SD Bus Power. 1 = Card Inserted 0 = Reset or Debouncing or No Card	
Reserved	[15:12]	Reserved	
BUFRDRDY	[11]	Buffer Read Enable (ROC) This status is used for non-DMA read transfers. The Host Controller implements multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer status. If this bit is 1, readable data exists in the buffer. A change of this bit from 1 to 0 occurs if all the block data is read from the buffer. A change of this bit from 0 to 1 occurs if block data is ready in the buffer and generates the Buffer Read Ready interrupt. 1 = Enables Read 0 = Disables Read	0
BUFWTRDY	[10]	Buffer Write Enable (ROC) This status is used for non-DMA write transfers. The Host Controller implements multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, data is written to the buffer. A change of this bit from 1 to 0 occurs if all the block data is written to the buffer. A change of this bit from 0 to 1 occurs if top of block data is written to the buffer and generates the Buffer Write Ready interrupt. 1 = Write enable 0 = Write disable	0
RDTRANACT	[9]	Read Transfer Active (ROC) This status is used to detect completion of a read transfer. This bit is set to 1 for either of the following conditions: (1) After the end bit of the read command. (2) If writing a 1 to Continue Request in the Block Gap Control register to restart a read transfer. This bit is cleared to 0 for either of the following conditions: (1) If the last data block as specified by block length is transferred to the System. (2) If all valid data blocks have been transferred to the System and no current block transfers are being sent as a result of the Stop At Block Gap Request being set to 1. A Transfer Complete interrupt is generated if this bit changes to 0. 1 = Transferring data 0 = No valid data	0

PRNSTS	Bit	Description	Reset Value
WTTRANACT	[8]	Write Transfer Active (ROC) This status indicates that a write transfer is active. If this bit is 0, it means no valid write data exists in the Host Controller. This bit is set in either of the following cases: (1) After the end bit of the write command. (2) If 1 is written to Continue Request in the *Block Gap Control* register to restart a write transfer. This bit is cleared in either of the following cases: (1) After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) (2) After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request. During a write transaction, if this bit is changed to 0 a Block Gap Event interrupt is generated, as result of the Stop At Block Gap Request being set. This status is useful for the Host Driver to determine the right time to issue commands during write busy. 1 = Transferring data 0 = No valid data	0
Reserved	[7:3]	Reserved	0
DATLINEACT	[2]	DAT Line Active (ROC) This bit indicates whether one of the *DAT* line on SD Bus is in use. (a) In the case of read transactions This status indicates if a read transfer is executing on the SD Bus. Change in this value from 1 to 0 between data blocks generates a Block Gap Event interrupt in the *Normal Interrupt Status* register. This bit is set in either of the following cases: (1) After the end bit of the read command. (2) If 1 is written to Continue Request in the *Block Gap Control* register to restart a read transfer. This bit is cleared in either of the following cases: (1) If the end bit of the last data block is sent from the SD Bus to the Host Controller. (2) When beginning a wait read transfer at a stop at the block gap initiated by a Stop At Block Gap Request. The Host Controller waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the Host Controller waits for current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use suspend/ resume function. (b) In the case of write transactions This status indicates that a write transfer is executing on the SD Bus. Change in this value from 1 to 0 generates a Transfer Complete interrupt in the *Normal Interrupt Status* register.	0

PRNSTS	Bit	Description	Reset Value
		This bit is set in either of the following cases: (1) After the end bit of the write command. (2) If 1 is written to Continue Request in the *Block Gap Control* register to continue a write transfer. This bit is cleared in either of the following cases: (1) If the SD card releases write busy of the last data block the Host Controller detects if output is not busy. If SD card does not drive busy signal for 8 SD Clocks, the Host Controller considers the card drive "Not Busy". (2) If the SD card releases write busy prior to waiting for write transfer as a result of a Stop At Block Gap Request. 1 = DAT Line Active 0 = DAT Line Inactive	
CMDINHDAT	[1]	Command Inhibit (DAT) (ROC) (ROC) This status bit is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this bit is 0, it indicates the Host Controller can issue the next SD Command. Commands with busy signal belong to Command Inhibit (DAT) (ex. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the *Normal Interrupt Status* register. **Note:** The SD Host Driver saves registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0. 1 = Cannot issue command which uses the *DAT* line 0 = Issues command which uses the *DAT* line	0
CMDINHCMD	[0]	Command Inhibit (CMD) (ROC) If this bit is 0, it indicates the *CMD* line is not in use and the Host Controller issues a SD Command using the *CMD* line. This bit is set immediately after the *Command* register (00Fh) is written. This bit is cleared if the command response is received. Even if the Command Inhibit (DAT) is set to 1, Commands using only the *CMD* line is issued if this bit is 0. Changing from 1 to 0 generates a Command Complete interrupt in the *Normal Interrupt Status* register. If the Host Controller cannot issue the command because of a command conflict error (Refer to Command CRC Error) or because of Command Not Issued By Auto CMD12 Error, this bit remains 1 and the Command Complete is not set. Status issuing Auto CMD12 is not read from this bit. 1 = Cannot issue command 0 = Issues command using only *CMD* line	0

NOTE: Buffer Write Enable in Present register must not be asserted for DMA transfers since it generates Buffer Write Ready interrupt

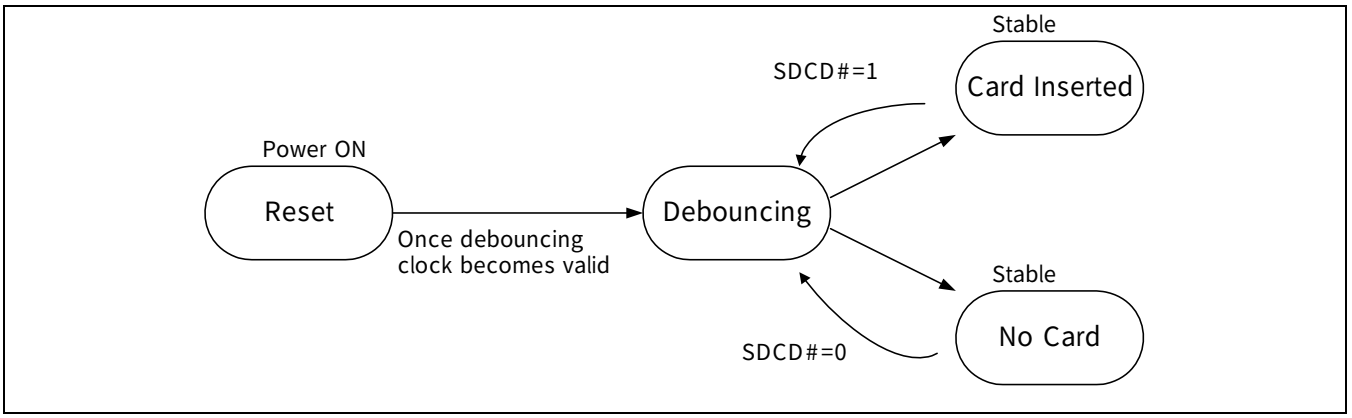


Figure 8.12-17 Card Detect State

The above Figure 8.12-17 shows the state definitions of hardware that handles "Debouncing".

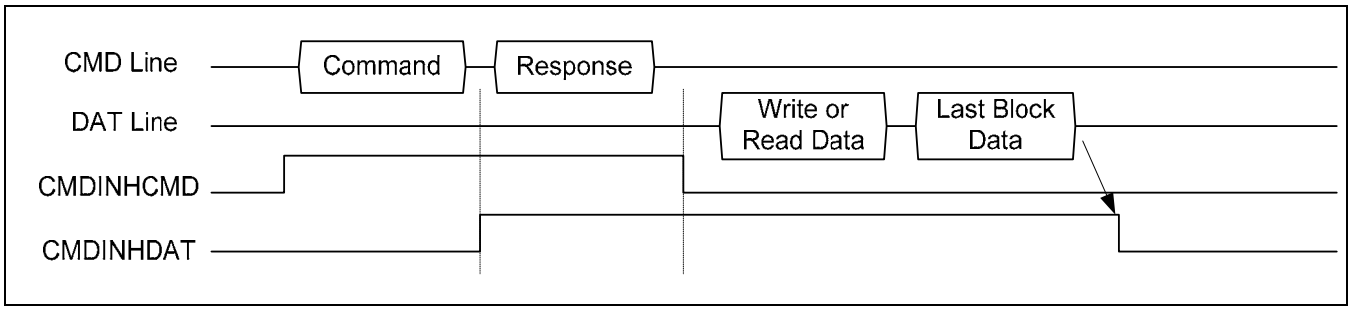


Figure 8.12-18 Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with Data Transfer

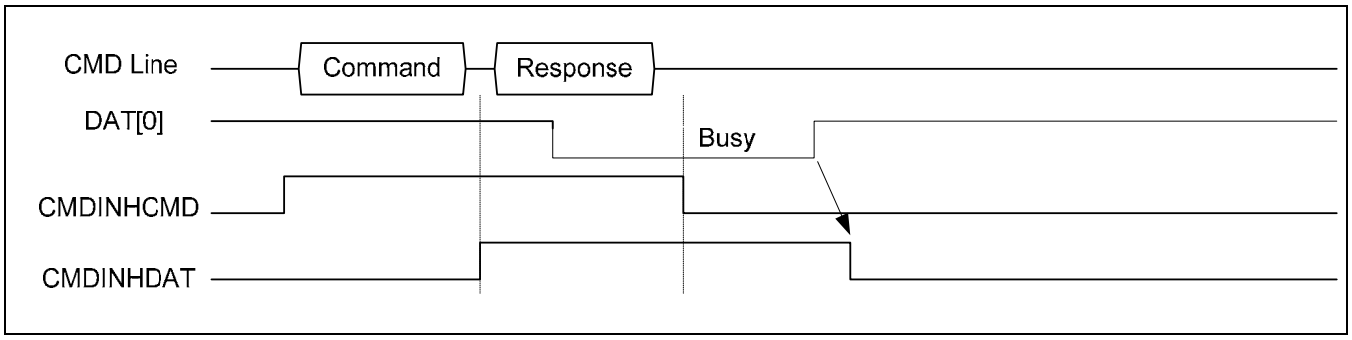


Figure 8.12-19 Timing of Command Inhibit (DAT) for the Case of Response with Busy

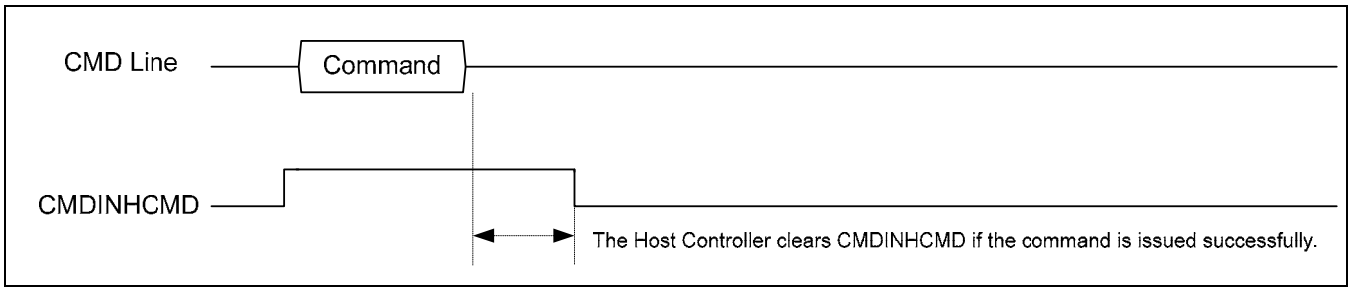


Figure 8.12-20 Timing of Command Inhibit (CMD) for the Case of No Response Command

9.10 HOST CONTROL REGISTER

Host Control Register

- HOSTCTL0, R/W, Address = 0xED80_0028
- HOSTCTL1, R/W, Address = 0xED90_0028
- HOSTCTL2, R/W, Address = 0xEDA0_0028

This register contains the SD Command Argument.

HOSTCTL	Bit	Description	Reset Value
- Reserved	[7]	Reserved This field should be fixed to LOW	0
- Reserved	[6]	Reserved This field should be fixed to LOW	0
WIDE8	[5]	Extended Data Transfer Width (It is for MMC 8-bit card.) '1' = 8-bit operation '0' = Bit width is designated by the bit 1 (Data Transfer Width)	0
DMASEL	[4:3]	DMA Select One of supported DMA modes can be selected. The host driver checks support of DMA modes by referring the *Capabilities* register. Use of selected DMA is determined by DMA Enable of the *Transfer Mode* register. 00 = Selects SDMA 01 = Reserved 10 = Selects 32-bit Address ADMA2 11 = Selects 64-bit Address ADMA2 (Not supported)	0
ENHIGHSPD	[2]	High Speed Enable This bit is optional. Before setting this bit, the Host Driver checks the High Speed Support in the *Capabilities* register. If this bit is set to 0 (default), the Host Controller outputs *CMD* line and *DAT* lines at the falling edge of the SD Clock (up to 25MHz). If this bit is set to 1, the Host Controller outputs *CMD* line and *DAT* lines at the rising edge of the SD Clock (up to 50MHz). '1' = High Speed mode '0' = Normal Speed mode	0
WIDE4	[1]	**Data Transfer Width** This bit selects the data width of the Host Controller. The Host Driver sets it to match the data width of the SD card. '1' = 4-bit mode '0' = 1-bit mode	0
- Reserved	[0]	**Reserved**	0

NOTE: Card Detect Pin Level does not simply reflect SDCD# pin, but selects from SDCD, DAT[3], or CDTestlvl depending on CDSSigSel and SDCDSel values.

9.11 POWER CONTROL REGISTER

Power Control Register

- PWRCON0, R/W, Address = 0xED80_0029
- PWRCON1, R/W, Address = 0xED90_0029
- PWRCON2, R/W, Address = 0xEDA0_0029

This register contains the SD Command Argument.

PWRCON	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	
SELPWRLVL	[3:1]	SD Bus Voltage Select If these bits are set, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver checks the Voltage Support bits in the Capabilities register. If an unsupported voltage is selected, the Host System does not supply SD Bus voltage. '111b' = 3.3V (Typ.) '110b' = 3.0V (Typ.) '101b' = 1.8V (Typ.) '100b' – '000b' = Reserved	0
PWRON	[0]	SD Bus Power Before setting this bit, the SD Host Driver sets SD Bus Voltage Select. If the Host Controller detects the No Card state, this bit is cleared. If this bit is cleared, the Host Controller immediately stops driving CMD and DAT[3:0] (tri-state) and drive SDCLK to low level. '1' = Power on '0' = Power off	0

9.12 BLOCK GAP CONTROL REGISTER

Block Gap Control Register

- BLKGAP0, R/W, 0xED80_002A
- BLKGAP1, R/W, 0xED90_002A
- BLKGAP2, R/W, 0xEDA0_002A

This register contains the SD Command Argument.

BLKGAP	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0
ENINTB GAP	[3]	Interrupt At Block Gap This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. If set to 1, it enables interrupt detection at the block gap for a multiple block transfer. If set to 0, it disables interrupt detection during a multiple block transfer. If the SD card cannot signal an interrupt during a multiple block transfer, this bit must be set to 0. If the Host Driver detects an SD card insertion, it sets this bit according to the CCCR of the SDIO card. (RW) '1' = Enables, '0' = Disables **Note:** Interrupt at Block Gap operation is not supported in S3C6410 controller, it should be fixed to 0.	0
ENRWAIT	[2]	Read Wait Control The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise the Host Controller has to stop the SD Clock to hold read data, which restricts commands generation. If the Host Driver detects an SD card insertion, it sets this bit according to the CCCR of the SDIO card. If the card does not support read wait, this bit will never be set to 1 otherwise DAT line conflict might occur. If this bit is set to 0, Suspend/ Resume cannot be supported. (RW) '1' = Enables Read Wait Control, '0' = Disables Read Wait Control	0
CONTREQ	[1]	Continue Request This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit 1 to restart the transfer. The Host Controller automatically clears this bit in either of the following cases: (1) If a read transaction, the DAT Line Active changes from 0 to 1 as a read transaction restarts. (2) If a write transaction, the Write Transfer Active changes from 0 to 1 as the write transaction restarts. Therefore it is not necessary for Host Driver to set this bit to 0. If Stop At Block Gap Request is set to 1, any write to this bit is ignored. (RWAC) '1' = Restart, '0' = Not affect	0
STOPB GAP	[0]	Stop At Block Gap Request This bit is used to stop executing a transaction at the next block gap for both	0

BLKGAP	Bit	Description	Reset Value
		DMA and non-DMA transfers. Until the Transfer Complete is set to 1, indicating a transfer completion the Host Driver leaves this bit set to 1. Clearing both the Stop At Block Gap Request and Continue Request does not restart the transaction. Read Wait stops the read transaction at the block gap. The Host Controller honours Stop At Block Gap Request for write transfers, but for read transfers it requires that the SD card support Read Wait. Therefore the Host Driver does not set this bit during read transfers unless the SD card supports Read Wait and has set Read Wait Control to 1. In the case of write transfers in which the Host Driver writes data to the *Buffer Data Port* register, the Host Driver sets this bit after all block data is written. If this bit is set to 1, the Host Driver does not write data to *Buffer Data Port* register. This bit affects Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the *Present State* register. Regarding detailed control of bits D01 and D00. (RW) '1' = Stop '0' = Transfer	

There are three cases to restart the transfer after stop at the block gap. Appropriate case depends on whether the Host Controller issues a Suspend command or the SD card accepts the Suspend command.

Cases are as follows:

(1) If the Host Driver does not issue a Suspend command, the **Continue Request** restarts the transfer.

(2) If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command restarts the transfer.

(3) If the Host Driver issues a Suspend command and the SD card does not accept it, the **Continue Request** restarts the transfer.

Any time Stop At Block Gap Request stops the data transfer, the Host Driver waits for Transfer Complete (in the Normal Interrupt Status register) before attempting to restart the transfer. If the data transfer by Continue Request is restarted, the Host Driver clears **Stop At Block Gap Request** before or simultaneously.

NOTE:

After setting **Stop At Block Gap Request** field, it must not be cleared unless Block Gap Event or Transfer Complete interrupt occurs. Otherwise, the module hangs.

9.13 WAKEUP CONTROL REGISTER

Wakeup Control Register

- WAKCON0, R/W, Address = 0xED80_002B
- WAKCON1, R/W, Address = 0xED90_002B
- WAKCON2, R/W, Address = 0xEDA0_002B

This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver sets SD Bus Power to 1 in the Power Control Register to maintains voltage on the SD Bus, if wakeup event via Card Interrupt is desired.

WAKCON	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0
StaWakeup	[3]	**Wakeup Event Status** This status is set if the Card Inserted/ Removed or Card Interrupt Stop mode Wakeup Event Occurred. **(ROC/RW1C)** '1' = Wakeup Interrupt Occurred '0' = Wakeup Interrupt Not occurred or Cleared.	0
ENWKUPREM	[2]	Wakeup Event Enable On SD Card Removal This bit enables wakeup event via Card Removal assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit. (RW) '1' = Enables '0' = Disable s	0
ENWKUPINS	[1]	Wakeup Event Enable On SD Card Insertion This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this bit. (RW) '1' = Enable s '0' = Disable	0
ENWKUPINT	[0]	Wakeup Event Enable On Card Interrupt This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. This bit is set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. (RW) '1' = Enables '0' = Disables	0

9.14 CLOCK CONTROL REGISTER

Clock Control Register

- CLKCON0, R/W, Address = 0xED80_002C
- CLKCON1, R/W, Address = 0xED90_002C
- CLKCON2, R/W, Address = 0xEDA0_002C

At the initialization of the Host Controller, the Host Driver sets the SDCLK Frequency Select according to the Capabilities register.

<table>
<tr><th>CLKCON</th><th>Bit</th><th>Description</th><th>Reset Value</th></tr>
<tr><td>SELF REQ</td><td>[15:8]</td><td>SDCLK Frequency Select
This register is used to select the frequency of SDCLK pin. The frequency is not programmed directly; rather this register holds the divisor of the Base Clock Frequency For SD Clock in the Capabilities register. Only the following settings are allowed.
<table>
<tr><td>80h</td><td>base clock divided by 256</td></tr>
<tr><td>40h</td><td>base clock divided by 128</td></tr>
<tr><td>20h</td><td>base clock divided by 64</td></tr>
<tr><td>10h</td><td>base clock divided by 32</td></tr>
<tr><td>08h</td><td>base clock divided by 16</td></tr>
<tr><td>04h</td><td>base clock divided by 8</td></tr>
<tr><td>02h</td><td>base clock divided by 4</td></tr>
<tr><td>01h</td><td>base clock divided by 2</td></tr>
<tr><td>00h</td><td>base clock (10MHz-63MHz)</td></tr>
</table>
Setting 00h specifies the highest frequency of the SD Clock. Setting multiple bits, the most significant bit is used as the divisor. But multiple bits must not be set. The two default divider values are calculated by the frequency that is defined by the Base Clock Frequency For SD Clock in the Capabilities register.
(1) 25MHz divider value
(2) 400kHz divider value
According to the SD Physical Specification Version 1.01 and the SDIO Card Specification Version 1.0, maximum SD Clock frequency is 25MHz, and never exceeds this limit.
The frequency of SDCLK is set by the following formula:
Clock Frequency = (Base Clock)/ divisor
Therefore, select the smallest possible divisor which results in a clock frequency that is less than or equal to the target frequency.
For example, if the Base Clock Frequency For SD Clock in the Capabilities register has the value 33MHz, and the target frequency is 25MHz, then selecting the divisor value of 01h yields 16.5MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400kHz, the divisor value of 40h yields the optimal clock value of 258kHz.</td><td>0</td></tr>
</table>

CLKCON	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	
STBLEXTCLK	[3]	**External Clock Stable** This bit is set to 1 if SD Clock output is stable after writing to **SD Clock Enable** in this register to 1. The SD Host Driver waits to issue command to start until this bit is set to 1. **(ROC)** '1' = Ready '0' = Not Ready	0
ENSDCLK	[2]	SD Clock Enable The Host Controller stops *SDCLK* if this bit is written to 0. SDCLK Frequency Select changes if this bit is 0. Then, the Host Controller maintains the same clock frequency until *SDCLK* is stopped (Stop at *SDCLK*=0). If the Card Inserted in the *Present State register* is cleared, this clears the bit (RW). '1' = Enables '0' = Disables	0
STBLINTCLK	[1]	Internal Clock Stable This bit is set to 1 if SD Clock is stable after writing to Internal Clock Enable in this register to 1. The SD Host Driver waits to set SD Clock Enable until this bit is set to 1. **Note:** This is useful if PLL is used for a clock oscillator that requires setup time. (ROC) '1' = Ready '0' = Not Ready	0
ENINTCLK	[0]	Internal Clock Enable This bit is set to 0 if the Host Driver is not using the Host Controller or the Host Controller awaits a wakeup interrupt. The Host Controller must stop its internal clock to go at very low power state. Still, registers is able to be read and written. Clock starts to oscillate when this bit is set to 1. If clock oscillation is stable, the Host Controller can be set Internal Clock Stable in this register to 1. This bit does not affect card detection. (RW) '1' = Oscillate '0' = Stop	

9.15 TIMEOUT CONTROL REGISTER

Timeout Control Register

- TIMEOUTCON0, R/W, Address = 0xED80_002E
- TIMEOUTCON1, R/W, Address = 0xED90_002E
- TIMEOUTCON2, R/W, Address = 0xEDA0_002E

At the initialization of the Host Controller, the Host Driver sets the Data Timeout Counter Value according to the Capabilities register.

TIMEOUTCON	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0
TIMEOUTCON	[3:0]	**Data Timeout Counter Value** This value determines the interval by which DAT line timeouts are detected. Refer to the **Data Timeout Error** in the *Error Interrupt Status* register for information on factors that dictate timeout generation. Timeout clock frequency is generated by dividing the base clock TMCLK value by this value. While setting this register, prevent inadvertent timeout events by clearing the **Data Timeout Error Status Enable** (in the *Error Interrupt tatus Enable* register) 1111b Reserved 1110b TMCLK x 2^{27} 1101b TMCLK x 2^{26} …… … 0001b TMCLK x 2^{14} 0000b TMCLK x 2^{13}	0

9.16 SOFTWARE RESET REGISTER

Software Reset Register

- SWRST0, R/W, Address = 0xED80_002F
- SWRST1, R/W, Address = 0xED90_002F
- SWRST2, R/W, Address = 0xEDA0_002F

A reset pulse is generated when writing 1 to each bit of this register. After completing the reset, the Host Controller clears each bit. Because it takes time to complete software reset, the SD Host Driver confirms that these bits are 0.

SWRST	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0
RSTDAT	[2]	**Software Reset For DAT Line** Only part of data circuit is reset. DMA circuit is also reset. (RWAC) The following registers and bits are cleared by this bit: *Buffer Data Port* register Buffer is cleared and initialized. *Present State* register **Buffer Read Enable** **Buffer Write Enable** **Read Transfer Active** **Write Transfer Active** **DAT Line Active** **Command Inhibit (DAT)** *Block Gap Control* register **Continue Request** **Stop At Block Gap Request** *Normal Interrupt Status* register **Buffer Read Ready** **Buffer Write Ready** **DMA Interrupt** **Block Gap Event** **Transfer Complete** '1' = Reset '0' = Work	0
RSTCMD	[1]	Software Reset For CMD Line Only part of command circuit is reset. (RWAC). The following registers and bits are cleared by this bit: *Present State* register Command Inhibit (CMD) *Normal Interrupt Status* register Command Complete '1' = Reset '0' = Work	0
RSTALL	[0]	Software Reset For All This reset affects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC, HWInit are cleared	0

SWRST	Bit	Description	Reset Value
		to 0. During its initialization, the Host Driver sets this bit to 1 to reset the Host Controller. The Host Controller reset this bit to 0 if capabilities registers are valid and the Host Driver reads them. If this bit is set to 1, the SD card resets itself and must be reinitialized by the Host Driver. (RWAC) '1' = Reset '0' = Work	

9.17 NORMAL INTERRUPT STATUS REGISTER

Normal Interrupt Status Register

- NORINTSTS0, ROC/RW1C, Address = 0xED80_0030
- NORINTSTS1, ROC/RW1C, Address = 0xED90_0030
- NORINTSTS2, ROC/RW1C, Address = 0xEDA0_0030

The Normal Interrupt Status Enable affects reads of this register, but Normal Interrupt Signal Enable does not affect these reads. An interrupt is generated if the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits except Card Interrupt and Error Interrupt, writing 1 to a bit clears it; writing 0 keeps the bit unchanged. More than one status is cleared with a single register write. The Card Interrupt is cleared if the card stops asserting the interrupt; that is, if the Card Driver services the interrupt condition.

NORINTSTS	Bit	Description	Reset Value
STAERR	[15]	Error Interrupt If any of the bits in the Error Interrupt Status register are set, then this bit is set. Therefore the Host Driver checks this bit first to efficiently tests for an error. This bit is read only. (ROC) '0' = No Error '1' = Error	0
STAFIA3	[14]	FIFO SD Address Pointer Interrupt 3 Status (RW1C) '0' = Occurred '1' = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 3 values, this status bit is asserted.	0
STAFIA2	[13]	FIFO SD Address Pointer Interrupt 2 Status (RW1C) '0' = Occurred '1' = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 2 values, this status bit is asserted.	0
STAFIA1	[12]	FIFO SD Address Pointer Interrupt 1 Status (RW1C) '0' = Occurred '1' = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 1 value, this status bit is asserted.	0
STAFIA0	[11]	FIFO SD Address Pointer Interrupt 0 Status (RW1C) '0' = Occurred '1' = Not Occurred If the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 0 value, this status bit is asserted.	0
STARWAIT	[10]	Read Wait Interrupt Status (RW1C) '0' = Read Wait Interrupt Not Occurred '1' = Read Wait Interrupt Occurred **Note:** After checking response for the suspend command, release	0

NORINTSTS	Bit	Description	Reset Value
		Read Wait interrupt status manually if BS = 0 (BS means 'Bus Status' field 'Bus Suspend' register in the SDIO card specification) **Note:** Read Wait operation procedure is started after 4-SDCLK from the end of the block data read transfer.	
STACCS	[9]	CCS Interrupt Status (RW1C) Command Complete Signal Interrupt Status bit is for CE-ATA interface mode. '0' = CCS Interrupt Occurred, '1' = CCS Interrupt Not Occurred	0
STACARDINT	[8]	**Card Interrupt** Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the Host Controller detects the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, therefore there are some sample delays between the interrupt signal from the SD card and the interrupt to the Host System. It is necessary to define how to handle this delay. If this status is set and the Host Driver needs to start this interrupt service, **Card Interrupt Signal Enable** in the Normal Interrupt Signal Enable register must be set to 0 in order to clear the card interrupt status latched in the Host Controller and to stop driving the interrupt signal to the Host System. After completion of the card interrupt service (It must reset interrupt factors in the SD card and the interrupt signal may not be asserted), write to one clear to this register field(RW1C) and set Card Interrupt Signal Enable to 1 to re-start sampling the interrupt signal. The Card Interrupt Status Enable must remain set to high. (**RW1C**) Note2,3 '1' = Generates Card Interrupt '0' = No Card Interrupt	0
STACARDREM	[7]	Card Removal This status is set if the Card Inserted in the Present State register changes from 1 to 0. If the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card detect state may possibly be changed if the Host Driver clear this bit and interrupt event may not be generated. (RW1C) '1' = Card removed '0' = Card state stable or Debouncing	0
STACARDINS	[6]	Card Insertion This status is set if the Card Inserted in the Present State register changes from 0 to 1. If the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card detect state may possibly be changed if the Host Driver clear this bit and interrupt event may not be generated. (RW1C) '1' = Card inserted '0' = Card state stable or Debouncing	0

NORINTSTS	Bit	Description	Reset Value
STABUFRDRDY	[5]	Buffer Read Ready This status is set if the Buffer Read Enable changes from 0 to 1. Refer to the Buffer Read Enable in the Present State register. (RW1C) '1' = Ready to read buffer '0' = Not ready to read buffer	0
STABUFWTRDY	[4]	Buffer Write Ready This status is set if the Buffer Write Enable changes from 0 to 1. Refer to the Buffer Write Enable in the Present State register. (RW1C) '1' = Ready to write buffer '0' = Not ready to write buffer	0
STADMAINT	[3]	DMA Interrupt This status is set if the Host Controller detects the Host SDMA Buffer boundary during transfer. Refer to the Host SDMA Buffer Boundary in the *Block Size* register. Other DMA interrupt factors may be added in the future. In case of ADMA, by setting interrupt field in the descriptor table, Host Controller generates this interrupt. If it is used for debugging. This interrupt is not generated after the Transfer Complete. (RW1C) '1' = Generates DMA Interrupt '0' = No DMA Interrupt	0
STABLKGAP	[2]	**Block Gap Event** If the Stop At Block Gap Request in the Block Gap Control register is set, this bit is set if both read/ write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1. (RW1C) (1) In the case of a Read Transaction This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing. The Read Wait must be supported in order to use this function). (2) Case of Write Transaction This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). '1' = Transaction stopped at block gap '0' = No Block Gap Event	0
STATRANCMPLT	[1]	**Transfer Complete** This bit is set if a read/ write transfer is complete. (1) In the case of a Read Transaction This bit is set at the falling edge of Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is if a data transfer is complete as specified by data length (After the last data has been read to the Host System). The second if data has stopped at the block gap and complete the data transfer by setting the Stop At Block Gap Request in the Block Gap Control register (After valid data has been read to the Host System). (2) In the case of a Write Transaction This bit is set at the falling edge of the DAT Line Active Status. There are two cases in which this interrupt is generated. The first if the last	0

<table>
<tr><th>NORINTSTS</th><th>Bit</th><th>Description</th><th>Reset Value</th></tr>
<tr><td></td><td></td><td>data is written to the SD card as specified by data length and the busy signal released. The second if data transfers are stopped at the block gap by setting Stop At Block Gap Request in the Block Gap Control register and data transfers complete. (After valid data is written to the SD card and the busy signal released). (RW1C)
The table below shows that Transfer Complete has higher priority than Data Timeout Error. If both bits are set to 1, the data transfer is considered complete. Relation between Transfer Complete and Data
<table>
<tr><th>Transfer Complete</th><th>Data Timeout Error</th><th>Meaning of the status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>0</td><td>1</td><td>Timeout occur during transfer</td></tr>
<tr><td>1</td><td>Don't care</td><td>Data transfer complete</td></tr>
</table>
'1' = Data Transfer Complete
'0' = No Transfer Complete</td><td></td></tr>
<tr><td>STACMDCMPLT</td><td>[0]</td><td>Command Complete
This bit is set when get the end bit of the command response. (Except Auto CMD12) Refer to Command Inhibit (CMD) in the Present State register.
The table below shows that Command Timeout Error has higher priority than Command Complete. If both bits are set to 1, it is considered that the response was not received correctly. (RW1C)
<table>
<tr><th>Command Complete</th><th>Command Timeout Error</th><th>Meaning of the status</th></tr>
<tr><td>0</td><td>0</td><td>Interrupted by another factor</td></tr>
<tr><td>Don't care</td><td>1</td><td>Response not received within 64 SDCLK cycles.</td></tr>
<tr><td>1</td><td>0</td><td>Response received</td></tr>
</table>
'1' = Command Complete
'0' = No command complete</td><td>0</td></tr>
</table>

NOTES:

1. Host Driver checks if interrupt is actually cleared by polling or monitoring the INTREQ port. If HCLK is much faster than SDCLK, it takes long time to be cleared for the bits actually.
2. Card Interrupt status bit keeps previous value until next card interrupt period (level interrupt) and is cleared if write to 1 (RW1C).
3. SD/MMC Controller of the S3C6410 does not support "card interrupt at block gap" used if the multiple block 4-bit operation.

9.18 ERROR INTERRUPT STATUS REGISTER

Error Interrupt Status Register

- ERRINTSTS0, ROC/RW1C, Address = 0xED80_0032
- ERRINTSTS1, ROC/RW1C, Address = 0xED90_0032
- ERRINTSTS2, ROC/RW1C, Address = 0xEDA0_0032

Signals defined in this register are enabled by the Error Interrupt Status Enable register, but not by the Error Interrupt Signal Enable register. Generates interrupt if the Error Interrupt Signal Enable is enabled and at least one of the statuses is set to 1. Writing to 1 clears the bit and writing to 0 keeps the bit unchanged. More than one status is cleared on one register write.

ERRINTSTS	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
STAADMAERR	[9]	ADMA Error This bit is set if the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the *ADMA Error Status* Register, In addition, the Host Controller generates this Interrupt if it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the *ADMA Error Status* indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor. '1' = Error '0' = No Error	0
STAACMDERR	[8]	Auto CMD12 Error Occurs if it detects that one of the bits in *Auto CMD12 Error Status* register has changed from 0 to 1. This bit is set to 1, if the errors in Auto CMD12 occur and if Auto CMD12 is not executed due to the previous command error. '1' = Error '0' = No Error	0
STACURERR	[7]	Current Limit Error Not implemented in this version. Always 0.	0
STADENDERR	[6]	Data End Bit Error Occurs if it detects 0 at the end bit position of read data which uses the *DAT* line or at the end bit position of the CRC Status. '1' = Error '0' = No Error	0
STADATCRCERR	[5]	Data CRC Error Occurs if it detects CRC error when transferring read data which uses the *DAT* line or if it detects the Write CRC status having a value of other than "010". '1' = Error '0' = No Error	0
STADATTOUTERR	[4]	Data Timeout Error	0

ERRINTSTS	Bit	Description	Reset Value
		Occurs if it detects one of following timeout conditions. (1) Busy timeout for R1b, R5b type (2) Busy timeout after Write CRC status (3) Write CRC Status timeout (4) Read Data timeout. '1' = Timeout '0' = No Error	
STACMDIDXERR	[3]	Command Index Error Occurs if a Command Index error occurs in the command response. '1' = Error '0' = No Error	0
STACMDEBITERR	[2]	Command End Bit Error Occurs if it detects that the end bit of a command response is 0. '1' = End bit Error generated '0' = No Error	
STACMDCRCERR	[1]	Command CRC Error Command CRC Error is generated in two cases. (1) If a response is returned and the Command Timeout Error is set to 0 (indicating no timeout), this bit is set to 1 if it detects a CRC error in the command response. (2) The Host Controller detects a *CMD* line conflict by monitoring the *CMD* line if a command is issued. If the Host Controller drives the *CMD* line to 1 level, but detects 0 levels on the *CMD* line at the next SDCLK edge, then the Host Controller aborts the command (Stop driving *CMD* line) and set this bit to 1. The Command Timeout Error also set to 1 to distinguish *CMD* line conflict. '1' = Generates CRC Error '0' = No Error	0
STACMDTOUTERR	[0]	Command Timeout Error Occurs if no response is returned within 64 *SDCLK* cycles from the end bit of the command. If the Host Controller detects a *CMD* line conflict, in which case Command CRC Error also set as shown in Table 33, this bit sets without waiting for 64 *SDCLK* cycles because the Host Controller aborts command. '1' = Timeout '0' = No Error	0

The relation between Command CRC Error and Command Timeout Error is shown in Table below.

The relation between Command CRC Error and Command Timeout Error

Command CRC Error	Command Timeout Error	Kinds of error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	***CMD*** line conflict

9.19 NORMAL INTERRUPT STATUS ENABLE REGISTER

Normal Interrupt Status Enable Register

- NORINTSTSEN0, R/W, Address = 0xED80_0034
- NORINTSTSEN1, R/W, Address = 0xED90_0034
- NORINTSTSEN2, R/W, Address = 0xEDA0_0034

Setting to 1 enables Interrupt Status.

NORINTSTSEN	Bit	Description	Reset Value
	[15]	Fixed to 0 The Host Driver controls error interrupts using the *Error Interrupt Status Enable* register. *(RO)*	0
ENSTAFIA3	[14]	FIFO SD Address Pointer Interrupt 3 Status Enable '1' = Enabled '0' = Masked	0
ENSTAFIA2	[13]	FIFO SD Address Pointer Interrupt 2 Status Enable '1' = Enabled '0' = Masked	0
ENSTAFIA1	[12]	FIFO SD Address Pointer Interrupt 1 Status Enable '1' = Enabled '0' = Masked	0
ENSTAFIA0	[11]	FIFO SD Address Pointer Interrupt 0 Status Enable '1' = Enabled '0' = Masked	0
ENSTARWAIT	[10]	Read Wait interrupt status enable '1' = Enabled '0' = Masked	0
ENSTACCS	[9]	CCS Interrupt Status Enable '1' = Enabled '0' = Masked	0
ENSTACARDINT	[8]	Card Interrupt Status Enable If this bit is set to 0, the Host Controller clears interrupt request to the System. The Card Interrupt detection is stopped if this bit is cleared and restarted if this bit is set to 1. The Host Driver must clear the Card Interrupt Status Enable before servicing the Card Interrupt and must set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. '1' = Enabled '0' = Masked	0
ENSTACARDREM	[7]	Card Removal Status Enable '1' = Enabled '0' = Masked	0

NORINTSTSEN	Bit	Description	Reset Value
ENSTACARDNS	[6]	Card Insertion Status Enable '1' = Enabled '0' = Masked	0
ENSTABUFRDRDY	[5]	Buffer Read Ready Status Enable '1' = Enabled '0' = Masked	0
ENSTABUFWTRDY	[4]	Buffer Write Ready Status Enable '1' = Enabled '0' = Masked	0
ENSTADMA	[3]	DMA Interrupt Status Enable '1' = Enabled '0' = Masked	0
ENSTABLKGAP	[2]	Block Gap Event Status Enable '1' = Enabled '0' = Masked	0
ENSTASTANSCMPLT	[1]	Transfer Complete Status Enable '1' = Enabled '0' = Masked	0
ENSTACMDCMPLT	[0]	Command Complete Status Enable '1' = Enabled '0' = Masked	0

9.20 ERROR INTERRUPT STATUS ENABLE REGISTER

Error Interrupt Status Enable Register

- ERRINTSTSEN0, R/W, 0xED80_0036
- ERRINTSTSEN1, R/W, 0xED90_0036
- ERRINTSTSEN2, R/W, 0xEDA0_0036

Setting to 1 enables Error Interrupt Status.

ERRINTSTSEN	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
ENSTAADMAERR	[9]	ADMA Error Status Enable '1' = Enabled '0' = Masked	0
ENSTAACMDERR	[8]	Auto CMD12 Error Status Enable '1' = Enabled '0' = Masked	0
ENSTACURERR	[7]	Current Limit Error Status Enable This function is not implemented in this version. '1' = Enabled '0' = Masked	0
ENSTADENDERR	[6]	Data End Bit Error Status Enable '1' = Enabled '0' = Masked	0
ENSTADATCRCERR	[5]	Data CRC Error Status Enable '1' = Enabled '0' = Masked	0
ENSTADATTOUTERR	[4]	Data Timeout Error Status Enable '1' = Enabled '0' = Masked	0
ENSTACMDIDXERR	[3]	Command Index Error Status Enable '1' = Enabled '0' = Masked	0
ENSTACMDEBITERR	[2]	Command End Bit Error Status Enable '1' = Enabled '0' = Masked	0
ENSTACMDCRCERR	[1]	Command CRC Error Status Enable '1' = Enabled '0' = Masked	0
ENSTACMDTOUTERR	[0]	Command Timeout Error Status Enable '1' = Enabled '0' = Masked	0

9.21 NORMAL INTERRUPT SIGNAL ENABLE REGISTER

Normal Interrupt Signal Enable Register

- NORINTSIGEN0, R/W, Address = 0xED80_0038
- NORINTSIGEN1, R/W, Address = 0xED90_0038
- NORINTSIGEN2, R/W, Address = 0xEDA0_0038

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits share the same1 bit interrupt line. To enable interrupt generate set any of this bit to 1.

NORINTSIGEN	Bit	Description	Reset Value
	[15]	Fixed to 0 The Host Driver controls error interrupts using the *Error Interrupt Signal Enable* register.	0
ENSIGFIA3	[14]	FIFO SD Address Pointer Interrupt 3 Signal Enable '1' = Enabled '0' = Masked	0
ENSIGFIA2	[13]	FIFO SD Address Pointer Interrupt 2 Signal Enable '1' = Enabled '0' = Masked	0
ENSIGFIA1	[12]	FIFO SD Address Pointer Interrupt 1 Signal Enable '1' = Enabled '0' = Masked	0
ENSIGFIA0	[11]	FIFO SD Address Pointer Interrupt 0 Signal Enable '1' = Enabled '0' = Masked	0
ENSIGRWAIT	[10]	Read Wait Interrupt Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCCS	[9]	CCS Interrupt Signal Enable Command Complete Signal Interrupt Status bit is for CE-ATA interface mode. '1' = Enabled '0' = Masked	0
ENSIGCARDINT	[8]	Card Interrupt Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCARDREM	[7]	Card Removal Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCARDNS	[6]	Card Insertion Signal Enable '1' = Enabled '0' = Masked	0

NORINTSIGEN	Bit	Description	Reset Value
ENSIGBUFRDRDY	[5]	Buffer Read Ready Signal Enable '1' = Enabled '0' = Masked	0
ENSIGBUFWTRDY	[4]	Buffer Write Ready Signal Enable '1' = Enabled '0' = Masked	0
ENSIGDMA	[3]	DMA Interrupt Signal Enable '1' = Enabled '0' = Masked	0
ENSIGBLKGAP	[2]	Block Gap Event Signal Enable '1' = Enabled '0' = Masked	0
ENSIGSTANSCMPLT	[1]	Transfer Complete Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCMDCMPLT	[0]	Command Complete Signal Enable '1' = Enabled '0' = Masked	0

9.22 ERROR INTERRUPT SIGNAL ENABLE REGISTER

Error Interrupt Signal Enable Register

- ERRINTSIGEN0, R/W, Address = 0xED80_003A
- ERRINTSIGEN1, R/W, Address = 0xED90_003A
- ERRINTSIGEN2, R/W, Address = 0xEDA0_003A

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits share the same 1 bit interrupt line. To enable interrupt generate set any of this bit to 1.

ERRINTSIGEN	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
ENSIGADMAERR	[9]	ADMA Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGACMDERR	[8]	Auto CMD12 Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCURERR	[7]	Current Limit Error Signal Enable This function is not implemented in this version. '1' = Enabled '0' = Masked	0
ENSIGDENDERR	[6]	Data End Bit Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGDATCRCERR	[5]	Data CRC Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGDATTOUTERR	[4]	Data Timeout Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCMDIDXERR	[3]	Command Index Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCMDEBITERR	[2]	Command End Bit Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCMDCRCERR	[1]	Command CRC Error Signal Enable '1' = Enabled '0' = Masked	0
ENSIGCMDTOUTERR	[0]	Command Timeout Error Signal Enable '1' = Enabled '0' = Masked	0

NOTE: Detailed documents must be copied from SD Host Standard Specification.

9.23 AUTOCMD12 ERROR STATUS REGISTER

Auto CMD12 Error Status Register

- ACMD12ERRSTS0, ROC, Address = 0xED80_003C
- ACMD12ERRSTS1, ROC, Address = 0xED90_003C
- ACMD12ERRSTS2, ROC, Address = 0xEDA0_003C

If Auto CMD12 Error Status is set, the Host Driver checks this register to identify what kind of error Auto CMD12 indicated. This register is valid if the Auto CMD12 Error is set.

ACMD12ERRSTS	Bit	Description	Reset Value
Reserved	[15:8]	Reserved	0
STANCMDAER	[7]	Command Not Issued By Auto CMD12 Error If thi bit is set to 1, it means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. '1' = Not Issued '0' = No error	0
Reserved	[6:5]	Reserved	0
STACMDIDXERR	[4]	Auto CMD12 Index Error Occurs if the Command Index error occurs in response to a command. '1' = Error '0' = No Error	0
STACMDEBITAER	[3]	Auto CMD12 End Bit Error Occurs it detects that the end bit of command response is 0. '1' = End Bit Error Generated '0' = No Error	0
STACMDCRCAER	[2]	Auto CMD12 CRC Error Occurs if it detects a CRC error in the command response. '1' = CRC Error Generated '0' = No Error	0
STACMDTOUTAER	[1]	Auto CMD12 Timeout Error Occurs if no response is returned within 64 *SDCLK* cycles from the end bit of command. If this bit is set to1, the other error status bits (D04-D02) are meaningless. '1' = Time out '0' = No Error	0
STANACMDAER	[0]	Auto CMD12 Not Executed If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. If this bit is set to 1, it means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless. '1' = Not executed '0' = Executed	0

The relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error is shown below.

The relation between Command CRC Error and Command Timeout Error

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Kinds of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	***CMD*** line conflict

The timing of changing *Auto CMD12 Error Status* is classified in three scenarios:

(1) If the Host Controller is going to issue Auto CMD12

Set D00 to 1 if Auto CMD12 cannot be issued due to an error in the previous command.

Set D00 to 0 if Auto CMD12 is issued.

(2) At the end bit of an Auto CMD12 response

Check received responses by checking the error bits D01, D02, D03 and D04.

Set to 1 if error is detected.

Set to 0 if error is not detected.

(3) Before reading the Auto CMD12 Error Status bit D07

Set D07 to 1 if there is a command cannot be issued

Set D07 to 0 if there is no command to issue

Timing to generate the Auto CMD12 Error and writing to the Command register are asynchronous. Then D07 are sampled if driver never writes to the Command register. Therefore before reading the Auto CMD12 Error Status register, set the D07 status bit. Generates Auto CMD12 Error Interrupt if one of the error bits D00 to D04 is set to 1. The Command Not Issued by Auto CMD12 Error does not generate an interrupt.

9.24 CAPABILITIES REGISTER

Capabilities Register

- CAPAREG0, HWInit, Address = 0xED80_0040
- CAPAREG1, HWInit, Address = 0xED90_0040
- CAPAREG2, HWInit, Address = 0xEDA0_0040

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller implements these values as fixed or loaded from flash memory during power on initialization. Refer to Software Reset for the Software Reset register for loading from flash memory and completion timing control.

CAPAREG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	
CAPAV18	[26]	Voltage Support 1.8V (HWInit) '1' = 1.8V Supported '0' = 1.8V Not Supported	1
CAPAV30	[25]	Voltage Support 3.0V (HWInit) '1' = 3.0V Supported '0' = 3.0V Not Supported	0
CAPAV33	[24]	Voltage Support 3.3V (HWInit) '1' = 3.3V Supported '0' = 3.3V Not Supported	1
CAPASUSRES	[23]	Suspend/Resume Support (HWInit) This bit indicates whether the Host Controller supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism are not supported and the Host Driver does not issue either Suspend or Resume commands. '1' = Supported '0' = Not Supported	1
CAPADMA	[22]	DMA Support (HWInit) This bit indicates whether the Host Controller is capable of using DMA to transfer data between system memory and the Host Controller directly. '1' = DMA Supported '0' = DMA Not Supported	1
CAPAHSPD	[21]	High Speed Support (HWInit) This bit indicates whether the Host Controller and the Host System support High Speed mode and they can supply SD Clock frequency from 25MHz to 50MHz. '1' = High Speed Supported '0' = High Speed Not Supported	1
Reserved	[20]	Reserved	0
CAPAADMA2	[19]	ADMA2 Support This bit indicates whether the Host Controller is capable of using ADMA2.	1

CAPAREG	Bit	Description	Reset Value
		'1' = ADMA2 Support '0' = ADMA2 not Support	
Reserved	[28]	Reserved	0
CAPAMAXBLKLEN	[17:16]	Max Block Length (HWInit) This value indicates the maximum block size that the Host Driver can read and write to the buffer in the Host Controller. The buffer transfers this block size without wait cycles. Three sizes are defined as indicated below. '00' = 512-byte, '01' = 1024-byte, '10' = 2048-byte, '11' = Reserved	0
Reserved	[15:14]	Reserved	0
CAPABASECLK	[13:8]	Base Clock Frequency For SD Clock (HWInit) This value indicates the base (maximum) clock frequency for the SD Clock. Unit values are 1MHz. If the real frequency is 16.5MHz, the lager value is set to 01 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to the SDCLK Frequency Select in the *Clock Control* register.) and it does not exceed upper limit of the SD Clock frequency. The supported clock range is 10MHz to 63MHz. If these bits are all 0, the Host System has to get information via another method. Not '0' = 1MHz to 63MHz 000000b = Get information via another method	0
CAPATOUTUNIT	[7]	Timeout Clock Unit (HWInit) This bit shows the unit of base clock frequency used to detect Data Timeout Error. '0' = kHz, '1' = MHz	1
Reserved	[6]	Reserved	0
CAPATOUTCLK	[5:0]	Timeout Clock Frequency (HWInit) This bit shows the base clock frequency used to detect Data Timeout Error. The Timeout Clock Unit defines the unit of this field value. Timeout Clock Unit =0 [kHz] unit: 1kHz to 63kHz Timeout Clock Unit =1 [MHz] unit: 1MHz to 63MHz Not 0 = 1kHz to 63kHz or 1MHz to 63MHz 00 0000b = Get information via another method	0

9.25 MAXIMUM CURRENT CAPABILITIES REGISTER

Maximum Current Capabilities Register

- MAXCURR0, HWInit, Address = 0xED80_0048
- MAXCURR1, HWInit, Address = 0xED90_0048
- MAXCURR2, HWInit, Address = 0xEDA0_0048

These registers indicate maximum current capability for each voltage. The value is meaningful if Voltage Support is set in the Capabilities register. If this information is supplied by the Host System via another method, all Maximum Current Capabilities register will be 0.

MAXCURR	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	
MAXCURR18	[23:16]	Maximum Current for 1.8V (HWInit)	0
MAXCURR30	[15:8]	Maximum Current for 3.0V (HWInit)	0
MAXCURR33	[7:0]	Maximum Current for 3.3V (HWInit)	0

This register measures current in 4mA steps. Each voltage level's current support is described using the Table below.

Maximum Current Value Definition

Register Value	Current Value
0	Get information via another method
1	4mA
2	8mA
3	12mA
…	…
255	1020mA

9.26 FORCE EVENT REGISTER FOR AUTO CMD12 ERROR STATUS

Force Event Auto CMD12 Error Interrupt Register

- FEAER0, W, Address = 0xED80_0050
- FEAER1, W, Address = 0xED90_0050
- FEAER2, W, Address = 0xEDA0_0050

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Auto CMD12 Error Status Register can be written.

Writing 1: set each bit of the Auto CMD12 Error Status Register

Writing 0: no effect

D15 D12

FEAER	Bit	Description	Reset Value
Reserved	[15:8]	- Reserved	0x0
FENCMDAER	[7]	Force Event for Command Not Issued By Auto CMD12 Error 1 = Generates Interrupt 0 = No Interrupt	0
Reserved	[6:5]	- Reserved	0
FECMDIDXERR	[4]	Force Event for Auto CMD12 Index Error 1 = Interrupt 0 = No Interrupt	0
FECMDEBITAER	[3]	Force Event for Auto CMD12 End Bit Error 1 = Generates Interrupt 0 = No Interrupt	0
FECMDCRCAER	[2]	Force Event for Auto CMD12 CRC Error 1 = Generates Interrupt 0 = No Interrupt	0
FECMDTOUTAER	[1]	Force Event for Auto CMD12 Timeout Error 1 = Generates Interrupt 0 = No Interrupt	0
FENACMDAER	[0]	Force Event for Auto CMD12 Not Executed 1 = Generates Interrupt 0 = No Interrupt	0

9.27 FORCE EVENT REGISTER FOR ERROR INTERRUPT STATUS

Force Event Error Interrupt Register Error Interrupt

- FEERR0, W, Address = 0xED80_0052
- FEERR1, W, Address = 0xED90_0052
- FEERR2, W, Address = 0xEDA0_0052

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Error Interrupt Status register is written. The effect of a write to this address is reflected in the Error Interrupt Status Register if the corresponding bit of the Error Interrupt Status Enable Register is set.

Writing 1: set each bit of the Error Interrupt Status Register

Writing 0: no effect

NOTE: By setting this register, the Error Interrupt can be set in the Error Interrupt Status register. In order to generate interrupt signal, both the Error Interrupt Status Enable and Error Interrupt Signal Enable shall be set.

FEERR	Bit	Description	Reset Value
Reserved	[15:12]	Reserved	0x0
Reserved	[11:10]	Reserved	0x0
FEADMAERR	[9]	Force Event for ADMA Error 1 = Generates Interrupt 0 = No Interrupt	0
FEACMDERR	[8]	Force Event for Auto CMD12 Error 1 = Generates Interrupt 0 = No Interrupt	0
Reserved	[7]	Reserved	0
FEDENDERR	[6]	Reserved	0
FEDATCRCERR	[5]	Force Event for Data CRC Error 1 = Generates Interrupt 0 = No Interrupt	0
FEDATTOUTERR	[4]	Force Event for Data Timeout Error 1 = Generates Interrupt 0 = No Interrupt	0
FECMDIDXERR	[3]	Force Event for Command Index Error 1 = Generates Interrupt 0 = No Interrupt	0
FECMDEBITERR	[2]	Force Event for Command End Bit Error 1 = Generates Interrupt 0 = No Interrupt	0
FECMDCRCERR	[1]	Force Event for Command CRC Error 1 = Generates Interrupt 0 = No Interrupt	0
FECMDTOUTERR	[0]	Force Event for Command Timeout Error 1 = Generates Interrupt 0 = No Interrupt	0

9.28 ADMA ERROR STATUS REGISTER

ADMA Error Status Register

- ADMAERR0, R/W, Address = 0xED80_0054
- ADMAERR1, R/W, Address = 0xED90_0054
- ADMAERR2, R/W, Address = 0xEDA0_0054

If ADMA Error Interrupt occurs, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address

ST_FDS: Current location set in the ADMA System Address register is the error descriptor address

ST_CADR: This sate is never set because do not generate ADMA error in this state.

ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address

In case of write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller.
The Host Controller generates the ADMA Error Interrupt if it detects invalid descriptor data (Valid = 0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver finds that the Valid bit is not set in the error descriptor.

ADMAERR	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x00
STAADMAFINBLK	[10]	ADMA Final Block Transferred (ROC) In ADMA operation mode, this field is set to High if the Transfer Complete condition and the block are final (no block transfer remains). If this bit is Low when the Transfer Complete condition and Transfer Complete is done due to the Stop at Block Gap, so data to be transferred still remains.	0
ADMACONTREQ	[9]	ADMA Continue Request (WO) If the stop state by ADMA Interrupt, ADMA operation set this bit to HIGHT to continue.	0
ADMASTAINT	[8]	ADMA Interrupt Status (RW1C) This bit is set to HIGH if INT attribute in the ADMA Descriptor Table is asserted. This bit is not affected by ADMA error interrupt.	0
	[7:3]	Reserved	0
ADMALENMISERR	[2]	ADMA Length Mismatch Error This error occurs in the following 2 cases. (1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. (2) Total data length can not be divided by the block length. '0' = No Error	00

ADMAERR	Bit	Description	Reset Value
		'1' = Error	
ADMAERRST	[1:0]	ADMA Error State This field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state. D01 – D00 ADMA Error State when error is occurred Contents of SYS_SDR register '00' = ST_STOP (Stop DMA) Points next of the error descriptor '01' = ST_FDS (Fetch Descriptor) Points the error descriptor '10' = Never set this state (Not used) '11' = ST_TFR (Transfer Data) Points the next of the error descriptor	0

9.29 ADMA SYSTEM ADDRESS REGISTER

ADMA System Address Register

- ADMASYSADDR0, R/W, Address = 0xED80_0058
- ADMASYSADDR1, R/W, Address = 0xED90_0058
- ADMASYSADDR2, R/W, Address = 0xEDA0_0058

This register contains the physical Descriptor address used for ADMA data transfer.

ADMASYSADDR	Bit	Description	Reset Value
ADMASYSAD	[31:0]	ADMA System Address This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver sets start address of the Descriptor table. The ADMA increments this register address, which points to next line, if every fetching a Descriptor line. If the ADMA Error Interrupt is generated, this register holds valid Descriptor address depending on the ADMA state. The Host Driver programs Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b. 32-bit Address ADMA Register Value 32-bit System Address xxxxxxxx 00000000h 00000000h xxxxxxxx 00000004h 00000004h xxxxxxxx 00000008h 00000008h xxxxxxxx 0000000Ch 0000000Ch …… …… xxxxxxxx FFFFFFFCh FFFFFFFCh **Note:** The data length of the ADMA Descriptor Table should be the word unit (multiple of the 4-byte).	00

9.30 CONTROL REGISTER 2

Control register 2

- CONTROL2_0, R/W, Address = 0xED80_0080
- CONTROL2_1, R/W, Address = 0xED90_0080
- CONTROL2_2, R/W, Address = 0xEDA0_0080

This register contains the SD Command Argument.

CONTROL2	Bit	Description	Reset Value
ENSTAASYNCCLR	[31]	Write Status Clear Async Mode Enable This bit makes async-clear enable about Normal and Error interrupt status bit. During the initialization procedure command operation, this bit should be enabled. '0' = Disable '1' = Enable	0
ENCMDCNFMSK	[30]	Command Conflict Mask Enable This bit can mask enable the Command Conflict Status (bit [1:0] of the "ERROR INTERRUPT STATUS REGISTER") 0 = Mask Disable, 1 = Mask Enable **Note:** If the ENHIGHSPD field in the Host Control Register is set (High Speed data transfer), this field should be enabled to prevent from command conflict status alarm.	0
Reserved	[29]	Reserved (must be 1'b0)	0
SELCARDOUT	[28]	Card Removed Condition Selection 0= Card Removed condition is "Not Card Insert" State (When the transition from "Card Inserted" state to "Debouncing" state in Figure 8.12-17) 1 = Card Removed state is "Card Out" State (If the transition from "Debouncing" state to "No Card" state in Figure 8.12-17)	0
FLTCLKSEL	[27:24]	Filter Clock (iFLTCLK) Selection Filter Clock period = 2^(FltClkSel + 5) x iSDCLK period 0000 = 25 x iSDCLK, 0001 = 26 x iSDCLK … 1111 = 220 x iSDCLK	0
LVLDAT	[23:16]	DAT line level Bit[23]=DAT[7], BIT[22]=DAT[6], BIT[21]=DAT[5], BIT[20]=DAT[4], Bit[19]=DAT[3], BIT[18]=DAT[2], BIT[17]=DAT[1], BIT[16]=DAT[0] (Read Only)	Line state
ENFBCLKTX	[15]	Feedback Clock Enable for Tx Data/Command Clock '0' = Disable, '1' = Enable	0
ENFBCLKRX	[14]	Feedback Clock Enable for Rx Data/Command Clock '0' = Disable, '1' = Enable	0
Reserved	[13]	Reserved (must be 1'b0)	0
SDOPSIGPC	[12]	SD Output Signal Power Control Support If set this field is enables output CMD and DAT referencing SD Bus Power bit in the "PWRCON register". '0' = CMD and DAT outputs are not controlled by SD Bus Power bit '1' = CMD and DAT outputs are controlled(masked) by SD Bus Power bit	0

CONTROL2	Bit	Description	Reset Value
ENBUSYCH KTXSTART	[11]	CE-ATA I/F mode Busy state check before Tx Data start state 0 = Disable, 1 = Enable	0
DFCNT	[10:9]	Debounce Filter Count Debounce Filter Count setting register for Card Detect signal input (SDCD#) 00 = No use debounce filter, 01 = 4 iSDCLK, 10 = 16 iSDCLK, 11 = 64 iSDCLK	0
ENCLKOUTH OLD	[8]	SDCLK Hold Enable Enter and exit of the SDCLK Hold state is done by Host Controller. 0 = Disable, 1 = Enable	0
RWAITMOD E	[7]	Read Wait Release Control 0 = Read Wait state is released by the Host Controller (Auto) 1 = Read Wait state is released by the Host Driver (Manual)	0
DISBUFRD	[6]	Buffer Read Disable 0 = Normal mode, user can read buffer(FIFO) data using 0x20 register 1 = User cannot read buffer(FIFO) data using 0x20 register. In this case, the buffer memory is read through memory area (Debug purpose).	0
SELBASE CLK	[5:4]	Base Clock Source Select 00 or 01 = HCLK, 10 = SCLK_MMC0, SCLK_MMC1, SCLK_MMC2 (from CLKCON), 11 = SCLK_MMC0_48, SCLK_MMC1_48, SCLK_MMC2_48, (XTI or XEXTCLK)	00
SDINPSIGPC	[3]	SD Input Signal Power Control Support If set this field enables input CMD and DAT referencing SD Bus Power bit in the "PWRCON register". '0' = No Sync, no switch input enable signal (Command, Data) '1' = Sync, control input enable signal (Command, Data)	0
- Reserved	[2]	Reserved	0
ENCLKOUT MSKCON	[1]	SDCLK output clock masking when Card Insert cleared If this field is High, it is used not to stop SDCLK if No Card state. '0' = Disable, '1' = Enable	0
HWINITFIN	[0]	SD Host Controller Hardware Initialization Finish 0 = Not Finish, 1 = Finish	0

NOTES:

1. Ensure to set SDCLK Hold Enable (EnSCHold) if the card does not support Read Wait to guarantee for Receive data not overwritten to the internal FIFO memory.
2. CMD_wo_DAT issue is prohibited during READ transfer if SDCLK Hold Enable is set.

9.31 CONTROL REGISTERS 3 REGISTER

FIFO Interrupt Control (Control Register 3)

- CONTROL3_0, R/W, Address = 0xED80_0084
- CONTROL3_1, R/W, Address = 0xED90_0084
- CONTROL3_2, R/W, Address = 0xEDA0_0084

CONTROL3	Bit	Description	Reset Value
FCSEL3	[31]	Feedback Clock Select [3] Reference Note (1)	0x0
FIA3	[30:24]	FIFO Interrupt Address register 3 FIFO (512Byte Buffer memory, word address unit) Initial value (0x7F) generates at 512-byte(128-word) position.	0x7F
FCSEL2	[23]	Feedback Clock Select [2] Reference Note (1)	0x0
FIA2	[22:16]	FIFO Interrupt Address register 2 FIFO (512Byte Buffer memory, word address unit) Initial value (0x5F) generates at 384-byte(96-word) position.	0x5F
FCSEL1	[15]	Feedback Clock Select [1] Reference Note (2)	0x0
FIA1	[14:8]	FIFO Interrupt Address register 1 FIFO (512Byte Buffer memory, word address unit) Initial value (0x3F) generates at 256-byte(64-word) position.	0x3F
FCSEL0	[7]	Feedback Clock Select [0] Reference Note (2)	0x0
FIA0	[6:0]	FIFO Interrupt Address register 0 FIFO (512Byte Buffer memory, word address unit) Initial value (0x1F) generates at 128-byte(32-word) position.	0x1F

NOTES:

1. FCSel[3:2] : Tx Feedback Clock Delay Control : Inverter delay means 10ns delay if SDCLK 50MHz setting
   '01' = Delay1 (basic delay), '11' = Delay2 (basic delay + 2ns),
   '00' = Delay3 (inverter delay), '10' = Delay4 (inverter delay + 2ns)
2. FCSel[1:0] : Rx Feedback Clock Delay Control : Inverter delay means10ns delay if SDCLK 50MHz setting
   '01' = Delay1 (basic delay), '11' = Delay2 (basic delay + 2ns),
   '00' = Delay3 (inverter delay), '10' = Delay4 (inverter delay + 2ns)
3. Tx Feedback inversion setting (FCSel[3:2] = '00' or '10'), Tx Feedback clock enable (ENFBCLKTX = 0) and Normal Speed mode (ENHIGHSPD = 0) setting make Tx data transfer mismatch (Do not set).

9.32 CONTROL REGISTER 4

Control register 4

- CONTROL4_0, R/W, Address = 0xED80_008C
- CONTROL4_1, R/W, Address = 0xED90_008C
- CONTROL4_2, R/W, Address = 0xEDA0_008C

CONTROL4	Bit	Description	Reset Value
Reserved -	[31:18]	Reserved	0
SELCLKPADDS	[17:16]	SD Clock Output PAD Drive Strength Select '00' = 2mA, '01' = 4mA, '10' = 7mA, '11' = 9mA **Note:** This setting is for HSMMC Controller Channel 0 and 1. For HSMMC Channel 2, "SELCLKPADDS" is located in SPCON(Special Port Control) register bit[19:18](DRVCON_SPICLK[1]) in GPIO module.	0
Reserved	[15:1]	Reserved	
STABUSY	[0]	Status Busy This bit is "High" if the clock domain crossing (HCLK to SDCLK) operation is under process. This bit is status bit and Read Only (RO)	0

9.33 HOST CONTROLLER VERSION REGISTER

Host Controller Version Register

- HCVER0, HWInit, Address = 0xED80_00FE
- HCVER1, HWInit, Address = 0xED90_00FE
- HCVER2, HWInit, Address = 0xEDA0_00FE

This register contains the SD Command Argument.

HCVER	Bit	Description	Reset Value
VENVER	[15:8]	Vendor Version Number This status is reserved for the vendor version number. The Host Driver should not use this status. 0x04 : SDMMC4.0 Host Controller	0x04
SPECVER	[7:0]	Specification Version Number This status indicates the Host Controller Specification. Version. The upper and lower 4-bits indicate the version '00' = SD Host Specification Version 1.0 '01' = SD Host Specification Version 2.00 Including the feature of the ADMA and Test Register Others = Reserved	0x01

9.1 DISPLAY CONTROLLER

1 OVERVIEW

The Overlay/ Display controller consists of logic for transferring image data from a local bus of the POST Processor or a video buffer located in system memory to an external LCD driver interface. LCD driver interface has four kinds of interface, i.e. the conventional RGB-interface, indirect-i80 Interface and ITU-R BT.601/656 interface. The display controller supports up to 5 overlay image windows, which support various color format, 256 level alpha blending, color key, x-y position control, soft scrolling, variable window size, and etc.

The display controller supports various color formats as RGB (1bpp to 24 bpp), and YCbCr 4:4:4 (only local bus).

The display controller are programmed to support the different requirements on the screen related to the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

The display controller transfers the video data and generates the necessary control signals such as, RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, VEN_VSYNC, VEN_HSYNC, VEN_FIELD, VEN_HREF, V601CLK, V656CLK and SYS_CS0, SYS_CS1, SYS_WE.... As well as the control signals, display controller has the data ports for video data, which are RGB_VD[23:0], SYS_VD, VEN_VD and V656_VD as shown in Figure 9.1-1

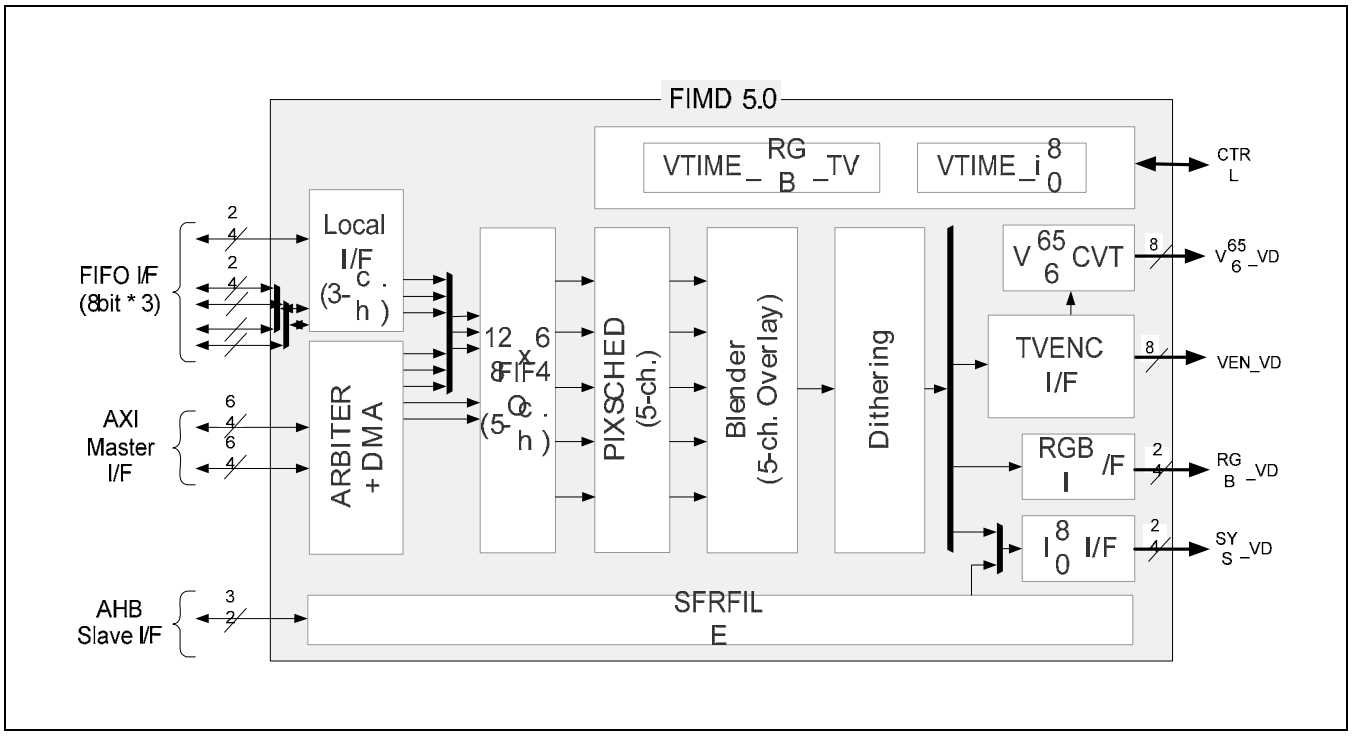


Figure 9.1-1 Top Block Diagram Display Controller

2 FEATURES

The features of the FIMD include:

Bus Interface	AMBA AXI 64bit Master / AHB 32-bit Slave LOCAL VIDEO BUS (YCbCr / RGB)
Video Output Interface	RGB IF (Parallel/Serial) Indirect i80 Interface ITU-R BT.601/656 interface (YUV422 8-bit only)
PIP (OSD) function	Supports 8-bpp (bit per pixel) palettized color Supports 16-bpp non-palettized color Supports unpacked 18-bpp non-palettized color Supports unpacked 24-bpp non-palettized color
	Supports X,Y indexed position
	Supports 8-bit Alpha blending : Plane / Pixel
CSC (Internal)	YCbCr(4:4:4) to RGB (only Local VIDEO BUS input) RGB to YCbCr(4:2:2) (ITU601/656 output)
Source format	**Window 0** Supports 1, 2, 4 or 8-bpp (bit per pixel) palettized color Supports 16, 18 or 24-bpp non-palettized color Supports YCbCr (4:4:4) local input from Local Bus (FIMC0) Supports RGB (8:8:8) local input from Local Bus (FIMC0) **Window 1** Supports 1, 2, 4 or 8-bpp (bit per pixel) palettized color Supports 16, 18 or 24-bpp non-palettized color Supports YCbCr (4:4:4) local input from Local Bus (FIMC1) Supports RGB (8:8:8) local input from Local Bus (FIMC1) Supports YCbCr (4:4:4) local input from Local Bus (Video Processor) **Window 2** Supports 1, 2, 4 or 8-bpp (bit per pixel) palettized color Supports 16, 18 or 24-bpp non-palettized color Supports YCbCr (4:4:4) local input from Local Bus (FIMC2) Supports RGB (8:8:8) local input from Local Bus (FIMC2) **Window 3 / 4** Supports 1, 2, 4 or 8-bpp (bit per pixel) palettized color Supports 16, 18 or 24-bpp non-palettized color
Configurable Burst Length	Programable 4 / 8 / 16 Burst DMA
Palette	**Window 0 / 1 / 2 / 3 / 4** 256 x 32 bits palette memory (5ea: One palette memory for each window)
Soft Scrolling	Horizontal = 1 Byte resloution Vertical = 1 pixel resolution
Virtual Screen	Virtual image can have up to 16 MB image size. Each window can have its own virtual area.

Transparent Overlay	Supports Transparent Overlay
Color Key (Chroma Key)	Supports Color key function
Partial Display	Supports LCD partial display fuction through i80 interface

3 FUNCTIONAL DESCRIPTION

3.1 BRIEF OF THE SUB-BLOCK

The display controller consists of a VSFR, VDMA, VPRCS, VTIME, and video clock generator. The VSFR has 107 programmable register sets and five 256x32-palette memories. This is used to configure the display controller. The VDMA is a dedicated display DMA, which transfers the video data in frame memory to VPRCS. By using this special DMA, displays the video data on the screen without CPU intervention. The VPRCS receives the video data from VDMA and sends the video data through the data ports (RGB_VD, VEN_VD, V656_VD or SYS_VD) to the display device (LCD) after changing them into a suitable data format, for example 8-bit per pixel mode (8 BPP Mode) or 16-bit per pixel mode (16 BPP Mode). The VTIME consists of programmable logic to support the variable requirement of interface timing and rates commonly found in different LCD drivers. The VTIME block generates RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, VEN_VSYNC, VEN_HSYNC, VEN_FIELD, VEN_HREF, V601CLK, V656CLK and SYS_CS0, SYS_CS1, SYS_WE and so on.

3.2 DATA FLOW

FIFO is present in the VDMA. If FIFO is empty or partially empty, VDMA requests data fetching from the frame memory based on the burst memory transfer mode. The data transfer rate determines size of FIFO. The display controller has five FIFOs because it needs to support the overlay window display mode. Use one FIFO for one screen display mode. VPRCS fetches data through FIFO, which has a blending, scheduling function for the final image data. VPRCS supports overlay function. This enables to overlay any image up to 5 window images whose is smaller or same size can be blended with main window image with programmable alpha blending or color (chroma) key function. Figure 9.1-2 shows the data flow from system bus to the output buffer. VDMA has 5 DMA channels and 3 Local Input I/F. Color Space Conversion (CSC) block changes YCbCr (Local input only) data to RGB data for the blending operation. Alpha values written in SFR determine the level of blending. Data from Output buffer appears to the Video Data Port.

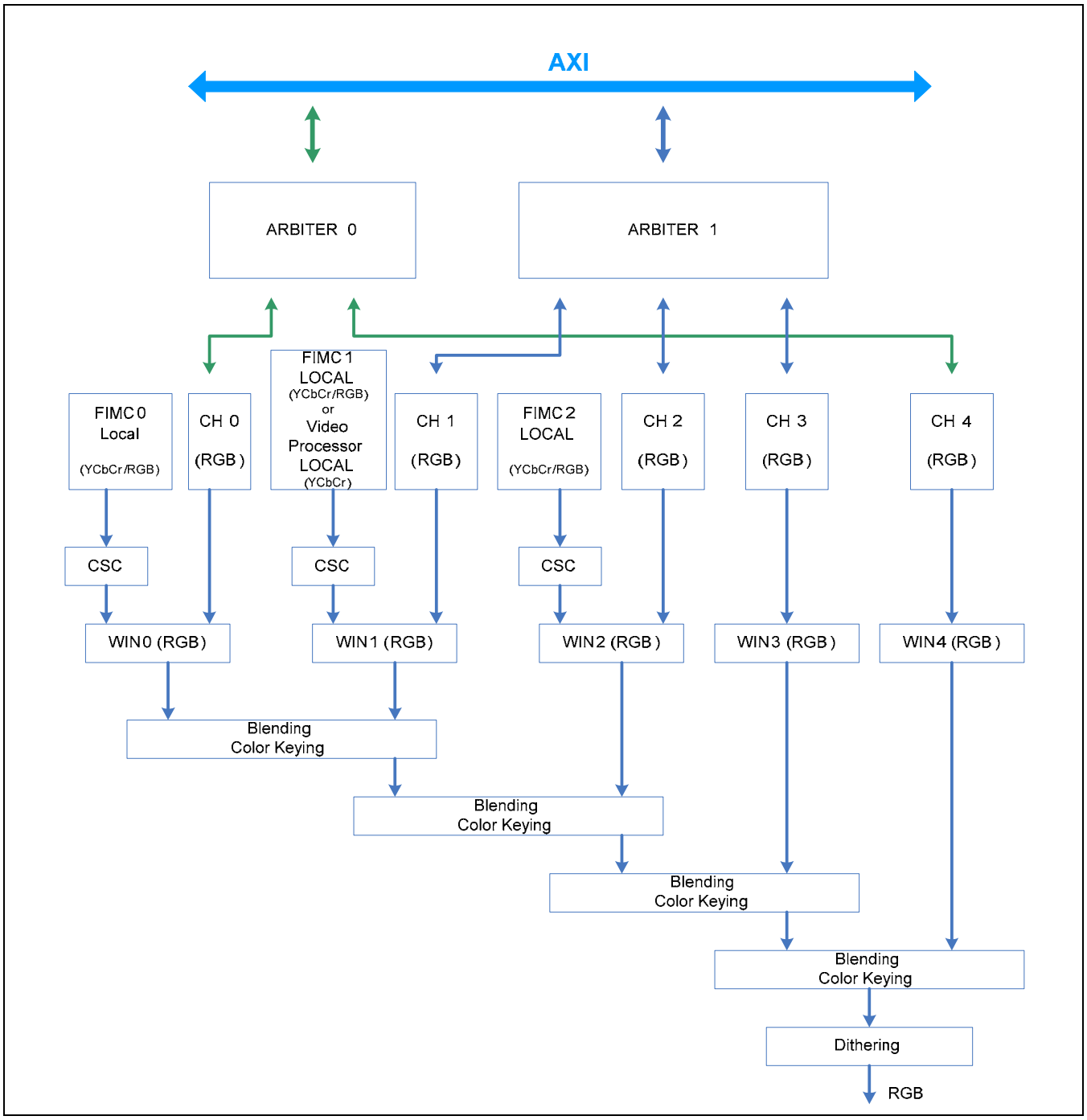


Figure 9.1-2 Block Diagram of the Data Flow

3.2.1 Interface

Display controller supports 4 types of display device. One type is the conventional RGB-interface, which uses RGB data, Vertical/horizontal sync, data valid signal and data sync clock. The Second type is Indirect i80 Interface which uses address, data, chip select, read/write control and register/status indicating signal. This type of LCD driver contains a frame buffer and has the function of self-refresh, so display controller updates one still image by writing only one time to the LCD. The Third and fourth types are ITU interfaces including both ITU-R BT.601 interface which uses YUV data, Vertical/horizontal sync, optional Field signal, data valid signal and data sync clock and ITU-R BT.656 interface which uses data sync clock and valid data which consist of YUV data, timing reference code and blanking data

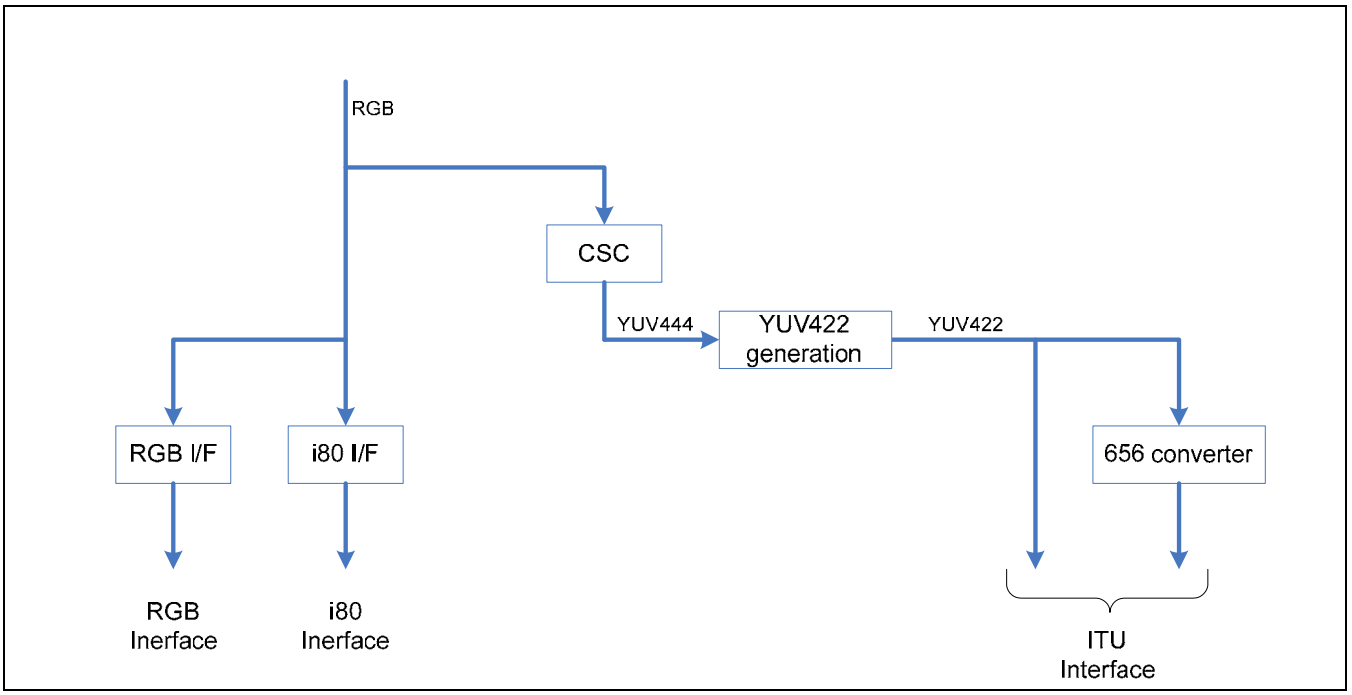


Figure 9.1-3 Block Diagram of the Interface

3.3 OVERVIEW OF THE COLOR DATA

3.3.1 RGB Data Format

The display controller requests the specified memory format of frame buffer. The table below shows some examples of each display mode.

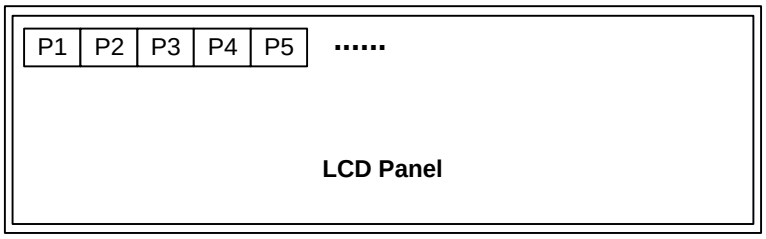
25BPP display (A888)

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:57]	D[56]	D[55:32]	D[31:25]	D[24]	D[23:0]
000H	Dummy Bit	AEN	P1	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3	Dummy Bit	AEN	P4
010H	Dummy Bit	AEN	P5	Dummy Bit	AEN	P6
…						

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:57]	D[56]	D[55:32]	D[31:25]	D[24]	D[23:0]
000H	Dummy Bit	AEN	P2	Dummy Bit	AEN	P1
008H	Dummy Bit	AEN	P4	Dummy Bit	AEN	P3
010H	Dummy Bit	AEN	P6	Dummy Bit	AEN	P5
…						



NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends with alpha value selected by AEN.

Alpha value is selected by SFR value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, ALPHA1_B. For more information, refer to "Section SFR".

NOTE 2: D [23:16] = Red data, D [15:8] = Green data, D [7:0] = Blue data
NOTE 3: 32BPP (8888) mode. (For more information, refer to "Section SFR) Data has Alpha value.

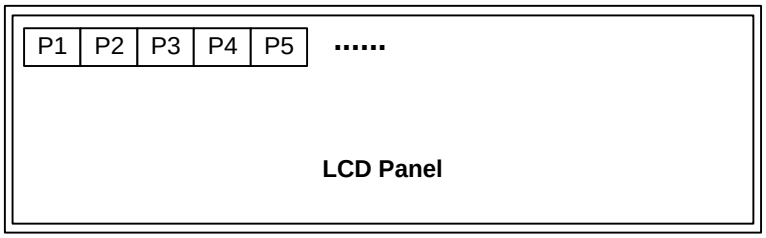
24BPP display (A887)

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:56]	D[55]	D[54:32]	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P1	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3	Dummy Bit	AEN	P4
010H	Dummy Bit	AEN	P5	Dummy Bit	AEN	P6
…						

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:56]	D[55]	D[54:32]	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P2	Dummy Bit	AEN	P1
008H	Dummy Bit	AEN	P4	Dummy Bit	AEN	P3
010H	Dummy Bit	AEN	P6	Dummy Bit	AEN	P5
…						



NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends with alpha value selected by AEN.

Alpha value is selected by sfr value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, ALPHA1_B. For more information, refer to "Section SFR.

NOTE 2: D [22:15] = Red data, D [14:7] = Green data, D [6:0] = Blue data

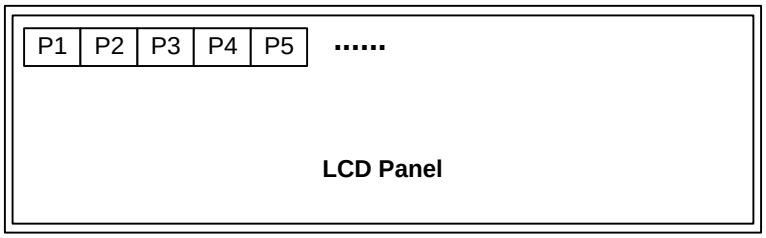
24BPP display (888)

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:56]	D[55:32]	D[31:24]	D[23:0]
000H	Dummy Bit	P1	Dummy Bit	P2
008H	Dummy Bit	P3	Dummy Bit	P4
010H	Dummy Bit	P5	Dummy Bit	P6
…				

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:56]	D[55:32]	D[31:24]	D[23:0]
000H	Dummy Bit	P2	Dummy Bit	P1
008H	Dummy Bit	P4	Dummy Bit	P3
010H	Dummy Bit	P6	Dummy Bit	P5
…				



NOTE: D [23:16] = Red data, D [15:8] = Green data, D [7:0] = Blue data

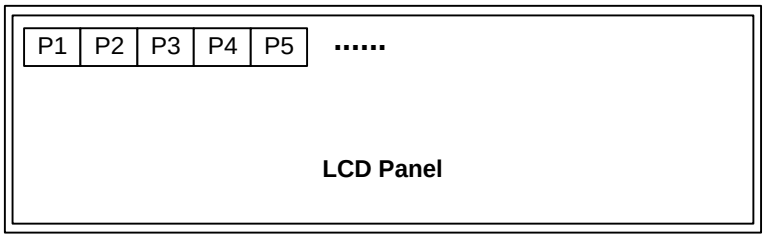
19BPP display (A666)

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:51]	D[50]	D[49:32]	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P1	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3	Dummy Bit	AEN	P4
010H	Dummy Bit	AEN	P5	Dummy Bit	AEN	P6
…						

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:51]	D[50]	D[49:32]	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P2	Dummy Bit	AEN	P1
008H	Dummy Bit	AEN	P4	Dummy Bit	AEN	P3
010H	Dummy Bit	AEN	P6	Dummy Bit	AEN	P5
…						



NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel belnds with alpha value selected by AEN. SFR selects Alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G; ALPHA1_B. For more information, refer to "Section SFR.

NOTE 2: D [17:12] = Red data, D [11:6] = Green data, D [5:0] = Blue data

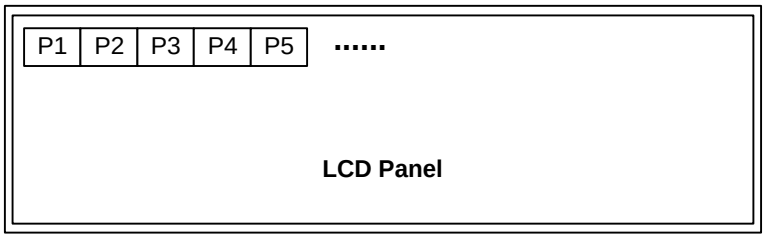
18BPP display (666)

(BSWP=0, HWSWP=0, WSWP=0)

	D[63:50]	D[49:32]	D[31:18]	D[17:0]
000H	Dummy Bit	P1	Dummy Bit	P2
008H	Dummy Bit	P3	Dummy Bit	P4
010H	Dummy Bit	P5	Dummy Bit	P6
…				

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:50]	D[49:32]	D[31:18]	D[17:0]
000H	Dummy Bit	P2	Dummy Bit	P1
008H	Dummy Bit	P4	Dummy Bit	P3
010H	Dummy Bit	P6	Dummy Bit	P5
…				



NOTE 1: D [17:12] = Red data, D [11:6] = Green data, D [5:0] = Blue data

16BPP display (A555)

(BSWP=0, HWSWP=0, WSWP=0)

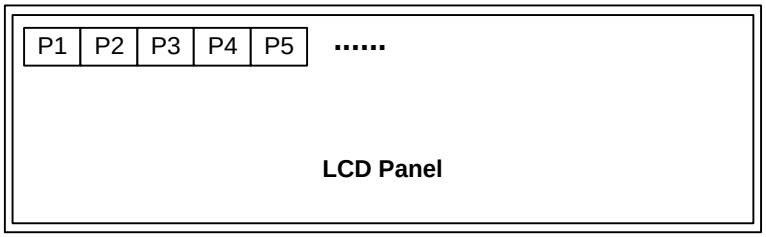
	D[63]	D[62:48]	D[47]	D[46:32]	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN1	P1	AEN2	P2	AEN3	P3	AEN4	P4
004H	AEN5	P5	AEN6	P6	AEN7	P7	AEN8	P8
008H	AEN9	P9	AEN10	P10	AEN11	P11	AEN12	P12
…								

(BSWP=0, HWSWP=0, WSWP=1)

	D[63]	D[62:48]	D[47]	D[46:32]	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN3	P3	AEN4	P4	AEN1	P1	AEN2	P2
004H	AEN7	P7	AEN8	P8	AEN5	P5	AEN6	P6
008H	AEN11	P11	AEN12	P12	AEN9	P9	AEN10	P10
…								

(BSWP=0, HWSWP=1, WSWP=0)

	D[63]	D[62:48]	D[47]	D[46:32]	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN4	P4	AEN3	P3	AEN2	P2	AEN1	P1
004H	AEN8	P8	AEN7	P7	AEN6	P6	AEN5	P5
008H	AEN12	P12	AEN11	P11	AEN10	P10	AEN9	P9
…								



NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends with alpha value selected by AEN.

Alpha value is selected by SFR value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, ALPHA1_B. For more information, refer to "Section SFR.

NOTE 2: D [14:10] = Red data, D [9:5] = Green data, D [4:0] = Blue data

16BPP display (1555)

(BSWP=0, HWSWP=0, WSWP=0)

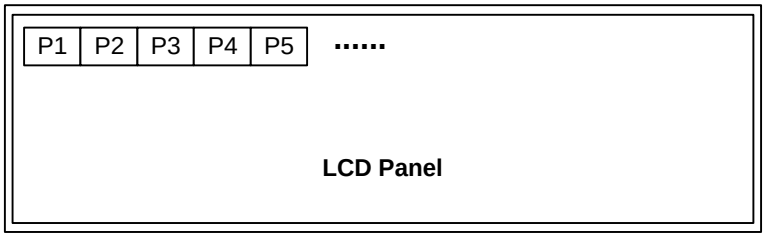
	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P1	P2	P3	P4
008H	P5	P6	P7	P8
010H	P9	P10	P11	P12
…				

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P3	P4	P1	P2
008H	P7	P8	P5	P6
010H	P11	P12	P9	P10
…				

(BSWP=0, HWSWP=1, WSWP=0)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P4	P3	P2	P1
008H	P8	P7	P6	P5
010H	P12	P11	P10	P9
…				



NOTE 1: {D [14:10], D [15]} = Red data, {D [9:5], D[15] } = Green data, {D[4:0], D[15]}= Blue data

6BPP display (565)

(BSWP=0, HWSWP=0, WSWP=0)

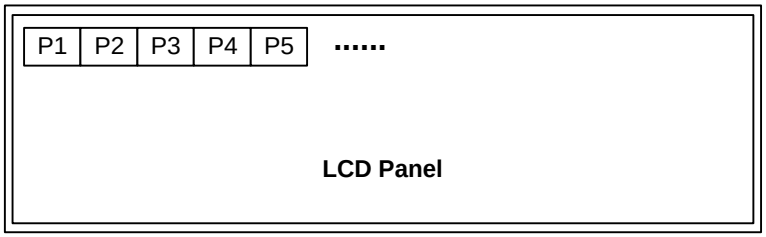
	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P1	P2	P3	P4
008H	P5	P6	P7	P8
010H	P9	P10	P11	P12
…				

(BSWP=0, HWSWP=0, WSWP=1)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P3	P4	P1	P2
008H	P7	P8	P5	P6
010H	P11	P12	P9	P10
…				

(BSWP=0, HWSWP=1, WSWP=0)

	D[63:48]	D[47:32]	D[31:16]	D[15:0]
000H	P4	P3	P2	P1
008H	P8	P7	P6	P5
010H	P12	P11	P10	P9
…				



NOTE 1: D [15:11] = Red data, D [10:5] = Green data, D [4:0] = Blue data

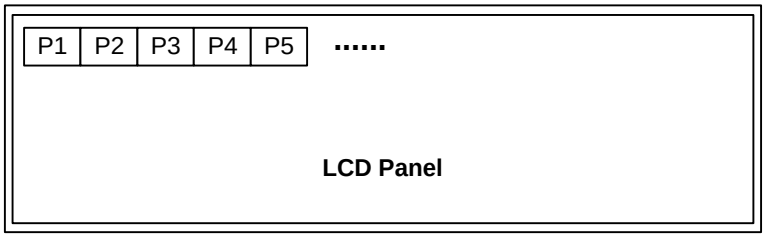
13BPP display (A444)

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:61]	D[62]	D[61:49]	D[48:46]	D[45]	D[44:32]	D[31:29]	D[28]	D[27:15]	D[31:29]	D[15]	D[14:0]
000H	Dummy	AEN1	P1	Dummy	AEN2	P2	Dummy	AEN3	P3	Dummy	AEN4	P4
004H	Dummy	AEN5	P5	Dummy	AEN6	P6	Dummy	AEN7	P7	Dummy	AEN8	P8
008H	Dummy	AEN9	P9	Dummy	AEN10	P10	Dummy	AEN11	P11	Dummy	AEN12	P12
…												

(BYSWP=0, HWSWP=1, WSWP=0)

	D[63:61]	D[62]	D[61:49]	D[48:46]	D[45]	D[44:32]	D[31:29]	D[28]	D[27:15]	D[31:29]	D[15]	D[14:0]
000H	Dummy	AEN4	P4	Dummy	AEN3	P3	Dummy	AEN2	P2	Dummy	AEN1	P1
004H	Dummy	AEN8	P8	Dummy	AEN7	P7	Dummy	AEN6	P6	Dummy	AEN5	P5
008H	Dummy	AEN12	P12	Dummy	AEN11	P11	Dummy	AEN10	P10	Dummy	AEN9	P9
…												



NOTE 1: If per-pixel blending is set, then this pixel blends with alpha value selected by D [31:24] D [31:24]. AR = AG = AB determines Alpha value. For more information, refer to "Section SFR.

NOTE 2: D [23:16] = Red data, D [15:8] = Green data, D[7:0] = Blue data

NOTE 3: 16BPP (4444) mode. (For more information, refer to "Section SFR) Data has Alpha value.

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:60]	D[59:48]	D[47:44]	D[43:32]	D[31:28]	D[27:16]	D[15:12]	D[11:0]
000H	ALPHA1	P1	ALPHA2	P2	ALPHA3	P3	ALPHA4	P4
004H	ALPHA5	P5	ALPHA6	P6	ALPHA7	P7	ALPHA8	P8
008H	ALPHA9	P9	ALPHA10	P10	ALPHA11	P11	ALPHA12	P12
…								

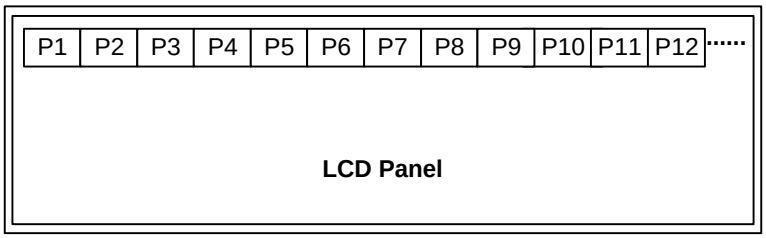
8BPP display (*A232*)

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63]	D[62:56]	D[55]	D[54:48]	D[47]	D[46:40]	D[39]	D[38:32]	D[31]	D[30:24]	D[23]	D[22:16]	D[15]	D[14:8]	D[7]	D[6:0]
000H	AEN	P1	AEN	P2	AEN	P3	AEN	P4	AEN	P5	AEN	P6	AEN	P7	AEN	P8
008H	AEN	P9	AEN	P10	AEN	P11	AEN	P12	AEN	P13	AEN	P14	AEN	P15	AEN	P16
010H	AEN	P17	AEN	P18	AEN	P19	AEN	P20	AEN	P21	AEN	P22	AEN	P23	AEN	P24
…																

(BYSWP=1, HWSWP=0, WSWP=0)

	D[63]	D[62:56]	D[55]	D[54:48]	D[47]	D[46:40]	D[39]	D[38:32]	D[31]	D[30:24]	D[23]	D[22:16]	D[15]	D[14:8]	D[7]	D[6:0]
000H	AEN	P8	AEN	P7	AEN	P6	AEN	P5	AEN	P4	AEN	P3	AEN	P2	AEN	P1
008H	AEN	P16	AEN	P15	AEN	P14	AEN	P13	AEN	P12	AEN	P11	AEN	P10	AEN	P9
010H	AEN	P24	AEN	P23	AEN	P22	AEN	P21	AEN	P20	AEN	P19	AEN	P18	AEN	P17
…																



NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends alpha value selected by AEN.
Alpha value is selected by SFR value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, ALPHA1_B. For more information, refer to "Section SFR.

NOTE 2: D [6:5] = Red data, D [4:2] = Green data, D [1:0] = Blue data

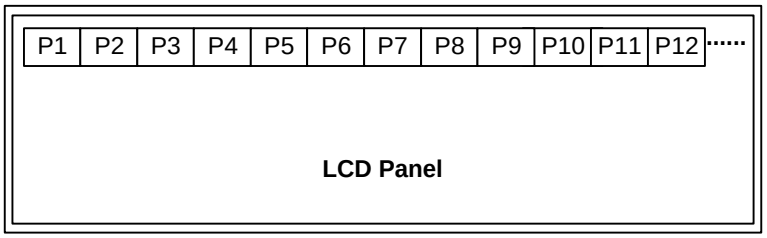
8BPP display (*Palette*)

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:56]	D[55:48]	D[47:40]	D[39:32]	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P1	P2	P3	P4	P5	P6	P7	P8
008H	P9	P10	P11	P12	P13	P14	P15	P16
010H	P17	P18	P19	P20	P21	P22	P23	P24
…								

(BYSWP=1, HWSWP=0, WSWP=0)

	D[63:56]	D[55:48]	D[47:40]	D[39:32]	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P8	P7	P6	P5	P4	P3	P2	P1
008H	P16	P15	P14	P13	P12	P11	P10	P9
010H	P24	P23	P22	P21	P20	P19	P18	P17
…								



NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends with alpha value selected by AEN.
SFR value selects Alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G; ALPHA1_B.
For more information, refer to "Section SFR.

4BPP Display (Palette)

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:60]	D[59:56]	D[55:52]	D[51:48]	D[47:44]	D[43:40]	D[39:36]	D[35:32]
000H	P1	P2	P3	P4	P5	P6	P7	P8
008H	P17	P18	P19	P20	P21	P22	P23	P24
…								

	D[31:28]	D[27:24]	D[23:20]	D[19:16]	D[15:12]	D[11:8]	D[7:4]	D[3:0]
000H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P25	P26	P27	P28	P29	P30	P31	P32
…								

(BYSWP=1, HWSWP=0, WSWP=0)

	D[63:60]	D[59:56]	D[55:52]	D[51:48]	D[47:44]	D[43:40]	D[39:36]	D[35:32]
000H	P15	P16	P13	P14	P11	P12	P9	P10
008H	P31	P32	P29	P30	P27	P28	P25	P26
…								

	D[31:28]	D[27:24]	D[23:20]	D[19:16]	D[15:12]	D[11:8]	D[7:4]	D[3:0]
000H	P7	P8	P5	P6	P3	P4	P1	P2
008H	P23	P24	P21	P22	P19	P20	P17	P18
…								

NOTE 1: AEN: Transparency value selection bit
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends with alpha value selected by AEN.
Alpha value is selected by SFR value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, ALPHA1_B. For more information, refer to "Section SFR.

BPP Display (Palette)

(BYSWP=0, HWSWP=0, WSWP=0)

	D[63:62]	D[61:60]	D[59:58]	D[57:56]	D[55:54]	D[53:52]	D[51:50]	D[49:48]
000H	P1	P2	P3	P4	P5	P6	P7	P8
008H	P33	P34	P35	P36	P37	P38	P39	P40
…								

	D[47:46]	D[45:44]	D[43:42]	D[41:40]	D[39:38]	D[37:36]	D[35:34]	D[33:32]
000H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P41	P42	P43	P44	P45	P46	P47	P48
…								

	D[31:30]	D[29:28]	D[27:26]	D[25:24]	D[23:22]	D[21:20]	D[19:18]	D[17:16]
000H	P17	P18	P19	P20	P21	P22	P23	P24
008H	P49	P50	P51	P52	P53	P54	P55	P56
…								

	D[15:14]	D[13:12]	D[11:10]	D[9:8]	D[7:6]	D[5:4]	D[3:2]	D[1:0]
000H	P25	P26	P27	P28	P29	P30	P31	P32
008H	P57	P58	P59	P60	P61	P62	P63	P64
…								

NOTE1: If ALPHAPAL is enabled, then the MSB of Palette memory is acting as AEN bit.
AEN = 0: Select ALPHA0
AEN = 1: Select ALPHA1

If per-pixel blending is set, then this pixel blends with alpha value selected by AEN. SFR selects Alpha value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G; ALPHA1_B. For more information, refer to "Section SFR.

3.4 PALETTE USAGE

3.4.1 Palette Configuration and Format Control

The display controller supports 256-color palette for various selection of color mapping.

You can select 256 colors from the 32-bit colors through these formats.
256 color palette consist of the 256 (depth) × 32-bit SPSRAM. Palette supports 8:8:8, 6:6:6, 5:6:5(R: G: B), etc format.
For example of A:5:5:5 format, write palette as shown in Table 9.1-2 and then connect VD pin to TFT LCD panel(R(5)=VD[23:19], G(5)=VD[15:11], and B(5)=VD[7:3]). The AEN bit controls the blending function enable or disable. At the end, Set WPALCON (**W1PAL, case window0)** register to 0'b101. Specially, 32-bit (8:8:8:8) format has an alpha value directly without using alpha value register (ALPHA_0/1)

Table 9.1-1 32BPP (8:8:8:8) Palette Data Format

INDEX / Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	ALPHA								R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
01h	ALPHA								R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	ALPHA								R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 9.1-2 25BPP (A: 8:8:8) Palette Data Format

INDEX / Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	-	-	-	-	-	-	-	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
01h	-	-	-	-	-	-	-	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	-	-	-	-	-	-	-	AEN	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 9.1-3 19BPP (A: 6:6:6) Palette Data Format

INDEX / Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
01h	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	22	21	20	19	18	15	14	13	12	11	10	7	6	5	4	3	2

Table 9.1-4 16BPP (A: 5:5:5) Palette Data Format

INDEX / Bit Pos.	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
01h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
.....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
FFh	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
Number of VD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	22	21	20	19	15	14	13	12	11	7	6	5	4	3

3.4.2 Palette Read/Write

You must not access Palette memory during the ACTIVE status of the Vertical Status (VSTATUS) register. The VSTATUS must be checked to do Read/Write operation on the palette.

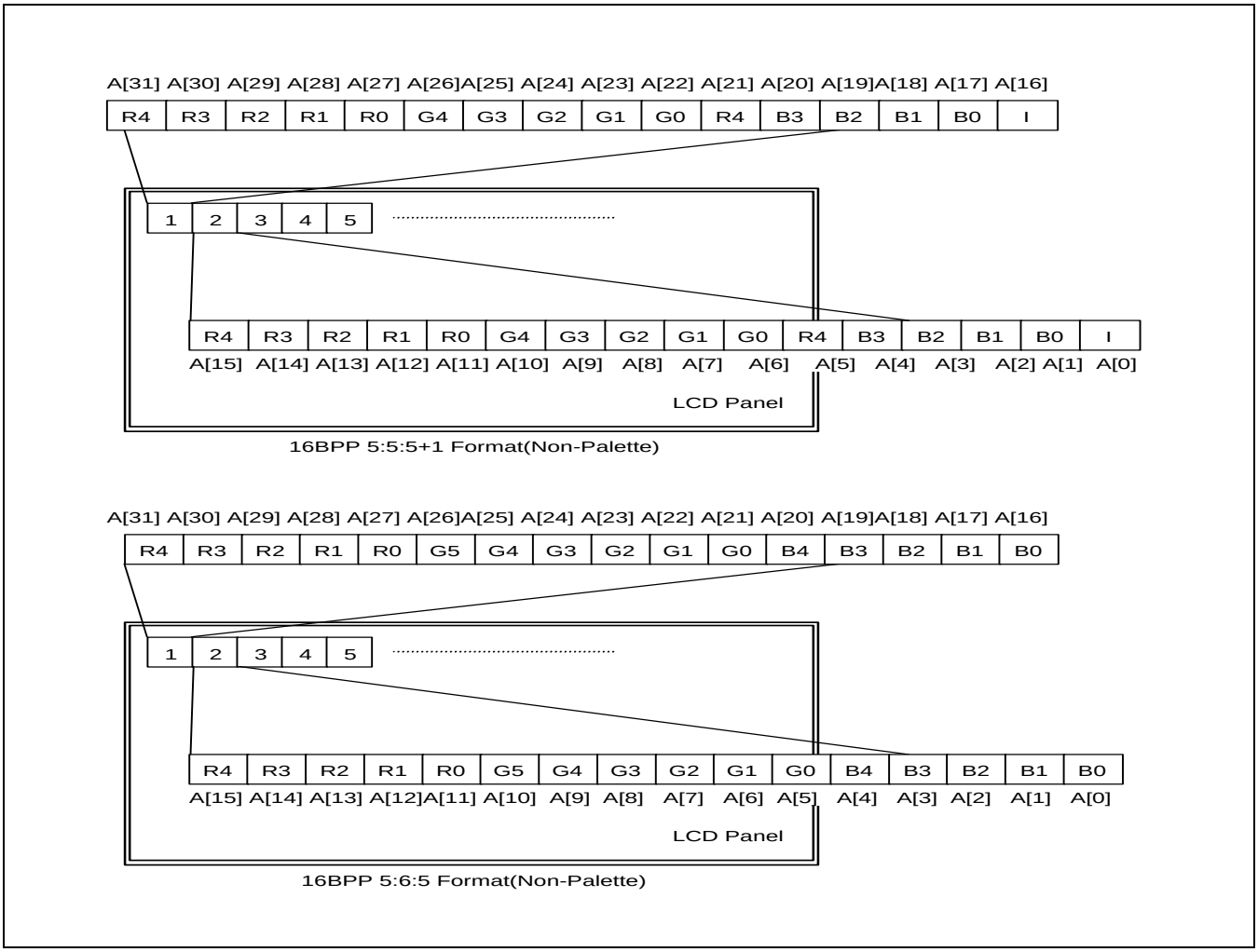


Figure 9.1-4 16BPP (5:6:5) Display Types

3.5 WINDOW BLENDING

3.5.1 Overview

The main function of the VPRCS module is window blending. Display controller has 5 window layer and details is described below. As a example of application, System uses win0 as a OS window, full TV screen window or etc., win1 as a small (next channel) TV screen with win2 as a memu, win3 as a caption, win 4 as a channel information. Win3 and win4 has the color limitation by using color index with Color LUT. This feature enhances the system performance by reducing the data rate of total system.

Total 5 window (example)

- win 0 (base): Local / (YCbCr ,RGB without palette)
- win 1 (Overlay1): RGB with palette
- win 2 (Overlay2): RGB with palette
- win 3 (Caption): RGB (1/2/4) with 16 level Color LUT
- win 4 (Cursor): RGB (1/2) with 4 level Color LUT

Overlay Priority

Win 4 > Win 3 > Win 2 > Win 1 > Win 0

Color Key

:The register value to ColorKey register must be set by 24-bit RGB format.

Blending equation

<Data blending>

Win01(R,G,B) = Win0(R,G,B) x b1 + Win1(R,G,B) x a1

Win012(R/G/B) = Win01(R/G/B) x b2 + Win2(R/G/B) x a2

Win0123(R/G/B) = Win012(R/G/B) x b3 + Win3(R/G/B) x a3

WinOut(R/G/B) = Win0123(R/G/B) x b4 + Win4(R/G/B) x a4

Where

Win0(R) = Window 0's Red data,

Win0(G) = Window 0's Green data,

Win0(B) = Window 0's Blue data,

Win1(R) = Window 1's Red data,...

...

b1 = Background's Data blending equation1 factor,

a1 = Foreground's Data blending equation1 factor,

b2 = Background's Data blending equation2 factor,

a2 = Foreground's Data blending equation2 factor,...

<Alpha value blending>

AR(G,B)01 = AR(G,B)0 x q1 + AR(G,B)1 x p1

AR(G,B)012 = AR(G,B)01 x q2 + AR(G,B)2 x p2

AR(G,B)0123 = AR(G,B)012 x q3 + AR(G,B)3 x p3

Where,

AR0 = Window 0's Red blending factor,

AG0 = Window 0's Green blending factor,

AB0 = Window 0's Blue blending factor,

AR1 = Window 1's Red blending factor,...

AR01 = Window01's Red blending factor (alpha value blending between AR0 and AR1),

AG01 = Window01's Green blending factor (alpha value blending between AG0 and AG1),

AB01 = Window01's Blue blending factor (alpha value blending between AB0 and AB1),

AR012 = Window012's Red blending factor (alpha value blending between AR01 and AR2),

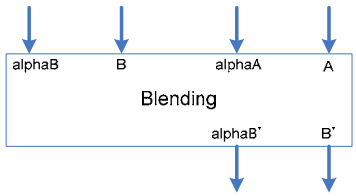
...

q1 = Background's Alpha value blending equation1 factor,

p1 = Foreground's Alpha value blending equation1 factor,

q2 = Background's Alpha value blending equation2 factor,

p2 = Foreground's Alpha value blending equation2 factor,...



B' = a x A + b x B

alphaB' = p x alphaA + q x alphaB

a/b/p/q = 0
or 1
or alphaA
or 1 - alphaA
or alphaB
or 1 - alphaB
or A
or 1 - A
or B
or 1 - B

Figure 9.1-5 Blending Equation

<Default blending equation>

- Data blending>

B' = B x (1-alphaA) + A x alphaA

- Alpha value blending>

alphaB' = 0 (= alphaB x 0 + alphaA x 0)

3.5.2 Blending Diagram/ Details

Display controller could blend 5 Layer for one pixel at the same time. ALPHA0_R,ALPHA0_G,ALPHA0_B, ALPHA1_R,ALPHA1_G,ALPHA1_B register controls alpha value (Blending factor), which are implemented for each window layer and color (R,G,B). The example below is for the R (Red) output using ALPHA_R value of each windows. As a special feature, all window has two kinds of alpha blending value. One is alpha value for transparnacy enable (AEN value ==1), the other is alpha value for transparnacy disable (AEN value == 0). **If WINEN_F is enabled and BLD_PIX is enabled** then AR will be choosen by the below equation.

AR = (Pixel(R)'s AEN value == 1'b1) ? Reg(ALPHA1_R) : Reg(ALPHA0_R);

AG = (Pixel(G)'s AEN value == 1'b1) ? Reg(ALPHA1_G) : Reg(ALPHA0_G);

AB = (Pixel(B)'s AEN value == 1'b1) ? Reg(ALPHA1_B) : Reg(ALPHA0_B);

(where, BLD_ PIX == 1)

If WINEN_F is enabled and BLD_PIX is disabled then AR will be ALPHA0_R only. In this case blending factor AR is fixed by ALPHA0_R, not using the AEN bit information anymore.

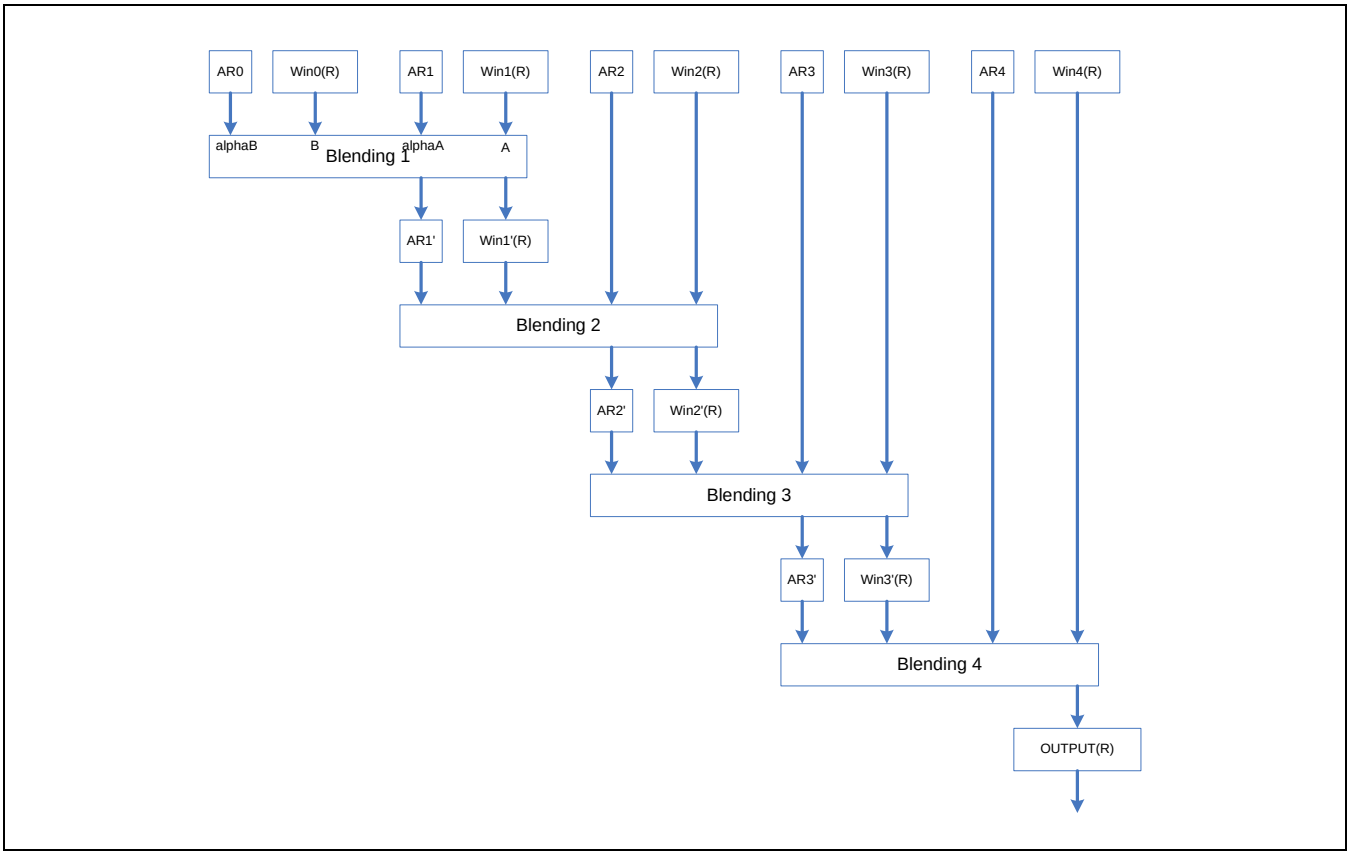


Figure9.1-6 Blending Diagram

Example: Window n's blending factor decision (n=0, 1, 2, 3, 4). For more information, refer to "Section SFR".

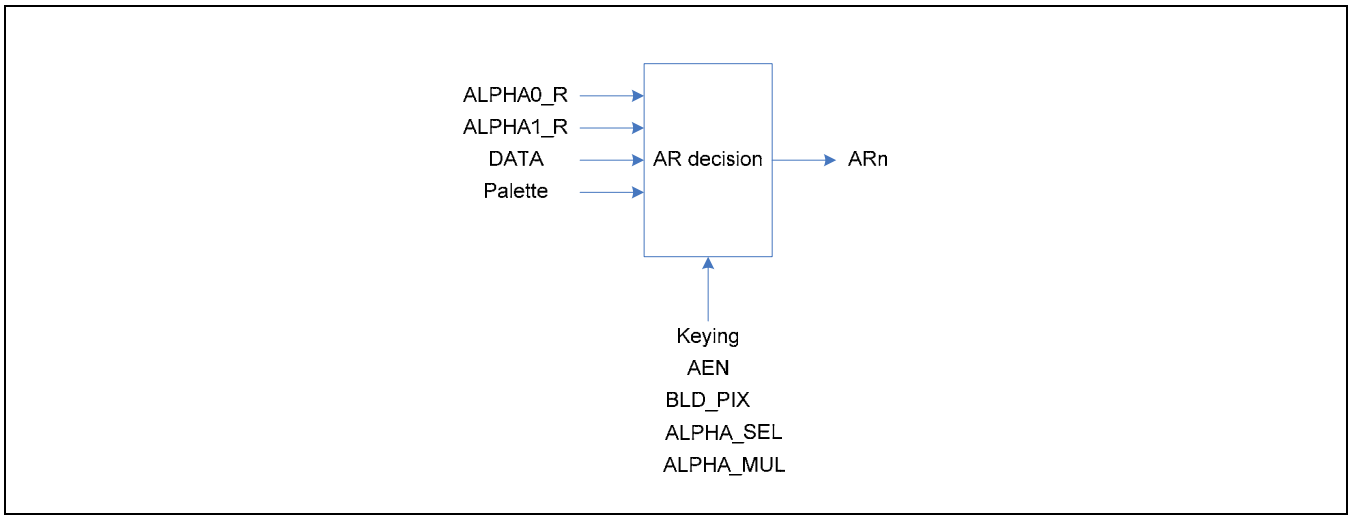


Figure 9.1-7 Blending Factor Decision

NOTE: Alpha value is {DATA [15:12], DATA [15:12]} (4bit->8bit expanding), if DATA [15:12] (BPPMODE_F = b'1110, ARGB4444 format) is used to blend.

3.5.3 COLOR-KEY Function

The display controller supports color-key function for the various effect of image mapping. COLOR-KEY register specifying Color image of OSD layer substitutes by background image for special functionality, as cursor image or pre-view image of the camera.

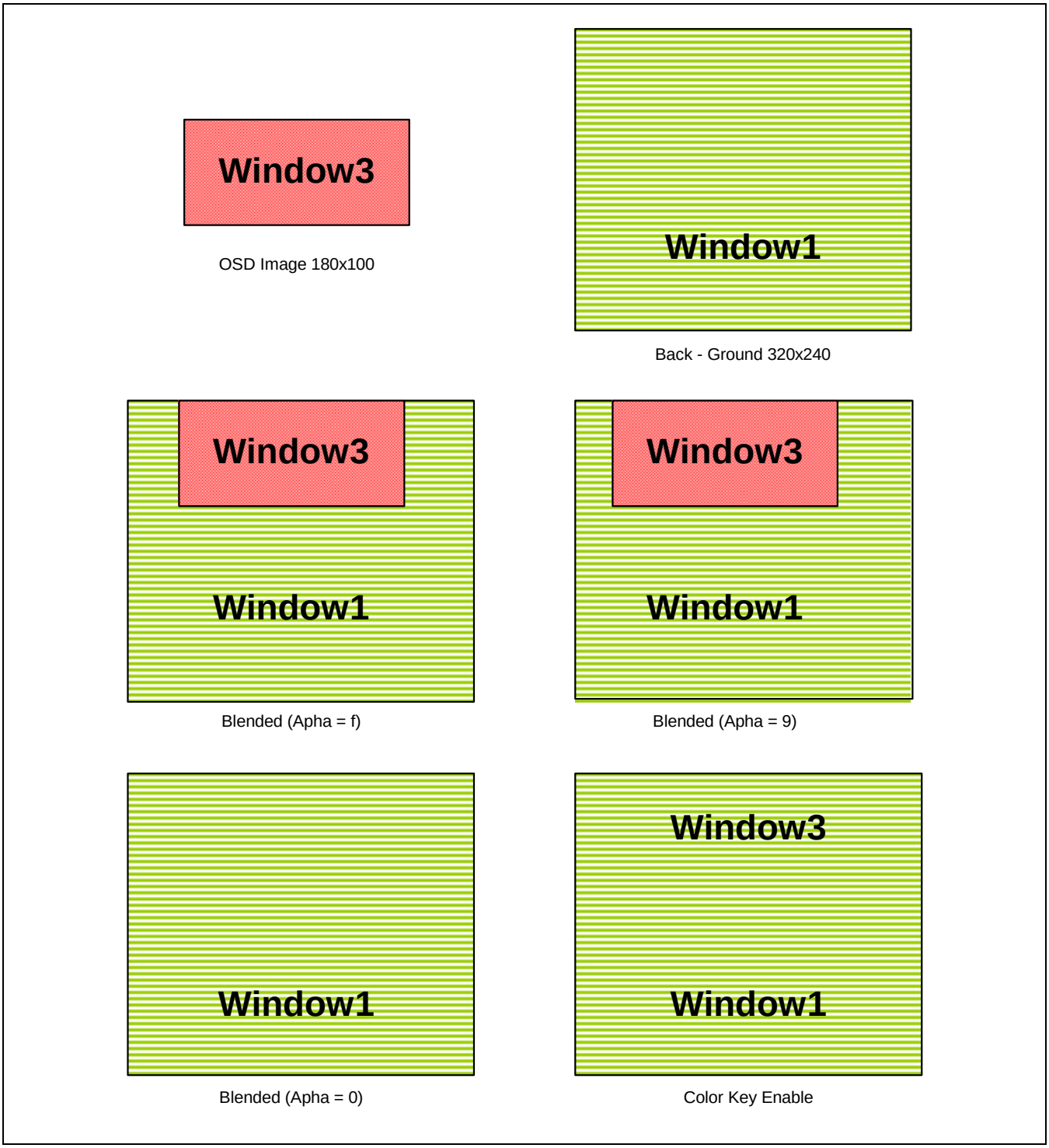


Figure 9.1-8 Color-Key Function Configurations

3.6 VTIME CONTROLLER OPERATION

VTIME has two blocks. One is VTIME_RGB_TV for RGB interface and ITU-R601/ 656 interface timing control. The other is VTIME_I80 for indirect i80 interface timing control.

3.6.1 RGB Interface Controller

The VTIME generates the control signals such as, RGB_VSYNC, RGB_HSYNC, RGB_VDEN and RGB_VCLK signal for RGB interface. These control signals are highly related with the configuration on the VIDTCON0/1/2 registers in the VSFR register. Based on these programmable configurations of the display control registers in VSFR, the VTIME module generates the programmable control signals suitable for the support of many different types of display device.

The RGB_VSYNC signal asserts to cause the LCD's line pointer to start over at the top of the display. The configuration of both HOZVAL field and LINEVAL registers controls pulse generation of RGB_VSYNC and RGB_HSYNC. The size of the LCD panel according to the following equations determines HOZVAL and LINEVAL:

- **HOZVAL = (Horizontal display size) -1**
- **LINEVAL = (Vertical display size) –1**

The CLKVAL field in the VIDCON0 register controls rate of RGB_VCLK signa. The table below defines the relationship of RGB_VCLK and CLKVAL. The minimum value of CLKVAL is 1.

RGB_VCLK (Hz) =HCLK/ (CLKVAL+1) where CLKVAL >= 1

Table 9.1-5 Relation 16BPP between VCLK and CLKVAL (TFT, Frequency of Video Clock Source=60MHz)

CLKVAL	60MHz/X	VCLK
2	60 MHz/3	20.0 MHz
3	60 MHz/4	15.0 MHz
:	:	:
63	60 MHz/64	937.5 kHz

VSYNC, VBPD, VFPD, HSYNC, HBPD, HFPD, HOZVAL and LINEVAL configure RGB_HSYNC and RGB_VSYNC signal. Refer to Figure 11.

The frame rate is RGB_VSYNC signal frequency. The frame rate is related with the field of VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, CLKVAL registers. Most LCD driver needs own adequate frame rate. To calculate frame rate use the following equation:

Frame Rate = 1/ [{ (VSPW+1) + (VBPD+1) + (LIINEVAL + 1) + (VFPD+1) } x {(HSPW+1) + (HBPD +1) + (HFPD+1) + (HOZVAL + 1) } x { (CLKVAL+1) / (Frequency of Clock source) }]

3.6.2 I80 Interface Controller

VTIME_I80 controls DISPLAY Controller for CPU style LDI, and has the following functions.

- Generates of I80 Interface Control Signals
- CPU Style LDI Command Control
- Timing Control for VDMA and VDPRCS

3.6.3 Output Control Signal Generation

VTIME_I80 generates SYS_CS0, SYS_CS1, SYS_WE and SYS_RS control signals (For more information, refer to Figure 9.1-13 for Timing Diagram). Their timing parameters, LCD_CS_SETUP, LCD_WR_SETUP, LCD_WR_ACT, and LCD_WR_HOLD are set through I80IFCONA0 and I80IFCONA1 SFRs.

3.6.4 Partial Display Control

Although partial display is a main feature of CPU-style LDI, VTIME_I80 does not support this function in H/W logic.

SFR setting (LINEVAL, HOZVAL, OSD_LeftTopX_F, OSD_LeftTopY_F, OSD_RightBotX_F, OSD_RightBotY_F, PAGEWIDTH, and OFFSIZE) implements this function.

3.6.5 LDI Command Control

LDI receives command and data. Command refers to index for the selection of SFR in LDI. In control signal for command and data, only SYS_RS signal has different operation. Generally, SYS_RS is polarity of '1' for command issue and vice versa.

DISPLAY Controller has two kinds of command control. One is auto command and the other is normal command.

Auto command is issued automatically (i.e. without S/W control) at a predefined rate (rate = 2, 4, 6 ...30. Rate = 4 means auto command are send to LDI at the end of every 4-image-frames). S/W control issues Normal command.

3.7 SETTING OF COMMANDS

3.7.1 Auto Command

For example, if 0x1(index), 0x32, 0x2(index), 0x8f, 0x4(index), 0x99 required to sent to LDI at every 10 frames, the following steps are recommended

LDI_CMD0 ← 0x1, LDI_CMD1 ← 0x32, LDI_CMD2 ← 0x2,

LDI_CMD3 ← 0x8f, LDI_CMD4 ← 0x4, LDI_CMD5 ← 0x99

CMD0_EN ← 0x2, CMD1_EN ← 0x2, CMD2_EN ← 0x2,

CMD3_EN ← 0x2, CMD4_EN ← 0x2, CMD5_EN ← 0x2

CMD0_RS ← 0x1, CMD1_RS ← 0x0, CMD2_RS ← 0x1,

CMD3_RS ← 0x0, CMD4_RS ← 0x1, CMD5_RS ← 0x0

AUTO_CMD_RATE ← 0x5

NOTE 1: For RS polarity, refer to your LDI specification.

NOTE 2: It is not required to pack LDI_CMD from LDI_CMD0 to LDI_CMD11 contiguously. For example, only the use of LDI_CMD0, LDI_CMD3 and LDI_CMD11 is possible.

NOTE 3: Maximum 12 auto commands are available.

3.7.2 Normal Command

1) Put commands into LDI_CMD0 ~ 11 (maximum 12 commands)
2) Set CMDx_EN in LDI_CMDCON0 to enable normal command x (For example, if you want to enable command 4, you have to set CMD4_EN to 0x01).
3) Set NORMAL_CMD_ST in I80IFCONB0/1

- DISPLAY Controller has the following miscellaneous traits for command operations.
- Auto / Normal / Auto and Normal command mode is possible for each 12 commands
- DISPLAY Controller sends 12 maximum commands between frames in its normal operation (Normal operation means ENVID=1 and video data are displayed in LCD panel)
- Issues Commands in the order of CMD0 → CMD1 → CMD2 → CMD3 → … → CMD10 → CMD11
- Disabled commands (CMDx_EN = 0x0) are skipped
- Sending over 12 commands

  : It is possible in normal command and suitable in system initialization.

  1) Setting 12 LDI_CMDx, CMDx_EN, CMDx_RS
  2) Set NORMAL_CMD_ST
  3) Read NORMAL_CMD_ST with polling. If 0, go to NORMAL_CMD_ST Setting

Command Setting Example

** CMD0_EN = 2'b10, CMD1_EN = 2'b11, CMD2_EN = 2'b01, CMD3_EN = 2'b11, CMD4_EN = 2'b01
(Auto Command: CMD0, CMD1, CMD3, Normal Command: CMD1, CMD2, CMD3, And CMD4)

** AUTO_COMMAND_RATE = 4'b0010 (per 4 frames)

** CMD0_RS = 1, CMD1_RS = 1, CMD2_RS = 0, CMD3_RS = 1, CMD4_RS = 0

** RSPOL = 0

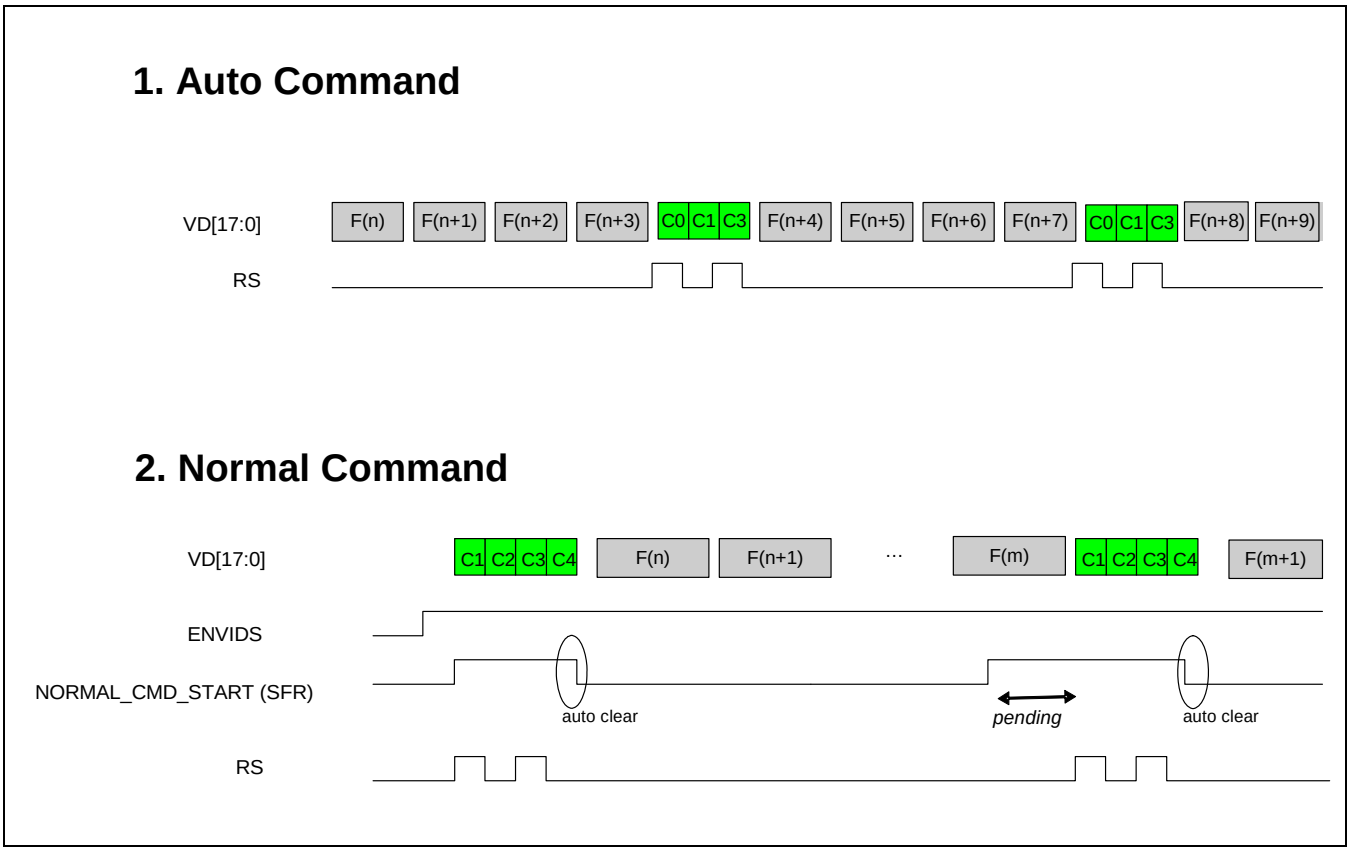


Figure 9.1- 9 Sending Command

Indirect I80 Interface Trigger

VTIME_I80 starts its operation if an S/W trigger occurs. There are two kinds of triggers.
S/W trigger is generated by setting TRGCON SFR.

3.7.3 Interrupt

Completion of one frame generates Frame Done Interrupt.

Indirect I80 Interface Output mode

The following table shows the output mode of Indirect i80 interface according to mode@VIDCON0

Table 9.1-6 i80 Output Mode

VIDCON0 register	value	BPP	BUS width	Split	DATA	Command
DSI_EN	1	24	24	X	{ R[7:0],G[7:0],B[7:0] }	CMD [23:0]
L0/1_DATA	000	16	16	X	{ R[7:3],G[7:2],B[7:3] }	CMD [15:0]
	001	18	16	O (1st) (2nd)	{ R[7:2],G[7:2],B[7:4] } { 14'b0,B[3:2] }	CMD [15:0] -
	010	18	9	O (1st) (2nd)	{ R[7:2],G[7:5] } { G[4:2],B[7:2] }	CMD [17:9] CMD [8:0]
	011	24	16	O (1st) (2nd)	{ R[7:0],G[7:0] } { B[7:0], 8'b0 }	- -
	100	18	18	X	{ R[7:2],G[7:2],B[7:2] }	CMD [17:0]
	101	16	8	O (1st) (2nd)	{ R[7:3],G[7:5] } { G[4:2],B[7:3] }	CMD [15:8] CMD [7:0]

3.8 VIRTUAL DISPLAY

The display controller supports the hardware horizontal or vertical scrolling. If the screen is scrolled, change the fields of LCDBASEU and LCDBASEL (refer to Figure 9.1-7) but not the values of PAGEWIDTH and OFFSIZE. The size of video buffer in which the image is stored should be larger than the LCD panel screen size.

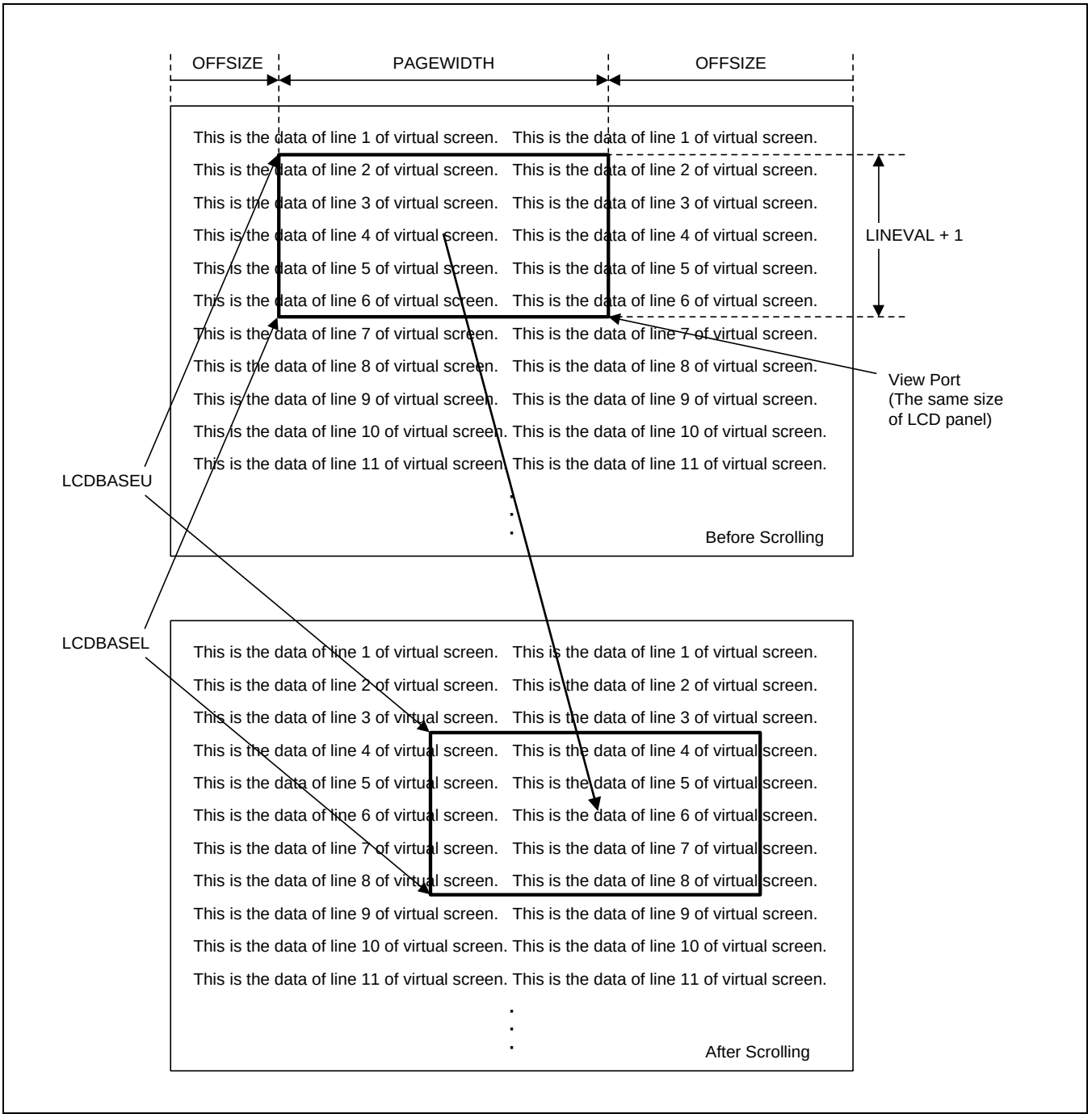


Figure 9.1-10 Example of Scrolling in Virtual Display

4 PROGRAMMER'S MODEL

4.1 OVERVIEW

Use the following registers to configure display controller:

1. VIDCON0: Configures Video output format and display enable/disable.
2. VIDCON1: RGB I/F control signal.
3. VIDCON2: Output data format control signal
4. I80IFCONx: CPU I/F control signal.
5. ITUIFCON: ITU I/F control signal.
6. VIDTCONx: Configures Video output Timing and determine the size of display.
7. WINCONx: Sets Each window format
8. VIDOSDxA, VIDOSDxB: Window position setting
9. VIDOSDxD: Sets OSD size
10. VIDWxALPHA0/1: Sets Alpha value
11. BLENDEQx: Sets Blending equation
12. VIDWxxADDx: Sets Source image address setting
13. WxKEYCONx: Color key value register
14. WINxMAP: Window color control
15. WPALCON: Palette control register

5 I/O DESCRIPTION

5.1 RGB INTERFACE

5.1.1 Signals

Function Signal	I/O	Description	PAD	Type
LCD_HSYNC	Output	Horizontal Sync. Signal	XvHSYNC	Muxed
LCD_VSYNC	Output	Vertical Sync. Signal	XvVSYNC	Muxed
LCD_VCLK	Output	LCD Video Clock	XvVCLK	Muxed
LCD_VDEN	Output	Data Enable	XvVDEN	Muxed
LCD_VD[23:0]	Output	YCbCr data output	XvVD[23:0]	Muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

5.1.2 Timing

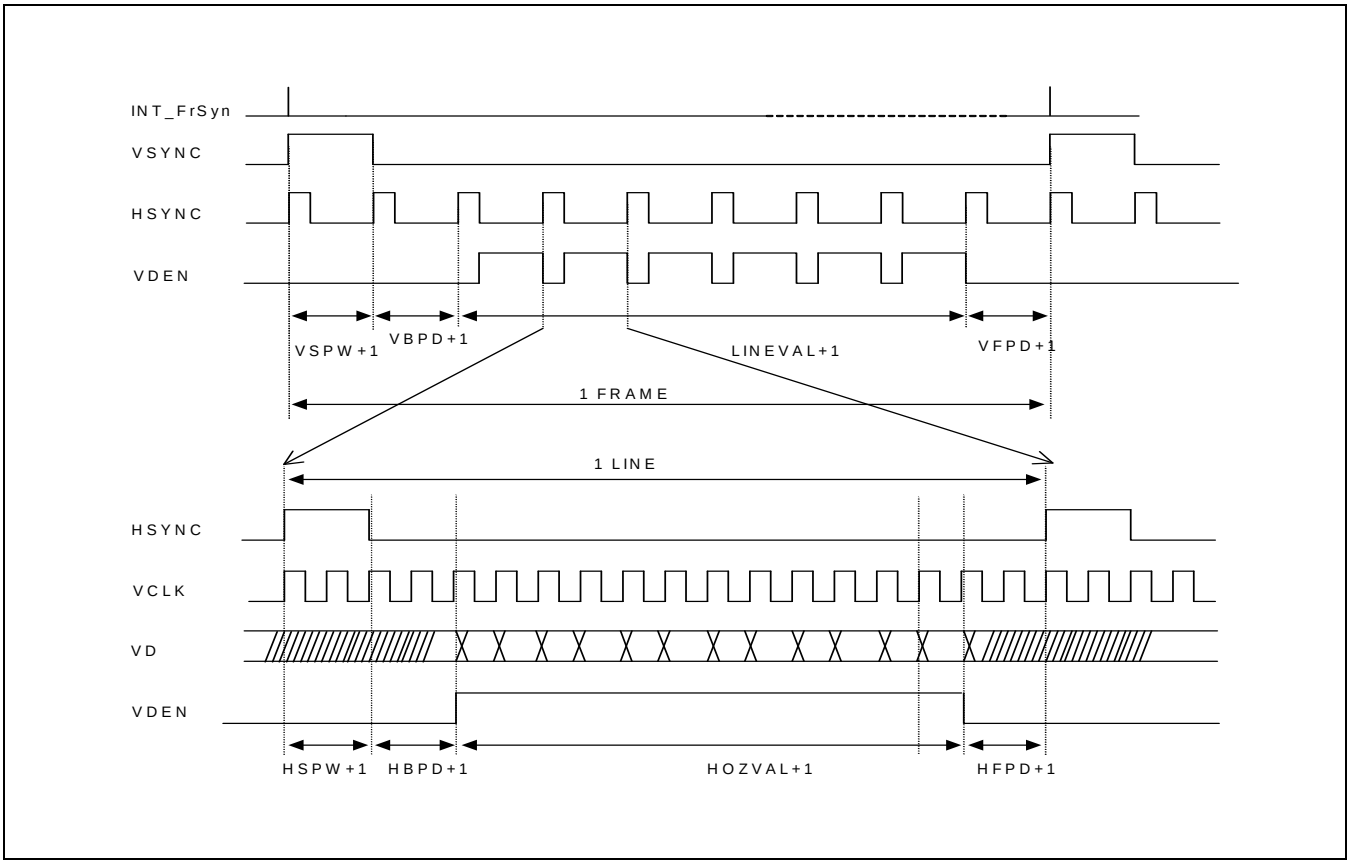


Figure 9.1-11 LCD RGB Interface Timing

5.2 LCD INDIRECT I80 SYSTEM INTERFACE

5.2.1 Signals

Function Signal	IO	Description	PAD	Type
SYS_VD[17:0]	Input/Output	Video Data	XvVD[17:0]	Muxed
SYS_CS0	Output	Chip select for LCD0	XvHSYNC	Muxed
SYS_CS1	Output	Chip select for LCD1	XvVSYNC	Muxed
SYS_WE	Output	Write enable	XvVCLK	Muxed
SYS_OE	Output	Output Enable	XvVD[23]	Muxed
SYS_RS/SYS_ADD[0]	Output	Address Output SYS_ADD[0] is Register/State select	XvVDEN	Muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

* MIPI DSI mode (when VIDCON0 [30] =1)

SYS_ADD [1] = SYS_ST: 0 when VDOUT is from Frame

1 when VDOUT is from Command

5.2.2 Timing

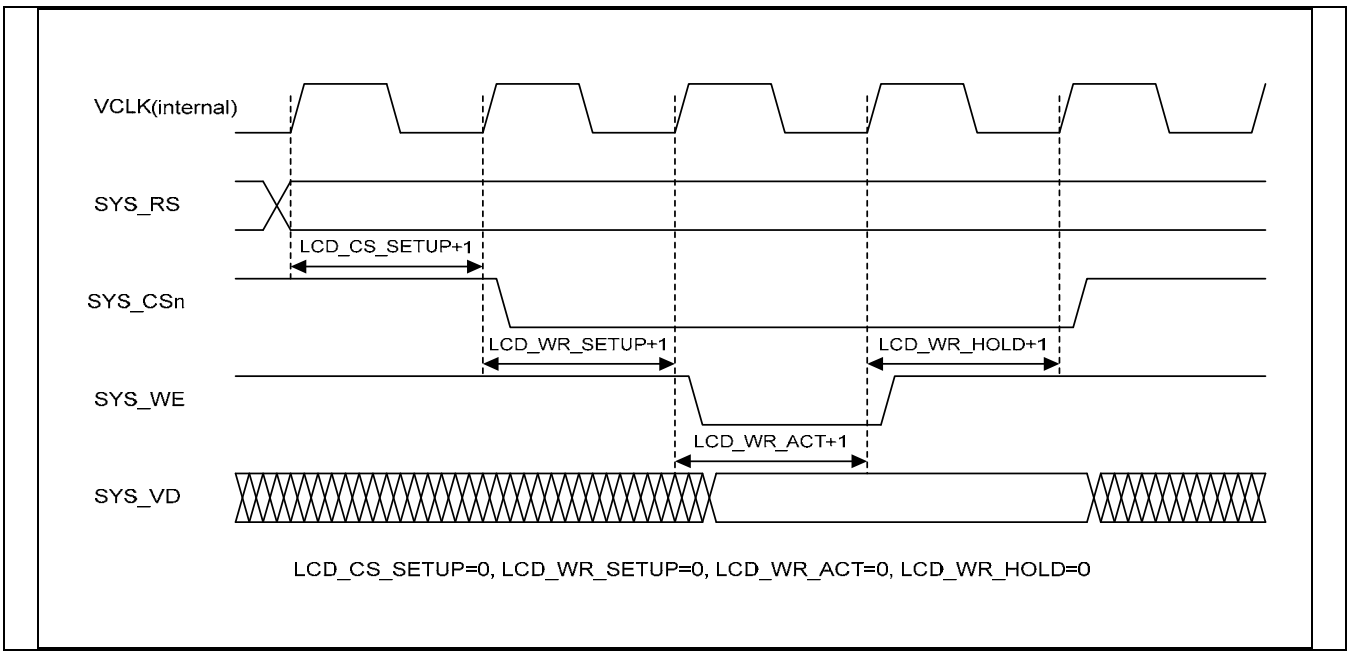


Figure 9.1-12 Indirect i80 System Interface WRITE Cycle Timing

5.3 ITU-R BT. 601/656 INTERFACE

5.3.1 Signals

Function Signal	IO	Description	PAD	Type
V601_CLK	Output	ITU 601 data clock	XvVCLK	Muxed
*VEN_HREF	Output	ITU601 DATA Enable	XvVDEN	Muxed
**VEN_VSYNC	Output	ITU601 Vertical Sync Signal	XvVSYNC	Muxed
VEN_HSYNC	Output	ITU601 Horizontal Sync Signal	XvHSYNC	Muxed
**VEN_FIELD	Output	ITU601 FIELD Signal (option)	XvVD[23]	Muxed
VEN_DATA[7:0]	Output	ITU601 YUV422 format data output	XvVD[7:0]	Muxed
V656_CLK	Output	ITU 656 data clock	XvVD[22]	Muxed
V656_DATA[7:0]	Output	ITU656 YUV422 format data output	XvVD[15:8]	Muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.
VEN_HREF: DATA Blank (when I601HREF [0] = 1) DATA Enable (If I601HREF [0] = 0)
VEN_VSYNC, VEN_FIELD (field information in interlace mode)

If SELVSYNC[0] = 1, Delay Cycle = DLYVSYNC[7:0] + 1

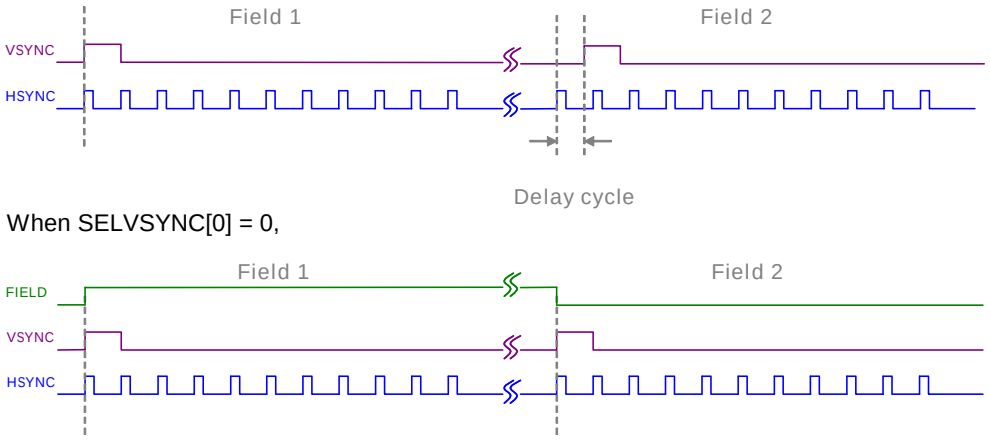
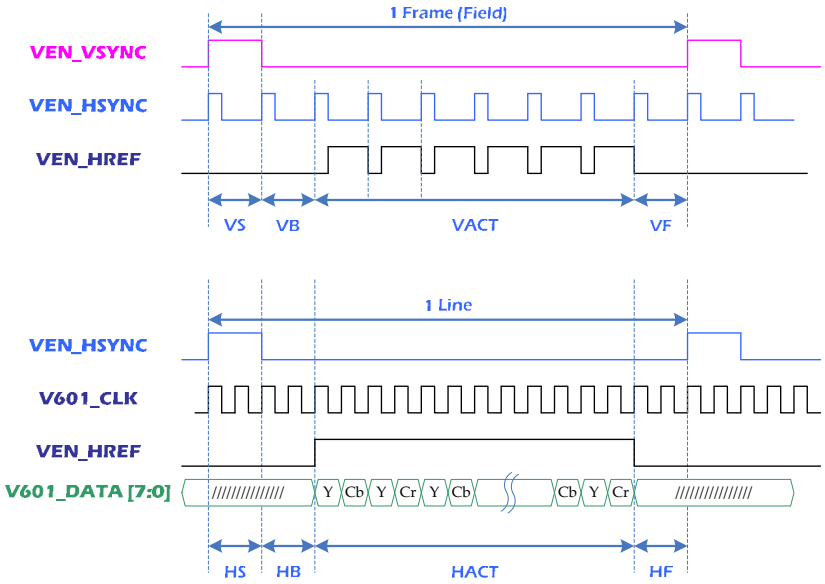


Figure 9.1-13 ITU-R BT.601 Controllable Vsync



HS	=	(HSPW + 1) x2 @VCLK	VS	=	(VSPW + 1) @HSYNC	
HB	=	(HBPD + 1) x2 @VCLK	VB	=	(VBPD + 1) @HSYNC	: Progressive, Odd Field(Interlaced)
HACT	=	(HOZVAL + 1) x2 @VCLK		=	(VBPDE + 1) @HSYNC	: Even Field (Interlaced)
HF	=	(HFPD + 1) x2 @VCLK	VACT	=	(LINEVAL + 1) @HSYNC	: Progressive
				=	(LINEVAL+1)/2 @HSYNC	: Interlaced
			VF	=	(VFPD + 1) @HSYNC	: Progressive, Odd Field(Interlaced)
				=	(VFPDE + 1) @HSYNC	: Even Field (Interlaced)

Figure 9.1-14 ITU-R BT.601 Interface Timing

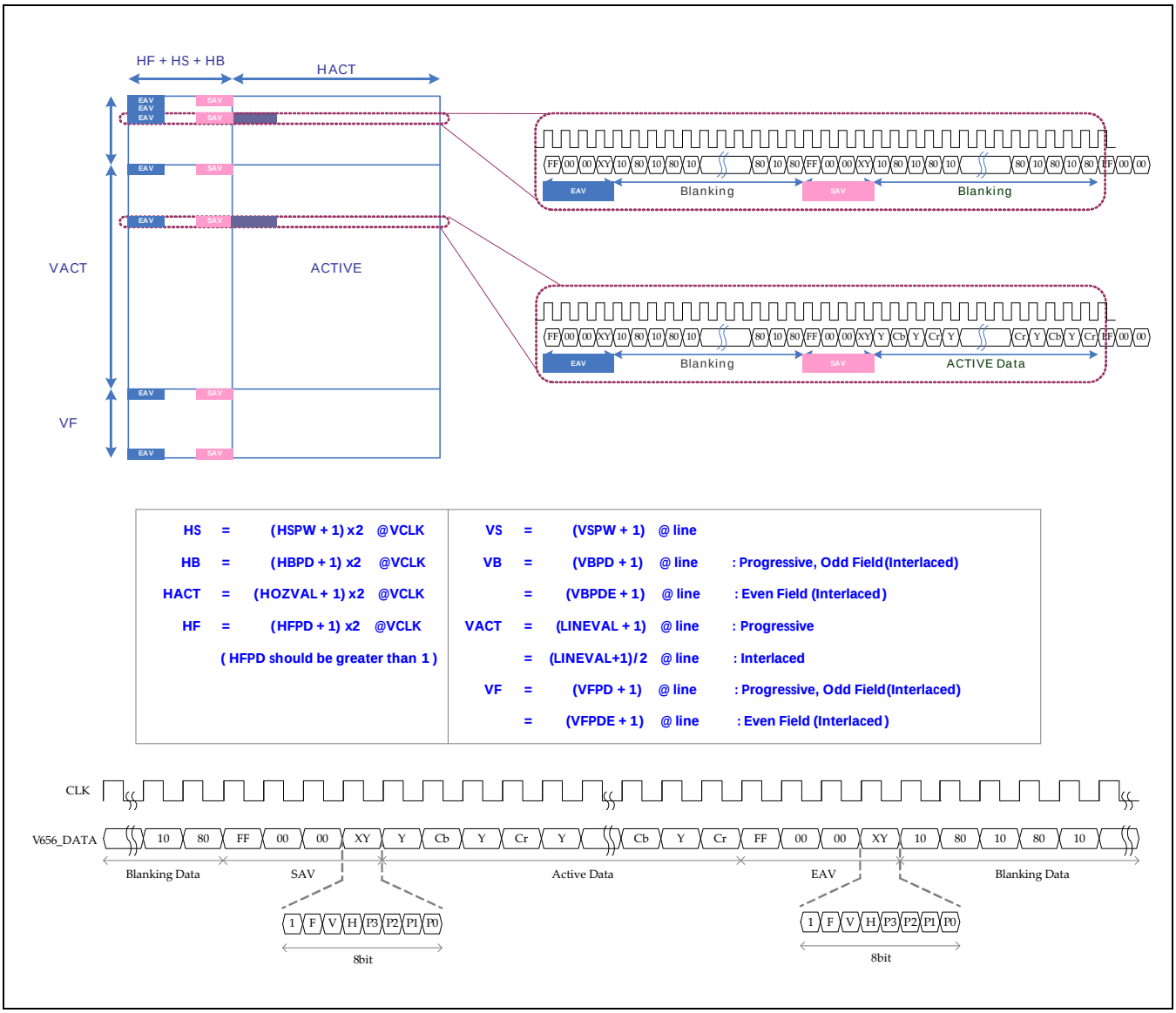


Figure 9.1-15 ITU-R BT.656 Interface Timing

bit		value
[7]	1	-
[6]	F	0 : during Field 1 1 : during Field 2
[5]	V	0 : elsewhere 1 : during field blanking
[4]	H	0 : in SAV 1 : in EAV

bit		value
[3]	P3	V ^ H
[2]	P2	F ^ H
[1]	P1	F ^ V
[0]	P0	F ^ V ^ H

Table 9.1-7 Timing Reference Code (XY Definition)

	Parallel RGB			Serial RGB		601	656
	24BPP (888)	18BPP (666)	16BPP (565)	24BPP (888)	18BPP (666)		
VD[23]	R[7]	R[5]	R[4]	D[7]	D[5]	VEN_FIELD*	
VD[22]	R[6]	R[4]	R[3]	D[6]	D[4]		V656_CLK**
VD[21]	R[5]	R[3]	R[2]	D[5]	D[3]		
VD[20]	R[4]	R[2]	R[1]	D[4]	D[2]		
VD[19]	R[3]	R[1]	R[0]	D[3]	D[1]		
VD[18]	R[2]	R[0]	-	D[2]	D[0]		
VD[17]	R[1]	-	-	D[1]	-		
VD[16]	R[0]	-	-	D[0]	-		
VD[15]	G[7]	G[5]	G[5]	-	-		
VD[14]	G[6]	G[4]	G[4]	-	-		
VD[13]	G[5]	G[3]	G[3]	-	-		
VD[12]	G[4]	G[2]	G[2]	-	-		
VD[11]	G[3]	G[1]	G[1]	-	-		
VD[10]	G[2]	G[0]	G[0]	-	-		
VD[9]	G[1]	-	-	-	-		
VD[8]	G[0]	-	-	-	-		
VD[7]	B[7]	B[5]	B[4]	-	-	VEN_DATA[7]	V656_DATA[7]
VD[6]	B[6]	B[4]	B[3]	-	-	VEN_DATA[6]	V656_DATA[6]
VD[5]	B[5]	B[3]	B[2]	-	-	VEN_DATA[5]	V656_DATA[5]
VD[4]	B[4]	B[2]	B[1]	-	-	VEN_DATA[4]	V656_DATA[4]
VD[3]	B[3]	B[1]	B[0]	-	-	VEN_DATA[3]	V656_DATA[3]
VD[2]	B[2]	B[0]	-	-	-	VEN_DATA[2]	V656_DATA[2]
VD[1]	B[1]	-	-	-		VEN_DATA[1]	V656_DATA[1]
VD[0]	B[0]	-	-	-	-	VEN_DATA[0]	V656_DATA[0]

Table 9.1-8 Parallel/Serial RGB, 601 Data Pin Map (-: Not Used)

- VEN_FIELD: VEN_FIELD is not data signal but control signal in 601 interfaces.
- V656_CLK: V656_CLK is clock in 656 interfaces.

	I80 CPU I/F (Parallel)									
	16BPP(565)	18BPP(666)		18BPP(666)		24BPP (888)		18BPP(666)	16BPP(565)	
Lx_DATA16	000	001		010		011		100	101	
		1st	2nd	1st	2nd	1st	2nd		1st	2nd
VD[23]	-	-	-	-	-	-	-	-	-	-
VD[22]	-	-	-	-	-	-	-	-	-	-
VD[21]	-	-	-	-	-	-	-	-	-	-
VD[20]	-	-	-	-	-	-	-	-	-	-
VD[19]	-	-	-	-	-	-	-	-	-	-
VD[18]	-	-	-	-	-	-	-	-	-	-
VD[17]	-	-	-	-	-	-	-	R[5]	-	-
VD[16]	-	-	-	-	-	-	-	R[4]	-	-
VD[15]	R[4]	R[5]	-	-	-	R[7]	B[7]	R[3]	-	-
VD[14]	R[3]	R[4]	-	-	-	R[6]	B[6]	R[2]	-	-
VD[13]	R[2]	R[3]	-	-	-	R[5]	B[5]	R[1]	-	-
VD[12]	R[1]	R[2]	-	-	-	R[4]	B[4]	R[0]	-	-
VD[11]	R[0]	R[1]	-	-	-	R[3]	B[3]	G[5]	-	-
VD[10]	G[5]	R[0]	-	-	-	R[2]	B[2]	G[4]	-	-
VD[9]	G[4]	G[5]	-	-	-	R[1]	B[1]	G[3]	-	-
VD[8]	G[3]	G[4]	-	R[5]	G[2]	R[0]	B[0]	G[2]	-	-
VD[7]	G[2]	G[3]	-	R[4]	G[1]	G[7]	-	G[1]	R[4]	G[2]
VD[6]	G[1]	G[2]	-	R[3]	G[0]	G[6]	-	G[0]	R[3]	G[1]
VD[5]	G[0]	G[1]	-	R[2]	B[5]	G[5]	-	B[5]	R[2]	G[0]
VD[4]	B[4]	G[0]	-	R[1]	B[4]	G[4]	-	B[4]	R[1]	B[4]
VD[3]	B[3]	B[5]	-	R[0]	B[3]	G[3]	-	B[3]	R[0]	B[3]
VD[2]	B[2]	B[4]	-	G[5]	B[2]	G[2]	-	B[2]	G[5]	B[2]
VD[1]	B[1]	B[3]	B[1]	G[4]	B[1]	G[1]	-	B[1]	G[4]	B[1]
VD[0]	B[0]	B[2]	B[0]	G[3]	B[0]	G[0]	-	B[0]	G[3]	B[0]

6 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
VIDCON0	0xEE00_0000	R/W	Video control 0 register	0x0000_0000
VIDCON1	0xEE00_0004	R/W	Video control 1 register	0x0000_0000
VIDCON2	0xEE00_0008	R/W	Video control 2 register	0x0000_0000
PRTCON	0xEE00_000C	R/W	Protect Control register	0x0000_0000
VIDTCON0	0xEE00_0010	R/W	Video time control 0 register	0x0000_0000
VIDTCON1	0xEE00_0014	R/W	Video time control 1 register	0x0000_0000
VIDTCON2	0xEE00_0018	R/W	Video time control 2 register	0x0000_0000
WINCON0	0xEE00_0020	R/W	Window control 0 register	0x0000_0000
WINCON1	0xEE00_0024	R/W	Window control 1 register	0x0000_0000
WINCON2	0xEE00_0028	R/W	Window control 2 register	0x0000_0000
WINCON3	0xEE00_002C	R/W	Window control 3 register	0x0000_0000
WINCON4	0xEE00_0030	R/W	Window control 4 register	0x0000_0000
VIDOSD0A	0xEE00_0040	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0B	0xEE00_0044	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0C	0xEE00_0048	R/W	Video Window 0's size control register	0x0000_0000
VIDOSD1A	0xEE00_0050	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1B	0xEE00_0054	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1C	0xEE00_0058	R/W	Video Window 1's alpha control register	0x0000_0000
VIDOSD1D	0xEE00_005C	R/W	Video Window 1's size control register	0x0000_0000
VIDOSD2A	0xEE00_0060	R/W	Video Window 2's position control register	0x0000_0000
VIDOSD2B	0xEE00_0064	R/W	Video Window 2's position control register	0x0000_0000
VIDOSD2C	0xEE00_0068	R/W	Video Window 2's alpha control register	0x0000_0000
VIDOSD2D	0xEE00_006c	R/W	Video Window 2's size control register	0x0000_0000
VIDOSD3A	0xEE00_0070	R/W	Video Window 3's position control register	0x0000_0000
VIDOSD3B	0xEE00_0074	R/W	Video Window 3's position control register	0x0000_0000
VIDOSD3C	0xEE00_0078	R/W	Video Window 3's alpha control register	0x0000_0000
VIDOSD4A	0xEE00_0080	R/W	Video Window 4's position control register	0x0000_0000
VIDOSD4B	0xEE00_0084	R/W	Video Window 4's position control register	0x0000_0000
VIDOSD4C	0xEE00_0088	R/W	Video Window 4's alpha control register	0x0000_0000
VIDW00ADD0B0	0xEE00_00A0	R/W	Window 0's buffer start address register, buffer 0	0x0000_0000
VIDW00ADD0B1	0xEE00_00A4	R/W	Window 0's buffer start address register, buffer 1	0x0000_0000
VIDW01ADD0B0	0xEE00_00A8	R/W	Window 1's buffer start address register, buffer 0	0x0000_0000

Register	Address	R/W	Description	Reset Value
VIDW01ADD0B1	0xEE00_00AC	R/W	Window 1's buffer start address register, buffer 1	0x0000_0000
VIDW02ADD0	0xEE00_00B0	R/W	Window 2's buffer start address register	0x0000_0000
VIDW03ADD0	0xEE00_00B8	R/W	Window 3's buffer start address register	0x0000_0000
VIDW04ADD0	0xEE00_00C0	R/W	Window 4's buffer start address register	0x0000_0000
VIDW00ADD1B0	0xEE00_00D0	R/W	Window 0's buffer end address register, buffer 0	0x0000_0000
VIDW00ADD1B1	0xEE00_00D4	R/W	Window 0's buffer end address register, buffer 1	0x0000_0000
VIDW01ADD1B0	0xEE00_00D8	R/W	Window 1's buffer end address register, buffer 0	0x0000_0000
VIDW01ADD1B1	0xEE00_00DC	R/W	Window 1's buffer end address register, buffer 1	0x0000_0000
VIDW02ADD1	0xEE00_00E0	R/W	Window 2's buffer end address register	0x0000_0000
VIDW03ADD1	0xEE00_00E8	R/W	Window 3's buffer end address register	0x0000_0000
VIDW04ADD1	0xEE00_00F0	R/W	Window 4's buffer end address register	0x0000_0000
VIDW00ADD2	0xEE00_0100	R/W	Window 0's buffer size register	0x0000_0000
VIDW01ADD2	0xEE00_0104	R/W	Window 1's buffer size register	0x0000_0000
VIDW02ADD2	0xEE00_0108	R/W	Window 2's buffer size register	0x0000_0000
VIDW03ADD2	0xEE00_010C	R/W	Window 3's buffer size register	0x0000_0000
VIDW04ADD2	0xEE00_0110	R/W	Window 4's buffer size register	0x0000_0000
VP1TCON0	0xEE00_0118	R/W	VP 1 interface timing control 0 register	0x0000_0000
VP1TCON1	0xEE00_011c	R/W	VP 1 interface timing control 1 register	0x0000_0000
VIDINTCON0	0xEE00_0130	R/W	Indicate the Video interrupt control register	0x0000_0000
VIDINTCON1	0xEE00_0134	R/W	Video Interrupt Pending register	0x0000_0000
W1KEYCON0	0xEE00_0140	R/W	Color key control register	0x0000_0000
W1KEYCON1	0xEE00_0144	R/W	Color key value (transparent value) register	0x0000_0000
W2KEYCON0	0xEE00_0148	R/W	Color key control register	0x0000_0000
W2KEYCON1	0xEE00_014C	R/W	Color key value (transparent value) register	0x0000_0000
W3KEYCON0	0xEE00_0150	R/W	Color key control register	0x0000_0000
W3KEYCON1	0xEE00_0154	R/W	Color key value (transparent value) register	0x0000_0000
W4KEYCON0	0xEE00_0158	R/W	Color key control register	0x0000_0000
W4KEYCON1	0xEE00_015C	R/W	Color key value (transparent value) register	0x0000_0000
DITHMODE	0xEE00_0170	R/W	Dithering mode register.	0x0000_0000
WIN0MAP	0xEE00_0180	R/W	Window color control	0x0000_0000
WIN1MAP	0xEE00_0184	R/W	Window color control	0x0000_0000
WIN2MAP	0xEE00_0188	R/W	Window color control	0x0000_0000
WIN3MAP	0xEE00_018C	R/W	Window color control	0x0000_0000
WIN4MAP	0xEE00_0190	R/W	Window color control	0x0000_0000
WPALCON_H	0xEE00_019c	R/W	Window Palette control register	0x0000_0000
WPALCON_L	0xEE00_01A0	R/W	Window Palette control register	0x0000_0000

Register	Address	R/W	Description	Reset Value
TRIGCON	0xEE00_01A4	R/W	I80 / RGB Trigger Control Register	0x0000_0000
ITUIFCON	0xEE00_01A8	R/W	ITU-R (BT.601/656) Interface Control Register	0x0000_0000
I80IFCONA0	0xEE00_01B0	R/W	I80 Interface control 0 for Main LDI	0x0000_0000
I80IFCONA1	0xEE00_01B4	R/W	I80 Interface control 0 for Sub LDI	0x0000_0000
I80IFCONB0	0xEE00_01B8	R/W	I80 Interface control 1 for Main LDI	0x0000_0000
I80IFCONB1	0xEE00_01BC	R/W	I80 Interface control 1 for Sub LDI	0x0000_0000
LDI_CMDCON0	0xEE00_01D0	R/W	I80 Interface LDI Command Control 0	0x0000_0000
LDI_CMDCON1	0xEE00_01D4	R/W	I80 Interface LDI Command Control 1	0x0000_0000
SIFCCON0	0xEE00_01E0	R/W	LCD i80 System Interface Command Control 0	0x0000_0000
SIFCCON1	0xEE00_01E4	R/W	LCD i80 System Interface Command Control 1	0x0000_0000
SIFCCON2	0xEE00_01E8	R	LCD i80 System Interface Command Control 2	0x0000_0000
VIDW0ALPHA0	0xEE00_0200	R/W	Window 0's alpha value 0 register	0x0000_0000
VIDW0ALPHA1	0xEE00_0204	R/W	Window 0's alpha value 1 register	0x0000_0000
VIDW1ALPHA0	0xEE00_0208	R/W	Window 1's alpha value 0 register	0x0000_0000
VIDW1ALPHA1	0xEE00_020c	R/W	Window 1's alpha value 1 register	0x0000_0000
VIDW2ALPHA0	0xEE00_0210	R/W	Window 2's alpha value 0 register	0x0000_0000
VIDW2ALPHA1	0xEE00_0214	R/W	Window 2's alpha value 1 register	0x0000_0000
VIDW3ALPHA0	0xEE00_0218	R/W	Window 3's alpha value 0 register	0x0000_0000
VIDW3ALPHA1	0xEE00_021c	R/W	Window 3's alpha value 1 register	0x0000_0000
VIDW4ALPHA0	0xEE00_0220	R/W	Window 4's alpha value 0 register	0x0000_0000
VIDW4ALPHA1	0xEE00_0224	R/W	Window 4's alpha value 1 register	0x0000_0000
BLENDEQ1	0xEE00_0244	R/W	Window 1's blending equation Control register	0x0000_00c2
BLENDEQ2	0xEE00_0248	R/W	Window 2's blending equation Control register	0x0000_00c2
BLENDEQ3	0xEE00_024c	R/W	Window 3's blending equation Control register	0x0000_00c2
BLENDEQ4	0xEE00_0250	R/W	Window 4's blending equation Control register	0x0000_00c2
BLENDCON	0xEE00_0260	R/W	Blending Control register	0x0000_0000
LDI_CMD0	0xEE00_0280	R/W	I80 Interface LDI Command 0	0x0000_0000
LDI_CMD1	0xEE00_0284	R/W	I80 Interface LDI Command 1	0x0000_0000
LDI_CMD2	0xEE00_0288	R/W	I80 Interface LDI Command 2	0x0000_0000
LDI_CMD3	0xEE00_028C	R/W	I80 Interface LDI Command 3	0x0000_0000
LDI_CMD4	0xEE00_0290	R/W	I80 Interface LDI Command 4	0x0000_0000
LDI_CMD5	0xEE00_0294	R/W	I80 Interface LDI Command 5	0x0000_0000

Register	Address	R/W	Description	Reset Value
LDI_CMD6	0xEE00_0298	R/W	I80 Interface LDI Command 6	0x0000_0000
LDI_CMD7	0xEE00_029C	R/W	I80 Interface LDI Command 7	0x0000_0000
LDI_CMD8	0xEE00_02A0	R/W	I80 Interface LDI Command 8	0x0000_0000
LDI_CMD9	0xEE00_02A4	R/W	I80 Interface LDI Command 9	0x0000_0000
LDI_CMD10	0xEE00_02A8	R/W	I80 Interface LDI Command 10	0x0000_0000
LDI_CMD11	0xEE00_02AC	R/W	I80 Interface LDI Command 11	0x0000_0000

PalRam (Palette Memory)

	Start address	End address	R/W	Description	Reset value
Win0 PalRam	0xEE00_2400 (0xEE00_0400)	0xEE00_27FC (0xEE00_07FC)	R/W	0~255 entry palette data	Undefined
Win1 PalRam	0xEE00_2800 (0xEE00_0800)	0xEE00_2BFC (0xEE00_0BFC)	R/W	0~255 entry palette data	Undefined
Win2 PalRam	0xEE00_2C00	0xEE00_2FFC	R/W	0~255 entry palette data	Undefined
Win3 PalRam	0xEE00_3000	0xEE00_33FC	R/W	0~255 entry palette data	Undefined
Win4 PalRam	0xEE00_3400	0xEE00_37FC	R/W	0~255 entry palette data	Undefined

6.1 VIDEO MAIN CONTROL 0 REGISTER (VIDCON0, R/W, ADDRESS = 0XEE00_0000)

VIDCON0	Bit	Description	Reset Value
Reserved	[31]	Reserved (should be 0)	0
DSI_EN	[30]	MIPI DSI enable 0 = Disables 1 = Enables (i80 24bit data interface, SYS_ADD[1])	0
INTERLACE_F	[29]	Interlace or Progressive 0 = Progressive 1 = Interlace (only ITU601/656 Interface)	0
Reserved	[28]	Reserved (Should be zero)	0
VIDOUT	[27:26]	It determines the output format of Video Controller 000: RGB I/F 001 = ITU601/656 010 = Indirect I80 I/F for LDI0 011 = Indirect I80 I/F for LDI1	000
L1_DATA16	[25:23]	Selects output data format mode of Indirect I80 I/F (LDI1.) (If, VIDOUT[1:0] == 2'b11) 000 = 16 bit mode (16 bpp) 001 = 16 + 2-bit mode (18 bpp) 010 = 9 + 9-bit mode (18 bpp) 011 = 16 + 8-bit mode (24 bpp) 100 = 18-bit mode (18bpp) 101 = 8 + 8-bit mode (16bpp)	000
L0_DATA16	[22:20]	Selects output data format mode of Indirect I80 I/F (LDI0.) (If, VIDOUT[1:0] == 2'b10) 000 = 16-bit mode (16 bpp) 001 = 16 + 2-bit mode (18 bpp) 010 = 9 + 9-bit mode (18 bpp) 011 = 16 + 8-bit mode (24 bpp) 100 = 18 bit-mode (18bpp) 101 = 8 + 8-bit mode (16bpp)	000
Reserved	[19]	Reserved (should be 0)	0
PNRMODE	[18:17]	Selects the display mode (Where, VIDOUT[1:0] == 2'b00). 00 = RGB Parallel format (RGB) 01 = RGB Parallel format (BGR) 10 = Serial Format (R->G->B) 11 = Serial Format (B->G->R) Select the display mode (Where, VIDOUT[1:0] != 2'b00) 00 = RGB Parallel format (RGB)	00
CLKVALUP	[16]	Selects CLKVAL_F update timing control 0 = Always	0

VIDCON0	Bit	Description	Reset Value
		1 = Start of a frame (only once per frame)	
Reserved	[15:14]	Reserved	
CLKVAL_F	[13:6]	Determines the rates of VCLK and CLKVAL[7:0] VCLK = HCLK / (CLKVAL+1) where CLKVAL >= 1 **Note.** 1. The maximum frequency of VCLK is 66MHz. 2. CLKSEL_F register selects Video Clock Source	0
VCLKFREE	[5]	VCLK Free run control (Only valid at the RGB IF mode) 0 = Normal mode (control by ENVID) 1 = Free-run mode	0
CLKDIR	[4]	Selects the clock source as direct or divide using CLKVAL_F register 0 = Direct clock (frequency of VCLK = frequency of Clock source) 1 = Divided by CLKVAL_F	0x00
Reserved	[3]	Should be 0	0x0
CLKSEL_F	[2]	Selects the Video Clock source 0 = HCLK 1 = SCLK_LCD	0
ENVID	[1]	Video output and the logic immediately enable/ disable. 0 = Disables the video output and the Display control signal. 1 = Enables the video output and the Display control signal.	0
ENVID_F	[0]	Video output and the logic enable/disable at current frame end. 0 = Disables the video output and the Display control signal. 1 = Enables the video output and the Display control signal. * If this bit is set on and off, then you will read "H" and video controller enable until the end of current frame.	0

NOTE: Display On: ENVID & ENVID_F set to "1"
Direct Off: ENVID & ENVID_F set to "0" simultaneously
Per Frame Off: ENVID_F set "0" & ENVID set"1"

Caution 1: If the VIDCON0 is setting for Per Frame off in interlace mode, the value of INTERLACE_F should be set to "0" in the same time.

Caution 2: If display controller is off using direct off, it is impossible to turn on the display controller without reset.

6.2 VIDEO MAIN CONTROL 1 REGISTER (VIDCON1, R/W, ADDRESS = 0XEE00_ 0004)

VIDCON1	Bit	Description	Reset Value
LINECNT (read only)	[26:16]	Provide the status of the line counter (read only) Up count from 0 to LINEVAL	0
FSTATUS	[15]	Field Status (read only). 0 = ODD Field 1 = EVEN Field	0
VSTATUS	[14:13]	Vertical Status (read only). 00 = VSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
Reserved	[12:8]	Reserved	
IVCLK	[7]	This bit controls the polarity of the VCLK active edge. 0 = Video data is fetched at VCLK falling edge 1 = Video data is fetched at VCLK rising edge	0
IHSYNC	[6]	This bit indicates the HSYNC pulse polarity. 0 = Normal 1 = Inverted	0
IVSYNC	[5]	This bit indicates the VSYNC pulse polarity. 0 = Normal 1 = Inverted	0
IVDEN	[4]	This bit indicates the VDEN signal polarity. 0 = Normal 1 = Inverted	0
Reserved	[3:0]	Reserved	0x0

6.3 VIDEO MAIN CONTROL 2 REGISTER (VIDCON2, R/W, ADDRESS = 0XEE00_ 0008)

VIDCON2	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
EN601	[23]	Control ITU601 output enable 0 = Disables 1 = Enables	0
EN656	[22]	Control ITU656 output enable. 0 = Disables 1 = Enables	0
Reserved	[21:15]	Reserved	0
Reserved	[14]	Reserved ***Note*** *:This bit should be 1*	0
TVFORMATSEL	[13:12]	This bit indicates output format of YUV data. 00 = RGB 01 = YUV422 1x = YUV444	0
Reserved	[11:9]	Reserved	0
OrgYCbCr	[8]	This bit indicates order of YUV data. 0 = Y - CbCr 1 = CbCr – Y	0
YUVOrd	[7]	This bit indicates order of Chroma data. 0 = Cb - Cr 1 = Cr – Cb	0
Reserved	[6:0]	Reserved	0

6.4 PROTECT CONTROL REGISTER(PRTCON, R/W, ADDRESS = 0XEE00_000C)

PRTCON	Bit	Description	Reset Value
Reserved	[31:12]	Reserved (Should be 0)	0
DontAddLd	[11]	Protect to update Window's buffer shadow register (xxx_F) 0 = update shadow register per frame 1 = protect to update (update shadow register at next frame after 'DontAddLd' turns to be 1'b0) Note. Wx_VBANK_F, Wx_VBASEU_F, Wx_VBASEL_F, Wx_PAGEWIDTH_F, Wx_OFFSIZE_F	0
Reserved	[10:0]	Reserved (Should be 0)	0

6.5 VIDEO TIME CONTROL 0 REGISTER (VIDTCON0, R/W, ADDRESS = 0XEE00_ 0010)

VIDTCON0	Bit	Description	Reset Value
VBPDE	[31:24]	Vertical back porch is the number of inactive lines at the start of a frame, after vertical synchronization period. (Only for the even field of YVU interface)	0x00
VBPD	[23:16]	Vertical back porch is the number of inactive lines at the start of a frame, after vertical synchronization period.	0x00
VFPD	[15:8]	Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period.	0x00
VSPW	[7:0]	Vertical sync pulse width determines the VSYNC pulse's high level width by counting the number of inactive lines.	0x00

6.6 VIDEO TIME CONTROL 1 REGISTER (VIDTCON1, R/W, ADDRESS = 0XEE00_ 0014)

VIDTCON1	Bit	Description	Reset Value
VFPDE	[31:24]	Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period. (Only for the even field of YVU interface)	0
HBPD	[23:16]	Horizontal back porch is the number of VCLK periods between the falling edge of HSYNC and the start of active data.	0x00
HFPD	[15:8]	Horizontal front porch is the number of VCLK periods between the end of active data and the rising edge of HSYNC.	0x00
HSPW	[7:0]	Horizontal sync pulse width determines the HSYNC pulse's high level width by counting the number of the VCLK.	0x00

6.7 VIDEO TIME CONTROL 2 REGISTER (VIDTCON2, R/W, ADDRESS = 0XEE00_ 0018)

VIDTCON2	Bit	Description	Reset Value
LINEVAL	[21:11]	These bits determine the vertical size of display In Interlace mode, (LINEVAL + 1) should have even value.	0
HOZVAL	[10:0]	These bits determine the horizontal size of display	0

NOTE: HOZVAL = (Horizontal display size) -1, LINEVAL = (Vertical display size) –1

6.8 WINDOW 0 CONTROL REGISTER (WINCON0, R/W, ADDRESS = 0XEE00_ 0020)

WINCON0	Bit	Description	Reset Value
nWide/Narrow	[27:26]	Choose color space conversion equation from YCbCr to RGB according to input value range. 2'00 for YCbCr Wide range and 2'11 for YCbCr Narrow range - Wide Range : Y/Cb/Cr: 255-0 - Narrow Range : Y: 235-16 , Cb/Cr: 240-16	00
TRGSTATUS	[25]	Window 0 SW Trigger Update Status [Read Only] 0 = Update 1 = Not Update When window 0 SW trigger happens, this bit is automatically set to '1'. Only after updating shadow register sets is this value cleared.	0
Reserved	[24:23]	Reserved	00
ENLOCAL	[22]	Select Data access method. 0 = Dedicated DMA 1 = Local Path (CAMIF 0 FIFO Out) This register must be disabled at the ENWIN_F disable state	0
BUFSTATUS	[21]	Buffer Status (Read Only) 0 = buffer set 0 1 = buffer set 1	0
BUFSEL	[20]	Select Buffer set (0/1) 0 = buffer set 0 1 = buffer set 1	0
BUFAUTOEN	[19]	Double Buffer Auto control bit 0 = Fixed by BUFSEL, 1 = Auto changed by Trigger Input	0
BITSWP	[18]	Bit swap control bit. 0 = Disables Swap 1 = Enables Swap	0
BYTSWP	[17]	Byte swaps control bit. 0 = Disables Swap 1 = Enables Swap	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
WSWP	[15]	Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
Reserved	[14]	Reserved	0
InRGB	[13]	It indicates the input color space of source image. (Only for 'EnLcal' enable) 0 = RGB 1 = YCbCr	0
Reserved	[12:11]	Reserved (should be 00)	0
BURSTLEN	[10:9]	DMA's Burst Maximum Length selection: 00 = 16 word– burst 01 = 8 word– burst 10 = 4 word– burst	0
Reserved	[8:7]	Reserved	0
BLD_PIX	[6]	Selects blending category (In case of window0, this is only required for window 0's blending factor decision) 0 = Per plane blending	0

WINCON0	Bit	Description	Reset Value
		1 = Per pixel blending	
BPPMODE_F	[5:2]	Selects the BPP (Bits Per Pixel) mode Window image. 0000 = 1 bpp 0001 = 2 bpp 0010 = 4 bpp 0011 = 8 bpp (palletized) 0100 = 8 bpp (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 bpp (non-palletized, R:5-G:6-B:5) 0110 = 16 bpp (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = unpacked 18 bpp (non-palletized, R:6-G:6-B:6) 1001 = unpacked 18 bpp (non-palletized, A:1-R:6-G:6-B:5) 1010 = unpacked 19 bpp (non-palletized, A:1-R:6-G:6-B:6) 1011 = unpacked 24 bpp (non-palletized R:8-G:8-B:8) 1100 = unpacked 24 bpp (non-palletized A:1-R:8-G:8-B:7) *1101 = unpacked 25 bpp (non-palletized A:1-R:8-G:8-B:8) *1110 = unpacked 13 bpp (non-palletized A:1-R:4-G:4-B:4) 1111 = unpacked 15 bpp (non-palletized R:5-G:5-B:5) *Note. *1101 = supports unpacked 32 bpp (non-palletized A:8-R:8-G:8-B:8) for per pixel blending.* *1110 = support 16 bpp (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	0
ALPHA_SEL	[1]	Select Alpha value by If Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values If Per pixel blending (BLD_PIX ==1) 0 = selected by AEN (A value) 1 = using DATA[31:24] data in word boundary *(only when BPPMODE_F = 4'b1101)* DATA[31:28], [15:12] data in word boundary (*only when BPPMODE_F = 4'b1110)*	0
ENWIN_F	[0]	Video output and the logic immediately enable/disable. 0 = Disables the video output and the VIDEO control signal. 1 = Enables the video output and the VIDEO control signal.	0

6.9 WINDOW 1 CONTROL REGISTER (WINCON1, R/W, ADDRESS = 0XEE00_ 0024)

WINCON1	Bit	Description	Reset Value
nWide/Narrow	[27:26]	Choose color space conversion equation from YCbCr to RGB according to input value range. 2'00 for YCbCr Wide range and 2'11 for YCbCr Narrow range - Wide Range : Y/Cb/Cr: 255-0 - Narrow Range : Y: 235-16 , Cb/Cr: 240-16	00
TRGSTATUS	[25]	Window 0 SW Trigger Update Status [Read Only] 0 = Update 1 = Not Update When window 1 SW trigger happens, this bit is automatically set to '1'. Only after updating shadow register sets is this value cleared.	0
VP_EN	[24]	VP interface enable 0 = Disables 1 = Enables	0
LOCALSEL	[23]	Select Local Path Source 0 = CAMIF1 (FIFO Out) 1 = VP **Note**: If the CAMIF1 is selected, there is a constraint that the source of CAMIF1 must be memory not external camera. MS-DMA transfers from memory to CAMIF1.	0
ENLOCAL	[22]	Select Data access method. 0 = Dedicated DMA 1 = Local Path **Note:** This register must be disabled at the ENWIN_F disable state	0
BUFSTATUS	[21]	Buffer Status (Read Only) 0 = buffer set 0 1 = buffer set 1	0
BUFSEL	[20]	Select Buffer set (0/1) 0 = buffer set 0 1 = buffer set 1	0
BUFAUTOEN	[19]	Double Buffer Auto control bit 0 = Fixed by BUFSEL, 1 = Auto changed by Trigger Input	0
BITSWP	[18]	Bit swap control bit. 0 = Disables Swap 1 = Enables Swap	0
BYTSWP	[17]	Byte swaps control bit. 0 = Disables Swap 1 = Enables Swap	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
WSWP	[15]	Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
Reserved	[14]	Reserved	0
InRGB	[13]	It indicates the input color space of source image. (Only for 'EnLcal' enable) 0 = RGB 1 = YCbCr	0
Reserved	[12:11]	Reserved (should be 00)	0
BURSTLEN	[10:9]	DMA's Burst Maximum Length selection : 00 = 16 word– burst	0

WINCON1	Bit	Description	Reset Value
		01 = 8 word– burst 10 = 4 word– burst	
Reserved	[8]	Reserved (should be 0)	0
ALPHA_MUL	[7]	Multiplied Alpha value mode 0 = Disables 1 = Multiplied mode If ALPHA_MUL is 1, you should set BLD_PIX =1 , ALPHA_SEL=1 and (BPPMODE_F[5:2] = 4'b1101 or 4'b1110) **Note.** i) Alpha value = alpha_pixel(from data) * ALPHA0_R/G/B ii) In this mode, you can select two cases @BLENDEQ1 - A_FUNC = alphaA, B_FUNC = 1 – alphaA - A_FUNC = alphaA, B_FUNC = max (pre-multiplied)	0
BLD_PIX	[6]	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	Select the BPP (Bits Per Pixel) mode Window image. 0000 = 1 bpp 0001 = 2 bpp 0010 = 4 bpp 0011 = 8 bpp (palletized) 0100 = 8 bpp (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 bpp (non-palletized, R:5-G:6-B:5) 0110 = 16 bpp (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = unpacked 18 bpp (non-palletized, R:6-G:6-B:6) 1001 = unpacked 18 bpp (non-palletized, A:1-R:6-G:6-B:5) 1010 = unpacked 19 bpp (non-palletized, A:1-R:6-G:6-B:6) 1011 = unpacked 24 bpp (non-palletized R:8-G:8-B:8) 1100 = unpacked 24 bpp (non-palletized A:1-R:8-G:8-B:7) *1101 = unpacked 25 bpp (non-palletized A:1-R:8-G:8-B:8) *1110 = unpacked 13 bpp (non-palletized A:1-R:4-G:4-B:4) 1111 = unpacked 15 bpp (non-palletized R:5-G:5-B:5) *Note. *1101 = supports unpacked 32 bpp (non-palletized A:8-R:8-G:8-B:8) for per pixel blending.* *1110 = support 16 bpp (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	0
ALPHA_SEL	[1]	Select Alpha value by If Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values If Per pixel blending (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[31:24] data in word boundary *(only when BPPMODE_F = 4'b1101)* DATA[31:28], [15:12] data in word boundary (*only when*	0

WINCON1	Bit	Description	Reset Value
		BPPMODE_F = 4'b1110)	
ENWIN_F	[0]	Video output and the logic immediately enable/disable. 0 = Disables the video output and the VIDEO control signal. 1 = Enables the video output and the VIDEO control signal.	0

6.10 WINDOW 2 CONTROL REGISTER (WINCON2, R/W, ADDRESS = 0XEE00_ 0028)

WINCON2	Bit	Description	Reset Value
nWide/Narrow	[27:26]	Choose color space conversion equation from YCbCr to RGB according to input value range. 2'00 for YCbCr Wide range and 2'11 for YCbCr Narrow range - Wide Range : Y/Cb/Cr: 255-0 - Narrow Range : Y: 235-16 , Cb/Cr: 240-16	00
Reserved	[25:24]	Reserved	00
Reserved	[23]	Should be '0'	0
ENLOCAL	[22]	Selects Data access method 0 = Dedicated DMA 1 = Local Path (CAMIF2 FIFO Out) This register must be disabled at the ENWIN_F disable state	0
Reserved	[21:19]	Reserved	
BITSWP	[18]	Bit swap control bit. 0 = Disables Swap 1 = Enables Swap	0
BYTSWP	[17]	Byte swaps control bit. 0 = Disables Swap 1 = Enables Swap	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
WSWP	[15]	Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
Reserved	[14]	Reserved	0
InRGB	[13]	It indicates the input color space of source image. (Only for 'EnLcal' enable) 0 = RGB 1 = YCbCr	0
Reserved	[12:11]	Reserved (should be 00)	0
BURSTLEN	[10:9]	DMA's Burst Maximum Length selection : 00 = 16 word– burst 01 = 8 word– burst 10 = 4 word– burst	0
Reserved	[8]	Reserved (should be 0)	0
ALPHA_MUL	[7]	Multiplied Alpha value mode 0 = Disables 1 = Multiplied mode If ALPHA_MUL is 1, you should set BLD_PIX =1 , ALPHA_SEL=1 and (BPPMODE_F[5:2] = 4'b1101 or 4'b1110)	0

WINCON2	Bit	Description	Reset Value
		Note. i) Alpha value = alpha_pixel(from data) * ALPHA0_R/G/B ii) In this mode, you can only select two cases @BLENDEQ2 - A_FUNC = alphaA, B_FUNC = 1 – alphaA - A_FUNC = alphaA, B_FUNC = max (pre-multiplied)	
BLD_PIX	[6]	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	Select the BPP (Bits Per Pixel) mode Window image. 0000 = 1 bpp 0001 = 2 bpp 0010 = 4 bpp 0011 = 8 bpp (palletized) 0100 = 8 bpp (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 bpp (non-palletized, R:5-G:6-B:5) 0110 = 16 bpp (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = unpacked 18 bpp (non-palletized, R:6-G:6-B:6) 1001 = unpacked 18 bpp (non-palletized, A:1-R:6-G:6-B:5) 1010 = unpacked 19 bpp (non-palletized, A:1-R:6-G:6-B:6) 1011 = unpacked 24 bpp (non-palletized R:8-G:8-B:8) 1100 = unpacked 24 bpp (non-palletized A:1-R:8-G:8-B:7) *1101 = unpacked 25 bpp (non-palletized A:1-R:8-G:8-B:8) *1110 = unpacked 13 bpp (non-palletized A:1-R:4-G:4-B:4) 1111 = unpacked 15 bpp (non-palletized R:5-G:5-B:5) *Note. *1101 = supports unpacked 32 bpp (non-palletized A:8-R:8-G:8-B:8) for per pixel blending.* *1110 = support 16 bpp (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	0
ALPHA_SEL	[1]	Select Alpha value by If Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values If Per pixel blending (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[31:24] data in word boundary *(only when BPPMODE_F = 4'b1101)* DATA[31:28], [15:12] data in word boundary (*only when BPPMODE_F = 4'b1110)*	0
ENWIN_F	[0]	Video output and the logic immediately enable/disable. 0 = Disable the video output and the VIDEO control signal. 1 = Enable the video output and the VIDEO control signal.	0

6.11 WINDOW 3 CONTROL REGISTER (WINCON3, R/W, ADDRESS = 0XEE00_ 002C)

WINCON3	Bit	Description	Reset Value
BITSWP	[18]	Bit swap control bit. 0 = Disables Swap 1 = Enables Swap	0
BYTSWP	[17]	Bytes swap control bit. 0 = Disables Swap 1 = Enables Swap	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
WSWP	[15]	Word swap control bit. 0 = Disables Swap 1 = Enables Swap	
Reserved	[14:11]	Reserved (should be 00)	0
BURSTLEN	[10:9]	DMA's Burst Maximum Length selection: 00 = 16 word– burst 01 = 8 word– burst 10 = 4 word– burst	0
Reserved	[8]	Reserved **(should be 0)**	0
ALPHA_MUL	[7]	Multiplied Alpha value mode 0 = Disables 1 = Multiplied mode If ALPHA_MUL is 1, you should set BLD_PIX =1 , ALPHA_SEL=1 and (BPPMODE_F[5:2] = 4'b1101 or 4'b1110) **Note.** i) Alpha value = alpha_pixel(from data) * ALPHA0_R/G/B ii) In this mode, you can only select two cases @BLENDEQ3 - A_FUNC = alphaA, B_FUNC = 1 – alphaA - A_FUNC = alphaA, B_FUNC = max (pre-multiplied)	0
BLD_PIX	[6]	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	Select the BPP (Bits Per Pixel) mode Window image. 0000 = 1 bpp 0001 = 2 bpp 0010 = 4 bpp 0011 = 8 bpp (palletized) 0100 = 8 bpp (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 bpp (non-palletized, R:5-G:6-B:5) 0110 = 16 bpp (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = unpacked 18 bpp (non-palletized, R:6-G:6-B:6) 1001 = unpacked 18 bpp (non-palletized, A:1-R:6-G:6-B:5) 1010 = unpacked 19 bpp (non-palletized, A:1-R:6-G:6-B:6) 1011 = unpacked 24 bpp (non-palletized R:8-G:8-B:8) 1100 = unpacked 24 bpp (non-palletized A:1-R:8-G:8-B:7) *1101 = unpacked 25 bpp (non-palletized A:1-R:8-G:8-B:8)	0

WINCON3	Bit	Description	Reset Value
		*1110 = unpacked 13 bpp (non-palletized A:1-R:4-G:4-B:4) 1111 = unpacked 15 bpp (non-palletized R:5-G:5-B:5) *Note. *1101 = support unpacked 32 bpp (non-palletized A:8-R:8-G:8-B:8) for per pixel blending.* *1110 = support 16 bpp (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	
ALPHA_SEL	[1]	Select Alpha value by If Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values If Per pixel blending (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[31:24] data in word boundary *(only when BPPMODE_F = 4'b1101)* DATA[31:28], [15:12] data in word boundary (*only when BPPMODE_F = 4'b1110)*	0
ENWIN_F	[0]	Video output and the logic immediately enable/disable. 0 = Disables the video output and the VIDEO control signal. 1 = Enables the video output and the VIDEO control signal.	0

6.12 WINDOW 4 CONTROL REGISTER (WINCON4, R/W, ADDRESS = 0XEE00_ 0030)

WINCON4	Bit	Description	Reset Value
BITSWP	[18]	Bit swap control bit. 0 = Disables Swap 1 = Enables Swap	0
BYTSWP	[17]	Byte swaps control bit. 0 = Disables Swap 1 = Enables Swap	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Disables Swap 1 = Enables Swap	0
WSWP	[15]	Word swap control bit. 0 = Disables Swap 1 = Enables Swap	
Reserved	[14:11]	Reserved (should be 00)	0
BURSTLEN	[10:9]	DMA's Burst Maximum Length selection : 00 = 16 word– burst 01 = 8 word– burst 10 = 4 word– burst	0
Reserved	[8]	Reserved **(should be 0)**	0
ALPHA_MUL	[7]	Multiplied Alpha value mode 0 = Disables 1 = multiplied mode If ALPHA_MUL is 1, you should set BLD_PIX =1 , ALPHA_SEL=1 and (BPPMODE_F[5:2] = 4'b1101 or 4'b1110) **Note**. i) Alpha value = alpha_pixel(from data) * ALPHA0_R/G/B ii) In this mode, you can only select two cases @BLENDEQ4 - A_FUNC = alphaA, B_FUNC = 1 – alphaA - A_FUNC = alphaA, B_FUNC = max (pre-multiplied)	0
BLD_PIX	[6]	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
BPPMODE_F	[5:2]	Select the BPP (Bits Per Pixel) mode Window image. 0000 = 1 bpp 0001 = 2 bpp 0010 = 4 bpp 0011 = 8 bpp (palletized) 0100 = 8 bpp (non-palletized, A: 1-R:2-G:3-B:2) 0101 = 16 bpp (non-palletized, R:5-G:6-B:5) 0110 = 16 bpp (non-palletized, A:1-R:5-G:5-B:5) 0111 = 16 bpp (non-palletized, I :1-R:5-G:5-B:5) 1000 = unpacked 18 bpp (non-palletized, R:6-G:6-B:6) 1001 = unpacked 18 bpp (non-palletized, A:1-R:6-G:6-B:5) 1010 = unpacked 19 bpp (non-palletized, A:1-R:6-G:6-B:6) 1011 = unpacked 24 bpp (non-palletized R:8-G:8-B:8) 1100 = unpacked 24 bpp (non-palletized A:1-R:8-G:8-B:7) *1101 = unpacked 25 bpp (non-palletized A:1-R:8-G:8-B:8)	0

WINCON4	Bit	Description	Reset Value
		*1110 = unpacked 13 bpp (non-palletized A:1-R:4-G:4-B:4) 1111 = unpacked 15 bpp (non-palletized R:5-G:5-B:5) Note. *1101 = support unpacked 32 bpp (non-palletized A:8-R:8-G:8-B:8) for per pixel blending. *1110 = support 16 bpp (non-palletized A:4-R:4-G:4-B:4) for per pixel blending. (16 level blending)	
ALPHA_SEL	[1]	Select Alpha value by If Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values If Per pixel blending (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[31:24] data in word boundary *(only when BPPMODE_F = 4'b1101)* DATA[31:28], [15:12] data in word boundary (*only when BPPMODE_F = 4'b1110)*	0
ENWIN_F	[0]	Video output and the logic immediately enable/disable. 0 = Disables the video output and the VIDEO control signal. 1 = Enables the video output and the VIDEO control signal.	0

6.13 WINDOW 0 POSITION CONTROL A REGISTER (VIDOSD0A, R/W, ADDRESS = 0XEE00_ 0040)

VIDOSD0A	Bit	Description	Reset Value
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be even value.)	0

6.14 WINDOW 0 POSITION CONTROL B REGISTER (VIDOSD0B, R/W, ADDRESS = 0XEE00_ 0044)

VIDOSD0B	Bit	Description	Reset Value
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be odd value.)	0

NOTE: Registers must have word boundary X position.
So, 24 BPP mode must have X position by 1 pixel. (Ex, X = 0, 1, 2, 3….)
16 BPP mode must have X position by 2 pixel. (Ex, X = 0, 2, 4, 6….)
8 BPP mode must have X position by 4 pixel. (Ex, X = 0, 4, 8, 12….)

6.15 WINDOW 0 POSITION CONTROL C REGISTER (VIDOSD0C, R/W, ADDRESS = 0XEE00_ 0048)

VIDOSD0C	Bit	Description	Reset Value
Reserved	[25:24]	Reserved **(should be 0)**	0
OSDSIZE	[23:0]	Window Size Example: Height * Width (Number of Word) **Note.** Set filed value for Interlace YUV if (ITU601/656 if)	0

6.16 WINDOW 1 POSITION CONTROL A REGISTER (VIDOSD0C, R/W, ADDRESS = 0XEE00_ 0050)

VIDOSD1A	Bit	Description	Reset Value
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be even value.)	0

6.17 WINDOW 1 POSITION CONTROL B REGISTER (VIDOSD1B, R/W, ADDRESS = 0XEE00_ 0054)

VIDOSD1B	Bit	Description	Reset Value
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be odd value.)	0

NOTE1: Registers must have word boundary X position.
Therefore, 24 BPP mode must have X position by 1 pixel. (Ex, X = 0, 1, 2, 3….)
16 BPP mode must have X position by 2 pixel. (Ex, X = 0, 2, 4, 6….)
8 BPP mode must have X position by 4 pixel. (Ex, X = 0, 4, 8, 12….)

NOTE2: In case of VP interface, window's height should be greater than 1. (RigthBotY – LeftTopY > 0)

6.18 WINDOW 1 POSITION CONTROL C REGISTER (VIDOSD1C, R/W, ADDRESS = 0XEE00_ 0058)

VIDOSD1C	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
ALPHA0_R_H	[23:20]	Red Alpha upper value(case AEN == 0)	0
ALPHA0_G_H	[19:16]	Green Alpha upper value(case AEN == 0)	0
ALPHA0_B_H	[15:12]	Blue Alpha upper value(case AEN == 0)	0
ALPHA1_R_H	[11:8]	Red Alpha upper value(case AEN == 1)	0
ALPHA1_G_H	[7:4]	Green Alpha upper value(case AEN == 1)	0
ALPHA1_B_H	[3:0]	Blue Alpha upper value(case AEN == 1)	0

NOTE: Refer to VIDW1ALPHA0,1 register

6.19 WINDOW 1 POSITION CONTROL D REGISTER (VIDOSD1D, R/W, ADDRESS = 0XEE00_ 005C)

VIDOSD1D	Bit	Description	Reset Value
Reserved	[25:24]	Reserved (should be 0)	0
OSDSIZE	[23:0]	Window Size Example: Height * Width(Number of Word) **Note**. Set filed value for Interlace YUV if (ITU601/656 if)	0

6.20 WINDOW 2 POSITION CONTROL A REGISTER (VIDOSD2A, R/W, ADDRESS = 0XEE00_ 0060)

VIDOSD2A	Bit	Description	Reset Value
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be even value.)	0

6.21 WINDOW 2 POSITION CONTROL B REGISTER (VIDOSD2B, R/W, ADDRESS = 0XEE00_ 0064)

VIDOSD2B	Bit	Description	Reset Value
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be odd value.)	0

NOTE: Registers must have word boundary X position.
Therefore, 24 BPP mode must have X position by 1 pixel. (Ex, X = 0, 1, 2, 3….)
16 BPP mode must have X position by 2 pixel. (Ex, X = 0, 2, 4, 6….)
8 BPP mode must have X position by 4 pixel. (Ex, X = 0, 4, 8, 12….)

6.22 WINDOW 2 POSITION CONTROL C REGISTER (VIDOSD2C, R/W, ADDRESS = 0XEE00_ 0068)

VIDOSD2C	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
ALPHA0_R_H	[23:20]	Red Alpha upper value(case AEN == 0)	0
ALPHA0_G_H	[19:16]	Green Alpha upper value(case AEN == 0)	0
ALPHA0_B_H	[15:12]	Blue Alpha upper value(case AEN == 0)	0
ALPHA1_R_H	[11:8]	Red Alpha upper value(case AEN == 1)	0
ALPHA1_G_H	[7:4]	Green Alpha upper value(case AEN == 1)	0
ALPHA1_B_H	[3:0]	Blue Alpha upper value(case AEN == 1)	0

NOTE: Refer to VIDW2ALPHA0,1 register

6.23 WINDOW 2 POSITION CONTROL D REGISTER (VIDOSD2D, R/W, ADDRESS = 0XEE00_ 006C)

VIDOSD2D	Bit	Description	Reset Value
Reserved	[25:24]	Reserved (should be 0)	0
OSDSIZE	[23:0]	Window Size Example: Height * Width(Number of Word) **Note.** Set filed value for Interlace YUV if (ITU601/656 if)	0

6.24 WINDOW 3 POSITION CONTROL A REGISTER (VIDOSD3A, R/W, ADDRESS = 0XEE00_ 0070)

VIDOSD3A	Bit	Description	Reset Value
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be even value.)	0

6.25 WINDOW 3 POSITION CONTROL B REGISTER (VIDOSD3B, R/W, ADDRESS = 0XEE00_ 0074)

VIDOSD3B	Bit	Description	Reset Value
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be odd value.)	0

NOTE: Registers must have word boundary X position.
Therefore, 24 BPP mode must have X position by 1 pixel. (Ex, X = 0, 1, 2, 3….)
16 BPP mode must have X position by 2 pixel. (Ex, X = 0, 2, 4, 6….)
8 BPP mode must have X position by 4 pixel. (Ex, X = 0, 4, 8, 12….)

6.26 WINDOW 3 POSITION CONTROL C REGISTER (VIDOSD3C, R/W, ADDRESS = 0XEE00_ 0078)

VIDOSD3C	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
ALPHA0_R_H	[23:20]	Red Alpha upper value(case AEN == 0)	0
ALPHA0_G_H	[19:16]	Green Alpha upper value(case AEN == 0)	0
ALPHA0_B_H	[15:12]	Blue Alpha upper value(case AEN == 0)	0
ALPHA1_R_H	[11:8]	Red Alpha upper value(case AEN == 1)	0
ALPHA1_G_H	[7:4]	Green Alpha upper value(case AEN == 1)	0
ALPHA1_B_H	[3:0]	Blue Alpha upper value(case AEN == 1)	0

NOTE: Refer to VIDW3ALPHA0, 1 register

6.27 WINDOW 4 POSITION CONTROL A REGISTER (VIDOSD4A, R/W, ADDRESS = 0XEE00_ 0080)

VIDOSD4A	Bit	Description	Reset Value
OSD_LeftTopX_F	[21:11]	Horizontal screen coordinate for left top pixel of OSD image	0
OSD_LeftTopY_F	[10:0]	Vertical screen coordinate for left top pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be even value.)	0

6.28 WINDOW 4 POSITION CONTROL B REGISTER (VIDOSD4B, R/W, ADDRESS = 0XEE00_ 0084)

VIDOSD4B	Bit	Description	Reset Value
OSD_RightBotX_F	[21:11]	Horizontal screen coordinate for right bottom pixel of OSD image	0
OSD_RightBotY_F	[10:0]	Vertical screen coordinate for right bottom pixel of OSD image (For interlace TV output, this value MUST be set to half of the original screen y coordinate. The original screen y coordinate MUST be odd value.)	0

NOTE: Registers must have word boundary X position.
Therefore, 24 BPP mode must have X position by 1 pixel. (Ex, X = 0, 1, 2, 3….)
16 BPP mode must have X position by 2 pixel. (Ex, X = 0, 2, 4, 6….)
8 BPP mode must have X position by 4 pixel. (Ex, X = 0, 4, 8, 12….)

6.29 WINDOW 4 POSITION CONTROL C REGISTER (VIDOSD4C, R/W, ADDRESS = 0XEE00_ 0088)

VIDOSD4C	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
ALPHA0_R_H	[23:20]	Red Alpha upper value(case AEN == 0)	0
ALPHA0_G_H	[19:16]	Green Alpha upper value(case AEN == 0)	0
ALPHA0_B_H	[15:12]	Blue Alpha upper value(case AEN == 0)	0
ALPHA1_R_H	[11:8]	Red Alpha upper value(case AEN == 1)	0
ALPHA1_G_H	[7:4]	Green Alpha upper value(case AEN == 1)	0
ALPHA1_B_H	[3:0]	Blue Alpha upper value(case AEN == 1)	0

NOTE: Refer to VIDW4ALPHA0,1 register

6.30 FRAME BUFFER ADDRESS 0 REGISTER (VIDW)

- VIDW00ADD0B0, R/W, Address = 0xEE00_ 00A0
- VIDW00ADD0B1, R/W, Address = 0xEE00_ 00A4
- VIDW01ADD0B0, R/W, Address = 0xEE00_ 00A8
- VIDW01ADD0B1, R/W, Address = 0xEE00_ 00AC
- VIDW02ADD0, R/W, Address = 0xEE00_ 00B0
- VIDW03ADD0, R/W, Address = 0xEE00_ 00B8
- VIDW04ADD0, R/W, Address = 0xEE00_ 00C0

VIDWxxADD0	Bit	Description	Reset Value
VBANK_F	[31:24]	These bits indicate A [31:24] of the bank location for the video buffer in the system memory.	0
VBASEU_F	[23:0]	These bits indicate A [23:0] of the start address of the Video frame buffer.	0

6.31 FRAME BUFFER ADDRESS 1 REGISTER (VIDW)

- VIDW00ADD1B0, R/W, Address = 0xEE00_ 00D0
- VIDW00ADD1B1, R/W, Address = 0xEE00_ 00D4
- VIDW01ADD1B0, R/W, Address = 0xEE00_ 00D8
- VIDW01ADD1B1, R/W, Address = 0xEE00_ 00DC
- VIDW02ADD1, R/W, Address = 0xEE00_ 00E0
- VIDW03ADD1, R/W, Address = 0xEE00_ 00E8
- VIDW04ADD1, R/W, Address = 0xEE00_ 00F0

VIDWxxADD1	Bit	Description	Reset Value
VBASEL_F	[23:0]	These bits indicate A[23:0] of the end address of the Video frame buffer. VBASEL = VBASEU + (PAGEWIDTH+OFFSIZE) x (LINEVAL+1)	0x0

6.32 FRAME BUFFER ADDRESS 2 REGISTER (VIDW)

- VIDW00ADD2, R/W, Address = 0xEE00_ 0100
- VIDW01ADD2, R/W, Address = 0xEE00_ 0104
- VIDW02ADD2, R/W, Address = 0xEE00_ 0108
- VIDW03ADD2, R/W, Address = 0xEE00_ 010C
- VIDW04ADD2, R/W, Address = 0xEE00_ 0110

VIDWxxADD2	Bit	Description	Reset Value
OFFSIZE_F	[25:13]	Virtual screen offset size (the number of byte). This value defines the difference between the address of the last byte displayed on the previous Video line and the address of the first byte to be displayed in the new Video line. **OFFSIZE_F must have value that is multiple of 4-byte size or 0.**	0
PAGEWIDTH_F	[12:0]	Virtual screen page width (the number of byte). This value defines the width of the view port in the frame. **PAGEWIDTH must have bigger value than the burst size and the size must be aligned word boundary.**	0

NOTE: 'PAGEWIDTH + OFFSET' should be aligned double-word aligned (8-byte)

6.33 VP 1 INTERFACE TIMING CONTROL 0 REGISTER (VP1TCON0, R/W, ADDRESS = 0XEE00_ 0118)

VP1TCON0	Bit	Description	Reset Value
VP1_RATECON_EN	[31]	RATE Register control enable (VP- FIMD clk ratio) 0 = auto control (when FIMD VCLK is determined by CLKVAL_F and HCLK) 1 = rate control by register (VP1_CLKRATE)	0
Reserved	[30: 8]	Reserved	0
VP1_CLKRATE	[7: 0]	Ratio (= VP1_CLKATE + 1) (should be less than System clock (VP) / VCLK(FIMD)) range : 0x01 ~ 0xFE	0

6.34 VP 1 INTERFACE TIMING CONTROL 1 REGISTER (VP1TCON1, R/W, ADDRESS = 0XEE00_ 011C)

VP1TCON1	Bit	Description	Reset Value
VP1_VTREGCON_EN	[31]	Register control enable (Vertical Timing) 0 = auto control (controlled by VBPD@VIDTCON0) 1 = timing control by register (VP1_VBPD)	0
Reserved	[31:28]	Reserved	0
VP1_VBPD	[27:16]	VP Vertical BackPorch should be sufficiently big to prevent VP interrupt (smaller than VBPD@VIDTCON0)	0
Reserved	[15: 0]	Reserved	0

6.35 VIDEO INTERRUPT CONTROL 0 REGISTER (VIDINTCON0, R/W, ADDRESS = 0XEE00_ 0130)

VIDINTCON0	Bit	Description	Reset Value
INTVPEN	[26]	VP interrupts Enable control bit. 0 = Disables VP interrupt 1 = Enables VP interrupt **Note:** This bit is meaningful if INTEN is high.	0
FIFOINTERVAL	[25:20]	These bits control the interval of the FIFO interrupt.	0
SYSMAINCON	[19]	Sending complete interrupt enable bit to Main LCD 0 = Disables Interrupt 1 = Enables Interrupt **Note:** This bit is meaningful if both INTEN and I80IFDONE are high.	0
SYSSUBCON	[18]	Sending complete interrupt enable bit to Sub LCD 0 = Disables Interrupt. 1 = Enables Interrupt. **Note:** This bit is meaningful if both INTEN and I80IFDONE are high.	0
I80IFDONE	[17]	I80 Interface Interrupt Enable control (only for I80 Interface mode). 0 = Disables Interrupt. 1 = Enables Interrupt. **Note:** This bit is meaningful if INTEN is high.	0
FRAMESEL0	[16:15]	Video Frame Interrupt 0 at start of : 00 = BACK Porch 01 = VSYNC 10 = ACTIVE 11 = FRONT Porch	0
FRAMESEL1	[14:13]	Video Frame Interrupt 1 at start of : 00 = None 01 = BACK Porch 10 = VSYNC 11 = FRONT Porch	0
INTFRMEN	[12]	Video Frame interrupts Enable control bit. 0 = Disables Video Frame Interrupt 1 = Enables Video Frame Interrupt **Note:** This bit is meaningful when INTEN is high.	0
FIFOSEL	[11:5]	FIFO Interrupt control bit, each bit has the meaning of [11] Window 4 control (0 = disable, 1 = enable) [10] Window 3 control (0 = disable, 1 = enable) [9] Window 2 control (0 = disable, 1 = enable) [8] reserved [7] reserved [6] Window 1 control (0 = disable, 1 = enable) [5] Window 0 control (0 = disable, 1 = enable) **Note:** This bit is meaningful if both INTEN and INTFIFOEN are high	0
FIFOLEVEL	[4:2]	Video FIFO Interrupt Level Select 000 = 0 ~ 25% 001 = 0 ~ 50% 010 = 0 ~ 75% 011 = 0% (empty) 100 = 100% (full)	0

VIDINTCON0	Bit	Description	Reset Value
INTFIFOEN	[1]	Video FIFO interrupts Enable control bit. 0 = Disables Video FIFO Level Interrupt 1 = Enables Video FIFO Level Interrupt **Note:** This bit is meaningful if INTEN is high.	0
INTEN	[0]	Video interrupts Enable control bit. 0 = Disables Video Interrupt 1 = Enables Video Interrupt	0

NOTE 1: It is possible to select maximum two points if the video frame interrupt occurred by setting FRAMESEL0 and FRAMESEL1. For example, if FRAMESEL0=00 and FRAMESEL1=11, video frame interrupt is triggered both at the start of back porch and front porch.

NOTE 2: S5PC100 interrupt controller has 4 interrupt sources which is related to display controller. Those are LCD[0], LCD[1], LCD[2] and LCD[3]. (Refer to *4.1 VECTORED INTERRUPT CONTROLLER*). LCD[0] is FIFO Level interrupt. LCD[1] is video frame sync interrupt. LCD[2] is i80 done interface interrupt. LCD[3] is VP under-run interrupt.

6.36 VIDEO INTERRUPT CONTROL 1 REGISTER (VIDINTCON1, R/W, ADDRESS = 0XEE00_ 0134)

VIDINTCON1	Bit	Description	Reset Value
INTVPPEND-	[5]	VP under-run interrupt. Write "1" to clear this bit 0 = Interrupt has not been requested 1 = VP under-run status has asserted the interrupt request	0
Reserved	[4:3]	Reserved **(should be 0)**	0
INTI80PEND	[2]	i80 Done interrupt. Write "1" to clear this bit . 0 = Interrupt has not been requested 1 = i80 Done status has asserted the interrupt request	0
INTFRMPEND	[1]	Frame sync interrupt. Write "1" to clear this bit. 0 = Interrupt has not been requested 1 = Frame sync status has asserted the interrupt request	0
INTFIFOPEND	[0]	FIFO Level interrupt. Write "1" to clear this bit. 0 = Interrupt has not been requested 1 = FIFO empty status has asserted the interrupt request	0

6.37 WIN1 COLOR KEY 0 REGISTER (W1KEYCON0, R/W, ADDRESS = 0XEE00_ 0140)

W1KEYCON0	Bit	Description	Reset Value
KEYBLEN_F	[26]	Color Key (Chroma key) Enable control 0 = Disables (Disables blending operation) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = Disables color key 1 = Enables color key	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value match fore-ground image with COLVAL, pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value match back-ground with COLVAL, pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit is corresponding to the COLVAL [23:0]. If some position bit is set then it disables position bit of COLVAL.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0, A_FUNC = 0x2, B_FUNC = 0x3 to use alpha blending using color key

6.38 WIN 1 COLOR KEY 1 REGISTER (W1KEYCON1, R/W, ADDRESS = 0XEE00_ 0144)

W1KEYCON1	Bit	Description	Reset Value
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

6.39 WIN2 COLOR KEY 0 REGISTER (W2KEYCON0, R/W, ADDRESS = 0XEE00_ 0148)

W2KEYCON0	Bit	Description	Reset Value
KEYBLEN_F	[26]	Color Key (Chroma key) Enable control 0 = Disables (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = Disables color key 1 = Enables color key	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value matches fore-ground image with COLVAL, pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value matches back-ground with COLVAL, pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit corresponds to the COLVAL [23:0]. If some position bit is set then it disables position bit of COLVAL.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0, A_FUNC = 0x2, B_FUNC = 0x3 to use alpha blending using color key

6.40 WIN2 COLOR KEY 1 REGISTER (W2KEYCON1, R/W, ADDRESS = 0XEE00_ 014C)

W2KEYCON1	Bit	Description	Reset Value
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

6.41 WIN3 COLOR KEY 0 REGISTER (W3KEYCON0, R/W, ADDRESS = 0XEE00_ 0150)

W3KEYCON0	Bit	Description	Reset Value
KEYBLEN_F	[26]	Color Key (Chroma key) Enable control 0 = Disables (Disables blending operation) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = Disables color key 1 = Enables color key	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value match fore-ground image with COLVAL, pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value match back-ground with COLVAL, pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit is corresponding to the COLVAL [23:0]. If some position bit is set then it disables COLVAL position bit.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0, A_FUNC = 0x2, B_FUNC = 0x3 to use alpha blending using color key

6.42 WIN3 COLOR KEY 1 REGISTER (W3KEYCON1, R/W, ADDRESS = 0XEE00_ 0154)

W3KEYCON1	Bit	Description	Reset Value
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

6.43 WIN4 COLOR KEY 0 REGISTER (W4KEYCON0, R/W, ADDRESS = 0XEE00_ 0158)

W4KEYCON0	Bit	Description	Reset Value
KEYBLEN_F	[26]	Color Key (Chroma key) Enable control 0 = Disables (Disables blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = Disables color key 1 = Enables color key enable	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value match fore-ground image with COLVAL, pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value match back-ground with COLVAL, pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit is corresponding to the COLVAL [23:0]. If some position bit is set then it disables COLVAL position bit.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0, A_FUNC = 0x2, B_FUNC = 0x3 to use alpha blending using color key

6.44 WIN4 COLOR KEY 1 REGISTER (W4KEYCON1, R/W, ADDRESS = (W4KEYCON1, R/W, ADDRESS = 0XEE00_015C)

W4KEYCON1	Bit	Description	Reset Value
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

NOTE: COLVAL and COMPKEY use 24bit color data at all bpp mode.

@ BPP24 mode: 24-bit color value is valid.

A. COLVAL

- Red: COLVAL [23:17]
- Green: COLVAL [15: 8]
- Blue: COLVAL [7:0]

B. COMPKEY

- Red: COMPKEY [23:17]
- Green: COMPKEY [15: 8]
- Blue: COMPKEY [7:0]

@ BPP16 (5:6:5) mode: 16-bit color value is valid

A. COLVAL

- Red: COLVAL [23:19]
- Green: COLVAL [15: 10]
- Blue: COLVAL [7:3]

B. COMPKEY

- Red: COMPKEY [23:19]
- Green: COMPKEY [15: 10]
- Blue: COMPKEY [7:3]
- COMPKEY [18:16] must be 0x7.
- COMPKEY [9: 8] must be 0x3.
- COMPKEY [2:0] must be 0x7.

NOTE: COMPKEY register must be set properly for the each bpp mode.

6.45 DITHERING CONTROL 1 REGISTER (DITHMODE, R/W, ADDRESS = 0XEE00_0170)

DITHMODE	Bit	Description	Reset Value
	[7]	Not used for normal access (Write not-zero values to these register make to come out abnormal result)	0
RDithPos	[6:5]	Red Dither bit control 00 = 8-bit 01 = 6-bit 10 = 5-bit	0
GDithPos	[4:3]	Green Dither bit control 00 = 8-bit 01 = 6-bit 10 = 5-bit	0
BDithPos	[2:1]	Blue Dither bit control 00 = 8-bit 01 = 6-bit 10 = 5-bit	0
DITHEN_F	[0]	Dithering Enable bit 0 = Disables dithering 1 = Enables dithering	0

6.46 WIN0 COLOR MAP (WIN0MAP, R/W, ADDRESS = 0XEE00_ 0180)

WIN0MAP	Bit	Description	Reset Value
MAPCOLEN_F	[24]	Window's color mapping control bit . If this bit is enabled then Video DMA will stop, and MAPCOLOR will be appear on back-ground image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	Color Value	0

6.47 WIN1 COLOR MAP (WIN1MAP, R/W, ADDRESS = 0XEE00_ 0184)

WIN1MAP	Bit	Description	Reset Value
MAPCOLEN_F	[24]	Window's color mapping control bit. If this bit is enabled then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	Color Value	0

6.48 WIN2 COLOR MAP (WIN2MAP, R/W, ADDRESS = 0XEE00_ 0188)

WIN2MAP	Bit	Description	Reset Value
MAPCOLEN_F	[24]	Window's color mapping control bit. If this bit is enabled then Video DMA stops, and MAPCOLOR appears on back-ground image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	Color Value	0

6.49 WIN3 COLOR MAP (WIN3MAP, R/W, ADDRESS = 0XEE00_ 018C)

WIN3MAP	Bit	Description	Reset Value
MAPCOLEN_F	[24]	Window's color mapping control bit . If this bit is enabled then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	Color Value	0

6.50 WIN4 COLOR MAP (WIN4MAP, R/W, ADDRESS = 0XEE00_ 0190)

WIN4MAP	Bit	Description	Reset Value
MAPCOLEN_F	[24]	Window's color mapping control bit. If this bit is enabled then Video DMA stops and MAPCOLOR appears on background image instead of original image. 0 = Disables 1 = Enables	0
MAPCOLOR	[23:0]	Color Value	0

6.51 WINDOW PALETTE CONTROL REGISTER (WPALCON_H, R/W, ADDRESS = 0XEE00_ 019C)

WPALCON_H	Bit	Description	Reset Value
Reserved	[31:19]	Reserved	0
W4PAL_H	[18:17]	W4PAL[2:1]	0
Reserved	[16:15]	Reserved	0
W3PAL_H	[14:13]	W3PAL[2:1]	0
Reserved	[12:11]	Reserved	0
W2PAL_H	[10: 9]	W2PAL[2:1]	0
Reserved	[8: 0]	Reserved	0

6.52 WINDOW PALETTE CONTROL REGISTER (WPALCON_L, R/W, ADDRESS = 0XEE00_ 01A0)

WPALCON_L	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
PALUPDATEEN	[9]	0 = Normal Mode 1 = Enable (Palette Update)	0
W4PAL_L	[8]	W4PAL[0]	0
W3PAL_L	[7]	W3PAL[0]	0
W2PAL_L	[6]	W2PAL[0]	0
W1PAL_L	[5: 3]	W1PAL[2:0]	0
W0PAL_L	[2: 0]	W0PAL[2:0]	0

NOTE: WPALCON = {WPALCON_H,WPALCON_L}

WPALCON	Description	Reset Value
PALUPDATEEN	0 = Normal Mode 1 = Enable (Palette Update)	0
W4PAL[3:0]	This bit determines the size of the palette data format of Window 4 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A : 8bit)	0
W3PAL[2:0]	This bit determines the size of the palette data format of Window 3 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5)	0

WPALCON	Description	Reset Value
	100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A : 8bit)	
W2PAL[2:0]	This bit determines the size of the palette data format of Window 2 000 = 16-bit (5:6:5) 001 = 16-bit (A:5:5:5) 010 = 18-bit (6:6:6) 011 = 18-bit (A:6:6:5) 100 = 19-bit (A:6:6:6) 101 = 24-bit (8:8:8) 110 = 25-bit (A:8:8:8) 111 = 32-bit (8:8:8:8) (A : 8bit)	0
W1PAL[2:0]	This bit determines the size of the palette data format of Window 1 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5) 111 = 32-bit (8:8:8:8) (A : 8bit)	0
W0PAL[2:0]	This bit determines the size of the palette data format of Window 0 000 = 25-bit (A:8:8:8) 001 = 24-bit (8:8:8) 010 = 19-bit (A:6:6:6) 011 = 18-bit (A:6:6:5) 100 = 18-bit (6:6:6) 101 = 16-bit (A:5:5:5) 110 = 16-bit (5:6:5) 111 = 32-bit (8:8:8:8) (A : 8bit)	0

NOTE: W0/W1'S BIT MAP IS DIFFERENT FROM W2/W3/W4'S

6.53 I80 / RGB TRIGGER CONTROL REGISTER (TRIGCON, R/W, ADDRESS = 0XEE00_ 01A4)

TRIGCON	Bit	Description	Reset Value
W1BUFTRGCMD	[7]	1 = Window 0 Buffer Software Triggering Command [Write Only]	0
W0BUFTRGCMD	[6]	1 = Window 1 Buffer Software Triggering Command [Write Only]	0
Reserved	[5:3]	Reserved	0
SWFRSTATUS	[2]	I80 Frame Done Status [Read Only] 0 = Not Requested 1 = Requested * Clear Condition: Read or New Frame Start * Only when TRGMODE is '1'	0
SWTRGCMD	[1]	1 = i80 Software Triggering Command [Write Only] * If TRGMODE is '1'	0
TRGMODE	[0]	0 = i80 Software Trigger Disable 1 = i80 Software Trigger Enable	0

NOTE: Two continuous SW trigger inputs, which generates in some video clocks (VCLK) is recognized as one.

6.54 ITU INTERFACE CONTROL 0 (ITUIFCON0, R/W, ADDRESS = 0XEE00_ 01A8)

I80IFCONAx	Bit	Description	Reset Value
Reserved	[26:25]	Reserved	0
SELVSYNC	[24]	Select the Vsync mode 0 = equal leading edge with Hsync 1 = Delayed Vsync (standard)	0
DLYVSYNC	[23:16]	Numbers of clock cycles for delay of the VSYNC signal (If SELVSYNC is '1') DLYVSYNC+1 * over 64cycles (standard)	0
Reserved	[15:10]	Reserved	0
I656FIELD	[9]	The polarity of the F value (in timing reference code) 0 = normal 1 = inverted	0
I656CLK	[8]	The polarity of the V656_CLK active edge. 0 = normal 1 = inverted	0
Reserved	[7]	Reserved	0
I601HREF	[6]	The polarity of the VEN_HREF Signal 0 = normal 1 = inverted	0
I601VSYNC	[5]	The polarity of the VEN_VSYNC Signal 0 = normal 1 = inverted	0
I601HSYNC	[4]	The polarity of the VEN_HSYNC Signal 0 = normal 1 = inverted	0
I601FIELD	[3]	The polarity of the VEN_FIELD Signal 0 = normal 1= inverted	0
I601CLK	[2]	The polarity of the V601_CLK active edge 0 = normal 1 = inverted	0
Reserved	[1:0]	Reserved	0

6.55 LCD I80 INTERFACE CONTROL 0 (I80IFCONAX)

- I80IFCONA0, R/W, Address = 0xEE00_ 01B0
- I80IFCONA1, R/W, Address = 0xEE00_ 01B4

I80IFCONAx	Bit	Description	Reset Value
Reserved	[22:20]	Reserved	0
LCD_CS_SETUP	[19:16]	Numbers of clock cycles for the active period of the address signal enable to the chip select enable.	0
LCD_WR _SETUP	[15:12]	Numbers of clock cycles for the active period of the CS signal enable to the write signal enable.	0
LCD_WR_ACT	[11:8]	Numbers of clock cycles for the active period of the chip select enable.	0
LCD_WR _HOLD	[7:4]	Numbers of clock cycles for the active period of the chip select disable to the write signal disable.	0
Reserved	[3]	Reserved	
RSPOL	[2]	The polarity of the RS Signal 0 = Low 1 = High.	0
Reserved	[1]	Reserved	0
I80IFEN	[0]	LCD I80 Interface control 0 = Disables 1 = Enables	0

6.56 LCD I80 INTERFACE CONTROL 1 (I80IFCONBX)

- I80IFCONB0, R/W, Address = 0xEE00_ 01B8
- I80IFCONB1, R/W, Address = 0xEE00_ 01BC

I80IFCONBx	Bit	Description	Reset Value
Reserved	[11:10]	Reserved	0
NORMAL_CMD_ST	[9]	1 = Normal Command Start * Auto clear after sending one set of commands	0
Reserved	[8:7]	Reserved	
FRAME_SKIP	[6:5]	I80 Interface Output Frame Decimation Factor 00 = 1 (No Skip) 01 = 2 10 = 3	00
Reserved	[4]	Reserved	0
AUTO_CMD_RATE	[3:0]	0000 : Disables auto command 0001 = per 2 Frames 0010 = per 4 Frames 0011 = per 6 Frames … 1111 = per 30 Frames	0000

6.57 LCD I80 INTERFACE COMMAND CONTROL 0 (LDI_CMDCON0, R/W, ADDRESS = 0XEE00_ 01D0)

LDI_CMDCON0	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	
CMD11_EN	[23:22]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 : Enables Normal and Auto Command	00
CMD10_EN	[21:20]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD9_EN	[19:18]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD8_EN	[17:16]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD7_EN	[15:14]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD6_EN	[13:12]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD5_EN	[11:10]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD4_EN	[9:8]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD3_EN	[7:6]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD2_EN	[5:4]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command	00

LDI_CMDCON0	Bit	Description	Reset Value
		11 = Normal and Auto Command Enable	
CMD1_EN	[3:2]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00
CMD0_EN	[1:0]	00 = Disables 01 = Enables Normal Command 10 = Enables Auto Command 11 = Enables Normal and Auto Command	00

6.58 LCD I80 INTERFACE COMMAND CONTROL 1 (LDI_CMDCON1, R/W, ADDRESS = 0XEE00_ 01D4)

LDI_CMDCON1	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0
CMD11_RS	[11]	Command 11 RS control	0
CMD10_RS	[10]	Command 10 RS control	0
CMD9_RS	[9]	Command 9 RS control	0
CMD8_RS	[8]	Command 8 RS control	0
CMD7_RS	[7]	Command 7 RS control	0
CMD6_RS	[6]	Command 6 RS control	0
CMD5_RS	[5]	Command 5 RS control	0
CMD4_RS	[4]	Command 4 RS control	0
CMD3_RS	[3]	Command 3 RS control	0
CMD2_RS	[2]	Command 2 RS control	0
CMD1_RS	[1]	Command 1 RS control	0
CMD0_RS	[0]	Command 0 RS control	0

6.59 I80 SYSTEM INTERFACE MANUAL COMMAND CONTROL 0 (SIFCCON0, R/W, ADDRESS = 0XEE00_ 01E0)

SIFCCON0	Bit	Description	Reset Value
Reserved	[7]	Reserved (should be 0)	0
SYS_ST_CON	[6]	LCD i80 System Interface ST Signal control. 0 = Low 1 = High	0
SYS_RS_CON	[5]	LCD i80 System Interface RS Signal control. 0 = Low 1 = High	0
SYS_nCS0_CON	[4]	LCD i80 System Interface nCS0 (main) Signal control 0 = Disables (High) 1 = Enables (Low)	0
SYS_nCS1_CON	[3]	LCD i80 System Interface nCS1 (sub) Signal control 0 = Disables (High) 1 = Enables (Low)	0
SYS_nOE_CON	[2]	LCD i80 System Interface nOE Signal control 0 = Disables (High) 1 = Enables (Low)	0
SYS_nWE_CON	[1]	LCD i80 System Interface nWE Signal control 0 = Disables (High) 1 = Enables (Low)	0
SCOMEN	[0]	LCD i80 System Interface Command Mode Enable 0 = Disables (Normal Mode) 1= Enables (Manual Command Mode)	

6.60 I80 SYSTEM INTERFACE MANUAL COMMAND CONTROL 1 (SIFCCON1, R/W, ADDRESS = 0XEE00_ 01E4)

SIFCCON1	Bit	Description	Reset Value
SYS_WDATA	[23:0]	LCD i80 System Interface Write Data Control	0

6.61 I80 SYSTEM INTERFACE MANUAL COMMAND CONTROL 2 (SIFCCON2, R/W, ADDRESS = 0XEE00_ 01E8)

SIFCCON2	Bit	Description	Reset Value
SYS_RDATA	[23:0]	LCD i80 System Interface Read Data Control	0

6.62 WINDOW 0 ALPHA0 CONTROL REGISTER (VIDW0ALPHA0, R/W, ADDRESS = 0XEE00_ 0200)

VIDW0ALPHA0	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
ALPHA0_R	[23:16]	Red Alpha value(case AEN == 0)	0
ALPHA0_G	[15:8]	Green Alpha value(case AEN == 0)	0
ALPHA0_B	[7:0]	Blue Alpha value(case AEN == 0)	0

6.63 WINDOW 0 ALPHA1 CONTROL REGISTER (VIDW0 ALPHA1, R/W, ADDRESS = 0XEE00_ 0204)

VIDW0ALPHA1	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
ALPHA1_R	[23:16]	Red Alpha value(case AEN == 1)	0
ALPHA1_G	[15:8]	Green Alpha value(case AEN == 1)	0
ALPHA1_B	[7:0]	Blue Alpha value(case AEN == 1)	0

6.64 WINDOW 1 ALPHA0 CONTROL REGISTER (VIDW1ALPHA0, R/W, ADDRESS = 0XEE00_ 0208)

VIDW1ALPHA0	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA0_R_L	[19:16]	Red Alpha lower value(case AEN == 0)	0
Reserved	[15:12]	Reserved	0
ALPHA0_G_L	[11: 8]	Green Alpha lower value(case AEN == 0)	0
Reserved	[7: 4]	Reserved	0
ALPHA0_B_L	[3: 0]	Blue Alpha lower value(case AEN == 0)	0

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R(G,B)_H[3:0]@VIDOSD1C
ALPHA0_R(G,B) [3:0] = ALPHA0_R(G,B)_L[3:0]@VIDW1ALPHA0

6.65 WINDOW 1 ALPHA1 CONTROL REGISTER (VIDW0ALPHA1, R/W, ADDRESS = 0XEE00_020C)

VIDW0ALPHA1	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA1_R_L	[19:16]	Red Alpha lower value(case AEN == 1)	0
Reserved	[15:12]	Reserved	0
ALPHA1_G_L	[11: 8]	Green Alpha lower value(case AEN == 1)	0
Reserved	[7: 4]	Reserved	0
ALPHA1_B_L	[3: 0]	Blue Alpha lower value(case AEN == 1)	0

NOTE: ALPHA1_R(G,B) [7:4] = ALPHA1_R(G,B)_H[3:0]@VIDOSD1C
ALPHA1_R(G,B) [3:0] = ALPHA1_R(G,B)_L[3:0]@VIDW1ALPHA1

6.66 WINDOW 2 ALPHA0 CONTROL REGISTER (VIDW0ALPHA0, R/W, ADDRESS = 0XEE00_ 0210)

VIDW0ALPHA0	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA0_R_L	[19:16]	Red Alpha lower value(case AEN == 0)	0
Reserved	[15:12]	Reserved	0
ALPHA0_G_L	[11: 8]	Green Alpha lower value(case AEN == 0)	0
Reserved	[7: 4]	Reserved	0
ALPHA0_B_L	[3: 0]	Blue Alpha lower value(case AEN == 0)	0

NOTE: ALPHA0_R (G, B) [7:4] = ALPHA0_R (G, B)_H[3:0]@VIDOSD2C
ALPHA0_R (G, B) [3:0] = ALPHA0_R (G, B)_L[3:0]@VIDW2ALPHA0

6.67 WINDOW 2 ALPHA1 CONTROL REGISTER (VIDW2ALPHA1, R/W, ADDRESS = 0XEE00_ 0214)

VIDW2ALPHA1	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA1_R_L	[19:16]	Red Alpha lower value(case AEN == 1)	0
Reserved	[15:12]	Reserved	0
ALPHA1_G_L	[11: 8]	Green Alpha lower value(case AEN == 1)	0
Reserved	[7: 4]	Reserved	0
ALPHA1_B_L	[3: 0]	Blue Alpha lower value(case AEN == 1)	0

NOTE: ALPHA1_R (G, B) [7:4] = ALPHA1_R (G, B)_H[3:0]@VIDOSD2C
ALPHA1_R (G, B) [3:0] = ALPHA1_R (G, B)_L[3:0]@VIDW2ALPHA1

6.68 WINDOW 3 ALPHA0 CONTROL REGISTER (VIDW0ALPHA0, R/W, ADDRESS = 0XEE00_ 0218)

VIDW0ALPHA0	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA0_R_L	[19:16]	Red Alpha lower value(case AEN == 0)	0
Reserved	[15:12]	Reserved	0
ALPHA0_G_L	[11: 8]	Green Alpha lower value(case AEN == 0)	0
Reserved	[7: 4]	Reserved	0
ALPHA0_B_L	[3: 0]	Blue Alpha lower value(case AEN == 0)	0

NOTE: ALPHA0_R(G,B) [7:4] = ALPHA0_R(G,B)_H[3:0]@VIDOSD3C
ALPHA0_R(G,B) [3:0] = ALPHA0_R(G,B)_L[3:0]@VIDW3ALPHA0

6.69 WINDOW 3 ALPHA1 CONTROL REGISTER (VIDW3ALPHA1, R/W, ADDRESS = 0XEE00_ 021C)

VIDW3ALPHA1	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:16]	Reserved	0
ALPHA1_R_L	[19:16]	Red Alpha lower value(case AEN == 1)	0
Reserved	[15:12]	Reserved	0
ALPHA1_G_L	[11: 8]	Green Alpha lower value(case AEN == 1)	0
Reserved	[7: 4]	Reserved	0
ALPHA1_B_L	[3: 0]	Blue Alpha lower value(case AEN == 1)	0

NOTE: ALPHA1_R(G,B) [7:4] = ALPHA1_R(G,B)_H[3:0]@VIDOSD3C
ALPHA1_R(G,B) [3:0] = ALPHA1_R(G,B)_L[3:0]@VIDW3ALPHA1

6.70 WINDOW 4 ALPHA0 CONTROL REGISTER (VIDW4 ALPHA0, R/W, ADDRESS = 0XEE00_ 0220)

VIDW4ALPHA0	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA0_R_L	[19:16]	Red Alpha lower value(case AEN == 0)	0
Reserved	[15:12]	Reserved	0
ALPHA0_G_L	[11: 8]	Green Alpha lower value(case AEN == 0)	0
Reserved	[7: 4]	Reserved	0
ALPHA0_B_L	[3: 0]	Blue Alpha lower value(case AEN == 0)	0

NOTE: ALPHA0_R(G,B) [7:4] = ALPHA0_R(G,B)_H[3:0]@VIDOSD4C
ALPHA0_R(G,B) [3:0] = ALPHA0_R(G,B)_L[3:0]@VIDW4ALPHA0

6.71 WINDOW 4 ALPHA1 CONTROL REGISTER (VIDW0ALPHA1, R/W, ADDRESS = 0XEE00_ 0224)

VIDW0ALPHA1	Bit	Description	Reset Value
Reserved	[24]	Reserved	0
Reserved	[23:20]	Reserved	0
ALPHA1_R_L	[19:16]	Red Alpha lower value(case AEN == 1)	0
Reserved	[15:12]	Reserved	0
ALPHA1_G_L	[11: 8]	Green Alpha lower value(case AEN == 1)	0
Reserved	[7: 4]	Reserved	0
ALPHA1_B_L	[3: 0]	Blue Alpha lower value(case AEN == 1)	0

NOTE: ALPHA1_R(G,B) [7:4] = ALPHA1_R(G,B)_H[3:0]@VIDOSD4C
ALPHA1_R(G,B) [3:0] = ALPHA1_R(G,B)_L[3:0]@VIDW4ALPHA1

6.72 WINDOW 1 BLENDING EQUATION CONTROL REGISTER (BLENDEQ1, R/W, ADDRESS = 0XEE00_ 0244)

BLENDEQ1	Bit	Description	Reset Value
reserved	[31:22]	reserved	0x000
Q_FUNC	[21:18]	Constant used in alphaB (alpha value of *background) 0000 = 0 (zero) 0001 = 1 (max) 0010 = **alphaA (alpha value of *foreground) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 011x = reserved 100x = reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = reserved	0x0
reserved	[17:16]	Reserved	00
P_FUNC	[15:12]	Constant used in alpha Same as above (refer the COEF_Q)	0x0
reserved	[11:10]	Reserved	00
B_FUNC	[9:6]	Constant used in B Same as above (refer the COEF_Q)	0x3
reserved	[5:4]	Reserved	00
A_FUNC	[3:0]	Constant used in A Same as above (refer the COEF_Q)	0x2

NOTE: Refer to Figure 5. Blending equation
background = Window 0, foreground = Window 1 (in Blend Equation 1)
alphaA, alphaB is decided by BPPMODE_F, BLD_PIX, ALPHA_SEL @WINCONx and WxPAL @WPALCON.

6.73 WINDOW 2 BLENDING EQUATION CONTROL REGISTER (BLENDEQ2, R/W, ADDRESS = 0XEE00_0248)

BLENDEQ2	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0x000
Q_FUNC	[21:18]	Constant used in alphaB(alpha value of *background) 0000 =0 (zero) 0001 =1 (max) 0010 =**alphaA (alpha value of *foreground) 0011 =1 – alphaA 0100 =alphaB 0101 =1 – alphaB 011x =reserved 100x =reserved 1010 =A (foreground color data) 1011 =1 – A 1100 =B (background color data) 1101 =1 – B 111x = reserved	0x0
Reserved	[17:16]	Reserved	00
P_FUNC	[15:12]	Constant used in alphaA Same as above (refer the COEF_Q)	0x0
Reserved	[11:10]	Reserved	00
B_FUNC	[9:6]	Constant used in B Same as above (refer the COEF_Q)	0x3
Reserved	[5:4]	Reserved	00
A_FUNC	[3:0]	Constant used in A Same as above (refer the COEF_Q)	0x2

NOTE: Refer to Figure 5. Blending equation
background = Window 01, foreground = Window 2 (in Blend Equation 2)
alphaA, alphaB is decided by BPPMODE_F, BLD_PIX, ALPHA_SEL @WINCONx and WxPAL @WPALCON.

6.74 WINDOW 3 BLENDING EQUATION CONTROL REGISTER (BLENDEQ3, R/W, ADDRESS = 0XEE00_024C)

BLENDEQ3	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0x000
Q_FUNC	[21:18]	Constant used in alphaB(alpha value of *background) 0000 = 0 (zero) 0001 = 1 (max) 0010 = **alphaA (alpha value of *foreground) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 011x = reserved 100x = reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = reserved	0x0
Reserved	[17:16]	Reserved	00
P_FUNC	[15:12]	Constant used in alphaA Same as above (refer the COEF_Q)	0x0
Reserved	[11:10]	Reserved	00
B_FUNC	[9:6]	Constant used in B Same as above (refer the COEF_Q)	0x3
Reserved	[5:4]	Reserved	00
A_FUNC	[3:0]	Constant used in A Same as above (refer the COEF_Q)	0x2

NOTE: Refer to Figure 5. Blending equation
background = Window 012, foreground = Window 3 (in Blend Equation 3)
alphaA, alphaB is decided by BPPMODE_F, BLD_PIX, ALPHA_SEL @WINCONx and WxPAL @WPALCON.

6.75 WINDOW 4 BLENDING EQUATION CONTROL REGISTER (BLENDEQ4, R/W, ADDRESS = 0XEE00_ 0250)

BLENDEQ4	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0x000
Q_FUNC	[21:18]	Constant used in alphaB(alpha value of *background) 0000 = 0 (zero) 0001 = 1 (max) 0010 = **alphaA (alpha value of *foreground) 0011 = 1 – alphaA 0100 = alphaB 0101 = 1 – alphaB 011x = reserved 100x = reserved 1010 = A (foreground color data) 1011 = 1 – A 1100 = B (background color data) 1101 = 1 – B 111x = reserved	0x0
Reserved	[17:16]	Reserved	00
P_FUNC	[15:12]	Constant used in alphaA Same as above (refer the COEF_Q)	0x0
Reserved	[11:10]	Reserved	00
B_FUNC	[9:6]	Constant used in B Same as above (refer the COEF_Q)	0x3
Reserved	[5:4]	Reserved	00
A_FUNC	[3:0]	Constant used in A Same as above (refer the COEF_Q)	0x2

NOTE: Refer to Figure 5. Blending equation
background = Window 0123, foreground = Window 4 (in Blend Equation 4)
alphaA, alphaB is decided by BPPMODE_F, BLD_PIX, ALPHA_SEL @WINCONx and WxPAL @WPALCON.

6.76 BLENDING EQUATION CONTROL REGISTER (BLENDCON, R/W, ADDRESS = 0XEE00_ 0260)

BLENDCON	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x000
BLEND_NEW	[0]	Alpha value width 0 = 4-bit alpha value 1 = 8-bit alpha value	0x0

6.77 LCD I80 INTERFACE COMMAND (I80IFCONX)

- LDI_CMD0, R/W, Address = 0xEE00_ 0280
- LDI_CMD1, R/W, Address = 0xEE00_ 0284
- LDI_CMD2, R/W, Address = 0xEE00_ 0288
- LDI_CMD3, R/W, Address = 0xEE00_ 028C
- LDI_CMD4, R/W, Address = 0xEE00_ 0290
- LDI_CMD5, R/W, Address = 0xEE00_ 0294
- LDI_CMD6, R/W, Address = 0xEE00_ 0298
- LDI_CMD7, R/W, Address = 0xEE00_ 029C
- LDI_CMD8, R/W, Address = 0xEE00_ 02A0
- LDI_CMD9, R/W, Address = 0xEE00_ 02A4
- LDI_CMD10, R/W, Address = 0xEE00_ 02A8
- LDI_CMD11, R/W, Address = 0xEE00_ 02AC

I80IFCONx	Bit	Description	Reset Value
LDI_CMD	[23:0]	LDI command	0

6.78 WIN0 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Address	R/W	Description	Reset Value
00	0xEE00_2400 (0xEE00_0400)	R/W	Window 0 Palette entry 0 address	undefined
01	0xEE00_2404 (0xEE00_0404)	R/W	Window 0 Palette entry 1 address	undefined
-	-	-	-	-
FF	0xEE00_27FC (0xEE00_07FC)	R/W	Window 0 Palette entry 255 address	undefined

6.79 WIN1 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Address	R/W	Description	Reset Value
00	0xEE00_2800 (0xEE00_0800)	R/W	Window 1 Palette entry 0 address	undefined
01	0xEE00_2804 (0xEE00_0804)	R/W	Window 1 Palette entry 1 address	undefined
-	-	-	-	-
FF	0xEE00_2BFC (0xEE00_0BFC)	R/W	Window 1 Palette entry 255 address	undefined

6.80 WIN2 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Offset	R/W	Description	Reset Value
00	0xEE00_2C00	R/W	Window 2 Palette entry 0 address	undefined
01	0xEE00_2C04	R/W	Window 2 Palette entry 1 address	undefined
-	-	-	-	-
FF	0xEE00_2FFC	R/W	Window 2 Palette entry 255 address	undefined

6.81 WIN3 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Offset	R/W	Description	Reset Value
00	0xEE00_3000	R/W	Window 3 Palette entry 0 address	undefined
01	0xEE00_3004	R/W	Window 3 Palette entry 1 address	undefined
-	-	-	-	-
FF	0xEE00_33FC	R/W	Window 3 Palette entry 255 address	undefined

6.82 WIN4 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Register	Offset	R/W	Description	Reset Value
00	0xEE00_3400	R/W	Window 4 Palette entry 0 address	undefined
01	0xEE00_3404	R/W	Window 4 Palette entry 1 address	undefined
-	-	-	-	-
FF	0xEE00_37FC	R/W	Window 4 Palette entry 255 address	undefined

9.2 IMAGE ROTATOR

1 OVERVIEW

Image Rotator performs rotating/flipping image data. It is composed of Rotate FSM, Rotate Buffer, AMBA AHB 2.0 master/slave interface, and Register files. Overall features are summarized as follows.

2 FEATURES

The features of image rotator include:

- Image format: YCbCr 4:2:2(interleave), YCbCr 4:2:0(non-interleave), RGB565 and RGB888 (unpacked)
- Rotate degree: 90, 180, and 270, flip vertical and flip horizontal
- Image size: 64K by 64K

3 BLOCK DIAGRAM

The Figure 9.2-1 shows the block diagram of Image Rotator.

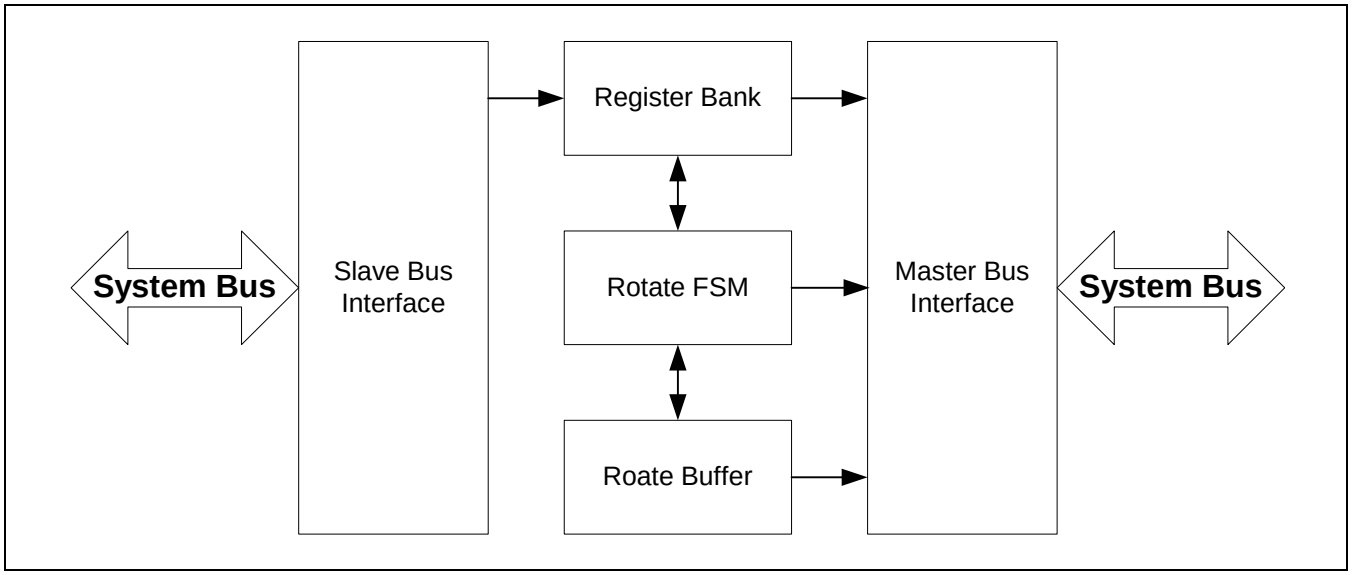


Figure 9.2-1 Image Rotator Block Diagram

3.1 ORIGINAL IMAGE

3.2 FLIP VERTICAL

3.3 FLIP HORIZONTAL

3.4 180-DEGREE ROTATION

3.5 90 AND 270-DEGREE ROTATION

4 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
CTRLCFG	0xEE10_0000	R/W	Rotator Control Register	0x000_0000
SRCADDRREG0	0xEE10_0004	R/W	Rotator Source Image Address Register	0x0000_0000
SRCADDRREG1	0xEE10_0008	R/W	Rotator Source Image Address Register	0x0000_0000
SRCADDRREG2	0xEE10_000C	R/W	Rotator Source Image Address Register	0x0000_0000
SRCSIZEREG	0xEE10_0010	R/W	Rotator Source Image Size Register	0x0000_0000
DESTADDRREG0	0xEE10_0018	R/W	Rotator Destination Image Address Register	0x0000_0000
DESTADDRREG1	0xEE10_001C	R/W	Rotator Destination Image Address Register	0x0000_0000
DESTADDRREG2	0xEE10_0020	R/W	Rotator Destination Image Address Register	0x0000_0000
STATCFG	0xEE10_002C	R	Rotator Status Register	0x0000_0000

4.1 ROTATOR CONTROL REGISTER(CTRLREG, R/W, ADDRESS= 0XEE10_0000)

CTRLREG	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	000_0000b
Enable Int.	[24]	Interrupt Enable 0 = Disables interrupt 1 = Enables interrupt	0b
Reserved	[23:16]	Reserved	0x00
Input Image format	[15:13]	Input image format to be rotated 000 = YCbCr 4:2:0(non-interleave) 001 = Reserved 010 = Reserved 011 = YCbCr 4:2:2(interleave) 100 = RGB 565 101 = RGB888 (unpacked)	000b
Reserved	[12:8]	Reserved	0_0000b
Rotation Degree	[7:6]	Determines the rotation degree. 00 = Does not rotate 01 = 90 degree 10 = 180 degree 11 = 270 degree1 **Note:** These bits should be zero, if flip direction is not zero.	00b
Flip Direction	[5:4]	Determines the flip direction 00 = Does not flip 10 = Flip vertical 11 = Flip horizontal **Note:** These bits should be zero, if rotation degree is not zero.	00b
Reserved	[3:1]	Reserved	000b
Start rotate	[0]	Rotate enable signal. If this bit is set, Rotator starts the operation. This bit is cleared if rotator starts to move image. 0 = Does not work 1 = Start rotate operation	0b

4.2 ROTATOR SOURCE IMAGE ADDRESS REGISTER 0 (RGB OR Y COMPONENT) (SRCADDRREG, R/W, ADDRESS = 0XEE10_0004)

SRCADDRREG0	Bit	Description	Reset Value
Source Address	[31:0]	Address of source image.	0x0000_0000

4.3 ROTATOR SOURCE IMAGE ADDRESS REGISTER 1 (CB COMPONENT) (SRCADDREG, R/W, ADDRESS = 0XEE10_0008)

SRCADDRREG1	Bit	Description	Reset Value
Source Address	[31:0]	Address of source image.	0x0000_0000

4.4 ROTATOR SOURCE IMAGE ADDRESS REGISTER 2 (CR COMPONENT) (SRCADDRREG2, R/W, ADDRESS = 0XEE10_000C)

SRCADDRREG2	Bit	Description	Reset Value
Source Address	[31:0]	Address of source image.	0x0000_0000

4.5 ROTATOR SOURCE IMAGE SIZE REGISTER (SRCSIZEREG, R/W, ADDRESS = 0XEE10_0010)

SRCSIZEREG	Bit	Description	Reset Value
Vertical Size	[31:16]	Vertical Image size of source image	0x0000
Horizontal Size	[15:0]	Horizontal Image size of source image	0x0000

4.6 ROTATOR DESTINATION IMAGE ADDRESS REGISTER 0 (RGB OR Y COMPONENT) (CONFIGREG0, R/W, ADDRESS = 0XEE10_0018)

CONFIGREG0	Bit	Description	Reset Value
Destination Address	[31:0]	Address of destination image.	0x0000_0000

4.7 ROTATOR DESTINATION IMAGE ADDRESS REGISTER 1 (CB COMPONENT) (CONFIGREG1, R/W, ADDRESS = 0XEE10_001C)

CONFIGREG1	Bit	Description	Reset Value
Destination Address	[31:0]	Address of destination image.	0x0000_0000

4.8 ROTATOR DESTINATION IMAGE ADDRESS REGISTER 2 (CR COMPONENT) (CONFIGREG2, R/W, ADDRESS = 0XEE10_0020)

CONFIGREG2	Bit	Description	Reset Value
Destination Address	[31:0]	Address of destination image.	0x0000_0000

4.9 ROTATOR STATUS REGISTER (STATREG, R, ADDRESS = 0XEE10_002C)

STATREG	Bit	Description	Reset Value
Current line number	[31:16]	Indicates from where rotator accesses image. This value shows the line number of image handled.	0
Reserved	[15:9]	Reserved	0x00
Interrupt Pending	[8]	This bit is set if an image rotation is complete. Writing '1' makes this bit clear.	0
Reserved	[7:1]	Reserved	0x00
Rotator status	[0]	This bits show the rotator operation status. 00 =No work in progress (IDLE) 01 = Reserved 10 = Rotating a image (BUSY) 11 = Rotating a image, and has one more job to rotate (BUSY)	0

9.3 CAMERA INTERFACE

1 OVERVIEW

This specification defines the interface of camera. The Camera Interface in S5PC100X (Fully Interactive Mobile Camera interface 4.0) supports ITU R BT-601/656 standard, AXI interface and MIPI (CSI). Maximum input size is 8192 x 8192 pixels. S5PC100x has three CAMERA Interface units as illustrated in Figure 9.3-1. Each of them consists of several functions. *T_patternMux* generates test pattern. Test pattern generation is used to calibrate input sync signals as HREF, and VSYNC. *Capture* is the capturing signal and window cut. Use register setting to invert Video sync signals and pixel clock polarity in the camera interface side. *Scaler* generates various sizes for useful image. *Input DMA (read only)* reads from the memory image data. *Output DMA (write only)* writes image data to memory. CAMIF has *image rotator (90' clockwise)* and *image effect*. These features are very useful in folder type cellular phone.

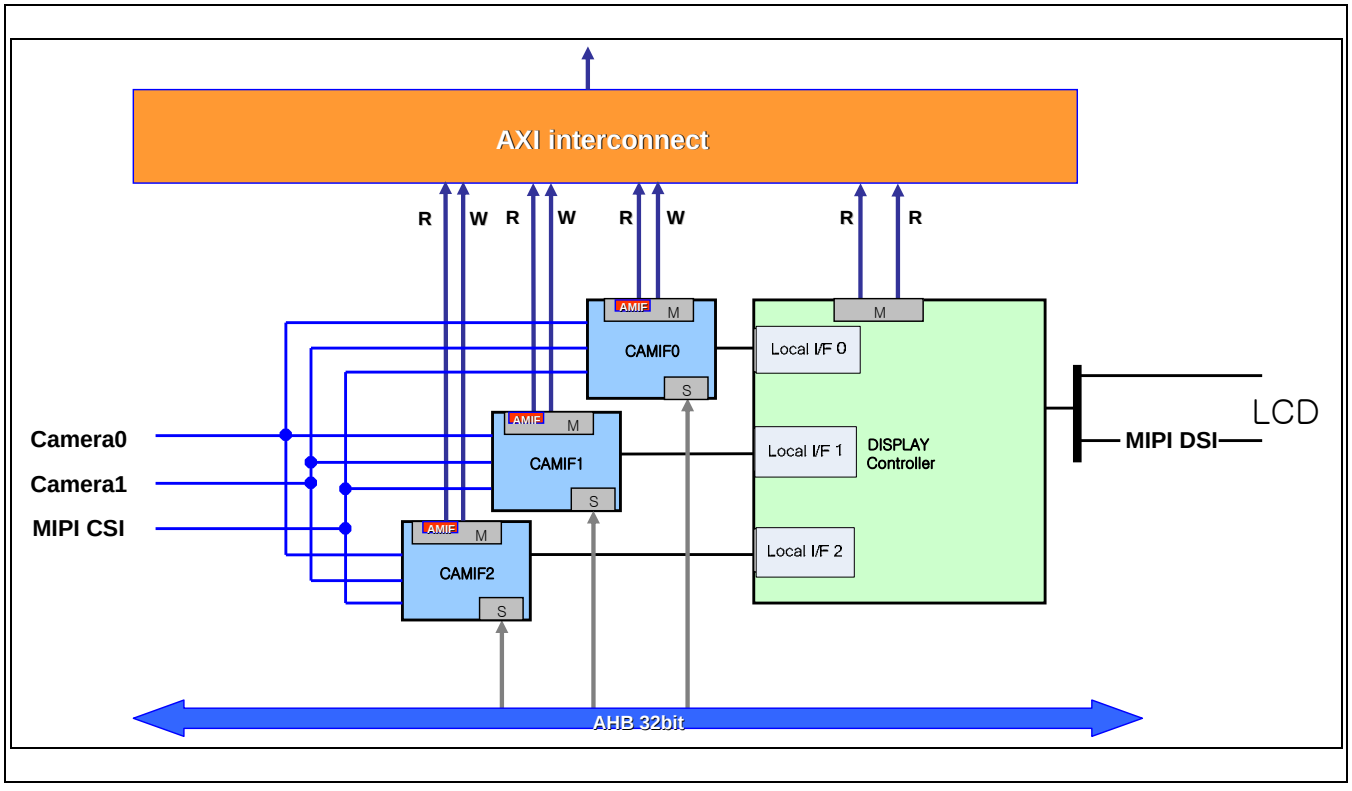


Figure 9.3-1 Sub_block of Visual System in S5PC100x

2 FEATURES

The features of CAMIF include:

- Multiple input support
  - ITU-R BT 601/656 mode
  - DMA (AXI 64bit interface) mode
  - MIPI (CSI) mode
- Multiple output support
  - DMA (AXI 64bit interface) mode
  - Direct FIFO mode
- Digital Zoom In (DZI) capability
- Multiple camera input
- Programmable polarity of video sync signals
- Supports maximum 8192 x 8192 pixels input
- Image mirror and rotation (X-axis mirror, Y-axis mirror, 90°, 180° and 270° rotation)
- Generates various image format
- Supports capture frame control
- Supports image effect

Table 9.3-1 Maximum Size (Refer to Appendix at end of the Chapter)

	Item	Max Size		
		CAMIF0	CAMIF1	CAMIF2
Scaler	Scaler input Hsize (=PreDstWidth)	3264 pixels	1280 pixels	1440 pixels
	Scaler bypass mode	8192 pixels	8192 pixels	8192 pixels
Output Rotator	TargetHsize (without output rotation)	3264 pixels	1280 pixels	NA
	TargetHsize (with output rotation)	1280 pixels	768 pixels	NA
Input Rotator	REAL_WIDTH (without input rotation)	8192 pixels	8192 pixels	NA
	REAL_HEIGHT (with input rotation)	1280 pixels	768 pixels	NA

NOTE: Scaler and Rotator maximum size depends on line buffer size.

NOTE: The max size of the Output rotator & input rotator is under the scaler on mode.

NOTE: Rotator memory shares CAMIF and Low-Power Audio. If you want to use rotator function, you must set the value of LPMP3_MODE_SEL@LPMP3_MODE_SEL (0xE020_0308) as '0'. For more information refer to "*Chapter 2.3 CLOCK STRATEGY*".

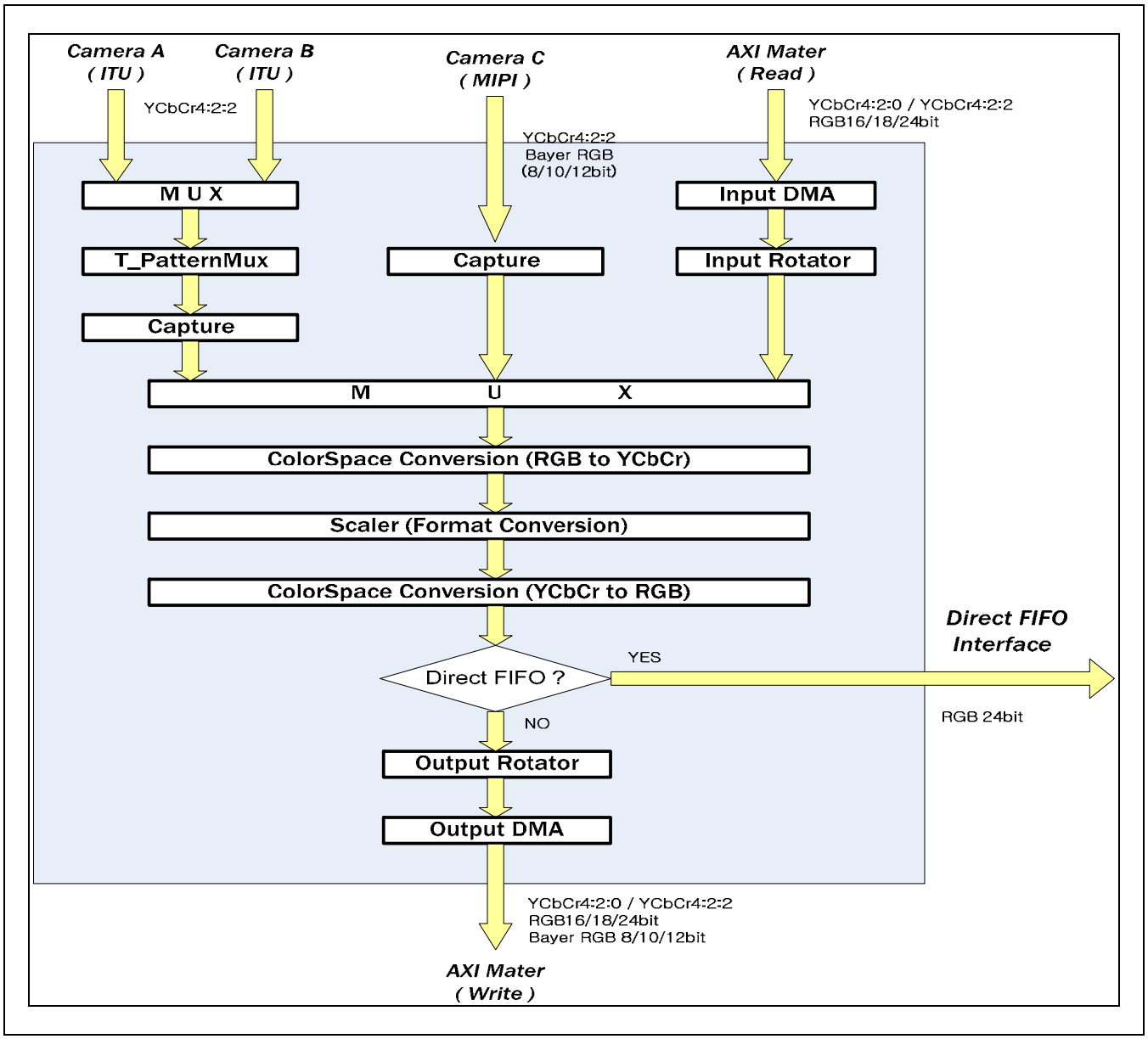


Figure 9.3-2 Camera Interface Overview

3 EXTERNAL INTERFACE

Camera Interface supports the next video standards. Two video standards are as follows:

- ITU-R BT 601 YCbCr 8-bit mode
- ITU-R BT 656 YCbCr 8-bit mode
- MIPI mode

4 I/O DESCRIPTION

Table 9.3-2 Camera Interface Signal Description

Function Signal	I/O	Description	PAD	Type
External Camera Processor Interface Signal				
CAM_A_PCLK	I	Pixel Clock, driven by the external Camera processor A	XciPCLK	Muxed
CAM_A_VSYNC	I	Frame Sync, driven by the external Camera processor A	XciVSYNC	Muxed
CAM_A_HREF	I	Horizontal Sync, driven by the external Camera processor A	XciHREF	Muxed
CAM_A_D[7:0]	I	Pixel Data driven by the external Camera processor A	XciD[7:0]	Muxed
CAM_A_RESET	O	Software Reset or Power Down for the external Camera processor A	XciRESET	Muxed
CAM_A_FIELD	O	FIELD signal, driven by the external Camera processor A	XciFIELD	Muxed
CAM_A_CLKOUT	O	Clock for a external ISP	XciCLKenb	Muxed
External video player interface signal				
CAM_B_D[7:0]	I	Pixel Data driven by the external Camera processor B	XEINT[23:16]	Muxed
CAM_B_PCLK	I	Pixel Clock, driven by the external Camera processor B	XEINT[24]	Muxed
CAM_B_VSYNC	I	Frame Sync, driven by the external Camera processor B	XEINT[25]	Muxed
CAM_B_HREF	I	Horizontal Sync, driven by the external Camera processor B	XEINT[26]	Muxed
CAM_B_FIELD	I	FIELD signal, driven by the external Camera processor B	XEINT[27]	Muxed

NOTE: I/O direction. I: input, O: output, B: bi-direction

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

NOTE: Set CAM_MUX_SEL(0xE020_0300) to designate source of RESET_A and FIELD_A. The source of those signals can be CAMIF0 or CAMIF1 or CAMIF2. For more information refer to "*Chapter 2.3 CLOCK CONTROL"*.

5 TIMING DIAGRAM OF CAMERA

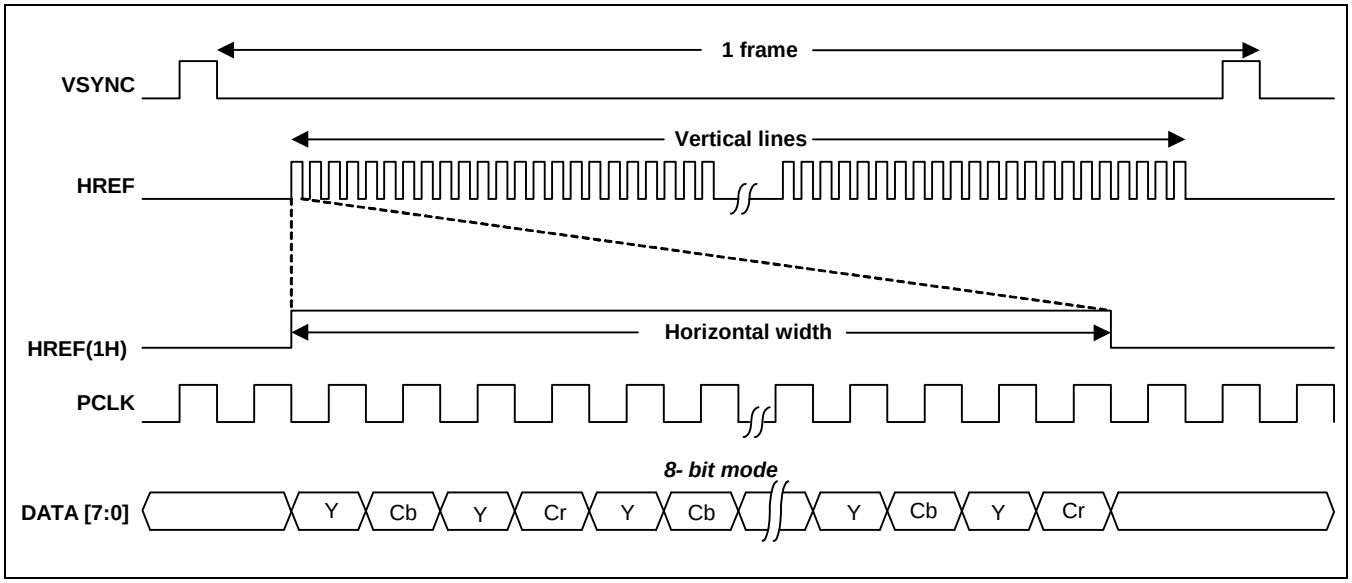


Figure 9.3-3 ITU-R BT 601 Input Timing Diagram

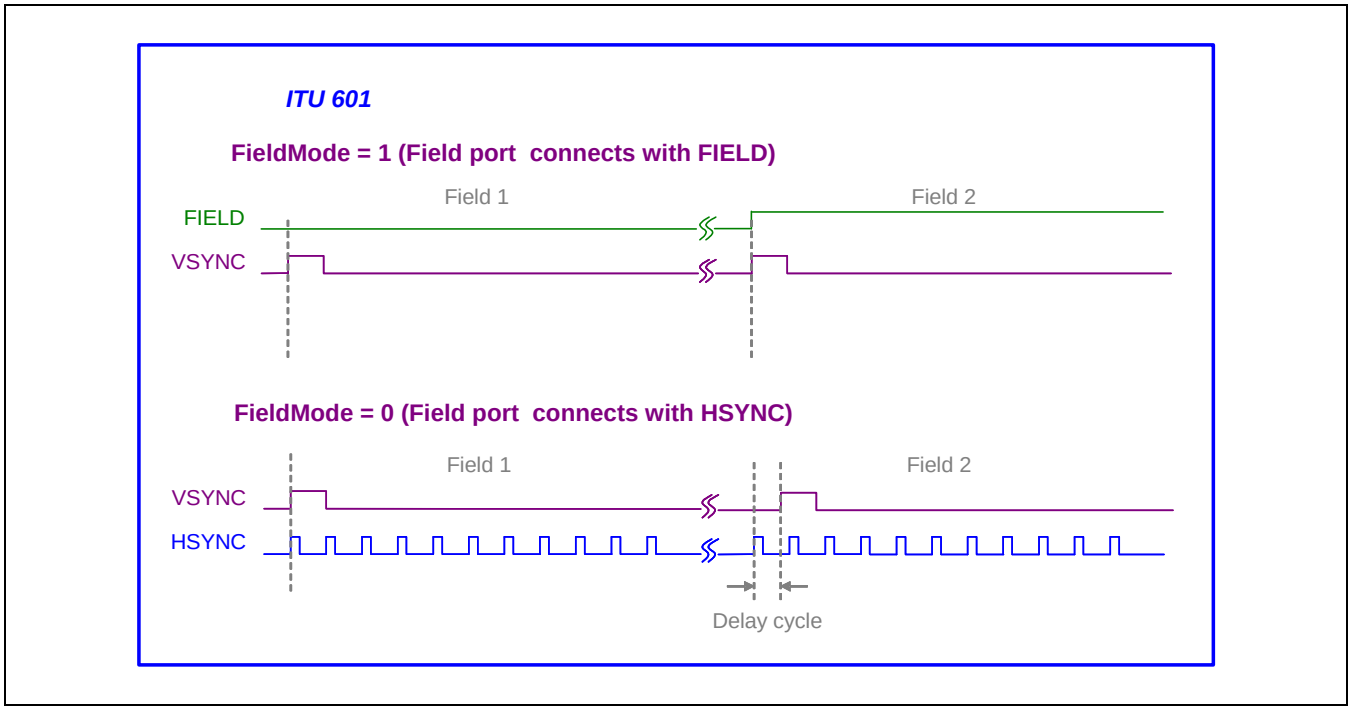


Figure 9.3-4 ITU-R BT 601 Interlace Handling Diagram

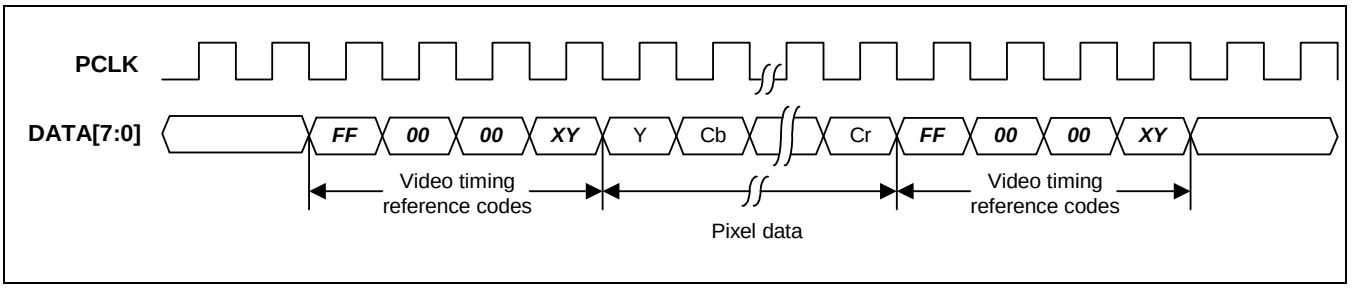


Figure 9.3-5 ITU-R BT 656 Input Timing Diagram

There are two timing reference signals in ITU-R BT 656 format, one at the beginning of each video data block (start of active video, SAV) and one at the end of each video data block(end of active video, EAV) as shown in Figure 9.3-5 and Table 9.3-3

Table 9.3-3 Video Timing Reference Codes of ITU-656 8Bit Format

Data bit number	First word	Second word	Third word	Fourth word
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2
1	1	0	0	P1
0	1	0	0	P0

NOTE: F = 0 (during field 1), 1 (during field 2)
V = 0 (elsewhere), 1 (during field blanking)
H = 0 (in SAV : Start of Active Video), 1 (in EAV : End of Active Video)
P0, P1, P2, P3 = protection bit
Camera interface logic catches the video sync bits like H (SAV, EAV) and V (Frame Sync) after reserved data as "FF-00-00".

Caution! All external camera interface IO's must not be combined with any other GPIO or bi-directional ports.

Caution! All external camera interface IO's are recommended to be shmitt-trigger type IO for noise reduction.

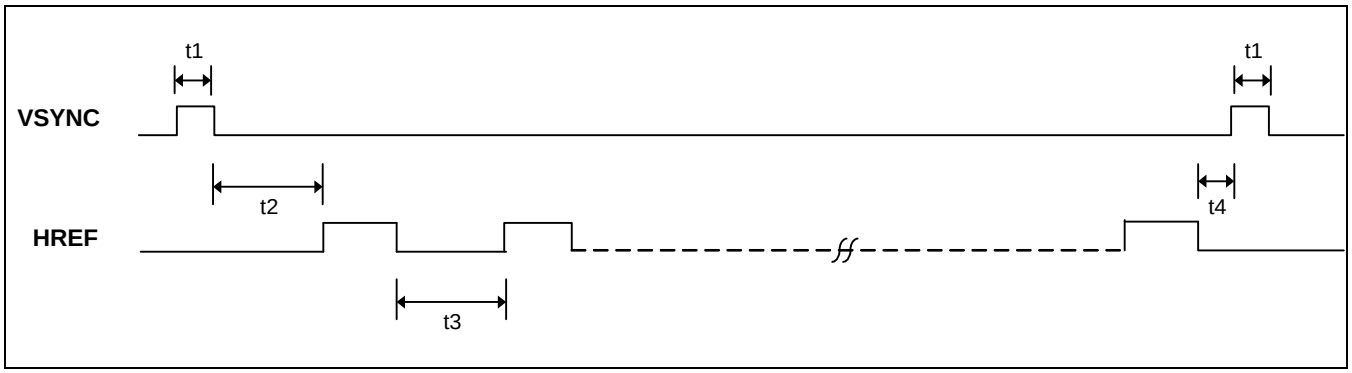


Figure 9.3-6 Sync Signal Timing Diagram

Table 9.3-4 Sync Signal Timing Requirement

	Minimum	Maximum
t1	12 cycles of Pixel clock	-
t2	12 cycles of Pixel clock	-
t3	2 cycles of Pixel clock	-
t4	12 cycles of Pixel clock	-

NOTE: If rotator is enabled, (t4 + t1) must be long enough to finish DMA transactions. It is because, DMA transaction for rotator line buffer are delayed by 4 or 8 horizontal lines.

6 EXTERNAL CONNECTION GUIDE

All Camera Interface input signals must not occur inter-skewing to pixel clock line. Recommend next pin location and routing.

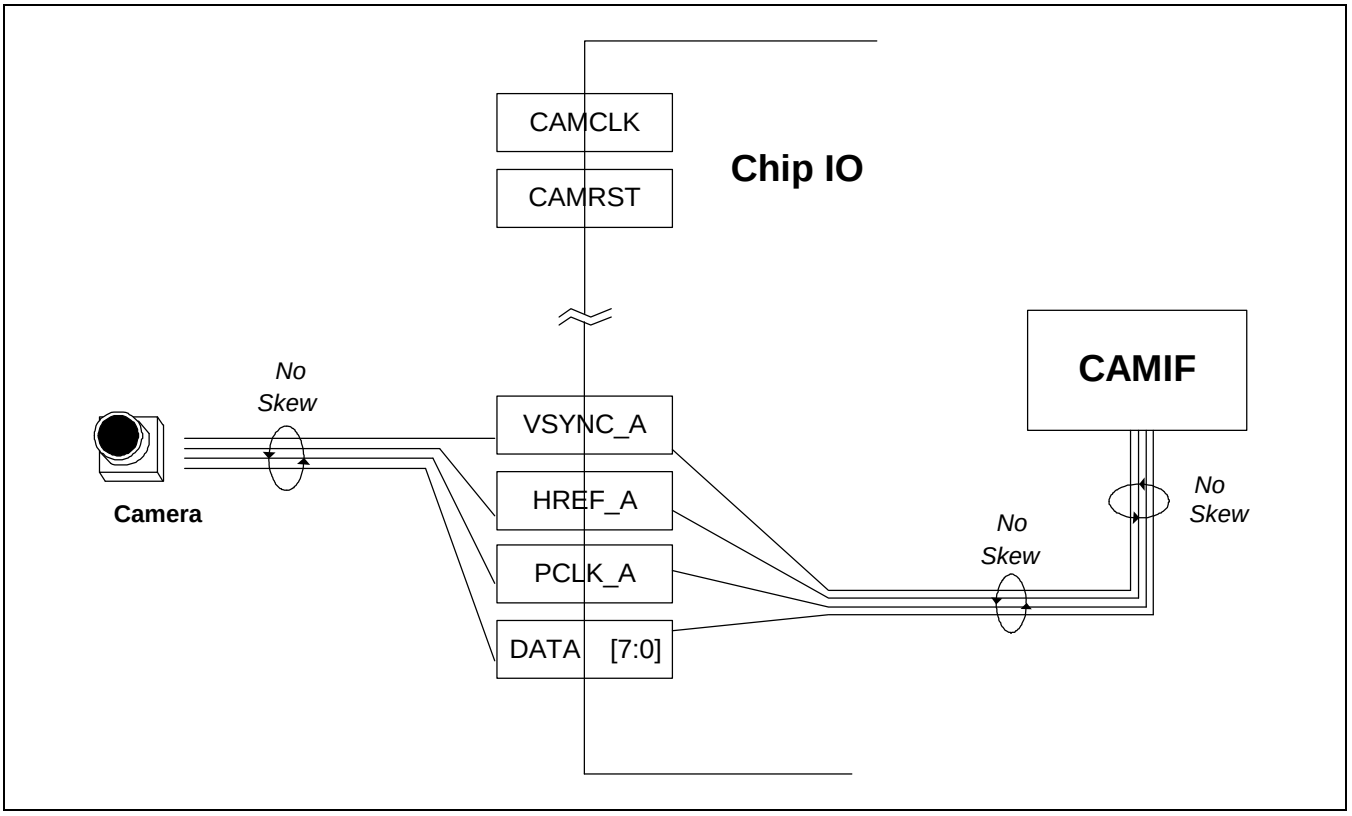


Figure 9.3-7 IO Connection Guide

7 CAMERA INTERFACE OPERATIONS

7.1 TWO DMA PORTS

CAMIF has two DMA ports. At the view of system bus, two ports are independent. The Input DMA port read the image data from memory. The Output DMA port stores the image data into memory. These two master ports support the variable applications like Digital Steel Camera (DSC), MPEG-4 video conference, video recording, etc.

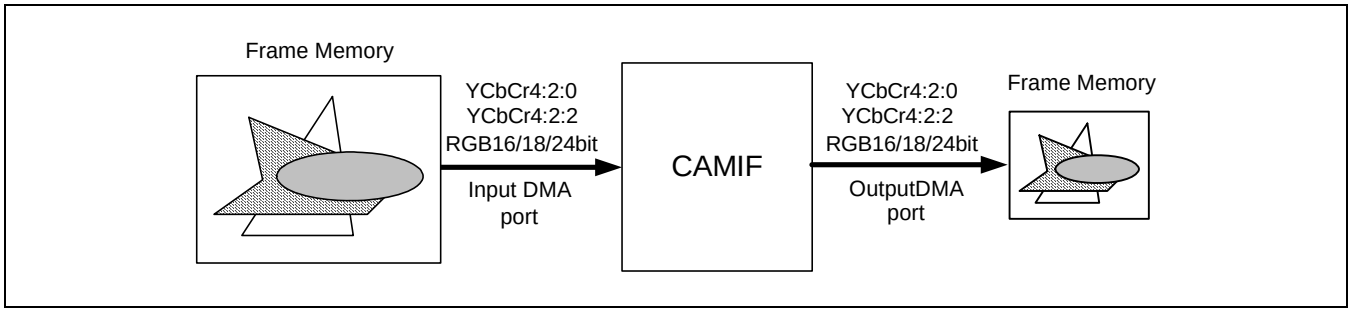


Figure 9.3-8 Two DMA Ports

7.2 CLOCK DOMAIN

CAMIF has three clock domains. The first clock is the system bus clock. The second clock is the camera pixel clock, which is PCLK. The third clock is the internal core clock. *The system clock must be faster than pixel clock*. As shown in Figure 9.3-9. CAMCLK must be divided from the fixed frequency like PLL clock. If external clock oscillator is used, CAMCLK should be floated. It is not necessary for three clock domains to be synchronized. Other signals as PCLK should be similarly connected to Schmitt-triggered level shifter.

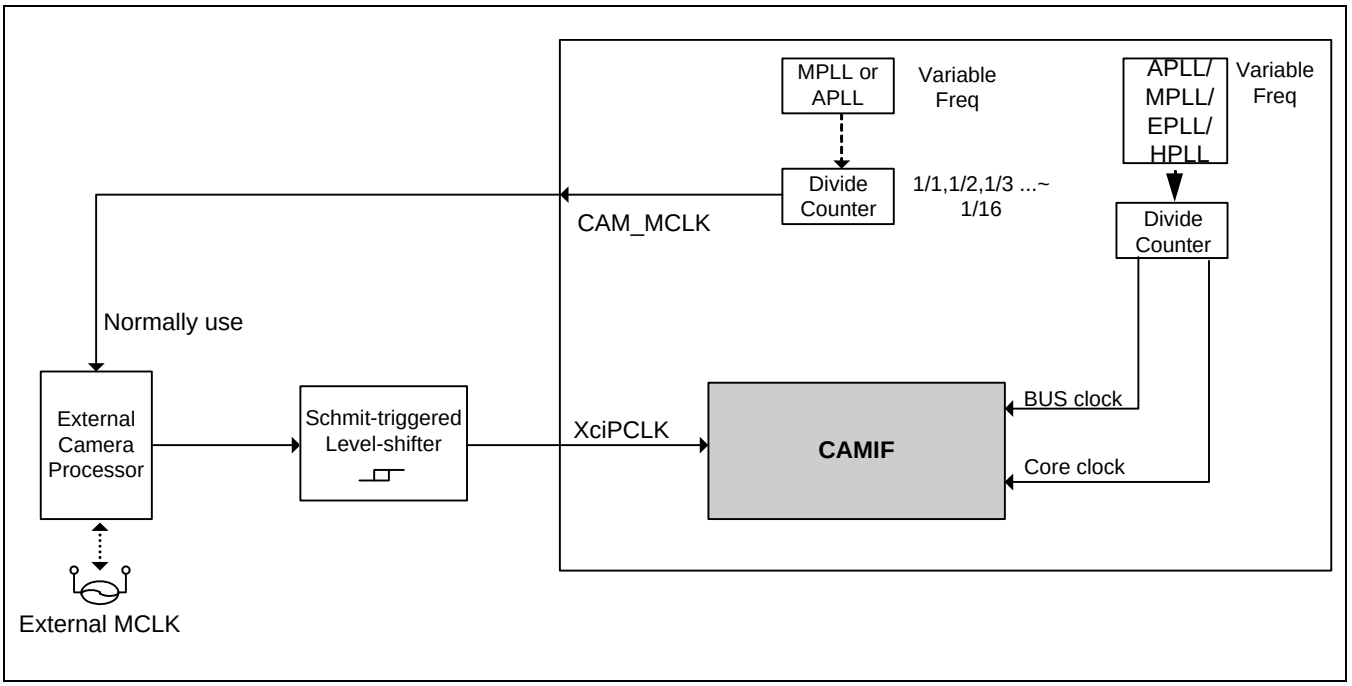


Figure 9.3-9 CAMIF Clock Generation

NOTE: The maximum frequency of core clock is 133MHz. MPLL is recommended for the source of core clock.

The maximum frequency of CAM_MCLK is 83MHz. The maximum frequency of XciPCLK is also 83MHz.

7.3 FRAME MEMORY HIERARCHY

Frame memories consist of four ping-pong memories for output DMA ports. Ping-pong memories have three element memories that are luminance Y, chrominance Cb, and chrominance Cr. It is recommended that the arbitration priority of CAMIF must be higher than any other masters except LCD controller. It is strongly recommended that CAMIF priorities should be the fixed priorities, not rotation priorities. In multi-AHB bus case, the priority of system bus including CAMIF must be higher than others. If bus is traffic enough that DMA operation is not ending during one horizontal period plus blank, it might mal-function. Therefore, the priority of CAMIF must be separated to other round robin or circular arbitration priorities. It is recommended that bus which includes CAMIF, should have higher priority than any other buses in memory matrix system. The CAMIF should not be the default master of AMBA system

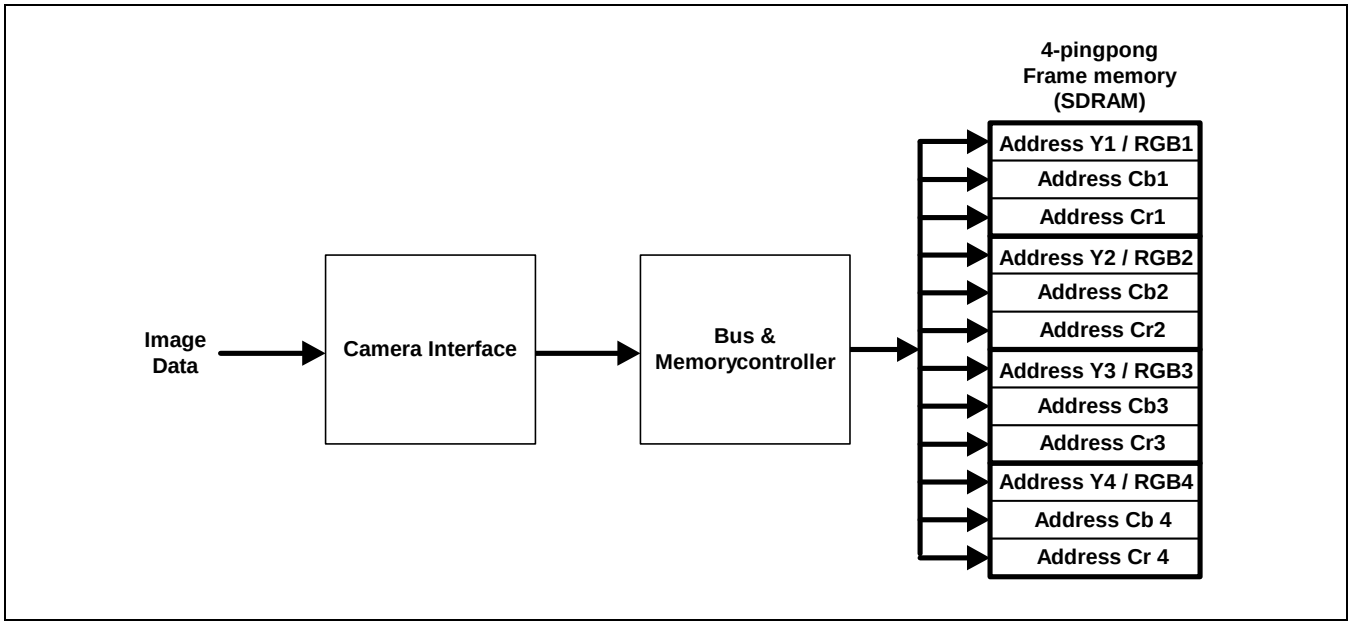


Figure 9.3-10 Ping-pong Memory Hierarchy

7.4 MEMORY STORING METHOD

The storing method to the frame memory is the little-endian method. The first entering pixels stored into LSB sides, and the last entering pixels stored into MSB sides. The carried data by AXI bus is 64-bit. Therefore, CAMIF make the each Y-Cb-Cr words by little endian style. Refer to Figure 9.3-11.

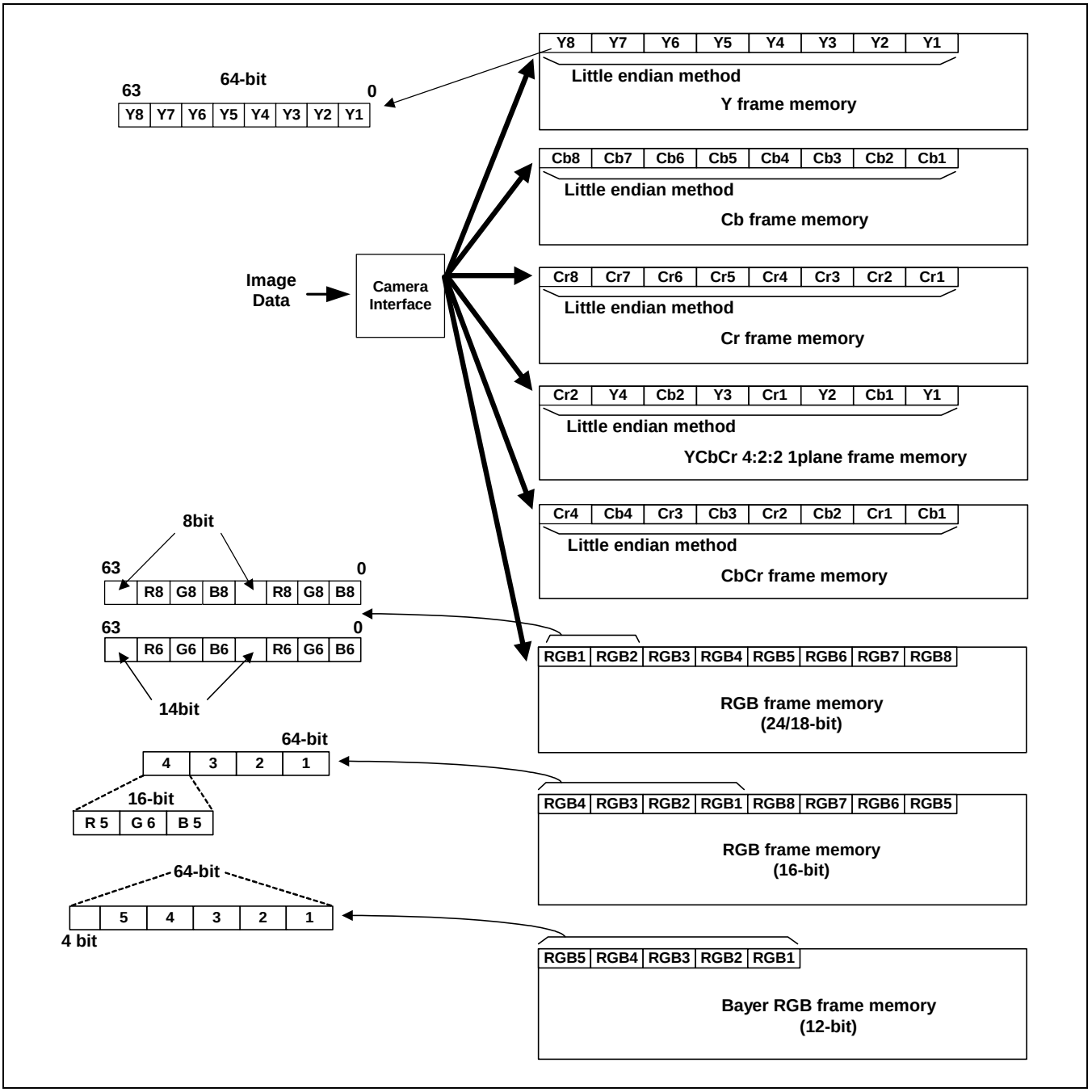
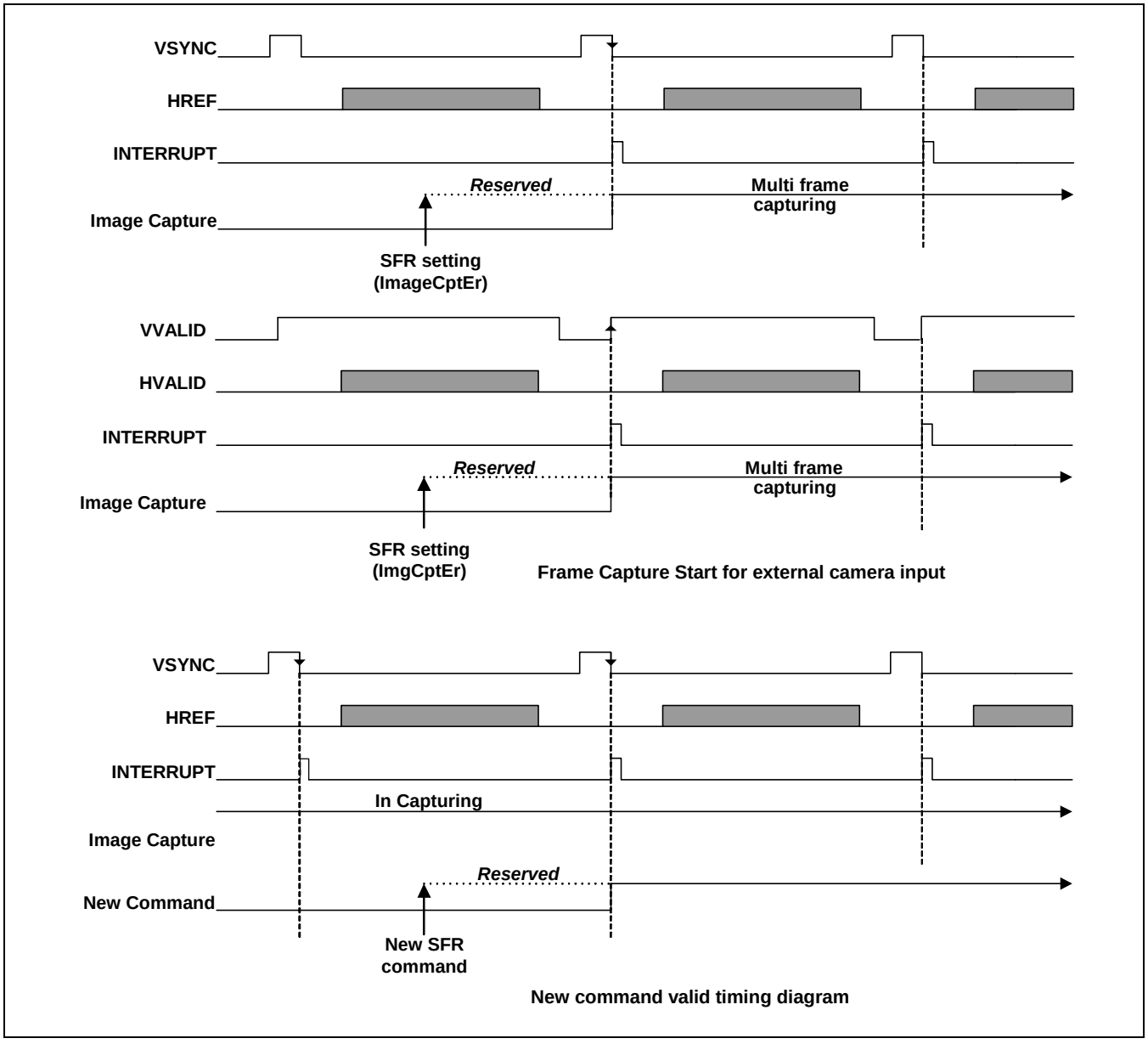


Figure 9.3-11 Memory Storing Style

7.5 TIMING DIAGRAM FOR REGISTER SETTING

The first register setting for frame capture command occurs anywhere in frame period. It is recommended to do first set the VSYNC "L" state, Input DMA start "L" state and VVALID "H" state. VSYNC and VVALID information can be read from status SFR. Refer to Figure 9.3-12. All command including ImgCptEn, is valid at VSYNC falling edge or VVALID rising edge. Make sure that except first SFR setting; all command should be programmed in Interrupt Service Routine (ISR). Size, image mirror or rotation, windowing, and Zoom In settings are allowed to change in capturing operation. In case od DMA input mode, all command should be programmed after InputDMA and OutputDMA operation end.



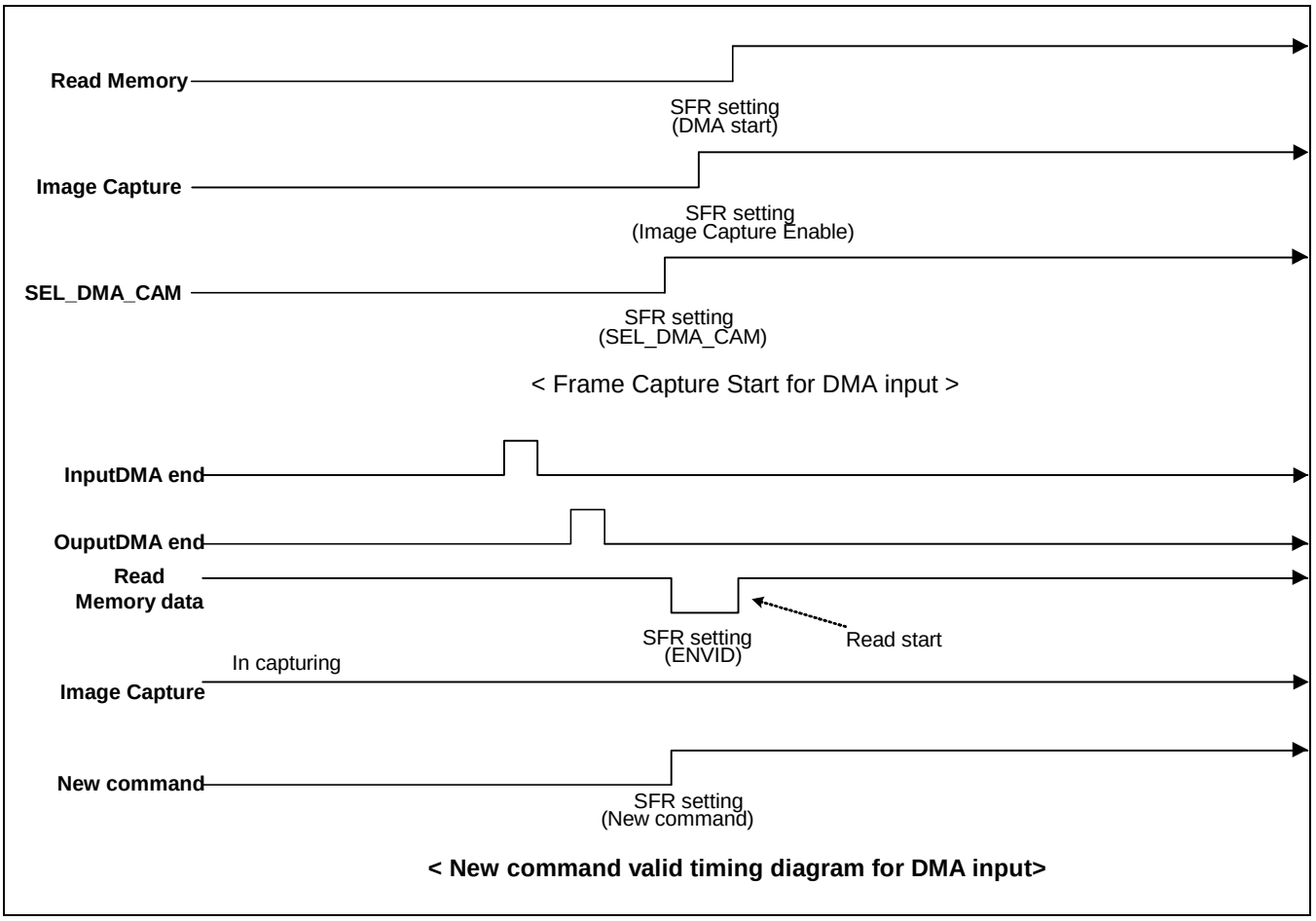


Figure 9.3-12 Timing Diagram for Register Setting

7.6 TIMING DIAGRAM FOR LAST IRQ

IRQ except LastIRQ is generated before image capturing. Last IRQ which means camera signal capture-end can be set by following timing diagram. LastIRQEn is ISR setting for next frame command. Therefore, for adequate last IRQ, you should follow next sequence between LastIRQEn and ImgCptEn/ImgCptEn_Sc. It is recommended that ImgCptEn/ImgCptEn_SC is set at same time and at last of SFR setting in ISR. FrameCnt which is read in ISR means next frame count. On following diagram, last captured frame count is "1." That is, Frame 1 is the last-captured frame among frame 0~3. FrameCnt is increased by 1 at IRQ rising. DMA input can be selected by SFR setting. In this case, IRQ is generated after Output DMA operation done per frame. This mode is aware of starting point by user's SFR setting (ENVID_M '0' → '1'). Therefore, this mode does not need IRQ of starting point and LastIRQ. FrameCnt is increased by 1 at ENVID_M (InputDMA start) low to rising ('0' → '1') and ImgCptEn_SC '1'

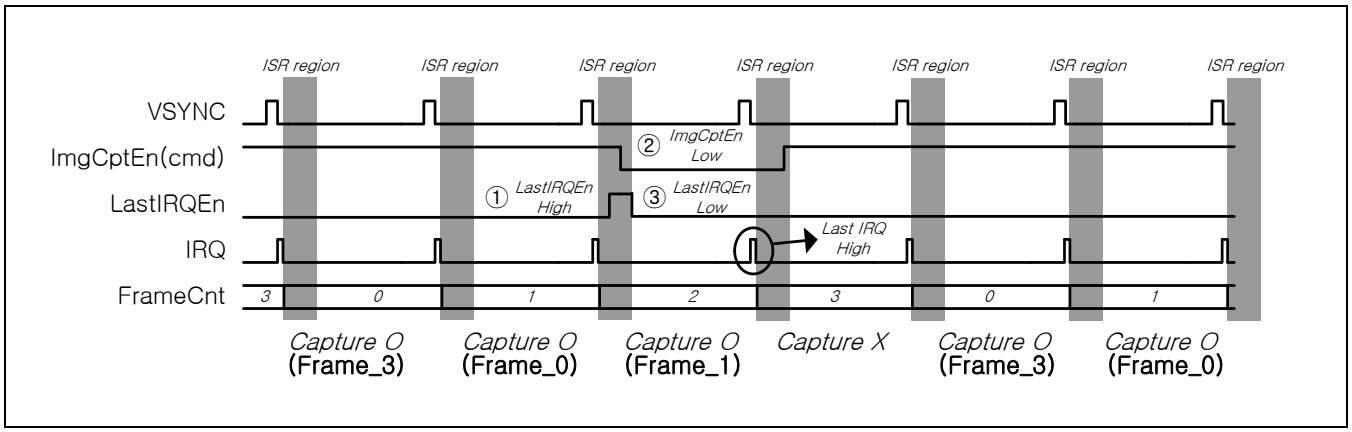


Figure 9.3-13 Timing Diagram for Last IRQ (LastIRQEn is Enabled)

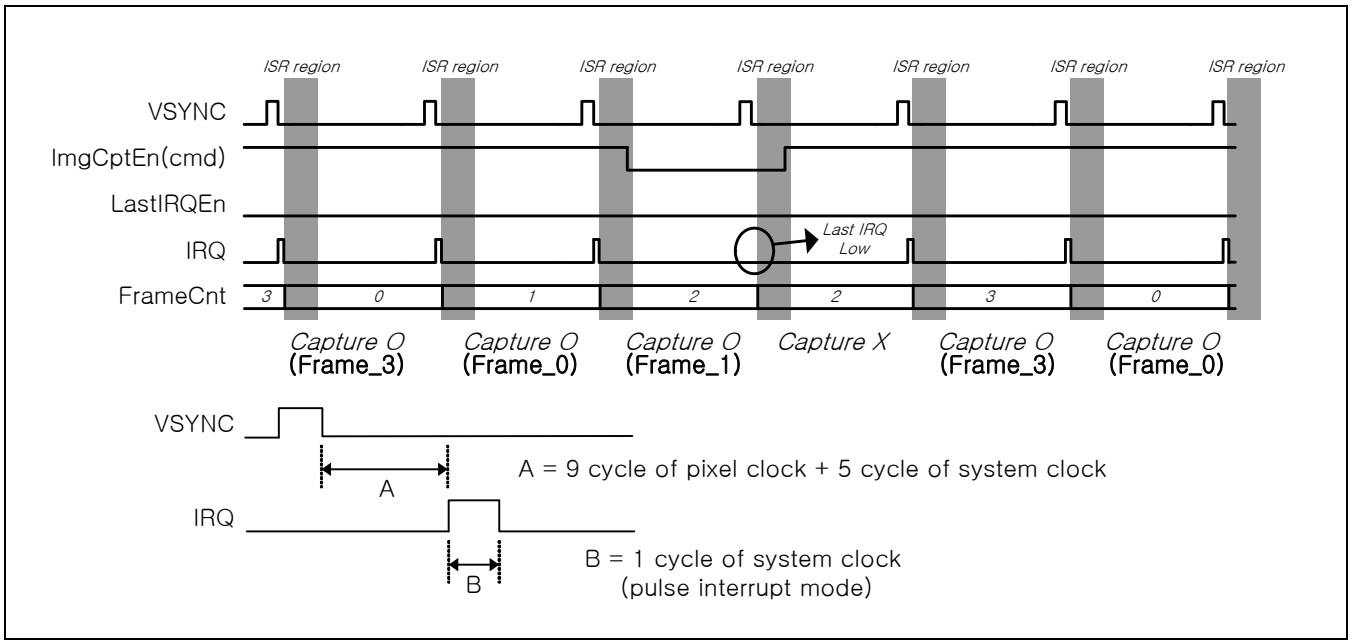


Figure 9.3-14 Diagram for Last IRQ (LastIRQEn is Disabled) and Timing Requirement

7.7 TIMING DIAGRAM FOR IRQ (MEMORY DATA SCALING MODE)

Input DMA can be selected by SFR setting. In this case, after DMA operation is complete for per frame, generates IRQ. This mode is aware of starting point by user's SFR setting (ENVID_M '0' → '1'). Therefore, this mode does not required IRQ of starting point and LastIRQ. FrameCnt is increased by 1 at ENVID_M low to rising ('0' → '1') and ImgCptEn_SC '1'

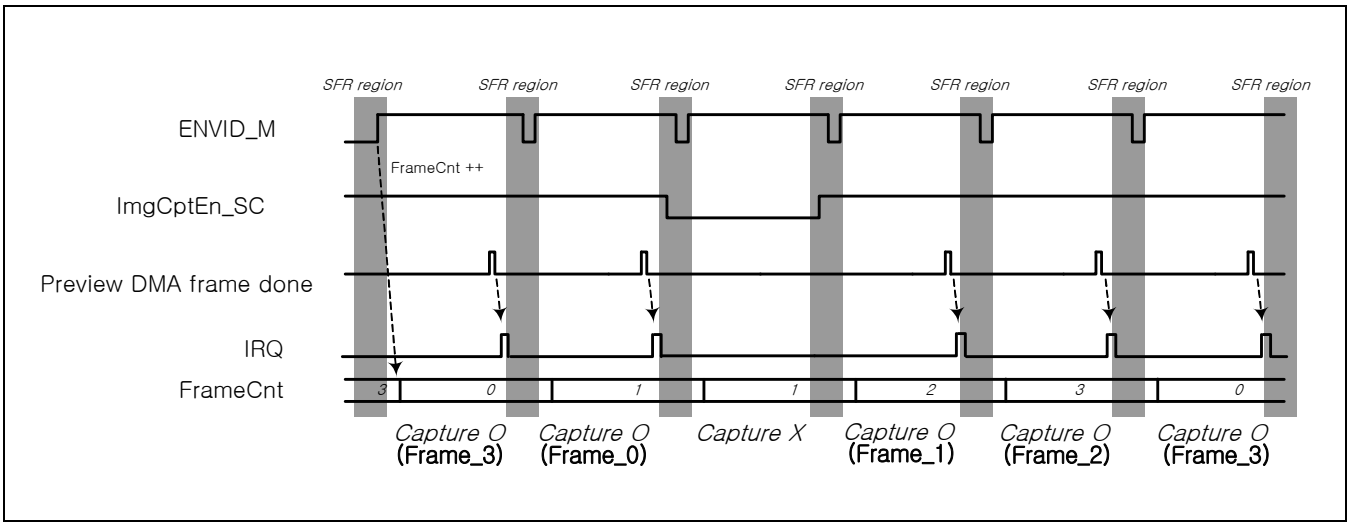


Figure 9.3-15 Timing Diagram for IRQ (Input DMA Path)

7.8 INPUT DMA FEATURE

Input DMA supports memory data scaling. Especially two different image data is required for Picture-in-Picture (PIP) operation. First image is saved memory by some codec (H.264, Camera, MPEG4 etc.) Second image is saved memory through input DMA path. The input DMA path has YCbCr/RGB output format through scaler/ DMA path. Two images are displayed and controlled by LCD controller. If input DMA is (reading the memory data) required to use in the path. SEL_DMA_CAM (MSCTRL bit [3]) signal must be set '1'. This input path is called Memory Scaling DMA path. This path is not allowed windowing zoom function.

NOTE:

Memory image format for input DMA input are:
YCbCr 4:2:0 (non-interleave)
YCbCr 4:2:2 (non-interleave)
YCbCr 4:2:2 (Interleave)
RGB

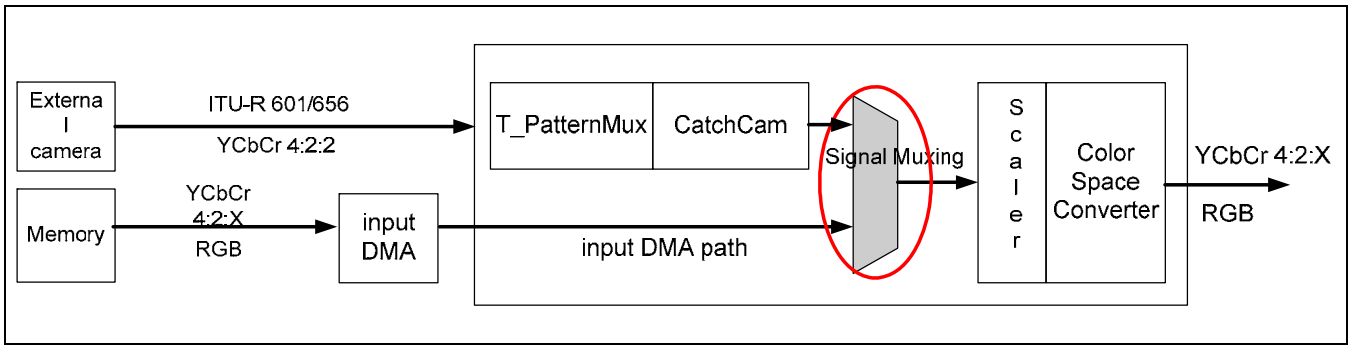


Figure 9.3-16 Input DMA or External Camera Interface

7.9 CAMERA INTERLACE INPUT SUPPORT

S5PC100 supports ITU-R BT 601 YCbCr 8 bit mode and ITU-R BT 656 YCbCr 8-bit mode in order to get data from external camera. S5PC100x supports not only progressive input but also progressive input in both modes

7.9.1 Progressive Input

In progressive mode, all the input data is stored in four buffers (Ping pong memory which is designated by SFR) sequentially by the unit of frame. For more information refer to Figure 9.3-14.

7.9.2 Interlaced Input

In interlace mode, the input data is stored in four buffers (Ping pong memory which is designated by SFR). In this mode, even field frame data and odd field frame data are stored in turn. Therefore even field frame data is stored in 1st and 3rd ping pong memory while odd field frame data is stored 2nd and 4th ping pong memory. In case of image capture, start frame is always even field frame.

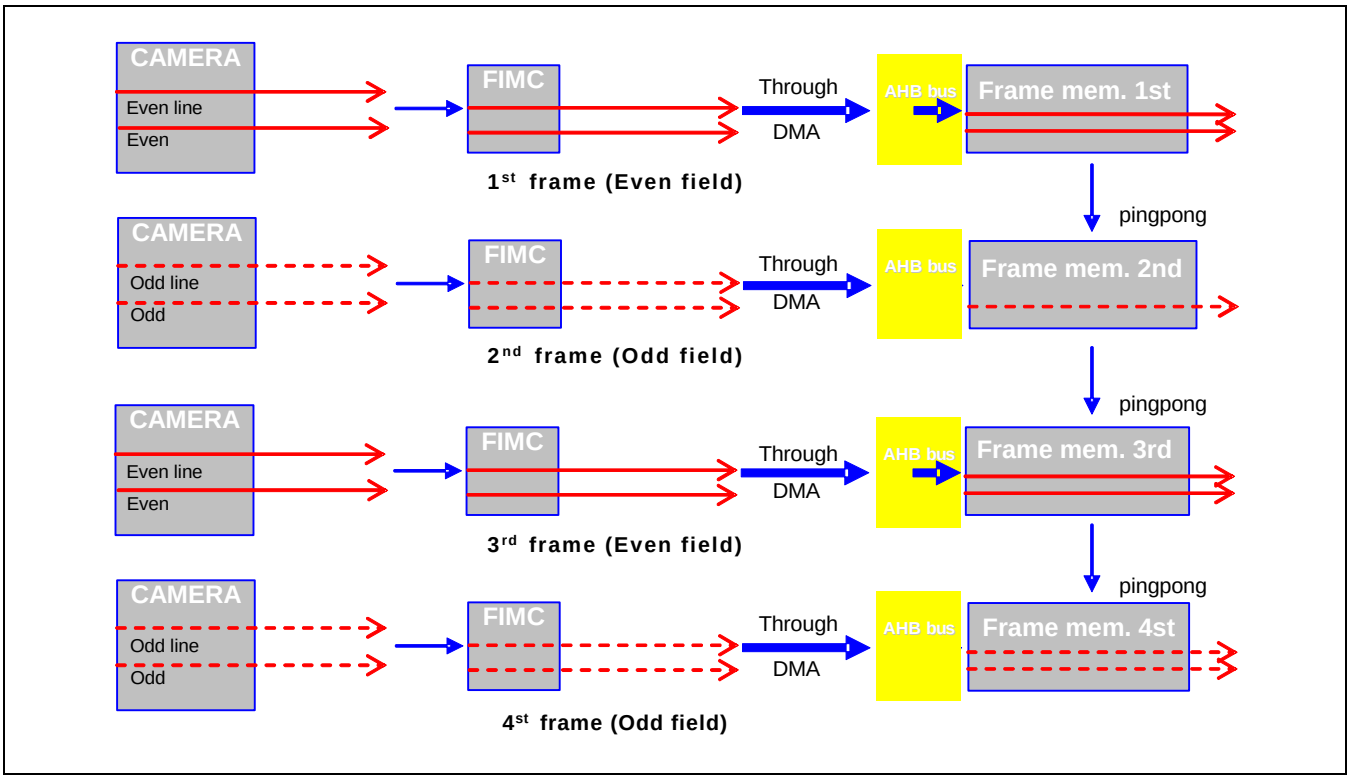


Figure 9.3-17 Frame Buffer Control

8 REGISTER DESCRIPTION

CAMIF0 Base address : 0xEE20_0000				
CAMIF1 Base address : 0xEE30_0000				
CAMIF2 Base address : 0xEE40_0000				
Register	**Offset**	**R/W**	**Description**	**Reset Value**
CISRCFMTn	0x00	R/W	Input Source Format	0x0000_0000
CIWDOFSTn	0x04	R/W	Window offset register	0x0000_0000
CIGCTRLn	0x08	R/W	Global control register	0x2001_0080
CIWDOFST2n	0x14	R/W	Window offset register 2	0x0000_0000
CIOYSA1n	0x18	R/W	Y 1st frame start address for output DMA	0x0000_0000
CIOYSA2n	0x1C	R/W	Y 2nd frame start address for output DMA	0x0000_0000
CIOYSA3n	0x20	R/W	Y 3rd frame start address for output DMA	0x0000_0000
CIOYSA4n	0x24	R/W	Y 4th frame start address for output DMA	0x0000_0000
CIOCBSA1n	0x28	R/W	Cb 1st frame start address for output DMA	0x0000_0000
CIOCBSA2n	0x2c	R/W	Cb 2nd frame start address for output DMA	0x0000_0000
CIOCBSA3n	0x30	R/W	Cb 3rd frame start address for output DMA	0x0000_0000
CIOCBSA4n	0x34	R/W	Cb 4th frame start address for output DMA	0x0000_0000
CIOCRSA1n	0x38	R/W	Cr 1st frame start address for output DMA	0x0000_0000
CIOCRSA2n	0x3c	R/W	Cr 2nd frame start address for output DMA	0x0000_0000
CIOCRSA3n	0x40	R/W	Cr 3rd frame start address for output DMA	0x0000_0000
CIOCRSA4n	0x44	R/W	Cr 4th frame start address for output DMA	0x0000_0000
CITRGFMTn	0x48	R/W	Target image format	0x0000_0000
CIOCTRLn	0x4c	R/W	Output DMA control related	0x0000_0000
CISCPRERATIOn	0x50	R/W	Pre-scaler control 1	0x0000_0000
CISCPREDSTn	0x54	R/W	Pre-scaler control 2	0x0000_0000
CISCCTRLn	0x58	R/W	Main-scaler control	0x1800_0000
CITAREAn	0x5c	R/W	Target area	0x0000_0000
CISTATUSn	0x64	R/W	Status register	0x0000_0000
CIIMGCPTn	0xc0	R/W	Image capture enable command	0x0000_0000
CICPTSEQn	0xc4	R/W	Capture sequence related	0xFFFF_FFFF
CIIMGEFFn	0xd0	R/W	Image Effects related	0x0010_0080
CIIYSA0n	0xd4	R/W	Y frame start address for Input DMA	0x0000_0000
CIICBSA0n	0xd8	R/W	Cb frame start address for Input DMA	0x0000_0000
CIICRSA0n	0xdc	R/W	Cr frame start address for Input DMA	0x0000_0000
CIREAL_ISIZEn	0xf8	R/W	Real Input DMA image size	0x0000_0000
MSCTRLn	0xfc	R/W	Input DMA control register	0x0400_0000

CIOYOFFn	0x168	R/W	Output DMA Y offset	0x0000_0000
CIOCBOFFn	0x16c	R/W	Output DMA Cb offset	0x0000_0000
CIOCROFFn	0x170	R/W	Output DMA Cr offset	0x0000_0000
CIIYOFFn	0x174	R/W	Input DMA Y offset	0x0000_0000
CIICBOFFn	0x178	R/W	Input DMA Cb offset	0x0000_0000
CIICROFFn	0x17c	R/W	Input DMA Cr offset	0x0000_0000
ORGISIZEn	0x180	R/W	Input DMA original image size	0x0000_0000
ORGOSIZEn	0x184	R/W	Output DMA original image size	0x0000_0000
CIEXTENn	0x188	R/W	Real Output DMA image size	0x0000_0000
CIDMAPARAMn	0x18c	R/W	DMA Parameter register	0x0000_0000
CSIIMGFMTn	0x194	R/W	MIPI CSI image format register	0x0000_001E

NOTE : The last 'L' column means that SFR can change at vsync edge during camera capturing. (O : possible change, X : impossible change). Also, 'M' column means that SFRs have relationship capturing result during using input DMA path. (O : relationship, X : no relationship)

8.1 Camera Source Format Register (CISRCFMTn)

- CISRCFMT0, R/W, Address = 0xEE20_0000
- CISRCFMT1, R/W, Address = 0xEE30_0000
- CISRCFMT2, R/W, Address = 0xEE40_0000

CISRCFMTn	Bit	Description	Reset Value	M	L
ITU601_656n	[31]	1 = ITU-R BT.601 YCbCr 8-bit mode enable 0 = ITU-R BT.656 YCbCr 8-bit mode enable	0	X	X
UVOffset	[30]	Cb,Cr value offset control. 1 = Cb=Cb+128 , Cr=Cr+128 0 = +0 (normally used)	0	X	X
Reserved	[29]	Should be '0'	0	X	X
SrcHsize_CAM	[28:16]	Source horizontal pixel number. (16's multiple. Must be 4's multiple of PreHorRatio if WinOfsEn is 0). Refer to the gathering extension register. SrcHsize_CAM_ext	0	X	O
Order422_CAM	[15:14]	Camera Input YCbCr order inform for 8-bit mode 8-bit mode Data Flow → 00 = $Y_0Cb_0Y_1Cr_{0...}$ 01 = $Y_0Cr_0Y_1Cb_{0...}$ 10 = $Cb_0Y_0Cr_0Y_{1...}$ 11 = $Cr_0Y_0Cb_0Y_{1...}$	0	X	X
SrcVsize_CAM	[13:0]	Source vertical pixel number. (Must be multiple of PreVerRatio if V scale down if WinOfsEn is 0)	0	X	O

8.2 Camera Window Offset Register (CIWDOFSTn)

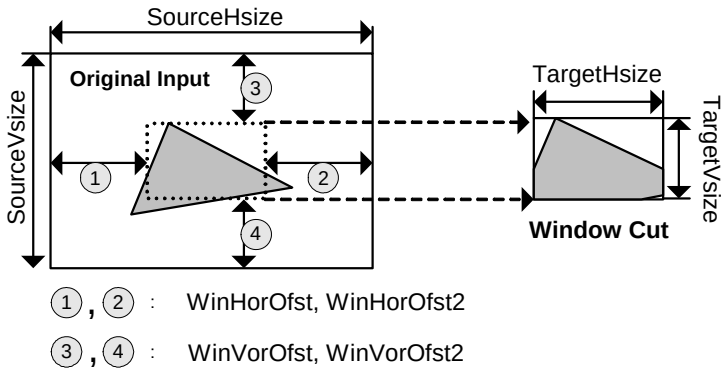


Figure 9.3-18 Camera Window Offset Sheme

(WinHorOfst2 & WinVerOfst2 are assigned in the CIWDOFST2 register)

- CIWDOFST0, R/W, Address = 0xEE20_0004
- CIWDOFST1, R/W, Address = 0xEE30_0004
- CIWDOFST2, R/W, Address = 0xEE40_0004

CIWDOFSTn	Bit	Description	Reset Value	M	L
WinOfsEn	[31]	1 = Enables window offset enable 0 = No offset	0	X	O
ClrOvFiY	[30]	1 = Clears the overflow indication flag of input FIFO Y 0 = Normal	0	X	X
ClrOvRLB	[29]	Clears the overflow indication flag of Line Buffer for Rotation	0	X	X
Reserved	[28:27]	Reserved	0	X	X
WinHorOfst	[26:16]	Window horizontal offset by pixel unit. (It should be 2's multiple) **Caution:** SourceHsize-WinHorOfst- WinHorOfst2 should be 16's multiple. Refer to the gathering extension register. WinHorOfst_ext	0	X	O
ClrOvFiCb	[15]	1 = Clears the overflow indication flag of input FIFO Cb 0 = Normal	0	X	X
ClrOvFiCr	[14]	1 = Clears the overflow indication flag of input FIFO Cr 0 = Normal	0	X	X
Reserved	[13:12]	Reserved	0	X	X
WinVerOfst	[11:0]	Window vertical offset by pixel unit.	0	X	O

NOTE: Clear bits should be set by zero after clearing the flags.

Crop Hsize (= SourceHsize - WinHorOfst - WinHorOfst2) must be 16's multiple and 4's multiple of PreHorRatio.

Crop Vsize (= SourceVsize - WinVerOfst - WinVerOfst2) must be multiple of PreVerRatio when V scale down. Must be an even number and minimum 8 if output image format YCbCr 4:2:0

< Example >

Crop Hsize	Permitted Prescale_ratio	PreDstWidth_xx
8n	2	4n
16n	2 or 4	4n
32n	2, 4 or 8	4n

8.3 Global Control Register (CIGCTRLn)

- CIGCTRL0, R/W, Address = 0xEE20_0008
- CIGCTRL1, R/W, Address = 0xEE30_0008
- CIGCTRL2, R/W, Address = 0xEE40_0008

CIGCTRLn	Bit	Description	Reset Value	M	L
SwRst	[31]	Camera interface software reset. Before you set this bit, set the ITU601_656n bit of CISRCFMT as "1" temporarily at first SFR setting. Next sequence is recommended. (ITU601 case : ITU601_656n "1" → SwRst "1" → SwRst "0" for first SFR setting , ITU656 case : ITU601_656n "1" → SwRst "1" → SwRst "0" → ITU601_656n "0" for first SFR setting)	0	X	X
CamRst_A	[30]	External camera processor A Reset or Power Down control	0	X	X
SelCam_ITU	[29]	External multiple ITU camera select 1 = ITU Camera A select 0 = ITU Camera B select	1	X	X
TestPattern	[28:27]	This register should be set at only ITU-T 601 8-bit mode. Not allowed with ITU-T 656 mode. Source CAM size should be 640 x 480 size and Order422_CAM register should be '00'. 00 = External camera processor input (normal) 01 = Color bar test pattern 10 = Horizontal increment test pattern 11 = Vertical increment test pattern	0	X	X
InvPolPCLK	[26]	1 = Inverse the polarity of PCLK 0 = Normal	0	X	X
InvPolVSYNC	[25]	1 = Inverse the polarity of VSYNC 0 = Normal	0	X	X
InvPolHREF	[24]	1 = Inverse the polarity of HREF 0 = Normal	0	X	X
Reserved	[23]	Should be '0'	0	X	X

CIGCTRLn	Bit	Description	Reset Value	M	L
IRQ_Ovfen	[22]	1 = Enables Overflow interrupt (Interrupt is generated during overflow occurrence) 0 = Disables Overflow interrupt (normal)	0	X	X
Href_mask	[21]	1 = Mask out Href during Vsync blank 0 = No mask	0	X	X
IRQ_LEVEL	[20]	1 = Level interrupt 0 = Edge trigger interrupt (default) This bit should be set as '1' since S5PC100 uses only level interrupt.	0	X	X
IRQ_CLR	[19]	This bit is related only Level interrupt. Write IRQ_CLR to '1' to clear Interrupt. This bit Auto-clear.	0	X	X
Reserved	[18:17]	Reserved	0	X	X
IRQ_Enable	[16]	1 = Enables Interrupt (default) 0 = Disables Interrupt	1	X	X
Reserved	[15:8]	Reserved	0	X	X
Reserved	[7]	Should be '1'	1		
Reserved	[6:5]	Reserved	0	X	X
InvPolHSYNC	[4]	1 = Inverse the polarity of HSYNC (this bit is useful only delay count interlace mode and FIELD port is connected HSYNC) 0 = Normal	0	X	X
SelCam_CAMIF	[3]	External camera select 1 = Selects MIPI Camera 0 = Selects ITU Camera	0	X	X
FIELDMODE	[2]	ITU601 Interlace field mode (Do not care this bit in ITU656 mode) 1 = Using the FIELD port mode (FIELD port = FIELD signal) 0 = Edge delay count mode (FIELD port = HSYNC signal) **Note:** Check!! FIELD port connection	0	X	X
InvPolFIELD	[1]	1 = Inverse the polarity of FIELD 0 = Normal	0	X	X
Cam_Interlace	[0]	External Cameraj scan method 1 = Interlace 0 = Progressive	0	X	X

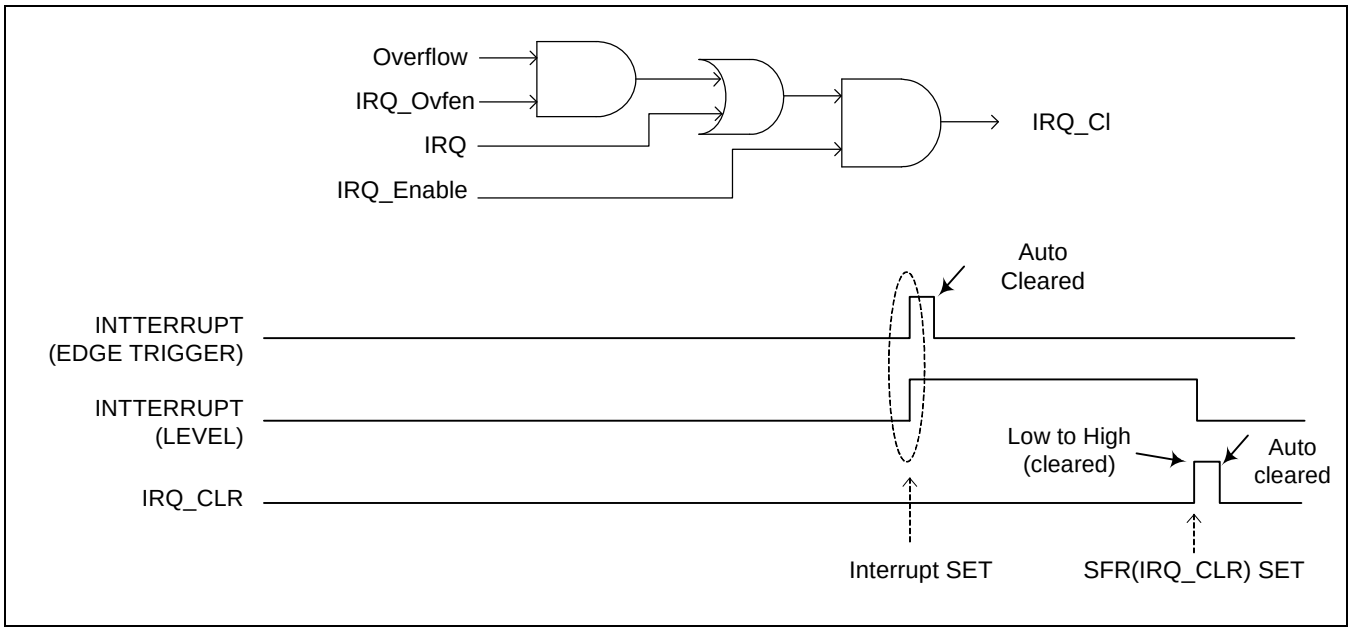


Figure 9.3-19 Interrupt Generation Scheme

8.4 Window Offset Register 2 (CIWDOFST2n)

- CIWDOFST20, R/W, Address = 0xEE20_0014
- CIWDOFST21, R/W, Address = 0xEE30_0014
- CIWDOFST22, R/W, Address = 0xEE40_0014

CIWDOFST2n	Bit	Description	Reset Value	M	L
Reserved	[31:28]	Reserved	0	X	X
WinHorOfst2	[27:16]	Window horizontal offset2 by pixel unit. (It should be 2's multiple) Caution: Camera SourceHsize-WinHorOfst- WinHorOfst2 should be 16's multiple and minimum 16.	0	X	O
Reserved	[15:12]	Reserved	0	X	X
WinVerOfst2	[11:0]	Window vertical offset2 by pixel unit	0	X	O

8.5 Output DMA Y1 Start Address Register (CIOYSA1n)

- CIOYSA10, R/W, Address = 0xEE20_0018
- CIOYSA11, R/W, Address = 0xEE30_0018
- CIOYSA12, R/W, Address = 0xEE40_0018

CIOYSA1n	Bit	Description	Reset Value	M	L
CIOYSA1	[31:0]	Output format : YCbCr 2/3 plane → Y 1st frame start address Output format : YCbCr 1 plane → YCbCr 1st frame start address Output format : RGB → RGB 1st frame start address	0	O	X

8.6 Output DMA Y2 Start Address Register (CIOYSA2n)

- CIOYSA20, R/W, Address = 0xEE20_001C
- CIOYSA21, R/W, Address = 0xEE30_001C
- CIOYSA22, R/W, Address = 0xEE40_001C

CIOYSA2n	Bit	Description	Reset Value	M	L
CIOYSA2	[31:0]	Output format : YCbCr 2/3 plane → Y 2nd frame start address Output format : YCbCr 1 plane → YCbCr 2nd frame start address Output format : RGB → RGB 2nd frame start address	0	O	X

8.7 Output DMA Y3 Start Address Register (CIOYSA3n)

- CIOYSA30, R/W, Address = 0xEE20_0020
- CIOYSA31, R/W, Address = 0xEE30_0020
- CIOYSA32, R/W, Address = 0xEE40_0020

CIOYSA3n	Bit	Description	Reset Value	M	L
CIOYSA3	[31:0]	Output format : YCbCr 2/3 plane → Y 3rd frame start address Output format : YCbCr 1 plane → YCbCr 3rd frame start address Output format : RGB → RGB 3rd frame start address	0	O	X

8.8 Output DMA Y4 Start Address Register (CIOYSA4n)

- CIOYSA40, R/W, Address = 0xEE20_0024
- CIOYSA41, R/W, Address = 0xEE30_0024
- CIOYSA42, R/W, Address = 0xEE40_0024

CIOYSA4n	Bit	Description	Reset Value	M	L
CIOYSA4	[31:0]	Output format : YCbCr 2/3 plane → Y 4th frame start address Output format : YCbCr 1 plane → YCbCr 4th frame start address Output format : RGB → RGB 4th frame start address	0	O	X

8.9 Output DMA Cb1 Start Address Register (CIOCBSA1n)

- CIOCBSA10, R/W, Address = 0xEE20_0028
- CIOCBSA11, R/W, Address = 0xEE30_0028
- CIOCBSA12, R/W, Address = 0xEE40_0028

CIOCBSA1n	Bit	Description	Reset Value	M	L
CIOCBSA1	[31:0]	Output format : YCbCr 3 plane → Cb 1st frame start address Output format : YCbCr 2 plane → CbCr 1st frame start address	0	O	X

8.10 Output DMA Cb2 Start Address Register (CIOCBSA2n)

- CIOCBSA20, R/W, Address = 0xEE20_002C
- CIOCBSA21, R/W, Address = 0xEE30_002C
- CIOCBSA22, R/W, Address = 0xEE40_002C

CIOCBSA2n	Bit	Description	Reset Value	M	L
CIOCBSA2	[31:0]	Output format : YCbCr 3 plane → Cb 2nd frame start address Output format : YCbCr 2 plane → CbCr 2nd frame start address	0	O	X

8.11 Output DMA Cb3 Start Address Register (CIOCBSA3n)

- CIOCBSA30, R/W, Address = 0xEE20_0030
- CIOCBSA31, R/W, Address = 0xEE30_0030
- CIOCBSA32, R/W, Address = 0xEE40_0030

CIOCBSA3n	Bit	Description	Reset Value	M	L
CIOCBSA3	[31:0]	Output format : YCbCr 3 plane → Cb 3rd frame start address Output format : YCbCr 2 plane → CbCr 3rd frame start address	0	O	X

8.12 Output DMA Cb4 Start Address Register (CIOCBSA4n)

- CIOCBSA40, R/W, Address = 0xEE20_0034
- CIOCBSA41, R/W, Address = 0xEE30_0034
- CIOCBSA42, R/W, Address = 0xEE40_0034

CIOCBSA4n	Bit	Description	Reset Value	M	L
CIOCBSA4	[31:0]	Output format : YCbCr 3 plane → Cb 4th frame start address Output format : YCbCr 2 plane → CbCr 4th frame start address	0	O	X

8.13 Output DMA Cr1 Start Address Register (CIOCRSA1n)

- CIOCRSA10, R/W, Address = 0xEE20_0038
- CIOCRSA11, R/W, Address = 0xEE30_0038
- CIOCRSA12, R/W, Address = 0xEE40_0038

CIOCRSA1n	Bit	Description	Reset Value	M	L
CIOCRSA1	[31:0]	Output format : YCbCr 3 plane → Cr 1st frame start address	0	O	X

8.14 Output DMA Cr2 Start Address Register (CIOCRSA2n)

- CIOCRSA20, R/W, Address = 0xEE20_003C
- CIOCRSA21, R/W, Address = 0xEE30_003C
- CIOCRSA22, R/W, Address = 0xEE40_003C

CIOCRSA2n	Bit	Description	Reset Value	M	L
CIOCRSA2	[31:0]	Output format : YCbCr 3 plane → Cr 2nd frame start address	0	O	X

8.15 Output DMA Cr3 Start Address Register (CIOCRSA3n)

- CIOCRSA30, R/W, Address = 0xEE20_0040
- CIOCRSA31, R/W, Address = 0xEE30_0040
- CIOCRSA32, R/W, Address = 0xEE40_0040

CIOCRSA3n	Bit	Description	Reset Value	M	L
CIOCRSA3	[31:0]	Output format : YCbCr 3 plane → Cr 3rd frame start address	0	O	X

8.16 Output DMA Cr4 Start Address Register (CIOCRSA4n)

- CIOCRSA40, R/W, Address = 0xEE20_0044
- CIOCRSA41, R/W, Address = 0xEE30_0044
- CIOCRSA42, R/W, Address = 0xEE40_0044

CIOCRSA4n	Bit	Description	Reset Value	M	L
CIOCRSA4	[31:0]	Output format : YCbCr 3 plane → Cr 4th frame start address	0	O	X

8.17 Target Format Register (CITRGFMTn)

- CITRGFMT0, R/W, Address = 0xEE20_0048
- CITRGFMT1, R/W, Address = 0xEE30_0048
- CITRGFMT2, R/W, Address = 0xEE40_0048

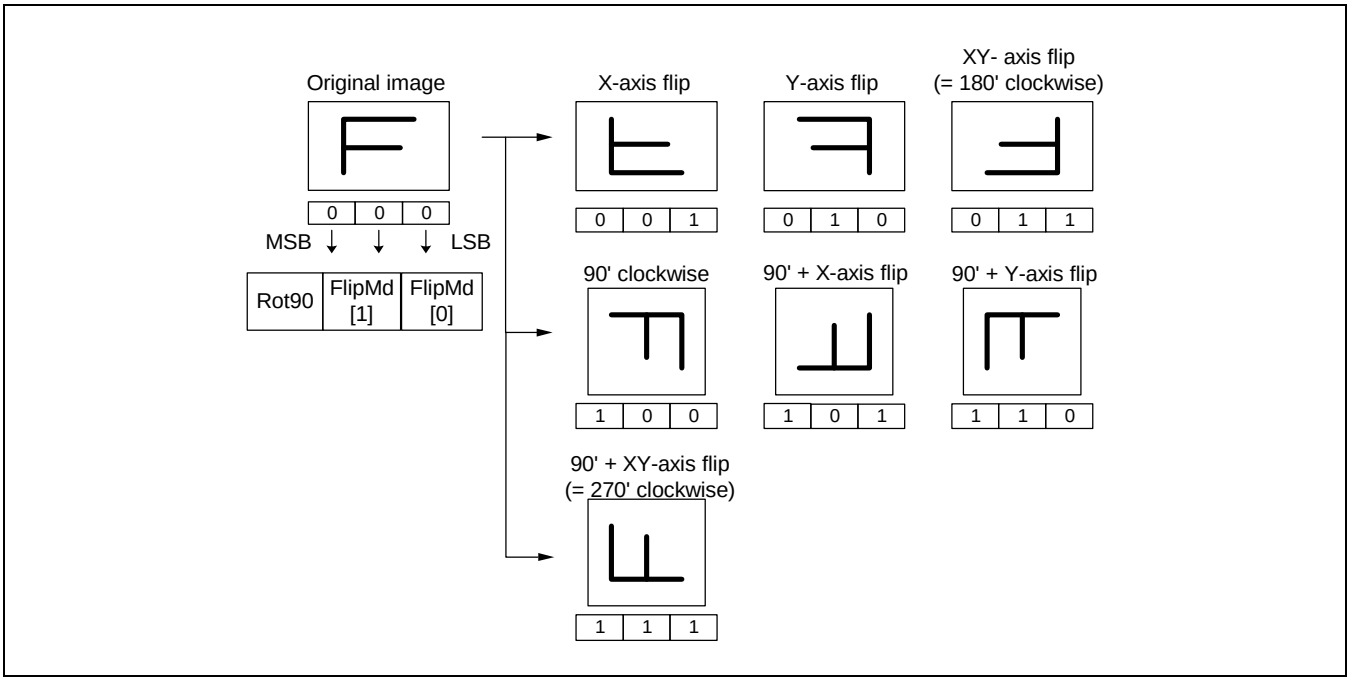


Figure 9.3-20 Image Mirror and Rotation

CITRGFMTn	Bit	Description	Reset Value	M	L
InRot90	[31]	1 = Rotate clockwise 90° (Using the Input Rotator only). Input Rotator & Output Rotator does not use at the same time. If InRot90 enable mode, output data path should be LCD FIFO path) 0 = Input Rotator bypass	0	O	O
OutFormat	[30:29]	00 = YCbCr 4:2:0 output image format. (2 or 3 plane) 01 = YCbCr 4:2:2 output image format. (2 or 3 plane) (ref. 2 or 3 plane format register → C_INT_OUT) 10 = YCbCr 4:2:2 output image format. (1 plane) 11 = RGB output image format. (Ref. RGB format register → OutRGB_FMT).	0	O	O
TargetHsize	[28:16]	Horizontal pixel number of target image. Refer to gathering extension register. TargetHsize_ext	0	O	O
OutFlipMd	[15:14]	Image mirror and rotation for output DMA 00 = Normal 01 = X-axis mirror 10 = Y-axis mirror 11 = 180° rotation	0	O	O
OutRot90	[13]	1 = Rotate clockwise 90° (Using the Output Rotator) 0 = Output Rotator bypass	0	O	O
TargetVsize	[12:0]	Vertical pixel number of target image. Minimum number is 4.. Refer to gathering extension register. TargetVsize_ext	0	O	O

TargetHsize and TargetVsize should not be larger than Camera SourceHsize and Camera SourceVsize. InputDMA source size don't care.

Caution! *Only input rotator supports InputDMA image data. Output rotator supports Camera or InputDMA image data. But, input and output rotator should not use at the same time. Because, Input and output rotator memory are shared for saving the memory size.*

Note! *If TargetVsize value is set to an odd number (N) when output format is YCbCr 4:2:0. The odd numbers (N) of Y lines and the (N-1)/2 of Cb, Cr lines are generated. Also, X-flip or XY-flip is not allowed. Thus YCbCr 4:2:0 ouput format should use an even TargetVsize number.*

8.18 Output DMA Control Register (CIOCTRLn)

- CIOCTRL0, R/W, Address = 0xEE20_004C
- CIOCTRL1, R/W, Address = 0xEE30_004C
- CIOCTRL2, R/W, Address = 0xEE40_004C

<table>
<tr><th>CIOCTRLn</th><th>Bit</th><th>Description</th><th>Reset Value</th><th>M</th><th>L</th></tr>
<tr><td>Reserved</td><td>[31:26]</td><td>Reserved</td><td>0</td><td>X</td><td>X</td></tr>
<tr><td>Order2p_out</td><td>[25:24]</td><td>YCbCr 4:2:0 or 4:2:2 2plane output Chroma memory storing style order (should be C_INT_OUT = 1)
<table>
<tr><td>bit</td><td>MSB LSB</td></tr>
<tr><td>00</td><td>$Cr_3Cb_3Cr_2Cb_2Cr_1Cb_1Cr_0Cb_0$</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table></td><td>0</td><td>O</td><td>O</td></tr>
<tr><td>Reserved</td><td>[23:4]</td><td>Reserved</td><td>0</td><td>X</td><td>X</td></tr>
<tr><td>C_INT_OUT</td><td>[3]</td><td>1 = YCbCr 4:2:0 or 4:2:2 2plane output format
0 = YCbCr 4:2:0 or 4:2:2 3plane output format</td><td>0</td><td>O</td><td>O</td></tr>
<tr><td>LastIRQEn</td><td>[2]</td><td>1 = enable last IRQ at the end of frame capture (It is recommended to check the done signal of capturing image for JPEG)
0 = normal</td><td>0</td><td>X</td><td>X</td></tr>
<tr><td>Order422_out</td><td>[1:0]</td><td>YCbCr 4:2:2 1plane output memory storing style order.
<table>
<tr><td>bit</td><td>MSB LSB</td></tr>
<tr><td>00</td><td>$Cr_1Y_3Cb_1Y_2Cr_0Y_1Cb_0Y_0$</td></tr>
<tr><td>01</td><td>$Cb_1Y_3Cr_1Y_2Cb_0Y_1Cr_0Y_0$</td></tr>
<tr><td>10</td><td>$Y_3Cr_1Y_2Cb_1Y_1Cr_0Y_0Cb_0$</td></tr>
<tr><td>11</td><td>$Y_3Cb_1Y_2Cr_1Y_1Cb_0Y_0Cr_0$</td></tr>
</table></td><td>0</td><td>O</td><td>O</td></tr>
</table>

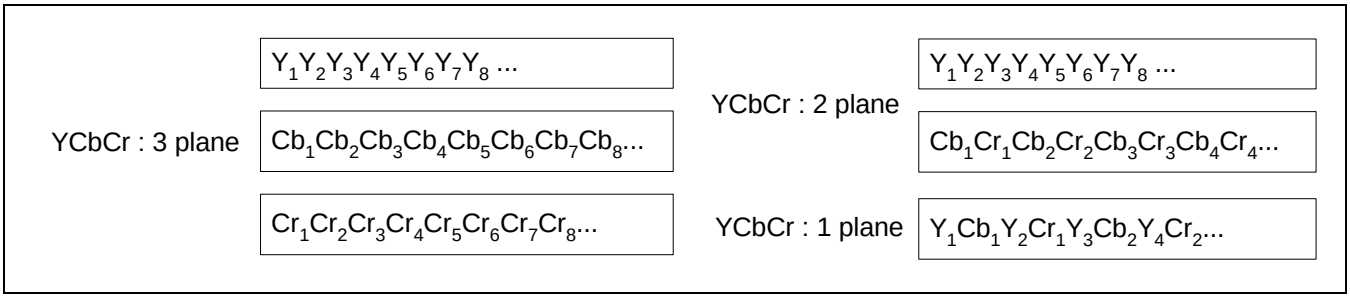


Figure 9.3-21 YCbCr Plane Memory Storing Style.

- **REGISTER SETTING GUIDE FOR SCALER**

SRC_Width and DST_Width satisfy the double word (8bytes) boundary constraints such that the number of horizontal pixel represents kn where n = 1, 2, 3 … and k = 1 / 2 / 8 for 24bppRGB / 16bppRGB / YCbCr420 image, respectively.

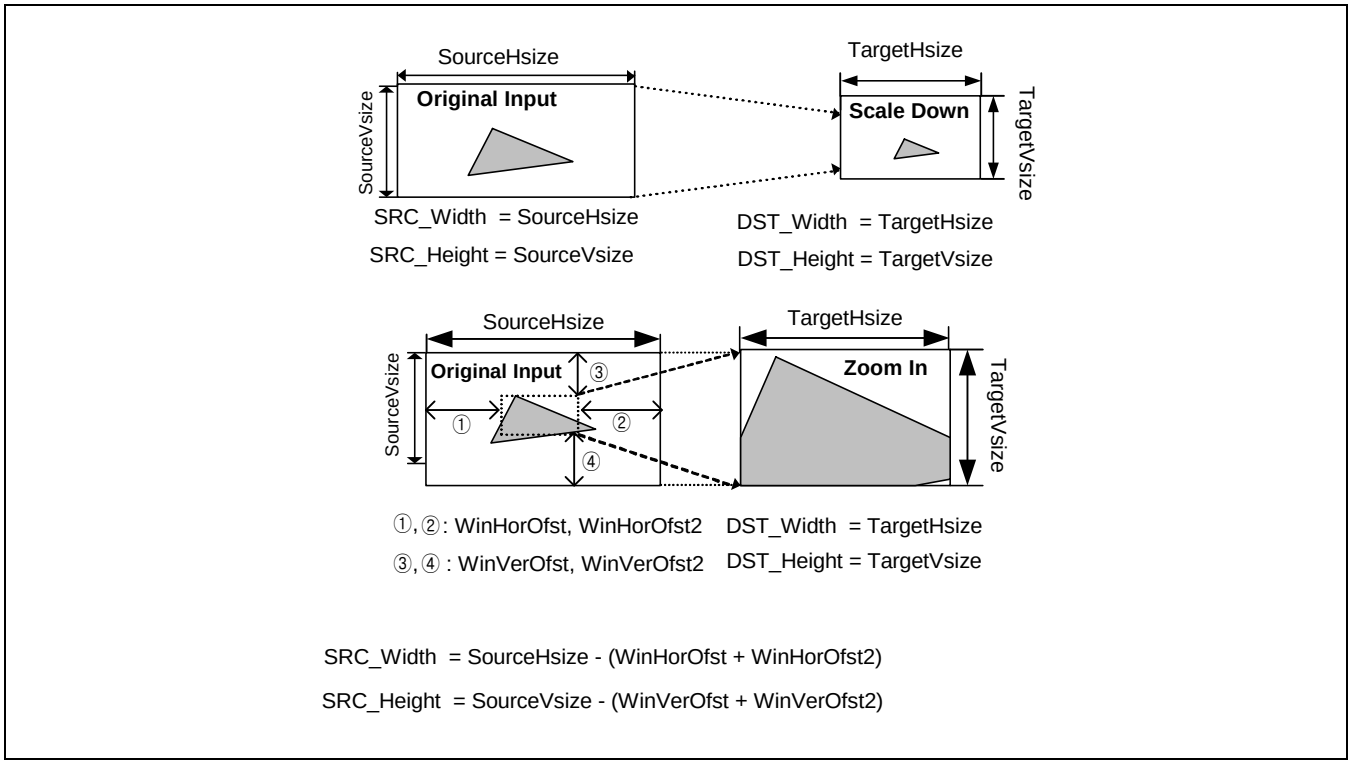


Figure 9.3-22 Scaling Scheme

The other control registers of pre-scaled image size, pre-scale ratio, pre-scale shift ratio and main scale ratio are defined according to the following equations.

If (SRC_Width >= 64 × DST_Width) { Exit(-1); /* Out Of Horizontal Scale Range */ }

else if (SRC_Width >= 32 × DST_Width) { PreHorRatio = 32; H_Shift = 5 }

else if (SRC_Width >= 16 × DST_Width) { PreHorRatio = 16; H_Shift = 4 }

else if (SRC_Width >= 8 × DST_Width) { PreHorRatio = 8; H_Shift = 3 }

else if (SRC_Width >= 4 × DST_Width) { PreHorRatio = 4; H_Shift = 2 }

else if (SRC_Width >= 2 × DST_Width) { PreHorRatio = 2; H_Shift = 1 }

else { PreHorRatio = 1; H_Shift = 0 }

PreDstWidth = SRC_Width / PreHorRatio

MainHorRatio = (SRC_Width << 8) / (DST_Width << H_Shift)

If (SRC_Height >= 64 × DST_Height) { Exit(-1); /* Out Of Vertical Scale Range */ }

else if (SRC_Height >= 32 × DST_Height) { PreVerRatio = 32; V_Shift = 5 }

else if (SRC_Height >= 16 × DST_Height) { PreVerRatio = 16; V_Shift = 4 }

else if (SRC_Height >= 8 × DST_Height) { PreVerRatio = 8; V_Shift = 3 }

else if (SRC_Height >= 4 × DST_Height) { PreVerRatio = 4; V_Shift = 2 }

else if (SRC_Height >= 2 × DST_Height) { PreVerRatio = 2; V_Shift = 1 }

else { PreVerRatio = 1; V_Shift = 0 }

PreDstHeight = SRC_Height / PreVerRatio

MainVerRatio = (SRC_Height << 8) / (DST_Height << V_Shift)

SHfactor = 10 – (H_Shift + V_Shift)

Caution! In Zoom-In case, you should check the next equation (CAM-In case).

((SourceHsize - (WinHorOfst + WinHorOfst2)) / PreHorRatio) ≤ Max. scaler line buffer size width

8.19 Pre-Scaler Control Register 1 (CISCPRERATIOn)

- CISCPRERATIO0, R/W, Address = 0xEE20_0050
- CISCPRERATIO1, R/W, Address = 0xEE30_0050
- CISCPRERATIO2, R/W, Address = 0xEE40_0050

CISCPRERATIOn	Bit	Description	Reset Value	M	L
SHfactor	[31:28]	Shift factor for pre-scaler	0	O	O
Reserved	[27:23]	Reserved	0	X	X
PreHorRatio	[22:16]	Horizontal ratio of pre-scaler	0	O	O
Reserved	[15:7]	Reserved	0	X	X
PreVerRatio	[6:0]	Vertical ratio of pre-scaler	0	O	O

8.20 Pre-Scaler Control Register 2 (CISCPREDSTn)

- CISCPREDST0, R/W, Address = 0xEE20_0054
- CISCPREDST1, R/W, Address = 0xEE30_0054
- CISCPREDST2, R/W, Address = 0xEE40_0054

CISCPREDSTn	Bit	Description	Reset Value	M	L
Reserved	[31:30]	Reserved	0	X	X
PreDstWidth	[29:16]	Destination width for pre-scaler	0	O	O
Reserved	[15:14]	Reserved	0	X	X
PreDstHeight	[13:0]	Destination height for pre-scaler	0	O	O

8.21 Main-Scaler Control Register (CISCCTRLn)

- CISCCTRL0, R/W, Address = 0xEE20_0058
- CISCCTRL1, R/W, Address = 0xEE30_0058
- CISCCTRL2, R/W, Address = 0xEE40_0058

CISCCTRLn	Bit	Description	Reset Value	M	L
ScalerBypass	[31]	Scaler bypass. In this case, ImgCptEn_SC should be 0, but ImgCptEn should be 1. Generally this mode uses large image size upper scaler maximum size. (This mode is intended to capture JPEG input image for DSC application). In this case, input pixel buffering depends on only input FIFOs, therefore system bus should not be busy in this mode. ScalerBypass has some restriction. Size scaling, color space conversion, Input DMA mode and any RGB format are not allowed. If input format is YCbCr4:2:2, output format should be YCbCr4:2:2 or YCbCr4:2:0	0	X	X
ScaleUp_H	[30]	Horizontal scale up/down flag for scaler (In 1:1 scale ratio, this bit should be "1") 1: up, 0:down	0	O	O
ScaleUp_V	[29]	Vertical scale up/down flag for scaler (In 1:1 scale ratio, this bit should be "1") 1: up, 0:down	0	O	O
CSCR2Y	[28]	YCbCr Data Dynamic Range Selection for the Color Space Conversion RGB to YCbCr 1 = Wide => Y/Cb/Cr (0 ~ 255) : Wide default 0 = Narrow => Y (16 ~ 235), Cb/Cr (16 ~ 240) ※ Recommend CSC range setting CSCR2Y= CSCY2R (Wide=Wide or Narrow=Narrow)	1	O	O

CISCCTRLn	Bit	Description	Reset Value	M	L
CSCY2R	[27]	YCbCr Data Dynamic Range Selection for the Color Space Conversion YCbCr to RGB 1 = Wide => Y/Cb/Cr (0 ~ 255) : Wide default 0 = Narrow => Y (16 ~ 235), Cb/Cr (16 ~ 240)	1	O	O
LCDPathEn	[26]	FIFO Mode Enable. 1 for FIFO output and 0 for DMA output. If FIFO mode enable, Input mode should be DMA input and WSWP bit at WINCON0(0xEE00_0020) in LCD controller should be '0'. FIFO mode output format is RGB24bit. its selection depends on OutFormat register. OutFormat = RGB → RGB24bit.	0	O	O
Interlace	[25]	Output scan method selection register if FIFO mode (LCDPathEn =1). 1 for Interlace scan and 0 for progressive scan. In case of DMA output (LCDPathEn = 0), only progressive scan is applied whatever this value has. This mode is not allowed if Input image data is from Camera processor.	0	O	X
MainHorRatio	[24:16]	Horizontal scale ratio for main-scaler	0	O	O
ScalerStart	[15]	Scaler start 1 = Scaler start 0 = Scaler stop or scaler bypass	0	O	O
InRGB_FMT	[14:13]	Input DMA RGB format. 00 = RGB565 , 01 = RGB666 10 = RGB888 , 11 = Reserved	0	O	X
OutRGB_FMT	[12:11]	Output DMA RGB format 00 = RGB565 , 01 = RGB666 10 = RGB888 , 11 = Reserved	0	O	O
Ext_RGB	[10]	Input RGB data extension enable bit for the conversion of RGB565/666 mode into RGB888 mode. 1 = Extension , 0 = normal i) Input R = 5-bit in RGB565 mode 10100 -> 10100101 (Extension) : [7]=[2], [6]=[1], [5]=[0] 10100 -> 10100000 (normal) ii) Input R = 6-bit in RGB666 mode 101100 -> 10110010 (Extension) : [7]=[1], [6]=[0] 101100 -> 10110000 (normal)	0	O	O
One2One	[9]	Scaler does not run interpolation but run repetition for upsampling. Sometimes other IP needs this method if input format are YCbCr4:2:0 and YCbCr4:2:2. Scalerbypass should be set '0' and Don't care plane. (Caution : One2One should be used if input / output format and size are same) Ex) input YCbCr4:2:0 2plane -> output YCbCr4:2:0 3plane (O.K)	0	O	O
MainVerRatio	[8:0]	Vertical scale ratio for main-scaler	0	O	O

Table 9.3- 5 Color Space Conversion Equations

	Wide	Narrow
CSCY2R (601)	R = Y + 1.371(Cr-128) G = Y – 0.698(Cr-128) – 0.336(Cb-128) B = Y + 1.732(Cb-128)	R = 1.164(Y-16) + 1.596(Cr-128) G = 1.164(Y-16) – 0.813(Cr-128) – 0.391(Cb-128) B = 1.164(Y-16) + 2.018(Cb-128)
CSCY2R (709)	R = Y + 1.540(Cr-128) G = Y – 0.459(Cr-128) – 0.183(Cb-128) B = Y + 1.816(Cb-128)	R = 1.164(Y-16) + 1.793(Cr-128) G = 1.164(Y-16) – 0.534(Cr-128) – 0.213(Cb-128) B = 1.164(Y-16) + 2.115(Cb-128)
CSCR2Y (601)	Y = 0.299R + 0.587G + 0.114B Cb = – 0.172R – 0.339G + 0.511B +128 Cr = 0.511R – 0.428G – 0.083B + 128	Y = 0.257R + 0.504G + 0.098B + 16 Cb = – 0.148R – 0.291G + 0.439B + 128 Cr = 0.439R – 0.368G – 0.071B +128
CSCR2Y (709)	Y = 0.213R + 0.715G + 0.072B Cb = – 0.117R – 0.394G + 0.511B +128 Cr = 0.511R – 0.464G – 0.047B + 128	Y = 0.183R + 0.614G + 0.062B + 16 Cb = – 0.101R – 0.338G + 0.439B + 128 Cr = 0.439R – 0.399G – 0.040B +128

DMA Mode Operation (Normal mode): DMA Input → DMA Output

Source image format is one of YCbCr420, YCbCr422, and RGB16/18/24-bit format. Destination image format is one of YCbCr420, YCbCr422, and RGB 16/18/24 bit format. All source and destination image data need to be stored in memory system aligned with double word boundary. Supports DMA operation. Therefore, the width of source and destination image should be selected to satisfy the double word boundary condition.

FIFO Mode Operation: DMA Input → FIFO Output

In FIFO Mode, (LCDPathEn =1), two types of color space conversion such as RGB2YCbCr and YCbCr2RGB are available like a DMA mode operation. Destination image is transferred to the FIFO in display controller (or some other IP with FIFO interface) without additional memory bandwidth such as CAMIF-to-Memory and Memory-to-Display Controller. Output data format is determined by only Output format register: OutFormat = RGB format (24-bit RGB). The source image format and the destination image format restriction are described below table.

Table 9.3-6 FIFO Mode Image Format

Input Image Format (Progressive)		Output image Format (Progressive / Interlace)
YCbCr	YCbCr420 : 3/2 plane	RGB 24-bit
	YCbCr422 : 3/2/1 plane	
RGB	RGB 16/18/24-bit	

In FIFO mode (LCDPathEnable =1), either progressive or interlace scan mode are selectable according to "interlace" control register. Register Files Lists. The "interlace" control bit is available if LCDPathEn=1, otherwise its value is unaffected to DMA mode operation which support only progressive.

Even if an interlaced scan mode is enable (LCDPathEn = 1 and Interlace = 1), per frame management, which consists of even field and odd filed, is automic. This means that user interruption is unnecessary to inter field switching in the same frame. Therefore, the frame management scheme is identical for both progressive and interlace scan mode. Interlace is not supported if camera processor is selected input data

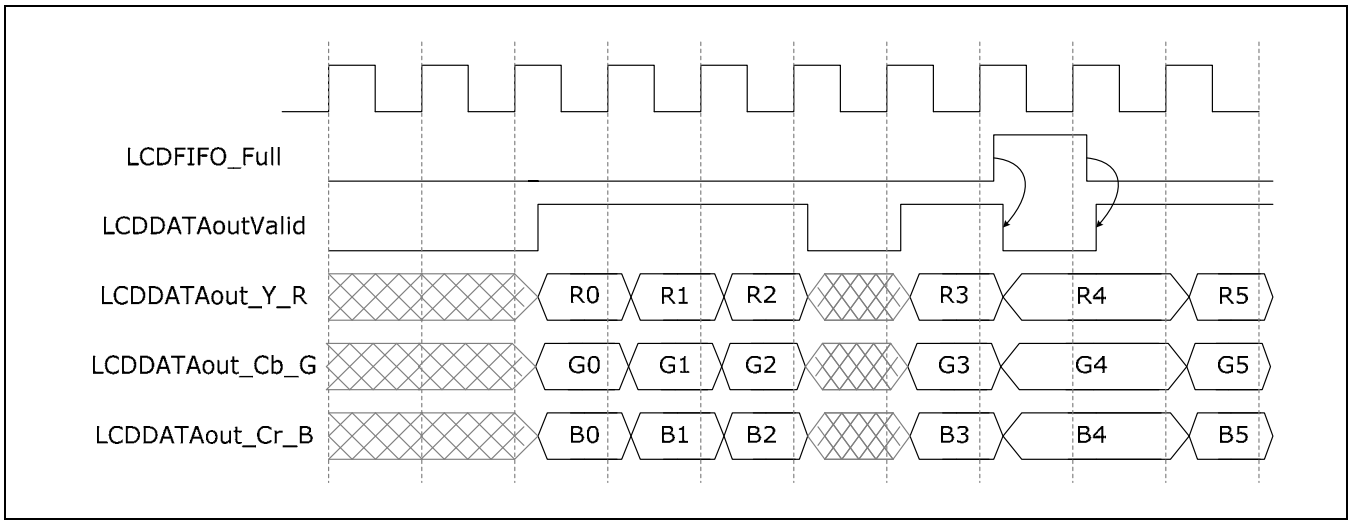


Figure 9.3-23 I/O Timing Diagram for Direct Path

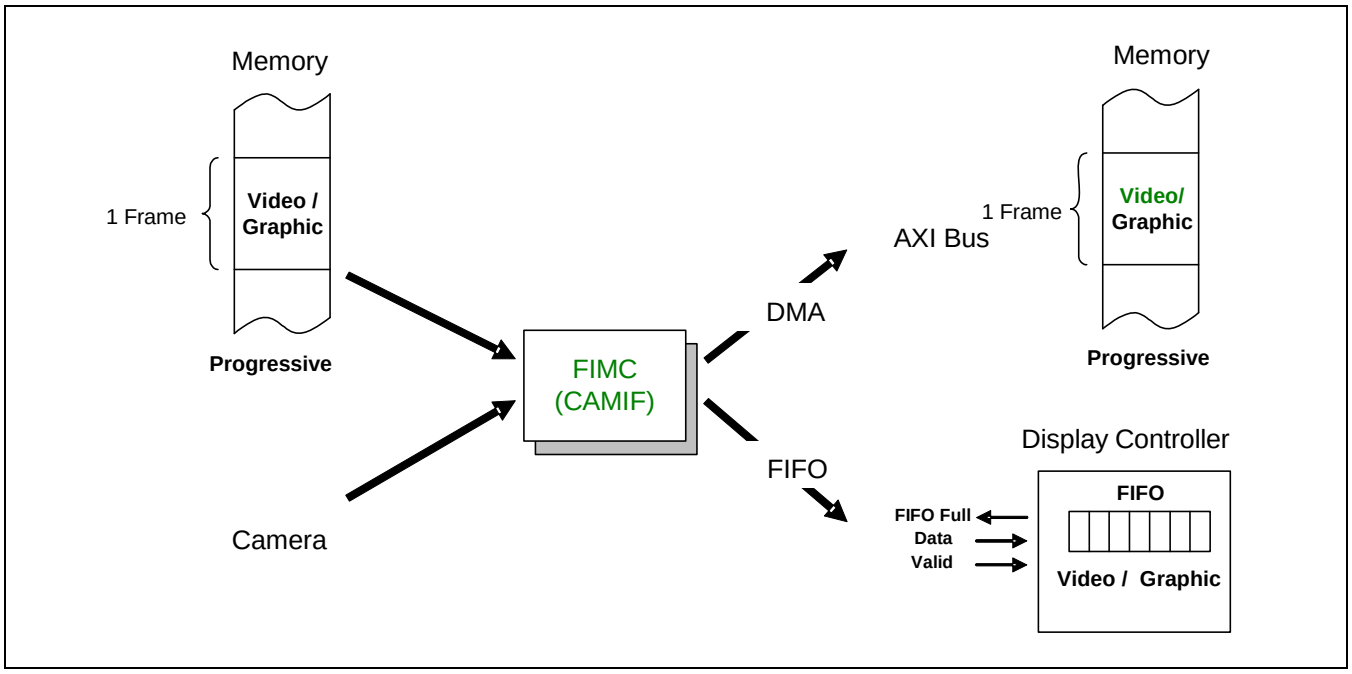


Figure 9.3-24 Input & Output Modes in CAMIF

8.22 Output DMA Target Area Register (CITAREAn)

- CITAREA0, R/W, Address = 0xEE20_005C
- CITAREA1, R/W, Address = 0xEE30_005C
- CITAREA2, R/W, Address = 0xEE40_005C

CITAREAn	Bit	Description	Reset Value	M	L
Reserved	[31:28]	Reserved	0	X	X
CITAREA	[27:0]	Target area for output DMA = Target H size x Target V size	0	O	O

8.23 Status Register (CISTATUSn)

- CISTATUS0, Address = 0xEE20_0064
- CISTATUS1, Address = 0xEE30_0064
- CISTATUS2, Address = 0xEE40_0064

CISTATUSn	Bit	Description	RW	Reset Value	M	L
OvFiY	[31]	Overflow state of FIFO Y	R	0	X	X
OvFiCb	[30]	Overflow state of FIFO Cb	R	0	X	X
OvFiCr	[29]	Overflow state of FIFO Cr	R	0	X	X
VSYNC	[28]	Camera VSYNC (CPU refers this bit for first SFR setting after external camera muxing. It is seen in the ITU-R BT 656 mode)	R	0	X	X
FrameCnt	[27:26]	Frame count of output DMA. This counter value means the next frame number	R	0	X	X
WinOfstEn	[25]	Window offset enable status.	R	0	X	X
FlipMd	[24:23]	Flip mode of output DMA	R	0	X	X
ImgCptEn	[22]	Image capture enable of global camera interface	R	0	X	X
ImgCptEn_SC	[21]	Image capture enable of scaler path	R	0	X	X
VSYNC_A	[20]	External camera A VSYNC. Polarity inversion was not adopted.	R	X	X	X
VSYNC_B	[19]	External camera B VSYNC. Polarity inversion was not adopted.	R	X	X	X
OvRLB	[18]	Overflow status of Line Buffer for Rotation.	R	0	X	X
FrameEnd	[17]	When frame operation finish, FrameEnd is generated. and FrameEnd is clear by user setting ' 0'	R/W	0	X	X
LastCaptureEnd	[16]	Last frame capture status. LastCaptureEnd is clear by user setting ' 0' . This singnal is applied only Camera input mode.	R/W	0	X	X
VVALID_A	[15]	External camera A VVALID	R	X	X	X
VVALID_B	[14]	External camera B VVALID	R	X	X	X
IRQ_CAM	[13]	Interrupt Status for Camera Input Mode	R	0	X	X
IRQ_DMAend	[12]	Interrupt Status for DMA Frame End for DMA Input Mode	R	0	X	X
FrameCptStatus	[11]	Capture frame control status. 1 = Present capture enable , 0 = Present capture disable	R	0	X	X
FrameFieldStatus	[10]	ITU CAMERA field status. Internal value after inverse polarity. 1 = Present frame Field1 , 0 = Present frame Field0	R	0	X	X
Reserved	[9:0]	Reserved	-	0	X	X

8.24 Image Capture Enable Register (CIIMGCPTn)

- CIIMGCPT0, R/W, Address = 0xEE20_00C0
- CIIMGCPT1, R/W, Address = 0xEE30_00C0
- CIIMGCPT2, R/W, Address = 0xEE40_00C0

CIIMGCPTn	Bit	Description	Reset Value	M	L
ImgCptEn	[31]	camera interface global capture enable	0	X	O
ImgCptEn_Sc	[30]	Capture enable for scaler. This bit must be zero in scaler-bypass mode.	0	O	O
Reserved	[29:26]	Reserved	0	X	X
Cpt_FrEn	[25]	Capture frame control. (only camera progressive input is applied) 1 = Enables (Step-by-Step frame one shot mode) 0 = Disables (FreeRun mode)	0	X	O
Reserved	[24]	Reserved	0	X	X
Cpt_FrPtr	[23:19]	Capture sequence turn-around pointer	0	X	X
Cpt_ FrMod	[18]	Capture frame control mode 1 = Apply Cpt_FrCnt mode (capture Cpt_FrCnt frames along the Cpt_FrSeq after capture dma frame control becomes enable. If Cpt_FrCnt = 0, then no more capture) 0 = Apply Cpt_FrEn mode (capture frames along the Cpt_FrSeq during Cpt_FrEn is high. This sequence repeats until capture frame control disables)	0	X	X
Cpt_ FrCnt	[17:10]	Wanted number of frames to be captured. If register read, you will see the value of a shadow register which is downcounted if a frame is captured. That is, Cpt_FrCnt has an initially loaded value still after a frame is captured.	0	X	X
Reserved	[9:0]	Reserved	0	X	X

8.25 Capture Sequence Register (CICPTSEQn)

- CICPTSEQ0, R/W, Address = 0xEE20_00C4
- CICPTSEQ1, R/W, Address = 0xEE30_00C4
- CICPTSEQ2, R/W, Address = 0xEE40_00C4

CICPTSEQn	Bit	Description	Reset Value	M	L
Cpt_FrSeq	[31:0]	Capture sequence pattern. This register is valid if Cpt_FrEn has high value.	FFFF_FFFF	X	X

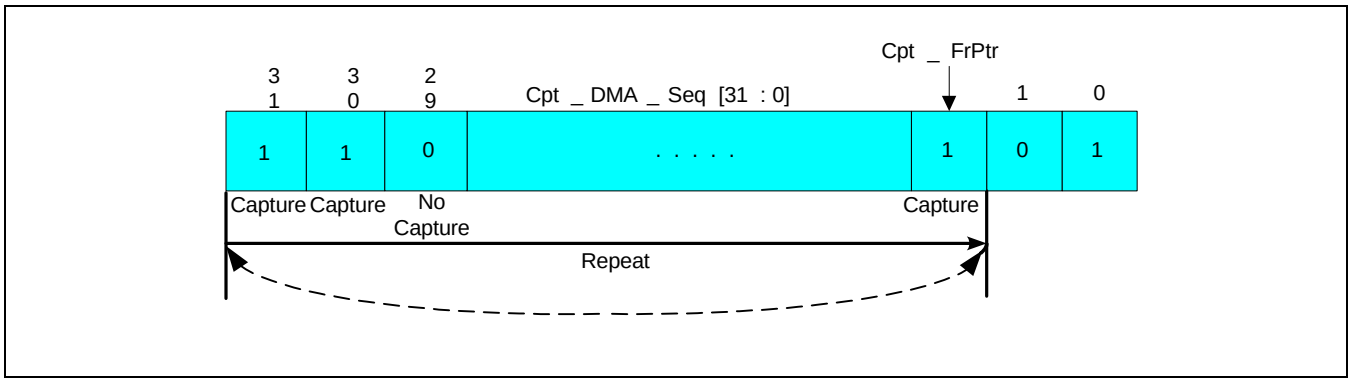


Figure 9.3-25 Capture Frame Control

※ For skipped frames, IRQ is not generated and FrameCnt is not increased.

8.26 Image Effects Register (CIIMGEFFn)

- CIIMGEFF0, R/W, Address = 0xEE20_00D0
- CIIMGEFF1, R/W, Address = 0xEE30_00D0
- CIIMGEFF2, R/W, Address = 0xEE40_00D0

CIIMGEFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
IE_ON	[30]	0 = Disables image effect function , 1 = Enables	0	O	O
IE_AFTER_SC	[29]	Image Effect location 1 = After scaling (camera and input DMA image are applied except scaler bypass mode) 0 = Before scaling (only itu camera image should be applied) In case of *Before scaling*, it applies image effect even though it is in scalar bypass mode.	0	O	O
FIN	[28:26]	Image Effect selection. 3'd0 = Bypass 3'd1 = Arbitrary Cb/Cr 3'd2 = Negative 3'd3 = Art Freeze 3'd4 = Embossing 3'd5 = Silhouette	0	O	O
Reserved	[25:21]	Reserved	0	X	X
PAT_Cb	[20:13]	It is used only for FIN is Arbitrary Cb/Cr (PAT_Cb/Cr == 8'd128 for GRAYSCALE) Wide CSC Range : 0 < PAT_Cb < 255 Narrow CSC Range : 16 ≤ PAT_Cb ≤ 240	8'd128	O	O
Reserved	[12:8]	Reserved	0	X	X
PAT_Cr	[7:0]	It is used only for FIN is Arbitrary Cb/Cr (PAT_Cb/Cr == 8'd128 for GRAYSCALE) Wide CSC Range : 0 < PAT_Cr < 255 Narrow CSC Range : 16 ≤ PAT_Cr ≤ 240	8'd128	O	O

Cf) sepia: PAT_Cb = 8'd115, PAT_Cr = 8'd145

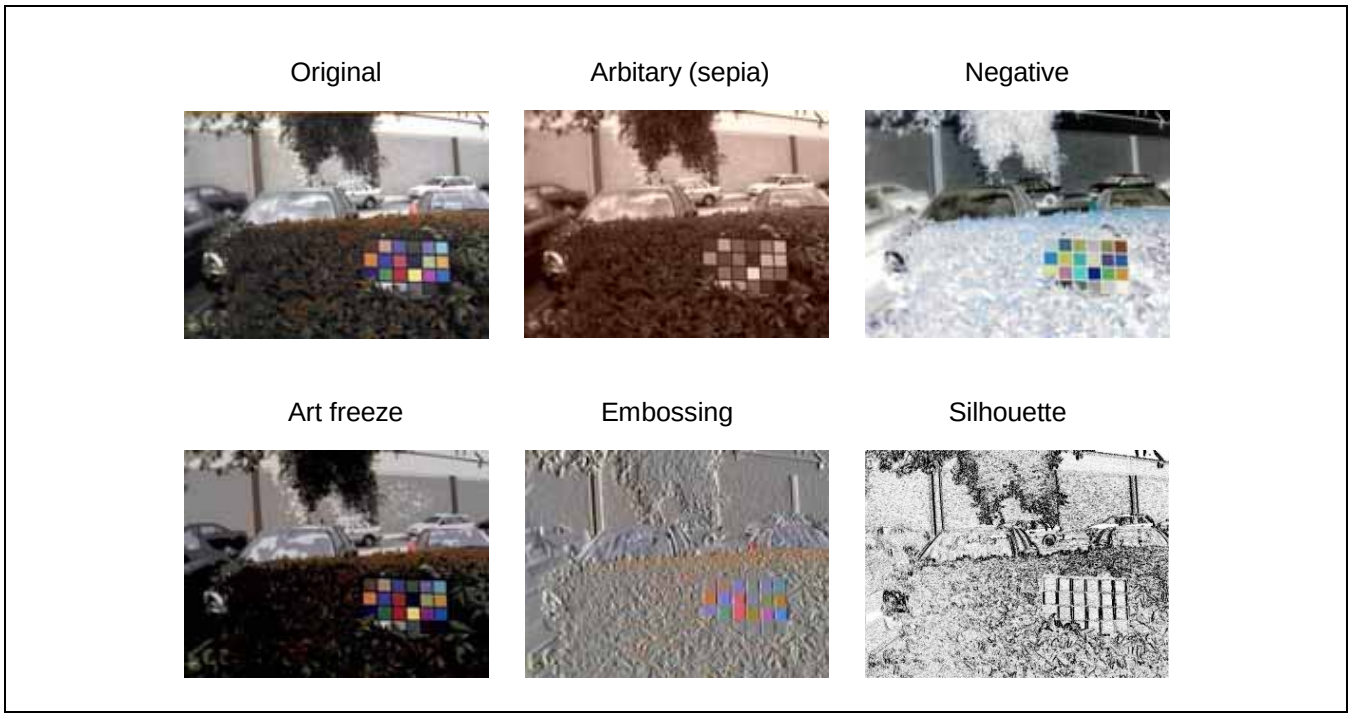


Figure 9.3-26 Image Effect

8.27 Input DMA Y0 Start Register (CIIYSA0n)

- CIIYSA0, R/W, Address = 0xEE20_00D4
- CIIYSA1, R/W, Address = 0xEE30_00D4
- CIIYSA2, R/W, Address = 0xEE40_00D4

CIIYSA0n	Bit	Description	Reset Value	M	L
CIIYSA0	[31:0]	Input format : YCbCr 2/3 plane → Y frame start address Input format : YCbCr 1 plane → YCbCr frame start address Input format : RGB → RGB frame start address	0	O	X

8.28 Input DMA Cb0 Start Register

- CIICBSA00, R/W, Address = 0xEE20_00D8
- CIICBSA01, R/W, Address = 0xEE30_00D8
- CIICBSA02, R/W, Address = 0xEE40_00D8

CIICBSA0n	Bit	Description	Reset Value	M	L
CIICBSA0	[31:0]	Input format : YCbCr 3 plane → Cb frame start address Input format : YCbCr 2 plane → CbCr frame start address	0	O	X

8.29 Input DMA Cr0 Start Register (CIICRSA0n)

- CIICRSA00, R/W, Address = 0xEE20_00DC
- CIICRSA01, R/W, Address = 0xEE30_00DC
- CIICRSA02, R/W, Address = 0xEE40_00DC

CIICRSA0n	Bit	Description	Reset Value	M	L
CIICRSA0	[31:0]	Input format : YCbCr 3 plane → Cr frame start address	0	O	X

8.30 Real Input DMA Size Register (CIREAL_ISIZEn)

- CIREAL_ISIZE0, R/W, Address = 0xEE20_00F8
- CIREAL_ISIZE1, R/W, Address = 0xEE30_00F8
- CIREAL_ISIZE2, R/W, Address = 0xEE40_00F8

CIREAL_ISIZEn	Bit	Description	Reset Value	M	L
AutoLoadEnable	[31]	Input DMA Automatically restart (Only Software trigger mode) At the first frame start, it requires ENVID_M start setting. After first frame, next frame does not need ENVID_M setting. If autoload function is running, size & format value should be fixed. 0 = Disables AutoLoad , 1 = Enables AutoLoad	0	O	X
ADDR_CH_DIS	[30]	Input DMA Address Change Disable (Only Software trigger mode) At the first frame start needs ADDR_CH_DIS = '0' 0 : Address change enable , 1 : Address change disable	0	O	X
REAL_HEIGHT	[29:16]	Input DMA real image vertical pixel size. Minimum 8. Must be multiple of PreVerRatio. If InRot90 = 1, must be 16's multiple. Must be 4's multiple of PreHorRatio. Minimum 16.	0	O	X
Reserved	[15:14]	Reserved	0	X	X
REAL_WIDTH	[13:0]	Input DMA real image horizontal pixel size. Must be 16's multiple. Must be 4's multiple of PreHorRatio. Minimum 16. If InRot90 = 1, minimum 8. Must be multiple of PreVerRatio.	0	O	X

8.31 Input DMA Control Register (MSCTRLn)

- MSCTRL0, Address = 0xEE20_00FC
- MSCTRL1, Address = 0xEE30_00FC
- MSCTRL2, Address = 0xEE40_00FC

<table>
<tr><th>MSCTRLn</th><th>Bit</th><th>Description</th><th>RW</th><th>Reset Value</th><th>M</th><th>L</th></tr>
<tr><td>Reserved</td><td>[31:28]</td><td>Reserved</td><td>R/W</td><td>0</td><td>X</td><td>X</td></tr>
<tr><td>Successive_cnt</td><td>[27:24]</td><td>Input DMA burst successive count. 4 is default. (4, 3, 2, 1 is possible). This value should not be '0'.</td><td>R/W</td><td>4'd4</td><td>O</td><td>X</td></tr>
<tr><td>Reserved</td><td>[23:18]</td><td>Reserved</td><td>R/W</td><td>0</td><td>X</td><td>X</td></tr>
<tr><td>Order2p_in</td><td>[17:16]</td><td>Source Input DMA image is YCbCr 4:2:0 or 4:2:2 2plane memory reading style order.
<table><tr><td>bit</td><td>MSB LSB</td></tr><tr><td>00</td><td>$Cr_3Cb_3Cr_2Cb_2Cr_1Cb_1Cr_0Cb_0$</td></tr><tr><td>01</td><td>Reserved</td></tr><tr><td>10</td><td>Reserved</td></tr><tr><td>11</td><td>Reserved</td></tr></table></td><td>R/W</td><td>0</td><td>O</td><td>X</td></tr>
<tr><td>C_INT_IN</td><td>[15]</td><td>1 = YCbCr 4:2:0 or 4:2:2 2plane input format
0 = YCbCr 4:2:0 or 4:2:2 3plane input format</td><td>R/W</td><td>0</td><td>O</td><td>X</td></tr>
<tr><td>InFlipMd</td><td>[14:13]</td><td>Image mirror and rotation for Input DMA
00 = Normal
01 = X-axis mirror
10 = Y-axis mirror
11 = 180° rotation (XY-axis mirror)</td><td>R/W</td><td>0</td><td>O</td><td>X</td></tr>
<tr><td>Reserved</td><td>[12:7]</td><td>Reserved</td><td></td><td>0</td><td>X</td><td>X</td></tr>
<tr><td>EOF_M</td><td>[6]</td><td>If Input DMA operation is complete, it generates End Of Frame (read only this signal).</td><td>R</td><td>0</td><td>O</td><td>X</td></tr>
<tr><td>Order422_M</td><td>[5:4]</td><td>If source Input DMA image is 1plane YCbCr 4:2:2,
1plane YCbCr 4:2:2 input memory reading order style.
<table><tr><td>bit</td><td>MSB LSB</td></tr><tr><td>00</td><td>$Y_3Cb_1Y_2Cr_1Y_1Cb_0Y_0Cr_0$</td></tr><tr><td>01</td><td>$Cb_1Y_3Cr_1Y_2Cb_0Y_1Cr_0Y_0$</td></tr><tr><td>10</td><td>$Y_3Cr_1Y_2Cb_1Y_1Cr_0Y_0Cb_0$</td></tr><tr><td>11</td><td>$Cr_1Y_3Cb_1Y_2Cr_0Y_1Cb_0Y_0$</td></tr></table></td><td>R/W</td><td>0</td><td>O</td><td>X</td></tr>
<tr><td>SEL_DMA_CAM</td><td>[3]</td><td>Input data selection.
0 = External camera input path
1 = Memory data input path (Input DMA)</td><td>R/W</td><td>0</td><td>O</td><td>X</td></tr>
</table>

MSCTRLn	Bit	Description	RW	Reset Value	M	L
InFormat_M	[2:1]	Source image format for Input DMA 00 = YCbCr 4:2:0 input image format. (2 or 3 plane) 01 = YCbCr 4:2:2 input image format. (2 or 3 plane) (ref. 2 or 3 plane format register → C_INT_IN) 10 = YCbCr 4:2:2 input image format. (1 plane) 11 = RGB input image format. (ref. RGB format register → InRGB_FMT)	R/W	0	O	X
ENVID_M	[0]	Input DMA operation start. (Software setting triggers Low to High) Hardware clears automatically. 1) SEL_DMA_CAM = '0', ENVID_M don't care (using external camera signal) 2) SEL_DMA_CAM = '1', ENVID_M is set (0→1) then Input DMA operation start	R/W	0	O	X

NOTE: ENVID_M SFR must be set at last. Starting order for using DMA input path.
SEL_DMA_CAM (others SFR setting) → Image Capture Enable & Scaler start SFR setting → ENVID_M SFR setting.

Cf.) Image Capture Enable SFR must be set at last. Starting order for using Camera input path.

SEL_DMA_CAM →SEL_CAMIF (others SFR setting) →Image Capture Enable & Scaler start SFR setting

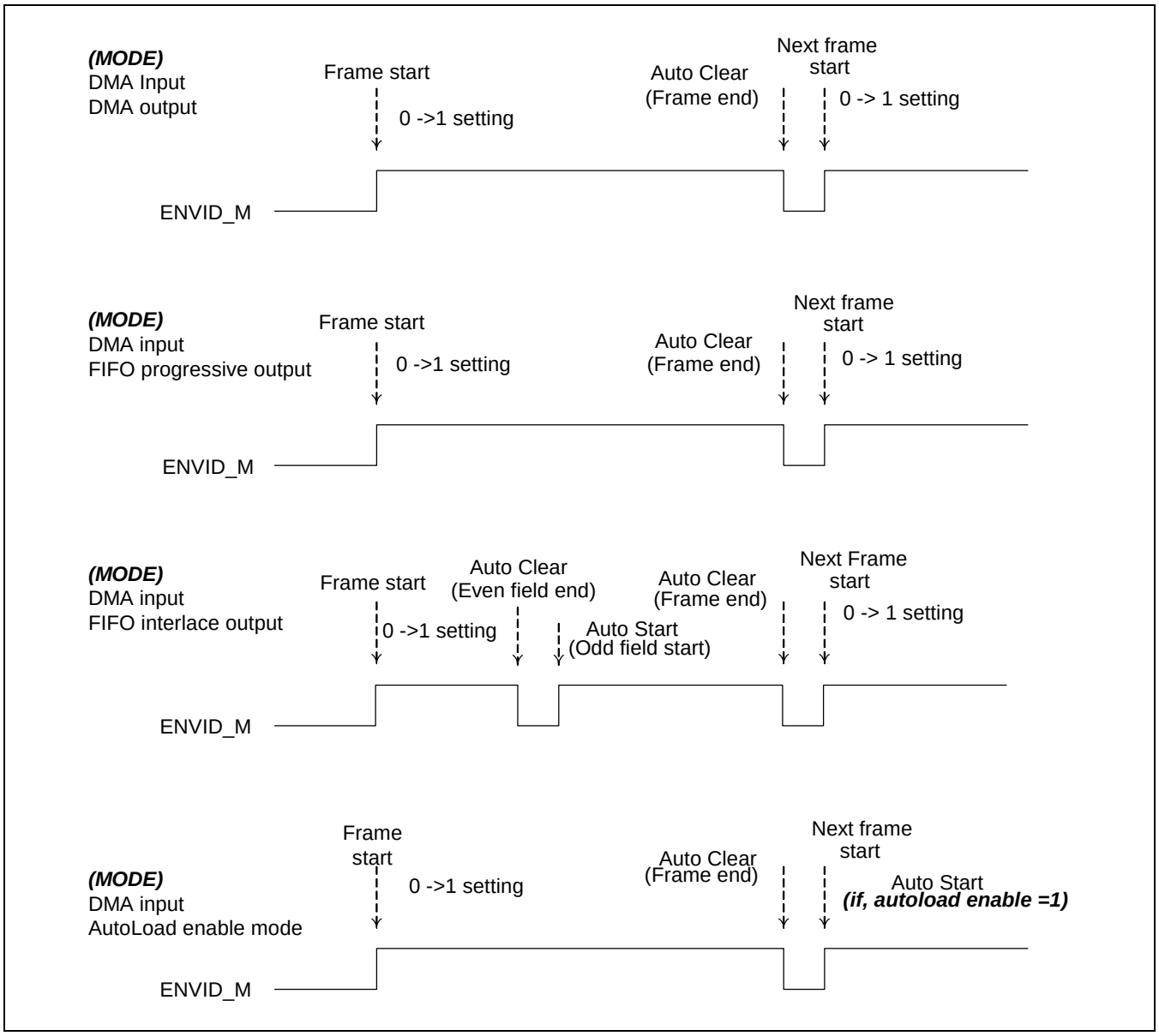


Figure 9.3-27 ENVID_M SFR Setting When Input DMA Start to Read Memory Data

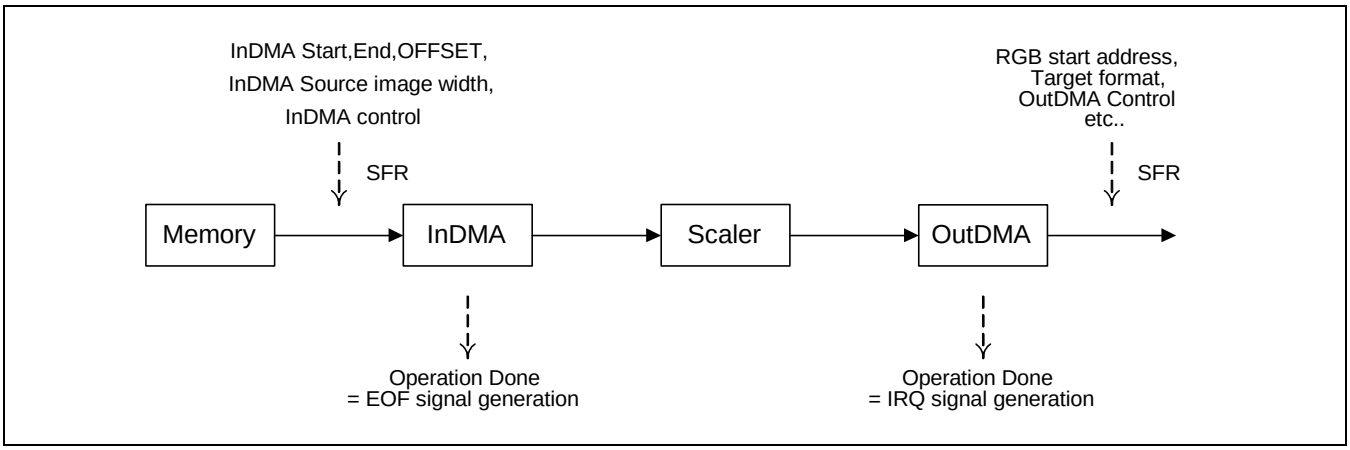


Figure 9.3-28 SFR & Operation (Related Each DMA When Selected Input DMA Path)

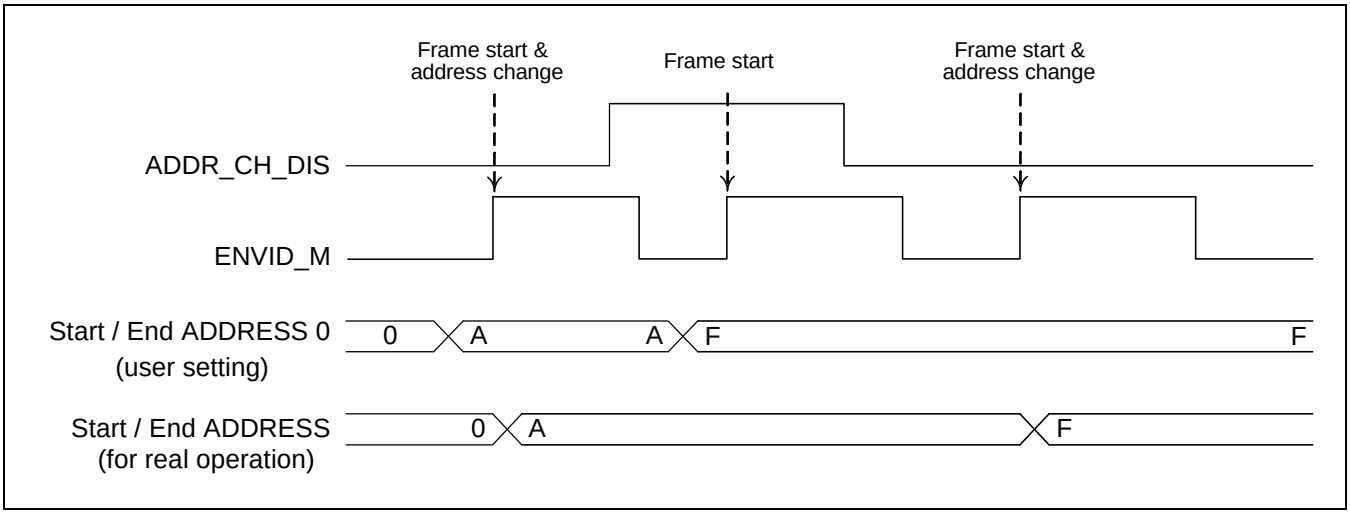


Figure 9.3-29 Address Change Timing (Related Input DMA)

8.32 Output DMA Y Offset Register (CIOYOFFn)

- CIOYOFF0, R/W, Address = 0xEE20_0168
- CIOYOFF1, R/W, Address = 0xEE30_0168
- CIOYOFF2, R/W, Address = 0xEE40_0168

CIOYOFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
OYOFF_V	[30:16]	Output DMA vertical offset for Y component Output format : YCbCr 2/3 plane → Y height offset Output format : YCbCr 1 plane → YCbCr height offset Output format : RGB → RGB height offset	0	O	O
Reserved	[15]	Reserved	0	X	X
OYOFF_H	[14:0]	Output DMA horizontal offset for Y component Output format : YCbCr 2/3 plane → Y width offset Output format : YCbCr 1 plane → YCbCr width offset Output format : RGB → RGB width offset	0	O	O

8.33 Output DMA Cb Offset Register (CIOCBOFFn)

- CIOCBOFF0, R/W, Address = 0xEE20_016C
- CIOCBOFF1, R/W, Address = 0xEE30_016C
- CIOCBOFF2, R/W, Address = 0xEE40_016C

CIOCBOFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
OCBOFF_V	[30:16]	Output DMA vertical offset for Cb component Output format : YCbCr 3 plane → Cb height offset Output format : YCbCr 2 plane → CbCr height offset	0	O	O
Reserved	[15]	Reserved	0	X	X
OCBOFF_H	[14:0]	Output DMA horizontal offset for Cb component Output format : YCbCr 3 plane → Cb width offset Output format : YCbCr 2 plane → CbCr width offset	0	O	O

8.34 Output DMA Cr Offset Register

- CIOCROFF0, R/W, Address = 0xEE20_0170
- CIOCROFF1, R/W, Address = 0xEE30_0170
- CIOCROFF2, R/W, Address = 0xEE40_0170

CIOCROFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
OCROFF_V	[30:16]	Output DMA vertical offset for Cr component Output format : YCbCr 3 plane → Cr height offset	0	O	O
Reserved	[15]	Reserved	0	X	X
OCROFF_H	[14:0]	Output DMA horizontal offset for Cr component Output format : YCbCr 3 plane → Cr width offset	0	O	O

8.35 Input DMA Y Offset Register (CIIYOFFn)

- CIIYOFF0, R/W, Address = 0xEE20_0174
- CIIYOFF1, R/W, Address = 0xEE30_0174
- CIIYOFF2, R/W, Address = 0xEE40_0174

CIIYOFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
IYOFF_V	[30:16]	Input DMA vertical offset for Y component Input format : YCbCr 2/3 plane → Y height offset Input format : YCbCr 1 plane → YCbCr height offset Input format : RGB → RGB height offset	0	O	X
Reserved	[15]	Reserved	0	X	X
IYOFF_H	[14:0]	Input DMA horizontal offset for Y component Input format : YCbCr 2/3 plane → Y width offset Input format : YCbCr 1 plane → YCbCr width offset Input format : RGB → RGB width offset	0	O	X

8.36 Input DMA Cb Offset Register (CIICBOFFn)

- CIICBOFF0, R/W, Address = 0xEE20_0178
- CIICBOFF1, R/W, Address = 0xEE30_0178
- CIICBOFF2, R/W, Address = 0xEE40_0178

CIICBOFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
ICBOFF_V	[30:16]	Input DMA vertical offset for Cb component Input format : YCbCr 3 plane → Cb height offset Input format : YCbCr 2 plane → CbCr height offset	0	O	X
Reserved	[15]	Reserved	0	X	X
ICBOFF_H	[14:0]	Input DMA horizontal offset for Cb component Input format : YCbCr 3 plane → Cb width offset Input format : YCbCr 2 plane → CbCr width offset	0	O	X

8.37 Input DMA Cr Offset Register (CIICROFFn)

- CIICRFF0, R/W, Address = 0xEE20_017C
- CIICRFF1, R/W, Address = 0xEE30_017C
- CIICRFF2, R/W, Address = 0xEE40_017C

CIICROFFn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
ICROFF_V	[30:16]	Input DMA vertical offset for Cr component Input format : YCbCr 3 plane → Cr height offset	0	O	X
Reserved	[15]	Reserved	0	X	X
ICROFF_H	[14:0]	Input DMA horizontal offset for Cr component Input format : YCbCr 3 plane → Cr width offset	0	O	X

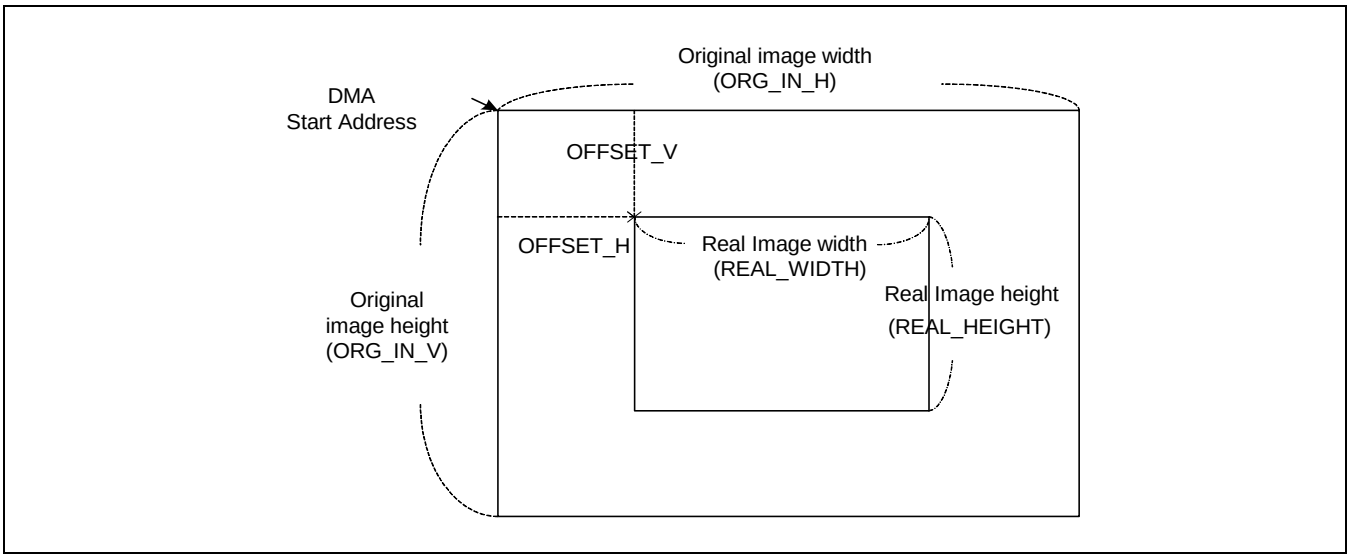


Figure 9.3-30 Input DMA Offset & Image Size

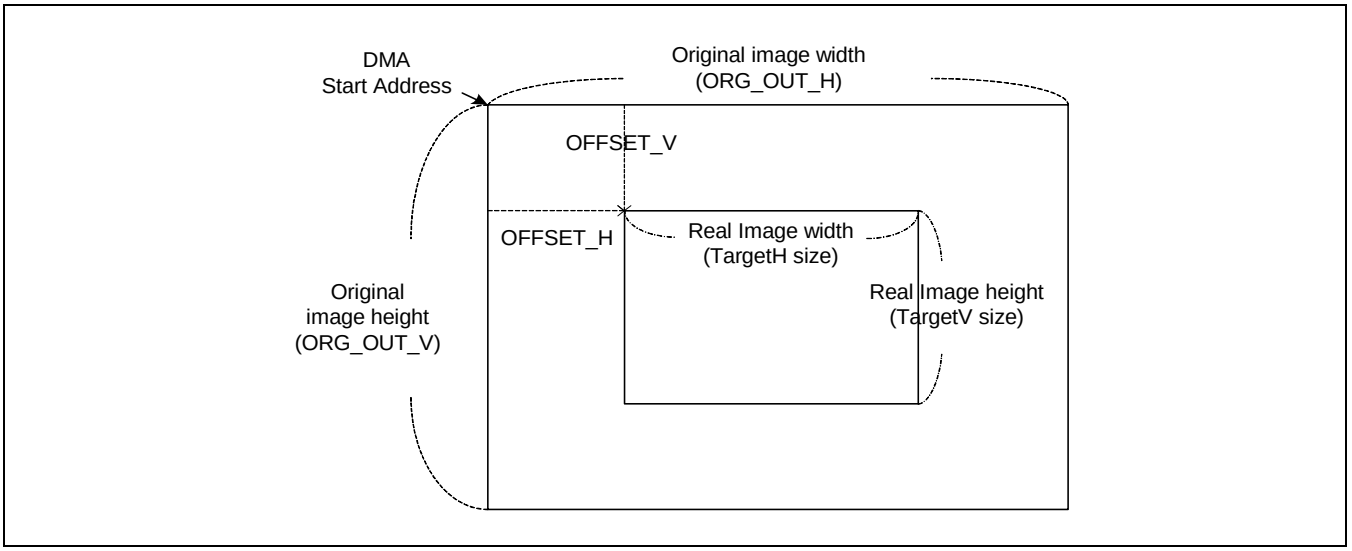


Figure 9.3-31 Output DMA Offset & Image Size

–DMA Start Address

Start address of ADDRStart_Y/Cb/Cr/RGB points the first address where the corresponding component of Y/Cb/Cr/RGB is read or written. Each one should be aligned with double word boundary (i.e. ADDRStart_X[2:0] = 3'b000). ADDRStart_Cb is valid only for the 2 or 3 planes YCbCr420, 422 source image formats. ADDRStart_Cr is valid only for the 3 plane YCbCr420, 422 source image.

–DMA OFFSET

- Offset_H_Y = Memory size for Y offset per a horizontal line (8's multiple)

= Number of pixel (or sample) in horizontal offset × *ByteSize_Per_Pixel*

Cf.) ByteSize_Per_Pixel: 1 for YCbCr420 3plane / YCbCr422 3plane

2 for YCbCr422 1plane or RGB 16bit

4 for RGB 18/24bit

- Offset_H_Cb = Memory size for Cb offset per a horizontal line (8's multiple)

= Number of pixel (or sample) in horizontal offset × *ByteSize_Per_Pixel*

Cf.) ByteSize_Per_Pixel: 1/2 for YCbCr420 3plane / YCbCr422 3plane

1 for YCbCr420 2plane / YCbCr422 2plane

- Offset_H_ Cr = Memory size for Cr offset per a horizontal line (8's multiple)

= Number of pixel (or sample) in horizontal offset × *ByteSize_Per_Pixel*

Cf.) ByteSize_Per_Pixel : 1/2 for YCbCr420 3plane / YCbCr422 3plane

- Offset_V_Y = Number of vertical Y offset

- Offset_V_Cb = Number of vertical Cb offset × *Variable Line*

Cf.) Variable Line: 1/2 for YCbCr420 – 2 or 3 plane

1 for YCbCr422 – 2 or 3 plane

- Offset_V_Cr = Number of vertical Cr offset × *Variable Line*

Cf.) Variable Line: 1/2 for YCbCr420 3 plane

1 for YCbCr422 3 plane

8.38 Original Input DMA Image Size (ORGISIZEn)

- ORGISIZE0, R/W, Address = 0xEE20_0180
- ORGISIZE1, R/W, Address = 0xEE30_0180
- ORGISIZE2, R/W, Address = 0xEE40_0180

ORGISIZEn	Bit	Description	Reset Value	M	L
Reserved	[31:30]	Reserved	0	X	X
ORG_IN_V	[29:16]	Input DMA original image vertical pixel size. Minimum 8. This size should not be less than REAL_HEIGHT register.	0	O	X
Reserved	[15:14]	Reserved	0	X	X
ORG_IN_H	[13:0]	Input DMA source image horizontal pixel size. Must be 8's multiple. This size should not be less than REAL_WIDTH register.	0	O	X

8.39 Original Output DMA Image Size (ORGOSIZEn)

- ORGOSIZE0, R/W, Address = 0xEE20_0184
- ORGOSIZE1, R/W, Address = 0xEE30_0184
- ORGOSIZE2, R/W, Address = 0xEE40_0184

ORGOSIZEn	Bit	Description	Reset Value	M	L
Reserved	[31:30]	Reserved	0	X	X
ORG_OUT_V	[29:16]	Output DMA original image vertical pixel size. Minimum 8. This size should not be less than TargetVsize register. If output rotator is running, this size should not be less than TargetHsize register	0	O	O
Reserved	[15:14]	Reserved	0	X	X
ORG_OUT_H	[13:0]	Output DMA source image horizontal pixel size. Must be 8's multiple. This size should not be less than TargetHsize register. If output rotator is running, this size should not be less than TargetVsize register.	0	O	O

8.40 Gathering Extension Register (CIEXTENn)

- CIEXTEN0, R/W, Address = 0xEE20_0188
- CIEXTEN1, R/W, Address = 0xEE30_0188
- CIEXTEN2, R/W, Address = 0xEE40_0188

CIEXTENn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
SrcHsize_CAM_ext	[30]	Bit value [13] of the camera source horizontal pixel number register. {SrcHsize_CAM_ext,SrcHsize_CAM} = {[13], [12:0]}. Thus, total camera source horizontal size = [13:0]	0	O	O
Reserved	[29]	Reserved	0	X	X
WinHorOfst_ext	[28]	Bit value [11] of the window horizontal offset register. {WinHorOfst_ext,WinHorOfst} = {[11],[10:0]}. Thus, total window horizontal offset size = [11:0]	0	O	O
Reserved	[27]	Reserved			
TargetHsize_ext	[26]	Bit value [13] of the target image horizontal pixel number register. {TargetHsize_ext,TargetHsize} = {[13],[12:0]} Thus, total target image horizontal size = [13:0]	0	O	O
Reserved	[25]	Reserved	0	X	X
TargetVsize_ext	[24]	Bit value [13] of the target image vertical number register. {TargetVsize_ext,TargetVsize} = {[13],[12:0]} Thus, total target image vertical size = [13:0]	0	O	O
Reserved	[23]	Reserved	0	X	X
Reserved	[22]	Should be '0'	0	O	O
Reserved	[21]	Reserved	0	X	X
Reserved	[20]	Should be '0'	0	O	X
Reserved	[19:0]	Reserved	0	X	X

8.41 DMA Parameter Register (CIDMAPARAMn)

- CIDMAPARAM0, R/W, Address = 0xEE20_018C
- CIDMAPARAM1, R/W, Address = 0xEE30_018C
- CIDAMPARAM2, R/W, Address = 0xEE40_018C

CIDMAPARAMn	Bit	Description	Reset Value	M	L
Reserved	[31]	Reserved	0	X	X
MODE_R	[30:29]	INPUT DMA address access style 0 = Linear 1 = Reserved 2 = Reserved 3 = 64x32 tile	0	O	X
Reserved	[28:15]	Reserved	0	X	X
MODE_W	[14:13]	OUTPUT DMA address access style 0 = Linear 1 = Reserved 2 = Reserved 3 = 64x32 tile	0	X	X
Reserved	[12:4]	Reserved	0	X	X
Reserved	[3:0]	Reserved	0	X	X

NOTE: see '4.1 Tiled memory format' in chapter 9.1 for the tile mode description.

8.42 Mipi Input Format Register (CSIIMGFMTn)

- CSIIMGFMT0, R/W, Address = 0xEE20_0194
- CSIIMGFMT1, R/W, Address = 0xEE30_0194
- CSIIMGFMT2, R/W, Address = 0xEE40_0194

CSIIMGFMTn	Bit	Description	Reset Value	M	L
Reserved	[31:6]	Reserved	0	X	X
ImgFormOfCh0	[5:0]	Image format of MIPI Channel 0. If RAW format is image format, image format conversion is not possible. Set scaler bypass mode 0x1E = YUV422 8-bit 0x2A = RAW8 0x2B = RAW10 0x2C = RAW12	0x1E	X	X

NOTE: FRAME END ADDRESS calculation method (useful only TILE 64x32 access mode)

Example) Image pixel size: 720p (1280 x 720) , Format : YCbCr4:2:0 2plane (NV12)

* hor_img_size (width) = 1280byte, ver_img_size (height) = 720

if (hor_img_size % 16 == 0) hor_img_offset = 0

else hor_img_offset = 16 – (hor_img_size % 16)

if (ver_img_size % 16 == 0) ver_img_offset = 0

elsever_img_offset = 16 – (ver_img_size % 16)

if (Luma) { // Y plane

pixel_x = hor_img_size+hor_img_offset = 1280

pixel_y = ver_img_size+ver_img_offset = 720

}

else if (Chroma) { // Cb/Cr plane

pixel_x = hor_img_size+hor_img_offset = 1280

pixel_y = (ver_img_size+ver_img_offset) / 2 = 360

}

1) Luma case

pixel_x_minus = pixel_x - 1 = 1279

pixel_y_minus = pixel_y - 1 = 719 = 1011001111 (binary)

roundup_x = INT (INT ((pixel_x - 1)/16)/8) + 1 = 10

roundup_y = INT(INT((pixel_y - 1)/16)/4) + 1 = 12

if (pixel_y_minus[5] == 0) // pixel_y_minus[5:0]='b 001111 , pixel_y_minus[5]=0

pic_range = pixel_y_minus[14:6] * roundup_x + pixel_x_minus[14:8] + 1 = 11 * 10 + 4 + 1 = 115

else

pic_range = roundup_x * roundup_y

2) Chroma case

pixel_x_minus = pixel_x - 1 = 1279

pixel_y_minus = pixel_y - 1 = 359 = 101100111 (binary)

roundup_x = INT (INT ((pixel_x - 1)/16)/8) + 1 = 10

roundup_y = INT(INT((pixel_y - 1)/16)/4) + 1 = 6

if (pixel_y_minus[5] == 0) // pixel_y_minus[5:0]='b 100111 , pixel_y_minus[5]=1

pic_range = pixel_y_minus[14:6] * roundup_x + pixel_x_minus[14:8] + 1

else

pic_range = roundup_x * roundup_y = 10 * 6 = 60

Thus, each plane frame end address = Base_address[31:0] + {pic_range, 2'b0, 11'b0}

9.4 JPEG

1 OVERVIEW

JPEG is to compress the original raw image and to decompress the JPEG encoded image. It performs all functions required for image compression/ decompression such as Discrete Cosine Transform (DCT), Quantization and Huffman coding. It is composed of control circuit, DCT/ quantization, Huffman codec, marker process block and AHB interface control as shown in Figure 9.4-1. It has control registers inside. It is possible to set the operation modes and conditions such as the Huffman table number and restart interval value into these registers.

2 FEATURES

- Compression/ decompression up to 7168x7168
- Supports following format of compression (Refer to Table 9.4-1)
  - Input raw image: YCbCr4:2:2 or RGB565
  - Output JPEG file: Baseline JPEG of YCbCr4:2:2 or YCbCr4:2:0
  - Supports following format of decompression (Refer to Table 9.4-1)
  - Input JPEG file: Baseline JPEG of YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0, gray
  - Output raw image: YCbCr4:2:2 or YCbCr4:2:0
- Supports general-purpose color-space converter

Table 9.4- 1. JPEG Color Format

Encoding	Input Color Format	YCbCr4:2:2 1 plane (MSB ---------- LSB) (Cr0 - Y1- Cb0 - Y0) RGB565 1 plane (R1-G1-B1-R0-G0-B0)
	Output Color Format	YCbCr4:2:2 YCbCr4:2:0
Decoding	Input Color Format	YCbCr4:4:4 YCbCr4:2:2 YCbCr4:2:0 gray
	Output Color Format	YCbCr4:2:2 1plane (Cr - Y1- Cb - Y0) YCbCr4:2:0 2plane (Y3-Y2-Y1-Y0 , Cr1-Cb1-Cr0-Cb0)

3 BLOCK DIAGRAM

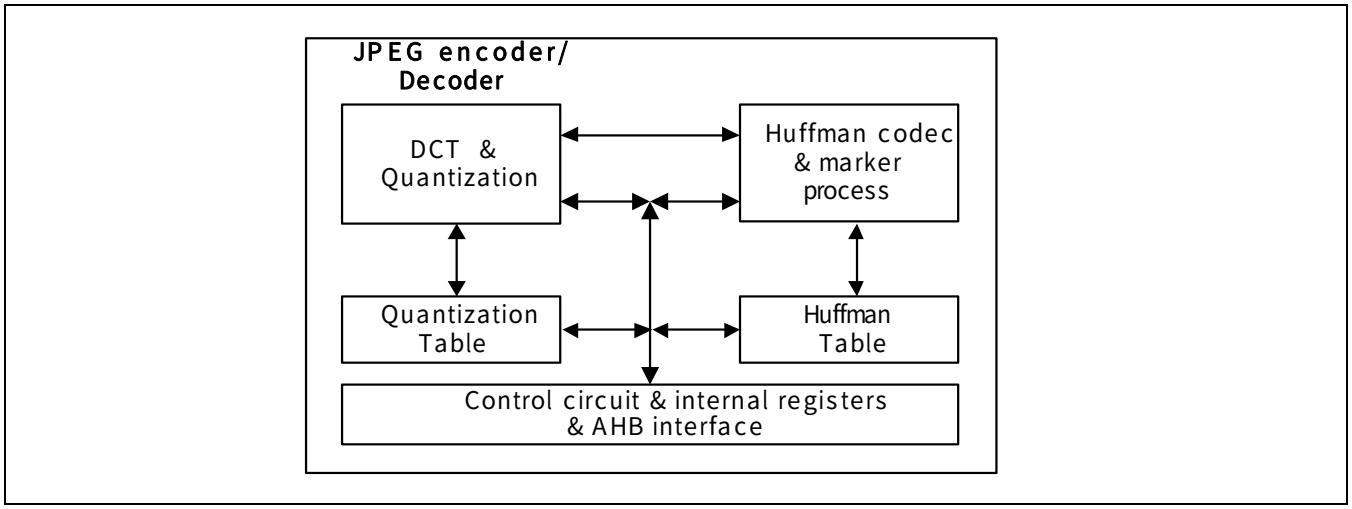


Figure 9.4-1 JPEG Block Diagram

3.1 CONTROL CIRCUIT AND AHB INTERFACE

This block sets and initializes the operation mode. It has registers inside. It is possible to set the operation modes and conditions such as Huffman table number and restart interval value into these registers.

3.2 DCT/ QUANTIZATION

During compression, it transforms 8x8 image data to DCT coefficients. Then the quantization process is performed over DCT coefficients by utilizing the quantization tables. During decompression, dequantization is done and then DCT coefficients is transformed into image data.

3.3 HUFFMAN CODER AND MARKER PROCESS

During compression, Huffman encoding is done according to the Huffman table and marker process makes the JPEG bitstream. During decompression, marker process parses a JPEG file and Huffman decoding is done.

3.4 QUANTIZATION TABLE

It is the place to store quantization tables.

3.5 HUFFMAN TABLE

It is the place to store Huffman tables.

3.6 PERFORMANCE

JPEG IP supports compression/ decompression of image file with size up to 65536x65536.

In case of image file with 320x240 size, decoding of jpeg file requires 3ms and encoding of raw image spends 6ms from start command to final interrupt request under 120MHz operation frequency.

3.7 JPEG HALF CLOCK

JPEG has a half clock. To select this, write 1 or 0 on JPEG Color Mode Register [0] bit, HALF_EN [0].
(Refer to JPEG Color Mode Register on Page 16)

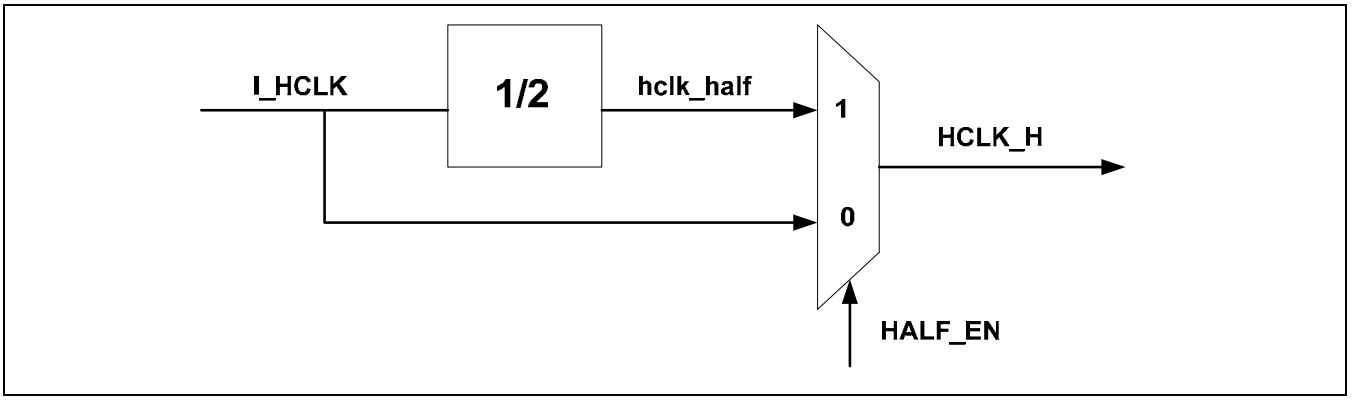


Figure 9.4-2 JPEG Half Clock

4 DESCRIPTION OF SUPPORTED COLOR FORMAT

JPEG supports several color formats during compression/ decompression.

4.1 IN COMPRESSION MODE

Before compression starts, raw image data must be in main memory whose address is specified in IMGADR register. The raw images are stored in interleaved YCbCr4:2:2 or RGB565 color format. It is described visually in Figure 9.4- 3

After compression is complete, result file is baseline JPEG in YCbCr4:2:0 or YCbCr4:2:2 format with interleaved scan. Therefore color space conversion (RGB→YCbCr) and decimation of chrominance component is necessary. JPEG IP has its own color space converter of which coefficients is set by COEF1, COEF2 and COEF3 register. Decimation of JPEG IP is downsampling process. For example, decimation from YCbCr4:2:2 to YCbCr4:2:0 needs 2:1 vertical downsampling for Cb and Cr component as operation in Figure 9.4- 4

4.2 IN DECOMPRESSION MODE

In decompression mode, input file is baseline JPEG in YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0, gray with interleaved scan and output raw image has interleaved YCbCr4:2:2 or YCbCr4:2:0 formats. Therefore for input file with YCbCr format, decimation and interpolation process is done during decompression. In this case, decimation is same as downsampling and interpolation is sample-and-hold (repetition of recent value) process. Each operation is described in Figure 9.4- 3. For input of gray format JPEG file, result YCbCr4:2:2 or YCbCr4:2:0 raw images have Y component which is result of decompression, and Cb and Cr component with "128" value.

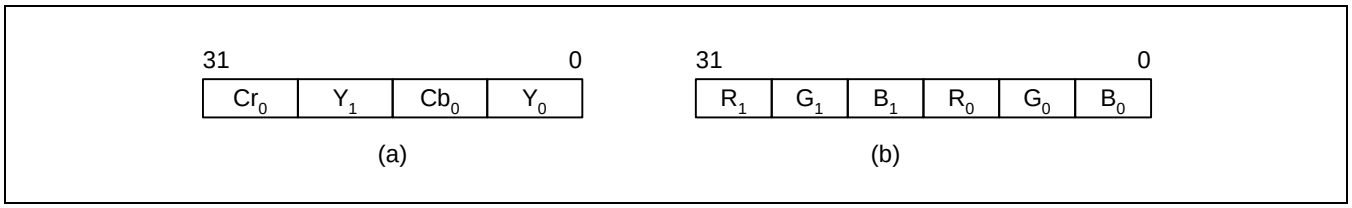


Figure 9.4- 3 Raw Image Format in Memory (a) YCbCr4:2:2 (b) RGB5:6:5

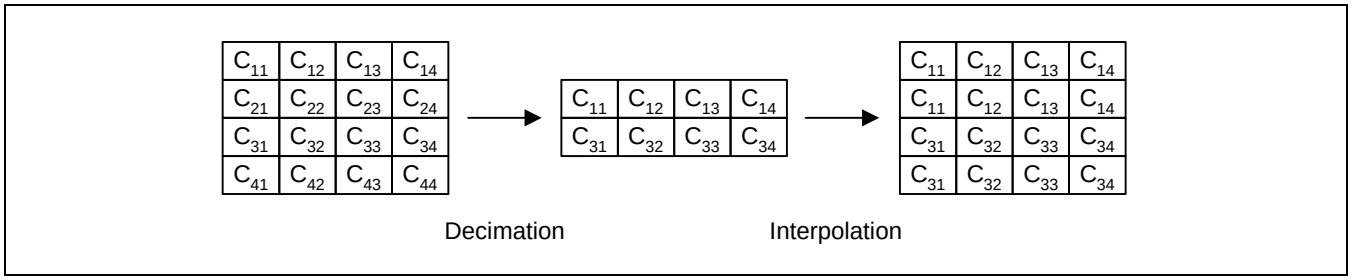


Figure 9.4- 4 Decimation and 1:2 Interpolation in Vertical Direction

Figure 9.4- 5 illustrates the input format of YCbCr4:2:2. This JPEG codec supports type (a).
If input YCbCr4:2:2 format is type (b), decoder does not work.

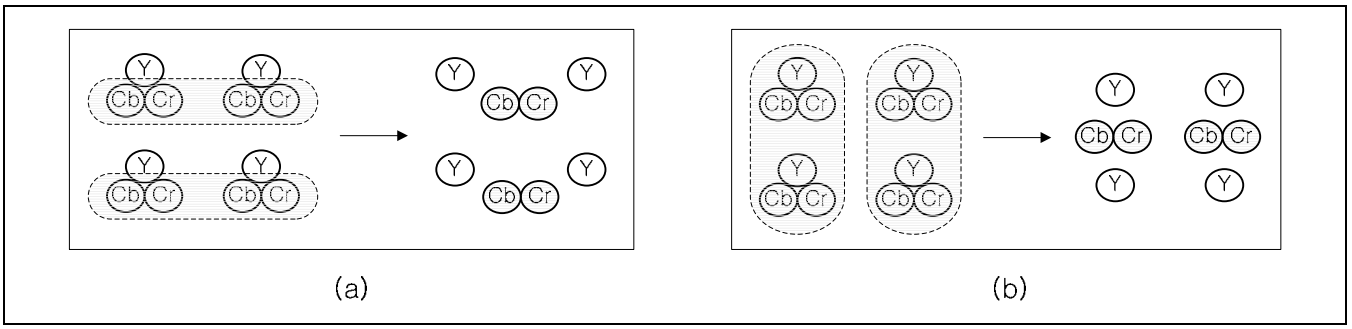


Figure 9.4- 5 YCbCr4:2:2 Color Format

5 PROCESS

5.1 REGISTER ACCESS

The registers are modified

1. After reset, until a new job starts, or
2. After the generation of the process completion interruption signal, until a new job starts.

   Other conditions indicate that the core is in the normal operation, therefore register modification is not allowed.

5.2 TABLE ACCESS

Four Huffman tables (AC & DC, 2 tables for each) and four quantization tables must be configured before compression. To set any quantization table and Huffman table, corresponding table entry register must be accessed first. Then write transfers should follow. To understand the write transfer, refer to Figure 9.4-6. The access order for each table is shown below.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figure 9.4- 6 Access Order in Quantizer Table

5.3 STARTING PROCESS

Process start signal from the scheduler instructs to start compression or decompression process of one picture after setting various registers. After getting the start signal, the core processing starts and then JPGOPR register is read as 1. Operation cannot be guaranteed if the start signal is issued again during processing. Do not set like that.

Table 9.4- 2. Registers That Must be Configured before Compression

Registers	Description	At comp	At Decomp
JPGMOD	Process Mode Register	Essential	Essential
QHTBL	Quantization and Huffman Table Number Register	Essential	--
JPGDRI	Reset Interval Registers	Essential	--
JPGY	Vertical Size Register	Essential	
JPGX	Horizontal Size Register	Essential	
QTBL0	Quantizer Table0 Entries Register.	Essential	--
QTBL1	Quantizer Table1 Entry Register	Essential	--
QTBL2	Quantizer Table2 Entry Register	Essential	--
QTBL3	Quantizer Table3 Entry Register	Essential	--
OUTFORM	Output Color Format of Decompression	--	Essential
DEC_STREAM_SIZE	Input JPEG Stream Byte Size for Decompression	--	Essential
ENC_STREAM_INTSE	Compressed Stream Size Interrupt Setting Register	Essential	--
HDTBL0, HDCTBLG0	DC Huffman Table0 Entry Register	Essential	--
HACTBL0, HACTBLG0	AC Huffman Table0 Entry Register	Essential	--
HDCTBL1, HDCTBLG1	DC Huffman Table1 Entry Register	Essential	--
HACTBL1, HACTBLG1	AC Huffman Table1 Entry Register	Essential	--

Contents of registers in Table 5 changes in following cases.

1. After the registers are written again by user.
2. After reset operation is done.
3. After decompression of arbitrary JPEG file. In this case, the registers have the value from header of input JPEG file after header parsing process.

Except these cases, it is possible to process next picture by only performing the process start signal after process of a picture is completed.

5.4 PROCESS FOR IMAGE SIZE

Size of images contained in JPEG file have to be a specific value which is the multiple of block size because JPEG file is composed of blocks. Size of images contained in JPEG file should be double block size in S5PC100. In other words, raw image (encoding input/ decoding output) is greater than and equal to double block size. If image size in JPEG file header is not multiple of block size, actual image size in the file is the minimum among values which are the multiple of the block size and larger than the value in the header, but decoder shows cropped image with the size in file header. Block size is determined by color format as described in Table 9.4-3.

Table 9.4- 3. Relationship between Block Size and Color Format

Color format	MCU Block size (WxH)	Minimum size
YCbCr4:4:4	8x8	16x8 or 8x16
YCbCr4:2:2	16x8	32x8 or 16x16
YCbCr4:2:0	16x16	32x16 or 16x32
Gray (Y only)	8x8	16x8 or 8x16

1. Decompression

Input JPEG file has information about color format, width and height of the image in the frame header. Those are known to user after header parsing process. Actual raw image size after decompression is the minimum among the values which are the multiple of block size (known from color format) and larger than or equal to the image size in the header. For example, if JPEG file is YCbCr4:2:0 format and its size is 170x170, actual size of decompressed raw image is 176x176.

Proper process such as cropping is needed to display or store the result raw image in the width and height in the file header.

2. Compression

Width and height of input raw image have to be the multiple of the block size corresponding to the output JPEG color format. If input raw image has arbitrary size, the size must be modified to the multiple of block size by padding process. The modified size is the minimum values which are the multiple of the block size and larger than or equal to the original value. However, register setting value of width and height for compression must be the original value.

5.5 PROCESS FOR INPUT STREAM SIZE

In case of decompression of an illegal JPEG stream, JPEG IP does not recognize the end of the stream if some important markers are damaged. Therefore, it is necessary to notify the input stream size to JPEG IP.

Before start of decompression, you must write byte size of input stream to DEC_STREAM_SIZE register. The register must be set for each decompression because previous written value does not remain.

If decompression does not finish within the given file size, error interrupt occurs. Stream data beyond the given size is ignored.

5.6 INTERRUPT SIGNAL

Interrupt signal is generated under the following conditions, and the JPGINTST register identifies causes:

1. Compression or decompression process for one picture is finished,
2. During decompression, if the registers storing the image size and sampling factor are ready to be read out after the header analysis.
3. Internal timer counting ends before finish of compression or decompression.
4. During compression, the byte size of output stream is larger than the predefined bound size in ENC_STREAM_BOUND.

With interrupt case 1, the normal process is finished. To clear the interrupt request, read the JPGINTST register and JPGOPR register. If there is no encoding or decoding error, JPGINTST must be read as 0x40. If another value is read, the operation result may not be correct.

To clear the interrupt case 2 is also done by reading JPGINTST and JPGOPR. If there is no error during header parsing, it will be read as 0x08. The interrupt case 2 indicates that the decompression process is paused. Process restarts by getting restart signal from software.

If interrupt case 3 occurs, TIMER_INT_STAT is read as 1 and it is cleared by writing 1 in TIMER_INT_STAT register. In this case, JPEG needs reset or S/W reset before next operation.

If interrupt case 4 occurs, ENC_STREAM_INT_STAT is read as 1, JPEG operation is stopped and there is no further memory access by JPEG. This interrupt is cleared by writing 1 in ENC_STREAM_INT_STAT register. In this case, JPEG needs reset or S/W reset before next operation.

5.7 INTERRUPT SETTING

JPGINTSE register specifies whether the interrupt is allowed or not after header parsing and for illegal input file for decompression.

To allow interrupt request after header analysis during decompression, set HEAD_INT_EN to 1, before start the decompression process. If this interrupt occurs, JPEG pauses the decompression process (During the pause, JPGOPR has value "1" because the process is not complete).

To deal with an illegal input JPEG file for decompression, set ERR_INT_EN to 0x7, before start the decompression process. If input file is illegal, JPGINTST register shows abnormal value at the end of operation (Normal value is 0x8 after header parsing and 0x40 after the end of operation).

To enable timer interrupt, set TIMER_INT_EN to 1 before start or restart.

To enable compressed stream size interrupt, set ENC_STREAM_INT_EN to 1 before starting compression.

5.8 S/W RESET

JPEG IP has a register for S/W reset. S/W reset is executed by following 3 steps.

1. Set 1 in the SW_RESET register.
2. Wait until SW_RESET register value changes to 0.
3. After the value changes to 0, set the proper register for next operation and start the operation.

If JPEG IP is terminated abnormally or holds operation, S/W reset is needed to start new operation.

5.9 MARKER PROCESS

The following markers are generated during compression.

Table 9.4- 4. Markers in JPEG Codec

Marker	Codes (hex)	Description
SOI	FFD8	Start of image
SOF_0	FFC0	Baseline JPEG
SOS	FFDA	Start of scan
DQT	FFDB	Define quantization table
DHT	FFC4	Define Huffman table
DRI	FFDD	Define restart interval
RSTm	FFD0~FFD7	Restart with module 8 count "m"
EOI	FFD9	End of image

The markers in Talbe 9.4-4 are subject to process during decompression. The other markers except SOF1~SOFF and JPG are ignored.

5.10 BITSTREAM OF COMPRESSED FILE

The created JPEG bit stream is shown below. In the figure, ECS is an acronym of 'entropy-coded segment' which is a sequence of entropy coded bytes.

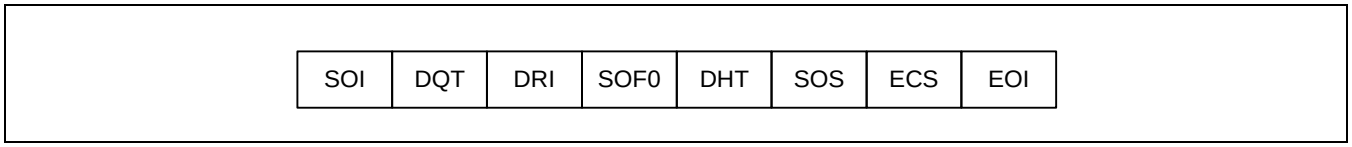


Figure 9.4- 7 Bitstream of Compressed File

5.11 JPEG COMPRESSION FLOW

This is a pseudo instruction set that indicates the write or read on specific registers. It is assumed that the Huffman and quantization tables were written already.

// JPEG encoder initialization

Write JPGCMOD	0x21	*// Mode selection and core clock setting*
Write JPGMOD	0x1	*// Encoding and YCbCr4:2:2*
Write JPGDRI	0x0	*// No DRI if 0. Set an appropriate value.*
Write QHTBL	0x0	*// Choose the appropriate table index for Huffman & Quantization tables.*
Write JPGY	0x40	*// Vertical resolution*
Write JPGX	0x40	*// Horizontal resolution*
Write IMGADR	0x1000_0000	*// Address for an image to compress*
Write JPGADR	0x1001_0000	*// Address for the compressed JPEG file*

Write COEF1 0x4D_971E *// Color converter coefficients*

Write COEF2 0x2c_5783 *// Color converter coefficients*

Write COEF3 0x83_6e13 *// Color converter coefficients*

// Encoding start.

Write JSTART 0x1

// After interrupt is detected, clear the pending register

Read JPGCNT *// Read the file size in bytes*

Read JPGINTST *// It must be read 0x40.*

Read JPGOPR *// It must be read zero.*

5.12 JPEG DECOMPRESSION FLOW

This is a pseudo instruction set that indicates the write or read on specific registers.

// JPEG decoder initialization

Write JPGCMOD 0x21 *// Core clock setting*

Write JPGMOD 0x8 *// Decoding mode*

Write JPGINTSE 0x78 *// Enable interrupt after header analysis and for illegal stream*

Write OUTFORM 0x1 *// Output raw image is YCbCr4:2:0*

Write IMGADR 0x1000_0000 *// Address for a decompressed raw image*

Write JPGADR 0x1001_0000 *// Address for a JPEG file to decompress*

Write DEC_STREAM_SIZE 0x0000_1000 *// Byte size of input JPEG file*

// Header parsing start

Write JSTART 0x1

// After interrupt is detected, clear the pending register

Read JPGINTST *// It must be read 0x8.*

// Resume the decoding operation

Write JRSTART 0x1

// After interrupt is detected, clear the pending register

Read JPGINTST *// It must be read 0x40.*

Read JPGOPR *// It must be read 0x0.*

6 REGISTER DESCRIPTION

JPEG has the control registers as shown in Table 9.4- 1and table assignment in Table 9.4-

Table 9.4- 5. JPEG Codec Control Registers

Register	Address	R/W	Description	Reset Value
JPGMOD	0xEE50_0000	R/W	Sub-sampling Mode Register	0x0000_0000
JPGOPR	0xEE50_0004	R	Operation Status Register	0x0000_0000
QHTBL	0xEE50_0008	R/W	Quantization Table Number Register and Huffman Table Number Register	0x0000_0000
JPGDRI	0xEE50_000C	R/W	MCU, which inserts RST marker	0x0000_0000
JPGY	0xEE50_0010	R/W	Vertical Resolution	0x0000_0000
JPGX	0xEE50_0014	R/W	Horizontal Resolution	0x0000_0000
JPGCNT	0xEE50_0018	R	The amount of the compressed data in bytes	0x0000_0000
JPGINTSE	0xEE50_001C	R/W	Interrupt Setting Register	0x0000_0000
JPGINTST	0xEE50_0020	R	Interrupt Status Register	0x0000_0000
Reserved	0xEE50_0024 ……. 0xEE50_004C	-	-	-
IMGADR	0xEE50_0050	R/W	Source or Destination Image Address	0x0000_0000
Reserved	0xEE50_0054	-	-	-
JPGADR	0xEE50_0058	R/W	Source or Destination JPEG File Address	0x0000_0000
COEF1	0xEE50_005C	R/W	Coefficient Values for RGB ↔ YCbCr Converter	0x0000_0000
COEF2	0xEE50_0060	R/W	Coefficient Values for RGB ↔ YCbCr Converter	0x0000_0000
COEF3	0xEE50_0064	R/W	Coefficient Values for RGB ↔ YCbCr Converter	0x0000_0000
JPGCMOD	0xEE50_0068	R/W	Mode Selection and Core Clock Setting	0x0000_0021
JPGCLKCON	0xEE50_006C	R/W	Power On/ Off and Clock Down Control	0x0000_0002
JSTART	0xEE50_0070	W	Start Compression or Decompression	0x0000_0000
JRSTART	0xEE50_0074	W	Restart Decompression after Header Analysis	0x0000_0000
SW_RESET	0xEE50_0078	R/W	S/W Reset	0x0000_0000

TIMER_SE	0xEE50_007C	R/W	Internal Timer Setting Register	0x7FFF_FFFF
TIMER_ST	0xEE50_0080	R/W	Internal Timer Status Register	0x7FFF_FFFF
COMSTAT	0xEE50_0084	R	Command Status Register	0x0000_0000
OUTFORM	0xEE50_0088	R/W	Output Color Format of Decompression	0x0000_0000
VERSION	0xEE50_008C	R	Version Register	0x0000_0003
Reserved	0xEE50_0090	-	-	-
DEC_STREAM_SIZE	0xEE50_0094	R/W	Input JPEG Stream Byte Size for Decompression	0x1FFF_FFFE
ENC_STREAM_INTSE	0xEE50_0098	R/W	Compressed Stream Size Interrupt Setting Register	0x00FF_FFE0
ENC_STREAM_INTST	0xEE50_009C	R/W	Compressed Stream Size Interrupt Status Register	0x0000_0000

6.1 JPEG Mode Register (JPGMOD, R/W, Address = 0XEE50_ 0000)

JPGMOD	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
PROC_MODE	[3]	Process mode. 0 = Compression process. 1 = Decompression process.	0
SUBSAMPLING_MODE	[2:0]	Sub sampling mode 0x0 = chroma 4:4:4 format 0x1 = chroma 4:2:2 format. 0x2 = chroma 4:2:0 format 0x3 = Gray format (Single Component) Others are reserved. During decompression, these are read-only. During compression, only 0x1 or 0x2 are allowed.	0

6.2 JPEG Operation Status Register (JPGOPR, R, Address = 0XEE50_ 0004)

JPGOPR	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
JPGOPR	[0]	0 = JPEG is not operating. 1 = JPEG is operating.	0

6.3 Quantization Table And Huffman Table Number Register (QHTBL, R/W, Address = 0XEE50_ 0008)

QHTBL	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0
QT_NUM3	[13:12]	Quantization table number for Cr component.	0
QT_NUM2	[11:10]	Quantization table number for Cb component	0
QT_NUM1	[9:8]	Quantization table number for Y component.	0
Reserved	[7:6]	Reserved	0
HT_NUM3_AC	[5]	Huffman table number for Cr component AC.	0
HT_NUM3_DC	[4]	Huffman table number for Cr component DC.	0
HT_NUM2_AC	[3]	Huffman table number for Cb component AC.	0
HT_NUM2_DC	[2]	Huffman table number for Cb component DC.	0
HT_NUM1_AC	[1]	Huffman table number for Y component AC.	0
HT_NUM1_DC	[0]	Huffman table number for Y component DC.	0

6.4 JPEG Restart Interval Register (JPGDRI, R/W, Address = 0XEE50_ 000C)

JPGDRI	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
JPGDRI	[15:0]	It is a restart interval that identifies the distance between two adjacent Restart Maker (RST) in terms of Minimum Coded Unit (MCU). It is valid in compression mode. If JPGDRI is set to 0, Define Restart Interval Marker (DRI) and RST is not inserted.	0

6.5 JPEG Vertical Resolution Register (JPGY, R/W, Address = 0XEE50_ 0010)

JPGY	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
VER_RES	[15:0]	Image size value in the vertical direction. You are not allowed to Set 0. This register is read-only during decompression.	0

6.6 JPEG Horizontal Resolution Register (JPGX, R/W, Address = 0XEE50_ 0014)

JPGX	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
HOR_RES	[15:0]	Image size value in the horizontal direction. You are not allowed to Set 0. This register is read-only during decompression..	0

6.7 JPEG Byte Count Register (JPGCNT, R, Address = 0XEE50_ 0018)

JPGCNT	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
BYTE_CNT	[23:0]	Compressed JPEG file size in the number of bytes.	0x00_0000

6.8 JPEG Interrupt Setting Register (JPGINTSE, R/W, Address = 0XEE50_ 001C)

JPGINTSE	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
ERR_INT_EN	[6:4]	It is valid in decompression mode. 0x7 = Enable interrupt for illegal JPEG stream. If it is not set 0x7, interrupt is not generated in case of decompression of illegal JPEG stream. It is recommended to Set 0x7.	0
HEAD_INT_EN	[3]	It is valid in decompression mode. 0 = Disables interrupt after header analysis. 1= Enables interrupt after header analysis. If enabled, it is possible to read the image size and sampling factor value in the result of header analysis. It is recommended to Set 1.	0
INT_EN	[2:0]	Write 0x0 to enable interrupt.	0

6.9 JPEG Interrupt Status Register (JPGINTST, R, Address = 0XEE50_ 0020)

JPGINTST	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
RESULT_STAT	[6]	Result status. 0 = Processing was finished abnormally. 1= Processing was done normally	0
Reserved	[5]	Reserved	0
STREAM_STAT	[4]	Bitstream error status. Valid during decompression only. 0 = There is no syntax error on the compressed file. 1 = There is syntax error on the compressed file.	0
HEADER_STAT	[3]	Header status. Valid during decompression only. 0 = Image size and sampling factor value cannot be read. 1= Image size and sampling factor value can be read.	0
Reserved	[2:0]	Reserved	0

6.10 Raw Image Data R/W Address Register (IMGADR, R/W, Address = 0XEE50_ 0050)

IMGADR	Bit	Description	Reset Value
IMG_ADR	[31:0]	It is start address of raw image data. Value for this register has to be multiple of 32. In compression mode, raw image before compression is read from this address. In decompression mode, raw image after decompression is stored from address.	0

6.11 JPEG File R/W Address Register (JPGADR, R/W, Address = 0XEE50_ 0058)

JPGADR	Bit	Description	Reset Value
JPG_ADR	[31:0]	It is start address of JPEG file data. Value for this register has to be multiple of 32. In compression mode, JPEG file after compression is stored from this address. In decompression mode, JPEG file before compression is read from this address.	0

6.12 Coefficient for RGB-TO-YCBCR Converter Register (COEF1, R/W, Address = 0XEE50_ 005C)

COEF1	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
COEF11	[23:16]	Coefficient value of COEF11	0
COEF12	[15:8]	Coefficient value of COEF12	0
COEF13	[7:0]	Coefficient value of COEF13	0

6.13 Coefficient for RGB-TO-YCBCR Converter Register (COEF2, R/W, Address = 0XEE50_ 0060)

COEF2	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
COEF21	[23:16]	Coefficient value of COEF21	0
COEF22	[15:8]	Coefficient value of COEF22	0
COEF23	[7:0]	Coefficient value of COEF23	0

6.14 Coefficient for RGB-TO-YCBCR Converter Register (COEF3, R/W, Address = 0XEE50_ 0064)

COEF3	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
COEF31	[23:16]	Coefficient value of COEF31	0
COEF32	[15:8]	Coefficient value of COEF32	0
COEF33	[7:0]	Coefficient value of COEF33	0

The expression of 8-bit COEFxx is shown in Table 9.4-3. For example, if COEFxx is set as 1100_0000b, the decimal value of COEFxx is 0.75. (= 0.5 + 0.25)

Table 9.4- 6. Bitwise Expression of COEFxx

Bit	7	6	5	4	3	2	1	0
Value	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125	0.00390625

6.15 JPEG Color Mode Register (JPGCMOD, R/W, Address = 0XEE50_ 0068)

JPGCMOD	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
MOD_SEL	[7:5]	Color space of input raw image 0x1 = YCbCr4:2:2 0x2 = RGB 565 Others are reserved.	1
Reserved	[4:2]	It must be set 0x0.	0
MODE_Y16	[1]	Y_16 selector for Y component 0 = c1 = 0 1 = c1 = 16 c1 is used in RGB-to-YCbCr converter, refer to the expression on page 13.	0
HALF_EN	[0]	Clock for JPEG core 0 = Full system clock 1= Half system clock JPEG core includes DCT, quantization, and Huffman coding and marker process. Bus interface always operate on full system clock regardless of this value.	1

6.16 JPEG Clock Control Register (JPGCLKCON, R/W, Address = 0XEE50_ 006C)

JPGCLKCON	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0
CLK_DOWN_READY	[1]	0 = Clock is enabled. 1 = JPEG is ready for disabling clock (Default). This value is changed to 1 if POWER_ON is set as 0 and JPEG is not working. This value is changed to 0 if POWER_ON is set as 1. If this value is 1, JSTART command is ineffective. This bit is read only.	1
POWER_ON	[0]	0 = Disables Clock (Default). 1 = Activates Clock.	0

6.17 JPEG Start Register (JSTART, W, Address = 0XEE50_ 0070)

JSTART	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
JSTART	[0]	To start compression/ decompression, set this value to 1. After one clock, it is cleared with 0 internally. Before starting operation, you must set essential registers.	0

6.18 JPEG Restart Register (JRSTART, W, Address = 0XEE50_ 0074)

JRSTART	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
JRSTART	[0]	Decompression of JPEG pauses after header analysis. To restart decompression from pause, set this value to 1. After one clock, it is cleared with 0 internally. Before restarting operation, take information of image from proper registers.	0

6.19 JPEG SW Reset Register (SW_RESET, R/W, Address = 0XEE50_ 0078)

SW_RESET	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
SW_RESET	[0]	Writing 1 resets JPEG IP. Before reset finishes, its value is kept as 1. After reset is done, it is cleared with 0 internally. Therefore it is necessary to check this value is 0 before setting registers and starting operation.	0

6.20 JPEG Timer Setting Register(TIME_SE, R/W, Address = 0XEE50_ 007C)

TIMER_SE	Bit	Description	Reset Value

TIMER_INT_EN	[31]	0 = Disables Interrupt by timer. 1= Enables Interrupt by timer.	0
TIMER_INIT	[30:0]	Target counting value is stored in this register. After start or restart, timer starts to down-count from this value to 0.	0x7FFF_FFFF

6.21 JPEG Timer Status Register (TIMER_ST, R/W, Address = 0XEE50_ 0080)

TIMER_ST	Bit	Description	Reset Value
TIMER_INT_STAT	[31]	Timer interrupt status. If timer interrupt is enabled and timer counting value reaches 0, it is set 1. Writing 1 clears this value. Writing 0 has no effect.	0
TIMER_CNT	[30:0]	Timer counting value. If start or restart, it is initiated by TIMER_INIT value and starts to down-count. If JPEG operation finishes before end of counting, it holds the counter value at that time. This bit is read only.	0x7FFF_FFFF

6.22 JPEG Command Status Register (COMSTAT, R, Address = 0XEE50_ 0084)

COMSTAT	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0
CUR_PROC_MODE	[1]	Current process mode. It is renewed if you write "1" on JSTART or JRSTART register. 0 = Compression process mode 1 = Decompression process mode	0
CUR_COM_MODE	[0]	Current command mode. It is renewed if you write "1" on JSTART or JRSTART register. 0 = After user writes "1" on JSTART register. 1 = After user writes "1" on JRSTART register.	0

6.23 JPEG Decompression Output Format Register (OUTFORM, R/W, Address = 0XEE50_ 0080)

OUTFORM	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
DEC_OUT_FORMAT	[0]	Output color format of decompressed raw image during decompression. 0 = YCbCr 4:2:2 1 = YCbCr 4:2:0	0

6.24 JPEG Version Register (VERSION, R, Address = 0XEE50_ 008C)

VERSION	Bit	Description	Reset Value
VERSION	[31:0]	Version Register	0x0000_0003

6.25 JPEG Decompression Input Stream Size Register (DEC_STREAM_SIZE, R/W, Address = 0XEE50_ 0094)

DEC_STREAM_SIZE	Bit	Description	Reset Value
Reserved	[31:29]	Reserved	0
DEC_STREAM_SIZE	[28:0]	Input JPEG stream size in the number of bytes for decompression. It has to be set before start of each decompression operation. Set 0 or 536,870,911(0x1FFF_FFFF) is prohibited.	0x1FFF_FFFE

6.26 JPEG Compressed Stream Size Interrupt Setting Register (ENC_STREAM_INTSE, R/W, Address = 0XEE50_ 0098)

TIMER_SE	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0
ENC_STREAM_INT_EN	[24]	0 = Disables Compressed stream size interrupt. 1 = Enables Compressed stream size interrupt.	0
ENC_STREAM_BOUND	[23:0]	The upper bound of the byte size of output compressed stream is stored in this register. This value should be multiple of 32.	0xFF_FFE0

6.27 JPEG Compressed Stream Size Interrupt Status Register (ENC_STREAM_INTST, R/W, Address = 0XEE50_ 009C)

TIMER_ST	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
ENC_STREAM_INT_STAT	[0]	Compressed stream size interrupt status. If the byte size of output compressed stream is larger than the size predefined in ENC_STREAM_BOUND, it is set 1. Writing 1 clears this value. Writing 0 has no effect.	0

6.28 JPEG HUFFMAN AND QUANTIZATION REGISTER TABLES

Each data uses the least significant 8 bits of 32-bit register.

Table 9.4- 7. JPEG Codec Table Assignment

Register	Address	R/W	Description	Reset Value
QTBL0	0xEE50_0400 0xEE50_0404 ……. 0xEE50 _04FC	W	Quantization of table number 0 (64 data with the distance of 4 on address)	0x0000 0000 for 64 each data
QTBL1	0xEE50_0500 0xEE50_0504 …… 0xEE50 _05FC	W	Quantization of table number 1 (64 data with the distance of 4 on address)	0x0000 0000 for 64 each data
QTBL2	0xEE50_0600 0xEE50_0604 …… 0xEE50_06FC	W	Quantization of table number 2 (64 data with the distance of 4 on address)	0x0000 0000 for 64 each data
QTBL3	0xEE50_0700 0xEE50_0704 …… 0xEE50_07FC	W	Quantization of table number 3 (64 data with the distance of 4 on address)	0x0000 0000 for 64 each data
HDCTBL0	0xEE50_0800 0xEE50_0804 …… 0xEE50_083C	W	JPEG DC Huffman Table 0 Register The number of code per code length (16 data with the distance of 4 on address)	-
HDCTBLG0	0xEE50_0840 0xEE50_0844 …… 0xEE50_086C	W	JPEG DC Huffman Table 0 Register Group number of the order for occurrence (12 data with the distance of 4 on address)	-

Register	Address	R/W	Description	Reset Value
HACTBL0	0xEE50_0880 0xEE50_0884 …… 0xEE50_08BC	W	JPEG AC Huffman Table 0 Register The number of code per code length (16 data with the distance of 4 on address)	-
HACTBLG0	0xEE50_08C0 0xEE50_08C4 …… 0xEE50_0B44	W	JPEG AC Huffman Table 0 Register Group number of the order for occurrence/ Group number (162 data with the distance of 4 on address)	-
HDCTBL1	0xEE50_0C00 0xEE50_0C04 …… 0xEE50_0C3C	W	JPEG DC Huffman Table 1 Register The number of code per code length (16 data with the distance of 4 on address)	-
HDCTBLG1	0xEE50_0C40 0xEE50_0C44 …… 0xEE50_0C6C	W	JPEG DC Huffman Table 1 Register Group number of the order for occurrence (12 data with the distance of 4 on address)	-
HACTBL1	0xEE50_0C80 0xEE50_0C84 ….. 0xEE50_0CBC	W	JPEG AC Huffman Table 1 Register The number of code per code length (16 data with the distance of 4 on address)	-
HACTBLG1	0xEE50_0CC0 0xEE50_0CC4 …… 0xEE50_0F44	W	JPEG AC Huffman Table 1 Register Group number of the order for occurrence/ Group number (162 data with the distance of 4 on address)	-

NOTES

9.5 2D-ACCELERATOR

1 OVERVIEW

2D-ACCELERATOR is a 2D graphics accelerator that supports three types of primitive drawings namely; Line/Point Drawing, Bit Block Transfer (BitBLT) and Color Expansion (Text Drawing).

Rendering a primitive has two-steps: 1) configure the rendering parameters, such as foreground color and the coordinate data, by setting the drawing-context registers; 2) start the rendering process by setting the relevant command registers accordingly.

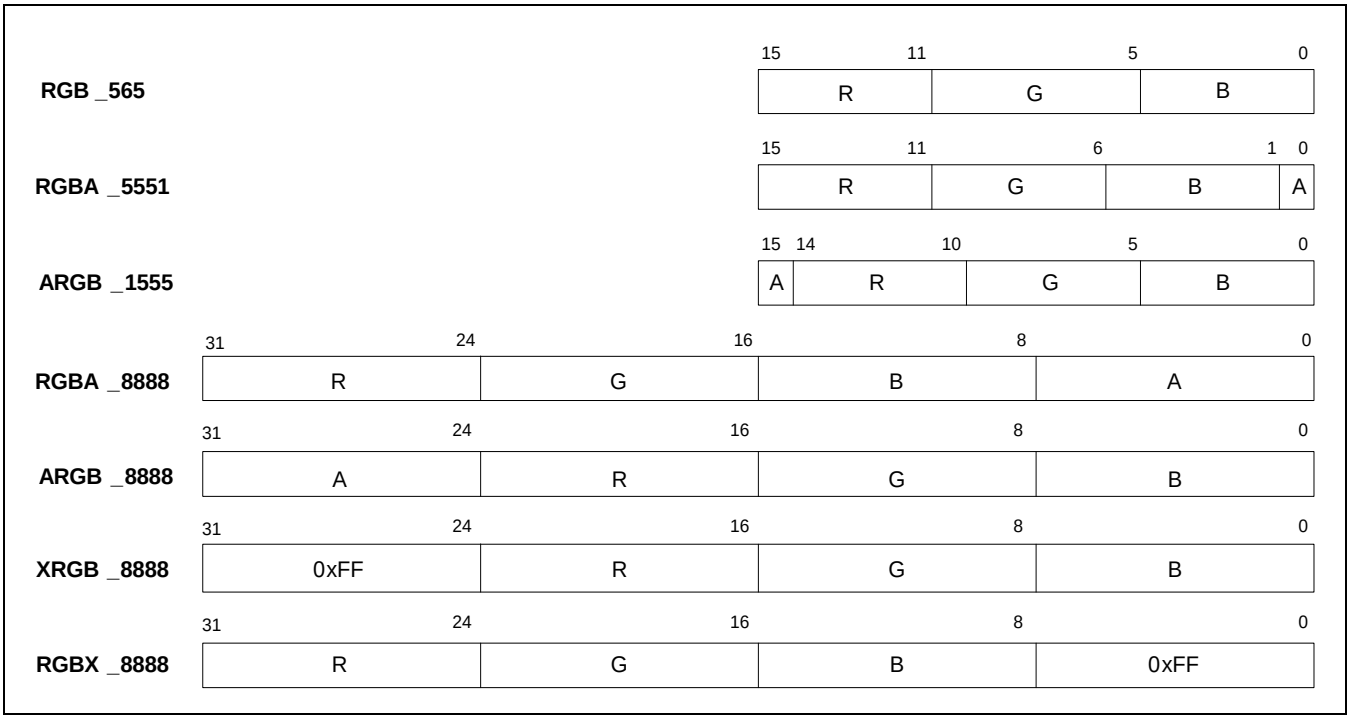
2 FEATURES

- **Primitives**
  - ♦ Line/Point Drawing
    - – DDA (Digital Differential Analyzer) algorithm
    - – Do-Not-Draw Last Point support
  - ♦ BitBLT
    - – stretched BitBLT support (Nearest sampling)
    - – Memory to Screen
    - – Host to Screen
  - ♦ Color Expansion
    - – Memory to Screen
    - – Host to Screen
- **Per-pixel Operation**
  - ♦ Maximum 2040*2040 image size
  - ♦ Window Clipping
  - ♦ 90°/180°/270°/X-flip/Y-flip Rotation
  - ♦ Total of 256 3-operand Raster Operation (ROP)
  - ♦ Alpha Blending
    - – Alpha Blending with a user-specified 256-level alpha value
    - – Per-pixel Alpha Blending
  - ♦ 8x8x16-bpp pattern drawing

- **Data Format**
  - ♦ Supports 16/24/32-bpp color format
  - ♦ Supports YUV input (4:2:2, 2-planar)
  - ♦ Supports Little/Big Endian
  - ♦ 11.11 fixed point format for coordinate data

3 COLOR FORMAT CONVERSION

2D-ACCELERATOR supports seven color formats namely: RGB_565, RGBA_5551, ARGB_1555, RGBA_8888, ARGB_8888, XRGB_8888, and RGBX_8888. The structure of each color format is illustrated in the figure below.



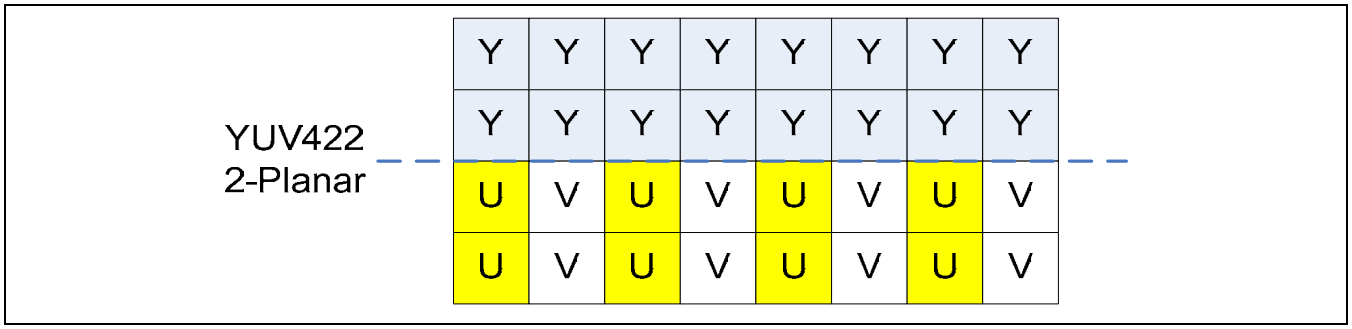
The internal computations use ARGB_8888 format. All data's (source, destination, foreground, background, blue-screen, and pattern) are converted to ARGB_8888 format before computation. The results are converted to the color format specified by DEST_COLOR_MODE_REG before writing to frame buffer.

If a 16-bit color data is converted to 32-bit, the data of each field is shifted (8 – *x*) bits to left, where *x* is the bit-width of the field. The least significant *x* bits of the new field data are padded with the most significant *x* bits of the original field data. For example, if the R-value in RGB_565 format is 5'b11010, it is converted to 8'b11010110, with three LSBs padded with three MSBs (3'b110) from the original R value. Note that, the A field in RGBA_5551 and ARGB_1555 only has one bit, so it is converted to either 8'b00000000 or 8'b11111111 (A=1'b1).

If a 32-bit color data is converted to 16-bit, the data of each field is truncated to *x* bits, where *x* is the bit-width of the field in the new color format. For example, if the R value in RGBA_8888 format is 8'b11001110, it is converted to 5'b11001 in the RGB_565 format, with the three LSBs discarded. **Note:** If the A field of the 32-bit color data is not 0, the A field in RGBA_5551 and ARGB_1555 will be 1'b1; otherwise, 1'b0.

2D-ACCELERATOR also supports YUV input (format: YUV422, 2-planar). The memory allocation of a 16-pixel image of YUV422 format is illustrated in the figure below. **Note:** If YUV format is used, the source image

horizontal resolution must be an even number.



4 COMMAND FIFO

2D-ACCELERATOR has a 32-word command FIFO. All data written to command registers and parameter setting registers is written to the FIFO first. If the graphics engine is idle (no command is executed), the data is written to the designated register in one cycle; otherwise, the data is stored in the FIFO and waits to be dispatched after the current rendering process completes.

It is user's responsibility to make sure that the data written to the FIFO do not exceed its maximum capacity. User can monitor the number of data entries used in FIFO by reading FIFO_USED bits in FIFO_STAT_REG, or request graphics engine to give an interrupt signal if the number of entries in FIFO reaches a certain level by setting FIFO_INTC_REG and E bit in INTEN_REG.

5 RENDERING PIPELINE

The rendering pipeline of 2D-ACCELERATOR is illustrated in Figure 9.5-1. The functionality and related registers of each stage are introduced in detail in the rest of this chapter.

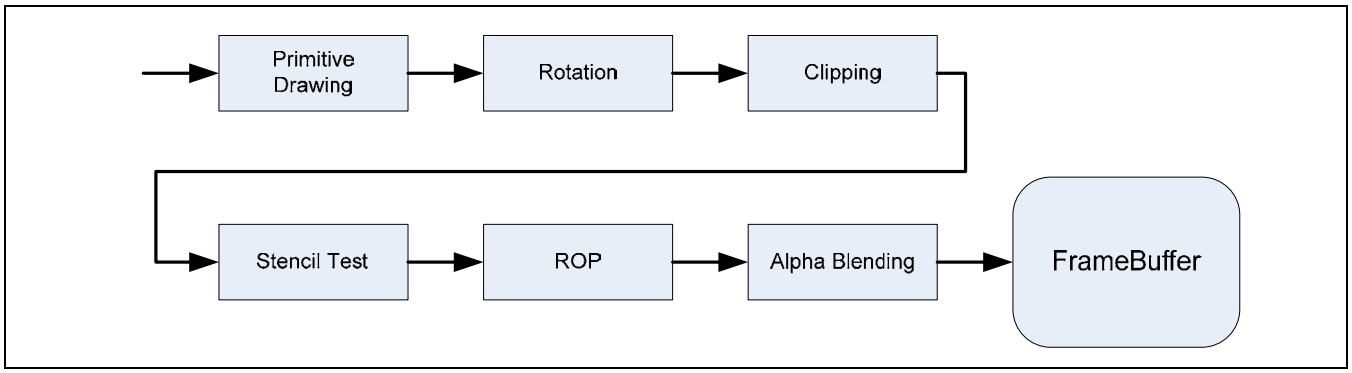


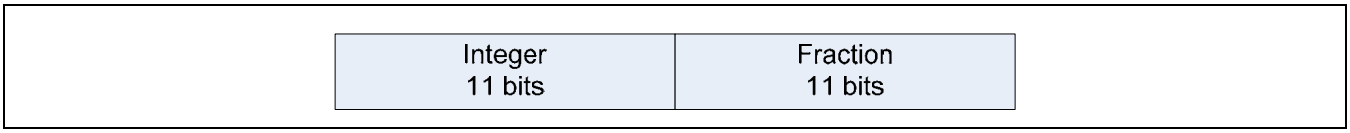
Figure 9.5-1 2D-ACCELERATOR Rendering Pipeline

5.1 PRIMITIVE DRAWING

Primitive Drawing determines the pixels to fill, and pass their coordinates to the next stage for further operations. 2D-ACCELERATOR supports three types of primitive drawing: 1) Line/ Point drawing; 2) Bit block transfer; 3) Color expansion.

5.1.1 Line/Point Drawing

Line Drawing renders a line between the starting point (sx, sy) and the ending point (ex, ey) specified by the user. If the distance of these two points along y axis is greater than that along x axis (|ey - sy| > |ex - sx|), the Major Axis should be set to y-axis; otherwise, x-axis. If y-axis is the Major Axis, the y-coordinate of a pixel on the line is increased or decreased by 1 from its preceding pixel, while the x-coordinate increased or decreased by X-INCR (smaller than 1). In the same vein, if x-axis is the Major Axis, the x-coordinate is increased or decreased by 1 while the y-coordinate by Y-INCR. Note that X-INCR and Y-INCR should be given in 2's complement format as shown below. . For example, 1.75 can be represented by 00000000001.11000000000, while -1.75 by 1111111110.01000000000.



Related Registers

COORD_0	Coordinate of the starting point
COORD_2	Coordinate of the ending point (Ignored if a point is rendered).
X-INCR	X increment value (Ignored if x-axis is the Major Axis or a point is rendered). X-INCR = (ex-sx)/ |ey - sy|
Y-INCR	Y increment value (Ignored if y-axis is the Major Axis or a point is rendered). Y-INCR = (ey – sy)/ |ex - sx|
FG_COLOR	The color of the drawn line/ point
CMD0_REG	Configure the line/ point drawing parameters, such as whether the Major-Axis is x-axis or y-axis, whether to draw a line or a point, and etc. Note that writing to this register starts the rendering process.

5.1.2 Bit Block Transfer

A Bit Block Transfer is a transformation of a rectangular block of pixels. Typical applications include copying the off-screen pixel data to frame buffer, combining to bitmap patterns by Raster Operation, changing the dimension of a rectangular image etc.

On-Screen Rendering

On-screen bit block transfer copies a rectangular block of pixels on screen to another position on the same screen. On-screen rendering has the following restriction:

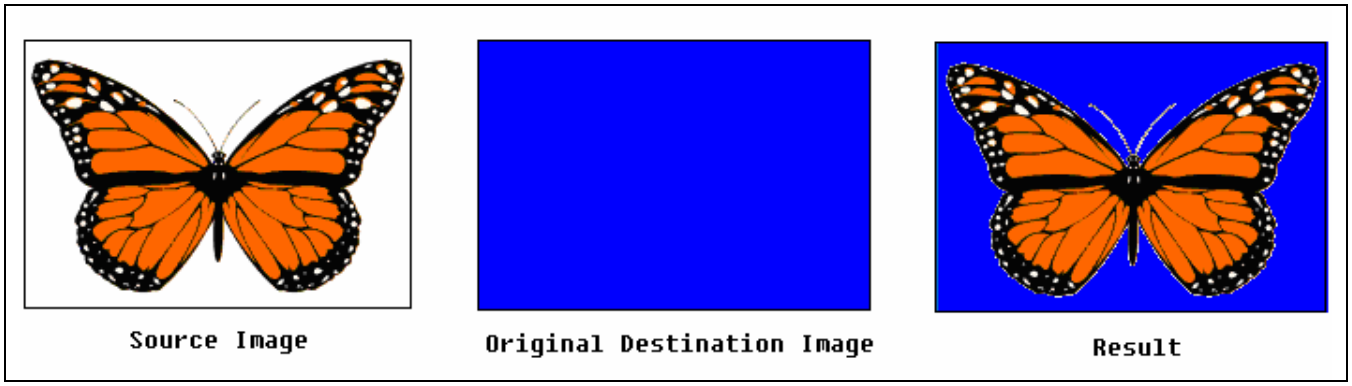
1) SRC_BASE_ADDR = DEST_BASE_ADDR
2) SRC_HORI_RES_REG = DEST_HORI_RES_REG
3) SRC_COLOR_MODE = DEST_COLOR_MODE
4) If the destination block overlaps with the source block, stretch mode and rotation must not be used.

Off-Screen Rendering

Off-screen bit block transfer copies pixel data from off-screen memory to frame buffer. Color space conversion is performed automatically if SRC_COLOR_MODE differs from DEST_COLOR_MODE. YUV 4:2:2 input is supported.

Transparent Mode

2D-ACCELERATOR renders image in Transparent Mode. In this mode, the pixels having the same color with blue screen color (BS_COLOR) are discarded, resulting in a transparent effect. The function of Transparent Mode is illustrated in the images below, in which the BS_COLOR is set to white.



Blue screen Mode

2D-ACCELERATOR also supports Blue Screen Mode, in which the pixels having the same color with blue screen color (BS_COLOR) are replaced by the back ground color (BG_COLOR).

2D-ACCELERATOR supports both host-to-screen mode and memory-to-screen mode of BLT.

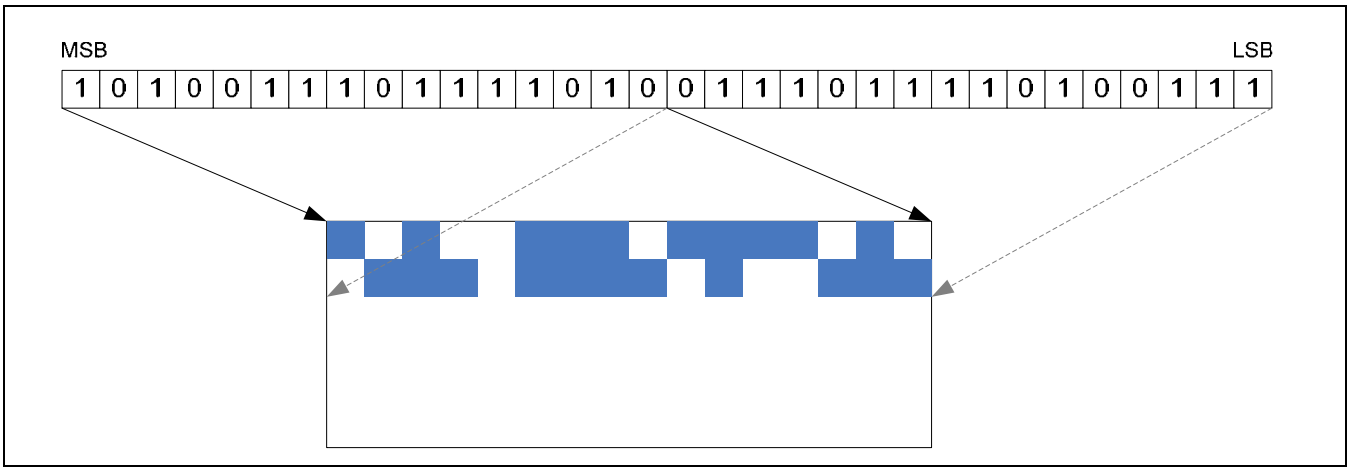
Related Registers

COORD_0	Coordinates of the leftmost topmost coordinate of the source image
COORD_1	Coordinates of the rightmost bottommost coordinate of the source image
COORD_2	Coordinates of the leftmost topmost coordinate of the destination image
COORD_3	Coordinates of the rightmost bottommost coordinate of the destination image
X-INCR	X increment value of the source image coordinates. If it is greater than 1, the image is shrunk horizontally; smaller than 1, stretched. This value is ignored if S bit in CMDR_1 is disabled or host-to-screen mode is used. X-INCR = (COORD1_X – COORD0_X) / (COORD3_X – COORD2_X) X-INCR is a (11.11) fixed point number as used in Line Drawing, but it has to be positive number in Bit Block Transfer.
Y-INCR	Y increment value of the source image coordinates. If it is greater than 1, the image is shrunk vertically; smaller than 1, stretched. This value is ignored when S bit in CMDR_1 is disabled or host-to-screen mode is used. Y_INCR = (COORD1_Y – COORD0_Y) / (COORD3_Y – COORD2_Y) Y-INCR is a (11.11) fixed point number as used in Line Drawing, but it has to be positive number in Bit Block Transfer.
SRC_BASE_ADDR	The base address of the source image (If memory-to-screen mode is used).

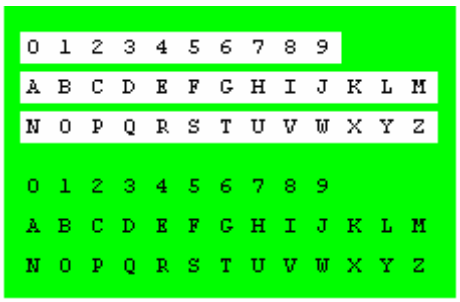
DEST_BASE_ADDR	The base address of the destination image (Usually the frame buffer base address)
SRC_HORI_RES_REG	The horizontal resolution of the source image
SRC_VERT_RES_REG	The vertical resolution of the source image (Used in YUV mode)
SC_HORI_RES_REG	The screen resolution
SRC_COLOR_MODE	The color mode of the source image
DEST_COLOR_MODE	The color mode of the destination image
BG_COLOR	Background color used in the Transparent Mode and Blue Screen Mode.
BS_COLOR	Blue screen color, used in the Blue Screen Mode.
ROP_REG	Enable/ disable Transparent Mode or Blue Screen Mode.
CMD1_REG	Writing to this register starts the rendering process of memory-to-screen Bit Block Transfer. If S bit is set, the image will be shrunk or stretched, depending on the values of X-INCR and Y-INCR.
CMD2_REG / CMD3_REG	The host provides the source image data through these two command registers. If the host writes the first 32-bit data into CMD2_REG, the rendering process starts in the host-to-screen mode. Then the host should provide the rest of data by writing into CMD3_REG continuously. The data written to CMD2_REG/CMD3_REG each time represents only one pixel, regardless of the source color format. If the source color format is 16-bpp (e.g., RGB565), the upper 16 bits of the data are ignored.

5.1.3 Color Expansion (Font Drawing)

Color Expansion expands the monochrome color to either background (BG_COLOR) or foreground (FG_COLOR) color. Each bit of the source data presents a pixel, with '1' indicating the foreground color and '0' the background color. The bit sequence is from MSB to LSB. The MSB of the first data corresponds to the leftmost topmost pixel of the destination image. The image below serves as a good illustration of the function and data type of Color Expansion. In this example, the foreground color is blue and background white, and the destination image is 16-pixel wide.



2D-ACCELERATOR renders Color Expansion image in Transparent Mode. In this mode, the pixels with background color (the corresponding bits are '0's) are discarded, resulting in a transparent effect. The transparent effect on Color Expansion is illustrated in the image below, in which the lower three lines are drawn with Transparent Mode enabled while the upper three disabled. Note that the background color is set to white and the foreground black.



2D-ACCELERATOR supports both host-to-screen mode and memory-to-screen mode of Color Expansion.

Related Registers

COORD_0	Coordinates of the leftmost topmost coordinate of the destination window
COORD_1	Coordinates of the rightmost bottommost coordinate of the destination window
FG_COLOR	Foreground Color
BG_COLOR	Background Color
ROP_REG	Enable/ disable Transparent Mode
CMD3_REG	The base address of the font data. Writing to this register starts the rendering process in the memory-to-screen mode.
CMD4_REG/CMD5_REG	The host provides the font data through these two command registers. If the host writes the first 32-bit data into CMD4_REG, the rendering process starts in the host-to-screen mode. Then the host should provide the rest of data by writing them into CMD5_REG continuously.

5.2 ROTATION

The pixels is rotated around the reference point (ox, oy) by 90/180/270 degree clockwise or perform an X-axis/Y-axis flip around the horizontal or vertical line on which (ox, oy) lies. The effects of all rotation options are summarized in the following table and illustrated in Figure 9.5-2.

Related Registers

ROT_OC_REG	Coordinates of the rotation reference point
ROTATE_REG	Rotation mode configuration

Rotation Effect

	0°	90°	180°	270°	X-flip	Y-flip
x	dcx	-dcy + (ox+oy)	-dcx + 2ox	dcy + (ox-oy)	dcx	-dcx + 2ox
y	dcy	dcx - (ox-oy)	-dcy + 2oy	-dcx + (ox+oy)	-dcy + 2oy	dcy

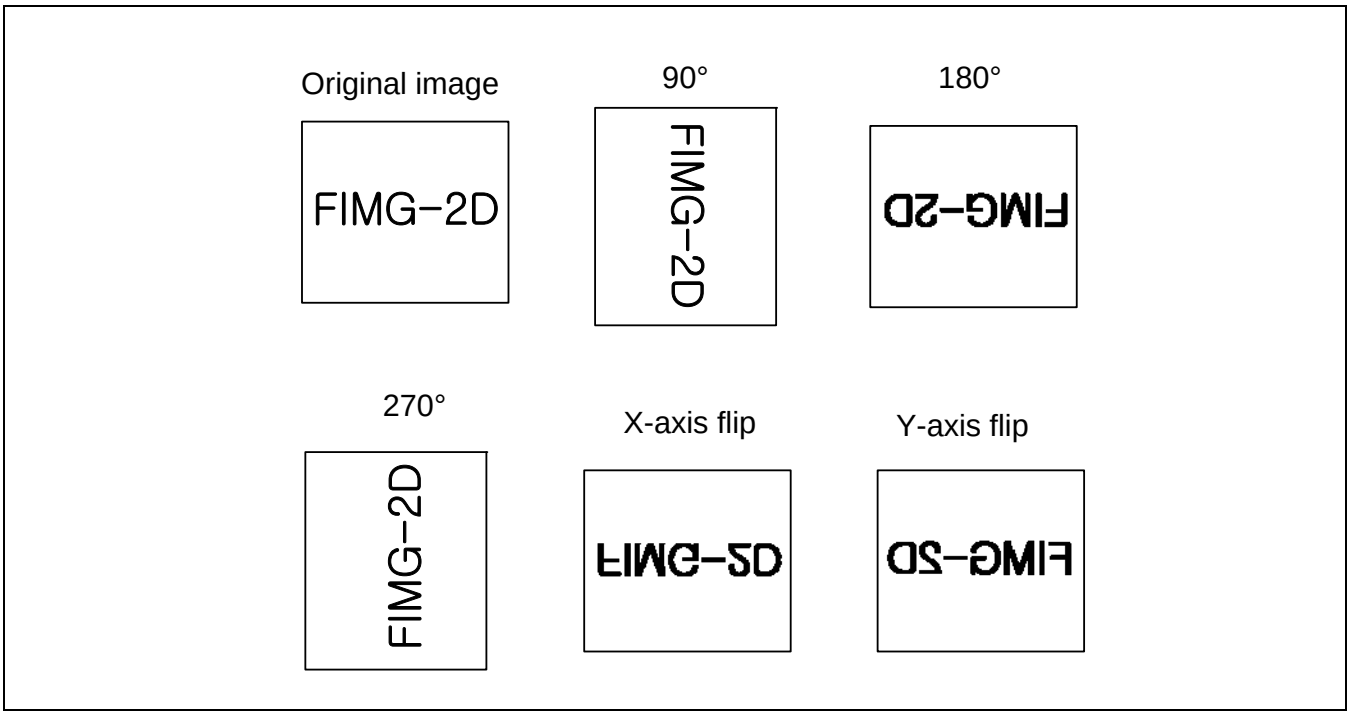


Figure 9.5-2 Rotation Example

5.3 CLIPPING

Clipping discards the pixels (after rotation) outside the clipping window. The discarded pixels will not go through the rest of rendering pipelines.

NOTE: The clipping windows must reside totally inside the screen. Setting the clipping window the same size with the screen disables the clipping effect, and a clipping window bigger than the screen size is not allowed.

Related Registers

CW_LT_REG	Coordinates of the leftmost topmost point of the clipping window
CW_RB_REG	Coordinates of the rightmost bottommost point of the clipping window

5.4 STENCIL TEST

The Stencil Test conditionally discards a pixel based on the outcome of a comparison between the color value of this pixel of the source image and the DR (min)/DR(max) values. If each field (R, G, B, A) of the color value falls in the range of [DR (min), DR (max)], this pixel is passed to the next stage; otherwise, discarded. User can disable the stencil test on a specific field by clearing the corresponding bits in COLORKEY_CNTL. Note that each field of DR_MIN and DR_MAX is 8-bit wide, regardless of the source color mode setting.

Related Registers

COLORKEY_CNTL	Stencil Test configurations, such as enable/ disable the test etc.
COLORKEY_DR_MIN	Set the DR(Minimum) value for each field
COLORKEY_DR_MAX	Set the DR(Maximum) value for each field

5.5 RASTER OPERATION

Raster Operation performs Boolean operations on three operands: source, destination and the third operand according to the 8-bit ROP value specified by the user. The truth table of ROP is given in the following table.

Source	Destination	Third Operand	ROP Value
0	0	0	Bit0
0	0	1	Bit1
0	1	0	Bit2
0	1	1	Bit3
1	0	0	Bit4
1	0	1	Bit5
1	1	0	Bit6
1	1	1	Bit7

The third operand can be pattern or foreground color, configurable by the OS bit in the ROP_REG.

Pattern is a user-specified 8x8x16-bpp image; the pattern data must be in RGB565 format. The following equation is used to calculate the pattern index of pixel (x, y):

```
index = ( ((patternOffsetY + y) & 0x7 )<<3 ) + ((patternOffsetX + x)&0x7),
```

where patternOffsetY and patternOffsetX are the offset value specified in register PATOFF_REG.

Here are some examples on how to use the ROP value to perform the operations:

1) Final Data = Source. Only the Source data matter, so ROP Value = "11110000".
2) Final Data = Destination. Only the Destination data matter, so ROP Value = "11001100".
3) Final Data = Pattern. Only the Pattern data matter, so ROP Value = "10101010".
4) Final Data = Source AND Destination. ROP Value = "11110000" & "11001100" = "11000000"
5) Final Data = Source OR Pattern. ROP Value = "11110000" | "10101010" = "11111010".

NOTE: The Raster Operation only applies on R, G, B fields of the color data; the A field will not be affected.

Related Registers

PATTERN_REG[0:31]	Pattern data
PATOFF_REG	Pattern offset X, Y
ROP_REG	ROP configurations and ROP Value

5.6 ALPHA BLENDING

Alpha Blending combines the source color and the destination color in the frame buffer to get the new destination color.

The conventional alpha blending equation is: final data = src * alpha + dest * (1.0 – alpha). 2D-ACCELERATOR uses 8-bit integer to represent the alpha value, with 0 indicating 1/256 and 255 indicating 1.0. The equation of converting 8-bit ALPHA value to the actual fractional alpha value is: alpha = (ALPHA+1) / 256.

The internal computation of alpha blending and fading is as follows:

User-specified alpha value: ALPHA (given by ALPHA_REG, from 0 to 255)

```
[Alpha Blending]

data = ( source * (ALPHA+1) + destination * (255-ALPHA) ) >> 8

[Fading]

data = (( source * (ALPHA+1) ) >> 8) + fading offset
```

```
Per-pixel alpha blending: ALPHA (given by the source image, from 0 to 255)

[Alpha Blending]

data = ( source * (ALPHA+1) + destination * (255-ALPHA) ) >> 8

[Fading]

data = ((source * (ALPHA+1) ) >> 8 ) + fading offset
```

Related Registers

ROP_REG	Alpha blending configurations: alpha blending disable/ enable, per-pixel alpha blending disable/ enable, fading disable/ enable.
ALPHA_REG	Alpha value and fading value.

6 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
General Registers				
CONTROL_REG	0xEE80_0000	W	Control Register.	0x0000_0000
INTEN_REG	0xEE80_0004	R/W	Interrupt Enable Register.	0x0000_0000
FIFO_INTC_REG	0xEE80_0008	R/W	Interrupt Control Register.	0x0000_0018
INTC_PEND_REG	0xEE80_000C	R/W	Interrupt Control Pending Register.	0x0000_0000
FIFO_STAT_REG	0xEE80_0010	R	Command FIFO Status Register.	0x0000_0600
Command Registers				
CMD0_REG	0xEE80_0100	W	Command Register for Line/Point Drawing.	-
CMD1_REG	0xEE80_0104	W	Command Register for BitBLT.	-
CMD2_REG	0xEE80_0108	W	Command Register for Host to Screen Bitblt Transfer Start	-
CMD3_REG	0xEE80_010C	W	Command Register for Host to Screen Bitblt Transfer Continue	-
CMD4_REG	0xEE80_0110	W	Command Register for Color Expansion. (Host to Screen, Font Start)	-
CMD5_REG	0xEE80_0114	W	Command Register for Color Expansion (Host to Screen, Font Continue)	-
CMD6_REG	0xEE80_0118	W	Reserved	-
CMD7_REG	0xEE80_011C	W	Command Register for Color Expansion (Memory to Screen)	-
Parameter Setting Registers				
Resolution				
SRC_ RES_REG	0xEE80_0200	R/W	Source Image Resolution	0x0000_0000
SRC_HORI_RES_REG	0xEE80_0204	R/W	Source Image Horizontal Resolution	0x0000_0000
SRC_VERT_RES_REG	0xEE80_0208	R/W	Source Image Vertical Resolution	0x0000_0000
SC_RES_REG	0xEE80_0210	R/W	Screen Resolution	0x0000_0000
SC_HORI_RES _REG	0xEE80_0214	R/W	Screen Horizontal Resolution	0x0000_0000
SC_VERT_RES _REG	0xEE80_0218	R/W	Screen Vertical Resolution	0x0000_0000
Clipping Window				
CW_LT_REG	0xEE80_0220	R/W	Left Top Coordinates of Clip Window	0x0000_0000
CW_LT_X_REG	0xEE80_0224	R/W	Left X Coordinate of Clip Window	0x0000_0000
CW_LT_Y_REG	0xEE80_0228	R/W	Top Y Coordinate of Clip Window	0x0000_0000
CW_RB_REG	0xEE80_0230	R/W	Right Bottom Coordinate of Clip Window	0x0000_0000
CW_RB_X_REG	0xEE80_0234	R/W	Right X Coordinate of Clip Window	0x0000_0000
CW_RB_Y_REG	0xEE80_0238	R/W	Bottom Y Coordinate of Clip Window	0x0000_0000

Coordinates				
COORD0_REG	0xEE80_0300	R/W	Coordinates 0 Register	0x0000_0000
COORD0_X_REG	0xEE80_0304	R/W	X Coordinate of Coordinates 0.	0x0000_0000
COORD0_Y_REG	0xEE80_0308	R/W	Y Coordinate of Coordinates 0	0x0000_0000
COORD1_REG	0xEE80_0310	R/W	Coordinates 1 Register	0x0000_0000
COORD1_X_REG	0xEE80_0314	R/W	X Coordinate of Coordinates 1	0x0000_0000
COORD1_Y_REG	0xEE80_0318	R/W	Y Coordinate of Coordinates 1	0x0000_0000
COORD2_REG	0xEE80_0320	R/W	Coordinates 2 Register	0x0000_0000
COORD2_X_REG	0xEE80_0324	R/W	X Coordinate of Coordinates 2.	0x0000_0000
COORD2_Y_REG	0xEE80_0328	R/W	Y Coordinate of Coordinates 2	0x0000_0000
COORD3_REG	0xEE80_0330	R/W	Coordinates 3 Register	0x0000_0000
COORD3_X_REG	0xEE80_0334	R/W	X Coordinate of Coordinates 3	0x0000_0000
COORD3_Y_REG	0xEE80_0338	R/W	Y Coordinate of Coordinates 3	0x0000_0000
Rotation				
ROT_OC_REG	0xEE80_0340	R/W	Rotation Origin Coordinates.	0x0000_0000
ROT_OC_X_REG	0xEE80_0344	R/W	X Coordinate of Rotation Origin Coordinates.	0x0000_0000
ROT_OC_Y_REG	0xEE80_0348	R/W	Y Coordinate of Rotation Origin Coordinates.	0x0000_0000
ROTATE_REG	0xEE80_034C	R/W	Rotation Mode Register.	0x0000_0001
Data Format				
ENDIAN	0xEE80_0350	R/W	Big & little ENDIAN Select	0x0000_0000
X,Y Increment Setting				
X_INCR_REG	0xEE80_0400	R/W	X Increment Register	0x0000_0000
Y_INCR_REG	0xEE80_0404	R/W	Y Increment Register	0x0000_0000
ROP & Alpha Setting				
ROP_REG	0xEE80_0410	R/W	Raster Operation Register	0x0000_0000
ALPHA_REG	0xEE80_0420	R/W	Alpha Value, Fading Offset.	0x0000_0000
Color				
FG_COLOR_REG	0xEE80_0500	R/W	Foreground Color / Alpha Register.	0x0000_0000
BG_COLOR_REG	0xEE80_0504	R/W	Background Color Register	0x0000_0000
BS_COLOR_REG	0xEE80_0508	R/W	Blue Screen Color Register	0x0000_0000
SRC_COLOR_MODE_REG	0xEE80_0510	R/W	Src Image Color Mode Register.	0x0000_0000
DEST_COLOR_MODE_REG	0xEE80_0514	R/W	Dest Image Color Mode Register	0x0000_0000
Pattern				
PATTERN_REG[0:31]	0xEE80_0600 0xEE80_067C	R/W	Pattern Memory.	0x0000_0000
PATOFF_REG	0xEE80_0700	R/W	Pattern Offset XY Register	0x0000_0000
PATOFF_X_REG	0xEE80_0704	R/W	Pattern Offset X Register	0x0000_0000
PATOFF_Y_REG	0xEE80_0708	R/W	Pattern Offset Y Register	0x0000_0000

Stencil Test				
STENCIL_CNTL_REG	0xEE80_0720	R/W	Stencil Control Register	0x0000_0000
STENCIL_DR_MIN_REG	0xEE80_0724	R/W	Stencil Decision Reference MIN Register	0x0000_0000
STENCIL_DR_MAX_REG	0xEE80_0728	R/W	Stencil Decision Reference MAX Register	0xFFFF_FFFF
Image Base Address				
SRC_BASE_ADDR_REG	0xEE80_0730	R/W	Source Image Base Address Register	0x0000_0000
DEST_BASE_ADDR_REG	0xEE80_0734	R/W	Dest Image Base Address Register (in most cases, frame buffer address)	0x0000_0000

6.1 GENERAL REGISTERS

6.1.1 Control Register (CONTROL_REG, W, Address = 0xEE80_ 0000)

CONTROL_REG	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x0
R	[0]	Software Reset Write to this bit results in a one-cycle reset signal to FIMG2D graphics engine. Every command register and parameter setting register will be assigned the "Reset Value", and the command FIFO will be cleared.	0x0

6.1.2 Interrupt Enable Register (INTEN_REG, R/W, Address = 0xEE80_ 0004)

INTEN_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
CCF	[10]	Current Command Finished interrupt enable. If this bit is set, when the graphics engine finishes the execution of current command, an interrupt occurs, and the INTP_CMD_FIN flag in INTC_PEND_REG will be set.	0x0
ACF	[9]	All Commands Finished interrupt enable. If this bit is set, when the graphics engine finishes the execution of all commands in the command FIFO, an interrupt occurs, and the INTP_ALL_FIN flag in INTC_PEND_REG will be set.	0x0
FIFO_FULL	[8]	Command FIFO Full interrupt enable. If this bit is set, when command FIFO is full (32 entries), an interrupt occurs, and the INTP_FULL flag in the interrupt pending register (INTC_PEND_REG) will be set.	0x0
Reserved	[7:1]	Reserved	0x0
FIFO_INT_E	[0]	If this bit is set, when the number of entries occupied in command FIFO is greater or equal to FIFO_INT_LEVEL (in FIFO_INTC_REG), an interrupt occurs, and the INTP_FIFO_LEVEL flag in the interrupt pending register (INTC_PEND_REG) will be set.	0x0

6.1.3 FIFO Interrupt Control Register (FIFO_INTC_REG, R/W, Address = 0xEE80_ 0008)

FIFO_INTC_REG	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0x0
FIFO_INT_LEVEL	[5:0]	If FIFO_INT_E (in INTEN_REG) is set, when FIFO_USED (in FIFO_STAT_REG) is greater or equal to FIFO_INT_LEVEL, an interrupt occurs.	0x18

6.1.4 Interrupt Pending Register (INTC_PEND_REG, R/W, Address = 0xEE80_ 000C)

INTC_PEND_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
INTP_CMD_FIN	[10]	Current Command Finished interrupt flag. Writing '1' to this bit clears this flag.	0x0
INTP_ALL_FIN	[9]	All Commands Finished interrupt flag. Writing '1' to this bit clears this flag.	0x0
INTP_FULL	[8]	Command FIFO Full interrupt flag. Writing '1' to this bit clears this flag.	0x0
Reserved	[7:1]	Reserved	0x0
INTP_FIFO_LEVEL	[0]	FIFO_USED reaches FIFO_INT_LEVEL interrupt flag. Writing '1' to this bit clears this flag.	0x0

6.1.5 FIFO status Register (FIFO_STAT_REG, R/W, Address = 0xEE80_ 0010)

FIFO_STAT_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
CMD_FIN	[10]	1= The graphics engine completes the execution of current command. 0 = In middle of rendering process.	0x1
ALL_FIN	[9]	1 = Graphics engine is in idle state. The graphics engine completes the execution of all commands in the command FIFO. Note that ALL_FIN = CMD_FIN && (FIFO_USED==0). 0 = In the middle of rendering process, or FIFO_USED is greater than 0.	0x1
FIFO_OVERFLOW	[8]	1= Command FIFO is full, no more commands can be handled 0 = Command FIFO is not full.	0x0
Reserved	[7]	Reserved	0x0
FIFO_USED	[6:1]	The number of entries occupied in command FIFO.	0x0
FIFO_LEVEL_INT	[0]	1 = FIFO_USED is greater or equal to FIFO_INT_LEVEL 0 = FIFO_USED is smaller than FIFO_INT_LEVEL	0x0

6.2 COMMAND REGISTERS

6.2.1 Line Drawing Register (CMD0_REG, W, Address = 0xEE80_ 0100)

CMD0_REG	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	-
D	[9]	0 = Draw Last Point 1 = Do-not-Draw Last Point.	-
M	[8]	0 = Major axis is Y. 1 = Major axis is X.	-
Reserved	[7:2]	Reserved	-
L	[1]	0 = Nothing. 1 = Line Drawing.	-
P	[0]	0 = Nothing. 1 = Point Drawing.	-

6.2.2 BitBLT Register (CMD1_REG, W, Address = 0xEE80_0104)

CMD1_REG	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	-
S	[1]	0 = Nothing 1 = Stretch BitBLT	-
N	[0]	0 = Nothing 1 = Normal BitBLT	-

6.2.3 Host to Screen Start BitBLT Register (CMD2_REG, W, Address = 0xEE80_ 0108)

CMD2_REG	Bit	Description	Reset Value
Data	[31:0]	BitBLT data (Start) **NOTE:** The data written to this register represents only one pixel, regardless of the source color mode. If the source color mode is 16-bpp (e.g., RGB565), the upper 16 bits of the data are ignored.	-

6.2.4 Host to Screen Continue BitBLT Register (CMD3_REG, W, Address = 0xEE80_ 010C)

CMD3_REG	Bit	Description	Reset Value
Data	[31:0]	BitBLT data (Continue) **NOTE:** The data written to this register represents only one pixel, regardless of the source color mode. If the source color mode is 16-bpp (e.g., RGB565), the upper 16 bits of the data are ignored.	-

6.2.5 Host to Screen Start Color Expansion Register (CMD4_REG, W, Address = 0xEE80_ 0110)

CMD4_REG	Bit	Description	Reset Value
Data	[31:0]	Color Expansion Data (Start)	-

6.2.6 Host to Screen Continue Color Expansion Register (CMD5_REG, W, Address = 0xEE80_ 0114)

CMD5_REG	Bit	Description	Reset Value
Data	[31:0]	Color Expansion Data (Continue)	-

6.2.7 Memory to Screen Color Expansion Register (CMD7_REG, W, Address = 0xEE80_ 011C)

CMD7_REG	Bit	Description	Reset Value
Memory Address	[31:0]	Bitmap data base address (Used in memory-to-screen mode, must be word-aligned).	-

6.3 PARAMETER SETTING REGISTERS

6.3.1 Source Image Resolution (SRC_RES_REG, R/W, Address = 0xEE80_ 0200)

SRC_RES_REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
VertRes	[26:16]	Vertical resolution of source image Range: 1 ~ 2040	0x0
Reserved	[15:11]	Reserved	0x0
HoriRes	[10:0]	Horizontal resolution of source image Range: 1 ~ 2040. **NOTE:** In YUV mode, HoriRes must be an even number.	0x0

6.3.2 Source Image Horizontal Resolution (SRC_HORI_RES_REG, R/W, Address = 0xEE80_ 0204)

SRC_HORI_RES_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
HoriRes	[10:0]	Horizontal resolution of source image. Range: 1 ~ 2040. **NOTE:** In YUV mode, HoriRes must be an even number.	0x0

6.3.3 Source Image Vertical Resolution (SRC_VERT_RES_REG, R/W, Address = 0xEE80_ 0208)

SRC_VERT_RES_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
VertRes	[10:0]	Vertical resolution of source image. Range: 1 ~ 2040	0x0

6.3.4 Screen Resolution (SC_RES_REG, R/W, Address = 0xEE80_ 0210)

SC_RES_REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
VertRes	[26:16]	Vertical resolution of the screen. Range: 1 ~ 2040	0x0
Reserved	[15:11]	Reserved	0x0
HoriRes	[10:0]	Horizontal resolution of the screen. Range: 1 ~ 2040	0x0

6.3.5 Screen Horizontal Resolution (SC_HORI_RES_REG, R/W, Address = 0xEE80_ 0214)

SC_HORI_RES_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
HoriRes	[10:0]	Horizontal resolution of the screen. Range: 1 ~ 2040	0x0

6.3.6 Screen Vertical Resolution (SC_VERT_RES_REG, R/W, Address = 0xEE80_ 0218)

SC_VERT_RES_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
VeriRes	[10:0]	Vertical resolution of the screen. Range: 1 ~ 2040	0x0

6.3.7 LeftTop Clipping Window (CW_LT_REG, R/W, Address = 0xEE80_ 0220)

CW_LT_REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
TopCW_Y	[26:16]	Top Y Clipping Window Requirement: TopCW_Y < BottomCW_Y	0x0
Reserved	[15:11]	Reserved	0x0
LeftCW_X	[10:0]	Left X Coordinate of Clipping Window. Requirement: LeftCW_X < RightCW_X	0x0

6.3.8 Left X Clipping Window (CW_LT_X_REG, R/W, Address = 0xEE80_ 0224)

CW_LT_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
LeftCW_X	[10:0]	Left X Clipping Window Requirement: LeftCW_X < RightCW_X	0x0

6.3.9 Top Y Clipping Window (CW_LT_Y_REG, R/W, Address = 0xEE80_ 0228)

CW_LT_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
TopCW_Y	[10:0]	Top Y Clipping Window Requirement: TopCW_Y < BottomCW_Y	0x0

6.3.10 RightBottom Clipping Window (CW_RB_REG, R/W, Address = 0xEE80_ 0230)

CW_RB_REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
BottomCW_Y	[26:16]	Bottom Y Clipping Window	0x0

		Requirement: BottomCW_Y < VeriRes (SC_VERI_RES_REG)	
Reserved	[15:11]	Reserved	0x0
RightCW_X	[10:0]	Right X Clipping Window Requirement: RightCW_X < HoriRes (SC_HORI_RES_REG)	0x0

6.3.11 Right X Clipping Window (CW_RB_X_REG, R/W, Address = 0xEE80_ 0234)

CW_RB_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
RightCW_X	[10:0]	Right X Clipping Window Requirement: RightCW_X < HoriRes (SC_HORI_RES_REG)	0x0

6.3.12 Bottom Y Clipping Window (CW_RB_Y_REG, R/W, Address = 0xEE80_ 0238)

CW_RB_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
BottomCW_Y	[10:0]	Bottom Y Clipping Window Requirement: BottomCW_Y < VeriRes (SC_VERI_RES_REG)	0x0

6.3.13 Coordinate_0 Register (COORD0_REG, R/W, Address = 0xEE80_ 0300)

COORD0_REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
Y	[26:16]	Coordinate_0 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	Reserved	0x0
X	[10:0]	Coordinate_0 X Range: 0 ~ 2039	0x0

6.3.14 Coordinate_0 X Register (COORD0_X_REG, R/W, Address = 0xEE80_ 0304)

COORD0_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD0_X	[10:0]	Coordinate_0 X Range: 0 ~ 2039	0x0

6.3.15 Coordinate_0 Y Register (COORD0_Y_REG, R/W, Address = 0xEE80_ 0308)

COORD0_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD0_Y	[10:0]	Coordinate_0 Y Range: 0 ~ 2039	0x0

6.3.16 Coordinate_1 Register (COORD1_ REG, R/W, Address = 0xEE80_ 0310)

COORD1_ REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
Y	[26:16]	Coordinate_1 Y	0x0

		Range: 0 ~ 2039	
Reserved	[15:11]	Reserved	0x0
X	[10:0]	Coordinate_1 X Range: 0 ~ 2039	0x0

6.3.17 Coordinate_1 X Register (COORD1_X_REG, R/W, Address = 0xEE80_ 0314)

COORD1_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD1_X	[10:0]	Coordinate_1 X Range: 0 ~ 2039	0x0

6.3.18 Coordinate_1 Y Register (COORD1_Y_REG, R/W, Address = 0xEE80_ 0318)

COORD1_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD1_Y	[10:0]	Coordinate_1 Y Range: 0 ~ 2039	0x0

6.3.19 Coordinate_2 Register (COORD2_ REG, R/W, Address = 0xEE80_ 0320)

COORD2_ REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
Y	[26:16]	Coordinate_2 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	Reserved	0x0
X	[10:0]	Coordinate_2 X Range: 0 ~ 2039	0x0

6.3.20 Coordinate_2 X Register (COORD2_X_REG, R/W, Address = 0xEE80_ 0324)

COORD2_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD2_X	[10:0]	Coordinate_2 X Range: 0 ~ 2039	0x0

6.3.21 Coordinate_2 Y Register (COORD2_Y_REG, R/W, Address = 0xEE80_ 0328)

COORD2_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD2_Y	[10:0]	Coordinate_2 Y Range: 0 ~ 2039	0x0

6.3.22 Common Resource Coordinate_3 Register (COORD3_ REG, R/W, Address = 0xEE80_ 0330)

COORD3_ REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
Y	[26:16]	Coordinate_3 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]	Reserved	0x0
X	[10:0]	Coordinate_3 X Range: 0 ~ 2039	0x0

6.3.23 Coordinate_3 X Register (COORD3_X_REG, R/W, Address = 0xEE80_ 0334)

COORD3_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD3_X	[10:0]	Coordinate_3 X Range: 0 ~ 2039	0x0

6.3.24 Coordinate_3 Y Register (COORD3_Y_REG, R/W, Address = 0xEE80_ 0338)

COORD3_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
COORD3_Y	[10:0]	Coordinate_3 Y Range: 0 ~ 2039	0x0

6.3.25 Rotation Origin Coordinate (ROT_OC_REG, R/W, Address = 0xEE80_ 0340)

ROT_OC_REG	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0x0
Y	[26:16]	X coordinate of the reference point of rotation Range: 0 ~ 2039	0x0
Reserved	[15:11]	Reserved	0x0
X	[10:0]	Y coordinate of the reference point of rotation Range 0 ~ 2039	0x0

6.3.26 Rotation Origin Coordinate X (ROT_OC_X_REG, R/W, Address = 0xEE80_ 0344)

ROT_OC_X_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
ROT_OC_X	[10:0]	X coordinate of the reference point of rotation Range: 0 ~ 2039	0x0

6.3.27 Rotation Origin Coordinate Y (ROT_OC_Y_REG, R/W, Address = 0xEE80_ 0348)

ROT_OC_Y_REG	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0x0
ROT_OC_Y	[10:0]	Y coordinate of the reference point of rotation Range 0 ~ 2039	0x0

6.3.28 Rotation Register (ROTATE_REG, R/W, Address = 0xEE80_ 034C)

ROTATE_REG	Bit	Description	Reset Value
Reserved	[31:6]	Reserved	0x0
FY	[5]	Y-flip	0x0
FX	[4]	X-flip	0x0
R3	[3]	270° Rotation	0x0
R2	[2]	180° Rotation	0x0
R1	[1]	90° Rotation	0x0
R0	[0]	0° Rotation	0x1

** If the two or more of Rn are set to 1 at the same time, drawing engine operates unpredictably.*

6.3.29 Endian (ENDIAN_REG, R/W, Address = 0xEE80_ 0350)

ENDIAN_REG	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0x0
DEST_ENDIAN	[1]	Big Endian setting for dest data (1: Big Endian; 0: Little Endian)	0x0
SRC_ENDIAN	[0]	Big Endian setting for source data(1: Big Endian; 0: Little Endian)	0x0

6.3.30 X Increment Register (X_INCR_REG, R/W, Address = 0xEE80_ 0400)

Field	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0x0
X_INCR	[21:0]	X increment value (2's complement, 11-digit fraction)	0x0

6.3.31 Y Increment Register (Y_INCR_REG, R/W, Address = 0xEE80_ 0404)

Field	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0x0
Y_INCR	[21:0]	Y increment value (2's complement, 11-digit fraction)	0x0

6.3.32 Raster Operation Register (ROP_REG, R/W, Address = 0xEE80_ 0410)

ROP_REG	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0x0
OS	[13]	Third Operand Select : 1'b0 = Pattern 1'b1 = Foreground Color	0x0
ABM	[12:10]	Alpha Mode : 3'b000 = No Alpha Blending 3'b001 = Perpixel Alpha Blending with Source Bitmap 3'b010 = Alpha Blending with Alpha Register 3'b100 = Fading Others = Reserved **NOTE:** Perpixel Alpha Blending is applied in bit block transfer.	0x0
T	[9]	0 = Opaque Mode 1 = Transparent Mode	0x0
B	[8]	0 = Blue-screen Mode Disable 1 = Blue-screen Mode Enable **NOTE:** T and B must not be set at the same time.	0x0
ROP Value	[7:0]	Raster Operation Value	0x0

6.3.33 Alpha Register (ALPHA_REG, R/W, Address = 0xEE80_ 0420)

ALPHA_REG	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0x0
Fading	[15:8]	Fading Offset Value	0x0
Alpha	[7:0]	Alpha Value	0x0

6.3.34 Foreground Color Register (FG_COLOR_REG, R/W, Address = 0xEE80_0500)

FG_COLOR_REG	Bit	Description	Reset Value
ForegroundColor	[31:0]	Foreground Color Value. The alpha field of the foreground color is discarded.	0x0

6.3.35 Background Color Register (BG_COLOR_REG, R/W, Address = 0xEE80_ 0504)

BG_COLOR_REG	Bit	Description	Reset Value
BackgroundColor	[31:0]	Background Color Value. The alpha field of the background color is discarded.	0x0

6.3.36 BlueScreen Color Register (BS_COLOR_REG, R/W, Address = 0xEE80_ 0508)

BS_COLOR_REG	Bit	Description	Reset Value

BlueScreenColor	[31:0]	BlueScreen Color Value. The alpha field of the blue screen color is discarded.	0x0

6.3.37 Source Image Color Mode (SRC_COLOR_MODE_REG, R/W, Address = 0xEE80_ 0510)

SRC_COLOR_MODE_REG	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0x0
Narrow	[4]	1 = YUV narrow range (Y:16-235, UV: 16-240) 0 = YUV wide range (YUV: 0-255)	0x0
YUV	[3]	1 = YUV mode 0 = RGB mode This bit should be set to 0 in point/ line drawing mode and color expansion mode.	0x0
Color Setting	[2:0]	3'b000: RGB_565 3'b001: RGBA_5551 3'b010: ARGB_1555 3'b011: RGBA_8888 3'b100: ARGB_8888 3'b101: XRGB_8888 3'b110: RGBX_8888 The Color Setting is ignored if YUV mode is selected	0x0

6.3.38 Destination Image Color Mode (DEST_COLOR_MODE_REG, R/W, Address = 0xEE80_ 0514)

DEST_COLOR_MODE_REG	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0x0
Color Setting	[2:0]	3'b000: RGB_565 3'b001: RGBA_5551 3'b010: ARGB_1555 3'b011: RGBA_8888 3'b100: ARGB_8888 3'b101: XRGB_8888 3'b110: RGBX_8888	0x0

6.3.39 Pattern Offset Register (PATOFF_REG, R/W, Address = 0xEE80_ 0700)

PATOFF_REG	Bit	Description	Reset Value
Reserved	[31:19]	Reserved	0x0
POffsetY	[18:16]	Pattern Offset Y Value	0x0
Reserved	[15:3]	Reserved	0x0
POffsetX	[2:0]	Pattern OffsetX Value	0x0

6.3.40 Pattern Offset X Register (PATOFF_X_REG, R/W, Address = 0xEE80_ 0704)

PATOFF_X_REG	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0x0
POffsetX	[2:0]	Pattern OffsetX Value	0x0

6.3.41 Pattern Offset Y Register (PATOFF_Y_REG, R/W, Address = 0xEE80_ 0708)

PATOFF_Y_REG	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0x0
POffsetY	[2:0]	Pattern OffsetY Value	0x0

6.3.42 Colorkey Control Register (COLORKEY_CNTL_REG, R/W, Address = 0xEE80_ 0720)

COLORKEY_CNTL_REG	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0x0
StencilInverse	[4]	0 = Normal stencil test 1 = Inversed stencil test This bit should be set to 0 if the stencil test of every color field is disabled.	0x0
StencilOnR	[3]	0 = Stencil Test Off for R value 1 = Stencil Test On for R value	0x0

StencilOnG	[2]	0 = Stencil Test Off for G value 1 = Stencil Test On for G value	0x0
StencilOnB	[1]	0 = Stencil Test Off for B value 1= Stencil Test On for B value	0x0
StencilOnA	[0]	0 = Stencil Test Off for A value 1 = Stencil Test On for A value	0x0

6.3.43 Colorkey Decision Reference Minimum Register (COLORKEY _DR_MIN_REG, R/W, Address = 0xEE80_ 0724)

COLORKEY _DR_MIN_REG	Bit	Description	Reset Value
A_DR(min)	[31:24]	Alpha DR MIN value	0x0
R_DR(min)	[23:16]	RED DR MIN value	0x0
G_DR(min)	[15:8]	GREEN DR MIN value	0x0
B_DR(min)	[7:0]	BLUE DR MIN value	0x0

6.3.44 Colorkey Decision Reference Maximum Register (COLORKEY _DR_MAX_REG, R/W, Address = 0xEE80_ 0728)

COLORKEY _DR_MAX_REG	Bit	Description	Reset Value
A_DR(max)	[31:24]	Alpha DR MAX value	0xFF
R_DR(max)	[23:16]	RED DR MAX value	0xFF
G_DR(max)	[15:8]	GREEN DR MAX value	0xFF
B_DR(max)	[7:0]	BLUE DR MAX value	0xFF

6.3.45 Source Image Base Address Register (SRC_BASE_ADDR_REG, R/W, Address = 0xEE80_ 0730)

SRC_BASE_ADDR_REG	Bit	Description	Reset Value
ADDR	[31:0]	Base address of the source image	0x0

6.3.46 Destination Image Base Address Register (DEST_BASE_ADDR_REG, R/W, Address = 0xEE80_ 0734)

DEST_BASE_ADDR_REG	Bit	Description	Reset Value
ADDR	[31:0]	Base address of the destination image (in most cases, it is also the frame buffer base address).	0x0

9.6 3D-ACCELERATOR

1 INTRODUCTION

1.1 FEATURES

- 7.8M vertices/s @133MHz (single light)
- 167M pixels/s fill-rates @133MHz (shaded pixels)
- Programmable Shader Model 3.0 support
- 128-bit (32-bit x 4) Floating-point Vertex Shader
- Geometry-texture cache support
- 128-bit (32-bit x 4) Floating-point two Fragment Shaders
- Max. 4K x 4K frame-buffer (16/32-bpp)
- 32-bit depth buffer (8-bit stencil/24-bit Z)
- Texture format: 1/2/4/8/16/32-bpp RGB, YUV 422, S3TC Compressed
- Support max. 8 user-defined textures
- API Support: OpenGL ES 1.1 & 2.0, D3D Mobile
- Vertex Buffer & Vertex Cache
- H/W Clipping (Near & Far)
- Primitive assembly & hard-wired triangle setup engine
- Two pixels/cycle hard-wired rasterizer
- Two texturing engine
  - ♦ Nearest/bilinear/trilinear filtering
  - ♦ 8-layered multi-texturing support
  - ♦ One bilinear-filtered texel/cycle each
- Fragment processing: Alpha/Stencil/Z/Dither/Mask/ROP
- Hierarchical caching: L1/L2 Texture-caches, Z/Color caches

1.2 OVERALL ARCHITECTURE

Overall graphics pipeline consists of one vertex pipeline and two pixel pipelines.

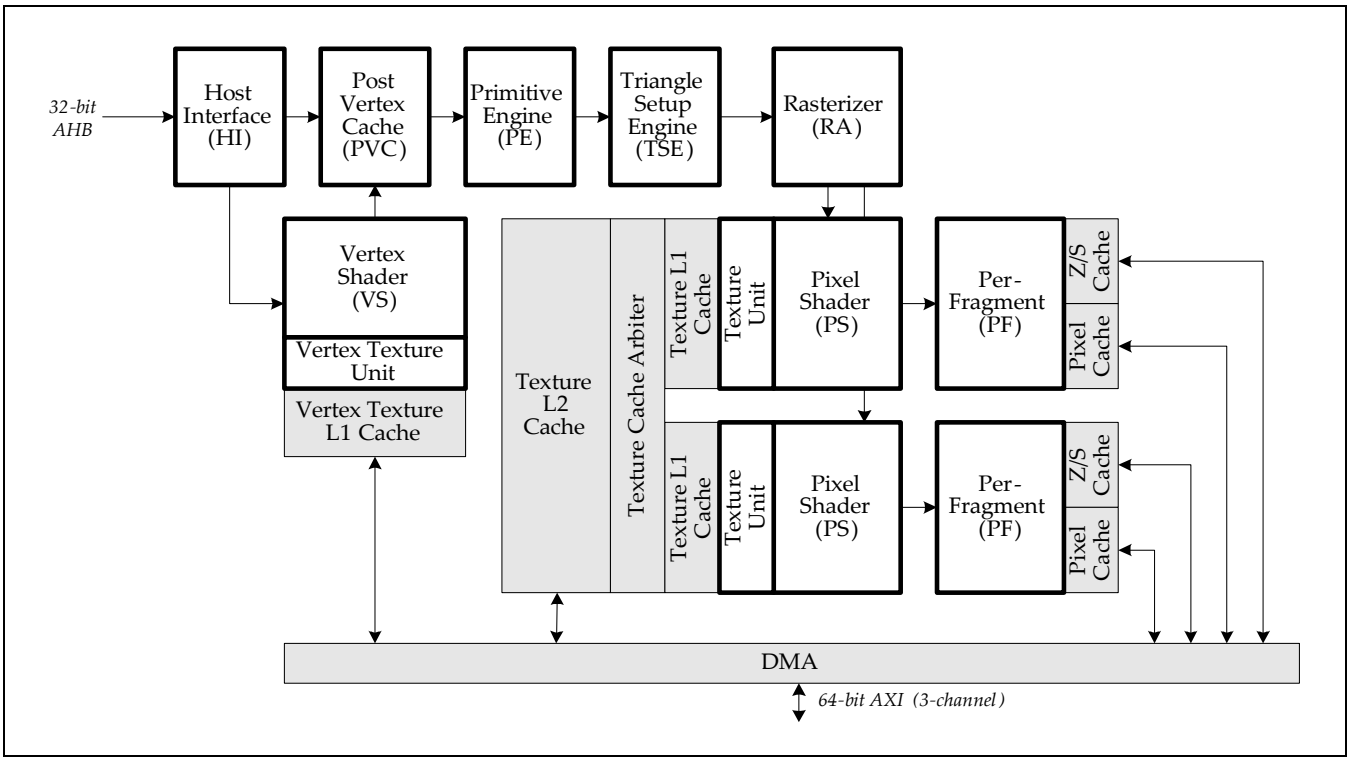


Figure 9.6- 1 Overall Block Diagram.

1.3 SPECIAL FUNCTION REGISTER SUMMARY (GLOBAL)

Register	Address	R/W	Description	Reset Value
GB_PIPESTATE	0x0000_0000	R	The status of pipeline	0x00000000
GB_CACHECTL	0x0000_0004	R/W	Cache control register	0x00000000
GB_RST	0x0000_0008	W	The SW reset control	0x00000000
GB_INTPENDING	0x0000_0040	R/W	Interrupt Pending Register	0x00000000
GB_INTMASK	0x0000_0044	R/W	Enables of Disables interrupts.	0x00000000
GB_PIPEMASK	0x0000_0048	R/W	Specifies the blocks in 3D-ACCELERATOR which are candidates to generate interrupts. The bit position of each block is as same as that of GB_PIPESTATE.	0x00000000
GB_PIPETGTSTATE	0x0000_004C	R/W	Specifies the value of pipeline-state when interrupts are to occur. When GB_PIPESTATE becomes GB_PIPETGTSTATE, an interrupt occurs.	0x00000000
GB_PIPEINTSTATE	0x0000_0050	R/W	Captures the first pipeline-state when several interrupts occur.	0x00000000

1.4 SPECIAL FUNCTION REGISTER SUMMARY (HI)

Register	Address	R/W	Description	Reset Value
HI_DWSPACE	0x0000_8000	R	The number of empty slots of the FIFO in HI.	0x00000000
HI_CONTROL	0x0000_8008	R/W	Host interface control register.	0x00010000
HI_IDXOFFSET	0x0000_800C	R/W	Index offset register (signed value)	0x00000001
HI_VBADDR	0x0000_8010	R/W	Vertex Buffer Address	0x00000000
HI_ATTRIB0	0x0000_8040	R/W	Input attribute 0 control register	0x800000E4
HI_ATTRIB1	0x0000_8044	R/W	Input attribute 1 control register	0x800000E4
HI_ATTRIB2	0x0000_8048	R/W	Input attribute 2 control register	0x800000E4
HI_ATTRIB3	0x0000_804C	R/W	Input attribute 3 control register	0x800000E4
HI_ATTRIB4	0x0000_8050	R/W	Input attribute 4 control register	0x800000E4
HI_ATTRIB5	0x0000_8054	R/W	Input attribute 5 control register	0x800000E4
HI_ATTRIB6	0x0000_8058	R/W	Input attribute 6 control register	0x800000E4
HI_ATTRIB7	0x0000_805C	R/W	Input attribute 7 control register	0x800000E4
HI_ATTRIB8	0x0000_8060	R/W	Input attribute 8 control register	0x800000E4
HI_ATTRIB9	0x0000_8064	R/W	Input attribute 9 control register	0x800000E4
HI_ATTRIB0_VBCTRL	0x0000_8080	R/W	Vertex buffer control of input attribute 0	0x00000000
HI_ATTRIB1_VBCTRL	0x0000_8084	R/W	Vertex buffer control of input attribute 1	0x00000000
HI_ATTRIB2_VBCTRL	0x0000_8088	R/W	Vertex buffer control of input attribute 2	0x00000000
HI_ATTRIB3_VBCTRL	0x0000_808C	R/W	Vertex buffer control of input attribute 3	0x00000000
HI_ATTRIB4_VBCTRL	0x0000_8090	R/W	Vertex buffer control of input attribute 4	0x00000000

Register	Address	R/W	Description	Reset Value
HI_ATTRIB5_VBCTRL	0x0000_8094	R/W	Vertex buffer control of input attribute 5	0x00000000
HI_ATTRIB6_VBCTRL	0x0000_8098	R/W	Vertex buffer control of input attribute 6	0x00000000
HI_ATTRIB7_VBCTRL	0x0000_809C	R/W	Vertex buffer control of input attribute 7	0x00000000
HI_ATTRIB8_VBCTRL	0x0000_80A0	R/W	Vertex buffer control of input attribute 8	0x00000000
HI_ATTRIB9_VBCTRL	0x0000_80A4	R/W	Vertex buffer control of input attribute 9	0x00000000
HI_ATTRIB0_VBBASE	0x0000_80C0	R/W	Vertex buffer base address of input attribute 0	0x00000000
HI_ATTRIB1_VBBASE	0x0000_80C4	R/W	Vertex buffer base address of input attribute 1	0x00000000
HI_ATTRIB2_VBBASE	0x0000_80C8	R/W	Vertex buffer base address of input attribute 2	0x00000000
HI_ATTRIB3_VBBASE	0x0000_80CC	R/W	Vertex buffer base address of input attribute 3	0x00000000
HI_ATTRIB4_VBBASE	0x0000_80D0	R/W	Vertex buffer base address of input attribute 4	0x00000000
HI_ATTRIB5_VBBASE	0x0000_80D4	R/W	Vertex buffer base address of input attribute 5	0x00000000
HI_ATTRIB6_VBBASE	0x0000_80D8	R/W	Vertex buffer base address of input attribute 6	0x00000000
HI_ATTRIB7_VBBASE	0x0000_80DC	R/W	Vertex buffer base address of input attribute 7	0x00000000
HI_ATTRIB8_VBBASE	0x0000_80E0	R/W	Vertex buffer base address of input attribute 8	0x00000000
HI_ATTRIB9_VBBASE	0x0000_80E4	R/W	Vertex buffer base address of input attribute 9	0x00000000
HI_DWENTRY	0x0000_C000 ~ 0x0000_DFFF	W	The input port of the FIFO in HI. Burst writes are possible. DWORDs written in 0x0000C000 ~ 0x0000DFFF are stored into FIFO of Host Interface	-
HI_VBDATA	0x0000_E000 ~ 0x0000_FFFF	R/W	*Write* : Used to write geometry data into VB. Burst writes are possible. DWORDs written in 0x0000E000~0x0000FFFF are stored in Vertex Buffer *Read* : The last data written into HI_VBDATA is read.	0x00000000

1.5 SPECIAL FUNCTION REGISTER SUMMARY (VS)

Register	Address	R/W	Description	Reset Value
VS_INSTMEM	0x0001_0000 ~ 0x0001_1FFF	R/W	Instruction memory of the vertex shader	0xX
VS_CFLOAT	0x0001_4000 ~ 0x0001_4FFF	R/W	Constant float register of the vertex shader	
VS_CINT	0x0001_8000 ~ 0x0001_803F	R/W	A constant integer register of vertex shader	0xXXXXXXXX
VS_CBOOL	0x0001_8400	R/W	A constant boolean register of vertex shader	0xXXXXXXXX
VS_Config	0x0001_C800	W	Configuration register of vertex shader	0x00000000
VS_Status	0x0001_C804	R	Internal status register (Reserved)	0x00000000
VS_PCRange	0x0002_0000	R/W	Vertex shader program start and end address	0x01FF0000
VS_AttributeNum	0x0002_0004	R/W	The number of attributes of current context	0x00000000
VS_InAttrIndex0	0x0002_0008	W	Index of input attributes 0~3	0x03020100

VS_InAttrIndex1	0x0002_000C	W	Index of input attributes 4~7	0x07060504
VS_InAttrIndex2	0x0002_0010	W	Index of input attributes 8~11	0x0B0A0908
VS_OutAttrIndex0	0x0002_0014	W	Index of output attributes 0~3	0x03020100
VS_OutAttrIndex1	0x0002_0018	W	Index of output attributes 4~7	0x07060504
VS_OutAttrIndex2	0x0002_001C	W	Index of output attributes 8~11	0x0B0A0908

1.6 SPECIAL FUNCTION REGISTER SUMMARY (PE)

Register	Address	R/W	Description	Reset Value
PE_VERTEX_CONTEXT	0x0003_0000	R/W	Vertex context format definition	0x00000000
PE_VIEWPORT_OX	0x0003_0004	R/W	The x-coordinate of viewport center	0xX
PE_VIEWPORT_OY	0x0003_0008	R/W	The y-coordinate of viewport center	0xX
PE_VIEWPORT_HALF_PX	0x0003_000C	R/W	Half of viewport width	0xX
PE_VIEWPORT_HALF_PY	0x0003_0010	R/W	Half of viewport height	0xX
PE_DEPTHRANGE_HALF_F_SUB_N	0x0003_0014	R/W	Half of depth range far minus near	0x3F000000
PE_DEPTHRANGE_HALF_F_ADD_N	0x0003_0018	R/W	Half of depth range far plus near	0x3F000000

1.7 SPECIAL FUNCTION REGISTER SUMMARY (RA)

Register	Address	R/W	Description	Reset Value
RA_PixSamp	0x0003_8000	R/W	Indicate pixel sampling position register	0x00000000
RA_DOffEn	0x0003_8004	R/W	Depth offset enable register	0x00000000
RA_DOffFactor	0x0003_8008	R/W	Depth offset factor register	0x00000000
RA_DOffUnits	0x0003_800C	R/W	Depth offset units register	0x00000000
RA_DOffRIn	0x0003_8010	R/W	Depth offset implementation constant r value	0x34000001
RA_BFCULL	0x0003_8014	R/W	Back-face culling control register	0x00000000
RA_YCLIP	0x0003_8018	R/W	Y clipping coordinate register	0x00000000
RA_LODCTL	0x0003_C000	R/W	Indicate LOD calculation control register	0x00000000
RA_CLIPX	0x0003_C004	R/W	X clip coordinate register	0x00000000
RA_PWIDTH	0x0003_801C	R/W	Point Width control register	0x3F800000
RA_PSIZE_MIN	0x0003_8020	R/W	Point Width Min value control register	0x3F800000
RA_PSIZE_MAX	0x0003_8024	R/W	Point Width Max value control register	0x45000000
RA_COORDREPLACE	0x0003_8028	R/W	Coord Replace control register	0x00000000
RA_LWIDTH	0x0003_802C	R/W	Line Width control register	0x3F800000

1.8 SPECIAL FUNCTION REGISTER SUMMARY (PS)

Register	Address	R/W	Description	Reset Value
INSTMEM	0x0004_0000 ~0x0004_1FFF	R/W	Instruction memory of pixel shader	0xX
PS_CFLOAT	0x0004_4000 ~ 0x0004_4FFF	R/W	Constant float register of pixel shader	
PS_CINT	0x0004_8000 ~ 0x0004_803F	R/W	A constant integer register of pixel shader	0xXXXXXXXX
PS_CBOOL	0x0004_8400	R/W	Constant bool register of the pixel shader	0x0
PS_ExeMode	0x0004_C800	R/W	Pixel shader execution mode control register	0x0
PS_PCStart	0x0004_C804	R/W	Start address of pixel shader program	0x0
PS_PCEnd	0x0004_C808	R/W	End address of pixel shader program	0x0
PS_PCCopy	0x0004_C80C	R/W	Copy PSPCS_ADDR value to program counter	0x0
PS_AttributeNum	0x0004_C810	R/W	Number of attribute of current context	0x0
PS_IBStatus	0x0004_C814	R	Stauts signal of PS Input Buffer initialization is NotReady	0x0

1.9 SPECIAL FUNCTION REGISTER SUMMARY (TU)

Register	Address	R/W	Description	Reset Value
TU_TSTA0	0x0006_0000	R/W	Texture 0's status	0x08000000
TU_USIZE0	0x0006_0004	R/W	Texture 0's U Size	0x00000000
TU_VSIZE0	0x0006_0008	R/W	Texture 0's V Size	0x00000000
TU_PSIZE0	0x0006_000C	R/W	Texture 0's P Size	0x00000000
TU_TOFFS_L1_0	0x0006_0010	R/W	Texture 0's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_0	0x0006_0014	R/W	Texture 0's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_0	0x0006_0018	R/W	Texture 0's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_0	0x0006_001C	R/W	Texture 0's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_0	0x0006_0020	R/W	Texture 0's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_0	0x0006_0024	R/W	Texture 0's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_0	0x0006_0028	R/W	Texture 0's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_0	0x0006_002C	R/W	Texture 0's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_0	0x0006_0030	R/W	Texture 0's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_0	0x0006_0034	R/W	Texture 0's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_0	0x0006_0038	R/W	Texture 0's Level 11 Texture Offset	0x00000000
TU_T_MIN_L0	0x0006_003C	R/W	Texture 0's Mipmap Min Level	0x00000000
TU_T_MAX_L0	0x0006_0040	R/W	Texture 0's Mipmap Max Level	0x00000000
TU_TBADD0	0x0006_0044	R/W	Texture 0's base address	0x00000000
TU_TSTA1	0x0006_0050	R/W	Texture 1's status	0x08000000
TU_USIZE1	0x0006_0054	R/W	Texture 1's U Size	0x00000000
TU_VSIZE1	0x0006_0058	R/W	Texture 1's V Size	0x00000000
TU_PSIZE1	0x0006_005C	R/W	Texture 1's P Size	0x00000000
TU_TOFFS_L1_1	0x0006_0060	R/W	Texture 1's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_1	0x0006_0064	R/W	Texture 1's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_1	0x0006_0068	R/W	Texture 1's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_1	0x0006_006C	R/W	Texture 1's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_1	0x0006_0070	R/W	Texture 1's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_1	0x0006_0074	R/W	Texture 1's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_1	0x0006_0078	R/W	Texture 1's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_1	0x0006_007C	R/W	Texture 1's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_1	0x0006_0080	R/W	Texture 1's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_1	0x0006_0084	R/W	Texture 1's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_1	0x0006_0088	R/W	Texture 1's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 1	0x0006_008C	R/W	Texture 1's Mipmap Min Level	0x00000000
TU_T_MAX_L1	0x0006_0090	R/W	Texture 1's Mipmap Max Level	0x00000000

Register	Address	R/W	Description	Reset Value
TU_TBADD1	0x0006_0094	R/W	Texture 1's base address	0x00000000
TU_TSTA2	0x0006_00A0	R/W	Texture 2's status	0x08000000
TU_USIZE2	0x0006_00A4	R/W	Texture 2's U Size	0x00000000
TU_VSIZE2	0x0006_00A8	R/W	Texture 2's V Size	0x00000000
TU_PSIZE2	0x0006_00AC	R/W	Texture 2's P Size	0x00000000
TU_TOFFS_L1_2	0x0006_00B0	R/W	Texture 2's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_2	0x0006_00B4	R/W	Texture 2's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_2	0x0006_00B8	R/W	Texture 2's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_2	0x0006_00BC	R/W	Texture 2's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_2	0x0006_00C0	R/W	Texture 2's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_2	0x0006_00C4	R/W	Texture 2's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_2	0x0006_00C8	R/W	Texture 2's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_2	0x0006_00CC	R/W	Texture 2's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_2	0x0006_00D0	R/W	Texture 2's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_2	0x0006_00D4	R/W	Texture 2's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_2	0x0006_00D8	R/W	Texture 2's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 2	0x0006_00DC	R/W	Texture 2's Mipmap Min Level	0x00000000
TU_T_MAX_L2	0x0006_00E0	R/W	Texture 2's Mipmap Max Level	0x00000000
TU_TBADD2	0x0006_00E4	R/W	Texture 2's base address	0x00000000
TU_TSTA3	0x0006_00F0	R/W	Texture 3's status	0x08000000
TU_USIZE3	0x0006_00F4	R/W	Texture 3's U Size	0x00000000
TU_VSIZE3	0x0006_00F8	R/W	Texture 3's V Size	0x00000000
TU_PSIZE3	0x0006_00FC	R/W	Texture 3's P Size	0x00000000
TU_TOFFS_L1_3	0x0006_0100	R/W	Texture 3's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_3	0x0006_0104	R/W	Texture 3's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_3	0x0006_0108	R/W	Texture 3's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_3	0x0006_010C	R/W	Texture 3's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_3	0x0006_0110	R/W	Texture 3's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_3	0x0006_0114	R/W	Texture 3's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_3	0x0006_0118	R/W	Texture 3's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_3	0x0006_011C	R/W	Texture 3's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_3	0x0006_0120	R/W	Texture 3's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_3	0x0006_0124	R/W	Texture 3's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_3	0x0006_0128	R/W	Texture 3's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 3	0x0006_012C	R/W	Texture 3's Mipmap Min Level	0x00000000
TU_T_MAX_L3	0x0006_0130	R/W	Texture 3's Mipmap Max Level	0x00000000
TU_TBADD3	0x0006_0134	R/W	Texture 3's base address	0x00000000

Register	Address	R/W	Description	Reset Value
TU_TSTA4	0x0006_0140	R/W	Texture 4's status	0x08000000
TU_USIZE4	0x0006_0144	R/W	Texture 4's U Size	0x00000000
TU_VSIZE4	0x0006_0148	R/W	Texture 4's V Size	0x00000000
TU_PSIZE4	0x0006_014C	R/W	Texture 4's P Size	0x00000000
TU_TOFFS_L1_4	0x0006_0150	R/W	Texture 4's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_4	0x0006_0154	R/W	Texture 4's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_4	0x0006_0158	R/W	Texture 4's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_4	0x0006_015C	R/W	Texture 4's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_4	0x0006_0160	R/W	Texture 4's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_4	0x0006_0164	R/W	Texture 4's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_4	0x0006_0168	R/W	Texture 4's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_4	0x0006_016C	R/W	Texture 4's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_4	0x0006_0170	R/W	Texture 4's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_4	0x0006_0174	R/W	Texture 4's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_4	0x0006_0178	R/W	Texture 4's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 4	0x0006_017C	R/W	Texture 4's Mipmap Min Level	0x00000000
TU_T_MAX_L4	0x0006_0180	R/W	Texture 4's Mipmap Max Level	0x00000000
TU_TBADD4	0x0006_0184	R/W	Texture 4's base address	0x00000000
TU_TSTA5	0x0006_0190	R/W	Texture 5's status	0x08000000
TU_USIZE5	0x0006_0194	R/W	Texture 5's U Size	0x00000000
TU_VSIZE5	0x0006_0198	R/W	Texture 5's V Size	0x00000000
TU_PSIZE5	0x0006_019C	R/W	Texture 5's P Size	0x00000000
TU_TOFFS_L1_5	0x0006_01A0	R/W	Texture 5's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_5	0x0006_01A4	R/W	Texture 5's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_5	0x0006_01A8	R/W	Texture 5's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_5	0x0006_01AC	R/W	Texture 5's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_5	0x0006_01B0	R/W	Texture 5's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_5	0x0006_01B4	R/W	Texture 5's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_5	0x0006_01B8	R/W	Texture 5's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_5	0x0006_01BC	R/W	Texture 5's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_5	0x0006_01C0	R/W	Texture 5's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_5	0x0006_01C4	R/W	Texture 5's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_5	0x0006_01C8	R/W	Texture 5's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 5	0x0006_01CC	R/W	Texture 5's Mipmap Min Level	0x00000000

Register	Address	R/W	Description	Reset Value
TU_T_MAX_L5	0x0006_01D0	R/W	Texture 5's Mipmap Max Level	0x00000000
TU_TBADD5	0x0006_01D4	R/W	Texture 5's base address	0x00000000
TU_TSTA6	0x0006_01E0	R/W	Texture 6's status	0x08000000
TU_USIZE6	0x0006_01E4	R/W	Texture 6's U Size	0x00000000
TU_VSIZE6	0x0006_01E8	R/W	Texture 6's V Size	0x00000000
TU_PSIZE6	0x0006_01EC	R/W	Texture 6's P Size	0x00000000
TU_TOFFS_L1_6	0x0006_01F0	R/W	Texture 6's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_6	0x0006_01F4	R/W	Texture 6's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_6	0x0006_01F8	R/W	Texture 6's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_6	0x0006_01FC	R/W	Texture 6's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_6	0x0006_0200	R/W	Texture 6's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_6	0x0006_0204	R/W	Texture 6's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_6	0x0006_0208	R/W	Texture 6's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_6	0x0006_020C	R/W	Texture 6's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_6	0x0006_0210	R/W	Texture 6's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_6	0x0006_0214	R/W	Texture 6's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_6	0x0006_0218	R/W	Texture 6's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 6	0x0006_021C	R/W	Texture 6's Mipmap Min Level	0x00000000
TU_T_MAX_L6	0x0006_0220	R/W	Texture 6's Mipmap Max Level	0x00000000
TU_TBADD6	0x0006_0224	R/W	Texture 6's base address	0x00000000
TU_TSTA7	0x0006_0230	R/W	Texture 7's status	0x08000000
TU_USIZE7	0x0006_0234	R/W	Texture 7's U Size	0x00000000
TU_VSIZE7	0x0006_0238	R/W	Texture 7's V Size	0x00000000
TU_PSIZE7	0x0006_023C	R/W	Texture 7's P Size	0x00000000
TU_TOFFS_L1_7	0x0006_0240	R/W	Texture 7's Level 1 Texture Offset	0x00000000
TU_TOFFS_L2_7	0x0006_0244	R/W	Texture 7's Level 2 Texture Offset	0x00000000
TU_TOFFS_L3_7	0x0006_0248	R/W	Texture 7's Level 3 Texture Offset	0x00000000
TU_TOFFS_L4_7	0x0006_024C	R/W	Texture 7's Level 4 Texture Offset	0x00000000
TU_TOFFS_L5_7	0x0006_0250	R/W	Texture 7's Level 5 Texture Offset	0x00000000
TU_TOFFS_L6_7	0x0006_0254	R/W	Texture 7's Level 6 Texture Offset	0x00000000
TU_TOFFS_L7_7	0x0006_0258	R/W	Texture 7's Level 7 Texture Offset	0x00000000
TU_TOFFS_L8_7	0x0006_025C	R/W	Texture 7's Level 8 Texture Offset	0x00000000
TU_TOFFS_L9_7	0x0006_0260	R/W	Texture 7's Level 9 Texture Offset	0x00000000
TU_TOFFS_L10_7	0x0006_0264	R/W	Texture 7's Level 10 Texture Offset	0x00000000
TU_TOFFS_L11_7	0x0006_0268	R/W	Texture 7's Level 11 Texture Offset	0x00000000
TU_T_MIN_L 7	0x0006_026C	R/W	Texture 7's Mipmap Min Level	0x00000000
TU_T_MAX_L7	0x0006_0270	R/W	Texture 7's Mipmap Max Level	0x00000000

Register	Address	R/W	Description	Reset Value
TU_TBADD7	0x0006_0274	R/W	Texture 7's base address	0x00000000
TU_CKEY1	0x0006_0280	R/W	3D color key1 register	0x00000000
TU_CKEY2	0x0006_0284	R/W	3D color key2 register	0x00000000
TU_CKYUV	0x0006_0288	R/W	3D color key YUV register	0x00000000
TU_CKMASK	0x0006_028C	R/W	3D color key mask register	0x00000000
TU_PALLETTE_ADDR	0x0006_0290	W	Palette address for indexed texture	0x00000000
TU_PALLETTE_IN	0x0006_0294	W	Palette data in	0x00000000
VT_VTSTA0	0x0006_02C0	R/W	Vertex texture 0's status	0x00000000
VT_VTSTA1	0x0006_02C8	R/W	Vertex texture 1's status	0x00000000
VT_VTSTA2	0x0006_02D0	R/W	Vertex texture 2's status	0x00000000
VT_VTSTA3	0x0006_02D8	R/W	Vertex texture 3's status	0x00000000
VT_VTBADDR0	0x0006_02C4	R/W	Vertex texture 0's base address	0x00000000
VT_VTBADDR1	0x0006_02CC	R/W	Vertex texture 1's base address	0x00000000
VT_VTBADDR2	0x0006_02D4	R/W	Vertex texture 2's base address	0x00000000
VT_VTBADDR3	0x0006_02DC	R/W	Vertex texture 3's base address	0x00000000

1.10 SPECIAL FUNCTION REGISTER SUMMARY (PF)

Register	Address	R/W	Description	Reset Value
PF_SCISSOR_X	0x0007_0000	R/W	X coordinate pixel clipping region (X coordinate's range is from 0 to 2047)	0x00000000
PF_SCISSOR_Y	0x0007_0004	R/W	Y coordinate pixel clipping region (Y coordinate's range is from 0 to 2047)	0x00000000
PF_ALPHAT	0x0007_0008	R/W	Alpha test control register	0x00000000
PF_FRONTST	0x0007_000C	R/W	Front face stencil test control register	0x0000000
PF_BACKST	0x0007_0010	R/W	Back face stencil test control register	0x0000000
PF_DEPTHT	0x0007_0014	R/W	Depth test control register	0x00000002
PF_CCLR	0x0007_0018	R/W	Blend constant color	0x00000000
PF_BLEND	0x0007_001C	R/W	Blending control register	0x00000000
PF_LOGOP	0x0007_0020	R/W	RGBA color logical operation enable & function	0x00000000
PF_CBMSK	0x0007_0024	R/W	Color write mask in RGBA mode	0x00000000
PF_DBMSK	0x0007_0028	R/W	Depth/Stencil buffer write mask	0x00000000
PF_FBCTL	0x0007_002C	R/W	Frame buffer write control register	0x00000000
PF_DBADDR	0x0007_0030	R/W	Depth buffer offset address	0x00000000
PF_CBADDR	0x0007_0034	R/W	Color buffer offset address	0x0000000
PF_FBW	0x0007_0038	R/W	Frame buffer width	0x00000800

2 GLOBAL REGISTER

2.1 OVERVIEW

Global registers is a set of overall states in 3D-ACCELERATOR.

2.2 THE ROLE OF GB_PIPESTATE SFRS

Each bit field in GB_PIPESTATE represents whether the corresponding block processes the geometry data. If one of bits is 1, then this means the geometry data is processed in the corresponding block. The 0 value represents the corresponding block waits for the geometry data and does nothing. GB_PIPESTATE is used to determine the timming when the state of each block is updated. For example, after CPU sending the geometry data, CPU wants to set the next state of the per-fragment unit. If CPU updated the new state of the per-fragment unit when the previsous geometry data is in the vertex shader, the remain data would be affected by the new state of the per-fragment unit. The result would be wrong. In this case, CPU checks the GB_PIPESTATE and determines where the gemoetry data is processed. CPU waits for the geometry data to be transferred after the per-fragment unit. Only when all the blocks before the per-fragment unit is free, the state of the per-fragment unit can be updated.

All the geometry data can be processed and sent to the frame buffer. At this moment, CPU can update the state of the per-fragment unit safely. However, this can affect the performace of 3D-ACCELERATOR waiting the whole pipeline to be empty. Regarding to the perfomance, this is not desirable. If CPU knows the proper time to update states, then the perfomance will be increased. This is the reason why GB_PIPESTATE exists.

2.3 DATA TRANSFER USING 3D-ACCELERATOR'S INTERRTUP

The data transfer includes the modification of SFR values and the geometry data transfer; interrupts can be used to change SFR values and send geometry data.

Interrupts from 3D-ACCELERATOR's pipeline-state can be used to know when to change SFR values for a 3D-ACCELERATOR block. SFR values for a block can be changed only when the previous blocks are empty for the safe operation. Otherwise, the remained geometry data in the 3D-ACCELERATOR pipeline is affected by the new SFR value instead of the previous SFR values intended to be applied to geometry data. CPU can repeat to read the pipeline-state, which is known as polling, to know when to update SFRs. However, CPU should do another job instead of investigating pipeline-state and spending cycles. In this case, CPU can set interrupt conditions and do another job. If the interrupt condition is met and an interrupt occurs, CPU can change the SFR values.

Interrupts can be used to transfer geometry data. CPU transfers geometry data when there is free space in the Host-FIFO of 3D-ACCELERATOR's Host Interface. CPU can keep watching pipeline-state in order to know when to transfer the other geometry data to the Host-FIFO of 3D-ACCELERATOR's Host Interface with polling. This wastes performance investigating the pipeline-state. Interrupts can be used for this situation. In this case, CPU sets interrupt conditions for the next geometry data after sending a bunch of geometry data and performs another pending job. When an interrupt occurs, CPU can transfer the rest of geometry data to 3D-ACCELERATOR.

Figure 9.6- 2 shows an example illustrating how to transfer geometry data.

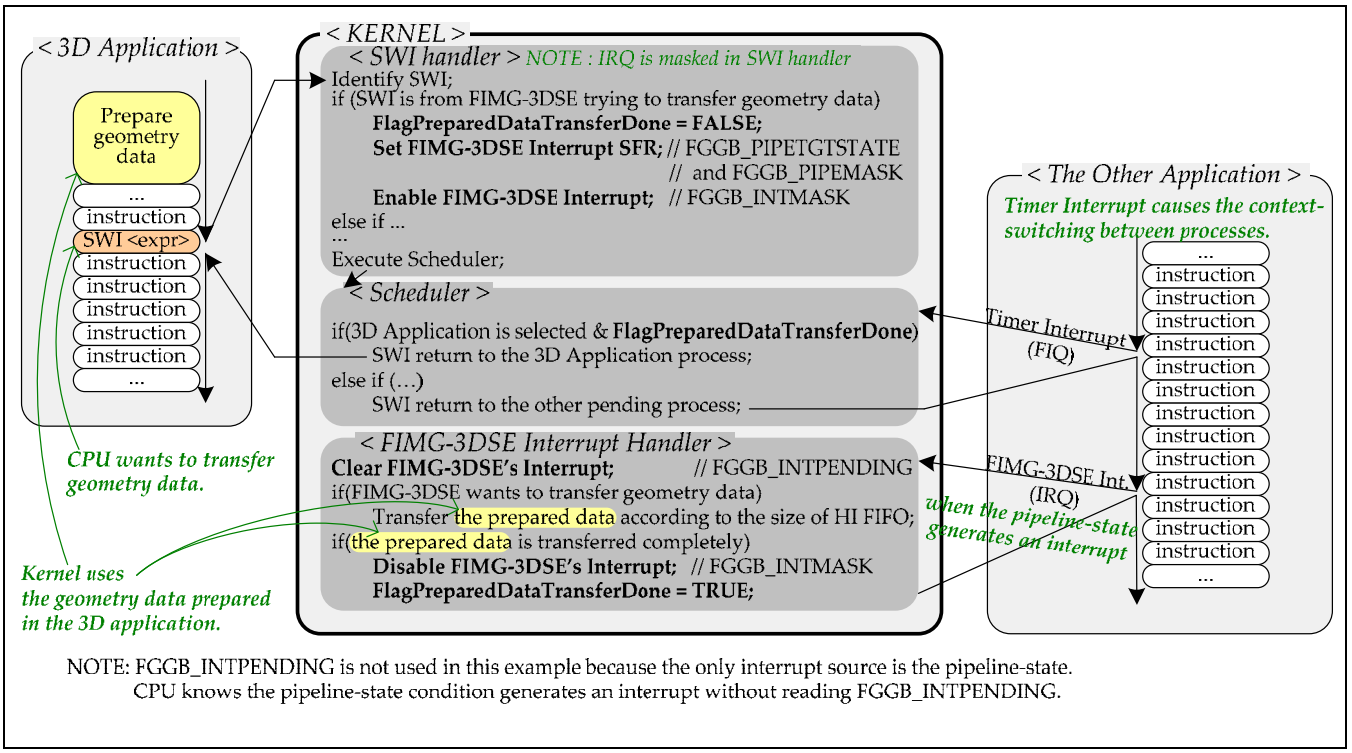


Figure 9.6- 3 A Simple Example of Transferring Geometry Data. (Other Methods Can be Used as Well.)

Note that when 3D-ACCELERATOR is empty, the pipeline-state generates an interrupt right after 3D-ACCELERATOR interrupt is enabled. However, the 3D-ACCELERATOR interrupt handler is executed when CPU executes the other user-mode application program (IRQ interrupt in enabled); in the above figure.

Figure 9.6- 4 illustrates how to change SFR values. SFR can be changed when all the pipe-state becomes empty.

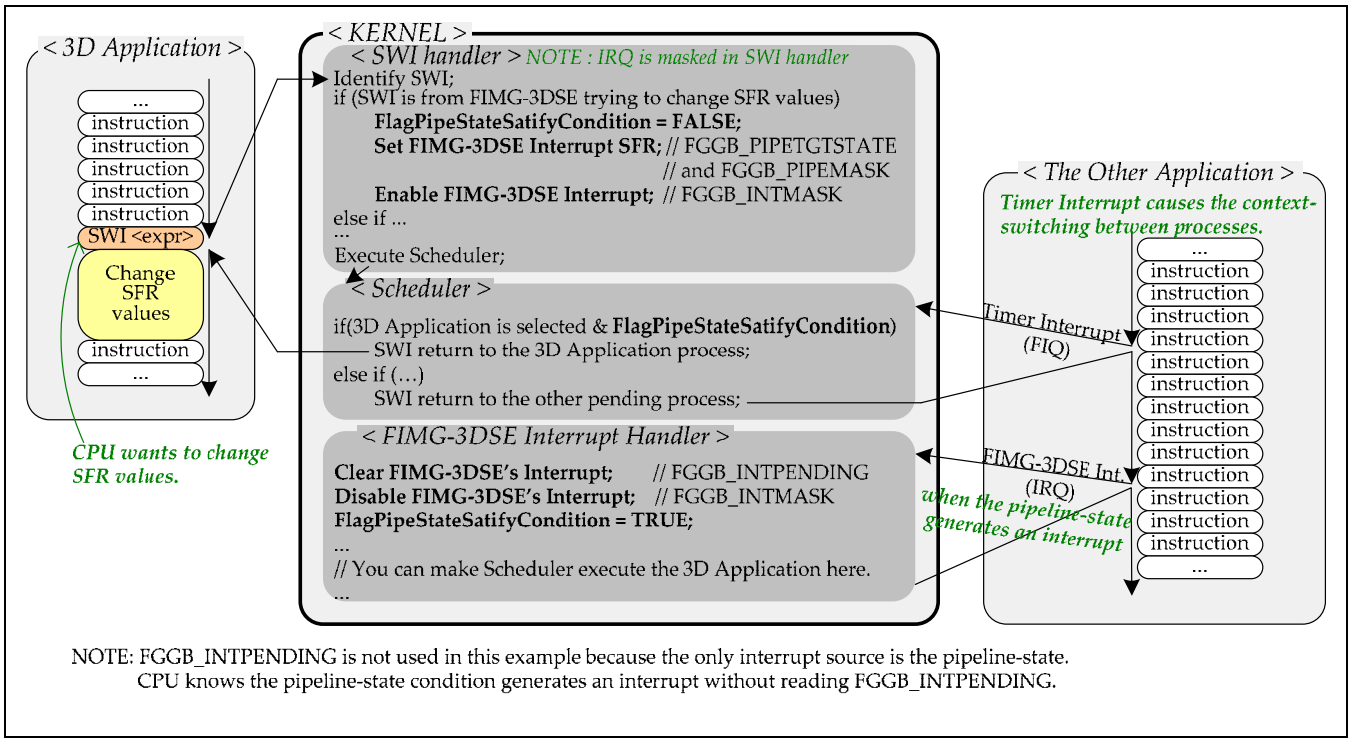


Figure 9.6- 5 A Simple Example of Changing SFR Values. (Other Methods Can be Used as Well.)

Note that the above scheme can be used also to transfer geometry data.

2.4 GLOBAL SPECIAL REGISTERS

2.4.1 Pipeline Status Register (GB_PIPESTATE, R, Address = 0X0000_0000)

GB_PIPESTATE	Bit	Description	Reset Value
Reserved	[31:18]	Reserved	0
PF1	[17]	0b = per-fragment unit 1 is empty. 1b = per-fragment unit 1 is not empty (busy).	0b
PF0	[16]	0b = per-fragment unit 0 is empty. 1b = per-fragment unit 0 is not empty (busy).	0b
Reserved	[15:14]	Reserved	0
PS1	[13]	0b = pixel shader unit 1 is empty. 1b = pixel shader unit 1 is not empty (busy).	0b
PS0	[12]	0b = pixel shader unit 0 is empty. 1b = pixel shader unit 0 is not empty (busy).	0b
Reserved	[11]	Reserved	0
RA	[10]	0b = raster engine is empty. 1b = raster engine is not empty (busy).	0b
TSE	[9]	0b = triangle setup engine is empty. 1b = triangle setup engine is not empty (busy).	0b
PE	[8]	0b = primitive engine is empty. 1b = primitive engine is not empty (busy).	0b
Reserved	[7:5]	Reserved	0
VS	[4]	0b = vertex shader is empty. 1b = vertex shader is not empty (busy).	0b
VC	[3]	0b = vertex cache is empty. 1b = vertex cache is not empty (busy).	0b
HVF	[2]	0b = FIFO between Host Interface and vertex shader is empty. 1b = FIFO between Host Interface and vertex shader is not empty (busy).	0
HI	[1]	0b = Host Interface is empty. 1b = Host Interface is not empty (busy).	0b
HOST-FIFO	[0]	0b = Host-FIFO in Host Interface is empty. 1b = Host-FIFO in Host Interface is not empty (busy).	0b

2.4.2 Cache Control Register (GB_CACHECTL, R/W, Address = 0X0000_0004)

If you set VTCCLEAR to 1, VTCCLEAR becomes 0 automatically after a cycle. TCCLEAR is used to invalidate the contents of texture cache0 and texture cache1. You can set TCCLEAR to 01, 10, or 11. After a cycle, TCCLEAR becomes 00. CCFLUSH and ZCFLUSH fields in GB_CACHECTL are used to flush cache data into color and z buffer. If you set CCFLUSH to 11, CCFULSH becomes 00 automatically when the flush operation is completed. ZCFLUSH's operation is as same as CCFULSH.

GB_CACHECTL	Bit	Description	Reset Value
Reserved	[31:13]	Reserved	0
VTCCLEAR	[12]	Vertex texture cache clear (Automatically set to 0b after a cycle) 0 = default states; vertex texture cache invalidation unchanged. 1 = vertex texture cache starts invalidation.	0b
Reserved	[11:10]	Reserved	0
TCCLEAR	[9:8]	Texture cache clear (Automatically set to 0b after a cycle) 00 = default states; both texture cache0 and cache1 unchanged. 01 = texture cache0 starts invalidation; texture cache1 unchanged 10 = texture cache1 starts invalidation; texture cache0 unchanged. 11 = both texture cache0 and cache1 start invalidation	00b
Reserved	[7:6]	Reserved	0
CCFLUSH	[5:4]	Color cache flush (Automatically set to 00b after flushing) 00 = color cache0, cache1 flush end 11 = color cache0, cache1 flush start	00b
Reserved	[3:2]	Reserved	0
ZCFLUSH	[1:0]	Z cache flush (Automatically set to 00b after flushing) 00b = Z cache0, cache1 flush end 11b = Z cache0, cache1 flush start	00b

2.4.3 Software Reset Register (GB_RST, W, Address = 0X0000_0008)

You can reset the core of 3D-ACCELERATOR with GB_RST register. However, the SFR values are not affected by GB_RST. The reset bit of GB_RST is not recovered to 0 automatically. You must set GB_RST to 0 for the 3D-ACCELERATOR's operation.

GB_RST	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
Reset	[0]	Reset signal for 3D-ACCELERATOR core (logic and internal memory) 1 = Reset, 0 = Work	0b

2.4.4 Version Information Register (GB_VERSION, R, Address = 0X0000_0010)

By reading GB_INFO register, you can identify which 3D-ACCELERATOR is implemented in a system.

GB_VERSION	Bit	Description	Reset Value
major	[31:24]	Major version.	0x01
minor	[23:0]	Minor version. Ex) Version 1.2 : GB_INFO = 0x01020000	0x020100

2.4.5 Interrupt Pending Register (GB_INTPENDING, R/W, Address = 0X0000_0040)

When CPU receives an interrupt from 3D-ACCELERATOR, CPU must investigate which functional block in 3D-ACCELERATOR generates an interrupt. CPU can figure out the interrupt-generating block by reading GB_INTPENDING.

Any value must be written into GB_INTPENDING in the interrupt service routine to clear interrupts from 3D-ACCELERATOR. By writing any value into GB_INTPENDING, GB_INTPENDING is automatically cleared and 3D-ACCELERATOR can generate another interrupt. The written value into GB_INTPENDING is not important; the write operation into GB_INTPENDING clears its value.

Currently, GB_PIPESTATE (Pipeline-State) in HI can only generate an interrupt. Once 3D-ACCELERATOR generates an interrupt, CPU knows that GB_PIPESTATE is the interrupt source without reading GB_INTPENDING.

GB_INTPENDING	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
Pipeline-State	[0]	Read: "Pipeline State interrupt" is generated. 1 = Interrupt Occurs, 0 = No Interrupt. Write: Clear the value into zero. The written value is not important.	0b

2.4.6 Interrupt Mask Register (GB_INTMASK, R/W, Address = 0X0000_0044)

GB_INTMASK can enable or disable interrupts from 3D-ACCELERATOR. Currently, interrupts can be generated only by HI (Pipeline-State). Hence, LSB of GB_INTMASK is used to enable or disable interrupts.

Note: There is another method to disable interrupts from the Pipeline Status; refer to the explanation for the GB_PIPEMASK. GB_INTMASK is the global control while GB_PIPEMASK is the bit-wise control.

GB_INTMASK	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
Pipeline State	[0]	"Pipeline State" generates an interrupt. 1 = Enable Interrupt, 0 = Disable Interrupt.	0b

2.4.7 Pipeline Mask Register (GB_PIPEMASK, R/W, Address = 0X0000_0048)

GB_PIPEMASK specifies the interesting 3D-ACCELERATOR block for interrupt generation. The 3D-ACCELERATOR blocks having value one in GB_PIPEMASK are candidates for interrupts; the blocks having zero value are ignored during interrupt generation.

GB_PIPEMASK	Bit	Description	Reset Value
Reserved	[31:18]	Reserved	0
PF1	[17]	0b = don't care 1b = used to generate interrupts	0b
PF0	[16]	0b = don't care 1b = used to generate interrupts	0b
Reserved	[15:14]	Reserved	0
PS1	[13]	0b = don't care 1b = used to generate interrupts	0b
PS0	[12]	0b = don't care 1b = used to generate interrupts	0b
Reserved	[11]	Reserved	0
RA	[10]	0b = don't care 1b = used to generate interrupts	0b
TSE	[9]	0b = don't care 1b = used to generate interrupts	0b
PE	[8]	0b = don't care 1b = used to generate interrupts	0b
Reserved	[7:5]	Reserved	0
VS	[4]	0b = don't care 1b = used to generate interrupts	0b
VC	[3]	0b = don't care 1b = used to generate interrupts	0b
HVF	[2]	0b = don't care 1b = used to generate interrupts	0b
HI	[1]	0b = don't care 1b = used to generate interrupts	0b
HOSTFIFO	[0]	0b = don't care 1b = used to generate interrupts	0b

2.4.8 Pipeline Target State Register (GB_PIPETGTSTATE, R/W, Address = 0X0000_004C)

As mentioned before, GB_PIPEMASK specifies the interesting 3D-ACCELERATOR block for interrupt generation. GB_PIPETGTSTATE specifies the value of pipeline-state when interrupts occur. Note that the GB_PIPETGTSTATE value for a block with 0 value in GB_PIPEMASK is ignored.

GB_PIPETGTSTATE	Bit	Description	Reset Value
Reserved	[31:18]	Reserved	0
PF1	[17]	0b = interrupts when the PF1 is not working. (Empty) 1b = interrupts when the PF1 is working. (not-empty)	0b
PF0	[16]	0b = interrupts when the PF0 is not working. (Empty) 1b = interrupts when the PF0 is working. (not-empty)	0b
Reserved	[15:14]	Reserved	0
PS1	[13]	0b = interrupts when the PS1 is not working. (Empty) 1b = interrupts when the PS1 is working. (not-empty)	0b
PS0	[12]	0b = interrupts when the PS0 is not working. (Empty) 1b = interrupts when the PS0 is working. (not-empty)	0b
Reserved	[11]	Reserved	0
RA	[10]	0b = interrupts when the RA is not working. (Empty) 1b = interrupts when the RA is working. (not-empty)	0b
TSE	[9]	0b = interrupts when the TSE is not working. (Empty) 1b = interrupts when the TSE is working. (not-empty)	0b
PE	[8]	0b = interrupts when the PE is not working. (Empty) 1b = interrupts when the PE is working. (not-empty)	0b
Reserved	[7:5]	Reserved	0
VS	[4]	0b = interrupts when the VS is not working. (Empty) 1b = interrupts when the VS is working. (not-empty)	0b
VC	[3]	0b = interrupts when the VC is not working. 1b = interrupts when the VC is working.	0
HVF	[2]	0b = interrupts when the FIFO between HI and VS is empty. 1b = interrupts when the FIFO between HI and VS is not-empty.	0b
HI	[1]	0b = interrupts when the HI is not working. (Empty) 1b = interrupts when the HI is working. (not-empty)	0b
HOSTFIFO	[0]	0b = interrupts when the Host-FIFO is not working. (Empty) 1b = interrupts when the Host-FIFO is working. (not-empty)	0b

2.4.9 Pipeline Interrupt State Register (GB_PIPEINTSTATE, R, Address = 0X0000_0050)

GB_PIPEINTSTATE captures the pipeline-state when interrupts occur. When several interrupts occur, the GB_PIPEINTSTAE holds the first pipeline-state.

Note that GB_PIPEINTSTATE depends on GB_PIPEMASKE.

GB_PIPEINTSTATE	Bit	Description	Reset Value
Reserved	[31:18]	Reserved	0
PF1	[17]	0b = the PF1 was empty when an interrupt occurred. 1b = the PF1 was not empty when an interrupt occurred.	0b
PF0	[16]	0b = the PF0 was empty when an interrupt occurred. 1b = the PF0 was not empty when an interrupt occurred.	0b
Reserved	[15:14]	Reserved	0
PS1	[13]	0b = the PS1 was empty when an interrupt occurred. 1b = the PS1 was not empty when an interrupt occurred.	0b
PS0	[12]	0b = the PS0 was empty when an interrupt occurred. 1b = the PS0 was not empty when an interrupt occurred.	0b
Reserved	[11]	Reserved	0
RA	[10]	0b = the RA was empty when an interrupt occurred. 1b = the RA was not empty when an interrupt occurred.	0b
TSE	[9]	0b = the TSE was empty when an interrupt occurred. 1b = the TSE was not empty when an interrupt occurred.	0b
PE	[8]	0b = the PE was empty when an interrupt occurred. 1b = the PE was not empty when an interrupt occurred.	0b
Reserved	[7:5]	Reserved	0
VS	[4]	0b = the VS was empty when an interrupt occurred. 1b = the VS was not empty when an interrupt occurred.	0b
VC	[3]	0b = the VC was empty when an interrupt occurred. 1b = the VC was not empty when an interrupt occurred.	0b
HVF	[2]	0b = the FIFO between HI and VS was empty when an interrupt occurred. 1b = the FIFO between HI and VS was not empty when an interrupt occurred.	0
HI	[1]	0b = the HI was empty when an interrupt occurred. 1b = the HI was not empty when an interrupt occurred.	0b
HOSTFIFO	[0]	0b = the Host-FIFO was empty when an interrupt occurred. 1b = the Host-FIFO was not empty when an interrupt occurred.	0b

3 HOST INTERFACE

3.1 OVERVIEW

The major function of the Host Interface unit is to receive data from CPU converting several data format into floating point data format. The Host Interface also transfers state data (SFR values, VS inst-memory, etc) to CPU.

The data transferred from CPU to 3D-ACCELERATOR are classified according to the data characteristics; state and geometry data. The CPU set the state data first. And then, the geometry data is transferred to 3D-ACCELERATOR. 3D-ACCELERATOR renders the transferred geometry data using the state data. For the next geometry data to be rendered, the appropriate state data must be transferred to 3D-ACCELERATOR. At this moment, the previous geometry data can be affected by the new state. Hence, state data can be transferred to the 3D-ACCELERATOR only when the state data does not affect the previous geometry. PIPELINE STATUS REGISTER (GB_PIPESTATE), which is explained the previous chapter, is used to query where the geometry data is processed in 3D-ACCELERATOR. (CPU decides when to update the states using GB_PIPESTATE.) Suppose there are blocks, A, B, C, and D in 3D-ACCELERATOR and the state data of C is to be updated. If B processes the geometry data, then CPU must not update the state of C because the geometry data in B can be affected by the new state of C instead of the previous state. CPU must wait for the geometry data in B to pass through C and reach to D. At this time, CPU can update the state of C because A, B, and C are empty (there is no geometry to be affected by the new state data).

If the contents of the frame buffer are to be used for textures, CPU must copy the frame buffer data to the texture.

The next section describes how to feed the geometry data to the Host Interface.

In this document, DWORD represents 32-bit data.

A vertex is composed of several attributes. Vertex attributes are used in the vertex shader program. The vertex shader program determines what the given attributes represent; they can be vertex position, vertex color, normal vector, texture coordinates, and so on. Usually, these vertex attributes are supposed to be transferred from CPU (or DMA) requiring lots of memory bandwidth. The vertex buffer, which is 4KB internal memory in 3D-ACCELERATOR, can be used to reduce memory bandwidth. After the geometry data, arrays of vertex attributes in other words, is saved in the vertex array, a series of indices from CPU can be transferred instead of vertex attributes saving memory bandwidth. Each index represents an array index in each vertex attributes. (HI_ATTR0_VBBASE~ HI_ATTR9_VBBASE SFR represent the base addresses of vertex attributes in the vertex buffer.) Section 3.3 shows how the geometry data is transferred to the internal host interface hardware and what kind of SFRs are used.

3.2 OPERATING MODE

Two modes can be used for Host Interface depending on the way how to transfer the geometry data: index mode, and non-index mode.

Index mode: After CPU stores the geometry data (input attributes to vertex shader) into the Vertex Buffer in Host Interface, CPU transfers the indices to the stored geometry. This scheme consumes low bus bandwidth.

Non-index mode: CPU transfers the geometry data directly. This scheme is useful when there is no space in the Vertex Buffer. Also the *infrequently* used geometry data can be transferred to Host Interface using this scheme.

3.2.1 Index Mode

Index mode **uses the Vertex Buffer** in Host Interface. There are two operation modes: auto-increment mode and index-transfer mode. NOTE: All of the geometry data must be in the Vertex Buffer. Also, the range of used indices is deeply related to the VB size: the VB address calculated with indices must be the available VB address. Hence the indices must be carefully controlled by the application program (or the device driver).

- **Auto-increment mode (HI_CONTROL.EnVB=1, HI_CONTROL.AutoInc=1)**: CPU sends two DWORDs representing a count (the number of vertices) and an index (the first index into the Vertex Buffer), to the Host Interface. Host Interface uses the transferred index for the index to the Vertex Buffer. And then, the next index is automatically calculated; HI_IDXOFFSET.VALUE is added to the previous index (usually set to 1). This process is repeated count-times. Each pair of DWORDs (count and index value) represents a set of indices. Therefore, this scheme maximizes the performance in transferring geometry data.

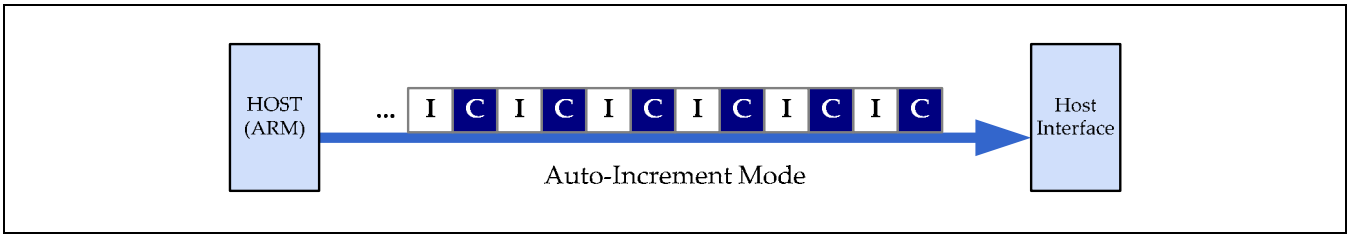


Figure 9.6- 6 Auto-Increment Mode

In Figure 9.6- 7, six sets of the geometries are transferred from CPU.

- **Index-transfer mode (HI_CONTROL.EnVB=1, HI_CONTROL.AutoInc=0)**: In index-transfer mode, CPU sends the individual indices. After CPU sending the number of indices (count), a set of random indices are transferred. These indices are used to index the Vertex Buffer. {TO BE DONE: HI_IDXOFFSET can be used to bias the sent index; (sent index + HI_IDXOFFSET) is used as a new index}.

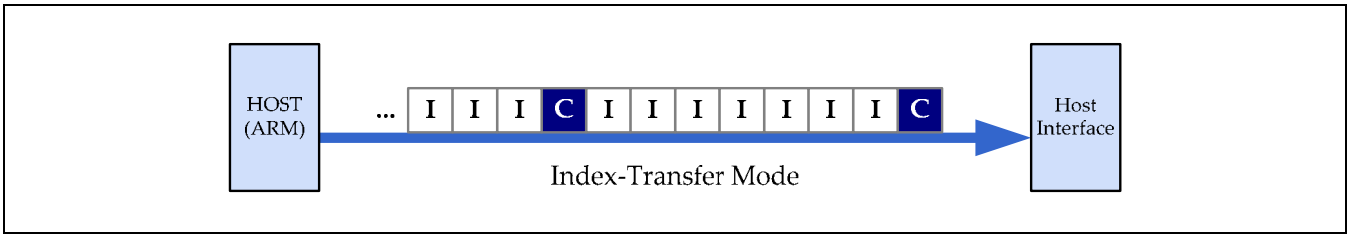


Figure 9.6- 8 Index-Transfer Mode

3.2.2 Non-Index Mode (HI_CONTROL.ENVB=0, HI_CONTROL.AUTOINC=1)

In non-index mode, ***the Vertex Buffer is not used***. Instead, CPU sends all of the geometry data. Just like index-mode, the number of vertices and an index must be transferred first. In this case, a dummy value (0xFFFFFFFF) must be used as an index.

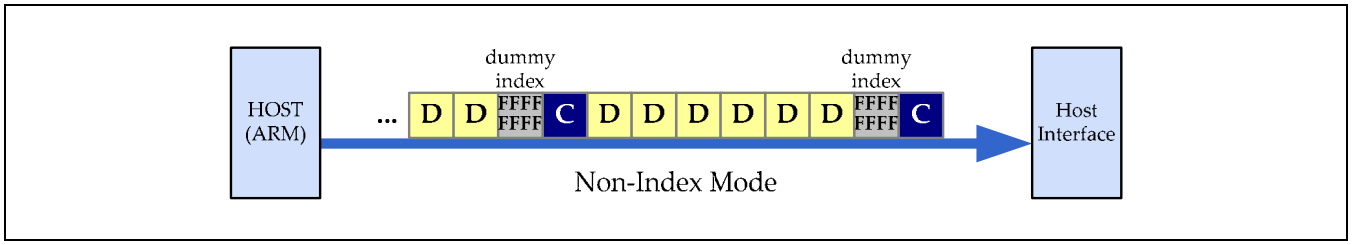


Figure 9.6- 9 Non-Index Mode

The transferred data are decoded using HI_ATTRIB0~HI_ATTRIB9.

Note that when float or half-float format data is transferred to the Host-Interface, NaN or Infinite values must not be transferred.

3.3 HOW TO TRANSFER GEOMETRY DATA

A geometry data is transferred from CPU or from the Vertex Buffer in the Host Interface after special registers for Host Interface are set (explained in the next section). The geometry data is a set of input attributes to vertex shader. In other words, a number of attributes for a vertex can be transferred from CPU or from the Vertex Buffer. (From time to time, a vertex can have only an attribute.) A set of attributes constituting of a vertex is defined in HI_ATTRIB0~HI_ATTRIB9. The number of components in each attributes is determined by HI_ATTRIB*n*.NumComp. The LastAttr bit in HI_ATTRIB*n* represents whether the HI_ATTRIB*n* is used or not. For example, HI_ATTRIB0.LastAttr = HI_ATTRIB1.LastAttr = HI_ATTRIB2.LastAttr = 0 and HI_ATTRIB3.LastAttr = 1 mean that a vertex is composed of 4 attributes. (HI_ATTRIB4.LastAttr and so on do not determine the number of attributes for a vertex.)

The number of vertices (count) is transferred only by CPU. (The count is written into HI_DWENTRY of the Host Interface.) Then, CPU can transfer indices or raw geometry data into the same HI_DWENTRY. However, the Vertex Buffer can feed only geometry data to the Host Interface. The usage of the Vertex Buffer is determined by HI_CONTROL.EnVB. If the Vertex Buffer is used, indices are required to index the geometry data in the Vertex Buffer. The required indices can be sent by CPU (index-transfer mode) or can be generated in the Host Interface (auto-increment mode). There can be several indices in a DWORD from CPU depending on HI_CONTROL.IdxType (data type of index). These transferred or generated indices are used in the following way.

```
1.  Get a count from CPU
2.  Get an index from CPU or previous index;  // depending on HI_CONTROL.AutoInc. In Non-Index mode, 0xFFFFFFFF is used.
3.  Add HI_IDXOFFSET.VALUE to the index  for Index mode // In Non-Index mode, this step is skipped.
4.  for each n from 0x0 to 0xF
5.     if(HI_CONTROL.EnVB == 1 && HI_ATTRIB[n]_VBCTRL.Range != 0 && index < HI_ATTRIB[n]_VBCTRL.Range)
6.        Use DWORDs fetched from
              VertexBuffer[HI_ATTRIB[n]_VBBASE.Addr + index*HI_ATTRIB[n]_VBCTRL.Stride];  // Index mode
7.     else
8.        Fetch DWORDs from CPU and use them as the geometry data;   // Non-Index mode
9.        Transform DWORDs into floating point using HI_ATTRIB[n].Dt and send them to vertex shader.
10.    if (HI_ATTRIB[n].LastAttrib == 1) break;
11. end for
12. repeat step 2~11 (count) times. // the (count) value in step 1 is used.
```

In step 8 and 9, the number of used DWORDs is determined by HI_ATTRIB*n*.Dt (data type of transferred DWORD) and HI_ATTRIB*n*.NumComp (the number of components for each attribute). In step 8, the Host Interface does not recognize what the DWORDs really are. The Host Interface only fetches the required number of DWORDs from CPU. This is the reason why CPU must transfer indices and geometry data properly.

In step 9, HI_ATTRIB[n].Dt determines how the transferred DWORD is transformed into the floating point format. (Normalization can be performed.) Also, in the same step, the order of components of each attribute can be switched by SrcX~SrcW in each HI_ATTRIB*n*. SrcX~SrcW is initialized as 0 value mapping first transferred data to SrcX~SrcW. If (x, y, z, w) attribute is to be transferred in order, SrcW~SrcX must be set to the value 2'b11, 2'b10, 2'b01, 2'b00 respectively. This configuration is useful when color value is transferred in BGRA instead of RGBA. If the number of each component is not four, then 0.0, 0.0, and 1.0 are automatically appended. For example, if (x, y) data is transferred, then (FLOAT(x), FLOAT(y), 0.0, 1.0) is sent to the vertex shader.

Note that DWORDs transferred from CPU or in the Vertex Buffer for vertex attributes must be ***DWORD aligned***. If (8-bit x, 8-bit y, 8-bit z) is to be transferred, the last 8-bit data ([31:24] fields of the transferred DWORD) is ignored.

Note that HI_ATTRIB[n]_VBCTRL.Range must be set properly; the indices from CPU must be less than HI_ATTRIB[n]_VBCTRL.Range.

3.4 HOW TO SEND DWORDS FROM CPU TO THE HOST INTERFACE

There is a Host-FIFO in the Host Interface. CPU can transfer only DWORD data to the Host Interface. If CPU writes more data than the Host-FIFO can actually store, the Host Interface makes HREADY, one of AMBA bus signals, low. In this case, AMBA bus is granted to 3D-ACCELERATOR and any other IP on the same AMBA bus can not get a right to use the bus, which is not a desirable situation. The read-only HI_DWSPACE register is used to ease this situation. HI_DWSPACE holds the number of empty DWORD space in the Host-FIFO. Host-FIFO is in Host Interface and is able to store upto 32 DWORDs. ***Whenever CPU sends the count, indices, or the geometry data, CPU must get HI_DWSPACE value and transfer DWORDs as much as the value of HI_DWSPACE*.**

Generally speaking, HI_DWSPACE does not tell CPU the exact empty space because the clock signals fed into 3D-ACCELERATOR and AMBA bus can be different. (If clock signals are same, HI_DWSPACE has the exact value.) Hence, HI_DWSPACE is affected by both AMBA bus (writing DWORDs into the Host-FIFO) and the internal situation (fetching DWORDs from the Host-FIFO).

If the read value of HI_DWSPACE is less than the actual free space in the Host-FIFO, the writing operation ends without any problem. On the other hand, if the read value of HI_DWSPACE is more than the number of written DWORDs, the Host Interface makes HREADY signal low and extends the transfer. However, the difference between the read HI_DWSPACE value and the actual value is usually small.

After CPU reading HI_DWSPACE and transferring DWORDs as much as the read value, CPU can do the other job or process, or continue to send the other part of the geometry data repeating the same procedure. The CPU can use the interrupt scheme of 3D-ACCELERATOR. It depends wholy on the device driver.

Interrupts and Vertex Buffer are very useful schemes when a geometry is transferred. (See Section "HOW TO USE THE VERTEX BUFFER AS A TEMPORAL BUFFER USING INTERRRUPTS" for more information.)

3.5 THE TYPE OF INDEX TRANSFERRED FROM CPU

The IdxType field in HI_CONTROL controls how much index exists in a DWORD from CPU. If IdxType is unsigned int type, there is only one 32-bit index in a transferred DWORD. In the case of unsigned short type, two 16-bit indices are in a DWORD. In the case of unsigned byte type, four 8-bit indices are available.

The remained indices in a DWORD, when all indices are used, are ignored. For example, if three vertices with unsigned byte index type are transferred, a DWORD data is used for them. In this case, the last unsigned byte in the DWORD is ignored.

3.6 DATA TRANSFER TO THE VERTEX BUFFER

Before the contents of the Vertex Buffer are used, the geometry data must be resided in the Vertex Buffer. First, the 16-byte-aligned destination address in the Vertex Buffer is set to HI_VBADDR. And then, a serise of DWORD written into HI_VBDATA is stored into the Vertex Buffer (Burst writes are possible). The address in HI_VBADDR is automatically incremented by 16 (in bytes) whenever 4 DWORDs are written into HI_VBDATA. Therefore, the destination address does not need to be updated for every DWORD written into HI_VBDATA. Note that the number of DWORD written into HI_VBDATA must be the multiples of 4. Only when four DWORDs are transferred from CPU, thoes four DWORDs are stored into the Vertex Buffer. (If 3 DWORDs are transferred, those DWORDs are not stored into the Vertex Buffer wating another DWORD to be transferred.)

If the size of the geometry is not a multiples of 4, then send addtitional DWORD (usually 0x00000000) into the Vertex Buffer. The additional DWORDs for dword-aligned does not affect the value of HI_ATTRIB*n*_VBCT-RL.Range. The value of actual index range must be written in HI_ATTRIB*n*_VBCTRL.Range.

Note that NaN or infinite floating point or half-float value must not be written in the Vertex Buffer just like Non-Index Mode.

Also, note that the geometry data in the Vertex Buffer must be DWORD aligned. Refer to the "Attribute Control Register" in section "HOST INTERFACE SPECIAL REGISTERS."

3.7 HOW TO USE THE VERTEX BUFFER AS A TEMPORAL BUFFER USING INTERRUPTS

There are 32 DWORD space in Host-FIFO. If CPU sends a lot of DWORDs with HI_DWSPACE polling, CPU has to waste lots of cycles reading HI_DWSPACE without doing any other useful job until all the DWORDs are transferred. This is an undesirable situation.

Vertex buffer and interrupt scheme can be used in this situation. Vertex buffer is used for one-time used geometries in this situation: remind that Vertex Buffer usually stores geometry data which is supposed to be used several times for performance.

CPU sends a part of DWORDs for geometries into Vertex Buffer instead of Host-FIFO. After saving DWORDs into Vertex Buffer, CPU sets 3D-ACCELERATOR's interrupt scheme making the interrupt-unit send an interrupt to CPU when the values of GB_PIPESTATE for Host-FIFO and Host Interface become zero. At this time, CPU can do other valuable job, such as Operating-System or Sound related processes, waiting for an interrupt from 3D-ACCELERATOR. If an interrupt from 3D-ACCELERATOR is occurred and CPU is allowed to handle the geometry-sending process, CPU continues to send the rest of geometries using the same procedure.

You can make an interrupt occur when all the 3D-ACCELERATOR pipeline stages become empty. You can make your own decision when to make an interrupt occur. There is one thing you should pay attention to in this case: you must send the exact number of vertices. For example, when the Primitive-Engine, which is the next block to the Vertex Cache, is supposed to get a TRIANGLE data, the number of vertices must be the multiples of 3. If (3n+1) vertices are sent and interrupt unit is waiting for the HI_PIPESTATE to be zero, an interrupt from 3D-ACCELERATOR never occurs under this situation because the Primitive-Engine's value of HI_PIPESTATE is 1 waiting for another two vertices. (However, if the Primitive-Engine is set to receive triangle strip data, the number of transferred vertices is not important.)

If Vertex Buffer is used in this way, Vertex Cache is suggested to be disabled because all DWORDs (or Indices) are used only one time; Vertex Cache does not have Hit-Case.

3.8 VERTEX CACHE CONTROL

The EnVC and NumOutAttrib fields in HI_CONTROL controls the way how the vertex cache works. If EnVC field is 0, then the vertex cache is disabled. The NumOutAttrib field stores the number of output attributes from the vertex shader. The number of output attributes determined by NumOutAttrib field is transferred to the primitive engine.

Note that when HI_CONTROL is wrtten by CPU, the Post Vertex Cache is cleared (or initialized) automatically. When you send a series of indices for a geometry data, you send another different geometry. In this case, the index for the previous geometry data which is remained in Vertex Cache can be hit when the index of the new geometry data is sent. Hence, when you send multiple geometry data using index mode, you must clear the content of Vertex Cache between geometries. Vertex Cache is cleared automatically when HI_CONTROL is written by CPU. Although the HI_CONTROL value is not changed, HI_CONTROL can be rewritten with the same value to clear the contents of Vertex Cache.

3.9 HOST INTERFACE SPECIAL REGISTERS

NOTE : Host Interface is abbribiated to HI.

3.9.1 Host-FIFO'S Free Dword Space Register (HI_DWSPACE, R, Address = 0X0000_8000)

HI_DWSPACE	Bit	Description	Reset Value
VAL	[7:0]	The number of empty slots	0x20

3.9.2 Host-FIFO Entry Port Register (HI_DWENTRY, W, Address = 0X0000_C000~0X0000_DFFF)

HI_DWENTRY	Bit	Description	Reset Value
DATA	[31:0]	The number of vertices, indices, and the geometry data is transferred into this register.	-

3.9.3 Host Interface Control Register (HI_CONTROL, R/W, Address = 0X0000_8008)

HI_CONTROL	Bit	Description	Reset Value
EnVB	[31]	Enable Vertex Buffer	0b
Reserved	[30:26]	Reserved	0
IdxType	[25:24]	Transferred index type 00b = unsigned int 01b = unsigned short 10b = reserved 11b = unsigned byte	00b
Reserved	[23:17]	Reserved	0
AutoInc	[16]	Auto increment mode	1b
Reserved	[15:5]	Reserved	0
EnVC	[4]	Enable vertex cache	0b
NumOutAttrib	[3:0]	The number of vertex shader output attributes. When point-sprite is used, this number must be (the number of shader outputs + 1). See the Raster Engine chapter for more information.	0000b

3.9.4 Index Offset Register (HI_IDXOFFSET, R/W, Address = 0X0000_800C)

HI_IDXOFFSET	Bit	Description	Reset Value
VAL	[31:0]	Index offset value When an index is auto-incremented, VAL is added to the index. The first transferred index from CPU is used as is. Therefore, the used indices are: index, index+VAL, index+2*VAL, etc. When indices transferred from CPU are used, VAL is added to each transferred indices. Let's say index0, index1, index2, etc are sent from CPU. Then, the used indices in the HI are: index0+VAL, index1+VAL, index2+VAL, etc. If the recalculated indices are within HI_ATTRIB*n*_VBCTRL.Range, the geometry data in Vertex Buffer is used.	0x00000001

3.9.5 Vertex Buffer Address Register (HI_VBADDR, R/W, Address = 0X0000_8010)

HI_VBADDR	Bit	Description	Reset Value
VAL	[31:0]	Start address of attribute to copy the geometry data.	0x0

3.9.6 Vertex Buffer Entry Port Register (HI_VBDATA, R/W, Address = 0X0000_E000)

3.9.7 Vertex Buffer Entry Port Register (HI_VBDATA, R/W, Address = 0X0000_FFFF)

VBD	Bit	Description	Reset Value
DATA	[31:0]	Data input port to Vertex Buffer. This register should be written in multiples of 4. The start address is incremented automatically.	0x0

3.9.8 Attribute Control Registers (HI_ATTRIB0~HI_ATTRIB9)

Attribute control Registers (HI_ATTRIB0~HI_ATTRIB9) represents how the DWORD data from CPU or in the vertex buffer is interpreted.

If vertex data type is byte, unsigned byte, normalized byte, or normalized unsigned byte, a DWORD transferred from CPU to Host Interface must contain four components. In the last DWORD, the unused 8-bit data is ignored.

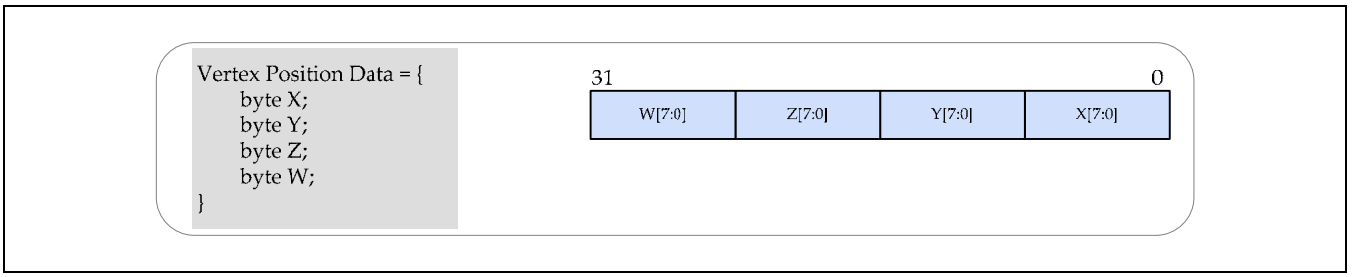


Figure 9.6- 10 Memory Layouts for Byte-type Vertex Attributes in a DWORD from CPU or in the Vertex Buffer.

In the above example, (8-bit x, 8-bit y, 8-bit z, 8-bit w) attribute can be transferred with one DWORD.If vertex data type is short, unsigned short, normalized short, or normalized unsigned short, two attributes can be reside in a DWORD. Therefore, two DWORDs are required for (16-bit x, 16-bit y, 16-bit z, and 16-bit w).

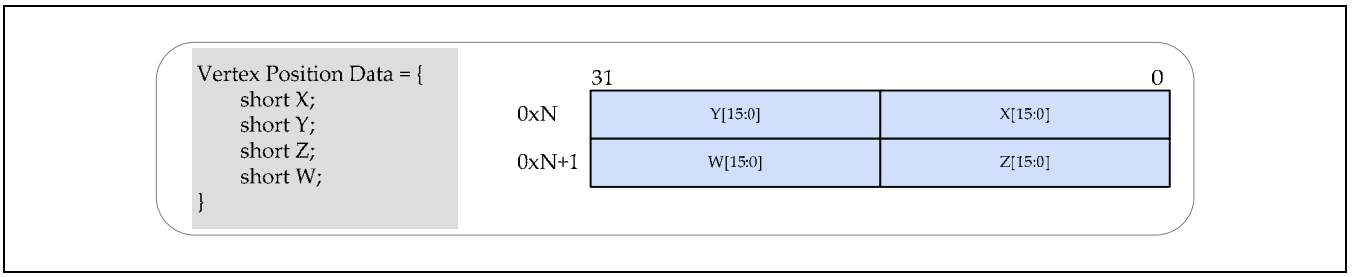


Figure 9.6- 11 Memory Layouts for Short-type Vertex Attribute in DWORDs from CPU or in the Vertex Buffer.

Note that when DWORDs are stored in the Vertex Buffer, the above rule is also applied.

- Input attribute 0 Control Register, R/W, Address = 0x0000_8040
- Input attribute 1 Control Register, R/W, Address = 0x0000_8044
- Input attribute 2 Control Register, R/W, Address = 0x0000_8048
- Input attribute 3 Control Register, R/W, Address = 0x0000_804C
- Input attribute 4 Control Register, R/W, Address = 0x0000_8050
- Input attribute 5 Control Register, R/W, Address = 0x0000_8054
- Input attribute 6 Control Register, R/W, Address = 0x0000_8058
- Input attribute 7 Control Register, R/W, Address = 0x0000_805C
- Input attribute 8 Control Register, R/W, Address = 0x0000_8060
- Input attribute 9 Control Register, R/W, Address = 0x0000_8064

HI_ATTRIBn	Bit	Description	Reset Value
LastAttr	[31]	0b = indicates the ATTRIB*n* is used 1b = indicates the ATTRIB*n* is the last attribute After reset, all the "last" value is zero, which means one attribute, is used by default. Ex) HI_ATTRIB0[31] = 0, HI_ATTRIB1[31] = 0, HI_ATTRIB2[31] = 1, HI_ATTRIB3~9[31] = don't care → HI_ATTRIB0 ~ 2 are used.	1b
Reserved	[30:16]	Reserved	0
Dt	[15:12]	Each component of attribute n is transferred as Bit / Data Type / Range 0000 / byte / -127 ~ 128 0001 / short / -23768 ~ 32767 0010 / int / -2147483648 ~ 2147483647 0011 / fixed / -32768 ~ 32768 0100 / unsigned byte / 0 ~ 255 0101 / unsigned sort / 0 ~ 65535 0110 / unsigned int / 0 ~ 4294967295 0111 / float / IEEE 754 single precision 1000 / normalized byte / -1.0f ~ 1.0f 1001 / normalized short / -1.0f ~ 1.0f 1010 / normalized int / -1.0f ~ 1.0f 1011 / normalized fixed / 0.0f ~ 1.0f 1100 / normalized unsigned byte / 0.0f ~ 1.0f 1101 / normalized unsigned short / 0.0f ~ 1.0f 1110 / normalized unsigned int / 0.0f ~ 1.0f 1111 / Half-float (h) / s/5/10 format When floating-point or half-float data type is used, NaN or infinite number must not be used.	0
Reserved	[11:10]	Reserved	0
NumComp	[9:8]	Number of components 00b = only one component is transferred (a, b, c, d) = (1^{st}, 0, 0, 1) 01b = two component are transferred (a, b, c, d) = (1^{st}, 2^{nd}, 0, 1) 10b = three component are transferred (a, b, c, d) = (1^{st}, 2^{nd}, 3^{rd}, 1) 11b = four component are transferred (a, b, c, d) = (1^{st}, 2^{nd}, 3^{rd}, 4^{th})	00b

HI_ATTRIBn	Bit	Description	Reset Value
		(a, b, c, d) is used to select (X, Y, Z, W) generally	
SrcW	[7:6]	Select W component 00b = select 'a' component as W 01b = select 'b' component as W 10b = select 'c' component as W 11b = select 'd' component as W NOTE: a~d is defined in NumComp field.	11b
SrcZ	[5:4]	Select Z component 00b = select 'a' component as Z 01b = select 'b' component as Z 10b = select 'c' component as Z 11b = select 'd' component as Z NOTE: a~d is defined in NumComp field.	10b
SrcY	[3:2]	Select Y component 00b = select 'a' component as Y 01b = select 'b' component as Y 10b = select 'c' component as Y 11b = select 'd' component as Y NOTE : a~d are defined in NumComp field	01b
SrcX	[1:0]	Select X component 00b = select 'a' component as X 01b = select 'b' component as X 10b = select 'c' component as X 11b = select 'd' component as X NOTE : a~d are defined in NumComp field	00b

3.9.9 Vertex Buffer Control Register (HI_ATTRIB0_VBCTRL ~ HI_ATTRIB9_VBCTRL)

HI_ATTRIBn_VBCTRL.Stride represents the number of bytes to the next input attribute in the Vertex Buffer.
HI_ATTRIBn_VBCTRL.num represnets how many the input attributes are in the Vertex Buffer.

- Vertex Buffer Control of Input Attribute 0, R/W, Address = 0x0000_8080
- Vertex Buffer Control of Input Attribute 1, R/W, Address = 0x0000_8084
- Vertex Buffer Control of Input Attribute 2, R/W, Address = 0x0000_8088
- Vertex Buffer Control of Input Attribute 3, R/W, Address = 0x0000_808C
- Vertex Buffer Control of Input Attribute 4, R/W, Address = 0x0000_8090
- Vertex Buffer Control of Input Attribute 5, R/W, Address = 0x0000_8094
- Vertex Buffer Control of Input Attribute 6, R/W, Address = 0x0000_8098
- Vertex Buffer Control of Input Attribute 7, R/W, Address = 0x0000_809C
- Vertex Buffer Control of Input Attribute 8, R/W, Address = 0x0000_80A0
- Vertex Buffer Control of Input Attribute 9, R/W, Address = 0x0000_80A4

HI_ATTRIB*n*_VBCTRL	Bit	Description	Reset Value
Stride	[31:24]	Next attribute position in bytes	0x0
Reserved	[31:16]	Reserved	0
Range	[15:0]	Valid index range of index in Vertex Buffer. The index number from CPU must be less than this value.	0x0

3.9.10 Vertex Buffer Base Address Register (HI_ATTR0_VBBASE~HI_ATTR9_VBBASE)

- Vertex Buffer Base Address Of Input Attribute 0, R/W, Address = 0x0000_80C0
- Vertex Buffer Base Address Of Input Attribute 1, R/W, Address = 0x0000_80C4
- Vertex Buffer Base Address Of Input Attribute 2, R/W, Address = 0x0000_80C8
- Vertex Buffer Base Address Of Input Attribute 3, R/W, Address = 0x0000_80CC
- Vertex Buffer Base Address Of Input Attribute 4, R/W, Address = 0x0000_80D0
- Vertex Buffer Base Address Of Input Attribute 5, R/W, Address = 0x0000_80D4
- Vertex Buffer Base Address Of Input Attribute 6, R/W, Address = 0x0000_80D4
- Vertex Buffer Base Address Of Input Attribute 7, R/W, Address = 0x0000_80D8
- Vertex Buffer Base Address Of Input Attribute 8, R/W, Address = 0x0000_80E0
- Vertex Buffer Base Address Of Input Attribute 9, R/W, Address = 0x0000_80E4

HI_ATTRIB*n*_VBBASE	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
Addr	[15:0]	Base address of input attribute n in Vertex Buffer	0x0

4 VERTEX SHADER

4.1 OVERVIEW

The vertex shader is the 3D graphics specific processor that processes vertices instead of traditional fixed-function graphics pipeline. The vertex shader can enable the user-defined special effects more than the common transform and lighting. This vertex shader supports the shader model 3.0 including the features of vertex texture and various flow controls. Refer DirectX reference, OpenGL 2.0 specification and OpenGL shading language specification for more information.

4.2 INITIAL OPERATION

Vertex shader program is composed of instruction sequences, constant floating-point values for the arithmetic operations, constant integer and boolean values for the flow control. These should be stored in the register or memory region before executing the program. Vertex shader starts automatically when the host writes all attributes for a vertex.

4.3 VERTEX SHADER SPECIAL REGISTERS

The shader instruction and the constant values should be stored in the special register region for the vertex shader operation. These registers can be updated via the host interface.

4.3.1 Instruction Memory (VS_INSTMEM)

- Instruction Memory of The Vertex Shader, R/W, Address = 0x0001_0000
- Instruction Memory of The Vertex Shader, R/W, Address = 0x0001_1FFF

The instruction memory has 512 slots and 1 slot is composed of 4 words.

4.3.2 Constant Float Register (VS_CFLOAT)

- Constant Float Register of The Vertex Shader, R/W, Address = 0x0001_4000 ~ 0x0001_4FFF

The constant floating-point numbers can be stored in the constant float register for calculating in the program. The constant float register has 256 entries. Each entry is composed of 4 channels, x, y, z, w as below. Each channel is 32-bit word and has IEEE single precision floating-point format. (De-normalized number is not supported.) Initial value of this register is un-known because it is stored internal sram. User has to initialize this register before use it.

4.3.3 Constant Float W Component Value (WORD 3, R/W, Address = 0X0001_4XXC, 00 <= XX <= FF)

WORD3	Bit	Description	Reset Value
W	[127:96]	Constant float W component value	0xXXXXXXXX

4.3.4 Constant Float Z Component Value (WORD 2, R/W, Address = 0X0001_4XX8, 00 <= XX <= FF)

WORD2	Bit	Description	Reset Value
Z	[95:64]	Constant float Z component value	0xXXXXXXXX

4.3.5 Constant Float Y Component Value (WORD 1, R/W, Address = 0X0001_4XX4, 00 <= XX <= FF)

WORD1	Bit	Description	Reset Value
Y	[63:32]	Constant float Y component value	0xXXXXXXXX

4.3.6 Constant Float X Component Value (WORD 0, R/W, Address = 0X0001_4XX0, 00 <= XX <= FF)

WORD0	Bit	Description	Reset Value
X	[31:0]	Constant float X component value	0xXXXXXXXX

※ IEEE single precision floating-point format

	Bit	Description	Reset Value
S	[31]	Sign bit	XXXXXXh
E	[30:23]	Biased exponent	XXh
F	[22:0]	Fraction	

Floating-point value V is Not a Number (NaN) if E=255 and F is nonzero

- -Infinity (-INF) if E=255 and F=0 and S=1
- +Infinity (+INF) if E=255 and F=0 and S=0

$(-1)^S * 2^{(E-127)} * (1.F)$ if 0<E<255 where "1.F" is intended to represent the binary number created by prefixing F with an implicit leading 1 and a binary point

De-normalized number $(-1)^S * 2^{(-126)} * (0.F)$ if E=0 and F is nonzero (Not supported)

- -0 if E=0 and F=0 and S=1
- +0 if E=0 and F=0 and S=0

4.3.7 Constant Integer Register (VS_CINT, R/W, Address = 0X0001_8000 ~ 0X0001_803F)

The constant integer values can be stored in the constant integer register. The constant integer value is only used for the flow control - that is the iteration count for loop or the index of relative addressing. The constant integer register has 16 entries, and each entry is composed of 4-channel 8-bit unsigned integer value. User has to initialize this register before use it.

4.3.8 Word (0X0001_80XX, XX IS Word Address Ranged FROM 00 TO 3C))

WORD	Bit	Description	Reset Value
W	[31:24]	Constant integer W component value	0xX
Z	[23:16]	Constant integer Z component value	0xX
Y	[15:8]	Constant integer Y component value	0xX
X	[7:0]	Constant integer X component value	0xX

4.3.9 Constant BOOL Register (VS_CBOOL, R/W, Address = 0X0001_8400)

The constant bool value can be stored in the constant bool register. The constant bool value is only used for the static flow control. It is referenced in the conditional branch instruction. The constant bool register is 16-bit boolean register. The register number corresponds to the each bit position. 'TRUE' is represented by '1' and 'FALSE' by '0'. User has to initialize this register before use it.

CBOOL	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0xXXXX
REG15	[15]	Constant Bool register 15	X
REG14	[14]	Constant Bool register 14	X
REG13	[13]	Constant Bool register 13	X
REG12	[12]	Constant Bool register 12	X
REG11	[11]	Constant Bool register 11	X
REG10	[10]	Constant Bool register 10	X
REG9	[9]	Constant Bool register 9	X
REG8	[8]	Constant Bool register 8	X
REG7	[7]	Constant Bool register 7	X
REG6	[6]	Constant Bool register 6	X
REG5	[5]	Constant Bool register 5	X
REG4	[4]	Constant Bool register 4	X
REG3	[3]	Constant Bool register 3	X
REG2	[2]	Constant Bool register 2	X
REG1	[1]	Constant Bool register 1	X
REG0	[0]	Constant Bool register 0	X

4.3.10 Configuration Register for Vertex Shader (FGVS) (VS_CONFIG, W, ADDRESS = 0X0001_C800)

Global register contains various configurations and environment setting for global operation.

VS_Config	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
ClrStatus	[1]	When this bit is set to '1', the all values of VS_Status register are cleared. This bit is automatically cleared to '0' after clear operation.	0b
CopyPC	[0]	When this bit is set to '1', VS_PCRange register value is actually copied to the vertex shader inside. This bit is cleared to '0' after copy automatically. Without copy command, the value of VS_PCRange is not used, and the previous values are used for program start and end address.	0b

4.3.11 Configuration Register for Vertex Shader (FGVS) (VS_ STATUS, R, ADDRESS = 0X0001_C804)

Global register contains various configurations and environment setting for global operation.

VS_Status	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
ClrStatus	[1]	When this bit is set to '1', the all values of VS_Status register are cleared. This bit is automatically cleared to '0' after clear operation.	0b
CopyPC	[0]	When this bit is set to '1', VS_PCRange register value is actually copied to the vertex shader inside. This bit is cleared to '0' after copy automatically. Without copy command, the value of VS_PCRange is not used, and the previous values are used for program start and end address.	0b

4.3.12 Configuration Register for Vertex Shader (FGVS) (VS_ PCRANGE, R/W, ADDRESS = 0X0002_0000)

VS_PCRange	Bit	Description	Reset Value
IgnorePCEnd	[31]	When this bit is set to 1, PCEnd value is ignored and the only program counter stack empty condition is used for program termination.	0
Reserved	[30:25]	Reserved	0
PCEnd	[24:16]	When the program counter is reached at the value of PCEnd, the shader program is terminated after execution of the instruction at this register. This method can save instruction slot and number of instruction count to execute. The other way to terminate vertex shader program : Vertex shader program can be terminated by the extra "ret" instruction which makes program counter stack empty condition. Usually, "call" and "ret" instruction works a pair. But the intentional unmatched "ret" makes vertex shader termination condition. By this exception, vertex shader program can be terminated.	0x1FF
Reserved	[15:9]	Reserved	0
PCStart	[8:0]	When the vertex shader start operation, the first instruction, which is stored at PCStart, is fetched from instruction memory. This register value should be copied to vertex shader program counter by PCCopy register before shader starts.	0x00

4.3.13 Configuration Register for Vertex Shader (FGVS) (VS_ ATTRIBUTENUM, R/W, ADDRESS = 0X0002_0004)

VS_AttributeNum	Bit	Description	Reset Value
Reserved	[31:20]	Reserved	0
OutAttributeNum	[19:16]	The number of attributes for the vertex shader output. This should be match the number of output registers in current shader program.	0x1
Reserved	[15:4]	Reserved	0
InAttributeNum	[3:0]	The number of attributes for the vertex shader input. This should be matching the number of input registers in current shader program.	0x1

4.3.14 Index Register for Input Attribute (FGVS)

Generally, the register number of the input register of the vertex shader matches with the order of input attributes from the host. This relationship can be remapped by the input attribute index registers for flexibility. The N-th input attribute from host is actually read from the position indicated by the index looked up from the AttribN corresponding to the register number of input register in the shader program.

- VS_InAttrIndex0, W, Address = 0x0002_0008
- VS_InAttrIndex1, W, Address = 0x0002_000C
- VS_InAttrIndex2, W, Address = 0x0002_0010

VS_InAttrIndex0	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
Attrib3	[27:24]	Index of input attribute 3	0x3
Reserved	[23:20]	Reserved	0
Attrib 2	[19:16]	Index of input attribute 2	0x2
Reserved	[15:12]	Reserved	0
Attrib 1	[11:8]	Index of input attribute 1	0x1
Reserved	[7:4]	Reserved	0
Attrib 0	[3:0]	Index of input attribute 0	0x0

VS_InAttrIndex1	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
Attrib7	[27:24]	Index of input attribute 7	0x7
Reserved	[23:20]	Reserved	0
Attrib6	[19:16]	Index of input attribute 6	0x6
Reserved	[15:12]	Reserved	0
Attrib5	[11:8]	Index of input attribute 5	0x5
Reserved	[7:4]	Reserved	0
Attrib4	[3:0]	Index of input attribute 4	0x4
VS_InAttrIndex2	**Bit**	**Description**	**Reset Value**
Reserved	[31:28]	Reserved	0
Attrib11	[27:24]	Index of input attribute 11	0xB
Reserved	[23:20]	Reserved	0
Attrib10	[19:16]	Index of input attribute 10	0xA
Reserved	[15:12]	Reserved	0
Attrib9	[11:8]	Index of input attribute 9	0x9
Reserved	[7:4]	Reserved	0
Attrib8	[3:0]	Index of input attribute 8	0x8

4.3.15 Index Register for Output Attribute (VS_OUTATTRINDEX)

Generally, the register number of input register for pixel shader corresponds to those of output register for vertex shader. This relationship can be remapped by the output attributes index registers. The N-th output attribute is actually written to the position indicated by the index looked up from the AttribN corresponding to the register number of output register in the shader program. The input register 0 gets the value from the output register 1 becuase the output register 0 of the vertex shader is specially dedicated for the position. So, the output registers 1-8 of the vertex shader correspond to the input registers 0-7 of the pixel shader.

- VS_OutAttrIndex0, W, Address = 0x0002_0014
- VS_OutAttrIndex1, W, Address = 0x0002_0018
- VS_OutAttrIndex2, W, Address = 0x0002_001C

VS_OutAttrIndex0	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
Attrib3	[27:24]	Index of output attribute 3	0x3
Reserved	[23:20]	Reserved	0
Attrib2	[19:16]	Index of output attribute 2	0x2
Reserved	[15:12]	Reserved	0
Attrib1	[11:8]	Index of output attribute 1	0x1
Reserved	[7:4]	Reserved	0
Attrib0	[3:0]	Index of output attribute 0	0x0

VS_OutAttrIndex1	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
Attrib7	[27:24]	Index of output attribute 7	0x7
Reserved	[23:20]	Reserved	0
Attrib6	[19:16]	Index of output attribute 6	0x6
Reserved	[15:12]	Reserved	0
Attrib5	[11:8]	Index of output attribute 5	0x5
Reserved	[7:4]	Reserved	0
Attrib4	[3:0]	Index of output attribute 4	0x4

VS_OutAttrIndex2	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
Attrib11	[27:24]	Index of output attribute 11	0xB
Reserved	[23:20]	Reserved	0
Attrib10	[19:16]	Index of output attribute 10	0xA
Reserved	[15:12]	Reserved	0
Attrib9	[11:8]	Index of output attribute 9	0x9

VS_OutAttrIndex2	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0
Attrib8	[3:0]	Index of output attribute 8	0x8

5 PRIMITIVE ENGINES

5.1 OVERVIEW

Primitive engine is a hardware-wired block that is processing a series of operations including primitive assembly, flat shading, view frustum clipping, perspective division and viewport mapping. View frustum clipping is divided into near/far plane clipping operations and left/right/top/bottom plane clipping operations. In primitive engine, only near/far plane clipping is processed, except when the w-coordinates of clipped vertex is zero. Using the window clipping operation, other clipping operations are processed by triangle setup engine and rasterizer. When the w-coordinates of a vertex is zero, the primitive engine perform other clipping operations to avoid the w-coordinates of the output vertex to be zero.

The following figure represents the conceptual functional level block diagram of primitive engine to illustrate operations of the primitive engine.

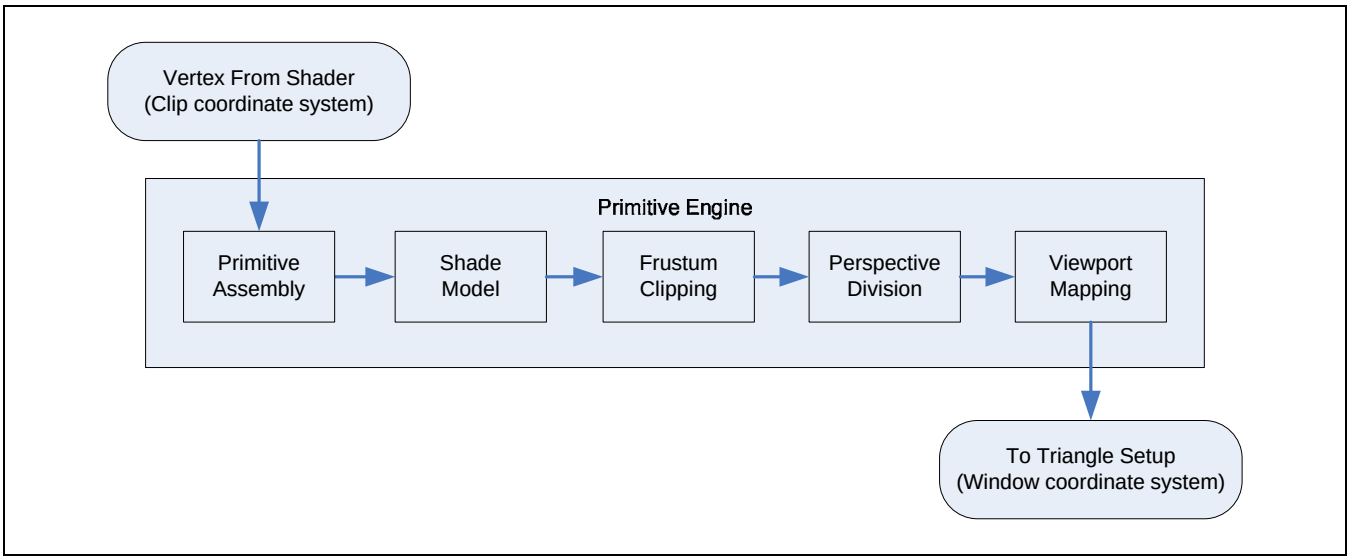


Figure 9.6- 12 PE Pipeline

5.2 PRIMITIVE ENGINE SPECIAL REGISTERS

There are two kinds of special registers in primitive engine. One is for vertex information such as primitive types, the number of associated data, and the shade model. And the other is for viewport parameters.

5.3 VERTEX CONTEXT REGISTER

Vertex context register is divided into three kinds of information for processing an input vertex in the primitive engine. The bit fields between 29 and 19 represent the primitive type to render. In the primitive engine, triangles, triangle fan, triangle strip, line, line loop, line strip, point, and point sprite are supported. The other types such as polygon, quad, and quad strip are not supported but the reserved bits for the types exist in the bit fields. Each bit field for the primitive types is orthogonal to another. This implies that the bit for point sprite must be set with resetting the bit for points when using point sprite mode.

The bit fields between 13 and 10 represent the number of associated data with the vertex in encoded format. MSB is bit 13 and the LSB is bit 10. The number of associated data is counted assuming that the bit width of associated data is 128. If the associated data of a vertex are five 32-bit values, then the number of associated data will be set as 2. When you want to use vertex shader program point size mode, the number of associated data must be increased by 1. As described above, it is related with the varying variable of vertex shader and vertex shader program point size mode, so it ranges from 0 to 9. The number of associated data, VSOUT, can be expressed as follows.

$$VSOUT = \begin{cases} \left\lfloor \frac{(\text{the number of varying variables})}{4} \right\rfloor + 1 & , \text{when vertex program point size mode} \\ \left\lfloor \frac{(\text{the number of varying variables})}{4} \right\rfloor & , \text{otherwise} \end{cases}$$

Figure 9.6- 13 Number of Associated Data for Vertex Shader Output in PE Stage

The bit fields between 9 and 0 represent the shade model information. The bit field 9 is the master flag that represents where shade model is smooth or flat. The bit field from 0 to 8 is associated with the vertex shader output slot. When the master flag is 0, the bit fields associated with vertex shader output slot are ignored. For example, to use vertex shader output slot0 as flat color channel, bit field 9 and bit field 0 must be set as 1.

The bit field 31 and 30 are used internally, so don't touch these bit fields.

Note. In 3D-ACCELERATOR, the vertex shader supports the number of input attributes up to 8 and that of varying variables up to 32.

5.3.1 Reserved For Internal Usage (PE_VERTEX_CONTEXT, R/W, Address = 0X0003_0000)

VCTX	Bit	Description	Reset Value
INTUSE	[31:30]	Reserved for internal usage. Don't touch this bit field.	0b
POLYGON	[29]	Reserved for polygon primitive type	0b
QUADS	[28]	Reserved for quads primitive type	0b
QUADSTRIP	[27]	Reserved for quad strip primitive type	0b
TRIANGLES	[26]	1b = triangles primitive type 0b = not triangles primitive type	0b
TRIANGLEFAN	[25]	1b = triangle fan primitive type 0b = not triangle fan primitive type	0b
TRIANGLESTRIP	[24]	1b = triangle strip primitive type 0b = not triangle strip primitive type	0b
LINES	[23]	1b = lines primitive type 0b = not lines primitive type	0b
LINELOOP	[22]	1b = line loop primitive type 0b = not line loop primitive type	0b
LINESTRIP	[21]	1b = line strip primitive type 0b = not line strip primitive type	0b

VCTX	Bit	Description	Reset Value
POINTS	[20]	1b = points primitive type 0b = not points primitive type	0b
POINTSPRITE	[19]	1b = point sprite primitive type 0b = not point sprite primitive type	0b
VERTEXPROGRAM POINTSIZE	[18]	1b = vertex shader program point size mode 0b = not vertex shader program point size mode	0b
Reserved	[17:14]	Reserved	0
VSOUT[3:0]	[13:10]	Excluding position, the number of output that vertex shader uses.	0x0
FLAT_SHADE	[9]	1b = using flat shade model, 0b = using smooth shade model	0b
FLAT_MODEL8	[8]	1b = vertex shader output8 is use flat shade model, 0b = vertex shader output8 using smooth shade model.	0b
FLAT_MODEL7	[7]	1b = vertex shader output7 is use flat shade model, 0b = vertex shader output7 using smooth shade model	0b
FLAT_MODEL6	[6]	1b = vertex shader output6 is use flat shade model, 0b = vertex shader output6 using smooth shade model	0b
FLAT_MODEL5	[5]	1b = vertex shader output5 is use flat shade model, 0b = vertex shader output5 using smooth shade model	0b
FLAT_MODEL4	[4]	1b = vertex shader output4 is use flat shade model, 0b = vertex shader output4 using smooth shade model	0b
FLAT_MODEL3	[3]	1b = vertex shader output3 is use flat shade model, 0b = vertex shader output3 using smooth shade model	0b
FLAT_MODEL2	[2]	1b = vertex shader output2 is use flat shade model, 0b = vertex shader output2 using smooth shade model	0b
FLAT_MODEL1	[1]	1b = vertex shader output1 is use flat shade model, 0b = vertex shader output1 using smooth shade model	0b
FLAT_MODEL0	[0]	1b = vertex shader output0 is use flat shade model, 0b = vertex shader output0 using smooth shade model	0b

5.3.2 Viewport Parameter Registers

The viewport transformation is determined by the viewport's width and height in pixels, p_x and p_y, respectively, and its center (o_x, o_y) also in pixels. The vertex's window coordinates, $\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix}$, are given by

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{pmatrix} (p_x/2)x_d + o_x \\ (p_y/2)y_d + o_y \\ [(f-n)/2]z_d + (n+f)/2 \end{pmatrix}$$

Figure 9.6- 14 View Port Transform

The factor and offset applied to z_d encoded by near depth range, n and far depth range, f. The center of viewport, (o_x, o_y) can be expressed as ($x_0+p_x/2$, $y_0+p_y/2$), assuming that the origin of viewport is (x_0, y_0). In 3D-ACCELERATOR, y-axes flipped window coordinates system is used. To generate y-axes flipped window coordinates system, we simply replace y-axes related equations, $y_w=(p_y/2)y_d+o_y$ and $o_y=y_0+p_y/2$ with $y_w=(-p_y/2)y_d+o_y$ and $o_y=$*(window height)*$-y_0-p_y/2$, respectively.

- PE_VIEWPORT_OX, R/W, Address = 0x0003_0004
- PE_VIEWPORT_OY, R/W, Address = 0x0003_0008
- PE_VIEWPORT_HALF_PX, R/W, Address = 0x0003_000C
- PE_VIEWPORT_HALF_PY, R/W, Address = 0x0003_0010
- PE_DEPTHRANGE_HALF_F_SUB_N, R/W, Address = 0x0003_0014
- PE_DEPTHRANGE_HALF_F_ADD_N, R/W, Address = 0x0003_0018

The detailed descriptions are using the following notation.

- p_x : the width of viewport in terms of pixel
- p_y : the height of viewport in terms of pixel
- x_0 : the y-coordiante of the viewport origin in widow coordinate system
- y_0 : the y-coordiante of the viewport origin in widow coordinate system
- n : near value of the depth range
- f : far value of the depth range
- H : the height of the window in terms of pixel

	Bit	Description	Reset Value
PE_VIEWPORT_OX	[31:0]	The x-coordinate of viewport center $x_0 + \frac{p_x}{2}$	0xX

	Bit	Description	Reset Value
PE_VIEWPORT_OY	[31:0]	The y-coordinate of viewport center $y_0 + \frac{p_y}{2}$ If you want to generate y-flipped window coordinates, set this SFR as follows. $(H - y_0) - \frac{p_y}{2}$	0xX

	Bit	Description	Reset Value
PE_VIEWPORT_HALF_PX	[31:0]	The half value of viewport width $\frac{p_x}{2}$	0xX

	Bit	Description	Reset Value
PE_VIEWPORT_HALF_PY	[31:0]	Half of viewport height $\frac{p_y}{2}$ If you want to generate y-flipped window coordinates, set this SFR as follows. $-\frac{p_y}{2}$	0xX

	Bit	Description	Reset Value
PE_DEPTHRANGE_HALF_F_SUB_N	[31:0]	The half value of subtract depth range far from near $\frac{f-n}{2} \quad (n,\ f \in [0,1])$	0x3F000000

	Bit	Description	Reset Value
PE_DEPTHRANGE_HALF_F_ADD_N	[31:0]	The half value of add depth range far to near $\frac{f+n}{2} \quad (n,\ f \in [0,1])$	0x3F000000

6 RASTER ENGINE

6.1 OVERVIEW

Raster engine include triangle setup engine and rasterizer.

6.1.1 Triangle Setup Engine Overview

- Primitive y-value bounding
- Back face culling
- Edge interpolation
- Triangle gradients calculation
- Depth offset calculation

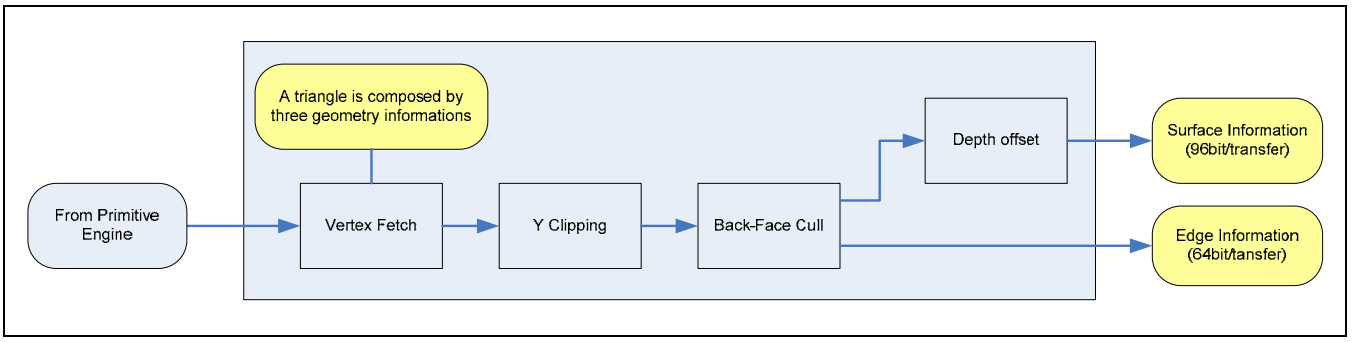


Figure 9.6- 15 Triangle Setup Stage Overview

6.1.2 Rasterizer Overview

- Triangle, Line, Point, and Point Sprite are supported. (Aliased only)
- 2-fragments are generated concurrently by a 2x1 stamp.
- Fragment position value calculation. (Screen coordinates[x, y], depth [z, iw])
- Fragment value calculation. (Color / fog / normal / texture coordinates)
- Coefficients calculation for LOD selection.

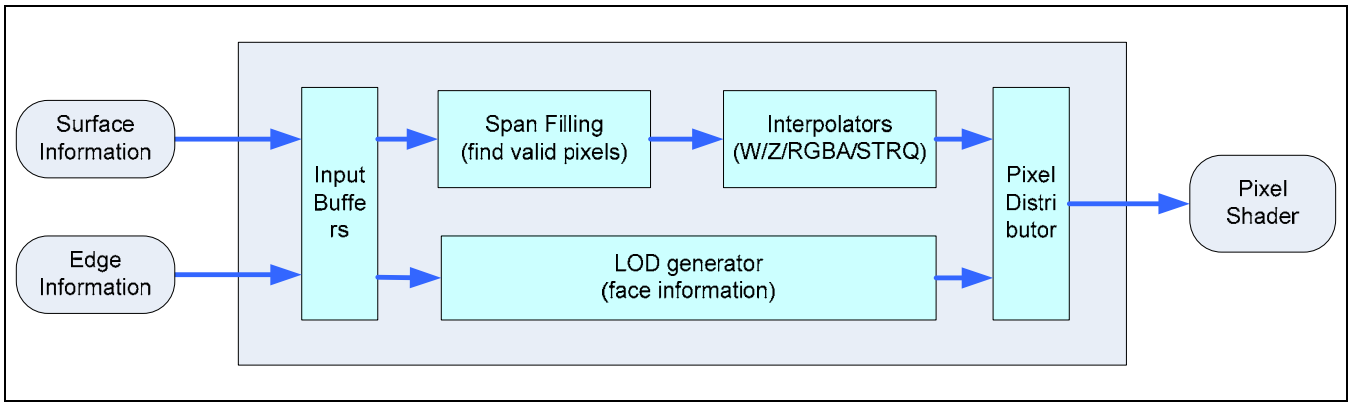


Figure 9.6- 16 Rasterizer Pipeline Overview

6.2 INITIAL OPERATION

The details of raster engine operation:

Sampling position control register: This register indicates whether pixel is sampled at the center or corner of the pixel. So, it is enough to set this value one time initially.

Depth offset control register: It is necessary to set this register and the values of factor and units for an object that uses depth offset. This register value can be changed only in context switching time.

Back face culling control register: To process back face cull, cull face register, front face register and enable register all should be set. This register value can be changed only in context switching time.

LOD control register: It is necessary to set this register if pixel shader uses DDX/DDY/LOD. This register value should be set to 0b when pixel shader doesn't use DDX/DDY/LOD. This register value can be changed only in context switching time.

Window X/Y-Clipping region control register: To clip window region given by viewing frustum.

Coord Replace control register: It is used for only Point Sprite rendering. For PointSprite, texture coordinate coefficients generated in TSE are attached behind of the last attribute. Therefore, the Coord Replace control bit should be set correctly in the attribute number corresponding to the attached texture coordinate coefficients.

6.3 RASTER ENGINE SPECIAL REGISTERS

6.3.1 Sampling Position Register (RA_PIXSAMP, R/W, Address = 0X0003_8000)

RA_PixSamp	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	
PixCornerSamp	[0]	Select sample position used when fetching pixels for texture and shading 0b = (0.5, 0.5) center position 1b = (0, 0) left-top corner position	0b

6.3.2 Depth Offset Registers (FGRA)

The offset value *o* for a polygon is

$$o = m * factor + r * units.$$

m means the maximum value of incremental ratio to polygon's depth value. *o* can have the value of [0, 1] range. And, *r* (0x34000001) is a H/W characteristic value related to the usable resolution of the depth buffer. The *factor* scales *m* and the *units* scales *r*. The offset value o is applied to depth values of all pixels of a polygon.

- RA_DOffEn, R/W, Address = 0x0003_8004
- RA_DOffFactor, R/W, Address = 0x0003_8008
- RA_DOffUnits, R/W, Address = 0x0003_800C
- RA_DOffRIn, R/W, Address = 0x0003_8010

RA_DOffEn	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
DOffEn	[0]	Depth offset usage control register 0b = Disable 1b = Enable	0b

RA_DOffFactor	Bit	Description	Reset Value
DOffFactor	[31:0]	Factor scales the max depth slope of the polygon. Used to calculate the depth offset value.	0x0

RA_DOffUnits	Bit	Description	Reset Value
DOffUnits	[31:0]	Unit scale an implementation dependent constant that relates to the usable resolution of the depth buffer. Used to calculate the depth offset value.	0x0

RA_DOffRIn	Bit	Description	Reset Value
DOffRIn	[31:0]	An implementation dependent constant that relates to the usable resolution of the depth buffer. Used to calculate the depth offset value of a polygon.	0x34000001

6.3.3 Back-Face Cull Control Register (RA_BFCULL, R/W, Address = 0X0003_8014)

RA_BFCULL	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
BCullEn	[3]	Back-face culling enable 0b = Disable 1b = Enable	0b
BCullFront	[2]	Front face selection 0b = CCW 1b = CW	0b
BCullFace	[1:0]	Culling face selection 00b = back 01b = front 10b = reserved 11b = front and back	00b

6.3.4 Window Y Clipping Region Register (RA_YCLIP, R/W, Address = 0X0003_8018)

In programmer's view, screen has (0, 0) pixel in the left-bottom corner. But, in the view of H/W, (0, 0) pixel is in the left-top corner. So, it is necessary to flip the y coordinate value when Y clipping value is set. After receiving the value of *dd_min_val* and *dd_max_val* from device driver, the following equation is used to get *MIN, MAX* values.

MAX = (screen_height_val - 1) - dd_min_val

MIN = (screen_height_val - 1) - dd_max_val

RA_YCLIP	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
YMaxVal	[27:16]	Y clipping max coordinate (y < YMaxVal)	0x0
Reserved	[15:12]	Reserved	0
YMinVal	[11:0]	Y clipping min coordinate (YMinVal ≤ y)	0x0

6.3.5 Level of Detail Control Register (RA_LODCTL, R/W, Address = 0X0003_C000)

This register serves more programmability to pixel shader. LOD coefficients (K1 ~ K12) can be used in pixel shader and are controlled by LODCTL. LOD coefficients generated according to LODCTL are as follows.

DDY	DDX	LOD	LOD coefficients
0	0	0	No
0	0	1	K1 K3 K5 K2 K4 K6
0	1	0	K1 K3 K2 K4 K7 K9 K8 K10
0	1	1	K1 K3 K5 K2 K4 K6 K7 K9 K8 K10
1	0	0	K1 K5 K2 K6 K7 K11 K8 K12
1	0	1	K1 K3 K5 K2 K4 K6 K7 K11 K8 K12
1	1	0	K1 K3 K5 K2 K4 K6 K7 K9 K11 K8 K10 K12
1	1	1	K1 K3 K5 K2 K4 K6 K7 K9 K11 K8 K10 K12

RA_LODCTL	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
LodCon7	[23:21]	{DDY, DDX, LOD} for attribute 7 000b = all LOD coefficient disable 001b = LOD coefficients calculation enable 010b = DDX coefficients calculation enable 011b = DDX & LOD coefficients enable 100b = DDY coefficients calculation enable 101b = DDY & LOD coefficient enable 110b = DDY & DDX coefficient enable 111b = all LOD coefficient enable	000b
LodCon6	[20:18]	Same as above	000b
LodCon5	[17:15]	Same as above	000b
LodCon4	[14:12]	Same as above	000b
LodCon3	[11:9]	Same as above	000b
LodCon2	[8:6]	Same as above	000b
LodCon1	[5:3]	Same as above	000b
LodCon0	[2:0]	Same as above	000b

6.3.6 Window X Clipping Region Register (RA_CLIPX, R/W, Address = 0X0003_C004)

RA_CLIPX	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
Xright	[27:16]	X-clip right(Max) coordinate (x < Xright)	0x0
Reserved	[15:12]	Reserved	0
Xleft	[11:0]	X clip left(Min) coordinate (Xleft ≤ x)	0x0

6.3.7 Point Width Control Registers (FGRA)

- RA_PWIDTH, R/W, Address = 0x0003_801C
- RA_PSIZE_MIN, R/W, Address = 0x0003_8020
- RA_PSIZE_MAX, R/W, Address = 0x0003_8024

Point width value is clamped by Point Width Min and Point Width Max values.

RA_PWIDTH	Bit	Description	Reset Value
PointWidth	[31:0]	Specify point width value (floating point).	0x3F800000

RA_PSIZE_MIN	Bit	Description	Reset Value
PointSize_Min	[31:0]	Specify point width minimum value (floating point).	0x3F800000

RA_PSIZE_MAX	Bit	Description	Reset Value
PointSize_Max	[31:0]	Specify point width maximum value (floating point).	0x45000000

6.3.8 Coord Replace Control Register (RA_COORDREPLAC, R/W, Address = 0X0003_8028)

This register is used for texture coordinate generation in time of Point Sprite rendering. Only one bit of the eight control bits should be activated. Generated texture coordinates for Point Sprite are stored in the attribute register denoted by the selected number.

RA_COORDREPLAC	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x0
CoordReplace7	[7]	Coord Replace control bit for Attribute #7	0
CoordReplace6	[6]	Coord Replace control bit for Attribute #6	0
CoordReplace5	[5]	Coord Replace control bit for Attribute #5	0
CoordReplace4	[4]	Coord Replace control bit for Attribute #4	0
CoordReplace3	[3]	Coord Replace control bit for Attribute #3	0
CoordReplace2	[2]	Coord Replace control bit for Attribute #2	0
CoordReplace1	[1]	Coord Replace control bit for Attribute #1	0
CoordReplace0	[0]	Coord Replace control bit for Attribute #0	0

6.3.9 Line Width Control Register (RA_LWIDTH, RW, Address = 0X0003_802C)

RA_LWIDTH	Bit	Description	Reset Value
LineWidth	[31:0]	Specify line width value (floating point).	0x3F800000

7 PIXEL SHADER

7.1 OVERVIEW

The proposed shader consists of 4-way floating point SIMD architecture and small scalar core. Each data path has 4-way float type register and scalar register, respectively.

The simplified pixel shader block diagram and its interface are shown in the Figure 9.6- 17. The input data is pixel attributes such as position, color and texture coordinate coming from rasterizer. Instruction and predefined constants are downloaded from host processor for the shader execution. Temporary register, loop counter register, predicate register are cooperated for pixel processing. For the texture mapping, the pixel shader interacts with texture unit. The processed pixel data is finally transferred to per-fragment units through output register.

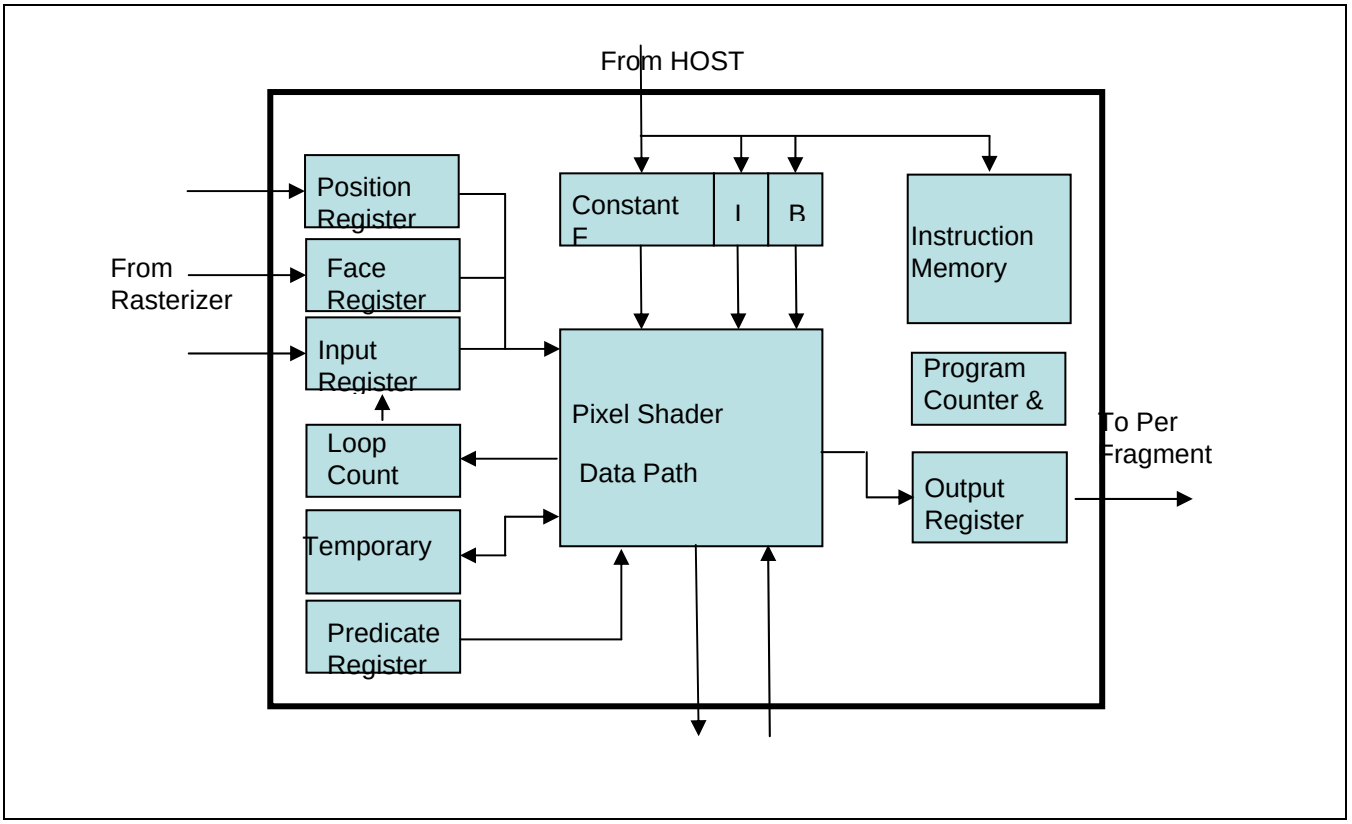


Figure 9.6- 17 Pixel Shader Block Diagram

Programmable shader has two register groups according to its usage. One is special function register (SFR) for HW configuration and the other is program register for shader program. SFR can be accessed by only HOST CPU. Some of program registers such as instruction memory, constant float register, constant integer register and constant Boolean register can be access by both HOST CPU and shader itself according to programmable shader operation mode. Other program register can be access by programmable shader and interface block such as rasterizer or per-fragment unit. In this document, the register which can be accessed by HOST CPU is briefly overviewed. The details of pixel shader operation are covered separated document "the Programmable Shader S/W Developer's Guide".

7.1.1 Instruction Memory (INSTMEM, R/W, Address = 0X0004_0000 ~ 0X0004_1FFF)

The instruction memory has 512 slots and 1 slot is composed of 4 words.

Register	Address	R/W	Description	Reset Value
INSTMEM	0x00040000 ~ 0x00041FFF	R/W	Instruction memory of pixel shader	0xX

7.1.2 Constant Float Register (PS_CFLOAT, R/W, Address = 0X0004_4000 ~ 0X0004_4FFF)

The constant floating-point numbers can be stored in the constant float register for calculating in the program. The constant float register has 256 entries. Each entry is composed of 4 channels, x, y, z, w. Each channel is 32-bit word and has IEEE single precision floating-point format. (De-normalized number is not supported.) Initial value of this register is un-known because it is stored internal sram. User has to initialize this register before use it.

- Constant Float W Component Value (Word 3, R/W Address = 0x0004_4XXC, 0<= XX <= FF)
- Constant Float Z Component Value (Word 2, R/W Address = 0x0004_4XXC, 0<= XX <= FF)
- Constant Float Y Component Value (Word 1, R/W Address = 0x0004_4XXC, 0<= XX <= FF)
- Constant Float X Component Value (Word 0, R/W Address = 0x0004_4XXC, 0<= XX <= FF)

WORD3	Bit	Description	Reset Value
W	[127:96]	Constant float W component value	0xXXXXXXXX

WORD2	Bit	Description	Reset Value
Z	[95:64]	Constant float Z component value	0xXXXXXXXX

WORD1	Bit	Description	Reset Value
Y	[63:32]	Constant float Y component value	0xXXXXXXXX

WORD0	Bit	Description	Reset Value
X	[31:0]	Constant float X component value	0xXXXXXXXX

※ IEEE single precision floating-point format

	Bit	Description	Reset Value
S	[15]	Sign bit	0xXXXXXX
E	[14:10]	Biased exponent	0xXX
F	[9:0]	Fraction	0xXX

Floating-point value V is Not a Number (NaN) if E=63 and F is nonzero

- -Infinity (-INF) if E=63 and F=0 and S=1
- +Infinity (+INF) if E=63 and F=0 and S=0

$(-1)^S * 2^{(E-31)} * (1.F)$ if 0<E<63 where "1.F" is intended to represent the binary number created by prefixing F with an implicit leading 1 and a binary point

De-normalized number $(-1)^S * 2^{(-126)} * (0.F)$ if E=0 and F is nonzero (Not supported)

- -0 if E=0 and F=0 and S=1
- +0 if E=0 and F=0 and S=0

7.1.3 Constant Integer Register (PS_CINT, R/W, Address = 0X0004_8000 ~ 0X0004_803F)

The constant integer values can be stored in the constant integer register. The constant integer value is only used for the flow control – that is the iteration count for loop or the index of relative addressing. The constant integer register has 16 entries, and each entry is composed of 4-channel 8-bit unsigned integer value. User has to initialize this register before use it.

7.1.4 Word (0X0004_80XX, XX IS Word Address Ranged FROM 00 TO 3C)

WORD	Bit	Description	Reset Value
W	[31:24]	Constant integer W component value	0xX
Z	[23:16]	Constant integer Z component value	0xX
Y	[15:8]	Constant integer Y component value	0xX
X	[7:0]	Constant integer X component value	0xX

7.1.5 Constant Bool Register (PS_CBOOL, R/W, Address = 0X0004_8400)

The constant bool value can be stored in the constant bool register. The constant bool value is only used for the static flow control. It is referenced in the conditional branch instruction. The constant bool register is 16-bit boolean register. The register number corresponds to the each bit position. 'TRUE' is represented by '1' and 'FALSE' by '0'. User has to initialize this register before use it.

PS_CBOOL	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
REG15	[15]	Constant Bool register 15	0
REG14	[14]	Constant Bool register 14	0
REG13	[13]	Constant Bool register 13	0
REG12	[12]	Constant Bool register 12	0
REG11	[11]	Constant Bool register 11	0
REG10	[10]	Constant Bool register 10	0
REG9	[9]	Constant Bool register 9	0
REG8	[8]	Constant Bool register 8	0
REG7	[7]	Constant Bool register 7	0
REG6	[6]	Constant Bool register 6	0
REG5	[5]	Constant Bool register 5	0
REG4	[4]	Constant Bool register 4	0
REG3	[3]	Constant Bool register 3	0
REG2	[2]	Constant Bool register 2	0
REG1	[1]	Constant Bool register 1	0
REG0	[0]	Constant Bool register 0	0

7.1.6 Special Function Register For Hw Configuration (FGPS) (PS_EXEMODE, R/W, ADDRESS = 0X0004_C800)

PS_ExeMode	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
PS_ExeMode	[0]	Select Pixel Shader Execution Mode Register 0 = HOST Access Mode (PSHostMode). HOST CPU can access HW configuration register. Some program register such as instruction memory, float/integer/bool constant register cab be downloaded in this mode. 1 = Pixel Shader Execution Mode. (PSExeMode). Pixel shader runs normal operation. In this mode, host CPU can read only the status register of PS_IBStatus and PS_IsNotEmpty. If host CPU change the other registers value, the pixel shader operation is unpredictable. The mode change constraints. PSHostMode → PSExeMode : Set All configuration register value Load Instruction, Constant F/B/I register Confirm PS_IBStatus is '0'. Assert PS_ExeMode to '1'. PSExeMode → PSHostMode : Confirm IsNotEmpty_PS is '0' for all pixel shader. Assert PS_ExeMode to '0'	0b

7.1.7 Special Function Register For Hw Configuration (FGPS) (PS_ PCSTART, R/W, ADDRESS = 0X0004_C804)

PS_PCStart	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
PS_PCStart	[8:0]	When the pixel shader start operation, the first instruction, which is stored at PS_PCStart, is fetched from instruction memory. This register value should be copied to pixel shader program counter by PS_PCCopy register before shader starts.This value is the address of first instruction which is executed when the pixel shader starts. It is copied to program counter of pixel shader by command.	0x0

7.1.8 Special Function Register For Hw Configuration (FGPS) (PS_ PCEND, R/W, ADDRESS = 0X0004_C808)

PS_PCEnd	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0
PS_IgnorePCEnd	[9]	When this bit is set to 1, PS_PCEnd value is ignored and the only program counter stack empty condition is used for program termination.	0
PS_PCEnd	[8:0]	When the program counter is reached at the value of PS_PCEnd, the shader program is terminated after execution of the instruction at this register. This method can save instruction slot and number of instruction count to execute. The other way to terminate pixel shader program : Pixel shader program can be terminated by the extra "ret" instruction which makes program counter stack empty condition. Usually, "call" and "ret" instruction works a pair. But the intentional unmatched "ret" makes pixel shader termination condition. By this exception, pixel shader program can be terminated.	0x1FF

7.1.9 Special Function Register For Hw Configuration (FGPS) (PS_ PCCOPY, R/W, ADDRESS = 0X0004_C80C)

PS_PCCopy	Bit	Description	Reset Value
Reserved	[31:1]	Eeserved	0
PS_PCCopy	[0]	When this bit is set to '1', PS_PCStart register value is actually copied to the pixel shader inside. This bit is auto cleared to '0' after copy. Without copy command, the value of PS_PCStart is not used, and the previous value is used for program start.	0b

7.1.10 Special Function Register For Hw Configuration (FGPS) (PS_ ATTRIBUTENUM, R/W, ADDRESS = 0X0004_C810)

PS_AttributeNum	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
PS_AttributeNum	[3:0]	This register value can have the value ranged between 1 and 8 according to the number of semantics such as color and texture coordinate which are transferred to pixel shader input register from rasterizer. If no semantics are transferred into input register and only position is transferred to position register, PS_ AttributeNum should be set to '1'. Otherwise, this register is set to the number of semantics transferred to pixel shader input register. If pixel shader program use more semantics than that transferred to pixel shader input register, the pixel shader output is unpredictable.	0x8

7.1.11 Special Function Register For Hw Configuration (FGPS) (PS_ IBSTATUS, R, ADDRESS = 0X0004_C814)

PS_IBStatus	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
PS_IBStatus	[0]	This is read only register for monitoring pixel shader input buffer IsNotReady. When PS_AttributeNum value is changed, the pixel shader input buffer initialization sequence starts. During the input bffer initialization, this bit is set to '1'. After initialization, it is automatically cleared to '0'.	0b

8 TEXTURE UNITS

8.1 OVERVIEW

Texture unit (for pixel)

Texture unit includes address generation logic, filtering unit, texture cache, and decompression unit. Vertex texture unit includes address generation unit and vertex texture cache.

Texture unit supports Non 2^N-sized Textures (Arbitrary Rectangular Texture).

- Up to Eight Simultaneous Textures (Max. 8 Texture Samplers)
  - Each texture context is fully configured by dedicated register sets.
  - Controllable Min./Max. Mipmap levels for each texture.
- Max. Width/Height of Textures: 2048x2048
- Min. Width/Height of Textures: 1x1
- Max. Mipmap Levels: 12 Levels
- Texture Size fo Mipmap Level *i*
  - Max [1, floor {(Width of level 0 texture) / 2^i}] x Max[1, floor{(Height of level 0 texture) / 2^i}]
- Texture unit supports 2D texture, Cubemap, and 3D texture.
  - For 3D textures, Level 0 texture only
- Bilinear, Trilinear Filtering is supported. Also S3TC compression format and paletted texture format are supported.
  - For S3TC compression format, Min. Width/Height of Textures is 4 x 4.
  - Paletted texture format : 1bpp, 2bpp, 4bpp, 8bpp
- Mipmapping and the following Wrap Modes are supported
  - Non-parametric coordinate: Clamp to edge
  - Parametric coordinate: Repeat, Flip (Mirrored repeat), Clamp to edge

Vertex texture unit (for vertex)

Filtering is not supported. Vertex texture unit is only used to fetch 32bit texture data.

Both compress format and paletted texture are not supported.

8.2 TEXTURE UNITS SPECIAL REGISTERS

8.2.1 Texture Status Register 0 ~ 7 (TU_TSTA)

- TU_TSTA0, R/W, Address = 0x0006_0000
- TU_TSTA1, R/W, Address = 0x0006_0050
- TU_TSTA2, R/W, Address = 0x0006_00A0
- TU_TSTA3, R/W, Address = 0x0006_00F0
- TU_TSTA4, R/W, Address = 0x0006_0140
- TU_TSTA5, R/W, Address = 0x0006_0190
- TU_TSTA6, R/W, Address = 0x0006_01E0
- TU_TSTA7, R/W, Address = 0x0006_0230

TU_TSTAn	Bit	Description	Reset Value
Reserved	[31:29]	Reserved	000b
TYPE	[28:27]	Texture type 00b = reserved 01b = 2D enable 10b = Cube Enable 11b = 3D enable	01b
Reserved	[26:23]	Reserved	0000b
CK_SEL	[22:21]	Color Key Enable/Selection 00b = Disable 01b = Enable (Use Color Key Register 1 or Color Key YUV Register) 10b = Disable 11b = Enable (Use Color Key Register 2 or Color Key YUV Register)	00b
TEX_EXP	[20]	Texture Value Expansion Method 0b = Duplicate LSB 1b = Zero Padding	0b
AFORMAT_SEL	[19]	Alpha Location Selection 0b = ARGB 1b = RGBA	0b
PAL_TEX_FORMAT	[18:17]	RGB Format in Palette 00b = 1555 01b = 565 10b = 4444 11b = 8888	00b
TEXTURE_FORMAT	[16:12]	Texture Format 00000b = 1555 00001b = 565 00010b = 4444 00011b = Depth Component 16 00100b = 88(intensity/alpha) 00101b = 8 (monochromatic map, replicated in all channel)	00000b

TU_TSTAn	Bit	Description	Reset Value
		00110b = 8888 00111b = 1BPP 01000b = 2BPP 01001b = 4BPP 01010b = 8BPP 01011b = S3TC 01100b = YUV422 , Y1VY0U ordering 01101b = YUV422 , VY1UY0 ordering 01110b = YUV422 , Y1UY0V ordering 01111b = YUV422 , UY1VY0 ordering 10000b~11111b = reserved	
UADDR_MODE	[11:10]	Mode used in U address 00b = Repeat 01b = Flip 10b = Clamp to edge 11b = reserved *Note: In non-parametric coordinate system, UADDR_MODE are set to clamp to edge.	00b
VADDR_MODE	[9:8]	Mode used in V address 00b = Repeat 01b = Flip 10b = Clamp to edge 11b = reserved *Note: In non-parametric coordinate system, VADDR_MODE are set to clamp to edge	00b
PADDR_MODE	[7:6]	Mode used in P address 00b = Repeat 01b = Flip 10b = Clamp to edge 11b = reserved *Note: In non-parametric coordinate system, PADDR_MODE are set to clamp to edge	00b
Reserved	[5]	Eeserved	0b
TEX_COOR	[4]	Texture Addressing Coordinate System 0b = Parametric 1b = Non-parametric	0b
MAG_FILTER	[3]	Bilinear Filter Control (Magnification) 0b = Don't filter between pixels in a map 1b = Bilinear filter between pixels in a map	0b
TEX_FLT_EN	[2]	Bilinear filter Control (Minification) 0b = Don't filter between pixels in a map 1b = Bilinear filter between pixels in a map	0b
MIPMAP_EN	[1:0]	MIPMAP Control 00b = Don't use MIPAMP 01b = MIPMAP use, select nearest map 10b = MIPMAP use, and linear interpolation between maps. 11b = reserved	00b

8.2.2 Texture U Size Register 0 ~ 7 (TU_USIZE)

- TU_USIZE0, R/W, Address = 0x0006_0004
- TU_USIZE1, R/W, Address = 0x0006_0054
- TU_USIZE2, R/W, Address = 0x0006_00A4
- TU_USIZE3, R/W, Address = 0x0006_00F4
- TU_USIZE4, R/W, Address = 0x0006_0144
- TU_USIZE5, R/W, Address = 0x0006_0194
- TU_USIZE6, R/W, Address = 0x0006_01E4
- TU_USIZE7, R/W, Address = 0x0006_0234

TU_USIZEn	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0
U_SIZE	[10:0]	U Size of Level 0 Texture	0x0

8.2.3 Texture V Size Register 0 ~ 7 (TU_VSIZE)

- TU_VSIZE0, R/W, Address = 0x0006_0008
- TU_VSIZE1, R/W, Address = 0x0006_0058
- TU_VSIZE2, R/W, Address = 0x0006_00A8
- TU_VSIZE3, R/W, Address = 0x0006_00F8
- TU_VSIZE4, R/W, Address = 0x0006_0148
- TU_VSIZE5, R/W, Address = 0x0006_0198
- TU_VSIZE6, R/W, Address = 0x0006_01E8
- TU_VSIZE7, R/W, Address = 0x0006_0238

TU_VSIZEn	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0
V_SIZE	[10:0]	V Size of Level 0 Texture	0x0

8.2.4 Texture P Size Register 0 ~ 7 (TU_PSIZE)

- TU_PSIZE0, R/W, Address = 0x0006_000C
- TU_PSIZE0, R/W, Address = 0x0006_005C
- TU_PSIZE0, R/W, Address = 0x0006_00AC
- TU_PSIZE0, R/W, Address = 0x0006_00FC
- TU_PSIZE0, R/W, Address = 0x0006_014C
- TU_PSIZE0, R/W, Address = 0x0006_019C
- TU_PSIZE0, R/W, Address = 0x0006_01EC
- TU_PSIZE0, R/W, Address = 0x0006_023C

TU_PSIZEn	Bit	Description	Reset Value
Reserved	[31:11]	Reserved	0
P_SIZE	[10:0]	P Size of Level 0 Texture (the Depth of 3D Textures)	0x0

8.2.5 Texture L1 Offset Register 0 ~ 7 (TU_TOFFS_L1)

- TU_TOFFS_L1_0, R/W, Address = 0x0006_0010
- TU_TOFFS_L1_1, R/W, Address = 0x0006_0060
- TU_TOFFS_L1_2, R/W, Address = 0x0006_00B0
- TU_TOFFS_L1_3, R/W, Address = 0x0006_0100
- TU_TOFFS_L1_4, R/W, Address = 0x0006_0150
- TU_TOFFS_L1_5, R/W, Address = 0x0006_01A0
- TU_TOFFS_L1_6, R/W, Address = 0x0006_01F0
- TU_TOFFS_L1_7, R/W, Address = 0x0006_0240

TU_TOFFS_L1n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 1 Texture Offset	0x0

8.2.6 Texture L2 Offset Register 0 ~ 7 (TU_TOFFS_L2)

- TU_TOFFS_L2_0, R/W, Address = 0x0006_0014
- TU_TOFFS_L2_1, R/W, Address = 0x0006_0064
- TU_TOFFS_L2_2, R/W, Address = 0x0006_00B4
- TU_TOFFS_L2_3, R/W, Address = 0x0006_0104
- TU_TOFFS_L2_4, R/W, Address = 0x0006_0154
- TU_TOFFS_L2_5, R/W, Address = 0x0006_01A4
- TU_TOFFS_L2_6, R/W, Address = 0x0006_01F4
- TU_TOFFS_L2_7, R/W, Address = 0x0006_0244

TU_TOFFS_L2n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 2 Texture Offset	0x0

8.2.7 Texture L3 Offset Register 0 ~ 7 (TU_TOFFS_L3)

- TU_TOFFS_L3_0, R/W, Address = 0x0006_0018
- TU_TOFFS_L3_1, R/W, Address = 0x0006_0068
- TU_TOFFS_L3_2, R/W, Address = 0x0006_00B8
- TU_TOFFS_L3_3, R/W, Address = 0x0006_0108
- TU_TOFFS_L3_4, R/W, Address = 0x0006_0158
- TU_TOFFS_L3_5, R/W, Address = 0x0006_01A8
- TU_TOFFS_L3_6, R/W, Address = 0x0006_01F8
- TU_TOFFS_L3_7, R/W, Address = 0x0006_0248

TU_TOFFS_L3n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 3 Texture Offset	0x0

8.2.8 Texture L4 Offset Register 0 ~ 7 (TU_TOFFS_L4)

- TU_TOFFS_L4_0, R/W, Address = 0x0006_001C
- TU_TOFFS_L4_1, R/W, Address = 0x0006_006C
- TU_TOFFS_L4_2, R/W, Address = 0x0006_00BC
- TU_TOFFS_L4_3, R/W, Address = 0x0006_010C
- TU_TOFFS_L4_4, R/W, Address = 0x0006_015C
- TU_TOFFS_L4_5, R/W, Address = 0x0006_01AC
- TU_TOFFS_L4_6, R/W, Address = 0x0006_01FC
- TU_TOFFS_L4_7, R/W, Address = 0x0006_024C

TU_TOFFS_L4n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 4 Texture Offset	0x0

8.2.9 Texture L5 Offset Register 0 ~ 7 (TU_TOFFS_L5)

- TU_TOFFS_L5_0, R/W, Address = 0x0006_0020
- TU_TOFFS_L5_1, R/W, Address = 0x0006_0070
- TU_TOFFS_L5_2, R/W, Address = 0x0006_00C0
- TU_TOFFS_L5_3, R/W, Address = 0x0006_0110
- TU_TOFFS_L5_4, R/W, Address = 0x0006_0160
- TU_TOFFS_L5_5, R/W, Address = 0x0006_01B0
- TU_TOFFS_L5_6, R/W, Address = 0x0006_0200
- TU_TOFFS_L5_7, R/W, Address = 0x0006_0250

TU_TOFFS_L5n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 5 Texture Offset	0x0

8.2.10 Texture L6 Offset Register 0 ~ 7 (TU_TOFFS_L6)

- TU_TOFFS_L6_0, R/W, Address = 0x0006_0024
- TU_TOFFS_L6_1, R/W, Address = 0x0006_0074
- TU_TOFFS_L6_2, R/W, Address = 0x0006_00C4
- TU_TOFFS_L6_3, R/W, Address = 0x0006_0114
- TU_TOFFS_L6_4, R/W, Address = 0x0006_0164
- TU_TOFFS_L6_5, R/W, Address = 0x0006_01B4
- TU_TOFFS_L6_6, R/W, Address = 0x0006_0204
- TU_TOFFS_L6_7, R/W, Address = 0x0006_0254

TU_TOFFS_L6n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 6 Texture Offset	0x0

8.2.11 Texture L7 Offset Register 0 ~ 7 (TU_TOFFS_L7)

- TU_TOFFS_L7_0, R/W, Address = 0x0006_0028
- TU_TOFFS_L7_1, R/W, Address = 0x0006_0078
- TU_TOFFS_L7_2, R/W, Address = 0x0006_00C8
- TU_TOFFS_L7_3, R/W, Address = 0x0006_0118
- TU_TOFFS_L7_4, R/W, Address = 0x0006_0168
- TU_TOFFS_L7_5, R/W, Address = 0x0006_01B8
- TU_TOFFS_L7_6, R/W, Address = 0x0006_0208
- TU_TOFFS_L7_7, R/W, Address = 0x0006_0258

TU_TOFFS_L7n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 7 Texture Offset	0x0

8.2.12 Texture L8 Offset Register 0 ~ 7 (TU_TOFFS_L8)

- TU_TOFFS_L8_0, R/W, Address= 0x0006_002C
- TU_TOFFS_L8_1, R/W, Address= 0x0006_007C
- TU_TOFFS_L8_2, R/W, Address= 0x0006_00CC
- TU_TOFFS_L8_3, R/W, Address= 0x0006_011C
- TU_TOFFS_L8_4, R/W, Address= 0x0006_016C
- TU_TOFFS_L8_5, R/W, Address= 0x0006_01BC
- TU_TOFFS_L8_6, R/W, Address= 0x0006_020C
- TU_TOFFS_L8_7, R/W, Address= 0x0006_025C

TU_TOFFS_L8n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 8 Texture Offset	0x0

8.2.13 Texture L9 Offset Register 0 ~ 7 (TU_TOFFS_L9)

- TU_TOFFS_L9_0, R/W, Address = 0x0006_0030
- TU_TOFFS_L9_1, R/W, Address = 0x0006_0080
- TU_TOFFS_L9_2, R/W, Address = 0x0006_00D0
- TU_TOFFS_L9_3, R/W, Address = 0x0006_0120
- TU_TOFFS_L9_4, R/W, Address = 0x0006_0170
- TU_TOFFS_L9_5, R/W, Address = 0x0006_01C0
- TU_TOFFS_L9_6, R/W, Address = 0x0006_0210
- TU_TOFFS_L9_7, R/W, Address = 0x0006_0260

TU_TOFFS_L9n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 9 Texture Offset	0x0

8.2.14 Texture L10 Offset Register 0 ~ 7 (TU_TOFFS_L10)

- TU_TOFFS_L10_0, R/W, Address = 0x0006_0034
- TU_TOFFS_L10_1, R/W, Address = 0x0006_0084
- TU_TOFFS_L10_2, R/W, Address = 0x0006_00D4
- TU_TOFFS_L10_3, R/W, Address = 0x0006_0124
- TU_TOFFS_L10_4, R/W, Address = 0x0006_0174
- TU_TOFFS_L10_5, R/W, Address = 0x0006_01C4
- TU_TOFFS_L10_6, R/W, Address = 0x0006_0214
- TU_TOFFS_L10_7, R/W, Address = 0x0006_0264

TU_TOFFS_L10n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 10 Texture Offset	0x0

8.2.15 Texture L11 Offset Register 0 ~ 7 (TU_TOFFS_L11)

- TU_TOFFS_L11_0, R/W, Address = 0x0006_0038
- TU_TOFFS_L11_1, R/W, Address = 0x0006_0088
- TU_TOFFS_L11_2, R/W, Address = 0x0006_00D8
- TU_TOFFS_L11_3, R/W, Address = 0x0006_0128
- TU_TOFFS_L11_4, R/W, Address = 0x0006_0178
- TU_TOFFS_L11_5, R/W, Address = 0x0006_01C8
- TU_TOFFS_L11_6, R/W, Address = 0x0006_0218
- TU_TOFFS_L11_7, R/W, Address = 0x0006_0268

TU_TOFFS_L11n	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 11 Texture Offset	0x0

8.2.16 Texture Min Level Register 0 ~ 7 (TU_T_MIN_L)

- TU_T_MIN_L0, R/W, Address = 0x0006_003C
- TU_T_MIN_L1, R/W, Address = 0x0006_008C
- TU_T_MIN_L2, R/W, Address = 0x0006_00DC
- TU_T_MIN_L3, R/W, Address = 0x0006_012C
- TU_T_MIN_L4, R/W, Address = 0x0006_017C
- TU_T_MIN_L5, R/W, Address = 0x0006_01CC
- TU_T_MIN_L6, R/W, Address = 0x0006_021C
- TU_T_MIN_L7, R/W, Address = 0x0006_026C

TU_T_MIN_Ln	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
MIN_LEVEL	[3:0]	Texture Mipmap Min level	0x0

8.2.17 Texture Max Level Register 0 ~ 7 (TU_T_MAX_L)

- TU_T_MAX_L0, R/W, Address = 0x0006_0040
- TU_T_MAX_L1, R/W, Address = 0x0006_0090
- TU_T_MAX_L2, R/W, Address = 0x0006_00E0
- TU_T_MAX_L3, R/W, Address = 0x0006_0130
- TU_T_MAX_L4, R/W, Address = 0x0006_0180
- TU_T_MAX_L5, R/W, Address = 0x0006_01D0
- TU_T_MAX_L6, R/W, Address = 0x0006_0220
- TU_T_MAX_L7, R/W, Address = 0x0006_0270

TU_T_MAX_Ln	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
MAX_LEVEL	[3:0]	Texture Mipmap Max Level	0x0

8.2.18 Texture Base Address Register 0 ~ 7 (TU_TBADD)

- TU_TBADD0, R/W, Address = 0x0006_0044
- TU_TBADD1, R/W, Address = 0x0006_0094
- TU_TBADD2, R/W, Address = 0x0006_00E4
- TU_TBADD3, R/W, Address = 0x0006_0134
- TU_TBADD4, R/W, Address = 0x0006_0184
- TU_TBADD5, R/W, Address = 0x0006_01D4
- TU_TBADD6, R/W, Address = 0x0006_0224
- TU_TBADD7, R/W, Address = 0x0006_0274

TU_TBADDn	Bit	Description	Reset Value
ADDR	[31:0]	Texture Base Address for Level 0	0xXXXXXXXX

8.2.19 Texture Color Key Register (TU_CKEY)

- TU_CKEY1, R/W, Address = 0x0006_0280
- TU_CKEY2, R/W, Address = 0x0006_0284
- TU_ CKYUV, R/W, Address = 0x0006_0288
- TU_ CKMASK, R/W, Address = 0x0006_028C

TU_CKEY1	Bit	Description	Reset Value
reserved	[31:24]	reserved	
R	[23:16]	Color key red value not YUV and CK_SEL = 01	0x0
G	[15:8]	Color key green value not YUV and CK_SEL = 01	0x0
B	[7:0]	Color key blue value not YUV and CK_SEL = 01	0x0

TU_CKEY2	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	
R	[23:16]	Color key red value not YUV and CK_SEL = 11	0x0
G	[15:8]	Color key green value not YUV and CK_SEL = 11	0x0
B	[7:0]	Color key blue value not YUV and CK_SEL = 11	0x0

TU_CKYUV	Bit	Description	Reset Value
Reserved	[31:0]	Reserved	
VALU	[15:8]	Color key U value YUV and CK_SEL(0) = 1	0x0
VALV	[7:0]	Color key V value YUV and CK_SEL(0) = 1	0x0

TU_CKMASK	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	
VAL	[2:0]	000b = mask no bits of color key 001b = mask 1 lsb of each CK color component 010b = mask 2 lsb of each CK color component 011b = mask 3 lsb of each CK color component 100b = mask 4 lsb of each CK color component 101b = mask 5 lsb of each CK color component 110b = mask 6 lsb of each CK color component 111b = mask 7 lsb of each CK color component	000b

8.2.20 Texture Palette Register (TU_PALETTE)

- TU_ PALETTE_ADDR, W, Address = 0x0006_0290
- TU_ PALETTE_IN, W, Address = 0x0006_0294

TU_PALETTE_ADDR	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
ADDR	[7:0]	Palette address	0x0

TU_PALETTE_IN	Bit	Description	Reset Value
DATA	[31:0]	Palette data in	0x0

8.2.21 Vertex Texture Status Register (VT_VTSTA)

- VT_VTSTA0, R/W, Address = 0x0006_02C0
- VT_VTSTA1, R/W, Address = 0x0006_02C8
- VT_VTSTA2, R/W, Address = 0x0006_02D0
- VT_VTSTA3, R/W, Address = 0x0006_02D8

VT_VTSTAn	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	
UMOD	[11:10]	Mode used in u address 00b = repeat 01b = flip 11b = clamp to edge 11b = reserved	00b
VMOD	[9:8]	Mode used in v address 00b = repeat 01b = flip 11b = clamp to edge 11b = reserved	00b
USIZE	[7:4]	Texture u size 0000b = 1 pixel 0001b = 2 pixels 0010b = 4 pixels 0011b = 8 pixels 0100b = 16 pixels 0101b = 32 pixels 0110b = 64 pixels 0111b = 128 pixels 1000b = 256 pixels 1001b = 512 pixels 1010b = 1024 pixels 1011b = 2048 pixels 1100b ~ 1111b = reserved	0x0
VSIZE	[3:0]	Texture v size 0000b = 1 pixel 0001b = 2 pixels 0010b = 4 pixels 0011b = 8 pixels 0100b = 16 pixels 0101b = 32 pixels 0110b = 64 pixels 0111b = 128 pixels 1000b = 256 pixels 1001b = 512 pixels 1010b = 1024 pixels 1011b = 2048 pixels 1100b ~ 1111b = reserved	0x0

8.2.22 Vertex Texture Base Address Register (VT_VTBADDR)

- VT_VTBADDR0, R/W, Address = 0x0006_02C4
- VT_VTBADDR1, R/W, Address = 0x0006_02CC
- VT_VTBADDR2, R/W, Address = 0x0006_02D4
- VT_VTBADDR3, R/W, Address = 0x0006_02DC

VT_VTBADDRn	Bit	Description	Reset Value
ADDR	[31:0]	Vertex texture bass address	0xXXXXXXXX

9 PER-FRAGMENT UNIT

9.1 OVERVIEW

All OpenGL 2.0 per-fragment operations are supported. Depth Buffer and Stencil Buffer are supported, depth buffer bit depth is 24-bit and stencil buffer bit depth is 8-bit.

- Per-Fragment Unit support Scissor Test
- Per-Fragment Unit support Alpha Test
- Per-Fragment Unit support Stencil Test (Stencil Buffer is 8-bit) and Stencil Operation. Also support both front stencil buffer and back stencil buffer.
- Per-Fragment Unit support Depth Test (Depth Buffer is 24-bit)
- Per-Fragment Unit Support Alpha Blending
- Per-Fragment Unit Support Logical Operation
- Per-Fragment Unit Support 16/32bit color mode
- For Enhanced Color Per-Fragment Unit support Dithering

Pixel ownership test is determining whether the destination pixel is visible or obscured by an overlapping window.

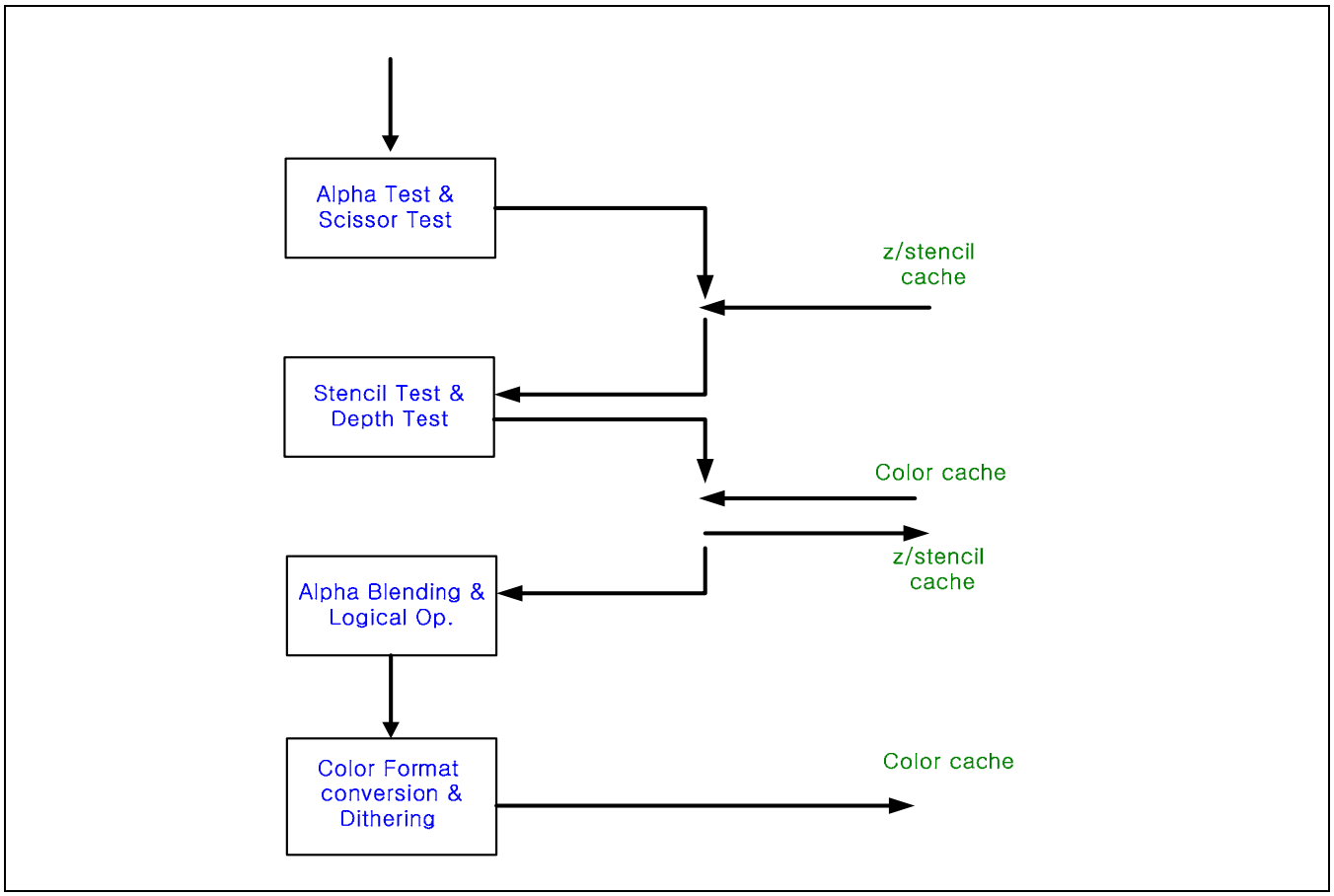


Figure 9.6- 18 Per-Fragment Function Block Diagram

9.2 PER-FRAGMENT UNIT SPECIAL REGISTERS

9.2.1 Scissor Test Control Register (PF_SCISSOR)

If you want to discard all fragments, please set XMin=0 , XMax=0, YMin=0 and YMax=0. Also if you want to pass all fragments, please set XMin=0, XMax= Max screen X width, YMin=0, and YMax= Max screen Y height.

- PF_SCISSOR_X, R/W, Address = 0x0007_0000
- PF_SCISSOR_Y, R/W, Address = 0x0007_0004

PF_SCISSOR_X	Bit	Description	Reset Value
ScissorTestEnable	[31]	Scissor test enable bit 0b = disable 1b = enable	0b
Reserved	[30:28]	Reserved	0
XMax	[27:16]	Pixel's X coordinate >= MAX is not written to the frame buffer	0x0
reserved	[15:12]	Reserved	0
XMin	[11:0]	Pixel's X coordinate < MIN is not written to the frame buffer	0x0

PF_SCISSOR_Y	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
YMax	[27:16]	Pixel's Y coordinate >= MAX is not written to the frame buffer	0x0
Reserved	[15:12]	Reserved	0
YMin	[11:0]	Pixel's Y coordinate < MIN is not written to the frame buffer	0x0

9.2.2 Alpha Test Control Register (PF_ALPHAT , R/W, Address = 0X0007_0008)

PF_ALPHAT	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	0
AlphaTestValue	[11:4]	8bit alpha test reference value	0x0
AlphaTestMode	[3:1]	Mode used in alpha test 000b = NEVER 001b = ALWAYS 010b = LESS 011b = LEQUAL(less than or equal to) 100b = EQUAL 101b = GREATER 110b = GEQUAL 111b = NOTEQUAL	000b
AlphaTestEnable	[0]	Alpha test enable 0b = alpha test disabled 1b = alpha test enabled	0b

9.2.3 Stencil Test Control Register (FGPF)

- PF_FRONTST, R/W, Address = 0x0007_000C
- PF_ BACKST, R/W, Address = 0x0007_0010

PF_FRONTST	Bit	Description	Reset Value
FrontStencil_dppass	[31:29]	Stencil depth buffer pass action 000b = KEEP 001b = ZERO 010b = REPLACE 011b = INCR 100b = DECR 101b = INVERT 110b = INCR_WRAP 111b = DECR_WRAP	000b
FrontStencil_dpfail	[28:26]	Stencil depth buffer fail action Same as above	000b
FrontStencil_sfail	[25:23]	Stencil fail action Same as above	000b
reserved	[22:20]	reserved	
FrontStencilMaskValue	[19:12]	8-bit stencil mask value	0x0
FrontStencilTestValue	[11:4]	8-bit stencil reference value	0x0
FrontStencilTestMode	[3:1]	Mode used in stencil test 000b = NEVER 001b = ALWAYS 010b = LESS 011b = LEQUAL(less than or equal to) 100b = EQUAL 101b = GREATER 110b = GEQUAL 111b = NOTEQUAL	000b
StencilTestEnable	[0]	0b = stencil test disabled 1b = stencil test enabled	0b

PF_BACKST	Bit	Description	Reset Value
BackStencil_dppass	[31:29]	Stencil depth buffer pass action 000b = KEEP 001b = ZERO 010b = REPLACE 011b = INCR 100b = DECR 101b = INVERT 110b = INCR_WRAP 111b = DECR_WRAP	000b
BackStencil_dpfail	[28:26]	Stencil depth buffer fail action Same as above	000b
BackStencil_sfail	[25:23]	Stencil fail action Same as above	000b
Reserved	[22:20]	Reserved	
BackStencilMaskValue	[19:12]	8-bit Stencil mask value	0x0
BackStencilTestValue	[11:4]	8-bit Stencil reference value	0x0
BackStencilTestMode	[3:1]	Mode used in Stencil test 000b = NEVER 001b = ALWAYS 010b = LESS 011b = LEQUAL 100b = EQUAL 101b = GREATER 110b = GEQUAL 111b = NOTEQUAL	000b
Reserved	[0]	Reserved	0

9.2.4 Depth Test Control Register (PF_DEPTHT, R/W, Address = 0X0007_0014)

PF_DEPTHT	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	
DepthTestMode	[3:1]	Mode used in depth test 000b = NEVER 001b = ALWAYS 010b = LESS 011b = LEQUAL 100b = EQUAL 101b = GREATER 110b = GEQUAL 111b = NOTEQUAL	001b
DepthTestEnable	[0]	0b = depth buffer test disabled 1b = depth buffer test enabled	0b

9.2.5 Blending Control Register (FGPF)

- PF_CCLR, R/W, Address = 0x0007_0018
- PF_ BLEND, R/W, Address = 0x0007_001C

PF_CCLR	Bit	Description	Reset Value
VAL	[31:0]	Blend constant RGBA color	0x0

PF_BLEND	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0
ABlendEquation	[22:20]	Alpha Blend Equation 000b = Add 001b = Subtract 010b = Reverse Subtract 011b = Min 100b = Max	000b
BlendEquation	[19:17]	Blend Equation 000b = Add 001b = Subtract 010b = Reverse Subtract 011b = Min 100b = Max	000b
AlphaDstBlendFunc	[16:13]	Mode used in Blending destination function 0000b = ZERO 0001b = ONE 0010b = SRC_COLOR 0011b = ONE_MINUS_SRC_COLOR 0100b = DST_COLOR 0101b = ONE_MINUS_DST_COLOR 0110b = SRC_ALPHA 0111b = ONE_MINUS_SRC_ALPHA 1000b = DST_ALPHA 1001b = ONE_MINUS_DST_ALPHA 1010b = CONSTANT_COLOR 1011b = ONE_MINUS_CONSTANT_COLOR 1100b = CONSTANT_ALPHA 1101b = ONE_MINUS_CONSTANT_ALPHA 1110b = SRC_ALPHA_SATURATE	0x0
ColorDstBlendFunc	[12:9]	Mode used in Blending destination function Same as above	0x0
AlphaSrcBlendFunc	[8:5]	Mode used in Blending source function Same as above	0x0
ColorSrcBlendFunc	[4:1]	Mode used in Blending source function Same as above	0x0
BlendingEnable	[0]	0b = blending disabled 1b = blending enabled	0b

9.2.6 Logical Operation Control Register (PF_LOGOP, R/W, Address = 0X0007_0020)

This register supports binary bitwise logic operation between source and destination.

PF_LOGOP	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
AlphaLogOpEnable	[8:5]	Mode used in Logical Operation Argument value / Operation 0000 CLEAR 0 0001 AND s & d 0010 AND_REVERSE s & ~d 0011 COPY s 0100 AND_INVERTED ~s & d 0101 NOOP d 0110 XOR s xor d 0111 OR s | d 1000 NOR ~(s | d) 1001 EQUIV ~(s xor d) 1010 INVERT ~d 1011 OR_REVERSE s | ~d 1100 COPY_INVERTED ~s 1101 OR_INVERTED ~s | d 1110 NAND ~(s & d) 1111 SET all 1's	0x0
ColorLogOpEnable	[4:1]	Mode used in color logical operation. Same as above.	0x0
LogOpEnable	[0]	0b = color logical operation disabled 1b = color logical operation enabled	0b

9.2.7 Color Buffer Write Mask Register (PF_CBMSK, R/W, Address = 0X0007_0024)

PF_CBMSK	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	
FbColorWrMask	[3:0]	This register is used to mask the writing of R, G, B and A values to the color buffer or buffers. r, g, b, and a indicate whether R, G, B, or A values, respectively, are written or not. In the Reset Value, all bits and all color values are enabled for writing. 0000b = all mask disable 0001b = a mask enable 0010b = b mask enable 0100b = g mask enable 1000b = r mask enable	0x0

9.2.8 Depth/Stencil Buffer Write Mask Register (PF_DBMSK, R/W, Address = 0X0007_0028)

PF_DBMSK	Bit	Description	Reset Value
Back_FbStencilWrMask	[31:24]	This register is used for stencil buffer write mask. Especially for Back side pixel. The stencil buffer write can be enabled or disabled. In the Reset Value, the stencil buffer is enabled for writing when stencil test turns on. Stencil buffer is consist of 8-bit per pixel, so each bit of this register can mask each value of stencil buffer For example, 00000000b, in this case All 8bits are updated to stencil buffer. 00000001b, in this case All 8bits are updated to stencil buffer except lsb 1 bit. 00000011b, in this case All 8bits are updated to stencil buffer except lsb 2 bit.	0x0
Front_FbStencilWrMask	[23:16]	This register is used for stencil buffer write mask. Especially for Front side pixel The stencil buffer write can be enabled or disabled. In the Reset Value, the stencil buffer is enabled for writing when stencil test turns on. Stencil buffer is consist of 8-bit per pixel, so each bit of this register can mask each value of stencil buffer	0x0
Reserved	[15:1]	Reserved	0
FbDepthWrMask	[0]	The depth buffer can be enabled or disabled for writing depth value. In the Reset Value, the depth buffer is enabled for writing. 0b = depth buffer writing 1b = no depth buffer writing	0b

9.2.9 Frame Buffer Control Register (PF_FBCTL, R/W, Address = 0X0007_002C)

PF_FBCTL	Bit	Description	Reset Value
Reserved	[31:21]	Reserved	0
OpaqueAlpha	[20]	1 = After alpha blending , Alpha value is forced to opaque 0 = Normal operation	0b

AlphaThreshold	[19:12]	Used when encoding 16-bit 1555 format.
		If(Internal alpha value > Alpha threshold value)
		Alpha = 1;
		else
		Alpha = 0;
			0x0h

AlphaConst	[11:4]	Constant alpha value	0x0h
DitherOn	[3]	Control conversion of pixels from internal ARGB8888 format to 16-bit output pixel 0b = dithering disable 1b = dithering enable	0b
ColorMode	[2:0]	Mode used in Frame Buffer Color 000b = 555, RGB, 16-bit (top bit written as register alpha[7]) 001b = 565, RGB, 16-bit 010b = 4444, RGB, 16-bit 011b = 1555, ARGB, 16-bit 100b = 0888, RGB, 32-bit(top byte written as register alpha) 101b = 8888, ARGB, 32-bit 110 ~ 111b = reserved	000b

9.2.10 Depth Buffer Base Address Register (PF_DBADDR, R/W, Address = 0X0007_0030)

PF_DBADDR	Bit	Description	Reset Value
FbDepthOffset	[31:0]	Depth buffer offset address	0x00000000

9.2.11 Color Buffer Base Address Register (PF_CBADDR, R/W, Address = 0X0007_0034)

PF_CBADDR	Bit	Description	Reset Value
FbColorOffset	[31:0]	Color buffer offset address	00000000h

9.2.12 Frame Buffer Width Register (PF_FBW, R/W, Address = 0X0007_0038)

PF_FBW	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	0
FbWidth	[11:0]	Frame buffer width (0~2048)	800h

10 AXI ARBITER & AXI DMA

10.1 OVERVIEW

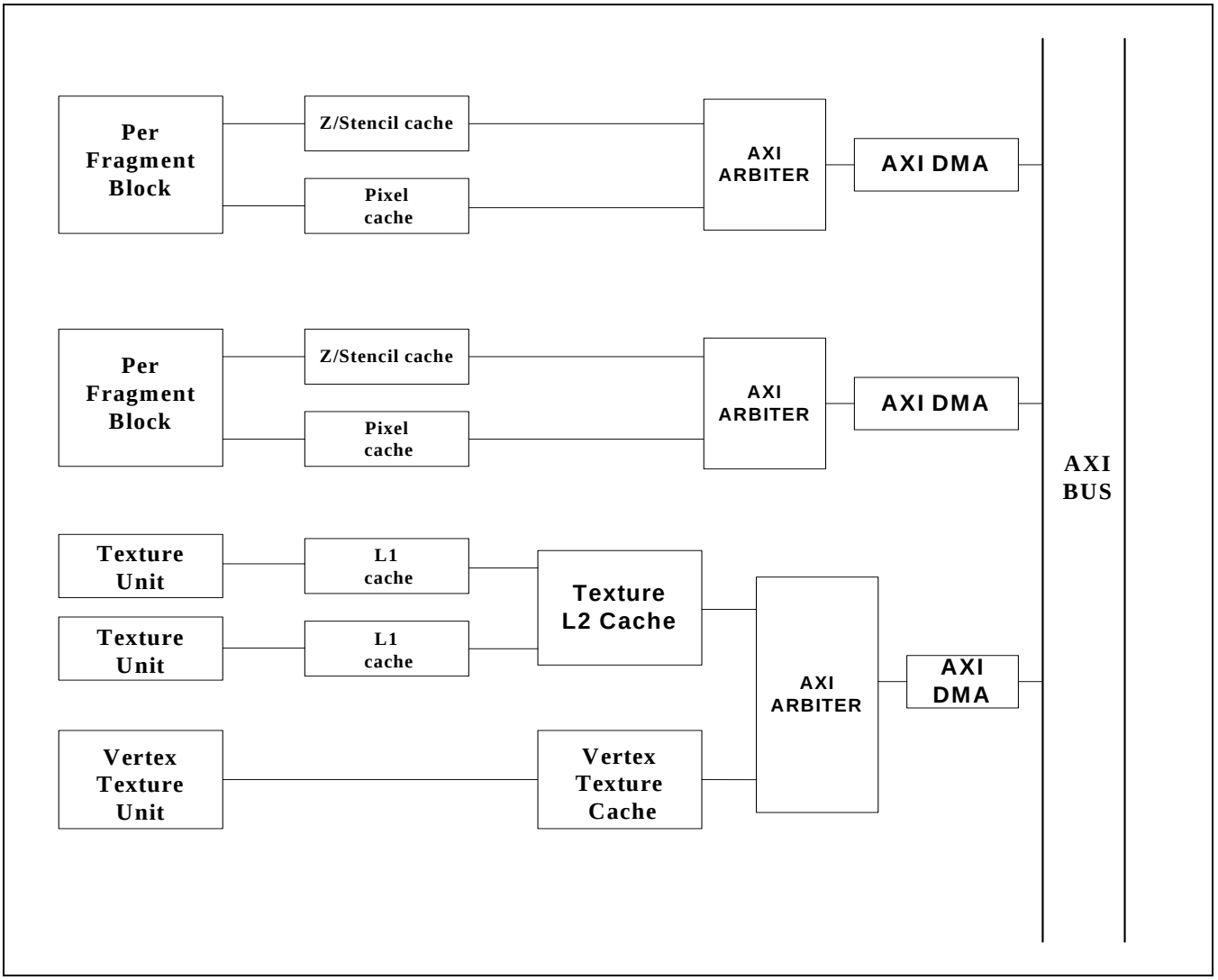


Figure 9.6- 19 AXI DMA Function Block Diagram

10.2 AXI ARBITER FEATURES

- 2 X 1 Arbiter
- Zero Wait Arbitration
- AXI DMA INTERFACE
  - ♦ FIFO INTERFACE
  - ♦ GIVE TRANSACTION ID TO AXI DMA FOR EFFICIENT MEMOTY ACCESS
- IP CORE INTERFACE
  - ♦ FIFO INTERFACE
  - ♦ EASY TO CONNECT WITH IP CORE

10.3 AXI DMA FEATURES

- AXI BUS INTERFACE
  - ♦ SEPARATE ADDR/CRTL, DATA CHANNEL
  - ♦ SEPARATE READ, WRITE CHANNEL
  - ♦ SUPPORT 64-bit DATA BUS SIZE
  - ♦ SUPPORT VARIOUS BURST LENGTH
  - ♦ SUPPORT READ MULTIPLE OUTSTANDING ADDRESS
  - ♦ SUPPORT WRITE MULTIPLE OUTSTANDING ADDRESS
  - ♦ SUPPORT WRITE BYTE MASK
  - ♦ SUPPORT VARIOUS BURST TYPE (INCR, WRAP)
  - ♦ SUPPORT LOW POWER CHANNEL
- IP CORE INTERFACE
  - ♦ FIFO INTERFACE
  - ♦ EASY TO CONNECT WITH IP CORE

Interface with AXI Arbiter0 / AXI Arbiter1 (z/stencil cache or color cache)

- ♦ Interface between AXI Arbiter and DMA is FIFO Interface
- ♦ Separate Read Address Channel and Read Data Channel
- ♦ Separate Write Address Channel and Write Data Channel
- ♦ Burst length is fixed to INCR8 or WRAP8
- ♦ Data transfer size is fixed to Two Word (64-Bit)

Interface with AXI Arbter2 (Vertex texture cache / Texture L2 cache)

- ♦ Interface between AXI Arbiter and DMA is FIFO Interface
- ♦ Separate Read Address Channel and Read Data Channel
- ♦ Burst length is fixed to INCR8 or WRAP8
- ♦ Data transfer size is fixed to Two Word (64-Bit)

Interface with AXI Bus

- FIMG_3DSEV1.1 AXI DMA support AMBA AXI BUS protocol

9.7 TVOUT & VIDEO DAC

1 OVERVIEW

The TVOUT module supports ITU-R BT.470 and EIA-770 compliant analog TV signals with 3 channel 10-bit DAC. The signal format is either CVBS, S-video, and interlaced/ progressive scan YPbPr/ RGB. It also supports EIA-608 compliant closed caption and extended data service, IEC61880 / ITU-R BT.1119 compliant wide screen signaling, and EIA-J CPR1204-1 compliant analog copy generation management system.

2 FEATURE

The TVOUT module includes following features:

- I/O and Control
  - ITU-R BT.601 (YCbCr 4:4:4) input format
  - 10-Bit, 4X over sampled CVBS/S video/YPbPr/RGB output data to 3-channel 54 MHz DAC
  - AHB Slave I/F for register control
- Video Standard Compliances for CVBS and S-video:
  - (M) NTSC, NTSC-J
  - (B/D/G/H/I) PAL, (M) PAL, (N) PAL, (Nc) PAL
  - PAL-60, NTSC4.43
- Ancillary Data Insertion
  - EIA-608 compliant Closed Caption(CC) and Extended Data Service(XDS)
  - IEC61880 / ITU-R BT.1119 compliant Wide Screen Signaling(WSS)
  - EIA-J CPR1204-1 compliant analog copy generation management system(CGMS-A)
- Video Standard Compliances for Component YPbPr and RGB:(NOTE)
  - 525/60 Hz progressive
  - 525/30 Hz interlaced
  - 625/50 Hz progressive
  - 625/25 Hz interlaced

- Post Processing
  - Color Compensation for Invalid RGB Data
  - Programmable 23-Tap Luma/ Chroma Filters for Luma/ Chroma anti-aliasing for CVBS
  - Programmable 95-Tap oversampling filter capable of frequency response compensation

NOTE: SCART RGB (Interlaced-scan) is not supported with 3-channel DAC.

3 FUNCTIONAL DESCRIPTION

3.1 DATA FLOW

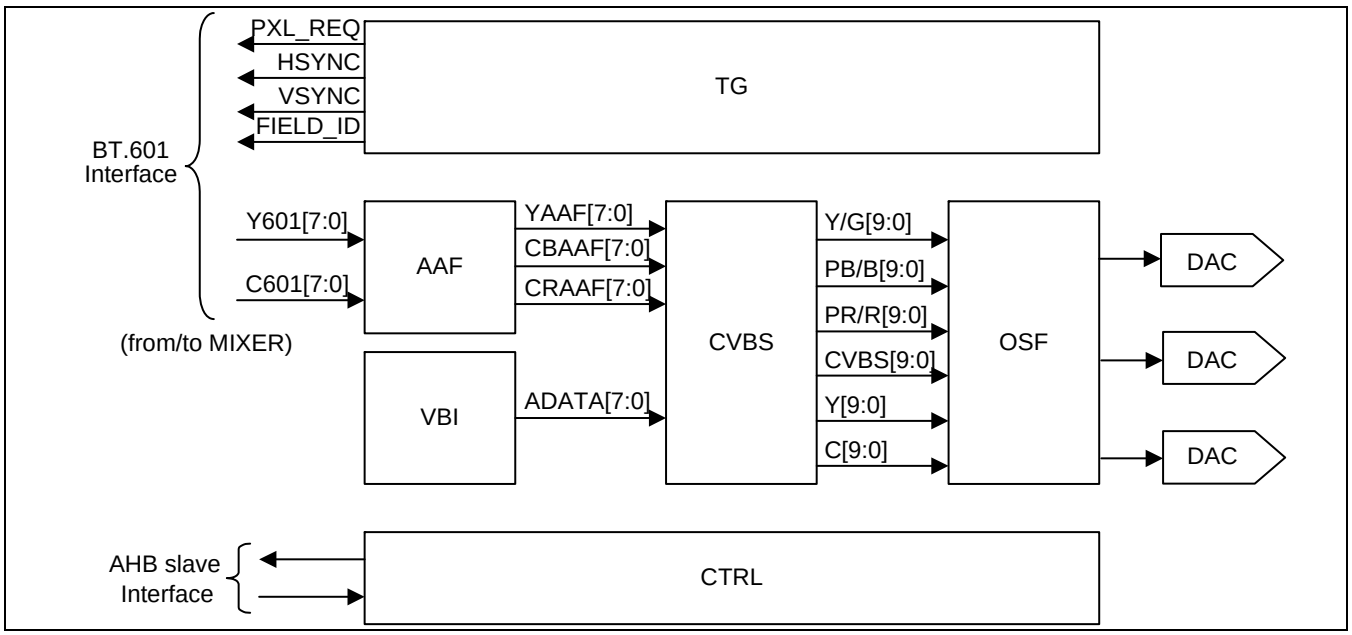


Figure 9.7-1 Data Flow of TVOUT Module

The TVOUT module is composed of the following data sub-modules:

- TG: Timing Generation
- CVBS: Waveform Generation, QAM Modulation, and YCbCr Video Processing
- AAF: Affine Transformation, luma/ chroma Anti-Aliasing Filter
- OSF: 4x Oversampling Filter
- VBI: Ancillary data insertion during vertical blanking interval
- CTRL: Register Control

NOTE: 1.Image Mixer is directly connected with TVOUT and HDMI. So, for making connection completely, it has to configure MIXER_OUT_SEL at Clock Controller.

3.2 TIMING GENERATION (TG MODULE)

The Timing Generation (TG) sub-module generates all the timing information signals required for ITU-R BT.470 compliant TV signals. Horizontal and vertical timing signals are generated by an internal pixel counter and the phase signals of sub-carrier are made by an internal Discrete Time Oscillator (DTO).

There are two kinds of horizontal and vertical timings in TG module:

3.2.1 525/60 Hz

Video standard: NTSC (M), NTSC-J, PAL (M), NTSC 4.43, and PAL 60
Horizontal frequency (FH): 15.734 kHz, 858 samples per line @ 13.5 MHz sample rate
Vertical frequency (FV): 59.94 Hz, 525 lines per frame

3.2.2 625/50 Hz

Video standard: PAL (BGHID), PAL (N), and PAL (Nc)
Horizontal frequency (FH): 15.625 kHz, 864 samples per line @ 13.5 MHz sample rate
Vertical frequency (FV): 50.00 Hz, 625 lines per frame

There are four kinds of discrete timing oscillation for sub-carrier generation in TG module:

3.2.3 3.579545 MHz

Video standard: NTSC (M) and NTSC-J
Sub-carrier frequency (FSC): 910/4 * FH

3.2.4 4.43361875 MHz

Video standard: PAL (BGHID), PAL (N), NTSC 4.43, and PAL 60
Sub-carrier frequency (FSC): (1135/4 + 1/625) * FH

3.2.5 3.57561149 MHz

Video standard: PAL (M)
Sub-carrier frequency (FSC): 909/4 * FH

3.2.6 3.58205625 MHz

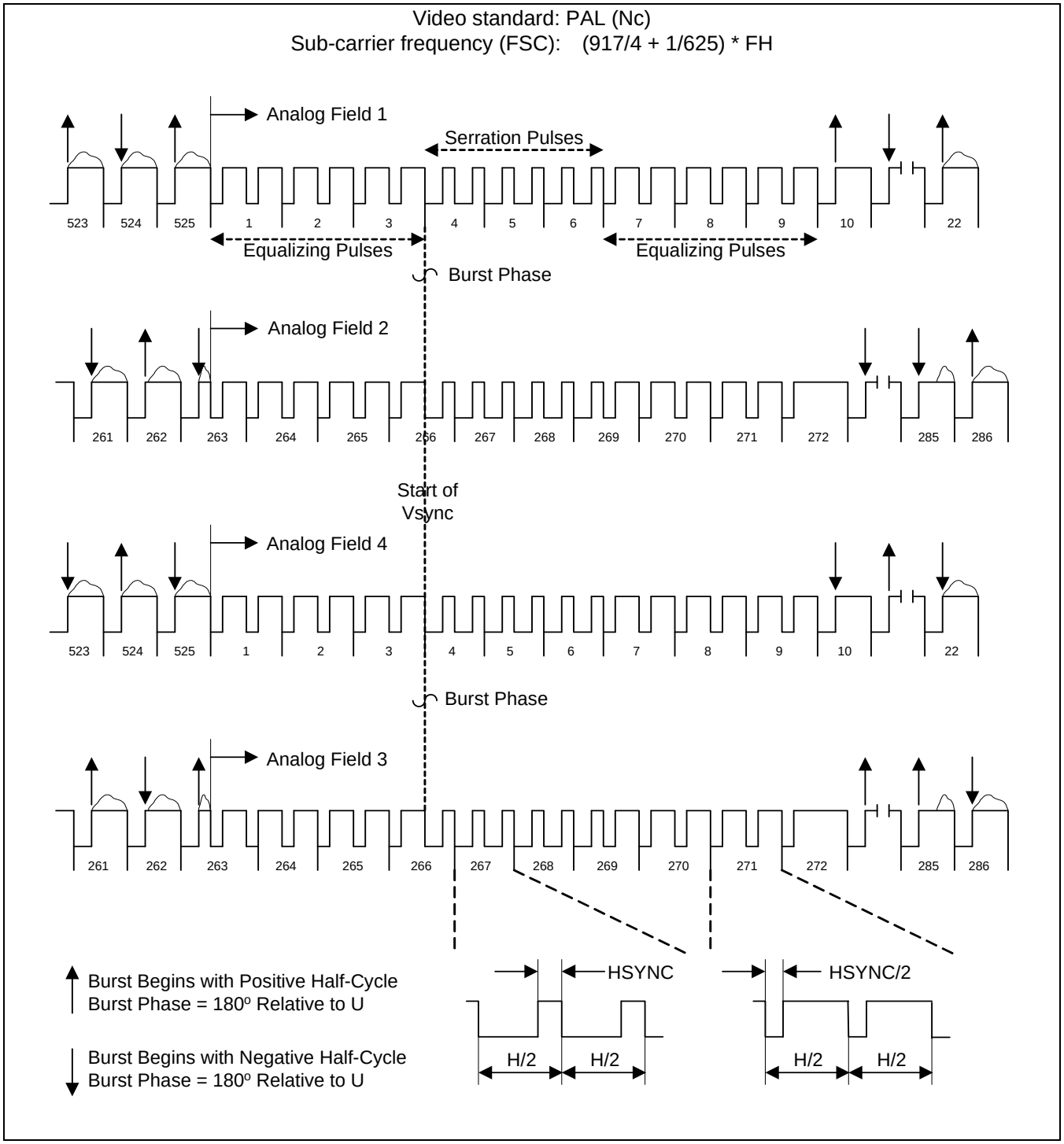


Figure 9.7-2 Four Field NTSC (M) Sequence and Burst Blanking

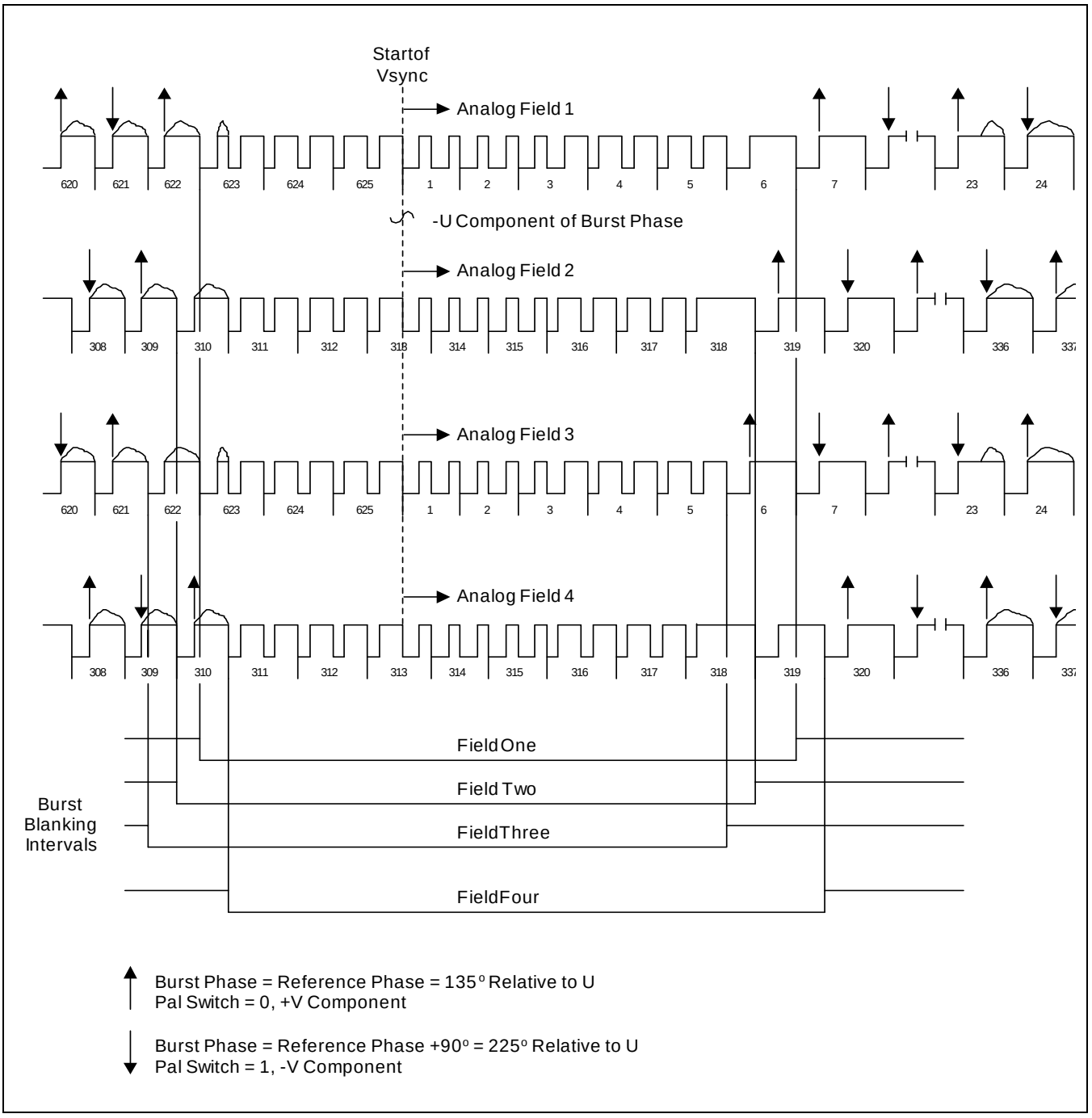


Figure 9.7-3 Field PAL (BGHIDNc) Sequence and Burst Blanking

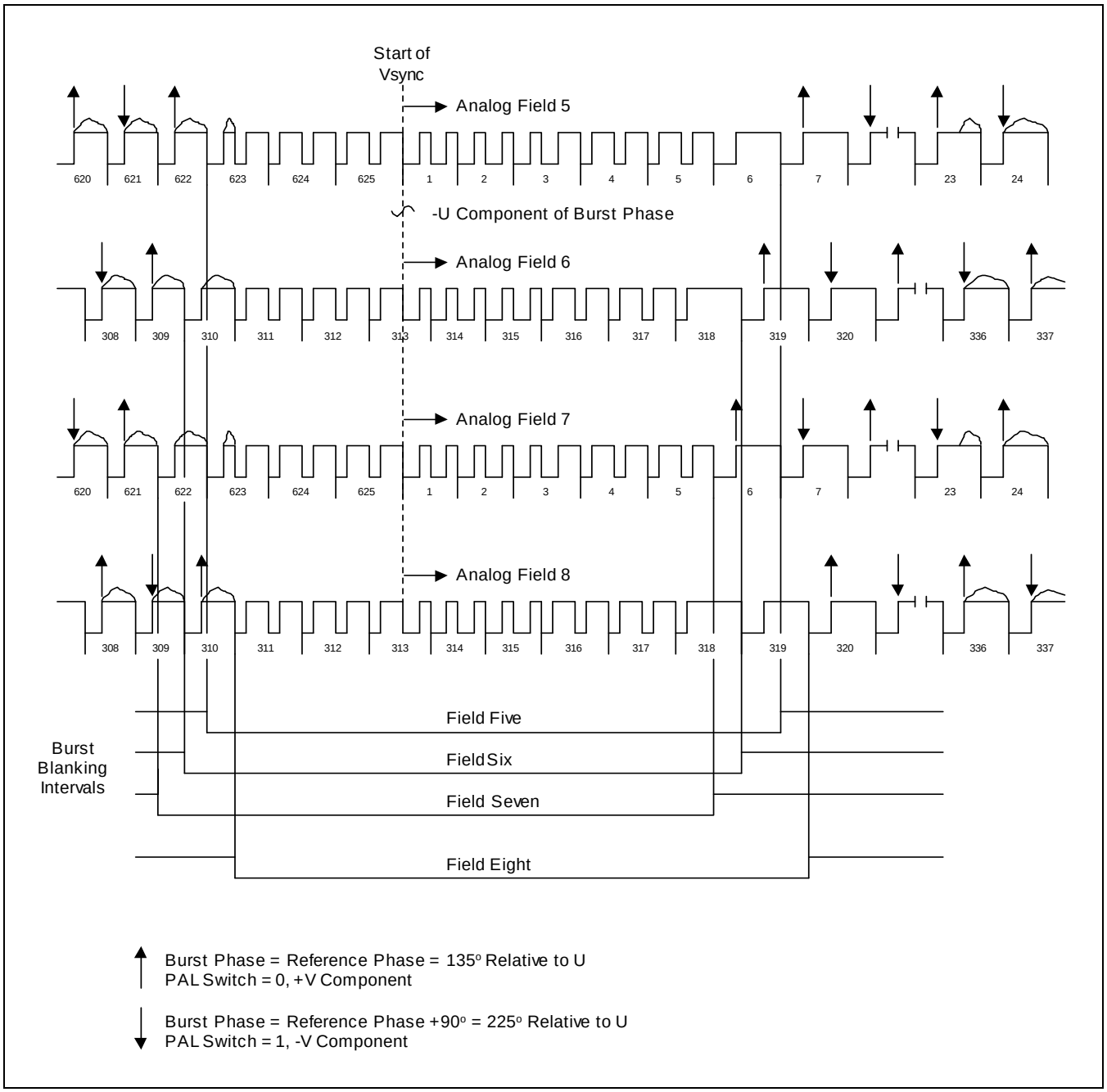


Figure 9.7-4 Eight Field PAL (BGHIDNc) Sequence and Burst Blanking

The internal pixel rate is 13.5 MHz which is 1/4 times of 54 MHz video clock which is used for DAC. With 13.5 MHz pixel rate, the horizontal blanking timing and active video timing are defined as follows:

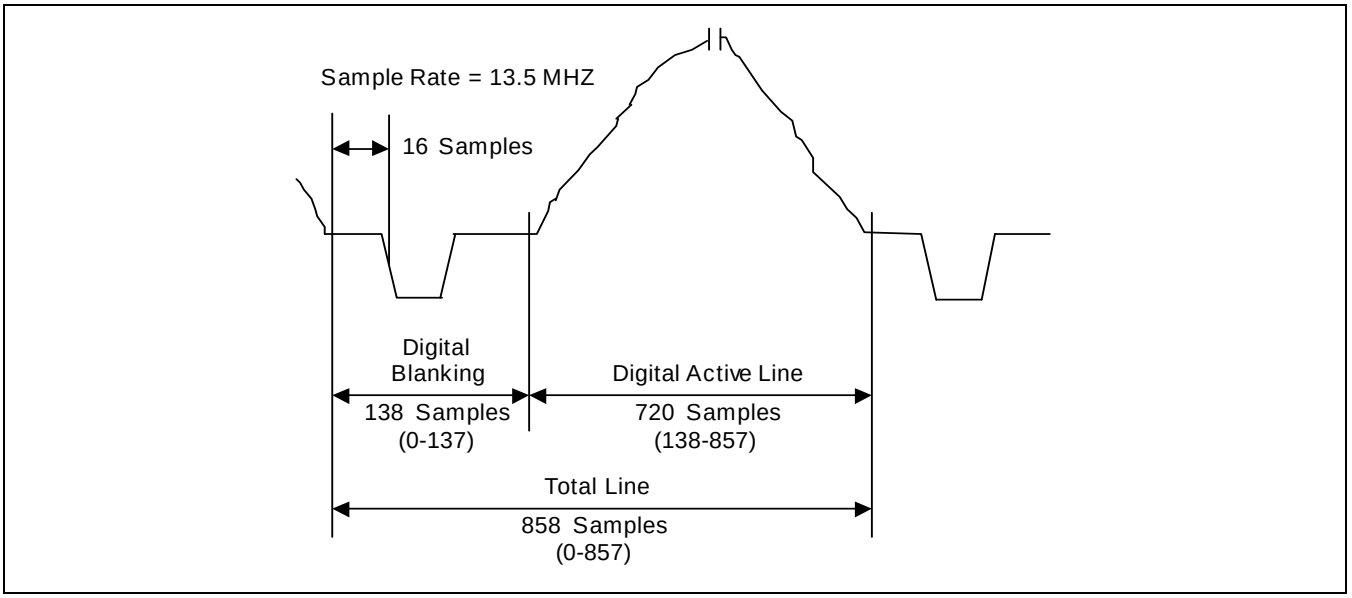


Figure 9.7-5 Horizontal Blanking and Active Video Timing @ 525/60 Hz

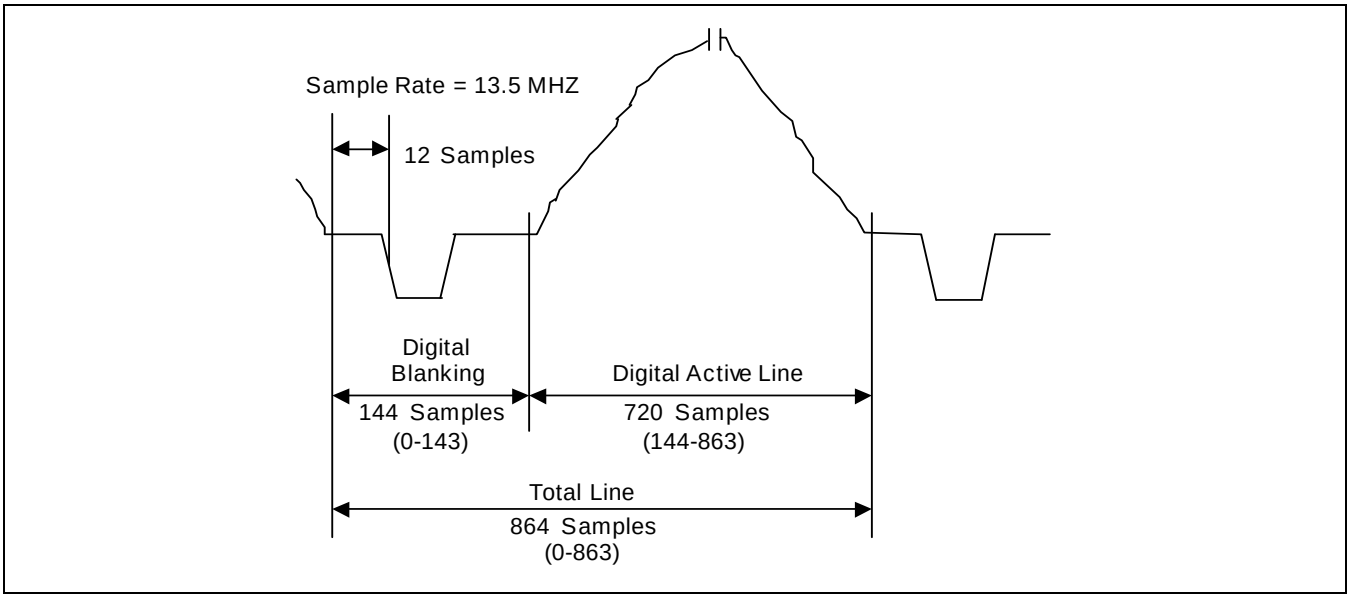


Figure 9.7-6 Horizontal Blanking and Active Video Timing @ 625/50 Hz

All the timing signals in TG module are controlled by VIDEO [3:0] bits in SDO_CONFIG register. Finally note that all the internal counters and discrete time oscillators operate with 54 MHz video clock. The hsync, vsync, field_id, and data request signals are generated and used internally at 54 MHz rate. If they are delivered to 'DispPipe' module their timings are re-synchronized with 135 MHz system clock.

3.3 ANTI ALIASING FILTER (AAF MODULE)

The color TV signals are composed of luminance video, chrominance video and audio data which are modulated by different sub-carriers. Hence, some spectral shaping to avoid the aliasing between them is required. ITU-R BT.407 recommends that the luminance data (Y) should be band-limited within 4.2, 5.0, 5.5, or 6.0 MHz with respect to video standards to avoid the crosstalk between audio sub-carrier and the luminance data. It also recommends that the chrominance data (Cb and Cr) should be band-limited within 1.3 MHz to avoid the crosstalk between chrominance sub-carrier and the luminance data. If audio multiplexing is not required, band-limiting of luminance data (Y) may be skipped. Similarly, band-limiting of chrominance data (C) may be skipped at S-video encoding rather than CVBS encoding.

The AAF sub-module in TVOUT provides 23-tap linear FIR filters for spectral shaping of luminance data (Y) and chrominance data (Cb and Cr). The filter responses are fully programmable since filter coefficients themselves are controllable. Since linear FIR filters have symmetric coefficients, 11 coefficients completely defines the filter responses. The registers SDO_Y0, SDO_Y1… and SDO_Y11 are used to control the luminance filter response. The register SDO_CB0, SDO_CB1… SDO_CB11, and SDO_CR0, SDO_CR1… SDO_CR11 are used to control the chrominance Cb and Cr filter responses, respectively.

The following table show typical settings of filter coefficients. Note that there is no filtering on luminance (Y) data. Since many CE devices usually do not output multiplexed audio, filtering on luminance (Y) data is not required any more. Then, note also that the filter is not normalized and the dc gain of the filter may vary with the video scale. Refer to SDO Video Scale Configuration Register and SDO Anti Aliasing Filter Coefficients.

Table 9.7-1 Filter Coefficients of Anti-aliasing Filters for Luminance Y

Register	7.5 IRE Setup/ 7:3 Sync	7.5 IRE Setup/ 10:4 Sync	0 IRE Setup/ 7:3 Sync	0 IRE Setup/ 10:4 Sync
SDO_Y0	0	0	0	0
SDO_Y1	0	0	0	0
SDO_Y2	0	0	0	0
SDO_Y3	0	0	0	0
SDO_Y4	0	0	0	0
SDO_Y5	0	0	0	0
SDO_Y6	0	0	0	0
SDO_Y7	0	0	0	0
SDO_Y8	0	0	0	0
SDO_Y9	0	0	0	0
SDO_Y10	0	0	0	0
SDO_Y11	252	25D	281	28F

Table 9.7-2 Filter Coefficients of Anti-aliasing Filters for Chrominance Cb

Register	7.5 IRE Setup/ 7:3 Sync	7.5 IRE Setup/ 10:4 Sync	0 IRE Setup/ 7:3 Sync	0 IRE Setup/ 10:4 Sync
SDO_CB0	0	0	0	0
SDO_CB1	0	0	0	0
SDO_CB2	0	0	0	0
SDO_CB3	0	0	0	0
SDO_CB4	0	0	0	0
SDO_CB5	1	1	1	1
SDO_CB6	6	7	7	7
SDO_CB7	13	14	15	15
SDO_CB8	28	28	2A	2B
SDO_CB9	3F	3F	44	45
SDO_CB10	51	52	57	59
SDO_CB11	56	5A	5F	61

Table 9.7-3 Filter Coefficients of Anti-aliasing Filters for Chrominance Cr

Register	7.5 IRE Setup/ 7:3 Sync	7.5 IRE Setup/ 10:4 Sync	0 IRE Setup/ 7:3 Sync	0 IRE Setup/ 10:4 Sync
SDO_CR0	0	0	0	0
SDO_CR1	0	0	0	0
SDO_CR2	0	0	0	0
SDO_CR3	0	0	0	0
SDO_CR4	0	0	0	0
SDO_CR5	2	1	2	2
SDO_CR6	5	9	A	A
SDO_CR7	18	1C	1D	1E
SDO_CR8	37	39	3C	3D
SDO_CR9	5A	5A	5F	61
SDO_CR10	76	74	7B	7A
SDO_CR11	7E	7E	86	8F

The following figures show the magnitude and phase responses of CB and CR anti aliasing filters with the above settings. Note that these filters are applied only to CVBS an S-video.

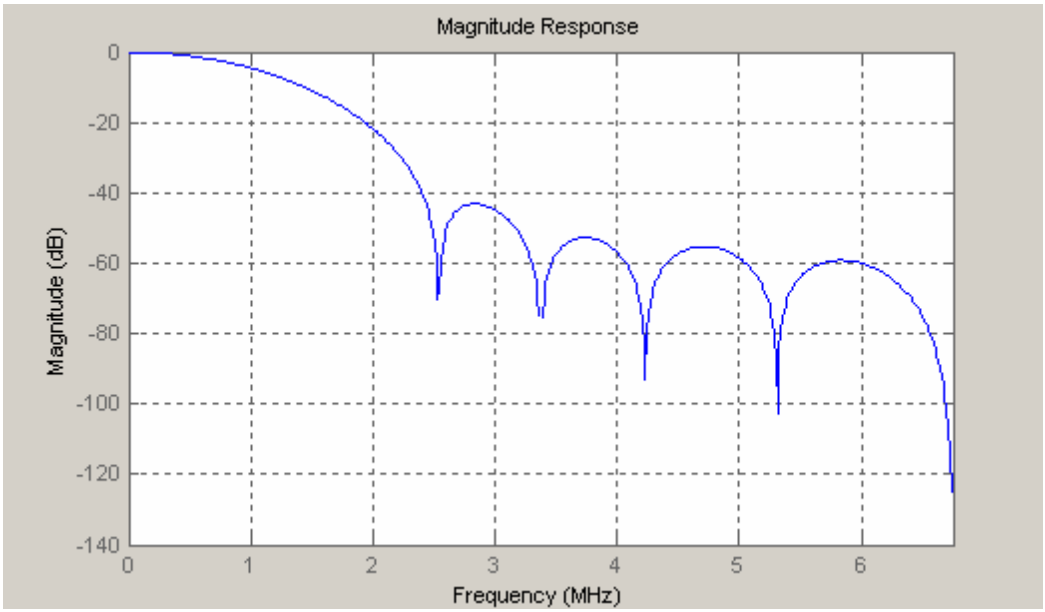


Figure 9.7-7 Magnitude Response of CB and CR Anti Aliasing Filter @ 13.5 MHz Sampling Rate

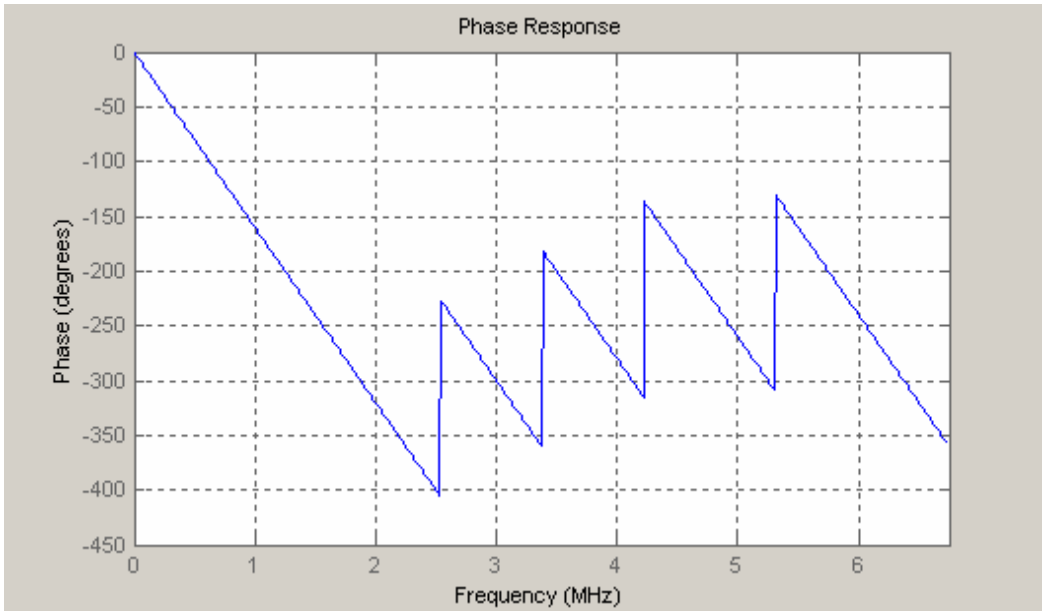


Figure 9.7-8 Phase Response of CB and CR Anti Aliasing Filter @ 13.5 MHz Sampling Rate

3.4 ANCILLARY DATA INSERTION (VBI MODULE)

The ITU-R BT.470 compliant TV signals have blanking lines which do not possess video data. Some ancillary data such as closed caption, content information, display aspect ratio control, and copy control information are delivered within these blanking periods. The TVOUT module supports EIA-608 compliant closed caption (CC) and extended data service (XDS) and IEC61880 / ITU-R BT.1119 compliant wide screen signaling (WSS). The VBI sub-module draws the waveform of the ancillary data delivering signals.

The physical waveform of EIA-608 closed caption and extended data service signals is as follows:

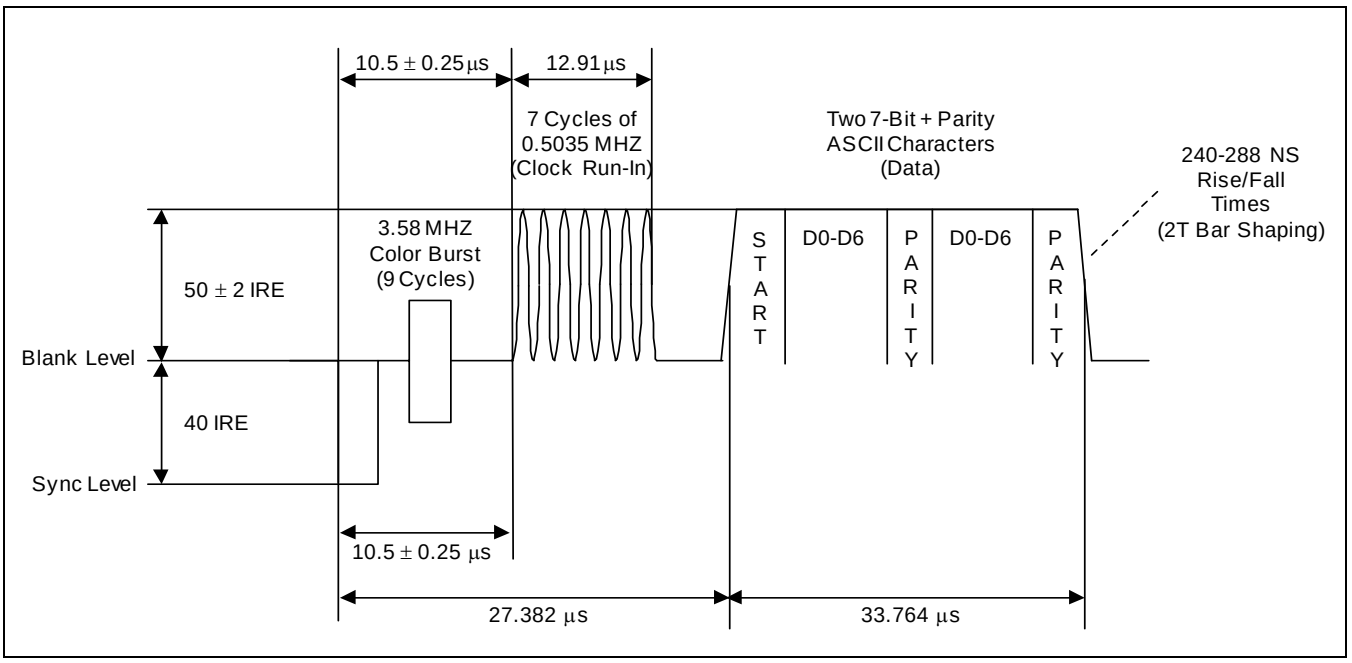


Figure 9.7-9 EIA-608 Closed Caption and Extended Data Service

This waveform is inserted at the 21'st line of 525/60 standard video. The register SDO_ARMCC is used for carrying the 2 byte data into the waveform including the parity bits. For the CC data and XDS data packet format and its usage, refer to the Recommendation EIA-608, "Recommended Practice for Line 21 Data Service".

The physical waveform of IEC 61880 compliant wide screen signaling (WSS) signals is as follows:

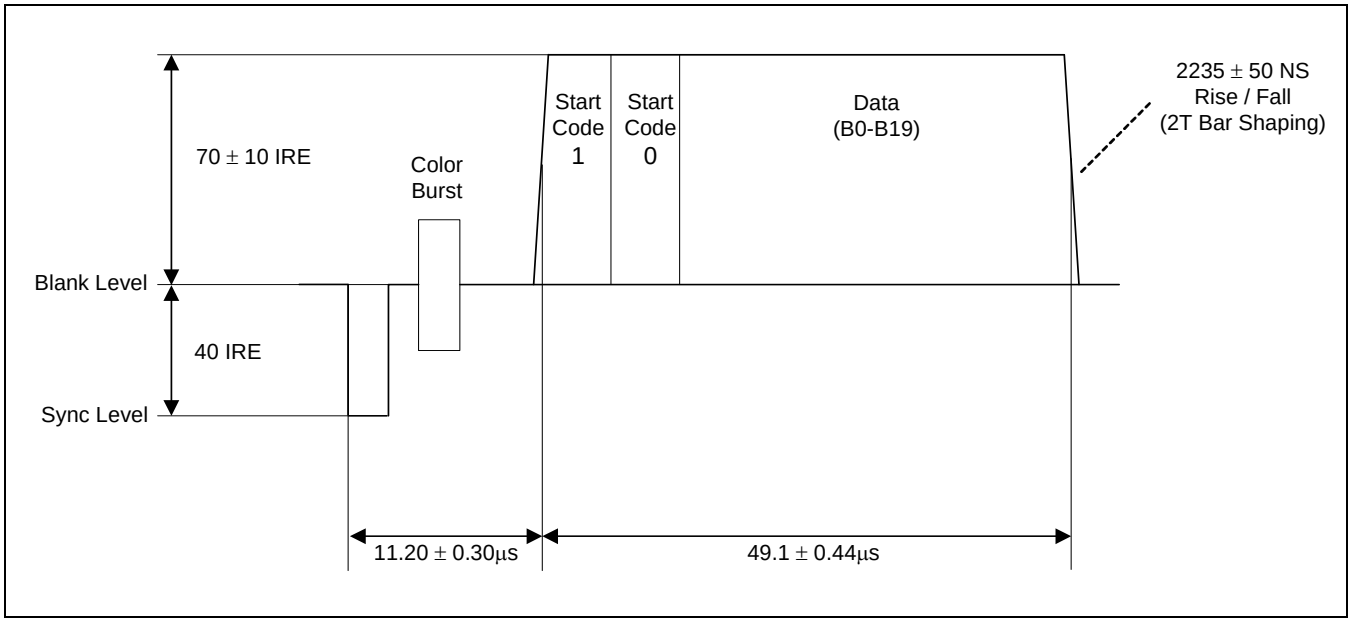


Figure 9.7-10 IEC 61880 Wide Screen Signaling

This waveform is inserted at the 20-th line of 525/60 standard video. The register SDO_ARMWSS525 is used for carrying the 20 bit data into the waveform. Bits {b1, b0} define display aspect ratio control, bit {b7, b6, b5, b4, b3, b2} define copy control information, and bits {b13, b12, b11, b10, b9, b8} is used to specify the operation of the reserved functions. Bits {b19, b18, b17, b16, b15, b14} are used for CRC error check.

The physical waveform of ITU-R BT.1119 compliant Wide Screen Signaling (WSS) signals is as follows:

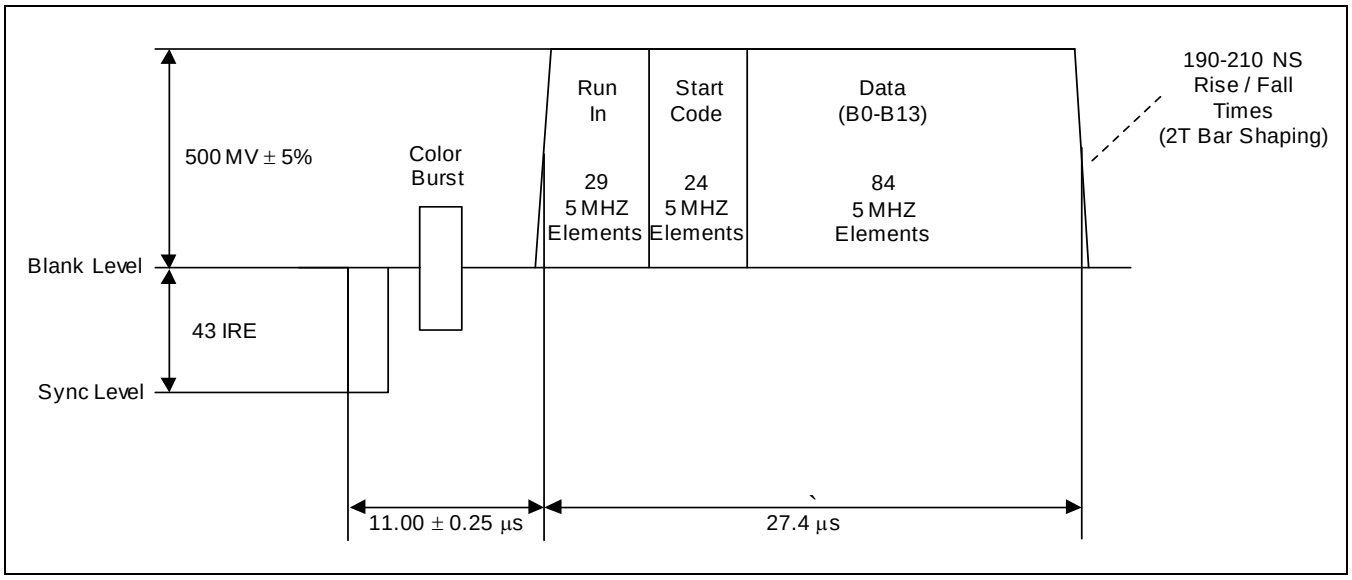


Figure 9.7-11 ITU-R BT.1119 Wide Screen Signaling

This waveform is inserted at the 23rd line of 625/50 standard video. The register SDO_ARMWSS625 is used for carrying the 14-bit data into the waveform. Bits {b3, b2, b1, b0} specify display aspect ratio control, bits {b7, b6, b5, b4} are used for enhanced TV service, bits {b10, b9, b8} is used for European teletext subtitle control, and bits {b13, b12, b11} are used for copy control.

3.5 WAVEFORM GENERATION AND CHROMA MODULATION (CVBS MODULE)

The CVBS sub-module combines the timing information and video data to make the waveforms of ITU-R BT 470 TV signals. This procedure is done by two different data paths. One is for luminance data (Y) and the other is for chrominance data (C). For the luminance data path, horizontal/ vertical synchronization pulses are formed and merged to a properly scaled and offset luminance video data (Y). For the chrominance data paths, base band chrominance data (Cb and Cr) are modulated with a sub-carrier FSC along with the video standard, i.e.

$$C(n) = U(n) * \sin(2\pi\ FSC * n) + V(n) * \cos(2\pi\ FSC * n),$$

where U(n) and V(n) denote properly scaled and offset versions of Cb(n) and Cr(n), respectively. Then a pilot sinusoidal waveform, called a burst, is formed and added prior to the start of modulated chrominance data C at each line. At the end of CVBS sub-module data paths, the luminance data (Y) and the chrominance data (C) is merged into one channel and form composite data (CVBS).

The CVBS sub-module also provides the color space conversion for YPbPr/RGB component outputs. The following figure summarizes the data path in the CVBS sub-module.

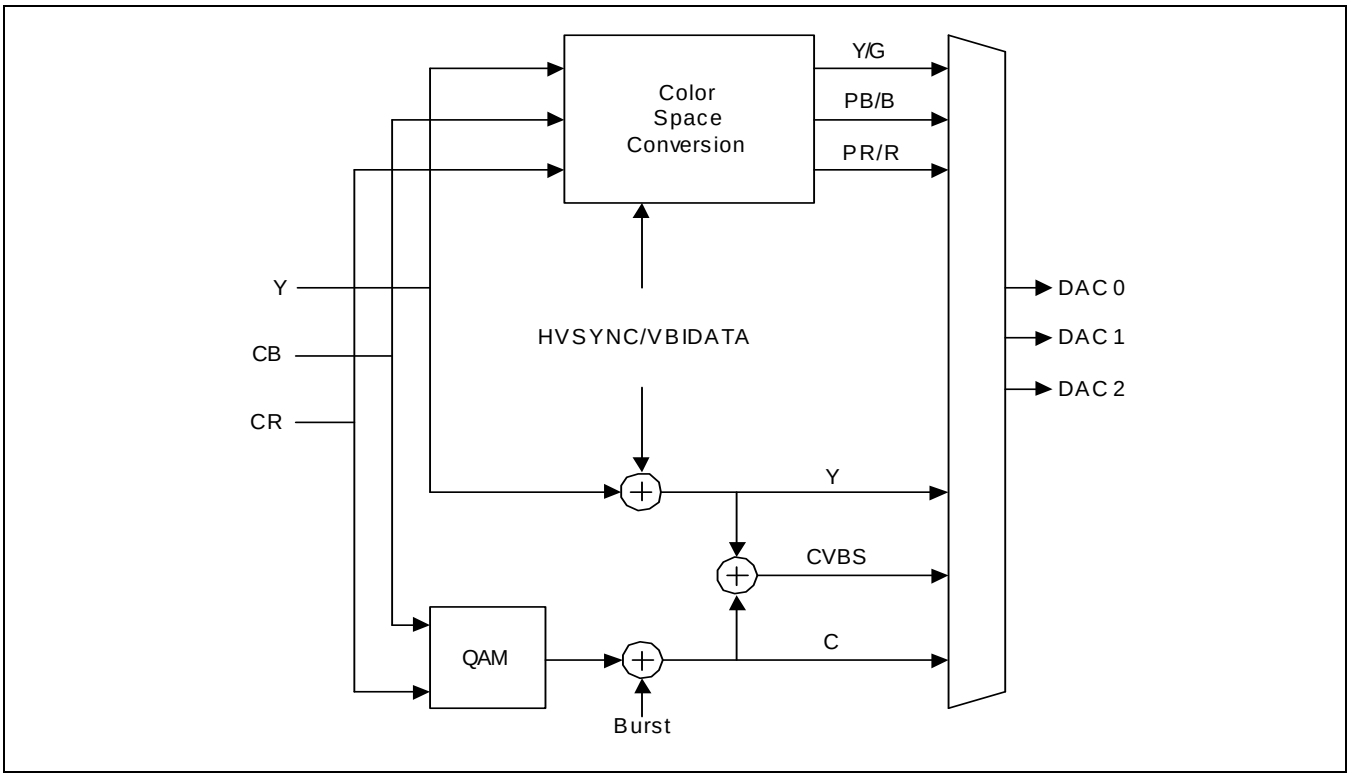


Figure 9.7-12 Data Flow of CVBS Sub-Module

The SEL_COMPONENT and SEL_CHAN0_OUTPUT… SEL_CHAN5_OUTPUT bits in SDO_CONFIG register is used for output selection.

The Figures 9.7-13 and 9.7-14 depict the typical CVBS waveforms generated. They show that the setup level and the ratios of video scale to sync depth are different according to video standards. This is the same for YPbPr/RGB outputs except. The setup level and video-to-sync ratio are controlled by CSETUP, CSYNC, VSETUP and VSYNC bits in SDO_SCALE registers. Note that the configuration of setup level and video-to-sync ratio in our implementation are set regardless of video standards and output format.

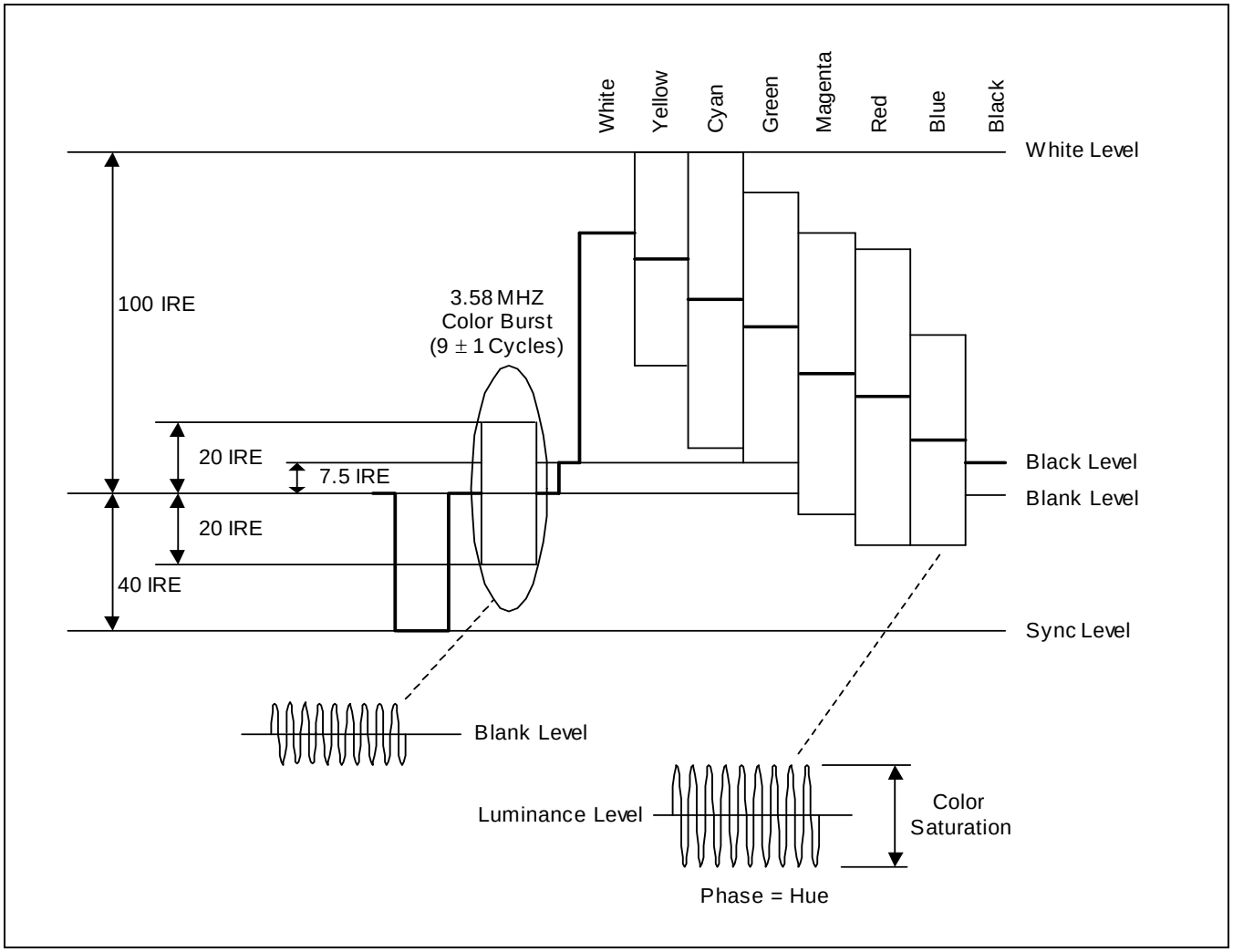


Figure 9.7-13 NTSC (M) Composite Video Signal with 75% Color Bars

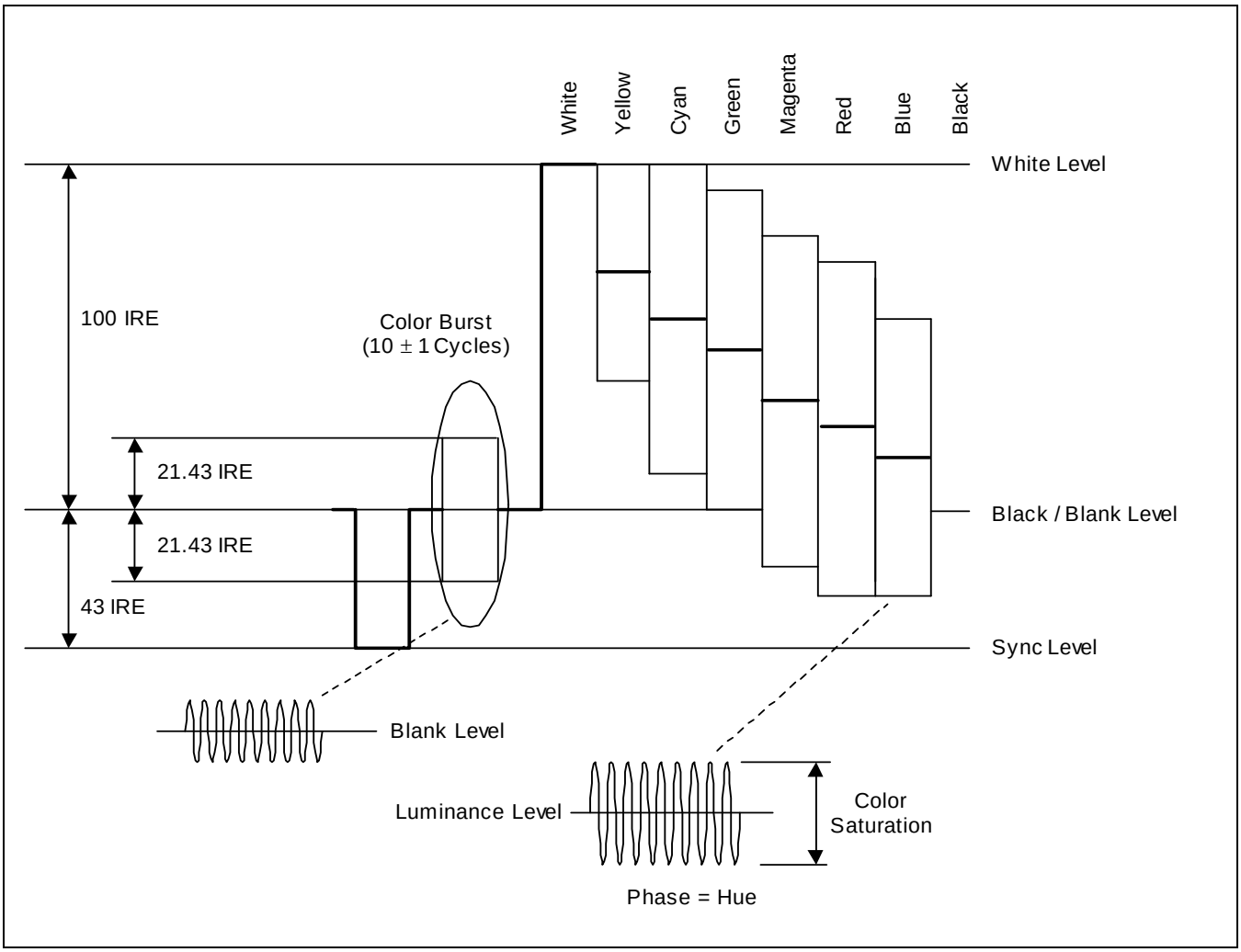


Figure 9.7-14 PAL (BGHIDNc) Composite Video Signal with 75% Color Bars

3.6 ILLEGAL COLOR COMPENSATION (CVBS MODULE)

The CVBS sub-module also supports color compensation for illegal RGB data. Video data are usually processed in the YCbCr coordinates. At the result of processing such as filtering and scaling, the values of YCbCr can exceed their nominal range and becomes to have ill-defined values when they are transformed into RGB coordinates. This causes unwanted artifacts at display. The Figure 9.7.15 illustrates the relation between YCbCr coordinates and RGB coordinates:

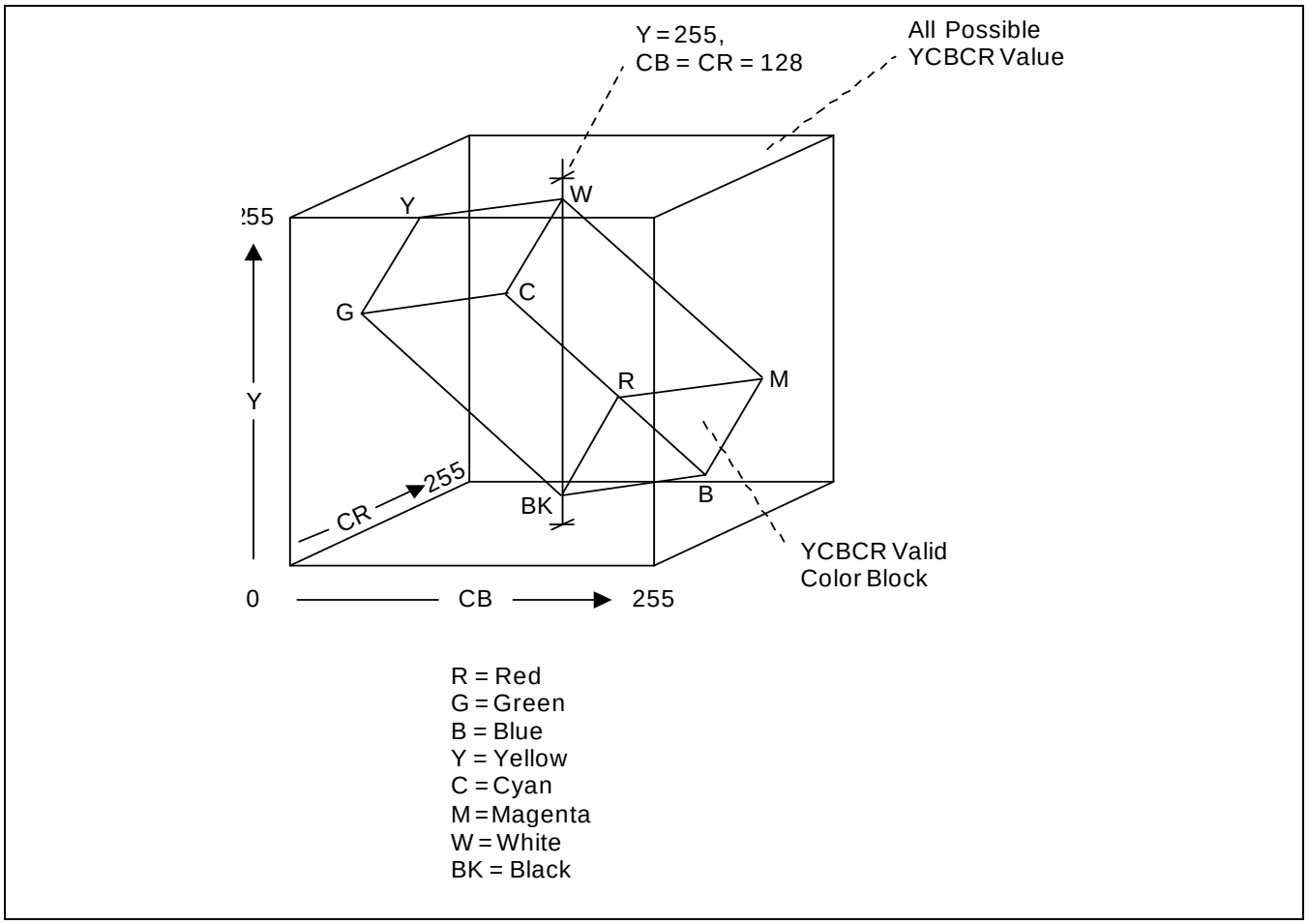


Figure 9.7-15 Color Cube Comparison

If YCbCr data outside of the RGB cube occurs, the CVBS sub-module compensates the value so that the result falls within the RGB cube. A constant luma and constant hue approach is used for this compensation. The luminance Y is not altered while the chrominance Cb and Cr are limited to the maximum valid values having the same hue as the invalid color prior to limiting. The SDO_RGB_CC register controls the size of RGB cube that determine color compensation range. Meanwhile, if illegal YCbCr data are transformed to CVBS or S-Video, there might be an overflow which exceeds DAC conversion range. The CVBS sub-module also compensates this error. The SDO_CVBS_CC_Y1, SDO_CVBS_CC_Y2, and SDO_CVBS_CC_C registers are used for the color compensation for CVBS output and the SDO_YC_CC_Y1, SDO_YC_CC_Y2, and SDO_YC_CC_C registers are used for the compensation for S-video output, respectively.

3.7 OVERSAMPLING & DAC COMPENSATION FILTER (OSF MODULE)

The TVOUT module provides 2x or 4X oversampling filter prior to DAC in order to ease the design of analog anti-image filter circuitry. For interlaced scanning case whose sample rate is 13.5MHz, four times oversampling provides the final output sampling rate as 54Msps (samples per second). For progressive scanning with double sample rate, 27MHz, OSF module performs two times oversampling with the result of also 54Msps.
The DAC has high frequency attenuation which comes from the sin(x)/x characteristic of sample-and-hold nature and additionally may have more attenuation by some reasons. As OSF has enough number of taps with programmable coefficients, compensation of such attenuation is done by making the filter response have a boost at high frequency.
OSF operates in different ways according to the oversampling ratio; the number of taps and the meanings of the coefficient registers are different for each case.

1. 4x oversampling case: 13.5Msps interlaced mode. It operates as 4-polyphase 95-tap FIR filter. All the coefficient registers (**SDO_OSFC00_0~SDO_OSFC23_0<DAC #0>, SDO_OSFC00_1~SDO_OSFC23_1<DAC #1>, SDO_OSFC00_2~SDO_OSFC23_2<DAC #2>**) are used.
2. 2x oversampling case: 27Msps progressive mode. It operates as 2-polyphase 47-tap FIR filter. Half of the filter coefficient registers (**SDO_OSFC12_0~SDO_OSFC23_0<DAC #0>, SDO_OSFC12_0~SDO_OSFC23_0<DAC #1>, SDO_OSFC12_2~SDO_OSFC23_2<DAC #2>,**) are used. The remaining values should be set to zero.

The target filter should be a center symmetric, that is,

Let h(i) a filter coefficient whose index is i.
N is odd number. (For 4x, N = 95 and for 2x, N=47)
h(i) = h(N-i) for i=0~[N/2]-1, where [] means rounding to the nearest integers towards zero.

Table 9.7-4 Over-sampling Filter Coefficients Configuration

SDO Register Name(_0,1,2)	Registers (_0, 1, 2)	4x Case Meaning (N=95)	2x Case Meaning (N=47)	Number of Bits (Including Sign Bit)
SDO_OSFC00	osf_coef00	(h[0]+h[2])/2 = (h[94]+h[92])/2	0	8
	osf_coef01	h[1] = h[93]	0	8
SDO_OSFC01	osf_coef02	(h[0]-h[2])/2 = (h[94]–h[92])/2	0	8
	osf_coef03	h[3] = h[91]	0	8
SDO_OSFC02	osf_coef04	(h[4]+h[6])/2 = (h[90]+h[88])/2	0	8
	osf_coef05	h[5] = h[89]	0	8
SDO_OSFC03	osf_coef06	(h[4]-h[6])/2 = (h[90]–h[88])/2	0	8
	osf_coef07	h[7] = h[87]	0	8
SDO_OSFC04	osf_coef08	(h[8]+h[10])/2 = (h[86]+h[84])/2	0	8
	osf_coef09	h[9] = h[85]	0	8
SDO_OSFC05	osf_coef10	(h[8]-h[10])/2 = (h[86]-h[84])/2	0	8
	osf_coef11	h[11] = h[83]	0	8
SDO_OSFC06	osf_coef12	(h[12]+h[14])/2= (h[82]+h[80])/2	0	9
	osf_coef13	h[13] = h[81]	0	9
SDO_OSFC07	osf_coef14	(h[12]-h[14])/2= (h[82]-h[80])/2	0	9
	osf_coef15	h[15] = h[79]	0	9
SDO_OSFC08	osf_coef16	(h[16]+h[18])/2= (h[78]+h[76])/2	0	9
	osf_coef17	h[17] = h[77]	0	9
SDO_OSFC09	osf_coef18	(h[16]-h[18])/2 = (h[78]-h[76])/2	0	9
	osf_coef19	h[19] = h[75]	0	9
SDO_OSFC10	osf_coef20	(h[20]+h[22])/2 = (h[74]+h[72])/2	0	9
	osf_coef21	h[21] = h[73]	0	9
SDO_OSFC11	osf_coef22	(h[20]-h[22])/2 = (h[74]-h[72])/2	0	9
	osf_coef23	h[23] = h[71]	0	9
SDO_OSFC12	osf_coef24	(h[24]+h[26])/2 = (h[70]+h[68])/2	h[0] = h[46]	10
	osf_coef25	h[25] = h[69]	h[1] = h[45]	10
SDO_OSFC13	osf_coef26	(h[24]-h[26])/2 = (h[70]-h[68])/2	h[2] = h[44]	10
	osf_coef27	h[27] = h[67]	h[3] = h[43]	10
SDO_OSFC14	osf_coef28	(h[28]+h[30])/2 = (h[66]+h[64])/2	h[4] = h[42]	10
	osf_coef29	h[29] = h[65]	h[5] = h[41]	10
SDO_OSFC15	osf_coef30	(h[28]-h[30])/2 = (h[66]-h[64])/2	h[6] = h[40]	10
	osf_coef31	h[31] = h[63]	h[7] = h[39]	10
SDO_OSFC16	osf_coef32	(h[32]+h[34])/2 = (h[62]+h[60])/2	h[8] = h[38]	10
	osf_coef33	h[33] = h[61]	h[9] = h[37]	10

SDO Register Name(_0,1,2)	Registers (_0, 1, 2)	4x Case Meaning (N=95)	2x Case Meaning (N=47)	Number of Bits (Including Sign Bit)
SDO_OSFC17	osf_coef34	(h[32]-h[34])/2 = (h[62]-h[60])/2	h[10] = h[36]	10
	osf_coef35	h[35] = h[59]	h[11] = h[35]	10
SDO_OSFC18	osf_coef36	(h[36]+h[38])/2 = (h[58]+h[56])/2	h[12] = h[34]	10
	osf_coef37	h[37] = h[57]	h[13] = h[33]	10
SDO_OSFC19	osf_coef38	(h[36]-h[38])/2 = (h[58]-h[56])/2	h[14] = h[32]	10
	osf_coef39	h[39] = h[55]	h[15] = h[31]	10
SDO_OSFC20	osf_coef40	(h[40]+h[42])/2 = (h[54]+h[52])/2	h[16] = h[30]	11
	osf_coef41	h[41] = h[53]	h[17] = h[29]	11
SDO_OSFC21	osf_coef42	(h[40]-h[42])/2 = (h[54]-h[52])/2	h[18] = h[28]	11
	osf_coef43	h[43] = h[51]	h[19] = h[27]	11
SDO_OSFC22	osf_coef44	(h[44]+h[46])/2 = (h[50]+h[48])/2	h[20] = h[26]	12
	osf_coef45	h[45] = h[49]	h[21] = h[25]	12
SDO_OSFC23	osf_coef46	(h[44]–h[46])/2 = (h[50]-h[48])/2	h[22] = h[24]	12
	osf_coef47	h[47]	h[23]	12

3.8 BLACK BURST GENERATION FOR DAC CROSSTALK TEST

In the three-channel video DACs, the signal of one channel may leak to others so that some part of it may be detected at the other channel outputs. Such interference is called crosstalk. In order to measure them, the SDO provides a special test mode – Black burst test mode. In this mode, one selected DAC channel generates normal CVBS signal and the remaining two channels do black burst signal. The black burst signal is the same as CVBS but does not contain any active video information. By doing that, the measured any non-black signal during active video of black burst channel is the amount of crosstalk.

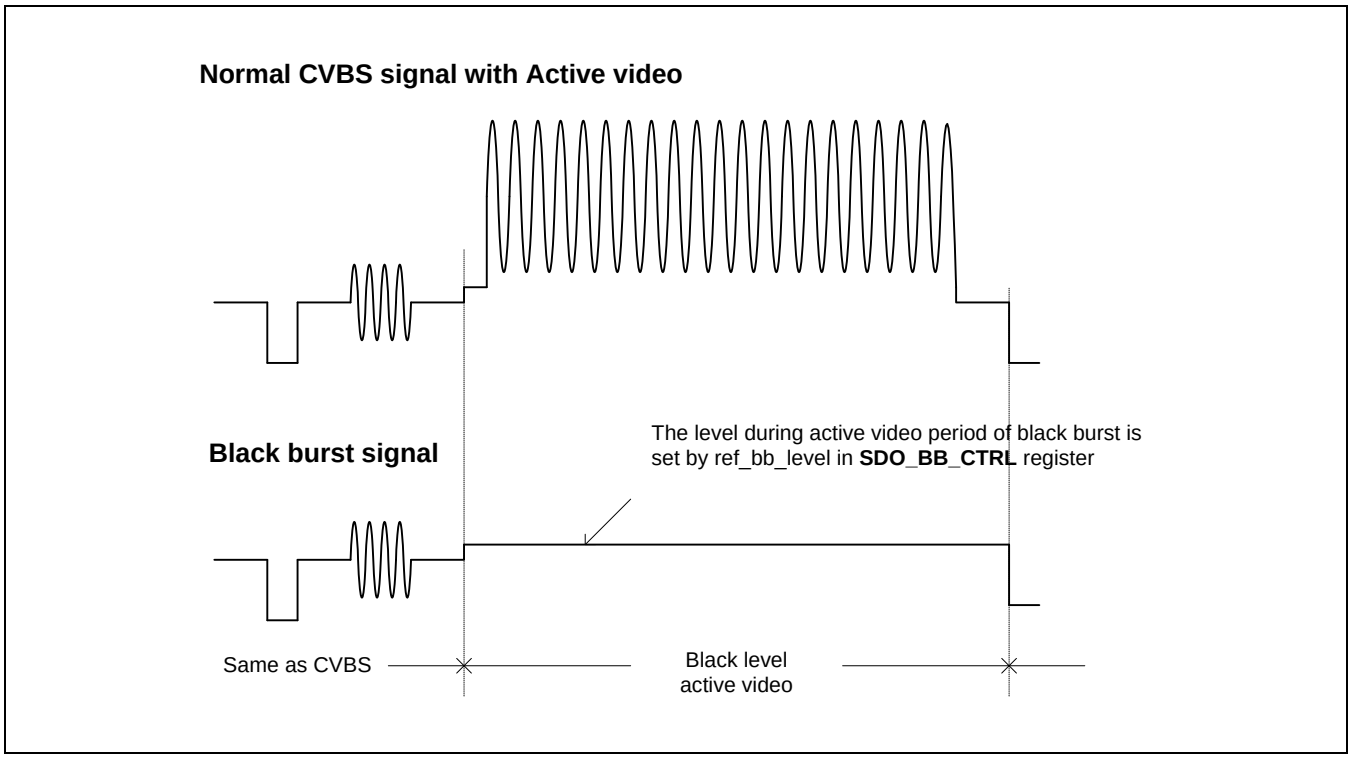


Figure 9.7-16 Black Burst Signal

Set *sel_bb_chan* bits in **SDO_BB_CTRL** register to select test channel as shown in the following figure.

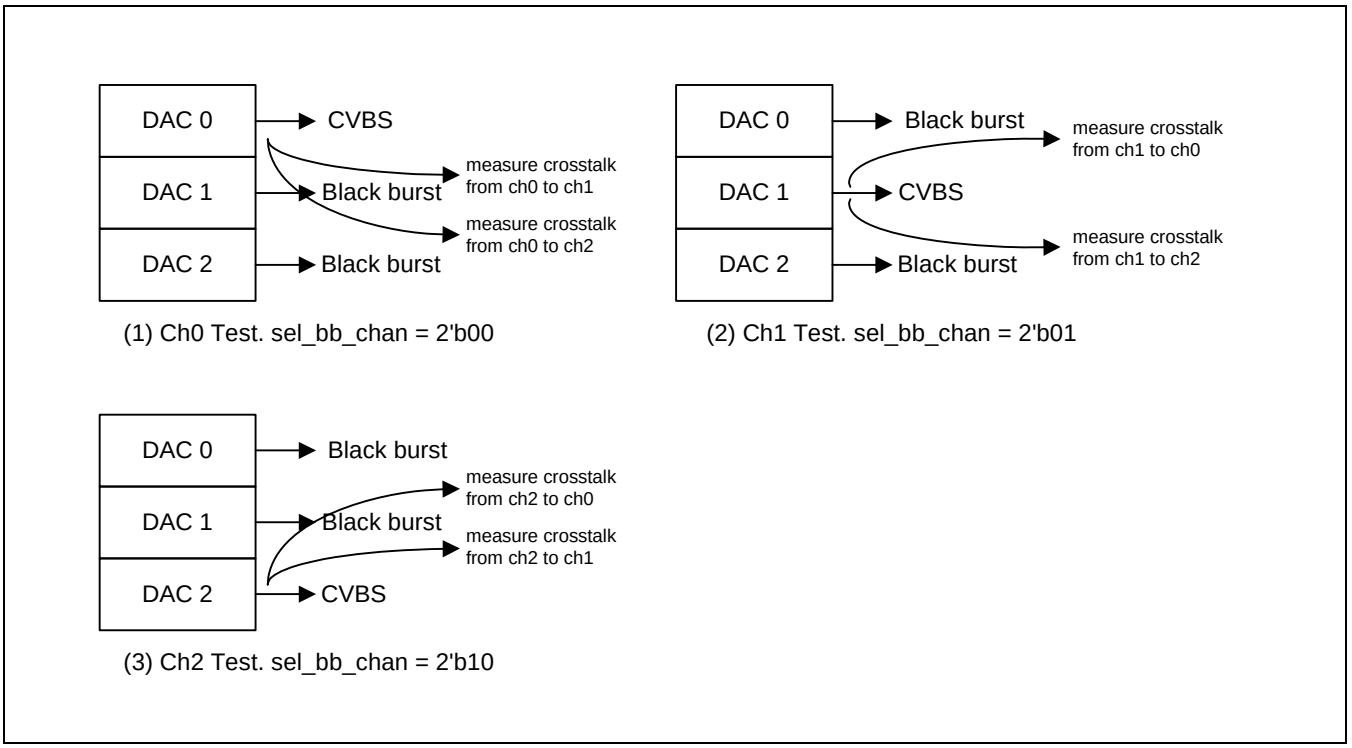
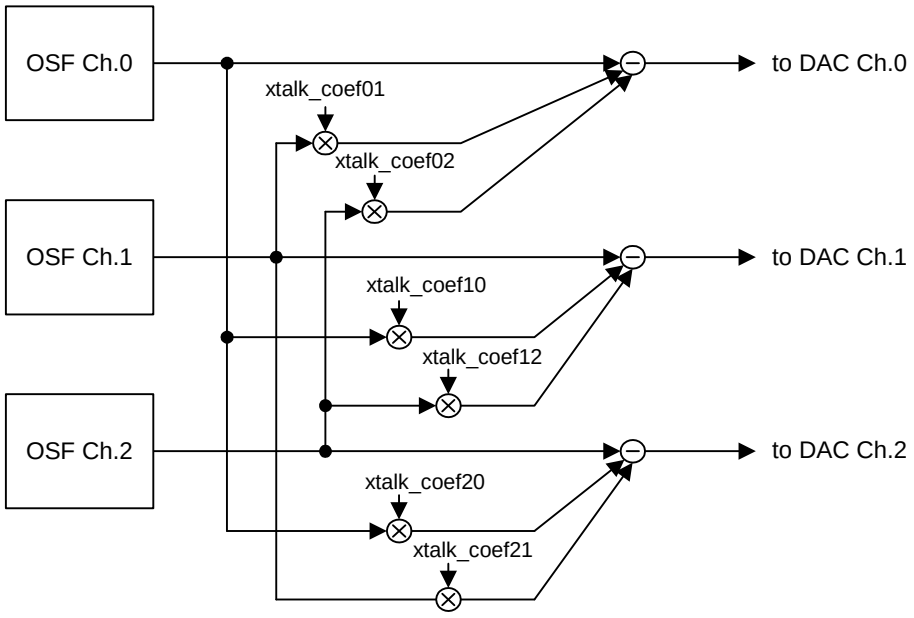


Figure 9.7-17 Crosstalk Measurement Using Black Burst Generation

3.9 CROSSTALK CANCELLATION

As mentioned in the previous section, the three-channel video DACs may have noticeable crosstalk between each other channel. In that case, if the transfer function of the crosstalk is impulse-like, then the crosstalk can be cancelled by subtraction of other channels with gains. The following figure shows crosstalk cancellation network. Each coefficient in the figure is programmable.



$$DACch\ 0 = OSFch\ 0 - xtalk\ \_\ coef\ 01 \cdot OSFch\ 1 - xtalk\ \_\ coef\ 02 \cdot OSFch\ 2$$

$$DACch\ 1 = OSFch\ 1 - xtalk\ \_\ coef\ 10 \cdot OSFch\ 0 - xtalk\ \_\ coef\ 12 \cdot OSFch\ 2$$

$$DACch\ 2 = OSFch\ 2 - xtalk\ \_\ coef\ 20 \cdot OSFch\ 0 - xalk\ \_\ coef\ 21 \cdot OSFch\ 1$$

Figure 9.7-18 Crosstalk Cancellation Network and Equations

3.10 TVOUT CALIBRATION FOR REDUCTION OF GAIN VARIATION AND BETTER CHANNEL BLANACE

Usually, video DAC experiences chip-to-chip variation in gain. This variation affects DAC channel balance if the variation occurs independently among each channel. The error in channel balance causes chrominance hue error when component YPbPr signal is output.

The programmability of digital gain in each video channel in TVOUT is used for gain calibration, together with the use of e-fused CHIPID bits. The gain errors of each sample are recorded at ATE test and the information on the error are e-fused into CHIPID bits at mass production.

In S5L8720, 15 CHIPID bits are assigned for the calibration - 5 CHIPID bits per channel. Please refer to DAC0_CAL, DAC1_CAL, and DAC2_CAL bits in the CHIPIDL register. For digital gain programming, SDO_Y_G, SDO_PB_B, SDO_PR_R, SDO_Y, SDO_C, SDO_CVBS registers are used according to the setting of SDO_CONFIG registers. Table 9.7- 5 shows the set of recommended gain values corresponding to each CHIPID bits. This recommendation guarantees that the gain variation of each video channel is within ± 0.4375% and thus channel balance mismatch within ± 0.875 %, which corresponds to ± 0.25˚ of chrominance hue error at the worst case.

Table 9.7- 5 Gain Settings for Each CHIPID Bits (Nominal Full Scale 1.3V @ 6.34k RSET and BGR VREF)

CHIPID Bits	TVOUT Gain
01111	X 2357/2048
01110	X 2334/2048
01101	X 2311/2048
01100	X 2288/2048
01011	X 2266/2048
01010	X 2244/2048
01001	X 2223/2048
01000	X 2202/2048
00111	X 2182/2048
00110	X 2161/2048
00101	X 2142/2048
00100	X 2122/2048
00011	X 2103/2048
00010	X 2084/2048
00001	X 2066/2048
00000	X 2048/2048
11111	X 2030/2048
11110	X 2013/2048
11101	X 1996/2048
11100	X 1979/2048
11011	X 1962/2048
11010	X 1946/2048
11001	X 1930/2048
11000	X 1914/2048
10111	X 1898/2048
10110	X 1883/2048
10101	X 1868/2048
10100	X 1853/2048
10011	X 1839/2048
10010	X 1824/2048
10001	X 1810/2048
10000	X 1796/2048

3.11 REGISTER CONTROL (CTRL MODULE)

The TVOUT module supports AHB+ slave bus interface for register control. All the registers are synchronized with system bus clock.

3.12 OUTPUT INTERFACE

TVOUT output data are directly feed to DAC. The output data are synchronized to video clock vclk. Note that output data transit with positive edge of video clock vclk. It implies that the two video clocks fed to TVOUT and DAC, respectively, are recommended to have opposite phase to each other.

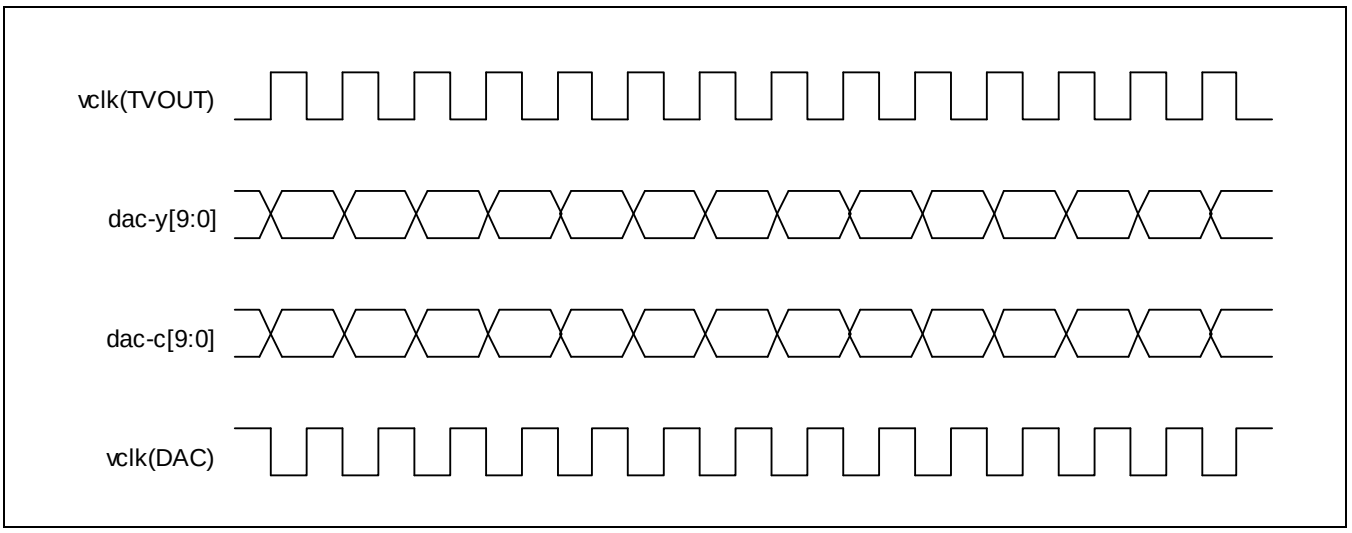


Figure 9.7-19 Output Interface of TVOUT Module

4 I/O DESCRIPTION

Function Signal	*I/O*	*Description*	*Pad*	*Type*
DAC_COMP	*Output*	*Compensation Pin. This is a compensation pin for the internal reference amplifier. A 0.1uF ceramic capacitor must be connected between COMP and AVDD30A.*	*XdacCOMP*	*Dedicated*
DAC_IREF	*Input*	*Full Scale Adjust control. The full-scale current drive on each of the output channels is determined by the value of a resistor RSET connected between this terminal and AVSS30A1.*	*XdacIREF*	*Dedicated*
DAC_VREF	*Input*	*Voltage reference for DAC. An Internal voltage reference of nominally 1.26V is provided. Can be driven with an external reference source.*	*XdacVREF*	*Dedicated*
DAC_OUT[0]	*Output*	*DAC current output.*	*XdacOUT[0]*	*Dedicated*
DAC_OUT[1]	*Output*	*DAC current output.*	*XdacOUT[1]*	*Dedicated*
DAC_OUT[2]	*Output*	*DAC current output.*	*XdacOUT[2]*	*Dedicated*

5 REGISTERS DESCRIPTION

Register	Address	R/W	Description	Reset Value
SDO_CLKCON	0xF000_0000	R/W	Clock Control Register	0x0000_0000
SDO_CONFIG	0xF000_0008	R/W	Video Standard Configuration Register	0x0024_2430
SDO_SCALE	0xF000_000C	R/W	Video Scale Configuration Register	0x0000_0006
SDO_SYNC	0xF000_0010	R/W	Video Sync Configuration Register	0x0000_0001
SDO_VBI	0xF000_0014	R/W	VBI Configuration Register	0x0007_77FF
SDO_SCALE_CH0	0xF000_001C	R/W	Scale Control Register for DAC Channel0	0x0000_0800
SDO_SCALE_CH1	0xF000_0020	R/W	Scale Control Register for DAC Channel1	0x0000_0800
SDO_SCALE_CH2	0xF000_0024	R/W	Scale Control Register for DAC Channel2	0x0000_0800
SDO_YCDELAY	0xF000_0034	R/W	Video Delay Control Register	0x0000_FA00
SDO_SCHLOCK	0xF000_0038	R/W	SCH Phase Control Register	0x0000_0000
SDO_DAC	0xF000_003C	R/W	DAC Configuration Register	0x0000_0000
SDO_FINFO	0xF000_0040	R	Status Register	0x0000_0002
SDO_Y0	0xF000_0044	R/W	Y- AAF 1'st and 23'th Coefficient (AAF : Anti-Aliasing Filter)	0x0000_0000
SDO_Y1	0xF000_0048	R/W	Y- AAF 2'nd and 22'th Coefficient	0x0000_0000
SDO_Y2	0xF000_004C	R/W	Y- AAF 3'rd and 21'th Coefficient	0x0000_0000
SDO_Y3	0xF000_0050	R/W	Y- AAF 4'th and 20'th Coefficient	0x0000_0000

Register	Address	R/W	Description	Reset Value
SDO_Y4	0xF000_0054	R/W	Y- AAF 5'th and 19'th Coefficient	0x0000_0000
SDO_Y5	0xF000_0058	R/W	Y- AAF 6'th and 18'th Coefficient	0x0000_0000
SDO_Y6	0xF000_005C	R/W	Y- AAF 7'th and 17'th Coefficient	0x0000_0000
SDO_Y7	0xF000_0060	R/W	Y- AAF 8'th and 16'th Coefficient	0x0000_0000
SDO_Y8	0xF000_0064	R/W	Y - AAF 9'th and 15'th Coefficient	0x0000_0000
SDO_Y9	0xF000_0068	R/W	Y- AAF 10'th and 14'th Coefficient	0x0000_0000
SDO_Y10	0xF000_006C	R/W	Y- AAF 11'th and 13'th Coefficient	0x0000_0000
SDO_Y11	0xF000_0070	R/W	Y- AAF 12'th Coefficient	0x0000_025D
SDO_CB0	0xF000_0080	R/W	CB- AAF 1'st and 23'th Coefficient	0x0000_0000
SDO_CB1	0xF000_0084	R/W	CB- AAF 2'nd and 22'th Coefficient	0x0000_0000
SDO_CB2	0xF000_0088	R/W	CB- AAF 3'rd and 21'th Coefficient	0x0000_0000
SDO_CB3	0xF000_008C	R/W	CB-AAF 4'th and 20'th Coefficient	0x0000_0000
SDO_CB4	0xF000_0090	R/W	CB- AAF 5'th and 19'th Coefficient	0x0000_0000
SDO_CB5	0xF000_0094	R/W	CB- AAF 6'th and 18'th Coefficient	0x0000_0001
SDO_CB6	0xF000_0098	R/W	CB- AAF 7'th and 17'th Coefficient	0x0000_0007
SDO_CB7	0xF000_009C	R/W	CB- AAF 8'th and 16'th Coefficient	0x0000_0014
SDO_CB8	0xF000_00A0	R/W	CB- AAF 9'th and 15'th Coefficient	0x0000_0028
SDO_CB9	0xF000_00A4	R/W	CB- AAF 10'th and 14'th Coefficient	0x0000_003F
SDO_CB10	0xF000_00A8	R/W	CB- AAF 11'th and 13'th Coefficient	0x0000_0052
SDO_CB11	0xF000_00AC	R/W	CB- AAF 12'th Coefficient	0x0000_005A
SDO_CR0	0xF000_00C0	R/W	CR- AAF 1'st and 23'th Coefficient	0x0000_0000
SDO_CR1	0xF000_00C4	R/W	CR- AAF 2'nd and 22'th Coefficient	0x0000_0000
SDO_CR2	0xF000_00C8	R/W	CR- AAF 3'rd and 21'th Coefficient	0x0000_0000
SDO_CR3	0xF000_00CC	R/W	CR-AAF 4'th and 20'th Coefficient	0x0000_0000
SDO_CR4	0xF000_00D0	R/W	CR- AAF 5'th and 19'th Coefficient	0x0000_0000
SDO_CR5	0xF000_00D4	R/W	CR- AAF 6'th and 18'th Coefficient	0x0000_0001
SDO_CR6	0xF000_00D8	R/W	CR- AAF 7'th and 17'th Coefficient	0x0000_0009
SDO_CR7	0xF000_00DC	R/W	CR- AAF 8'th and 16'th Coefficient	0x0000_001C
SDO_CR8	0xF000_00E0	R/W	CR- AAF 9'th and 15'th Coefficient	0x0000_0039
SDO_CR9	0xF000_00E4	R/W	CR- AAF 10'th and 14'th Coefficient	0x0000_005A
SDO_CR10	0xF000_00E8	R/W	CR- AAF 11'th and 13'th Coefficient	0x0000_0074
SDO_CR11	0xF000_00EC	R/W	CR- AAF 12'th Coefficient	0x0000_007E
SDO_CCCON	0xF000_0180	R/W	Color Compensation On/ Off Control	0x0000_0000
SDO_YSCALE	0xF000_0184	R/W	Brightness Control for Y	0x0080_0000
SDO_CBSCALE	0xF000_0188	R/W	Hue/ Saturation Control for CB	0x0080_0000
SDO_CRSCALE	0xF000_018C	R/W	Hue/ Saturation Control for CR	0x0000_0080

Register	Address	R/W	Description	Reset Value
SDO_CB_CR_OFFSET	0xF000_0190	R/W	Hue/ Sat Offset Control for CB/CR	0x0000_0000
SDO_RGB_CC	0xF000_0194	R/W	Color Compensation of RGB Output	0x0000_EB10
SDO_CVBS_CC_Y1	0xF000_0198	R/W	Color Compensation of CVBS Output	0x0200_0000
SDO_CVBS_CC_Y2	0xF000_019C	R/W	Color Compensation of CVBS Output	0x03FF_0200
SDO_CVBS_CC_C	0xF000_01A0	R/W	Color Compensation of CVBS Output	0x0000_01FF
SDO_YC_CC_Y	0xF000_01A4	R/W	Color Compensation of S-video Output	0x03FF_0000
SDO_YC_CC_C	0xF000_01A8	R/W	Color Compensation of S-video Output	0x0000_01FF
SDO_CSC_525_PORCH	0xF000_01B0	R/W	Porch Position Control of CSC in 525 Line	0x008A_0359
SDO_CSC_625_PORCH	0xF000_01B4	R/W	Porch Position Control of CSC in 625 Line	0x0096_035C
SDO_RGBSYNC	0xF000_01C0	R/W	VESA RGB Sync Control Register	0x0000_0000
SDO_OSFC00_0	0xF000_0200	R/W	OverSampling Filter (OSF) Coefficient 1 & 0. of channel #0	0x00FD_00FE
SDO_OSFC01_0	0xF000_0204	R/W	OSF Coefficient 3 & 2 of Channel #0	0x0000_0000
SDO_OSFC02_0	0xF000_0208	R/W	OSF Coefficient 5 & 4 of Channel #0	0x0005_0004
SDO_OSFC03_0	0xF000_020C	R/W	OSF Coefficient 7 & 6 of Channel #0	0x0000_00FF
SDO_OSFC04_0	0xF000_0210	R/W	OSF Coefficient 9 & 8 of Channel #0	0x00F7_00FA
SDO_OSFC05_0	0xF000_0214	R/W	OSF Coefficient 11 & 10 of Channel #0	0x0000_0001
SDO_OSFC06_0	0xF000_0218	R/W	OSF Coefficient 13 & 12 of Channel #0	0x000E_000A
SDO_OSFC07_0	0xF000_021C	R/W	OSF Coefficient 15 & 14 of Channel #0	0x0000_01FF
SDO_OSFC08_0	0xF000_0220	R/W	OSF Coefficient 17 & 16 of Channel #0	0x01EC_01F2
SDO_OSFC09_0	0xF000_0224	R/W	OSF Coefficient 19 & 18 of Channel #0	0x0000_0001
SDO_OSFC10_0	0xF000_0228	R/W	OSF Coefficient 21 & 20 of Channel #0	0x001D_0014
SDO_OSFC11_0	0xF000_022C	R/W	OSF Coefficient 23 & 22 of Channel #0	0x0000_01FE
SDO_OSFC12_0	0xF000_0230	R/W	OSF Coefficient 25 & 24 of Channel #0	0x03D8_03E4
SDO_OSFC13_0	0xF000_0234	R/W	OSF Coefficient 27 & 26 of Channel #0	0x0000_0002
SDO_OSFC14_0	0xF000_0238	R/W	OSF Coefficient 29 & 28 of Channel #0	0x0038_0028
SDO_OSFC15_0	0xF000_023C	R/W	OSF Coefficient 31 & 30 of Channel #0	0x0000_03FD
SDO_OSFC16_0	0xF000_0240	R/W	OSF Coefficient 33 & 32 of Channel #0	0x03B0_03C7
SDO_OSFC17_0	0xF000_0244	R/W	OSF Coefficient 35 & 34 of Channel #0	0x0000_0005
SDO_OSFC18_0	0xF000_0248	R/W	OSF Coefficient 37 & 36 of Channel #0	0x0079_0056
SDO_OSFC19_0	0xF000_024C	R/W	OSF Coefficient 39 & 38 of Channel #0	0x0000_03F6
SDO_OSFC20_0	0xF000_0250	R/W	OSF Coefficient 41 & 40 of Channel #0	0x072C_0766
SDO_OSFC21_0	0xF000_0254	R/W	OSF Coefficient 43 & 42 of Channel #0	0x0000_001B
SDO_OSFC22_0	0xF000_0258	R/W	OSF Coefficient 45 & 44 of Channel #0	0x028B_0265
SDO_OSFC23_0	0xF000_025C	R/W	OSF Coefficient 47 & 46 of Channel #0	0x0400_0ECC

Register	Address	R/W	Description	Reset Value
SDO_XTALK0	0xF000_0260	R/W	Crosstalk Cancel Coefficient for Ch.0	0x0000_0000
SDO_XTALK1	0xF000_0264	R/W	Crosstalk Cancel Coefficient for Ch.1	0x0000_0000
SDO_XTALK2	0xF000_0268	R/W	Crosstalk Cancel Coefficient for Ch.2	0x0000_0000
SDO_BB_CTRL	0xF000_026C	R/W	Blackburst Test Control	0x0001_1A00
SDO_IRQ	0xF000_0280	R/W	Interrupt Request Register	0x0000_0000
SDO_IRQMASK	0xF000_0284	R/W	Interrupt Request Enable Register	0x0000_0000
SDO_OSFC00_1	0xF000_02C0	R/W	OverSampling Filter (OSF) Coefficient 1 & 0. of Channel #1	0x00FD_00FE
SDO_OSFC01_1	0xF000_02C4	R/W	OSF Coefficient 3 & 2 of Channel #1	0x0000_0000
SDO_OSFC02_1	0xF000_02C8	R/W	OSF Coefficient 5 & 4 of Channel #1	0x0005_0004
SDO_OSFC03_1	0xF000_02CC	R/W	OSF Coefficient 7 & 6 of Channel #1	0x0000_00FF
SDO_OSFC04_1	0xF000_02D0	R/W	OSF Coefficient 9 & 8 of Channel #1	0x00F7_00FA
SDO_OSFC05_1	0xF000_02D4	R/W	OSF Coefficient 11 & 10 of Channel #1	0x0000_0001
SDO_OSFC06_1	0xF000_02D8	R/W	OSF Coefficient 13 & 12 of Channel #1	0x000E_000A
SDO_OSFC07_1	0xF000_02DC	R/W	OSF Coefficient 15 & 14 of Channel #1	0x0000_01FF
SDO_OSFC08_1	0xF000_02E0	R/W	OSF Coefficient 17 & 16 of Channel #1	0x01EC_01F2
SDO_OSFC09_1	0xF000_02E4	R/W	OSF Coefficient 19 & 18 of Channel #1	0x0000_0001
SDO_OSFC10_1	0xF000_02E8	R/W	OSF Coefficient 21 & 20 of Channel #1	0x001D_0014
SDO_OSFC11_1	0xF000_02EC	R/W	OSF Coefficient 23 & 22 of Channel #1	0x0000_01FE
SDO_OSFC12_1	0xF000_02E0	R/W	OSF Coefficient 25 & 24 of Channel #1	0x03D8_03E4
SDO_OSFC13_1	0xF000_02F4	R/W	OSF Coefficient 27 & 26 of Channel #1	0x0000_0002
SDO_OSFC14_1	0xF000_02F8	R/W	OSF Coefficient 29 & 28 of Channel #1	0x0038_0028
SDO_OSFC15_1	0xF000_02FC	R/W	OSF Coefficient 31 & 30 of Channel #1	0x0000_03FD
SDO_OSFC16_1	0xF000_0300	R/W	OSF Coefficient 33 & 32 of Channel #1	0x03B0_03C7
SDO_OSFC17_1	0xF000_0304	R/W	OSF Coefficient 35 & 34 of Channel #1	0x0000_0005
SDO_OSFC18_1	0xF000_0308	R/W	OSF Coefficient 37 & 36 of Channel #1	0x0079_0056
SDO_OSFC19_1	0xF000_030C	R/W	OSF Coefficient 39 & 38 of Channel #1	0x0000_03F6
SDO_OSFC20_1	0xF000_0310	R/W	OSF Coefficient 41 & 40 of Channel #1	0x072C_0766
SDO_OSFC21_1	0xF000_0314	R/W	OSF Coefficient 43 & 42 of Channel #1	0x0000_001B
SDO_OSFC22_1	0xF000_0318	R/W	OSF Coefficient 45 & 44 of Channel #1	0x028B_0265
SDO_OSFC23_1	0xF000_031C	R/W	OSF Coefficient 47 & 46 of Channel #1	0x0400_0ECC
SDO_OSFC00_2	0xF000_0320	R/W	OverSampling Filter (OSF) Coefficient 1 & 0. of Channel #2	0x00FD_00FE
SDO_OSFC01_2	0xF000_0324	R/W	OSF Coefficient 3 & 2 of Channel #2	0x0000_0000
SDO_OSFC02_2	0xF000_0328	R/W	OSF Coefficient 5 & 4 of Channel #2	0x0005_0004
SDO_OSFC03_2	0xF000_032C	R/W	OSF Coefficient 7 & 6 of Channel #2	0x0000_00FF
SDO_OSFC04_2	0xF000_0330	R/W	OSF Coefficient 9 & 8 of Channel #2	0x00F7_00FA

Register	Address	R/W	Description	Reset Value
SDO_OSFC05_2	0xF000_0334	R/W	OSF Coefficient 11 & 10 of Channel #2	0x0000_0001
SDO_OSFC06_2	0xF000_0338	R/W	OSF Coefficient 13 & 12 of Channel #2	0x000E_000A
SDO_OSFC07_2	0xF000_033C	R/W	OSF Coefficient 15 & 14 of Channel #2	0x0000_01FF
SDO_OSFC08_2	0xF000_0340	R/W	OSF Coefficient 17 & 16 of Channel #2	0x01EC_01F2
SDO_OSFC09_2	0xF000_0344	R/W	OSF Coefficient 19 & 18 of Channel #2	0x0000_0001
SDO_OSFC10_2	0xF000_0348	R/W	OSF Coefficient 21 & 20 of Channel #2	0x001D_0014
SDO_OSFC11_2	0xF000_034C	R/W	OSF Coefficient 23 & 22 of Channel #2	0x0000_01FE
SDO_OSFC12_2	0xF000_0350	R/W	OSF Coefficient 25 & 24 of Channel #2	0x03D8_03E4
SDO_OSFC13_2	0xF000_0354	R/W	OSF Coefficient 27 & 26 of Channel #2	0x0000_0002
SDO_OSFC14_2	0xF000_0358	R/W	OSF Coefficient 29 & 28 of Channel #2	0x0038_0028
SDO_OSFC15_2	0xF000_035C	R/W	OSF Coefficient 31 & 30 of Channel #2	0x0000_03FD
SDO_OSFC16_2	0xF000_0360	R/W	OSF Coefficient 33 & 32 of Channel #2	0x03B0_03C7
SDO_OSFC17_2	0xF000_0364	R/W	OSF Coefficient 35 & 34 of Channel #2	0x0000_0005
SDO_OSFC18_2	0xF000_0368	R/W	OSF Coefficient 37 & 36 of Channel #2	0x0079_0056
SDO_OSFC19_2	0xF000_036C	R/W	OSF Coefficient 39 & 38 of Channel #2	0x0000_03F6
SDO_OSFC20_2	0xF000_0370	R/W	OSF Coefficient 41 & 40 of Channel #2	0x072C_0766
SDO_OSFC21_2	0xF000_0374	R/W	OSF Coefficient 43 & 42 of Channel #2	0x0000_001B
SDO_OSFC22_2	0xF000_0378	R/W	OSF Coefficient 45 & 44 of Channel #2	0x028B_0265
SDO_OSFC23_2	0xF000_037C	R/W	OSF Coefficient 47 & 46 of Channel #2	0x0400_0ECC
SDO_ARMCC	0xF000_03C0	R/W	Closed Caption Data Register	0x0000_0000
SDO_ARMWSS525	0xF000_03C4	R/W	WSS 525 Data Register	0x0000_0000
SDO_ARMWSS625	0xF000_03C8	R/W	WSS 625 Data Register	0x0000_0000
SDO_ARMCGMS525	0xF000_03CC	R/W	CGMS-A 525 Data Register	0x0000_0000
SDO_ARMCGMS625	0xF000_03D4	R/W	CGMS-A 625 Data Register	0x0000_0000
SDO_VERSION	0xF000_03D8	R	TVOUT Version Number Read Register	0x0000_000C

5.1 SDO CLOCK CONTROL REGISTER (SDO_CLKCON, R/W, ADDRESS = 0XF000_ 0000)

SDO_CLKCON	Bit	Description	Reset Value
Reserved	[31:5]	Reserved, read as zero, do not modify	0
SDO software reset	[4]	This bit controls software reset of TVOUT. Software reset is active high signal. 0 = No reset 1 = Software reset enable	0
Reserved	[3:2]	Reserved, read as zero, do not modify	
SDO clock down ready (read only)	[1]	Indicates whether host controller can stop the clock for the TVOUT. 0 = Clock-down not ready 1 = Clock-down ready Normally this bit is 0. After SDO_CLKCON [0]bit is 0, if the internal line counter and pixel counter are 0 (just before starting line 1), this bit will be 1.	0
SDO clock on	[0]	This bit determines run/ stop mode of TVOUT. 0: TVOUT clock off. TVOUT requests for clock down to host controller. If SDO is ready for clock down, SDO_CLKCON [1] bit will be 1. The host controller should stop the clock for the TVOUT after that. 1 = TVOUT clock on. TVOUT starts running. NOTE : The SFRs of Video Processor and Image Mixer is updated by Vertical Sync of TVENC's Timing Generator. So, those are configured before this bit is enabled. The sequence of enabling TVSS is following as: "VP -> MIXER -> TVENC". Also, because of the same reason, the disabling sequence is following as : "VP -> MIXER -> TVNEC".	0

5.2 SDO VIDEO STANDARD CONFIGURATION REGISTER (SDO_CONFIG, R/W, ADDRESS = 0XF000_0008)

SDO_CONFIG	Bit	Description	Reset Value
Reserved	[31:22]	Reserved, read as zero, do not modify	0
Selection of Video Mux for DAC 2	[21:20]	If bit [5] is set to YPbPr, 0 = Y signal 1 = Pb signal 2 = Pr signal Otherwise, 0 = G signal 1 = B signal 2 = R signal	2
Selection of Video Mux for DAC 1	[19:18]	If bit [5] is set to YpbPr, 0 = Y signal 1 = Pb signal 2 = Pr signal Otherwise, 0 = G signal 1 = B signal 2 = R signal	1
Selection of Video Mux for DAC 0	[17:16]	If bit [5] is set to YpbPr, 0 = Y signa 1 = Pb signal 2 = Pr signal Otherwise, 0 = G signal 1 = B signal 2 = R signal	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
Selection of Video Mux for DAC 2	[13:12]	0 = CVBS signal 1 = Y signal 2 = C signal	2
Selection of Video Mux for DAC 1	[11:10]	0 = CVBS signal 1 = Y signal 2 = C signal	1
Selection of Video Mux for DAC 0	[9:8]	0 = CVBS signal 1 = Y signal 2 = C signal	0
Reserved	[7]	Reserved, read as zero, do not modify	0

SDO_CONFIG	Bit	Description	Reset Value
Selection of Video Mux for DAC 0, 1, and 2	[6]	0 = Setting of bits [13:8] is valid (composite signals) 1 = Setting of bits [21:16] is valid (component signals)	0
Selection of Video Type for Component Outputs	[5]	0 = RGB 1 = YPbPr This setting is valid if bit [6] is set to be component.	1
Selection of Video Scan for Component Output	[4]	0 = Interlaced 1 = Progressive The scan rate of video output is not changed if bit [6] is set to be component. But the setting of this bit has an impact on pixel requests. If they are working in interlaced scan mode, this bit should be also set Interlaced.	1
Selection of Video Standard	[3:0]	0 = NTSC (M) 1 = PAL (M) 2 = PAL (BGHID) 3 = PAL (N) 4 = PAL (Nc) 8 = NTSC 4.43 9 = PAL 60	0

5.3 SDO VIDEO SCALE CONFIGURATION REGISTER (SDO_SCALE, R/W, ADDRESS = 0XF000_000C)

SDO_SCALE	Bit	Description	Reset Value
Reserved	[31:4]	Reserved, read as zero, do not modify	0
Setup Level Selection for YpbPr / RGB Component	[3]	0 = 0 IRE 1 = 7.5 IRE This setting is valid if bit[6] of SDO_CONFIG register is set to component.	0
Video-to-Sync Ratio Selection for YpbPr / RGB Component	[2]	0 = 10:4 1 = 7:3 This setting is valid if bit[6] of SDO_CONFIG register is set to component.	1
Setup Level Selection for Composite / S-Video	[1]	0 = 0 IRE 1 = 7.5 IRE This setting is valid if bit[6] of SDO_CONFIG register is set to composite.	1
Video-to-Sync Ratio Selection for Composite / S-Video	[0]	0 = 10:4 1 = 7:3 This setting is valid if bit[6] of SDO_CONFIG register is set to composite.	0

NOTE: The Table 9.7-6 and 9.7--7 according to the value of bit [3:2]

Table 9.7-6 Video Scale According to Video Format (YpbPr)

Format	Output Signal	Signal Amplitudes (volts)	Notes
SMPTE EBU N10	Y	+ 0.700	0% setup on Y 100% saturation three wire = (Y + sync), (R'-Y), (B'-Y)
	sync	- 0.300	
	R'-Y, B'-Y	± 0.350	
525-Line Betacam(1)	Y	+ 0.714	7.5% setup on Y 100% saturation three wire = (Y + sync), (R'-Y), (B'-Y)
	sync	- 0.286	
	R'-Y, B'-Y	± 0.467	
625-Line Betacam(1)	Y	+ 0.700	0% setup on Y 100% saturation three wire = (Y + sync), (R'-Y), (B'-Y)
	sync	- 0.300	
	R'-Y, B'-Y	± 0.350	
525-Line MII(2)	Y	+ 0.700	7.5% setup on Y 100% saturation three wire = (Y + sync), (R'-Y), (B'-Y)
	sync	- 0.300	
	R'-Y, B'-Y	± 0.324	
625-Line MII(2)	Y	+ 0.700	0% setup on Y 100% saturation three wire = (Y + sync), (R'-Y), (B'-Y)
	sync	- 0.300	
	R'-Y, B'-Y	± 0.350	

NOTE: 1.Trademark of Sony Corporation.
2.Trademark of Matsushita Corporation.

Table 9.7-7 Video Scale According to Video Format (RGB)

Format	Output Signal	Signal Amplitudes (volts)	Notes
SMPTE, EBU N10	G', B', R'	+ 0.700	0% setup on G', B', and R' 100% saturation three wire = (G' + sync), B', R'
	sync	- 0.300	
NTSC (setup)	G', B', R'	+ 0.714	7.5% setup on G', B', and R' 100% saturation three wire = (G' + sync), B', R'
	sync	- 0.286	
NTSC (no setup)	G', B', R'	+ 0.714	0% setup on G', B', and R' 100% saturation three wire = (G' + sync), B', R'
	sync	- 0.286	
525-Line MII(NOTE)	G', B', R'	+ 0.700	7.5% setup on G', B', and R' 100% saturation three wire = (G' + sync), B', R'

NOTE: Trademark of Matsushita Corporation.

5.4 SDO VIDEO SYNC CONFIGURATION REGISTER (SDO_SYNC, R/W, ADDRESS = 0XF000_ 0010)

SDO_SYNC	Bit	Description	Reset Value
Reserved	[31:2]	Reserved, read as zero, do not modify	0
Configuration Sync Insertion for YpbPr / RGB Component Video	[1:0]	0 = Sync pulse is absent. 1 = Sync pulse is delivered only on Y channel or G channel. 2 = Reserved 3 = Sync pulses are delivered on all the Y/Pb/Pr channels or G/B/R channels This setting is valid if bit [6] of SDO_CONFIG register is set to component. For YpbPr video, it is recommended to deliver sync pulse only on Y channels. For European Scart RGB, no sync delivery is recommended. The sync informations are delivered on CVBS channel which is accompanied with the RGB channels. For VERA RGB (monitor RGB), no sync delivery is recommended, provided that the sync information is carried on HSYNC/VSYNC channels. If the HSYNC/VSYNC channels are not allowed, it is recommended to deliver sync pulse on G channel.	1

5.5 SDO VBI CONFIGURATION REGISTER (SDO_VBI, R/W, ADDRESS = 0XF000_ 0014)

SDO_VBI	Bit	Description	Reset Value
Reserved	[31:15]	Reserved, read as zero, do not modify	E
Wide Screen Signaling Configuration for CVBS Channel	[14]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = 525 line WSS Insertion at 20H and 283H Otherwise, 0 = No Ancillary Data Insertion 1 = 625 line WSS Insertion at 23H This setting is valid if the bit [6] of SDO_CONFIG register is set to composite	1
Closed Caption Configuration for CVBS Channel	[13:12]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = US Closed Caption Insertion at 21H 2 = US Closed Caption Insertion at 21H and 284H 3 = Reserved for Other Use Otherwise, No Ancillary Data Insertion. (NOTE) This setting is valid if the bit [6] of SDO_CONFIG register is set to composite. **NOTE:** European closed caption is not supported.	3
Reserved	[11]	Reserved, read as zero, do not modify	0
Wide Screen Signaling Configuration for Y Channel of S-Video	[10]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = 525 line WSS Insertion at 20H and 283H Otherwise, 0 = No Ancillary Data Insertion 1 = 625 line WSS Insertion at 23H This setting is valid if the bit [6] of SDO_CONFIG register is set to composite.	1

SDO_VBI	Bit	Description	Reset Value
Close Caption Configuration for Y Channel of S-Video	[9:8]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = US Closed Caption Insertion at 21H 2 = US Closed Caption Insertion at 21H and 284H 3 = Reserved for Other Use Otherwise, No Ancillary Data Insertion. This setting is valid if the bit [6] of SDO_CONFIG register is set to composite.	3
CGMS-A Configuration for Progressive RGB Video	[7]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = CGMS-A Insertion at 41H Otherwise, 0 = No Ancillary Data Insertion 1 = CGMS-A Insertion at 43H For RGB component video, the VBI data are delivered on the same channel as the sync pulses are delivered. This setting is valid if the SDO_CONFIG register is set to progressive-scan component RGB video.	1
Wide Screen Signaling Configuration for Interlaced RGB Video	[6]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = 525 line WSS Insertion at 20H and 283H Otherwise, 0 = No Ancillary Data Insertion 1 = 625 line WSS Insertion at 23H For RGB component video, the VBI data are delivered on the same channel as the sync pulses are delivered. This setting is valid if the SDO_CONFIG register is set to interlaced-scan component RGB video.	1

SDO_VBI	Bit	Description	Reset Value
Close Caption Configuration for Interlaced RGB Video	[5:4]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = US Closed Caption Insertion at 21H 2 = US Closed Caption Insertion at 21H and 284H 3 = Reserved for Other Use Otherwise, No Ancillary Data Insertion. For RGB component video, the VBI data are delivered on the same channel as the sync pulses are delivered. This setting is valid if the SDO_CONFIG register is set to interlaced-scan component RGB video.	3
CGMS-A Configuration for Progressive YpbPr Video	[3]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = CGMS-A Insertion at 41H Otherwise, 0 = No Ancillary Data Insertion 1 = CGMS-A Insertion at 43H For YpbPr component video, the VBI data are delivered on the same channel as the sync pulses are delivered on. This setting is valid if the SDO_CONFIG register is set to progressive-scan component YpbPr video.	1
Wide Screen Signaling Configuration for Interlaced YpbPr Video	[2]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = 525 line WSS Insertion at 20H and 283H Otherwise, 0 = No Ancillary Data Insertion 1 = 625 line WSS Insertion at 23H For YpbPr component video, the VBI data are delivered on the same channel as the sync pulses are delivered on. This setting is valid if the SDO_CONFIG register is set to interlaced-scan component YpbPr video.	1

SDO_VBI	Bit	Description	Reset Value
Close Caption Configuration for Interlaced YpbPr Video	[1:0]	If the Selection of Video Standard in SDO_CONFIG Register is set to NTSC (M), PAL (M), PAL 60, or NTSC 4.43, 0 = No Ancillary Data Insertion 1 = US Closed Caption Insertion at 21H 2 = US Closed Caption Insertion at 21H and 284H 3 = Reserved for Other Use Otherwise, No Ancillary Data Insertion. For YpbPr component video, the VBI data are delivered on the same channel as the sync pulses are delivered on. This setting is valid if the SDO_CONFIG register is set to interlaced-scan component YpbPr video.	3

5.6 SDO CHANNEL #0 SCALE CONTROL REGISTER (SDO_SCALE_CH0, R/W, ADDRESS = 0XF000_001C)

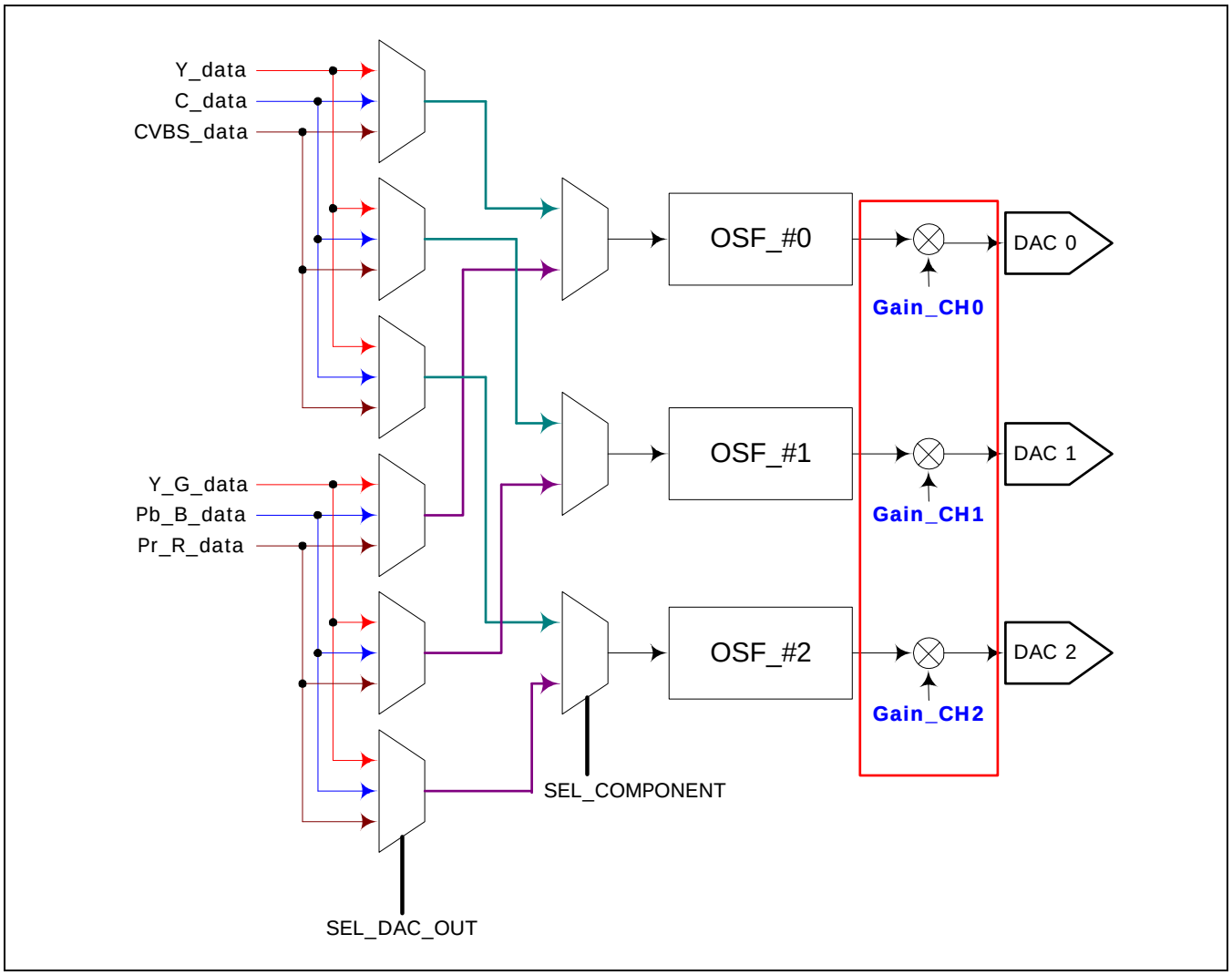


Figure 9.7-20 Individual Gain & Offset Control for DAC Channel Balancing

SDO_SCALE_CH0	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Offset of Channel 0 Signal Scale Conversion	[25:16]	Function F(x) = (X+Offset) * Gain 0x1FF = +511 … 0x001 = +1 0x000 = 0 0x3FF = -1 … 0x200 = -512 This setting is valid if the bit [6] of SDO_CONFIG register is set to component.	000
Reserved	[15:12]	Reserved, read as zero, do not modify	0
Gain of Channel 0 Signal Scale Conversion	[11:0]	Function F(x) = (X+Offset) * Gain 0x000 = x0.0 … 0x400 = x0.5 … 0x800 = x1.0 … 0xC00 = x1.5 … 0xFFF = x1.999512, (2048*2 – 1)/2048 This setting is valid if the bit [6] of SDO_CONFIG register is set to component.	800

5.7 SDO CHANNEL #1 SCALE CONTROL REGISTER (SDO_SCALE_CH1, R/W, ADDRESS = 0XF000_0020)

SDO_SCALE_CH1	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Offset of channel 1 Signal Scale Conversion	[25:16]	Function F(x) = (X+Offset) * Gain 0x1FF = +511 … 0x001 = +1 0x000 = 0 0x3FF = -1 … 0x200 = -512 This setting is valid if the bit [6] of SDO_CONFIG register is set to component.	000
Reserved	[15:12]	Reserved, read as zero, do not modify	0
Gain of channel 1 Signal Scale Conversion	[11:0]	Function F(x) = (X+Offset) * Gain 0x000 = x0.0 … 0x400 = x0.5 … 0x800 = x1.0 … 0xC00 = x1.5 … 0xFFF = x1.999512, (2048*2 – 1)/2048 This setting is valid if the bit [6] of SDO_CONFIG register is set to component.	800

5.8 SDO CHANNEL #2 SCALE CONTROL REGISTER (SDO_SCALE_CH2, R/W, ADDRESS = 0XF000_0024)

SDO_SCALE_CH2	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Offset of channel 2 Signal Scale Conversion	[25:16]	Function F(x) = (X+Offset) * Gain 0x1FF = +511 … 0x001 = +1 0x000 = 0 0x3FF = -1 … 0x200 = -512 This setting is valid if the bit [6] of SDO_CONFIG register is set to component.	000
Reserved	[15:12]	Reserved, read as zero, do not modify	0
Gain of channel 2 Signal Scale Conversion	[11:0]	Function F(x) = (X+Offset) * Gain 0x000 = x0.0 … 0x400 = x0.5 … 0x800 = x1.0 … 0xC00 = x1.5 … 0xFFF = x1.999512, (2048*2 – 1)/2048 This setting is valid if the bit [6] of SDO_CONFIG register is set to component.	800

5.9 SDO VIDEO DELAY CONTROL REGISTER (SDO_YCDELAY, R/W, ADDRESS = 0XF000_ 0034)

SDO_YCDELAY	Bit	Description	Reset Value
Reserved	[31:20]	Reserved, read as zero, do not modify	0
Delay of Y Signal with Respect to C Signal	[19:16]	0x0 = 0.000 usec 0x1 = 0.074 usec … 0xF = 1.111 usec	0
Offset of Video Active Start Position	[15:8]	0x3F = +4.667 usec … 0x01 = +0.074 usec 0x00 = 0.000 usec 0xFF = -0.074 usec … 0x40 = -4.741 usec	FA
Offset of Video Active End Position	[7:0]	0x3F = +4.667 usec … 0x01 = +0.074 usec 0x00 = 0.000 usec 0xFF = -0.074 usec … 0x40 = -4.741 usec	00

5.10 SDO SCH PHASE CONTROL REGISTER (SDO_SCHLOCK, R/W, ADDRESS = 0XF000_ 0038)

SDO_SCHLOCK	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
Color Sub-Carrier Phase Adjustment	[0]	0 = Never adjusted 1 = Every field is adjusted such that color sub-carrier frequency and horizontal frequency are locked to each other.	0

5.11 SDO DAC CONFIGURATION REGISTER (SDO_DAC, R/W, ADDRESS = 0XF000_ 003C)

SDO_DAC	Bit	Description	Reset Value
Reserved	[31:3]	Reserved, read as zero, do not modify	0
Power Down for DAC2	[2]	0 = DAC2 power down 1 = DAC2 power on	0
Power Down for DAC1	[1]	0 = DAC1 power down 1 = DAC1 power on	0
Power Down for DAC0	[0]	0 = DAC0 power down 1 = DAC0 power on	0

NOTE: If SDO_DAC [2:0] =0 (all DAC power down), the internal BGR generation circuit is automatically power down to save a static current (35mA). The mode all DAC power down and only BGR generation circuit alive does not exist.

5.12 SDO STATUS REGISTER (SDO_FINFO, R, ADDRESS = 0XF000_ 0040)

SDO_FINFO	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Field Counter Modulo 1001	[25:16]	This counter is used for 59.94/60.0 Hz field rate conversion.	0
Reserved	[15:2]	Reserved, read as zero, do not modify	0
Field ID	[1]	0 = Top field 1 = Bottom field	1
Field ID with Progressive/ Interlaced Indication	[0]	If the SDO_CONFIG register is set to interlaced, 0 = Top field 1 = Bottom field. If the SDO_CONFIG register is set to progressive, this bit would be fixed to zero	0

5.13 SDO ANTI-ALIASING FILTER COEFFICIENTS REGISTERS

5.13.30 SDO Anti-Aliasing Filter Coefficients (SDO_CR5, R/W, Address = 0xF000_ 00D4)

5.13.31 SDO Anti-Aliasing Filter Coefficients (SDO_CR6, R/W, Address = 0xF000_ 00D8)

5.13.32 SDO Anti-Aliasing Filter Coefficients (SDO_CR7, R/W, Address = 0xF000_ 00DC)

5.13.33 SDO Anti-Aliasing Filter Coefficients (SDO_CR8, R/W, Address = 0xF000_ 00E0)

5.13.34 SDO Anti-Aliasing Filter Coefficients (SDO_CR9, R/W, Address = 0xF000_ 00E4)

5.13.35 SDO Anti-Aliasing Filter Coefficients (SDO_CR10, R/W, Address = 0xF000_ 00E8)

5.13.36 SDO Anti-Aliasing Filter Coefficients (SDO_CR11, R/W, Address = 0xF000_ 00EC)

SDO_Yn / SDO_CBn / SDO_CRn	Bit	Description	Reset Value
Reserved	[31:11]	Reserved, read as zero, do not modify	0
11 Bit Signed Filter Coefficients	[10:0]	Setting values of anti-aliasing filter coefficients is constrained in such a way that total sum of filter coefficients should be equal to a predefined constant. Otherwise, DC component of filter output would be re-scaled from the original one. The value of the constant depends on the setting of SDO_SCALE register :for Y channel, 0x251, at 7.5 IRE setup and 7:3 ratio 0x25D, at 7.5 IRE setup and 10:4 ratio 0x281, at 0 IRE setup and 7:3 ratio 0x28F, at 0 IRE setup and 7:3 ratio,for CB channel, 0x1F3, at 7.5 IRE setup and 7:3 ratio 0x200, at 7.5 IRE setup and 10:4 ratio 0x21E, at 0 IRE setup and 7:3 ratio 0x228, at 0 IRE setup and 7:3 ratio, and for CR channel, 0x2C0, at 7.5 IRE setup and 7:3 ratio 0x2D1, at 7.5 IRE setup and 10:4 ratio 0x2C0, at 0 IRE setup and 7:3 ratio 0x30D, at 0 IRE setup and 7:3 ratio. This setting is valid if the bit [6] of SDO_CONFIG register is set to composite. The setting of Y Filter is applied only to CVBS output.	The reset values are set for the case of ITU-R BT.470 compliant NTSC signal which has 7.5 IRE setup and 10:4 video-to-sync ratio. Refer to "1.15 RESISTERS DESCRIPTION".

5.14 SDO COLOR COMPENSATION ON/OFF CONTROL (SDO_CCCON, R/W, ADDRESS = 0XF000_ 0180)

SDO_CCCON	Bit	Description	Reset Value
Reserved	[31:5]	Reserved, read as zero, do not modify	0
On/Off Control of Brightness/ Hue/Saturation Adjustment	[4]	0 = On 1 = Bypass This setting enables/ disables the brightness/ hue/ saturation controls which are controlled by SDO_YSCALE, SDO_CBSCALE, and SDO_CRSCALE.	0
On/Off Control of YpbPr Color Compensation	[3]	0 = On 1 = Bypass The YpbPr color compensation imposes a saturation operation on the YcbCr data which exceeds their nominal range; 16~235 for Y 16~240 for Cb and Cr in 8 bit resolution.	0
On/Off Control of RGB Color Compensation	[2]	0 = On 1 = Bypass The RGB color compensation imposes a transform operation on the YcbCr data which become to have ill-defined values at transformation into RGB format, i.e. values over the range 16~235 for R, G, and B. The brightness and hue would be maintained but saturation reduced appropriately after the transformation.	0
On/Off Control of YC Color Compensation	[1]	0 = On 1 = Bypass The YC color compensation imposes a saturation operation on the Y/C data. Values which exceed DAC conversion range 0~1023, in 10 bit resolution, would be saturated.	0
On/Off Control of CVBS Color Compensation	[0]	0 = On 1 = Bypass The CVBS color compensation imposes a saturation operation on the CVBS data. Values which exceed DAC conversion range 0~1023, in 10 bit resolution, would be saturated.	0

5.15 SDO BRIGHTNESS CONTROL FOR Y (SDO_YSCALE, R/W, ADDRESS = 0XF000_ 0184)

SDO_YSCALE	Bit	Description	Reset Value
Reserved	[31:24]	Reserved, read as zero, do not modify	0

Gain of Brightness Control with	[23:16]	F(Y) = Gain * Y + Offset 0x00 = 0.0 … 0x80 = 1.0 … 0xFF = 1.992188, (128*2 – 1)/128 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	80
Reserved	[15:9]	Reserved, read as zero, do not modify	0
Offset of Brightness Control with	[8:0]	F(Y) = Gain * Y + Offset 0x0FF = +255 … 0x001 = +1 0x000 = 0 0x1FF = -1 … 0x100 = -256 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	00

5.16 SDO HUE/SATURATION CONTROL FOR CB (SDO_CBSCALE, R/W, ADDRESS = 0XF000_ 0188)

SDO_CBSCALE	Bit	Description	Reset Value
Reserved	[31:25]	Reserved, read as zero, do not modify	0
Gain0_CB	[24:16]	Gain0 of Hue/Saturation Control of CB with F(CB,CR) = CB * Gain0 + CR * Gain1 + Offset 0x0FF = 1.992188, (128*2 – 1)/128 … 0x080 = 1.0 … 0x000 = 0.0 … 0x180 = -1.0 … 0x100 = -2.0 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	80
Reserved	[15:9]	Reserved, read as zero, do not modify	0
Gain1_CB	[8:0]	Gain1 of Hue/Saturation Control of CB with F(CB,CR) = CB * Gain0 + CR * Gain1 + Offset 0x0FF = 1.992188, (128*2 – 1)/128 … 0x080 = 1.0 … 0x000 = 0.0 … 0x180 = -1.0 … 0x100 = -2.0 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	00

5.17 SDO HUE/SATURATION CONTROL FOR CR (SDO_CRSCALE, R/W, ADDRESS = 0XF000_ 018C)

SDO_CRSCALE	Bit	Description	Reset Value
Reserved	[31:25]	Reserved, read as zero, do not modify	0
Gain0_CR	[24:16]	Gain0 of Hue/Saturation Control of CR with F(CB,CR) = CB * Gain0 + CR * Gain1 + Offset 0x0FF = 1.992188, (128*2 – 1)/128 … 0x080 = 1.0 … 0x000 = 0.0 … 0x180 = -1.0 … 0x100 = -2.0 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	00
Reserved	[15:9]	Reserved, read as zero, do not modify	0
Gain1_CR	[8:0]	Gain1 of Hue/Saturation Control of CR with F(CB,CR) = CB * Gain0 + CR * Gain1 + Offset 0x0FF = 1.992188, (128*2 – 1)/128 … 0x080 = 1.0 … 0x000 = 0.0 … 0x180 = -1.0 … 0x100 = -2.0 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	80

5.18 SDO HUE/SATURATION CONTROL FOR CB/CR (SDO_CB_CR_OFFSET, R/W, ADDRESS = 0XF000_ 0190)

SDO_CB_CR_OFFSET	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Offset_CR	[25:16]	Offset of Hue/Saturation Control of CR with 0x1FF = +511 … 0x001 = +1 0x000 = 0 0x3FF = -1 … 0x200 = -512 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	00
Reserved	[15:10]	Reserved, read as zero, do not modify	0
Offset_CB	[9:0]	Offset of Hue/Saturation Control of CB with 0x1FF = +511 … 0x001 = +1 0x000 = 0 0x3FF = -1 … 0x200 = -512 This setting is valid for all the CVBS/Y/C/YpbPr/RGB outputs.	00

5.19 COLOR COMPENSATION CONTROL REGISTER FOR RGB OUTPUT (SDO_RGB_CC, R/W, ADDRESS = 0XF000_0194)

SDO_RGB_CC	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
Max_RGB_Cube	[15:8]	Maximum Boundary of Legal RGB Cube 0xFF = 255 … 0x00 = 0	EB
Min_RGB_Cube	[7:0]	Minimum Boundary of Legal RGB Cube 0xFF = 255 … 0x00 = 0	10

It should be set such that Max_RGB_ Cube [15:8] >= Max_RGB_Cube [7:0]. It is highly recommended for users not to alter their reset values. This setting is valid only if SDO_CCCON [2] is set to 'On'.

5.20 COLOR COMPENSATION CONTROL REGISTER FOR CVBS OUTPUT (SDO_CVBS_CC_Y1, R/W, ADDRESS = 0XF000_ 0198)

SDO_CVBS_CC_Y1	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Y_Lower_Mid_CVBS_Corn	[25:16]	Lower Mid Y Boundary of Legal CVBS Corn 0x3FF = 1023 … 0x000 = 0	200
Reserved	[15:10]	Reserved, read as zero, do not modify	0
Y_Bottom_CVBS_Corn	[9:0]	Bottom Y Boundary of Legal CVBS Corn 0x3FF = 1023 … 0x000 = 0	000

5.21 COLOR COMPENSATION CONTROL REGISTER FOR CVBS OUTPUT (SDO_CVBS_CC_Y2, R/W, ADDRESS = 0XF000_ 019C)

SDO_CVBS_CC_Y2	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Y_Top_CVBS_Corn	[25:16]	Top Y Boundary of Legal CVBS Corn 0x3FF = 1023 … 0x000 = 0	3FF
	[15:10]	Reserved, read as zero, do not modify	0
Y_Upper_Mid_CVBS_Corn	[9:0]	Upper mid Y Boundary of Legal CVBS Corn 0x3FF = 1023 … 0x000 = 0	200

5.22 COLOR COMPENSATION CONTROL REGISTER FOR CVBS OUTPUT (SDO_CVBS_CC_C, R/W, ADDRESS = 0XF000_ 01A0)

SDO_CVBS_CC_C	Bit	Description	Reset Value
Reserved	[31:9]	Reserved, read as zero, do not modify	0
Radius_CVBS_Corn	[8:0]	Radius of Legal CVBS Corn 0x1FF = 511 … 0x000 = 0	1FF
It should be set such that Y_Top_CVBS_Corn >= Y_Upper_Mid_CVBS_Corn >= Y_Lower_Mid_CVBS_Corn >= Y_Bottom_CVBS_Corn. It is highly recommended for users not to alter their reset values. This setting is valid if SDO_CCCON [0] is set to 'On'.			

5.23 COLOR COMPENSATION CONTROL REGISTER FOR S-VIDEO OUTPUT (SDO_YC_CC_Y, R/W, ADDRESS = 0XF000_ 01A4)

SDO_YC_CC_Y	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
Y_Top_YC_Cylinder	[25:16]	Top Y Boundary of Legal YC Cylinder 0x3FF = 1023 … 0x000 = 0	3FF
Reserved	[15:10]	Reserved, read as zero, do not modify	0
Y_Bottom_YC_Cylinder	[9:0]	Bottom Y Boundary of Legal YC Cylinder 0x3FF = 1023 … 0x000 = 0	0

5.24 COLOR COMPENSATION CONTROL REGISTER FOR S-VIDEO OUTPUT (SDO_YC_CC_C, R/W, ADDRESS = 0XF000_ 01A8)

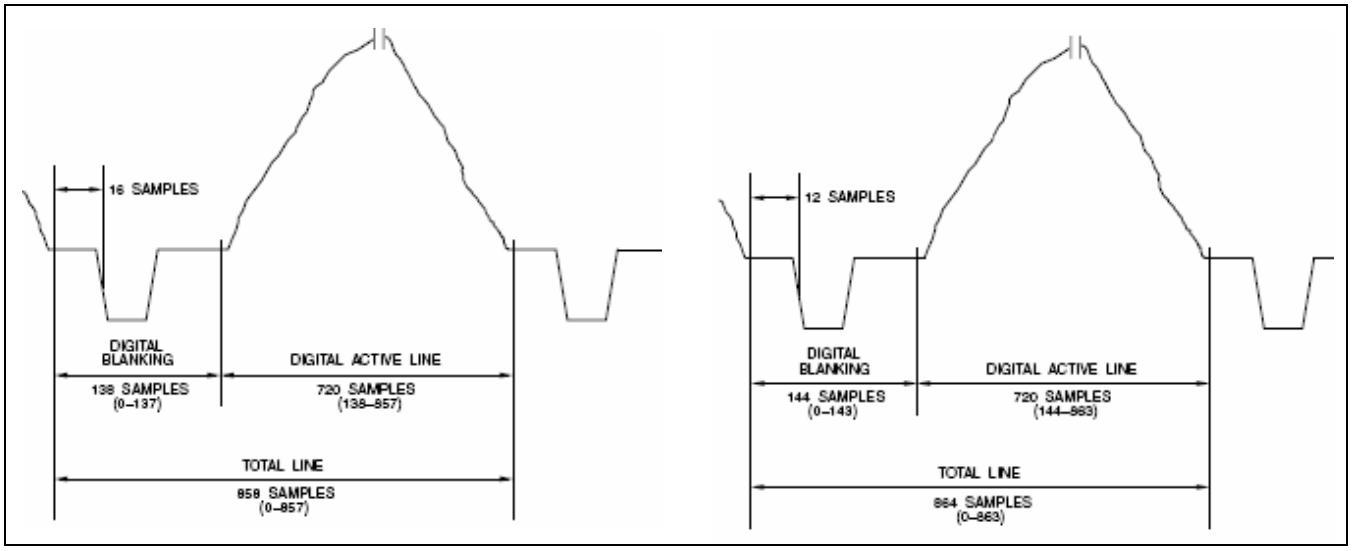
SDO_YC_CC_C	Bit	Description	Reset Value
Reserved	[31:9]	Reserved, read as zero, do not modify	0
Radius_YC_Cylinder	[8:0]	Radius of Legal YC Cylinder 0x1FF = 511 … 0x000 = 0	1FF
It should be set such that Y_Top_YC_Cylinder >= Y_Bottom_YC_Cylinder. It is highly recommended for users not to alter their reset values. This setting is valid if SDO_CCCON [1] is set to 'On'.			

5.25 SDO 525 LINE COMPONENT FRONT/BACK PORCH POSITION CONTROL RESIGTER(SDO_CSC_525_PORCH, R/W, ADDRESS = 0XF000_ 01B0)

SDO_CSC_525_PORCH	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
525 line back-porch position	[25:16]	Back-porch start position	8A
Reserved	[15:10]	Reserved, read as zero, do not modify	0
525 line front-porch position	[9:0]	Front-porch start position	359
- Resolution in progressive : 1/27MHz, - resolution in interlaced : 1/13.5MHz - Compare line count value with porch position			

5.26 SDO 625 LINE COMPONENT FRONT/BACK PORCH POSITION CONTROL RESIGTER(SDO_CSC_625_PORCH, R/W, ADDRESS = 0XF000_ 01B4)

SDO_CSC_625_PORCH	Bit	Description	Reset Value
Reserved	[31:26]	Reserved, read as zero, do not modify	0
625 line back-porch position	[25:16]	Back-porch start position	96
Reserved	[15:10]	Reserved, read as zero, do not modify	0
625 line front-porch position	[9:0]	Front-porch start position	35C
- Resolution in progressive : 1/27MHz, - resolution in interlaced : 1/13.5MHz - Compare line count value with porch position			



*** One Line Count Value of 525 Line** *** One Line Count Value of 625 Line**

5.27 SDO VESA RGB SYNC CONTROL REGISTER (SDO_RGBSYNC, R/W, ADDRESS = 0XF000_ 01C0)

SDO_RGBSYNC	Bit	Description	Reset Value
Reserved	[31:9]	Reserved, read as zero, do not modify	0
Configuration of HSYNC/VSYNC Types	[8]	0 = Composite 1 = Separate If this bit is set to composite, only HSYNC output is used. It possesses both horizontal sync pulses as well as vertical sync pulses.	0
Reserved	[7:5]	Reserved, read as zero, do not modify	0
Polarity Control for VSYNC Output	[4]	0 = Low active 1 = High active	0
Reserved	[3:1]	Reserved, read as zero, do not modify	0
Polarity Control for HSYNC Output	[0]	0 = Low active 1 = High active	0

5.28 SDO OVERSAMPLING #0 FILTER COEFFICIENT (SDO_OSFC00_0, R/W, ADDRESS = 0XF000_ 0200)

SDO_OSFC*N*_*0* (N is 00~23)	Bit	Description	Reset Value
osf_coef(2x*N*+1)	[23:16]~ [27:16]	(2x*N*+1)'th oversampling filter coefficient. Signed 8~12bit integer. Actual value is osf_coef/(210)	Refer to below table
osf_coef(2x*N*)	[7:0] ~[11:0]	(2x*N*)'th oversampling filter coefficient. Signed 8~12bit integer. Actual value is osf_coef/(210)	

5.29 SDO OVERSAMPLING #1 FILTER COEFFICIENT (SDO_OSFC00_1, R/W, ADDRESS = 0XF000_ 02C0)

SDO_OSFC*N*_*1* (N is 00~23)	Bit	Description	Reset Value
osf_coef(2x*N*+1)	[23:16]~[27:16]	(2x*N*+1)'th oversampling filter coefficient. Signed 8~12bit integer. Actual value is osf_coef/(210)	Refer to below table
osf_coef(2x*N*)	[7:0] ~[11:0]	(2x*N*)'th oversampling filter coefficient. Signed 8~12bit integer. Actual value is osf_coef/(210)	

5.30 SDO OVERSAMPLING #2 FILTER COEFFICIENT (SDO_OSFC00_2, R/W, ADDRESS = 0XF000_ 0320)

SDO_OSFC*N*_2 (N is 00~23)	Bit	Description	Reset Value
osf_coef(2x*N*+1)	[23:16]~[27:16]	(2x*N*+1)'th oversampling filter coefficient. Signed 8~12bit integer. Actual value is osf_coef/(210)	Refer to below table
osf_coef(2x*N*)	[7:0] ~[11:0]	(2x*N*)'th oversampling filter coefficient. Signed 8~12bit integer. Actual value is osf_coef/(210)	

5.30.1 SDO Oversampling #2 Filter Coefficient (SDO_OSFC01_2, R/W, Address = 0xF000_ 0324)

5.30.2 SDO Oversampling #2 Filter Coefficient (SDO_OSFC02_2, R/W, Address = 0xF000_ 0328)

5.30.3 SDO Oversampling #2 Filter Coefficient (SDO_OSFC03_2, R/W, Address = 0xF000_ 032C)

5.30.4 SDO Oversampling #2 Filter Coefficient (SDO_OSFC04_2, R/W, Address = 0xF000_ 0330)

5.30.5 SDO Oversampling #2 Filter Coefficient (SDO_OSFC05_2, R/W, Address = 0xF000_ 0334)

5.30.6 SDO Oversampling #2 Filter Coefficient (SDO_OSFC06_2, R/W, Address = 0xF000_ 0338)

5.30.7 SDO Oversampling #2 Filter Coefficient (SDO_OSFC07_2, R/W, Address = 0xF000_ 033C)

5.30.8 SDO Oversampling #2 Filter Coefficient (SDO_OSFC08_2, R/W, Address = 0xF000_ 0340)

5.30.9 SDO Oversampling #2 Filter Coefficient (SDO_OSFC09_2, R/W, Address = 0xF000_ 0344)

5.30.10 SDO Oversampling #2 Filter Coefficient (SDO_OSFC10_2, R/W, Address = 0xF000_ 0348)

5.30.11 SDO Oversampling #2 Filter Coefficient (SDO_OSFC11_2, R/W, Address = 0xF000_ 034C)

5.30.12 SDO Oversampling #2 Filter Coefficient (SDO_OSFC12_2, R/W, Address = 0xF000_ 0350)

5.30.13 SDO Oversampling #2 Filter Coefficient (SDO_OSFC13_2, R/W, Address = 0xF000_ 0354)

5.30.14 SDO Oversampling #2 Filter Coefficient (SDO_OSFC14_2, R/W, Address = 0xF000_ 0358)

5.30.15 SDO Oversampling #2 Filter Coefficient (SDO_OSFC15_2, R/W, Address = 0xF000_ 035C)

5.30.16 SDO Oversampling #2 Filter Coefficient (SDO_OSFC16_2, R/W, Address = 0xF000_ 0360)

5.30.17 SDO Oversampling #2 Filter Coefficient (SDO_OSFC17_2, R/W, Address = 0xF000_ 0364)

5.30.18 SDO Oversampling #2 Filter Coefficient (SDO_OSFC18_2, R/W, Address = 0xF000_ 0368)

5.30.19 SDO Oversampling #2 Filter Coefficient (SDO_OSFC19_2, R/W, Address = 0xF000_ 036C)

5.30.20 SDO Oversampling #2 Filter Coefficient (SDO_OSFC20_2, R/W, Address = 0xF000_ 0370)

5.30.21 SDO Oversampling #2 Filter Coefficient (SDO_OSFC21_2, R/W, Address = 0xF000_ 0374)

5.30.22 SDO Oversampling #2 Filter Coefficient (SDO_OSFC22_2, R/W, Address = 0xF000_ 0378)

5.30.23 SDO Oversampling #2 Filter Coefficient (SDO_OSFC23_2, R/W, Address = 0xF000_ 037C)

5.30.24 4x Oversampling Case Coefficient (Flat response up to 6MHz) – Default reset value

Coefficient	Value	Coefficient	Value	Coefficient	Value
osf_coef00	-2	osf_coef16	-14	osf_coef32	-57
osf_coef01	-3	osf_coef17	-20	osf_coef33	-80
osf_coef02	0	osf_coef18	1	osf_coef34	5
osf_coef03	0	osf_coef19	0	osf_coef35	0
osf_coef04	4	osf_coef20	20	osf_coef36	86
osf_coef05	5	osf_coef21	29	osf_coef37	121
osf_coef06	-1	osf_coef22	-2	osf_coef38	-10
osf_coef07	0	osf_coef23	0	osf_coef39	0
osf_coef08	-6	osf_coef24	-28	osf_coef40	-154
osf_coef09	-9	osf_coef25	-40	osf_coef41	-212
osf_coef10	1	osf_coef26	2	osf_coef42	27
osf_coef11	0	osf_coef27	0	osf_coef43	0
osf_coef12	10	osf_coef28	40	osf_coef44	613
osf_coef13	14	osf_coef29	56	osf_coef45	651
osf_coef14	-1	osf_coef30	-3	osf_coef46	-308
osf_coef15	0	osf_coef31	0	osf_coef47	1024

5.31 2X OVERSAMPLING CASE COEFFICIENT (FLAT RESPONSE UP TO 12MHZ)

Coefficient	*Value*	*Coefficient*	*Value*	*Coefficient*	*Value*
osf_coef00	*0*	*osf_coef16*	*0*	*osf_coef32*	*-19*
osf_coef01	*0*	*osf_coef17*	*0*	*osf_coef33*	*0*
osf_coef02	*0*	*osf_coef18*	*0*	*osf_coef34*	*28*
osf_coef03	*0*	*osf_coef19*	*0*	*osf_coef35*	*0*
osf_coef04	*0*	*osf_coef20*	*0*	*osf_coef36*	*-39*
osf_coef05	*0*	*osf_coef21*	*0*	*osf_coef37*	*0*
osf_coef06	*0*	*osf_coef22*	*0*	*osf_coef38*	*55*
osf_coef07	*0*	*osf_coef23*	*0*	*osf_coef39*	*0*
osf_coef08	*0*	*osf_coef24*	*-3*	*osf_coef40*	*-79*
osf_coef09	*0*	*osf_coef25*	*0*	*osf_coef41*	*0*
osf_coef10	*0*	*osf_coef26*	*5*	*osf_coef42*	*120*
osf_coef11	*0*	*osf_coef27*	*0*	*osf_coef43*	*0*
osf_coef12	*0*	*osf_coef28*	*-8*	*osf_coef44*	*-211*
osf_coef13	*0*	*osf_coef29*	*0*	*osf_coef45*	*0*
osf_coef14	*0*	*osf_coef30*	*13*	*osf_coef46*	*650*
osf_coef15	*0*	*osf_coef31*	*0*	*osf_coef47*	*1024*

5.32 SDO CHANNEL CROSSTALK CANCELLATION COEFFICIENT FOR CH. 0 (SDO_XTALK0, R/W, ADDRESS = 0XF000_ 0260)

SDO_XTALK0	Bit	Description	Reset Value
Reserved	[31:24]	Reserved, read as zero, do not modify	0
xtalk_coef02	[23:16]	Signed 8 bit integer. Actual value is xtalk_coef02/(2^{10}) 0x7F : 0.124 0x7E : 0.123 … 0x01 : 0.000977 0x00 : 0.000000 0xFF :-0.000977 0xFE :-0.001953 ... 0x81 :-0.124 0x80 : -0.125	00
Reserved	[15:8]	Reserved, read as zero, do not modify	0
xtalk_coef01	[7:0]	Same as xtalk_coef01	00

5.33 SDO CHANNEL CROSSTALK CANCELLATION COEFFICIENT FOR CH. 1 (SDO_XTALK1, R/W, ADDRESS = 0XF000_ 0264)

SDO_XTALK1	Bit	Description	Reset Value
Reserved	[31:24]	Reserved, read as zero, do not modify	0
xtalk_coef12	[23:16]	Same as xtalk_coef02	00
Reserved	[15:8]	Reserved, read as zero, do not modify	0
xtalk_coef10	[7:0]	Same as xtalk_coef02	00

5.34 SDO CHANNEL CROSSTALK CANCELLATION COEFFICIENT FOR CH. 2 (SDO_XTALK2, R/W, ADDRESS = 0XF000_ 0268)

SDO_XTALK2	Bit	Description	Reset Value
Reserved	[31:24]	Reserved, read as zero, do not modify	0
xtalk_coef21	[23:16]	Same as xtalk_coef02	00
Reserved	[15:8]	Reserved, read as zero, do not modify	0
xtalk_coef20	[7:0]	Same as xtalk_coef02	00

5.35 SDO BLACK BURST CONTROL REGISTER (SDO_BB_CTRL, R/W, ADDRESS = 0XF000_ 026C)

SDO_XTALK2	Bit	Description	Reset Value
Reserved	[31:18]	Reserved, read as zero, do not modify	0
ref_bb_level	[17:8]	Black level setting value. It specifies the level during horizontal active video for black burst signal. The recommended values are NTSC : 0x11A (include 7.5 IRE setup) PAL : 0xFB (without setup)	0x11A
Reserved	[7:6]	Reserved, read as zero, do not modify	0
sel_bb_chan	[5:4]	Black burst (BB) test channel selection DAC0 DAC1 DAC2 00 = CVBS BB BB 01 = BB CVBS BB 10 = BB BB CVBS 11 = reserved	00
Reserved	[3:1]	Reserved, read as zero, do not modify	0
BB mode	[0]	Black burst test mode enable. If set, all of the bit fields in **SDO_CONFIG** register except *'Selection of Video Mux for DAC 0, 1, and 2'* bit are discarded and SDO enters black burst test mode. In order to enable black burst test, *'Selection of Video Mux for DAC 0, 1, and 2' bit has to be set to zero*.	0

5.36 SDO INTERRUPT REQUEST REGISTER (SDO_IRQ, R/W, ADDRESS = 0XF000_ 0280)

SDO_IRQ	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
Vsync Interrupt Request	[0]	0 = No interrupt 1 = Interrupt request pending (This interrupt is requested if TVOUT module generates the falling edge of vertical synchronization pulses at each field. Write 1 to reset this bit.(Writing '0' has no effect).	0

5.37 SDO INTERRUPT REQUEST MASKING REGISTER (SDO_IRQMASK, R/W, ADDRESS = 0XF000_ 0284)

SDO_IRQMASK	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
Vsync Interrupt Request Masking	[0]	0 = Enables Interrupt request 1 = Disables Interrupt request (The status pending bit of SDO Interrupt Request Register is asserted even if the request is disabled. Only the request to MCU will be disabled.)	0

5.38 SDO CLOSED CAPTION DATA REGISTERS (SDO_ARMCC, R/W, ADDRESS = 0XF000_ 03C0)

SDO_ARMCC	Bit	Description	Reset Value
Reserved	[31:16]	Reserved, read as zero, do not modify	0
Display Control Character of Closed Caption Data	[15:8]	Bit alignment of the Display Control Character register is in their incoming order. The first incoming bit becomes LSB, i.e. Display Control Character [7:0] = {p, b6, b5, b4, b3, b2, b1, b0}, where b*n* represents data bit with their incoming order *n*, and *p* denotes their odd parity bit.	0
Non Display Control Character of Closed Caption Data	[7:0]	Bit alignment of the Non Display Control Character register is in their incoming order. The first incoming bit becomes LSB, i.e. Non Display Control Character [7:0] = {p, b6, b5, b4, b3, b2, b1, b0}, where b*n* represents data bit with their incoming order *n*, and *p* denotes their odd parity bit.	0

NOTE: This register is used for European Caption as well as US Closed Caption.

5.39 SDO WSS 525 DATA REGISTERS (SDO_ARMWSS525, R/W, ADDRESS = 0XF000_ 03C4)

SDO_ARMWSS525	Bit	Description	Reset Value
Reserved	[31:20]	Reserved, read as zero, do not modify	0
CRC of WSS 525 Data	[19:14]	Bit alignment of the CRC register is according to their incoming order. The first incoming bit becomes LSB, i.e. CRC [19:14] = {b19, b18, b17, b16, b15, b14}, where bn represents data bit with their incoming order n. The CRC used is X^6+ X + 1, all preset to 1.	0
Word 2 of WSS 525 Data	[13:6]	Bit alignment of the Word 2 register is according to their incoming order. The first incoming bit becomes LSB, i.e. Word 2 [13:6] = {b13, b12, b11, b10, b9, b8, b7, b6}, where bn represents data bit with their incoming order n. The Word 2 data are used for copy control: b7, b6 : 00 copying permitted 01 one copy permitted 10 reserved 11 no copying permitted b9 b8 : (reserved) b10 : not analog pre-recorded medium analog pre-recorded medium b13, b12, b11 : (reserved)	0
Word 1 of WSS 525 Data	[5:2]	Bit alignment of the Word 1 register is according to their incoming order. The first incoming bit becomes LSB, i.e. Word 1 [5:2] = {b5, b4, b3, b2}, where bn represents data bit with their incoming order n. The Word 1 data are used to indicate the existence of Word 2 data: b5, b4, b3, b2 : 0000 copy control information 1111 default	0

SDO_ARMWSS525	Bit	Description	Reset Value
Word 0 of WSS 525 Data	[1:0]	Bit alignment of the Word 0 register is according to their incoming order. The first incoming bit becomes LSB, i.e. Word 0 [1:0] = {b1, b0}, where bn represents data bit with their incoming order n. The Word 0 data are used for display aspect ratio control: b1, b0 : 00 4:3 aspect ratio normal 01 16:9 aspect ratio anamorphic 10 4:3 aspect ratio letterbox 11 reserved	0

5.40 SDO WSS 625 DATA REGISTERS (SDO_ARMWSS625, R/W, ADDRESS = 0XF000_ 03C8)

SDO_ARMWSS625	Bit	Description	Reset Value
Reserved	[31:14]	Reserved, read as zero, do not modify	0
Group D of WSS 625 Data	[13:11]	Bit alignment of the Group D register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group D [13:11] = {b13, b12, b11}, where bn represents data bit with their incoming order n. The Group D data are used for surround sound and copy control: b11 : surround sound no yes b12 : copyright no copyright asserted or unknown copyright asserted b13 : copy protection copying not restricted copying restricted	0

SDO_ARMWSS625	Bit	Description	Reset Value
Group C of WSS 625 Data	[10:8]	Bit alignment of the Group C register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group C [10:8] = {b10, b9, b8}, where bn represents data bit with their incoming order n. The Group C data are used for subtitles: b8 : teletext subtitles 0 no 1 yes b10 , b9 : open subtitles 00 no 01 inside active picture 10 outside active picture 11 reserved	0
Group B of WSS 625 Data	[7:4]	Bit alignment of the Group B register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group B [7:4] = {b7, b6, b5, b4}, where bn represents data bit with their incoming order n. The Group B data are used for enhanced video services: b4 : mode 0 camera mode 1 film mode b5 : color encoding normal PAL Motion Adaptive ColorPlus b6 : helper signals not present present b7 : fixed to 0	0

SDO_ARMWSS625	Bit	Description	Reset Value
Group A of WSS 625 Data	[3:0]	Bit alignment of the Group A register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group A [3:0] = {b3, b2, b1, b0}, where bn represents data bit with their incoming order n. The Group A data are used for display aspect ratio control: b3, b2 b1, b0 : 1000 4:3 full format - 576 lines 0001 14:9 letterbox center 504 lines 0010 14:9 letterbox top 504 lines 1011 16:9 letterbox center 430 lines 0100 16:9 letterbox top 430 lines 1101 >16:9 letterbox center - 1110 14:9 full format center 576 lines 0111 16:9 anamorphic - 576 lines	0

5.41 SDO CGMS-A 525 DATA REGISTERS (SDO_ARMCGMS525, R/W, ADDRESS = 0XF000_ 03CC)

SDO_ARMCGMS525	Bit	Description	Reset Value
Reserved	[31:20]	Reserved, read as zero, do not modify	0
CRC of CGMS-A 525 Data	[19:14]	Bit alignment of the CRC register is according to their incoming order. The first incoming bit becomes LSB, i.e. CRC [19:14] = {b19, b18, b17, b16, b15, b14}, where bn represents data bit with their incoming order n. The CRC used is X^6+ X + 1, all preset to 1.	0
Word 2 of CGMS-A 525 Data	[13:6]	Bit alignment of the Word 2 register is according to their incoming order. The first incoming bit becomes LSB, i.e. Word 2 [13:6] = {b13, b12, b11, b10, b9, b8, b7, b6}, where bn represents data bit with their incoming order n. The Word 2 data are used for copy control: b7, b6 : 00 copying permitted 01 one copy permitted 10 reserved 11 no copying permitted b9 b8 : (reserved) b10 : No analog pre-recorded medium, analog pre-recorded medium b13, b12, b11 : (reserved)	0
Word 1 of CGMS-A 525 Data	[5:2]	Bit alignment of the Word 1 register is according to their incoming order. The first incoming bit becomes LSB, i.e. Word 1 [5:2] = {b5, b4, b3, b2}, where bn represents data bit with their incoming order n. The Word 1 data are used to indicate the existence of Word 2 data: b5, b4, b3, b2 : 0000 copy control information 1111 default	0

SDO_ARMCGMS525	Bit	Description	Reset Value
Word 0 of CGMS-A 525 Data	[1:0]	Bit alignment of the Word 0 register is according to their incoming order. The first incoming bit becomes LSB, i.e. Word 0 [1:0] = {b1, b0}, where bn represents data bit with their incoming order n. The Word 0 data are used for display aspect ratio control: b1, b0 : 00 4:3 aspect ratio normal 01 16:9 aspect ratio anamorphic 10 4:3 aspect ratio letterbox 11 reserved	0

5.42 SDO CGMS-A 625 DATA REGISTERS (SDO_ARMCGMS625, R/W, ADDRESS = 0XF000_ 03D4)

SDO_ARMCGMS625	Bit	Description	Reset Value
Reserved	[31:14]	Reserved, read as zero, do not modify	0
Group D of CGMS-A 625 Data	[13:11]	Bit alignment of the Group D register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group D [13:11] = {b13, b12, b11}, where bn represents data bit with their incoming order n. The Group D data are used for surround sound and copy control: b11 : surround sound no yes b12 : copyright no copyright asserted or unknown copyright asserted b13 : copy protection copying not restricted copying restricted	0

SDO_ARMCGMS625	Bit	Description	Reset Value
Group C of CGMS-A 625 Data	[10:8]	Bit alignment of the Group C register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group C [10:8] = {b10, b9, b8}, where bn represents data bit with their incoming order n. The Group C data are used for subtitles: b8 : teletext subtitles 0 no 1 yes b10 , b9 : open subtitles 00 no 01 inside active picture 10 outside active picture 11 reserved	0
Group B of CGMS-A 625 Data	[7:4]	Bit alignment of the Group B register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group B [7:4] = {b7, b6, b5, b4}, where bn represents data bit with their incoming order n. The Group B data are used for enhanced video services: b4 : mode 0 camera mode 1 film mode b5 : color encoding normal PAL Motion Adaptive ColorPlus b6 : helper signals not present present b7 : fixed to 0	0

SDO_ARMCGMS625	Bit	Description	Reset Value
Group A of CGMS-A 625 Data	[3:0]	Bit alignment of the Group A register is according to their incoming order. The first incoming bit becomes LSB, i.e. Group A [3:0] = {b3, b2, b1, b0}, where bn represents data bit with their incoming order n. The Group A data are used for display aspect ratio control: b3, b2 b1, b0 : 1000 4:3 full format - 576 lines 0001 14:9 letterbox center 504 lines 0010 14:9 letterbox top 504 lines 1011 16:9 letterbox center 430 lines 0100 16:9 letterbox top 430 lines 1101 >16:9 letterbox center - 1110 14:9 full format center 576 lines 0111 16:9 anamorphic - 576 lines	0

5.43 SDO VERSION REGISTER (SDO_VERSION, R, ADDRESS = 0XF000_ 03D8)

SDO_VERSION	Bit	Description	Reset Value
TVOUT version number	[31:0]	Read only register of TVOUT version number	C

6 SHADOW REGISTERS

Register	Address	R/W	Description	Reset Value
SDO_CC	0xF000_0380	R/W	Closed Caption Data Shadow register	0x0000_0000
SDO_WSS525	0xF000_0384	R/W	WSS 525 Data Shadow Register	0x0000_0000
SDO_WSS625	0xF000_0388	R/W	WSS 625 Data Shadow Register	0x0000_0000
SDO_CGMS525	0xF000_038C	R/W	CGMS-A 525 Data Shadow Register	0x0000_0000
SDO_CGMS625	0xF000_0394	R/W	CGMS-A 625 Data Shadow Register	0x0000_0000

NOTE: SDO_ARMCC, SDO_ARMWSS525, SDO_ARMWSS625, SDO_ARMCGMS525, and SDO_ ARMCGMS625 are paired with above shadow registers, respectively, as follows:

- SDO_ARMCC : SDO_CC
- SDO_ARMWSS525 : SDO_WSS525
- SDO_ARMWSS625 : SDO_WSS625
- SDO_ARMCGMS525: SDO_CGMS525
- SDO_ARMCGMS625: SDO_CGMS625

If MCU set values in the source registers with the prefix of SDO_ARMXXX, they are not immediately effective, but the values are copied into the corresponding shadow registers, which are named with SDO_XXX, at the next vertical sync interrupt. Then the values become effective during VBI interval at the next field.

Please avoid setting direct value to these shadow resigers.

6.1 SDO CLOSED CAPTION DATA SHADOW REGISTER (SDO_CC, R/W, ADDRESS = 0XF000_ 0380)

SDO_CC	Bit	Description	Reset Value
Reserved	[31:16]	Reserved, read as zero, do not modify	0
Display Control Character of Closed Caption Data	[15:8]	If MCU set values of SDO_ARMCC, the values are copied into this shadow register at the next vertical sync interrupt. Do not set values to this shadow register.	0
Non Display Control Character of Closed Caption Data	[7:0]		0

6.2 SDO WSS 525 DATA SHADOW REGISTERS (SDO_WSS525, R/W, ADDRESS = 0XF000_ 0384)

SDO_WSS525	Bit	Description	Reset Value
Reserved	[31:20]	Reserved, read as zero, do not modify	0
CRC of WSS 525 Data	[19:14]	If MCU set values of SDO_ARMWSS525, the values are copied into this shadow register at the next vertical sync interrupt. Do not set values to this shadow register.	0
Word 2 of WSS 525 Data	[13:6]		0
Word 1 of WSS 525 Data	[5:2]		0
Word 0 of WSS 525 Data	[1:0]		0

6.3 SDO WSS 625 DATA SHADOW REGISTERS (SDO_WSS625, R/W, ADDRESS = 0XF000_ 0388)

SDO_ARMWSS625	Bit	Description	Reset Value
Reserved	[31:14]	Reserved, read as zero, do not modify	0
Group D of WSS 625 Data	[13:11]	If MCU set values of SDO_ARMWSS625, the values are copied into this shadow register at the next vertical sync interrupt. Do not set values to this shadow register.	0
Group C of WSS 625 Data	[10:8]		0
Group B of WSS 625 Data	[7:4]		0
Group A of WSS 625 Data	[3:0]		0

6.4 SDO CGMS-A 525 DATA SHADOW REGISTERS (SDO_CGMS525, R/W, ADDRESS = 0XF000_ 038C)

SDO_CGMS525	Bit	Description	Reset Value
Reserved	[31:20]	Reserved, read as zero, do not modify	0
CRC of CGMS-A 525 Data	[19:14]	If MCU set values of SDO_ARMCGMS525, the values are copied into this shadow register at the next vertical sync interrupt. Do not set values to this shadow register.	0
Word 2 of CGMS-A 525 Data	[13:6]		0
Word 1 of CGMS-A 525 Data	[5:2]		0
Word 0 of CGMS-A 525 Data	[1:0]		0

6.5 SDO CGMS-A 625 DATA REGISTERS (SDO_CGMS625, R/W, ADDRESS = 0XF000_ 0394)

SDO_CGMS625	Bit	Description	Reset Value
Reserved	[31:14]	Reserved, read as zero, do not modify	0
Group D of CGMS-A 625 Data	[13:11]	If MCU set values of SDO_ARMCGMS625, the values are copied into this shadow register at the next vertical sync interrupt. Do not set values to this shadow register.	0
Group C of CGMS-A 625 Data	[10:8]		0
Group B of CGMS-A 625 Data	[7:4]		0
Group A of CGMS-A 625 Data	[3:0]		0

7 VIDEO DAC

7.1 GENERAL DESCRIPTION

The dac6620t_50p is a 10-bit 3channel CMOS Digital-to-Analog converter for general applications.
Its maximum conversion rate is 54MHz. It operates at analog power, 2.7V to 3.3V and provides full scale output currents of 6.5mA at one channel with 200 ohm load for 1.3V.
The dac6620t_50p has a power down mode to reduce power consumption during inactive periods.

7.2 FEATURES

- 65nm Low Power Process
- Maximum 54MHz Update Rate.
- 10-bit Current Output DAC
- 1.3Vpp Triple Output Compliance Range
- Internal Voltage Reference
- Fine Full Scale control: 91.1% ~ 114.8%
- Power Down Mode (Low Active)

7.3 TYPICAL APPLICATIONS

- Graphic display
- Video applications (SD, HD)
- General purpose high-speed

7.4 FUNCTIONAL BLOCK DIAGRAM

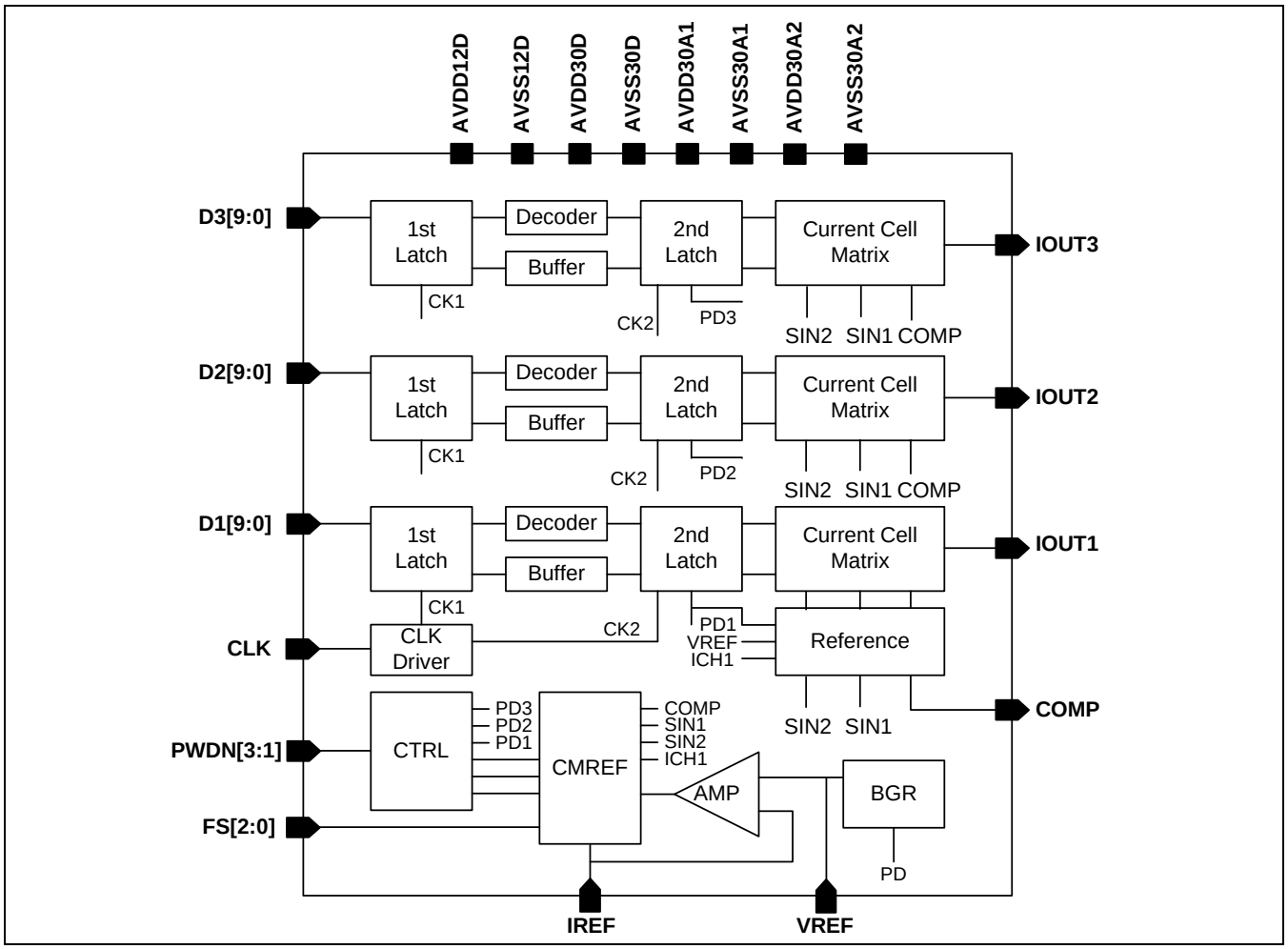


Figure 9.7-21 Block Diagram of Video DAC

7.5 CORE PORT DESCRIPTION

Table 9.7-8 Port Description of Video DAC

Name	Width	I/O	Description
D1[9:0], D2[9:0], D3[9:0]	10	DI	Data Bit 10(1.2V). Offset binary format.
CLK	1	DI	Clock Input. Data latched on positive edge of clock.
PWDN[3:1]	3	DI	Power Down Control Pin. Power down mode is active low.
FS[2:0]		DI	Full scale voltage control (91.1%~114.8%) Default : FS[2:0]=000
VREF	-	AI	Voltage reference for DAC. An Internal voltage reference of nominally 1.26V is provided. Is driven with an externalreference source.
IREF	-	AI	Full Scale Adjust control. The full-scale current drive on each of the output channels is determined by the value of a resistor RSET connected between this terminal and AVSS30A1.
IOUT1, IOUT2, IOUT3	-	AO	DAC current output. Full scale output is achieved if all input bits are set to binary 1.
COMP	-	AO	Compensation Pin. This is a compensation pin for the internal reference amplifier. A 0.1uF ceramic capacitor must be connected between COMP and AVDD30A1.
AVDD30A1, AVDD30A2	-	AP	Analog Supply Voltage. Accepts a supply voltage range of 2.7V to 3.3V.
AVSS30AR, AVSS30A2	-	AG	Analog Ground for 3.0V Analog Supply Voltage.
AVDD30D	-	DP	Digital Supply Voltage. Accepts a supply voltage range of 2.7V to 3.3V.
AVSS30D	-	DG	Digital Ground for 3.0V Digital Supply Voltage.
AVDD12D	-	LP	1.2V Digital Supply Voltage. Share the 1.2V supply voltage pad with digital logic.
AVSS12D	-	LG	Digital Ground for 1.2V Digital Supply Voltage. Share the ground pad with digital logic.

7.5.1 I/O Type Abbr.

AI: Analog Input, DI: Digital Input, AO: Analog Output, AP: Analog Power, AG: Analog Ground,
DP: Digital Power, DG: Digital Ground, LP: Logic Power, LG: Logic Ground

7.6 FUNCTIONAL DESCRIPTION

This is a 10-bit 54MSPS digital-to-analog data converter and uses segment architecture for 6bits of MSB, binaryweighted architecture for 4-bits of LSB. It consist of 1st latch block, decoder block, 2nd latch block, amp block, Current Mirror (CM) block and current cell matrix block. This IP uses reference current to decide the 1LSB current size by dividing the reference current by 31 times. Therefore the reference current must be constant by using high DC gain amp block. The most significant block of this IP is current cell matrix and it must maintain the uniformity at each analog current cell, therefore layout designer must take care of the matching characteristics of analog current cell matrix and CM block. More than 90% of supply current is dissipated at current cell matrix block and AMP block. It uses SEC standard cell as all digital cell of latch, decoder and buffer. To adjust full scale output current, the "R_{SET}" resistor value (connected to IREF pin) must be decided. Its voltage output is obtained by connecting R_{LOAD} aganst GND (connected to IOUT pin)

7.6.1 Integral Non Linearity (or ILE):

INL is the maximum deviation of the input/ output characteristic from a straight line passed through its end points. The difference between the ideal and actual characteristics is called the INL profile.

7.6.2 Differential Non Linearity (or DLE):

DNL is the maximum deviation in the output step size from the ideal value of one least significant bit.

7.6.3 Monotonicity:

A D/A converter is monotonic if the output either increases or remains constants as the digital input increases.

7.6.4 Offset Error:

The deviation of the output current from the ideal of zero is called offset error. For IO, 0mV output expected if the inputs are all 0s.

7.6.5 Gain Error:

The difference between the actual and ideal output span. The actual span is determined by the output if all inputs are set to 1s minus the output if all inputs are set to 0s.

7.6.6 Output Compliance Range:

The range of allowable voltage at the output of a current-steering DAC. Operation beyond the maximum compliance limits causes either output stage saturation or breakdown resulting in nonlinear performance.

7.6.7 Full scale output current

The full scale output current I_{FS} is set by the reference voltage V_{VREF} and the external resistor R_{SET}. This is expressed as:

$I_{FS}=V_{VREF} \times 1023/(R_{SET} \times 31)$

7.6.8 Full scale output voltage control

The full scale output voltage V_{FS} is set by full scale output current I_{FS} and the external resistor R_{LOAD}. This is expressed as:

$VFS=I_{FS} \times R_{LOAD}$

FS [2:0] are the control bits to vary V_{FS}, [000] being default as shown in the following table. This is used for small adjustment in V_{FS}.

Table 9.7-9 Full Scale Output Voltage Control

FS[2:0]	V_{FS}		Comments
	Ratio	%	
000	31/31	100%	Default
001	31/32	96.8%	
010	31/33	93.9%	
011	31/34	91.1%	
100	31/27	114.8%	
101	31/28	110.7%	
110	31/29	106.8%	
111	31/30	103.3%	

7.7 RECOMMENDED OPERATING CONDITIONS

Table 9.7-10 Operating Conditions

Characteristics	Symbol	Min	Typ	Max	Unit
Operating Supply Voltage	AVDD30A1 AVDD30A2 AVDD30D	2.7	3.0	3.3	V
	AVDD12D	0.99	1.2	1.32	V
Operating Temperature Range	Top	–40	–	85	°C

7.8 DC ELECTRICAL CHARACTERISTICS

(AVDD30A*=3.0V, AVDD12D=1.2V, AVSS30A*=AVSS30D=AVSS12D= 0V, PD [3:1] =all High, Temp=25°C, R (IREF) =6.34KΩ, Load resistance=200Ω unless otherwise specified.)

Table 9.7-11 DC Electrical Characteristic

Characteristics		Symbol	Min	Typ	Max	Unit
Resolution		Bit	–	–	10	Bits
Differential linearity error		DNL	–	–	±1	LSB
Integral linearity error		INL	–	–	±3	LSB
Monotonicity		Guaranteed				
Reference Voltage		V_{VREF}	–	1.26	–	V
Maximum Output Compliance		V_{OC}	–	–	1.43	V
Full Scale Output Current Per Channel		I_{FS}	5.85	6.5	7.15	mA
Full Scale Output Voltage		V_{FS}	1.17	1.3	1.43	V
Operating Current		I_{OP}	19	23	27	mA
Power Down Current		I_{PD}	0	–	100	uA
Channel Balance	Y Channel – Pb Channel	CB_{YPB}	–	1.2	5.0	%
	Y Channel – Pr Channel	CB_{YPR}	–	1.2	5.0	%
	Pb Channel – Pr Channel	CB_{PBPR}	–	0.9	3.5	%

7.9 AC ELECTRICAL CHARACTERISTICS

(AVDD30A*=3.0V, AVDD12D=1.2V, AVSS30A*=AVSS30D=AVSS12D= 0V, PD [3:1] =all High, Temp=25°C, R (IREF) =6.34KΩ, load resistance=200Ω unless otherwise specified.)

Table 9.7-12 AC Electrical Characteristic

Characteristics		Symbol	Min	Typ	Max	Unit
Maximum Conversion Rate		F_{CLK}	–	–	54	MHz
Analog Output Delay		T_D	–	–	7	ns
Analog Output Rise Time		T_R	–	–	5	ns
Analog Output Fall Time		T_F	–	–	5	ns
Setup Time		T_S	–	4	–	nsec
Hold time		T_H	–	4	–	nsec
PSRR		PSRR	–	-44	-40	dB
Crosstalk	Y Channel – Pb Channel	X_{YPB}	–	-50	-40	dB
	Y Channel – Pr Channel	X_{YPR}	–	-50	-40	dB
	Pb Channel – Pr Channel	X_{PBPR}	–	-50	-40	dB
SNR		SNR	45	51	–	dB
SFDR(Spurious Free Dynamic Range) (Fout=1MHz, Fclk=54MHz)		SFDR	–	-60	-50	dBc

NOTE: 1. Analog output delay, analog output rise/ fall time and Setup/ Hold time are simulation values, not a test result. It varies based on the difference output parastic conditions.

2. SFDR varies depends on the parasitic capacitor value under given test condition. Above specification is the result of parasitic capacitor 10pF

7.10 TIMING DIAGRAM

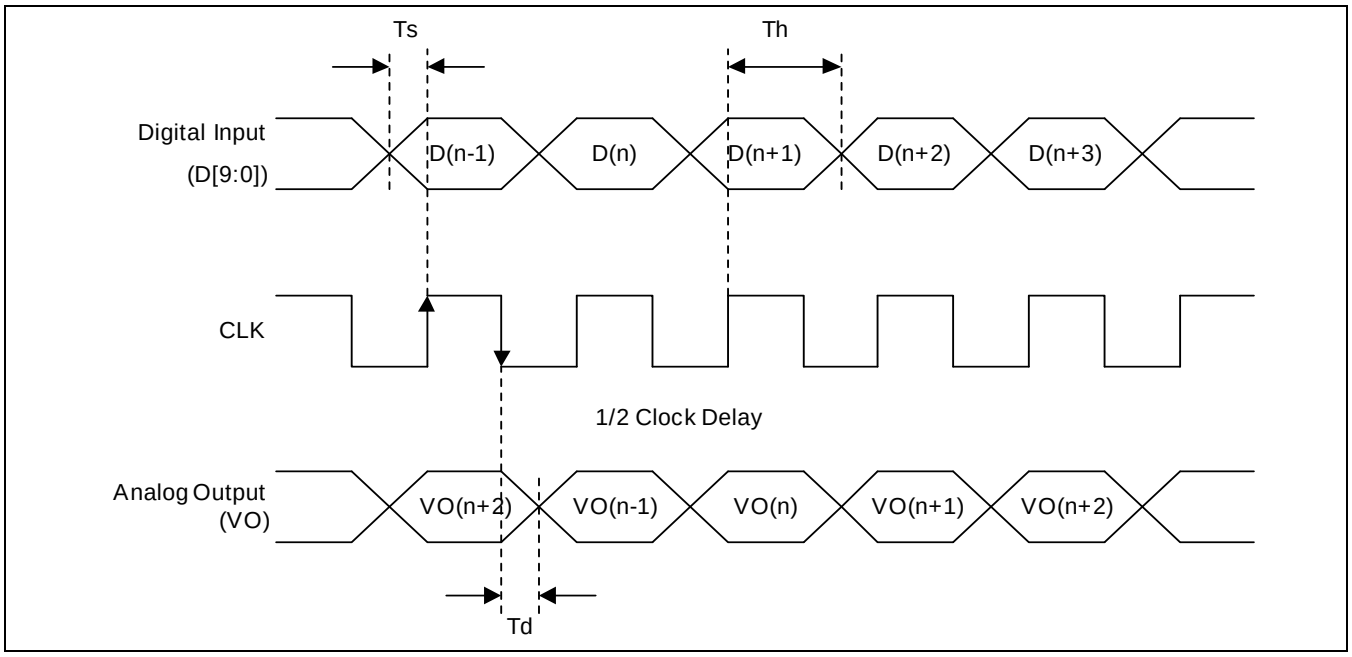
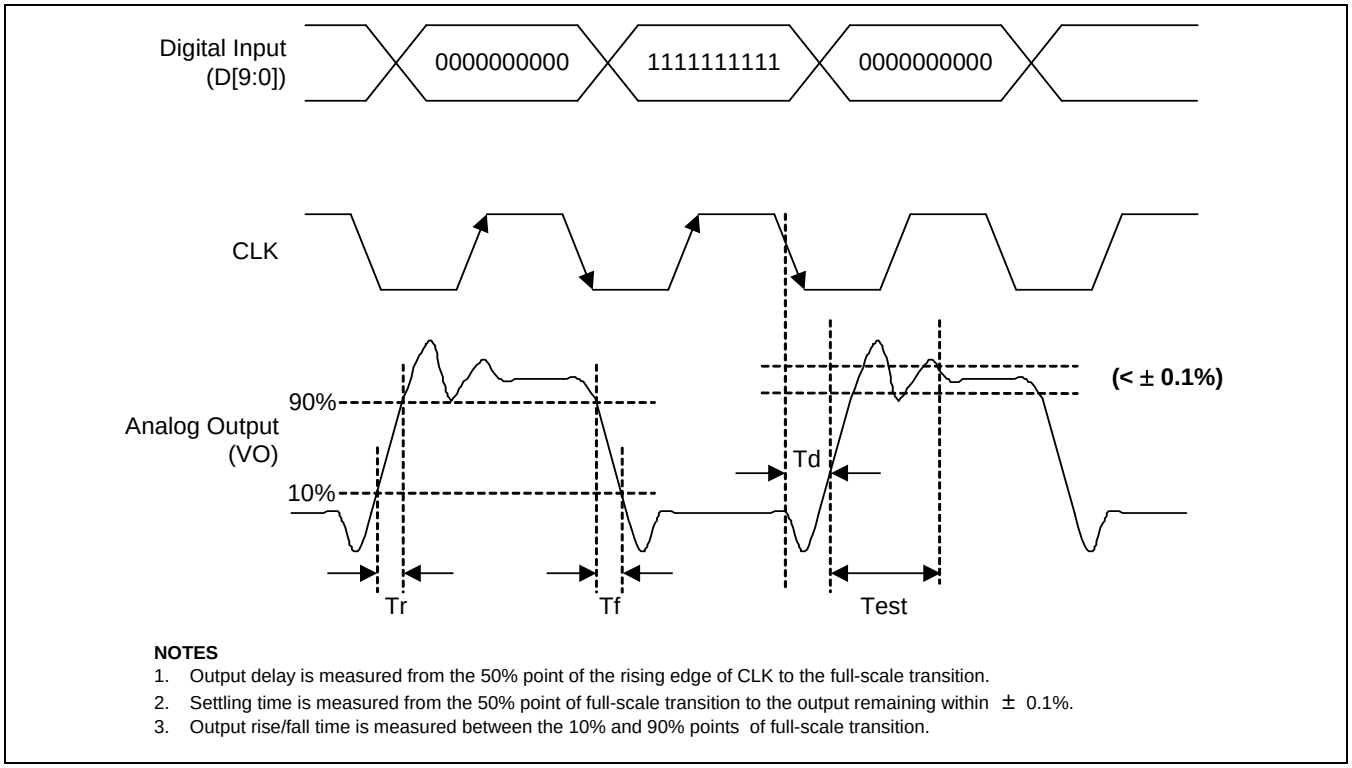


Figure 9.7-22 Timing Diagram of Digital-to-Analog Conversion



NOTES

1. Output delay is measured from the 50% point of the rising edge of CLK to the full-scale transition.
2. Settling time is measured from the 50% point of full-scale transition to the output remaining within ± 0.1%.
3. Output rise/fall time is measured between the 10% and 90% points of full-scale transition.

Figure 9.7-23 Rise and Falling Time

7.11 BOARD CONFIGURATION GUIDE

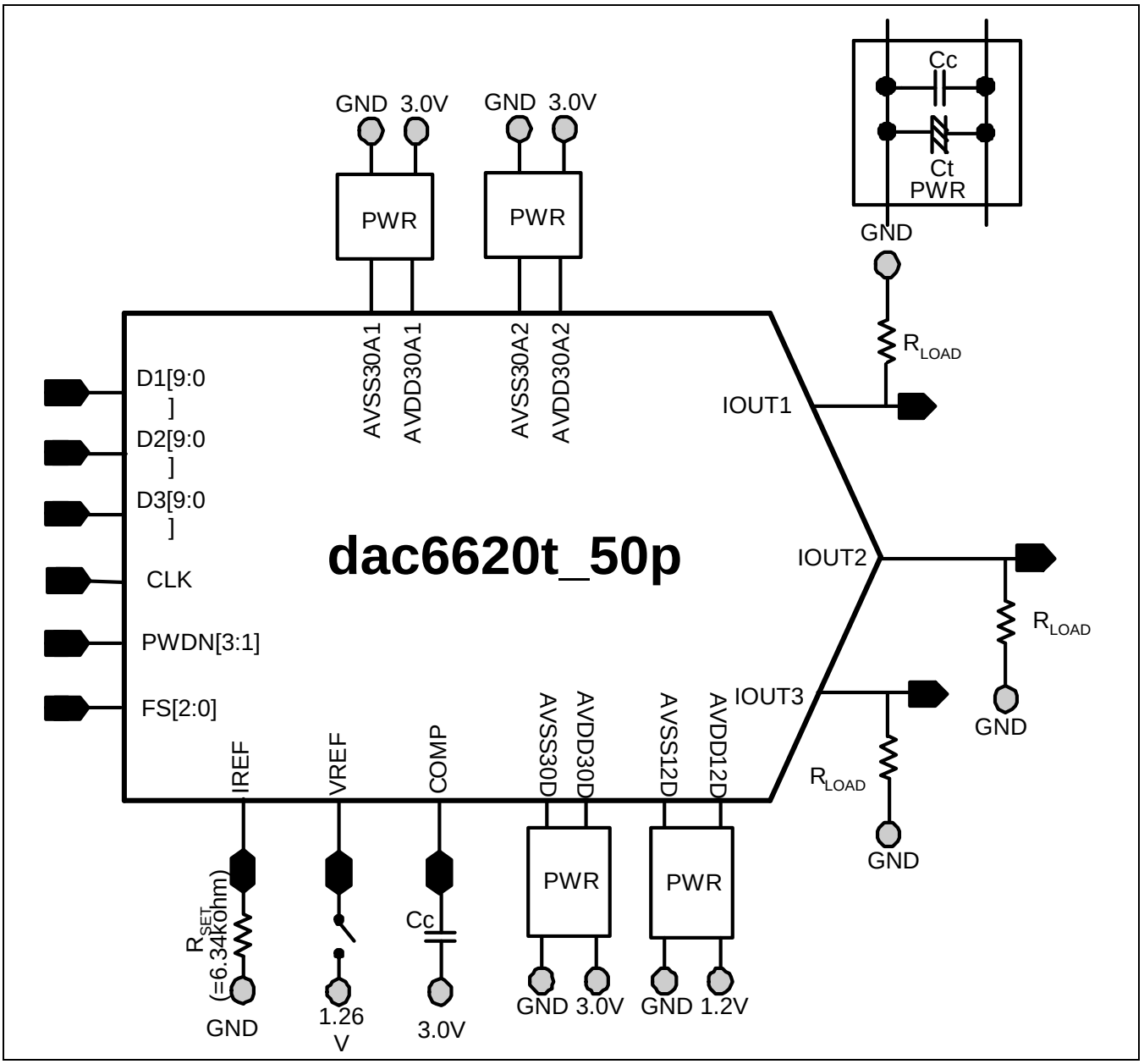


Figure 9.7-24 Recommended Board Configuration

Table 9.7-13 Recommended Parameters

Location	Description
Ct	10uF tantalum capacitor (Tolerance ±10%)
Cc	0.1uF ceramic capacitor (Tolerance ±10%)
R_{SET}	6.34 KΩ or 8.3 KΩ (Tolerance ±1%)
R_{LOAD}	200Ω (Tolerance ±1%)

7.11.1 Full Scale Voltage Modification

Table 9.7-14 Recommended RSET and RO According to Full Scale Voltage

Full Scale Voltage	RSET	RO
1.3V	6.34KΩ	200Ω
1.0V	8.3KΩ	200Ω

8 APPENDIX

8.1 VERTICAL BAR PHEOMENON

The goal of oversampling filter in TVOUT module is to obtain 54MHz sample-rate data for DAC from 13.5MHz/ 27MHz source sample-rate. The quality of interpolation result using FIR filter depends on the number of FIR filter taps. Unfortunately, ideal interpolation filter needs infinite number of taps, which cannot be implemented. Thus, practical consideration into the trade-off between interpolation quality and computational complexity is needed.

In TV application, it is the sub-carrier waveform that is the most sensitive to the quality of interpolation filter. Source rate of 13.5MHz for CVBS output is too coarse to represent the shape of sub-carrier and the quality of interpolation highly affects the shape of resulting 54 MHz sub-carriers. The error of sub-carrier induced by non-ideal interpolation filter is basically implies distortions in chrominance components. However, in the case of CVBS signal, the chrominance component and the luminance component are together to be mixed and transmitted via one channel. Then, parts of the interpolation error appears in luminance parts if the CVBS signal is separated into luminance component and chrominance components by the comb filter of TV decoder. According to an analysis on this interpolation error, the error pattern is repeated with a frequency of 102.3 kHz and the error spectrum is highly concentrated around 102.3 kHz. Thus, most of interpolation error is classified into luminance part by TV decoder and this error is shown in the shape of brightness distortion in TV monitor.

The oversampling filter in S5L8720 adopts 95-tap interpolation for CVBS signal in order to minimize the vertical bar phenomenon. Compared to 47-tap interpolation of legacy TVOUT version, the peak of vertical bar reduces by about 0.1 IRE assuming the use of 6^{th} order Butterworth analog interpolation filter.

9.8 VIDEO PROCESSOR

1 OVERVIEW

Video processor (VP) is responsible for video scaling, de-interlacing, & video post processing of TV-out data path. VP reads reconstructed YCbCr 4:2:0 video sequences from DRAM, processes the sequence, and sends it to MIXER on-the-fly as shown in Figure 9.8-1.

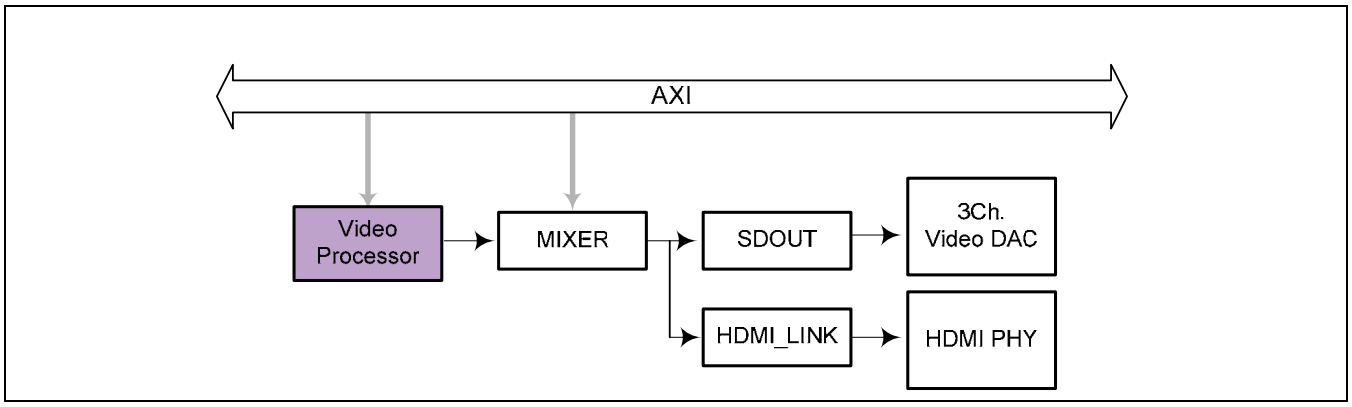


Figure 9.8-1 Video Data Path

1.1 FEATURES

Input YCbCr sequence of VP is up to 1280x720@60Hz. Basic features of VP are as follows:

- Supported Image Specification
  - ♦ BOB / 2D TILE (YUV420 NV12 type, Note : refer to MFC user's manual for 2D TILE)
  - ♦ Input source size up to 1280x720 (min : 32x4)
- Produce YCbCr 4:4:4 outputs to help MIXER to blend video and graphics
- Supports 1/4X to 16X vertical scaling with 4-tap/16-phase poly-phase filter
- Supports 1/4X to 16X horizontal scaling with 8-tap/16-phase poly-phase filter
- Supports Pan & Scan, Letterbox, and NTSC/PAL conversion using scaling
- Supports Flexible scaled video positioning within display area
- Supports 1/16 pixel resolution Pan&Scan mode
- Supports Flexible post video processing
  - ♦ Color saturation, Brightness/Contrast enhancement, Edge enhancement
  - ♦ Color space conversion between BT.601 and BT.709

2 BLOCK DIAGRAM

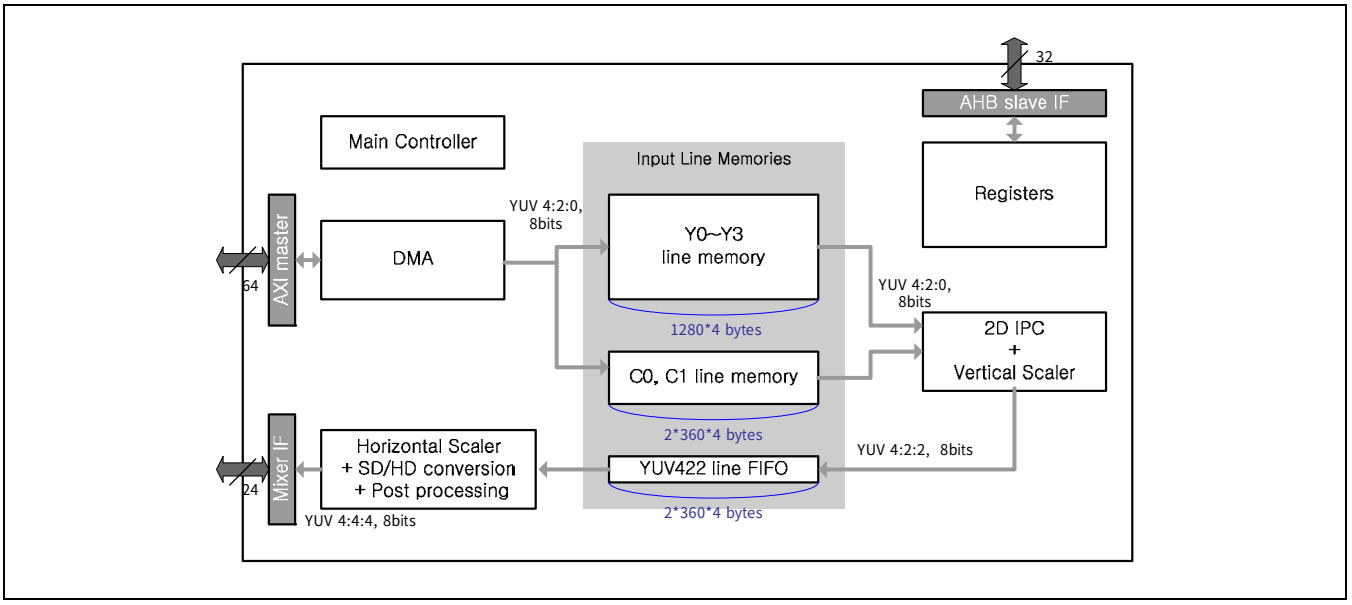


Figure 9.8-2 Block Diagram of Video Processor

DMA: DMA reads the image from memory

Input Line Memory: It stores data for processing the image.

Registers: The configuration of Video Processor

2D-IPC and Vertical Scaler : It does IPC and vertical scaling.

Horizontal Scaler : Horizontal scaling and post processing

3 FUNCTION DESCRIPTION

3.1 BOB IN VIDEO PROCESSOR

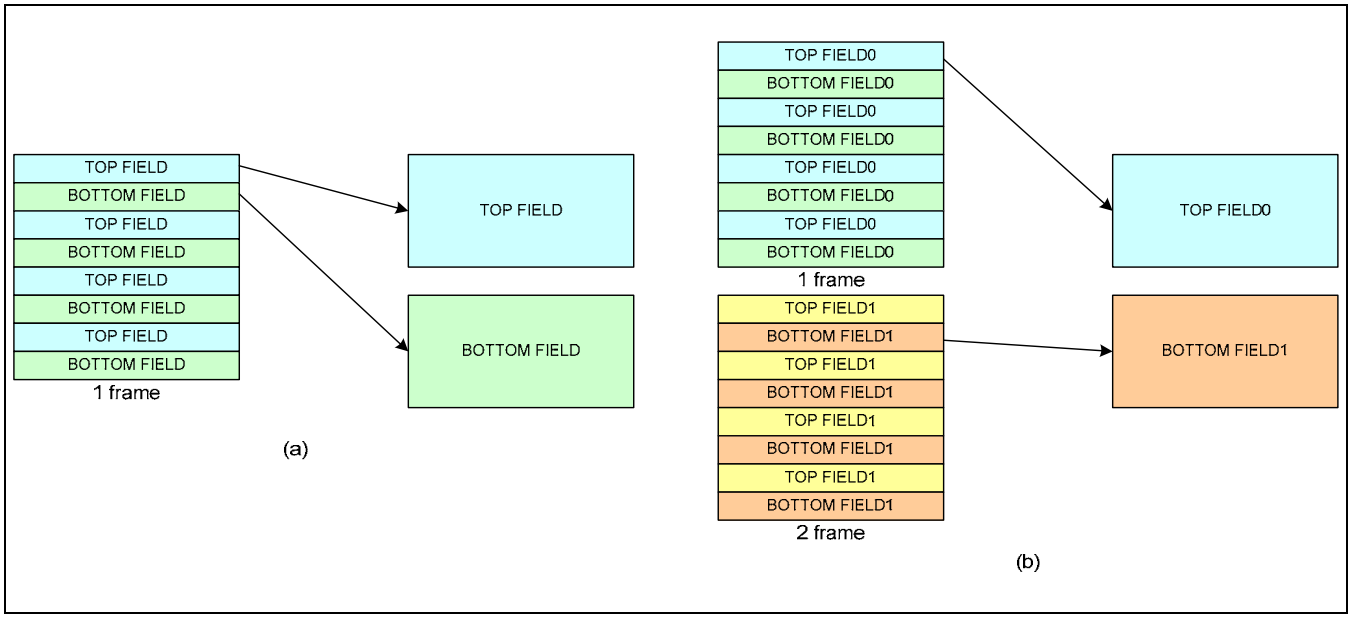


Figure 9.8-3 Data Type for BOB

In some applications, it is necessary to display an interlaced video signal on a non-interlaced display. Thus, some form of "de-interlacing" or "progressive can conversion" may be required. Video mode is one of fundamental de-interlacing algorithm. Video mode de-interlacing can be further broken down into inter-filed and intra-field processing. Particular, Intra-field processing in video mode is the simplest method for generating additional scan lines using only information in the original field. The computer industry has coined this technique as "BOB".
But, BOB in VP consists of Intra-field or inter-field. Inter-field comes from a frame as Figure 9.8-3 (a). Also, Intra-field is made up fields that come from two frames as Figure 9.8-3 (b). Video Processor in S5PC100 supports two BOB types.

3.2 IPC (INTERLACE TO PROGRESSIVE CONVERSION)

IPC(Interlace to Progressive Conversion) plays a role to convert interlaced to progressive. It is distinguished with vertical x2 scale.

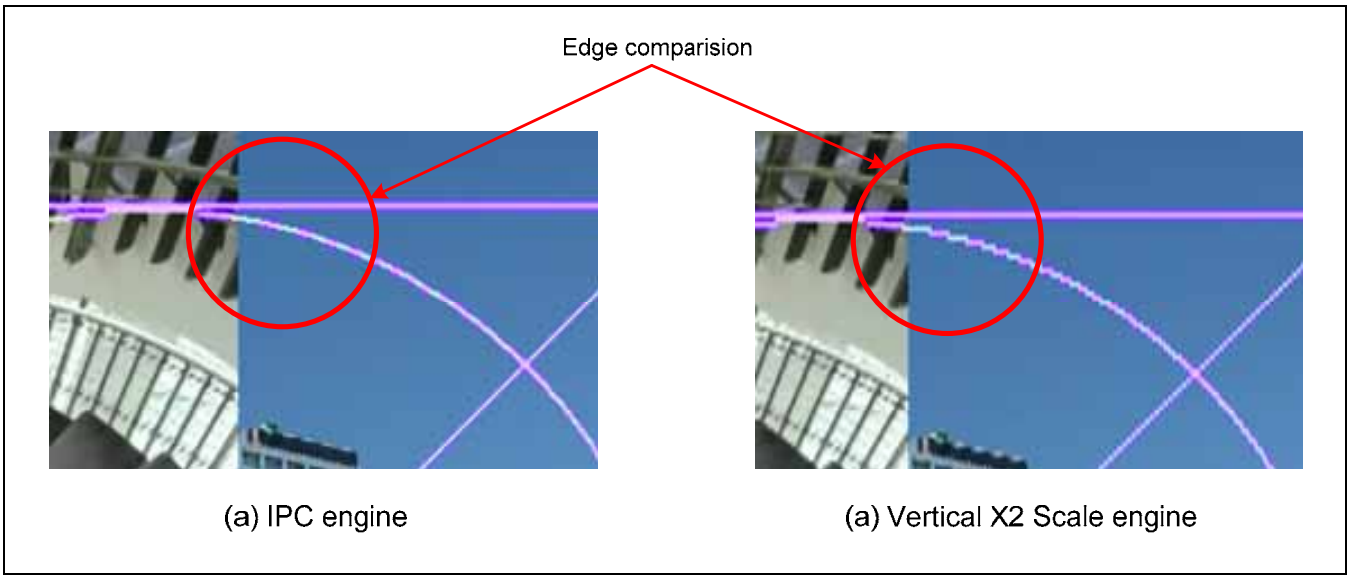


Figure 9.8-4 The Concept of BOB

IPC engine executes "Edge Detection Function" which is based on the edge diagnosis method. So, IPC can estimate edge line and display more natural image.

4 REGISTERS DESCRIPTION

4.1 REGISTER OVERVIEW

Register	Address	R/W	Description	Reset Value
VP_ENABLE	0xF010_0000	R/W	Power-Down Ready & Enable	0x0000_0002
VP_SRESET	0xF010_0004	R/W	Software Reset	0x0000_0000
VP_SHADOW_UPDATE	0xF010_0008	R/W	Shadow Register Update Enable	0x0000_0000
VP_FIELD_ID	0xF010_000C	R/W	Field ID of the "Source" Image	0x0000_0000
VP_MODE	0xF010_0010	R/W	VP Operation Mode	0x0000_0000
VP_IMG_SIZE_Y	0xF010_0014	R/W	Luminance Date Size	0x0000_0000
VP_IMG_SIZE_C	0xF010_0018	R/W	Chrominance Date Size	0x0000_0000
VP_TOP_Y_PTR	0xF010_0028	R/W	Base Address for Y of Top Field (Frame)	0x0000_0000
VP_BOT_Y_PTR	0xF010_002C	R/W	Base Address for Y of Bottom Field	0x0000_0000
VP_TOP_C_PTR	0xF010_0030	R/W	Base Address for C of Top Field(frame)	0x0000_0000
VP_BOT_C_PTR	0xF010_0034	R/W	Base Address for C of Bottom Field	0x0000_0000
VP_ENDIAN_MODE	0xF010_03CC	R/W	Big/Little Endian Mode Selection	0x0000_0000
VP_SRC_H_POSITION	0xF010_0044	R/W	Horizontal Offset in the Source Image	0x0000_0000
VP_SRC_V_POSITION	0xF010_0048	R/W	Vertical Offset in the Source Image	0x0000_0000
VP_SRC_WIDTH	0xF010_004C	R/W	Width of the Source Image	0x0000_0000
VP_SRC_HEIGHT	0xF010_0050	R/W	Height of the Source Image	0x0000_0000
VP_DST_H_POSITION	0xF010_0054	R/W	Horizontal Offset in the Display	0x0000_0000
VP_DST_V_POSITION	0xF010_0058	R/W	Vertical Offset in the Display	0x0000_0000
VP_DST_WIDTH	0xF010_005C	R/W	Width of the Display	0x0000_0000
VP_DST_HEIGHT	0xF010_0060	R/W	Height of the Display	0x0000_0000
VP_H_RATIO	0xF010_0064	R/W	Horizontal Zoom Ratio of SRC:DST	0x0000_0000
VP_V_RATIO	0xF010_0068	R/W	Vertical Zoom Ratio of SRC:DST	0x0000_0000
VP_POLY8_Y0_LL	0xF010_006C	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y0_LH	0xF010_0070	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y0_HL	0xF010_0074	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y0_HH	0xF010_0078	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y1_LL	0xF010_007C	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y1_LH	0xF010_0080	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y1_HL	0xF010_0084	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000

Register	Address	R/W	Description	Reset Value
VP_POLY8_Y1_HH	0xF010_0088	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y2_LL	0xF010_008C	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y2_LH	0xF010_0090	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y2_HL	0xF010_0094	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y2_HH	0xF010_0098	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y3_LL	0xF010_009C	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y3_LH	0xF010_00A0	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y3_HL	0xF010_00A4	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY8_Y3_HH	0xF010_00A8	R/W	8-Tap Poly-phase Filter Coefficients for Luminance Horizontal Scaling	0x0000_0000
VP_POLY4_Y0_LL	0xF010_00EC	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y0_LH	0xF010_00F0	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y0_HL	0xF010_00F4	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y0_HH	0xF010_00F8	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y1_LL	0xF010_00FC	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y1_LH	0xF010_0100	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y1_HL	0xF010_0104	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y1_HH	0xF010_0108	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y2_LL	0xF010_010C	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y2_LH	0xF010_0110	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y2_HL	0xF010_0114	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y2_HH	0xF010_0118	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y3_LL	0xF010_011C	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y3_LH	0xF010_0120	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000

Register	Address	R/W	Description	Reset Value
VP_POLY4_Y3_HL	0xF010_0124	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_Y3_HH	0xF010_0128	R/W	4-Tap Poly-phase Filter Coefficients for Luminance Vertical Scaling	0x0000_0000
VP_POLY4_C0_LL	0xF010_012C	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C0_LH	0xF010_0130	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C0_HL	0xF010_0134	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C0_HH	0xF010_0138	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C1_LL	0xF010_013C	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C1_LH	0xF010_0140	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C1_HL	0xF010_0144	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
VP_POLY4_C1_HH	0xF010_0148	R/W	4-Tap Poly-phase Filter Coefficients for Chrominance Horizontal Scaling	0x0000_0000
PP_CSC_Y2Y_COEF	0xF010_01D4	R/W	Y to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CB2Y_COEF	0xF010_01D8	R/W	CB to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CR2Y_COEF	0xF010_01DC	R/W	CR to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_Y2CB_COEF	0xF010_01E0	R/W	Y to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CB2CB_COEF	0xF010_01E4	R/W	CB to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CR2CB_COEF	0xF010_01E8	R/W	CR to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_Y2CR_COEF	0xF010_01EC	R/W	Y to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CB2CR_COEF	0xF010_01F0	R/W	CB to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CR2CR_COEF	0xF010_01F4	R/W	CR to Y CSC Coefficient Setting	0x0000_0000
PP_BYPASS	0xF010_0200	R/W	Disable the Post Image Processor	0x0000_0001
PP_SATURATION	0xF010_020C	R/W	Color Saturation Factor	0x0000_0080
PP_SHARPNESS	0xF010_0210	R/W	Control for the Edge Enhancement	0x0000_0500
PP_LINE_EQ0	0xF010_0218	R/W	Line Equation for Contrast Duration "0"	0x0000_0000
PP_LINE_EQ1	0xF010_021C	R/W	Line Equation for Contrast Duration "1"	0x0000_0000
PP_LINE_EQ2	0xF010_0220	R/W	Line Equation for Contrast Duration "2"	0x0000_0000
PP_LINE_EQ3	0xF010_0224	R/W	Line Equation for Contrast Duration "3"	0x0000_0000
PP_LINE_EQ4	0xF010_0228	R/W	Line Equation for Contrast Duration "4"	0x0000_0000
PP_LINE_EQ5	0xF010_022C	R/W	Line Equation for Contrast Duration "5"	0x0000_0000
PP_LINE_EQ6	0xF010_0230	R/W	Line Equation for Contrast Duration "6"	0x0000_0000
PP_LINE_EQ7	0xF010_0234	R/W	Line Equation for Contrast Duration "7"	0x0000_0000

Register	Address	R/W	Description	Reset Value
PP_BRIGHT_OFFSET	0xF010_0238	R/W	Brightness Offset Control for Y	0x0000_0000
PP_CSC_EN	0xF010_023C	R/W	Color Space Conversion Control	0x0000_0002
VP_VERSION_INFO	0xF010_03FC	R	VP Version Information	0x0000_0010

4.2 SHADOW REGISTER DESCRIPTION

Video processor has special registers called "Shadow Register". Software set the appropriate values to VP registers and this information are copied to the corresponding shadow registers if V-SYNC is invoked. Video processor is actually working according to these shadow registers.

Register	Address	R/W	Description	Reset Value
VP_FIELD_ID_S	0xF010_016C	R	Field ID of the "Source" Image	0x0000_0000
VP_MODE_S	0xF010_0170	R	VP Operation Mode	0x0000_0000
VP_IMG_SIZE_Y_S	0xF010_0174	R	Luminance Date Tiled Size	0x0000_0000
VP_IMG_SIZE_C_S	0xF010_0178	R	Chrominance Date Tiled Size	0x0000_0000
VP_TOP_Y_PTR_S	0xF010_0190	R	Base Address for Y of Top Field	0x0000_0000
VP_BOT_Y_PTR_S	0xF010_0194	R	Base Address for Y of Bottom Field	0x0000_0000
VP_TOP_C_PTR_S	0xF010_0198	R	Base Address for C of Top Frame	0x0000_0000
VP_BOT_C_PTR_S	0xF010_019C	R	Base Address for C of Bottom field	0x0000_0000
VP_ENDIAN_MODE_S	0xF010_03EC	R	Big/ Little Endian Mode Selection	0x0000_0000
VP_SRC_H_POSITION_S	0xF010_01AC	R	Horizontal Offset in the Source Image	0x0000_0000
VP_SRC_V_POSITION_S	0xF010_01B0	R	Vertical Offset in the Source Image	0x0000_0000
VP_SRC_WIDTH_S	0xF010_01B4	R	Width of the Source Image	0x0000_0000
VP_SRC_HEIGHT_S	0xF010_01B8	R	Height of the Source Image	0x0000_0000
VP_DST_H_POSITION_S	0xF010_01BC	R	Horizontal Offset in the Display	0x0000_0000
VP_DST_V_POSITION_S	0xF010_01C0	R	Vertical Offset in the Display	0x0000_0000
VP_DST_WIDTH_S	0xF010_01C4	R	Width of the Display	0x0000_0000
VP_DST_HEIGHT_S	0xF010_01C8	R	Height of the Display	0x0000_0000
VP_H_RATIO_S	0xF010_01CC	R	Horizontal Zoom Ratio of SRC:DST	0x0000_0000
VP_V_RATIO_S	0xF010_01D0	R	Vertical Zoom Ratio of SRC:DST	0x0000_0000
PP_BYPASS_S	0xF010_0258	R	Disable the Post Image Processor	0x0000_0000
PP_SATURATION_S	0xF010_025C	R	Color Saturation Factor	0x0000_0000
PP_SHARPNESS_S	0xF010_0260	R	Control for the Edge Enhancement	0x0000_0000
PP_LINE_EQ0_S	0xF010_0268	R	Line Equation for Contrast Duration "0"	0x0000_0000
PP_LINE_EQ1_S	0xF010_026C	R	Line Equation for Contrast Duration "1"	0x0000_0000
PP_LINE_EQ2_S	0xF010_0270	R	Line Equation for Contrast Duration "2"	0x0000_0000
PP_LINE_EQ3_S	0xF010_0274	R	Line Equation for Contrast Duration "3"	0x0000_0000
PP_LINE_EQ4_S	0xF010_0278	R	Line Equation for Contrast Duration "4"	0x0000_0000
PP_LINE_EQ5_S	0xF010_027C	R	Line Equation for Contrast Duration "5"	0x0000_0000
PP_LINE_EQ6_S	0xF010_0280	R	Line Equation for Contrast Duration "6"	0x0000_0000
PP_LINE_EQ7_S	0xF010_0284	R	Line Equation for Contrast Duration "7"	0x0000_0000
PP_BRIGHT_OFFSET_S	0xF010_0288	R	Brightness Offset Control for Y	0x0000_0000
PP_CSC_EN_S	0xF010_028C	R	Color Space Conversion Control	0x0000_0000

Register	Address	R/W	Description	Reset Value
PP_CSC_Y2Y_COEF_S	0xF010_0290	R	Y to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CB2Y_COEF_S	0xF010_0294	R	CB to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CR2Y_COEF_S	0xF010_0298	R	CR to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_Y2CB_COEF_S	0xF010_029C	R	Y to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CB2CB_COEF_S	0xF010_02A0	R	CB to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CR2CB_COEF_S	0xF010_02A4	R	CR to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_Y2CR_COEF_S	0xF010_02A8	R	Y to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CB2CR_COEF_S	0xF010_02AC	R	CB to Y CSC Coefficient Setting	0x0000_0000
PP_CSC_CR2CR_COEF_S	0xF010_02B0	R	CR to Y CSC Coefficient Setting	0x0000_0000

4.3 DETAILED DESCRIPTION

4.3.1 Video Processor Enable/Disable Control Register (VP_ENABLE, Address = 0xF010_0000)

VP_ENABLE	Bit	Description	RW	Reset Value
Reserved	[31:3]	Reserved, read as zero, do not modify	R/W	0
VP_ON_S	[2]	This bit is read-only. Shadow bit of the bit [0]	R	0
Reserved	[1]	Reserved, read as zero, do not modify	R/W	1
VP_ON	[0]	This bit is read-write. 0 = Disabled 1 = Enabled NOTE: The SFRs of Video Processor and Image Mixer is updated by Vertical Sync of TVENC's Timing Generator. So, those are configured before this bit is enabled. The sequence of enabling TVSS is following as: "VP -> MIXER TVENC(HDMI)". Also, because of the same reason, the disabling sequence is following as : "VP -> MIXER -> TVNEC(HDMI)".	R/W	0

4.3.2 Video Processor Software Reset (VP_SRESET, R/W, Address = 0xF010_0004)

VP_SRESET	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
VP_SRESET	[0]	0 = Software reset is set and the last soft reset is complete. 1 = VP is processing software reset sequence.	0

4.3.3 Video Processor Shadow Register Update Enable Control Register (VP_SHADOW_UPDATE, W, Address = 0xF010_ 0008)

VP_SHADOW_UPDATE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
VP_SHADOW_UPDATE	[0]	0 = Shadow registers are not updated at the rising edge of vertical sync. 1 = Shadow registers are updated and this register is cleared by H/W at the rising edge of vertical sync. (Shadow registers are listed in SHADOW REGISTER MAP table)	0

4.3.4 Video Processor Input Field ID Control Register (VP_FIELD_ID, R/W, Address = 0xF010_ 000C)

VP_FIELD_ID	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
VP_FILED_ID	[0]	When VP_MODE[2] is set to 'high', this bit shows current FIELD information. Otherwise, when VP_MODE[2] is set to 'low', this control the pointer of top and bottom field. 0 = Top field 1 = Bottom field	0

4.3.5 Video Processor Operation Mode Control Register (VP_MODE, R/W, Address = 0xF010_ 0010)

VP_MODE	Bit	Description	Reset Value
Reserved	[31:6]	Reserved, read as zero, do not modify	0
LINE_SKIP	[5]	This bit can control DMA operation. If it is set to '1', DMA skips a line per two line while it reads line data. 0 = OFF 1 = ON	0
MEM_MODE	[4]	0 = Linear Mode 1 = 2D-Tile Mode (refer to MFC user's manual)	0
CROMA_EXPANSION	[3]	If it is set to '0', only refer to the chrominance of TOP filed. But set to '1', it uses the chrominance both TOP and BOTTOM. 0 = Using only C_TOP_PTR 1 = Using both C_TOP_PTR and C_BOT_PTR	0

FIELD_ID_AUTO_TOGGLING	[2]	0 = FIELD_ID is defined by user 1 = FIELD_ID is automatically toggled by V_SYNC DMA base address is changed by this bit. (**NOTE:** VP_FIELD_ID_S register is toggled if this bit is 1, not VP_FIELD_ID)	0
2D_IPC	[1]	Interlace to progressive conversion. VP displays progressive scan as using one filed image 0 = Disables 2D-IPC 1 = Enables 2D-IPC	0
Reserved	[0]	Reserved. It must be '0'	0

The guide of Configuration

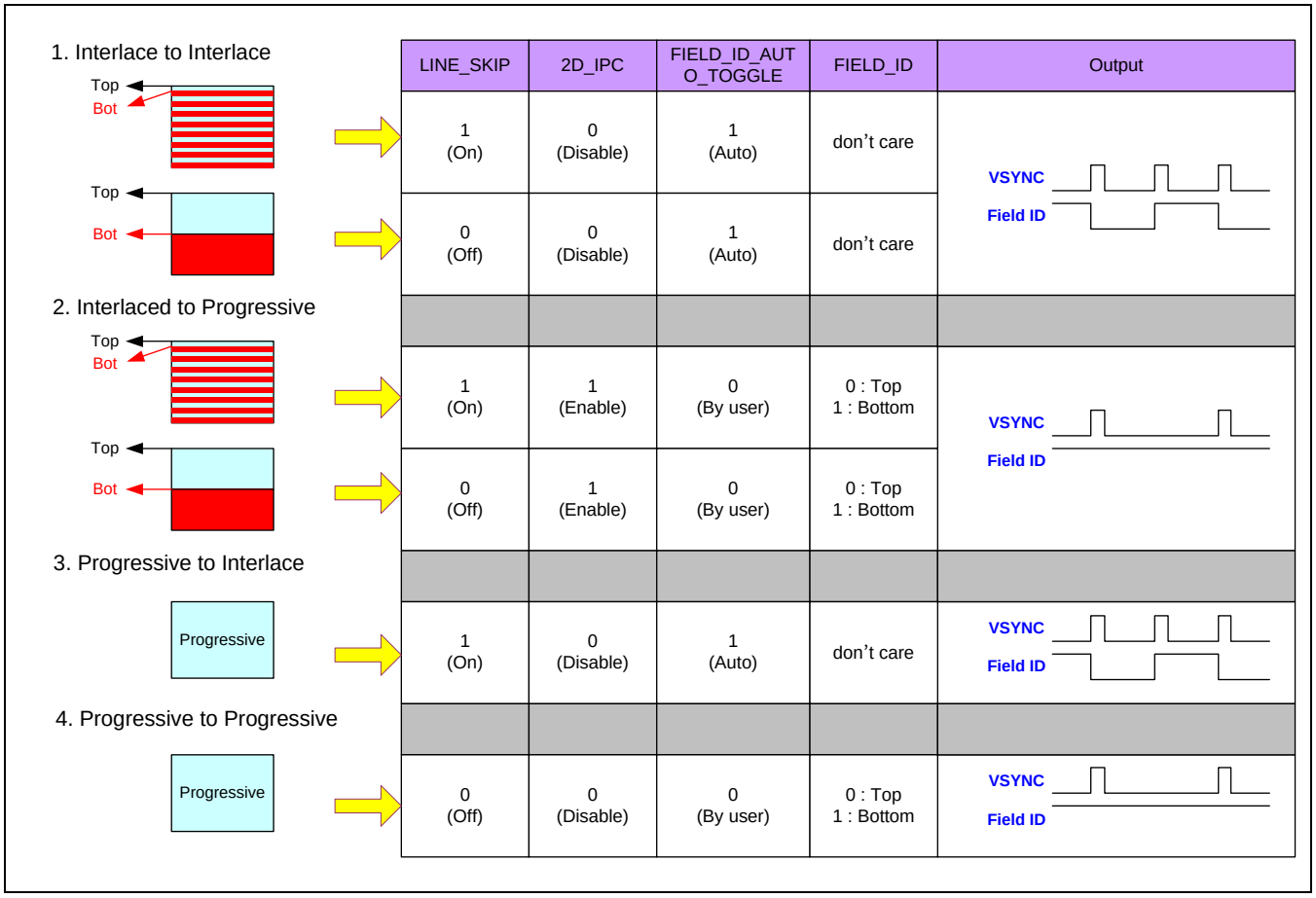


Figure 9.8-5 The Examples of Usage Cases

Video Processor Luminance Image Size Control Register (VP_IMG_SIZE_Y, R/W, Address = 0xF010_ 0014)

VP_IMG_SIZE_Y	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
VP_IMG_HSIZE_Y	[29:16]	Horizontal size of image (8~8192). (Without minus 1). LSB [2:0] must be 3'b000 for 64 bit interface. Zero value and values greater than 8192 are not allowed.	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
VP_IMG_VSIZE_Y	[13:0]	Vertical size of image (1 ~ 8192)	0

4.3.6 Video Processor Chrominance Image Size Control Register (VP_IMG_SIZE_C, R/W, Address = 0xF010_ 0018)

VP_ IMG _SIZE_C	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
VP_IMG_HSIZE_C	[29:16]	Horizontal size of image (8~8192). (Without minus 1). LSB [2:0] must be 3'b000 for 64 bit interface. Zero value and values greater than 8192 are not allowed.	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
VP_IMG_VSIZE_C	[13:0]	Vertical size of image (1 ~ 8192)	0

4.3.7 Video Processor Top Luminance Picture Pointer Control Register (VP_TOP_Y_PTR, R/W, Address = 0xF010_ 0028)

VP_TOP_Y_PTR	Bit	Description	Reset Value
VP_TOP_Y_PTR	[31:0]	Base address for luminance of top field. It should be integer multiples of 8. (LSB[2:0] must be 3'b000)	0

4.3.8 Video Processor Bottom Luminance Picture Pointer Control Register (VP_BOT_Y_PTR, R/W, Address = 0xF010_ 002C)

VP_TOP_C_PTR	Bit	Description	Reset Value
VP_BOT_Y_PTR	[31:0]	Base address for luminance of bottom field. It should be integer multiples of 8. (LSB[2:0] must be 3'b000) If 2D-TILE mode is enable, VP_BOT_Y_PTR = VP_TOP_Y_PTR + 0x40	0

4.3.9 Video Processor Top Chrominance Picture Pointer Control Register (VP_TOP_C_PTR, R/W, Address = 0xF010_0030)

VP_CR_PTR	Bit	Description	Reset Value
VP_TOP_C_PTR	[31:0]	Base address for chrominance of top field. It should be integer multiples of 8. (LSB[2:0] must be 3'b000)	0

4.3.10 Video Processor Bottom Chrominance Picture Pointer Control Register (VP_BOT_C_PTR, R/W, Address = 0xF010_ 0034)

VP_BOT_Y_PTR	Bit	Description	Reset Value
VP_BOT_C_PTR	[31:0]	Base address for chrominance of bottom field. It should be integer multiples of 8. (LSB[2:0] must be 3'b000) If 2D-TILE mode is enable, VP_BOT_C_PTR = VP_TOP_C_PTR + 0x40	0

4.3.11 Video Processor Picture Endian Mode Control Register (VP_ENDIAN_MODE, R/W, Address = 0xF010_ 03CC)

VP_ENDIAN_MODE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
VP_ENDIAN_MODE	[0]	0 = Big Endian 1 = Little Endian Refer to Figure 37-3	0

NOTE:

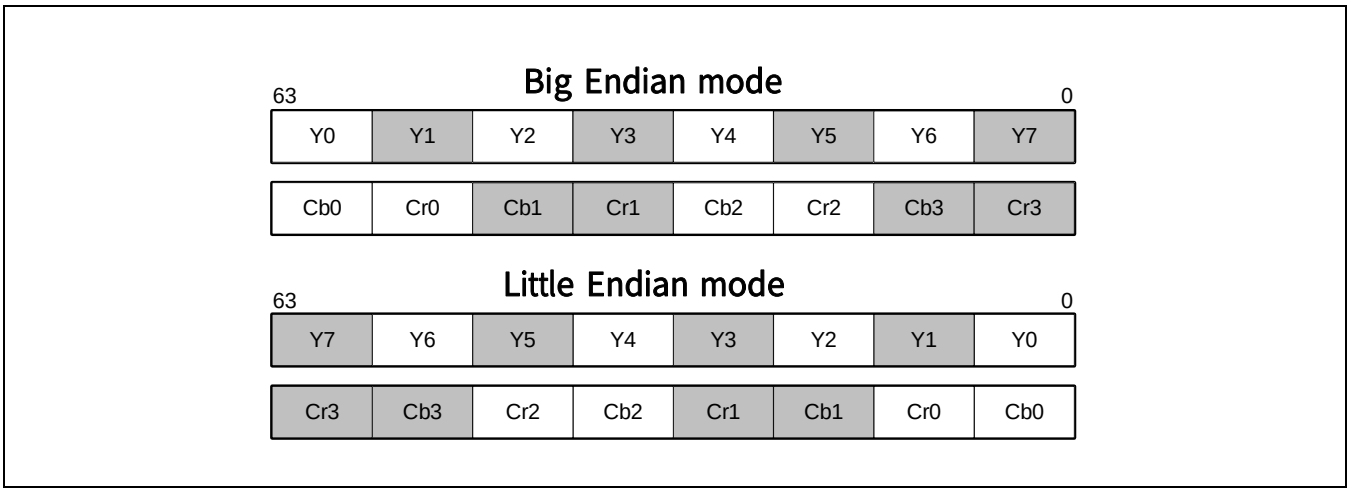


Figure 9.8-6 Endian Mode

4.3.12 Video Processor Horizontal Offset of Source Image Control Register (VP_SRC_H_POSITION, R/W, Address = 0xF010_ 0044)

VP_SRC_H_POSITION	Bit	Description	Reset Value
Reserved	[31:15]	Reserved, read as zero, do not modify	0
VP_SRC_H_POSITION	[14:0]	Horizontal offset in the source image - (11.4) format * For source image cropping, VP_SRC_H_POSITION + VP_SRC_WIDTH should be less than or equal to VP_IMG_HSIZE_Y * Note : (11.4) format means that - '11' is a integer. - '4' is a fraction. Example) In case of H Position = 4 4(0x4(h) = 0100(b)) is integer. Because of 4bit fraction, 0100(b) is had to do 4 time left shift operation. As a result, register value is 4 * 2^4 = 64 = 0x40.	0

4.3.13 Video Processor Vertical Offset of Source Image Control Register (VP_SRC_V_POSITION, R/W, Address = 0xF010_ 0048)

VP_SRC_V_POSITION	Bit	Description	Reset Value
Reserved	[31:10]	Reserved, read as zero, do not modify	0

VP_SRC_V_POSITION	[9:0]	Vertical offset in the source image. This value should be in the range between 0 and VP_SRC_HEIGHT. If LINE_SKIP is 1, VP_SRC_V_POSITION should be a half of that if LINE_SKIP is 0.	0

4.3.14

4.3.15 Video Processor Width of Source Image Control Register (VP_SRC_WIDTH, R/W, Address = 0xF010_ 004C)

VP_SRC_WIDTH	Bit	Description	Reset Value
Reserved	[31:11]	Reserved, read as zero, do not modify	0
VP_SRC_WIDTH	[10:0]	Width of the source image (MIN : 32)	0

4.3.16 Video Processor Height of Source Image Control Register (VP_SRC_HEIGHT, R/W, Address = 0xF010_ 0050)

VP_SRC_HEIGHT	Bit	Description	Reset Value
Reserved	[31:10]	Reserved, read as zero, do not modify	0
VP_SRC_HEIGHT	[9:0]	Height of the source image If LINE_SKIP is 1, VP_SRC_HEIGHT should be a half of that if LINE_SKIP is 0. (MIN : 4)	0

4.3.17 Video Processor Horizontal Offset of Destination Image Control Register (VP_DST_H_POSITION, R/W, Address = 0xF010_ 0054)

VP_DST_H_POSITION	Bit	Description	Reset Value
Reserved	[31:11]	Reserved, read as zero, do not modify	0
VP_DST_H_POSITION	[10:0]	Horizontal offset in the display	0

4.3.18 Video Processor Vertical Offset of Destination Image Control Register (VP_DST_V_POSITION, R/W, Address = 0xF010_ 0058)

VP_DST_V_POSITION	Bit	Description	Reset Value
Reserved	[31:10]	Reserved, read as zero, do not modify	0
VP_DST_V_POSITION	[9:0]	Vertical offset in the display	0

4.3.19 Video Processor Width of Destination Image Control Register (VP_DST_WIDTH, R/W, Address = 0xF010_ 005C)

VP_DST_WIDTH	Bit	Description	Reset Value
Reserved	[31:11]	Reserved, read as zero, do not modify	0
VP_DST_WIDTH	[10:0]	Width of the display	0

4.3.20 Video Processor Height of Destination Image Control Register (VP_DST_HEIGHT, R/W, Address = 0xF010_ 0060)

VP_DST_HEIGHT	Bit	Description	Reset Value
Reserved	[31:10]	Reserved, read as zero, do not modify	0
VP_DST_HEIGHT	[9:0]	Height of the display	0

NOTE:

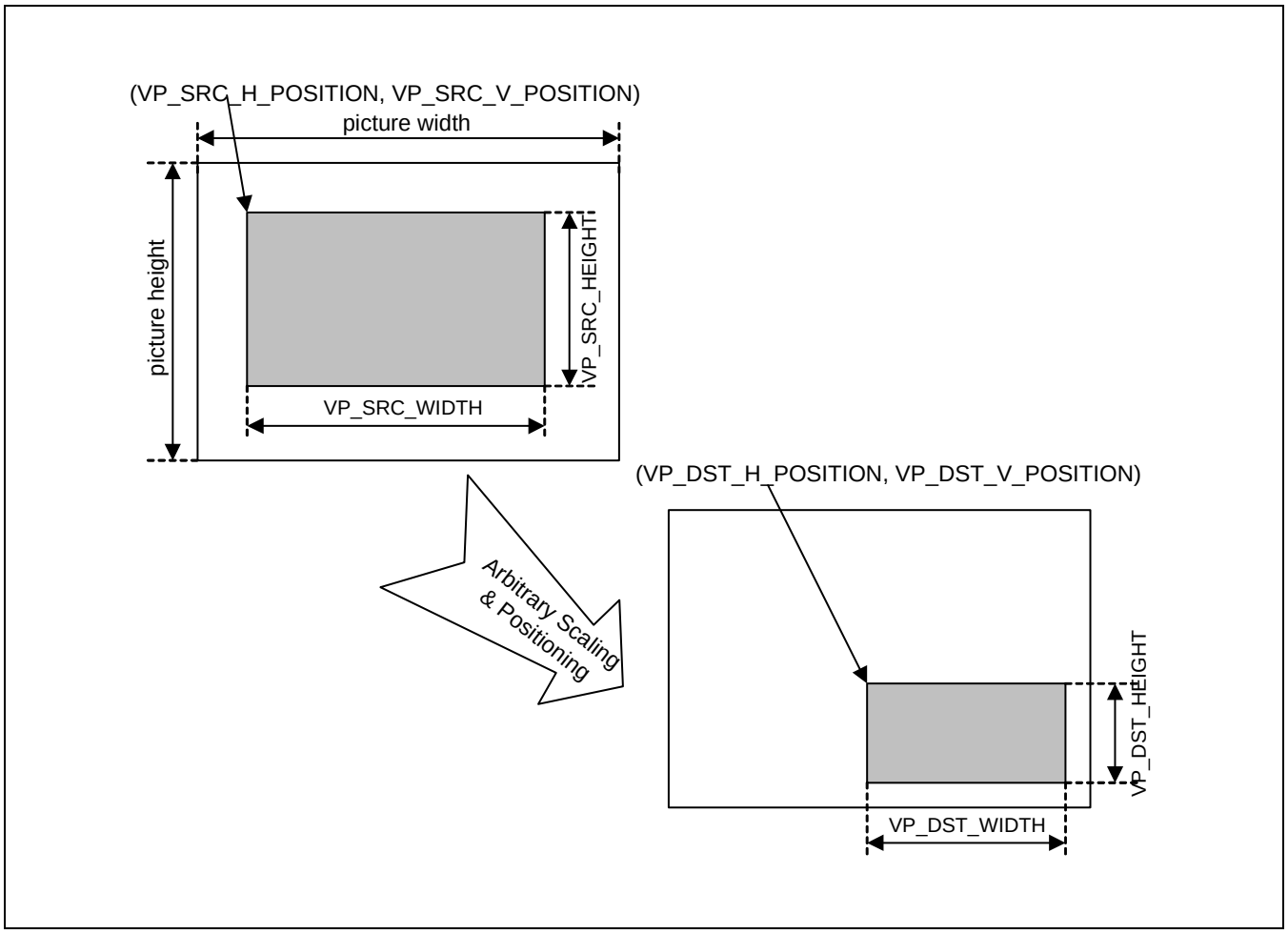


Figure 9.8-7 Video Scaling & Positioning on TV Display

4.3.21 Video Processor Horizontal Zoom Ratio (VP_H_RATIO, R/W, Address = 0xF010_ 0064)

VP_H_RATIO	Bit	Description	Reset Value
Reserved	[31:19]	Reserved, read as zero, do not modify	0
VP_H_RATIO	[18:0]	Horizontal zoom ratio of SRC:DST - 3.16 format * Note : (3.16) format means that - '3' is a integer.(Located in MSB) - '16' is a fraction.(Located in LSB) Example) SRC : DST = 1 : 2 Because of 16bit fraction, it is had to do 16 time left shift operation. As a result, register value is 1/2 * 2^16 = 0x8000	0

4.3.22 Video Processor Vertical Zoom Ratio (VP_V_RATIO, R/W, Address = 0xF010_ 0068)

VP_V_RATIO	Bit	Description	Reset Value
Reserved	[31:19]	Reserved, read as zero, do not modify	0
VP_V_RATIO	[18:0]	Vertical zoom ratio of SRC:DST - 3.16 format This register should be as follows. (1) BOB mode, IPC disable VP_V_RATIO = SRC / DST (2) BOB mode, IPC enable VP_V_RATIO = 2 * SRC / DST (This is because destination line number is doubled by de-interlacing process itself) * Note : (3.16) format means that - '3' is a integer. (Located in MSB) - '16' is a fraction. (Located in LSB) Example) SRC : DST = 1 : 2 Because of 16bit fraction, it is had to do 16 time left shift operation. As a result, register value is 1/2 * 2^16 = 0x8000	0

4.3.23 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y0_LL, R/W, Address = 0xF010_ 006C

VP_POLY8_Y0_LL	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph0	[26:24]	Poly-phase Filter Coefficients	0
Reserved	[23:19]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph1	[18:16]	Poly-phase Filter Coefficients	0
Reserved	[15:11]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph2	[10:8]	Poly-phase Filter Coefficients	0
Reserved	[7:3]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph3	[2:0]	Poly-phase Filter Coefficients	0

4.3.24 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y0_LH, R/W, Address = 0xF010_ 0070)

VP_POLY8_Y0_LH	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph4	[26:24]	Poly-phase Filter Coefficients	0
Reserved	[23:19]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph5	[18:16]	Poly-phase Filter Coefficients	0
Reserved	[15:11]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph6	[10:8]	Poly-phase Filter Coefficients	0
Reserved	[7:3]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph7	[2:0]	Poly-phase Filter Coefficients	0

4.3.25 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y0_HL, R/W, Address = 0xF010_ 0074)

VP_POLY8_Y0_HL	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph8	[26:24]	Poly-phase Filter Coefficients	0
Reserved	[23:19]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph9	[18:16]	Poly-phase Filter Coefficients	0
Reserved	[15:11]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph10	[10:8]	Poly-phase Filter Coefficients	0
Reserved	[7:3]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph11	[2:0]	Poly-phase Filter Coefficients	0

4.3.26 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y0_HH, R/W, Address = 0xF010_ 0078)

VP_POLY8_Y0_HH	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph12	[26:24]	Poly-phase Filter Coefficients	0
Reserved	[23:19]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph13	[18:16]	Poly-phase Filter Coefficients	0
Reserved	[15:11]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph14	[10:8]	Poly-phase Filter Coefficients	0
Reserved	[7:3]	Reserved, read as zero, do not modify	0
vp_poly8_y0_ph15	[2:0]	Poly-phase Filter Coefficients	0

4.3.27 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y1_LL, R/W, Address = 0xF010_ 007C)

VP_POLY8_Y1_LL	Bit	Description	Reset Value
Reserved	[31:29]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph0	[28:24]	Poly-phase Filter Coefficients	0
Reserved	[23:21]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph1	[20:16]	Poly-phase Filter Coefficients	0
Reserved	[15:13]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph2	[12:8]	Poly-phase Filter Coefficients	0
Reserved	[7:5]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph3	[4:0]	Poly-phase Filter Coefficients	0

4.3.28 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y1_LH, R/W, Address = 0xF010_ 0080)

VP_POLY8_Y1_LH	Bit	Description	Reset Value
Reserved	[31:29]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph4	[28:24]	Poly-phase Filter Coefficients	0
Reserved	[23:21]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph5	[20:16]	Poly-phase Filter Coefficients	0
Reserved	[15:13]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph6	[12:8]	Poly-phase Filter Coefficients	0
Reserved	[7:5]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph7	[4:0]	Poly-phase Filter Coefficients	0

4.3.29 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y1_HL, R/W, Address = 0xF010_ 0084)

VP_POLY8_Y1_HL	Bit	Description	Reset Value
Reserved	[31:29]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph8	[28:24]	Poly-phase Filter Coefficients	0
Reserved	[23:21]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph9	[20:16]	Poly-phase Filter Coefficients	0
Reserved	[15:13]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph10	[12:8]	Poly-phase Filter Coefficients	0
Reserved	[7:5]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph11	[4:0]	Poly-phase Filter Coefficients	0

4.3.30 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y1_HH, R/W, Address = 0xF010_ 0088)

VP_POLY8_Y1_HH	Bit	Description	Reset Value
Reserved	[31:29]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph12	[28:24]	Poly-phase Filter Coefficients	0
Reserved	[23:21]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph13	[20:16]	Poly-phase Filter Coefficients	0
Reserved	[15:13]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph14	[12:8]	Poly-phase Filter Coefficients	0
Reserved	[7:5]	Reserved, read as zero, do not modify	0
vp_poly8_y1_ph15	[4:0]	Poly-phase Filter Coefficients	0

4.3.31 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y2_LL, R/W, Address = 0xF010_ 008C)

VP_POLY8_Y2_LL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph0	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph1	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph2	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph3	[6:0]	Poly-phase Filter Coefficients	0

4.3.32 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y2_LH, R/W, Address = 0xF010_ 0090)

VP_POLY8_Y2_LH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph4	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph5	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph6	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph7	[6:0]	Poly-phase Filter Coefficients	0

4.3.33 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y2_HL, R/W, Address = 0xF010_ 0094)

VP_POLY8_Y2_HL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph8	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph9	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph10	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph11	[6:0]	Poly-phase Filter Coefficients	0

4.3.34 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y2_HH, R/W, Address = 0xF010_ 0098)

VP_POLY8_Y2_HH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph12	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph13	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph14	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y2_ph15	[6:0]	Poly-phase Filter Coefficients	0

4.3.35 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y3_LL, R/W, Address = 0xF010_ 009C)

VP_POLY8_Y3_LL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph0	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph1	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph2	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph3	[6:0]	Poly-phase Filter Coefficients	0

4.3.36 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y3_LH R/W, Address = 0xF010_ 00A0)

VP_POLY8_Y3_LH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph4	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph5	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph6	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph7	[6:0]	Poly-phase Filter Coefficients	0

4.3.37 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y3_HL, R/W, Address = 0xF010_ 00A4)

VP_POLY8_Y3_HL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph8	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph9	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph10	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph11	[6:0]	Poly-phase Filter Coefficients	0

4.3.38 Video Processor Luminance 8-tap Poly-phase Filter Coefficients (VP_POLY8_Y3_HH, R/W, Address = 0xF010_ 00A8)

VP_POLY8_Y3_HH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph12	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph13	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph14	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly8_y3_ph15	[6:0]	Poly-phase Filter Coefficients	0

4.3.39 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y0_LL, R/W, Address = 0xF010_ 00EC)

VP_POLY4_Y0_LL	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph0	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph1	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph2	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph3	[5:0]	Poly-phase Filter Coefficients	0

4.3.40 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y0_LH, R/W, Address = 0xF010_ 00F0)

VP_POLY4_Y0_LH	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph4	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph5	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph6	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph7	[5:0]	Poly-phase Filter Coefficients	0

4.3.41 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y0_HL, R/W, Address = 0xF010_ 00F4)

VP_POLY4_Y0_HL	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph8	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph9	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph10	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph11	[5:0]	Poly-phase Filter Coefficients	0

4.3.42 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y0_HH, R/W, Address = 0xF010_ 00F8)

VP_POLY4_Y0_HH	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph12	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph13	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph14	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y0_ph15	[5:0]	Poly-phase Filter Coefficients	0

4.3.43 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y1_LL, R/W, Address = 0xF010_ 00FC)

VP_POLY4_Y1_LL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph0	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph1	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph2	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph3	[6:0]	Poly-phase Filter Coefficients	0

4.3.44 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y1_LH, R/W, Address = 0xF010_ 0100)

VP_POLY4_Y1_LH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph4	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph5	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph6	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph7	[6:0]	Poly-phase Filter Coefficients	0

4.3.45 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y1_HL, R/W, Address = 0xF010_ 0104)

VP_POLY4_Y1_HL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph8	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph9	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph10	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph11	[6:0]	Poly-phase Filter Coefficients	0

4.3.46 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y1_HH, R/W, Address = 0xF010_ 0108)

VP_POLY4_Y1_HH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph12	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph13	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph14	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y1_ph15	[6:0]	Poly-phase Filter Coefficients	0

4.3.47 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y2_LL, R/W, Address = 0xF010_ 010C)

VP_POLY4_Y2_LL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph0	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph1	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph2	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph3	[6:0]	Poly-phase Filter Coefficients	0

4.3.48 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y2_LH, R/W, Address = 0xF010_ 0110)

VP_POLY4_Y2_LH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph4	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph5	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph6	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph7	[6:0]	Poly-phase Filter Coefficients	0

4.3.49 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y2_HL, R/W, Address = 0xF010_ 0114)

VP_POLY4_Y2_HL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph8	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph9	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph10	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph11	[6:0]	Poly-phase Filter Coefficients	0

4.3.50 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y2_HH, R/W, Address = 0xF010_ 0118)

VP_POLY4_Y2_HH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph12	[30:24]	Poly-phase Filter Coefficients	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph13	[22:16]	Poly-phase Filter Coefficients	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph14	[14:8]	Poly-phase Filter Coefficients	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_y2_ph15	[6:0]	Poly-phase Filter Coefficients	0

4.3.51 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y3_LL, R/W, Address = 0xF010_ 011C)

VP_POLY4_Y3_LL	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph0	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph1	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph2	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph3	[5:0]	Poly-phase Filter Coefficients	0

4.3.52 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y3_LH, R/W, Address = 0xF010_ 0120)

VP_POLY4_Y3_LH	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph4	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph5	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph6	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph7	[5:0]	Poly-phase Filter Coefficients	0

4.3.53 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y3_HL, R/W, Address = 0xF010_ 0124)

VP_POLY4_Y3_HL	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph8	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph9	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph10	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph11	[5:0]	Poly-phase Filter Coefficients	0

4.3.54 Video Processor Luminance 4-tap Poly-phase Filter Coefficients (VP_POLY4_Y3_HH, R/W, Address = 0xF010_ 0128)

VP_POLY4_Y3_LL	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph12	[29:24]	Poly-phase Filter Coefficients	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph13	[21:16]	Poly-phase Filter Coefficients	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph14	[13:8]	Poly-phase Filter Coefficients	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_y3_ph15	[5:0]	Poly-phase Filter Coefficients	0

4.3.55 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C0_LL, R/W, Address = 0xF010_ 012C)

Unlike VP_POLY4_Y registers, there are only a half of the coefficient registers for horizontal Chroma Scaler. The coefficients are assumed to be symmetric so that only half of them are kept. Some parts of them are unsigned integer and the other parts are signed integer. You must be careful while setting them.

VP_POLY4_C0_LL	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph0	[30:24]	Signed 7-bit integer (-64~63)	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph1	[22:16]	Signed 7-bit integer (-64~63)	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph2	[14:8]	Signed 7-bit integer (-64~63)	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph3	[6:0]	Signed 7-bit integer (-64~63)	0

4.3.56 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C0_LH, R/W, Address = 0xF010_ 0130)

VP_POLY4_C0_LH	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph4	[29:24]	Signed 6-bit integer (-32~31)	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph5	[21:16]	Signed 6-bit integer (-32~31)	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph6	[13:8]	Signed 6-bit integer (-32~31)	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph7	[5:0]	Signed 6-bit integer (-32~31)	0

4.3.57 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C0_HL, R/W, Address = 0xF010_ 0134)

VP_POLY4_C0_HL	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph8	[29:24]	Signed 6-bit integer (-32~31)	0
Reserved	[23:22]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph9	[21:16]	Signed 6-bit integer (-32~31)	0
Reserved	[15:14]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph10	[13:8]	Signed 6-bit integer (-32~31)	0
Reserved	[7:6]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph11	[5:0]	Signed 6-bit integer (-32~31)	0

4.3.58 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C0_HH, R/W, Address = 0xF010_ 0138)

VP_POLY4_C0_HH	Bit	Description	Reset Value
Reserved	[31:29]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph12	[28:24]	Signed 5-bit integer (-16~15)	0
Reserved	[23:21]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph13	[20:16]	Signed 5-bit integer (-16~15)	0
Reserved	[15:13]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph14	[12:8]	Signed 5-bit integer (-16~15)	0
Reserved	[7:5]	Reserved, read as zero, do not modify	0
vp_poly4_c0_ph15	[4:0]	Signed 5-bit integer (-16~15)	0

4.3.59 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C1_LL, R/W, Address = 0xF010_ 013C)

VP_POLY4_C1_LL	Bit	Description	Reset Value
vp_poly4_c1_ph0	[31:24]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph1	[23:16]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph2	[15:8]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph3	[7:0]	Unsigned 8-bit integer (0~255)	0

4.3.60 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C1_LH, R/W, Address = 0xF010_ 0140)

VP_POLY4_C1_LH	Bit	Description	Reset Value
vp_poly4_c1_ph4	[31:24]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph5	[23:16]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph6	[15:8]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph7	[7:0]	Unsigned 8-bit integer (0~255)	0

4.3.61 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C1_HL, R/W, Address = 0xF010_ 0144)

VP_POLY4_C1_HL	Bit	Description	Reset Value
vp_poly4_c1_ph8	[31:24]	Unsigned 8-bit integer (0~255)	0
vp_poly4_c1_ph9	[23:16]	Signed 8-bit integer (-128~127)	0
vp_poly4_c1_ph10	[15:8]	Signed 8-bit integer (-128~127)	0
vp_poly4_c1_ph11	[7:0]	Signed 8-bit integer (-128~127)	0

4.3.62 Video Processor Chrominance 4-tap Poly-phase Filter Coefficients (VP_POLY4_C1_HH, R/W, Address = 0xF010_ 0148)

VP_POLY4_C1_HH	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	0
vp_poly4_c1_ph12	[30:24]	Signed 7-bit integer (-64~63)	0
Reserved	[23]	Reserved, read as zero, do not modify	0
vp_poly4_c1_ph13	[22:16]	Signed 7-bit integer (-64~63)	0
Reserved	[15]	Reserved, read as zero, do not modify	0
vp_poly4_c1_ph14	[14:8]	Signed 7-bit integer (-64~63)	0
Reserved	[7]	Reserved, read as zero, do not modify	0
vp_poly4_c1_ph15	[6:0]	Signed 7-bit integer (-64~63)	0

4.3.63 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_Y2Y_COEF, R/W, Address = 0xF010_ 01D4)

- **BT.601 to BT.709 Color Space Conversion Matrix**
  - Y709 = 1.0 * Y601 – 0.118188 * Cb601 – 0.212685 * Cr601
  - Cb709 = 0.0 * Y601 + 1.018640 * Cb601 – 0.114618 * Cr601
  - Cr709 = 0.0 * Y601 + 0.075049 * Cb601 + 1.025327 * Cr601
- BT.709 to BT.601 Color Space Conversion Matrix
  - Y601 = 1.0 * Y709 + 0.101579 * Cb709 + 0.196076 * Cr709
  - Cb601 = 0.0 * Y709 + 0.989854 * Cb709 – 0.110653 * Cr709
  - Cr601 = 0.0 * Y709 – 0.072453 * Cb709 + 0.983398 * Cr709
- Above all equations are written without interface offsets of +16 for Luminance and +128 for Chrominance.
- CSC module calculates above all equations without +128 offset for Chrominance, and generates final CSC results with +128 offset.
- In case of two Luminance equations, +16 offset is selectable by control register (PP_CSC_EN [1]). If Y offset (+16) exists in matrix input data, the coefficient of above equations should be redefined.

PP_CSC_Y2Y_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_Y2Y_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for Y to Y [11] : Sign bit [10] : Integer bit [9:0] : Fraction bit 0x7FF : 1.999 … 0x400 : 1.0 … 0x0 : 0 0xFFF : - 0.0001 … 0xC00 : - 1.0 … 0x800 : - 2.0	0

4.3.64 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_CB2Y_COEF, R/W, Address = 0xF010_ 01D8)

PP_CSC_CB2Y_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_CB2Y_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for CB to Y	0

4.3.65 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_CR2Y_COEF, R/W, Address = 0xF010_ 01DC)

PP_CSC_CR2Y_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_CR2Y_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for CR to Y	0

4.3.66 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_Y2CB_COEF, R/W, Address = 0xF010_ 01E0)

PP_CSC_Y2CB_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_Y2CB_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for Y to CB	0

4.3.67 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_CB2CB_COEF, R/W, Address = 0xF010_ 01E4)

PP_CSC_CB2CB_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_CB2CB_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for CB to CB	0

4.3.68 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_CB2CR_COEF, R/W, Address = 0xF010_ 01E8)

PP_CSC_CB2CR_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_CB2CR_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for CB to CR	0

4.3.69 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_Y2CR_COEF, R/W, Address = 0xF010_ 01EC)

PP_CSC_Y2CR_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_Y2CR_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for Y to CR	0

4.3.70 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_CR2CB_COEF, R/W, Address = 0xF010_ 01F0)

PP_CSC_CR2CB_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_CR2CB_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for CR to CB	0

4.3.71 Video Processor Post-processing Color Space Conversion Coefficient Register (PP_CSC_CR2CR_COEF, R/W, Address = 0xF010_ 01F4)

PP_CSC_CR2CR_COEF	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
PP_CSC_CR2CR_COEF	[11:0]	BT.601 to BT.709 (or BT.709 to BT.601) CSC coefficient for CR to CR	0

4.3.72 Video Processor Post-processing Image Bypass Mode Control Register (PP_BYPASS, R/W, Address = 0xF010_0200)

PP_BYPASS	Bit	Description	Reset Value
Reserved	[31:1]	Reserved, read as zero, do not modify	0
PP_BYPASS	[0]	Disables the post image processor (Post image processor executes color saturation control, sharpness enhancement, contrast and brightness control.) 0 = Enable 1 = Disable (default)	1

4.3.73 Video Processor Color Saturation Control Register (PP_SATURATION, R/W, Address = 0xF010_020C)

PP_SATURATION	Bit	Description	Reset Value
Reserved	[31:8]	Reserved, read as zero, do not modify	0
PP_SATURATION	[7:0]	Color saturation factor (Unsigned 1.7 format) 0x00 = 0.0 … 0x80 = 1.0 … 0xFF = 1.992188, (128*2 – 1)/128	80

4.3.74 Video Processor Picture Sharpness Control Register (PP_SHARPNESS, R/W, Address = 0xF010_0210)

PP_SHARPNESS	Bit	Description	Reset Value
Reserved	[31:2]	Reserved, read as zero, do not modify	0
PP_TH_HNOISE	[15:8]	Threshold value setting to decide minimum vertical edge value	0x5
Reserved	[7:2]	Reserved, read as zero, do not modify	0
PP_SHARPNESS	[1:0]	Control for the edge enhancement - 0: no effect - 1: Minimum edge enhancement - 2: Moderate edge enhancement - 3: Maximum edge enhancement	0

4.3.75 Video Processor Brightness & Contrast Control Register (PP_LINE_EQ0 ~ PP_LINE_EQ7)

- PP_LINE_EQ0, R/W, Address = 0xF010_0218
- PP_LINE_EQ1, R/W, Address = 0xF010_021C
- PP_LINE_EQ2, R/W, Address = 0xF010_0220
- PP_LINE_EQ3, R/W, Address = 0xF010_0224
- PP_LINE_EQ4, R/W, Address = 0xF010_0228
- PP_LINE_EQ5, R/W, Address = 0xF010_022C
- PP_LINE_EQ6, R/W, Address = 0xF010_0230
- PP_LINE_EQ7, R/W, Address = 0xF010_0234

PP_LINE_EQx	Bit	Description	Reset Value
Reserved	[31:24]	Reserved, read as zero, do not modify	0
LINE_INTC	[23:8]	Intercept, signed 9.7 format * Note : (9.7) format means that - '1' is a signed bit. - '8' is a integer. - '7' is a fraction. (Located in LSB) Example) INTC = 3 Because of 7bit fraction, it is had to do 7 time left shift operation. As a result, register value is 3 * 2^7 = 0x18000	0

LINE_SLOPE	[7:0]	Slope, unsigned 1.7 format. (because of 1bit integer, LINE_SLOPE has range from 0 to 1.9921875.) * Note : (1.7) format means that - '1' is a integer - '7' is a fraction. (Located in LSB) Example) LINE_SLOPE = 0.5 = 1*2^(-1) Because of 7bit fraction, it is had to do 7 time left shift operation. As a result, register value is 1/2 * 2^7 = 0x40	0

NOTE :
8 equation is related with Figure 9,8-7. Input luminance value between 0 ~ 255 is divide by 8 steps. Each of them is matched with each of 8 equations. So, we can make the new curve of the contrast and luminance as using 8 equation's combination. Each equation is matched like following as :

- PP_LINE_EQ0 : 0 ~ 31
- PP_LINE_EQ1 : 31 ~ 63
- PP_LINE_EQ2 : 64 ~ 95
- PP_LINE_EQ4 : 96 ~ 127
- PP_LINE_EQ5 : 128 ~ 159
- PP_LINE_EQ6 : 160 ~ 191
- PP_LINE_EQ7 : 192 ~ 223
- PP_LINE_EQ8 : 224 ~ 255

4.3.76 Video Processor Brightness Offset Control Register for Y (PP_BRIGHT_OFFSET, R/W, Address = 0xF010_ 0238)

PP_BRIGHT_OFFSET	Bit	Description	Reset Value
Reserved	[31:9]	Reserved, read as zero, do not modify	0
PP_BRIGHT_OFFSET	[8:0]	Offset for Y brightness control (Signed 1.8 format) Bright enhanced Y = Org Y + BRIGHT_OFFSET 0xFF : +255 … 0x1 : + 1 0x0 : 0 0x1FF : – 1 … 0x100 : - 256	0

NOTE: Figure 9.8- shows examples of how VP controls brightness and contrast of image sequence using PP_LINE_EQ0 ~ PP_LINE_EQ7 registers and PP_BRIGHT_OFFSET register. Input to output luminance mapping curve is approximated by 8 sub-lines described by PP_LINE_EQ0 ~ PP_LINE_EQ7. Consequently, brightness and contrast is controlled in very flexible way.

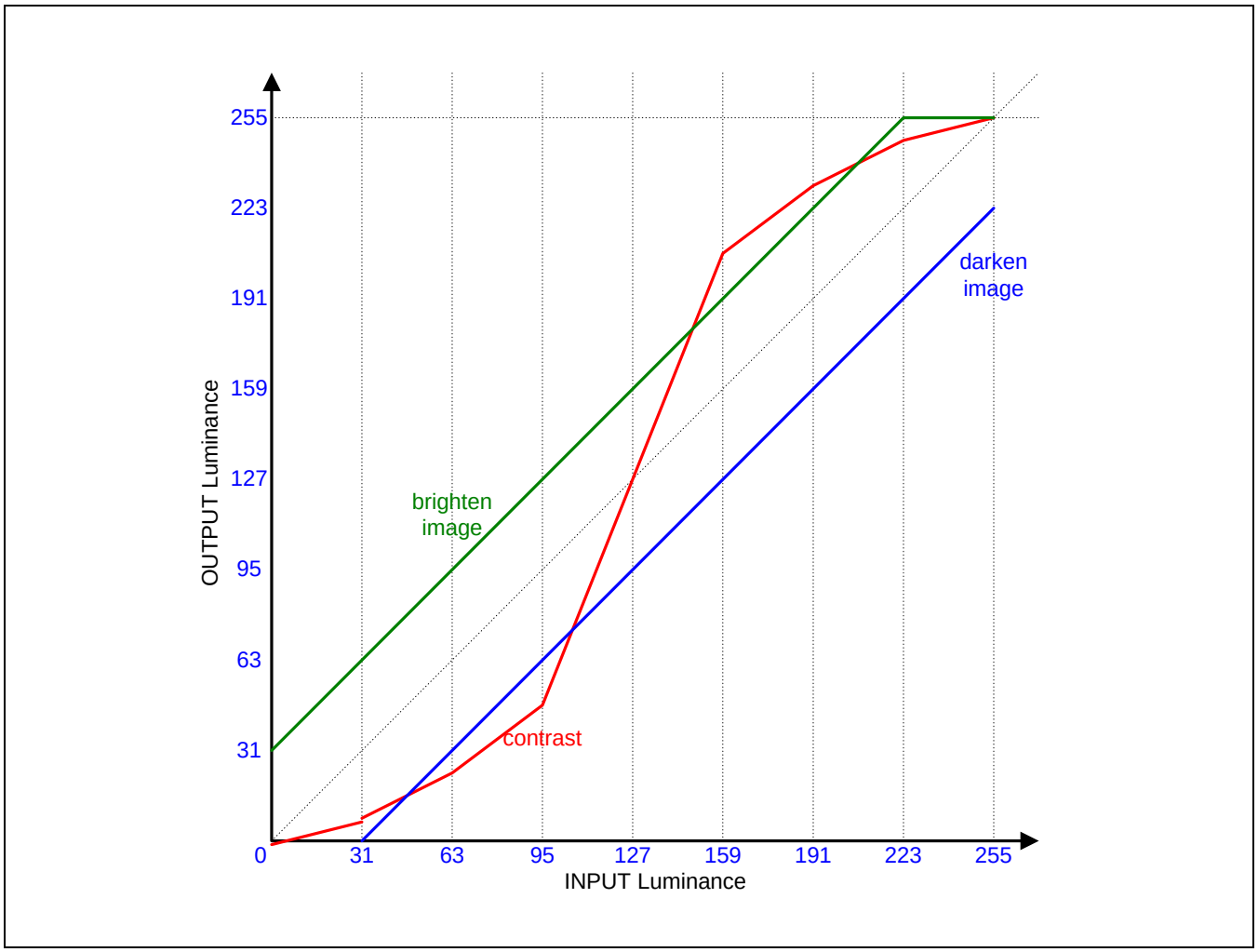


Figure 9.8-8 Image Brightness & Contrast Control

4.3.77 Video Processor Color Space Conversion Control Register (PP_CSC_EN, R/W, Address = 0xF010_023C)

PP_CSC_EN	Bit	Description	Reset Value
Reserved	[31:2]	Reserved, read as zero, do not modify	0

SUB_Y_OFFSET_EN	[1]	Y offset control for color space conversion
		If (SUB_Y_OFFSET_EN == 1)
		Y' = (Y-16)*Y2Y_coef + (Cb-128)*Cb2Y_coef + (Cr-128)* Cr2Y_coef
		Cb' = (Y-16)*Y2Cb_coef + (Cb-28)*Cb2Cb_coef + (Cr-128)*Cr2Cb_coef
		Cr' = (Y-16)*Y2Cr_coef + (Cb-28)*Cb2Cr_coef + (Cr-128)*Cr2Cr_coef
		Else
		Y' = Y*Y2Y_coef + (Cb-128)*Cb2Y_coef + (Cr-128)* Cr2Y_coef
		Cb' = Y*Y2Cb_coef + (Cb-28)*Cb2Cb_coef + (Cr-128)*Cr2Cb_coef
		Cr' = Y*Y2Cr_coef + (Cb-28)*Cb2Cr_coef + (Cr-128)*Cr2Cr_coef
			1

CSC_EN	[0]	Color space conversion enable control 0 = Disable 1 = Enable	0

4.3.78 Video Processor Version Information Register (VP_VERSION_INFO, R, Address = 0xF010_03FC)

VP_VERSION_INFO	Bit	Description	Reset Value
VERSION_INFO	[31:0]	VP version information	0x0000_0010

The Idea of Poly-phase Filtering in Video Processor

Figure 9.8- shows basic concept of poly-phase filtering in video processor in case of 4-tap vertical luminance filter. Pixels in gray color are from decoded pictures and used to interpolate the dotted pixels which are transferred to MIXER. The vertical positions of pixels to be interpolated are calculated with VP_SRC_V_POSITION and VP_V_RATIO. Once the vertical position is calculated, the nearest pixel phase (with 1/16 resolution) and which pixels are used for interpolation are decided.

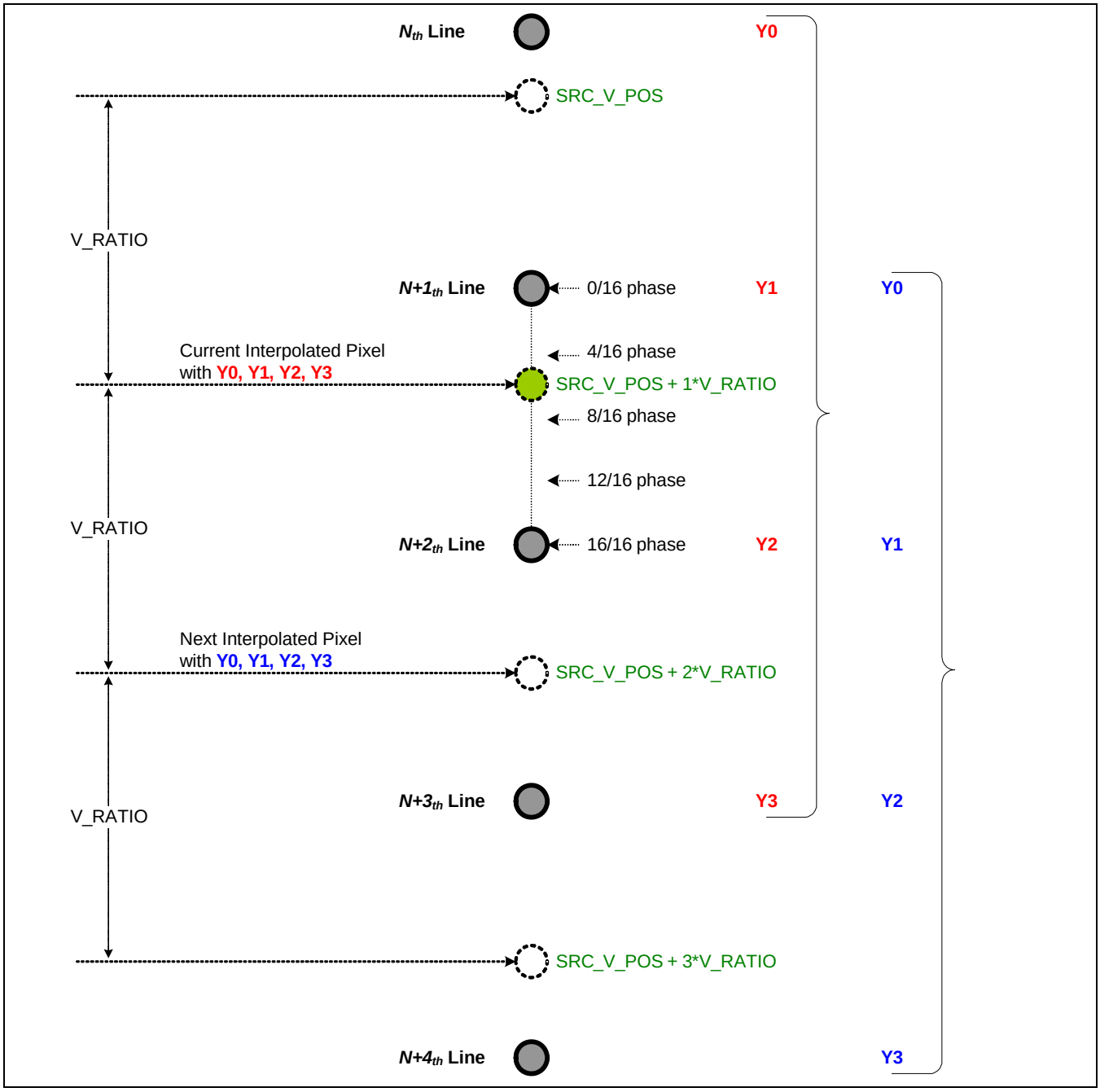


Figure 9.8-9 4-Tap Vertical Poly-phase Filter

If the calculated vertical position is 10.45, for example, pixels of 9th, 10th, 11th, and 12th lines are used for poly-phase filtering and the pixel phase is 7/16, which means the filter coefficients are vp_poly4_y0_ph7, vp_poly4_y1_ph7, vp_poly4_y2_ph7, and vp_poly4_y3_ph7.

8-tap luminance horizontal poly-phase filter and 4-tap chrominance horizontal poly-phase filter use the exact same scheme.

At the boundaries of pictures (top, bottom, left, and right), some pixels in filter window are not available. In this case, value of the nearest pixel is repeated as shown in Figure 9.8-.

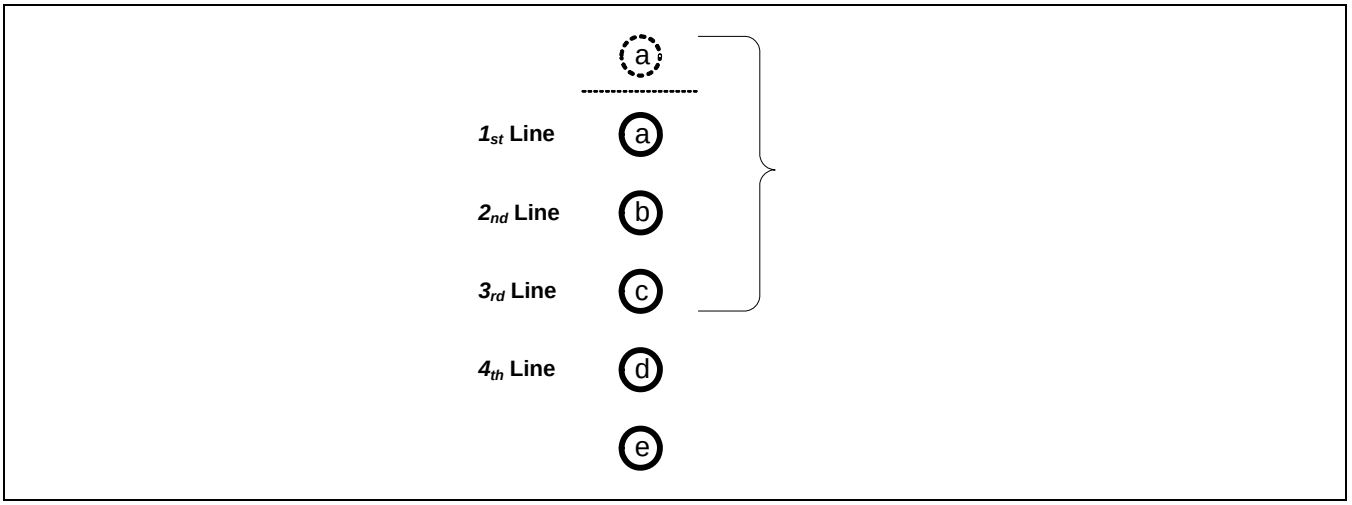


Figure 9.8-10 Pixel Repetition at Picture Boundary

NOTES

9.9 MIXER

1 OVERVIEW

Mixer overlaps or blends the input data such as graphic, video, background and sends the result data to the TVOUT module that generates all the video control signals.

Graphic data is transferred to the mixer from External DRAM memory through AXI interface. However, the video data is transferred to the mixer through direct connection with pre-defined protocol. The data generated from mixer is directly transferred to TVENC/HDMI module for the real time transfer.

1.1 FEATURE

Supports AXI Master & AHB Slave Interface

a) AXI Master interface for graphic layer data fetch
b) AHB Slave interface for control register setup
c) Supports Little/ Big Endian

Input

a) Multiple Layers
- Background layer
- Graphic0 layer
- Graphic1 layer
- Video layer

b) Input Control features
- Blending between each layers
- Selectable graphic layer frame buffer
- Enable/disable each layer
- Source cropping for graphic layer

Output

a) Overlapped and blended input layers
b) YCbCr 4:4:4
c) Supports interlaced/ progressive scan
d) Supports 480i/p, 576i/p , and 720p display sizing

Graphic0,1 Layer

a) Source: External DRAM frame buffer memory

b) Color Format : differently configurable between each graphic layer
- 16bpp Direct RGB[565]
- 16bpp Direct ARGB[1555]
- 16bpp Direct ARGB[4444]
- 32bpp Direct ARGB[8888]

c) Maximum graphic layer size
- 480i/p : 720x480 pixel
- 576i/p : 720x576 pixel
- 720p: 1280x720 pixel

d) Blending
- Maximum 256 level pixel and layer blending
- Separately configurable layer blending factor between each layer

Video Layer

a) Source: Video processor module

b) Color Format: 24bpp Direct YCbCr [888]

c) Maximum Resolution
- 480i/p : 720x480 pixel
- 576i/p : 720x576 pixel

Background layer

a) Source: Configuration register

b) Lowest layer

c) Gradient color : 24bpp YCbCr[888]

Layer ordering

a) Background → (Video ↔ Graphic0 ↔ Graphic1)

b) Video layer and 2 graphic layers is fully ordered and blended

Blending

a) Pixel blending and Layer blending

b) Alpha blending

c) Pre-multiplied blending

1.2 BLOCK DIAGRAM

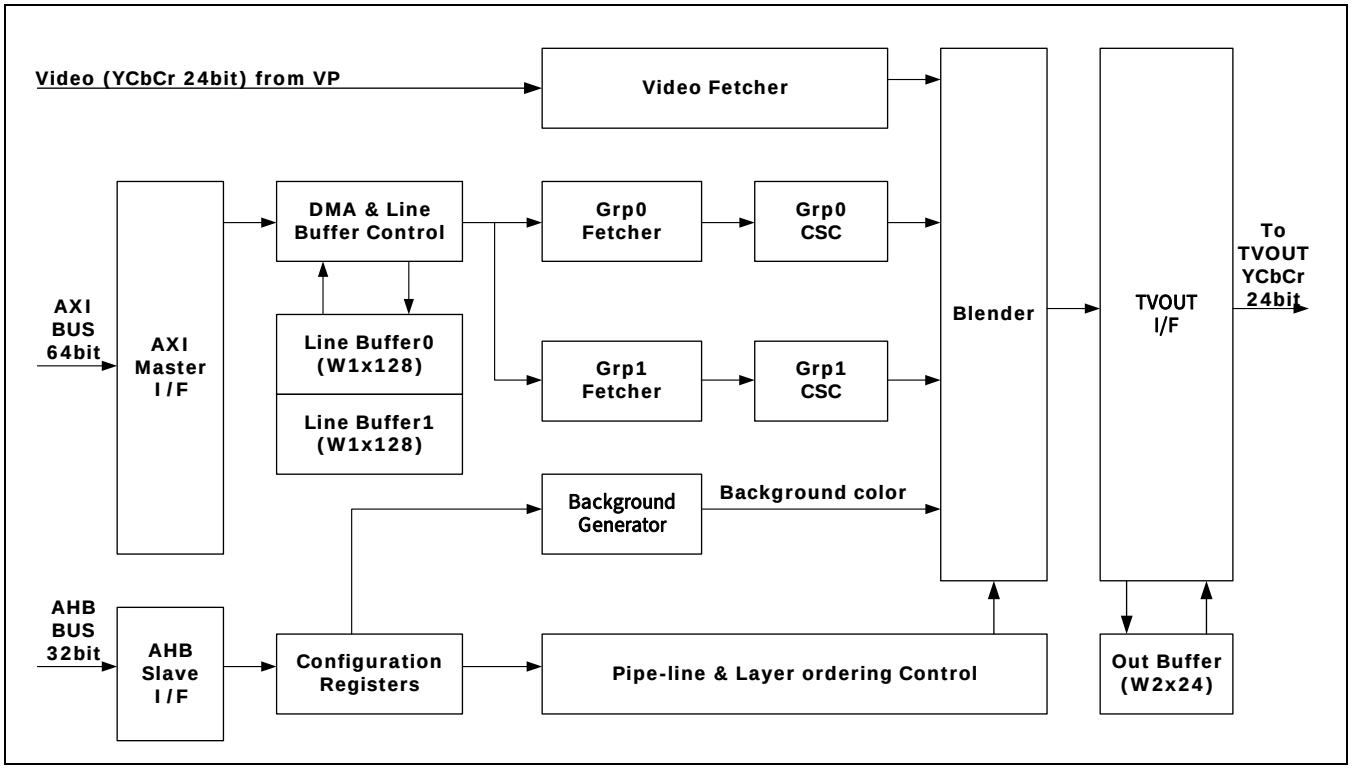
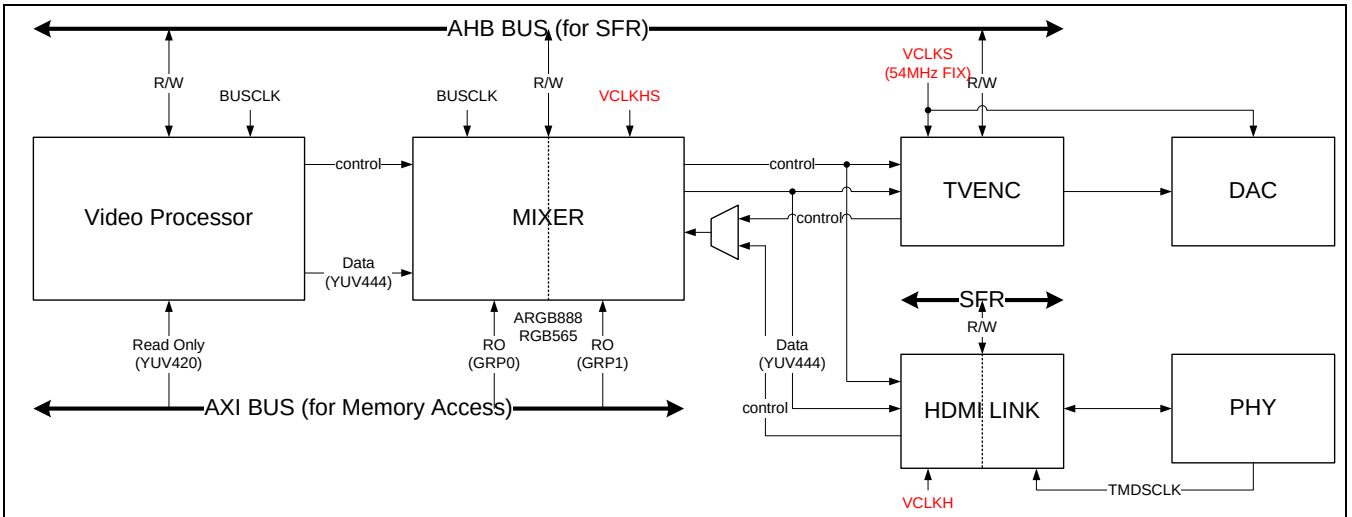


Figure 9.9-1 Mixer Block Diagram

1.3 VIDEO CLOCK RELATION



VCLKHS(or VCLKH) usage frequency

Vertical Freq.	Format	Pixel Freq.
59.94 Hz	480I	27MHz
	480P	27MHz
	720P	74.175MHz
	1080I	74.175MHz
60.00 Hz	480I	27.027MHz
	480P	27.027MHz
	720P	74.250MHz
	1080I	74.250MHz
50.00 Hz	576I	27MHz
	576P	27MHz
	720P	74.250MHz
	1080I	74.250MHz

Figure 9.9-2 TV Sub-System Block Diagram and Usage Frequency

There are two paths that are TVENC-out and HDMI-out. It is selected exclusively at the Clock Controller (refer to 9.6.7 MIXER_OUT_SEL register). When TV-out is selected, Mixer I/F clock (VCLKHS) and VCLKS (TVENC clock) is fixed by 54MHz. Thus, you must configure MIXER_SEL register (9.2.3 CLK_SRC2, Clock Controller) by VCLK_54(0x01). Otherwise, in HDMI-out selection, MIXER_OUT_SEL register is configured properly. Then, you make the same clock configuration between MIXER I/F clock (VCLKHS) and VCLKH(HDMI pixel clock). The clock which is generated by HPLL is selected properly (refer to VCLKHS(or VCLKH) usage frequency table). you can configure clock source through CLK_SRC2 register.

2 REGISTERS DESCRIPTION

Register	Address	R/W	Description	Reset Value
Mixer Global Setting				
MIXER_STATUS	0xF020_0000	R/W	Status of MIXER Operation	0x0000_0006
MIXER_CFG	0xF020_0004	R/W	MIXER Mode Setting	0x0000_0000
Mixer Interrupt				
MIXER_INT_EN	0xF020_0008	R/W	Interrupt Enable	0x0000_0000
MIXER_INT_STATUS	0xF020_000C	R/W	Interrupt Status	0x0000_0000
Video & Blender Configuration				
MIXER_LAYER_CFG	MIXER_0xF020+ 0x0010	R/W	Video & Graphic Layer Priority and On/ Off	0x0000_0000
MIXER_VIDEO_CFG	0xF020_0014	R/W	Video Layer Configuration	0x0000_0000
Graphic0 Layer Configuration				
MIXER_GRAPHIC0_CFG	0xF020_0020	R/W	Graphic Layer0 Configuration	0x0000_0000
MIXER_GRAPHIC0_BASE	0xF020_0024	R/W	Base Address for Graphic Layer0	0x0000_0000
MIXER_GRAPHIC0_SPAN	0xF020_0028	R/W	Span for Graphic Layer0	0x0000_0000
MIXER_GRAPHIC0_SXY	0xF020_002C	R/W	Source X/Y Positions for Graphic Layer0	0x0000_0000
MIXER_GRAPHIC0_WH	0xF020_0030	R/W	Width/ Height for Graphic Layer0	0x0000_0000
MIXER_GRAPHIC0_DXY	0xF020_0034	R/W	Destination X/Y Positions for Graphic Layer0	0x0000_0000
MIXER_GRAPHIC0_BLANK	0xF020_0038	R/W	Blank Pixel Value for Graphic Layer0	0x0000_0000
Graphic1 Layer Configuration				
MIXER_GRAPHIC1_CFG	0xF020_0040	R/W	Graphic Layer1 Configuration	0x0000_0000
MIXER_GRAPHIC1_BASE	0xF020_0044	R/W	Base Address for Graphic Layer1	0x0000_0000
MIXER_GRAPHIC1_SPAN	0xF020_0048	R/W	Span for Graphic Layer1	0x0000_0000
MIXER_GRAPHIC1_SXY	0xF020_004C	R/W	Source X/Y Positions for Graphic Layer1	0x0000_0000
MIXER_GRAPHIC1_WH	0xF02_0050	R/W	Width/ Height for Graphic Layer1	0x0000_0000
MIXER_GRAPHIC1_DXY	0xF020_0054	R/W	Destination X/Y Positions for Graphic Layer1	0x0000_0000
MIXER_GRAPHIC1_BLANK	0xF020_0058	R/W	Blank Pixel Value for Graphic Layer1	0x0000_0000
Background Layer Configuration				
MIXER_BG_COLOR0	0xF020_0064	R/W	Background Color of First Point	0x0000_0000
MIXER_BG_COLOR1	0xF020_0068	R/W	Background Color of Second Point	0x0000_0000
MIXER_BG_COLOR2	0xF020_006C	R/W	Background Color of Last Point	0x0000_0000
Color Space Conversion Coefficient				

Register	Address	R/W	Description	Reset Value
MIXER_CM_COEFF_Y	0xF020_0080	R/W	Scaled Color Space Conversion (RGB to Y) Coefficient for Graphic Layer	0x0844_0832
MIXER_CM_COEFF_CB	0xF020_0084	R/W	Scaled Color Space Conversion (RGB to CB) Coefficient for Graphic Layer	0x3b4d_ace1
MIXER_CM_COEFF_CR	0xF020_0088	R/W	Scaled Color Space Conversion (RGB to Cr) Coefficient for Graphic Layer	0x0e1d_13dc
Mixer Global Setting Shadowing Register				
MIXER_STATUS_S	0xF020_2000	R	Status of MIXER Operation (Shadow)	0x0000_0006
MIXER_CFG_S	0xF020_2004	R	MIXER Mode Setting (Shadow)	0x0000_0000
Video & Blender Configuration Shadowing Register				
MIXER_LAYER_CFG_S	0xF020_2010	R	Video & Graphic Layer Priority and On/ Off (Shadow)	0x0000_0000
MIXER_VIDEO_CFG_S	0xF020_2014	R	Video Layer Configuration (Shadow)	0x0000_0000
Graphic0 Layer Configuration Shadowing Register				
MIXER_GRAPHIC0_CFG_S	0xF020_2020	R	Graphic Layer0 Configuration (Shadow)	0x0000_0000
MIXER_GRAPHIC0_BASE_S	0xF020_2024	R	Graphic0 Base Address (Shadow)	0x0000_0000
MIXER_GRAPHIC0_SPAN_S	0xF020_2028	R	Graphic0 Span (Shadow)	0x0000_0000
MIXER_GRAPHIC0_SXY_S	0xF020_202C	R	Graphic0 Source X/Y Coordinates (Shadow)	0x0000_0000
MIXER_GRAPHIC0_WH_S	0xF020_2030	R	Graphic0 Width/ Height (Shadow)	0x0000_0000
MIXER_GRAPHIC0_DXY_S	0xF020_2034	R	Graphic0 Destination X/Y Coordinates (Shadow)	0x0000_0000
MIXER_GRAPHIC0_BLANK_PIXEL_S	0xF020_2038	R	Graphic0 Blank Pixel (Shadow)	0x0000_0000
Graphic1 Layer Configuration Shadowing Register				
MIXER_GRAPHIC1_CFG_S	0xF020_2040	R	Graphic Layer1 Configuration (Shadow)	0x0000_0000
MIXER_GRAPHIC1_BASE_S	0xF020_2044	R	Graphic1 Base Address (Shadow)	0x0000_0000
MIXER_GRAPHIC1_SPAN_S	0xF020_2048	R	Graphic1 Span (Shadow)	0x0000_0000
MIXER_GRAPHIC1_SXY_S	0xF020_204C	R	Graphic1 Source X/Y Coordinates (Shadow)	0x0000_0000
MIXER_GRAPHIC1_WH_S	0xF020_2050	R	Graphic1 Width/ Height (Shadow)	0x0000_0000
MIXER_GRAPHIC1_DXY_S	0xF020_2054	R	Graphic1 Destination X/Y Coordinates (Shadow)	0x0000_0000
MIXER_GRAPHIC1_BLANK_PIXEL_S	0xF020_2058	R	Graphic1 Blank Pixel (Shadow)	0x0000_0000
Background Layer Configuration Shadowing Register				

Register	Address	R/W	Description	Reset Value
MIXER_BG_COLOR0_S	0xF02_2064	R	Background First Color (Shadow)	0x0000_0000
MIXER_BG_COLOR1_S	0xF020_2068	R	Background Second Color (Shadow)	0x0000_0000
MIXER_BG_COLOR2_S	0xF020_206C	R	Background Last Color (Shadow)	0x0000_0000
Version Register				
MIXER_VER	0xF020_0100	R	Mixer Version	0x07D6_0A11

2.1 SHADOW REGISTERS (READ ONLY)

If SYNC_ENABLE signal is set to 1, the written values to internal registers are not directly applied to the mixer operation. They are temporarily stored in the internal register and wait for next v_sync signal. After the v_sync signal occurs, the stored internal register values are updated to the shadow registers.

In interlaced display mode, the shadow registers are updated only at top-field. In progressive display mode, the shadow registers are updated at every v_sync.

2.2 MIXER_STATUS REGISTER (MIXER_STATUS, R/W, ADDRESS = 0XF020_ 0000)

MIXER_STATUS	Bit	Description	Reset Value
Reserved	[31:8]	Reserved, read as zero, do not modify	0
16_BURST_MODE	[7]	16 burst mode(64Bit Bus) enabled in DMA 1 = 16Beat Burst Mode 0 = 8Beat Burst Mode	0
Reserved	[6:4]	Reserved.	0
BIG_ENDIAN	[3]	0 = Little Endian Source Format 1 = Big Endian Source Format	0
SYNC_ENABLE	[2]	0 = Values set by user will not be applied to the mixer operation although v_sync is detected. 1 = Values set by user can be applied to the mixer operation after v_sync detected.	1
Reserved	[1]	Reserved, read as zero, do not modify	1
REG_RUN	[0]	The mixer operation control. This register is also updated after V_SYNC. 0 = The mixer stops. 1 = The mixer starts processing. NOTE : The SFRs of Video Processor and Image Mixer is updated by Vertical Sync of TVENC's Timing Generator. So, those are configured before this bit is enabled. The sequence of enabling TVSS is following as: "VP -> MIXER -> TVENC(HDMI)". Also, because of the same reason, the disabling sequence is following as : "VP -> MIXER -> TVNEC(HDMI)".	0

2.3 MIXER_CFG REGISTER (MIXER_CFG, R/W, ADDRESS = 0XF020_ 0004)

MIXER_CFG	Bit	Description	Reset Value
Reserved	[31:7]	Reserved, read as zero, do not modify	0
REG_HD_MODE	[6]	720p selection 0 = 720p 1 = Reserved	0
REG_GRAPHIC1_EN	[5]	Graphic1 layer display control bit. 0 = Disable 1 = Enable	0
REG_GRAPHIC0_EN	[4]	Graphic0 layer display control bit. 0 = Disable 1 = Enable	0
REG_VIDEO_EN	[3]	Video layer display control bit. 0 = Disable 1 = Enable	0
REG_SCAN_MODE	[2]	Display scanning mode of TV. 0 = Interlaced mode 1 = Progressive mode	0
REG_NTSC_PAL	[1]	Display standard of TV. If you set this bit '0' and set REG_SCAN_MODE '1', output has to be call 480p standard. 0 = NTSC (720x480) 1 = PAL (720x576)	0
REG_HD_SD	[0]	HD or SD selection 0 = SD 1 = HD If REG_HD_SD is 1, REG_HD_MODE = 0 for 720p	0

NOTE: All changes to this register are valid on a vertical sync signal of next frame.

2.4 MIXER_INT_EN REGISTER (MIXER_INT_EN, R/W, ADDRESS = 0XF020_ 0008)

MIXER_INTR	Bit	Description	Reset Value
Reserved	[31:11]	Reserved, read as zero, do not modify	0
INT_EN_VP	[10]	The VP underflow interrupt enable. 0 = Disables interrupt 1 = Enables interrupt Setting this bit to '0' disables only the interrupt request to host controller. It does not mask the change of the MIXER_INT_STATUS[10] bit status	0
INT_EN_GRP1	[9]	The graphic layer1 line buffer underflow interrupt enable. 0 = Disables interrupt 1 = Enables interrupt Setting this bit to '0' disables only the interrupt request to host controller. It does not mask the change of the MIXER_INT_STATUS[9] bit status	0
INT_EN_GRP0	[8]	The graphic layer0 line buffer underflow interrupt enable. 0 = Disables interrupt 1 = Enables interrupt Setting this bit to '0' disables only the interrupt request to host controller. It does not mask the change of the MIXER_INT_STATUS[8] bit status	0
Reserved	[7:0]	Reserved.	0

2.5 MIXER_INT_STATUS REGISTER (MIXER_INT_STATUS, R/W, ADDRESS = 0XF020_ 000C)

MIXER_INTR	Bit	Description	Reset Value
Reserved	[31:11]	Reserved, read as zero, do not modify	0
INT_STATUS_VP	[10]	The VP underflow interrupts status. 0 = Interrupt is not fired 1 = Interrupt is fired Writing '1' to this bit clears the interrupt. This interrupt is automatically asserted by line buffer controller when generated underflow in line buffer	0
INT_STATUS_GRP1	[9]	The graphic layer1 line buffer underflow interrupt status. 0 = Interrupt is not fired 1 = Interrupt is fired Writing '1' to this bit clears the interrupt. This interrupt is automatically asserted by line buffer controller when generated underflow in line buffer	0
INT_STATUS_GRP0	[8]	The graphic layer0 line buffer underflow interrupt status. 0 = Interrupt is not fired 1 = Interrupt is fired Writing '1' to this bit clears the interrupt. This interrupt is automatically asserted by line buffer controller when generated underflow in line buffer	0
Reserved	[7:0]	Reserved, read as zero, do not modify	0

2.6 MIXER_LAYER_CFG REGISTER (MIXER_LAYER_CFG, R/W, ADDRESS = 0XF020_ 0010)

The priority value for video and each graphic layer can be set. The priority field is used to determine the priority of a graphic layer. The graphic layer of higher value has the higher priority. This field is also used as on/off switch. If one field is set to zero, the corresponding graphic layer is not displayed. If some layers have the same value, the priority is like this: graphic layer 1 > graphic layer 0 > video. The priority is only determined by difference of priority's value. For example, case1 and case 2 have the same effect.

Case1 : GRP1 priority is 2 , GRP0 priority 3 , Video Priority 1

Case2 : GRP1 priority is 14 , GRP0 priority 15 , Video Priority 13

MIXER_LAYER_CFG	Bit	Description	Reset Value
Reserved	[31:12]	Reserved, read as zero, do not modify	0
Graphic layer 1 priority	[11:8]	15 ~ 1 = the priority value 0 = Hides the graphic layer 1	0
Graphic layer 0 priority	[7:4]	15 ~ 1 = the priority value 0 = Hides the graphic layer 0	0
Video layer priority	[3:0]	15 ~ 1 = the priority value 0 = Hides the video layer 0	0

NOTE: All the changes of this register are valid on a vertical sync signal of next frame when SYNC_ENABLE flag is set to one. And the "Hide" means that layer data is ready but is not displayed.

2.7 MIXER_VIDEO_CFG REGISTER (MIXER_VIDEO_CFG, R/W, ADDRESS = 0XF020_ 0014)

MIXER_VIDEO_CFG	Bit	Description	Reset Value
Reserved	[31:17]	Reserved, read as zero, do not modify	0
REG_BLEND_EN	[16]	If set to 1, it enables the blending of all the video layer onto the lower layer using the blending factor, REG_ALPHA_VID.	0
Reserved	[15:8]	Reserved, read as zero, do not modify	0
REG_ALPHA_VID	[7:0]	Video layer blending factor. This factor is used over all the pixels in the video layer to blend with lower layer. α * video_layer_pixel_value + (1 – α) * lower_layer_pixel_value If REG_ALPHA_VID is 0, α is 0. If REG_ALPHA_VID is not 0, α = (REG_ALPHA_VID + 1)/256.	0

NOTE: All changes to this register are valid on a vertical sync signal of next frame.

2.8 MIXER_GRAPHIC0_CFG REGISTER (MIXER_GRAPHIC0_CFG, R/W, ADDRESS = 0XF020_ 0020)

MIXER_GRAPHIC0_CFG	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0
BLANK_CHANGE0	[21]	0 = No modification is applied to the input graphic layer0's pixel data 1 = If the input graphic layer0's pixel is blank, it will be changed to a new pixel that is not blank pixel.	0
PRE_MUL_MODE0	[20]	Pre Multiplied_blending mode in graphic layer0. 1 = Pre-Multiplied mode 0 = Normal mode In this mode, graphic pixel data must be pre-multiplied with graphic pixel alpha. In pre-multiplied mode, REG_PIXEL0_BLEND_EN must be enabled. Graphic and lower layer blending factor. This factor is used all over the pixels in the graphic and lower layer to blend with. A blending factor(α) is as follows depending on the a layer blending factor and a pixel blending factor values: See Table 9.9-1	0
Reserved	[19:18]	Reserved, read as zero	
REG_WIN0_BLEND_EN	[17]	Blending by a blending factor set by REG_ALPHA_WIN0 register on all over the graphic layer0.	0
REG_PIXEL0_BLEND_EN	[16]	Blending by a blending factor that each pixel have is enabled in graphic layer0.	0
Reserved	[15:12]	Reserved, read as zero	0

MIXER_GRAPHIC0_CFG	Bit	Description	Reset Value
EG_COLOR_FORMAT0	[11:8]	Graphic layer0 color format. 0 = Reserved 1 = Reserved 2 = Reserved 3 = Reserved 4 = RGB 565 5 = ARGB 1555 6 = ARGB 4444 7 = ARGB 8888 8 = Reserved 9 = Reserved A = Reserved B = Reserved C = Reserved D = Reserved E = Reserved F = Reserved	0
REG_ALPHA_WIN0	[7:0]	Graphic layer0 and lower layer blending factor. This factor is used all over the pixels in the graphic and lower layer to blend with. A blending factor(α) is as follows depending on the a layer blending factor and a pixel blending factor values: α * graphic_layer_pixel_value + (1 – α) * lower_layer_pixel_value See Table 9.9-1 If REG_ALPHA_WIN0 is 0, blending_factor_layer is 0. If REG_ALPHA_WIN0 is not 0, blending_factor_layer = (REG_ALPHA_WIN + 1) / 256. If A(blending factor of each pixel) is 0, blending_factor_each_pixel is 0 If A(blending factor of each pixel) is not 0, blending_factor_each_pixel = (A + 1) / 256. The pixel blending factors comes from the pixel in direct modes except the direct 16bpp 565 format.	0

NOTE: All the changes of this register are valid on a vertical sync signal of next frame.

2.9 MIXER_GRAPHIC1_CFG REGISTER (MIXER_GRAPHIC1_CFG, R/W, ADDRESS = 0XF020_ 0040)

MIXER_GRAPHIC1_CFG	Bit	Description	Reset Value
Reserved	[31:22]	Reserved	0
BLANK_CHANGE1	[21]	0 = No modification is applied to the input graphic layer1's pixel data 1 = If the input graphic layer1's pixel is blank, it will be changed to a new pixel that is not blank pixel.	0
PRE_MUL_MODE1	[20]	Pre Multiplied_blending mode in graphic layer1. 1 : Pre-Multiplied mode 0 : Normal mode In this mode, graphic pixel data must be pre-multiplied with graphic pixel alpha. In pre-multiplied mode, REG_PIXEL1_BLEND_EN must be enabled. Graphic and lower layer blending factor. This factor is used all over the pixels in the graphic and lower layer to blend with. A blending factor(α) is as follows depending on the a layer blending factor and a pixel blending factor values: See Table 9.9-1	0
Reserved	[19:18]	Reserved, read as zero	0
REG_WIN1_BLEND_EN	[17]	Blending by a blending factor set by REG_ALPHA_WIN1 register on all over the graphic layer1.	0
REG_PIXEL1_BLEND_EN	[16]	Blending by a blending factor that each pixel have is enabled in graphic layer1.	0
Reserved	[15:12]	Reserved, read as zero	0

MIXER_GRAPHIC1_CFG	Bit	Description	Reset Value
REG_COLOR_FORMAT1	[11:8]	Graphic layer1 color format. 0 = Reserved 1 = Reserved 2 = Reserved 3 = Reserved 4 = RGB 565 5 = ARGB 1555 6 = ARGB 4444 7 = ARGB 8888 8 = Reserved 9 = Reserved A = Reserved B = Reserved C = Reserved D = Reserved E = Reserved F = Reserved	0
REG_ALPHA_WIN1	[7:0]	Graphic layer1 and lower layer blending factor. This factor is used all over the pixels in the graphic and lower layer to blend with. A blending factor(α) is as follows depending on the a layer blending factor and a pixel blending factor values: α * graphic_layer_pixel_value + (1 – α) * lower_layer_pixel_value See Table 9.9-1 If REG_ALPHA_WIN1 is 0, blending_factor_layer is 0. If REG_ALPHA_WIN1 is not 0, blending_factor_layer = (REG_ALPHA_WIN + 1) / 256. If A(blending factor of each pixel) is 0, blending_factor_each_pixel is 0 If A(blending factor of each pixel) is not 0, blending_factor_each_pixel = (A + 1) / 256. The pixel blending factors comes from the pixel in direct modes except the direct 16bpp 565 format.	0

NOTE: All the changes of this register are valid on a vertical sync signal of next frame.

Table 9.9-1 Graphic Blending-factor Alpha in Case of Normal Mode

MIXER_GRAPHICx_CFG. REG_WINx_BLEND_EN	MIXER_GRAPHICx_CFGx. REG_PIXELx_BLEND_EN	ALPHA VALUE (Blending factor of each pixel)
0	0	1
0	1	blending_factor_each_pixel(A)
1	0	blending_factor_layer (MIXER_GRAPHICn_CFG[7:0])
1	1	blending_factor_layer * blending_factor_each_pixel

Table 9.9-2 Graphic Blending Method

Pixel Blend	Window Blend	Normal mode	Pre multiplied mode
0	0	-	-
0	1	Alpha_gw * graphic_pixel + (1 – alpha_gw) * lower layer	Alpha_gw * graphic_pixel + (1 – alpha_gw) * lower layer
1	0	Alpha_gp * graphic_pixel + (1 – alpha_gp) * lower layer	graphic_pixel + (1 – alpha_gp) * lower layer
1	1	(Alpha_gp * alpha_gw) * graphic_pixel + (1 – alpha_gp * alpha_gw) * lower layer	alpha_gw * graphic_pixel + (1 – alpha_gp * alpha_gw) * lower layer

In pre-multiplied mode, the input graphic data is multiplied by the pixel blending factor (alpha_gp) and truncated to the size of source format bits. For example, although the result of the multiplication of 8 bit data by 8 bit pixel blending factor is 16 bits which is the first term of the blending equation in the

Table 9.9-2 Normal mode, the supplied data is truncated to 8 bits that results the loss of the lower significant 8bits during calculation. In direct 32bpp modes, this loss data is somewhat indistinguishable in visually (+/- 1difference). But in direct 16bpp modes, the loss data can make visually different result from the original. For example, in 4633 direct mode, the pre-multiplied Cb data is truncated to 3 bits that result in the difference value from +15 to -15. This difference range can result a visual difference. We can't reduce the error that is resulted from the 3 bits input source. So we recommend using the 32bpp mode to use the pre-multiplied mode.

2.10 MIXER_GRAPHIC0_BASE, R/W, ADDRESS = 0XF020_ 0024

2.11 MIXER_GRAPHIC1_BASE, R/W, ADDRESS = 0XF020_ 0044

MIXER_GRAPHICn_BASE	Bit	Description	Reset Value
REG_GRAPHICn_BASE	[31:0]	Base address of frame buffer for graphic layer. This address should be word aligned, so least 2 significant bits [1:0] will be set 2'b00 automatically unless a user set this bit or not	0

2.12 MIXER_GRAPHIC0_SPAN, R/W, ADDRESS = 0XF020_ 0028

2.13 MIXER_GRAPHIC1_SPAN, R/W, ADDRESS = 0XF020_ 0048

MIXER_GRAPHICn_SPAN	Bit	Description	Reset Value
Reserved	[31:15]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_SPAN	[14:0]	Horizontal pixel interval between line and line in graphic layerx's source image Note : SPAN is the number of the original image 's horizontal pixel count. For example, 640x480's span is '640'.It doesn't care BPP(bit per pixel).	0

2.14 MIXER_GRAPHIC0_WH, R/W, ADDRESS = 0XF020_ 0030

2.15 MIXER_GRAPHIC1_WH, R/W, ADDRESS = 0XF020_ 0050

MIXER_GRAPHICn_ WH	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_W	[26:16]	Width of graphic layer (pixel unit)	0
Reserved	[15:10]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_H	[10:0]	Height of graphic layer (pixel unit).	0

NOTE: All the changes of this register are valid on a vertical sync signal of next frame.
NOTE: When specifying the X coordinates and the width of a graphic layer, it should be located inside the display region, for example, 720x480 region in NTSC display mode and 720x576 region in PAL display. The coordinates (x, y) and the size (width, height) should be based on the progressive mode although it is interlaced display mode.
NOTE: Graphic width and height should be larger than 0 if the corresponding REG_GRAPHICx_EN field(MIXER_CFG[5], MIXER_CFG[4]) is set to 1

2.16 MIXER_GRAPHIC0_SXY, R/W, ADDRESS = 0XF020_ 002C

2.17 MIXER_GRAPHIC1_SXY, R/W, ADDRESS = 0XF020_ 004C

MIXER_GRAPHICn_XY	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_SX	[26:16]	X coordinate of upper left corner of graphic layer in source frame (pixel unit). Allowed range = 0 ~ 719 at SD mode = 0 ~ 1279 at HD mode(720p)	0
Reserved	[15:10]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_SY	[10:0]	Y coordinate of upper left corner of graphic layer in source frame (pixel unit). Allowed range = 0 ~ 479 at NTSC mode = 0 ~ 575 at PAL mode = 0 ~ 719 at HD mode(720p)	0

2.18 MIXER_GRAPHIC0_DXY, R/W, ADDRESS = 0XF020_ 0034

2.19 MIXER_GRAPHIC1_DXY, R/W, ADDRESS = 0XF020_ 0054

MIXER_GRAPHICn_DXY	Bit	Description	Reset Value
Reserved	[31:27]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_DX	[26:16]	X coordinate of upper left corner of graphic layer in destination frame (pixel unit). Allowed range = 0 ~ 719 at SD mode = 0 ~ 1279 at HD mode(720p)	0
Reserved	[15:10]	Reserved, read as zero, do not modify	0
REG_GRAPHICn_DY	[10:0]	Y coordinate of upper left corner of graphic layer in destination frame (pixel unit). Allowed range = 0 ~ 479 at NTSC mode = 0 ~ 575 at PAL mode = 0 ~ 719 at HD mode(720p)	0

2.20 MIXER_GRAPHIC0_BLANK, R/W, ADDRESS = 0XF020_ 0038

2.21 MIXER_GRAPHIC1_BLANK, R/W, ADDRESS = 0XF020_ 0058

MIXER_GRAPHICn_BLANK	Bit	Description	Reset Value
REG_GRAPHICn_BLANK	[31:0]	Blank pixel value for graphic layerN	0

2.22 MIXER_BG_COLOR0, R/W, ADDRESS = 0XF020_ 0064

2.23 MIXER_BG_COLOR1, R/W, ADDRESS = 0XF020_ 0068

2.24 MIXER_BG_COLOR2, R/W, ADDRESS = 0XF020_ 006C

MIXER_BG_COLOR0/1/2	Bit	Description	Reset Value
-	[31:24]	Reserved, read as zero, do not modify	0
Y	[23:16]	Y component of background color	0
Cb	[15:8]	Cb component of background color	0
Cr	[7:0]	Cr component of background color	0

NOTE: You can choose proper YCbCr value for BT.601 or BT.709.

2.25 MIXER_COLOR_SPACE_CONVERSION_COEF_Y REGISTER (MIXER_CM_COEFF_Y, R/W, ADDRESS = 0XF020_ 0080)

MIXER_CM_COEFF_Y	Bit	Description	Reset Value
Reserved	[31]	Reserved, read as zero, do not modify	
WIDE_SEL	[30]	0 = Narrow 1 = Wide	0
REG_COEFF_00	[29:20]	Scaled color space conversion coefficient (C_{00}). [29] : Sign-bit [28:20] : Fractional bit (Default and Recommended value : 0.257 in decimal)	0x84
REG_COEFF_10	[19:10]	Scaled color space conversion coefficient (C_{10}). [19] : Sign-bit [18:10] : Fractional bit (Default and Recommended value : 0.504 in decimal)	0x102
REG_COEFF_20	[9:0]	Scaled color space conversion coefficient (C_{20}). [9] = Sign-bit [8:0] = Fractional bit (Default and Recommended value : 0.098 in decimal)	0x32

NOTE: RGB to YCbCr Conversion Equations

1) RGB data with 16-235 range

Y601 = 0.299R + 0.587G + 0.114B

Cb = -0.172R – 0.339G + 0.511B + 128

Cr = 0.511R – 0.428G – 0.083B + 128

Y709 = 0.213R + 0.715G + 0.072B

Cb = -0.117R – 0.394G + 0.511B + 128

Cr = 0.511R – 0.464G – 0.047B + 128

2) RGB data with 0-255 range

Y601 = 0.257R + 0.504G + 0.098B + 16

Cb = -0.148R – 0.291G + 0.439B + 128

Cr = 0.439R – 0.368G – 0.071B + 128

Y709 = 0.183R + 0.614G + 0.062B + 16

Cb = -0.101R - 0.338G + 0.439B + 128

Cr = 0.439 – 0.399G – 0.040B + 128

Fraction Number (exaple, 1bit is signed bit and 9bits are fraction bit)

Ex1) 0.098 = 0.5* '0' + 0.25* '0' + 0.125* '0' + 0.0625* '1' + 0.03125* '1' + 0.015625* '0' + 0.0078125* '0' + 0.00390625* '1' + 0.001953125* '0' = 0(signed bit) 0 0 0 1 1 0 0 1 0 = 0x032

Ex2) -0.148 -> First of all, let's think about 0.148

0. 148 = 0.5* '0' + 0.25* '0' + 0.125* '1' + 0.0625* '0' + 0.03125* '0' + 0.015625* '1' + 0.0078125* '0' + 0.00390625* '1' + 0.001953125* '1' = 0(signed bit) 0 0 1 0 0 1 0 1 1 = 10'b0001001011

Now, to change the number from 0.148 to -0.148, we have to derive 2's compliment of 0.148

10'b 0 0 0 1 0 0 1 0 1 1 -> 10'b 1 1 1 0 1 1 0 1 0 0 (bitwise invert) + 1 = 10'b 1 1 1 0 1 1 0 1 0 1 = 0x3B5

2.26 MIXER_COLOR_SPACE_CONVERSION_COEF_CB REGISTER (MIXER_CM_COEFF_CB, R/W, ADDRESS = 0XF020_ 0084)

MIXER_CM_COEFF_CB	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
REG_COEFF_01	[29:20]	Scaled color space conversion coefficient (C01). [29] = Sign-bit [28:20] = Fractional bit (Default value: -0.0742785 in decimal) (Recommended value : -0.148 in decimal, 0x3b5 in hexa-decimal)	0x3b4
REG_COEFF_11	[19:10]	Scaled color space conversion coefficient (C11). [19] = Sign-bit [18:10] = Fractional bit (Default value: -0.1455078125 in decimal) (Recommended value : -0.291 in decimal, 0x36c in hexa-decimal)	0x36b
REG_COEFF_21	[9:0]	Scaled color space conversion coefficient (C21). [9] = Sign-bit [8:0] = Fractional bit (Default and Recommended value : 0.439 in decimal)	0xe1

2.27 MIXER_COLOR_SPACE_CONVERSION_COEF_CR REGISTER (MIXER_CM_COEFF_CR, R/W, ADDRESS = 0XF020_ 0088)

MIXER_CM_COEFF_CR	Bit	Description	Reset Value
Reserved	[31:30]	Reserved, read as zero, do not modify	0
REG_COEFF_02	[29:20]	Scaled color space conversion coefficient (C_{02}). [29] = Sign-bit [28:20] = Fractional bit (Default and Recommended value : 0.439 in decimal)	0xe1
REG_COEFF_12	[19:10]	Scaled color space conversion coefficient (C_{12}). [19] = Sign-bit [18:10] = Fractional bit (Default and Recommended value : -0.368 in decimal)	0x344
REG_COEFF_22	[9:0]	Scaled color space conversion coefficient (C_{22}). [9] = Sign-bit [8:0] = Fractional bitobs (Default and Recommended value : -0.071 in decimal)	0x3dc

3 LAYERS

Mixer blends all the image sources such as video layer, graphic layer, and background layer and transfers the blended pixel data to TVNEC/HDMI. Set layer's priority value to select the order of the blending operation. Video and 2 graphic layers can be fully ordered and blended. The background layer is always the lowest layer.

- Background → (Video ↔ Graphic0 ↔ Graphic1)

Video layer and graphic layer are enabled or disabled by the register setting. The blending factor differs layer by layer like following:

- Background layer: No blending factor, as it is the lowest layer.
- Video layer: Video layer has one blending factor that is applied to all the pixels in the video layer. MIXER_VIDEO_CFG [7:0](REG_ALPHA_VID) is the blending factor. The video blending is enabled or disabled.
- Graphic0 layer: Graphic0 layer supports pixel blending and window blending. Pixel blending factors are applied pixel-by-pixel although window blending factor is applied all the pixels in the graphic layer. These two blending factors are applied simultaneously to a pixel.
- Graphic1 layer: Graphic1 layer supports pixel blending and window blending. Pixel blending factors are applied pixel-by-pixel although window blending factor is applied on all the pixels in the graphic layer. These two blending factors are applied simultaneously to a pixel.

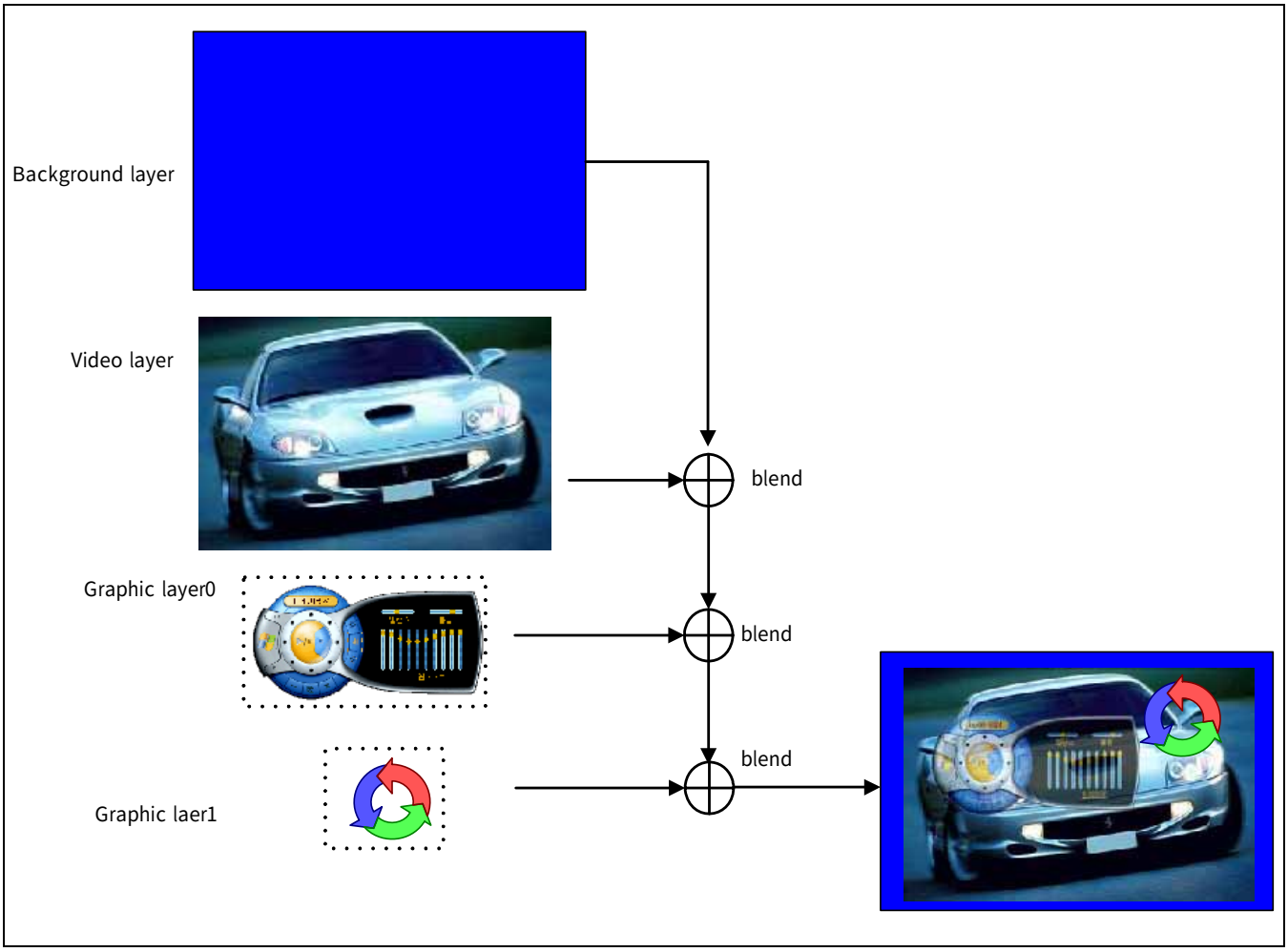


Figure 9.9-3 Mixer Blending

3.1 VIDEO LAYER

Video data is directly transferred from Video Processor to Mixer in YCbCr [888] 4:4:4 formats. As the Video Processor scales the source image in letterbox mode, the display region of the video data is smaller than the screen size. In this case, background layer is seen in the blank region.

3.2 GRAPHIC LAYER

ARM or Graphic Accelerator generates the graphic source data in the external memory and they are transferred to Mixer by AXI access. Mixer supports the following graphic formats.

- 16bpp RGB [565]
- 16bpp ARGB [1555]
- 16bpp ARGB [4444]
- 32bpp ARGB [8888]

In 16/32-bpp direct modes, the value of a pixel data directly indicates the RGB but the bit width for R, G, and B are different for each mode. For 16bpp direct ARGB [4444] mode, the RGB component is assigned to 16 bit length like the following.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
alpha factor				R				G				B			

Figure 9.9-4 16bpp ARGB Example

The internal data path is processed with YCbCr[888] format, therefore RGB format is converted to YCbCr by color matrix conversion.

If the bit-per-pixel (BPP) of color format is smaller than 8 bits, that value is used after expanding (Refer Figure 9.9-4 example (**ARGB [1555]**)).

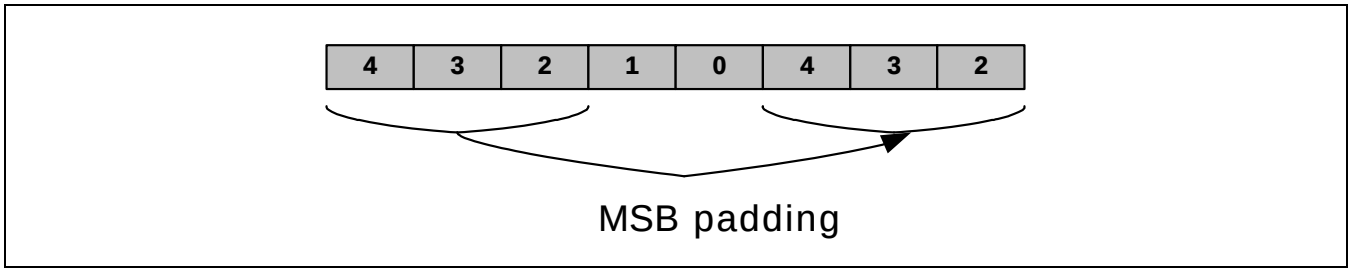


Figure 9.9-5 Example of Expanding

Mixer supports up to two graphic layers and one video layer. Each layer is enabled or disabled and user can configure the priority between layers. There is each blending factor between layers. The graphic layers has different color format.

When specifying the X/Y coordinates and the width/height of a graphic layer, the graphic layers should be located in the display region (720x480 in NTSC display mode, 720x576 in PAL display mode, 1280x720p in HD display mode). The Mixer does not support the clipping operation for the pixels that are displayed out of screen.

3.3 BLANK PIXEL

Blank pixel data in graphic layer is a pixel data that is transparent to the lower layer. You can define a blank pixel data in the register(MIXER_GRAPHICn_BLANK) and if the graphic data is same as the blank pixel value, a lower image is seen instead of the blank pixel.

3.4 SOURCE DATA IN MEMORY

As the graphic data comes from the external memory through the bus, the memory format for the source data is dependent on the bus system endian. In little endian system, the lower 8 bits are stored in the lower address.

The source data format in the Mixer is aligned in little endian or big endian format. This different endian format is applicable to the graphic data.

Mixer supports many graphic formats. The register format of the supported source formats is shown in. The

following picture shows the pixels in a display that is seen through human eyes.

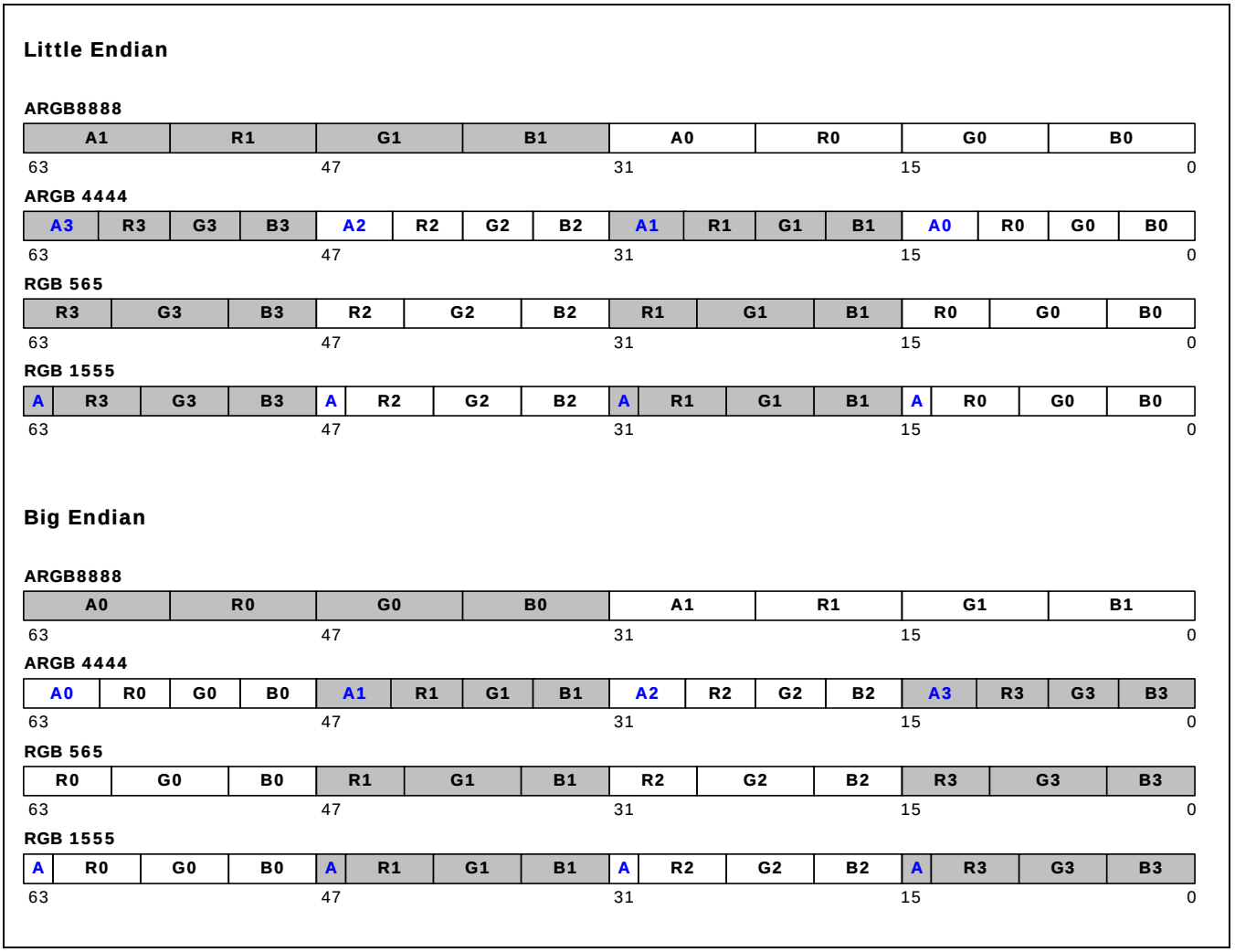


Figure 9.9-6 Graphic Data Format in Memory

Pixels with lower X coordinates are located from the left. These pixel data are represented by different digital data depending on the graphic format setting. For example, one graphic pixel is represented by 16 bit digital data in 16 BPP mode. If these pixels are processed in the Mixer, the word format to represent these pixels is different depending on the endian format.

In the big endian mode, the pixel with lower X coordinate is positioned to the MSB parts in the 64bit register of Mixer. However, in the little endian mode, the pixel with lower X coordinate is positioned to the LSB parts in the 64bit register.

3.5 BACKGROUND LAYER

If there is no video or graphic, other layers in the display region, background color is seen in that region. Use YCbCr[888] format to set the background color in the register.

NOTES

9.10 HDMI

1 OVERVIEW

This section describes the functionality of S5PC100X's HDMI system. This module is designed to transmit video and audio data through HDMI physical layer.

1.1 FEATURE

- HDMI v1.2/DVI v1.0 compliant
  - √ HDCP v1.1 compliant
  - √ 80-bits key in EFROM is used for HAES key to generate AES key
  - √ Video formats
  - √ 480p @ 59.94/60Hz
  - √ 576p @ 50Hz
  - √ 720p @ 50/59.94/60Hz
  - √ Other various formats up to 74.25MHz pixel clock (See Clock Controller Chapter)
    - 480p / 576p : 27Mhz
    - 720p : 74.25Mhz
  - √ 4:4:4 YCbCr format for HDMI & DVI input
  - √ 4:4:4 YCbCr format & 8:8:8 RGB Limited range format for HDMI output
  - √ 8:8:8 RGB Limited / Full range format for DVI output
- Packet transmission capabilities
  - √ Audio Clock Regeneration (ACR) Packet is transmitted at 128xFs/N rate.
  - √ GCP packet for AV mute control
  - √ ACP, ISRC1 and ISRC2
  - √ AVI, Audio and MPEG Source Infoframes
  - √ Vender Specific Infoframe or Source Product Descriptor Infoframe
- SPDIF Receiver
- Supports Linear PCM data interface and Non-linear PCM formats
  - √ Supports up to 2 channels @ 192 kHz 24 bit L-PCM
  - √ IEC61937 encoded bit-stream format
  - √ SACD DSD stream up to 6 channels
- CSC
  - √ BT.709 YCbCr to RGB (full range (0~255)/limited range (16~235))

1.2 BLOCK DESCRIPTION

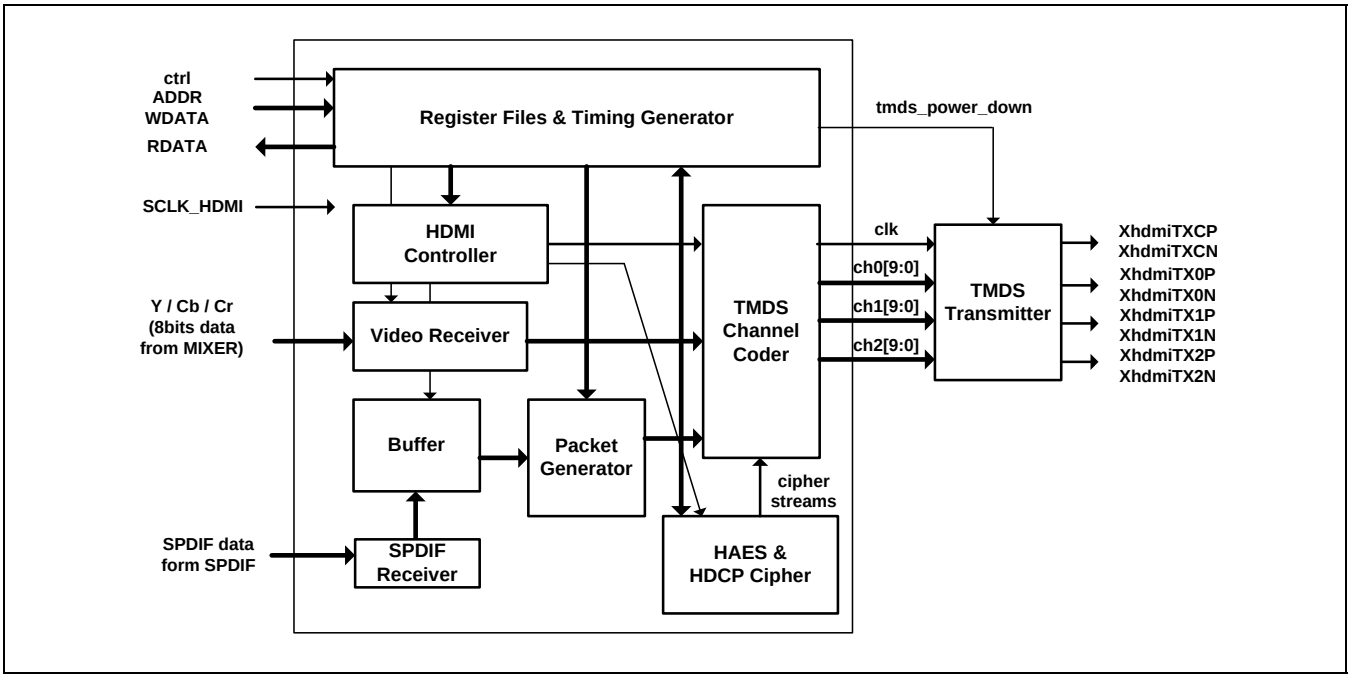


Figure 9.10-1 HDMI Transmitter Block Diagram

1.2.1 Register File

Register File contains all registers setting and its interface uses 8-bit bus.
All of the packet data except the audio sample packet are stored in the register file.

1.2.2 HDMI Controller

Timing Generator in HDMI Controller generates the HDMI timing information such as pixel and line numbers according to the horizontal active signal, horizontal sync signal, vertical sync signal and etc.

Second role is to control the packet transmission. It checks the timing information and if available, it generates control signals to audio FIFO and packet generator.

a) Packet transmission control consideration

Most of packets are transmitted once per video frame (or field), such as GCP, AVI, and audio infoframes.

Especially, ACR packet should be transmitted at a rate of 128xFs/N where Fs are sampling frequency of audio sample and N is content of ACR packet.

To meet this requirement, the system controller has video pixel clock counter and it counts that clock. HDMI transmits ACR packet each time after it reaches to the CTS value.

Audio sample packets are transmitted any time if audio sample is available and packet transmission is allowed to meet the audio sample packet jitter requirement.

1.2.3 Video Receiver

Video Receiver deals with the input pixel data as specified format. Main functions of this module are as follows

a) Color Space Conversion

It converts the input color space to another as listed below.

- By-pass
- RGB full range to YCbCr BT.601 (or BT.709)
- RGB limited range to YCbCr BT.601 (or BT.709)
- YCbCr BT.601 (or BT.709) to RGB full range or limited range

b) 4:2:2 Mode Management

Rearrange the input pixel values to fit the 4:2:2 12-bit formats.

1.2.4 SPDIF Receiver

SPDIF Recevier decodes serial input data of SPDIF format type to extract 2-ch PCM or 5.1-ch stream format audio data and 4-bits extra information. It sends decoded audio data to HDMI audio buffer.
It handles up to 192 KHz Fs data if 2-ch PCM is used.

1.2.5 Buffer

Buffer receives the audio PCM samples from the external and keeps them on the buffer (FIFO). The buffer size must be selected carefully because buffering just one sample for 8 channels requires 24x8-bit registers, which makes the chip size bigger!!

The audio sample buffer size is 8x8x24-bits.

In addition, it receives and keeps 2 channels' sub-frame data, such as U, V and C of IEC60958 format. Parity bits are automatically generated in this module.

1.2.6 Packet Generator

It has packet multiplexer and CRC generator for each sub-packet and header data.

1.2.7 TMDS Channel Coder

TMDS Channel coder translates the 8-bit input data to 10-bit transition minimized code as specified in HDMI v1.x.

1.2.8 TMDS Transmitter

TMDS Transmitter block receives R/G/B DC-balanced parallel data from TMDS Channel Coder block and serializes data.

1.2.9 HDCP Cipher

It generates the cipher streams for HDMI data as specified in HDCP specification v1.1. It also processes the data for first, second and the third authentication.

It requires external key memory.

1.3 AUDIO INTERFACE

1.3.1 Audio Mode Setting

There are 4 audio modes. In multi-channel mode, the SP_PRE bits are used for Audio Sample Packet (ASP) header. In 2 channel mode, those bits are automatically generated.

Audio Mode	ASP_CON Register Setting		
	SACD_EN[5]	AUD_MODE[4]	SP_PRE[3:0]
2 channel L-PCM	0	0	don't care
-	0	1	Reserved
2 channel DSD (SACD)	1	0	don't' care
-	1	1	Reserved

NOTE: The number in the brace is the bit index on ASP_CON register

1.3.2 Two Channels L-PCM/ DSD Mode

In this mode, i_pcm0L and i_pcm0R channels are used. In addition, full 24-bits of i_pcm0L and i_pcm0R are used. The i_cuvL and i_cuvR is attached to the i_pcm0L and i_pcm0R to generate ASP.

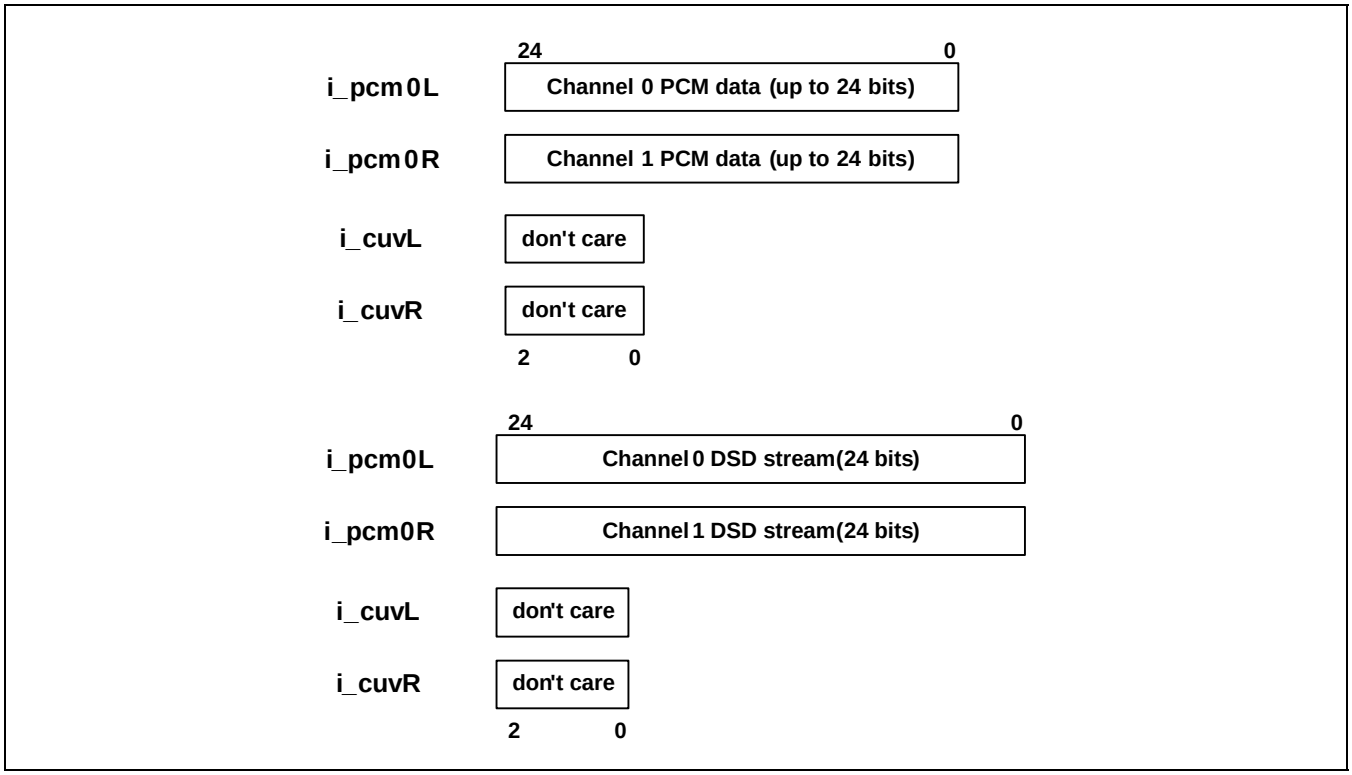


Figure 9.10-2 24-Bits 2-Ch L-PCM or 5.1-Ch DSD Mode

For shorter bit width case, for example, 16-bits per channel, each 16-bit PCM data should be located as left justified with LSB zero padding. The '0' in the following figure is in hexadecimal format.

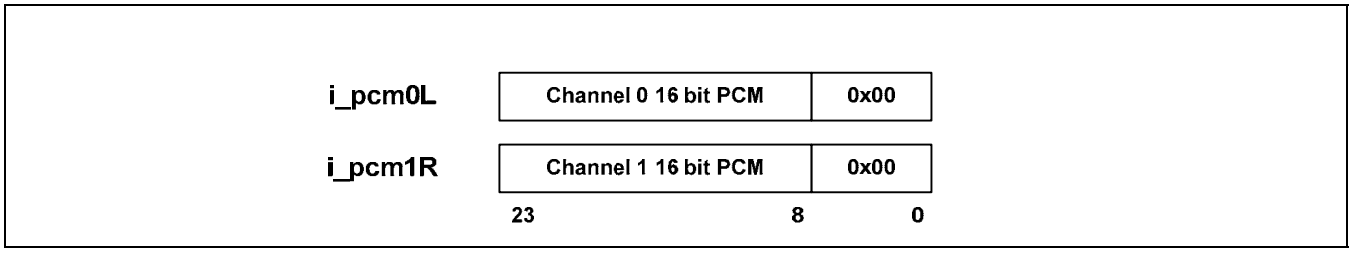


Figure 9.10-3 16-Bit 2-Ch L-PCM / DSD Mode

This is also applicable to 2 channel DSD mode.

1.4 HAES

HAES module decrypts AES-encrypted data in external memory-type device.

Each device supporting HDMI should have its own HDCP key to enable HDMI full features. S5PC100 system needs to protect HDCP key data from hacking and hijacking using 80bits Key in EFROM and AES encryption technique.

NOTE: Utility program to generate HAES encrypted HDCP KEY will be delivered with NDA.

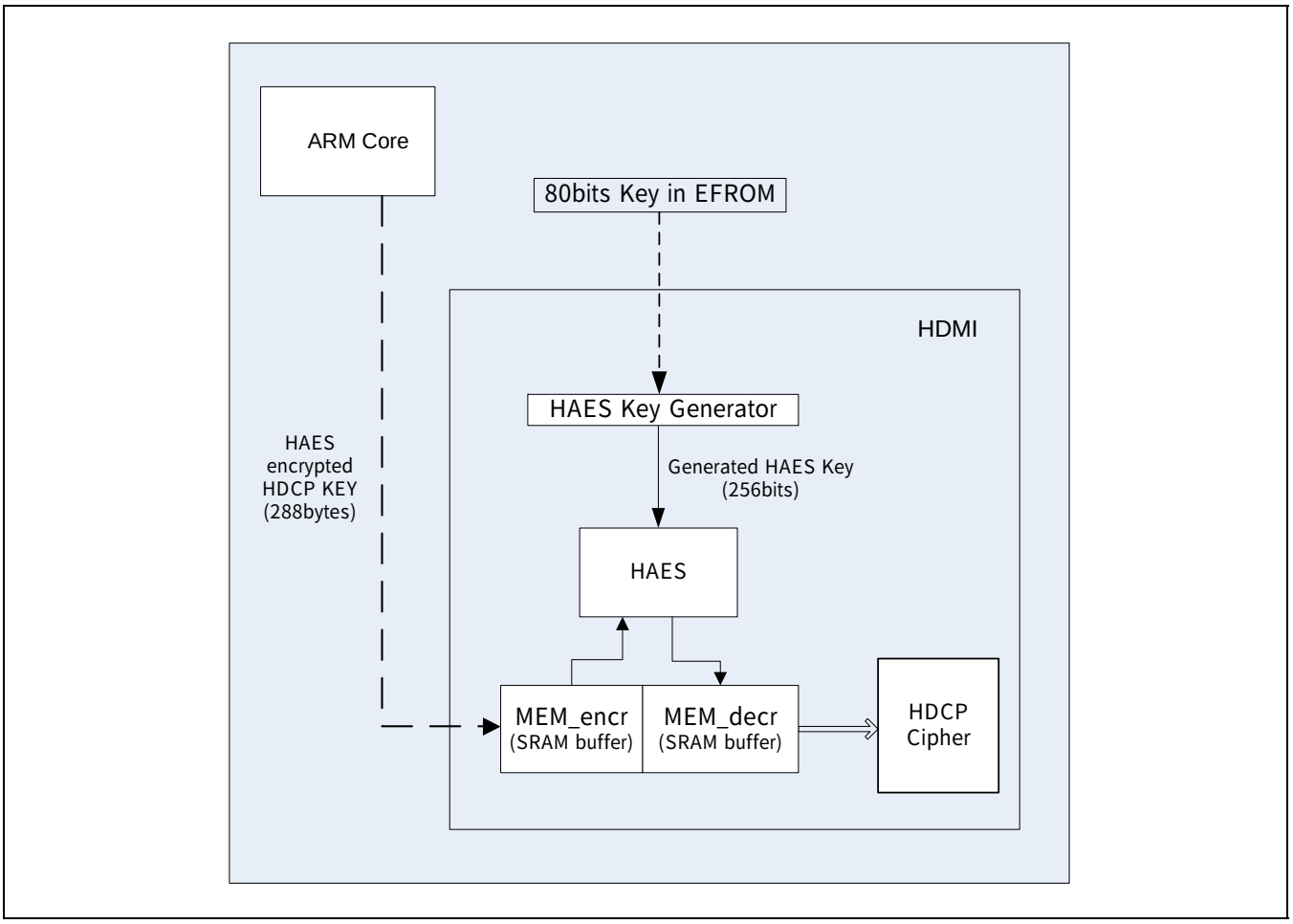


Figure 9.10-4 HDMI Key Protection

To decrypt HAES encrypted HDCP key, S5PC100 gets information of HAES key used in encryption by utility program and HAES encrypted HDCP key message before starting HDMI. HAES Key Generator generated HAES key based on 80bits Key in EFROM and generated HAES Key will be fed to HAES core automatically when HAES_CON[0] is set. ARM core writes HAES encrypted HDCP key message into MEM_encr with the amount of decryption size desired and sets HAES_DATA_SIZE_H/L SFR to decide how many bytes to decrypt and it should be 128-bit aligned because HAES core is operating with 128-bit data.

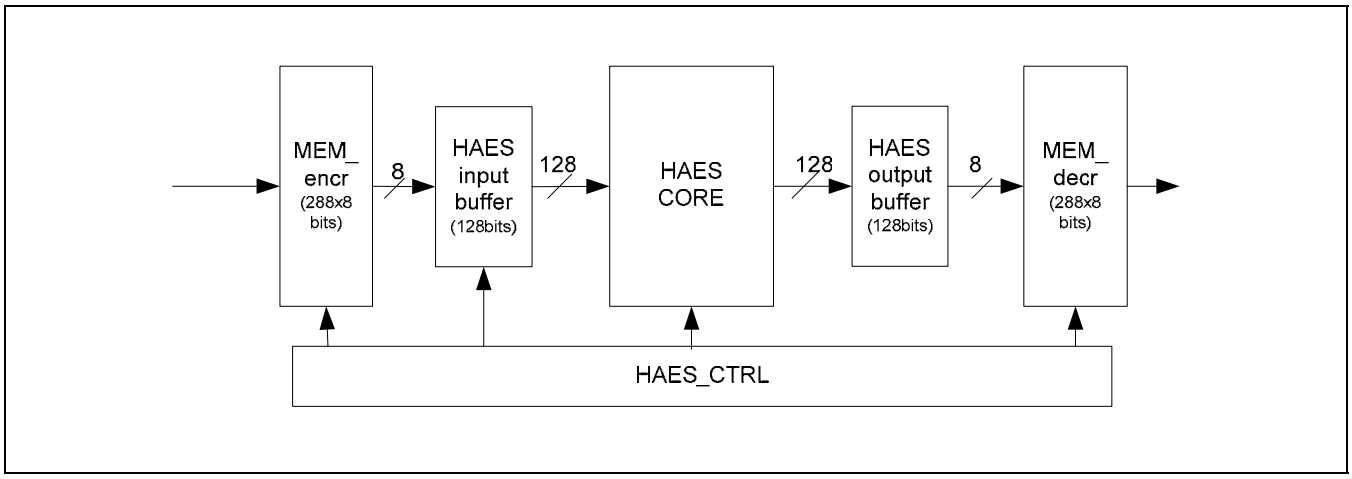


Figure 9.10-5 Decryption Flow of HDCP Key

HAES core starts to decrypt HAES encrypted message data when HAES_CON[0] bit is set. HAES core reads generated HAES key and HAES encrypted message data in MEM_encr. If the data is decrypted and MEM_decr is full with decrypted data, then the HAES_CON[0] flag goes down to 0 to indicate the decryption and transfer is complete (MEM_encr means SRAM buffer to hold AES-encrypted data. MEM_decr means SRAM buffer to hold AES-descrypted data).

After checking that HAES_CON[0] flag is 0, then decrypted data can be used for HDCP key. After generation of HDCP key is done, HDCP module in HDMI will reference HDCP key automatically when HDCP sequence (HDCP_CTRL[1] == 1'b1) is started.

1.5 COLOR SPACE CONVERTER

Color Space Converter (CSC) in Video Recevier block outputS RGB or YCbCr format according to display mode. CSC coverts YCbCr601 to YCbCr709, YCbCr601 to RGB (Full range/ Limited range), YCbCr709 bypass, YCbCr709 to RGB (Full range/ Limited range).

Transform matrix

$$\begin{bmatrix} out\_aG\_Y \\ out\_aB\_Cb \\ out\_aR\_Cr \end{bmatrix} = \begin{bmatrix} Y\_G\_coef & Y\_B\_coef & Y\_R\_coef \\ Cb\_G\_coef & Cb\_B\_coef & Cb\_R\_coef \\ Cr\_G\_coef & Cr\_B\_coef & Cr\_R\_coef \end{bmatrix} \bullet \begin{bmatrix} i\_Y \\ i\_Cb \\ i\_Cr \end{bmatrix}$$

a) YCbCr (601) => YCbCr (709)

$$\begin{bmatrix} {}^{709}_{219}Y' \\ C_B \\ C_R \end{bmatrix} = \begin{bmatrix} 1 & -0.115550 & -0.207938 \\ 0 & 1.018640 & 0.114618 \\ 0 & 0.075049 & 1.025327 \end{bmatrix} \bullet \begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix}$$

$$\begin{bmatrix} {}^{709}_{219}Y' \\ C_B \\ C_R \end{bmatrix} = \frac{1}{256}\begin{bmatrix} 256 & -29.5808 & -53.232128 \\ 0 & 260.77184 & 29.342208 \\ 0 & 19.212544 & 262.483712 \end{bmatrix} \bullet \begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix}$$

b) YCbCr (601) => RGB (16~235)

$$\begin{bmatrix} {}_{219}R' \\ {}_{219}G' \\ {}_{219}B' \end{bmatrix} = \frac{1}{256}\begin{bmatrix} 256 & 0 & 350.901 \\ 256 & -86.132 & -178.738 \\ 256 & 443.506 & 0 \end{bmatrix} \bullet \left(\begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

$$\begin{bmatrix} {}_{219}R' \\ {}_{219}G' \\ {}_{219}B' \end{bmatrix} = \frac{1}{256}\begin{bmatrix} 256 & 0 & 350.901 \\ 256 & -86.132 & -178.738 \\ 256 & 443.506 & 0 \end{bmatrix} \bullet \left(\begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

c) YCbCr (601) => RGB (0~255)

$$\begin{bmatrix} {}_{255}R' \\ {}_{255}G' \\ {}_{255}B' \end{bmatrix} = \begin{bmatrix} 1.1643828 & 0 & 1.596027 \\ 1.1643828 & -0.390625 & -0.81296875 \\ 1.1643828 & 1.7324453 & 0 \end{bmatrix} \bullet \left(\begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

d) YCbCr (709) => RGB (16~235)

$$\begin{bmatrix} {}_{219}R' \\ {}_{219}G' \\ {}_{219}B' \end{bmatrix} = \frac{1}{256}\begin{bmatrix} 256 & 0 & 394.150 \\ 256 & -46.885 & -117.165 \\ 256 & 464.430 & 0 \end{bmatrix} \bullet \left(\begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

e) YCbCr (709) => RGB (0~255)

$$\begin{bmatrix} {}_{255}R' \\ {}_{255}G' \\ {}_{255}B' \end{bmatrix} = \frac{1}{256}\begin{bmatrix} 298.082 & 0 & 458.942 \\ 298.082 & -54.592 & -136.425 \\ 298.082 & 540.775 & 0 \end{bmatrix} \bullet \left(\begin{bmatrix} {}^{601}_{219}Y' \\ C_B \\ C_R \end{bmatrix} - \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \right)$$

1.6 TIMING GENERATOR

TG generates reference timing according to video format of output. Figure 9.10-2 and Figure 9.10-2 shows parameter sets. Figure 9.10-2 shows the reference timing for progressive mode and Figure 9.10-6 for interlaced mode.

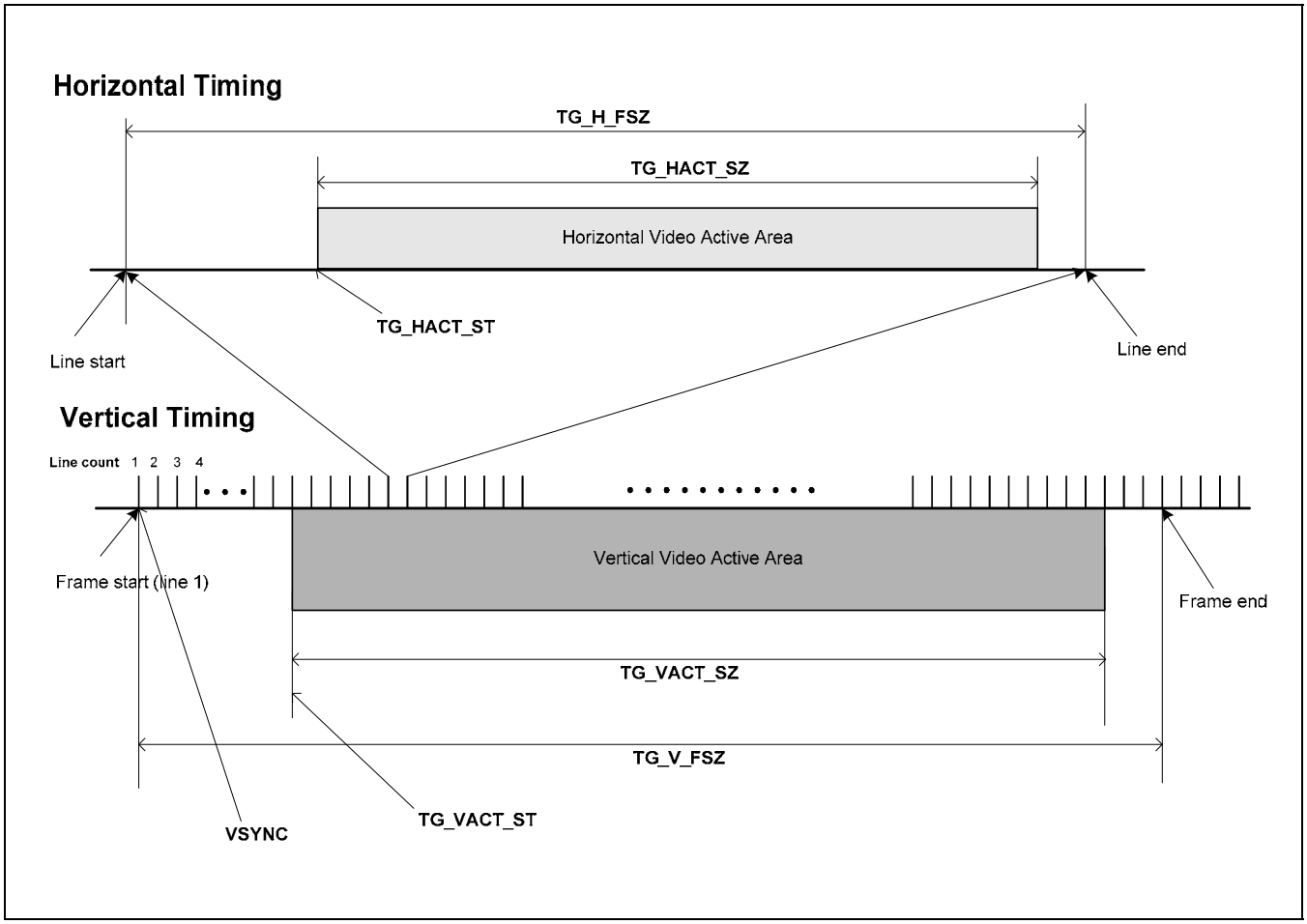


Figure 9.10-6 Progressive Mode Reference Timing Generation

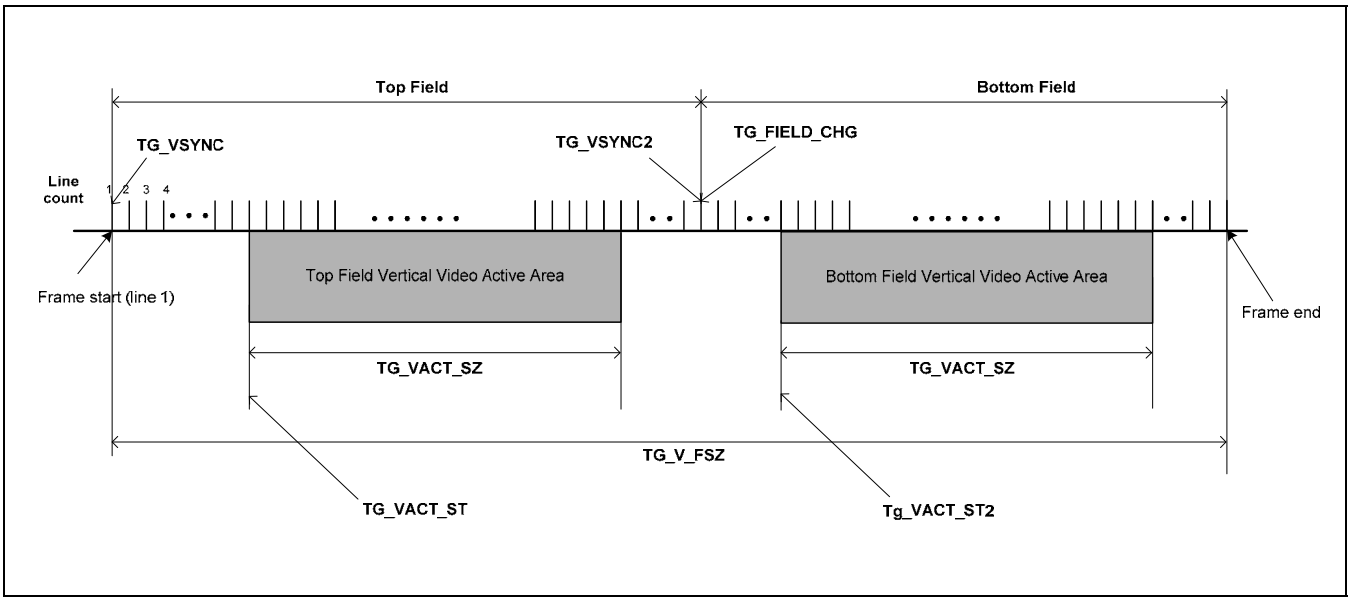


Figure 9.10- 7 Interlace Mode Reference Timing Generation (Expansion from Progressive Mode)

1.6.1 TG Register Setting Example

Table 9.10-1 TG Setting Example

Register Name	60Hz		50Hz	
	480p	720p	576p	720p
TG_H_FSZ_H/L	858	1650	864	1980
TG_HACT_ST_H/L	138	370	144	700
TG_HACT_SZ_H/L	720	1280	720	1280
TG_V_FSZ_H/L	525	750	625	750
TG_VSYNC_H/L		1		1
TG_VSYNC2_H/L	NA	NA	NA	NA
TG_VACT_ST_H/L	45	30	49	30
TG_VACT_SZ_H/L	480	720	480	720
TG_FIELD_CHG_ H/L	NA	NA	NA	NA
TG_VACT_ST2_H/L	NA	NA	NA	NA
TG_VSYNC_TOP_HDMI_H/L		1		1
TG_VSYNC_BOT_HDMI_H/L		563		563
TG_FIELD_TOP_HDMI_H/L		1		1
TG_FIELD_BOT_HDMI_H/L		563		563

2 I/O DESCRIPTION

Function Signal	I/O	Description	Pad	Type
HDMI_TX0P	Output	Differential Data 0 Positive	XhdmiTX0P	Dedicated
HDMI_TX0N	Output	Differential Data 0 Negative	XhdmiTX0N	Dedicated
HDMI_TX1P	Output	Differential Data 1 Positive	XhdmiTX1P	Dedicated
HDMI_TX1N	Output	Differential Data 1 Negative	XhdmiTX1N	Dedicated
HDMI_TX2P	Output	Differential Data 2 Positive	XhdmiTX2P	Dedicated
HDMI_TX2N	Output	Differential Data 2 Negative	XhdmiTX2N	Dedicated
HDMI_TXCP	Output	Differential Clock Positive	XhdmiTXCP	Dedicated
HDMI_TXCN	Output	Differential Clock Negative	XhdmiTXCN	Dedicated
HDMI_REXT	-	External Register	XhdmiREXT	Dedicated

NOTES:

1. The coloum Type in the above table indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.
2. 1.5Kohm register is required to XhdmiREXT.

3. Drive strength for OSC_27M to generate SCLK_HDMI should be more than 2x. Refer Clock Controller & Pad_Control manual how to set properly for ETC4DRV[0]

3 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
HDMI_CON_0	0xF030_0000	R/W	HDMI System Control Register 0	0x00
HDMI_CON_1	0xF030_0004	R/W	HDMI System Control Register 1	0x00
HDMI_CON_2	0xF030_0008	R/W	HDMI System Control Register 2.	0x00
STATUS	0xF030_0010	R/W	HDMI System Status Register	0x00
STATUS_EN	0xF030_0020	R/W	HDMI System Status Enable Register	0x00
HPD	0xF030_0030	R/W	Hot Plug Detection Control Register	0x00
MODE_SEL	0xF030_0040	R/W	HDMI/DVI Mode Selection	0x00
ENC_EN	0xF030_0044	R/W	HDCP Encryption Enable Register	0x00
BLUE_SCREEN_0	0xF030_0050	R/W	Pixel Values for Blue Screen	0x00
BLUE_SCREEN_1	0xF030_0054	R/W	Pixel Values for Blue Screen	0x00
BLUE_SCREEN_2	0xF030_0058	R/W	Pixel Values for Blue Screen	0x00
HDMI_YMAX	0xF030_0060	R/W	Maximum Y (or R,G,B) Pixel Value	0xEB
HDMI_YMIN	0xF030_0064	R/W	Minimum Y (or R,G,B) Pixel Value	0x10
HDMI_CMAX	0xF030_0068	R/W	Maximum Cb/ Cr Pixel Value	0xF0
HDMI_CMIN	0xF030_006C	R/W	Minimum Cb/ Cr Pixel Value	0x10
VBI_ST_MG	0xF030_0080	R/W	Packet Transmission Start Margin during VBI.	0x3C
VBI_END_MG	0xF030_0084	R/W	Packet Transmission End Margin during VBI	0x30
VACT_ST_MG	0xF030_0088	R/W	Packet Transmission Start Margin during Active Video	0x2C
VACT_END_MG	0xF030_008C	R/W	Packet Transmission Start Margin during Active Video.	0x2E
H_BLANK_0	0xF030_00A0	R/W	Horizontal Blanking Setting	0x00
H_BLANK_1	0xF030_00A4	R/W	Horizontal Blanking Setting	0x00
V_BLANK_0	0xF030_00B0	R/W	Vertical Blanking Setting	0x00
V_BLANK_1	0xF030_00B4	R/W	Vertical Blanking Setting	0x00
V_BLANK_2	0xF030_00B8	R/W	Vertical Blanking Setting	0x00
H_V_LINE_0	0xF030_00C0	R/W	Hori. Line and Ver. Line	0x00
H_V_LINE_1	0xF030_00C4	R/W	Hori. Line and Ver. Line	0x00
H_V_LINE_2	0xF030_00C8	R/W	Hori. Line and Ver. Line	0x00
SYNC_MODE	0xF030_00E4	R/W	Vertical Sync Polarity Control Register	0x00
INT_PRO_MODE	0xF030_00E8	R/W	Interlace/ Progressive Control Register	0x00
SEND_START_0	0xF030_00F0	R/W	HDCP Enable Signal Position Register	0x00

Register	Address	R/W	Description	Reset Value
SEND_START_1	0xF030_00F4	R/W	HDCP Enable Signal Position Register	0x00
SEND_END_0	0xF030_0100	R/W	HDCP Enable Signal Position Register	0xFF
SEND_END_1	0xF030_0104	R/W	HDCP Enable Signal Position Register	0xFF
SEND_END_2	0xF030_0108	R/W	HDCP Enable Signal Position Register	0x0F
V_BLANK_F_0	0xF030_0110	R/W	Vertical Blanking Setting for Bottom Field	0x00
V_BLANK_F_1	0xF030_0114	R/W	Vertical Blanking Setting for Bottom Field	0x00
V_BLANK_F_2	0xF030_0118	R/W	Vertical Blanking Setting for Bottom Field	0x00
H_SYNC_GEN_0	0xF030_0120	R/W	Horizontal Sync Generation Setting	0x00
H_SYNC_GEN_1	0xF030_0124	R/W	Horizontal Sync Generation Setting	0x00
H_SYNC_GEN_2	0xF030_0128	R/W	Horizontal Sync Generation Setting	0x00
V_SYNC_GEN_1_0	0xF030_0130	R/W	Vertical Sync Generation for Top Field or Frame.	0x01
V_SYNC_GEN_1_1	0xF030_0134	R/W	Vertical Sync Generation for Top Field or Frame.	0x10
V_SYNC_GEN_1_2	0xF030_0138	R/W	Vertical Sync Generation for Top Field or Frame.	0x00
Reserved	0xF030_0140	R/W	-	0x01
Reserved	0xF030_0144	R/W	-	0x10
Reserved	0xF030_0148	R/W	-	0x00
Reserved	0xF030_0150	R/W	-	0x01
Reserved	0xF030_0154	R/W	-	0x10
Reserved	0xF030_0158	R/W	-	0x00
ASP_CON	0xF030_0160	R/W	ASP Packet Control Register	0x00
ASP_SP_FLAT	0xF030_0164	R/W	ASP Packet sp_flat Bit Control	0x00
ASP_CHCFG0	0xF030_0170	R/W	ASP Audio Channel Configuration	0x04
ASP_CHCFG1	0xF030_0174	R/W	ASP Audio Channel Configuration	0x1A
ASP_CHCFG2	0xF030_0178	R/W	ASP Audio Channel Configuration	0x2C
ASP_CHCFG3	0xF030_017C	R/W	ASP Audio Channel Configuration	0x3E
ACR_CON	0xF030_0180	R/W	ACR Packet Control Register	0x00
ACR_MCTS0	0xF030_0184	R	Measured CTS Value	0x01
ACR_MCTS1	0xF030_0188	R	Measured CTS Value	0x00
ACR_MCTS2	0xF030_018C	R	Measured CTS Value	0x00
ACR_CTS0	0xF030_0190	R/W	CTS Value for Fixed CTS Transmission Mode.	0xE8
ACR_CTS1	0xF030_0194	R/W	CTS Value for Fixed CTS Transmission Mode.	0x03

Register	Address	R/W	Description	Reset Value
ACR_CTS2	0xF030_0198	R/W	CTS Value for Fixed CTS Transmission Mode.	0x00
ACR_N0	0xF030_01A0	R/W	N Value for ACR Packet.	0xE8
ACR_N1	0xF030_01A4	R/W	N Value for ACR Packet.	0x03
ACR_N2	0xF030_01A8	R/W	N Value for ACR Packet.	0x00
ACR_LSB2	0xF030_01B0	R/W	Altenate LSB for Fixed CTS Transmission Mode	0x00
ACR_TXCNT	0xF030_01B4	R/W	Number of ACR Packet Transmission per frame	0x1F
ACR_TXINTERVAL	0xF030_01B8	R/W	Interval for ACR Packet Transmission	0x63
ACR_CTS_OFFSET	0xF030_01BC	R/W	CTS Offset for Measured CTS mode.	0x00
GCP_CON	0xF030_01C0	R/W	GCP Packet Control register	0x00
GCP_BYTE1	0xF030_01D0	R/W	GCP Packet Body	0x00
ACP_CON	0xF030_01E0	R/W	ACP Packet Control register	0x00
ACP_TYPE	0xF030_01E4	W	ACP Packet Header	0x00
ACP_DATA00~16	0xF030_ 0200 ~ 0240	W	ACP Packet Body	0x00
ISRC_CON	0xF030_0250	R/W	ISRC Packet Control Register	0x00
ISRC1_HEADER1	0xF030_0264	W	ISCR1 Packet Header	0x00
ISRC1_DATA00~15	0xF030_ 0270 ~ 02AC	W	ISRC1 Packet Body	0x00
ISRC2_DATA00~15	0xF030_ 02B0 ~ 02EC	W	ISRC2 Packet Body	0x00
AVI_CON	0xF030_0300	R/W	AVI Packet Control Register	0x00
AVI_CHECK_SUM	0xF030_0310	W	AVI Packet Checksum	0x00
AVI_BYTE01~13	0xF030_ 0320 ~ 0350	R/W	AVI Packet Body	0x00
AUI_CON	0xF030_0360	R/W	AUI Packet Control Register	0x00
AUI_CHECK_SUM	0xF030_0370	R/W	AUI Packet Checksum	0x00
AUI_BYTE1~4	0xF030_ 0380 ~ 038C	R/W	AUI Packet Body	0x00
MPG_CON	0xF030_03A0	R/W	MPG Packet Control Register	0x00
MPG_CHECK_SUM	0xF030_03B0	W	MPG Packet Checksum	0x00
MPG_BYTE1~5	0xF030_ 03C0 ~ 03D0	W	MPG Packet Body	0x00
SPD_CON	0xF030_0400	R/W	SPD Packet Control Register	0x00
SPD_HEADER0	0xF030_0410	W	SPD Packet Header	0x00
SPD_HEADER1	0xF030_0414	W	SPD Packet Header	0x00

Register	Address	R/W	Description	Reset Value
SPD_HEADER2	0xF030_0418	W	SPD Packet Header	0x00
SPD_DATA00~27	0xF030_ 0420 ~ 048C	W	SPD Packet Body	0x00
HD_CSC_CON	0xF030_0490	R/W	Color Space Conversion Control Register	0x33
HDMI_CSC_COEF00L	0xF030_04A0	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF00H	0xF030_04A4	R/W	CSC Coefficient Registers	0x01
HDMI_CSC_COEF01L	0xF030_04A8	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF01H	0xF030_04AC	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF02L	0xF030_04B0	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF02H	0xF030_04B4	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF10L	0xF030_04B8	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF10H	0xF030_04BC	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF11L	0xF030_04C0	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF11H	0xF030_04C4	R/W	CSC Coefficient Registers	0x01
HDMI_CSC_COEF12L	0xF030_04C8	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF12H	0xF030_04CC	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF20L	0xF030_04D0	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF20H	0xF030_04D4	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF21L	0xF030_04D8	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF21H	0xF030_04DC	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF22L	0xF030_04E0	R/W	CSC Coefficient Registers	0x00
HDMI_CSC_COEF22H	0xF030_04E4	R/W	CSC Coefficient Registers	0x01
HDCP_RX_SHA1_0_0	0xF030_0600	R/W	SHA-1 Value from Repeater	0x00
HDCP_RX_SHA1_0_1	0xF030_0604	R/W	SHA-1 Value from Repeater	0x00
HDCP_RX_SHA1_0_2	0xF030_0608	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_0_3	0xF030_060C	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_1_0	0xF030_0610	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_1_1	0xF030_0614	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_1_2	0xF030_0618	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_1_3	0xF030_061C	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_2_0	0xF030_0620	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_2_1	0xF030_0624	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_2_2	0xF030_0628	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_2_3	0xF030_062C	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_3_0	0xF030_0630	R/W	SHA-1 value from Repeater	0x00

Register	Address	R/W	Description	Reset Value
HDCP_RX_SHA1_3_1	0xF030_0634	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_3_2	0xF030_0638	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_3_3	0xF030_063C	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_4_0	0xF030_0640	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_4_1	0xF030_0644	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_4_2	0xF030_0648	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_SHA1_4_3	0xF030_064C	R/W	SHA-1 value from Repeater	0x00
HDCP_RX_KSV_0_0	0xF030_0650	R/W	Receiver's KSV 0	0x00
HDCP_RX_KSV_0_1	0xF030_0654	R/W	Receiver's KSV 0	0x00
HDCP_RX_KSV_0_2	0xF030_0658	R/W	Receiver's KSV 0	0x00
HDCP_RX_KSV_0_3	0xF030_065C	R/W	Receiver's KSV 0	0x00
HDCP_RX_KSV_0_4	0xF030_0660	R/W	Receiver's KSV 0	0x00
HDCP_KSV_LIST_CTRL	0xF030_0664	R/W	HDCP KSV List Control	0x00
HDCP_SHA_RESULT	0xF030_0670	R/W	2^{nd} authentication status	0x00
HDCP_CTRL	0xF030_0680	R/W	HDCP Control	0x00
HDCP_CHECK_RESULT	0xF030_0690	R/W	HDCP Ri, Pj, V result	0x00
HDCP_BKSV_0_0	0xF030_06A0	R/W	Receiver's BKSV	0x00
HDCP_BKSV_0_1	0xF030_06A4	R/W	Receiver's BKSV	0x00
HDCP_BKSV_0_2	0xF030_06A8	R/W	Receiver's BKSV	0x00
HDCP_BKSV_0_3	0xF030_06AC	R/W	Receiver's BKSV	0x00
HDCP_BKSV_1	0xF030_06B0	R/W	Receiver's BKSV	0x00
HDCP_AKSV_0_0	0xF030_06C0	R/W	Transmitter's AKSV	0x00
HDCP_AKSV_0_1	0xF030_06C4	R/W	Transmitter's AKSV	0x00
HDCP_AKSV_0_2	0xF030_06C8	R/W	Transmitter's AKSV	0x00
HDCP_AKSV_0_3	0xF030_06CC	R/W	Transmitter's AKSV	0x00
HDCP_AKSV_1	0xF030_06D0	R/W	Transmitter's AKSV	0x00
HDCP_An_0_0	0xF030_06E0	R/W	Transmitter's An	0x00
HDCP_An_0_1	0xF030_06E4	R/W	Transmitter's An	0x00
HDCP_An_0_2	0xF030_06E8	R/W	Transmitter's An	0x00
HDCP_An_0_3	0xF030_06EC	R/W	Transmitter's An	0x00
HDCP_An_1_0	0xF030_06F0	R/W	Transmitter's An	0x00
HDCP_An_1_1	0xF030_06F4	R/W	Transmitter's An	0x00
HDCP_An_1_2	0xF030_06F8	R/W	Transmitter's An	0x00
HDCP_An_1_3	0xF030_06FC	R/W	Transmitter's An	0x00

Register	Address	R/W	Description	Reset Value
HDCP_BCAPS	0xF030_0700	R/W	Receiver's BCAPS	0x00
HDCP_BSTATUS_0	0xF030_0710	R/W	Receiver's BSTATUS	0x00
HDCP_BSTATUS_1	0xF030_0714	R/W	Receiver's BSTATUS	0x00
HDCP_Ri_0	0xF030_0740	R/W	Transmitter's Ri	0x00
HDCP_Ri_1	0xF030_0744	R/W	Transmitter's Ri	0x00
HDCP_Pj	0xF030_0750	R/W	Transmitter's Pj	0x00
HDCP_OFFSET_TX_0	0xF030_0760	R/W	Memory Address Offset	0x00
HDCP_OFFSET_TX_1	0xF030_0764	R/W	Memory Address Offset	0x00
HDCP_OFFSET_TX_2	0xF030_0768	R/W	Memory Address Offset	0x00
HDCP_OFFSET_TX_3	0xF030_076C	R/W	Memory Address Offset	0x00
HDCP_CYCLE_AA	0xF030_0770	R/W	Memory Read Cycle Count	0x00
TG_CMD	0xF030_1000	R/W	Command Register	0x00
TG_H_FSZ_L	0xF030_1018	R/W	Horizontal Full Size	0x72
TG_H_FSZ_H	0xF030_101C	R/W	Horizontal Full Size	0x06
TG_HACT_ST_L	0xF030_1020	R/W	Horizontal Active Start	0x05
TG_ HACT_ST_H	0xF030_1024	R/W	Horizontal Active Start	0x01
TG_ HACT_SZ_L	0xF030_1028	R/W	Horizontal Active Size	0x00
TG_ HACT_SZ_H	0xF030_102C	R/W	Horizontal Active Size	0x05
TG_V_FSZ_L	0xF030_1030	R/W	Vertical Full Line Size	0xEE
TG_V_FSZ_H	0xF030_1034	R/W	Vertical Full Line Size	0x02
TG_VSYNC_L	0xF030_1038	W	Vertical Sync Position	0x01
TG_VSYNC_H	0xF030_103C	W	Vertical Sync Position	0x00
TG_VSYNC2_L	0xF030_1040	W	Vertical Sync Position for Bottom Field	0x33
TG_VSYNC2_H	0xF030_1044	W	Vertical Sync Position for Bottom Field	0x02
TG_VACT_ST_L	0xF030_1048	R/W	Vertical Sync Active Start Position	0x1a
TG_VACT_ST_H	0xF030_104C	R/W	Vertical Sync Active Start Position	0x00
TG_VACT_SZ_L	0xF030_1050	R/W	Vertical Active Size	0xd0
TG_VACT_SZ_H	0xF030_1054	R/W	Vertical Active Size	0x02
TG_FIELD_CHG_L	0xF030_1058	R/W	Field Change Position	0x33
TG_FIELD_CHG_H	0xF030_105C	R/W	Field Change Position	0x02
TG_VACT_ST2_L	0xF030_1060	R/W	Vertical Sync Active Start Position for Bottom Field	0x48
TG_VACT_ST2_H	0xF030_1064	R/W	Vertical Sync Active Start Position for Bottom Field	0x02
TG_VSYNC_TOP_HDMI_L	0xF030_1078	R/W	HDMI Vsync Positon for Top Field	0x01

Register	Address	R/W	Description	Reset Value
TG_VSYNC_TOP_HDMI_H	0xF030_107C	R/W	HDMI Vsync Positon for Top Field	0x00
TG_VSYNC_BOT_HDMI_L	0xF030_1080	R/W	HDMI Vsync Positon for Bottom Field	0x33
TG_VSYNC_BOT_HDMI_H	0xF030_1084	R/W	HDMI Vsync Positon for Bottom Field	0x02
TG_FIELD_TOP_HDMI_L	0xF030_1088	R/W	HDMI Top Field Start Position	0x01
TG_FIELD_TOP_HDMI_H	0xF030_108C	R/W	HDMI Top Field Start Position	0x00
TG_FIELD_BOT_HDMI_L	0xF030_1090	R/W	HDMI Bottom Field Start Position	0x33
TG_FIELD_BOT_HDMI_H	0xF030_1094	R/W	HDMI Bottom Field Start Position	0x02
SPDIFIN_CLK_CTRL	0xF030_5000	R/W	SPDIFIN_CLK_CTRL [1:0]	0x02
SPDIFIN_OP_CTRL	0xF030_5004	R/W	SPDIFIN_OP_CTRL [1:0]	0x00
SPDIFIN_IRQ_MASK	0xF030_5008	R/W	SPDIFIN_IRQ_MASK[7:0]	0x00
SPDIFIN_IRQ_STATUS	0xF030_500C	R/W	SPDIFIN_IRQ_STATUS [7:0]	0x00
SPDIFIN_CONFIG_1	0xF030_5010	R/W	SPDIFIN_CONFIG [7:0]	0x00
SPDIFIN_CONFIG_2	0xF030_5014	R/W	SPDIFIN_CONFIG [11:8]	0x00
SPDIFIN_USER_VALUE_1	0xF030_5020	R/W	SPDIFIN_USER_VALUE [7:0]	0x00
SPDIFIN_USER_VALUE_2	0xF030_5024	R/W	SPDIFIN_USER_VALUE [15:8]	0x00
SPDIFIN_USER_VALUE_3	0xF030_5028	R/W	SPDIFIN_USER_VALUE [23:16]	0x00
SPDIFIN_USER_VALUE_4	0xF030_502C	R/W	SPDIFIN_USER_VALUE [31:24]	0x00
SPDIFIN_CH_STATUS_0_1	0xF030_5030	R/W	SPDIFIN_CH_STATUS_0 [7:0]	0x00
SPDIFIN_CH_STATUS_0_2	0xF030_5034	R/W	SPDIFIN_CH_STATUS_0 [15:8]	0x00
SPDIFIN_CH_STATUS_0_3	0xF030_5038	R/W	SPDIFIN_CH_STATUS_0 [23:16]	0x00
SPDIFIN_CH_STATUS_0_4	0xF030_503C	R/W	SPDIFIN_CH_STATUS_0 [31:24]	0x00
SPDIFIN_CH_STATUS_1	0xF030_5040	R/W	SPDIFIN_CH_STATUS_1	0x00
SPDIFIN_FRAME_PERIOD_1	0xF030_5048	R/W	SPDIF_FRAME_PERIOD [7:0]	0x00
SPDIFIN_FRAME_PERIOD_2	0xF030_504C	R/W	SPDIF_FRAME_PERIOD [15:8]	0x00
SPDIFIN_Pc_INFO_1	0xF030_5050	R/W	SPDIFIN_Pc_INFO [7:0]	0x00
SPDIFIN_Pc_INFO_2	0xF030_5054	R/W	SPDIFIN_Pc_INFO [15:8]	0x00
SPDIFIN_Pd_INFO_1	0xF030_5058	R/W	SPDIFIN_Pd_INFO [7:0]	0x00
SPDIFIN_Pd_INFO_2	0xF030_505C	R/W	SPDIFIN_Pd_INFO [15:8]	0x00
SPDIFIN_DATA_BUF_0_1	0xF030_5060	R/W	SPDIFIN_DATA_BUF_0 [7:0]	0x00
SPDIFIN_DATA_BUF_0_2	0xF030_5064	R/W	SPDIFIN_DATA_BUF_0 [15:8]	0x00
SPDIFIN_DATA_BUF_0_3	0xF030_5068	R/W	SPDIFIN_DATA_BUF_0 [23:16]	0x00
SPDIFIN_USER_BUF_0	0xF030_506C	R/W	SPDIFIN_DATA_BUF_0 [31:28]	0x00
SPDIFIN_DATA_BUF_1_1	0xF030_5070	R/W	SPDIFIN_DATA_BUF_1 [7:0]	0x00
SPDIFIN_DATA_BUF_1_2	0xF030_5074	R/W	SPDIFIN_DATA_BUF_1 [15:8]	0x00

Register	Address	R/W	Description	Reset Value
SPDIFIN_DATA_BUF_1_3	0xF030_5078	R/W	SPDIFIN_DATA_BUF_1 [23:16]	0x00
SPDIFIN_USER_BUF_1	0xF030_507C	R/W	SPDIFIN_DATA_BUF_1 [31:28]	0x00
HAES_CON	0xF030_6000	R/W	HAES_CON	0x00
HAES_DATA_SIZE _L	0xF030_6020	R/W	HAES_DATA_SIZE_L	0x20
HAES_DATA_SIZE_H	0xF030_6024	R/W	HAES_DATA_SIZE_H	0x01
HAES_DATA	0xF030_6030	W	HAES_DATA	0xXX

3.1 HDMI SYSTEM CONTROL REGISTER 0 (HDMI_CON_0, R/W, ADDRESS = 0XF030_0000)

HDMI_CON_0	Bit	Description	Reset Value
Reserved	[7:6]	Reserved	0
BLUE_SCR_EN	[5]	Blue screen mode control. If this is set, the input video pixels are discarded and BLUESCREEN_0/1/2 register values are transmitted for all video data period.	0
Reserved	[4:3]	Reserved	0
ASP_EN	[2]	Audio sample packet generation control. This bit is only valid if system_en is set. 0 = Discard audio sample 1 = If the audio sample is received, it generates audio sample packet.	0
PWDN_ENB	[1]	TMDS analog block power down mode. If this is set to 0, data cannot be transferred to a receiver. 0 = Power down 1 = Normal operation mode	0
HDMI_EN	[0]	Enable HDMI module in HDMI System. HDMI system has HDMI module, HDCP module, HAES module, SPDIFR module and TG module. To enable and run HDMI system properly, S/W should enable each module. 0 = Disables HDMI module 1 = Enables HDMI module **NOTE**: Before enabling HDMI IP, VP & MIXER should be enabled.	0

3.2 HDMI SYSTEM CONTROL REGISTER 1 (HDMI_CON_1, R/W, ADDRESS = 0XF030_0004)

HDMI_CON_1	Bit	Description	Reset Value
Reserved	[7]	Reserved	0
PX_LMT_CTRL	[6:5]	Pixel value limitation control 00 = By-pass (Do not limit the pixel value) 01 = RGB mode The value of G in RGB mode is limited according to HDMI_YMAX and HDMI_YMIN register. The values of B and R in RGB mode are limited according to HDMI_CMAX and HDMI_CMIN register 10 = YCbCr mode All channel's video input pixels are limited according to HDMI_YMAX, HDMI_YMIN, HDMI_CMAX and HDMI_CMIN register 11 = Reserved	0
Reserved	[4:0]	Reserved	0

3.3 HDMI SYSTEM CONTROL REGISTER 2 (HDMI_CON_2, R/W, ADDRESS = 0XF030_0008)

HDMI_CON_2	Bit	Description	Reset Value
Reserved	[7:6]	Reserved	00
VID_PERIOD_EN	[5]	Video preamble control. 0 = Applies Video Preamble. (HDMI mode) 1 = Video Preamble is not applied (DVI mode)	0
Reserved	[4:2]	Reserved	00
DVI_ BAND_EN	[1]	In DVI mode, the leading guard band is not used. 0 = Applies Guard band (HDMI mode) 1 = Guard band is not applied (DVI mode)	0
Reserved	[0]	Reserved	0

3.4 HDMI SYSTEM STATUS FLAG REGISTER (STATUS, R/W, ADDRESS = 0XF030_0010)

STATUS	Bit	Description	Reset Value
AUTHEN_ACK	[7]	If hdcp is authenticated, it occurs. This bit keeps the authentication signal constantly. It is not cleared. This bit is not an interrupt source. 0 = not authenticated, 1= authenticated	0
AUD_FIFO_OVF	[6]	If audio FIFO is overflowed, this bit will be set. Once it is set, it should be cleared by host. 0 = not full, 1= full	0
Reserved	[5]	Reserved	0
UPDATE_RI_INT	[4]	HDCP generates this signal. Refer to HDCP documentation. If it is written by 1, it is cleared. 0 = not occurred, 1= interrupt occurred	0
UPDATE_PJ_INT	[3]	HDCP generates this signal. Refer to HDCP documentation. If it is written by 1, it is cleared. 0 = not occurred, 1= interrupt occurred	0
ExchangeKSV_INT	[2]	HDCP generates this signal. Refer to HDCP documentation. If it is written by 1, it is cleared. 0 = not occurred, 1= interrupt occurred	0
WATCHDOG_INT	[1]	HDCP generates this signal. Refer to HDCP documentation. If it is written by 1, it is cleared. 0 = not occurred, 1= interrupt occurred	0
WtForActiveRx_INT	[0]	HDCP generates this signal. Refer to HDCP documentation. If it is written by 1, it is cleared. 0 = not occurred, 1= interrupt occurred	0

NOTE: Refer to HDCP Transmitter Authentication Protocol State Diagram in HDCP spec.

3.5 HDMI SYSTEM STATUS ENABLE REGISTER (STATUS_EN, R/W, ADDRESS = 0XF030_0020)

STATUS_EN	Bit	Description	Reset Value
Reserved	[7]	Reserved	0
AUD_FIFO_OVF_EN	[6]	Set write enable 0 = Disables, 1 = Enables If it is set to '1', interrupt assertion is written on the STATUS registers.	0
Reserved	[5]	Reserved	0
UPDATE_RI_INT_EN	[4]	Set write enable. 0 = Disables, 1 = Enables	0
UPDATE_PJ_INT_EN	[3]	Set write enable. 0 = Disables, 1 = Enables	0
ExchangeKSV_int_EN	[2]	Set write enable. 0 = Disables, 1 = Enables	0
WATCHDOG_INT_EN	[1]	Set write enable. 0 = Disables, 1 = Enables	0
WtForActiveRx_int_EN	[0]	Set write enable. 0 = Disables, 1 = Enables	0

NOTE: Update_Ri_int_en and Update_Ri_int_en should be enabled after s/w gets ExchangeKSV_int.

3.6 SOFTWARE HOT PLUG DETECTION. (HPD, R/W, ADDRESS = 0XF030_0030)

HPD (Hot Plug Detection)	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
SW_HPD	[1]	SW_HPD signal is used for Hot plug (HDMI/ DVI cable plugging) detection. If S/W detects Hot plug detection, S/W should set this field. 0 = Low (unplugged), 1 = High (plugged)	0
Reserved	[0]	Reserved	0

NOTE: Refer to HDCP Transmitter Link State Diagram in HDCP specification.

3.7 HDMI/DVI MODE SELECTION FOR HDCP. AFTER MODE DECISION, ONE OF TWO BITS HAS TO BE SET. MODE_SEL (MODE_SEL, R/W, ADDRESS = 0XF030_0040)

MODE_SEL	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
HDMI_MODE	[1]	Select a mode. 0 = Disables, 1 = Enables	0
DVI_MODE	[0]	Select a mode. 0 = Disables, 1 = Enables	0

* DVI mode gets the higher priority.

3.8 HDCP ENCRYPTION ENABLE REGISTER (ENC_EN, R/W, ADDRESS = 0XF030_0044)

ENC_EN	Bit	Description	Reset Value
Reserved	[7:1]	Reserved	0
HDCP_ENC_EN	[0]	If set, HDCP encryption is applied. Before this bit is set, the HDCP authentication process has to be accomplished. 0 = Disables Encryption, 1 = Enables	0

* If HDCP_ENC_EN is low, raw data is bypassed. Therefore in this case we have to send blue screen, or some pictures irrelavent to real contents.

3.9 BLUE_SCREEN_0 (BLUE_SCREEN_0, R/W, ADDRESS = 0XF030_0050)

BLUE_SCREEN	Bit	Description	Reset Value
BLUESCREEN_0	[7:0]	Channel 0 color setting (Cb value or B value).	0

3.10 BLUE_SCREEN_1 (BLUE_SCREEN_0, R/W, ADDRESS = 0XF030_0054)

BLUE_SCREEN	Bit	Description	Reset Value
BLUESCREEN_1	[7:0]	Channel 1 color setting (Y value or G value).	0

3.11 BLUE_SCREEN_2 (BLUE_SCREEN_0, R/W, ADDRESS = 0XF030_0058)

BLUE_SCREEN	Bit	Description	Reset Value
BLUESCREEN_2	[7:0]	Channel 2 color setting (Cr value or R value).	0

3.12 PIXEL LIMIT VALUE REGISTERS (HDMI_YMAX, R/W, ADDRESS = 0XF030_0060)

HDMI_YMAX	Bit	Description	Reset Value
HDMI_YMAX	[7:0]	These registers are used according to PX_LMT_CTRL bits in HDMI_CON_1 register. For Y values, if (input_color_value > HDMI_YMAX x 16) output_color_value = HDMI_YMAX x 16 else if (input_color_value < HDMI_YMIN x 16) output_color_value = HDMI_YMIN x 16 else output_color_value = input_color_value **NOTE**: Value 16 at each line is to compensate the difference of bit width between the input pixels and register value.	0xEB

3.13 PIX PIXEL LIMIT VALUE REGISTERS (HDMI_YMIN, R/W, ADDRESS = 0XF030_0064)

HDMI_YMIN	Bit	Description	Reset Value
HDMI_YMIN	[7:0]	These registers are used according to PX_LMT_CTRL bits in HDMI_CON_1 register. For Y values, if (input_color_value > HDMI_YMAX x 16) output_color_value = HDMI_YMAX x 16 else if (input_color_value < HDMI_YMIN x 16) output_color_value = HDMI_YMIN x 16 else output_color_value = input_color_value **NOTE**: Value 16 at each line is to compensate the difference of bit width between the input pixels and register value.	0x10

3.14 PIXEL LIMIT VALUE REGISTERS (HDMI_CMAX, R/W, ADDRESS = 0XF030_0068)

HDMI_CMAX	Bits	Description	Reset Value
HDMI_CMAX	[7:0]	These registers are used according to PX_LMT_CTRL bits in HDMI_CON_1 register. For Cb and Cr values, if (input_color_value > HDMI_CMAX x 16) output_color_value = HDMI_CMAX x 16 else if (input_color_value < HDMI_CMIN x 16) output_color_value = HDMI_CMIN x 16 else output_color_value = input_color_value **NOTE**: Value 16 at each line is to compensate the difference of bit width between the input pixels and register value.	0xF0

3.15 PIXEL LIMIT VALUE REGISTERS (HDMI_CMIN, R/W, ADDRESS = 0XF030_006C)

HDMI_CMIN	Bits	Description	Reset Value
HDMI_CMIN	[7:0]	These registers are used according to PX_LMT_CTRL bits in HDMI_CON_1 register. For Cb and Cr values, if (input_color_value > HDMI_CMAX x 16) output_color_value = HDMI_CMAX x 16 else if (input_color_value < HDMI_CMIN x 16) output_color_value = HDMI_CMIN x 16 else output_color_value = input_color_value **NOTE**: Value 16 at each line is to compensate the difference of bit width between the input pixels and register value.	0x10

3.16 PACKET TRANSMISSION START MARGIN DURING VBI (VBI_ST_MG, R/W, ADDRESS = 0XF030_0080)

VBI_ST_MG	Bits	Description	Reset Value
VBI_ST_MG	7:0	The number of cycles for preventing transmission of any packet data from the start of the video line during VBI period. **NOTE**: Strongly recommend that do NOT modify this field	0x3C

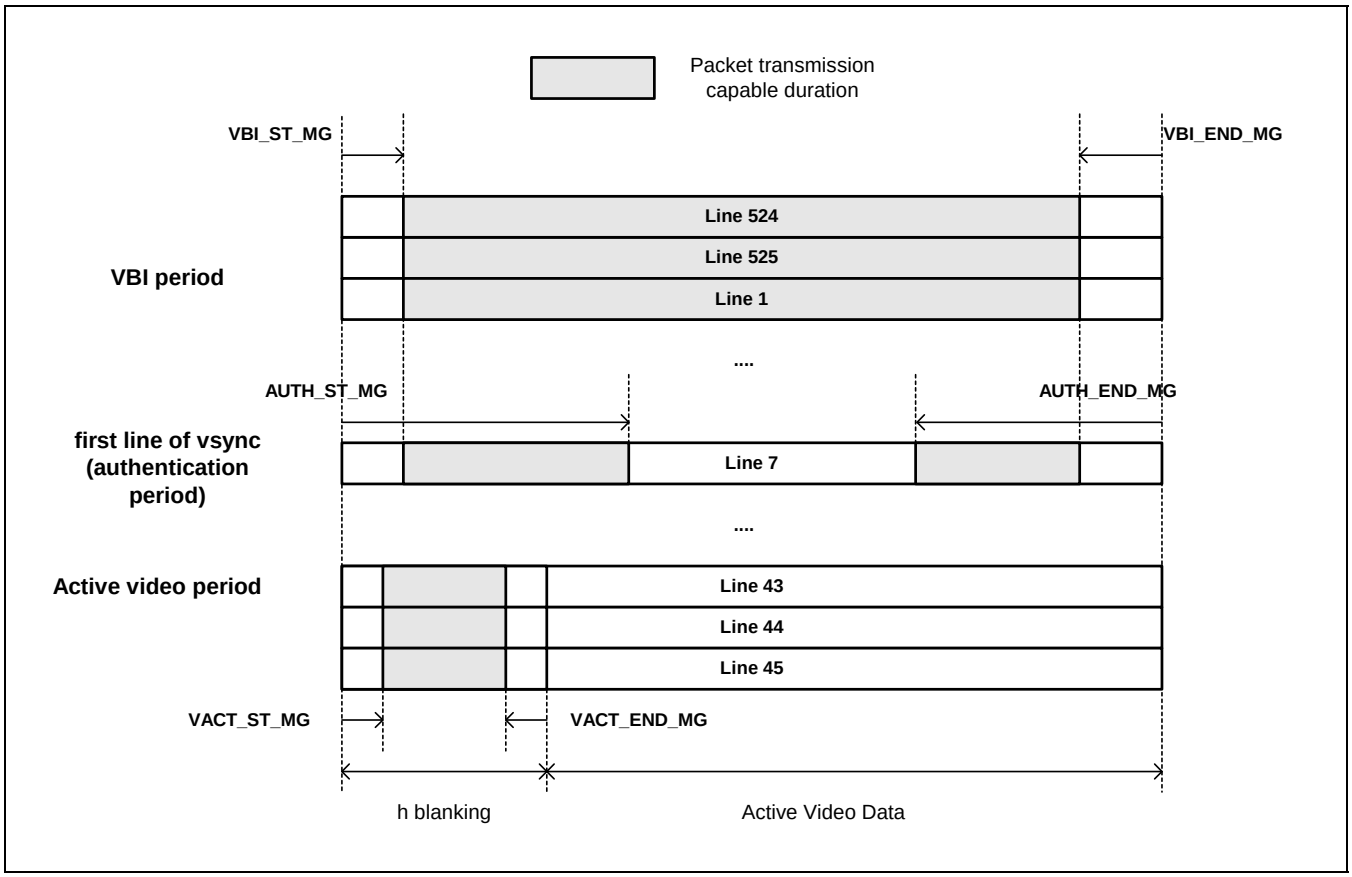


Figure 9.10-8 Packet Transmission Timing Setting (480p @60Hz Case)

The above figure shows the 480p @ 59.94/60Hz timing diagram for packet transmission capable duration.

There are 6 parameters, VBI_ST_MG, VBI_END_MG, VACT_ST_MG, VACT_END_MG, AUTH_ST_MG and AUTH_END_MG.

3.17 PACKET TRANSMISSION END MARGIN DURING VBI (VBI_END_MG, R/W, ADDRESS = 0XF030_0084)

VBI_END_MG	Bits	Description	Reset Value
VBI_END_MG	7:0	The number of cycles for preventing transmission of any packet data from the end of the video line during VBI period. **NOTE**: Strongly recommend that do NOT modify this field	0x30

3.18 PACKET TRANSMISSION START MARGIN DURING ACTIVE VIDEO (VACT_ST_MG, R/W, ADDRESS = 0XF030_0088)

VACT_ST_MG	Bits	Description	Reset Value
VACT_ST_MG	7:0	The number of cycles for preventing transmission of any packet data from the start of the video line during active video period (Horizontal Blanking Period). **NOTE**: Strongly recommend that do NOT modify this field	0x2C

3.19 PACKET TRANSMISSION END MARGIN DURING ACTIVE VIDEO (VACT_END_MG, R/W, ADDRESS = 0XF030_008C)

Name	Bits	Description	Reset Value
VACT_END_MG	7:0	The number of cycles for preventing transmission of any packet data from the start of the video line during active video period (Horizontal Blanking Period). **NOTE**: Strongly recommend that do NOT modify this field	0x2E

3.20 CLOCK CYCLES OF HORIZONTAL BLANKING SIZE (H_BLANK_0, R/W, ADDRESS = 0XF030_ 00A0)

H_BLANK_0	Bit	Description	Reset Value
H_BLANK_L	[7:0]	Clock Cycles of horizontal Blanking Size (Lower part).	0x00

3.21 CLOCK CYCLES OF HORIZONTAL BLANKING SIZE (H_BLANK_1, R/W, ADDRESS = 0XF030_ 00A4)

H_BLANK_1	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x00
H_BLANK_H	[2:0]	Clock Cycles of horizontal Blanking Size (Upper part).	000

60Hz	720x480p	1280x720p
H_BLANK_H/L	138(8Ah)	370(172h)
50Hz	**720x576p**	**1280x720p**
H_BLANK_H/L	144(90h)	700(2bch)

3.22 V_BLANK SIZE THAT DEPENDS ON RESOLUTION (V_BLANK_0, R/W, ADDRESS = 0XF030_ 00B0)

V_BLANK_0	Bit	Description	Reset Value
V2_BLANK_L	[7:0]	Number of line size for end part of vertical blanking region (V1_BLANK+Active Lines) (Lower part).	0

3.23 V_BLANK SIZE THAT DEPENDS ON RESOLUTION (V_BLANK_1, R/W, ADDRESS = 0XF030_ 00B4)

V_BLANK_1	Bit	Description	Reset Value
V1_BLANK_L	[7:3]	Number of line size for front part of vertical blanking region (Lower part).	0
V2_BLANK_H	[2:0]	Number of line size for end part of vertical blanking region (V1_BLANK+Active Lines) (Upper part).	0

60Hz	720x480p	1280x720p
V2_BLANK_H/L	525(d)	750(d)
50Hz	**720x576p**	**1280x720p**
V2_BLANK_H/L	625(d)	750(d)

3.24 V_BLANK SIZE THAT DEPENDS ON RESOLUTION (V_BLANK_2, R/W, ADDRESS = 0XF030_ 00B8)

V_BLANK_2	Bit	Description	Reset Value
Reserved	[7:6]	Reserved	0
V1_BLANK_H	[5:0]	Number of line size for front part of vertical blanking region(Upper part).	0

60Hz	720x480p	1280x720p
V1_BLANK_H/L	45(d)	30(d)
50Hz	**720x576p**	**1280x720p**
V1_BLANK_H/L	49(d)	30(d)

3.25 H_LINE & V_LINE: REAL RESOLUTION. IT DEPENDS ON RESOLUTIONS (H_V_LINE_0, R/W, ADDRESS = 0XF030_00C0)

H_V_LINE_0	Bit	Description	Reset Value
V_LINE_L	[7:0]	Vertical Line Length. (Lower part)	0

3.26 H_LINE & V_LINE: REAL RESOLUTION. IT DEPENDS ON RESOLUTIONS (H_V_LINE_1, R/W, ADDRESS = 0XF030_00C4)

H_V_LINE_1	Bit	Description	Reset Value
H_LINE_L	[7:4]	Horizontal Line Length. (Lower part)	0
V_LINE_H	[3:0]	Vertical Line Length. (Upper part)	0

60Hz	720x480p	1280x720p
V_LINE_H/L	525(d)	750(d)
50Hz	**720x576p**	**1280x720p**
V_LINE_H/L	625(d)	750(d)

3.27 H_LINE & V_LINE: REAL RESOLUTION. IT DEPENDS ON RESOLUTIONS (H_V_LINE_2, R/W, ADDRESS = 0XF030_00C8)

H_V_LINE_2	Bit	Description	Reset Value
H_LINE_H	[7:0]	Vertical Line Length. (Upper part)	0

60Hz	720x480p	1280x720p
H_LINE_H/L	858(d)	1650(d)
50Hz	**720x576p**	**1280x720p**
H_LINE_H/L	864(d)	1980(d)

3.28 VERTICAL SYNC POLARITY CONTROL REGISTER (SYNC_MODE, R/W, ADDRESS = 0XF030_ 00E4)

SYNC_MODE	Bit	Description	Reset Value
Reserved	[7:1]	Reserved	0
V_SYNC_POL_SEL	[0]	For Start Point Detection. 720p's sync shapes are different from 480p and 576p's. They are inverted shapes. If v_sync is inverted, then hard to detect start point, therefore it is needed in this design. 0 = Active High, 1 = Active Low	0

50/60 Hz	720x480p	720x576p	1280x720p
VSYNC_POL	1	1	0

3.29 INTERLACE OR PROGRESSIVE MODE SELECTION (INT_PRO_MODE, R/W, ADDRESS = 0XF030_00E8)

INT_PRO_MODE	Bit	Description	Reset Value
Reserved	[7:1]	Reserved	0
INT_PRO_MODE	[0]	Interlace or Progressive Mode Selection 0 = Progressive 1 = Interlace.	0

3.30 THE START POSITION OF HDCP ENABLE (SEND_START_0, R/W, ADDRESS = 0XF030_ 00F0)

SEND_START_0	Bit	Description	Reset Value
SEND_START_0	[7:0]	The start positon of HDCP enable (Lower part) Before getting an authentication we have not to send data for 16cycles from 512 to 528 (rising to rising)	0x000

3.31 THE START POSITION OF HDCP ENABLE (SEND_START_1, R/W, ADDRESS = 0XF030_ 00F4)

SEND_START_1	Bit	Description	Reset Value
SEND_START_1	[7:0]	The start positon of HDCP enable (Upper part) Before getting an authentication we have not to send data for 16cycles from 512 to 528 (rising to rising) **NOTE**: Recommend value of SEND_START_0/1 is 513d	0x000

3.32 THE END POSITION OF NOT SEND PERIOD (SEND_END_0, R/W, ADDRESS = 0XF030_ 0100)

SEND_END_0	Bit	Description	Reset Value
SEND_END_PD_L	[7:0]	The end positon of not send period (Lower part) Don't send data period with SEND_START, After this period hdmi control block should apply to HDCP block an encryption enable signal with one clock duty from setting SEND_END_ENC_H/L	0x0FF

3.33 THE END POSITION OF NOT SEND PERIOD / THE END POSITION OF HDCP ENC_EN SIGNAL (SEND_END_1, R/W, ADDRESS = 0XF030_ 0104)

SEND_END_1	Bit	Description	Reset Value
SEND_END_ENC_L	[7:2]	The end position of HDCP ENC_EN signal (Lower part)	0x3F
SEND_END_PD_H	[1:0]	The end position of not send period (Upper part) **NOTE**: Recommend value of SEND_END_PD_H/L is 529d	0x3

3.34 THE END POSITION OF HDCP ENC_EN SIGNAL (SEND_END_2, R/W, ADDRESS = 0XF030_ 0108)

SEND_END_2	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0x0

SEND_END_ENC_H	[3:0]	The end position of HDCP ENC_EN signal (Upper part) **NOTE**: Recommend value of SEND_END_ENC_H/L is 530d	0xF

3.35 THE START(END) POSITION OF BOTTOM FIELD'S BLANK REGION (V_BLANK_F_0, R/W, ADDRESS = 0XF030_ 0110)

V_BLANK_F_0	Bit	Description	Reset Value
V_BOT_ST_L	[7:0]	The start position of bottom field's blank region (Lower part).	0x000

* This register has nothing to do with progressive mode. It affects just interlace mode.

3.36 THE START(END) POSITION OF BOTTOM FIELD'S BLANK REGION (V_BLANK_F_1, R/W, ADDRESS = 0XF030_ 0114)

V_BLANK_F_1	Bit	Description	Reset Value
V_BOT_END_L	[7:3]	In the interlace mode, vertical blank length of even field and odd field is different. This register specifies the end position of bottom field's blank region (Lower part).	0x000
V_BOT_ST_H	[2:0]	The start position of bottom field's blank region (Upper part).	0x000

* This register has nothing to do with progressive mode. It affects just interlace mode.

50/60 Hz	720x480p	720x576p	1280x720p
V_BOT_ST_H/L	Don't care	Don't care	Don't care

3.37 THE START(END) POSITION OF BOTTOM FIELD'S BLANK REGION (V_BLANK_F_2, R/W, ADDRESS = 0XF030_ 0118)

V_BLANK_F_2	Bit	Description	Reset Value
Reserved	[7:6]	Reserved	0
V_BOT_END_H	[5:0]	In the interlace mode, vertical blank length of even field and odd field is different. This register specifies the end position of bottom field's blank region (Upper part).	0x000

* This register has nothing to do with progressive mode. It affects just interlace mode.

50/60 Hz	720x480p	720x576p	1280x720p
V_BOT_END_H/L	Don't care	Don't care	Don't care

3.38 HDMI_H_SYNC_GEN. (H_SYNC_GEN_0, R/W, ADDRESS = 0XF030_0120)

H_SYNC_GEN_0	Bit	Description	Reset Value
HSYNC_START_L	[7:0]	Set the start point of H sync (Lower part).	0x000

3.39 HDMI_H_SYNC_GEN. (H_SYNC_GEN_1, R/W, ADDRESS = 0XF030_0124)

H_SYNC_GEN_1	Bit	Description	Reset Value
HSYNC_END_L	[7:2]	Set the end point of H sync (Lower part).	0x000
HSYNC_START_H	[1:0]	Set the start point of H sync (Upper part).	0x000

60Hz	720x480p	1280x720p
HSYNC_START_H/L	14(d)	108(d)
50Hz	**720x576p**	**1280x720p**
HSYNC_START_H/L	10(d)	438(d)

3.40 HDMI_H_SYNC_GEN. (H_SYNC_GEN_2, R/W, ADDRESS = 0XF030_0128)

H_SYNC_GEN_2	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	0x0
HSYNC_POL	[4]	Set this bit for inverting the generated signal to meet the modes. Do not invert the signals in 720p modes. Signals must be inverted for other modes. 0 = Active High, 1 = Active Low	0x0
HSYNC_END_H	[3:0]	Set the end point of H sync (Upper part).	0x000

60Hz	720x480p	1280x720p
HSYNC_END_H/L HSYNC_POL	76(d) 1	148(d) 0
50Hz	**720x576p**	**1280x720p**
HSYNC_END_H/L HSYNC_POL	74(d) 1	478(d) 0

3.41 HDMI_V_SYNC_GEN_1.(V_SYNC_GEN_1_0, R/W, ADDRESS = 0XF030_0130)

V_SYNC_GEN_1_0	Bit	Description	Reset Value
VSYNC_T_END_L	[7:0]	Top field V sync end line number for interlaced mode or frame V sync end line number for progressive mode (Lower part)	0x01

3.42 HDMI_V_SYNC_GEN_1.(V_SYNC_GEN_1_1, R/W, ADDRESS = 0XF030_0134)

V_SYNC_GEN_1_1	Bit	Description	Reset Value
VSYNC_T_ST_L	[7:4]	Top field V sync start line number for interlaced mode or frame V sync start line number for progressive mode (Upper part)	0x1
VSYNC_T_END_H	[3:0]	Top field V sync end line number for interlaced mode or frame V sync end line number for progressive mode (Upper part)	0x0

50/60 Hz	720x480p	720x576p	1280x720p
VSYNC_T_END_H/L	15(d)	10(d)	10(d)

3.43 HDMI_V_SYNC_GEN_1.(V_SYNC_GEN_1_2, R/W, ADDRESS = 0XF030_0138)

V_SYNC_GEN_1_2	Bit	Description	Reset Value
VSYNC_T_ST_H	[7:0]	Top field V sync start line number for interlaced mode or frame V sync start line number for progressive mode (Upper part)	0x00

50/60 Hz	720x480p	720x576p	1280x720p
VSYNC_T_ST_H/L	9(d)	5(d)	5(d)

3.44 RESERVED (R/W, ADDRESS = 0XF030_ 0140 ~ 0XF030_0158)

3.45 AUDIO SAMPLE PACKET GENERATION CONTROL REGISTER (ASP_CON, R/W, ADDRESS = 0XF030_0160)

ASP_CON	Bit	Description	Reset Value
Reserved	[7:6]	Reserved	0
SACD_EN	[5]	SACD DSD stream (one bit audio sample) mode enable bit. 0 = L-PCM mode 1 = SACD DSD mode	0
AUD_MODE	[4]	Two channel or multi-channel mode selection This bit also is also used for layout bit in ASP header. 0 = 2 channel mode 1 = Multi channel mode	0
SP_PRE	[3:0]	Control sub-packet usage for multi channel mode only. If two-channel mode, this register value is not used.	0000

3.46 SP_FLAT OR SAMPLE_INVALID VALUE FOR ASP HEADER (ASP_SP_FLAT, R/W, ADDRESS = 0XF030_ 0164)

ASP_SP_FLAT	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0
SP_FLAT	[3:0]	The sp_flat/sample_invalid value in the ASP header. Refer to the HDMI specification v1.2 (5.3.4 and 5.3.9).	0x00

3.47 AUDIO CHANNEL CONFIGURATION REGISTER (ASP_CHCFG)

- ASP_CHCFG0, R/W, Address = 0xF030_ 0170
- ASP_CHCFG1, R/W, Address = 0xF030_ 0174
- ASP_CHCFG2, R/W, Address = 0xF030_ 0178
- ASP_CHCFG3, R/W, Address = 0xF030_ 017C

ASP_CHCFG0, ASP_CHCFG1 ASP_CHCFG2 ASP_CHCFG3	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	0
SPK3R_SEL	[29:27]	Audio channel Selection for subpacket 3 right channel data in multi channel mode. 000 = i_pcm0L is used for sub packet 0 Left channel 001 = i_pcm0R is used for sub packet 0 Left channel 010 = i_pcm1L is used for sub packet 0 Left channel 011 = i_pcm1R is used for sub packet 0 Left channel 100 = i_pcm2L is used for sub packet 0 Left channel 101 = i_pcm2R is used for sub packet 0 Left channel 110 = i_pcm3L is used for sub packet 0 Left channel 111 = i_pcm3R is used for sub packet 0 Left channel	111
SPK3L_SEL	[26:24]	Audio channel Selection for subpacket 3 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	110
Reserved	[23:22]	Reserved	0
SPK2R_SEL	[21:19]	Audio channel Selection for subpacket 2 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	101
SPK2L_SEL	[18:16]	Audio channel Selection for subpacket 2 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	100
Reserved	[15:14]	Reserved	0
SPK1R_SEL	[13:11]	Audio channel Selection for subpacket 1 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	011
SPK1L_SEL	[10:8]	Audio channel Selection for subpacket 1 left channel data in multi channel mode. The meaning is the same as SPK3R_SEL	010
Reserved	[7:6]	Reserved	0
SPK0R_SEL	[5:3]	Audio channel Selection for subpacket 0 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	001
SPK0L_SEL	[2:0]	Audio channel Selection for subpacket 0 right channel data in multi channel mode. The meaning is the same as SPK3R_SEL	000

3.48 ACR PACKET TRANSMISSION CONTROL REGISTER (ACR_CON, R/W, ADDRESS = 0XF030_0180)

ACR_CON	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	0
ALT_CTS_RATE	[4:3]	In some audio format, the CTS value is changed alternately. CTS value 1 = ACR_CTS[19:0] CTS value 2 = {ARC_CTS[19:8], ACR_LSB2} These two values are transmitted alternately at the ratio of this register setting. 00 = always CTS value 1 01 = 1:1 (CTS value 1 = CTS value2) 10 = 2:1 (CTS value 1 = CTS value2) 11 = 3:1 (CTS value 1 = CTS value2) Measured CTS mode, this value is not used.	00
ACR_TX_MODE	[2:0]	000 = No not Tx. 001 = Tx once – Transmit ACR packet once when packet is available at anytime after this value is set. After transmitting, these bits are reset to all zero. 010 = Tx ACR_TXCNT times during every VBI period 011 = Tx by counting video_pixel_clock for a given CTS value in the ACR_CTS0~2 registers. 100 = Measured CTS mode. Make ACR packet with CTS value by counting TMDS clock for Fs x 128 / N duration. In this case, the 7 LSBs of N value (ACR_N register) should be all zero.	00

3.49 ACR PACKET TRANSMISSION CONTROL REGISTER (ACR_MCTS)

- ACR_MCTS0, R, Address = 0xF030_ 0184
- ACR_MCTS1, R, Address = 0xF030_ 0188
- ACR_MCTS2, R, Address = 0xF030_ 018C

ACR_MCTS0 ACR_MCTS1 ACR_MCTS2	Bit	Description	Reset Value
Reserved	[23:20]	Reserved	0
ACR_MCTS	[19:0]	This value is the TMDS clock cycles for N [19:7] number of audio sample inputs. It is valid if measured CTS mode is set on ACR_CON register. Least significant byte first.	0x00001

3.50 CTS VALUE IN ACR PACKET (ACR_CTS)

- ACR_CTS0, R/W, Address = 0xF030_ 0190
- ACR_CTS1, R/W, Address = 0xF030_ 0194
- ACR_CTS2, R/W, Address = 0xF030_ 0198

ACR_CTS0 ACR_CTS1 ACR_CTS2	Bit	Description	Reset Value
Reserved	[23:20]	Reserved	0
ACR_CTS	[19:0]	CTS value for transmission mode other than 'measured CTS' mode. Least significant byte first.	0x0003E8

3.51 THE N VALUE IN ACR PACKET (ACR_N)

- ACR_N0, R/W, Address = 0xF030_01A0
- ACR_N1, R/W, Address = 0xF030_01A4
- ACR_N2, R/W, Address = 0xF030_01A8

ACR_N0 ACR_N1 ACR_N2	Bit	Description	Reset Value
Reserved	[23:20]	Reserved	0
ACR_N	[19:0]	The N value in ACR packet. Least significant byte first.	0x003E8

3.52 SEE ALT_CTS_RATE IN ACR_CON REGISTER (ACR_LSB2, R/W, ADDRESS = 0XF030_01B0)

ACR_LSB2	Bit	Description	Reset Value
ACR_LSB2	[7:0]	Alternate CTS least significant byte. See ALT_CTS_RATE in ACR_CON register.	0x00

3.53 ACR PACKET TRANSMISSION COUNT REGISTER (ACR_TXCNT, R/W, ADDRESS = 0XF030_01B4)

ACR_TXCNT	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	0
ACR_TXCNT	[4:0]	If ACR_TX_MODE is '10', the ACR packet will be transmitted 'ACR_TXCNT + 1' times per every VBI period. ALT_CTS_RATE is also applied. This register is valid if ACR_TX_MODE is '10'.	0x1F

3.54 ACR PACKET TX INTERVAL REGISTER (ACR_TX_INTERVAL, R/W, ADDRESS = 0XF030_01B8)

ACR_TX_INTERVAL	Bit	Description	Reset Value
ACR_TX_INTERVAL	[7:0]	If ACR_TX_MODE is '10', the ACR packet is transmitted ACR_TXCNT times during VBI. This register specifies the number of cycles between each ACR packets. This register is used to avoid continuous transmission more than 18 packets within single DI band. This register is only valid if ACR_TX_MODE is '10'.	0x63

3.55 THE OFFSET FOR MEASURED CTS VALUE (ACR_CTS_OFFSET, R/W, ADDRESS = 0XF030_01BC)

ACR_CTS_OFFSET	Bit	Description	Reset Value
ACR_CTS_OFFSET	[7:0]	If 'measured CTS mode' is used, the CTS value is measured by counting the TMDS clock for a given duration. This value is added to measured CTS value. It is 8-bit signed integer, so subtraction is possible.	0x00

3.56 GCP PACKET TRANSMISSION CONTROL REGISTER (GCP_CON, R/W, ADDRESS = 0XF030_01C0)

GCP_CON	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
GCP_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync. GCP packet is transmitted within 384 cycles after active vsync.	00

3.57 GCP DATA BYTE (GCP_BYTE1, R/W, ADDRESS = 0XF030_ 01D0)

GCP_BYTE1	Bit	Description	Reset Value
GCP_BYTE1	[7:0]	GCP packet's first data byte. It is either 0x10 or 0x01. Refer to Table 5-17 of HDMI v1.2 specification.	0x00

3.58 ACP PACKET TRANSMISSION CONTROL REGISTER (ACP_CON, R/W, ADDRESS = 0XF030_ 01E0)

ACP_CON	Bit	Description	Reset Value
ACP_FR_RATE	[7:3]	Transmit ACP packet once per every ACP_FR_RATE+1 frames (or fields).	0
Reserved	[2]	Reserved	0
ACP_TX_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync with ACP_FR_RATE	00

3.59 ACP PACKET HEADER HB1 REGISTER (ACP_TYPE, R/W, ADDRESS = 0XF030_01E4)

ACP_TYPE	Bit	Description	Reset Value
ACP_TYPE	[7:0]	ACP packet header. (HB1 of ACP packet header) Refer to Table 5-18 in HDMI v1.2 specification	0x00

3.60 ACP PACKET DATA REGISTERS (ACP_DATA00 ~ ACP_DATA16)

- ACP_DATA00, W, Address = 0xF030_0200
- ACP_DATA01, W, Address = 0xF030_0204
- ACP_DATA02, W, Address = 0xF030_0208
- ACP_DATA03, W, Address = 0xF030_020C
- ACP_DATA04, W, Address = 0xF030_0210
- ACP_DATA05, W, Address = 0xF030_0214
- ACP_DATA06, W, Address = 0xF030_0218
- ACP_DATA07, W, Address = 0xF030_021C
- ACP_DATA08, W, Address = 0xF030_0220
- ACP_DATA09, W, Address = 0xF030_0224
- ACP_DATA10, W, Address = 0xF030_0228
- ACP_DATA11, W, Address = 0xF030_022C
- ACP_DATA12, W, Address = 0xF030_0230
- ACP_DATA13, W, Address = 0xF030_0234
- ACP_DATA14, W, Address = 0xF030_0238
- ACP_DATA15, W, Address = 0xF030_023C
- ACP_DATA16, W, Address = 0Xf030_0240

ACP_DATAxx	Bit	Description	Reset Value
ACP_DATA00~16	[7:0]	ACP packet body data (PB0~PB16 of ACP packet body). Refer to 9.3 in HDMI v1.2 specification. Least significant byte first.	0x00

3.61 ISRC PACKET TRANSMISSION CONTROL REGISTER (ISRC_CON, R/W, ADDRESS = 0XF030_ 0250)

ISRC_CON	Bit	Description	Reset Value
ISRC_FR_RATE	[7:3]	Transmit ISRC1 (with ISRC2 or not) packet once per every ISRC_FR_RATE+1 frames (or fields).	0
ISRC2_EN	[2]	Transmit ISRC2 packet with ISRC1 packet	0
ISRC_TX_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync with ISRC_FR_RATE	00

3.62 ISRC PACKET HEADER (ISRC1_HEADER1, R/W, ADDRESS = 0XF030_0264)

ISRC1_HEADER1	Bit	Description	Reset Value
ISRC_Cont	[7]	Refer to Table 5-20 in HDMI v1.2 specification	0
ISRC_Valid	[6]	Refer to Table 5-20 in HDMI v1.2 specification	0
Reserved	[5:3]	Reserved	000
ISRC status	[2:0]	Refer to Table 5-20 in HDMI v1.2 specification	000

3.63 ISRC1 PACKET DATA (ISRC1_DATA00 ~ ISRC1_DATA15)

- ISRC1_DATA00, W, Address = 0xF030_0270
- ISRC1_DATA01, W, Address = 0xF030_0274
- ISRC1_DATA02, W, Address = 0xF030_0278
- ISRC1_DATA03, W, Address = 0xF030_027C
- ISRC1_DATA04, W, Address = 0xF030_0280
- ISRC1_DATA05, W, Address = 0xF030_0284
- ISRC1_DATA06, W, Address = 0xF030_0288
- ISRC1_DATA07, W, Address = 0xF030_028C
- ISRC1_DATA08, W, Address = 0xF030_0290
- ISRC1_DATA09, W, Address = 0xF030_0294
- ISRC1_DATA10, W, Address = 0xF030_0298
- ISRC1_DATA11, W, Address = 0xF030_029C
- ISRC1_DATA12, W, Address = 0xF030_02A0
- ISRC1_DATA13, W, Address = 0xF030_02A4
- ISRC1_DATA14, W, Address = 0xF030_02A8
- ISRC1_DATA15, W, Address = 0xF030_02AC

ISRC1_DATAxx	Bit	Description	Reset Value
ISRC1_DATA00~15	[7:0]	ISRC2 packet body data (PB0~15 of ISRC2 packet body). Refer to Table 5-21 in HDMI v1.2 specification. Least significant byte first.	0x00

3.64 ISRC2 PACKET DATA (ISRC2_DATA00 ~ ISRC2_DATA15)

- ISRC2_DATA00, W, Address = 0xF030_ 02B0
- ISRC2_DATA01, W, Address = 0xF030_ 02B4
- ISRC2_DATA02, W, Address = 0xF030_ 02B8
- ISRC2_DATA03, W, Address = 0xF030_ 02BC
- ISRC2_DATA04, W, Address = 0xF030_ 02C0
- ISRC2_DATA05, W, Address = 0xF030_ 02C4
- ISRC2_DATA06, W, Address = 0xF030_ 02C8
- ISRC2_DATA07, W, Address = 0xF030_ 02CC
- ISRC2_DATA08, W, Address = 0xF030_ 02D0
- ISRC2_DATA09, W, Address = 0xF030_ 02D4
- ISRC2_DATA10, W, Address = 0xF030_ 02D8
- ISRC2_DATA11, W, Address = 0xF030_ 02DC
- ISRC2_DATA12, W, Address = 0xF030_ 02E0
- ISRC2_DATA13, W, Address = 0xF030_ 02E4
- ISRC2_DATA14, W, Address = 0xF030_ 02E8
- ISRC2_DATA15, W, Address = 0XF030_ 02EC

ISRC2_DATAxx	Bit	Description	Reset Value
ISRC2_DATA00~15	[7:0]	ISRC2 packet body data (PB0~15 of ISRC2 packet body). Refer to Table 5-23 in HDMI v1.2 specification. Least significant byte first.	0x00

3.65 AVI INFOFRAME PACKET TRANSMISSION CONTROL REGISTER (AVI_CON, R/W, ADDRESS = 0XF030_ 0300)

AVI_CON	Bit	Description	Reset Value
-	[7:2]	Reserved	0
AVI_TX_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync	00

3.66 AVI INFOFRAME CHECKSUM DATA (AVI_CHECK_SUM, W, ADDRESS = 0XF030_ 0310)

AVI_CHECK_SUM	Bit	Description	Reset Value
AVI_CHECK_SUM	[7:0]	AVI InfoFrame checksum byte. Refer to HDMI v1.2 specification for detailed information.	0x00

3.67 AVI INFOFRAME PACKET DATA (AVI_BYTE01 ~ AVI_BYTE13)

- AVI_BYTE01, W, Address = 0xF030_ 0320
- AVI_BYTE02, W, Address = 0xF030_ 0324
- AVI_BYTE03, W, Address = 0xF030_ 0328
- AVI_BYTE04, W, Address = 0xF030_ 032C
- AVI_BYTE05, W, Address = 0xF030_ 0330
- AVI_BYTE06, W, Address = 0xF030_ 0334
- AVI_BYTE07, W, Address = 0xF030_ 0338
- AVI_BYTE08, W, Address = 0xF030_ 033C
- AVI_BYTE09, W, Address = 0xF030_ 0340
- AVI_BYTE10, W, Address = 0xF030_ 0344
- AVI_BYTE11, W, Address = 0xF030_ 0348
- AVI_BYTE12, W, Address = 0xF030_ 034C
- AVI_BYTE13, W, Address = 0xF030_ 0350

AVI_BYTExx	Bit	Description	Reset Value
AVI_DATA01~ AVI_DATA13	[7:0]	AVI Infoframe packet data registers. Least significant byte first. Refer to HDMI HDMI v1.2 specification for detailed information.	0x00

3.68 AUDIO INFOFRAME (AUI) PACKET TRANSMISSION CONTROL REGISTER (AUI_CON, R/W, ADDRESS = 0XF030_0360)

AUI_TX_CON	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
AUI_TX_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync	00

3.69 AUI PACKET CHECKSUM BYTE (AUI_CHECK_SUM, W, ADDRESS = 0XF030_ 0370)

AUI_CHECK_SUM	Bit	Description	Reset Value
AUI_CHECK_SUM	[7:0]	AUI Infoframe checksum data (PB0 byte of AUI packet body). Refer to HDMI v1.2 specification.	0x00

3.70 AUI INFOFRAME PACKET DATA (AUI_BYTE1 ~ AUI_BYTE4)

- AUI_BYTE1, R/W, Address = 0xF030_0380
- AUI_BYTE2, R/W, Address = 0xF030_0384
- AUI_BYTE3, R/W, Address = 0xF030_0388
- AUI_BYTE4, R/W, Address = 0xF030_038C

AUI_BYTEx	Bit	Description	Reset Value
AUI_BYTE1 ~ AUI_BYTE4	[7:0]	AUI Infoframe packet body (PB1~PB5 bytes of AUI packet body). Refer to HDMI v1.2 specification. Least significant byte first.	0x00

3.71 MPG INFOFRAME TRANSMISSION CONTROL REGISTER (MPG_CON, R/W, ADDRESS = 0XF030_03A0)

MPG_CON	Bit	Description	Reset Value
MPG_TX_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync	00

3.72 MPG INFOFRAME CHECKSUM BYTE (MPG_CHECK_SUM, W, ADDRESS = 0XF030_03B0)

MPG_CHECK_SUM	Bit	Description	Reset Value
MPG_CHECK_SUM	[7:0]	MPG infoframe checksum register (PB0 byte of MPG packet body). Refer to HDMI v1.2 specification.	0x00

3.73 MPG INFOFRAME PACKET BODY (MPG_BYTE1 ~ MPG_BYTE5)

- MPG_BYTE1, W, Address = 0xF030_ 03C0
- MPG_BYTE2, W, Address = 0xF030_ 03C4
- MPG_BYTE3, W, Address = 0xF030_ 03C8
- MPG_BYTE4, W, Address = 0xF030_ 03CC
- MPG_BYTE5, W, Address = 0xF030_ 03D0

MPG_BYTEx	Bit	Description	Reset Value
MPG_DTAT1 ~ MPG_DATA5	[7:0]	MPG Infoframe packet data (PB1~PB5 bytes of MPG packet body). Refer to HDMI v1.2 specification. Least significant byte first.	0x00

3.74 SPD PACKET TRANSMISSION CONTROL REGISTER (SPD_CON, R/W, ADDRESS = 0XF030_ 0400)

SPD_CON	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
SPD_TX_CON	[1:0]	00 = Do not transmit 01 = Transmit once 1x = Transmit every vsync	00

NOTE: SPD related registers are used for Source Product Descriptor (SPD) packet transmission. They consist of full 3 bytes header register and 28 bytes packet body registers, that is, they are used for any format's packet transmission.

3.75 SPD PACKET HEADER (SPD_HEADER)

- SPD_HEADER0, W, Address = 0xF030_ 0410
- SPD_HEADER1, W, Address = 0xF030_ 0414
- SPD_HEADER2, W, Address = 0xF030_ 0418

SPD_HEADER0 SPD_HEADER1 SPD_HEADER2	Bit	Description	Reset Value
SPD_HEADER0 SPD_HEADER1 SPD_HEADER2	[7:0] [7:0] [7:0]	HB0 byte of SPD packet header HB1 byte of SPD packet header HB2 byte of SPD packet header (Refer to HDMI v1.2 Specification.)	0x00 0x00 0x00

3.76 SPD PACKET DATA (SPD_DATA00~27)

- SPD_DATA00, W, Address = 0xF030_0420
- SPD_DATA01, W, Address = 0xF030_0424
- SPD_DATA02, W, Address = 0xF030_0428
- SPD_DATA03, W, Address = 0xF030_042C
- SPD_DATA04, W, Address = 0xF030_0430
- SPD_DATA05, W, Address = 0xF030_0434
- SPD_DATA06, W, Address = 0xF030_0438
- SPD_DATA07, W, Address = 0xF030_043C
- SPD_DATA08, W, Address = 0xF030_0440
- SPD_DATA09, W, Address = 0xF030_0444
- SPD_DATA10, W, Address = 0xF030_0448
- SPD_DATA11, W, Address = 0xF030_044C
- SPD_DATA12, W, Address = 0xF030_0450
- SPD_DATA13, W, Address = 0xF030_0454
- SPD_DATA14, W, Address = 0xF030_0458
- SPD_DATA15, W, Address = 0xF030_045C
- SPD_DATA16, W, Address = 0xF030_0460
- SPD_DATA17, W, Address = 0xF030_0464
- SPD_DATA18, W, Address = 0xF030_0468
- SPD_DATA19, W, Address = 0xF030_046C
- SPD_DATA20, W, Address = 0xF030_0470
- SPD_DATA21, W, Address = 0xF030_0474
- SPD_DATA22, W, Address = 0xF030_0478
- SPD_DATA23, W, Address = 0xF030_047C
- SPD_DATA24, W, Address = 0xF030_0480
- SPD_DATA25, W, Address = 0xF030_0484
- SPD_DATA26, W, Address = 0xF030_0488
- SPD_DATA27, W, Address = 0xF030_048C

SPD_DATAxx	Bit	Description	Reset Value
SPD_DATA00~ SPD_DATA27	[7:0]	SPD packet data registers (PB0~PB27 bytes). Refer to HDMI v1.2 specification. Least significant byte first.	0x00

3.77 COLOR SPACE CONVERSION CONTROL REGISTER (HDMI_CSC_CON, R/W, ADDRESS = 0XF030_0490)

HDMI_CSC_CON	Bit	Description	Reset Value
Reserved	[7:6]	Reserved	0
OUT_OFFSET_SEL	[5:4]	Output offset control. It specifies ooffset0 and ooffset1 values shown in Figure 9.10-9. 00: RGB FR (ooffset0 = 0, ooffset1 = 0) 01 = reserved 10 = RGB LR (ooffset0 = 16, ooffset1 = 16) 11 = YCbCr (ooffset0 = 16, ooffset1 = 128)	11
Reserved	[3]	Reserved	0
IN_CLIP	[2]	0 = Disables clipping. 1 = Enables clipping. If 16 offset is subtracted and the result is negative, then it is clipped to zero.	0
IN_OFFSET_S	[1:0]	Output offset control. It specifies ioffset0 and ioffset1 values shown in Figure 9.10-9. 00 = RGB FR (ioffset0 = 0, ioffset1 =0) 01 = reserved 10 = RGB LR (ioffset0 = 16, ioffset1 = 16) 11= YCbCr (ioffset0=16, ioffset1 = 128)	11

NOTE: The HDMI handles various pixel formats (color space), such as BT.601 YCbCr, BT.709 YCbCr, full range RGB, limited range RGB and so on. The color space conversion process consists of pre-offset management, matrix multiplication, and post-offset management as the following equation.

$$\begin{bmatrix} o\_Y\_G \\ o\_Cb\_B \\ o\_Cr\_R \end{bmatrix} = \begin{bmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{bmatrix} \bullet \left(\begin{bmatrix} i\_Y\_G \\ i\_Cb\_B \\ i\_Cr\_R \end{bmatrix} - \begin{bmatrix} ioffset0 \\ ioffset1 \\ ioffset1 \end{bmatrix} \right) + \begin{bmatrix} ooffset0 \\ ooffset1 \\ ooffset1 \end{bmatrix}$$

Figure 9.10-9 Color Space Conversion Equation

CSC Coefficient Examples

Coefficient	Bypass	YCbCr601 to RGB LR	YCbCr601 to RGB FR	YCbCr601 to RGB LR	YCbCr601 to RGB FR	YCbCr601 to YCbcr709
HDMI_CSC_CON	0x00	0x23	0x03	0x23	0x03	0x33
HDMI_CSC_coef00	256	256	298	256	298	256
HDMI_CSC_coef01	0	-86(=938)	-100(=924)	-46(=978)	-54(=970)	-29 (=995)
HDMI_CSC_coef02	0	-178(=846)	-208(=816)	-117(=907)	-136(=888)	-53 (=971)
HDMI_CSC_coef10	0	256	298	256	298	0
HDMI_CSC_coef11	256	443	516	464	540	260
HDMI_CSC_coef12	0	0	0	0	0	29
HDMI_CSC_coef20	0	256	298	256	298	0
HDMI_CSC_coef21	0	0	0	0	0	19
HDMI_CSC_coef22	256	350	408	394	458	262

Coefficient	RGB FR to RGB LR	RGB FR to YCbCr601	RGB FR to YCbCr709
CSC_CON	0x20	0x30	0x30
HDMI_CSC_coef00	220	129	157
HDMI_CSC_coef01	0	25	16
HDMI_CSC_coef02	0	65	47
HDMI_CSC_coef10	0	-74	-87
HDMI_CSC_coef11	220	112	112
HDMI_CSC_coef12	0	-38	-25
HDMI_CSC_coef20	0	-94	-102
HDMI_CSC_coef21	0	-18	-10
HDMI_CSC_coef22	220	112	112

NOTE: RGB FR stands for RGB Full Range. RGB LR stans for RGB Limited Range.

3.78 COEF00 OF FIGURE 9.10-9 SIGNED 2.8 FIXED-POINT FORMAT (HDMI_CSC_COEF00)

- HDMI_CSC_COEF00L, R/W, Address = 0xF030_ 04A0
- HDMI_CSC_COEF00H, R/W, Address = 0xF030_04A4

HDMI_CSC_COEF00L HDMI_CSC_COEF00H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF00	[9:0]	*coef00* of Figure 9.10-9 Signed 2.8 fixed-point format. Least significant byte first.	0x0100

3.79 COEF01 OF FIGURE 9.10-9. SIGNED 2.8 FIXED-POINT FORMAT (HDMI_CSC_COEF01)

- HDMI_CSC_COEF01L, R/W, Address = 0xF030_04A8
- HDMI_CSC_COEF01H, R/W, Address = 0xF030_04AC

HDMI_CSC_COEF01L HDMI_CSC_COEF01H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF01	[9:0]	*coef01* of Figure 9.10-9. Signed 2.8 fixed-point format. Least significant byte first.	0x0000

3.80 COEF02 OF FIGURE 9.10-9. SIGNED 2.8 FIXED-POINT FORMAT ((HDMI_CSC_COEF02)

- HDMI_CSC_COEF02L, R/W, Address = 0xF030_04B0
- HDMI_CSC_COEF02H, R/W, Address = 0xF030_04B4

HDMI_CSC_COEF02L HDMI_CSC_COEF02H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF02	[9:0]	*coef02* of Figure 9.10-9. Signed 2.8 fixed-point format. Least significant byte first.	0x0000

3.81 COEF10 OF FIGURE 9.10-9. SIGNED 2.8 FIXED-POINT FORMAT (HDMI_CSC_COEF10)

- HDMI_CSC_COEF10L, R/W, Address = 0xF030_04B8
- HDMI_CSC_COEF10H, R/W, Address = 0xF030_04BC

HDMI_CSC_COEF10L HDMI_CSC_COEF10H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF10	[9:0]	*coef10* of Figure 9.10-9. Signed 2.8 fixed-point format. Least significant byte first.	0x0000

3.82 COEF11 OF FIGURE 9.10-9 SIGNED 3.8 FIXED-POINT FORMAT ((HDMI_CSC_COEF11)

- HDMI_CSC_COEF11L, R/W, Address = 0xF030_ 04C0
- HDMI_CSC_COEF11H, R/W, Address = 0xF030_ 04C4

HDMI_CSC_COEF02L HDMI_CSC_COEF02H	Bit	Description	Reset Value
Reserved	[15:11]	Reserved	0
HDMI_CSC_COEF02	[10:0]	*coef11* of Figure 9.10-9 Signed 3.8 fixed-point format. Least significant byte first.	0x0100

3.83 COEF12 OF FIGURE 9.10-9 SIGNED 2.8 FIXED-POINT FORMAT ((HDMI_CSC_COEF12)

- HDMI_CSC_COEF12L, R/W, Address = 0xF030_04C8
- HDMI_CSC_COEF12H, R/W, Address = 0xF030_04CC

HDMI_CSC_COEF12L HDMI_CSC_COEF12H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF12	[9:0]	*coef12* of Figure 9.10-9. Signed 2.8 fixed-point format. Least significant byte first.	0x0000

3.84 COEF20 OF FIGURE 9.10-9 SIGNED 2.8 FIXED-POINT FORMAT (HDMI_CSC_COEF20)

- HDMI_CSC_COEF20L, R/W, Address = 0xF030_ 04D0
- HDMI_CSC_COEF20H, R/W, Address = 0xF030_ 04D4

HDMI_CSC_COEF20L HDMI_CSC_COEF20H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF20	[9:0]	*coef20* of Figure 9.10-9. Signed 2.8 fixed-point format. Least significant byte first.	0x0000

3.85 COEF21 OF FIGURE 9.10-9 SIGNED 2.8 FIXED-POINT FORMAT (HDMI_CSC_COEF21)

- HDMI_CSC_COEF21L, R/W, Address = 0xF030_ 04D8
- HDMI_CSC_COEF21H, R/W, Address = 0xF030_ 04DC

HDMI_CSC_COEF21L HDMI_CSC_COEF21H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF21	[9:0]	*coef21* of Figure 9.10-9 Signed 2.8 fixed-point format. Least significant byte first.	0x0000

3.86 COEF21 OF FIGURE 9.10-9 SIGNED 2.8 FIXED-POINT FORMAT (HDMI_CSC_COEF22)

- HDMI_CSC_COEFF22L, R/W, Address = 0xF030_ 04E0
- HDMI_CSC_COEFF22H, R/W, Address = 0xF030_ 04E4

HDMI_CSC_COEF22L HDMI_CSC_COEF22H	Bit	Description	Reset Value
Reserved	[15:10]	Reserved	0
HDMI_CSC_COEF22	[9:0]	*coef22* of Figure 9.10-9 Signed 2.8 fixed-point format. Least significant byte first.	0x0100

3.87 HDCP_RX_SHA1_00 – HDCP_RX_SHA1_19

- HDCP_RX_SHA1_00, W, Address = 0xF030_ 0600
- HDCP_RX_SHA1_01, W, Address = 0xF030_ 0604
- HDCP_RX_SHA1_02, W, Address = 0xF030_ 0608
- HDCP_RX_SHA1_03, W, Address = 0xF030_ 060C
- HDCP_RX_SHA1_04, W, Address = 0xF030_ 0610
- HDCP_RX_SHA1_05, W, Address = 0xF030_ 0614
- HDCP_RX_SHA1_06, W, Address = 0xF030_ 0618
- HDCP_RX_SHA1_07, W, Address = 0xF030_ 061C
- HDCP_RX_SHA1_08, W, Address = 0xF030_ 0620
- HDCP_RX_SHA1_09, W, Address = 0xF030_ 0624
- HDCP_RX_SHA1_10, W, Address = 0xF030_ 0628
- HDCP_RX_SHA1_11, W, Address = 0xF030_ 062C
- HDCP_RX_SHA1_12, W, Address = 0xF030_ 0630
- HDCP_RX_SHA1_13, W, Address = 0xF030_ 0634
- HDCP_RX_SHA1_14, W, Address = 0xF030_ 0638
- HDCP_RX_SHA1_15, W, Address = 0xF030_ 063C
- HDCP_RX_SHA1_16, W, Address = 0xF030_ 0640
- HDCP_RX_SHA1_17, W, Address = 0xF030_ 0644
- HDCP_RX_SHA1_18, W, Address = 0xF030_ 0648
- HDCP_RX_SHA1_19, W, Address = 0xF030_ 064C

HDCP_RX_SHA1xx	Bit	Description	Reset Value
HDCP_SHA1	[159:0]	160-bit HDCP repeater's SHA-1 value. Least significant byte first.	0x00

3.88 THESE REGISTERS ARE READABLE, BUT THEY ARE NOT MODIFIED BY HDCP H/W (HDCP_RX_KSV_0)

- HDCP_RX_KSV_0_0, R/W, Address = 0xF030_0650
- HDCP_RX_KSV_0_1, R/W, Address = 0xF030_0654
- HDCP_RX_KSV_0_2, R/W, Address = 0xF030_0658
- HDCP_RX_KSV_0_3, R/W, Address = 0xF030_065C
- HDCP_RX_KSV_0_4, R/W, Address = 0xF030_0660

HDCP_RX_KSV_0_x	Bit	Description	Reset Value
HDCP_KSV_LIST	[39:0]	Least significant byte first. One KSV value of the HDCP repeaters's KSV list	0x0000000000

3.89 HDCP KSV LIST CONTROL REGISTER (HDCP_KSV_LIST_CTRL, ADDRESS = 0XF030_ 0664)

HDCP_KSV_LIST_CTRL	Bit	Description	RW	Reset Value
Reserved	[7:3]	Reserved		0
HDCP_KSV_LIST_EMPTY	[2]	If the number of KSV list is zero, set this value to enable SHA-1 module to calculate without KSV list. 0: Not Empty 1: Empty	W	0
HDCP_KSV_END	[1]	It is used to indicate that current KSV value in HDCP_KSV_LIST_X registers is the last one. 0: Not End, 1: End	W	0
HDCP_KSV_READ	[0]	After writing KSV data into HDCP_KSV_LIST_X registers and then writing any value into this register, HDCP SHA-1 module keeps that KSV value into internal buffer and set this flag into '1' to notify that it has been read. 0: Not Read 1: Read	R/W	0

NOTE: When HDCP_KSV_LIST_CTRL[2:1] is read, those value are garbage data

3.90 THE RESULT OF COMPARING THE SHA-1 VALUES (HDCP_SHA_RESULT, R/W, ADDRESS = 0XF030_0670)

HDCP_SHA_RESULT	Bit	Description	Reset Value
Reserved	[7:2]	Reserved	0
HDCP_SHA_VALID_READY	[1]	Indicates that the SHA comparison has been done by the HW. Must be cleared by SW by writing 0	0
HDCP_SHA_VALID	[0]	Indicates that the SHA-1 comparison succeeds. Must be cleared by SW by writing 0	0

3.91 HDCP CONTROL REGISTER (HDCP_CTRL, R/W, ADDRESS = 0XF030_ 0680)

HDCP_CTRL	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	0
EN_PJ	[4]	Enables Pj port update 1 = Enables 0 = Disables	0
Reserved	[3]	Reserved	0
TIMEOUT	[2]	Should be set if Rx is repeater **NOTE**: KSV list is not ready until 5 sec waiting.	0
CP_DESIRED	[1]	HDCP enable 1 = Enables 0 = Disables **NOTE**: To enable HDCP feature, proper s/w sequence is needed. 1) Set HDMI_STATUS_EN, HDMI_HPD, HDCP_OFFSET_TXx & HDCP_CYCLE_AA registers. 2) Set AES_DATA_SIZE_H/L registers and run HAES_CON[0] register. 3) Set HDCP_CTRL registers.	0
ENABLE_1.1_FEATURE	[0]	Advance cipher option enable 1 = Enables 0 = Disables	0

3.92 AFTER SW CHECKS RI OF RX AND THAT OF TX ARE SAME, THE RESULT HAS TO BE WRITTEN TO THIS REGISTER (HDCP_CHECK_RESULT, R/W, ADDRESS = 0XF030_ 0690)

HDCP_Check_Result	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0

Pi_MATCH_RESULT	[3:2]	Write the result of comparison between Ri of Rx and Tx as the following values (Pi : Tx, Pi' : Rx). 0x = don't care 11 = Pj = Pj' 10 = Pi ≠ Pi'	00
Ri_MATCH_RESULT	[1:0]	Write the result of comparison between Ri of Rx and Tx as the following values. (Ri : Tx, Ri' : Rx) 0x : don't care 11 = Ri = Ri' 10 = Ri ≠ Ri'	00

3.93 KEY SELECTION VECTOR (KSV) VALUE FROM RECEIVER (HDCP_BKSV_0-4)

- HDCP_BKSV_0, W, Address = 0xF030_ 06A0
- HDCP_BKSV_1, W, Address = 0xF030_ 06A4
- HDCP_BKSV_2, W, Address = 0xF030_ 06A8
- HDCP_BKSV_3, W, Address = 0xF030_ 06AC
- HDCP_BKSV_4, W, Address = 0xF030_ 06B0

HDCP_BKSV_x	Bit	Description	Reset Value
HDCP_BKSV	[39:0]	Key selection vector (KSV) value from receiver. Least significant byte first.	0x0000000000

3.94 HDCP_AKSV [39:0]. LITTLE ENDIAN ADDRESSING (HDCP_AKSV_0-4)

- HDCP_AKSV_0, R, Address = 0xF030_ 06C0
- HDCP_AKSV_1, R, Address = 0xF030_ 06C4
- HDCP_AKSV_2, R, Address = 0xF030_ 06C8
- HDCP_AKSV_3, R, Address = 0xF030_ 06CC
- HDCP_AKSV_4, R, Address = 0xF030_ 06D0

HDCP_AKSV_x	Bit	Description	Reset Value
HDCP_AKSV	[39:0]	KSV value of transmitter. Least significant byte first.	0x0000000000

3.95 HDCP_AN.LITTLE ENDIAN ADDRESSING

- HDCP_An_0_0, R, Address = 0xF030_ 06E0
- HDCP_An_0_1, R, Address = 0xF030_ 06E4
- HDCP_An_0_2, R, Address = 0xF030_ 06E8
- HDCP_An_0_3, R, Address = 0xF030_ 06EC
- HDCP_An_1_0, R, Address = 0xF030_ 06F0
- HDCP_An_1_1, R, Address = 0xF030_ 06F4
- HDCP_An_1_2, R, Address = 0xF030_ 06F8
- HDCP_An_1_3, R, Address = 0xF030_ 06FC

HDCP_An_x_x	Bit	Description	Reset Value
HDCP_An	[63:0]	64-bit Random number generated by Tx (An). Least significant byte first.	All zeros

3.96 BCAPS INFORMATION FROM RX. THIS VALUE IS THE DATA READ FROM RX (HDCP_BCAPS, R/W, ADDRESS = 0XF030_0700)

HDCP_BCAPS	Bit	Description	Reset Value
Reserved	[7]	Reserved	0
REPEATER	[6]	The receiver supports downstream connections	0
READY	[5]	KSV FIFO, SHA-1 calculation ready	0
FAST	[4]	The receiver devices supports 400KHz transfer	0
Reserved	[3:2]	Must be 0's	0
1.1_FEATURES	[1]	Supports EESS, Advance cipher, and Enhanced link verification 1 = Set, 0 = un-set	0
FAST_REAUTHENTICATION	[0]	ALL HDMI receiver should be capable of reauthentication 1 = Set, 0 = Un-Set	0

3.97 HDCP BSTATUS VALUE FROM RX. THIS IS THE DATA READ FROM RX (HDCP_BSTATUS)

- HDCP_Bstatus_0, R/W, Address = 0xF030_ 0710
- HDCP_Bstatus_1, R/W, Address = 0xF030_ 0714

HDCP_Bstatus_0 HDCP_Bstatus_1	Bit	Description	Reset Value
Reserved	[15:13]	Reserved	0
HDMI_MODE	[12]	HDMI mode indication. If set, HDCP is in HDMI mode.	0
MAX_CASCADE_EXCEEDED	[11]	Topology error	0
DEPTH	[10:8]	Cascade depth	0
MAX_DEVS_EXCEEDED	[7]	Topology error indicator 1 = Error 0 = No Error	0
DEVICE_COUNT	[6:0]	Total number of attached downstream devices	0

3.98 HDCP RI VALUE OF THE TRANSMITTER (HDCP_RI)

- HDCP_Ri_0, R, Address = 0xF030_ 0740
- HDCP_Ri_1, R, Address = 0xF030_ 0744

HDCP_Ri_0 HDCP_Ri_1	Bit	Description	Reset Value
HDCP_Ri	[15:0]	HDCP Ri value of the transmitter. Least significant byte first.	0

3.99 HDCP PJ VALUE OF THE TRANSMITTER (HDCP_PJ, R, ADDRESS = 0XF030_ 0750)

HDCP_Pj	Bit	Description	Reset Value
HDCP_Pj	[7:0]	HDCP Pj value of the transmitter. Least significant byte first.	0

3.100 HDCP MEMORY ADDRESS OFFSET [25:0] (HDCP_OFFSET_TX)

- HDCP_OFFSET_TX_0, R/W, Address = 0xF030_ 0760
- HDCP_OFFSET_TX_1, R/W, Address = 0xF030_ 0764
- HDCP_OFFSET_TX_2, R/W, Address = 0xF030_ 0768
- HDCP_OFFSET_TX_3, R/W, Address = 0xF030_ 076C

HDCP_OFFSET_TX_0 HDCP_OFFSET_TX_1 HDCP_OFFSET_TX_2 HDCP_OFFSET_TX_3	Bits	Description	Reset Value
HDCP_KEY_OFFSET	[26:13]	Private Keys address offset [12:0] (Least significant byte first.) After HAES decrypts key information, valid key information lies on 285 bytes of 288 bytes of decrypted key information. Using HDCP_KEY_OFFSET, start position of KEY in 288 bytes is declared. Recommended value = 0x05	0x00
HDCP_KSV_OFFSET	[12:0]	KSV address offset [12:0] (Least significant byte first.) After HAES decrypts key information, valid key information lies on 285 bytes of 288 bytes of decrypted key information. Using HDCP_KSV_OFFSET, start position of KSV in 288 bytes is declared. Recommended value = 0x00	0x00

NOTE: Usually, first 5 bytes of decrypted key information is for KSV and 280 bytes (56-bits x 40) after KSV is for KEYs.

3.101 HDCP MEMORY READ CYCLE COUNT (HDCP_CYCLE_AA, R/W, ADDRESS = 0XF030_ 0770)

HDCP_CYCLE_AA	Bits	Description	Reset Value
HDCP_CYCLE_AA	[7:0]	Memory read cycle adjustment count value for MEM_decr in HDCP Recommended value is 0x04	0xff

3.102 TG COMMAND REGISTER (TG_CMD, R/W, ADDRESS = 0XF030_1000)

TG_CMD	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	000
getsync_type	[4]	Timing correction enable bit. If this bit is set, the input vsync timing error relative to output vsync is corrected. 0 = Disables 1 = Enables	0
getsync_en	[3]	BT656 input synchronization enable	0
Reserved	[2]	Reserved	0
field_en	[1]	Field mode enable. This should be 0	0
tg_en	[0]	TG global enable bit.	0

3.103 HORIZONTAL FULL SIZE (TG_H_FSZ_L, R/W, ADDRESS = 0XF030_ 1018)

TG_H_FSZ_L	Bit	Description	Reset Value
TG_H_FSZ_L	[7:0]	Horizontal full size (1~8191) (Lower part)	0x72

3.104 HORIZONTAL FULL SIZE (TG_H_FSZ_H, R/W, ADDRESS = 0XF030_ 101C)

TG_H_FSZ_H	Bit	Description	Reset Value
Reserved	[7:5]	Reserved	0x0
TG_H_FSZ_H	[4:0]	Horizontal full size (1~8191) (Upper part)	0x6

3.105 HORIZONTAL ACTIVE START POSITION (TG_HACT_ST_L, R/W, ADDRESS = 0XF030_ 1020)

TG_HACT_ST_L	Bit	Description	Reset Value
TG_HACT_ST_L	[7:0]	Horizontal active start position (1~4095) (Lower part)	0x05

3.106 HORIZONTAL ACTIVE START POSITION (TG_HACT_ST_H, R/W, ADDRESS = 0XF030_ 1024)

TG_HACT_ST_H	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0x0
TG_HACT_ST_H	[3:0]	Horizontal active start position (1~4095) (Upper part)	0x1

3.107 HORIZONTAL ACTIVE SIZE (TG_HACT_SZ_L, R/W, ADDRESS = 0XF030_ 1028)

TG_HACT_SZ_L	Bit	Description	Reset Value
TG_HACT_SZ_L	[7:0]	Horizontal active size (0~4095) (Lower part)	0x00

3.108 HORIZONTAL ACTIVE SIZE (TG_HACT_SZ_H, R/W, ADDRESS = 0XF030_ 102C)

TG_HACT_SZ_H	Bit	Description	Reset Value
Reserved	[7:4]	Reserved	0x0
TG_HACT_SZ_H	[3:0]	Horizontal active size (0~4095) (Upper part)	0x5

3.109 VERTICAL FULL SIZE (TG_V_FSZ_L, R/W, ADDRESS = 0XF030_ 1030)

TG_V_FSZ_L	Bit	Description	Reset Value
TG_V_FSZ_L	[7:0]	Vertical full size (1~2047) (Lower part)	0xEE

3.110 VERTICAL FULL SIZE (TG_V_FSZ_H, R/W, ADDRESS = 0XF030_ 1034)

TG_V_FSZ_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_V_FSZ_H	[2:0]	Vertical full size (1~2047) (Upper part)	0x2

3.111 VSYNC POSITION (TG_VSYNC_L, W, ADDRESS = 0XF030_ 1038)

TG_VSYNC_L	Bit	Description	Reset Value
TG_VSYNC_L	[7:0]	Vertical sync position. If field enable is set, this is top field vsync position. (1~2047) (Lower part)	0x01

3.112 VSYNC POSITION (TG_VSYNC_H, W, ADDRESS = 0XF030_ 103C)

TG_VSYNC_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_VSYNC_H	[2:0]	Vertical sync position. If field enable is set, this is top field vsync position. (1~2047) (Upper part)	0x0

3.113 BOTTOM FIELD VSYNC POSITION (TG_VSYNC2_L, W, ADDRESS = 0XF030_ 1040)

TG_VSYNC2_L	Bit	Description	Reset Value
TG_VSYNC2_L	[7:0]	Vertical sync position for bottom field. (1~2047) (Lower part)	0x33

3.114 BOTTOM FIELD VSYNC POSITION (TG_VSYNC2_H, W, ADDRESS = 0XF030_ 1044)

TG_VSYNC2_H	Bit	Description	Reset Value
Reserved	[7:0]	Reserved	0x0
TG_VSYNC2_H	[2:0]	Vertical sync position for bottom field. (1~2047) (Upper part)	0x2

3.115 VERTICAL ACTIVE START POSITION (TG_VACT_ST_L, R/W, ADDRESS = 0XF030_ 1048)

TG_VACT_ST_L	Bit	Description	Reset Value
TG_VACT_ST_L	[7:0]	Vertical active start position. (1~2047) (Lower part)	0x1a

3.116 VERTICAL ACTIVE START POSITION (TG_TACT_ST_H, R/W, ADDRESS = 0XF030_ 104C)

TG_VACT_ST_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_VACT_ST_H	[2:0]	Vertical active start position. (1~2047) (Upper part)	0x0

3.117 VERTICAL ACTIVE SIZE (TG_VACT_SZ_L, R/W, ADDRESS = 0XF030_ 1050)

TG_VACT_SZ_L	Bit	Description	Reset Value
TG_VACT_SZ_L	[7:0]	Vertical active size. (0~2047) (Lower part)	0xD0

3.118 VERTICAL ACTIVE SIZE (TG_TACT_SZ_H, R/W, ADDRESS = 0XF030_ 1054)

TG_VACT_SZ_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_VACT_SZ_H	[2:0]	Vertical active size. (0~2047) (Upper part)	0x2

3.119 FIELD CHANGE POSITION (TG_FIELD_CHG_L, R/W, ADDRESS = 0XF030_ 1058)

TG_FIELD_CHG_L	Bit	Description	Reset Value
TG_FIELD_CHG_L	[7:0]	Field change position (1~2047) (Lower part)	0x33

3.120 FIELD CHANGE POSITION (TG_FIELD_CHG_H, R/W, ADDRESS = 0XF030_ 105C)

TG_FIELD_CHG_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_FIELD_CHG_H	[2:0]	Field change position (1~2047) (Upper part)	0x2

3.121 BOTTOM FIELD VERTICAL ACTIVE START POSITION (TG_VACT_ST2_L, R/W, ADDRESS = 0XF030_ 1060)

TG_VACT_ST2_L	Bit	Description	Reset Value
TG_VACT_ST2_L	[7:0]	Vertical sync active start position for bottom field (1~2047) (Lower part).	0x48

3.122 BOTTOM FIELD VERTICAL ACTIVE START POSITION (TG_VACT_ST2_H, R/W, ADDRESS = 0XF030_ 1064)

TG_VACT_ST2_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_VACT_ST2_H	[2:0]	Vertical sync active start position for bottom field (1~2047) (Upper part).	0x2

3.123 VSYNC POSITION FOR HDMI (TG_VSYNC_TOP_HDMI_L, R/W, ADDRESS = 0XF030_ 1078)

TG_VSYNC_TOP_HDMI_L	Bit	Description	Reset Value
TG_VSYNC_TOP_HDMI_L	[7:0]	HDMI vsync position for top field (Lower part).	0x01

3.124 VSYNC POSITION FOR HDMI (TG_VSYNC_TOP_HDMI_H, R/W, ADDRESS = 0XF030_ 107C)

TG_VSYNC_TOP_HDMI_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_VSYNC_TOP_HDMI_H	[2:0]	HDMI vsync position for top field (Upper part).	0x0

3.125 BOTTOM FIELD VSYNC POSITION FOR HDMI (TG_VSYNC_BOT_HDMI_L, R/W, ADDRESS = 0XF030_ 1080)

TG_VSYNC_BOT_HDMI_L	Bit	Description	Reset Value
TG_VSYNC_BOT_HDMI_L	[7:0]	HDMI vsync position for bottom field (Lower part).	0x01

3.126 BOTTOM FIELD VSYNC POSITION FOR HDMI (TG_VSYNC_BOT_HDMI_H, R/W, ADDRESS = 0XF030_ 1084)

TG_VSYNC_BOT_HDMI_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_VSYNC_BOT_HDMI_H	[2:0]	HDMI vsync position for bottom field (Upper part).	0x0

3.127 TOP FIELD CHANGE POSITION FOR HDMI (TG_FIELD_TOP_HDMI_L, R/W, ADDRESS = 0XF030_ 1088)

TG_FIELD_TOP_HDMI_L	Bit	Description	Reset Value
TG_FIELD_TOP_HDMI_L	[7:0]	HDMI top field start position (Lower part).	0x01

3.128 TOP FIELD CHANGE POSITION FOR HDMI (TG_FIELD_TOP_HDMI_H, R/W, ADDRESS = 0XF030_ 108C)

TG_FIELD_TOP_HDMI_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_FIELD_TOP_HDMI_H	[2:0]	HDMI top field start position (Upper part).	0x0

3.129 BOTTOM FIELD CHANGE POSITION FOR HDMI (TG_FIELD_BOT_HDMI_L, R/W, ADDRESS = 0XF030_ 1090)

TG_FIELD_BOT_HDMI_L	Bit	Description	Reset Value
TG_FIELD_BOT_HDMI_L	[7:0]	HDMI bottom field start position (Lower part).	0x33

3.130 BOTTOM FIELD CHANGE POSITION FOR HDMI (TG_FIELD_BOT_HDMI_H, R/W, ADDRESS = 0XF030_ 1094)

TG_FIELD_BOT_HDMI_H	Bit	Description	Reset Value
Reserved	[7:3]	Reserved	0x0
TG_FIELD_BOT_HDMI_H	[2:0]	HDMI bottom field start position (Upper part).	0x2

3.131 SPDIFIN_CLK_CTRL, R/W, ADDRESS = 0XF030_ 5000

SPDIFIN_CLK_CTRL	Bit	Description	Reset Value
ready_clk_down	[1]	0 = Enables Clock 1 = Ready for disabling clock (Default) This bit is read-only bit.	1
power_on	[0]	0 = Disables Clock (default) 1 = Activates Clock	0

3.132 SPDIFIN_OP_CTRL, R/W, ADDRESS = 0XF030_5004

SPDIF_OP_CTRL	Bit	Description	Reset Value
op_ctrl	[1:0]	00b = software reset During a software reset all state machines are set to the idle or init state and all internal registers are set to their initial values. Interrupt status registers are cleared, all other registers that are writable by the system processor keep their values. 01b = status checking mode (run) This command should be asserted after SPDIFIN_CLK_CTRL.power_on is set. SPDIFIN starts clock recovery. If recovery is done, SPDIFIN begins detecting preambles of SPDIF signal format and stream data header, abnormal time signal input, abnormal signal input. It also reports this status via interrupts in SPDIFIN_IRQ_STATUS. 11b = status checking + HDMI operation mode (run with HDMI) "01b" case operations + checking internal buffer overflow + write received data which is either audio sample word of PCM or payload of stream. Transfers Data via HDMI. Others = undefined, do not use Users hould assert 'op_ctrl'=11b after SPDIFIN_IRQ_STATUS. ch_status_recovered_ir was asserted at least once if SPDIFIN_CONFIG. data_type with PCM data mode. Or you should assert 'op_ctrl'=11b after SPDIFIN_IRQ_STATUS.stream_header_detected_ir was asserted at least once if SPDIFIN_CONFIG.data_type with stream data mode. - If SPDIFIN_CONFIG.data_type is stream, HDMI operation will be done only for the duration of burst-payload. When SPDIFIN_CONFIG.data_type is PCM, HDMI operation will start at subframe with preamble of B or M.	00

3.133 SPDIFIN_IRQ_MASK, R/W, ADDRESS = 0XF030_ 5008

SPDIFIN_IRQ_MASK	Bit	Description	Reset Value
buf_overflow_ir_en	7	Mask bit for Interrupt 8 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
abnormal_Pd_ir_en	6	Mask bit for Interrupt 7 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
stream_header_not_detected_ at_righttime_ir_en	5	Mask bit for Interrupt 6 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
stream_header_detected_ir_en	4	Mask bit for Interrupt 5 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
stream_header_not_detected_ir_ en	3	Mask bit for Interrupt 4 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
wrong_preamble_ir_en	2	Mask bit for Interrupt 3 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
ch_status_recovered_ir_en	1	Mask bit for Interrupt 2 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0
wrong_signal_ir_en	0	Mask bit for Interrupt 0 0 = Interrupt generation is disabled. 1 = Interrupt generation is enabled.	0

3.134 SPDIFIN_IRQ_STATUS, R/W, ADDRESS = 0XF030_500C

SPDIFIN_IRQ_STATUS	Bit	Description	Reset Value
buf_overflow_ir	7	SPDIFIN internal buffer(s) (SPDIFIN_DATA_BUF_x) are overflowed because HDMI did not transfer the buffer(s) to memory in time. - This interrupt asserts only if SPDIFIN_OP_CTRL.op_ctrl is set as "011". - If user does not handle this interrupt, SPDIFIN overwrited next subframe data to the internal data buffer (SPDIFIN_DATA_BUF_x) and continue data transfer via HDMI. 0 = No Interrupt 1 = Internal buffer overflow Writing '0' has no effect. Writing '1' clears the interrupt request.	0
abnormal_Pd_ir	6	Pd value used exceeds 1 block size of stream data which is extracted from Pc – Pc, Pd value is dependent on SPDIFIN_CONFIG.PcPd_value_mode. The condition is checked every Pd time if stream mode. - This interrupt assertes if SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - This interrupt assertes if SPDIFIN_CONFIG.data_type is 'stream'. - If user does not handle this interrupt, SPDIFIN will stop transferring stream payload via HDMI then wait for the next stream header detection. 0 = No interrupt 1 = Pd exceeds 1 block size of stream Writing '0' has no effect. Writing '1' clears the interrupt request.	0
stream_header_not_detected_at_righttime_ir *1)*	5	0 = no interrupt 1 = Pa~Pd was not detected at right time Writing '0' has no effect. Writing '1' clears the interrupt request. - This interrupt assertes if SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - This interrupt assertes if SPDIFIN_CONFIG.data_type is 'stream'.	0
stream_header_detected_ir *1)*	4	0 = No interrupt 1 = Detects Stream data header (Pa~Pd) Writing '0' has no effect. Writing '1' clears the interrupt request. - This interrupt assertes if SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - Cases for interrupt Case1 =: Initially after power_on	0

SPDIFIN_IRQ_STATUS	Bit	Description	Reset Value
		Case2 = Next stream header at right time if stream data received with SPDIFIN_CONFIG.data_type set as 'stream mode'. Case3 = Initially detected stream header if stream data received with SPDIFIN_CONFIG.data_type set as 'PCM mode'.	
stream_header_not_detected_ir *1)*	3	0 = No interrupt 1 = Stream data header not detected for 4096 repetition time Writing '0' has no effect. Writing '1' clears the interrupt request. - This interrupt assertes if SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - Cases for interrupt Case1 = Initially after power_on Case2 = SPDIFIN was receiving stream but could not find next stream header for 4096 repetition time since previous stream header Case3 = Cannot find stream header for 4096 repetition time since previous reset of repetition time counter after previous interrupt of 'stream_header_not_detected_ir'.	0
wrong_preamble_ir	2	0 = No interrupt 1 = Detects Preamble but there is a problem with detected time Writing '0' has no effect. Writing '1' clears the interrupt request. - This interrupt assertes if SPDIFIN_OP_CTRL.op_ctrl=001b or 011b. - Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL. op_ctrl=01b - Cases for interrupt Case1 = Deetcts preamble in the middle of a subframe audio sample word time Case2 = Next preamble was not detected at exact time after a subframe duration Case3 = It was time for preamble B(or M or W) to be detected but other preamble was detected at that time	0
ch_status_recovered_ir	1	0 = No interrupt 1 = Recovered channel status Writing '0' has no effect. Writing '1' clears the interrupt request. Detectes preamble of 2 consecutive B-preamble thus recovered 192-bit wide channel status. - Only supports consumer mode, therefore just 36-bits are reconstructed. If you want to see the channel status bits through SPDIFIN_CH_STATUS_x, better read two	0

SPDIFIN_IRQ_STATUS	Bit	Description	Reset Value
		consecutive 'ch_status_recovered_ir' and read that register each time; if these two channel status value are same, you can rely on that value.	
wrong_signal_ir	0	0 = No interrupt 1 = Clock recovery fail Writing '0' has no effect. Writing '1' clears the interrupt request. Cannot recover clock from input because of tolerable range violation (unlock) or because of no signal from outside or because of non-biphase in non-preamble duration - Meaningless until ch_status_recovered_ir is asserted initially after SPDIFIN_OP_CTRL. op_ctrl=01b	0

1) Detection of stream header

- Wait for matching of Pa, Pb; 0xF872, 0x4E1F respectively
- Wait for the repetition time (From decoded Pc value or from user-set Pc in SPDIFIN_USER_VALUE. repetition_time_manual according to SPDIFIN_CONFIG. PcPd_value_mode)
- Check for matching of Pa, Pb on right time.

3.135 SPDIFIN_CONFIG_1, R/W, ADDRESS = 0XF030_ 5010

SPDIFIN_CONFIG_1	Bit	Description	Reset Value
noise_filter_samples	6	Noise filtering is done for over-sampled SPDIF input signal. This operation is performed as follows. If 'noise_filter_samples' is 0, 3 consecutive over-sampled signal are regarded as a high or low if those 3 samples are all high or low respectively. If 1 or 2 samples are low or high respectively for 3 over-sample duration, that noise filtered signal keeps previous value. If 'noise_filter_samples' is 1, 2 consecutive over-sampled signal are regarded as a high or low if those 2 samples are all high or low respectively. This setting are used for reduced over-sampling ratio; recommended over-sampling ratio is 10 (Refer to clk_divisor field in SPDIFIN_CONFIG_2) 0 = Filtering with 3 consecutive samples 1 = Filtering with 2 consecutive samples	0
data_type	5	This setting is used for data transfer via HDMI; see SPDIFIN_DATA_BUF_x. 0 = Linear PCM type data 1 = Nonlinear PCM type data (Stream type)	0
PcPd_value_mode	4	If 0 for automatic setting, Pc and Pd values are chosen by value of Pc, Pd from decoded stream header reported as in SPDIFIN_Px_INFO. If you set this register, the receiver uses SPDIFIN_USER_VALUE[31:16], SPDIFIN_USER_VALUE[15:4] value as Pc and Pd respectively instead of decoded data from stream header as reported in SPDIFIN_Px_INFO. (cf) Burst payload length, whether it is automatically set or manually set, affects the data size to be written in memory via HDMI by dumping the full sub-frame for the last bit for burst payload length. For example, if burst payload length is 257-bit, i.e. (16sub-frame * 16-bit + 1-bit), then HDMI writes data in 17 consecutive sub-frames. 0 = Automatically set 1 = Manually set	0
word_length_ value_mode	3	If 0 for automatic setting, word length value is chosen by value of channel status from decoded SPDIF format as reported in SPDIFIN_CH_STATUS_1. word_length. If you set this register, the receiver uses SPDIFIN_USER_VALUE[3:0] value as word length instead of decoded data from channel status as reported in SPDIFIN_CH_STATUS_1.word_length. 0 = Automatically set 1 = Manually set	0

SPDIFIN_CONFIG_1	Bit	Description	Reset Value
U_V_C_P_report	2	Report for each subframe. Valid only if SPDIFIN_CONFIG. data_align is set for 32-bit mode 0 = Neglects 'user_bit', 'validity_bit', 'channel status' 'parity_bit' of SPDIF format. 1 = Reports 'user_bit', 'validity_bit', 'channel status' 'parity_bit' of SPDIF format	0
HDMI_burst_size	1	0 = 1 burst 1 = 2 burst	0
data_align	0	16-bit: Takes 16-bits from MSB in a subframe of SPDIF format then concatenates two consecutive 16-bit data in one 32-bit register of SPDIFIN_DATA_BUF_x. 32-bit: Data from one subframe with zero padding to MSB part. (ex: 0x00ffffff for 24-bit data) If stream mode, you should set 'word_length_value_mode' as 1 and set SPDIFIN_USER_VALUE.word_length_manual as 3'b000. - These two modes is applied to both modes of SPDIFIN_CONFIG.data_type, i.e. PCM or stream; see also SPDIFIN_DATA_BUF_x. 0 = 16-bit mode 1 = 32-bit mode	0

3.136 SPDIFIN_CONFIG_2, R/W, ADDRESS = 0XF030_ 5014

SPDIFIN_CONFIG_2	Bit	Description	Reset Value
clk_divisor	3:0	SPDIFIN over-samples the SPDIF input signal with internally made clock which is divided from system clock. Recommended over-sampling ratio is 8~10, thus following calculation holds. Recommended SPDIFIN_internal_clock = Sampling Frequency of SPDIF Input Signal * 64-bits * 10 times-over-sampling (ex) 48 kHz * 64-bits * 10 times-over-sampling = 31 Mhz SPDIFIN_internal_clock = system_clock / (clk_divisor + 1) (SPDIFIN_internal_clock $\leq$ 135 Mhz)	0000

3.137 REPETITION TIME (SPDIFIN_USER_VALUE_1, R/W, ADDRESS = 0XF030_5020)

SPDIFIN_USER_VALUE_1	Bit	Description	Reset Value
repetition_time_manual_low	7:4	Repetition time Repetition_time_manual register is 12-bits value. this register is low 4-bits Used for counting one block of stream data; valid if SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: frames (1 frame = 2 subframes) of SPDIF format) The value should be (actual repetition time – 1). For example, if you want to use 1536 as manually set repetition time, you should write 1535.	0000
word_length_manual	3:0	Word length Used as size for transferring data to memory via HDMI; valid if SPDIFIN_CONFIG.word_length_value_mode is set for manual mode [0] is 1, [0] is 0 [3:1] 101 = 24 bits, 20 bits 001 = 23 bits, 19 bits 010 = 22 bits, 18 bits 011 = 21 bits, 17 bits 100 = 20 bits, 16 bits	0000

3.138 REPETITION TIME (SPDIFIN_USER_VALUE_2, R/W, ADDRESS = 0XF030_ 5024)

SPDIFIN_USER_VALUE_2	Bit	Description	Reset Value
repetition_time_manual_high	[7:0]	Repetition_time_manual register is 12-bits value. This register is high 8-bits Used for counting one block of stream data; valid if SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: frames (1 frame = 2 subframes) of SPDIF format) The value should be (actual repetition time – 1). For example, if you want to use 1536 as manually set repetition time, you should write 1535.	0x00

3.139 LENGTH OF STREAM DATA PAYLOAD (SPDIFIN_USER_VALUE_3, R/W, ADDRESS = 0XF030_ 5028)

SPDIFIN_USER_VALUE_3	Bit	Description	Reset Value
burst_payload_length_manual_low	[7:0]	Burst_payload_length_manual register is 16bits value. This register is low 8bits Valid if SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: bits)	0x00

3.140 LENGTH OF STREAM DATA PAYLOAD (SPDIFIN_USER_VALUE_4, R/W, ADDRESS = 0XF030_ 502C)

SPDIFIN_USER_VALUE_4	Bit	Description	Reset Value
burst_payload_length_manual_high	[7:0]	Burst_payload_length_manual register 16-bits value. This register is high 8-bits Valid if SPDIFIN_CONFIG.PcPd_value_mode is set for manual mode. (Unit: bits)	0x00

3.141 HDCP PJ INTERRUPT STATUS (SPDIFIN_CH_STATUS_0_1, R/W, ADDRESS = 0XF030_ 5030)

SPDIFIN_CH_STATUS_0_1	Bit	Description	Reset Value
channel_status_mode	7:6	00 = Mode 0 Others = Reserved	00
emphasis	5:3	000 = Emphasis not indicated 100 = Emphasis – CD type	000
copyright_assertion	2	0 = Copyright 1 = No copyright	0
audio_sample_word	1	0 = Linear PCM 1 = Non-linear PCM	0
channel_status_block	0	0 = Consumer format 1 = Professional format	0

3.142 CATEGORY_CODE (SPDIFIN_CH_STATUS_0_2, R/W, ADDRESS = 0XF030_ 5034)

SPDIFIN_CH_STATUS_0_2	Bit	Description	Reset Value
category_code	[7:0]	Equipment type : [8:15] CD player: 1000_0000 DAT player: 1100_000L DCC player: 1100_001L Mini disc: 1001_001L (L = information about generation status of the material) This register is updated every 192 frames (1 block) of SPDIF format.	0x00

3.143 SPDIFIN_CH_STATUS_0_3, R/W, ADDRESS = 0XF030_5038)

SPDIFIN_CH_STATUS_0_3	Bit	Description	Reset Value
channel_number	7:4	(Bit 20 is LSB) This register is updated every 192 frames (1 block) of SPDIF format.	0000
source_number	3:0	(Bit 16 is LSB) This register is updated every 192 frames (1 block) of SPDIF format.	0000

3.144 SPDIFIN_CH_STATUS_0_4, R/W, ADDRESS = 0XF030_503C

SPDIFIN_CH_STATUS_0_4	Bit	Description	Reset Value
Reserved	7:6	Reserved	00
clock_accuracy	5:4	[5:4] 10 = level I, ±50ppm 00 = level II, ±1000ppm 01 = level III, variable pitch shifted This register is updated every 192 frames (1 block) of SPDIF format.	00
sampling_frequency	3:0	[0,1,2,3] 0010 = 22.05kHz 0000 = 44.1 kHz 0001 = 88.2kHz 0011 = 176.4kHz 0110 = 24kHz 0100 = 48 kHz 0101 = 96kHz 0111 = 192kHz 1100 = 32 kHz This register is updated every 192 frames (1 block) of SPDIF format.	0000

3.145 SPDIFIN_CH_STATUS_1, R/W, ADDRESS = 0XF030_5040

SPDIFIN_CH_STATUS_1	Bit	Description	Reset Value
Reserved	7:4	Reserved	0000
word_length	3:1	(field_size = 1), (field_size = 0) 000 = not indicated 101 = 24 bits, 20 bits 001 = 23 bits, 19 bits 010 = 22 bits, 18 bits 011 = 21 bits, 17 bits 100 = 20 bits, 16 bits This register is updated every 192 frames (1 block) of SPDIF format.	000
Field_size	0	0 = Maximum length 20 bits 1 = Maximum length 24 bits This register is updated every 192 frames (1 block) of SPDIF format.	0

3.146 COUNTING VALUE FOR A FRAME OF SPDIF FORMAT (SPDIFIN_FRAME_PERIOD_1, R/W, ADDRESS = 0XF030_ 5048)

SPDIFIN_FRAME_PERIOD_1	Bit	Description	Reset Value
Rrame_cnt_low	[0]	Counting value for a frame of SPDIF format Frame_cnt register is 16-bits value. This is low 8-bits. The period of a frame (2 sub-frames), thus this registeris updated every 2 sub-frames. It is measured by 'SPDIFIN_internal_clk' made with SPDIFIN_CONFIG.clk_divisor. (Unit: SPDIF_internal_clk Cycles) (Recommended value for locking incoming signals : Over 0x220 (8.5timesx64-bits))	0

3.147 COUNTING VALUE FOR A FRAME OF SPDIF FORMAT (SPDIFIN_FRAME_PERIOD_2, R/W, ADDRESS = 0XF030_ 504C)

SPDIFIN_FRAME_PERIOD_2	Bit	Description	Reset Value
Rrame_cnt_high	[7:0]	Counting value for a frame of SPDIF format Frame_cnt register is 16-bits value. This is high 8-bits. The period of a frame (2 sub-frames), thus this register is updated every 2 sub-frames. It is measured by 'SPDIFIN_internal_clk' made with SPDIFIN_CONFIG.clk_divisor. (Unit: SPDIF_internal_clk Cycles) (Recommended value for locking incoming signals : Over 0x220 (8.5timesx64-bits))	0x00

3.148 SPDIFIN_PC_INFO_1, R/W, ADDRESS = 0XF030_ 5050

SPDIFIN_Pc_INFO_1	Bit	Description	Reset Value
Error_flag	7	0 = Valid burst payload 1 = Vurst payload contains errors	0
Reserved	6:5	Reserved	00
compressed_data_type	4:0	0d = Null data 1d = Dolby AC-3 2d = Reserved 3d = Pause 4d = MPEG-1 layer 1 5d = MPEG-1 layer 2 or 3 or MPEG-2 w/o extension 6d = MPEG-2 w/ extension 7d = Reserved 8d = MPEG-2 layer 1 low sampling freq. 9d = MPEG-2 layer 2 or 3 low sampling freq. 10d = Reserved 11d, 12d, 13d = DTS 14d~31d = Reserved	00000

3.149 DATA TYPE DEPENDENT INFORMATION (SPDIFIN_PC_INFO_2, R/W, ADDRESS = 0XF030_ 5054)

SPDIFIN_Pc_INFO_2	Bit	Description	Reset Value
bit_stream_number	7:5	Bit stream number.	000
data_type_dependent_info	4:0	Data type dependent information.	000000

3.150 PCM OR STREAM DATA FOR 1ST BURST OF HDMI (SPDIFIN_DATA_BUF_0_1, R/W, ADDRESS = 0XF030_ 5060)

SPDIFIN_DATA_BUF_0_1	Bit	Description	Reset Value
Received_data	7:0	PCM or stream data for 1st burst of HDMI If SPDIFIN_CONFIG.data_align is 0 for 16-bit received_data = {data_(N)th, data_(N+1)th} If SPDIFIN_CONFIG.data_align is 1 for 32-bit received_data = {U, V, C, P, zero-padding, data [n: 0]} received_data = {zero-padding, data [n: 0]} (if SPDIFIN_CONFIG.U_V_P_report is 0) ('n' is dependent on SPDIFIN_CH_STATUS_1. word_length when SPDIFIN_CONFIG. data_type is 0 for PCM. Or 'n' is 15 when SPDIFIN_CONFIG.data_type is 1 for stream)	0x00

3.151 PCM OR STREAM DATA FOR 1ST BURST OF HDMI (SPDIFIN_DATA_BUF_0_2, R/W, ADDRESS = 0XF030_ 5064)

SPDIFIN_DATA_BUF_0_2	Bit	Description	Reset Value
Received_data	7:0	PCM or stream data for 1st burst of HDMI	0x00

3.152 PCM OR STREAM DATA FOR 1ST BURST OF HDMI (SPDIFIN_DATA_BUF_0_3, R/W, ADDRESS = 0XF030_ 5068)

SPDIFIN_DATA_BUF_0_3	Bit	Description	Reset Value
Received_data	7:0	PCM or stream data for 1st burst of HDMI	0x00

3.153 PCM OR STREAM DATA FOR 1ST BURST OF HDMI (SPDIFIN_USER_BUF_0, R/W, ADDRESS = 0XF030_ 506C)

SPDIFIN_USER_BUF_0	Bit	Description	Reset Value
Unused	7:4	Unused	0000
received_data	3:0	PCM or stream data for 1st burst of HDMI	0000

3.154 PCM OR STREAM DATA FOR 2ND BURST OF HDMI (SPDIFIN_DATA_BUF_1_1, R/W, ADDRESS = 0XF030_ 5070)

SPDIFIN_DATA_BUF_1_1	Bit	Description	Reset Value
Received_data	7:0	PCM or stream data for 2nd burst of HDMI If SPDIFIN_CONFIG.HDMI_burst_size is 2 bursts, HDMI accesses SPDIFIN_DATA_BUF_0 and SPDIFIN_DATA_BUF_1 together. Endian of each 'data'	0x00

3.155 PCM OR STREAM DATA FOR 2ND BURST OF HDMI (SPDIFIN_DATA_BUF_1_2, R/W, ADDRESS = 0XF030_ 5074)

SPDIFIN_DATA_BUF_1_2	Bit	Description	Reset Value
received_data	7:0	PCM or stream data for 2nd burst of HDMI	0x00

3.156 PCM OR STREAM DATA FOR 2ND BURST OF HDMI (SPDIFIN_DATA_BUF_1_3, R/W, ADDRESS = 0XF030_ 5078)

SPDIFIN_DATA_BUF_1_3	Bit	Description	Reset Value
received_data	7:0	PCM or stream data for 2nd burst of HDMI	0x00

3.157 PCM OR STREAM DATA FOR 2ND BURST OF HDMI (SPDIFIN_USER_BUF_1, R/W, ADDRESS = 0XF030_ 507C)

SPDIFIN_USER_BUF_1	Bit	Description	Reset Value
Unused	7:4	Unused	0000
received_data	3:0	PCM or stream data for 2nd burst of HDMI	0000

3.158 HAES CONTROL (HAES_CON, R/W, ADDRESS = 0XF030_ 6000)

HAES_CON	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	24'h0
HAES_KEY_START	[7]	Start HAES Key generation. 0 = Disables HAES Key generation 1 = Enables HAES key generation. If HAES_KEY_START is set, it takes 128 cycles to generate HAES key. If generation is done, HAES_KEY_DONE is set.	0
HAES_KEY_DONE	[6]	End of HAES Key generation 0 = Generating HAES Key 1 = End of generation of HAES key	0
Reserved	[5:1]	Reserved	0x0
HAES_START	[0]	HAES Start signal If specified amount of data is decrypted and written in memory, then HAES start signal goes to 0 Before HAES_START is set, s/w should ensure that HAES_KEY_DONE is set. 0 = HAES does not decrypt data 1 = HAES starts to decrypt data from MEM_encr to generate HDCP Key.	0

3.159 HAES DATA SIZE (IN BYTES) TO DECRYPT (HAES_DATA_SIZE_L, R/W, ADDRESS = 0XF030_ 6020)

HAES_DATA_SIZE_L	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x00
HAES_DATA_SIZE_L	[7:0]	HAES data size (in bytes) to decrypt If the data size is not the multiple of 128-bits, zeros should be padded 128-bit align Maximum number is 120h (Internal memory size limits the maximum data size) Default value : 288 bytes	0x20

3.160 HAES DATA SIZE (IN BYTES) TO DECRYPT (HAES_DATA_SIZE_H, R/W, ADDRESS = 0XF030_ 6024)

HAES_DATA_SIZE_H	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0x00
HAES_DATA_SIZE_H	[0]	HAES data size (in bytes) to decrypt If the data size is not the multiple of 128 bits, zeros should be padded 128bit align Maximum number is 120h (Internal memory size limits the maximum data size) Default value : 288 bytes	0x01

3.161 ENCRYPTED DATA TO BE WRITTEN IN MEMORY BEFORE DECRYPTION (HAES_DATA, W, ADDRESS = 0XF030_ 6030)

HAES_DATA	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	24'hXX
HAES_DATA	[7:0]	Encrypted data to be written in memory before decryption Memory address is automatically increased. Zeros should be padded for 128-bit align.	0xXX

9.11 MFC (MULTI FORMAT CODEC)

1 OVERVIEW

The Multi Format Video Codec MFC (FIMV-MFC V4.0) is a synthesizable core which can perform encoding and decoding of multiple streams at 30fps of upto 720p resolution image (1280x720). The FIMV-MFC V4.0 is a real-time solution for dual stream video decoding in a single core at moderate clock rates.

The streams being decoded simultaneously could also be of different standards – the codec can support upto two standards.

The top level of the FIMV-MFC V4.0 contains the hardware modules which includes an ARM-7 CPU. The hardware and the firmware in the ARM-7 CPU performs the decoding functions for H.264 baseline, main, and high profiles, DIVX versions 4.x, 5.x, 6.x, and MPEG-4 simple and advanced simple profile, MPEG-2 simple and main profile, VC-1, and H.263 profile3 and the encoding functions for H.264 baseline, main, and high profile and MPEG-4 simple and advance simple profile. The optimum partition of the codec functions into software and hardware has ensured that the small sized hardware supports multiple standards.

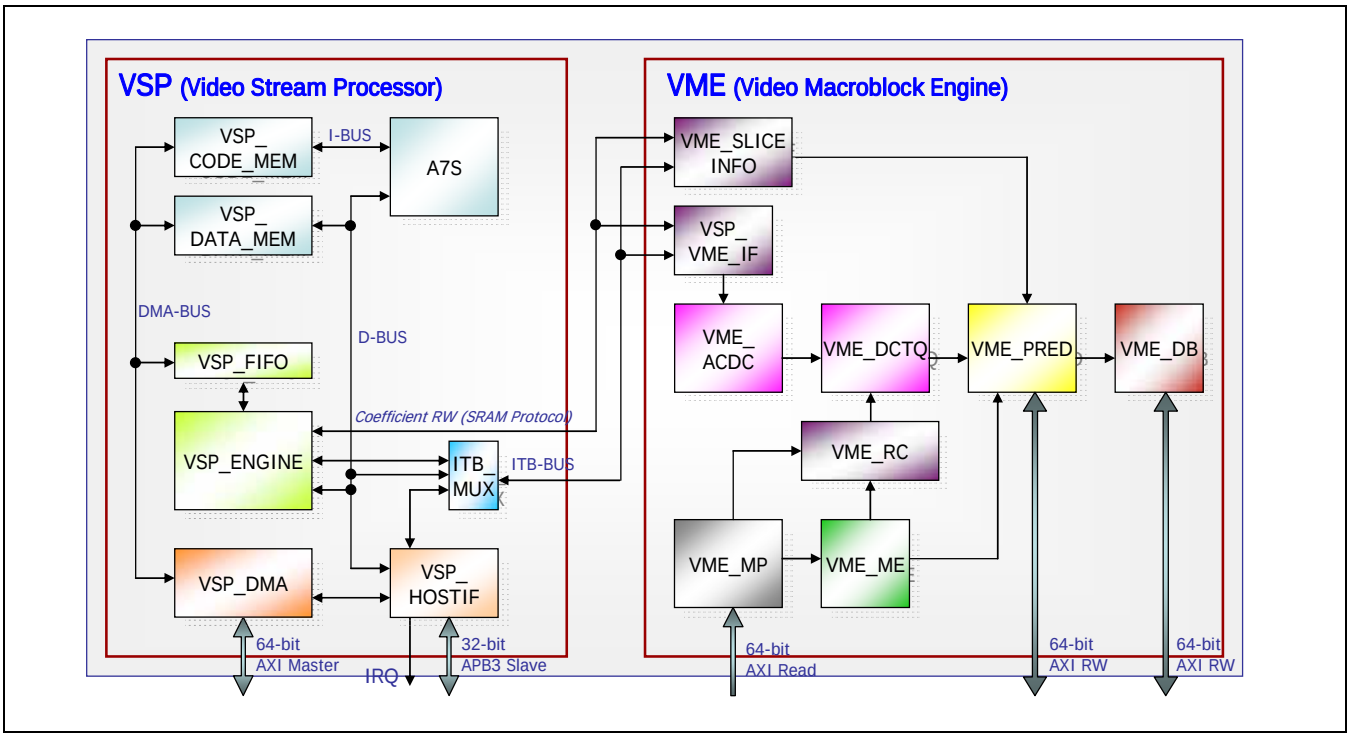


Figure 9.11-1 FIMV-MFC V4.0 Top Block Diagram

2 FEATURES AND STANDARD COMPLIANCE

2.1 SUPPORTED STANDARDS

- ITU-T H.264, ISO/IEC 14496-10
  - ♦ Decoding of H.264 baseline, main and high profile compliant video streams.
  - ♦ Encoding of H.264 baseline, main and high profiles.
- ITU-T H.263 Profile level 3
  - ♦ Decoding of H.263 P3 video streams.
  - ♦ Encoding of H.263 baseline
- ISO/IEC 14496-2 MPEG-4 ASP and DivX.
  - ♦ Decoding of MPEG-4 simple and advanced simple profile compliant video streams
  - ♦ Decoding of DivX 4.x, 5.x, 6.x compliant video streams.
  - ♦ Decoding of Xvid streams.
  - ♦ Encoding of MPEG-4 simple and advanced simple profiles.
- ISO/IEC 13818-2 MPEG-2
  - ♦ Decoding of MPEG-2 simple and main profile compliant video streams.
- SMPTE 421M VC-1 (upto Adv. Profile @ Level 2.0) decoding.

2.2 FEATURES

Synthesizable high-performance multi-format codec (FIMV-MFC V4.0) is decoding upto 1280x720 resolution image @ 30fps.

- Resolutions: upto 1280x720(720p, D1, VGA, QVGA, CIF, QCIF)
- Minimum size: 32x16, 16x32 for decoder, 32x32 for encoder
- Supports single stream 720p @ 30fps encoding/decoding.
- Time-multiplexed multi-stream encoding/decoding with fine-granular context switching.
- Supported chroma format: encoding 4:2:0 and decoding: 4:2:0, 4:0:0 8 bit per sample.
- Chrominance Interleaving.
- Supports CABAC/CABAD in H.264
- FMO and ASO not supported in H.264 decoding/encoding.
- Supports only one rectangular visual object in MPEG-4 encoding/decoding.
- Only forward reversible VLC (RVLC) is supported in MPEG-4 decoding.
- MPEG-4 post-processing by re-using H.264 in-loop filter.
- Supports only stationary warping function in GMC.

- Supports error resilience tool in MPEG-4 decoding.
- Error detection and concealment in decoding.
- Up to quarter-pel search for H.264
- Up to half-pel search for MPEG4
- Motion estimation search range : [+/-64, +/-32]
- All variable block sizes are supported. But 8x4, 4x8, 4x4 block sizes are not supported in encoding.
- Supports only DC prediction in MPEG-4 encoding.
- Rate control: CBR (Constant Bit-Rate) and VBR (Variable Bit-Rate)
- Frame level (H.264/MPEG4/H.263) and macroblock level (H.264) rate control can be enabled or disabled selectively.
- Progressive encoding only support.
- Non paired field mode is not supported.
- Startcode must be included at every position of frame or slice in stream at decoding.
- Input image YUV should be multiple of 16.
- B-frame encoding is not supported.
- Maximum number of slice in a frame: 32 slices at H.264 encoding.

2.3 TARGET PERFORMANCE AND FUNCTIONS

2.3.1 VIDEO DECODING CAPABILITY

- Single stream decoding up to 720p@30fps at 133MHz core clock frequency
- Dual stream decoding up to SD525@30fps at 133MHz core clock frequency

2.3.2 VIDEO ENCODING CAPABILITY

- Single stream encoding up to 720p@30fps at 133MHz core clock frequency
- Dual stream encoding up to SD525@30fps at 133MHz core clock frequency

2.3.3 OUT-LOOP POST PROCESSING

- Supports post deblock filtering by re-using H.264 in-loop filter. MPEG-4 deblocking function is implemented in H/W. In-loop filter included in H.264 is re-used for MPEG-4 De-blocking.

2.3.4 ERROR DETECTION AND CONCEALMENT

In case of corrupted or non compliant bitstreams, error detection and concealment is supported by FIMV-MFC V4.0. The error detection and concealment above picture layer (MPEG-4) or slice layer (H.264) is performed by software. The error detection and concealment is performed by the FIMV-MFC V4.0 without software interaction.

FIMV-MFC V4.0 performs error detection, e.g.:

- Detecting illegal values of syntax elements
- Detecting semantic errors (e.g. illegal prediction modes, wrong number of motion vectors, DCT coefficients)

  and error concealment:
- Recoverable errors: Recoverable errors are corrected and processing continues
- Non-recoverable errors: Corrupted data are discarded and lost data are concealed according to the selected concealment mode.
- Only minimum interaction with system processor required (autonomous concealment until next picture or slice header)

Different concealment modes are available, e.g. use of dedicated concealment motion vectors if available, use of neighbor or co-located motion vectors or use of zero motion vectors.

3 INTERNAL BLOCK DESCRIPTION

The FIMV-MFC V 4.0 is composed of two main blocks: the Video Stream Processor (VSP) and the Video Macroblock Engine (VME). The VSP parses the bitstreams and performs the entropy functions of video coding. It also passes the extracted informations from the bitstream to the VME module. The VSP performs these operations with the embedded ARM-7 microprocessor and hardware modules.

The VME module performs the video processing function of the video coding like ac/dc prediction, quantization, transform, motion estimation, motion compensation, deblock filtering and post filter. All the functions in this module are implemented in hardware.

3.1 VIDEO STREAM PROCESSOR PART

3.1.1 Video Stream Processor (VSP)

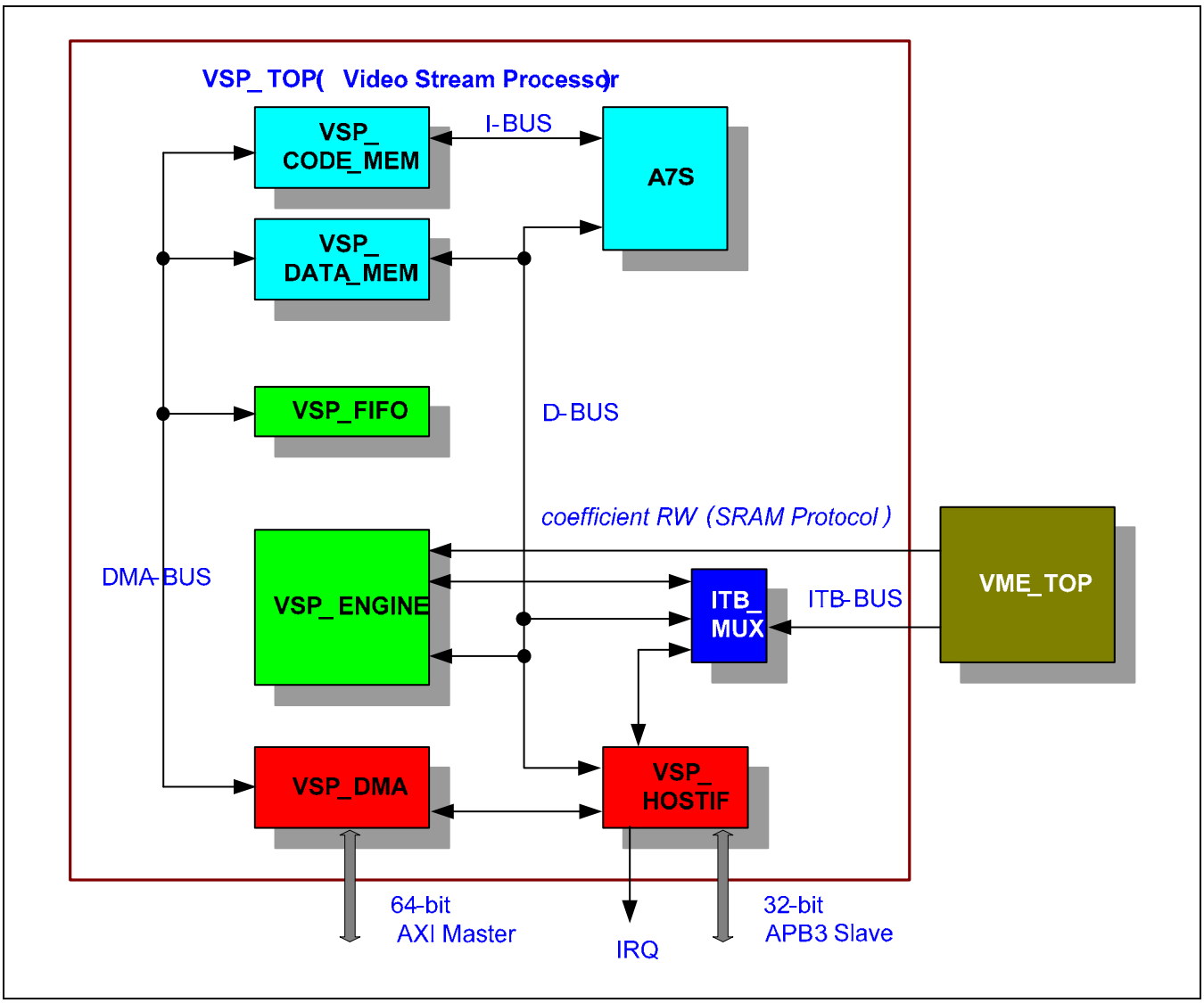


Figure 9.11-2 Video Stream Processor Block Diagram

All syntax elements in the bitstream are parsed by the VSP. Using hardwired based table lookup method, the ARM-7 microprocessor reads the syntax elements from the bitstreams. The ARM-7 can support two standards at a time and two bitstreams in time multiplexed manner. The microprocessor uses the syntax information to control the hardware modules including those in the codec module and passes required information to them.

The entropy functions of the VSP, encoding and decoding of coefficients using CABAC/CABAD or CAVLC/CAVLD and RLC/RLD, are performed by hardware modules in the VSP.

The hardware modules have 32 bit registers for data transfer to and from the firmware.

The ARM-7 inside of the VSP communicates with the host processor through the host interface. The host interface provides registers which can be accessed by both the ARM-7 and the host processor.

A DMA module the VSP reads and writes bitstream to and from the external memory. The DMA also initially loads the required firmware into the code memory in the VSP during initialization time.

The Syntax module performs encoding of syntax of data provided by the ARM-7 during encoding and passes syntax data to the ARM-7 during decoding. The syntax module reads data from the stream buffer which or writes data to stream buffer. The stream buffer is periodically filled/flushed while decoding/encoding by the DMA.

The Coefficient module handles the CABAC/CABAD and CAVLC/CAVLD and RLC/RLD entropy functions of video coding. When controlled by the ARM-7 this module performs the entropy functions using encoded data in the stream buffer and decoded data in the coefficient memory. The coefficient memorys are filled with data from the quantization block in the entropy module

3.2 VIDEO MACROBLOCK ENGINE PART

3.2.1 Video Macroblock Engine (VME)

Figure 9.11-4 shows the data flow of the encoding and decoding process in VME. In encoding, prediciton is performed using result of ME module. Prediction module loads only chrominance data of the reference frame in encoding. A luminance data is read from the local memory of the ME module to decrease additional bus-loading. In decoding, prediction is performed using reference luminance and chrominance data stored in mempool module. The mempool module loads reference data from external memory according to decoded motion vector. After prediction is operated, reconstructed pixel is filtered by deblocking module. VME uses the 64-bit AXI bus interface for data transfer from/to an external memory.

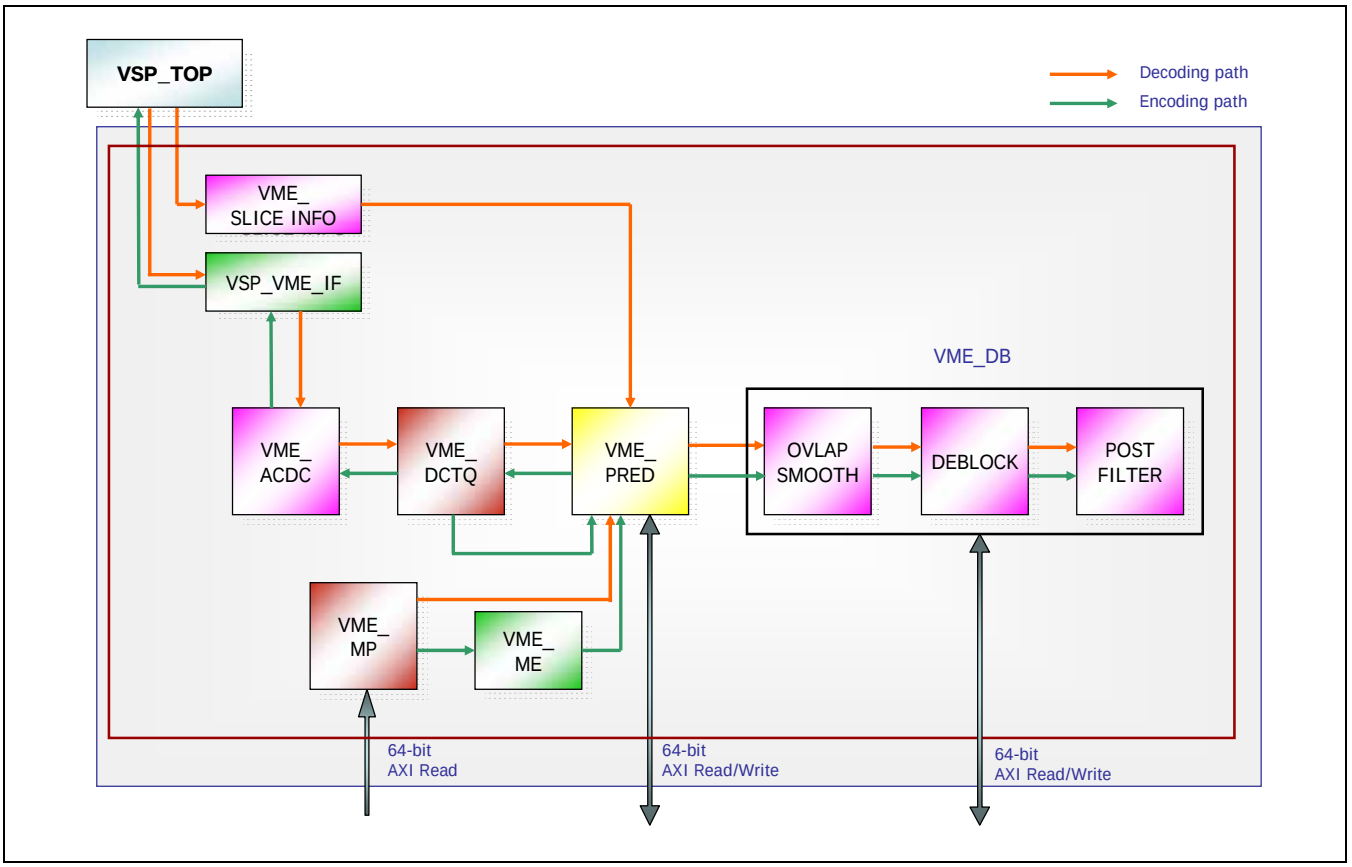


Figure 9.11-3 Video Macroblock Engine Block Diagram

3.2.2 MOTION ESTIMATION

- The Motion Estimation Block uses MRMCS (Multi-resolution search using multiple candidates and spatial correlation of motion field) algorithm, and the search range is [+/-64, +/-32] pixel.

Motion estimation supports:

- UMV (Unrestricted Motion Vector) mode
- Up to quarter-pel search for H.264
- Up to half-pel search for MPEG4
- Supports 16x16/8x8 block for MPEG4
- Supports 16x16/16x8/8x16/8x8 block for H.264

MRMCS algorithm

The Motion Estimation block has input as current pixels and pixels of search area, and calculates the optimal MV (Motion vector) that has the smallest cost function between original MB pixels and reference MB pixels. To calculate MVP (Motion vector prediction), ME block reads the upper MV from Line buffer memory and the left MV from previous MB. After Motion Estimation, block type information, MVD (motion vector difference), predicted pixels are transferred to the prediction block.

3.2.3 TRANSFORM & QUANTIZATION

Each residual macroblock is transformed, quantized and coded to remove the spatial redundancy. MPEG4 uses 8x8 DCT as the basic transform. H.264 encoder uses three transforms depending on the type of residual data that is to be coded - transform for 4x4 array of luma DC coefficients in intra 16x16 macroblock, transform for 2x2 array of chroma DC coefficients in any macroblock and transform for all other 4x4 blocks in the residual data. H.264 decoder supports 8x8 DCT and weighted quantization for high profile.

MPEG4 transform operates on 8x8 blocks of original data in intra macroblock or residual data after motion-compensated prediction. MPEG4 decoder support weighted quantization for advanced simple profile.

H.264 transform operates on 4x4 blocks of residual data after motion-compensated prediction or Intra prediction. The transform is based on the DCT but with some fundamental differences. Firstly, H.264 transform is integer transform. Secondly, the core part of the transform is multiply-free, it only requires additions and shifts. Thirdly, a scaling multiplication is integrated into the quantizer. The entire process of transform and quantization can be carried out using 16-bits integer arithmetic and only a single multiply per coefficient without any loss of accuracy.

Transform and quantization IP does not use DMA. This IP operating frequency is under the 133MHz.

3.2.4 PREDICTION

Natural video image has both temporal and spatial redundancy. Therefore, through the prediction, coding efficiency will be increased by transferring just residual for motion compression. For removing spatial redundancy, H.264 standard additionally includes intra-prediction compared with previous standards. VME_PRED can operate in Encoder or Decoder mode and each is composed of two parts. One is the INTRA PREDICTION part which extracts predictors from neighbor pixels in same picture. The other is the INTER PREDICTION part which extracts predictor that is interpolated by MV precision after being read from reference frames. When VME_PRED is operated in decoder, decoder stores reconstructed pixels, which are derived from adding residual which is

extracted by VME_DCTQ and predictor which is extracted by MV, into DPB (Decoded Picture Buffer) memory area. Predictors, which are muxed with Inter predictors and Intra predictors as MB mode, are added to Residual values. And the results go to VME_DB part. After deblock filtering, VME_DB part write filtered MB into the DPB (Decoded Picture Buffer) area which include reference pictures. VME_PRED has one dedicated DMA modules, so it writes current motion vector and reads co-loccated motion vector for direct mode in B picture.

In INTRA PREDICTION, intra prediction is operated in H.264 case. Intra prediction mode is derived from INTRA MODE DECISION that uses the full search methods for Intra4x4 block, Intra16x16 block and Chroma components. In INTER PREDICTION of Encoder, MV (Motion Vector) for the INTER PREDICTION is derived from ME (Motion Estimation). In VC-1 decoding, intensity compensation can be performed only once. Prediction mode is calculated by intra prediction mode or inter prediction mode according to MB prediction mode. The predicted data is subtracted from the original pixel and the results go to Forward Transform/Q. After forward Transform/Q, one path goes to the VSP part, the second path is similar to the decoder operation. Residual values of this second (decoding) path are added to the predictor and the results go to the VME_DB part. In Encoder, no DMA module is used. The VME_PRED uses only 1 reference frame for the prediction

3.2.5 DEBLOCKING FILTER

Deblocking filter removes blocking artifacts resulted from quantization. The MFC4.0 deblocking filter supports on-the-fly H.264 in-loop filter, H.263 annex J in-loop filter, VC1 overlap smoothing filter, VC1 in-loop filter and post filter. For multi format overlap smoothing and in-loop filter, the deblocking filter operates within coding loop. Filtered frames are used as reference frames for motion compensation of subsequent coded frames. Post filter operates outside the coding loop for display.

The data flow of multi format overlap smoothing and in-loop filter is shown in Figure 9.11-5. Under the multi format overlap smoothing and in-loop filter operation, filtered-data is written in reconstruction area through direct memory access (DMA) and used as reference frames. Beside unfiltered-data is written in upper row area and used as neighbor macroblocks at following row operation.

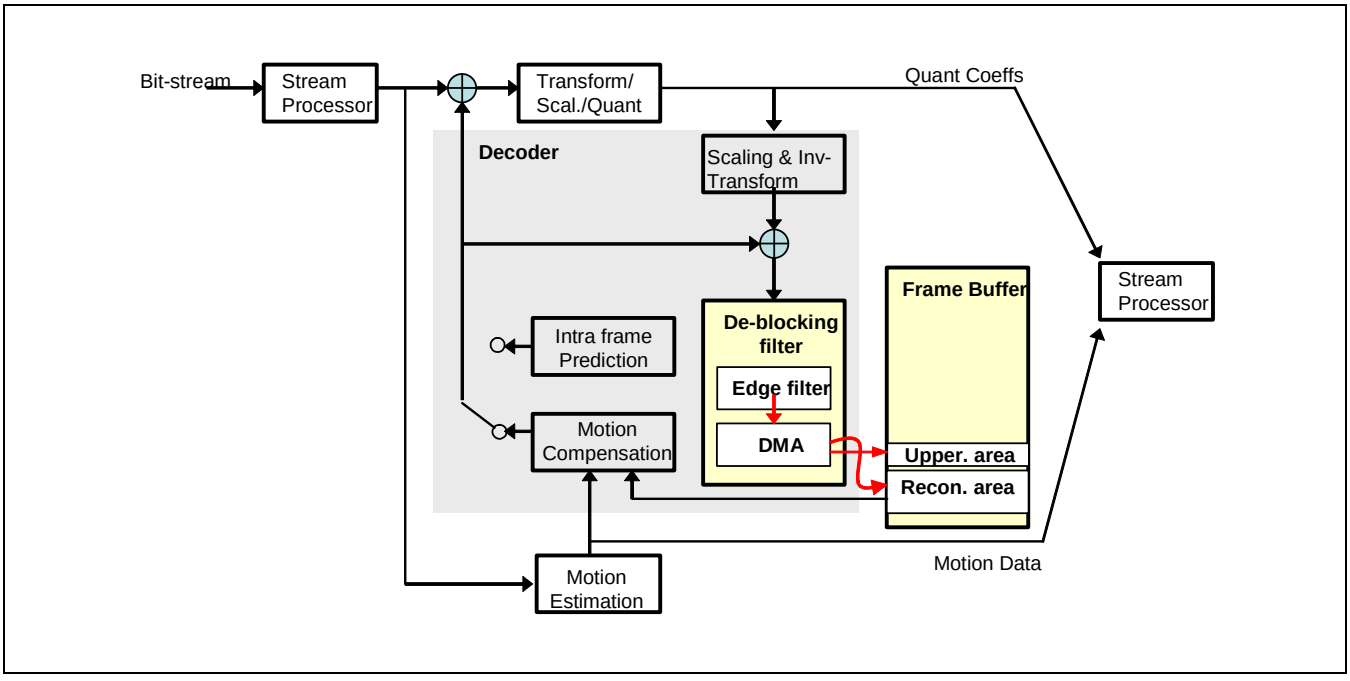


Figure 9.11-4 Data Flow of H.264 In-Loop Filter

Under the post filter operation, unfiltered-data for reference frame is written in reconstruction area and edge filtered-data for display frame is written in display area of frame buffer. Figure 9.11-6 shows the data flow of post filter. In the case of MPEG4, all I, P and B frames are written in display area regardless of post filter control value. And I and P frames, frames used for reference, are written in reconstruction area but B frames are not written. In the case of VC1, H.263 and H.264, all I, P and B frames are written in display area only when the post filter is turned on

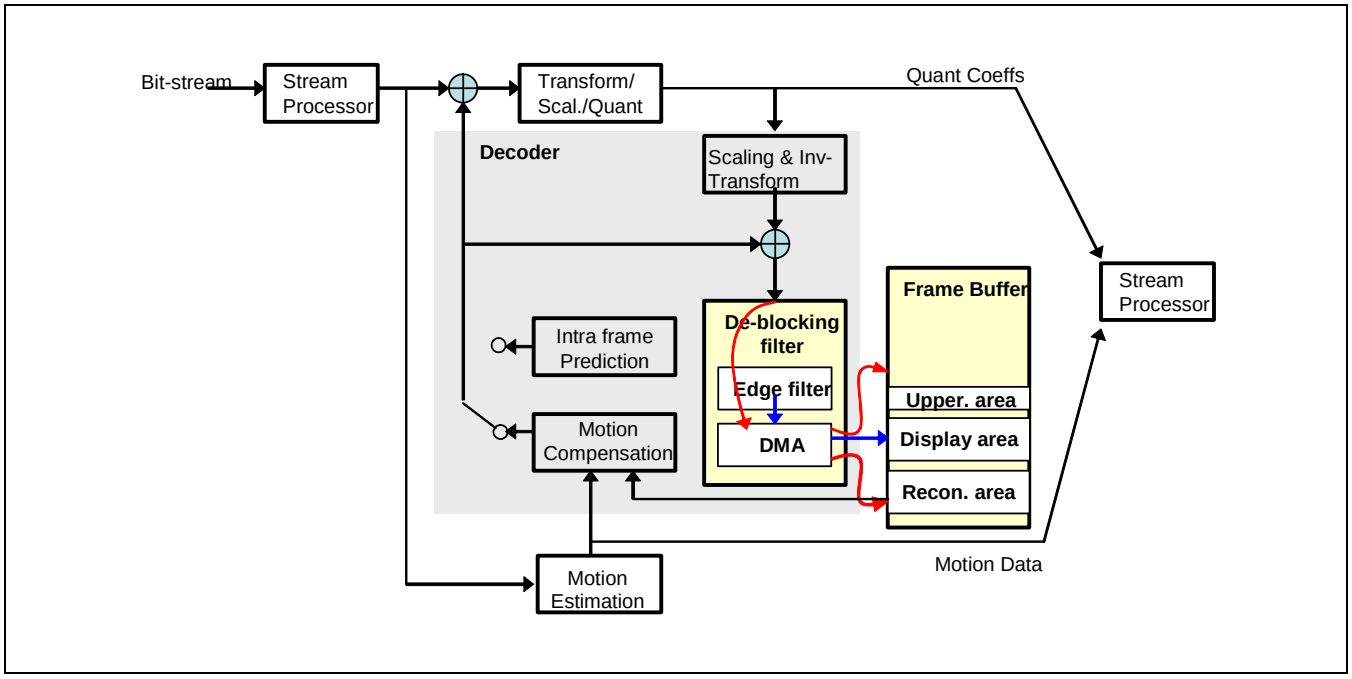


Figure 9.11-5 Data Flow of Post Filter

Block Diagram

Deblocking filter has eight internal buffers (four 1t0x44 buffers for luma coefficients, four 96x44 buffers for chroma coefficients). Memory word width is 44bits (four pixels unit) to support VC1 overlap smoothing filter. In the case of VC1, predicted coefficient is not clamped, so the bit width of predicted coefficient is 10. During the overlap smoothing filter, bit overflow can be happened.

Deblock filter is composed in two stages, one is filtering operation (Stage0) and the other is DMA operation (Stage1). Predicted coefficients are read from coefficient buffer in prediction block into internal buffers. Preceding operation block transfers macroblock information through information interface.

H.264 In-loop Filter

A filtering is applied to all 4x4 block edges except the boundary of the picture. For each macroblock, vertical edges are filtered first from left-hand side to right-hand side of the macroblock, and then horizontal edges are filtered from top to bottom of the macroblock. Pixel data changed by vertical edges filtering are used as input of the horizontal edges filtering.

The filter adjusts its strength depending on macroblock mode (Intra or Inter), the quantization parameter, and motion vector and pixel values. Global parameters for filtering operation such as loop filter disable value, chroma quantization offset, indexA, and indexB are set by the video stream processor.

H.263 Annex J In-loop Filter

The filtering is performed on 8x8 block edges, except across picture edge. The filtered data are clipped to the range of 0 to 255. The horizontal edges are filtered first from top to bottom, and then vertical edges are filtered from left to right. The pixels that are used in filtering across a horizontal edge shall not have been influenced by previous filtering across a vertical edge. Global parameters for filtering operation such as annex J control value are set by the video stream processor.

VC1 Overlap Smoothing Filter

VC-1 overlap-smoothing filtering shall be performed subsequent to decoding the frame, and prior to deblocking filter. The edges of an 8x8 block that separate two intra blocks are filtered. Overlap smoothing filer shall be carried out on the unclamped 10 bit reconstruction. In progressive and filed picture, vertical edges shall be filtered first, followed by the horizontal edges. In interlace picture, only the vertical edges shall be filtered and horizontal block boundaries shall not be filtered. The core filter applied to the four pixels is given blow:

Subsequent to filtering, the constant value of 128 shall be added to each pixel of the block, which shall be clamped to the range [0 255] to produce the reconstructed output. The video stream processor writes the overlap smoothing filer control value, conditional overlap flag, slice type, profile and so on.

VC1 In-loop Filter

VC-1 deblocking filtering process operates on the pixels that border neighboring blocks. For P pictures, the block boundaries may occur at every 4^{th}, 8^{th}, 12^{th}, etc pixel row or column depending on whether an inverse transform type. For I pictures filtering occurs at every 8^{th}, 16^{th}, 24^{th}, etc pixel row and column.

The horizontal boundary lines shall be filtered first followed by the vertical lines. For I frames, all the 8x8 boundaries are filtered. For P pictures, blocks may be intra or inter-coded. Intra-coded blocks use an 8x8 inverse transform, whereas inter-coed blocks use an 8x8, 8x4, 4x8 or 4x4 inverse transform. The boundary between transform blocks shall be filtered. In each 4-pixel segment, the result of third pixel pair filter operation determines whether the other three pixels in the segment are also filtered. In interlace frame, the filtering shall be done using the same field lines, never mixing different field lines. For horizontal block boundary, the two top field lines are filtered across the block boundary using top field lines only and the two bottom field lines across the block boundary shall be filtered using bottom field lines only

Post Filter

Post filter operates re-using H.264 in-loop filtering. The filter strength is determined as intra macroblock mode. All 8x8 block edges in MPEG4 and H.263 or all 4x4 block edges in H.264 and VC1 are filtered in the same order of H.264 in-loop filter.

4 FRAME MEMORY

A frame memory area is specified with base address (base_addr), horizontal / vertical image size (IMG_HSIZE/IMG_VSIZE). A complete image consists of Y, Cb and Cr components. The Cb and Cr pixels are stored in byte interleaved fashion. Therefore, an image needs 2 frame buffers, one for Y and another for Cb/Cr components, as shown in Figure 9.11-7. The sum of image horizontal size and image horizontal offset (horizontal offet makes the value of horizontal size to be multiple by 16). should be multiple of 16. The vertical image size for Cb/Cr frame buffer is half of the Y frame buffer.

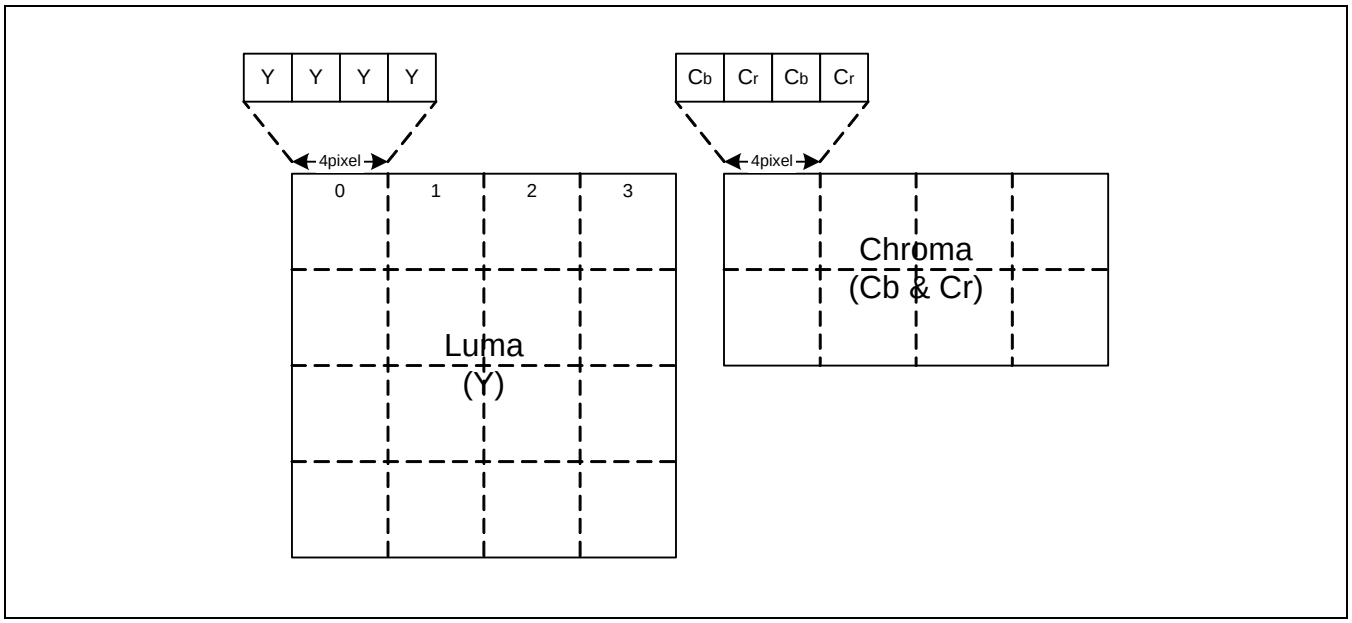


Figure 9.11-6 Luma & Chroma Pixel (8byts-aligned)

Frame can be stored in two ways, linear memory structure or tile mode memory structure. The host sets external memory parameters and memory structure method. The physical memory address of each pixel data is determined with memory structure, base address and coordinate of pixel in the frame.

4.1 2-PLANE INPUT IMAGE FORMAT

The input image source is 2-plane (chrominance interleaving) mode.

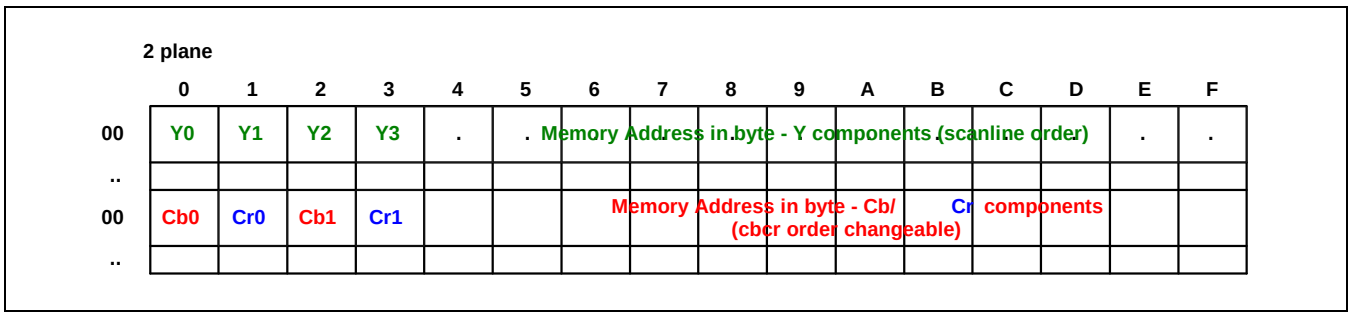


Figure 9.11-7 2-Plane (Chrominance Interleaving Mode)

4.2 TILED MEMORY STRUCTURE

For fast memory acces, the distribution of data is highly related so that, if multiple lines of data are allocated into a single DRAM page, DRAM utilization efficiency can be improved by open page scheme. This type of memory layout is called "tiled memory."

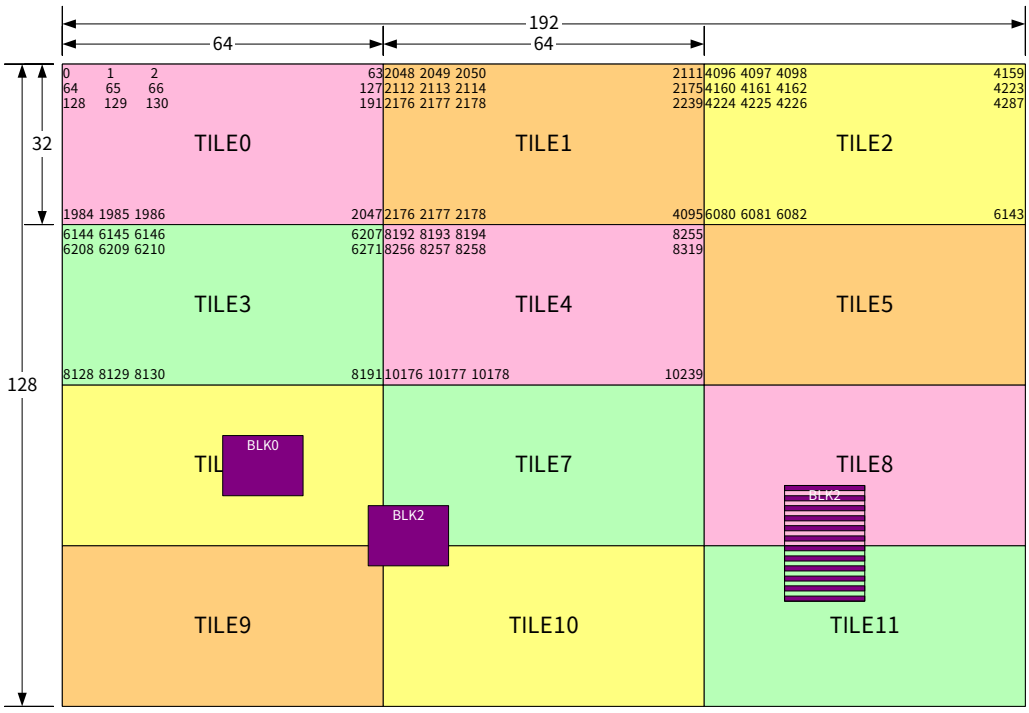


Figure 9.11-8 A Tiled Memory Layout for 192x128 Image-size with 64x32 Tile-size

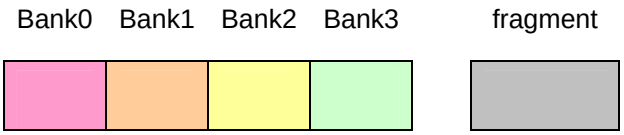


Figure 9.11-9 The Expansion of Tiles of Figure 9.11-8. in Horizontal Direction

5 ENCODING/DECODING COMMAND FLOW

5.1 DECODING FLOW

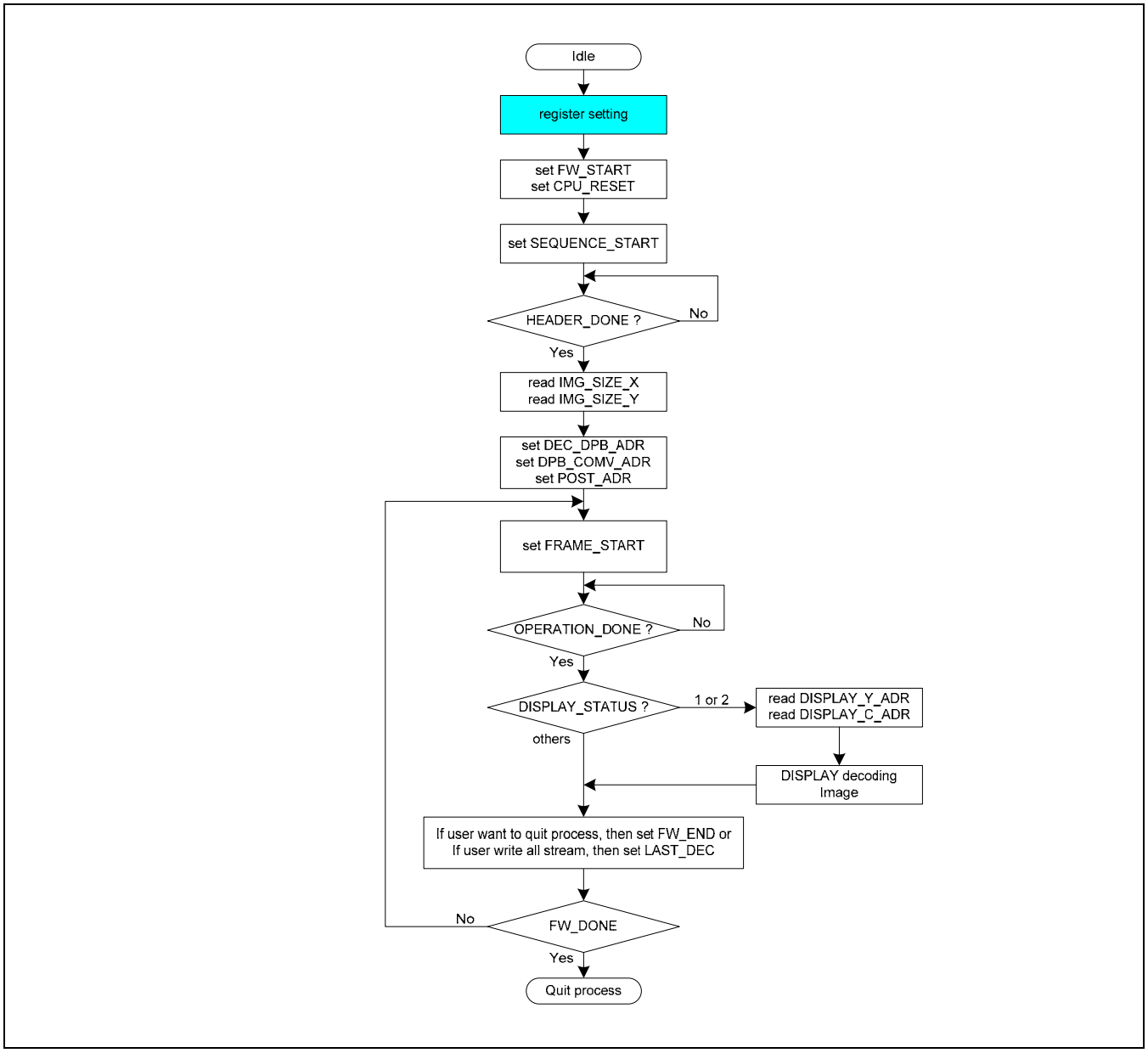


Figure 9.11-10 Decoding Flow

5.2 ENCODING FLOW

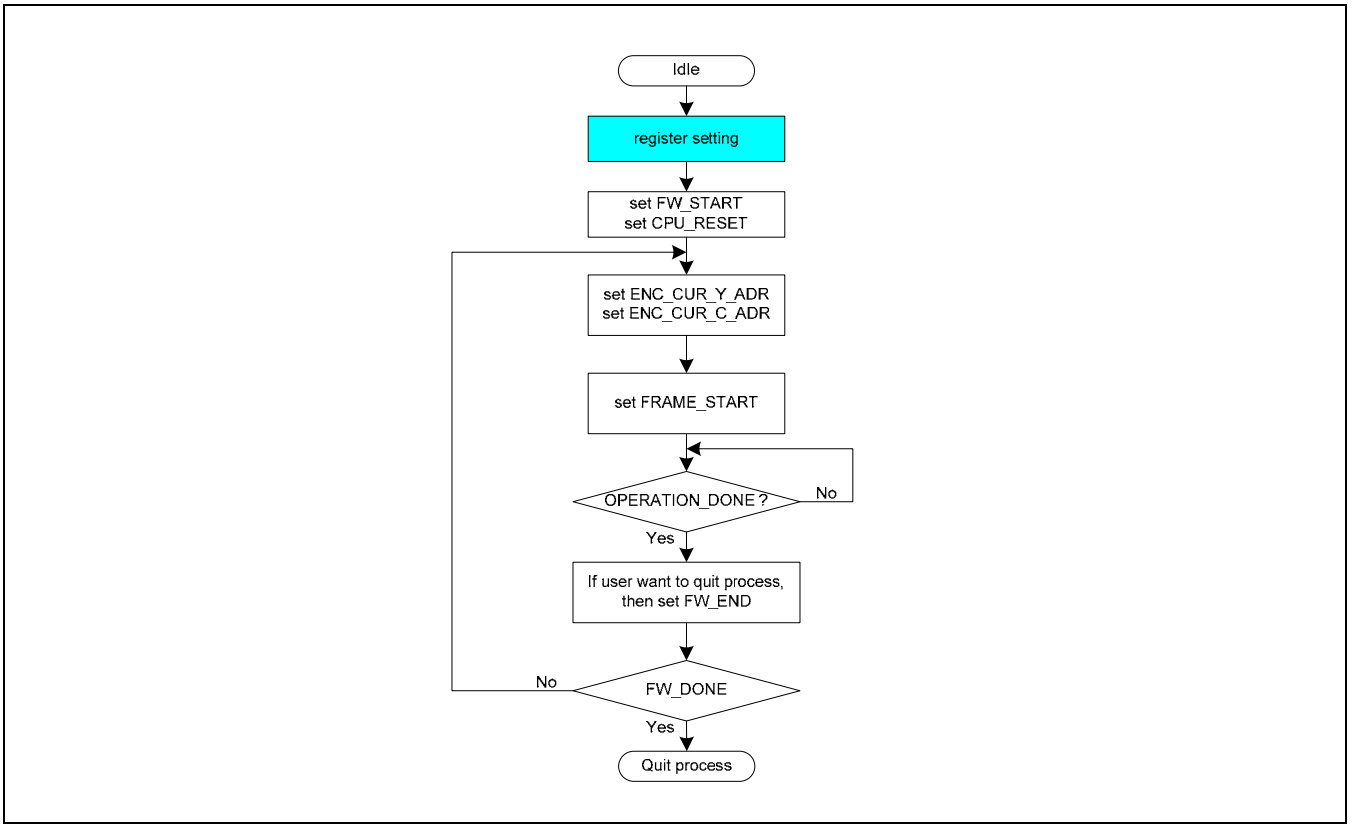


Figure 9.11-11 Encoding Flow

6 MULTI-CHANNEL CONTROL.

6.1 OVERALL ARCHITECTURE / MODEL CONVENTIONS

The architecture of the firmware for multi-channel implementation is described in this section. The Figure 9.11- explains the whole system. The whole system can be summarized as below.

- The host starts the processing by loading the boot code (firmware).
- The host initializes the hardware and starts the boot code (firmware).
- The F/W waits for a command of outside host.
- On receiving the valid command the firmware starts the task according to the input command.
- After processing the request the firmware again waits for the next command.

The commands used in the multi-channel implementation can be divided into four types. Those are INIT_CODEC, r, CHANNEL_SET, and CHANNEL_END. Each command is controlled by CH_ID (channel id). This is useful for multi-stream encoding/ decoding. For example, if you want to decode MPEG4 decoder and H.264 decoder at the same time(time multiplexing), you divide time-line per frames, decodes one frame of H.264 stream, and then decode one frame of MPEG4 stream and give CH_ID to each frame. If the frame rates are different, it can control the order of multi-stream using decoding MPEG4 stream twice and decoding H.264 stream one time. The detailed description of the various command are given later in the next section.

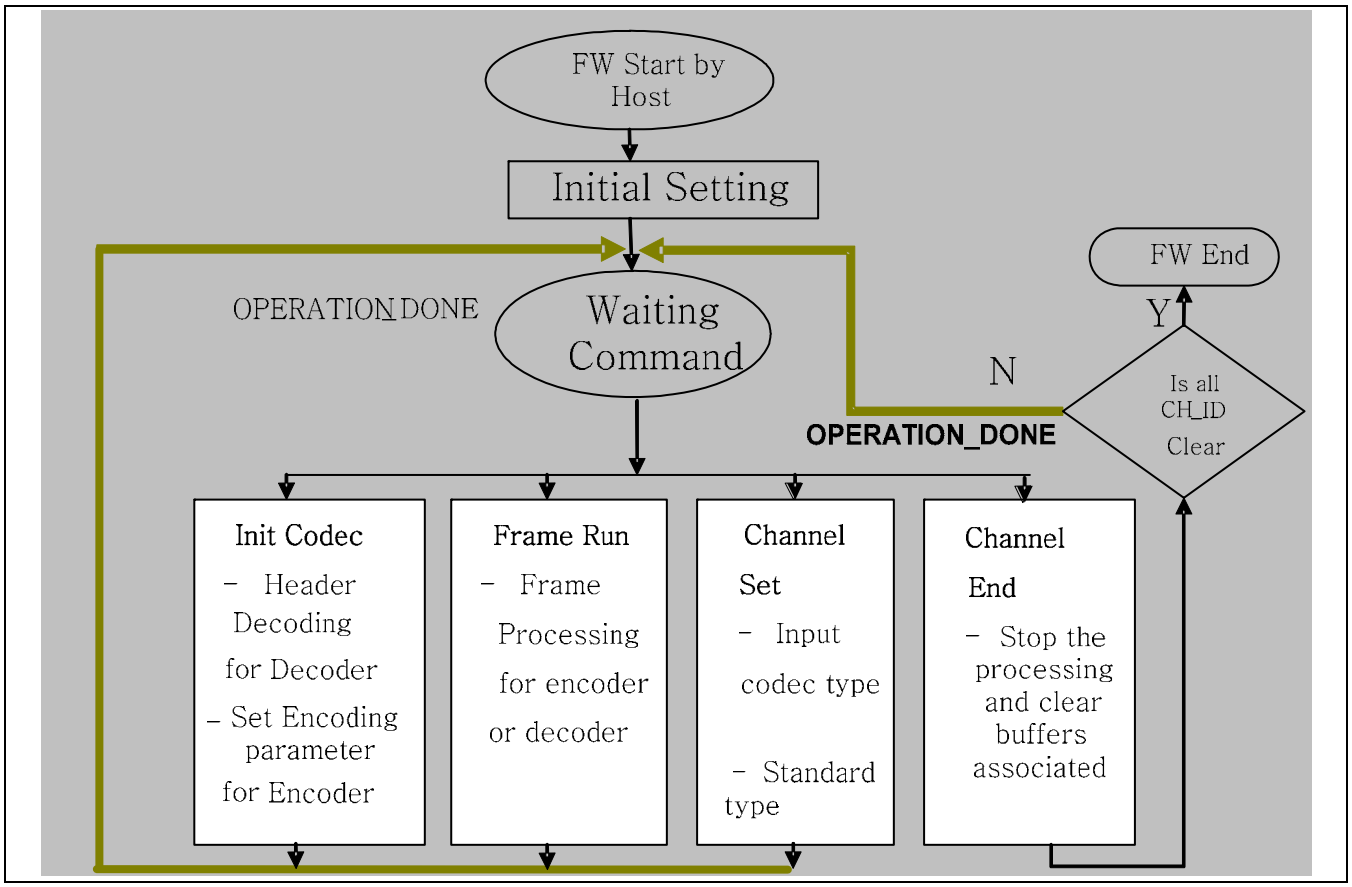


Figure 9.11-12 Multi-Channel Command Flow.

The multi-instance scheme can be implemented to run the multiple instance of the same standard. The whole functionality can be understood for the diagram given below.

In case of multi-channel decoder the input to the system is the bit steams of same or different (in case of multi-standard support) standard and the output will be store buffer of each channel.

In case of multi-channel encoder the input to the system is the YUV data and the output will be the bit stream of required standard.

6.2 COMMAND DESCRIPTION

In this section the tasks performed by different commands are described.

6.2.1 CHANNEL_SET

This is the most basic command. This command assigns each channel the standard and entity (decoding/ encoding) and does the initialization for the channel. This command has to execute in the beginning of the encoding/decoding and should be executed once for a particular channel before setting of CHANNEL_END, FRAME_RUN.. If the other commands are executed before this command then it is an error. The section below describes the detailed functions of the command on host and firmware side.

Host process

- Channel ID write (CH_ID)
- Write the
  - ♦ Codec standard (STANDARD_SEL)
  - ♦ Decoding/Encoding (STANDARD_SEL)
  - ♦ Command type (MFC_COMMAND_TYPE)
- Set the FRAME_START = 1
- Host wait for OPERATION_DONE = 1

ARM7 process

- Backup the values setting in Host according to CH_ID (Standard, codec type etc.)
- Set the command status as SET in the command status array.
- DMA the program data to the internal memory for the selected codec if it is not DMADONE earlier.
- Set OPERATION_DONE = 1

6.2.2 CHANNEL_END

The CHANNEL_END command is mainly responsible for cleaning all the memory of a channel when the operation is over or if the user wants to manually kill the channel.

Host process

- Channel ID write (Host Interface Register CH_ID)
- Write the Command type (MFC_COMMAND_TYPE)
- Set the FRAME_START = 1
- Host wait for OPERATION_DONE = 1

ARM7 process

- Flush stack memory areas according to channel ID.
- Flush memory areas according to channel ID.
- Flush control data according to channel ID.

- Check whether all the channels are flushed from CH_ID control buffer. End the F/W if all the channel IDs are checked out or get into the command waiting state if all the channels are not flushed.
- Set OPERATION_DONE = 1

6.2.3 INIT_CODEC

This INIT_CODEC command does different functions for encoder and decoder.

For encoder this command parses the configuration file in host and the firmware stores all the parameters from host through the host interface registers.

For decoder this command does the Header Parsing for the coded bit stream. Host needs image size when calculates DPB buffer's size or recon buffer. HOST processor waits until HEADER_DONE register will be high after writing to FRAME_START register. If HEADER_DONE register is high, HOST processor reads IMG_SIZE_X and IMG_SIZE_Y register and calculates the SDRAM areas according to image size and write DEC_DPB_ADDR, DPB_COMV_ADDR, POST_ADDR register. HOST processor waits until OPERATION_DONE regitser will be high after wrting to SEQ_START regitser.

This command has to be executed once for a particular channel and it is an error if it is tried to be executed after that.

Host process

- Channel ID write (Host Interface Register CH_ID)
- Aloocated channel working buffer(VSP_BUF_ADDR,DB_STT_ADDR).
- Write the
  - Start address of stream header
  - Framesize of stream header
  - Display delay and number of extra buffer(NUM_EXTRA_BUF)
  - Decoding/Encoding (STANDARD_SEL)
  - Command type (MFC_COMMAND_TYPE)
- Set the FRAME_START = 1
- It starts the header decoding in the firmware side and waits for the signal HEADER_DONE to be set to 1
- After this it gets the DPB sizes, height, and width value. And allocated other buffers
- Set the SEQ_START = 1
- The host waits for OPERATION_DONE = 1

ARM7 process

- Does the header decoding of the stream
- Set the status of the INIT_CODEC command to set.
  - Sends the signal HEADER_DONE = '1' and OPERATION_DONE = '1' once it finished

6.2.4 FRAME_RUN

This command is for decoding/encoding one frame.

HOST processor waits until OPERATION_DONE register will be '1' after setting FRAME_RUN command. If OPERATION_DONE register is '1', HOST processor reads DISPLAY_STATUS register to display decoded image and to check display status when decoding. At this time, if enable status bit in display status register is '1', the host can read the values of DISPLAY_Y_ADDR register and DISPLAY_C_ADDR register. The address values stored in those registers enable the current image to be displayed.

When encoding, HOST processor has to check the encoding end signal. And then prepares next encoding.

Host process

- Write the Channel ID
- Update the required host interface registers
  - ♦ Decoding : Update decoder control register
  - ♦ Encoding : Update Encoder control register
- Set the FRAME_START = 1
- After this the frame processing will start in the firmware side and the host will be waiting for the signal from firmware

ARM7 process

- Execute Frame encoding/decoding.
- Done signal generation (OPERATION_DONE = 1)

6.3 FUNCTIONAL MODEL

6.3.1 CODE FLOW

The diagram below describes the flow of the encoder/decoder command control code.

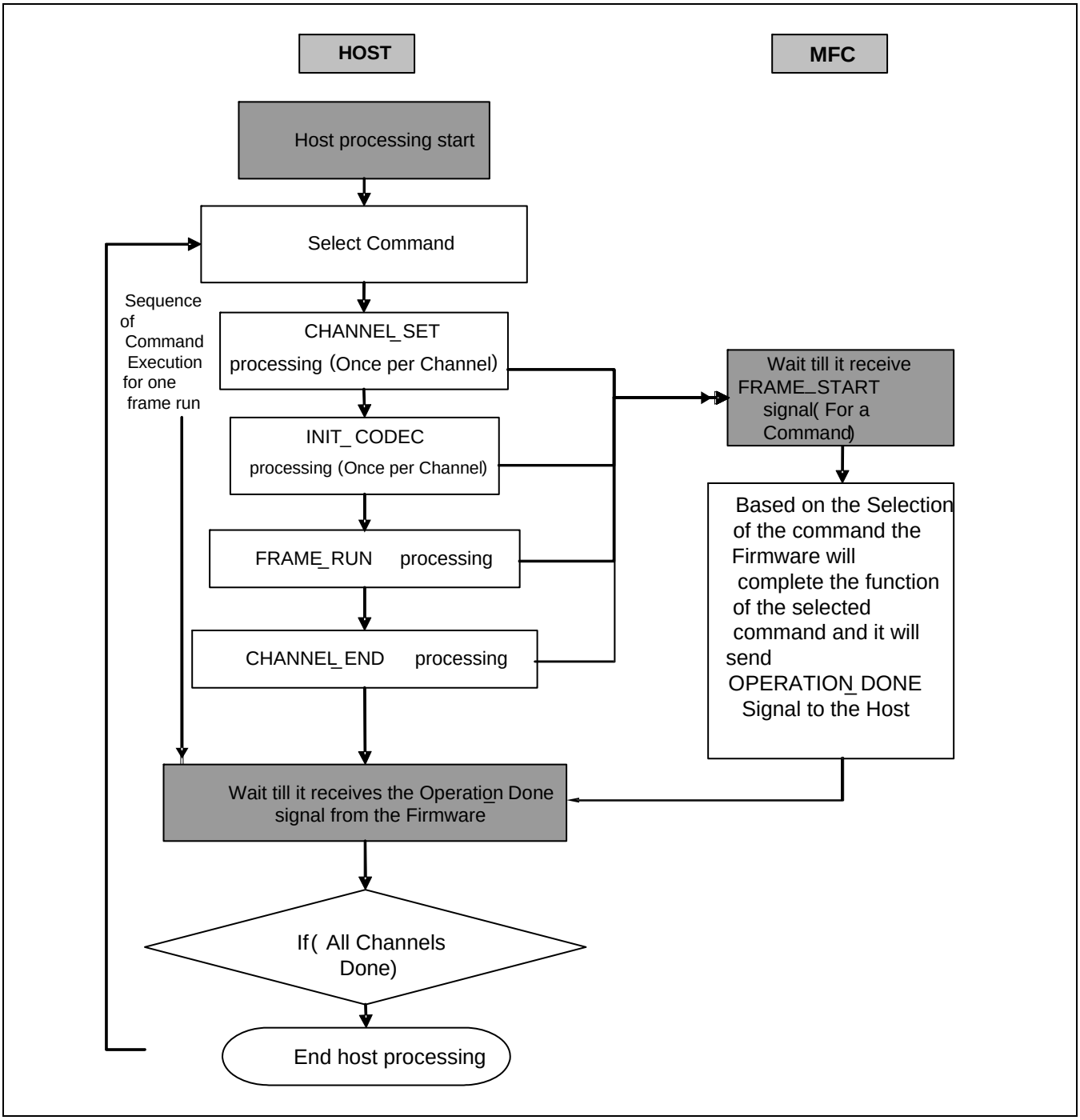


Figure 9.11-13 Multi-Channel Command Flow

6.3.2 CODE FLOW DESCRIPTION

This section contains the description of the Code Flow diagram.

The diagram shows the basic flow of the encoder or decoder and also the sequence of the commands needs to be executed for one frame processing. For running one complete frame the sequence of command execution should be like CHANNEL_SET, INIT_CODEC and FRAME_RUN. The command CHANNEL_SET and INIT_CODEC should execute once per each channel. If the host wants to stop the execution of one channel then the CHANNEL_END command is executed. During one command processing, other command shouldn't be issued.

For one successful frame running the above sequence should be followed, otherwise the command status register will show error status.

7 BITSTREAM BUFFER CONTROL

7.1 FW LOADING.

The Host can load the FW from the external memory into the internal Code memory using the FIMV-MFCV4.0 CODEC H/W as follows.

- Set the **BITS_ENDIAN** register to '0'. (little endian)
- Set the **BUS_MASTER** register to '1'.
- Set the **DMA_EXT_ADDR** register. (start address of Command Control FW)
- Set the **BOOTCODE_SIZE** register. (word count size of Command Control FW)
- Set the **DMA_START** register.
- Wait until the DMA_DONE is 1 to check the FW loading done.
- Set the **BITS_ENDIAN** register to '1'. (big endian)
- Set the **BUS_MASTER** register to '0'.

7.2 STREAM BUFFER MANAGEMENT

The stream buffer is controlled in Line Buffer Scheme. So in case of decoding, the stream for that frame shall be ready in the external memory before decoding and in case of encoding, the stream for that frame shall be ready in the external memory after encoding.

The detailed description is explained below.

7.2.1 Decoder case

The stream consists of stream header and frame data. For example, the H.264 stream consists of stream header like SPS, PPS and SEI and the frame data which may contain several slices. The Host shall notify to the VSP_DMA where the stream for the frame data or the frame data with the stream header is using the Host Interface before starting decoding a frame.

The following figures show the stream format example for each standard. The each stream which is represented as a rectangular contains each frame data and stream header data if it exists.

For this purpose, there are 5 registers. These are the EXT_BUF_START_ADDR, EXT_BUF_END_ADDR, START_BYTE_NUM, DEC_UNIT_SIZE and the HOST_PTR. The EXT_BUF_START_ADDR indicates where the stream starts, and the EXT_BUF_END_ADDR indicates where the stream ends, and the START_BYTE_NUM indicates how many bytes are valid in the EXT_BUF_START_ADDR which shall be 8 byte aligned, and the DEC_UNIT_SIZE indicates how many bytes shall be decoded for the corresponding frame. The HOST_PTR is not required for the Line Buffer scheme but the VSP_DMA is designed as the Circular Buffer Scheme, so it shall be set as same with the EXT_BUF_END_ADDR.

MPEG4, H.264 and VC-1 Case

The Stream header contains Visual Object Sequence, Visual Object, Video Object, Visual Object Layer and User Data for the MPEG4 and SPS, PPS and SEI for the H.264 and MetaData, Sequence Layer and Entry-Pointer Layer for the VC-1.

Except for the first frame, the Host sets the FRAME_START to decode a frame and checks the OPERATION_DONE. In case of the first frame, the Host sets the SEQUENCE_START to decode stream header and checks the HEADER_DONE and then the Host sets the FRAME_START to decode a frame and checks the OPERATION_DONE.

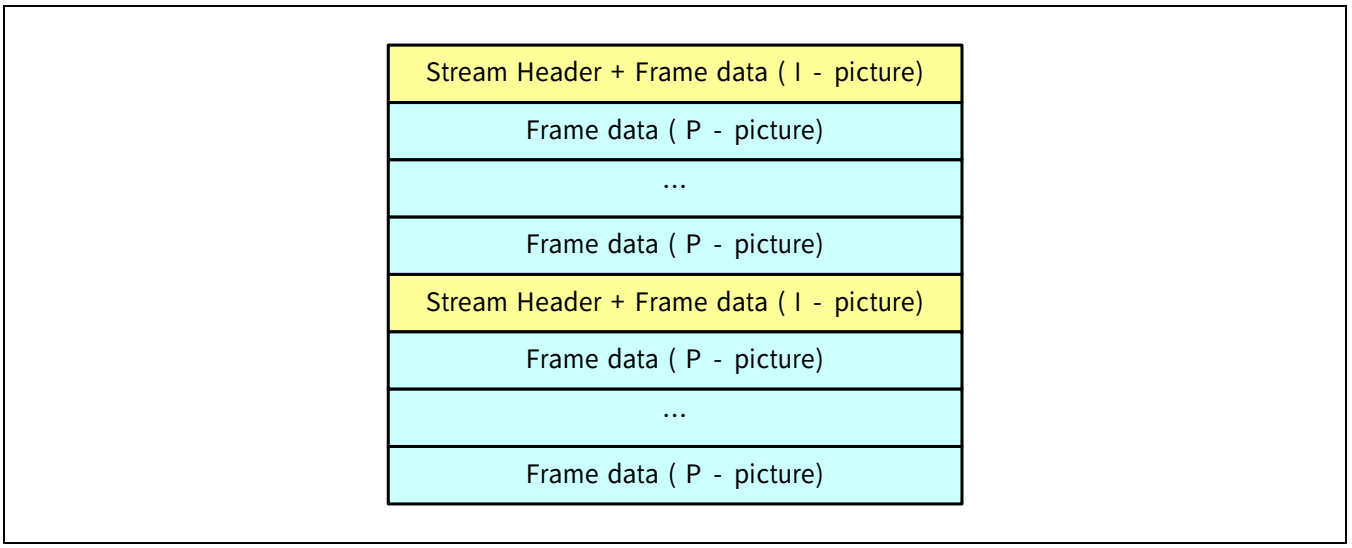


Figure 9.11-14 The Stream header contains

H.263 Case

In case of H.263, there is no sequence header and there is only frame header so the stream format is as following figure.

Except for the first frame, the Host sets the FRAME_START to decode a frame and checks the OPERATION_DONE. In case of the first frame, the Host sets the SEQUENCE_START to decode stream header and checks the HEADER_DONE and then the Host sets the FRAME_START to decode a frame and checks the OPERATION_DONE. Because there is no Stream header in the H.263, for the first frame, the Host sets the same Stream Buffer configuration twice. The first configuration is for the Stream Header and the second configuration is for the frame data.

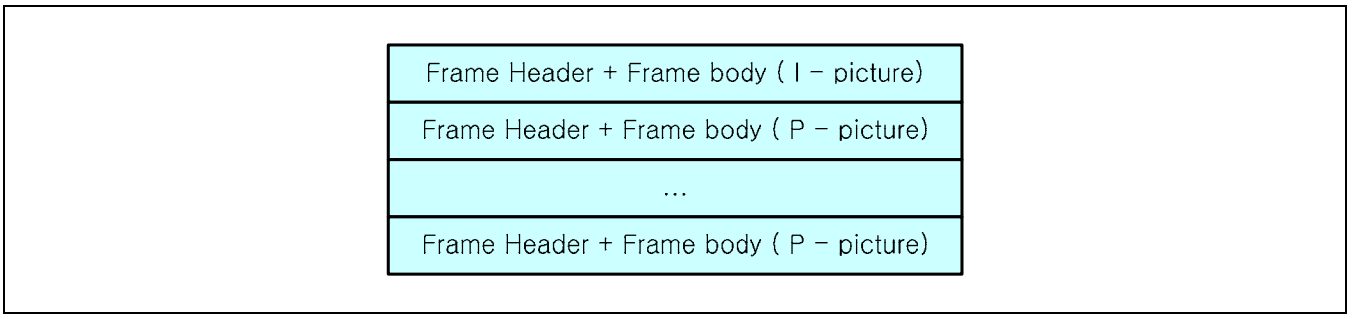


Figure 9.11-15 No sequence header

7.2.2 ENCODER CASE

The stream format is same with the decoder. Each stream contains the frame data and the stream header if it exists.

The Host shall set the VSP_DMA configuration (EXT_BUF_START_ADDR, EXT_BUF_END_ADDR and HOST_PTR) at every frame including the first frame with the stream header. The EXT_BUF_START_ADDR indicates where the encoded stream shall start and the EXT_BUF_END_ADDR indicates until where the encoded stream can be written and the HOST_PTR shall be set as same with the EXT_BUF_START_ADDR even though it is not used because the VSP_DMA is designed as the Circular Buffer Scheme. After encoding a frame, the Host can get information about how many bits are encoded for the corresponding frame using the ENC_UNIT_SIZE.

The Host should guarantee that the EXT_BUF_START_ADDR and the EXT_BUF_END_ADDR have a range enough to include one frame-sized stream in encoding. If not, it may be stuck.

H.264 and MPEG4 Case

The Stream header is located at the first position of the stream. So the encoded stream for the first frame shall contain stream header as well as the frame data. The Stream header contains the SPS and PPS. Except for the first frame, the encoded stream contains only frame data. And there is another SFR named ENC_HEADER_SIZE to indicate how many bytes are for the stream header.

The Host shall set the FRAME_START and then check the OPERATION_DONE at every frame. In case of encoding, there is no need to set the SEQUENCE_START and check the HEADER_DONE.

H.263 Case

In case of H.263, there is no stream header in the encoded stream. So the encoded stream contains only the frame data.

8 POWER CONTROL

The main function of the power control feature is to continue operation from the same point where the operation was ended because of power off for MFC.

8.1 SLEEP

Host process

- Channel ID write '-1' (CH_ID)
- Write the Command type to SLEEP(MFC_COMMAND_TYPE)
- Set the FRAME_START = 1
- Host wait for OPERATION_DONE = 1
- MFC can be switched power off

8.2 WAKEUP

Host process

- Host enable MFC block power on
- Load Command Control FW to MFC
- Start F/W (set FW_START '1')
- Channel ID write '-1' (CH_ID)
- Write the Command type to WAKEUP (MFC_COMMAND_TYPE)
- Host wait for OPERATION_DONE = 1
- Host can start with the normal operation

9 REGISTER DESCRIPTION

Register	Offset	R/W	Description	Reset Value
DMA/STREAM REGISTER				
DMA_START	0xF100_0000	R/W	DMA Start Register	0x0000_0000
BOOTCODE_SIZE	0xF100_000C	R/W	Firmware Size Register	0x0000_0000
DMA_EXTADDR	0xF100_0014	R/W	DMA External Memory Address Register	0x0000_0000
EXT_BUF_START_ADDR	0xF100_0018	R/W	External Buffer Start Address Register	0x0000_0000
EXT_BUF_END_ADDR	0xF100_001C	R/W	External Buffer End Address Register	0x0000_0000
DMA_INT_ADDR	0xF100_0020	R/W	Dma Internal Address Register	0x0000_0000
HOST_PTR_ADDR	0xF100_0024	R/W	Host Pointer Address Register	0x0000_0000
LAST_DEC	0xF100_0028	R/W	Last Decoding Control Register	0x0000_0000
DONE_M	0xF100_002C	R/W	Done F/W transfer Register	0x0000_0000
BITS_ENDIAN	0xF100_0044	R/W	Dma Endian Setting Register	0x0000_0000
DEC_UNIT_SIZE	0xF100_0054	R/W	Decoding Stream Unit Size Register	0x0000_0000
ENC_UNIT_SIZE	0xF100_0058	R	Encoded Stream Unit Size Register	0x0000_0000
START_BYTE_NUM	0xF100_005C	R/W	Start Byte Number Register	0x0000_0000
ENC_HEADER_SIZE	0xF100_0060	R	Encoded Header Size Register	0x0000_0000
COMMAND CONTORL				
STD_SEL_REG	0xF100_0100	R/W	Standard Selection Register	0x0000_0000
CH_ID	0xF100_0104	R/W	Channel ID Register	0x0000_0000
CPU_RESET	0xF100_0108	R/W	Cpu Reset Register	0x0000_0000
FW_END	0xF100_010c	R/W	Firmware End Register	0x0000_0000
BUS_MASTER	0xF100_0110	R/W	Bus Master SettingRegister	0x0000_0000
FRAME_START	0xF100_0114	R/W	Frame Start Register	0x0000_0000
IMG_SIZE_X	0xF100_0118	R/W	Horizontal Image Size Register	0x0000_0000
IMG_SIZE_Y	0xF100_011C	R/W	Vertical Image Size Register	0x0000_0000
POST_ON	0xF100_0124	R/W	Mpeg4 Post Filter On/Off Register	0x0000_0000
FRAME_RATE	0xF100_0128	R/W	Frame Rate Setting Register	0x0000_0000
SEQ_START	0xF100_012C	R/W	Sequence Start Control Register	0x0000_0000
SW_RESET	0xF100_0130	R/W	Software Reset Register	0x0000_0000
FW_START	0xF100_0134	R/W	Firmware Start Register	0x0000_0001
ARM_ENDIAN	0xF100_0138	R/W	Arm Endian Register	0x0000_0001
ERR_CTRL	0xF100_013C	R/W	Error Control Setting Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
F/W ADDRESS				
FW_STT_ADDR_0	0xF100_0200	R/W	Mp4 encoder F/W Address Register	0x0000_0000
FW_STT_ADDR_1	0xF100_0204	R/W	Mp4 Decoder F/W Address Register	0x0000_0000
FW_STT_ADDR_2	0xF100_0208	R/W	H264 encoder F/W Address Register	0x0000_0000
FW_STT_ADDR_3	0xF100_020C	R/W	H264 Decoder F/W Address Register	0x0000_0000
FW_STT_ADDR_4	0xF100_0210	R/W	VC-1 Decoder F/W Address Register	0x0000_0000
FW_STT_ADDR_5	0xF100_0214	R/W	MPEG-2 Decoder F/W Address Register	0x0000_0000
FW_STT_ADDR_6	0xF100_0218	R/W	H263 Decoder F/W Address Register	0x0000_0000
VSP_BUF_ADDR	0xF100_0230	R/W	VSP Working Buffer Address Register	0x0000_0000
DB_STT_ADDR	0xF100_0234	R/W	Deblock Line Buffer Setting Address Register	0x0000_0000
ENCODER CONTROL				
PROFILE	0xF100_0300	R/W	Profile Setting Register	0x0000_0000
IDR_PERIOD	0xF100_0304	R/W		0x0000_0000
I_PERIOD	0xF100_0308	R/W	I-Picture Period Register	0x0000_0000
FRAME_QP_INIT	0xF100_030C	R/W	Register	0x0000_0000
H264_OPTION	0xF100_0310	R/W	Entropy Encoding Control Register	0x0000_0000
DB_FILTER_OPTION	0xF100_0314	R/W	Deblocking Filter Control Register	0x0000_0000
SHORT_HD_ON	0xF100_0318	R/W	Short Header On/Off Control Register	0x0000_0000
MSLICE_ENA	0xF100_031C	R/W	Multi-Slice Enable Register	0x0000_0000
MSLICE_SEL	0xF100_0320	R/W	Multi-Slice Selection Register	0x0000_0000
MSLICE_MB	0xF100_0324	R/W	Multi-Slice Interval MB Register	0x0000_0000
MSLICE_BYTE	0xF100_0328	R/W	Multi-Slice Interval Byte Register	0x0000_0000
DECODER CONTROL				
DISPLAY_Y_ADDR	0xF100_0400	R/W	Display Luminace Address Register	0x0000_0000
DISPLAY_C_ADDR	0xF100_0404	R/W	Display Chrominace Address Register	0x0000_0000
DISPLAY_STATUS	0xF100_0408	R/W	Display Status Register	0x0000_0000
HEADER_DONE	0xF100_040C	R/W	Header_done reading Register	0x0000_0000
FRAME_NUM	0xF100_0410	R/W	Frame_Number Reading Register	0x0000_0000
INTERRUPT CONTROL				
INT_OFF	0xF100_0500	R/W	Interrupt Control Register	0x0000_0000
INT_MODE	0xF100_0504	R/W	Interrupt Level Selection Register	0x0000_0000
INT_DONE_CLEAR	0xF100_0508	R/W	Interrupt Clear Register	0x0000_0000
OPERATION_DONE	0xF100_050C	R/W	Operation Status Register	0x0000_0000
FW_DONE	0xF100_0510	R/W	Firmware Status Register	0x0000_0000
INT_STATUS	0xF100_0514	R/W	Interrupt Status Register	0x0000_0000
INT_ENABLE	0xF100_0518	R/W	Interrupt Enable Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
MEMORY CONTROLLER SETTING				
MEM_STRUCT_SET	0xF100_0600	R/W	Memoory Structure Setting Register	0x0000_0000
ENCODER BUFFER ADDRESS				
ENC_CUR_Y_ADDR	0xF100_0800	R/W	Encoder Current Y Buffer Address Register	0x0000_0000
ENC_CUR_CBCR_ADDR	0xF100_0804	R/W	Encoder Current C Buffer Address Register	0x0000_0000
ENC_DBP_ADDR	0xF100_0808	R/W	Encoder Dpb start Address Register	0x0000_0000
CIR_MB_NUM	0xF100_080C	R/W	Intra Refresh Macroblock Number Register	0x0000_0000
DECODER BUFFER ADDRESS				
DEC_DPB_ADDR	0xF100_0900	R/W	Decoder Decoded Picture Buffer Address Register	0x0000_0000
DPB_COMV_ADDR	0xF100_0904	R/W	Colocated Motion Vector Buffer Address Register	0x0000_0000
POST_ADDR	0xF100_0908	R/W	Post Filter Output Address Setting Register	0x0000_0000
DPB_SIZE	0xF100_090C	R	Decoded Picture Buffer Size Register	0x0000_0000
MFCV4.0 VERSION				
RC_CONFIG	0xF100_0a00	R/W	Rate Control ConfigurationRegister	0x0000_0000
RC_FRAME_RATE	0xF100_0a04	R/W	Rate Control Framerate Register	0x0000_0000
RC_BIT_RATE	0xF100_0a08	R/W	Rate Control BitrateRegister	0x0000_0000
RC_QBOUND	0xF100_0a0C	R/W	Quantizer Parameter Boundary Register	0x0000_0000
RC_RPARA	0xF100_0a10	R/W	Reaction Coefficient Register	0x0000_0000
RC_MB_CTRL	0xF100_0a14	R/W	MacroBlock Level Rate Control Register	0x0000_0000
RC_QOUT	0xF100_0a18	R/W	Internal Reference Quantization Scale Register	0x0000_0000
MFCV4.0 VERSION				
MFC4_VERSION	0xF100_0b00	R/W	MFC4 Version Register	0x0000_0040
CROPPING INFORMATION				
CROP1	0xF100_0C00	R/W	Cropping Information 1 Register	0x0000_0000
CROP2	0xF100_0C04	R/W	Cropping Information 2 Register	0x0000_0000
DEC_FRM_SIZE	0xF100_0C08	R/W	Decoded Frame Size Register	0x0000_0000
MULTI-CHANNEL COMMAND				
COMMAND_TYPE	0XF100_0D00	R/W	Command Type Register	0x0000_0000
MULTI-CHANNEL COMMAND				
FRAME_TYPE	0xF100_0C0C	R	Frame Type Register	0x0000_0000
NUM_EXTRA_DPB	0xF100_0D04	W	Number Of Extra DPB Register	0x0000_0000
CODEC_COMMAND	0xF100_0D08	W	Codec Command Register	0x0000_0000

9.1 DMA CONTROL REGISTER

9.1.1 DMA START REGISTER (DMA_START, R/W, ADDRESS = 0xf100_0000)

DMA_START	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
DMA_START	[0]	Start signal that it is triggered to transfer initial boot code to internal code memory. It is triggered when it's high.	0

9.1.2 FIRMWARE SIZE REGISTER (BOOTCODE_SIZE, R/W, ADDRESS = 0xf100_000C)

BOOTCODE_SIZE	Bit	Description	Reset Value
BOOTCODE_SIZE	[31:0]	Word count size of Command Control firmware to transfer into internal memory using DMA. byte count size of codec firmware to transfer into internal memory using DMA.	0

9.1.3 DMA EXTERNAL MEMORY ADDRESS REGISTER (DMA_EXT_ADDR, R/W, ADDRESS = 0xf100_0014)

DMA_EXT_ADDR	Bit	Description	Reset Value
DMA_EXTADDR	[31:0]	Start address of the external memory to transfer Firmware.	0

9.1.4 EXTERNAL BUFFER START ADDRESS REGISTER (EXT_BUF_START_ADDR, R/W, ADDRESS = 0xf100_0018)

EXT_BUF_START_ADDR	Bit	Description	Reset Value
EXT_BUF_START_ADDR	[31:0]	Start address of Bitstream Buffer. **Note**: Lower three bits shall be 0.	0

9.1.5 EXTERNAL BUFFER END ADDRESS REGISTER (EXT_BUF_END_ADDR, R/W, ADDRESS = 0xf100_001C)

EXT_BUF_END_ADDR	Bit	Description	Reset Value
EXT_BUF_END_ADDR	[31:0]	End address of Bitstream Buffer. must set 'frame start address + frame size+0x64' **Note:** Lower three bits shall be 0.	0

9.1.6 DMA INTERNAL ADDRESS REGISTER (DMA_INT_ADDR, R/W, ADDDRESS = 0xf100_0020)

DMA_INT_ADDR	Bit	Description	Reset Value
DMA_INT_ADDR	[31:0]	This is an address of internal memory accessed by DMA. It shall be 0 to transfer FW code.	0

9.1.7 HOST POINTER ADDRESS REGISTER (HOST_PTR_ADDR, R/W, ADDRESS = 0xf100_0024)

HOST_PTR_ADDR	Bit	Description	Reset Value
HOST_PTR_ADDR	[31:0]	This register is used for Bitstream Buffer control. It indicates until where the host reads stream in Bitstream Buffer during encoding. If stream buffer size is guaranteed more than one frame size, this value should be set EXT_BUF_END_ADDR value.	0

9.1.8 LAST DECODING CONTROL REGISTER (LAST_DEC, R/W, ADDRESS = 0xf100_0028)

LAST_DEC	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
LAST_DEC	[0]	When this bit is set to 1 in decoding, this shows no more stream will be added in the bitstream buffer. HOST_PTR_ADDR register's value indicates the last address of the existing stream **Note:** This bit is reset to 0 automatically when it is read.	0

9.1.9 DONE FIRMWARE TRANSFER REGISTER (DONE_M, R/W, ADDRESS = 0xf100_002C)

DONE_M	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
DONE_M	[0]	Done signal of Firmware transfer to internal memory. **Note:** This bit is reset to 0 automatically when it is read.	0

9.1.10 DMA ENDIAN SETTING REGISTER (BITS_ENDIAN, R/W, ADDRESS = 0xf100_0044)

BITS_ENDIAN	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
BITS_ENDIAN	[0]	DMA transfer endian setting register 0 = little endian 1 = big endian.	0

9.1.11 DECODING STREAM UNIT SIZE (DEC_UNIT_SIZE, R/W, ADDRESS = 0xf100_0054)

DEC_UNIT_SIZE	Bit	Description	Reset Value
DEC_UNIT_SIZE	[31:0]	Stream unit size per frame External host informs MFC how many bytes one decoding frame in the stream has	0

9.1.12 ENCODED STREAM UNIT SIZE (ENC_UNIT_SIZE, R, ADDRESS = 0xf100_0058)

ENC_UNIT_SIZE	Bit	Description	Reset Value
ENC_UNIT_SIZE	[31:0]	MFC informs the host of the encoded frame size	0

9.1.13 START BYTE NUMBER (START_BYTE_NUM, R/W, ADDRESS = 0xf100_005c)

START_BYTE_NUM	Bit	Description	Reset Value
START_BYTE_NUM	[3:0]	Remaining byte position of the stream when it is not word-aligned. If the stream is word-aligned, this bits will be set to 8 instead of 0	0

9.1.14 ENCODED HEADER SIZE (ENC_HEADER_SIZE, R, ADDRESS = 0xf100_0060)

ENC_HEADER_SIZE	Bit	Description	Reset Value
ENC_HEADER_SIZE	[31:0]	Encoded header size number. ENC_UNIT_SIZE include this count number.	0

9.2 COMMON CONTROL REGISTER

9.2.1 STANDARD SELECTION REGISTER (STD_SEL_REG, R/W, ADDRESS = 0xf100_0100)

STD_SEL_REG	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
ENC_ON	[4]	Encoder/Decoder selection bit 0 = decoder 1 = encoder	0
Reserved	[3]	Reserved	0
STANDARD_SEL	[2:0]	MFC Codec standard selection bits. Enc_ON is '1' 0 = MPEG-4 1 = H.264 Enc_ON is '0' 0 = MPEG-4(Divx,XviD) 1 = H.264 4 = H.263 5 = MPEG-2 6 = VC-1	0

9.2.2 CHANNEL ID REGISTER (CH_ID, R/W, ADDRESS = 0xf100_0104)

CH_ID	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
CH_ID	[4:0]	Channel index is an ID identifying host's job during multi-channel encoding or decoding.	0

9.2.3 CPU RESET REGISTER (CPU_RESET, R/W, ADDRESS = 0xf100_0108)

CPU_RESET	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
CPU_RESET	[0]	This connects reset of internal ARM7. If reset, PC count reset 0. 0 = internal CPU reset. 1 = internal CPU running.	0

9.2.4 FIRMWARE END REGISTER (FW_END, R/W, ADDRESS = 0xf100_010c)

FW_END	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
FW_END	[0]	This signal means external host notifies encoding/decoding operation about the end of all video processing jobs. IP terminates ARM7 processing after receiving this signal. 1 = Terminate F/W 0 = Normal operation	0

9.2.5 BUS MASTER SETTING REGISTER (BUS_MASTER, R/W, ADDRESS = 0xf100_0110)

BUS_MASTER	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
BUS_MASTER	[0]	This register sets the master of DMA before the setting of DMA in external host. If 1, external bus master is the host 0 = bus master is MFC. <ex. MFC can write DMA register> 1 = bus master is Host. <ex. Host can write DMA register>	0

9.2.6 FRAME START REGISTER (FRAME_START, R/W, ADDRESS = 0xf100_0114)

FRAME_START	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
FRAME_START	[0]	This is a frame level start signal that IP, video codec, encodes/decodes from external host. It is triggered when it's high. **Note**: This bit is reset to 0 automatically when the FW reads this bit.	0

9.2.7 HORIZONTAL IMAGE SIZE REGISTER (IMG_SIZE_X, W,R, ADDRESS = 0xf100_0118)

IMG_SIZE_X	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
IMG_SIZE_X	[15:0]	Horizontal image real size. **Note:** In Encoder, Write only register. In Decoder, Read only register	0

9.2.8 VERTICAL IMAGE SIZE REGISTER (IMG_SIZE_Y, R/W, ADDRESS = 0xf100_011c)

IMG_SIZE_Y	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
IMG_SIZE_Y	[15:0]	Vertical image real size. **Note:** In Encoder, Write only register. In Decoder, Read only register	0

9.2.9 MPEG4 POST FILTER ON/OFF CONTROL REGISTER (POST_ON, R/W, ADDRESS = 0xf100_0124)

POST_ON	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
POST_ON	[0]	mpeg4 post filter on/off signal 0 = off 1 = on	0

9.2.10 FRAME RATE SETTING REGISTER (FRAME_RATE, R/W, ADDRESS = 0xf100_0128)

FRAME_RATE	Bit	Description	Reset Value
QUOTIENT_VAL	[31:16]	Quotient value for rate control	0
REMAINDER_VAL	[15:0]	Remainder value for rate control	0

9.2.11 SEQUENCE START CONTROL REGISTER (SEQUENCE_START, R/W, ADDRESS = 0xf100_012C)

SEQUENCE_START	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
SEQUENCE_START	[0]	Stream sequence start signal.	0

9.2.12 SOFTWARE RESET REGISTER (SW_RESET, R/W, ADDRESS = 0xf100_0130)

SW_RESET	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
SW_RESET	[0]	Software reset control signal 1 = S/W reset, 0 = normal **Note:** This bit is reset to 0 automatically.	0

9.2.13 FIRMWARE START REGISTER (FW_START, R/W, ADDRESS = 0xf100_0134)

FW_START	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
FW_START	[0]	When this bit is set to 1, the FW starts encoding/ decoding.	1

9.2.14 ARM ENDIAN SETTING REGISTER (ARM_ENDIAN, R/W, ADDRESS = 0xf100_0138)

ARM_ENDIAN	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
ARM_ENDIAN	[0]	ARM code memory endian setting value. 0 = little endian 1 = big endian	1

9.2.15 ERROR CONTROL SETTING REGISTER (ERR_CTRL, R/W, ADDRESS = 0xf100_013c)

ERR_CTRL	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
ERR_CTRL	[0]	When error mets, 0 = Concealment ON and decoding one frame. 1 = Interrrupt and stop.	0

9.3 F/W START ADDRESS AND WORKING BUFFER ADDRESS REGISTER

9.3.1 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_0, R/W, ADDRESS = 0xf100_0200)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_0	[31:0]	This is MP4 encoder F/W start address.	0

9.3.2 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_1, R/W, ADDRESS = 0xf100_0204)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_1	[31:0]	This is MP4 decoder F/W start address.	0

9.3.3 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_2, R/W, ADDRESS = 0xf100_0208)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_2	[31:0]	This is H.264 encoder F/W start address.	0

9.3.4 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_3, R/W, ADDRESS = 0xf100_020c)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_3	[31:0]	This is H.264 decoder F/W start address.	0

9.3.5 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_4, R/W, ADDRESS = 0xf100_0210)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_4	[31:0]	This is VC-1 decoder F/W start address.	0

9.3.6 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_5, R/W, ADDRESS = 0xf100_0214)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_5	[31:0]	This is MPEG-2 decoder F/W start address.	0

9.3.7 FIRMWARE START ADDRESS REGISTER (FW_STT_ADDR_6, R/W, ADDRESS = 0xf100_0218)

FW_STT_ADDR	Bit	Description	Reset Value
FW_STT_ADDR_6	[31:0]	This is H263 decoder F/W start address.	0

9.3.8 VSP WORKING BUFFER ADDRESS SETTING REGISTER (VSP_BUF_ADDR, R/W, ADDRESS = 0xf100_0230)

FW_STT_ADDR	Bit	Description	Reset Value
VSP_BUF_ADDR	[31:0]	VSP temporal buffer start address. VSP temporal buffer use temporal SPS, PPS data saving area. The size needs about 327 kbytes. (Or 81.75K Words).	0

9.3.9 DEBLOCK LINE BUFFER SETTING ADDRESS (DB_STT_ADDR, R/W, ADDRESS = 0xf100_0234)

FW_STT_ADDR	Bit	Description	Reset Value
DB_STT_ADDR	[31:0]	Deblock Line Buffer Start Address. DB line buffer size is (IMG_HSIZE+IMG_HOFFSET)*4*32.	0

9.4 ENCODER CONTROL REGISTER

9.4.1 PROFILE SETTING REGISTER (PROFILE, R/W, ADDRES = 0xf100_0300)

PROFILE	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
LEVEL	[15:8]	Level in H.264(supported 1.1~3.1) ex) 31 stands for level 3.1 When RC is disable. Host must set '31'.	0
Reserved	[7:5]	reserved	-
PROFILE	[4:0]	MPEG4 : 0 = SP 1 = ASP H.264 : 0 = Baseline 1 = Main 2 = High H.264 : 0 = Baseline	0

9.4.2 I-PICTURE PERIOD REGISTER (I_PERIOD, R/W, ADDRESS = 0xf100_0308)

I_PERIOD	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0
I_PERIOD	[15:0]	The number of P-picture <16'hFFFF: all sequence P-picture> <16'h0: all sequence I-picture> <16'h1: IPIPIP....> <16'h2: IPPIPPIPP....>	0

9.4.3 ENTROPY ENCODING CONTROL REGISTER (ENTROPY_CONTROL, R/W, ADDRESS = 0xf100_0310)

ENTROPY_CONTROL	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	-
FIXED_NUMBER	[3:2]	Model number (0, 1, 2) for fixed method for inter slice. Valid at ENTROPY_SEL is 1.	0
Reserved	[1]	Reserved	-
ENTROPY_SEL	[0]	0 = CAVLC 1 = CABAC	0

9.4.4 DEBLOCKING FILTER CONTROL REGISTER (DEBLOCKING_CONTROL, R/W, ADDRESS = 0xf100_0314)

LOOPFILTER_DIS	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	-
SLICE_ALPHA_C0_OFFSET_DIV2	[11:7]	Slice_alpah_c0_offset_div2 value in slice header. This value is 2's complement value. This value's range is -6 to 6.	0x00
SLICE_BETA_OFFSET_DIV2	[6:2]	Slice_beta_offset_div2 value in slice header. This value is 2's complement value. This value's range is -6 to 6.	0x00
DISABLE_DEBLOCKING_FILTER_IDC	[1:0]	Disable deblocking filter IDC in slice header 0 = All blocking boundary filtering. 1 = Disable deblocking filter. 2 = Filtering except slice boundary.	0x0

9.4.5 SHORT HEADER ON/OFF CONTROL REGISTER (SHORT_HD_ON, R/W, ADDRESS = 0xf100_0318)

SHORT_HD_ON	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
SHORT_HD_ON	[0]	0 = short header off. 1 = short header on.	0

9.4.6 MULTI-SLICE ENABLE REGISTER (MSLICE_ENA, R/W, ADDRESS = 0xf100_031c)

MSLICE_ENA	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
MSLICE_ENA	[0]	Multi-slice select control register. 1 = Enable multi-slice per frame 0 = one slice per frame	0

9.4.7 MULTI-SLICE SELECTION REGISTER (MSLICE_SEL, R/W, ADDRESS = 0xf100_0320)

MSLICE_SEL	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	0
MSLICE_SEL	[1:0]	Select method of multi-slice. 0 = Multi slicing is done by MB count value 1 = Multi slicing is done by byte count value. 2 = Multi slicing is done by variable MB count value.	0

9.4.8 MULTI-SLICE INTERVAL REGISTER (MSLICE_MB, R/W, ADDRESS = 0xf100_0324)

MSLICE_MB	Bit	Description	Reset Value
MSLICE_MB	[31:0]	This register is used MSLICE_ENA ==1, and When MSLICE_SEL == 0, number of MB per slice. MSLICE_MB value can be 1 to (Horizontal image size * Vertical image size / 16 /16).	0

9.4.9 MULTI-SLICE INTERVAL REGISTER (MSLICE_BYTE, R/W, ADDRESS = 0xf100_0324)

MSLICE_BYTE	Bit	Description	Reset Value
MSLICE_BYTE	[31:0]	This register is used when MSLICE_ENA == 1 and MSLICE_SEL == 1. When multi-slice, this byte value uses for dividing by byte count. This isn't over MSLICE_BYTE.	0

9.5 DECODER CONTROL REGISTER

9.5.1 DISPLAY LUMINANCE ADDRESS SETTING REGISTER (DISPLAY_Y_ADDR, R, ADDRESS = 0xf100_0400)

DISPLAY_Y_ADDR	Bit	Description	Reset Value
DISPLAY_Y_ADDR	[31:0]	Display Luminance address.	0

9.5.2 DISPLAY CHROMINANCE ADDRESS SETTING REGISTER (DISPLAY_C_ADDR, R, ADDRESS = 0xf100_0404)

DISPLAY_C_ADDR	Bit	Description	Reset Value
DISPLAY_C_ADDR	[31:0]	Display Chrominance address.	0

9.5.3 DISPLAY STATUS REGISTER (DISPLAY_STATUS, R, ADDRESS = 0xf100_0408)

DISPLAY_STATUS	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	0
ERROR_GEN	[4]	1. Error detected at the current frame decoding. 0: No error	0
DISPLAY_STATUS	[3:0]	This register shows the status of display. 0: decoding only. [1:0] 0 = decoding only (no display) 1 = decoding and display. 2 = display only. [2] 0 = progressive frame. 1 = interlace frame [3] 0 = normal 1 = Cropping information is exist	0

9.5.4 HEADER_DONE READING REGISTER (HEADER_DONE, R, ADDRESS = 0xf100_040c)

HEADER_DONE	Bit	Description	Reset Value
HEADER_DONE	[0]	0 = F/W running. 1 = header parsing done.	0

9.5.5 FRAME NUMBER READING REGISTER (FRAME_NUM, R, ADDRESS = 0xf100_0410)

FRAME_NUM	Bit	Description	Reset Value
FRAME_NUM	[31:0]	Frame count value indicates to external host. Host can check what frame is operated.	0

9.6 INTERRUPT CONTROL REGISTER

9.6.1 INTERRUPT CONTROL REGISTER (INT_OFF, R/W, ADDRESS = 0xf100_0500)

INT_OFF	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
INT_OFF	[0]	Interrupt control register. If Interrupt is disabled, then all done signal checking is by polling mode. 0 = interrupt enable 1 = interrupt disable	0

9.6.2 INTERRUPT LEVEL SELECTION CONTROL REGISTER (INT_PULSE_SEL, R/W, ADDRESS = 0xf100_0504)

INT_PULSE_SEL	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
INT_PULSE_SEL	[0]	0 = Level Interrupt 1 = Pulse Interrupt	0

9.6.3 INTERRUPT CLEAR REGISTER (INT_DONE_CLEAR, R/W, ADDRESS = 0xf100_0508)

INT_DONE_CLEAR	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
INT_DONE_CLEAR	[0]	Interrupt clear when INT_MODE is 0 and INT_OFF is 0.	0

9.6.4 OPERATION STATUS REGISTER (OPERATION_DONE, R, ADDRESS = 0xf100_050c)

OPERATION_DONE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
OPERATION_DONE	[0]	MFC done check signal, use polling mode. 0 = MFC encoder/decoder running. 1 = MFC encoder/decoder done.	0

9.6.5 FIRMWARE STATUS REGISTER (FW_DONE, R, ADDRESS = 0xf100_0510)

FW_DONE	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
FW_DONE	[0]	F/W done signal check register. Polling mode use. 0 = F/W running. 1 = F/W done.	0

9.6.6 INTERRUPT STATUS REGISTER (INT_STATUS, R, ADDRESS = 0xf100_0514)

INT_STATUS	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
FRAME_DONE_STAT	[8]	Frame done status	0
DMA_DONE_STAT	[7]	DMA done status	0
Reserved	[6]	Reserved	0
FW_DONE_STAT-	[5]	FW done status	0
Reserved	[4:0]	Reserved	0

9.6.7 INTERRUPT ENABLE REGISTER (INT_ENABLE_REG, R/W, ADDRESS = 0xf100_0518)

INT_MASK_REG	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
FRAME_DONE_ENABLE	[8]	Frame done enable	0
DMA_DONE_ ENABLE	[7]	DMA done enable	0
Reserved	[6]	Reserved	0
FW_DONE_ ENABLE	[5]	F/W done enable	0
Reserved	[4:0]	Reserved	0

9.6.8 INTERRUPT CONTROL

To a host processor

The FIMV-MFCV4.0 codec processor can generate an interrupt request to a host processor. Basically, this interrupt is used to indicate completion of encoding or decoding of a frame. The interrupt signal, IREQ, is active HIGH and is retained till the host processor clears it by writing '1' to interrupt clear register of the host interface.

As interrupt sources, there're 3 signals.

1. Frame done signal - means the end of frame's encoding/decoding.
2. DMA done signal - means the end of DMA's whole job in VSP during a job of loading initial F/W.
3. F/W done signal - means that F/W's job ends completely.

From a host processor

A host processor can control interrupt control using mask register.

9.7 MEMORY CONTROLLER SETTING REGISTER

9.7.1 MEMORY STRUCTURE SETTING REGISTER (MEM_STRUCT_SET, R/W, ADDRESS = 0xf100_0600)

MEM_STRUCT_SET	Bit	Description	Reset Value
Reserved	[31: 2]	Reserved	0
TILE_MODE	[1:0]	Memory structure of macroblock 0 = Linear mode. 2 = 16x16 tile mode for encoder 3 = 64x32 tile mode for decoder Encoder has to set 2 only when encoder wants to set tile mode. Decoder has to set 3 only when decoder wants to set tile mode.	0

Tile mode memory size calculation fomular.

One frame size can be calculated using below formular.

```
if (luma) {
  pixel_x = hor_img_size+hor_img_offset (ex: 1280)
  pixel_y = ver_img_size+ver_img_offset (ex: 720)
}
else {//chorma
  pixel_x = hor_img_size+hor_img_offset (ex: 1280)
  pixel_y = (ver_img_size+ver_img_offset)/2 (ex: 360)
}
pixel_x_minus = pixel_x - 1;
pixel_y_minus = pixel_y - 1;
roundup_x = INT (INT ((pixel_x - 1)/16)/8) + 1;
roundup_y = INT (INT ((pixel_y - 1)/16)/4) + 1;

if (pixel_y_minus[5] == 0)
 pic_range = pixel_y_minus [14:6] * roundup_x + pixel_x_minus [14:8] + 1
```

else

pic_range = roundup_x * roundup_y

The DPB start address is generated by summation of base address and picture range:

DPB_addr_N=base_addr+ (pic_range*N) <<13

N: number of frame

DPB_addrN: N-th frame start address in DPB buffer

IMG_OFFSET: This vaule is added value for making multiple by 16

(ex. IMG_OFFSET is 3, when IMG_SIZE is 77)

Ex) DPB_addr_0 = base_addr, DPB_addr_1 = base_addr + pic_range<<13

Table 9.11-3 Example of Pic_range

image_size	pic_range
1280x720	115
720x480	45
352x288	14
176x144	5

NOTE1: example address DPB_addr_0 [12:0] (Base_addr) must be set to 0. i.e All base address lower 13 bit must be set to '0' Value

NOTE2: At this case, picure vertical size is under 64 pixels, Chrominance picture range is same with luminance picture range.

9.8 ENCODER ADDRESS REGISTER

9.8.1 ENCODER CURRENT Y BUFFER ADDRESS REGISTER (ENC_CUR_Y_ADDR, R/W, ADDRESS = 0xf100_0800)

ENC_CUR_Y_ADDR	Bit	Description	Reset Value
ENC_CUR_Y_ADDR	[31:0]	External memory setting address. Current Y image start address	0

9.8.2 ENCODER CURRENT C BUFFER ADDRESS REGISTER (ENC_CUR_C_ADDR, R/W, ADDRESS = 0xf100_0804)

ENC_CUR_C_ADDR	Bit	Description	Reset Value
ENC_CUR_ C _ADDR	[31:0]	External memory setting address. Current Cb / Cr image start address. This chrominance format is 2-plane interleaving mode.	0

9.8.3 ENCODER DPB START ADDRESS REGISTER (ENC_DPB_ADDR, R/W, ADDRESS = 0xf100_080c)

ENC_DPB_ADDR	Bit	Description	Reset Value
ENC_DPB_ADDR	[31:0]	DPB start address. Encoder DPB area is two frame data size. Frame size calcultion methods have to confer MEM_STRUCT_SET register.	0

9.8.4 INTRA REFRESH MACROBLOCK NUMBER REGISTER (CIR_MB_NUM, R/W, ADDRESS = 0xf100_0810)

CIR_MB_NUM	Bit	Description	Reset Value
Reserved	[31:12]	Reserved	0
CIR_MB_NUM	[11:0]	Intra refresh macroblock number In multi-channel encoding case, CIR_MB_NUM must be set to 0	0

9.8.5 CYCLIC INTRA REFRESH

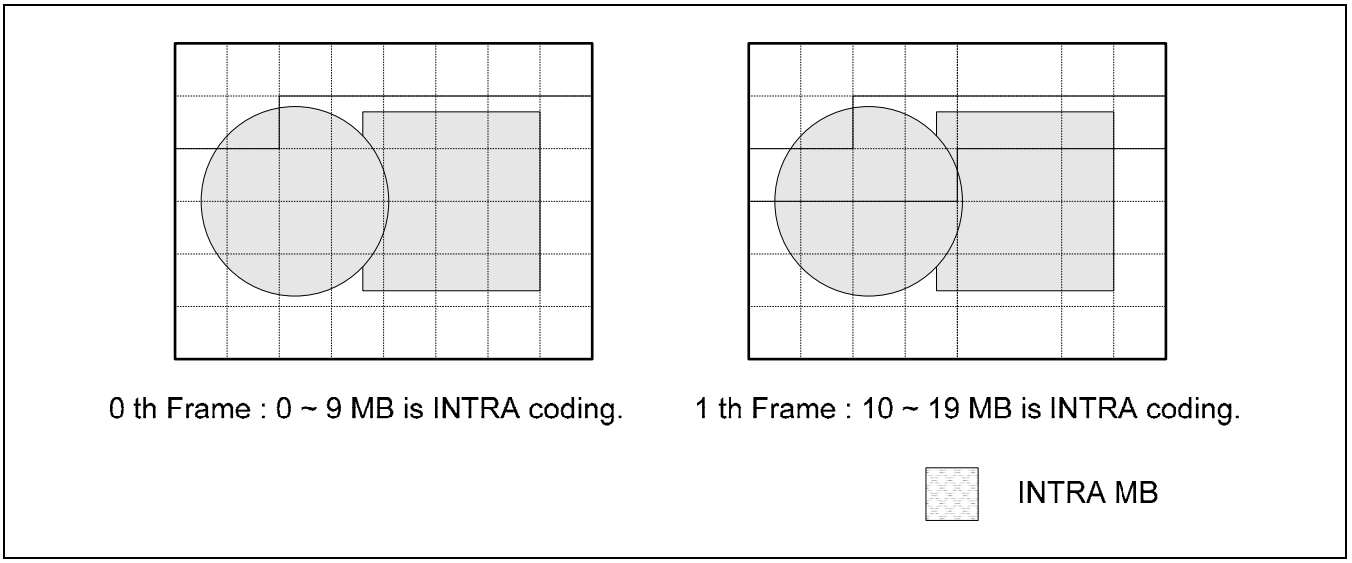


Figure 9.11-16 Cyclic Intra Refresh (CIR) CIR_MB_NUM = 10 case)

The Cyclic Intra Refresh (CIR) technique basically consists of cyclically refreshing the whole image to avoid the propagation of errors for too long a time. This is done by intra coding a few macroblocks per frame.

In CIR, for a macroblock to be considered refreshed, it has to be intra coded a predefined number of macroblock. Thus, this technique has one control parameters of macroblck numbers. In Figure 9.11-16 that case macrblock number is 10.

Cyclic Intra refresh (CIR) is set by external host at the first Special Function Register (SFR) setting. CIR value is used in order to decide macroblock mode.

9.9 DECODER ADDRESS REGISTER

9.9.1 DECODER DECODED PICTURE BUFFER ADDRESS REGISTER (DEC_DPB_ADDR, R/W, ADDRESS = 0xf100_0900)

DEC_DPB_ADDR	Bit	Description	Reset Value
DEC_DPB_ADDR	[31:0]	External address of DPB start address. <4:2:0 format>. Below area need to set. Frame data size is depending on tile mode or not. Frame size calcultion methods have to confer MEM_STRUCT_SET register. **NOTE:** Make reference of DPB buffer size below value. H.264 decoder: DPB_SIZE (0x90C register) * one frame data size. MPEG4 decoder : 2 frame MPEG2 decoder : 2 frame H.263 decoder : 2 frame VC-1 decoder : 4 frame	0

9.9.2 COLOCATED MOTION VECTOR BUFFER ADDRESS REGISTER (DPB_COMV_ADDR, R/W, ADDRESS = 0xf100_0904)

DPB_COMV_ADDR	Bit	Description	Reset Value
DPB_COMV_ADDR	[31:0]	Exteranl temporal address DPB Colocated MV saving buffer start address. **NOTE:** Make reference of DPB_COMV_ADDR buffer size below value. H.264 decoder: DPB_Colocated_buffer_size = **128 byte x ((hor_img_size+hor_img_offset)/16) x** (((ver_img_size+ver_img_offset)/16) + 1) * **DPB_SIZE** MPEG4, MPEG2, H.263 decoder: DPB_Colocated_buffer_size = **128 byte x ((hor_img_size+hor_img_offset)/16) x** (((ver_img_size+ver_img_offset)/16) + 1) * **2** VC-1 decoder: DPB_Colocated_buffer_size = **32 byte x ((hor_img_size+hor_img_offset)/16) x** (((ver_img_size+ver_img_offset)/16) + 1) * **4** ***IMG_OFFSET: This vaule is added value for making multiple by 16*** ***(ex. IMG_OFFSET is 3, when IMG_SIZE is 77)***	0

9.9.3 POST FILTER OUTPUT ADDRESS SETTING REGISTER (POST_OUT_ADDR, R/W, ADDRESS = 0xf100_0908)

POST_OUT_ADDR	Bit	Description	Reset Value
POST_OUT_ADDR	[31:0]	Post filter output start address. **NOTE:** H.264: No use this register. uses DPB buffer for Display. VC1 : No use this register. uses DPB buffer for Display. H.263: No use this register. uses DPB buffer for Display. MPEG4, MPEG2: 3 frame buffer sizes	0

9.9.4 DECODED PICTURE BUFFER SIZE RETURN VALUE SETTING REGISTER (DPB_SIZE, R, ADDRESS = 0xf100_090c)

DPB_SIZE	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
DPB_SIZE	[7:0]	MFC indicates to host about decoded picture buffer number. This value can be uesd at H.264 decoing mode.	0

9.10 RATE CONTROL REGISTER

9.10.1 RATE CONTROL CONFIGURATION REGISTER (RC_CONFIG, R/W, ADDRESS = 0xf100_0a00)

RC_CONFIG	Bits	Description	Reset Value
Reserved	[31:10]	Reserved	0
FR_RC_EN	[9]	Frame level rate control enable 0 = Disable Frame level rate control by F/W 1 = Enable Frame level rate control by F/W	0
MB_RC_EN	[8]	Macroblock level rate control enable 0 = Disable MB level rate control by H/W 1 = Enable MB level rate control by H/W	0
Reserved	[7:6]	Reserved	0
FRAME_QP	[5:0]	Frame quantization parameter. This value is in the range of 0 to 51. **Case RC_CONFIG [9:8] is 2'b00.** Constant quantization parameter (QP) is applied to all macroblocks in the current picture. **Case RC_CONFIG [9:8] is 2'b01.** Quantization parameter (QP) of the first macroblock in the current frame. The QP of the next macroblocks can be obtained by macroblock adaptive scaling. **Case RC_CONFIG [9:8] is 2'b10.** FRAME_QP can be changed by the difference between the numbers of target bit and generated bit during the encoding a picture. But macroblock adaptive scaling is not applied. **Case RC_CONFIG [9:8] is 2'b11.** Quantization parameter (QP) of the first macroblock in the current frame. FRAME_QP can be changed by the difference between the numbers of target bit and generated bit during the encoding a picture. The QP of the next macroblocks can be obtained by macroblock adaptive scaling	0

9.10.2 FRAME RATE REGISTER (RC_FRAME_RATE, R/W, ADDRESS = 0xf100_0a04)

RC_FRAME_RATE	Bits	Description	Reset Value
Reserved	[31:8]	Reserved	0
FRAME_RATE	[7:0]	Frames per second. '0' is forbidden. **NOTE:** Valid only when Frame level RC is enabled.	0

9.10.3 BIT RATE REGISTER (RC_BIT_RATE, R/W, ADDRESS = 0xf100_0a08)

RC_BIT_RATE	Bits	Description	Reset Value
BIT_RATE	[31:0]	Bits per second. '0' is forbidden. **NOTE:** Valid only when Frame level RC is enabled.	0

9.10.4 QUANTIZER PARAMETER BOUNDARY REGISTER (RC_QBOUND, R/W, ADDRESS = 0xf100_0a0c)

RC_QBOUND	Bits	Description	Reset Value
Reserved	[31:14]	Reserved	0
MAX_QP	[13:8]	Maximum quantization parameter (0~51 at H.264 encoding, 1~31 at H.263, MPEG4 encoding).	0
Reserved	[7:6]	Reserved	0
MIN_QP	[5:0]	Minimum quantization parameter (0~51 at H.264 encoding, 1~31 at H.263, MPEG4 encoding).	0

NOTE: For the unbounded range of quantization parameter, you have to set this register to 0x0000_3300 (e.g. MAX_QP = 51, MIN_QP = 0).

NOTE: MAX_QP must be greater than or equal to MIN_QP.

9.10.5 REACTION COEFFICIENT REGISTER (RC_RPARA, R/W, ADDRESS = 0xf100_0a10)

RC_RPARA	Bits	Description	Reset Value
Reserved	[31:16]	Reserved	0
REACT_PARA	[15:0]	Rate control reaction coefficient. '0' is forbidden. **NOTE:** Valid only when the Frame level RC is enabled. **NOTE:** For tight CBR, this field must be small. (Ex.2 ~10). For VBR, this field must be large. (Ex.100 ~1000). **NOTE:** It is not recommended to use the greater number than FRAME_RATE * (10^9 / BIT_RATE).	0

9.10.6 ACROBLOCK LEVEL RATE CONTROL REGISTER (RC_MB_CTRL, R/W, ADDRESS = 0xf100_0a14)

RC_MB_CTRL	Bits	Description	Reset Value
Reserved	[31:4]	Reserved	0
DARK_DISABLE	[3]	Disable Dark Region Adaptive feature. 0 = Enable Dark Region Adaptive feature. QP of dark MB may not be smaller than frame QP Although it is smooth, static or small activity image. 1 = Disable Dark Region Adaptive feature.	0
SMOOTH_DISABLE	[2]	Disable Smooth Region Adaptive feature. 0 = Enable Smooth Region Adaptive feature. QP of smooth MB may be smaller than frame QP. 1 = Disable Smooth Region Adaptive feature.	0
STATIC_DISABLE	[1]	Disable Static Region Adaptive feature. 0 = Enable Static Region Adaptive feature. QP of static MB may be smaller than frame QP. 1 = Disable Static Region Adaptive feature.	0
ACT_DISABLE	[0]	Disable MB Activity Adaptive feature. 0 = Enable MB Activity Adaptive feature. QP of MB that has small activity may be smaller than frame QP and QP of MB that has large activity may be larger than frame QP 1 = Disable MB Activity Adaptive feature.	0

NOTE: Valid only when the Macroblock level RC is enabled.

9.10.7 INTERNAL REFERENCE QUANTIZATION SCALE REGISTER (RC_QOUT, R, ADDRESS = 0xf100_0a18)

RC_QOUT	Bits	Description	Reset Value
Reserved	[31:6]	Reserved	0
QP_OUT	[5:0]	Final frame quantization parameter (FRAME_QP) referenced for encoding the last macroblock of the current picture.	0

NOTE: QP_OUT can be referenced for the frame quantization parameter (FRAME_QP) of the next frame.

9.11 MFCV VERSION REGISTER

9.11.1 MFC4 VERSION (MFC4_VERSION, R, ADDRESS = 0xf100_0b00)

MFC4_VERSION	Bits	Description	Reset Value
MFC_VERSION	[31:0]	MFC codec version checking register	0x0000_0040

9.12 CROPPING INFORMATION REGISTER FOR H.264 DECODING

9.12.1 MFC4 Cropping Information Register One (MFC_CROP1, R, Address = 0xf100_0c00)

MFC_CROP1	Bits	Description	Reset Value
CROP_RIGHT_OFFSET	[31:16]	Cropping right offset information	0
CROP_LEFT_OFFSET	[15:0]	Cropping left offset information	0

NOTE: When DISPLAY_STATUS [3] is '1'. It is valid.

9.12.2 MFC4 CROPPING INFORMATION REGISTER TWO (MFC_CROP2, R, ADDRESS = 0xf100_0c04)

MFC_CROP1	Bits	Description	Reset Value
CROP_BOTTOM_OFFSET	[31:16]	Cropping bottom offset information	0
CROP_TOP_OFFSET	[15:0]	Cropping top offset information	0

NOTE: When DISPLAY_STATUS [3] is '1'. It is valid.

9.12.3 MFC4 Decoded Frame Size (MFC_DEC_FRM_SIZE, R, ADDRESS = 0xf100_0c08)

MFC_CROP1	Bits	Description	Reset Value
MFC_DEC_FRAME_SIZE	[31:0]	Decoded frame size for Divx, MPEG4, Xvid	0

9.13 MULTI-CHANNEL CONTROL REGISTER

9.13.1 COMMAND TYPE CONTROL REGISTER (MFC_COMMAND_TYPE, R/W, ADDRESS = 0xf100_0d00)

MFC_COMMAND_TYPE	Bits	Description	Reset Value
Reserved	[31:4]	Reserved	0
COMMAND_TYPE	[3:0]	Normal operation command: 0 = CHANNEL_SET 2 = CHANNEL_END 3 = INIT_CODEC 4 = FRAME_RUN Power mode command : 6 = SLEEP 7 = WAKEUP	0

9.14 CUSTOM CONTROL REGISTER

9.14.1 FRAME TYPE REGISTER (FRAME_TYPE, R, ADDRESS = 0xf100_0c0c)

FRAME_TYPE	Bits	Description	Reset Value
Reserved	[31:2]	Reserved	0
FRAME_TYPE	[1:0]	<Encoding> 0: Not coded frame 1: I frame 2: P frame <Decoding> 0: Not coded frame 1: I frame 2: P frame 3: B frame	0

9.14.2 Number of extra dpb register (NUM_EXTRA_DPB, W, Address = 0xf100_0d04)

NUM_EXTRA_DPB	Bits	Description	Reset Value
DISPLAY_DELAY	[31:16]	Number of frames for display delay. MFC is forced to return frames for display even if DPB is not filled. It is valid for H.264 decoder only	0
NUM_EXTRA_DPB	[15:0]	Number of extra DPB that host prepared for decoding in addition to the minimum DPB. Total number of DPB is a sum of the minimum DPB and NUM_EXTRA_DPB	0

9.14.3 Codec Command Register (CODEC_COMMAND, W, Address = 0xf100_0d08)

CODEC_COMMAND	Bits	Description	Reset Value
Reserved	[31:3]	Reserved	0
DISABLE_REORDERING	[2]	Disable reordering in H.264 decoding. When reordering is disabled, display order is the same as the decode order regardless of POC type. The decoded frame can be displayed immediately and therefore DISPLAY_STATUS[1:0]=1 always. 0: Normal operation 1: Disable reordering for immediate display	0
INSERT_NOTCODED_FRAME	[1]	A not coded frame is inserted in the middle of MPEG4 encoding. 0: Normal operation. 1: Not coded frame insertion.	0
INSERT_I_FRAME	[0]	An I frame is inserted in the middle of an encoding session without closing and reopening a channel 0: Normal operation 1: I frame Insertion	0

10.1 AUDIO SUB-SYSTEM

1 OVERVIEW

The Audio Sub-system is specially desinged to increase audio-playback time. The Audio Sub-system has an AHB bus component, an I2S controller and an SRAM wrapper. The SRAM wrapper uses the RAM of CAMIF0 and CAMIF1 as a buffer for sound data.

To save power efficiently, CPU saves sound data to the SRAM wrapper. Then, all the other power domains are turned off while the Audio Sub-system plays the data.

2 BLOCK DIAGRAM

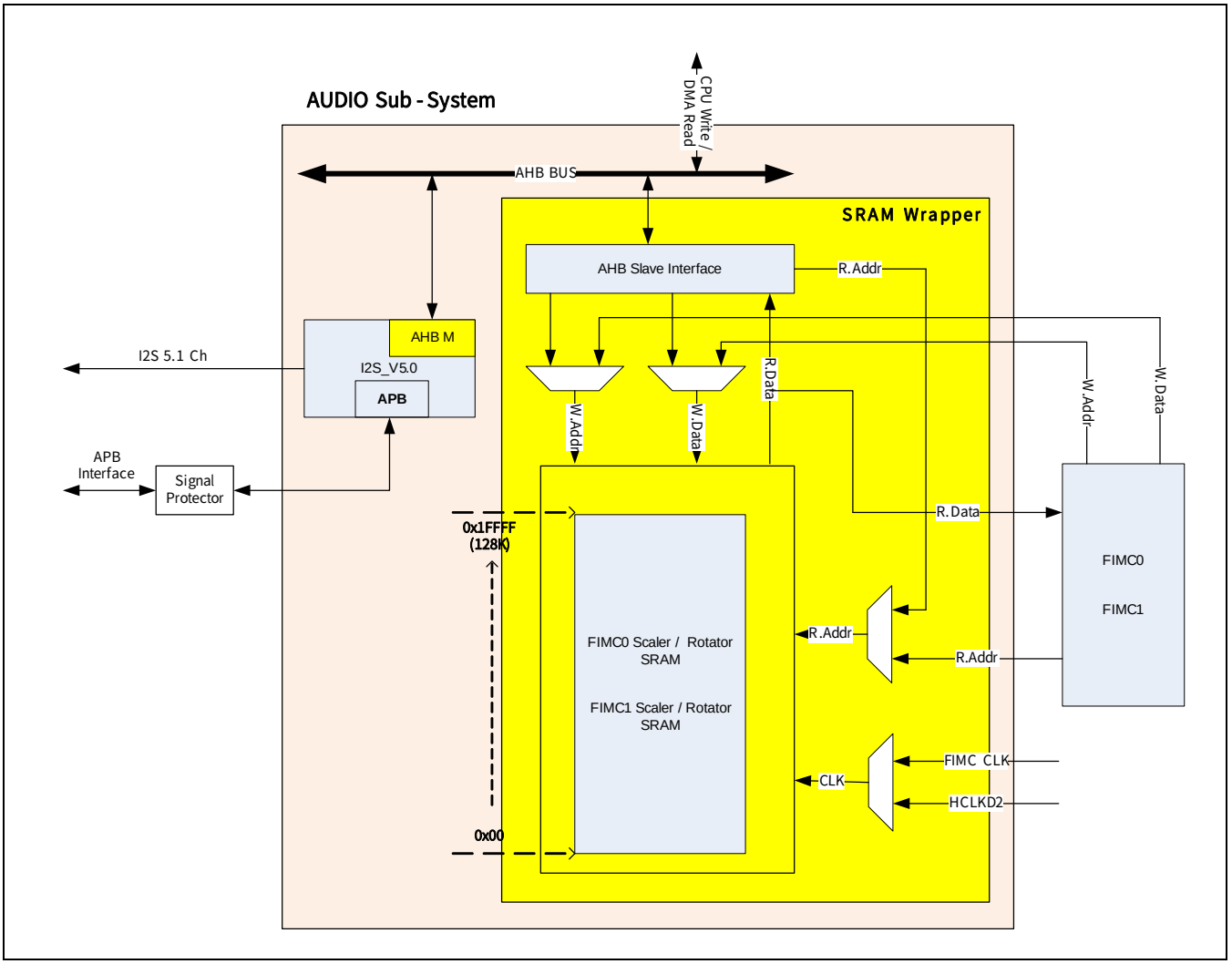


Figure 10.1-1 Audio Sub-system Block Diagram

3 FUNCTIONAL DESCRIPTIONS

3.1 I2S

The major difference between the I2S of the Audio-Subsystem and the other I2S's is the presence of a master port. The I2S of the Audio-Subsystem gets audio data by itself whenever the internal FIFO in the I2S lacks data. The I2S issues only 32-bit single read transaction.

The I2S has SFR interface on the APB bus. If S5PC100x enters into the Deep IDLE Mode, the APB interface signals may be unknown state. To prevent this situation, S5PC100x has signal protectors on the APB interface signals as shown in Figure 10.1-1.

3.2 SRAM WRAPPER

The less CPU wakes up, the more power it saves. To achieve that goal – "the less CPU wakes up", I2S should have very large buffers. But considering the size and cost, Audio sub-system reuses SRAMs controlled by other IPs powered off in low-power MP3 mode, instead of enlarging I2S internal buffer. In S5PC100x, to make 128 KB buffer for I2S in low-power MP3 mode, SRAM wrapper module has rotation SRAM buffers in CAMIF0 and CAMIF1. The address range on SRAM related to CAMIF is 0xC000_0000 – 0xC000_1FFF if LPMP3_MODE_SEL bit in Clock Controller module is on.

SRAM wrapper also has AHB slave interface. AHB slave interface handles single read transactions from I2S, multi-burst write transactions from DMA, and single or burst two write transactions (using incremental burst option) from CPU.

When the CPU reads from the SRAM wrapper, the latency will be 5 bus cycles. Data can be only read by word unit and burst tranfser is not supported.

When the CPU writes to the SRAM wrapper, there is no latency because the data are pipelined internally. Data can be written by word unit and it supports upto 8-burst tranfser.

4 SCENARIO EXAMPLE

1. Setup

If system has a plan to use Deep IDLE mode for low-power audio play, proper configurations should be needed according to system. For example, I2S0_FIFO_CLK_SEL bit in Clock Controller must be set.

2. Normal Mode

CPU writes decoded MP3 data to external DDR memory.

External DMA outside Audio sub-system transfers data from external DDR memory to I2S in case of I2S request.

3. Enter to Deep IDLE Mode from Normal Mode

Wait for external DMA operation to complete.

After end of external DMA operation, programmers can know where the next data to be sent is.

Assert LPMP3_MODE_SEL bit in Clock Controller module. This operation changes mux interface in SRAM wrapper (Refer to Figure 10.1-1). Now, SRAM wrapper switches rotation buffers in CAMIF to temporary sound buffers.

CPU can transfer decoded MP3 data to SRAM wrapper in Audio sub-system.

If the proper size of MP3 decoded data is in SRAM wrapper, internal DMA in I2S is turned on to supply decoded MP3 data from SRAM wrapper to I2S internal FIFO.

After CPU transfers all the decoded MP3 data to SRAM wrapper, S5PC100 enters to Deep IDLE mode.

4. Exit from Deep IDLE Mode

Using the pre-defined address based on interrupt condition like level interrupt or transfer-done interrupt, external DMA is configured.

After external DMA is on, LPMP3_MODE_SEL bit should be low for I2S to communicate with external DMA. This operation changed mux interface in SRAM wrapper. Now, SRAMs in SRAM wrapper are returned to CAMIF.

10.2 I2S CONTROLLER(5.1CH)

1 OVERVIEW

Inter-IC Sound (I2S) is one of the popular digital audio interfaces. The bus handles audio data, while the other signals, such as sub-coding and control, is transferred separately. It is possible to transmit data between two I2S bus. To minimize the number of pins required and to keep wiring simple, basically, a 3-line serial bus is used. It consists of a line for two time-multiplexed data channels, a word select line and a clock line.

I2S interface transmits or receives sound data from external stereo audio codec. To transmit or receive data, two 32-bitsx64 First-In-First-Out (FIFO) data structures are included and DMA transfer mode to transmit or receive samples is supported. I2S-specific clock is supplied from internal system clock controller through I2S clock divider or direct clock source.

I2S V5.0 (I2S0) handles up to 2 sound sources. For example, default system sound using OS s/w sound mixer is delivered to primary sound path and special application bypassing OS s/w sound mixer to secondary sound path. I2S V5.0 mixes primary sound source and secondary sound source.

2 FEATURE

- Mixes up to 2 sound sources: Primary sound source and Secondary sound source.
- Primary sound source drives up to 5.1ch I2S-bus for audio interface with external DMA-based operation
- Secondary sound source supports stereo sound channels with internal DMA
- Serial, 8/ 16/ 24-bit per channel data transfers
- Supports I2S, MSB-justified and LSB-justified data format
- I2S v5.0 (I2S0) interrupt wakes-up system from IDLE and DEEP_IDLE mode.

3 BLOCK DIAGRAM

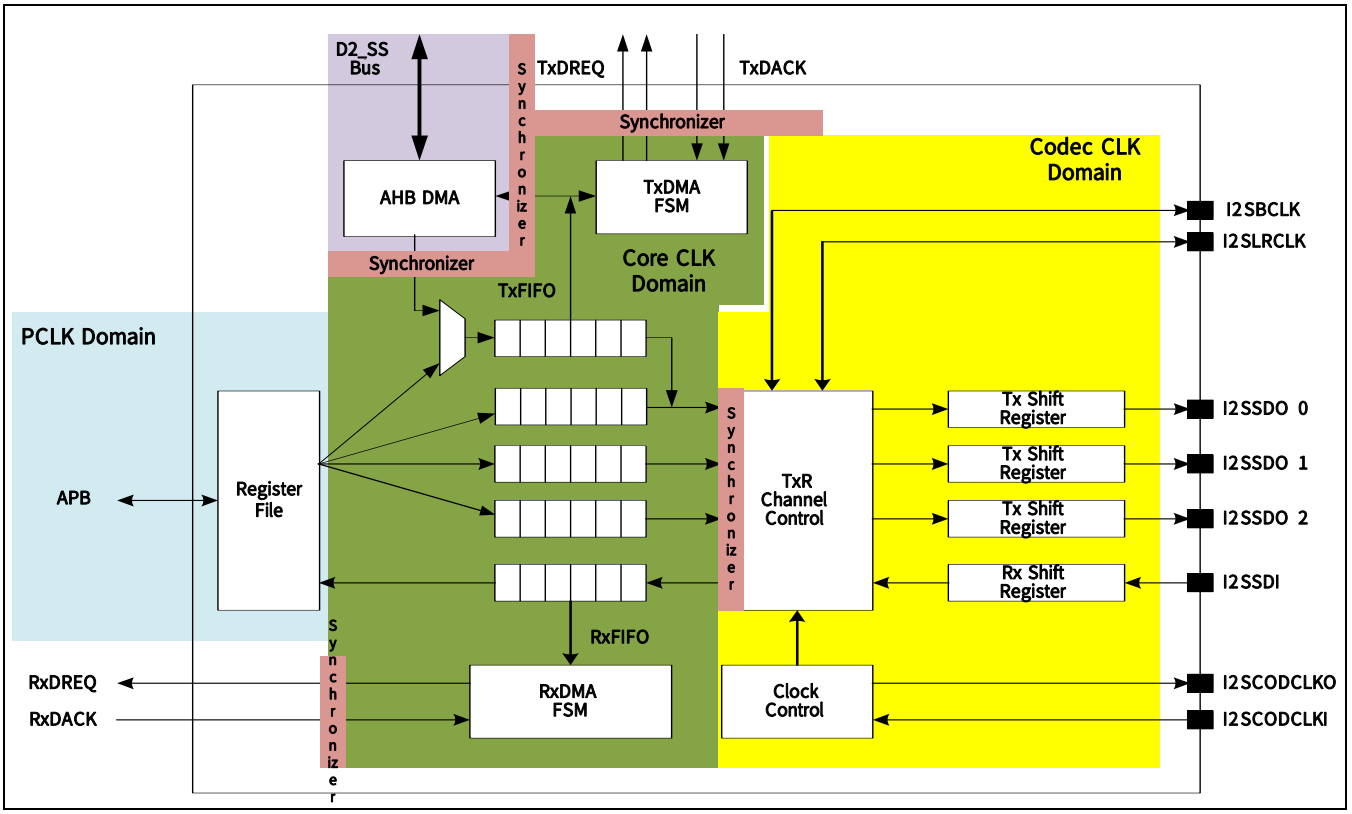


Figure 10.2-1 I2S-Bus Block Diagram

4 FUNCTIONAL DESCRIPTIONS

I2S interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as shown in Figure 10.2-1. Note that each FIFO has 32-bit width * 64 depth structure, which contains left/ right channel data. Therefore, FIFO access and data transfer are handled with left/ right pair unit. Figure 10.2-1 shows the functional block diagram of I2S interface.

4.1 MASTER/ SLAVE MODE

Set IMS bit of I2SMOD register to select master or slave mode. In master mode, I2SSCLK and I2SLRCLK are generated internally and supplied to external device. Therefore a root clock is needed for generating I2SSCLK and I2SLRCLK. The I2S pre-scaler (clock divider) generates a root clock with divided frequency from internal system clock. In external master mode, the root clock is fed directly from external I2S. The I2SSCLK and I2SLRCLK are supplied from the pin (GPIOs) in slave mode.

Master/ Slave mode is different with TX/RX. Master/Slave mode presents the direction of I2SLRCLK and I2SSCLK. The direction of I2SCDCLK (This is only auxiliary) is not a concern. If I2S bus interface transmits clock signals to I2S codec, I2S bus is master mode. But if I2S bus interface receives clock signal from I2S codec, I2S bus is slave mode. TX/RX mode indicates the direction of data flow. If I2S bus interface transmits data to I2S codec, this is TX mode. Conversely, I2S bus interface receives data from I2S codec that is RX mode. Let us distinguish Master/ Slave mode from TX/RX mode.

Figure 10.2-2 & Figure 10.2-3 shows the route of the root clock with internal master or external master mode setting in I2S clock control block and system controller. Note that RCLK indicates root clock and this clock is supplied to external I2S CODEC chip at internal master mode. To configure I2S V5.0 to work properly, refer CLK_GATE_D2_0 in Clock Controller manual.

NOTE: CLK_GATE_D2_0[1:0] should be set to 2'b11.

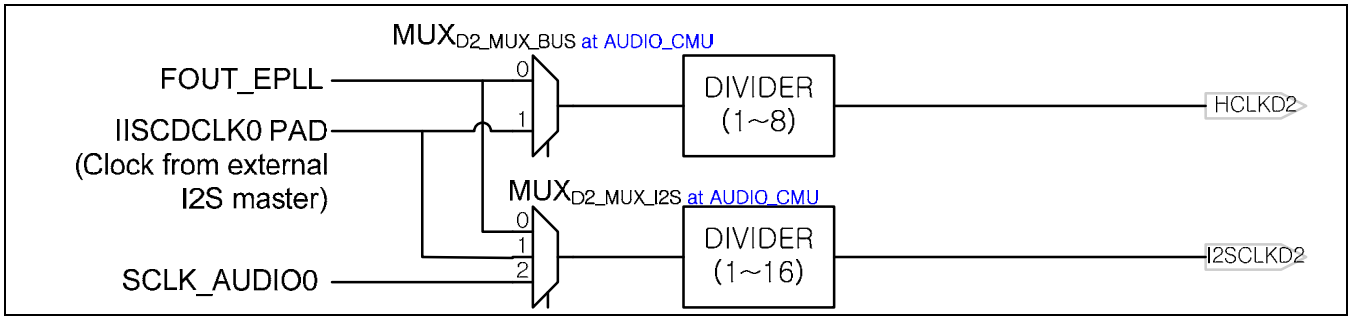


Figure 10.2-2 S5PC100x Audio_CMU

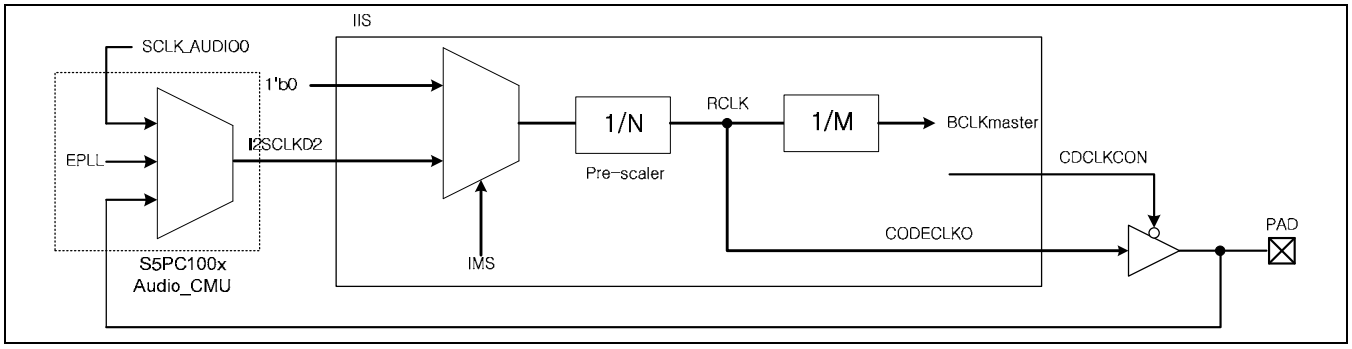


Figure 10.2-3 I2S Clock Control Block Diagram

4.1.1 External DMA Transfer

To transfer up to 5.1 channel primary sound from s/w mixer to I2S, external DMA or SFR interface are used. To play primary sound for 5.1 channels or record 2 channel sound, I2S has TXFIFO0, TXFIFO1, TXFIFO2, TXFIFO_S and RXFIFO. I2S mixes primary sound in TXFIFO0 and secondary sound in TXFIFO_S and output mixed sound stream to external codec logic.

In the external DMA transfer mode, the transmitter or receiver FIFO are accessible by external DMA controller. DMA service request is activated internally by the transmitter or receiver FIFO state. The FTXEMPT, FRXEMPT, FTXFULL, and FRXFULL bits of I2SCON register represent the transmitter or receiver FIFO data state. Especially, FTXEMPT and FRXFULL bit are the ready flag for DMA service request; the transmit DMA service request is activated if TXFIFO is not empty and the receiver DMA service request is activated if RXFIFO is not full.

The external DMA transfer uses only handshaking method for single data. Note that during external DMA acknowledge activation, the data read or write operation should be performed.

* Reference: DMA request point

- TX mode: (FIFO is not full) and (TXDMACTIVE is active)
- RX mode: (FIFO is not empty) and (RXDMACTIVE is active)

4.1.2 Internal DMA Transfer

To transfer up to 2 channel secondary sound to I2S, internal DMA or SFR interface are used. To play secondary sound for 2 channels, internal DMA in I2S gets sound data from address range between 0xC000_0000 and 0xC01F_FFFF to TXFIFO_S if LPMP3_MODE_SEL[0] bit is high. I2S mixes primary sound in TXFIFO0 and secondary sound in TXFIFO_S and output mixed sound stream to external codec logic.

Like external DMA transfer mode, in the internal DMA transfer mode, the internal DMA is activated if TXFIFO_S is not full. If activated, internal DMA runs according to SFR configurations and signals an interrupt after completion.

4.1.3 Sound Mixing

I2S mixes primary sound in TXFIFO0 and secondary sound in TXFIFO_S if two sound sourses have the same sampling rate and PCM format.

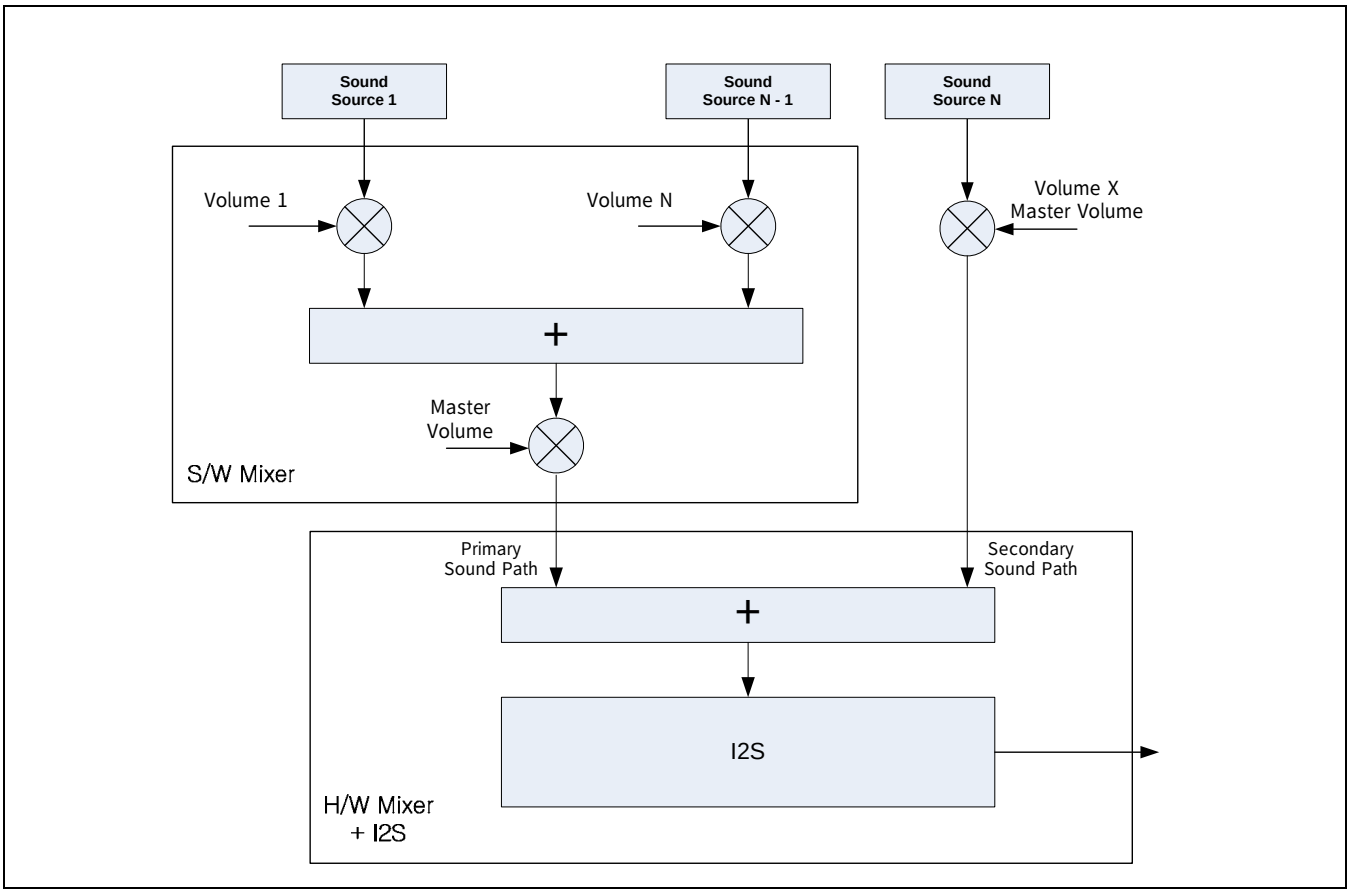


Figure 10.2-4 Concept of Mixer in I2S

This function has two limitations:

1. Normalization should be pre-processed in S/W configurations or settings.
2. Synchronization between two sound sources is not guaranteed.

5 AUDIO SERIAL DATA FORMAT

5.1 I2S-BUS FORMAT

The I2S bus has four lines including serial data input I2SSDI, serial data output I2SSDO, left/right channel select clock I2SLRCLK, and serial bit clock I2SBCLK; the device generating I2SLRCLK and I2SBCLK is the master.

Serial data is transmitted in 2's complement with the MSB first with a fixed position, whereas the position of the LSB depends on the word length. The transmitter sends the MSB of the next word at one clock period after the I2SLRCLK is changed. Serial data sent by the transmitter is synchronized with either the trailing or the leading edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and therefore there are some restrictions to transmit data that is synchronized with the leading edge.

The LR channel select line indicates the channel being transmitted. I2SLRCLK is changed either on a trailing or leading edge of the serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that is set up for transmission. Furthermore, it enables the receiver to store the previous word and clears the input for the next word.

5.2 MSB (LEFT) JUSTIFIED

MSB-Justified (Left-Justified) format is similar to I2S bus format, except that in MSB-justified format, the transmitter always sends the MSB of the next word at the same time whenever the I2SLRCLK is changed.

5.3 LSB (RIGHT) JUSTIFIED

LSB-Justified (Right-Justified) format is opposite to the MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Figure 10.2-5 shows the audio serial format of I2S, MSB-justified, and LSB-justified. Note that in this figure, the word length is 16-bit and I2SLRCLK makes transition every 24 cycle of I2SBCLK (BFS is 48 fs, where fs is sampling frequency; I2SLRCLK frequency).

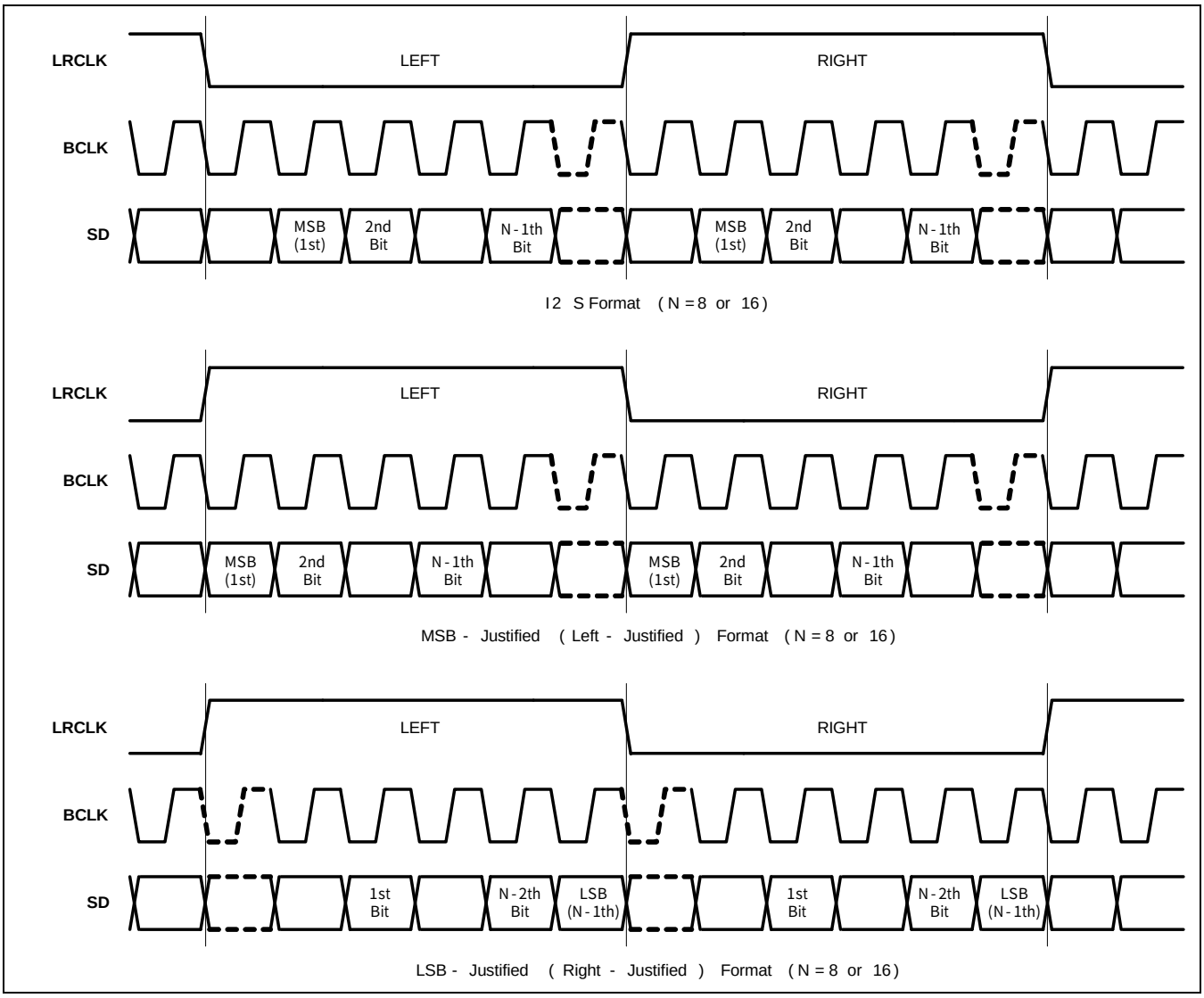


Figure 10.2-5 I2S Audio Serial Data Formats

5.4 SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (RCLK) is selected by sampling frequency as shown in Table 10.2-1. Because RCLK is made by I2S pre-scaler, the pre-scaler value and RCLK type (256fs or 384fs or 512fs or 768fs) should be determined properly.

Table 10.2-1 CODEC Clock (CODECLK = 256fs, 384fs, 512fs, 768fs)

I2SLRCK (fs)	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
CODE CLK (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
	512fs									
	4.0960	5.6448	8.1920	11.2900	16.3840	22.5790	24.5760	32.7680	45.1580	49.1520
	768fs									
	6.1440	8.4672	12.2880	16.9340	24.5760	33.8690	36.8640	49.1520	67.7380	73.7280

5.5 I2S CLOCK MAPPING TABLE

Before selecting BFS, RFS, and BLC bits of I2SMOD register, please refer to the Table 10.2-2. Table 10.2-2 shows the allowable clock frequency mapping relations.

Table 10.2-2 I2S Clock Mapping Table

Clock Frequency		RFS			
		256 fs (00B)	512 fs (01B)	384 fs (10B)	768 fs (11B)
BFS	16 fs (10B)	(a)	(a)	(a)	(a)
	24 fs (11B)	-	-	(a)	(a)
	32 fs (00B)	(a) (b)	(a) (b)	(a) (b)	(a) (b)
	48 fs (01B)	-	-	(a) (b) (c)	(a) (b) (c)
Descriptions		(a) Allowed if BLC is 8-bit (b) Allowed if BLC is 16-bit (c) Allowed if BLC is 24-bit			

6 PROGRAMMING GUIDE

The I2S bus interface is accessed either by the processor using programmed I/O instructions or by the DMA controller.

6.1 INITIALIZATION

1. Before you use I2S bus interface, you must configure GPIOs to I2S mode and check signal's direction. The I2SLRCLK, I2SSCLK and I2SCDCLK are inout-type. The each of I2SSDI and I2SSDO is input and output.
2. Select clock source. The S5PC100 has three clock sources namely PCLK, EPLL and external codec. For more details refer to Figure 10.2-2 and Figure 10.2-3.

6.2 PLAY MODE (TX MODE) WITH DMA

1. TXFIFO is flushed before operation. If you do not distinguish Master/ Slave mode from TX/RX mode, you must study Master/Slave mode and TX/RX mode. For more information refer to "Section 5.1 Master/ Slave".
2. To configure I2SMOD register and I2SPSR (I2S pre-scaler register) properly.
3. To operate system in stability, the internal TXFIFO should be almost full before transmission. Starts DMA to fill TXFIFO.
4. Basically, I2S bus does not support the interrupt. Therefore, you check state by polling through accessing SFR.
5. If TXFIFO is full, assert the I2SACTIVE.

6.3 RECORDING MODE (RX MODE) WITH DMA

1. RXFIFO is flushed before operation. If you do not distinguish between Master/ Slave mode and TX/RX mode, you must study Master/ Slave mode and TX/RX mode. For more information refer to "Section 5.1 Master/Slave.
2. To configure I2SMOD register and I2SPSR (I2S pre-scaler register) properly.
3. To operate system in stability, the internal RXFIFO should have minimum one data before DMA operation. Assert the I2SACTIVE.
4. Check RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, start RXDMACTIVE.

6.4 EXAMPLE CODE

TX CHANNEL

The I2S TX channel provides a single stereo compliant output. The transmit channel operates in master or slave mode. Data is transferred between the processor and the I2S controller via an APB access or a DMA access.
The processor must write words in multiples of two (i.e. for left and right audio sample). The words are serially shifted out timed with respect to the audio bitclk, BCLK and word select clock, LRCLK.

TX Channel has 64X32 bits wide FIFO where the processor or DMA writes up to 16 left/ right data samples after enabling the channel for transmission.

For Example:

Ensure the PCLK and CODCLKI are coming correctly to the I2S controller and FLUSH the TX FIFO using the TFLUSH bit in the I2SFIC Register (I2S FIFO Control Register).

Please ensure that I2S Controller is configured in one of the following modes.

- TX only mode
- TX/RX simultaneous mode

This is done by programming the TXR bit in the I2SMOD Register (I2S Mode Register).

1. Then Program the following parameters according to the need
   - ♦ IMS
   - ♦ SDF
   - ♦ BFS
   - ♦ BLC
   - ♦ LRP

   For Programming, the above-mentioned fields please refer to I2SMOD Register (I2S Mode Register).
2. After ensuring that the input clocks for I2S controller are up and running and step 1 and 2 have been completed, you can write to TX FIFO.

The write to the TX FIFO has to be carried out thorough the I2STXD Register (I2S TX FIFO Register).
This 32-bit data occupies position 0 of the FIFO and any further data is written to position 2, 3 and so on.

The Data is aligned in the TX FIFO for 8-bit/channel or 16-bit/channel BLC are shown in below figure.

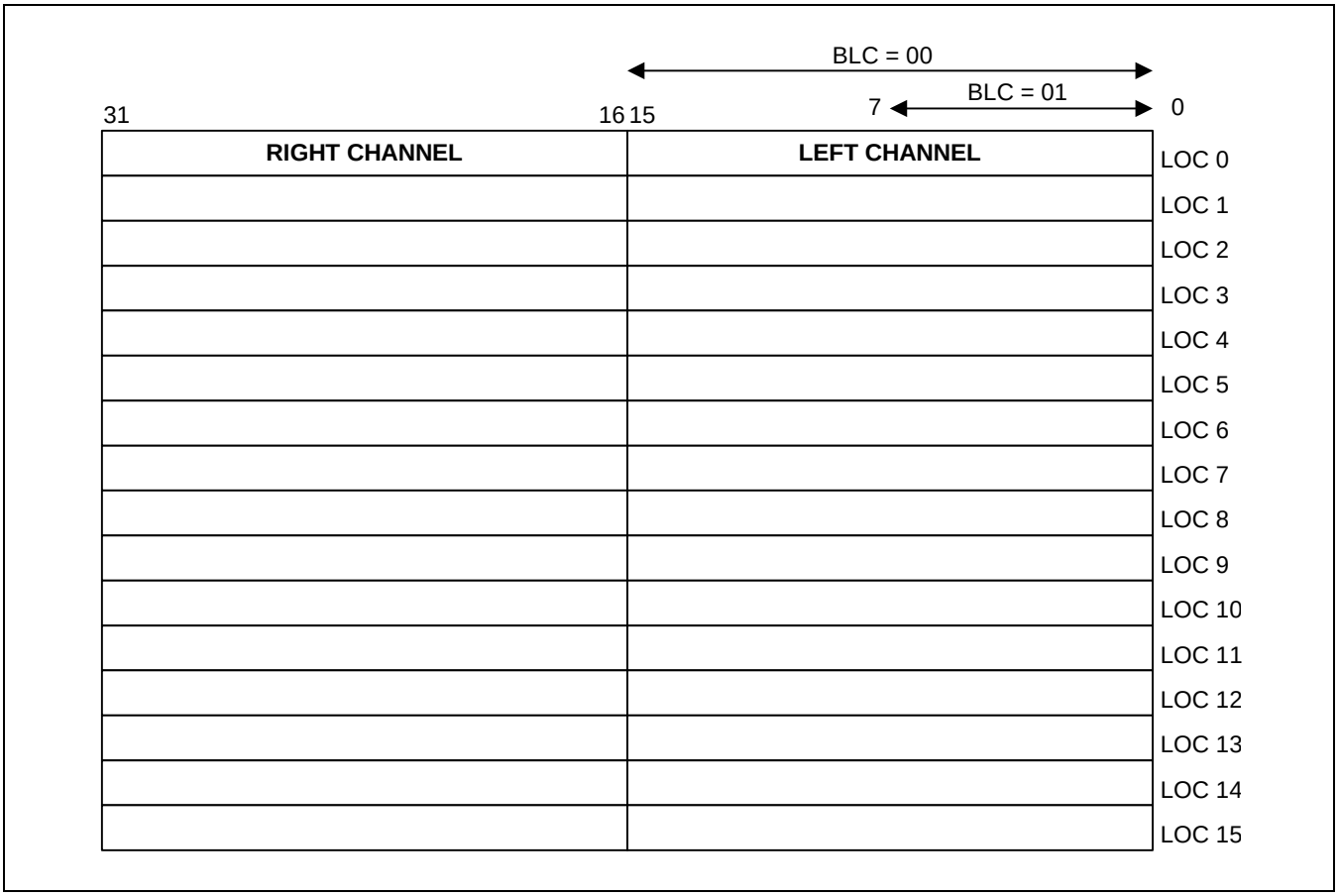


Figure 10.2-6 TX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the TX FIFO for 24-bit/channel BLC are shown in below figure.

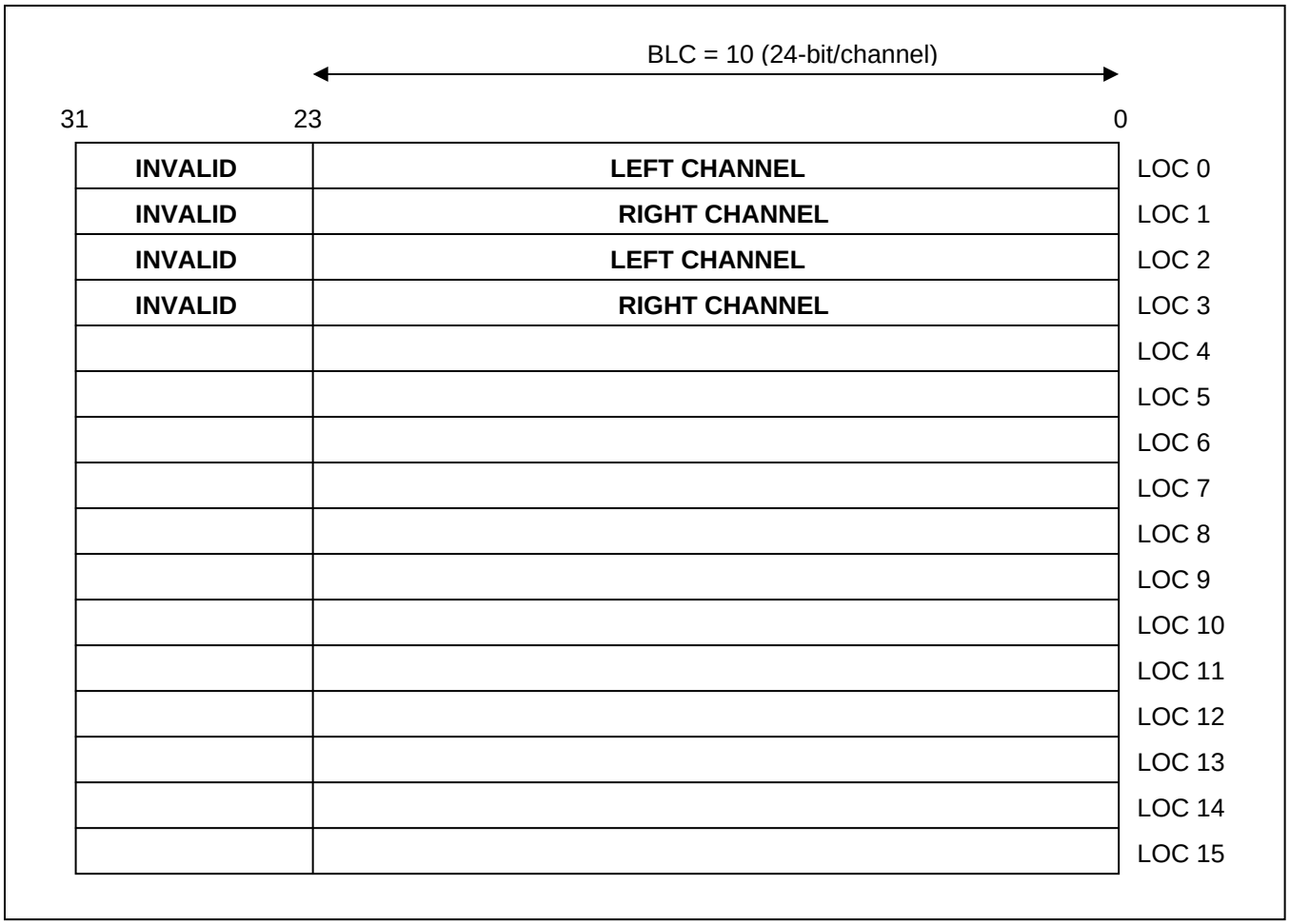


Figure 10.2-7 TX FIF0 Structure for BLC = 10 (24-bit/channel)

Once the data is written to the TX FIFO the TX channel is made active by enabling the I2SACTIVE bit in the I2SCON Register (I2S Control Register).
The data is then serially shifted out with respect to the bit clock BCLK and word select clock LRCLK.

The TXCHPAUSE in the I2SCON Register (I2S Control Register) stops the serial data transmission on the I2SSDO.The transmission is stopped once the current Left/ Right channel is transmitted.
If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the TX channel.

If the TX channel is enabled while the FIFO is empty, no samples are read from the FIFO.
The Status of TX FIFO can be checked by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

RX CHANNEL

The I2S RX channel provides a single stereo compliant output. The receive channel operates in master or slave mode. Data is received from the input line and transferred into the RX FIFO. Then processor reads this data via an APB read or a DMA access.

RX Channel has 64 X 32-bit wide RX FIFO where the processor or DMA is read up to 16 left/ right data samples after enabling the channel for reception.

For Example:

Ensure the PCLK and CODCLKI are coming correctly to the I2S controller. FLUSH the RX FIFO using the RFLUSH bit in the I2SFIC Register (I2S FIFO Control Register). The I2S controller is configured in any of the following modes

- Receive only.
- Receive/ Transmit simultaneous mode

This is done by Programming the TXR bit in the I2SMOD Register (I2S Mode Register)

1. Then Program the following parameters according to the need
   - ♦ IMS
   - ♦ SDF
   - ♦ BFS
   - ♦ BLC
   - ♦ LRP

   For Programming, the above mentioned fields please refer to I2SMOD Register (I2S Mode Register)

2. After ensuring that the input clocks for I2S controller are up and running and step 1 and 2 have been completed, you must put the I2SACTIVE high to enable any reception of data, the I2S Controller receives data on the LRCLK change.

The Data must be read from the RX FIFO using the I2SRXD Register (I2S RX FIFO Register) only after looking at the RX FIFO count in the I2SFIC Register (I2S FIFO Control Register). The count would only increment once the complete left channel and right have been received.

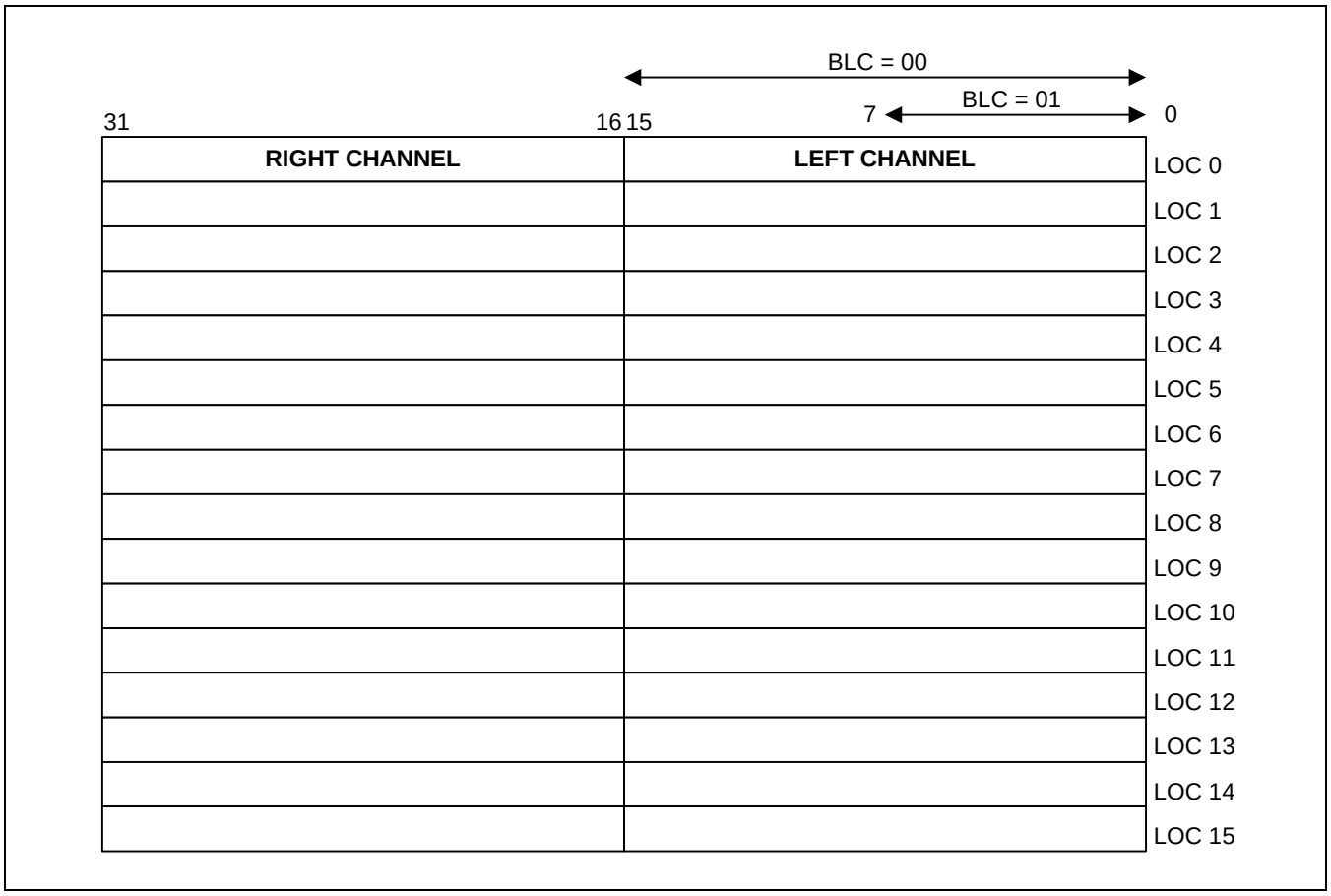


Figure 10.2-8 RX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the RX FIFO for 24-bit/ channel BLC is shown in below figure.

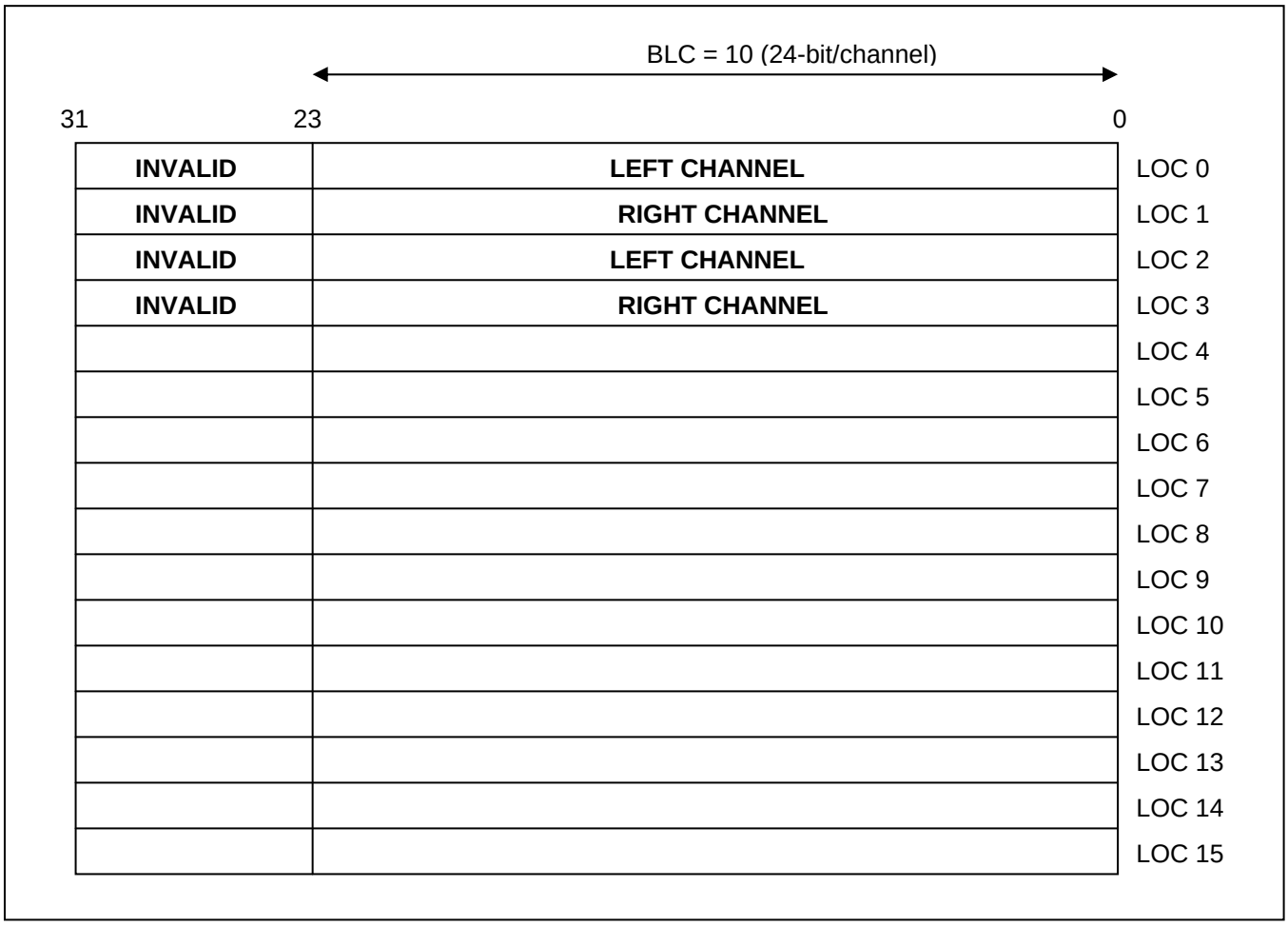


Figure 10.2-9 RX FIF0 Structure for BLC = 10 (24-bit/channel)

The RXCHPAUSE in the I2SCON register stops the serial data reception on the I2SSDI. The reception is stopped once the current Left/ Right channel is received.
If the control registers in the I2SCON Register (I2S Control Register) and I2SMOD Register (I2S Mode Register) are to be reprogrammed then it is advisable to disable the RX channel.

Check the Status of RX FIFO by checking the bits in the I2SFIC Register (I2S FIFO Control Register).

7 I/O DESCRIPTION

Funtion Signal	I/O	Description	Pad	Type
I2S0_SCLK	I/O	Bit Clock Input	Xi2s0SCLK	Dedicated
I2S0_LRCK	I/O	LR Channel Clock Input	Xi2s0LRCK	Dedicated
I2S0_CDCLK	I/O	Codec Clock Out	Xi2s0CDCLK	Dedicated
I2S0_SDI	I	I2S Serial Data Input	Xi2s0SDI	Dedicated
I2S0_SDO[0]	O	I2S Serial Data Out 0	Xi2s0SDO[0]	Dedicated
I2S0_SDO[1]	O	I2S Serial Data Out 1	Xi2s0SDO[1]	Dedicated
I2S0_SDO[2]	O	I2S Serial Data Out 2	Xi2s0SDO[2]	Dedicated

8 REGISTER DESCRIPTION

Table 10.2-3 Register Summary of I2S Interface

Register	Address	R/W	Description	Reset Value
I2SCON	0xF200_0000	R/W	I2S Interface Control Register	0x000
I2SMOD	0xF200_0004	R/W	I2S Interface Mode Register	0x0
I2SFIC	0xF200_0008	R/W	I2S Interface Primary Tx FIFO & Rx FIFO Control Register	0x0
I2SPSR	0xF200_000C	R/W	I2S Interface Clock Divider Control Register	0x0
I2STXD	0xF200_0010	W	I2S Interface Transmit Primary Sound Data Register	0x0
I2SRXD	0xF200_0014	R	I2S Interface Receive Data Register	0x0
I2SFICS	0xF200_0018	R/W	I2S Interface Secondary TXFIFO_S Control Register	0x0
I2STXDS	0xF200_001C	W	I2S Interface Secondary Transmit Data Register	0x0
I2SAHB	0xF200_0020	R/W	I2S AHB DMA Control Register	0x0
I2SSTR	0xF200_0024	R/W	I2S AHB DMA Start Address Register	0x0
I2SSIZE	0xF200_0028	R/W	I2S AHB DMA Size Register	0x8000_0000
I2STRNCNT	0xF200_002C	R	I2S AHB DMA Transfer Count Register	0x0
I2SLVL0ADDR	0xF200_0030	R/W	I2S AHA DMA Interrupt Level 0 Register	0x0000_0000
I2SLVL1ADDR	0xF200_0034	R/W	I2S AHA DMA Interrupt Level 1 Register	0x0000_0000
I2SLVL2ADDR	0xF200_0038	R/W	I2S AHA DMA Interrupt Level 2 Register	0x0000_0000
I2SLVL3ADDR	0xF200_003C	R/W	I2S AHA DMA Interrupt Level 3 Register	0x0000_0000

NOTE: All registers of I2S interface are accessible by word unit with STR/ LDR instructions.

8.1 I2S INTERFACE CONTROL REGISTER (I2SCON, R/W, ADDRESS = 0XF200_0000)

I2SCON	Bit	Description	R/W	Reset Value
SW_RST	[31]	I2S s/w reset control. This should be set to 1 after I2S clock is stable. 0 = Reset I2S module (default) 1 = Un-reset I2S module	R/W	0
Reserved	[30:27]	Reserved	R	0x0
FRXOFSTATUS	[26]	RX FIFO Over Flow Interrupt Status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing '1'. 0 = Interrupt did not occurred. 1 = Interrupt occurred.	R/W	0

I2SCON	Bit	Description	R/W	Reset Value
FRXORINTEN	[25]	RX FIFO OverFlow Interrupt Enable 0 = Disables RXFIFO Under-run INT 1 = Enables RXFIFO Under-run INT	R/W	0
FTXSUR STATUS	[24]	Secondary TX FIFO_S under-run interrupt status. This is used by interrupt clear bit. If this is high, write '1' to clear the interrupt. 0 = Interrupt did not occur. 1 = Interrupt occurred.	R/W	0
FTXSURINTEN	[23]	Secondary TX FIFO_S Under-run Interrupt Enable 0 = Disables TXFIFO_S Under-run INT 1 = Enables TXFIFO_S Under-run INT	R/W	0
FTXSEMPT	[22]	Secondary TX FIFO_S empty Status Indication 0 = TX FIFO_S is not empty (Ready to transmit Data) 1= TX FIFO_S is empty (Not Ready to transmit Data)	R	0
FTXSFULL	[21]	Secondary TX FIFO_S full Status Indication 0 = TX FIFO_S is not full 1 = TX FIFO_S is full	R	0
TXSDMAPAUSE	[20]	Tx DMA operation for secondary TX FIFIO_S pause command. **Note:** If this bit is activated at any time, the DMA request halts after current on-going DMA transfer is complete. 0 = No pause DMA operation for TX FIFO_S 1 = Pause DMA operation for TX FIFO_S	R/W	0
Reserved	[19]	Reserved	R	0
TXSDMACTIVE	[18]	Tx DMA active for secondary TX FIFO_S (start DMA request). **Note:** If this bit is set from high to low, the DMA operation is forced to stop immediately. 0 = Inactive, 1 = Active	R/W	0
FTXURSTATUS	[17]	Primary TX FIFOx under-run interrupt status. This is used by interrupt clear bit. If this is high, write '1' to clear the interrupt. 0 = Interrupt did not occur. 1= Interrupt occurred.	R/W	0
FTXURINTEN	[16]	Primary TX FIFOx Under-run Interrupt Enable 0 = Disables TXFIFO Under-run INT 1 = Enables TXFIFO Under-run INT	R/W	0
FTX2EMPT	[15]	Primary TX FIFO2 empty Status Indication 0 = TX FIFO2 is not empty (Ready to transmit Data) 1 = TX FIFO2 is empty (Not Ready to transmit Data)	R	0

I2SCON	Bit	Description	R/W	Reset Value
FTX1EMPT	[14]	Primary TX FIFO1 empty Status Indication 0 = TX FIFO1 is not empty (Ready to transmit Data) 1= TX FIFO1 is empty (Not Ready to transmit Data)	R	0
FTX2FULL	[13]	Primary TX FIFO2 full Status Indication 0 = TX FIFO2 is not full 1= TX FIFO2 is full	R	0
FTX1FULL	[12]	Primary TX FIFO1 full Status Indication 0 = TX FIFO1 is not full 1= TX FIFO1 is full	R	0
LRI	[11]	Left/ Right channel clock indication. **Note:** LRI meaning depends on the value of LRP bit of I2SMOD register. 0 = Left (If LRP bit is low) or right (If LRP bit is high) 1= Right (If LRP bit is low) or left (If LRP bit is high)	R	0
FTX0EMPT	[10]	Primary Tx FIFO0 empty status indication. 0 = FIFO is not empty (ready for transmit data to channel) 1= FIFO is empty (not ready for transmit data to channel)	R	0
FRXEMPT	[9]	Rx FIFO empty status indication. 0 = FIFO is not empty 1 = FIFO is empty	R	0
FTX0FULL	[8]	Primary Tx FIFO0 full status indication. 0 = FIFO is not full 1 = FIFO is full	R	0
FRXFULL	[7]	Rx FIFO full status indication. 0 = FIFO is not full (Ready to receive data from channel) 1 = FIFO is full (Not ready to receive data from channel)	R	0
TXDMAPAUSE	[6]	Tx DMA operation pause command for primary TX FIFOx. **Note:** If this bit is activated at any time, the DMA request halts after current on-going DMA transfer is complete. 0 = No pause DMA operation for TX FIFOx 1 = Pause DMA operation for TX FIFOx	R/W	0
RXDMAPAUSE	[5]	Rx DMA operation pause command. **Note:** If this bit is activated at any time, the DMA request halts after current on-going DMA transfer is complete. 0 = No pause DMA operation 1= Pause DMA operation	R/W	0

I2SCON	Bit	Description	R/W	Reset Value
TXCHPAUSE	[4]	Tx channel operation pause command **Note:** If this bit is activated at any time, the channel operation halts after left-right channel data transfer is complete. 0 = No pause operation for TX Channel 1 = Pause operation for TX Channel	R/W	0
RXCHPAUSE	[3]	Rx channel operation pause command. **Note:** If this bit is activated at any time, the channel operation halts after left-right channel data transfer is complete. 0 = No pause operation for RX Channel 1= Pause operation for RX Channel	R/W	0
TXDMACTIVE	[2]	Tx DMA active for primary TX FIFOx (start DMA request). **Note:** If this bit is set from high to low, the DMA operation is forced to stop immediately. 0 = Inactive, 1 = Active	R/W	0
RXDMACTIVE	[1]	Rx DMA active (start DMA request). **Note:** If this bit is set from high to low, the DMA operation is forced to stop immediately. 0 = Inactive, 1 = Active	R/W	0
I2SACTIVE	[0]	I2S interface active (start operation). 0 = Inactive, 1 =Active	R/W	0

8.2 I2S INTERFACE MODE REGISTER (I2SMOD, R/W, ADDRESS = 0XF200_0004)

I2SMOD	Bit	Description	R/W	Reset Value
OP_CLK	[31:30]	Operation clock for I2S logic. 00 = Codec clock out 01 = Codec clock in 10 = Bit clock out 11 = PCLK	R/W	00
Reserved	[29]	Reserved	R	0
OP_MUX_SEL	[28]	Mux selection for secondary TX FIFO_S 0 = TX FIFO_S gets data from APB SFR interface 1= TX FIFO_S gets data form internal DMA interface Before trying to change this field to 1 from 0, s/w must poll I2STRNCNT register to confirm that all the transfer is done according to internal DMA setting. There is no restriction on switching to 1 from 0.	R/W	0

I2SMOD	Bit	Description	R/W	Reset Value
BLC_S	[27:26]	Bit Length Control Bit which decides transmission of 8/16/24 bits per audio channel for Secondary TX FIFO_S 00 = 16 Bits per channel 01 = 8 Bits Per Channel 10 = 24 Bits Per Channel 11 = Reserved	R/W	00
BLC_P	[25:24]	Bit Length Control Bit Which decides transmission of 8/16/24 bits per audio channel for Primary TX FIFOx 00 = 16 Bits per channel 01 = 8 Bits Per Channel 10 = 24 Bits Per Channel 11 = Reserved	R/W	00
Reserved	[23:22]	Reserved	R	00
CDD2	[21:20]	Channel-2 Data Discard. Discard means zero padding. It only supports 8/16 bit mode. 00 = No Discard 01 = I2STXD[15:0] Discard 10 = I2STXD[31:16] Discard 11 = Reserved	R/W	00
CDD1	[19:18]	Channel-1 Data Discard. Discard means zero padding. It only supports 8/16 bit mode. 00 = No Discard 01 = I2STXD[15:0] Discard 10 = I2STXD[31:16] Discard 11 = Reserved	R/W	00
DCE	[17:16]	Data Channel Enable. [17] = SD2 channel enable [16] = SD1 channel enable	R/W	00
Reserved	[15]	Reserved	R	0
BLC	[14:13]	Bit Length Control Bit Which decides transmission of 8/16/24 bits per audio channel for final mixed sound Tx output or Rx input. 00 = 16 Bits per channel 01 = 8 Bits Per Channel 10 = 24 Bits Per Channel 11 = Reserved	R/W	00

I2SMOD	Bit	Description	R/W	Reset Value
CDCLKCON	[12]	Determine codec clock source 0 = Use internal codec clock source 1 = Get codec clock source from external codec chip (For more information refer to Figure 10.2-2 and Figure 10.2-3)	R/W	0
IMS	[11:10]	I2S master (internal/external) or slave mode select. 00 = Master mode (divide mode, using PCLK) 01 = Master mode (bypass mode, using I2SCLK) 10: = Slave mode (divide mode, using PCLK) 11 = Slave mode (bypass mode, using I2SCLK) (For more information refer to Figure 10.2-2 and Figure 10.2-3)	R/W	00
TXR	[9:8]	Transmit or receive mode select. 00 = Transmit only mode 01 = Receive only mode 10 = Transmit and receive simultaneous mode 11 = Reserved	R/W	00
LRP	[7]	Left/ Right channel clock polarity select. 0 = Low for left channel and high for right channel 1 = High for left channel and low for right channel	R/W	0
SDF	[6:5]	Serial data format. 00 = I2S format 01 = MSB-justified (left-justified) format 10 = LSB-justified (right-justified) format 11 = Reserved	R/W	00
RFS	[4:3]	I2S root clock (codec clock) frequency select. 00 = 256 fs, where fs is sampling frequency 01 = 512 fs 10 = 384 fs 11 =768 fs	R/W	00
BFS	[2:1]	Bit clock frequency select. 00 = 32 fs, where fs is sampling frequency 01 = 48 fs 10 = 16 fs 11 = 24 fs	R/W	00
Reserved	[0]	Reserved	R	0

8.3 I2S INTERFACE FIFO CONTROL REGISTER (I2SFIC, R/W, ADDRESS = 0XF200_0008)

I2SFIC	Bit	Description	R/W	Reset Value
Reserved	[31]	Reserved	W	0
FTX2CNT	[30:24]	Primary TX FIFO2 data count. FIFO has 16 depth, therefore value ranges from 0 to 15. N: Data count N of FIFO	R	0x00
Reserved	[23]	Reserved	R	0
FTX1CNT	[22:16]	Primary TX FIFO1 data count. FIFO has 16 depth, therefore value ranges from 0 to 15. N: Data count N of FIFO	R	0x00
TFLUSH	[15]	Primary TX FIFO flush command. 0 = No flush, 1 = Flush	R/W	0
FTX0CNT	[14:8]	Primary TX FIFO0 data count. FIFO has 16 dept, therefore value ranges from 0 to 16. N: Data count N of FIFO	R	0x00
RFLUSH	[7]	RX FIFO flush command. 0 = No flush, 1 = Flush	R/W	0
FRXCNT	[6:0]	RX FIFO data count. FIFO has 16 dept, so value ranges from 0 to 16. N: Data count N of FIFO	R	0x00

8.4 I2S INTERFACE CLOCK DIVIDER CONTROL REGISTER (I2SPSR, R/W, ADDRESS = 0XF200_000C)

I2SPSR	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	R	0x00
PSRAEN	[15]	Active Pre-scaler (Clock divider) 0: Inactive, 1: Active	R/W	0
Reserved	[14]	Reserved	R	0
PSVALA	[13:8]	Pre-scaler (Clock divider) a division value. N: Division factor is N+1	R/W	0x00
Reserved	[7:0]	Reserved	R	0x00

8.5 I2S INTERFACE TRANSMIT DATA REGISTER (I2STXD, W, ADDRESS = 0XF200_0010)

I2STXD	Bit	Description	R/W	Reset Value
I2STXD	[31:0]	Primary TX FIFO write data. **Note:** The left/ right channel data is allocated as the following bit fields. R[31:16], L[15:0] if 16-bit BLC R[23:16], L[7:0] if 8-bit BLC Refer to Figure 10.2-7 if 24-bit BLC.	W	0x00

8.6 I2S INTERFACE RECEIVE DATA REGISTER (I2SRXD, R, ADDRESS = 0XF200_0014)

I2SRXD	Bit	Description	R/W	Reset Value
I2SRXD	[31:0]	RX FIFO read data. **Note:** The left/ right channel data is allocated as the following bit fields. R[31:16], L[15:0] if 16-bit BLC R[23:16], L[7:0] if 8-bit BLC Refer to Figure 10.2-9 if 24-bit BLC.	R	0x00

8.7 I2S INTERFACE TXFIFO_S CONTROL REGISTER (I2SFICS, R/W, ADDRESS = 0XF200_0018)

I2SFICS	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	R	0x00
TFLUSHS	[15]	Secondary TX FIFO_S flush command. 0 = No flush, 1= Flush	R/W	0
FTXSCNT	[14:8]	Secondary TX FIFO_S data count. FIFO has 64 depth, therefore value ranges from 0 to 64. N: Data count N of FIFO	R	0x00
Reserved	[7:0]	Reserved	R	0x00

8.8 I2S INTERFACE TRANSMIT DATA REGISTER FOR TXFIFO_S (I2STXDS, W, ADDRESS = 0XF200_001C)

I2STXDS	Bit	Description	R/W	Reset Value
I2STXDS	[31:0]	Secondary TX FIFO_S write data. **Note:** The left/ right channel data is allocated as the following bit fields. R[31:16], L[15:0] if 16-bit BLC R[23:16], L[7:0] if 8-bit BLC Refer to Figure 10.2-7 if 24-bit BLC	W	0x00

8.9 I2S AHB DMA CONTROL REGISTER (I2SAHB, R/W, ADDRESS = 0XF200_0020)

I2SAHB	Bit	Description	R/W	Reset Value
Reserved	[31:28]	Reserved	R	0x00
I2SLVL3EN	[27]	Enable buffer level 3 interrupt. 0 = Disables I2SLVL3INT. 1 = Enables I2SLVL3INT.	R/W	0
I2SLVL2EN	[26]	Enable buffer level 2 interrupt. 0 = Disables I2SLVL2INT. 1 = Enables I2SLVL2INT.	R/W	0
I2SLVL1EN	[25]	Enable buffer level 1 interrupt. 0 = Disables I2SLVL1INT. 1 = Enables I2SLVL1INT.	R/W	0
I2SLVL0EN	[24]	Enable buffer level 0 interrupt. 0 = Disables I2SLVL0INT. 1 = Enables I2SLVL0INT.	R/W	0
I2SLVL3INT	[23]	Buffer level 3 interrupt status flag. During DMA operation, if generated address in DMA matches with I2SLVL3ADDR, this flag is set. To clear this flag, use I2SLVL3CLR field.	R	0
I2SLVL2INT	[22]	Buffer level 2 interrupt status flag. During DMA operation, if generated address in DMA matches with I2SLVL2ADDR, this flag is set. To clear this flag, use I2SLVL2CLR field.	R	0
I2SLVL1INT	[21]	Buffer level 1 interrupt status flag. During DMA operation, if generated address in DMA matches with I2SLVL1ADDR, this flag is set. To clear this flag, use I2SLVL1CLR field.	R	0
I2SLVL0INT	[20]	Buffer level 0 interrupt status flag. During DMA operation, if generated address in DMA matches with I2SLVL0ADDR, this flag is set. To clear this flag, use I2SLVL0CLR field.	R	0

I2SAHB	Bit	Description	R/W	Reset Value
I2SLVL3CLR	[19]	Clear I2SLVL3INT flag If I2SLVL3INT is set, setting I2SLVL3CLR to 1 clears I2SLVL3INT to 0. Writing to zero has no meaning.	R/W	0
I2SLVL2CLR	[18]	Clear I2SLVL2INT flag If I2SLVL2INT is set, setting I2SLVL2CLR to 1 clears I2SLVL2INT to 0. Writing to zero has no meaning.	R/W	0
I2SLVL1CLR	[17]	Clear I2SLVL1INT flag If I2SLVL1INT is set, setting I2SLVL1CLR to 1 clears I2SLVL1INT to 0. Writing to zero has no meaning.	R/W	0
I2SLVL0CLR	[16]	Clear I2SLVL0INT flag If I2SLVL0INT is set, setting I2SLVL0CLR to 1 clears I2SLVL0INT to 0. Writing to zero has no meaning.	R/W	0
Reserved	[15:6]	Reserved	R	0x00
I2SDMARLD-	[5]	Auto-reload I2S internal DMA Configuration if DMA operation is done and re-start I2S internal DMA automatically. 0 = Disables auto-reload function 1 = Enables auto-reload function Before switching to 0 from 1, s/w must check if DMA_EN is set.	R/W	0
Reserved	[4]	Reserved	R	0
I2SINTMASK	[3]	Disable interrupt request signal 0 = Enables interrupt request if DMA auto-reload is on. 1 = Disables interrupt request if DMA auto-reload is on. After DMA transfers the data related to DMA configuration, interrupt signal occurs. If I2SINTMASK bit is set, I2SDMAINT and interrupt signal will NOT be set. I2SINTMASK does NOT effect I2SLVLxINT and under-run interrupt.	R/W	0
I2SDMAINT	[2]	DMA interrupt status flag. After DMA operation is end, this flag is set. To clear this flag, use I2SDMACLR field.	R	0
I2SDMACLR	[1]	Clear DMA interrupt status flag If I2SINT is set, setting I2SDMACLR to 1 clears I2SDMAINT to 0. Writing to zero has no meaning.	R/W	0

I2SAHB	Bit	Description	R/W	Reset Value
I2SDMAEN	[0]	Enable I2S internal DMA Use internal DMA in I2S after this bit field is ON. Internal DMA issues 32-bit single read transaction for AHB and TXFIFO0 holds data returned by DMA. **Warning:** If I2SDMARLD is set, I2SDMAEN bit is automatically cleared when all the transfer specified in SFR is done. Then I2SDMAEN bit is automatically set after auto-reload operation is done, If auto-reload operation is ongoing, s/w intervention on this field causes mal-function of internal DMA operations. To manipulate I2SAHB register, s/w must check that I2SDMAEN is in stable state.	R/W	0

8.10 I2S AHB DMA START ADDRESS REGISTER (I2SSTR, R/W, ADDRESS = 0XF200_0024)

I2SSTR	Bit	Description	R/W	Reset Value
I2SSTR	[31:0]	Start address of I2S internal DMA operation. If DMAEN is ON, internal DMA in I2S starts DMA operation based on I2SSTR address. Internal DMA handles word-aligned address only. To get best performance, I2SSTR should be 64 word-aligned address Valid address range for I2SSTR is from 0xC000_0000 to 0xC01F_FFFF.	R/W	0x00

8.11 I2S AHB DMA SIZE REGISTER (I2SSIZE, R/W, ADDRESS = 0XF200_0028)

I2SSIZE	Bit	Description	R/W	Reset Value
TRNS_SIZE	[31:16]	Transfer block size for I2S internal DMA If I2S internal DMA is enabled, I2S internal DMA transfers TRNS_SIZE word(s) data from memory before DMA done interrupt occurs. Valid ranges for TRNS_SIZE is from 0x0000 to 0x8000.	R/W	0x7FFF
Reserved	[15:0]	Reserved	R	0x0000

8.12 I2S AHB DMA TRANSFER COUNT REGISTER (I2STRNCNT, R, ADDRESS = 0XF200_002C)

I2STRNCNT	Bit	Description	R/W	Reset Value
Reserved	[31:24]	Reserved	R	
I2STRNCNT	[23:0]	Number of transferred data using I2S internal DMA (word unit). User program terminates I2S internal DMA operation turning DMA_EN off. After DMA_EN is 0, user program reads I2STRNCNT value to know where I2S internal DMA has stopped.	R	

8.13 I2S AHB DMA LEVEL 0 INTERRUPT ADDRESS REGISTER (I2SLVL0ADDR, R/W, ADDRESS = 0XF200_0030)

I2SLVL0ADDR	Bit	Description	R/W	Reset Value
I2SLVL0ADDR	[31:10]	AHB DMA level 0 interrupt address While I2SLVL0EN in I2SAHB register is set, AHB DMA compares this register to generated address in DMA. If two values match, I2SLVL0INT in I2SAHB is set. Valid address range for I2SLVL0ADDR is from 0xC000_0000 to 0xC01F_FFFF.	R	0x00
Reserved	[9:1]	Reserved	R	0x00
I2SLVL0STOP	[0]	Precise stop enable 0 = Do not stop DMA operation 1 = Stop DMA operation if DMA working addresses is matched with I2SLVL0ADDR. I2SDMAEN in I2SAHB is turned off automatically.	R/W	0

8.14 I2S AHB DMA LEVEL 1 INTERRUPT ADDRESS REGISTER (I2SLVL1ADDR, R/W, ADDRESS = 0XF200_0034)

I2SLVL1ADDR	Bit	Description	R/W	Reset Value
I2SLVL1ADDR	[31:10]	AHB DMA level 1 interrupt address While I2SLVL1EN in I2SAHB register is set, AHB DMA compares this register to generated address in DMA. If two values match, I2SLVL1INT in I2SAHB is set. Valid address range for I2SLVL1ADDR is from 0xC000_0000 to 0xC01F_FFFF.	R	0x00
Reserved	[9:1]	Reserved	R	0x00
I2SLVL1STOP	[0]	Precise stop enable 0 = Do not stop DMA operation 1 = Stop DMA operation if DMA working addresses is matched with I2SLVL1ADDR. I2SDMAEN in I2SAHB is turned off automatically.	R/W	0

8.15 I2S AHB DMA LEVEL 2 INTERRUPT ADDRESS REGISTER (I2SLVL2ADDR, R/W, ADDRESS = 0XF200_0038)

I2SLVL2ADDR	Bit	Description	R/W	Reset Value
I2SLVL2ADDR	[31:10]	AHB DMA level 2 interrupt address While I2SLVL2EN in I2SAHAB register is set, AHB DMA compares this register to generated address in DMA. If two values match, I2SLVL2INT in I2SAHB is set. Valid address range for I2SLVL2ADDR is from 0xC000_0000 to 0xC01F_FFFF.	R	0x00
Reserved	[9:1]	Reserved	R	0x00
I2SLVL2STOP	[0]	Precise stop enable 0 = Do not stop DMA operation 1 = Stop DMA operation if DMA working addresses is matched with I2SLVL2ADDR. I2SDMAEN in I2SAHB is turned off automatically.	R/W	0

8.16 I2S AHB DMA LEVEL 3 INTERRUPT ADDRESS REGISTER (I2SLVL3ADDR, R/W, ADDRESS = 0XF200_003C)

I2SLVL3ADDR	Bit	Description	R/W	Reset Value
I2SLVL3ADDR	[31:10]	AHB DMA level 3 interrupt address While I2SLVL3EN in I2SAHB register is set, AHB DMA compares this register to generated address in DMA. If two values match, I2SLVL3INT in I2SAHB is set. Valid address range for I2SLVL3ADDR is from 0xC000_0000 to 0xC01F_FFFF.	R	0x00
Reserved	[9:1]	Reserved	R	0x00
I2SLVL3STOP	[0]	Precise stop enable 0 = Do not stop DMA operation 1= Stop DMA operation if DMA working addresses is matched with I2SLVL3ADDR. I2SDMAEN in I2SAHB is turned off automatically.	R/W	0

NOTES

10.3 I2S CONTROLLER(2CH)

1 OVERVIEW

Inter-IC Sound (I2S) is one of the popular digital audio interfaces. The bus handles audio data, while the other signals, such as sub-coding and control, is transferred separately. It is possible to transmit data between two I2S bus. To minimize the number of pins required and to keep wiring simple, basically, a 3-line serial bus is used. It consists of a line for two time-multiplexed data channels, a word select line and a clock line.

I2S interface transmits or receives sound data from external stereo audio codec. To transmit or receive data, two 32x16 First-In-First-Out (FIFO) data structures are included and DMA transfer mode to transmit or receive samples is supported. I2S-specific clock is supplied from internal system clock controller through I2S clock divider or direct clock source.

2 FEATURE

- Supports 2-ch I2S-bus (I2S1, I2S2) for audio interface with DMA-based operation
- Supports Serial, 8/ 16/ 24-bit per channel data transfers
- Supports I2S, MSB-justified and LSB-justified data format

3 BLOCK DIAGRAM

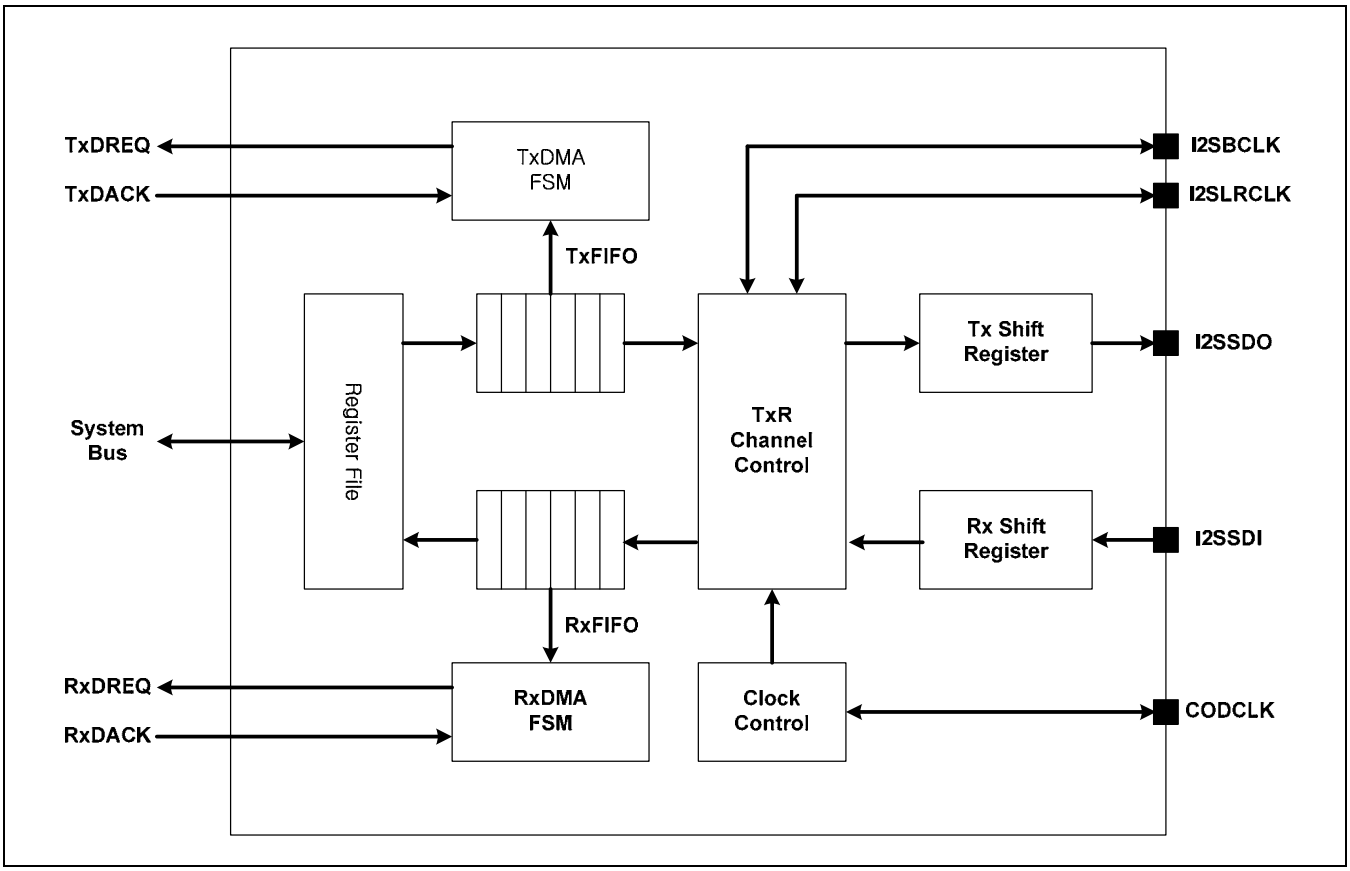


Figure 10.3-1 I2S-Bus Block Diagram

4 FUNCTIONAL DESCRIPTIONS

I2S interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as shown in Figure 10.3-1. Note that each FIFO has 32-bit width and 16 depth structure, which contains left/ right channel data. Therefore, FIFO access and data transfer are handled with left/ right pair unit. Figure 10.3-1 shows the functional block diagram of I2S interface.

4.1 MASTER/SLAVE MODE

Set IMS bit of I2SMOD register to select master or slave mode. In master mode, I2SSCLK and I2SLRCLK are generated internally and supplied to external device. Therefore a root clock is needed for generating I2SSCLK and I2SLRCLK by dividing. The I2S pre-scaler (clock divider) generates a root clock with divided frequency from internal system clock. In external master mode, the root clock is fed directly from external I2S. The I2SSCLK and I2SLRCLK are supplied from the pin (GPIOs) in slave mode.

Master/Slave mode is different with TX/RX. Master/Slave mode presents the direction of I2SLRCLK and I2SSCLK. The direction of I2SCDCLK (This is only auxiliary) is not a concern. If I2S bus interface transmits clock signals to I2S codec, I2S bus is master mode. But if I2S bus interface receives clock signal from I2S codec, I2S bus is slave mode. TX/RX mode indicates the direction of data flow. If I2S bus interface transmits data to I2S codec, this is TX mode. Conversely, I2S bus interface receives data from I2S codec that is RX mode. Let us distinguish Master/ Slave mode from TX/RX mode.

Figure 10.3-2 shows the route of the root clock with internal master or external master mode setting in I2S clock control block and system controller. Note that RCLK indicates root clock and this clock is supplied to external I2S codec chip at internal master mode. To control CLOCK CONTROLLER, Refer to AUDIO1_SEL and AUDIO2_SEL of 9.2.4 Select clock source 3 Register (Audio & Others clock source register) in 2.3 CLOCK STRATEGY.

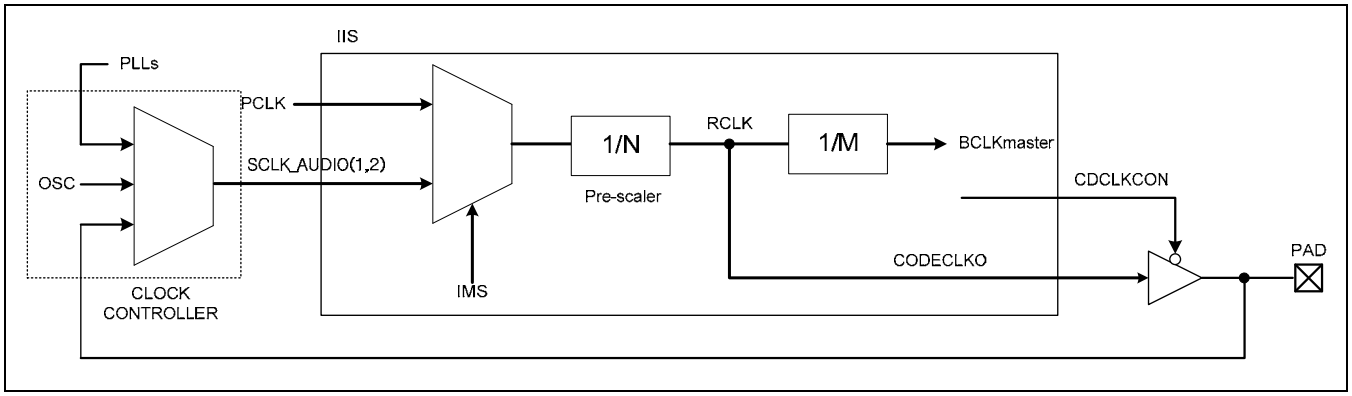


Figure 10.3-2 I2S Clock Control Block Diagram

4.2 DMA TRANSFER

In the DMA transfer mode, the transmitter or receiver FIFO are accessible by DMA controller. DMA service request is activated internally by the transmitter or receiver FIFO state. The FTXEMPT, FRXEMPT, FTXFULL, and FRXFULL bits of I2SCON register represent the transmitter or receiver FIFO data state. Especially, FTXEMPT and FRXFULL bit are the ready flag for DMA service request; the transmit DMA service request is activated if TXFIFO is not empty and the receiver DMA service request is activated if RXFIFO is not full.

The DMA transfer uses only handshaking method for single data. Note that during DMA acknowledge activation, the data read or write operation should be performed.

* Reference: DMA request point

- TX mode: (FIFO is not full) and (TXDMACTIVE is active)
- RX mode: (FIFO is not empty) and (RXDMACTIVE is active)

4.3 AUDIO SERIAL DATA FORMAT

4.3.1 I2S-bus format

The I2S bus has four lines including serial data input I2SSDI, serial data output I2SSDO, left/ right channel select clock I2SLRCLK, and serial bit clock I2SBCLK; the device generating I2SLRCLK and I2SBCLK is the master.

Serial data is transmitted in 2's complement with the MSB first with a fixed position, whereas the position of the LSB depends on the word length. The transmitter sends the MSB of the next word at one clock period after the I2SLRCLK is changed. Serial data sent by the transmitter may be synchronized with either the trailing or the leading edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal. Therefore there are some restrictions to transmit data that is synchronized with the leading edge.

The LR channel select line indicates the channel being transmitted. I2SLRCLK is changed either on a trailing or leading edge of the serial clock, but it does not need to be symmetrical. In the slave, this signal is latched on the leading edge of the clock signal. The I2SLRCLK line changes one clock period before the MSB is transmitted.

This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clears the input for the next word.

4.3.2 MSB (Left) Justified

MSB-Justified (Left-Justified) format is similar to I2S bus format, except that in MSB-justified format, the transmitter always sends the MSB of the next word at the same time whenever the I2SLRCLK is changed.

4.3.3 LSB (Right) Justified

LSB-Justified (Right-Justified) format is opposite to the MSB-justified format. In other word, the transferring serial data is aligned with ending point of I2SLRCLK transition.

Figure 10.3-3 shows the audio serial format of I2S, MSB-justified, and LSB-justified. Note that in this figure, the word length is 16-bit and I2SLRCLK makes transition every 24 cycle of I2SBCLK (BFS is 48 fs, where fs is sampling frequency; I2SLRCLK frequency).

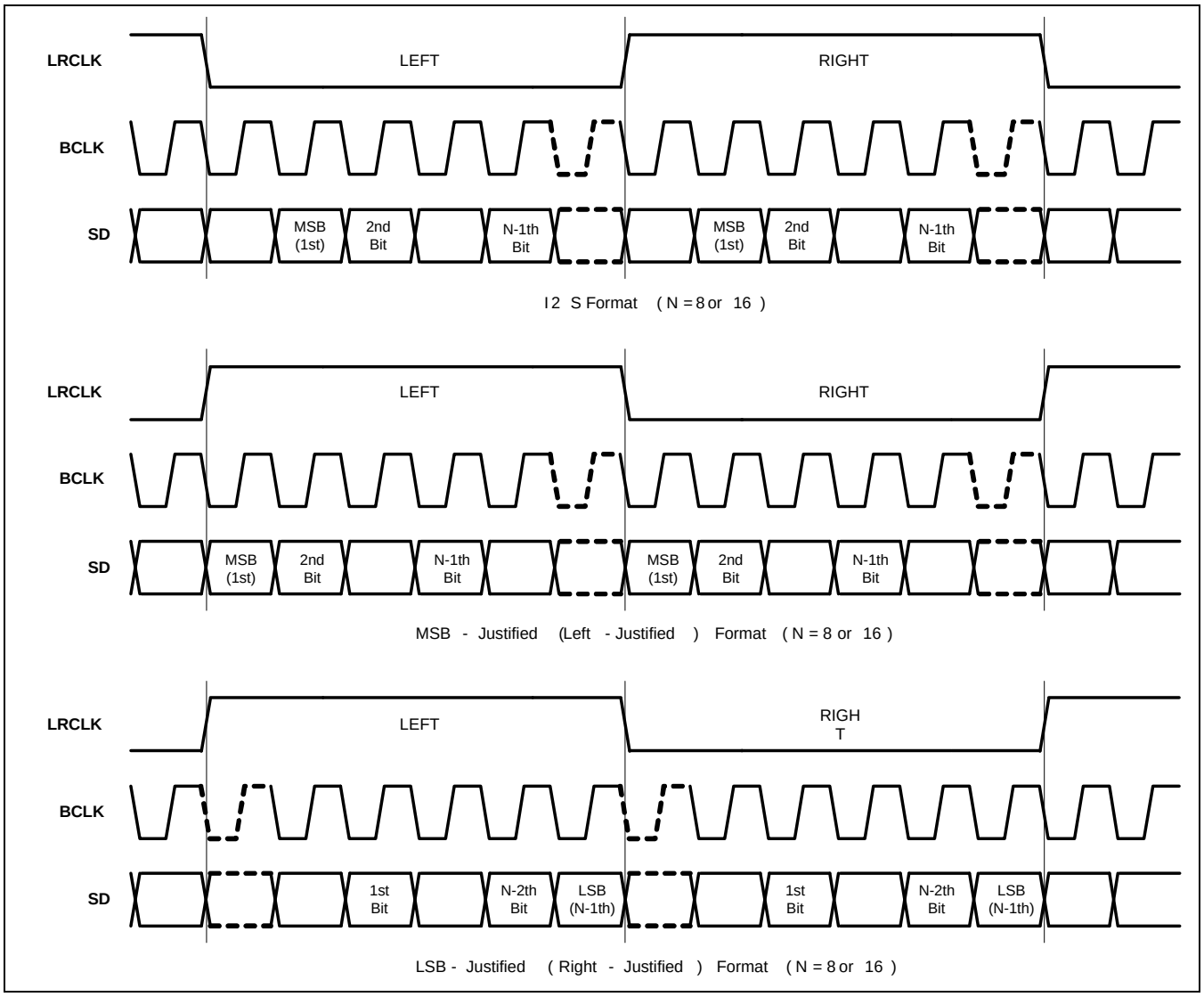


Figure 10.3-3 I2S Audio Serial Data Formats

4.3.4 sampling frequency and master clock

Master clock frequency (RCLK) is selected by sampling frequency as shown in Table 10.3-1. Because RCLK is made by I2S pre-scaler, the pre-scaler value and RCLK type (256fs or 384fs or 512fs or 768fs) should be determined properly.

Table 10.3-1 CODEC Clock (CODECLK = 256fs, 384fs, 512fs, 768fs)

I2SLRCK (fs)	8.000 kHz	11.025 kHz	16.000 kHz	22.050 kHz	32.000 kHz	44.100 kHz	48.000 kHz	64.000 kHz	88.200 kHz	96.000 kHz
CODECLK (MHz)	256fs									
	2.0480	2.8224	4.0960	5.6448	8.1920	11.2896	12.2880	16.3840	22.5792	24.5760
	384fs									
	3.0720	4.2336	6.1440	8.4672	12.2880	16.9344	18.4320	24.5760	33.8688	36.8640
	512fs									
	4.0960	5.6448	8.1920	11.2900	16.3840	22.5790	24.5760	32.7680	45.1580	49.1520
	768fs									
	6.1440	8.4672	12.2880	16.9340	24.5760	33.8690	36.8640	49.1520	67.7380	73.7280

4.3.5 I2S clock mapping table

Before selecting BFS, RFS, and BLC bits of I2SMOD register, please refer to Table 10.3-2. Table 10.3-2 shows the allowed clock frequency mapping relations.

Table 10.3-2 I2S Clock Mapping Table

Clock Frequency		RFS			
		256 fs (00B)	512 fs (01B)	384 fs (10B)	768 fs (11B)
BFS	16 fs (10B)	(a)	(a)	(a)	(a)
	24 fs (11B)	-	-	(a)	(a)
	32 fs (00B)	(a) (b)	(a) (b)	(a) (b)	(a) (b)
	48 fs (01B)	-	-	(a) (b) (c)	(a) (b) (c)
Descriptions		(a) Allowed if BLC is 8-bit (b) Allowed if BLC is 16-bit (c) Allowed if BLC is 24-bit			

5 PROGRAMMING GUIDE

The I2S bus interface is accessed either by the processor using programmed I/O instructions or by the DMA controller.

5.1 INITIALIZATION

1. Before you use I2S bus interface, you must configure GPIOs to I2S mode and check signal's direction. The I2SLRCLK, I2SSCLK and I2SCDCLK are inout-type. The each of I2SSDI and I2SSDO is input and output.
2. Select clock source. The S5PC100 has three clock sources namely PCLK, EPLL and external codec. For more details refer to Figure 10.3-2.

5.2 PLAY MODE (TX MODE) WITH DMA

1. TXFIFO is flushed before operation. If you do not distinguish Master/ Slave mode from TX/RX mode, you must study Master/ Slave mode and TX/RX mode. For more information refer to section Master/Slave.
2. To configure I2SMOD register and I2SPSR (I2S pre-scaler register) properly.
3. To operate system in stability, the internal TXFIFO should be almost full before transmission. First of all, DMA starts because of that reason.
4. Basically, I2S bus does not support the interrupt. Therefore, check state by polling through accessing SFR.
5. If TXFIFO is full, now then you make I2SACTIVE be asserted.

5.3 RECORDING MODE (RX MODE) WITH DMA

1. RXFIFO is flushed before operation. If you do not distinguish between Master/ Slave mode and TX/RX mode, you must study Master/Slave mode and TX/RX mode. For more information refer to "Section Master/Slave chapter.
2. To configure I2SMOD register and I2SPSR (I2S pre-scaler register) properly.
3. To operate system in stability, the internal RXFIFO should have minimum one data before DMA operation. Because of that reason, you make I2SACTIVE be asserted.
4. Check RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, start RXDMACTIVE.

6 EXAMPLE CODE

TX CHANNEL

The I2S TX channel provides a single stereo compliant output. The transmit channel operates in master or slave mode. Data is transferred between the processor and the I2S controller via an APB access or a DMA access.
The processor must write words in multiples of two (i.e. for left and right audio sample).The words are serially shifted out timed with respect to the audio bitclk, BCLK and word select clock, LRCLK.
TX Channel has 16X32-bit wide FIFO where the processor or DMA writes upto 16 left/right data samples after enabling the channel for transmission.
An Example sequence is as follows.
Ensure the PCLK and CODCLKI are coming correctly to the I2S controller and FLUSH the TX FIFO using the TFLUSH bit in the Please ensure that I2S Controller is configured in one of the following modes.

- TX only mode

- TX/RX simultaneous mode

TBD

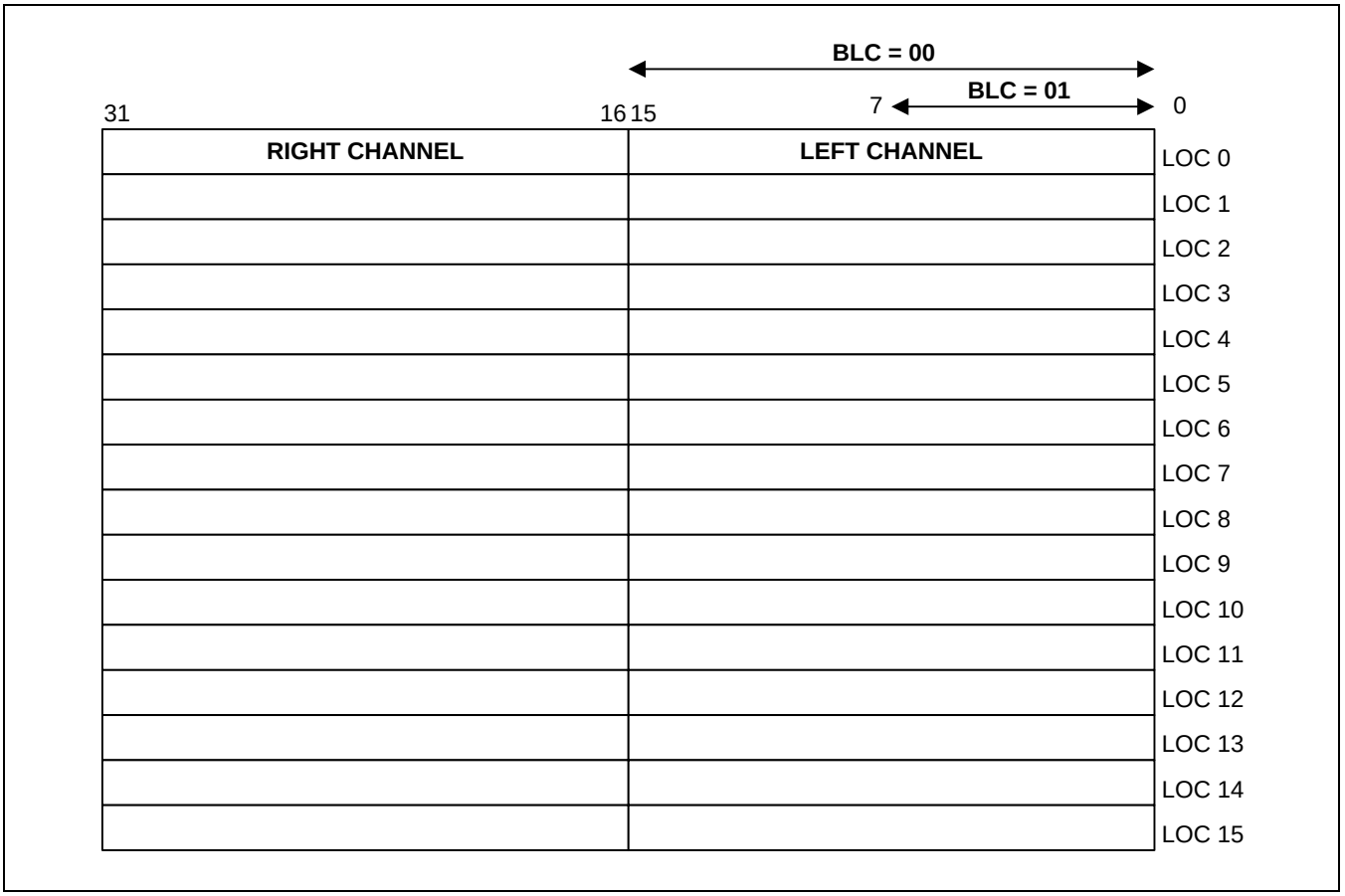


Figure 10.3-4 TX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the TX FIFO for 24-bit/channel BLC as shown in below figure.

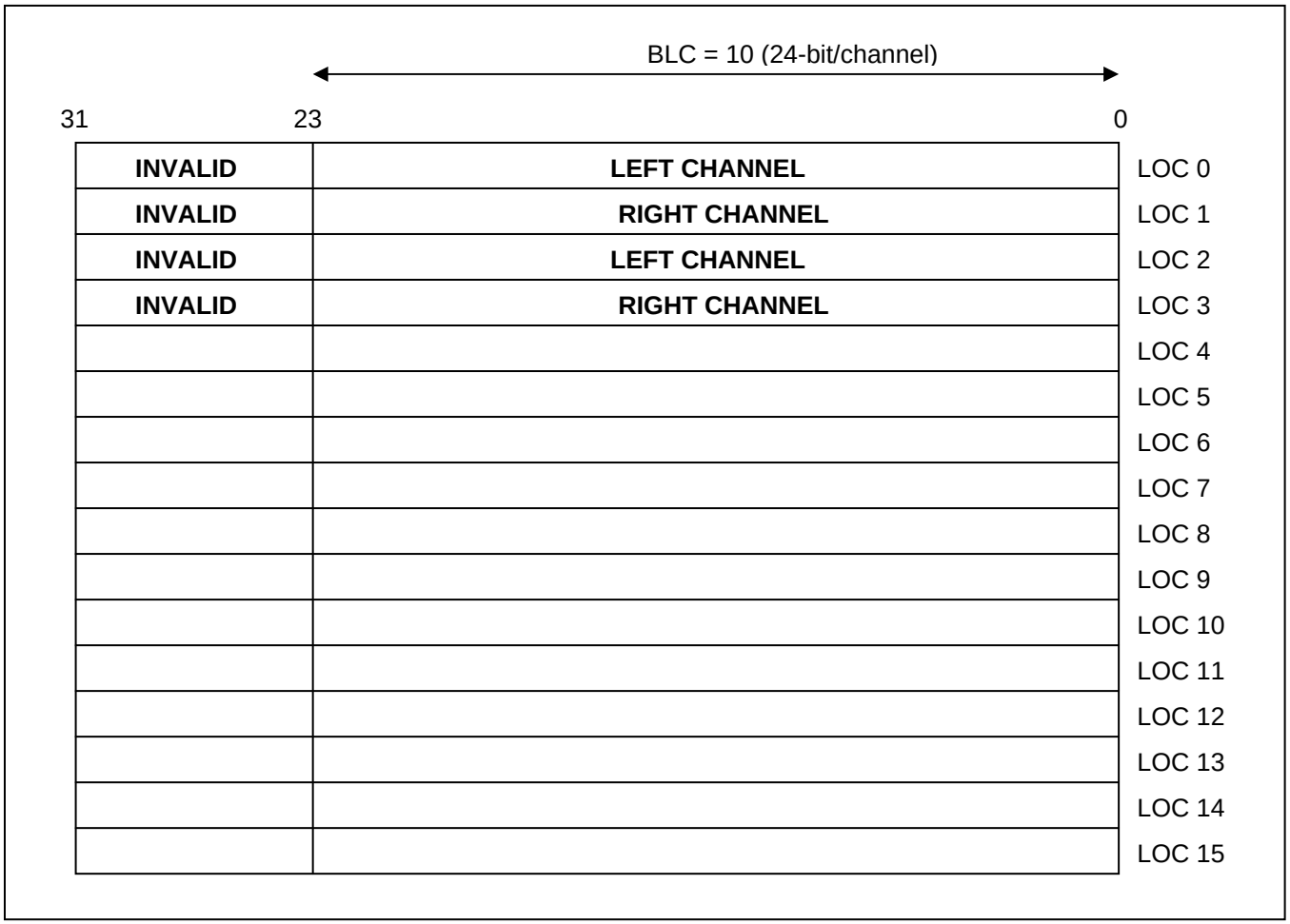


Figure 10.3-5 TX FIF0 Structure for BLC = 10 (24-bit/channel)

TBD

RX CHANNEL

The I2S RX channel provides a single stereo compliant output. The receive channel operates in master or slave mode. Data is received from the input line and transferred into the RX FIFO. The processor then reads this data via an APB read or a DMA access can access this data.

RX Channel has a 16X32-bit wide RX FIFO where the processor or DMA reads upto 16 left/ right data samples after enabling the channel for reception.

TBD

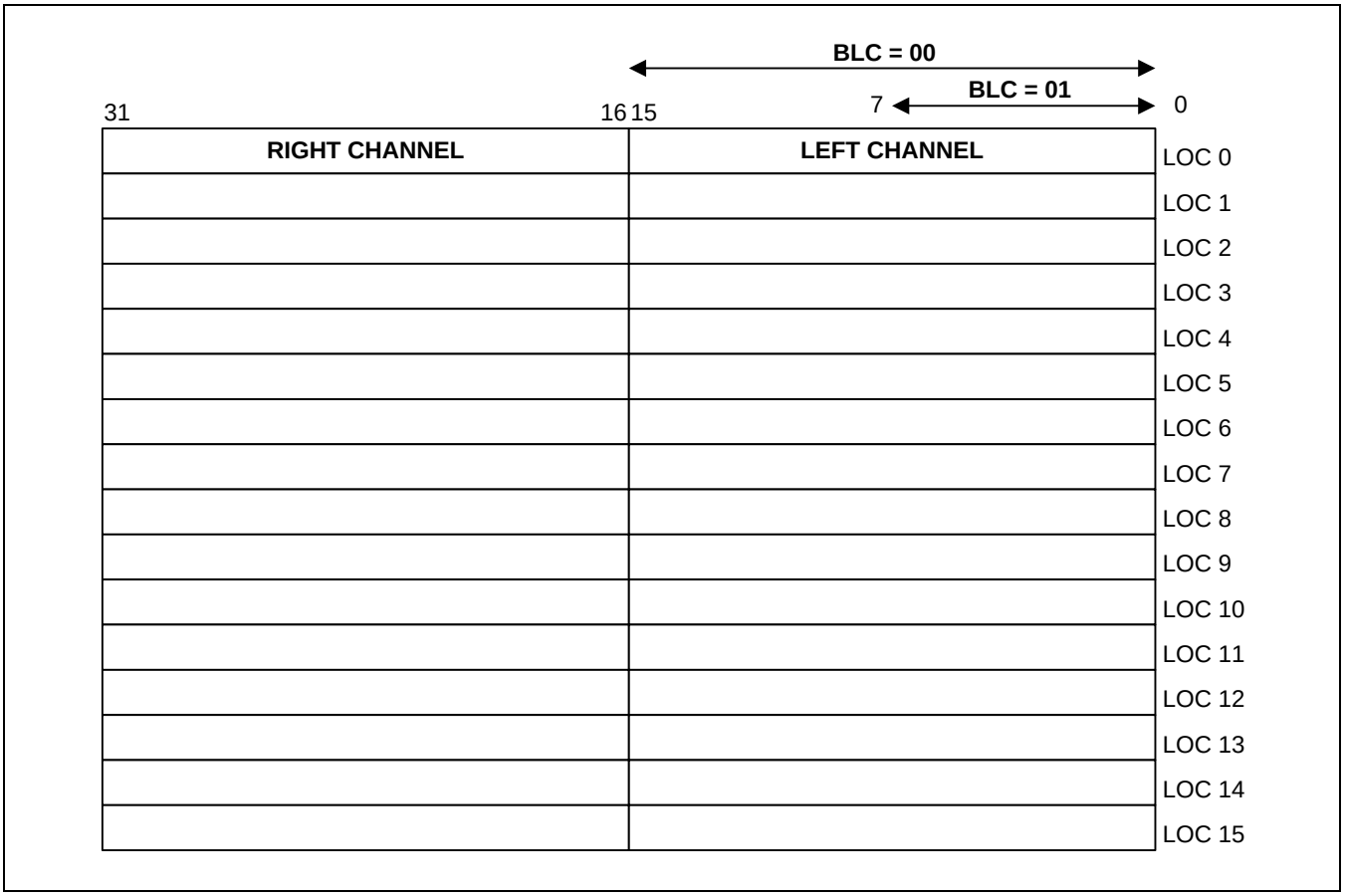


Figure 10.3-6 RX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the RX FIFO for 24-bit/channel BLC as shown in the figure below.

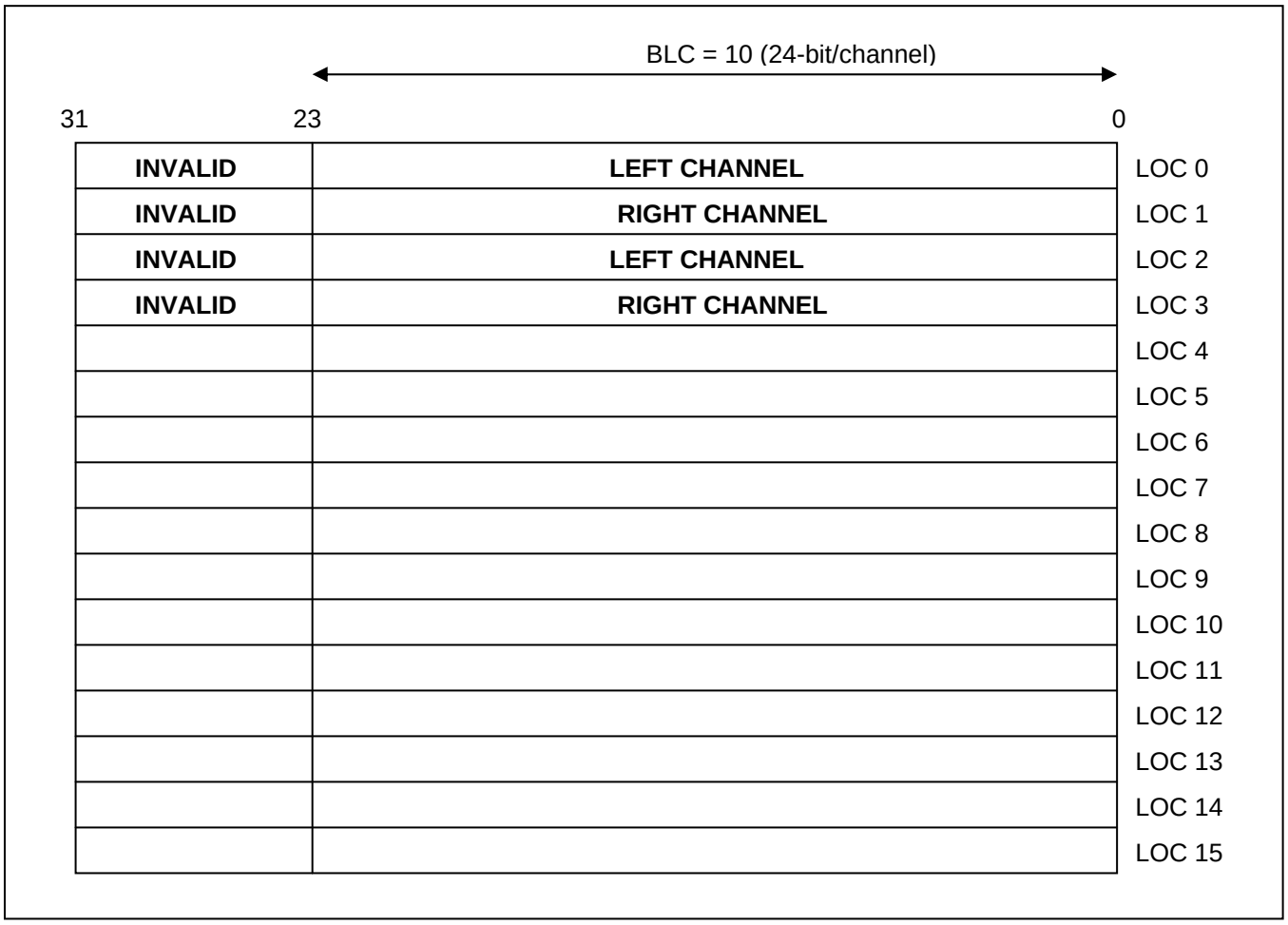


Figure 10.3-7 RX FIF0 Structure for BLC = 10 (24-bit/channel)

TBD

7 I/O DESCRIPTION

I2S (v3.2) external pads are shared with I2S and SDMMC. In order to use these pads for I2S, GPIO must be set before the I2S started. For mode information, please refer to the Chapter GPIO.

Funtion Signal	I/O	Description	Pad	Type
I2S1_CDCLK	Input/Output	I2S1 Codec Clock Input/ Output	Xi2s1CDCLK	muxed
I2S1_SCLK	Input/Output	I2S1 Bit Clock Input/ Output	Xi2s1SCLK	muxed
I2S1_LRCK	Input/Output	I2S1 LR Channel Clock Input/ Output	Xi2s1LRCK	muxed
I2S1_SDI	Input	I2S1 Serial Data Input	Xi2s1SDI	muxed
I2S1_SDO	Output	I2S1 Serial Data Out	Xi2s1SDO	muxed
I2S2_CDCLK	Input/Output	I2S2 Codec Clock Input/ Output	Xmmc2CMD	muxed
I2S2_SCLK	Input/Output	I2S2 Bit Clock Input/ Output	Xmmc2CLK	muxed
I2S2_LRCK	Input/Output	I2S2 LR Channel Clock Input/ Output	Xmmc2DATA[0]	muxed
I2S2_SDI	Input	I2S2 Serial Data Input	Xmmc2DATA[1]	muxed
I2S2_SDO	Output	I2S2 Serial Data Out	Xmmc2DATA[2]	muxed

8 REGISTER DESCRIPTION

8.1 REGISTER OVERVIEW

Table 10.3-3 Register Summary of I2S Interface

Register	Address	R/W	Description	Reset Value
I2SCON	0XF210_0000 0XF220_0000	R/W	I2S Interface Control Register	0xE00
I2SMOD	0XF210_0004 0XF220_0004	R/W	I2S Interface Mode Register	0x0
I2SFIC	0XF210_0008 0XF220_0008	R/W	I2S Interface FIFO Control Register	0x0
I2SPSR	0XF210_000C 0XF220_000C	R/W	I2S Interface Clock Divider Control Register	0x0
I2STXD	0XF210_0010 0XF220_0010	W	I2S Interface Transmit Data Register	0x0
I2SRXD	0XF210_0014 0XF220_0014	R	I2S Interface Receive Data Register	0x0

NOTE: All registers of I2S interface are accessible by word unit with STR/LDR instructions.

8.2 DETAILED DESCRIPTION

8.2.1 I2S interface control register (I2SCON, R/W, Address = 0XF210_0000, 0XF220_0000)

- I2SCON1, R/W, Address = 0XF210_0000
- I2SCON2, R/W, Address = 0XF220_0000

I2SCON	Bit	Description	R/W	Reset Value
Reserved	[31:20]	Reserved	R/W	
FRXOFSTATUS	[19]	RX FIFO OverFlow Interrupt Status.This is used by interrupt clear bit. If this is high, write '1' to clear interrupt. 0 = Interrupt did not occurred. 1 = Interrupt occurred.	R/W	
FRXORINTEN	[18]	RX FIFO OverFlow Interrupt Enable 0 = Disables RXFIFO Under-run INT 1 = Enables RXFIFO Under-run INT	R/W	
FTXURSTATUS	[17]	TX FIFO under-run interrupt status. This is used by interrupt clear bit. If this is high, write '1' to clear the interrupt. 0 = Interrupt did not occurred. 1 = Interrupt occurred.	R/W	
FTXURINTEN	[16]	TX FIFO Under-run Interrupt Enable 0 = Disables TXFIFO Under-run INT 1 = Enables TXFIFO Under-run INT	R/W	
Reserved	[15:12]	Reserved	R/W	

I2SCON	Bit	Description	R/W	Reset Value
LRI	[11]	Left/ Right channel clock indication. Note that LRI meaning is dependent on the value of LRP bit of I2SMOD register. 0 = Left (If LRP bit is low) or right (If LRP bit is high) 1= Right (If LRP bit is low) or left (If LRP bit is high)	R	
FTXEMPT	[10]	Tx FIFO empty status indication. 0 = FIFO is not empty (ready for transmit data to channel) 1 = FIFO is empty (not ready for transmit data to channel)	R	
FRXEMPT	[9]	Rx FIFO empty status indication. 0 = FIFO is not empty 1 = FIFO is empty	R	
FTXFULL	[8]	Tx FIFO full status indication. 0 = FIFO is not full 1 = FIFO is full	R	
FRXFULL	[7]	Rx FIFO full status indication. 0 = FIFO is not full (ready to receive data from channel) 1= FIFO is full (not ready to receive data from channel)	R	
TXDMAPAUSE	[6]	Tx DMA operation pause command. **Note:** If this bit is activated at any time, the DMA request halts after current on-going DMA transfer is complete. 0 = No pause DMA operation 1= Pause DMA operation	R/W	
RXDMAPAUSE	[5]	Rx DMA operation pause command. **Note:** If this bit is activated at any time, the DMA request halts after current on-going DMA transfer is complete. 0 = No pause DMA operation 1 = Pause DMA operation	R/W	
TXCHPAUSE	[4]	Tx channel operation pause command. **Note:** If this bit is activated at any time, the channel operation halts after left-right channel data transfer is complete. 0 = No pause operation 1 = Pause operation	R/W	
RXCHPAUSE	[3]	Rx channel operation pause command. **Note:** If this bit is activated at any time, the channel operation halts after left-right channel data transfer is complete. 0 = No pause operation 1 = Pause operation	R/W	
TXDMACTIVE	[2]	Tx DMA active (start DMA request). **Note:** If this bit is set from high to low, the DMA operation is forced to stop immediately. 0 = Inactive, 1 = Active	R/W	

I2SCON	Bit	Description	R/W	Reset Value
RXDMACTIVE	[1]	Rx DMA active (start DMA request). **Note:** If this bit is set from high to low, the DMA operation is forced to stop immediately. 0 = Inactive, 1= Active	R/W	
I2SACTIVE	[0]	I2S interface active (start operation). 0 = Inactive, 1 = Active	R/W	

8.2.2 I2S Interface Mode Register (I2SMOD, R/W, Address = 0XF210_0004, 0XF220_0004)

- I2SMOD1, R/W, Address = 0XF210_0004
- I2SMOD2, R/W, Address = 0XF220_0004

I2SMOD	Bit	Description	R/W	Reset Value
Reserved	[31:15]	Reserved	R/W	
BLC	[14:13]	Bit Length Control Bit Which decides transmission of 8/16/24 bits per audio channel 00 = 16 Bits per channel 01 = 8 Bits per Channel 10 = 24 Bits per Channel 11 = Reserved	R/W	
CDCLKCON	[12]	Determine codec clock source 0 = Use internal codec clock source 1 = Get codec clock source from external codec chip (For more information refer to Figure 10.3-2)	R/W	
IMS	[11:10]	I2S master (internal/external) or slave mode select. 00 = Master mode (divide mode, using PCLK) 01 = Master mode (bypass mode, using I2SCLK) 10 = Slave mode (divide mode, using PCLK) 11 = Slave mode (bypass mode, using I2SCLK) (For more information refer to Figure 10.3-2)	R/W	
TXR	[9:8]	Transmit or receive mode select. 00 = Transmit only mode 01 = Receive only mode 10 = Transmit and receive simultaneous mode 11 = Reserved	R/W	
LRP	[7]	Left/ Right channel clock polarity select. 0 = Low for left channel and high for right channel 1= High for left channel and low for right channel	R/W	
SDF	[6:5]	Serial data format. 00 = I2S format 01 = MSB-justified (left-justified) format 10 = LSB-justified (right-justified) format 11 = Reserved	R/W	
RFS	[4:3]	I2S root clock (codec clock) frequency select. 00 = 256 fs, where fs is sampling frequency 01 = 512 fs 10 = 384 fs 11 = 768 fs	R/W	

I2SMOD	Bit	Description	R/W	Reset Value
BFS	[2:1]	Bit clock frequency select. 00 = 32 fs, where fs is sampling frequency 01 = 48 fs 10 = 16 fs 11 = 24 fs	R/W	
Reserved	[0]	Reserved	R/W	

8.2.3 I2S Interface FIFO Control Register (I2SFIC, R/W, Address = 0XF210_0008, 0XF220_0008)

- I2SFIC1, R/W, Address = 0XF210_0008
- I2SFIC2, R/W, Address = 0XF220_0008

I2SFIC	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	R/W	
TFLUSH	[15]	TX FIFO flush command. 0: No flush, 1: Flush	R/W	
Reserved	[14:13]	Reserved	R/W	
FTXCNT	[12:8]	TX FIFO data count. FIFO has 16 dept, therefore value ranges from 0 to 16. N: Data count N of FIFO	R	
RFLUSH	[7]	RX FIFO flush command. 0 = No flush, 1 = Flush	R/W	
Reserved	[6:5]	Reserved	R/W	
FRXCNT	[4:0]	RX FIFO data count. FIFO has 16 dept, therefore value ranges from 0 to 16. N: Data count N of FIFO	R	

8.2.4 I2S Interface Clock Divider Control Register (I2SPSR, R/W, Address = 0XF210_000C, 0XF220_000C)

- I2SPSR1, R/W, Address = 0XF210_000C
- I2SPSR2, R/W, Address = 0XF220_000C

I2SPSR	Bit	Description	R/W	Reset Value
Reserved	[31:16]	Reserved	R/W	
PSRAEN	[15]	Pre-scaler (Clock divider) A active. 0 = Inactive, 1 = Active	R/W	
Reserved	[14]	Reserved	R/W	

I2SPSR	Bit	Description	R/W	Reset Value
PSVALA	[13:8]	Pre-scaler (Clock divider) A division value. N: Division factor is N+1	R/W	
Reserved	[7:0]	Reserved	R/W	

8.2.5 I2S Interface Transmit Data Register (I2STXD, W, Address = 0XF210_0010, 0XF220_0010)

- I2STXD1, W, Address = 0XF210_0010
- I2STXD2, W, Address = 0XF220_0010

I2STXD	Bit	Description	R/W	Reset Value
I2STXD	[31:0]	TX FIFO write data. Note that the left/right channel data is allocated as the following bit fields (For more information refer to Figure10.3-4 ~10.3-5). R(L) [23:0] when 24-bit BLC R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC	W	

8.2.6 I2S Interface Receive Data Register (I2SRXD, R, Address = 0XF210_0014, 0XF220_0014)

- I2SRXD1, R, Address = 0XF210_0014
- I2SRXD2, R, Address = 0XF220_0014

I2SRXD	Bit	Description	R/W	Reset Value
I2SRXD	[31:0]	RX FIFO read data. Note that the left/right channel data is allocated as the following bit fields (For more information refer to Figure 10.3-6 ~10.3-7) R(L)[23:0] when 24-bit BLC R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC	R	

10.4 AC97 CONTROLLER

This chapter describes the functions and usage of AC97 Controller in S5PC100 RISC microprocessor.

1 OVERVIEW

The AC97 Controller Unit of the S5PC100 supports the AC97 revision 2.0 features. AC97 Controller communicates with AC97 Codec using Audio Controller Link (AC-link). The Controller sends the stereo PCM data to Codec. The external Digital-to-Analog Converter (DAC) in the Codec converts the audio sample to an analog audio waveform. The Controller receives the stereo PCM data and the mono Mic data from Codec then stores in memories. This chapter describes the programming model for the AC97 Controller Unit. The prerequisite to understand this chapter requires a knowledge of the AC97 revision 2.0 specifications.

2 FEATURE

The AC97 Controller includes the following features:

- Independent channels for stereo PCM In, stereo PCM Out, and mono MIC In.
- DMA-based operation and interrupt based operation.
- All channels support only 16-bit samples.
- Variable sampling rate AC97 Codec interface (48 kHz and below)
- 16-bit, 16 entry FIFOs per channel
- Only primary Codec support

3 AC97 CONTROLLER OPERATION

This section explains the AC97 Controller operation such as AC-Link, Power-down sequence and Wake-up sequence.

3.1 BLOCK DIAGRAM

Figure 10.4-1 shows the functional block diagram of S5PC100 AC97 Controller. The AC97 signals form the AC-link, which is a point-to-point synchronous serial inter-connecting that supports full-duplex data transfers. All digital audio streams and command/ status information are communicated through AC-link.

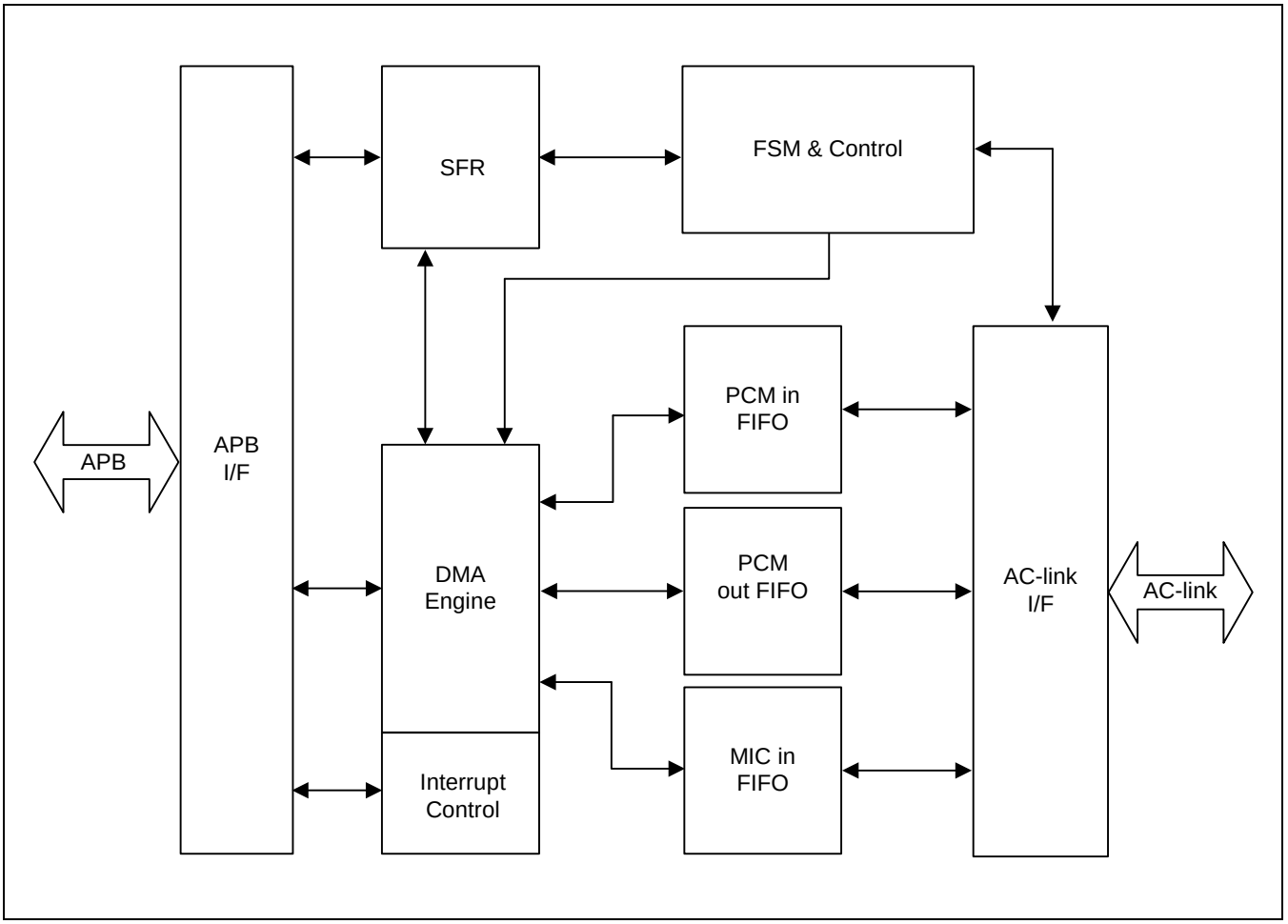


Figure 10.4-1 AC97 Block Diagram

3.2 INTERNAL DATA PATH

Figure 10.4-2 shows the internal data path of S5PC100 AC97 Controller. It has stereo Pulse Code Modulated (PCM) In, Stereo PCM Out and mono Mic-in buffers, which consist of 16-bit and 16 entries buffer. It also has 20-bit I/O shift register via AC-link.

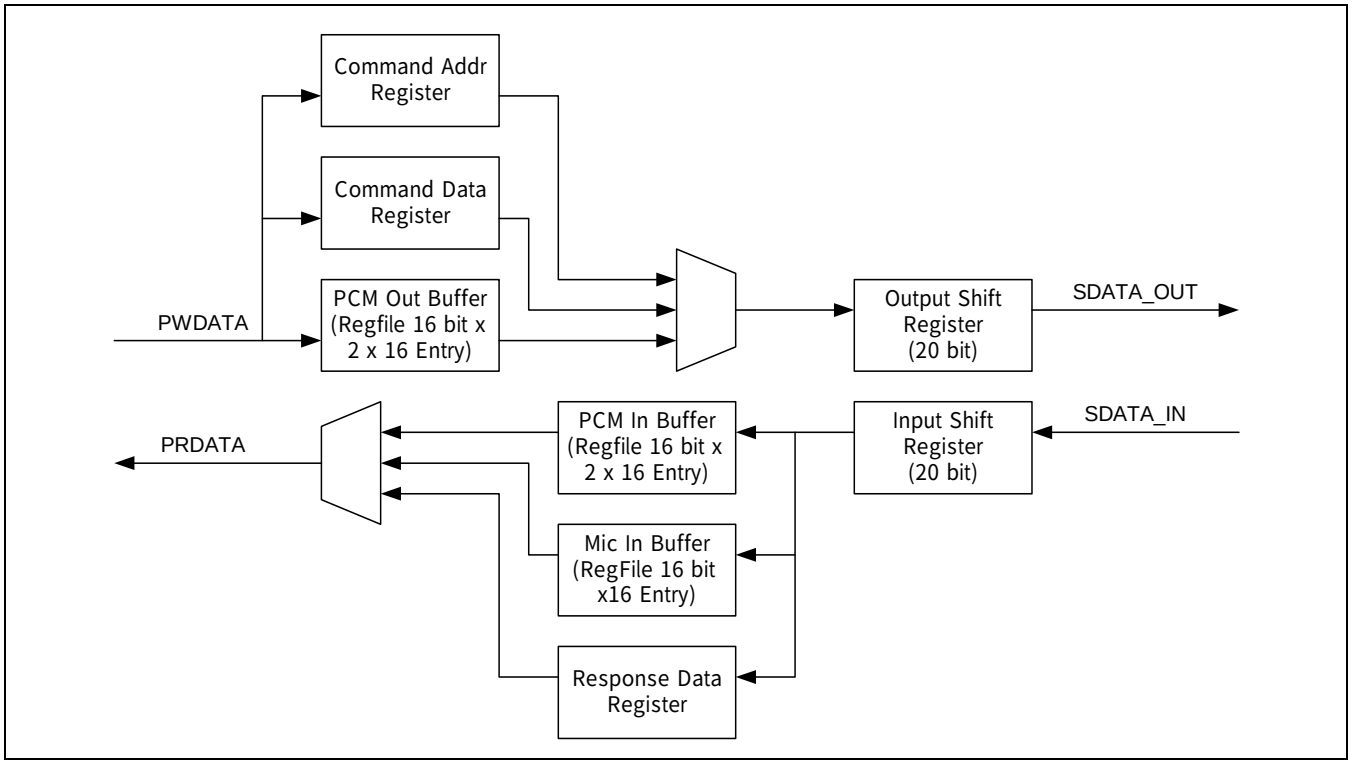


Figure 10.4-2 Internal Data Path

3.3 OPERATION FLOW CHART

If you initialize the AC97 controller, you must assert system reset or cold reset, because we do not know the previous state of the external AC97 audio-codec. This assures that GPIO is ready. Then you enable the codec ready interrupt. Check codec ready interrupt by polling or interrupt. If interrupt occurs, you must de-assert codec ready interrupt. Now transmit data from memory to register or from register to memory by using DMA or PIO (directly to write data to register). If internal FIFOs (TX FIFO or RX FIFO) are not empty, then let data be transmitted. In addition, you can previously turn on AC-Link.

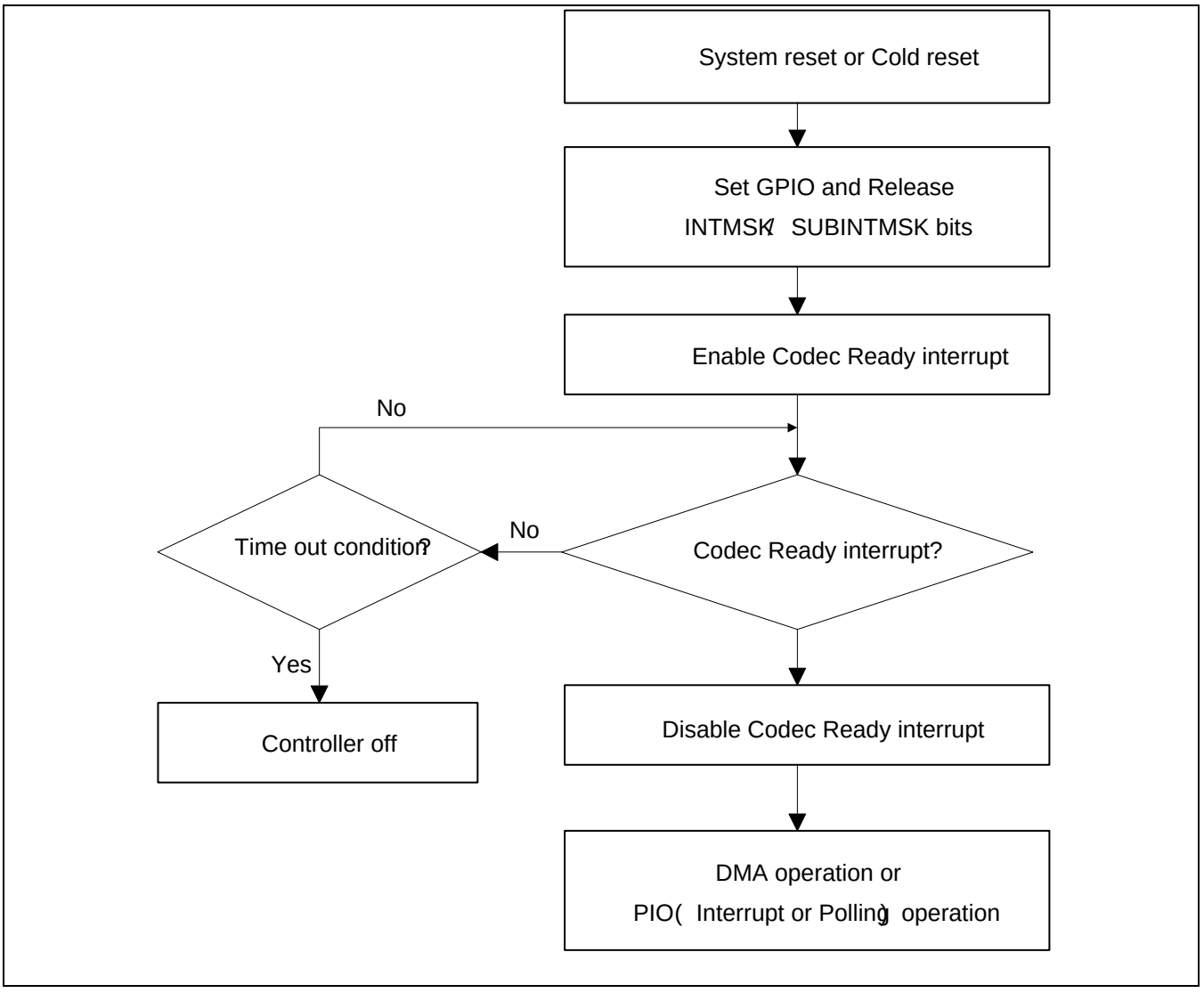


Figure 10.4-3 AC97 Operation Flow Chart

3.4 AC-LINK DIGITAL INTERFACE PROTOCOL

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the S5PC100 AC97 Controller. AC-link is a full-duplex, fixed-clock and PCM digital stream. It employs a Time Division Multiplexed (TDM) scheme to handle control register accesses and multiple input and output audio streams. The AC-link architecture divides each audio frame into 12 outgoing and 12 incoming data streams. Each stream has 20-bit sample resolution and requires a DAC and an Analog-to-Digital Converter (ADC) with a minimum 16-bit resolution.

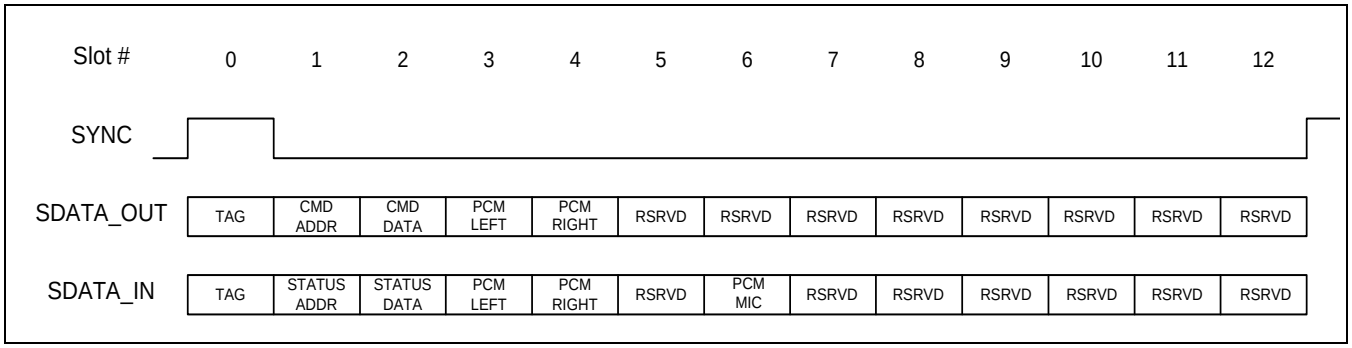


Figure 10.4-4 Bi-directional AC-link Frame with Slot Assignments

Figure 10.4-4 shows the slot definitions supported by S5PC100 AC97 Controller. The S5PC100 AC97 Controller provides synchronization for all data transaction on the AC-link.

A data transaction is made up of 256 bits of information broken into groups of 13 time slots and is called a frame. Time slot 0 is called the Tag Phase and it is 16 bits long. The other 12 time slots are called the Data Phase. The Tag Phase contains one bit that identifies a valid frame and 12 bits that identify the time slots in the Data Phase that contain valid data. Each time slot in the Data Phase is 20 bits long. A frame begins when SYNC goes high. The amount of time that SYNC is high corresponds to the Tag Phase. AC97 frames occur at fixed 48 kHz intervals and are synchronous to the 12.288 MHz bit rate clock, BITCLK. The controller and the Codec use the SYNC and BITCLK to determine when to send transmit data and when to sample received data. A transmitter transitions the serial data stream on each rising edge of BITCLK and a receiver samples the serial data stream on falling edges of BITCLK. The transmitter must tag the valid slots in its serial data stream. The valid slots are tagged in slot 0. Serial data on the AC-link is ordered most significant bit (MSB) to least significant bit (LSB). The Tag Phase's first bit is bit 15 and the first bit of each slot in Data Phase is bit 19. The last bit in any slot is bit 0.

3.4.1 AC-link Output Frame (SDATA_OUT)

Slot 0: Tag Phase

In slot 0, the first bit is a bit (SDATA_OUT, bit 15) which represents the validity of the entire frame. If bit 15 is a 1, the current frame contains at least a valid time slot. The next 12-bit positions correspond each 12 time slot contains valid data. Bits 0 and 1 of slot 0 are used as CODEC IO bits for I/O reads and writes to the CODEC registers as described in the next section. In this way, data streams of differing sample rate are transmitted across AC-link at its fixed 48 kHz audio frame rate.

Slot 1: Command Address Port

In slot 1, it communicates control register address and write/ read command information to the AC97 controller. If software accesses the primary CODEC, the hardware configures the frame as follows:

- In slot 0, the valid bit for 1 and 2 slots are set.
- In slot 1, bit 19 is set (read) or clear (write). Bits 18-12 (of slot 1) are configured to specify the index to the CODEC register. Others are filled with 0's (reserved).
- In slot 2, it configured with the data which is for writing because of output frame.

Slot 2: Command Data Port

In slot 2, this is the write data with 16-bit resolution ([19:4] is valid data)

Slot 3: PCM Playback Left channel

Slot 3 which is audio output frame is the composite digital audio left stream. If a sample has a resolution that is less than 16 bits, the AC97 controller fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Playback Right channel

Slot 4 which is audio output frame is the composite digital audio right stream. If a sample has a resolution that is less than 16 bits, the AC97 controller fills all training non-valid bit positions in the slot with zeroes.

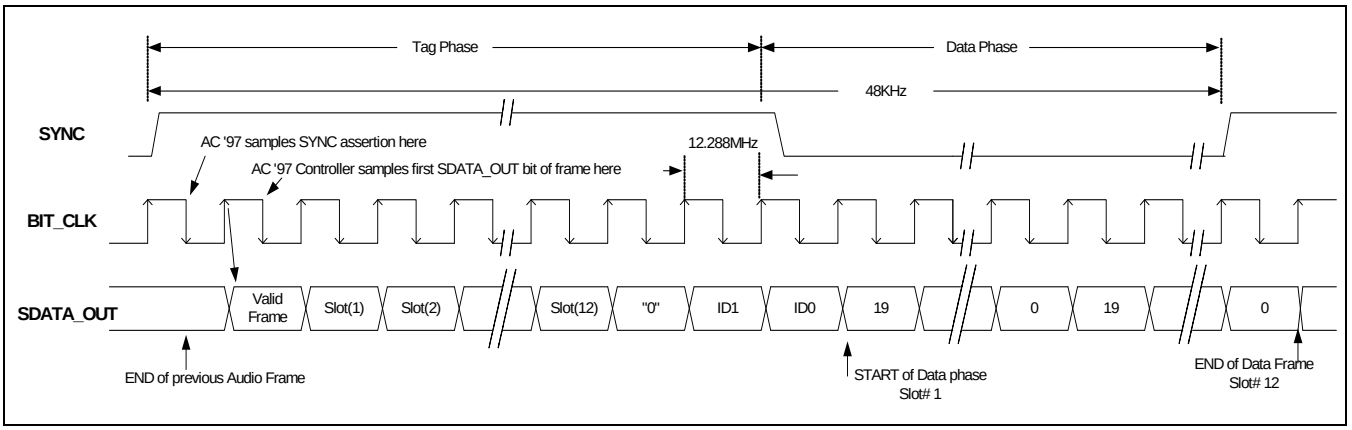


Figure 10.4-5 AC-link Output Frame

3.5 AC-LINK INPUT FRAME (SDATA_IN)

Slot 0: Tag Phase

In slot 0, the first bit (SDATA_OUT, bit 15) indicates whether the AC97 controller is in the CODEC ready state. If the CODEC Ready bit is 0, it means that the AC97 controller is not ready for normal operation. This condition is normal after the power is de-asserted on reset and the AC97 controller voltage references are settling.

Slot 1: Status Address Port/SLOTREQ bits

The status port monitors the status for the AC97 controller functions. It is not limited to mixer settings and power management. Audio input frame slot 1s stream echoes the control register index for the data to be returned in slot 2, if the controller tags slots 1 and 2 as valid during slot 0. The controller only accepts status data if the accompanying status address matches the last valid command address issued during the most recent read command. For multiple sample rate output, the CODEC examines its sample-rate control registers, its FIFOs' states, and the incoming SDATA_OUT tag bits at the beginning of each audio output frame to determine which SLOTREQ bits to set active (low). SLOTREQ bits asserted during the current audio input frame indicates which output slots require data from the controller in the next audio output frame. For fixed 48 kHz operation, the SLOTREQ bits are set active (low), and a sample is transferred in each frame. For multiple sample-rate input, the "tag" bit for each input slot indicates whether valid data is present.

Table 10.4-1 Input Slot 1 Bit Definitions

Bit	Description
19	RESERVED (Filled with zero)
18-12	Control register index (Filled with zeroes if AC97 tags is invalid)
11	Slot 3 request : PCM Left channel
10	Slot 4 request : PCM Right channel
9	Slot 5 request : NA
8	Slot 6 request : MIC channel
7	Slot 7 request : NA
6	Slot 8 request : NA
5	Slot 9 request : NA
4	Slot 10 request : NA
3	Slot 11 request : NA
2	Slot 12 request : NA
1, 0	RESERVED (Filled with zero)

Slot 2: Status Data Port

In slot 2, this is the status data with 16-bit resolution ([19:4] is valid data)

Slot 3: PCM Record Left channel

Slot 3 which is audio input frame is the left channel audio output of the AC97 Codec. If a sample has a resolution that is less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 4: PCM Right channel audio

Slot 4 which is audio input frame is the right channel audio output of the AC97 Codec. If a sample has a resolution less than 16 bits, the AC97 Codec fills all training non-valid bit positions in the slot with zeroes.

Slot 6: Microphone Record Data

The AC97 Controller only supports 16-bit resolution for the MIC-in channel.

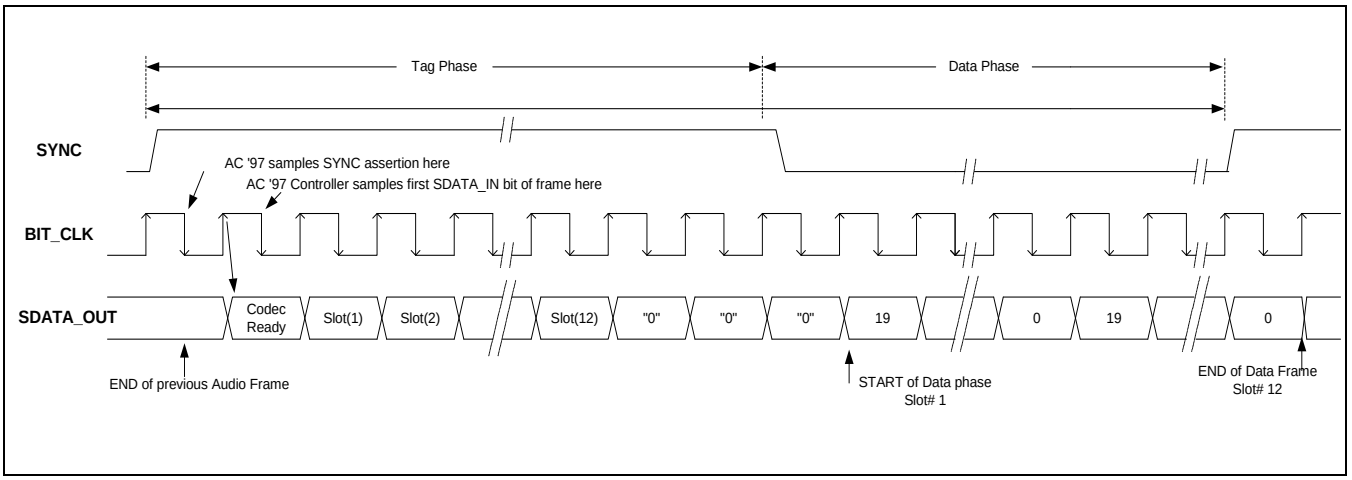


Figure 10.4-6 AC-link Input Frame

3.6 AC97 POWER-DOWN

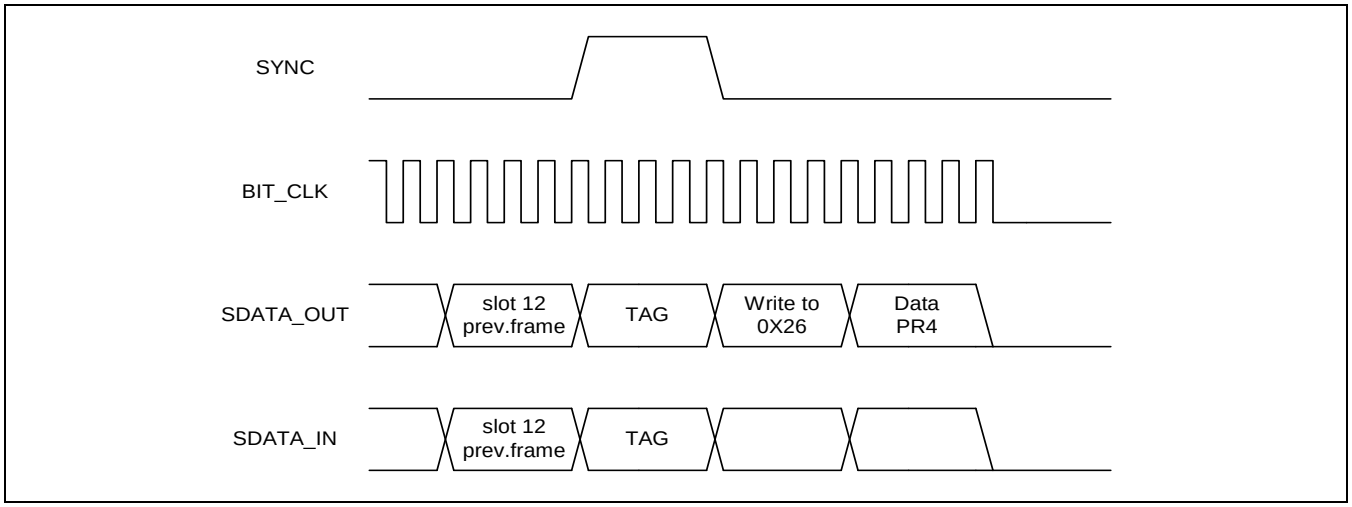


Figure 10.4-7 AC97 Power-down Timing

3.6.1 Powering Down the AC-link

The AC-link signals enter a low power mode if the AC97 Codec Power-down register (0x26) bit PR4 is set to 1 (by writing 0x1000). Then the Primary Codec drives both BITCLK and SDATA_IN to a logic low voltage level. The sequence follows the timing diagram as shown in Figure 10.4-7.

The AC97 Controller transmits the write to Power-down register (0x26) over the AC-link. Set up the AC97 Controller so that it does not transmit data to slots 3-12 when it writes to the Power-down register bit PR4 (data 0x1000), and it does not require the Codec to process other data when it receives a power down request. If the Codec processes the request it immediately transitions BITCLK and SDATA_IN to a logic low level. The AC97 Controller drives SYNC and SDATA_OUT to a logic low level after programming the AC_GLBCTRL register.

3.6.2 Waking up the AC-link - Wake Up Triggered by the AC97 Controller

AC-link protocol provides a cold AC97 reset and a warm AC97 reset. The current power-down state ultimately dictates which AC97 reset is used. Registers must stay in the same state during all power-down modes unless a cold AC97 reset is performed. In a cold AC97 reset, the AC97 registers are initialized to their default values. After a power down, the AC-link must wait for a minimum of four audio frame times after the frame in which the power down occurred before it can be reactivated by reasserting the SYNC signal. If AC-link powers up, it indicates readiness through the Codec ready bit (input slot 0, bit 15).

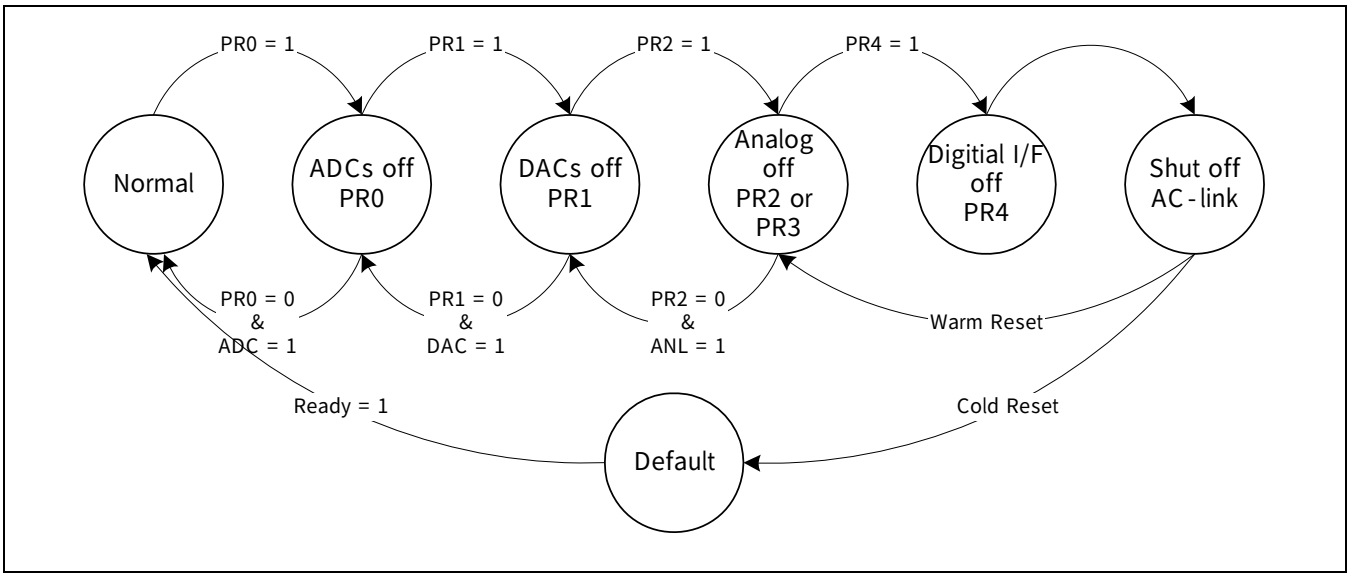


Figure 10.4-8 AC97 Power down/Power up Flow

3.6.3 Cold AC97 Reset

A cold reset is generated if the nRESET pin is asserted through the AC_GLBCTRL. Asserting and deasserting nRESET activates BITCLK and SDATA_OUT. All AC97 control registers are initialized to their default power on reset values. nRESET is an asynchronous AC97 input.

3.6.4 Warm AC97 Reset

A Warm AC97 reset reactivates the AC-link without altering the current AC97 register values. A warm reset is generated if BITCLK is absent and SYNC is driven high. In normal audio frames, SYNC is a synchronous AC97 input. If BITCLK is absent, SYNC is treated as an asynchronous input used to generate a warm reset to AC97.The AC97 Controller must not activate BITCLK until it samples SYNC low again. This prevents a new audio frame from being falsely detected.

3.7 AC97 STATE DIAGRAM

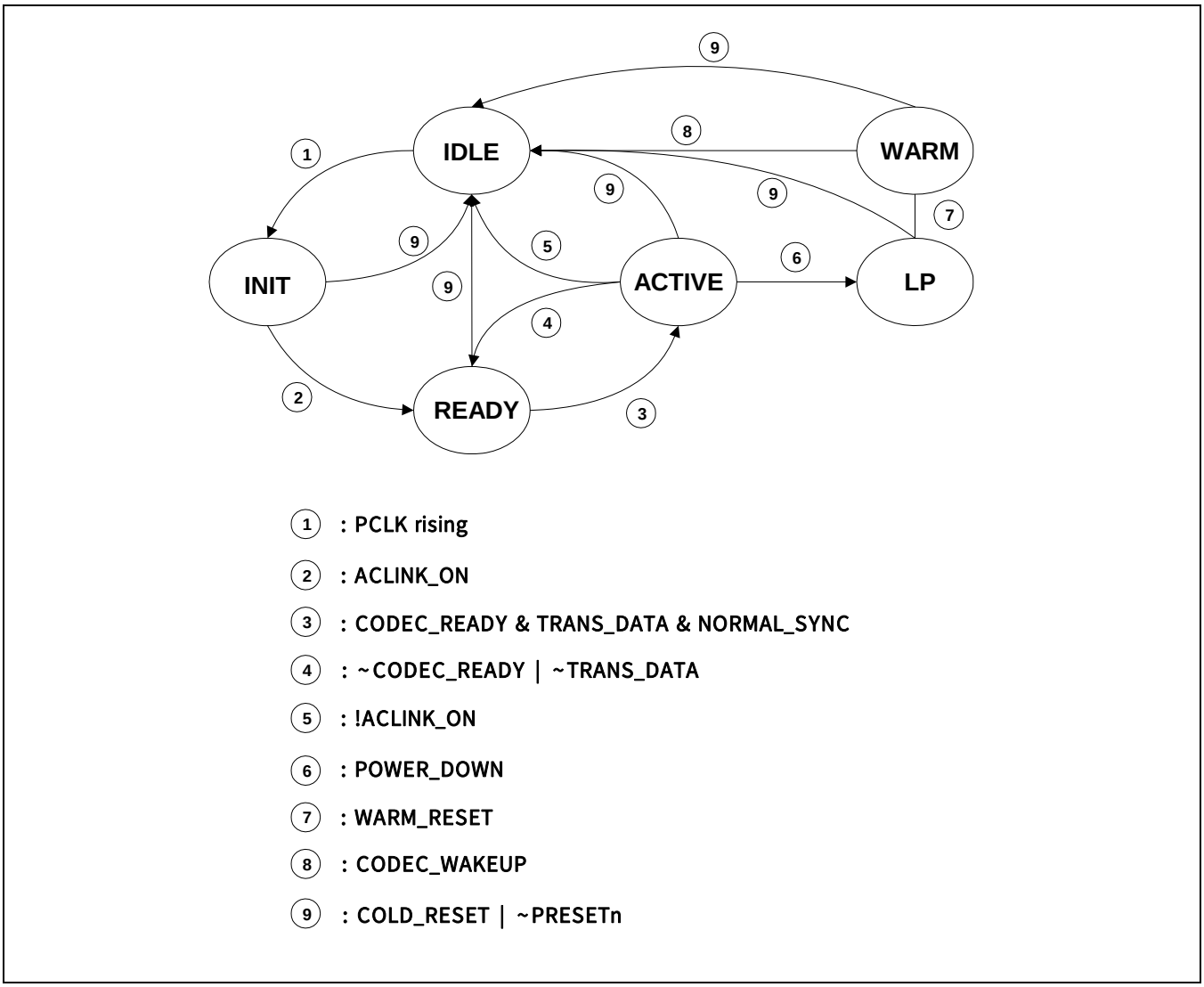


Figure 10.4-9 AC97 State Diagram

This is the state diagram of AC97 controller. It is helpful to understand AC97 controller state machine. State above figure is synchronized by peripheral clock (PCLK). It is able to monitor state at AC_GLBSTAT register.

4 I/O DESCRIPTION

AC97 external pads are shared with I2S. In order to use these pads for AC97, GPIO must be set before the AC97 started. For mode information, refer to the Chapter GPIO

Funtion Signal	I/O	Description	Pad	Type
AC97_RESETn	Output	Active-low CODEC Reset.	Xi2s1CDCLK	muxed
AC97_BITCLK	Input	12.288MHz Bit-Rate Clock.	Xi2s1SCLK	muxed
AC97_SYNC	Output	48 kHz Frame Indicator and Synchronizer	Xi2s1LRCK	muxed
AC97_SDO	Output	Serial Audio Output Data.	Xi2s1SDO	muxed
AC97_SDI	Input	Serial Audio Input Data.	Xi2s1SDI	muxed

5 REGISTER DESCRIPTION

5.1 REGISTER OVERVIEW

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0xF230_0000	R/W	AC97 Global Control Register	0x00000000
AC_GLBSTAT	0xF230_0004	R	AC97 Global Status Register	0x00000001
AC_CODEC_CMD	0xF230_0008	R/W	AC97 Codec Command Register	0x00000000
AC_CODEC_STAT	0xF230_000C	R	AC97 Codec Status Register	0x00000000
AC_PCMADDR	0xF230_0010	R	AC97 PCM Out/In Channel FIFO Address Register	0x00000000
AC_MICADDR	0xF230_0014	R	AC97 Mic In Channel FIFO Address Register	0x00000000
AC_PCMDATA	0xF230_0018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000
AC_MICDATA	0xF230_001C	R/W	AC97 MIC In Channel FIFO Data Register	0x00000000

5.2 DETAILED DESCRIPTION

5.2.1 AC97 Global Control Register (AC_GLBCTRL, R/W, Address = 0xF230_0000)

This is the global register of the AC97 controller. There are interrupt control registers, DMA control registers, AC-Link control register, data transmission control register and related reset control register.

AC_GLBCTRL	Bit	Description	Reset Value
Reserved	[31]	Reserved	0
Codec ready interrupt clear	[30]	1 = Interrupt clear (write only)	0
PCM out channel underrun interrupt clear	[29]	1 = Interrupt clear (write only)	0
PCM in channel overrun interrupt clear	[28]	1 = Interrupt clear (write only)	0
Mic in channel overrun interrupt clear	[27]	1 = Interrupt clear (write only)	0
PCM out channel threshold interrupt clear	[26]	1 = Interrupt clear (write only)	0
PCM in channel threshold interrupt clear	[25]	1 = Interrupt clear (write only)	0
MIC in channel threshold interrupt clear	[24]	1 = Interrupt clear (write only)	0
Reserved	[23]	Reserved	0
Codec ready interrupt enable	[22]	0 = Disables 1 = Enables	0
PCM out channel underrun interrupt enable	[21]	0 = Disables 1 = Enables (FIFO is empty)	0
PCM in channel overrun interrupt enable	[20]	0 = Disables 1 = Enables (FIFO is full)	0

AC_GLBCTRL	Bit	Description	Reset Value
Mic in channel overrun interrupt enable	[19]	0 = Disables 1 = Enables (FIFO is full)	0
PCM out channel threshold interrupt enable	[18]	0 = Disables 1 = Enables (FIFO is half empty)	0
PCM in channel threshold interrupt enable	[17]	0 = Disables 1 = Enables (FIFO is half full)	0
MIC in channel threshold interrupt enable	[16]	0 = Disables 1 = Enables (FIFO is half full)	0
Reserved	[15:14]	Reserved	00
PCM out channel transfer mode	[13:12]	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
PCM in channel transfer mode	[11:10]	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
MIC in channel transfer mode	[9:8]	00 = Off 01 = PIO 10 = DMA 11 = Reserved	00
Reserved	[7:4]	Reserved	0000
Transfer data enable using AC-link	[3]	0 = Disables 1 = Enables	0
AC-Link on	[2]	0 = Off 1 = SYNC signal transfer to Codec	0
Warm reset	[1]	0 = Normal 1 = Wake up codec from power down	0
Cold reset	[0]	0 = Normal 1 = Reset Codec and Controller logic	0

5.2.2 AC97 Global Status Register (AC_GLBSTAT, R, Address = 0xF230_0004)

This is the status register. If the interrupt occurs, check what the interrupt source is.

AC_GLBSTAT	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0x00
Codec ready interrupt	[22]	0 = Not requested 1 = Requested	0
PCM out channel underrun interrupt	[21]	0 = Not requested 1 = Requested	0
PCM in channel overrun interrupt	[20]	0 = Not requested 1 = Requested	0
MIC in channel overrun interrupt	[19]	0 = Not requested 1 = Requested	0
PCM out channel threshold interrupt	[18]	0 = Not requested 1 = Requested	0
PCM in channel threshold interrupt	[17]	0 = Not requested 1 = Requested	0
MIC in channel threshold interrupt	[16]	0 = Not requested 1 = Requested	0
Reserved	[15:3]	Reserved	0x000
Controller main state	[2:0]	000 = Idle 001 = Init 010 = Ready 011 = Active 100 = LP 101 = Warm	001

5.2.3 AC97 Codec Command Register (AC_CODEC_CMD, R/W, Address = 0xF230_0008)

When you control writing or reading, you must set the Read enable bit. If you want to write data to the AC97 Codec, set the index (or address) of the AC97 Codec and data.

AC_CODEC_CMD	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0x00
Read enable	[23]	0 = Command write (1) 1 = Status read	0
Address	[22:16]	Codec command address	0x00
Data	[15:0]	Codec command data	0x0000

NOTE: If the commands are written on the AC_CODDEC_CMD register, It is recommended to set the delay time between the command and the next command as more than 1 / 48kHz.

5.2.4 AC97 Codec Status Register (AC_CODEC_STAT, R, Address = 0xF230_000C)

If the Read enable bit is 1 and Codec command address is valid, Codec status data is also valid.

AC_CODEC_STAT	Bit	Description	Reset Value
Reserved	[31:23]	Reserved	0x00
Address	[22:16]	Codec status address	0x00
Data	[15:0]	Codec status data	0x0000

NOTE: If you want to read data from AC97 codec register via the AC_CODDEC_STAT register, you must take the following steps.

1. Write command address and data on the AC_CODEC_CMD register with Bit[23] =1.
2. Have a delay time.
3. Read command address and data from AC_CODEC_STAT register.

5.2.5 AC97 PCM Out/In Channel FIFO Address Register (AC_PCMADDR, R, Address = 0xF230_0010)

To index the internal PCM FIFOs address.

AC_PCMADDR	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0000
Out read address	[27:24]	PCM out channel FIFO read address	0000
Reserved	[23:20]	Reserved	0000
In read address	[19:16]	PCM in channel FIFO read address	0000
Reserved	[15:12]	Reserved	0000
Out write address	[11:8]	PCM out channel FIFO write address	0000
Reserved	[7:4]	Reserved	0000
In write address	[3:0]	PCM in channel FIFO write address	0000

5.2.6 AC97 MIC In Channel FIFO Address Register (AC_MICADDR, R, Address = 0xF230_0014)

To index the internal MIC-in FIFO address.

AC_MICADDR	Bit	Description	Reset Value
Reserved	[31:20]	Reserved	0000
Read address	[19:16]	MIC in channel FIFO read address	0000
Reserved	[15:4]	Reserved	0x000
Write address	[3:0]	MIC in channel FIFO write address	0000

5.2.7 AC97 PCM Out/In Channel FIFO Data Register (AC_PCMDATA, R/W, Address = 0xF230_0018)

This is PCM out/in channel FIFO data register.

AC_PCMDATA	Bit	Description	Reset Value
Right data	[31:16]	PCM out/in right channel FIFO data Read = PCM in right channel Write = PCM out right channel	0x0000
Left data	[15:0]	PCM out/in left channel FIFO data Read = PCM in left channel Write = PCM out left channel	0x0000

5.2.8 AC97 MIC In Channel FIFO Data Register (AC_MICDATA, R/W, Address = 0xF230_001C)

This is MIC-in channel FIFO data register.

AC_MICDATA	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0x0000
Mono data	[15:0]	MIC in mono channel FIFO data	0x0000

NOTES

10.5 PCM AUDIO INTERFACE

1 PCM OVERVIEW

The PCM Audio Interface module provides PCM bi-directional serial interface to an external Codec.

1.1 FEATURE

The PCM Audio interface includes the following features:

- ARM APB interface
- Master mode only, this block always sources the main shift clock
- All PCM serial timings and strobes including the main shift clock, are based on an external PCM Audio clock input
- Optional timing based on the internal APB PCLK
- Input and output FIFOs to buffer data
- Optional DMA interface for Tx and/ or Rx

2 PCM AUDIO INTERFACE

The PCM Audio Interface provides a serial interface to an external Codec. The PCM module receives an input PCMCODEC_CLK that is used to generate the serial shift timing. The PCM interface outputs a serial data out, a serial shift clock, and a sync signal. Data is received from the external Codec over a serial input line. The serial data in, serial data out, and sync signal are all synchronized to the serial shift clock.

The serial shift clock, PCMSCLK, is generated from a programmable divide of the input PCMCODEC_CLK. The sync signal, PCMSYNC, is generated based upon a programmable number of serial clocks and is one serial clock wide.

The PCM data words are 16-bit wide, serially shifted out 1-bit per PCMSCLK. Only one 16-bit word is shifted out for each PCMSYNC. The PCMSCLK continues to toggle even after all 16-bit have been shifted out. The PCMSOUT data will be a undefined after the 16-bit word is complete. The next PCMSYNC signals the start of the next PCM data word.

The TX FIFO provides the 16-bit data word to be serially shifted out. This data is serially shifted out MSB first, one bit per PCMSCLK. The rising edge of the PCMSCLK is used to clock out PCM serial output data (PCMSOUT). The MSB bit position relative to the PCMSYNC is programmable to be either coincident with the PCMSYNC or one PCMCLK later. After all 16-bit have been shifted out, generation of an interrupt is optional to indicate the end of the transfer.

At the same time data is being shifted out, the PCMSIN input is used to serially shift data in from the external codec. The data is received MSB first and is clocked on the falling edge of PCMSCLK. The position of the first bit is programmable to correspond with the PCMSYNC or one PCMSCLK later.

The first 16-bit are serially shifted into the PCM_DATAIN register which is then loaded into the RX FIFO. Subsequent bits are ignored until the next PCMSYNC.

Various Interrupts are available to indicate the status of the RX and TX FIFO. Each FIFO has a programmable flag to indicate if the CPU needs to service the FIFO. For the RX FIFO there is an interrupt which is raised if the FIFO exceeds a certain programmable almost_full depth. Similarly there is a programmable almost_empty interrupt for the TX FIFO.

3 PCM TIMING

The following figures show the timing relationship for the PCM transfers. Note in all cases, the PCM shift timing is derived by dividing the input clock, PCMCODEC_CLK. While the timing is based upon the PCMCODEC_CLK, there is no attempt to realign the rising edge of the output PCMSCLK with the original PCMCODEC_CLK input clock. These edges are skewed by internal delay through the pads as well as the divider logic. This does not represent a problem because the actual shift clock, PCMSCLK, is output with the data. Furthermore, even if the PCMSCLK output is not used, the skew is significantly less than the period of the PCMCODEC_CLK and should not represent a problem since most PCM interfaces capture data on the falling edge of the clock.

Figure 10.5-1 shows a PCM transfer with the MSB configured to correspond with the PCMSYNC. This MSB positioning corresponds to setting the MSB_POS_WR and MSB_POS_RD bits in DSP_PCMCTL register to be LOW.

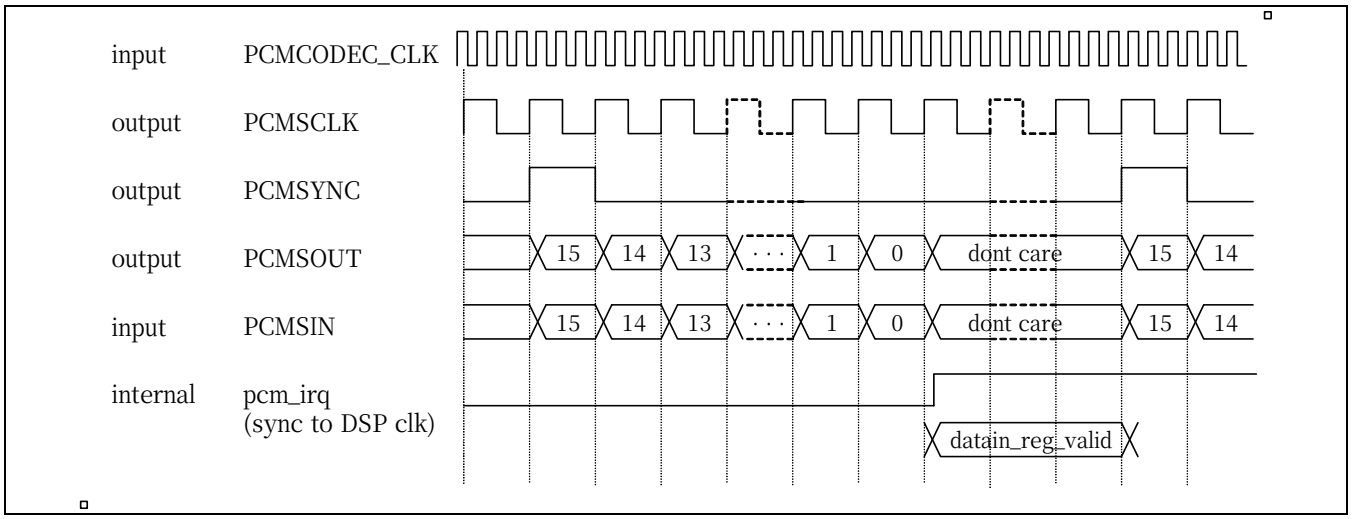


Figure 10.5-1 PCM Timing, POS_MSB_WR/RD = 0

Figure 10.5-2 shows a PCM transfer with the MSB configured one shift clock after the PCMSYNC. This MSB positioning corresponds to setting the MSB_POS_WR and MSB_POS_RD bits in DSP_PCMCTL register to be HIGH.

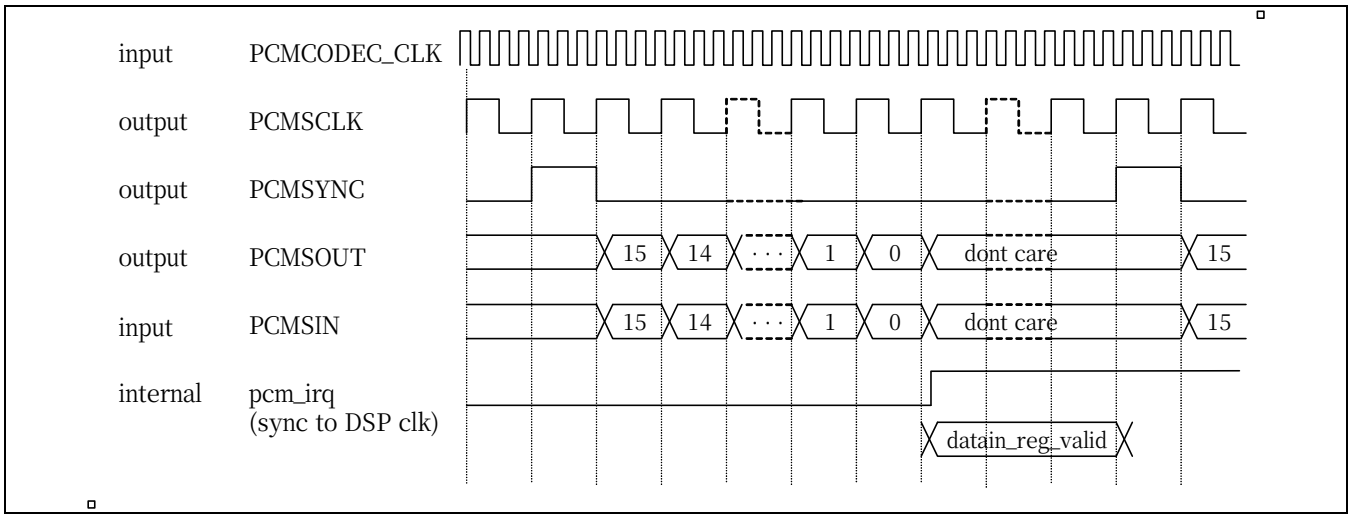


Figure 10.5-2 PCM Timing, POS_MSB_WR/RD = 1

4 PCM INPUT CLOCK DIAGRAM

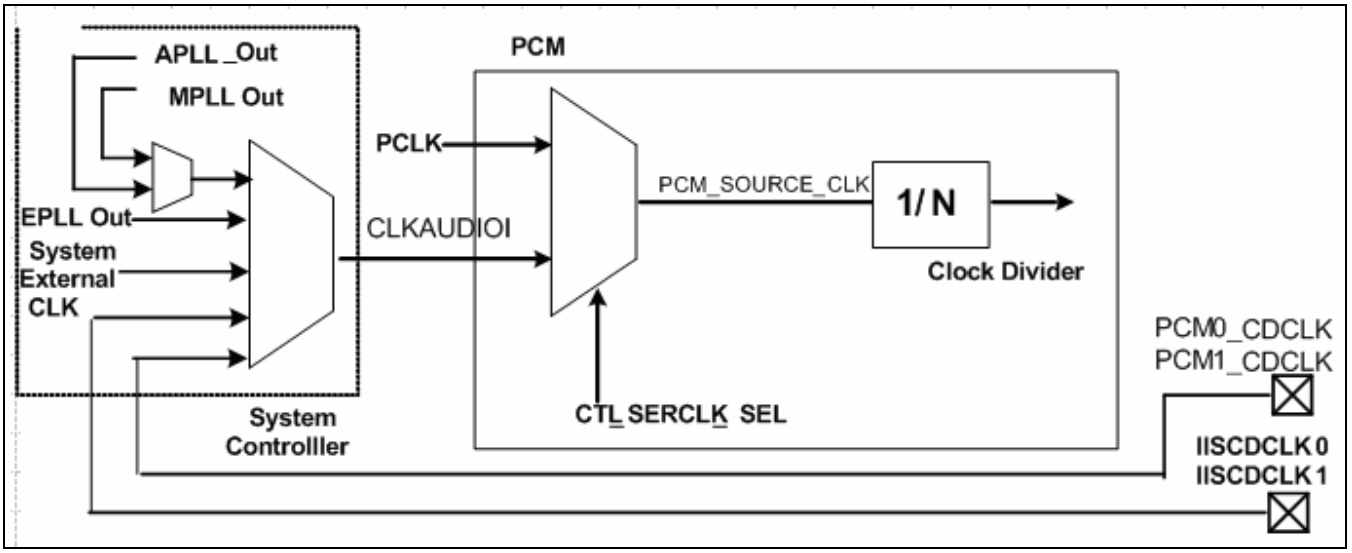


Figure 10.5-3 Input Clock Diagram for PCM

S5PC100X provideS PCM with a variety of clock. As described in the Figure 10.5-3, PCM interface is able to select clock from two PCLK or AUDIO clock (CLKAUDIOI) which is from system controller. You can select CLKAUDIO among PLLs or external input clock (PCMCDCLK and IISCDCLK). To select CLKAUDIO, please refer to the Clock Controller.

5 I/O DESCRIPTIONS

Funtion Signal	I/O	Description	Pad	Type
PCM0_SCLK	Output	PCM0 Serial Shift Clock	Xmmc2CLK	muxed
PCM0_EXTCLK	Input	PCM0 External Clock source	Xmmc2CMD	muxed
PCM0_FSYNC	Output	PCM0 Serial Data Indicator and Synchronizer	Xmmc2DATA[0]	muxed
PCM0_SIN	Input	PCM0 Serial Input Data	Xmmc2DATA[1]	muxed
PCM0_SOUT	Output	PCM0 Serial Output Data	Xmmc2DATA[2]	muxed
PCM1_SCLK	Output	PCM1 Serial Shift Clock	Xi2s1SCLK	muxed
PCM1_EXTCLK	Input	PCM1 External Clock Source	Xi2s1CDCLK	muxed
PCM1_FSYNC	Output	PCM1 Serial Data Indicator and Synchronizer	Xi2s1LRCLK	muxed
PCM1_SIN	Input	PCM1 Serial Input Data	Xi2s1SDI	muxed
PCM1_SOUT	Output	PCM1 Serial Output Data	Xi2s1SD0	muxed

6 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
PCM_0_CTL	0xF240_0000	R/W	PCM0 Main Control	0x00000000
PCM_0_CLKCTL	0xF240_0004	R/W	PCM0 Clock and Shift control	0x00000000
PCM_0_TXFIFO	0xF240_0008	R/W	PCM0 TxFIFO write port	0x00010000
PCM_0_RXFIFO	0xF240_000C	R/W	PCM0 RxFIFO read port	0x00010000
PCM_0_IRQ_CTL	0xF240_0010	R/W	PCM0 Interrupt Control	0x00000000
PCM_0_IRQ_STAT	0xF240_0014	R/O	PCM0 Interrupt Status	0x00000000
PCM_0_FIFO_STAT	0xF240_0018	R/O	PCM0 Tx Defualt Value	0x00000000
PCM_0_CLRINT	0xF240_0020	W/O	PCM0 Interrupt Clear	-
PCM_1_CTL	0xF250_0000	R/W	PCM1 Main Control	0x00000000
PCM_1_CLKCTL	0xF250_0004	R/W	PCM1 Clock and Shift control	0x00000000
PCM_1_TXFIFO	0xF250_0008	R/W	PCM1 TxFIFO write port	0x00010000
PCM_1_RXFIFO	0xF250_000C	R/W	PCM1 RxFIFO read port	0x00010000
PCM_1_IRQ_CTL	0xF250_0010	R/W	PCM1 Interrupt Control	0x00000000
PCM_1_IRQ_STAT	0xF250_0014	R/O	PCM1 Interrupt Status	0x00000000
PCM_1_FIFO_STAT	0xF250_0018	R/O	PCM1 Tx Defualt Value	0x00000000
PCM_1_CLRINT	0xF250_0020	W/O	PCM1 Interrupt Clear	-

6.1 PCM CONTROL REGISTER

- PCM_0_CTL, R/W, Address = 0xF240_0000
- PCM_1_CTL, R/W, Address = 0xF250_0000

The PCM_n_CTL register is used to control the various aspects of the PCM module. It also provides a status bit to provide an option to use polling instead of interrupt based control.

The bit definitions for the PCM_n_CTL Control Register are described below:

PCM_n_CTL	Bit	Description	Reset Value
Reserved	[31:19]	Reserved	
TXFIFO _DIPSTICK	[18:13]	Determines if the almost_full and almost_empty flags go active for the TXFIFO Almost_empty: fifo_depth < fifo_dipstick Almost_full: fifo_depth > (32 – fifo_dipstick) **NOTE:** If fifo_dipstick == 0 Almost_empty and Almost_full are invalid **NOTE:** For DMA loading of TX FIFO Txfifo_dipstick >= 2 This is required since the PCM_TXDMA uses almost_full as the DMA request (keep requesting data until the FIFO is almost full). In some circumstances, the DMA write one more word after the DMA_req goes away. Thus the almost_full flag most go active with at least space for one extra word in the FIFO	0
RXFIFO _DIPSTICK	[12:7]	Determines if the almost_full and almost_empty flags go active for the RXFIFO Almost_empty: fifo_depth < fifo_dipstick Almost_full: fifo_depth > (32 – fifo_dipstick) **NOTE:** If fifo_dipstick == 0 Almost_empty and Almost_full are invalid **NOTE:** For DMA, RXFIFO_DIPSTICK is a don't care DMA unloading of RX fifo uses the rx_fifo_empty flag as the DMA request **NOTE:** non-DMA IRQ/polling RXFIFO_DIPSTICK should be 0x20 This will have the effect of rx_fifo_almost_full acting as a rx_fifo_not_empty flag.	0
PCM_TX _DMA_EN	[6]	Enable the DMA interface for the TXFIFO DMA_TX request occurs whenever the TXFIFO is not almost full	0
PCM_RX _DMA_EN	[5]	Enable the DMA interface for the RXFIFO DMA_RX request occurs whenever the RXFIFO is not empty.	0

PCM_n_CTL	Bit	Description	Reset Value
TX_MSB _POS	[4]	Controls the position of the MSB bit in the serial output stream relative to the PCMSYNC signal 0: MSB sent during the same clock that PCMSYNC is high 1: MSB sent on the next PCMSCLK cycle after PCMSYNC is high	0
RX_MSB _POS	[3]	Controls the position of the MSB bit in the serial input stream relative to the PCMSYNC signal 0: MSB is captured on the falling edge of PCMSCLK during the same cycle that PCMSYNC is high 1: MSB is captured on the falling edge of PCMSCLK during the cycle after the PCMSYNC is high	0
PCM_TXFIFO _EN	[2]	Enable the TXFIFO If the enable is LOW the internal FIFOs clears and reinitialize	0
PCM_RXFIFO _EN	[1]	Enable the RXFIFO If the enable is LOW the internal FIFOs clears and reinitialize	0
PCM_PCM _ENABLE	[0]	PCM enable signal. Enables the serial shift state machines. The enable must be set HIGH for the PCM to operate. If the enable is LOW, the PCM outputs will not toggle (PCMSCLK, PCMSYNC, and PCMSOUT). Additionally if the enable is LOW, the internal divider-counters are held in reset.	0

6.2 PCM CLK CONTROL REGISTER

- PCM_0_CLKCTL, R/W, Address = 0xF240_0004
- PCM_1_CLKCTL, R/W, Address = 0xF250_0004

The bit definitions for the PCM_n_CTL Control Register are described below:

PCM_n_CLKCTL	Bit	Description	Reset Value
Reserved	[31:20]	Reserved	0
CTL_SERCLK_EN	[19]	Enables the serial clock division logic. Must be HIGH for the PCM to operate	0
CTL_SERCLK_SEL	[18]	Selects the source of the serial clock 0 - SCLK_AUDIO 1 - PCLK	0
SCLK_DIV	[17:9]	Controls the divider used to create the PCMSCLK based on the PCMCODEC_CLK Final clock is source_clk / 2*(sclk_div+1)	000
SYNC_DIV	[8:0]	Controls the frequency of the PCMSYNC signal based on the PCMSCLK.	000

6.3 THE PCM TX FIFO REGISTER

- PCM_0_TXFIFO, R/W, Address = 0xF240_0008
- PCM_1_TXFIFO, R/W, Address = 0xF250_0008

The bit definitions for the PCM_n_TXFIFO Register are described below:

PCM_n_TXFIFO	Bit	Description	Reset Value
Reserved	[31:17]	Reserved	0
TXFIFO_DVALID	[16]	TXFIFO data is valid Write: Not valid Read: TXFIFO read data valid 1 = Valid 0 = Invalid (probably read an empty fifo)	1
TXFIFO_DATA	[15:0]	TXFIFO DATA Write: TXFIFO_DATA is written into the TXFIFO Read: TXFIFO is read using the APB interface **NOTE:** reading the TXFIFO is meant to support debugging. Online the TXFIFO is read by the PCM serial shift engine, not the APB	0

6.4 PCM RX FIFO REGISTER

- PCM_0_RXFIFO, R/W, Address = 0xF240_000C
- PCM_1_RXFIFO, R/W, Address = 0xF250_000C

The bit definitions for the PCM_n_RXFIFO Register are described below:

PCM_n_RXFIFO	Bit	Description	Reset Value
Reserved	[31:17]	Reserved	0
RXFIFO_DVALID	[16]	RXFIFO data is valid Write: Not Valid Read: TXFIFO read data valid 1 = Valid 0 = Invalid (probably read an empty fifo)	1
RXFIFO_DATA	[15:0]	RXFIFO DATA Write: RXFIFO_DATA is written into the RXFIFO **NOTE:** Writing the RXFIFO is meant to support debugging. Online the RXFIFO is written by the PCM serial shift engine, not the APB Read: TXFIFO is read using the APB interface	0

6.5 PCM INTERRUPT CONTROL REGISTER

- PCM_0_IRQ_CTL, R/W, Address = 0xF240_0010
- PCM_1_IRQ_CTL, R/W, Address = 0xF250_0010

The PCM_n_IRQ_CTL register is used to control the various aspects of the PCM interrupts.

The bit definitions for the PCM_n_IRQ_CTL Control Register are described below:

PCM_n_IRQ_CTL	Bit	Description	Reset Value
Reserved	[31:15]	Reserved	0
EN_IRQ_TO_ARM	[14]	Controls whether or not the PCM interrupt is sent to the ARM 1 = PCM IRQ is forwarded to the ARM subsystem 0 = PCM IRQ is NOT forwarded to the ARM subsystem	0
Reserved	[13]	Reserved	0
TRANSFER_DONE	[12]	Interrupt is generated every time the serial shift for a word completes 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_EMPTY	[11]	Interrupt is generated if the TxFIFO is empty 1 = Enables IRQ source 0 = Disables IRQ source	0

PCM_n_IRQ_CTL	Bit	Description	Reset Value
TXFIFO_ALMOST _EMPTY	[10]	Interrupt is generated if the TxFIFO is ALMOST empty. Almost empty is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_FULL	[9]	Interrupt is generated if the TxFIFO is full 1= Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_ALMOST _FULL	[8]	Interrupt is generated if the TxFIFO is ALMOST full. Almost full is defined as FIXME words remaining 1= Enables IRQ source enabled 0 = Disables IRQ source	0
TXFIFO_ERROR _STARVE	[7]	Interrupt is generated for TxFIFO starve ERROR. This occurs whenever the TxFIFO is read when it is still empty. This is considered as an ERROR and will have unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_ERROR _OVERFLOW	[6]	Interrupt is generated for TxFIFO overflow ERROR. This occurs if the TxFIFO is written when it is already full. This is considered as an ERROR and will have unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0
RXFIFO_EMPTY	[5]	Interrupt is generated if the RxFIFO is empty 1 = Enables IRQ source 0 = Disables IRQ source	0
RXFIFO_ALMOST _EMPTY	[4]	Interrupt is generated if the RxFIFO is ALMOST empty. Almost empty is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
RX_FIFO_FULL	[3]	Interrupt is generated if the RxFIFO is full 1 = Enables IRQ source 0 = Disables IRQ source	0
RX_FIFO _ALMOST _FULL	[2]	Interrupt is generated if the RxFIFO is ALMOST full. Almost full is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
RXFIFO_ERROR _STARVE	[1]	Interrupt is generated for RxFIFO starve ERROR. This occurs if the RxFIFO is read when it is still empty. This is considered as an ERROR and has unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0

PCM_n_IRQ_CTL	Bit	Description	Reset Value
RXFIFO_ERROR _OVERFLOW	[0]	Interrupt is generated for RxFIFO overflow ERROR. This occurs if the RxFIFO is written when it is already full. This is considered as an ERROR and has unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0

6.6 PCM INTERRUPT STATUS REGISTER

- PCM_0_IRQ_STAT, R/O, Address = 0xF240_0014
- PCM_1_IRQ_STAT, R/O, Address = 0xF250_0014

The PCM_n_IRQ_STAT register is used to report IRQ status.

The bit definitions for the PCM_n_IRQ_STATUS Register are described below:

PCM_n_IRQ_STAT	Bit	Description	Reset Value
Reserved	[31:14]	Reserved	0
IRQ_PENDING	[13]	Controls whether or not the PCM interrupt is sent to the DSP 1 = PCM IRQ is forwarded to the DSP subsystem 0 = PCM IRQ is NOT forwarded to the DSP subsystem	0
TRANSFER _DONE	[12]	Interrupt is generated every time the serial shift for a word completes 1= Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_EMPTY	[11]	Interrupt is generated if the TxFIFO is empty 1= Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_ALMOST _EMPTY	[10]	Interrupt is generated if the TxFIFO is ALMOST empty. Almost empty is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_FULL	[9]	Interrupt is generated if the TxFIFO is full 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_ALMOST _FULL	[8]	Interrupt is generated whenever the TxFIFO is ALMOST full. Almost full is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_ERROR _STARVE	[7]	Interrupt is generated for TxFIFO starve ERROR. This occurs if the TxFIFO is read when it is still empty. This is considered an ERROR and has unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0
TXFIFO_ERROR _OVERFLOW	[6]	Interrupt is generated for TxFIFO overflow ERROR. This occurs if the TxFIFO is written when it is already full. This is considered an ERROR and has unexpected results 1 =Enables IRQ source 0 = Disables IRQ source	0

PCM_n_IRQ_STAT	Bit	Description	Reset Value
RXFIFO_EMPTY	[5]	Interrupt is generated if the RxFIFO is empty 1 = Enables IRQ source 0 = Disables IRQ source	0
RXFIFO_ALMOST_EMPTY	[4]	Interrupt is generated if the RxFIFO is ALMOST empty. Almost empty is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
RX_FIFO_FULL	[3]	Interrupt is generated if the RxFIFO is full 1 = Enables IRQ source 0 = Disables IRQ source	0
RX_FIFO_ALMOST_FULL	[2]	Interrupt is generated if the RxFIFO is ALMOST full. Almost full is defined as FIXME words remaining 1 = Enables IRQ source 0 = Disables IRQ source	0
RXFIFO_ERROR_STARVE	[1]	Interrupt is generated for RxFIFO starve ERROR. This occurs if the RxFIFO is read when it is still empty. This is considered an ERROR and has unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0
RXFIFO_ERROR_OVERFLOW	[0]	Interrupt is generated for RxFIFO overflow ERROR. This occurs if the RxFIFO is written when it is already full. This is considered an ERROR and has unexpected results 1 = Enables IRQ source 0 = Disables IRQ source	0

6.7 PCM FIFO STATUS REGISTER

- PCM_0_FIFO_STAT, R/O, Address = 0xF240_001C
- PCM_1_FIFO_STAT, R/O, Address = 0xF250_001C

The PCM_n_FIFO_STAT register is used to report FIFO status.

The bit definitions for the PCM_n_IRQ_STATUS Register are described below:

PCM_n_FIFO_STAT	Bit	Description	Reset Value
Reserved	[31:20]	Reserved	0
TXFIFO_COUNT	[19:14]	To indicate TXFIFO usage.	0
TXFIFO_EMPTY	[13]	To indicate whether TXFIFO is empty.	0
TXFIFO_ALMOST_EMPTY	[12]	To indicate whether TXFIFO is almost empty.	0
TXFIFO_FULL	[11]	To indicate whether TXFIFO is full.	0
TXFIFO_ALMOST_FULL	[10]	To indicate whether TXFIFO is almost full.	0
RXFIFO_COUNT	[9:4]	To indicate RXFIFO usage.	0
RXFIFO_EMPTY	[3]	To indicate whether RXFIFO is empty.	0
RXFIFO_ALMOST_EMPTY	[2]	To indicate whether RXFIFO is almost empty.	0
RX_FIFO_FULL	[1]	To indicate whether RXFIFO is full.	0
RX_FIFO_ALMOST_FULL	[0]	To indicate whether RXFIFO is almost full.	0

6.8 PCM INTERRUPT CLEAR REGISTER

- PCM_0_CLRINT, W/O, Address = 0xF240_0020
- PCM_1_CLRINT, W/O, Address = 0xF250_0020

The PCM_n_CLRINT register is used to clear the interrupt. Interrupt service routine is responsible for clearing interrupt asserted. Writing any values on this register clears interrupts for both ARM and DSP. Reading this register is not allowed. Clearing interrupt must be prior to resolving the interrupt condition, else another interrupt that would occur after this interrupt may be ignored.

PCM_n_CLRINT	Bit	Description	Reset Value
PCM_CLRINT	[31:0]	Interrupt is cleared if this bit is written to any value	

10.6 SPDIF TRANSMITTER

1 OVERVIEW

This standard describes a serial, un-directional, and self-clocking interface for the interconnection of digital audio equipment for consumer and professional applications. If this is used in a consumer digital processing environment, the interface is primarily intended to carry stereophonic programs, with a resolution of up to 20 bits per sample, an extension to 24-bit per sample being possible. If this is used in a broadcasting studio environment, the interface is primarily intended to carry monophonic or stereophonic programs, at a 48 kHz sampling frequency and with a resolution of up to 24-bit per sample; it may alternatively be used to carry one or two signals sampled at 32 kHz.

In both cases, the clock references and auxiliary information are transmitted along with the program. Provision is is made to allow the interface to carry data related to computer software.

2 FEATURE

- SPDIFOUT module only supports the consumer application in S5PC100X
- Supports Linear PCM up to 24-bit per sample
- Supports Non-linear PCM formats such as AC3, MPEG1 and MPEG2
- 2 x 24-bit buffers which is alternately filled with data

3 BLOCK DIAGRAM

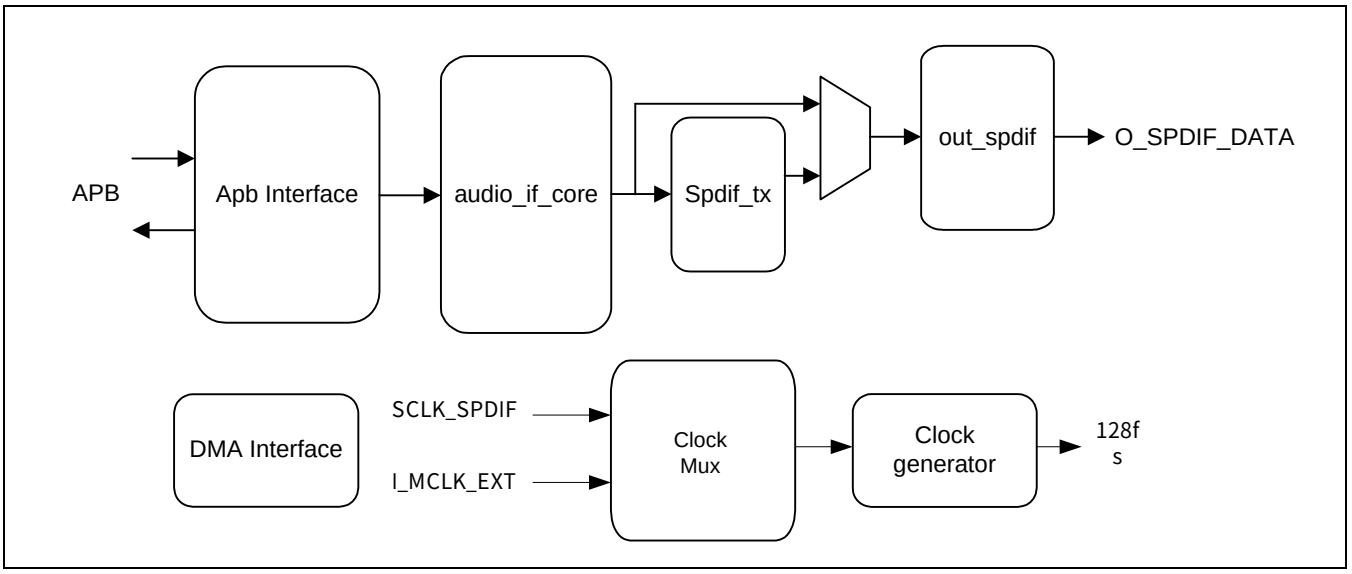


Figure 10.6-1 SPDIFOUT Block Diagram

- **APB Interface Block:** Defines register banks to control the driving of SPDIFOUT module and data buffers to store linear or non-linear PCM data.
- **DMA Interface Block:** Requests DMA service to IODMA depending on the status of data buffer in APB Interface block
- **Clock Generator Block:** Makes 128fs (sampling frequency) clock used in out_spdif block from system audio clock (MCLK)
- **Clock Multiplex Block:** System audio clock (MCLK) can be selected as internal MCLK or external MCLK.
- **Audio_if_core Block:** Acts as interface block between data buffer and out_spdif block. Finite-state machine controls the flow of PCM data.
- **spdif_tx Block:** Inserts burst preamble and executes zero-stuffing in the nonlinear PCM stream. Linear PCM data are bypassed by spdif_tx module.
- **out_spdif Block:** Makes SPDIF format. It inserts 4-bit preamble, 16- or 20- or 24-bit data, user-data bit, validity bit, channel status bit and parity bit into the appropriate position of 32-bit word. It modulates each bit to bi-phase format.

4 FUNCTIONAL DESCRIPTIONS

4.1 DATA FORMAT OF SPDIF

4.1.1 Frame Format

A frame is uniquely composed of two sub-frames. The transmission rate of frames corresponds exactly to the source sampling frequency.

In the 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames. Sub-frames related to channel 1(left or "A" channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once every 192 frame. This unit composed of 192 frames defines the block structure used to organize the channel status information. Sub-frames of channel 2 (right or "B" in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

In the single channel operation mode in broadcasting studio environment the frame format is identical to the 2-channel mode. Data is carried only in channel 1. In the sub-frames allocated to channel 2, time slot 28 (validity flag) is set to logical "1" (not valid).

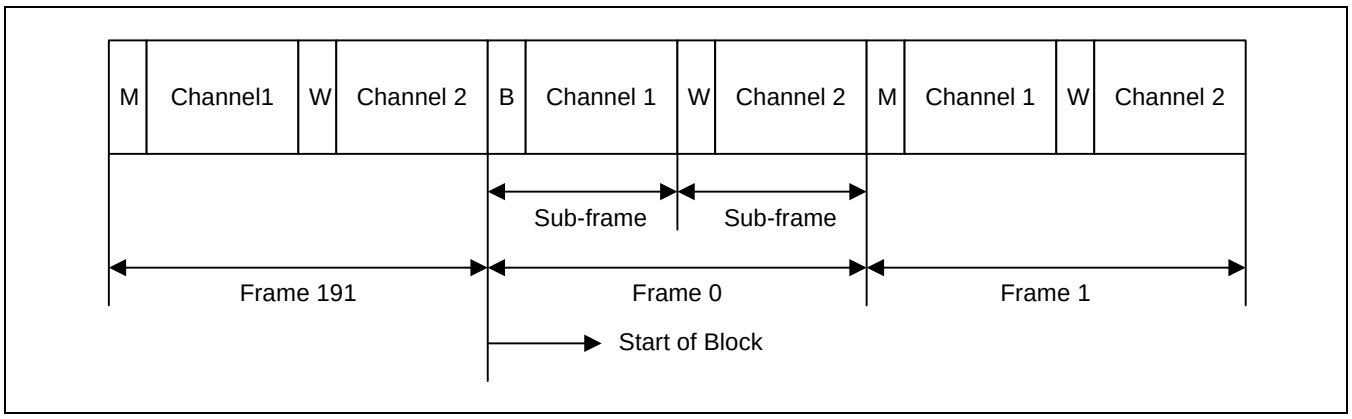


Figure 10.6-2 SPDIF Frame Format

4.1.2 Sub-frame Format (IEC 60958)

Each sub-frame is divided into 32 time slot, numbered from 0 to 31. Time slot 0 to 3 carries one of the three permitted preambles. These are used to affect synchronization of sub-frames, frames and blocks. Time slots 4 to 27 carry the audio sample word in linear 2's complement representation. The most significant bit is carried by time slot 27. If a 24-bit coding range is used, the least significant bit is in time slot 4. If a 20-bit coding range is sufficient, the least significant bit is in time slot 8 and time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (24 or 20), the unused least significant bits is set to a logical "0". By this procedure, equipment using different numbers of bits is connected. Time slot 28 carries the validity flag associated with the audio sample word. This flag is set to logical "0" if the audio sample is reliable. Time slot 29 carries one bit of the user data associated with the audio channel transmitted in the same sub-frame. The default value of the user bit is logical "0". Time slot 30 carries one bit of the channel status words associated with the audio channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that time slots 4 to 31 inclusive carries an even number of ones and an even number of zeros.

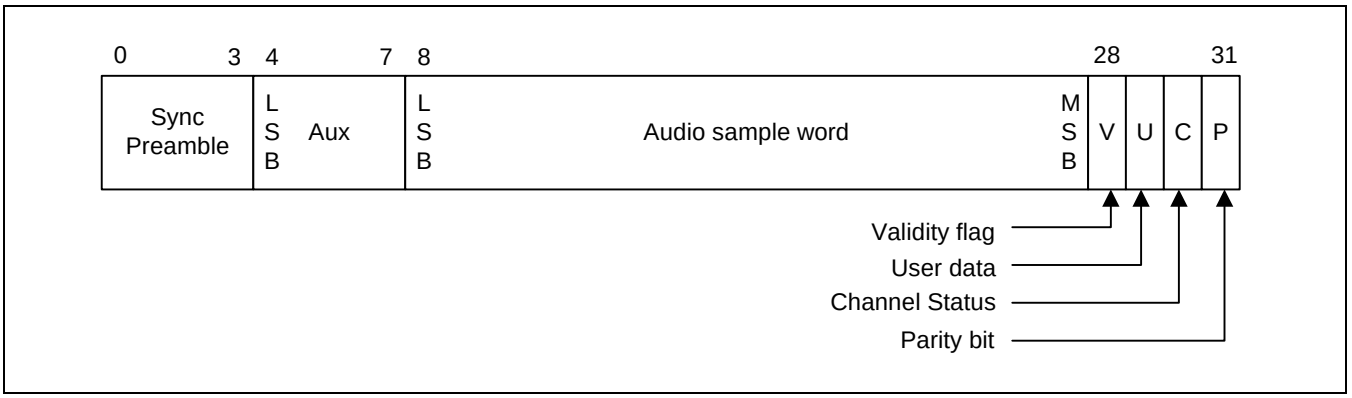


Figure 10.6-3 SPDIF Sub-frame Format

4.2 CHANNEL CODING

To minimize the dc component on the transmission line, to facilitate clock recovery from the data stream and to make the interface insensitive to the polarity of connections, time slots 4 to 31 are encoded in biphase-mark. Each bit to be transmitted is represented by a symbol comprising two consecutive binary states. The first state of a symbol is always different from the second state of the previous symbol. The second state of the symbol is identical to the first if the bit to be transmitted is logical "0", is different from the first if the bit is logical "1".

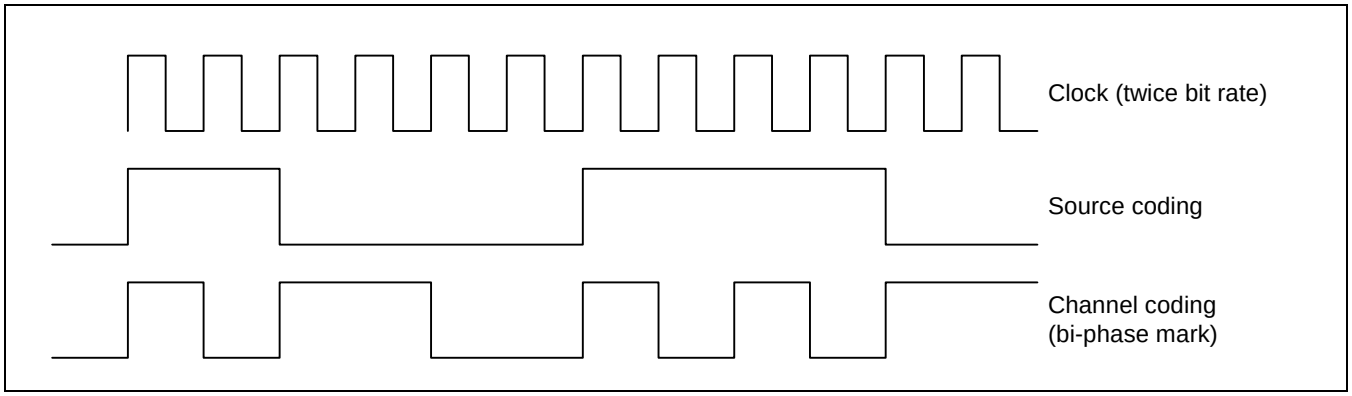


Figure 10.6-4 Channel Coding

4.3 PREAMBLE

Preambles are specific patterns providing synchronization and identification of the sub-frames and blocks. A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.
Like bi-phase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid bi-phase sequence.

4.4 NON-LINEAR PCM ENCODED SOURCE(IEC 61937)

The non-linear PCM encoded audio bitstream is transferred using the basic 16-bit data area of the IEC 60958 subframes, i.e. in time slots 12 to 27. Each IEC 60958 frame transfers 32-bit of the non-PCM data in consumer application mode.

If the SPDIF bitstream conveys linear PCM audio, the symbol frequency is 64 times the PCM sampling frequency (32 time slots per PCM sample times two channels). If a non-linear PCM encoded audio bitstream is conveyed by the interface, the symbol frequency is 64 times the sampling rate of the encoded audio within that bitstream. But in the case where a non-linear PCM encoded audio bitstream is conveyed by the interface containing audio with low sampling frequency, the symbol frequency is 128 times the sampling rate of the encoded audio within that bitstream.

Each data burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc, Pd), followed by the burst-payload which contains data of an encoded audio frame.
The burst-preamble consists of four mandatory fields. Pa and Pb represent a synchronization word; Pc gives information about the type of data and some information/control for the receiver; Pd gives the length of the burst-payload, limited to 216(=65,535) bits.
The four preamble words are contained in two sequential SPDIF frames. The frame beginning the data-burst contains preamble word Pa in subframe 1 and Pb in subframe 2. The next frame contains Pc in subframe 1 and Pd in subframe 2. When placed into a SPDIF subframe, the MSB of a 16-bit burst-preamble is placed into time slot 27 and the LSB is placed into time slot 12.

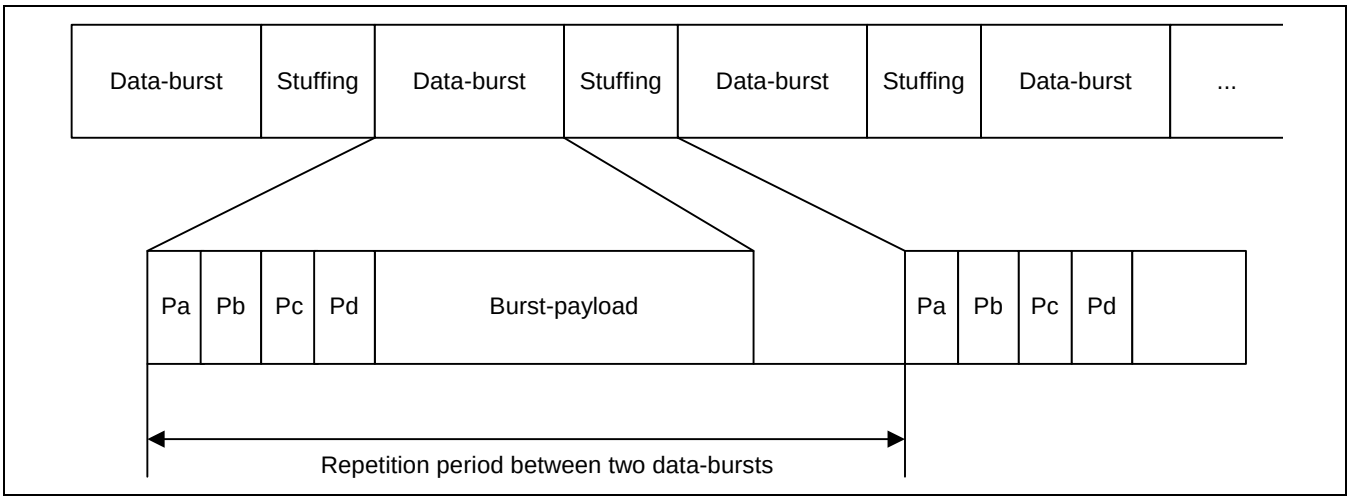


Figure 10.6-5 Format of Burst Payload

Table 10.6-1 Burst Preamble Words

Preamble word	Length of field	Contents	Value MSB.. LSB
Pa	16 bits	Sync word 1	0xF872
Pb	16 bits	Sync word 2	0x4E1F
Pc	16 bits	Burst-info	Refer to SPDBSTAS_SHD[15:0]
Pd	16 bits	Length-code	Refer to SPDBSTAS_SHD[31:16]

4.5 SPDIF OPERATION

Since the bit frequency of SPDIF is 128fs (fs: sampling frequency), the main clock of SPDIF is made by dividing audio main clock (MCLK) depending on the frequency of MCLK. MCLK is divided by 2 in case of 256fs, by 3 in case of 384fs and by 4 in case of 512fs.

SPDIF module in S5PC100X plays the role of transforming audio sample data into the format of SPDIF. To do this, SPDIF module inserts preamble data, channel status data, user data, error check bit and parity bit into the appropriate time slots. Preamble data are fixed in the module and inserted depending on subframe counter. Channel status data are set in the SPDCSTAS register and used by one bit per frame. User data always have zero values.

For non-linear PCM data, burst-preamble which consists of Pa, Pb, Pc and Pd must be inserted before burst-payload and zero is stuffed from the end of burst-payload to the repetition count. Pa(=16'hF872) and Pb(=16'h4E1F) is fixed in the module and Pc and Pd is set in the register SPDBSTAS. To stuff zero, the end of burst-payload is calculated from Pd value and repetition count which depends on data type in the preamble Pc is acquired from register SPDCNT.

Audio data are justified to the LSB. 16-, 20- or 24-bit PCM data and 16-bit stream data are supported. The unoccupied upper bits of 32-bit word are ignored.

Data are fetched through DMA request. If one of two data buffers is empty, DMA service is requested. Audio data stored in the data buffers are transformed into SPDIF format and output to the port. For non-linear PCM data, interrupt is generated after audio data are output up to the value specified in the SPDCNT register. Interrupt makes the registers such as SPDBSTAS and SPDCNT be set to new values if data type of new bitstream is different from the previous one.

4.6 SHADOWED REGISTER

Both SPDBSTAS_SHD register and SPDCNT_SHD register are shadowed registers which are related to SPDBSTAS register and SPDCNT register, respectively. They are updated from related registers every stream end interrupt signal. Their usage is as follows.

1. Sets burst status and repetition count information to their respective registers.
2. Turns on SPDIF module, and stream end interrupt is asserted immediately.
3. With stream end interrupt, shadowed registers are updated from their related registers and SPDIF data starts to go out. Now next stream information (burst status and repetition count) is written to SPDBSTAS and SPDCNT register because previous information is copied to their respective shadowed registers.
4. Set next stream information to SPDBSTAS and SPDCNT register.
5. Wait for stream end interrupt which signals the end of the first stream.
6. With stream end interrupt, the 2nd stream data starts to go out. Sets 3rd stream information to registers.

The usage of user bit registers is similar to that of stream information registers except that they are not related to SPDIF end interrupt but to user data interrupt. As soon as SPDIF is on, shadowed user bit registers are updated from their related registers and user bit starts to be shifted out with user data interrupt asserted. User can write the next user data to registers with this interrupt. After all of the 96 user bits are shifted out, user data interrupt asserts again and 3rd user bits is written to registers with 2nd user bits going out.

5 I/O DESCRIPTION

SPDIF external pads are shared with I2S and SDMMC. In order to use these pads for SPDIF, GPIO must be set before the SPDIF started. For mode information, refer to the Chapter GPIO.

Funtion Signal	I/O	Description	Pad	Type
SPDIF_EXTCLK	Input	Global Audio Main Clock (External MCLK)	Xi2c1SCL or Xmmc2CDn	muxed
SPDIF_OUT	Output	SPDIFOUT Data Output	Xi2c1SDA or Xmmc2DATA[3]	muxed

6 REGISTER DESCRIPTION

6.1 REGISTER OVERVIEW

Name	Address	R/W	Description	Reset Value
SPDCLKCON	0xF260_0000	R/W	Clock Control Register	0x0000_0002
SPDCON	0xF260_0004	R/W	Control Register	0x0000_0000
SPDBSTAS	0xF260_0008	R/W	Burst Status Register	0x0000_0000
SPDCSTAS	0xF260_000C	R/W	Channel Status register	0x0000_0000
SPDDAT	0xF260_0010	W	SPDIFOUT Data Buffer	0x0000_0000
SPDCNT	0XF260_0014	R/W	Repetition Count Register	0x0000_0000
SPDBSTAS_SHD	0XF260_0018	R	Shadowed Burst Status Register	0x0000_0000
SPDCNT_SHD	0XF260_001C	R	Shadowed Repetition Count Register	0x0000_0000
USERBIT1	0XF260_0020	R/W	Subcode Q1 ~ Q32	0x0000_0000
USERBIT2	0XF260_0024	R/W	Subcode Q33 ~ Q64	0x0000_0000
USERBIT3	0XF260_0028	R/W	Subcode Q65 ~ Q96	0x0000_0000
USERBIT1_SHD	0XF260_002C	R	Shadowed Register Userbit1	0x0000_0000
USERBIT2_SHD	0XF260_0030	R	Shadowed Register Userbit2	0x0000_0000
USERBIT3_SHD	0XF260_0034	R	Shadowed Register Userbit3	0x0000_0000
VERSION_INFO	0XF260_0038	R	RTL Version Information	0x0000_000B

6.2 DETAILED DESCRIPTION

6.2.1 Clock Control Register (SPDCLKCON, R/W, Address = 0XF260_0000)

SPDCLKCON	Bit	Description	Reset Value
Reserved	[31:3]	Reserved	0
Main Audio Clock Selection	[2]	0 = Internal clock (SCLK_SPDIF) 1 = External clock (I_MCLK_EXT)	0
SPDIFOUT clock down ready (read only)	[1]	0 = Clock-down not ready 1 = Clock-down ready	1
SPDIFOUT power on	[0]	0 = Power Off 1 = Power On	0

6.2.2 SPDIFOUT Control Register (SPDCON, R/W, Address = 0XF260_0004)

SPDCON	Bit	Description	Reset Value
Reserved	[31:26]	Reserved	0
FIFO Level	[25:22]	FIFO Level Monitoring (Read Only) FIFO depth is 8 *0: Empty of FIFO Level, 8: Full of FIFO Level	0000
FIFO Level Threshold	[21:19]	FIFO Threshold Level is controllable 000 = 0-FIFO Level 001 = 1-FIFO Level 010 = 2-FIFO Level 011 = 3-FIFO Level 100 = 4-FIFO Level 101 = 5-FIFO Level 110 = 6-FIFO Level 111 = 7-FIFO Level	000
FIFO transfer mode	[18:17]	00 = DMA transfer mode 01 = Polling mode 10 = Interrupt mode 11 = Reserved	00
FIFO_level Interrupt Status	[16]	Read Operation 0 = No interrupt pending. 1 = Interrupt pending. Write Operation 0 = No effect. 1 = Clear this flag.	0
FIFO_level Interrupt Enable	[15]	0 = Interrupt masked 1 = Interrupt enable	0

SPDCON	Bit	Description	Reset Value
endian format	[14:13]	00 = big endian o_data = {in_data[23:0]} 01 = 4 byte swap o_data={in_data[15:8], in_data[23:16], in_data[31:24]} 10 = 3 byte swap o_data={in_data[7:0], in_data[15:8], in_data[23:16]} 11 = 2 byte swap o_data={0x00,in_data[7:0], in_data[15:8]} *in_data: BUS → in port of SPDIF o_data: in port of SPDIF → Logic	0
user_data_attach	[12]	0 = User data is stored in USERBIT register. User data of subframe is out from USERBIT1,2,3 (96-bit) 1 = User data is stored in 23rd bit of audio data. User data is out in PCM data's 23th bit.	0
User Data Interrupt Status	[11]	Read Operation 0 = No interrupt pending. 1= Interrupt pending when 96bit of user data is out. Write Operation 0 = No effect. 1 = Clear this flag.	0
User Data Interrupt Enable	[10]	0 = Interrupt masked 1 = Interrupt enable	0
Buffer Empty Interrupt Status	[9]	Read Operation 0 = No interrupt pending. 1 = Interrupt pending. Write Operation 0 = No effect. 1= Clear this flag.	0
Buffer Empty Interrupt Enable	[8]	0 = Interrupt masked 1 = Interrupt enable	0
Stream End Interrupt Status	[7]	Read Operation 0 = No interrupt pending. 1 = Interrupt pending if the number of output audio data reaches repetition count in SPDCNT register. Write Operation 0 = No effect. 1= Clear this flag.	0
Stream End Interrupt Enable	[6]	0 = Interrupt masked 1 = Interrupt enable	0

SPDCON	Bit	Description	Reset Value
Software Reset	[5]	0 = Normal operation 1 = Software reset Software reset is 1-cycle pulse (auto clear) I_MCLK should be enabled before software reset assertion because SPDIF uses synchronous reset	0
Main Audio Clock Frequency	[4:3]	00 = 256fs 01 = 384fs 10 = 512fs 11 = reserved	0
PCM Data Size	[2:1]	00 = 16-bit 01 = 20-bit 10 = 24-bit 11 = reserved	0
PCM or Stream	[0]	0 = stream 1 = PCM	0

6.2.3 SPDIFOUT Burst Status Register (SPDBSTAS, R/W, Address = 0XF260_0008)

SPDBSTAS	Bit	Description	Reset Value
Burst data length bit	[31:16]	ES size in bits (Burst Preamble Pd) ES size: Elementary Stream size This indicates Burst-payload length	0
Bitstream number	[15:13]	Bit_stream_number, shall be set to 0	0
Data type dependent info	[12:8]	Data type dependent information	0
Error flag	[7]	0 = Error flag indicating a valid burst_payload 1 = Error flag indicating that the burst payload may contain errors	0
Reserved	[6:5]	Reserved	0
Compressed data type	[4:0]	00000 = Null Data 00001 = AC-3 00010 = Reserved 00011 = Pause 00100 = MPEG1 (layer1) 00101 = MPEG1 (layer2, 3), MPEG2-bc 00110 = MPEG2 – extension 00111 = Reserved 01000 = MPEG2 (layer1 – lsf) 01001 = MPEG2 (layer2, layer3 – lsf) others = Reserved	0

6.2.4 SPDIFOUT Channel Status Register (SPDCSTAS, R/W, Address = 0XF260_000C)

SPDCSTAS	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	0
Clock accuracy	[29:28]	10 = Level I, ±50 ppm 00 = Level II, ±1000 ppm 01 = Level III, variable pitch shifted	0
Sampling frequency	[27:24]	0000 = 44.1 kHz 0010 = 48 kHz 0011 = 32 kHz 1010 = 96 kHz	0
Channel number	[23:20]	Bit 20 is LSB	0
Source number	[19:16]	Bit 16 is LSB	0
Category code	[15:8]	Equipment type CD player = 0000_0001 DAT player = L000_0011 DCC player = L100_0011 Mini disc = L100_1001 (L: information about generation status of the material)	0
Channel status mode	[7:6]	00 = Mode 0 Others = Reserved	0
Emphasis	[5:3]	If bit1 = 0, 000 = 2 audio channels without pre-emphasis 001 = 2 audio channels with 50us / 15us pre-emphasis If bit1 = 1, 000 = Default State	0
Copyright assertion	[2]	0 = Copyright 1 = No copyright	0
Audio sample word	[1]	0 = Linear PCM 1 = Non-linear PCM	0
Channel status block	[0]	0 = Consumer format 1 = Professional format	0

6.2.5 SPDIFOUT Data Buffer (SPDDAT, W, Address = 0XF260_0010)

SPDDAT	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
SPDIFOUT data	[23:0]	PCM or stream data	0

6.2.6 SPDIFOUT Repetition Count Register (SPDCNT, R/W, Address = 0XF260_0014)

SPDCNT	Bit	Description	Reset Value
Reserved	[31:13]	Reserved	0
Stream repetition count	[12:0]	Repetition count according to data type. This bit is valid for stream data.	0

6.2.7 Shadowed SPDIF Burst Status Register (SPDBSTAS_SHD, R, Address = 0XF260_0018)

SPDBSTAS_SHD	Bit	Description	Reset Value
Burst Data Length Bit	[31:16]	ES size in bits (Burst Preamble Pd) ES size : Elementary Stream size This indicates Burst-payload length	0
Bitstream number	[15:13]	Bit_stream_number, shall be set to 0	0
Data Type Dependent Info	[12:8]	Data type dependent information	0
Error Flag	[7]	0 = Error flag indicating a valid burst_payload 1 = Error flag indicating that the burst payload may contain errors	0
Reserved	[6:5]	Reserved	0
Compressed Data Type	[4:0]	00000 = Null Data 00001 = AC-3 00010 = Reserved 00011 = Pause 00100 = MPEG1 (layer1) 00101 = MPEG1 (layer2, 3), MPEG2-bc 00110 = MPEG2 – extension 00111 = Reserved 01000 = MPEG2 (layer1 – lsf) 01001 = MPEG2 (layer2, layer3 – lsf) Others = Reserved	0

6.2.8 Shadowed SPDIF Repetition Count Register (SPDCNT_SHD, R, Address = 0XF260_001C)

SPDCNT_SHD	Bit	Description	Reset Value
Reserved	[31:13]	Reserved	0
Stream Repetition Count	[12:0]	Repetition count according to data type This bit is valid for stream data.	0

6.2.9 User Data Register

- Subcode Q1 ~ Q32 (USERBIT1, R/W, Address = 0XF260_0020)
- Subcode Q33 ~ Q64 (USERBIT2, R/W, Address = 0XF260_0024)
- Subcode Q65 ~ Q96 (USERBIT3, R/W, Address = 0XF260_0028)

USERBIT1~3	Bit	Description	Reset Value
User Data Bit (subcode Q for CD)	[31:0]	USERBIT1 : Q1 ~ Q32 USERBIT2 : Q33 ~ Q64 USERBIT3 : Q65 ~ Q96 User Data Bit has the Digital Audio Track information like (Track NO, Play Time etc.). 1176 bits of these being taken out in a row.	0

6.2.10 Shadowed User Data Register

- Shadowed Register Userbit1 (USERBIT1_SHD, R, Address = 0XF260_002C)
- Shadowed Register Userbit2 (USERBIT2_SHD, R, Address = 0XF260_0030)
- Shadowed Register Userbit3 (USERBIT3_SHD, R, Address = 0XF260_0034)

USERBIT_SHD	Bit	Description	Reset Value
User Data Bit	[31:0]	USERBIT1_SHD : Q1 ~ Q32 USERBIT2_SHD : Q33 ~ Q64 USERBIT3_SHD : Q65 ~ Q96 User Data Bit has the Digital Audio Track information like (Track NO, Play Time etc.). 1176 bits of these being taken out in a row.	0

6.2.11 Version Information Register (VERSION_INFO, R, Address = 0XF260_0038)

VERSION_INFO	Bit	Description	Reset Value
Version Information	[31:0]	RTL Version Information	0x0000_000B

NOTES

10.7 ADC & TOUCH SCREEN INTERFACE

This chapter describes the functions and usage of ADC & Touch Screen interface in S5PC100 RISC microprocessor.

1 OVERVIEW

The 10-bit or 12-bit CMOS Analog to Digital Converter (ADC) is a recycling type device with 10-channel analog inputs. It converts the analog input signal into 10-bit or 12-bit binary digital codes at a maximum conversion rate of 1MSPS with 5MHz A/D converter clock. A/D converter operates with on-chip sample-and-hold function. The power down mode is supported.

Touch Screen Interface controls input pads (XP, XM, YP, and YM) to obtain X/Y-position on the external touch screen device. Touch Screen Interface contains three main block, these are touch screen pads control logic, ADC interface logic and interrupt generation logic.

2 FEATURES

The ADC & Touch Screen interface includes the following features:

- Resolution: 10-bit / 12-bit (optional)
- Differential Linearity Error: ± 1.0 LSB
- Integral Linearity Error: ± 2.0 LSB
- Maximum Conversion Rate: 1 MSPS
- Low Power Consumption
- Power Supply Voltage: 3.3V
- Analog Input Range: 0 ~ 3.3V
- On-chip sample-and-hold function
- Normal Conversion Mode
- Separate X/Y Position Conversion Mode
- Auto (Sequential) X/Y Position Conversion Mode
- Waiting for Interrupt Mode
- STOP Mode Wakeup Source

3 ADC AND TOUCH SCREEN INTERFACE OPERATION

3.1 BLOCK DIAGRAM

Figure 10.7-1 is the functional block diagram of A/D converter and Touch Screen Interface.

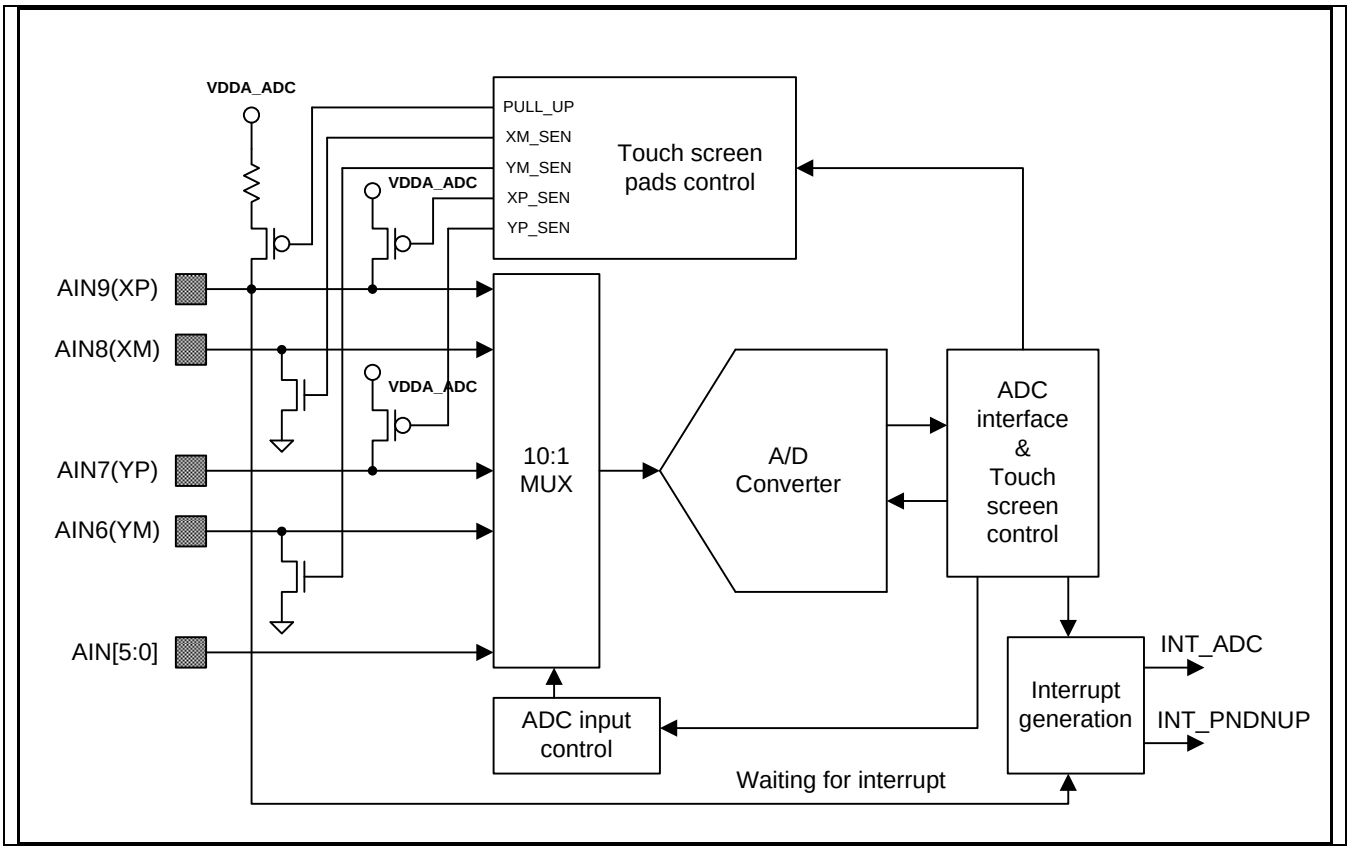


Figure 10.7-1 ADC and Touch Screen Interface Functional Block Diagram

NOTE

If Touch Screen device is used, XM or YM is only connected to ground for Touch Screen I/F. If Touch Screen device is not used, XM or YM connects Analog Input Signal for Normal ADC conversion.

NOTE

XP, XM, YP, YM pin state is Hi-Z in Sleep mode

4 FUNCTION DESCRIPTIONS

A/D Conversion Time

If the PCLK frequency is 66MHz and the prescaler value is 65, total 10-bit or 12-bit conversion time is as follows.

A/D converter freq. = 66MHz/(65+1) = 1MHz

Conversion time = 1/(1MHz / 5cycles) = 1/200kHz = 5 us

NOTE: This A/D converter was designed to operate at maximum 5MHz clock, therefore the conversion rate can go up to 1MSPS.

Touch Screen Interface Mode

1. Normal Conversion Mode (AUTO_PST = 0, XY_PST = 0)

The operation of this mode is identical with AIN0~AIN5's. It can be initialized by setting the ADCCON (ADC Control Register) and ADCTSC (ADC touch screen control register). All of the switches and pull-up resister should be turned off (reset value 0x58 makes switches turn-off). The converted data can be read out from ADCDAT0 (ADC conversion data 0 register).

2. Separate X/Y position conversion Mode (AUTO_PST = 0, XY_PST:control)

This mode consists of two states; one is X-position measurement state and the other is Y-position measurement state.
X-position measurement state is operated as the following way; set XY_PST is '2b'01' and read out the converted data (X-position) from ADCDAT0. When XY_PST is '1', XP and XM switch is automatically on and ADC channel selection bits are automatically changed to '5'. The end of X-position conversion can be notified by interrupt (INT_ADC).
Y-position measurement state is operated as the following way; set XY_PST is '2' and read out the converted data (Y-position) from ADCDAT1. When XY_PST is '2', YP and YM switch is automatically on and ADC channel selection bits are automatically changed to '7'. The end of Y-position conversion can be notified by interrupt (INT_ADC).

State	XP	XM	YP	YM
X-position measurement	VDDA_ADC	VSSA_ADC	AIN7	Hi-z
Y-position measurement	AIN9	Hi-z	VDDA_ADC	VSSA_ADC

3. Auto (Sequential) X/Y Position Conversion Mode (AUTO_PST = 1, XY_PST = 0)

Auto (Sequential) X/Y Position Conversion Mode is operated in the following method: Touch screen controller sequentially converts X-Position and Y-Position that is touched. After Touch screen controller writes X-measurement data to ADCDAT0 and writes Y-measurement data to ADCDAT1, Touch screen interface generates Interrupt (INT_ADC). The measurement states are automatically changed. When X-Position is detected, XP and XM switch is automatically on and ADC channel selection bits are automatically changed to '5'. And then when X-Position is detected, YP and YM switch is automatically on and ADC channel selection bits are automatically changed to '7'. After Auto X/Y position conversion, mode is changed for pull-up interrupt detection (ADCTSC = 0x173)

.

4. Waiting for Interrupt Mode (ADCTSC = 0xd3)

Touch screen controller generates an interrupt signal (INT_PNDNUP) when the stylus pen is down or up. The value of ADCTSC(ADC touch screen control register) is '0xd3', PULL_UP is '0', XP_SEN is '1', XM_SEN is '0', YP_SEN is '1' and YM_SEN is '1'.
After touch screen controller generates interrupt signal (INT_PNDNUP), waiting for interrupt Mode must be cleared. (XY_PST sets to the No operation Mode)

Mode	XP	XM	YP	YM
Waiting for Interrupt Mode	VDDA_ADC(Pull-up enable)	Hi-z	Hi-z	VSSA_ADC

5. Standby Mode

Set ADCCON [2] to '1' to activate Standby mode. In this mode, A/D conversion operation halts and ADCDAT0, ADCDAT1 register contains the previous converted data.

5 PROGRAMMING NOTES

The A/D converted data can be accessed by means of interrupt or polling method. With interrupt method, the overall conversion time - from A/D converter start to convert data read - may be delayed because of the return time of interrupt service routine and data access time. With polling method, by checking the ADCCON [15] – end of conversion flag – bit, the read time from ADCDAT register can be determined.

A/D conversion can be activated in different way. After ADCCON [1] - A/D conversion start-by-read mode-is set to 1. A/D conversion starts simultaneously when converted data is read.

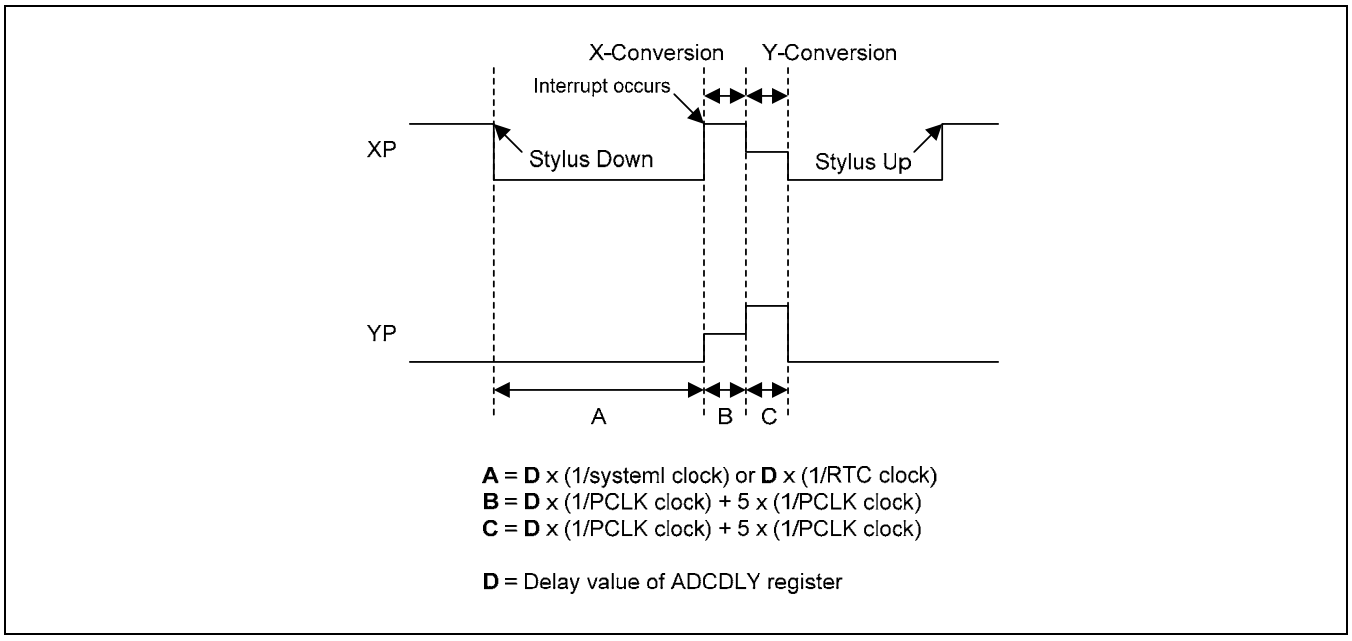


Figure 10.7-2 ADC and Touch Screen Operation Signal

If pen down/up interrupt is used as an wakeup source in IDLE and STOP mode, XY_PST bit (ADCTSC[1:0]) should be set to waiting for interrupt mode (2b'11). Use UD_SEN bit (ADCTSC[8]) to choose stylus pen up or pen down wakeup.

6 ADC & TOUCH SCREEN INTERFACE INPUT CLOCK DIAGRAM

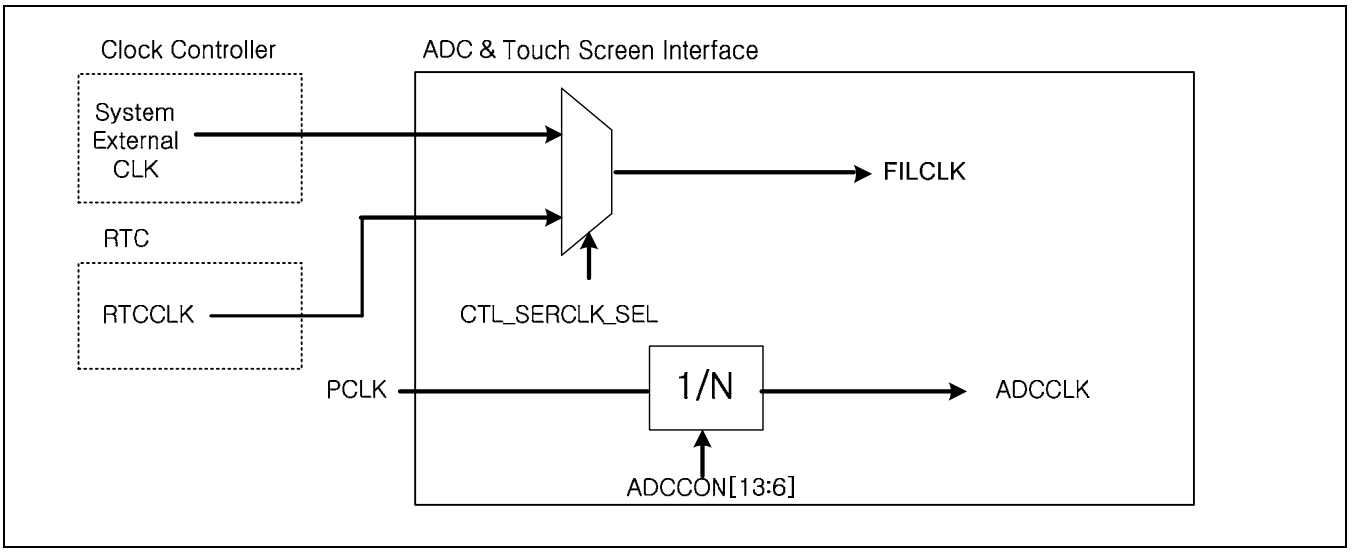


Figure 10.5-3 Input Clock Diagram for ADC & Touch Screen Interface

7 I/O DESCRIPTIONS

Function Signal	I/O	Descrioption	Pad	Type
AIN[9]	Input	ADC Channel[9] Analog Input	XadcAIN[9]	Analog
AIN[8]	Input	ADC Channel[8] Analog Input	XadcAIN[8]	Analog
AIN[7]	Input	ADC Channel[7] Analog Input	XadcAIN[7]	Analog
AIN[6]	Input	ADC Channel[6] Analog Input	XadcAIN[6]	Analog
AIN[5]	Input	ADC Channel[5] Analog Input	XadcAIN[5]	Analog
AIN[4]	Input	ADC Channel[4] Analog Input	XadcAIN[4]	Analog
AIN[3]	Input	ADC Channel[3] Analog Input	XadcAIN[3]	Analog
AIN[2]	Input	ADC Channel[2] Analog Input	XadcAIN[2]	Analog
AIN[1]	Input	ADC Channel[1] Analog Input	XadcAIN[1]	Analog
AIN[0]	Input	ADC Channel[0] Analog Input	XadcAIN[0]	Analog
ADC_VREF	Input	Vref is used to set conversion top range. Vref is connect to VDD_ADC	XadcVref	Analog

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

8 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
ADCCON	0xF300_0000	R/W	ADC Control Register	0x0000_3FC4
ADCTSC	0xF300_0004	R/W	ADC Touch Screen Control Register	0x0000_0058
ADCDLY	0xF300_0008	R/W	ADC Start or Interval Delay Register	0x0000_00FF
ADCDAT0	0xF300_000C	R	ADC Conversion Data Register	-
ADCDAT1	0xF300_0010	R	ADC Conversion Data Register	-
ADCUPDN	0xF300_0014	R/W	Stylus Up or Down Interrupt Register	0x0000_0000
ADCCLRINT	0xF300_0018	W	Clear ADC Interrupt	-
ADCMUX	0xF300_001C	R/W	Analog input channel selection	0x0000_0000
ADCCLRINTPNDNUP	0xF300_0020	W	Clear Pen Down/Up Interrupt	-

8.1 ADC CONTROL REGISTER (ADCCON, R/W, ADDRESS = 0XF300_0000)

ADCCON	Bit	Description	Reset Value
RES	[16]	ADC output resolution selection 0 = 10-bit A/D conversion 1 = 12-bit A/D conversion	0
ECFLG	[15]	End of conversion flag(Read only) 0 = A/D conversion in process 1 = End of A/D conversion	0
PRSCEN	[14]	A/D converter prescaler enable 0 = Disables 1 = Enables	0
PRSCVL	[13:6]	A/D converter prescaler value Data value: 5 ~ 255 Note that division factor is (N+1) if the prescaler value is N. **NOTE:** ADC frequency should be set less than PCLK by 5 times. (Ex. If PCLK=10MHz, ADC Frequency<2MHz)This A/D converter is designed to operate at maximum 5MHz clock	0xFF
Reserved	[5:3]	Reserved	0
STDBM	[2]	Standby mode select 0 = Normal operation mode 1 = Standby mode	1
READ_ START	[1]	A/D conversion start by read 0 = Disables start by read operation 1 = Enables start by read operation	0
ENABLE_START	[0]	A/D conversion starts by enable. If READ_START is enabled, this value is not valid. 0 = No operation 1 = A/D conversion starts and this bit is automatically cleared after the start-up.	0

8.2 ADC TOUCH SCREEN CONTROL REGISTER (ADCTSC, R/W, ADDRESS = 0XF300_0004)

ADCTSC	Bit	Description	Reset Value
UD_SEN	[8]	Detects Stylus Up or Down status. 0 = Detects Stylus Down Interrupt Signal. 1 = Detects Stylus Up Interrupt Signal.	0
YM_SEN	[7]	YM to GND Switch Enable 0 = Disables Switch (YM = AIN6, Hi-z) 1 = Enables Switch (YM = VSSA_ADC)	0
YP_SEN	[6]	YP to VDD Switch Enable 0 = Enables Switch (YP = VDDA_ADC) 1 = Disables Switch (YP = AIN7, Hi-z)	1
XM_SEN	[5]	XM to GND Switch Enable 0 = Disables Switch (XM = AIN8, Hi-z) 1 = Enables Switch (XM = VSSA_ADC)	0
XP_SEN	[4]	XP to VDD Switch Enable 0 = Enables Switch (XP = VDDA_ADC) 1 = Disables Switch (XP = AIN9, Hi-z)	1
PULL_UP	[3]	Pull-up Switch Enable 0 = Enables XP Pull-up 1 = Disables XP Pull-up	1
AUTO_PST	[2]	Automatic sequencing conversion of X-Position and Y-Position 0 = Normal ADC conversion. 1 = Auto Sequential measurement of X-position and Y-position.	0
XY_PST	[1:0]	Manually measurement of X-Position or Y-Position. 00 = No operation mode 01 = X-position measurement 10 = Y-position measurement 11 = Waiting for Interrupt Mode	0

NOTES:

1. While waiting for Touch screen Interrupt, XP_SEN bit must be set to '1', namely 'XP Output disable' and PULL_UP bit must be set to '0', namely 'XP Pull-up enable'.
2. AUTO_PST bit should be set '1' only in Automatic and Sequential X/Y Position conversion.
3. If you do not use AIN[7], you must tie AIN [7] to VDDA_ADC or ADCTSC register must be set to 0xd3.

8.3 TOUCH SCREEN PIN CONDITIONS IN X/Y POSITION CONVERSION.

	XP	XM	YP	YM	ADC ch. select
X Position	Vref	GND	AIN[7]	Hi-Z	YP
Y Position	AIN[9]	Hi-Z	Vref	GND	XP

8.4 ADC START DELAY REGISTER (ADCDLY, R/W, ADDRESS = 0XF300_0008)

ADCDLY	Bit	Description	Reset Value
FILCLKsrc	[16]	ADCDLY clock source. 0 = system clock. 1 = RTC clock.	0
DELAY	[15:0]	ADC delay counter value. **Note: Do not use Zero value(0x0000)**	00ff

NOTES: 1) In case of ADC conversion mode (Normal, Separate, Auto conversion):

ADC conversion is delayed by counting the value of DELAY bits. Counting clock is PCLK.
→ ADC conversion delay value.

2) In case of waiting for Interrupt mode:

If stylus down occurs in waiting for interrupt mode, Pen Up/Down Interrupt (INT_PNDNUP) occurs after counting the value of DELAY bits . Counting clock can be system clock or RTC clock by the value of . FLKCLKsrc bit
→ Interrupt generation delay value.

3) In case of IDLE or STOP mode :

If stylus down occurs in IDLE or STOP mode, Wake-Up signal is generated after counting the value of DELAY bits. Counting clock can be system clock or RTC clock by the value of FLKCLKsrc bit .

→ Wake-up Interrupt generation delay value.

Before entering STOP mode, set FLKCLKsrc bit to only 1'b1 to use RTC clock

8.5 ADC CONVERSION DATA REGISTER (ADCDAT0, R, ADDRESS = 0XF300_000C)

ADCDAT0	Bit	Description	Reset Value
UPDOWN	[15]	Up or Down state of Stylus at Waiting for Interrupt Mode. 0 = Stylus down state. 1 = Stylus up state.	-
AUTO_PST	[14]	Automatic sequencing conversion of X-Position and Y-Position 0 = Normal ADC conversion. 1 = Sequencing measurement of X-position, Y-position.	-
XY_PST	[13:12]	Manual measurement of X-Position or Y-Position. 00 = No operation mode 01 = X-position measurement 10 = Y-position measurement 11 = Waiting for Interrupt Mode	-
XPDATA (Normal ADC)	[11:0]	X-Position Conversion data value (Includes Normal ADC Conversion data value) Data value : 0x0 ~ 0xFFF	-

8.6 ADC CONVERSION DATA REGISTER (ADCDAT1, R, ADDRESS = 0XF300_0010)

ADCDAT1	Bit	Description	Reset Value
UPDOWN	[15]	Up or Down state of Stylus at Waiting for Interrupt Mode. 0 = Stylus down state. 1 = No stylus down state.	-
AUTO_PST	[14]	Automatic sequencing conversion of X-Position and Y-Position 0 = Normal ADC conversion. 1 = Sequencing measurement of X-position, Y-position.	-
XY_PST	[13:12]	Manual measurement of X-Position or Y-Position. 00 = No operation mode 01 = X-position measurement 10 = Y-position measurement 11 = Waiting for Interrupt Mode	-
YPDATA	[11:0]	Y-Position Conversion data value Data value: 0x0 ~ 0xFFF	-

8.7 ADC TOUCH SCREEN UP-DOWN REGISTER (ADCUPDN, R/W, ADDRESS = 0XF300_0014)

ADCUPDN	Bit	Description	Reset Value
TSC_UP	[1]	Stylus Up Interrupt history (After check, this bit should be cleared manually). 0 = No stylus up state. 1 = Stylus up interrupt has occurred.	0
TSC_DN	[0]	Stylus Down Interrupt history. (After check, this bit should be cleared manually) 0 = No stylus down state. 1 = Stylus down interrupt has occurred.	0

8.8 ADC TOUCH SCREEN INTERRUPT CLEAR REGISTER (ADCCLRINT , W, ADDRESS = 0XF300_0018)

These registers are used to clear the interrupts. Interrupt service routine is responsible for clearing interrupts after the interrupt service is completed. Writing any values on this register clears the relevant interrupts asserted. If it is read, undefined value returns.

ADCCLRINT	Bit	Description	Reset Value
INTADCCLR	[0]	INT_ADC interrupt clear	-

8.9 ADC CHANNEL MUX REGISTER (ADCMUX, R/W, ADDRESS = 0XF300_001C)

ADCMUX	Bit	Description	Reset Value
SEL_MUX	[3:0]	Analog input channel select 0000 = AIN 0 0001 = AIN 1 0010 = AIN 2 0011 = AIN 3 0100 = AIN 4 0101 = AIN 5 0110 = YM 0111 = YP 1000 = XM 1001 = XP	0

NOTE: SEL_MUX is invalid, when TSADC is set as 1) Separate X/Y position conversion Mode or 2) Auto (Sequential) X/Y Position Conversion Mode

8.10 ADC TOUCH SCREEN PEN DOWN/UP INTERRUPT CLEAR REGISTER (ADCCLRINTPNDNUP, W, ADDRESS = 0XF300_0020)

ADCCLRINTPNDNUP	Bit	Description	Reset Value
INTPNDNUPCLR	[0]	INT_PNDNUP interrupt clear	-

NOTES

10.8 KEYPAD INTERFACE

1 OVERVIEW

The Key Pad Interface block in S5PC100 provides communication with external keypad devices. The ports multiplexed with GPIO ports provide up to 8 rows and 8 columns. The events of key press or key release are detected to the CPU by an interrupt. If interrupt occurs from row lines, the software must scan the column lines using the proper procedure to detect one or multiple key press or release.

It provides interrupt status register bits at the moment of key pressed or key released or both cases (If two interrupt conditions are enabled). To prevent the switching noises, internal debouncing filter is provided.

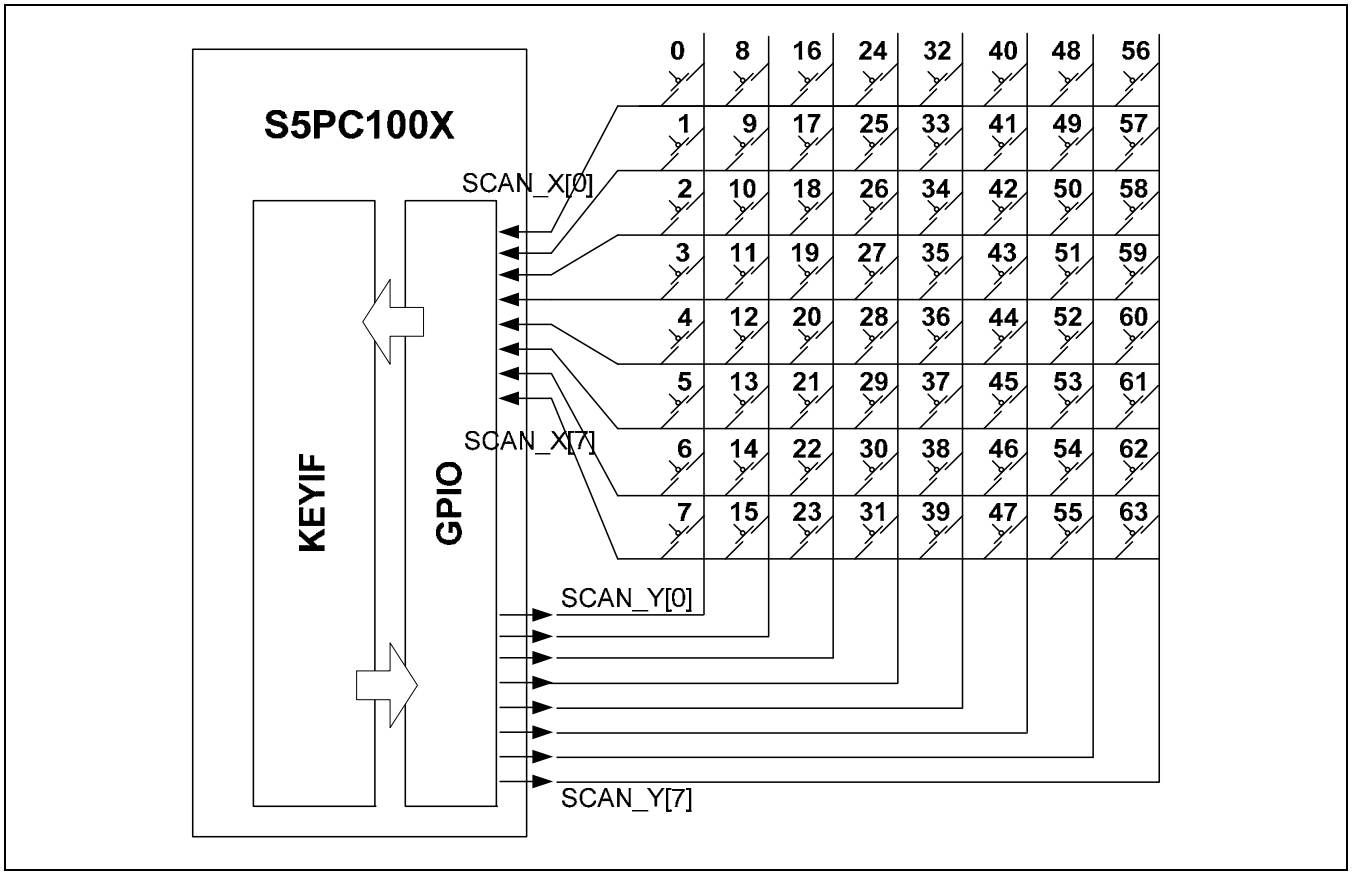


Figure 10.8-1 Key Matrix Interface External Connection Guide

2 DEBOUNCING FILTER

The debouncing filter is supported for keypad interrupt of any key input. The filtering width is approximately 62.5usec ("FCLK" two-clock, if the FCLK is 32kHz). The keypad interrupt (key pressed or key released) to the CPU is an ANDed signal of all row input lines after filtering.

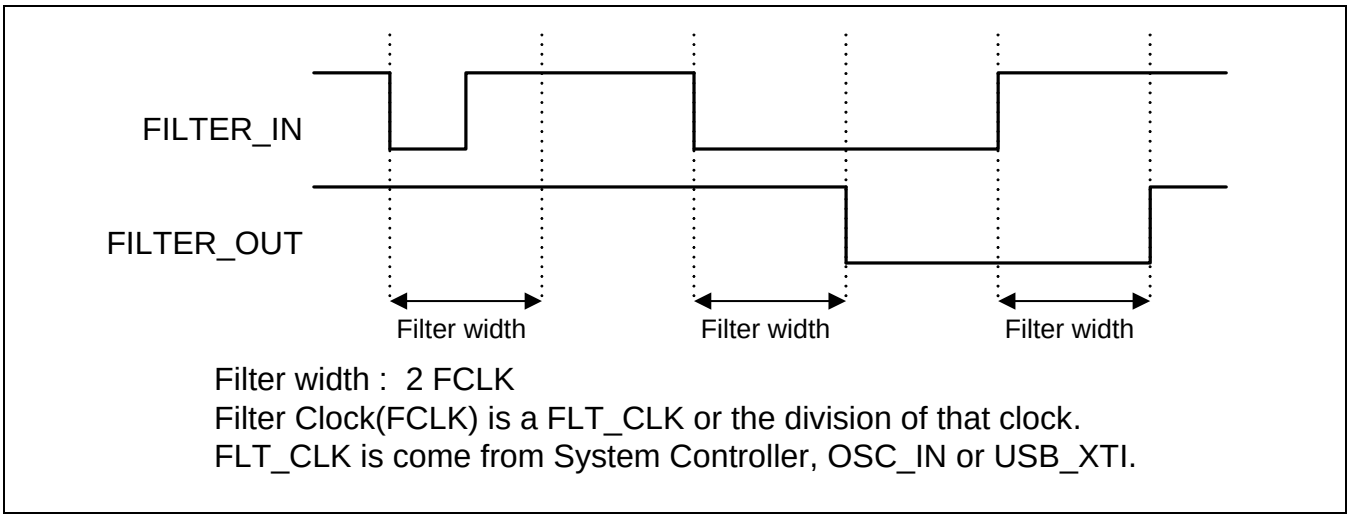


Figure 10.8-2 Internal Debouncing Filter Operation

3 FILTER CLOCK

KEYPAD interface debouncing filter clock (FCLK) is divided from FLT_CLK that is OSC_IN. You can set compare value for 10-bit up-counter (KEYIFFC). If filter enable bit (FC_EN) is HIGH, filter clock divider is ON. The frequency of FCLK is frequency of FLT_CLK / ((KEYIFFC + 1) x 2). On the contrary FC_EN is Low, filter clock divider does not divide FLT_CLK.

4 WAKEUP SOURCE

If the Key input is used to wakeup source from IDLE, STOP or SLEEP mode, KEYPAD I/F register setting is not required. GPIO register setting (GPH2CON, GPH3CON) for KEYPAD I/F and SYSCON register (PWR_CFG) to mask are required for wakeup.

5 KEYPAD SCANNING PROCEDURE

At initial state, all column lines (outputs) are low level. But column data output tri-state enable bits are all high, therefore, if the tri-state enable mode is not used, these bits should be written to zeros. If no key pressed state, all row lines (inputs) are high (used pull-up pads). If any key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row line, generating a keypad interrupt. The CPU (software) writes with a LOW level on one column line and HIGH on the others to the KEYIFCOL register. Each write time, the CPU reads the value of the KEYIFROW register and detects if one key of the corresponding column line is pressed. When the scanning procedure is end, the pressed key (one or more) can be detected.

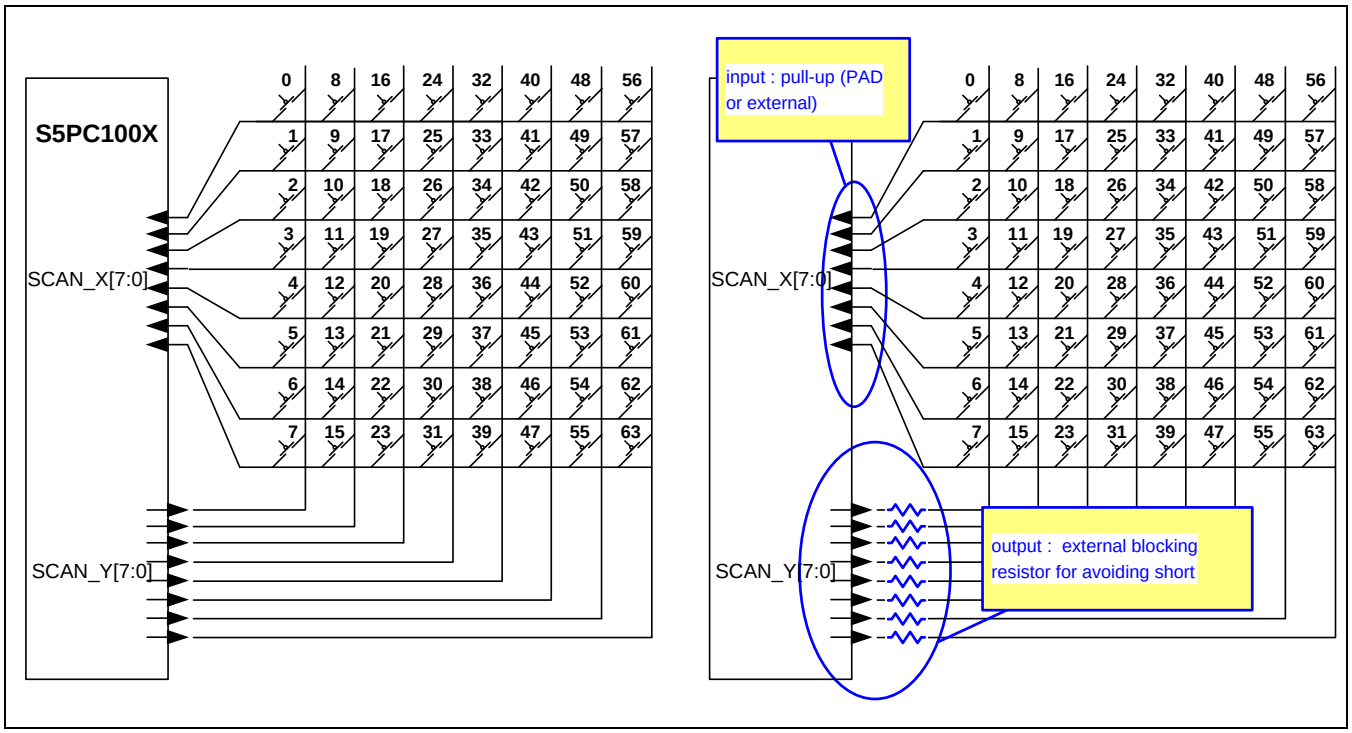


Figure 10.8-3 Keypad Scanning Procedure I

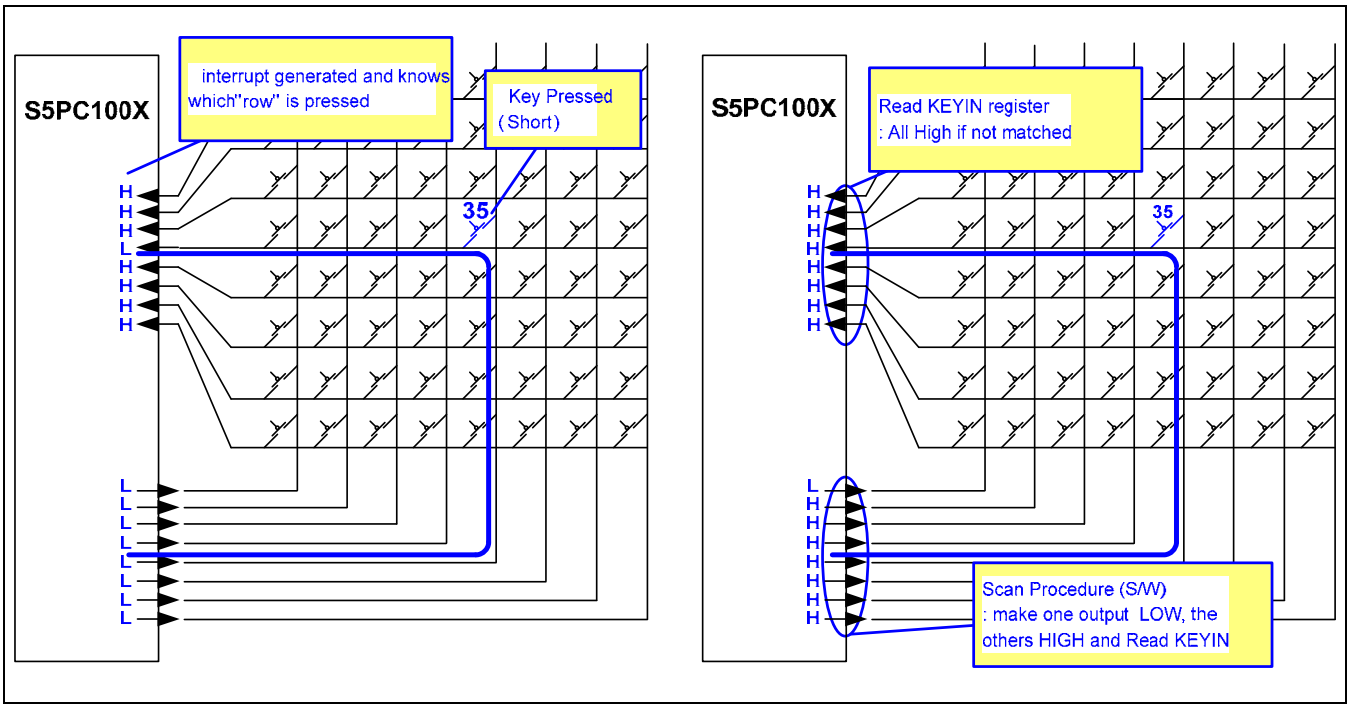


Figure 10.8-4 Keypad Scanning Procedure II

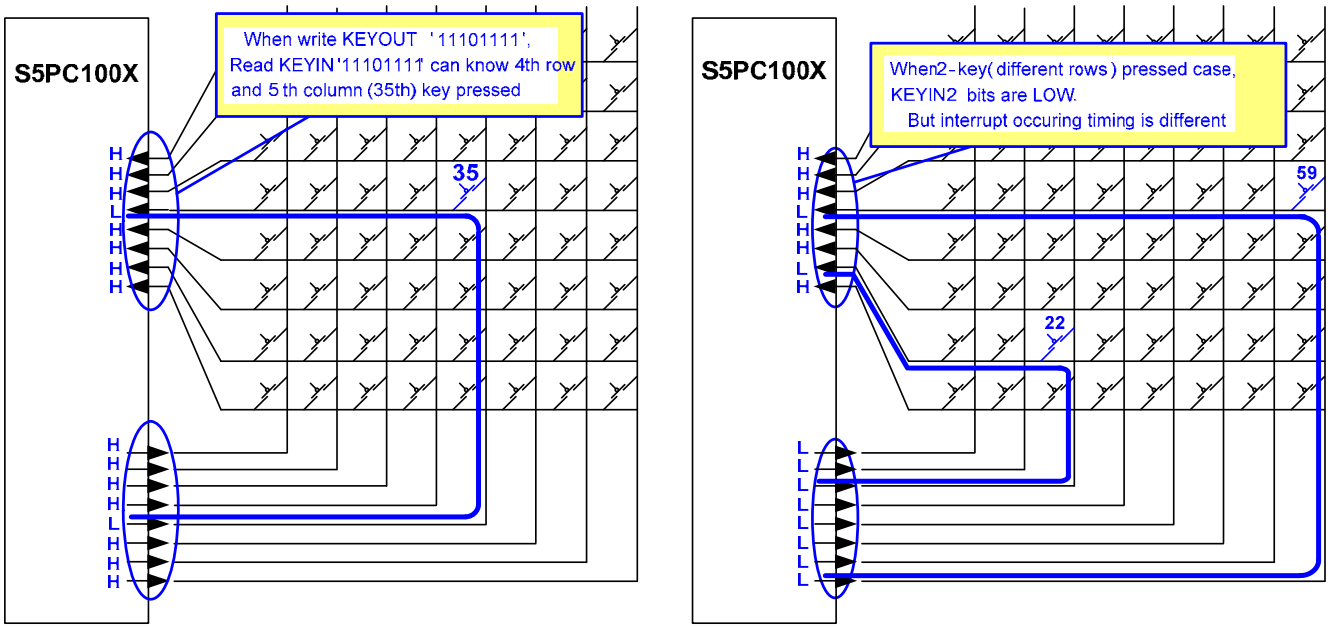


Figure 10.8-5 Keypad Scanning Procedure III

NOTE: Delay time is needed between setting a column data register and reading a row data register.

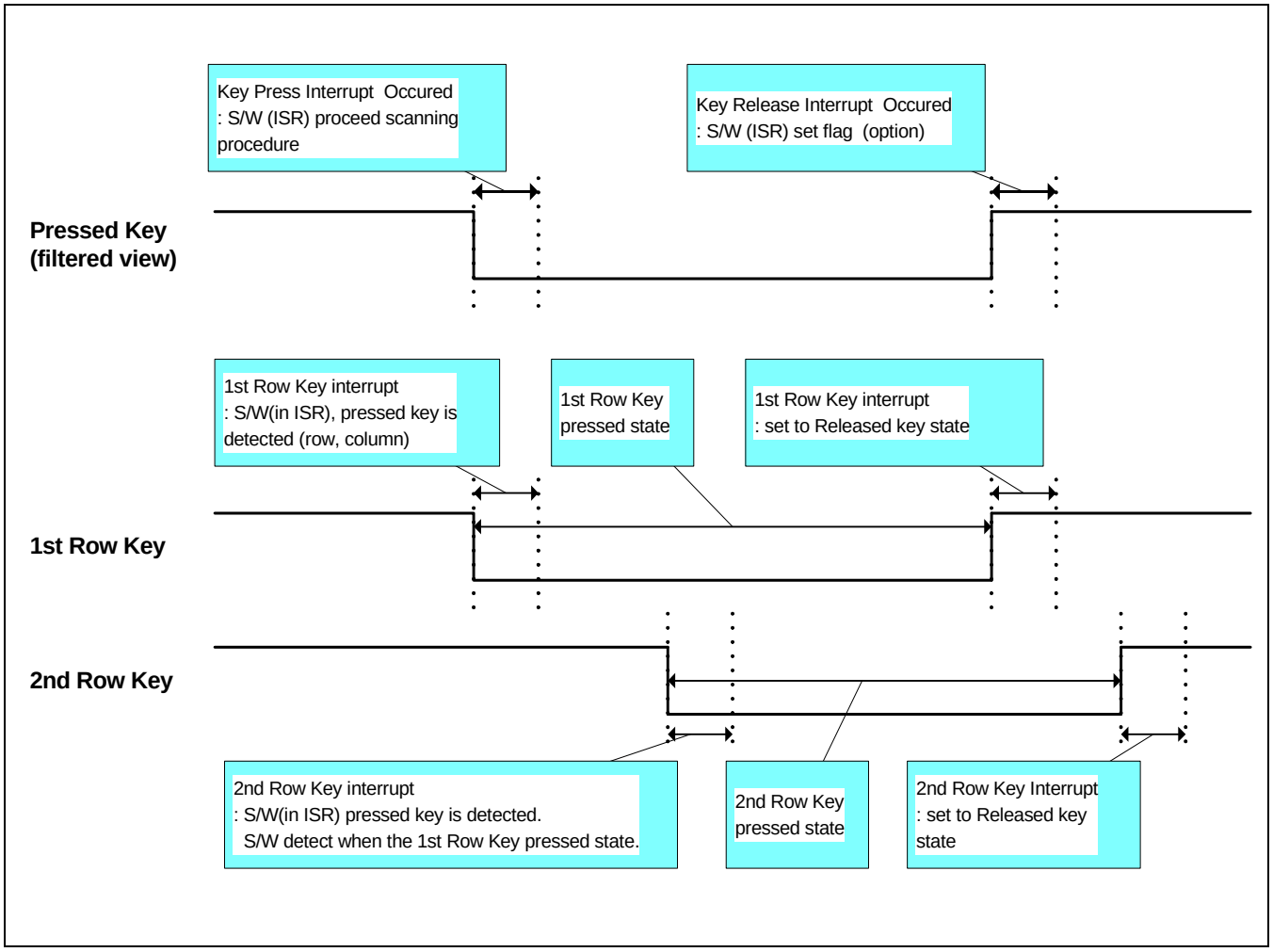


Figure 10.8-6 Keypad Scanning Procedure when the Two-Key Pressed with Different Row

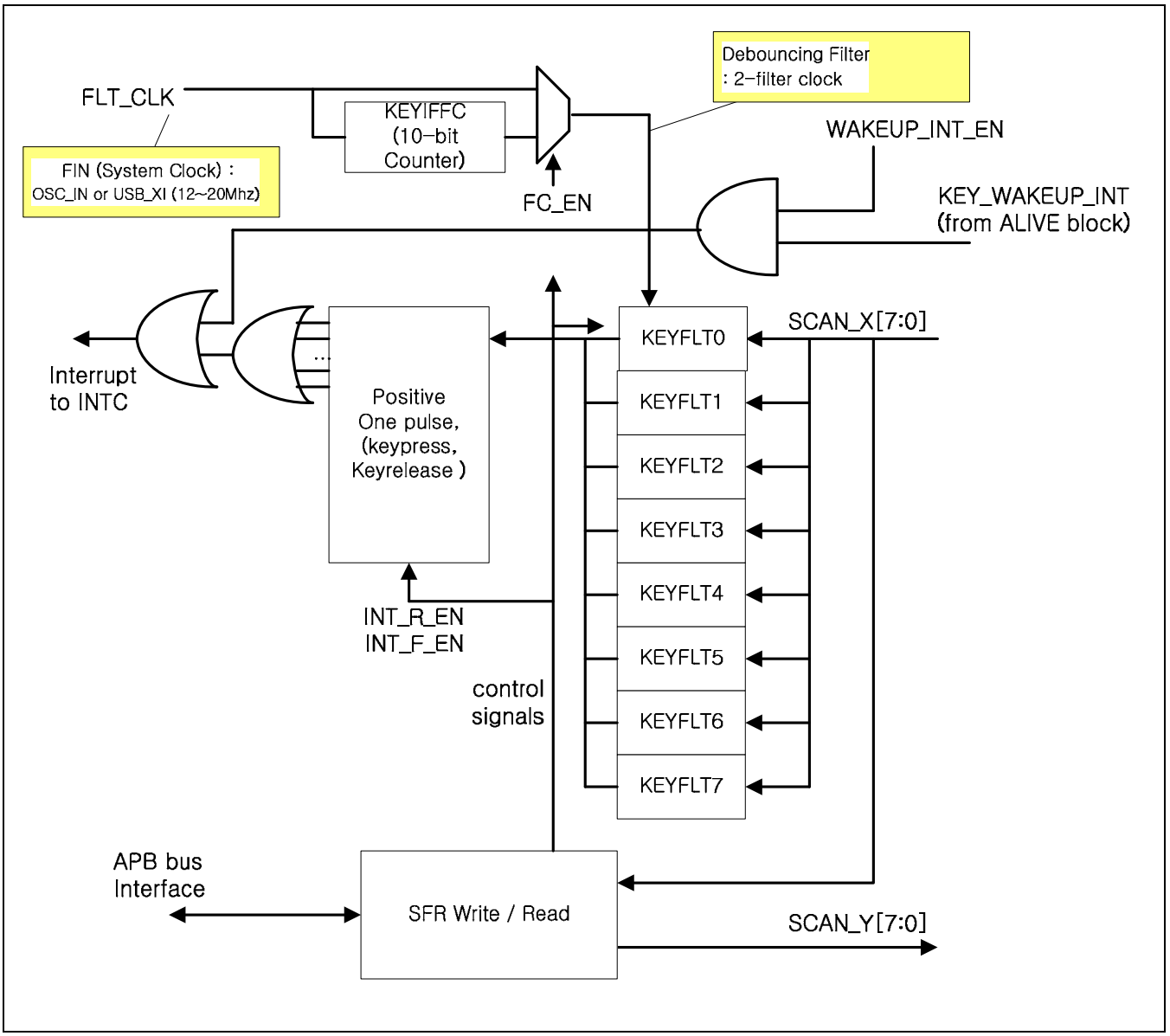


Figure 10.8-7 Keypad I/F Block Diagram

6 I/O DESCRIPTION

Function Signal	I/O	Descrioption	Pad	Type
KP_ROW[7]	Input	KEYPAD Interface Row[7] Data	XEINT[31]	Muxed
KP_ROW[6]	Input	KEYPAD Interface Row[6] Data	XEINT[30]	Muxed
KP_ROW[5]	Input	KEYPAD Interface Row[5] Data	XEINT[29]	Muxed
KP_ROW[4]	Input	KEYPAD Interface Row[4] Data	XEINT[28]	Muxed
KP_ROW[3]	Input	KEYPAD Interface Row[3] Data	XEINT[27]	Muxed
KP_ROW[2]	Input	KEYPAD Interface Row[2] Data	XEINT[26]	Muxed
KP_ROW[1]	Input	KEYPAD Interface Row[1] Data	XEINT[25]	Muxed
KP_ROW[0]	Input	KEYPAD Interface Row[0] Data	XEINT[24]	Muxed
KP_COL[7]	Output	KEYPAD Interface Column[7] Data	XEINT[23]	Muxed
KP_COL [6]	Output	KEYPAD Interface Column[6] Data	XEINT[22]	Muxed
KP_COL [5]	Output	KEYPAD Interface Column[5] Data	XEINT[21]	Muxed
KP_COL [4]	Output	KEYPAD Interface Column[4] Data	XEINT[20]	Muxed
KP_COL [3]	Output	KEYPAD Interface Column[3] Data	XEINT[19]	Muxed
KP_COL [2]	Output	KEYPAD Interface Column[2] Data	XEINT[18]	Muxed
KP_COL [1]	Output	KEYPAD Interface Column[1] Data	XEINT[17]	Muxed
KP_COL [0]	Output	KEYPAD Interface Column[0] Data	XEINT[16]	Muxed

NOTE: Type field indicates whether pads are dedicated to the signal or pads are connected to the multiplexed signals.

7 REGISTER DESCRIPTION

Register	Address	R/W	Description	Reset Value
KEYIFCON	0xF310_0000	R/W	KEYPAD Interface Control Register	0x00000000
KEYIFSTSCLR	0xF310_0004	R/W	KEYPAD Interface Status and Clear Register	0x00000000
KEYIFCOL	0xF310_0008	R/W	KEYPAD Interface Column Data Output Register	0x0000FF00
KEYIFROW	0xF310_000C	R	KEYPAD Interface Row Data Input Register	Reflects input ports
KEYIFFC	0xF310_0010	R/W	KEYPAD Interface Debouncing Filter Clock Division Register	0x00000000

7.1 KEYPAD INTERFACE CONTROL REGISTERS (KEYIFCON, R/W, ADDRESS = 0XF310_0000)

KEYIFCON	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0x0000000
WAKEUPEN	[4]	KEYPAD interrupt generation by KEYPAD Stop / Idle mode wakeup source '0' = Disables '1' = KEYPAD interrupt is generated by KEYPAD wakeup source	0
FC_EN	[3]	10-bit counter (for Debouncing digital filter clock) enable '0' = Disables: Division counter is not used • FCLK = FLT_CLK '1' = Enables: Use division counter • FCLK = FLT_CLK / ((KEYIFFC[9:0] + 1) x 2)	0
DF_EN	[2]	KEYPAD input port debouncing filter enable '0' = Disables, '1' = Enables	0
INT_R_EN	[1]	KEYPAD input port rising edge (key-released) interrupt '0' = Disables, '1' = Enables	0
INT_F_EN	[0]	KEYPAD input port falling edge (key-pressed) interrupt '0' = Disables, '1' = Enables	0

NOTE: Both edge interrupt is selected if both INT_F_EN and INT_R_EN are set.

7.2 KEYPAD INTERRUPT STATUS AND CLEAR REGISTER (KEYIFSTSCLR, R/W, ADDRESS = 0XF310_0004)

KEYIFSTSCLR	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	0x0000
R_INT	[15:8]	KEYPAD input "release" interrupt (rising edge) status(read) and clear(write) Read: '1' = Released interrupt occurs, '0' = Do not occur Write: Write '1' to clear Released interrupt. The R_INT[7:0] indicate that each key pressed from 0 to 7 has a dedicated interrupt to it from R_INT[8] to R_INT[15]	0x00
P_INT	[7:0]	KEYPAD input "press" interrupt (falling edge) status (read) and clear(write) Read: '1' = Pressed interrupt occurs, '0' = Do not occur Write: Write '1' to clear Pressed interrupt The P_INT[7:0] indicate that each key released from 0 to 7 has a dedicated interrupt to it from P_INT[0] to P_INT[7]	0x00

NOTE: Keypad wakeup interrupt is cleared if the write access to the KEYIFSTSCLR.

7.3 KEYPAD INTERFACE COLUMN DATA OUTPUT REGISTER (KEYIFCOL, R/W, ADDRESS = 0XF310_0008)

KEYIFCOL	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x0000FF
KEYIFCOL	[7:0]	KEYPAD interface column data output register	0x00

NOTE: To lock specific columns, set related bits in GPH2CON as GPI mode. Refer to GPIO chapter

7.4 KEYPAD INTERFACE ROW DATA INPUT REGISTER (KEYIFROW, R, ADDRESS = 0XF310_000C)

KEYIFROW	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0x000000
KEYIFROW	[7:0]	KEYPAD interface row data input register (read only) This register values from input ports are not filtered data.	Reflects input ports

7.5 KEYPAD INTERFACE DEBOUNCING FILTER CLOCK DIVISION REGISTER(KEYIFFC, R/W, ADDRESS = 0XF310_0010)

KEYIFFC	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0x000000
KEYIFFC	[9:0]	KEYPAD interface debouncing filter clock division register. User can set compare value for 10-bit up-counter. This register value means when FC_EN bit is HIGH. FCLK = FLT_CLK / ((KEYIFFC[9:0] + 1) x 2) (FLT_CLK is OSC_IN or USB_XI)	0x000

NOTES

11.1 SECURITY-SYSTEM

1 OVERVIEW

S5PC100 supports following security features

- Secure Booting
- Secure JTAG
- Secure Access Control
- Security Engines

Table 11.1-1 shows the detail security features of S5PC100.

For secure booting, on chip 32KB secure boot ROM, 96KB secure RAM and the 160-bit secure boot key hash value e-fuse ROM are provided. For more information refer to 'Chapter 02.06 Booting sequence'.

The multi-level HW authentication module for Secure JTAG is new addition to security subsystem. The 80-bit e-fuse ROM for secure JTAG key has value is also provided.

S5PC100 provides not only TrustZone HW access control as previsous Samsung AP such as S5PC64xx, it also provide Samsung's own secure access control solution which uses TrustZone HW. Samsung's own solution named Mobile Device Security Platform (MDSP) is composed of Secure Domain Manager (SDM) HW module and its secure SW solution. 128-bit e-fuse ROM is also provided as root key protected by secure access control.

It also includes Security engines such as DES/TDES, AES, SHA-1, PRNG and PKA.

Chapter 11.2 describes SDM module for secure access control. The Chapter 11.3, describes the Advanced Crypto Engine (ACE) composed of security engines and HW authentication module for secure JTAG. Tecurity keys for various security features are described in Chapter 11.4

Table 11.1-1 Security Features of S5PC100

	Description
Secure Booting	On chip 32KB secure boot ROM On chip 96KB secure SRAM 160-bit e-fuse ROM for secure boot key hash value
Secure JTAG	Samsung's own multi-level Hardware (H/W) authentication module 80-bit e-fuse ROM for secure JTAG key hash value
Secure Access Control	Using TrustZone H/W access control scheme Provide Mobile Device Security Platform (MDSP) : Samsung's own secure domain managing solution 128-bit e-fuse ROM for root key
Security Engines	DES/TDES, AES, SHA-1, PRNG and PKA

11.2 SECURE DOMAIN MANAGER

1 INTRODUCTION

1.1 DOMAIN ISOLATION CONCEPT

One key concept of the Security Module is that the services hosted in it are separated as much as possible from non-secure software. To achieve this goal, domain isolation should be implemented by hardware and software. Figure 11.2-1 shows the concept of domain isolation scheme in S5PC100. In previous CPU design, there are two level of protection mode such as privileged mode and user mode. For security protection, the concept of normal domain and the secure domain is also introduced as shown in Figure 11.2-1. Generally, all applications are running as user mode in normal domain. If one of the applications needs security function, it calls secure function in previliged mode. If that function is successfully authenticated, working domain is changed to secure domain and the secure function is progressed in secure domain. After secure function is complete, it is returned to user mode via previliged mode.

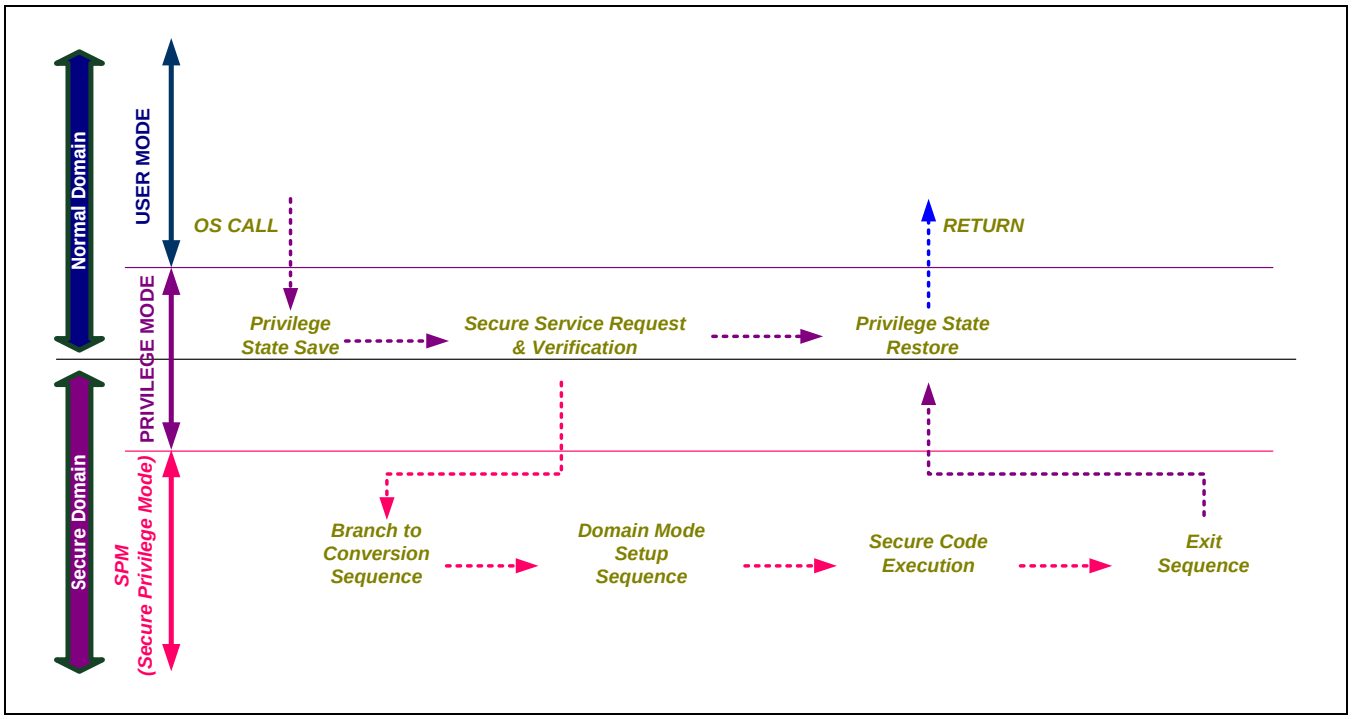


Figure 11.2-1 Concept Diagram of Domain Isolation

1.2 SDM DESIGN CONCEPT

The secure domain manager provides the switching function between non-secure domain and secure domain through the predefined secure domain manage program in order to ensure the trusted execution of secure program. If a request is made to secure program, the secure domain manager switchs from non-secure domain to secure domain by DMZ entrance sequences that sets up the secure program execution environment. After secure program is complete, it changes mode to non-secure domain by DMZ exit sequences that set up the normal program execution environment.

The purpose of switching over the special domain for secure program is to protect the secure resources of system from normal program that may have threat codes, by permitting only secure program to access secure resources.

- **Secure Domain:** State from secure boot state and DMZ entrance to end of the secure programs, and all slave & memory areas are accessible.
- **Non-Secure Domain:** All other state except the secure boot state, DMZ, and the secure domain state, and only the non-secure slave & memory areas are accessible.
- **DMZ:** All mode changes program sequence is progressing in DMZ state, and all slave & memory areas are accessible.
  - ♦ DMZ entrance state: non-secure to secure mode transition
  - ♦ DMZ exit state: secure to non-secure mode transition.

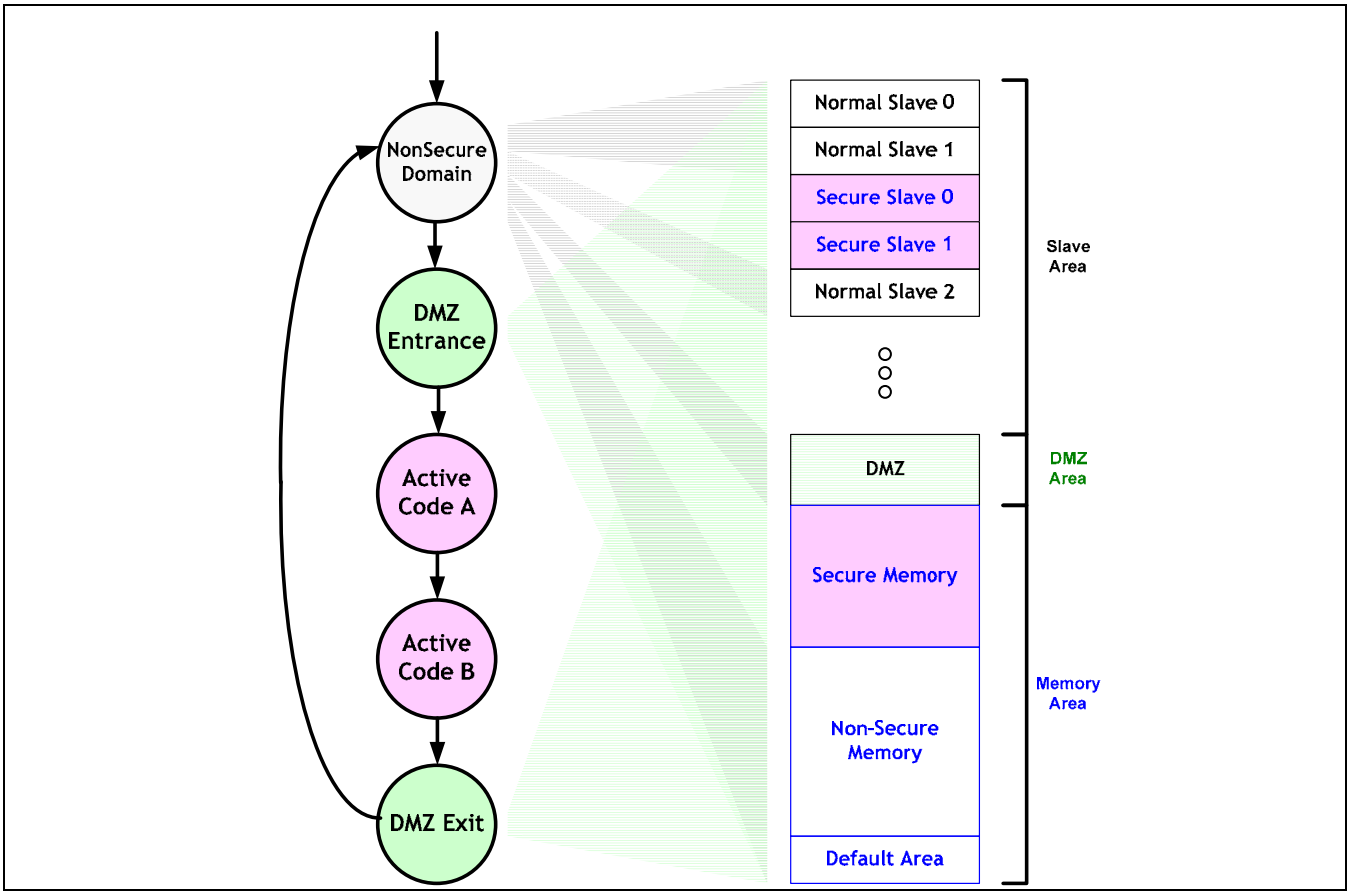


Figure 11.2-2 Rsource Access in Non-Secure/ Secure Domain

1.3 MAIN COMPONENTS

The Secure Domain Conversion (SDC) hardware is largely composed of the Secure Domain Manager (SDM) and the Trust Zone (TZ) hardware.

- **Secure Domain Manger:** Composed of the main FSM which generates the control signals to proper block, the bitmap based violation detector which checks the security rule violation, the LFSR which generates the Non-Cacheable Previliged Access Random Branch (NC-PA) offset, and the SFR bank for SDM.
- **DMZ ROM:** Composed of the region that provides the NC-PA random branch instruction and the hardwired ROM which generates DMZ entrance code and DMZ code. It is accessed like general ROM.
- **TZ Hardware:** Please refer to TZ documents from ARM Home page. (www.arm.com)

2 SDM FUNCTIONAL DESCRIPTION

SDM is the main block of domain isolation Hardware scheme in S5PC100. It provides the secure access control function by replacing CPU's AxPROT[1] with own access control signal. Therefore it does not need TZ Software. Figure 11.2-3 shows system level block diagram of SDM. SDM controls its own 'secure domain on' bit by observing cortexA8's bus signals.

DMZ is the special ROM area that connects from normal domain to secure domain, and it is the non-secure memory area. The pre-SW sequences that progresses the pre-processing to protect the secure resource are stored in DMZ ROM. The start address of block used in switching between non-secure and secure domain is stored in SFRs protected by special protection logic. Those SFRs are programmed only in the secure booting sequence. After secure booting is complete, their value cannot be altered until system reset occurs

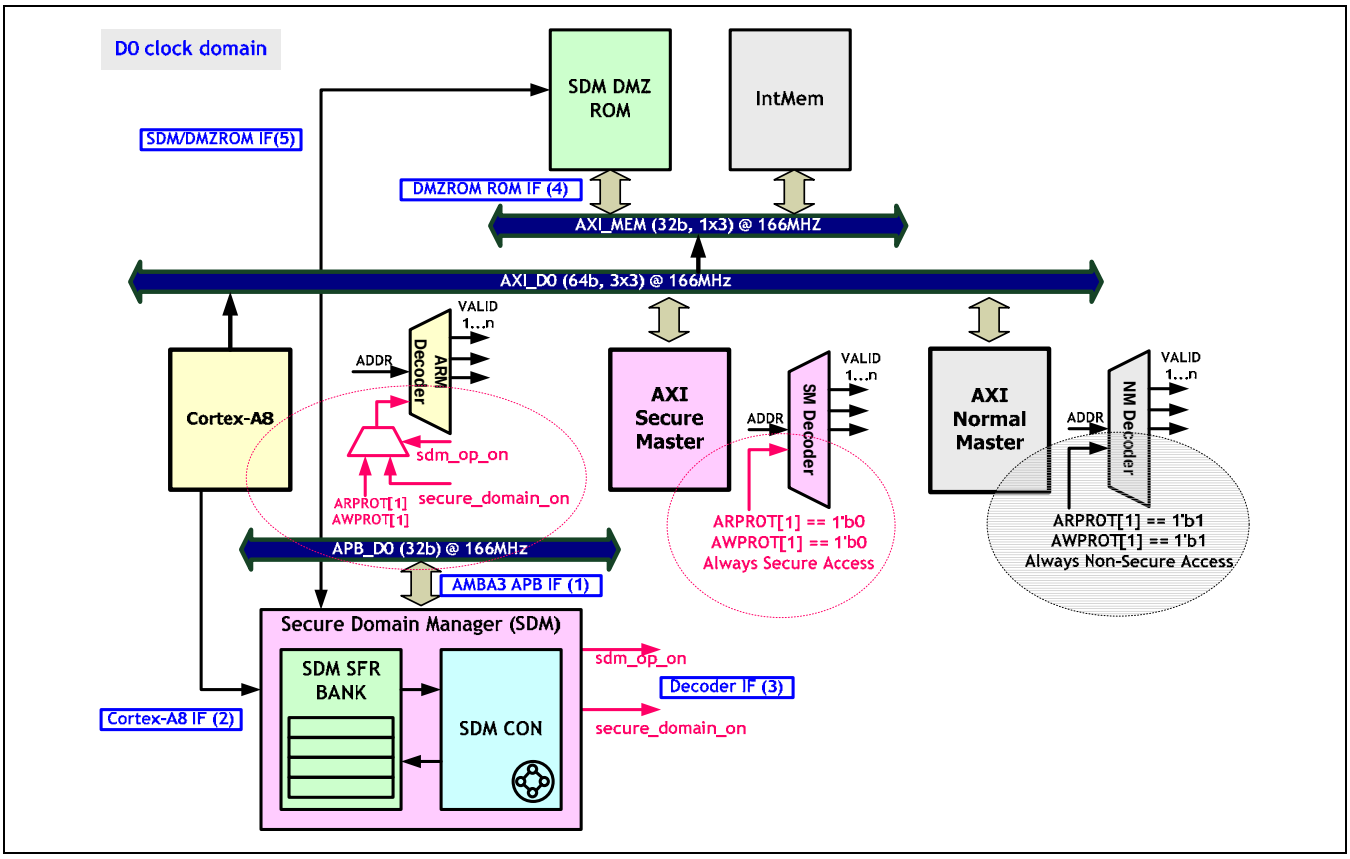


Figure 11.2-3 System Level Block Diagram of SDM

3 REGISTER DESCRIPTION

Register	Offset	R/W	Description	Reset Value
SDM_CON	0xF510_0000	R/W	SDM Control Register	0x0000_0000
SDM_SECURE_BOOT _STS	0xF510_0004	R/W	SDM Secure Boot Status Register	0x0000_0000
SDM_STS	0xF510_0008	R	SDM Status Register	0x0000_0080
SDM_FIQ2SBITON_DLY	0xF510_000C	R/W	SDM FIQ On to SBIT On Interval Delay Cycle Count Register	0x0000_0000
SDM_IMZOFF2DMZ_ DLY	0xF510_0010	R/W	SDM IMZ OFF to DMZ Entrance Interval Delay Cycle Count Register	0x0000_0000
SDM_RANDOM_BR_ADDR	0xF510_0014	R/W	SDM DMZ Entrance Mode 2 NC-PA Random Branch Start Address Register	0x0000_0000
SDM_RANDOM_BR_CNT	0xF510_0018	R/W	SDM DMZ Entrance Mode 2 NC-PA Random Branch Count Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
SDM_SEED_POLYNOMIAL	0xF510_001C	W	SDM DMZ Entrance Mode 2 NC-PA Random Branch LFSR Seed Polynomial Register	0x0000_0000
SDM_DMZ_ENTR_ADDR	0xF510_0020	R/W	SDM DMZ Entrance Address Register	0x0000_0000
SDM_DMZ_EXIT_ADDR	0xF510_0024	R/W	SDM DMZ Exit Address Register	0x0000_0000
SDM_SCODE_ENTR_ADDR	0xF510_0028	R/W	SDM Secure Code Entrance Address Register	0x0000_0000
SDM_SCODE_BASE_ADDR	0xF510_002C	R/W	SDM Secure Code Base Address Register	0x0000_0000
SDM_SCODE_BASE_MASK	0xF510_0030	R/W	SDM Secure Code Base Mask Value Register	0x0000_0000
SDM_BL_RETURN_INST	0xF510_0034	R/W	SDM DMZ Entrance Mode 2 NC-PA Random Branch Return Instruction Register	0x0000_0000
INITSCODE0_0	0xF510_0038	R/W	DMZ Configurable Initial Secure Code 0 Area 1st Instruction Register	0x0000_0000
INITSCODE0_1	0xF510_003C	R/W	DMZ Configurable Initial Secure Code 0 Area 2nd Instruction Register	0x0000_0000
INITSCODE0_2	0xF510_0040	R/W	DMZ Configurable Initial Secure Code 0 Area 3rd Instruction Register	0x0000_0000
INITSCODE0_3	0xF510_0044	R/W	DMZ Configurable Initial Secure Code 0 Area 4th Instruction Register	0x0000_0000
INITSCODE0_4	0xF510_0048	R/W	DMZ Configurable Initial Secure Code 0 Area 5th Instruction Register	0x0000_0000
INITSCODE0_5	0xF510_004C	R/W	DMZ Configurable Initial Secure Code 0 Area 6th Instruction Register	0x0000_0000
INITSCODE0_6	0xF510_0050	R/W	DMZ Configurable Initial Secure Code 0 Area 7th Instruction Register	0x0000_0000
INITSCODE0_7	0xF510_0054	R/W	DMZ Configurable Initial Secure Code 0 Area 8th Instruction Register	0x0000_0000
INITSCODE0_8	0xF510_0058	R/W	DMZ Configurable Initial Secure Code 0 Area 9th Instruction Register	0x0000_0000
INITSCODE0_9	0xF510_005C	R/W	DMZ Configurable Initial Secure Code 0 Area 10th Instruction Register	0x0000_0000
INITSCODE0_10	0xF510_0060	R/W	DMZ Configurable Initial Secure Code 0 Area 11th Instruction Register	0x0000_0000
INITSCODE0_11	0xF510_0064	R/W	DMZ Configurable Initial Secure Code 0 Area 12th Instruction Register	0x0000_0000
INITSCODE0_12	0xF510_0068	R/W	DMZ Configurable Initial Secure Code 0 Area 13th Instruction Register	0x0000_0000
INITSCODE0_13	0xF510_006C	R/W	DMZ Configurable Initial Secure Code 0 Area 14th Instruction Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
INITSCODE0_14	0xF510_0070	R/W	DMZ Configurable Initial Secure Code 0 Area 15th Instruction Register	0x0000_0000
INITSCODE0_15	0xF510_0074	R/W	DMZ Configurable Initial Secure Code 0 Area 16th Instruction Register	0x0000_0000
INITSCODE0_16	0xF510_0078	R/W	DMZ Configurable Initial Secure Code 0 Area 17th Instruction Register	0x0000_0000
INITSCODE0_17	0xF510_007C	R/W	DMZ Configurable Initial Secure Code 0 Area 18th Instruction Register	0x0000_0000
INITSCODE0_18	0xF510_0080	R/W	DMZ Configurable Initial Secure Code 0 Area 19th Instruction Register	0x0000_0000
INITSCODE0_19	0xF510_0084	R/W	DMZ Configurable Initial Secure Code 0 Area 20th Instruction Register	0x0000_0000
INITSCODE0_20	0xF510_0088	R/W	DMZ Configurable Initial Secure Code 0 Area 21th Instruction Register	0x0000_0000
INITSCODE0_21	0xF510_008C	R/W	DMZ Configurable Initial Secure Code 0 Area 22th Instruction Register	0x0000_0000
INITSCODE0_22	0xF510_0090	R/W	DMZ Configurable Initial Secure Code 0 Area 23th Instruction Register	0x0000_0000
INITSCODE0_23	0xF510_0094	R/W	DMZ Configurable Initial Secure Code 0 Area 24th Instruction Register	0x0000_0000
INITSCODE0_24	0xF510_0098	R/W	DMZ Configurable Initial Secure Code 0 Area 25th Instruction Register	0x0000_0000
INITSCODE0_25	0xF510_009C	R/W	DMZ Configurable Initial Secure Code 0 Area 26th Instruction Register	0x0000_0000
INITSCODE0_26	0xF510_00A0	R/W	DMZ Configurable Initial Secure Code 0 Area 27th Instruction Register	0x0000_0000
INITSCODE0_27	0xF510_00A4	R/W	DMZ Configurable Initial Secure Code 0 Area 28th Instruction Register	0x0000_0000
INITSCODE0_28	0xF510_00A8	R/W	DMZ Configurable Initial Secure Code 0 Area 29th Instruction Register	0x0000_0000
INITSCODE0_29	0xF510_00AC	R/W	DMZ Configurable Initial Secure Code 0 Area 30th Instruction Register	0x0000_0000
INITSCODE1ADDR0	0xF510_00B0	R/W	DMZ Configurable Initial Secure Code 1 Area 1st Instruction Address Register	0x0000_0000
INITSCODE1ADDR1	0xF510_00B4	R/W	DMZ Configurable Initial Secure Code 1 Area 2nd Instruction Address Register	0x0000_0000
INITSCODE1ADDR2	0xF510_00B8	R/W	DMZ Configurable Initial Secure Code 1 Area 3rd Instruction Address Register	0x0000_0000
INITSCODE1ADDR3	0xF510_00BC	R/W	DMZ Configurable Initial Secure Code 1 Area 4th Instruction Address Register	0x0000_0000
INITSCODE1ADDR4	0xF510_00C0	R/W	DMZ Configurable Initial Secure Code 1 Area 5th Instruction Address Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
INITSCODE1ADDR5	0xF510_00C4	R/W	DMZ Configurable Initial Secure Code 1 Area 6th Instruction Address Register	0x0000_0000
INITSCODE1ADDR6	0xF510_00C8	R/W	DMZ Configurable Initial Secure Code 1 Area 7th Instruction Address Register	0x0000_0000
INITSCODE1ADDR7	0xF510_00CC	R/W	DMZ Configurable Initial Secure Code 1 Area 8th Instruction Address Register	0x0000_0000
INITSCODE1_0	0xF510_00D0	R/W	DMZ Configurable Initial Secure Code 1 Area 1st Instruction Register	0x0000_0000
INITSCODE1_1	0xF510_00D4	R/W	DMZ Configurable Initial Secure Code 1 Area 2nd Instruction Register	0x0000_0000
INITSCODE1_2	0xF510_00D8	R/W	DMZ Configurable Initial Secure Code 1 Area 3rd Instruction Register	0x0000_0000
INITSCODE1_3	0xF510_00DC	R/W	DMZ Configurable Initial Secure Code 1 Area 4th Instruction Register	0x0000_0000
INITSCODE1_4	0xF510_00E0	R/W	DMZ Configurable Initial Secure Code 1 Area 5th Instruction Register	0x0000_0000
INITSCODE1_5	0xF510_00E4	R/W	DMZ Configurable Initial Secure Code 1 Area 6th Instruction Register	0x0000_0000
INITSCODE1_6	0xF510_00E8	R/W	DMZ Configurable Initial Secure Code 1 Area 7th Instruction Register	0x0000_0000
INITSCODE1_7	0xF510_00EC	R/W	DMZ Configurable Initial Secure Code 1 Area 8th Instruction Register	0x0000_0000

3.1 SDM CONTROL REGISTER (SDM_CON, R/W, ADDRESS = 0XF510_0000)

SDM_CON	Bit	Description	Reset Value
SDMOn	[0]	System Domain Transition Method Select Bit. (0 = TZ Software, 1 = SDM)	1'b0
SecureIntMode	[1]	Secure Interrupt Mode Select Bit. (0 = IRQ/FIQ is used for Secure Interrupt, 1 = FIQ Only is used for Secure Interrupt)	1'b0
DMZEntranceMode	[3:2]	DMZ Entrance Mode Select Bit. (0 :SMI, 1 = IMZOFF<->DMZ Entrance Address 2 = NC-PA Random Branch Mode)	2'b0
DMZEntr2FailHandling	[4]	DMZ Entrance Mode 2 Fail Handling Select Bit. (0 = Fail -> Not Violation, 1= Fail -> Violation)	1'b0
SCodeOnlyExecution	[5]	Secure Domain Secure Code Only Execution Select Bit. (0 = Normal/Secure Code Execution, 1= Secure Code Only Execution)	1'b0
NCodeViolaMode	[8:6]	Normal Code State Violation Mode Select Bit. (3'b001=IRQ, 3'b010=FIQ, 3'b100=System Reset)	1'b0
NCPABrViolaMode	[11:9]	NC-PA Random Branch State Violation Handling Mode Select Bit. (3'b001=IRQ, 3'b010=FIQ, 3'b100=System Reset)	1'b0
SCodeViolaMode	[14:12]	Secure Code State Violation Handling Mode Select Bit. (3'b001=IRQ, 3'b010=FIQ, 3'b100=System Reset)	1'b0
NonCacheable	[15]	ARCACHE[1] Non-cacheable Selection Bit (0 = Non-cacheable, 1 = Cacheable)	1'b0
Reserved	[31:16]	Reserved	16'h0

3.2 SDM SECURE BOOT STATUS REGISTER (SDM_SECURE_BOOT_STS, R/W, ADDRESS = 0XF510_0004)

SDM_SECURE_BOOT_STS	Bit	Description	Reset Value
SecureBootSts	[0]	Secure Boot Status Bit. (0 = Secure Boot On-going, 1 = Secure Boot Done)	1'b0
Reserved	[31:1]	Reserved	31'h0

3.3 SDM STATUS REGISTER (SDM_STS, R, ADDRESS = 0XF510_0008)

SDM_STS	Bit	Description	Reset Value
NCodeViolaCase1	[0]	Normal Code Violation Case 1 Status Bit.	1'b0
NCodeViolaCase2	[1]	Normal Code Violation Case 2 Status Bit.	1'b0
NCPABrFailCase1	[2]	NC-PA Random Branch Fail Case 1 Status Bit.	1'b0
NCPABrFailCase2	[3]	NC-PA Random Branch Fail Case 2 Status Bit.	1'b0
NCPABrFailCase3	[4]	NC-PA Random Branch Fail Case 3 Status Bit.	1'b0
NCPABrFailCase4	[5]	NC-PA Random Branch Fail Case 4 Status Bit.	1'b0
SCodeViolaDetect	[6]	Secure Code Violation Detect Status Bit.	1'b0
SDMSecureDomainOn	[7]	Secure Domain On Status Bit	1'b1
Reserved	[31:8]	Reserved	25'h0

3.4 SDM FIQ ON TO SBIT ON INTERVAL DELAY CYCLE COUNT REGISTER (SDM_FIQ2SBITON_DLY, R/W, ADDRESS = 0XF510_000C)

SDM_FIQ2SBITON_DLY	Bit	Description	Reset Value
Fiq2SBitOnDly	[15:0]	FIQ On to SBIT On Interval Delay Cycle Count Bit.	16'h0
Reserved	[31:16]	Reserved	16'h0

3.5 SDM IMZ OFF TO DMZ ENTRANCE INTERVAL DELAY CYCLE COUNT REGISTER (SDM_IMZOFF2DMZ_DLY, R/W, ADDRESS = 0XF510_0010)

SDM_IMZOFF2DMZ_DLY	Bit	Description	Reset Value
ImzOff2DmzDly	[7:0]	IMZ OFF to DMZ Entrance Interval Delay Cycle Count Bit.	8'h0
Reserved	[31:8]	Reserved	24'h0

3.6 SDM DMZ ENTRANCE MODE 2 NC-PA RANDOM BRANCH START ADDRESS REGISTER (SDM_RANDOM_BR_ADDR, R/W, ADDRESS = 0XF510_0014)

SDM_RANDOM_BR_ADDR	Bit	Description	Reset Value

RandomBrAddr	[31:0]	DMZ Entrance Mode 2 NC-PA Random Branch Start Address Bit.	32'h0

3.7 SDM DMZ ENTRANCE MODE 2 NC-PA RANDOM BRANCH COUNT REGISTER (SDM_RANDOM_BR_CNT, R/W, ADDRESS = 0XF510_0018)

SDM_RANDOM_BR_CNT	Bit	Description	Reset Value
RandomBrCnt	[7:0]	DMZ Entrance Mode 2 NC-PA Random Branch Count Bit.	8'h0
Reserved	[31:8]		24'h0

3.8 SDM DMZ ENTRANCE MODE 2 NC-PA RANDOM BRANCH LFSR SEED POLYNOMIAL REGISTER (SDM_SEED_POLYNOMIAL, W, ADDRESS = 0XF510_001C)

SDM_SEED_POLYNOMIAL	Bit	Description	Reset Value
SeedPolynomial	[29:0]	DMZ Entrance Mode 2 NC-PA Random Branch LFSR Seed Polynomial Bit.	30'h0
Reserved	[31:30]		2'b0

3.9 SDM DMZ ENTRANCE ADDRESS REGISTER (SDM_DMZ_ENTR_ADDR, R/W, ADDRESS = 0XF510_0020)

SDM_DMZ_ENTR_ADDR	Bit	Description	Reset Value
DmzEntrAddr	[31:0]	DMZ Entrance Address Bit.	32'h0

3.10 SDM DMZ EXIT ADDRESS REGISTER (SDM_DMZ_EXIT_ADDR, R/W, ADDRESS = 0XF510_0024)

SDM_DMZ_EXIT_ADDR	Bit	Description	Reset Value
DmzExitAddr	[31:0]	DMZ Exit Address Bit.	32'h0

3.11 SDM SECURE CODE ENTRANCE ADDRESS REGISTER (SDM_SCODE_ENTR_ADDR, R/W, ADDRESS = 0XF510_0028)

SDM_SCODE_ENTR_ADDR	Bit	Description	Reset Value
ScodeEntrAddr	[31:0]	Secure Code Entrance Address Bit.	32'h0

3.12 SDM SECURE CODE BASE ADDRESS REGISTER (SDM_SCODE_BASE_ADDR, R/W, ADDRESS = 0XF510_002C)

SDM_SCODE_BASE_ADDR	Bit	Description	Reset Value
ScodeBaseAddr	[31:0]	Secure Code Base Address Bit.	32'h0

3.13 SDM SECURE CODE BASE MASK VALUE REGISTER (SDM_SCODE_BASE_MASK, R/W, ADDRESS = 0XF510_0030)

SDM_SCODE_BASE_MASK	Bit	Description	Reset Value
ScodeBaseMask	[31:0]	Secure Code Base Mask Value Bit.	32'h0

3.14 SDM DMZ ENTRANCE MODE 2 NC-PA RANDOM BRANCH RETURN INSTRUCTION REGISTER (SDM_BL_RETURN_INST, R/W, ADDRESS = 0XF510_0034)

SDM_BL_RETURN_INST	Bit	Description	Reset Value
BlRetrunInst	[31:0]	DMZ Entrance Mode 2 NC-PA Random Branch Return Instruction Bit.	32'h0

3.15 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 1ST INSTRUCTION REGISTER (INITSCODE0_0, R/W, ADDRESS = 0XF510_0038)

INITSCODE0_0	Bit	Description	Reset Value
InitScode0_0	[31:0]	DMZ Configurable Initial Secure Code 0 Area 1st Instruction Bit.	32'h0

3.16 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 2ND INSTRUCTION REGISTER (INITSCODE0_1, R/W, ADDRESS = 0XF510_003C)

INITSCODE0_1	Bit	Description	Reset Value
InitScode0_1	[31:0]	DMZ Configurable Initial Secure Code 0 Area 2nd Instruction Bit.	32'h0

3.17 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 3RD INSTRUCTION REGISTER (INITSCODE0_2, R/W, ADDRESS = 0XF510_0040)

INITSCODE0_2	Bit	Description	Reset Value
InitScode0_2	[31:0]	DMZ Configurable Initial Secure Code 0 Area 3rd Instruction Bit.	32'h0

3.18 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 4TH INSTRUCTION REGISTER (INITSCODE0_3, R/W, ADDRESS = 0XF510_0044)

INITSCODE0_3	Bit	Description	Reset Value
InitScode0_3	[31:0]	DMZ Configurable Initial Secure Code 0 Area 4th Instruction Bit.	32'h0

3.19 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 5TH INSTRUCTION REGISTER (INITSCODE0_4, R/W, ADDRESS = 0XF510_0048)

INITSCODE0_4	Bit	Description	Reset Value
InitScode0_4	[31:0]	DMZ Configurable Initial Secure Code 0 Area 5th Instruction Bit.	32'h0

3.20 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 6TH INSTRUCTION REGISTER (INITSCODE0_5, R/W, ADDRESS = 0XF510_004C)

INITSCODE0_5	Bit	Description	Reset Value
InitScode0_5	[31:0]	DMZ Configurable Initial Secure Code 0 Area 6th Instruction Bit.	32'h0

3.21 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 7TH INSTRUCTION REGISTER (INITSCODE0_6, R/W, ADDRESS = 0XF510_0050)

INITSCODE0_6	Bit	Description	Reset Value
InitScode0_6	[31:0]	DMZ Configurable Initial Secure Code 0 Area 7th Instruction Bit.	32'h0

3.22 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 8TH INSTRUCTION REGISTER (INITSCODE0_7, R/W, ADDRESS = 0XF510_0054)

INITSCODE0_7	Bit	Description	Reset Value
InitScode0_7	[31:0]	DMZ Configurable Initial Secure Code 0 Area 8th Instruction Bit.	32'h0

3.23 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 9TH INSTRUCTION REGISTER (INITSCODE0_8, R/W, ADDRESS = 0XF510_0058)

INITSCODE0_8	Bit	Description	Reset Value
InitScode0_8	[31:0]	DMZ Configurable Initial Secure Code 0 Area 9th Instruction Bit.	32'h0

3.24 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 10TH INSTRUCTION REGISTER (INITSCODE0_9, R/W, ADDRESS = 0XF510_005C)

INITSCODE0_9	Bit	Description	Reset Value
InitScode0_9	[31:0]	DMZ Configurable Initial Secure Code 0 Area 10th Instruction Bit.	32'h0

3.25 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 11TH INSTRUCTION REGISTER (INITSCODE0_10, R/W, ADDRESS = 0XF510_0060)

INITSCODE0_10	Bit	Description	Reset Value
InitScode0_10	[31:0]	DMZ Configurable Initial Secure Code 0 Area 11th Instruction Bit.	32'h0

3.26 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 12TH INSTRUCTION REGISTER (INITSCODE0_11, R/W, ADDRESS = 0XF510_0064)

INITSCODE0_11	Bit	Description	Reset Value
InitScode0_11	[31:0]	DMZ Configurable Initial Secure Code 0 Area 12th Instruction Bit.	32'h0

3.27 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 13TH INSTRUCTION REGISTER (INITSCODE0_12, R/W, ADDRESS = 0XF510_0068)

INITSCODE0_12	Bit	Description	Reset Value
InitScode0_12	[31:0]	DMZ Configurable Initial Secure Code 0 Area 13th Instruction Bit.	32'h0

3.28 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 14TH INSTRUCTION REGISTER (INITSCODE0_13, R/W, ADDRESS = 0XF510_006C)

INITSCODE0_13	Bit	Description	Reset Value
InitScode0_13	[31:0]	DMZ Configurable Initial Secure Code 0 Area 14th Instruction Bit.	32'h0

3.29 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 15TH INSTRUCTION REGISTER (INITSCODE0_14, R/W, ADDRESS = 0XF510_0070)

INITSCODE0_14	Bit	Description	Reset Value
InitScode0_14	[31:0]	DMZ Configurable Initial Secure Code 0 Area 15th Instruction Bit.	32'h0

3.30 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 16TH INSTRUCTION REGISTER (INITSCODE0_15, R/W, ADDRESS = 0XF510_0074)

INITSCODE0_15	Bit	Description	Reset Value
InitScode0_15	[31:0]	DMZ Configurable Initial Secure Code 0 Area 16th Instruction Bit.	32'h0

3.31 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 17TH INSTRUCTION REGISTER (INITSCODE0_16, R/W, ADDRESS = 0XF510_0078)

INITSCODE0_16	Bit	Description	Reset Value
InitScode0_16	[31:0]	DMZ Configurable Initial Secure Code 0 Area 17th Instruction Bit.	32'h0

3.32 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 18TH INSTRUCTION REGISTER (INITSCODE0_17, R/W, ADDRESS = 0XF510_007C)

INITSCODE0_17	Bit	Description	Reset Value
InitScode0_17	[31:0]	DMZ Configurable Initial Secure Code 0 Area 18th Instruction Bit.	32'h0

3.33 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 19TH INSTRUCTION REGISTER (INITSCODE0_18, R/W, ADDRESS = 0XF510_0080)

INITSCODE0_18	Bit	Description	Reset Value
InitScode0_18	[31:0]	DMZ Configurable Initial Secure Code 0 Area 19th Instruction Bit.	32'h0

3.34 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 20TH INSTRUCTION REGISTER (INITSCODE0_19, R/W, ADDRESS = 0XF510_0084)

INITSCODE0_19	Bit	Description	Reset Value
InitScode0_19	[31:0]	DMZ Configurable Initial Secure Code 0 Area 20th Instruction Bit.	32'h0

3.35 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 21TH INSTRUCTION REGISTER (INITSCODE0_20, R/W, ADDRESS = 0XF510_0088)

INITSCODE0_20	Bit	Description	Reset Value
InitScode0_20	[31:0]	DMZ Configurable Initial Secure Code 0 Area 21th Instruction Bit.	32'h0

3.36 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 22TH INSTRUCTION REGISTER (INITSCODE0_21, R/W, ADDRESS = 0XF510_008C)

INITSCODE0_21	Bit	Description	Reset Value
InitScode0_21	[31:0]	DMZ Configurable Initial Secure Code 0 Area 22th Instruction Bit.	32'h0

3.37 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 23TH INSTRUCTION REGISTER (INITSCODE0_22, R/W, ADDRESS = 0XF510_0090)

INITSCODE0_22	Bit	Description	Reset Value
InitScode0_22	[31:0]	DMZ Configurable Initial Secure Code 0 Area 23th Instruction Bit.	32'h0

3.38 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 24TH INSTRUCTION REGISTER (INITSCODE0_23, R/W, ADDRESS = 0XF510_0094)

INITSCODE0_23	Bit	Description	Reset Value
InitScode0_23	[31:0]	DMZ Configurable Initial Secure Code 0 Area 24th Instruction Bit.	32'h0

3.39 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 25TH INSTRUCTION REGISTER (INITSCODE0_24, R/W, ADDRESS = 0XF510_0098)

INITSCODE0_24	Bit	Description	Reset Value
InitScode0_24	[31:0]	DMZ Configurable Initial Secure Code 0 Area 25th Instruction Bit.	32'h0

3.40 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 26TH INSTRUCTION REGISTER (INITSCODE0_25, R/W, ADDRESS = 0XF510_009C)

INITSCODE0_25	Bit	Description	Reset Value
InitScode0_25	[31:0]	DMZ Configurable Initial Secure Code 0 Area 26th Instruction Bit.	32'h0

3.41 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 27TH INSTRUCTION REGISTER (INITSCODE0_26, R/W, ADDRESS = 0XF510_00A0)

INITSCODE0_26	Bit	Description	Reset Value
InitScode0_26	[31:0]	DMZ Configurable Initial Secure Code 0 Area 27th Instruction Bit.	32'h0

3.42 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 28TH INSTRUCTION REGISTER (INITSCODE0_27, R/W, ADDRESS = 0XF510_00A4)

INITSCODE0_27	Bit	Description	Reset Value
InitScode0_27	[31:0]	DMZ Configurable Initial Secure Code 0 Area 28th Instruction Bit.	32'h0

3.43 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 29TH INSTRUCTION REGISTER (INITSCODE0_28, R/W, ADDRESS = 0XF510_00A8)

INITSCODE0_28	Bit	Description	Reset Value
InitScode0_28	[31:0]	DMZ Configurable Initial Secure Code 0 Area 29th Instruction Bit.	32'h0

3.44 DMZ CONFIGURABLE INITIAL SECURE CODE 0 AREA 30TH INSTRUCTION REGISTER (INITSCODE0_29, R/W, ADDRESS = 0XF510_00AC)

INITSCODE0_29	Bit	Description	Reset Value
InitScode0_29	[31:0]	DMZ Configurable Initial Secure Code 0 Area 30th Instruction Bit.	32'h0

3.45 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 1ST INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR0, R/W, ADDRESS = 0XF510_00B0)

INITSCODE1ADDR0	Bit	Description	Reset Value
InitScode1Addr0	[19:0]	DMZ Configurable Initial Secure Code 1 Area 1st Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.46 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 2ND INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR1, R/W, ADDRESS = 0XF510_00B4)

INITSCODE1ADDR1	Bit	Description	Reset Value
InitScode1Addr1	[19:0]	DMZ Configurable Initial Secure Code 1 Area 2nd Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.47 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 3RD INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR2, R/W, ADDRESS = 0XF510_00B8)

INITSCODE1ADDR2	Bit	Description	Reset Value
InitScode1Addr2	[19:0]	DMZ Configurable Initial Secure Code 1 Area 3rd Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.48 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 4TH INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR3, R/W, ADDRESS = 0XF510_00BC)

INITSCODE1ADDR3	Bit	Description	Reset Value
InitScode1Addr3	[19:0]	DMZ Configurable Initial Secure Code 1 Area 4th Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.49 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 5TH INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR4, R/W, ADDRESS = 0XF510_00C0)

INITSCODE1ADDR4	Bit	Description	Reset Value
InitScode1Addr4	[19:0]	DMZ Configurable Initial Secure Code 1 Area 5th Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.50 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 6TH INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR5, R/W, ADDRESS = 0XF510_00C4)

INITSCODE1ADDR5	Bit	Description	Reset Value
InitScode1Addr5	[19:0]	DMZ Configurable Initial Secure Code 1 Area 6th Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.51 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 7TH INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR6, R/W, ADDRESS = 0XF510_00C8)

INITSCODE1ADDR6	Bit	Description	Reset Value
InitScode1Addr6	[19:0]	DMZ Configurable Initial Secure Code 1 Area 7th Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.52 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 8TH INSTRUCTION ADDRESS REGISTER (INITSCODE1ADDR7, R/W, ADDRESS = 0XF510_00CC)

INITSCODE1ADDR7	Bit	Description	Reset Value
InitScode1Addr7	[19:0]	DMZ Configurable Initial Secure Code 1 Area 8th Instruction Address Bit.	20'h0
Reserved	[31:20]	Reserved	12'h0

3.53 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 1ST INSTRUCTION REGISTER (INITSCODE1_0, R/W, ADDRESS = 0XF510_00D0)

INITSCODE1_0	Bit	Description	Reset Value
InitScode1_0	[31:0]	DMZ Configurable Initial Secure Code 1 Area 1st Instruction Bit.	32'h0

3.54 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 2ND INSTRUCTION REGISTER (INITSCODE1_1, R/W, ADDRESS = 0XF510_00D4)

INITSCODE1_1	Bit	Description	Reset Value
InitScode1_1	[31:0]	DMZ Configurable Initial Secure Code 1 Area 2nd Instruction Bit.	32'h0

3.55 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 3RD INSTRUCTION REGISTER (INITSCODE1_2, R/W, ADDRESS = 0XF510_00D8)

INITSCODE1_2	Bit	Description	Reset Value
InitScode1_2	[31:0]	DMZ Configurable Initial Secure Code 1 Area 3rd Instruction Bit.	32'h0

3.56 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 4TH INSTRUCTION REGISTER (INITSCODE1_3, R/W, ADDRESS = 0XF510_00DC)

INITSCODE1_3	Bit	Description	Reset Value
InitScode1_3	[31:0]	DMZ Configurable Initial Secure Code 1 Area 4th Instruction Bit.	32'h0

3.57 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 5TH INSTRUCTION REGISTER (INITSCODE1_4, R/W, ADDRESS = 0XF510_00E0)

INITSCODE1_4	Bit	Description	Reset Value
InitScode1_4	[31:0]	DMZ Configurable Initial Secure Code 1 Area 5th Instruction Bit.	32'h0

3.58 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 6TH INSTRUCTION REGISTER (INITSCODE1_5, R/W, ADDRESS = 0XF510_00E4)

INITSCODE1_5	Bit	Description	Reset Value
InitScode1_5	[31:0]	DMZ Configurable Initial Secure Code 1 Area 6th Instruction Bit.	32'h0

3.59 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 7TH INSTRUCTION REGISTER (INITSCODE1_6, R/W, ADDRESS = 0XF510_00E8)

INITSCODE1_6	Bit	Description	Reset Value
InitScode1_6	[31:0]	DMZ Configurable Initial Secure Code 1 Area 7th Instruction Bit.	32'h0

3.60 DMZ CONFIGURABLE INITIAL SECURE CODE 1 AREA 8TH INSTRUCTION REGISTER (INITSCODE1_7, R/W, ADDRESS = 0XF510_00EC)

INITSCODE1_7	Bit	Description	Reset Value
InitScode1_7	[31:0]	DMZ Configurable Initial Secure Code 1 Area 8th Instruction Bit.	32'h0

NOTE: All SDM SFR Write is protected after Secure Boot Operation.

NOTES

11.3 CRYPTO ENGINE BLOCK

1 INTRODUCTION

1.1 FEATURES

The Advanced Crypto Engine (ACE) is a crypto function accelerator targeted for general purpose mobile processors such as application processor (AP) and modem chip. ACE is integrated in a SoC or used as a co-processor to enhance the security level of the target processor.

The architecture of ACE also provides a high-speed bulk data processing by providing double-layer AHB bus and FIFOs. FIFO-Rx and FIFO-Tx are programmed to monitor AES or DES/TDES or SHA-1/PRNG module and automatically transfer input/output data from the target module. This scheme does not require host's intervention and achieves high-speed bulk data processing.

ACE has Montgomery Multiplier with dedicated 2KB memory for large-number modular multiplication up to 2048-bit modulus. This modular multiplier is used for implementing public key cryptographic functions like RSA.

Additionally, ACE has a secure JTAG controller. This controller applies the authentication mechanism to check the authorized user and the access mode. The system protects the system access from the unauthorized user through JTAG port.

ACE block is operated upto 83MHz frequency.

Figure 11.3-1 shows the block diagram of the ACE and its main features are as follows:

- Symmetric Key Accelerator
  - ♦ AES: ECB, CBC and CTR mode support
  - ♦ DES/TDES: ECB,and CBC mode support
- Public Key Accelerator (PKA)
  - ♦ Montgomery Multiplier (up to 2048-bit, 32-bit granule)
  - ♦ 2 KB Public-Key RAM (PKRAM)
- Hash Engine
  - ♦ SHA-1, 200Mbps@100MHz
  - ♦ H/W HMAC support
- Secure JTAG Engine
  - ♦ Password based JTAG scheme
- Random Number Generator
  - ♦ PRNG 320-bit generation per 160 cycles
- FIFO-Rx/Tx: (two 32x32b) for input and output streaming.
- AHB I/F: 4 Slave

- Interrupt signals
- DMA I/F signals

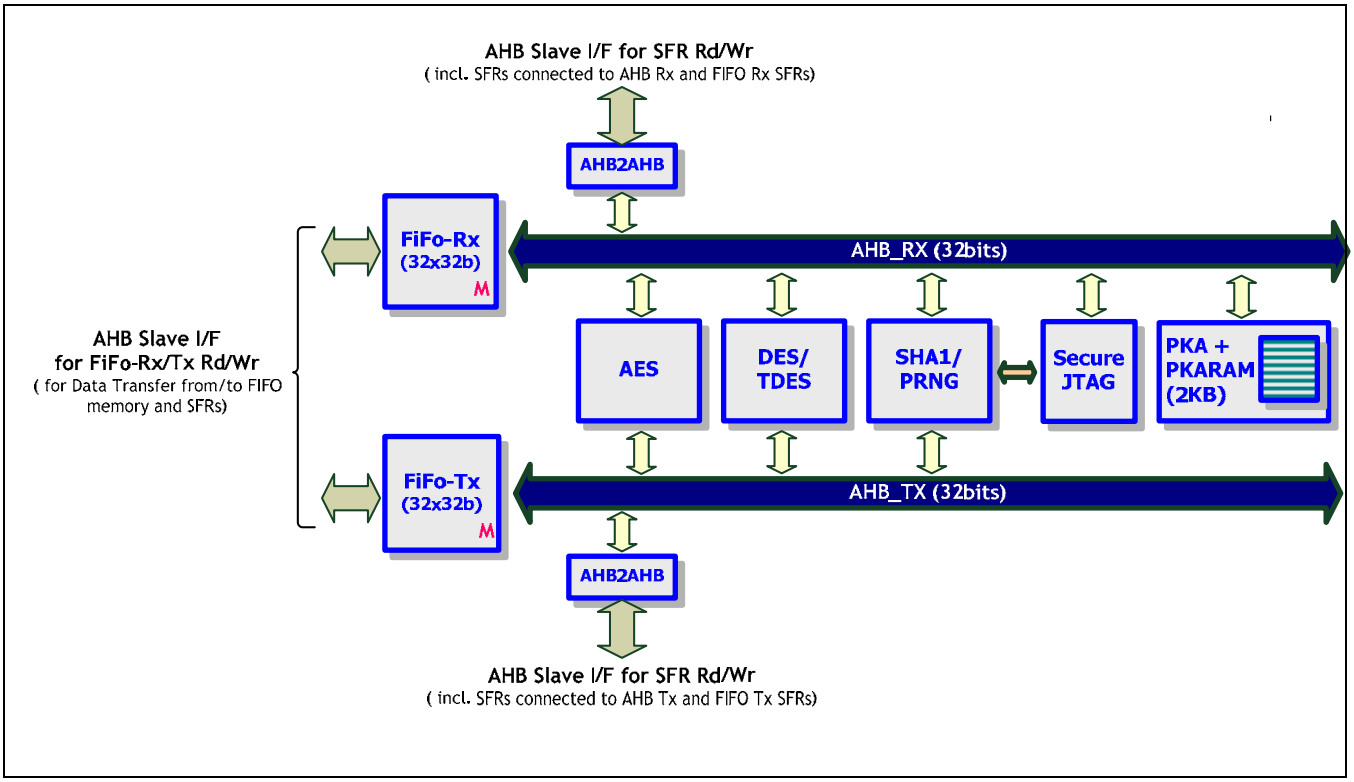


Figure 11.3-1 ACE Top Level Block Diagram

1.2 MEMORY MAPPING

Address map for ACE is divided into four *areas* and reserved areas. The base address of each area is determined by the host and you can customize according to your requirement. The four areas are:

- AHB-Rx area: Memory space and SFRs connected to AHB-Rx bus. (incl. FIFO-Rx)
- AHB-Tx area: Memory space and SFRs connected to AHB-Tx bus. (incl. FIFO-Tx)
- DMA-Rx area: Memory space and SFRs in FIFO-Rx.
- DMA-Tx area: Memory space and SFRs in FIFO-Tx.

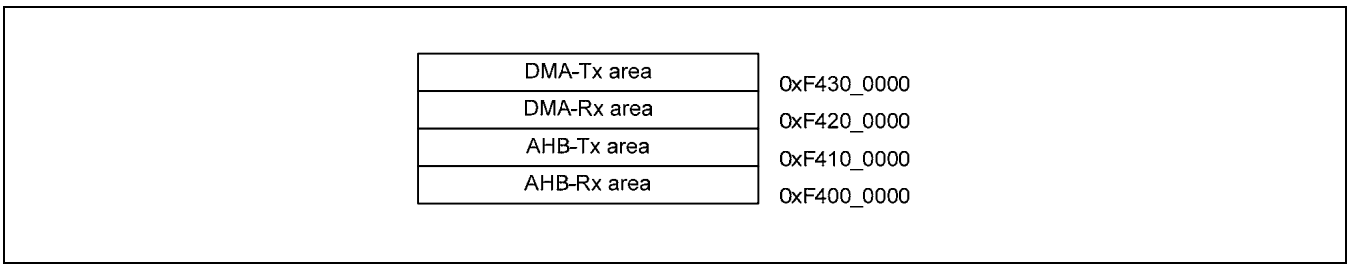


Figure 11.3-2 ACE Area Memory Map

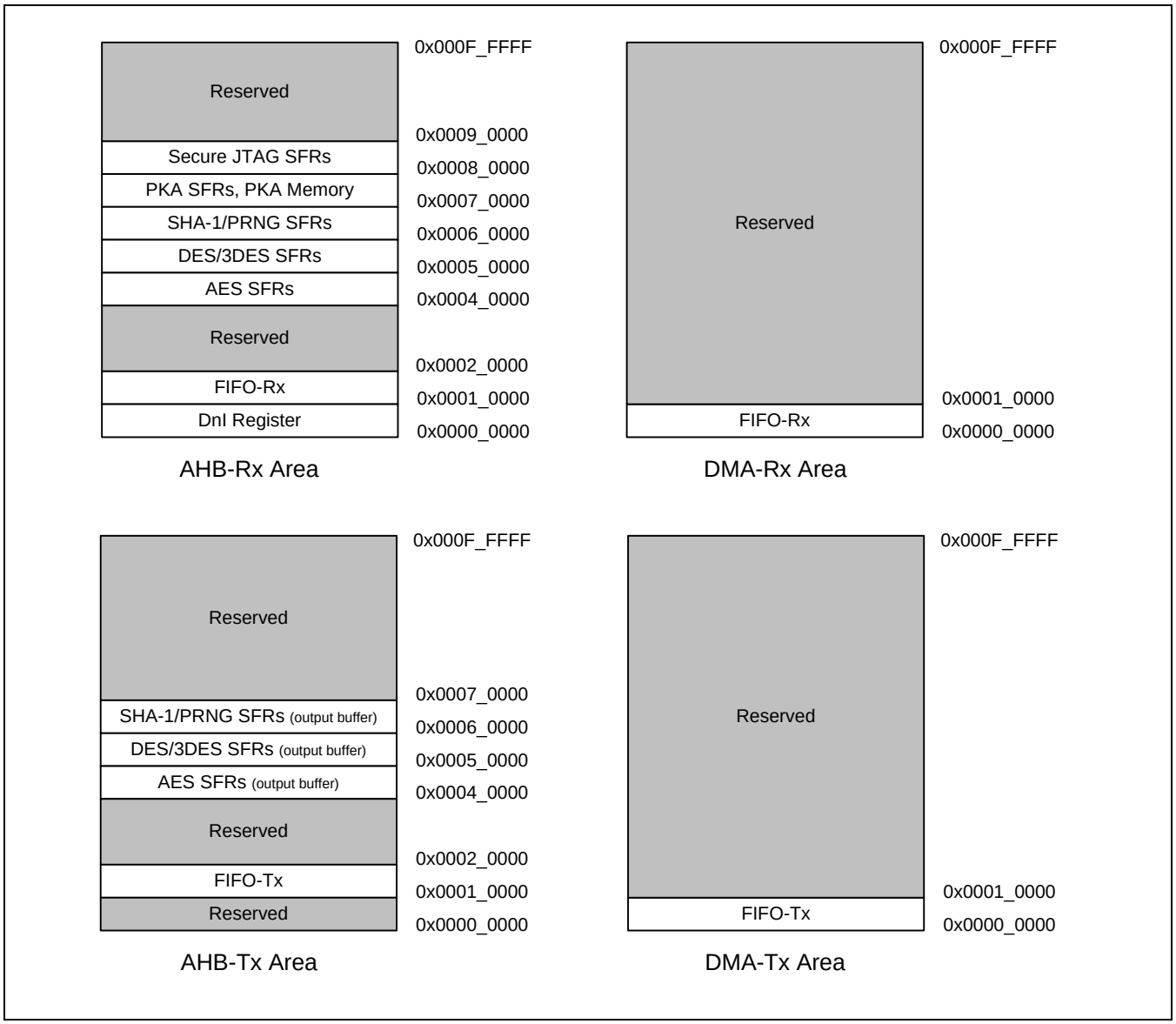


Figure 11.3-3 ACE Internal Memory Map Per Each Area

NOTE: PKA Memory: 0x0007_0000 ~ 0x0007_07ff, PKA SFRs: 0x0007_1000 ~ 0x0007_0010

2 COMPONENTS

2.1 AES (ADVANCED ENCRYPTION STANDARD) ENGINE

AES is one of the symmetric encryption modules and supports 128-bit/ 192-bit/ 256-bit key and the ECB/ CBC/ CTR operation mode. For stream decryption, it also supports 'No Decryption Key Generation Decryption (Continuous Decryption)'.

AES engine's features are summarized as follows

- AES (Advanced Encryption Standard: FIPS PUB 197) and Operation Mode Design
- AHB Slave Interface Support
- Half-Duplex Encryption/ Decryption
- Mode: ECB, CBC and CTR Mode
- Key Length: 128-bit, 192-bit, and 256-bit Support
- On-the-fly Key Scheduler
- Througput: 931Mbps@80MHz (Key: 128-bit), 788Mbps@80MHz (Key: 192-bit), 683@80MHz (Key: 256-bit)

2.2 TDES (DES: DATA ENCRYPTION STANDARD) ENGINE

ACE TDES is one of the symmetric encryption modules and operates in 64-bit key DES mode and 64-bit key Triple-DES mode. It supports ECB/ CBC operation mode.

TDES engine's features are summarized as follows:

- TDES/DES (Triple DES: FIPS PUB 46-3, DES: FIPS PUB 46-81) and Operation Mode Design
- AHB Slave Interface Support
- Half-Duplex Encryption/ Decryption
- Mode: ECB, CBC Mode
- Key Length: 192-bit(TDES), 64-bit(DES)
- On-the-fly Key Scheduler

2.3 SHA-1/PRNG ENGINE

SHA-1/PRNG supports SHA-1, HMAC and PRNG algorithms.

2.4 PKA (PUBLIC KEY ALGORITHM) ENGINE

PKA supports digital-serial radix-4 Montgomery Multiplication Method which is superior to general modular multiplication algorithms.

PKA engine's features are summarized as follows:

- Modular Multiplier Design
- Supports AHB Slave Interface
- Mode: AxB Mode, Ax1 Mode
- Supported Key Size: 2048-bit
- CryptoRAM Size: 2KB
- Performance: 1129 clocks

2.5 SECURE JTAG

Secure JTAG is implemented in ACE like other security engines.

Secure JTAG is protection for JTAG port from the mul-user who uses debugger to attack S5PC100 system. To enable or disable use e-fuse ROM Secure JTAG lock on bit. If the Secure JTAG lock on bit is set to "1", the long sequence of password should be entered into the debugger to debug S5PC100. The debugging levels are decided by the e-fuse ROM Secure Access Type bit. Set this bit to '0' to debug the all region include secure part.

Table 11.3- 1 Secure JTAG Debug Mode Selection

Secure Access Type	Secure JTAG Lock on	Debug Mode
1	0	Non-secure non-protection mode
0	0	Secure non-protection mode
1	1	Non-secure soft lock mode
0	1	Secure soft lock mode

It receives the password sequences through AHB bus from debugger and authenticates the debugger with that password. If authentication passes, it permits debugger to start debug system. The 80-bit hash value is used for authentication process.

Figure 11.3-4 shows secure JTAG implementation in S5PC100. Debugger is connected to S5PC100 through Coresight SubSystem (CSSYS) which provides AHB master operation. Before authentication, debugger can access only Secure JTAG authentication module in ACE via CSSYS's AHB access port. After authentication, secure JTAG asserts internal signals so that access controller module drives ARM's authentication input ports such as DBGEN, NIDEN, SPIDEN and SPNIDEN.

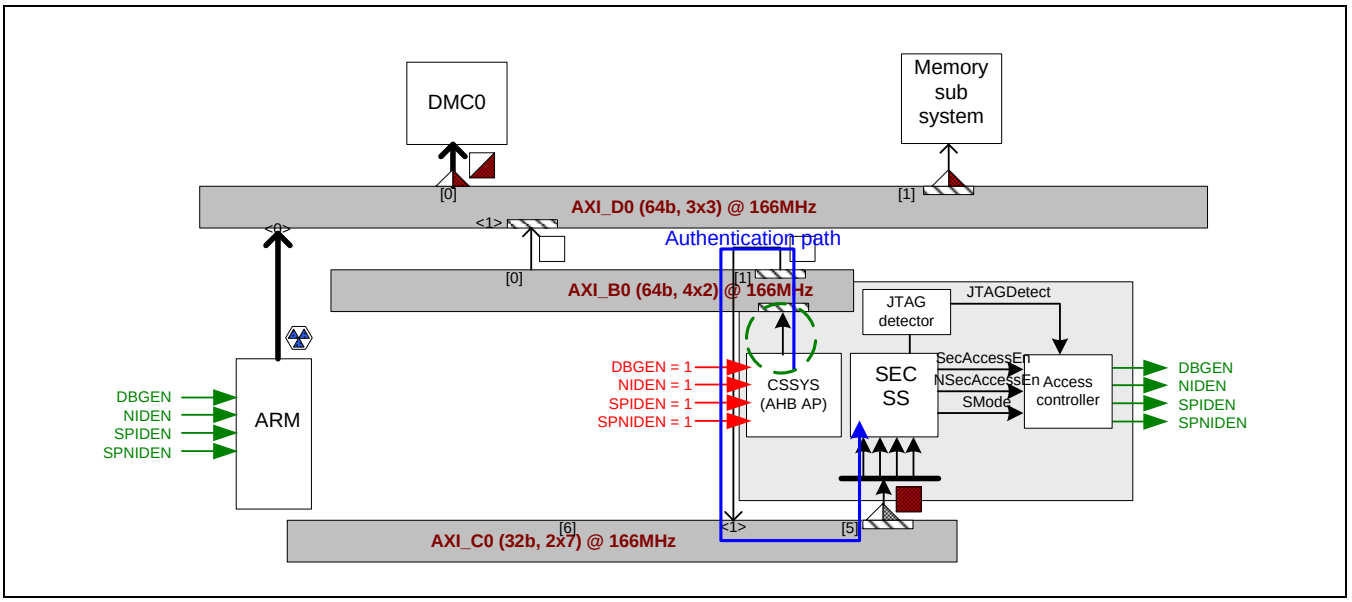


Figure 11.3-4 Secure JTAG System in S5PC100

3 PROGRAMMER'S MODEL

This Programmer's model describes ACE(SECSS) operation sequences for programmer. The ACE has three operation mode : CPU mode, FIFO mode, and DMA mode. For correct each mode operation, the ACE must be configured and started according to the following sequences. Followings are ACE's SFR(Special Function Register) configuration and operation sequence to the each mode.

3.1 CPU MODE

① Sets Control Registers of the crypto engine
② Write Data to Data Input Registers : Repeat following step from step② to step④ as needed
③ Starts crypto engine
④ Read Data form Data Output Register

3.2 FIFO MODE

① Sets Control Registers of the crypto engine
② Sets Control Register of Rx/Tx FIFO
③ Start Rx/Tx FIFO
④ Write Data to Rx FIFO
⑤ Polls Tx FIFO Transfer Done
⑥ Read Data from Tx FIFO

3.3 DMA MODE

① Sets Control Registers of crypto engine
② Sets DMA and Interrupt control register of the ACE
③ Sets Control Register of Rx/Tx FIFO
④ Start Rx/Tx FIFO
⑤ Write Data to Memory
⑥ DMA Rx Channel Start : Write Data to Rx FIFO
⑦ DMA Tx Channel Start : Read Data from Tx FIFO
⑧ Polls Tx DMA Operation Done
⑨ Polls Tx FIFO Transfer Done

3.4 CHANGE MODE FROM FIFO MODE TO DMA MODE

When programmers change ACE operation mode from FIFO mode to DMA mode, the first ACE Tx FIFO DMA request must be cleared by executing DMA ACK instruction(DMAFLUSHP P1)

① FIFO Mode Operation Done
② Clear Tx FIFO DMA Request
③ Sets Control Registers of crypto engine
④ Sets DMA and Interrupt control register of the ACE
⑤ Sets Control Register of Rx/Tx FIFO
⑥ Start Rx/Tx FIFO
⑦ Write Data to Memory
⑧ DMA Rx Channel Start : Write Data to Rx FIFO
⑨ DMA Tx Channel Start : Read Data from Tx FIFO
⑩ Polls Tx DMA Operation Done

⑪ Polls Tx FIFO Transfer Done

Following is example c-code to clear Tx FIFO DMA Request of 3.4 CHANGE FROM FIFO MODE TO DMA MODE step②.

```
Main {
        ...
        TxClearReq(1);
        ...
}

void TxClearReq ( unsigned int channel ) {
        const unsigned int SECSS_PERI_NUM = 1;
        static unsigned int flush_code;
        channel &= 0x00000007;
        flush_code = 0xff000035 | ( SECSS_PERI_NUM << 11 );
        DMA_DBGINST0 = 0x00a00000 | ( channel << 24 );
        DMA_DBGINST1 = ( unsigned int )&flush_code;
        DMA_DMBCMD = 0x0;
}
```

4 REGISTER DESCRIPTION

4.1 REGISTER MAP

4.1.1 DnI (DMA and Interrupt) Configuration Register Area (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
DnI_Cfg	0xF400_0000	R/W	DMA and interrupt configuration Control/ Status Register.	0x0000_0000

4.1.2 FIFO-Rx Area (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
FRx_Ctrl	0xF401_0000	R/W	FIFO-Rx Control/ Status Register. (Only MSB 16-bit readable)	0x0420_0000
FRx_MLen	0xF401_0004	R/W	FIFO-Rx Message Length Register.	0x0000_0000
FRx_BlkSz	0xF401_0008	R/W	FIFO-Rx Crypto Algorithm Block Size Register.	0x0000_0000
FRx_DestAddr	0xF401_000C	R/W	FIFO-Rx Inout Buffer Address Register.	0x0000_0000
FRx_MLenCnt	0xF401_0010	R/W	FIFO-Rx Message Count Register. (Number of words left)	0x0000_0000
FRx_WrBuf	0xF401_0040 ~ 0xF401_0080	W	FIFO-Rx write buffer (32x32-bit)	0x0000_0000

NOTE: Write access to FRx_WrBuf makes FIFO-Rx to write data to the FIFO memory regardless of the address given. That is, any address between 0x0040 and 0x0080 triggers the FIFO memory write. This feature lets the programmer use burst write to the FIFO-Rx.

4.1.3 FIFO-Tx Area (AHB-Tx Area)

Register	Address	R/W	Description	Reset Value
FTx_Ctrl	0xF411_0000	R/W	FIFO-Tx Control/ Status Register. (Only MSB 16-bit readable)	0x0400_2000
FTx_MLen	0xF411_0004	R/W	FIFO-Tx Message Length Register.	0x0000_0000
FTx_BlkSz	0xF411_0008	R/W	FIFO-Tx Crypto Algorithm Block Size Register.	0x0000_0000
FTx_SrcAddr	0xF411_000C	R/W	FIFO-Tx Output Buffer Address Register.	0x0000_0000
FTx_MLenCnt	0xF411_0010	R/W	FIFO-Tx Message Count Register. (Number of words left)	0x0000_0000
FTx_RdBuf	0xF411_0040 ~ 0xF411_0080	R	FIFO-Tx read buffer (32x32-bit)	0x0000_0000

NOTE: Read access to FTx_WrBuf makes FIFO-Tx to read data from the FIFO memory regardless of the address given. That is, any address between 0x0040 and 0x0080 triggers the FIFO memory read. This feature makes the programmer use burst read to the FIFO-Tx.

4.1.4 AES SFRs (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
AES_Rx_CTRL	0xF404_0000	R/W	AES Rx Contrl / Status Register.	0x0000_0200
AES_Rx_DIN_01	0xF404_0010	R/W	AES Rx Data Input Register. 01	0x0000_0000
AES_Rx_DIN_02	0xF404_0014	R/W	AES Rx Data Input Register. 02	0x0000_0000
AES_Rx_DIN_03	0xF404_0018	R/W	AES Rx Data Input Register 03	0x0000_0000
AES_Rx_DIN_04	0xF404_001C	R/W	AES Rx Data Input Register 04	0x0000_0000
AES_Rx_DOUT_01	0xF404_0020	R	AES Rx Data Output Register 01	0x0000_0000
AES_Rx_DOUT_02	0xF404_0024	R	AES Rx Data Output Register 02	0x0000_0000
AES_Rx_DOUT_03	0xF404_0028	R	AES Rx Data Output Register 03	0x0000_0000
AES_Rx_DOUT_04	0xF404_002C	R	AES Rx Data Output Register 04	0x0000_0000
AES_Rx_KEY_01	0xF404_0080	R/W	AES Rx Key Input Register 01	0x0000_0000
AES_Rx_KEY_02	0xF404_0084	R/W	AES Rx Key Input Register 02	0x0000_0000
AES_Rx_KEY_03	0xF404_0088	R/W	AES Rx Key Input Register 03	0x0000_0000
AES_Rx_KEY_04	0xF404_008C	R/W	AES Rx Key Input Register 04	0x0000_0000
AES_Rx_KEY_05	0xF404_0090	R/W	AES Rx Key Input Register 05	0x0000_0000
AES_Rx_KEY_06	0xF404_0094	R/W	AES Rx Key Input Register 06	0x0000_0000
AES_Rx_KEY_07	0xF404_0098	R/W	AES Rx Key Input Register 07	0x0000_0000
AES_Rx_KEY_08	0xF404_009C	R/W	AES Rx Key Input Register 08	0x0000_0000
AES_Rx_IV_01	0xF404_00A0	R/W	AES Rx IV Input Register 01	0x0000_0000
AES_Rx_IV_02	0xF404_00A4	R/W	AES Rx IV Input Register 02	0x0000_0000
AES_Rx_IV_03	0xF404_00A8	R/W	AES Rx IV Input Register 03	0x0000_0000
AES_Rx_IV_04	0xF404_00AC	R/W	AES Rx IV Input Register 04	0x0000_0000
AES_Rx_CTR_01	0xF404_00B0	R/W	AES Rx Counter Preload Register 01	0x0000_0000
AES_Rx_CTR_02	0xF404_00B4	R/W	AES Rx Counter Preload Register 02	0x0000_0000
AES_Rx_CTR_03	0xF404_00B8	R/W	AES Rx Counter Preload Register 03	0x0000_0000
AES_Rx_CTR_04	0xF404_00BC	R/W	AES Rx Counter Preload Register 04	0x0000_0000

4.1.5 AES SFRs (AHB-Tx Area)

Register	Address	R/W	Description	Reset Value
AES_Tx_DOUT_01	0xF414_0020	R	AES Tx Output Data Register 01	0x0000_0000
AES_Tx_DOUT_02	0xF414_0024	R	AES Tx Output Data Register 02	0x0000_0000
AES_Tx_DOUT_03	0xF414_0028	R	AES Tx Output Data Register 03	0x0000_0000
AES_Tx_DOUT_04	0xF414_002C	R	AES Tx Output Data Register 04	0x0000_0000

4.1.6 DES/TDES SFRs (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
TDES_Rx_CTRL	0xF405_0000	R/W	DES/TDES Rx Control / Status Register	0x0000_0040
TDES_Rx_KEY1_0	0xF405_0010	R/W	DES/TDES Rx Key Input Register 1_0	0x0000_0000
TDES_Rx_KEY1_1	0xF405_0014	R/W	DES/TDES Rx Key Input Register 1_1	0x0000_0000
TDES_Rx_KEY2_0	0xF405_0018	R/W	DES/TDES Rx Key Input Register 2_0	0x0000_0000
TDES_Rx_KEY2_1	0xF405_001C	R/W	DES/TDES Rx Key Input Register 2_1	0x0000_0000
TDES_Rx_KEY3_0	0xF405_0020	R/W	DES/TDES Rx Key Input Register 3_0	0x0000_0000
TDES_Rx_KEY3_1	0xF405_0024	R/W	DES/TDES Rx Key Input Register 3_1	0x0000_0000
TDES_Rx_INPUT_0	0xF405_0040	R/W	DES/TDES Rx Data Input Register 0	0x0000_0000
TDES_Rx_INPUT_1	0xF405_0044	R/W	DES/TDES Rx Data Input Register 1	0x0000_0000
TDES_Rx_OUTPUT_0	0xF405_0048	R	DES/TDES Rx Output Data Register 0	0x0000_0000
TDES_Rx_OUTPUT_1	0xF405_004C	R	DES/TDES Rx Output Data Register 1	0x0000_0000
TDES_Rx_IV_0	0xF405_0050	R/W	TDES Rx IV Input Register 0	0x0000_0000
TDES_Rx_IV_1	0xF405_0054	R/W	TDES Rx IV Input Register 1	0x0000_0000

4.1.7 DES/TDES SFRs (AHB-Tx Area)

Register	Address	R/W	Description	Reset Value
TDES_Tx_OUTPUT_0	0xF415_0048	R	DES/TDES Rx Output Data Register 0	0x0000_0000
TDES_Tx_OUTPUT_1	0xF415_004C	R	DES/TDES Rx Output Data Register 1	0x0000_0000

4.1.8 SHA-1/PRNG SFRs (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
HASH_CONTROL	0xF406_0000	R/W	Hash engine control Register	0x0000_0000
HASH_DATA	0xF406_0004 ~ 0xF406_0020	W	HASH data or HMAC Key Input Register (If HASH_CONTROL[5] == 1'b0)	0x0000_0000
SEED_DATA_01	0xF406_0008	W	PRNG Seed data[31:0] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_02	0xF406_000C	W	PRNG Seed data[63:32] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_03	0xF406_0010	W	PRNG Seed data[95:64] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_04	0xF406_0014	W	PRNG Seed data[127:96] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_05	0xF406_0018	W	PRNG Seed data[159:128] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_06	0xF406_001C	W	PRNG Seed data[191:160] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_07	0xF406_0020	W	PRNG Seed data[223:192] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_08	0xF406_0024	W	PRNG Seed data[255:224] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_09	0xF406_0028	W	PRNG Seed data[287:256] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
SEED_DATA_10	0xF406_002C	W	PRNG Seed data[319:288] (If HASH_CONTROL[5] == 1'b1)	0x0000_0000
HASH_STATUS	0xF406_0030	R	Status check	0x0000_0010
HASH_OUTPUT_01 (PRNG_OUTPUT_01)	0xF406_0034	R	HASH (PRNG) output (h0)	0x0000_0000
HASH_OUTPUT_02 (PRNG_OUTPUT_02)	0xF406_0038	R	HASH (PRNG) output (h1)	0x0000_0000
HASH_OUTPUT_03 (PRNG_OUTPUT_03)	0xF406_003C	R	HASH (PRNG) output (h2)	0x0000_0000
HASH_OUTPUT_04 (PRNG_OUTPUT_04)	0xF406_0040	R	HASH (PRNG) output (h3)	0x0000_0000
HASH_OUTPUT_05 (PRNG_OUTPUT_05)	0xF406_0044	R	HASH (PRNG) output (h4)	0x0000_0000
PRNG_OUTPUT_06	0xF406_0048	R	PRNG output	0x0000_0000
PRNG_OUTPUT_07	0xF406_004C	R	PRNG output	0x0000_0000
PRNG_OUTPUT_08	0xF406_0050	R	PRNG output	0x0000_0000
PRNG_OUTPUT_09	0xF406_0054	R	PRNG output	0x0000_0000

PRNG_OUTPUT_10	0xF406_0058	R	PRNG output	0x0000_0000
HASH_MIDOUT_01	0xF406_005C	R	HASH_MIDOUT [159:128]	0x0000_0000
HASH_MIDOUT_02	0xF406_0060	R	HASH_MIDOUT [127:96]	0x0000_0000
HASH_MIDOUT_03	0xF406_0064	R	HASH_MIDOUT [95:64]	0x0000_0000
HASH_MIDOUT_04	0xF406_0068	R	HASH_MIDOUT [63:32]	0x0000_0000
HASH_MIDOUT_05	0xF406_006C	R	HASH_MIDOUT[31:0]	0x0000_0000
HASH_IV_01	0xF406_0070	W	HASH initial value 01	0x0000_0000
HASH_IV_02	0xF406_0074	W	HASH initial value 02	0x0000_0000
HASH_IV_03	0xF406_0078	W	HASH initial value 03	0x0000_0000
HASH_IV_04	0xF406_007C	W	HASH initial value 04	0x0000_0000
HASH_IV_05	0xF406_0080	W	HASH initial value 05	0x0000_0000
PRE_MSG_LENGTH_1	0xF406_0084	W	Pre HASH length [63:32]	0x0000_0000
PRE_MSG_LENGTH_2	0xF406_0088	W	Pre HASH length [31: 0]	0x0000_0000

4.1.9 SHA-1/PRNG SFRs (AHB-Tx Area)

Register	Address	R/W	Description	Reset Value
HASH_OUTPUT_01 (PRNG_OUTPUT_01)	0xF416_0034	R	HASH (PRNG) Output (h0)	0x0000_0000
HASH_OUTPUT_02 (PRNG_OUTPUT_02)	0xF416_0038	R	HASH (PRNG) Output (h1)	0x0000_0000
HASH_OUTPUT_03 (PRNG_OUTPUT_03)	0xF416_003C	R	HASH (PRNG) Output (h2)	0x0000_0000
HASH_OUTPUT_04 (PRNG_OUTPUT_04)	0xF416_0040	R	HASH (PRNG) Output (h3)	0x0000_0000
HASH_OUTPUT_05 (PRNG_OUTPUT_05)	0xF416_0044	R	HASH (PRNG) Output (h4)	0x0000_0000
PRNG_OUTPUT_06	0xF416_0048	R	PRNG Output	0x0000_0000
PRNG_OUTPUT_07	0xF416_004C	R	PRNG Output	0x0000_0000
PRNG_OUTPUT_08	0xF416_0050	R	PRNG Output	0x0000_0000
PRNG_OUTPUT_09	0xF416_0054	R	PRNG Output	0x0000_0000
PRNG_OUTPUT_10	0xF416_0058	R	PRNG Output	0x0000_0000

4.1.10 PKA Memory and SFRs (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
PKA_MEM	0xF407_0000 ~ 0xF407_07FF	R/W	PKA Memory (2KB)	0x0000_0000
PKA_SFR0	0xF407_1000	R/W	PKA Special Function Register 0	0x0000_0000
PKA_SFR1	0xF407_1004	R/W	PKA Special Function Register 1	0x0000_0000
PKA_SFR2	0xF407_1008	R/W	PKA Special Function Register 2	0x0000_0000
PKA_SFR3	0xF407_100C	R/W	PKA Special Function Register 3	0x0000_0000
PKA_SFR4	0xF407_1010	R/W	PKA Special Function Register 4	0x0000_0000

4.1.11 Secure JTAG SFRs (AHB-Rx Area)

Register	Address	R/W	Description	Reset Value
SJ_HW_VERSION_ REG	0xF408_0000	R	Secure JTAG H/W Version Register	0xxxxx_xxxx
SJ_STATUS_REG	0xF408_0004	R	Secure JTAG Status Register	0x0000_0000
SJ_RESULT_REG	0xF408_0008	R	Secure JTAG Result Register	0x0000_0000
SJ_HEADER_INFO_ REG	0xF408_000C	R	Secure JTAG Header Information Register	0x0000_0000
SJ_PASSWORD_REG	0xF408_0020	R/W	Secure JTAG Password Register	0x0000_0000
SJ_PASSWORD_ CTRL_REG	0xF408_0024	W	Secure JTAG Password Control Register	0x0000_0000

4.2 DMA AND INTERRUPT CONFIGURATION REGISTER

4.2.1 DMA and Interrupt Configuration Control/Status Register. (DnI_Cfg, R/W, Address = 0xF400_0000)

DnI_Cfg	Bit	Description	Reset Value
WrPrivMismatch	[31]	SFR Write Access Privilege Mismatch Status bit. If set to '1', SFR Write Access Privilege Mismatch occurs.	1'b0
RdPrivMismatch	[30]	SFR Read Access Privilege Mismatch Status bit. If set to '1', SFR Read Access Privilege Mismatch occurs.	1'b0
Reserved	[29:23]	Reserved	
SHA_intr_Status	[22]	SHA-1/PRNG interrupt status (Resets to 0 if read).	1'b0
DES_intr_Status	[21]	DES/TDES interrupt status (Resets to 0 if read).	1'b0
AES_intr_Status	[20]	AES interrupt status (Resets to 0 if read).	1'b0
Reserved	[19:18]	Reserved	
FTx_intr_Status	[17]	FIFO-Tx interrupt status (Resets to 0 if read).	1'b0
FRx_intr_Status	[16]	FIFO-Rx interrupt status (Resets to 0 if read).	1'b0
Reserved	[15]	Reserved	
SHA_intr_En	[14]	SHA-1/PRNG interrupt enabled	1'b0
DES_intr_En	[13]	DES/TDES interrupt enabled	1'b0
AES_intr_En	[12]	AES interrupt enabled	1'b0
Reserved	[11:10]	Reserved	
FTx_intr_En	[9]	FIFO-Tx interrupt enabled	1'b0
FRx_intr_En	[8]	FIFO-Rx interrupt enabled	1'b0
Reserved	[7]	Reserved	
TxTrgLevel	[6:5]	Tx side DMA trigger level setting. 00 = 8-word 01 = 16-word 10 = 24-word 11 = Reserved	2'b00
TxDmaEnb	[4]	Tx side DMA enable bit (1= Enable)	1'b0
Reserved	[3]	Reserved	
RxTrgLevel	[2:1]	Rx side DMA trigger level setting. 00 = 8-word 01 = 16-word 10 = 24-word 11 = Reserved	2'b00
RxDmaEnb	[0]	Rx side DMA enable bit (1= Enable)	1'b0

Note: 1. The SHA-1 related modules(HMAC, SHA-1, and PRNG) do not support 24-word trigger level setting (DnI_Cfg[2:1], DnI_Cfg[6:5]).

2. The SECSS interrupt must be masked in case of SECSS DMA mode operation.

3. The SECSS does not recommend SECSS interrupt mode instead of polling mode in case of SECSS FiFo mode operation.

4. When users change SECSS operation mode from FiFo mode to DMA mode, the first SECSS Tx FiFo DMA request must be cleared by executing DMA ACK instruction(DMAFLUSHP).

4.3 FIFO-RX/TX REGISTER

4.3.1 FIFO-Rx Control Registers (FRx_Ctrl, R/W, Address = 0xF401_0000)

FRx_Ctrl	Bit	Description	Reset Value
FRx_WrPrivError	[31]	Sets to 1 if write access to FIFO-Rx has resulted in a privilege error (e.g. Host or the command are not allowed to access FIFO-Rx). Resets to 0 on FRx_Reset = 1. Read only.	1'b0
FRx_RdPrivError	[30]	Sets to 1 if read access to FIFO-Rx has resulted in a privilege error (e.g. Host or the command are not allowed to access FIFO-Rx). Resets to 0 on FRx_Reset = 1. Read only.	1'b0
Reserved	[29:28]	Reserved	
FRx_Full	[27]	Sets to 1 if FIFO-Rx buffer (FRx_WrBuf) is full. Resets to 0 on FRx_Reset = 1. Read only.	1'b0
FRx_Empty	[26]	Sets to 1 if FIFO-Rx buffer (FRx_WrBuf) is empty. Resets to 1 on FRx_Reset = 1. Read only.	1'b1
FRx_Done	[25]	Sets to 1 if FIFO-Rx has finished transferring FRx_MLen words of data to the destination. Resets to 0 on FRx_Reset = 1 or on read access to FRx_Ctrl register. Read only.	1'b0
FRx_Running	[24]	Sets to 1 if FIFO-Rx is transferring data to the destination or waiting for destination input buffer is ready. Sets to 1 if FRx_Start bit resets to 0. Resets to 0 on FRx_Reset = 1. Read only.	1'b0
FRx_Wd2Write	[23:16]	Number of words that is written to FIFO memory (FRx_WrBuf). Resets to 0 on FRx_Reset = 1. Read only.	8'h20
FRx_Wd2Read	[15:8]	Number of words that is read from FIFO memory (FRx_WrBuf). Resets to 0 on FRx_Reset = 1. Read only.	8'h00
FRx_Dest_Module	[7:6]	Destination module selection. (00 : AES, 01:DES/TDES, 10: SHA-1/PRNG, 11: Not Used) Resets to 0 on FRx_Reset = 1. Read/Write.	1'b0
FRx_Host_Rd_En	[5]	Enables Host read from FRx_Ctrl[31:16] and FRx_MLenCnt. Resets to 0 on FRx_Reset = 1. Read/Write.	1'b0
FRx_Host_Wr_En	[4]	Enables Host write to FRx_WrBuf. Resets to 0 on FRx_Reset = 1. Read/Write.	1'b0
FRx_Sync_Tx	[3]	If enabled, FIFO-Rx waits for FIFO-Tx to retrieve output data from source module before transferring data to the destination module. Resets to 0 on FRx_Reset = 1. Read/Write.	1'b0
FRx_Reset	[2]	Stops current FIFO-Rx transfer and resets FSM and all the register. Resets to 0 after reset. Read/Write.	1'b0
FRx_ERROR_En	[1]	Enables ERROR response via HRESP port when host tries to access FIFO-Rx and access is not enabled by FRx_Ctrl[4] or [5]. Resets to 0 on FRx_Reset = 1. Read/Write.	1'b0
FRx_Start	[0]	FIFO-Rx transfer start bit. Resets to 0 when internal FSM starts transferring data to destination. Resets to 0 on FRx_Reset = 1. Read/Write.	1'b0

4.3.2 FIFO-Rx Message Length Registers (FRx_MLen, R/W, Address = 0xF401_0004)

FRx_MLen	Bit	Description	Reset Value
FRx_MLen	[31:0]	Message Length in word (32-bit) unit. Resets to its reset value if FRx_Reset field of FRx_Ctrl register is set. Read/Write.	31'b0

4.3.3 FIFO-Rx Block Size Register (FRx_BlkSz, R/W, Address = 0xF401_0008)

FRx_BlkSz	Bit	Description	Reset Value
Reserved	[31:18]	Not used.	14'b0
FRx_LastValidByte	[17:16]	Last valid byte in the last word transferred to the SHA-1/PRNG module (Valid if FRx_Dest_Module=2'b10(in FRx_Ctrl). End of the SHA text byte 00 = First byte (LSB in 32-bit) 01 = Second byte 10 = Third byte 11 = Fourth byte (full 32-bit) Resets to its reset value if FRx_Reset field of FRx_Ctrl register is set. Read/Write.	2'b0
FRx_BlkSz	[15:0]	Block size of destination module (in word (32-bit) unit). FIFO-Rx transfers FRx_BlkSz word and then triggers the destination module to start processing. The destination module is selected by FRx_Dest_Module field in FRx_Ctrl. Resets to its reset value if FRx_Reset field of FRx_Ctrl register is set. Read/Write.	16'b0

4.3.4 FIFO-Rx Destination Address Register (FRx_DestAddr, R/W, Address = 0xF401_000C)

FRx_DestAddr	Bit	Description	Reset Value
FRx_DestAddr	[31:0]	Internal memory address of destination input buffer. FIFO-Rx transfers data to this address. Resets to its reset value if FRx_Reset field of FRx_Ctrl register is set. Read/Write.	31'b0

4.3.5 FIFO-Rx Message Length Counter (FRx_MLenCnt, R, Address = 0xF401_0010)

FRx_MLenCnt	Bit	Description	Reset Value
FRx_MLenCnt	[31:0]	Number of words left for transfer. Resets to 0 on FRx_Reset = 1. After counting down to 0, this register resets to the value in FRx_MLen. Read/Write.	31'b0

4.3.6 FIFO-Rx Write Buffer (FRx_WrBuf, W, Address = 0xF401_0040~0xF401_0080)

FRx_WrBuf	Bit	Description	Reset Value
FRx_WrBuf	[31:0]	Write access to FRx_WrBuf makes FIFO-Rx to write data to the FIFO memory regardless of the address given. That is, any address between 0x0040 and 0x0080 triggers the FIFO memory write. This feature lets the programmer use burst write to the FIFO-Rx.	

4.3.7 FIFO-Tx Control Register (FTx_Ctrl, R/W, Address = 0xF411_0000)

FTx_Ctrl	Bit	Description	Reset Value
FTx_WrPrivError	[31]	Sets to 1 if write access to FIFO-Tx has resulted in a privilege error (e.g. Host or the command are not allowed to access FIFO-Tx). Resets to 0 on FTx_Reset = 1. Read only.	1'b0
FTx_RdPrivError	[30]	Sets to 1 if read access to FIFO-Tx has resulted in a privilege error (e.g. Host or the command are not allowed to access FIFO-Tx). Resets to 0 on FTx_Reset = 1. Read only.	1'b0
Reserved	[29:28]	Reserved	
FTx_Full	[27]	Sets to 1 if FIFO-Tx buffer (FTx_RdBuf) is full. Resets to 0 on FTx_Reset = 1. Read only.	1'b0
FTx_Empty	[26]	Sets to 1 if FIFO-Tx buffer (FTx_RdBuf) is empty. Resets to 1 on FTx_Reset = 1. Read only.	1'b1
FTx_Done	[25]	Sets to 1 if FIFO-Tx has finished transferring FTx_MLen words of data from the source. Resets to 0 on FTx_Reset = 1 or on read access to FTx_Ctrl register. Read only.	1'b0
FTx_Running	[24]	Sets to 1 if FIFO-Tx is transferring data from the source or waiting for source output buffer is ready. Sets to 1 when FTx_Start bit resets to 0. Resets to 0 on FTx_Reset = 1. Read only.	1'b0
FTx_Wd2Read	[23:16]	Number of words that is read from FIFO memory (FTx_RdBuf). Resets to 0 on FTx_Reset = 1. Read only.	8'h00
FTx_Wd2Write	[15:8]	Number of words that is written to FIFO memory (FTx_RdBuf). Resets to 0 on FTx_Reset = 1. Read only.	8'h20
FTx_Src_Module	[7:6]	Source module selection. (00 : AES, 01:DES/TDES, 10: SHA-1/PRNG, 11: Not Used) Resets to 0 on FTx_Reset = 1. Read/Write.	1'b0
FTx_Host_Rd_En	[5]	Enables Host read from FTx_Ctrl[31:16] and FTx_MLenCnt. Resets to 0 on FTx_Reset = 1. Read/Write.	1'b0
FTx_Host_Wr_En	[4]	Enables Host read from FTx_RdBuf. Resets to 0 on FTx_Reset = 1. Read/Write.	1'b0
Reserved	[3]	Reserved	
FTx_Reset	[2]	Stops current FIFO-Tx transfer and resets FSM and all the register. Resets to 0 after reset. Read/Write.	1'b0
FTx_ERROR_En	[1]	Enables ERROR response via HRESP port if host tries to access FIFO-Tx and access is not enabled by FTx_Ctrl[4] or [5]. Resets	1'b0

		to 0 on FTx_Reset = 1. Read/Write.	
FTx_Start	[0]	FIFO-Tx transfer start bit. Resets to 0 if internal FSM starts transferring data to destination. Resets to 0 on FTx_Reset = 1. Read/Write.	1'b0

4.3.8 FIFO-Tx Message Length Register(FTx_MLen, R/W, Address = 0xF411_0004)

FTx_MLen	Bit	Description	Reset Value
FTx_MLen	[31:0]	Message Length in word (32-bit) unit. Resets to its reset value if FTx_Reset field of FTx_Ctrl register is set. Read/Write.	31'b0

4.3.9 FIFO-Tx Block Size Register (FTx_BlkSz, R/W, Address = 0xF411_0008)

FTx_BlkSz	Bit	Description	Reset Value
FTx_BlkSz	[31:0]	Block size of destination module (in word(32-bit) unit). FIFO-Tx transfers FTx_BlkSz word and then triggers the destination module to start processing. The destination module is selected by FTx_Dest_Module field in FTx_Ctrl. Resets to its reset value if FTx_Reset field of FTx_Ctrl register is set. Read/Write.	32'b0

4.3.10 FIFO-Tx Source Address Register (FTx_SrcAddr, R/W, Address = 0xF411_000C)

FTx_SrcAddr	Bit	Description	Reset Value
FTx_SrcAddr	[31:0]	Internal memory address of source output buffer. FIFO-Tx transfers data from this address. Resets to its reset value if FTx_Reset field of FTx_Ctrl register is set. Read/Write.	31'b0

4.3.11 FIFO-Tx Message Length Counter (FTx_MLenCnt, R/W, Address = 0xF411_0010)

FTx_MLenCnt	Bit	Description	Reset Value
FTx_MLenCnt	[31:0]	Number of words left for transfer. After counting down to 0, this register resets to the value in FTx_MLen.	31'b0

4.3.12 FIFO-Tx Read Buffer (FTx_RdBuf, W, Address = 0xF411_0040~0xF411_0080)

FTx_RdBuf	Bit	Description	Reset Value
FTx_RdBuf	[31:0]	Read access to FTx_RdBuf makes FIFO-Tx to read data from the FIFO memory regardless of the address given. That is, any address between 0x0040 and 0x0080 triggers the FIFO memory read. This feature lets the programmer use burst read from FIFO-Tx	

4.4 AES SPECIAL FUNCTION REGISTER (SFRS)

4.4.1 AES Control / Status Register (AES_Rx_CTRL, R/W, Address = 0xF404_0000)

AES_Rx_CTRL	Bit	Description	Reset Value
WrPrivMismatch	[31]	SFR Write Access Privilege Mismatch Status bit. If set to '1', SFR Write Access Privilege Mismatch occurs. Read only (Read Clear)	1'b0
RdPrivMismatch	[30]	SFR Read Access Privilege Mismatch Status bit. If set to '1', SFR Read Access Privilege Mismatchoccurs. Read only (Read Clear)	1'b0
Reserved	[29:11]	Reserved	19'h0
AesOutReady	[10]	If set to '1', AES Output Buffer is Full, and ARM or Rx FiFo is permitted to Read current 128bits result data Read only	1'b0
AesInReady	[9]	If set to '1', AES Input Buffer is Empty, and ARM or Rx FiFo is permitted to write next 128bits data. Read only	1'b1
AesContDecOn	[8]	Continuous Decryption Enable Bits 0 = Decryption Key is changed 1 = Decryption Key is not changed Read/Write	1'b0
CtrWidth	[7:6]	Counter Mode Counter Width Bits 00 = 16Bits Counter 01 = 32Bits Counter 10 = 64Bits Counter Read/Write	2'b00
AesOpMode	[5:4]	AES Operation Mode Selection Bits 01 = ECB Mode 10 = CBC Mode 11 = CTR Mode Read/Write	2'b0
AesOpDirection	[3]	AES Operation Direction Selection Bit. 0 = Encryption	1'b0

		1 = Decryption Read/Write	
AesKeyMode	[2:1]	AES Key Mode Selection Bits. 00 = 128bits 01 = 192bits 10 = 256bits Read/Write	2'b00
AesOpEnable	[0]	If set to '1', AES starts operation by ARM. If the aes_op_done is generated, It becomes '0'. Read/Write	1'b0

4.4.2 AES Data Input Register 01 (Least Significant) (AES_Rx_DIN_01, R/W, Address = 0xF404_0010)

AES_Rx_DIN_01	Bit	Description	Reset Value
AesDin01	[31:0]	AES 1_{st} 32-bit Data Input Register	32'h0

4.4.3 AES Data Input Register 02 (AES_Rx_DIN_02, R/W, Address = 0xF404_0014)

AES_Rx_DIN_02	Bit	Description	Reset Value
AesDin02	[31:0]	AES 2_{nd} 32-bit Data Input Register	32'h0

4.4.4 AES Data Input Register 03 (AES_Rx_DIN_03, R/W, Address = 0xF404_0018)

AES_Rx_DIN_03	Bit	Description	Reset Value
AesDin03	[31:0]	AES 3_{rd} 32bit Data Input Register	32'h0

4.4.5 AES Data Input Register 04 (AES_Rx_DIN_04, R/W, Address = 0xF404_001C)

AES_Rx_DIN_04	Bit	Description	Reset Value
AesDin04	[31:0]	AES 4_{th} 32-bit Data Input Register	32'h0

4.4.6 AES Data Output Register 01 (Least Significant) (AES_Rx_DOUT_01, R, Address = 0xF404_0020)

AES_Rx_DOUT_01	Bit	Description	Reset Value
AesDout01	[31:0]	AES 1_{st} 32-bit Data Output Register	32'h0

4.4.7 AES Data Output Register 01 (Least Significant) (AES_Tx_DOUT_01, R, Address = 0xF414_0020)

AES_Tx_DOUT_01	Bit	Description	Reset Value
AesDout01	[31:0]	AES 1_{st} 32-bit Data Output Register	32'h0

4.4.8 AES Data Output Register 02 (AES_Rx_DOUT_02, R, Address = 0xF404_0024)

AES_Rx_DOUT_02	Bit	Description	Reset Value
AesDout02	[31:0]	AES 2_{nd} 32-bit Data Output Register	32'h0

4.4.9 AES Data Output Register 02 (AES_Tx_DOUT_02, R, Address = 0xF414_0024)

AES_Tx_DOUT_02	Bit	Description	Reset Value
AesDout02	[31:0]	AES 2_{nd} 32-bit Data Output Register	32'h0

4.4.10 AES Data Output Register 03 (AES_Rx_DOUT_03, R, Address = 0xF404_0028)

AES_Rx_DOUT_03	Bit	Description	Reset Value
AesDout03	[31:0]	AES 3_{rd} 32-bit Data Output Register	32'h0

4.4.11 AES Data Output Register 03 (AES_Tx_DOUT_03, R, Address = 0xF414_0028)

AES_Tx_DOUT_03	Bit	Description	Reset Value
AesDout03	[31:0]	AES 3_{rd} 32-bit Data Output Register	32'h0

4.4.12 AES Data Output Register 04 (AES_Rx_DOUT_04, R, Address = 0xF404_002C)

AES_Rx_DOUT_04	Bit	Description	Reset Value
AesDout04	[31:0]	AES 4_{th} 32-bit Data Output Register	32'h0

4.4.13 AES Data Output Register 04 (AES_Tx_DOUT_04, R, Address = 0xF414_002C)

AES_Tx_DOUT_04	Bit	Description	Reset Value
AesDout04	[31:0]	AES 4_{th} 32-bit Data Output Register	32'h0

4.4.14 AES Key Input Register 01 (Least Significant) (AES_Rx_KEY_01, R/W, Address = 0xF404_0080)

AES_Rx_KEY_01	Bit	Description	Reset Value
AesKey01	[31:0]	AES 1_{st} 32-bit Key Input Register	32'h0

4.4.15 AES Key Input Register 02 (AES_Rx_KEY_02, R/W, Address = 0xF404_0084)

AES_Rx_KEY_02	Bit	Description	Reset Value
AesKey02	[31:0]	AES 2_{nd} 32-bit Key Input Register	32'h0

4.4.16 AES Key Input Register 03 (AES_Rx_KEY_03, R/W, Address = 0xF404_0088)

AES_Rx_KEY_03	Bit	Description	Reset Value
AesKey03	[31:0]	AES 3_{rd} 32-bit Key Input Register	32'h0

4.4.17 AES Key Input Register 04 (AES_Rx_KEY_04, R/W, Address = 0xF404_008c)

AES_Rx_KEY_04	Bit	Description	Reset Value
AesKey04	[31:0]	AES 4_{th} 32-bit Key Input Register	32'h0

4.4.18 AES Key Input Register 05 (AES_Rx_KEY_05, R/W, Address = 0xF404_0090)

AES_Rx_KEY_05	Bit	Description	Reset Value
AesKey05	[31:0]	AES 5_{th} 32-bit Key Input Register	32'h0

4.4.19 AES Key Input Register 06 (AES_Rx_KEY_06, R/W, Address = 0xF404_0094)

AES_Rx_KEY_06	Bit	Description	Reset Value
AesKey06	[31:0]	AES 6_{th} 32-bit Key Input Register	32'h0

4.4.20 AES Key Input Register 07(AES_Rx_KEY_07, R/W, Address = 0xF404_0098)

AES_Rx_KEY_07	Bit	Description	Reset Value
AesKey07	[31:0]	AES 7_{th} 32-bit Key Input Register	32'h0

4.4.21 AES Key Input Register 08 (AES_Rx_KEY_08, R/W, Address = 0xF404_009c)

AES_Rx_KEY_08	Bit	Description	Reset Value
AesKey08	[31:0]	AES 8_{th} 32-bit Key Input Register	32'h0

4.4.22 AES IV Input Register 01 (AES_Rx_IV_01, R/W, Address = 0xF404_00a0)

AES_Rx_IV_01	Bit	Description	Reset Value
AesIv01	[31:0]	AES 1_{st} 32-bit IV Input Register	32'h0

4.4.23 AES IV Input Register 02 (AES_Rx_IV_02, R/W, Address = 0xF404_00a4)

AES_Rx_IV_02	Bit	Description	Reset Value
AesIv02	[31:0]	AES 2_{nd} 32-bit IV Input Register	32'h0

4.4.24 AES IV Input Register 03 (AES_Rx_IV_03, R/W, Address = 0xF404_00a8)

AES_Rx_IV_03	Bit	Description	Reset Value
AesIv03	[31:0]	AES 3_{rd} 32-bit IV Input Register	32'h0

4.4.25 AES IV Input Register 04 (AES_Rx_IV_04, R/W, Address = 0xF404_00ac)

AES_Rx_IV_04	Bit	Description	Reset Value
AesIv04	[31:0]	AES 4_{th} 32-bit IV Input Register	32'h0

4.4.26 AES Counter Preload Input Register 01 (AES_Rx_CTR_01, R/W, Address = 0xF404_00b0)

AES_Rx_CTR_01	Bit	Description	Reset Value
AesCtr01	[31:0]	AES 1_{st} 32-bit Counter Preload Input Register	32'h0

4.4.27 AES Counter Preload Input Register 02 (AES_Rx_CTR_02, R/W, Address = 0xF404_00b4)

AES_Rx_CTR_02	Bit	Description	Reset Value
AesCtr02	[31:0]	AES 2_{nd} 32-bit Counter Preload Input Register	32'h0

4.4.28 AES Counter Preload Input Register 03 (AES_Rx_CTR_03, R/W, Address = 0xF404_00b8)

AES_Rx_CTR_03	Bit	Description	Reset Value
AesCtr03	[31:0]	AES 3_{rd} 32-bit Counter Preload Input Register	32'h0

4.4.29 AES Counter Preload Input Register 04 (AES_Rx_CTR_04, R/W, Address = 0xF404_00bc)

AES_Rx_CTR_04	Bit	Description	Reset Value
AesCtr04	[31:0]	AES 4_{th} 32-bit Counter Preload Input Register	32'h0

4.5 TDES SPECIAL FUNCTION REGISTERS (SFRS)

4.5.1 TDES control / status register (TDES_Rx_CTRL, R/W, Address = 0xF405_0000)

TDES_Rx_CTRL	Bit	Description	Reset Value
WrPrivMismatch	[31]	SFR Write Access Privilege Mismatch Status bit. If set to '1', SFR Write Access Privilege Mismatch occurs. Read only (Read Clear)	1'b0
RdPrivMismatch	[30]	SFR Read Access Privilege Mismatch Status bit. If set to '1', SFR Read Access Privilege Mismatch is occurs. Read only (Read Clear)	1'b0
Reserved	[29:8]	Reserved	22'h0
TdesOutReady	[7]	If set to '1', AES Out Buffer is Full, and ARM or Rx FiFo is permitted to Read current 128-bits result data Read only	1'b0
TdesInReady	[6]	If set to '1', TDES Input Buffer is Empty, and ARM or Rx FiFo is permitted to write next 128-bits data. Read only	1'h1
DesOrTdes	[5]	DES or TDES Operation Selection Bit 0 = DES Only Mode 1 = TDES Mode Read/Write	1'b0
TdesMode	[4:3]	TDES Mode Selection Bit 01 = ECB Mode 10 = CBC Mode Read/Write	2'b00
TdesOpDirection	[2]	TDES Operation Direction Selection Bit. 0 = Encryption 1 = Decryption Read/Write	1'b0
TdesIntMode	[1]	TDES Operation End Mode Selection Bit 0 = Polling Mode 1 = Interrupt Mode Read/Write	1'b0
TdesOpEnable	[0]	If set to '1', TDES starts operation by ARM. If the des_or_tdes_op_done is generated, It becomes '0'. Read/Write	1'b0

4.5.2 TDES Key Input Register 1_0 (Least Significant) (TDES_Rx_KEY1_0, R/W, Address = 0xF405_0010)

TDES_Rx_KEY1_0	Bit	Description	Reset Value
TdesKey1_0	[31:0]	TDES 1_{st} 32-bit Key Input Register	32'h0

4.5.3 TDES Key Input Register 1_1 (TDES_Rx_KEY1_1, R/W, Address = 0xF405_0014)

TDES_Rx_KEY1_1	Bit	Description	Reset Value
TdesKey1_1	[31:0]	TDES 2_{nd} 32-bit Key Input Register	32'h0

NOTE: If DES Only Mode, TDES_#_KEY1_0 and TDES_#_KEY1_1 register is only used for DES operation.

4.5.4 TDES Key Input Register 2_0 (TDES_Rx_KEY2_0, R/W, Address = 0xF405_0018)

TDES_Rx_KEY2_0	Bit	Description	Reset Value
TdesKey2_0	[31:0]	TDES 3_{rd} 32-bit Key Input Register	32'h0

4.5.5 TDES Key Input Register 2_1 (TDES_Rx_KEY2_1, R/W, Address = 0xF405_001c)

TDES_Rx_KEY2_1	Bit	Description	Reset Value
TdesKey2_1	[31:0]	TDES 4_{th} 32-bit Key Input Register	32'h0

4.5.6 TDES Key Input Register 3_0 (TDES_Rx_KEY3_0, R/W, Address = 0xF405_0020)

TDES_Rx_KEY3_0	Bit	Description	Reset Value
TdesKey3_0	[31:0]	TDES 5_{th} 32-bit Key Input Register	32'h0

4.5.7 TDES Key Input Register 3_1 (TDES_Rx_KEY3_1, R/W, Address = 0xF405_0024)

TDES_Rx_KEY3_1	Bit	Description	Reset Value
TdesKey3_1	[31:0]	TDES 6_{th} 32-bit Key Input Register	32'h0

4.5.8 TDES Data Input Register 0 (Least Significant) (TDES_Rx_INPUT_0, R/W, Address = 0xF405_0040)

TDES_Rx_INPUT_0	Bit	Description	Reset Value
TdesDin0	[31:0]	TDES 1_{st} 32-bit Data Input Register	32'h0

4.5.9 TDES Data Input Register 0 (Most Significant) (TDES_Rx_INPUT_1, R/W, Address = 0xF405_0044)

TDES_Rx_INPUT_1	Bit	Description	Reset Value
TdesDin1	[31:0]	TDES 2_{nd} 32-bit Data Input Register	32'h0

4.5.10 TDES Data Output Register 0 (Least Significant) (TES_Rx_OUTPUT_0, R, Address = 0xF405_0048)

TDES_Rx_OUTPUT_0	Bit	Description	Reset Value
TdesDout0	[31:0]	TDES 1_{st} 32-bit Data Output Register	32'h0

4.5.11 TDES Data Output Register 0 (Least Significant) (TES_Tx_OUTPUT_0, R, Address = 0xF415_0048)

TDES_Tx_OUTPUT_0	Bit	Description	Reset Value
TdesDout0	[31:0]	TDES 1_{st} 32-bit Data Output Register	32'h0

4.5.12 TDES Data Output Register 1 (Most Significant) (TDES_Rx_OUTPUT_1, R, Address = 0xF405_004C)

TDES_Rx_OUTPUT_1	Bit	Description	Reset Value
TdesDout1	[31:0]	TDES 2_{nd} 32-bit Data Output Register	32'h0

4.5.13 TDES Data Output Register 1 (Most Significant) (TDES_Tx_OUTPUT_1, R, Address = 0xF415_004C)

TDES_Tx_OUTPUT_1	Bit	Description	Reset Value
TdesDout1	[31:0]	TDES 2_{nd} 32-bit Data Output Register	32'h0

4.5.14 TDES IV Input Register 0 (TDES_Rx_IV_0, R/W, Address = 0xF405_0050)

TDES_Rx_IV_0	Bit	Description	Reset Value
TdesIv0	[31:0]	TDES 1_{st} 32-bit IV Input Register	32'h0

4.5.15 TDES IV Input Register 1 (TDES_Rx_IV_1, R/W, Address = 0xF405_0054)

TDES_Rx_IV_1	Bit	Description	Reset Value
TdesIv1	[31:0]	TDES 2_{nd} 32-bit IV Input Register	32'h0

4.6 SHA-1/PRNG SPECIAL FUNCTION REGISTERS (SFRS)

4.6.1 Hash Engine Control Register (HASH_CONTROL, R/W, Address = 0xF406_0000)

HASH_CONTROL	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	
USE_IV	[8]	Use arbitrary IV instead of SHA-1 constants 1= Arbitrary IV, 0= Constants Write/Auto-reset	1'b0
End_of_Hash_byte	[7:6]	End of the SHA text byte 00 = First byte (LSB in 32-bit) 01 = Second byte 10 = Third byte 11 = Fourth byte (full 32-bit) Write/Auto-reset	2'b00
SEED_SETTING_ENABLE	[5]	Seed setting enable (1'b1) Read/Write	1'b0
Hash_input_finished	[4]	Finished the hash input (Cleared by hardware) Write/Auto-reset	1'b0
Hash_start	[3]	Start the hash (software reset) (Automatically cleared by hardware) Write/Auto-reset	1'b0
Data_Selection	[2]	Indicates whether the next data of register values are keys or text 1 = Key, 0 = Text Read/Write	1'b0
Engine_Selection	[1:0]	To use as SHA-1 or HMAC or PRNG 00 = HMAC, 01 =SHA1, 10 = PRNG Read/Write	2'b00

4.6.2 Hash Data (HASH_DATA, W, Address = 0xF406_0004, 0xF406_0020)

HASH_DATA	Bit	Description	Reset Value
HASH_DATA	[31:0]	Hash data input register (HASH_CONTROL[5] == 1'b0)	0x0000_0000

4.6.3 PRNG Seed Data (01) (SEED_DATA_01, W, Address = 0xF406_0008)

SEED_DATA_01	Bit	Description	Reset Value
SEED_DATA_01	[31:0]	PRNG seed data [31:0] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.4 PRNG Seed Data (02) (SEED_DATA_02, W, Address = 0xF406_000c)

SEED_DATA_02	Bit	Description	Reset Value
SEED_DATA_02	[31:0]	PRNG seed data [63:32] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.5 PRNG Seed Data (03) (SEED_DATA_03, W, Address = 0xF406_0010)

SEED_DATA_03	Bit	Description	Reset Value
SEED_DATA_03	[31:0]	PRNG seed data [95:64] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.6 PRNG Seed Data (04) (SEED_DATA_04, W, Address = 0xF406_0014)

SEED_DATA_04	Bit	Description	Reset Value
SEED_DATA_04	[31:0]	PRNG seed data [127:96] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.7 PRNG Seed Data (05) (SEED_DATA_05, W, Address = 0xF406_0018)

SEED_DATA_05	Bit	Description	Reset Value
SEED_DATA_05	[31:0]	PRNG seed data [159:128] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.8 PRNG Seed Data (06) (SEED_DATA_06, W, Address = 0xF406_001c)

SEED_DATA_06	Bit	Description	Reset Value
SEED_DATA_06	[31:0]	PRNG seed data [191:160] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.9 PRNG Seed Data (07) (SEED_DATA_07, W, Address = 0xF406_0020)

SEED_DATA_07	Bit	Description	Reset Value
SEED_DATA_07	[31:0]	PRNG seed data [223:192] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.10 PRNG Seed Data (08) (SEED_DATA_08, W, Address = 0xF406_0024)

SEED_DATA_08	Bit	Description	Reset Value
SEED_DATA_08	[31:0]	PRNG seed data [255:224] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.11 PRNG Seed Data (09) (SEED_DATA_09, W, Address = 0xF406_0028)

SEED_DATA_09	Bit	Description	Reset Value
SEED_DATA_09	[31:0]	PRNG seed data [287:256] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.12 PRNG Seed Data (10) (SEED_DATA_10, W, Address = 0xF406_002c)

SEED_DATA_10	Bit	Description	Reset Value
SEED_DATA_10	[31:0]	PRNG seed data [319:288] (HASH_CONTROL[5] == 1'b1)	0x0000_0000

4.6.13 Hash status (HASH_STATUS, R, Address = 0xF406_0030)

HASH_STATUS	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	
BUFFER_IN_ENABLE	[4]	1 = Enables Buffer Input (Buffer is empty) 0 = Buffer Input Not Enable (Buffer is full) Read	1'b1
HASH_engine_ready	[3]	Ready to receive next 64 bytes of input data (Cleared by software after checking) Read/Reset-on-read	1'b0
Random_Number_Ready	[2]	Random number ready (Cleared by hardware after '1' is read) Read/Reset-on-reset	1'b0
32bit_ready	[1]	Used only in test Next 32-bit is ready to be processed (Cleared by hardware after '1' is read) Read/Reset-on-read	1'b0
HASH_output_ready	[0]	160 bits of the hash calculation is finished and ready to be used (Cleared by hardware after '1' is read) Read/Reset-on-read	1'b0

4.6.14 Hash Output (01) or PRNG Output [31:1] (HASH_OUTPUT_01/PRNG_OUTPUT_01, R, Address = 0xF406_0034)

HASH_OUTPUT_01/ PRNG_OUTPUT_01	Bit	Description	Reset Value
HASH_OUTPUT_01	[31:0]	PRNG_output[31:0] if Engine_selection[1:0] == 2'b10 Else Hash_output[31:0]	0x0000_0000

4.6.15 Hash Output (02) or PRNG Output [63:32] (HASH_OUTPUT_02/PRNG_OUTPUT_02, R, Address = 0xF406_0038)

HASH_OUTPUT_02/ PRNG_OUTPUT_02	Bit	Description	Reset Value
HASH_OUTPUT_02	[31:0]	PRNG_output[63:32] if Engine_selection[1:0] == 2'b10 Else Hash_output[63:32]	0x0000_0000

4.6.16 Hash Output (03) or PRNG Output [95:64] (HASH_OUTPUT_03/PRNG_OUTPUT_03, R, Address = 0xF406_003C)

HASH_OUTPUT_03/ PRNG_OUTPUT_03	Bit	Description	Reset Value
HASH_OUTPUT_03	[31:0]	PRNG_output[95:64] if Engine_selection[1:0] == 2'b10 Else Hash_output[95:64]	0x0000_0000

4.6.17 Hash Output (04) or PRNG Output [127:96] (HASH_OUTPUT_04/PRNG_OUTPUT_04, R, Address = 0xF406_0040)

HASH_OUTPUT_04/ PRNG_OUTPUT_04	Bit	Description	Reset Value
HASH_OUTPUT_04	[31:0]	PRNG_output[127:96] if Engine_selection[1:0] == 2'b10 Else Hash_output[127:96]	0x0000_0000

4.6.18 Hash Output (05) or PRNG Output [159:128] (HASH_OUTPUT_05/PRNG_OUTPUT_05, R, Address = 0xF406_0044)

HASH_OUTPUT_05/ PRNG_OUTPUT_05	Bit	Description	Reset Value
HASH_OUTPUT_05	[31:0]	PRNG_output[159:128] if Engine_selection[1:0] == 2'b10 Else Hash_output[159:128]	0x0000_0000

4.6.19 PRNG_output [191:160] (PRNG_OUTPUT_06, R, Address = 0xF406_0048)

PRNG_OUTPUT_06	Bit	Description	Reset Value
PRNG_OUTPUT_06	[31:0]	PRNG_output[191:160]	0x0000_0000

4.6.20 PRNG_output [223:192] (PRNG_OUTPUT_07, R, Address = 0xF406_004c)

PRNG_OUTPUT_07	Bit	Description	Reset Value
PRNG_OUTPUT_07	[31:0]	PRNG_output[223:192]	0x0000_0000

4.6.21 PRNG_output [255:224] (PRNG_OUTPUT_08, R, Address = 0xF406_0050)

PRNG_OUTPUT_08	Bit	Description	Reset Value
PRNG_OUTPUT_08	[31:0]	PRNG_output[255:224]	0x0000_0000

4.6.22 PRNG_output [287:256] (PRNG_OUTPUT_09, R, Address = 0xF406_0054)

PRNG_OUTPUT_09	Bit	Description	Reset Value
PRNG_OUTPUT_09	[31:0]	PRNG_output[287:256]	0x0000_0000

4.6.23 PRNG_output [319:288] (PRNG_OUTPUT_10, R, Address = 0xF406_0058)

PRNG_OUTPUT_10	Bit	Description	Reset Value
PRNG_OUTPUT_10	[31:0]	PRNG_output[319:288]	0x0000_0000

4.6.24 HASH_MIDOUT [159:128] (HASH_MIDOUT_01, R, Address = 0xF406_005c)

HASH_MIDOUT_01	Bit	Description	Reset Value
HASH_MIDOUT_01	[31:0]	HASH_MIDOUT [159:128]	0x0000_0000

4.6.25 HASH_MIDOUT [127:96] (HASH_MIDOUT_02, R, Address = 0xF406_0060)

HASH_MIDOUT_02	Bit	Description	Reset Value
HASH_MIDOUT_02	[31:0]	HASH_MIDOUT [127:96]	0x0000_0000

4.6.26 HASH_MIDOUT [95:64] (HASH_MIDOUT_03, R, Address = 0xF406_0064)

HASH_MIDOUT_03	Bit	Description	Reset Value
HASH_MIDOUT_03	[31:0]	HASH_MIDOUT [95:64]	0x0000_0000

4.6.27 HASH_MIDOUT [63:32] (HASH_MIDOUT_04, R, Address = 0xF406_0068)

HASH_MIDOUT_04	Bit	Description	Reset Value
HASH_MIDOUT_04	[31:0]	HASH_MIDOUT [63:32]	0x0000_0000

4.6.28 HASH_MIDOUT [31:0] (HASH_MIDOUT_05, R, Address = 0xF406_006c)

HASH_MIDOUT_05	Bit	Description	Reset Value
HASH_MIDOUT_05	[31:0]	HASH_MIDOUT [31:0]	0x0000_0000

4.6.29 HASH_IV [159:128] (HASH_IV_01, W, Address = 0xF406_0070)

HASH_IV_01	Bit	Description	Reset Value
HASH_IV_01	[31:0]	HASH_IV [159:128]	0x0000_0000

4.6.30 HASH_IV [127:96] (HASH_IV_02, W, Address = 0xF406_0074)

HASH_IV_02	Bit	Description	Reset Value
HASH_IV_02	[31:0]	HASH_IV [127:96]	0x0000_0000

4.6.31 HASH_IV [95:64] (HASH_IV_03, W, Address = 0xF406_0078)

HASH_IV_03	Bit	Description	Reset Value
HASH_IV_03	[31:0]	HASH_IV [95:64]	0x0000_0000

4.6.32 HASH_IV [63:32] (HASH_IV_04, W, Address = 0xF406_007c)

HASH_IV_04	Bit	Description	Reset Value
HASH_IV_04	[31:0]	HASH_IV [63:32]	0x0000_0000

4.6.33 HASH_IV [31:0] (HASH_IV_05, W, Address = 0xF406_0080)

HASH_IV_05	Bit	Description	Reset Value
HASH_IV_05	[31:0]	HASH_IV [31:0]	0x0000_0000

4.6.34 PRE_MSG_LENGTH [63:32] (PRE_MSG_LENGTH_01, W, Address = 0xF406_0084)

PRE_MSG_LENGTH_01	Bit	Description	Reset Value
PRE_MSG_LENGTH_01	[31:0]	PRE_MSG_LENGTH [63:32]	0x0000_0000

4.6.35 PRE_MSG_LENGTH [31:0] (PRE_MSG_LENGTH_02, W, Address = 0xF406_0088)

PRE_MSG_LENGTH_02	Bit	Description	Reset Value
PRE_MSG_LENGTH_02	[31:0]	PRE_MSG_LENGTH [31:0]	0x0000_0000

4.7 PKA SPECIAL FUNCTION REGISTERS (SFRS)

4.7.1 PKA Special Function Register 0 (PKA_SFR0, R/W, Address = 0xF407_1000)

PKA_SFR0	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	25'b0
CHNK_SZ	[6:3]	Chunk Size Bits 0000 = (Do not use) [*Default*] 0001 = (Do not use) 0010 = (Do not use) 0011 = 128-bits 0100 = 160-bits 0101 = 192-bits 0110 = 224-bits 0111 = 256-bits 1000 = 288-bits 1001 = 320-bits 1010 = 352-bits 1011 = 384-bits 1100 = 416-bits 1101 = 448-bits 1110 = 480-bits 1111 = 512-bits	4'b0
Reserved	[2]	Reserved	1'b0
PREC_ID	[1:0]	Precision Bits 00 = Single Precision [*Default*] 01 = Double Precision 10 = Triple Precision 11 = Quadruple Precision	2'b0

4.7.2 PKA Special Function Register 1 (PKA_SFR1, R/W, Address = 0xF407_1004)

PKA_SFR1	Bit	Description	Reset Value
Reserved	[31:4]	Reserved	28'b0
PLDM_ON	[3]	Control the Pre-loading of the Least Significant Chunk of Modulus M bit 0 = Do not pre-load the least significant chunk of modulus M [*Default*] 1 = Pre-load the least significant chunk of modulus M	1'b0
Reserved	[2:1]	Reserved	2'b0
EXEC_ON	[0]	Control and Monitor the execution of PKA bit 0 = Do not run execution [*Default*] 1 = Run Execution	1'b0

4.7.3 PKA Special Function Register 2 (PKA_SFR2, R/W, Address = 0xF407_1008)

PKA_SFR2	Bit	Description	Reset Value
Reserved	[31:29]	Reserved	3'b0
A_SEG_ID	[28:24]	Multiplicand A Memory Segment ID Bits 00000 = Segment 0 [*Default*] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26	5'b0

PKA_SFR2	Bit	Description	Reset Value
		11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (Do not use) 11111 = (Do not use)	
Reserved	[23:21]	Reserved	3'b0
B_SEG_ID	[20:16]	Multiplier B Memory Segment ID Bits 00000 = Segment 0 [*Default*] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (Do not use) 11111 = (Do not use)	5'b0
Reserved	[15:13]	Reserved	3'b0
M_SEG_ID	[12:8]	Modulus M Memory Segment ID Bits 00000 = Segment 0 [*Default*] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6	5'b0

PKA_SFR2	Bit	Description	Reset Value
		00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20 10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (Do not use) 11111 = (Do not use)	
Reserved	[7:5]	Reserved	3'b0
S_SEG_ID	[4:0]	Result S Memory Segment ID Bits 00000 = Segment 0 [*Default*] 00001 = Segment 1 00010 = Segment 2 00011 = Segment 3 00100 = Segment 4 00101 = Segment 5 00110 = Segment 6 00111 = Segment 7 01000 = Segment 8 01001 = Segment 9 01010 = Segment 10 01011 = Segment 11 01100 = Segment 12 01101 = Segment 13 01110 = Segment 14 01111 = Segment 15 10000 = Segment 16 10001 = Segment 17 10010 = Segment 18 10011 = Segment 19 10100 = Segment 20	5'b0

PKA_SFR2	Bit	Description	Reset Value
		10101 = Segment 21 10110 = Segment 22 10111 = Segment 23 11000 = Segment 24 11001 = Segment 25 11010 = Segment 26 11011 = Segment 27 11100 = Segment 28 11101 = Segment 29 11110 = (Do not use) 11111 = (Do not use)	

4.7.4 PKA Special Function Register 3 (PKA_SFR3, R/W, Address = 0xF407_100C)

PKA_SFR3	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	2'b0
S_SEG_ID	[29:0]	Segment Sign Bits xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx**0**: Segment 0 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx**1**: Segment 0 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xx**0**x: Segment 1 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xx**1**x: Segment 1 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x**0**xx: Segment 2 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_x**1**xx: Segment 2 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_**0**xxx: Segment 3 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_**1**xxx: Segment 3 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx**0**_xxxx: Segment 4 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxx**1**_xxxx: Segment 4 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_xx**0**x_xxxx: Segment 5 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_xx**1**x_xxxx: Segment 5 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_x**0**xx_xxxx: Segment 6 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_x**1**xx_xxxx: Segment 6 is negative xx_xxxx_xxxx_xxxx_xxxx_xxxx_**0**xxx_xxxx: Segment 7 is positive xx_xxxx_xxxx_xxxx_xxxx_xxxx_**1**xxx_xxxx: Segment 7 is negative xx_xxxx_xxxx_xxxx_xxxx_xxx**0**_xxxx_xxxx: Segment 8 is positive xx_xxxx_xxxx_xxxx_xxxx_xxx**1**_xxxx_xxxx: Segment 8 is negative xx_xxxx_xxxx_xxxx_xxxx_xx**0**x_xxxx_xxxx: Segment 9 is positive xx_xxxx_xxxx_xxxx_xxxx_xx**1**x_xxxx_xxxx: Segment 9 is negative xx_xxxx_xxxx_xxxx_xxxx_x**0**xx_xxxx_xxxx: Segment 10 is positive xx_xxxx_xxxx_xxxx_xxxx_x**1**xx_xxxx_xxxx: Segment 10 is negative xx_xxxx_xxxx_xxxx_xxxx_**0**xxx_xxxx_xxxx: Segment 11 is positive xx_xxxx_xxxx_xxxx_xxxx_**1**xxx_xxxx_xxxx: Segment 11 is negative xx_xxxx_xxxx_xxxx_xxx**0**_xxxx_xxxx_xxxx: Segment 12 is positive xx_xxxx_xxxx_xxxx_xxx**1**_xxxx_xxxx_xxxx: Segment 12 is negative xx_xxxx_xxxx_xxxx_xx**0**x_xxxx_xxxx_xxxx: Segment 13 is positive xx_xxxx_xxxx_xxxx_xx**1**x_xxxx_xxxx_xxxx: Segment 13 is negative xx_xxxx_xxxx_xxxx_x**0**xx_xxxx_xxxx_xxxx: Segment 14 is positive xx_xxxx_xxxx_xxxx_x**1**xx_xxxx_xxxx_xxxx: Segment 14 is negative xx_xxxx_xxxx_xxxx_**0**xxx_xxxx_xxxx_xxxx: Segment 15 is positive xx_xxxx_xxxx_xxxx_**1**xxx_xxxx_xxxx_xxxx: Segment 15 is negative xx_xxxx_xxxx_xxx**0**_xxxx_xxxx_xxxx_xxxx: Segment 16 is positive xx_xxxx_xxxx_xxx**1**_xxxx_xxxx_xxxx_xxxx: Segment 16 is negative xx_xxxx_xxxx_xx**0**x_xxxx_xxxx_xxxx_xxxx: Segment 17 is positive xx_xxxx_xxxx_xx**1**x_xxxx_xxxx_xxxx_xxxx: Segment 17 is negative xx_xxxx_xxxx_x**0**xx_xxxx_xxxx_xxxx_xxxx: Segment 18 is positive xx_xxxx_xxxx_x**1**xx_xxxx_xxxx_xxxx_xxxx: Segment 18 is negative xx_xxxx_xxxx_**0**xxx_xxxx_xxxx_xxxx_xxxx: Segment 19 is positive xx_xxxx_xxxx_**1**xxx_xxxx_xxxx_xxxx_xxxx: Segment 19 is negative xx_xxxx_xxx**0**_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 20 is positive xx_xxxx_xxx**1**_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 20 is negative xx_xxxx_xx**0**x_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 21 is positive xx_xxxx_xx**1**x_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 21 is negative xx_xxxx_x**0**xx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 22 is positive xx_xxxx_x**1**xx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 22 is negative xx_xxxx_**0**xxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 23 is positive xx_xxxx_**1**xxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 23 is negative xx_xxx**0**_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 24 is positive xx_xxx**1**_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 24 is negative xx_xx**0**x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 25 is positive xx_xx**1**x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 25 is negative xx_x**0**xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 26 is positive xx_x**1**xx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 26 is negative xx_**0**xxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 27 is positive xx_1xxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 27 is negative x**0**_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 28 is positive x**1**_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 28 is negative **0**x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 29 is positive **1**x_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx: Segment 29 is negative	30'b0

4.7.5 PKA Special Function Register 4 (PKA_SFR4, R/W, Address = 0xF407_1010)

PKA_SFR4	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	25'b0
SEG_SIZE	[6:5]	Segment Size Bits 00 = 256 bytes (= 2048 bits) [*Default*] 01 = 128 bytes (= 1024 bits) 10 = 64 bytes (= 512 bits)	2'b0
Reserved	[4:1]	Reserved	4'b0
FUNC_ID	[0]	Montgomery Product Function ID 0 = Montgomery Multiplication (A by B) [*Default*] 1 = Montgomery Multiplication (A by 1)	1'b0

4.8 SECURE JTAG SPECIAL FUNCTION REGISTERS (SFRS)

4.8.1 Secure JTAG H/W Version Register (SJ_HW_VERSION_REG, R, Address = 0xF408_0000)

SJ_HW_VERSION_REG	Bit	Description	Reset Value
SJ_HW_VERSION	[31:0]	Secure JTAG H/W version	32'h20070727

4.8.2 Secure JTAG status register (SJ_STATUS_REG, R, Address = 0xF408_0004)

SJ_STATUS_REG	Bit	Description	Reset Value
Reserved	[31:15]	Reserved	17'b0
SJ_FRAME_CNT	[14:8]	Secure JTAG Frame Counter	7'b0
Reserved	[7:2]	Reserved	6'b0
SJ_HASH_BUSY	[1]	External Hash Engine Status Register 1 = Hash Engine Busy 0 = Hash Engine Ready	1'b0
SJ_JTAG_DETECT	[0]	Secure JTAG detection flag 1 = JTAG access detection 0 = No JTAG access detection	1'b0

4.8.3 Secure JTAG result register (SJ_RESULT_REG, R, Address = 0xF408_0008)

SJ_RESULT_REG	Bit	Description	Reset Value
Reserved	[31:16]	Reserved	16'b0
SJ_ACCESS_MODE	[15:8]	Secure JTAG access mode	8'b0
Reserved	[7:6]	Reserved	2'b0
SJ_SEC_ACCESS_EN	[5]	Secure JTAG Secure Access Enable	1'b0
SJ_NSEC_ACCESS_EN	[4]	Secure JTAG Non-secure Access Enable	1'b0
Reserved	[3]	Reserved	1'b0
SJ_SM_OKAY	[2]	Secure JTAG access mode integrity check 1 = Pass 0 = Fail or not checked	1'b0
SJ_ID_OKAY	[1]	Secure JTAG password ID check 1 = Pass 0 = Fail or not checked	1'b0
SJ_KEY_HASH_OKAY	[0]	Secure JTAG password integrity check 1 = Pass 0 = Fail or not checked	1'b0

4.8.4 Secure JTAG Header Information Register (SJ_HEADER_INFO_REG, R, Address = 0xF408_000C)

SJ_HEADER_INFO_REG	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	8'b0
sj_header_info	[23:0]	Secure JTAG header information	24'b0

4.8.5 Secure JTAG Password Register (SJ_PASSWORD_REG, R/W, Address = 0xF408_0020)

SJ_PASSWORD_REG	Bit	Description	Reset Value
sj_password	[31:0]	Secure JTAG password register	32'b0

4.8.6 Secure JTAG Password Control Register (SJ_PASSWORD_CTRL_REG, W, Address = 0xF408_0024)

SJ_PASSWORD_CTRL_REG	Bit	Description	Reset Value
Reserved	[31:2]	Reserved	30'b0
SJ_PWD_FINISH	[1]	Secure JTAG password finish command	1'b0
SJ_PWD_START	[0]	Secure JTAG password start command	1'b0

11.4 SECURITY KEY

1 OVERVIEW

S5PC100 provides 128-bit, 80-bit x2 and 96-bit Electrical fuse ROM for security key. You can use it as root key, secure boot key and secure JTAG key by programming it through E-from test ports. Hence these electrical fuse ROMs are used for security function, it also provides protection scheme to prevent mal-user from reading key values through E-from test ports.

1.1 FEATURES

- 128-bit electrical fuse ROM0 for Root key
  - Generally, Root key is used as a seed for encryption/decryption of security data.
- 80-bit electrical fuse ROM1,2 for Secure boot key
  - Secure boot key is used as a public key hash value for integrity check in secure booting sequence,
  - In booting time, 1st boot loader reads secure boot key. And if secure boot key is all zero, then it boots without security check.
- 96-bit electrical fuse ROM3 for Secure JTAG key
  - Secure JTAG key is used as key for checking password sequence in secure JTAG sequence.
  - It also has read lock bit for each electrical fuse ROM.
  - Electrical fuse ROM3 [82]: Read lock for electrical fuse ROM0 (Root key)
  - Electrical fuse ROM3 [83]: Read lock for electrical fuse ROM1,2 (Secure boot key)
  - Electrical fuse ROM3 [84]: Read lock for electrical fuse ROM3 (Secure JTAG key)

1.2 BLOCK DIAGRAM

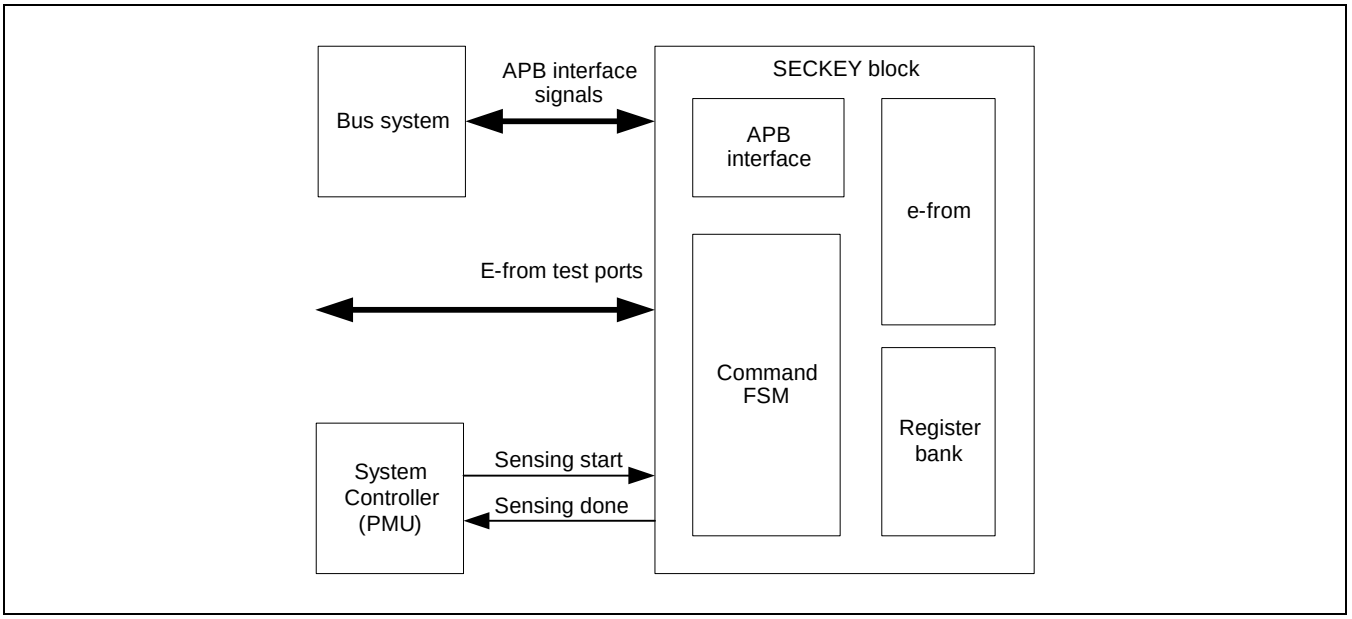


Figure 11.4-1 Security Key Block Diagram

2 ELECTRICAL FUSE ROM

e-fuse ROM	Size	Purpose	Size (bit)	SW Access	Description
0	128	Root key	128	O	Root key for encryption/ decryption of general key.
1	80	Secure boot key	160	O	Secure boot key for secure booting
2	80				
3	96	Secure JTAG key	80	X	Secure JTAG key for JTAG protection
		JTAG access mode	1	X	The access mode of debugging transaction (0: Non-secure Access, 1: Secure Access)
		JTAG soft lock	1		Authentication on/off (0: JTAG Security Check Off, 1: JTAG Security Check On)
		e-fuse read lock for Root key	1		E-from test port read disable (0 : Enables e-fuse port read, 1: Disables e-fuse port read)
		e-fuse read lock for Secure JTAG key	1		E-from test port read disable (0: Enables e-fuse port read, 1: Disables e-fuse port read)
		e-fuse read lock for Secure boot key	1		E-from test port read disable (0: Enables e-fuse port read, 1: Disables e-fuse port read)
		reserved	11	O	Spare E-from area

3 I/O DESCRIPTION

Funtion Signal	I/O	Description	Pad	Type
efrom_fsource_0	I	Programming voltage. VDD to program fuses, VSS at all other time	XefFSOURCE_0	Analog
efrom_fvgate_0	I	Separate power supply for programming FET gate and selection logic. VSS to program fuses, VDD(1.2V). at all other time	XefVGATE_0	Analog

4 REGISTER DESCRIPTION

Register	Address	Type	Description	Reset Value
ROOTKEY0	0xF500_0000	R	Root Key 0 Register	0xXXXX_XXXX
ROOTKEY1	0xF500_0004	R	Root Key 1 Register	0xXXXX_XXXX
ROOTKEY2	0xF500_0008	R	Root Key 2 Register	0xXXXX_XXXX
ROOTKEY3	0xF500_000C	R	Root Key 3 Register	0xXXXX_XXXX
SECKEY0	0xF500_0018	R	Secure Boot Key 0 Register	0xXXXX_XXXX
SECKEY1	0xF500_001C	R	Secure Boot Key 1 Register	0xXXXX_XXXX
SECKEY2	0xF500_0020	R	Secure Boot Key 2 Register	0xXXXX_XXXX
SECKEY3	0xF500_0024	R	Secure Boot Key 3 Register	0xXXXX_XXXX
SECKEY4	0xF500_0028	R	Secure Boot Key 4 Register	0xXXXX_XXXX
SPARE_EFUSE_AREA	0xF500_0038	R	SPARE E-fuse Area Register	0x0000_07FF

NOTES:

1. Although the reset value of registers is 0xXXXX_XXXX, as power on sequence is progressing, the e-fuse values are loaded to the registers. Hence, SW cannot read register reset value itself, but it can only read the loaded current e-fuse values.
2. The initial value of all e-fuse bit is '0'.

4.1 ROOT KEY 0 REGISTER (ROOTKEY0, R, ADDRESS = 0XF500_0000)

ROOTKEY0	Bit	Description	Reset Value
RootKey0	[31:0]	Root Key [31:0]	0xXXXX_XXXX

4.2 ROOT KEY 1 REGISTER (ROOTKEY1, R, ADDRESS = 0XF500_0004)

ROOTKEY1	Bit	Description	Reset Value
RootKey1	[31:0]	Root Key [63:32]	0xXXXX_XXXX

4.3 ROOT KEY 2 REGISTER (ROOTKEY2, R, ADDRESS = 0XF500_0008)

ROOTKEY2	Bit	Description	Reset Value
RootKey2	[31:0]	Root Key [95:64]	0xXXXX_XXXX

4.4 ROOT KEY 3 REGISTER (ROOTKEY3, R, ADDRESS = 0XF500_000C)

ROOTKEY3	Bit	Description	Reset Value
RootKey3	[31:0]	Root Key [127:96]	0xXXXX_XXXX

4.5 SECURE BOOT KEY 0 REGISTER (SECKEY0, R, ADDRESS = 0XF500_0018)

SECKEY0	Bit	Description	Reset Value
Seckey0	[31:0]	Secure Boot Key [31:0]	0xXXXX_XXXX

4.6 SECURE BOOT KEY 1 REGISTER (SECKEY1, R, ADDRESS = 0XF500_001C)

SECKEY1	Bit	Description	Reset Value
Seckey1	[31:0]	Secure Boot Key [63:32]	0xXXXX_XXXX

4.7 SECURE BOOT KEY 2 REGISTER (SECKEY2, R, ADDRESS = 0XF500_0020)

SECKEY2	Bit	Description	Reset Value
Seckey2	[31:0]	Secure Boot Key [95:64]	0xXXXX_XXXX

4.8 SECURE BOOT KEY 3 REGISTER (SECKEY3, R, ADDRESS = 0XF500_0024)

SECKEY3	Bit	Description	Reset Value
Seckey3	[31:0]	Secure Boot Key [127:96]	0xXXXX_XXXX

4.9 SECURE BOOT KEY 4 REGISTER (SECKEY4, R, ADDRESS = 0XF500_0028)

SECKEY4	Bit	Description	Reset Value
Seckey4	[31:0]	Secure Boot Key [159:128]	0xXXXX_XXXX

4.10 SPARE E-FUSE AREA REGISTER (SPARE_EFUSE_AREA, R, ADDRESS = 0XF500_0038)

SPARE_EFUSE_AREA	Bit	Description	Reset Value
Reserved	[31:12]	Write as zero.	0
EFUSE_BIT	[10:0]	E-fuse reserved bits	0x7FF

12.1 ELECTRICAL DATA

1 ABSOLUTE MAXIMUM RATINGS

Any stress beyond the absolute maximum ratings listed in Table 12.1-1 can cause permanent damage to the device.

Table 12.1- 1 Absolute Maximum Rating

Parameter	Symbol	Value	Unit
DC Power Supply	VDD_APLL, VDD_MPLL, VDD_EPLL, VDD_HPLL, VDD_INT, VDD_ALIVE, VDD_ARM, VDD12_UOTG, VDD12_HDMI, VDD12_MIPI	1.8	V
	VDD18_MIPI_PLL, VDDQ18_MIPI, VDDQ_DDR	2.5	
	VDDQ_M0, VDDQ_AUD, VDDQ_CAN, VDDQ_EXT, VDDQ_MSM, VDDQ_LCD, VDDQ_MMC2, VDDQ_SYS0, VDDQ_SYS2, VDDQ_SYS5, VDDQ_MMC2, VDDQ_CI, VDD33_UOTG, VDDQ_UHOST, VDD_RTC, VDD30_DAC_A, VDD30_DAC_D, VDD33_ADC	4.6	
DC Signal Input	1.8v Input Buffer	2.5	
	3.3v Input Buffer	4.6	
	3.3v with 5v tolerance Input Buffer	6.5	
I/O Current	$I_{I/O}$	± 20	mA
Storage Temperature	T_{STG}	– 65 ~ 150	°C

2 GUARANTEED OPERATING CONDITIONS

S5PC100 should be used under the conditions listed in Table 12.1- 2

Table 12.1- 2 Guaranteed Operating Conditions (Ta=-25~85°C , Unit=V)

Pad Name	Min	Typ	Max	Note
VDD_ALIVE	1.15	1.2	1.25	
VDD_APLL VDD_MPLL VDD_EPLL VDD_HPLL	1.15	1.2	1.25	
VDD_INT	1.15	1.2	1.25	For ARM Clk = 667Mhz
	1.25	1.30	1.35	For ARM Clk = 833Mhz
VDD12_UOTG,	1.15	1.2	1.25	
VDD_ARM	1.15	1.2	1.25	For ARM Clk = 667Mhz
	1.30	1.35	1.40	For ARM Clk = 833Mhz
VDDQ_AUD	1.7	1.8~3.3	3.6	
VDDQ_CAN	1.7	1.8~3.3	3.6	
VDDQ_EXT	1.7	1.8~3.3	3.6	
VDDQ_MSM	1.7	1.8~3.3	3.6	
VDDQ_LCD	1.7	1.8~3.3	3.6	
VDDQ_SYS0	1.7	1.8~3.3	3.6	
VDDQ_SYS2	1.7	1.8~3.3	3.6	
VDDQ_SYS5	1.7	1.8~3.3	3.6	
VDDQ_MMC2	1.7	1.8~3.3	3.6	
VDDQ_CI	1.7	1.8~3.3	3.6	
VDDQ_DDR	1.7	1.8	1.9	For mDDR/DDR2
	1.15	1.2	1.25	For LPDDR2
VDDQ_M0	1.7	1.8~3.3	3.6	
VDD12_MIPI	1.15	1.2	1.25	
VDDQ18_MIPI, VDD18_MIPI_PLL	1.7	1.8	1.9	
VDD12_HDMI	1.15	1.2	1.25	
VDDQ_RTC	1.7	1.8~3.3	3.6	
VDD33_UOTG,	3.15	3.3	3.45	
VDDQ33_UHOST	3.0	3.3	3.6	
VDD33_ADC	3.0	3.3	3.6	
VDD30_DAC_A VDD30_DAC_D	2.6	3.0	3.3	

3 D.C. ELECTRICAL CHARACTERISTICS

The entire DC characteristics listed in table below for each pin include input sense levels, output drive levels, and currents. Use these parameters to determine Max DC loading and to determine Max transition times for a given load. Table 12.1- 3 and Table 12.1- 4 shows the DC operating conditions for the high- and low-strength input, output, and I/O pins. .

Table 12.1- 3 Normal I/O PAD DC Electrical Characteristics (Vtyp – 3.3v)

VDD=1.7v~3.60v, Vext=3.0~3.6v, TA=-40 to 85°C

Parameter		Condition		Min	Typ	Max	Unit
Vih	High Level Input Voltage						
	LVCMOS Interface			0.7*VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	LVCMOS Interface			-0.3		0.3*VDD	V
ΔV	Hysteresis Voltage			0.1*VDD			V
Iih	High Level Input Current						
	Input Buffer	Vin=VDD		-10		10	uA
	Tolerant Input Buffer**	Vin=Vext		-10		10	uA
	Input Buffer with pull-down	Vin=VDD	VDD=3.3V	20	70	130	uA
			VDD=2.5V	10	40	80	
			VDD=1.8V	5	20	40	
	Tolerant Input Buffer with pull-up**	Vin=5V	VDD=3.3V	10	30	60	uA
		Vin=3.3V	VDD=2.5V	6	16	50	
		Vin=3.3V	VDD=1.8V	2	8	18	
Iil	Low Level Input Current						
	Input Buffer	Vin=VSS		-10		10	uA
	Input Buffer with pull-up	Vin=VSS	VDD=3.3V	-130	-70	-20	uA
			VDD=2.5V	-80	-40	-10	
			VDD=1.8V	-40	-20	-5	
Voh	Type A,B,C	Ioh=-100uA		VDD-0.2			V
Vol	Type A,B,C	Iol=100uA				0.2	V
Ioz	Tri-State Output Leakage Current	Vout=VSS or VDD		-10		10	uA
CIN	Input capacitance	Any input and Bidirectional buffers				5	pF
COUT	Output capacitance	Any output buffer				5	pF

Table 12.1- 4 Normal I/O PAD DC Electrical Characteristics (Vtyp – 2.5v)

VDD=1.7V~2.7V, Vext=3.0~3.6V , T_A = -40 to 85℃

Parameter		Condition		Min	Typ	Max	Unit
Vih	High Level Input Voltage						
	LVCMOS Interface			0.7*VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	LVCMOS Interface			-0.3		0.3*VDD	V
ΔV	Hysteresis Voltage			0.1*VDD			V
Iih	High Level Input Current						
	Input Buffer	Vin=VDD		-10		10	uA
	Tolerant Input Buffer**	Vin=Vext		-10		10	uA
	Input Buffer with pull-down	Vin=VDD	VDD=2.5V	10	40	80	uA
			VDD=1.8V	5	20	40	
	Tolerant Input Buffer with pull-up**	Vin=3.3V	VDD=2.5V	3	10	40	uA
		Vin=3.3V	VDD=1.8V	1	4	10	
Iil	Low Level Input Current						
	Input Buffer	Vin=VSS		-10		10	uA
	Input Buffer with pull-up	Vin=VSS	VDD=2.5V	-80	-40	-10	uA
			VDD=1.8V	-40	-20	-5	
Voh	Type A,B,C	Ioh=-100uA		VDD-0.2			V
Vol	Type A,B,C	Iol=100uA				0.2	V
Ioz	Tri-State Output Leakage Current	Vout=VSS or VDD		-10		10	uA
CIN	Input capacitance	Any input and Bidirectional buffers				5	pF
COUT	Output capacitance	Any output buffer				5	pF

Table 12.1- 5 Special Memory I/O PAD DC Electrical Characteristics (Memory Port 0)

Symbol	Parameter	Min	Typ	Max	Unit
VDDm0	Output supply voltage	1.7	2.5	3.6	V
VDD_INT NT	Internal Voltage	1.15	1.2	1.25	V
Temp	Ambient Temperature	-25	25	85	°C
V_{IH}	dc Input Logic High	0.7 * VDDm0	-	VDDm0+0.3	V
V_{IL}	dc Input Logic Low	-0.3	-	0.3 * VDDm0	V
I_{IH}	High Level Input Current	-10	-	10	uA
I_{IL}	Low Level Input Current	-10	-	10	uA
I_{IH}	High Level Input Current with Pull Down (VDD=3.3V)	20	70	130	uA
	High Level Input Current with Pull Down (VDD=2.5V)	10	40	80	uA
	High Level Input Current with Pull Down (VDD=1.8V)	5	20	40	uA
I_{IL}	Low Level Input Current with Pull Up (VDD=3.3V)	-130	-70	-20	uA
	Low Level Input Current with Pull Up (VDD=2.5V)	-80	-40	-10	uA
	Low Level Input Current with Pull Up (VDD=1.8V)	-40	-20	-5	uA
V_{OH}	Output High Voltage(@Ioh=-100uA)	VDDm0 - 0.2	-	-	V
V_{OL}	Output Low Voltage(@Iol=100uA)	-	-	0.2	V

Table 12.1- 6 Special Memory DDR I/O PAD DC Electrical Characteristics (Memory Port 1)

Symbol	Parameter	Min	Typ	Max	Unit
VDDm1	Output supply voltage for LPDDR2	1.15	1.2	1.25	V
VDDm1	Output supply voltage for LPDDR1	1.7	1.8	1.9	V
VDD_INT NT	Internal Voltage	1.15	1.2	1.25	V
Temp	Ambient Temperature	-25	25	85	°C
V_{IH}	dc Input Logic High	0.7 * VDDm1	-	VDDm1+0.3	V
V_{IL}	dc Input Logic Low	-0.3	-	0.3 * VDDm1	V
I_{IH}	High Level Input Current	-10	-	10	uA
I_{IL}	Low Level Input Current	-10	-	10	uA
I_{IH}	High Level Input Current with Pull Down (VDD=2.5V)	10	40	80	uA
	High Level Input Current with Pull Down (VDD=1.8V)	5	20	40	uA
I_{IL}	Low Level Input Current with Pull Up (VDD=2.5V)	-80	-40	-10	uA
	Low Level Input Current with Pull Up (VDD=1.8V)	-40	-20	-5	uA
V_{OH}	Output High Voltage(@Ioh=-100uA)	VDDm1 – 0.2	-	-	V
V_{OL}	Output Low Voltage(@Iol=100uA)	-	-	0.2	V

Table 12.1- 7 USB DC Electrical Characteristics

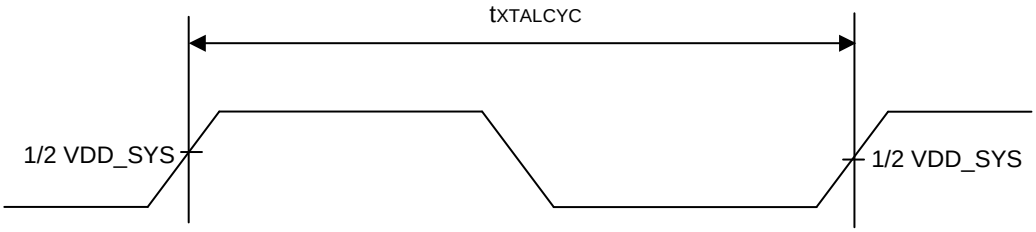
Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	High level input voltage		2.0		V
V_{IL}	Low level input voltage			0.8	V
I_{IH}	High level input current	Vin = 3.3V	-10	10	μA
I_{IL}	Low level input current	Vin = 0.0V	-10	10	μA
V_{OH}	Static Output High	15Kohm to GND	2.8	3.6	V
V_{OL}	Static Output Low	1.5Kohm to 3.6V		0.3	V

Table 12.1- 8 RTC OSC DC Electrical Characteristics

Symbol	Parameter	Min	Type	Max	Unit
VDDrtc	Output supply voltage @CD0 of XrtcXTI PAD is one	1.7	1.8~2.2	2.3	V
	Output supply voltage @CD0 of XrtcXTI PAD is zero	2.3	2.5~3.3	3.6	V
V_{IH}	DC input logic high	0.7*VDDrtc			V
V_{IL}	DC input logic low			0.3*VDDrtc	V
I_{IH}	High level input current	-10		10	μA
I_{IL}	Low level input current	-10		10	μA

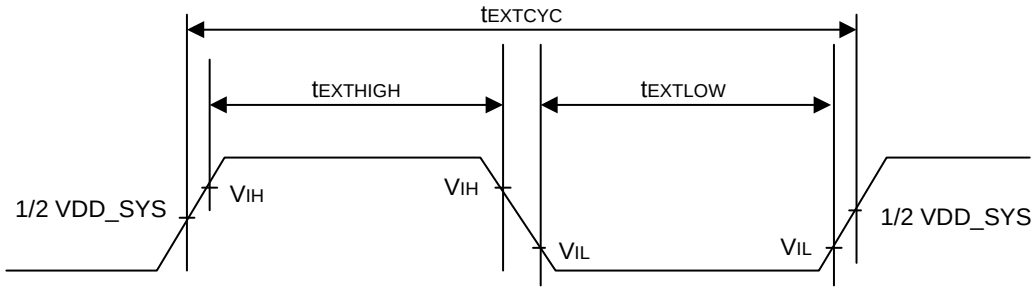
4 CLK A.C. ELECTRICAL CHARACTERISTICS

Pin's Alternating-Current (AC) characteristics include input and output capacitance. These factors determine the loading for external drivers and other load analyses. The AC characteristics also include a derating factor, which indicates how much the AC timings might vary with different loads.



NOTE: The clock input from the XTIpll pin.

Figure 12.1- 1 XTIpll Clock Timing



NOTE: The clock input from the EXTCLK pin.

Figure 12.1- 2 EXTCLK Clock Input Timing

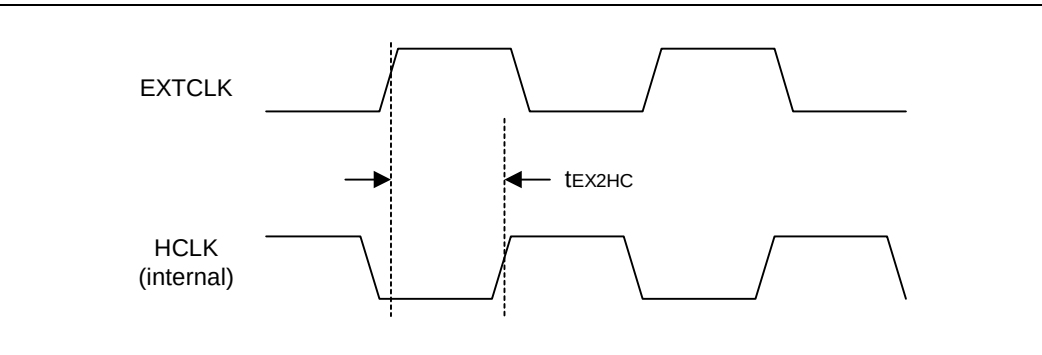


Figure 12.1- 3 EXTCLK/HCLK in Case that EXTCLK is Used without the PLL

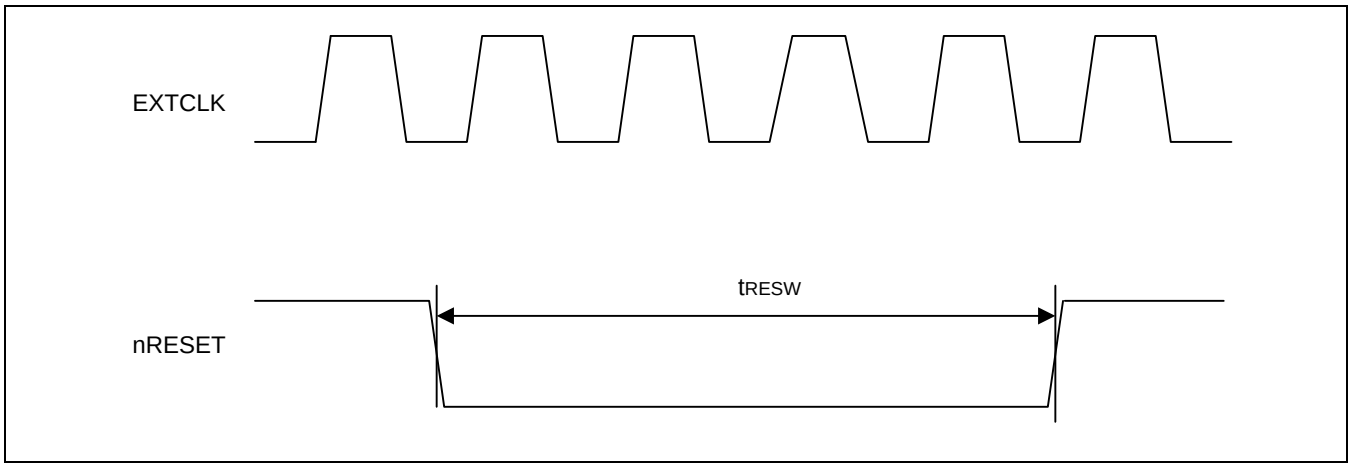


Figure 12.1- 4 Manual Reset Input Timing

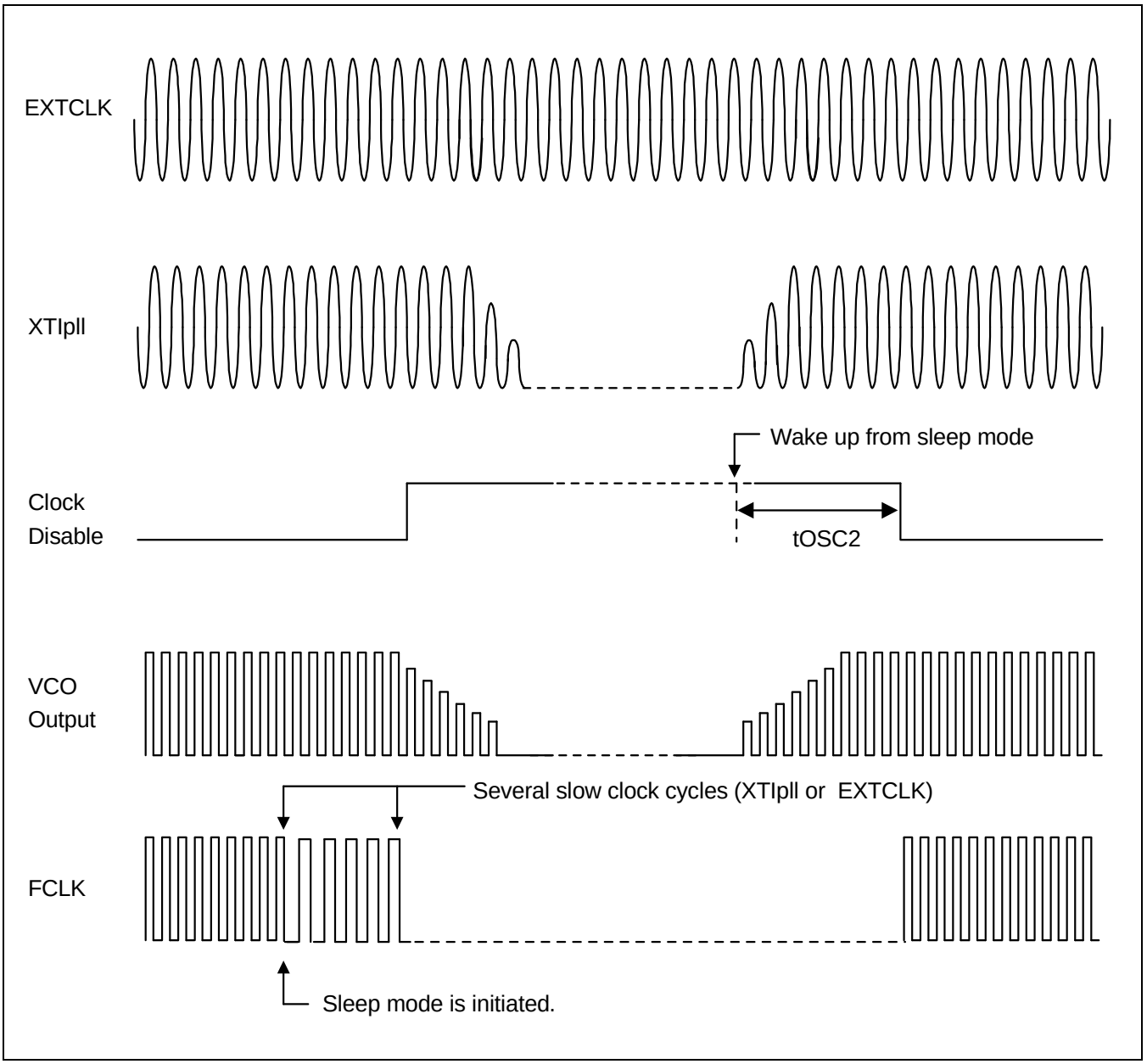


Figure 12.1- 5 Sleep Mode Return Oscillation Setting Timing

Table 12.1- 9 Clock Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDSYS = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min	Typ	Max	Unit
VDDpadIO to VDDalive	tOA	0			ms
VDDalive to VDD_INTNT/ VDDarm	tAI	1			us
VDDarm to PWR_EN(PWRRGTON)	tAE	1		10	ns
VDDLOGIC/VDDarm to Oscillator stabilization	tOSC	10			cycle
Oscillator stabilization to nRESET and nTRST high	tOR	1			us
External clock input high level pulse width	$t_{EXTHIGH}$	25		-	ns
External clock to HCLK (without PLL)	t_{EX2HC}	5		10	ns
HCLK (internal) to CLKOUT	t_{HC2CK}	4		10	ns
HCLK (internal) to SCLK	$t_{HC2SCLK}$	2		8	ns
Reset assert time after clock stabilization	t_{RESW}	4		-	XTIpll or EXTCLK
APLL and MPLL Lock Time	t_{PLL}	-		300	us
EPLL Lock Time		-		300	us
Sleep mode return oscillation setting time. (2)	t_{OSC2}	2^4		2^{16}	XTIpll or EXTCLK
Interval before CPU runs after nRESET is released.	$t_{RST2RUN}$	5		-	XTIpll or EXTCLK

5 ROM/ SRAM AC ELECTRICAL CHARACTERISTICS

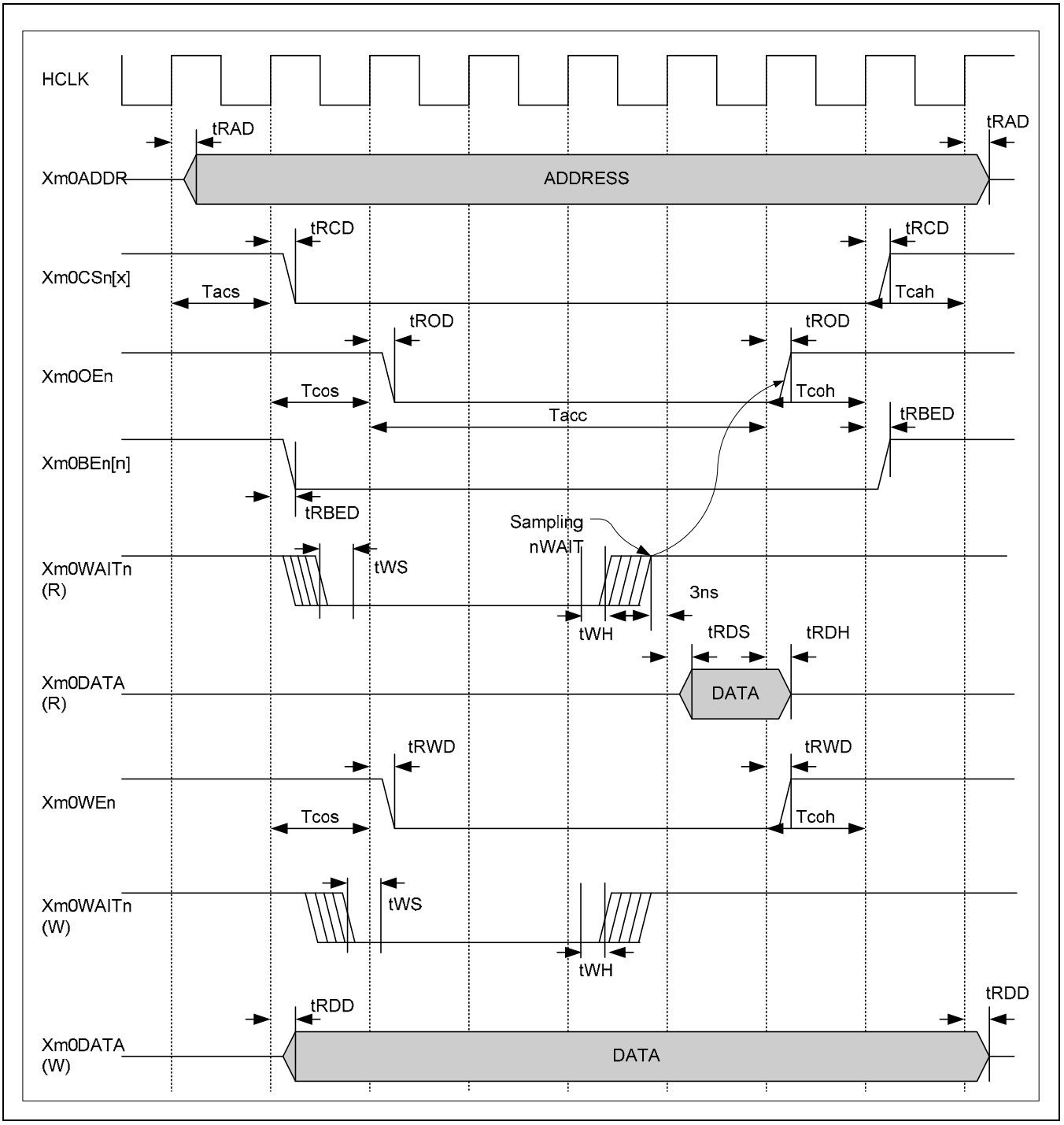


Figure 12.1- 6 ROM/ SRAM Timing
(Tacs = 0, Tcos = 0, Tacc = 2, Tcoh = 0, Tcah = 0, PMC = 0, ST = 0, DW = 16-bit)

Table 12.1- 10 ROM/SRAM Bus Timing Constants

(VDD_INTNT = 1.2V± 0.05V, TA = -40 to 85°C, VDDm0 = 1.7V - 3.6V)

Parameter	Symbol	Min	Max	Unit
ROM/SRAM Address Delay	t_{RAD}	1.46	6.98	ns
ROM/SRAM Chip Select 0 Delay	t_{RCD}	1.94	6.83	ns
ROM/SRAM Chip Select 1 Delay	t_{RCD}	2.06	7.04	ns
ROM/SRAM Chip Select 2 Delay	t_{RCD}	2.34	7.69	ns
ROM/SRAM Chip Select 3 Delay	t_{RCD}	1.90	6.42	ns
ROM/SRAM Chip Select 4 Delay	t_{RCD}	2.00	6.70	ns
ROM/SRAM Chip Select 5 Delay	t_{RCD}	1.66	5.95	ns
ROM/SRAM nOE(Output Enable) Delay	t_{ROD}	2.34	7.12	ns
ROM/SRAM nWE(Write Enable) Delay	t_{RWD}	2.29	7.27	ns
ROM/SRAM Byte Enable Delay	t_{RBED}	1.76	6.56	ns
ROM/SRAM Output Data Delay	t_{RDD}	1.70	8.07	ns
ROM/SRAM Read Data Setup Time	t_{RDS}	2.00	-	ns
ROM/SRAM Write Data Hold Time	t_{RDH}	1.00	-	ns

6 ONENAND AC ELECTRICAL CHARACTERISTICS

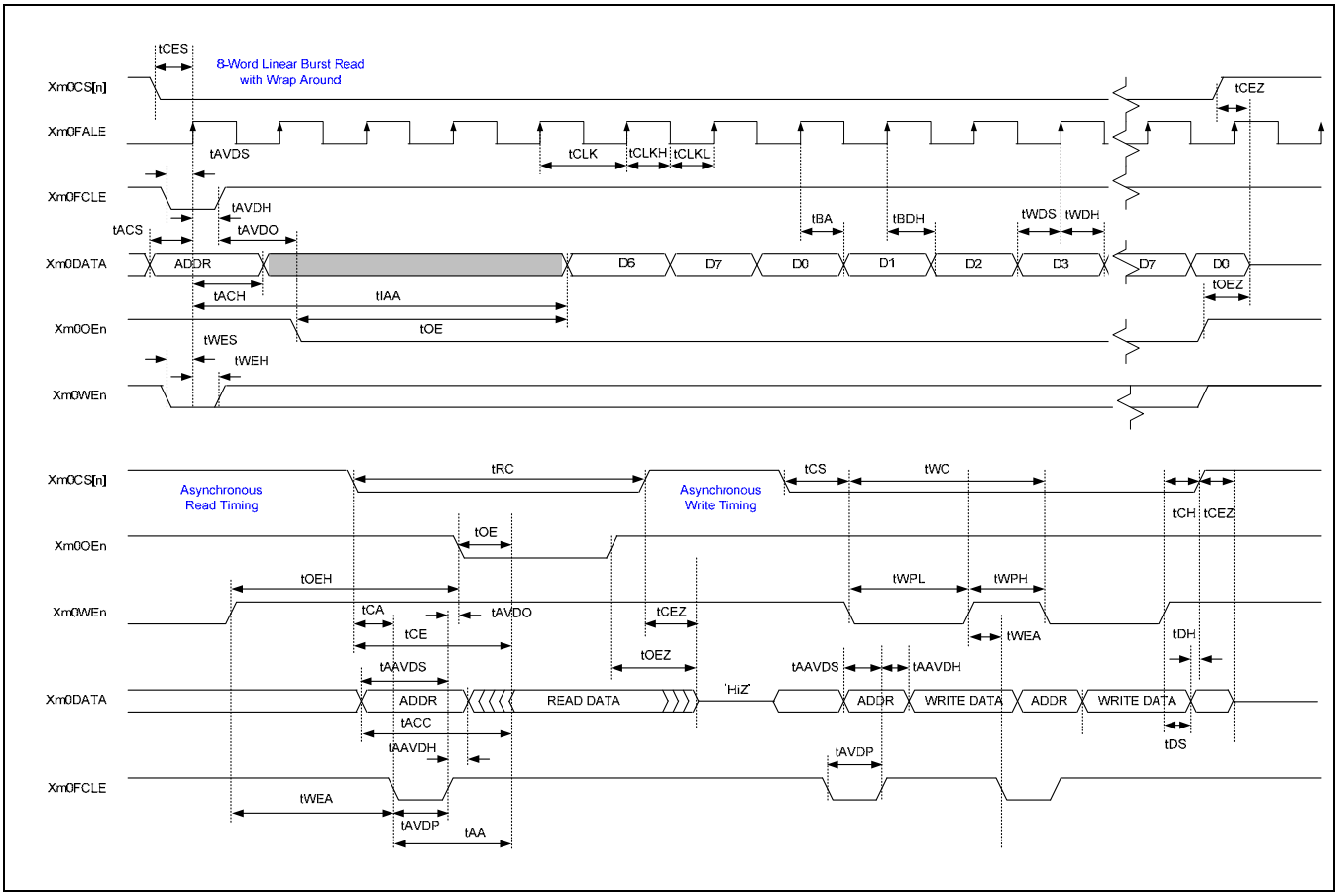


Figure 12.1- 7 OneNand Flash Timing

Table 12.1- 11 OneNAND Bus Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDm0 = 1.7V – 1.9V)

Parameter	Symbol	Min	Max	Unit
OneNAND SMCLK cycle	t_{CLK}	12	-	ns
OneNAND Clock High time	t_{CLKH}	5	-	ns
OneNAND Clock Low time	t_{CLKL}	5	-	ns
OneNAND CSn Setup time to SMCLK	t_{CES}	4.5	-	ns
OneNAND Initial Access time	t_{IAA}	-	70	ns
OneNAND Burst Access time valid SMCLK to Output delay	t_{BA}	-	9	ns
OneNAND Data Hold time from next clock cycle	t_{BDH}	2	-	ns
OneNAND Output Enable to Data	t_{OE}	-	20	ns
OneNAND CSn Disable to Output High Z	t_{CEZ}	-	20	ns
OneNAND OEn Disable to Output High Z	t_{OEZ}	-	15	ns
OneNAND Address Setup time to SMCLK	t_{ACS}	4	-	ns
OneNAND Address Hold time to SMCLK	t_{ACH}	6	-	ns
OneNAND ADRVALID Setup time to SMCLK	t_{AVDS}	4	-	ns
OneNAND ADRVALID Hold time to SMCLK	t_{AVDH}	6	-	ns
OneNAND Write Data Setup time to SMCLK	t_{WDS}	4	-	ns
OneNAND Write Data Hold time to SMCLK	t_{WDH}	2	-	ns
OneNAND WEn Setup time to SMCLK	t_{WES}	4	-	ns
OneNAND WEn Hold time to SMCLK	t_{WEH}	6	-	ns
OneNAND ADRVALID high to OEn low	t_{AVDO}	0	-	ns
OneNAND Access time from CSn low	t_{CE}	-	76	ns
OneNAND Asynchronous Access time from ADRVALID low	t_{AA}	-	76	ns
OneNAND Asynchronous Access time from address valid	t_{ACC}	-	76	ns
OneNAND Read Cycle time	t_{RC}	76	-	ns
OneNAND ADRVALID low pulse width	t_{AVDP}	12	-	ns
OneNAND Address Setup to rising edge of ADRVALID	t_{AAVDS}	5	-	ns
OneNAND Address Hold to rising edge of ADRVALID	t_{AAVDH}	7	-	ns
OneNAND CSn Setup to ADRVALID falling edge	t_{CA}	0	-	ns
OneNAND WEn Disable to ADRVALID enable	t_{WEA}	15	-	ns
OneNAND Address to OEn low	t_{ASO}	10	-	ns
OneNAND WEn Cycle time	t_{WC}	70	-	ns
OneNAND Data Setup time	t_{DS}	30	-	ns

Parameter	Symbol	Min	Max	Unit
OneNAND Data Hold time	t_{DH}	0	-	ns
OneNAND CSn Setup time	t_{CS}	0	-	ns
OneNAND CSn Hold time	t_{CH}	0	-	ns
OneNAND WEn Pulse width low	t_{WPL}	40	-	ns
OneNAND WEn Pulse width high	t_{WPH}	30	-	ns

7 NFCON AC ELECTRICAL CHARACTERISTICS

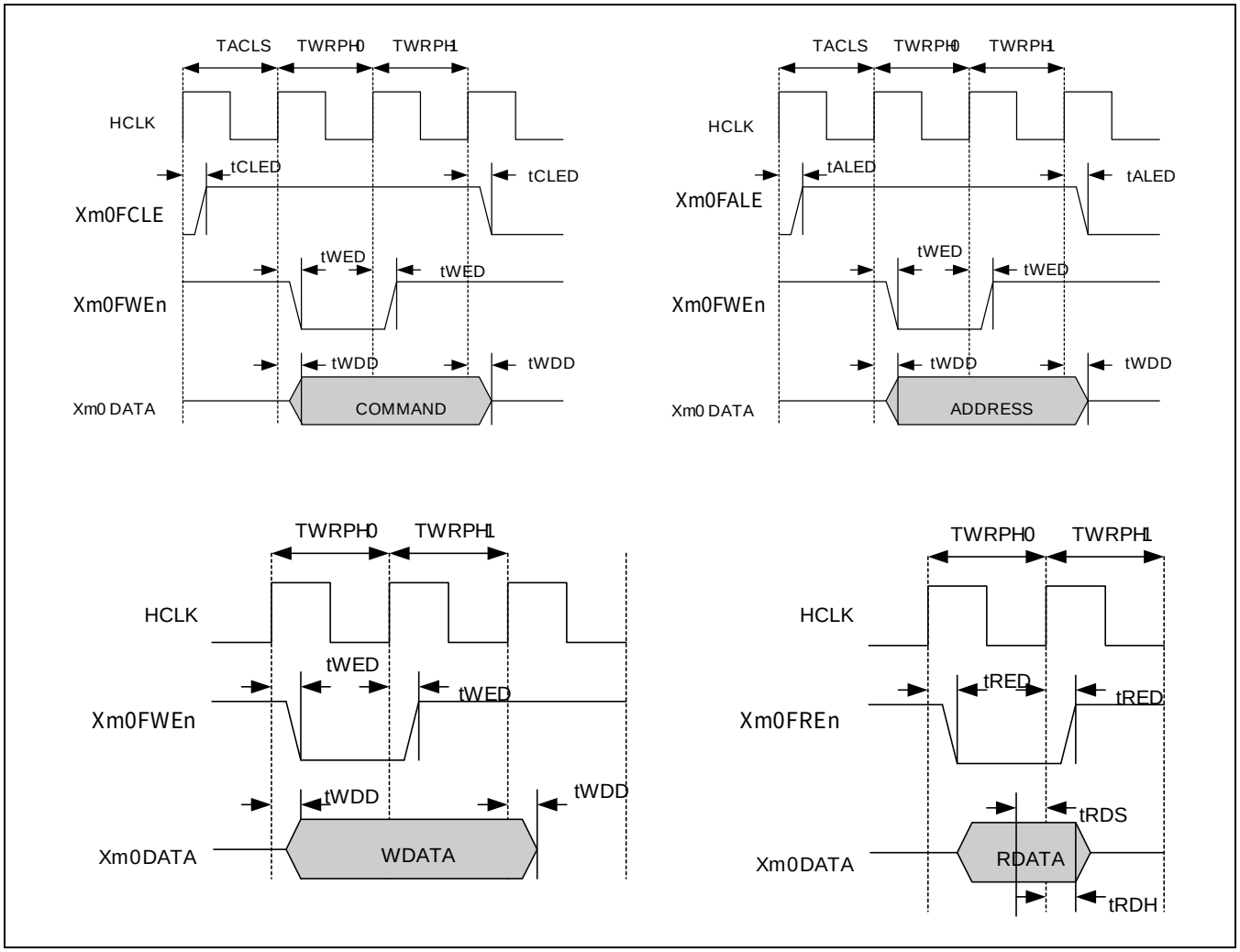


Figure 12.1- 8 NAND Flash Timing

Table 12.1- 12 NFCON Bus Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDm0 = 1.7V - 3.6V)

Parameter	Symbol	Min	Max	Unit
NFCON Chip Enable delay	t_{CED}	-	8.06	ns
NFCON CLE delay	t_{CLED}	-	7.63	ns
NFCON ALE delay	t_{ALED}	-	8.16	ns
NFCON Write Enable delay	t_{WED}	-	7.75	ns
NFCON Read Enable delay	t_{RED}	-	7.18	ns
NFCON Write Data delay	t_{WDD}	-	7.96	ns
NFCON Read Data Setup requirement time	t_{RDS}	1.00	-	ns
NFCON Read Data Hold requirement time	t_{RDH}	0.20	-	ns

8 LPDDR1 (MDDR) SDRAM ELECTRICAL CHARACTERISTICS

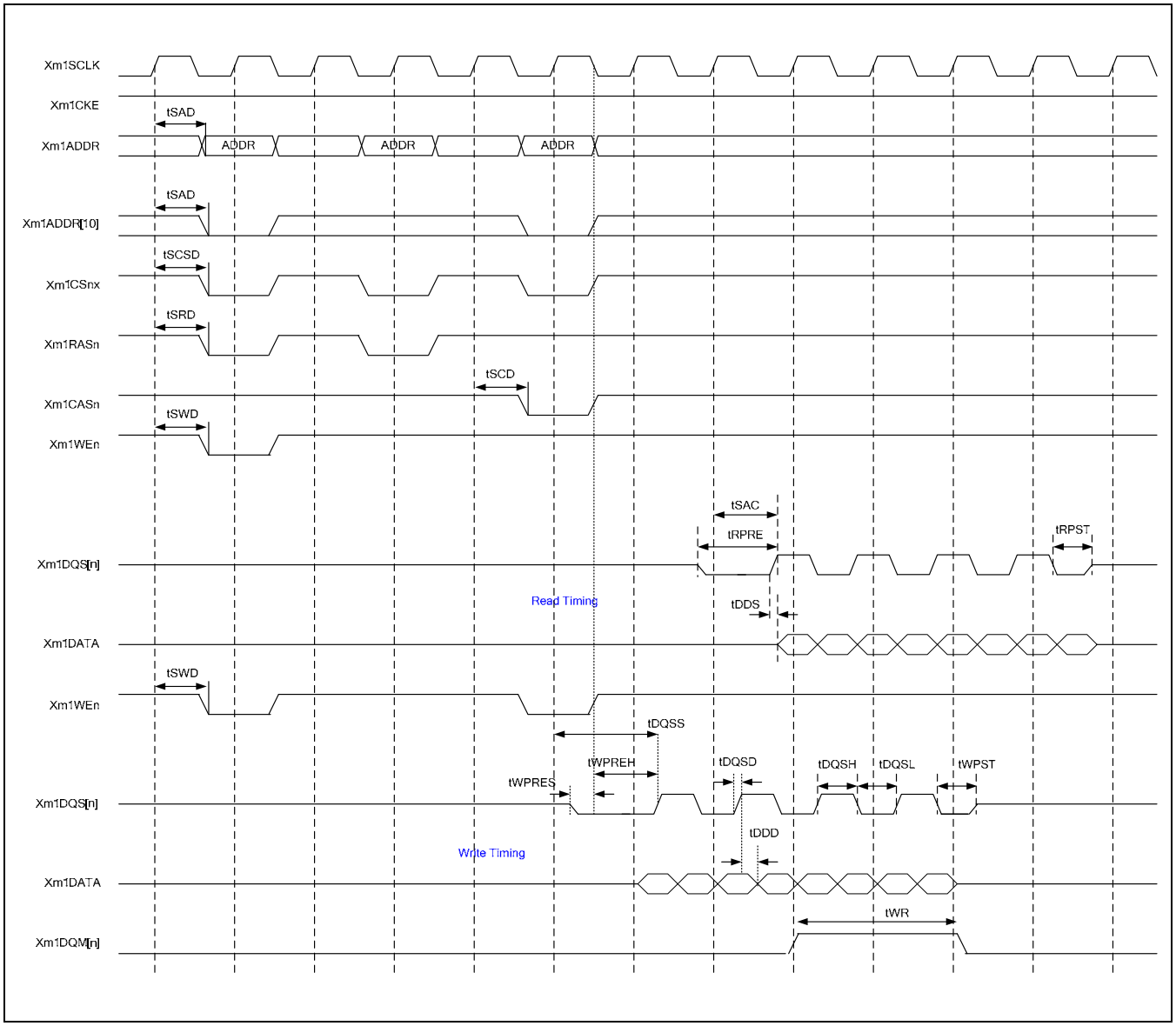


Figure 12.1- 9 LPDDR1 SDRAM Read / Write Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit)

Table 12.1- 13 Memory Port 1 Interface Timing Constants (LPDDR1 SDRAM)

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDm1 = 1.7V – 1.9V)

Parameter	Symbol	Min	Max	Unit
DDR SDRAM Address Delay	tSAD	2.86	3.73	ns
DDR SDRAM Chip Select Delay	tSCSD	2.75	2.95	ns
DDR SDRAM Row active Delay	tSRD	2.86	3.61	ns
DDR SDRAM Column active Delay	tSCD	2.85	3.50	ns
DDR SDRAM Byte Enable Delay	tSBED	2.87	3.46	ns
DDR SDRAM Write enable Delay	tSWD	2.86	3.62	ns
DDR SDRAM Output data access time from CK	tSAC	2.00	5.50	ns
DDR SDRAM Row Precharge time	tRP	18.00	-	ns
DDR SDRAM RAS to CAS delay	tRCD	18.00	-	ns
DDR SDRAM Write recovery time	tWR	12.00	-	ns
DDR SDRAM Clock low level width	tCL	0.45	0.55	tCK
DDR SDRAM Read Preamble	tRPRE	0.90	1.10	tCK
DDR SDRAM Read Postamble	tRPST	0.40	0.60	tCK
DDR SDRAM Write Postamble time	tWPST	0.40	0.60	tCK
DDR SDRAM Clock to valid DQS-In	tDQSS	0.75	1.25	tCK
DDR SDRAM DQS-In Setup time	tWPRES	1.30	-	ns
DDR SDRAM DQS-In Hold time	tWPREH	1.30	-	ns
DDR SDRAM DQS-In high level width	tDQSH	0.35	0.60	tCK
DDR SDRAM DQS-In low level width	tDQSL	0.35	0.60	tCK
DDR SDRAM read Data Setup time	tDDS	-	0.50	ns

Load Capacitance	
Xm1*	< 15pF

9 LCD CONTROLLER AC ELECTRICAL CHARACTERISTICS

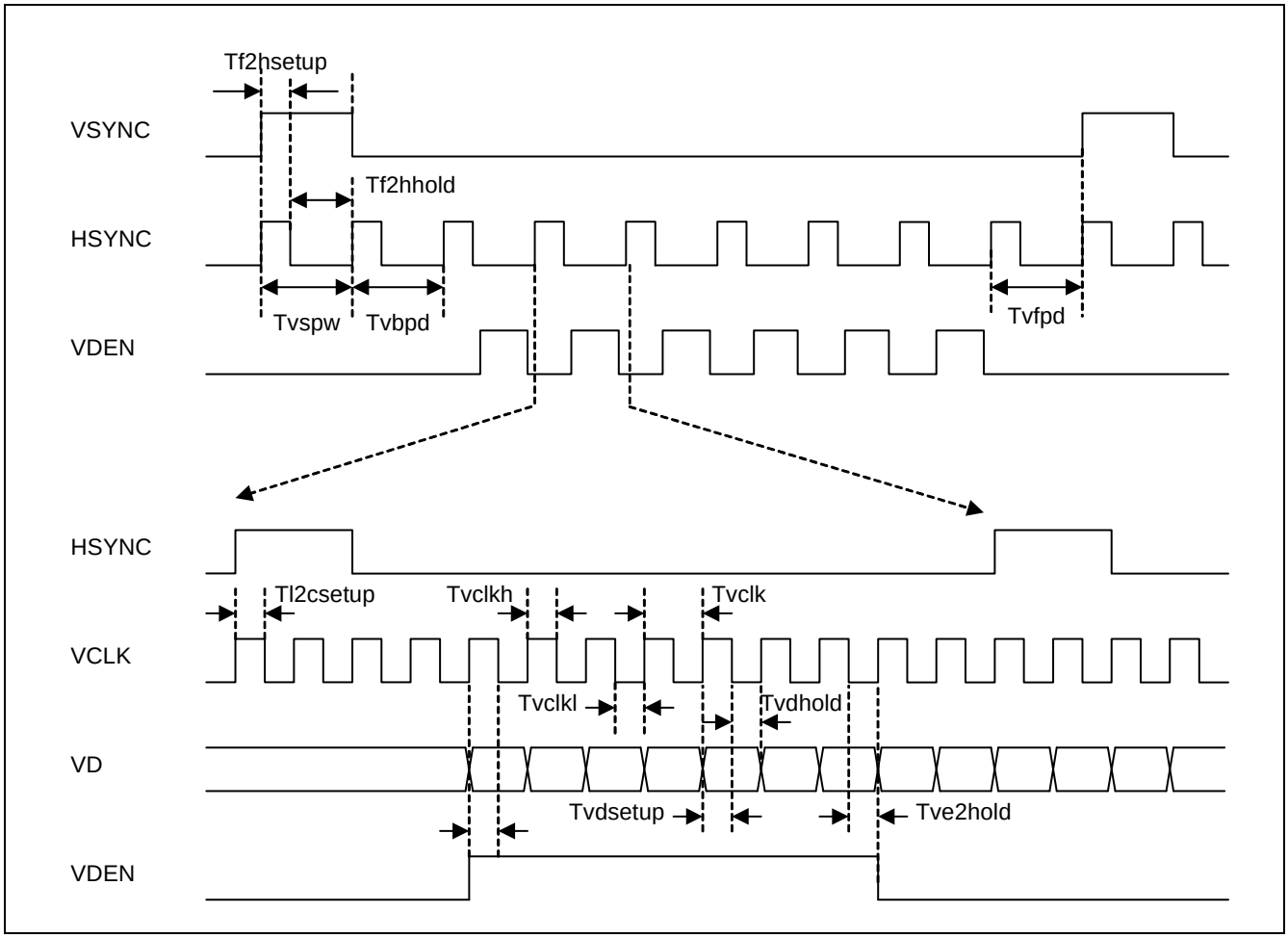


Figure 12.1- 10 LCD Controller Timing

Table 12.1- 14 TFT LCD Controller Module Signal Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDlcd = 1.7V - 3.6V)

Parameter	Symbol	Min	Typ	Max	Units
VCLK pulse width	Tvclk	12	–	–	ns
VCLK pulse width high	Tvclkh	0.3	–	–	Pvclk(1)
VCLK pulse width low	Tvclkl	0.3	–	–	Pvclk
Vertical sync pulse width	Tvspw	VSPW + 1	–	–	Phclk(2)
Vertical back porch delay	Tvbpd	VBPD+1	–	–	Phclk
Vertical front porch dealy	Tvfpd	VFPD+1	–	–	Phclk
Hsync setup to VCLK falling edge	Tl2csetup	0.3	–	–	Pvclk
VDEN setup to VCLK falling edge	Tde2csetup	0.3	–	–	Pvclk
VDEN hold from VCLK falling edge	Tde2chold	0.3	–	–	Pvclk
VD setup to VCLK falling edge	Tvd2csetup	0.3	–	–	Pvclk
VD hold from VCLK falling edge	Tvd2chold	0.3	–	–	Pvclk
VSYNC setup to HSYNC falling edge	Tf2hsetup	HSPW + 1	–	–	Pvclk
VSYNC hold from HSYNC falling edge	Tf2hhold	HBPD + HFPD + HOZVAL + 3	–	–	Pvclk

NOTES:

1. VCLK period
2. HSYNC period

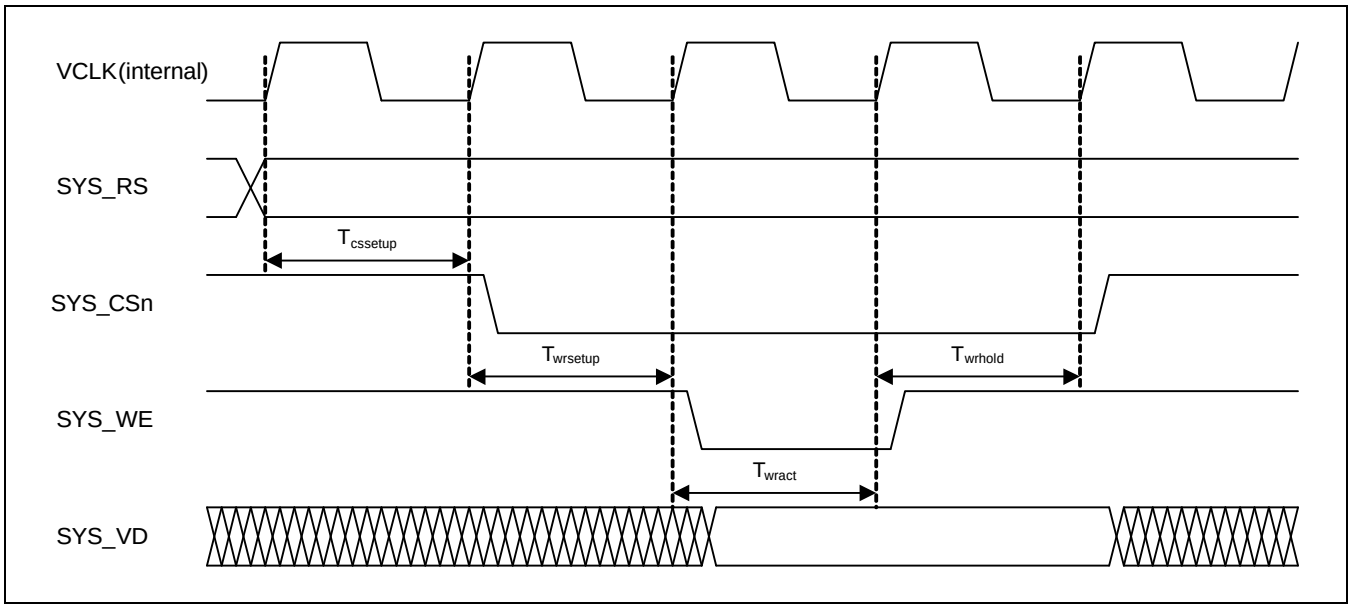


Figure 12.1- 11 LCD I80 InterfaceTiming

Table 12.1- 15 LCD I80 Interface Signal Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDlcd = 1.7V - 3.6V)

Parameter	Symbol	Min	Typ	Max	Units
SYS_RS to SYS_CSn Low	$T_{cssetup}$	-	LCD_CS_SETUP + 1	–	Pvclk*
SYS_CSn Low to SYS_WR Low	$T_{wrsetup}$	-	LCD_WR_SETUP + 1	–	Pvclk
SYS_WE Pulse Width	T_{wract}	-	LCD_WR_ACT + 1	–	Pvclk
SYS_WE Hight to SYS_CSn High	T_{wrhold}	-	LCD_WR_HOLD + 1	–	Pvclk

NOTE: Internal VCLK period

10 CAMERA INTERFACE AC ELECTRICAL CHARACTERISTICS

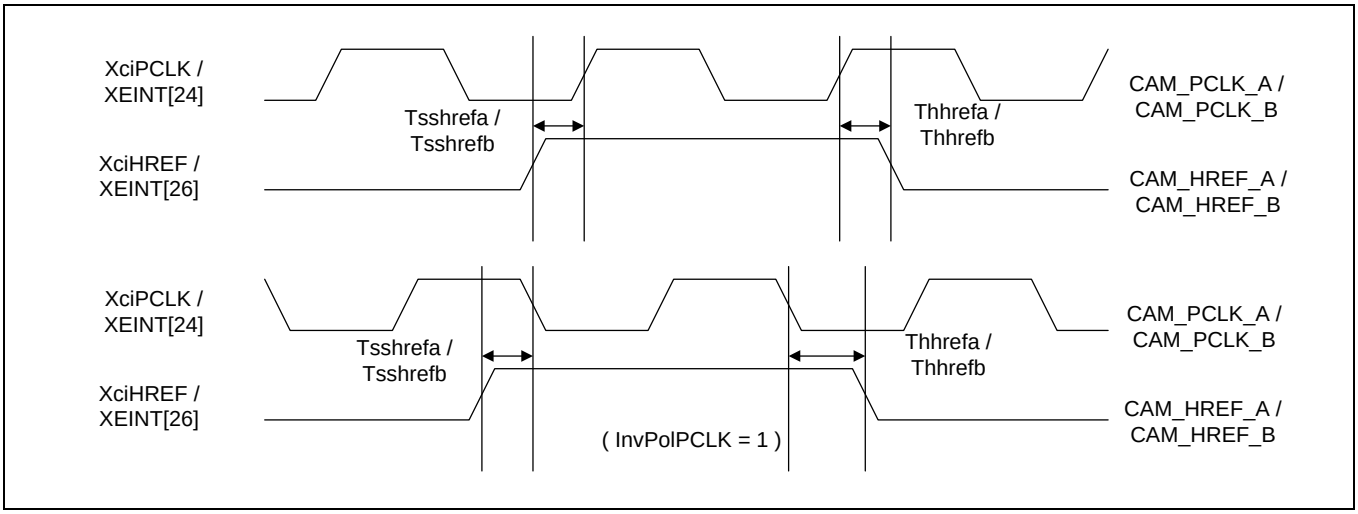


Figure 12.1- 12 Camera Interface HREF Timing

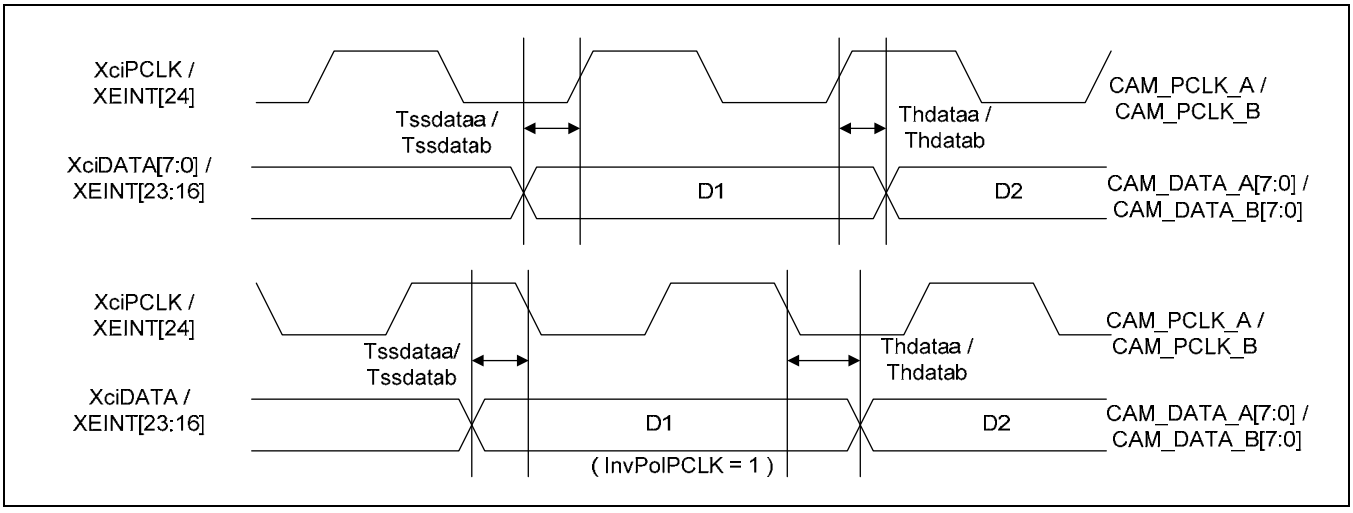


Figure 12.1- 13 Camera Interface Data Timing

Table 12.1- 16 Camera Controller Module Signal Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDext = 1.7V - 3.6V)

Parameter	Symbol	Min	Typ	Max	Units
XciHREF(CAM_HREF_A) input Setup time	$T_{sshrefa}$	3.15	-	PA – 0.59	ns
XciHREF(CAM_HREF_A) input Hold time	T_{hhrefa}	0.59	-	PA – 3.15	ns
XciDATA(CAM_DATA_A) input Setup time	$T_{ssdataa}$	0.75	-	PA – 1.75	ns
XciDATA(CAM_DATA_A) input Hold time	T_{hdataa}	1.75	-	PA – 0.75	ns

NOTE: PA denotes period (ns) of CAM_PCLK_A

Parameter	Symbol	Min	Typ	Max	Units
XEINT[26] (CAM_HREF_B) input Setup time	$T_{sshrefb}$	0.98	-	PA – 2.88	ns
XEINT[26] (CAM_HREF_B) input Hold time	T_{hhrefb}	2.88	-	PB – 0.98	ns
XEINT[23:16] (CAM_DATA_B) input Setup time	$T_{ssdatab}$	0.93	-	PA – 3.78	ns
XEINT[23:16] (CAM_DATA_B) input Hold time	T_{hdatab}	3.78	-	PB – 0.93	ns

NOTE: PB denotes period (ns) of CAM_PCLK_B

11 SDMMC AC ELECTRICAL CHARACTERISTICS

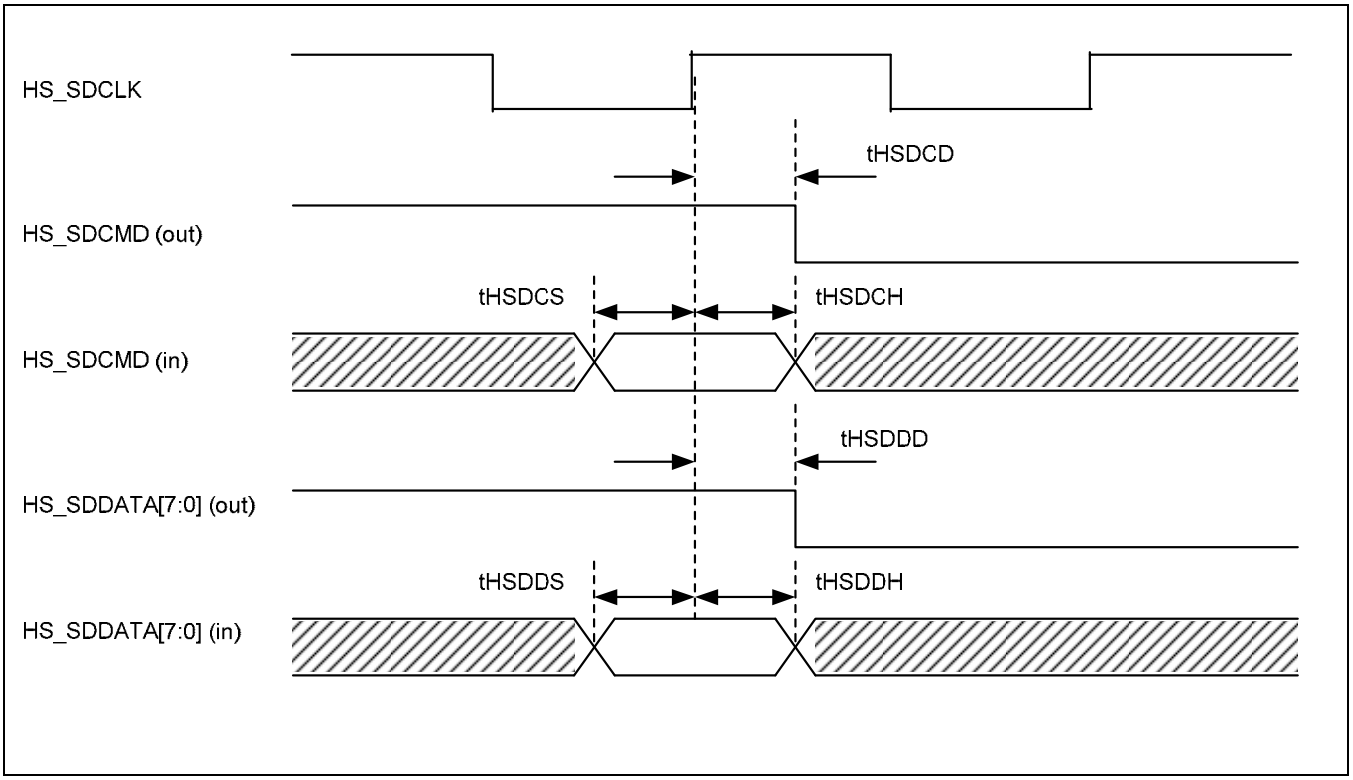


Figure 12.1- 14 High Speed SDMMC Interface Timing

Table 12.1- 17 High Speed SDMMC Interface Transmit/Receive Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDmmc = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min	Type.	Max	Unit
SD Command output Delay time	tSDCD	1.0	–	14.0	ns
SD Command input Setup time	tSDCS	4.0(1)	–	–	ns
SD Command input Hold time	tSDCH	–	–	0.1	ns
SD Data output Delay time	tSDDD	1.0	–	14.0	ns
SD Data input Setup time	tSDDS	4.0(2)	–	–	ns
SD Data input Hold time	tSDDH	–	–	0.1	ns

NOTE (1), (2): This values shows when the Rx Feedback Clock selections are enabled. If the Rx Feedback Clock selection disabled, setup time increases to 14ns (this setting should be used in low speed mode).

12 SPI AC ELECTRICAL CHARACTERISTICS

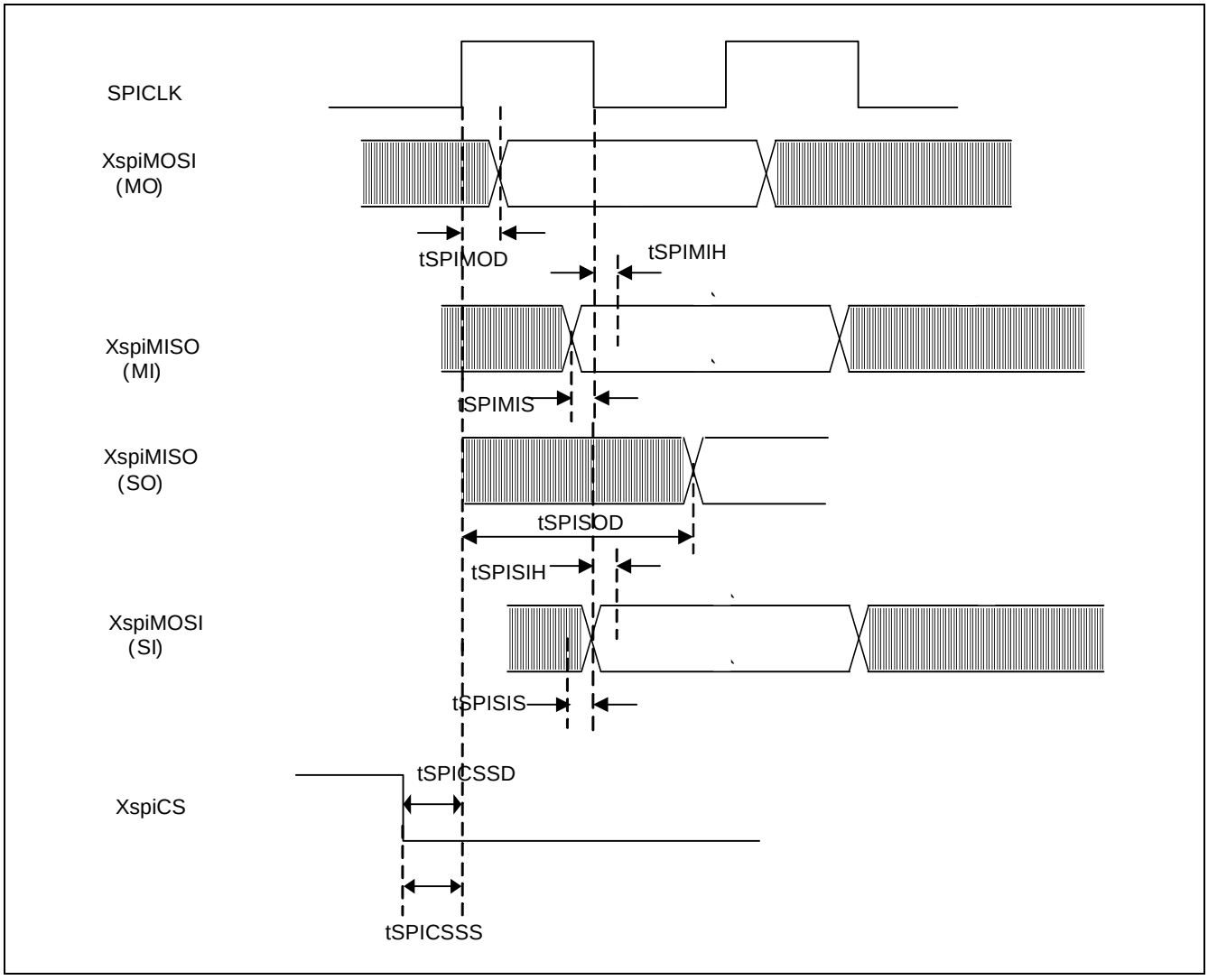


Figure 12.1- 15 SPI Interface Timing (CPHA = 0, CPOL = 1)

Table 12.1- 18 SPI Interface Transmit/ Receive Timing Constants

(VDD_INTNT= 1.2V± 0.05V, TA = -40 to 85°C, VDDext = 3.3V)

Parameter		Symbol	Min	Typ.	Max	Unit
Ch 0	SPI MOSI Master Output Delay time	tSPIMOD	-	-	4	ns
	SPI MISO Master Input Setup time (Feedback Delay- 0nS)	tSPIMIS	3	-	-	ns
	SPI MISO Master Input Setup time(Feedback Delay- 2nS)		2	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 4nS)		1	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 6nS)		1	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 0nS)	tSPIMIH	4	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 2nS)		5	-	-	ns
	SPI MISO Master Master Input Hold time (Feedback Delay- 4nS)		7	-	-	ns
	SPI MISO Master Input Hold time(Feedback Delay- 6nS)		9	-	-	ns
	SPI MOSI Slave Input Setup time	tSPISIS	3	-	-	ns
	SPI MOSI Slave Input Hold time	tSPISIH	3	-	-	ns
	SPI MISO Slave output Delay time	tSPISOD	-	-	10	ns
	SPI nSS Master Output Delay time	tSPICSSD	-	-	23	ns
	SPI nSS Slave Input Setup time	tSPICSSS	-	-	20	ns
Ch 1	SPI MOSI Master Output Delay time	tSPIMOD	-	-	4	ns
	SPI MISO Master Input Setup time (Feedback Delay- 0nS)	tSPIMIS	3	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 2nS)		2	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 4nS)		1	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 6nS)		1	-	-	ns

Parameter		Symbol	Min	Typ.	Max	Unit
	SPI MISO Master Input Hold time (Feedback Delay- 0nS)	tSPIMIH	4	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 2nS)		5	-	-	ns
	SPI MISO Master Master Input Hold time (Feedback Delay- 4nS)		7	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 6nS)		9	-	-	ns
	SPI MOSI Slave Input Setup time	tSPISIS	3	-	-	ns
	SPI MOSI Slave Input Hold time	tSPISIH	3	-	-	ns
	SPI MISO Slave output Delay time	tSPISOD	-	-	10	ns
	SPI nSS Master Output Delay time	tSPICSSD	-	-	23	ns
	SPI nSS Slave Input Setup time	tSPICSSS	-	-	20	ns
Ch 2	SPI MOSI Master Output Delay time	tSPIMOD	-	-	4	ns
	SPI MISO Master Input Setup time (Feedback Delay- 0nS)	tSPIMIS	3	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 2nS)		2	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 4nS)		1	-	-	ns
	SPI MISO Master Input Setup time (Feedback Delay- 6nS)		1	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 0nS)	tSPIMIH	4	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 2nS)		5	-	-	ns
	SPI MISO Master Master Input Hold time (Feedback Delay- 4nS)		7	-	-	ns
	SPI MISO Master Input Hold time (Feedback Delay- 6nS)		9	-	-	ns
	SPI MOSI Slave Input Setup time	tSPISIS	3	-	-	ns

Parameter		Symbol	Min	Typ.	Max	Unit
	SPI MOSI Slave Input Hold time	tSPISIH	3	-	-	ns
	SPI MISO Slave Output Delay time	tSPISOD	-	-	10	ns
	SPI nSS Master Output Delay time	tSPICSSD	-	-	23	ns
	SPI nSS Slave Input Setup time	tSPICSSS	-	-	20	ns

NOTE: SPICLKout = 50MHz

13 I2C AC ELECTRICAL CHARACTERISTICS

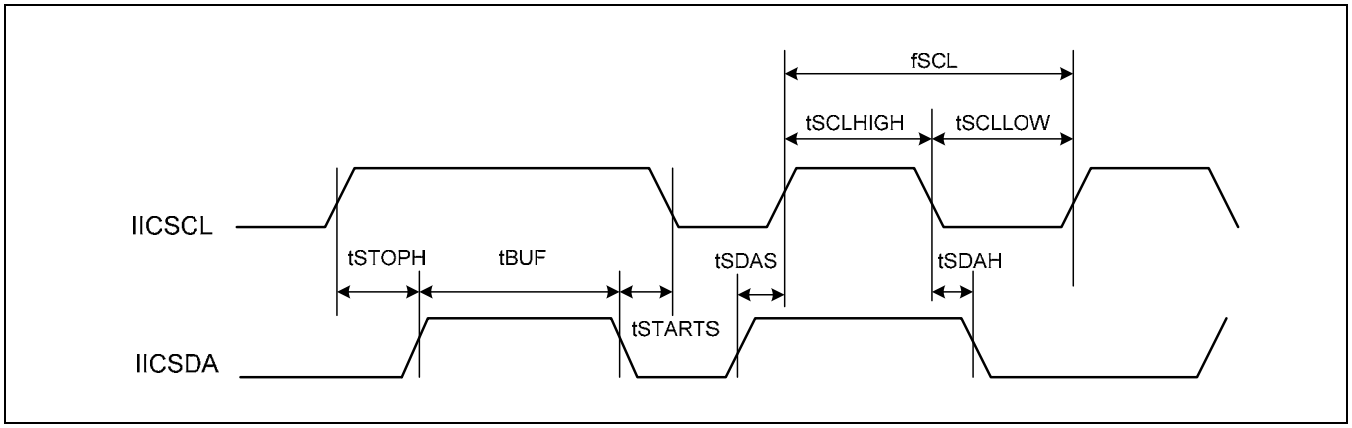


Figure 12.1- 16 IIC Interface Timing

Table 12.1- 19 IIC BUS Controller Module Signal Timing

(VDD_INTNT, VDDarm = 1.2±0.05V , TA = -40 to 85℃, VDDext = 3.3V±0.3V)

Parameter	Symbol	Min	Typ.	Max	Unit
SCL clock frequency	fSCL	-	-	std. 100 fast 400	kHz
SCL high level pulse width	tSCLHIGH	std. 4.0 fast 0.6	-	-	µs
SCL low level pulse width	tSCLLOW	std. 4.7 fast 1.3	-	-	µs
Bus free time between STOP and START	tBUF	std 4.7 fast 1.3	-	-	µs
START hold time	tSTARTS	std. 4.0 fast 0.6	-	-	µs
SDA hold time	tSDAH	std. 0 fast 0	-	std.-fast 0.9	µs
SDA setup time	tSDAS	std. 250 fast 100	-	-	µs
STOP setup time	tSTOPH	std. 4.0 fast 0.6	-	-	µs

NOTES: std. stands for Standard Mode and fast means Fast Mode.

1. The IIC data hold time (tSDAH) is Min 0ns.
(IIC data hold time is Min 0ns for standard/ fast bus mode IIC specification v2.1)
Check whether the data hold time of your IIC device is 0 ns or not.
2. The IIC controller supports IIC bus device only (standard/fast bus mode), and does not support C bus device.

.