SAMSUNG ELECTRONICS

USER'S MANUAL

S3C6410X

RISC Microprocessor

Aug 22, 2008

REV 1.10

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.

S3C6410X RISC Microprocessor
User's Manual, Revision 1.10

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City Gyeonggi-Do, Korea
446-711

Home Page: http://www.samsungsemi.com/

E-Mail: mobilesol.cs@samsung.com

Printed in the Republic of Korea

Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
1.00	- Public Release	-	W.J.Kim	July 18, 2008
1.10	Overview, System Controller, GPIO, CAMIF, IISBUSI/F, Key I/F, IIS Multi Audio I/F, Electrical Data has been modified	Errata Rev00	W.J.Kim	Aug 22, 2008

Table of Contents

Chapter 1 Product Overview

Table of Contents

Chapter 1 Product Overview

Chapter 2 Memory Map

Chapter 3 System Controller

Table of Contents (Continued)

Chapter 3 System Controller

Table of Contents (Continued)

Chapter 4 Memory Sub-system

Chapter 5 DRAM Controller

Table of Contents (Continued)

Chapter 6 SROM Controller

Chapter 7 OneNAND Controller

Table of Contents (Continued)

Chapter 7 OneNAND Controller

Table of Contents (Continued)

Chapter 8 NAND Flash Controller

Table of Contents (Continued)

Chapter 9 CF Controller

Table of Contents (Continued)

Chapter 9 CF Controller

Chapter 10 GPIO

Table of Contents (Continued)

Chapter 11 DMA Controller

Table of Contents (Continued)

Chapter 12 Vectored Interrupt Controllers

Chapter 13 Security SUB-Systems

Table of Contents (Continued)

Chapter 13 Security SUB-Systems

Chapter 14 Display Controller

Table of Contents (Continued)

Chapter 14 Display Controller

Table of Contents (Continued)

Chapter 14 Display Controller

Table of Contents (Continued)

Chapter 15 Post Processor

Table of Contents (Continued)

Chapter 16 TV Scaler

Table of Contents (Continued)

Chapter 17 TV Encoder

Table of Contents (Continued)

Chapter 18 GRAPHICS 2D

Table of Contents (Continued)

Chapter 18 GRAPHICS 2D

Chapter 19 Image Rotator

Table of Contents (Continued)

Chapter 19 Image Rotator

Chapter 20 Camera Interface

Table of Contents (Continued)

Chapter 20 Camera Interface

Table of Contents (Continued)

Chapter 20 Camera Interface

Chapter 21 Multi-FORMAT Video Codec

Table of Contents (Continued)

Chapter 21 Multi-FORMAT Video Codec

Table of Contents (Continued)

Chapter 22 JPEG Codec

Table of Contents (Continued)

Chapter 22 JPEG Codec

Chapter 23 Modem Interface

Table of Contents (Continued)

Chapter 24 Host Interface

Table of Contents (Continued)

Chapter 25 USB HOST Controller

Chapter 26 USB2.0 HS OTG

Chapter 27 SD/MMC HOST Controller

Table of Contents (Continued)

Chapter 27 SD/MMC HOST Controller

Table of Contents (Continued)

Chapter 28 MIPI HSI interface controller

Table of Contents (Continued)

Chapter 29 SPI

Chapter 30 IIC-Bus Interface

Chapter 31 UART

Table of Contents (Continued)

Chapter 31 UART

Chapter 32 PWM Timer

Table of Contents (Continued)

Chapter 32 PWM Timer

Chapter 33 Real Time Clock (RTC)

Table of Contents (Continued)

Chapter 34 WatchDog Timer

Chapter 35 AC97 Controller

Table of Contents (Continued)

Chapter 36 IIS-BUS Interface

Chapter 37 PCM Audio Interface

Table of Contents (Continued)

Chapter 38 IrDA Controller

Chapter 39 ADC & Touch Screen Interface

Table of Contents (Continued)

Chapter 40 KEYPAD Interface

Chapter 41 IIS Multi Audio Interface

Table of Contents (Continued)

Chapter 42 Graphic 3D

Table of Contents (Continued)

Chapter 42 Graphic 3D

Table of Contents (Continued)

Chapter 42 Graphic 3D

Chapter 43 AXI Buses

Table of Contents (Continued)

Chapter 43 AXI Buses

Chapter 44 Electrical Data

Table of Contents (Continued)

List of Figures

List of Figures

List of Figures

List of Figures

List of Figures

List of Figures

List of Figures

List of Figures

List of Figures

List of Figures

List of Tables

List of Tables

List of Tables

List of Tables

List of Tables

1 PRODUCT OVERVIEW

1.1 ARCHITECTURAL OVERVIEW

The S3C6410X is a 16/32-bit RISC microprocessor, which is designed to provide a cost-effective, low-power capabilities, high performance Application Processor solution for mobile phones and general applications. To provide optimized H/W performance for the 2.5G & 3G communication services, the S3C6410X adopts 64/32-bit internal bus architecture. The 64/32-bit internal bus architecture is composed of AXI, AHB and APB buses. It also includes many powerful hardware accelerators for tasks such as motion video processing, audio processing, 2D graphics, display manipulation and scaling. An integrated Multi Format Codec (MFC) supports encoding and decoding of MPEG4/H.263/H.264 and decoding of VC1. This H/W Encoder/Decoder supports real-time video conferencing and TV out for both NTSC and PAL mode. Graphic 3D (hereinafter 3D Engine) is a 3D Graphics Hardware Accelerator which can accelerate OpenGL ES 1.1 & 2.0 rendering. This 3D Engine includes two programmable shaders: one vertex shader and one pixel shader.

The S3C6410X has an optimized interface to external memory. This optimized interface to external memory is capable of sustaining the high memory bandwidths required in high-end communication services. The memory system has dual external memory ports, DRAM and Flash/ROM. The DRAM port can be configured to support mobile DDR, DDR, mobile SDRAM and SDRAM. The Flash/ROM port supports NOR-Flash, NAND-Flash, OneNAND, CF and ROM type external memory.

To reduce total system cost and enhance overall functionality, the S3C6410X includes many hardware peripherals such as a Camera Interface, TFT 24-bit true color LCD controller, System Manager (power management & etc.), 4-channel UART, 32-channel DMA, 5-channel 32bit Timers with 2PWM output, General Purpose I/O Ports, I2S-Bus interface, I2C-BUS interface, USB Host, USB OTG Device operating at high speed (480Mbps), 3-channel SD/MMC Host Controller and PLLs for clock generation.

The ARM subsystem is based on the ARM1176JZF-S core. It includes separate 16KB Instruction and 16KB data caches, 16KB Instruction and 16KB Data TCM. It also includes a full MMU to handle virtual memory management. The ARM1176JZF-S is a single chip MCU, which includes support for JAVA acceleration. The ARM1176JZF-S includes a dedicated vector floating point coprocessor allowing efficient implementation of various encryption schemes as well as high quality 3D graphics applications. The S3C6410X adopts the de-facto standard AMBA bus architecture. These powerful, industry standard features allow the S3C6410X to support many of the industry standard Operating Systems.

By providing a complete set of common system peripherals, the S3C6410X minimizes overall system costs and eliminates the need to configure additional components. The S3C6410X is implemented using an advanced 65nm CMOS process. The low-power, simple, elegant and fully static-design scheme is particularly suitable for cost-sensitive and power-sensitive applications.

We provide two kinds of POP product which is based on S3C6410X.
The 6410X PoP A TYPE is a POP(Package On Package) product stacked with S3C6410X and MCP Memory (2G OneNAND + 512Mb Moblie DDR x 2ea)
The 6410X PoP D TYPE is a POP(Package On Package) product stacked with S3C6410X and MCP Memory (2Gb NAND + 512Mb OneDRAM + 512Mb Mobile DDR)

1.2 FEATURES

This section summarizes the features of the S3C6410X. Figure 1-1 is an overall block diagram of the S3C6410X.

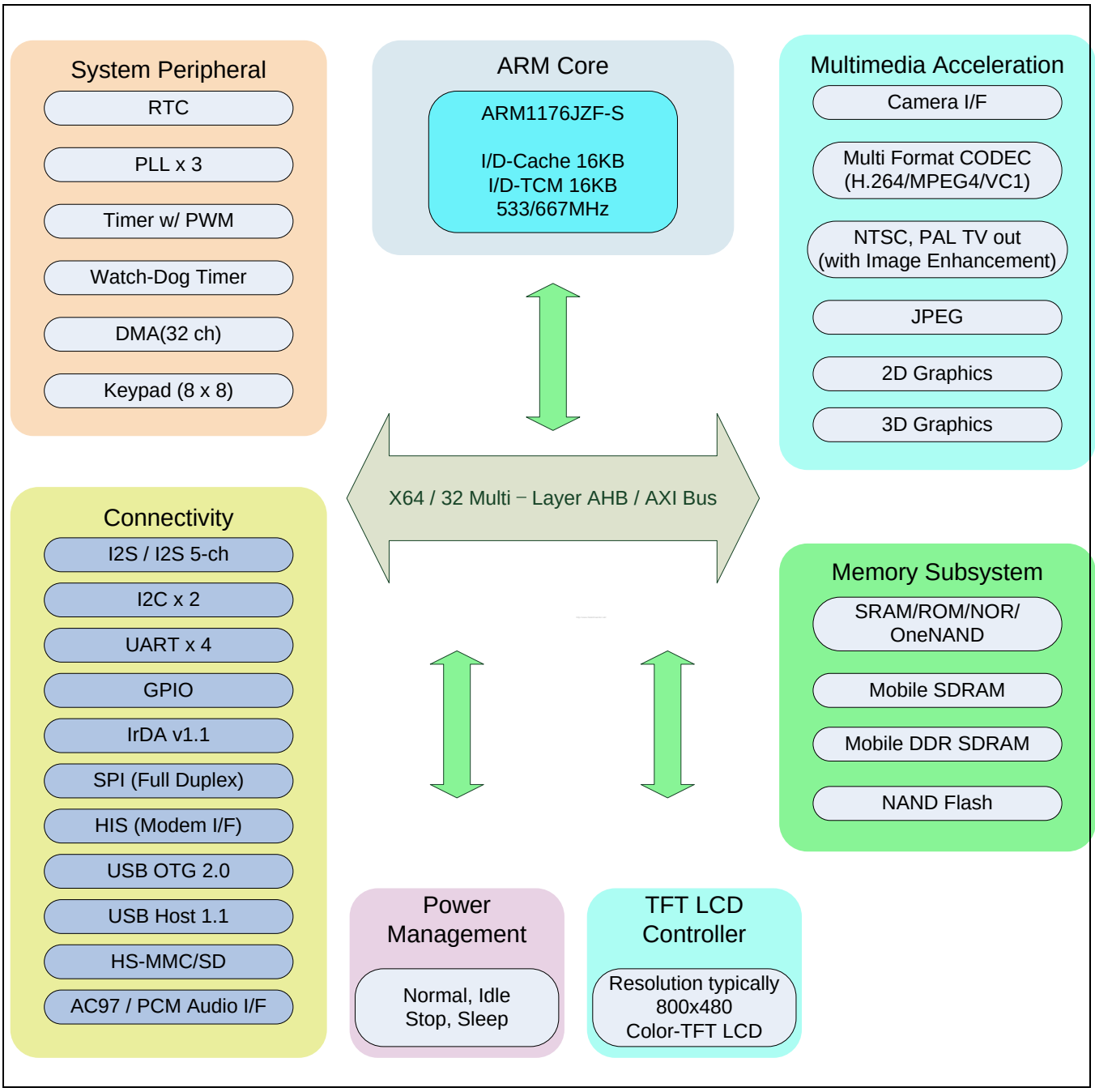


Figure 1-1 S3C6410X Block Diagram

1.2.1 S3C6410X RISC MICROPROCESSOR FEATURES SUMMARY

The features of S3C6410X RISC Microprocessor include:

- The ARM1176JZF-S based CPU subsystem with Java acceleration engine and 16KB/16KB I/D Cache and 16KB/16KB I/D TCM.
- 533MHz at 1.1V and 667MHz at 1.2 V respectively.
- 8-bit ITU 601/656 Camera interface up to 4M pixel for scaled and 16M pixel for unscaled resolution.
- Multi Format Codec provides encoding and decoding of MPEG-4/H.263/H.264 up to 30fps@SD and decoding of VC1 video up to 30fps@SD.
- 2D graphics acceleration with BitBlit and rotation.
- 3D graphics acceleration with 4M triangles/s @133Mhz (Transform only)
- AC-97 audio codec interface and PCM serial audio interface.
- 1/2/4bpp Palletized or 16bpp/24bpp Non-Palletized Color-TFT
- I2S and I2C interface support.
- Dedicated IrDA port for FIR, MIR and SIR.
- Flexibly configurable GPIOs.
- Port USB 2.0 OTG supporting high speed as Device (480Mbps, on-chip transceiver).
- Port USB 1.1 Host supporting full speed (12Mbps, on-chip transceiver).
- SD/MMC/SDIO/CE-ATA Host Controller
- Real time clock, PLL, timer with PWM and watch dog timer.
- 32 channel DMA controller.
- Support 8x8 key matrix.
- Advanced power management for mobile applications.
- Memory Subsystem
  - SRAM/ROM/NOR Flash Interface with x8 or x16 data bus.
  - Muxed OneNAND Interface with x16 data bus.
  - NAND Flash Interface with x8 data bus.
  - SDRAM Interface with x32(Port1) data bus.
  - Mobile SDRAM Interface with x32(Port1) data bus
  - DDR Interface with x32(Port1) data bus
  - Mobile DDR Interface with x32(Port1) data bus

1.2.2 MICROPROCESSOR

The ARM1176JZF-S processor incorporates an integer unit that implements the ARM11 ARM architecture v6. It supports the ARM, Thumb™ instruction sets and Jazelle technology to enable direct execution of Java bytecodes, and a range of SIMD DSP instructions that operate on 16-bit or 8-bit data values in 32-bit registers.

The features of ARM1176JZF-S processor include:

- High-speed *Advanced Microprocessor Bus Architecture* (AMBA) *Advanced Extensible Interface* (AXI) level two interfaces supporting prioritized multiprocessor implementations.
- Integer unit with integral EmbeddedICE-RT logic.
- Eight-stage pipeline.
- Branch prediction with return stack.
- Low interrupt latency configuration.
- coprocessors CP14 and CP15.
- Instruction and Data *Memory Management Units* (MMUs), managed using MicroTLB structures backed by a unified Main TLB.
- Instruction and data caches, including a non-blocking data cache with *Hit-Under-Miss* (HUM).
- Virtually indexed and physically addressed caches.
- 64-bit interface to both caches.
- *Vector Floating-Point* (VFP) coprocessor support.

1.2.3 MEMORY SUBSYSTEM

The S3C6410X microprocessor provides the following Memory Subsystem features:

- High bandwidth Memory Matrix subsystem
- Two independent external memory ports (1 SROM port and 1 DRAM ports)
- Matrix architecture increases overall bandwidth with the simultaneous access capability

1.2.3.1 SROM Memory Port configurable to support the following memory types:

- Support SRAM/ROM/NOR Flash Interface
  - x8 or x16 data bus
  - Address range support: max. 27-bit (128MB)
- Muxed OneNAND Flash interface
  - x16 data bus
  - Support muxed type OneNAND.
- NAND Flash
  - Supports both SLC and MLC NAND Flash memory
- CF interface
  - Compatible with CF+ and CompactFlash Spec (Rev 3.0)

1.2.3.2 DRAM port configurable to support the following memory types:

- SDRAM Interface
  - x16/x32 data bus
  - Density support: up to 2Gb per Bank
- Mobile SDRAM Interface
  - Only x32 data bus with 133Mbps/pin data rate
  - 133MHz address and command bus speed
  - Density support: up to 2Gb per Bank
  - Mobile SDRAM feature support:
    - DS (Driver Strength Control)
    - TCSR (Temperature Compensated Self-Refresh Control)
    - PASR (Partial Array Self-Refresh Control)
- DDR / Mobile DDR Interface
  - x16 or x32 data bus with 266Mbs/pin double data rate (DDR)
  - Density support: up to 2Gb per Bank

1.2.4 MULTIMEDIA ACCELERATION

The S3C6410X microprocessor provides the following Multimedia Acceleration features:

1.2.4.1 Camera Interface

- ITU-R 601/ITU-R 656 format input support. 8-bit input is supported
- Both progressive and interlaced input are supported
- Camera input resolution up to 4096x4096 in YCbCr 4:2:2 format.
  - 4096x4096 input resolution assumes the hardware down-scaling units will be bypass
  - Up to 2048x2048 input resolution can optionally be input to the hardware down-scaling unit
- Resolution down-scaling hardware support for input resolutions up to 2048x2048
- Codec/Preview output image generation (RGB 16/18/24-bit format and YCbCr 4:2:0/4:2:2 format)
- Image windowing and digital zoom-in function
- Test pattern generation
- Image mirror and rotation supports Y-mirror, X-mirror, 90’, 180’ and 270’ rotation
- H/W Color Space Conversion
- LCD controller direct path supported in MSDMA
- Image effect supported.

1.2.4.2 Multi Format Codec (MFC)

- Multi Format Codec
  - MPEG-4 part-II simple profile encoding/decoding
  - H.264/AVC baseline encoding/decoding
  - H.263 profile3 encoding/decoding
  - VC1 decoding
  - Multi-part cell and Multi Standard are supported
- Encoding tools
  - [-16,+16] 1/2 and 1/4 pel accuracy motion estimation using the full-search algorithm
  - Variable block sizes: 16x16, 16x8, 8x16 and 8x8
  - Unrestricted motion vector
  - MPEG-4 AC/DC prediction
  - H.264/AVC intra-prediction (hardwired mode decision)
  - In-loop deblocking filter for both H.264 and H.263 P3
  - Error resilience tools
  - MPEG-4 resync. Marker and data-partitioning with RVLC
  - MPEG-4/AVC FMO
  - Bit-rate control (CBR and VBR)
- Decoding tools
  - Support all features of the standards
- Pre/post rotation/mirroring
  - 8 mirroring/rotation modes

1.2.4.3 JPEG Codec

- Compression/decompression up to UXGA size
- Encoding format: YCbCr 4:2:2 / RGB565
- Decoding format: YCbCr 4:4:4/4:2:2/4:2:0 or gray

1.2.5 2D GRAPHICS ACCELERATOR

The S3C6410X microprocessor supports the following 2D Graphics Accelerator features:

- Line/Point drawing, BitBLT and Color Expansion /Text Drawing

1.2.6 3D GRAPHICS ACCELERATOR

- 4M triangles/s @133MHz (Transform Only)
- 75.8M 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 surfaces (max. 8 user-defined textures)
- API Support: OpenGL ES 1.1 & 2.0, D3D Mobile
- Intelligent Host Interface
  - 15 input data-types, Vertex Buffer & Vertex Cache
- H/W Clipping (Near & Far)
- 8-stage five-threaded Shader architecture
- Primitive assembly & hard-wired triangle setup engine
- One pixels/cycle hard-wired rasterizer
- One texturing engine (one bilinear-filtered texel/cycle each)
  - Nearest/bilinear/trilinear filtering
  - 8-layered multi-texturing support
- Fragment processing: Alpha/Stencil/Z/Dither/Mask/ROP
- Memory bandwidth optimization through hierarchical caching
  - L1/L2 Texture-caches, Z/Color caches
- System bus interface
  - Host interface: 32-bit AHB (AMBA 2.0)
  - Memory Interface: two 64-bit AXI (AMBA 3.0) channels

1.2.6 IMAGE ROTATOR

The S3C6410X microprocessor supports the following Image Rotator features:

- Image format: YCbCr 4:2:2 (interleave), YCbCr 4:2:0 (non-interleave), RGB565 and RGB888(unpacked)
- Rotate degree: 90, 180, 270, flip vertical and flip horizontal
- Image size: 2048x2048

1.2.7 DISPLAY CONTROL

The S3C6410X microprocessor provides the following Display Control features:

1.2.7.1 TFT LCD Interface

- 1/2/4/8bpp Palletized or 16/18/24-bpp Non-Palletized Color-TFF supports
- Typical actual screen size: 640 x 480, 320 x 240, 800 x 480
- Maximum 16M virtual screen size
- Support 5 Window Layer for PIP or OSD
- Realtime overlay plane multiplexing
- Programmable OSC window positioning
- 16-level alpha blending

1.2.7.2 Video Post Processor

- Video input format conversion
- Video/Graphic scaling up/down or zooming in/out
- Color space conversion from YCbCr to RGB and from RGB to YCbCr
- Dedicated local interface for display
- Dedicated scaler for TV Encoder

1.2.7.3 TV (NTSC/PAL) Video Encoder with Image Enhancer

- Support NTSC-M,J / PAL-B,D,G,H,I,M,Nc compliant video format
- Built in the MIE(Mobile Image Enhancer) Engine
  - Black & White Stretch
  - Blue Stretch & Flesh-Tone Correction
  - Dynamic Horizontal Peaking & LTI
  - Black and White Noise reduction
  - Full Size and Wide Size Video-Out

1.2.8 AUDIO INTERFACE

The S3C6410X microprocessor provides the following Audio Interface features:

1.2.8.1 AC97 Controller

- Variable sampling rate (48kHz and below)
- Single Codec Controller
- 1 port stereo inputs/1 port stereo outputs/mono MIC input
- 16-bit stereo (2-channel) audio

1.2.8.2 PCM serial Audio Interface

- Master mode bi-directional serial audio interface
- Accepts an external input clock to generate exact Audio timings
- Optional DMA-based operation

1.2.8.3 IIS-Bus Interface

- 2 ch IIS-bus 2ea and 5.1ch IIS-bus 1ea for the audio-codec interface
- Optional DMA-based operation
- Serial, 8/16/24-bit per channel data transfers
- Supports IIS, MSB-justified and LSB-justified data format
- Can operate in Master or Slave mode
- Various bit clock frequency and codec clock frequency support
  - 16,24,32,48 fs of bit clock frequency and 256,384,512,768 fs of codec clock frequency
- Supports 8kHz ~ 192kHz sampling frequency.

1.2.9 USB SUPPORT

The S3C6410X microprocessor provides the following USB Support features:

1.2.9.1 USB OTG 2.0 High Speed

- Complies with the On-The-Go Supplement to the USB 2.0 Specification (Revision 1.0a)
- Supports high speed (480Mbps), full speed (12Mbps, Device only), low speed (1.5Mbps, Host only)
- Configures as USB 1.1 full/low speed DRD(Dual-Role Device), Host-only or Device only controller

1.2.9.2 USB Host

- 2-port USB Host
- Complies with OHCI Rev. 1.0
- Compatible with the USB Specification version 1.1
- Supports full speed up to 12Mbps

1.2.10 IRDA V1.1

The S3C6410X microprocessor provides the following IrDA v1.1 features:

- Dedicated IrDA for v1.1 (1.152Mbps and 4 Mbps)
- Supports FIR (4Mbps),
- Internal 64-byte Tx/Rx FIFO

1.2.11 SERIAL COMMUNICATION

The S3C6410X microprocessor provides the following Serial Communication features:

1.2.11.1 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 (EXT_UCLK0, refer to UART chapter)
- Programmable baud rate
- Supports IrDA 1.0 SIR (115.2Kbps) mode
- Loop back mode for testing
- Each channel has internal 64-byte Tx FIFO and 64-byte Rx FIFO
- Non-integer clock divides in Baud clock generation

1.2.11.2 IIC-Bus Interface

- 2-ch Multi-Master IIC-Bus
- Serial, 8-bit oriented and bi-directional data transfers can be made at up to 100 Kbps in the standard mode
- up to 400 Kbps in the fast mode

1.2.11.3 SPI Interface

- 2ch Serial Peripheral Interface
- 64byte buffers for receive/transmit
- DMA-based or interrupt-based operation
- 50Mbps Master Tx/Rx, 50Mbps Slave Rx, 20Mbps Slave Tx

1.2.11.4 MIPI HSI

- A uni-direction high speed serial interface
- Supports Tx and Rx
- 128 Byte (32-bit x 32) Tx FIFO
- 256 Byte (32-bit x 64) Rx FIFO
- TX : PCLK bps, RX : up to 100Mbps

1.2.12 MODEM & HOST INTERFACE

The S3C6410X microprocessor provides the following Modem Interface features:

1.2.12.1 Parallel Modem Chip Interface

- Asynchronous direct SRAM interface style interface
- 16-bit parallel bus for data transfer
- On-chip 8K bytes internal dual-port SRAM buffer
- Interrupt request for data exchange
- Programmable interrupt port address
- Support from 1.8V to 3.3V I/O voltage range
- AP Booting for Modem procedure provides a dual-port memory as a Modem boot memory.

1.2.12.1 Host Interface

- Asynchronous indirect SRAM interface style interface (i80 interface)
- 16-bit protocol register
- On-chip Write FIFO and Read FIFO (each 288-word) to support indirect burst transfer
- Single R/W on the SFR/memory in the system memory map
- Burst R/W on the SFR/memory in the system memory map
- Repeated Burst Write on the SFR/memory in the system memory map
- Supports Modem Booting that enables HOST to control AP boot.

1.2.13 GPIO (GENERAL PURPOSE I/O)

The S3C6410X microprocessor provides the following GPIO features:

- 187 Flexibly configurable GPIO

1.2.14 INPUT DEVICES

The S3C6410X microprocessor provides the following Input Devices features:

1.2.14.1 Keypad Interface

- Support 8x8 Key Matrix
- Provides internal debounce filter

1.2.14.2 A/D Converter and Touch Screen Interface

- 8-ch multiplexed ADC
- Max. 1M samples/sec and 10-bit/12-bit resolution

1.2.15 STORAGE DEVICES

The S3C6410X microprocessor provides the following Storage Devices features:

1.2.15.1 SD/MMC Host Controller

- Multimedia Card Protocol version 4.0 compatible
- SD Memory Card Protocol version 2.0 compatible
- SDIO Card Protocol version 1.0 compatible
- 128 words FIFO for Tx/Rx
- DMA based or Interrupt based operation
- 3 channel SD/MMC Host Controller
- Support CE-ATA interface

1.2.16 SYSTEM PERIPHERALS

The S3C6410X microprocessor provides the following System Peripherals features:

1.2.16.1 DMA controller

- 4 General DMAs embedded.
- 8-channel supported per each DMA; totally 32 channel is supported.
- Supports memory to memory, peripheral to memory, memory to peripheral, and peripheral to peripheral
- Burst transfer mode to enhance the transfer rate.

1.2.16.2 Vectored Interrupt Controller

- Supports for 64 vectored IRQ interrupts
- Fixed hardware interrupt priority levels
- Programmable interrupt priority levels
- Hardware interrupt priority level masking
- IRQ and FIQ generation
- Raw interrupt status
- Interrupt request status
- Privileged mode support for restricted access
- Support for ARM v6 processor VIC port, enabling faster interrupt servicing

1.2.16.3 TrustZone Protection Controller

- Provides a software interface to the protection bits in a secure system in a TrustZone design
- Protection bits to enable programming of up to 24 areas of memory as secure or non-secure
- AMBA APB interface

1.2.16.4 Timer with PWM (Pulse Width Modulation)

- 5-channel 32bit Timers with 2 PWM output
- Programmable duty cycle, frequency, and polarity
- Dead-zone generation
- Support external clock source

1.2.16.5 16-bit Watchdog Timer

- Interrupt request or system reset at time-out

1.2.16.6 RTC (Real Time Clock)

- Full clock features: msec, sec, min, hour, day, week, month, year
- 32.768kHz operation
- Alarm interrupts
- Time-tick interrupts

1.2.17 SECURITY SUB-SYSTEM

- AES accelerator : ECB, CBC, CTR mode support
- DES/3DES accelerator : ECB, CBC mode support
- SHA-1 Hash engine
- H/W HMAC support
- Random Number Generator : PRNG 320-bit generation per 160 cycles
- FIFO-Rx/Tx : (two 32-word) for input and output streaming.
- DMA I/F to SDMA1(Security DMA 1)

1.2.18 SYSTEM MANAGEMENT

The S3C6410X microprocessor provides the following System Management features:

- Little Endian format support

1.2.19 SYSTEM OPERATING FREQUENCIES

The S3C6410X microprocessor provides the following System Operation Frequencies features:

- ARM1176JZF-S core clock rate maximum is 533MHz@1.1 V, 667MHz@1.2V (VDDarm)
- System operating clock generation
  - Three on-chip PLLs, APLL, MPLL & EPLL
  - APLL generates an independent ARM operating clock
  - MPLL generates the system reference clock
  - EPLL generates clocks for peripheral IPs

1.3 POWER MANAGEMENT

- Clock-off control for individual components
- Various power-down modes are available such as Idle, Stop and Sleep mode
- Wake-up by one of external interrupts or by various interrupt sources. Refer system controller manual.

1.4 ELECTRICAL CHARACTERISTICS

- Operating Conditions
  - Supply Voltage for Logic Core: VDD_INT 1.2V, VDD_ARM depends on Operation Frequency
  - Memory Port 0(VDDM0) : 1.8/2.5V/3.3V
  - Memory Port1 (VDDM1) : 1.8/2.5V
  - External I/O Interface: 1.8/2.5/3.3V
- Operational Frequency(VDDarm)
  - 533MHz@ 1.1 V
  - 667Mhz@ 1.2V

1.5 PACKAGE

- 424-Pin FBGA (13mm x 13mm)

1.6 PIN ASSIGNMENTS

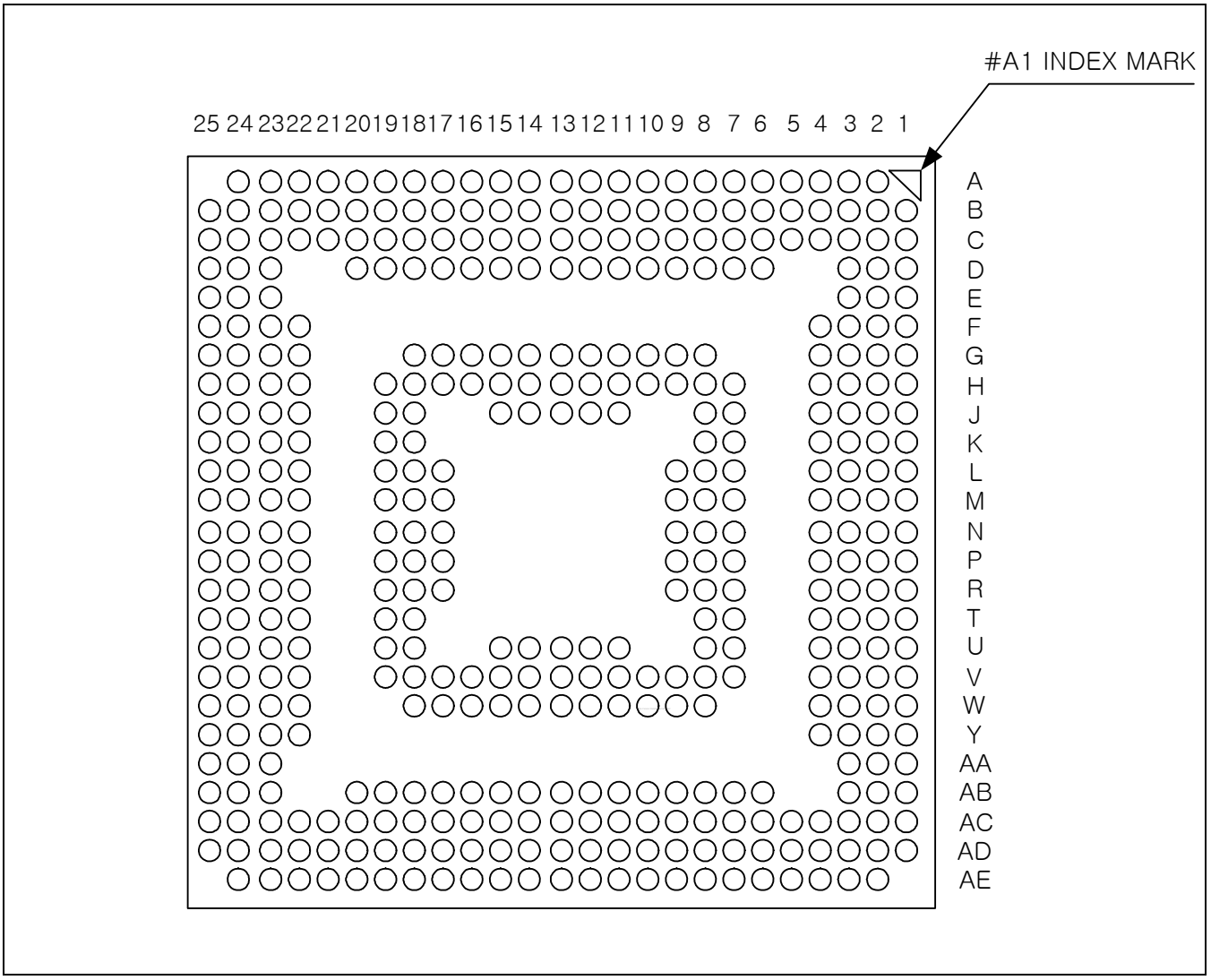


Figure 1-2. Pin Assignments

Table 1-1. 424-Pin FBGA Pin Assignments – Pin Number Order

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A2	NC_C	B3	XPCMEXTCLK1/GPE1	C3	XPCMSOUT1/GPE4
A3	XPCMSOUT0/GPD4	B4	XPCMSIN0/GPD3	C4	XPCMFSYNC1/GPE2
A4	VDDPCM	B5	XPCMEXTCLK0/GPD1	C5	XPCMDCLK1/GPE0
A5	XM1DQM0	B6	XM1DATA0	C6	XM1DATA4
A6	XM1DATA1	B7	XM1DATA3	C7	XM1DATA2
A7	VDDINT	B8	VDDM1	C8	XM1DATA5
A8	VDDARM	B9	VDDM1	C9	XM1DATA7
A9	XM1DATA6	B10	XM1DATA13	C10	VDDARM
A10	XM1DATA9	B11	VDDARM	C11	XM1DATA14
A11	XM1DATA12	B12	XM1DATA16	C12	XM1DATA10
A12	XM1DATA18	B13	XM1DATA17	C13	XM1DATA19
A13	XM1SCLK	B14	XM1DQS2	C14	VDDM1
A14	XM1SCLKN	B15	XM1DATA22	C15	XM1DATA20
A15	XMMCDATA1_4/GHP6	B16	XMMCDATA1_2/GPH4	C16	XMMCDATA1_6/GPH8
A16	XMMCCMD1/GPH1	B17	VDDMMC	C17	XMMCDATA1_1/GPH3
A17	XMMCCDN0/GPG6	B18	XMMCDATA0_0/GPG2	C18	XMMCDATA0_2/GPG4
A18	XMMCCLK0/GPG0	B19	XSPIMISO1/GPC4	C19	XSPIMOSI1/GPC6
A19	XSPIMOSI0/GPC2	B20	XSPIMISO0/GPC0	C20	XSPICS0/GPC3
A20	XI2CSCL/GPB5	B21	XUTXD3/GPB3	C21	VDDEXT
A21	XUTXD2/GPB1	B22	XUTXD1/GPA5	C22	XURTSN1/GPA7
A22	XURTSN0/GPA3	B23	XCIYDATA7/GPF12	C23	XPWMECLK/GPF13
A23	XUTXD0/GPA1	B24	XCIYDATA5/GPF10	C24	XCIYDATA2/GPF7
A24	NC_D	B25	NC_F	C25	XCIYDATA0/GPF5
B1	NC_B	C1	XM0ADDR0	D1	XM0ADDR2
B2	XPCMSIN1/GPE3	C2	VDDARM	D2	XM0ADDR3

Table 1-1. 424-Pin FBGA Pin Assignments – Pin Number Order (Continued)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
D3	VDDARM	F1	XM0ADDR8/GPO8	G24	XM1DATA27
D6	XPCMFSYNC0/GPD2	F2	XM0ADDR6/GPO6	G25	XM1DATA30
D7	XPCMDCLK0/GPD0	F3	VDDARM	H1	VDDINT
D8	VDDARM	F4	VDDM0	H2	XM0ADDR13/GPO13
D9	XM1DQS0	F22	XCIPCLK/GPF2	H3	XM0ADDR15/GPO15
D10	XM1DATA15	F23	XM1DATA24	H4	XM0ADDR12/GPO12
D11	XM1DATA11	F24	XM1DATA25	H7	XM0ADDR4
D12	XM1DATA8	F25	XM1DATA26	H8	VSSIP
D13	VDDINT	G1	XM0ADDR11/GPO11	H9	XMMCDATA1_7/GPH9
D14	XM1DQM2	G2	XM0ADDR10/GPO10	H10	XMMCDATA1_3/GPH5
D15	XM1DATA21	G3	VDDM0	H11	XMMCDATA1_0/GPH2
D16	XM1DATA23	G4	XM0ADDR7/GPO7	H12	XSPICLK1/GPC5
D17	XSPICS1/GPC7	G8	XM1DQM1	H13	XMMCDATA0_1/GPG3
D18	VDDINT	G9	XM1DQS1	H14	XSPICLK0/GPC1
D19	XURXD2/GPB0	G10	VDDM1	H15	XUCTSN1/GPA6
D20	XURXD0/GPA0	G11	XMMCDATA1_5/GPH7	H16	XPWMTOUT0/GPF14
D23	XPWMTOUT1/GPF15	G12	XMMCDATA0_3/GPG5	H17	XCIYDATA4/GPF9
D24	XCIVSYNC/GPF4	G13	XMMCCMD0/GPG1	H18	VSSPERI
D25	XCIHREF/GPF1	G14	XI2CSDA/GPB6	H19	XCIRSTN/GPF3
E1	XM0ADDR5	G15	XIRSDBW/GPB4	H22	XM1DQM3
E2	VDDARM	G16	XUCTSN0/GPA2	H23	XM1DATA31
E3	XM0ADDR1	G17	XCIYDATA6/GPF11	H24	XM1ADDR0
E23	XCIYDATA1/GPF6	G18	XCIYDATA3/GPF8	H25	XM1ADDR3
E24	XM1DATA28	G22	XCICLK/GPF0	J1	XM0ADDR16/GPQ8
E25	XM1DQS3	G23	XM1DATA29	J2	XM0WEN

Table 1-1. 424-Pin FBGA Pin Assignments – Pin Number Order (Continued)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
J3	VDDARM	K24	XM1ADDR12	M19	XM1WEN
J4	XM0ADDR14/GPO14	K25	XM1ADDR5	M22	VDDINT
J7	VSSMEM	L1	XM0BEN0	M23	XM1ADDR10
J8	XM0ADDR9/GPO9	L2	XM0DATA13	M24	XM1CKE1
J11	XMMCCLK1/GPH0	L3	XM0SMCLK/GPp1	M25	XHIDATA17/GPL14
J12	VSSIP	L4	XM0OEN	N1	XM0DATA1
J13	VSSPERI	L7	XM0DATA10	N2	XM0DATA0
J14	XURXD3/GPB2	L8	XM0DATA12	N3	XM0DATA3
J15	XURXD1/GPA4	L9	VSSIP	N4	XM0DATA6
J18	VDDINT	L17	VDDINT	N7	XM0CSN0
J19	VDDM1	L18	XM1CSN1	N8	XM0CSN5/GPO3
J22	XM1ADDR9	L19	XM1ADDR4	N9	VSSIP
J23	XM1ADDR2	L22	XM1RASN	N17	XHIDATA16/GPL13
J24	XM1ADDR1	L23	XM1CSN0	N18	XHIDATA14/GPK14
J25	XM1ADDR6	L24	XM1CASN	N19	VDDUH
K1	XM0DATA15	L25	XM1ADDR15	N22	XUHDP
K2	VDDM0	M1	VDDM0	N23	XHIDATA15/GPK15
K3	VDDARM	M2	XM0DATA8	N24	XHIDATA13/GPK13
K4	XM0DATA14	M3	XM0DATA11	N25	XHIDATA12/GPK12
K7	XM0BEN1	M4	XM0DATA9	P1	VDDINT
K8	VSSIP	M7	XM0DATA2	P2	XM0DATA5
K18	XM1ADDR7	M8	XM0DATA4	P3	XM0DATA7
K19	XM1ADDR11	M9	VSSMEM	P4	XM0CSN2/GPO0
K22	XM1ADDR13	M17	XM1ADDR14	P7	GPO5
K23	XM1ADDR8	M18	XM1CKE0	P8	XM0ADDR19/GPQ1

Table 1-1. 424-Pin FBGA Pin Assignments – Pin Number Order (Continued)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
P9	VSSSS	T4	GPQ5	U25	XHIADR8/GPL8
P17	VSSIP	T7	XEFFVDD	V1	VDDSS
P18	XHIDATA11/GPK11	T8	VSSMPLL	V2	GPQ6
P19	XHIDATA9/GPK9	T18	XHIADR7/GPL7	V3	GPQ4
P22	XUHDN	T19	XHIADR9/GPL9	V4	XM0WEATA/GPP12
P23	XHIDATA10/GPK10	T22	XHIDATA1/GPK1	V7	VSSEPLL
P24	VDDHI	T23	XHIDATA3/GPK3	V8	XOM3
P25	XHIDATA8/GPK8	T24	XHIDATA2/GPK2	V9	XNRESET
R1	VDDM0	T25	XHIDATA0/GPK0	V10	XEINT1/GPN1
R2	XM0CSN3/GPO1	U1	GPQ3	V11	XEINT6/GPN6
R3	XM0CSN1	U2	XM0ADDR18/GPQ0	V12	XEINT12/GPN12
R4	XM0WAITN/GPP2	U3	XM0ADDR17/GPQ7	V13	XVVD3/GPI3
R7	XM0INTATA/GPP8	U4	XM0INTSM1_FREN/GPP6	V14	XVVD8/GPI8
R8	XM0RDY0_ALE/GPP3	U7	XM0CDATA/GPP14	V15	XVVD12/GPI12
R9	VSSIP	U8	VSSMEM	V16	XVVD16/GPJ0
R17	VSSPERI	U11	VSSPERI	V17	VSSPERI
R18	VDDALIVE	U12	VSSPERI	V18	XHICSN_MAIN/GPM1
R19	XHIADR12/GPL12	U13	VSSIP	V19	XVVCLK/GPJ11
R22	XHIDATA5/GPK5	U14	VSSPERI	V22	XHIOEN/GPM4
R23	XHIDATA4/GPK4	U15	VDDALIVE	V23	XHIADR6/GPL6
R24	XHIDATA6/GPK6	U18	XHIADR2/GPL2	V24	VDDHI
R25	XHIDATA7/GPK7	U19	XHIADR0/GPL0	V25	XHIADR5/GPL5
T1	GPQ2	U22	XHIADR4/GPL4	W1	VDDINT
T2	GPO4	U23	XHIADR11/GPL11	W2	XM0RDY1_CLE/GPP4
T3	XM0CSN4/GPO2	U24	XHIADR10/GPL10	W3	XM0RESETATA/GPP9

Table 1-1. 424-Pin FBGA Pin Assignments – Pin Number Order (Continued)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
W4	VSSAPLL	AA2	XM0INPACKATA/GPP10	AB25	XVVD20/GPJ4
W8	VSSMEM	AA3	XM0REGATA/GPP11	AC1	XADCAIN0
W9	XOM1	AA23	XHICSN/GPM0	AC2	XADCAIN1
W10	VDDALIVE	AA24	XVDEN/GPJ10	AC3	XADCAIN7
W11	XEXTCLK	AA25	XVHSYNC/GPJ8	AC4	VDDADC
W12	XEINT8/GPN8	AB1	VDDEPLL	AC5	VSSDAC
W13	XEINT14/GPN14	AB2	VDDMPLL	AC6	XDACOUT0
W14	XVVD1/GPI1	AB3	XM0OEATA/GPP13	AC7	XDACCOMP
W15	XVVD6/GPI6	AB6	VSSMEM	AC8	XUSBREXT
W16	XVVD11/GPI11	AB7	VSSOTG	AC9	VDDOTG
W17	XVVD14/GPI14	AB8	VSSOTGI	AC10	VDDOTGI
W18	XVVD22/GPJ6	AB9	XRTCXTI	AC11	VDDRTC
W22	XVVSYNC/GPJ9	AB10	XJTRSTN	AC12	XJTDO
W23	XHIADR3/GPL3	AB11	XJTCK	AC13	XOM2
W24	XHIADR1/GPL1	AB12	XJTDI	AC14	VSSPERI
W25	XHIIRQN/GPM5	AB13	XJDBGSEL	AC15	VDDSYS
Y1	XM0RPN_RNB/GPP7	AB14	XXTO27	AC16	XXTI
Y2	XM0ADRVALID/GPP0	AB15	XXTI27	AC17	XXTO
Y3	XM0INTSM0_FWEN/GPP5	AB16	XSELNAND	AC18	XEINT5/GPN5
Y4	XPLLEFILTER	AB17	XEINT3/GPN3	AC19	XEINT7/GPN7
Y22	XVVD18/GPJ2	AB18	XEINT10/GPN10	AC20	VDDINT
Y23	XHIWEN/GPM3	AB19	VDDALIVE	AC21	XVVD9/GPI9
Y24	XHICSN_SUB/GPM2	AB20	XVVD5/GPI5	AC22	XVVD10/GPI10
Y25	VDDINT	AB23	XVVD23/GPJ7	AC23	VDDLCD
AA1	VDDAPLL	AB24	XVVD21/GPJ5	AC24	XVVD15/GPI15

Table 1-1. 424-Pin FBGA Pin Assignments – Pin Number Order (Continued)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
AC25	XVVD19/GPJ3	AD25	NC_I		
AD1	NC_G	AE2	NC_H		
AD2	XADCAIN2	AE3	XADCAIN4		
AD3	XADCAIN3	AE4	XADCAIN6		
AD4	XADCAIN5	AE5	XDACOUT1		
AD5	VSSADC	AE6	XDACIREF		
AD6	VDDDAC	AE7	XDACVREF		
AD7	XUSBXTI	AE8	VSSOTG		
AD8	XUSBXTO	AE9	XUSBDM		
AD9	XUSBVBUS	AE10	XUSBDP		
AD10	XUSBID	AE11	XUSBDRVVBUS		
AD11	VDDOTG	AE12	XJTMS		
AD12	XRTCXTO	AE13	XJRTCK		
AD13	XOM0	AE14	XOM4		
AD14	XPWRRGTON	AE15	XNBATF		
AD15	WR_TEST	AE16	VDDINT		
AD16	XNRSTOUT	AE17	XEINT0/GPN0		
AD17	XEINT2/GPN2	AE18	XEINT4/GPN4		
AD18	VDDSYS	AE19	XEINT9/GPN9		
AD19	XEINT11/GPN11	AE20	XEINT13/GPN13		
AD20	XEINT15/GPN15	AE21	XVVD0/GPI0		
AD21	XVVD4/GPI4	AE22	XVVD2/GPI2		
AD22	VDDLCD	AE23	XVVD7/GPI7		
AD23	XVVD13/GPI13	AE24	NC_J		
AD24	XVVD17/GPJ1				

Pin Name	Default Function	I/O state @Sleep	I/O state @Reset	I/O Type
XOM0		-	I	hag
XOM1		-	I	hag
XOM2		-	I	hag
XOM3		-	I	hag
XOM4		-	I	hag
XEXTCLK	EXTCLK	-	I	hag
XXTI	XTI	-	I	scb
XXTI27	XTI27	-	I	scb
XXTO	XTO	-	O	scb
XXTO27	XTO27	-	O	scb
XRTCXTI	RTCXTI	-	I	sca
XRTCXTO	RTCXTO	-	O	sca
XNRESET	NRESET	-	I	hag
WR_TEST	WR_TEST	-	I	hag
XNRSTOUT	NRSTOUT	-	O(L)	hag_a
XJDBGSEL	JDBGSEL	-	I	hag
XJRTCK	JRTCK	O	O(L)	hag
XJTCK	JTCK	-	I	hag
XJTDI	JTDI	-	I	hag
XJTDO	JTDO	O	I	hag
XJTMS	JTMS	-	I	hag
XJTRSTN	JTRSTN	-	I	hag
XSELNAND		-	I	hag
XNBATF		-	I	hag
XPWRRGTON		O(L)	O(L)	hag
XM0CSN0	XM0CSN0	-	O(H)	hbg
XM0CSN1	XM0CSN1	-	O(H)	hbg
XM0CSN2/GPO0	XM0CSN2	-	O(H)	hbg
XM0CSN3/GPO1	XM0CSN3	-	O(H)	hbg
XM0CSN4/GPO2	XM0CSN4	-	O(H)	hbg
XM0CSN5/GPO3	XM0CSN5	-	O(H)	hbg
GPO4	Reserved	-	O(H)	hbg
GPO5	Reserved	-	O(H)	hbg
XM0ADRVALID/GPP0	XM0ADRVALID	-	O(L)	hbg
XM0SMCLK/GPP1	XM0SMCLK	-	O(H)	hbg

XM0WAITN/GPP2	XM0WAITN	-	I	hbg
XM0RDY0_ALE/GPP3	XM0RDY0_ALE	-	I/O(L)	hbg
XM0RDY1_CLE/GPP4	XM0RDY1_CLE	-	I/O(L)	hbg
XM0INTSM0_FWEN/GPP5	XM0INTSM0_FWEN	-	I/O(H)	hbg
XM0INTSM1_FREN/GPP6	XM0INTSM1_FREN	-	I/O(H)	hbg
XM0RPN_RNB/GPP7	XM0RPN_RNB	-	O(L)/I	hbg
XM0INTATA/GPP8	XM0INTATA	-	I	hbg
XM0RESETATA/GPP9	XM0RESETATA	-	O(H)	hbg
XM0INPACKATA/GPP10	XM0INPACKATA	-	I	hb_c
XM0REGATA/GPP11	XM0REGATA	-	O(H)	hbg
XM0WEATA/GPP12	XM0WEATA	-	O(H)	hbg
XM0OEATA/GPP13	XM0OEATA	-	O(H)	hbg
XM0CDATA/GPP14	XM0CDATA	-	I	hbg
XM0ADDR18/GPQ0	XM0ADDR18	-	O(H)	hbg
XM0ADDR19/GPQ1	XM0ADDR19	-	O(H)	hbg
GPQ2	Reserved	-	O(L)	hbg
GPQ3	Reserved	-	O(H)	hbg
GPQ4	Reserved	-	O(H)	hbg
GPQ5	Reserved	-	I	hbg
GPQ6	Reserved	-	I	hbg
XM0ADDR17/GPQ7	XM0ADDR17	-	O(H)	hbg
XM0ADDR16/GPQ8	XM0ADDR16	-	O(L)	hbg
XM0OEN	XM0OEN	-	O(H)	hbg
XM0WEN	XM0WEN	-	O(H)	hbg
XM0BEN0	XM0BEN0	-	O(L)	hbg
XM0BEN1	XM0BEN1	-	O(L)	hbg
XM0ADDR0	XM0ADDR0	-	O(L)	hbg
XM0ADDR1	XM0ADDR1	-	O(L)	hbg
XM0ADDR2	XM0ADDR2	-	O(L)	hbg
XM0ADDR3	XM0ADDR3	-	O(L)	hbg
XM0ADDR4	XM0ADDR4	-	O(L)	hbg
XM0ADDR5	XM0ADDR5	-	O(L)	hbg
XM0ADDR6/GPO6	XM0ADDR6	-	O(L)	hbg
XM0ADDR7/GPO7	XM0ADDR7	-	O(L)	hbg
XM0ADDR8/GPO8	XM0ADDR8	-	O(L)	hbg
XM0ADDR9/GPO9	XM0ADDR9	-	O(L)	hbg
XM0ADDR10/GPO10	XM0ADDR10	-	O(L)	hbg

XM0ADDR11/GPO11	XM0ADDR11	-	O(L)	hbg
XM0ADDR12/GPO12	XM0ADDR12	-	O(L)	hbg
XM0ADDR13/GPO13	XM0ADDR13	-	O(L)	hbg
XM0ADDR14/GPO14	XM0ADDR14	-	O(L)	hbg
XM0ADDR15/GPO15	XM0ADDR15	-	O(L)	hbg
XM0DATA0	XM0DATA0	-	I	hbg
XM0DATA1	XM0DATA1	-	I	hbg
XM0DATA2	XM0DATA2	-	I	hbg
XM0DATA3	XM0DATA3	-	I	hbg
XM0DATA5	XM0DATA5	-	I	hbg
XM0DATA6	XM0DATA6	-	I	hbg
XM0DATA7	XM0DATA7	-	I	hbg
XM0DATA8	XM0DATA8	-	I	hbg
XM0DATA4	XM0DATA8	-	I	hbg
XM0DATA9	XM0DATA9	-	I	hbg
XM0DATA10	XM0DATA10	-	I	hbg
XM0DATA11	XM0DATA11	-	I	hbg
XM0DATA12	XM0DATA12	-	I	hbg
XM0DATA13	XM0DATA13	-	I	hbg
XM0DATA14	XM0DATA14	-	I	hbg
XM0DATA15	XM0DATA15	-	I	hbg
XM1CKE0	XM1CKE0	-	O(H)	mbg
XM1CKE1	XM1CKE1	-	O(H)	mbg
XM1CSN0	XM1CSN0	-	O(H)	mbg
XM1CSN1	XM1CSN1	-	O(H)	mbg
XM1DQM0	XM1DQM0	-	O(L)	mbg
XM1DQM1	XM1DQM1	-	O(L)	mbg
XM1DQM2	XM1DQM2	-	O(L)	mbg
XM1DQM3	XM1DQM3	-	O(L)	mbg
XM1DQS0	XM1DQS0	-	I	mbg
XM1DQS1	XM1DQS1	-	I	mbg
XM1DQS2	XM1DQS2	-	I	mbg
XM1DQS3	XM1DQS3	-	I	mbg
XM1RASN	XM1RASN	-	O(H)	mbg
XM1CASN	XM1CASN	-	O(H)	mbg
XM1WEN	XM1WEN	-	O(H)	mbg
XM1SCLK	XM1SCLK	-	O(L)	mbg
XM1SCLKN	XM1SCLKN	-	O(H)	mbg

XM1ADDR0	XM1ADDR0	-	O(L)	mbg
XM1ADDR1	XM1ADDR1	-	O(L)	mbg
XM1ADDR10	XM1ADDR10	-	O(L)	mbg
XM1ADDR2	XM1ADDR2	-	O(L)	mbg
XM1ADDR3	XM1ADDR3	-	O(L)	mbg
XM1ADDR4	XM1ADDR4	-	O(L)	mbg
XM1ADDR5	XM1ADDR5	-	O(L)	mbg
XM1ADDR6	XM1ADDR6	-	O(L)	mbg
XM1ADDR7	XM1ADDR7	-	O(L)	mbg
XM1ADDR8	XM1ADDR8	-	O(L)	mbg
XM1ADDR9	XM1ADDR9	-	O(L)	mbg
XM1ADDR11	XM1ADDR11	-	O(L)	mbg
XM1ADDR12	XM1ADDR12	-	O(L)	mbg
XM1ADDR13	XM1ADDR13	-	O(L)	mbg
XM1ADDR14	XM1ADDR14	-	O(L)	mbg
XM1ADDR15	XM1ADDR15	-	O(L)	mbg
XM1DATA0	XM1DATA0	-	I	mbg
XM1DATA1	XM1DATA1	-	I	mbg
XM1DATA2	XM1DATA2	-	I	mbg
XM1DATA3	XM1DATA3	-	I	mbg
XM1DATA4	XM1DATA4	-	I	mbg
XM1DATA5	XM1DATA5	-	I	mbg
XM1DATA6	XM1DATA6	-	I	mbg
XM1DATA7	XM1DATA7	-	I	mbg
XM1DATA8	XM1DATA8	-	I	mbg
XM1DATA9	XM1DATA9	-	I	mbg
XM1DATA10	XM1DATA10	-	I	mbg
XM1DATA11	XM1DATA11	-	I	mbg
XM1DATA12	XM1DATA12	-	I	mbg
XM1DATA13	XM1DATA13	-	I	mbg
XM1DATA14	XM1DATA14	-	I	mbg
XM1DATA15	XM1DATA15	-	I	mbg
XM1DATA16	XM0ADDR16	-	O(L)	mbg
XM1DATA17	XM0ADDR17	-	O(L)	mbg
XM1DATA18	XM0ADDR18	-	O(L)	mbg
XM1DATA19	XM0ADDR19	-	O(L)	mbg
XM1DATA20	XM0ADDR20	-	O(L)	mbg

XM1DATA21	XM0ADDR21	-	O(L)	mbg
XM1DATA22	XM0ADDR22	-	O(L)	mbg
XM1DATA23	XM0ADDR23	-	O(L)	mbg
XM1DATA24	XM0ADDR24	-	O(L)	mbg
XM1DATA25	XM0ADDR25	-	O(L)	mbg
XM1DATA26	XM0ADDR26	-	O(L)	mbg
XM1DATA27	XM1DATA27	-	O(L)	mbg
XM1DATA28	XM1DATA28	-	O(L)	mbg
XM1DATA29	XM1DATA29	-	O(L)	mbg
XM1DATA30	XM1DATA30	-	O(L)	mbg
XM1DATA31	XM1DATA31	-	O(L)	mbg
XURXD0/GPA0	GPA0	-	I	hag
XUTXD0/GPA1	GPA1	-	I	hag
XUCTSN0/GPA2	GPA2	-	I	hag
XURTSN0/GPA3	GPA3	-	I	hag
XURXD1/GPA4	GPA4	-	I	hag
XUTXD1/GPA5	GPA5	-	I	hag
XUCTSN1/GPA6	GPA6	-	I	hag
XURTSN1/GPA7	GPA7	-	I	hag
XURXD2/GPB0	GPB0	-	I	hag
XUTXD2/GPB1	GPB1	-	I	hag
XURXD3/GPB2	GPB2	-	I	hag
XUTXD3/GPB3	GPB3	-	I	hag
XIRSDBW/GPB4	CF DATA DIR	-	O(L)	hag
XI2CSCL/GPB5	GPB5	-	I	hag
XI2CSDA/GPB6	GPB6	-	I	hag
XSPIMISO0/GPC0	GPC0	-	I	hag
XSPICLK0/GPC1	GPC1	-	I	hag
XSPIMOSI0/GPC2	GPC2	-	I	hag
XSPICS0/GPC3	GPC3	-	I	hag
XSPIMISO1/GPC4	GPC4	-	I	hag
XSPICLK1/GPC5	GPC5	-	I	hag
XSPIMOSI1/GPC6	GPC6	-	I	hag
XSPICS1/GPC7	GPC7	-	I	hag
XPCMDCLK0/GPD0	GPD0	-	I	hag
XPCMEXTCLK0/GPD1	GPD1	-	I	hag
XPCMFSYNC0/GPD2	GPD2	-	I	hag
XPCMSIN0/GPD3	GPD3	-	I	hag

XPCMSOUT0/GPD4	GPD4	-	I	hag
XPCMDCLK1/GPE0	GPE0	-	I	hag
XPCMEXTCLK1/GPE1	GPE1	-	I	hag
XPCMFSYNC1/GPE2	GPE2	-	I	hag
XPCMSIN1/GPE3	GPE3	-	I	hag
XPCMSOUT1/GPE4	GPE4	-	I	hag
XCICLK/GPF0	GPF0	-	I	hag
XCIHREF/GPF1	GPF1	-	I	hag
XCIPCLK/GPF2	GPF2	-	I	hag
XCIRSTN/GPF3	GPF3	-	I	hag
XCIVSYNC/GPF4	GPF4	-	I	hag
XCIYDATA0/GPF5	GPF5	-	I	hag
XCIYDATA1/GPF6	GPF6	-	I	hag
XCIYDATA2/GPF7	GPF7	-	I	hag
XCIYDATA3/GPF8	GPF8	-	I	hag
XCIYDATA4/GPF9	GPF9	-	I	hag
XCIYDATA5/GPF10	GPF10	-	I	hag
XCIYDATA6/GPF11	GPF11	-	I	hag
XCIYDATA7/GPF12	GPF12	-	I	hag
XPWMECLK/GPF13	GPF13	-	I	hag
XPWMTOUT0/GPF14	GPF14	-	I	hag
XPWMTOUT1/GPF15	GPF15	-	I	hag
XMMCCLK0/GPG0	GPG0	-	I	hag
XMMCCMD0/GPG1	GPG1	-	I	hag
XMMCDATA0_0/GPG2	GPG2	-	I	hag
XMMCDATA0_1/GPG3	GPG3	-	I	hag
XMMCDATA0_2/GPG4	GPG4	-	I	hag
XMMCDATA0_3/GPG5	GPG5	-	I	hag
XMMCCDN0/GPG6	GPG6	-	I	hag
XMMCCLK1/GPH0	GPH0	-	I	hag
XMMCCMD1/GPH1	GPH1	-	I	hag
XMMCDATA1_0/GPH2	GPH2	-	I	hag
XMMCDATA1_1/GPH3	GPH3	-	I	hag
XMMCDATA1_2/GPH4	GPH4	-	I	hag
XMMCDATA1_3/GPH5	GPH5	-	I	hag
XMMCDATA1_4/GHP6	GPH6	-	I	hag
XMMCDATA1_5/GPH7	GPH7	-	I	hag

XMMCDATA1_6/GPH8	GPH8	-	I	hag
XMMCDATA1_7/GPH9	GPH9	-	I	hag
XVVD0/GPI0	GPI0	-	I	hag_a
XVVD1/GPI1	GPI1	-	I	hag_a
XVVD2/GPI2	GPI2	-	I	hag_a
XVVD3/GPI3	GPI3	-	I	hag_a
XVVD4/GPI4	GPI4	-	I	hag_a
XVVD5/GPI5	GPI5	-	I	hag_a
XVVD6/GPI6	GPI6	-	I	hag_a
XVVD7/GPI7	GPI7	-	I	hag_a
XVVD8/GPI8	GPI8	-	I	hag_a
XVVD9/GPI9	GPI9	-	I	hag_a
XVVD10/GPI10	GPI10	-	I	hag_a
XVVD11/GPI11	GPI11	-	I	hag_a
XVVD12/GPI12	GPI12	-	I	hag_a
XVVD13/GPI13	GPI13	-	I	hag_a
XVVD14/GPI14	GPI14	-	I	hag_a
XVVD15/GPI15	GPI15	-	I	hag_a
XVVD16/GPJ0	GPJ0	-	I	hag_a
XVVD17/GPJ1	GPJ1	-	I	hag_a
XVVD18/GPJ2	GPJ2	-	I	hag_a
XVVD19/GPJ3	GPJ3	-	I	hag_a
XVVD20/GPJ4	GPJ4	-	I	hag_a
XVVD21/GPJ5	GPJ5	-	I	hag_a
XVVD22/GPJ6	GPJ6	-	I	hag_a
XVVD23/GPJ7	GPJ7	-	I	hag_a
XVHSYNC/GPJ8	GPJ8	-	I	hag_a
XVVSYNC/GPJ9	GPJ9	-	I	hag_a
XVDEN/GPJ10	GPJ10	-	I	hag_a
XVVCLK/GPJ11	GPJ11	-	I	hag_a
XHIDATA0/GPK0	XHIDATA0	-	I	hag_a
XHIDATA1/GPK1	XHIDATA1	-	I	hag_a
XHIDATA2/GPK2	XHIDATA2	-	I	hag_a
XHIDATA3/GPK3	XHIDATA3	-	I	hag_a
XHIDATA4/GPK4	XHIDATA4	-	I	hag_a
XHIDATA5/GPK5	XHIDATA5	-	I	hag_a
XHIDATA6/GPK6	XHIDATA6	-	I	hag_a
XHIDATA7/GPK7	XHIDATA7	-	I	hag_a

XHIDATA8/GPK8	XHIDATA8	-	I	hag_a
XHIDATA9/GPK9	XHIDATA9	-	I	hag_a
XHIDATA10/GPK10	XHIDATA10	-	I	hag_a
XHIDATA11/GPK11	XHIDATA11	-	I	hag_a
XHIDATA12/GPK12	XHIDATA12	-	I	hag_a
XHIDATA13/GPK13	XHIDATA13	-	I	hag_a
XHIDATA14/GPK14	XHIDATA14	-	I	hag_a
XHIDATA15/GPK15	XHIDATA15	-	I	hag_a
XHIDATA16/GPL13	XHIDATA16	-	I	hag_a
XHIDATA17/GPL14	XHIDATA17	-	I	hag_a
XHIADR0/GPL0	XHIADR0	-	I	hag_a
XHIADR1/GPL1	XHIADR1	-	I	hag_a
XHIADR2/GPL2	XHIADR2	-	I	hag_a
XHIADR3/GPL3	XHIADR3	-	I	hag_a
XHIADR4/GPL4	XHIADR4	-	I	hag_a
XHIADR5/GPL5	XHIADR5	-	I	hag_a
XHIADR6/GPL6	XHIADR6	-	I	hag_a
XHIADR7/GPL7	XHIADR7	-	I	hag_a
XHIADR8/GPL8	XHIADR8	-	I	hag_a
XHIADR9/GPL9	XHIADR9	-	I	hag_a
XHIADR10/GPL10	XHIADR10	-	I	hag_a
XHIADR11/GPL11	XHIADR11	-	I	hag_a
XHIADR12/GPL12	XHIADR12	-	I	hag_a
XHICSN/GPM0	XHICSN	-	I	hag_a
XHICSN_MAIN/GPM1	XHICSN_MAIN	-	I	hag_a
XHICSNUB/GPM2	XHICSNUB	-	I	hag_a
XHIIRQN/GPM5	XHIIRQN	-	O(H)	hag_a
XHIOEN/GPM4	XHIOEN	-	I	hag_a
XHIWEN/GPM3	XHIWEN	-	I	hag_a
XEINT0/GPN0	GPN0	-	I	hag_a
XEINT1/GPN1	GPN1	-	I	hag_a
XEINT2/GPN2	GPN2	-	I	hag_a
XEINT3/GPN3	GPN3	-	I	hag_a
XEINT4/GPN4	GPN4	-	I	hag_a
XEINT5/GPN5	GPN5	-	I	hag_a
XEINT6/GPN6	GPN6	-	I	hag_a
XEINT7/GPN7	GPN7	-	I	hag_a

XEINT8/GPN8	GPN8	-	I	hag_a
XEINT9/GPN9	GPN9	-	I	hag_a
XEINT10/GPN10	GPN10	-	I	hag_a
XEINT11/GPN11	GPN11	-	I	hag_a
XEINT12/GPN12	GPN12	-	I	hag_a
XEINT13/GPN13	GPN13	-	I	hag_a
XEINT14/GPN14	GPN14	-	I	hag_a
XEINT15/GPN15	GPN15	-	I	hag_a
XEFFVDD		-	AI	htr00
XPLLEFILTER		-	AI	r_h
XUHDN	UHDN	-	AI	usb1
XUHDP	UHDP	-	AI	usb1
XUSBDM	USBDM	-	AI	htr
XUSBDP	USBDP	-	AI	htr
XUSBDRVVBUS	USBDRVVBUS	-	O(L)	hag
XUSBID	USBID	-	AI	hr
XUSBREXT	USBREXT	-	AI	hr
XUSBVBUS	USBVBUS	-	AI	htr
XUSBXTI	USBXTI	-	AI	scb
XUSBXTO	USBXTO	-	AO	scb
XADCAIN0	ADCAIN0	-	AI	hr
XADCAIN1	ADCAIN1	-	AI	hr
XADCAIN2	ADCAIN2	-	AI	hr
XADCAIN3	ADCAIN3	-	AI	hr
XADCAIN4	ADCAIN4	-	AI	hr
XADCAIN5	ADCAIN5	-	AI	hr
XADCAIN6	ADCAIN6	-	AI	hr
XADCAIN7	ADCAIN7	-	AI	hr
XDACCOMP	DACCOMP	-	AO	hr
XDACIREF	DACIREF	-	AI	hr
XDACOUT0	DACOUT0	-	AO	hr
XDACOUT1	DACOUT1	-	AO	hr
XDACVREF	DACVREF	-	AI	hr
VDDADC	VDDADC	P	P	dth
VDDDAC	VDDDAC	P	P	dth
VDDAPLL	VDDAPLL	P	P	dtlh
VDDEPLL	VDDEPLL	P	P	dtlh
VDDMPLL	VDDMPLL	P	P	dtlh

VDDRTC	VDDRTC	P	P	drtc
VDDALIVE	VDDALIVE	P	P	dih
VDDARM	VDDARM	P	P	dich
VDDINT	VDDINT	P	P	dih
VDDM0	VDDM0	P	P	dth
VDDSS	VDDSS	P	P	dth
VDDM1	VDDM1	P	P	dtm
VDDM1	VDDM1	P	P	dtm
VDDHI	VDDHI	P	P	dth
VDDLCD	VDDLCD	P	P	dth
VDDSYS	VDDSYS	P	P	dth
VDDEXT	VDDEXT	P	P	dth
VDDMMC	VDDMMC	P	P	dth
VDDPCM	VDDPCM	P	P	dth
VDDUH	VDDUH	P	P	dth
VDDOTG	VDDOTG	P	P	dth
VDDOTGI	VDDOTGI	P	P	dih_u
VSSADC	VSSADC	P	P	sth
VSSDAC	VSSDAC	P	P	sth
VSSAPLL	VSSAPLL	P	P	stlh
VSSEPLL	VSSEPLL	P	P	stlh
VSSMPLL	VSSMPLL	P	P	stlh
VSSIP	VSSIP	P	P	si
VSSMEM	VSSMEM	P	P	sth
VSSSS	VSSMEM	P	P	sth
VSSPERI	VSSPERI	P	P	sth
VSSOTG	VSSOTG	P	P	sth
VSSOTGI	VSSOTGI	P	P	si_u

The table below shows I/O types and descriptions.

Input (I)/Output (O) Type	Descriptions
dih(vddivh), si(vssipvh)	Vdd/Vss for internal logic with internal pad power ring
dich(vddicvh)	Vdd for only internal logic
dth(vddtvh), sth(vsstvh)	1.8~3.3V Vdd/Vss for external logic
dtm(vddtvm)	1.8~2.5V Vdd for external logic
dtlh(vddtvlh), stlh(vsstvlh)	1.2V Vdd/Vss for external and internal logic
drtc(vddrtcvh)	1.8~3.0V Vdd for RTC power
dih_u(vddivh_usb)	Vdd for usb phy core
si_u(vssipvh_usb)	Vss for usb phy core
hag(pvhbsudtartg)	1.8V~3.3V Wide Range Bi-directional Buffer with Schmitt Trigger Input, Controllable Pull-up/down Resistor and A type Output driver
hag_a (pvhbsudtag_alv)	1.8V~3.3V Wide Range Bi-directional Alive Buffer with Schmitt Trigger Input, Controllable Pull-up/down Resistor and A type Output driver
hbg(pvhbsudtbrtg)	1.8V~3.3V Wide Range Bi-directional Buffer with Schmitt Trigger Input, Controllable Pull-up/down Resistor and B type Output driver
hb_c(pvhbsudtbrt_ckds)	1.8V~3.3V Wide Range Bi-directional Buffer with clock driver input for pulse clock or small amplitude clock, Schmitt Trigger Input, Controllable Pull-up/down Resistor and B type Output driver
mbg(pvmbsudtbrtg)	1.8V~2.5V Wide Range Bi-directional Buffer with Schmitt Trigger Input, Controllable Pull-up/down Resistor and B type Output driver
sca(pvhsosca)	1.8V~3.3V wide range oscillator for RTC Interface
scb(pvhsoscbrt)	1.8V~3.3V wide range oscillator for Wide Frequency
usb1(usb6002x1)	USB 1.1 pad
hr(pvhbr)	1.8V~3.3V wide range analog bi-direction path-through PAD with 3 different paths which have no resistor, 50ohm or 100ohm resistor
htr(pvhtbr)	1.8V~3.3V wide range analog tolerant bi-direction path-through PAD with 3 different paths which have no resistor, 50ohm or 100ohm resistor
htr00(pvhtbr00_efuse)	1.8V~3.3V wide range analog bi-direction path-through PAD without resistor for efuse memory
r_h(pvbr_h)	1.2V bi-direction path-through PAD with 3 different paths which have no resistor, 50ohm or 100ohm resistor

1.7 PIN DESCRIPTION

1.7.1 EXTERNAL MEMORY INTERFACE

- Shared Memory Port (SROMC / OneNAND / NAND / ATA)

Signal	I/O						Function					
	0	1	2	3	4	5	0	1	2	3	4	5
Xm0BEn[1:0]	O	O	O	O			nBE[1:0]	2'b11	2'b11	2'b11		
Xm0CSn[1:0]	O						nCS					
Xm0CSn[3:2]	O	O	O		IO	I	nCS	nCS	nCS		GPO[1:0]	EINT7[1:0]
Xm0CSn[5:4]	O			O	IO	I	nCS			nCS	GPO[3:2]	EINT7[3:2]
GPO[5:4]	O				IO	I	Reserved				GPO[5:4]	EINT7[5:4]
Xm0ADDR[5:0]	O	O	O	O			ADDR	6'h0	6'h0	ADDR		
Xm0ADDR[10:6]	O	O	O	O			ADDR	5'h0	5'h0	ADDR	GPO[10:6]	EINT7[10:6]
Xm0ADDR[15:11]	O	O	O	O			ADDR	5'h0	5'h0	5'h0	GPO[15:11]	EINT7[15:11]
Xm0OEn	O	O	O	O			nOE	nOE	1'b1	nIORD_CF		
Xm0WEn	O	O	O	O			nWE	nWE	1'b1	nIOWR_CF		
Xm0ADRVALID		O			IO	I		nADV			GPP[0]	EINT8[0]
Xm0SMCLK		O			IO	I		SMCLK			GPP[1]	EINT8[1]
Xm0DATA[15:0]	IO	IO	IO	IO			DATA	DATA	DATA	DATA		
Xm0WAITn	I			I	IO	I	nWAIT			IORDY	GPP[2]	EINT8[2]
Xm0RDY0_ALE		I	O		IO	I		RDY0	ALE		GPP[3]	EINT8[3]
Xm0RDY1_CLE		I	O		IO	I		RDY1	CLE		GPP[4]	EINT8[4]
Xm0INTsm0_FWEn		I	O		IO	I		INT0	FWEn		GPP[5]	EINT8[5]
Xm0INTsm1_FREn		I	O		IO	I		INT1	FREn		GPP[6]	EINT8[6]
Xm0RPn_RnB		O	I		IO	I		nRP	RnB		GPP[7]	EINT8[7]
Xm0ADDR[18]	O				IO	I	ADDR18				GPQ[0]	EINT9[0]
Xm0ADDR[19]	O				IO	I	ADDR19				GPQ[1]	EINT9[1]
GPQ2	O				IO	I	Reserved				GPQ[2]	EINT9[2]
GPQ3	O				IO	I	Reserved				GPQ[3]	EINT9[3]
GPQ4	O				IO	I	Reserved				GPQ[4]	EINT9[4]
GPQ5	IO				IO	I	Reserved				GPQ[5]	EINT9[5]
GPQ6	IO				IO	I	Reserved				GPQ[6]	EINT9[6]
Xm0ADDR[17]	O				IO	I	ADDR17				GPQ[7]	EINT9[7]
Xm0ADDR[16]	O				IO	I	ADDR16				GPQ[8]	EINT9[8]

Signal	I/O						Function					
	0	1	2	3	4	5	0	1	2	3	4	5
Xm0INTata				I	IO	I				INTata	GPP[8]	EINT8[8]
Xm0RESETata				O	IO	I				RESETata	GPP[9]	EINT8[9]
Xm0INPACKata				I	IO	I				INPACKata	GPP[10]	EINT8[10]
Xm0REGata				O	IO	I				REGata	GPP[11]	EINT8[11]
Xm0WEata				O	IO	I				WEnata	GPP[12]	EINT8[12]
Xm0OEata				O	IO	I				OEnata	GPP[13]	EINT8[13]
Xm0CData				I	IO	I				CDnata	GPP[14]	EINT8[14]

Signal	I/O	IO Power	Description
ADDR[19:0]	O	VDDM0	Memory port 0 common address bus
DATA[15:0]	O	VDDM0	Memory port 0 common data bus
nCS[5:4]	O	VDDM0	Memory port 0 SROM/CF Chip Select support up to 2 memory bank.
nCS[3:2]	O	VDDM0	Memory port 0 SROM / OneNAND / NAND Flash Chip Select support up to 2 memory bank.
nCS[1:0]	O	VDDM0	Memory port 0 SROM Chip Select support up to 2 memory bank.
nBE[1:0]	O	VDDM0	Memory port 0 SROM Byte Enable
nWAIT	I	VDDM0	Memory port 0 SROM Wait
nOE	O	VDDM0	Memory port 0 SROM / OneNAND Output Enable
nWE	O	VDDM0	Memory port 0 SROM / OneNAND Write Enable
nADV	O	VDDM0	Memory port 0 OneNAND Address Valid
SMCLK	O	VDDM0	Memory port 0 OneNAND Clock
RDY[1:0]	I	VDDM0	Memory port 0 OneNAND Bank 0,1 Ready
INT[1:0]	I	VDDM0	Memory port 0 OneNAND Bank 0,1 Interrupt
nRP	O	VDDM0	Memory port 0 OneNAND Reset
ALE	O	VDDM0	Memory port 0 NAND Flash Address Latch Enable
CLE	O	VDDM0	Memory port 0 NAND Flash Command Latch Enable
FWEn	O	VDDM0	Memory port 0 NAND Flash Write Enable
FREn	O	VDDM0	Memory port 0 NAND Flash Read Enable
RnB	I	VDDM0	Memory port 0 NAND Flash Ready/Busy
nIORD_CF	O	VDDM0	Memory port 0 CF Read strobe for I/O mode
nIOWR_CF	O	VDDM0	Memory port 0 CF Write strobe for I/O mode

Signal	I/O	IO Power	Description
IORDY	I	VDDM0	Memory port 0 CF Wait signal form CF card
INTata	I	VDDSS	Memory port 0 CF Interrupt request from ATA controller
RESETata	O	VDDSS	Memory port 0 CF CARD Reset
INPACKata	I	VDDSS	Memory port 0 CF Input acknowledge in I/O mode
REGata	O	VDDSS	Memory port 0 CF Interrupt request from CF card
WEnata	O	VDDSS	Memory port 0 CF Write enable strobe
OEnata	O	VDDSS	Memory port 0 CF Output enable strobe
CDnata	I	VDDSS	Memory port 0 CF Card detection
GPO[5:4]	IO	VDDM0	GPIO
GPQ[6:2]	IO	VDDM0	GPIO

- DRAM(mDDR and mSDRAM) Memory Port
- Shared Memory Port (DRAM1 / SROMC)

Signal	I/O		Function	
	0	1	0	1
Xm1CKE	O		Xm1CKE	
Xm1SCLK	O		Xm1SCLK	
Xm1SCLKn	O		Xm1SCLKn	
Xm1CSn[1:0]	O		Xm1CSN[1:0]	
Xm1ADDR[15:0]	O		Xm1ADDR[15:0]	
Xm1RASn	O		Xm1RAS	
Xm1CASn	O		Xm1CAS	
Xm1WEn	O		Xm1WEN	
Xm1DATA[15:0]	IO		Xm1DATA[15:0]	
Xm1DATA[31:16]	IO	O	Xm1DATA[31:16]	Xm0ADDR[26:16]
Xm1DQM[3:0]	O		Xm1DQM[3:0]	
Xm1DQS[3:0]	IO		Xm1DQS[3:0]	

Signal	I/O	IO Power	Description
Xm1CKE[1:0]	O	VDDM1	Memory port 1 DRAM Clock Enable
Xm1SCLK	O	VDDM1	Memory port 1 DRAM Clock
Xm1SCLKn	O	VDDM1	Memory port 1 DRAM Inverted Clock of Xm1SCLK
Xm1CSn[1:0]	O	VDDM1	Memory port 1 DRAM Chip Select support up to 2 memory bank.
Xm1ADDR[15:0]	O	VDDM1	Memory port 1 DRAM Address bus
Xm1RASn	O	VDDM1	Memory port 1 DRAM Row Address Strobe
Xm1CASn	O	VDDM1	Memory port 1 DRAM Column Address Strobe
Xm1WEn	O	VDDM1	Memory port 1 DRAM Write Enable
Xm1DATA[15:0]	IO	VDDM1	Memory port 1 DRAM Lower Half Data bus.
Xm1DATA[31:16]	IO	VDDM1	Xm0ADDR[26:16] of SROMC. Xm1DATA[31:16] can be used as Memory port 1 DRAM Upper Half Data bus by System Controller setting.
Xm1DQM[3:0]	O	VDDM1	Memory port 1 DRAM Data Mask
Xm1DQS[3:0]	IO	VDDM1	Memory port 1 DRAM Data Strobe

1.7.2 SERIAL COMMUNICATION

- UART / IrDA / CF / I2C (Channel 1)

Signal	I/O							Function						
	0	1	2	3	4	5	6	0	1	2	3	4	5	6
XuRXD[0]	IO	I					I	GPA[0]	XuRXD[0]					EINT1[0]
XuTXD[0]	IO	O					I	GPA[1]	XuTXD[0]					EINT1[1]
XuCTSn[0]	IO	I					I	GPA[2]	XuCTSn[0]					EINT1[2]
XuRTSn[0]	IO	O					I	GPA[3]	XuRTSn[0]					EINT1[3]
XuRXD[1]	IO	I					I	GPA[4]	XuRXD[1]					EINT1[4]
XuTXD[1]	IO	O					I	GPA[5]	XuTXD[1]					EINT1[5]
XuCTSn[1]	IO	I					I	GPA[6]	XuCTSn[1]					EINT1[6]
XuRTSn[1]	IO	O					I	GPA[7]	XuRTSn[1]					EINT1[7]
XuRXD[2]	IO	I		I			I	GPB[0]	XuRXD[2]		XirRXD	ADDR_CF[0]		EINT1[8]
XuTXD[2]	IO	O		O	O		I	GPB[1]	XuTXD[2]		XirTXD	ADDR_CF[1]		EINT1[9]
XuRXD[3]	IO	I	I		O	IO	I	GPB[2]	XuRXD[3]	XirRXD		ADDR_CF[2]	Xi2cSCL[1]	EINT1[10]
XuTXD[3]	IO	O	O		O	IO	I	GPB[3]	XuTXD[3]	XirTXD			Xi2cSDA[1]	EINT1[11]
XirSDBW	IO	I	I	O			I	GPB[4]	XirSDBW	XcamFIELD	XcfDIR			EINT1[12]

Signal	I/O	IO Power	Description
XuRXD[0]	I	VDDEXT	UART 0 receives data input
XuTXD[0]	O	VDDEXT	UART 0 transmits data output
XuCTSn[0]	I	VDDEXT	UART 0 clear to send input signal
XuRTSn[0]	O	VDDEXT	UART 0 request to send output signal
XuRXD[1]	I	VDDEXT	UART 1 receives data input
XuTXD[1]	O	VDDEXT	UART 1 transmits data output
XuCTSn[1]	I	VDDEXT	UART 1 clear to send input signal
XuRTSn[1]	O	VDDEXT	UART 1 request to send output signal
XuRXD[2]	I	VDDEXT	UART 2 receive data input
XuTXD[2]	O	VDDEXT	UART 2 transmits data output
XuRXD[3]	I	VDDEXT	UART 3 receive data input
XuTXD[3]	O	VDDEXT	UART 3 transmits data output
XirSDBW	O	VDDEXT	IrDA transiver control signal (Shutdown and bandwidth control)
XirRXD	I	VDDEXT	IrDA Rx data
XirTXD	O	VDDEXT	IrDA Tx data
XcamFIELD	I	VDDEXT	601 FIELD signal for External Camera Interface
XcfDir	O	VDDEXT	CF Data transfer direction
ADDR_CF[2:0]	O	VDDEXT	CF card address
Xi2cSCL[1]	IO	VDDEXT	IIC-bus clock (Channel 1)
Xi2cSDA[1]	IO	VDDEXT	IIC-bus data (Channel 1)
EINT1[12:0]	I	VDDEXT	External Interrupt 1

- IIC Bus (Channel 0)

Signal	I/O			Function		
	0	1	2	0	1	2
Xi2cSCL[0]	IO	IO	I	GPB[5]	Xi2cSCL[0]	EINT1[13]
Xi2cSDA[0]	IO	IO	I	GPB[6]	Xi2cSDA[0]	EINT1[14]

Signal	I/O	IO Power	Description
Xi2cSCL[0]	IO	VDDEXT	IIC-bus clock (Channel 0)
Xi2cSDA[0]	IO	VDDEXT	IIC-bus data (Channel 0)
EINT1[14:13]	I	VDDEXT	External Interrupt 1

- SPI(2-Ch) / I2S Multi Channel Port(Data Out)

Signal	I/O					Function				
	0	1	2	3	4	0	1	2	3	4
XspiMISO[0]	IO	IO			I	GPC[0]	XspiMISO[0]			EINT2[0]
XspiCLK[0]	IO	IO			I	GPC[1]	XspiCLK[0]			EINT2[1]
XspiMOSI[0]	IO	IO			I	GPC[2]	XspiMOSI[0]			EINT2[2]
XspiCS[0]	IO	IO			I	GPC[3]	XspiCS[0]			EINT2[3]
XspiMISO[1]	IO	IO	IO	O	I	GPC[4]	XspiMISO[1]	XmmcCMD2	I2SMULTI_DO[0]	EINT2[4]
XspiCLK[1]	IO	IO	O	O	I	GPC[5]	XspiCLK[1]	XmmcCLK2	I2SMULTI_DO[1]	EINT2[5]
XspiMOSI[1]	IO	IO			I	GPC[6]	XspiMOSI[1]			EINT2[6]
XspiCS[1]	IO	IO		O	I	GPC[7]	XspiCS[1]		I2SMULTI_DO[2]	EINT2[7]

Signal	I/O	IO Power	Description
XspiMISO[0]	IO	VDDEXT	SPI MISO[0]. SPI master data input line
XspiCLK[0]	IO	VDDEXT	SPI CLK[0]. SPI clock for channel 0
XspiMOSI[0]	IO	VDDEXT	SPI MOSI[0]. SPI master data output line
XspiCS[0]	IO	VDDEXT	SPI chip select(only for slave mode)
XspiMISO[1]	IO	VDDMMC	SPI MISO[1]. SPI master data input line
XspiCLK[1]	IO	VDDMMC	SPI CLK[1]. SPI clock for channel 1
XspiMOSI[1]	IO	VDDMMC	SPI MOSI[1]. SPI master data output line
XspiCS[1]	IO	VDDMMC	SPI chip select(only for slave mode)
EINT2[3:0]	I	VDDEXT	External Interrupt 2
EINT2[7:4]	I	VDDMMC	External Interrupt 2
XmmcCMD2	IO	VDDMMC	COMMAND/RESPONSE (SD/SDIO/MMC card interface channel 2)
XmmcCLK2	O	VDDMMC	CLOCK (SD/SDIO/MMC card interface channel 2)
I2SMULTIDO[2:0]	O	VDDMMC	IIS Multi channel-bus serial data output

- PCM(2-Ch) / IIS / AC97

Signal	I/O					Function				
	0	1	2	3	4	0	1	2	3	4
XpcmDCLK[0]	IO	O	IO	I	I	GPD[0]	XpcmDCLK[0]	Xi2sCLK[0]	X97BITCLK	EINT3[0]
XpcmEXTCLK[0]	IO	I	O	O	I	GPD[1]	XpcmEXTCLK[0]	Xi2sCDCLK[0]	X97RESETn	EINT3[1]
XpcmFSYNC[0]	IO	O	IO	O	I	GPD[2]	XpcmFSYNC[0]	Xi2sLRCK[0]	X97SYNC	EINT3[2]
XpcmSIN[0]	IO	I	I	I	I	GPD[3]	XpcmSIN[0]	Xi2sDI[0]	X97SDI	EINT3[3]
XpcmSOUT[0]	IO	O	O	O	I	GPD[4]	XpcmSOUT[0]	Xi2sDO[0]	X97SDO	EINT3[4]
XpcmDCLK[1]	IO	O	IO	I		GPE[0]	XpcmDCLK[1]	Xi2sCLK[1]	X97BITCLK	
XpcmEXTCLK[1]	IO	I	O	O		GPE[1]	XpcmEXTCLK[1]	Xi2sCDCLK[1]	X97RESETn	
XpcmFSYNC[1]	IO	O	IO	O		GPE[2]	XpcmFSYNC[1]	Xi2sLRCK[1]	X97SYNC	
XpcmSIN[1]	IO	I	I	I		GPE[3]	XpcmSIN[1]	Xi2sDI[1]	X97SDI	
XpcmSOUT[1]	IO	O	O	O		GPE[4]	XpcmSOUT[1]	Xi2sDO[1]	X97SDO	

Signal	I/O	IO Power	Description
XpcmDCLK[0]	O	VDDPCM	PCM Serial Shift Clock
XpcmEXTCLK[0]	I	VDDPCM	Optional reference clock (divided internally to generate PCM timing and XpcmDCLK)
XpcmFSYNC[0]	O	VDDPCM	PCM Sync indicating start of word
XpcmSIN[0]	I	VDDPCM	PCM Serial Data Input
XpcmSOUT[0]	O	VDDPCM	PCM Serial Data Output
XpcmDCLK[1]	O	VDDPCM	PCM Serial Shift Clock
XpcmEXTCLK[1]	I	VDDPCM	Optional reference clock(divided internally to generate PCM timing and XpcmDCLK)
XpcmFSYNC[1]	O	VDDPCM	PCM Sync indicating start of word
XpcmSIN[1]	I	VDDPCM	PCM Serial Data Input
XpcmSOUT[1]	O	VDDPCM	PCM Serial Data Output
Xi2sLRCK[1:0]	IO	VDDPCM	IIS-bus channel select clock
Xi2sCDCLK[1:0]	O	VDDPCM	IIS CODEC system clock
Xi2sCLK[1:0]	IO	VDDPCM	IIS-bus serial clock
Xi2sDI[1:0]	I	VDDPCM	IIS-bus serial data input
Xi2sDO[1:0]	O	VDDPCM	IIS-bus serial data output
X97BITCLK	I	VDDPCM	AC-Link bit clock(12.288MHz) from AC97 Codec to AC97 Controller
X97RESETn	O	VDDPCM	AC-link Reset to Codec
X97SYNC	O	VDDPCM	AC-link Frame Synchronization (Sampling Frequency 48kHz) from AC97 Controller to AC97 Codec
X97SDI	I	VDDPCM	AC-link Serial Data input from AC97 Codec
X97SDO	O	VDDPCM	AC-link Serial Data output to AC97 Codec
EINT3[4:0]	I	VDDPCM	External Interrupt 3

- USB Host

Signal	I/O	IO Power	Description
XuhDN	IO	VDDUH	USB Data pin DATA(–) for USB 1.1 Host
XuhDP	IO	VDDUH	USB Data pin DATA(+) for USB 1.1 Host

- USB OTG

Signal	I/O	IO Power	Description
XusbDP	IO	VDDOTG	USB Data pin DATA(+)
XusbDM	IO	VDDOTG	USB Data pin DATA(–)
XusbXTI	I	VDDOTG	Crystal Oscillator XI signal
XusbXTO	I	VDDOTG	Crystal Oscillator XO signal
XusbREXT	IO	VDDOTG	External 44.2-ohm (+/- 1%) resistor connection
XusbVBUS	IO	VDDOTG	USB Mini-Receptacle Vbus
XusbID	I	VDDOTG	USB Mini-Receptacle Identifier
XusbDRVVBUS	O	VDDOTG	Drive Vbus for Off-Chip Charge Pump

1.7.3 PARALLEL COMMUNICATION

- External Interrupts

Signal	I/O			Function		
	0	1	2	0	1	2
XEINT[0]	IO	I	I	GPN[0]	XEINT[0]	XkpROW[0]
XEINT[1]	IO	I	I	GPN[1]	XEINT[1]	XkpROW[1]
XEINT[2]	IO	I	I	GPN[2]	XEINT[2]	XkpROW[2]
XEINT[3]	IO	I	I	GPN[3]	XEINT[3]	XkpROW[3]
XEINT[4]	IO	I	I	GPN[4]	XEINT[4]	XkpROW[4]
XEINT[5]	IO	I	I	GPN[5]	XEINT[5]	XkpROW[5]
XEINT[6]	IO	I	I	GPN[6]	XEINT[6]	XkpROW[6]
XEINT[7]	IO	I	I	GPN[7]	XEINT[7]	XkpROW[7]
XEINT[15:8]	IO	I		GPN[15:8]	XEINT[15:8]	

Signal	I/O	IO Power	Description
XEINT[15:0]	I	VDDSYS	External interrupts
XkpROW[7:0]	I	VDDSYS	Keypad I/F Row

1.7.4 MODEM INTERFACE

- Host I/F / HIS(MIPI) / Key I/F / ATA

Signal	I/O						Function					
	0	1	2	3	4	5	0	1	2	3	4	5
XhiCSn	IO	I	I		IO	O	GPM[0]	XhiCSn	XEINT[23]		DATA_CF[10]	CE_CF[0]
XhiCSn_main	IO	I	I		IO	O	GPM[1]	XhiCSn_main	XEINT[24]		DATA_CF[11]	CE_CF[1]
XhiCSn_sub	IO	I	I	IO	IO	O	GPM[2]	XhiCSn_sub	XEINT[25]	XhiMDPSYNC	DATA_CF[12]	IORD_CF
XhiWEn	IO	I	I		IO	O	GPM[3]	XhiWEn	XEINT[26]		DATA_CF[13]	IOWR_CF
XhiOEn	IO	I	I		IO	I	GPM[4]	XhiOEn	XEINT[27]		DATA_CF[14]	IORDY_CF
XhiINTR	IO	O			IO		GPM[5]	XhiINTR			DATA_CF[15]	
XhiADDR[2:0]	IO	I	O			O	GPL[2:0]	XhiADDR[2:0]	XkpCOL[2:0]			ADDR_CF[2:0]
XhiADDR[3]	IO	I	O			I	GPL[3]	XhiADDR[3]	XkpCOL[3]			Xm0INTata
XhiADDR[4]	IO	I	O			O	GPL[4]	XhiADDR[4]	XkpCOL[4]			Xm0RESETata
XhiADDR[5]	IO	I	O			I	GPL[5]	XhiADDR[5]	XkpCOL[5]			Xm0INPACKata
XhiADDR[6]	IO	I	O			O	GPL[6]	XhiADDR[6]	XkpCOL[6]			Xm0REGata
XhiADDR[7]	IO	I	O			I	GPL[7]	XhiADDR[7]	XkpCOL[7]			Xm0Cdata
XhiADDR[8]	IO	I	I		O		GPL[8]	XhiADDR[8]	XEINT[16]		CE_CF[0]	
XhiADDR[9]	IO	I	I		O		GPL[9]	XhiADDR[9]	XEINT[17]		CE_CF[1]	
XhiADDR[10]	IO	I	I		O		GPL[10]	XhiADDR[10]	XEINT[18]		IORD_CF	
XhiADDR[11]	IO	I	I		O		GPL[11]	XhiADDR[11]	XEINT[19]		IOWR_CF	
XhiADDR[12]	IO	I	I		I		GPL[12]	XhiADDR[12]	XEINT[20]		IORDY_CF	
XhiDATA[0]	IO	IO	O		IO		GPK[0]	XhiDATA[0]	XhrxREADY		DATA_CF[0]	
XhiDATA[1]	IO	IO	I		IO		GPK[1]	XhiDATA[1]	XhrxWAKE		DATA_CF[1]	
XhiDATA[2]	IO	IO	I		IO		GPK[2]	XhiDATA[2]	XhrxFLAG		DATA_CF[2]	
XhiDATA[3]	IO	IO	I		IO		GPK[3]	XhiDATA[3]	XhrxDATA		DATA_CF[3]	
XhiDATA[4]	IO	IO	I		IO		GPK[4]	XhiDATA[4]	XhtxREADY		DATA_CF[4]	
XhiDATA[5]	IO	IO	O		IO		GPK[5]	XhiDATA[5]	XhtxWAKE		DATA_CF[5]	
XhiDATA[6]	IO	IO	O		IO		GPK[6]	XhiDATA[6]	XhtxFLAG		DATA_CF[6]	
XhiDATA[7]	IO	IO	O		IO		GPK[7]	XhiDATA[7]	XhtxDATA		DATA_CF[7]	
XhiDATA[15:8]	IO	IO	I		IO		GPK[15:8]	XhiDATA[15:8]	XkpROW[7:0]		DATA_CF[15:8]	
XhiDATA[16]	IO	IO	I		IO		GPL[13]	XhiDATA[16]	XEINT[21]		DATA_CF[8]	
XhiDATA[17]	IO	IO	I		IO		GPL[14]	XhiDATA[17]	XEINT[22]		DATA_CF[9]	

Signal	I/O	IO Power	Description
XhiCSn	I	VDDHI	Chip select, driven by the Modem chip
XhiCSn_main	I	VDDHI	Chip select for LCD bypass main, driven by the Modem chip
XhiCSn_sub	I	VDDHI	Chip select for LCD bypass sub, driven by the Modem chip
XhiWEn	I	VDDHI	Write enable, driven by the Modem chip
XhiOEn	I	VDDHI	Read enable, driven by the Modem chip
XhiINTR	O	VDDHI	Interrupt request to the Modem chip
XhiADDR[12:0]	I	VDDHI	Address bus, driven by the Modem chip
XhiDATA[17:0]	IO	VDDHI	Data bus, driven by the Modem chip
XEINT[27:16]	I	VDDHI	External interrupts
XkpCOL[7:0]	O	VDDHI	Keypad interface column outputs
XhrxREADY	O	VDDHI	The READY-signal indicates that the transmission of a new physical layer frame can begin. (Modem to Application Processor)
XhrxWAKE	I	VDDHI	The WAKE-signal is used to indicate to the receiver that the transmitter is willing to start a transmission. (Modem to Application Processor)
XhrxFLAG	I	VDDHI	(Modem to Application Processor)
XhrxDATA	I	VDDHI	Bit transmission in both directions occurs over a two-wire (DATA+FLAG) serial interface. The bits are transmitted sequentially, starting with the most significant bit of the physical layer frame. The DATA line always reflects the value being transmitted and the FLAG only toggles when the bit value transmitted remains constant. (Modem to Application Processor)
XhtxREADY	I	VDDHI	The READY-signal indicates that the transmission of a new physical layer frame can begin. (Application to Modem)
XhtxWAKE	O	VDDHI	The WAKE-signal is used to indicate to the receiver that the transmitter is willing to start a transmission.(Application to Modem)
XhtxFLAG	O	VDDHI	(Application to Modem)
XhtxDATA	O	VDDHI	Bit transmission in both directions occurs over a two-wire (DATA+FLAG) serial interface. The bits are transmitted sequentially, starting with the most significant bit of the physical layer frame. The DATA line always reflects the value being transmitted and the FLAG only toggles when the bit value transmitted remains constant. (Application to Modem)
XkpROW[7:0]	I	VDDHI	Keypad interface row inputs
DATA_CF[15:0]	IO	VDDHI	CF card data
CE_CF[1:0]	O	VDDHI	CF card enable strobe
IORD_CF	O	VDDHI	CF Read strobe for I/O mode
IOWR_CF	O	VDDHI	CF Write strobe for I/O mode
IORDY_CF	I	VDDHI	CF Wait signal form CF card
ADDR_CF[2:0]	O	VDDHI	CF card address

- PWM

Signal	I/O			Function		
	0	1	2	0	1	2
XpwmECLK	IO	I	I	GPF[13]	XpwmECLK	EINT4[13]
XpwmTOUT[0]	IO	O	O	GPF[14]	XpwmTOUT[0]	XCLKOUT
XpwmTOUT[1]	IO	O		GPF[15]	XpwmTOUT[1]	

Signal	I/O	IO Power	Description
XpwmECLK	I	VDDEXT	PWM Timer External Clock
XpwmTOUT[1:0]	O	VDDEXT	PWM Timer Output
XCLKOUT	O	VDDEXT	Clock output signal.
EINT4[13]	I	VDDEXT	External Interrupt 4

1.7.5 IMAGE/VIDEO PROCESSING

- Camera Interface

Signal	I/O			Function		
	0	1	2	0	1	2
XciCLK	IO	O	I	GPF[0]	XciCLK	EINT4[0]
XciHREF	IO	I	I	GPF[1]	XciHREF	EINT4[1]
XciPCLK	IO	I	I	GPF[2]	XciPCLK	EINT4[2]
XciRSTn	IO	O	I	GPF[3]	XciRSTn	EINT4[3]
XciVSYNC	IO	I	I	GPF[4]	XciVSYNC	EINT4[4]
XciYDATA[7:0]	IO	I	I	GPF[12:5]	XciYDATA[7:0]	EINT4[12:5]

Signal	I/O	IO Power	Description
XciCLK	O	VDDEXT	Master Clock to the Camera processor
XciHREF	I	VDDEXT	Horizontal Sync, driven by the Camera processor
XciPCLK	I	VDDEXT	Pixel Clock, driven by the Camera processor
XciVSYNC	I	VDDEXT	Vertical Sync, driven by the Camera processor
XciRSTn	O	VDDEXT	Software Reset to the Camera processor
XciYDATA[7:0]	I	VDDEXT	Pixel Data for YCbCr in 8-bit mode or for Y in 16-bit mode, driven by the Camera processor
EINT4[12:0]	I	VDDEXT	External Interrupt 4
XcamFIELD	I	VDDEXT	601 FIELD signal for External Camera Interface (refer to the XirSDBW signal muxing information)

1.7.6 DISPLAY CONTROL

- TFT LCD Display Interface

Signal	I/O				Function			
	0	1			0	1		
	-	-	-	-	-	LCD_SEL = 2'b00	LCD_SEL = 2'b01	LCD_SEL = 2'b10
XvVD[7:0]	IO	O	IO	O	GPI[7:0]	XvVD[7:0]	XvSYS_VD[7:0]	VEN_DATA[7:0]
XvVD[15:8]	IO	O	IO		GPI[15:8]	XvVD[15:8]	XvSYS_VD[15:8]	
XvVD[17:16]	IO	O	IO		GPJ[1:0]	XvVD[17:16]	XvSYS_VD[17:16]	
XvVD[21:18]	IO	O			GPJ[5:2]	XvVD[21:18]		
XvVD[22]	IO	O	O		GPJ[6]	XvVD[22]	XvSYS_VSYNC_LDI	
XvVD[23]	IO	O	O	O	GPJ[7]	XvVD[23]	XvSYS_OEn	VEN_FIELD
XvVCLK	IO	O	O	O	GPJ[11]	XvVCLK	XvSYS_Wen	V601_CLK
XvVSYNC	IO	O	O	O	GPJ[9]	XvVSYNC	XvSYS_CSn_sub	VEN_VSYNC
XvHSYNC	IO	O	O	O	GPJ[8]	XvHSYNC	XvSYS_CSn_main	VEN_HSYNC
XvVDEN	IO	O	O	O	GPJ[10]	XvVDEN	XvSYS_RS	VEN_HREF

Signal	I/O	IO Power	Description
XvVD[23:0]	O	VDDLCD	LCD pixel data output for RGB interface
XvVCLK	O	VDDLCD	Pixel clock signal for RGB interface
XvVSYNC	O	VDDLCD	Vertical synchronous signal for RGB interface
XvHSYNC	O	VDDLCD	Horizontal synchronous signal for RGB interface
XvVDEN	O	VDDLCD	Data enable signal for RGB interface
XvSYS_VD[17:0]	IO	VDDLCD	LCD pixel data output for i80 system interface
XvSYS_VSYNC_LDI	O	VDDLCD	I80 VSYNC interface control
XvSYS_OEn	O	VDDLCD	i80 Output Enable control
XvSYS_Wen	O	VDDLCD	i80 Write Enable control
XvSYS_CSn_sub	O	VDDLCD	i80 Sub LCD Chip select control
XvSYS_CSn_main	O	VDDLCD	i80 Main LCD Chip select control
XvSYS_RS	O	VDDLCD	i80 Register/Status Signal control
VEN_DATA[7:0]	O	VDDLCD	LCD pixel data output for 601 interface
VEN_FIELD	O	VDDLCD	FIELD signal for 601 interface
V601_CLK	O	VDDLCD	601 data clock
VEN_VSYNC	O	VDDLCD	601 Vertical Sync signal
VEN_HSYNC	O	VDDLCD	601 Horizontal Sync signal
VEN_HREF	O	VDDLCD	601 Data Enable Signal

- 2-ch DAC

Signal	I/O	IO Power	Description
XdacVREF	AI	VDDDAC	Reference voltage input (1.26V). External 100nF capacitor connection
XdacIREF	AI	VDDDAC	External 6.49kohm(+/- 1%) resistor connection
XdacCOMP	AI	VDDDAC	External 100nF capacitor connection
XdacOUT_0	AO	VDDDAC	Analog output of DAC
XdacOUT_1	AO	VDDDAC	Analog output of DAC

- ADC

Signal	I/O	IO Power	Description
Xadc_AIN[7:0]	AI	VDDADC	ADC Analog Input. If you don't use AIN[7], you must tie AIN[7] to VDDA_ADC or ADCTSC register must be setting to 0xd3.

- PLL

Signal	I/O	IO Power	Description
XpllEFILTER		VDDEPLL	Loop filter capacitor for PLL

1.7.7 STORAGE DEVICES

- MMC 2 channel / I2S Multi Channel Port

Signal	I/O							Function						
	0	1	2	3	4	5	6	0	1	2	3	4	5	6
XmmcCLK0	IO	O			O		I	GPG[0]	XmmcCLK0					EINT5[0]
XmmcCMD0	IO	IO			O		I	GPG[1]	XmmcCMD0					EINT5[1]
XmmcDAT0[0]	IO	IO			O		I	GPG[2]	XmmcDAT0[0]					EINT5[2]
XmmcDAT0[1]	IO	IO					I	GPG[3]	XmmcDAT0[1]					EINT5[3]
XmmcDAT0[2]	IO	IO					I	GPG[4]	XmmcDAT0[2]					EINT5[4]
XmmcDAT0[3]	IO	IO					I	GPG[5]	XmmcDAT0[3]					EINT5[5]
XmmcCDN0	IO	I	I				I	GPG[6]	XmmcCDN0	XmmcCDN1				EINT5[6]
XmmcCLK1	IO	IO		O	O		I	GPH[0]	XmmcCLK1		XkpCOL[0]			EINT6[0]
XmmcCMD1	IO	IO		O	O		I	GPH[1]	XmmcCMD1		XkpCOL[1]			EINT6[1]
XmmcDAT1[0]	IO	IO		O	O		I	GPH[2]	XmmcDAT1[0]		XkpCOL[2]			EINT6[2]
XmmcDAT1[1]	IO	IO		O			I	GPH[3]	XmmcDAT1[1]		XkpCOL[3]			EINT6[3]
XmmcDAT1[2]	IO	IO		O			I	GPH[4]	XmmcDAT1[2]		XkpCOL[4]			EINT6[4]
XmmcDAT1[3]	IO	IO		O			I	GPH[5]	XmmcDAT1[3]		XkpCOL[5]			EINT6[5]
XmmcDAT1[4]	IO	IO	IO	O	IO	O	I	GPH[6]	XmmcDAT1[4]	XmmcDAT2[0]	XkpCOL[6]	I2SMULTI_BCLK	ADDR_CF[0]	EINT6[6]
XmmcDAT1[5]	IO	IO	IO	O	IO	O	I	GPH[7]	XmmcDAT1[5]	XmmcDAT2[1]	XkpCOL[7]	I2SMULTI_CDCLK	ADDR_CF[1]	EINT6[7]
XmmcDAT1[6]	IO	IO	IO		IO	O	I	GPH[8]	XmmcDAT1[6]	XmmcDAT2[2]		I2SMULTI_LRCLK	ADDR_CF[2]	EINT6[8]
XmmcDAT1[7]	IO	IO	IO		I		I	GPH[9]	XmmcDAT1[7]	XmmcDAT2[3]		I2SMULTI_DI		EINT6[9]

Signal	I/O	IO Power	Description
XmmcCLK0	O	VDDMMC	CLOCK (SD/SDIO/MMC card interface channel 0)
XmmcCMD0	IO	VDDMMC	COMMAND/RESPONSE (SD/SDIO/MMC card interface channel 0)
XmmcDAT0[3:0]	IO	VDDMMC	DATA (SD/SDIO/MMC card interface channel 0)
XmmcCDN0	I	VDDMMC	CARD DETECT (SD/SDIO/MMC card interface channel 0)
XmmcCLK1	O	VDDMMC	CLOCK (SD/SDIO/MMC card interface channel 1)
XmmcCMD1	IO	VDDMMC	COMMAND/RESPONSE (SD/SDIO/MMC card interface channel 1)
XmmcDAT1[7:0]	IO	VDDMMC	DATA (SD/SDIO/MMC card interface channel 1)
XmmcCLK2	O	VDDMMC	CLOCK (SD/SDIO/MMC card interface channel 2)
XmmcCMD2	IO	VDDMMC	COMMAND/RESPONSE (SD/SDIO/MMC card interface channel 2)
XmmcDAT2[3:0]	IO	VDDMMC	DATA (SD/SDIO/MMC card interface channel 2)
ADDR_CF[2:0]	O	VDDMMC	CF card address
Xkp_COL[7:0]	O	VDDMMC	Key interface Column Output
EINT5[6:0]	I	VDDMMC	External Interrupt 5
EINT6[9:0]	I	VDDMMC	External Interrupt 6
I2SMULTI_SCLK	I/O	VDDMMC	IIS Multi channel bus Serial clock
I2SMULTI_CDCLK	I/O	VDDMMC	IIS Multi channel system CODEC clock
I2SMULTI_LRCLK	I/O	VDDMMC	IIS Multi channel bus channel selection clock
I2SMULTI_DI	I	VDDMMC	IIS Multi channel bus serial data input

1.7.8 SYSTEM MANAGEMENT

- Reset

Signal	I/O	IO Power	Description
XnRESET	I	VDDSYS	XnRESET suspends any operation in progress and places S3C6410X into a known reset state. For a reset, XnRESET must be held to L level for at least 4 FCLK after the processor power has been stabilized.
XnRSTOUT	O	VDDSYS	For external device reset control (nRSTOUT = nRESET & nWDTRST)

- Clock

Signal	I/O	IO Power	Description
XrtcXTI	I	VDDRTC	32kHz crystal input for RTC.
XrtcXTO	O	VDDRTC	32kHz crystal output for RTC.
XXTI27	I	VDDSYS	27MHz Crytal Input for display modules
XXTO27	O	VDDSYS	27MHz Crystal output for display modules
XXTI	I	VDDSYS	Crystal Input for internal osc circuit.
XXTO	O	VDDSYS	Crystal output for internal osc circuit.
XEXTCLK	I	VDDSYS	External clock source.

- JTAG

Signal	I/O	IO Power	Description
XjTRSTn	I	VDDSYS	XjTRSTn (TAP Controller Reset) resets the TAP controller at start. If debugger is used, A 10K pull-up resistor has to be connected. If debugger is not used, XjTRSTn pin must be at L or low active pulse.
XjTMS	I	VDDSYS	XjTMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states. A 10K pull-up resistor has to be connected to TMS pin.
XjTCK	I	VDDSYS	XjTCK (TAP Controller Clock) provides the clock input for the JTAG logic. . A 10K pull-dn resistor has to be connected to TMS pin
XjRTCK	O	VDDSYS	XjRTCK (TAP Controller Returned Clock) provides the clock output for the JTAG logic.
XjTDI	I	VDDSYS	XjTDI (TAP Controller Data Input) is the serial input for test instructions and data. A 10K pull-up resistor must be connected to TDI pin.
XjTDO	O	VDDSYS	XjTDO (TAP Controller Data Output) is the serial output for test instructions and data. It is possible to control pull-down by GPIO register. Refer GPIO manual.
XjDBGSEL	I	VDDSYS	JTAG selection. 1: Peripherals JTAG, 0: ARM1176JZF-S Core JTAG

- MISC

Signal	I/O	IO Power	Description
XOM[4:0]	I	VDDSYS	Operation mode selection. Refer System controller
XPWRRGTON	O	VDDSYS	Power Regulator enable
XSELNAND	I	VDDSYS	Select Flash Memory. 0: OneNAND, 1: NAND.
XnBATF	I	VDDSYS	Battery fault indication
WR_TEST	I	VDDSYS	Test Pin, It should be tied-up with VDDSYS.

1.8 POWER-SUPPLY GROUPS

- VDD

Signal	I/O	Description	Voltage
VDDALIVE	P	Internal power for alive block	1.2
VDDARM	P	Internal power for ARM1176 core and cache	1.1/1.2
VDDINT	P	Internal power for logic	1.2
VDDMPLL	P	Power for MPLL core	1.2
VDDAPLL	P	Power for APLL core	1.2
VDDEPLL	P	Power for EPLL core	1.2
VDDOTG	P	Power for USB OTG PHY	3.3
VDDOTGI	P	Internal power for USB OTG PHY	1.2
VDDMMC	P	IO power for SDMMC	1.8~3.3
VDDHI	P	IO power for Host I/F	1.8~3.3
VDDLCD	P	IO power for LCD	1.8~3.3
VDDPCM	P	IO power for PCM (Audio I/F – I2S, AC97)	1.8~3.3
VDDEXT	P	IO power for external I/F (UART, I2C, Camera I/F, etc.)	1.8~3.3
VDDSYS	P	IO power for system control. (Clock, reset, operation mode, JTAG, etc)	1.8~3.3
VDDUH	P	Power for USB Host	3.3
VDDADC	P	Power for ADC core and IO	3.3
VDDDAC	P	Power for DAC core and IO	3.3
VDDRTC	P	Power for RTC logic and IO	1.8~3.0
VDDM0	P	IO power for Memory Port 0	1.8~3.3
VDDSS	P	IO power for ATA IO muxed in MEM0 port	1.8~3.3
VDDM1	P	IO power for Memory Port 1	1.8/2.5

- VSS

Signal	I/O	Description
VSSIP	P	Internal Ground for Logic& ARM1176 core and cache
VSSMEM	P	IO ground for memory port 0 and 1
VSSOTG	P	Ground for USB OTG PHY.
VSSOTGI	P	Internal Ground for USB OTG PHY
VSSPERI	P	IO ground for USB HOST, SDMMC, Host I/F, LCD, PCM, External I/F and System Controller
VSSAPLL	P	Ground for APLL core
VSSMPLL	P	Ground for MPLL core
VSSEPLL	P	Ground for EPLL core
VSSADC	P	Ground for ADC core
VSSDAC	P	Ground for DAC core
VSSSS	P	Ground for SS

NOTES:

1. IO stands for input/output.
2. AI/AO stands for analog input/output.
3. ST stands for Schmitt-trigger.
4. P stands for power.

2 MEMORY MAP

2.1 MEMORY SYSTEM BLOCK DIAGRAM

Table 2-1 Address map.

Start Address	End Address	Int. ROM	Stepping Stone (NAND Ctrl.)	SROM Ctrl.	One NAND Ctrl. 0	One NAND Ctrl. 1	DRAM Ctrl 1
0x00000000	0x07FFFFFF	O[1]	-	O[1]	O[1]	-	-
0x08000000	0x0BFFFFFF	O	-	-	-	-	-
0x0C000000	0x0FFFFFFF	-	O	-	-	-	-
0x10000000	0x17FFFFFF	-	-	O	-	-	-
0x18000000	0x1FFFFFFF	-	-	O	-	-	-
0x20000000	0x27FFFFFF	-	-	O[2]	O[2]	-	-
0x28000000	0x2FFFFFFF	-	-	O[2]		O[2]	-
0x30000000	0x37FFFFFF	-	-	O	-	-	-
0x38000000	0x3FFFFFFF	-	-	O	-	-	-
0x40000000	0x47FFFFFF	-	-	-	-	-	-
0x48000000	0x4FFFFFFF	-	-	-	-	-	-
0x50000000	0x5FFFFFFF	-	-	-	-	-	O
0x60000000	0x6FFFFFFF	-	-	-	-	-	O

Boxes in thick borders show that the address ranges shown on the first two columns are directed to the corresponding memory controllers. Refer to the footnote at the bottom of this table for cases where single address range can be directed to multiple memory controllers.

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

[1] Refer to Memory sub-system chapter for details.

[2] This address range can be assigned to both SROM controller and OneNAND controller. The decision is made by System Controller. Refer to Memory sub-system chapter for details.

S3C6410X supports 32-bit physical address field and that address field can be seperated into two parts, one part is for memory, the other part is for pheriperal.

Main memory is accessed via SPINE bus, and its address range is from 0x0000_0000 to 0x6FFF_FFFF. This main memory part is seperated into four areas, boot image area, internal memory area, static memory area, and dynamic memory area.

Address range of boot image area is from 0x0000_0000 to 0x07FF_FFFF, but there is no real mapped-memory. Boot image area has mirrored image which points a partial region of internal memory area or static memory area. Start address of boot image is fixed to 0x0000_0000.

Internal memory area is used to access internal ROM and internal SRAM for boot loader, which is also called Steppingstone. Start address for each internal memory is fixed. Address range of internal ROM is from 0x0800_0000 to 0x0BFF_FFFF, but real storage is only 32KB. This region is read-only, and can be mapped to boot image area when internal ROM booting is selected. Address range of internal SRAM is from 0x0C00_0000 to 0x0FFF_FFFF, but real storage is only 4KB.

Address range of static memory area is from 0x1000_0000 to 0x3FFF_FFFF. SROM, SRAM, NOR Flash, asyncronous NOR interface device, OneNAND Flash, and Steppingstone can be accessed by this address area. Each area stands for a chip select, for example, address range from 0x1000_0000 to 0x17FF_FFFF stands for Xm0CSn[0]. Start address for each chip select is fixed. NAND Flash and CF/ATA cannot be accessed via static memory area, so if any of Xm0CSn[5:2] is mapped to NFCON or CFCON, related address region should not be accessed. One exception is that if Xm0CSn[2] is used for NAND Flash, Steppingstone is mirrored to address region from 0x2000_0000 to 27FF_FFFF.

Address range of dynamic memory area is from 0x4000_0000 to 0x6FFF_FFFF. DMC1 has right to use address range from 0x5000_0000 to 0x6FFF_FFFF. Start address for each chip select is configurable.

Pheripheral is accessed via PERI bus, and its address range is from 0x7000_0000 to 0x7FFF_FFFF. All SFRs can be accessed in this address range. Also, if data is needed to transfer from NFCON or CFCON, those data should be transferred via PERI bus.

2.2 DEVICE SPECIFIC ADDRESS SPACE

Table 2-2. Device Specific Address Space

Address		Size(MB)	Description	Note
0x0000_0000	0x07FF_FFFF	128MB	Booting Device Region by XOM Setting	Mirrored Region
0x0800_0000	0x0BFF_FFFF	64MB	Internal ROM	
0x0C00_0000	0x0FFF_FFFF	64MB	Stepping Stone (Boot Loader)	
0x1000_0000	0x17FF_FFFF	128MB	SROMC Bank0	
0x1800_0000	0x1FFF_FFFF	128MB	SROMC Bank 1	
0x2000_0000	0x27FF_FFFF	128MB	SROMC Bank 2	
0x2800_0000	0x2FFF_FFFF	128MB	SROMC Bank 3	
0x3000_0000	0x37FF_FFFF	128MB	SROMC Bank 4	
0x3800_0000	0x3FFF_FFFF	128MB	SROMC Bank 5	
0x4000_0000	0x47FF_FFFF	128MB	Reserved	
0x4800_0000	0x4FFF_FFFF	128MB		
0x5000_0000	0x5FFF_FFFF	256MB	DRAM Controller of the Memory Port1	
0x6000_0000	0x6FFF_FFFF	256MB		

Table 2-3. AHB Bus Memory Map

Address		Description	Note
0x7000_0000	0x700F_FFFF	SROM SFR	
0x7010_0000	0x701F_FFFF	OneNAND SFR	
0x7020_0000	0x702F_FFFF	NFCON SFR	
0x7030_0000	0x703F_FFFF	CFCON SFR	
0x7040_0000	0x70FF_FFFF	Reserved	
0x7100_0000	0x710F_FFFF	TZIC0	
0x7110_0000	0x711F_FFFF	TZIC1	
0x7120_0000	0x712F_FFFF	VIC0	
0x7130_0000	0x713F_FFFF	VIC1	
0x7140_0000	0x71FF_FFFF	Reserved	
0x7200_0000	0x72FF_FFFF	FIMG-3DSE_SFR	
0x7300_0000	0x730F_0FFF	ETB Memory	
0x7310_0000	0x731F_FFFF	ETB Registers	
0x7320_0000	0x73FF_FFFF	Reserved	
0x7400_0000	0x740F_FFFF	Indirect Host I/F	
0x7410_0000	0x741F_FFFF	Direct Host I/F	
0x7420_0000	0x742F_FFFF	Reserved	
0x7430_0000	0x743F_FFFF	USB Host	
0x7440_0000	0x744F_FFFF	MDP I/F	
0x7450_0000	0x74FF_FFFF	Reserved	
0x7500_0000	0x750F_FFFF	DMA0	
0x7510_0000	0x751F_FFFF	DMA1	
0x7520_0000	0x752F_FFFF	Reserved	
0x7530_0000	0x753F_FFFF	Reserved	
0x7540_0000	0x75FF_FFFF	Reserved	
0x7600_0000	0x760F_FFFF	Reserved	
0x7610_0000	0x761F_FFFF	2D Graphics	
0x7620_0000	0x762F_FFFF	TV Encoder	
0x7630_0000	0x763F_FFFF	TV Scaler	
0x7640_0000	0x76FF_FFFF	Reserved	

0x7700_0000	0x770F_FFFF	Post Processor	
0x7710_0000	0x771F_FFFF	LCD Controller	
0x7720_0000	0x772F_FFFF	Rotator	
0x7730_0000	0x77FF_FFFF	Reserved	
0x7800_0000	0x783F_FFFF	Camera I/F	
0x7840_0000	0x787F_FFFF	Reserved	
0x7880_0000	0x78BF_FFFF	JPEG	
0x78C0_0000	0x78FF_FFFF	Reserved	
0x7900_0000	0x79FF_FFFF	Reserved	
0x7A00_0000	0x7AFF_FFFF	Reserved	
0x7B00_0000	0x7BFF_FFFF	Reserved	
0x7C00_0000	0x7C0F_FFFF	USB OTG	
0x7C10_0000	0x7C1F_ FFFF	USB OTG SFR	
0x7C20_0000	0x7C2F_ FFFF	SD-MMC Controller 0 (High-Speed/CE-ATA)	
0x7C30_0000	0x7C3F_ FFFF	SD-MMC Controller 1 (High-Speed/CE-ATA)	
0x7C40_0000	0x7C4F_ FFFF	SD-MMC Controller 2 (High-Speed/CE-ATA)	
0x7C50_0000	0x7C5F_ FFFF	Reserved	
0x7D00_0000	0x7D0F_FFFF	D&I (Security Subsystem Config) SFR	
0x7D10_0000	0x7D1F_FFFF	AES_RX	
0x7D20_0000	0x7D2F_FFFF	DES_RX	
0x7D30_0000	0x7D3F_FFFF	HASH (SHA/PRNG)_RX	
0x7D40_0000	0x7D4F_ FFFF	RX FIFO SFR	
0x7D50_0000	0x7D5F_ FFFF	AES_TX	
0x7D60_0000	0x7D6F_ FFFF	DES_TX	
0x7D70_0000	0x7D7F_ FFFF	HASH(SHA/PRNG)_TX	
0x7D80_0000	0x7D8F_ FFFF	TX FIFO SFR	
0x7D90_0000	0x7D9F_ FFFF	RX_FIFO	
0x7DA0_0000	0x7DAF_ FFFF	TX_FIFO	
0x7DB0_0000	0x7DBF_ FFFF	SDMA0	
0x7DC0_0000	0x7DCF_ FFFF	SDMA1	

Table 2-4. APB Bus Memory Map

Address		Description	Note
0x7E00_0000	0x7E00_0FFF	RESERVED	
0x7E00_1000	0x7E00_1FFF	DMC1 SFR	
0x7E00_2000	0x7E00_2FFF	MFC SFR	
0x7E00_3000	0x7E00_3FFF	AXI_SYS	
0x7E00_4000	0x7E00_4FFF	Watch-Dog Timer	
0x7E00_5000	0x7E00_5FFF	RTC	
0x7E00_6000	0x7E00_6FFF	HSI TX	
0x7E00_7000	0x7E00_7FFF	HIS RX	
0x7E00_8000	0x7E00_8FFF	AXI_PERI	
0x7E00_9000	0x7E00_9FFF	AXI_SFR	
0x7E00_A000	0x7E00_AFFF	Keypad I/F	
0x7E00_B000	0x7E00_BFFF	ADC/Touch Screen	
0x7E00_C000	0x7E00_CFFF	ETM	
0x7E00_D000	0x7E00_DFFF	E-Fused register of the SECURE KEY	
0x7E00_E000	0x7E00_EFFF	CHIP_ID	
0x7E00_F000	0x7E00_FFFF	System Controller	
0x7E01_0000	0x7EFF_FFFF	Reserved	
0x7F00_0000	0x7F00_0FFF	TZPC	
0x7F00_1000	0x7F00_1FFF	AC97	
0x7F00_2000	0x7F00_2FFF	I2S Ch0	
0x7F00_3000	0x7F00_3FFF	I2S Ch1	
0x7F00_4000	0x7F00_4FFF	I2C	
0x7F00_5000	0x7F00_5FFF	UART	
0x7F00_6000	0x7F00_6FFF	PWM Timer	
0x7F00_7000	0x7F00_7FFF	IrDA	
0x7F00_8000	0x7F00_8FFF	GPIO	
0x7F00_9000	0x7F00_9FFF	PCM Ch0	
0x7F00_A000	0x7F00_AFFF	PCM Ch1	
0x7F00_B000	0x7F00_BFFF	SPI0	
0x7F00_C000	0x7F00_CFFF	SPI1	
0x7F00_D000	0x7F00_DFFF	I2S V40	
0x7F00_F000	0x7F00_FFFF	I2C1	

3 SYSTEM CONTROLLER

This chapter describes the functions and usage of system controller in S3C6410X RISC microprocessor.

3.1 OVERVIEW

The System Controller consists of two parts; System Clock Control and System Power-management Control. The System Clock Control logic in S3C6410X generates the required system clock signals, ARMCLK for CPU, HCLK for AXI/AHB-bus peripherals, and PCLK for the APB bus peripherals. There are three PLLs in S3C6410X. One is for ARMCLK only. Second is for HCLK and PCLK. The third thing is for peripheral, especially for audio related clocks. The clock control logic generates slow-rate clock-signals for ARMCLK, HCLK and PCLK by bypassing externally supplied clock sources. The clock signal to each peripheral block can be enabled or disabled by software control to reduce the power consumption.

In the power control logic, S3C6410X has various power management schemes to keep optimal power consumption for a given task. The power management in S3C6410X consists of four modes: General Clock gating mode, IDLE mode, STOP mode, and SLEEP mode.

General Clock Gating mode is used to control the ON/OFF of clocks for internal peripherals in S3C6410X. You can optimize the power consumption of S3C6410X using this General Clock Gating mode by supplying clocks for peripherals that are required for a certain application. For example, if a timer is not required, then you can disconnect the clock to the timer to reduce power.

IDLE mode disconnects the ARMCLK only to CPU core while it supplies the clock to all peripherals. By using IDLE mode, the power consumed by the CPU core is reduced.

STOP mode freezes all clocks to the CPU as well as peripherals by disabling PLLs. The power consumption is only due to the leakage current in S3C6410X.

SLEEP mode disconnects the internal power. Therefore, the power consumption due to CPU and the internal logic except the wakeup logic will be zero. In order to use the SLEEP mode two independent power sources are required. One of the two power sources supplies the power for the wake-up logic. The other one supplies the other internal logic including CPU, and must be controlled in order to be turned ON/OFF. In SLEEP mode, the second power supply source for the CPU and internal logic will be turned off.

A detailed description of the power-saving modes such as the entering sequence to the specific power-down mode or the wake-up sequence from a power-down mode is explained in the following Power Management section.

3.2 FEATURES

The System Controller includes the following features:

- Three PLLs: ARM PLL, main PLL, extra PLL (for the modules those use special frequency)
- Five power-saving mode: NORMAL, IDLE, STOP, DEEP-STOP, and SLEEP
- Six controllable power domain: domain-G, domain-V, domain-I, domain-P, domain-F, domain-S
- Control operating clocks of internal sub-blocks
- Control bus priority

3.3 FUNCTIONAL DESCRIPTION

This section describes the functionality of S3C6410X system controller. It covers the clock architecture, reset scheme, and power management modes.

3.3.1 HARDWARE ARCHITECTURE

Figure 3-1 illustrates S3C6410X block diagram. S3C6410X consists of ARM1176 processor, several media and graphic co-processors and various peripheral IPs. ARM1176 processor is connected to several memory controllers through 64-bit AXI-bus. This is done to meet bandwidth requirements. Media and graphic co-processors, which include MFC (Multi-Format Codec), JPEG, Camera interface, TV encoder, 3D accelerator and etc, are divided into six power domains. The six power domain can be controlled independently to reduce unwanted power consumption when the IPs is not required for an application program.

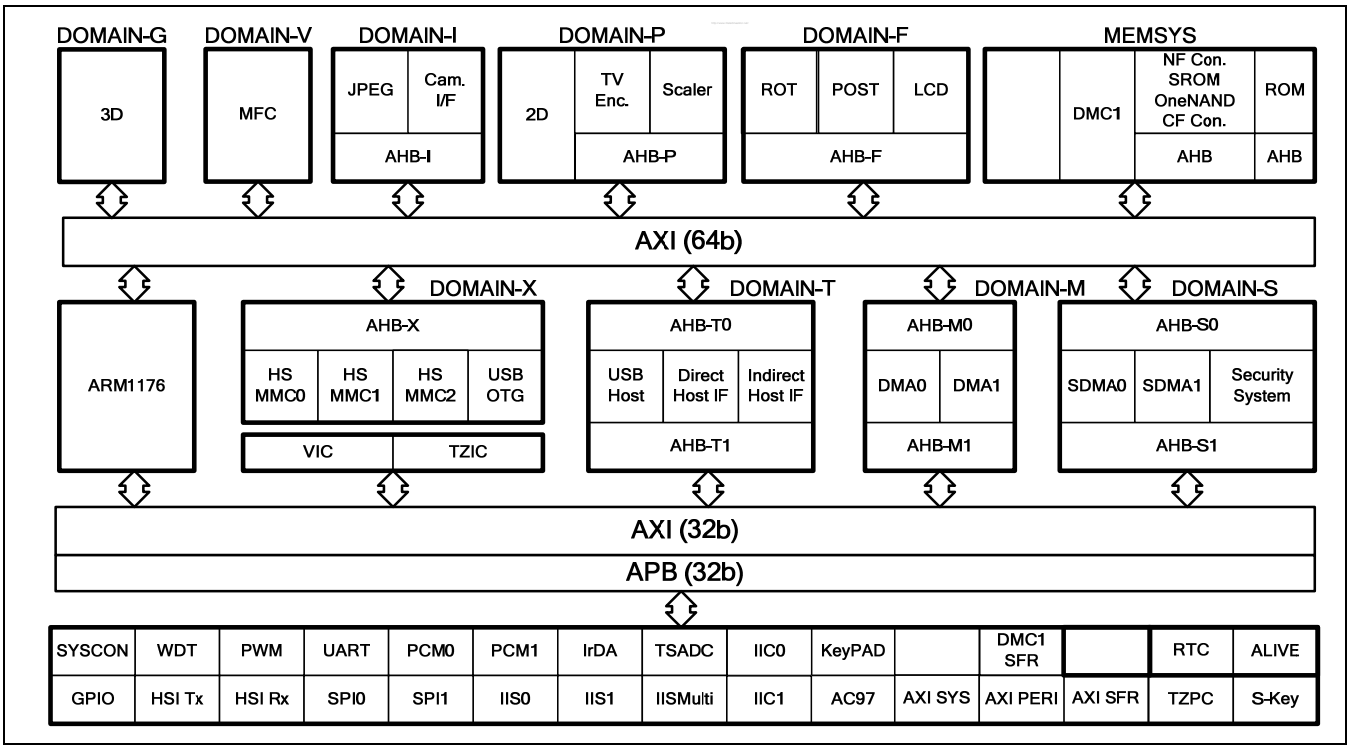


Figure 3-1. S3C6410X block diagram

3.3.2 CLOCK ARCHITECTURE

Figure 3-2 illustrates the block diagram of the clock generation module. The clock source selects between an external crystal (XXTIpll) and external clock (XEXTCLK). The clock generator consists of three PLLs (Phase Locked Loop) which generate high frequency clock signals up to 1.6GHz.

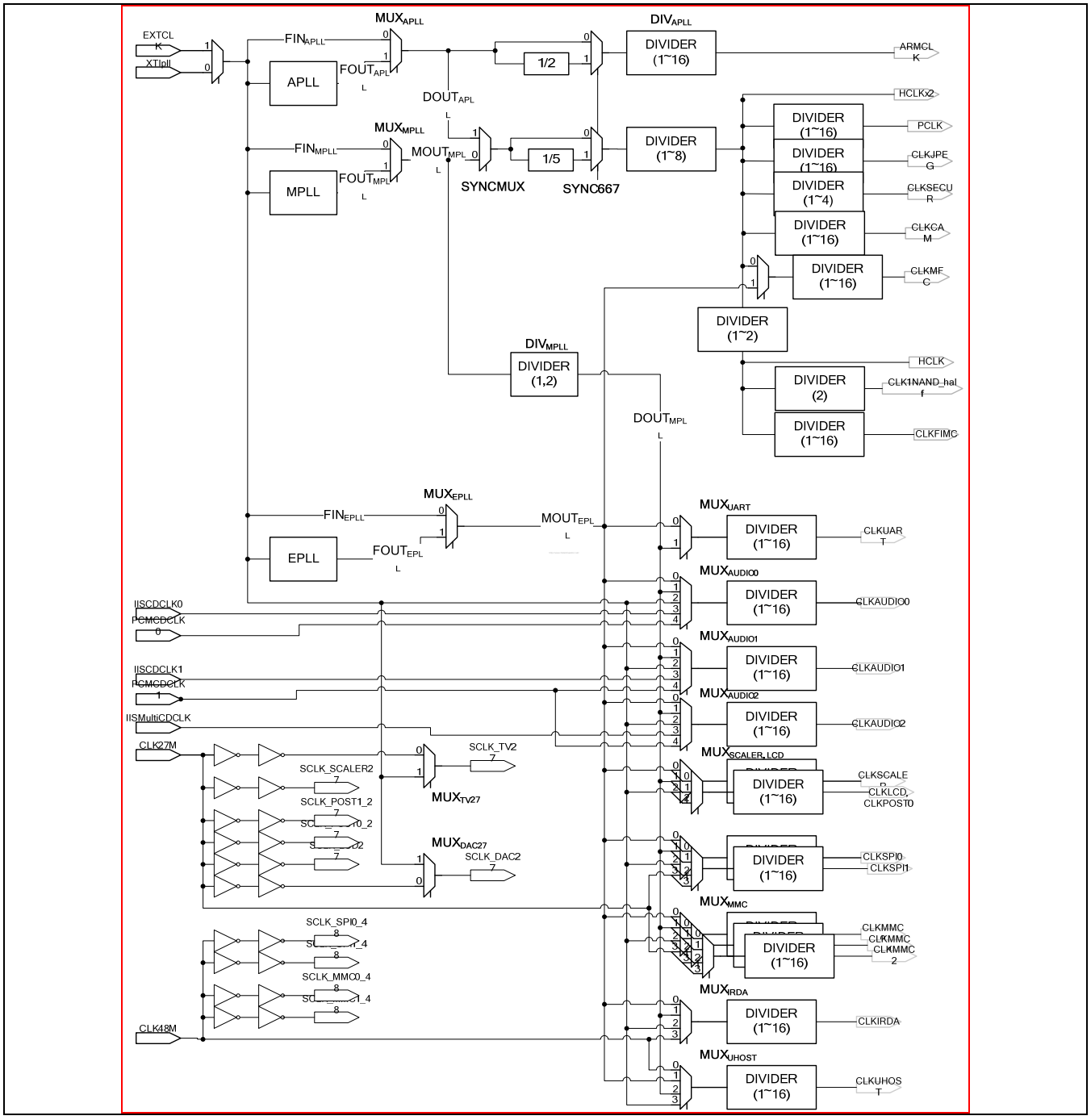


Figure 3-2. The block diagram of clock generator

3.3.3 CLOCK SOURCE SELECTION

Internal clocks will be generated using external clock source as illustrated in Table 3-1. The OM[4:0] pins determines the operating mode of S3C6410X when the external reset signal is asserted. As described in the table, the OM[0] selects the external clock source, i.e., if the OM[0] is 0, the XXTIpll (external crystal) is selected. Otherwise, XEXTCLK is selected.

Table 3-1. Device operating mode selection at boot-up

XSELNAND	OM[4:0]	GPN[15:13]	Boot Device	Function	Clock Source
1	0000X	XXX	RESERVED	RESERVED	XXTIpll if OM[0] is 0. XEXTCLK if OM[0] is 1.
1	0001X			RESERVED	
1	0010X			RESERVED	
1	0011X			RESERVED	
X	0100X		SROM(8bit)	-	
X	0101X		SROM(16bit)	-	
0	0110X		OneNAND[1)]	Don't use NAND Device	
X	0111X		MODEM	Don't use Xm0CSn2 for SROMC	
X	1111X	000	IROM[2)]	SD/MMC(CH0)	
0		001		OneNAND	
1		010		NAND(512Byte, 3-Cycle)	
1		011		NAND(512Byte, 4-Cycle)	
1		100		NAND(2048Byte, 4-Cycle)	
1		101		NAND(2048Byte, 5-Cycle)	
1		110		NAND(4096Byte, 5-Cycle)	
X		111		SD/MMC(CH1)	

Note 1) Only 6410X PoP D type doesn't support OneNAND booting.

Note 2) 6410X PoP A type doesn't support IROM booting based on NAND Flash. 6410X PoP D type doesn't support IROM booting based on OneNAND Flash.

The operating mode is mainly classified into six categories according to the boot device. The boot device can be among SROM, NOR, OneNAND, MODEM and Internal ROM. When NAND Flash device is used, XSELNAND pin must be 1, even if it is used as boot device or storage device. When OneNAND Flash device is used, XSELNAND must be 0, even if it is used as boot device or storage device. When NAND/OneNAND device is not used, XSELNAND can be 0 or 1.

3.3.4 PHASE LOCKED LOOP (PLL)

Three PLLs within S3C6410X, APLL, MPLL, and EPLL, synchronizes an output signal with a reference input clock in operating frequency and phase. In this application, it includes the following basic blocks as illustrated in Figure 3-3. The Voltage Controlled Oscillator (VCO) generates the output frequency proportional to input DC voltage. The pre-divider divides the input frequency (FIN) by P. The main divider divides the VCO output frequency by M, which is input to Phase Frequency Detector (PFD). The post scaler divides the VCO output frequency by S. The phase difference detector calculates the phase difference and the charge pump increases / decreases the output

voltage. The output clock frequencies of each PLL can be calculated

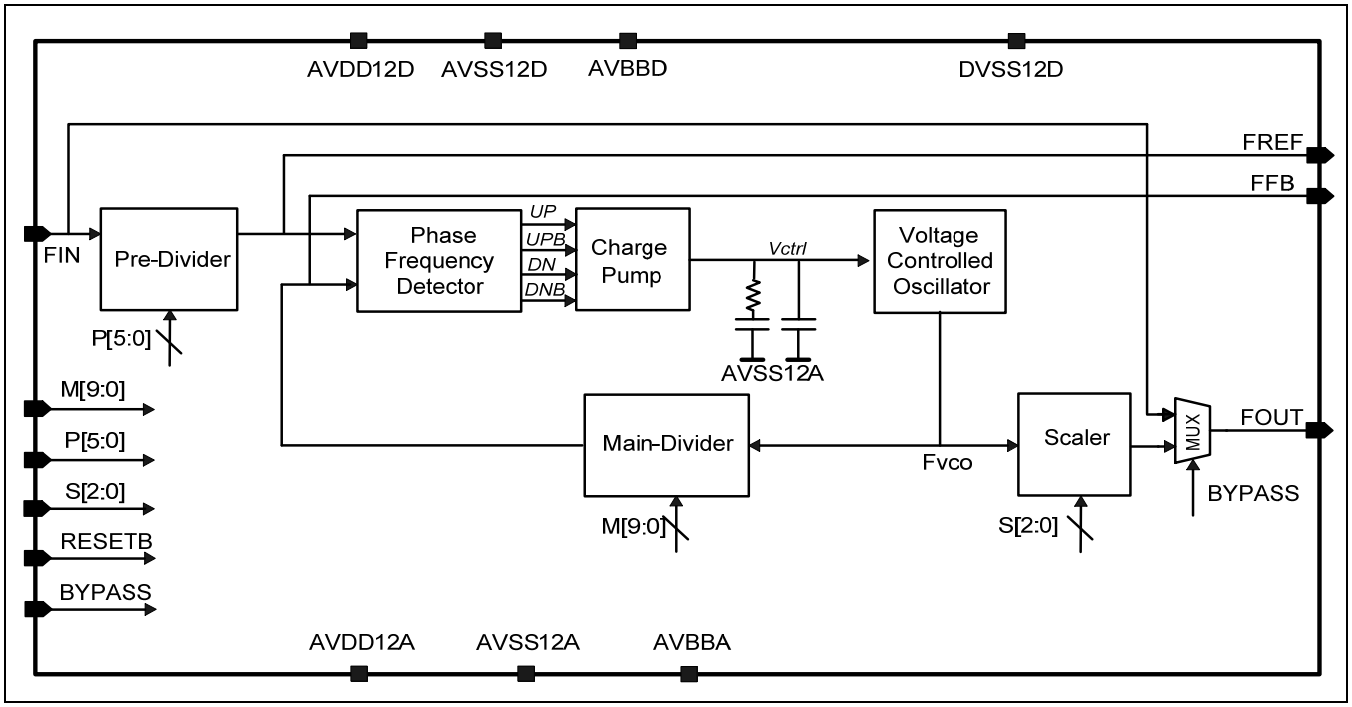


Figure 3-3. PLL block diagram (APLL, MPLL only)

3.3.4.1 Clock selection between PLLs and input reference clock

Figure 3-4 illustrates the clock generation logic. S3C6410X has three PLLs which are APLL for ARM operating clock, MPLL for main operating clock, and EPLL for special purpose. The operating clocks are divided into three groups. The first thing is ARM clock, which is generated from APLL. MPLL generates the main system clocks, which are used for operating AXI, AHB, and APB bus operation. The last group is generated from EPLL. Mainly, the generated clocks are used for peripheral IPs, i.e., UART, IIS, IIC, and etc.

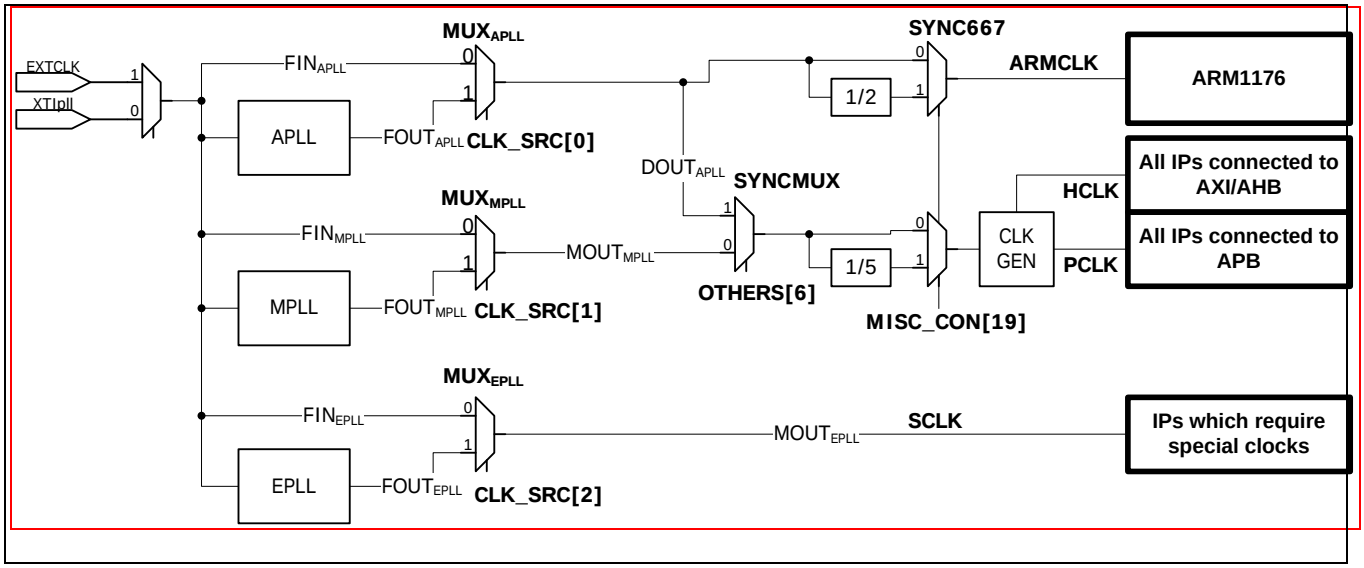


Figure 3-4. Clock generation from PLL outputs

The lowest three bits of CLK_SRC register control the source clocks of three groups. When the bit has 0, then the input clock is bypassed to the group. Otherwise, the PLL output will be applied to the group.

3.3.4.2 ARM and AXI/AHB/APB bus clock generation

ARM1176 processor of S3C6410X runs up to maximum 667MHz. The operating frequency can be controlled by the internal clock divider, DIV_{ARM}, without changing PLL frequency. The divider ratio varies from 1 to 16. ARM processor decreases the operating speed to reduce power dissipation.

S3C6410X consists of AXI bus, AHB bus, and APB bus to optimize the performance requirements. Internal IPs are connected to appropriate bus systems to meet their I/O bandwidth and operating performance. When they are attached to AXI bus or AHB bus, the operating speed can be up to maximum 133MHz. While they are attached to APB bus, the maximum operating speed can be up to 66MHz. Moreover, the bus speed between AHB and APB has high dependency to synchronize data transmission. Figure 3-5 illustrates the part of bus clock generation to meet the requirements of bus system clocks.

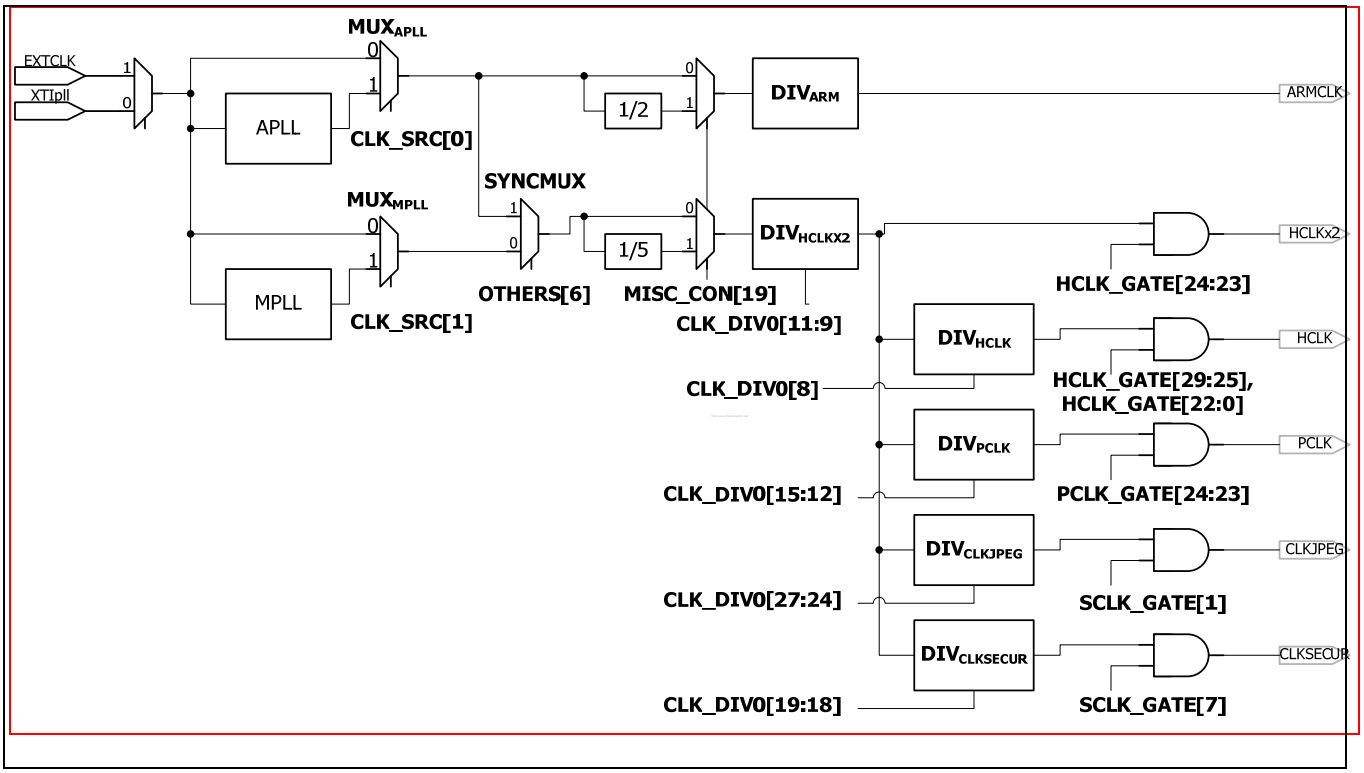


Figure 3-5. ARM and Bus clock generation

HCLKX2 clocks are supplied to two DDR controllers, DDR0 and DDR1, of S3C6410X. The operating speed can be up to maximum 266MHz to send and to receive data through DDR controllers. Each HCLKX2 clock can be masked independently to reduce redundant power dissipation on clock distribution network when the operation is not required. All AHB bus clocks are generated from DIV_{HCLK} clock divider. The generated clocks can be masked independently to reduce redundant power dissipation. HCLK_GATE register controls the mask operation of HCLKX2 and HCLK.

Low-speed interconnection IPs transfer data through APB bus system. APB clocks of them are running at up to 66MHz as described in the above section and generated from DIV_{PCLK} clock divider. They are also masked using PCLK_GATE register. As described, the frequency ratio between AHB clock and APB clock must be an even integer value. For example, if DIV_{HCLK} has 1 of CLK_DIV0[8], then DIV_{PCLK} must be 1, 3, ... of CLK_DIV0[15:12]. Otherwise, the IPs on APB bus system cannot transfer data correctly.

JPEG and security sub-system on AHB bus system cannot be running at 133MHz.AHB clocks are independently generated with $DIV_{CLKJPEG}$ and $DIV_{CLKSECUR}$. Therefore, they have same constraints as APB clock, i.e., the operating frequency ratio must be even number ($DIV_{CLKJPEG}$ and $DIV_{CLKSECUR}$ must be 1, 3, ... of CLK_DIV0[27:24] and CLK_DIV0[19:18], respectively).

Table 3-2 illustrates the recommended clock divider ratio.

Table 3-2. Typical value setting for clock dividers (SFR setting value / output frequency)

APLL	MPLL	DIV_{ARM}	DIV_{HCLKX2}	DIV_{HCLK}	DIV_{PCLK}	$DIV_{CLKJPEG}$	$DIV_{CLKSECUR}$
266MHz	266MHz	0 / 266MHz	0 / 266MHz	1 / 133MHz	3 / 66MHz	3 / 66MHz	3 / 66MHz
400MHz	266MHz	0 / 400MHz	0 / 266MHz	1 / 133MHz	3 / 66MHz	3 / 66MHz	3 / 66MHz
533MHz	266MHz	0 / 533MHz	0 / 266MHz	1 / 133MHz	3 / 66MHz	3 / 66MHz	3 / 66MHz
667MHz	266MHz	0 / 667MHz	0 / 266MHz	1 / 133MHz	3 / 66MHz	3 / 66MHz	3 / 66MHz

The divider for ARM independently uses the output clock of APLL and there is no constraint for clock divider value as described in the above table.

3.3.4.3 MFC clock generation

The MFC block requires a special clock in addition to HCLK and PCLK. The additional clock is generated as shown in Figure 3-6.

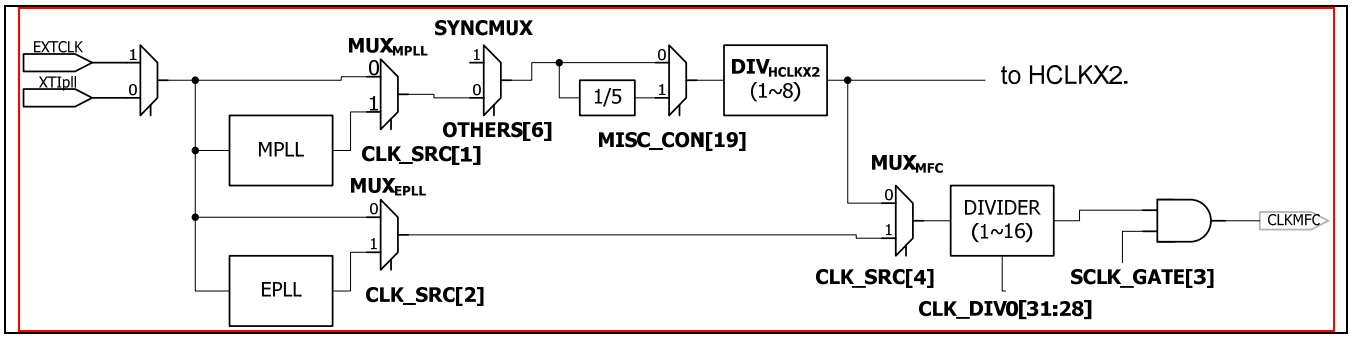


Figure 3-6. MFC clock generation

The source clock is selected between HCLKX2 and $MOUT_{EPLL}$. The operating clock is divided using HCLKX2. The operating frequency of HCLKX2 is fixed as 266MHz by default. Thus, CLK_DIV0[31:28] must be 4'b0001 to generate 133MHz. When MFC is not required full-performance, there are two way to decrease the operating frequency. The first way is to use output clock of EPLL when CLK_SRC[4] is set 1. Generally, EPLL is used for audio clocks and the output clock will be lower than output frequency of MPLL. Another way is to adjust clock divider ratio of CLK_DIV0[31:28]. Using this value, the lower frequency can be applied to MFC block using CLK_SRC[4] field to reduce redundant power dissipation. Since the output frequency of EPLL is independent of HCLKX2 or HCLK.

3.3.4.4 Camera I/F clock generation

Figure 3-7 illustrates the clock generator for the camera interface. All data for camera interface is transferred/received based on this clock. The maximum operating clock is up to 133MHz.

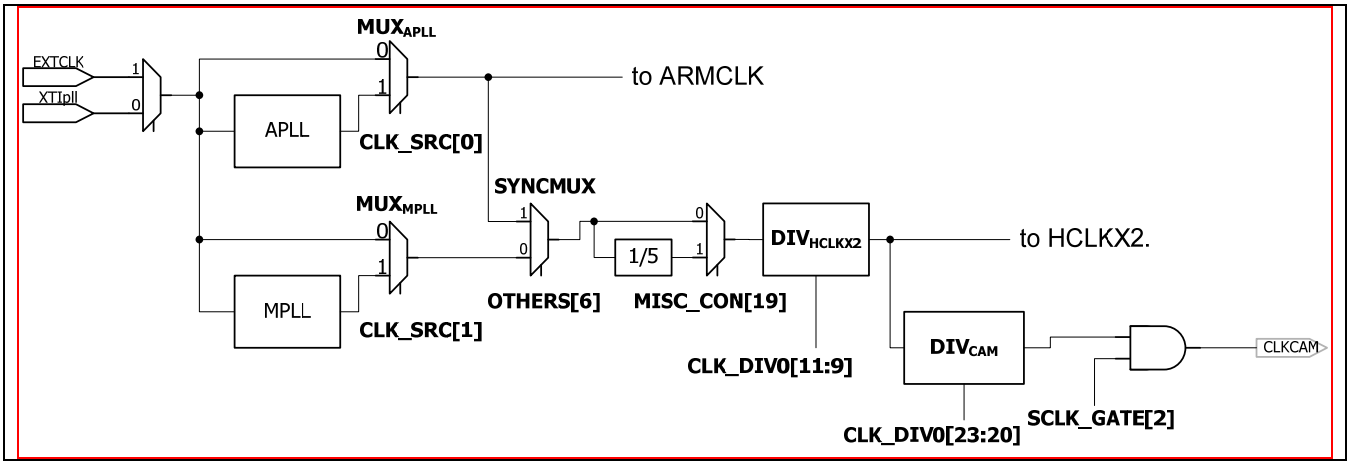


Figure 3-7. Camera I/F clock generation

3.3.4.5 Clock generation for display (POST, LCD, and scaler)

Figure 3-8 illustrates the clock generator for display blocks. Usually LCD controller requires image post-processor and scaler logic. The operating clocks can be independently controlled with this clock generator. CLKLCD and CLKPOST are connected to LCD controller and post-processor, respectively, within domain-F. CLKSCALER is connected to scaler block within domain-P.

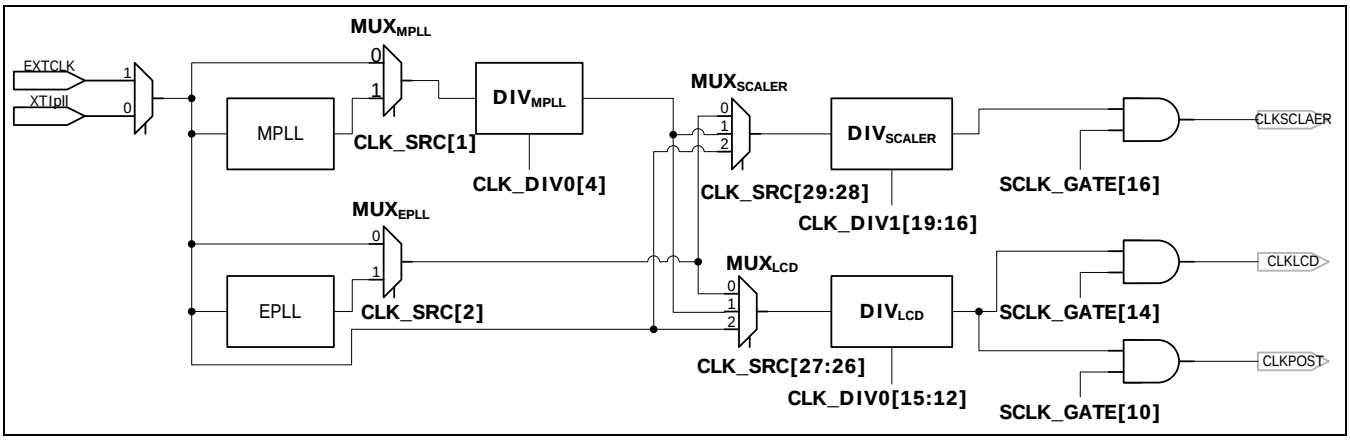


Figure 3-8. Display clock generation

3.3.4.6 Clock generation for audio (IIS and PCM)

Figure 3-9 generates special clocks for audio interface logics, which include IIS and PCM. S3C6410X has two IIS channels and two PCM channels. It supports only two channels at any time. Generally, EPLL generates one special clock for an audio interface. If S3C6410X requires two independent clock frequency, i.e., there is no integer relationship between two audio interfaces, the remaining clock can be supplied directly through external oscillators or using MPLL.

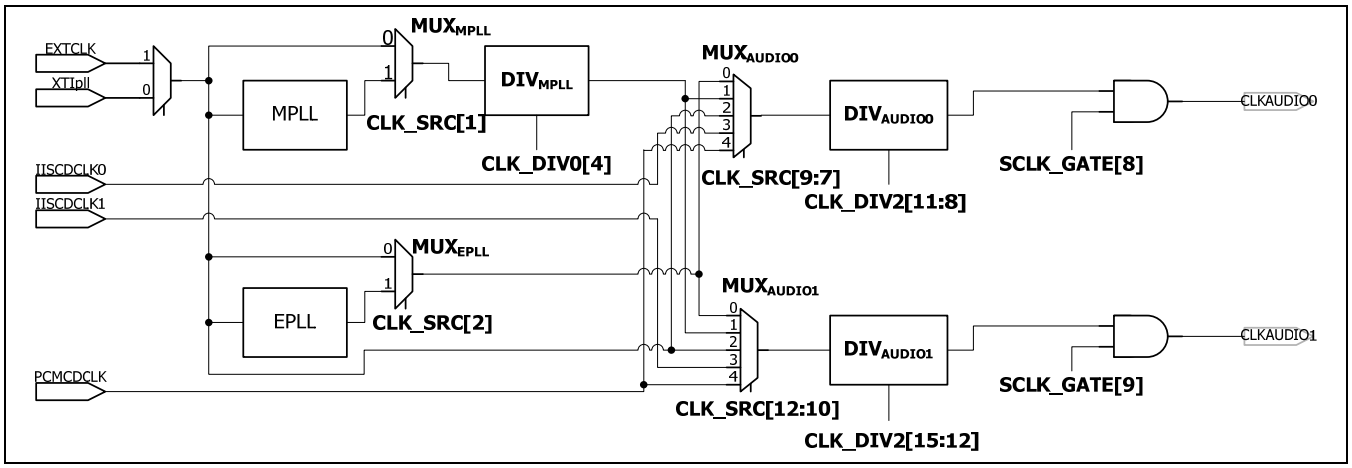


Figure 3-9. Audio clock generation

3.3.4.7 Clock generation for UART, SPI, and MMC

Figure 3-10 illustrates the clock generator for UART, SPI and MMC. There is one additional clock source, CLK27M, to give more flexibility.

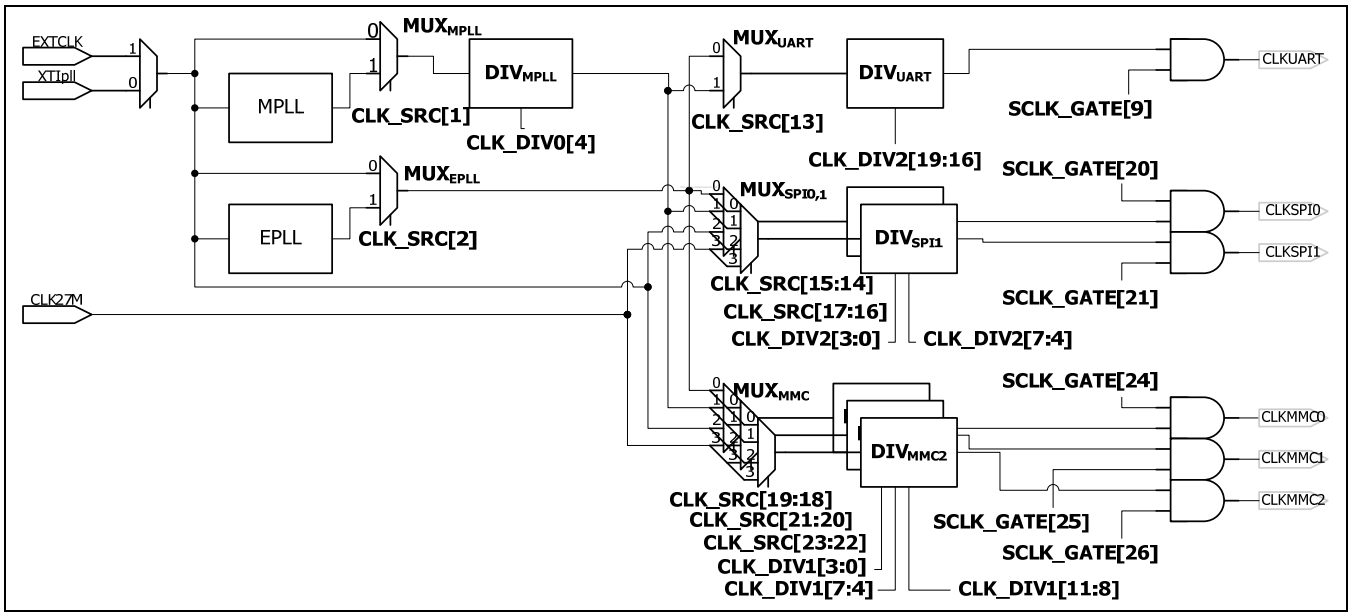


Figure 3-10. UART/SPI/MMC clock generation

3.3.4.8 Clock generation for IrDA, USB host

Figure 3-11 illustrates the clock generator for IrDA and USB host. Usually USB interface requires 48MHz operating clock and the direct 48MHz path from USB-OTG is inserted for this purpose as illustrated in Figure 3-11.

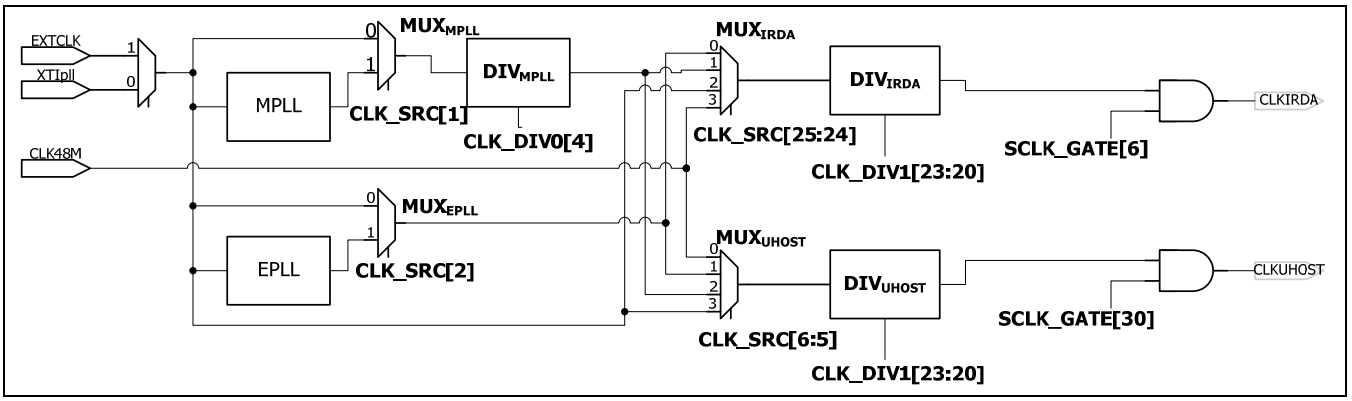


Figure 3-11. IrDA/USB host clock generation

3.3.4.9 Clock ON/OFF control

As illustrated in the above figures, HCLK_GATE, PCLK_GATE, and SCLK_GATE control the clock operation. If a bit is set, the corresponding clock will be supplied through each clock divider. Otherwise, it will be masked.

HCLK_GATE controls HCLK for each IPs. The AHB interface logic of each IP is masked independently to reduce dynamic power consumption. PCLK_GATE controls PCLK for each IPs. Certain IPs requires special clocks to operate correctly. The clocks are controlled by SCLK_GATE.

3.3.4.10 Clock output

S3C6410X has clock output port, which generate internal clock. This clock is used for regular interrupt or debugging purpose. For more information, please refer to CLK_OUT register.

3.3.5 SYNCHRONOUS 667MHZ OPERATING MODE

The clock ratio between ARMCLK and HCLK must be integer to use synchronous interface between ARM core and AXI bus interface. S3C6410X does not have any limitation up to 533MHz synchronous interface, i.e., ARMCLK = 533MHz, HCLKX2= 266MHz, HCLK = 133MHz. However, there is some constraints over 533MHz, typically 667MHz interface. The supported clock ratio is only 1:2.5:5 (ARMCLK = 667MHz, HCLKX2=266MHz, HCLK = 133MHz).

3.3.5.1 Clock divider structure

Figure 3- 12 shows the clock structure to generate ARMCLK, HCLKX2, and HCLK. As shown in Figure 3- 12, clock divider consists of two parts, pre-dividers and post divers. There are two pre-dividers to divide by 2 and 5, which are fixed and not software configurable. The post-dividers can be configured by software using CLK_DIV0 register. When SYNC667MHz of MISC_CON is set, two pre-dividers are running and generated divided clocks from APLL output. If FOUT of APLL is 1.33MHz, then the pre-divider output of ARMCLK side is 667MHz, and that of HCLK side is 266MHz. In this way, the input clock frequency of post-dividers can be lowered. The input clocks of post-dividers are divided by configuration of CLK_DIV0 register.

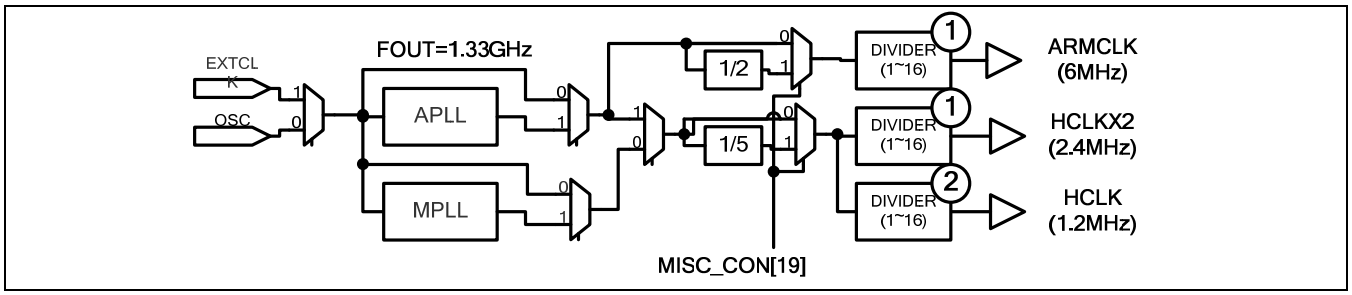
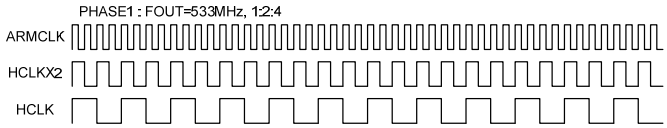


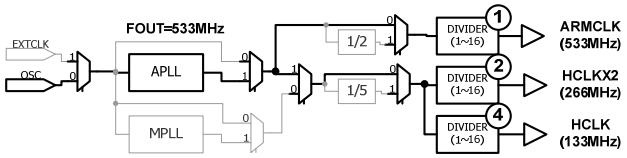
Figure 3- 12 Clock divider structure

3.3.5.2 Transition to synchronous 667MHz operating mode

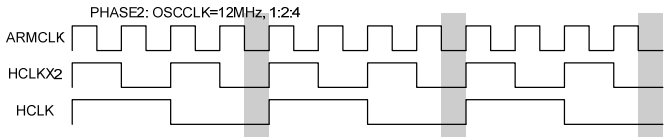
S3C6410X can be use synchronous 667MHz operating mode through the following 5 steps.

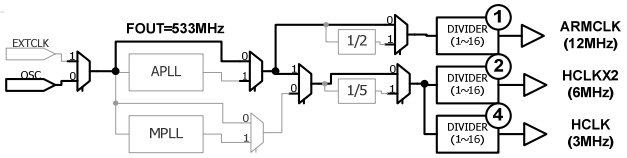
1. STEP1-Normal operating mode: SYNC667 field is cleared and there is no limitation for clock dividers. Typical clock divider ratio will be 1:2:4. (ARMCLK = 533MHz, HCLKX2 = 266MHz, HCLK = 133MHz)



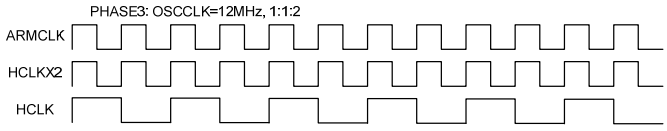


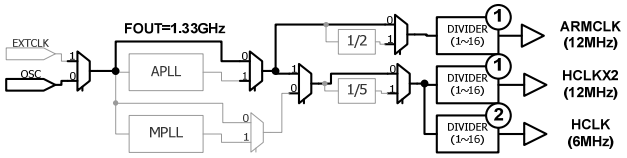
2. STEP2-OSC mode: before set SYNC_667MHZ field, the operating clock of S3C6410X must be external oscillator. (set APLL_SEL of CLK_SRC as '0') Clock divider ratio is unchanged, but the operating frequencies are 12MHz, 6MHz, and 3MHz.



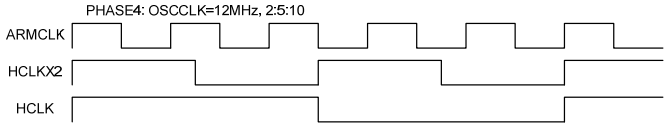


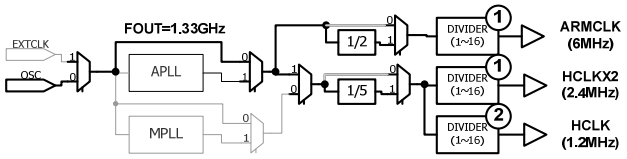
3. STEP3-DIV112 mode: in this step, clock divider ratio must be set as 1:1:2 from 1:2:4.



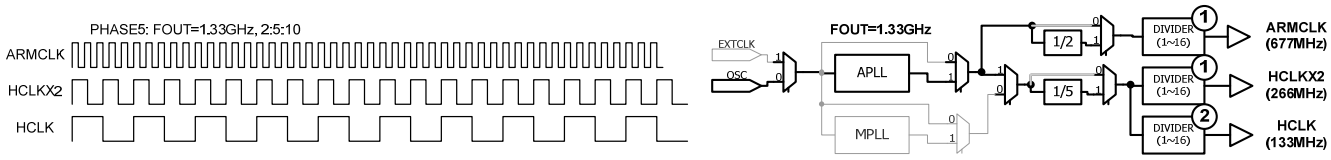


4. STEP4-SYNC667 low frequency mode: set SYNC_667MHZ field. Now, overall clock divider ratio will be 2:5:10. If input oscillator is 12MHz, then ARMCLK, HCLKX2, and HCLK will be 6MHz, 2.4MHz, and 1.2MHz (2:5:10), respectively.





5. STEP5-SYNC667 mode: if FOUT of APLL is 1.33GHz, then the output will be 667MHz, 266MHz, and 133MHz. S3C6410X runs at 667MHz synchronous interface mode.



When exiting from SYNC667 mode, the sequence is reverse order (from STEP5 to STEP1).

3.3.5.3 Limitation of synchronous 667MHz operating mode

There are some limitations as follows when S3C6410X runs at SYNC667 mode.

- Cannot use other clock divider ratio at SYNC667 mode. The fields, ARM_RATIO, HCLKX2_RATIO, and HCLK_RATIO of CLK_DIV0 must be 0x0, 0x0, and 0x1, respectively.
- Use asynchronous interface mode at SYNC667 mode if ARM_RATIO, HCLKX2_RATIO, and HCLK_RATIO of CLK_DIV0 are different values from 0x0, 0x0, and 0x1.
- Cannot use DVFS(Dynamic Voltage Frequency Scale) at synchronous interface mode. As described the above, SYNC667 mode does not support dynamic clock ratio change. Interface mode must be asynchronous interface mode before changing clock divider ratio.
- Cannot enter into STOP/D-STOP/SLEEP mode. The SYNC667 mode must be cleared before entering into STOP/D-STOP/SLEEP mode.

3.3.6 LOW POWER MODE OPERATION

S3C6410X supports low power application through low power mode operation as illustrated in Table 3-3. There are four power states, which are normal state, retention state, power gating state, and power off state. All internal logics including F/Fs and memory are running at normal state. Retention state reduces unwanted power consumption during STOP/DEEP-STOP mode, however, retains previous states and supports fast wake-up time from STOP/DEEP-STOP mode. Some blocks, which are DOMAIN-G, DOMAIN-V, DOMAIN-I, DOMAIN-P, DOMAIN-F, and DOMAIN-S, have no state retention feature. They can be power gating to reduce power consumption through an internal power switch circuitry. The response time of the internal circuitry, about several usec, is faster than that of an external power regulator. In SLEEP mode, an external regulator will be OFF to reduce power consumption and S3C6410X minimizes power consumption and lose all information except ALIVE and RTC block. Table 3-3 summarizes four power states for S3C6410X.

Table 3-3. Four power states for S3C6410X

State	External regulator	Internal F/F	Internal Memory
Normal	ON	Normal operation	Normal operation
Retention	ON	Retain previous state	Retain previous state
Power gating	ON	Lost previous state	Lost previous state
Power off	OFF	Lost previous state	Lost previous state

3.3.6.1 Power domain in S3C6410X

S3C6410X consists of several power domains as illustrated in Figure 3-12. Sub-power domains, DOMAIN-G, DOMAIN-V, DOMAIN-I, DOMAIN-P, DOMAIN-F, and DOMAIN-S, are controlled by NORMAL_CFG and STOP_CFG. When S3C6410X runs at NORMAL or IDLE mode, NORMAL_CFG controls them. If the controlled bit is clear, corresponding block changes power-gating mode and lost previous state. Therefore, user software must store internal state before clearing the corresponding bit. When S3C6410x changes to STOP or DEEP-STOP mode, sub-power domains automatically change to power-gating mode.

STOP_CFG only controls ARM1176 and top module. If user software requires fast response time, the memory and logic of ARM1176 must be set and retained during STOP mode. In this case, the logic power of top block must be set and the memory power of top block can be configured. Otherwise, S3C6410X may not return to the previous state. ARM1176 leakage current can be minimized when ARM1176 power is OFF (bit 29 and 17 of STOP_CFG are '0'.) This configuration is called as DEEP-STOP mode. The software must store program status information including internal registers, CPSR, SPSR and etc, before going to DEEP-STOP mode.

3.3.6.2 NORMAL/IDLE mode

In NORMAL mode, ARM1176 core, media co-processors, and all peripherals can operate fully. Typical system-bus operating frequency is up to 133MHz. The clock to each media co-processors and peripherals can be stopped selectively by software to reduce power consumption. The ON/OFF clock gating of the individual clock source of each IP block is performed by controlling of each corresponding clock enable bit, which is specified by HCLK_GATE, PCLK_GATE, and SCLK_GATE configuration registers.

In IDLE mode, ARM1176 is stopped without any change of other IPs. Typically, ARM1176 waits a wake-up event to return to NORMAL mode.

All IPs can run at maximum operating frequency at NORMAL/IDLE mode. When some IPs is not required to run, S3C6410X can cut supply power using internal power-gating circuitry. As illustrated in Figure 3-12, five power domains can be independently controlled with NORMAL_CFG configuration register. When all functional IPs are not required to run, software can cut supply power of the corresponding power domain, which is highlighted in grey color in Figure 3-12. All internal status of the corresponding domain will be lost after the corresponding power domain is OFF. Therefore, user software must store all information, which is required to restore internal state.

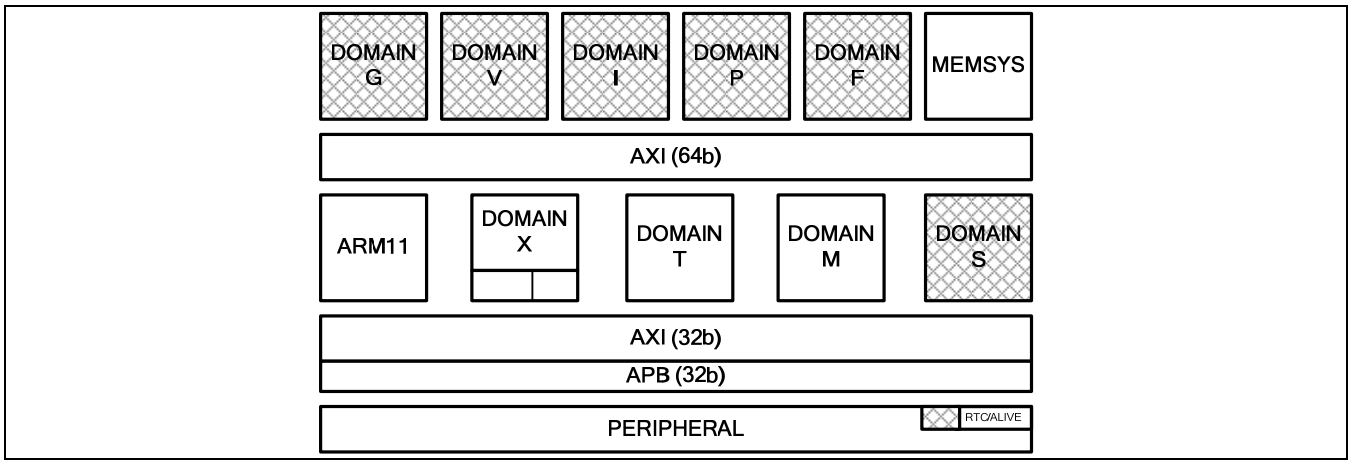


Figure 3-13. Power domains at NORMAL/IDLE mode (grey colored domain can be ON/OFF by NORMAL_CFG configuration register)

3.3.6.3 STOP mode

In STOP mode, sub-power domains, which are denoted as black boxes, are OFF with internal power-gating circuitry as illustrated in Figure 3-13. Other blocks which are denoted by gray boxes and ARM1176, are retaining the previous state (Retention state). Thus, when external wake-up events occur, internal states are recovered without software assistance. STOP mode gives fast response time, but requires a little leakage current. (Please refer to the electrical specification for the detailed information.)

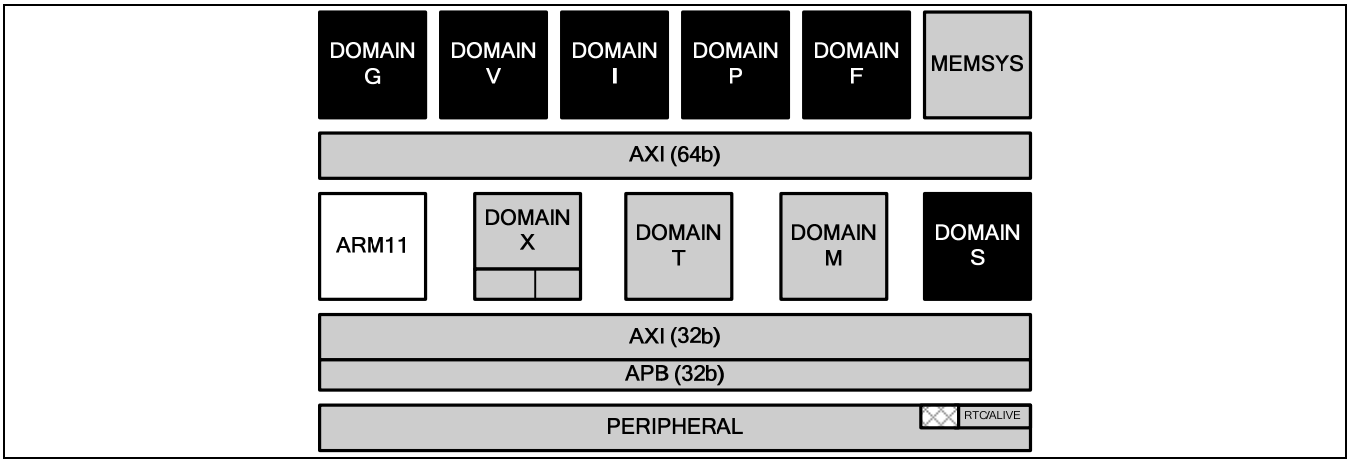


Figure 3-14. Power domains at STOP mode (grey domains represent state-retained domain and black domains represent power-gating domain)

STOP mode entering sequence is as follows:

1. User software sets PWR_CFG[6:5] as STOP mode
2. User software generates STANDBYWFI signal by MCR instruction (MCR p15, 0, Rd, c7, c0, 4)
3. SYSCON requests bus controller to finish current AHB bus transaction.
4. AHB bus controller sends acknowledge to SYSCON after current bus transaction is completed.
5. SYSCON requests DOMAIN-V to finish current AXI-bus transaction.
6. AXI bus controller sends acknowledge to SYSCON after current bus transaction is completed.
7. SYSCON requests external memory controllers to enter into self-refresh mode, since the contents in the external memory must be preserved during STOP mode.
8. The memory controllers send acknowledges when they are self-refresh mode.
9. SYSCON changes clock source from PLL output to external oscillator if PLL is used.
10. SYSCON disables power-gating circuitries to eliminate leakage current. (only applied for DEEP-STOP mode)
11. SYSCON disables PLL operations and crystal oscillator.

To exit from STOP mode, all wake-up sources except normal interrupts are available. The wake-up sequence from STOP mode is as follows:

1. SYSCON asserts reset signal of ARM1176 during transition period to NORMAL mode (only applied for DEEP-STOP mode)
2. SYSCON enables crystal oscillator and wait for oscillator stable period, which is configured by OSC_STABLE.
3. SYSCON enables clock-gating circuitries to supply operating power and wait for stable time, which is configured by MTC_STABLE. (only applied for DEEP-STOP mode)

4. SYSCON enables PLL logics and wait for PLL locking period, which is configured by A/M/EPLL_LOCK.
5. SYSCON changes clock source from external oscillator to PLL output if PLL is used.
6. SYSCON releases self-refresh mode requests to memory controllers.
7. The memory controllers send acknowledges when they are ready.
8. SYSCON releases AXI/AHB bus down request.
9. SYSCON releases reset signal of ARM1176 (only applied for DEEP-STOP mode)

3.3.6.4 DEEP-STOP mode

Most mobile applications require longer standby period and reasonable response time from low power state. DEEP-STOP mode is focused for the requirements. External power ON/OFF control generally requires long transition time (~3ms).

Figure 3-14 illustrates the status at DEEP-STOP mode. The black boxes denote power-gating blocks and eliminate leakage current during DEEP-STOP mode while top module retains the previous states as STOP mode.

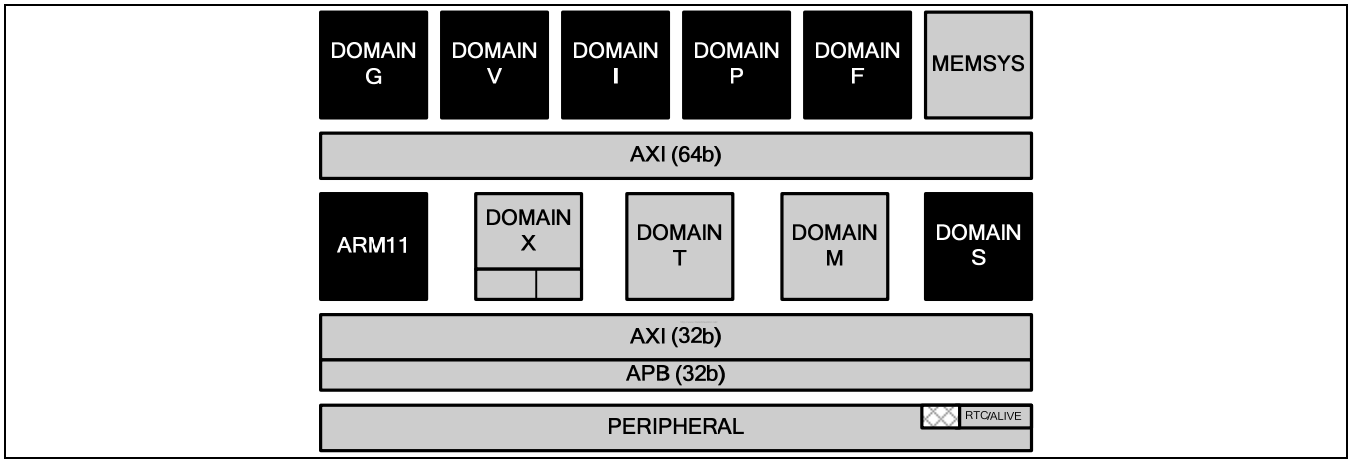


Figure 3-15. Power domains at DEEP-STOP mode (ARM11 is OFF and lost internal state)

Since the entering and exiting sequence is similar to STOP mode, refer to STOP mode sequence for entering and exiting sequence of DEEP-STOP mode.

3.3.6.5 SLEEP mode

In SLEEP mode, all hardware logics except ALIVE and RTC blocks, are power-OFF using external power-regulator. SLEEP mode supports the longest standby period, while user software must store all internal status to external storage devices. ALIVE block waits an external wake-up event and RTC stores time information. User software can configure wake-up source and the status of I/O pins with GPIO configuration.

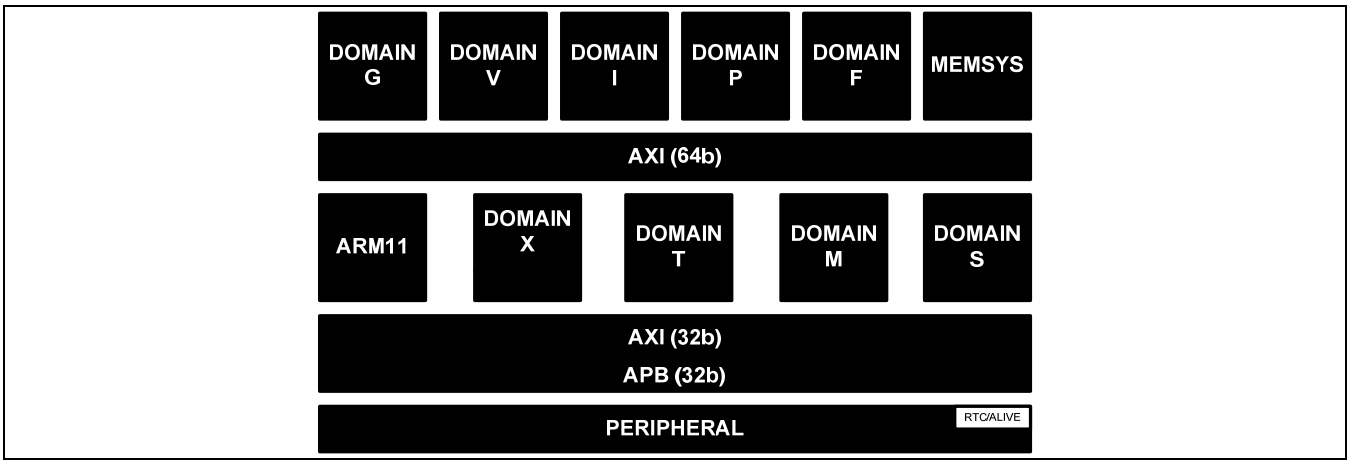


Figure 3-16. Power domains at SLEEP mode (only ALIVE and RTC keep internal state)

SLEEP mode entering sequence is as follows:

1. User software sets PWR_CFG[6:5] as SLEEP mode
2. User software generates STANDBYWFI signal by MCR instruction (MCR p15, 0, Rd, c7, c0, 4)
3. SYSCON requests bus controller to finish current AHB bus transaction
4. AHB bus controller sends acknowledge to SYSCON after current bus transaction is completed.
5. SYSCON requests DOMAIN-V to finish current AXI-bus transaction.
6. AXI bus controller sends acknowledge to SYSCON after current bus transaction is completed.
7. SYSCON requests external memory controllers to enter into self-refresh mode, since the contents in the external memory must be preserved during SLEEP mode.
8. The memory controllers send acknowledges when they are self-refresh mode.
9. SYSCON changes clock source from PLL output to external oscillator if PLL is used.
10. SYSCON disables PLL operations and crystal oscillator.
11. Finally, SYSCON disables an external power source for internal logic by asserting XPWRRGTON pin to low state. XPWRRGTON signal controls an external regulator.

The exiting sequence is as follows:

1. SYSCON enables an external power source by asserting XPWRRGTON pin to high state and waits for the stable time, which is configured by PWR_STABLE.
2. SYSCON generates system clocks including HCLK, PCLK, and ARMCLK.
3. SYSCON releases system reset signals including HRESETn and PRESETn.
4. SYSCON releases ARM reset signals.

3.3.6.6 Wakeup

Table 3-4 illustrates various wake-up sources from low power state, IDLE, (DEEP) STOP, and SLEEP. According to the low power state, different wake-up sources are available.

Table 3-4. Power mode wake-up sources

Power mode			Wakeup sources
IDLE			All interrupt sources
	STOP		MMC0, MMC1, MMC2
			TS ADC
		SLEEP	External interrupt sources
			RTC Alarm
			TICK
			Keypad interrupt
			MSM (MODEM)
			Battery Fault
			HSI

3.3.6.7 reset

S3C6410X has three types of reset signals and SYSCON can place the system into one of three resets.

- Hardware reset: It is generated by asserting XnRESET. It is an uncompromised, ungated, total and complete reset that is used when you do not require information in system any more. It fully initializes all system.
- Watchdog reset: It is generated by a special hardware block, i.e., watchdog timer. When the system is hanged due to an unpredictable software error, the hardware block monitors internal hardware status and generates reset signal to escape from this status.
- Wakeup reset: It is generated when S3C6410X wake up from SLEEP mode. Since internal hardware states are not available any more after SLEEP mode, they must be initialized.

3.3.6.8 Hardware reset

The hardware reset is invoked when XnRESET pin is asserted and all units in the system (except RTC) are reset to pre-defined states. During this period, the following actions occur.

- All internal registers and ARM1176 core go to the pre-defined reset states.
- All pins get their reset state.
- XnRSTOUT pin is asserted when XnRESET is asserted.

XnRESET is un-maskable and is always enabled. Upon assertion of XnRESET, S3C6410X enters into reset state regardless of the previous mode. XnRSET must be held long enough to allow internal stabilization and propagation of the reset state to enter proper reset state.

Power regulator for S3C6410X must be stable prior to the deassertion of XnRESET. Otherwise, it may damage S3C6410X and the operation is unpredictable. Figure 3-17 is the timing diagram of power-on reset and pll turn-on sequence.

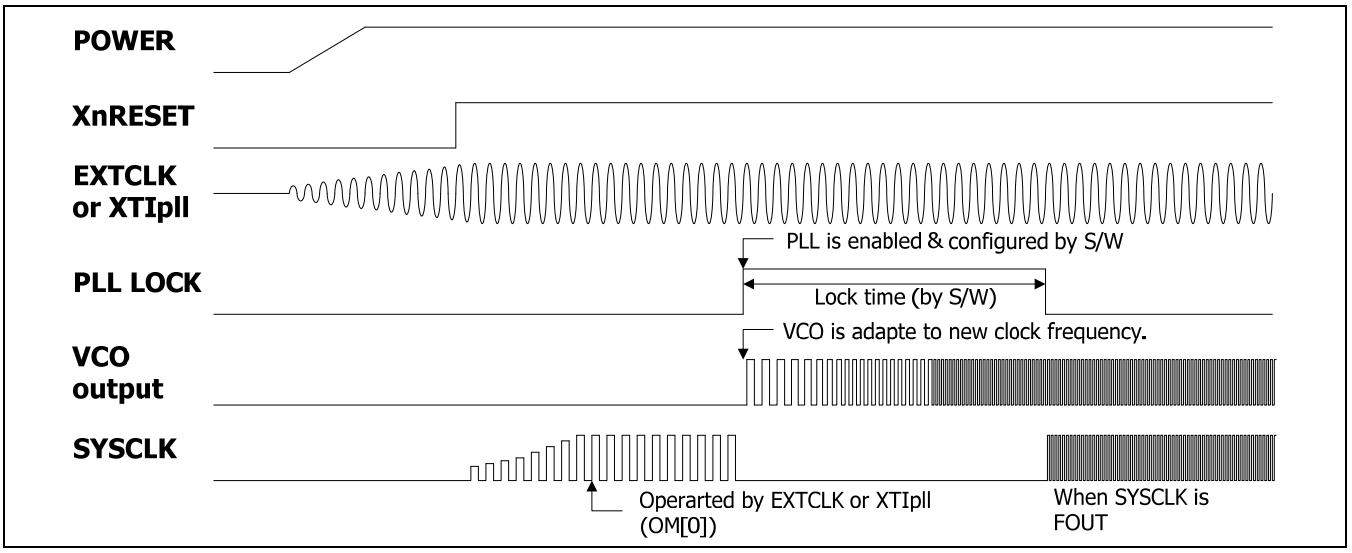


Figure 3-17. Power-on reset sequence

3.3.6.9 Watchdog reset

Watchdog reset is invoked when a software hang-up. Then, the software cannot initialize a register within WDT and WDT makes time-out signals for watchdog reset. As the occurrence of watchdog reset means that system has fatal problem, it behaves like external reset except reset status register. During the watchdog reset, the following actions occur:

- All blocks except reset status register in ALIVE block go to their pre-defined reset state.
- All pins get their reset state.
- The nRSTOUT pin is asserted during watchdog reset.

Watchdog reset can be activated in NORMAL and IDLE mode, since WDT can generate time-out signal. It is invoked when watchdog timer and reset are enabled. Then, the following sequence occurs:

1. WDT generate time-out signal.
2. SYSCON invokes reset signals and initialize internal IPs.
3. The reset including nRSTOUT will be asserted until the reset counter, RST_STABLE, is expired.

3.3.6.10 Wakeup reset

Wakeup reset is invoked when S3C6410 is woken-up from SLEEP by a wakeup event. The details are described in SLEEP mode section.

Table 3-5. Register initialization due to various resets

Block	Registers	XnRESET	Watchdog	Wakeup from SLEEP
SYSCON	PWR_CFG, EINT_MASK, NORMAL_CFG, STOP_CFG, SLEEP_CFG, OSC_FREQ, OSC_STABLE, PWR_STABLE, FPC_STABLE, MTC_STABLE, OTHERS, WAKEUP_STAT, BLK_PWR_STAT, INFORM0, INFORM1, INFORM2, INFORM3	X	O	O
RTC	RTCCON, TICCNT, RTCALM, ALMSEC, ALMMIN, ALMHOUR, ALMDAY, ALMMON, ALMYEAR, RTCRST	X	O	O
GPIO	GPICONSLP, GPIPUDSLP, GPJCONSLP, GPJPUDSLP, GPKCON0, GPKCON1, GPKDAT, GPKPUD, GPLCON0, GPLCON1, GPLDAT, GPLPUD, GPMCON, GPMDAT, GPMPUD, GPNCON, GPNDAT,GPNPUD, GPOCON, GPOPUD, GPPCON, GPPPUD, GPQCON, GPQPUD, EINT0CON0, EINT0CON1, EINT0FLTCON0, EINT0FLTCON1, EINT0FLTCON2, EINT0FLTCON3, EINT0MASK, EINT0PEND, SPCONSLP, SLPEN	X	O	O
Others	-	O	O	O

3.3.7 MISCELENEOUS.

There are several registers to control S3C6410X. S3C6410X consists of several AHB buses and one-AXI components. Several master-IPs are connected in AHB buses. Generally, bus utilization is the most important performance parameter in a bus system. Since the arbitration scheme and priority heavily affect on it, most bus system can configure the scheme and priority. S3C6410X also supports two bus arbitration schemes and several priority configurations. AHB_CONx registers control AHB bus arbitration scheme and priority.

S3C6410X supports TrustZone feature of ARM1176. If DMA operation is not protected properly, the contents of external memory will be broken by unprivileged software's. Unprivileged access can be protected by SDMA_SEL. The registers within SYSCON have protection scheme using security scheme. Therefore, the access having a proper security level can change SDMA_SEL values and protects the access of unprivileged software.

3.4 REGISTER DESCRIPTION

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

3.4.1 MEMORY MAP

The followings highlight the 34 registers within system controller.

Register	Address	R/W	Description	Reset Value
APLL_LOCK	0x7E00_F000	R/W	Control PLL locking period for APLL	0x0000_FFFF
MPLL_LOCK	0x7E00_F004	R/W	Control PLL locking period for MPLL	0x0000_FFFF
EPLL_LOCK	0x7E00_F008	R/W	Control PLL locking period for EPLL	0x0000_FFFF
APLL_CON	0x7E00_F00C	R/W	Control PLL output frequency for APLL	0x0190_0302
MPLL_CON	0x7E00_F010	R/W	Control PLL output frequency for MPLL	0x0214_0603
EPLL_CON0	0x7E00_F014	R/W	Control PLL output frequency for EPLL	0x0020_0102
EPLL_CON1	0x7E00_F018	R/W	Control PLL output frequency for EPLL	0x0000_9111
CLK_SRC	0x7E00_F01C	R/W	Select clock source	0x0000_0000
CLK_DIV0	0x7E00_F020	R/W	Set clock divider ratio	0x0105_1000
CLK_DIV1	0x7E00_F024	R/W	Set clock divider ratio	0x0000_0000
CLK_DIV2	0x7E00_F028	R/W	Set clock divider ratio	0x0000_0000
CLK_OUT	0x7E00_F02C	R/W	Select clock output	0x0000_0000
HCLK_GATE	0x7E00_F030	R/W	Control HCLK clock gating	0xFFFF_FFFF
PCLK_GATE	0x7E00_F034	R/W	Control PCLK clock gating	0xFFFF_FFFF
SCLK_GATE	0x7E00_F038	R/W	Control SCLK clock gating	0xFFFF_FFFF
MEM0_CLK_GATE	0x7E00_F03C	R/W	Control MEM0 clock gating	0xFFFF_FFFF
RESERVED	0x7E00_F040~ 0x7E00_F0FC	-	RESERVED	-
AHB_CON0	0x7E00_F100	R/W	Configure AHB I/P/X/F bus	0x0400_0000
AHB_CON1	0x7E00_F104	R/W	Configure AHB M1/M0/T1/T0 bus	0x0000_0000
AHB_CON2	0x7E00_F108	R/W	Configure AHB R/S1/S0 bus	0x0000_0000
CLK_SRC2	0x7E00_F10C	R/W	Select Audio2 clock source	0x0000_0000
SDMA_SEL	0x7E00_F110	R/W	Select secure DMA input	0x0000_0000
RESERVED	0x7E00_F114	R/W	RESERVED	0x0000_0000
SYS_ID	0x7E00_F118	R	System ID for revision and pass	0x3641_0101
SYS_OTHERS	0x7E00_F11C	R/W	SYSCON others control register	0x0000_0000
MEM_SYS_CFG	0x7E00_F120	R/W	Configure memory subsystem	0x0000_0080
RESERVED	0x7E00_F124	R/W	RESERVED	0x0000_0000
QOS_OVERRIDE1	0x7E00_F128	R/W	Override DMC1 QOS	0x0000_0000
MEM_CFG_STAT	0x7E00_F12C	R	Memory subsystem setup status	0x0000_0000

RESERVED	0x7E00_F130	R/W	Should be 0x0	0x0000_0000
RESERVED	0x7E00_F130~ 0x7E00_F1FC	-	RESERVED	-
RESERVED	0x7E00_F200 ~ 0x7E00_F23C	R/W	Should be 0x0	0x0000_0000 ~ 0x0000_0000
RESERVED	0x7E00_F240~ 0x7E00_F800	-	RESERVED	-
PWR_CFG	0x7E00_F804	R/W	Configure power manager	0x0000_0001
EINT_MASK	0x7E00_F808	R/W	Configure EINT(external interrupt) mask	0x0000_0000
RESERVED	0x7E00_F80C	-	RESERVED	-
NORMAL_CFG	0x7E00_F810	R/W	Configure power manager at NORMAL mode	0xFFFF_FF00
STOP_CFG	0x7E00_F814	R/W	Configure power manager at STOP mode	0x2012_0100
SLEEP_CFG	0x7E00_F818	R/W	Configure power manager at SLEEP mode	0x0000_0000
STOP_MEM_CFG	0x7E00_F81C	R/W	Configure memory power at STOP mode	0x0000_007f
OSC_FREQ	0x7E00_F820	R/W	Oscillator frequency scale counter	0x0000_000F
OSC_STABLE	0x7E00_F824	R/W	Oscillator pad stable counter	0x0000_0001
PWR_STABLE	0x7E00_F828	R/W	Power stable counter	0x0000_0001
RESERVED	0x7E00_F82C	-	RESERVED	-
MTC_STABLE	0x7E00_F830	R/W	MTC stable counter	0xFFFF_FFFF
MISC_CON	0x7E00_F838	R/W	Bus control/SYNC667 control	0x0000_0000
RESERVED	0x7E00_F838~ 0x7E00_F8FC	-	RESERVED	-
OTHERS	0x7E00_F900	R/W	Others control register	0x0000_801E
RST_STAT	0x7E00_F904	R	Reset status register	0x0000_0001
WAKEUP_STAT	0x7E00_F908	R/W	Wakeup status register	0x0000_0000
BLK_PWR_STAT	0x7E00_F90C	R	Block power status register	0x0000_00FF
INFORM0	0x7E00_FA00	R/W	Information register0	0x0000_0000
INFORM1	0x7E00_FA04	R/W	Information register1	0x0000_0000
INFORM2	0x7E00_FA08	R/W	Information register2	0x0000_0000
INFORM3	0x7E00_FA0C	R/W	Information register3	0x0000_0000

SFRs consist of five parts. The SFRs, whose address are 0x7E00_F0XX, controls PLL and the clock generators. They control the output frequency of three PLLs, clock source selection, and clock divider ratio. The SFRs, whose address is 0x7E00_F1XX, control the bus system, the memory system. The SFRs, whose address is 0x7E00_F8XX, control the power management block. The SFRs, whose address is 0x7E00_F9XX, show the internal status. The information registers, whose address is 0x7E00_FA0X, save user information until the hardware reset signal (XnRESET) is asserted.

3.4.2 INDIVIDUAL REGISTER DESCRIPTIONS

3.4.2.1 PLL Control Registers

S3C6410 has three internal PLLs, which are APLL, MPLL, and EPLL. They are controlled by the following seven special registers.

REGISTER	ADDRSS	R/W	DESCRIPTION	RESET VALUE
APLL_LOCK	0x7E00_F000	R/W	Control PLL locking period for APLL	0x0000_FFFF
MPLL_LOCK	0x7E00_F004	R/W	Control PLL locking period for MPLL	0x0000_FFFF
EPLL_LOCK	0x7E00_F008	R/W	Control PLL locking period for EPLL	0x0000_FFFF
APLL_CON	0x7E00_F00C	R/W	Control PLL output frequency for APLL	0x0190_0302
MPLL_CON	0x7E00_F010	R/W	Control PLL output frequency for MPLL	0x0214_0603
EPLL_CON0	0x7E00_F014	R/W	Control PLL output frequency for EPLL	0x0020_0102
EPLL_CON1	0x7E00_F018	R/W	Control PLL output frequency for EPLL	0x0000_9111

A PLL requires locking period when 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, output will be masked '0'.

APLL_LOCK / MPLL_LOCK / EPLL_LOCK	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:16]	RESERVED	0x0000
PLL_LOCKTIME	[15:0]	Required period to generate a stable clock output	0xFFFF

PLL	Max. LockTime(us)	PLL_LOCK(FIN:12MHz)
APLL	300	0xE11
MPLL	300	0xE11
EPLL	300	0xE11

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, SDIV, and KDIV values. APLL_LOCK, MPLL_LOCK, and EPLL_LOCK fields denote the number of external clock. User can adjust this fields, which must be larger than maximum lock time (A/MPLL is 300us, EPLL is 300us).

APLL_CON / MPLL_CON	BIT	DESCRIPTION	RESET VALUE
ENABLE	[31]	PLL enable control (0: disable, 1: enable)	0
RESERVED	[30:26]	RESERVED	0x00
MDIV	[25:16]	PLL M divide value	0x190 / 0x214
RESERVED	[15:14]	RESERVED	0x0
PDIV	[13:8]	PLL P divide value	0x3 / 0x6
RESERVED	[7:3]	RESERVED	0x00
SDIV	[2:0]	PLL S divide value	0x2 / 0x3

The reset value of APLL_CON / MPLL_CON generate 400MHz and 133MHz output clock respectively, if the input clock frequency is 12MHz.

NOTE1:

The output frequency is calculated using 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 :

MDIV: $64 \leq MDIV \leq 1023$

PDIV: $1 \leq PDIV \leq 63$

SDIV: $0 \leq SDIV \leq 5$

F_{VCO} (=MDIV X F_{IN} / PDIV): $800MHz \leq F_{VCO} \leq 1600MHz$

F_{OUT}: $40MHz \leq F_{VCO} \leq 1600MHz$

F_{IN} : $10MHz \leq F_{IN} \leq 20MHz$

Don't set the value P and M to all zeros.

NOTE2:

Although there is the equation for choosing PLL value, we recommend only the values in the PLL value recommendation table. If you have to use another value, please contact us.

FIN (MHz)	Target FOUT (MHz)	MDIV	PDIV	SDIV
12	100	400	3	4
12	200	400	3	3
12	266	266	3	2
12	400	400	3	2
12	533	266	3	1
12	667	333	3	1

EPLL_CON0	BIT	DESCRIPTION	RESET VALUE
ENABLE	[31]	PLL enable control (0: disable, 1: enable)	0
RESERVED	[30:24]	RESERVED	0x00
MDIV	[23:16]	PLL M divide value	0x20
RESERVED	[15:14]	RESERVED	0x0
PDIV	[13:8]	PLL P divide value	0x1
RESERVED	[7:3]	RESERVED	0x00
SDIV	[2:0]	PLL S divide value	0x2

EPLL_CON1	BIT	DESCRIPTION	RESET VALUE
RESERVED	[30:18]	RESERVED	0x0
RESERVED	[17:16]	Should be 0x0	0x0
KDIV	[15:0]	PLL K divide value	0x9111

The reset value of EPLL_CON0 / EPLL_CON1 generate 97.70MHz output clock respectively, if the input clock frequency is 12MHz.

NOTE1:

The output frequency is calculated by the following equation:

$F_{OUT} = (MDIV + KDIV / 2^{16}) \times F_{IN} / (PDIV \times 2^{SDIV})$

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

MDIV: $16 \le MDIV \le 255$

PDIV: $1 \le PDIV \le 63$

KDIV: $0 \le KDIV \le 65535$

SDIV: $0 \le SDIV \le 4$

F_{VCO} (= $(MDIV + KDIV / 2^{16}) \times F_{IN} / PDIV$) : $300MHz \le F_{VCO} \le 600MHz$

F_{OUT} : $20MHz \le F_{OUT} \le 600MHz$

F_{IN} : $10MHz \le F_{IN} \le 20MHz$

Don't set the value P and M to all zeros.

NOTE2:

Although there is the equation for choosing PLL value, we recommend only the values in the PLL value recommendation table. If you have to use another value, please contact us.

FIN (MHz)	FOUT (MHz)	MDIV	PDIV	SDIV	KDIV
12	36	48	1	4	0
12	48	32	1	3	0
12	60	40	1	3	0
12	72	48	1	3	0
12	84	28	1	2	0
12	96	32	1	2	0
12	32.768	43	1	4	45264
12	45.158	30	1	3	6903
12	49.152	32	1	3	50332
12	67.738	45	1	3	10398
12	73.728	49	1	3	9961

During the operation of S3C6410 in NORMAL mode, if the user wants to change the frequency by writing the PMS value, the PLL lock time is automatically inserted. During the lock time, the clock is not supplied to the internal blocks in S3C6410. The timing diagram is as follow.

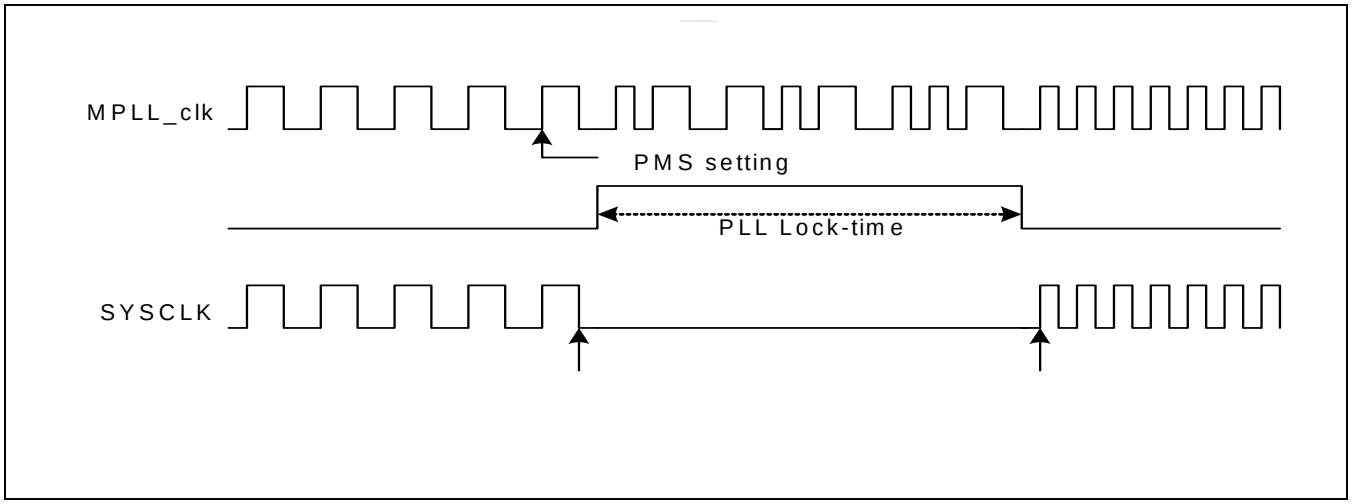


Figure 3-18. The case that changes Slow clock by setting PMS value

NOTE : Changing PMS value can cause a problem in LCD display. In the S3C6410, the LCD screen-refresh timing is dependent on the HCLK (HCLK clock is also dependent on the A(M)PLL clock output).

3.4.2.2 Clock source control register

S3C6410 has many clock sources, which include three PLL outputs, the external oscillator, the external clock, and other clock sources from GPIO configuration. CLK_SRC register controls the source clock of each clock divider.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CLK_SRC	0x7E00_F01C	R/W	Select clock source	0x0000_0000
CLK_SRC2	0x7E00_F10C	R/W	Select Audio2 clock source	0x0000_0000

CLK_SRC	BIT	DESCRIPTION	RESET VALUE
TV27_SEL	[31]	Control MUXTV27, which is the source clock of TV27MHz (0: 27MHz, 1: FIN_{EPLL})	0
DAC27_SEL	[30]	Control MUX_{DAC27}, which is the source clock of DAC27MHz (0:27MHz, 1: FIN_{EPLL})	0
SCALER_SEL	[29:28]	Control MUX_{SCALER}, which is the source clock of TVSCALER (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL})	0x0
LCD_SEL	[27:26]	Control MUX_{LCD}, which is the source clock of LCD (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL})	0x0
IRDA_SEL	[25:24]	Control MUX_{IRDA}, which is the source clock of IRDA (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL}, 11: 48MHz)	0x0
MMC2_SEL	[23:22]	Control MUX_{MMC2}, which is the source clock of MMC2 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL} , 11: 27MHz)	0x0
MMC1_SEL	[21:20]	Control MUX_{MMC1}, which is the source clock of MMC1 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL} , 11: 27MHz)	0x0
MMC0_SEL	[19:18]	Control MUX_{MMC0}, which is the source clock of MMC0 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL} , 11: 27MHz)	0x0
SPI1_SEL	[17:16]	Control MUX_{SPI1}, which is the source clock of SPI1 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL} , 11: 27MHz)	0x0
SPI0_SEL	[15:14]	Control MUX_{SPI0}, which is the source clock of SPI0 (00:$MOUT_{EPLL}$, 01: $DOUT_{MPLL}$, 10: FIN_{EPLL} , 11: 27MHz)	0x0
UART_SEL	[13]	Control MUX_{UART0}, which is the source clock of UART (0:$MOUT_{EPLL}$, 1: $DOUT_{MPLL}$)	0
AUDIO1_SEL	[12:10]	Control MUX_{AUDIO1}, which is the source clock of IIS1, PCM1, and AC97 1 (000:$MOUT_{EPLL}$,0 01: $DOUT_{MPLL}$, 010:FIN_{EPLL}, 011: IISCDCLK1, 100: PCMCDCLK)	0x0
AUDIO0_SEL	[9:7]	Control MUX_{AUDIO0}, which is the source clock of IIS0, PCM0, and AC97 0 (000:$MOUT_{EPLL}$, 001: $DOUT_{MPLL}$, 010:FIN_{EPLL}, 011: IISCDCLK0, 10x: PCMCDCLK)	0x0
UHOST_SEL	[6:5]	Control MUX_{UHOST}, which is the source clock of USB Host	0x0

		(00:48MHz, 01:$MOUT_{EPLL}$, 10: $DOUT_{MPLL}$, 11:FIN_{EPLL})	
MFCCLK_SEL	[4]	Control MUXMFC, which is the source clock of MFC	0
RESERVED	[3]	RESERVED	0
EPLL_SEL	[2]	Control MUX_{EPLL} (0:FIN_{EPLL}, 1:$FOUT_{EPLL}$)	0
MPLL_SEL	[1]	Control MUX_{MPLL} (0:FIN_{MPLL}, 1:$FOUT_{MPLL}$)	0
APLL_SEL	[0]	Control MUX_{APLL} (0:FIN_{APLL}, 1:$FOUT_{APLL}$)	0

CLK_SRC2	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:3]	RESERVED	0x0
AUDIO2_SEL	[2:0]	Control MUX_{AUDIO2}, which is the source clock of AUDIO2 (000:$MOUT_{EPLL}$, 001: $DOUT_{MPLL}$, 010:FIN_{EPLL}, 011: I2SMultiCDCLK, 10x: PCMCDCLK[1])	0x0

3.4.2.3 Clock divider control register

S3C6410 has several clock dividers to support various operating clock frequency. The clock divider ratio can be controlled by CLK_DIV0, CLK_DIV1, and CLK_DIV2.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CLK_DIV0	0x7E00_F020	R/W	Set clock divider ratio	0x0105_1000
CLK_DIV1	0x7E00_F024	R/W	Set clock divider ratio	0x0000_0000
CLK_DIV2	0x7E00_F028	R/W	Set clock divider ratio	0x0000_0000

CLK_DIV0 mainly controls the system clocks and special clocks of multimedia IPs. The output frequencies of APLL and MPLL are divided by ARM_RATIO and MPLL_RATIO. HCLKX2 clock is the base clock of other operating system clocks and divided by HCLKX2_RATIO. There is operating frequency limitation. The maximum operating frequency of HCLKX2, HCLK, and PCLK are 266MHz, 133MHz, and 66MHz, respectively. NAND, SECUR, JPEG operating clock cannot exceed 66MHz. MFC and CAM operating clock cannot exceed 133MHz. This operating clock condition must be met through CLK_DIV0 configuration.

User software must be responsible for the clock divider controlled by CLK_DIV0. Since the output frequency will be varying during the clock ratio-changing period as shown in

CLK_DIV0	BIT	DESCRIPTION	RESET VALUE
MFC_RATIO	[31:28]	MFC clock divider ratio *CLKMFC = $CLKMFC_{IN}$ / (MFC_RATIO + 1)*	0x0
JPEG_RATIO	[27:24]	JPEG clock divider ratio, which must be odd value. In other words, S3C6410 supports only even divider ratio. *CLKJPEG = HCLKX2 / (JPEG_RATIO + 1)*	0x1
CAM_RATIO	[23:20]	CAM clock divider ratio *CLKCAM = HCLKX2 / (CAM_RATIO + 1)*	0x0
SECUR_RATIO	[19:18]	Security clock divider ratio, which must be 0x1 or 0x3. *CLKSECUR = HCLKX2 / (SECUR_RATIO + 1)*	0x1
RESERVED	[17:16]	RESERVED	0x1
PCLK_RATIO	[15:12]	PCLK clock divider ratio, which must be odd value. In other words, S3C6410 supports only even divider ratio. *PCLK = HCLKX2 / (PCLK_RATIO + 1)*	0x1
HCLKX2_RATIO	[11:9]	HCLKX2 clock divider ratio *HCLKX2 = $HCLKX2_{IN}$ / (HCLKX2_RATIO + 1)*	0x0
HCLK_RATIO	[8]	HCLK clock divider ratio *HCLK = HCLKX2 / (HCLK_RATIO + 1)*	0
RESERVED	[7:5]	RESERVED	0x0
MPLL_RATIO	[4]	DIV_{MPLL} clock divider ratio *$DOUT_{MPLL}$ = $MOUT_{MPLL}$ / (MPLL_RATIO + 1)*	0
ARM_RATIO	[3:0]	DIV_{ARM} clock divider ratio *ARMCLK = $DOUT_{APLL}$ / (ARM_RATIO + 1)*	0x0

CLK_DIV1 controls MMC, LCD, TV scaler, and UHOST clocks.

CLK_DIV1	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:28]	RESERVED	0x0
FIMC_RATIO	[27:24]	FIMC clock divider ratio *CLKFIMC = HCLK / (FIMC_RATIO + 1)*	0x0
UHOST_RATIO	[23:20]	USB host clock divider ratio *CLKUHOST = $CLKUHOST_{IN}$ / (UHOST_RATIO + 1)*	0x0
SCALER_RATIO	[19:16]	TV Scaler clock divider ratio *CLKSCALER = $CLKSCALER_{IN}$ / (SCALER_RATIO + 1)*	0x0
LCD_RATIO	[15:12]	LCD clock divider ratio *CLKLCD = $CLKLCD_{IN}$ / (LCD_RATIO + 1)*	0x0
MMC2_RATIO	[11:8]	MMC2 clock divider ratio *CLKMMC2 = $CLKMMC2_{IN}$ / (MMC2_RATIO + 1)*	0x0
MMC1_RATIO	[7:4]	MMC1 clock divider ratio *CLKMMC1 = $CLKMMC1_{IN}$ / (MMC1_RATIO + 1)*	0x0
MMC0_RATIO	[3:0]	MMC0 clock divider ratio *CLKMMC0 = $CLKMMC0_{IN}$ / (MMC0_RATIO + 1)*	0x0

CLK_DIV2 controls SPI, AUDIO, UART, and IrDA clocks.

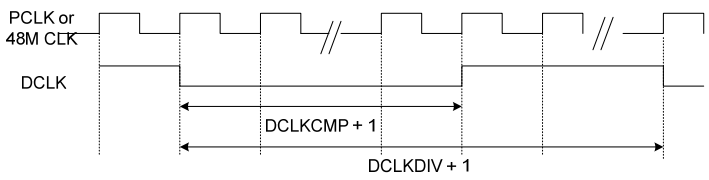
CLK_DIV2	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:28]	RESERVED	0x0
AUDIO2_RATIO	[27:24]	AUDIO2 clock divider ratio *CLKAUDIO2 = $CLKAUDIO2_{IN}$ / (AUDIO2_RATIO + 1)*	0x0
IRDA_RATIO	[23:20]	IRDA clock divider ratio *CLKIRDA = $CLKIRDA_{IN}$ / (IRDA_RATIO + 1)*	0x0
UART_RATIO	[19:16]	UART clock divider ratio *CLKUART = $CLKUART_{IN}$ / (UART_RATIO + 1)*	0x0
AUDIO1_RATIO	[15:12]	AUDIO1 clock divider ratio *CLKAUDIO1 = $CLKAUDIO1_{IN}$ / (AUDIO1_RATIO + 1)*	0x0
AUDIO0_RATIO	[11:8]	AUDIO0 clock divider ratio *CLKAUDIO0 = $CLKAUDIO0_{IN}$ / (AUDIO0_RATIO + 1)*	0x0
SPI1_RATIO	[7:4]	SPI1 clock divider ratio *CLKSPI1 = $CLKSPI1_{IN}$ / (SPI1_RATIO + 1)*	0x0
SPI0_RATIO	[3:0]	SPI0 clock divider ratio *CLKSPI0 = $CLKSPI0_{IN}$ / (SPI0_RATIO + 1)*	0x0

3.4.2.4 Clock output configuration register

Internal clocks can be monitored through GPIO port, which is GPIO port-F. CLK_OUT register selects an internal clock among PLL output, HCLK, 48MHz, 27MHz, RTC, and TICK. It also divides the selected clock.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
CLK_OUT	0x7E00_F02C	R/W	Select clock output	0x0000_0000

CLK_OUT	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:20]	RESERVED	0x000
DIVVAL	[19:16]	Divide ratio (Divide ratio = DIVVAL + 1	0x0
CLKSEL	[15:12]	0000 = $FOUT_{APLL}$/4 0001 = $FOUT_{EPLL}$ 0010 = HCLK 0011 = CLK48M 0100 = CLK27M 0101 = RTC 0110 =TICK 0111 = DOUT 1000 = FIN (EXTCLK or OSCCLK) 1001 = FrefAPLL 1010 = FfbAPLL/2 1011 = FrefMPLL 1100 = FfbMPLL/2 1101 = FrefEPLL 1110 = FfbEPLL/2	0x0
DCLKCMP	[11:8]	This field changes the clock duty of DCLK. Thus, it must be smaller than DCLKDIV. It is valid only when CLKSEL is 0x7 If the DCLKCMP 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:disable, 1:enable)	0



3.4.2.5 Clock gating control register

S3C6410 can disable the clock operation of each IP when it does not require running. The following three registers control clock disable/enable operation.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
HCLK_GATE	0x7E00_F030	R/W	HCLK clock gating control	0xFFFF_FFFF
PCLK_GATE	0x7E00_F034	R/W	PCLK clock gating control	0xFFFF_FFFF
SCLK_GATE	0x7E00_F038	R/W	Special clock gating control	0xFFFF_FFFF
MEM0_CLK_GATE	0x7E00_F03C	R/W	MEM0 clock gating control	0xFFFF_FFFF

HCLK_GATE controls HCLK of all IPs. If a field has '1', then HCLK is supplied. Otherwise, HCLK is masked. When S3C6410x changes to a power down mode, system controller checks the status of some block, IROM, MEM0, MEM1, and MFC block.

HCLK_GATE	BIT	DESCRIPTION	RESET VALUE
HCLK_3DSE	[31]	Gating HCLK for 3D (0: mask, 1: pass)	1
RESERVED	[30]	RESERVED	1
HCLK_UHOST	[29]	Gating HCLK for UHOST (0: mask, 1: pass)	1
HCLK_SECUR	[28]	Gating HCLK for security sub-system (0: mask, 1: pass)	1
HCLK_SDMA1	[27]	Gating HCLK for SDMA1 (0: mask, 1: pass)	1
HCLK_SDMA0	[26]	Gating HCLK for SDMA0 (0: mask, 1: pass)	1
HCLK_IROM	[25]	Gating HCLK for IROM (0: mask, 1: pass)	1
HCLK_DDR1	[24]	Gating HCLK for DDR1 (0: mask, 1: pass)	1
RESERVED	[23]	RESERVED	1
HCLK_MEM1	[22]	Gating HCLK for DMC1 (0: mask, 1: pass)	1
HCLK_MEM0	[21]	Gating HCLK for SROM, OneNAND, NFCON, CFCON (0: mask, 1: pass)	1
HCLK_USB	[20]	Gating HCLK for USB OTG (0: mask, 1: pass)	1
HCLK_HSMMC2	[19]	Gating HCLK for HSMMC2 (0: mask, 1: pass)	1
HCLK_HSMMC1	[18]	Gating HCLK for HSMMC1 (0: mask, 1: pass)	1
HCLK_HSMMC0	[17]	Gating HCLK for HSMMC0 (0: mask, 1: pass)	1
HCLK_MDP	[16]	Gating HCLK for MDP (0: mask, 1: pass)	1
HCLK_DHOST	[15]	Gating HCLK for direct HOST interface (0: mask, 1: pass)	1
HCLK_IHOST	[14]	Gating HCLK for indirect HOST interface (0: mask, 1: pass)	1
HCLK_DMA1	[13]	Gating HCLK for DMA1 (0: mask, 1: pass)	1
HCLK_DMA0	[12]	Gating HCLK for DMA0 (0: mask, 1: pass)	1
HCLK_JPEG	[11]	Gating HCLK for JPEG (0: mask, 1: pass)	1
HCLK_CAMIF	[10]	Gating HCLK for camera interface (0: mask, 1: pass)	1
HCLK_SCALER	[9]	Gating HCLK for scaler (0: mask, 1: pass)	1
HCLK_2D	[8]	Gating HCLK for 2D (0: mask, 1: pass)	1
HCLK_TV	[7]	Gating HCLK for TV encoder (0: mask, 1: pass)	1

RESERVED	[6]	RESERVED	1
HCLK_POST0	[5]	Gating HCLK for POST0 (0: mask, 1: pass)	1
HCLK_ROT	[4]	Gating HCLK for rotator (0: mask, 1: pass)	1
HCLK_LCD	[3]	Gating HCLK for LCD controller (0: mask, 1: pass)	1
HCLK_TZIC	[2]	Gating HCLK for trust interrupt controller (0: mask, 1: pass)	1
HCLK_INTC	[1]	Gating HCLK for vectored interrupt controller (0: mask, 1: pass)	1
HCLK_MFC	[0]	Gating HCLK for MFC (0: mask, 1: pass)	1

PCLK_GATE controls PCLK of all IPs.

PCLK_GATE	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:28]	RESERVED	0x1F
PCLK_IIC1	[27]	Gating PCLK for IIC1 (0: mask, 1: pass)	1
PCLK_IIS2	[26]	Gating PCLK for IIS2 (0: mask, 1: pass)	1
RESERVED	[25]	RESERVED	1
PCLK_SKEY	[24]	Gating PCLK for security key (0: mask, 1: pass)	1
PCLK_CHIPID	[23]	Gating PCLK for chip ID (0: mask, 1: pass)	1
PCLK_SPI1	[22]	Gating PCLK for SPI1 (0: mask, 1: pass)	1
PCLK_SPI0	[21]	Gating PCLK for SPI0 (0: mask, 1: pass)	1
PCLK_HSIRX	[20]	Gating PCLK for HSI receiver (0: mask, 1: pass)	1
PCLK_HSITX	[19]	Gating PCLK for HSI transmitter (0: mask, 1: pass)	1
PCLK_GPIO	[18]	Gating PCLK for GPIO (0: mask, 1: pass)	1
PCLK_IIC0	[17]	Gating PCLK for IIC0 (0: mask, 1: pass)	1
PCLK_IIS1	[16]	Gating PCLK for IIS1 (0: mask, 1: pass)	1
PCLK_IIS0	[15]	Gating PCLK for IIS0 (0: mask, 1: pass)	1
PCLK_AC97	[14]	Gating PCLK for AC97 (0: mask, 1: pass)	1
PCLK_TZPC	[13]	Gating PCLK for TZPC (0: mask, 1: pass)	1
PCLK_TSADC	[12]	Gating PCLK for touch screen ADC (0: mask, 1: pass)	1
PCLK_KEYPAD	[11]	Gating PCLK for Key PAD (0: mask, 1: pass)	1
PCLK_IRDA	[10]	Gating PCLK for IRDA (0: mask, 1: pass)	1
PCLK_PCM1	[9]	Gating PCLK for PCM1 (0: mask, 1: pass)	1
PCLK_PCM0	[8]	Gating PCLK for PCM0 (0: mask, 1: pass)	1
PCLK_PWM	[7]	Gating PCLK for PWM (0: mask, 1: pass)	1
PCLK_RTC	[6]	Gating PCLK for RTC (0: mask, 1: pass)	1
PCLK_WDT	[5]	Gating PCLK for watch dog timer (0: mask, 1: pass)	1
PCLK_UART3	[4]	Gating PCLK for UART3 (0: mask, 1: pass)	1
PCLK_UART2	[3]	Gating PCLK for UART2 (0: mask, 1: pass)	1
PCLK_UART1	[2]	Gating PCLK for UART1 (0: mask, 1: pass)	1

PCLK_UART0	[1]	Gating PCLK for UART0 (0: mask, 1: pass)	1
PCLK_MFC	[0]	Gating PCLK for MFC (0: mask, 1: pass)	1

SCLK_GATE controls special clocks of IPs.

SCLK_GATE	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31]	RESERVED	1
SCLK_UHOST	[30]	Gating special clock for USB-HOST (0: mask, 1: pass)	1
SCLK_MMC2_48	[29]	Gating special clock for MMC2 (0: mask, 1: pass)	1
SCLK_MMC1_48	[28]	Gating special clock for MMC1 (0: mask, 1: pass)	1
SCLK_MMC0_48	[27]	Gating special clock for MMC0 (0: mask, 1: pass)	1
SCLK_MMC2	[26]	Gating special clock for MMC2 (0: mask, 1: pass)	1
SCLK_MMC1	[25]	Gating special clock for MMC1 (0: mask, 1: pass)	1
SCLK_MMC0	[24]	Gating special clock for MMC0 (0: mask, 1: pass)	1
SCLK_SPI1_48	[23]	Gating special clock for SPI (0: mask, 1: pass)	1
SCLK_SPI0_48	[22]	Gating special clock for SPI (0: mask, 1: pass)	1
SCLK_SPI1	[21]	Gating special clock for SPI (0: mask, 1: pass)	1
SCLK_SPI0	[20]	Gating special clock for SPI (0: mask, 1: pass)	1
SCLK_DAC27	[19]	Gating special clock for DAC (0: mask, 1: pass)	1
SCLK_TV27	[18]	Gating special clock for TV encoder (0: mask, 1: pass)	1
SCLK_SCALER27	[17]	Gating special clock for scaler27 (0: mask, 1: pass)	1
SCLK_SCALER	[16]	Gating special clock for scaler (0: mask, 1: pass)	1
SCLK_LCD27	[15]	Gating special clock for LCD controller (0: mask, 1: pass)	1
SCLK_LCD	[14]	Gating special clock for LCD controller (0: mask, 1: pass)	1
SCLK_FIMC	[13]	Gating special clock for camera & LCD (0: mask, 1: pass)	1
SCLK_POST0_27	[12]	Gating special clock for POST0 (0: mask, 1: pass)	1
SCLK_AUDIO2	[11]	Gating special clock for IIS2 (0: mask, 1: pass)	1
SCLK_POST0	[10]	Gating special clock for POST0 (0: mask, 1: pass)	1
SCLK_AUDIO1	[9]	Gating special clock for PCM1, IIS1 (0: mask, 1: pass)	1
SCLK_AUDIO0	[8]	Gating special clock for PCM0, IIS0 (0: mask, 1: pass)	1
SCLK_SECUR	[7]	Gating special clock for security block (0: mask, 1: pass)	1
SCLK_IRDA	[6]	Gating special clock for IRDA (0: mask, 1: pass)	1
SCLK_UART	[5]	Gating special clock for UART0~3 (0: mask, 1: pass)	1
RESERVED	[4]	RESERVED	1
SCLK_MFC	[3]	Gating special clock for MFC (0: mask, 1: pass)	1
SCLK_CAM	[2]	Gating special clock for camera interface (0: mask, 1: pass)	1
SCLK_JPEG	[1]	Gating special clock for JPEG (0: mask, 1: pass)	1
RESERVED	[0]	RESERVED	1

MEM0_CLK_GATE	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:6]	RESERVED	0x3ff_ffff
HCLK_CFCON	[5]	Gating special clock for CFCON (0: mask, 1: pass)	1
HCLK_OneNAND1	[4]	Gating special clock for OneNAND1 (0: mask, 1: pass)	1
HCLK_OneNAND0	[3]	Gating special clock for OneNAND0 (0: mask, 1: pass)	1
HCLK_NFCON	[2]	Gating special clock for NFCON (0: mask, 1: pass)	1
HCLK_SROM	[1]	Gating special clock for SROM (0: mask, 1: pass)	1
RESERVED	[0]	RESERVED	1

Note) Set HCLK_NFCON to '0' in case of 6410X PoP A type. Set HCLK_OneNAND1 and HCLK_OneNAND0 to '0' in case of 6410X PoP D type.

3.4.2.6 AHB bus control register

Most multimedia IPs are connected through AHB 2.0 bus system. The bus priority can be controlled using the following three registers.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
AHB_CON0	0x7E00_F100	R/W	Configure AHB I/P/X/F bus	0x0400_0000
AHB_CON1	0x7E00_F104	R/W	Configure AHB M1/M0/T1/T0 bus	0x0000_0000
AHB_CON2	0x7E00_F108	R/W	Configure AHB R/S1/S0 bus	0x0000_0000

The arbitration method for each AHB bus can be configured using “PRIOR_TYPE_*name*” field as follows

- 00: Fixed priority type. Its priority is defined using “FIX_PRIOR_*name*” field.
- 01: Last grant lowest. Last granted master has lowest arbitration value.
- 10: Rotation.

When the priority type filed selects the fixed priority(00), the priority order is defined as follows,

FIX_PRIOR_*name*	Highest
000	0 – 1 – 2 – 3 – 4 – 5 – 6 – 7
001	1 – 2 – 3 – 4 – 5 – 6 – 0 – 7
010	2 – 3 – 4 – 5 – 6 – 0 – 1 – 7
011	3 – 4 – 5 – 6 – 0 – 1 – 2 – 7
100	4 – 5 – 6 – 0 – 1 – 2 – 3 – 7
101	5 – 6 – 0 – 1 – 2 – 3 – 4 – 7
110	6 – 0 – 1 – 2 – 3 – 4 – 5 – 7

DISABLE_HLOCK field controls the locked operation of AHB bus system. When a master request locked operation, a bus arbiter gives a grant when the field is ‘0’. If the field is ‘1’, the master cannot get a locked grant.

AHB_CON0 controls AHB-F, AHB-X, AHB-P, AHB-I bus systems.

AHB_CON0	BIT	DESCRIPTION	RESET VALUE
DISABLE_HLOCK_I	[31]	Control HLOCK for I-block (0: disable, 1:enable)	0
RESERVED	[30]	RESERVED	0
PRIOR_TYPE_I	[29:28]	Arbitration type for AHB-I	0
RESERVED	[27]	RESERVED	0
FIX_PRIOR_I	[26:24]	Fixed priority order for AHB-I	0x4
DISABLE_HLOCK_P	[23]	Control HLOCK for P-block (0: disable, 1:enable)	0
RESERVED	[22]	RESERVED	0
PRIOR_TYPE_P	[21:20]	Arbitration type for AHB-P	0
RESERVED	[19]	RESERVED	0
FIX_PRIOR_P	[18:16]	Fixed priority order for AHB-P	0x0
DISABLE_HLOCK_X	[15]	Control HLOCK for X-block (0: disable, 1:enable)	0
RESERVED	[14]	RESERVED	0
PRIOR_TYPE_X	[13:12]	Arbitration type for AHB-X	0
RESERVED	[11]	RESERVED	0
FIX_PRIOR_X	[10:8]	Fixed priority order for AHB-X	0x0
DISABLE_HLOCK_F	[7]	Control HLOCK for F-block (0: disable, 1:enable)	0
RESERVED	[6]	RESERVED	0
PRIOR_TYPE_F	[5:4]	Arbitration type for AHB-F	0
RESERVED	[3]	RESERVED	0
FIX_PRIOR_F	[2:0]	Fixed priority order for AHB-F	0x0

The arbitration numbers of the above bus system are as follows:

Arbitration Number	AHB-F	AHB-X	AHB-P	AHB-I
0	Display controller 0	USB OTG	TV Scaler	Camera I/F
1	Display controller 1	HS-MMC0		Camera I/F
2	Display controller 2	HS-MMC1	-	Camera I/F
3	Display controller 3	HS-MMC2	-	Camera I/F
4	Display controller 4	-	-	JPEG
5	POST	-	-	-
6	Rotator	-	-	-
7	Default bus master			

AHB_CON1 controls AHB-T0, AHB-T1, AHB-M0, AHB-M1 bus systems.

AHB_CON1	BIT	DESCRIPTION	RESET VALUE
DISABLE_HLOCK_M	[31]	Control HLOCK for M-block (0: disable, 1:enable)	0
RESERVED	[30]	RESERVED	0
PRIOR_TYPE_M1	[29:28]	Arbitration type for AHB-M1	0
RESERVED	[27]	RESERVED	0
FIX_PRIOR_M1	[26:24]	Fixed priority order for AHB-M1	0x0
RESERVED	[23:22]	RESERVED	0x0
PRIOR_TYPE_M0	[21:20]	Arbitration type for AHB-M0	0
RESERVED	[19]	RESERVED	0
FIX_PRIOR_M0	[18:16]	Fixed priority order for AHB-M0	0x0
DISABLE_HLOCK_T	[15]	Control HLOCK for T-block (0: disable, 1:enable)	0
RESERVED	[14]	RESERVED	0
PRIOR_TYPE_T1	[13:12]	Arbitration type for AHB-T1	0
RESERVED	[11]	RESERVED	0
FIX_PRIOR_T1	[10:8]	Fixed priority order for AHB-T1	0x0
RESERVED	[7:6]	RESERVED	0x0
PRIOR_TYPE_T0	[5:4]	Arbitration type for AHB-T0	0
RESERVED	[3]	RESERVED	0
FIX_PRIOR_T0	[2:0]	Fixed priority order for AHB-T0	0x0

The arbitration numbers of the above bus system are as follows:

Arbitration Number	AHB-T0	AHB-T1	AHB-M0	AHB-M1
0	HOST I/F	HOST I/F	DMA0	DMA0
1	USB Host	-	DMA1	DMA1
2	MDP I/F	-	-	-
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7	Default bus master			

AHB_CON2 controls AHB-S0, AHB-S1, AHB-R bus systems.

AHB_CON2	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:24]	RESERVED	0x00
DISABLE_HLOCK_R	[23]	Control HLOCK for R-block (0: disable, 1: enable)	0
RESERVED	[22:16]	RESERVED	0
DISABLE_HLOCK_S	[15]	Control HLOCK for S-block (0: disable, 1:enable)	0
RESERVED	[14]	RESERVED	0
PRIOR_TYPE_S1	[13:12]	Arbitration type for AHB-S1	0
RESERVED	[11]	RESERVED	0
FIX_PRIOR_S1	[10:8]	Fixed priority order for AHB-S1	0x0
RESERVED	[7:6]	RESERVED	0x0
PRIOR_TYPE_S0	[5:4]	Arbitration type for AHB-S0	0
RESERVED	[3]	RESERVED	0
FIX_PRIOR_S0	[2:0]	Fixed priority order for AHB-S0	0x0

The arbitration numbers of the above bus system are as follows:

Arbitration Number	AHB-S0	AHB-S1	AHB-R
0	Secure DMA0	Secure DMA0	CFCON
1	Secure DMA1	Secure DMA1	-
2	-	-	-
3	-	-	-
4	-	-	-
5	-	-	-
6	-	-	-
7	Default bus master		

3.4.2.7 Secure DMA control register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
SDMA_SEL	0x7E00_F110	R/W	Secure DMA input selection	0x0000_0000

SDMA_SEL	BIT	DESCRIPTION	RESET VALUE
SECURITY_TX	[31]	DMA selection for security Tx (always selects SDMA1 regardless of SECURITY_TX field)	0
SECURITY_RX	[30]	DMA selection for security Rx (always selects SDMA1 regardless of SECURITY_TX field)	0
RESERVED	[29]	RESERVED	0
RESERVED	[28]	RESERVED	0
EXTERNAL	[27]	DMA selection for external (0: SDMA1, 1: DMA1)	0
IRDA	[26]	DMA selection for IrDA (0: SDMA1, 1: DMA1)	0
PWM	[25]	DMA selection for IrDA (0: SDMA1, 1: DMA1)	0
AC_MICIN	[24]	DMA selection for IrDA (0: SDMA1, 1: DMA1)	0
AC_PCMIN	[23]	DMA selection for PCM input (0: SDMA1, 1: DMA1)	0
AC_PCMOUT	[22]	DMA selection for PCM output (0: SDMA1, 1: DMA1)	0
SPI1_RX	[21]	DMA selection for SPI1 Rx (0: SDMA1, 1: DMA1)	0
SPI1_TX	[20]	DMA selection for SPI1 Tx (0: SDMA1, 1: DMA1)	0
I2S1_RX	[19]	DMA selection for I2S1 Rx (0: SDMA1, 1: DMA1)	0
I2S1_TX	[18]	DMA selection for I2S1 Tx (0: SDMA1, 1: DMA1)	0
PCM1_RX	[17]	DMA selection for PCM1 Rx (0: SDMA1, 1: DMA1)	0
PCM1_TX	[16]	DMA selection for PCM1 Tx (0: SDMA1, 1: DMA1)	0
HSI_RX	[15]	DMA selection for HSI Rx (0: SDMA0, 1: DMA0)	0
HSI_TX	[14]	DMA selection for HSI Tx (0: SDMA0, 1: DMA0)	0
SPI0_RX	[13]	DMA selection for SPI0 Rx (0: SDMA0, 1: DMA0)	0
SPI0_TX	[12]	DMA selection for SPI0 Tx (0: SDMA0, 1: DMA0)	0
I2S0_RX	[11]	DMA selection for I2S0 Rx (0: SDMA0, 1: DMA0)	0
I2S0_TX	[10]	DMA selection for I2S0 Tx (0: SDMA0, 1: DMA0)	0
PCM0_RX	[9]	DMA selection for PCM0 Rx (0: SDMA0, 1: DMA0)	0
PCM0_TX	[8]	DMA selection for PCM0 Tx (0: SDMA0, 1: DMA0)	0
UART3[1]	[7]	DMA selection for UART3 (0: SDMA0, 1: DMA0)	0
UART3[0]	[6]	DMA selection for UART3 (0: SDMA0, 1: DMA0)	0
UART2[1]	[5]	DMA selection for UART2 (0: SDMA0, 1: DMA0)	0
UART2[0]	[4]	DMA selection for UART2 (0: SDMA0, 1: DMA0)	0
UART1[1]	[3]	DMA selection for UART1 (0: SDMA0, 1: DMA0)	0
UART1[0]	[2]	DMA selection for UART1 (0: SDMA0, 1: DMA0)	0
UART0[1]	[1]	DMA selection for UART0 (0: SDMA0, 1: DMA0)	0
UART0[0]	[0]	DMA selection for UART0 (0: SDMA0, 1: DMA0)	0

3.4.2.8 System ID register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
SYS_ID	0x7E00_F118	R	System ID for revision and pass	0x3641_0101

SYS_ID	BIT	DESCRIPTION	RESET VALUE
ID	[31:8]	ID	0x3641_01
Revision	[7:4]	Specification revision	0x0
Pass	[3:0]	Layout revision	0x1

3.4.2.9 System Others register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
SYS_OTHERS	0x7E00_F11C	R/W	System ID for revision and pass	0x0

SYS_OTHERS	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:6]	RESERVED	0
PMU_IRQ_ENABLE	[5]	Enable ARM nPMUIRQ (Performance Monitor Unit IRQ) interrupt (0: disable, 1: enable)	0
RESERVED	[4]	RESERVED	0
MODEM_RX1_SDMA_SEL	[3]	DMA selection for MODEM RX1 (0: SDMA0, 1: DMA0)	0
MODEM_RX0_SDMA_SEL	[2]	DMA selection for MODEM RX0 (0: SDMA0, 1: DMA0)	0
MODEM_TX1_SDMA_SEL	[1]	DMA selection for MODEM TX1 (0: SDMA0, 1: DMA0)	0
MODEM_TX0_SDMA_SEL	[0]	DMA selection for MODEM TX0 (0: SDMA0, 1: DMA0)	0

3.4.2.10 Memory controller status register

Memory controller status registers must be initialized by software except MEM_CFG_STAT.

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
MEM_SYS_CFG	0x7E00_F120	R/W	Memory Subsystem configuration register	0x0000_0080
RESERVED	0x7E00_F124	R/W	RESERVED	0x0000_0000
QOS_OVERRIDE1	0x7E00_F128	R/W	DMC1 QOS Override register	0x0000_0000
MEM_CFG_STAT	0x7E00_F12C	R	Memory Subsystem setup status register	0x0000_0000

MEM_SYS_CFG	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:15]	RESERVED	0x0000_0
INDEP_CF	[14]	Use CF interface independently. 0 = Use memory port 0 shared by EBI. 1 = Use independent CF interface.	0
RESERVED	[13]	Should be '0'	0
BUS_WIDTH	[12]	Select initial state of SROMC CS0 memory bus width. 0 = 8-bit data width. 1 = 16-bit data width. If NOR booting (OM[4:1] = 0101) is selected, this setting is ignored and 16-bit data width is selected. Even this bit is set to 0 or 1, this selects only reset value of DataWidth0 of SROM_BW SFR in SROMC. Bus width of CS0 for SROMC follows DataWidth0 setting.	0
EBI_PRI	[11]	Set EBI priority scheme. 0 = Fixed priority scheme. 1 = Circular priority scheme.	0
EBI_FIX_PRI	[10:8]	Set EBI fixed priority setting. Highest <-> Lowest 0,6,7: SROMC - OneNANDC CS0 - OneNANDC CS1 – NFCON – CFCON 1: OneNANDC CS0 – OneNANDC CS1 – SROMC – NFCON – CFCON 2: OneNANDC CS1 – NFCON – SROMC - OneNANDC CS0 – CFCON 3: NFCON – SROMC - OneNANDC CS0 - OneNANDC CS1 – CFCON 4: CFCON – SROMC - OneNANDC CS0 - OneNANDC CS1 – NFCON 5: SROMC - OneNANDC CS0 - OneNANDC CS1 – NFCON – CFCON	000
ADDR_EXPAND (CFG_SROM_ADDR_EXPAND_To	[7]	Set usage of Xm1DATA[31:16] pins. 0 = Xm1DATA[26:16] pins are used for DMC1 upper halfword	1

<table>
<tr><td>_DRAM1</td><td></td><td>data field, data[26:16].
1 = Xm1DATA[26:16] pins are used for SROMC upper 11-bit address field, address[26:16].</td><td></td></tr>
<tr><td>RESERVED</td><td>[6]</td><td>RESERVED</td><td>0</td></tr>
<tr><td>MP0_CS_CFG</td><td>[5:0]</td><td>Set static memory chip selection multiplexing of memory port 0.
Setting for MP0_CS_CFG[0] and MP0_CS_CFG[2] are ignored. Distinguishing OneNANDC and NFCON is done by XSELNAND pin value instead of MP0_CS_CFG[0] and MP0_CS_CFG[2]. When XSELNAND is 0, OneNANDC is selected. When XSELNAND is 1, NFCON is selected.
When OneNAND booting (OM[4:1] = 0110) is selected, the setting values of MP0_CS_CFG[1] and MP0_CS_CFG[3] are ignored and Xm0CSn[2] and Xm0CSn[3] are used as OneNANDC CS0 and OneNANDC CS1. In this case, XSELNAND should be set to 0.
<table>
<tr><td></td><td colspan="6">MP0_CS_CFG</td><td></td></tr>
<tr><td></td><td>[5]</td><td>[4]</td><td>[3]</td><td>[2]</td><td>[1]</td><td>[0]</td><td></td></tr>
<tr><td>Xm0CSn[0]</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>SROMC CS0</td></tr>
<tr><td>Xm0CSn[1]</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>SROMC CS1</td></tr>
<tr><td rowspan="3">Xm0CSn[2]</td><td>-</td><td>-</td><td>-</td><td>-</td><td>1</td><td>-</td><td>SROMC CS2</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>-</td><td>OneNANDC CS0</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>-</td><td>NFCON CS0</td></tr>
<tr><td rowspan="3">Xm0CSn[3]</td><td>-</td><td>-</td><td>1</td><td>-</td><td>-</td><td>-</td><td>SROMC CS3</td></tr>
<tr><td>-</td><td>-</td><td>0</td><td>-</td><td>-</td><td>-</td><td>OneNANDC CS1</td></tr>
<tr><td>-</td><td>-</td><td>0</td><td>-</td><td>-</td><td>-</td><td>NFCON CS1</td></tr>
<tr><td rowspan="2">Xm0CSn[4]</td><td>-</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>SROMC CS4</td></tr>
<tr><td>-</td><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>CFCON CS0</td></tr>
<tr><td rowspan="2">Xm0CSn[5]</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>SROMC CS5</td></tr>
<tr><td>1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>CFCON CS1</td></tr>
</table>
</td><td>0x00</td></tr>
</table>

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

<table>
<tr><th>QOS_OVERRID1</th><th>BIT</th><th>DESCRIPTION</th><th>RESET VALUE</th></tr>
<tr><td>RESERVED</td><td>[31:16]</td><td>RESERVED</td><td>0x0000_0000</td></tr>
<tr><td>QOS_OV_ID</td><td>[15:0]</td><td>Override Quality of Service for DMC1.
When one or more bits are high, and when the arid match bits in DMC1 are equivalent to the QOS_OV_ID bits, then the quality of service of the read access is forced to minimum latency.
AXI master IDs are as follows:
<table>
<tr><th>Assigned AXI ID</th><th>Master name</th><th>Related IPs</th></tr>
<tr><td>0x00</td><td>I_BLOCK</td><td>Camera and JPEG</td></tr>
<tr><td>0x01</td><td>F_BLOCK</td><td>Display</td></tr>
<tr><td>0x02</td><td>P_BLOCK</td><td>POST</td></tr>
<tr><td>0x03</td><td>V_BLOCK</td><td>MFC</td></tr>
<tr><td>0x04</td><td>X_BLOCK</td><td>HSMMC and OTG</td></tr>
<tr><td>0x05</td><td>T_BLOCK</td><td>Host I/F</td></tr>
<tr><td>0x06</td><td>M_BLOCK</td><td>DMA</td></tr>
<tr><td>0x07</td><td>S_BLOCK</td><td>Security</td></tr>
<tr><td>0x08</td><td>ARMI</td><td>ARM Core Instruction</td></tr>
<tr><td>0x09</td><td>ARMRW</td><td>ARM Core Data</td></tr>
<tr><td>0x0A</td><td>ARMD</td><td>ARM Core DMA</td></tr>
<tr><td>0x0B</td><td>CF</td><td>CFCON</td></tr>
<tr><td>0x0C</td><td>G_BLOCK</td><td>3D</td></tr>
<tr><td>0x0D</td><td>G_BLOCK</td><td>3D</td></tr>
<tr><td>0x0E</td><td>G2D</td><td>G2D</td></tr>
</table>
</td><td>0000</td></tr>
</table>

MEM_CFG_STAT	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:17]	RESERVED	0x0000_00
RESERVED	[16]	RESERVED	0
CFG_PRI_TYPE	[15]	Current EBI priority scheme. See the EBI_PRI field of MEM_SYS_CFG register	0
CFG_FIX_PRI_TYPE	[14:12]	Current EBI fixed priority setting. See the EBI_FIX_PRI field of MEM_SYS_CFG register.	0
RESERVED	[11]	RESERVED	0
CFG_INDEP_CF	[10]	Show CF interface independently.	0
CFG_MUX_FLASH	[9]	Show NAND Flash type setting. 0 = Use OneNAND Controller.	0

		1 = RESERVED	
CFG_NOR_BOOT	[8]	0 = 16-bit width NOR booting is not selected. 1 = 16-bit width NOR booting is selected.	XOM dependent
RESERVED	[7]	RESERVED	
CFG_BOOT_LOC	[6:5]	Show with which area 0x00000000 address area is aliased. 00 = RESERVED 01 = SROMC CS0 10 = OneNANDC CS0 11 = Internal ROM	XOM dependent
CFG_ADDR_EXPAND (CFG_SROM_ADDR_EXPAND_To_DRAM1)	[4]	Show whether Xm1DATA[31:16] pins are used for SROMC address field or not. 0 = Xm1DATA[26:16] pins are used for DMC1 upper halfword data field, data[26:16]. 1 = Xm1DATA[26:16] pins are used for SROMC upper 10-bit address field, address[26:16].	0
RESERVED	[3]	RESERVED	0
RESERVED	[2]	RESERVED	0
CFG_BUS_WIDTH	[1]	Show SROMC CS0 memory bus width initial setting. 0 = 8-bit 1 = 16-bit	0
RESERVED	[0]	RESERVED	0

Note) 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

3.4.2.11 Power mode control register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
PWR_CFG	0x7E00_F804	R/W	Configure power manager	0x0000_0001
EINT_MASK	0x7E00_F808	R/W	Configure EINT mask	0x0000_0000
NORMAL_CFG	0x7E00_F810	R/W	Configure power manager at NORMAL mode	0xFFFF_FF00
STOP_CFG	0x7E00_F814	R/W	Configure power manager at STOP mode	0x2012_0100
SLEEP_CFG	0x7E00_F818	R/W	Configure power manager at SLEEP mode	0x0000_0000
STOP_MEM_CFG	0x7E00_F81C	R/W	Configure memory power at STOP mode	0x0000_007F

PWR_CFG	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:18]	RESERVED	0x0000
OSC_{OTG}_DISABLE	[17]	Control OSC_{OTG} clock pad (0: clock enable, 1: clock disable)	0
MMC2_WAKEUP_MASK	[16]	MMC2 wake-up source (0: use as a wakeup source, 1: disable)	0
MMC1_WAKEUP_MASK	[15]	MMC1 wake-up source (0: use as a wakeup source, 1: disable)	0
MMC0_WAKEUP_MASK	[14]	MMC0 wake-up source (0: use as a wakeup source, 1: disable)	0
HSI_WAKEUP_MASK	[13]	HSI wake-up source (0: use as a wakeup source, 1: disable)	0
TS_WAKEUP_MASK	[12]	Touch screen wake-up source (0: use as a wakeup source, 1: disable)	0
TICK_WAKEUP_MASK	[11]	RTC TICK wake-up source (0: use as a wakeup source, 1: disable)	0
ALARM_WAKEUP_MASK	[10]	RTC alarm wake-up source (0: use as a wakeup source, 1: disable)	0
MSM_WAKEUP_MASK	[9]	MSM modem wake-up source (0: use as a wakeup source, 1: disable)	0
KEY_WAKEUP_MASK	[8]	Key pad wake-up source (0: use as a wakeup source, 1: disable)	0
BATF_WAKEUP_MASK	[7]	BATF wake-up source (0: disable, 1: use as a wakeup source and only use when CFG_BATFLT field has 01)	0
CFG_STANDBYWFI	[6:5]	Configure ARM1176 STADNBYWFI 00: ignore 01: Enter IDLE mode 10 : Enter STOP mode 11 : Enter SLEEP mode	0x0
CFG_BATFLT	[4:3]	Configure nBAT_FLT 00 : ignore	0x0

		01 : generate interrupt (only use when BATF_WAKEUP_MASK field has 1) 11 : enter SLEEP mode or E-SLEEP mode	
CFG_BATF_WKUP	[2]	Configure wakeup source after XnBATF is cleared This bit should be 1 to use all SLEEP wakeup sources as BATF_WKUP sources	0
RESERVED	[1]	RESERVED (DO NOT SET THIS BIT)	0
OSC27_EN	[0]	Control 27MHz X-tal oscillator pad (0: clock disable, 1: clock enable)	1

EINT_MASK	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:28]	RESERVED	0x0
EINT_WAKEUP_MASK	[27:0]	External interrupt wake-up mask EINT[27:0] (0: use as a wakeup source, 1: disable) This field affects on NORMAL mode. Therefore, this field must clear when EINT is used as a normal external interrupt source.	0x00_0000

NORMAL_CFG	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31]	RESERVED	1
IROM	[30]	0: LP mode(OFF), 1: active mode(ON)	1
RESERVED	[29]	RESERVED	1
RESERVED	[28:21]	DO NOT CHANGE	0xFF
RESERVED	[20:17]	RESERVED	0xF
DOMAIN_ETM	[16]	0: LP mode(OFF), 1: active mode(ON)	1
DOMAIN_S	[15]	0: LP mode(OFF), 1: active mode(ON)	1
DOMAIN_F	[14]	0: LP mode(OFF), 1: active mode(ON)	1
DOMAIN_P	[13]	0: LP mode(OFF), 1: active mode(ON)	1
DOMAIN_I	[12]	0: LP mode(OFF), 1: active mode(ON)	1
RESERVED	[11]	RESERVED	0x1
DOMAIN_G	[10]	0: LP mode(OFF), 1: active mode(ON)	0x1
DOMAIN_V	[9]	0: LP mode(OFF), 1: active mode(ON)	1
RESERVED	[8]	DO NOT CHANGE	1
RESERVED	[7:0]	RESERVED	0x0

STOP_CFG	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31]	DO NOT CHANGE	0
RESERVED	[30]	RESERVED	0
MEMORY_ARM	[29]	0: LP mode(OFF), 1: active mode(ON)	1
RESERVED	[28:21]	RESERVED	0x00
TOP_MEMORY	[20]	0: LP mode(OFF), 1: active mode(Retention)	1
RESERVED	[19:18]	RESERVED	0x0
ARM_LOGIC	[17]	0: LP mode(OFF), 1: active mode(ON)	1
RESERVED	[16:9]	DO NOT CHANGE	0x00
TOP_LOGIC	[8]	0: LP mode(Retention), 1: active mode(ON), This field must be '0' before entering STOP mode.	1
RESERVED	[7:1]	RESERVED	0x00
OSC_EN	[0]	Control X-tal oscillator pad in STOP mode (0: disable, 1: enable)	0

SLEEP_CFG	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:1]	RESERVED	0x0000_0000
OSC_EN	[0]	Control X-tal oscillator pad in SLEEP mode (0: disable, 1: enable)	0

STOP_MEM_CFG	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:7]	RESERVED	0x0
MODEMIF_CFG	[6]	MODEM IF Block Memory control (0: Off, 1 : Retention)	1
HOSTIF_CFG	[5]	HOST IF Block Memory control (0: Off, 1 : Retention)	1
OTG_CFG	[4]	OTG Memory control (0: Off, 1 : Retention)	1
HSMMC_CFG	[3]	HSMMC Memory control (0: Off, 1 : Retention)	1
IROM_CFG	[2]	IROM control (0: Off, 1 : Retention)	1
IrDA_CFG	[1]	IrDA memory control (0: Off, 1 : Retention)	1
NFCON_CFG	[0]	NFCON Memory control (0: Off, 1 : Retention)	1

Note) 6410X PoP A type doesn't support NAND Flash. Set NFCON_CFG to '0' when stepping stone memory is not used.

3.4.2.12 System stabilization counter

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
OSC_FREQ	0x7E00_F820	R/W	Oscillator frequency scale counter	0x0000_000F
OSC_STABLE	0x7E00_F824	R/W	Oscillation pad stable counter	0x0000_0001
PWR_STABLE	0x7E00_F828	R/W	Power stable counter	0x0000_0001
MTC_STABLE	0x7E00_F830	R/W	MTC stable counter	0xFFFF_FFFF

OSC_FREQ	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:4]	RESERVED	0x0000
OSC_FREQ_VALUE	[3:0]	Oscillator frequency scale counter (OSC_FREQ_VALUE / *oscillator_frequency* > 200ns)	0xF

OSC_STABLE	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:20]	RESERVED	0x0000_000
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, OSC_CNT_VALUE[19:4] must be smaller than PWR_CNT_VALUE[19:4])	
OSC_CNT_VALUE	[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

PWR_STABLE	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:20]	RESERVED	0x0000_000
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, OSC_CNT_VALUE[19:4] must be smaller than PWR_CNT_VALUE[19:4])	
PWR_CNT_VALUE	[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

MTC_STABLE	BIT	DESCRIPTION	RESET VALUE
DOMAIN_G	[31:28]	Memory power stabilization counter for domain G	0xF
DOMAIN_ETM	[27:24]	Memory power stabilization counter for domain ETM	0xF
DOMAIN_S	[23:20]	Memory power stabilization counter for domain S	0xF
DOMAIN_F	[19:16]	Memory power stabilization counter for domain F	0xF
DOMAIN_P	[15:12]	Memory power stabilization counter for domain P	0xF
DOMAIN_I	[11:8]	Memory power stabilization counter for domain I	0xF
DOMAIN_V	[7:4]	Memory power stabilization counter for domain V	0xF
DOMAIN_TOP	[3:0]	Memory power stabilization counter for domain TOP	0xF

MTC_STABLE represents the number of external oscillator (or clock) cycles. When a sub-block returns from MTC mode to normal operation mode, the internal power stabilization time is required. This period must be larger than 200nsec and it can be estimated using MTC_STABLE value and OSC_FREQ registers.

3.4.2.13 Cacheable bus transaction selection register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
MISC_CON	0x7E00_F838	R/W	Bus/SYNC667 control	0x0000_0000

MISC_CON	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:20]	RESERVED	0x0000
SYNC667	[19]	0 : Normal Moe, 1 : Sync 667MHz Mode	0
STOP_EXIT_CKE_L	[18]	0 : Normal, 1 : CKE Keep L until Stop Exit	0
RESERVED	[17:12]	RESERVED	0
CACHEABLE_AHB_CF	[11]	Sets whether master transaction from CF controller is cacheable or not. 0: non-cacheable ,1: cacheable	0
RESERVED	[10:8]	RESERVED	0
CACHEABLE_AHB_S	[7]	Sets whether master transaction from AHB_S sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0
CACHEABLE_AHB_M	[6]	Sets whether master transaction from AHB_M sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0
CACHEABLE_AHB_T	[5]	Sets whether master transaction from AHB_T sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0

CACHEABLE_AHB_X	[4]	Sets whether master transaction from AHB_X sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0
WRITE_CACHEABLE_ AXI_V	[3]	Sets whether master write transaction from AXI_V sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0
CACHEABLE_AHB_P	[2]	Sets whether master transaction from AHB_P sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0
CACHEABLE_AHB_F	[1]	Sets whether master transaction from AHB_F sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0
CACHEABLE_AHB_I	[0]	Sets whether master transaction from AHB_I sub-block is cacheable or not. 0: non-cacheable ,1: cacheable	0

3.4.2.14 Others control register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
OTHERS	0x7E00_F900	R/W	Others control register	0x0000_801E

OTHERS	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:24]	RESERVED	0x0000
STABLE COUNTER TYPE	[23]	Indicate OSC_STABLE, PWR_STABLE counter type 0 : Exponential Scale, 1 : Set by SFR	0
RESERVED	[22:17]	DO NOT CHANGE	0x0000
USB_SIG_MASK	[16]	USB signal mask to prevent unwanted leakage. (This bit must set before USB PHY is used.)	0
RESERVED	[15:14]	RESERVED	0x2
CLEAR_DBGACK	[13]	Clear DBGACK signal when this field has 1. ARM1176 asserts DBGACK signal to indicate the system has entered Debug state. If DBGACK is asserted, this state is store in SYSCON until software clear it using this field.	0
CLEAR_BATF_INT	[12]	Clear interrupt caused by battery fault when this bit is set.	0
SYNCACK	[11:8]	SYNC mode acknowledge (Read Only)	0x0
SYNCMODE	[7]	SYNCMODEREQ to ARM 0: Asynchronous mode, 1: Synchronous mode	0
SYNCMUXSEL	[6]	SYS CLOCK SELECT IN CMU	0

		0: $MOUT_{MPLL}$, 1: $DOUT_{APLL}$	
RESEVED	[5:3]	DO NOT CHANGE	0x3
RESERVED	[2]	Should be '1'	1
RESERVED	[1]	Should be '1'	1
CP15DISABLE	[0]	Disables write asses to some system control processor registers of ARM1176. (0: enable, 1: disable)	0

3.4.2.15 Status register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
RST_STAT	0x7E00_F904	R	Reset status register	0x0000_0001
WAKEUP_STAT	0x7E00_F908	R/W	Wake-up status register	0x0000_0000
BLK_PWR_STAT	0x7E00_F90C	R	Block power status register	0x0000_007F

RST_STAT	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:7]	RESERVED	0x0000_000
DEEP_STOP_WAKEUP	[6]	Reset by DEEP_STOP mode wake-up	0
RESERVED	[5]	RESERVED	0
E-SLEEP WAKEUP	[4]	Reset by E-SLEEP mode wake-up	0
SLEEP_WAKEUP	[3]	Reset by SLEEP mode wake-up	0
WDT_RESET	[2]	Watch dog timer reset by WDTRST	0
RESERVED	[1]	RESERVED	0
HW_RESET	[0]	External reset by XnRESET	1

WAKEUP_STAT	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:12]	RESERVED	0x0000_0
MMC2_WAKEUP	[11]	Wake-up by MMC2. This is cleared by writing 1.	0
MMC1_WAKEUP	[10]	Wake-up by MMC1. This is cleared by writing 1.	0
MMC0_WAKEUP	[9]	Wake-up by MMC0. This is cleared by writing 1.	0
HSI_WAKEUP	[8]	Wake-up by HSI. This is cleared by writing 1.	0
RESERVED	[7]	RESERVED	0
BATFLT_WAKEUP	[6]	Wake-up by battery fault. This is cleared by writing 1.	0
MSM_WAKEUP	[5]	Wake-up by MSM modem. This is cleared by writing 1.	0
KEY_WAKEUP	[4]	Wake-up by Key PAD. This is cleared by writing 1.	0
TS_WAKEUP	[3]	Wake-up by touch screen. This is cleared by writing 1.	0
RTC_TICK_WAKEUP	[2]	Wake-up by tick interrupt. This is cleared by writing 1.	0
RTC_ALARM_WAKEUP	[1]	Wake-up by RTC alarm. This is cleared by writing 1.	0
EINT_WAKEUP	[0]	Wake-up by external interrupts. This is cleared by writing 1.	0

BLK_PWR_STAT	BIT	DESCRIPTION	RESET VALUE
RESERVED	[31:9]	RESERVED	0x0000_000
RESERVED	[8]	RESERVED	0
BLK_G	[7]	Block G power ready	1
BLK_ETM	[6]	Block ETM power ready	1
BLK_S	[5]	Block S power ready	1
BLK_F	[4]	Block F power ready	1
BLK_P	[3]	Block P power ready	1
BLK_I	[2]	Block I power ready	1
BLK_V	[1]	Block V power ready	1
BLK_TOP	[0]	Block TOP power ready	1

3.4.2.16 Information register

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
INFORM0	0x7E00_FA00	R/W	Information register 0	0x0000_0000
INFORM1	0x7E00_FA04	R/W	Information register 1	0x0000_0000
INFORM2	0x7E00_FA08	R/W	Information register 2	0x0000_0000
INFORM3	0x7E00_FA0C	R/W	Information register 3	0x0000_0000

INFORMn	BIT	DESCRIPTION	RESET VALUE
INFORM	[31:0]	User defined information. INFORM0~3 registers are cleared by asserting XnRESET pin.	0x0000_0000

4 MEMORY SUB-SYSTEM

4.1. OVERVIEW

The S3C6410X Memory Subsystem includes seven memory controllers, one SROM controller, two OneNAND controllers, one NAND Flash controller, one CF controller, and one DRAM controllers. Static memory controllers , OneNAND Controller, NAND Controller and CF Controller share memory port 0 by using EBI.

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

4.2. INTRODUCTION

S3C6410X Memory Sub-system features are as follows:

- Memory Subsystem has one 64-bit AXI slave interface, one 32-bit AXI slave interface, one 32-bit AHB Master interface, two 32-bit AHB slave interfaces, one for data transfer and the other for SFR setting, and one APB interface for DMC SFR setting.
- Memory Subsystem gets booting method and CS selection information from the System Controller.
- Internal AHB data bus connects 32-bit AHB slave data bus with SROMC, two OneNANDC and NFCON.
- Internal AHB SFR bus connects 32-bit AHB slave SFR bus with SROMC, two OneNANDC, CFCON and NFCON.
- Internal AHB master bus is used for CFCON.
- DMC1 uses 64-bit AXI slave interface and APB interface.
- Memory port 0 is shared by using EBI (External Bus Interface).
- Memory port 1 is used only by DMC1.
- Selection of using NAND Flash or OneNAND is supported.
- EBI module supports AMBA AXI 3.0 low power interface (CSYSREQ, CACTIVE, CSYSACK) to prevent memory controllers from accessing memories.
- Data pin [26:16] of memory port 1 can be used as Address pin [26:16] of memory port 0 by configuration from system controller.
- EBI module supports the share of pad interface used by 5 memory controllers (SROMC, two OneNANDCs, CFCON, and NFCON).
- Pad interface ownership is determined by the priority which can be changed.
- The handshaking between the EBI and the memory controller consists of s a three-wire interface, EBIREQ, EBIGNT, and EBIBACKOFF, all active HIGH.

4.3. STRUCTURE

Figure 4-1 shows the structure of Memory sub-system. It holds eight memory controllers, which are shown in grey boxes. Interface ports of Memory sub-system includes two AHB slave ports, each of which is for accessing SFRs and memory contents, two APB slave ports , two AXI slave ports, single AHB master ports, and two memory interface ports. Configuration signal ports, which changes the operation options of Memory Sub-system, are not shown in Figure 4-1 for clarity. Description of these configuration signal ports are detailed in later section. For memory port 0, EBI is used to multiplex output signals coming from each memory controller to generate a unified control signals for memory devices. Figure 4-1 also shows “reverse data” block, which reverses the order of bit assignments of data port of DRAM controller #1. This is useful when OneDRAM is attached to DRAM controller #1 as OneDRAM has reverse data port order compared to mobile DRAM.

4.4. AHB SLAVE INTERFACE (SFR)

ARM can access SFR area of all memory controllers through AHB slave interface port (SFR) of Memory Sub-system. Table 4-1 shows the assignment of slave ID to each memory controller.

Table 4-1 Slave ID for AHB slave interface (SFR)

Slave ID	Memory Controller ID
0	SROM Controller
1	OneNAND Controller #0
2	NAND Flash Controller
3	CF Controller
4	OneNAND Controller #1
5-14	none
15	Default slave

Note) 6410X PoP A type doesn’t support NAND Flash. Don’t care the description regarding NAND Flash. 6410X PoP D type doesn’t support OneNAND Flash. Don’t care the description regarding OneNAND Flash.

4.5. AHB SLAVE INTERFACE (SPINE)

ARM can access memory area of all memory controllers through AHB slave interface port (SPINE) of Memory Sub-system. Table 4-2 shows the assignment of slave ID to each memory controller.

Table 4-2 Slave ID for AHB slave interface (SPINE)

Slave ID	Memory Controller ID
0	NAND Flash Controller
1	SROM Controller
2	OneNAND Controller #0
3	Internal ROM
4	OneNAND Controller #1
5-14	none
15	Default slave

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

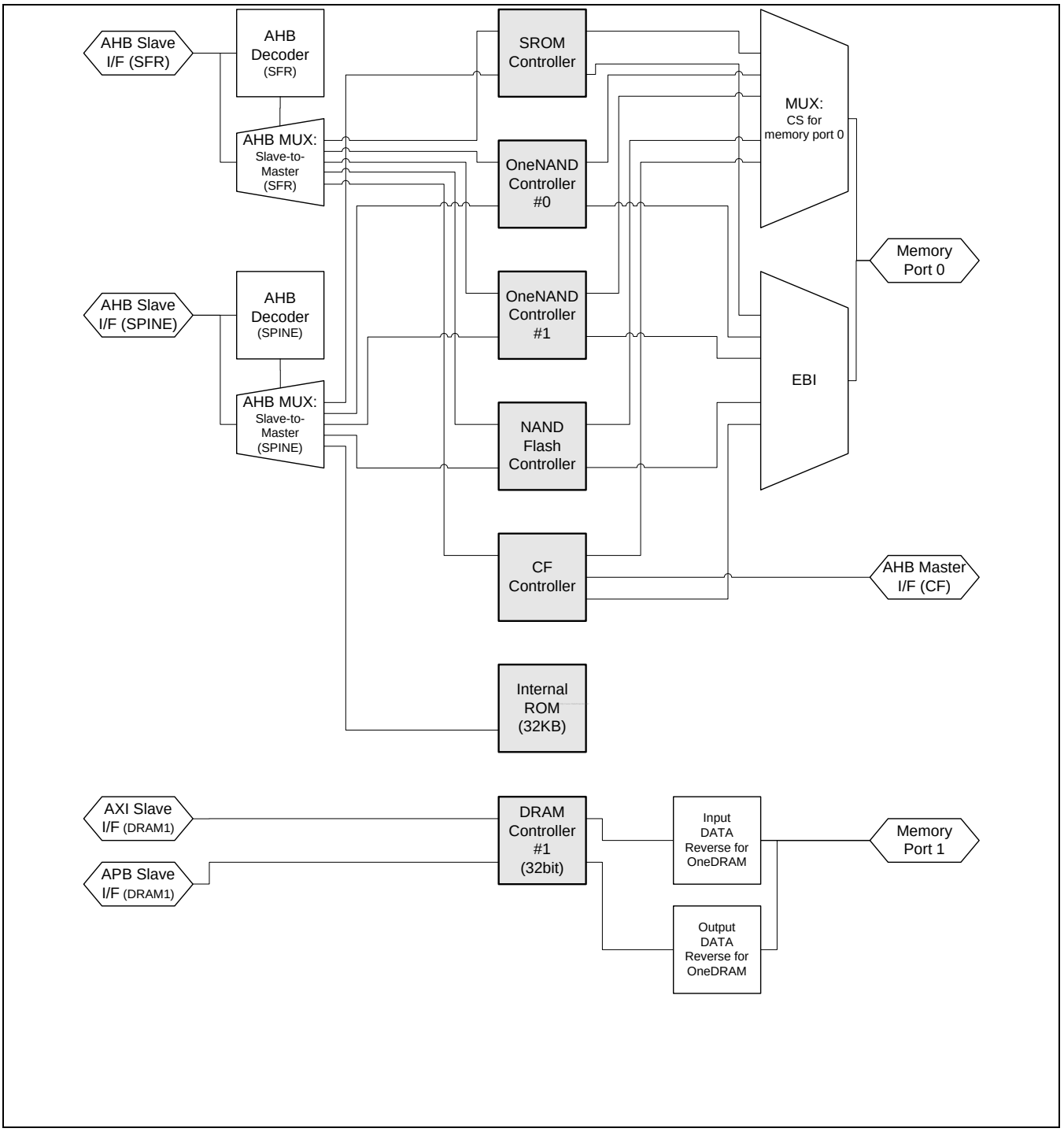


Figure 4-1 Structure of Memory Sub-system

4.6. EBI MULTIPLEXING

EBI is used to multiplex output signals coming from each memory controller to generate a unified control signals for memory devices. Table 4-3 shows which of memory controller participates in to generate each memory control signals.

Table 4-3 EBI Multiplexing Table.

ID	Memory Controller	Port name							
		Address (ADDR)	Output Data (DO)	Write Enable (WE)	Output Enable (OE)	Byte Enable (BE)	Reset Output (RP)	Advance (ADV)	Data OE (DOEN)
1	SROM Controller	O (ADDR)	O (WDATA)	O (nWE)	O (nOE)	O (nWBE)	X	X	O (DIR)
2	OneNAND Controller#0	X	O (DO)	O (nWE)	O (nOE)	X	O (nRP)	O (nADV)	O (nDOEN)
3	NAND Flash Controller	X	O (DO)	X	X	X	X	X	O (nDOEN)
4	CF Controller	O (ADDR)	O (DO)	O (nWE_IOWR)	O (nOE_IORD)	X	X	X	O (nDOEN)
5	OneNAND Controller#1	X	O (DO)	O (nWE)	O (nOE)	X	O (nRP)	O (nADV)	O (nDOEN)

Each row stands for each memory controller while each column shows whether each memory controller participates in to become an output memory control signal. The ID in the first column represents the identification number for EBI muxing and it is not related to AHB slave IDs.

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

As many different memory controllers compete to get the ownership of memory interface port 0, EBI uses priority table shown in Table 4-4 to decide which memory controller to have the ownership.

Table 4-4 EBI Priority. CfgFixPriority is a SFR in System Controller memory region.

CfgFixPriority [2:0]	1st (Highest)	2nd	3rd	4th	5th	6th (Lowest)
0	-	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
1	-	OneNAND Controller 0	OneNAND Controller 1	SROM Controller	NAND Flash Controller	CF Controller
2	-	OneNAND Controller 1	NAND Flash Controller	SROM Controller	OneNAND Controller 0	CF Controller
3	-	NAND Flash Controller	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	CF Controller
4	-	CF Controller	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller
5	-	-	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
6	-	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
7	-	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller

We can also set whether to use fixed priority as shown in Table 4-4 or to rotate them so that all the memory controllers can have a fair amount of opportunity to use the memory port 0. This is controlled by CfgPriType configuration register.

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

4.7. CONFIGURATION OPTIONS

There are several configuration options available to Memory Sub-system as summarized in Table 4-5. All of these configuration options are generated by system controller. Refer to system controller chapter for details.

Table 4-5 Summary of configuration options for Memory Sub-system

Name	Default Values	Description
CfgAddrExpandToDMC1	0	Decides whether to use upper data ports of memory port 1 to hold high address output port of SROM Controller. 0: Data port of memory port 1 is used by DRAM controller, 1: Part of data port of memory port 1 is used by SROM Controller to hold higher address bits.
CfgBootLoc		Decides booting device 00: RESERVED 01: SROM Controller 10: OneNAND Controller 11: Internal ROM
CfgPriTyp		Decides whether fixed priority type or rotating priority type is used for EBI. 0: fixed priority 1: rotating priority
CfgFixPriTyp		Decides fixed priority of each memory controller for using EBI MUX.
CfgSelECC		Decides whether to use ECC logic in NAND Flash Controller.

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash.
6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

Table 4-6 Sharing of XM1DATA between DRAM Controller 1 and SROM Controller

	XM1DATA]31:16]															
CfgAddr-Expand-ToDMC1	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	DRAM 1 Data Port															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
1	SROM Controller Address port															
	-	-	-	-	-	26	25	24	23	22	21	20	19	18	17	16

.

4.8. USE OF WIDE ADDRESS SPACES FOR SROM CONTROLLER (USING DRAM 1 PORT)

SROM Controller has 27bit of address output ports, which allows it to access up to 128MBytes of memory space. But the default configuration only allows to use only lower 20bits of address ports of SROM Controller with limited address range of 1MBytes. When more than 1MBytes of SROM/RAM is necessary, we can you use upper 11bits of SROM address ports, which shares output ports with upper data ports of DRAM Controller 1 (XM1DATA). CfgAddrExpandToDMC1 in Table 4-6 decides which of SROM Controller or DRAM Controller 1 uses XM1DATA ports.

4.9. SHARING CHIP SELECT FOR MEMORY PORT 0

Like memory control output signals, multiple chip select (CS) output is also shared between multiple memory controllers. But, unlike memory control output signals who is dynamically assigned to one memory controller at one time and another controller at other times, the use of chip select is fixed and cannot be changed during operation. The decision of which memory controller owns specific memory bank is decided both by following three factors.

- Booting mode
- MP0_CS_CFG regsiter residing in System Controller
- XSELNAND input port value

Following sections shows Chip Select Mux table for each booting mode.

NOR BOOTING MODE

Table 4-7 Chip Select Mux table for NOR booting mode

	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
Xm0CSn[0]	SROMC_ nCS[0]				
Xm0CSn[1]	SROMC_ nCS[1]	-	-	-	-
Xm0CSn[2]	SROMC_ nCS[2]	OneNANDC0 _nCS	-	NFCON_ nCS[0]	-
	MP0_CS_ CFG[1] = H	MP0_CS_ CFG[1] = L & XSELNAND=L		MP0_CS_ CFG[1] = L & XSELNAND=H	
Xm0CSn[3]	SROMC_ nCS[3]	-	OneNANDC1 _nCS	NFCON_ nCS[1]	-
	MP0_CS_ CFG[3] = H	-	MP0_CS_ CFG[3] = L & XSELNAND=L	MP0_CS_ CFG[3] = L & XSELNAND=H	
Xm0CSn[4]	SROMC_ nCS[4]	-	-	-	CFCON_ nCS[0]
	MP0_CS_CFG [4] = L	-	-	-	MP0_CS_CFG [4] = H
Xm0CSn[5]	SROMC_ nCS[5]	-	-	-	CFCON_ nCS[1]
	MP0_CS_CFG [5] = L	-	-	-	MP0_CS_CFG [5] = H

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

A. ONENAND BOOTING MODE

Table 4-8 Chip Select Mux table for NAND booting mode

	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
Xm0CSn[0]	SROMC_ nCS[0]				
Xm0CSn[1]	SROMC_ nCS[1]	-	-	-	-
Xm0CSn[2]	-	OneNANDC0 _nCS	-	-	-
Xm0CSn[3]	-	-	OneNANDC1 _nCS	-	-
Xm0CSn[4]	SROMC_ nCS[4]	-	-	-	CFCON_ nCS[0]
	MP0_CS_CFG [4] = L	-	-	-	MP0_CS_CFG [4] = H
Xm0CSn[5]	SROMC_ nCS[5]	-	-	-	CFCON_ nCS[1]
	MP0_CS_CFG [5] = L	-	-	-	MP0_CS_CFG [5] = H

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

B. MODEM BOOTING MODE

Table 4-9 Chip Select Mux table for Modem booting mode

	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
Xm0CSn[0]	SROMC_ nCS[0]				
Xm0CSn[1]	SROMC_ nCS[1]	-	-	-	-
Xm0CSn[2]	-	OneNANDC0 _nCS	-	NFCON_ nCS[0]	-
	-	MP0_CS_ CFG[1] = L & XSELNAND=L		MP0_CS_ CFG[1] = L & XSELNAND=H	
Xm0CSn[3]	-	-	OneNANDC1 _nCS	NFCON_ nCS[1]	-
	-	-	MP0_CS_ CFG[3] = L & XSELNAND=L	MP0_CS_ CFG[3] = L & XSELNAND=H	
Xm0CSn[4]	SROMC_ nCS[4]	-	-	-	CFCON_ nCS[0]
	MP0_CS_CFG [4] = L	-	-	-	MP0_CS_CFG [4] = H
Xm0CSn[5]	SROMC_ nCS[5]	-	-	-	CFCON_ nCS[1]
	MP0_CS_CFG [5] = L	-	-	-	MP0_CS_CFG [5] = H

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

C. IROM MOVINAND BOOTING MODE

Table 4-10 Chip Select Mux table for IROM MoviNAND booting mode

	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
Xm0CSn[0]	SROMC_ nCS[0]				
Xm0CSn[1]	SROMC_ nCS[1]	-	-	-	-
Xm0CSn[2]	SROMC_ nCS[2]	OneNANDC0 _nCS	-	NFCON_ nCS[0]	-
	MP0_CS_ CFG[1] = H	MP0_CS_ CFG[1] = L & XSELNAND=L		MP0_CS_ CFG[1] = L & XSELNAND=H	
Xm0CSn[3]	SROMC_ nCS[3]	-	OneNANDC1 _nCS	NFCON_ nCS[1]	-
	MP0_CS_ CFG[3] = H	-	MP0_CS_ CFG[3] = L & XSELNAND=L	MP0_CS_ CFG[3] = L & XSELNAND=H	
Xm0CSn[4]	SROMC_ nCS[4]	-	-	-	CFCON_ nCS[0]
	MP0_CS_CFG [4] = L	-	-	-	MP0_CS_CFG [4] = H
Xm0CSn[5]	SROMC_ nCS[5]	-	-	-	CFCON_ nCS[1]
	MP0_CS_CFG [5] = L	-	-	-	MP0_CS_CFG [5] = H

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

D. INTERNAL ROM BOOTING (SECURE ROM BOOTING) MODE

Table 4-11 Chip Select Mux table for internal ROM booting mode

	SROM Controller	OneNAND Controller 0	OneNAND Controller 1	NAND Flash Controller	CF Controller
Xm0CSn[0]	SROMC_ nCS[0]				
Xm0CSn[1]	SROMC_ nCS[1]	-	-	-	-
Xm0CSn[2]	SROMC_ nCS[2]	OneNANDC0 _nCS	-	NFCON_ nCS[0]	-
	MP0_CS_ CFG[1] = H	MP0_CS_ CFG[1] = L & XSELNAND=L		MP0_CS_ CFG[1] = L & XSELNAND=H	
Xm0CSn[3]	SROMC_ nCS[3]	-	OneNANDC1 _nCS	NFCON_ nCS[1]	-
	MP0_CS_ CFG[3] = H	-	MP0_CS_ CFG[3] = L & XSELNAND=L	MP0_CS_ CFG[3] = L & XSELNAND=H	
Xm0CSn[4]	SROMC_ nCS[4]	-	-	-	CFCON_ nCS[0]
	MP0_CS_CFG [4] = L	-	-	-	MP0_CS_CFG [4] = H
Xm0CSn[5]	SROMC_ nCS[5]	-	-	-	CFCON_ nCS[1]
	MP0_CS_CFG [5] = L	-	-	-	MP0_CS_CFG [5] = H

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

4.10. REMAPPING

Address ranges from 0x00000000 to 0x07FFFFFF are shared by bootable devices. According to configurations, this address ranges can be assigned to SROM Controller, internal ROM, NAND Flash controller, or OneNAND controller.

Table 4-12 Remapping Condition

Booting Device (Device remapped to address 0)	Conditions		
	REMAP	CfgBootLoc	MP0_CS_CFG[1]
Internal ROM	1'b1	2'b11	doesn't care
SROM (External ROM)	1'b0	2'b01	H
OneNAND	1'b0	2'b10	L

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

4.11. ADDRESS DECODING

Address ranges are described in Table 4-.

Table 4-13 Address map.

Start Address	End Address	Int. ROM	Stepping Stone (NAND Ctrl.)	SROM Ctrl.	One NAND Ctrl. 0	One NAND Ctrl. 1	DRAM Ctrl 1
0x00000000	0x07FFFFFF	O[1]	-	O[1]	O[1]	-	-
0x08000000	0x0BFFFFFF	O	-	-	-	-	-
0x0C000000	0x0FFFFFFF	-	O	-	-	-	-
0x10000000	0x17FFFFFF	-	-	O	-	-	-
0x18000000	0x1FFFFFFF	-	-	O	-	-	-
0x20000000	0x27FFFFFF	-	-	O[2]	O[2]	-	-
0x28000000	0x2FFFFFFF	-	-	O[2]		O[2]	-
0x30000000	0x37FFFFFF	-	-	O	-	-	-
0x38000000	0x3FFFFFFF	-	-	O	-	-	-
0x40000000	0x47FFFFFF	-	-	-	-	-	-
0x48000000	0x4FFFFFFF	-	-	-	-	-	-
0x50000000	0x5FFFFFFF	-	-	-	-	-	O
0x60000000	0x6FFFFFFF	-	-	-	-	-	O

Boxes in thick borders show that the address ranges shown on the first two columns are directed to the corresponding memory controllers. Refer to the footnote at the bottom of this table for cases where single address range can be directed to multiple memory controllers.

Note) 6410X PoP A type doesn't support NAND Flash. Don't care the description regarding NAND Flash. 6410X PoP D type doesn't support OneNAND Flash. Don't care the description regarding OneNAND Flash.

[1] Refer to Table 4-12 for details.
[2] Refer to Table 4-9 ~ 4-13.

5 DRAM CONTROLLER

5.1 OVERVIEW

DRAM Controller is from ARM PrimeCell CP003 AXI Dynamic Memory Controller (PL340). Original AMBA APB 3.0 port for programming configuration registers is covered by using AxiToApb bridge component, which implements an AXI slave port connected to an APB master port.

DRAM Controller has AMBA AXI compatible bus for programming its configuration registers and for access to SDRAM. DRAM Controller can be programmed by writing chip configuration, ID configuration, and memory timing parameters in PL340 configuration registers.

DRAM Controller can receive a direct command for SDRAM or DRAM Controller itself. By writing command to direct_cmd register, DRAM Controller can send commands like ‘Prechargeall’, ‘Autorefresh’, ‘NOP’, and ‘MRS’ (‘EMRS’) to SDRAM. By writing command to memc_cmd register, DRAM Controller can enter into states like ‘Config’, ‘Ready’, and ‘Low_power’.

DRAM Controller supports power-down in two ways. The DRAM Controller can automatically place the SDRAM into either the pre-charge power-down or active power-down state, by de-asserting DMC1_CKE when the SDRAM has been inactive for a number of clock cycles set in memory_cfg. The DRAM controller can place the SDRAM into the self-refresh state when enter the power down mode such as STOP, Deep Stop, Sleep Mode.

The auto-refresh command is issued to SDRAM periodically when refresh counter reaches the value of the refresh period in auto-refresh period register.

5.2 FEATURES

- Supports SDR SDRAM, mobile SDR SDRAM, DDR SDRAM, and mobile DDR SDRAM
- Supports 2 external memory chips.
- Supports 64-bit AMBA AXI bus.
- Supports 16/32-bit memory bus.
  Memory Port 1: Support 16bit DDR SDRAM and mobile DDR SDRAM.
  Support 32bit DDR SDRAM ,mobile DDR SDRAM, mobile SDR SDRAM and SDR SDRAM
- Address space: Memory Port1 can support up to 2Gbit per chip select.
- Provides active and pre-charge power down.
- Quality of Service features for low latency transfers.
- Optimized utilization of external memory bus.
- Support to select external memory types by setting SFR.
- Supports 2 outstanding exclusive access transfers.
- Configurable memory access timing by using SFRs.
- Support extended MRS (EMRS) set.
- For Memory Port 1, not supports 16bit SDR SDRAM, mobile SDR SDRAM
- Operation voltage
  Memory Port 1: 1.8V, 2.5V (typical)

Note) 6410X PoP A type and PoP D type don't support SDR SDRAM, mobile SDR SDRAM, and DDR SDRAM because those types contain mobile DDR SDRAM only.

5.3 SDRAM MEMORY INTERFACE

DRAM Controller supports up to two chips of same type and can assign a maximum of 256 Mbytes(Memory Port1)address space per chip. All chips in the same port share all pins, except clock enable signals and chip select signals. External Memory Pin configuration is as shown in Table 5-1

Reset value of CKE is controlled by SPCONSLP[5]. If the value is zero Xm1CKE are zero when reset. If the value is one, Xm1CKE are one when reset. But Reset value of CKE must be changed during without DRAM access.

Table 5-1. Memory Port 1 Pin Description

Signal	Type	Description
Xm1SCLK	Output	Memory clock
Xm1SCLKn	Output	Memory clock (negative)
Xm1CKE[1:0]	Output	Clock enable per chip
Xm1CSN[1:0]	Output	Chip select per chip (active low)
Xm1RAS	Output	Row address strobe (active low)
Xm1CAS	Output	Column address strobe (active low)
Xm1WEN	Output	Write enable (active low)
Xm1ADDR[13:0]	Output	Address bus
Xm1ADDR[15:14]	Output	Bank select
Xm1DATA[31:0]	Inout	Data bus
Xm1DQM[3:0]	Output	Data bus mask bits
Xm1DQS[3:0]	Inout	Data strobe inout, DDR and mDDR only

5.4 SDRAM INITIALIZATION SEQUENCE

On power-on reset, software must initialize the DRAM controller and each of the SDRAM connected to the DRAM controller. Refer to the SDRAM data sheet for the start up procedure. Example sequences are given below.

5.4.1 DRAM CONTROLLER INITIALIZATION SEQUENCE

- Program memc_cmd to '3'b100', which makes DRAM Controller enter 'Config' state.
- Write memory timing parameter, chip configuration, and id configuration registers.
- Wait 200us to allow SDRAM power and clock to stabilize. However, when CPU starts working, power and clock would already be stabilized.
- Execute memory initialization sequence.
- Program memc_cmd to '3'b000', which makes DRAM Controller enter 'Ready' state.
- Check memory status field in memc_stat until memory status becomes '2'b01', which means 'Ready'.

5.4.2 SDR/MOBILE SDR SDRAM INITIALIZATION SEQUENCE

- Program mem_cmd in direct_cmd to '2'b10', which makes DRAM Controller issue 'NOP' memory command.
- Program mem_cmd in direct_cmd to '2'b00', which makes DRAM Controller issue 'Prechargeall' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- If memory type is mobile SDR SDRAM,
  - Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command
  - Bank address for EMRS must be set.
- Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command.
  - Bank address for MRS must be set.

5.4.3 DDR/MOBILE DDR SDRAM INITIALIZATION SEQUENCE

- Program mem_cmd in direct_cmd to '2'b10', which makes DRAM Controller issue 'NOP' memory command.
- Program mem_cmd in direct_cmd to '2'b00', which makes DRAM Controller issue 'Prechargeall' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd in direct_cmd to '2'b11', which makes DRAM Controller issue 'Autorefresh' memory command.
- Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command
  - Bank address for EMRS must be set.
- Program mem_cmd to '2'b10' in direct_cmd, which makes DRAM Controller issue 'MRS' memory command.
  - Bank address for MRS must be set.

5.5 REGISTER DESCRIPTION

5.5.1 DRAM CONTROLLER STATUS REGISTER

Register	Address	R/W	Description	Reset Value
P1MEMSTAT	0x7E001000	R	32-bit DRAM controller status register	0x8C4

PnMEMSTAT	Bit	Description	Initial State
Reserved	[31:14]	Read undefined.	
Reserved	[13:12]	Read always 0	00
Reserved	[11:10]	Read always “10”	10
Reserved	[9]	Read always zero.	0
Memory chips	[8:7]	The maximum number of different chip selects that DRAM controller can supports: 01 = 2 chips 6410X only supports 2 chips, and “Memory chips” reads only as 01.	01
Memory type	[6:4]	The type of SDRAM that DRAM controller supports: 100 = Any of the followings: MSDR, SDR, MDDR, or DDR	100
Memory width	[3:2]	The width of the external memory 00 = 16-bit 01 = 32-bit 10 = reserved 11 = reserved	01
Controller status	[1:0]	The status of the DRAM controller 00 = Config. 01 = Ready 10 = Paused 11 = Low-Power	00

5.5.2 DRAM CONTROLLER COMMAND REGISTER

Register	Address	R/W	Description	Reset Value
P1MEMCCMD	0x7E001004	W	32-bit DRAM controller command register	

PnMEMCCMD	Bit	Description	Initial State
	[31:3]	Undefined. Write as Zero	
Memc_cmd	[2:0]	Changes the state of the DRAM controller 000 = Go 001 = Sleep 010 = Wakeup 011 = Pause 100 = Configure101~111 = Reserved	

5.5.3 DIRECT COMMAND REGISTER

Register	Address	R/W	Description	Reset Value
P1 DIRECTCMD	0x7E001008	W	32-bit DRAM controller direct command register	

PnDIRECTCMD	Bit	Description	Initial State
Reserved	[31:23]	Undefined. Write as Zero	
Extended Memory command	[22]	Extended memory command, see note after the table	
Chip number	[21:20]	Bits mapped to external memory chip address bits.	
Memory command	[19:18]	Determines the command required, see note after the table.	
Bank address	[17:16]	Bits mapped to external memory bank address bits when command is MRS or EMRS access.	
	[15:14]	Undefined. Write as Zero	
Address_13_to_0	[13:0]	Bits mapped to external memory address bits [13:0] when command is MRS or EMRS access.	

Note: Memory command encoding. This encoding uses the “Extended Memory command” bits concatenated to “Memory command”, therefore providing 3 bits.

3’b000 = Prechargeall

3’b001 = Autorefresh

3’b010 = Modereg or Extended modereg access

3’b011 = NOP

3’b100 = DPD (Deep Power Down)

All other combinations are illegal and might cause undefined behavior. A NOP command asserts all chip selects that are set as active_chips when the chip_nmbr is set to 0.

5.5.4 MEMORY CONFIGURATION REGISTER

Register	Address	R/W	Description	Reset Value
P1MEMCFG	0x7E00100C	R/W	32-bit DRAM controller memory config register	0x01_0020

PnMEMCFG	Bit	Description	Initial State
Reserved	[31]	Reserved. It should be write as Zero.	0
Reserved	[30:23]	Read undefined. Write as zero.	
Active chips	[22:21]	Enables the refresh command generation for the number of memory chips. It is only possible to generate commands up to and including the number of chips in the configuration defined in the DRAM controller status register: 00 = 1 chip 01 = 2 chips 10 = Reserved 11 = Reserved	00
QoS master bits	[20:18]	Encodes the four bits of the 8-bit AXI ARID that are used to select one of the 16 QoS values: 000 = ARID[3:0] 001 = ARID[4:1] 010 = ARID[5:2] 011 = ARID[6:3] 100 = ARID[7:4] 101~111 = Reserved	000
Memory burst	[17:15]	Encodes the number of data accesses that are performed to the SDRAM for each Read and Write command: 000 = Burst 1 001 = Burst 2 010 = Burst 4 011 = Burst 8 100 = Burst 16 101~111 = Reserved This value must also be programmed into SDRAM mode register using the DIRECTCMD register and must match it.	010
Stop_mem_clock	[14]	When enabled the memory clock is dynamically stopped when not performing an access to the SDRAM.	0
Auto power down	[13]	When Auto power down is set, the memory interface automatically places the SDRAM into power-down state by de-asserting CKE when the command FIFO has been empty for Power_down_prd memory clock cycles.	0
Power_down_prd	[12:7]	Number of memory clock cycles for auto power-down of SDRAM.	000000
AP bit	[6]	Encodes the position of the auto-precharge bit in the memory address: 0 = address bit 10. 1 = address bit 8.	0
Row bits	[5:3]	Encodes the number of bits of the AXI address that comprise the	100

		row address: 000 = 11 bits 001 = 12 bits 010 = 13 bits 011 = 14 bits 100 = 15 bits 101 = 16 bits	
Column bits	[2:0]	Encodes the number of bits of the AXI address that comprise the column address: 000 = 8 bits 001 = 9 bits 010 = 10 bits 011 = 11 bits 100 = 12 bits	000

5.5.5 REFRESH PERIOD REGISTER

Register	Address	R/W	Description	Reset Value
P1REFRESH	0x7E001010	R/W	32-bit DRAM controller refresh period register	0xA60

PnREFRESH	Bit	Description	Initial State
	[31:15]	Read undefined. Write as Zero	
Refresh period	[14:0]	Memory refresh period in memory clock cycles.	0xA60

5.5.6 CAS LATENCY REGISTER

Register	Address	R/W	Description	Reset Value
P1CASLAT	0x7E001014	R/W	32-bit DRAM controller CAS latency register	0x6

PnCASLAT	Bit	Description	Initial State
	[31:4]	Read undefined. Write as Zero	
CAS Latency	[3:1]	CAS latency in memory clock cycles.	011
CAS Half cycle	[0]	Encodes whether the CAS latency is half a memory clock cycle more than the value given in bits[3:1] 0 = Zero cycle offset to value in [3:1]. [0] is forced to 0 in MDDR and SDR mode. 1 = Half cycle offset to the value in [3:1].	0

5.5.7 T_DQSS REGISTER

Register	Address	R/W	Description	Reset Value
P1T_DQSS	0x7E001018	R/W	32-bit DRAM controller t_DQSS register	0x1

PnT_DQSS	Bit	Description	Initial State
	[31:2]	Read undefined. Write as Zero	
t_DQSS	[1:0]	Write to DQS in memory clock cycles.	1

5.5.8 T_MRD REGISTER

Register	Address	R/W	Description	Reset Value
P1T_MRD	0x7E00101C	R/W	32-bit DRAM controller t_MRD register	0x02

PnT_MRD	Bit	Description	Initial State
	[31:7]	Read undefined. Write as Zero	
t_MRD	[6:0]	Set mode register command time in memory clock cycles.	0x02

5.5.9 T_RAS REGISTER

Register	Address	R/W	Description	Reset Value
P1T_RAS	0x7E001020	R/W	32-bit DRAM controller t_RAS register	0x7

PnT_RAS	Bit	Description	Initial State
	[31:4]	Read undefined. Write as Zero	
t_RAS	[3:0]	Set RAS to precharge delay in memory clock cycles.	0x7

5.5.10 T_RC REGISTER

Register	Address	R/W	Description	Reset Value
P1T_RC	0x7E001024	R/W	32-bit DRAM controller t_RC register	0xB

PnT_RC	Bit	Description	Initial State
	[31:4]	Read undefined. Write as Zero	
t_RC	[3:0]	Set Active bank x to Active bank x delay in memory clock cycles.	0xB

5.5.11 T_RCD REGISTER

Register	Address	R/W	Description	Reset Value
P1T_RCD	0x7E001028	R/W	32-bit DRAM controller t_RCD register	0x1D

PnT_RCD	Bit	Description	Initial State
	[31:6]	Read undefined. Write as Zero	
scheduled_RCD	[5:3]	Set t_RCD-3	011
t_RCD	[2:0]	Set the RAS to CAS minimum delay in memory clock cycles	101

5.5.12 T_RFC REGISTER

Register	Address	R/W	Description	Reset Value
P1T_RFC	0x7E00102C	R/W	32-bit DRAM controller t_RFC register	0x212

PnT_RFC	Bit	Description	Initial State
	[31:10]	Read undefined. Write as Zero	
scheduled_RFC	[9:5]	Set t_RFC -3.	0x10
t_RFC	[4:0]	Set the autorefresh command time in memory clock cycles	0x12

5.5.13 T_RP REGISTER

Register	Address	R/W	Description	Reset Value
P1T_RP	0x7E001030	R/W	32-bit DRAM controller t_RP register	0x1D

PnT_RP	Bit	Description	Initial State
	[31:6]	Read undefined. Write as Zero	
scheduled_RP	[5:3]	Set t_RP -3.	011
t_RP	[2:0]	Set the precharge to RAS delay in memory clock cycles	101

5.5.14 T_RRD REGISTER

Register	Address	R/W	Description	Reset Value
P1T_RRD	0x7E001034	R/W	32-bit DRAM controller t_RRD register	0x2

PnT_RRD	Bit	Description	Initial State
	[31:4]	Read undefined. Write as Zero	
t_RRD	[3:0]	Set Active bank x to Active bank y delay in memory clock cycles.	0x2

5.5.15 T_WR REGISTER

Register	Address	R/W	Description	Reset Value
P1T_WR	0x7E001038	R/W	32-bit DRAM controller t_WR register	0x3

PnT_WR	Bit	Description	Initial State
	[31:3]	Read undefined. Write as Zero	
t_WR	[2:0]	Set the write to precharge delay in memory clock cycles.	011

5.5.16 T_WTR REGISTER

Register	Address	R/W	Description	Reset Value
P1T_WTR	0x7E00103C	R/W	32-bit DRAM controller t_WTR register	0x2

PnT_WTR	Bit	Description	Initial State
	[31:3]	Read undefined. Write as Zero	
t_WTR	[2:0]	Set the write to read delay in memory clock cycles.	010

5.5.17 T_XP REGISTER

Register	Address	R/W	Description	Reset Value
P1T_XP	0x7E001040	R/W	32-bit DRAM controller t_XP register	0x01

PnT_XP	Bit	Description	Initial State
	[31:8]	Read undefined. Write as Zero	
t_XP	[7:0]	Set the exit power down command time in memory clock cycles.	0x01

5.5.18 T_XSR REGISTER

Register	Address	R/W	Description	Reset Value
P1T_XSR	0x7E001044	R/W	32-bit DRAM controller t_XSR register	0x0A

PnT_XSR	Bit	Description	Initial State
	[31:8]	Read undefined. Write as Zero	
t_XSR	[7:0]	Set the exit self refresh command time in memory clock cycles.	0x0A

5.5.19 T_ESR REGISTER

Register	Address	R/W	Description	Reset Value
P1T_ESR	0x7E001048	R/W	32-bit DRAM controller t_ESR register	0x14

PnT_ESR	Bit	Description	Initial State
	[31:8]	Read undefined. Write as Zero	
t_ESR	[7:0]	Set the self refresh command time in memory clock cycles.	0x14

5.5.20 MEMORY CONFIGURATION 2 REGISTER

Register	Address	R/W	Description	Reset Value
P1MEMCFG2	0x7E00104C	R/W	32-bit DRAM controller configuration register	0x0B45

PnMEMCFG2	Bit	Description	Initial State
Reserved	[31:13]	Read undefined. Write as Zero.	
Read delay	[12:11]	Encodes the delay used when reading from the pad interface to allow for de-skew of incoming read data 00 = Read delay 0 cycle (usually for SDR SDRAM. The SDR configuration requires read_dealy set to zero.) 01 = Read delay 1 cycle (usually for DDR SDRAM and mobile DDR SDRAM) 10, 11 = Read delay 2 cycle	01
Memory type	[10:8]	The type of SDRAM that is attached to DRAM controller: 000 = SDR SDRAM 001 = DDR SDRAM 011 = Mobile DDR SDRAM Note It is only legal to program the memory type between SDR and (LP)DDR for a memory controller configuration that supports it	011
Memory width	[7:6]	The width of the external memory 00 = 16-bit 01 = 32-bit 10 = Reserved 11 = Reserved Note Only a memory width that is legal for the memory controller can be programmed.	00 / 01
Reserved	[5:4]	Read undefined. Write as Zero.[1]	00
cke_init	[3]	Sets the level for the cke outputs after reset.	1/0
DQM init	[2]	Sets the level for the dqm outputs after reset.	1
a_gt_m_sync	[1]	Requires to be set HIGH when running the aclk and mclk synchronously but with aclk running faster than mclk.	0
sync	[0]	Set high when aclk and mclk are synchronous.	1

[1] bank_bits is not programmable in 6410X.

5.5.21 MEMORY CONFIGURATION 3 REGISTER

Register	Address	R/W	Description	Reset Value
P1MEMCFG3	0x7E001050	R/W	32-bit DRAM controller configuration register	0

PnMEMCFG3	Bit	Description	Initial State
Reserved	[31:12]	Read undefined. Write as Zero.	
Prescale	[11:3]	Prescalar counter value	0
max_outs_refs	[2:0]	Maximum number of outstanding refresh commands.	111

5.5.22 ID_N_CFG REGISTER

Register	Address	R/W	Description	Reset Value
P1_id_0_cfg ~P1_id_15_cfg	0x7E001100 ~0x7E00113C	R/W	32-bit DRAM controller id_<n>_cfg register	0x000

Pn_id_<n>_cfg	Bit	Description	Initial State
	[31:10]	Read undefined. Write as Zero	
QoS_MAX	[9:2]	Set a maximum quality of service.	0x00
QoS_MIN	[1]	Set a minimum quality of service.	0
QoS_Enable	[0]	Enables a quality of service value to be applied to memory reads from address ID <n>.	0

Table of the AXI bus master IDs

AXI ID	Master bus name	Related IPs
0000_0000	I Block	Camera, JPEG
0000_0001	F Block	Display Controller
0000_0010	P Block	TV Encoder, TV Scaler
XXXX_0011[2]	V Block	MFC
0000_0100	X Block	HSMMC, USB OTG
0000_0101	T Block	Host I/F
0000_0110	M Block	DMA0, DMA1
0000_0111	S Block	Security Sub Block, SDMA0, SDMA1
0000_1000	ARM Instruction	ARM Core Instruction
0000_1001	ARM Data	ARM Core Data
0000_1010	ARM DMA	ARM Core DMA
0000_1011	CF	CFCON
000X_1100[2]	G block	G3D
000X_1101[2]	G block	G3D
0000_1110	G2D	G2D

5.5.23 CHIP_N_CFG REGISTER

Register	Address	R/W	Description	Reset Value
P1_chip_0_cfg P1_chip_1_cfg	0x7E001200 0x7E001204	R/W	32-bit DRAM controller chip_<n>_cfg register	0x0FF00

Pn_chip_<n>_cfg	Bit	Description	Initial State
	[31:17]	Read undefined. Write as Zero	
BRC_RBC	[16]	Selects the memory organization as decoded from the AXI address: 0 = Row-Bank-Column organization. 1 = Bank-Row-Column organization.	0
Address match	[15:8]	Comparison value for AXI address bits [31:24] to determine which chip is selected.	0xFF
Address mask	[7:0]	The mask for AXI address bits [31:24] to determine which chip is selected: 1 = corresponding address bit is to be used for comparison	0x00

[2] X refers to ID that can be dynamically changed by bus master.

5.5.24 USER_STATUS REGISTER

Register	Address	R/W	Description	Reset Value
P1_user_stat	0x7E001300	R	32-bit DRAM controller user_stat register	0x00

Details are same as USER_CONFIG register, refer to 5.5.25.

5.5.25 USER_CONFIG REGISTER

Register	Address	R/W	Description	Reset Value
P1_user_cfg	0x7E001304	W	32-bit DRAM controller user_cfg register	0x00

P1_user_cfg	Bit	Description	Initial State
Reserved	[31:16]	Read undefined. Write as Zero	0
Reserved	[15]		0
DQS3 input chain delay selection	[14:12]	Sets DQS[3] input chain delay value. When it is set to 3'b000, minimum delay chain is used.	0
Reserved	[11]		0
DQS2 input chain delay selection	[10:8]	Sets DQS[2] input chain delay value. When it is set to 3'b000, minimum delay chain is used.	0
Reserved	[7]		0
DQS1 input chain delay selection	[6:4]	Sets DQS[1] input chain delay value. When it is set to 3'b000, minimum delay chain is used.	0
Reserved	[3]		0
DQS0 input chain delay selection	[2:0]	Sets DQS[0] input chain delay value. When it is set to 3'b000, minimum delay chain is used.	0

6 SROM CONTROLLER

6.1 OVERVIEW

The S3C6410 SROM Controller (SROMC) supports external 8, 16-bit NOR Flash, PROM, SRAM memory. S3C6410 SROM Controller supports 6-bank memory of maximum 128 MB size only.

6.2 FEATURE

S3C6410 SROM Controller features include:

- Supports SRAM, various ROMs and NOR flash memory
- Supports only 8 or 16-bit data bus
- Address space: Up to 128MB 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

6.3 BLOCK DIAGRAM

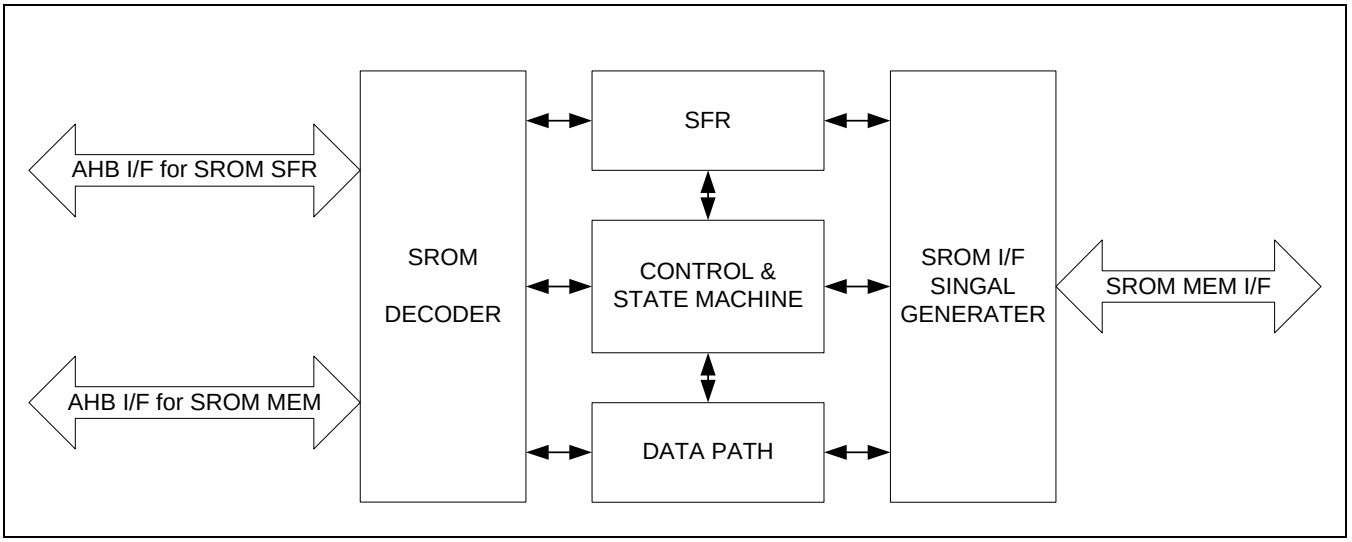


Figure 6-1. SROM Controller Block Diagram

6.4 SROM CONTROLLER FUNCTION DESCRIPTION

SROM Controller support SROM interface for Bank0 to Bank5. In case of OneNAND boot, SROM controller cannot control Bank2 and Bank3 because its mastership is on OneNAND Controller. In case of NAND boot, SROM controller cannot control Bank2 and Bank3 because its mastership is on NAND Flash Controller.

6.4.1 nWAIT PIN OPERATION

If the WAIT operation corresponding to each memory bank is enabled, the nOE duration will 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.

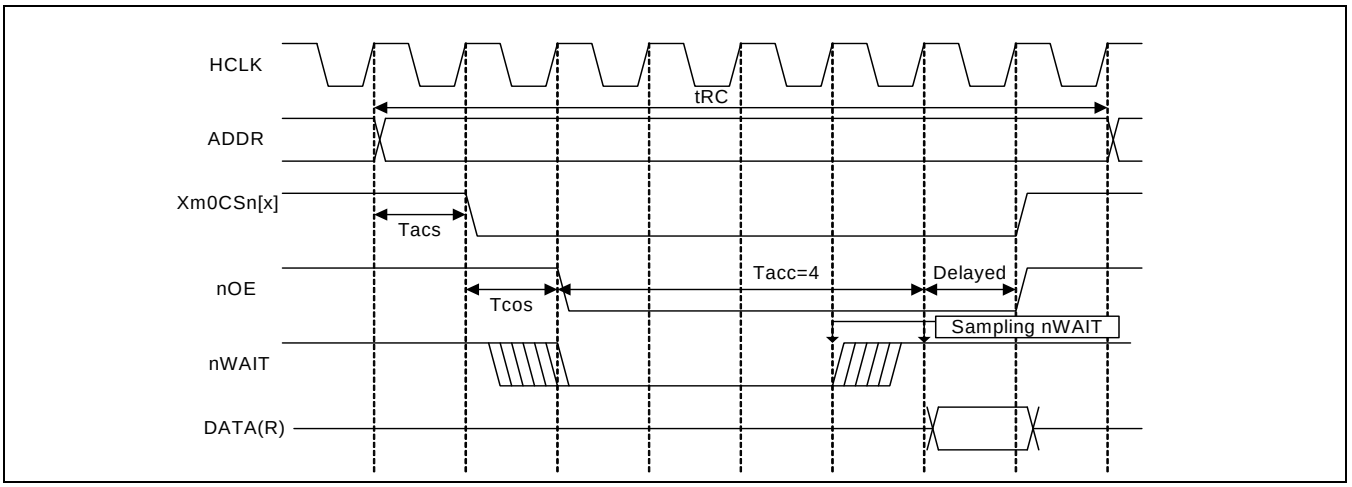


Figure 6-2. SROM Controller nWAIT Timing Block Diagram

6.5 PROGRAMMABLE ACCESS CYCLE

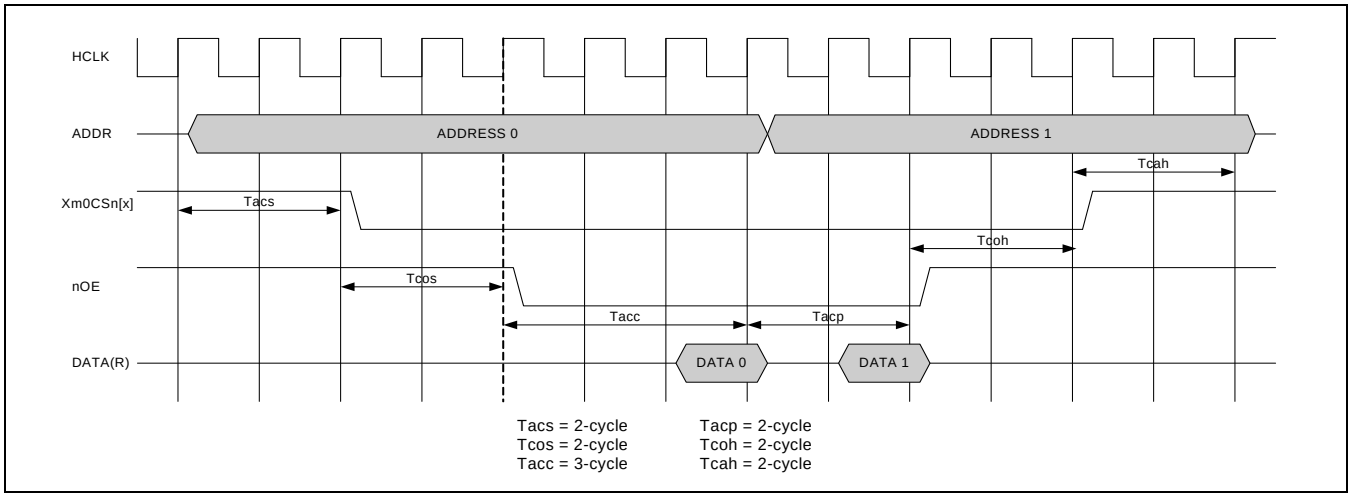


Figure 6-3. SROM Controller Read Timing Block Diagram

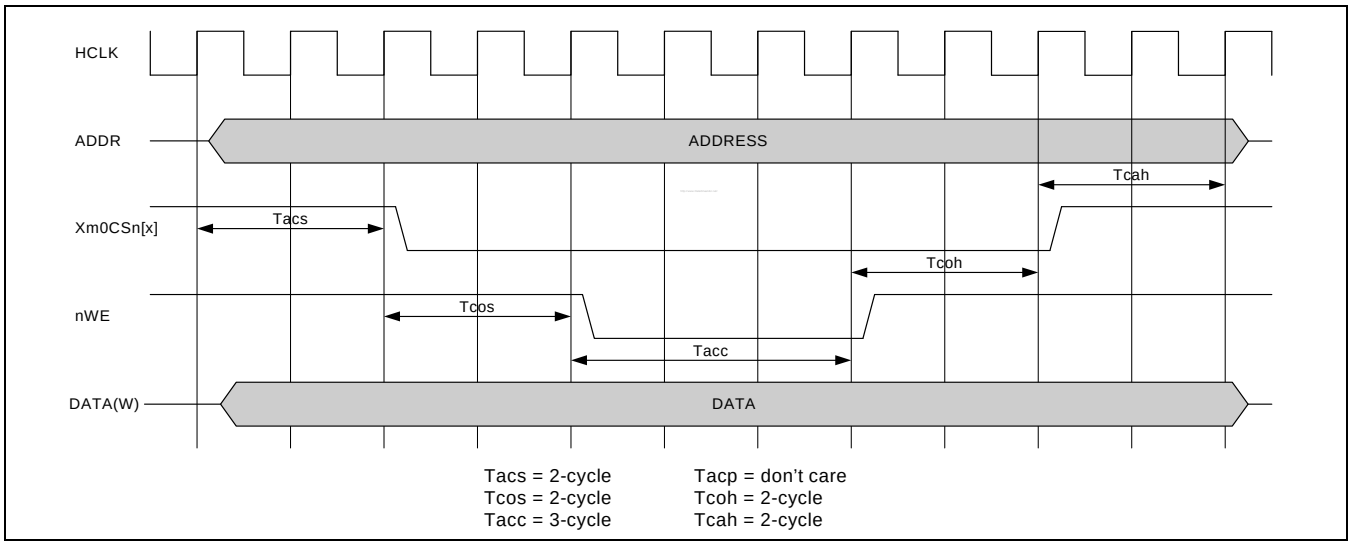


Figure 6-4. SROM Controller Write Timing Block Diagram

NOTE: Page mode is only supported on read cycle.

6.6 SPECIAL FUNCTION REGISTERS

6.6.1 SROM BUS WIDTH & WAIT CONTRL REGISTER(SROM_BW)

Register	Address	R/W	Description	Reset Value
SROM_BW	0x70000000	R/W	SROM Bus width & wait control	0x0000_000x

SROM_BW	Bit	Description	Initial State
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]	Wait enable control for Memory Bank5 0 = WAIT disable 1 = WAIT enable	0
Reserved	[21]	Reserved	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]	Wait enable control for Memory Bank4 0 = WAIT disable 1 = WAIT enable	0
Reserved	[17]	Reserved	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 = WAIT disable 1 = WAIT enable	0
Reserved	[13]	Reserved	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 = WAIT disable 1 = WAIT enable	0

SROM_BW	Bit	Description	Initial State
Reserved	[9]	Reserved	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 = WAIT disable 1 = WAIT enable	0
Reserved	[5]	Reserved	0
DataWidth1	[4]	Data bus width control for Memory Bank1 0 = 8-bit 1 = 16-bit	0
ByteEnable0	[3]	nWBE / nBE(for UB/LB) control for Memory Bank0 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
WaitEnable0	[2]	Wait enable control for Memory Bank0 0 = WAIT disable 1 = WAIT enable	0
Reserved	[1]	Reserved	0
DataWidth0	[0]	Data bus width control for Memory Bank0. Reset value is configured by OM setting. 0 = 8-bit 1 = 16-bit	H/W Set

6.6.2 SROM BANK CONTROL REGISTER (SROM_BC : XRCSN0 ~ XRCSN2)

Register	Address	R/W	Description	Reset Value
SROM_BC0	0x70000004	R/W	SROM Bank0 control register	0x000F_0000
SROM_BC1	0x70000008	R/W	SROM Bank1 control register	0x000F_0000
SROM_BC2	0x7000000C	R/W	SROM Bank2 control register	0x000F_0000
SROM_BC3	0x70000010	R/W	SROM Bank3 control register	0x000F_0000
SROM_BC4	0x70000014	R/W	SROM Bank4 control register	0x000F_0000
SROM_BC5	0x70000018	R/W	SROM Bank5 control register	0x000F_0000

SROM_BCn	Bit	Description	Initial State
Tacs	[31:28]	Adress set-up before Xm0CSn[x] 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
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

SROM_BCn	Bit	Description	Initial State
Tcah	[11:8]	Address holding time after Xm0CSn[x] 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
Tacp	[7:4]	Page mode access cycle @ Page mode 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	
PMC	[1:0]	Page mode configuration 00 = normal (1 data) 01 = 4 data 10 = Reserved 11 = Reserved	00

7 ONENAND CONTROLLER

This chapter describes the functions and usage of OneNAND controller in S3C6410X RISC microprocessor.

7.1 OVERVIEW

S3C6410X supports external 16-bit bus for both asynchronous and synchronous OneNAND external memory via shared memory port 0. It supports maximum 2 banks by using two controllers. The OneNAND Controller is an *Advanced Microcontroller Bus Architecture* (AMBA 2) compliant System-on-Chip peripheral. The OneNAND Controller provides simultaneous support for maximum two memory banks. Each memory bank supports only Muxed OneNAND. To use OneNAND Flash instead of NAND Flash, 'XSELNAND' pin must be connected to zero (Low level).

7.2 FEATURE

The OneNAND controller includes the following:

- Supports maximum 2 banks by using two OneNAND Controllers
- Supports asynchronous/synchronous muxed OneNAND memory
- Supports 16-bit wide external memory data paths
- Data buffering in order to achieve maximum performance
- Asynchronous FIFOs between the flash controller core and the bus system interface for speed matching
- Supports Erase commands through address mapping
- Supports write-synchronous mode if OneNAND device ID is 0x0040, 0x0048, and 0x0058.
- Supports write-synchronous mode if OneNAND device ID is 0x0030, 0x0034 and OneNAND version ID bit [9:8] is not 2'b00.
- Map11 command is used primarily for testing and debug of errors. (Except the special case)
- Supports eight AHB burst transfer with map01/10 command. This corresponds to 16 half-word burst transfer for OneNAND device.

7.3 BLOCK DIAGRAM

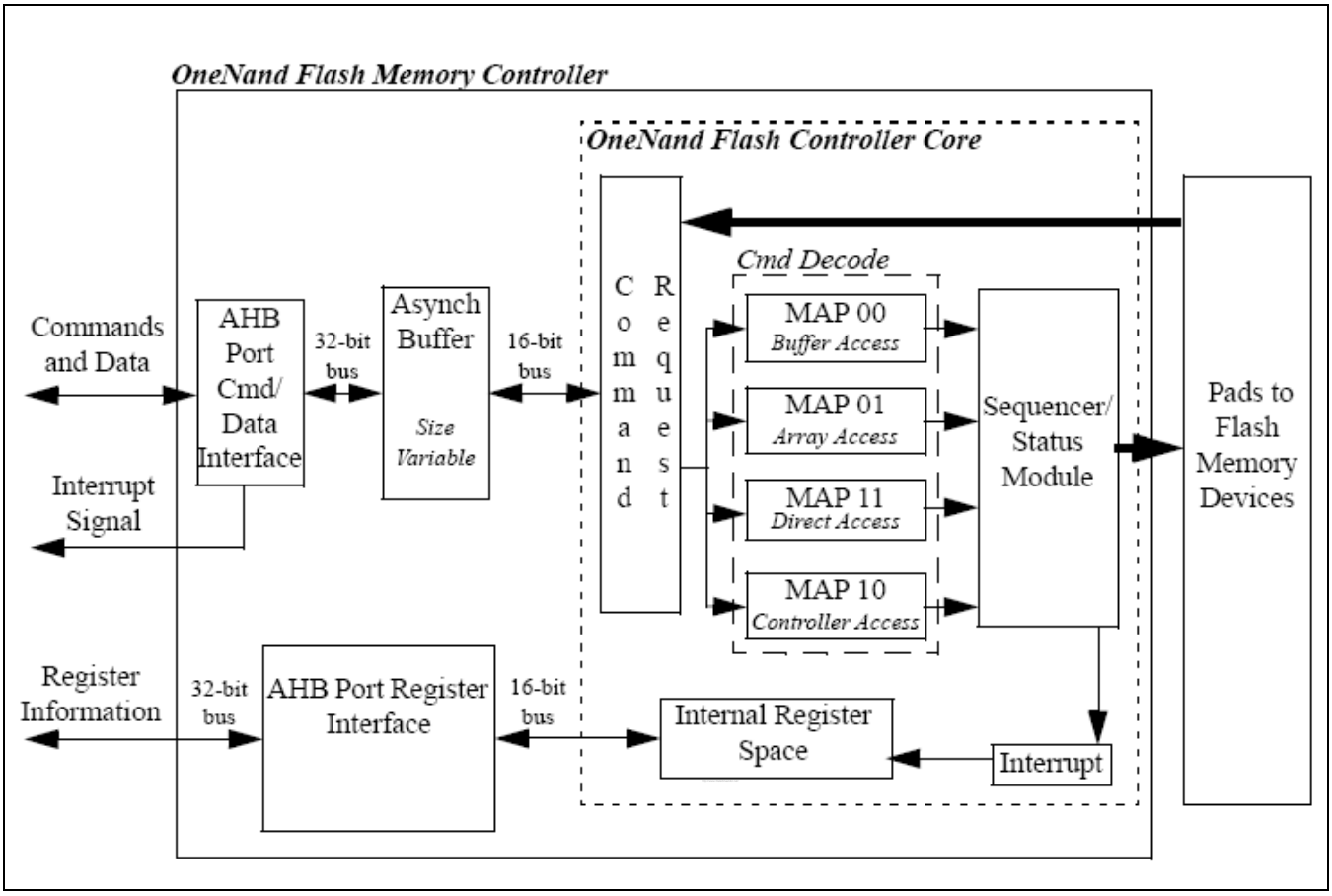


Figure 7-1. OneNAND Controller Block Diagram

7.4 SIGNAL DESCRIPTION

7.4.1 EXTERNAL MEMORY INTERFACE

Signal	I/O	Description
Xm0DATA [15:0]	IO	Xm0DATA[15:0] (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.
Xm0CSn[3:2]	O	Xm0CSn[3:2] (Chip Select) are activated when the address of a memory is within the address region of each bank. Xm0CSn[3:2] can be assigned to either SROMC or OneNAND controller by System Controller SFR setting. Active LOW.
Xm0WEn	O	Xm0WEn (Write Enable) indicates that the current bus cycle is a write cycle. Active LOW.
Xm0OEn	O	Xm0OEn (Output Enable) indicates that the current bus cycle is a read cycle. Active LOW.
Xm0INTsm0_FWEn Xm0INTsm1_FREn	I	Interrupt inputs from OneNAND memory Bank 0, 1. If OneNAND memory is not used, these signals must be tied to zero.
Xm0ADDRVALID	O	Address valid output. In the POP products, address and data are multiplexed. Xm0ADDRVALID indicate when the bus is used for address. Active LOW.
Xm0RPn_RnB	O	System reset output for OneNAND memory. Active LOW.
Xm0RDY0_ALE Xm0RDY1_CLE	I	Xm0RDY is a synchronous burst wait input that the external device uses to delay a synchronous burst transfer. Xm0RDY indicates data valid in synchronous read modes and is activated while Xm0CSn is low.
Xm0SMCLK	O	Static memory clock for synchronous static memory devices.

7.5 INPUT CLOCKS

The OneNAND controller has three clock source inputs. Bus system interface gets AHB bus clock, HCLK. Flash controller core gets two flash clocks, mclk and mclk_flash. Frequency of mclk must be double of mclk_flash, which is supplied to OneNAND flash memory.

NOTE: mclk : controller core clock (using HCLK)
mclk_flash : onenand interface clock(Xm0SMCLK)

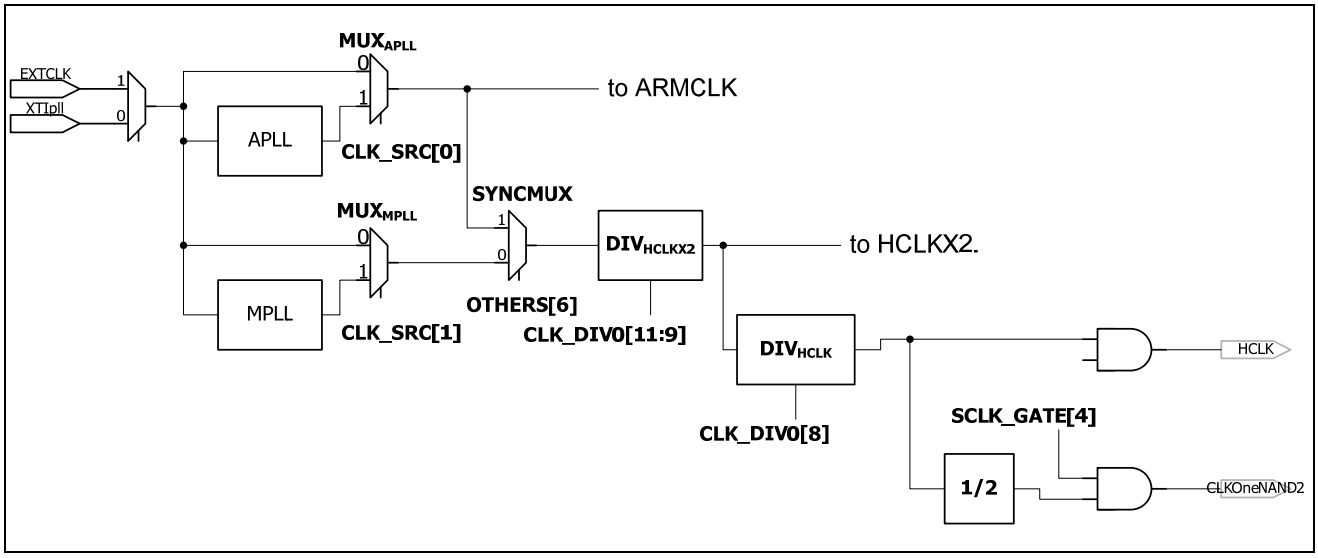


Figure 7-2. OneNAND Clock Generaion (Refer to System Controller Chapter)

7.6 MEMORY ADDRESS MAPPING

The OneNAND Controller reads the memory device's dev_id size field to determine the address map and automatically configures the MEM_ADDR field of the address map to support the device. Table 7-1 "MEM_ADDR Fields" defines the field sizes for several OneNAND memory devices.

Table 7-1. MEM_ADDR Fields

dev_id size field	Density	# of Blocks	Page Size	MAP location	MEM_ADDR Field					
					Reserved	DFS_DBS	FBA	FPA	FSA	Reserved
0000	128Mb	256	1KB	[25:24]	[23:20]	N/A	[19:12]	[11:6]	[4][1]	[3:0]
0001	256Mb	512	1KB	[25:24]	[23:21]	N/A	[20:12]	[11:6]	[4][2]	[3:0]
0010	512Mb	512	2KB	[25:24]	[23:21]	N/A	[20:12]	[11:6]	[5:4]	[3:0]
0011	1Gb Dual Die	1024	2KB	[25:24]	[23:22]	[21]	[20:12]	[11:6]	[5:4]	[3:0]
0011	1Gb	1024	2KB	[25:24]	[23]	N/A	[21:12]	[11:6]	[5:4]	[3:0]
0100	2Gb Dual Die	2048	2KB	[25:24]	[23]	[22]	[21:12]	[11:6]	[5:4]	[3:0]
0100	2Gb	2048	2KB	[25:24]	N/A	N/A	[22:12]	[11:6]	[5:4]	[3:0]
0101	4Gb Dual Die	4096	2KB	[25:24]	N/A	[23]	[22:12]	[11:6]	[5:4]	[3:0]
0101	4Gb	4096	2KB	[25:24]	N/A	N/A	[23:12]	[11:6]	[5:4]	[3:0]

7.7 COMMAND MAPPING

There are four kinds of commands supported by the OneNand flash memory controller. These commands are selected through the value of bits 23 and 22 of the incoming address. The command mapping determines the way in which the lower 22 bits of the address will be used.

7.7.1 "00" = MAP 00 COMMANDS

MAP 00 commands are used to access a controller-selected buffer in the flash memory device including boot, DataRAM 0 and DataRAM 1. The addressing will always begin at 0x0 and the memory controller will map that address to the appropriate buffer. The user will know the maximum address of the buffer through the BOOT_BUF_SIZE and DATA_BUF_SIZE registers.

[1] In case of dev_id of 0000, MEM_ADDR[5] is not used.

[2] In case of dev_id of 0001, MEM_ADDR[6] is not used.

In most cases, the buffer used will be the boot buffer, and the user will use this command to read data from this buffer. However, MAP 00 commands are also used in read/modify/write operations. When the read/modify/write command sequence is followed, map 00 commands may be used to read or write any word in the buffer. The memory controller will select DataRAM 0 or DataRAM 1 automatically. The selection is invisible to the command requestor.

Table 7-2. MAP 00 Address Mapping

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Address
25:24	CMD_MAP	00 = Read or Write of XIP buffer
23:17	RESV	Reserved space
16:1	MEM_ADDR	Buffer Address on the Memory Device
0	BYTE	Must be set to zero

7.7.2 "01" = MAP 01 COMMANDS.

MAP 01 commands are used for normal high-speed accesses to the memory array. The user can read and write data to a particular page of the array by specifying the FBA and FPA for the command. Since the OneNAND flash 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 transfer. Even if multiple commands are required, the same address must be used until the entire block has been transferred.

Each read or write data word will be 32-bit in size. The asynchronous buffers will separate the write data into two 16-bit words or concatenate the read data into one 32-bit word.

Table 7-3. MAP 01 Address Mapping

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Address
25:24	CMD_MAP	01 = Read or Write to the Memory Device
23:0	MEM_ADDR	Refer Table 7-1.

7.7.3 "10" = MAP 10 COMMANDS.

MAP 10 commands are used to control the special functions of the memory device. This is a command pathway in which the data passed is aimed at the memory controller and not the memory device. Unlike the other command types, the data (input or output) related to these transactions does not affect the contents of the memory, but is used to specify and perform the exact commands for the memory controller. The input and output data streams will always be 32 bits. However, only the lower 16 bits of the data path contain relevant information.

Table 7-4. MAP 10 Address Mapping

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Address
25:24	CMD_MAP	10 = Initiate a special function of the flash device or read the status of the memory controller
23:0	MEM_ADDR	Refer Table 7-1.

7.7.3.1 Erase Operation

The OneNand flash controller supports single block and multi-block erases. Multi-block erases are performed in parallel if the memory device supports this action. If memory device does not support this action, then the erases will happen in sequential order. When using the multi-block erase option, the user will specify the address of each block and then issue a single block erase command for the final block which initiates the entire erase.

The actual erase command used is determined by the type of transaction on the AHB bus (read or write) and the low byte of the data bus.

Address	Cmd Type	Data	Function
[25:24] = 10 DFS_DBS and FBA are used. FPA and FSA are unused for Erase Operations and must be cleared.	Write	0x00	Save the status of the current erase operation to the memory controller.
	Read	-	If the previous command was a MAP 10 Write 0x00, then this command returns the status of the erase. If the previous command was NOT a MAP 10 Write 0x00, then return a 0 if the erase has been completed. 0 = No Erase is in progress, or erase has been completed. 1 = Erase is in progress.
	Write	0x01	Save the block address for a multi-block erase. This does NOT initiate the erase.
	Write	0x03	Save the block address for a single-block erase and initiate the erase. Also used to specify the final block of a multi-block erase and initiate the multi-block erase.

7.7.3.2 Lock, Unlock and Lock-Tight Operations

The OneNAND flash controller supports all flash locking operations. However, the memory device may have limited support for these functions. If locking functions are not supported, all of these commands will be ignored. If the "unlock all" function is not supported, an interrupt will be triggered.

A memory area that has been locked can also be “locked-tight”. Once an area is "locked tight", it requires a reset in order to be unlocked.

The actual lock/unlock/lock-tight command used is determined by the type of transaction on the AHB bus (read or write) and the low byte of the data bus.

Address	Cmd Type	Data	Function
[25:24] = 10 DFS_DBS and FBA are used. FPA and FSA are unused for Lock Operations and must be cleared.	Write	0x08	Save this address as the start address for the unlock.
	Write	0x09	Save this address as the end address for unlock and initiate unlock.
	Write	0x0A	Save this address as the start address for the lock.
	Write	0x0B	Save this address as the end address for the lock and initiate the lock.
	Write	0x0C	Save this address as the start address for the lock-tight. Note that the memory controller will not verify that the specified block is locked before issuing the lock-tight command, but the lock-tight will only be successful on locked blocks.
	Write	0x0D	Save this address as the end address for the lock-tight and initiate the lock-tight. Note that the memory controller will not verify that the specified block is locked before issuing the lock-tight command, but the lock-tight will only be successful on locked blocks.
	Write	0x0E	Unlock the entire memory array. If this function is not supported, a command error interrupt will be generated.

7.7.3.3 Copy Back Operations

The OneNand flash controller supports copy operations. However, the memory device may have limited support for this function. If the copy back function is not supported, an interrupt will be triggered. An interrupt will also be triggered if the source block is not set before the destination block is specified, or if the destination block is not specified in the next command following a source block specification.

The parameter PP is used to set the number of pages for copy back. This is used to copy multiple consecutive pages in one command.

The actual copy back command used is determined by the type of transaction on the AHB bus (read or write) and the lower two bytes of the data bus.

Address	Cmd Type	Data	Function
[25:24] = 10	Write	0x1000	Save this address as the source address for the copy.
DFS_DBS, FBA, FPA, and FSA are used.	Write	0x20PP	Save this address as the destination address for the copy. Initiate a copy of PP pages from the source to destination addresses.

7.7.3.4 OTP and Spare Area Access Operations

There are three major sections of the flash array and the OneNAND flash controller supports access to all sections. However, the memory device may have limited support for these areas. If access to the OTP or spare areas is not supported, an interrupt will be triggered. Commands that are conflicting or that can not be executed in parallel will be held off until the conflicting command completes.

The actual access command used is determined by the type of transaction on the AHB bus (read or write) and the low byte of the data bus.

Address	Cmd Type	Data	Function
[25:24] = 10	Write	0x12	Configure the controller to access the OTP area of the flash.
DFS_DBS, FBA, FPA, and FSA are used.	Write	0x14	Configure the controller to access the main data areas of the flash. This is the default area, you must reset after accessing OTP and spare areas.

7.7.3.5 Verify Read Operations

Multi-block erases must be verified through a verify read command. This command operates serially and verifies one block at a time. Each block of the multi-block erase must be verified. An interrupt (Ers_Fail) will be triggered if the block does not verify.

The verify read command used is determined by the type of transaction on the AHB bus (read or write) and the low byte of the data bus.

Address	Cmd Type	Data	Function
[25:24] = 10 DFS_DBS and FBA are used. FPA and FSA are unused for Erase Verify Operations and must be cleared.	Write	0x15	Save the block address for an erase verification and initiate the verify command.

7.7.3.6 Pipeline Read-Ahead or Write-Ahead Operations

The OneNAND flash controller supports pipeline read-ahead and write-ahead operations. However, the memory device may have limited support for this function. If pipelined read-ahead or write-ahead is not supported, these commands will be ignored.

The pipeline read-ahead function allows for a continuous reading of the memory, even when the read request has not been issued to the controller. The pipeline read-ahead function utilizes a set of registers in the controller to hold the specified pages of data starting at a given address. The group of reads is accessed as a continuous series of pages. By pulling this data into the memory controller prior to the actual read request being issued, the memory controller is able to reduce latency on returning read data to the AHB interface.

The pipeline write-ahead function allows for the AHB interface to accept write data, even when the write request has not been issued to the controller. The pipeline write-ahead function utilizes a set of registers in the controller to hold the specified pages of data starting at a given address. The data is accepted as a continuous series of pages. By pulling this data into the memory controller prior to the actual write request being issued, the memory controller is able to reduce latency on writing the data to the flash.

Note that the buffers used for pipelined read-ahead and write-ahead commands are the same. Data that is pre-read must be read out to the AHB interface before proceeding. Data that is pre-written must be written to flash before proceeding.

Address	Cmd Type	Data	Function
[25:24] = 10 DFS_DBS, FBA, FPA, and FSA are used.	Write	0x4WPP	Save this address as the initial address for the read/write. For reads, "W"=0 and this will initiate a copy of PP pages from the memory to memory controller. For writes, "W"=1 and this will initiate acceptance of PP pages from the AHB interface to the memory controller.

NOTE: The Pipelined Read-Ahead or Write-Ahead for a single area must request at least 2 pages.

7.7.3.7 Read/Modify/Write Operations

The user may need to read a specific page or modify a few words, bytes or bits in a page. The read/modify/write operations are used for this purpose. A read command pulls the desired data from memory to a buffer. The user will then modify the information in the buffer and then issue another command to write that information back to memory.

The read/modify/write command used is determined by the type of transaction on the AHB bus (read or write) and the low byte of the data bus.

Address	Cmd Type	Data	Function
[25:24] = 10 DFS_DBS, FBA, FPA, and FSA are used.	Write	0x10	Load the page specified by the FBA, FPA and FSA to the map 00 XIP buffer.
	Write	0x11	Write the data in the map 00 XIP buffer to the page specified by the FBA, FPA and FSA.

7.7.4 "11" = MAP 11 COMMANDS.

MAP 11 commands are used for direct memory accesses to the memory device. This command type is used primarily for testing and debug of errors by allowing the user to communicate directly with the part to read or write data to a particular address or register. While this access type allows direct contact with all areas of the memory (BootRAM, DataRAM 0, DataRAM 1, spare, command address space, registers and OTP), it is slow and cumbersome and must only be used when required. Only lower half-word is used for data transfer with MAP 11 command.

Table 7-5. MAP 11 Address Mapping

Address Bits	Name	Description
31:26	AHB_int_add	AHB Port Address
25:24	CMD_MAP	11 = Read or Write of OneNAND
23:18	RESV	Reserved space
17:2	Address	Memory device address
1:0	RESV	Must be set to zero

7.8 PIPELINE READ/WRITE AHEAD COMMAND

When a pipeline read-ahead or write-ahead command is received, and the controller is idle, this command will initiate a load operation immediately into one of the DataRAM buffers of the memory device. Note that the read-ahead command does NOT return the data to the AHB interface, and the write-ahead command does NOT write data to the flash address. The read data is loaded and read data will be returned to the AHB port only when map 01 commands are issued to read this data. Similarly, the write data is loaded and will be written to the flash only when map 01 commands are issued to write this data.

On read-ahead commands, the controller internally manages the read operations required. The controller will issue cache reads to the Flash memory device for each page requested. If the TRANS_SPARE register is set, then the main data area and the spare area will both be transferred to the memory controller during a pipeline read-ahead request.

Pipeline read-ahead commands may be entered sequentially, and pipeline write-ahead commands may be entered sequentially. If all of the data has been read/written for the existing read-ahead/write-ahead command, a pipelined read-ahead command may be followed by a pipelined write-ahead command, and vice versa.

If a current group of reads or writes is in process, the memory controller will hold off the new command until that set of operations is completed. The OneNAND flash memory controller may have up to three read-ahead or write-ahead commands pending in addition to the one being executed. The controller will not return to the idle state until all read-ahead pipelined data has been read from the buffer or all write-ahead data has been written to memory though map 01 commands.

Pipelined read-ahead or write-ahead takes precedence over register programming operations.

7.8.1 SET UP A SINGLE AREA FOR PIPELINED READ-AHEAD

The procedure to set up an area for pipelined read-ahead is as follows:

1. The user must set the CMD_MAP to "MAP 10" and set the starting address of the block to pre-read in the FBA, FPA and FSA of the address.
2. The command type must be set to "Write" and the data bus must be driven with a value of 0x40PP, where the "0" sets this command as a read-ahead and PP is the number of pages to pre-read.
3. To read the data, you must issue MAP 01 commands to the memory controller with the same starting address and the desired page(s). If the read command received following a pipeline read-ahead request is not to a page pre-read, then an interrupt bit will be set and the pipeline read/write-ahead registers will be cleared. A new pipeline read-ahead request must be issued to re-load the same data.
4. All of the data that was pre-read must be read through MAP 01 commands before the controller will return to the idle state.

NOTE: A pipelined read-ahead or write-ahead for a single area must request at least 2 pages.

7.8.2 SET UP MULTIPLE AREAS FOR PIPELINED READ-AHEAD

The procedure to set up multiple areas for pipelined read-ahead is as follows:

1. You must set the CMD_MAP to "MAP 10" and set the starting address of the first block to pre-read in the FBA, FPA and FSA of the address.
2. The command type must be set to "Write" and the data bus must be driven with a value of 0x40PP, where the "0" sets this command as a read-ahead and PP is the number of pages to pre-read.
3. If the user wishes to have any data returned on the AHB interface, MAP 01 commands must be issued to the memory controller with the same starting address and the desired page(s). These read commands must be to an address within the pre-read pages. The user is NOT required to read all of the data, or even any of the data, before issuing another pipelined read-ahead.
4. The user must set the CMD_MAP to "MAP 10" and set the starting address of the second block to pre-read in the FBA, FPA and FSA of the address.
5. The command type must be set to “Write” and the data bus must be driven with a value of 0x40PP, where the “0” sets this command as a read-ahead and PP is the number of pages to pre-read.
6. To read any data from the first or the second pre-read page, you must issue MAP 01 commands to any addresses pre-read.
7. To pre-read additional blocks, follow steps 4 to 6. The MAP 01 reads can be interleaved with pipelined read-ahead or pipelined write-ahead commands with a maximum of three pending requests (of either type) other than the one being processed. However, the OneNAND flash memory controller will not return to idle until all of the pre-read data has been read through the AHB interface.

7.9 CONTROLLER USAGE EXCEPTION

The OneNAND controller is designed with following exceptions:

1. The OneNAND Flash Memory Controller does not enforce any timing restrictions around the use of the reset register.
2. The burst length value of OneNAND controller must be programmed with the same to or less than the burst length value of the OneNAND flash device.
3. AMBA transactions must 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.
4. Data access must use a transfer size of a WORD.
5. Main & Spare areas must be accessed using the same burst length.
6. When OneNAND controller is used, IOBE field should be enabled.
7. The MAP00 command must not be issued while 1-page data is transferred using the MAP01 commands. After OneNAND booting, it is recommended that Stepping-Stone area is remapped to 0x00 through MMU virtual address setting or use the high vectors configuration mode.

7.10 BOOT WITH ONENAND CONTROLLER

The OneNAND controller supports OneNAND boot with following steps:

1. External reset is de-asserted.
2. System controller generates global reset.
3. OneNAND controller starts counting "FLASH_COLD_RST_DELAY" with core clock.
4. After ARM Core reset is de-asserted, Instruction fetch is started, but this fetch is suspended by OneNAND controller.
5. When the count value reaches to predefined FLASH_COLD_RST_DELAY value (= 534*ACCESS_CLK cycles), OneNAND controller start reading memory-dependent information.

   A. Memory-dependent information consists of Manufacturer ID, Device ID, Version ID, Data Buffer Size, Boot Buffer Size, Number of Buffer, and Technology.

   B. While sending first information-read request, OneNAND controller de-asserts Xm0RPn.
6. After reading memory-dependent information, OneNAND controller starts servicing data transaction requests.

CAUTION: To read the OneNAND array using the MAP01 commands on booting, the code must not be run in the BootRAM of the OneNAND Device. After the basic system setting, it is recommended that 1K Bytes of the BootRAM are copied to Stepping-Stone area. Then, jump to stepping stone area and proceed transferring the data of the OneNAND array to SDRAM.

7.11 REGISTER DESCRIPTION

7.11.1 MEMORY MAP

Register	Address	R/W	Description	Reset Value
MEM_CFG0	0x70100000	R/W	Bank0 Memory Device Configuration Register	0x0000
BURST_LEN0	0x70100010	R/W	Bank0 Burst Length Register	0x0000
MEM_RESET0	0x70100020	R/W	Bank0 Memory Reset Register	0x0000
INT_ERR_STAT0	0x70100030	R/W	Bank0 Interrupt Error Status Register	0x0000
INT_ERR_MASK0	0x70100040	R/W	Bank0 Interrupt Error Mask Register	0x0000
INT_ERR_ACK0	0x70100050	W	Bank0 Interrupt Error Acknowledge Register	0x0000
ECC_ERR_STAT0	0x70100060	R/W	Bank0 ECC Error Status Register	0x0000
MANUFACT_ID0	0x70100070	R	Bank0 Manufacturer ID Register	Mem.dep.
DEVICE_ID0	0x70100080	R	Bank0 Device ID Register	Mem.dep.
DATA_BUF_SIZE0	0x70100090	R	Bank0 Data Buffer Size Register	Mem.dep.
BOOT_BUF_SIZE0	0x701000A0	R	Bank0 Boot Buffer Size Register	Mem.dep.
BUF_AMOUNT0	0x701000B0	R	Bank0 Amount of Buffer Register	Mem.dep.
TECH0	0x701000C0	R	Bank0 Technology Register	Mem.dep.
FBA_WIDTH0	0x701000D0	R/W	Bank0 FBA Width Register	0x000A
FPA_WIDTH0	0x701000E0	R/W	Bank0 FPA Width Register	0x0006
FSA_WIDTH0	0x701000F0	R/W	Bank0 FSA Width Register	0x0002
REVISION0	0x70100100	R	Bank0 Revision Register	0x0002
DATARAM00	0x70100110	R/W	Bank0 DataRAM0 Code Register	0x0002
DATARAM10	0x70100120	R/W	Bank0 DataRAM1 Code Register	0x0003
SYNC_MODE0	0x70100130	R	Bank0 Synchronous Mode Register	0x0000
TRANS_SPARE0	0x70100140	R/W	Bank0 Transfer Size Register	0x0000
Reserved	0x70100150	-	Reserved	-
DBS_DFS_WIDTH0	0x70100160	R/W	Bank0 DBS_DFS width Register	0x0000
PAGE_CNT0	0x70100170	R	Bank0 Page Count Register	0x0000
ERR_PAGE_ADDR0	0x70100180	R	Bank0 Error Page Address Register	0x0000
BURST_RD_LAT0	0x70100190	R	Bank0 Burst Read Latency Register	0x0006
INT_PIN_ENABLE0	0x701001A0	R/W	Bank0 Interrupt Pin Enable Register	0x0000
INT_MON_CYC0	0x701001B0	R/W	Bank0 Interrupt Monitor Cycle Count Register	0x01F4
ACC_CLOCK0	0x701001C0	R/W	Bank0 Access Clock Register	0x0003
SLOW_RD_PATH0	0x701001D0	R/W	Bank0 Slow Read Path Register	0x0000
ERR_BLK_ADDR0	0x701001E0	R	Bank0 Error Block Address Register	0x0000
FLASH_VER_ID0	0x701001F0	R	Bank0 Flash Version ID Register	Mem.dep.
FLASH_AUX_CNTRL0	0x70100300	RW	Bank0 Flash Auxiliary control register	0x0000

FLASH_AFIFO_CNT 0	0x70100310	R	Number of data in asynchronous FIFO in flash controller 0.	0x0000

Register	Address	R/W	Description	Reset Value
MEM_CFG1	0x70180000	R/W	Bank1 Memory Device Configuration Register	0x0000
BURST_LEN1	0x70180010	R/W	Bank1 Burst Length Register	0x0000
MEM_RESET1	0x70180020	R/W	Bank1 Memory Reset Register	0x0000
INT_ERR_STAT1	0x70180030	R/W	Bank1 Interrupt Error Status Register	0x0000
INT_ERR_MASK1	0x70180040	R/W	Bank1 Interrupt Error Mask Register	0x0000
INT_ERR_ACK1	0x70180050	R/W	Bank1 Interrupt Error Acknowledge Register	0x0000
ECC_ERR_STAT1	0x70180060	W	Bank1 ECC Error Status Register	0x0000
MANUFACT_ID1	0x70180070	R	Bank1 Manufacturer ID Register	Mem.dep.
DEVICE_ID1	0x70180080	R	Bank1 Device ID Register	Mem.dep.
DATA_BUF_SIZE1	0x70180090	R	Bank1 Data Buffer Size Register	Mem.dep.
BOOT_BUF_SIZE1	0x701800A0	R	Bank1 Boot Buffer Size Register	Mem.dep.
BUF_AMOUNT1	0x701800B0	R	Bank1 Amount of Buffer Register	Mem.dep.
TECH1	0x701800C0	R	Bank1 Technology Register	Mem.dep.
FBA_WIDTH1	0x701800D0	R/W	Bank1 FBA Width Register	0x000A
FPA_WIDTH1	0x701800E0	R/W	Bank1 FPA Width Register	0x0006
FSA_WIDTH1	0x701800F0	R/W	Bank1 FSA Width Register	0x0002
REVISION1	0x70180100	R	Bank1 Revision Register	0x0002
DATARAM01	0x70180110	R/W	Bank1 DataRAM0 Code Register	0x0002
DATARAM11	0x70180120	R/W	Bank1 DataRAM1 Code Register	0x0003
SYNC_MODE1	0x70180130	R	Bank1 Synchronous Mode Register	0x0000
TRANS_SPARE1	0x70180140	R/W	Bank1 Transfer Size Register	0x0000
Reserved	0x70180150	-	Reserved	-
DBS_DFS_WIDTH1	0x70180160	R/W	Bank1 DBS_DFS width Register	0x0000
PAGE_CNT1	0x70180170	R	Bank1 Page Count Register	0x0000
ERR_PAGE_ADDR1	0x70180180	R	Bank1 Error Page Address Register	0x0000
BURST_RD_LAT1	0x70180190	R	Bank1 Burst Read Latency Register	0x0006
INT_PIN_ENABLE1	0x701801A0	R/W	Bank1 Interrupt Pin Enable Register	0x0000
INT_MON_CYC1	0x701801B0	R/W	Bank1 Interrupt Monitor Cycle Count Register	0x01F4
ACC_CLOCK1	0x701801C0	R/W	Bank1 Access Clock Register	0x0003
SLOW_RD_PATH1	0x701801D0	R/W	Bank1 Slow Read Path Register	0x0000
ERR_BLK_ADDR1	0x701801E0	R	Bank1 Error Block Address Register	0x0000
FLASH_VER_ID1	0x701801F0	R	Bank1 Flash Version ID Register	Mem.dep.
FLASH_AUX_CNTRL0	0x70100300	RW	Bank1 Flash Auxiliary control register	0x0000

FLASH_AFIFO_CNT0	0x70100310	R	Number of data in asynchronous FIFO in flash controller 1.	0x0000

NOTE: Mem.dep. : Memory dependent

7.12 INDIVIDUAL REGISTER DESCRIPTIONS

7.12.1 MEMORY DEVICE CONFIGURATION REGISTER

The value programmed will depend on the actual memory device being used. This value is related with system configuration register 1(0xF221) of OneNAND Flash. This data is used to configure the flash for the hardware and software environment and may include burst length, read latency, transfer mode, ECC configuration, polarity levels, etc. This register is set by software during initialization, and when there is no OneNAND memory access. Please refer to appropriate model manual.

Register	Address	R/W	Description	Reset Value
MEM_CFG0 MEM_CFG1	0x70100000 0x70180000	R/W	Memory Device Configuration Register	0x00000000

MEM_CFGn	Bit	Description	Initial State
Reserved	[31:16]		
RM	[15]	Sets the transfer mode for read operations as synchronous or asynchronous. Default value is 0x0. Set by software during initialization. • 0 = Asynchronous Mode. • 1 = Activate Synchronous Mode.	0
BURST_RD_LAT	[14:12]	Sets the burst read latency in cycles. • 000~010 = Reserved • 011 = 3 (up to 40MHz) • 100 = 4 • 101 = 5 • 110 = 6 • 111 = 7	000
BURST_LENGTH	[11:9]	Sets the burst length. • 000 = Continuous • 001 = 4 half-words • 010 = 8 half-words • 011 = 16 half-words (Recommended) • 100 = 32 half-words • 101 = 1K half-words (Block Read Only) • 110~111 = Reserved	0
ECC	[8]	Error Correction Operation. • 0 = With correction. • 1 = without correction (by-passed).	0
RDYPOL	[7]	RDY signal polarity. • 0 = Low for ready. • 1 = High for ready.	0

MEM_CFGn	Bit	Description	Initial State
INTPOL	[6]	INT signal polarity. • 0 = Low for interrupt pending. • 1 = High for interrupt pending.	0
IOBE	[5]	I/O Buffer enable for INT and RDY signals, INT and RDY outputs are hi-Z at start up, bit 7 and 6 become valid after IOBE is set to 1. IOBE can be reset only by Cold reset. • Should be 1	0
RDYCONF	[4]	RDY configuration. • 0 = Active with valid data. • 1 = Active one clock before valid data.	0
Reserved	[3]		
HF	[2]	High frequency enable. • 0 = High frequency disable. (66MHz and under) • 1 =High frequency enable. (over 66MHz)	0
WM	[1]	Sets the transfer mode for write operations as synchronous or asynchronous. Default value is 0x0. Set by software during initialization. • 0 = Asynchronous Mode. • 1 = Activate Synchronous Mode.	0
BWPS	[0]	Boot buffer write protected status • 0 = Locked. • 1 = Unlocked.	0

NOTE: While OneNAND Controller does not obtain memory bus, updating MEM_CFG is stalled, and if another write access to MEM_CFG, it is possible to be lost. After writing MEM_CFG register, read system configuration register (0xF221) on OneNAND by using MAP11 command to check that register updated.

7.12.2 BURST LENGTH REGISTER

Register	Address	R/W	Description	Reset Value
BURST_LEN0 BURST_LEN1	0x70100010 0x70180010	R/W	Burst Length Register	0x0000

BURST_LENn	Bit	Description	Initial State
Reserved	[31:16]		0
BURST_LENGTH	[5:0]	Sets the burst length (half-word count) of the controller. This register has no default value. The value should be the same to or less than the value of OneNAND 0xF221 burst length field is recommended. Set by software during initialization. 4: 4 half-words. 8: 8 half-words 16: 16 half-words. Others: Reserved	0

7.12.3 MEMORY RESET REGISTER

Register	Address	R/W	Description	Reset Value
MEM_RESET0 MEM_RESET1	0x70100020 0x70180020	R/W	Memory Reset Register	0x0000

MEM_RESETn	Bit	Description	Initial State
Reserved	[31:3]		0
RESET_CODE	[2:0]	Sets the reset code. This register will reset to 0x0 after the reset sequence has been completed. This register is controlled through software. • 001 = Warm Reset. • 010 = Core Reset. • 011 = Hot Reset. • All other settings Reserved.	0

7.12.4 INTERRUPT ERROR STATUS REGISTER

Register	Address	R/W	Description	Reset Value
INT_ERR_STAT0 INT_ERR_STAT1	0x70100030 0x70180030	R/W	Interrupt Error Status Register	0x0000

INT_ERR_STATn	Bit	Description	Initial State
Reserved	[31:14]		0
CACHE_OP_ERR	[13]	An error occurred during a cache read or write setup or operation.	0
RST_CMP	[12]	The controller has completed its reset and initialization process. Be sure to check whether this bit is one before OneNAND access is executed.	0
RDY_ACT	[11]	The memory device's RDY pin is actively transitioning.	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 when an invalid command is received, or when 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]	The block read or writes transfer has been completed. This interrupt relates to "verify read" and "pipeline read-ahead" commands.	0
ERS_CMP	[6]	The erase operation has been completed. This interrupt is automatically reset at the beginning of an erase operation.	0
PGM_CMP	[5]	The program operation has been completed. This interrupt is automatically reset at the beginning of a program operation.	0
LOAD_CMP	[4]	The load operation has been completed.	0
ERS_FAIL	[3]	The erase operation was unsuccessful.	0
PGM_FAIL	[2]	The program operation was unsuccessful.	0
INT_TO	[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

NOTE: When OneNAND Controller receives cold reset, it waits until the predefined time as COLD reset wait time passes. After COLD reset wait cycles, it automatically generates read transactions to obtain memory dependent information from OneNAND memory. After read transactions are done, RST_CMP bit becomes one.

7.12.5 INTERRUPT ERROR MASK REGISTER

Register	Address	R/W	Description	Reset Value
INT_ERR_MASK0 INT_ERR_MASK1	0x70100040 0x70180040	R/W	Interrupt Error Mask Register	0x0000

INT_ERR_MASKn	Bit	Description	Initial State
Reserved	[31:14]		0
CACHE_OP_ERR	[13]	When a bit is set to zero, mask the interrupt correspond to the bit in the INT_ERR_STAT register. When a bit is set to one, allow the interrupt correspond to the bit in the INT_ERR_STAT register. Setting this bit to one allows the controller to issue this type of interrupt. Set through software.	0
RST_CMP	[12]		
RDY_ACT	[11]		
INT_ACT	[10]		
UNSUP_CMD	[9]		
LOCKED_BLK	[8]		
BLK_RW_CMP	[7]		
ERS_CMP	[6]		
PGM_CMP	[5]		
LOAD_CMP	[4]		
ERS_FAIL	[3]		
PGM_FAIL	[2]		
INT_TO	[1]		
LD_FAIL_ECC_ERR	[0]		

7.12.6 INTERRUPT ERROR ACKNOWLEDGE REGISTER

Register	Address	R/W	Description	Reset Value
INT_ERR_ACK0 INT_ERR_ACK1	0x70100050 0x70180050	W	Interrupt Error Acknowledge Register	0x0000

INT_ERR_ACKn	Bit	Description	Initial State
Reserved	[31:14]		0
CACHE_OP_ERR	[13]	Acknowledge bits that correspond to the bits in the INT_ERR_STAT register. Setting this bit resets or acknowledges the associated interrupt. Set by software.	0
RST_CMP	[12]		
RDY_ACT	[11]		
INT_ACT	[10]		
UNSUP_CMD	[9]		
LOCKED_BLK	[8]		
BLK_RW_CMP	[7]		
ERS_CMP	[6]		
PGM_CMP	[5]		
LOAD_CMP	[4]		
ERS_FAIL	[3]		
PGM_FAIL	[2]		
INT_TO	[1]		
LD_FAIL_ECC_ERR	[0]		

7.12.7 ECC ERROR STATUS REGISTER

Register	Address	R/W	Description	Reset Value
ECC_ERR_STAT0 ECC_ERR_STAT1	0x70100060 0x70180060	R/W	ECC Error Status Register	0x0000

ECC_ERR_STATn	Bit	Description	Initial State
Reserved	[31:16]		0
ECC_ERR_STAT	[15:0]	The value programmed will depend on the actual memory device being used. This data is used to report ecc error information.	0

7.12.8 MANUFACTURER ID REGISTER

Register	Address	R/W	Description	Reset Value
MANUFACT_ID0 MANUFACT_ID1	0x70100070 0x70180070	R	Manufacturer ID Register	Memory dependent

MANUFACT_IDn	Bit	Description	Initial State
Reserved	[31:16]		
MANUFACT_ID	[15:0]	The value programmed will depend on the actual memory device being used. This register is set by the flash controller after reset. Read-Only.	

7.12.8 DEVICE ID REGISTER

Register	Address	R/W	Description	Reset Value
DEVICE_ID0 DEVICE_ID1	0x70100080 0x70180080	R	Device ID Register	Memory dependent

DEVICE_IDn	Bit	Description	Initial State
Reserved	[31:16]		
DEVICE_ID	[15:9]	The value programmed will depend on the actual memory device being used. This register is set by the flash controller after reset. Read-Only.	
BOOT_INFO	[8]	Boot information. Read-Only. 0 = Bottom Boot 1 = reserved.	
DENSITY	[7:4]	Density of OneNAND. Read-Only. 0000 = 128 Mb. 0001 = 256 Mb. 0010 = 512 Mb. 0011 = 1 Gb. 0100 = 2 Gb. 0101 = 4 Gb.	
DDP	[3]	Data Path. Read-Only. 0 = Single Data Path. 1 = Dual Data Path	
MUXED	[2]	Read-Only. 0 = Muxed. 1 = Demuxed.	
VCC	[1:0]	Read-Only. 00 = 1.8V 01/10/11 = reserved.	

7.12.10 DATA BUFFER SIZE REGISTER

Register	Address	R/W	Description	Reset Value
DATA_BUF_SIZE0 DATA_BUF_SIZE1	0x70100090 0x70180090	R	Data Buffer Size Register	Memory dependent

DATA_BUF_SIZEn	Bit	Description	Initial State
Reserved	[31:16]		
DATA_BUF	[15:0]	The value programmed will depend on the actual memory device being used. This register is set by the flash controller after reset. Read-Only.	

7.12.11 BOOT BUFFER SIZE REGISTER

Register	Address	R/W	Description	Reset Value
BOOT_BUF_SIZE0 BOOT_BUF_SIZE1	0x701000A0 0x701800A0	R	Boot Buffer Size Register	Memory dependent

BOOT_BUF_SIZEn	Bit	Description	Initial State
Reserved	[31:16]		
BOOT_BUF	[15:0]	The value programmed will depend on the actual memory device being used. This register is set by the flash controller after reset. Read-Only.	

7.12.12 AMOUNT OF BUFFER REGISTER

Register	Address	R/W	Description	Reset Value
BUF_AMOUNT0 BUF_AMOUNT1	0x701000B0 0x701800B0	R	Amount of Buffer Register	Memory dependent

BUF_AMOUNTn	Bit	Description	Initial State
Reserved	[31:16]		
AMOUNT	[15:0]	The value programmed will depend on the actual memory device being used. This register is set by the flash controller after reset. Read-Only.	

7.12.13 TECHNOLOGY REGISTER

Register	Address	R/W	Description	Reset Value
TECH0 TECH1	0x701000C0 0x701800C0	R	Technology Register	Memory dependent

TECHn	Bit	Description	Initial State
Reserved	[31:16]		
TECHNOLOGY	[15:0]	The value programmed will depend on the actual memory device being used. This register is set by the flash controller after reset. Read-Only.	

7.12.14 FBA WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
FBA_WIDTH0 FBA_WIDTH1	0x701000D0 0x701800D0	R/W	FBA Width Register	0x000A

FBA_WIDTHn	Bit	Description	Initial State
Reserved	[31:5]		0
FBA	[4:0]	Sets the number of bits that will be used to represent the number of blocks. The default value is 0x0A. Set by software during initialization.	0x0A

7.12.15 FPA WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
FPA_WIDTH0 FPA_WIDTH1	0x701000E0 0x701800E0	R/W	FPA Width Register	0x0006

FPA_WIDTHn	Bit	Description	Initial State
Reserved	[31:5]		0
FPA	[4:0]	Sets the number of bits that will be used to represent the number of pages. The default value is 6. Set by software during initialization.	0x06

7.12.16 FSA WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
FSA_WIDTH0 FSA_WIDTH1	0x701000F0 0x701800F0	R/W	FSA Width Register	0x0002

FSA_WIDTHn	Bit	Description	Initial State
Reserved	[31:3]		0
FSA	[2:0]	Sets the number of bits that will be used to represent the number of sectors. The default value is 0x2. Set by software during initialization.	0x2

7.12.17 REVISION REGISTER

Register	Address	R/W	Description	Reset Value
REVISION0 REVISION1	0x70100100 0x70180100	R	Revision Register	0x00000002

REVISIONn	Bit	Description	Initial State
Reserved	[31:16]		0x0000
REVISION	[15:0]	Holds the controller revision number. Default value is 0x1. Read-Only.	0x0002

7.12.18 DATARAM0 CODE REGISTER

Register	Address	R/W	Description	Reset Value
DATARAM00 DATARAM01	0x70100110 0x70180110	R/W	DataRAM0 Code Register	0x0002

DATARAM0n	Bit	Description	Initial State
Reserved	[31:16]		0
DATARAM0	[3:0]	Sets the non-sector part of the data for ram0. Default value is 0x2. Set by software during initialization.	2

7.12.19 DATARAM1 CODE REGISTER

Register	Address	R/W	Description	Reset Value
DATARAM10 DATARAM11	0x70100120 0x70180120	R/W	DataRAM1 Code Register	0x0003

DATARAM1n	Bit	Description	Initial State
Reserved	[31:4]		0
DATARAM1	[3:0]	Sets the non-sector part of the data for ram1. Default value is 0x3. Set by software during initialization.	3

7.12.20 SYNCHRONOUS MODE REGISTER

Register	Address	R/W	Description	Reset Value
SYNC_MODE0 SYNC_MODE1	0x70100130 0x70180130	R	Synchronous Mode Register	0x0000

SYNC_MODEn	Bit	Description	Initial State
Reserved	[31:1]		0
RM	[1]	Sets the transfer mode for read operations as synchronous or asynchronous. Default value is 0x0. Set by software during initialization. This value is copied from the MEM_CFG 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 MEM_CFG register[1]. Read-Only. • 0 = Asynchronous Mode. • 1 = Activate Synchronous Mode.	0

7.12.21 TRANSFER SPARE REGISTER

Register	Address	R/W	Description	Reset Value
TRANS_SPARE0 TRANS_SPARE1	0x70100140 0x70180140	R/W	Transfer Size Register	0x0000

TRANS_SPAREn	Bit	Description	Initial State
Reserved	[31:1]		0
TSRF	[0]	On all read or write commands through map 01, if this bit is set, the data in the spare area of memory will be transferred to the asynchronous FIFO of the memory controller along with the main data. Size of the spare area is part dependent depending on the # of sectors. • 0 =Transfer data only. • 1 = Increase sector size. The main data area for the page will be transferred first and then the spare area.	0

7.12.22 DBS-DFS WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
DBS_DFS_WIDTH0 DBS_DFS_WIDTH1	0x70100160 0x70180160	R/W	DBS_DFS width Register	0x0000

DBS_DFS_WIDTHn	Bit	Description	Initial State
Reserved	[31:2]		0
WIDTH	[1:0]	Sets the DBS and DFS width. The default value is 0. Set by software during initialization. Ignored if not relevant.	0

7.12.23 PAGE COUNT REGISTER

Register	Address	R/W	Description	Reset Value
PAGE_CNT0 PAGE_CNT1	0x70100170 0x70180170	R	Page Count Register	0x0000

PAGE_CNTn	Bit	Description	Initial State
Reserved	[31:8]		0
PAGE_COUNT	[7:0]	Holds the copy-backed page count by multi-page copy-back command currently being executed. Read-Only.	0

7.12.24 ERROR PAGE ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
ERR_PAGE_ADDR0 ERR_PAGE_ADDR1	0x70100180 0x70180180	R	Error Page Address Register	0x0000

ERR_PAGE_ADDRn	Bit	Description	Initial State
Reserved	[31:6]		0
FAIL_PAGE_ADDR	[5:0]	After a program, load or erase error interrupt, this register will hold the page address of the failing operation. Read-Only.	0

7.12.25 BURST READ LATENCY REGISTER

Register	Address	R/W	Description	Reset Value
BURST_RD_LAT0 BURST_RD_LAT1	0x70100190 0x70180190	R	Burst Read Latency Register	0x0006

BURST_RD_LATn	Bit	Description	Initial State
Reserved	[31:3]		0
BURST_RD_LAT	[2:0]	Sets the burst read latency in cycles. The default value is 0x6. This value is copied from the MEM_CFG register[14:12]. Read-Only.	6

7.12.26 INTERRUPT PIN ENABLE REGISTER

Register	Address	R/W	Description	Reset Value
INT_ENABLE0 INT_ENABLE1	0x701001A0 0x701801A0	R/W	Interrupt Enable Register	0x0000

INT_PIN_ENABLEn	Bit	Description	Initial State
Reserved	[31:1]		0
INT_EN	[0]	Interrupt Enable. Enables generating interrupt signal to CPU with the information from INT_ERR_STAT and INT_ERR_MASK. Set by software during initialization. • 0 = Interrupt Disable. • 1 = Interrupt Enable.	0

7.12.27 INTERRUPT MONITOR CYCLE REGISTER

Register	Address	R/W	Description	Reset Value
INT_MON_CYC0 INT_MON_CYC1	0x701001B0 0x701801B0	R/W	Interrupt Monitor Cycle Count Register	0x01F4

INT_MON_CYCn	Bit	Description	Initial State
Reserved	[31:12]		0
INT_MON_CYC	[11:0]	Sets the number of cycles in between checks of the INT_ERR_STAT register and the memory device's status register. This register is only used if the Flash configuration register bit IOBE is clear.	500

7.12.28 ACCESS CLOCK REGISTER

Register	Address	R/W	Description	Reset Value
ACC_CLOCK0 ACC_CLOCK1	0x701001C0 0x701801C0	R/W	Access Clock Register	0x0003

ACC_CLOCKn	Bit	Description	Initial State
Reserved	[31:3]		0
ACCESS_CLK	[2:0]	Sets the number of cycles required to cover the access time of the Flash memory device. Before using Pipeline Read Ahead function, this value should be set to 7. Otherwise, Follows the formula (35ns/(Xm0SMCLK Period)+1).	3

Flash Core Clock (MHz)	Interface Clock (MHz)	ACCESS_CLK
133	66.5	3
100	50	2
60	30	2

7.12.29 SLOW READ PATH REGISTER

Register	Address	R/W	Description	Reset Value
SLOW_RD_PATH0 SLOW_RD_PATH1	0x701001D0 0x701801D0	R/W	Slow Read Path Register	0x0000

SLOW_RD_PATHn	Bit	Description	Initial State
Reserved	[31:1]		0
SRP	[0]	Delays the read data capture by 1/2 cycle to accommodate the board delay. Default is 0x0.	0

7.12.30 ERROR BLOCK ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
ERR_BLK_ADDR0 ERR_BLK_ADDR1	0x701001E0 0x701801E0	R	Error Block Address Register	0x0000

ERR_BLK_ADDRn	Bit	Description	Initial State
Reserved	[31:12]		0
FAIL_BLK_ADDR	[11:0]	After a program, load or erase error interrupt, this register will hold the block address of the failing operation. Read-Only.	0

7.12.31 FLASH VERSION ID REGISTER

Register	Address	R/W	Description	Reset Value
FLASH_VER_ID0 FLASH_VER_ID1	0x701001F0 0x701801F0	R	Flash Version ID Register	Memory dependent

FLASH_VER_IDn	Bit	Description	Initial State
Reserved	[31:16]		
FLASH_VER	[15:0]	The value programmed will depend on the actual memory device being used. The register is set by the flash controller after reset. Read-Only.	

7.12.32 FLASH AUXILIARY CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FLASH_AUX_CNTRL0	0x70100300	R/W	Bank0 Flash Auxiliary control register	0x0000
FLASH_AUX_CNTRL1	0x70180300	R/W	Bank1 Flash Auxiliary control register	0x0000

FLASH_AUX_CNTRL0/1	Bit	Description	Initial State
Reserved	[31:1]		0
Disable_Watchdog_reset	[0]	Watchdog timer reset disable. 0 = Enable watchdog timer reset logic in OneNAND controller. When enabled, watchdog state machine monitors various state machines of the flash controller and resets them if an operation appears to have hung. OneNAND controller is taken as hung if 663,900 memory clock counts are passed after the last valid command is given. 1 = Disable watchdog timer reset logic in OneNAND controller.	0

7.12.33 FLASH ASYNCHRONOUS FIFO COUNT REGISTER

Register	Address	R/W	Description	Reset Value
FLASH_AFIFO_CNT0	0x70100310	R	Number of data in asynchronous FIFO in flash controller 0.	0x0000
FLASH_AFIFO_CNT1	0x70180310	R	Number of data in asynchronous FIFO in flash controller 1.	0x0000

FLASH_AFIFO_CNT0/1	Bit	Description	Initial State
Reserved	[31:13]		0
Datacount_wr_fifo	[12:8]	Number of data in the write FIFO of OneNAND controller	0
Reserved	[7:6./]		0
Datacount_rd_fifo	[5:0]	Number of data in the read FIFO of OneNAND controller	0

8 NAND FLASH CONTROLLER

8.1 OVERVIEW

Recently NOR flash memory price has increased and price for SDRAM and a NAND flash memory is moderatly placed.

The S3C6410X is equipped with an internal SRAM buffer called ‘Steppingstone’.

Generally, the boot code will copy NAND flash content to SDRAM. Using hardware ECC, the NAND flash data validity will be checked. After the NAND flash content is copied to SDRAM, main program will be executed on SDRAM.

To use NAND Flash, 'XSELNAND' pin must be connected to High Level.

8.2 FEATURES

NAND flash controller features include:

1. NAND Flash memory I/F: Support 512Bytes and 2KB Page .
2. Software mode: User can directly access NAND flash memory. *for example this feature can be used in read/erase/program NAND flash memory*.
3. Interface: 8-bit NAND flash memory interface bus.
4. Hardware ECC generation, detection and indication (Software correction).
5. Support both SLC and MLC NAND flash memory : 1-bit ECC, 4-bit and 8-bit ECC for NAND flash.
   (Recommend: 1bit ECC for SLC, 4bit and 8bit ECC for MLC NAND Flash)
6. SFR I/F: Support Byte/half word/word access to Data and ECC Data register, and Word access to other registers
7. SteppingStone I/F: Support Byte/half word/word access.
8. The Steppingstone 8-KB internal SRAM buffer can be used for another purpose .

8.3 BLOCK DIAGRAM

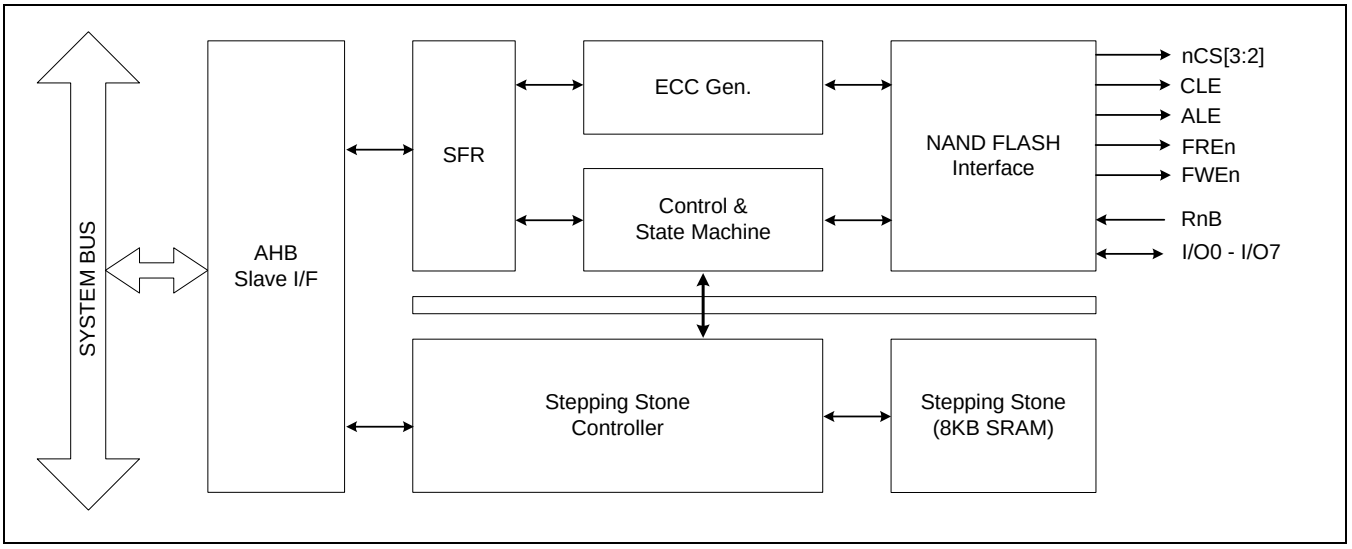


Figure 8-1. NAND Flash Controller Block Diagram

8.4 NAND FLASH MEMORY TIMING

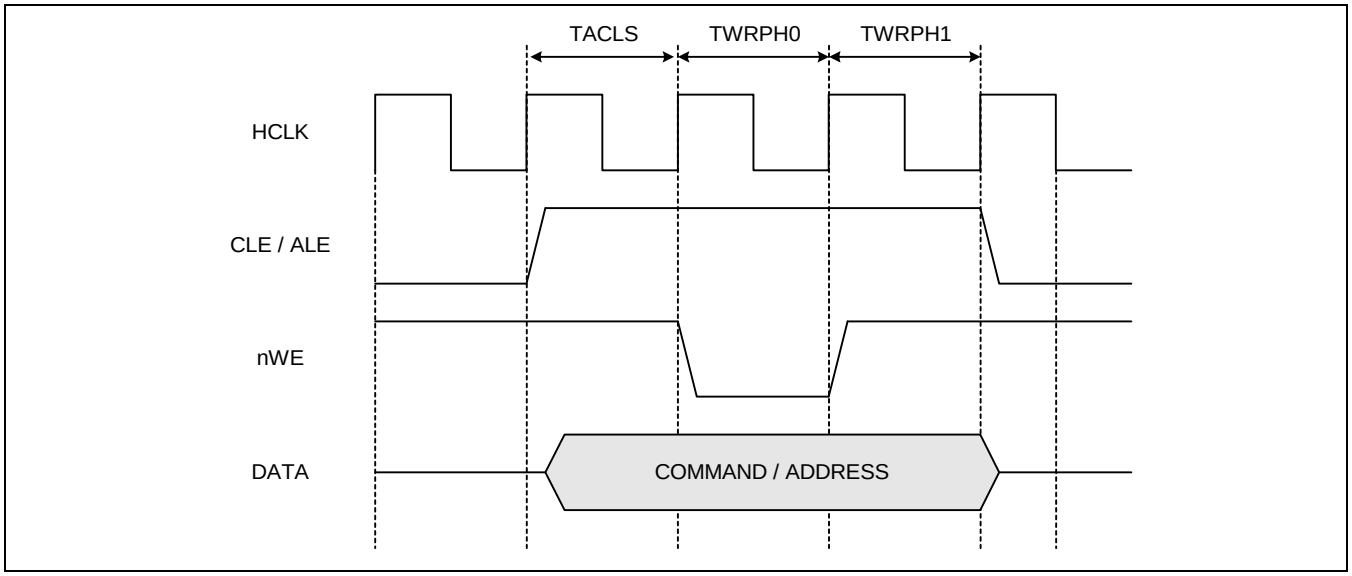


Figure 8-2. CLE & ALE Timing (TACLS=1, TWRPH0=0, TWRPH1=0) Block Diagram

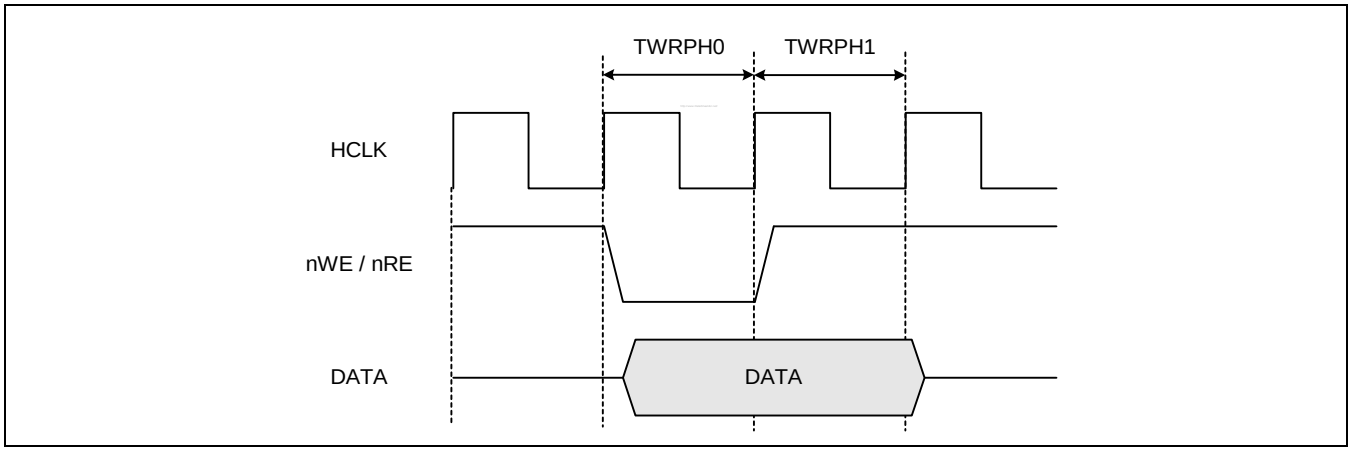


Figure 8-3. nWE & nRE Timing (TWRPH0=0, TWRPH1=0) Block Diagram

8.5 NAND FLASH ACCESS

S3C6410X does not support NAND flash access mechanism directly. It only supports signal control mechanism for NAND flash access. Therefore software is responsible for accessing NAND flash memory correctly.

1. Writing to the command register (NFCMMD) = the NAND Flash Memory command cycle
2. Writing to the address register (NFADDR) = the NAND Flash Memory address cycle
3. Writing to the data register (NFDATA) = write data to the NAND Flash Memory (write cycle)
4. Reading from the data register (NFDATA) = read data from the NAND Flash Memory (read cycle)
5. Reading main ECC registers and Spare ECC registers (NFMECCD0/1, NFSECCD) = read data from the NAND Flash Memory

NOTE:

In NAND flash access, you must check the RnB status input pin by polling the signal or using interrupt.

8.6 DATA REGISTER CONFIGURATION

1. 8-bit NAND Flash Memory Interface
   A. Word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	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	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Invalid value	Invalid value	2nd I/O[7:0]	1st I/O[7:0]

   C. Byte Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	Invalid value	Invalid value	Invalid value	1st I/O[7:0]

8.7 STEPPINGSTONE (8KB SRAM)

You can use this area*(Stemppingstone)* for various other purpose.

8.8 1-BIT / 4-BIT / 8-BIT ECC (ERROR CORRECTION CODE)

NAND flash controller has four ECC (Error Correction Code) modules for 1 bit ECC , one for 4bit ECC and one for 8bit ECC.

The 1bit ECC modules for main data area can be used for (up to) 2048 bytes ECC parity code generation, and 1 bit ECC module for spare area can be used for (up to) 4 bytes ECC Parity code generation.

Both 4bit and 8bit ECC modules can be used for only 512 bytes ECC parity code generation.

4 bit and 8bit ECC modules generate the parity codes for each 512 byte. However, 1 bit ECC modules generate parity code per byte lane separately.

4bit ECC modules generate max 7byte parity codes and 8 bit ECC modules generate 13byte parity codes at each 512/24 bytes.

8.8.1 ECC MODULE FEATURES

ECC generation is controlled by the ECC Lock (MainECCLock, SpareECCLock) bit of the Control register. When ECCLock is Low, ECC codes are generated by the H/W ECC modules.

1-BIT ECC Register Configuration

Following tables shows the configuration of 1-bit ECC value read from spare area of external NAND flash memory. For comparing to ECC parity code generated by the H/W modules, each ECC data read from memory must be written to NFMECCDn for main area and NFSECCD for spare area.

NOTE:

4-bit ECC decoding scheme is different to 1-bit ECC.

1. NAND Flash Memory Interface

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFMECCD0	Not used	2nd ECC for I/O[7:0]	Not used	1st ECC for I/O[7:0]
NFMECCD1	Not used	4th ECC for I/O[7:0]	Not used	3rd ECC for I/O[7:0]

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFSECCD	Not used	2nd ECC for I/O[7:0]	Not used	1st ECC for I/O[7:0]

8.8.2 1-BIT ECC PROGRAMMING ENCODING AND DECODING

1. To use 1-bit ECC in software mode, reset the ECCType to '0' (enable 1-bit ECC)'. ECC module generates ECC parity code for all read / write data when MainECCLock (NFCONT[7]) and SpareECCLock (NFCONT[6]) are unlocked('0'). You must reset ECC value by writing the InitMECC (NFCONT[5]) and InitSECC (NFCONT[4]) bit as '1' and have to clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before read or write data.
   MainECCLock (NFCONT[7]) and SpareECCLock(NFCONT[6]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is read or written, the ECC module generates ECC parity code on register NFMECC0/1.
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 will not be changed.
4. To generate spare area ECC parity code, Clear SpareECCLock (NFCONT[6]) bit to '0' (unlock).
5. Whenever data is read or written, the spare area ECC module generates ECC parity code on register NFSECC.
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 will not be changed.
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, after generating of ECC codes for main data area, you have to move the ECC parity codes (is stored to spare area) to NFMECCD0 and NFMECCD1. From this time, the NFECCERR0 have the valid error status values.

NOTE:

NFSECCD is for the ECC value in spare area. Usually, the user will write the ECC value generated from main data area to Spare area, which value will be the same as NFMECC0/1.

8.8.3 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 4-bit ECC). ECC module generates ECC parity code for 512-byte write data. So, you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' and have to clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before write data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is written, the 4-bit ECC module generates ECC parity code internally.
3. After you finish writing 512-byte data (not include spare area data), the parity codes are automatically updated to NFMECC0, NFMECC1 register. If you use 512-byte NAND flash memory, you can 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, you have to copy these parity codes to other memory like DRAM. After writing all main data, you can 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 '10'(enable 4-bit ECC). ECC module generates ECC parity code for 24-byte write data. So you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as '1' and have to clear the MainECCLock (NFCONT[7]) bit to '0'(Unlock) before write data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
5. Whenever data is written, the 4-bit ECC module generates ECC parity code internally.
6. When you finish writing 24-byte meta or extra data, the parity codes are automatically updated to NFMECC0, NFMECC1 register.You can program these parity codes to spare area.The parity codes have self-correctable information include parity code itself.

8.8.4 4-BIT ECC PROGRAMMING GUIDE (DECODING)

1. To use 4-bit ECC, set the MsgLength to 0(512-byte message length) and set the ECCType to “10”(enable 4-bit ECC). ECC module generates ECC parity code for 512-byte read data. So, you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as ‘1’ and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before read data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.
3. After you complete read 512-byte (does not include spare area data), you have to read parity codes. 4-bit ECC module needs parity codes to detect whether error bits are or not. So you have to read ECC parity code right after read 512-byte. Once ECC parity code is read, 4-bit ECC engine start to search any error internally. 4-bit ECC error searching engine need minimum 155 cycles to find any error.
4. When ECCDecDone (NFSTAT[6]) is set (‘1’), NFECCERR0 indicates whether error bit exist or not. If any error exists, you can fix it by referencing NFECCERR0/1 and NFMLCBITPT register.
5. If you have more main data to read, continue to step 2.
6. For meta data error check, set the MsgLength to 1(24-byte message length) and set the ECCType to ‘10’(enable 4-bit ECC). ECC module generates ECC parity code for 24-byte read data. So you have to reset ECC value by writing the InitMECC (NFCONT[5]) bit as ‘1’ and have to clear the MainECCLock (NFCONT[7]) bit to ‘0’(Unlock) before read data.
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.
7. Whenever data is read, the 4-bit ECC module generates ECC parity code internally.
8. After you complete read 24-byte, you have to read parity codes. 4-bit ECC module needs parity codes to detect whether error bits are or not. So you have to read ECC parity codes right after read 24-byte. Once ECC parity code is read, 4-bit ECC engine start to search any error internally. 4-bit ECC error searching engine need minimum 155 cycles to find any error.
9. When ECCDecDone (NFSTAT[6]) is set (‘1’), NFECCERR0 indicates whether error bit exist or not. If any error exists, you can fix it by referencing NFECCERR0/1 and NFMLCBITPT register.

8.8.5 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). ECC module generates ECC parity code for 512-byte write data. In order to start the ECC module, you have to write ‘1’ on the InitMECC (NFCONT[5]) bit after cleaning the MainECCLock (NFCONT[7]) bit to ‘0’ (Unlock). MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

   Note. In 8bit ECC, MainECCLock should be cleared before initiating InitMECC.

2. Whenever data is written, the 8bit ECC module generates ECC parity code internally.

3. After you finish writing 512-byte data (not include spare area data), the parity codes are automatically updated to NF8MECC0, NFMECC1, NF8MECC2, NF8MECC3 register. You have to check encoding done at NFSTAT register. And set the MainECCLock bit to ‘1’(lock). If you use 512-byte page size NAND flash memory, you can program these values directly to spare area. However, if you use NAND flash memory more than 512-byte page, you can’t program immediately. In this case, you have to copy these parity codes to other memory like DRAM. After writing all main data, you can 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). 8bit ECC module generates the ECC parity code for 24-byte data. In order to initiating the module, you have to write ‘1’ on the InitMECC (NFCONT[5]) bit after clearing the

MainECCLock (NFCONT[7]) bit to '0'(Unlock).
MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

Note. In 8bit ECC, MainECCLock should be cleared before initiating InitMECC.

5. Whenever data is written, the 8bit ECC module generates ECC parity code internally.

6. When you finish writing 24-byte meta or extra data, the parity codes are automatically updated to NF8MECC0, NFMECC1, NF8MECC2, NF8MECC3 register. you have to check encoding done at NFSTAT register. And set the MainECCLock bit to '1'(lock) . You can program these parity codes to spare area. The parity codes have self-correctable information include parity code itself.

8.8.6 8-BIT ECC PROGRAMMING GUIDE (DECODING)

1. To use 8bit ECC in software mode, set the MsgLength to 0(512-byte message length) and set the ECCType to "01"(enable 8bit ECC). 8bit ECC module generates ECC parity code for 512-byte read data. In order to initiating 8bit ECC module, you have to write '1' on the InitMECC (NFCONT[5]) bit after clearing the MainECCLock (NFCONT[7]) bit to '0'(Unlock).
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

   Note. In 8bit ECC, MainECCLock should be cleared before InitMECC

2. Whenever data is read, the MLC ECC module generates ECC parity code internally.

3. After you complete reading 512-byte data (not including spare area data), you must set the MainECCLock (NFCONT[7]) bit to '1'(Lock) after reading parity codes. 8bit ECC module needs parity codes to detect whether error bits exists or not. So you have to read the ECC parity code of 512-byte main data right after reading the 512-byte data. Once the ECC parity code is read, 8bit ECC engine starts searching any error internally. 8bit ECC error searching engine needs minimum 372 cycles to find any error. And set the MainECCLock bit to '1'(lock). ECCDecDone(NFSTAT[6]) can be used to check whether ECC decoding is completed or not.

4. When ECCDecDone (NFSTAT[6]) is set ('1'), NF8ECCERR0 indicates whether error bit exists or not. If any error exists, you can fix it by referencing NF8ECCERR0/1/2 and NFMLC8BITPT0/1 register.

5. If you have more main data to read, continue doing 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 8bit ECC). ECC module generates the ECC parity code for 24-byte data. In order to initiating the 8bit ECC module, you have to write '1' on the InitMECC (NFCONT[5]) bit after clearing the MainECCLock (NFCONT[7]) bit to '0'(Unlock).
   MainECCLock (NFCONT[7]) bit controls whether ECC Parity code is generated or not.

   Note. In 8bit ECC, MainECCLock should be cleared before InitMECC

7. Whenever data is read, the 8bit ECC module generates ECC parity code internally.

8. After you complete reading 24-byte, you must set the MainECCLock (NFCONT[7]) bit to '1'(Lock) after read ing the parity code for 24-byte data. MLC ECC module needs parity codes to detect whether error bits exists or not. So you have to read ECC parity codes right after reading 24-byte data. Once ECC parity code is read, 8bit ECC engine starts searching any error internally. 8bit ECC error searching engine needs minimum 372 cycles to find any error. And set the MainECCLock bit to '1'(lock). ECCDecDone(NFSTAT[6]) can be used to check whether ECC decoding is completed or not.

9. When ECCDecDone (NFSTAT[6]) is set ('1'), NF8ECCERR0 indicates whether error bit exist or not. If any

error exists, you can fix it by referencing NF8ECCERR0/1/2 and NF8MLCBITPT register.

8.9 Memory mapping(NOR-flash boot)

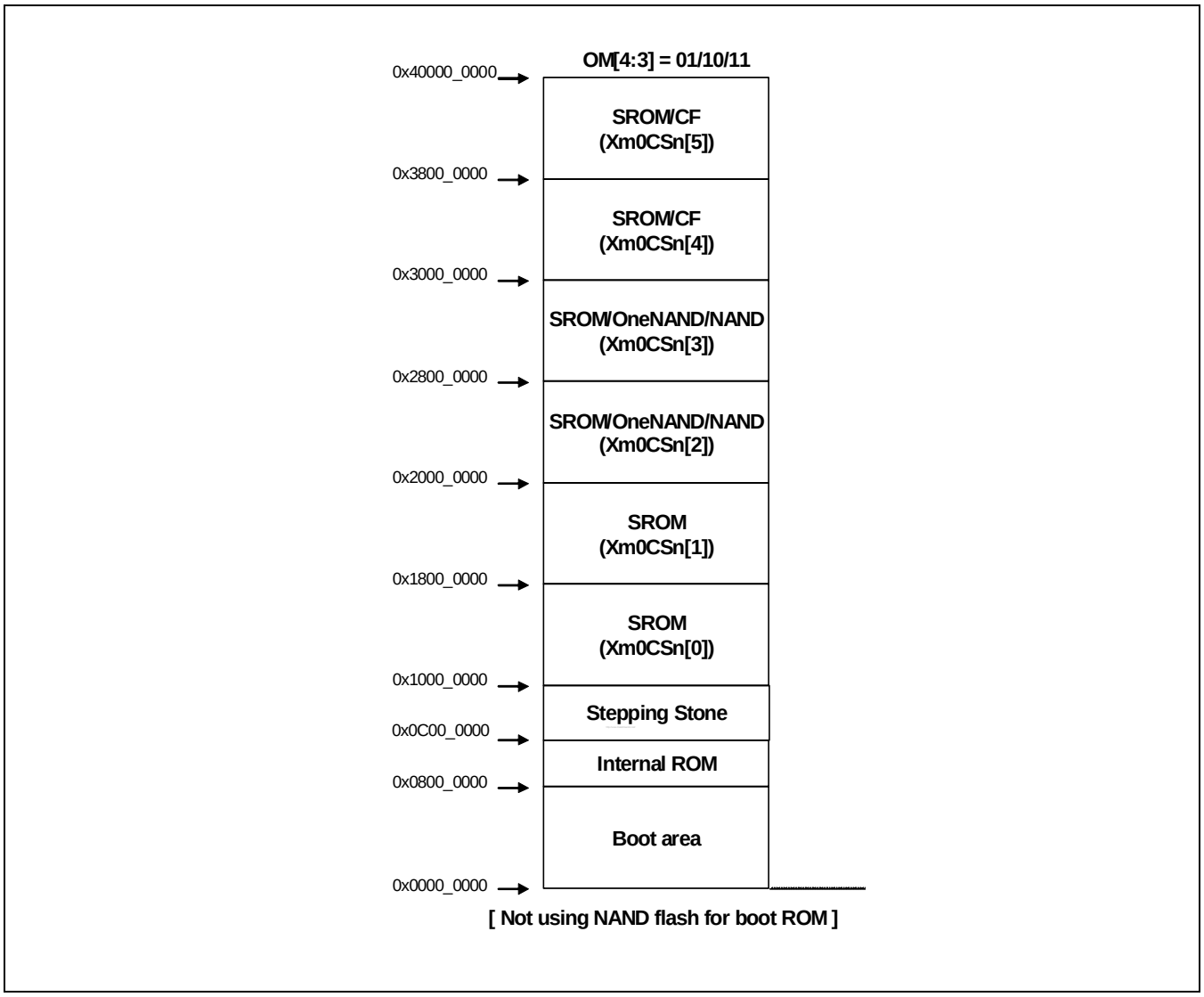


Figure 8-4. NAND Flash Memory Mapping Block Diagram

NOTE: SROM means ROM or SRAM or NOR type memory

8.10 NAND FLASH MEMORY CONFIGURATION

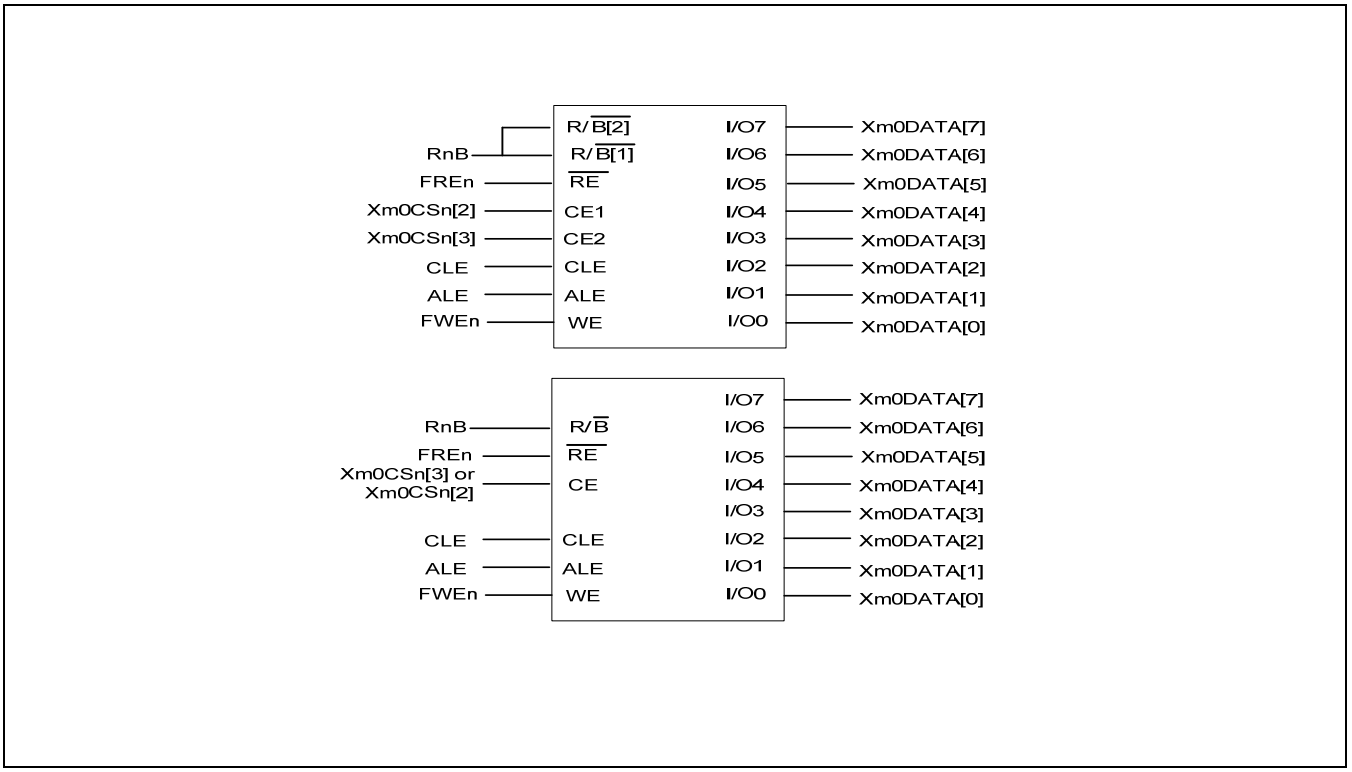


Figure 8-5. A 8-bit NAND Flash Memory Interface Block Diagram

NOTE: NAND CONTROLLER can support to control two nand flash memories .

	Other BOOT
Xm0CSn[2]	Configurable
Xm0CSn[3]	Configurable

8.11 NAND FLASH CONTROLLER SPECIAL REGISTERS

8.11.1 NAND FLASH CONTROLLER REGISTER MAP

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0xX000_100X	NFCONF	Configuration register
Base + 0x04	R/W	0x0001_00C6	NFCONT	Control register
Base + 0x08	R/W	0x0000_0000	NFCMMD	Command register
Base + 0x0c	R/W	0x0000_0000	NFADDR	Address register
Base + 0x10	R/W	0xXXXX_XXXX	NFDATA	Data register
Base + 0x14	R/W	0x0000_0000	NFMECCD0	1st and 2nd main ECC data register
Base + 0x18	R/W	0x0000_0000	NFMECCD1	3rd and 4th main ECC data register
Base + 0x1c	R/W	0x0000_0000	NFSECCD	Spare ECC read register
Base + 0x20	R/W	0x0000_0000	NFSBLK	Programmable start block address register
Base + 0x24	R/W	0x0000_0000	NFEBLK	Programmable end block address register
Base + 0x28	R/W	0x0080_001D	NFSTAT	NAND status registet
Base + 0x2C	R	0xXXXX_XXXX	NFECCERR0	ECC error status0 register
Base + 0x30	R	0x0000_0000	NFECCERR1	ECC error status1 register
Base + 0x34	R	0xXXXX_XXXX	NFMECC0	Generated ECC status0 register
Base + 0x38	R	0xXXXX_XXXX	NFMECC1	Generated ECC status1 register
Base + 0x3C	R	0xXXXX_XXXX	NFSECC	Generated Spare area ECC status register
Base + 0x40	R	0x0000_0000	NFMLCBITPT	4-bit ECC error bit pattern register
Base + 0x44	R	0x4000_0000	NF8ECCERR0	8bit ECC error status0 register
Base + 0x48	R	0x0000_0000	NF8ECCERR1	8bit ECC error status1 register
Base + 0x4C	R	0x0000_0000	NF8ECCERR2	8bit ECC error status2 register
Base + 0x50	R	0xXXXX_XXXX	NFM8ECC0	Generated 8-bit ECC status0 register
Base + 0x54	R	0xXXXX_XXXX	NFM8ECC1	Generated 8-bit ECC status1 register
Base + 0x58	R	0xXXXX_XXXX	NFM8ECC2	Generated 8-bit ECC status2 register
Base + 0x5C	R	0xXXXX_XXXX	NFM8ECC3	Generated 8-bit ECC status3 register
Base + 0x60	R	0x0000_0000	NFMLC8BITPT 0	8-bit ECC error bit pattern 0 register
Base + 0x64	R	0x0000_0000	NFMLC8BITPT 1	8-bit ECC error bit pattern 1 register

Base = 0x7020_0000
Stepping STON : 0x0C00_0000 ~ 0x0C00_1FFF (8K)

0x0000_0000 ~ 0x0000_1FFF (8K)*

*In 6410 memory map, stepping stone memory is in the area between 0x0C00_0000 and 0x0C00_1FFF.

8.11.2 NAND FLASH CONFIGURATION REGISTER

Register	Address	R/W	Description	Reset Value
NFCONF	0x70200000	R/W	NAND Flash Configuration register	0xX000100X

NFCONF	Bit	Description	Initial State
Reserved			
MLCClkCtrl	[30]	Clock control for 4-bit ECC & 8-bit ECC engine.(Hidden Spec.) 0: Recommended when system clock is more than 66MHz. 1: Recommended when system clock is less than 66MHz	0
Reserved	[29:26]	Reserved	0000
MsgLength	[25]	Message (Data) length for 4/8 bit ECC 0: 512-byte 1: 24-byte	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 Note. Don't confuse the value of 4-bit ECC and 8-bit ECC.	H/W Set (CfgBootEcc)
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
Reserved			
Reserved	[2]	Reserved. Must be written 1	1
Reserved			
Reserved	[0]	Reserved. Must be written 0.	0

8.11.3 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
NFCONT	0x70200004	R/W	NAND Flash control register	0x000100C6

NFCONT	Bit	Description	Initial State
Reserved	[31:19]	Reserved	0
ECC Direction	[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
Lock-tight	[17]	Lock-tight configuration 0: Disable lock-tight 1: Enable lock-tight, Once this bit is set to 1, you cannot clear. Only reset or wake up from sleep mode can make this bit disable (cannot cleared by software). When it is set to 1, the area setting in NFSBLK (0x70200020) to NFEBLK (0x70200024) is unlocked, and except this area, write or erase command will be invalid and only read command is valid. When you try to write or erase locked area, the illegal access will be occurred (NFSTAT [5] bit will be set). If the value of NFSBLK is bigger than that of NFEBLK, entire area will be locked.	0
Soft Lock	[16]	Soft Lock configuration 0: Disable lock 1: Enable lock Soft lock area can be modified at any time by software. When it is set to 1, the area setting in NFSBLK (0x70200020) to NFEBLK (0x70200024) is unlocked, and except this area, write or erase command will be invalid and only read command is valid. When you try to write or erase locked area, the illegal access will be occurred (NFSTAT [5] bit will be set). If the NFSBLK and NFEBLK are same, entire area will be locked.	1
Reserved	[15:13]	Reserved. Should be written to 0.	000
EnbECCDecINT	[12]	4-bit, 8-bit ECC decoding completion interrupt control 0: Disable interrupt 1: Enable interrupt	0
8bitStop	[11]	8-bit ECC encoding/decoding operation initialization. 8bit ECC module generates parity code for 512/24 byte data. If you want to stop generating ECC parity before completing current work, you must set this value to “1” for initializing 8bit ECC module. This bit will be cleared automatically.	0

NFCONT	Bit	Description	Initial State
EnbIllegalAccINT	[10]	Illegal access interrupt control 0: Disable interrupt 1: Enable interrupt Illegal access interrupt will occurs when CPU tries to program or erase locking area (the area setting in NFSBLK (0x70200020) to NFEBLK (0x70200024).	0
EnbRnBINT	[9]	RnB status input signal transition interrupt control 0: Disable RnB interrupt 1: Enable RnB interrupt	0
RnB_TransMode	[8]	RnB transition detection configuration 0: Detect rising edge 1: Detect falling edge	0
MainECCLock	[7]	Lock Main area ECC generation 0: Unlock Main area ECC 1: Lock Main area ECC Main area ECC status register is NFMECC0/1(0x70200034/38),	1
SpareECCLock	[6]	Lock Spare area ECC generation. 0: Unlock Spare ECC 1: Lock Spare ECC Spare area ECC status register is NFSECC(0x7020003C),	1
InitMECC	[5]	1: Initialize main area ECC decoder/encoder (write-only) Caution : In case of 8bit ECC, you must set this bit carefully. If you set this bit before completing current encoding/decoding, it cause some trouble to 8bit ECC module. If you want to stop current work and start encoding/decoding for new data, you must set 8bitStop(NFCONT[11]) before this bit.	0
InitSECC	[4]	1: Initialize spare area ECC decoder/encoder (write-only)	0
Reserved	[3]	Reserved	0
Reg_nCE1	[2]	NAND Flash Memory Xm0CSn3 signal control 0: Force Xm0CSn3 to low(Enable chip select) 1: Force Xm0CSn3 to High(Disable chip select) **Note**: Even Reg_nCE1 and Reg_nCE0 are set to zero simultaneously, only one of them is asserted.	1
Reg_nCE0	[1]	NAND Flash Memory Xm0CSn2 signal control 0: Force Xm0CSn2 to low(Enable chip select) 1: Force Xm0CSn2 to High(Disable chip select) **Note**: This value is only valid while MODE bit is 1	1
MODE	[0]	NAND Flash controller operating mode 0: NAND Flash Controller Disable (Don't work) 1: NAND Flash Controller Enable	0

8.11.4 COMMAND REGISTER

Register	Address	R/W	Description	Reset Value
NFCMMD	0x70200008	R/W	NAND Flash command set register	0x00

NFCMMD	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x00
NFCMMD	[7:0]	NAND Flash memory command value	0x00

8.11.5 ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
NFADDR	0x7020000C	R/W	NAND Flash address set register	0x0000XX00

REG_ADDR	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x00
NFADDR	[7:0]	NAND Flash memory address value	0x00

8.11.6 DATA REGISTER

Register	Address	R/W	Description	Reset Value
NFDATA	0x70200010	R/W	NAND Flash data register	0xXXXX

NFDATA	Bit	Description	Initial State
NFDATA	[31:0]	NAND Flash read/program data value for I/O (Note) Refer to **DATA REGISTER CONFIGURATION**.	0xXXXX

8.11.7 MAIN DATA AREA ECC REGISTER

Register	Address	R/W	Description	Reset Value
NFMECCD0	0x70200014	R/W	NAND Flash ECC 1st 2nd register for main area data read (Note) Refer to **ECC MODULE FEATURES**.	0x00000000
NFMECCD1	0x70200018	R/W	NAND Flash ECC 3rd 4th register for main area data read (Note) Refer to **ECC MODULE FEATURES**.	0x00000000

NFMECCD0	Bit	Description	Initial State
Reserved	[31:24]	Not used	0x00
ECCData1	[23:16]	ECC1 for I/O[7:0]	0x00
Reserved	[15:8]	Not used	0x00
ECCData0	[7:0]	ECC0 for I/O[7:0]	0x00

NOTE: Only word access is valid.

NFMECCD1	Bit	Description	Initial State
Reserved	[31:24]	Not used	0x00
ECCData3	[23:16]	ECC3 for I/O[7:0]	0x00
Reserved	[15:8]	Not used	0x00
ECCData2	[7:0]	ECC2 for I/O[7:0]	0x00

8.11.8 SPARE AREA ECC REGISTER

Register	Address	R/W	Description	Reset Value
NFSECCD	0x7020001C	R/W	NAND Flash ECC(Error Correction Code) register for spare area data read	0x00000000

NFSECCD	Bit	Description	Initial State
Reserved	[31:24]	Not used	0x00
SECCData1	[23:16]	2nd Spare area ECC for I/O[7:0]	0x00
Reserved	[15:8]	Not used	0x00
SECCData0	[7:0]	1st Spare area ECC for I/O[7:0]	0x00

NOTE: Only word or half word access is valid.

8.11.9 PROGRMMABLE BLOCK ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
NFSBLK	0x70200020	R/W	NAND Flash programmable start block address	0x000000
NFEBLK	0x70200024	R/W	NAND Flash programmable end block address Nand Flash can be programmed between start and end address. When the Soft lock or Lock-tight is enabled and the Start and End address has same value, Entire area of NAND flash will be locked.	0x000000

NFSBLK	Bit	Description	Initial State
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

NFEBLK	Bit	Description	Initial State
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

The NFSLK and NFEBLK can be changed while Soft lock bit(NFCONT[16]) is enabled. But cannot be changed when Lock-tight bit(NFCONT[17]) is set.

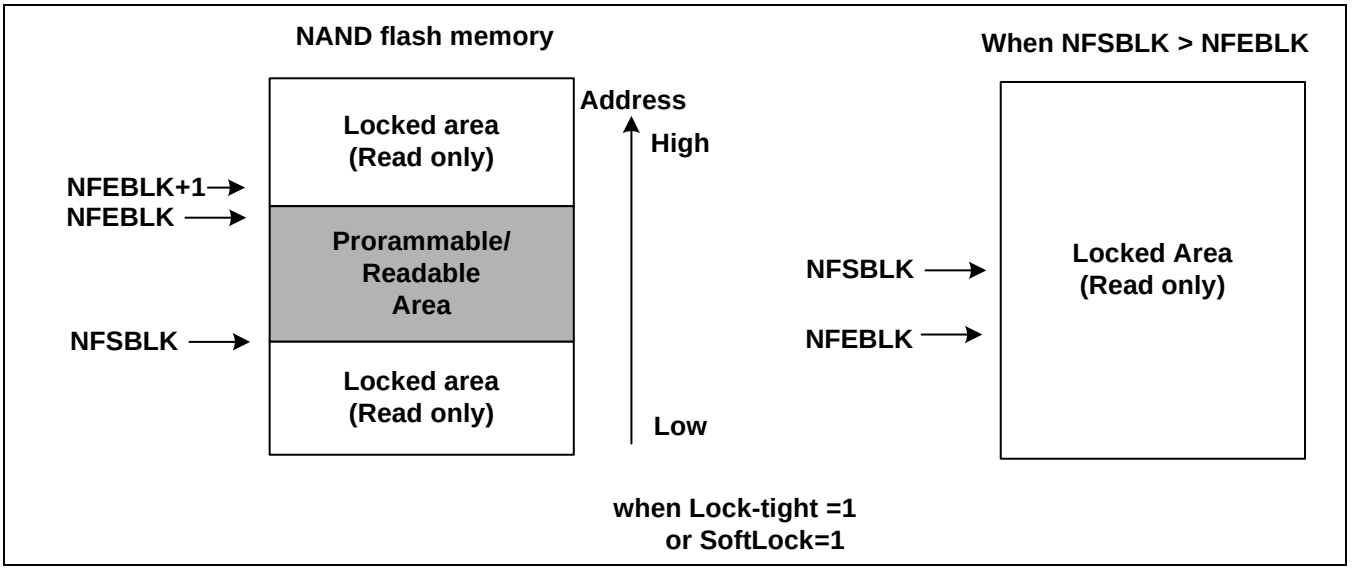


Figure 8-6. Softlock and Lock-tight

8.11.10 NFCON STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFSTAT	0x70200028	R/W	NAND Flash operation status register	0x0080001D

NFSTAT	Bit	Description	Initial State
Reserved	[31:24]	Read undefined	0x00
Reserved			
Reserved	[22:7]	Reserved	0x00
ECCDecDone	[6]	When 4-bit ECC or 8-bit ECC decoding is finished, this value set and issue interrupt if enabled. 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 makes this bit set. 0: illegal access is not detected 1: illegal access is detected	0
RnB_TransDetect	[4]	When RnB low to high transition is occurred, this value set and issue interrupt if enabled. 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
NCE[1] (Read-only)	[3]	The status of Xm0CSn3 output pin	1
NCE[0] (Read-only)	[2]	The status of Xm0CSn2 output pin	1
Reserved	[1]	Reserved	0
RnB (Read-only)	[0]	The status of RnB input pin. 0: NAND Flash memory busy 1: NAND Flash memory ready to operate	1

8.11.11 ECC0/1 ERROR STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFECCERR0	0x7020002C	R	NAND Flash ECC Error Status register for I/O [7:0]	0xX0XX_XXXX
NFECCERR1	0x70200030	R	NAND Flash ECC Error Status register for I/O [7:0]	0x0000_0000

When ECCType is 1-bit ECC.

NFECCERR0	Bit	Description	Initial State
Reserved	[31:25]	Reserved	0x00
SErrorDataNo	[24:21]	In spare area, Indicates which number data is error	0011
SErrorBitNo	[20:18]	In spare area, Indicates which bit is error	111
MErrorDataNo	[17:7]	In main data area, Indicates which number data is error	0x7FF
MErrorBitNo	[6:4]	In main data area, Indicates which bit is error	111
SpareError	[3:2]	Indicates whether spare area bit fail error occurred 00: No Error 01: 1-bit error(correctable) 10: Uncorrectable 11: ECC area error	10
MainError	[1:0]	Indicates whether main data area bit fail error occurred 00: No Error 01: 1-bit error(correctable) 10: Uncorrectable 11: ECC area error	10

NFECCERR1	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x00

NOTE: The above values are only valid when both ECC register and ECC status register have valid value.

When ECCType is 4-bit ECC.

NFECCERR0	Bit	Description	Initial State
ECC Busy	[31]	Indicates the 4-bit ECC decoding engine is searching whether a error exists or not 0: Idle 1: Busy	0
ECC Ready	[30]	ECC Ready bit	1
Free Page	[29]	Inidicates the page data red from NAND flash has all 'FF' value.	0
4-bit MECC Error	[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 Note : If it happens that there are more errors than 4 bits, 4-bit ECC module does not ensure right detection.	000
2nd Bit Error Location	[25:16]	Error byte location of 2nd bit error	0x00
Reserved	[15:10]	Reserved	
1st Bit Error Location	[9:0]	Error byte location of 1st bit error	0x00

NOTE: These values are updated when ECCDecDone (NFSTAT[6]) is set ('1').

NFECCERR1	Bit	Description	Initial State
Reserved	[31:26]	Reserved	0x00
4th Bit Error Location	[25:16]	Error byte location of 4th bit error	0x00
Reserved	[15:10]	Reserved	
3rd Bit Error Location	[9:0]	Error byte location of 3rd bit error	0x00

NOTE: These values are updated when ECCDecDone (NFSTAT[6]) is set ('1').

8.11.12 MAIN DATA AREA ECC0 STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFMECC0	0x70200034	R	NAND Flash ECC status register	0xXXXXXX
NFMECC1	0x70200038	R	NAND Flash ECC status register	0xXXXXXX

When ECCType is 1-bit ECC

NFMECC0	Bit	Description	Initial State
MECC0_3	[31:24]	ECC3 for data[7:0]	0xXX
MECC0_2	[23:16]	ECC2 for data[7:0]	0xXX
MECC0_1	[15:8]	ECC1 for data[7:0]	0xXX
MECC0_0	[7:0]	ECC0 for data[7:0]	0xXX

NFMECC1	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x00000000

NOTE: The NAND flash controller generate NFMECC when read or write main area data while the MainECCLock (NFCONT[7]) bit is '0'(Unlock).

When ECCType is 4-bit ECC.

NFMECC0	Bit	Description	Initial State
4th Parity	[31:24]	4th Check Parity generated from main area	0x00
3rd Parity	[23:16]	3rd Check Parity generated from main area	0x00
2nd Parity	[15:8]	2nd Check Parity generated from main area	0x00
1st Parity	[7:0]	1st Check Parity generated from main area	0x00

NFMECC1	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0x00
7th Parity	[23:16]	7th Check Parity generated from main area	0x00
6th Parity	[15:8]	6th Check Parity generated from main area	0x00
5th Parity	[7:0]	5th Check Parity generated from main area	0x00

NOTE: The NAND flash controller generate these ECC parity codes when write main area data while the MainECCLock (NFCONT[7]) bit is '0' (unlock).

8.11.13 SPARE AREA ECC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFSECC	0x7020003C	R	NAND Flash ECC register for I/O [7:0]	0xXXXXXX

NFSECC	Bit	Description	Initial State
Reserved	[31:16]	Reserved	0xXXXX
SECC0_1	[15:8]	Spare area ECC1 Status for I/O[7:0]	0xXX
SECC0_0	[7:0]	Spare area ECC0 Status for I/O[7:0]	0xXX

NOTE: The NAND flash controller generate NFSECC when read or write spare area data while the SpareECCLock (NFCONT[6]) bit is '0' (Unlock).

8.11.14 4-BIT ECC ERROR PATTEN REGISTER

Register	Address	R/W	Description	Reset Value
NFMLCBITPT	0x70200040	R	NAND Flash 4-bit ECC Error Pattern register for data[7:0]	0x00000000

NFMLCBITPT	Bit	Description	Initial State
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

8.11.15 ECC 0/1/2 FOR 8BIT ECC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NF8ECCERR0	0x7020_0044	R	NAND Flash ECC Error Status register 0	0x4000_0000
NF8ECCERR1	0x7020_0048	R	NAND Flash ECC Error Status register 1	0x0000_0000
NF8ECCERR2	0x7020_004C	R	NAND Flash ECC Error Status register 2	0x0000_0000

NFECCERR0	Bit	Description	Initial State
MLC8ECCBusy	[31]	Indicates the 8-bit ECC decoding engine is searching whether a error exists or not 0: Idle 1: Busy	b'0
MLC8ECCReady	[30]	ECC Ready bit	b'1
Reserved	[29]	-	-
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	b'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 when ECCDecodeDone (NFSTAT[6]) is set ('1').

NFECCERR1	Bit	Description	Initial State
MLCErrLocation5	[31:22]	Error byte location of 5th bit error	0x000
Reserved	[21]	Reserved	b'0
MLCErrLocation4	[20:11]	Error byte location of 4th bit error	0x000
Reserved	[10]	Reserved	b'0
MLCErrLocation3	[9:0]	Error byte location of 3rd bit error	0x000

Note : These values are updated when ECCDecodeDone (NFSTAT[6]) is set ('1').

NFECCERR1	Bit	Description	Initial State
MLCErrLocation8	[31:22]	Error byte location of 8th bit error	0x000
Reserved	[21]	Reserved	b'0
MLCErrLocation7	[20:11]	Error byte location of 7th bit error	0x000
Reserved	[10]	Reserved	b'0
MLCErrLocation6	[9:0]	Error byte location of 6th bit error	0x000

Note : These values are updated when ECCDecodeDone (NFSTAT[6]) is set ('1').

8.11.16 8BIT ECC MAIN DATA ECC 0/1/2/3 STATUS REGISTER

Register	Address	R/W	Description	Reset Value
NFM8ECC0	0x7020_0050	R	8bit ECC status register	0xXXXX_XXXX
NFM8ECC1	0x7020_0054	R	8bit ECC status register	0xXXXX_XXXX
NFM8ECC2	0x7020_0058	R	8bit ECC status register	0xXXXX_XXXX
NFM8ECC3	0x7020_005C	R	8bit ECC status register	0xXXXX_XXXX

NFM8ECC0	Bit	Description	Initial State
4th Parity	[31:24]	4th Check Parity generated from main area (512-byte)	0xXX
3rd Parity	[23:16]	3rd Check Parity generated from main area (512-byte)	0xXX
2nd Parity	[15:8]	2nd Check Parity generated from main area (512-byte)	0xXX
1st Parity	[7:0]	1st Check Parity generated from main area (512-byte)	0xXX

NFM8ECC1	Bit	Description	Initial State
8th Parity	[31:24]	8th Check Parity generated from main area (512-byte)	0xXX
7th Parity	[23:16]	7th Check Parity generated from main area (512-byte)	0xXX
6th Parity	[15:8]	6th Check Parity generated from main area (512-byte)	0xXX
5th Parity	[7:0]	5th Check Parity generated from main area (512-byte)	0xXX

NFM8ECC2	Bit	Description	Initial State
12th Parity	[31:24]	12th Check Parity generated from main area (512-byte)	0xXX
11th Parity	[23:16]	11th Check Parity generated from main area (512-byte)	0xXX
10th Parity	[15:8]	10th Check Parity generated from main area (512-byte)	0xXX
9th Parity	[7:0]	9th Check Parity generated from main area (512-byte)	0xXX

NFM8ECC3	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x000000
13th Parity	[7:0]	13th Check Parity generated from main area (512-byte)	0x00

Note: The NAND flash controller generate these ECC parity codes when write main area data while the MainECCLock (NFCONT[7]) bit is '0'(unlock).

8.11.17 8bit ECC ERROR PATTERN REGISTER

Register	Address	R/W	Description	Reset Value
NFMLC8BITPT0	0x7020_0060	R	NAND Flash 8-bit ECC Error Pattern register0 for data[7:0]	0x0000_0000
NFMLC8BITPT1	0x7020_0064	R	NAND Flash 8-bit ECC Error Pattern register1 for data[7:0]	0x0000_0000

NFMLC8BITPT0	Bit	Description	Initial State
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

NFMLC8BITPT1	Bit	Description	Initial State
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

9 CF CONTROLLER

This chapter describes the functions and usage of CF Controller in S3C6410X RISC microprocessor.

9.1 OVERVIEW

CF controller supports both PC card memory/IO mode & True-IDE mode.

CF controller is compatible with CF standard specification R3.0.

9.2 FEATURES

9.2.1 THE CF CONTROLLER FEATURES:

The CF controller supports only 1 slot.

The CF controller consists of 2 parts – PC card controller & ATA controller. They are multiplexing from or to PAD signals. You must use only 1 mode, PC card or True-IDE mode. Default mode is PC card mode. The CF controller has a top level SFR that includes card power enable bit, output port enable bit & mode select (True-IDE or PC card) bit.

9.2.2 THE PC CARD CONTROLLER FEATURES:

The PC card controller has 2 half-word (16bits) write buffers & 4 half-word (16bits) read buffers.

The PC card controller has 5 word-sized (32bits) Special Function Registers:

- 3 timing configuration registers. (Attribute memory, Common memory, I/O interface)
- 1 status & control configuration register
- 1 interrupt source & mask register

Timing configuration register consists of 3 parts – Setup, Command & Hold.

- PC card interface includes 4 states (IDLE, SETUP, COMMAND & HOLD)
- Each part of register indicates the operation timing of each state.

9.2.3 THE ATA CONTROLLER FEATURES:

The ATA controller is compatible with the ATA standard.

The ATA controller has 1 FIFO that is 16 x 32-bit.

The ATA controller has internal DMA controller (from ATA device to memory or from memory to ATA device).

AHB master (DMA controller) support 8 burst & word size transfer.

Direct mode and Indirect mode support

- Direct mode: support UDMA mode only.
- Indirect mode: support I/O mode, memory mode, and True-IDE mode(except UDMA mode).

9.3 I/O DESCRIPTION

Table 9-1. I/O signal description

Indirect mode	Direct mode (UDMA mode only)	I/O	Description
Xm0CSn[4]	XhiCSn XhiADR[8]	O	Card enable strobe PC card mode : lower byte enable strobe True-IDE mode : chip selection (nCS0)
Xm0CSn[5]	XhiCSn_main XhiADR[9]	O	Card enable strobe PC card mode : higher byte enable strobe True-IDE mode : chip selection (nCS1)
Xm0REGata	Xm0REGata XhiADDR[6]	O	Register in CF card strobe PC card mode : It is used for accessing register in CF card True-IDE mode : DMA Acknowledge
Xm0OEata	Xm0OEata	O	Output enable strobe PC card mode : output enable strobe for memory True-IDE mode : GND.
Xm0RESETata	Xm0RESETata XhiADDR[4]	O	CF card reset PC card mode : active high True-IDE mode : active low
Xm0WEata	Xm0WEata	O	Write enable strobe PC card mode : output enable strobe for memory True-IDE mode: VCC.
Xm0OEn	XhiCSn_sub XhiADR[10]	O	Read strobe for I/O mode UDMA mode : host strobe
Xm0WEn	XhiWEn XhiADR[11]	O	Write strobe for I/O mode
Xm0ADDR[0]	XhiADDR[0], XuRXD[2], XmmcDATA1[4]	O	CF card address PC card mode : full address use True-IDE mode : only ADDR[2:0] use, The other address line is connected to GND.
Xm0ADDR[1]	XhiADDR[1], XuTXD[2], XmmcDATA1[5]	O	
Xm0ADDR[2]	XhiADDR[2], XmmcDATA1[6], XuRXD[3]	O	
Xm0ADDR[10:3]		O	
Xm0DATA[15:0]	XhiDATA[0]	B	CF data bus
	XhiDATA[1]	B	
	XhiDATA[2]	B	
	XhiDATA[3]	B	
	XhiDATA[4]	B	
	XhiDATA[5]	B	
	XhiDATA[6]	B	
	XhiDATA[7]	B	
	XhiDATA[8] (XhiDATA[16])	B	

	XhiDATA[9] (XhiDATA[17])	B	
	XhiDATA[10](XhiCSn)	B	
	XhiDATA[11] (XhiCSn_main)	B	
	XhiDATA[12] (XhiCSn_sub)	B	
	XhiDATA[13] (XhiWE)	B	
	XhiDATA[14] (XhiOEn)	B	
	XhiDATA[15](XhiIRQn)	B	
Xm0CData	Xm0CData	I	Card detect signals
	XhiADDR[7]		
Xm0INTata	Xm0INTata	I	Interrupt request from CF card. PC card mode : active low (memory mode : level triggering, I/O mode : edge triggering) True-IDE mode : active high
	XhiADDR[3]		
Xm0WAITn	XhiADR[12]	I	Wait signal from CF card UDMA mode : device strobe
	XhiOEnI		
Xm0INPACKata	Xm0INPACKata	I	Input acknowledge in I/O mode PC card mode : not used True-IDE mode : DMA request
	XhiADDR[5]		

9.4 BLOCK DIAGRAM

9.4.1 CF CONTROLLER BLOCK DIAGRAM

The CF Controller block diagram is shown below in Figure 9-1.

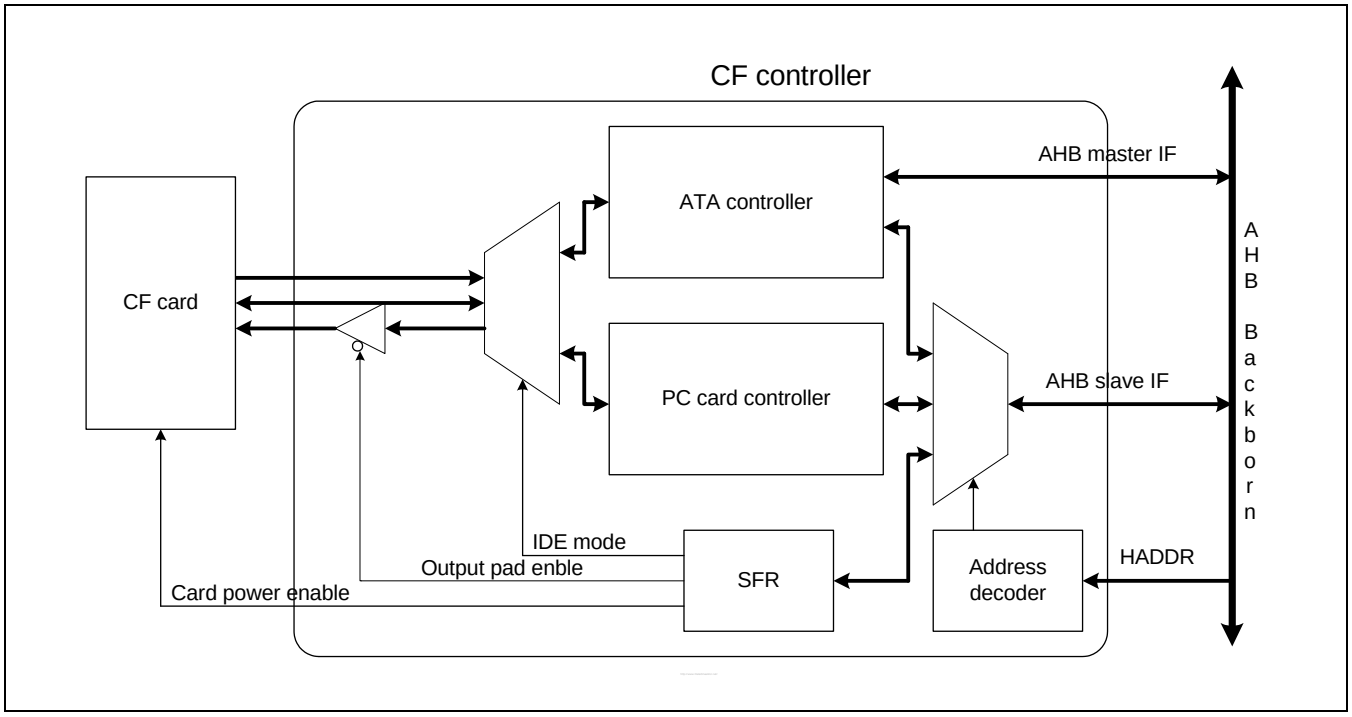


Figure 9-1. CF Controller Block Diagram

9.5 TIMING DIAGRAM

9.5.1 PC CARD MODE

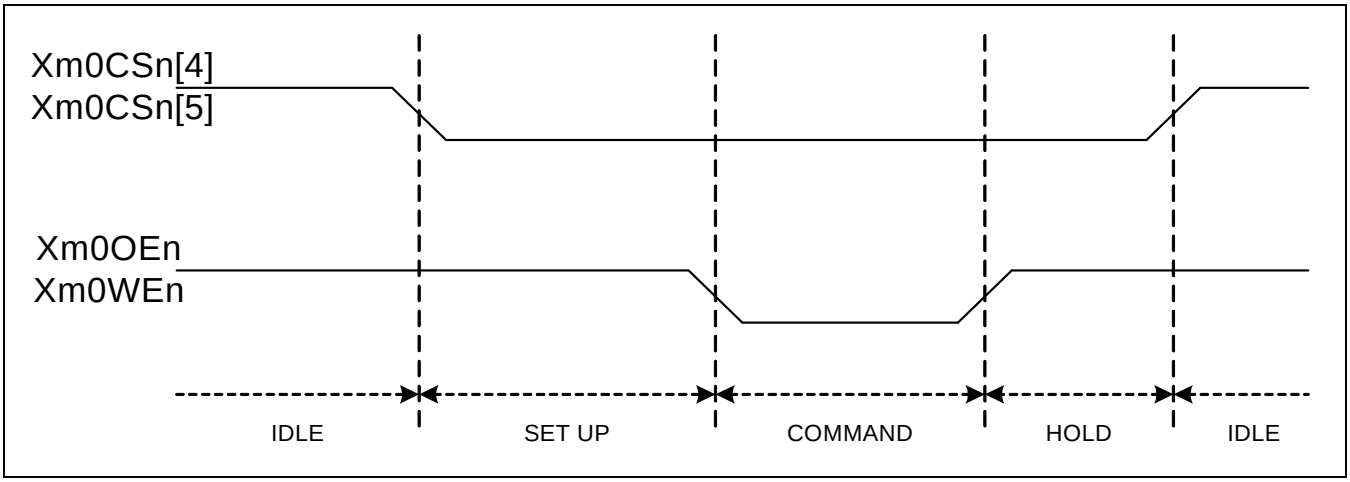


Figure 9-2. PC Card State Definition

Table 9-2. Timing Parameter Each PC Card Mode

Area	Attribute memory	I/O interface	Common memory
	(min, Max) nS		
Set up	(30, --)	(70, --)	(30, --)
Command	(150, --)	(165, --)	(150, --)
Hold	(30, --)	(20, --)	(20, --)
S + C + H	(300, --)	(290, --)	(--, --)

9.5.2 TRUE-IDE MODE

PIO Mode

PIO Mode Waveform

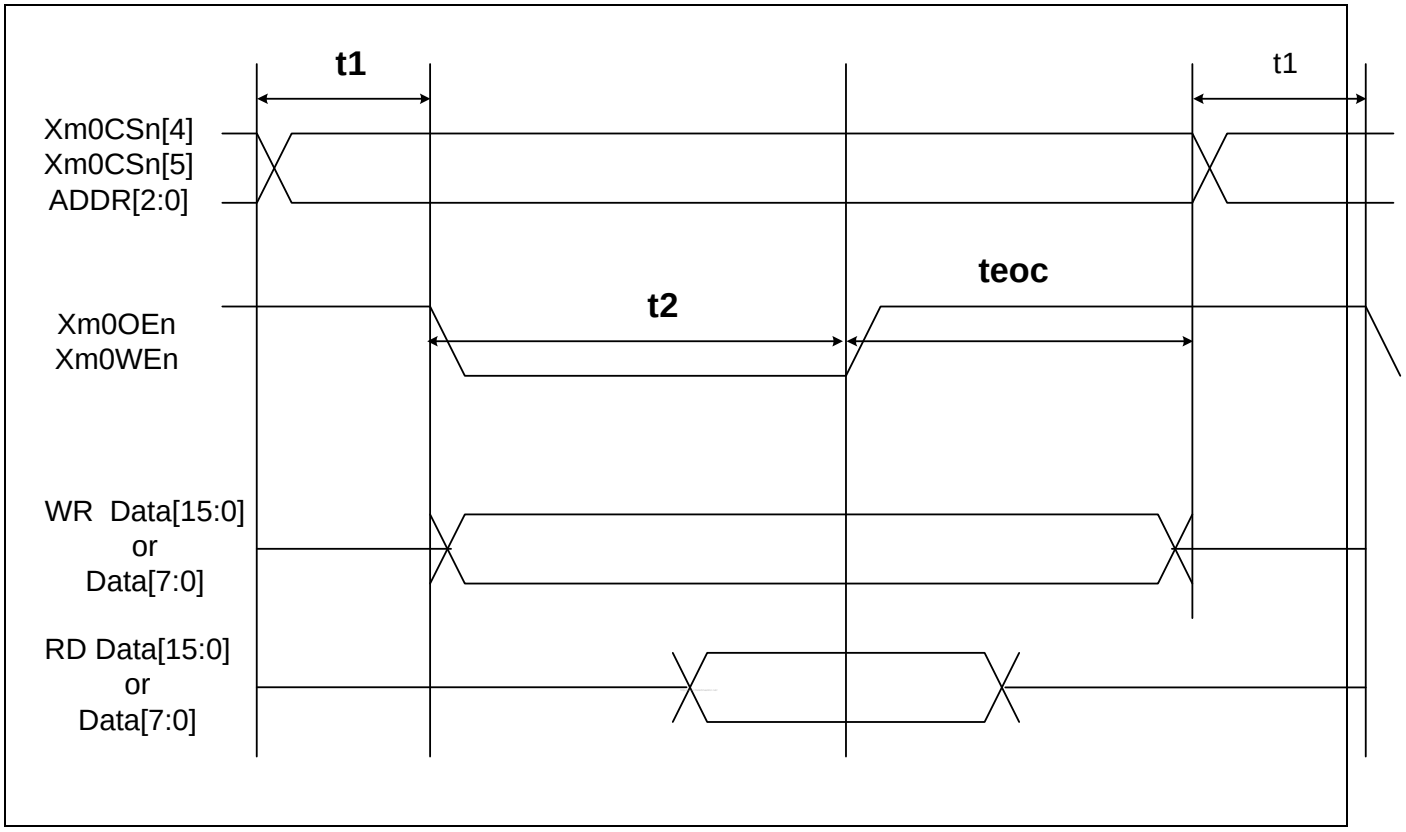


Figure 9-3. PIO Mode Waveform

Timing Parameter In PIO Mode

Table 9-3. Timing Parameter Each PIO Mode

PIO mode	PIO 0	PIO 1	PIO 2	PIO 3	PIO 4
T1	(70, --)	(50, --)	(30, --)	(30, --)	(25, --)
T2 (16-bit)	(165, --)	(125, --)	(100, --)	(80, --)	(70, --)
T2 Register (8-bit)	(290, --)	(290, --)	(290, --)	(80, --)	(70, --)
TEOC	(20, --)	(15, --)	(10, --)	(10, --)	(10, --)
T1 + T2 + TEOC	(600, --)	(383, --)	(240, --)	(180, --)	(120, --)

ATA_PIO_TIME (Tpara) = PIO mode(min, max) / system clock – 1

9.5.3 UDMA MODE

Direct mode and Indirect mode

Host can control device through EBI in indirect mode. If the IO voltage of external memory is not 3.3V, level shifter is required for interface signals of external devices. Level shifter requires a direction control bit for data bus because data is bidirectional signals. Two pins, XhiIRQn or XirSDBW, can be selected for a direction control bit. (These two pins are used as a control bit not only in UDMA mode but also in PC-CARD mode and PIO mode.) CF card or micro-drive can be connected directly to S3C6410X chip without being through memory port 0 in direct mode. There are multiplexed signals which refer to Table 9-1 direct mode column for direct mode.

UDMA-In Transfer (termination by device)

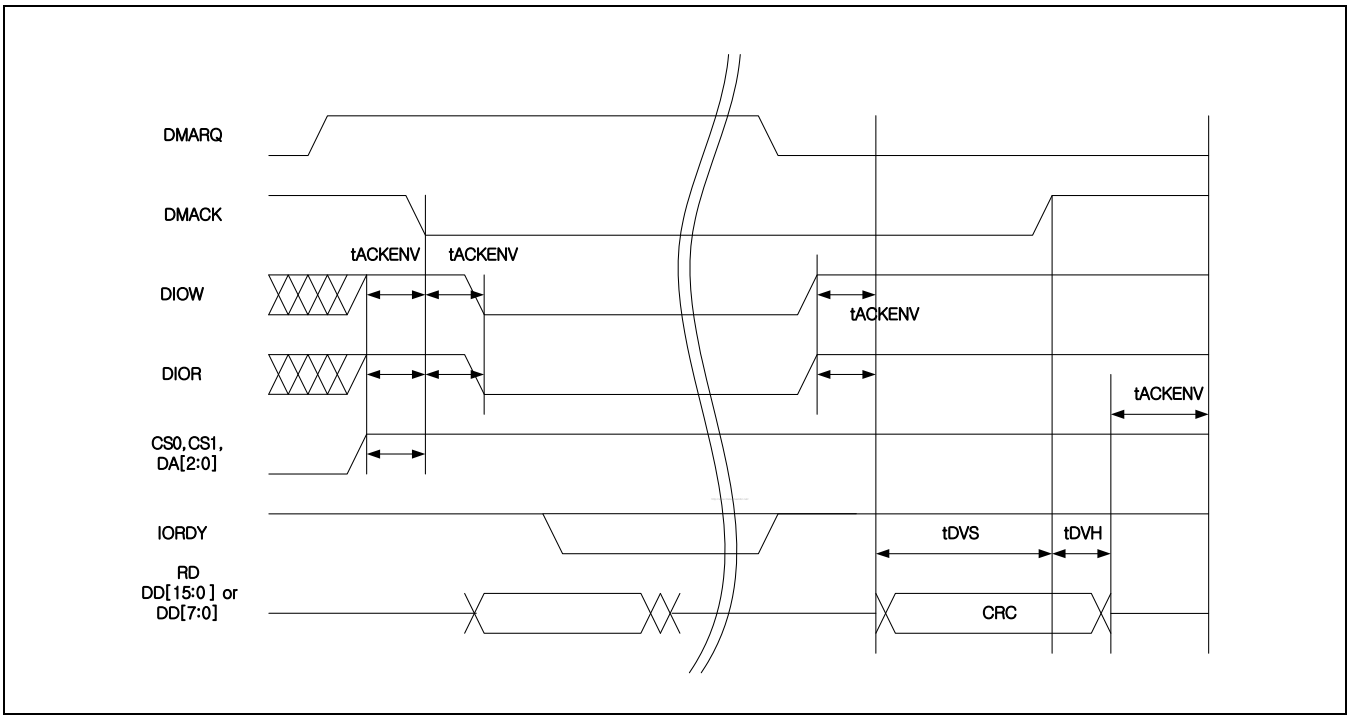


Figure 9-4. UDMA - In operation (terminated by device)

UDMA-In Transfer (termination by host)

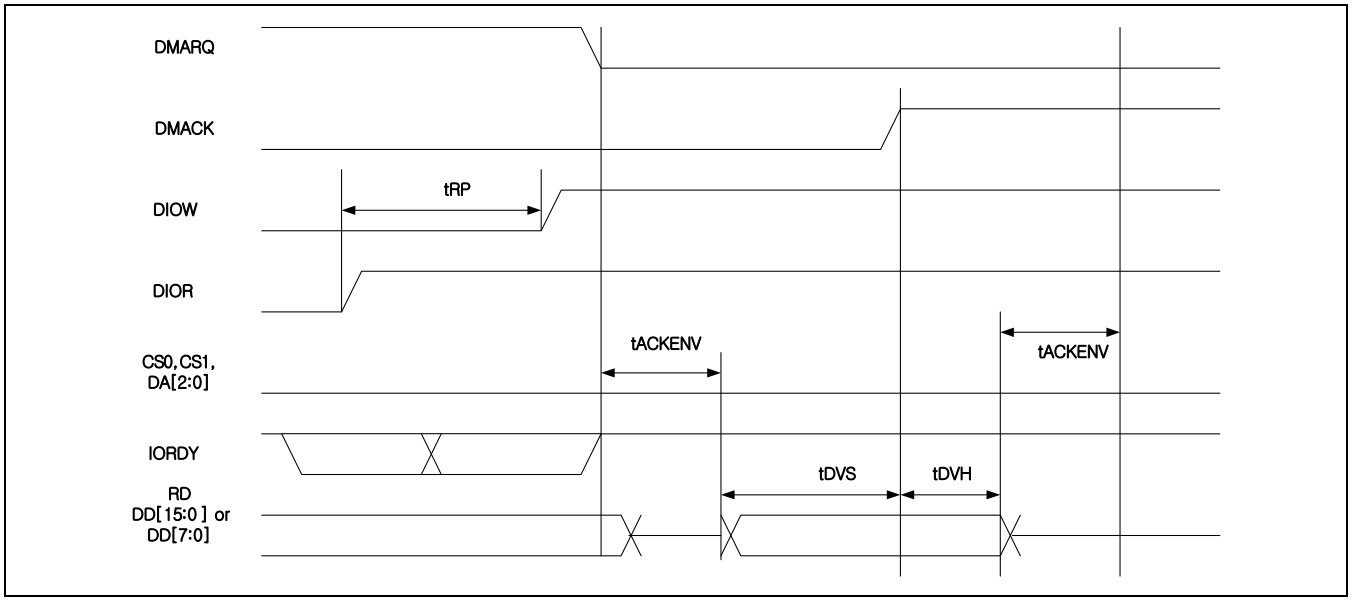


Figure 9-5. UDMA - In Operation (terminated by host)

UDMA-Out Transfer (termination by device)

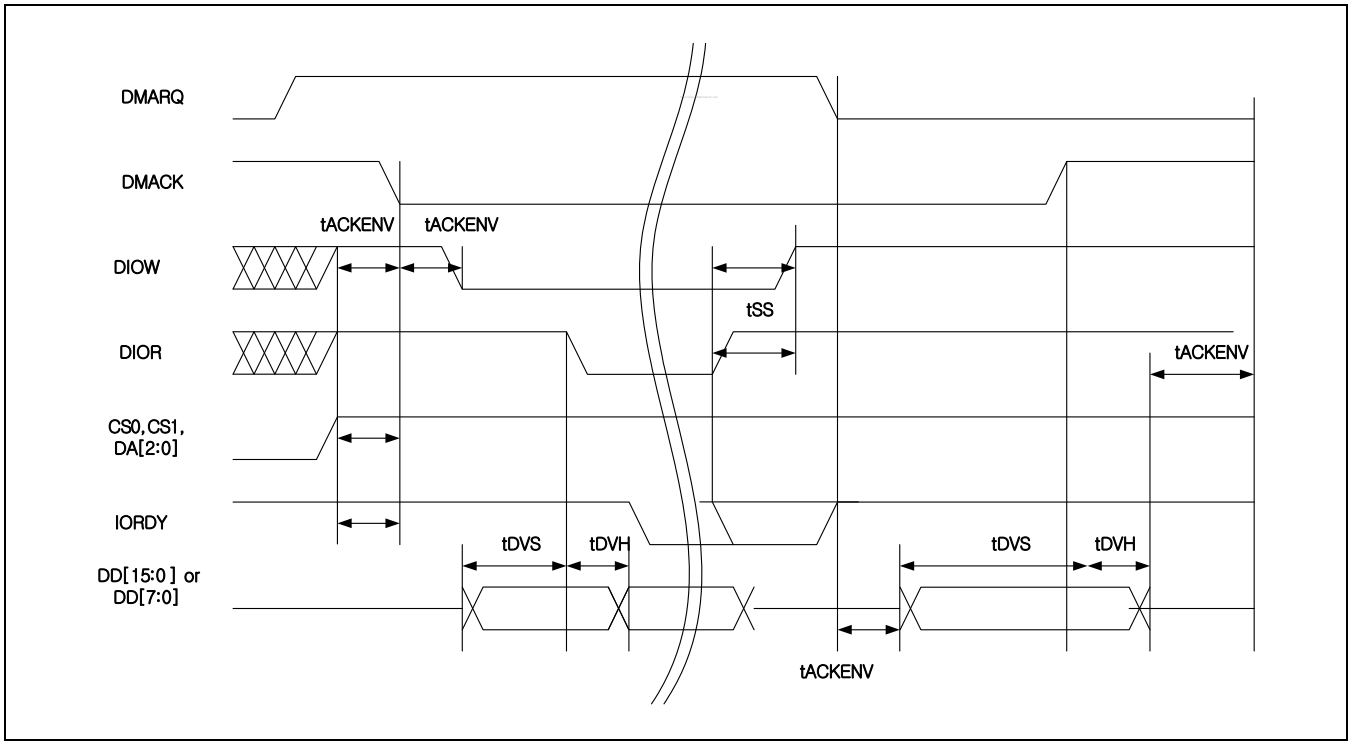


Figure 9-6. UDMA - Out Operation (terminated by device)

UDMA-Out Transfer (termination by host)

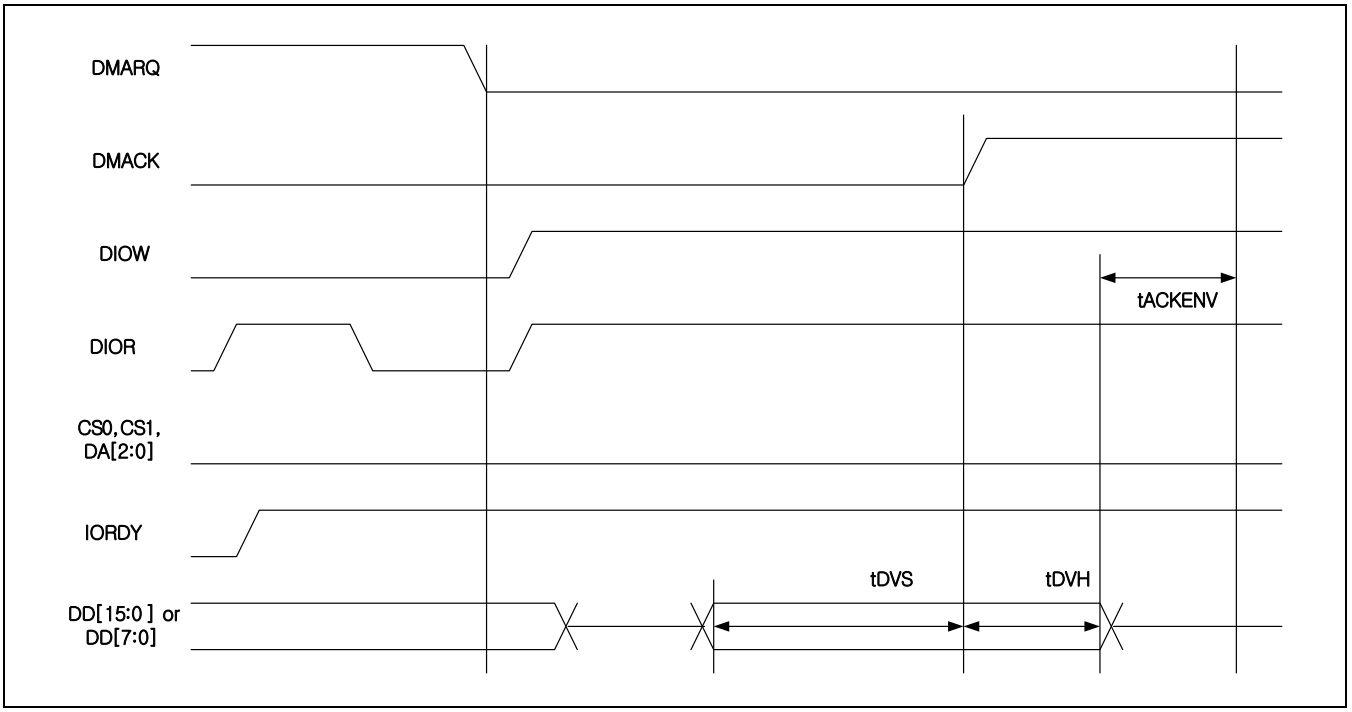


Figure 9-7. UDMA - Out Operation (terminated by host)

Table 9-4. 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)
tRP	(160, --)	(125, --)	(100, --)	(100, --)	(100, --)
tSS	(50, --)	(50, --)	(50, --)	(50, --)	(50, --)
tDVS	(70, --)	(48, --)	(31, --)	(20, --)	(6.7, --)
tDVH	(6.2, --)	(6.2, --)	(6.2, --)	(6.2, --)	(6.2, --)
tDVS+tDVH	(120, --)	(80, --)	(60, --)	(45, --)	(30, --)

ATA_UDMA_TIME (Tpara) = UDMA mode(min, max) / HCLK – 1

9.6 SPECIAL FUNCTION REGISTERS

9.6.1 MEMORY MAP

Memory Map Diagram (CFCON_Base = 0x7030_0000)

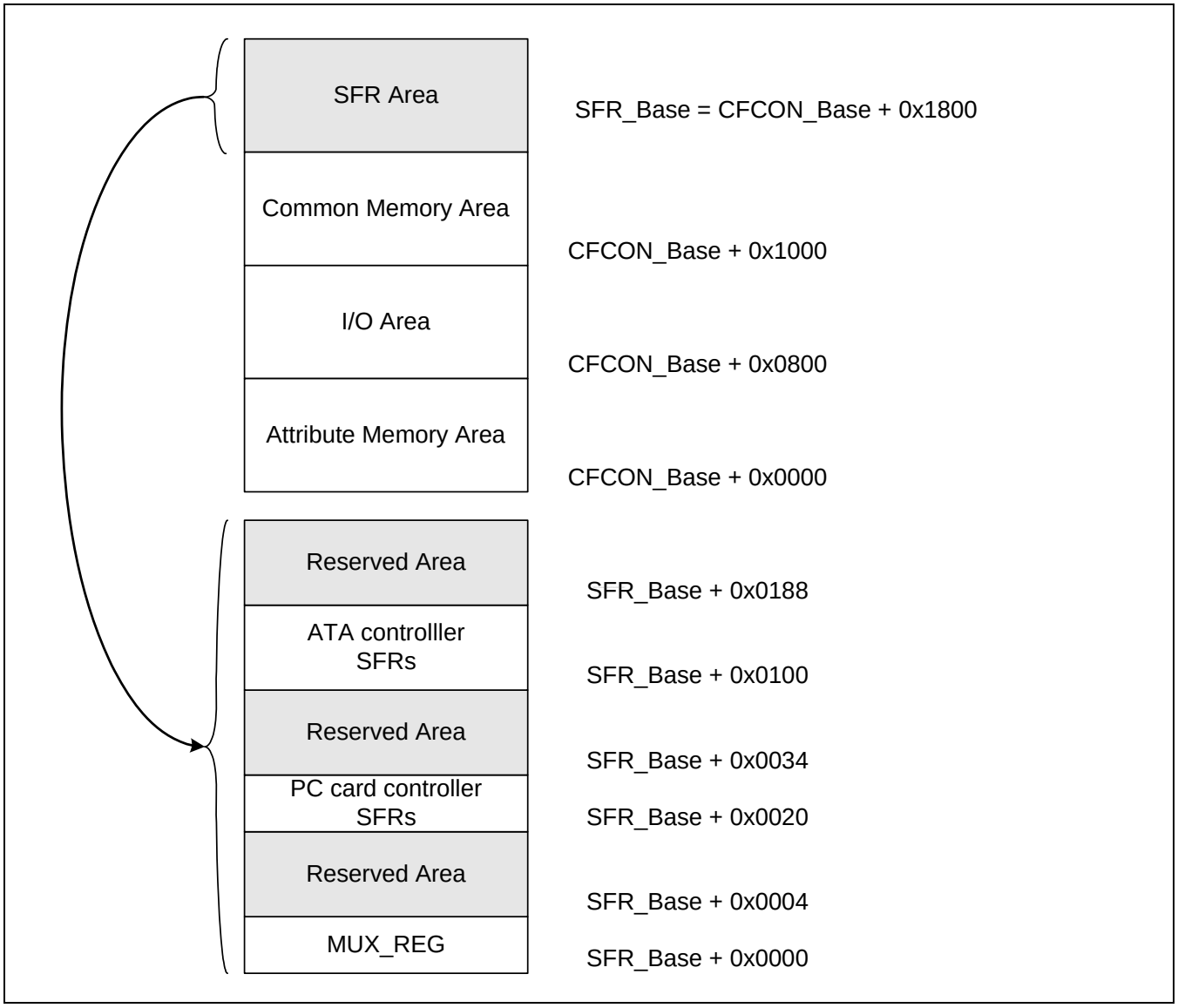


Figure 9-8. Memory Map Diagram

9.6.2 REGISTER ADDRESS TABLE

Table 9-5. Register Address Table

Register	Address	Description	Reset Value
SFR_BASE	0x70301800	CF card host controller base address	
MUX_REG	0x70301800	Top level control & configuration register	0x00000006
Reserved	~ 0x001C	Reserved area	
PCCARD_BASE	0x70301820	PC card controller base address	
PCCARD_ CNFG&STATUS	0x70301820	PC card configuration & status register	0x00000F07
PCCARD_ INTMSK&SRC	0x70301824	PC card interrupt mask & source regiseter	0x00000700
PCCARD_ATTR	0x70301828	PC card attribute memory area operation timing config regiseter	0x00031909
PCCARD_I/O	0x7030182C	PC card I/O area operation timing configuration regiseter	0x00031909
PCCARD_COMM	0x70301830	PC card common memory area operation timing config regiseter	0x00031909
Reserved	~ 0x00FC	Reserved area	
ATA_BASE	**0x70301900**	**ATA controller base address**	
ATA_CONTROL	0x70301900	ATA enable and clock down status	0x00000002
ATA_STATUS	0x70301904	ATA status	0x00000000
ATA_COMMAND	0x70301908	ATA command	0x00000000
ATA_SWRST	0x7030190C	ATA software reset	0x00000000
ATA_IRQ	0x70301910	ATA interrupt sources	0x00000000
ATA_IRQ_MASK	0x70301914	ATA interrut mask	0x0000001F
ATA_CFG	0x70301918	ATA configuration for ATA interface	0x00000000
Reserved	0x7030191C ~ 0x70301928	Reserved	
ATA_PIO_TIME	0x7030192C	ATA PIO timing	0x0001C238
ATA_UDMA_TIME	0x70301930	ATA UDMA timing	0x020b1362
ATA_XFR_NUM	0x70301934	ATA transfer number	0x00000000
ATA_XFR_CNT	0x70301938	ATA current transfer count	0x00000000
ATA_TBUF_START	0x7030193C	ATA start address of track buffer	0x00000000
ATA_TBUF_SIZE	0x70301940	ATA size of track buffer	0x00000000
ATA_SBUF_START	0x70301944	ATA start address of source buffer	0x00000000
ATA_SBUF_SIZE	0x70301948	ATA size of source buffer	0x00000000
ATA_CADR_TBUF	0x7030194C	ATA current write address of track buffer	0x00000000

Register	Address	Description	Reset Value
ATA_CADR_SBUF	0x70301950	ATA current read address of source buffer	0x00000000
ATA_PIO_DTR	0x70301954	ATA PIO device data register	0x00000000
ATA_PIO_FED	0x70301958	ATA PIO device Feature/Error register	0x00000000
ATA_PIO_SCR	0x7030195C	ATA PIO sector count register	0x00000000
ATA_PIO_LLR	0x70301960	ATA PIO device LBA low register	0x00000000
ATA_PIO_LMR	0x70301964	ATA PIO device LBA middle register	0x00000000
ATA_PIO_LHR	0x70301968	ATA PIO device LBA high register	0x00000000
ATA_PIO_DVR	0x7030196C	ATA PIO device register	0x00000000
ATA_PIO_CSD	0x70301970	ATA PIO device command/status register	0x00000000
ATA_PIO_DAD	0x70301974	ATA PIO device control/alternate status register	0x00000000
ATA_PIO_RDATA	0x7030197C	ATA PIO read data from device data register	0x00000000
ATA_FIFO_STATUS	0x70301994	ATA internal ATA FIFO status	0x00000000

9.6.3 INDIVIDUAL REGISTER DESCRIPTIONS

9.6.3.1 MUX_REG

Register	Address	Description	Reset Value
MUX_REG	0x70301800	MUX_REG is used to set the internal mode, output port enable & card power enable.	0x0000_0006

MUX_REG	Bit	Description	R/W	Reset Value
Reserved	[31:1]	Reserved bits		0x3
IDE_MODE	[0]	Internal operation mode select 0 : PC card mode 1 : True-IDE mode	R/W	0x0

9.6.3.2 PCCARD_CNFG&STATUS

Register	Address	Description	Reset Value
PCCARD_CNFG& STATUS	0x70301820	PCCARD_CNFG&STATUS is used to set the configuration & read the status of card.	0x0000_0F07

PCCARD_CNFG&STATUS	Bits	Description	R/W	Reset Value
Reserved	[31:14]	Reserved bits	R	0x0
CARD_RESET	[13]	CF card reset in PC card mode 0 : no reset 1 : reset	R/W	0x0
INT_SEL	[12]	Card interrupt request type(Device) select 0 : edge triggering 1 : level triggering User should use edge triggering except CF card which refers to use Level trigger for interrupt	R/W	0x0
nWAIT_EN	[11]	nWAIT(from CF card) enable 0 : disable(always ready) 1 : enable	R/W	0x1
DEVICE_ATT	[10]	Device type is 16bits or 8bits (Attribute memory area) 0 : 8-bit device 1 : 16-bit device	R/W	0x1
DEVICE_COMM	[9]	Device type is 16bits or 8bits (Common memory area) 0 : 8-bit device 1 : 16-bit device	R/W	0x1
DEVICE_IO	[8]	Device type is 16bits or 8bits (I/O area) 0 : 8-bit device 1 : 16-bit device	R/W	0x1
Reserved	[7:4]	Reserved bits	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]	Card detect 0 : card detect 1 : card not detect	R	0x1

9.6.3.3 PCCARD_INTMSK&SRC

Register	Address	Description	Reset Value
PCCARD_INTMSK &SRC	0x70301824	PCCARD_INTMSK&SRC is interrupt source & interrupt mask register.	0x0000_0700

PCCARD_INTMSK &SRC	Bits	Description	R/W	Reset Value
Reserved	[31:11]	Reserved bits	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 bits	R	0x0
INTSRC_ERR_N	[2]	When host access no card in slot. CPU can clear this interrupt by writing "1".	R/W	0x0
INTSRC_IREQ	[1]	When CF card interrupt request CPU can clear this interrupt by writing "1".	R/W	0x0
INTSRC_CD	[0]	When CF card is detected in slot CPU can clear this interrupt by writing "1".	R/W	0x0

9.6.3.4 PCCARD_ATTR

Register	Address	Description	Reset Value
PCCARD_ATTR	0x70301828	PCCARD_ATTR is used to set the card access timing.	0x0003_1909

PCCARD_ATTR	Bits	Description	R/W	Reset Value
Reserved	[31:23]	Reserved bits	R	0x0
HOLD_ATTR	[22:16]	Hold state timing of attribute memory area Hold time = HCLK period * (HOLD_ATTR + 1)	R/W	0x03
Reserved	[15]	Reserved bits	R	0x0
CMND_ATTR	[14:8]	Command state timing of attribute memory area Command time = HCLK period * (CMND_ATTR + 1)	R/W	0x19
Reserved	[7]	Reserved bits	R	0x0
SETUP_ATTR	[6:0]	Setup state timing of attribute memory area Setup time = HCLK period * (SETUP_ATTR + 1)	R/W	0x09

9.6.3.5 PCCARD_I/O

Register	Address	Description	Reset Value
PCCARD_I/O	0x7030182C	PCCARD_I/O is used to set the card access timing.	0x0003_1909

PCCARD_I/O	Bits	Description	R/W	Reset Value
Reserved	[31:23]	Reserved bits	R	0x0
HOLD_IO	[22:16]	Hold state timing of I/O area Hold time = HCLK period * (HOLD_IO + 1)	R/W	0x03
Reserved	[15]	Reserved bits	R	0x0
CMND_IO	[14:8]	Command state timing of I/O area Command time = HCLK period * (CMND_IO + 1)	R/W	0x19
Reserved	[7]	Reserved bits	R	0x0
SETUP_IO	[6:0]	Setup state timing of I/O area Setup time = HCLK period * (SETUP_IO + 1)	R/W	0x09

9.6.3.6 PCCARD_COMM

Register	Address	Description	Reset Value
PCCARD_COMM	0x70301830	PCCARD_COMM is used to set the card access timing.	0x0003_1909

PCCARD_COMM	Bits	Description	R/W	Reset Value
Reserved	[31:23]	Reserved bits	R	0x0
HOLD_COMM	[22:16]	Hold state timing of common memory area Hold time = HCLK period * (HOLD_COMM + 1)	R/W	0x03
Reserved	[15]	Reserved bits	R	0x0
CMND_COMM	[14:8]	Command state timing of common memory area Command time = HCLK period * (CMND_COMM + 1)	R/W	0x19
Reserved	[7]	Reserved bits	R	0x0
SETUP_COMM	[6:0]	Setup state timing of common memory area Setup time = HCLK period * (SETUP_COMM + 1)	R/W	0x09

9.6.3.7 ATA_CONTROL

Register	Address	Description	Reset Value
ATA_CONTROL	0x70301900	ATA enable and clock down status	0x0000_0002

ATA_CONTROL	Bits	Description	R/W	Reset Value
Reserved	[31:2]	Reserved bits	R	0x0
CLK_DOWN_READY	[1]	Status for clock down This bit is asserted in idle state when ATA_CONTROL bit [0] is zero. 0 : not ready for clock down 1 : ready for clock down	R	0x1
ATA_ENABLE	[0]	ATA enable 0 : ATA is disabled and preparation for clock down maybe in progress 1 : ATA is enabled. When this value is set to 1, delay of 200ms will be required.	R/W	0x0

9.6.3.8 ATA_STATUS

Register	Address	Description	Reset Value
ATA_STATUS	0x70301904	ATA controller status	0x0000_0000

ATA_STATUS	Bits	Description	R/W	Reset Value
Reserved	[31:5]	Reserved bits	R	0x0
ATADEV_IRQ	[4]	ATA interrupt signal line	R	0x0
ATADEV_IORDY	[3]	ATA iordy signal line	R	0x0
ATADEV_DMAREQ	[2]	ATA dmareq signal line	R	0x0
XFR_STATE	[1:0]	Transfer state 2'b00 : idle state 2'b01 : transfer state 2'b11 : wait for completion state	R	0x0

9.6.3.9 ATA_COMMAND

Register	Address	Description	Reset Value
ATA_COMMAND	0x70301908	ATA command	0x0000_0000

ATA_COMMAND	Bits	Description	R/W	Reset Value
Reserved	[31:2]	Reserved bits	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 is used to start data transfer. The "STOP" command can pause transfer temporarily. The "CONTINUE" command shall be used after "STOP" command or internal state of "pause" when 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 : command stop 01 : command start (Only available in idle state) 10 : command abort 11 : command continue (Only available in transfer pause) ** After issuing the ABORT command, make software reset by setting ATA_SWRST [0] to clear the leftover values of internal registers.	R/W	0x0

The STOP command is used when CPU wants to pause data transfer. The CPU uses STOP command to judge the transmission data is valid or not while transfer transmits.

To resume the data transfer use CONTINUE command.

The STOP command does control 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 is empty at read operation. In case of write operation, the DMA acts the same way until the FIFO is full.

The ABORT command is used when the transmitting data has proved useless data. Also, this command discontinues absurd state by error interrupt from device.

At that time, all data in ATA Host controller (register, FIFO) cleared and the transmission state machine switches to IDLE.

The Software Reset's meaning become clear all registers irrespective of the ABORT command being executed before do configuration register set for next transmission. It is not mandatory.

9.6.3.10 ATA_SWRST

Register	Address	Description	Reset Value
ATA_SWRST	0x7030190C	ATA software reset	0x0000_0000

ATA_SWRST	Bits	Description	R/W	Reset Value
Reserved	[31:1]	Reserved bits	R	0x0
ATA_SWRSTN	[0]	Software reset for the ATA host 0: No reset 1: Software reset for all ATA host module. After software reset, to continue transfer, you must configure all registers of host controller and device registers.	R/W	0x0

9.6.3.11 ATA_IRQ

Register	Address	Description	Reset Value
ATA_IRQ	0x70301910	ATA interrupt source	0x0000_0000

ATA_IRQ	Bits	Description	R/W	Reset Value
Reserved	[31:5]	Reserved bits	R	0x0
SBUF_EMPTY_INT	[4]	When source buffer is empty. CPU can clear this interrupt by writing "1".	R/W	0x0
TBUF_FULL_INT	[3]	When track buffer is half full. CPU can clear this interrupt by writing "1".	R/W	0x0
ATADEV_IRQ_INT	[2]	When ATA device generates interrupt. CPU can clear this interrupt by writing "1".	R/W	0x0
UDMA_HOLD_INT	[1]	When ATA device makes early termination in UDMA class. CPU can clear this interrupt by writing "1".	R/W	0x0
XFR_DONE_INT	[0]	When all data transfers are finished. CPU can clear this interrupt by writing "1".	R/W	0x0

NOTES:

1. All interrupts from ATA interface are level-triggered. Therefore, IRQ clear operation is necessary when driver is implemented.
2. In DMA mode, XFR_DONE_INT must be used to check the DMA transfer done. When XFR_DONE_INT occurs, ATA_STATUS[1:0] must be idle state(2'b00). Otherwise, delay will be required until ATA_STATUS[1:0] is set to idle state(2'b00) by hardware.

9.6.3.12 ATA_IRQ_MASK

Register	Address	Description	Reset Value
ATA_IRQ_MASK	0x70301914	ATA interrupt mask	0x0000_001F

ATA_IRQ_MASK	Bits	Description	R/W	Reset Value
Reserved	[31:2]	Reserved bits	R	0x0
MASK_SBUT_EMPTY_INT	[4]	Interrupt mask bit of source buffer empty 0 : unmask 1 : mask	R/W	0x1
MASK_TBUF_FULL_INT	[3]	Interrupt mask bit of target buffer full 0 : unmask 1 : mask	R/W	0x1
MASK_ATADEV_IRQ_INT	[2]	Interrupt mask bit of ATA device interrupt request 0 : unmask 1 : mask	R/W	0x1
MASK_UDMA_HOLD_INT	[1]	Interrupt mask bit of UDMA hold 0 : unmask 1 : mask	R/W	0x1
MASK_XFR_DONE_INT	[0]	Interrupt mask bit of xfr done 0 : unmask 1 : mask	R/W	0x1

9.6.3.13 ATA_CFG

Register	Address	Description	Reset Value
ATA_CFG	0x70301918	ATA configuration for ATA interface	0x0000_0000

ATA_CFG	Bits	Description	R/W	Reset Value
Reserved	[31:10]	Reserved bits	R	0x0
UDMA_AUTO _MODE	[9]	Determines whether to continue automatically in case of early termination in UDMA mode by Device. This bit must not be changed during runtime operation. 0: stay in pause state and wait for CPU's action. 1: continue automatically	R/W	0x0
SBUF_EMPTY _MODE	[8]	Determines whether to continue automatically when source buffer is empty. This bit must not be changed during runtime operation. 0: continue automatically with new source buffer address. 1: stay in pause state and wait for CPU's action. ** With the SBUF_EMPTY_MODE is "0" and the transmission data size is bigger than the source buffer size, the source buffer empty interrupt (SBUF_EMPTY_INT) happens before setting of the second source buffer base address and size. Then ATA host controller brings data from the first source buffer repeatedly. To avoid this, after 1st source buffer is empty, the "SBUF_EMPTY_MODE" bit automatically changes to HIGH even though the default is "0". Therefore you must issue "CONTINUE" command. If you don't want CPU to interfere, change the next source buffer address to "0" at the bit 8 before/after the next base address and size.	R/W	0x0
TBUF_FULL _MODE	[7]	Determines whether to continue automatically when track buffer is full. This bit must not be changed during runtime operation. 0: continue automatically with new track buffer address. 1: stay in pause state and wait for CPU's action. ** With the TBUF_FULL_MODE is "0" and the transmission data size is bigger than the target buffer size, the target buffer full interrupt(TBUF_FULL_INT) happens before setting of the second target buffer base address and size. Then ATA host controller sends data to the first target buffer repeatedly. To avoid this, after 1st target buffer is full, the "TBUF_FULL_MODE" bit automatically changes to HIGH even though the default is "0". Therefore you must issue "CONTINUE" command. If you don't want CPU to interfere, change the next source buffer address to "0" at the bit 8 before/after the next base address and size.	R/W	0x0

ATA_CFG	Bits	Description	R/W	Reset Value
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]	ATA transfer class select 2'b00 : transfer class is PIO 2'b01 : transfer class is PIO DMA 2'b1x : transfer class is UDMA	R/W	0x0
ATA_IORDY_EN	[1]	Determines whether IORDY input can extend data transfer. 0 : IORDY disable(ignored) 1 : IORDY enable (can extend)	R/W	0x0
ATA_RST	[0]	ATA device reset by this host. 0 : no reset 1 : reset	R/W	0x0

9.6.3.14 ATA_PIO_TIME

Register	Address	Description	Reset Value
ATA_PIO_TIME	0x7030192C	ATA PIO timing	0x0001_C238

ATA_PIO_TIME	Bits	Description	R/W	Reset Value
Reserved	[31:20]	Reserved bits	R	0x0
PIO_TEOC	[19:12]	PIO timing parameter, teoc, end of cycle time It cannot have zero value. teoc = HCLK period * (pio_teoc + 1)	R/W	0x1C
PIO_T2	[11:4]	PIO timing parameter, t2, DIOR/Wn pulse width It cannot have zero value. t2 = HCLK period * (pio_t2 + 1)	R/W	0x23
PIO_T1	[3:0]	PIO timing parameter, t1, address valid to DIOR/Wn t1 = HCLK period * (pio_t1 + 1)	R/W	0x8

9.6.3.15 ATA_UDMA_TIME

Register	Address	Description	Reset Value
ATA_UDMA_TIME	0x70301930	ATA UDMA timing	0x020b_1362

ATA_UDMA_TIME	Bits	Description	R/W	Reset Value
Reserved	[31:28]	Reserved bits	R	0x0
UDMA_TDVH	[27:24]	UDMA timing parameter tDVH tDVH = HCLK period* (UDMA_TDVH + 1)	R/W	0x2
UDMA_TDVS	[23:16]	UDMA timing parameter tDVS It shall not have zero value. tDVS = HCLK period * (UDMA_TDVS + 1)	R/W	0x0B
UDMA_TRP	[15:8]	UDMA timing parameter tRP tRP = HCLK period * (UDMA_TRP + 1)	R/W	0x13
UDMA_TSS	[7:4]	UDMA timing parameter, tSS tSS = HCLK period * (UDMA_TSS + 1)	R/W	0x6
UDMA_TACKENV	[3:0]	UDMA timing parameter tENV(envelope time) (from DMACKn to STOP and HDMARDYn), tACK(setup and hold time for DMACKn) tENV = HCLK period * (UDMA_TACKENV + 1)	R/W	0x2

9.6.3.16 ATA_XFR_NUM

Register	Address	Description	Reset Value
ATA_XFR_NUM	0x70301934	ATA transfer number	0x0000_0000

ATA_XFR_NUM	Bits	Description	R/W	Reset Value
XFR_NUM	[31:1]	Data transfer number.	R/W	0x0000_0000
Reserved	[0]	Reserved bits	R	0x0

9.6.3.17 ATA_XFR_CNT

Register	Address	Description	Reset Value
ATA_XFR_CNT	0x70301938	ATA current transfer count	0x0000_0000

ATA_XFR_CNT	Bits	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 when pre-defined all data has been transferred.	R	0x0000_0000
Reserved	[0]	Reserved bits	R	0x0

9.6.3.18 ATA_TBUF_START

Register	Address	Description	Reset Value
ATA_TBUF_START	0x7030193C	ATA start address of track buffer	0x0000_0000

ATA_TBUF_START	Bits	Description	R/W	Reset Value
TRACK_BUFFER_START	[31:0]	Start address of track buffer (4byte unit address)	R/W	0x00000000

9.6.3.19 ATA_TBUF_SIZE

Register	Address	Description	Reset Value
ATA_TBUF_SIZE	0x70301940	ATA size of track buffer	0x0000_0000

ATA_TBUF_SIZE	Bits	Description	R/W	Reset Value
TRACK_BUFFER_SIZE	[31:0]	Size of track buffer (32byte unit size). This must be set to "size_of_data_in_bytes – 1". For example, to transfer 1-sector (512-byte, 32'h200), you must set 32'h1FF (= 32'h200 – 1).	R/W	0x0000000

9.6.3.20 ATA_SBUF_START

Register	Address	Description	Reset Value
ATA_SBUF_START	0x70301944	ATA start address of source buffer	0x0000_0000

Name	Bits	Description	R/W	Reset Value
SRC_BUFFER_START	[31:0]	Start address of source buffer (4byte unit address)	R/W	0x0000_0000

9.6.3.21 ATA_SBUF_SIZE

Register	Address	Description	Reset Value
ATA_SBUF_SIZE	0x70301948	ATA size of source buffer	0x0000_0000

ATA_SBUF_SIZE	Bits	Description	R/W	Reset Value
SRC_BUFFER_SIZE	[31:0]	Size of source buffer (32byte unit size). This must be set to "size_of_data_in_bytes – 1". For example, to transfer 1-sector (512-byte, 32'h200), you must set 32'h1FF (= 32'h200 – 1).	R/W	0x0000000

9.6.3.22 ATA_CADDR_TBUR

Register	Address	Description	Reset Value
ATA_CADDR_TBUF	0x7030194C	ATA current write address of track buffer	0x0000_0000

ATA_CADDR_TBUF	Bits	Description	R/W	Reset Value
track_buf_cur_adr	[31:0]	Current address of track buffer (4byte unit address)	R	0x00000000

9.6.3.23 ATA_CADDR_SBUF

Register	Address	Description	Reset Value
ATA_CADDR_SBUF	0x70301950	ATA current read address of source buffer	0x0000_0000

ATA_CADDR_SBUF	Bits	Description	R/W	Reset Value
src_buf_cur_adr	[31:0]	Current address of source buffer (4byte unit address)	R	0x00000000

9.6.3.24 ATA_PIO_DTR

Register	Address	Description	Reset Value
ATA_PIO_DTR	0x70301954	ATA PIO device data register	0x0000_0000

ATA_PIO_DTR	Bits	Description	R/W	Reset Value
Reserved	[31:16]	Reserved bits	R	0x0
PIO_DEV_DTR*	[15:0]	16-bit PIO data register	W	0x0000

NOTE: PIO_DEV_DTR can be read by accessing register ATA_PIO_RDATA

9.6.3.25 ATA_PIO_FED

Register	Address	Description	Reset Value
ATA_PIO_FED	0x70301958	ATA PIO device Feature/Error register	0x0000_0000

ATA_PIO_FED	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_FED	[7:0]	8-bit PIO device feature/error (command block) register	W	0x00

NOTE: PIO_DEV_FED can be read by accessing register ATA_PIO_RDATA

9.6.3.26 ATA_PIO_SCR

Register	Address	Description	Reset Value
ATA_PIO_SCR	0x7030195C	ATA PIO sector count register	0x0000_0000

ATA_PIO_SCR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_SCR	[7:0]	8-bit PIO device sector count (command block) register	W	0x00

NOTE: PIO_DEV_SCR can be read by accessing register ATA_PIO_RDATA.

9.6.3.27 ATA_PIO_LLR

Register	Address	Description	Reset Value
ATA_PIO_LLR	0x70301960	ATA PIO device LBA low register	0x0000_0000

ATA_PIO_LLR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_LLR	[7:0]	8-bit PIO device LBA low (command block) register	W	0x00

NOTE: PIO_DEV_LLR can be read by accessing register ATA_PIO_RDATA.

9.6.3.28 ATA_PIO_LMR

Address = 0x70301964

Register	Address	Description	Reset Value
ATA_PIO_LMR	0x70301964	ATA PIO device LBA middle register	0x0000_0000

ATA_PIO_LMR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_LMR	[7:0]	8-bit PIO device LBA middle (command block) register	W	0x00

NOTE: PIO_DEV_LMR can be read by accessing register ATA_PIO_RDATA.

9.6.3.29 ATA_PIO_LHR

Register	Address	Description	Reset Value
ATA_PIO_LHR	0x70301968	ATA PIO device LBA high register	0x0000_0000

ATA_PIO_LHR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_LHR	[7:0]	8-bit PIO LBA high (command block) register	W	0x00

NOTE: PIO_DEV_LHR can be read by accessing register ATA_PIO_RDATA

9.6.3.30 ATA_PIO_DVR

Register	Address	Description	Reset Value
ATA_PIO_DVR	0x7030196C	ATA PIO device register	0x0000_0000

ATA_PIO_DVR	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_DVR	[7:0]	8-bit PIO device (command block) register	W	0x00

NOTE: PIO_DEV_DVR can be read by accessing register ATA_PIO_RDATA.

9.6.3.31 ATA_PIO_CSD

Register	Address	Description	Reset Value
ATA_PIO_CSD	0x70301970	ATA PIO device command/status register	0x0000_0000

ATA_PIO_CSD	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_CSD	[7:0]	8-bit PIO device command/status (command block) register	W	0x00

NOTE: PIO_DEV_CSD can be read by accessing register ATA_PIO_RDATA.

9.6.3.32 ATA_PIO_DAD

Register	Address	Description	Reset Value
ATA_PIO_DAD	0x70301974	ATA PIO device control/alternate status register	0x0000_0000

ATA_PIO_DAD	Bits	Description	R/W	Reset Value
Reserved	[31:8]	Reserved bits	R	0x0
PIO_DEV_DAD	[7:0]	8-bit PIO device control/alternate status (control block) register	W	0x00

NOTE: PIO_DEV_DAD can be read by accessing register ATA_PIO_RDATA.

9.6.3.33 ATA_PIO_RDATA

Register	Address	Description	Reset Value
ATA_PIO_RDATA	0x7030197C	ATA PIO read data from device data register	0x0000_0000

ATA_PIO_RDATA	Bits	Description	R/W	Reset Value
Reserved	[31:16]	Reserved bits	R	0x0
PIO_RDATA	[15:0]	PIO read data register while HOST read from ATA device register	R	0x0000

NOTE. When you want to read ATA_PIO_XXX (such as ATA_PIO_DTR, ATA_PIO_FED and so on), you may access ATA_PIO_XXX directly. However you may only get garbage data. If you want to get correct data, check the ATA_STATE at ATA_FIFO_STATUS register after accessing ATA_PIO_XXX until you get the value as IDLE state. Since then you can get the correct value of ATA_PIO_XXX by reading ATA_PIO_RDATA.

9.6.3.34 BUS_FIFO_STATUS

Register	Address	Description	Reset Value
BUS_FIFO_STATUS	0x70301990	ATA internal AHB FIFO status	0x0000_0000

BUS_FIFO_STATUS	Bits	Description	R/W	Reset Value
Reserved	[31:19]	Reserved bits	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	R	0x00
Reserved	[15:14]	Reserved bits	R	0x0
BUS_FIFO_RDPNT	[13:8]	bus fifo read pointer	R	0x00
Reserved	[7:6]	Reserved bits	R	0x0
BUS_FIFO_WRPNT	[5:0]	bus fifo write pointer	R	0x00

9.6.3.35 ATA_FIFO_STATUS

Register	Address	Description	Reset Value
ATA_FIFO_STATUS	0x70301994	ATA internal ATA FIFO status	0x0000_0000

ATA_FIFO_STATUS	Bits	Description	R/W	Reset Value
Reserved	[31]	Reserved bit	R	0x0
ATA_STATE	[30:28]	3'b000 : IDLE Others : reserved Refer following NOTE.	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	[23:21]	Reserved bits	R	0x0
UDMA_STATE	[20:16]	5'b00000 : IDLE 5'b00100 : END Another value is in UDMA operation.	R	0x00
Reserved	[15:0]	Reserved bits	R	0x0

NOTES

10 GPIO

10.1 OVERVIEW

S3C6410 includes 187multi-functional input/output port pins. There are 17 ports as listed below:

PortName	Number of Pins.	Muxed pins	Power Inform.
GPA port	8	UART/EINT	1.8~3.3V
GPB port	7	UART/IrDA/I2C/CF/Ext.DMA/EINT	1.8~3.3V
GPC port	8	SPI/SDMMC/I2S_V40/EINT	1.8~3.3V
GPD port	5	PCM/I2S/AC97/EINT	1.8~3.3V
GPE port	5	PCM/I2S/AC97	1.8~3.3V
GPF port	16	CAMIF/PWM/EINT	1.8~3.3V
GPG port	7	SDMMC/EINT	1.8~3.3V
GPH port	10	SDMMC/KEYPAD/CF/I2S_V40/EINT	1.8~3.3V
GPI port	16	LCD	1.8~3.3V
GPJ port	12	LCD	1.8~3.3V
GPK port	16	HostIF/HIS/KEYPAD/CF	1.8~3.3V
GPL port	15	HostIF/KEYPAD/CF/OTG/EINT	1.8~3.3V
GPM port	6	HostIF/CF/EINT	1.8~3.3V
GPN port	16	EINT/KEYPAD	1.8~3.3V
GPO port	16	MemoryPort0/EINT	1.8~3.3V
GPP port	15	MemoryPort0/EINT	1.8~3.3V
GPQ port	9	MemoryPort0/EINT	1.8~3.3V

10.2 FEATURES

The GPIO provides the following features:

- Controls 127 External Interrupts
- 187 multi-functional input/output ports
- Controls pin states in Sleep Mode except GPK, GPL, GPM, and GPN

10.3 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 can keep their values during sleep mode.

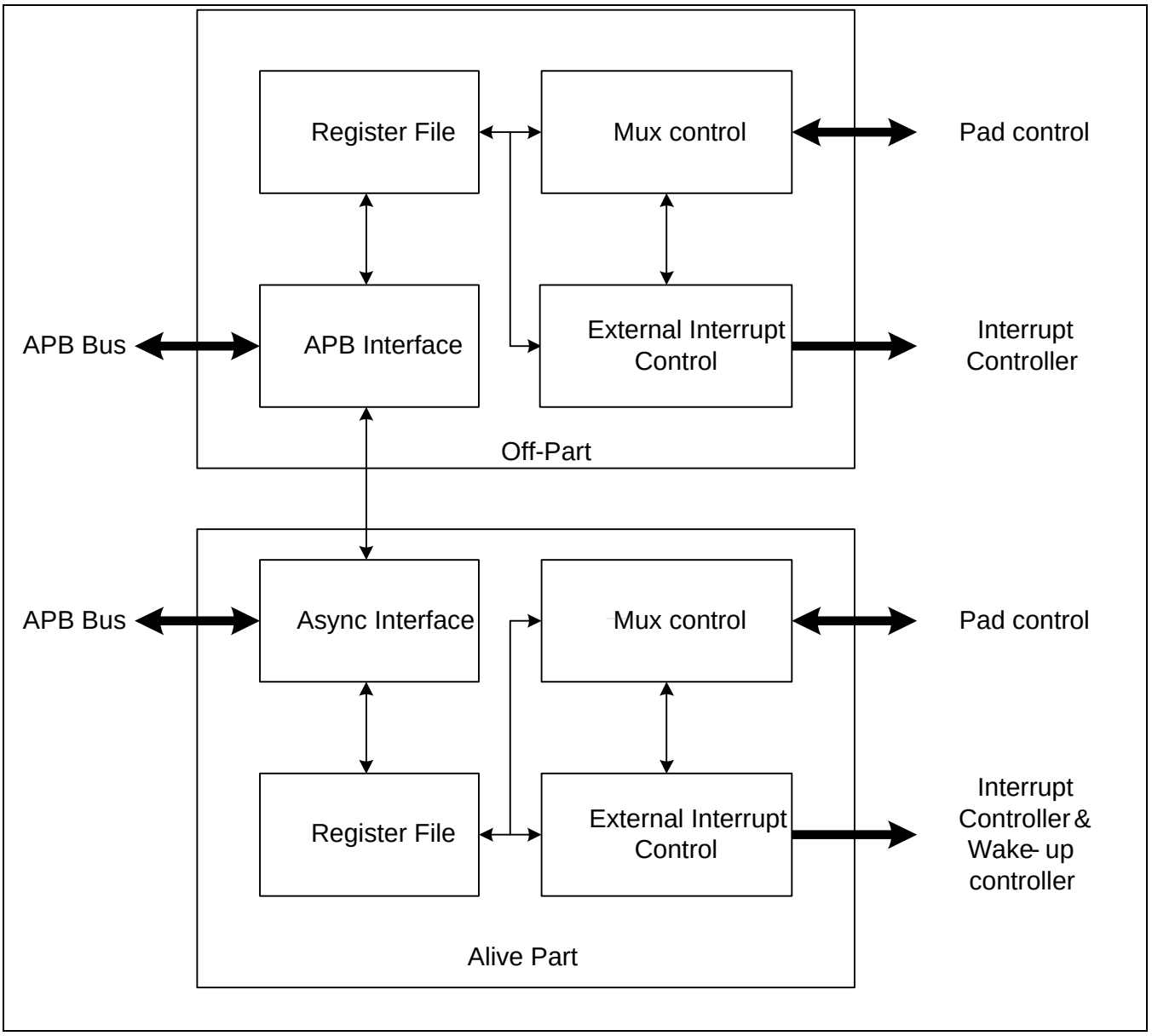


Figure 10-1. GPIO block diagram

10.4 REGISTER DESCRIPTION

10.4.1 MEMORY MAP

Register	Address	R/W	Description	Reset Value
GPACON	0x7F008000	R/W	Port A Configuration Register	0x0
GPADAT	0x7F008004	R/W	Port A Data Register	Undefined
GPAPUD	0x7F008008	R/W	Port A Pull-up/down Register	0x00005555
GPACONSLP	0x7F00800C	R/W	Port A Sleep mode Configuration Register	0x0
GPAPUDSLP	0x7F008010	R/W	Port A Sleep mode Pull-up/down Register	0x0
GPBCON	0x7F008020	R/W	Port B Configuration Register	0x40000
GPBDAT	0x7F008024	R/W	Port B Data Register	Undefined
GPBPUD	0x7F008028	R/W	Port B Pull-up/down Register	0x00001555
GPBCONSLP	0x7F00802C	R/W	Port B Sleep mode Configuration Register	0x0
GPBPUDSLP	0x7F008030	R/W	Port B Sleep mode Pull-up/down Register	0x0
GPCCON	0x7F008040	R/W	Port C Configuration Register	0x0
GPCDAT	0x7F008044	R/W	Port C Data Register	Undefined
GPCPUD	0x7F008048	R/W	Port C Pull-up/down Register	0x00005555
GPCCONSLP	0x7F00804C	R/W	Port C Sleep mode Configuration Register	0x0
GPCPUDSLP	0x7F008050	R/W	Port C Sleep mode Pull-up/down Register	0x0
GPDCON	0x7F008060	R/W	Port D Configuration Register	0x0
GPDDAT	0x7F008064	R/W	Port D Data Register	Undefined
GPDPUD	0x7F008068	R/W	Port D Pull-up/down Register	0x00000155
GPDCONSLP	0x7F00806C	R/W	Port D Sleep mode Configuration Register	0x0
GPDPUDSLP	0x7F008070	R/W	Port D Sleep mode Pull-up/down Register	0x0
GPECON	0x7F008080	R/W	Port E Configuration Register	0x0
GPEDAT	0x7F008084	R/W	Port E Data Register	Undefined
GPEPUD	0x7F008088	R/W	Port E Pull-up/down Register	0x00000155
GPECONSLP	0x7F00808C	R/W	Port E Sleep mode Configuration Register	0x0
GPEPUDSLP	0x7F008090	R/W	Port E Sleep mode Pull-up/down Register	0x0
GPFCON	0x7F0080A0	R/W	Port F Configuration Register	0x0
GPFDAT	0x7F0080A4	R/W	Port F Data Register	Undefined
GPFPUD	0x7F0080A8	R/W	Port F Pull-up/down Register	0x55555555
GPFCONSLP	0x7F0080AC	R/W	Port F Sleep mode Configuration Register	0x0
GPFPUDSLP	0x7F0080B0	R/W	Port F Sleep mode Pull-up/down Register	0x0

Register	Address	R/W	Description	Reset Value
GPGCON	0x7F0080C0	R/W	Port G Configuration Register	0x0
GPGDAT	0x7F0080C4	R/W	Port G Data Register	Undefined
GPGPUD	0x7F0080C8	R/W	Port G Pull-up/down Register	0x00001555
GPGCONSLP	0x7F0080CC	R/W	Port G Sleep mode Configuration Register	0x0
GPGPUDSLP	0x7F0080D0	R/W	Port G Sleep mode Pull-up/down Register	0x0
GPHCON0	0x7F0080E0	R/W	Port H Configuration Register	0x0
GPHCON1	0x7F0080E4	R/W	Port H Configuration Register	0x0
GPHDAT	0x7F0080E8	R/W	Port H Data Register	Undefined
GPHPUD	0x7F0080EC	R/W	Port H Pull-up/down Register	0x00055555
GPHCONSLP	0x7F0080F0	R/W	Port H Sleep mode Configuration Register	0x0
GPHPUDSLP	0x7F0080F4	R/W	Port H Sleep mode Pull-up/down Register	0x0
GPICON	0x7F008100	R/W	Port I Configuration Register	0x0
GPIDAT	0x7F008104	R/W	Port I Data Register	Undefined
GPIPUD	0x7F008108	R/W	Port I Pull-up/down Register	0x55555555
GPICONSLP	0x7F00810C	R/W	Port I Sleep mode Configuration Register	0x0
GPIPUDSLP	0x7F008110	R/W	Port I Sleep mode Pull-up/down Register	0x0
GPJCON	0x7F008120	R/W	Port J Configuration Register	0x0
GPJDAT	0x7F008124	R/W	Port J Data Register	Undefined
GPJPUD	0x7F008128	R/W	Port J Pull-up/down Register	0x00555555
GPJCONSLP	0x7F00812C	R/W	Port J Sleep mode Configuration Register	0x0
GPJPUDSLP	0x7F008130	R/W	Port J Sleep mode Pull-up/down Register	0x0
GPKCON0	0x7F008800	R/W	Port K Configuration Register 0	0x22222222
GPKCON1	0x7F008804	R/W	Port K Configuration Register 1	0x22222222
GPKDAT	0x7F008808	R/W	Port K Data Register	Undefined
GPKPUD	0x7F00880C	R/W	Port K Pull-up/down Register	0x55555555
GPLCON0	0x7F008810	R/W	Port L Configuration Register	0x22222222
GPLCON1	0x7F008814	R/W	Port L Configuration Register	0x02222222
GPLDAT	0x7F008818	R/W	Port L Data Register	Undefined
GPLPUD	0x7F00881C	R/W	Port L Pull-up/down Register	0x15555555
GPMCON	0x7F008820	R/W	Port M Configuration Register	0x00222222
GPMDAT	0x7F008824	R/W	Port M Data Register	Undefined
GPMPUD	0x7F008828	R/W	Port M Pull-up/down Register	0x000002AA
GPNCON	0x7F008830	R/W	Port N Configuration Register	0x0
GPNDAT	0x7F008834	R/W	Port N Data Register	Undefined
GPNPUD	0x7F008838	R/W	Port N Pull-up/down Register	0x55555555

Register	Address	R/W	Description	Reset Value
GPOCON	0x7F008140	R/W	Port O Configuration Register	0xAAAAAAAA
GPODAT	0x7F008144	R/W	Port O Data Register	Undefined
GPOPUD	0x7F008148	R/W	Port O Pull-up/down Register	0x0
GPOCONSLP	0x7F00814C	R/W	Port O Sleep mode Configuration Register	0x0
GPOPUDSLP	0x7F008150	R/W	Port O Sleep mode Pull-up/down Register	0x0
GPPCON	0x7F008160	R/W	Port P Configuration Register	0x2AAAAAAA
GPPDAT	0x7F008164	R/W	Port P Data Register	Undefined
GPPPUD	0x7F008168	R/W	Port P Pull-up/down Register	0x1011AAA0
GPPCONSLP	0x7F00816C	R/W	Port P Sleep mode Configuration Register	0x0
GPPPUDSLP	0x7F008170	R/W	Port P Sleep mode Pull-up/down Register	0x0
GPQCON	0x7F008180	R/W	Port Q Configuration Register	0x0002AAAA
GPQDAT	0x7F008184	R/W	Port Q Data Register	Undefined
GPQPUD	0x7F008188	R/W	Port Q Pull-up/down Register	0x0
GPQCONSLP	0x7F00818C	R/W	Port Q Sleep mode Configuration Register	0x0
GPQPUDSLP	0x7F008190	R/W	Port Q Sleep mode Pull-up/down Register	0x0
SPCON	0x7F0081A0	R/W	Special Port Configuration Register	0xBFC11500
MEM0CONSLP0	0x7F0081C0	R/W	Memory Port 0 Sleep mode configure 0	0x0
MEM0CONSLP1	0x7F0081C4	R/W	Memory Port 0 Sleep mode configure 1	0x0
MEM1CONSLP	0x7F0081C8	R/W	Memory Port 1 Sleep mode configure	0x0
MEM0DRVCON	0x7F0081D0	R/W	Memory Port 0 Drive strength Control Register	0x10555551
MEM1DRVCON	0x7F0081D4	R/W	Memory Port 0 Drive strength Control Register	0x555555
EINT0CON0	0x7F008900	R/W	External Interrupt configuration Register 0	0x0
EINT0CON1	0x7F008904	R/W	External Interrupt configuration Register 1	0x0
EINT0FLTCON0	0x7F008910	R/W	External Interrupt Filter Control Register 0	0x0
EINT0FLTCON1	0x7F008914	R/W	External Interrupt Filter Control Register 1	0x0
EINT0FLTCON2	0x7F008918	R/W	External Interrupt Filter Control Register 2	0x0
EINTF0LTCON3	0x7F00891C	R/W	External Interrupt Filter Control Register 3	0x0
EINT0MASK	0x7F008920	R/W	External Interrupt Mask Register	0x0FFFFFFF
EINT0PEND	0x7F008924	R/W	External Interrupt Pending Register	0x0
SPCONSLP	0x7F008880	R/W	Special Port Sleep mode configure Register	0x00000010
SLPEN	0x7F008930	R/W	Sleep Mode Pad Configure Register	0x0

Register	Address	R/W	Description	Reset Value
EINT12CON	0x7F008200	R/W	External Interrupt 1,2 Configuration Register	0x0
EINT34CON	0x7F008204	R/W	External Interrupt 3,4 Configuration Register	0x0
EINT56CON	0x7F008208	R/W	External Interrupt 5,6 Configuration Register	0x0
EINT78CON	0x7F00820C	R/W	External Interrupt 7,8 Configuration Register	0x0
EINT9CON	0x7F008210	R/W	External Interrupt 9 Configuration Register	0x0
EINT12FLTCON	0x7F008220	R/W	External Interrupt 1,2 Filter Control Register	0x0
EINT34FLTCON	0x7F008224	R/W	External Interrupt 3,4 Filter Control Register	0x0
EINT56FLTCON	0x7F008228	R/W	External Interrupt 5,6 Filter Control Register	0x0
EINT78FLTCON	0x7F00822C	R/W	External Interrupt 7,8 Filter Control Register	0x0
EINT9FLTCON	0x7F008230	R/W	External Interrupt 9 Filter Control Register	0x0
EINT12MASK	0x7F008240	R/W	External Interrupt 1,2 Mask Register	0x00FF7FFF
EINT34MASK	0x7F008244	R/W	External Interrupt 3,4 Mask Register	0x3FFF03FF
EINT56MASK	0x7F008248	R/W	External Interrupt 5,6 Mask Register	0x03FF007F
EINT78MASK	0x7F00824C	R/W	External Interrupt 7,8 Mask Register	0x7FFFFFFF
EINT9MASK	0x7F008250	R/W	External Interrupt 9 Mask Register	0x000001FF
EINT12PEND	0x7F008260	R/W	External Interrupt 1,2 Pending Register	0x0
EINT34PEND	0x7F008264	R/W	External Interrupt 3,4 Pending Register	0x0
EINT56PEND	0x7F008268	R/W	External Interrupt 5,6 Pending Register	0x0
EINT78PEND	0x7F00826C	R/W	External Interrupt 7,8 Pending Register	0x0
EINT9PEND	0x7F008270	R/W	External Interrupt 9 Pending Register	0x0
PRIORITY	0x7F008280	R/W	Priority Control Register	0x000003FF
SERVICE	0x7F008284	R	Current Service Register	0x0
SERVICEPEND	0x7F008288	R	Current Service Pending Register	0x0

NOTE: Please do not access the address area, which is not defined in the above table.

10.5 INDIVIDUAL REGISTER DESCRIPTIONS

10.5.1 PORT A CONTROL REGISTER

There are five control registers including GPACON, GPADAT, GPAPUD, GPACONSLP and GPAPUDSLP in the Port A Control Registers.

Register	Address	R/W	Description	Reset Value
GPACON	0x7F008000	R/W	Port A Configuration Register	0x0000
GPADAT	0x7F008004	R/W	Port A Data Register	Undefined
GPAPUD	0x7F008008	R/W	Port A Pull-up Register	0x0005555
GPACONSLP	0x7F00800C	R/W	Port A Sleep mode Configuration Register	0x0
GPAPUDSLP	0x7F008010	R/W	Port A Sleep mode Pull-up/down Register	0x0

GPACON	Bit	Description	Initial State
GPA0	[3:0]	0000 = Input 0001 = Output 0010 = UART RXD[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [0]	0000
GPA1	[7:4]	0000 = Input 0001 = Output 0010 = UART TXD[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [1]	0000
GPA2	[11:8]	0000 = Input 0001 = Output 0010 = UART CTSn[0] 0011 = Rreserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [2]	0000
GPA3	[15:12]	0000 = Input 0001 = Output 0010 = UART RTSn[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [3]	0000
GPA4	[19:16]	00 = Input 01 = Output 10 = UART RXD[1] 11 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [4]	0000
GPA5	[23:20]	0000 = Input 0001 = Output 0010 = UART TXD[1] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [5]	0000
GPA6	[27:24]	0000 = Input 0001 = Output 0010 = UART CTSn[1] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [6]	0000
GPA7	[31:28]	0000 = Input 0001 = Output 0010 = UART RTSn[1] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1 [7]	0000

GPADAT	Bit	Description
GPA[7:0]	[7:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPAPUD	Bit	Description
GPA[n]	[2n+1:2n] n = 0~7	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPACONSLP	Bit	Description	Initial State
GPA[n]	[2n+1:2n] n = 0~7	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPAPUDSLP	Bit	Description
GPA[n]	[2n+1:2n] n = 0~7	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.2 PORT B CONTROL REGISTER

There are five control registers including GPBCON, GPBDAT, GPBPUD, GPBCONSLP and GPBPUDSLP in the Port B Control Registers.

Register	Address	R/W	Description	Reset Value
GPBCON	0x7F008020	R/W	Port B Configuration Register	0x40000
GPBDAT	0x7F008024	R/W	Port B Data Register	Undefined
GPBPUD	0x7F008028	R/W	Port B Pull-up Register	0x00001555
GPBCONSLP	0x7F00802C	R/W	Port B Sleep mode Configuration Register	0x0
GPBPUDSLP	0x7F008030	R/W	Port B Sleep mode Pull-up/down Register	0x0

GPBCON	Bit	Description	Initial State
GPB0	[3:0]	0000 = Input 0001 = Output 0010 = UART RXD[2] 0011 = Ext. DMA Request 0100 = IrDA RXD 0101 = ADDR_CF[0] 0110 = Reserved 0111 = External Interrupt Group 1[8]	0000
GPB1	[7:4]	0000 = Input 0001 = Output 0010 = UART TXD[2] 0011 = Ext. DMA Ack 0100 = IrDA TXD 0101 = ADDR_CF[1] 0110 = Reserved 0111 = External Interrupt Group 1[9]	0000
GPB2	[11:8]	0000 = Input 0001 = Output 0010 = UART RXD[3] 0011 = IrDA RXD 0100 = Ext. DMA Req 0101 = ADDR_CF[2] 0110 = I2C SCL[1] 0111 = External Interrupt Group 1[10]	0000
GPB3	[15:12]	0000 = Input 0001 = Output 0010 = UART TXD[3] 0011 = Irda TXD 0100 = Ext. DMA Ack 0101 = Reserved 0110 = I2C SDA[1] 0111 = External Interrupt Group 1[11]	0000
GPB4	[19:16]	0000 = Input 0001 = Output 0010 = IrDA SDBW 0011 = CAM FIELD 0100 = CF Data DIR 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1[12]	0100
GPB5	[23:20]	0000 = Input 0001 = Output 0010 = I2C SCL[0] 0011 = reserved 0100 = Reseved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1[13]	0000
GPB6	[27:24]	0000 = Input 0001 = Output 0010 = I2C SDA[0] 0011 = reserved 0100 = Reseved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 1[14]	0000

GPBDAT	Bit	Description
GPB[6:0]	[6:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPBPUD	Bit	Description
GPB[n]	[2n+1:2n] n = 0~6	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPBSLPCON	Bit	Description	Initial State
GPB[n]	[2n+1:2n] n = 0~6	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPBPUDSLP	Bit	Description
GPB[n]	[2n+1:2n] n = 0~6	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.3 PORT C CONTROL REGISTER

There are five control registers including GPCCON, GPCDAT, GPCPUD, GPCCONSLP and GPCPUDSLP in the Port C Control Registers.

Register	Address	R/W	Description	Reset Value
GPCCON	0x7F008040	R/W	Port C Configuration Register	0x00
GPCDAT	0x7F008044	R/W	Port C Data Register	Undefined
GPCPUD	0x7F008048	R/W	Port C Pull-up/down Register	0x00005555
GPCCONSLP	0x7F00804C	R/W	Port C Sleep mode Configuration Register	0x0
GPCPUDSLP	0x7F008050	R/W	Port C Sleep mode Pull-up/down Register	0x0

GPCCON	Bit	Description	Initial State
GPC0	[3:0]	0000 = Input 0001 = Output 0010 = SPI MISO[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 2[0]	0000
GPC1	[7:4]	0000 = Input 0001 = Output 0010 = SPI CLK[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 2[1]	0000
GPC2	[11:8]	0000 = Input 0001 = Output 0010 = SPI MOSI[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 2[2]	0000
GPC3	[15:12]	0000 = Input 0001 = Output 0010 = SPI CSn[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 2[3]	0000
GPC4	[19:16]	0000 = Input 0001 = Output 0010 = SPI MISO[1] 0011 = MMC CMD2 0100 = Reserved 0101 = I2S_V40 DO[0] 0110 = Reserved 0111 = External Interrupt Group 2[4]	0000
GPC5	[23:20]	0000 = Input 0001 = Output 0010 = SPI CLK[1] 0011 = MMC CLK2 0100 = Reserved 0101 = I2S_V40 DO[1] 0110 = Reserved 0111 = External Interrupt Group 2[5]	0000
GPC6	[27:24]	0000 = Input 0001 = Output 0010 = SPI MOSI[1] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 2[6]	0000
GPC7	[31:28]	0000 = Input 0001 = Output 0010 = SPI CSn[1] 0011 = Reserved 0100 = Reserved 0101 = I2S_V40 DO[2] 0110 = Reserved 0111 = External Interrupt Group 2[7]	0000

GPCDAT	Bit	Description
GPC[7:0]	[7:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPCPUD	Bit	Description
GPC[n]	[2n+1:2n] n = 0~7	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPCSLPCON	Bit	Description	Initial State
GPC[n]	[2n+1:2n] n = 0~7	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

.

GPCPUDSLP	Bit	Description
GPC[n]	[2n+1:2n] n = 0~7	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.4 PORT D CONTROL REGISTER

There are five control registers including GPDCON, GPDDAT, GPDPUD, GPDCONSLP and GPDPUDSLP in the Port D Control Registers.

Register	Address	R/W	Description	Reset Value
GPDCON	0x7F008060	R/W	Port D Configuration Register	0x00
GPDDAT	0x7F008064	R/W	Port D Data Register	Undefined
GPDPUD	0x7F008068	R/W	Port D Pull-up/down Register	0x00000155
GPDCONSLP	0x7F00806C	R/W	Port D Sleep mode Configuration Register	0x0
GPDPUDSLP	0x7F008070	R/W	Port D Sleep mode Pull-up/down Register	0x0

GPDCON	Bit	Description	Initial State
GPD0	[3:0]	0000 = Input 0001 = Output 0010 = PCM SCLK[0] 0011 = I2S CLK[0] 0100 = AC97 BITCLK 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 3[0]	0000
GPD1	[7:4]	0000 = Input 0001 = Output 0010 = PCM EXTCLK[0] 0011 = I2S CDCLK[0] 0100 = AC97 RESETn 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 3[1]	0000
GPD2	[11:8]	0000 = Input 0001 = Output 0010 = PCM FSYNC[0] 0011 = I2S LRCLK[0] 0100 = AC97 SYNC 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 3[2]	0000
GPD3	[15:12]	0000 = Input 0001 = Output 0010 = PCM SIN[0] 0011 = I2S DI[0] 0100 = AC97 SDI 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 3[3]	0000
GPD4	[19:16]	0000 = Input 0001 = Output 0010 = PCM SOUT[0] 0011 = I2S DO[0] 0100 = AC97 SDO 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 3[4]	0000

GPDDAT	Bit	Description
GPD[4:0]	[4:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPDPUD	Bit	Description
GPD[n]	[2n+1:2n] n = 0~4	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved

GPDSLPCON	Bit	Description	Initial State
GPD[n]	[2n+1:2n] n = 0~4	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPDPUDSLP	Bit	Description
GPD[n]	[2n+1:2n] n = 0~4	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.5 PORT E CONTROL REGISTERS

There are five control registers including GPECON, GPEDAT, GPEPUD GPECONSLP and GPEPUDSLP in the Port E Control Registers.

Register	Address	R/W	Description	Reset Value
GPECON	0x7F008080	R/W	Port E Configuration Register	0x00
GPEDAT	0x7F008084	R/W	Port E Data Register	Undefined
GPEPUD	0x7F008088	R/W	Port E Pull-up/down Register	0x00000155
GPECONSLP	0x7F00808C	R/W	Port E Sleep mode Configuration Register	0x0
GPEPUDSLP	0x7F008090	R/W	Port E Sleep mode Pull-up/down Register	0x0

GPDCON	Bit	Description	Initial State
GPE0	[3:0]	0000 = Input 0001 = Output 0010 = PCM SCLK[1] 0011 = I2S CLK[1] 0100 = AC97 BITCLK 0101 = Reserved 0110 = Reserved 0111 = Reserved	0000
GPE1	[7:4]	0000 = Input 0001 = Output 0010 = PCM EXTCLK[1] 0011 = I2S CDCLK[1] 0100 = AC97 RESETn 0101 = Reserved 0110 = Reserved 0111 = Reserved	0000
GPE2	[11:8]	0000 = Input 0001 = Output 0010 = PCM FSYNC[1] 0011 = I2S LRCLK[1] 0100 = AC97 SYNC 0101 = Reserved 0110 = Reserved 0111 = Reserved	0000
GPE3	[15:12]	0000 = Input 0001 = Output 0010 = PCM SIN[1] 0011 = I2S DI[1] 0100 = AC97 SDI 0101 = Reserved 0110 = Reserved 0111 = Reserved	0000
GPE4	[19:16]	0000 = Input 0001 = Output 0010 = PCM SOUT[1] 0011 = I2S DO[1] 0100 = AC97 SDO 0101 = Reserved 0110 = Reserved 0111 = Reserved	0000

GPEDAT	Bit	Description
GPE[4:0]	[4:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPEPUD	Bit	Description
GPE[n]	[2n+1:2n] n = 0~4	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved

GPESLPCON	Bit	Description	Initial State
GPE[n]	[2n+1:2n] n = 0~4	00 = output 0 01 = output 1 10 = input 11 = Previous State	00

GPEPUDSLP	Bit	Description
GPE[n]	[2n+1:2n] n = 0~4	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.6 PORT F CONTROL REGISTERS

There are five control registers including GPFCON, GPFDAT, GPFPUD, GPFECONSLP and GPFPUDSLP in the Port F Control Registers.

Register	Address	R/W	Description	Reset Value
GPFCON	0x7F0080A0	R/W	Port F Configuration Register	0x00
GPFDAT	0x7F0080A4	R/W	Port F Data Register	Undefined
GPFPUD	0x7F0080A8	R/W	Port F Pull-up/down Register	0x55555555
GPFECONSLP	0x7F0080AC	R/W	Port F Sleep mode Configuration Register	0x0
GPFPUDSLP	0x7F0080B0	R/W	Port F Sleep mode Pull-up/down Register	0x0

GPFCON	Bit	Description	Initial State
GPF0	[1:0]	00 = Input 01 = Output 10 = CAMIF CLK 11 = External Interrupt Group 4[0]	00
GPF1	[3:2]	00 = Input 01 = Output 10 = CAMIF HREF 11 = External Interrupt Group 4[1]	00
GPF2	[5:4]	00 = Input 01 = Output 10 = CAMIF PCLK 11 = External Interrupt Group 4[2]	00
GPF3	[7:6]	00 = Input 01 = Output 10 = CAMIF RSTn 11 = External Interrupt Group 4[3]	00
GPF4	[9:8]	00 = Input 01 = Output 10 = CAMIF VSYNC 11 = External Interrupt Group 4[4]	00
GPF5	[11:10]	00 = Input 01 = Output 10 = CAMIF YDATA[0] 11 = External Interrupt Group 4[5]	00
GPF6	[13:12]	00 = Input 01 = Output 10 = CAMIF YDATA[1] 11 = External Interrupt Group 4[6]	00
GPF7	[15:14]	00 = Input 01 = Output 10 = CAMIF YDATA[2] 11 = External Interrupt Group 4[7]	00
GPF8	[17:16]	00 = Input 01 = Output 10 = CAMIF YDATA[3] 11 = External Interrupt Group 4[8]	00
GPF9	[19:18]	00 = Input 01 = Output 10 = CAMIF YDATA[4] 11 = External Interrupt Group 4[9]	00
GPF10	[21:20]	00 = Input 01 = Output 10 = CAMIF YDATA[5] 11 = External Interrupt Group 4[10]	00
GPF11	[23:22]	00 = Input 01 = Output 10 = CAMIF YDATA[6] 11 = External Interrupt Group 4[11]	00
GPF12	[25:24]	00 = Input 01 = Output 10 = CAMIF YDATA[7] 11 = External Interrupt Group 4[12]	00
GPF13	[27:26]	00 = Input 01 = Output 10 = PWM ECLK 11 = External Interrupt Group 4[13]	00
GPF14	[29:28]	00 = Input 01 = Output 10 = PWM TOUT[0] 11 = CLKOUT[0]	00
GPF15	[31:30]	00 = Input 01 = Output 10 = PWM TOUT[1] 11 = Reserved	00

GPFDAT	Bit	Description
GPF[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPFPUD	Bit	Description
GPF[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPFSLPCON	Bit	Description	Initial State
GPF[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPFPUDSLP	Bit	Description
GPF[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.7 PORT G CONTROL REGISTERS

There are five control registers including GPGCON, GPGDAT, GPGPU, GPGECONSLP and GPGPUDSLP in the Port G Control Registers.

Register	Address	R/W	Description	Reset Value
GPGCON	0x7F0080C0	R/W	Port G Configuration Register	0x00
GPGDAT	0x7F0080C4	R/W	Port G Data Register	Undefined
GPGPUD	0x7F0080C8	R/W	Port G Pull-up/down Register	0x00001555
GPGECONSLP	0x7F0080CC	R/W	Port G Sleep mode Configuration Register	0x0
GPGPUDSLP	0x7F0080D0	R/W	Port G Sleep mode Pull-up/down Register	0x0

GPGCON	Bit	Description	Initial State
GPG0	[3:0]	0000 = Input 0001 = Output 0010 = MMC CLK0 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[0]	0000
GPG1	[7:4]	0000 = Input 0001 = Output 0010 = MMC CMD0 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[1]	0000
GPG2	[11:8]	0000 = Input 0001 = Output 0010 = MMC DATA0[0] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[2]	0000
GPG3	[15:12]	0000 = Input 0001 = Output 0010 = MMC DATA0[1] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[3]	0000
GPG4	[19:16]	0000 = Input 0001 = Output 0010 = MMC DATA0[2] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[4]	0000
GPG5	[23:20]	0000 = Input 0001 = Output 0010 = MMC DATA0[3] 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[5]	0000
GPG6	[27:24]	0000 = Input 0001 = Output 0010 = MMC CDn0 0011 = MMC CDn1 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 5[6]	0000

GPGDAT	Bit	Description
GPG[6:0]	[6:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPGPUD	Bit	Description
GPG[n]	[2n+1:2n] n = 0~6	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved

GPGSLPCON	Bit	Description	Initial State
GPG[n]	[2n+1:2n] n = 0~6	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPGPUDSLP	Bit	Description
GPG[n]	[2n+1:2n] n = 0~6	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved

10.5.8 PORT H CONTROL REGISTERS

There are six control registers including GPHCON0, GPHCON1, GPHDAT, GPHPUD, GPHCONSLP and GPHPUDSLP in the Port H Control Registers.

Register	Address	R/W	Description	Reset Value
GPHCON0	0x7F0080E0	R/W	Port H Configuration Register	0x00
GPHCON1	0x7F0080E4	R/W	Port H Configuration Register	0x00
GPHDAT	0x7F0080E8	R/W	Port H Data Register	Undefined
GPHPUD	0x7F0080EC	R/W	Port H Pull-up/down Register	0x00055555
GPHCONSLP	0x7F0080F0	R/W	Port H Sleep mode Configuration Register	0x0
GPHPUDSLP	0x7F0080F4	R/W	Port H Sleep mode Pull-up/down Register	0x0

GPHCON0	Bit	Description	Initial State
GPH0	[3:0]	0000 = Input 0001 = Output 0010 = MMC CLK1 0011 = Reserved 0100 = Key pad COL[0] 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 6[0]	0000
GPH1	[7:4]	0000 = Input 0001 = Output 0010 = MMC CMD1 0011 = Reserved 0100 = Key pad COL[1] 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 6[1]	0000
GPH2	[11:8]	0000 = Input 0001 = Output 0010 = MMC DATA1[0] 0011 = Reserved 0100 = Key pad COL[2] 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 6[2]	0000
GPH3	[15:12]	0000 = Input 0001 = Output 0010 = MMC DATA1[1] 0011 = Reserved 0100 = Key pad COL[3] 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 6[3]	0000
GPH4	[19:16]	0000 = Input 0001 = Output 0010 = MMC DATA1[2] 0011 = Reserved 0100 = Key pad COL[4] 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 6[4]	0000
GPH5	[23:20]	0000 = Input 0001 = Output 0010 = MMC DATA1[3] 0011 = Reserved 0100 = Key pad COL[5] 0101 = Reserved 0110 = Reserved 0111 = External Interrupt Group 6[5]	0000
GPH6	[27:24]	0000 = Input 0001 = Output 0010 = MMC DATA1[4] 0011 = MMC DATA2[0] 0100 = Key pad COL[6] 0101 = I2S_V40 BCLK 0110 = ADDR_CF[0] 0111 = External Interrupt Group 6[6]	0000
GPH7	[31:28]	0000 = Input 0001 = Output 0010 = MMC DATA1[5] 0011 = MMC DATA2[1] 0100 = Key pad COL[7] 0101 = I2S_V40 CDCLK 0110 = ADDR_CF[1] 0111 = External Interrupt Group 6[7]	0000

GPHCON1	Bit	Description	Initial State
GPH8	[3:0]	0000 = Input 0001 = Output 0010 = MMC DATA1[6] 0011 = MMC DATA2[2] 0100 = Reserved 0101 = I2S_V40 LRCLK 0110 = ADDR_CF[2] 0111 = External Interrupt Group 6[8]	0000
GPH9	[7:4]	0000 = Input 0001 = Output 0010 = MMC DATA1[7] 0011 = MMC DATA2[3] 0100 = Reserved 0101 = I2S_V40 DI 0110 = Reserved 0111 = External Interrupt Group 6[9]	0000

GPHDAT	Bit	Description
GPH[9:0]	[9:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPHPUD	Bit	Description
GPH[n]	[2n+1:2n] n = 0~9	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPHSLPCON	Bit	Description	Initial State
GPH[n]	[2n+1:2n] n = 0~9	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPHPUDSLP	Bit	Description
GPH[n]	[2n+1:2n] n = 0~9	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.9 PORT I CONTROL REGISTERS

There are five control registers including GPICON, GPIDAT, GPIPUD, GPICONSLP, and GPIPUDSLP in the Port I Control Registers.

Register	Address	R/W	Description	Reset Value
GPICON	0x7F008100	R/W	Port I Configuration Register	0x00
GPIDAT	0x7F008104	R/W	Port I Data Register	Undefined
GPIPUD	0x7F008108	R/W	Port I Pull-up/down Register	0x55555555
GPICONSLP	0x7F00810C	R/W	Port I Sleep mode Configuration Register	0x0
GPIPUDSLP	0x7F008110	R/W	Port I Sleep mode Pull-up/down Register	0x0

GPICON	Bit	Description	Initial State
GPI0	[1:0]	00 = Input 01 = Output 10 = LCD VD[0] 11 = reserved	00
GPI1	[3:2]	00 = Input 01 = Output 10 = LCD VD[1] 11 = reserved	00
GPI2	[5:4]	00 = Input 01 = Output 10 = LCD VD[2] 11 = reserved	00
GPI3	[7:6]	00 = Input 01 = Output 10 = LCD VD[3] 11 = reserved	00
GPI4	[9:8]	00 = Input 01 = Output 10 = LCD VD[4] 11 = reserved	00
GPI5	[11:10]	00 = Input 01 = Output 10 = LCD VD[5] 11 = reserved	00
GPI6	[13:12]	00 = Input 01 = Output 10 = LCD VD[6] 11 = reserved	00
GPI7	[15:14]	00 = Input 01 = Output 10 = LCD VD[7] 11 = reserved	00
GPI8	[17:16]	00 = Input 01 = Output 10 = LCD VD[8] 11 = reserved	00
GPI9	[19:18]	00 = Input 01 = Output 10 = LCD VD[9] 11 = reserved	00
GPI10	[21:20]	00 = Input 01 = Output 10 = LCD VD[10] 11 = reserved	00
GPI11	[23:22]	00 = Input 01 = Output 10 = LCD VD[11] 11 = reserved	00
GPI12	[25:24]	00 = Input 01 = Output 10 = LCD VD[12] 11 = reserved	00
GPI13	[27:26]	00 = Input 01 = Output 10 = LCD VD[13] 11 = reserved	00
GPI14	[29:28]	00 = Input 01 = Output 10 = LCD VD[14] 11 = reserved	00
GPI15	[31:30]	00 = Input 01 = Output 10 = LCD VD[15] 11 = reserved	00

GPIDAT	Bit	Description
GPI[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPIPUD	Bit	Description
GPI[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPISLPCON	Bit	Description	Initial State
GPI[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 1* = input	00

GPIPUDSLP	Bit	Description
GPI[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

NOTE:

1. When LCD Bypass mode is set (MIFPCON[3]=1) in sleep mode, GPISLPCON and GPIPUDSLP cannot control I port. Because in this case I port IO cells are controlled by Host I/F block(Modem I/F block) and signals from K, L, M port IO cells. For more information, please refer to figure 10-3.

10.5.10 PORT J CONTROL REGISTERS

There are five control registers including GPJCON, GPJDAT, and GPJPUD, GPJCONSLP and GPJPUDSLP in the Port J Control Registers.

Register	Address	R/W	Description	Reset Value
GPJCON	0x7F008120	R/W	Port J Configuration Register	0x00
GPJDAT	0x7F008124	R/W	Port J Data Register	Undefined
GPJPUD	0x7F008128	R/W	Port J Pull-up/down Register	0x00555555
GPJCONSLP	0x7F00812C	R/W	Port J Sleep mode Configuration Register	0x0
GPJPUDSLP	0x7F008130	R/W	Port J Sleep mode Pull-up/down Register	0x0

GPJCON	Bit	Description	Initial State
GPJ0	[1:0]	00 = Input 01 = Output 10 = LCD VD[16] 11 = reserved	00
GPJ1	[3:2]	00 = Input 01 = Output 10 = LCD VD[17] 11 = reserved	00
GPJ2	[5:4]	00 = Input 01 = Output 10 = LCD VD[18] 11 = reserved	00
GPJ3	[7:6]	00 = Input 01 = Output 10 = LCD VD[19] 11 = reserved	00
GPJ4	[9:8]	00 = Input 01 = Output 10 = LCD VD[20] 11 = reserved	00
GPJ5	[11:10]	00 = Input 01 = Output 10 = LCD VD[21] 11 = reserved	00
GPJ6	[13:12]	00 = Input 01 = Output 10 = LCD VD[22] 11 = reserved	00
GPJ7	[15:14]	00 = Input 01 = Output 10 = LCD VD[23] 11 = reserved	00
GPJ8	[17:16]	00 = Input 01 = Output 10 = LCD HSYNC 11 = Reserved	00
GPJ9	[19:18]	00 = Input 01 = Output 10 = LCD VSYNC 11 = Reserved	00
GPJ10	[21:20]	00 = Input 01 = Output 10 = LCD VDEN 11 = Reserved	00
GPJ11	[23:22]	00 = Input 01 = Output 10 = LCD VCLK 11 = Reserved	00

GPJDAT	Bit	Description
GPJ[15:0]	[11:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPJPUD	Bit	Description
GPJ[n]	[2n+1:2n] n = 0~11	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPJSLPCON	Bit	Description	Initial State
GPJ[n]	[2n+1:2n] n = 0~11	00 = output 0 01 = output 1 1* = input	00

GPJPUDSLP	Bit	Description
GPJ[n]	[2n+1:2n] n = 0~11	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

NOTE:

1. When LCD Bypass mode is set (MIFPCON[3]=1) in sleep mode, GPJSLPCON and GPJPUDSLP cannot control J port. Because in this case J port IO cells are controlled by Host I/F block(Modem I/F block) and signals from K, L, M port IO cells. For more information, please refer to figure 10-3.

10.5.11 PORT K CONTROL REGISTERS

There are four control registers including GPKCON0, GPKCON1, GPKDAT, and GPKPUD in the Port K Control Registers. GPKCON0, GPKCON1, GPKDAT and GPKPUD are alive part.

Register	Address	R/W	Description	Reset Value
GPKCON0	0x7F008800	R/W	Port K Configuration Register 0	0x22222222
GPKCON1	0x7F008804	R/W	Port K Configuration Register 1	0x22222222
GPKDAT	0x7F008808	R/W	Port K Data Register	Undefined
GPKPUD	0x7F00880C	R/W	Port K Pull-up/down Register	0x55555555

GPKCON0	Bit	Description	Initial State
GPK0	[3:0]	0000 = Input 0001 = Output 0010 = Host I/F DATA[0] 0011 = HSI RX READY 0100 = Reserved 0101 = DATA_CF[0] 0110 = Reserved 0111 = Reserved	0010
GPK1	[7:4]	0000 = Input 0001 = Output 0010 = Host I/F DATA[1] 0011 = HSI RX WAKE 0100 = Reserved 0101 = DATA_CF[1] 0110 = Reserved 0111 = Reserved	0010
GPK2	[11:8]	0000 = Input 0001 = Output 0010 = Host I/F DATA[2] 0011 = HSI RX FLAG 0100 = Reserved 0101 = DATA_CF[2] 0110 = Reserved 0111 = Reserved	0010
GPK3	[15:12]	0000 = Input 0001 = Output 0010 = Host I/F DATA[3] 0011 = HSI RX DATA 0100 = Reserved 0101 = DATA_CF[3] 0110 = Reserved 0111 = Reserved	0010
GPK4	[19:16]	0000 = Input 0001 = Output 0010 = Host I/F DATA[4] 0011 = HSI TX READY 0100 = Reserved 0101 = DATA_CF[4] 0110 = Reserved 0111 = Reserved	0010
GPK5	[23:20]	0000 = Input 0001 = Output 0010 = Host I/F DATA[5] 0011 = HSI TX WAKE 0100 = Reserved 0101 = DATA_CF[5] 0110 = Reserved 0111 = Reserved	0010
GPK6	[27:24]	0000 = Input 0001 = Output 0010 = Host I/F DATA[6] 0011 = HSI TX FLAG 0100 = Reserved 0101 = DATA_CF[6] 0110 = Reserved 0111 = Reserved	0010
GPK7	[31:28]	0000 = Input 0001 = Output 0010 = Host I/F DATA[7] 0011 = HSI TX DATA 0100 = Reserved 0101 = DATA_CF[7] 0110 = Reserved 0111 = Reserved	0010

GPKCON1	Bit	Description		Initial State
GPK8	[3:0]	0000 = Input 0010 = Host I/F DATA[8] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[0] 0101 = DATA_CF[8] 0111 = Reserved	0010
GPK9	[7:4]	0000 = Input 0010 = Host I/F DATA[9] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[1] 0101 = DATA_CF[9] 0111 = Reserved	0010
GPK10	[11:8]	0000 = Input 0010 = Host I/F DATA[10] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[2] 0101 = DATA_CF[10] 0111 = Reserved	0010
GPK11	[15:12]	0000 = Input 0010 = Host I/F DATA[11] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[3] 0101 = DATA_CF[11] 0111 = Reserved	0010
GPK12	[19:16]	0000 = Input 0010 = Host I/F DATA[12] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[4] 0101 = DATA_CF[12] 0111 = Reserved	0010
GPK13	[23:20]	0000 = Input 0010 = Host I/F DATA[13] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[5] 0101 = DATA_CF[13] 0111 = Reserved	0010
GPK14	[27:24]	0000 = Input 0010 = Host I/F DATA[14] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[6] 0101 = DATA_CF[14] 0111 = Reserved	0010
GPK15	[31:28]	0000 = Input 0010 = Host I/F DATA[15] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Key pad ROW[7] 0101 = DATA_CF[15] 0111 = Reserved	0010

GPKDAT	Bit	Description
GPK[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPKPUD	Bit	Description
GPK[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.12 PORT L CONTROL REGISTERS

There are four control registers including GPLCON0, GPLCON1, GPLDAT and GPLPUD, in the Port L Control Registers. GPLCON0, GPLCON1, GPLDAT and GPLPUD are alive part.

Register	Address	R/W	Description	Reset Value
GPLCON0	0x7F008810	R/W	Port L Configuration Register 0	0x22222222
GPLCON1	0x7F008814	R/W	Port L Configuration Register 1	0x02222222
GPLDAT	0x7F008818	R/W	Port L Data Register	Undefined
GPLPUD	0x7F00881C	R/W	Port L Pull-up/down Register	0x15555555

GPLCON0	Bit	Description		Initial State
GPL0	[3:0]	0000 = Input 0010 = Host I/F ADDR[0] 0100 = Reserved 0110 = ADDR_CF[0]	0001 = Output 0011 = Key pad COL[0] 0101 = Reserved 0111 = Reserved	0010
GPL1	[7:4]	0000 = Input 0010 = Host I/F ADDR[1] 0100 = Reserved 0110 = ADDR_CF[1]	0001 = Output 0011 = Key pad COL[1] 0101 = Reserved 0111 = Reserved	0010
GPL2	[11:8]	0000 = Input 0010 = Host I/F ADDR[2] 0100 = Reserved 0110 = ADDR_CF[2]	0001 = Output 0011 = Key pad COL[2] 0101 = Reserved 0111 = Reserved	0010
GPL3	[15:612	0000 = Input 0010 = Host I/F ADDR[3] 0100 = Reserved 0110 = MEM0_INTata	0001 = Output 0011 = Key pad COL[3] 0101 = Reserved 0111 = Reserved	0010
GPL4	[19:16]	0000 = Input 0010 = Host I/F ADDR[4] 0100 = Reserved 0110 = MEM0_RESETata	0001 = Output 0011 = Key pad COL[4] 0101 = Reserved 0111 = Reserved	0010
GPL5	[23:20]	0000 = Input 0010 = Host I/F ADDR[5] 0100 = Reserved 0110 = MEM0_INPACKata	0001 = Output 0011 = Key pad COL[5] 0101 = Reserved 0111 = Reserved	0010
GPL6	[27:24]	0000 = Input 0010 = Host I/F ADDR[6] 0100 = Reserved 0110 = MEM0_REGata	0001 = Output 0011 = Key pad COL[6] 0101 = Reserved 0111 = Reserved	0010
GPL7	[31:28]	0000 = Input 0010 = Host I/F ADDR[7] 0100 = Reserved 0110 = MEM0_CData	0001 = Output 0011 = Key pad COL[7] 0101 = Reserved 0111 = Reserved	0010

GPLCON1	Bit	Description		Initial State
GPL8	[3:0]	0000 = Input 0010 = Host I/F ADDR[8] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt[16] 0101 = CE_CF[0] 0111 = Reserved	0010
GPL9	[7:4]	0000 = Input 0010 = Host I/F ADDR[9] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt [17] 0101 = CE_CF[1] 0111 = Reserved	0010
GPL10	[11:8]	0000 = Input 0010 = Host I/F ADDR[10] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt [18] 0101 = IORD_CF 0111 = Reserved	0010
GPL11	[15:12]	0000 = Input 0010 = Host I/F ADDR[11] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt [19] 0101 = IOWR_CF 0111 = Reserved	0010
GPL12	[19:16]	0000 = Input 0010 = Host I/F ADDR[12] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt [20] 0101 = IORDY_CF 0111 = Reserved	0010
GPL13	[23:20]	0000 = Input 0010 = Host I/F DATA[16] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt [21] 0101 = Reserved 0111 = Reserved	0010
GPL14	[27:24]	0000 = Input 0010 = Host I/F DATA[17] 0100 = Reserved 0110 = Reserved	0001 = Output 0011 = Ext. Interrupt [22] 0101 = Reserved 0111 = Reserved	0010

GPLDAT	Bit	Description
GPL[14:0]	[14:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPLPUD	Bit	Description
GPL[n]	[2n+1:2n] n = 0~14	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.13 PORT M CONTROL REGISTERS

There are three control registers including GPMCON, GPMDAT and GPMPUD in the Port M Control Registers. GPMCON, GPMDAT and GPMPUD are alive part.

Register	Address	R/W	Description	Reset Value
GPMCON	0x7F008820	R/W	Port M Configuration Register	0x00222222
GPMDAT	0x7F008824	R/W	Port M Data Register	Undefined
GPMPUD	0x7F008828	R/W	Port M Pull-up/down Register	0x000002AA

GPMCON	Bit	Description	Initial State
GPM0	[3:0]	0000 = Input 0001 = Output 0010 = Host I/F CSn 0011 = Ext. Interrupt[23] 0100 = Reserved 0101 = Reserved 0110 = CE_CF[0] 0111 = Reserved	0010
GPM1	[7:4]	0000 = Input 0001 = Output 0010 = Host I/F CSn_main 0011 = Ext. Interrupt [24] 0100 = Reserved 0101 = Reserved 0110 = CE_CF[1] 0111 = Reserved	0010
GPM2	[11:8]	0000 = Input 0001 = Output 0010 = Host I/F CSn_sub 0011 = Ext. Interrupt [25] 0100 = Host I/F MDP_VSYNC 0101 = Reserved 0110 = IORD_CF 0111 = Reserved	0010
GPM3	[15:12]	0000 = Input 0001 = Output 0010 = Host I/F WEn 0011 = Ext. Interrupt [26] 0100 = Reserved 0101 = Reserved 0110 = IOWR_CF 0111 = Reserved	0010
GPM4	[19:16]	0000 = Input 0001 = Output 0010 = Host I/F OEn 0011 = Ext. Interrupt [27] 0100 = Reserved 0101 = Reserved 0110 = IORDY_CF 0111 = Reserved	0010
GPM5	[23:20]	0000 = Input 0001 = Output 0010 = Host I/F INTRn 0011 = CF Data Dir. 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = Reserved	0010

GPMDAT	Bit	Description
GPM[5:0]	[5:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPLPUD	Bit	Description
GPM[n]	[2n+1:2n] n = 0~5	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.14 PORT N CONTROL REGISTERS

There are three control registers including GPNCON, GPNDAT and GPNPUD in the Port N Control Registers. GPNCON, GPNDAT and GPNPUD are alive part.

Register	Address	R/W	Description	Reset Value
GPNCON	0x7F008830	R/W	Port N Configuration Register	0x00
GPNDAT	0x7F008834	R/W	Port N Data Register	Undefined
GPNPUD	0x7F008838	R/W	Port N Pull-up/down Register	0x55555555

GPNCON	Bit	Description	Initial State
GPN0	[1:0]	00 = Input 01 = Output 10 = Ext. Interrupt[0] 11 = Key pad ROW[0]	00
GPN1	[3:2]	00 = Input 01 = Output 10 = Ext. Interrupt[1] 11 = Key pad ROW[1]	00
GPN2	[5:4]	00 = Input 01 = Output 10 = Ext. Interrupt[2] 11 = Key pad ROW[2]	00
GPN3	[7:6]	00 = Input 01 = Output 10 = Ext. Interrupt[3] 11 = Key pad ROW[3]	00
GPN4	[9:8]	00 = Input 01 = Output 10 = Ext. Interrupt[4] 11 = Key pad ROW[4]	00
GPN5	[11:10]	00 = Input 01 = Output 10 = Ext. Interrupt[5] 11 = Key pad ROW[5]	00
GPN6	[13:12]	00 = Input 01 = Output 10 = Ext. Interrupt[6] 11 = Key pad ROW[6]	00
GPN7	[15:14]	00 = Input 01 = Output 10 = Ext. Interrupt[7] 11 = Key pad ROW[7]	00
GPN8	[17:16]	00 = Input 01 = Output 10 = Ext. Interrupt[8] 11 = Reserved	00
GPN9	[19:18]	00 = Input 01 = Output 10 = Ext. Interrupt[9] 11 = Reserved	00
GPN10	[21:20]	00 = Input 01 = Output 10 = Ext. Interrupt[10] 11 = Reserved	00
GPN11	[23:22]	00 = Input 01 = Output 10 = Ext. Interrupt[11] 11 = Reserved	00
GPN12	[25:24]	00 = Input 01 = Output 10 = Ext. Interrupt[12] 11 = Reserved]	00
GPN13	[27:26]	00 = Input 01 = Output 10 = Ext. Interrupt[13] 11 = Reserved	00
GPN14	[29:28]	00 = Input 01 = Output 10 = Ext. Interrupt[14] 11 = Reserved	00
GPN15	[31:30]	00 = Input 01 = Output 10 = Ext. Interrupt[15] 11 = Reserved	00

GPNDAT	Bit	Description
GPN[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPNPUD	Bit	Description
GPN[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

10.5.15 PORT O CONTROL REGISTERS

There are five control registers including GPOCON, GPODAT, GPOPUD, POCONSLP and GPOPUDSLP in the Port O Control Registers.

Register	Address	R/W	Description	Reset Value
GPOCON	0x7F008140	R/W	Port O Configuration Register	0xAAAAAAAA
GPODAT	0x7F008144	R/W	Port O Data Register	Undefined
GPOPUD	0x7F008148	R/W	Port O Pull-up/down Register	0x0
GPOCONSLP	0x7F00814C	R/W	Port O Sleep mode Configuration Register	0x0
GPOPUDSLP	0x7F008150	R/W	Port O Sleep mode Pull-up/down Register	0x0

GPOCON	Bit	Description	Initial State
GPO0	[1:0]	00 = Input 01 = Output 10 = MEM0_nCS[2] 11 = Ext. Interrupt Group7[0]	10
GPO1	[3:2]	00 = Input 01 = Output 10 = MEM0_nCS[3] 11 = Ext. Interrupt Group7 [1]	10
GPO2	[5:4]	00 = Input 01 = Output 10 = MEM0_nCS[4] 11 = Ext. Interrupt Group7 [2]	10
GPO3	[7:6]	00 = Input 01 = Output 10 = MEM0_nCS[5] 11 = Ext. Interrupt Group7 [3]	10
GPO4	[9:8]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group7 [4]	10
GPO5	[11:10]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group7 [5]	10
GPO6	[13:12]	00 = Input 01 = Output 10 = MEM0_ADDR[6] 11 = Ext. Interrupt Group7 [6]	10
GPO7	[15:14]	00 = Input 01 = Output 10 = MEM0_ADDR[7] 11 = Ext. Interrupt Group7 [7]	10
GPO8	[17:16]	00 = Input 01 = Output 10 = MEM0_ADDR[8] 11 = Ext. Interrupt Group7 [8]	10
GPO9	[19:18]	00 = Input 01 = Output 10 = MEM0_ADDR[9] 11 = Ext. Interrupt Group7 [9]	10
GPO10	[21:20]	00 = Input 01 = Output 10 = MEM0_ADDR[10] 11 = Ext. Interrupt Group7 [10]	10
GPO11	[23:22]	00 = Input 01 = Output 10 = MEM0_ADDR[11] 11 = Ext. Interrupt Group7 [11]	10
GPO12	[25:24]	00 = Input 01 = Output 10 = MEM0_ADDR[12] 11 = Ext. Interrupt Group7 [12]	10
GPO13	[27:26]	00 = Input 01 = Output 10 = MEM0_ADDR[13] 11 = Ext. Interrupt Group7 [13]	10
GPO14	[29:28]	00 = Input 01 = Output 10 = MEM0_ADDR[14] 11 = Ext. Interrupt Group7 [14]	10
GPO15	[31:30]	00 = Input 01 = Output 10 = MEM0_ADDR[15] 11 = Ext. Interrupt Group7 [15]	10

GPODAT	Bit	Description
GPO[15:0]	[15:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPOPUD	Bit	Description
GPO[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPOCONSLP	Bit	Description	Initial State
GPO[n]	[2n+1:2n] n = 0~15	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPOPUDSLP	Bit	Description
GPO[n]	[2n+1:2n] n = 0~15	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

NOTES:

1. When the ports are set to memory interface signals, the pull-up/down are disabled
2. When the ports are set to memory interface signals, their states are controlled by MEM0CONSTOP in stop mode and MEM0CONSLP0 in sleep mode

10.5.16 PORT P CONTROL REGISTERS

There are five control registers including GPPCON, GPPDAT, GPPPUD, GPPCONSLP and GPPPUDSLP in the Port P Control Registers.

Register	Address	R/W	Description	Reset Value
GPPCON	0x7F008160	R/W	Port P Configuration Register	0x2AAAAAAA
GPPDAT	0x7F008164	R/W	Port P Data Register	Undefined
GPPPUD	0x7F008168	R/W	Port P Pull-up/down Register	0x1011AAA0
GPPCONSLP	0x7F00816C	R/W	Port P Sleep mode Configuration Register	0x0
GPPPUDSLP	0x7F008170	R/W	Port P Sleep mode Pull-up/down Register	0x0

GPPCON	Bit	Description	Initial State
GPP0	[1:0]	00 = Input 01 = Output 10 = MEM0_ADDRV 11 = Ext. Interrupt Group8[0]	10
GPP1	[3:2]	00 = Input 01 = Output 10 = MEM0_SMCLK 11 = Ext. Interrupt Group8[1]	10
GPP2	[5:4]	00 = Input 01 = Output 10 = MEM0_nWAIT 11 = Ext. Interrupt Group8 [2]	10
GPP3	[7:6]	00 = Input 01 = Output 10 = MEM0_RDY0_ALE 11 = Ext. Interrupt Group8 [3]	10
GPP4	[9:8]	00 = Input 01 = Output 10 = MEM0_RDY1_CLE 11 = Ext. Interrupt Group8 [4]	10
GPP5	[11:10]	00 = Input 01 = Output 10 = MEM0_INTsm0_FWE 11 = Ext. Interrupt Group8 [5]	10
GPP6	[13:12]	00 = Input 01 = Output 10 = MEM0_INTsm1_FRE 11 = Ext. Interrupt Group8 [6]	10
GPP7	[15:14]	00 = Input 01 = Output 10 = MEM0_RPn_RnB 11 = Ext. Interrupt Group8 [7]	10
GPP8	[17:16]	00 = Input 01 = Output 10 = MEM0_INTata 11 = Ext. Interrupt Group8 [8]	10
GPP9	[19:18]	00 = Input 01 = Output 10 = MEM0_RESETata 11 = Ext. Interrupt Group8 [9]	10
GPP10	[21:20]	00 = Input 01 = Output 10 = MEM0_INPACKata11 = Ext. Interrupt Group8 [10]	10
GPP11	[23:22]	00 = Input 01 = Output 10 = MEM0_REGata 11 = Ext. Interrupt Group8 [11]	10
GPP12	[25:24]	00 = Input 01 = Output 10 = MEM0_WEata 11 = Ext. Interrupt Group8 [12]	10
GPP13	[27:26]	00 = Input 01 = Output 10 = MEM0_OEata 11 = Ext. Interrupt Group8 [13]	10
GPP14	[29:28]	00 = Input 01 = Output 10 = MEM0_CData 11 = Ext. Interrupt Group8 [14]	10

GPPDAT	Bit	Description
GPP[14:0]	[14:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPPPUD	Bit	Description
GPP[n]	[2n+1:2n] n = 0~14	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPPCONSLP	Bit	Description	Initial State
GPP[n]	[2n+1:2n] n = 0~14	00 = output 0 01 = output 1 10 = input 11 = Previuos state	00

GPPPUDSLP	Bit	Description
GPP[n]	[2n+1:2n] n = 0~14	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

NOTE:

1. When the ports are set to memory interface signals, their states are controlled by MEM0CONSTOP in stop mode, MEM0CONSLP1 in sleep mode

10.5.17 PORT Q CONTROL REGISTERS

There are five control registers including GPQCON, GPQDAT, GPQPUD, GPQCONSLP and GPQPUDSLP in the Port Q Control Registers.

Register	Address	R/W	Description	Reset Value
GPQCON	0x7F008180	R/W	Port Q Configuration Register	0x0002AAAA
GPQDAT	0x7F008184	R/W	Port Q Data Register	Undefined
GPQPUD	0x7F008188	R/W	Port Q Pull-up/down Register	0x0
GPQCONSLP	0x7F00818C	R/W	Port Q Sleep mode Configuration Register	0x0
GPQPUDSLP	0x7F008190	R/W	Port Q Sleep mode Pull-up/down Register	0x0

GPQCON	Bit	Description	Initial State
GPQ0	[1:0]	00 = Input 01 = Output 10 = MEM0_ADDR18 11 = Ext. Interrupt Group9[0]	10
GPQ1	[3:2]	00 = Input 01 = Output 10 = MEM0_ADDR19 11 = Ext. Interrupt Group9[1]	10
GPQ2	[5:4]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group9 [2]	10
GPQ3	[7:6]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group9 [3]	10
GPQ4	[9:8]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group9 [4]	10
GPQ5	[11:10]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group9 [5]	10
GPQ6	[13:12]	00 = Input 01 = Output 10 = Reserved 11 = Ext. Interrupt Group9 [6]	10
GPQ7	[15:14]	00 = Input 01 = Output 10 = MEM0_ADDR17 11 = Ext. Interrupt Group9 [7]	10
GPQ8	[17:16]	00 = Input 01 = Output 10 = MEM0_ADDR16 11 = Ext. Interrupt Group9 [8]	10

GPQDAT	Bit	Description
GPQ[8:0]	[8:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPQPUD	Bit	Description
GPQ[n]	[2n+1:2n] n = 0~8	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

GPQCONSLP	Bit	Description	Initial State
GPQ[n]	[2n+1:2n] n = 0~8	00 = output 0 01 = output 1 10 = input 11 = Previous state	00

GPQPUDSLP	Bit	Description
GPQ[n]	[2n+1:2n] n = 0~8	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

NOTES:

1. When the ports are set to memory interface signals, their states are controlled by MEM0CONSTOP in stop mode, MEM0CONSLP0 in sleep mode
2. When When the ports GPQCON[3:0] and GPQCON[17:14] are set to memory interface signals, the pull-up/down are disabled

10.5.18 SPECIAL PORT CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
SPCON	0x7F0081A0	R/W	Special Port Control Register	0xBFC11500

SPCON	Bit	Description	Initial State
DRVCON_CAM	[31:30]	CAMERA Port Drive strength 00 = 2mA 01 = 4mA 10 = 7mA 11 = 9mA	10
DRVCON_HSSPI	[29:28]	HSSPI Port Drive strength 00 = 2mA 01 = 4mA 10 = 7mA 11 = 9mA	11
DRVCON_HSMMC	[27:26]	HSMMC Port Drive strength 00 = 2mA 01 = 4mA 10 = 7mA 11 = 9mA	11
DRVCON_LCD	[25:24]	LCD Port Drive strength 00 = 2mA 01 = 4mA 10 = 7mA 11 = 9mA	11
DRVCON_MODEM	[23:22]	MODEM Port Drive strength 00 = 2mA 01 = 4mA 10 = 7mA 11 = 9mA	11
Reserved	[21]	Reserved	0
nRSTOUT_OEN	[20]	Reset Out pin(XnRSTOUT) output enable 0 = enable 1 = disable (hi-Z)	0
DRVCON_SPICLK1	[19:18]	SPICLK[1] Port Drive strength 00 = 2mA 01 = 4mA 10 = 7mA 11 = 9mA	11
MEM1_DQS_PUD	[17:16]	Memory Port 1 DQS pin pull-up/down control 00 = disabled 01 = pull-down 10 = pull-up 11 = Reserved	01
MEM1_D_PUD1	[15:14]	Memory Port 1 Data pin [31:16] pull-up/down control 00 = disabled 01 = pull-down 10 = pull-up 11 = Reserved.	00
MEM1_D_PUD0	[13:12]	Should be set 1	01
Reserved	[11:10]	Reserved	-
MEM0_D_PUD	[9:8]	Memory Port 0(ROM bank) Data pin pull-up/down control 00 = disabled 01 = pull-down 10 = pull-up 11 = Reserved.	01
USBH_DMPD	[7]	USB_Host DM Pull-down[3] control 0 = disable 1 = enable	0
USBH_DPPD	[6]	USB_Host DP Pull-down[3] control 0 = disable 1 = enable	0
USBH_PUSW2	[5]	USB_Host Pull-up[2] switch2 is controlable when USB_PUSW1 is on. 0 = off 1 = on	0
USBH_PUSW1	[4]	USB_Host Pull-up[1] switch1 control 0 = off 1 = on	0
USBH_SUSPND	[3]	Make USB Tranceiver PAD to enter suspend mode. 0 = Normal mode 1 = Suspend mode	0
Reserved	[2]	Reserved	0
LCD_SEL	[1:0]	Select LCD I/F pin configure	00

		00 = Host I/F style 10 = 601/656 sytle	01 = RGB I/F style 11 = reserved.	

NOTES:

1. Pull-up resistance is 1.2kohm. Refer to figure 10-2.
2. Pull-up resistance is 0.5kohm. Refer to figure 10-2.
3. Pull-down resistance is 20kohm. Refer to figure 10-2.

Pin	LCD_SEL[1:0]			
	00	01	10	11
XvVD[6:0]	XvSYS_VD[6:0]	XvRGBVD[6:0]	VEN_DATA[6:0]	Reserved
XvVD[7]	XvSYS_VD[7]	XvRGBVD[7]	VEN_DATA[7]	
XvVD[13:8]	XvSYS_VD[13:8]	XvRGBVD[13:8]	V656_DATA[5:0]	
XvVD[15:14]	XvSYS_VD[15:14]	XvRGBVD[15:14]	V656_DATA[7:6]	
XvVD[17:16]	XvSYS_VD[17:16]	XvRGBVD[17:16]		
XvVD[20:18]		XvRGBVD[20:18]		
XvVD[21]		XvRGBVD[21]		
XvVD[22]	XvSYS_VSYNC_ldi	XvRGBVD[22]	V656_CLK	
XvVD[23]	XvSYS_OEn	XvRGBVD[23]	VEN_FIELD	
XvHSYNC	XvSYS_CSn_main	XvHSYNC	VEN_HSYNC	
XvVSYNC	XvSYS_CSn_sub	XvVSYNC	VEN_VSYNC	
XvVDEN	XvSYS_RS	XvVDEN	VEN_HREF	
XvVCLK	XvSYS_WEn	XvVCLK	V601_CLK	

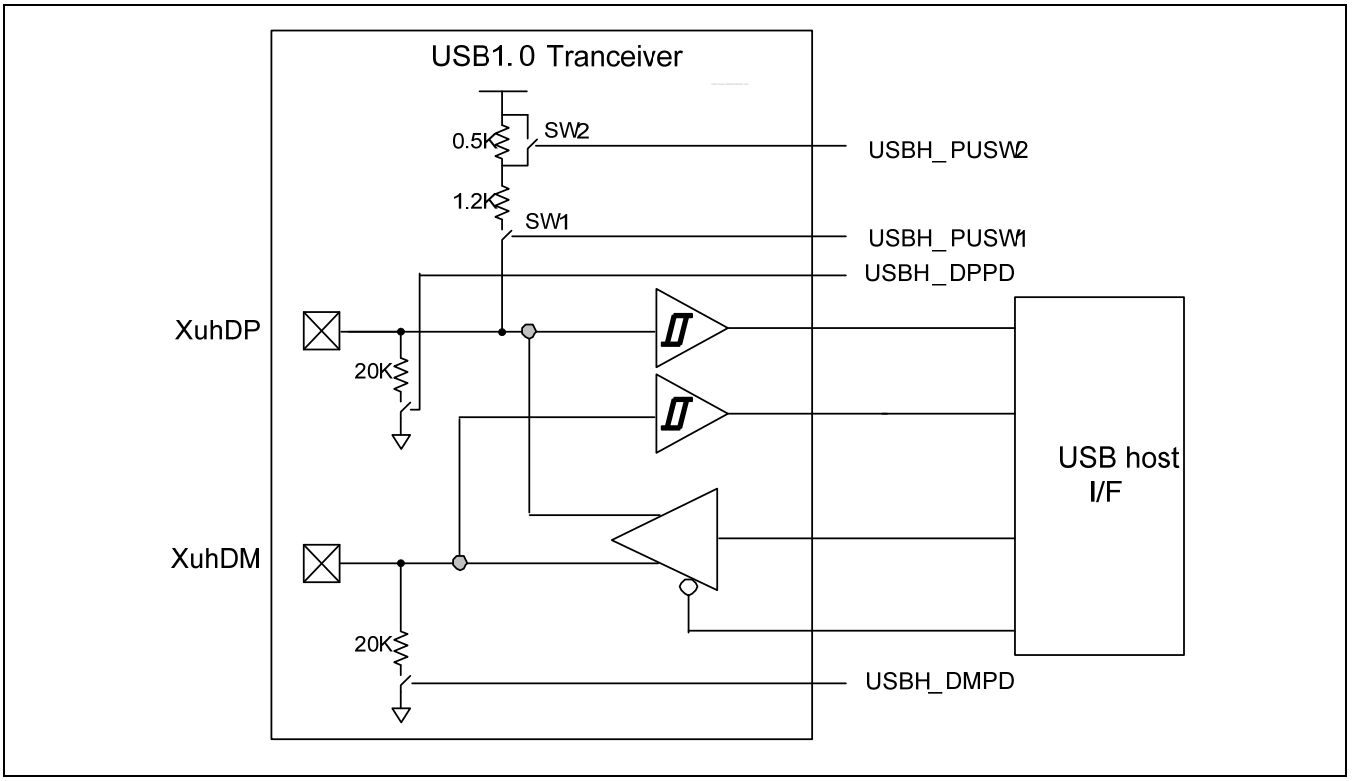


Figure 10-2. USB transceiver block diagram

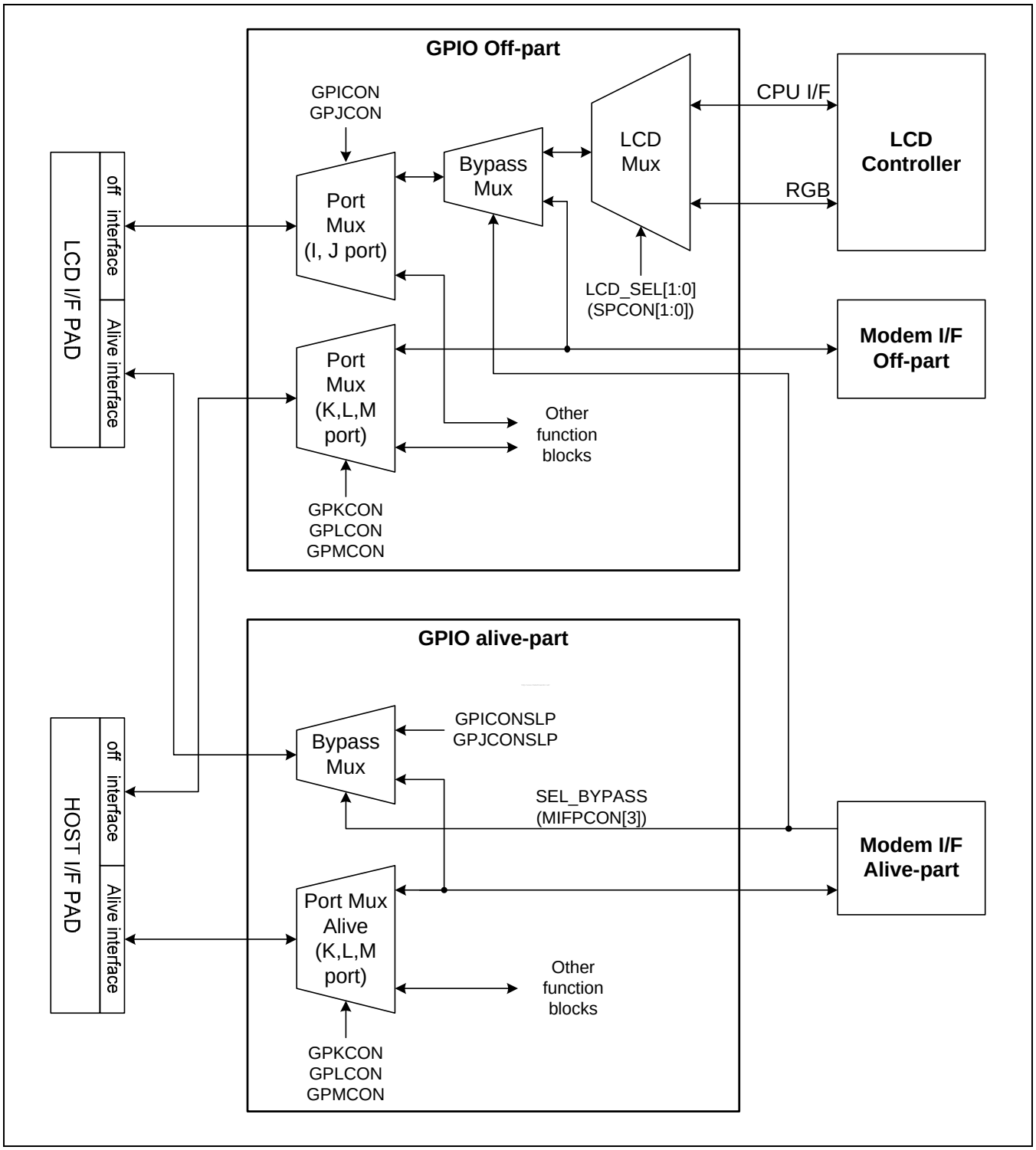


Figure 10-3. Diagram of LCD bypass logic

10.5.19 MEMORY INTERFACE PIN CONFIGURATION REGISTER IN STOP MODE

Register	Address	R/W	Description	Reset Value
MEM0CONSTOP	0x7F0081B0	R/W	Memory Port 0 Configuration Register	0x0
MEM1CONSTOP	0x7F0081B4	R/W	Memory Port 1 Configuration Register	0x0

MEM0CONSTOP	Bit	Description	Initial State
Reserved	[31:29]	Reserved	000
MEM0_RESET	[28]	Memory port 0 RESET pin (Xm0RESET) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_RP	[27]	Memory port 0 RP pin (Xm0RP) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_ADDRVLD	[26]	Memory port 0 ADDRVLD pin (Xm0ADDRVLD) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_FREn	[25]	Memor port 0 FREn pin(Xm0FREn) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_FWEn	[24]	Memor port 0 FWEn pin(Xm0FWEn) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_CLE	[23]	Memor port 0 CLE pin(Xm0CLE) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_ALE	[22]	Memor port 0 ALE pin(Xm0ALE) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_OEn	[21]	Memory port 0 Output Enable pin (Xm0OEn) Configure 0 = Previous state 1 = Hi-Z	0
Reserved	[20:18]	Must be zero	0
MEM0_BEn	[17]	Memory port 0 BEn[1:0] pin (Xm0BEn[1:0]) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_A	[16]	Memory port0 Address pin (Xm0ADDR) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_ADDR19	[15]	Memory port 0 MEM0_ADDR19 pin (Xm0ADDR19) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_ADDR18	[14]	Memory port 0 MEM0_ADDR18 pin (Xm0ADDR18) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_WEn	[13]	Memory port 0 Write Enable pin (Xm0WEn) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_CSn	[12]	Memory port 0 Chip Select pin (Xm0CSn) Configure 0 = Previous state 1 = Hi-Z	0
Reserved	[11:7]	Must be zero.	00
MEM0_ADDR17	[6]	Memory port 0 MEM0_ADDR17 pin (Xm0ADDR17) Configure 0 = Previous state 1 = Hi-Z	0
MEM0_ADDR16	[5]	Memory port 0 MEM0_ADDR16 pin (Xm0ADDR16) Configure 0 = Previous state 1 = Hi-Z	0

MEM0_SMCLK	[4]	Memory port 0 SSMC Clock pin(Xm0SMCLK) Configure 0 = Previous state 1 = Hi-Z	0
Reserved	[3:0]	Reserved	0000

MEM1CONSTOP	Bit	Description	Initial State
Reserved	[31:21]	reserved	0x0000
MEM1_SCLKn	[20]	Memory port 1 SCLKn pin (Xm1SCLKn) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_SCLK	[19]	Memory port 1 SCLK pin (Xm1SCLK) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_CKE	[18]	Memory port 1 CKE pin (Xm1CKE) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_DQM	[17]	Memory port 1 DQM pin (Xm1DQM) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_A	[16]	Memory port 1 Address pin (Xm1ADDR) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_CASn	[15]	Memory port 1 CAS pin (Xm1CASn) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_RASn	[14]	Memory port 1 RAS pin (Xm1RASn) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_WEn	[13]	Memory port 1 Write Enable pin (Xm1WEn) Configure 0 = Previous state 1 = Hi-Z	0
MEM1_CSn	[12]	Memory port 1 Chip Select pin (Xm1CSn) Configure 0 = Previous state 1 = Hi-Z	0
Reserved	[11:0]	Reserved	00

10.5.20 MEMORY INTERFACE PIN CONFIGURATION REGISTER IN SLEEP MODE

Register	Address	R/W	Description	Reset Value
MEM0CONSLP0	0x7F0081C0	R/W	Memory Port 0 pin Configuration Register 0	0x0
MEM0CONSLP1	0x7F0081C4	R/W	Memory Port 0 pin Configuration Register 1	0x0
MEM1CONSLP	0x7F0081C8	R/W	Memory Port 1 pin Configuration Register	0x0

MEM0CONSLP0	Bit	Description	Initial State
Reserved	[31:30]	Reserved	00
Reserved	[29:28]	Reserved (1)	00
Reserved	[27:26]	Reserved (1)	00
Reserved	[25:24]	Reserved (1)	00
MEM0_BEn	[23:22]	Memory port0 BEn[1:0] pin (Xm0BEn[1:0]) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_A	[21:20]	Memory port0 Address pin (Xm0ADDR) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_ADDR19	[19:18]	Memory port0 Address pin (Xm0ADDR19) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_ADDR18	[17:16]	Memory port0 Address pin (Xm0ADDR18) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_WEn	[15:14]	Memory port 0 Write Enable pin (Xm0WEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_CSn	[13:12]	Memory port 0 Chip Select pin (Xm0CSn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
Reserved	[11:8]	Reserved (2)	0000
MEM0_ADDR17	[7:6]	Memory port0 Address pin (Xm0ADDR17)Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_ADDR16	[5:4]	Memory port0 Address pin (Xm0ADDR16)Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM0_D	[3:0]	Memory Port 0 Data pin(Xm0DATA) Configure 0000 = output 0 0001 = output 1 0100 = input (hi-Z) 0101 = input pull-down enable 0110 = inupt pull-up enable 0111 = do not use	0000

		10xx = Previous state	

Note (1) GPQ[4:2] Should be use for GPIO. If not used, should be set 00 or 01

(2) GPQ[6:5] Should be use for GPIO. If not used, should be set 0000 or 0001

MEM0CONSLP1	Bit	Description	Initial State
Reserved	[31:26]	Reserved	0
MEM0_nOEata	[25:24]	ATA I/F Output Enable pin(Xm0OEata) Configure 00 = output 0, 01 = output 1 1x = input (pull-up)	0
MEM0_nWEata	[23:22]	ATA I/F Write enable pin(Xm0WEata) Configure 00 = output 0, 01 = output 1 1x = input (pull-up)	0
MEM0_SMCLK	[21:20]	ROM bank Clock pin(Xm0SMCLK) Configure 00 = output 0, 01 = output 1 1x = input (pull-up)	0
MEM0_WAIT	[19:18]	ROM bank Wait pin(Xm0WAITn) Configure 00 = output 0, 01 = output 1 1x = input (pull-up)	0
MEM0_REGata	[17:16]	Nand Flash RnB pin (Xm0REGata) Configure 00 = output 0, 01 = output 1 1x = input (Hi-Z)	0
MEM0_RESETata	[15:14]	Memory port 0 RESET pin (Xm0RESETata) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_RP_RnB	[13:12]	Memory port 0 RP pin (Xm0RP) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_ADDRVLD	[11:10]	Memory port 0 ADDRVLD pin (Xm0ADDRVLD) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_INTsm1_FREn	[9:8]	Memor port 0 FREn pin(Xm0INTsm1_FREn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_INTsm0_FWEn	[7:6]	Memor port 0 FWEn pin(Xm0INTsm0_FWEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_RDY0_CLE	[5:4]	Memor port 0 CLE pin(Xm0RDY0_CLE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_RDY1_ALE	[3:2]	Memor port 0 ALE pin(Xm0RDY1_ALE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	0
MEM0_OEn	[1:0]	Memory port 0 Output Enable pin (Xm0OEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00

MEM1CONSLP	Bit	Description	Initial State
Reserved	[31:30]	Reserved	00
MEM1_SCLKn	[29:28]	Memory port 1 SCLKn pin (Xm1SCLKn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_SCLK	[27:26]	Memory port 1 SCLK pin (Xm1SCLK) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_CKE	[25:24]	Memory port 1 CKE pin (Xm1CKE) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_DQM	[23:22]	Memory port 1 DQM pin (Xm1DQM) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_A	[21:20]	Memory port 1 Address pin (Xm1ADDR) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_CASn	[19:18]	Memory port 1 CAS pin (Xm1CASn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_RASn	[17:16]	Memory port 1 RAS pin (Xm1RASn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_WEn	[15:14]	Memory port 1 Write Enable pin (Xm1WEn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_CSn	[13:12]	Memory port 1 Chip Select pin (Xm1CSn) Configure 00 = output 0, 01 = output 1 1x = output disable (hi-Z)	00
MEM1_DQS	[11:8]	Memory port 1 DQS pin (Xm1DQS) Configure 0000 = output 0 0001 = output 1 0100 = input (hi-Z) 0101 = input pull-down enable 0110 = inupt pull-up enable 0111 = do not use 10xx = Previous state	00
MEM1_D1	[7:4]	Memory Port 1 Data pin[31:16](Xm1DATA[31:16]) Configure 0000 = output 0 0001 = output 1 0100 = input (hi-Z) 0101 = input pull-down enable 0110 = inupt pull-up enable 0111 = do not use 10xx = Previous state	00
MEM1_D0	[3:0]	Memory Port 1 Data pin[15:0](Xm1DATA[15:0]) Configure 0000 = output 0 0001 = output 1 0100 = input (hi-Z) 0101 = input pull-down enable 0110 = inupt pull-up enable 0111 = do not use 10xx = Previous state	00

10.5.21 MEMORY INTERFACE DRIVE STRENGTH CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MEM0DRVCON	0x7F0081D0	R/W	Memory Port 0 Drive strength Control Register	0x10555551
MEM1DRVCON	0x7F0081D4	R/W	Memory Port 1 Drive strength Control Register	0x555555

MEM0DRVCON	Bit	Description	Initial State
MEM0_CF	[31:30]	Memory port 0 CF pin(Xm0INTata, Xm0RESETata, Xm0REGata, Xm0WEata, Xm0OEata, Xm0CData) Configure	00
MEM0_ADDRVLD_RP	[29:28]	Memory port 0 ADDRVLD, RP pin (Xm0ADDRVLD, Xm0RP) Configure	01
MEM0_FWE_FRE	[27:26]	Memory port 0 FWEn, FREn pin (Xm0FWEn, Xm0FREn) Configure	00
MEM0_ALE_CLE	[25:24]	Memory port 0 CLE, ALE pin (Xm0CLE, Xm0ALE) Configure	00
Reserved	[23:16]	Reserved	-
MEM0_A	[15:14]	Memory port0 Address pin (Xm0ADDR) Configure	01
MEM0_BEn	[13:12]	Memory port 0 BEn pin (Xm0BEn) Configure	01
MEM0_WEn_OEn	[11:10]	Memory port 0 Write Enable, Out Enable pin (Xm0WEn, Xm0OEn) Configure	01
MEM0_ADDR19_18	[9:8]	Memory port0 Address pin (Xm0ADDR[19:18]) Configure	01
MEM0_CSn5_4	[7:6]	Memory port 0 Chip Select pin (Xm0CSn[5:4]) Configure	01
MEM0_CSn3_0	[5:4]	Memory port 0 Chip Select pin (Xm0CSn[3:0]) Configure	01
MEM0_NAND	[3:2]	Memory port 0 NAND pin(Xm0SMCLK, Xm0RDY0_ALE, Xm0RDY1_CLE, Xm0INTsm0_FWEn, Xm0INTsm1_FREn, Xm0RPn_RnB) Configure	00
MEM0_D15_0	[1:0]	Memory port 0 Data pin (Xm0DATA[15:0]) Configure	01

MEM0DRVCON	Bit	Description
	[2n+1:2n] n = 0~11	In case of VDDmem0 = 1.8V 00 = 3mA 01 = 7mA 10 = 10mA 11 = 13mA In case of VDDmem0 = 2.5V 00 = 4mA 01 = 9mA 10 = 13mA 11 = 18mA In case of VDDmem0 = 3.3V 00 = 5mA 01 = 10mA 10 = 16mA 11 = 21mA

MEM1DRVCON	Bit	Description	Initial State
Reserved	[31:24]	reserved	0x00
MEM1_SCLKn	[23:22]	Memory port 1 SCLKn pin (Xm1SCLKn) Configure	01
MEM1_DQS	[21:20]	Memory port 1 DQS pin (Xm1DQS) Configure	01
MEM1_CKE	[19:18]	Memory port 1 CKE pin (Xm1CKE) Configure	01
MEM1_SCLK	[17:16]	Memory port 1 SCLK pin (Xm1SCLK) Configure	01
MEM1_A	[15:14]	Memory port 1 Address pin (Xm1ADDR) Configure	01
MEM1_DQM	[13:12]	Memory port 1 DQM pin (Xm1DQM) Configure	01
MEM1_WEn	[11:10]	Memory port 1 Write Enable pin (Xm1WEn) Configure	01
MEM1_RASn_CASn	[9:8]	Memory port 1 RAS, CAS pin (Xm1RASn, Xm1CASn) Configure	01
MEM1_CSn1	[7:6]	Memory port 1 Chip Select pin (Xm1CSn[1]) Configure	01
MEM1_CSn0	[5:4]	Memory port 1 Chip Select pin (Xm1CSn[0]) Configure	01
MEM1_D31_16	[3:2]	Memory port 1 Data pin (Xm1DATA[31:16]) Configure	01
MEM1_D15_0	[1:0]	Memory port 1 Data pin (Xm1DATA[15:0]) Configure	01

MEM1DRVCON	Bit	Description
	[2n+1:2n] n = 0~11	In case of VDDmem1 = 1.8V 00 = 5mA 01 = 10mA 10 = 15mA 11 = 20mA In case of VDDmem1 = 2.5V 00 = 7mA 01 = 13mA 10 = 20mA 11 = 26mA

10.5.22 EXTERNAL INTERRUPT CONTROL REGISTERS

External Interrupt is consists of 10 groups numbered from 0 to 9. Only external interrupt group 0 is used for wake-up source in Stop and Sleep mode. And, In idle mode, all interrupts can be wake-up source, the other groups of external interrupts also can be the sources.

The following table is the list of external interrupt control registers. Group 0 has dedicated pins and each interrupt in this group can be controlled more detail than the other groups. For that, S3C6410 presents several registers for group 0 and they have bits for a pair of interrupt signals. In case of the other groups, there are several registers for these groups, but one register can control 2 or more groups. Digital filter count source of group 0 is FIN(Ext.Clock), and other group digital filter count is PCLK.

Register	Address	R/W	Description	Reset Value
EINT0CON0	0x7F008900	R/W	External Interrupt 0(Group0) Configuration Register 0	0x0
EINT0CON1	0x7F008904	R/W	External Interrupt 0(Group0) Configuration Register 1	0x0
EINT0FLTCON0	0x7F008910	R/W	External Interrupt 0(Group0) Filter Control Register 0	0x0
EINT0FLTCON1	0x7F008914	R/W	External Interrupt 0(Group0) Filter Control Register 1	0x0
EINT0FLTCON2	0x7F008918	R/W	External Interrupt 0(Group0) Filter Control Register 2	0x0
EINT0FLTCON3	0x7F00891C	R/W	External Interrupt 0(Group0) Filter Control Register 3	0x0
EINT0MASK	0x7F008920	R/W	External Interrupt 0(Group0) Mask Register	0x0FFFFFFF
EINT0PEND	0x7F008924	R/W	External Interrupt 0(Group0) Pending Register	0x0
EINT12CON	0x7F008200	R/W	External Interrupt 1,2(Group1,2) Configuration Register	0x0
EINT34CON	0x7F008204	R/W	External Interrupt 3,4(Group3,4) Configuration Register	0x0
EINT56CON	0x7F008208	R/W	External Interrupt 5,6(Group5,6) Configuration Register	0x0
EINT78CON	0x7F00820C	R/W	External Interrupt 7,8(Group7,8) Configuration Register	0x0
EINT9CON	0x7F008210	R/W	External Interrupt 9(Group9) Configuration Register	0x0
EINT12FLTCON	0x7F008220	R/W	External Interrupt 1,2(Group1,2) Filter Control Register	0x0
EINT34FLTCON	0x7F008224	R/W	External Interrupt 3,4(Group3,4) Filter Control Register	0x0
EINT56FLTCON	0x7F008228	R/W	External Interrupt 5,6(Group5,6) Filter Control Register	0x0
EINT78FLTCON	0x7F00822C	R/W	External Interrupt 7,8(Group7,8) Filter Control Register	0x0
EINT9FLTCON	0x7F008230	R/W	External Interrupt 9(Group9) Filter Control Register	0x0
EINT12MASK	0x7F008240	R/W	External Interrupt 1,2(Group1,2) Mask Register	0x00FF7FFF
EINT34MASK	0x7F008244	R/W	External Interrupt 3,4(Group3,4) Mask Register	0x3FFF03FF
EINT56MASK	0x7F008248	R/W	External Interrupt 5,6(Group5,6) Mask Register	0x03FF007F
EINT78MASK	0x7F00824C	R/W	External Interrupt 7,8(Group7,8) Mask Register	0x7FFFFFFF
EINT9MASK	0x7F008250	R/W	External Interrupt 9(Group9) Mask Register	0x000001FF
EINT12PEND	0x7F008260	R/W	External Interrupt 1,2(Group1,2) Pending Register	0x0
EINT34PEND	0x7F008264	R/W	External Interrupt 3,4(Group3,4) Pending Register	0x0
EINT56PEND	0x7F008268	R/W	External Interrupt 5,6(Group5,6) Pending Register	0x0

EINT78PEND	0x7F00826C	R/W	External Interrupt 7,8(Group7,8) Pending Register	0x0
EINT9PEND	0x7F008270	R/W	External Interrupt 9(Group9) Pending Register	0x0
PRIORITY	0x7F008280	R/W	Priority Control Register	0x3FF
SERVICE	0x7F008284	R	Current Service Register	0x00
SERVICEPEND	0x7F008288	R/W	Current Service Pending Register	0x00

EINT0CON0	Bit	Description	Initial State
Reserved	[31]	Reserved	0
EINT15, 14	[30:28]	Setting the signaling method of the EINT15 and EINT14. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[27]	Reserved	0
EINT13, 12	[26:24]	Setting the signaling method of the EINT13 and EINT12. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[23]	Reserved	0
EINT11, 10	[22:20]	Setting the signaling method of the EINT11 and EINT10. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[19]	Reserved	0
EINT9, 8	[18:16]	Setting the signaling method of the EINT9 and EINT8. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[15]	Reserved	0
EINT7, 6	[14:12]	Setting the signaling method of the EINT7 and EINT6. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[11]	Reserved	0
EINT5, 4	[10:8]	Setting the signaling method of the EINT5 and EINT4. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[7]	Reserved	0
EINT3, 2	[6:4]	Setting the signaling method of the EINT3 and EINT2 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT1, 0	[2:0]	Setting the signaling method of the EINT0 and EINT1 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT0CON1	Bit	Description	Initial State
Reserved	[31:23]	Reserved	000000000
EINT27, 26	[22:20]	Setting the signaling method of the EINT27 and EINT26. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[19]	Reserved	0
EINT25, 24	[18:16]	Setting the signaling method of the EINT25 and EINT24. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[15]	Reserved	0
EINT23, 22	[14:12]	Setting the signaling method of the EINT23 and EINT22. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[11]	Reserved	0
EINT21, 20	[10:8]	Setting the signaling method of the EINT21 and EINT20. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[7]	Reserved	0
EINT19, 18	[6:4]	Setting the signaling method of the EINT19 and EINT18. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT17, 16	[2:0]	Setting the signaling method of the EINT17 and EINT16. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT0FLTCON0	Bit	Description	Initial State
FLTEN	[31]	Filter Enable for EINT6, 7 0 = disables 1 = enabled	0
FLTSEL	[30]	Filter Selection for EINT6, 7 0 = delay filter 1 = digital filter(clock count)	0
EINT6, 7	[29:24]	Filtering width of EINT6, 7 This value is valid when FLTSEL is 1.	000
FLTEN	[23]	Filter Enable for EINT4, 5 0 = disables 1 = enabled	0
FLTSEL	[22]	Filter Selection for EINT4, 5 0 = delay filter 1 = digital filter(clock count)	0
EINT4, 5	[21:16]	Filtering width of EINT4, 5 This value is valid when FLTSEL is 1.	000
FLTEN	[15]	Filter Enable for EINT2, 3 0 = disables 1 = enabled	0
FLTSEL	[14]	Filter Selection for EINT2, 3 0 = delay filter 1 = digital filter(clock count)	0
EINT2, 3	[13:8]	Filtering width of EINT2, 3 This value is valid when FLTSEL is 1.	000
FLTEN	[7]	Filter Enable for EINT 0, 1 0 = disables 1 = enabled	0
FLTSEL	[6]	Filter Selection for EINT0, 1 0 = delay filter 1 = digital filter(clock count)	0
EINT0, 1	[5:0]	Filtering width of EINT0, 1 This value is valid when FLTSEL is 1.	000

EINT0FLTCON1	Bit	Description	Initial State
FLTEN	[31]	Filter Enable EINT14, 15 0 = disables 1 = enabled	0
FLTSEL	[30]	Filter Selection for EINT 14, 15 0 = delay filter 1 = digital filter(clock count)	0
EINT14, 15	[29:24]	Filtering width of EINT14, 15 This value is valid when FLTSEL is 1.	000
FLTEN	[23]	Filter Enable for EINT 12, 13 0 = disables 1 = enabled	0
FLTSEL	[22]	Filter Selection for EINT 12, 13 0 = delay filter 1 = digital filter(clock count)	0
EINT12, 13	[21:16]	Filtering width of EINT12, 13 This value is valid when FLTSEL is 1.	000
FLTEN	[15]	Filter Enable for EINT 10, 11 0 = disables 1 = enabled	0
FLTSEL	[14]	Filter Selection for EINT 10, 11 0 = delay filter 1 = digital filter(clock count)	0
EINT10, 11	[13:8]	Filtering width of EINT10, 11 This value is valid when FLTSEL is 1.	000
FLTEN	[7]	Filter Enable for EINT 8, 9 0 = disables 1 = enabled	0
FLTSEL	[6]	Filter Selection for EINT 8, 9 0 = delay filter 1 = digital filter(clock count)	0
EINT8, 9	[5:0]	Filtering width of EINT8, 9 This value is valid when FLTSEL is 1.	000

EINT0FLTCON2	Bit	Description	Initial State
FLTEN	[31]	Filter Enable for EINT 22, 23 0 = disables 1 = enabled	0
FLTSEL	[30]	Filter Selection for EINT 22, 23 0 = delay filter 1 = digital filter(clock count)	0
EINT22, 23	[29:24]	Filtering width of EINT22, 23 This value is valid when FLTSEL is 1.	000
FLTEN	[23]	Filter Enable for EINT20, 21 0 = disables 1 = enabled	0
FLTSEL	[22]	Filter Selection for EINT20, 21 0 = delay filter 1 = digital filter(clock count)	0
EINT20, 21	[21:16]	Filtering width of EINT20, 21 This value is valid when FLTSEL is 1.	000
FLTEN	[15]	Filter Enable for EINT18, 19 0 = disables 1 = enabled	0
FLTSEL	[14]	Filter Selection for EINT18, 19 0 = delay filter 1 = digital filter(clock count)	0
EINT18, 19	[13:8]	Filtering width of EINT18, 19 This value is valid when FLTSEL is 1.	000
FLTEN	[7]	Filter Enable for EINT16, 17 0 = disables 1 = enabled	0
FLTSEL	[6]	Filter Selection for EINT16, 17 0 = delay filter 1 = digital filter(clock count)	0
EINT16, 17	[5:0]	Filtering width of EINT16, 17 This value is valid when FLTSEL is 1.	000

EINT0FLTCON3	Bit	Description	Initial State
FLTEN	[15]	Filter Enable for EINT26, 27 0 = disables 1 = enabled	0
FLTSEL	[14]	Filter Selection for EINT26, 27 0 = delay filter 1 = digital filter(clock count)	0
EINT26, 27	[13:8]	Filtering width of EINT26, 27 This value is valid when FLTSEL is 1.	000
FLTEN	[7]	Filter Enable for EINT24, 25 0 = disables 1 = enabled	0
FLTSEL	[6]	Filter Selection for EINT24, 25 0 = delay filter 1 = digital filter(clock count)	0
EINT24, 25	[5:0]	Filtering width of EINT24, 25 This value is valid when FLTSEL is 1.	000

EINT0MASK	Bit	Description	Initial State
EINT27	[27]	0 = Enable Interrupt 1= Masked	1
EINT26	[26]	0 = Enable Interrupt 1= Masked	1
EINT25	[25]	0 = Enable Interrupt 1= Masked	1
EINT24	[24]	0 = Enable Interrupt 1= Masked	1
EINT23	[23]	0 = Enable Interrupt 1= Masked	1
EINT22	[22]	0 = Enable Interrupt 1= Masked	1
EINT21	[21]	0 = Enable Interrupt 1= Masked	1
EINT20	[20]	0 = Enable Interrupt 1= Masked	1
EINT19	[19]	0 = Enable Interrupt 1= Masked	1
EINT18	[18]	0 = Enable Interrupt 1= Masked	1
EINT17	[17]	0 = Enable Interrupt 1= Masked	1
EINT16	[16]	0 = Enable Interrupt 1= Masked	1
EINT15	[15]	0 = Enable Interrupt 1= Masked	1
EINT14	[14]	0 = Enable Interrupt 1= Masked	1
EINT13	[13]	0 = Enable Interrupt 1= Masked	1
EINT12	[12]	0 = Enable Interrupt 1= Masked	1
EINT11	[11]	0 = Enable Interrupt 1= Masked	1
EINT10	[10]	0 = Enable Interrupt 1= Masked	1
EINT9	[9]	0 = Enable Interrupt 1= Masked	1
EINT8	[8]	0 = Enable Interrupt 1= Masked	1
EINT7	[7]	0 = Enable Interrupt 1= Masked	1
EINT6	[6]	0 = Enable Interrupt 1= Masked	1
EINT5	[5]	0 = Enable Interrupt 1= Masked	1
EINT4	[4]	0 = Enable Interrupt 1= Masked	1
EINT3	[3]	0 = Enable Interrupt 1= Masked	1
EINT2	[2]	0 = Enable Interrupt 1= Masked	1
EINT1	[1]	0 = Enable Interrupt 1= Masked	1
EINT0	[0]	0 = Enable Interrupt 1= Masked	1

EINT0PEND	Bit	Description	Initial State
EINT27	[27]	0 = Not occur 1= Occur interrupt	0
EINT26	[26]	0 = Not occur 1= Occur interrupt	0
EINT25	[25]	0 = Not occur 1= Occur interrupt	0
EINT24	[24]	0 = Not occur 1= Occur interrupt	0
EINT23	[23]	0 = Not occur 1= Occur interrupt	0
EINT22	[22]	0 = Not occur 1= Occur interrupt	0
EINT21	[21]	0 = Not occur 1= Occur interrupt	0
EINT20	[20]	0 = Not occur 1= Occur interrupt	0
EINT19	[19]	0 = Not occur 1= Occur interrupt	0
EINT18	[18]	0 = Not occur 1= Occur interrupt	0
EINT17	[17]	0 = Not occur 1= Occur interrupt	0
EINT16	[16]	0 = Not occur 1= Occur interrupt	0
EINT15	[15]	0 = Not occur 1= Occur interrupt	0
EINT14	[14]	0 = Not occur 1= Occur interrupt	0
EINT13	[13]	0 = Not occur 1= Occur interrupt	0
EINT12	[12]	0 = Not occur 1= Occur interrupt	0
EINT11	[11]	0 = Not occur 1= Occur interrupt	0
EINT10	[10]	0 = Not occur 1= Occur interrupt	0
EINT9	[9]	0 = Not occur 1= Occur interrupt	0
EINT8	[8]	0 = Not occur 1= Occur interrupt	0
EINT7	[7]	0 = Not occur 1= Occur interrupt	0
EINT6	[6]	0 = Not occur 1= Occur interrupt	0
EINT5	[5]	0 = Not occur 1= Occur interrupt	0
EINT4	[4]	0 = Not occur 1= Occur interrupt	0
EINT3	[3]	0 = Not occur 1= Occur interrupt	0
EINT2	[2]	0 = Not occur 1= Occur interrupt	0
EINT1	[1]	0 = Not occur 1= Occur interrupt	0
EINT0	[0]	0 = Not occur 1= Occur interrupt	0

NOTES: 1. Each bit is cleared by writing "1"
2. EINT0~27 of Group0 are wake-up source in stop and sleep mode.
3. When EINT0~27 are used as wake-up sources, EINT_mask(System controller register) must be set un-mask

EINT12CON	Bit	Description	Initial State
Reserved	[31:23]	Reserved	0
EINT2[7:4]	[22:20]	Setting the signaling method of the EINT2[7:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[19]	Reserved	0
EINT2[3:0]	[18:16]	Setting the signaling method of the EINT2[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[15]	Reserved	0
EINT1[14:12]	[14:12]	Setting the signaling method of the EINT1[14:12]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[11]	Reserved	0
EINT1[11:8]	[10:8]	Setting the signaling method of the EINT1[11:8]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[7]	Reserved	0
EINT1[7:4]	[6:4]	Setting the signaling method of the EINT1[7:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT1[3:0]	[2:0]	Setting the signaling method of the EINT1[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT34CON	Bit	Description	Initial State
Reserved	[31]	Reserved	0
EINT4[13:12]	[30:28]	Setting the signaling method of the EINT4[13:12]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[27]	Reserved	0
EINT4[11:8]	[26:24]	Setting the signaling method of the EINT4[11:8]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[23]	Reserved	0
EINT4[7:4]	[22:20]	Setting the signaling method of the EINT4[7:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[19]	Reserved	0
EINT4[3:0]	[18:16]	Setting the signaling method of the EINT4[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[15:8]	Reserved	0x00
Reserved	[7]	Reserved	0
EINT3[4]	[6:4]	Setting the signaling method of the EINT3[4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT3[3:0]	[2:0]	Setting the signaling method of the EINT3[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT56CON	Bit	Description	Initial State
Reserved	[31:27]	Reserved	0
EINT6[9:8]	[26:24]	Setting the signaling method of the EINT6[9:8]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[23]	Reserved	0
EINT6[7:4]	[22:20]	Setting the signaling method of the EINT6[7:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[19]	Reserved	0
EINT6[3:0]	[18:16]	Setting the signaling method of the EINT6[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[15:7]	Reserved	0
EINT5[6:4]	[6:4]	Setting the signaling method of the EINT5[6:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT5[3:0]	[2:0]	Setting the signaling method of the EINT5[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT78CON	Bit	Description	Initial State
Reserved	[31]	Reserved	0
EINT8[14:12]	[30:28]	Setting the signaling method of the EINT8[14:12]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[27]	Reserved	0
EINT8[11:8]	[26:24]	Setting the signaling method of the EINT8[11:8]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[23]	Reserved	0
EINT8[7:4]	[22:20]	Setting the signaling method of the EINT8[7:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[19]	Reserved	0
EINT8[3:0]	[18:16]	Setting the signaling method of the EINT8[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[15]	Reserved	0
EINT7[15:12]	[14:12]	Setting the signaling method of the EINT7[15:12]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[11]	Reserved	0
EINT7[11:8]	[10:8]	Setting the signaling method of the EINT7[11:8]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[7]	Reserved	0
EINT7[7:4]	[6:4]	Setting the signaling method of the EINT7[7:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT7[3:0]	[2:0]	Setting the signaling method of the EINT7[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT9CON	Bit	Description	Initial State
Reserved	[31:7]	Reserved	0x000000
EINT9[8:4]	[6:4]	Setting the signaling method of the EINT9[8:4]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
Reserved	[3]	Reserved	0
EINT9[3:0]	[2:0]	Setting the signaling method of the EINT9[3:0]. 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

EINT12FLTCON	Bit	Description	Initial State
Reserved	[31:24]	Rreserved	0x00
FLTEN2[7:0]	[23]	Filter Enable for EINT2[7:0] 0 = disables 1 = enabled	0
EINT2[7:0]	[22:16]	Filtering width of EINT2[7:0]	000
FLTEN1[14:8]	[15]	Filter Enable for EINT1[14:8] 0 = disables 1 = enabled	0
EINT1[14:8]	[14:8]	Filtering width of EINT1[14:8]	000
FLTEN1[7:0]	[7]	Filter Enable for EINT1[7:0] 0 = disables 1 = enabled	0
EINT1[7:0]	[6:0]	Filtering width of EINT1[7:0]	000

EINT34FLTCON	Bit	Description	Initial State
FLTEN4[13:8]	[31]	Filter Enable for EINT2[12:8] 0 = disables 1 = enabled	0
EINT4[13:8]	[30:24]	Filtering width of EINT2[12:8]	000
FLTEN4[7:0]	[23]	Filter Enable for EINT2[7:0] 0 = disables 1 = enabled	0
EINT4[7:0]	[22:16]	Filtering width of EINT2[7:0]	000
Reseved	[15:8]	Reserved	0x00
FLTEN3[4:0]	[7]	Filter Enable for EINT3[4:0] 0 = disables 1 = enabled	0
EINT3[4:0]	[6:0]	Filtering width of EINT3[4:0]	000

EINT56FLTCON	Bit	Description	Initial State
FLTEN6[9:8]	[31]	Filter Enable for EINT6[9:8] 0 = disables 1 = enabled	0
EINT6[9:8]	[30:24]	Filtering width of EINT6[9:8]	000
FLTEN6[7:0]	[23]	Filter Enable for EINT6[7:0] 0 = disables 1 = enabled	0
EINT6[7:0]	[22:16]	Filtering width of EINT6[7:0]	000
Reserved	[15:8]	Reserved	0x00
FLTEN5[6:0]	[7]	Filter Enable for EINT5[6:0] 0 = disables 1 = enabled	0
EINT5[6:0]	[6:0]	Filtering width of EINT5[6:0]	000

EINT78FLTCON	Bit	Description	Initial State
FLTEN8[15:8]	[31]	Filter Enable for EINT8[15:8] 0 = disables 1 = enabled	0
EINT8[15:8]	[30:24]	Filtering width of EINT8[15:8]	000
FLTEN8[7:0]	[23]	Filter Enable for EINT8[7:0] 0 = disables 1 = enabled	0
EINT8[7:0]	[22:16]	Filtering width of EINT8[7:0]	000
FLTEN7[15:8]	[15]	Filter Enable for EINT7[15:8] 0 = disables 1 = enabled	0
EINT7[15:8]	[14:8]	Filtering width of EINT7[15:8]	000
FLTEN7[7:0]	[7]	Filter Enable for EINT7[7:0] 0 = disables 1 = enabled	0
EINT7[7:0]	[6:0]	Filtering width of EINT7[7:0]	000

EINT9FLTCON	Bit	Description	Initial State
Reserved	[15:8]	Reserved	0x00
FLTEN9[8:0]	[7]	Filter Enable for EINT9[8:0] 0 = disables 1 = enabled	0
EINT9[8:0]	[6:0]	Filtering width of EINT9[8:0]	000

EINT12MASK	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0
EINT2[m]	[16+m] m = 0 ~ 7	0 = Enable Interrupt 1= Masked	1
Reserved	[15]	Reserved	0
EINT1[n]	[n] n = 0 ~ 14	0 = Enable Interrupt 1= Masked	1

EINT34MASK	Bit	Description	Initial State
Reserved	[31:30]	Reserved	0
EINT4[m]	[16+m] m = 0 ~ 13	0 = Enable Interrupt 1= Masked	1
Reserved	[15:5]	Reserved	0
EINT3[n]	[n] n = 0 ~ 4	0 = Enable Interrupt 1= Masked	1

EINT56MASK	Bit	Description	Initial State
Reserved	[31:26]	Reserved	0
EINT6[m]	[16+m] m = 0 ~ 9	0 = Enable Interrupt 1= Masked	1
Reserved	[15:7]	Reserved	0
EINT5[n]	[n] n = 0 ~ 6	0 = Enable Interrupt 1= Masked	1

EINT78MASK	Bit	Description	Initial State
EINT8[m]	[16+m] m = 0 ~ 14	0 = Enable Interrupt 1= Masked	1
EINT7[n]	[n] n = 0 ~ 15	0 = Enable Interrupt 1= Masked	1

EINT9MASK	Bit	Description	Initial State
Reserved	[31:9]	Reserved	0
EINT9[n]	[n] n = 0 ~ 8	0 = Enable Interrupt 1= Masked	1

EINT12PEND	Bit	Description	Initial State
Reserved	[31:24]	Reserved	0
EINT2[m]	[16+m] m = 0 ~ 7	0 = Not occur 1= Occur interrupt	0
Reserved	[15]	Reserved	0
EINT1[n]	[n] n = 0 ~ 14	0 = Not occur 1= Occur interrupt	0

EINT34PEND	Bit	Description	Initial State
Reserved	[31:30]	Reserved	0
EINT4[m]	[16+m] m = 0 ~ 13	0 = Not occur 1= Occur interrupt	0
Reserved	[15:5]	Reserved	0
EINT3[n]	[n] n = 0 ~ 4	0 = Not occur 1= Occur interrupt	0

EINT56PEND	Bit	Description	Initial State
Reserved	[31:26]	Reserved	0
EINT6[m]	[16+m] m = 0 ~ 9	0 = Not occur 1= Occur interrupt	0
Reserved	[15:7]	Reserved	0
EINT5[n]	[n] n = 0 ~ 6	0 = Not occur 1= Occur interrupt	0

EINT78PEND	Bit	Description	Initial State
EINT8[m]	[16+m] m = 0 ~ 14	0 = Not occur 1= Occur interrupt	0
EINT7[n]	[n] n = 0 ~ 15	0 = Not occur 1= Occur interrupt	0

EINT9PEND	Bit	Description	Initial State
Reserved	[31:9]	Reserved	0
EINT9[n]	[n] n = 0 ~ 8	0 = Not occur 1= Occur interrupt	0

PRIORITY REGISTER (PRIORITY)

Register	Address	R/W	Description	Reset Value
PRIORITY	0x7F008280	R/W	External Interrupt priority control register	0x000003FF

PRIORITY	Bit	Description	Initial State
ARB9	[9]	External Interrupt Group 9 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB8	[8]	External Interrupt Group 8 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB7	[7]	External Interrupt Group 7 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB6	[6]	External Interrupt Group 6 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB5	[5]	External Interrupt Group 5 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB4	[4]	External Interrupt Group 4 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB3	[3]	External Interrupt Group 3 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB2	[2]	External Interrupt Group 2 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB1	[1]	External Interrupt Group 1 priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1
ARB0	[0]	External Interrupt Groups priority rotate enable 0 = Priority does not rotate, 1 = Priority rotate enable	1

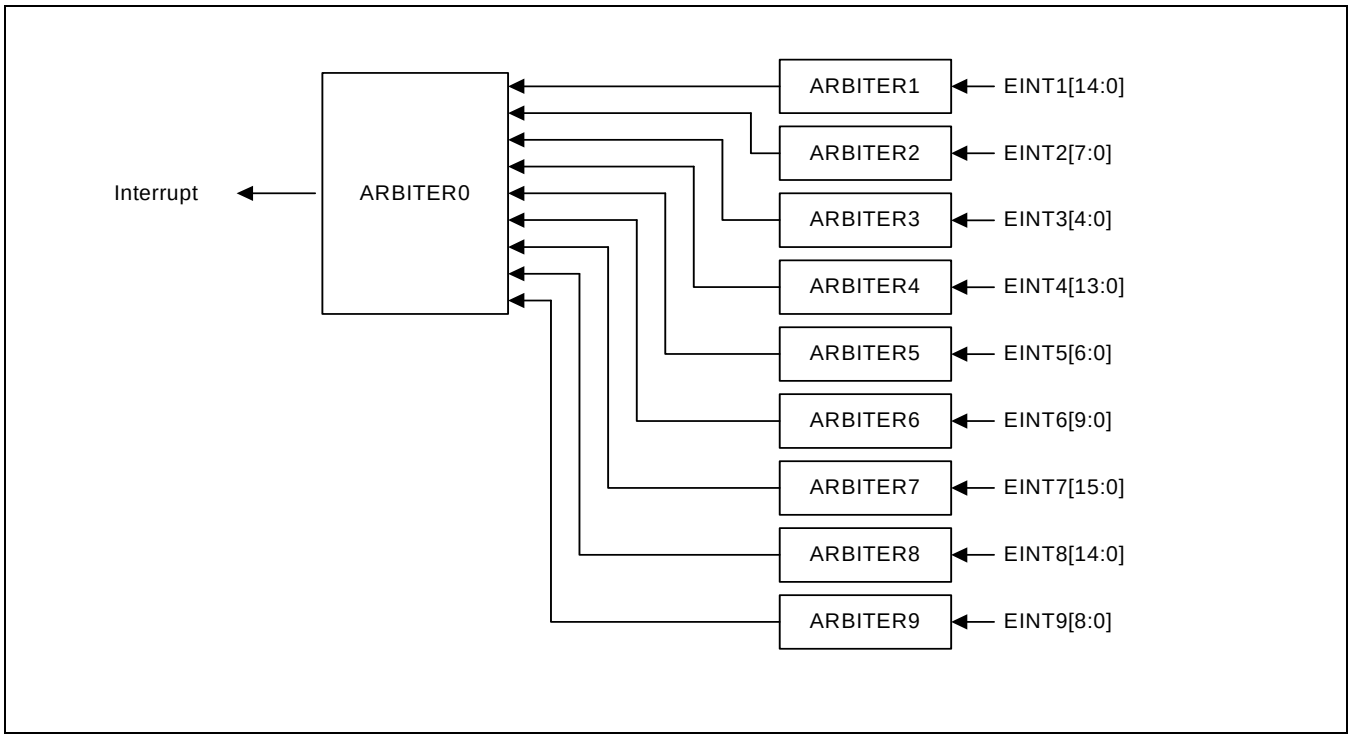


Figure 10-4. Priority Generating Block

10.6 CURRENT SERVICE REGISTER (SERVICE)

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

Current Service Pending Register presents which interrupt pending bit must be cleared. Only 1 bit is set 1 corresponding to Current Service Register. After complete interrupt service routine, you can clear the interrupt pending bit in Interrupt Pending Registers by write the value contained in this register.
For example, if the group field of Current Service Register is 4, you can clear the corresponding interrupt pending bit by writing the value of SERVICEPEND to EINT34PEND.

Register	Address	R/W	Description	Reset Value
SERVICE	0x7F008284	R	Current Service Register	0x00
SERVICEPEND	0x7F008288	R	Current Service Pending Register	0x00

SERVICE	Bit	Description	Initial State
Group	[7:4]	Group Number	0000
Interrupt No.	[3:0]	Interrupt number to be serviced. This is valid only when Group field is not 00.	0000

10.6.1 EXTERN PIN CONFIGURATION REGISTER IN SLEEP MODE

These registers keep their values during sleep mode.

Register	Address	R/W	Description	Reset Value
SPCONSLP	0x7F008880	R/W	Special Port Sleep mode configure Register	0x0000010
SLPEN	0x7F008930	R/W	Sleep mode Pad Configuer Register.	0x00

SPCONSLP	Bit	Description	Initial State
Reserved	[31:15]	Reserved	0
TDOPULLDOWN	[14]	XjTDO Pad Pull-Down Control at STOP mode 0 = Disable 1 = Pull-Down Enable	0
RSTOUT	[13:12]	Reset Out pin(XnRSTOUT) Configure 00 = output 0 01 = output 1 1x = output disable (hi-Z)	00
Reserved	[11:6]	Reserved	00
CKE1_INIT	[5]	Initial value for Memory port 1 CKE This value is valid only when system is in reset state of power-on or sleep-wakeup..	0
Reserved	[4:2]	Reserved	00
KP_COL	[1:0]	Key Pad Column bit Configure 00 = output 0 01 = output 1 1x = input	00

SLPEN	Bit	Description	Initial State
Reserved	[7:2]	reserved	0
SLPEN_CFG	[1]	0 : Automatically by Sleep mode 1: by SLPEN bit.	0
SLPEN	[0]	Sleep mode Pad state enable Register When this bit is set to ‘1’, external pins are controlled by sleep mode control register such as ACONSLP, MEM0CONSLP, etc. This bit is set to ‘1’ automatically when system enters into sleep mode and can be cleared by writing ‘0’ to this bit or cold reset (by XnRESET pin). After waken up from sleep mode, this bit maintains the value ‘1’.	00

11 DMA CONTROLLER

This chapter describes the DMA controller for the S3C6410 RISC microprocessor.

11.1 OVERVIEW

S3C6410 contains four DMA controllers. Each DMA controller consists of 8 transfer channels. Each channel of DMA controller can perform data movements between devices in the AXI SYSTEM bus and/or AXI PERIPHERAL bus through AHBtoAXI bridges without any restrictions. In other words, each channel can handle the following four cases:

1) Both source and destination are in the SYSTEM bus.
2) Source is available in the SYSTEM bus while destination is available in the PERIPHERAL bus.
3) Source is available in the PERIPHERAL bus while destination is available in the SYSTEM bus.
4) Both source and destination are available in the PERIPHERAL bus.

ARM PrimeCell DMA controller PL080 is used as S3C6410 DMA controller. The DMAC is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant *System-on-Chip* (SoC) peripheral that is developed, tested, and licensed by ARM Limited. The DMAC is an AMBA AHB module, and connects to the *Advanced High-performance Bus* (AHB).

The main advantage of DMA is that it can transfer the data without CPU intervention. The operation of DMA can be initiated by S/W, or the request from internal peripherals, or the external request pins.

11.2 FEATURES

The DMA controller provides the following features:

- S3C6410 contains four DMA controllers. Each DMA controller consists of 8 transfer channels. Each channel can support a unidirectional transfer.
- Each DMA controller provides 16 peripheral DMA request lines.
- Each peripheral connected to the DMAC can assert either a burst DMA request or a single DMA request. The DMA burst size is set by programming the DMAC.
- Supports Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral transfers.
- Scatter or gather DMA is supported through the use of linked lists.
- Hardware DMA channels priority. Each DMA channel has a specific hardware priority. DMA channel 0 has the highest priority down to channel 7 which has the lowest priority. If requests from two channels become active at the same time the channel with the highest priority is serviced first.
- The DMAC is programmed by writing to the DMA control registers over the AHB slave interface.
- Two AXI bus masters pass through AHB to AXI bridges for transferring data. These interfaces are used to transfer data when a DMA request goes active.
- Incrementing or non-incrementing address for source and destination.
- Programmable DMA burst size. The DMA burst size can be programmed to increase efficiency of transfer data. Usually the burst size is set to half the size of the FIFO in the peripheral.
- Internal four word FIFO per channel.
- Supports 8, 16 and 32-bit wide transactions.
- Separate and combined DMA error and DMA count interrupt requests. An interrupt to the processor can be generated on a DMA error or when a DMA count has reached 0 (this is usually used to indicate that a transfer has finished). Three interrupt request signals are used to do this:
  - **DMACINTTC** signals when a transfer has completed.
  - **DMACINTERR** signals when an error has occurred.
  - **DMACINTR** combines both the **DMACINTTC** and **DMACINTERR** interrupt request signals. The **DMACINTR** interrupt request can be used in systems, which have few interrupt controller request inputs.
- Interrupt masking. The DMA error and DMA terminal count interrupt requests can be masked.
- Raw interrupt status. The DMA error and DMA count raw interrupt status can be read prior to masking.

11.3 BLOCK DIAGRAM

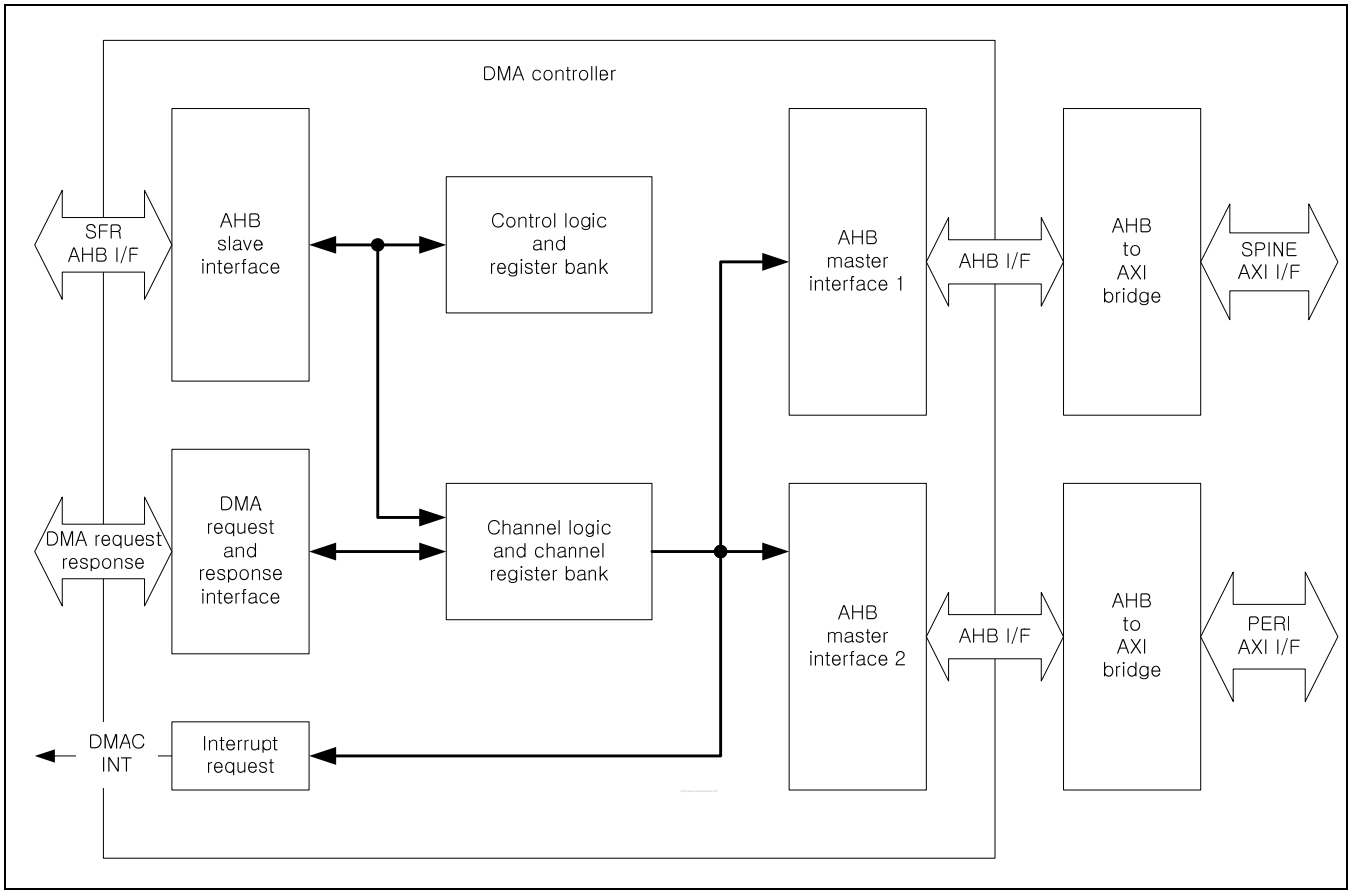


Figure 11-1. DMAC block diagram

11.4 DMA SOURCES

The S3C6410 supports 64 DMA sources as listed in the table below.
Reset value of SDMA_SEL register in System Controller is 0x0 which means SDMA selection. So, the configuration should be set to 1 in order to use general DMA. For more information, please refer to System Controller part.

Group	DMA No.	Sources	Description
DMA0, SDMA0	0	DMA_UART0[0]	UART0 DMA source 0
DMA0, SDMA0	1	DMA_UART0[1]	UART0 DMA source 1
DMA0, SDMA0	2	DMA_UART1[0]	UART1 DMA source 0
DMA0, SDMA0	3	DMA_UART1[1]	UART1 DMA source 1
DMA0, SDMA0	4	DMA_UART2[0]	UART2 DMA source 0
DMA0, SDMA0	5	DMA_UART2[1]	UART2 DMA source 1
DMA0, SDMA0	6	DMA_UART3[0]	UART3 DMA source 0
DMA0, SDMA0	7	DMA_UART3[1]	UART3 DMA source 1
DMA0, SDMA0	8	DMA_PCM0_TX	PCM0 DMA TX source
DMA0, SDMA0	9	DMA_PCM0_RX	PCM0 DMA RX source
DMA0, SDMA0	10	DMA_I2S0_TX	I2S0 TX DMA source
DMA0, SDMA0	11	DMA_I2S0_RX	I2S0 RX DMA source
DMA0, SDMA0	12	DMA_SPI0_TX	SPI0 TX DMA source
DMA0, SDMA0	13	DMA_SPI0_RX	SPI0 RX DMA source
DMA0, SDMA0	14	DMA_HSI_I2SV40_TX	I2S_V40 TX source or MIPI HSI DMA TX source
DMA0, SDMA0	15	DMA_HSI_I2SV40_RX	I2S_V40 RX source or MIPI HSI DMA RX source
DMA1, SDMA1	0	DMA_PCM1_TX	PCM1 DMA TX source
DMA1, SDMA1	1	DMA_PCM1_RX	PCM1 DMA RX source
DMA1, SDMA1	2	DMA_I2S1_TX	I2S1 TX DMA source
DMA1, SDMA1	3	DMA_I2S1_RX	I2S1 RX DMA source
DMA1, SDMA1	4	DMA_SPI1_TX	SPI1 TX DMA source
DMA1, SDMA1	5	DMA_SPI1_RX	SPI1 RX DMA source
DMA1, SDMA1	6	DMA_AC_PCMout	AC97 PCMout DMA source
DMA1, SDMA1	7	DMA_AC_PCMin	AC97 PCMin DMA source
DMA1, SDMA1	8	DMA_AC_MICin	AC97 MICin DMA source
DMA1, SDMA1	9	DMA_PWM	PWM DMA source
DMA1, SDMA1	10	DMA_IrDA	IrDA DMA source
DMA1, SDMA1	11	DMA_EXTERNAL	External DMA source
DMA1, SDMA1	12	Reserved	
DMA1, SDMA1	13	Reserved	
SDMA1	14	DMA_SECU_RX	Security RX DMA source
SDMA1	15	DMA_SECU_TX	Security TX DMA source

11.5 DMA INTERFACE

11.5.1 DMA REQUEST SIGNALS

The DMA request signals are used by peripherals to request a data transfer. The DMA request signals indicate:

- Whether a single word or a burst (multi-word)transfer of data is required
- Whether the transfer is the last in the data packet.

The DMA request signals to the DMA controller for each peripheral are as follows:

DMACxBREQ : Burst request signal. This executes a programmed burst number of words to be transferred.
DMACxSREQx: Single transfer request signal. This executes a single word to be transferred. The DMA controller transfers a single word to, or from the peripheral.

11.5.2 DMA RESPONSE SIGNALS

The DMA response signals indicate whether the transfer initiated by the DMA request signal is complete. The response signals can also be used to indicate whether a complete packet has been transferred.

The DMA response signals from the DMA controller for each peripheral are as follows:

DMACxCLRx: DMA clear or acknowledge signal.
DMACxTC: DMA terminal count signal.

The **DMACxCLRx** signal is used by the DMA controller to acknowledge a DMA request from the peripheral.

The **DMACxTC** signal is used by the DMA controller to indicate to the peripheral that the DMA transfer is complete.

11.5.3 TRANSFER TYPES

The DMA controller supports four types of transfer:

- memory-to-peripheral
- peripheral-to-memory
- memory-to-memory
- peripheral-to-peripheral.

Each transfer type can have either the peripheral or the DMA controller as the flow controller. Therefore there are four possible control scenarios.

11.5.3.1 Peripheral-to-memory transaction under DMA controller flow control

For transactions that are not a multiple of the burst size, use both the burst and single request signals as shown in Figure 11-2.

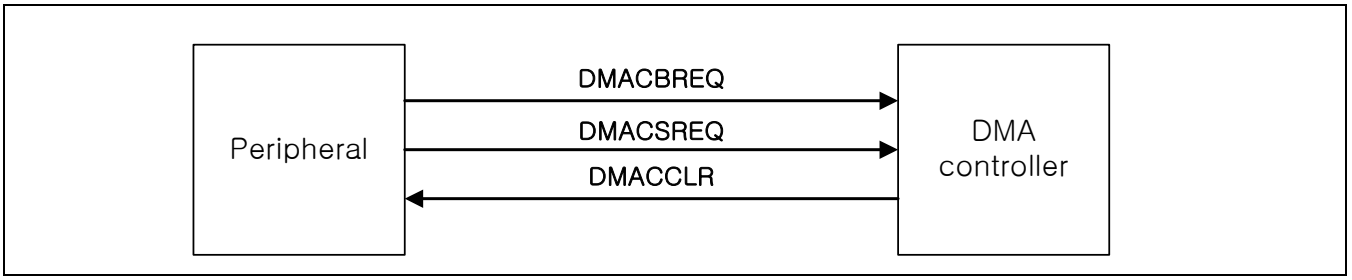


Figure 11-2. Peripheral-to-memory transaction comprising bursts and single requests

The two request signals are not mutually exclusive. The DMA controller monitors **DMACBREQ**, while the amount of data left to transfer is greater than the burst size, and commences a burst transfer (from the peripheral) when requested. When the amount of data left is less than the burst size, the DMA controller monitors **DMACSREQ** and commences single transfers when requested.

11.5.3.2 Memory-to- Peripheral transaction under DMA controller flow control

For transactions that are not a multiple of the burst size, use only the burst request signal as shown in Figure 11-3. The DMAC works out the amount of data to transfer based on the transfer size.

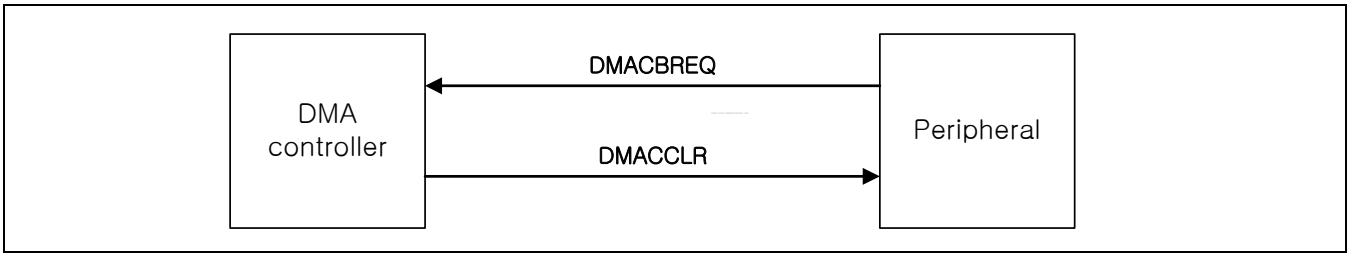


Figure 11-3. Memory-to-Peripheral transaction comprising bursts that are not multiples of the burst size

Only **DMACBREQ** is required. The DMA controller transfers full bursts of data while the amount of data left to transfer is greater than the burst size. When the amount of data left is less than the burst size, the DMAC again monitors **DMACBREQ** and transfers the rest of the data when requested.

11.5.3.3 Memory-to-memory transaction under DMA controller flow control

Software programs a DMA channel memory-to-memory transfer. When it is enabled, the DMA channel commences transfers without DMA request. It continues until one of the following occurs:

- All the data is transferred.
- The channel is disabled by software.

NOTE:

You must program memory-to-memory transfers with a low channel priority, otherwise the other DMA channels cannot access the bus until the memory-to-memory transfer has finished, or other AHB masters cannot perform any transaction.

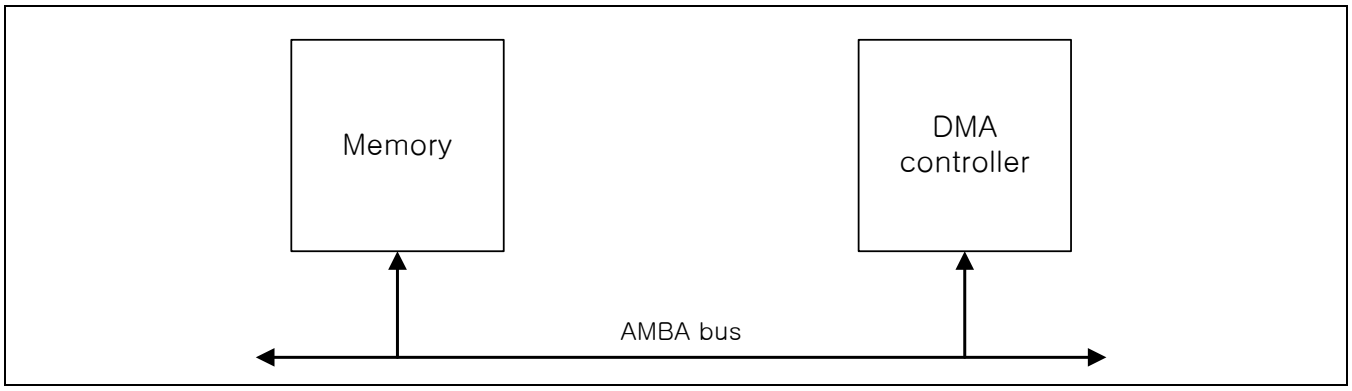


Figure 11-4. Memory-to-memory transaction under DMA flow control

11.5.3.4 Peripheral-to-peripheral transaction under DMA controller flow control

When the transaction is not a multiple of the burst size, use the following signals:

- The single and burst request signals (**DMACBREQ** and **DMACSREQ**)of the source peripheral
- The burst request signal (**DMACBREQ**) of the destination peripheral.

This is shown in Figure 11-4.

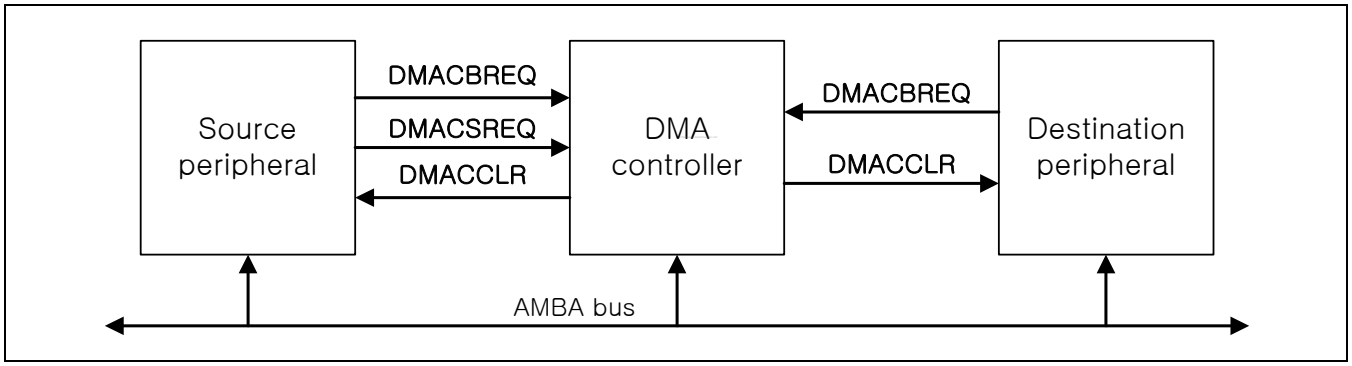


Figure 11-5. Peripheral-to-peripheral transaction comprising bursts and single requests

The next LLI is loaded when all read and write transfers are complete. You can use the **DMACTC** signal to indicate the last data has been transferred to the peripherals.

Transfer Direction	Request Generator	Request Signals Used
Peripheral-to-Memory	Peripheral	**DMACBREQ, DMACSREQ**
Memory-to-Peripheral	Peripheral	**DMACBREQ**
Memory-to-Memory	DMA Controller	**None**
Peripheral-to-Peripheral	Peripheral	**Src : DMACBREQ, DMACSREQ** **Des : DMACBREQ**

Note. DMACSREQ signal can be generated in SW Request only.

11.5.4 SIGNAL TIMING

The timing behavior of the DMA signals is described below:

DMA request signal DMAC{L}(B/S)REQx

Notifies the DMA controller about the peripheral which is ready to proceed with a DMA transfer of the indicated size.
Active HIGH. Sampled by the DMA controller on the positive edge of **HCLK** .The DMA request signals are used in conjunction with the **DMACCLR** signal to perform handshaking.

DMA Acknowledge or Clear DMACCLRx

Indicates to the slave that a DMA transfer is completed.
Active HIGH.

DMA Terminal Count DMACTCx

Indicates to the slave that the end of packet has been reached.
Active HIGH.

NOTE:

If the DMA request source does not use the same clock as the DMA controller, then the request must be synchronized by setting the relevant bit in the DMACSync register.

11.6 FUNCTIONAL TIMING DIAGRAM

A peripheral asserts a DMA request and holds it active. The **DMACCLR** signal is asserted by the DMA controller when the last data item has been transferred. When the peripheral notice that the **DMACCLR** signal has gone active it makes the DMA request signal inactive. The DMAC controller deasserts the **DMACLR** signal when the DMA request signal goes inactive.

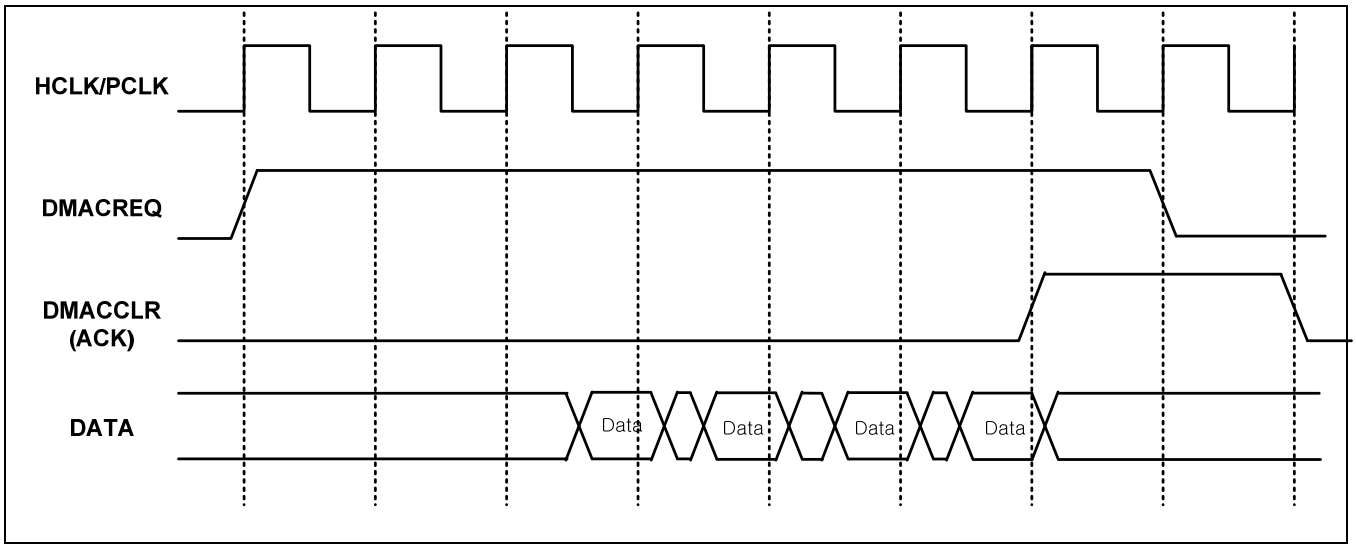


Figure 11-6. DMA interface timing

11.7 PROGRAMMER'S MODEL

11.7.1 PROGRAMMING THE DMA CONTROLLER

All transactions on the AHB Slave programming bus must be 32-bit wide.

11.7.2 ENABLING THE DMA CONTROLLER

To enable the DMA controller set the DMA Enable bit in the DMACConfiguration register.

11.7.3 DISABLING THE DMA CONTROLLER

To disable the DMA controller take the following steps:

1. Read the DMACEnbldChns register and ensure that all the DMA channels have been disabled. If any channels are active refer to *Disabling a DMA channel*.
2. Disable the DMA controller by writing to the DMA Enable bit in the DMACConfiguration register.

11.7.4 ENABLING A DMA CHANNEL

To enable the DMA channel set the Channel Enable bit in the relevant DMA channel configuration register.

NOTE:

The channel must be fully initialized before it is enabled. Additionally, the Enable bit of the DMA controller must be set before any channels are enabled.

11.7.5 DISABLING A DMA CHANNEL

A DMA channel can be disabled in following three ways:

- Write directly to the Channel Enable bit. Any outstanding data in the FIFOs will be lost if this method is used.
- Use the Active and Halt bits in conjunction with the Channel Enable bit.
- Wait until the transfer completes. The channel is disabled automatically.

11.7.5.1 Disabling a DMA channel and losing data in the FIFO:

Clear the relevant Channel Enable bit in the relevant channel configuration register. The current AHB transfers (if one is in progress) complete and the channel is disabled. Any data in the FIFO is lost.

11.7.5.2 Disabling a DMA channel without losing data in the FIFO:

Steps to disable a DMA channel without losing data in the FIFO:

1. Set the Halt bit in the relevant channel configuration register. This ignores any further DMA requests.
2. Poll the Active bit in the relevant channel configuration register until it reaches. This bit indicates whether there is any data in the channel which has to be transferred.
3. Clear the Channel Enable bit in the relevant channel configuration register.

11.7.6 SET UP A NEW DMA TRANSFER

Steps to set up a new DMA transfer:

1. If the channel is not set aside for the DMA transaction:
    a. Read the DMACEnbldChns controller register and find out which channels are inactive.
    b. Select an inactive channel which has the required priority.
2. Program the DMA controller.

11.7.7 HALTING A DMA CHANNEL

Set the Halt bit in the relevant DMA channel configuration register. The current source request is serviced. Any further source DMA requests are ignored until the Halt bit is cleared.

11.7.8 PROGRAMMING A DMA CHANNEL

Steps to program a DMA channel:

1. Decide whether use secure DMAC(SDMAC) or general DMAC(DMAC). In order to use general DMAC, disable secure DMA control register(SDMA_SEL) of system controller. (Reset value is SDMAC)
2. Select a free DMA channel with the priority needed. Where DMA channel 0 has the highest priority and DMA channel 7 the lowest priority.
3. Clear any pending interrupts on the channel to be used by writing to the DMACIntTCClr and DMACIntErrClr registers. The previous channel operation might have left interrupts active.
4. Write the source address into the DMACCxSrcAddr register.
5. Write the destination address into the DMACCxDestAddr register.
6. Write the address of the next LLI into the DMACCxLLI register. If the transfer comprises of a single packet of data then must be written into this register.

Offset	Contents
Next LLI address	Source Address for next transfer
Next LLI address + 0x04	Destination Address for next transfer
Next LLI address + 0x08	Next LLI address for next transfer
Next LLI address + 0x0C	DMACCxControl0 data for next transfer
Next LLI address + 0x10	DMACCxControl1 data for next transfer

7. Write the control information into the DMACCxControl register.
8. Write the channel configuration information into the DMACCxConfiguration register. If the Enable bit is set then the DMA channel is automatically enabled.

11.8 REGISTER DESCRIPTION

There are four DMA Controller named as DMAC0, DMAC1, SDMAC0, and SDMAC1.
The register base addresses of DMAC0, DMAC1, SDMAC0, and SDMAC1 are 0x7500_0000, 0x7510_0000, 0x7DB0_0000, and 0x7DC0_0000 respectively.
Page- access feature for OneNAND Controller is added to channel 3 of DMAC0 and SDAMC0.

11.8.1 DMA REGISTER LOCATION

Table 11-1. DMA register summary

Name	Type	Width	Description	Offset	Reset Value
DMACIntStatus	R	8	This register provides the interrupt status of the DMA controller. A HIGH bit indicates that a specific DMA channel interrupt is active.	0x000	0x00
DMACIntTCStatus	R	8	This register is used to determine whether an interrupt was generated due to the transaction completing (terminal count). A HIGH bit indicates that the transaction is completed.	0x004	0x00
DMACIntTCClear	W	8	When writing to this register, each data bit that is HIGH causes the corresponding bit in the DMACIntTCStatus and DMACRawIntTCStatus registers to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.	0x008	-
DMACIntErrorStatus	R	8	This register is used to determine whether an interrupt was generated due to an error being generated.	0x00C	0x00
DMACIntErrClr	W	8	When writing to this register, each data bit that is HIGH causes the corresponding bit in the DMACIntErrorStatus and DMACRawIntErrorStatus registers to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.	0x010	-
DMACRawIntTCStatus	R	8	This register provides the raw status of DMA terminal count interrupts prior to masking. A HIGH bit indicates that the interrupt request is active prior to masking.	0x014	-
DMACRawIntErrorStatus	R	8	This register provides the raw status of DMA error interrupts prior to masking. A HIGH bit indicates that the interrupt request is active prior to masking.	0x018	-
DMACEnbldChns	R	8	This register shows which DMA channels are enabled. A HIGH bit indicates that a DMA channel is enabled.	0x01C	0x00

Table 11-1 DMA register summary (continued)

Name	Type	Width	Description	Offset	Reset Value
DMACSoftBReq	R/W	16	This register allows DMA burst requests to be generated by software.	0x020	0x0000
DMACSoftSReq	R/W	16	This register allows DMA single requests to be generated by software.	0x024	0x0000
Reserved	-	16	-	0x028	0x0000
Reserved	-	16	-	0x02C	0x0000
DMACConfiguration	R/W	3	This register is used to configure the DMA controller.	0x030	0x000
DMACSync	R/W	16	This register enables or disables synchronization logic for the DMA request signals.	0x034	0x0000
DMACC0SrcAddr	R/W	32	DMA channel 0 source address.	0x100	0x00000000
DMACC0DestAddr	R/W	32	DMA channel 0 destination address.	0x104	0x00000000
DMACC0LLI	R/W	32	DMA channel 0 linked list address.	0x108	0x00000000
DMACC0Control0	R/W	32	DMA channel 0 control0.	0x10C	0x00000000
DMACC0Control1	R/W	32	DMA channel 0 control1.	0x110	0x00000000
DMACC0Configuration	R/W	19	DMA channel 0 configuration register.	0x114	0x00000
DMACC0ConfigurationExp	R/W	3	DMA channel 0 configuration expension reg.	0x118	0x0
DMACC1SrcAddr	R/W	32	DMA channel 1 source address.	0x120	0x00000000
DMACC1DestAddr	R/W	32	DMA channel 1 destination address.	0x124	0x00000000
DMACC1LLI	R/W	32	DMA channel 1 linked list address.	0x128	0x00000000
DMACC1Control0	R/W	32	DMA channel 1 control0.	0x12C	0x00000000
DMACC1Control1	R/W	32	DMA channel 1 control1.	0x130	0x00000000
DMACC1Configuration	R/W	19	DMA channel 1 configuration register.	0x134	0x00000
DMACC1ConfigurationExp	R/W	3	DMA channel 1 configuration expension reg.	0x138	0x0
DMACC2SrcAddr	R/W	32	DMA channel 2 source address.	0x140	0x00000000
DMACC2DestAddr	R/W	32	DMA channel 2 destination address.	0x144	0x00000000
DMACC2LLI	R/W	32	DMA channel 2 linked list address.	0x148	0x00000000
DMACC2Control0	R/W	32	DMA channel 2 control.	0x14C	0x00000000
DMACC2Control1	R/W	32	DMA channel 2 control.	0x150	0x00000000
DMACC2Configuration	R/W	19	DMA channel 2 configuration register.	0x154	0x00000
DMACC2ConfigurationExp	R/W	3	DMA channel 2 configuration expension reg.	0x158	0x0
DMACC3SrcAddr	R/W	32	DMA channel 3 source address.	0x160	0x00000000
DMACC3DestAddr	R/W	32	DMA channel 3 destination address.	0x164	0x00000000
DMACC3LLI	R/W	32	DMA channel 3 linked list address.	0x168	0x00000000

Table 11-1 DMA register summary (continued)

Name	Type	Width	Description	Offset	Reset Value
DMACC3Control0	R/W	32	DMA channel 3 control0.	0x16C	0x00000000
DMACC3Control1	R/W	32	DMA channel 3 control1.	0x170	0x00000000
DMACC3Configuration	R/W	19	DMA channel 3 configuration register.	0x174	0x00000
DMACC3ConfigurationExp	R/W	3	DMA channel 3 configuration expension reg.	0x178	0x0
DMACC4SrcAddr	R/W	32	DMA channel 4 source address.	0x180	0x00000000
DMACC4DestAddr	R/W	32	DMA channel 4 destination address.	0x184	0x00000000
DMACC4LLI	R/W	32	DMA channel 4 linked list address.	0x188	0x00000000
DMACC4Control0	R/W	32	DMA channel 4 control0.	0x18C	0x00000000
DMACC4Control1	R/W	32	DMA channel 4 control1.	0x190	0x00000000
DMACC4Configuration	R/W	19	DMA channel 4 configuration register.	0x194	0x00000
DMACC4ConfigurationExp	R/W	3	DMA channel 4 configuration expension reg.	0x198	0x0
DMACC5SrcAddr	R/W	32	DMA channel 5 source address.	0x1A0	0x00000000
DMACC5DestAddr	R/W	32	DMA channel 5 destination address.	0x1A4	0x00000000
DMACC5LLI	R/W	32	DMA channel 5 linked list address.	0x1A8	0x00000000
DMACC5Control0	R/W	32	DMA channel 5 control0.	0x1AC	0x00000000
DMACC5Control1	R/W	32	DMA channel 5 control1.	0x1B0	0x00000000
DMACC5Configuration	R/W	19	DMA channel 5 configuration register.	0x1B4	0x00000
DMACC5ConfigurationExp	R/W	3	DMA channel 5 configuration expension reg.	0x1B8	0x0
DMACC6SrcAddr	R/W	32	DMA channel 6 source address.	0x1C0	0x00000000
DMACC6DestAddr	R/W	32	DMA channel 6 destination address.	0x1C4	0x00000000
DMACC6LLI	R/W	32	DMA channel 6 linked list address.	0x1C8	0x00000000
DMACC6Control0	R/W	32	DMA channel 6 control0.	0x1CC	0x00000000
DMACC6Control1	R/W	32	DMA channel 6 control1.	0x1D0	0x00000000
DMACC6Configuration	R/W	19	DMA channel 6 configuration register.	0x1D4	0x00000
DMACC6ConfigurationExp	R/W	3	DMA channel 6 configuration expension reg.	0x1D8	0x0
DMACC7SrcAddr	R/W	32	DMA channel 7 source address.	0x1E0	0x00000000
DMACC7DestAddr	R/W	32	DMA channel 7 destination address.	0x1E4	0x00000000
DMACC7LLI	R/W	32	DMA channel 7 linked list address.	0x1E8	0x00000000
DMACC7Control0	R/W	32	DMA channel 7 control0.	0x1EC	0x00000000
DMACC7Control1	R/W	32	DMA channel 7 control1.	0x1F0	0x00000000
DMACC7Configuration	R/W	19	DMA channel 7 configuration register.	0x1F4	0x00000
DMACC7ConfigurationExp	R/W	3	DMA channel 7 configuration expension reg.	0x1F8	0x0

11.8.2 INTERRUPT STATUS REGISTER, DMACINTSTATUS

The DMACIntStatus register is read-only and indicates the status of the interrupts after masking. A HIGH bit indicates that a specific DMA channel interrupt request is active. The request can be generated from either the error or terminal count interrupt requests.

Table 11-2 shows the bit assignment of the DMACIntStatus register.

Table 11-2. Bit Assignment of DMACIntStatus register

DMACIntStatus	Bits	Type	Function
IntStatus	[7:0]	R	Status of the DMA interrupts after masking

11.8.3 INTERRUPT TERMINAL COUNT STATUS REGISTER, DMACINTTCSTATUS

The DMACIntTCStatus register is read-only and indicates the status of the terminal count after masking. This register must be used in conjunction with the DMACIntStatus register if the combined interrupt request, DMACINTCOMBINE, is used to request interrupts.

If the DMACINTTC interrupt request is used then you only have to read the DMACIntTCStatus register to ascertain the source of the interrupt request.

Table 11-3 shows the bit assignment of the DMACIntTCStatus register.

Table 11-3. Bit Assignment of DMACIntTCStatus register

DMACIntTCStatus	Bits	Type	Function
IntTCStatus	[7:0]	R	Interrupt terminal count request status

11.8.4 INTERRUPT TERMINAL COUNT CLEAR REGISTER, DMACINTTCCLEAR

The DMACIntTCClear register is write-only and is used to clear a terminal count interrupt request.

When writing to this register, each data bit that is set HIGH causes the corresponding bit in the status register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.

Table 11-4 shows the bit assignment of the DMACIntTCClear register.

Table 11-4. Bit Assignment of DMACIntTCClear register

DMACIntTCClear	Bits	Type	Function
IntTCClear	[7:0]	W	Terminal count request clear

11.8.5 INTERRUPT ERROR STATUS REGISTER, DMACINTERRORSTATUS

The DMACIntErrorStatus register is read-only register and indicates the status of the error request after masking.

This register must be used in conjunction with the DMACIntStatus register if the combined interrupt request, DMACINTCOMBINE, is used to request interrupts.

If the DMACINTERROR interrupt request is used only the DMACIntErrorStatus register needs to be read.

Table 11-5 shows the bit assignment of the DMACIntErrorStatus register.

Table 11-5. Bit Assignment of DMACIntErrorStatus register

DMACIntErrorStatus	Bits	Type	Function
IntErrorStatus	[7:0]	R	Interrupt error status

11.8.6 INTERRUPT ERROR CLEAR REGISTER, DMACINTERRCLR

The DMACIntErrClr register is a write-only register and is used to clear the error interrupt requests. When writing to this register, each data bit that is HIGH causes the corresponding bit in the status register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.

Table 11-6 shows the bit assignment of the DMACIntErrClr register.

Table 11-6. Bit Assignment of DMACIntErrClr register

DMACIntErrClr	Bits	Type	Function
IntErrClr	[7:0]	W	Interrupt error clear

11.8.7 RAW INTERRUPT TERMINAL COUNTER STATUS REGISTER, DMACRAWINTTCSTATUS

The DMACRawIntTCStatus register is read-only. It indicates which DMA channels are requesting a transfer complete (terminal count interrupt) prior to masking. A HIGH bit indicates that the terminal count interrupt request is active prior to masking.

Table 11-7 shows the bit assignment of the DMACRawIntTCStatus register.

Table 11-7. Bit Assignment of DMACRawIntTCStatus register

DMACRawIntTCStatus	Bits	Type	Function
RawIntTCStatus	[7:0]	R	Status of the terminal count interrupt prior to masking

11.8.8 RAW ERROR INTERRUPT STATUS REGISTER, DMACRAWINTERRORSTATUS

The DMACRawIntErrorStatus register is read-only. It indicates which DMA channels are requesting an error interrupt prior to masking. A HIGH bit indicates that the error interrupt request is active prior to masking.

Table 11-8 shows the bit assignment of register of the DMACRawIntErrorStatus register.

Table 11-8. Bit Assignment of DMACRawIntErrorStatus register

DMACRawIntErrorStatus	Bits	Type	Function
RawIntErrorStatus	[7:0]	R	Status of the error interrupt prior to masking

11.8.9 ENABLE CHANNEL REGISTER, DMACENBLDCHNS

The DMACEnbldChns register is read-only and indicates which DMA channels are enabled, as indicated by the Enable bit in the DMACCxConfiguration register. A HIGH bit indicates that a DMA channel is enabled. A bit is cleared on completion of the DMA transfer.

Table 11-9 shows the bit assignment of the DMACEnbldChns register.

Table 11-9. Bit Assignment of DMACEnbldChns register

DMACEnbldChns	Bits	Type	Function
EnabledChannels	[7:0]	R	Channel enable status

11.8.10 SOFTWARE BURST REQUEST REGISTER, DMACSOFTBREQ

The DMACSoftBReq register is read/write and it allows DMA burst requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction is complete. Writing 0 to this register has no effect.

Reading the register indicates which sources are requesting DMA burst transfers. A request can be generated from either a peripheral or the software request register.

Table 11-10 shows the bit assignment of the DMACSoftBReq register.

Table 11-10. Bit Assignment of DMACSoftBReq register

DMACSoftBReq	Bits	Type	Function
SoftBReq	[15:0]	R/W	Software burst request

NOTE: It is recommended that software and hardware peripheral requests are not used at the same time.

11.8.11 SOFTWARE SINGLE REQUEST REGISTER, DMACSOFTSREQ

The DMACSoftSReq read/write register allows DMA single requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction is complete. Writing to this register has no effect.

Reading the register indicates which sources are requesting single DMA transfers. A request can be generated from either a peripheral or the software request register.

Table 11-11 shows the bit assignment of the DMACSoftSReq register.

Table 11-11. Bit Assignment of DMACSoftSReq register

DMACSoftSReq	Bits	Type	Function
SoftSReq	[15:0]	R/W	Software single request

NOTE: It is recommended that software and hardware peripheral requests are not used at the same time.

11.8.14 CONFIGURATION REGISTER, DMACCONFIGURATION

The DMACConfiguration read/write register is used to configure the operation of the DMA controller. The AHB master interfaces are set to little-endian mode on reset.

Table 11-12 shows the bit assignment of the DMACConfiguration register.

Table 11-12. Bit Assignment of DMACConfiguration register

DMACConfiguration	Bits	Type	Function
Reserved	[2:1]	R/W	Should be 0
E	[0]	R/W	DMA controller enable: 0 =disabled 1 =enabled. This bit is reset to 0.Disabling the DMA controller reduces power consumption.

11.8.15 SYNCHRONIZATION REGISTER, DMACSYNC

The DMACSync read/write register is used to enable or disable synchronization logic for the DMA request signals.
A bit set to 0 enables the synchronization logic for a particular group of DMA requests.
A bit set to 1 disables the synchronization logic for a particular group of DMA requests.
This register is reset to 0, synchronization logic enabled.

Table 11-13 shows the bit assignment of the DMACSync register.

Table 11-13. Bit Assignment of DMACSync register

DMACSync	Bits	Type	Function
DMACSync	[15:0]	R/W	DMA synchronization logic for DMA request signals enabled or disabled. A LOW bit indicates that the synchronization logic for the **DMACBREQ[15:0]**, **DMACSREQ[15:0]**, **DMACLBREQ[15:0]**, and **DMACLSREQ[15:0]** request signals is enabled. A HIGH bit indicates that the synchronization logic is disabled.

NOTE: Synchronization logic must be used when the peripheral generating the DMA request runs on a different clock to the DMA controller. For peripherals running on the same clock as the DMA controller disabling the synchronization logic improves the DMA request response time. If necessary, the DMA response signals, **DMACCLR** and **DMACTC**, must be synchronized in the peripheral.

11.8.16 CHANNEL SOURCE ADDRESS REGISTER, DMACCXSRCADDR

The eight read/write DMACCxSrcAddr registers contain the current source address (byte-aligned) of the data to be transferred.

Each register is programmed directly by software before the appropriate channel is enabled. When the DMA channel is enabled this register is updated:

- As the source address is incremented
- By following the linked list when a complete packet of data has been transferred.

Reading the register when the channel is active does not provide useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when the channel has stopped, in which case it shows the source address of the last item read.

NOTE:

The source and destination addresses must be aligned to the source and destination widths.

Table 11-14 shows the bit assignment of the DMACCxSrcAddr registers.

Table 11-14. Bit Assignment of DMACCxSrcAddr register

DMACCxSrcAddr	Bits	Type	Function
SrcAddr	[31:0]	R/W	DMA Source address

11.8.17 CHANNEL DESTINATION ADDRESS REGISTER, DMACCXDESTADDR

The eight read/write DMACCxDestAddr registers contain the current destination address (byte-aligned) of the data to be transferred.

Each register is programmed directly by software before the channel is enabled. When the DMA channel is enabled, the register is updated as the destination address is incremented and by following the linked list when a complete packet of data has been transferred.

Reading the register when the channel is active does not provide useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when a channel has stopped, in which case it shows the destination address of the last item read.

Table 11-15 shows the bit assignment of a DMACCxDestAddr register.

Table 11-15. Bit Assignment of DMACCxDestAddr register

DMACCxDestAddr	Bits	Type	Function
DestAddr	[31:0]	R/W	DMA destination address

11.8.18 CHANNEL LINKED LIST ITEM REGISTER, DMACCXLLI

The eight read/write DMACCxLLI registers contain a word aligned address of the next *Linked List Item* (LLI). If the LLI is, then the current LLI is the last in the chain, and the DMA channel is disabled once all DMA transfers associated with it are completed.

NOTE:

Programming this register when the DMA channel is enabled has unpredictable side effects.
To make loading LLIs more efficient for some systems, the LLI data structures can be made 4-word aligned.

Table 11-16 shows the bit assignment of a DMACCxLLI register.

Table 11-16. Bit Assignment of DMACCxLLI register

DMACCxLLI	Bits	Type	Function
LLI	[31:2]	R/W	Linked list item. Bits [31:2] of the address for the next LLI. Address bits [1:0] are.
R	[1]	R/W	Reserved, and must be written as 0,masked on read.
LM	[0]	R/W	AHB master select for loading the next LLI: LM = 0 = AHB master 1 LM = 1 = AHB master 2.

11.8.19 CHANNEL CONTROL REGISTER, DMACCXCONTROL0

The eight read/write DMACCxControl0 registers contain DMA channel control information such as the burst size, and transfer width.
Each register is programmed directly by software before the DMA channel is enabled. When the channel is enabled the register is updated by following the linked list when a complete packet of data has been transferred. Reading the register whilst the channel is active does not give useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when a channel has stopped.

Table 11-17 shows the bit assignment of a DMACCxControl0 register.

Table 11-17. Bit Assignment of DMACCxControl0 register

DMACCxControl	Bits	Type	Function
I	[31]	R/W	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
Prot	[30:28]	R/W	Protection. Refer to Table 11-20
DI	[27]	R/W	Destination increment. When set the destination address is incremented after each transfer.
SI	[26]	R/W	Source increment. When set the source address is incremented after each transfer.
D	[25]	R/W	Destination AHB master select: 0 = AHB master 1 (AXI_SYSTEM) selected for the destination transfer. 1 = AHB master 2 (AXI_PERI) selected for the destination transfer.

Table 11-17. Bit Assignment of DMACCxControl register (continued)

DMACCxControl	Bits	Type	Function
S	[24]	R/W	Source AHB master select: 0 = AHB master 1 (AXI_SYSTEM) selected for the source transfer 1 = AHB master 2 (AXI_PERI) selected for the source transfer.
Dwidth	[23:21]	R/W	Destination transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data as required.
SWidth	[20:18]	R/W	Source transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data as required.
DBSize	[17:15]	R/W	Destination burst size. Indicates the number of transfers, which make up a destination burst transfer request. This value must be set to the burst size of the destination peripheral, or if the destination is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the **DMACxBREQ** signal goes active in the destination peripheral. The burst size is not related to the AHB **HBURST** signal.
SBSize	[14:12]	R/W	Source burst size. Indicates the number of transfers, which make up a source burst. This value must be set to the burst size of the source peripheral, or if the source is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the **DMACxBREQ** signal goes active in the source peripheral. The burst size is not related to the AHB **HBURST** signal.
Reserved	[11:0]	R	Reserved

Table 11-18. Source or destination burst size

Bit value of SBSize or DBSize	Source or destination burst transfer request size
0b000	1
0b001	4
0b010	8
0b011	16
0b100	Reserved
0b101	Reserved
0b110	Reserved
0b111	Reserved

Table 11-19. Source or destination transfer width

Bit value of SWidth or DWidth	Source or destination width
0b000	Byte (8-bit)
0b001	Half word (16-bit)
0b010	Word (32-bit)
0b011	Reserved
0b100	Reserved
0b101	Reserved
0b110	Reserved
0b111	Reserved

Note. DMAC has internal 4 word FIFO per each channel. So, burst size and transfer width is limited by the FIFO size. For example, if data width is word, available burst size is under 4. If data width is byte, available burst size is under 16.

AHB access information is provided to the source and destination peripherals when a transfer occurs. The transfer information is provided by programming the DMA channel (the Prot bit of the DMACCxControl register, and the Lock bit of the DMACCxConfiguration register). These bits are programmed by software and peripherals can use this information if necessary. Three bits of information are provided, and Table 11-20 shows the purpose of the three protection bits.

Table 11-20. Protection bits

Bits	Description	Purpose
0	Privileged or User	Indicates that the access is in User, or privileged mode: 0 = User mode 1 = privileged mode. This bit controls the AHB **HPROT[1]** signal.
1	Bufferable or not bufferable	Indicates that the access is bufferable, or not bufferable: 0 = not bufferable 1 = bufferable. This bit indicates that the access is bufferable. This bit can, for example, be used to indicate to an AMBA bridge that the read can complete in zero wait states on the source bus without waiting for it to arbitrate for the destination bus and for the slave to accept the data. This bit controls the AHB **HPROT[2]** signal.
2	Cacheable or not cacheable	Indicates that the access is cacheable or not cacheable: 0 = not cacheable 1 = cacheable. This indicates that the access is cacheable. This bit can, for example, be used to indicate to an AMBA bridge that when it saw the first read of a burst of eight it can transfer the whole burst of eight reads on the destination bus, rather than pass the transactions through one at a time. This bit controls the AHB **HPROT[3]** signal.

11.8.20 CHANNEL CONTROL REGISTER, DMACCXCONTROL1

The eight read/write DMACCxControl1 registers contain DMA channel control information such as the transfer size.

Each register is programmed directly by software before the DMA channel is enabled. When the channel is enabled the register is updated by following the linked list when a complete packet of data has been transferred.

Reading the register whilst the channel is active does not give useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when a channel has stopped.
Table 11-21 shows the bit assignment of a DMACCxControl1 register.

Table 11-21. Bit Assignment of DMACCxControl1 register

DMACCxControl	Bits	Type	Function
TransferSize	[24:0]	R/W	Transfer size. A write to this field indicates the size of transfer. A read from this field indicates the number of transfers completed on the destination bus. Reading the register when the channel is active does not give useful information, as by the time that the software has processed the value read, the channel might have progressed. It is intended to be used only when a channel is enabled and then disabled.

11.8.21 CHANNEL CONFIGURATION REGISTER, DMACCXCONFIGURATION

The eight DMACCxConfiguration registers are read/write and are used to configure the DMA channel. The registers are not updated when a new LLI is requested.

Table 11-22 shows the bit assignment of a DMACCxConfiguration register.

Table 11-22. Bit Assignment of DMACCxConfiguration register

DMACCxConfiguration	Bits	Type	Function
H	[18]	R/W	Halt: 0 = allow DMA requests 1 = ignore further source DMA requests. The contents of the channels FIFO are drained. This value can be used with the Active and Channel Enable bits to cleanly disable a DMA channel.
A	[17]	R	Active: 0 = there is no data in the FIFO of the channel 1 = the FIFO of the channel has data. This value can be used with the Halt and Channel Enable bits to cleanly disable a DMA channel.
L	[16]	R/W	Lock. When set this bit enables locked transfers.
ITC	[15]	R/W	Terminal count interrupt mask. When cleared this bit masks out the terminal count interrupt of the relevant channel.
IE	[14]	R/W	Interrupt error mask. When cleared this bit masks out the error interrupt of the relevant channel.
FlowCntrl	[13:11]	R/W	Flow control and transfer type. This value is used to indicate the flow controller and transfer type. The supported flow controller is only the DMA controller. The transfer type can be memory-to-memory, memory-to-peripheral, peripheral-to-memory, or peripheral-to-peripheral.
OneNandModeDst	[10]	R/W	This bit is used to support page-write features for OneNAND Controller. If this bit is set to 1 and the destination address points the address field of OneNAND Controller, destination address increment setting can support 01 command of OneNAND Controller. To be sure, when this bit is set to one, D should be “AHB master 1”, DI should be “increment”, DWidth should be “word”, and DBSize should be the multiple of four. Note that using DMAC0 does not guarantee correct operation.
DestPeripheral	[9:6]	R/W	Destination peripheral. This value selects the DMA destination request peripheral. This field is ignored if the destination of the transfer is to memory.

DMACCxConfiguration	Bits	Type	Function
OneNandModeSrc	[5]	R/W	This bit is used to support page-read features for OneNAND Controller. If this bit is set to one and the source address points the address field of OneNAND Controller, source address increment setting can support 01 command of OneNAND Controller. To be sure, when this bit is set to one, S should be “AHB master 1”, SI should be “increment”, SWidth should be “word”, and SBSize should be the multiple of four. Note that using DMAC0 does not guarantee correct operation.
SrcPeripheral	[4:1]	R/W	Source peripheral. This value selects the DMA source request peripheral. This field is ignored if the source of the transfer is from memory.
E	[0]	R/W	Channel enable. Reading this bit indicates whether a channel is currently enabled or disabled: 0 = channel disabled 1 = channel enabled. The Channel Enable bit status can also be found by reading the DMACEnbldChns register. A channel is enabled by setting this bit. A channel can be disabled by clearing the Enable bit. This causes the current AHB transfer (if one is in progress) to complete and the channel is then disabled. Any data in the channels FIFO is lost. Restarting the channel by simply setting the Channel Enable bit has unpredictable effects and the channel must be fully re-initialized. The channel is also disabled, and Channel Enable bit cleared, when the last LLI is reached or if a channel error is encountered. If a channel has to be disabled without losing data in a channels FIFO the Halt bit must be set so that further DMA requests are ignored. The Active bit must then be polled until it reaches 0, indicating that there is no data left in the channels FIFO. Finally the Channel Enable bit can be cleared.

Table 11-23 describes the bit values of the three flow control and transfer type bits.

Table 11-23. Flow control and transfer type bits

Bits value	Transfer type
000	Memory to memory
001	Memory to peripheral
010	Peripheral to memory
011	Source peripheral to destination peripheral
100~111	Reserved

11.8.22 CHANNEL CONFIGURATION EXPANSION REGISTER, DMACCXCONFIGURATIONEXP

The eight DMACCxConfigurationExp registers are read/write and are used to configure the DMA channel additionally.

Table 11-24 shows the bit assignment of a DMACCxConfigurationExp register.

Table 11-24. Bit Assignment of DMACCxConfigurationExp register

DMACCxConfigurationExp	Bits	Type	Function
PeriReqSel	[2:1]	R/W	Peripheral DMA Request for Mem-to-Mem Access Selection : 00 = MODEM_IF TX 0 Request 01 = MODEM_IF TX 1 Request 10 = MODEM_IF RX 0 Request 11 = MODEM_IF RX 1 Request
EnPeriReq	[0]	R/W	Enable for Peripheral DMA Request: 0 = Disable 1 = Enable This value can be used in Peripheral for Mem-to-Mem access not Peri-to-Mem access.

NOTES

12 VECTORED INTERRUPT CONTROLLERS

This chapter describes the functions and usage of Vectored Interrupt Controller in S3C6410X RISC microprocessor.

12.1 OVERVIEW

The interrupt controller in the S3C6410X is composed of 2 VIC's (Vectored Interrupt Controller, ARM PrimeCell PL192) and 2 TZIC's (TrustZone Interrupt Controller, SP890).

Two TZIC's and VIC's are daisy-chained to support up to 64 interrupt sources.

12.2 FEATURES

The Vectored Interrupt Controller features in S3C6410 incudes the following:

- Support for 32 vectored IRQ interrupts per VIC
- Fixed hardware interrupt priority levels
- Programmable interrupt priority levels
- Hardware interrupt priority level masking
- Programmable interrupt priority level masking
- IRQ and FIQ generation
- Software interrupt generation
- Raw interrupt status
- Interrupt request status
- Privileged mode support for restricted access

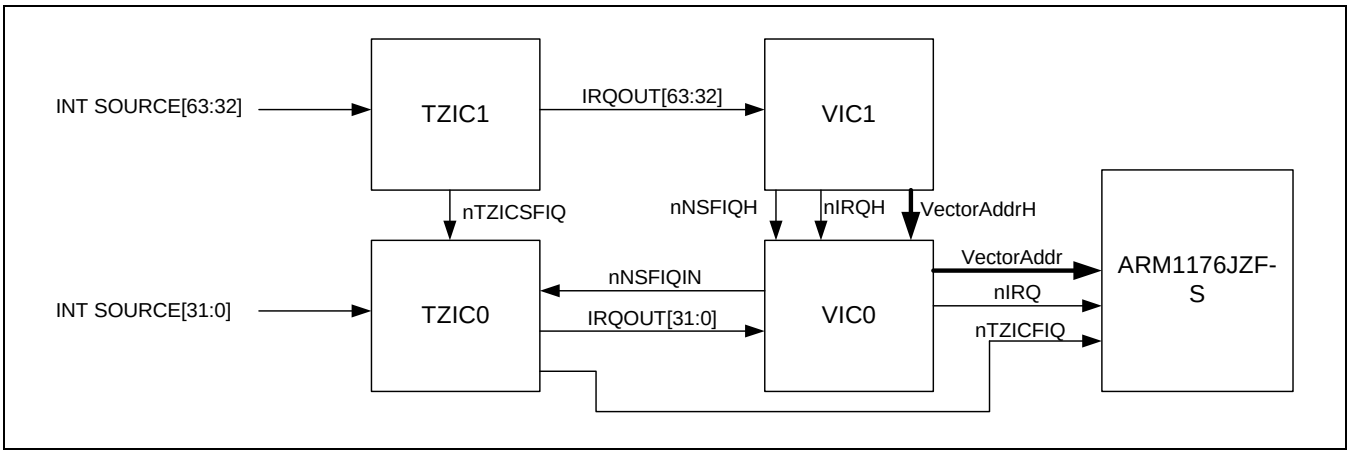


Figure 12-1. Interrupt Controller in S3C6410X

12.3 INTERRUPT SOURCES

The S3C6410X supports 64 interrupt sources as shown in the table below.

The S3C6410X can not support performance monitoring interrupt of the ARM1176JZF-S.

Int. No.	Sources	Description	Group
63	INT_ADC	ADC EOC interrupt	VIC1
62	INT_PENDNUP	ADC Pen down/up interrupt	VIC1
61	INT_SEC	Security interrupt	VIC1
60	INT_RTC_ALARM	RTC alarm interrupt	VIC1
59	INT_IrDA	IrDA interrupt	VIC1
58	INT_OTG	USB OTG interrupt	VIC1
57	INT_HSMMC1	HSMMC1 interrupt	VIC1
56	INT_HSMMC0	HSMMC0 interrupt	VIC1
55	INT_HOSTIF	Host Interface interrupt	VIC1
54	INT_MSM	MSM modem I/F interrupt	VIC1
53	INT_EINT4	External interrupt Group 1 ~ Group 9	VIC1
52	INT_HSIrx	HSI Rx interrupt	VIC1
51	INT_HSItx	HSI Tx interrupt	VIC1
50	INT_I2C0	I2C 0 interrupt	VIC1
49	INT_SPI1/INT_HSMMC2	SPI1 interrupt or HSMMC2 interrupt	VIC1
48	INT_SPI0	SPI0 interrupt	VIC1
47	INT_UHOST	USB Host interrupt	VIC1
46	INT_CFC	CFCON interrupt	VIC1
45	INT_NFC	NFCON interrupt	VIC1
44	INT_ONENAND1	OneNAND interrupt from bank 1	VIC1
43	INT_ONENAND0	OneNAND interrupt from bank 0	VIC1
42	INT_DMA1	DMA1 interrupt	VIC1

Int. No.	Sources	Description	Group
41	INT_DMA0	DMA0 interrupt	VIC1
40	INT_UART3	UART3 interrupt	VIC1
39	INT_UART2	UART2 interrupt	VIC1
38	INT_UART1	UART1 interrupt	VIC1
37	INT_UART0	UART0 interrupt	VIC1
36	INT_AC97	AC97 interrupt	VIC1
35	INT_PCM1	PCM1 interrupt	VIC1
34	INT_PCM0	PCM0 interrupt	VIC1
33	INT_EINT3	External interrupt 20 ~ 27	VIC1
32	INT_EINT2	External interrupt 12 ~ 19	VIC1
31	INT_LCD[2]	LCD interrupt. System I/F done	VIC0
30	INT_LCD[1]	LCD interrupt. VSYNC interrupt	VIC0
29	INT_LCD[0]	LCD interrupt. FIFO underrun	VIC0
28	INT_TIMER4	Timer 4 interrupt	VIC0
27	INT_TIMER3	Timer 3 interrupt	VIC0
26	INT_WDT	Watchdog timer interrupt	VIC0
25	INT_TIMER2	Timer 2 interrupt	VIC0
24	INT_TIMER1	Timer 1 interrupt	VIC0
23	INT_TIMER0	Timer 0 interrupt	VIC0
22	INT_KEYPAD	Keypad interrupt	VIC0
21	INT_ARM_DMAS	ARM DMAS interrupt	VIC0
20	INT_ARM_DMA	ARM DMA interrupt	VIC0
19	INT_ARM_DMAERR	ARM DMA Error interrupt	VIC0
18	INT_SDMA1	Secure DMA1 interrupt	VIC0
17	INT_SDMA0	Secure DMA0 interrupt	VIC0
16	INT_MFC	MFC interrupt	VIC0
15	INT_JPEG	JPEG interrupt	VIC0
14	INT_BATF	Battery fault interrupt	VIC0
13	INT_SCALER	TV Scaler interrupt	VIC0
12	INT_TVENC	TV Encoder interrupt	VIC0
11	INT_2D	2D interrupt	VIC0
10	INT_ROTATOR	Rotator interrupt	VIC0
9	INT_POST0	Post processor interrupt	VIC0
8	INT_3D	3D Graphic Controller interrupt	VIC0
7	Reserved	Reserved	VIC0

Int. No.	Sources	Description	Group
6	INT_I2S0 | INT_I2S1 | INT_I2SV40	I2S 0 interrupt or I2S 1 interrupt or I2S V40 interrupt	VIC0
5	INT_I2C1	I2C 1 interrupt	VIC0
4	INT_CAMIF_P	Camera interface interrupt	VIC0
3	INT_CAMIF_C	Camera interface interrupt	VIC0
2	INT_RTC_TIC	RTC TIC interrupt	VIC0
1	INT_EINT1	External interrupt 4 ~ 11	VIC0
0	INT_EINT0	External interrupt 0 ~ 3	VIC0

Note: 1. 6410X PoP A Type doesn't support "INT_NFC"

2. 6410X PoP D Type doesn't support "INT_ONENAND0" & "INT_ONENAND1"

12.4 FUNCTION

12.4.1 BLOCK DIAGRAM

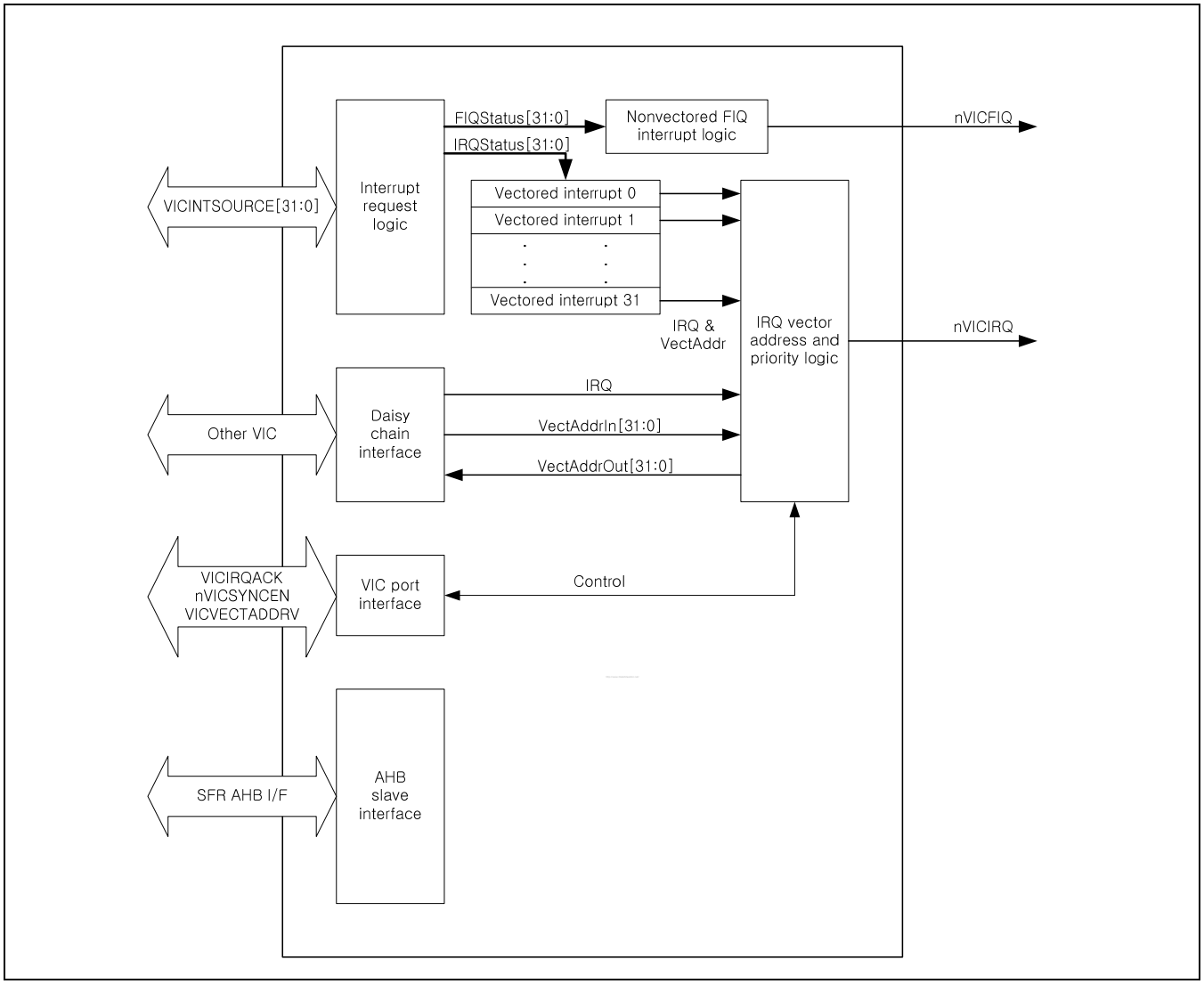


Figure 12-2. Vectored Interrupt Controller block diagram

12.5 SUMMARY OF VIC REGISTERS

- Base address of VIC0 is 0x7120_0000
- Base address of VIC1 is 0x7130_0000
- Address of control register = base address + offset

Register	Offset	Type	Description	Reset Value
VICxIRQSTATUS	0x000	R	IRQ Status Register	0x00000000
VICxFIQSTATUS	0x004	R	FIQ Status Register	0x00000000
VICxRAWINTR	0x008	R	Raw Interrupt Status Register	0x00000000
VICxINTSELECT	0x00C	RW	Interrupt Select Register	0x00000000
VICxINTENABLE	0x010	RW	Interrupt Enable Register	0x00000000
VICxINTENCLEAR	0x014	W	Interrupt Enable Clear Register	-
VICxSOFTINT	0x018	RW	Software Interrupt Register	0x00000000
VICxSOFTINTCLEAR	0x01C	W	Software Interrupt Clear Register	-
VICxPROTECTION	0x020	RW	Protection Enable Register	0x0
VICxSWPRIORITYMASK	0x024	RW	Software Priority Mask Register	0xFFFF
VICxPRIORITYDAISY	0x028	RW	Vector Priority Register for Daisy Chain	0xF
VICxVECTADDR0	0x100	RW	Vector Address 0 Register	0x00000000
VICxVECTADDR1	0x104	RW	Vector Address 1 Register	0x00000000
VICxVECTADDR2	0x108	RW	Vector Address 2 Register	0x00000000
VICxVECTADDR3	0x10C	RW	Vector Address 3 Register	0x00000000
VICxVECTADDR4	0x110	RW	Vector Address 4 Register	0x00000000
VICxVECTADDR5	0x114	RW	Vector Address 5 Register	0x00000000
VICxVECTADDR6	0x118	RW	Vector Address 6 Register	0x00000000
VICxVECTADDR7	0x11C	RW	Vector Address 7 Register	0x00000000
VICxVECTADDR8	0x120	RW	Vector Address 8 Register	0x00000000
VICxVECTADDR9	0x124	RW	Vector Address 9 Register	0x00000000
VICxVECTADDR10	0x128	RW	Vector Address 10 Register	0x00000000
VICxVECTADDR11	0x12C	RW	Vector Address 11 Register	0x00000000
VICxVECTADDR12	0x130	RW	Vector Address 12 Register	0x00000000
VICxVECTADDR13	0x134	RW	Vector Address 13 Register	0x00000000
VICxVECTADDR14	0x138	RW	Vector Address 14 Register	0x00000000
VICxVECTADDR15	0x13C	RW	Vector Address 15 Register	0x00000000
VICxVECTADDR16	0x140	RW	Vector Address 16 Register	0x00000000
VICxVECTADDR17	0x144	RW	Vector Address 17 Register	0x00000000
VICxVECTADDR18	0x148	RW	Vector Address 18 Register	0x00000000

Register	Offset	Type	Description	Reset Value
VICxVECTADDR19	0x14C	RW	Vector Address 19 Register	0x00000000
VICxVECTADDR20	0x150	RW	Vector Address 20 Register	0x00000000
VICxVECTADDR21	0x154	RW	Vector Address 21 Register	0x00000000
VICxVECTADDR22	0x158	RW	Vector Address 22 Register	0x00000000
VICxVECTADDR23	0x15C	RW	Vector Address 23 Register	0x00000000
VICxVECTADDR24	0x160	RW	Vector Address 24 Register	0x00000000
VICxVECTADDR25	0x164	RW	Vector Address 25 Register	0x00000000
VICxVECTADDR26	0x168	RW	Vector Address 26 Register	0x00000000
VICxVECTADDR27	0x16C	RW	Vector Address 27 Register	0x00000000
VICxVECTADDR28	0x170	RW	Vector Address 28 Register	0x00000000
VICxVECTADDR29	0x174	RW	Vector Address 29 Register	0x00000000
VICxVECTADDR30	0x178	RW	Vector Address 30 Register	0x00000000
VICxVECTADDR31	0x17C	RW	Vector Address 31 Register	0x00000000
VICxVECPRIORITY0	0x200	RW	Vector Priority 0 Register	0xF
VICxVECTPRIORITY1	0x204	RW	Vector Priority 1 Register	0xF
VICxVECTPRIORITY2	0x208	RW	Vector Priority 2 Register	0xF
VICxVECTPRIORITY3	0x20C	RW	Vector Priority 3 Register	0xF
VICxVECTPRIORITY4	0x210	RW	Vector Priority 4 Register	0xF
VICxVECTPRIORITY5	0x214	RW	Vector Priority 5 Register	0xF
VICxVECTPRIORITY6	0x218	RW	Vector Priority 6 Register	0xF
VICxVECTPRIORITY7	0x21C	RW	Vector Priority 7 Register	0xF
VICxVECTPRIORITY8	0x220	RW	Vector Priority 8 Register	0xF
VICxVECTPRIORITY9	0x224	RW	Vector Priority 9 Register	0xF
VICxVECTPRIORITY10	0x228	RW	Vector Priority 10 Register	0xF
VICxVECTPRIORITY11	0x22C	RW	Vector Priority 11 Register	0xF
VICxVECTPRIORITY12	0x230	RW	Vector Priority 12 Register	0xF
VICxVECTPRIORITY13	0x234	RW	Vector Priority 13 Register	0xF
VICxVECTPRIORITY14	0x238	RW	Vector Priority 14 Register	0xF
VICxVECTPRIORITY15	0x23C	RW	Vector Priority 15 Register	0xF
VICxVECTPRIORITY16	0x240	RW	Vector Priority 16 Register	0xF
VICxVECTPRIORITY17	0x244	RW	Vector Priority 17 Register	0xF
VICxVECTPRIORITY18	0x248	RW	Vector Priority 18 Register	0xF
VICxVECTPRIORITY19	0x24C	RW	Vector Priority 19 Register	0xF
VICxVECTPRIORITY20	0x250	RW	Vector Priority 20 Register	0xF

Register	Offset	Type	Description	Reset Value
VICxVECTPRIORITY21	0x254	RW	Vector Priority 21 Register	0xF
VICxVECTPRIORITY22	0x258	RW	Vector Priority 22 Register	0xF
VICxVECTPRIORITY23	0x25C	RW	Vector Priority 23 Register	0xF
VICxVECTPRIORITY24	0x260	RW	Vector Priority 24 Register	0xF
VICxVECTPRIORITY25	0x264	RW	Vector Priority 25 Register	0xF
VICxVECTPRIORITY26	0x268	RW	Vector Priority 26 Register	0xF
VICxVECTPRIORITY27	0x26C	RW	Vector Priority 27 Register	0xF
VICxVECTPRIORITY28	0x270	RW	Vector Priority 28 Register	0xF
VICxVECTPRIORITY29	0x274	RW	Vector Priority 29 Register	0xF
VICxVECTPRIORITY30	0x278	RW	Vector Priority 30 Register	0xF
VICxVECTPRIORITY31	0x27C	RW	Vector Priority 31 Register	0xF
VICxADDRESS	0xF00	RW	Vector Address Register	0x00000000

12.6 REGISTER DESCRIPTIONS

12.6.1 IRQ STATUS REGISTER, VICIRQSTATUS

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0IRQSTATUS	0x7120_0000	R	IRQ Status Register (VIC0)	0x0000_0000
VIC1IRQSTATUS	0x7130_0000	R	IRQ Status Register (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
IRQStatus	[31:0]	Show the status of the interrupts after masking by the VICxINTENABLE and VICxINTSELECT Registers: 0 = interrupt is inactive (reset) 1 = interrupt is active. There is one bit of the register for each interrupt source.	0x0

12.6.2 FIQ STATUS REGISTER, VICFIQSTATUS

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0FIQSTATUS	0x7120_0004	R	FIQ Status Register (VIC0)	0x0000_0000
VIC1FIQSTATUS	0x7130_0004	R	FIQ Status Register (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
FIQStatus	[31:0]	Show the status of the FIQ interrupts after masking by the VICINTENABLE and VICINTSELECT Registers: 0 = interrupt is inactive (reset) 1 = interrupt is active. There is one bit of the register for each interrupt source.	0x0

12.6.3 RAW INTERRUPT STATUS REGISTER, VICRAWINTR

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0RAWINTR	0x7120_0008	R	Raw Interrupt Status Register (VIC0)	0x0000_0000
VIC1RAWINTR	0x7130_0008	R	Raw Interrupt Status Register (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
RawInterrupt	[31:0]	Show 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.There is one bit of the register for each interrupt source.	0x0

12.6.4 INTERRUPT SELECT REGISTER, VICINTSELECT

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0INTSELECT	0x7120_000C	R/W	Interrupt Select Register (VIC0)	0x0000_0000
VIC1INTSELECT	0x7130_000C	R/W	Interrupt Select Register (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
IntSelect	[31:0]	Selects type of interrupt for interrupt request: 0 = IRQ interrupt (reset) 1 = FIQ interrupt There is one bit of the register for each interrupt source	0x0

12.6.5 INTERRUPT ENABLE REGISTER, VICINTENABLE

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0INTENABLE	0x7120_0010	R/W	Interrupt Enable Register (VIC0)	0x0000_0000
VIC1INTENABLE	0x7130_0010	R/W	Interrupt Enable Register (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
IntEnable	[31:0]	Enables the interrupt request lines, which allow the interrupts to reach the processor. Read: 0 = interrupt disabled (reset) 1 = Interrupt enabled The interrupt enable can only be set using this register. The VICINTENCLEAR Register must be used to disable the interrupt enable. Write: 0 = no effect 1 = interrupt enabled. On reset, all interrupts are disabled. There is one bit of the register for each interrupt source	0x0

12.6.6 INTERRUPT ENABLE CLEAR, VICINTENCLEAR

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0INTENCLEAR	0x7120_0014	W	Interrupt Enable Clear Register (VIC0)	-
VIC1INTENCLEAR	0x7130_0014	W	Interrupt Enable Clear Register (VIC1)	-

Name	BIT	DESCRIPTION	RESET VALUE
IntEnable Clear	[31:0]	Clears corresponding bits in the VICINTENABLE Register: 0 = no effect 1 = interrupt disabled in VICINTENABLE Register. There is one bit of the register for each interrupt source.	-

12.6.7 SOFTWARE INTERRUPT REGISTER, VICSOFTINT

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0SOFTINT	0x7120_0018	R/W	Software Interrupt Register (VIC0)	0x0000_0000
VIC1SOFTINT	0x7130_0018	R/W	Software Interrupt Register (VIC1)	0x0000_0000

Name	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 (reset) 1 = software interrupt active Write: 0 = no effect 1 = software interrupt enabled There is one bit of the register for each interrupt source	0x0

12.6.8 SOFTWARE INTERRUPT CLEAR REGISTER, VICSOFTINTCLEAR

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0SOFTINTCLEAR	0x7120_001C	W	Software Interrupt Clear Register (VIC0)	-
VIC1SOFTINTCLEAR	0x7130_001C	W	Software Interrupt Clear Register (VIC1)	-

Name	BIT	DESCRIPTION	RESET VALUE
SoftIntClear	[31:0]	Clears corresponding bits in the VICSOFTINT Register: 0 = no effect 1 = software interrupt disabled in the VICSOFTINT Register. There is one bit of the register for each interrupt source.	-

12.6.9 PROTECTION ENABLE REGISTER, VICPROTECTION

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0PROTECTION	0x7120_0020	R/W	Protection Enable Register (VIC0)	0x0000_0000
VIC1PROTECTION	0x7130_0020	R/W	Protection Enable Register (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
Reserved	[31:1]	Reserved, read as zero, do not modify.	0x0
Protection	[0]	Enables or disables protected register access: 0 = protection mode disabled (reset) 1 = protection mode enabled. When enabled, only privileged mode accesses (reads and writes) can access the interrupt controller registers. When disabled, both user mode and privileged mode can access the registers. This register can only be accessed in privileged mode, even when protection mode is disabled.	0

12.6.10 SOFTWARE PRIORITY MASK REGISTER, VICSWPRIORITYMASK

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0SWPRIORITY MASK	0x7120_0024	R/W	Software Priority Mask Register (VIC0)	0x0000_FFFF
VIC1SWPRIORITY MASK	0x7130_0024	R/W	Software Priority Mask Register (VIC1)	0x0000_FFFF

Name	BIT	DESCRIPTION	RESET VALUE
Reserved	[31:16]	Reserved, read as zero, 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 (reset). Each bit of the register is applied to each of the 16 interrupt priority levels.	0xFFFF

12.6.11 VECTOR PRIORITY REGISTER FOR DAISY CHAIN

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0PRIORITYDAISY	0x7120_0028	R/W	Vector Priority Register for Daisy Chain (VIC0)	0x0000_000F
VIC1PRIORITYDAISY	0x7130_0028	R/W	Vector Priority Register for Daisy Chain. (VIC1)	0x0000_000F

Name	BIT	DESCRIPTION	RESET VALUE
Reserved	[31:4]	Reserved, read as zero, do not modify.	0x0
PriorityDaisy	[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

12.6.12 VECTOR ADDRESS REGISGERS, VICVECTADDR[0-31]

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0VECTADDR[31:0]	0x7120_0100 ~ 0x7120_017C	R/W	Vector Address [31:0] Register (VIC0)	0x0000_0000
VIC0VECTADDR[31:0]	0x7130_0100 ~ 0x7130_017C	R/W	Vector Address [31:0] Register. (VIC1)	0x0000_0000

Name	BIT	DESCRIPTION	RESET VALUE
VectorAddr	[31:0]	Contains ISR vector addresses.	0x0000_0000

12.6.13 VECTOR PRIORITY REGISTERS, VICVECTPRIORITY[0-31]

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0VECTPRIORITY[31:0]	0x7120_0200 ~ 0x7120_027C	R/W	Vector Priority [31:0] Register (VIC0)	0x0000_000F
VIC1VECTPRIORITY[31:0]	0x7130_0200 ~ 0x7130_027C	R/W	Vector Priority [31:0] Register. (VIC1)	0x0000_000F

Name	BIT	DESCRIPTION	RESET VALUE
Reserved	[31:4]	Reserved, read as zero, 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

12.6.14 VECTOR ADDRESS REGISTER, VICADDRESS

REGISTER	ADDRESS	R/W	DESCRIPTION	RESET VALUE
VIC0ADDRESS	0x7120_0F00	R/W	Vector Address Register (VIC0)	0x0000_0000
VIC1ADDRESS	0x7130_0F00	R/W	Vector Address Register (VIC1)	0x0000_0000

Name	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 only 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.	0x0

13 SECURITY SUB-SYSTEMS

13.1 OVERVIEW

Security sub-system (SsS) is a crypto function accelerator. The architecture of SsS also provides high-speed bulk data processing, by providing double-layer AHB bus and FIFOs. FIFO-Rx and FIFO-Tx can be programmed to monitor AES or DES/3DES or SHA-1 module, and automatically transfer input/output data from the target module. This scheme does not require CPU's intervention and can achieve high-speed bulk data processing.

Figure 13-1 shows the block diagram of the SsS, and its main features are as follows.

13.2 FEATURES

- Symmetric key accelerator
  - AES : ECB, CBC, CTR mode support
  - DES/3DES : ECB, CBC mode support
- Hash engine
  - SHA-1
  - H/W HMAC support
- Random Number Generator
  - PRNG 320-bit generation per 160 cycles
- FIFO-Rx/Tx: (two 32-word) for input and output streaming.
  - AES, DES/3DES, SHA-1/HMAC mode support
- DMA I/F to SDMA1(Security DMA 1)

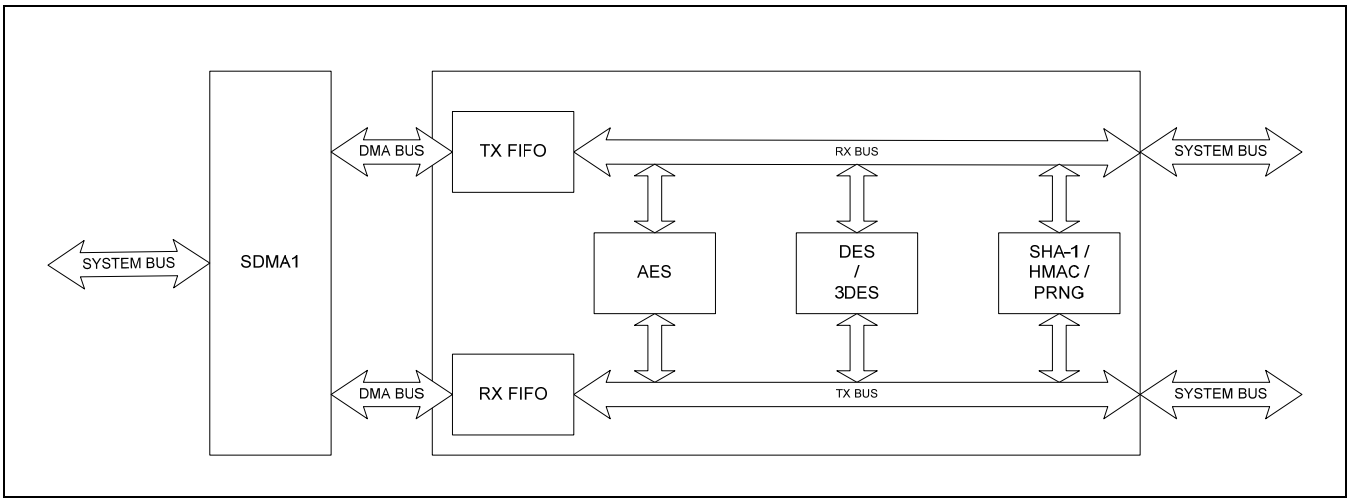


Figure 13-1. Block Diagram of the Security Sub-System

13.3 PROGRAMMING GUIDE

13.3.1 AES

AES can be executed by either FiFo Operation or ARM Operation depending on the transfer method of Data and Meta Data (IV, Counter) for Cipher execution. This document describes the FiFo Operation which is the basis.

1. Key and Meta Data(CBC Mode:IV(**AES_Rx_IV_0#**), CTR Mode: Counter Value(**AES_Rx_CTR_0#**), ECB Mode:Meta Data does not apply) are written in each register. Key is written to Key Register according to the Key Size(128/192/256-bit). (128-bit : **AES_Rx_KEY_01~AES_Rx_KEY_04**, 192-bit : **AES_Rx_KEY_01~AES_Rx_KEY_06**, 256-bit : **AES_Rx_KEY_01~AES_Rx_KEY_08**)

2. Set AES Configure Key Length for AES operation, Direction(En/Decryption), Operation Mode(ECB/CBC/CTR), Counter Size(for Counter Mode) for AES operation into AES Control Register.

3. Rx/Tx FiFo related registers are set up in the following order:
   (1) **FiFo-Rx Message Length Register** Setup : Total message length for AES Operation (32bit Unit)
   (2) **FiFo-Rx Block Size Register** Setup : AES Algorithm Specific Block Size (32bit Unit)
   (3) **FiFo-Rx Destination Register** Setup : AES Input Buffer(**AES_Rx_DIN_01**) Start Address
   (4) **FiFo-Tx Message** Setup : Total message length for AES Operation (32bit Unit)
   (5) **FiFo-Tx Block Size Register** Setup : AES Algorithm Specific Block Size (32bit Unit)
   (6) **FiFo-Tx Source Register** Setup : AES Output Buffer(**AES_Tx_DOUT_01)** Start Address
   (7) **FiFo-Rx Control Register** Setup : 1) '**FRx_Dest_Module**'(FRx_Ctrl[7:6] : Algorithm Specific Destination Module Selection(AES:00)) 2) '**FRx_Host_Rd_En**' (FRx_Ctrl[5] : Host FRx_Ctrl[31:16] and FRx_MlenCnt field Read Enable) 3) '**FRx_Host_Wr_En**'(FRx_Ctrl[4] : Host FiFo-Rx Write Enable) 4) '**FRx_Sync_Tx**'(FRx_Ctrl[3] : FiFo-Rx Write Wait Enable for FiFo-Tx Read Done) 5) '**FRx_Start**' (FRx_Ctrl[0] : FiFo-Rx Data Transfer to Destination Module Start)
   (8) **FiFo-Tx Control Register** Setup : 1) '**FTx_Src_Module**'(FTx_Ctrl[7:6] : Algorithm Specific Source Module Selection(AES:00)) 2) '**FTx_Host_Rd_En**'(FTx_Ctrl[5] : Host FTx_Ctrl[31:16] and FTx_MlenCnt field Read Enable) 3)' **FTx_Host_Wr_En**'(FTx_Ctrl[4] : Host FiFo-Tx Read Enable) 4) '**FTx_Start**' (FTx_Ctrl[0] : FiFo-Rx Data Transfer from Source Module Start)

4. Write Data(Encryption:PlainText, Decryption:CipherText) for AES Operation into FiFo-Rx. FiFo-Rx and FiFo-Tx communicates data with AES by following procedures.
   (1) **Write Data to AES_Rx_DIN_0# Register**
   (2) **AES Operation Start : <u>Repeat following Step from Step① to Step⑤ as needed</u>**
      ① **FiFo-Rx Polling AES Input Ready**
      ② **FiFo-Rx Write Data to AES_Rx_DIN_0# Register**
      ③ **FiFo-Tx Polling AES Output Ready**
      ④ **FiFo-Tx Read Data from AES_Tx_DOUT_0# Register**
      ⑤ **AES Operation Start**
   (3) **FiFo-Tx Polling AES Output Ready**
   (4) **FiFo-Tx Read Data from AES_Tx_DOUT_0# Register**

5. Uses FiFo-Rx Empty Check('**FRx_Wd2Read**'(checks if FRx_Ctrl[15:8]) is '0') to check and poll if AES Operation Data transmission from FiFo-Rx to AES is completed.
6. Confirm AES Operation result's transmission to FiFo-Tx by checking FiFo-Tx Transfer Done Check (check if '**FTx_Done**'(FTx_Ctrl[25]) is '1') and read AES Operation result from FiFo-Tx.

13.3.2 TDES

TDES can be executed by FiFo Operation or ARM Operation depending on transmission method of Data and Meta Data (IV) for Cipher. This document describes the FiFo Operation which is the basis.

1. Write Key (**TDES_Rx_KEY#_#**) and Meta Data(CBC Mode:IV(**TDES_Rx_IV_0#**), ECB Mode:Meta Data does not apply) to each register.
2. Configure either DES/TDES Selection, Direction(En/Decryption), or Operation Mode(ECB/CBC) into TDES Control Register (**TDES_Rx_CTRL**).
3. Set up Rx/Tx FiFo related Registers in following order.
   - **A. FiFo-Rx Message Length Register** Setup : Total message length for TDES Operation (32bit Unit)
   - **B. FiFo-Rx Block Size Register** Setup : TDES Algorithm Specific Block Size (32bit Unit)
   - **C. FiFo-Rx Destination Register** Setup : TDES Input Buffer(**TDES_Rx_INPUT_01**) Start Address
   - **D. FiFo-Tx Messgae** Setup : Total message length for TDES Operation (32bit Unit)
   - **E. FiFo-Tx Block Size Register** Setup : TDES Algorithm Specific Block Size (32bit Unit)
   - **F. FiFo-Tx Source Register** Setup : TDES Output Buffer(**TDES_Tx_OUTPUT_01)** Start Address
   - **G. FiFo-Rx Control Register** Setup : 1) '**FRx_Dest_Module**'(FRx_Ctrl[7:6] : Algorithme Specific Destination Module Selection(TDES/DES:01)) 2) '**FRx_Host_Rd _En**' (FRx_Ctrl[5] : Host FRx_Ctrl[31:16] and FRx_MlenCnt field Read Enable) 3) '**FRx_Host_Wr_En**'(FRx_Ctrl[4] : Host FiFo-Rx Write Enable) 4) '**FRx_Sync_ Tx**'(FRx_Ctrl[3] : FiFo-Rx Write Wait Enable for FiFo-Tx Read Done) 5) '**FRx_Start**' (FRx_Ctrl[0] : FiFo-Rx Data Transfer to Destnation Module Start)
   - **H. FiFo-Tx Control Register** Setup : 1) '**FTx_Src_Module**'(FTx_Ctrl[7:6] : Algorithm Specific Source Module Selection(TDES/DES:01)) 2) '**FTx_Host_Rd_En**' (FTx_Ctrl[5] : Host FTx_Ctrl[31:16] and FTx_MlenCnt field Read Enable) 3)' **FTx_Host_Wr_En**'(FTx_Ctrl[4] : Host FiFo-Tx Read Enable) 4) '**FTx_Start**' (FTx_Ctrl[0] : FiFo-Rx Data Transfer from Source Module Start)

4. Host writes Data(Encryption:PlainText, Decryption:CipherText) for TDES Operation into FiFo-Rx. FiFo-Rx and FiFo-Tx uses following method to transfer data between TDES.
   - **A. Write Data to TDES_Rx_INPUT_0# Register**
   - **B. TDES Operation Start : <u>Repeat following Step from Step① to Step⑤ as needed</u>**
     - **① FiFo-Rx Polling TDES Input Ready**
     - **② FiFo-Rx Write Data to TDES_Rx_INPUT_0# Register**
     - **③ FiFo-Tx Polling TDES Output Ready**
     - **④ FiFo-Tx Read Data from TDES_Tx_OUTPUT_0# Register**
     - **⑤ TDES Operation Start**
   - **C. FiFo-Tx Polling TDES Output Ready**
   - **D. FiFo-Tx Read Data from TDES_Tx_OUTPUT_0# Register**

5. Polls if all the data for TDES Operation are transmitted from FiFo-Rx to TDES by checking FiFo-Rx Empty Check (checks if '**FRx_Wd2Read**'(FRx_Ctrl[15:8]) is '0').

6. Confirms the transmission of TDES Operation result to FiFo-Tx by checking FiFo-Tx Transfer Done Check (check if '**FTx_Done**' is '1') and read the result from FiFo-Tx.

13.3.3 SHA-1

SHA-1 & PRNG is executed by either FiFo Operation or ARM Operation depending on transfer method of Data and Meta Data(IV) for hash execution. This document describes the FiFo Operation which is the basis.

1. Configure **HASH_CTRL Register**.(Select engine(SHA-1, HMAC-SHA-1), Key or text, engine start, seed or not)
   Note) Rx/Tx FiFo is not support PRNG engine.

2. Rx/Tx FiFo related registers are set up in following order:
   - **A. FiFo-Rx Message Length Register** Setup : Total message length of the operation (32bit Unit)
   - **B. FiFo-Rx Block Size Register** Setup : Hash Algorithm Specific Block Size (32bit Unit)
   - **C. FiFo-Rx Destination Register** Setup : Input Buffer Start Address
   - **D. FiFo-Tx Message Length Register** Setup : Total message length of the operation (32bit Unit)
   - **E. FiFo-Tx Block Size Register** Setup : Hash Algorithm Specific Block Size (32bit Unit)
   - **F. FiFo-Tx Source Register** Setup : Output Buffer Start Address
   - **G. FiFo-Rx Control Register** Setup : B1) '**FRx_Dest_Module**'(FRx_Ctrl[7:6] : Algorithme Specific Destination Module Selection(SHA-1/PRNG:01)) 2) '**FRx_Host_Rd_En**' (FRx_Ctrl[5] : Host FRx_Ctrl[31:16] and FRx_MlenCnt field Read Enable) 3) '**FRx_Host_Wr_En**'(FRx_Ctrl[4] : Host FiFo-Rx Write Enable) 4) '**FRx_Sync_Tx**'(FRx_Ctrl[3] : FiFo-Rx Write Wait Enable for FiFo-Tx Read Done) 5) '**FRx_Start**' (FRx_Ctrl[0] : FiFo-Rx Data Transfer to Destnation Module Start)
   - **H. FiFo-Tx Control Register** Setup : 1) '**FTx_Src_Module**'(FTx_Ctrl[7:6] : Algorithm Specific Source Module Selection(SHA-1/PRNG : 01))2) '**FTx_Host_Rd_En**'(FTx_Ctrl[5] : Host FTx_Ctrl[31:16] and FTx_MlenCnt field Read Enable) 3)' **FTx_Host_Wr_En**'(FTx_Ctrl[4] : Host FiFo-Tx Read Enable) 4) '**FTx_Start**' (FTx_Ctrl[0] : FiFo-Rx Data Transfer from Source Module Start)

3. Write data for Hash Operation to FiFo-Rx. FiFo-Rx and FiFo-Tx transmits Hash and Data by following method:
   - **A. Write Data to Hash_Rx_DIN_0 ~ HASH_Rx_DIN_15 Registers**
   - **B. Operation Start: Repeat following Step from Step① to Step② as needed**
     - ① **FiFo-Rx Polling Input Ready**
     - ② **FiFo-Rx Write Data to Hash_Rx_DIN_0 ~ Hash_Rx_DIN_15 Register**
   - **C. FiFo-Tx Polling SHA-1 Output Ready**
   - **D. FiFo-Tx Read Data from HASH_OUTPUT_0# Register**

4. Polls if data for operation is transmitted from FIFo-Rx to SHA-1/PRNG by checking FiFo-Rx Empty Check (check if '**FRx_Wd2Read**'(FRx_Ctrl[15:8]) is '0').

5. Check if SHA-1/PRNG Opearation result is transmitted to FiFo-Tx by checking FiFo-Tx Transfer Done Check (check if '**FTx_Done**'(FTx_Ctrl[25]) is '1') and read from FiFo-Tx.

13.4 SPECIAL FUNCTION REGISTERS

13.4.1 SECURITY SUB-SYSTEM REGISTER MAP

Table 13-1. DMA & Interrupt Control Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0x0000_0000	DnI_CFG	DMA and interrupt configuration register
* Base = 0x7D00_0000				

Table 13-2. FIFO-Rx Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0x0420_0000	FRx _Ctrl	FIFO-Rx Control & Status register
Base + 0x04	R/W	0x0000_0000	FRx _MLen	FIFO-Rx Message Length register
Base + 0x08	R/W	0x0000_0000	FRx _BlkSz	FIFO-Rx Crypto algorithm block size register
Base + 0x0C	R/W	0x0000_0000	FRx _DestAddr	FIFO-Rx Inout Buffer Address register
Base + 0x10	R/W	0x0000_0000	FRx _MLenCnt	FIFO-Rx Message Count register
Base + 0x40	W	0x0000_0000	FRx _WrBuf	FIFO-Rx write buffer
…	…	…	…	…
Base + 0x7C	W	0x0000_0000	FRx _WrBuf	FIFO-Rx write buffer
* Base = 0x7D40_0000 * Base=0x7D90_0000 (Have to use this address to transfer using SDMA1, SDMA1 only see this address.)				

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 0x007C will trigger the FIFO memory write. This feature lets the programmer use burst write to the FIFO-Rx. The FIFO Data path is not support PRNG module.

Table 13-3. FIFO-TX Register Map

Address	R/W	Reset value	Name	Description
Base + 0x00	R/W	0x0420_0000	FTx _Ctrl	FIFO-Tx Control & Status register
Base + 0x04	R/W	0x0000_0000	FTx _MLen	FIFO-Tx Message Length register
Base + 0x08	R/W	0x0000_0000	FTx _BlkSz	FIFO-Tx Crypto algorithm block size register
Base + 0x0C	R/W	0x0000_0000	FTx _DestAddr	FIFO-Tx Inout Buffer Address register
Base + 0x10	R/W	0x0000_0000	FTx _MLenCnt	FIFO-Tx Message Count register
Base + 0x40	R	0x0000_0000	FTx _RdBuf	FIFO-Tx read buffer
…	…	…	…	…
Base + 0x7C	R	0x0000_0000	FTx _RdBuf	FIFO-Tx read buffer
* Base = 0x7D80_0000 * Base=0x7DA0_0000 (Have to use this address to transfer using SDMA1, SDMA1 only see this address.)				

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 will trigger the FIFO memory read. This feature makes the programmer use burst read to the FIFO-Tx. The FIFO Data path is not support PRNG module.

Table 13-4. AES Register Map

Address	R/W	Reset value	Name	Description
Rx-AES Register Map (Rx side)				
Base + 0x00	R/W	0x0000_0200	AES_Rx_CTRL	AES Rx Contrl / Status Reg.
Base + 0x10	R/W	0x0000_0000	AES_Rx_DIN_01	AES Rx Data Input Reg. 01
Base + 0x14	R/W	0x0000_0000	AES_Rx_DIN_02	AES Rx Data Input Reg. 02
Base + 0x18	R/W	0x0000_0000	AES_Rx_DIN_03	AES Rx Data Input Reg. 03
Base + 0x1C	R/W	0x0000_0000	AES_Rx_DIN_04	AES Rx Data Input Reg. 04
Base + 0x20	R	0x0000_0000	AES_Rx_DOUT_01	AES Rx Data Output Reg. 01
Base + 0x24	R	0x0000_0000	AES_Rx_DOUT_02	AES Rx Data Output Reg. 02
Base + 0x28	R	0x0000_0000	AES_Rx_DOUT_03	AES Rx Data Output Reg. 03
Base + 0x2C	R	0x0000_0000	AES_Rx_DOUT_04	AES Rx Data Output Reg. 04
Base + 0x80	R/W	0x0000_0000	AES_Rx_KEY_01	AES Rx Key Input Reg. 01
Base + 0x84	R/W	0x0000_0000	AES_Rx_KEY_02	AES Rx Key Input Reg. 02
Base + 0x88	R/W	0x0000_0000	AES_Rx_KEY_03	AES Rx Key Input Reg. 03
Base + 0x8C	R/W	0x0000_0000	AES_Rx_KEY_04	AES Rx Key Input Reg. 04
Base + 0x90	R/W	0x0000_0000	AES_Rx_KEY_05	AES Rx Key Input Reg. 05
Base + 0x94	R/W	0x0000_0000	AES_Rx_KEY_06	AES Rx Key Input Reg. 06
Base + 0x98	R/W	0x0000_0000	AES_Rx_KEY_07	AES Rx Key Input Reg. 07
Base + 0x9C	R/W	0x0000_0000	AES_Rx_KEY_08	AES Rx Key Input Reg. 08
Base + 0xA0	R/W	0x0000_0000	AES_Rx_IV_01	AES Rx IV Input Reg. 01

Address	R/W	Reset value	Name	Description
Base + 0xA4	R/W	0x0000_0000	AES_Rx_IV_02	AES Rx IV Input Reg. 02
Base + 0xA8	R/W	0x0000_0000	AES_Rx_IV_03	AES Rx IV Input Reg. 03
Base + 0xAC	R/W	0x0000_0000	AES_Rx_IV_04	AES Rx IV Input Reg. 04
Base + 0xB0	R/W	0x0000_0000	AES_Rx_CTR_01	AES Rx Counter Preload Reg. 01
Base + 0xB4	R/W	0x0000_0000	AES_Rx_CTR_02	AES Rx Counter Preload Reg. 02
Base + 0xB8	R/W	0x0000_0000	AES_Rx_CTR_03	AES Rx Counter Preload Reg. 03
Base + 0xBC	R/W	0x0000_0000	AES_Rx_CTR_04	AES Rx Counter Preload Reg. 04
Tx-AES Register Map (Tx side)				
Base + 0x20	R	0x0000_0000	AES_Tx_DOUT_01	AES Rx Data Output Reg. 01
Base + 0x24	R	0x0000_0000	AES_Tx_DOUT_02	AES Rx Data Output Reg. 02
Base + 0x28	R	0x0000_0000	AES_Tx_DOUT_03	AES Rx Data Output Reg. 03
Base + 0x2C	R	0x0000_0000	AES_Tx_DOUT_04	AES Rx Data Output Reg. 04
* Rx Base = 0x7D10_0000 * Tx Base = 0x7D50_0000				

Table 13-5. DES/TDES Register Map

Address	R/W	Reset value	Name	Description
Rx-DES/3DES Register Map (Rx side)				
Base + 0x00	R/W	0x0000_0040	TDES_Rx_CTRL	DES/3DES Rx Contrl / Status Reg.
Base + 0x10	R/W	0x0000_0000	TDES_Rx_KEY1_0	DES/3DES Rx Key Input Reg. 1_0
Base + 0x14	R/W	0x0000_0000	TDES_Rx_KEY1_1	DES/3DES Rx Key Input Reg. 1_1
Base + 0x18	R/W	0x0000_0000	TDES_Rx_KEY2_0	DES/3DES Rx Key Input Reg. 2_0
Base + 0x1C	R/W	0x0000_0000	TDES_Rx_KEY2_1	DES/3DES Rx Key Input Reg. 2_1
Base + 0x20	R/W	0x0000_0000	TDES_Rx_KEY3_0	DES/3DES Rx Key Input Reg. 3_0
Base + 0x24	R/W	0x0000_0000	TDES_Rx_KEY3_1	DES/3DES Rx Key Input Reg. 3_1
Base + 0x40	R/W	0x0000_0000	TDES_Rx_INPUT_0	DES/3DES Rx Data Input Reg. 0
Base + 0x44	R/W	0x0000_0000	TDES_Rx_INPUT_1	DES/3DES Rx Data Input Reg. 1
Base + 0x48	R	0x0000_0000	TDES_Rx_OUTPUT_0	DES/3DES Rx Output Data Reg. 0
Base + 0x4C	R	0x0000_0000	TDES_Rx_OUTOUT_1	DES/3DES Rx Output Data Reg. 1
Base + 0x50	R/W	0x0000_0000	TDES_Rx_IV_0	TDES Rx IV Input Register 0
Base + 0x54	R/W	0x0000_0000	TDES_Rx_IV_1	TDES Rx IV Input Register 1
Tx-DES/3DES Register Map (Tx side)				
Base + 0x48	R	0x0000_0000	TDES_Tx_OUTPUT_0	DES/3DES Rx Output Data Reg. 0
Base + 0x4C	R	0x0000_0000	TDES_Tx_OUTPUT_1	DES/3DES Rx Output Data Reg. 1
* Rx Base = 0x7D20_0000 * Tx Base = 0x7D60_0000				

Table 13-6. SHA-1/PRNG Register Map

Address	R/W	Reset value	Name	Description
Rx-SHA-1/PRNG Register Map (Rx side)				
Base + 0x00	R/W	0x0000_0000	HASH_CONTROL	Hash engine control Reg.
Base + 0x04	R/W	0x0000_0000	HASH_DATA	HASH data or HMAC Key Input Reg.
Base + 0x08	R/W	0x0000_0000	HASH_DATA	HASH data or HMAC Key Input Reg.
…	…	…	…	…
Base + 0x20	R/W	0x0000_0000	HASH_DATA	HASH data or HMAC Key Input Reg.
Base + 0x08	R/W	0x0000_0000	SEED_DATA_01	PRNG Seed data[31:0]
Base + 0x0C	R/W	0x0000_0000	SEED_DATA_02	PRNG Seed data[63:32]
Base + 0x10	R/W	0x0000_0000	SEED_DATA_03	PRNG Seed data[95:64]
Base + 0x14	R/W	0x0000_0000	SEED_DATA_04	PRNG Seed data[127:96]
Base + 0x18	R/W	0x0000_0000	SEED_DATA_05	PRNG Seed data[159:128]
Base + 0x1C	R/W	0x0000_0000	SEED_DATA_06	PRNG Seed data[191:160]
Base + 0x20	R/W	0x0000_0000	SEED_DATA_07	PRNG Seed data[223:192]
Base + 0x24	R/W	0x0000_0000	SEED_DATA_08	PRNG Seed data[255:224]
Base + 0x28	R/W	0x0000_0000	SEED_DATA_09	PRNG Seed data[287:256]
Base + 0x2C	R/W	0x0000_0000	SEED_DATA_10	PRNG Seed data[319:288]
Base + 0x30	R	0x0000_0010	HASH_STATUS	Status check
Base + 0x34	R	0x0000_0000	HASH_OUTPUT_01 PRNG_OUTPUT_01	HASH (PRNG) output (h0)
Base + 0x38	R	0x0000_0000	HASH_OUTPUT_02 PRNG_OUTPUT_02	HASH (PRNG) output (h1)
Base + 0x3C	R	0x0000_0000	HASH_OUTPUT_03 PRNG_OUTPUT_03	HASH (PRNG) output (h2)
Base + 0x40	R	0x0000_0000	HASH_OUTPUT_04 PRNG_OUTPUT_04	HASH (PRNG) output (h3)
Base + 0x44	R	0x0000_0000	HASH_OUTPUT_05 PRNG_OUTPUT_05	HASH (PRNG) output (h4)
Base + 0x48	R	0x0000_0000	HASH_OUTPUT_06	PRNG output
Base + 0x4C	R	0x0000_0000	HASH_OUTPUT_07	PRNG output
Base + 0x50	R	0x0000_0000	HASH_OUTPUT_08	PRNG output
Base + 0x54	R	0x0000_0000	HASH_OUTPUT_09	PRNG output
Base + 0x58	R	0x0000_0000	HASH_OUTPUT_10	PRNG output
Base + 0x5C	R	0x0000_0000	HASH_MIDOUT_01	HASH_MIDOUT[159:128]
Base + 0x60	R	0x0000_0000	HASH_MIDOUT_02	HASH_MIDOUT[127:96]

Address	R/W	Reset value	Name	Description
Base + 0x64	R	0x0000_0000	HASH_MIDOUT_03	HASH_MIDOUT[95:64]
Base + 0x68	R	0x0000_0000	HASH_MIDOUT_04	HASH_MIDOUT[63:32]
Base + 0x6C	R	0x0000_0000	HASH_MIDOUT_05	HASH_MIDOUT[31:0]
Base + 0x70	W	0x0000_0000	HASH_IV_01	HASH initial value 01
Base + 0x74	W	0x0000_0000	HASH_IV_02	HASH initial value 02
Base + 0x78	W	0x0000_0000	HASH_IV_03	HASH initial value 03
Base + 0x7C	W	0x0000_0000	HASH_IV_04	HASH initial value 04
Base + 0x80	W	0x0000_0000	HASH_IV_05	HASH initial value 05
Base + 0x84	W	0x0000_0000	PRE_MSG_LENGT 1	Pre HASH length [63:32]
Base + 0x88	W	0x0000_0000	PRE_MSG_LENGT 2	Pre HASH length [31: 0]
Tx-SHA-1/PRNG Register Map (Tx side)				
Base + 0x34	R	0x0000_0000	HASH_OUTPUT_01 PRNG_OUTPUT_01	HASH (PRNG) output (h0)
Base + 0x38	R	0x0000_0000	HASH_OUTPUT_02 PRNG_OUTPUT_02	HASH (PRNG) output (h1)
Base + 0x3C	R	0x0000_0000	HASH_OUTPUT_03 PRNG_OUTPUT_03	HASH (PRNG) output (h2)
Base + 0x40	R	0x0000_0000	HASH_OUTPUT_04 PRNG_OUTPUT_04	HASH (PRNG) output (h3)
Base + 0x44	R	0x0000_0000	HASH_OUTPUT_05 PRNG_OUTPUT_05	HASH (PRNG) output (h4)
Base + 0x48	R	0x0000_0000	HASH_OUTPUT_06	PRNG output
Base + 0x4C	R	0x0000_0000	HASH_OUTPUT_07	PRNG output
Base + 0x50	R	0x0000_0000	HASH_OUTPUT_08	PRNG output
Base + 0x54	R	0x0000_0000	HASH_OUTPUT_09	PRNG output
Base + 0x58	R	0x0000_000	HASH_OUTPUT_10	PRNG output

		0		
Base + 0x5C	R	0x0000_0000	HASH_MIDOUT_01	HASH_MIDOUT[159:128]
Base + 0x60	R	0x0000_0000	HASH_MIDOUT_02	HASH_MIDOUT[127:96]
Base + 0x64	R	0x0000_0000	HASH_MIDOUT_03	HASH_MIDOUT[95:64]
Base + 0x68	R	0x0000_0000	HASH_MIDOUT_04	HASH_MIDOUT[63:32]
Base + 0x6C	R	0x0000_0000	HASH_MIDOUT_05	HASH_MIDOUT[31:0]
* Rx Base = 0x7D30_0000 * Tx Base = 0x7D70_0000				

13.5 DMA & INTERRUPT CONTROL MODULE

13.5.1 SECURITY SUB-SYSTEM DMA & INTERRUPT REGISTER

Register	Address	R/W	Description	Reset Value
DnI_Cfg	0x7D00_0000	R/W	DMA and interrupt configuration Control & Status Reg.	0x0000_0000

DnI_Cfg	Bit	Description	Initial State
WrPrivMismatch	[31]	SFR Write Access Privilege Mismatch Status bit. If set to '1', SFR Write Access Privilege Mismatch is occurred.	0b
RdPrivMismatch	[30]	SFR Read Access Privilege Mismatch Status bit. If set to '1', SFR Read Access Privilege Mismatch is occurred.	0b
Reserved	[29:23]	Reserved	0x00
SHA_intr_Status	[22]	SHA-1/PRNG interrupt status and pending bit. This is cleared when read.	0b
DES_intr_Status	[21]	DES/3DES interrupt status and pending bit. This is cleared when read.	0b
AES_intr_Status	[20]	AES interrupt status and pending bit. This is cleared when read.	0b
Reserved	[19:18]	Reserved	00b
FTx_intr_Status	[17]	FIFO-Tx interrupt status and pending bit. This is cleared when read.	0b
FRx_intr_Status	[16]	FIFO-Rx interrupt status and pending bit. This is cleared when read.	0b
Reserved	[15]	Reserved	0b
SHA_intr_En	[14]	SHA-1/PRNG interrupt enabled when finished.	0b
DES_intr_En	[13]	DES/3DES interrupt enabled when finished.	0b
AES_intr_En	[12]	AES interrupt enabled when finished.	0b
Reserved	[11:10]	Reserved	00b
FTx_intr_En	[9]	FIFO-Tx interrupt enabled when finished.	0b
FRx_intr_En	[8]	FIFO-Rx interrupt enabled when finished.	0b
TxTrgLevel	[7:5]	Tx side DMA trigger level setting. 000 = 1-word 001 = 4-word 010 = 8-word 011 = 12-word 100 = 16-word 101 = 20-word 110 = 24-word 111 = 28-word	000b
TxDmaEnb	[4]	Tx side DMA enable bit (1: enable)	0b
RxTrgLevel	[3:1]	Rx side DMA trigger level setting. 000 = 1-word 001 = 4-word 010 = 8-word 011 = 12-word 100 = 16-word 101 = 20-word 110 = 24-word 111 = 28-word	000b
RxDmaEnb	[0]	Rx side DMA enable bit (1: enable)	0b

13.6 SECURITY SUB-SYSTEM RX FIFO MODULE

13.6.1 FIFO-RX CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FRx_Ctrl	0x7D40_0000	R/W	FIFO-Rx Control/Status Reg. (Only MSB 16-bit readable)	0x0420_0000

FRx_Ctrl	Bit	Description	Initial State
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).	0b
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).	0b
Reserved	[29]	Reserved	0b
FRx_Full	[27]	Sets to 1 if FIFO-Rx buffer (FRx_WrBuf) is full.	0b
FRx_Empty	[26]	Sets to 1 if FIFO-Rx buffer (FRx_WrBuf) is empty.	1b
FRx_Done	[25]	Sets to 1 if FIFO-Rx has finished transferring FRx_MLen words of data to the destination.	0b
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 when FRx_Start bit resets to 0.	0b
FRx_Wd2Write	[23:16]	Number of words that can be written to FIFO memory (FRx_WrBuf)	0x00
FRx_Wd2Read	[15:8]	Number of words that can be read from FIFO memory (FRx_WrBuf)	0x00
FRx_Dest_Module	[7:6]	Destination module selection. (00 : AES, 01:DES/3DES, 10: SHA-1/HMAC, 11: Reserved)	0b
FRx_Host_Rd_En	[5]	Enables Host read from FRx_Ctrl[31:16] and FRx_MLenCnt.	0b
FRx_Host_Wr_En	[4]	Enables Host write to FRx_WrBuf	0b
FRx_Sync_Tx	[3]	When enabled, FIFO-Rx waits for FIFO-Tx to retrieve output data from source module before transferring data to the destination module.	0b
FRx_Reset	[2]	Stops current FIFO-Rx transfer and resets FSM and all the register.	0b
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].	0b
FRx_Start	[0]	FIFO-Rx transfer start bit. Resets to 0 when internal FSM starts transferring data to destination.	0b

13.6.2 FIFO-RX MESSAGE LENGTH REGISTER

Register	Address	R/W	Description	Reset Value
FRx_MLen	0x7D40_0004	R/W	FIFO-Rx Message Length Reg.	0x0000_0000

FRx_MLen	Bit	Description	Initial State
FRx_MLen	[31:0]	Message Length in word (32-bit) unit. Resets to its reset value when FRx_Reset field of FRx_Ctrl register is set.	0x0000_0000

13.6.3 FIFO-RX BLOCK SIZE REGISTER

Register	Address	R/W	Description	Reset Value
FRx_BlkSz	0x7D40_0008	R/W	FIFO-Rx Crypto Algorithm Block Size Reg.	0x0000_0000

FRx_BlkSz	Bit	Description	Initial State
Reserved	[31:18]	Reserved	0x0000
LastValidByte	[17:16]	Last valid byte in the last word transferred to the SHA-1 module (Only valid when 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 when FRx_Reset field of FRx_Ctrl register is set.	00b
BlkSz	[15:0]	Block size of destination module(in word(32-bit) unit). FIFO-Rx will transfer 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 when FRx_Reset field of FRx_Ctrl register is set.	0x0000

13.6.4 FIFO-RX DESTINATION ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
FRx_DestAddr	0x7D40_000C	R/W	FIFO-Rx Input Buffer Address Reg.	0x0000_0000

SKEY_IDx	Bit	Description	Initial State
DestAddr	[31:0]	Internal memory address of destination input buffer. FIFO-Rx will transfer data to this address. Resets to its reset value when FRx_Reset field of FRx_Ctrl register is set.	0x0000_0000

13.6.5 FIFO-RX MESSAGE LENGTH COUNTER

Register	Address	R/W	Description	Reset Value
FRx_MLenCnt	0x7D40_0010	R/W	FIFO-Rx Message Count Reg. (Number of words left)	0x0000_0000

FRx_MLenCnt	Bit	Description	Initial State
FRx_MLenCnt	[31:0]	Number of words left for transfer.	0x0000_0000

13.6.6 FIFO-RX WRITE BUFFER

Register	Address	R/W	Description	Reset Value
FRx_WrBuf	0x7D40_0040 ~ 0x7D40_007C	W	FIFO-Rx write buffer (32-word) Note: This address is for CPU access.	0x0000_0000
	0x7D90_0040 ~ 0x7D90_007C		FIFO-Rx write buffer (32-word) Note: This address is for SDMA1(Security DMA 1). You should use it to transfer from memory to FRx_WrBuf using SDMA1.	

FRx_WrBuf	Bit	Description	Initial State
FRx_WrBuf	[31:0]	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 0x007C will trigger the FIFO memory write. This feature lets the programmer use burst write to the FIFO-Rx.

13.7 SECURITY SUB-SYSTEM TX FIFO MODULE

13.7.1 FIFO-TX CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FTx_Ctrl	0x7D80_0000	R/W	FIFO-Tx Control/Status Reg. (Only MSB 16-bit readable)	0x0400_2000

FTx_Ctrl	Bit	Description	Initial State
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).	0b
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).	0b
Reserved	[29:28]	Reserved	00b
Full	[27]	Sets to 1 if FIFO-Tx buffer (FTx_RdBuf) is full.	0b
Empty	[26]	Sets to 1 if FIFO-Tx buffer (FTx_RdBuf) is empty.	1b
Done	[25]	Sets to 1 if FIFO-Tx has finished transferring FTx_MLen words of data from the source.	0b
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.	0b
Wd2Read	[23:16]	Number of words that can be read from FIFO memory (FTx_RdBuf)	0x00
Wd2Write	[15:8]	Number of words that can be written to FIFO memory (FTx_RdBuf)	0x20
Src_Module	[7:6]	Source module selection. 00 : AES 01:DES/3DES 10: SHA-1/HMAC 11: reserved	0b
Host_Rd_En	[5]	Enables Host read from FTx_Ctrl[31:16] and FTx_MLenCnt.	0b
Host_Wr_En	[4]	Enables Host read from FTx_RdBuf	0b
Reset	[2]	Stops current FIFO-Tx transfer and resets FSM and all the register.	0b
ERROR_En	[1]	Enables ERROR response via HRESP port when host tries to access FIFO-Tx and access is not enabled by FTx_Ctrl[4] or [5].	0b
Start	[0]	FIFO-Tx transfer start bit. Resets to 0 when internal FSM starts transferring data to destination.	0b

13.7.2 FIFO-TX MESSAGE LENGTH REGISTER

Register	Address	R/W	Description	Reset Value
FTx_MLen	0x7D80_0004	R/W	FIFO-Tx Message Length Reg.	0x0000_0000

FTx_MLen	Bit	Description	Initial State
FTx_MLen	[31:0]	Message Length in word(32-bit) unit. Resets to its reset value when FTx_Reset field of FTx_Ctrl register is set.	0x0000_0000

13.7.3 FIFO-TX BLOCK SIZE REGISTER

Register	Address	R/W	Description	Reset Value
FTx_BlkSz	0x7D80_0008	R/W	FIFO-Tx Crypto Algorithm Block Size Reg.	0x0000_0000

FTx_BlkSz	Bit	Description	Initial State
FTx_BlkSz	[31:0]	Block size of destination module(in word(32-bit) unit). FIFO-Tx will transfer 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 when FTx_Reset field of FTx_Ctrl register is set.	0x0000_0000

13.7.4 FIFO-TX SOURCE ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
FTx_SrcAddr	0x7D80_000C	R/W	FIFO-Tx output Buffer Address Reg.	0x0000_0000

FTx_SrcAddr	Bit	Description	Initial State
SrcAddr	[31:0]	Internal memory address of source output buffer. FIFO-Tx will transfer data from this address. Resets to its reset value when FTx_Reset field of FTx_Ctrl register is set.	0x0000_0000

13.7.5 FIFO-TX MESSAGE LENGTH COUNTER

Register	Address	R/W	Description	Reset Value
FTx_MLenCnt	0x7D80_0010	R/W	FIFO-Tx Message Count Reg. (Number of words left)	0x0000_0000

FTx_MLenCnt	Bit	Description	Initial State
MLenCnt	[31:0]	Number of words left for transfer.	0x0000_0000

13.7.6 FIFO-TX READ BUFFER

Register	Address	R/W	Description	Reset Value
FTx_RdBuf	0x7D80_0040 ~ 0x7D80_007C	R/W	FIFO-Tx read buffer (32-word) Note: This address is for CPU access.	0x0000_0000
	0x7DA0_0040 ~ 0x7DA0_007C		FIFO-Tx read buffer (32-word) Note: This address is for SDMA1(Security DMA 1). You should use it to transfer from FTx_RdBuf to memory using SDMA1.	

FTx_RdBuf	Bit	Description	Initial State
RdBuf	[31:0]	FIFO-Tx read buffer (32x32-bit)	0x0000_0000

NOTE: 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 will trigger the FIFO memory read. This feature lets the programmer use burst read from FIFO-Tx.

13.8 SECURITY SUB-SYSTEM AES MODULE

13.8.1 AES_CTRL

Register	Address	R/W	Description	Reset Value
AES_Rx_CTRL	0x7D10_0000	R/W	AES Control / Status Register	0x0000_0200

SKEY_IDx	Bit	Description	Initial State
WrPrivMismatch	[31]	SFR Write Access Privilege Mismatch Status bit. If set to '1', SFR Write Access Privilege Mismatch is occurred.	0b
RdPrivMismatch	[30]	SFR Read Access Privilege Mismatch Status bit. If set to '1', SFR Read Access Privilege Mismatch is occurred.	0b
Reserved	[29:11]	Reserved	0x000000
AesOutReady	[10]	If set to '1', AES Output Buffer is Full, and ARM or Rx FiFo is permitted to Read current 128 bits result data	0b
AesInReady	[9]	If set to '1', AES Input Buffer is Empty, and ARM or Rx FiFo is permitted to write next 128bits data.	1b
AesContDecOn	[8]	Continuous Decryption Enable Bits 0 : Decryption Key is changed 1 : Decryption Key is not changed	0b
CtrWidth	[7:6]	Counter Mode Counter Width Bits 00 : 16Bits Counter 01 : 32Bits Counter 10 : 64Bits Counter 11 = Reserved	00b
AesOpMode	[5:4]	AES Operation Mode Selection Bits 00 = reserved 01 = ECB Mode 10 = CBC Mode 11 = CTR Mode	00b
AesOpDirection	[3]	AES Operation Direction Selection Bit. 0 : Encryption 1 : Decryption	0b
AesKeyMode	[2:1]	AES Key Mode Selection Bits. 00 : 128bits 01 : 192bits 10 : 256bits 11 = reseved	00b
AesOpEnable	[0]	If set to '1', AES starts operation by ARM. If the aes_op_done is generated, It becomes '0'.	0b

13.8.2 AES_RX_DIN_01 ~ AES_RX_DIN_01

Register	Address	R/W	Description	Reset Value
AES_Rx_DIN_01	0x7D10_0010	R/W	AES Data Input Register 01	0x0000_0000
AES_Rx_DIN_02	0x7D10_0014	R/W	AES Data Input Register 02	0x0000_0000
AES_Rx_DIN_03	0x7D10_0018	R/W	AES Data Input Register 02	0x0000_0000
AES_Rx_DIN_04	0x7D10_001C	R/W	AES Data Input Register 03	0x0000_0000

AES_Rx_DIN	Bit	Description	Initial State
AesDin	[31:0]	AES 1st~ 4th 32-bit Data Input Register	0x0000_0000

13.8.3 AES_RX_DOUT_01 ~ AES_RX_DOUT_04 / AES_TX_DOUT_01 ~ AES_TX_DOUT_04

Register	Address	R/W	Description	Reset Value
AES_Rx_DOUT_01	0x7D10_0020	R/W	AES Data Output Register 01 (Least Significant)	0x0000_0000
AES_Rx_DOUT_02	0x7D10_0024	R/W	AES Data Output Register 02 (Least Significant)	0x0000_0000
AES_Rx_DOUT_03	0x7D10_0028	R/W	AES Data Output Register 03 (Least Significant)	0x0000_0000
AES_Rx_DOUT_04	0x7D10_002C	R/W	AES Data Output Register 04 (Least Significant)	0x0000_0000
AES_Tx_DOUT_01	0x7D50_0020	R/W	AES Data Output Register 01 (Least Significant)	0x0000_0000
AES_Tx_DOUT_02	0x7D50_0024	R/W	AES Data Output Register 02 (Least Significant)	0x0000_0000
AES_Tx_DOUT_03	0x7D50_0028	R/W	AES Data Output Register 03 (Least Significant)	0x0000_0000
AES_Tx_DOUT_04	0x7D50_002C	R/W	AES Data Output Register 04 (Least Significant)	0x0000_0000

AES_Tx_DOUT	Bit	Description	Initial State
AesDout	[31:0]	AES 1st ~ 4th 32-bit Data Output Register	0x0000_0000

13.8.4 AES_RX_KEY_01 ~ AES_RX_KEY_08

Register	Address	R/W	Description	Reset Value
AES_Rx_KEY_01	0x7D10_0080	R/W	AES Key Input Register 01	0x0000_0000
AES_Rx_KEY_02	0x7D10_0084	R/W	AES Key Input Register 02	0x0000_0000
AES_Rx_KEY_03	0x7D10_0088	R/W	AES Key Input Register 03	0x0000_0000
AES_Rx_KEY_04	0x7D10_008C	R/W	AES Key Input Register 04	0x0000_0000
AES_Rx_KEY_05	0x7D10_0090	R/W	AES Key Input Register 05	0x0000_0000
AES_Rx_KEY_06	0x7D10_0094	R/W	AES Key Input Register 06	0x0000_0000
AES_Rx_KEY_07	0x7D10_0098	R/W	AES Key Input Register 07	0x0000_0000
AES_Rx_KEY_08	0x7D10_009C	R/W	AES Key Input Register 08	0x0000_0000

AES_Rx_KEY	Bit	Description	Initial State
AesKey	[31:0]	AES 1st ~ 8th 32-bit Key Input Register	0x0000_0000

13.8.5 AES_RX_IV_01 ~ AES_RX_IV_04

Register	Address	R/W	Description	Reset Value
AES_Rx_IV_01	0x7D10_00A0	R/W	AES IV Input Register 01	0x0000_0000
AES_Rx_IV_02	0x7D10_00A4	R/W	AES IV Input Register 02	0x0000_0000
AES_Rx_IV_03	0x7D10_00A8	R/W	AES IV Input Register 03	0x0000_0000
AES_Rx_IV_04	0x7D10_00AC	R/W	AES IV Input Register 04	0x0000_0000

AES_Rx_IV	Bit	Description	Initial State
AesIv	[31:0]	AES 1st 32-bit IV Input Register	0x0000_0000

13.8.6 AES_RX_CTR_01 ~ AES_RX_CTR_04

Register	Address	R/W	Description	Reset Value
AES_Rx_CTR_01	0x7D10_00B0	R/W	AES Counter Preload Input Register 01	0x0000_0000
AES_Rx_CTR_02	0x7D10_00B4	R/W	AES Counter Preload Input Register 02	0x0000_0000
AES_Rx_CTR_03	0x7D10_00B8	R/W	AES Counter Preload Input Register 03	0x0000_0000
AES_Rx_CTR_04	0x7D10_00BC	R/W	AES Counter Preload Input Register 04	0x0000_0000

AES_Rx_CTR	Bit	Description	Initial State

AesCtr	[31:0]	AES 1st ~ 8th 32-bit Counter Preload Input Register	0x0000_0000

13.9 SECURITY SUB-SYSTEM DES/3DES MODULE

13.9.1 TDES_RX_CTRL

Register	Address	R/W	Description	Reset Value
TDES_Rx_CTRL	0x7D20_0000	R/W	TDES control / status register	0x0000_0040

TDES_Rx_CTRL	Bit	Description	Initial State
WrPrivMismatch	[31]	SFR Write Access Privilege Mismatch Status bit. If set to ‘1’, SFR Write Access Privilege Mismatch is occurred.	0b
RdPrivMismatch	[30]	SFR Read Access Privilege Mismatch Status bit. If set to ‘1’, SFR Read Access Privilege Mismatch is occurred.	0b
Reserved	[29:8]	Reserved	0x0000_00
TdesOutReady	[7]	If set to ‘1’, AES Out Buffer is Full, and ARM or Rx FiFo is permitted to Read current 128bits result data	0b
TdesInReady	[6]	If set to ‘1’, TDES Input Buffer is Empty, and ARM or Rx FiFo is permitted to write next 128bits data.	1b
DesOrTdes	[5]	DES or TDES Operation Selection Bit 0 : DES Only Mode 1 : TDES Mode	0b
TdesMode	[4:3]	TDES Mode Selection Bit 01 : ECB Mode 10 : CBC Mode	00b
TdesOpDirection	[2]	TDES Operation Direction Selection Bit. 0 : Encryption 1 : Decryption	0b
TdesIntMode	[1]	TDES Operation End Mode Selection Bit 0 : Polling Mode 1 : Interrupt Mode	0b
TdesOpEnable	[0]	If set to ‘1’, TDES starts operation by ARM. If the des_or_tdes_op_done is generated, It becomes ‘0’.	0b

13.9.2 TDES_RX_KEY1_0

Register	Address	R/W	Description	Reset Value
TDES_Rx_KEY1_0	0x7D20_0010	R/W	TDES Key Input Register 1_0	0x0000_0000
TDES_Rx_KEY1_1	0x7D20_0014	R/W	TDES Key Input Register 1_1	0x0000_0000
TDES_Rx_KEY2_0	0x7D20_0018	R/W	TDES Key Input Register 2_0	0x0000_0000
TDES_Rx_KEY2_1	0x7D20_001C	R/W	TDES Key Input Register 2_1	0x0000_0000
TDES_Rx_KEY3_0	0x7D20_0020	R/W	TDES Key Input Register 3_0	0x0000_0000
TDES_Rx_KEY3_1	0x7D20_0024	R/W	TDES Key Input Register 3_1	0x0000_0000

TDES_Rx_KEY	Bit	Description	Initial State
TdesKey	[31:0]	TDES 1st ~ 6th 32-bit Key Input Register	0x0000_0000

13.9.3 TDES_RX_INPUT_0 / TDES_RX_INPUT_1

Register	Address	R/W	Description	Reset Value
TDES_Rx_INPUT_0	0x7D20_0040	R/W	TDES Data Input Register 0 (Least Significant)	0x0000_0000
TDES_Rx_INPUT_1	0x7D20_0044	R/W	TDES Data Input Register 1 (Least Significant)	0x0000_0000

TDES_Rx_INPUT	Bit	Description	Initial State
TdesDinx	[31:0]	TDES 1st and 2nd 32-bit Data Input Register	0x0000_0000

TDES_Rx_OUTPUT_0 / TDES_Tx_OUTPUT_0 / TDES_Rx_OUTPUT_1 / TDES_Tx_OUTPUT_1

Register	Address	R/W	Description	Reset Value
TDES_Rx_OUTPUT_0	0x7D20_0048	R/W	TDES Data Output Register 0 (Least Significant)	0x0000_0000
TDES_Rx_OUTPUT_1	0x7D20_004C	R/W	TDES Data Output Register 1(Least Significant)	0x0000_0000
TDES_Tx_OUTPUT_0	0x7D60_0048	R/W	TDES Data Output Register 0(Least Significant)	0x0000_0000
TDES_Tx_OUTPUT_1	0x7D60_004C	R/W	TDES Data Output Register 1(Least Significant)	0x0000_0000

TDES_Tx_OUTPUT	Bit	Description	Initial State
TdesDout	[31:0]	TDES 1st and 2nd 32-bit Data Output Register	0x0000_0000

13.9.4 TDES_RX_IV_0 / TDES_RX_IV_1

Register	Address	R/W	Description	Reset Value
TDES_Rx_IV_0	0x7D20_0050	R/W	TDES IV Input Register 0	0x0000_0000
TDES_Rx_IV_1	0x7D20_0054	R/W	TDES IV Input Register 1	0x0000_0000

TDES_Rx_IV	Bit	Description	Initial State
TdesIv	[31:0]	TDES 1st and 2nd 32-bit IV Input Register	0x0000_0000

13.10 SECURITY SUB-SYSTEM SHA-1/PRNG MODULE

13.10.1 HASH_CONTROL

Register	Address	R/W	Description	Reset Value
HASH_CONTROL	0x7D30_0000	R/W	Hash engine control register	0x0000_0000

HASH_CONTROL	Bit	Description	Initial State
Reserved	[31:9]	Reserved	0x0000_00
USE_IV	[8]	Use arbitrary IV instead of SHA-1 constants 0: constants 1: arbitary IV,	0b
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)	00b
SEED_SETTING_ENABLE	[5]	Seed setting enable (1'b1)	0b
Hash_input_finished	[4]	Finished the hash input (will be cleared by hardware)	0b
Hash_start	[3]	Start the hash (software reset) (will be automatically cleared by hardware)	0b
Data_Selection	[2]	Indicates whether the next data of register values are keys or text 0: text 1: key	0b
Engine_Selection	[1:0]	To use as SHA-1 or HMAC or PRNG 00: HMAC 01: SHA1 10: PRNG 11 = reserved	00b

13.10.2 HASH_DATA

Register	Address	R/W	Description	Reset Value
HASH_DATA	0x7D30_0004	R/W	Hash data	0x0000_0000

HASH_DATA	Bit	Description	Initial State
HASH_DATA	[31:0]	Hash data input register (HASH_CONTROL[5] == 1'b0)	0x0000_0000

13.10.3 SEED_DATA_01 ~ SEED_DATA_10

Register	Address	R/W	Description	Reset Value
SEED_DATA_01	0x7D30_0008	R/W	PRNG seed data 1 ([31:0])	0x0000_0000
SEED_DATA_02	0x7D30_000C	R/W	PRNG seed data 2 [63:32]	0x0000_0000
SEED_DATA_03	0x7D30_0010	R/W	PRNG seed data 3 [95:64]	0x0000_0000
SEED_DATA_04	0x7D30_0014	R/W	PRNG seed data 4 [127:96]	0x0000_0000
SEED_DATA_05	0x7D30_0018	R/W	PRNG seed data 5 [159:128]	0x0000_0000
SEED_DATA_06	0x7D30_001C	R/W	PRNG seed data 6 [191:160]	0x0000_0000
SEED_DATA_07	0x7D30_0020	R/W	PRNG seed data 7 [223:192]	0x0000_0000
SEED_DATA_08	0x7D30_0024	R/W	PRNG seed data 8 [255:224]	0x0000_0000
SEED_DATA_09	0x7D30_0028	R/W	PRNG seed data 9 [287:256]	0x0000_0000
SEED_DATA_10	0x7D30_002C	R/W	PRNG seed data 10[319:288]	0x0000_0000

SEED_DATA	Bit	Description	Initial State
SEED_DATA	[31:0]	PRNG seed data (HASH_CONTROL[5] == 1'b1)	0x0000_0000

13.10.4 HASH_STATUS

Register	Address	R/W	Description	Reset Value
HASH_STATUS	0x7D30_0030	R	Hash status	0x0000_0010

HASH_STATUS	Bit	Description	Initial State
Reserved	[31:5]	Reserved	0x0000_00
BUFFER_IN_ENABLE	[4]	1 : Buffer Input Enable (Buffer is empty) 0 : Buffer Input Not Enable (Buffer is full)	1b
HASH_engine_ready	[3]	Ready to receive next 64 bytes of input data (will be cleared by software after checking)	0b
Random_Number_Ready	[2]	Random number ready (will be cleared by hardware after '1' is read)	0b
32bit_ready	[1]	Used only in test Next 32-bit is ready to be processed (will be cleared by hardware after '1' is read)	0b
HASH_output_ready	[0]	160 bits of the hash calculation is finished and ready to be used (will be cleared by hardware after '1' is read)	0b

13.10.5 HASH_OUTPUT_01 (PRNG_OUTPUT_01) ~ HASH_OUTPUT_10 (PRNG_OUTPUT_10)

Register	Address	R/W	Description	Reset Value
HASH_OUTPUT_01	0x7D30_0034	R	Hash output (01) or PRNG output [31:1]	0x0000_0000
HASH_OUTPUT_02	0x7D30_0038	R	Hash output (02) or PRNG output [63:32]	0x0000_0000
HASH_OUTPUT_03	0x7D30_003C	R	Hash output (03) or PRNG output [95:64]	0x0000_0000
HASH_OUTPUT_04	0x7D30_0040	R	Hash output (04) or PRNG output [127:96]	0x0000_0000
HASH_OUTPUT_05	0x7D30_0044	R	Hash output (05) or PRNG output [159:128]	0x0000_0000
HASH_OUTPUT_06	0x7D30_0048	R	Hash output (06) or PRNG output [191:160]	0x0000_0000
HASH_OUTPUT_07	0x7D30_004C	R	Hash output (07) or PRNG output [223:192]	0x0000_0000
HASH_OUTPUT_08	0x7D30_0050	R	Hash output (08) or PRNG output [255:224]	0x0000_0000
HASH_OUTPUT_09	0x7D30_0054	R	Hash output (09) or PRNG output [287:256]	0x0000_0000
HASH_OUTPUT_10	0x7D30_0058	R	Hash output (10) or PRNG output [319:288]	0x0000_0000
HASH_OUTPUT_01	0x7D70_0034	R	Hash output (01) or PRNG output [31:1]	0x0000_0000
HASH_OUTPUT_02	0x7D70_0038	R	Hash output (02) or PRNG output [63:32]	0x0000_0000
HASH_OUTPUT_03	0x7D70_003C	R	Hash output (03) or PRNG output [95:64]	0x0000_0000
HASH_OUTPUT_04	0x7D70_0040	R	Hash output (04) or PRNG output [127:96]	0x0000_0000
HASH_OUTPUT_05	0x7D70_0044	R	Hash output (05) or PRNG output [159:128]	0x0000_0000
HASH_OUTPUT_06	0x7D70_0048	R	Hash output (06) or PRNG output [191:160]	0x0000_0000
HASH_OUTPUT_07	0x7D70_004C	R	Hash output (07) or PRNG output [223:192]	0x0000_0000
HASH_OUTPUT_08	0x7D70_0050	R	Hash output (08) or PRNG output [255:224]	0x0000_0000
HASH_OUTPUT_09	0x7D70_0054	R	Hash output (09) or PRNG output [287:256]	0x0000_0000
HASH_OUTPUT_10	0x7D70_0058	R	Hash output (10) or PRNG output [319:288]	0x0000_0000

HASH_OUTPUT	Bit	Description	Initial State
HASH_OUTPUT	[31:0]	PRNG_output when Engine_selection[1:0] == 2'b10 Else Hash_output[31:0] for Rx/Tx	0x0000_0000

13.10.6 HASH_MIDOUT_01 ~ HASH_MIDOUT_05

Register	Address	R/W	Description	Reset Value
HASH_MIDOUT_01	0x7D30_005C	R	HASH_MIDOUT[159:128]	0x0000_0000
HASH_MIDOUT_02	0x7D30_0060	R	HASH_MIDOUT[127:96]	0x0000_0000
HASH_MIDOUT_03	0x7D30_0064	R	HASH_MIDOUT[95:64]	0x0000_0000
HASH_MIDOUT_04	0x7D30_0068	R	HASH_MIDOUT[63:32]	0x0000_0000
HASH_MIDOUT_05	0x7D30_006C	R	HASH_MIDOUT[31:0]	0x0000_0000
HASH_MIDOUT_01	0x7D70_005C	R	HASH_MIDOUT[159:128]	0x0000_0000
HASH_MIDOUT_02	0x7D70_0060	R	HASH_MIDOUT[127:96]	0x0000_0000
HASH_MIDOUT_03	0x7D70_0064	R	HASH_MIDOUT[95:64]	0x0000_0000
HASH_MIDOUT_04	0x7D70_0068	R	HASH_MIDOUT[63:32]	0x0000_0000
HASH_MIDOUT_05	0x7D70_006C	R	HASH_MIDOUT[31:0]	0x0000_0000

HASH_MIDOUT	Bit	Description	Initial State
HASH_MIDOUT	[31:0]	HASH_MIDOUT	0x0000_0000

HASH_IV_01 ~ HASH_IV_05

Register	Address	R/W	Description	Reset Value
HASH_IV_01	0x7D30_0070	W	HASH_IV [159:128]	0x0000_0000
HASH_IV_02	0x7D30_0074	W	HASH_IV [127:96]	0x0000_0000
HASH_IV_03	0x7D30_0078	W	HASH_IV [95:64]	0x0000_0000
HASH_IV_04	0x7D30_007C	W	HASH_IV [63:32]	0x0000_0000
HASH_IV_05	0x7D30_0080	W	HASH_IV [31:0]	0x0000_0000
HASH_IV_01	0x7D70_0070	W	HASH_IV [159:128]	0x0000_0000
HASH_IV_02	0x7D70_0074	W	HASH_IV [127:96]	0x0000_0000
HASH_IV_03	0x7D70_0078	W	HASH_IV [95:64]	0x0000_0000
HASH_IV_04	0x7D70_007C	W	HASH_IV [63:32]	0x0000_0000
HASH_IV_05	0x7D70_0080	W	HASH_IV [31:0]	0x0000_0000

HASH_IV	Bit	Description	Initial State
HASH_IV	[31:0]	HASH_IV	0x0000_0000

13.10.7 PRE_MSG_LENGTH_01 / PRE_MSG_LENGTH_02

Register	Address	R/W	Description	Reset Value
PRE_MSG_LENGTH_01	0x7D70_0084	R/W	PRE_MSG_LENGTH [63:32]	0x0000_0000
PRE_MSG_LENGTH_02	0x7D70_0088	R/W	PRE_MSG_LENGTH [31:0]	0x0000_0000

PRE_MSG_LENGTH	Bit	Description	Initial State
PRE_MSG_LENGTH	[31:0]	PRE_MSG_LENGTH	0x0000_0000

14 DISPLAY CONTROLLER

14.1 OVERVIEW

The Overlay/DISPLAY controller includes 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, I80 Interface and NTSC/PAL standard TV Encoder Interface and IT-R BT. 601 interface. The DISPLAY controller supports up to 5 overlay image windows. Overlay image windows support various color format, 16 level alpha blending, color key, x-y position control, soft scrolling, variable window size, and etc.

The DISPLAY controller supports various color formats such as RGB (1BPP to 24 BPP), and YCbCr 4:4:4(only local bus).

The DISPLAY controller can be programmed to support the different requirements on the screen. Requirements related to the number of horizontal and vertical pixels, data line width for the data interface, interface timing, and refresh rate.

The DISPLAY controller is used to transfer the video data and to generate the necessary control signals such as, RGB_VSYNC, RGB_HSYNC, RGB_VCLK, RGB_VDEN, and SYS_CS0(as well as the control signals). DISPLAY controller has the data ports for video data, which are RGB_VD[23:0], SYS_VD[17:0], and TV_OUT as shown in Figure 14-1.

14.2 BLOCK DIAGRAM

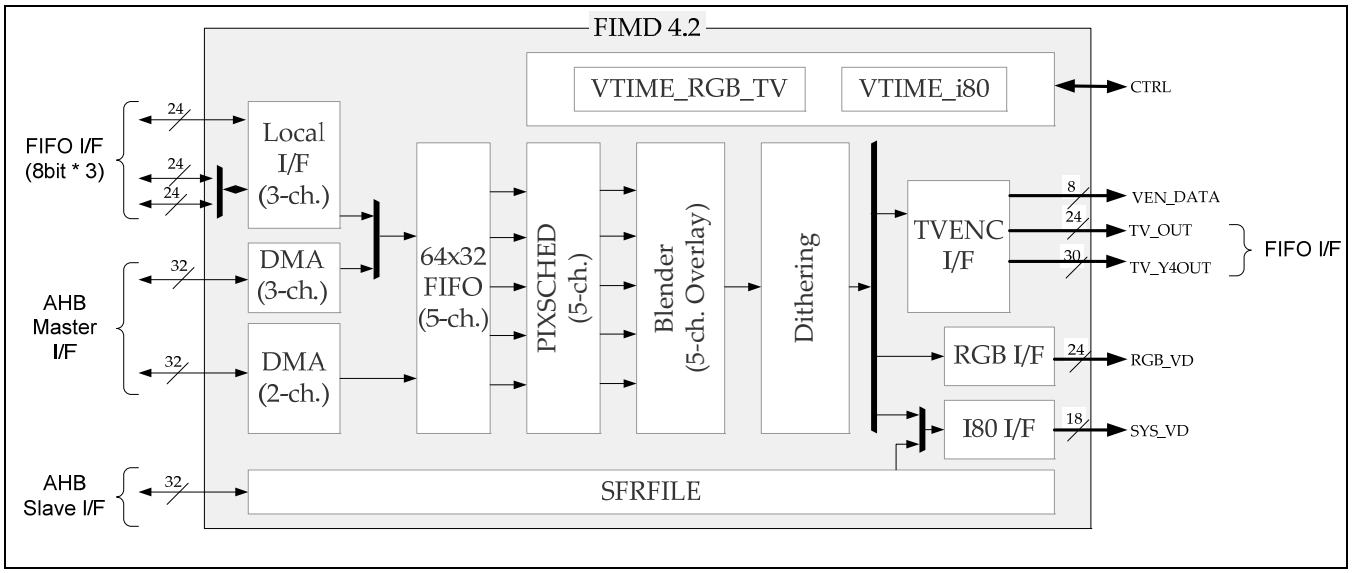


Figure 14-1. Top block diagram DISPLAY controller

14.3 FEATURES

The FIMD supports the following:

Video Output Interface	RGB IF (Parallel) RGB IF (Serial) I80 Interface TV Encoder Interface (NTSC, PAL standard) ITU-R BT.601 interface (YUV422 8bit)
PIP (OSD) function	Supports 8-BPP (bit per pixel) palletized color Supports 16-BPP non-palletized color Supports unpacked 18-BPP non-palletized color Supports unpacked 24-BPP non-palletized color Supports X, Y indexed position Supports 4 bit Alpha blending: Plane / Pixel(only supports 24-bit 8:8:8 mode)
Source format	Window 0 Supports 1, 2, 4 or 8-BPP (bit per pixel) palletized color Supports 16, 18 or 24-BPP non-palletized color Supports YCbCr (4:4:4) local input from Local Bus (Post Processor) Supports RGB (8:8:8) local input from Local Bus (Post Processor) Window 1 Supports 1, 2, 4 or 8-BPP (bit per pixel) palletized color Supports 16, 18 or 24-BPP non-palletized color Supports YCbCr (4:4:4) local input from Local Bus (TV Scaler) Supports RGB (8:8:8) local input from Local Bus (TV Scaler) Window 2 Supports 1, 2, or 4-BPP (bit per pixel) palletized color Supports 16, 18 or 24-BPP non-palletized color Supports YCbCr (4:4:4) local input from Local Bus (TV Scaler) Supports RGB (8:8:8) local input from Local Bus (TV Scaler) Window 3 Supports 1, 2 or 4-BPP (bit per pixel) palletized color Supports 16, 18 or 24-BPP non-palletized color Window 4 Supports 1 or 2-BPP (bit per pixel) palletized color Supports 16, 18 or 24-BPP non-palletized color
Configurable Burst Length	Programmable 4 / 8 / 16 Burst DMA
Palette/Look-up table	256 x 25 bits palette (2ea: One for the Window 0, the other for the Window1) 16(entry) x 16 bits Look-up table for Window 2 16(entry) x 16 bits Look-up table for Window 3 4(entry) x 16 bits Look-up table for Window 4
Soft Scrolling	Horizontal : 1 Byte resolution Vertical : 1 pixel resolution
Virtual Screen	Virtual image can have up to 16 Mbyte image size.
Transparent Overlay	Supports Transparent Overlay
Color Key (Chroma Key)	Supports Color key function
Partial Display	Supports LCD partial display function through I80 interface

14.4 FUNCTIONAL DESCRIPTION

14.4.1 BRIEF OF THE SUB-BLOCK

The DISPLAY controller consists of a VSFR, VDMA, VPRCS, VTIME, and video clock generator. The VSFR includes programmable register sets and two-256x 25 palette memories. These are used to configure the DISPLAY controller. The VDMA is a dedicated display DMA. It can transfer the video data in frame memory to VPRCS. By using this special DMA, the video data can be displayed on the screen without CPU intervention. The VPRCS receives the video data from VDMA and sends the video data through the data ports to the display device (LCD) after converting 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, SYS_CS1, SYS_CS0, etc.

14.4.2 DATA FLOW

FIFO is present in the VDMA. When FIFO is empty or partially empty, VDMA requests a data fetching from a frame memory with the burst memory transfer mode (Consecutive memory fetching of 4 / 8 / 16 words per one burst request without allowing the bus mastership to another bus master during the bus transfer). When bus arbitrator in the memory controller accepts this kind of transfer request, there will be 4 /8 /16 successive word data transfers from system memory to internal FIFO. The each size of FIFO is 64 words. The size of FIFO is determined by the data transfer rate. The DISPLAY controller has five FIFOs because it needs to support the overlay window display mode. In case of one screen display mode, the only one FIFO must be used. The data through FIFO is fetched by VPRCS, which has a blending, scheduling function for the final image data. VPRCS supports overlay function, which enables to overlay any image up to 5 window images. Irrespective of the size it can be blended with main window image with programmable alpha blending or color (chroma) key function. Figure 14-2 shows the data flow from system bus to the output buffer. VDMA has 5 DMA channels and 3 Local Input I/F. CSC (Color Space Conversion) block changes YCbCr data to RGB data for the blending operation. Alpha values written in SFR determine the level of blending. Data from Output buffer will be appearing to the Video Data Port.

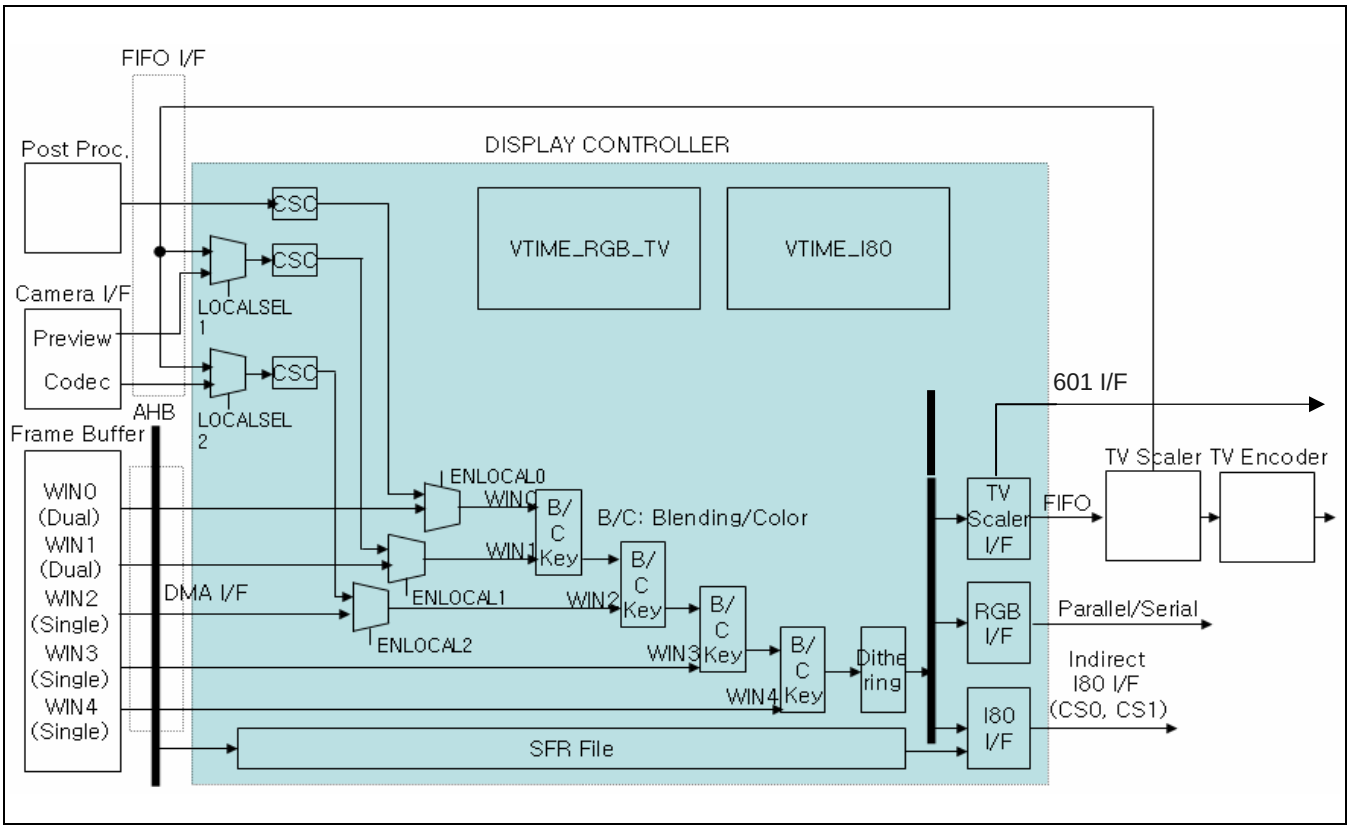


Figure 14-2. Block diagram of the Data Flow

14.4.3 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 I80 Interface which uses address, data, chip select, read/write control and register/status indicating signal. In this type of LCD driver it has a frame buffer and has the function of self-refresh, therefore DISPLAY controller updates one still image by writing only one time to the LCD. The Third type is ITU-R BT.601 interface. ITU-R BT.601 interface uses YUV data, Vertical/horizontal sync, optional Field signal, data valid signal and data sync clock. The fourth type is FIFO interface with TV Encoder.

14.4.4 OVERVIEW OF THE COLOR DATA

14.4.4.1 RGB Data format

The DISPLAY controller requests the specified memory format of frame buffer. The next table shows some examples of each display mode.

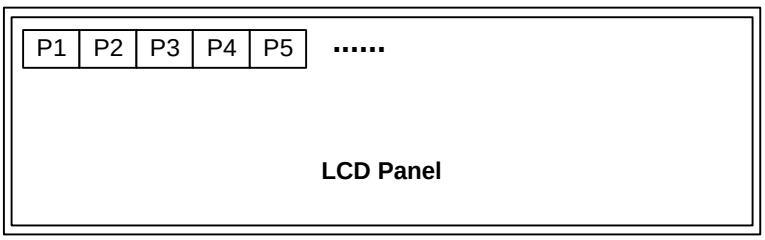
28BPP display (A4+*888*)

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23:0]
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
...		

(BSWP = 0, HWSWP = 0, BLD_PIX = 1, ALPHA_SEL = 1)

	D[31:28]	D[27:24]	D[23:0]
000H	Dummy Bit	Alpha value	P1
004H	Dummy Bit	Alpha value	P2
008H	Dummy Bit	Alpha value	P3
...			



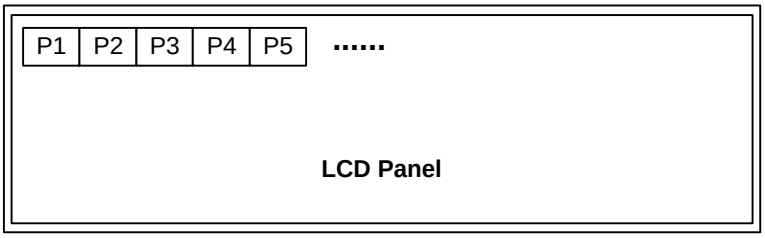
NOTE: D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

In case of BLD_PIX and ALPHA_SEL are set,
D[27:24] = Alpha value, D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

25BPP display (A888)

(BSWP = 0, HWSWP = 0)

	D[31:25]	D[24]	D[23:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
...			



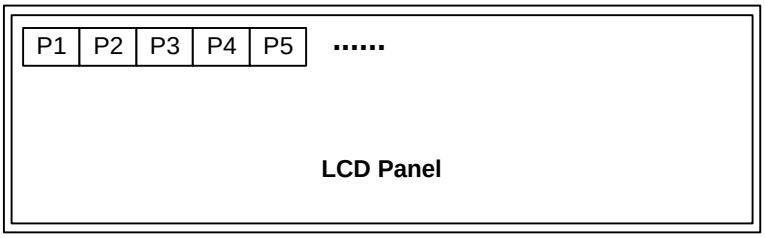
NOTES:

1. AEN : Transparency value selection bit
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended 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 description of SFR.
2. D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

24BPP display (A887)

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
...			



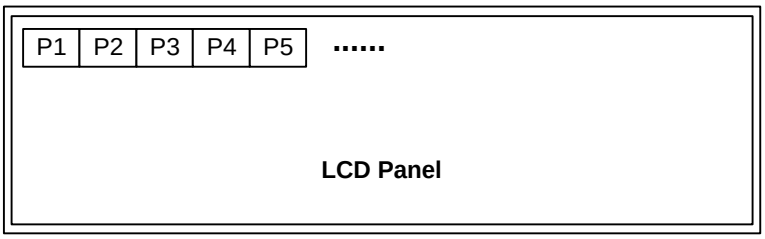
NOTES:

1. AEN : Transparency value selection bit
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended 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 description of SFR.
2. D[22:15] = Red data, D[14:7] = Green data, D[6:0] = Blue data

24BPP display (888)

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23:0]
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
...		

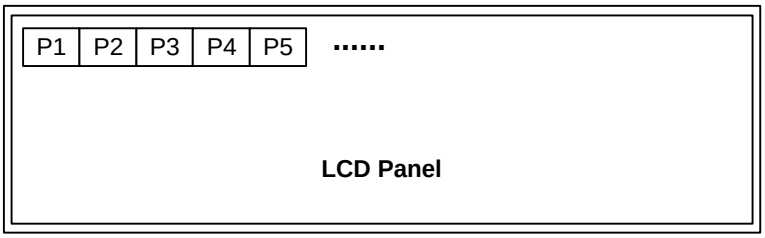


NOTE: 1. D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

19BPP display (A666)

(BSWP = 0, HWSWP = 0)

	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
...			



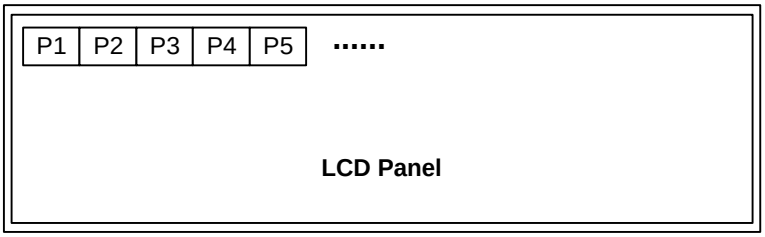
NOTES:

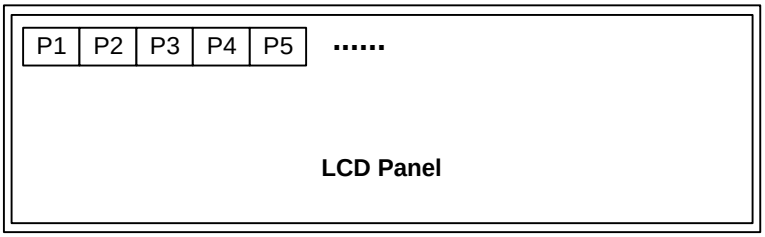
1. AEN : Transparency value selection bit
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended with alpha value selected by AEN.
   Alpha value is selected by SFR value as ALPHA0_R, ALPHA0_G, ALPHA0_B, ALPHA1_R, ALPHA1_G, ALPHA1_B. For more information refer to description of SFR.
2. D[17:12] = Red data, D[11:6] = Green data, D[5:0] = Blue data

18BPP display (666)

(BSWP = 0, HWSWP = 0)

	D[31:18]	D[17:0]
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
...		





NOTE: 1. D[17:12] = Red data, D[11:6] = Green data, D[5:0] = Blue data

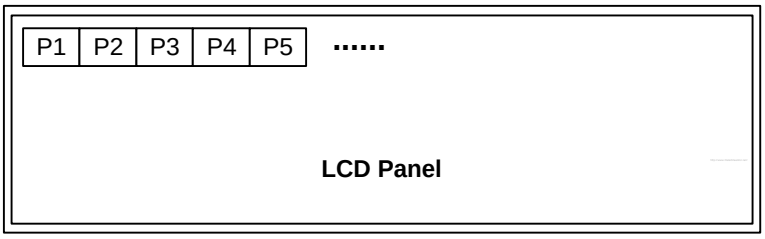
16BPP display (A555)

(BSWP = 0, HWSWP = 0)

	D[31]	D[30:16]	D[15]	D[14:0]
000H	AEN1	P1	AEN2	P2
004H	AEN3	P3	AEN4	P4
008H	AEN5	P5	AEN6	P6
...				

(BSWP = 0, HWSWP = 1)

	[31]	D[30:16]	D[15]	D[14:0]
000H	AEN2	P2	AEN1	P1
004H	AEN4	P4	AEN3	P3
008H	AEN6	P6	AEN5	P5
...				



NOTES:

1. AEN : Transparency value selection bit
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended 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 description of SFR.
2. D[14:10] = Red data, D[9:5] = Green data, D[4:0] = Blue data

16BPP display (1+555)

(BSWP = 0, HWSWP = 0)

	D[31:16]	D[15:0]
000H	P1	P2
004H	P3	P4
008H	P5	P6
...		

(BSWP = 0, HWSWP = 1)

	D[31:16]	D[15:0]
000H	P2	P1
004H	P4	P3
008H	P6	P5
...		

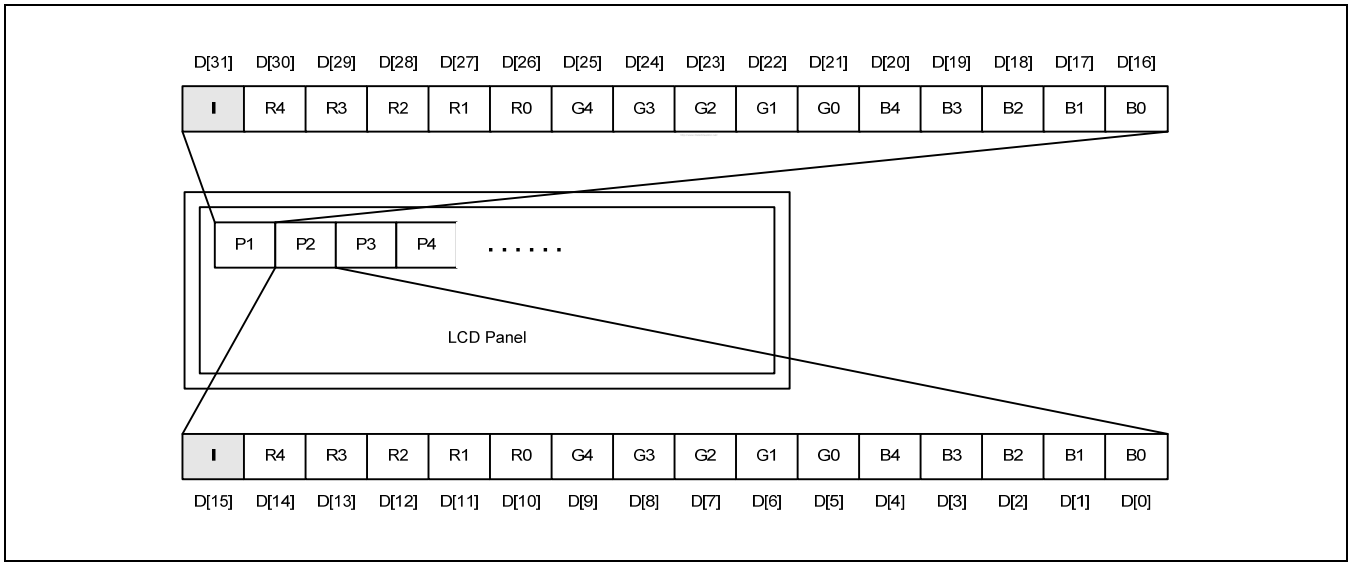


Figure 14-3. 16BPP(1+5:5:5) Display Types

NOTE: 1. {D[14:10], D[15] } = Red data, {D[9:5], D[15] } = Green data, {D[4:0], D[15]}= Blue data

16BPP display (565)

(BSWP = 0, HWSWP = 0)

	D[31:16]	D[15:0]
000H	P1	P2
004H	P3	P4
008H	P5	P6
...		

(BSWP = 0, HWSWP = 1)

	D[31:16]	D[15:0]
000H	P2	P1
004H	P4	P3
008H	P6	P5
...		

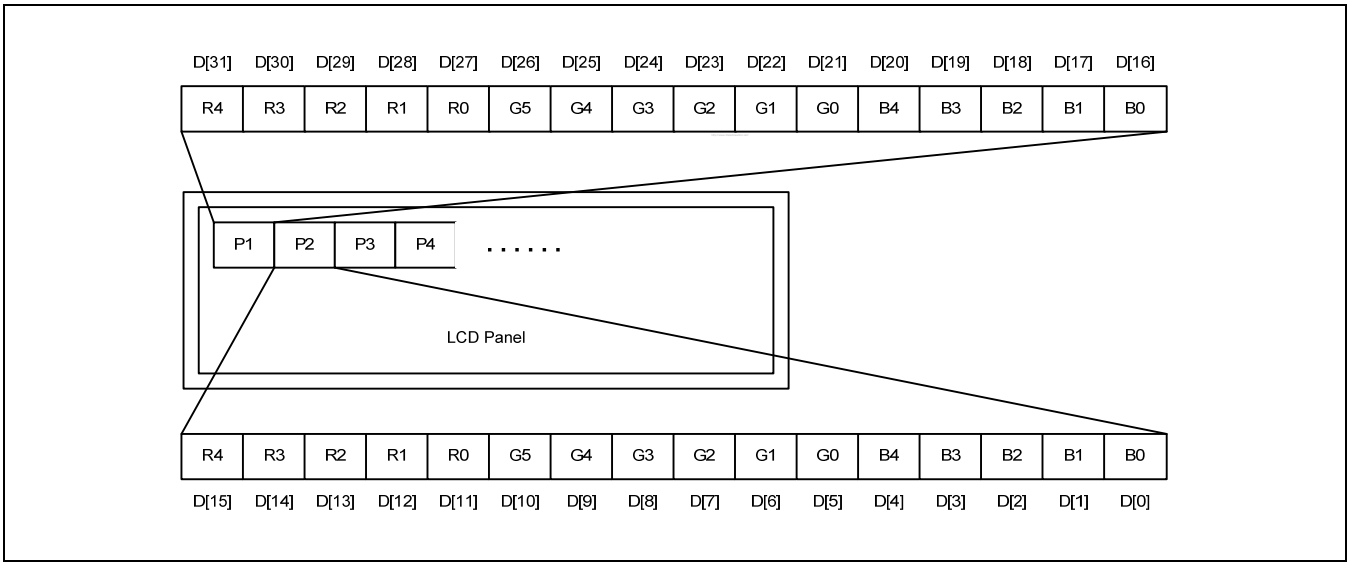


Figure 14-4. 16BPP(5:6:5) Display Types

NOTE: 1. D[15:11] = Red data, D[10:5] = Green data, D[4:0] = Blue data

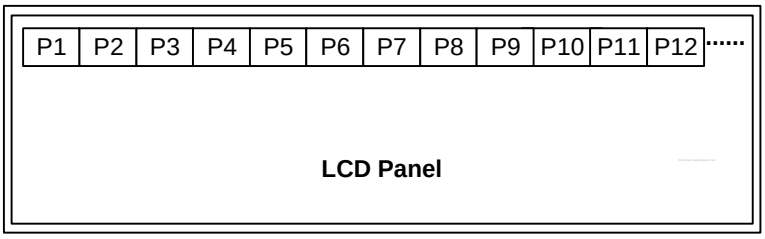
8BPP display (Palette)

(BSWP = 0, HWSWP = 0)

	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P1	P2	P3	P4
004H	P5	P6	P7	P8
008H	P9	P10	P11	P12
...				

(BSWP = 1, HWSWP = 0)

	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P4	P3	P2	P1
004H	P8	P7	P6	P5
008H	P12	P11	P10	P9
...				



NOTE:

1. The values of frame buffer are index of palette memory.
   The MSB value of Palette memory is AEN bit.
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended 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 description of SFR.

4BPP display (Palette)

(BSWP = 0, HWSWP = 0)

	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	P1	P2	P3	P4	P5	P6	P7	P8
004H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P17	P18	P19	P20	P21	P22	P23	P24
...								

(BSWP = 1, HWSWP = 0)

	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
004H	P15	P16	P13	P14	P11	P12	P9	P10
008H	P23	P24	P21	P22	P19	P20	P17	P18
...								

NOTE:

1. The values of frame buffer are index of palette memory.
   The MSB value of Palette memory is AEN bit.
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended 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 description of SFR.

2BPP display(Palette)

(BSWP = 0, HWSWP = 0)

D	[31:30]	[29:28]	[27:26]	[25:24]	[23:22]	[21:20]	[19:18]	[17:16]
000H	P1	P2	P3	P4	P5	P6	P7	P8
004H	P17	P18	P19	P20	P21	P22	P23	P24
008H	P33	P34	P35	P36	P37	P38	P39	P40
...								

D	[15:14]	[13:12]	[11:10]	[9:8]	[7:6]	[5:4]	[3:2]	[1:0]
000H	P9	P10	P11	P12	P13	P14	P15	P16
004H	P25	P26	P27	P28	P29	P30	P31	P32
008H	P41	P42	P43	P44	P45	P46	P47	P48
...								

NOTE:

1. The values of frame buffer are index of palette memory.
   The MSB value of Palette memory is AEN bit.
   AEN = 0 : ALPHA0_R/G/B values are applied
   AEN = 1 : ALPHA1_R/G/B values are applied
   If per-pixel blending is set, then this pixel would be blended 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 description of SFR.

1BPP display (*Palette*)

(BSWP = 0, HWSWP = 0)

	[31]	[30]	[29]	[28]	[27]	[26]	[25]	[24]	[23]	[22]	[21]	[20]	[19]	[18]	[17]	[16]
002H	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
006H	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48
…																
	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
000H	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32
004H	P49	P50	P51	P52	P53	P54	P55	P56	P57	P58	P59	P60	P61	P62	P63	P64
…																

14.4.5 PALETTE USAGE

14.4.5.1 Palette Configuration and Format Control

The DISPLAY controller can support the 256 colors palette for various selection of color mapping.

The user can select 256 colors from the 25-bit colors through these four formats.
256 color palette consist of the 256(depth) × 25-bit DPSRAM. Palette supports 8:8:8, 6:6:6, 5:6:5(R:G:B), and etc formats.
For example of A:5:5:5 format, write palette as specified in Table 14-2 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 Window Palette Control (W0PAL, case window0) register to 0'b101.

Table 14-1. 25BPP(A:8:8:8) Palette Data Format

INDEX\ Bit Pos.	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 14-2. 16BPP(A:5:5:5) Palette Data Format

INDEX\ Bit Pos.	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

.

14.4.6 WINDOW BLENDING

14.4.6.1 Overview

The main function of the VPRCS module is window blending. DISPLAY controller has 5 window layers and the details are described below. For example, System can use window0 as an OS window, full TV screen window or etc., window1 as a small (next channel) TV screen with window2 as a menu, window3 as a caption, window4 as a channel information. Window2, window3 and window4 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 windows (example)

Window 0 (base) : Local / (YCbCr ,RGB without palette)

Window 1 (Overlay1) : RGB with palette

Window 2 (Overlay2) : RGB with palette

Window 3 (Caption) : RGB (1/2/4) with 16 level Color LUT

Window 4 (Cursor) : RGB (1/2) with 4 level Color LUT

Overlay Priority

Window 4 > Window 3 > Window 2 > Window 1 > Window 0

Color Key

:The register value to Color-Key reg must be set by 24bit RGB format

Blending equation

Window 01(R) = Window 0(R) x ***Beta1*** + Window 1(R) x ***Alpha1***

Window 01(G) = Window 0(G) x ***Beta1*** + Window 1(G) x ***Alpha1***

Window 01(B) = Window 0(B) x ***Beta1***+ Window 1(B) x ***Alpha1***

Window 012(R) = Window 01(R) x ***Beta2*** + Window 2(R) x ***Alpha2***

Window 012(G) = Window 01(G) x ***Beta2*** + Window 2(G) x ***Alpha2***

Window 012(B) = Window 01(B) x ***Beta2*** + Window 2(B) x ***Alpha2***

Window 0123(R) = Window 012(R) x ***Beta3*** + Window 3(R) x ***Alpha3***

Window 0123(G) = Window 012(G) x ***Beta3*** + Window 3(G) x ***Alpha3***

Window 0123(B) = Window 012(B) x ***Beta3*** + Window 3(B) x ***Alpha3***

WinOut(R) = Window 0123(R) x ***Beta4*** + Window 4(R) x ***Alpha4***

WinOut(G) = Window 0123(G) x ***Beta4*** + Window 4(G) x ***Alpha4***

WinOut(B) = Window 0123(B) x ***Beta4*** + Window 4(B) x ***Alpha4***

Where,

if A bit is set then

AR1 = Window 1's Red blending factor (ALPHA1_R@VIDOSD1C)

AR2 = Window 2's Red blending factor (ALPHA1_R@VIDOSD2C)

AR3 = Window 3's Red blending factor (ALPHA1_R@VIDOSD3C),

AR4 = Window 4's Red blending factor (ALPHA1_R@VIDOSD4C),

AG1 = Window 1's Green blending factor (ALPHA1_G@VIDOSD1C)

AG2 = Window 2's Green blending factor (ALPHA1_G@VIDOSD2C)

AG3 = Window 3's Green blending factor (ALPHA1_G@VIDOSD3C),

AG4 = Window 4's Green blending factor (ALPHA1_G@VIDOSD4C),

AB1 = Window 1's Blue blending factor (ALPHA1_B@VIDOSD1C)

AB2 = Window 2's Blue blending factor (ALPHA1_B@VIDOSD2C)

AB3 = Window 3's Blue blending factor (ALPHA1_B@VIDOSD3C),

AB4 = Window 4's Blue blending factor (ALPHA1_B@VIDOSD4C),

Else if A bit is cleared then

AR1 = Window 1's Red blending factor (ALPHA0_R@VIDOSD1C)

AR2 = Window 2's Red blending factor (ALPHA0_R@VIDOSD2C)

AR3 = Window 3's Red blending factor (ALPHA0_R@VIDOSD3C),

AR4 = Window 4's Red blending factor (ALPHA0_R@VIDOSD4C),

AG1 = Window 1's Green blending factor (ALPHA0_G@VIDOSD1C)

AG2 = Window 2's Green blending factor (ALPHA0_G@VIDOSD2C)

AG3 = Window 3's Green blending factor (ALPHA0_G@VIDOSD3C),

AG4 = Window 4's Green blending factor (ALPHA0_G@VIDOSD4C),

AB1 = Window 1's Blue blending factor (ALPHA0_B@VIDOSD1C)

AB2 = Window 2's Blue blending factor (ALPHA0_B@VIDOSD2C)

AB3 = Window 3's Blue blending factor (ALPHA0_B@VIDOSD3C),

AB4 = Window 4's Blue blending factor (ALPHA0_B@VIDOSD4C),

In case Window 01(R), Alpha and Beta value is determined by

Alphax = ARx/16, ***Betax*** = (15-ARx)/16

if ARx == 0xF then ***Alphax*** =1 and ***Betax*** =0

if ARx == 0x0 then ***Alphax*** =0 and ***Betax*** =1

14.4.6.2 BLENDING DIAGRAM/DETAILS

DISPLAY controller can blend 5 Layer for only one pixel at the same time. The Blending factor, alpha value is controlled by ALPHA0_R,ALPHA0_G,ALPHA0_B, ALPHA1_R,ALPHA1_G,ALPHA1_B register, which are implemented for each window layer and color(R,G,B). The illustration below is described as the example of the R (Red) output using ALPHA_R value of each window. All windows have two kinds of alpha blending value. One is alpha value for transparency enable (AEN value ==1), the other is alpha value for transparency disable (AEN value == 0). If WINEN_F is enabled and BLD_PIX is enabled then AR will be selected by applying 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, the AEN bit information is not being used.

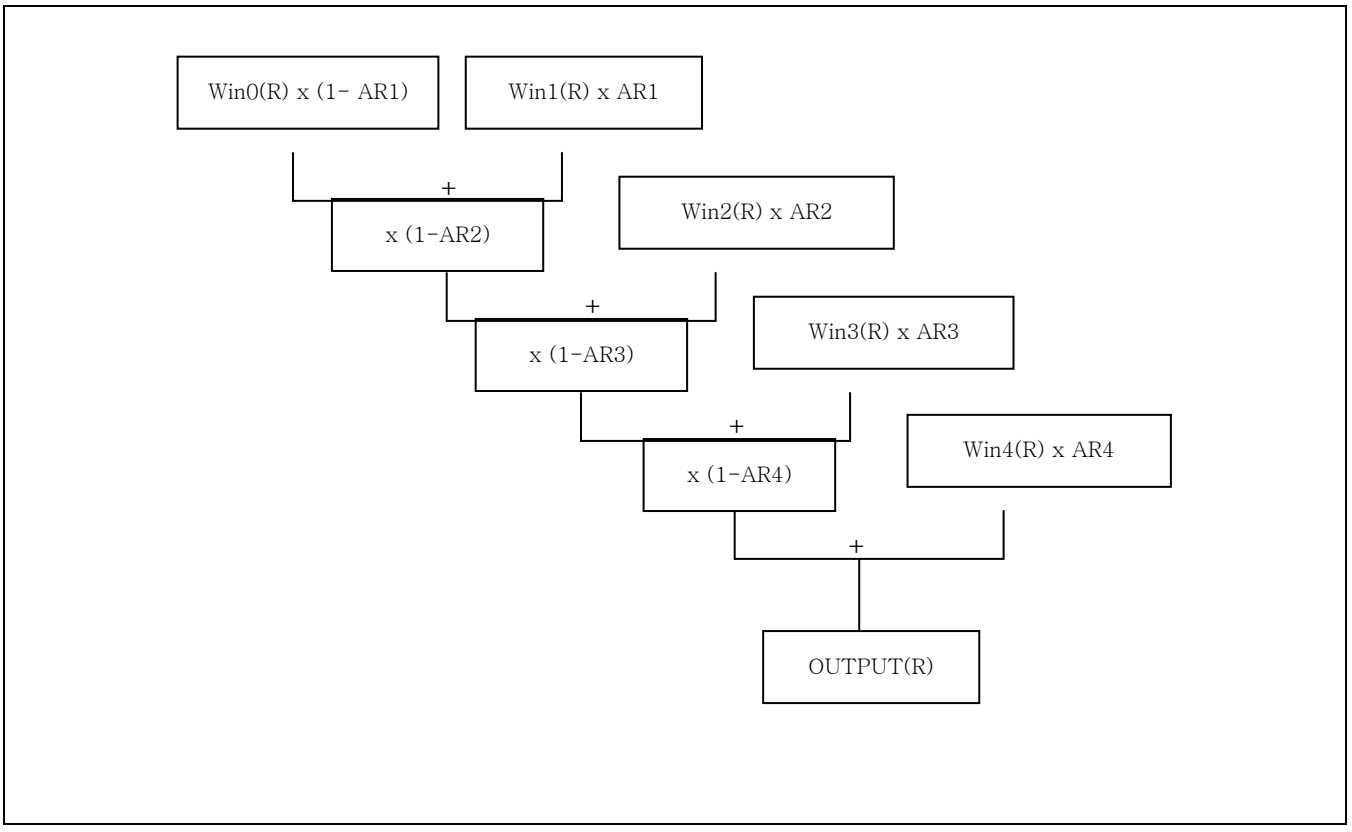


Figure 14-5. Blending diagram

Table 14-3. Blending User's Table

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">ALPHA_SEL[1] value @ WINCON1/2/3/4</th></tr>
<tr><th>'0'</th><th>'1'</th></tr>
<tr><td rowspan="2">BLD_PIX[6] value @WINCON1/2/3/4</td><td>‘0’</td><td>Plane blending using ALPHA0</td><td>Plane blending using ALPAH1</td></tr>
<tr><td>'1'</td><td>Pixel blending selected by AEN
<table>
<tr><th colspan="2">AEN value @ Frame Buffer</th></tr>
<tr><td>'0'</td><td>'1'</td></tr>
<tr><td>Using ALPHA0</td><td>Using ALPHA1</td></tr>
</table>
OR
Color key blending enabled by KEYBLEN
<table>
<tr><th colspan="2">KEYBLEN[26] value @W1/2/3/4KEYCON0</th></tr>
<tr><td>'0'</td><td>'1'</td></tr>
<tr><td>Key blending disable</td><td>Non-key area: using ALPHA0
Key area: using ALPHA1</td></tr>
</table></td><td>Pixel blending using DATA[27:24] in frame buffer, Only when BPPMODE_F[5:2] value @ WINCON1/2/3/4 is ‘b1101’</td></tr>
</table>

14.4.7 COLOR-KEY FUNCTION

The DISPLAY controller supports color-key function for the various effect of image mapping. Color image of OSD layer, which is specified by COLOR-KEY register, will be substituted by background image for special functionality. It will be substituted as cursor image or pre-view image of the camera.

The register value to ColorKey reg must be set by 24bit RGB format.

DIRCON (in Win1 Color Key 0 register) bit selects the window to be compared with COLVAL (in Win1 Color Key 1 Register). If this bit is set to '0', the comparison window is window1 (foreground window).

COMPKEY (in Win1 Color Key 0 register) value decides whether to compare COLVAL and selected window color. In other words, the comparator only compares COLVAL and selected window color bits where the corresponding bit in COMPKEY is '0'.

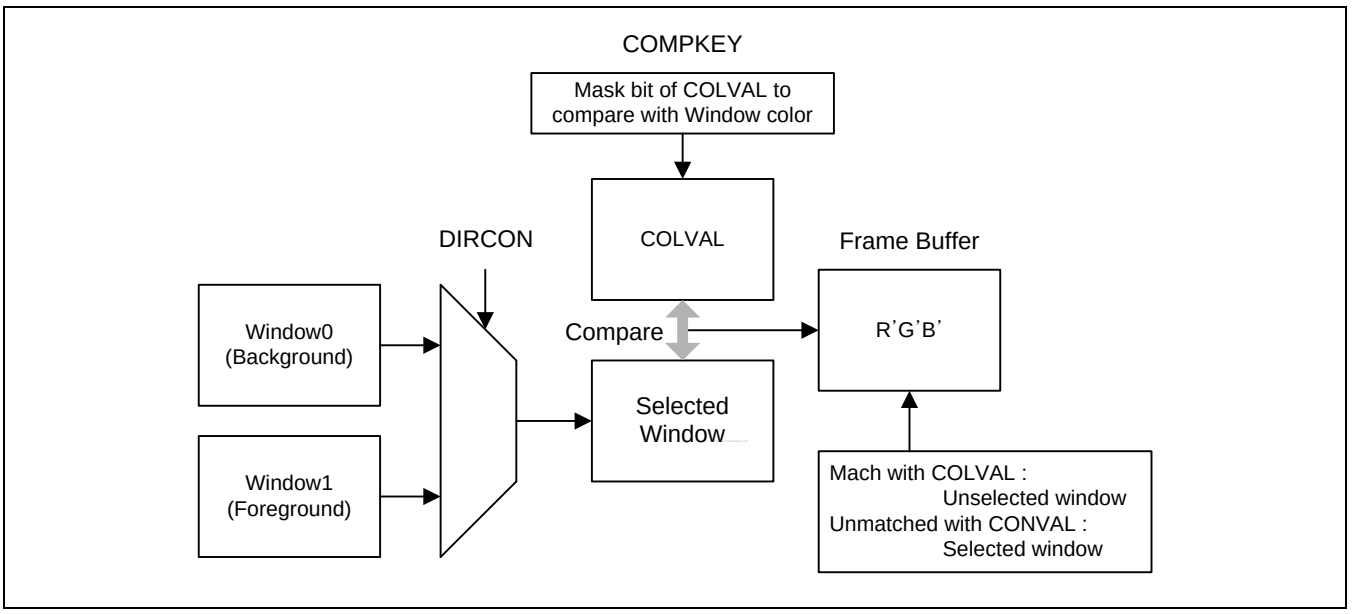


Figure 14-6. Color Key Operation

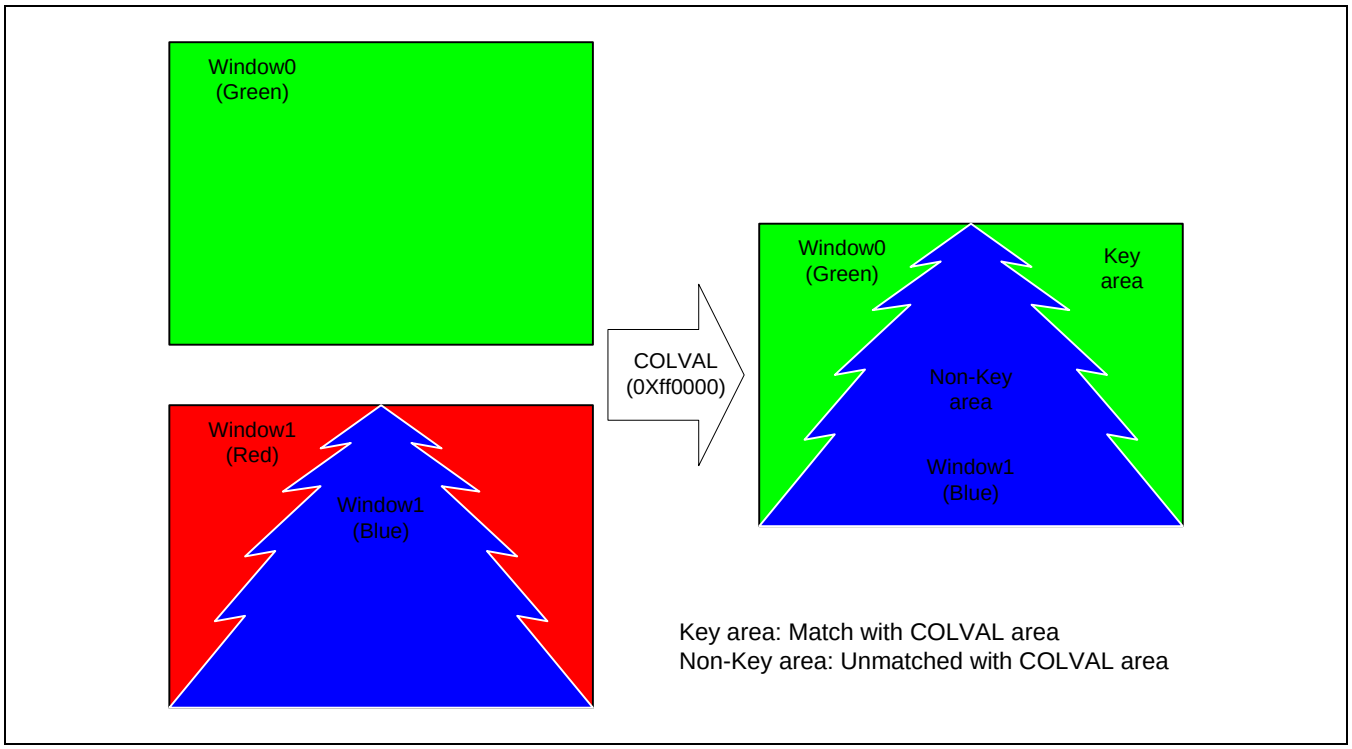


Figure 14-7. Color Key Operations

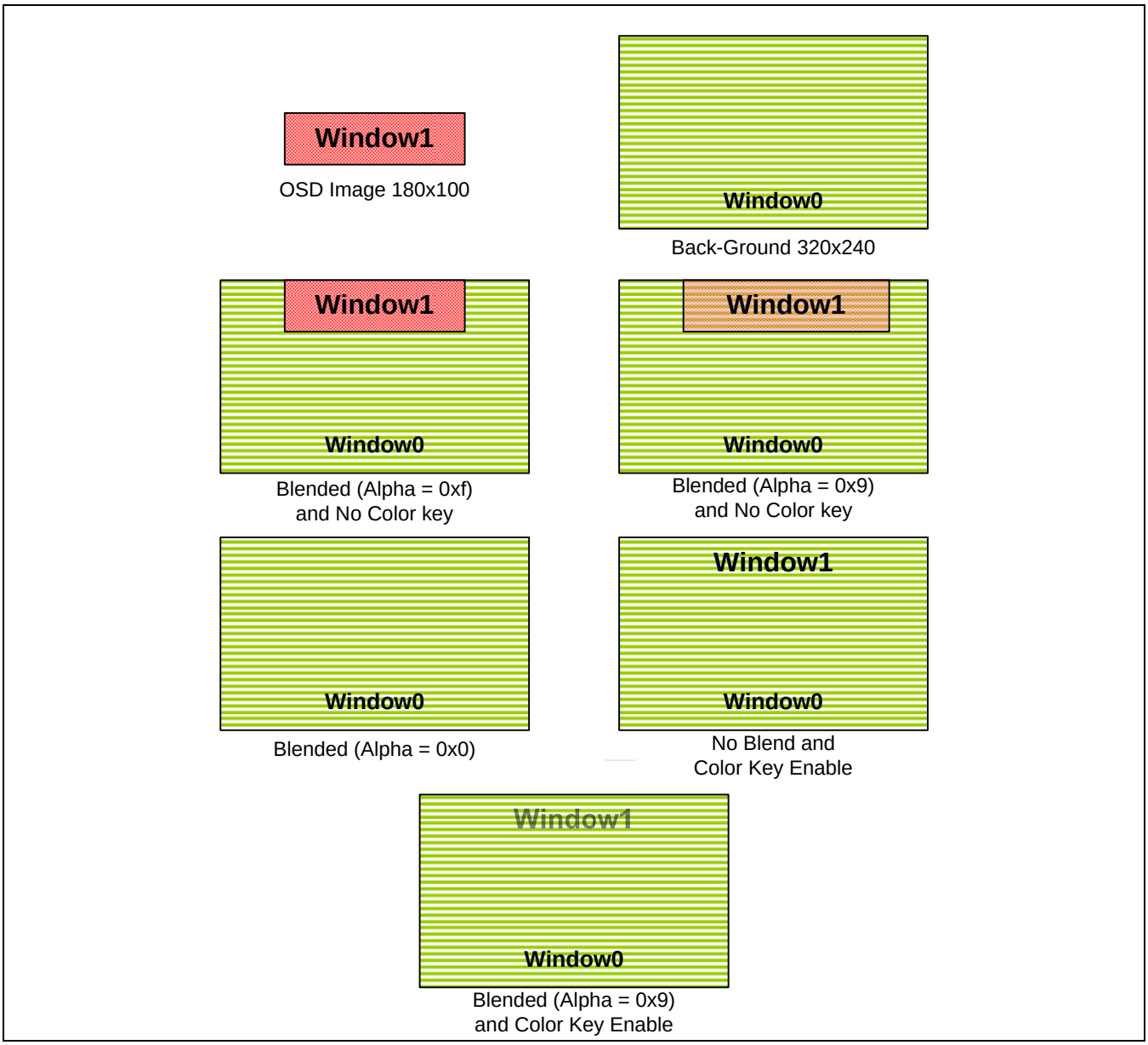


Figure 14-8. Color Key Function Configurations

14.4.8 VTIME CONTROLLER OPERATION

VTIME is mainly divided into two blocks. One is VTIME_RGB_TV for RGB interface, ITU-R601 interface and TV Encoder Interface timing control. The other is for I80 interface timing control.

14.4.8.1 RGB Interface

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 can generate the programmable control signals suitable for the support of many different types of display device.

The RGB_VSYNC signal is asserted to cause the LCD's line pointer to start over at the top of the display. The RGB_VSYNC and RGB_HSYNC pulse generation is controlled by the configuration of both the HOZVAL field and the LINEVAL registers. The HOZVAL and LINEVAL can be determined by the size of the LCD panel according to the following equations:

HOZVAL = (Horizontal display size) -1

LINEVAL = (Vertical display size) –1

The rate of RGB_VCLK signal can be controlled by the CLKVAL field in the VIDCON0 register. 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 14-4. Relation between VCLK and CLKVAL (TFT, Freq. of Video Clock Source=60MHz)

CLKVAL	60MHz/X	VCLK
1	60 MHz/2	30.0MHz
2	60 MHz/3	20.0 MHz
3	60 MHz/4	15.0 MHz
:	:	:
63	60 MHz/64	937.5 kHz

The RGB_HSYNC and RGB_VSYNC signal is configured by VSYNC, VBPD, VFPD, HSYNC, HBPD, HFPD, HOZVAL and LINEVAL. For more information refer to the Figure 14-11.

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 requires their own adequate frame rate. The frame rate is calculated as follows;

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) }]

14.4.8.2 I80 Interface Controller

VTIME_I80 controls DISPLAY Controller for cpu style LDI, and has the following functions.

- Generation of I80 Interface Control Signals
- CPU Style LDI Command Control
- Timing Control for VDMA and VDPRCS

14.4.8.2.1 Output Control Signal Generation

SYS_CS0, SYS_CS1, SYS_WE and SYS_RS control signals are generated by VTIME_I80. (For more information refer to Figure 14-12 for Timing Diagram). Their timing parameters, LCD_CS_SETUP, LCD_WR_SETUP, LCD_WR_ACT, LCD_WR_HOLD can be set through I80IFCONA0 and I80IFCONA1 SFRs.

14.4.8.2.2 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.

However, this function can be implemented by SFR setting (LINEVAL, HOZVAL, OSD_LeftTopX_F, OSD_LeftTopY_F, OSD_RightBotX_F, OSD_RightBotY_F, PAGEWIDTH, OFFSIZE).

14.4.9 LDI Command Control

LDI can receive 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). Normal command is issued by S/W control.

SETTING OF COMMANDS

14.4.9.1 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.
2) LDI_CMD need not to be packed from LDI_CMD0 to LDI_CMD11 contiguously. For example, only the use of LDI_CMD0, LDI_CMD3 and LDI_CMD11 is possible.
3) Maximum 12 auto commands are available.

14.4.9.2 Normal Command

1) Put commands into LDI_CMD0 ~ 11 (maximum 12 commands)
2) Set CMDx_EN in LDI_CMDCON0 for 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 can send 12 maximum commands between frames in its normal operation (Normal operation means ENVID=1 and video data are displayed in LCD panel)
- Commands are issued in the order of CMD0 → CMD1 → CMD2 → CMD3 → ….. → CMD10 →

CMD11

- Disabled commands (CMDx_EN = 0x0) are skipped
- Sending over 12 commands

  : It is only 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, 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

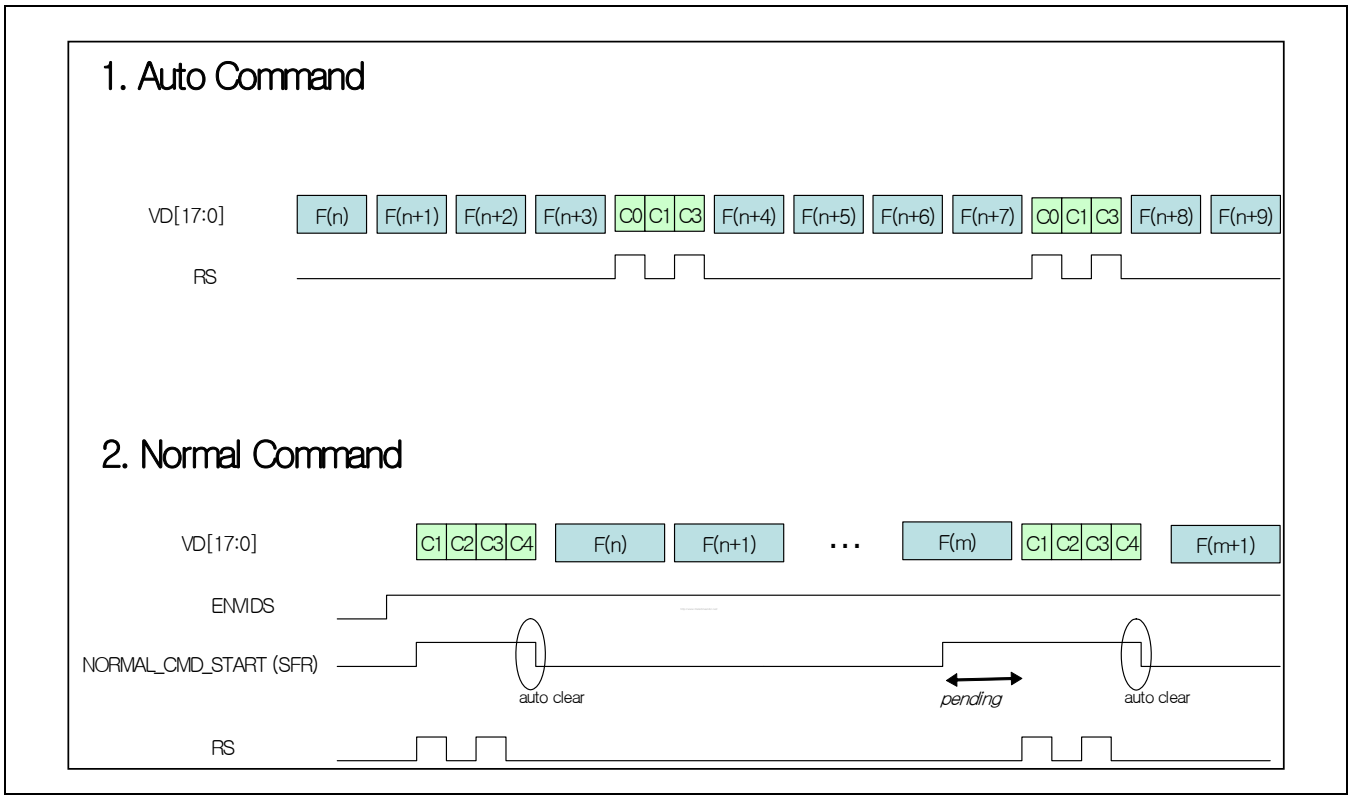


Figure 14-9. Sending Command

14.4.10 I80 CPU Interface Trigger

VTIME_I80 starts its operation only when a S/W trigger occurs. There are two kinds of triggers.

S/W trigger is generated by setting TRGCON SFR.

14.4.11 Interrupt

Frame Done Interrupt is generated at the completion of one frame.

14.4.12 Virtual Display

The DISPLAY controller supports the hardware horizontal or vertical scrolling. If the screen is scrolled, you must change the fields of LCDBASEU and LCDBASEL registers (For more information refer to Figure 14-10). You must not change the values of PAGEWIDTH and OFFSIZE. The size of video buffer in which the image is stored must be larger than the LCD panel screen size.

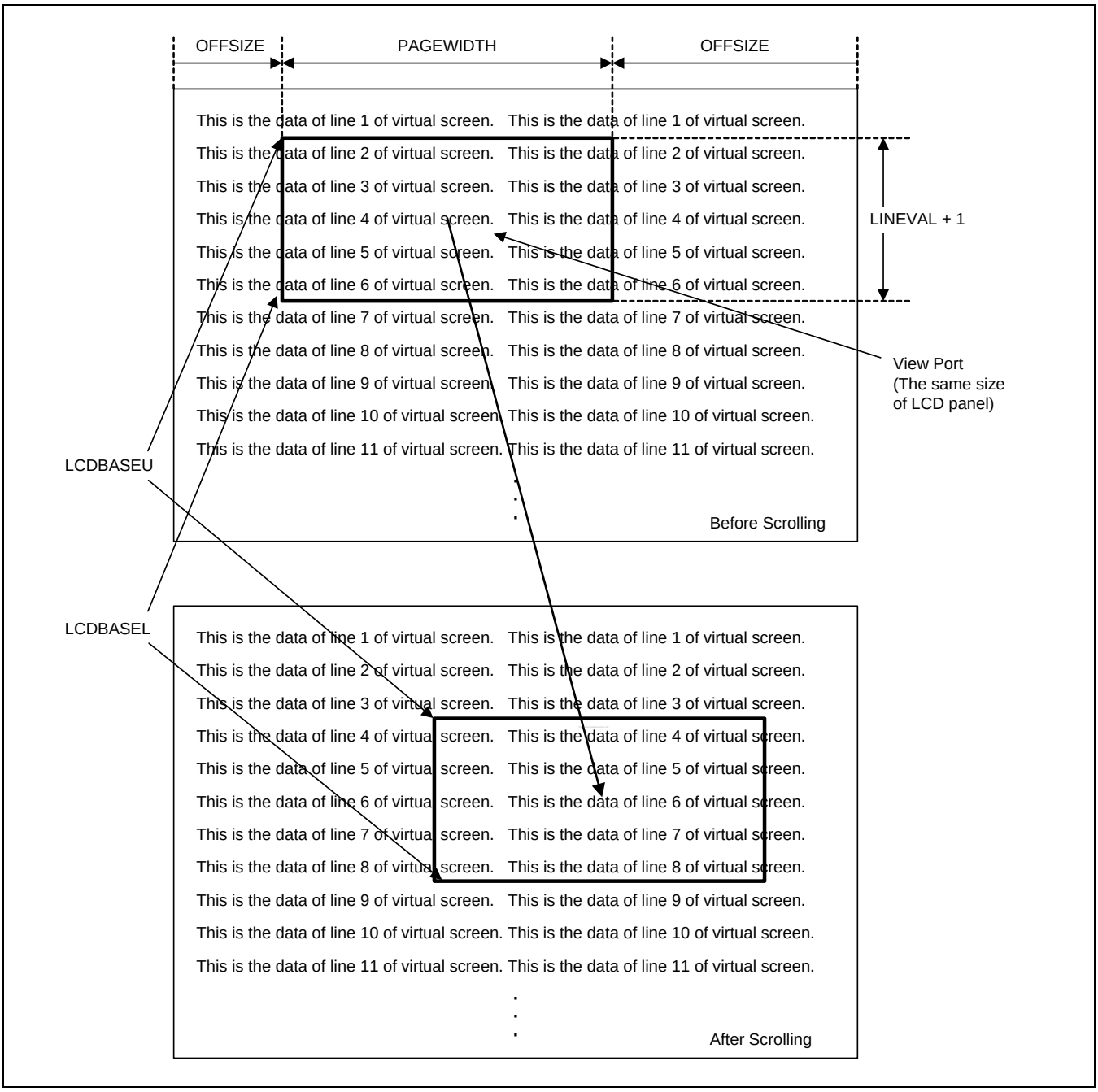


Figure 14-10. Example of Scrolling in Virtual Display

14.4.13 RGB INTERFACE IO

Signals

Table 14-5. RGB Interface Pin Description

Name	Type	Source/Destination	Description
RGB_HSYNC	Output	Pad	Horizontal Sync. Signal
RGB_VSYNC	Output	Pad	Vertical Sync. Signal
RGB_VCLK	Output	Pad	LCD Video Clock
RGB_VDEN	Output	Pad	Data Enable
RGB_VD[23:0]	Output	Pad	RGB data output. In 16bpp, pins match with following as RGB_VD[23:19] : R RGB_VD[15:10] : G RGB_VD[7:3] : B Please, refer to Table 14-7 for more information.

NOTE: LCD_SEL[1:0] value @ 0x7F0081A0 must be set as '01' to use RGB I/F Style. Please refer to GPIO Manual for more information.

Timing

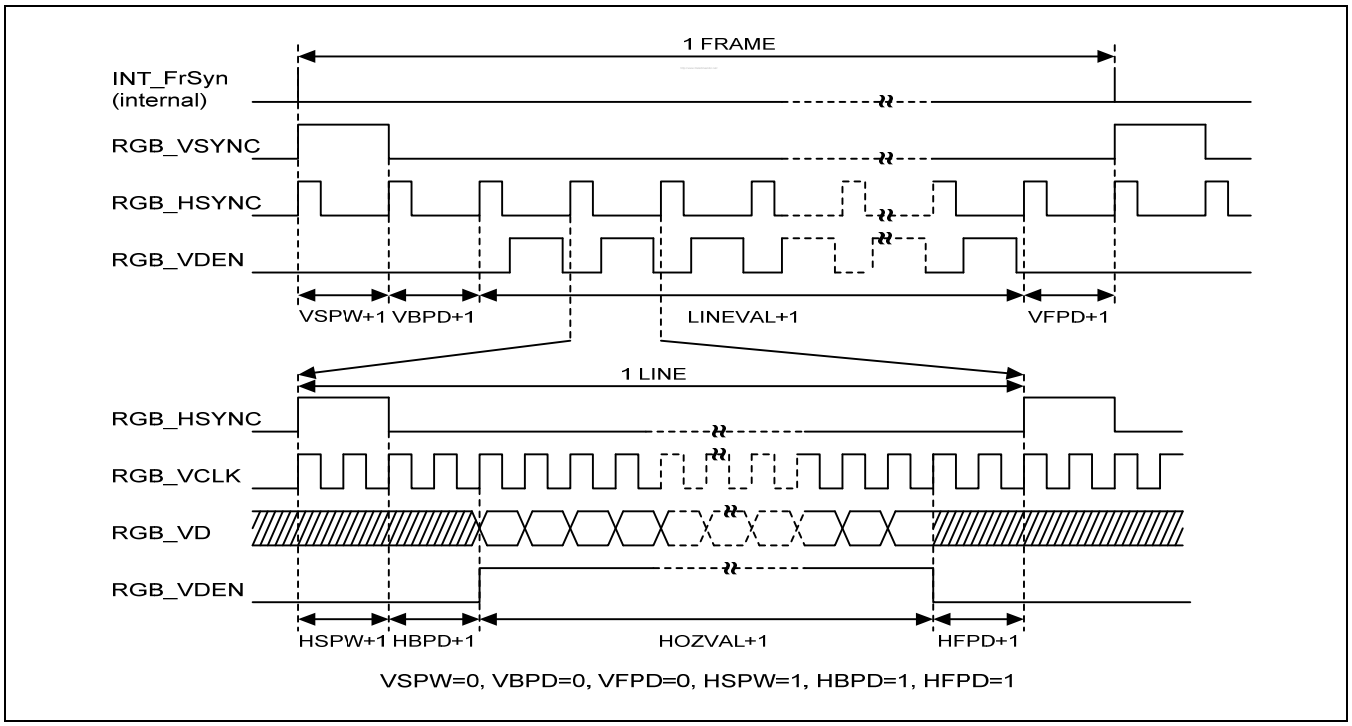


Figure 14-11. LCD RGB interface Timing

14.4.14 LCD I80 INTERFACE IO

Signals

Table 14-6. I80 CPU Interface Pin Description

Name	Type	Source/Destination	Description
SYS_VDIN[17:0]	In	Video Mux	Video Data Input
SYS_VDOUT[17:0]	Out	Video Mux	Video Data Output
SYS_CS0	Output	Video Mux	Chip select for LCD0
SYS_CS1	Output	Video Mux	Chip select for LCD1
SYS_WE	Output	Video Mux	Write enable
SYS_OE	Output	Video Mux	Output Enable
SYS_RS	Output	Video Mux	Register/State Select

NOTE: LCD_SEL[1:0] value @ 0x7F0081A0 must be set as '00' to use Host I/F Style. Please, refer to GPIO Manual for more information.

Timing

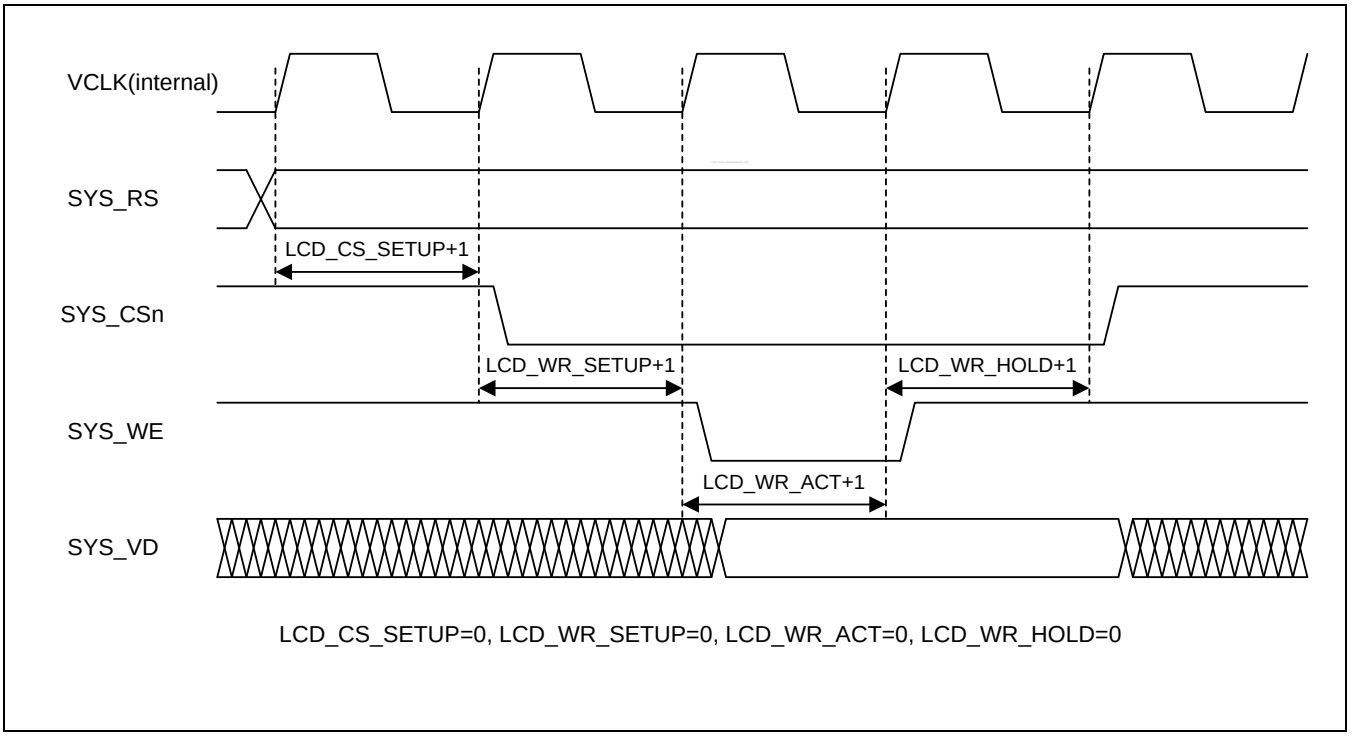


Figure 14-12. Write Cycle Timing

14.4.14 ITU-R BT.601 INTERFACE IO

Signals

Table 14-7. ITU-R BT.601 Interface Pin Description

Name	Type	Source/Destination	Description
V601_CLK	Output	Pad	ITU 601 data clock
*VEN_HREF	Output	Pad	DATA Enable
**VEN_VSYNC	Output	Pad	Vertical Sync Signal
VEN_HSYNC	Output	Pad	Horizontal Sync Signal
**VEN_FIELD	Output	Pad	FIELD Signal (option)
VEN_DATA[7:0]	Output	Pad	ITU601 YUV422 format data output

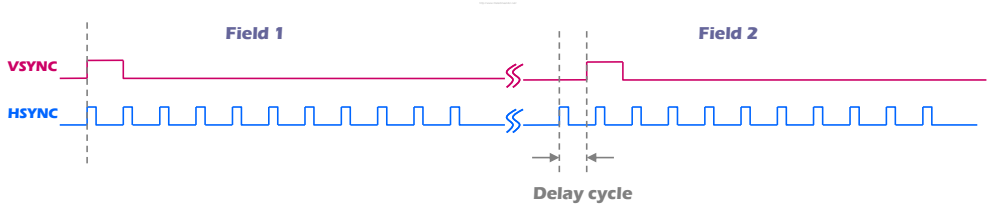
NOTE: LCD_SEL[1:0] value @ 0x7F0081A0 must be set as '10' to use 601 Style. Please, refer to GPIO Manual for more information

* VEN_HREF : DATA Blank (when I601HREF[0] = 1)

DATA Enable (when I601HREF[0] = 0)

** VEN_VSYNC, VEN_FIELD (field information in interlace mode)

When SELVSYNC[0] = 1, Delay Cycle = DLYVSYNC[7:0] + 1



When SELVSYNC[0] = 0,

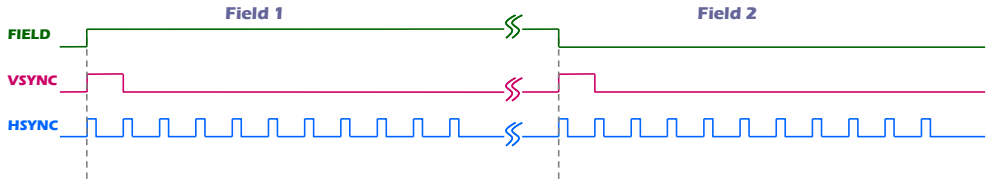


Figure 14-13. ITU-R BT.601 Controllable Vsync

14.4.15 LCD DATA PIN MAP

Table 14-8. Parallel/Serial RGB, 601 Data Pin Map (- : Not Used)

	Parallel RGB			Serial RGB		601
	24BPP (888)	18BPP (666)	16BPP (565)	24BPP (888)	18BPP (666)	
VD[23]	R[7]	R[5]	R[4]	D[7]	D[5]	
VD[22]	R[6]	R[4]	R[3]	D[6]	D[4]	
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]
VD[6]	B[6]	B[4]	B[3]	-	-	VEN_DATA[6]
VD[5]	B[5]	B[3]	B[2]	-	-	VEN_DATA[5]
VD[4]	B[4]	B[2]	B[1]	-	-	VEN_DATA[4]
VD[3]	B[3]	B[1]	B[0]	-	-	VEN_DATA[3]
VD[2]	B[2]	B[0]	-	-	-	VEN_DATA[2]
VD[1]	B[1]	-	-	-		VEN_DATA[1]
VD[0]	B[0]	-	-	-	-	VEN_DATA[0]

Table 14-9. CPU I/F Data Pin Map (- : Not Used)

	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]

14.4.16 LCD NORMAL/BY-PASS MODE SELECTION

The external modem or MCU can access the system interface LCD Panel through the by-pass. After reset, the initial output path of LCD controller is by-pass like described in below Figure 14-14. In order to operate in the normal display mode (RGB or CPU I/F), SEL_BYPASS[3] value @ 0x7410800C must be set as '0'(normal mode) instead of '1'(by-pass mode).

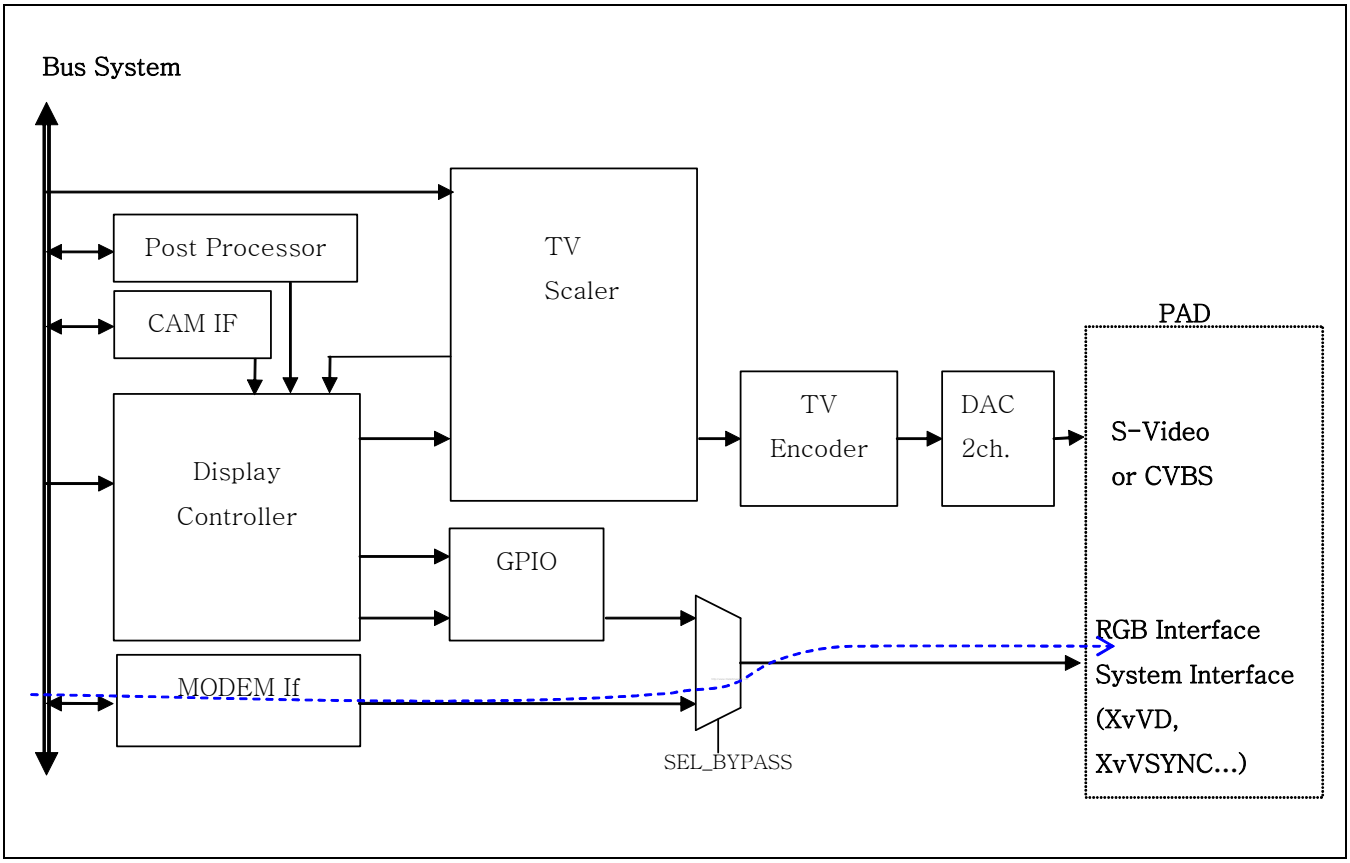


Figure 14-14. By-Pass Mode (Initial LCD Output Pass)

14.5 PROGRAMMER'S MODEL

14.5.1 OVERVIEW

The following registers are used to configure DISPLAY controller:

1. MOFPCON: SEL_BYPASS[3] value @ 0x7410800C must be set as '0'(normal mode) instead of '1'(by-pass mode).
2. SPCON: LCD_SEL[1:0] value @ 0x7F0081A0 must be set as '00' to use Host I/F Style or as '01' to use RGB I/F Style
3. VIDCON0: configure Video output format and display enable/disable.
4. VIDCON1: RGB I/F control signal.
5. I80IFCONx: i80-system I/F control signal.
6. ITUIFCON0 : ITU (BT.601) Interface Control
7. VIDTCONx: configure Video output Timing and determine the size of display.
8. WINCONx: each window format setting
9. VIDOSDxA, VIDOSDxB: Window position setting
10. VIDOSDxC: alpha value setting
11. VIDWxxADDx: source image address setting
12. WxKEYCONx: Color key value register
13. WINxMAP: window color control
14. WPALCON: Palette controls register
15. WxPDATAxx: Window Palette Data of the each Index.

14.5.2 SFR MEMORY MAP

Register	Address	R/W	Description	Reset Value
VIDCON0	0x77100000	R/W	Video control 0 register	0x0000_0000
VIDCON1	0x77100004	R/W	Video control 1 register	0x0000_0000
VIDCON2	0x77100008	R/W	Video control 2 register	0x0000_0000
VIDTCON0	0x77100010	R/W	Video time control 0 register	0x0000_0000
VIDTCON1	0x77100014	R/W	Video time control 1 register	0x0000_0000
VIDTCON2	0x77100018	R/W	Video time control 2 register	0x0000_0000
WINCON0	0x77100020	R/W	Window control 0 register	0x0000_0000
WINCON1	0x77100024	R/W	Window control 1 register	0x0000_0000
WINCON2	0x77100028	R/W	Window control 2 register	0x0000_0000
WINCON3	0x7710002C	R/W	Window control 3 register	0x0000_0000
WINCON4	0x77100030	R/W	Window control 4 register	0x0000_0000
VIDOSD0A	0x77100040	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0B	0x77100044	R/W	Video Window 0's position control register	0x0000_0000
VIDOSD0C	0x77100048	R/W	Video Window 0's size control register	0x0000_0000
VIDOSD1A	0x77100050	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1B	0x77100054	R/W	Video Window 1's position control register	0x0000_0000
VIDOSD1C	0x77100058	R/W	Video Window 1's alpha control register	0x0000_0000

Register	Address	R/W	Description	Reset Value
VIDOSD1D	0x7710005C	R/W	Video Window 1's size control register	0x0000_0000
VIDOSD2A	0x77100060	R/W	Video Window 2's position control register	0x0000_0000
VIDOSD2B	0x77100064	R/W	Video Window 2's position control register	0x0000_0000
VIDOSD2C	0x77100068	R/W	Video Window 2's alpha control register	0x0000_0000
VISOSD2D	0x7710006C	R/W	Video Window 2's size control register	0x0000_0000
VIDOSD3A	0x77100070	R/W	Video Window 3's position control register	0x0000_0000
VIDOSD3B	0x77100074	R/W	Video Window 3's position control register	0x0000_0000
VIDOSD3C	0x77100078	R/W	Video Window 3's alpha control register	0x0000_0000
VIDOSD4A	0x77100080	R/W	Video Window 4's position control register	0x0000_0000
VIDOSD4B	0x77100084	R/W	Video Window 4's position control register	0x0000_0000
VIDOSD4C	0x77100088	R/W	Video Window 4's alpha control register	0x0000_0000
VIDW00ADD0B0	0x771000A0	R/W	Window 0's buffer start address register, buffer 0	0x0000_0000
VIDW00ADD0B1	0x771000A4	R/W	Window 0's buffer start address register, buffer 1	0x0000_0000
VIDW01ADD0B0	0x771000A8	R/W	Window 1's buffer start address register, buffer 0	0x0000_0000
VIDW01ADD0B1	0x771000AC	R/W	Window 1's buffer start address register, buffer 1	0x0000_0000
VIDW02ADD0	0x771000B0	R/W	Window 2's buffer start address register	0x0000_0000
VIDW03ADD0	0x771000B8	R/W	Window 3's buffer start address register	0x0000_0000
VIDW04ADD0	0x771000C0	R/W	Window 4's buffer start address register	0x0000_0000
VIDW00ADD1B0	0x771000D0	R/W	Window 0's buffer end address register, buffer 0	0x0000_0000
VIDW00ADD1B1	0x771000D4	R/W	Window 0's buffer end address register, buffer 1	0x0000_0000
VIDW01ADD1B0	0x771000D8	R/W	Window 1's buffer end address register, buffer 0	0x0000_0000
VIDW01ADD1B1	0x771000DC	R/W	Window 1's buffer end address register, buffer 1	0x0000_0000
VIDW02ADD1	0x771000E0	R/W	Window 2's buffer end address register	0x0000_0000
VIDW03ADD1	0x771000E8	R/W	Window 3's buffer end address register	0x0000_0000
VIDW04ADD1	0x771000F0	R/W	Window 4's buffer end address register	0x0000_0000
VIDW00ADD2	0x77100100	R/W	Window 0's buffer size register	0x0000_0000
VIDW01ADD2	0x77100104	R/W	Window 1's buffer size register	0x0000_0000
VIDW02ADD2	0x77100108	R/W	Window 2's buffer size register	0x0000_0000
VIDW03ADD2	0x7710010C	R/W	Window 3's buffer size register	0x0000_0000
VIDW04ADD2	0x77100110	R/W	Window 4's buffer size register	0x0000_0000
VIDINTCON0	0x77100130	R/W	Indicate the Video interrupt control register	0x03F0_0000
VIDINTCON1	0x77100134	R/W	Video Interrupt Pending register	0x0000_0000
W1KEYCON0	0x77100140	R/W	Color key control register	0x0000_0000
W1KEYCON1	0x77100144	R/W	Color key value (transparent value) register	0x0000_0000
W2KEYCON0	0x77100148	R/W	Color key control register	0x0000_0000
W2KEYCON1	0x7710014C	R/W	Color key value (transparent value) register	0x0000_0000
W3KEYCON0	0x77100150	R/W	Color key control register	0x0000_0000
W3KEYCON1	0x77100154	R/W	Color key value (transparent value) register	0x0000_0000
W4KEYCON0	0x77100158	R/W	Color key control register	0x0000_0000
W4KEYCON1	0x7710015C	R/W	Color key value (transparent value) register	0x0000_0000

Register	Address	R/W	Description	Reset Value
DITHMODE	0x77100170	R/W	Dithering mode register.	0x0000_0000
WIN0MAP	0x77100180	R/W	Window color control	0x0000_0000
WIN1MAP	0x77100184	R/W	Window color control	0x0000_0000
WIN2MAP	0x77100188	R/W	Window color control	0x0000_0000
WIN3MAP	0x7710018C	R/W	Window color control	0x0000_0000
WIN4MAP	0x77100190	R/W	Window color control	0x0000_0000
WPALCON	0x771001A0	R/W	Window Palette control register	0x0000_0000
TRIGCON	0x771001A4	R/W	I80 / RGB Trigger Control register	0x0000_0000
ITUIFCON0	0x771001A8	R/W	ITU (BT.601) Interface Control	0x0000_0000
I80IFCONA0	0x771001B0	R/W	I80 Interface control 0 for Main LDI	0x0000_0000
I80IFCONA1	0x771001B4	R/W	I80 Interface control 0 for Sub LDI	0x0000_0000
I80IFCONB0	0x771001B8	R/W	I80 Interface control 1 for Main LDI	0x0000_0000
I80IFCONB1	0x771001BC	R/W	I80 Interface control 1 for Sub LDI	0x0000_0000
LDI_CMDCON0	0x771001D0	R/W	I80 Interface LDI Command Control 0	0x0000_0000
LDI_CMDCON1	0x771001D4	R/W	I80 Interface LDI Command Control 1	0x0000_0000
SIFCCON0	0x771001E0	R/W	LCD I80 System Interface Manual Command Control	0x0000_0000
SIFCCON1	0x771001E4	R/W	LCD I80 System Interface Manual Command Data Write Control	0x0000_0000
SIFCCON2	0x771001E8	R	LCD I80 System Interface Manual Command Data Read Control 2	undefined
LDI_CMD0	0x77100280	R/W	I80 Interface LDI Command 0	0x0000_0000
LDI_CMD1	0x77100284	R/W	I80 Interface LDI Command 1	0x0000_0000
LDI_CMD2	0x77100288	R/W	I80 Interface LDI Command 2	0x0000_0000
LDI_CMD3	0x7710028C	R/W	I80 Interface LDI Command 3	0x0000_0000
LDI_CMD4	0x77100290	R/W	I80 Interface LDI Command 4	0x0000_0000
LDI_CMD5	0x77100294	R/W	I80 Interface LDI Command 5	0x0000_0000
LDI_CMD6	0x77100298	R/W	I80 Interface LDI Command 6	0x0000_0000
LDI_CMD7	0x7710029C	R/W	I80 Interface LDI Command 7	0x0000_0000
LDI_CMD8	0x771002A0	R/W	I80 Interface LDI Command 8	0x0000_0000
LDI_CMD9	0x771002A4	R/W	I80 Interface LDI Command 9	0x0000_0000
LDI_CMD10	0x771002A8	R/W	I80 Interface LDI Command 10	0x0000_0000
LDI_CMD11	0x771002AC	R/W	I80 Interface LDI Command 11	0x0000_0000
W2PDATA01	0x77100300	R/W	Window 2 Palette Data of the Index 0,1	0x0000_0000
W2PDATA23	0x77100304	R/W	Window 2 Palette Data of the Index 2,3	0x0000_0000
W2PDATA45	0x77100308	R/W	Window 2 Palette Data of the Index 4,5	0x0000_0000
W2PDATA67	0x7710030C	R/W	Window 2 Palette Data of the Index 6,7	0x0000_0000
W2PDATA89	0x77100310	R/W	Window 2 Palette Data of the Index 8,9	0x0000_0000

Register	Address	R/W	Description	Reset Value
W2PDATAAB	0x77100314	R/W	Window 2 Palette Data of the Index A, B	0x0000_0000
W2PDATACD	0x77100318	R/W	Window 2 Palette Data of the Index C, D	0x0000_0000
W2PDATAEF	0x7710031C	R/W	Window 2 Palette Data of the Index E, F	0x0000_0000
W3PDATA01	0x77100320	R/W	Window 3 Palette Data of the Index 0,1	0x0000_0000
W3PDATA23	0x77100324	R/W	Window 3 Palette Data of the Index 2,3	0x0000_0000
W3PDATA45	0x77100328	R/W	Window 3 Palette Data of the Index 4,5	0x0000_0000
W3PDATA67	0x7710032C	R/W	Window 3 Palette Data of the Index 6,7	0x0000_0000
W3PDATA89	0x77100330	R/W	Window 3 Palette Data of the Index 8,9	0x0000_0000
W3PDATAAB	0x77100334	R/W	Window 3 Palette Data of the Index A, B	0x0000_0000
W3PDATACD	0x77100338	R/W	Window 3 Palette Data of the Index C, D	0x0000_0000
W3PDATAEF	0x7710033C	R/W	Window 3 Palette Data of the Index E, F	0x0000_0000
W4PDATA01	0x77100340	R/W	Window 4 Palette Data of the Index 0,1	0x0000_0000
W4PDATA23	0x77100344	R/W	Window 4 Palette Data of the Index 2,3	0x0000_0000

14.6 INDIVIDUAL REGISTER DESCRIPTIONS

14.6.1 VIDEO MAIN CONTROL 0 REGISTER

Register	Address	R/W	Description	Reset Value
VIDCON0	0x77100000	R/W	Video control 0 register	0x0000_0000

VIDCON0	Bit	Description	Initial State
Reserved	[31:30]	Reserved	0
INTERLACE_F	[29]	Interlace or Progressive 0 : Progressive 1 : Interlace	0
Reserved	[28]	Reserved (Must be zero)	0
VIDOUT	[27:26]	It determines the output format of Display Controller 00: RGB I/F 01: TV Interface (Encoder or ITU601) 10: I80 I/F for LDI0 11: I80 I/F for LDI1	00
L1_DATA16	[25:23]	Select the mode of output data format of I80 I/F (LDI1.) (Only when, 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

VIDCON0	Bit	Description	Initial State
L0_DATA16	[22:20]	Select the mode of output data format of I80 CPU I/F (LDI0.) (Only when, 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	0
PNRMODE	[18:17]	Select 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]	Select CLKVAL_F update timing control 0 = always 1 = start of a frame (only once per frame)	0
Reserved	[15:14]	Reserved	
CLKVAL_F	[13:6]	Determine the rates of VCLK and CLKVAL[7:0] VCLK = Video Clock Source / (CLKVAL+1) where CLKVAL >= 1 Note. 1. The maximum frequency of VCLK is 66MHz. 2. Video Clock Source is selected by CLKSEL_F register	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]	Select 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	0x0
CLKSEL_F	[3:2]	Select the Video Clock source 00 = HCLK 01 = LCD video Clock (from SYSCON) 10 = reserved 11 = 27MHz Ext Clock input	0
ENVID	[1]	Video output and the logic immediately enable/disable. 0 = Disable the video output and the Display control signal. 1 = Enable the video output and the Display control signal.	0
ENVID_F	[0]	Video output and the logic enable/disable at current frame end. 0 = Disable the video output and the Display control signal. 1 = Enable the video output and the Display control signal. * If set on and off this bit, then you will read "H" and video controller enable until the end of current frame.	0

NOTE: 1.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: In normal display mode, SEL_BYPASS@ MIFPCON (0x7410_800C) register must be set “0”.

Caution 2: When 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 3 : When display controller is off using direct-off, it is impossible to turn on the display controller without reset.

14.6.2 VIDEO MAIN CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
VIDCON1	0x77100004	R/W	Video control 1 register	0x0000_0000

VIDCON1	Bit	Description	Initial state
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	[10:8]	Reserved	
IVCLK	[7]	This bit controls the polarity of the VCLK active edge. 0 = RGB type LCD driver gets the video data at VCLK falling edge 1 = RGB type LCD driver gets the video data 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

14.6.3 VIDEO MAIN CONTROL 2 REGISTER

Register	Address	R/W	Description	Reset Value
VIDCON2	0x77100008	R/W	Video control 2 register	0x0000_0000

VIDCON1	Bit	Description	Initial state
-	[31:24]	Reserved	0
EN601	[23]	Control ITU601 output enable 0 = Disable 1 = Enable	0
-	[22:15]	Reserved	0
TVFORMATSEL0	[14]	This bit indicates method of YUV data format selection. 0 = Hardware 1 = Software (use TVFORMATSEL1[1:0] bits)	0
TVFORMATSEL1	[13:12]	This bit indicates output format of YUV data. 00 = RGB 01 = YUV422 1x = YUV444	0
	[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
-	[6:0]	Reserved	0

14.6.4 VIDEO TIME CONTROL 0 REGISTER

Register	Address	R/W	Description	Reset Value
VIDTCON0	0x77100010	R/W	Video time control 0 register	0x0000_0000

VIDTCON0	Bit	Description	Initial State
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)	0x0
VBPD	[23:16]	Vertical back porch is the number of inactive lines at the start of a frame, after vertical synchronization period.	0x0
VFPD	[15:8]	Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period.	0x0
VSPW	[7:0]	Vertical sync pulse width determines the VSYNC pulse's high level width by counting the number of inactive lines.	0x0

14.6.5 VIDEO TIME CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
VIDTCON1	0x77100014	R/W	Video time control 1 register	0x0000_0000

VIDTCON1	Bit	Description	Initial state
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)	0x0
HBPD	[23:16]	Horizontal back porch is the number of VCLK periods between the falling edge of HSYNC and the start of active data.	0x0
HFPD	[15:8]	Horizontal front porch is the number of VCLK periods between the end of active data and the rising edge of HSYNC.	0x0
HSPW	[7:0]	Horizontal sync pulse width determines the HSYNC pulse's high level width by counting the number of the VCLK.	0x0

14.6.6 VIDEO TIME CONTROL 2 REGISTER

Register	Address	R/W	Description	Reset Value
VIDTCON2	0x77100018	R/W	Video time control 2 register	0x0000_0000

VIDTCON2	Bit	Description	Initial state
LINEVAL	[21:11]	These bits determine the vertical size of display	0
HOZVAL	[10:0]	These bits determine the horizontal size of display	0

NOTE: HOZVAL = (Horizontal display size) -1, LINEVAL = (Vertical display size) –1

14.6.7 WINDOW 0 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
WINCON0	0x77100020	R/W	Window 0 control register	0x0000_0000

WINCON0	Bit	Description	Initial State
nWide/Narrow	[27:26]	Select 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	[25:23]	Reserved	0
ENLOCAL	[22]	Select Data access method. 0: Dedicated DMA 1: Local Path from Postprocessor (POST Processor FIFO Out) Note. This register must be disabled at the ENWIN_F disable state.	0

WINCON0	Bit	Description	Initial State
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 = Automatically changed by H/W control signal	0
BITSWP	[18]	Bit swap control bit. 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
reserved	[15:14]	Must be '0'	0
InRGB	[13]	It indicates the input color space of source image. (Only for 'ENLOCALl' enable) 0: RGB 1: YCbCr	0
reserved	[12:11]	Should be '0'	0
BURSTLEN	[10:9]	DMA's Burst Maximum Length selection: 00: 16 word – burst 01: 8 word – burst 10: 4 word – burst	0
Reserved	[8:6]	Reserved	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 = reserved 0101 = 16 BPP (non-palletized, R: 5-G:6-B:5) 0110 = reserved 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 = reserved 1010 = reserved 1011 = unpacked 24 BPP (non-palletized R:8-G:8-B:8) 11xx = reserved	0
Reserved	[1]	Reserved	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

14.6.8 WINDOW 1 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
WINCON1	0x77100024	R/W	Window control 1 register	0x0000_0000

WINCON1	Bit	Description	Initial State
nWide/Narrow	[27:26]	Select 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
LOCALSEL	[23]	Select Local Path Source 0 : TV scaler (FIFO Out) 1 : Camera preview local IF Note. When the camera preview local IF is selected, there is a constraint that the source of camera preview local IF must be memory not external camera. Transfer from memory to camera preview local IF can be done by MS-DMA.	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 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
reserved	[15:14]	Must be '0'	0
InRGB	[13]	It indicates the input color space of source image. (Only for 'ENLOCAL' enable) 0: RGB 1: YCbCr	0
reserved	[12:11]	Must be '0'	0

WINCON1	Bit	Description	Initial State
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]	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) 111x = reserved Note. 1101 can support unpacked 28 BPP also (non-palletized A:4-R:8-G:8-B:8) ,at BLD_PIX = 1.	0
ALPHA_SEL	[1]	Select Alpha value by When Per plane blending case(BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values When Per pixel blending (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[27:24] data (only when BPPMODE_F = 4'b1101)	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

14.6.9 WINDOW 2 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
WINCON2	0x77100028	R/W	Window 2 control register	0x0000_0000

WINCON2	Bit	Description	Initial State
nWide/Narrow	[27:26]	Select 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
LOCALSEL	[23]	Select Local Path Source 0 : TV scaler (FIFO Out) 1 : Camera codec local IF Note. When the camera codec local IF is selected, there is a constraint that the source of camera codec local IF must be memory not external camera. Transfer from memory to camera codec local IF can be done by MS-DMA.	0
ENLOCAL	[22]	Select Data access method. 0: Dedicated DMA 1: Local Path Note. This register must be disable at the ENWIN_F disable state.	0
Reserved	[21:19]	Reserved	
BITSWP	[18]	Bit swap control bit. 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
reserved	[15:12]	Must be '0'	0
InRGB	[13]	It indicates the input color space of source image. (Only for 'ENLOCAL' enable) 0: RGB 1: YCbCr	0
reserved	[12:11]	Should be '0'	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

WINCON2	Bit	Description	Initial State
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 = reserved 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) 111x = reserved Note. 1101 can support unpacked 28 BPP also (non-palletized A:4-R:8-G:8-B:8) ,at BLD_PIX = 1.	0
ALPHA_SEL	[1]	Select Alpha value by When (BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values When (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA [27:24] data (only when BPPMODE_F = 4'b1101)	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

14.6.10 WINDOW 3 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
WINCON3	0x7710002C	R/W	Window control 3 register	0x0000_0000

WINCON3	Bit	Description	Initial State
BITSWP	[18]	Bit swap control bit. 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Bytes swap control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
reserved	[15:11]	Must be '0'	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]	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 (LUT) 0001 = 2 BPP (LUT) 0010 = 4 BPP (LUT) 0011 = reserved 0100 = reserved 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) 111x = reserved Note. 1101 can support unpacked 28 BPP also (non-palletized A:4-R:8-G:8-B:8) ,at BLD_PIX = 1.	0
ALPHA_SEL	[1]	Select Alpha value by When (BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values When (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[27:24] data (only when BPPMODE_F = 4'b1101)	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

14.6.11 WINDOW 4 CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
WINCON4	0x77100030	R/W	Window control 4 register	0x0000_0000

WINCON4	Bit	Description	Initial State
BITSWP	[18]	Bit swap control bit. 0 = Swap Disable 1 = Swap Enable	0
BYTSWP	[17]	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
HAWSWP	[16]	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
Reserved	[15:11]	Must be '0'	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]	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 (LUT) 0001 = 2 BPP (LUT) 0010 = reserved 0011 = reserved 0100 = reserved 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) 111x = reserved **Note.** 1101 can support unpacked 28 BPP also (non-palletized A:4-R:8-G:8-B:8) ,at BLD_PIX = 1.	0
ALPHA_SEL	[1]	Select Alpha value by When (BLD_PIX ==0) 0 = using ALPHA0_R/G/B values 1 = using ALPHA1_R/G/B values When (BLD_PIX ==1) 0 = selected by AEN (A value) or chroma key 1 = using DATA[27:24] data (only when BPPMODE_F = 4'b1101)	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

14.6.12 WINDOW 0 POSITION CONTROL A REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD0A	0x77100040	R/W	Video Window 0's position control register	0x0000_0000

VIDOSD0A	Bit	Description	Initial state
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. And the original screen y coordinate MUST be even value.)	0

14.6.13 WINDOW 0 POSITION CONTROL B REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD0B	0x77100044	R/W	Video Window 0's position control register	0x0000_0000

VIDOSD0B	Bit	Description	Initial state
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. And 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….)

14.6.14 WINDOW 0 POSITION CONTROL C REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD0C	0x77100048	R/W	Video Window 0's Size control register	0x0000_0000

VIDOSD0C	Bit	Description	Initial state
-	[25:24]	Reserved	0
OSDSIZE	[23:0]	Window Size Eq. Height * Width (Number of Word) Note. Set filed value for YUV if (TV Encoder IF)	0

14.6.15 WINDOW 1 POSITION CONTROL A REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD1A	0x77100050	R/W	Video Window 1's position control 2 register	0x0000_0000

VIDOSD1A	Bit	Description	Initial state
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. And the original screen y coordinate MUST be even value.)	0

14.6.16 WINDOW 1 POSITION CONTROL B REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD1B	0x77100054	R/W	Video Window 1's position control register	0x0000_0000

VIDOSD1B	Bit	Description	Initial state
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. And 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….)

14.6.17 WINDOW 1 POSITION CONTROL C REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD1C	0x77100058	R/W	Video Window 1's alpha control register	0x0000_0000

VIDOSD1C	Bit	Description	Initial state
-	[24]	Reserved	0
ALPHA0_R	[23:20]	Red Alpha value(case AEN == 0)	0
ALPHA0_G	[19:16]	Green Alpha value(case AEN == 0)	0
ALPHA0_B	[15:12]	Blue Alpha value(case AEN == 0)	0
ALPHA1_R	[11:8]	Red Alpha value(case AEN == 1)	0
ALPHA1_G	[7:4]	Green Alpha value(case AEN == 1)	0
ALPHA1_B	[3:0]	Blue Alpha value(case AEN == 1)	0

14.6.18 WINDOW 1 POSITION CONTROL D REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD1D	0x7710005C	R/W	Video Window 0's Size control register	0x0000_0000

VIDOSD1D	Bit	Description	Initial state
-	[25]	Reserved	0
-	[24]	Reserved	0
OSDSIZE	[23:0]	Window Size Eq. Height * Width(Number of Word) **Note.** Set filed value for YUV if (TV Encoder IF)	0

14.6.19 WINDOW 2 POSITION CONTROL A REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD2A	0x77100060	R/W	Video Window 2's position control register	0x0000_0000

VIDOSD2A	Bit	Description	Initial state
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. And the original screen y coordinate MUST be even value.)	0

14.6.20 WINDOW 2 POSITION CONTROL B REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD2B	0x77100064	R/W	Video Window 2's position control register	0x0000_0000

VIDOSD2B	Bit	Description	Initial state
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. And 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….)

14.6.21 WINDOW 2 POSITION CONTROL C REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD2C	0x77100068	R/W	Video Window 2's alpha control register	0x0000_0000

VIDOSD2C	Bit	Description	Initial state
-	[24]	Reserved	0
ALPHA0_R	[23:20]	Red Alpha value(case AEN == 0)	0
ALPHA0_G	[19:16]	Green Alpha value(case AEN == 0)	0
ALPHA0_B	[15:12]	Blue Alpha value(case AEN == 0)	0
ALPHA1_R	[11:8]	Red Alpha value(case AEN == 1)	0
ALPHA1_G	[7:4]	Green Alpha value(case AEN == 1)	0
ALPHA1_B	[3:0]	Blue Alpha value(case AEN == 1)	0

14.6.22 WINDOW 2 POSITION CONTROL D REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD2D	0x7710006C	R/W	Video Window 0's Size control register	0x0000_0000

VIDOSD2D	Bit	Description	Initial state
-	[25:24]	Reserved	0
OSDSIZE	[23:0]	Window Size Eq. Height * Width(Number of Word) Note. Set filed value for YUV if (TV Encoder IF)	0

14.6.23 WINDOW 3 POSITION CONTROL A REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD3A	0x77100070	R/W	Video Window 3's position control register	0x0000_0000

VIDOSD3A	Bit	Description	Initial state
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. And the original screen y coordinate MUST be even value.)	0

14.6.24 WINDOW 3 POSITION CONTROL B REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD3B	0x77100074	R/W	Video Window 3's position control register	0x0000_0000

VIDOSD3B	Bit	Description	Initial state
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. And 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….)

14.6.25 WINDOW 3 POSITION CONTROL C REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD3C	0x77100078	R/W	Video Window 3's alpha control register	0x0000_0000

VIDOSD3C	Bit	Description	Initial state
-	[24]	Reserved	0
ALPHA0_R	[23:20]	Red Alpha value(case AEN == 0)	0
ALPHA0_G	[19:16]	Green Alpha value(case AEN == 0)	0
ALPHA0_B	[15:12]	Blue Alpha value(case AEN == 0)	0
ALPHA1_R	[11:8]	Red Alpha value(case AEN == 1)	0
ALPHA1_G	[7:4]	Green Alpha value(case AEN == 1)	0
ALPHA1_B	[3:0]	Blue Alpha value(case AEN == 1)	0

14.6.26 WINDOW 4 POSITION CONTROL A REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD4A	0x77100080	R/W	Video Window 4's position control register	0x0000_0000

VIDOSD4A	Bit	Description	Initial state
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. And the original screen y coordinate MUST be even value.)	0

14.6.27 WINDOW 4 POSITION CONTROL B REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD4B	0x77100084	R/W	Video Window 4's position control register	0x0000_0000

VIDOSD4B	Bit	Description	Initial state
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. And 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….)

14.6.28 WINDOW 4 POSITION CONTROL C REGISTER

Register	Address	R/W	Description	Reset Value
VIDOSD4C	0x77100088	R/W	Video Window 4's alpha control register	0x0000_0000

VIDOSD4C	Bit	Description	Initial state
-	[24]	Reserved	0
ALPHA0_R	[23:20]	Red Alpha value(case AEN == 0)	0
ALPHA0_G	[19:16]	Green Alpha value(case AEN == 0)	0
ALPHA0_B	[15:12]	Blue Alpha value(case AEN == 0)	0
ALPHA1_R	[11:8]	Red Alpha value(case AEN == 1)	0
ALPHA1_G	[7:4]	Green Alpha value(case AEN == 1)	0
ALPHA1_B	[3:0]	Blue Alpha value(case AEN == 1)	0

14.6.29 FRAME BUFFER ADDRESS 0 REGISTER

Register	Address	R/W	Description	Reset Value
VIDW00ADD0B0	0x771000A0	R/W	Window 0's buffer start address register, buffer 0	0x0000_0000
VIDW00ADD0B1	0x771000A4	R/W	Window 0's buffer start address register, buffer 1	0x0000_0000
VIDW01ADD0B0	0x771000A8	R/W	Window 1's buffer start address register, buffer 0	0x0000_0000
VIDW01ADD0B1	0x771000AC	R/W	Window 1's buffer start address register, buffer 1	0x0000_0000
VIDW02ADD0	0x771000B0	R/W	Window 2's buffer start address register	0x0000_0000
VIDW03ADD0	0x771000B8	R/W	Window 3's buffer start address register	0x0000_0000
VIDW04ADD0	0x771000C0	R/W	Window 4's buffer start address register	0x0000_0000

VIDWxxADD0	Bit	Description	Initial State
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

14.6.30 FRAME BUFFER ADDRESS 1 REGISTER

Register	Address	R/W	Description	Reset Value
VIDW00ADD1B0	0x771000D0	R/W	Window 0's buffer end address register, buffer 0	0x0000_0000
VIDW00ADD1B1	0x771000D4	R/W	Window 0's buffer end address register, buffer 1	0x0000_0000
VIDW01ADD1B0	0x771000D8	R/W	Window 1's buffer end address register, buffer 0	0x0000_0000
VIDW01ADD1B1	0x771000DC	R/W	Window 1's buffer end address register, buffer 1	0x0000_0000
VIDW02ADD1	0x771000E0	R/W	Window 2's buffer end address register	0x0000_0000
VIDW03ADD1	0x771000E8	R/W	Window 3's buffer end address register	0x0000_0000
VIDW04ADD1	0x771000F0	R/W	Window 4's buffer end address register	0x0000_0000

VIDWxxADD1	Bit	Description	Initial State
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

14.6.31 FRAME BUFFER ADDRESS 2 REGISTER

Register	Address	R/W	Description	Reset Value
VIDW00ADD2	0x77100100	R/W	Window 0's buffer size register	0x0000_0000
VIDW01ADD2	0x77100104	R/W	Window 1's buffer size register	0x0000_0000
VIDW02ADD2	0x77100108	R/W	Window 2's buffer size register	0x0000_0000
VIDW03ADD2	0x7710010C	R/W	Window 3's buffer size register	0x0000_0000
VIDW04ADD2	0x77100110	R/W	Window 4's buffer size register	0x0000_0000

VIDWxxADD2	Bit	Description	Initial State
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

14.6.32 VIDEO INTERRUPT CONTROL 0 REGISTER

Register	Address	R/W	Description	Reset Value
VIDINTCON0	0x77100130	R/W	Indicate the Video interrupt control 0 register	0x3F00000

VIDINTCON0	Bit	Description	Initial state
FIFOINTERVAL	[25:20]	These bits control the interval of the FIFO interrupt.	0x3F
SYSMAINCON	[19]	Sending complete interrupt enable bit to Main LCD 0 = Interrupt Disable. 1 = Interrupt Enable.	0
SYSSUBCON	[18]	Sending complete interrupt enable bit to Sub LCD 0 = Interrupt Disable. 1 = Interrupt Enable.	0
I80IFDONE	[17]	I80 Interface Interrupt Enable control (only for I80 Interface mode). 0 = Interrupt Disable. 1 = Interrupt Enable.	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

VIDINTCON0	Bit	Description	Initial state
INTFRMEN	[12]	Video Frame interrupts Enable control bit. 0 = Video Frame Interrupt Disable 1 = Video Frame Interrupt Enable	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)	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
INTFIFOEN	[1]	Video FIFO interrupts Enable control bit. 0 = Video FIFO Level Interrupt Disable 1 = Video FIFO Level Interrupt Enable	0
INTEN	[0]	Video interrupts Enable control bit. 0 = Video Interrupt Disable 1 = Video Interrupt Enable	0

14.6.33 VIDEO INTERRUPT CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
VIDINTCON1	0x77100134	R/W	the Video interrupt Pending register	0x0000_0000

VIDINTCON1	Bit	Description	Initial state
Reserved	[4:3]	Reserved	0
INTI80PEND	[2]	I80 Done interrupt. To clear this bit, write “1”. 0 = The interrupt has not been requested 1 = I80 Done status has asserted the interrupt request	0
INTFRMPEND	[1]	Frame sync interrupt. To clear this bit, write “1”. 0 = The interrupt has not been requested 1 = Frame sync status has asserted the interrupt request	0
INTFIFOPEND	[0]	FIFO Level interrupt. To clear this bit, write “1”. 0 = The interrupt has not been requested 1 = FIFO empty status has asserted the interrupt request	0

14.6.34 WIN1 COLOR KEY 0 REGISTER

Register	Address	R/W	Description	Reset Value
W1KEYCON0	0x77100140	R/W	Color key control register	0x0000_0000

W1KEYCON0	Bit	Description	Initial state
KEYBLEN	[26]	Color Key (Chroma key) Enable control 0 = Disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value of fore-ground image matches with COLVAL, the pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value of back-ground matches with COLVAL, the pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit is correspond to the COLVAL[23:0]. If some position bit is set then that position bit of COLVAL will be ignored in the fore-ground or back-ground match.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0 to use alpha blending using color key,

14.6.35 WIN 1 COLOR KEY 1 REGISTER

Register	Address	R/W	Description	Reset Value
W1KEYCON1	0x77100144	R/W	Color key value (transparent value) register	0x0000_0000

W1KEYCON1	Bit	Description	Initial state
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

14.6.36 WIN2 COLOR KEY 0 REGISTER

Register	Address	R/W	Description	Reset Value
W2KEYCON0	0x77100148	R/W	Color key control register	0x0000_0000

W2KEYCON0	Bit	Description	Initial state
KEYBLEN	[26]	Color Key (Chroma key) Enable control 0 = disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
DIRCON	[24]	Color key (Chroma key) direction control 0 = If the pixel value of fore-ground image matches with COLVAL, the pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value of back-ground matches with COLVAL, the 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 that position bit of COLVAL will be ignored in the fore-ground or back-ground match.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0 to use alpha blending using color key.

14.6.37 WIN2 COLOR KEY 1 REGISTER

Register	Address	R/W	Description	Reset Value
W2KEYCON1	0x7710014C	R/W	Color key value (transparent value) register	0x0000_0000

W2KEYCON1	Bit	Description	Initial state
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

14.6.38 WIN3 COLOR KEY 0 REGISTER

Register	Address	R/W	Description	Reset Value
W3KEYCON0	0x77100150	R/W	Color key control register	0x0000_0000

W3KEYCON0	Bit	Description	Initial state
KEYBLEN	[26]	Color Key (Chroma key) Enable control 0 = disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value of fore-ground image matches with COLVAL, the pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value of back-ground matches with COLVAL, the pixel from fore-ground image is displayed (only in OSD area)	0
COMPKEY	[23:0]	Each bit is correspond to the COLVAL[23:0]. If some position bit is set then that position bit of COLVAL will be ignored in the fore-ground or back-ground match.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0 to use alpha blending using color key,

14.6.39 WIN3 COLOR KEY 1 REGISTER

Register	Address	R/W	Description	Reset Value
W3KEYCON1	0x77100154	R/W	Color key value (transparent value) register	0x0000_0000

W3KEYCON1	Bit	Description	Initial state
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

14.6.40 WIN4 COLOR KEY 0 REGISTER

Register	Address	R/W	Description	Reset Value
W4KEYCON0	0x77100158	R/W	Color key control register	0x0000_0000

W4KEYCON0	Bit	Description	Initial state
KEYBLEN	[26]	Color Key (Chroma key) Enable control 0 = disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
KEYEN_F	[25]	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
DIRCON	[24]	Color key (Chroma key)direction control 0 = If the pixel value of fore-ground image matches with COLVAL, the pixel from back-ground image is displayed (only in OSD area) 1 = If the pixel value of back-ground matches with COLVAL, the 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 that position bit of COLVAL will be ignored in the fore-ground or back-ground match.	0

NOTE: Set BLD_PIX = 1, ALPHA_SEL =0 to use alpha blending using color key,

14.6.41 WIN4 COLOR KEY 1 REGISTER

Register	Address	R/W	Description	Reset Value
W4KEYCON1	0x7710015C	R/W	Color key value (transparent value) register	0x0000_0000

W4KEYCON1	Bit	Description	Initial state
COLVAL	[23:0]	Color key value for the transparent pixel effect.	0

NOTE: COLVAL and COMPKEY use 24-bit 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.

14.6.42 DITHERING CONTROL 1 REGISTER

Register	Address	R/W	Description	Reset Value
DITHMODE	0x77100170	R/W	Dithering mode register.	0x00000

DITHMODE	Bit	Description	Initial state
-	[7]	Should be zero	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 = dithering disable 1 = dithering enable	0

14.6.43 WIN0 COLOR MAP

Register	Address	R/W	Description	Reset Value
WIN0MAP	0x77100180	R/W	Window color control	0x00000

WIN0MAP	Bit	Description	Initial state
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 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

14.6.44 WIN1 COLOR MAP

Register	Address	R/W	Description	Reset Value
WIN1MAP	0x77100184	R/W	Window color control	0x00000

WIN1MAP	Bit	Description	Initial state
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 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

14.6.45 WIN2 COLOR MAP

Register	Address	R/W	Description	Reset Value
WIN2MAP	0x77100188	R/W	Window color control	0x00000

WIN2MAP	Bit	Description	Initial state
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 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

14.6.46 WIN3 COLOR MAP

Register	Address	R/W	Description	Reset Value
WIN3MAP	0x7710018C	R/W	Window color control	0x00000

WIN3MAP	Bit	Description	Initial state
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 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

14.6.47 WIN4 COLOR MAP

Register	Address	R/W	Description	Reset Value
WIN4MAP	0x77100190	R/W	Window color control	0x0000_0000

WIN4MAP	Bit	Description	Initial state
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 = disable 1 = enable	0
MAPCOLOR	[23:0]	Color Value	0

14.6.48 WINDOW PALETTE CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
WPALCON	0x771001A0	R/W	Window Palette control register	0x0000_0000

WPALCON	Bit	Description	Initial state
PALUPDATEEN	[9]	Palette memory access-right control bit. Users should set this bit before access (write or read) palette memory, in this case LCD controller cannot access palette. After update, users should clear this bit for operation of palletized LCD. 0: Normal Mode (LCD controller access) 1: Enable (ARM access)	0
W4PAL	[8]	This bit determines the size of the palette data format of Window 4 0 = 16-bit (5:6:5) 1 = 16-bit (A:5:5:5)	0
W3PAL	[7]	This bit determines the size of the palette data format of Window 3 0 = 16-bit (5:6:5) 1 = 16-bit (A:5:5:5)	0
W2PAL	[6]	This bit determines the size of the palette data format of Window 2 0 = 16-bit (5:6:5) 1 = 16-bit (A:5:5:5)	0
W1PAL	[5:3]	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)	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)	0

14.6.49 I80 / RGB TRIGGER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
TRIGCON	0x771001A4	R/W	I80 / RGB Trigger Control Register	0x0000_0000

TRIGCON	Bit	Description	Initial State
reserved	[7:3]	Must be '0'	0
SWFRSTATUS	[2]	Frame Done Status [Read Only] 0 : Indicate I80 frame transfer is not finished 1 : Indicate I80 frame transfer finished * Clear Condition: Read or New Frame Start * Only when TRGMODE is '1'	0
SWTRGCMD	[1]	1 : Software Triggering Command [Write Only] * Only when TRGMODE is '1'	0
TRGMODE	[0]	Software Trigger enable control 0 : Disable 1 : Enable	0

14.6.50 ITU 601 INTERFACE CONTROL 0

Register	Address	R/W	Description	Reset Value
ITUIFCON0	0x771001A8	R/W	ITU (BT.601) Interface Control	0x0000_0000

I80IFCONAx	Bit	Description	Initial State
Reserved	[26:25]	Reserved	0
SELVSYNC	[24]	Select the VSYNC mode 0: Equal leading edge with HSYNC 1: Delayed VSYNC (standard)	
DLYVSYNC	[23:16]	Numbers of clock cycles for delay of the VSYNC signal (Only when SELVSYNC is '1') DLYSYNC+1 *over 64 cycles (standard)	0
Reserved	[15:7]	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 * In normal mode, when the frame is even field, VEN_FIELD signal is High.	0
I601CLK	[2]	The polarity of the V601_CLK active edge 0: normal 1: inverted	0
-	[1:0]	Reserved	0

14.6.51 LCD I80 INTERFACE CONTROL 0

Register	Address	R/W	Description	Reset Value
I80IFCONA0	0x771001B0	R/W	I80 Interface control for Main LDI(LCD)	0x0
I80IFCONA1	0x771001B4	R/W	I80 Interface control for Sub LDI(LCD)	0x0

I80IFCONAx	Bit	Description	Initial State
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: RS signal is low during video data transfer. 1: RS signal is high during video data transfer.	0
Reserved	[1]	Reserved	0
I80IFEN	[0]	LCD I80 Interface control 0: Disable 1: Enable	0

14.6.52 LCD I80 INTERFACE CONTROL 1

Register	Address	R/W	Description	Reset Value
I80IFCONB0	0x771001B8	R/W	I80 Interface control for Main LDI(LCD)	0x0000_0000
I80IFCONB1	0x771001BC	R/W	I80 Interface control for Sub LDI(LCD)	0x0000_0000

I80IFCONBx	Bit	Description	Initial State
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 (Do Not Skip) 01 : 2 10 : 3	00
Reserved	[4]	Reserved	0
AUTO_CMD_RATE	[3:0]	0000 : Disable auto command 0001 : per 2 Frames 0010 : per 4 Frames 0011 : per 6 Frames … 1111 : per 30 Frames	0000

14.6.53 LCD I80 INTERFACE COMMAND CONTROL 0

Register	Address	R/W	Description	Reset Value
LDI_CMDCON0	0x771001D0	R/W	I80 System Interface Command Control 0	0x0000_0000

LDI_CMDCON0	Bit	Description	Initial state
Reserved	[31:24]	Reserved	
CMD11_EN	[23:22]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD10_EN	[21:20]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD9_EN	[19:18]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD8_EN	[17:16]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD7_EN	[15:14]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD6_EN	[13:12]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD5_EN	[11:10]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD4_EN	[9:8]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD3_EN	[7:6]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD2_EN	[5:4]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD1_EN	[3:2]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00
CMD0_EN	[1:0]	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	00

14.6.54 LCD I80 INTERFACE COMMAND CONTROL 1

Register	Address	R/W	Description	Reset Value
LDI_CMDCON1	0x771001D4	R/W	I80 System Interface Command Control 1	0x0000_0000

LDI_CMDCON1	Bit	Description	Initial state
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

14.6.55 I80 SYSTEM INTERFACE MANUAL COMMAND CONTROL 0

Register	Address	R/W	Description	Reset Value
SIFCCON0	0x771001E0	R/W	I80 System Interface Manual Command Control	0x0000_0000

SIFCCON0	Bit	Description	Initial State
Reserved	[7]	Reserved	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: Disable (High) 1: Enable (Low)	0
SYS_nCS1_CON	[3]	LCD I80 System Interface nCS1 (sub) Signal control 0: Disable (High) 1: Enable (Low)	0
SYS_nOE_CON	[2]	LCD I80 System Interface nOE Signal control 0: Disable (High) 1: Enable (Low)	0
SYS_nWE_CON	[1]	LCD I80 System Interface nWE Signal control 0: Disable (High) 1: Enable (Low)	0
SCOMEN	[0]	LCD I80 System Interface Command Mode Enable 0: Disable (Normal Mode) 1: Enable (Manual Command Mode)	0

14.6.56 I80 SYSTEM INTERFACE MANUAL COMMAND CONTROL 1

Register	Address	R/W	Description	Reset Value
SIFCCON1	0x771001E4	R/W	I80 System Interface Manual Command Data Write Control	0x0000_0000

SIFCCON1	Bit	Description	Initial State
SYS_WDATA	[17:0]	LCD I80 System Interface Write Data Control	0

14.6.57 I80 SYSTEM INTERFACE MANUAL COMMAND CONTROL 2

Register	Address	R	Description	Reset Value
SIFCCON2	0x771001E8	R	I80 System Interface Manual Command Data Read Control	undefined

SIFCCON2	Bit	Description	Initial State
SYS_RDATA	[17:0]	LCD I80 System Interface Read Data Control	undefined

14.6.58 LCD I80 INTERFACE COMMAND

Register	Address	R/W	Description	Reset Value
LDI_CMD0	0x77100280	R/W	I80 Interface Command 0	0x0000_0000
LDI_CMD1	0x77100284	R/W	I80 Interface Command 1	0x0000_0000
LDI_CMD2	0x77100288	R/W	I80 Interface Command 2	0x0000_0000
LDI_CMD3	0x7710028c	R/W	I80 Interface Command 3	0x0000_0000
LDI_CMD4	0x77100290	R/W	I80 Interface Command 4	0x0000_0000
LDI_CMD5	0x77100294	R/W	I80 Interface Command 5	0x0000_0000
LDI_CMD6	0x77100298	R/W	I80 Interface Command 6	0x0000_0000
LDI_CMD7	0x7710029c	R/W	I80 Interface Command 7	0x0000_0000
LDI_CMD8	0x771002A0	R/W	I80 Interface Command 8	0x0000_0000
LDI_CMD9	0x771002A4	R/W	I80 Interface Command 9	0x0000_0000
LDI_CMD10	0x771002A8	R/W	I80 Interface Command 10	0x0000_0000
LDI_CMD11	0x771002AC	R/W	I80 Interface Command 11	0x0000_0000

I80IFCONx	Bit	Description	Initial State
LDI_CMD	[17:0]	LDI command	0

14.6.59 WINDOW 2'S PALETTE DATA

Register	Address	R/W	Description	Reset Value
W2PDATA01	0x77100300	R/W	Window 2 Palette Data of the Index 0,1	0x0000_0000
W2PDATA23	0x77100304	R/W	Window 2 Palette Data of the Index 2,3	0x0000_0000
W2PDATA45	0x77100308	R/W	Window 2 Palette Data of the Index 4,5	0x0000_0000
W2PDATA67	0x7710030C	R/W	Window 2 Palette Data of the Index 6,7	0x0000_0000
W2PDATA89	0x77100310	R/W	Window 2 Palette Data of the Index 8,9	0x0000_000

				0
W2PDATAAB	0x77100314	R/W	Window 2 Palette Data of the Index A,B	0x0000_0000
W2PDATACD	0x77100318	R/W	Window 2 Palette Data of the Index C,D	0x0000_0000
W2PDATAEF	0x7710031C	R/W	Window 2 Palette Data of the Index E,F	0x0000_0000

W3PDATAxx	Bit	Description	Initial State
ODD_VAL	[31:16]	Lut Value register	0
EVEN_VAL	[15:0]	Lut Value register	0

14.6.60 WINDOW 3'S PALETTE DATA

Register	Address	R/W	Description	Reset Value
W3PDATA01	0x77100320	R/W	Window 3 Palette Data of the Index 0,1	0x0000_0000
W3PDATA23	0x77100324	R/W	Window 3 Palette Data of the Index 2,3	0x0000_0000
W3PDATA45	0x77100328	R/W	Window 3 Palette Data of the Index 4,5	0x0000_0000
W3PDATA67	0x7710032C	R/W	Window 3 Palette Data of the Index 6,7	0x0000_0000
W3PDATA89	0x77100330	R/W	Window 3 Palette Data of the Index 8,9	0x0000_0000
W3PDATAAB	0x77100334	R/W	Window 3 Palette Data of the Index A,B	0x0000_0000
W3PDATACD	0x77100338	R/W	Window 3 Palette Data of the Index C,D	0x0000_0000
W3PDATAEF	0x7710033C	R/W	Window 3 Palette Data of the Index E,F	0x0000_0000

W3PDATAxx	Bit	Description	Initial State
ODD_VAL	[31:16]	Lut Value register	0
EVEN_VAL	[15:0]	Lut Value register	0

14.6.61 WINDOW 4'S PALETTE DATA

Register	Address	R/W	Description	Reset Value
W4PDATA01	0x77100340	R/W	Window 4 Palette Data of the Index 0,1	0x0000_0000
W4PDATA23	0x77100344	R/W	Window 4 Palette Data of the Index 2,3	0x0000_0000

W4DATAxx	Bit	Description	Initial State
ODD_VAL	[31:16]	Lut Value register	0
EVEN_VAL	[15:0]	Lut Value register	0

14.6.62 WIN0 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Index	Address	R/W	Description	Reset Value
WIN0_PALENTRY0	0x77100400	R/W	Window 0 Palette entry 0 address	undefined

WIN0_PALENTRY1	0x77100404	R/W	Window 0 Palette entry 1 address	undefined
-	-	-	-	-
WIN0_PALENTRY255	0x771007FC	R/W	Window 0 Palette entry 255 address	undefined

14.6.63 WIN1 PALETTE RAM ACCESS ADDRESS (NOT SFR)

Index	Address	R/W	Description	Reset Value
WIN0_PALENTRY0	0x77100800	R/W	Window 1 Palette entry 0 address	undefined
WIN0_PALENTRY1	0x77100804	R/W	Window 1 Palette entry 1 address	undefined
-	-	-	-	-
WIN0_PALENTRY255	0x77100BFC	R/W	Window 1 Palette entry 255 -address	undefined

15 POST PROCESSOR

This chapter describes the functions and usage of the Postprocessor interface of the S3C6410X.

15.1 OVERVIEW

The Post processor administers the video/graphic scale, video format conversion and color space conversion. It is composed of Data-Path, DMA controller and Register files as shown in the Figure 15-1.

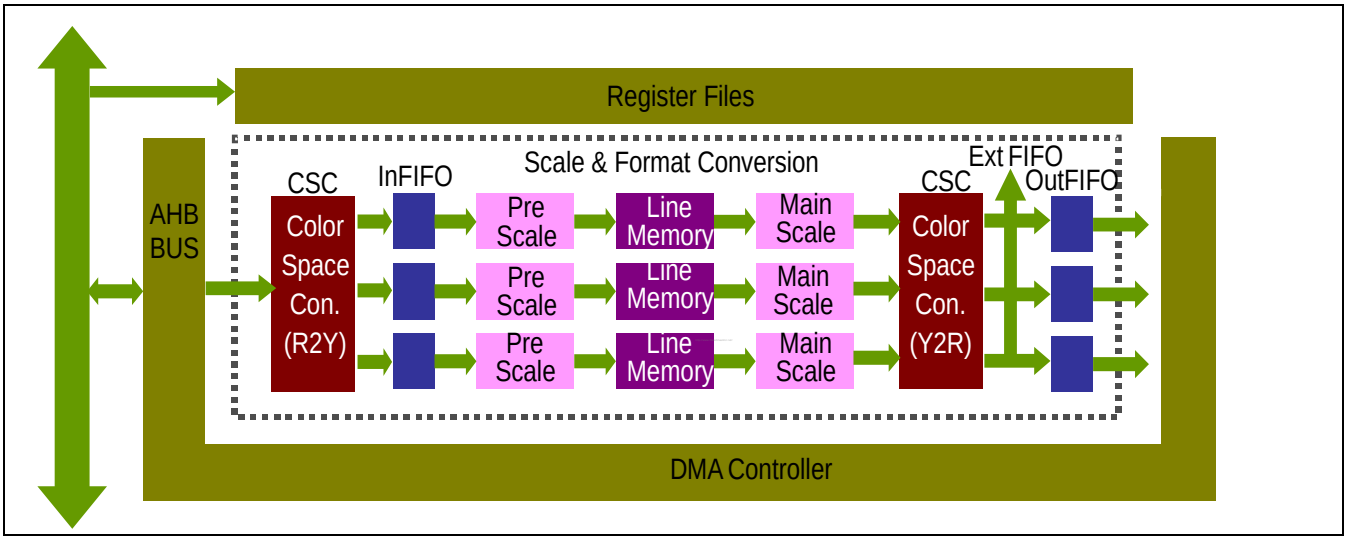


Figure 15-1. Block Diagram of Post Processor

15.2 FEATURES

- Dedicated DMA with offset address
- 3 Channel scaling pipelines for video/graphic scaling up/down or zooming in/out
- Video input format: 420, 422 format
- Graphic input format: 16-bit (565format) or 24-bit
- Graphics Output format to Memory: 16-bit (565 format) / 24-bit graphic data (progressive only)
- Video Output format to Memory: YCbCr420, YCbCr422
- Output format to external FIFO: YCbCr444 / RGB (30-bit) for interlace and progressive
- FreeRun Mode Operation
- Programmable source image size up to 4096 × 4096 resolution
- Programmable destination image size up to 2048 × 2048 resolution
- Programmable scaling ratio
- Format conversion for video signals
- Color space conversion from YCbCr to RGB
- Color Space conversion from RGB to YCbCr

15.3 A SOURCE AND DESTINATION IMAGE DATA FORMAT

In FIMV POST Processor, two output modes such as DMA mode and FIFO mode are available as shown in the following Figure 15-2.

In FIFO mode (if LCDPathEnable = 1, For more information refer to chapter 15.7 Register File List), destination image is transferred to the FIFO in display controller (or some other IP with FIFO interface) without additional memory bandwidth such as POST-to-Memory and Memory-to-Display Controller. The source image format and the destination image format is described in Chapter 15.3.2 FIFO mode.

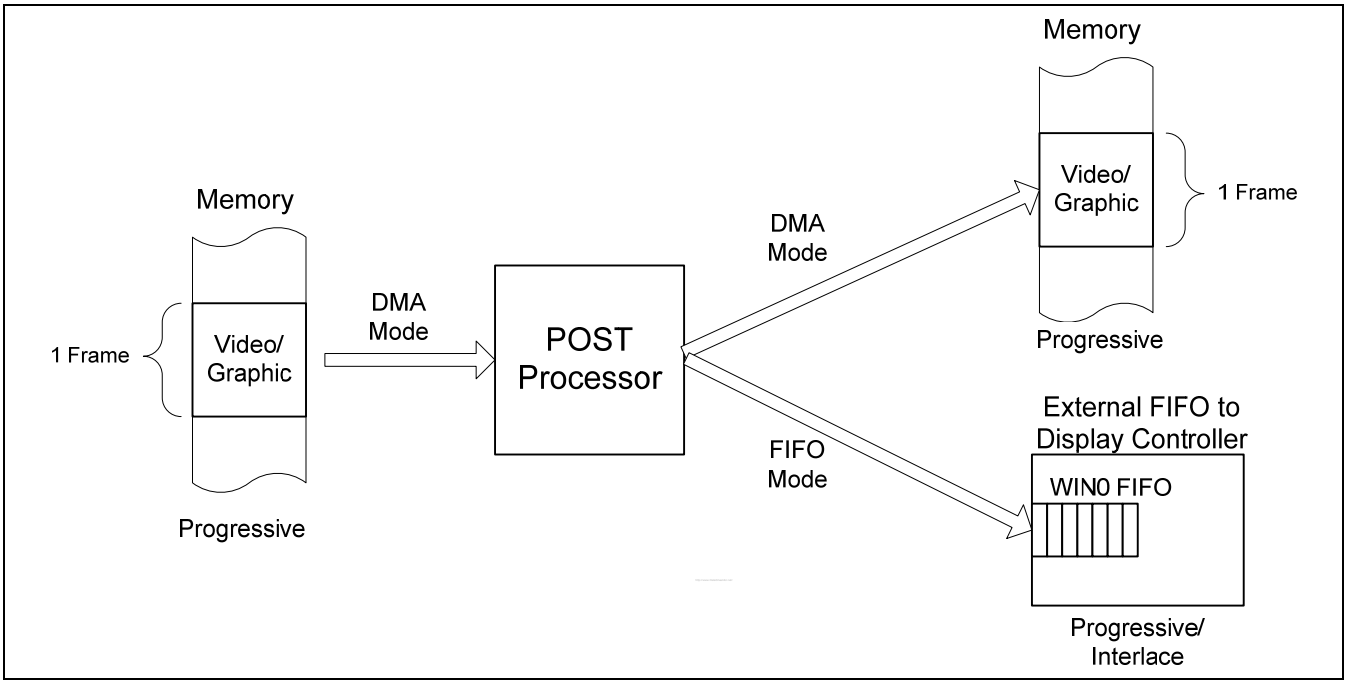


Figure 15-2. Input and Output modes in POST Processor Block Diagram

15.3.1 DMA MODE OPERATION

Various source and destination image formats can be selected according to the mode configuration as described in Table 15-1 a). Source image format is one of the following YCbCr420, YCbCr422, RGB16-bit (565format) and RGB 24-bit format. Destination image format is identical to the source image format such as YCbCr420, YCbCr422, RGB 16-bit (565format) and RGB 24-bit format. The detail control signal is defined in Table15-1 b)

In the case of YCbCr420 source/destination image format, each component of Y, Cb and Cr is stored in each own separated address space without any interleaving as shown in Case A of Figure 15-3 (a) and Figure 15-4. In the other cases, either byte or half-word interleaving is applied within unified address space as described in Figure 15-3 (b). Byte interleaving order of YCbCr422 source image can be selected either from YCbYCr or CbYCrY as shown in case B and C of Figure 15-3 (b) and Figure 15-4. Byte order of RGB 24-bit and half-word order of RGB 16-bit are described in case D and E of Figure 15-3 (b) and Figure 15-4.

In both cases of YCbCr420 and YCbCr422 source image format, whether MPEG4 format or MPEG2/H.263 format needs to be selected according to the sampling position of the chroma information as shown in Figure 15-4. Destination image have the same sampling position of source image when both source and destination image are YCbCr format. If source image is RGB format and destination image is YCbCr format, the sampling position of chroma component is the same position of Luma.

All source and destination image data must be stored in memory system aligned with word boundary. It means that neither byte nor half-word size DMA operations are supported *(For more information refer to section 15-5 for DMA operation)*. Therefore, the width of source and destination image must be selected to satisfy the word boundary condition *(For more information refer to section 15-4 for image size)*.

Table 15-1 a). Mode configuration for video/graphic source format and the corresponding data format

MODE[8] SRC420	MODE[3] InRGB	MODE[2] INTER-LEAVE	MODE[1] InRGB Format	MODE[15][0] InYCbCr Format	Description	
					Video/Graphic Format	Data Format in Figure15-3 and 4
1	0	0	1	X	420 YCbCr Format	A
0	0	1	1	00/10	422 YCbYCr Format	B/B’
0	0	1	1	01/11	422 CbYCrY Format	C/C’
0	1	1	1	X	RGB 24-bit true color	D
0	1	1	0	X	RGB 16-bit Format	E

Table 15-1 b). Mode configuration for video/graphic destination format and the corresponding data format

MODE[18] OutRGB	MODE[17] DST420	MODE[20][19] OutYCbCr Format	MODE[4] OutRGB Format	Description	
				Video/Graphic Format	Data Format in Figure15-3 and 4
0	1	X	X	420 YCbCr Format	A
0	0	00/10	X	422 YCbYCr Format	B/B’
0	0	01/11	X	422 CbYCrY Format	C/C’
1	X	X	1	RGB 24-bit true color	D
1	X	X	0	RGB 16-bit Format	E

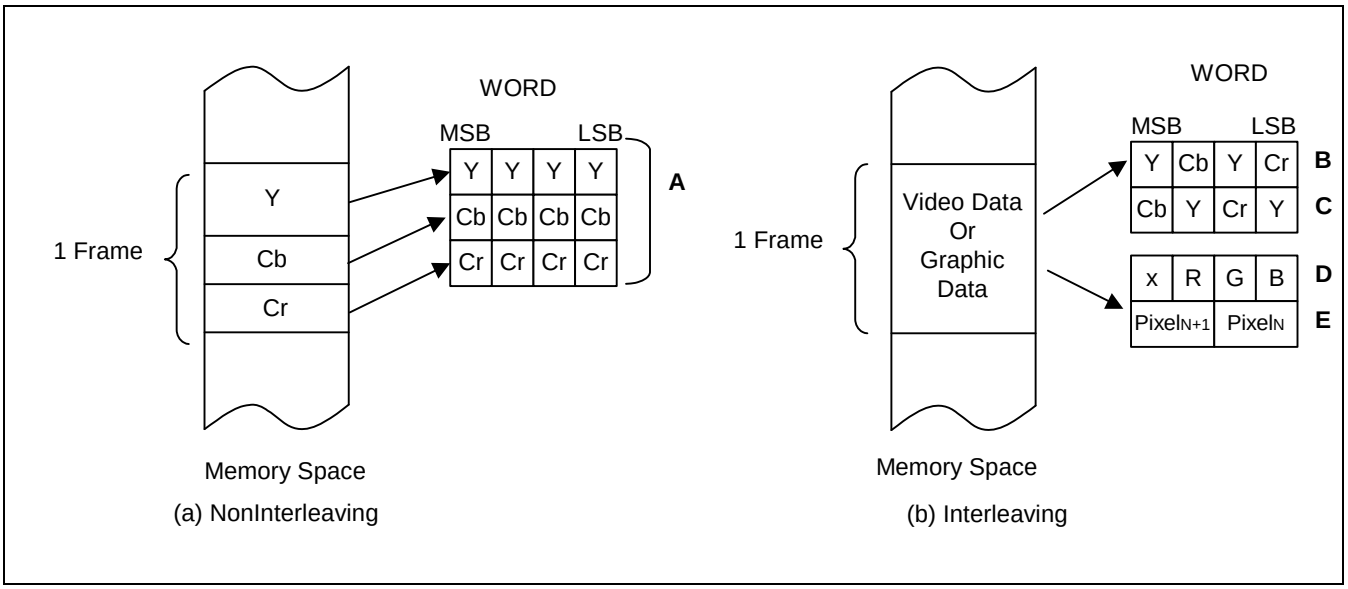
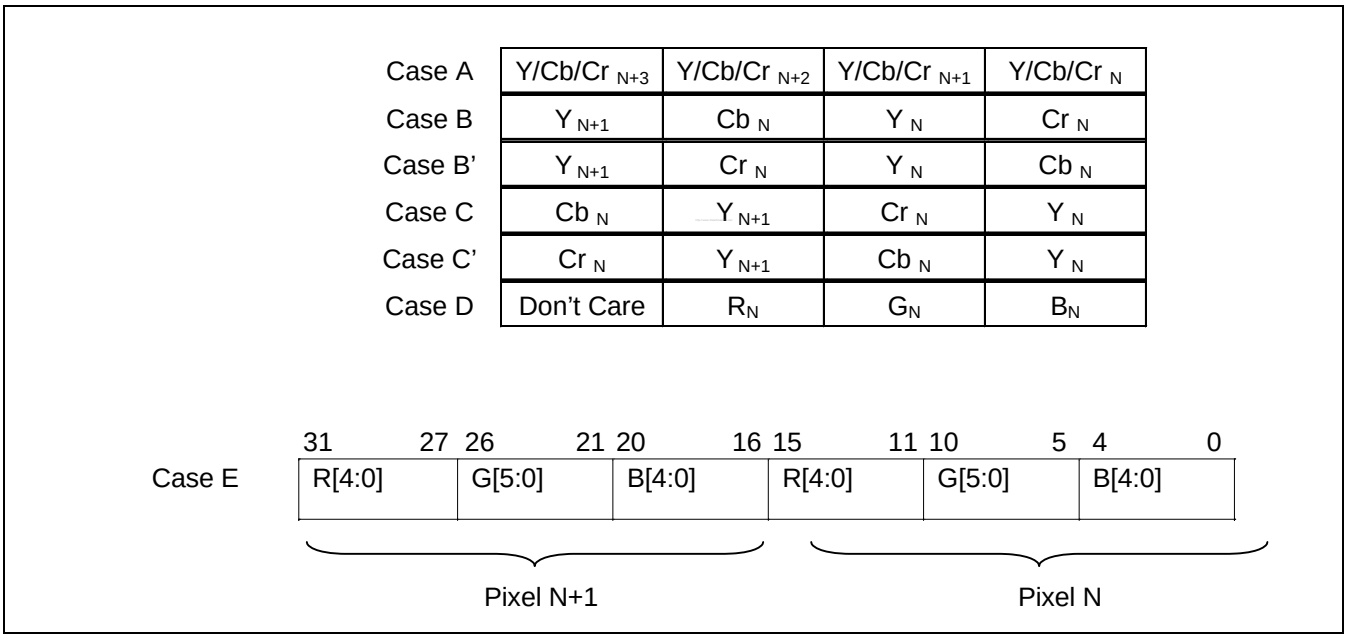


Figure 15-3. Data format stored in external



Case A	Y/Cb/Cr $_{N+3}$	Y/Cb/Cr $_{N+2}$	Y/Cb/Cr $_{N+1}$	Y/Cb/Cr $_{N}$
Case B	Y $_{N+1}$	Cb $_{N}$	Y $_{N}$	Cr $_{N}$
Case B'	Y $_{N+1}$	Cr $_{N}$	Y $_{N}$	Cb $_{N}$
Case C	Cb $_{N}$	Y $_{N+1}$	Cr $_{N}$	Y $_{N}$
Case C'	Cr $_{N}$	Y $_{N+1}$	Cb $_{N}$	Y $_{N}$
Case D	Don't Care	R_N	G_N	B_N

Figure 15-4. Byte and half-word organization Block Diagram

15.3.2 FIFO MODE OPERATION

Output data format is determined by MODE[18] as described in Table 15-1 b). The output data format is fixed to 30-bit data, 10-bit per each component RGB or YCbCr444. The other specific mode configuration signals mentioned in Table 15-1 b) are ignored if LCDPathEnable is set to “1”.

Table 15-2. Output Data Format in FIFO Mode

OutRGB (MODE[18])	Output Data Format (LCDPathEnable = 1) (Progressive / Interlace)
0	YCbCr 444
1	RGB 30-bit

In Output FIFO mode, either progressive or interlace scan mode can be selected according to “interlace” control register as defined in *section 15.7 Register Files Lists*. The “interlace” control bit is enabled only if LCDPathEnable = 1, otherwise its value is unaffected to DMA mode operation which support only progressive.

Even if an interlaced scan mode is enabled (LCDPathEnable = 1 and Interlace = 1), per frame management, which consists of even field and odd filed, operates automatically. 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.

15.4 IMAGE SIZE AND SCALE RATIO

The RGB graphic source image size is determined by number of pixels adjacent to horizontal and vertical directions. YCbCr420 and YCbCr422 source image size is determined only by numbers of Y samples adjacent to horizontal and vertical directions. Destination image size is determined by dimension of final RGB graphic image, after color space conversion if source image is YCbCr image.

As explained in the previous section, SRC_Width and DST_Width satisfies the word boundary constraints in such a way that the number of horizontal pixel can be represented by kn where n = 1,2,3, … and k = 1 / 2 / 8 for 24bppRGB / 16bppRGB / YCbCr420 image, respectively. Also SRC_Width must be 4’s multiple of PreScale_H_Ratio and SRC_Height must be 2’s multiple of PreScale_V_Ratio.

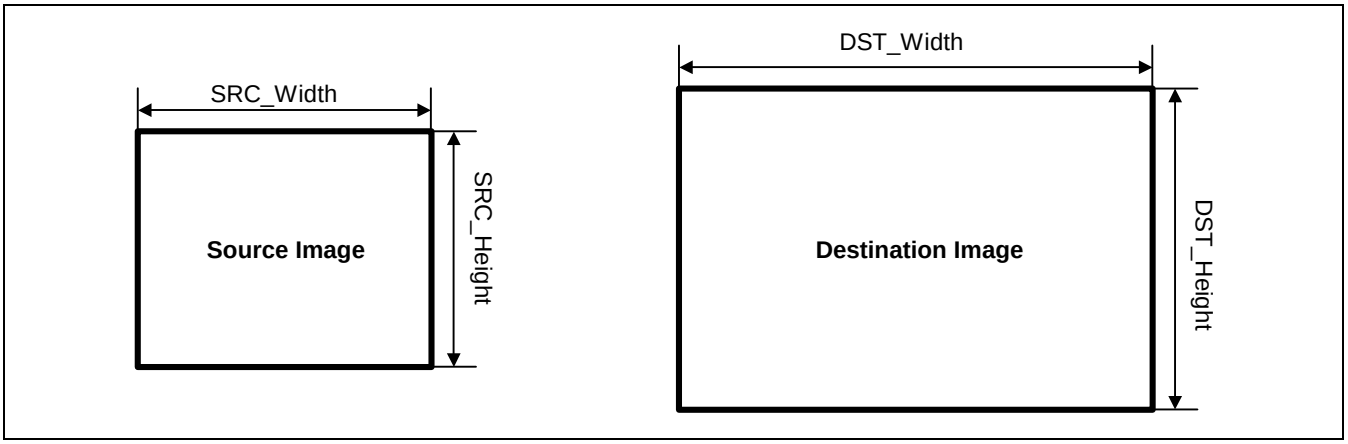


Figure 15-5. Source and Destination image size Block Diagram

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) { PreScale_H_Ratio = 32;    H_Shift = 5; }
else if (SRC_Width >= 16 × DST_Width) { PreScale_H_Ratio = 16;    H_Shift = 4; }
else if (SRC_Width >= 8 × DST_Width) { PreScale_H_Ratio = 8;    H_Shift = 3; }
else if (SRC_Width >= 4 × DST_Width) { PreScale_H_Ratio = 4;    H_Shift = 2; }
else if (SRC_Width >= 2 × DST_Width) { PreScale_H_Ratio = 2;    H_Shift = 1; }
else { PreScale_H_Ratio = 1;    H_Shift = 0; }

PreScale_DSTWidth = SRC_Width / PreScale_H_Ratio;
dx = ( 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) { PreScale_V_Ratio = 32;    V_Shift = 5; }
else if (SRC_Height >= 16 × DST_Height) { PreScale_V_Ratio = 16;    V_Shift = 4; }
else if (SRC_Height >= 8 × DST_Height) { PreScale_V_Ratio = 8;    V_Shift = 3; }
else if (SRC_Height >= 4 × DST_Height) { PreScale_V_Ratio = 4;    V_Shift = 2; }
else if (SRC_Height >= 2 × DST_Height) { PreScale_V_Ratio = 2;    V_Shift = 1; }
else { PreScale_V_Ratio = 1;    V_Shift = 0; }

PreScale_DSTHeight = SRC_Height / PreScale_V_Ratio;
dy = ( SRC_Height << 8 ) / ( DST_Height << V_Shift);

PreScale_SHFactor = 10 – ( H_Shift + V_Shift);
```

15.5 DMA OPERATION OF SOURCE AND DESTINATION IMAGE

There are three address categories such as start address, end address and offset address for DMA operation. Each address category consists of three source address components of Y/Cb/Cr and three destination address component of RGB/oCb/oCr. If a source image is stored in the non-interleaved format such as YCbCr420, all source address components are valid as shown in Figure 15-6 (a). If a source image is stored by the interleaved format such as a RGB graphic format or an YCbCr422 format, only Y component of three source components is valid and two chroma address components are invalid as shown in Figure 15-6 (b). If a destination image is stored in the non-interleaved format such as YCbCr420, all source address components RGB/oCb/oCr are valid as shown in Figure 15-6 (a). If a source image is stored by the interleaved format such as a RGB graphic format or an YCbCr422 format, only RGB component of three source components is valid and two chroma address components of oCb/oCr are invalid as shown in Figure 15-6 (b).

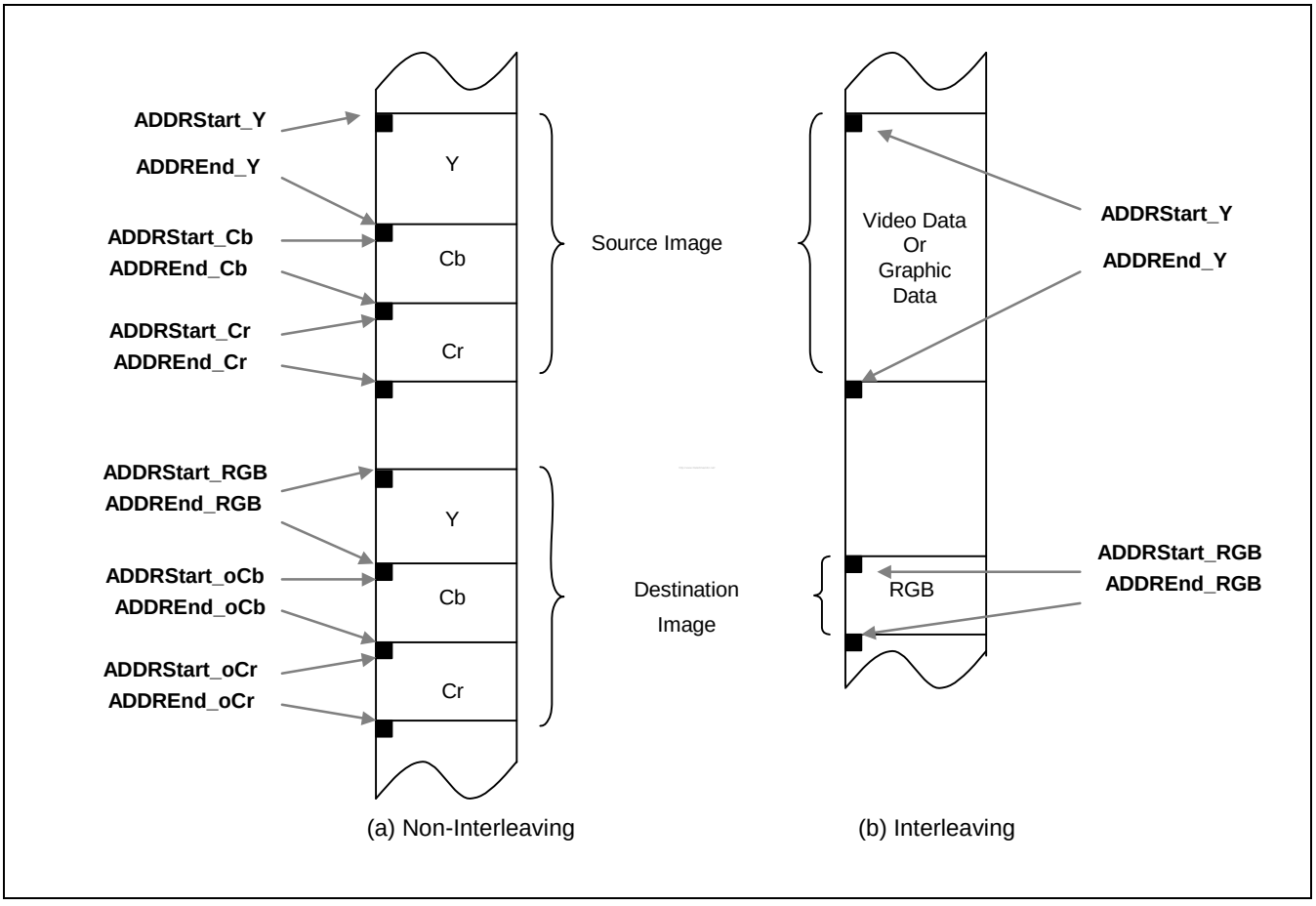


Figure 15-6. Start and end address set according to memory allocation type Block Diagram

15.5.1 START ADDRESS

Start address of ADDRStart_Y/Cb/Cr/RGB/oCb/oCr points the first word address where the corresponding component of Y/Cb/Cr/RGB/oCb/oCr is read or written. Each one must be aligned with word boundary (i.e. ADDRStart_X[1:0] = 00). ADDRStart_Cb and ADDRStart_Cr are valid only for the YCbCr420 source image format. ADDRStart_oCb and ADDRStart_oCr are valid only for the YCbCr420 destination image format.

15.5.2 END ADDRESS

< Source End Address >
ADDREnd_Y
= ADDRStart_Y + Memory size for the component of Y/RGB
= ADDRStart_Y + (SRC_Width × SRC_Height) × ByteSize_Per_Pixel + Offset_Y × (SRC_Height-1)

ADDREnd_Cb (Valid for YCbCr420 source format)
= ADDRStart_Cb + Memory size for the component of Cb
= ADDRStart_Cb + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cb × (SRC_Height/2-1)

ADDREnd_ Cr (Valid for YCbCr420 source format)
= ADDRStart_ Cr + Memory size for the component of Cr
= ADDRStart_Cr + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cr × (SRC_Height/2-1)

< Destination End Address >
ADDREnd_RGB
= ADDRStart_RGB + Memory size for the component of RGB data (or component of Y)
= ADDRStart_RGB + (DST_Width × DST_Height) × ByteSize_Per_Pixel + Offset_RGB × (DST_Height-1)

ADDREnd_oCb (Valid for YCbCr420 destination format)
= ADDRStart_oCb + Memory size for the component of Cb
= ADDRStart_oCb + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cb × (SRC_Height/2-1)

ADDREnd_ oCr (Valid for YCbCr420 destination format)
= ADDRStart_ oCr + Memory size for the component of Cr
= ADDRStart_oCr + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cr × (SRC_Height/2-1)

Where,
Offset_Y/Cb/Cr/RGB
= Memory size for offset per a horizontal line
= Number of pixel (or sample) in horizontal offset × ByteSize_Per_Pixel (or Sample)

$$\text{ByteSize\_Per\_Pixel} = \begin{cases} 1 \text{ for YCbCr420} \\ 2 \text{ for 16-bit RGB and YcbCr422} \\ 4 \text{ for 24-bit RGB} \end{cases}$$

Offset address is used for the following two situations. One is to fetch some parts of source image in order to zoom in/out as shown in Figure 15-7 (a). The other is to restore destination image for PIP (picture-in-picture) applications as shown in Figure 15-7 (b). The word boundary constraints must be satisfied in both cases.

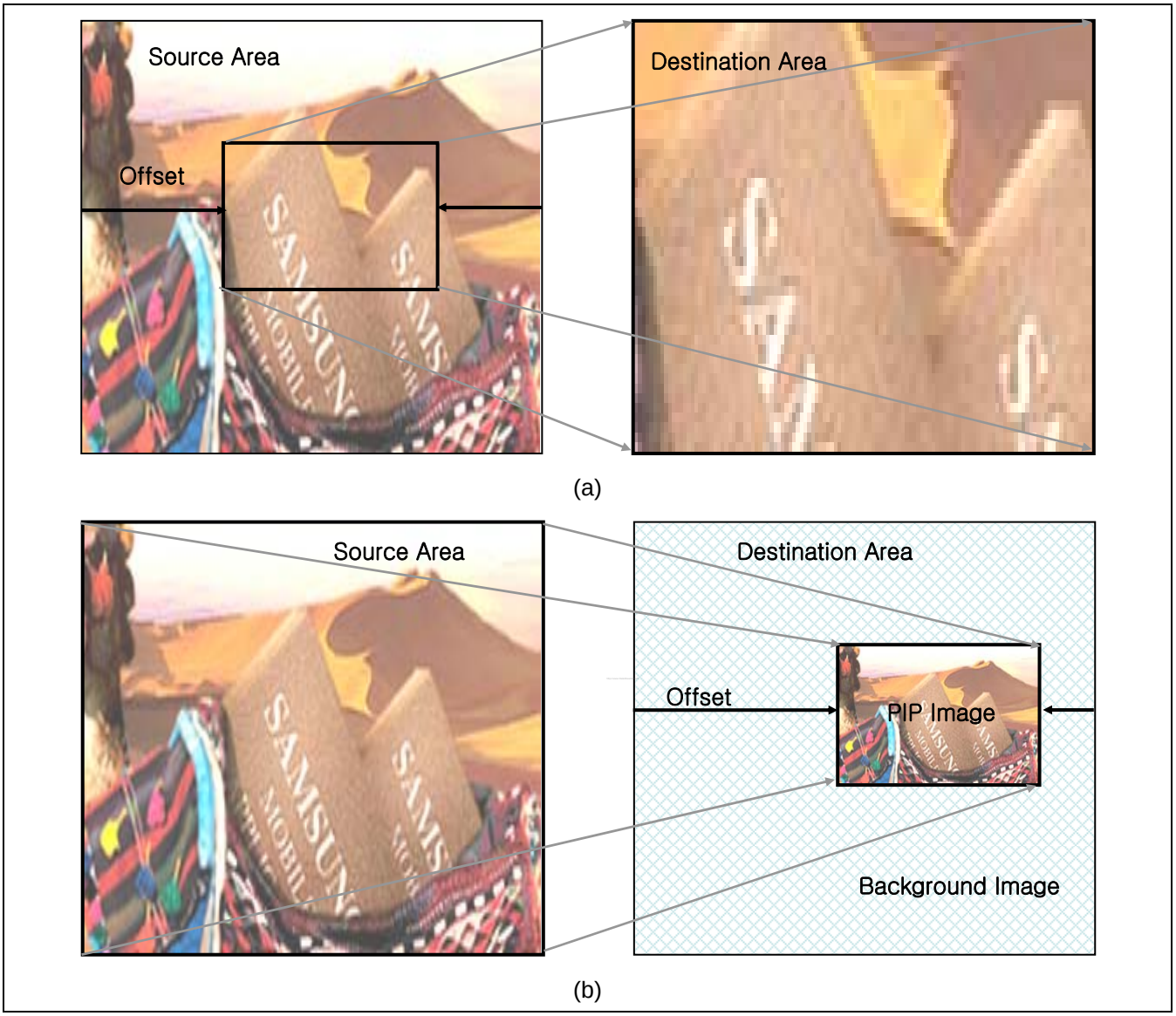


Figure 15-7. Offset address for (a) Source image for zoom in/out operation and (b) Destination image for PIP applications

15.6 FRAME MANAGEMENT OF POST PROCESSOR

15.6.1 PER FRAME MANAGEMENT MODE

Per frame management of POST-Processor are controlled by two controls register such as POSTENVID and POSTINT as shown in Figure 15-8. “POSTENVID” triggers the operation of POST PROCESSOR. It is automatically de-asserted when all operations of the given frame are completed. Before asserting “POSTENVID”, all control registers must be set to the proper value as explained in the previous chapters. When all operations are completed, interrupt pending register is asserted (POSTINT=1), if the interrupt enable signal is asserted (INTEN=1). The POSTINT signal, directing to the interrupt controller, must be cleared by the interrupt service routine. The polling POSTENVID is used to detect the end of the operation.

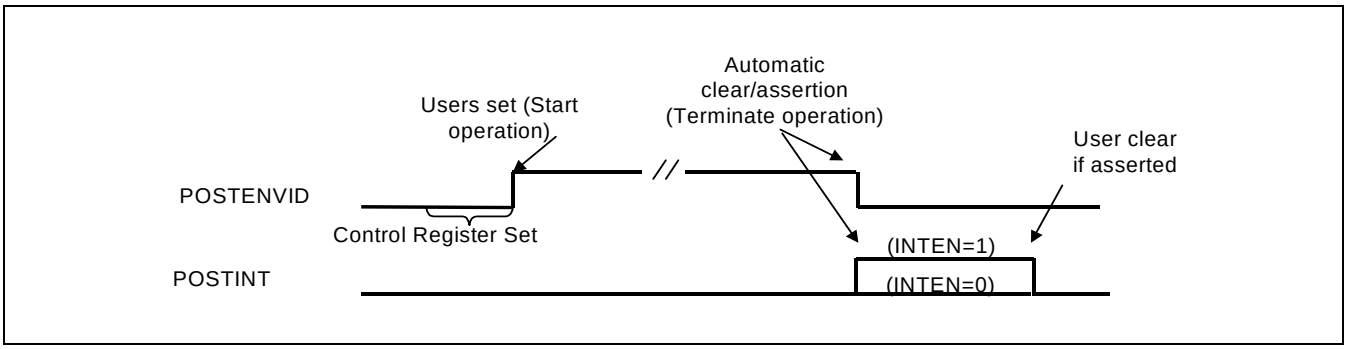


Figure 15-8. Start and termination of POST PROCESSOR operation (AutoLoadEnable = 0) Block Diagram

15.6.2 FREE RUN MODE

To activate the new frame management scheme of free run operation, you must set “AutoLoadEnable” bit to 1. In this mode, user can pre-define the next frame-related address set of NxtADDRXXX (defined in chapter 15.7 Register files) even when the current frame is under operation. When the current frame is completely finished, the following operations are executed step-by-step.

1. According to INTEN, interrupt signal is asserted or not.
2. Next frame address set of NxtADDRXXX is copied into the current frame address set of ADDRXXX.
3. ENVID is automatically asserted and the next frame operation starts up.

** In Free Run Mode, the other register value except the address stuff must remain the same value between the current frame and the next frame.

15.7 REGISTER FILE LISTS

Register	Address	R/W	Description	Reset Value
MODE	0x77000000	R/W	Mode Register	0x00070B12
PreScale_Ratio	0x77000004	R/W	Pre-Scale ratio for vertical and horizontal	0x0
PreScaleImgSize	0x77000008	R/W	Pre-Scaled image size	0x0
SRCImgSize	0x7700000C	R/W	Source image size	0x0
MainScale_H_Ratio	0x77000010	R/W	Main scale ratio along to horizontal direction	0x0
MainScale_V_Ratio	0x77000014	R/W	Main scale ratio along to vertical direction	0x0
DSTImgSize	0x77000018	R/W	Destination image size	0x0
PreScale_SHFactor	0x7700001C	R/W	Pre-scale shift factor	0x0
ADDRStart_Y	0x77000020	R/W	DMA (Buffer 0) Start address for source Y or RGB component	0x20000000
ADDRStart_Cb	0x77000024	R/W	DMA (Buffer 0) Start address for source Cb component	0x20000000
ADDRStart_Cr	0x77000028	R/W	DMA (Buffer 0) Start address for source Cr component	0x20000000
ADDRStart_RGB	0x7700002C	R/W	DMA (Buffer 0) Start address for destination Y or RGB component	0x20000000
ADDREnd_Y	0x77000030	R/W	DMA (Buffer 0) End address for source Y or RGB component	0x20006300
ADDREnd_Cb	0x77000034	R/W	DMA (Buffer 0) End address for source Cb component	0x20006300
ADDREnd_Cr	0x77000038	R/W	DMA (Buffer 0) End address for source Cr component	0x20006300
ADDREnd_RGB	0x7700003C	R/W	DMA (Buffer 0) End address for destination Y or RGB component	0x20006300
Offset_Y	0x77000040	R/W	Offset of Y or RGB component for fetching source image	0x0
Offset_Cb	0x77000044	R/W	Offset of Cb component for fetching source image	0x0
Offset_Cr	0x77000048	R/W	Offset of Cr component for fetching source image	0x0
Offset_RGB	0x7700004C	R/W	Offset of Y or RGB component for restoring destination image	0x0
RESERVED	0x77000050	-	-	-
NxtADDRStart_Y	0x77000054	R/W	Next Frame (Buffer 1) DMA Start address for source Y or RGB component	0x20000000
NxtADDRStart_Cb	0x77000058	R/W	Next Frame (Buffer 1) DMA Start address for source Cb component	0x20000000
NxtADDRStart_Cr	0x7700005C	R/W	Next Frame (Buffer 1) DMA Start address for source Cr component	0x20000000
NxtADDRStart_RGB	0x77000060	R/W	Next Frame (Buffer 1) DMA Start address for destination Y or RGB component	0x20000000
NxtADDREnd_Y	0x77000064	R/W	Next Frame (Buffer 1) DMA End address for source Y or RGB component	0x20006300

Register	Address	R/W	Description	Reset Value
NxtADDREnd_Cb	0x77000068	R/W	Next Frame (Buffer 1) DMA End address for source Cb component	0x20006300
NxtADDREnd_Cr	0x7700006C	R/W	Next Frame (Buffer 1) DMA End address for source Cr component	0x20006300
NxtADDREnd_RGB	0x77000070	R/W	Next Frame (Buffer 1) DMA End address for destination Y or RGB component	0x20006300
ADDRStart_oCb	0x77000074	R/W	DMA (Buffer 0) Start address for destination Cb component	0x20000000
ADDRStart_oCr	0x77000078	R/W	DMA (Buffer 0) Start address for destination Cr component	0x20000000
ADDREnd_oCb	0x7700007C	R/W	DMA (Buffer 0) End address for destination Cb component	0x20000000
ADDREnd_oCr	0x77000080	R/W	DMA (Buffer 0) End address for destination Cr component	0x20000000
Offset_oCb	0x77000084	R/W	Offset of Cb component for fetching destination image	0x0
Offset_oCr	0x77000088	R/W	Offset of Cr component for fetching destination image	0x0
NxtADDRStart_oCb	0x7700008C	R/W	Next Frame DMA (Buffer 1) Start address for destination Cb component	0x20006300
NxtADDRStart_oCr	0x77000090	R/W	Next Frame DMA (Buffer 1) Start address for destination Cr component	0x20006300
NxtADDREnd_oCb	0x77000094	R/W	Next Frame DMA (Buffer 1) End address for destination Cb component	0x20006300
NxtADDREnd_oCr	0x77000098	R/W	Next Frame DMA (Buffer 1) End address for destination Cr component	0x20006300
POSTENVID	0x7700009C	R/W	Enable Video Processing.	0x0
MODE_2	0x770000A0	R/W	Mode Register 2	0x0

15.7.1 MODE CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MODE	0x77000000	R/W	Mode Register [31:0]	0x00070B12

MODE	Bit	Description	Initial State
reserved	[31]	Must be “0”	0
CLKVALUP	[30]	Select CLKVAL_F update timing control 0 = always 1 = start of a frame (only once per frame)	0
CLKVAL_F	[29:24]	Determine the rates of TSCLK and CLKVAL[5:0] TSCLK = Clock source / (CLKVAL+1), where CLKVAL >= 1 Note. The maximum frequency of VCLK is 66MHz.	0
CLKDIR	[23]	Select the clock source as direct or divide using CLKVAL_F register 0 = Direct clock (frequency of TSCLK = frequency of Clock source) 1 = Divided by CLKVAL_F	0
CLKSEL_F	[22:21]	Select the Video Clock source 00 = HCLK 01 = Clock selected by CLK_SRC[27:26] (refer to Chapter3. System Controller) 10 = reserved 11 = 27MHz Ext Clock input	0
OutYCbCrFormat	[20:19]	It determines the byte organization of word data when the destination image is interleaved YCbCr format. For more information refer to Table.15-1b) .	0
OutRGB	[18]	It indicates the output color space of destination image. 0 for YCbCr or 1 for RGB.	1
DST420	[17]	0 for YCbCr422 and 1 for YCbCr420 destination format. It is valid only for YCbCr destination image (i.e. OutRGB = 0)	1
R2YSel	[16]	Select color space conversion equation from RGB to YCbCr. 1 for YCbCr Wide range and 0 for YCbCr Narrow range.	1
InYCbCrFormat_MSB	[15]	It determines the MSB of byte organization of word data when the source image is interleaved YCbCr format. For more information refer to Table.15-1a).	0
AutoLoadEnable	[14]	AutoLoadEnalbe. 0 for Per Frame mode and 1 for Free Run mode	0
LCDPathEnable	[13]	Out FIFO Mode Enable. 1 for FIFO mode and 0 for DMA mode	0
Interlace	[12]	Output scan method selection register only when FIFO mode (LCDPathEnable =1). 1 for Interlace scan and 0 for progressive scan. In DMA mode (LCDPathEnable = 0), progressive scan is executed irrespective of the value given.	0

MODE	Bit	Description	Initial State
Wide/Narrow	[11:10]	Select color space conversion equation from YCbCr to RGB according to input value range. 2'10 for YCbCr Wide range and 2'01 for YCbCr Narrow range	2'b10
SRC420	[8]	0 for YCbCr422 and 1 for YCbCr420 source format. It is valid only for YCbCr source image (i.e. InRGB = 0)	1
INTEN	[7]	Interrupt Enable. It determines whether the POSTINT signal is asserted or not, when the processing of the current frame is finished. 0: disable, 1 : enable.	0
POSTINT	[6]	Interrupt Pending Bit. If INTEN is enabled, it is automatically asserted right after finishing operation of the current frame. It must be cleared by interrupt service routine. 0: disable, 1: enable.	0
IRQ_LEVEL	[5]	It determines the interrupt generation scheme. 1 for level interrupt. (must be 1)	0
OutRGBFormat	[4]	It determines the output format of destination image. 0 for 16-bit (565 format) RGB and 1 for 24-bit RGB.	1
InRGB	[3]	It indicates the input color space of source image. 0 for YCbCr or 1 for RGB.	0
INTERLEAVE	[2]	It indicates the data format of YCbCr. 0 for Non-Interleaved format (Each component of Y, Cb and Cr is access by the word). 1 for Interleaved format (All components of Y, Cb and Cr are mixed inside single word). It should be 1 when source image is RGB data (or InRGB =1).	0
InRGBFormat	[1]	If the source image is in RGB color space (or InRGB=1), it indicates the data format of graphic image. 0 for 16-bit (565 format) and 1 for 24-bit. Otherwise (or InRGB=0), it should be remains to 1.	1
InYCbCrFormat_LSB	[0]	It determines the LSB of byte organization of word data when the source image is interleaved YCbCr format. For more information refer to Table.15-1a).	0

Caution:: [9] bit is reserved.

15.7.2 Pre-Scale Ratio Register

Register	Address	R/W	Description	Reset Value
PreScale_Ratio	0x77000004	R/W	Pre-Scale ratio for vertical and horizontal.	0x0

PreScale_Ratio	Bit	Description	Initial State
PreScale_V_Ratio	[13:7]	Pre-scale ratio for vertical direction (For more information refer to chapter 15-4)	0x0
PreScale_H_Ratio	[6:0]	Pre-scale ratio for horizontal direction (For more information refer to chapter 15-4)	0x0

15.7.3 PRE-SCALE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
PreScaleImgSize	0x77000008	R/W	Pre-Scaled image size	0x0

PreScaleImgSize	Bit	Description	Initial State
PreScale_DSTHeight	[23:12]	Pre-Scaled image height (For more information refer to chapter 15-4)	0x0
PreScale_DSTWidth	[11:0]	Pre-Scaled image width (For more information refer to chapter 15-4)	0x0

15.7.4 SOURCE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
SRCImgSize	0x7700000C	R/W	Source image size	0x0

SRCImgSize	Bit	Description	Initial State
SRCHeight	[23:12]	Source image height (For more information refer to chapter 15-4). When source image height is 4096, SRCHeight should be 0.	0x0
SRCWidth	[11:0]	Source image width (For more information refer to chapter 15-4). When source image width is 4096, SRCWidth should be 0.	0x0

15.7.5 HORIZONTAL MAIN SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
MainScale_H_Ratio	0x77000010	R/W	Main scale ratio for horizontal direction	0x0

SRC_Width	Bit	Description	Initial State
MainScale_H_Ratio	[8:0]	Main scale ratio for horizontal direction (For more information refer to chapter 15-4)	0x0

15.7.6 VERTICAL MAIN SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
MainScale_V_Ratio	0x77000014	R/W	Main scale ratio for vertical direction	0x0

SRC_Width	Bit	Description	Initial State
MainScale_V_Ratio	[8:0]	Main scale ratio for vertical direction (For more information refer to chapter 15-4)	0x0

15.7.7 DESTINATION IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
DSTImgSize	0x77000018	R/W	Destination image size	0x0

SRCImgSize	Bit	Description	Initial State
DSTHeight	[23:12]	Destination image height (For more information refer to chapter 15-4)	0x0
DSTWidth	[11:0]	Destination image width (For more information refer to chapter 15-4)	0x0

15.7.8 PRE-SCALE SHIFT FACTOR REGISTER

Register	Address	R/W	Description	Reset Value
PreScale_SHFactor	0x7700001C	R/W	Pre-scale shift factor	0x0

SRC_Width	Bit	Description	Initial State
PreScale_SHFactor	[3:0]	Pre-scale shift factor (For more information refer to chapter 15-4)	0x0

15.7.9 DMA START ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_Y	0x77000020	R/W	[30:0]	DMA (Buffer 0) Start address for source Y or RGB component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_Cb	0x77000024	R/W	[30:0]	DMA (Buffer 0) Start address for source Cb component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_Cr	0x77000028	R/W	[30:0]	DMA (Buffer 0) Start address for source Cr component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_RGB	0x7700002C	R/W	[30:0]	DMA (Buffer 0) Start address for destination Y or RGB component	0x20000000

15.7.10 DMA END ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_Y	0x77000030	R/W	[30:0]	DMA (Buffer 0) End address for source Y or RGB component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_Cb	0x77000034	R/W	[30:0]	DMA (Buffer 0) End address for source Cb component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_Cr	0x77000038	R/W	[30:0]	DMA (Buffer 0) End address for source Cr component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_RGB	0x7700003C	R/W	[30:0]	DMA (Buffer 0) End address for destination Y or RGB component (For more information refer to chapter 15-5)	0x20006300

15.7.11 CURRENT FRAME(BUFFER0) AND NEXT FRAME(BUFFER1) OFFSET REGISTER

Register	Address	R/W	Bit	Description	Reset Value
Offset_Y	0x77000040	R/W	[23:0]	Offset of Y or RGB component for fetching source image (For more information refer to chapter 15-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_Cb	0x77000044	R/W	[23:0]	Offset of Cb component for fetching source image (For more information refer to chapter 15-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_Cr	0x77000048	R/W	[23:0]	Offset of Cr component for fetching source image (For more information refer to chapter 15-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_RGB	0x7700004C	R/W	[23:0]	Offset of Y or RGB component for restoring destination image (For more information refer to chapter 15-5)	0x0

Caution:: 0X77000050 is reserved.

15.7.12 NEXT FRAME DMA START ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_Y	0x77000054	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for source Y or RGB component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_Cb	0x77000058	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for source Cb component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_Cr	0x7700005C	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for source Cr component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_RGB	0x77000060	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for destination Y or RGB component	0x20000000

15.7.13 NEXT FRAME DMA END ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_Y	0x77000064	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for source Y or RGB component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_Cb	0x77000068	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for source Cb component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_Cr	0x7700006C	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for source Cr component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_RGB	0x77000070	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for destination Y or RGB component (For more information refer to chapter 15-5)	0x20006300

15.7.14 DMA START ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_oCb	0x77000074	R/W	[30:0]	DMA (Buffer 0) Start address for destination Cb component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_oCr	0x77000078	R/W	[30:0]	DMA (Buffer 0) Start address for destination Cr component	0x20000000

15.7.15 DMA END ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_oCb	0x7700007C	R/W	[30:0]	DMA (Buffer 0) End address for destination Cb component (For more information refer to chapter 15-5)	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_oCr	0x77000080	R/W	[30:0]	DMA (Buffer 0) End address for destination Cr component (For more information refer to chapter 15-5)	0x20000000

15.7.16 CURRENT FRAME(BUFFER0) AND NEXT FRAME(BUFFER1) OFFSET REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
Offset_oCb	0x77000084	R/W	[23:0]	Offset of Cb component for fetching destination image (For more information refer to chapter 15-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_oCr	0x77000088	R/W	[23:0]	Offset of Cr component for fetching destination image (For more information refer to chapter 15-5)	0x0

15.7.17 NEXT FRAME DMA START ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_oCb	0x7700008C	R/W	[30:0]	Next Frame DMA (Buffer 1) Start address for destination Cb component	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_oCr	0x77000090	R/W	[30:0]	Next Frame DMA (Buffer 1) Start address for destination Cr component	0x20006300

15.7.18 NEXT FRAME DMA END ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_oCb	0x77000094	R/W	[30:0]	Next Frame DMA (Buffer 1) End address for destination Cb component (For more information refer to chapter 15-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_oCr	0x77000098	R/W	[30:0]	Next Frame DMA (Buffer 1) End address for destination Cr component (For more information refer to chapter 15-5)	0x20006300

15.7.19 POSTENVID REGISTER TO ENABLE VIDEO PROCESSING

Register	Address	R/W	Bit	Description	Reset Value
POSTENVID	0x7700009C	R/W	[31]	Enable Video Processing. It enables the operation of POST Processor. It is de-asserted automatically after operation of the current frame is finished. It must be disabled (POSTENVID=0) during control register configuration state. It can not be de-asserted during operation. But it can be de-asserted in case that POST Processor is only ready for operation.	0x0

15.7.20 MODE CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
MODE_2	0x770000A0	R/W	Mode Register 2	0x0

MODE_2	Bit	Description	Initial State
ADDR_CH_DIS	[4]	Next Address Change Disable in Free Run Mode When the current frame is completely finished and ADDR_CH_DIS is 0, Next frame address set of NxtADDRXXX is copied into the current frame address set of ADDRXXX. But if ADDR_CH_DIS is 1, ADDRXXX is not changed. (For more information refer to chapter 15.6.2) 0 = Address Change Enable 1 = Address Change Disable	0
BC_SEL	[3]	DMA address Change Selection 0 = Address change at EVEN/ODD FIELD end 1 = Address change at FRAME end	0
reserved	[2:0]	Must be 0	0

16 TV SCALER (POST PROCESSOR)

This chapter describes the functions and usage of TV Scaler in S3C6410X.

16.1 OVERVIEW

TV Scaler is similar to Post Processor except FIFO size (targeted SD TV) and Input FIFO Mode (Figure 16-2).

The TV Scaler performs video/graphic scale, video format conversion and color space conversion. It is composed of Data-Path, DMA controller and Register files as shown in the Figure 16-1.

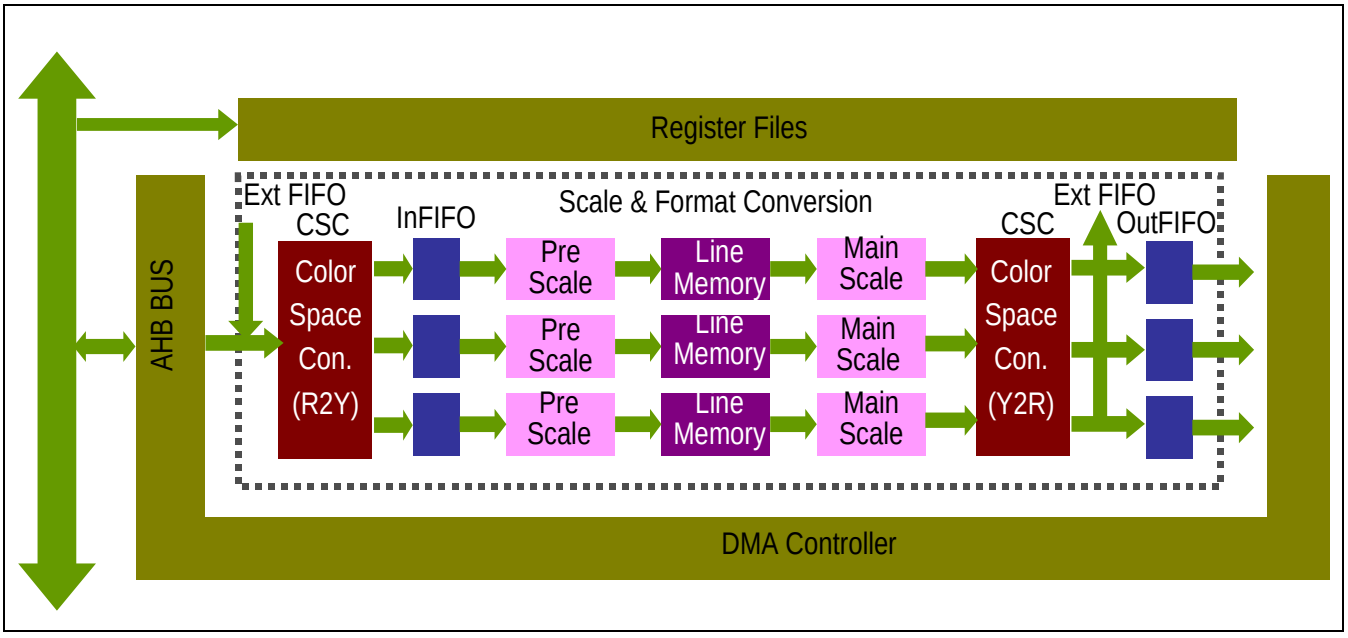


Figure 16-1. Block Diagram of TV Scaler

16.2 FEATURES

- Dedicated DMA with offset address
- 3 Channel scaling pipelines for video/graphic scaling up/down or zooming in/out
- Video input format: 420, 422 format
- Graphic input format: 16-bit (565format) or 24-bit
- Graphics Output format to Memory: 16-bit (565 format) / 24-bit graphic data (progressive only)
- Video Output format to Memory: YCbCr420, YCbCr422
- Input format to external FIFO: YCbCr444 / RGB (24-bit)
- Output format to external FIFO: YCbCr444 / RGB (30-bit) for interlace and progressive
- FreeRun Mode Operation
- Programmable source image size up to 800x2048 resolution
- Programmable destination image size up to 2048x2048 resolution
- Programmable scaling ratio
- Format conversion for video signals
- Color space conversion from YCbCr to RGB
- Color Space conversion from RGB to YCbCr

16.3 A SOURCE AND DESTINATION IMAGE DATA FORMAT

On top of FIMV TV Scaler, there are two output modes such as DMA mode and FIFO mode as shown in the following Figure 16-2.

In FIFO mode (if LCDPathEnable = 1, for more information refer to Chapter 16.7 Register File List), destination image is transferred to the FIFO in display controller (or some other IP with FIFO interface) without additional memory bandwidth such as TV SCALER-to-Memory and Memory-to-Display Controller. The source image format and the destination image format are described in section 16.3.2 FIFO mode.

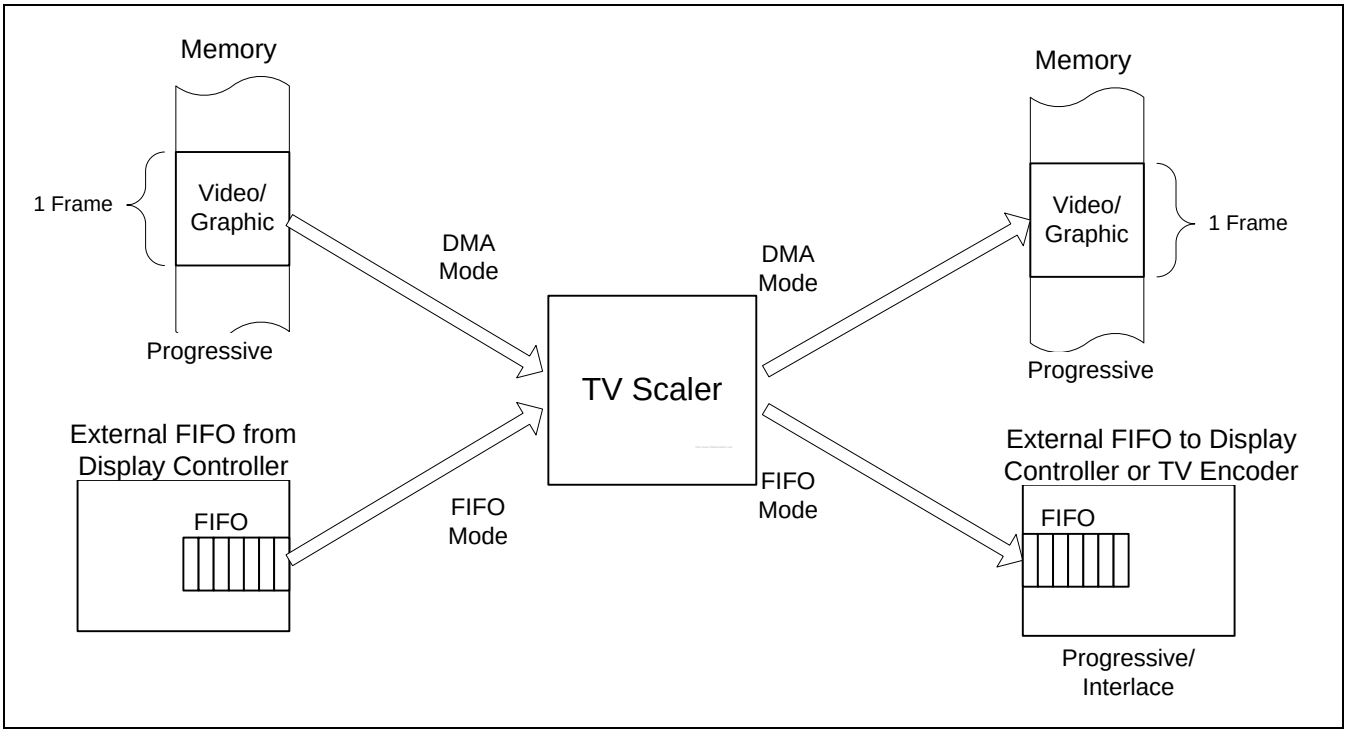


Figure 16-2. Two input and output modes in TV Scaler Block Diagram

16.3.1 DMA MODE OPERATION

Various source and destination image formats can be selected according to the mode configuration as described in Table 16-1 a. Source image format is one of the following YCbCr420, YCbCr422, RGB16-bit (565format) and RGB 24-bit format. Destination image format is identical to the source image format such as YCbCr420, YCbCr422, RGB 16-bit (565format) and RGB 24-bit format. The detail control signal is define in Table 16-1 b.

In the case of YCbCr420 source/destination image format, each component of Y, Cb and Cr is stored in each own separated address space without any interleaving as shown in Case A of Figure 16-3 (a) and Figure 16-4. In the other cases, either byte or half-word interleaving is applied within unified address space as described in Figure 16-3 (b). Byte interleaving order of YCbCr422 source image can be selected either from YCbYCr or CbYCrY as shown in case B and C of Figure 16-3 (b) (b) and Figure 16-4. Byte order of RGB 24-bit and half-word order of RGB 16-bit are described in case D and E of Figure 16-3 (b) and Figure 16-4.

In both cases of YCbCr420 and YCbCr422 source image format, whether MPEG4 format or MPEG2/H.263 format needs to be selected according to the sampling position of the chroma information as shown in Figure 16-4. Destination image have the same sampling position of source image when both source and destination image are YCbCr format. If source image is RGB format and destination image is YCbCr format, the sampling position of chroma component is the same position of Luma.

All source and destination image data must be stored in memory system aligned with word boundary. It means that neither byte nor half-word size DMA operations are supported (*For more information refer to section 16-5 for DMA operation*). Therefore, the width of source and destination image must be selected to satisfy the word boundary condition (*For more information refer to Section 16-4 for image size*).

Table 16-1 a. Mode configuration for video/graphic source format and the corresponding data format

MODE[8] SRC420	MODE[3] InRGB	MODE[2] INTER-LEAVE	MODE[1] InRGB Format	MODE[15][0] InYCbCr Format	Description	
					Video/Graphic Format	Data Format in Figure 16-3 and 4
1	0	0	1	×	420 YCbCr Format	A
0	0	1	1	00/10	422 YCbYCr Format	B/B’
0	0	1	1	01/11	422 CbYCrY Format	C/C’
0	1	1	1	×	RGB 24-bit true color	D
0	1	1	0	×	RGB 16-bit Format	E

Table 16-1 b. Mode configuration for video/graphic destination format and the corresponding data format

MODE[18] OutRGB	MODE[17] DST420	MODE[20][19] OutYCbCr Format	MODE[4] OutRGB Format	Description	
				Video/Graphic Format	Data Format in Figure 16-3and 4
0	1	×	X	420 YCbCr Format	A
0	0	00/10	X	422 YCbYCr Format	B/B’
0	0	01/11	X	422 CbYCrY Format	C/C’
1	X	×	1	RGB 24-bit true color	D
1	X	×	0	RGB 16-bit Format	E

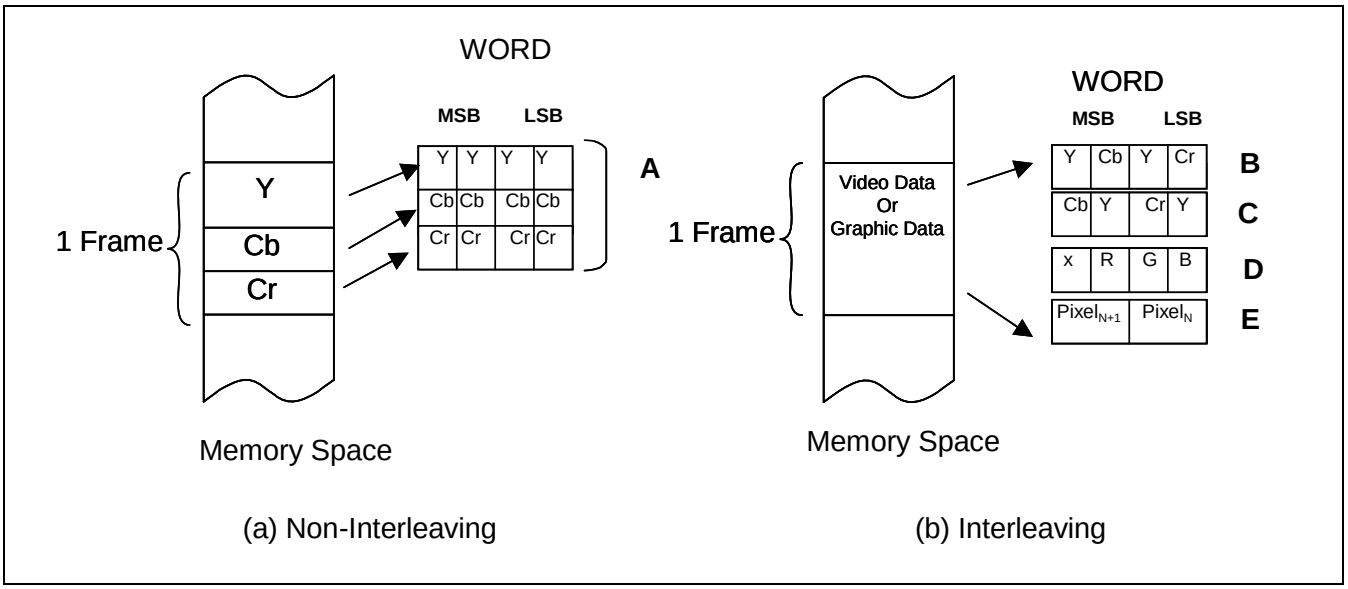


Figure 16-3. Data format stored in external memory

Case A	Y/Cb/Cr $_{N+3}$	Y/Cb/Cr $_{N+2}$	Y/Cb/Cr $_{N+1}$	Y/Cb/Cr $_{N}$
Case B	Y $_{N+1}$	Cb $_{N}$	Y $_{N}$	Cr $_{N}$
Case B'	Y $_{N+1}$	Cr $_{N}$	Y $_{N}$	Cb $_{N}$
Case C	Cb $_{N}$	Y $_{N+1}$	Cr $_{N}$	Y $_{N}$
Case C'	Cr $_{N}$	Y $_{N+1}$	Cb $_{N}$	Y $_{N}$
Case D	Don't Care	R_N	G_N	B_N

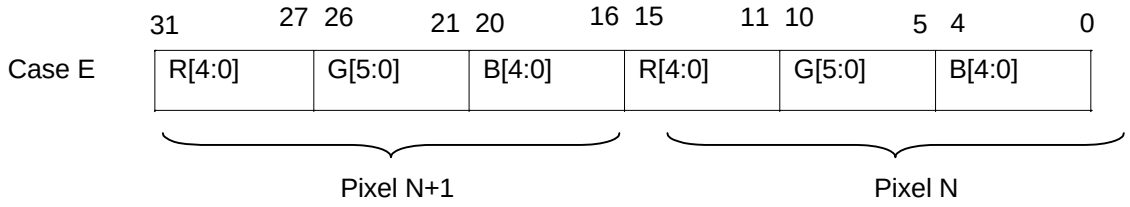


Figure 16-4. Byte and half-word organization Block Diagram

16.3.2 FIFO MODE OPERATION

Output data format is determined by MODE[18] as described in Table 16-1 b. The output data format is fixed to 30-bit data, 10-bit per each component RGB or YCbCr444. The other specific mode configuration signals mentioned in Table 16-1 b are ignored if ExtFIFOIn or LCDPathEnable is set to “1”.

Table 16-2 Input and Output Data Format in FIFO Mode

OutRGB (MODE[18])	Output Data Format (LCDPathEnable = 1) (Progressive / Interlace)
0	YCbCr 444
1	RGB 30-bit

In Output FIFO mode, either progressive or interlace scan mode can be selected according to “interlace” control register as defined in Section 16.7. Register Files Lists. The “interlace” control bit is enabled only if LCDPathEnable = 1, otherwise its value is unaffected to DMA mode operation which support only progressive.

Even if an interlaced scan mode is enabled (LCDPathEnable = 1 and Interlace = 1), per frame management, which consists of even field and odd filed, operates automatically. 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.

16.4 IMAGE SIZE AND SCALE RATIO

The RGB graphic source image size is determined by number of pixels along to horizontal and vertical directions. YCbCr420 and YCbCr422 source image size is determined only by numbers of Y samples along to horizontal and vertical directions. Destination image size is determined by final dimension of RGB graphic image. The final dimension of RGB graphic image is determined after color space conversion if source image is YCbCr image.

As explained in the previous section, SRC_Width and DST_Width satisfies the word boundary constraints such that the number of horizontal pixel can be represented by kn where n = 1,2,3, … and k = 1 / 2 / 8 for 24bppRGB / 16bppRGB / YCbCr420 image, respectively. Also SRC_Width must be 4’s multiple of PreScale_H_Ratio and SRC_Height must be 2’s multiple of PreScale_V_Ratio.

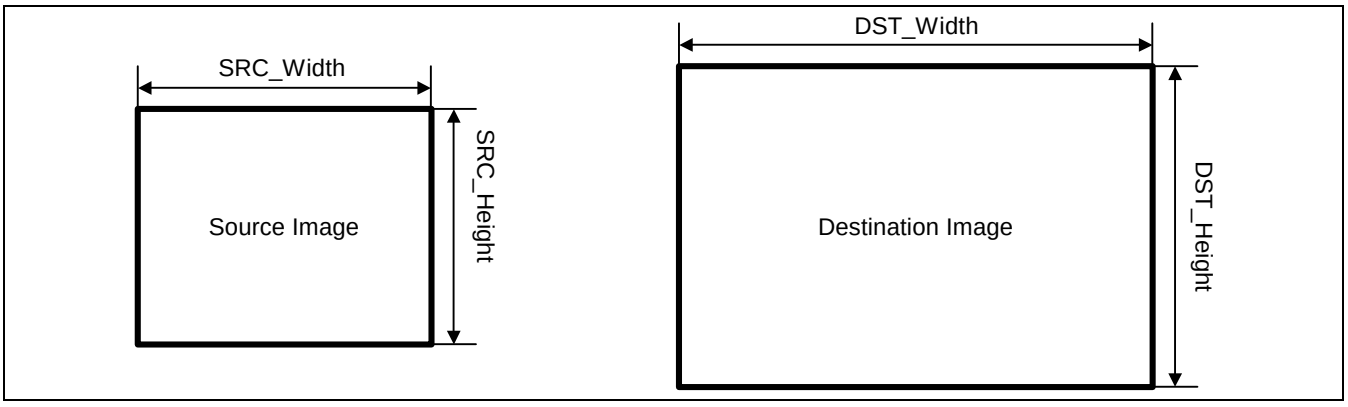


Figure 16-5. Source destination image size Block Diagram

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) { PreScale_H_Ratio = 32; H_Shift = 5; }
else if (SRC_Width >= 16 × DST_Width) { PreScale_H_Ratio = 16; H_Shift = 4; }
else if (SRC_Width >= 8 × DST_Width) { PreScale_H_Ratio = 8; H_Shift = 3; }
else if (SRC_Width >= 4 × DST_Width) { PreScale_H_Ratio = 4; H_Shift = 2; }
else if (SRC_Width >= 2 × DST_Width) { PreScale_H_Ratio = 2; H_Shift = 1; }
else { PreScale_H_Ratio = 1; H_Shift = 0; }

PreScale_DSTWidth = SRC_Width / PreScale_H_Ratio;
dx = (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) { PreScale_V_Ratio = 32; V_Shift = 5; }
else if (SRC_Height >= 16 × DST_Height) { PreScale_V_Ratio = 16; V_Shift = 4; }
else if (SRC_Height >= 8 × DST_Height) { PreScale_V_Ratio = 8; V_Shift = 3; }
else if (SRC_Height >= 4 × DST_Height) { PreScale_V_Ratio = 4; V_Shift = 2; }
else if (SRC_Height >= 2 × DST_Height) { PreScale_V_Ratio = 2; V_Shift = 1; }
else { PreScale_V_Ratio = 1; V_Shift = 0; }

PreScale_DSTHeight = SRC_Height / PreScale_V_Ratio;
dy = (SRC_Height << 8) / (DST_Height << V_Shift);

PreScale_SHFactor = 10 – (H_Shift + V_Shift);

16.5 DMA OPERATION OF SOURCE AND DESTINATION IMAGE

There are three address categories such as start address, end address and offset address for DMA operation. Each address category consists of three sources address components of Y/Cb/Cr and three destinations address component of RGB/oCb/oCr. If a source image is stored in the non-interleaved format such as YCbCr420, all source address components are valid as shown in Figure 16-6 (a). If a source image is stored by the interleaved format such as a RGB graphic format or an YCbCr422 format, only Y component of three source components is valid and two chroma address components are invalid as shown in Figure 16-6 (b). If a destination image is stored in the non-interleaved format such as YCbCr420, all source address components RGB/oCb/oCr are valid as shown in Figure 16-6 (a). If a source image is stored by the interleaved format such as a RGB graphic format or an YCbCr422 format, only RGB component of three source components is valid and two chroma address components of oCb/oCr are invalid as shown in Figure 16-6 (b).

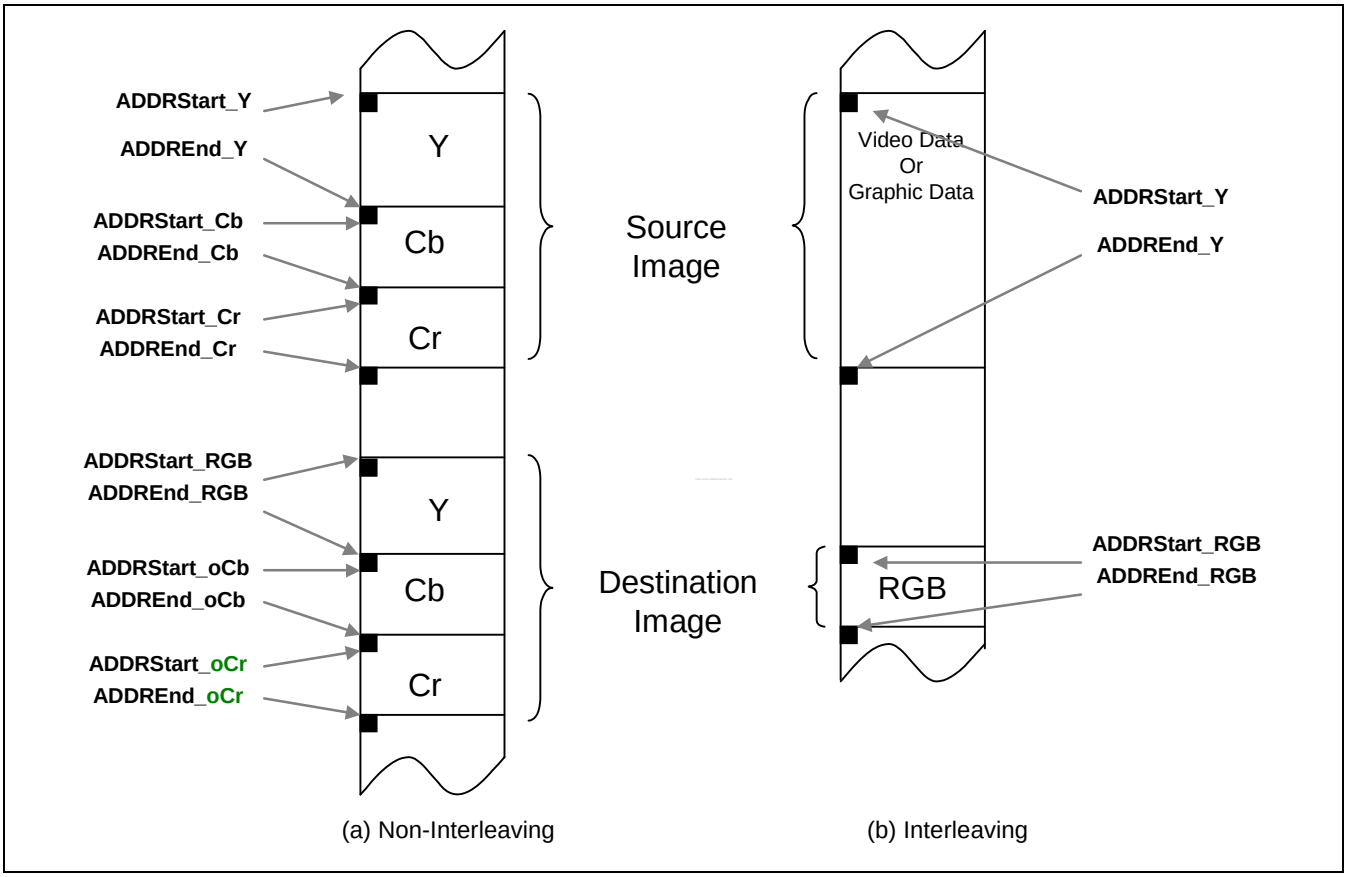


Figure 16-6. Start and end address set according to memory allocation type

16.5.1 START ADDRESS

Start address of ADDRStart_Y/Cb/Cr/RGB/oCb/oCr points the first word address where the corresponding component of Y/Cb/Cr/RGB/oCb/oCr is read or written. Each one must be aligned with word boundary (i.e. ADDRStart_X[1:0] = 00). ADDRStart_Cb and ADDRStart_Cr are valid only for the YCbCr420 source image format. ADDRStart_oCb and ADDRStart_oCr are valid only for the YCbCr420 destination image format.

16.5.2 END ADDRESS

< Source End Address >

ADDREnd_Y

= ADDRStart_Y + Memory size for the component of Y/RGB

= ADDRStart_Y + (SRC_Width × SRC_Height) × ByteSize_Per_Pixel + Offset_Y × (SRC_Height-1)

ADDREnd_Cb (Valid for YCbCr420 source format)

= ADDRStart_Cb + Memory size for the component of Cb

= ADDRStart_Cb + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cb × (SRC_Height/2-1)

ADDREnd_ Cr (Valid for YCbCr420 source format)

= ADDRStart_ Cr + Memory size for the component of Cr

= ADDRStart_Cr + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cr × (SRC_Height/2-1)

< Destination End Address >

ADDREnd_RGB

= ADDRStart_RGB + Memory size for the component of RGB data (or component of Y)

= ADDRStart_RGB + (DST_Width × DST_Height) × ByteSize_Per_Pixel + Offset_RGB × (DST_Height-1)

ADDREnd_oCb (Valid for YCbCr420 destination format)

= ADDRStart_oCb + Memory size for the component of Cb

= ADDRStart_oCb + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cb × (SRC_Height/2-1)

ADDREnd_ oCr (Valid for YCbCr420 destination format)

= ADDRStart_ oCr + Memory size for the component of Cr

= ADDRStart_oCr + (SRC_Width/2 × SRC_Height/2) × ByteSize_Per_Pixel + Offset_Cr × (SRC_Height/2-1)

Where,

Offset_Y/Cb/Cr/RGB

= Memory size for offset per a horizontal line

= Number of pixel (or sample) in horizontal offset × ByteSize_Per_Pixel (or Sample)

$$\text{ByteSize\_Per\_Pixel} = \begin{cases} 1 \text{ for YCbCr420} \\ 2 \text{ for 16-bit RGB and YcbCr422} \\ 4 \text{ for 24-bit RGB} \end{cases}$$

Offset address is used for the following two situations. One is to fetch some parts of source image in order to zoom in/out as shown in Figure 16-7 (a). The other is to restore destination image for PIP (picture-in-picture) applications as shown in Figure 16-7 (b). The word boundary constraints must be satisfied in both cases.

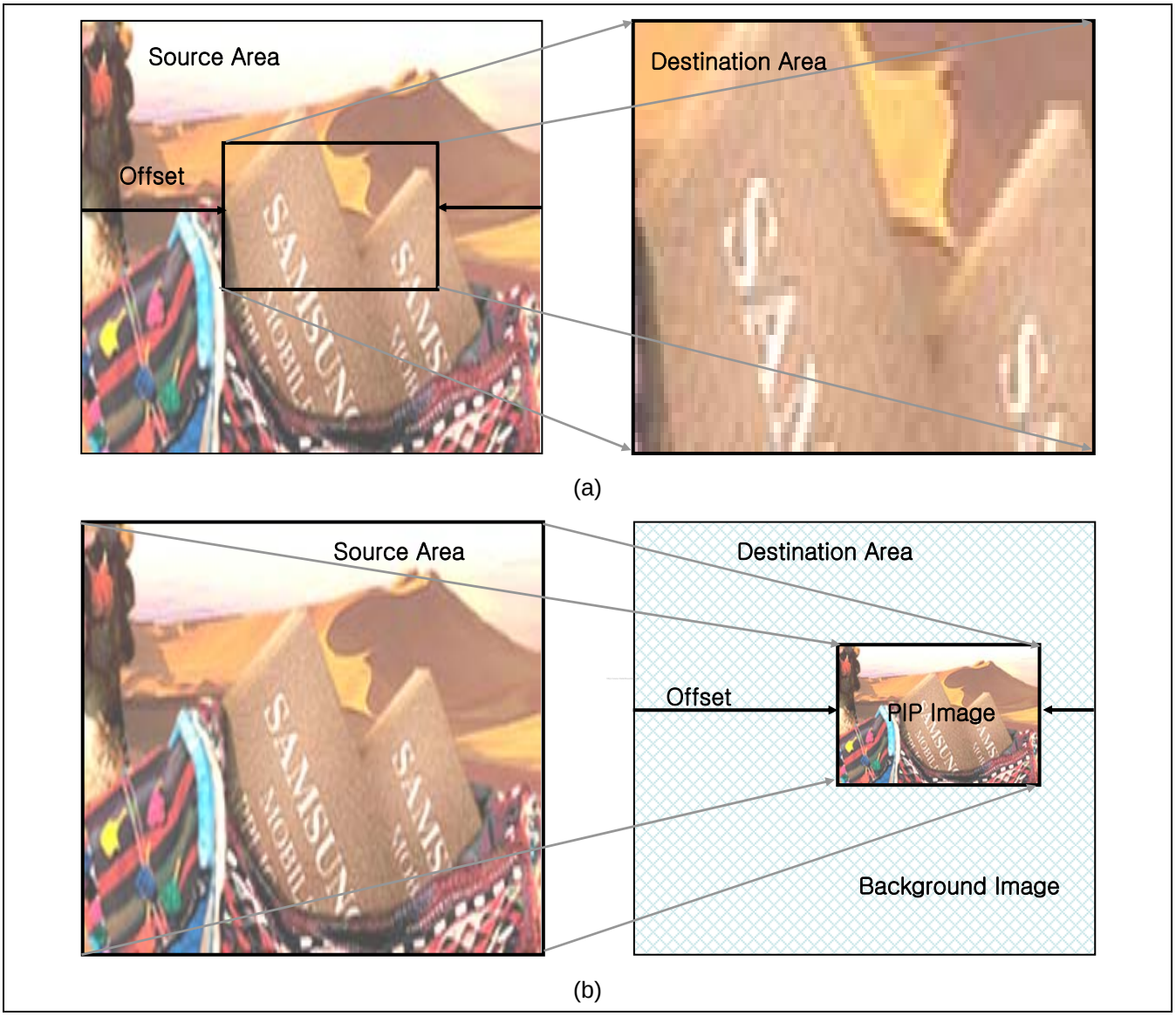


Figure 16-7. Offset address for (a) source image for zoom in/out operation and (b) Destination image for PIP applications

16.6 FRAME MANAGEMENT OF TV SCALER

16.6.1 PER FRAME MANAGEMENT MODE

Per frame management of TV Scaler are controlled by two controls register such as POSTENVID and POSTINT as shown in Figure 16-8. “POSTENVID” triggers the operation of TV SCALER. It is automatically de-asserted when all operations of the given frame are completed. Before asserting “POSTENVID”, all control registers must be set to the proper value as explained in the previous chapters. When all operations are completed, interrupt pending register is asserted (POSTINT=1), if the interrupt enable signal is asserted (INTEN=1). The POSTINT signal, directing to the interrupt controller, must be cleared by the interrupt service routine. The polling POSTENVID is also used to detect the end of the operation.

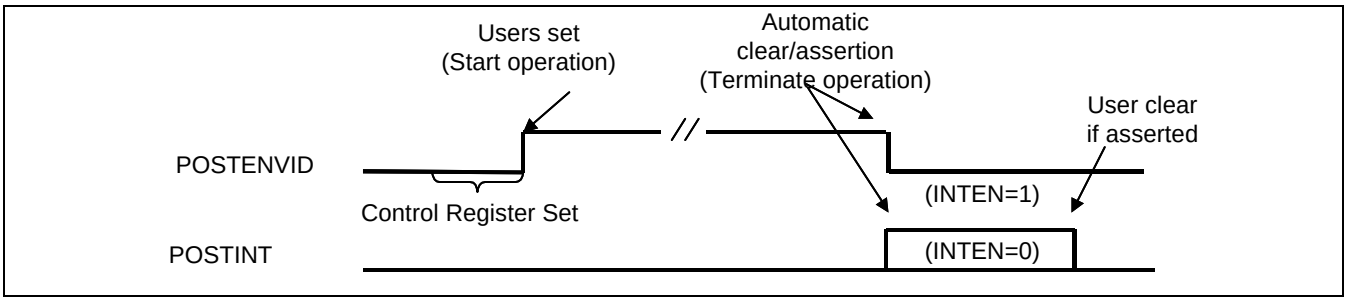


Figure 16-8. Start and termination of TV SCALER operation (AutoLoadEnable = 0)

16.6.2 FREE RUN MODE

To activate the new frame management scheme of free run operation, you must set “AutoLoadEnable” bit to 1. In this mode, user can pre-define the next frame-related address set of NxtADDRXXX (defined in chapter 16.7 Register files) even when the current frame is under operation. When the current frame is completely finished, the following operations are executed step-by-step.

1. According to INTEN, interrupt signal is asserted or not.
2. Next frame address set of NxtADDRXXX is copied into the current frame address set of ADDRXXX.
3. ENVID is automatically asserted and the next frame operation starts up.

** In Free Run Mode, the other register value except the address stuff must remain same value between the current frame and the next frame.

16.7 REGISTER FILE LISTS

Register	Address	R/W	Description	Reset Value
MODE	0x76300000	R/W	Mode Register	0x00070B12
PreScale_Ratio	0x76300004	R/W	Pre-Scale ratio for vertical and horizontal	0x0
PreScaleImgSize	0x76300008	R/W	Pre-Scaled image size	0x0
SRCImgSize	0x7630000C	R/W	Source image size	0x0
MainScale_H_Ratio	0x76300010	R/W	Main scale ratio along to horizontal direction	0x0
MainScale_V_Ratio	0x76300014	R/W	Main scale ratio along to vertical direction	0x0
DSTImgSize	0x76300018	R/W	Destination image size	0x0
PreScale_SHFactor	0x7630001C	R/W	Pre-scale shift factor	0x0
ADDRStart_Y	0x76300020	R/W	DMA (Buffer 0) Start address for source Y or RGB component	0x20000000
ADDRStart_Cb	0x76300024	R/W	DMA (Buffer 0) Start address for source Cb component	0x20000000
ADDRStart_Cr	0x76300028	R/W	DMA (Buffer 0) Start address for source Cr component	0x20000000
ADDRStart_RGB	0x7630002C	R/W	DMA (Buffer 0) Start address for destination Y or RGB component	0x20000000
ADDREnd_Y	0x76300030	R/W	DMA (Buffer 0) End address for source Y or RGB component	0x20006300
ADDREnd_Cb	0x76300034	R/W	DMA (Buffer 0) End address for source Cb component	0x20006300
ADDREnd_Cr	0x76300038	R/W	DMA (Buffer 0) End address for source Cr component	0x20006300
ADDREnd_RGB	0x7630003C	R/W	DMA (Buffer 0) End address for destination Y or RGB component	0x20006300
Offset_Y	0x76300040	R/W	Offset of Y or RGB component for fetching source image	0x0
Offset_Cb	0x76300044	R/W	Offset of Cb component for fetching source image	0x0
Offset_Cr	0x76300048	R/W	Offset of Cr component for fetching source image	0x0
Offset_RGB	0x7630004C	R/W	Offset of Y or RGB component for restoring destination image	0x0
RESERVED	0x76300050	-	-	-
NxtADDRStart_Y	0x76300054	R/W	Next Frame (Buffer 1) DMA Start address for source Y or RGB component	0x20000000
NxtADDRStart_Cb	0x76300058	R/W	Next Frame (Buffer 1) DMA Start address for source Cb component	0x20000000
NxtADDRStart_Cr	0x7630005C	R/W	Next Frame (Buffer 1) DMA Start address for source Cr component	0x20000000
NxtADDRStart_RGB	0x76300060	R/W	Next Frame (Buffer 1) DMA Start address for destination Y or RGB component	0x20000000

Register	Address	R/W	Description	Reset Value
NxtADDREnd_Y	0x76300064	R/W	Next Frame (Buffer 1) DMA End address for source Y or RGB component	0x20006300
NxtADDREnd_Cb	0x76300068	R/W	Next Frame (Buffer 1) DMA End address for source Cb component	0x20006300
NxtADDREnd_Cr	0x7630006C	R/W	Next Frame (Buffer 1) DMA End address for source Cr component	0x20006300
NxtADDREnd_RGB	0x76300070	R/W	Next Frame (Buffer 1) DMA End address for destination Y or RGB component	0x20006300
ADDRStart_oCb	0x76300074	R/W	DMA (Buffer 0) Start address for destination Cb component	0x20000000
ADDRStart_oCr	0x76300078	R/W	DMA (Buffer 0) Start address for destination Cr component	0x20000000
ADDREnd_oCb	0x7630007C	R/W	DMA (Buffer 0) End address for destination Cb component	0x20000000
ADDREnd_oCr	0x76300080	R/W	DMA (Buffer 0) End address for destination Cr component	0x20000000
Offset_oCb	0x76300084	R/W	Offset of Cb component for fetching destination image	0x0
Offset_oCr	0x76300088	R/W	Offset of Cr component for fetching destination image	0x0
NxtADDRStart_oCb	0x7630008C	R/W	Next Frame DMA (Buffer 1) Start address for destination Cb component	0x20006300
NxtADDRStart_oCr	0x76300090	R/W	Next Frame DMA (Buffer 1) Start address for destination Cr component	0x20006300
NxtADDREnd_oCb	0x76300094	R/W	Next Frame DMA (Buffer 1) End address for destination Cb component	0x20006300
NxtADDREnd_oCr	0x76300098	R/W	Next Frame DMA (Buffer 1) End address for destination Cr component	0x20006300
POSTENVID	0x7630009C	R/W	Enable Video Processing.	0x0
MODE_2	0x763000A0	R/W	Mode Register 2	0x0

16.7.1 MODE CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MODE	0x76300000	R/W	Mode Register [31:0]	0x00070B12

MODE	Bit	Description	Initial State
ExtFIFOIn	[31]	In FIFO Mode Enable. 1 for FIFO mode and 0 for DMA mode	0
CLKVALUP	[30]	Select CLKVAL_F update timing control 0 = always 1 = start of a frame (only once per frame)	0
CLKVAL_F	[29:24]	Determine the rates of TSCLK and CLKVAL[5:0] TSCLK = Clock source / (CLKVAL+1), where CLKVAL >= 1 Note. The maximum frequency of VCLK is 66MHz.	0
CLKDIR	[23]	Select the clock source as direct or divide using CLKVAL_F register 0 = Direct clock (frequency of TSCLK = frequency of Clock source) 1 = Divided by CLKVAL_F	0
CLKSEL_F	[22:21]	Select the Video Clock source 00 = HCLK 01 = Clock selected by CLK_SRC[27:26] (refer to Chapter3. System Controller) 10 = reserved 11 = 27MHz Ext Clock input	0
OutYCbCrFormat	[20:19]	It determines the byte organization of word data when the destination image is interleaved YCbCr format. For more information refer to Table 16-1 b.	0
OutRGB	[18]	It indicates the output color space of destination image. 0 for YCbCr or 1 for RGB.	1
DST420	[17]	0 for YCbCr422 and 1 for YCbCr420 destination format. It is valid only for YCbCr destination image (i.e. OutRGB = 0)	1
R2YSel	[16]	Select color space conversion equation from RGB to YCbCr. 1 for YCbCr Wide range and 0 for YCbCr Narrow range.	1
InYCbCrFormat_MSB	[15]	It determines the MSB of byte organization of word data when the source image is interleaved YCbCr format. For more information refer to Table 16-1 a.	0
AutoLoadEnable	[14]	AutoLoadEnalbe. 0 for Per Frame mode and 1 for Free Run mode	0
LCDPathEnable	[13]	Out FIFO Mode Enable. 1 for FIFO mode and 0 for DMA mode	0
Interlace	[12]	Output scan method selection register only when FIFO mode (LCDPathEnable =1). 1 for Interlace scan and 0 for progressive scan. In DMA mode (LCDPathEnable = 0), progressive scan is executed irrespective of the value given.	0
Wide/Narrow	[11:10]	Select color space conversion equation from YCbCr to RGB according to input value range. 2'10 for YCbCr Wide range and 2'01 for YCbCr Narrow range	2'b10

MODE	Bit	Description	Initial State
SRC420	[8]	0 for YCbCr422 and 1 for YCbCr420 source format. It is valid only for YCbCr source image (i.e. InRGB = 0)	1
INTEN	[7]	Interrupt Enable. It determines whether the POSTINT signal is asserted or not, when the processing of the current frame is finished. 0: disable, 1: enable.	0
POSTINT	[6]	Interrupt Pending Bit. If INTEN is enabled, it is automatically asserted right after finishing operation of the current frame. It must be cleared by interrupt service routine. 0: disable, 1: enable.	0
IRQ_LEVEL	[5]	It determines the interrupt generation scheme. 1 for level interrupt. (must be 1)	0
OutRGBFormat	[4]	It determines the output format of destination image. 0 for 16-bit (565 format) RGB and 1 for 24-bit RGB.	1
InRGB	[3]	It indicates the input color space of source image. 0 for YCbCr or 1 for RGB.	0
INTERLEAVE	[2]	It indicates the data format of YCbCr. 0 for Non-Interleaved format (Each component of Y, Cb and Cr is access by the word). 1 for Interleaved format (All components of Y, Cb and Cr are mixed inside single word). It should be 1 when source image is RGB data (or InRGB =1).	0
InRGBFormat	[1]	If the source image is in RGB color space (or InRGB=1), it indicates the data format of graphic image. 0 for 16-bit (565 format) and 1 for 24-bit. Otherwise (or InRGB=0), it should be remains to 1.	1
InYCbCrFormat_LSB	[0]	It determines the LSB of byte organization of word data when the source image is interleaved YCbCr format. For more information refer to Table 16-1 b.	0

Caution:: [9] bit is reserved.

16.7.2 Pre-Scale Ratio Register

Register	Address	R/W	Description	Reset Value
PreScale_Ratio	0x76300004	R/W	Pre-Scale ratio for vertical and horizontal.	0x0

PreScale_Ratio	Bit	Description	Initial State
PreScale_V_Ratio	[13:7]	Pre-scale ratio for vertical direction (For more information refer to chapter 16-4)	0x0
PreScale_H_Ratio	[6:0]	Pre-scale ratio for horizontal direction (For more information refer to chapter 16-4)	0x0

16.7.3 PRE-SCALE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
PreScaleImgSize	0x76300008	R/W	Pre-Scaled image size	0x0

PreScaleImgSize	Bit	Description	Initial State
PreScale_DSTHeight	[23:12]	Pre-Scaled image height (For more information refer to chapter 16-4)	0x0
PreScale_DSTWidth	[11:0]	Pre-Scaled image width (For more information refer to chapter 16-4)	0x0

16.7.4 SOURCE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
SRCImgSize	0x7630000C	R/W	Source image size	0x0

SRCImgSize	Bit	Description	Initial State
SRCHeight	[23:12]	Source image height (For more information refer to chapter 16-4)	0x0
SRCWidth	[11:0]	Source image width (For more information refer to chapter 16-4)	0x0

16.7.5 HORIZONTAL MAIN SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
MainScale_H_Ratio	0x76300010	R/W	Main scale ratio for horizontal direction	0x0

SRC_Width	Bit	Description	Initial State
MainScale_H_Ratio	[8:0]	Main scale ratio for horizontal direction (For more information refer to chapter 16-4)	0x0

16.7.6 VERTICAL MAIN SCALE RATIO REGISTER

Register	Address	R/W	Description	Reset Value
MainScale_V_Ratio	0x76300014	R/W	Main scale ratio for vertical direction	0x0

SRC_Width	Bit	Description	Initial State
MainScale_V_Ratio	[8:0]	Main scale ratio for vertical direction (For more information refer to chapter 16-4)	0x0

16.7.7 DESTINATION IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
DSTImgSize	0x76300018	R/W	Destination image size	0x0

SRCImgSize	Bit	Description	Initial State
DSTHeight	[23:12]	Destination image height (For more information refer to chapter 16-4)	0x0
DSTWidth	[11:0]	Destination image width (For more information refer to chapter 16-4)	0x0

16.7.8 PRE-SCALE SHIFT FACTOR REGISTER

Register	Address	R/W	Description	Reset Value
PreScale_SHFactor	0x7630001C	R/W	Pre-scale shift factor	0x0

SRC_Width	Bit	Description	Initial State
PreScale_SHFactor	[3:0]	Pre-scale shift factor (For more information refer to chapter 16-4)	0x0

16.7.9 DMA START ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_Y	0x76300020	R/W	[30:0]	DMA (Buffer 0) Start address for source Y or RGB component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_Cb	0x76300024	R/W	[30:0]	DMA (Buffer 0) Start address for source Cb component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_Cr	0x76300028	R/W	[30:0]	DMA (Buffer 0) Start address for source Cr component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_RGB	0x7630002C	R/W	[30:0]	DMA (Buffer 0) Start address for destination Y or RGB component	0x20000000

16.7.10 DMA END ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_Y	0x76300030	R/W	[30:0]	DMA (Buffer 0) End address for source Y or RGB component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_Cb	0x76300034	R/W	[30:0]	DMA (Buffer 0) End address for source Cb component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_Cr	0x76300038	R/W	[30:0]	DMA (Buffer 0) End address for source Cr component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_RGB	0x7630003C	R/W	[30:0]	DMA (Buffer 0) End address for destination Y or RGB component (For more information refer to chapter 16-5)	0x20006300

16.7.11 CURRENT FRAME (BUFFER0) AND NEXT FRAME (BUFFER1) OFFSET REGISTER

Register	Address	R/W	Bit	Description	Reset Value
Offset_Y	0x76300040	R/W	[23:0]	Offset of Y or RGB component for fetching source image (For more information refer to chapter 16-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_Cb	0x76300044	R/W	[23:0]	Offset of Cb component for fetching source image (For more information refer to chapter 16-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_Cr	0x76300048	R/W	[23:0]	Offset of Cr component for fetching source image (For more information refer to chapter 16-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_RGB	0x7630004C	R/W	[23:0]	Offset of Y or RGB component for restoring destination image (For more information refer to chapter 16-5)	0x0

Caution:: 0X76300050 is reserved.

16.7.12 NEXT FRAME DMA START ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_Y	0x76300054	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for source Y or RGB component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_Cb	0x76300058	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for source Cb component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_Cr	0x7630005C	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for source Cr component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_RGB	0x76300060	R/W	[30:0]	Next Frame (Buffer 1) DMA Start address for destination Y or RGB component	0x20000000

16.7.13 NEXT FRAME DMA END ADDRESS REGISTER

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_Y	0x76300064	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for source Y or RGB component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_Cb	0x76300068	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for source Cb component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_Cr	0x7630006C	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for source Cr component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_RGB	0x76300070	R/W	[30:0]	Next Frame (Buffer 1) DMA End address for destination Y or RGB component (For more information refer to chapter 16-5)	0x20006300

16.7.14 DMA START ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_oCb	0x76300074	R/W	[30:0]	DMA (Buffer 0) Start address for destination Cb component	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDRStart_oCr	0x76300078	R/W	[30:0]	DMA (Buffer 0) Start address for destination Cr component	0x20000000

16.7.15 DMA END ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_oCb	0x7630007C	R/W	[30:0]	DMA (Buffer 0) End address for destination Cb component (For more information refer to chapter 16-5)	0x20000000

Register	Address	R/W	Bit	Description	Reset Value
ADDREnd_oCr	0x76300080	R/W	[30:0]	DMA (Buffer 0) End address for destination Cr component (For more information refer to chapter 16-5)	0x20000000

16.7.16 CURRENT FRAME (BUFFER0) AND NEXT FRAME (BUFFER1) OFFSET REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
Offset_oCb	0x76300084	R/W	[23:0]	Offset of Cb component for fetching destination image (For more information refer to chapter 16-5)	0x0

Register	Address	R/W	Bit	Description	Reset Value
Offset_oCr	0x76300088	R/W	[23:0]	Offset of Cr component for fetching destination image (For more information refer to chapter 16-5)	0x0

16.7.17 NEXT FRAME DMA START ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_oCb	0x7630008C	R/W	[30:0]	Next Frame DMA (Buffer 1) Start address for destination Cb component	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDRStart_oCr	0x76300090	R/W	[30:0]	Next Frame DMA (Buffer 1) Start address for destination Cr component	0x20006300

16.7.18 NEXT FRAME DMA END ADDRESS REGISTER FOR OUTPUT CB AND CR

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_oCb	0x76300094	R/W	[30:0]	Next Frame DMA (Buffer 1) End address for destination Cb component (For more information refer to chapter 16-5)	0x20006300

Register	Address	R/W	Bit	Description	Reset Value
NxtADDREnd_oCr	0x76300098	R/W	[30:0]	Next Frame DMA (Buffer 1) End address for destination Cr component (For more information refer to chapter 16-5)	0x20006300

16.7.19 POSTENVID REGISTER FOR ENABLE VIDEO PROCESSING

Register	Address	R/W	Bit	Description	Reset Value
POSTENVID	0x7630009C	R/W	[31]	Enable Video Processing. It turns on the operation of TV Scaler. It is de-asserted automatically after operation of the current frame is finished. It should be disabled (POSTENVID=0) during control register configuration state. It can not be de-asserted during operation. But it can be de-asserted in case that TV Scaler is only ready for operation.	0x0

16.7.20 MODE CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
MODE_2	0x763000A0	R/W	Mode Register 2	0x0

MODE_2	Bit	Description	Initial State
FIFO_OUT_PATH	[6:5]	**FIFO output path selection** 00 = TV Encoder output 01 = FIMD WIN1 1x = FIMD WIN2	00
ADDR_CH_DIS	[4]	Next Address Change Disable in Free Run Mode When the current frame is completely finished and ADDR_CH_DIS is 0, Next frame address set of NxtADDRXXX is copied into the current frame address set of ADDRXXX. But if ADDR_CH_DIS is 1, ADDRXXX is not changed. (For more information refer to chapter 16.6.2) 0 = Address Change Enable 1 = Address Change Disable	0
BC_SEL	[3]	DMA address Change Selection 0 = Address change at EVEN/ODD FIELD end 1 = Address change at FRAME end	0
Reserved	[2:0]	Must be 0.	0

17 TV ENCODER

17.1 OVERVIEW

This is TV Encoder which coverts digital video data to analog composite. TV Encoder in S3C6410X has a few special features. First of all, it has the image enhancing engine. The image is enhanced by special effects. Second, it supports the image display of various sizes. There are full, wide and original modes.

Additionally, TV Encoder in S3C6410X supports analog composite-out and S-video out.

17.2 FEATURE

- Built in the MIE (Mobile Image Enhancer) engine
  - Black & White Stretch
  - Blue Stretch & Flesh-Tone Correction
  - Dynamic Horizontal Peaking & LTI
  - Black and White Noise reduction
  - Contrast, Sharpness, Gamma and Brightness Control
- It is possible to display stream, which is different with LCD.
- Original, full and wide size video-out
- Supports following formats :

  - NTSC-M, NTSC-J
  - PAL-B/D/G/H/I , PAL-M , PAL-Nc

17.3 BLOCK DIAGRAM

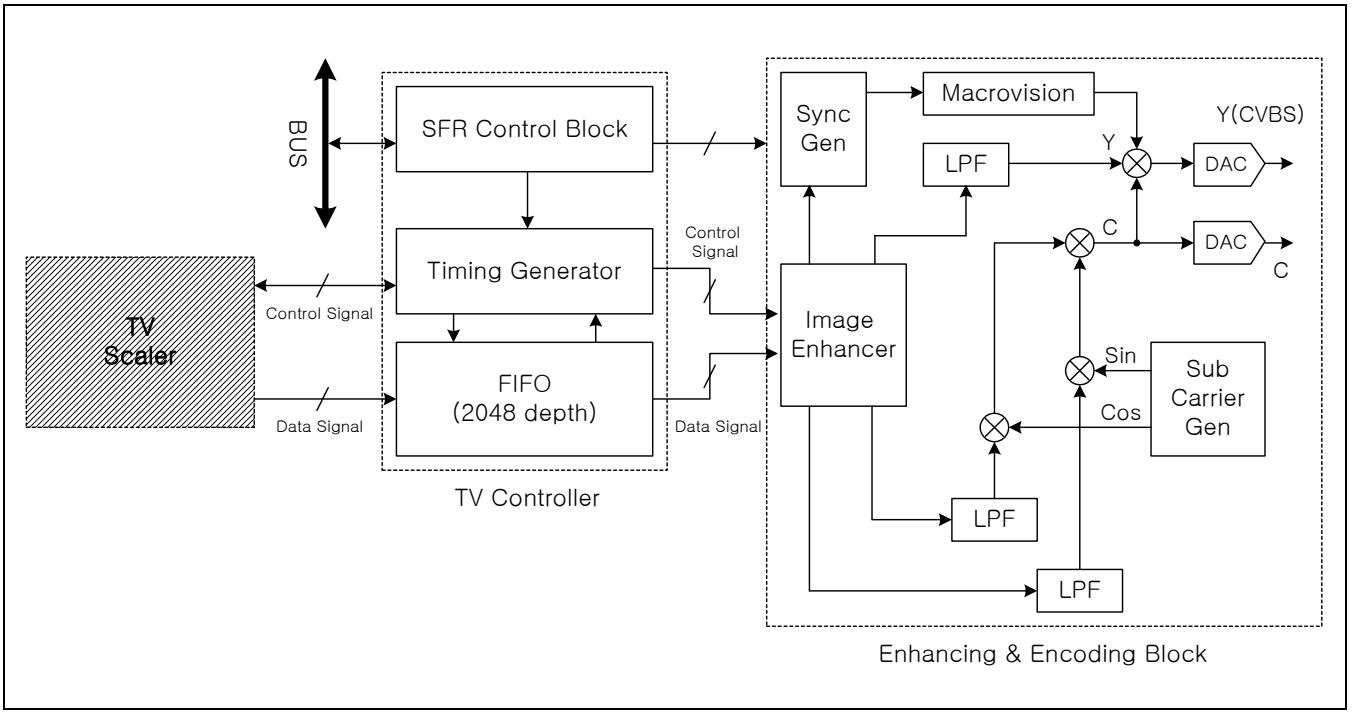


Figure 17-1. TV Encoder Block Diagram

17.4 FUNCTIONAL DESCRIPTIONS

TV Encoder encodes digital pixel data to ITU-R BT.656 format. TV Encoder largely consists of 4 parts.

The first part is the Enhancer & Encoder block. The Enhancer & Encoder block contains the encoder and the image enhancer. The image enhancer takes charge of an image enhancing by controlling Black & White Stretch, Gamma, Bright, and Contrast etc. We can get the image, which is reinforced with some effect. And then the encoder generated TV signal which is ITU-R BT.656 format.

DATA PATHS

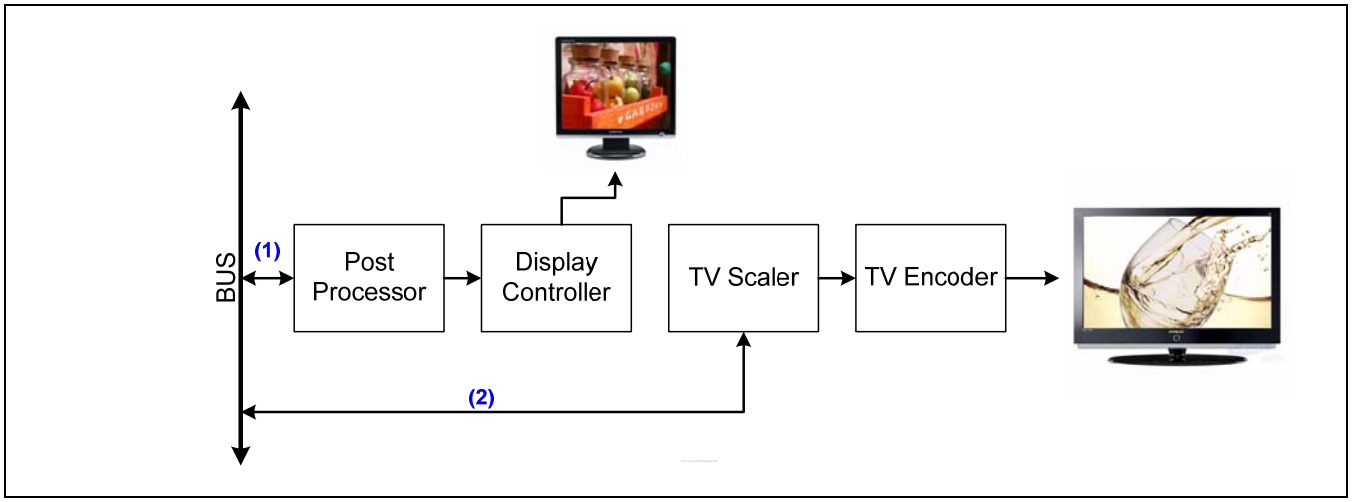


Figure 17-2. TV Encoder Data Path Concept

To display the different image at LCD and TV, there are two paths.

(Post Processor ->) Display controller -> LCD panel, path (1) (in figure).

Data are loaded directly from memory by either Post-Processor or Display Controller.

TV scaler -> TV Encoder, path (2) (in figure).

Data are loaded by TV scaler. It treats data that covert properly size and color-space. Finally, it sends them to TV encoder.

This path is able to display different images between LCD and TV encoder.

17.4.1 COMPOSITION OF ANALOG COMPOSITE SIGNAL

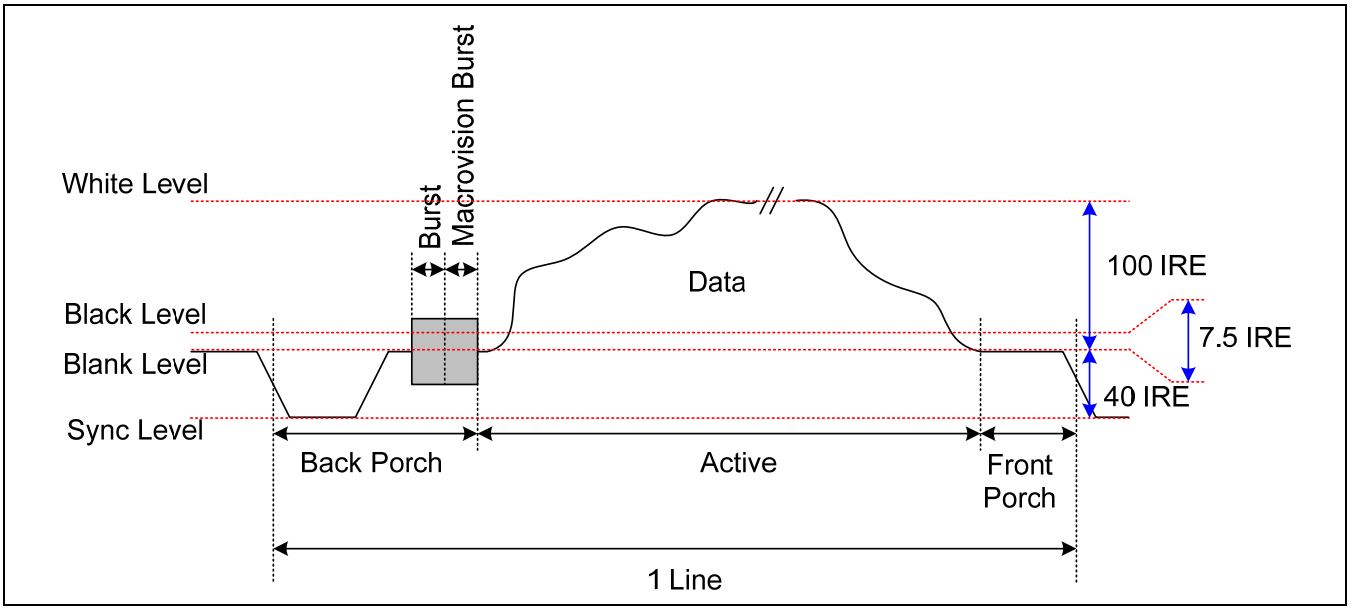


Figure 17-3. Composition of Analog Composite Signal

Figure 17-3 shows horizontal timing. In X axis for TV composite out, it divide 3 timing parts. It is Back-Porch, Active and Front-Porch. Back-Porch and Front-Porch are to synchronize signal. And Active-Region contains valid data.

In Y axis for TV composite out, it contains luminance and chrominance component. DC-level in Figure 17-3 present luminance component. Also, Chrominance component is sampled by Burst frequency. Chrominance component is shown by sine-wave form in Figure 17-3.

According to TV system (NTSC or PAL), it distinguishes Black-level from Blank-level.

17.4.2 COMMON NTSC SYSTEM

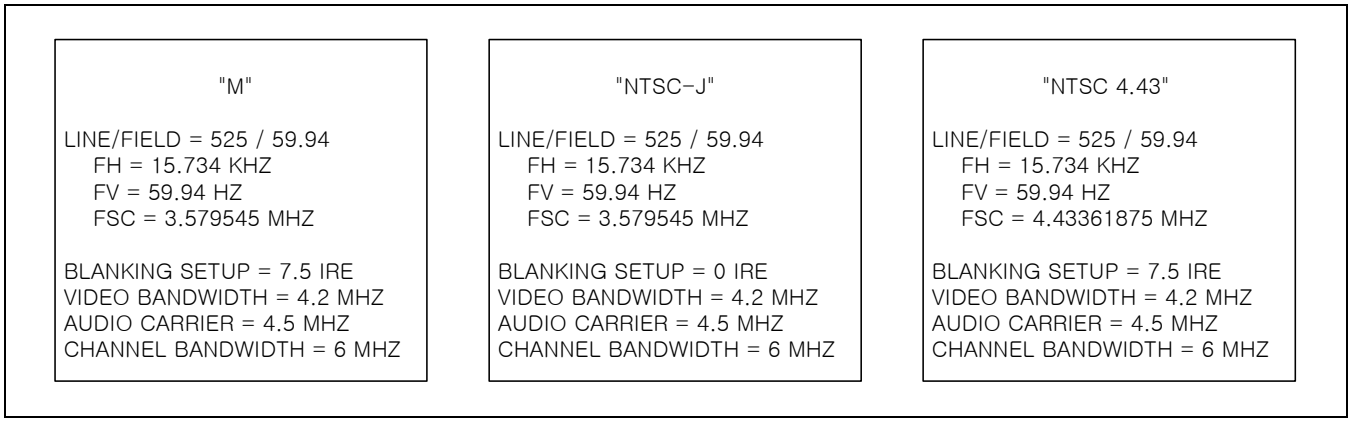


Figure 17-4. Common NTSC System

Types of NTSC are NTSC-M, NTSC-J and NTSC 4.43. U. S. and Republic of Korea use NTSC-M. Japan only uses NTSC-J, which is different with NTSC-M in blanking level (7.5IRE, pedestal). Few South-East Asian countries use NTSC 4.43. NTSC 4.43 has 4.43361875MHz sub-carrier frequency.

17.4.3 COMMON PAL SYSTEM

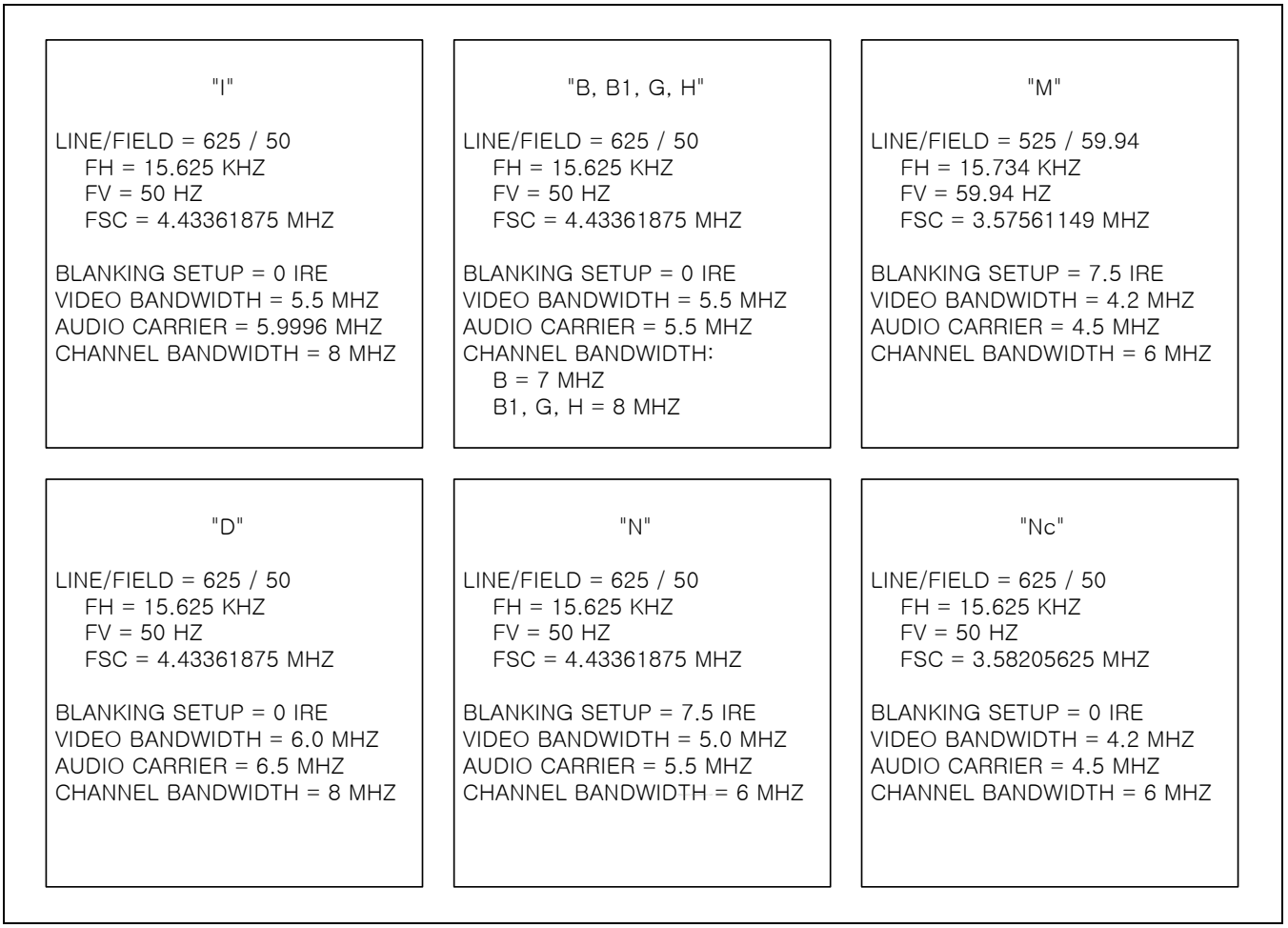


Figure 17-5. Common PAL System

There are several types in PAL system. For more information Refer to Figure 17-5 Common PAL System.

17.4.4 COMPOSITION OF SCREEN

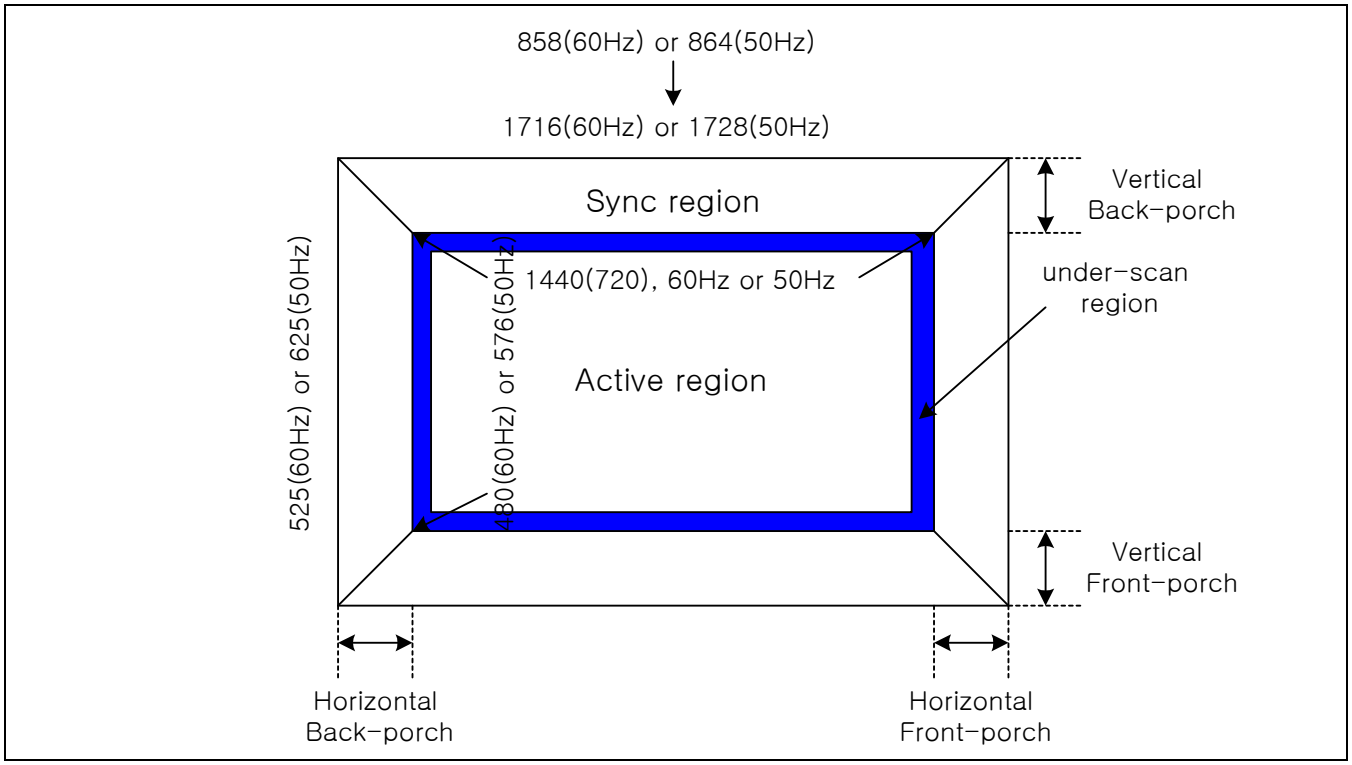


Figure 17-6. Composition of TV screen

In 60Hz type, the size of a frame is 858x525. This contains synchronous and real image region. Real image is 720x480. However, Figure 17-6 do not refer to 720 but 1440. This is because TV encoder, which embedded S3C6410 requires double horizontal data rate. It is used to enhance image.

Otherwise, the size of a frame is 864x625 in 50Hz type and real image is 720x576. It is only different with 50Hz at vertical size.

Additionally, all types have under-scan region. We display images, which are either 720x480 or 720x576, but even we can't watch whole region.

17.4.5 REQUESTED HORIZONTAL TIMING

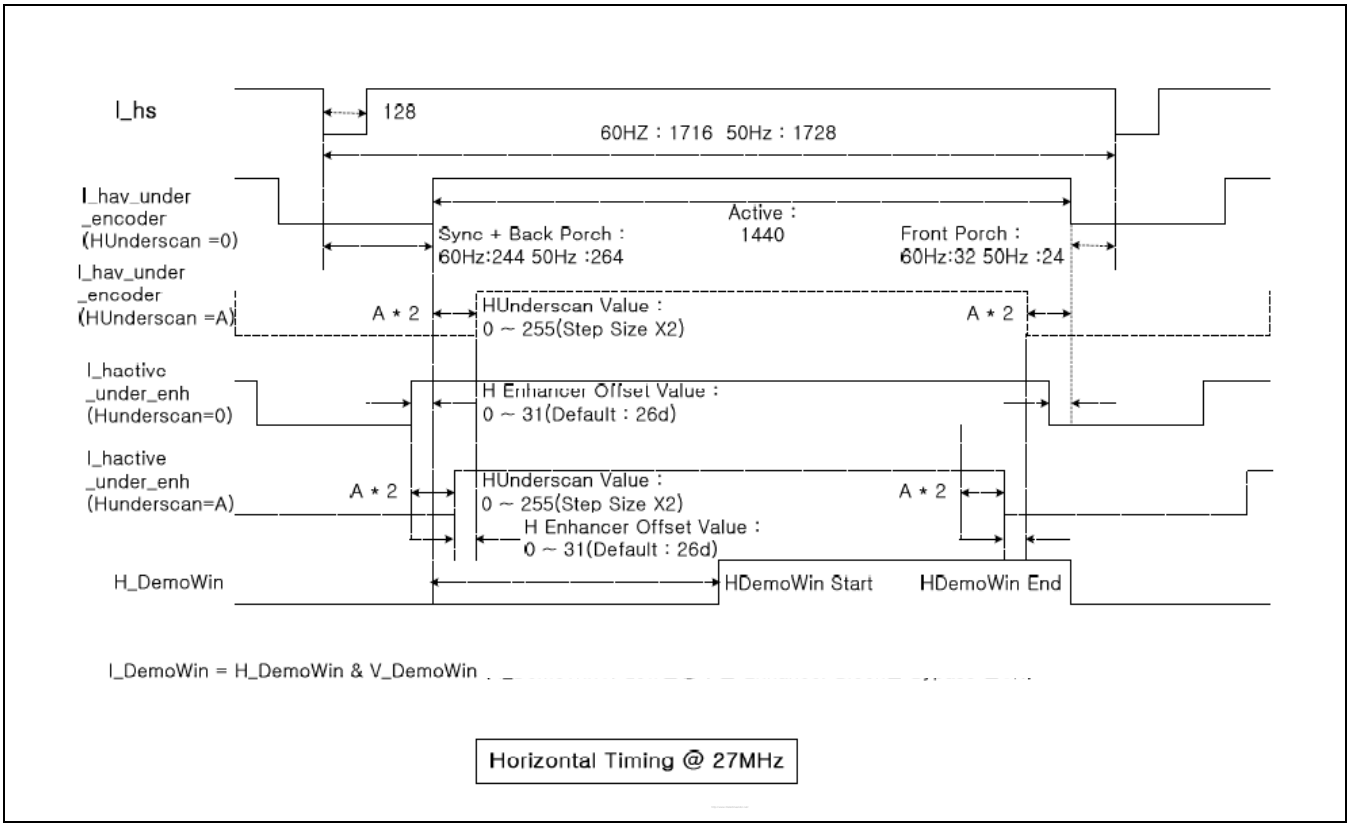


Figure 17-7. Requested Horizontal Timing Diagram

Figure 17-7 shows us the horizontal timing relation. The horizontal line is composed of active and synchronous region. In 60Hz, a line consists of active 1440 pixel, front porch 32 pixels and back porch (containing synch width) 244 pixel. Also, It is possible to configure horizontal under-scan size. Horizontal enhancer offset value means that enhance engine which embedded TV encoder needs 26 clocks to enhance image. Therefore, we transmit data and control signal prior to 26 clocks to adjust timing.

At the same ways, a line of 50Hz type consists of active 1440 pixel, front porch 24 pixels and back porch 264 pixel. The rest is same with 60Hz

17.4.6 REQUESTED VERTICAL TIMING

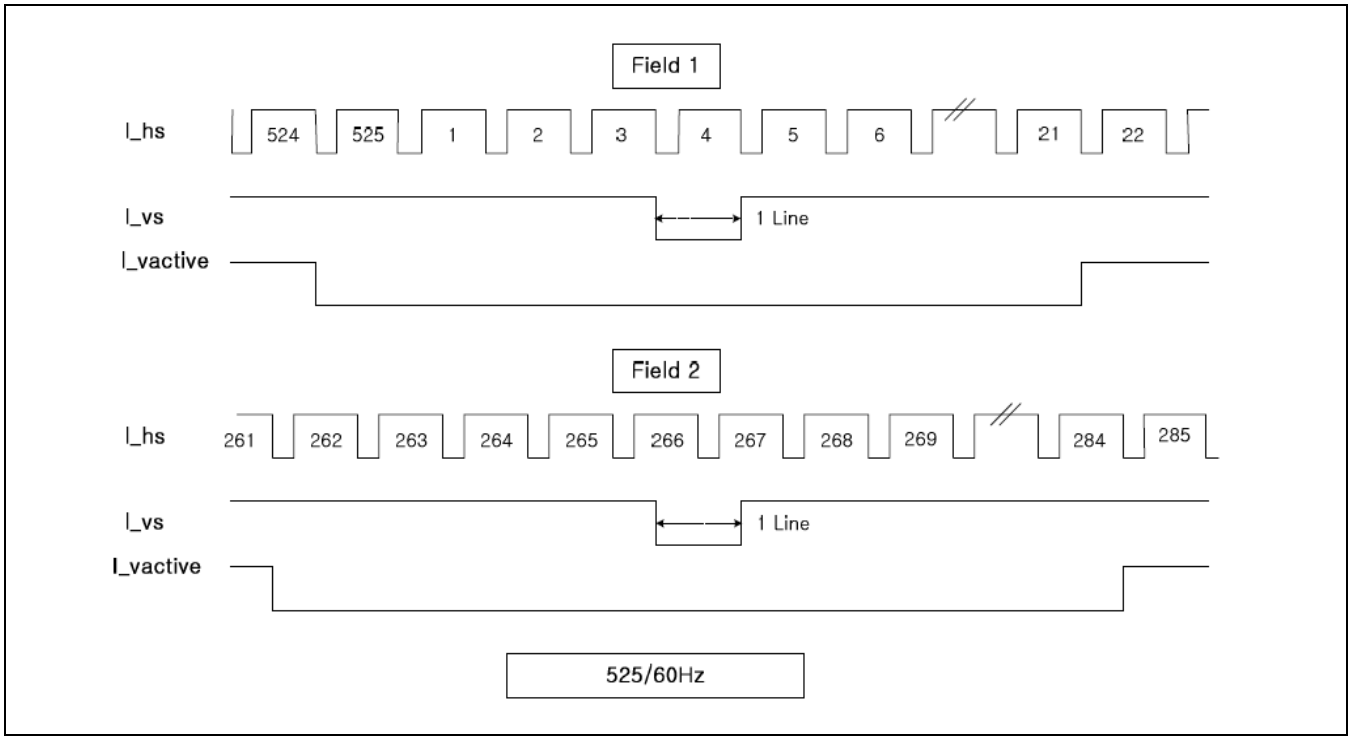


Figure 17-8. Requested Vertical Timing Diagram in 60Hz

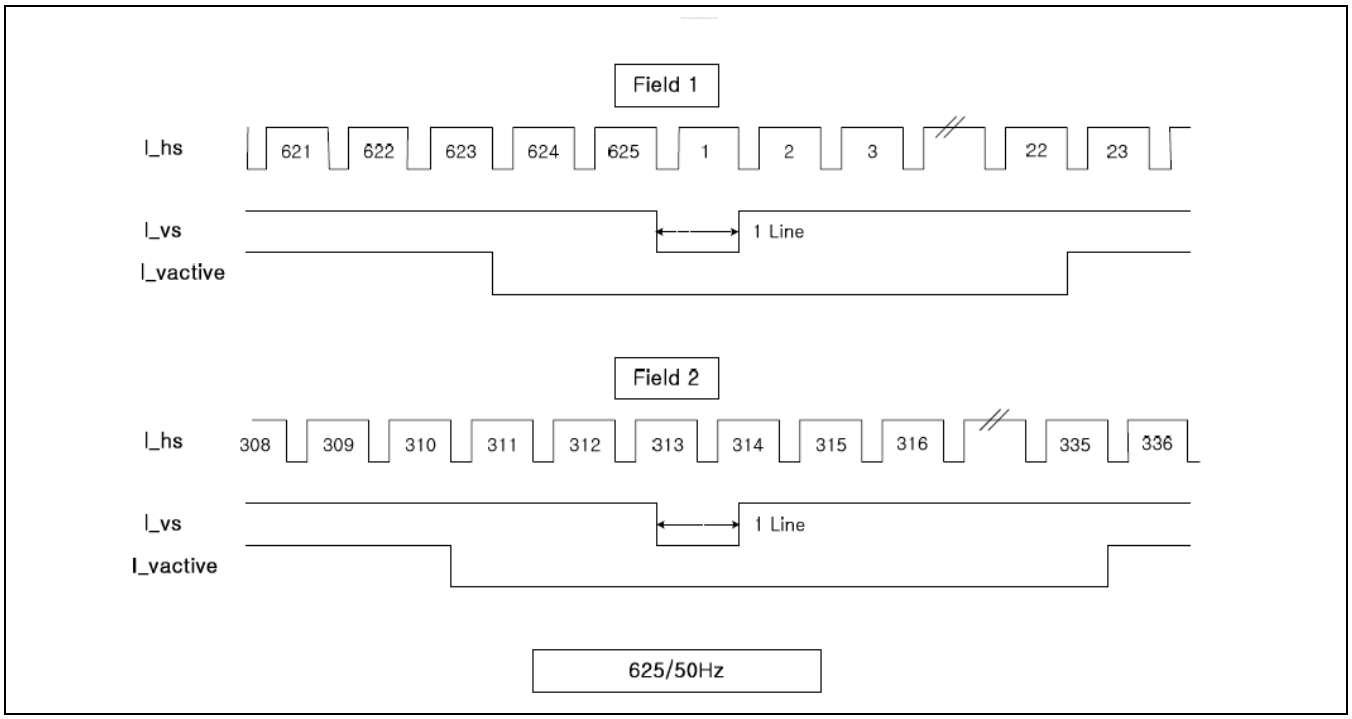


Figure 17-9. Requested Vertical Timing Diagram in 50Hz

17.4.7 MACROVISION (ANTI TAPING)

If you want to use anti-taping function, please contact us via e-mail.

e-mail : apcs.sec@samsung.com

17.5 DAC BOARD CONFIGURE GUIDE

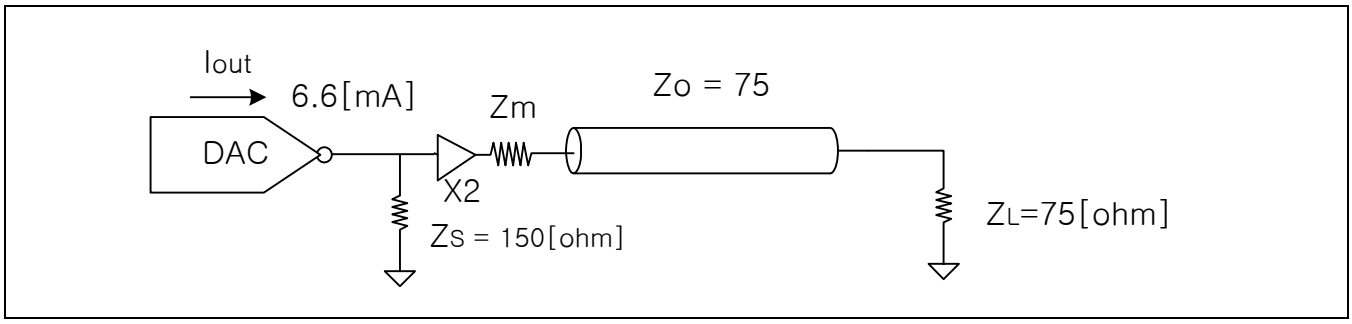


Figure 17-10. DAC Board Guide

It is the current drive DAC in S3C6410. Output current is 6.6mA. Therefore, it has to load 150 [ohm] resistors. It is recommended to use AMP equipment. It prevents electrical damage by ESD.

17.6 TV ENCODER REGISTER SUMMARY

Register	Address	R/W	Description	Reset Value
TVCTRL	0x76200000	R/W	TV Controller control SFR set	0x00010000
VBPORCH	0x76200004	R/W	Vertical back porch end point	0x011C0015
HBPORCH	0x76200008	R/W	Horizontal back porch end point	0x008000F4
HEnhOffset	0x7620000C	R/W	Horizontal enhancer offset	0x0000041A
VDemoWinSize	0x76200010	R/W	Vertical demo window size	0x00F00000
HDemoWinSize	0x76200014	R/W	Horizontal demo window size	0x05A00000
InImageSize	0x76200018	R/W	Input image size	0x01E005A0
PEDCTRL	0x7620001C	R/W	Encoder pedestal control	0x00000000
YCFilterBW	0x76200020	R/W	Y/C filter bandwidth control	0x00000043
HUECTRL	0x76200024	R/W	HUE control	0x00000000
FscCTRL	0x76200028	R/W	Fsc(Sub Carrier Frequency) control	0x00000000
FscDTOManCTRL	0x7620002C	R/W	Fsc DTO manual control	0x00000000
BGCTRL	0x76200034	R/W	Background control	0x00000110
BGHVAVCTRL	0x76200038	R/W	Background VAV & HAV control	0xB400F000
ContraBright	0x76200044	R/W	Contrast & Bright control	0x00000040
CbCrGainCTRL	0x76200048	R/W	Cb & Cr gain control	0x00400040
DemoWinCTRL	0x7620004C	R/W	Demo window control	0x00000010
FTCA	0x76200050	R/W	Flesh tone control	0x00D7008C
BWGAIN	0x76200058	R/W	Black & White stretch gain control	0x00000034
SharpCTRL	0x76200060	R/W	Sharpness control	0x0304501F
GammaCTRL	0x76200064	R/W	Gamma control	0x00000104
FscAuxCTRL	0x76200068	R/W	FSC auxiliary control	0x00000000
SyncSizeCTRL	0x7620006C	R/W	Sync size control	0x0000003D
BurstCTRL	0x76200070	R/W	Burst signal control	0x00690049
MacroBurstCTRL	0x76200074	R/W	Macrovision burst signal control	0x00000041
ActVidPoCTRL	0x76200078	R/W	Active video position control	0x03480078
EncCTRL	0x7620007C	R/W	Encoder control	0x00000011
MuteCTRL	0x76200080	R/W	Mute control	0x80801001

17.7 INDIVIDUAL REGISTER DESCRIPTIONS

17.7.1 TVENCREG1

Register	Address	R/W	Description	Reset Value
TVCTRL	0x76200000	R/W	TV Controller control SFR set	0x00010000

TVCTRL	Bit	Description	Reset Value
Reserved	[31:17]	Reserved	0
INTFIFOUR	[16]	FIFO under-run interrupt control 0 : Disable 1 : Enable	0x1
Reserved	[15:13]	Reserved	0
INTStatus	[12]	FIFO under-run status register If the register is high, the FIFO under-run interrupt is occurred. If you want to clear the interrupt, you also have to write high.	0
Reserved	[11:9]	Reserved	0
TVOUTTYPE	[8]	Select TV out type 0 : Composite out 1 : S-Video out	0
Reserved	[7]	Reserved	0
TVOUTFMT	[6:4]	Select TV Out Format 0 : NTSC-M 1 : NTSC-J 2 : PAL-B/D/G/H/I 3 : PAL-M 4 : PAL-Nc Other : Reserved	0
Reserved	[3:1]	Reserved	0
TVONOFF	[0]	TV encoder on/off control 0 : Off 1 : On	0

17.7.2 TVENCREG2

Register	Address	R/W	Description	Reset Value
VBPORCH	0x76200004	R/W	Vertical back porch end point	0x011C0015

VBPORCH	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0
VEFBPD	[24:16]	Vertical even field back porch end point NTSC : 0x11C(284) , PAL : 0x14F(335)	0x11C
Reserved	[15:8]	Reserved	0
VOFBPD	[7:0]	Vertical odd field back porch end point NTSC : 0x15(21), PAL : 0x16(22)	0x15

17.7.3 TVENCREG3

Register	Address	R/W	Description	Reset Value
HBPORCH	0x76200008	R/W	Horizontal back porch end point	0x008000F4

HBPORCH	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
HSPW	[23:16]	Horizontal sync pulse width Default : 0x80(128) (NTSC, PAL)	0x80
Reserved	[15:11]	Reserved	0
HBPD	[10:0]	Horizontal back porch end point NTSC : 0xF4(244), PAL : 0x108(264)	0xF4

17.7.4 TVENCREG4

Register	Address	R/W	Description	Reset Value
HEnhOffset	0x7620000C	R/W	enhancer offset	0x0000041A

HEnhOffset	Bit	Description	Reset Value
Reserved	[31:30]	Reserved	0
VACTWinCenCTRL	[29:24]	Vertical active window center control	0
HACTWinCenCTRL	[23:16]	Horizontal active window center control	0
Reserved	[15:11]	Reserved	0
DTOffset	[10:8]	Data input timing offset value NTSC, PAL : 0x4(4)	0x4
Reserved	[7:5]	Reserved	0
HEOV	[4:0]	Horizontal enhancer offset value. The enhancer needs 26-cycles before TV encoder starts(0 ~ 31) Default : 0x1A(26) (NTSC, PAL)	0x1A

17.7.5 TVENCREG5

Register	Address	R/W	Description	Reset Value
VDemoWinSize	0x76200010	R/W	Vertical demo window size	0x00F00000

VDemoWinSize	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0
VDWS	[24:16]	Vertical demo window size Default : 0xF0(240)	0xF0
Reserved	[15:9]	Reserved	0
VDWSP	[8:0]	Vertical demo window start point Default : 0x0(0)	0

17.7.6 TVENCREG6

Register	Address	R/W	Description	Reset Value
HDemoWinSize	0x76200014	R/W	Horizontal demo window size	0x05A00000

HDemoWinSize	Bit	Description	Reset Value
Reserved	[31:27]	Reserved	0
HDWEP	[26:16]	Horizontal demo window size Default : 0x5A0(1440)	0x5A0
Reserved	[15:11]	Reserved	0
HDWSP	[10:0]	Horizontal demo window start point Default : 0x0(0)	0

17.7.7 TVENCREG7

Register	Address	R/W	Description	Reset Value
InImageSize	0x76200018	R/W	Input image size	0x01E005A0

InImageSize	Bit	Description	Reset Value
Reserved	[31:26]	Reserved	0
ImageHeight	[25:16]	Height of Input Image.(Max value : 576)	0x1E0
Reserved	[15:11]	Reserved	0
ImageWidth	[10:0]	Width of Input Image. The input value is twice original output image width. For example, you must set 1440 when the image width is 720.(Max Value : 1440)	0x5A0

ENCODER

17.7.8 TVENCREG8

Register	Address	R/W	Description	Reset Value
PEDCTRL	0x7620001C	R/W	Encoder pedestal control	0x00000000

PEDCTRL	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
PEDOff	[0]	Encoder pedestal control 0 : Pedestal on(NTSCM & PALM) 1 : Pedestal off(NTSCJ & PALNc & PALB/D/G/H/I)	0

17.7.9 TVENCREG9

Register	Address	R/W	Description	Reset Value
YCFilterBW	0x76200020	R/W	Y/C filter bandwidth control	0x00000043

YCFilterBW	Bit	Description	Reset Value
Reserved	[31:7]	Reserved	0
YBW	[6:4]	Luminance bandwidth(-3dB) 0 : 6.0 MHz(Recommended at S-Video out) 1 : 3.8 MHz 2 : 3.1 MHz 3 : 2.6 MHz (Recommended at composite out, PAL 4.43MHz notch) 4 : 2.1 MHz (Recommended at composite out, NTSC/PALM 3.58 notch)	0x4
Reserved	[3:2]	Reserved	0
CBW	[1:0]	Chrominance bandwidth 0 : 1.2 MHz 1 : 1.0 MHz 2 : 0.8 MHz 3 : 0.6 MHz	0x3

17.7.10 TVENCREG10

Register	Address	R/W	Description	Reset Value
HUECTRL	0x76200024	R/W	HUE control	0x00000000

HUECTRL	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
HUE	[7:0]	Color HUE control(with an increment of 1.4063') 0x00 : 0' phase shift … 0x80 : 180' phase shift … 0xFF : 358.5938' phase shift	0

17.7.11 TVENCREG11

Register	Address	R/W	Description	Reset Value
FscCTRL	0x76200028	R/W	Fsc(Sub Carrier Frequency) control	0x00000000

FscCTRL	Bit	Description	Reset Value
Reserved	[31:15]	Reserved	0
FscCtrl	[14:0]	Fsc control (+ / -) <Equation> (note> FscCtrl : 2's) = Current DTO set value + FscCtrl[14:0] * (2^9)	0

17.7.12 TVENCREG12

Register	Address	R/W	Description	Reset Value
FscDTOManCTRL	0x7620002C	R/W	Fsc DTO manual control	0x00000000

FscTdoManCTRL	Bit	Description	Reset Value
FscMEn	[31]	Fsc DTO manually control enable	0
FscDTOManual	[30:0]	Fsc DTO manually control <Equation> FscDTOManual[30:0] = FSC * (2^33) / Fclk ex) NTSC 3.5795454545 * (2^33) / 27 = 0x43E0F83E	0

17.7.13 TVENCREG14

Register	Address	R/W	Description	Reset Value
BGCTRL	0x76200034	R/W	Background control	0x00000110

BGCTRL	Bit	Description	Reset Value
Reserved	[31:9]	Reserved	0
SME	[8]	Soft mixed enable 0 : Disable 1 : Enable soft mixed for background border	1
Reserved	[7]	Reserved	0
BGCS	[6:4]	Background color select 0 : Black 1 : Blue 2 : Red 3 : Magenta 4 : Green 5 : Cyan 6 : Yellow 7 : White	1
BGYOFS	[3:0]	Background luminance offset	0

17.7.14 TVENCREG15

Register	Address	R/W	Description	Reset Value
BGHVAVCTRL	0x76200038	R/W	Background VAV & HAV control	0xB400F000

BGHVAVCTRL	Bit	Description	Reset Value
BGHL	[31:24]	Background HAV length (x8) Default : 0xB4(180)	0xB4
BGHS	[23:16]	Background HAV start position (x2) Default : 0x0(0)	0x00
BGVL	[15:8]	Background VAV length (x1) Default : 0xF0(240)	0xF0
BGVS	[7:0]	Background VAV start position (x1) Default : 0x0(0)	0x00

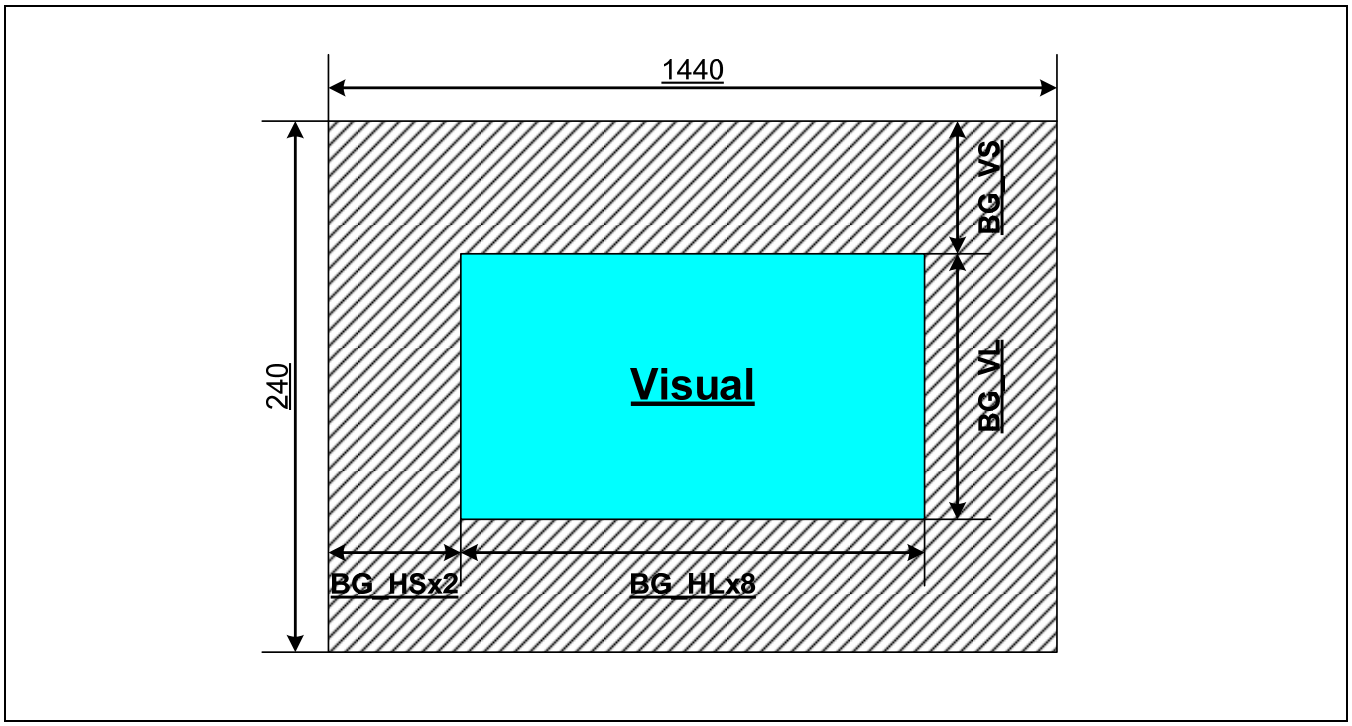


Figure 17-11. The Calculation Method of Back-ground

IMAGE ENHANCER

17.7.15 TVENCREG18

Register	Address	R/W	Description	Reset Value
ContraBright	0x76200044	R/W	Contrast & Bright control	0x00000040

ContraBright	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
BRIGHT	[23:16]	Brightness control(2's complement) 0x7F : Maximum brightness 0x80 : Minimum brightness	0x00
Reserved	[15:8]	Reserved	0
CONTRAST	[7:0]	Contrast gain control(2's complement)	0x40

17.7.16 TVENCREG19

Register	Address	R/W	Description	Reset Value
CbCrGainCTRL	0x76200048	R/W	Cb & Cr gain control	0x00400040

CbCrGainCTRL	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
CRGAIN	[23:16]	Cr gain control(2's complement)	0x40
Reserved	[15:8]	Reserved	0
CBGAIN	[7:0]	Cb gain control(2's complement)	0x40

17.7.17 TVENCREG20

Register	Address	R/W	Description	Reset Value
DemoWinCTRL	0x7620004C	R/W	Demo window control	0x00000010

DemoWinCTRL	Bit	Description	Reset Value
Reserved	[31:25]	Reserved	0
MVDemo	[24]	Enhancer demo window on/off 0 : Normal operation 1 : Enhancer demonstration window mode	0
Reserved	[23:17]	Reserved	0
FreshEn	[16]	Fresh tone correction on/off 0 : Fresh tone correction disable 1 : Fresh tone correction enable	0
Reserved	[15:13]	Reserved	0
BStOff	[12]	Black stretch off control 0 : Black stretch enable 1 : Black stretch disable	0
Reserved	[11:9]	Reserved	0
WStOff	[8]	White stretch off control 0 : White stretch enable 1 : White stretch disable	0
Reserved	[7:2]	Reserved	0
BSGn	[1:0]	Blue stretch gain control 0 : Blue stretch off 3 : Blue stretch max gain	0

17.7.18 TVENCREG21

Register	Address	R/W	Description	Reset Value
FTCA	0x76200050	R/W	Flesh tone control	0x00D7008C

FTCA	Bit	Description	Reset Value
Reserved	[31:24]	Reserved	0
FTCAC	[23:16]	Flesh tone correction angle : Cosine value <Equation> FTCAC = cos(x – 90') * (2^8) (x : 90 ~ 180 degree) ex) x = 123 degree FTCAC = cos(123' – 90')*(2^8) = 0xD7	0xD7
Reserved	[15:8]	Reserved	0
FTCAS	[7:0]	Flesh tone correction angle : Sine value <Equation> FTCAS = sin(x – 90') * (2^8) (x : 90 ~ 180 degree) ex) x = 123 degree FTCAS = sin(123' – 90')*(2^8) = 0x8B	0x8C

17.7.19 TVENCREG23

Register	Address	R/W	Description	Reset Value
BWGAIN	0x76200058	R/W	Black & White stretch gain control	0x00000034

BWGAIN	Bit	Description	Reset Value
Reserved	[31:8]	Reserved	0
WGain	[7:4]	White stretch gain	0x3
BGain	[3:0]	Black stretch gain	0x4

17.7.20 TVENCREG25

Register	Address	R/W	Description	Reset Value
SharpCTRL	0x76200060	R/W	Sharpness control	0x0304501F

SharpCTRL	Bit	Description	Reset Value
Reserved	[31:28]	Reserved	0
SHARPT	[27:20]	Dynamic sharpness tilt point	0x30
Reserved	[19:15]	Reserved	0
SDhCor	[14:12]	Sharpness coring control 0 : Disable coring 7 : Max coring	0x5
Reserved	[11:10]	Reserved	0
DShpF0	[9:8]	Sharpness center frequency.(recommend DShpF0 =0x2 above VGA(640x480) which is the output, DShpF0=0 below QVGA(320x160)). 0 : Low frequency(2.7MHz 3.4MHz 4.5MHz) 2 : High frequency(2.7MHz 3.4MHz 4.5MHz)	0
Reserved	[7:6]	Reserved	0
DShpGn	[5:0]	Dynamic sharpness gain control 0x00 : Reducing the high frequency 0x0F : No sharpness 0x3F : Max sharpness	0x1F

17.7.21 TVENCREG26

Register	Address	R/W	Description	Reset Value
GammaCTRL	0x76200064	R/W	Gamma control	0x00000104

GammaCTRL	Bit	Description	Reset Value
Reserved	[31:13]	Reserved	0
GamEn	[12]	Gamma enable 0 : Gamma disable 1 : Gamma enable	0
Reserved	[11:10]	Reserved	0
GamMode	[9:8]	Gamma control mode 0 : Min gamma gain 3 : Max gamma gain	0x1
Reserved	[7:3]	Reserved	0
DCTRAN	[2:0]	DC tran gain 0 : 80% 5 : 100% 7 : 110%	0x4

17.7.22 TVENCREG27

Register	Address	R/W	Description	Reset Value
FscAuxCTRL	0x76200068	R/W	Fsc auxiliary control	0x00000000

FscAuxCTRL	Bit	Description	Reset Value
Reserved	[31:5]	Reserved	0
Phalt	[4]	Sub-carrier phase alternation control (for PAL) 0 : Phase alternation disable 1 : Phase alternation enable	0
	[3:1]	Reserved	0
Fdrst	[0]	Sub-carrier reset enable 0 : Subcarrier freerun mode 1 : Subcarrier reset mode(Subcarrier resets at every 4 fields)	0

17.7.23 TVENCREG28

Register	Address	R/W	Description	Reset Value
SyncSizeCTRL	0x7620006C	R/W	Sync size control	0x0000003D

SyncSizeCTRL	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0
SySize	[9:0]	Hsync size 0x3D : NTSC 0x3E : PAL	0x3D

17.7.24 TVENCREG29

Register	Address	R/W	Description	Reset Value
BurstCTRL	0x76200070	R/W	Burst signal control	0x00690049

BurstCTRL	Bit	Description	Reset Value
Reserved	[31:26]	Reserved	0
BuEnd	[25:16]	Burst end position 0x69 : NTSC 0x6A : PAL	0x69
Reserved	[15:10]	Reserved	0
BuSt	[9:0]	Burst start position 0x49 : NTSC 0x4A : PAL	0x49

17.7.25 TVENCREG30

Register	Address	R/W	Description	Reset Value
MacroBurstCTRL	0x76200074	R/W	Macrovision burst signal control	0x00000041

BurstCTRL	Bit	Description	Reset Value
Reserved	[31:10]	Reserved	0
BumavSt	[9:0]	Macrovision burst start position 0x41 : NTSC 0x42 : PAL	0x41

17.7.26 TVENCREG31

Register	Address	R/W	Description	Reset Value
ActVidPoCTRL	0x76200078	R/W	Active video position control	0x03480078

ActVidPoCTRL	Bit	Description	Reset Value
Reserved	[31:26]	Reserved	0
AvonEnd	[25:16]	Active video end position 0x348 : NTSC 0x352 : PAL	0x348
Reserved	[15:10]	Reserved	0
AvonSt	[9:0]	Active video start position 0x78 : NTSC 0x82 : PAL	0x78

17.7.27 TVENCREG32

Register	Address	R/W	Description	Reset Value
EncCTRL	0x7620007C	R/W	Encoder control	0x00000011

EncCTRL	Bit	Description	Reset Value
Reserved	[31:1]	Reserved	0
BGEn	[0]	Background enable 0 : Disable 1 : Enable	0x1

17.7.28 TVENCREG33

Register	Address	R/W	Description	Reset Value
MuteCTRL	0x76200080	R/W	Mute control	0x80801001

MuteCTRL	Bit	Description	Reset Value
MuteCr	[31:24]	Mute Cr component	0x80
MuteCb	[23:16]	Mute Cb component	0x80
MuteY	[15:8]	Mute Y component	0x10
Reserved	[7:1]	Reserved	0
MuteOnOff	[0]	Video mute control 0 : Mute enable 1 : Mute disable	0x1

18 GRAPHICS 2D

18.1 OVERVIEW

GRAPHICS-2D is a 2D graphics accelerator that supports three types of primitive drawings: Line/Point Drawing, Bit Block Transfer (BitBLT) and Color Expansion (Text Drawing).

Rendering a primitive takes 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.

18.1.1 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
  - Totally 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

- ➢ 16/24/32-bpp color format support
- ➢ 11.11 fixed point format for coordinate data

18.2 COLOR FORMAT CONVERSION

GRAPHICS-2D supports seven color formats: RGB_565, RGBA_5551, ARGB_1555, RGBA_8888, ARGB_8888, XRGB_8888, and RGBX_8888. The following figure illustrates the structure of each color format.

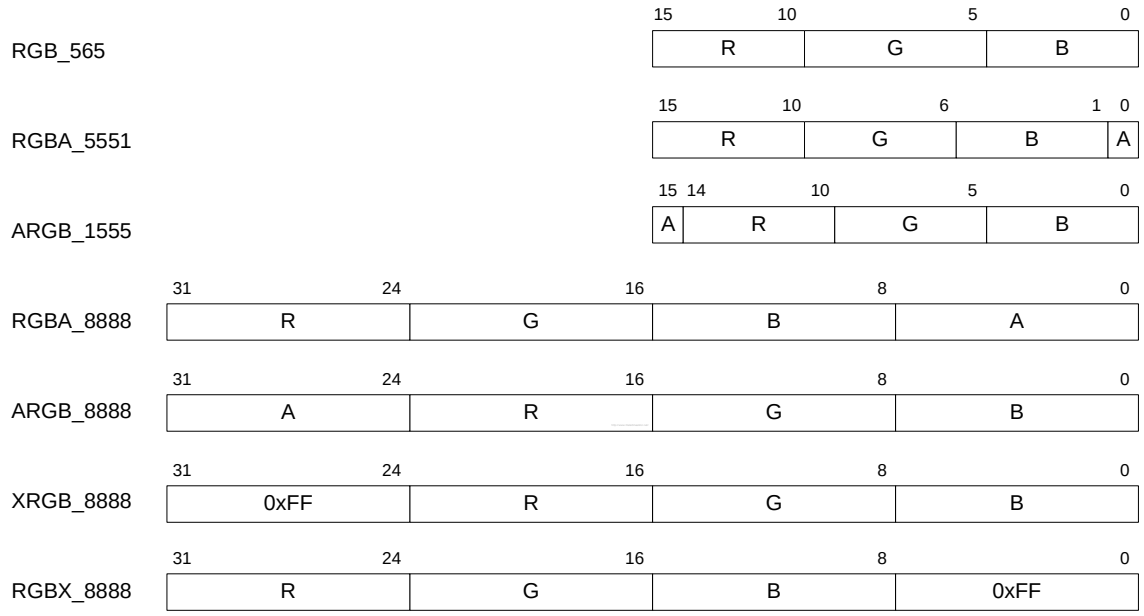


Figure 18-1. Structure of Color Format

The internal computations use ARGB_8888 format. All data (source, destination, foreground, background, blue-screen, pattern) are converted to ARGB_8888 format before computation, and the final result are converted to the color format specified by DEST_COLOR_MODE_REG before writing to frame buffer.

When 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 (8 – *x)* bits of the new field data are padded with the most significant (8 – *x)* bits of the original field data. For example, if the R value in RGB_565 format is 5'b11010, it will be 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).

When 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 will be converted to 5'b11001 in the RGB_565 format, with the three LSBs discarded. Note that, 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.

18.3 COMMAND FIFO

GRAPHICS-2D has a 32-word command FIFO. Every data written to *command registers* and *parameter setting registers* will be pushed into the FIFO first. If the graphics engine is *idle* (no command is being executed), the data will be written to the designated register in one cycle; otherwise, the data will be stored in the FIFO and wait 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 ask graphics engine to give an interrupt signal when the number of entries in FIFO reaches a certain level by setting FIFO_INTC_REG and E bit in INTEN_REG.

18.4 RENDERING PIPELINE

The rendering pipeline of GRAPHICS-2D is illustrated in Figure 18-2. The functionality and related registers of each stage are introduced in detail in the rest of this chapter.

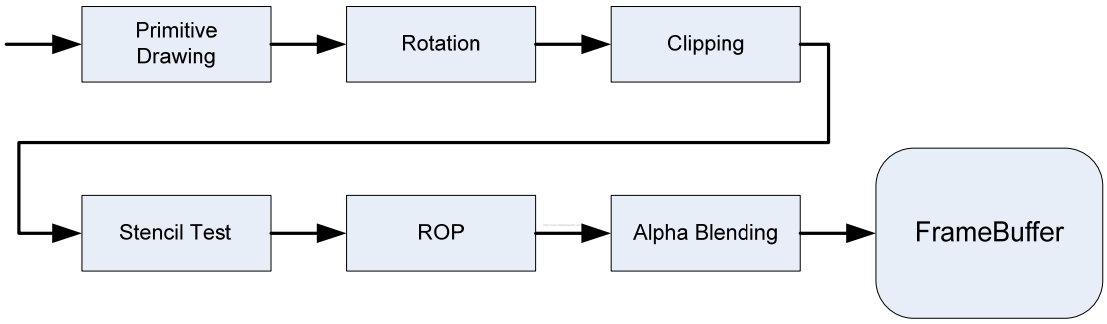


Figure 18-2. GRAPHICS-2D Rendering Pipeline

18.4.1 PRIMITIVE DRAWING

Primitive Drawing determines the pixels to fill, and pass their coordinates to the next stage for further operations.

GRAPHICS-2D supports three types of primitive drawing: 1) line/point drawing; 2) bit block transfer; 3) color expansion.

18.4.1.1 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.

Figure 18-3. X-INCR/Y-INCR format

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
CMDR_0	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 so on. Note that writing to this register starts the rendering process.

18.4.1.2 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 and so on.

On-Screen Rendering

On-screen bit block transfer copies a rectangular block of pixels on screen to another position on the same screen. Note that 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 should 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.

Transparent Mode

GRAPHICS-2D can render image in Transparent Mode. In this mode, the pixels having the same color with

background color (BG_COLOR) are discarded, resulting in a transparent effect. The function of Transparent Mode is illustrated in the images below, in which the BG_COLOR is set to white.

GRAPHICS-2D also support Blue Screen Mode, in which the pixels having the same color with background color (BG_COLOR) are replaced by the blue screen color (BS_COLOR).

GRAPHICS-2D supports both host-to-screen mode and memory-to-screen mode of BLT.

Figure 18-4. Tranparent Mode

Known Issue

In the stretch mode (when source image is scaled), the source coordinates are always rounded to the nearest. This rounding may cause some problem in the boundary when users try to scale the image by integer times. For example, if user wants to scale the image by four times, and set the X_INCR as 0.25, the source coordinates in sequence are 0, 0.25, 0.50, 0.75, 1.0 and so on. However, when the current source coordinate is 0.75, it is rounded to the nearest integer, which is 1, so the first pixel only repeats three times instead of four times. Such problem may not be an issue when both the source and destination are big pictures or when the scale is not an integer, but when it comes to small icons or when user wants every pixel to repeat exactly integer times, an obvious error will occur.

Related Registers

COORD_0	Coordinate of the leftmost topmost coordinate of the source image
COORD_1	Coordinate of the rightmost bottommost coordinate of the source image
COORD_2	Coordinate of the leftmost topmost coordinate of the destination image
COORD_3	Coordinate 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 when S bit in CMDR_1 is disabled or host-to-screen mode is used. X-INCR = (COORD1_X – COORD0_X) / (COORD3_X – COORD2_X)
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)
SRC_BASE_ADDR	The base address of the source image (when 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
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.
CMDR_1	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.
CMDR_2 / CMDR_3	The host provides the source image data through these two command registers. When the host writes the first 32-bit data into CMDR_2, the rendering process starts in the host-to-screen mode. Then the host should provide the rest of data by writing into CMDR_3 continuously.

18.4.1.3 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.

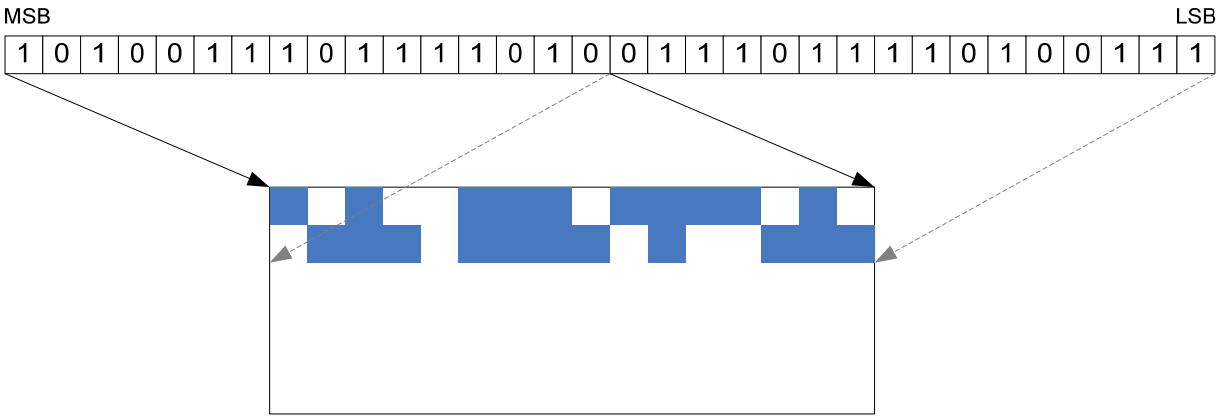


Figure 18-5. Function and Data type of Color Expansion

GRAPHICS-2D can render 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.

Figure 18-6. Transparent effect on Color Expansion

GRAPHICS-2D supports both host-to-screen mode and memory-to-screen mode of Color Expansion.

Related Registers

COORD_0	Coordinate of the leftmost topmost coordinate of the destination window
COORD_1	Coordinate of the rightmost bottommost coordinate of the destination window
FG_COLOR	Foreground Color
BG_COLOR	Background Color
ROP_REG	Enable/disable Transparent Mode
CMDR_3	The base address of the font data. Note that writing to this register starts the rendering process in the memory-to-screen mode.
CMDR_4/CMDR_5	The host provides the font data through these two command registers. When the host writes the first 32-bit data into CMDR_4, the rendering process starts in the host-to-screen mode. Then the host should provide the rest of data by writing them into CMDR_5 continuously.

18.4.2 ROTATION

The pixels can be rotated around the reference point (ox, oy) by 90/180/270 degree clockwise or perform a 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 18-8.

Related Registers

ROT_OC_REG	Coordinate 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

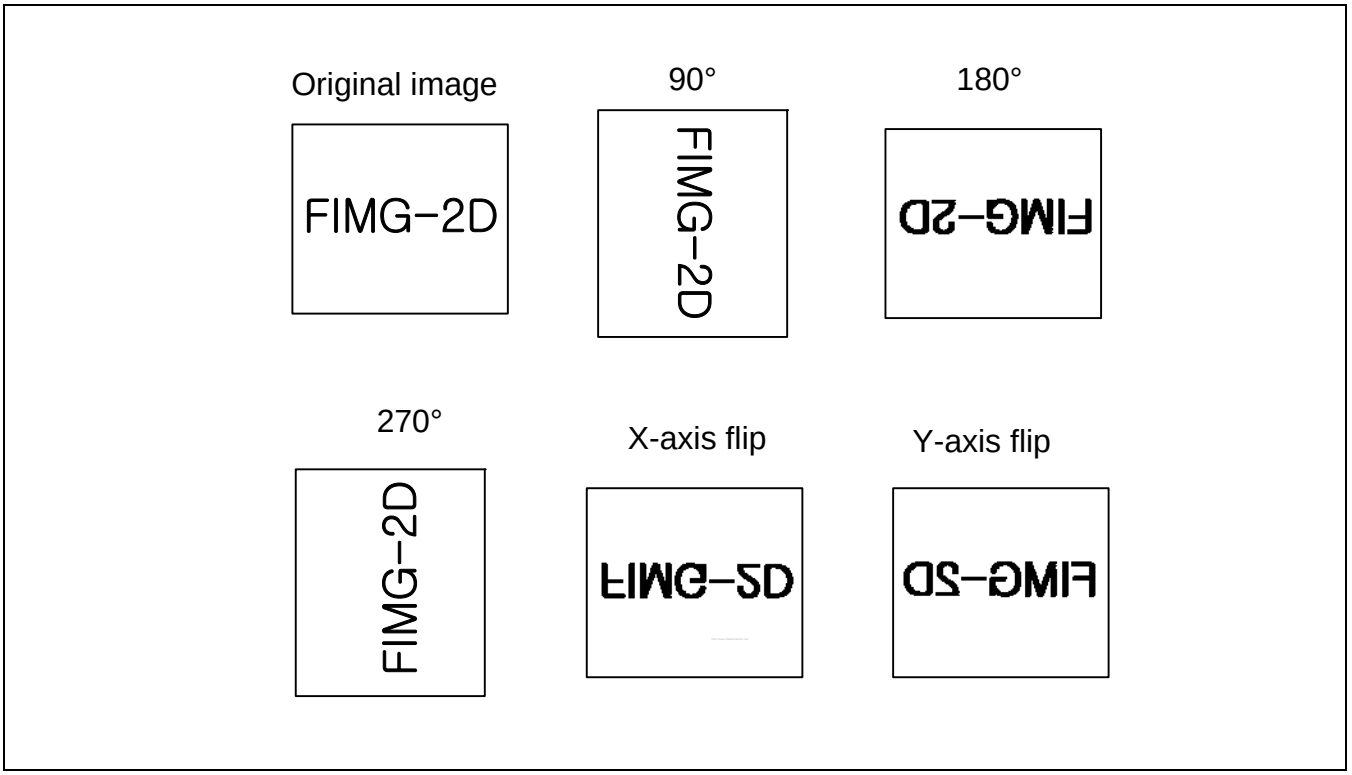


Figure 18-7. Rotation Example

18.4.3 CLIPPING

Clipping discards the pixels (after rotation) outside the clipping window. The discarded pixels will not go through the rest of rendering pipelines.

Related Registers

CW_LT_REG	Coordinate of the leftmost topmost point of the clipping window
CW_RB_REG	Coordinate of the rightmost bottommost point of the clipping window

18.4.4 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.

Related Registers

COLORKEY_CNTL	Stencil Test configurations, such as enable/disable the test and so on.
COLORKEY_DR_MIN	Set the DR(min) value for each field
COLORKEY_DR_MAX	Set the DR(max) value for each field

18.4.5 RASTER OPERATION

Raster Operation performs Boolean operations on three operands: source, destination and 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 should be given 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”.

Related Registers

PATTERN_REG[0:31]	Pattern data
PATOFF_REG	Pattern offset X, Y
ROP_REG	ROP configurations and ROP Value

18.4.6 ALPHA BLENDING

Alpha Blending combines the source color and the destination color in the frame buffer to get the new destination color. GRAPHICS-2D supports 256-level user-specified alpha value and per-pixel alpha blending as well. Fading effect is also supported.

```
User-specified alpha value: ALPHA (from 0 to 255)

[Alpha Blending]
      data = ( source * (ALPHA+1) + destination * (256-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 * (256-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.

18.5 REGISTER DESCRIPTION

Base address : 0x7610_0000				
Register	**Offset**	**R/W**	**Description**	**Reset Value**
General Registers				
CONTROL_REG	0x0000	W	Control register.	0x0000_0000
INTEN_REG	0x0004	R/W	Interrupt Enable register.	0x0000_0000
FIFO_INTC_REG	0x0008	R/W	Interrupt Control register.	0x0000_0018
INTC_PEND_REG	0x000C	R/W	Interrupt Control Pending register.	0x0000_0000
FIFO_STAT_REG	0x0010	R	Command FIFO Status register.	-
Command Registers				
CMD0_REG	0x0100	W	Command register for Line/Point drawing.	-
CMD1_REG	0x0104	W	Command register for BitBLT.	-
CMD2_REG	0x0108	W	Command register for Host to Screen Bitblt transfer start.	-
CMD3_REG	0x010C	W	Command register for Host to Screen Bitblt transfer continue.	-
CMD4_REG	0x0110	W	Command register for Color Expansion. (Host to Screen, Font Start)	-
CMD5_REG	0x0114	W	Command register for Color Expansion. (Host to Screen, Font Continue)	-
CMD6_REG	0x0118	W	Reserved	-
CMD7_REG	0x011C	W	Command register for Color Expansion. (Memory to Screen)	-
Parameter Setting Registers				
Resolution				
SRC_ RES_REG	0x0200	R/W	Source Image Resolution	0x0000_0000
SRC_HORI_RES_REG	0x0204	R/W	Source Image Horizontal Resolution	0x0000_0000
SRC_VERT_RES_REG	0x0208	R/W	Source Image Vertical Resolution	0x0000_0000
SC_RES_REG	0x0210	R/W	Screen Resolution	0x0000_0000
SC_HORI_RES _REG	0x0214	R/W	Screen Horizontal Resolution	0x0000_0000
SC_VERT_RES _REG	0x0218	R/W	Screen Vertical Resolution	0x0000_0000
Clipping Window				
CW_LT_REG	0x0220	R/W	LeftTop coordinates of Clip Window.	0x0000_0000
CW_LT_X_REG	0x0224	R/W	Left X coordinate of Clip Window.	0x0000_0000
CW_LT_Y_REG	0x0228	R/W	Top Y coordinate of Clip Window.	0x0000_0000
CW_RB_REG	0x0230	R/W	RightBottom coordinate of Clip Window.	0x0000_0000
CW_RB_X_REG	0x0234	R/W	Right X coordinate of Clip Window.	0x0000_0000

CW_RB_Y_REG	0x0238	R/W	Bottom Y coordinate of Clip Window.	0x0000_0000
Coordinates				
COORD0_REG	0x0300	R/W	Coordinates 0 register.	0x0000_0000
COORD0_X_REG	0x0304	R/W	X coordinate of Coordinates 0.	0x0000_0000
COORD0_Y_REG	0x0308	R/W	Y coordinate of Coordinates 0.	0x0000_0000
COORD1_REG	0x0310	R/W	Coordinates 1 register.	0x0000_0000
COORD1_X_REG	0x0314	R/W	X coordinate of Coordinates 1.	0x0000_0000
COORD1_Y_REG	0x0318	R/W	Y coordinate of Coordinates 1.	0x0000_0000
COORD2_REG	0x0320	R/W	Coordinates 2 register.	0x0000_0000
COORD2_X_REG	0x0324	R/W	X coordinate of Coordinates 2.	0x0000_0000
COORD2_Y_REG	0x0328	R/W	Y coordinate of Coordinates 2.	0x0000_0000
COORD3_REG	0x0330	R/W	Coordinates 3 register.	0x0000_0000
COORD3_X_REG	0x0334	R/W	X coordinate of Coordinates 3.	0x0000_0000
COORD3_Y_REG	0x0338	R/W	Y coordinate of Coordinates 3.	0x0000_0000
Rotation				
ROT_OC_REG	0x0340	R/W	Rotation Origin Coordinates.	0x0000_0000
ROT_OC_X_REG	0x0344	R/W	X coordinate of Rotation Origin Coordinates.	0x0000_0000
ROT_OC_Y_REG	0x0348	R/W	Y coorindate of Rotation Origin Coordinates.	0x0000_0000
ROTATE_REG	0x034C	R/W	Rotation Mode register.	0x0000_0001
RESERVED	0x0350	R/W	Reserved	0x0000_0000
X,Y Increment Setting				
X_INCR_REG	0x0400	R/W	X Increment register.	0x0000_0000
Y_INCR_REG	0x0404	R/W	Y Increment register.	0x0000_0000
ROP & Alpha Setting				
ROP_REG	0x0410	R/W	Raster Operation register.	0x0000_0000
ALPHA_REG	0x0420	R/W	Alpha value, Fading offset.	0x0000_0000
Color				
FG_COLOR_REG	0x0500	R/W	Foreground Color / Alpha register.	0x0000_0000
BG_COLOR_REG	0x0504	R/W	Background Color register	0x0000_0000
BS_COLOR_REG	0x0508	R/W	Blue Screen Color register	0x0000_0000
SRC_COLOR_MODE_REG	0x0510	R/W	Src Image Color Mode register.	0x0000_0000
DEST_COLOR_MODE_REG	0x0514	R/W	Dest Image Color Mode register	0x0000_0000
Pattern				
PATTERN_REG[0:31]	0x0600 ~0x067C	R/W	Pattern memory.	-
PATOFF_REG	0x0700	R/W	Pattern Offset XY register.	0x0000_0000
PATOFF_X_REG	0x0704	R/W	Pattern Offset X register.	0x0000_0000

PATOFF_Y_REG	0x0708	R/W	Pattern Offset Y register.	0x0000_0000
Stencil Test				
STENCIL_CNTL_REG	0x0720	R/W	Stencil control register	0x0000_0000
STENCIL_DR_MIN_REG	0x0724	W	Stencil decision reference MIN register	0x0000_0000
STENCIL_DR_MAX_REG	0x0728	W	Stencil decision reference MAX register	0xFFFF_FFFF
Image Base Address				
SRC_BASE_ADDR_REG	0x0730	R/W	Source Image Base Address register	0x0000_0000
DEST_BASE_ADDR_REG	0x0734	R/W	Dest Image Base Address register (in most cases, frame buffer address)	0x0000_0000

18.5.1 GENERAL REGISTERS

18.5.1.1 Control Register (CONTROL_REG)

Offset=0x0000, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:1]		0x0
R	[0]	Software Reset Write to this bit results in a one-cycle reset signal to GRAPHICS2D graphics engine. Every command register and parameter setting register will be assigned the “Reset Value”, and the command FIFO will be cleared.	0x0

18.5.1.2 Interrupt Enable REGISTER (INTEN_REG)

Offset=0x0004, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		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.	
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]		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

18.5.1.3 FIFO Interrupt Control REGISTER (FIFO_INTC_REG)

Offset=0x0008, R/W, Reset Value=0x0000_0018

Field	Bit	Description	Initial State
Reserved	[31:6]		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

18.5.1.4 Interrupt Pending REGISTER (INTC_PEND_REG)

Offset=0x000C, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
CLRSEL	[31]	Level interrupt & pulse interrupt mode select. 1 : Level interrupt mode 0: Pulse interrupt mode	0x0
Reserved	[30:11]		-
INTP_CMD_FIN	[10]	Current Command Finished interrupt flag. Writing '1' to this bit clears this flag.	-
INTP_ALL_FIN	[9]	All Commands Finished interrupt flag. Writing '1' to this bit clears this flag.	-
INTP_FULL	[8]	Command FIFO Full interrupt flag. Writing '1' to this bit clears this flag.	-
Reserved	[7:1]		-
INTP_FIFO_LEVEL	[0]	FIFO_USED reaches FIFO_INT_LEVEL interrupt flag. Writing '1' to this bit clears this flag.	-

18.5.1.5 FIFO status REGISTER (FIFO_STAT_REG)

Offset=0x0010, R, Reset Value=0xXXXX_XXXX

Field	Bit	Description	Initial State
Reserved	[31:11]		-
CMD_FIN	[10]	1: The graphics engine finishes the execution of current command. 0: In the middle of rendering process.	
ALL_FIN	[9]	1: Graphics engine is in idle state. The graphics engine finishes 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.	-
FIFO_OVERFLOW	[8]	1: Command FIFO is full, no more commands can be handled 0: Command FIFO is not full.	-
Reserved	[7]		-
FIFO_USED	[6:1]	The number of entries occupied in command FIFO.	-
FIFO_LEVEL_INT	[0]	1: FIFO_USED is greater or equal to FIFO_INT_LEVEL 0: FIFO_USED is smaller than FIFO_INT_LEVEL	-

18.5.2 COMMAND REGISTERS

18.5.2.1 Line Drawing REGISTER (CMD0_REG)

Offset=0x0100, W

Field	Bit	Description	Initial State
Reserved	[31:10]		-
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]		-
L	[1]	0 : Nothing. 1 : Line Drawing.	-
P	[0]	0 : Nothing. 1 : Point Drawing.	-

18.5.2.2 BitBLT REGISTER (CMD1_REG)

Offset=0x0104, W

Field	Bit	Description	Initial State
Reserved	[31:2]		-
S	[1]	0 : Nothing 1 : Stretch BitBLT	-
N	[0]	0 : Nothing 1 : Normal BitBLT	-

18.5.2.3 Host to Screen Start BitBLT REGISTER (CMD2_REG)

Offset=0x0108, W

Field	Bit	Description	Initial State
Data	[31:0]	BitBLT data (Start) Note that 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.	-

18.5.2.4 Host to Screen Continue BitBLT REGISTER (CMD3_REG)

Offset=0x010C, W

Field	Bit	Description	Initial State
Data	[31:0]	BitBLT data (Continue) Note that 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.	-

18.5.2.5 Host to Screen Start Color Expansion REGISTER (CMD4_REG)

Offset=0x0110, W

Field	Bit	Description	Initial State
Data	[31:0]	Color Expansion Data (Start)	-

18.5.2.6 Host to Screen Continue Color Expansion REGISTER (CMD5_REG)

Offset=0x0114, W

Field	Bit	Description	Initial State
Data	[31:0]	Color Expansion Data (Continue)	-

18.5.2.7 Memory to Screen Color Expansion REGISTER (CMD7_REG)

Offset=0x011C, W

Field	Bit	Description	Initial State
Memory Address	[31:0]	Bitmap data base address (used in memory-to-screen mode, should be word-aligned).	-

18.5.3 PARAMETER SETTING REGISTERS
RESOLUTION

18.5.3.1 Source Image Resolution (SRC_RES_REG)

Offset=0x0200, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
VertRes	[26:16]	Vertical resolution of source image. Range: 1 ~ 2040	0x0
Reserved	[15:11]		0x0
HoriRes	[10:0]	Horizontal resolution of source image. Range: 1 ~ 2040.	0x0

18.5.3.2 Source Image Horizontal Resolution (SRC_HORI_RES_REG)

Offset=0x0204, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:1]		0x0
HoriRes	[10:0]	Horizontal resolution of source image. Range: 1 ~ 2040.	0x0

18.5.3.3 Source Image Vertical Resolution (SRC_VERT_RES_REG)

Offset=0x0208, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:1]		0x0
VertRes	[10:0]	Vertical resolution of source image. Range: 1 ~ 2040	0x0

18.5.3.4 Screen Resolution (SC_RES_REG)

Offset=0x0210, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
VertRes	[26:16]	Vertical resolution of the screen. Range: 1 ~ 2040	0x0
Reserved	[15:11]		0x0
HoriRes	[10:0]	Horizontal resolution of the screen. Range: 1 ~ 2040	0x0

18.5.3.5 Screen Horizontal Resolution (SC_HORI_RES_REG)

Offset=0x0214, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
HoriRes	[10:0]	Horizontal resolution of the screen. Range: 1 ~ 2040	0x0

18.5.3.6 Screen Vertical Resolution (SC_VERI_RES_REG)

Offset=0x0218, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:1]		0x0
VeriRes	[10:0]	Vertical resolution of the screen. Range: 1 ~ 2040	0x0

CLIPPING WINDOW

18.5.3.7 LeftTop Clipping Window (CW_LT_REG)

Offset=0x0220, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
TopCW_Y	[26:16]	Top Y Clipping Window Requirement: TopCW_Y < BottomCW_Y	0x0
Reserved	[15:11]		0x0
LeftCW_X	[10:0]	Left X Coordinate of Clipping Window. Requirement: LeftCW_X < RightCW_X	0x0

18.5.3.8 Left X Clipping Window (CW_LT_X_REG)

Offset=0x0224, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
LeftCW_X	[10:0]	Left X Clipping Window Requirement: LeftCW_X < RightCW_X	0x0

18.5.3.9 Top Y Clipping Window (CW_LT_Y_REG)

Offset=0x0228, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
TopCW_Y	[10:0]	Top Y Clipping Window Requirement: TopCW_Y < BottomCW_Y	0x0

18.5.3.10 RightBottom Clipping Window (CW_RB_REG)

Offset=0x0230, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
BottomCW_Y	[26:16]	Bottom Y Clipping Window Requirement: BottomCW_Y < VeriRes (SC_VERI_RES_REG)	0x0
Reserved	[15:11]		0x0
RightCW_X	[10:0]	Right X Clipping Window Requirement: RightCW_X < HoriRes (SC_HORI_RES_REG)	0x0

18.5.3.11 Right X Clipping Window (CW_RB_X_REG)

Offset=0x0234, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
RightCW_X	[10:0]	Right X Clipping Window Requirement: RightCW_X < HoriRes (SC_HORI_RES_REG)	0x0

18.5.3.12 Bottom Y Clipping Window (CW_RB_Y_REG)

Offset=0x0238, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
BottomCW_Y	[10:0]	Bottom Y Clipping Window Requirement: BottomCW_Y < VeriRes (SC_VERI_RES_REG)	0x0

COORDINATES

18.5.3.13 Coordinate_0 REGISTER (COORD0_REG)

Offset=0x0300, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
Y	[26:16]	Coordinate_0 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]		0x0
X	[10:0]	Coordinate_0 X Range: 0 ~ 2039	0x0

18.5.3.14 Coordinate_0 X REGISTER (COORD0_X_REG)

Offset=0x0304, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD0_X	[10:0]	Coordinate_0 X Range: 0 ~ 2039	0x0

18.5.3.15 Coordinate_0 Y REGISTER (COORD0_Y_REG)

Offset=0x0308, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD0_Y	[10:0]	Coordinate_0 Y Range: 0 ~ 2039	0x0

18.5.3.16 Coordinate_1 REGISTER (COORD1_ REG)

Offset=0x0310, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
Y	[26:16]	Coordinate_1 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]		0x0
X	[10:0]	Coordinate_1 X Range: 0 ~ 2039	0x0

18.5.3.17 Coordinate_1 X REGISTER (COORD1_X_REG)

Offset=0x0314, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD1_X	[10:0]	Coordinate_1 X Range: 0 ~ 2039	0x0

18.5.3.18 Coordinate_1 Y REGISTER (COORD1_Y_REG)

Offset=0x0318, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD1_Y	[10:0]	Coordinate_1 Y Range: 0 ~ 2039	0x0

18.5.3.19 Coordinate_2 REGISTER (COORD2_ REG)

Offset=0x0320, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
Y	[26:16]	Coordinate_2 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]		0x0
X	[10:0]	Coordinate_2 X Range: 0 ~ 2039	0x0

18.5.3.20 Coordinate_2 X REGISTER (COORD2_X_REG)

Offset=0x0324, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD2_X	[10:0]	Coordinate_2 X Range: 0 ~ 2039	0x0

18.5.3.21 Coordinate_2 Y REGISTER (COORD2_Y_REG)

Offset=0x0328, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD2_Y	[10:0]	Coordinate_2 Y Range: 0 ~ 2039	0x0

18.5.3.22 Common Resource Coordinate_3 REGISTER (COORD3_ REG)

Offset=0x0330, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
Y	[26:16]	Coordinate_3 Y Range: 0 ~ 2039	0x0
Reserved	[15:11]		0x0
X	[10:0]	Coordinate_3 X Range: 0 ~ 2039	0x0

18.5.3.23 Coordinate_3 X REGISTER (COORD3_X_REG)

Offset=0x0334, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD3_X	[10:0]	Coordinate_3 X Range: 0 ~ 2039	0x0

18.5.3.24 Coordinate_3 Y REGISTER (COORD3_Y_REG)

Offset=0x0338, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
COORD3_Y	[10:0]	Coordinate_3 Y Range: 0 ~ 2039	0x0

ROTATION

18.5.3.25 Rotation Origin Coordinate (ROT_OC_REG)

Offset=0x0340, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:27]		0x0
Y	[26:16]	X coordinate of the reference point of rotation Range: 0 ~ 2039	0x0
Reserved	[15:11]		0x0
X	[10:0]	Y coordinate of the reference point of rotation Range 0 ~ 2039	0x0

18.5.3.26 Rotation Origin Coordinate X (ROT_OC_X_REG)

Offset=0x0344, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:11]		0x0
ROT_OC_X	[10:0]	X coordinate of the reference point of rotation Range: 0 ~ 2039	0x0

18.5.3.27 Rotation Origin Coordinate Y (ROT_OC_Y_REG)

Offset=0x0348, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:1]		0x0
ROT_OC_Y	[10:0]	Y coordinate of the reference point of rotation Range 0 ~ 2039	0x0

18.5.3.28 Rotation REGISTER (ROTATE_REG)

Offset=0x034C, R/W, Reset Value=0x0000_0001

Field	Bit	Description	Initial State
Reserved	[31:6]		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.

X,Y INCREMENT SETTING

18.5.3.30 X Increment REGISTER (X_INCR_REG)

Offset=0x0400, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:22]		0x0
X_INCR	[21:0]	X increment value (2's complement, 11-digit fraction)	0x0

18.5.3.31 Y Increment REGISTER (Y_INCR_REG)

Offset=0x0404, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:22]		0x0
Y_INCR	[21:0]	Y increment value (2's complement, 11-digit fraction)	0x0

ROP & ALPHA SETTING

18.5.3.32 Raster Operation REGISTER (ROP_REG)

Offset=0x0410, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:14]		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 that Perpixel Alpha Blending can only be applied on 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 that T and B must not be set at the same time.	0x0
ROP Value	[7:0]	Raster Operation Value	0x0

18.5.3.33 Alpha REGISTER (ALPHA_REG)

Offset=0x0420, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:16]		0x0
Fading	[15:8]	Fading Offset Value	0x0
Alpha	[7:0]	Alpha Value	0x0

COLOR

18.5.3.34 Foreground Color REGISTER (FG_COLOR_REG)

Offset=0x0500, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
ForegroundColor	[31:0]	Foreground Color Value. The alpha field of the foreground color will be discarded.	0x0

18.5.3.35 Background Color REGISTER (BG_COLOR_REG)

Offset=0x0504, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
BackgroundColor	[31:0]	Background Color Value. The alpha field of the background color will be discarded.	0x0

18.5.3.36 BlueScreen Color REGISTER (BS_COLOR_REG)

Offset=0x0508, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
BlueScreenColor	[31:0]	BlueScreen Color Value. The alpha field of the blue screen color will be discarded.	0x0

18.5.3.37 Source Image Color Mode (SRC_COLOR_MODE_REG)

Offset=0x0510, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:3]	Should be 0	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

18.5.3.38 Destination Image Color Mode (DEST_COLOR_MODE_REG)

Offset=0x0514, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:3]		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

PATTERN

18.5.3.39 Pattern Offset REGISTER (PATOFF_REG)

Offset=0x0700, R/W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:19]		0x0
POffsetY	[18:16]	Pattern Offset Y Value	0x0
Reserved	[15:3]		0x0
POffsetX	[2:0]	Pattern OffsetX Value	0x0

18.5.3.40 Pattern Offset X REGISTER (PATOFF_X_REG)

Offset=0x0704, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:3]		0x0
POffsetX	[2:0]	Pattern OffsetX Value	0x0

18.5.3.41 Pattern Offset Y REGISTER (PATOFF_Y_REG)

Offset=0x0708, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:3]		0x0
POffsetY	[2:0]	Pattern OffsetY Value	0x0

STENCIL TEST

18.5.3.42 Colorkey Control REGISTER (COLORKEY_CNTL_REG)

Offset=0x0720, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
Reserved	[31:5]		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

18.5.3.43 Colorkey Decision Reference Minimum REGISTER (COLORKEY _DR_MIN_REG)

Offset=0x0724, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
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

18.5.3.44 Colorkey Decision Reference Maximum REGISTER (COLORKEY _DR_MAX_REG)

Offset=0x0728, W, Reset Value=0xFFFF_FFFF

Field	Bit	Description	Initial State
A_DR(max)	[31:24]	Alpha DR MAX value	0xF
R_DR(max)	[23:16]	RED DR MAX value	0xF
G_DR(max)	[15:8]	GREEN DR MAX value	0xF
B_DR(max)	[7:0]	BLUE DR MAX value	0xF

IMAGE BASE ADDRESS

18.5.3.45 Source Image Base Address REGISTER (SRC_BASE_ADDR_REG)

Offset=0x0730, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
ADDR	[31:0]	Base address of the source image	0x0

18.5.3.46 Destination Image Base Address REGISTER (DEST_BASE_ADDR_REG)

Offset=0x0734, W, Reset Value=0x0000_0000

Field	Bit	Description	Initial State
ADDR	[31:0]	Base address of the destination image (in most cases, it is also the frame buffer base address).	0x0

19 IMAGE ROTATOR

19.1 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.

19.2 FEATURE

- Supports image format: YCbCr 4:2:2(interleave), YCbCr 4:2:0(non-interleave), RGB565 and RGB888(unpacked)
- Supports rotate degree: 90, 180, 270, flip vertical and flip horizontal

19.3 BLOCK DIAGRAM

Follow Figure 19-1 shows the block diagram of Image Rotator.

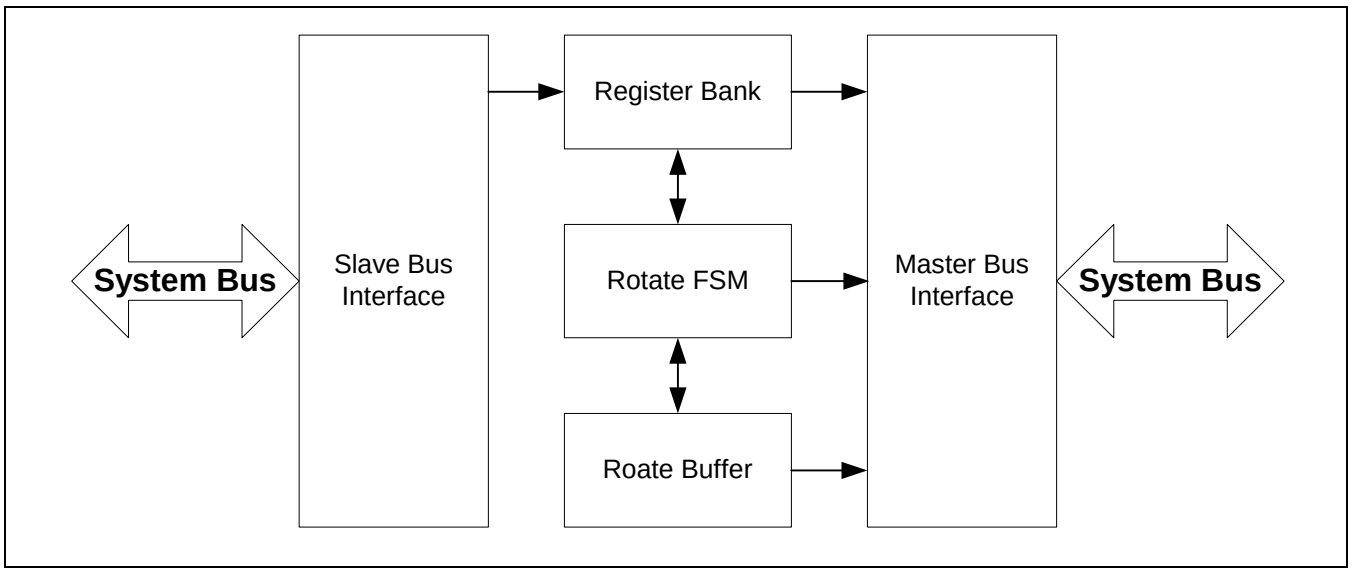


Figure 19-1. Image Rotator Block Diagram

19.4 IMAGE EXAMPLE

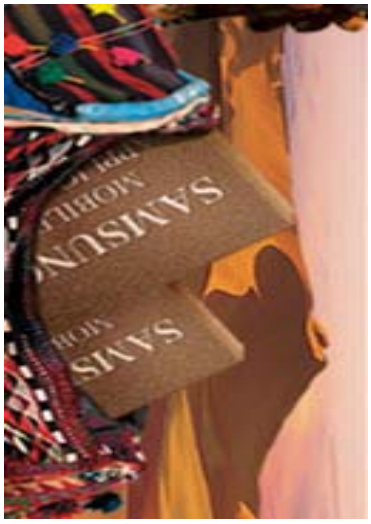
(a) Original Image

(b) Flip Vertical

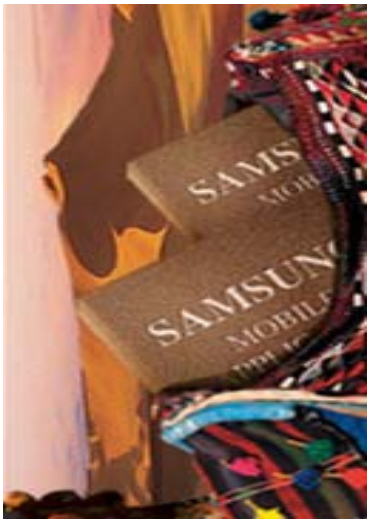
(c) Flip Horizontal

(d) 180-degree Rotation

(e) 90-degree Rotation



(f) 270-degree Rotation



19.5 REGISTER DESCRIPTION

19.5.1 MEMORY MAP

Register	Address	R/W	Description	Reset Value
CTRLCFG	0x7720_0000	R/W	Rotator Control Register	0x000_0000
SRCADDRREG0	0x7720_0004	R/W	Rotator Source Image (RGB or Y component) Address Register	0x0000_0000
SRCADDRREG1	0x7720_0008	R/W	Rotator Source Image (CB component) Address Register	0x0000_0000
SRCADDRREG2	0x7720_000C	R/W	Rotator Source Image (CR component) Address Register	0x0000_0000
SRCSIZEREG	0x7720_0010	R/W	Rotator Source Image Size Register	0x0000_0000
DESTADDRREG0	0x7720_0018	R/W	Rotator Destination Image (RGB or Y component) Address Register	0x0000_0000
DESTADDRREG1	0x7720_001C	R/W	Rotator Destination Image (CB component) Address Register	0x0000_0000
DESTADDRREG2	0x7720_0020	R/W	Rotator Destination Image (CR component) Address Register	0x0000_0000
STATCFG	0x7720_002C	R	Rotator Status Register	0x0000_0000

19.5.2 ROTATOR CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CTRLCFG	0x7720_0000	R/W	Rotator Control Register	0x000_0000

CTRLREG	Bit	Description	Initial State
Reserved	[31:25]	Reserved	000b
Enable Int.	[24]	Interrupt Enable 0: Disable interrupt 1: Enable 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]	Determine the rotation degree 00: Don't rotate 01: 90 degree 10: 180 degree 11: 270 degree Note: These bits should be zero, if flip direction isn't zero.	00b
Flip Direction	[5:4]	Determine the flip direction 00: Don't flip 10: Flip vertical 11: Flip horizontal Note: These bits should be zero, if rotation degree isn't zero.	00b
Reserved	[3:1]	Reserved	000b
Start rotate	[0]	Rotate enable signal, when this bit is set, Rotator starts the operation. This bit is cleared when rotator starts to move image. 0: Don't work 1: Start rotate operation	0b

19.5.3 ROTATOR SOURCE IMAGE ADDRESS REGISTER 0 (RGB OR Y COMPONENT)

Register	Address	R/W	Description	Reset Value
SRCADDRREG0	0x7720_0004	R/W	Rotator Source Image Address Register	0x0000_0000

SRCADDRREG0	Bit	Description	Initial State
Source Address	[30:0]	The address of source image.	0x0000_0000

19.5.4 ROTATOR SOURCE IMAGE ADDRESS REGISTER 1 (CB COMPONENT)

Register	Address	R/W	Description	Reset Value
SRCADDRREG1	0x7720_0008	R/W	Rotator Source Image Address Register	0x0000_0000

SRCADDRREG1	Bit	Description	Initial State
Source Address	[30:0]	The address of source image.	0x0000_0000

19.5.5 ROTATOR SOURCE IMAGE ADDRESS REGISTER 2 (CR COMPONENT)

Register	Address	R/W	Description	Reset Value
SRCADDRREG2	0x7720_000C	R/W	Rotator Source Image Address Register	0x0000_0000

SRCADDRREG2	Bit	Description	Initial State
Source Address	[30:0]	The address of source image.	0x0000_0000

19.5.6 ROTATOR SOURCE IMAGE SIZE REGISTER

Register	Address	R/W	Description	Reset Value
SRCSIZEREG	0x7720_0010	R/W	Rotator Source Image Size Register	0x0000_0000

SRCSIZEREG	Bit	Description	Initial State
Vertical Size	[31:16]	Vertical Image size of source image Vertical image size = this value	0x0000
Horizontal Size	[15:0]	Horizontal Image size of source image Horizontal image size = this value	0x0000

19.5.7 ROTATOR DESTINATION IMAGE ADDRESS REGISTER 0 (RGB OR Y COMPONENT)

Register	Address	R/W	Description	Reset Value
DESTADDRREG0	0x7720_0018	R/W	Rotator Destination Image Address Register	0x0000_0000

DESTADDRREG0	Bit	Description	Initial State
Destination Address	[30:0]	The address of destination image.	0x0000_0000

19.5.8 ROTATOR DESTINATION IMAGE ADDRESS REGISTER 1 (CB COMPONENT)

Register	Address	R/W	Description	Reset Value
DESTADDRREG1	0x7720_001C	R/W	Rotator Destination Image Address Register	0x0000_0000

DESTADDRREG1	Bit	Description	Initial State
Destination Address	[30:0]	The address of destination image.	0x0000_0000

19.5.9 ROTATOR DESTINATION IMAGE ADDRESS REGISTER 2 (CR COMPONENT)

Register	Address	R/W	Description	Reset Value
DESTADDRREG2	0x7720_0020	R/W	Rotator Destination Image Address Register	0x0000_0000

DESTADDRREG2	Bit	Description	Initial State
Destination Address	[30:0]	The address of destination image.	0x0000_0000

19.5.10 ROTATOR STATUS REGISTER

Register	Address	R/W	Description	Reset Value
STATCFG	0x7720_002C	R	Rotator Status Register	0x0000_0000

STATREG	Bit	Description	Initial State
Current line number	[31:16]	Indicate where rotator accesses image. This value shows what line number of image is handled.	0
Reserved	[15:9]	Reserved	0x00
Interrupt Pending	[8]	This bit is set whenever a image rotation is finished. And cleared by read this bit.	0
Reserved	[7:2]	Reserved	0x00
Rotator status	[1:0]	This bits show what kinds of operation are doing by rotator. 00:There is no work (IDLE) 01: Reserved 10: Rotating a image (BUSY) 11: Rotating a image, and has one more job to rotate (BUSY)	0

Note: If rotator status is 2'b11, another job is pending while one job is rotating. So, if you want to set another job on rotating, you should set Input Image format, Rotation Degree, Flip Direction and Start rotate at BUSY state (Rotator status=2'b10).

20 CAMERA INTERFACE

20.1 OVERVIEW

This specification defines the interface of camera. The CAMERA INTERFACE in S3C6410X (Fully Interactive Mobile Camera interface 3.2) supports ITU R BT-601/656 YCbCr 8-bit standard. Maximum input size is 4096x4096 pixels. The CAMERA INTERFACE in S3C6410X consists of several functions. *T_patternMux is* the test pattern generator. Test pattern generation can be used to calibration of input sync signals as HREF and VSYNC. *CatchCam* is the capturing ITU signal and window cut. Video sync signals and pixel clock polarity can be inverted in the camera interface side by using register setting. *Two scalers* exist. The one is the preview scaler, which is dedicated to generate smaller size image for preview. The other one is the codec scaler, which is dedicated to generate codec useful image. *Two Channel MSDMA(Memory Scaling DMA)*can read the memory data for each scaling path. *Two Output DMAs exist. The one is the Preview DMA. The other one is the Codec DMA.* Two DMA are dedicated to the YCbCr 4:2:2, YCbCr 4:2:0 and RGB output. The CAMERA INTERFACE in S3C6410X has *image rotator (90' clockwise)* and *image effect*. These features are very useful in folder type cellular phone.

20.2 FEATURES

The Camera Interface supports the following features:

- ITU-R BT 601/656 8-bit mode support
- DZI (Digital Zoom In) capability
- Programmable polarity of video sync signals
- Maximum. 4096 x 4096 pixels Camera input support (refer to the table 20-1 for scaler max. size)
- Codec / Preview Image mirror and rotation(only preview) (X-flip, Y-flip, 90° ,180° and 270° rotation)
- Codec / Preview output image generation (RGB 16/18/24-bit format and YCbCr 4:2:0/4:2:2 format)
- Camera image capture frame control support
- Scan line offset support
- YCbCr 4:2:2 image interleave format support
- Image effect support
- LCD controller direct path support (MSDMA only)
- Interlace Camera input support.

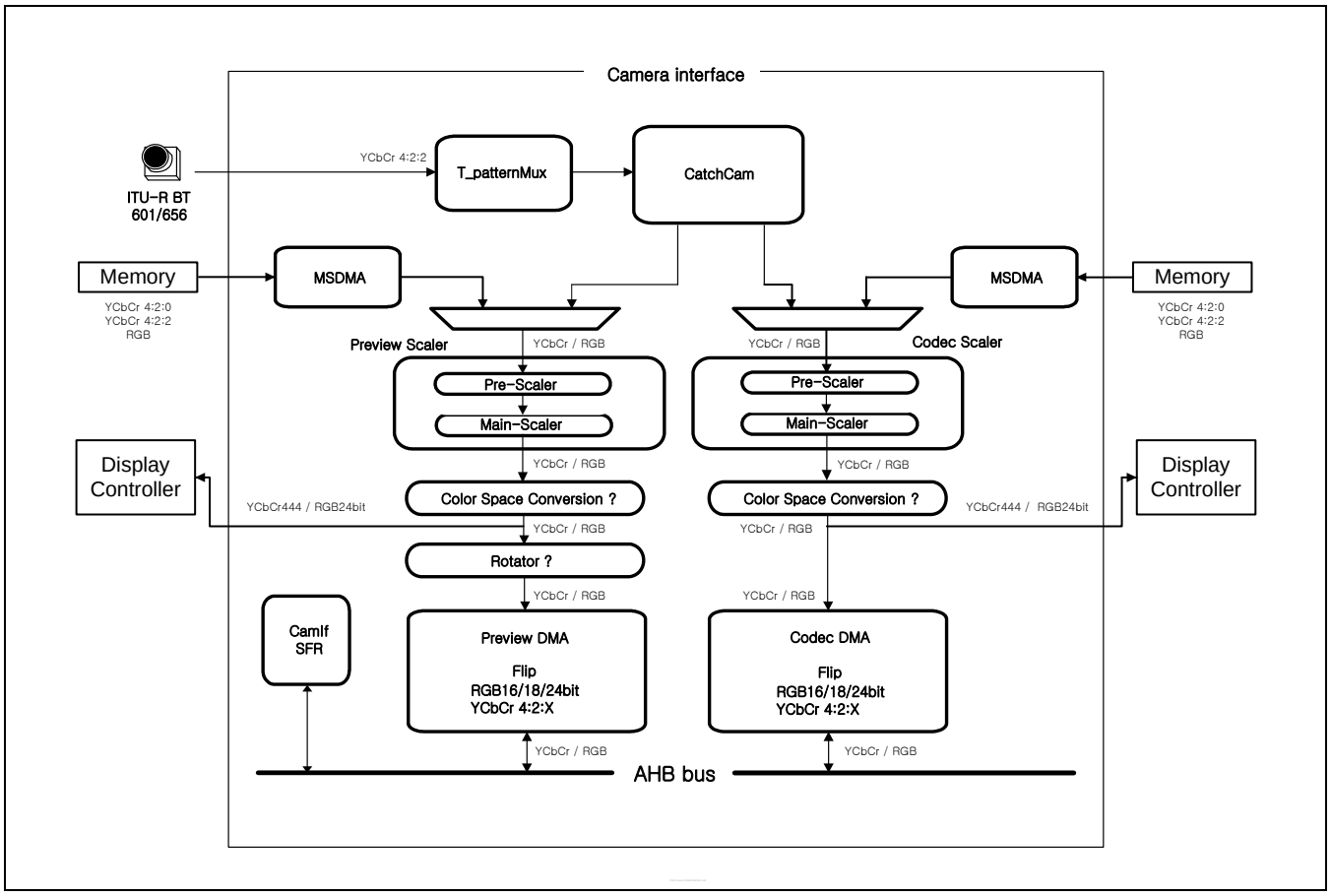


Figure 20-1. Camera interface overview

Table 20-1. Preview & Codec max. Horizontal size

	Preview	Codec
Prescaled input Max Hsize	720 pixels	2048 pixels
Scaler bypass	4096 pixels	4096 pixels
TargetHsize (no rotation)	4096 pixels (Bypass YCbCr) **720** pixels (Except Bypass)	4096 pixels (Bypass YCbCr) 2048 pixels (Except Bypass)
TargetHsize (with rotation)	720 pixels (RGB) 360 pixels (YCbCr)	

20.3 EXTERNAL INTERFACE

Camera Interface can support 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

20.4 SIGNAL DESCRIPTION

Table 20-2. Camera interface signal description

Name	I/O	Description
External camera processor interface signal		
XciPCLK	I	Pixel Clock, driven by the Camera processor A
XciVSYNC	I	Frame Sync, driven by the Camera processor A
XciHREF	I	Horizontal Sync, driven by the Camera processor A
XciYDATA [7:0]	I	Pixel Data driven by the Camera processor A
XciRSTn	O	Software Reset or Power Down for the Camera processor A
XciCLK	O	Clock for a external ISP

NOTE: I/O direction. I: input, O: output, B: bi-direction

20.5 TIMING DIAGRAM

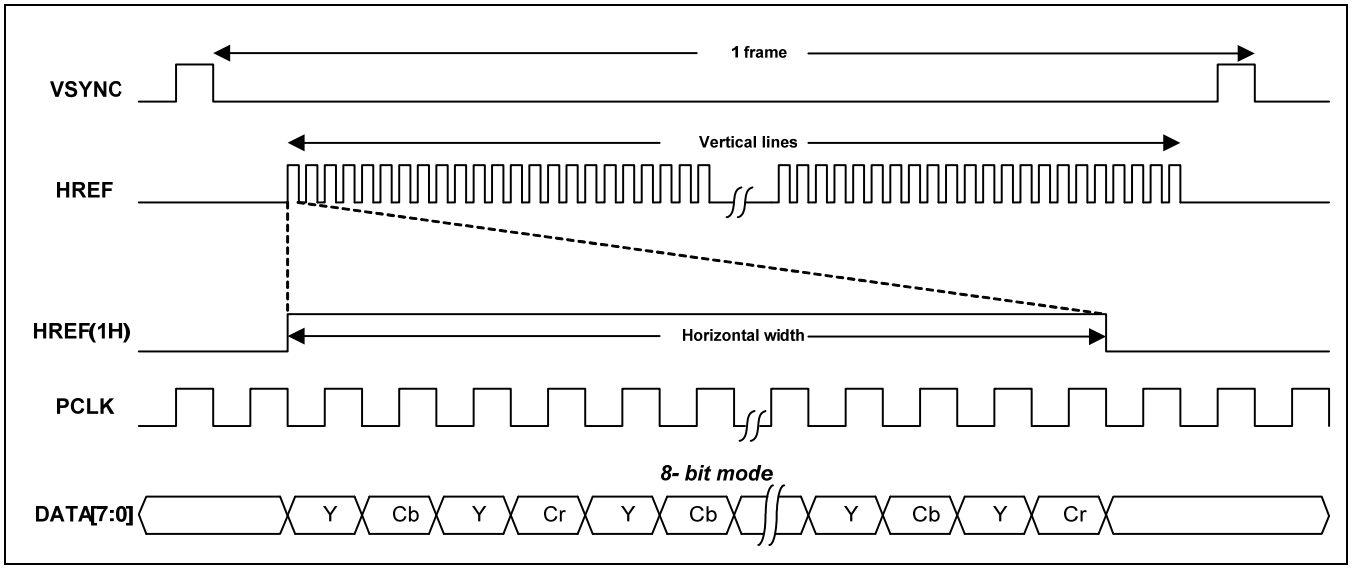


Figure 20-2. ITU-R BT 601 Input timing diagram

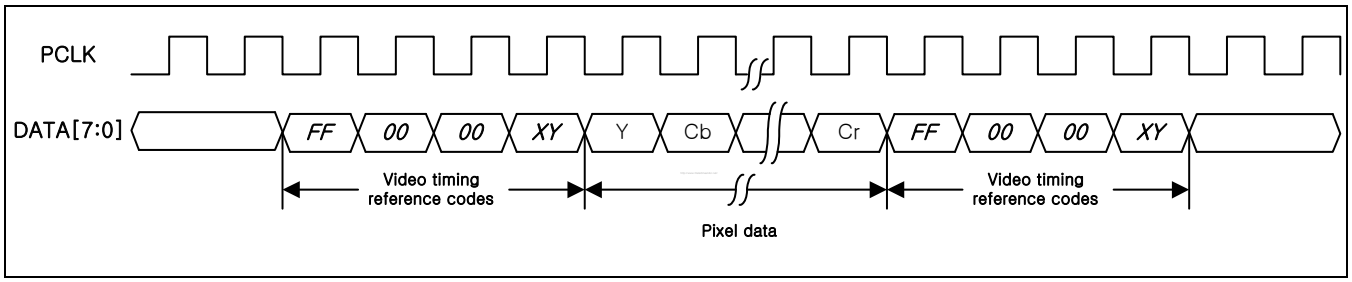


Figure 20-3. ITU-R BT 656 Input timing diagram

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 20-3 and Table 20-3.

Table 20-3. 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 can catch the video sync bits like H (SAV, EAV) and V (Frame Sync) after reserved data as "FF-00-00".

Caution! All external camera interface IOs must not be combined with any other GPIO or bi-directional ports.

Caution! All external camera interface IOs are recommended to be shmitt-trigger type IO for noise reduction.

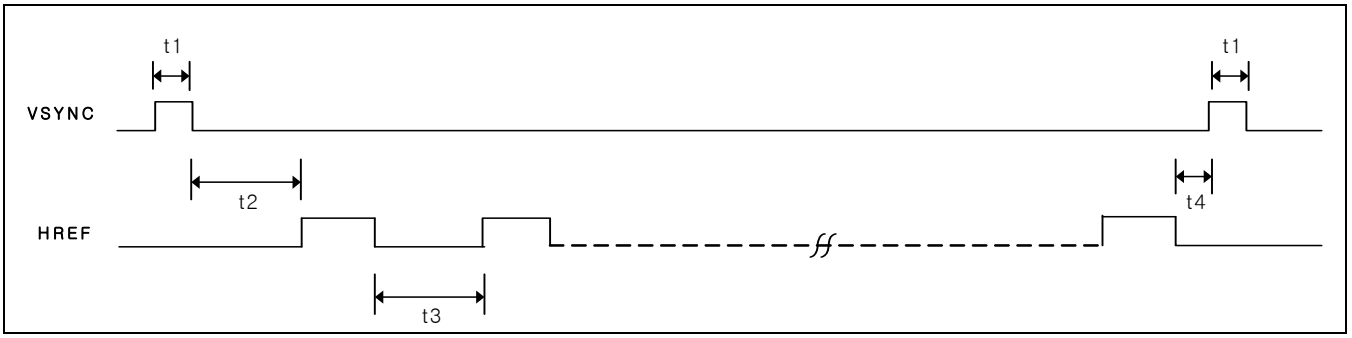


Figure 20-4. Sync signal timing diagram

Table 20-4. 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.

20.6 EXTERNAL/INTERNAL CONNECTION GUIDE

All Camera Interface input signals must not occur inter-skewing to pixel clock line. Recommend next pin location and routing.

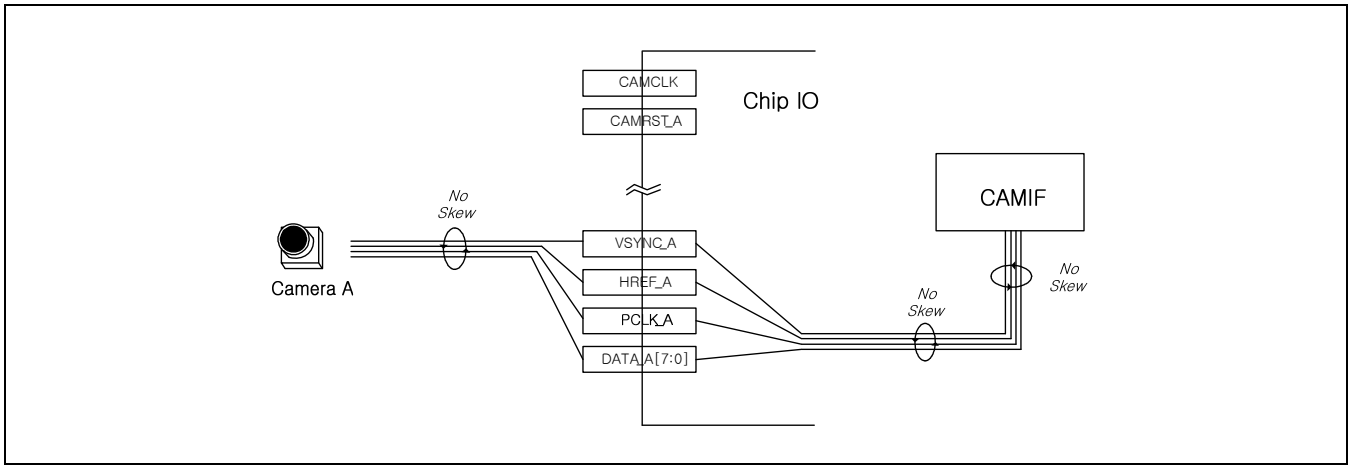


Figure 20-5. IO connection guide

20.7 CAMERA INTERFACE OPERATION

20.7.1 FOUR DMA PORTS

Camera Interface has Four DMA port. MSDMA input for preview, MSDMA input for codec, P-port(Preview out port) and C-port(Codec out port) are separated from each other on AHB bus. At the view of system bus, four ports are independent. The MSDMA reads the YCbCr 4:2:0 or YCbCr 4:2:2 or RGB image. The P-port and C-port store YCbCr 4:2:0 or YCbCr 4:2:2 or RGB image data in the memory. The P-port and C-port can be selected memory input data through MSDMA or camera input data through Camera. These four master ports support the variable applications like DSC (Digital Steel Camera), MPEG-4 video conference, video recording, etc. For example, P-port image can be used as preview image, and C-port image can be used as JPEG image in DSC application. The register setting can separately disabled for four DMA port.

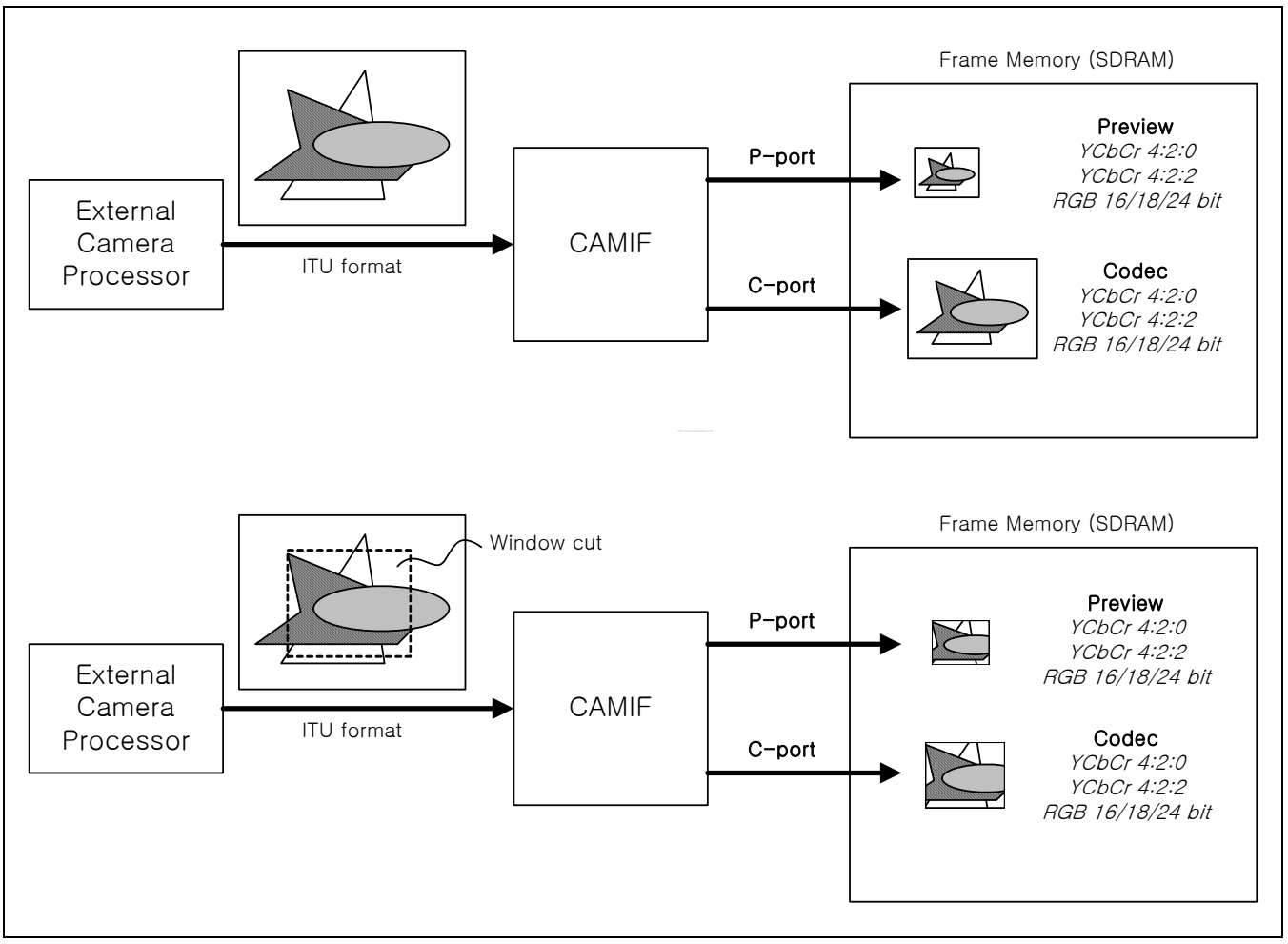


Figure 20-6. DMA ports through Camera processor data

20.7.2 CLOCK DOMAIN

Camera Interface has two clock domains. The one is the system bus clock, which is HCLK. The other is the pixel clock, which is PCLK. *The system clock must be faster than pixel clock*. As highlighted in Figure 20- 7, CAMCLK must be divided from the fixed frequency like APLL or MPLL clock. If external clock oscillator is used, CAMCLK must be floated. Internal scaler clock is system clock. It is not mandatory for two clock domains to be synchronized with each other. Other signals as PCLK must be similarly connected to shimitt-triggered level shifter.

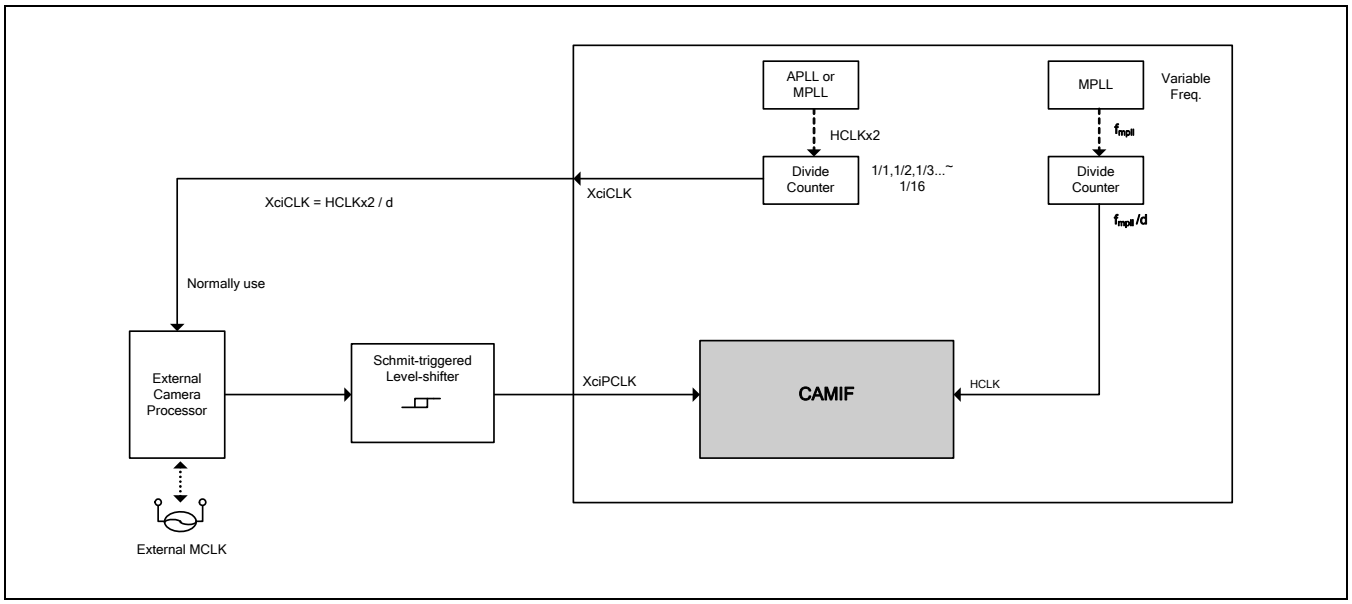


Figure 20-7. CAMERA INTERFACE clock generation

20.7.3 FRAME MEMORY HIERARCHY

Frame memories consist of four ping-pong memories for each P-ports and C-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 CAMERA INTERFACE must be higher than any other masters except LCD controller. It is strongly recommended that CAMERA INTERFACE priorities must be the fixed priorities, instead of rotation priorities. In multi-AHB bus case, the priority of system bus including CAMERA INTERFACE must be higher than others. If AHB-bus is traffic enough that DMA operation is not ending during one horizontal period plus blank, it might be entered into mal-function. Therefore, the priority of CAMERA INTERFACE must be separated to other round robin or circular arbitration priorities. It is also recommended that AHB bus which includes CAMERA INTERFACE must have higher priority than any other multi-AHB buses in memory matrix system. CAMERA INTERFACE must not be the default master of AMBA AHB system.

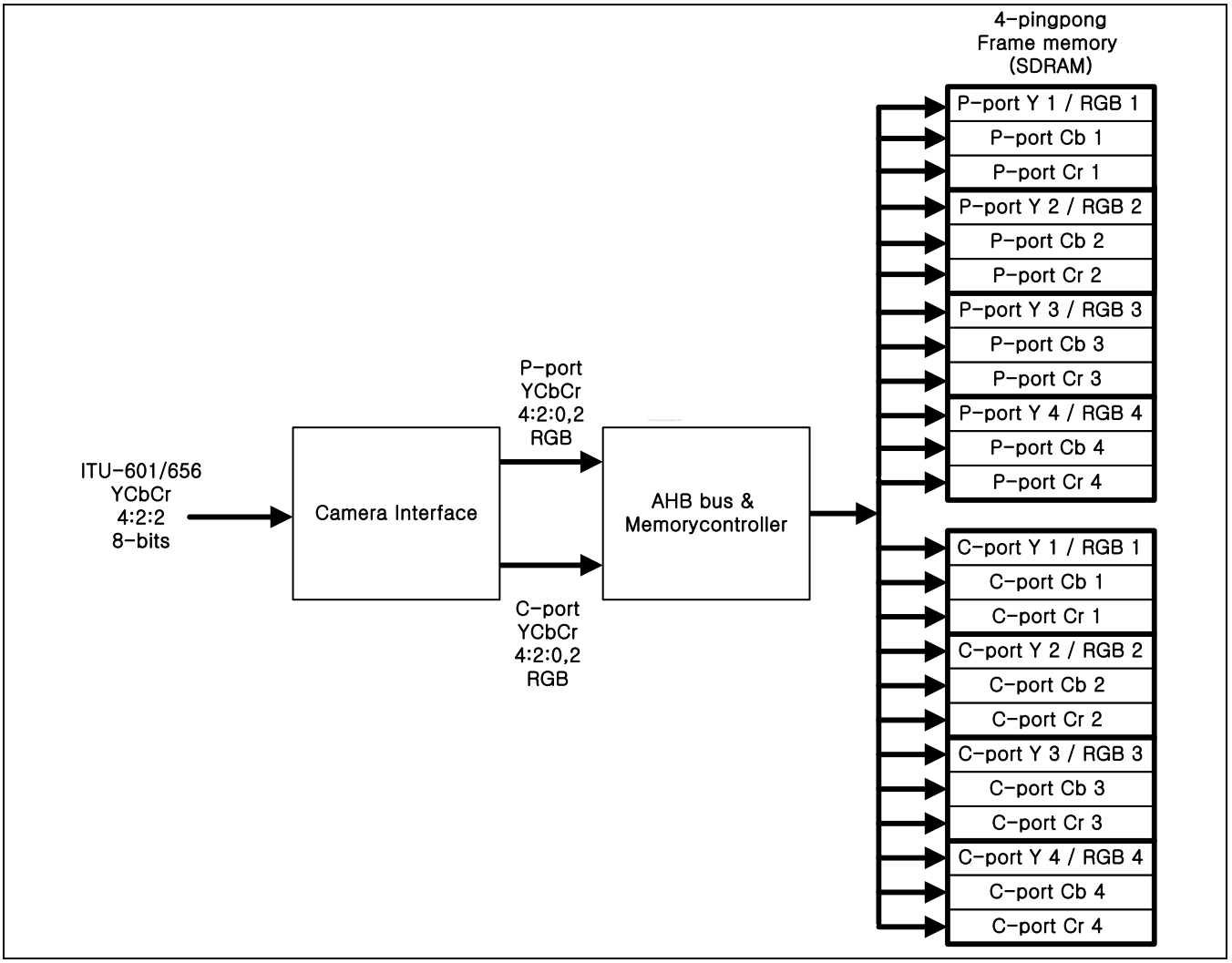


Figure 20-8. Ping-pong memory hierarchy

20.7.4 MEMORY STORING METHOD

The little-endian method is the storing method to the frame memory. The first entering pixels stored into LSB sides, and the last entering pixels stored into MSB sides. The carried data by AHB bus is 32-bit word. Therefore, CAMERA INTERFACE makes the each Y-Cb-Cr words by little endian style. One pixel (Color 1 pixel) is one word for RGB 24-bit/18-bit format. Otherwise, two pixels are one word for RGB 16-bit format and YCbCr 4:2:2 interleave format. Refer to Figure 20-9 for Memory storing style.

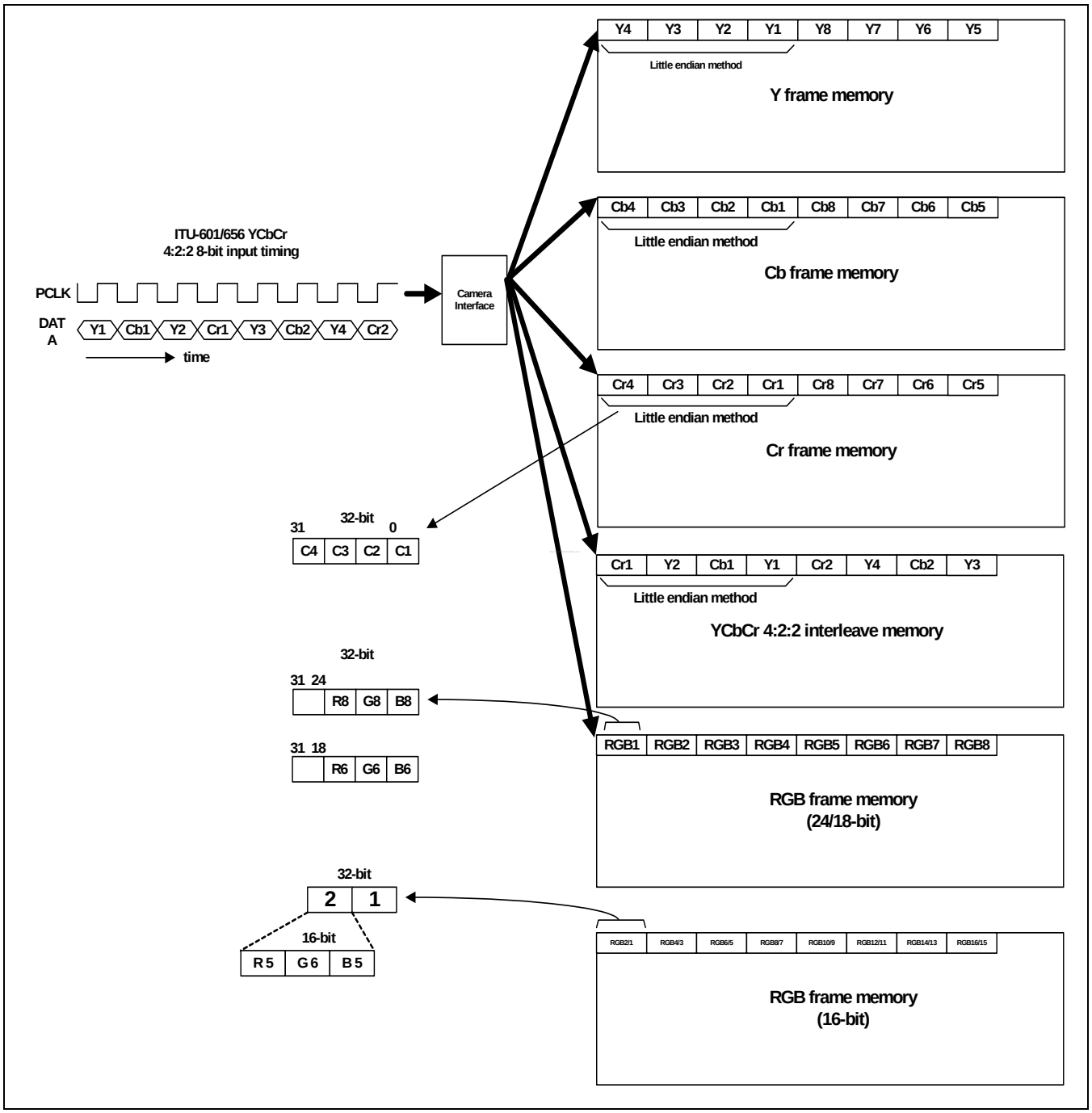
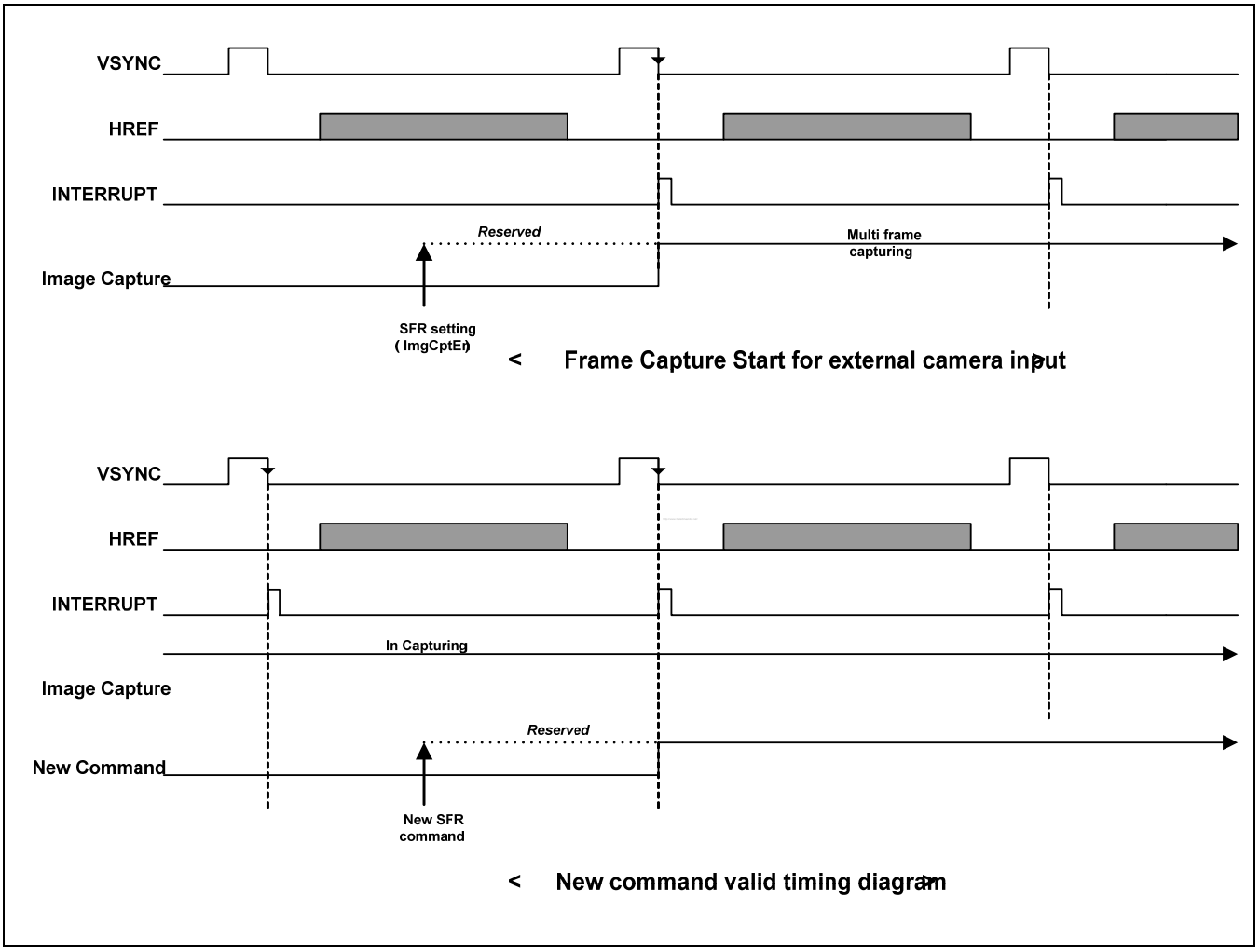


Figure 20-9. Memory storing style

20.7.5 TIMING DIAGRAM FOR REGISTER SETTING

The first register setting for frame capture command can occur in any part of frame period. It is recommend to do first setting at the VSYNC “L” state. VSYNC information can be read from status SFR. Refer to the below Figure 20-10. All command includes ImgCptEn and is valid at VSYNC falling edge. Make sure that except first SFR setting, all command must be programmed in ISR (Interrupt Service Routine). It is not allowed for target size information to be changed during capture operation. However, image mirror or rotation, windowing, and Zoom In settings can change in capturing operation. If some path select MSDMA input mode, all command must be programmed after MSDMA and P-port or C-Port DMA operation end.



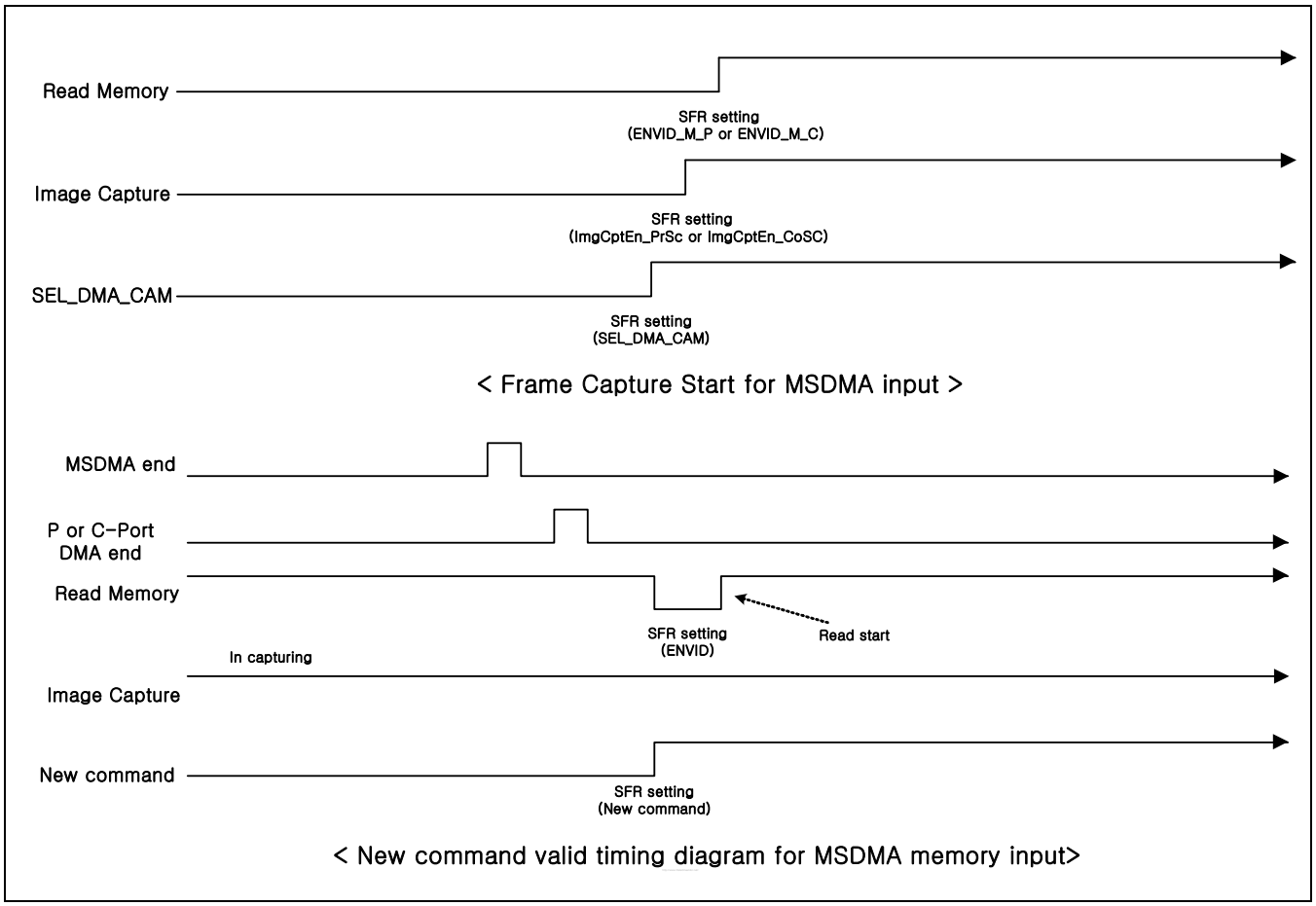


Figure 20-10. Timing diagram for register setting

20.7.6 TIMING DIAGRAM FOR LAST IRQ (CAMERA CAPTURE MODE)

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 auto-cleared and, as mentioned, SFR setting in ISR is for next frame command. Therefore, for adequate last IRQ, you must follow next sequence between LastIRQEn and ImgCptEn/ImgCptEn_CoSc/ImgCptEnPrSC. It is recommended that ImgCptEn/ImgCptEn_CoSc/ImgCptEnPrSC 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.

- Camera input capture path

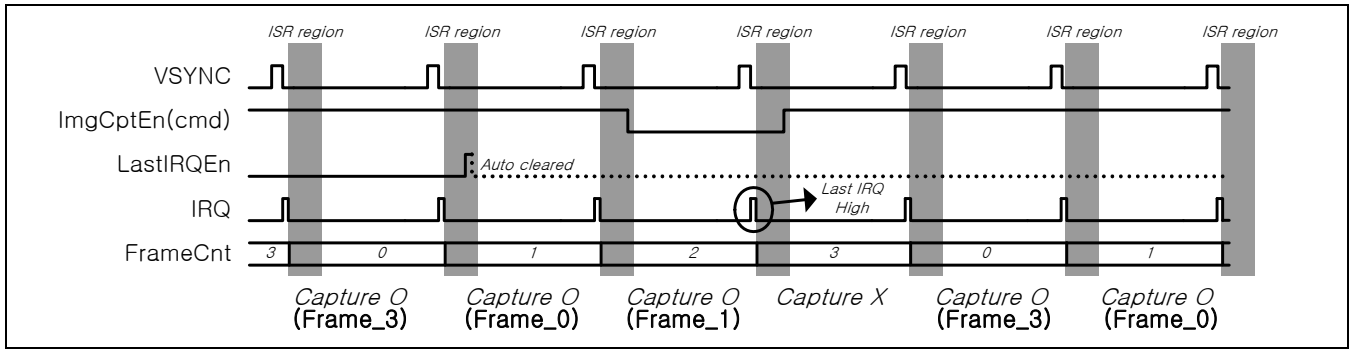


Figure 20-11. Timing diagram for last IRQ (LastIRQEn is enabled)

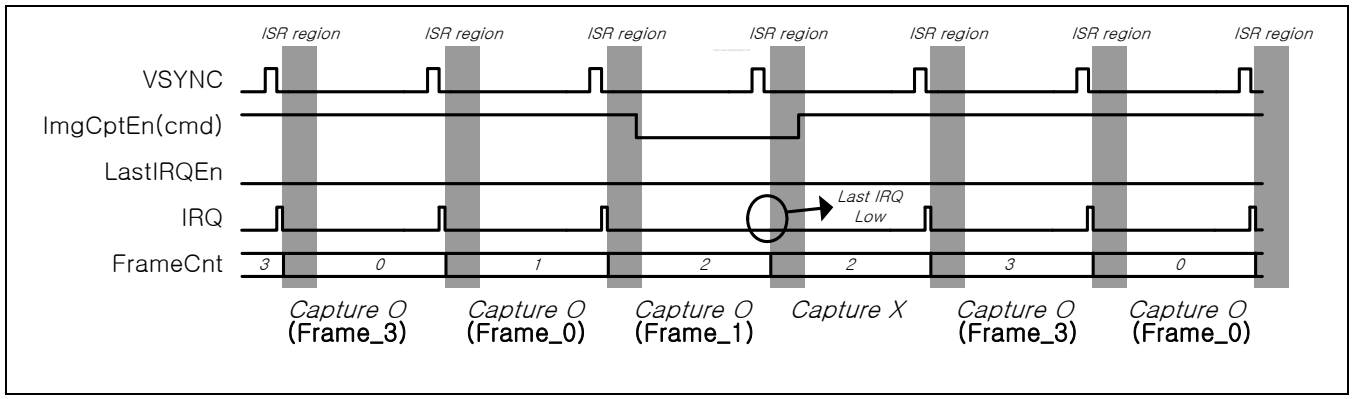


Figure 20-12. Timing diagram for last IRQ (LastIRQEn is disabled)

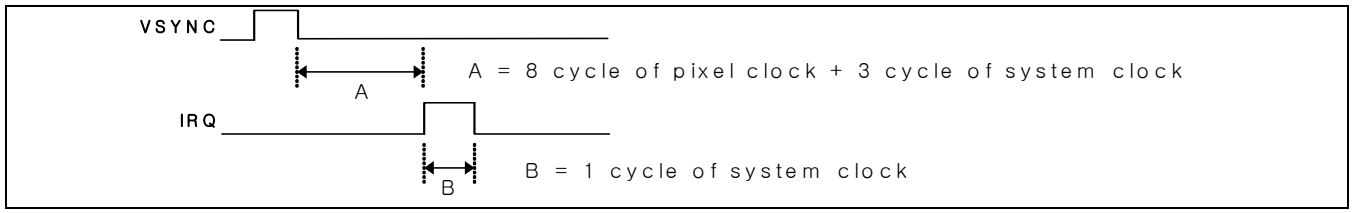


Figure 20-13. VSYNC & IRQ signals timing requirement

20.7.7 TIMING DIAGRAM FOR IRQ (MEMORY DATA SCALING MODE)

MSDMA input can be selected by SFR setting. In this case, IRQ is generated after P-port or C-port DMA operation is completed for per frame. This mode is aware of starting point by user's SFR setting (ENVID '0' → '1'). Therefore, this mode does not required IRQ of starting point and LastIRQ. FrameCnt is increased by 1 at ENVID_M_P(=ENVID_M_C) low to rising ('0' → '1') and ImgCptEn_PrSC(=ImgCptEn_CoSC) '1'

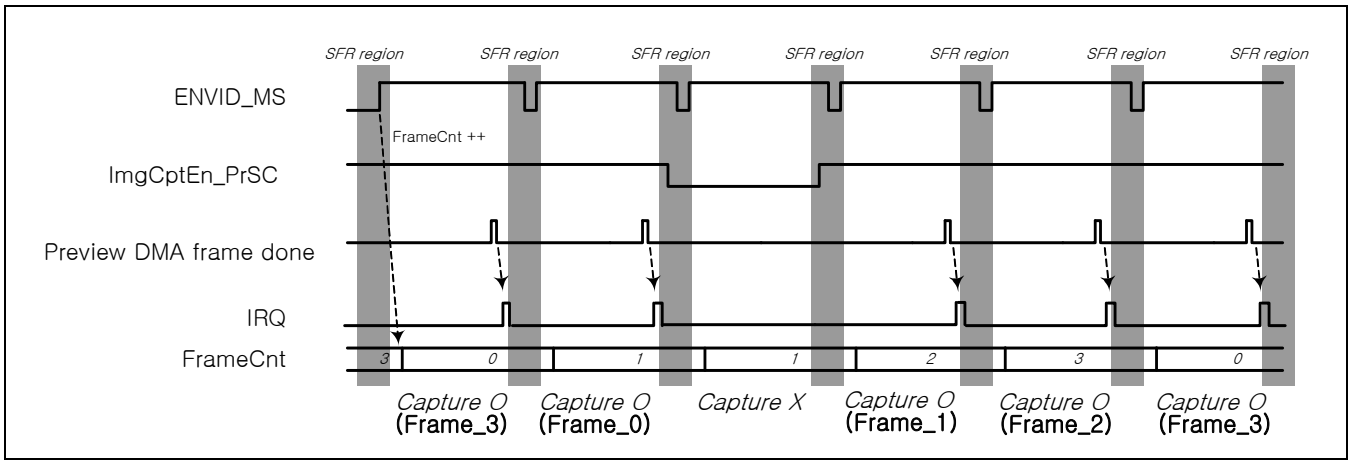


Figure 20-14. Timing diagram for IRQ (MSDMA path)

20.7.8 MSDMA FEATURE

MSDMA supports memory data scaling. Especially two different image data is required for PIP (Picture-in-Picture) operation. First image is saved memory by some codec (H.264, Camera, MPEG4 etc.) Second image is saved memory through MSDMA path. The MSDMA path has YCbCr/RGB output format through scaler/dma path. Two images are displayed and controlled by LCD controller. If MSDMA is (reading the memory data) required to use in preview path or codec path. SFR SEL_DMA_CAM_P(=SEL_DMA_CAM_C) 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 MSDMA input are:
YCbCr 4:2:0 (non-interleave)
YCbCr 4:2:2 (non-interleave)
YCbCr 4:2:2 (Interleave)
RGB

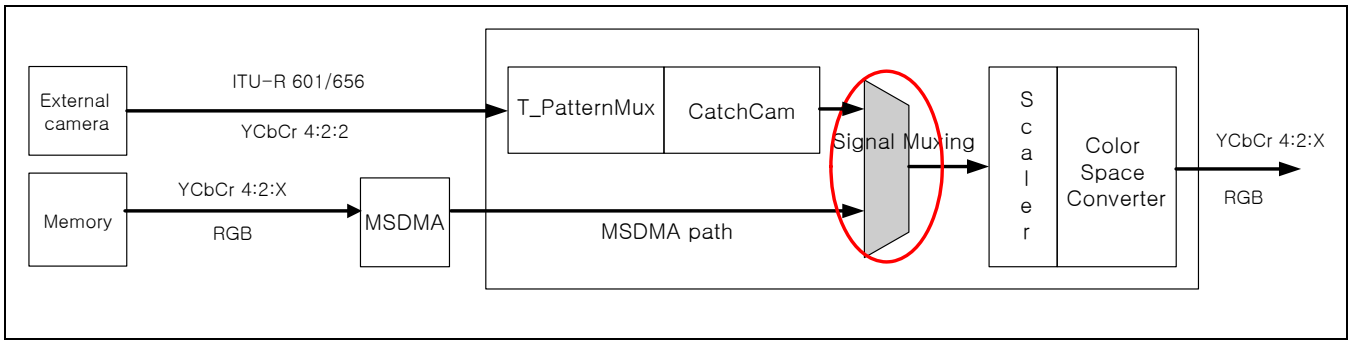


Figure 20-15. MSDMA or External Camera interface

20.7.8 CAMERA INTERLACE INPUT SUPPORT

6410 supports ITU-R BT 601 YCbCr 8/16 bit mode and ITU-R BT 656 YCbCr 8-bit mode in order to getting data from external camera. 6410 supports not only progressive input but also progressive input in both modes

20.7.8.1 PROGRESSIVE INPUT

In progressive mode, all the input data is stored in four buffers(pingpong memory which is designated by SFR) sequentially by the unit of frame. Refer Figure 20-8.

20.7.8.1 INTERACED INPUT

In interlace mode, the input data is stored in four buffers(pingpong memory which is designated by SFR). In this mode, even field frame data and odd field frame data is stored in turn. Therefore even field frame data is stored in 1st and 3rd pingpong memory while odd field frame data is stored 2nd and 4th pingpong memory. In case of image capture, start frame is always even field frame.

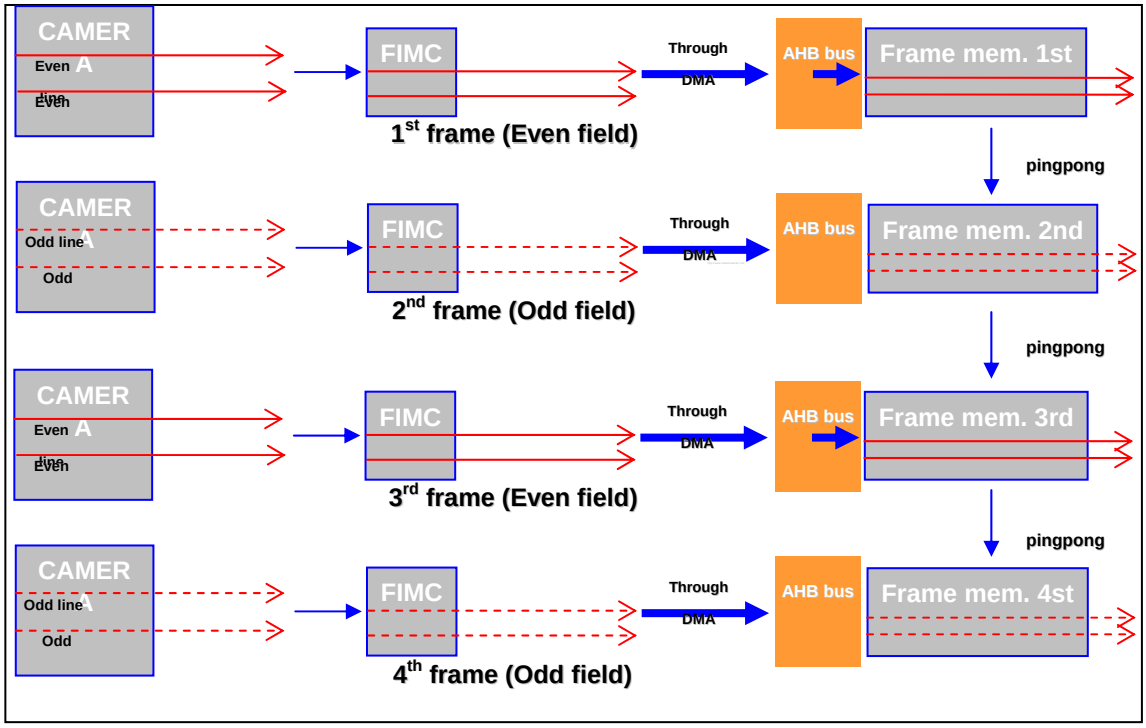


Figure 20-16. Frame Buffer Control

20.7.8.1 601 INTERFACE

In order to determine if the frame is even field or odd field, FIELD signal is used. If the FIELD signal is high, input data is for odd frame. Otherwise input data is for even frame. It is possible to invert the meaning of FIELD value. If you set InvPolFIELD(CIGCCTRL : 0x7800_0008) to '1', the high state of FIELD signal means that the current frame is even frame. Note that you must set the FIELDMODE(CIGCCTRL : 0x7800_0008) to '1' when using 601 interface mode.

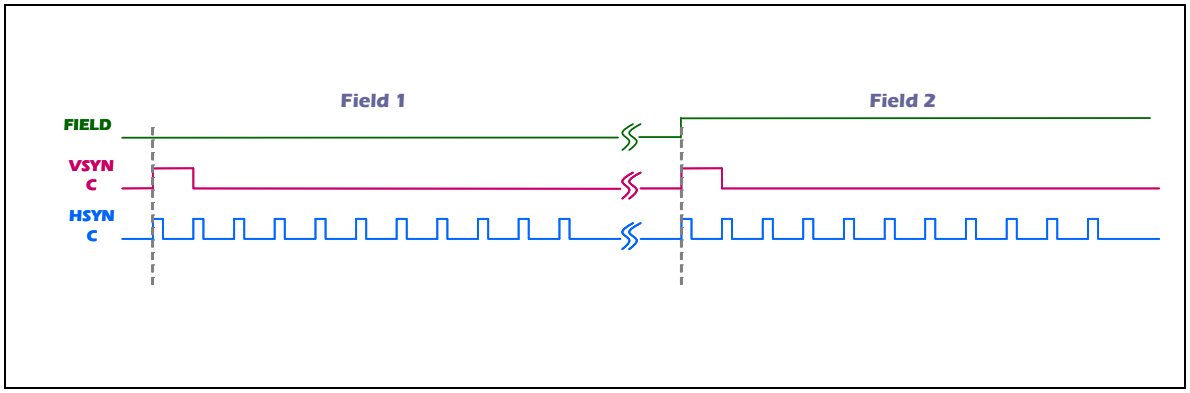


Figure 20-17. Frame Buffer Control

20.7.8.1 656 INTERFACE

In 656 interface, the field information of current frame is in 4th words in EAV and SAV.

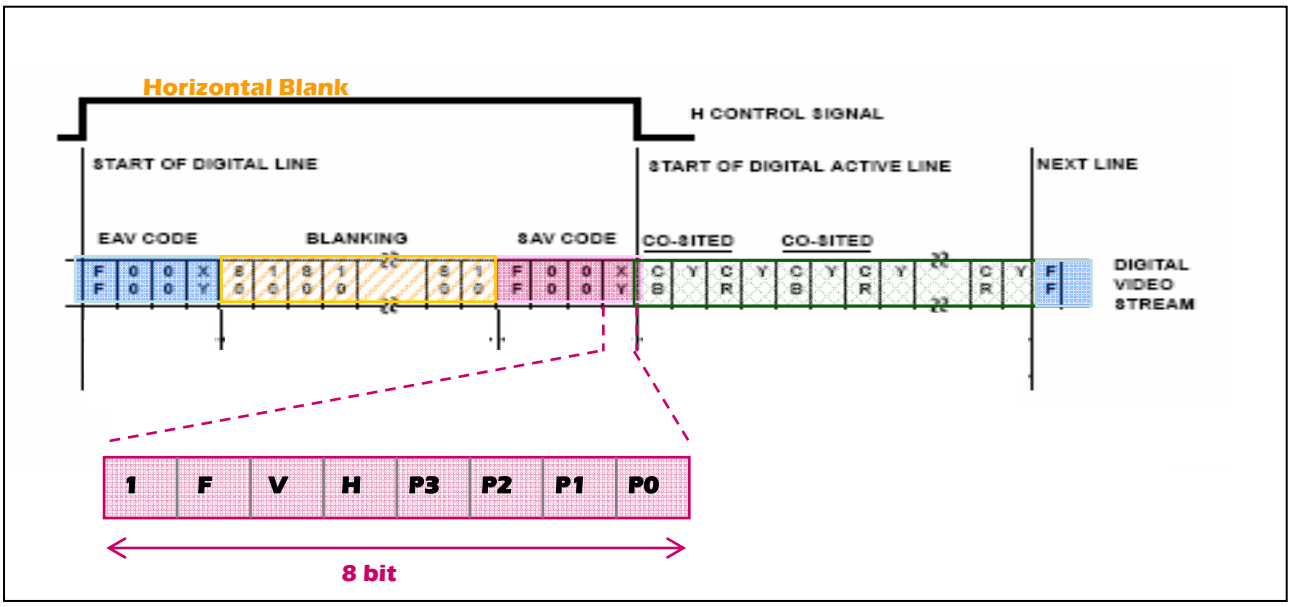


Figure 20-18. 656 Field Capture Control

Data bit Number	First WORD (FF)	SECOND word(00)	Third word(00)	fourth word(XY)
9(MSB)	1	0	0	1
8	1	0	0	F
7	1	0	0	V
6	1	0	0	H
5	1	0	0	P3
4	1	0	0	P2
3	1	0	0	P1
2	1	0	0	P0
1	1	0	0	0
0	1	0	0	0

F = 0 during FIELD 1
1 during FILED 2
V = 0 elsewhere
1 during field blanking
H = 0 in SAV
1 in EAV
P3, P2, P1, P0 : Protection bit

Figure 20-19. Video timing reference code

20.8 SOFTWARE INTERFACE (CAMERA INTERFACE IN S3C6410X SFR)

20.8.1 CAMERA INTERFACE SPECIAL FUNCTION REGISTERS

♦ 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 MSDMA path. (O : relationship, X : no relationship)

20.8.2 CAMERA SOURCE FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CISRCFMT	0x78000000	RW	Camera Input Source Format	0

CISRCFMT	Bit	Description	Initial State	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 : +128 0 : +0 (normally used)	0	X	X
reserved	[29]		0	X	X
SrcHsize_CAM	[28:16]	Camera source horizontal pixel number (must be 8's multiple. minimum 8. It must be 4's multiple of PreHorRatio if WinOfsEn is 0)	0	X	O
Order422_CAM	[15:14]	Camera Input YCbCr order inform for 8-bit mode 8-bit mode 00 : YCbYCr 01 : YCrYCb 10 : CbYCrY 11 : CrYCbY	0	X	X
Reserved	[13]		0	X	X
SrcVsize_CAM	[12:0]	Camera source vertical pixel number (minimum 8. It must be multiple of PreVerRatio when scale down if WinOfsEn is 0)	0	X	O

20.8.3 WINDOW OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIWDOFST	0x78000004	RW	Window offset register	0

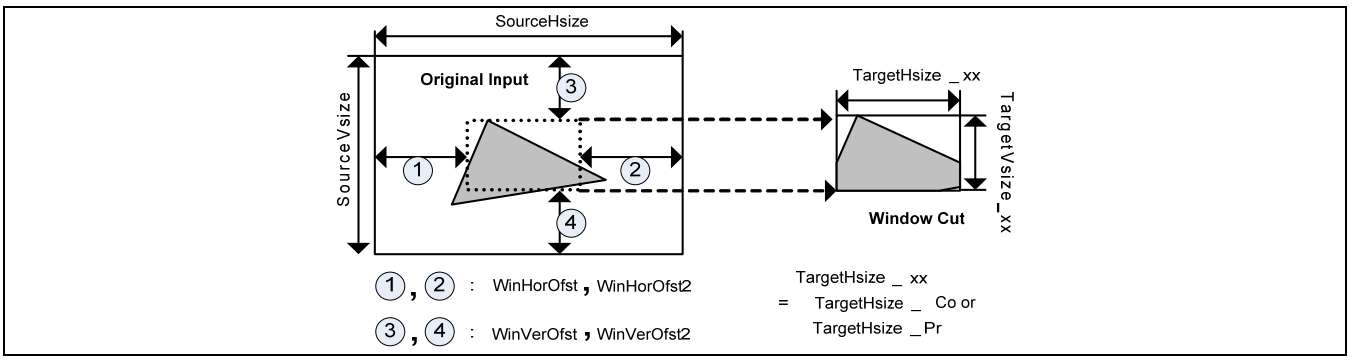


Figure 20-20. Window offset scheme
(WinHorOfst2 & WinVerOfst2 are assigned in the CIWDOFST2 register)

CIWDOFST	Bit	Description	Initial State	M	L
WinOfsEn	[31]	1 : window offset enable 0 : no offset	0	X	O
ClrOvCoFiY	[30]	1 : clear the overflow indication flag of input CODEC FIFO Y 0 : normal	0	X	X
reserved	[29]		0	X	X
ClrOvRLB_Pr	[28]	Clear the overflow indication flag of Line Buffer for Rotation in Preview path	0	X	X
ClrOvPrFiY	[27]	1 : clear the overflow indication flag of input PREVIEW FIFO Y 0 : normal	0	X	X
WinHorOfst	[26:16]	Window horizontal offset by pixel unit. (It must be 2's multiple)	0	X	O
ClrOvCoFiCb	[15]	1 : clear the overflow indication flag of input CODEC FIFO Cb 0 : normal	0	X	X
ClrOvCoFiCr	[14]	1 : clear the overflow indication flag of input CODEC FIFO Cr 0 : normal	0	X	X
ClrOvPrFiCb	[13]	1 : clear the overflow indication flag of input PREVIEW FIFO Cb 0 : normal	0	X	X
ClrOvPrFiCr	[12]	1 : clear the overflow indication flag of input PREVIEW FIFO Cr 0 : normal	0	X	X
Reserved	[11]		0	X	X
WinVerOfst	[10:0]	Window vertical offset by pixel unit.	0	X	O

NOTE: Clear bits emust be set to zero after clearing the flags.
Crop Hsize (= SourceHsize – WinHorOfst - WinHorOfst2) must be 8's multiple and 4's multiple of PreHorRatio.
Crop Vsize (= SourceVsize – WinVerOfst - WinVerOfst2) must be multiple of PreVerRatio when scale down. Must be an even number and minimum 8 if codec output 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

20.8.4 GLOBAL CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIGCTRL	0x78000008	RW	Global control register	2000_0000

CIGCTRL	Bit	Description	Initial State	M	L
SwRst	[31]	Camera interface software reset. Before setting this bit, you must 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	[30]	External camera processor Reset or Power Down control	0	X	X
reserved	[29]		1	X	X
TestPattern	[28:27]	This register must be set at only ITU-T 601 8-bit mode. Not allowed with input 16-bit mode or ITU-T 656 mode. (max. 1280 X 1024) 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]		0	X	X
IRQ_Ovfen	[22]	1 : Overflow interrupt enable (Interrupt is generated during overflow occurrence) 0 : Overflow interrupt disable (normal)	0	X	X
Href_mask	[21]	1 : mask out Href during Vsync high 0 : no mask	0	X	X
IRQ_LEVEL	[20]	1 : Level interrupt 0 : Edge trigger interrupt (default) * This bit should be set to ‘1’ because of using level interrupt method in S3C6410x	0	X	X
IRQ_CLR_c	[19]	This bit is related only to Level interrupt. Codec path interrupt is cleared when IRQ_CLR_c is written to ‘1’. This bit Auto-clear.	0	X	X
IRQ_CLR_p	[18]	This bit is related only to Level interrupt. Preview path interrupt is cleared when IRQ_CLR_p is written to ‘1’. This bit Auto-clear.	0	X	X
Reserved	[17:3]				

FIELDMODE	[2]	ITU601 Interlace field mode (Don't care this bit in ITU656 mode) 1 : Using the FIELD port mode 0 : Reserved	0	X	X
InvPolFIELD	[1]	1 : inverse the polarity of FIELD 0 : normal * In normal, when the FIELD signal is "LOW", the frame is even field	0	X	X
Cam_Interface	[0]	External Camera scan method 1: Interlace 0 : Progressive	0	X	X

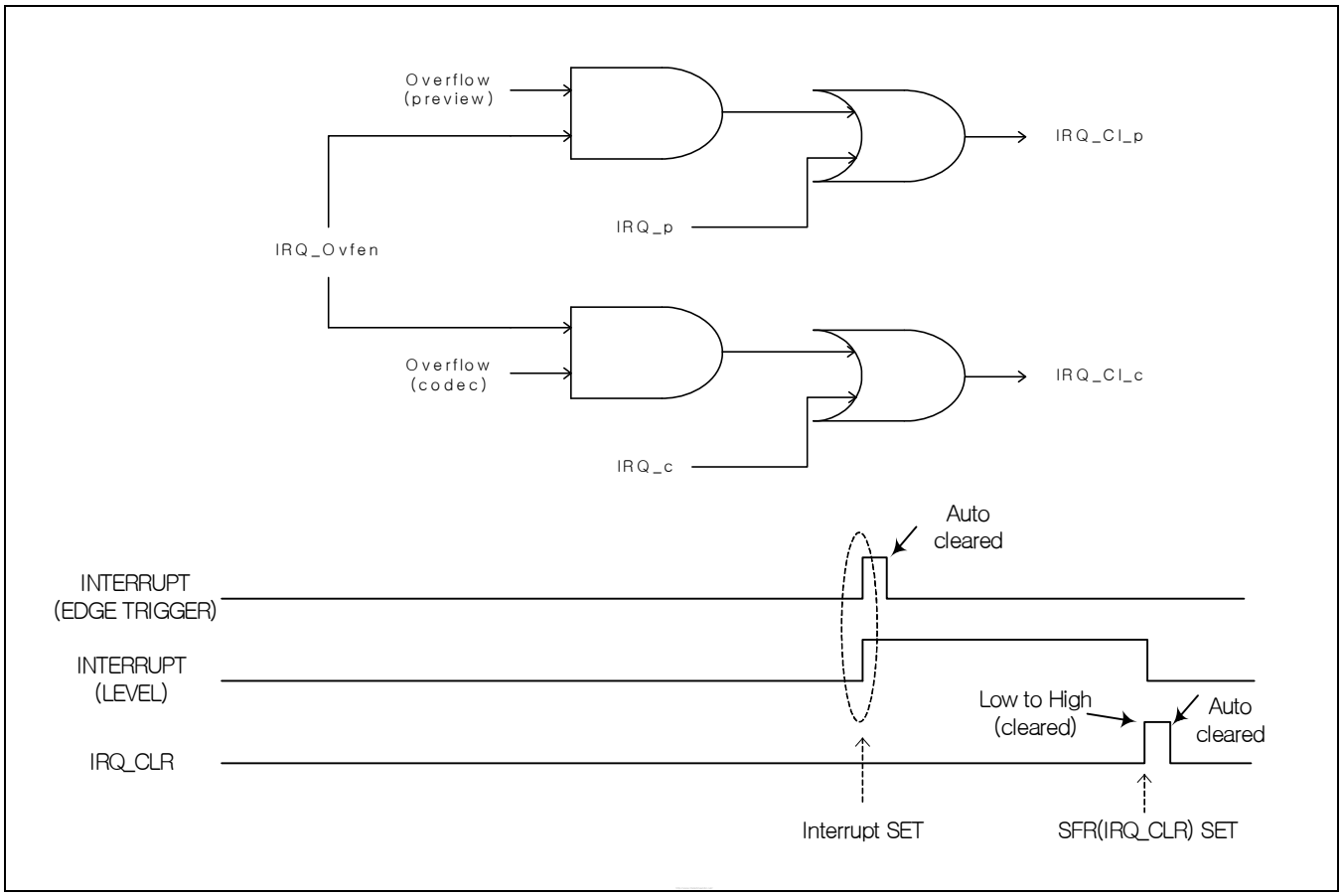


Figure 20-21. Interrupt generation scheme

20.8.5 WINDOW OFFSET REGISTER 2

Register	Address	R/W	Description	Reset Value
CIWDOFST2	0x78000014	RW	Window offset register 2	0

CIWDOFST2	Bit	Description	Initial State	M	L
Reserved	[31:27]		0	X	X
WinHorOfst2	[26:16]	Window horizontal offset2 by pixel unit. (It must be 2's multiple) **Caution :** SourceHsize-WinHorOfst- WinHorOfst2 must be 8's multiple and minimum 16.	0	X	O
Reserved	[15:11]		0	X	X
WinVerOfst2	[10:0]	Window vertical offset2 by pixel unit	0	X	O

20.8.6 CODEC OUTPUT Y1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA1	0x78000018	RW	1st frame start address for codec DMA	0

CICOYSA1	Bit	Description	Initial State	M	L
CICOYSA1	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 1st frame start address	0	O	X

20.8.7 CODEC OUTPUT Y2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA2	0x7800001C	RW	2nd frame start address for codec DMA	0

CICOYSA2	Bit	Description	Initial State	M	L
CICOYSA2	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 2nd frame start address	0	O	X

20.8.8 CODEC OUTPUT Y3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA3	0x78000020	RW	3rd frame start address for codec DMA	0

CICOYSA3	Bit	Description	Initial State	M	L
CICOYSA3	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 3rd frame start address	0	O	X

20.8.9 CODEC OUTPUT Y4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOYSA4	0x78000024	RW	4th frame start address for codec DMA	0

CICOYSA4	Bit	Description	Initial State	M	L
CICOYSA4	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 4th frame start address	0	O	X

20.8.10 CODEC OUTPUT CB1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA1	0x78000028	RW	Cb 1st frame start address for codec DMA	0

CICOCBSA1	Bit	Description	Initial State	M	L
CICOCBSA1	[31:0]	Cb 1st frame start address for codec DMA	0	O	X

20.8.11 CODEC OUTPUT CB2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA2	0x7800002C	RW	Cb 2nd frame start address for codec DMA	0

CICOCBSA2	Bit	Description	Initial State	M	L
CICOCBSA2	[31:0]	Cb 2nd frame start address for codec DMA	0	O	X

20.8.12 CODEC OUTPUT CB3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA3	0x78000030	RW	Cb 3rd frame start address for codec DMA	0

CICOCBSA3	Bit	Description	Initial State	M	L
CICOCBSA3	[31:0]	Cb 3rd frame start address for codec DMA	0	O	X

20.8.13 CODEC OUTPUT CB4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCBSA4	0x78000034	RW	Cb 4th frame start address for codec DMA	0

CICOCBSA4	Bit	Description	Initial State	M	L
CICOCBSA4	[31:0]	Cb 4thframe start address for codec DMA	0	O	X

20.8.14 CODEC OUTPUT CR1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA1	0x78000038	RW	Cr 1st frame start address for codec DMA	0

CICOCRSA1	Bit	Description	Initial State	M	L
CICOCRSA1	[31:0]	Cr 1st frame start address for codec DMA	0	O	X

20.8.15 CODEC OUTPUT CR2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA2	0x7800003C	RW	Cr 2nd frame start address for codec DMA	0

CICOCRSA2	Bit	Description	Initial State	M	L
CICOCRSA2	[31:0]	Cr 2nd frame start address for codec DMA	0	O	X

20.8.16 CODEC OUTPUT CR3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA3	0x78000040	RW	Cr 3rd frame start address for codec DMA	0

CICOCRSA3	Bit	Description	Initial State	M	L
CICOCRSA3	[31:0]	Cr 3rd frame start address for codec DMA	0	O	X

20.8.17 CODEC OUTPUT CR4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CICOCRSA4	0x78000044	RW	Cr 4th frame start address for codec DMA	0

CICOCRSA4	Bit	Description	Initial State	M	L
CICOCRSA4	[31:0]	Cr 4thframe start address for codec DMA	0	O	X

20.8.18 CODEC TARGET FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CICOTRGFMT	0x78000048	RW	Target image format of codec DMA	0

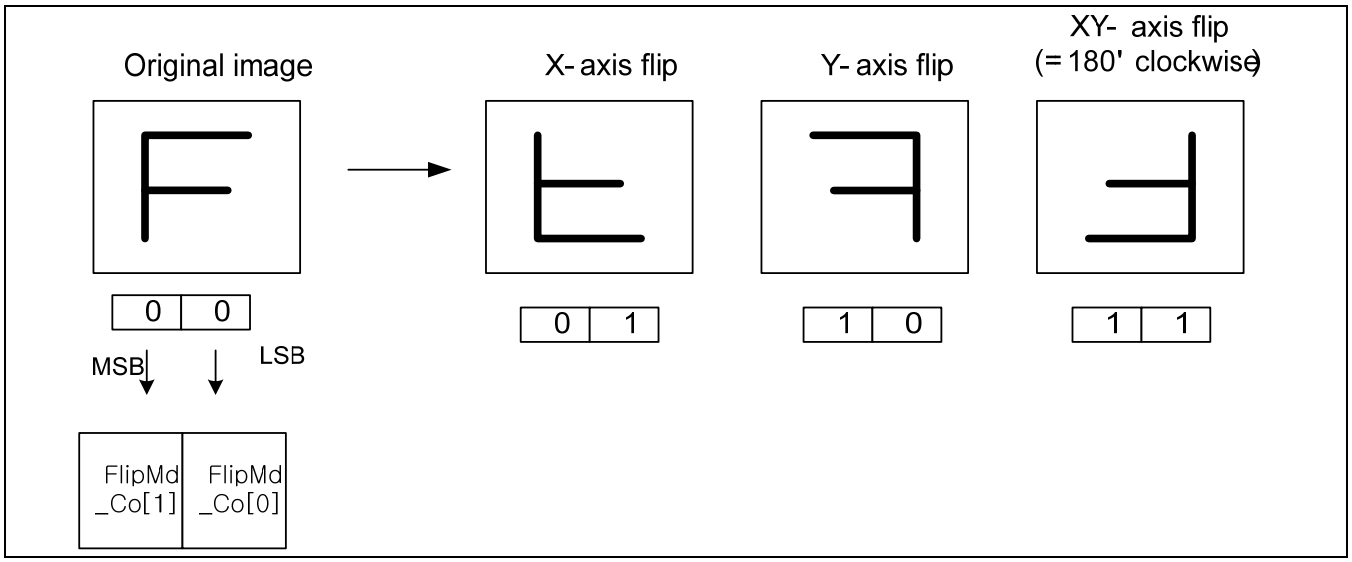


Figure 20-22. Codec image mirror

CICOTRGFMT	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
OutFormat_Co	[30:29]	00 : YCbCr 4:2:0 codec output image format. (Non-interleave) 01 : YCbCr 4:2:2 codec output image format. (Non-interleave) 10 : YCbCr 4:2:2 codec output image format. (Interleave) 11 : RGB codec output image format. (cf. RGB format register → OutRGB_FMT_Pr)	0	O	O
TargetHsize_Co	[28:16]	Horizontal pixel number of target image for codec DMA (16's multiple. minimum 16)	0	O	O
FlipMd_Co	[15:14]	Image mirror and rotation for codec DMA 00 : Normal 01 : X-axis mirror 10 : Y-axis mirror 11 : 180° rotation (XY-axis mirror)	0	O	O
reserved	[13]				
TargetVsize_Co	[12:0]	Vertical pixel number of target image for codec DMA. Minimum number is 4.	0	O	O

TargetHsize_Co and TargetVsize_Co must not be larger than Camera SourceHsize and Camera SourceVsize. DMA input source size can be ignored.

NOTE: If TargetVsize_Co value is set to an odd number (N) when output format is YCbCr 4:2:0 (Recommend to use an even number). The odd number (N) of Y lines and the (N-1)/2 of Cb, Cr lines is generated. X-flip or XY-flip are not allowed

20.8.19 CODEC DMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CICOCTRL	0x7800004C	RW	Codec DMA control related	0

CICOCTRL	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
Yburst1_Co	[23:19]	Main burst length for codec Y frames	0	O	O
Yburst2_Co	[18:14]	Remained burst length for codec Y frames	0	O	O
Cburst1_Co	[13:9]	Main burst length for codec Cb/Cr frames	0	O	O
Cburst2_Co	[8:4]	Remained burst length for codec Cb/Cr frames	0	O	O
Reserved	[3]		0	X	X
LastIRQEn_Co	[2]	1 : enable last IRQ at the end of frame capture (It is recommended to check the done signal of capturing image for JPEG. One pulse) 0 : normal	0	X	X
Order422_Co	[1:0]	Interleaved YCbCr 4:2:2 output order memory storing style 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	O	O

* Interleaved burst length (Interleaved YCbCr 4:2:2)

Y burst length	2 , 4 , 8
C burst length (C burst length = Y burst length / 2)	1 , 2 , 4
Wanted burst length (= Y + 2C)	4 , 8 , 16

NOTE: When Codec output format is YCbCr 4:2:2 interleave ,ScalerBypass_Co = 0 and ScaleUp_V_Co = 1 , Wanted main burst length = 16 and Wanted remained burst length ≠16 is not allowed.

♦ Non-Interleaved burst length (Y burst length – YcbCr 4:2:0, YCbCr 4:2:2)

Y	Main burst length = 4, 8, 16 Remained burst length = 4, 8, 16
C	Main burst length = 2, 4, 8, 16 Remained burst length = 2, 4, 8, 16

♦ Non-Interleaved burst length (RGB burst length)

Y	Main burst length = 4, 8, 16 Remained burst length = 4, 8, 16
C	No meaning

When YCbCr 4:2:2 interleave , burst size calculations are done to determine the wanted burst length. After finding the wanted burst length.

The SFR fields are programmed as shown below,

Y : wanted Main burst length = 2 * Yburst1_Co, and wanted Remained burst length = 2 * Yburst2_Co.

Cb/Cr : wanted Main burst length = Yburst1_Co / 2 = Cburst1_Co, and wanted Remained burst length = Yburst2_Co / 2 = Cburst2_Co

Example 1: Target image size: QCIF (horizontal Y width = 176 pixels. 1 pixel = 1 Byte)
If output format non-interleave YCbCr 4:2:2, 0 (1 word = 4 pixels)
Yword = 176 / 4 = 44 words, 44 % 8 = 4 → Y main burst = 8, Y remained burst = 4
Cword = 176 / 4 / 2 = 22 words, 22 % 4 = 2 → C main burst = 4, C remained burst = 2
If output format YCbCr 4:2:2 interleave (1 word = 2 pixel)
176 x (1 word / 2 pixles) = 88 words, 88 % 16 = 8 → Wanted main burst = 16, Wanted remained burst = 8
Wanted main burst = 16 = 2 * Yburst1 = 4 * Cburst1, Wanted remained burst = 8 = 2 * Yburst2 = 4 * Cburst2

Example 2: Target image size: VGA (horizontal Y width = 640 pixels. 1 pixel = 1 Byte)
If output format non-interleave YCbCr 4:2:2,0 (1 word = 4 pixel)
Yword = 640 / 4 = 160 word, 160 % 8 = 0 → Y main burst = 8, Y remained burst = 8
Cword = 640 / 4 / 2 = 80 word, 80 % 8 = 0 → C main burst = 8, C remained burst = 8
If output format Interleave 4:2:2 or RGB565 mode (1 word = 2 pixel)
640 x (1 word / 2 pixel) = 320 words, 320 % 16 = 0 → Wanted main burst = 16, Wanted remained burst = 16
If output format RGB888/666 mode (1 word = 1 pixel)
640 x (1 word / 1 pixels) = 640 words, 640 % 16 = 0 → Wanted main burst = 16, Wanted remained burst = 16

Example 3: Target image size : QCIF (horizontal C width = 88 pixels. 1 pixel = 1 Byte)
If output format non-interleave YCbCr 4:2:2,0 (1 word = 4 pixel)
88 / 4 = 22 word. , 22 % 4 = 2 → main burst = 4, remained burst = 2 (HTRANS==INCR)

Caution! CAMERA INTERFACE in S3C6410X generates INCR (HBURST of AMBA) transfer type at abnormal burst length as 2, and at crossing 1024 address boundary by single transfer. System controller including CAMERA INTERFACE in S3C6410X must support to treat INCR burst as several single transfer accesses. Watch over your memory controller and system arbiter specification!

20.8.20 REGISTER SETTING GUIDE FOR CODEC SCALER AND PREVIEW SCALER

SRC_Width and DST_Width satisfy the word boundary constraints such that the number of horizontal pixel can be represented to kn where n = 1,2,3, … and k = 1 / 2 / 8 for 24bppRGB / 16bppRGB / YCbCr420 image, respectively. TargetHsize must not be larger than Camera SourceHsize. Similarly, TargetVsize must not be larger than Camera SourceVsize. DMA input source size can be ignored.

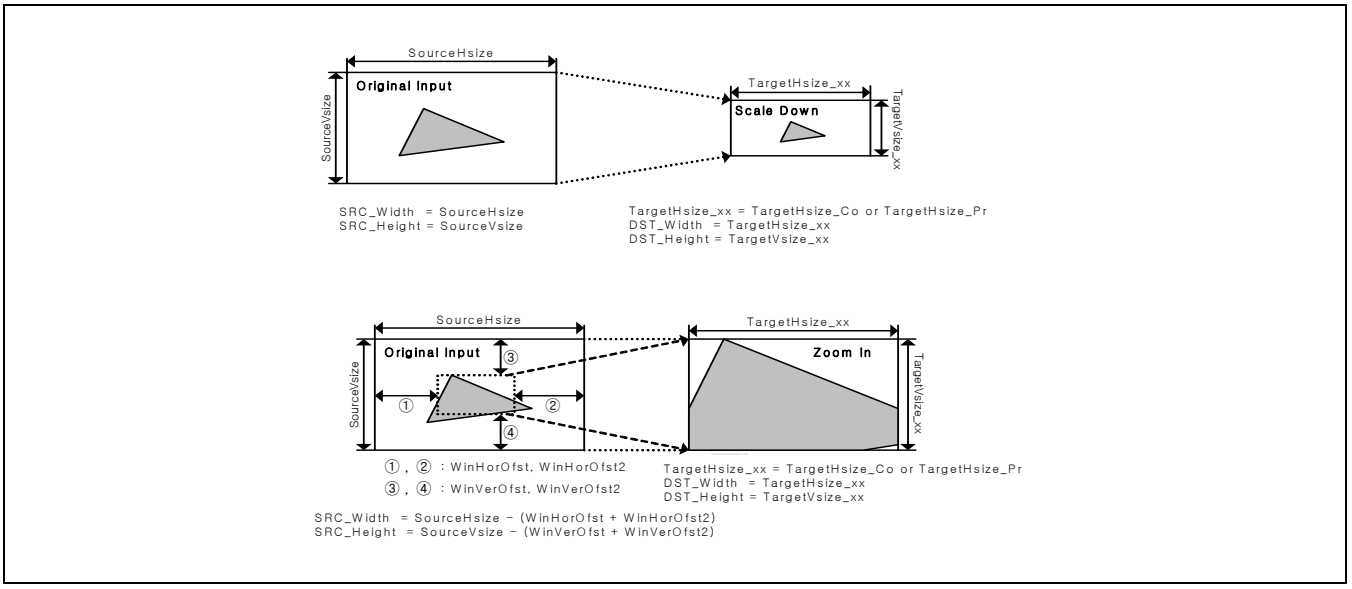


Figure 20-23. Scaling scheme

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_xx = 32;  H_Shift = 5; }
else if (SRC_Width >= 16 × DST_Width) { PreHorRatio_xx = 16;  H_Shift = 4; }
else if (SRC_Width >= 8 × DST_Width) { PreHorRatio_xx = 8;  H_Shift = 3; }
else if (SRC_Width >= 4 × DST_Width) { PreHorRatio_xx = 4;  H_Shift = 2; }
else if (SRC_Width >= 2 × DST_Width) { PreHorRatio_xx = 2;  H_Shift = 1; }
else { PreHorRatio_xx = 1;  H_Shift = 0; }
PreDstWidth_xx = SRC_Width / PreHorRatio_xx;
MainHorRatio_xx = ( 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_xx = 32;  V_Shift = 5; }
else if (SRC_Height >= 16 × DST_Height) { PreVerRatio_xx = 16;  V_Shift = 4; }
else if (SRC_Height >= 8 × DST_Height) { PreVerRatio_xx = 8;  V_Shift = 3; }
else if (SRC_Height >= 4 × DST_Height) { PreVerRatio_xx = 4;  V_Shift = 2; }
else if (SRC_Height >= 2 × DST_Height) { PreVerRatio_xx = 2;  V_Shift = 1; }
else { PreVerRatio_xx = 1;  V_Shift = 0; }
PreDstHeight_xx = SRC_Height / PreVerRatio_xx;
MainVerRatio_xx = ( SRC_Height << 8 ) / ( DST_Height << V_Shift);
SHfactor_xx = 10 – ( H_Shit + V_Shift);
```

Caution! In preview path, Pre-scaled H_width must be the less than 720. (The maximum size of preview path scaler's line buffer is 720.) → refer to the table 20-1.

Caution! In Zoom-In case, you must check the next equation.

((SourceHsize - (WinHorOfst + WinHorOfst2)) / PreHorRatio_Pr) <= 720 (preview scaler max. hsize)

20.8.21 CODEC PRE-SCALER CONTROL REGISTER 1

Register	Address	R/W	Description	Reset Value
CICOSCPRERATIO	0x78000050	RW	Codec pre-scaler ratio control	0

CICOSCPRERATIO	Bit	Description	Initial State	M	L
SHfactor_Co	[31:28]	Shift factor for codec pre-scaler	0	O	O
Reserved	[27:23]		0	X	X
PreHorRatio_Co	[22:16]	Horizontal ratio of codec pre-scaler	0	O	O
Reserved	[15:7]		0	X	X
PreVerRatio_Co	[6:0]	Vertical ratio of codec pre-scaler	0	O	O

20.8.22 CODEC PRE-SCALER CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CICOSCPREDST	0x78000054	RW	Codec pre-scaler destination format	0

CICOSCPREDST	Bit	Description	Initial State	M	L
Reserved	[31:28]		0	X	X
PreDstWidth_Co	[27:16]	Destination width for codec pre-scaler	0	X	O
Reserved	[15:12]		0	X	X
PreDstHeight_Co	[11:0]	Destination height for codec pre-scaler	0	X	O

20.8.23 CODEC MAIN-SCALER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CICOSCCTRL	0x78000058	RW	Codec main-scaler control	0x18000000

CICOSCCTRL	Bit	Description	Initial State	M	L
ScalerBypass_Co	[31]	Codec scaler bypass. In this case, ImgCptEn_CoSC must be 0, but ImgCptEn must be 1. Generally this mode uses large image size upper scaler maximum size. Therefore it is not recommended to capturing preview image. (Recommend, ImgCptEn_PrSC must be 0). This mode is intended to capture JPEG input image for DSC application). In this case, input pixel buffering depends on only input FIFOs, so system bus must not be busy in this mode. ScalerBypass has some restriction. it is not allowed size scaling, color space conversion, rotator and MSDMA memory input image. so, Input / output format is allowed YCbCr non-interleave 4:2:0,4:2:2 & interleave 4:2:2	0	O	O
ScaleUp_H_Co	[30]	Horizontal scale up/down flag for codec scaler (In 1:1 scale ratio, this bit must be “1”) 1: up, 0:down	0	O	O
ScaleUp_V_Co	[29]	Vertical scale up/down flag for codec scaler (In 1:1 scale ratio, this bit must be “1”) 1: up, 0:down	0	O	O
CSCR2Y_c	[28]	YCbCr Data Dynamic Range Selection for the Color Space Conversion RGB to YCbCr (Codec path) 1 : Wide => Y/Cb/Cr (0 ~ 255) : Wide default 0 : Narrow => Y (16 ~ 235), Cb/Cr (16 ~ 240) * Recommend CSC range setting CSCR2Y_c = CSCY2R_c (Wide=Wide or Narrow=Narrow)	1	O	O
CSCY2R_c	[27]	YCbCr Data Dynamic Range Selection for the Color Space Conversion YCbCr to RGB (Codec path) 1 : Wide => Y/Cb/Cr (0 ~ 255) : Wide default 0 : Narrow => Y (16 ~ 235), Cb/Cr (16 ~ 240)	1	O	O
LCDPathEn_Co	[26]	FIFO Mode Enable. 1 for FIFO mode and 0 for DMA mode	0	O	O
Interlace_Co	[25]	Output scan method selection register only when FIFO mode (LCDPathEn =1). 1 for Interlace scan and 0 for progressive scan. In DMA mode (LCDPathEn = 0), progressive scan is applied whatever this value has. This mode is not allowed when Input image data is from Camera processor.	0	O	O
MainHorRatio_Co	[24:16]	Horizontal scale ratio for codec main-scaler	0	O	O
CoScalerStart	[15]	Codec scaler start 1 : scaler start 0 : scaler stop	0	O	O
InRGB_FMT_Co	[14:13]	Input RGB format MSDMA for codec path dedicated 00 : RGB565 , 01 : RGB666 , 10 : RGB888 , 11 : Reserved	0	O	O

CICOSCCTRL	Bit	Description	Initial State	M	L
OutRGB_FMT_Co	[12:11]	Output RGB format for Codec write DMA 00 : RGB565 , 01 : RGB666 , 10 : RGB888 , 11 : Reserved	0	O	O
Ext_RGB_Co	[10]	Input RGB data extension enable bit for the conversion of RGB565/666 mode into RGB888 mode for codec path. 1 : Extension , 0 : normal i) Input R = 5bit in RGB565 mode *101*00 -> 10100*101* (Extension) 10100 -> 10100000 (normal) ii) Input R = 6-bit in RGB666 mode *10*1100 -> 10110*10* (Extension) 101100 -> 10110000 (normal)	0	O	O
One2One_Co	[9]	Non-interpolation data copy. (Caution : this register should be set at 1:1 scaler size for same in-out format and Image effect cannot support RGB format & One2One mode.) Ex) input YCbCr4:2:0 (VGA) -> output YCbCr4:2:0 (VGA)	0	O	O
MainVerRatio_Co	[8:0]	Vertical scale ratio for codec main-scaler	0	O	O

- **DMA Mode Operation (Nomal mode)**

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 must be stored in memory system aligned with word boundary. It means that neither byte nor half-word size DMA operations are supported. Therefore, the width of source and destination image must be selected to satisfy the word boundary condition.

- **FIFO Mode Operation**

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 FIMC-to-Memory and Memory-to-Display Controller. Output data format is determined by only Output format register: OutFormat_xx=RGB format (24bit RGB) or OutFormat_xx=YCbCr format (YCbCr444). The source image format and the destination image format restriction are described in the following table.

Input Image Format (Progressive)		Output image Format (Progressive / Interlace)
YCbCr	420 YCbCr Format	YCbCr 444 or RGB 24-bit
	422 YCbYCr (non-interleave)	
	422 YCbYCr (interleave)	
RGB	RGB 16/18/24-bit	

In FIFO mode (LCDPathEnable =1), either progressive or interlace scan mode can be selectable according to “interlace” control register. Register Files Lists. The “interlace” control bit is available only 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 when camera processor is selected input data.

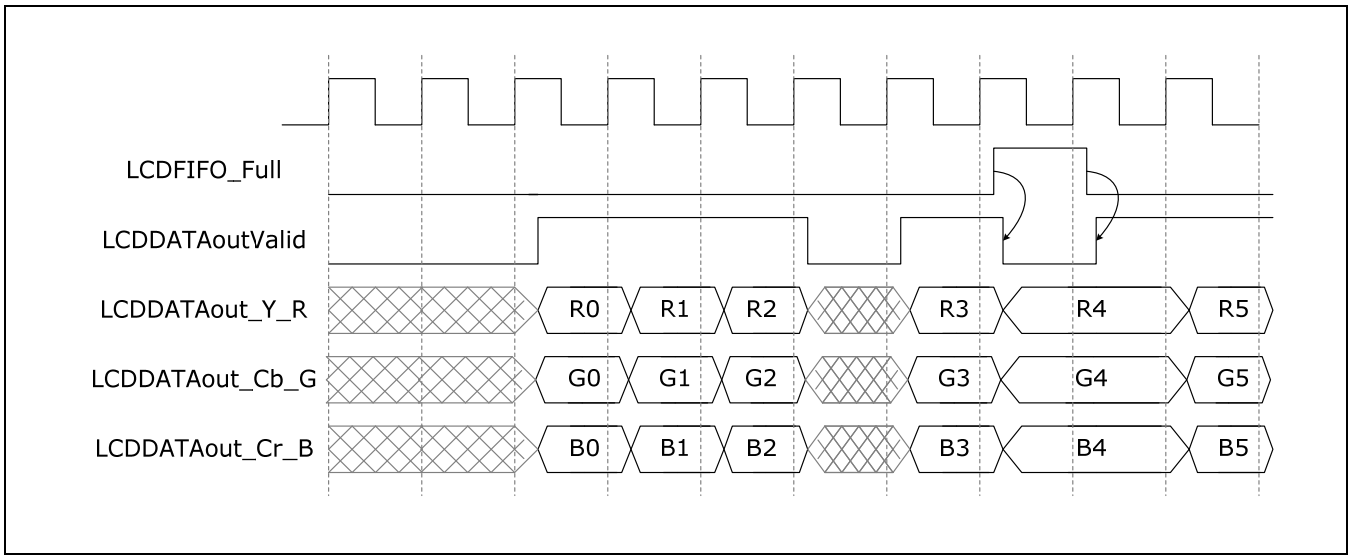


Figure 20-24. I/O Timing Diagram for LCD Path

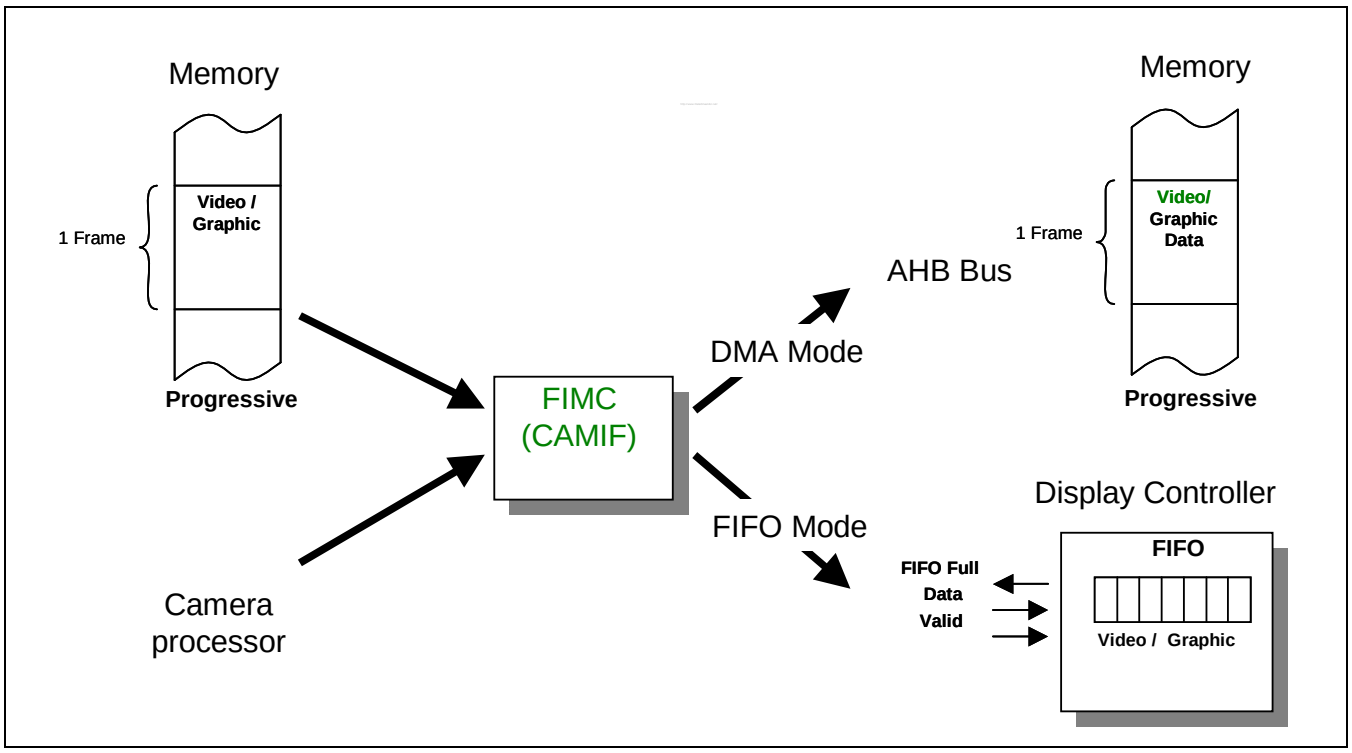


Figure 20-25. Two input & Two output modes in FIMC

20.8.24 CODEC DMA TARGET AREA REGISTER

Register	Address	R/W	Description	Reset Value
CICOTAREA	0x7800005C	RW	Codec dma target area	0

CICOTAREA	Bit	Description	Initial State	M	L
Reserved	[31:26]		0	X	X
CICOTAREA	[25:0]	Target area for codec DMA = Target H size x Target V size	0	O	O

20.8.25 CODEC STATUS REGISTER

Register	Address	R/W	Description	Reset Value
CICOSTATUS	0x78000064	R/W	Codec path status	0

CICOSTATUS	Bit	Description	Initial State	M	L
OvFiY_Co	[31]	Overflow state of codec FIFO Y	0	X	X
OvFiCb_Co	[30]	Overflow state of codec FIFO Cb	0	X	X
OvFiCr_Co	[29]	Overflow state of codec FIFO Cr	0	X	X
VSYNC	[28]	Camera VSYNC (This bit can be referred by CPU for first SFR setting after external camera muxing. It can be seen in the ITU-R BT 656 mode)	0	X	X
FrameCnt_Co	[27:26]	Frame count of codec DMA (This counter value means the next frame number)	0	X	X
WinOfstEn_Co	[25]	Window offset enable status	0	X	X
FlipMd_Co	[24:23]	Flip mode of codec DMA	0	X	X
ImgCptEn	[22]	Image capture enable of global camera interface	0	X	X
ImgCptEn_CoSC	[21]	Image capture enable of codec path	0	X	X
VSYNC_A	[20]	External camera A VSYNC (polarity inversion was not adopted.)	X	X	X
reserved	[19]		X	X	X
reserved	[18]		X	X	X
FrameEnd_Co	[17]	When codec frame operation finish, FrameEnd_Co is generated. and FrameEnd_Co is clear by user setting ‘0’	0	X	X
Reserved	[16:0]		0	X	X

20.8.26 PREVIEW OUTPUT Y1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRYSA1	0x7800006C	RW	1st frame start address for preview DMA	0

CIPRYSA1	Bit	Description	Initial State	M	L
CIPRYSA1	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 1st frame start address	0	O	X

20.8.27 PREVIEW OUTPUT Y2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRYSA2	0x78000070	RW	2nd frame start address for preview DMA	0

CIPRYSA2	Bit	Description	Initial State	M	L
CIPRYSA2	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 2nd frame start address	0	O	X

20.8.28 PREVIEW OUTPUT Y3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRYSA3	0x78000074	RW	3rd frame start address for preview DMA	0

CIPRYSA3	Bit	Description	Initial State	M	L
CIPRYSA3	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 3rd frame start address	0	O	X

20.8.29 PREVIEW OUTPUT Y4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRYSA4	0x78000078	RW	4th frame start address for preview DMA	0

CIPRCLRSA4	Bit	Description	Initial State	M	L
CIPRYSA4	[31:0]	Non-Interleave Y, Interleave YCbCr, RGB : 4th frame start address	0	O	X

20.8.30 PREVIEW OUTPUT CB1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCBSA1	0x7800007C	RW	1st frame start address for preview DMA	0

CIPRCBSA1	Bit	Description	Initial State	M	L
CIPRCBSA1	[31:0]	Cb 1st frame start address for preview DMA	0	O	X

20.8.31 PREVIEW OUTPUT CB2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCBSA2	0x78000080	RW	2nd frame start address for preview DMA	0

CIPRCBSA2	Bit	Description	Initial State	M	L
CIPRCBSA2	[31:0]	Cb 2nd frame start address for preview DMA	0	O	X

20.8.32 PREVIEW OUTPUT CB3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCBSA3	0x78000084	RW	3rd frame start address for preview DMA	0

CIPRCBSA3	Bit	Description	Initial State	M	L
CIPRCBSA3	[31:0]	Cb 3rd frame start address for preview DMA	0	O	X

20.8.33 PREVIEW OUTPUT CB4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCBSA4	0x78000088	RW	4th frame start address for preview DMA	0

CIPRCBSA4	Bit	Description	Initial State	M	L
CIPRCBSA4	[31:0]	Cb 4thframe start address for preview DMA	0	O	X

20.8.34 PREVIEW OUTPUT CR1 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCRSA1	0x7800008C	RW	1st frame start address for preview DMA	0

CIPRCRSA1	Bit	Description	Initial State	M	L
CIPRCRSA1	[31:0]	Cr 1st frame start address for preview DMA	0	O	X

20.8.35 PREVIEW OUTPUT CR2 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCRSA2	0x78000090	RW	2nd frame start address for preview DMA	0

CIPRCRSA2	Bit	Description	Initial State	M	L
CIPRCRSA2	[31:0]	Cr 2nd frame start address for preview DMA	0	O	X

20.8.36 PREVIEW OUTPUT CR3 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCRSA3	0x78000094	RW	3rd frame start address for preview DMA	0

CIPRCRSA3	Bit	Description	Initial State	M	L
CIPRCRSA3	[31:0]	Cr 3rd frame start address for preview DMA	0	O	X

20.8.37 PREVIEW OUTPUT CR4 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCRSA4	0x78000098	RW	4th frame start address for preview DMA	0

CIPRCRSA4	Bit	Description	Initial State	M	L
CIPRCRSA4	[31:0]	Cr 4thframe start address for preview DMA	0	O	X

20.8.38 PREVIEW TARGET FORMAT REGISTER

Register	Address	R/W	Description	Reset Value
CIPRTRGFMT	0x7800009C	RW	Target image format of preview DMA	0000_0000

CIPRTRGFMT	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
OutFormat_Pr	[30:29]	*00 : YCbCr 4:2:0 preview output image format. (Non-interleave)* *01 : YCbCr 4:2:2 preview output image format. (Non-interleave)* *10 : YCbCr 4:2:2 preview output image format. (Interleave)* *11 : RGB preview output image format. (cf. RGB format register → OutRGB_FMT_Co)*	0	O	O
TargetHsize_Pr	[28:16]	Horizontal pixel number of target image for preview DMA (16's multiple)	0	O	O
FlipMd_Pr	[15:14]	Image mirror and rotation for preview DMA 00 : normal 01 : x-axis mirror 10 : y-axis mirror 11 : 180° rotation	0	O	O
Rot90_Pr	[13]	1 : Rotate clockwise 90° 0 : Rotator bypass	0	O	O
TargetVsize_Pr	[12:0]	Vertical pixel number of target image for preview DMA. Minimum number is 4. (When Rot90_Pr is set, 8's multiple but, 4's multiple if RGB888/666 mode & H_WIDTH > 160)	0	O	O

TargetHsize_Pr and TargetVsize_Pr must not be larger than Camera SourceHsize and Camera SourceVsize. DMA input source size don't care. Line Buffer size for Rotation is *720 Words* per line. (It means that max TargetHsize is 720 pixels when RGB888/666 & Rot90_Pr=1).

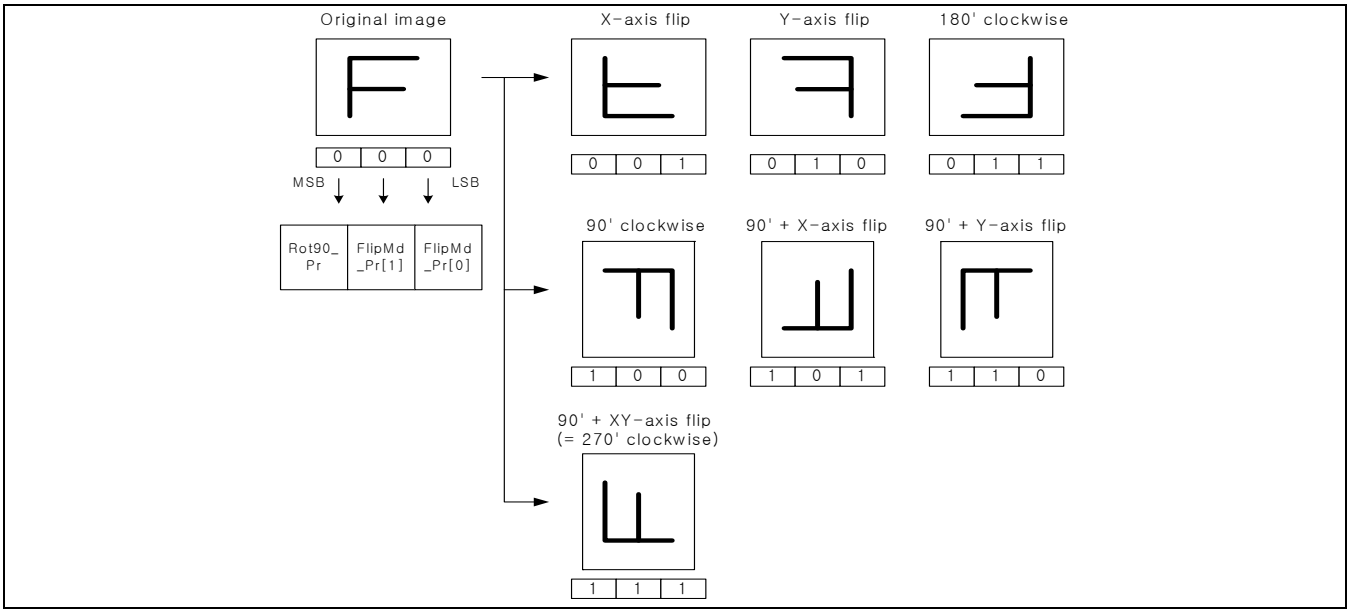


Figure 20-26. Preview image mirror and rotation

20.8.39 PREVIEW DMA CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIPRCTRL	0x780000A0	RW	Preview DMA control related	0

CIPRCTRL	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
Yburst1_Pr	[23:19]	Main burst length for preview Y / RGB frames	0	O	O
Yburst2_Pr	[18:14]	Remained burst length for preview Y / RGB frames	0	O	O
Cburst1_Pr	[13:9]	Main burst length for preview Cb/Cr frames	0	O	O
Cburst2_Pr	[8:4]	Remained burst length for preview Cb/Cr frames	0	O	O
Reserved	[3]		0	X	X
LastIRQEn_Pr	[2]	1 : enable last IRQ at the end of frame capture (One pulse) 0 : normal	0	X	X
Order422_Pr	[1:0]	Interleaved YCbCr 4:2:2 output order memory storing style 00 : LSB $Y_0Cb_0Y_1Cr_0$ MSB 01 : LSB $Y_0Cr_0Y_1Cb_0$ MSB 10 : LSB $Cb_0Y_0Cr_0Y_1$ MSB 11 : LSB $Cr_0Y_0Cb_0Y_1$ MSB	0	O	O

♦ Interleaved burst length (Interleave YCbCr 4:2:2)

	Rot90_Pr = 0	Rot90_Pr = 1
Y burst length	2 , 4 , 8	2(recommend) , 1
C burst length (C burst length = Y burst length / 2)	1 , 2 , 4	No meaning
Wanted burst length (= Y + 2C)	4 , 8 , 16	4(recommend), 2 (=2Y)

NOTE: When Preview output format is YCbCr 4:2:2 interleave ,ScalerBypass_Pr = 0 and ScaleUp_V_Pr = 1 , Wanted main burst length = 16 and Wanted remained burst length ≠16 is not allowed.

♦ Non-Interleaved burst length (Y burst length - YCbCr 4:2:0, YCbCr 4:2:2)

	Rot90_Pr = 0 & YCbCr output	Rot90_Pr = 1 & YCbCr output
Y	Main burst length = 4, 8, 16 Remained burst length = 4, 8, 16	Main burst length = 2 Remained burst length = 2
C	Main burst length = 2, 4, 8, 16 Remained burst length = 2, 4, 8, 16	Main burst length = 1 Remained burst length = 1

♦ Non-Interleaved burst length (RGB burst length)

	Rot90_Pr = 0 & RGB	Rot90_Pr = 1 & RGB18/24bit	Rot90_Pr = 1 & RGB16bit
Y	Main burst length = 4, 8, 16 Remained burst length = 4, 8, 16	Main burst length = 4, 8 Remained burst length = 4, 8	Main burst length = 4 Remained burst length = 4
C	No meaning	No meaning	No meaning

When YCbCr 4:2:2 interleave , burst size calculations are done to determine the wanted burst length. After finding the wanted burst length.

The SFR fields are programmed as described below,

Y : wanted Main burst length = 2 * Yburst1_Pr, and wanted Remained burst length = 2 * Yburst2_Pr.

Cb/Cr : wanted Main burst length = Yburst1_Pr / 2 = Cburst1_Pr,
and wanted Remained burst length = Yburst2_Pr / 2 = Cburst2_Pr

If Rot90_Pr = 0, no rotation, the wanted burst length is calculated from targetHsize.

If Rot90_Pr = 1, rotate90', the wanted burst length is calculated from targetVsize.

When Preview output is InterleaveYCbCr422 and OutRot90_Pr = 1, wanted burst length must be 4.

When Preview output is RGB565 and OutRot90_Pr = 1, wanted burst length must be $\leq$ 4.

When Preview output is Non-Interleave YCbCr and OutRot90_Pr = 1, wanted burst length must be 2(Y) and 1(C).

If Preview output is RGB888/666 mode and OutRot90_Pr = 1, and the original target horizontal width size before rotation(TargetHsize) is lager than 160 pixels, wanted burst length must be $\leq$ 4. But, the original target horizontal width size before rotation(TargetHsize) is smaller than 160 pixels, wanted burst length must be $\leq$ 8.

If Preview output is YCbCr mode (Non-interleave + Interleave) and OutRot90_Pr = 1, the original target horizontal width size before rotation(TargetHsize) must be smaller than 160 pixels and TargetVsize must be 8's multiple.

Caution! Preview path contains 320 pixel line buffer for rotation. Therefore, upper 640 pixels, input images must be pre-scaled by over 1/2 for capturing valid preview image.

20.8.40 PREVIEW PRE-SCALER CONTROL REGISTER 1

Register	Address	R/W	Description	Reset Value
CIPRSCPRERATIO	0x780000A4	RW	Preview pre-scaler ratio control	0

CIPRSCPRERATIO	Bit	Description	Initial State	M	L
SHfactor_Pr	[31:28]	Shift factor for preview pre-scaler	0	O	O
Reserved	[27:23]		0	X	X
PreHorRatio_Pr	[22:16]	Horizontal ratio of preview pre-scaler	0	O	O
Reserved	[15:7]		0	X	X
PreVerRatio_Pr	[6:0]	Vertical ratio of preview pre-scaler	0	O	O

NOTE: PreVerRatio must be the less than 8.

20.8.41 PREVIEW PRE-SCALER CONTROL REGISTER 2

Register	Address	R/W	Description	Reset Value
CIPRSCPREDST	0x780000A8	RW	Preview pre-scaler destination format	0

CIPRSCPREDST	Bit	Description	Initial State	M	L
Reserved	[31:28]		0	X	X
PreDstWidth_Pr	[27:16]	Destination width for preview pre-scaler	0	O	O
Reserved	[15:12]		0	X	X
PreDstHeight_Pr	[11:0]	Destination height for preview pre-scaler	0	O	O

20.8.42 PREVIEW MAIN-SCALER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCCTRL	0x780000AC	RW	Preview main-scaler control	0x18000000

CIPRSCCTRL	Bit	Description	Initial State	M	L
ScalerBypass_Pr	[31]	Preview scaler bypass. In this case, ImgCptEn_PrSC must be 0, but ImgCptEn must be 1. Generally this mode uses large image size than preview scaler maximum size. ScalerBypass has some restriction. It is not allowed size scaling, color space conversion and rotator. Therefore, Input / output format is allowed YCbCr non-interleave 4:2:0,4:2:2 & interleave 4:2:2	0	O	O
ScaleUp_H_Pr	[30]	Horizontal scale up/down flag for preview scaler (In 1:1 scale ratio, this bit must be “1”) 1: up, 0:down	0	O	O
ScaleUp_V_Pr	[29]	Vertical scale up/down flag for preview scaler (In 1:1 scale ratio, this bit must be “1”) 1: up, 0:down	0	O	O
CSCR2Y_Pr	[28]	YCbCr Data Dynamic Range Selection for the Color Space Conversion RGB to YCbCr (Preview path) 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
CSCY2R_Pr	[27]	YCbCr Data Dynamic Range Selection for the Color Space Conversion YCbCr to RGB (Preview path) 1 : Wide => Y/Cb/Cr (0 ~ 255) : Wide default 0 : Narrow => Y (16 ~ 235), Cb/Cr (16 ~ 240)	1	O	O
LCDPathEn_Pr	[26]	FIFO Mode Enable. 1 for FIFO mode and 0 for DMA mode FIFO mode output format is YCbCr4:4:4 or RGB24bit. its selection depends on OutFormat register. OutFormat_Pr = RGB → RGB24bit, otherwise YCbCr4:4:4	0	O	O
Interlace_Pr	[25]	Output scan method selection register only when FIFO mode (LCDPathEn =1). 1 for Interlace scan and 0 for progressive scan. In DMA mode (LCDPathEn = 0), progressive scan is applied whatever this value has Interlace mode is allowed when MSDMA input data & is not allowed when Camera processor input data	0	O	O
MainHorRatio_Pr	[24:16]	Horizontal scale ratio for preview main-scaler	0	O	O
PrScalerStart	[15]	Preview scaler start. This bit must be zero in preview scaler-bypass mode. 1 : scaler start 0 : scaler stop	0	O	O
InRGB_FMT_Pr	[14:13]	Input RGB format MSDMA for preview path dedicated 00 : RGB565 , 01 : RGB666 , 10 : RGB888 , 11 : Forbidden	0	O	O
OutRGB_FMT_Pr	[12:11]	Output RGB format for Preview write DMA 00 : RGB565 , 01 : RGB666 , 10 : RGB888 , 11 : Forbidden	0	O	O

CIPRSCCTRL	Bit	Description	Initial State	M	L
Ext_RGB_Pr	[10]	Input RGB data extension enable bit for the conversion of RGB565/666 mode into RGB888 mode for preview path 1 : Extension , 0 : normal i) Input R = 5-bit in RGB565 mode *101*00 -> 10100*101* (Extension) 10100 -> 10100000 (normal) ii) Input R = 6-bit in RGB666 mode *101*100 -> 10110*010* (Extension) 101100 -> 10110000 (normal)	0	O	O
One2One_Pr	[9]	Non-interpolation data copy. (Caution : this register should be set at 1:1 scaler size for same in-out format and Image effect cannot support RGB format & One2One mode.) **Ex) input YCbCr4:2:0 (VGA) -> output YCbCr4:2:0 (VGA)**	0	O	O
MainVerRatio_Pr	[8:0]	Vertical scale ratio for preview main-scaler	0	O	O

20.8.43 PREVIEW DMA TARGET AREA REGISTER

Register	Address	R/W	Description	Reset Value
CIPRTAREA	0x780000B0	RW	Preview dma target area	0

CIPRTAREA	Bit	Description	Initial State	M	L
Reserved	[31:26]		0	X	X
CIPRTAREA	[25:0]	Target area for preview DMA = Target H size x Target V size	0	O	X

20.8.44 PREVIEW STATUS REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSTATUS	0x780000B8	R/W	Preview path status	0

CIPRSTATUS	Bit	Description	Initial State	M	L
OvFiY_Pr	[31]	Overflow state of preview FIFO Y	0	X	X
OvFiCb_Pr	[30]	Overflow state of preview FIFO Cb	0	X	X
OvFiCr_Pr	[29]	Overflow state of preview FIFO Cr	0	X	X
Reserved	[28]		0	X	X
FrameCnt_Pr	[27:26]	Frame count of preview DMA	0	X	X
Reserved	[25]		0	X	X
FlipMd_Pr	[24:23]	Flip mode of preview DMA	0	X	X
Reserved	[22]		0	X	X
ImgCptEn_PrSC	[21]	Image capture enable of preview path	0	X	X
OvRLB_Pr	[20]	Overflow status of Line Buffer for Rotation in Preview path	0	X	X

FrameEnd_Pr	[19]	When preview frame operation finish, FrameEnd_Pr is generated. and FrameEnd_Pr is clear by user setting '0'	0	X	X
Reserved	[18:0]		0	X	X

20.8.45 IMAGE CAPTURE ENABLE REGISTER

Register	Address	R/W	Description	Reset Value
CIIMGCPT	0x780000C0	RW	Image capture enable command	0

CIIMGCPT	Bit	Description	Initial State	M	L
ImgCptEn	[31]	camera interface global capture enable	0	X	O
ImgCptEn_CoSc	[30]	capture enable for codec scaler. This bit must be zero in codec scaler-bypass mode.	0	X	O
ImgCptEn_PrSc	[29]	capture enable for preview scaler. This bit must be zero in preview scaler-bypass mode.	0	O	O
Reserved	[28:26]		0	X	X
Cpt_FrEn_Co	[25]	Capture codec frame control. (only camera input is applied) 1 : Enable (Step-by-Step frame one shot mode) 0 : Disable (FreeRun mode)	0	X	O
Cpt_FrEn_Pr	[24]	Capture preview frame control. (only camera input is applied) 1 : Enable (Step-by-Step frame one shot mode) 0 : Disable (Free Run mode)	0	X	O
Cpt_FrPtr	[23:19]	Capture sequence turn-around pointer (Common preview & codec)	0	X	X
Cpt_ FrMod	[18]	Capture frame control mode (Common preview & codec) 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 will be repeated until capture frame control disable)	0	X	X
Cpt_ FrCnt	[17:10]	Wanted number of frames to be captured (Common preview & codec). When read, you will see the value of a shadow register which is downcounted when 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]		0	X	X

20.8.46 CAPTURE CONTROL SEQUENCE REGISTER

Register	Address	R/W	Description	Reset Value
CICPTSEQ	0x780000C4	RW	Camera image capture sequence related	FFFF_FFFF

CICPTSEQ	Bit	Description	Initial State	M	L
Cpt_FrSeq	[31:0]	Capture sequence pattern	FFFF_FFFF	X	X

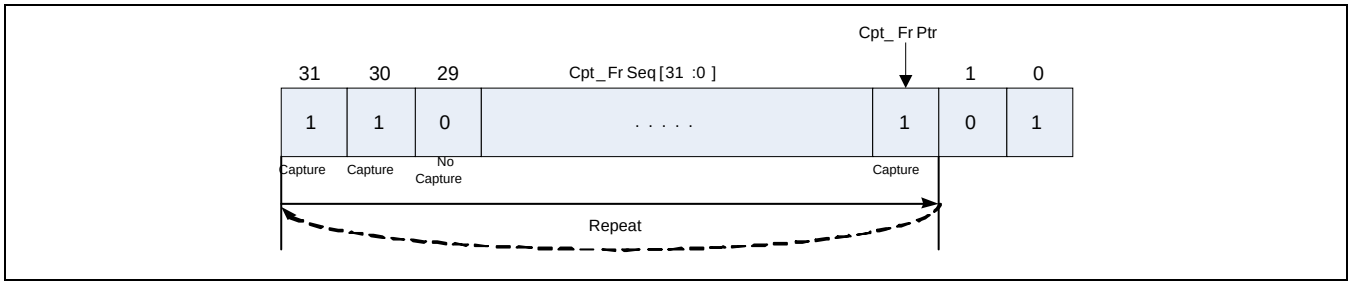


Figure 20-27. Capture frame control

- For skipped frmes, IRQ_CI is not generated. And FrameCnt is not increased.

20.8.47 IMAGE EFFECTS REGISTER

Register	Address	R/W	Description	Reset Value
CIIMGEFF	0x780000D0	RW	Image Effects related	0010_0080

CIIMGEFF	Bit	Description	Initial State	M	L
IE_ON_Pr	[31]	0 : image effect function disable in preview path , 1 : enable	0	O	O
IE_ON_Co	[30]	0 : image effect function disable in codec path , 1 : enable	0	O	O
IE_AFTER_SC	[29]	Image Effect location 1: After scaling (camera and MSDMA image are applied except scaler bypass mode) 0: Before scaling (only camera image must be applied)	0	O	O
FIN	[28:26]	Image Effect selection. (Common preview & codec path) 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]		0	X	X
PAT_Cb	[20:13]	It is used only for FIN is Arbitrary Cb/Cr (Common preview & codec path) (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

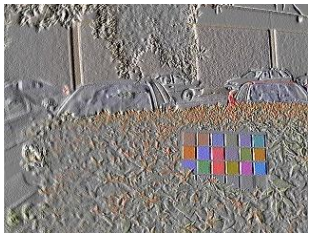
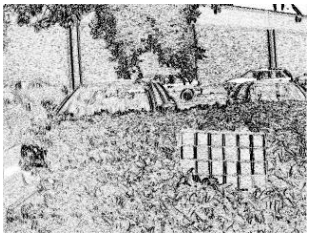
CIIMGEFF	Bit	Description	Initial State	M	L
Reserved	[12:8]		0	X	X
PAT_Cr	[7:0]	It is used only for FIN is Arbitrary Cb/Cr (Common preview & codec path) (PAT_Cb/Cr == 8'd128 for GRAYSCALE) Wide CSC Range : 0 ≤ PAT_Cr ≤ 255 Narrow CSC Range : 16 ≤ PAT_Cr ≤ 240	8'd128	O	O

Cf) sepia : PAT_Cb == 8'd115 , PAT_Cr == 8'd145

Original Arbitrary(sepia) Negative



Art freeze Embossing Silhouette

20.8.48 MSDMA FOR CODEC Y START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSCOY0SA	0x780000D4	RW	MSDMA Y0 start address related	0000_0000

MSCOYSA	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSCOY0SA	[30:0]	DMA start address for Y component (non-interleave YCbCr 4:2:0, 4:2:2) DMA start address for Interleave YCbCr 4:2:2 / RGB component	0	O	X

20.8.49 MSDMA FOR CODEC CB START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCB0SA	0x780000D8	RW	MSDMA Cb0 start address related	0000_0000

MSCBSA	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSCOCB0SA	[30:0]	DMA start address for Cb component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.50 MSDMA FOR CODEC CR START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCR0SA	0x780000DC	RW	MSDMA Cr0 start address related	0000_0000

MSCOCRSA	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSCOCR0SA	[30:0]	DMA start address for Cr component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.5.51 MSDMA FOR CODEC Y END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSCOY0END	0x780000E0	RW	MSDMA Y0 end address related	0000_0000

MSCOYEND	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSCOY0END	[30:0]	DMA End address for Y component (non-interleave YCbCr 4:2:0, 4:2:2) DMA End address for Interleave YCbCr 4:2:2 / RGB component	0	O	X

20.8.52 MSDMA FOR CODEC CB END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCB0END	0x780000E4	RW	MSDMA Cb0 end address related	0000_0000

MSCOCBEND	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSCOCB0END	[30:0]	DMA End address for Cb component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.53 MSDMA CR END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCR0END	0x780000E8	RW	MSDMA Cr0 end address related	0000_0000

MSCOCREND	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSCOCR0END	[30:0]	DMA End address for Cr component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.54 MSDMA FOR CODEC Y OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
MSCOYOFF	0x780000EC	RW	MSDMA Y offset related	0000_0000

MSCOYOFF	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
MSCOYOFF	[23:0]	Offset of Y component for fetching source image (non-interleave YCbCr 4:2:0, 4:2:2) Offset of Interleave YCbCr 4:2:2 / RGB component for fetching source image	0	O	X

20.8.55 MSDMA FOR CODEC CB OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCBOFF	0x780000F0	RW	MSDMA Cb offset related	0000_0000

MSCOCBOFF	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
MSCOCBOFF	[23:0]	Offset of Cb component for fetching source image(non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.56 MSDMA FOR CODEC CR OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCROFF	0x780000F4	RW	MSDMA Cr offset related	0000_0000

MSCOCROFF	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
MSCOCROFF	[23:0]	Offset of Cr component for fetching source image(non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.57 MSDMA FOR CODEC SOURCE IMAGE WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
MSCOWIDTH	0x780000F8	RW	MSDMA source image width related	0000_0000

MSCOWIDTH	Bit	Description	Initial State	M	L
AutoLoadEnable	[31]	MSDMA Automatically restart (Only Software trigger mode) At the first frame start requires ENVID start setting. After first frame, next frame does not need ENVID setting. 0 : AutoLoad Disable , 1 : AutoLoad Enable	0	O	X
ADDR_CH_DIS	[30]	MSDMA 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
Reserved	[29:28]		0	X	X
MSCOHEIGHT	[27:16]	MSDMA source image vertical pixel size. minimum 8. It must be multiple of PreVerRatio.	0	O	X
Reserved	[15:12]		0	X	X
MSCOWIDTH	[11:0]	MSDMA source image horizontal pixel size (must be 8's multiple. It must be 4's multiple of PreHorRatio. minimum 16)	0	O	X

♦ **MSDMA Start address**

Start address of ADDRStart_Y/Cb/Cr/RGB points the first word address where the corresponding component of Y/Cb/Cr/RGB is read or written. Each one must be aligned with word boundary (i.e. ADDRStart_X[1:0] = 00). ADDRStart_Cb and ADDRStart_Cr are valid only for the non-interleave YCbCr420, 422 source image format.

♦ **MSDMA End address**

ADDREnd_Y

= ADDRStart_Y + Memory size for the component of Y/RGB/YCbCr(interleave)

= ADDRStart_Y + (SRC_Width × SRC_Height) × *ByteSize_Per_Pixel* + Offset_Y × (SRC_Height-1)

ADDREnd_Cb (Valid for YCbCr420/422 non-interleave source format)

= ADDRStart_Cb + Memory size for the component of Cb

= ADDRStart_Cb + (SRC_Width/2 × *Real_Height*) × *ByteSize_Per_Pixel* + Offset_Cb × (*Real_Height* -1)

ADDREnd_ Cr (Valid for YCbCr420/422 non-interleave source format)

= ADDRStart_Cr + Memory size for the component of Cr

= ADDRStart_Cr + (SRC_Width/2 × *Real_Height*) × *ByteSize_Per_Pixel* + Offset_Cr × (*Real_Height* -1)

♦ **MSDMA OFFSET**

Offset_Y/Cb/Cr/RGB

= Memory size for offset per a horizontal line

= Number of pixel (or sample) in horizontal offset × *ByteSize_Per_Pixel* (or Sample)

Cf.) ByteSize_Per_Pixel = { 1 for YCbCr420 / YCbCr422 (non-interleave); 2 for YCbCr422 (interleave) / RGB 16-bit; 4 for RGB 18/24-bit

Real_Height = { SRC_Height / 2 : YCbCr 420 case; SRC_Height : YCbCr 422 (non-interleave) case

20.8.58 MSDMA FOR CODEC CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MSCOCTRL	0x780000FC	RW	MSDMA for codec control register	0000_0000

MSCOCTRL	Bit	Description	Initial State	M	L
Reserved	[31:7]		0	X	X
EOF_M_C	[6]	When MSDMA operation done, End Of Frame will be generated. (read only this signal)	0	O	X
Order422_M_C	[5:4]	When source MSDMA image is interleaved YCbCr 4:2:2, Interleaved YCbCr 4:2:2 input order style. [4:3] LSB MSB 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	O	X
SEL_DMA_CAM_C	[3]	Codec path input data selection. 0 : External camera input path 1 : Memory data input path (MSDMA)	0	O	X
InFormat_M_C	[2:1]	Source image format for MSDMA 00 : YCbCr 4:2:0 01 : YCbCr 4:2:2 (non-interleave) 10 : YCbCr 4:2:2 (interleave) 11 : RGB (cf. RGB format register → InRGB_FMT_Co)	0	O	X

MSCOCTRL	Bit	Description	Initial State	M	L
ENVID_M_C	[0]	MSDMA operation start. (When triggered Low to High by software setting) Hardware doesn't clear automatically This register is allowed set only Software Trigger mode. If Hardware trigger mode, this bit is read only. 1) SEL_DMA_CAM = '0' , ENVID don't care (using external camera signal for codec path) 2) SEL_DMA_CAM = '1', ENVID is set (0→1) then MSDMA operation start for codec.	0	O	X

NOTE: ENVID_M_C SFR must be set at last. Starting order for using MSDMA input path.
SEL_DMA_CAM_C (others SFR setting) → Image Capture Enable SFR setting → ENVID_M_C SFR setting.

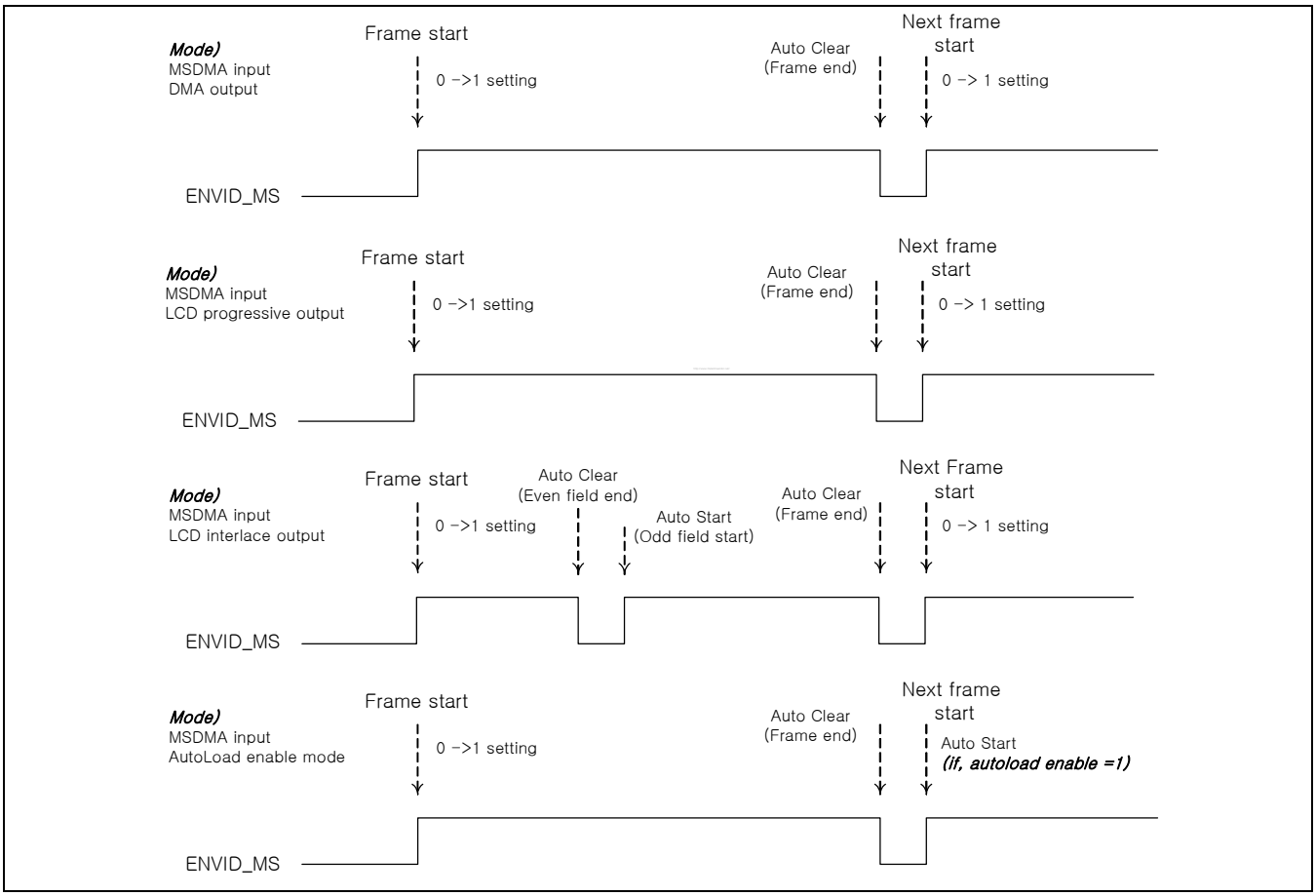


Figure 20-28. ENVID_MS SFR setting when DMA start to read memory data

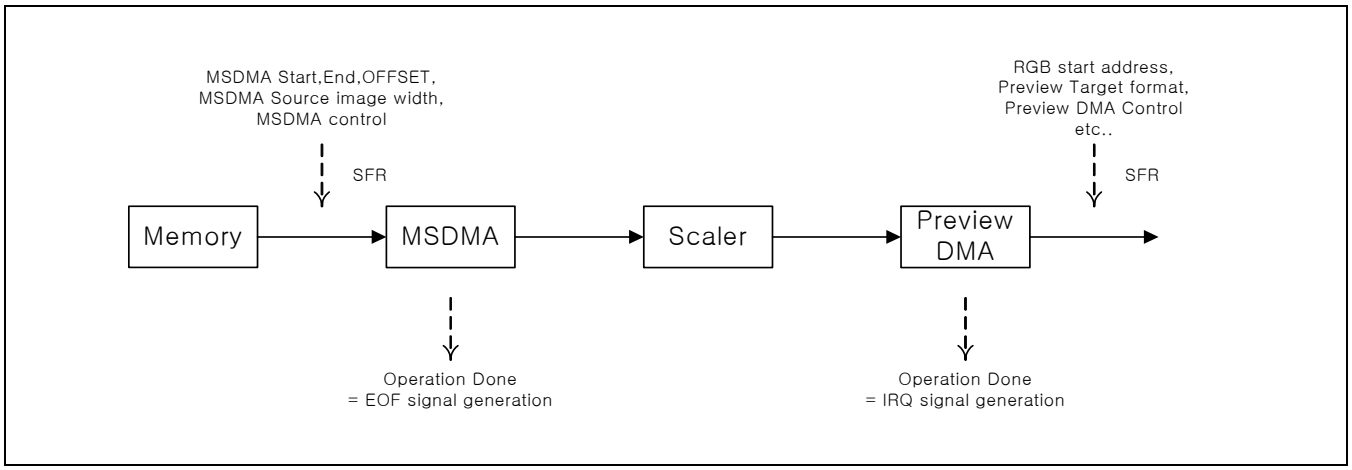


Figure 20-29. SFR & Operation (related each DMA when selected MSDMA input path)

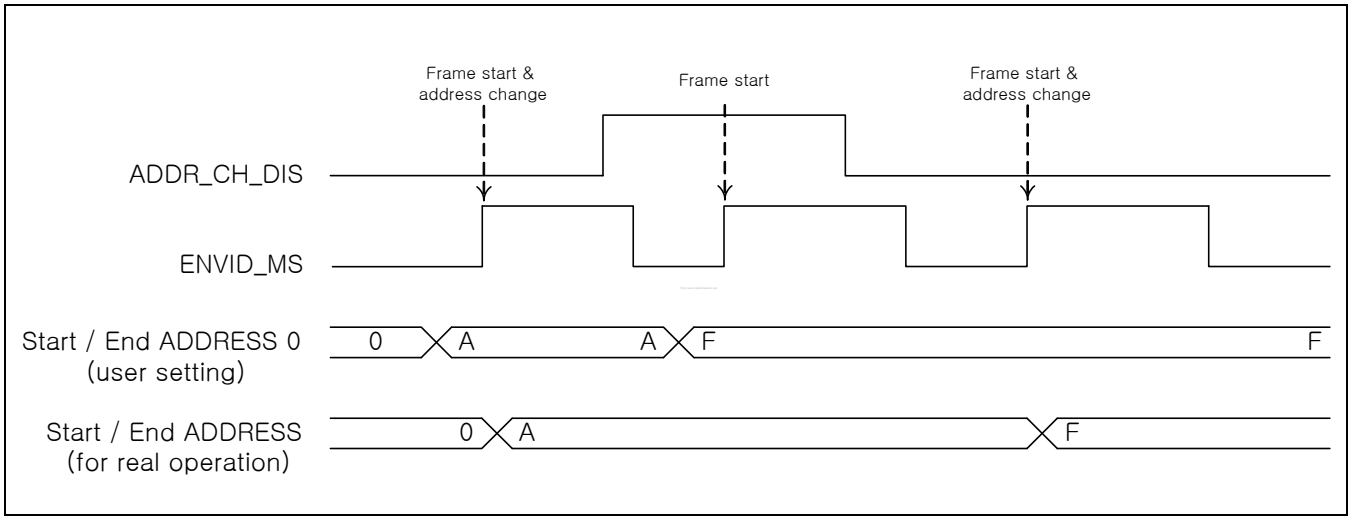


Figure 20-30. ADDRESS Change timing (related MSDMA)

20.8.59 MSDMA FOR PREVIEW Y0 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSPRY0SA	0x78000100	RW	MSDMA Y0 start address related	0000_0000

MSYSA	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSPRY0SA	[30:0]	DMA start address for Y component (non-interleave YCbCr 4:2:0, 4:2:2) DMA start address for Interleave YCbCr 4:2:2 / RGB component	0	O	X

20.8.60 MSDMA FOR PREVIEW CB0 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCB0SA	0x78000104	RW	MSDMA Cb0 start address related	0000_0000

MSCBSA	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSPRCB0SA	[30:0]	DMA start address for Cb component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.61 MSDMA FOR PREVIEW CR0 START ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCR0SA	0x78000108	RW	MSDMA Cr0 start address related	0000_0000

MSPRCRSA	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSPRCR0SA	[30:0]	DMA start address for Cr component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.62 MSDMA FOR PREVIEW Y0 END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSPRY0END	0x7800010C	RW	MSDMA Y0 end address related	0000_0000

MSPRYEND	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSPRY0END	[30:0]	DMA End address for Y component (non-interleave YCbCr 4:2:0, 4:2:2) DMA End address for Interleave YCbCr 4:2:2 / RGB component	0	O	X

20.8.63 MSDMA FOR PREVIEW CB0 END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCB0END	0x78000110	RW	MSDMA Cb0 end address related	0000_0000

MSPRCBEND	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSPRCB0END	[30:0]	DMA End address for Cb component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.64 MSDMA CR0 END ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCR0END	0x78000114	RW	MSDMA Cr0 end address related	0000_0000

MSPRCREND	Bit	Description	Initial State	M	L
Reserved	[31]		0	X	X
MSPRCR0END	[30:0]	DMA End address for Cr component (non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.65 MSDMA FOR PREVIEW Y OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
MSPRYOFF	0x78000118	RW	MSDMA Y offset related	0000_0000

MSPRYOFF	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
MSPRYOFF	[23:0]	Offset of Y component for fetching source image (non-interleave YCbCr 4:2:0, 4:2:2) Offset of Interleave YCbCr 4:2:2 / RGB component for fetching source image	0	O	X

20.8.66 MSDMA FOR PREVIEW CB OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCBOFF	0x7800011C	RW	MSDMA Cb offset related	0000_0000

MSPRCBOFF	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
MSPRCBOFF	[23:0]	Offset of Cb component for fetching source image(non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.67 MSDMA FOR PREVIEW CR OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCROFF	0x78000120	RW	MSDMA Cr offset related	0000_0000

MSPRCROFF	Bit	Description	Initial State	M	L
Reserved	[31:24]		0	X	X
MSPRCROFF	[23:0]	Offset of Cr component for fetching source image(non-interleave YCbCr 4:2:0, 4:2:2)	0	O	X

20.8.68 MSDMA FOR PREVIEW SOURCE IMAGE WIDTH/HEIGHT & AUTOLOAD REGISTER

Register	Address	R/W	Description	Reset Value
MSPRWIDTH	0x78000124	RW	MSDMA source image width related	0000_0000

MSPRWIDTH	Bit	Description	Initial State	M	L
AutoLoadEnable	[31]	MSDMA Automatically restart (Only Software trigger mode) At the first frame start needs ENVID start setting. After first frame, next frame does not need ENVID setting. 0 : AutoLoad Disable , 1 : AutoLoad Enable	0	O	X
ADDR_CH_DIS	[30]	MSDMA 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
Reserved	[29:28]		0	X	X
MSPRHEIGHT	[27:16]	MSDMA source image vertical pixel size. minimum 8. Also, must be multiple of PreVerRatio. If InRot90_Pr = 1, must be 8's multiple. Also, must be 4's multiple of PreHorRatio. minimum 16	0	O	X
Reserved	[15:12]		0	X	X
MSPRWIDTH	[11:0]	MSDMA source image horizontal pixel size (must be 8's multiple. Also, must be 4's multiple of PreHorRatio. minimum 16)	0	O	X

20.8.69 MSDMA FOR PREVIEW CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
MSPRCTRL	0x78000128	RW	MSDMA control register for preview	0000_0000

MSPRCTRL	Bit	Description	Initial State	M	L
Reserved	[31:7]		0	X	X
EOF_M_P	[6]	When MSDMA operation done, End Of Frame will be generated. (read only this signal)	0	O	X
Order422_M_P	[5:4]	When source MSDMA image is interleaved YCbCr 4:2:2, Interleaved YCbCr 4:2:2 input style. [4:3] LSB MSB 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	O	X
SEL_DMA_CAM_P	[3]	Preview path data selection. codec path don't care. 0 : External camera input path 1 : Memory data input path (MSDMA)	0	O	X
InFormat_M_P	[2:1]	Source image format for MSDMA 00 : YCbCr 4:2:0 01 : YCbCr 4:2:2 (non-interleave) 10 : YCbCr 4:2:2 (interleave) 11 : RGB (cf. RGB format register → InRGB_FMT_Pr)	0	O	X
ENVID_M_P	[0]	MSDMA operation start. (When triggered Low to High by software setting) Hardware clear automatically. This bit is allowed set only Software Trigger mode. If Hardware trigger mode, this bit is read only. 1) SEL_DMA_CAM = '0' , ENVID don't care (using external camera signal for preview path) 2) SEL_DMA_CAM = '1', ENVID is set (0→1) then MSDMA operation start for preview.	0	O	X

20.8.70 CODEC SCAN LINE Y OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CICOSCOSY	0x7800012C	RW	Codec scan line Y offset related	0

CICOSCOS	Bit	Description	Initial State	M	L
Reserved	[31:29]		0	X	X
Initial_Yoffset_Co	[28:16]	The number of the skipped pixels for initial Y offset. scanline Y offset can be used when Non-interleaved Y or Interleaved YCbCr422 or RGB format.	0	X	O
Reserved	[15:13]		0	X	X
Line_Yoffset_Co	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed. Scanline offset can be used when Non-interleaved Y or Interleaved YCbCr422 or RGB format	0	X	O

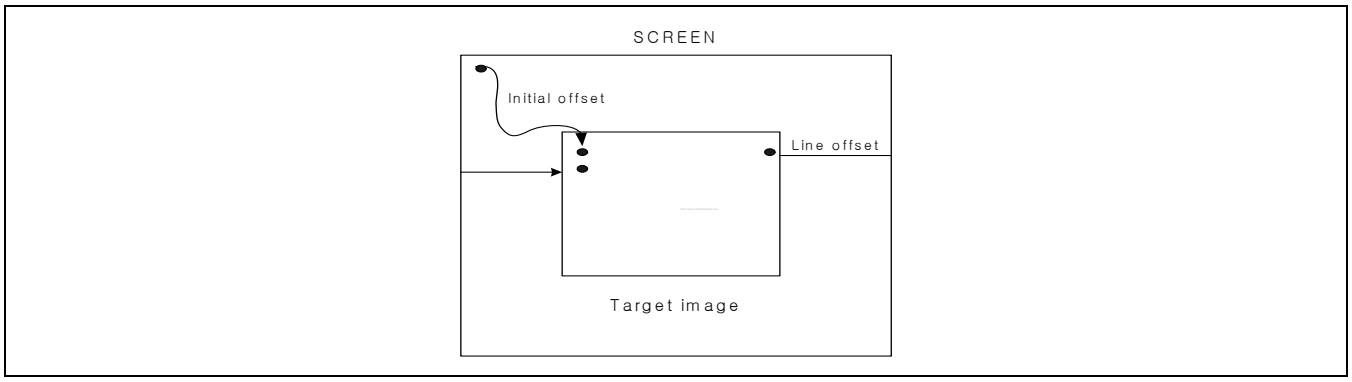


Figure 20-31. Scan line offset

NOTE: Scan line Offset constraint (for word boundary alignment)
- Initial / Line Y
Non-interleave Y : 4's multiple pixel (minimum 4)
Interleaved YCbCr 422 / RGB565 : 2's multiple (minimum 2)
RGB666/888 : 1's multiple (minimum 1)

- Initial / Line Cb,Cr (YCbCr 4:2:2 , 4:2:0)
Non-interleave Cb / Cr : 8's multiple pixel (minimum 8)

20.8.71 CODEC SCAN LINE CB OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CICOSCOSCB	0x78000130	RW	Codec scan line Cb offset related	0

CICOSCOS	Bit	Description	Initial State	M	L
Reserved	[31:29]		0	X	X
Initial_Cboffset_Co	[28:16]	The number of the skipped pixels for initial Cb offset. scanline Cb offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O
Reserved	[15:13]		0	X	X
Line_Cboffset_Co	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed. Scanline Cr offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O

20.8.72 CODEC SCAN LINE CR OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CICOSCOSCR	0x78000134	RW	Codec scan line Cr offset related	0

CICOSCOS	Bit	Description	Initial State	M	L
Reserved	[31:29]		0	X	X
Initial_Croffset_Co	[28:16]	The number of the skipped pixels for initial Cr offset. Scanline Cr offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O
Reserved	[15:13]		0	X	X
Line_Croffset_Co	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed. Scanline Cr offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O

20.8.73 PREVIEW SCAN LINE Y OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCOSY	0x78000138	RW	Preview scan line Y offset related	0

CIPRSCOS	Bit	Description	Initial State	M	L
Reserved	[31:29]		0	X	X
Initial_Yoffset_Pr	[28:16]	The number of the skipped pixels for initial Y offset. scanline Y offset can be used when Non-interleaved Y or Interleaved YCbCr422 or RGB format.	0	O	O
Reserved	[15:13]		0	X	X
Line_Yoffset_Pr	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed. Scanline offset can be used when Non-interleaved Y or Interleaved YCbCr422 or RGB format	0	O	O

20.8.74 PREVIEW SCAN LINE CB OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCOSCB	0x7800013c	RW	Preview scan line Cb offset related	0

CICOSCOS	Bit	Description	Initial State	M	L
Reserved	[31:29]		0	X	X
Initial_Cboffset_Pr	[28:16]	The number of the skipped pixels for initial Cb offset. scanline Cb offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O
Reserved	[15:13]		0	X	X
Line_Cboffset_Pr	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed. Scanline Cr offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O

20.8.75 PREVIEW SCAN LINE CR OFFSET REGISTER

Register	Address	R/W	Description	Reset Value
CIPRSCOSCR	0x78000140	RW	Preview scan line Cr offset related	0

CICOSCOS	Bit	Description	Initial State	M	L
Reserved	[31:29]		0	X	X
Initial_Croffset_Pr	[28:16]	The number of the skipped pixels for initial Cr offset. Scanline Cr offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O
Reserved	[15:13]		0	X	X
Line_Croffset_Pr	[12:0]	The number of the skipped pixels in the screen of the target image when scan line is changed. Scanline Cr offset can be used when Non-interleaved YCbCr4:2:0 / 4:2:2	0	X	O

21 MULTI-FORMAT VIDEO CODEC

This chapter describes the function and usages of Multi-Format Video codec in S3C6410X RISC microprocessor.

21.1 OVERVIEW

FIMV-MFC V1.0 is a high-performance video codec IP that supports H.263P3, MPEG-4 SP, H.264 and VC-1. FIMV-MFC V1.0 consists of the embedded BIT processor and video codec core module. The BIT processor parses or forms bitstream and controls the video codec. To speed up the bitstream processing, some hardware accelerators are included in the BIT processor. The program and data for the BIT processor are downloaded through the AMBA APB bus and the AMBA AXI bus.

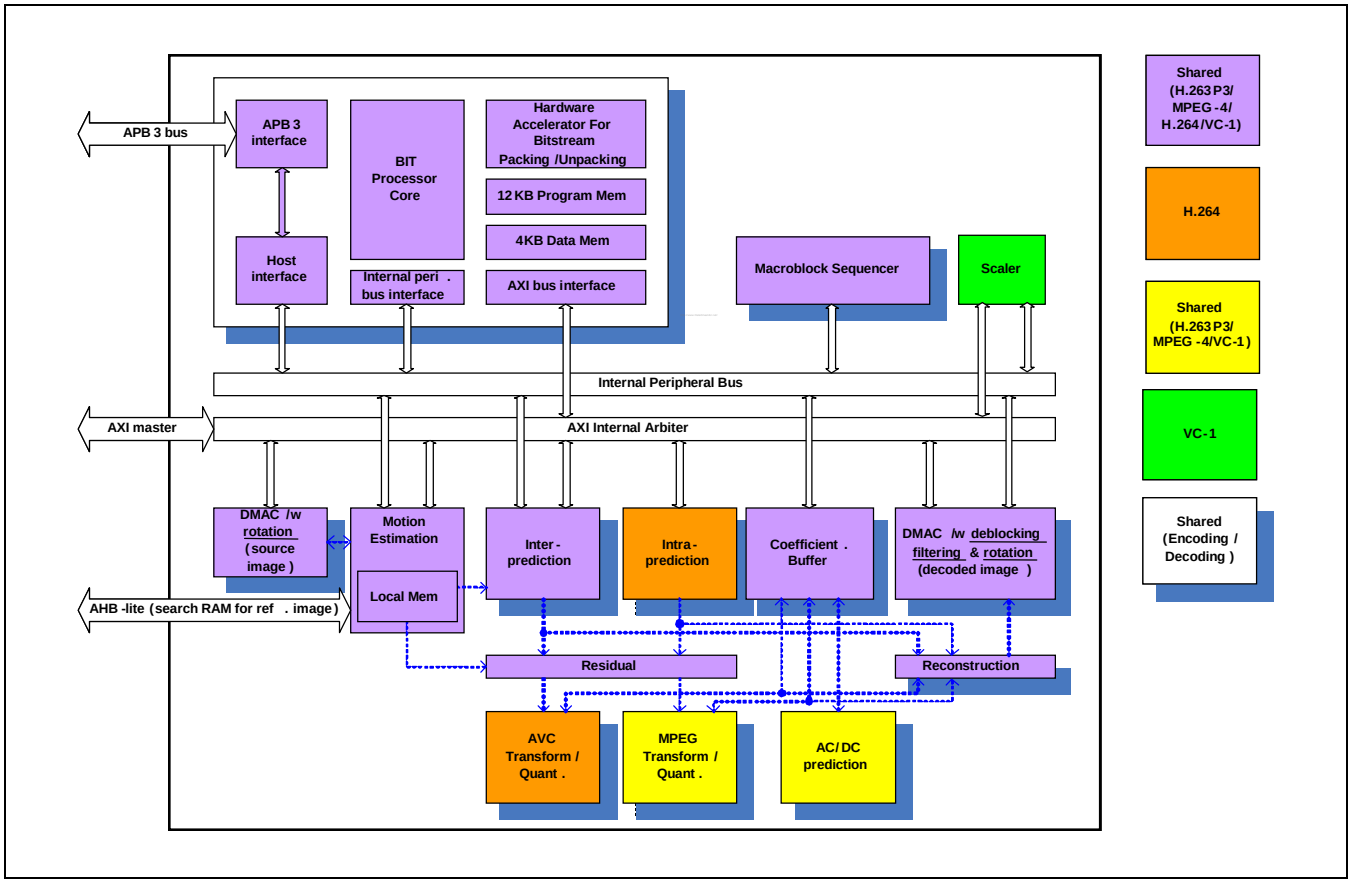


Figure 21-1. FIMV-MFC V1.0 block diagram

FIMV-MFC V1.0 video codec is optimized to reduce the logic gate count with sharing large parts of sub-modules for multi-standard. Motion estimation module uses a search RAM to reduce the bandwidth on the external SDRAM. Generally, motion estimation reads reference pixel data several times. The motion estimation module loads the reference pixel data from the external SDRAM and store them into the search RAM. The search RAM is accessed through the AMBA AHB.

The macroblock sequencer module schedules the processing flow of the functional blocks of the video codec to reduce loads on the BIT processor and complexity of the firmware. FIMV-MFC V1.0 includes a rotation/mirroring module. In case of rotating and/or mirroring the source image in the encoder, no additional bandwidth is required for the processing. However, in the decoder, the decoded image with any rotation and/or mirroring is written to the external memory.

The internal AXI arbiter module arbitrates requests from internal DMA controllers to ease the integration to user's SoC.

Figure 21-2 describes roles of the BIT processor, video codec core module and how to interface with application software. Basically, at the frame level, a host processor communicates with FIMV-MFC V1.0 through provided API's. To give the video codec more flexibility and debugging capability, all processes related to the bitstream are assigned to the BIT processor.

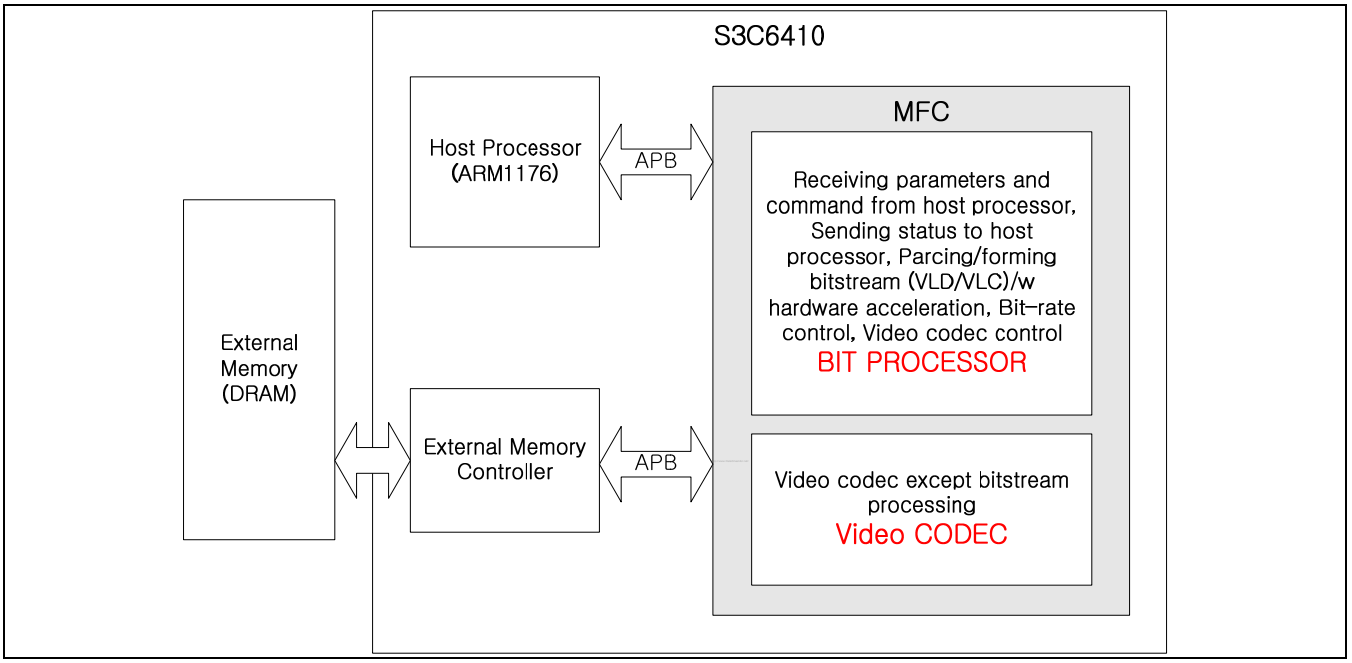


Figure 21-2. Roles of the BIT processor and the video codec module

21.2 FEATURES

FIMV-MFC V1.0 is a multi-standard video codec IP that can handle the H.263P3, MPEG-4 Single Profile H.264 Baseline Profile and VC-1 Main Profile in single codec hardware. The FIMV-MFC V1.0 includes the following features.

- Multi-standard video codec
  - MPEG-4 simple profile encoding/decoding
  - H.264/AVC baseline profile encoding/decoding
  - H.263 P3 encoding/decoding
  - VC-1(WMV9) main profile decoding
  - Multi-party call is supported

Ex. Simultaneous 1 stream encoding and 3 streams decoding are possible.

– Multi-format is supported

Ex. The video IP encodes the MPEG-4 bitstream, and decodes H.264 bitstream in a time-division multiplexing manner.

- Coding tools
  - [-16,+16] 1/2 and 1/4-pel accuracy motion estimation
  - All variable block sizes are supported.
    * In case of encoding, 8x4, 4x8, and 4x4 block sizes are not supported.
  - Unrestricted motion vector
  - MPEG-4 AC/DC prediction
  - H.264/AVC intra-prediction
  - H.263 Annex I, J, K(RS=0), and T are supported
    * In case of encoding, the Annex I and K (RS=1) are not supported.
  - Error resilience tools
    MPEG-4 resync. marker & data-partitioning with RVLC

    * Fixed number of bits/macroblocks between macroblocks

    H.264/AVC FMO
  - Bit-rate control (CBR:Constant Bit Rate & VBR:Variable Bit Rate)
  - CIR(Cyclic Intra Refresh)/AIR(Adaptive Intra Refresh)
- Pre/post rotation/mirroring
  - 8 rotation/mirroring modes for incoming image at encoder
  - 8 rotation/mirroring modes for output image at decoder
- Programmability
  - FIMV-MFC V1.0 embeds 16-bit DSP processor that is dedicated to processing bitstream and controlling the codec hardware.
  - General purpose registers and interrupt for communication between a host processor and the video IP
- Performance
  - Up to full-duplex VGA 30fps encoding/decoding
  - Up to half-duplex 720x480 30fps(720x576 25fps) encoding/ decoding

21.3 THE BIT PROCESSOR

This section describes the BIT processor that is optimized to process bitstream in various formats such as MPEG-4, H.263, H.264 and VC-1.

The BIT processor is an embedded programmable 16-bit DSP that is highly optimized to handle bitstream data. In addition to processing bitstream, the BIT processor controls the video codec and communicates with a host processor through the host interface. The BIT processor has program memory of 12KB and data memory of 4KB.

Figure 21-3 displays the block diagram of the BIT processor. The special registers include command, interrupt, and code download registers. The general purpose registers, 64 32-bit registers, can be used for the host processor to send parameters to the video codec. If an application needs more than 64 registers, an external memory can be used for extension because the BIT processor can access the external memory through the AXI bus interface.

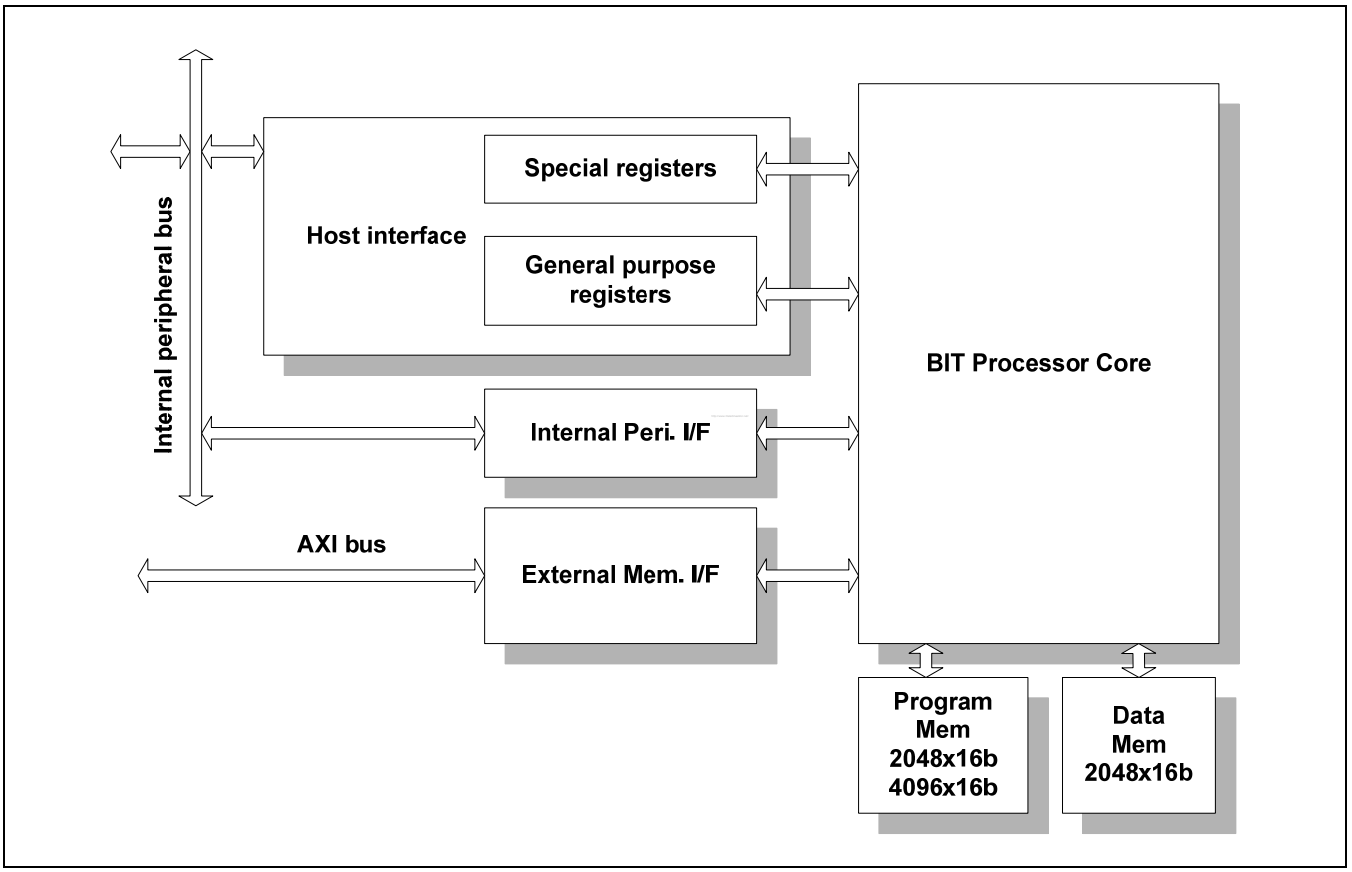


Figure 21-3. BIT processor block diagram

21.3.1 HARDWARE ACCELERATION

The BIT processor core embeds hardware acceleration sub-modules as followings.

- Accelerator to support bitstream packing instruction such as *put_bits*
- Accelerator to support bitstream unpacking instructions such as *get_bits* and *show_bits*
- Look-up table and searching module for VLC and VLD operation
- Motion vector prediction/reconstruction module
- DMA controller to transfer bitstream from/to external SDRAM

21.3.2 DOWNLOADING THE FIRMWARE

The BIT firmware to drive the video codec module and interface with a host processor is divided into 2 parts. One is for boot code that is downloaded by the host processor through the APB bus. The size of the boot code is 1024-byte. Another is for codes for codec processing such as MPEG-4, H.263, H.264 and VC-1. The procedure for downloading the firmware is executed only once at the initialization step.

Boot code

Before running codec, a host processor must download the BIT boot code to the program memory, 2048x16 synchronous single-port SRAM, through the host interface as following.

```
Step 1. addr = 0;

Step 2. code_data = (addr << 16) + bit_code[addr]

Step 3. write code_data to the CodeDownLoad register of the host
        interface embedded in the BIT processor

Step 4. addr = addr + 1

Step 5. if (addr < 512) go to step 1 else go to the procedure to
         download the codec firmware (refer to the page 21-6)
```

The bit_code is an array of which size is 512 * 16-bit.

NOTE:

Above bit_code[0-511] has to be also written in a region specified in the *CodeBufAddr* register of the host interface.

Codec firmware

In addition to the boot code, a package of the firmware for driving the IP is required. Basically, the package for MPEG-4, H.263P3, H.264 and VC-1 codec is provided. You must write the firmware to a region of external memory and send information about the base address of the region by writing it to the *CodeBufAddr* register. At run-time, part of firmware is automatically loaded to the internal BIT memory.

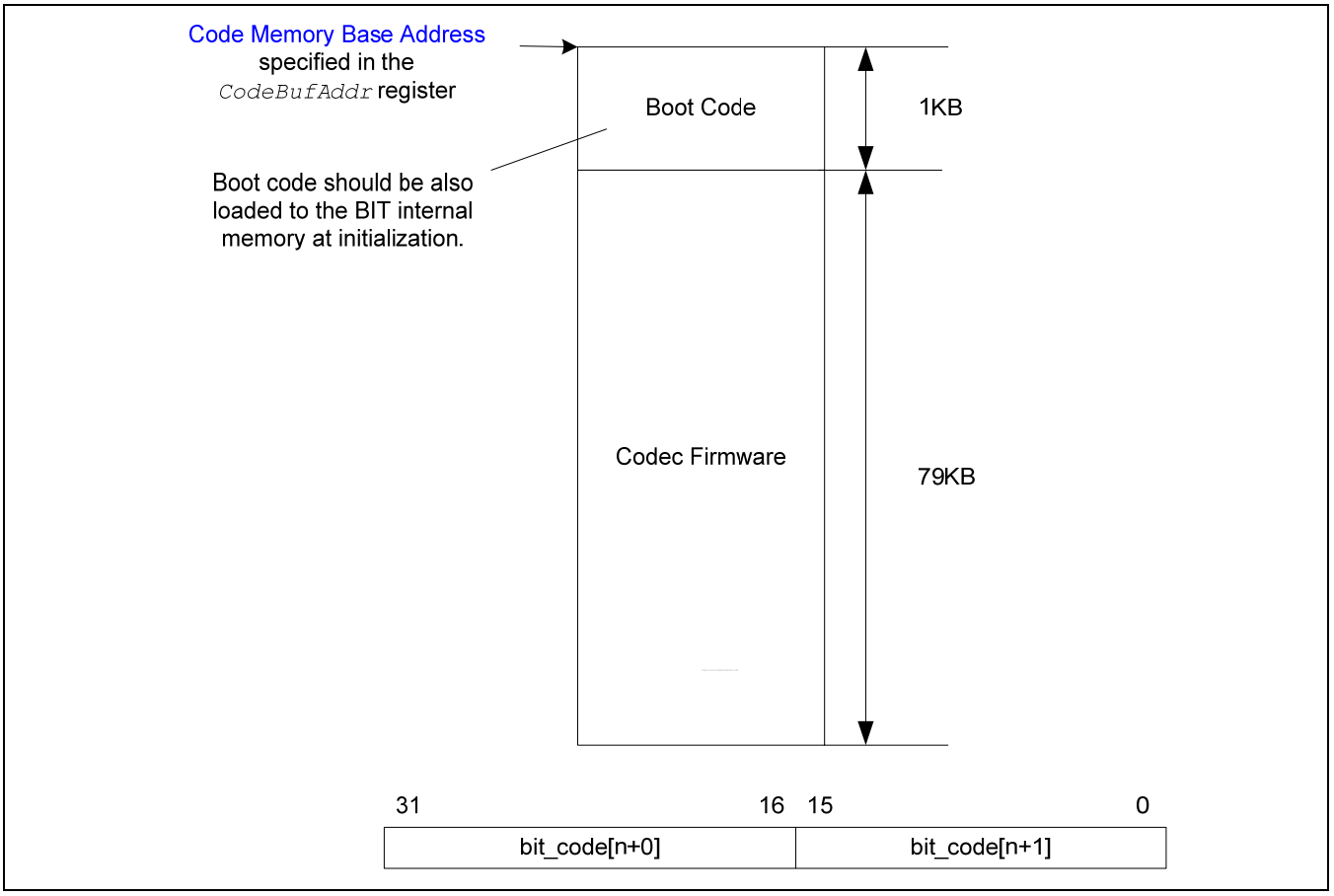


Figure 21-4. Memory space for the BIT firmware

Following is the procedure to download the codec firmware at the video codec initialization step.

```
Step 1. addr = 512;

        sdram_addr = *CodeBufAddr + 1024;

Step 2. code_data = (bit_code[addr] << 16) + bit_code[addr+1]

Step 3. *sdram_addr = code_data;

Step 4. addr = addr + 2;

Step 5. sdram_addr = sdram_addr + 4

Step 6. if (addr < (80*1024)) go to step 1 else finish downloading
```

21.3.3 RUNNING THE CODEC

This section describes how the BIT processor controls the video codec and communicates with a host processor.

The provided firmware can handle 8 processes simultaneously. Each process can have difference format –MPEG-4, H.263P3, H.264 or VC-1 - and codec process-encoding or decoding. For example, it is possible to handle 1 MPEG-4 encoding process, 1 H.264 encoding process, 1 H.263P3 decoding process and 1 VC-1 decoding process simultaneously. You can encode image and/or decode bitstream as following.

- Create processes: You can create and configure processes.
- Running processes: At a proper time instance, you can run a specific process. The proper time instance means when the codec is in idle state and image to be encoded or bitstream to be decoded is readyin the external memory.
- Quit processes: You can quit a specific process.

Figure 21-5 illustrates simplified state diagram for running the codec.

Figure 21-5. Codec firmware state diagram

Process ID - RunIndex

Each process is created with specific ID-named as *RunIndex*- range from 0 to 7. Basically, the ID is assigned based on the order of creation. After creating processes at initialization step, a host processor commands the BIT processor to execute process specified with the RunIndex.

If processes are created sequentially as (a) MPEG-4 encoding, (b) H.264 encoding, (c) H.263P3 decoding for bitstream A, (d) VC-1 decoding for bitstream B, the assigned ID will be as following.

- (a) MPEG-4 encoding : RunIndex = 0
- (b) H.264 encoding : RunIndex = 1
- (c) H.263P3 decoding for bitstream A : RunIndex = 2
- (d) VC-1 decoding for bitstream B : RunIndex = 3

Assigning coding format - RunCodStd

In addition to the process ID, the *RunCodStd* is used to define which coding standard is used in created process and whether the created process encodes image or decodes bitstream.

- RunCodStd = 0: MPEG-4/H.263P3 decoding
- RunCodStd = 1: MPEG-4/H.263P3 encoding
- RunCodStd = 2: H.264 decoding
- RunCodStd = 3: H.264 encoding
- RunCodStd = 4: VC-1 decoding

For example, if processes are created such as the 5.4.1.1, the RunIndex and the RunCodStd for each process is listed below:

- (a) MPEG-4 encoding process: RunIndex = 0, RunCodStd = 1
- (b) H.264 encoding process: RunIndex = 1, RunCodStd = 3
- (c) H.263P3 decoding process: RunIndex = 2, RunCodStd = 0
- (d) VC-1 decoding process: RunIndex = 3, RunCodStd = 4

Codec status

FIMV-MFC V1.0 provides the BusyFlag registers and interrupts that give you information about whether a requested process finished encoding or decoding 1 frame as specified format. When the IP is under operation, the BusyFlag is read as ‘1’. If encoding or decoding 1 frame is finished and the IP is in the wait state that can accept a command from a host processor, the BusyFlag becomes ‘0’. At the time the BusyFlag becomes ‘0’, the interrupt from the IP occurs. A host processor must clear the interrupt before requesting the next process.

If there are some errors in processing a frame, both the BusyFlag and the interrupt are also asserted. A host processor can know whether a process is terminated normally or not by referring to other status registers of the host interface.

21.3.4 INTERRUPT

To a host processor

The BIT processor can generate an interrupt request to a host processor. Basically, this interrupt is used to indicate completion of encoding or decoding 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.

The IREQ is synchronized to the CCLK.

From a host processor

A host processor can request an interrupt to the IP by writing the ‘1’ to the HostIntReq register of the host interface. The interrupt is automatically cleared after acknowledgement of the BIT processor.

21.4 VIDEO CODEC HARDWARE

This section describes the video codec hardware of FIMV-MFC V1.0. All video codec processing except handling coefficients for VLC and VLD are implemented with hardware.

21.4.1 OVERVIEW

21.4.1.1 H.264 encoder data flow

Figure 21-6 highlights the data flow of FIMV-MFC V1.0 H.264 encoding process. The inter-prediction module loads only chrominance data of the reference frame in encoding. The luminance data are read from the local memory of the motion estimation module so that additional bus-loading for the inter-prediction is removed. When the deblocking filter operates in the on-the-fly mode, parts of reconstructed, but not filtered, pixel data are written to external memory for later use in the intra-prediction module. When the deblocking filter operates in the stand-alone mode, a whole reconstructed pixel data in a macroblock is written to external memory, and it is filtered if all macroblocks are reconstructed.

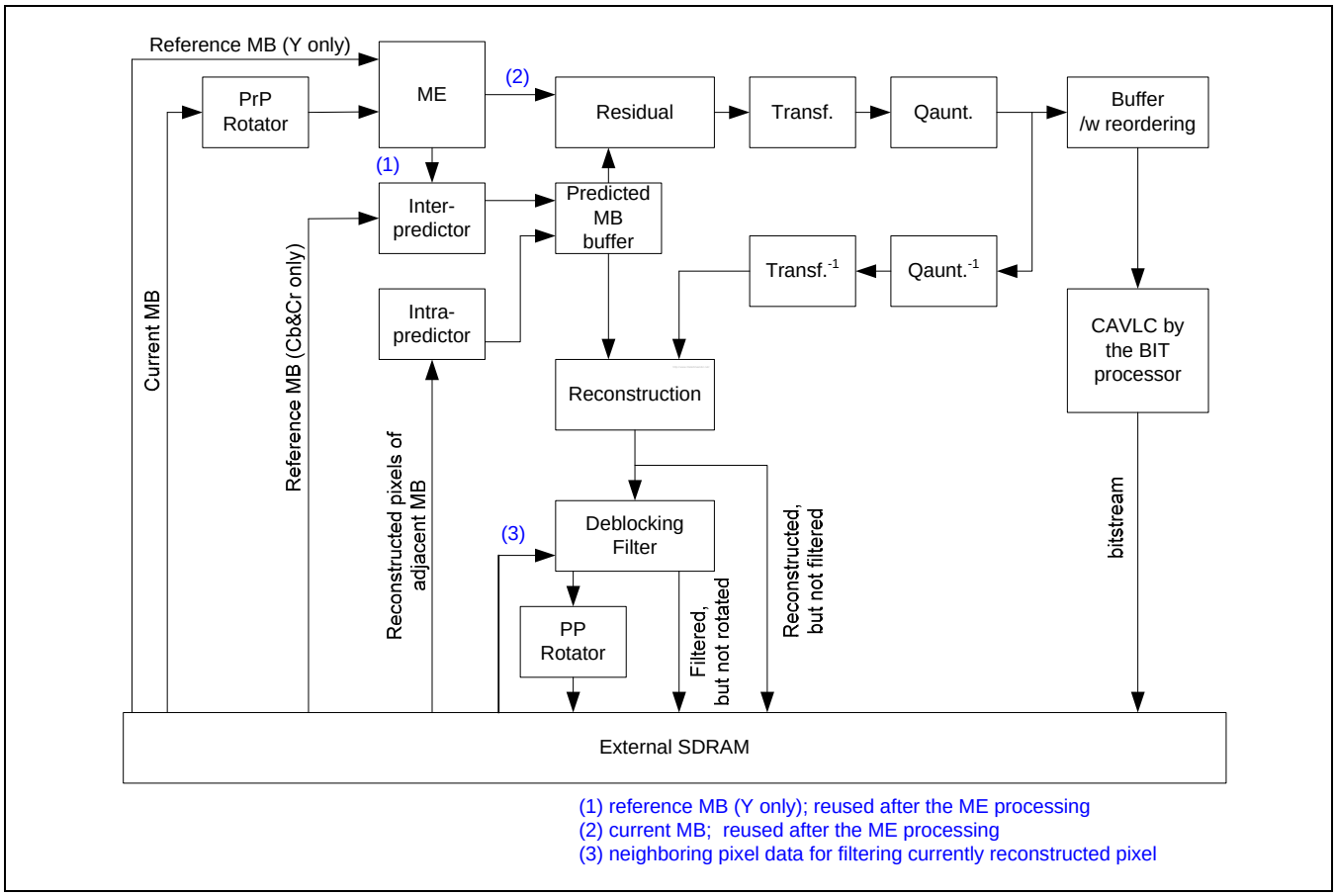


Figure 21-6. H.264 encoder data flow

21.4.1.2 MPEG-4 encoder data flow

Figure 21-7 highlights the data flow of FIMV-MFC V1.0 MPEG-4 encoding process. The data-flow is very similar to the H.264 encoding. The differences are listed below.

- The MPEG-4 encoding process includes the AC/DC prediction instead of the intra-prediction.
- The deblocking filtering is out of the encoding loop.

The bus-loading is a little bit lower than the H.264 because the 1/2-pel fractional sampling needs fewer reference pixel data than the H.264 1/4-pel sampling.

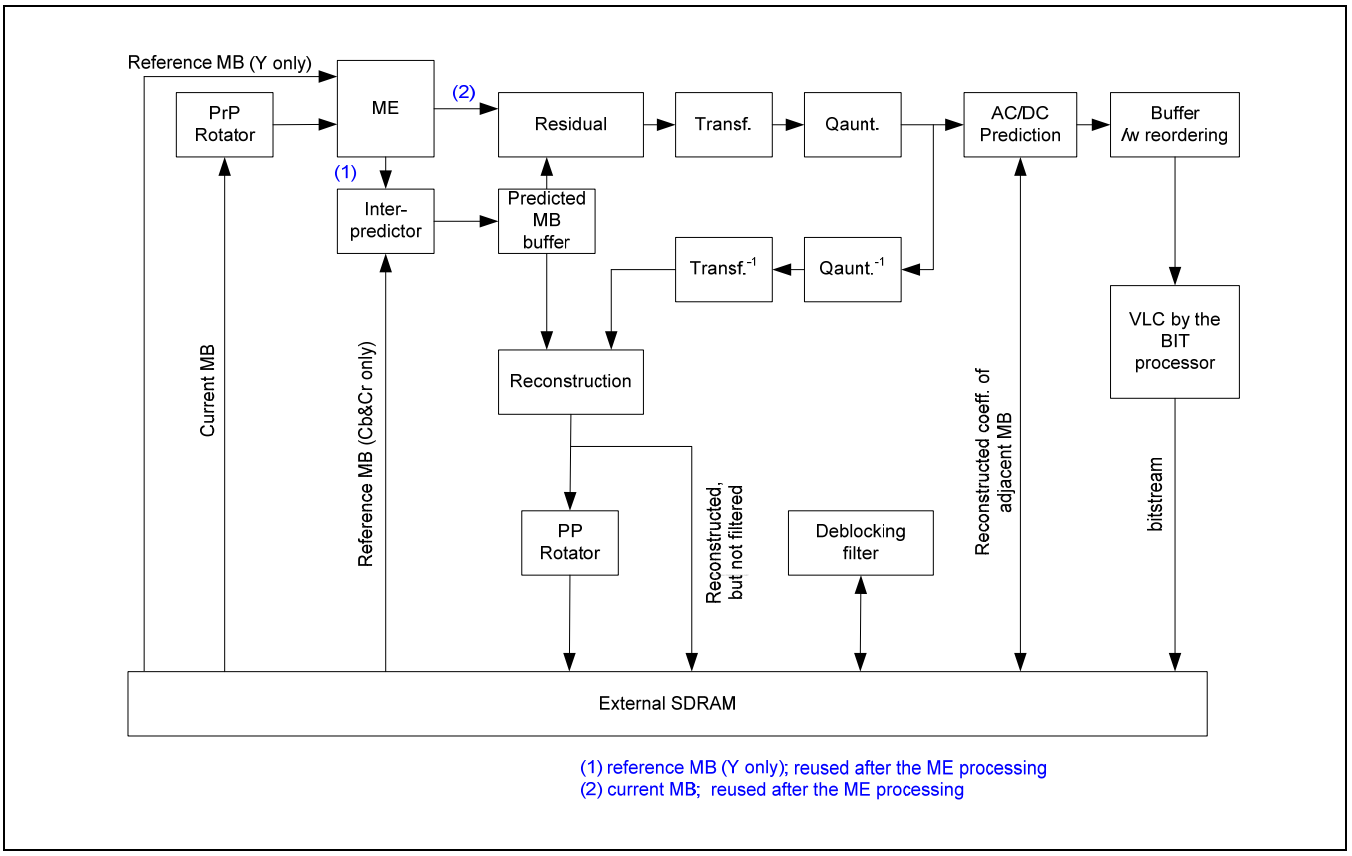


Figure 21-7. MPEG-4 encoder data flow

21.4.1.3 H.263 encoder data flow

The H.263 encoder data flow is the same as the MPEG-4 encoder except following.

- When the Annex I is enabled; the AC/DC prediction process is inserted before the quantization.

NOTE: FIMV-MFC V1.0 does not support the Annex I in encoding process.

- When the Annex J is enabled, the deblocking filtering operates inside the encoding loop.

21.4.1.4 H.264 decoder data flow

For the H.264 decoding process, the decoding data-path in the encoding loop is re-used except following.

- Bitstream is decoded by the BIT processor and decoded coefficients are stored in the coefficient buffer with hard-wired reordering.
- For the reference pixel data, the inter-prediction module reads both Y and Cb/Cr data form external memory. In case of encoding, the data comes from the internal memory of the motion estimation module.

21.4.1.5 MPEG-4 decoder data flow

The data flow of the MPEG-4 decoding process is the same as the decoding in the MPEG-4 encoding loop except incoming coefficients from the BIT processor and Y reference frame directly from external SDRAM, not from internal memory of the motion estimation module.

21.4.1.6 H.263 decoder data flow

The data flow of the H.263 decoding process is the same as the decoding in the H.263 encoding loop except incoming coefficients from the BIT processor and Y reference frame directly from external SDRAM, not from internal memory of the motion estimation module.

21.4.1.7 VC-1 decoder data flow

The data flow of the VC-1 decoding process is the same as the decoding in the H.264 decoding loop except using scaler. If there is MULTIRES flag in the sequence header, the decoded picture is scaled up by scaler.

21.4.1.8 Full-duplex codec processing

FIMV-MFC V1.0 decoding data flow reuses the data-path of the decoding in the encoding loop to reduce logic area. For a full-duplex codec application, time slots are made. For more information refer to the Figure 21-8.

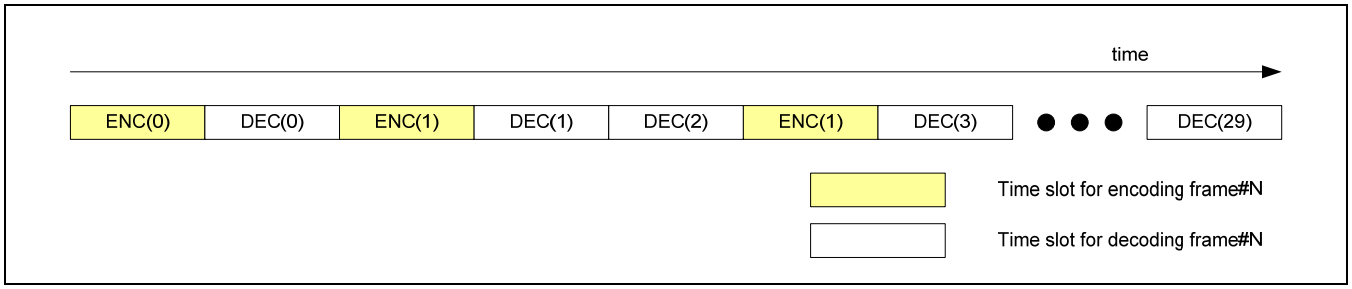


Figure 21-8. Full-duplex codec processing

21.4.1.9 Full-duplex codec bus-loading

The bus-loading for full-duplex is the same as the sum of bus-loadings for encoding and decoding.

21.4.2 FRAME BUFFER

This section describes the memory map of the frame buffer used in FIMV-MFC V1.0 video codec module.

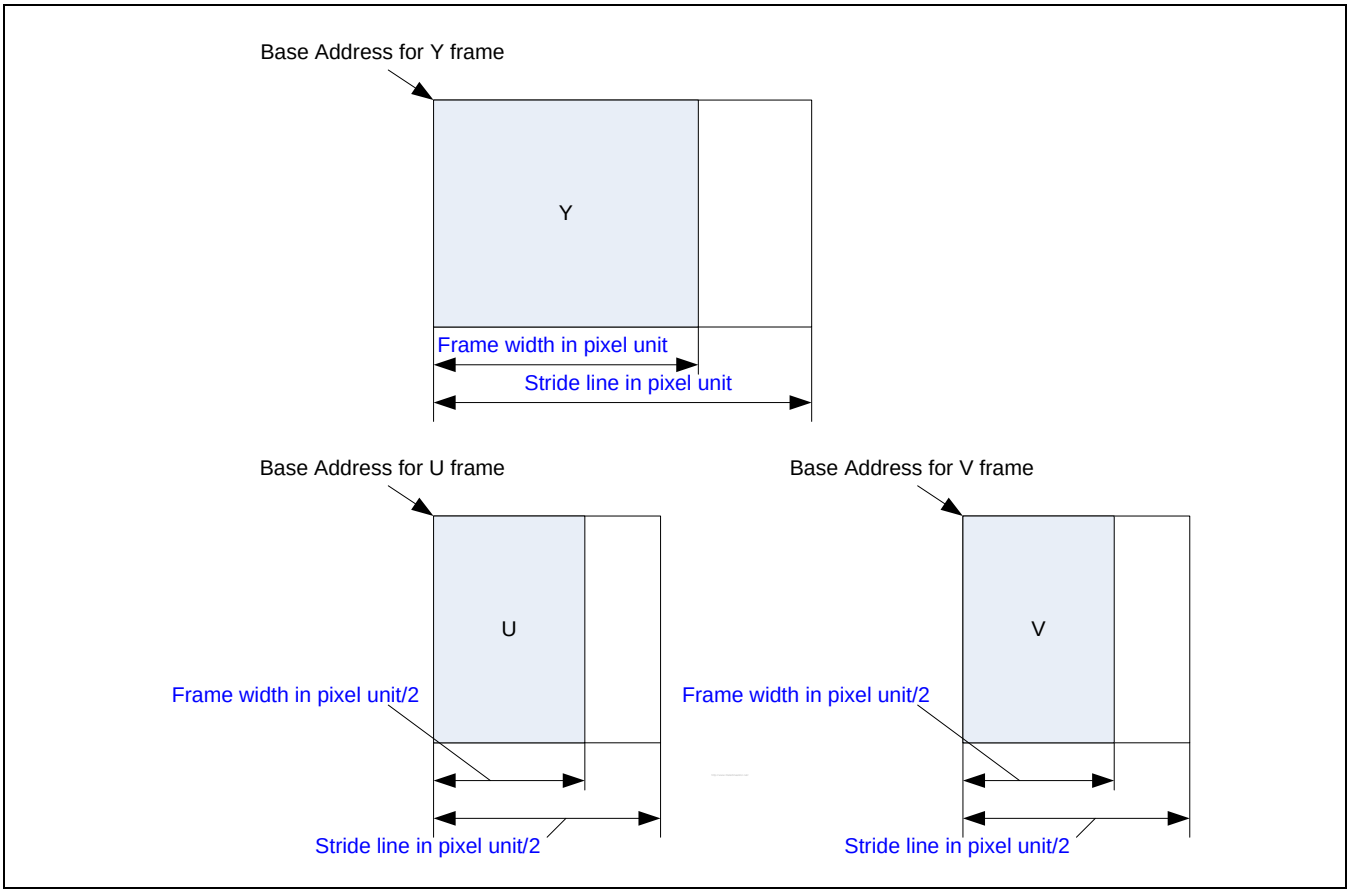


Figure 21-9. Frame buffer configuration

A frame buffer is specified with the base address and the stride line. A complete image consists of Y, U, and V component. Therefore, an image requires 3 frame buffers for Y, U, and V components. The stride line means the width of the luminance component buffer in pixel unit and must be multiple of 8. The stride line for the U and V frame buffers is a half of the Y frame buffer and is extracted automatically based on the stride line of the Y frame buffer. FIMV-MFC V1.0 supports 11-bit stride line.

Figure 21-10 highlights the memory map of the frame buffer. For V frame buffer, the memory map is the same as the U frame buffer except the base address.

FIMV-MFC V1.0 supports both little and big endian system. It means Y(0,0) in the Figure 21-10 could be located in the bit[31:24]. A user can specify the endian to the register of the host interface.

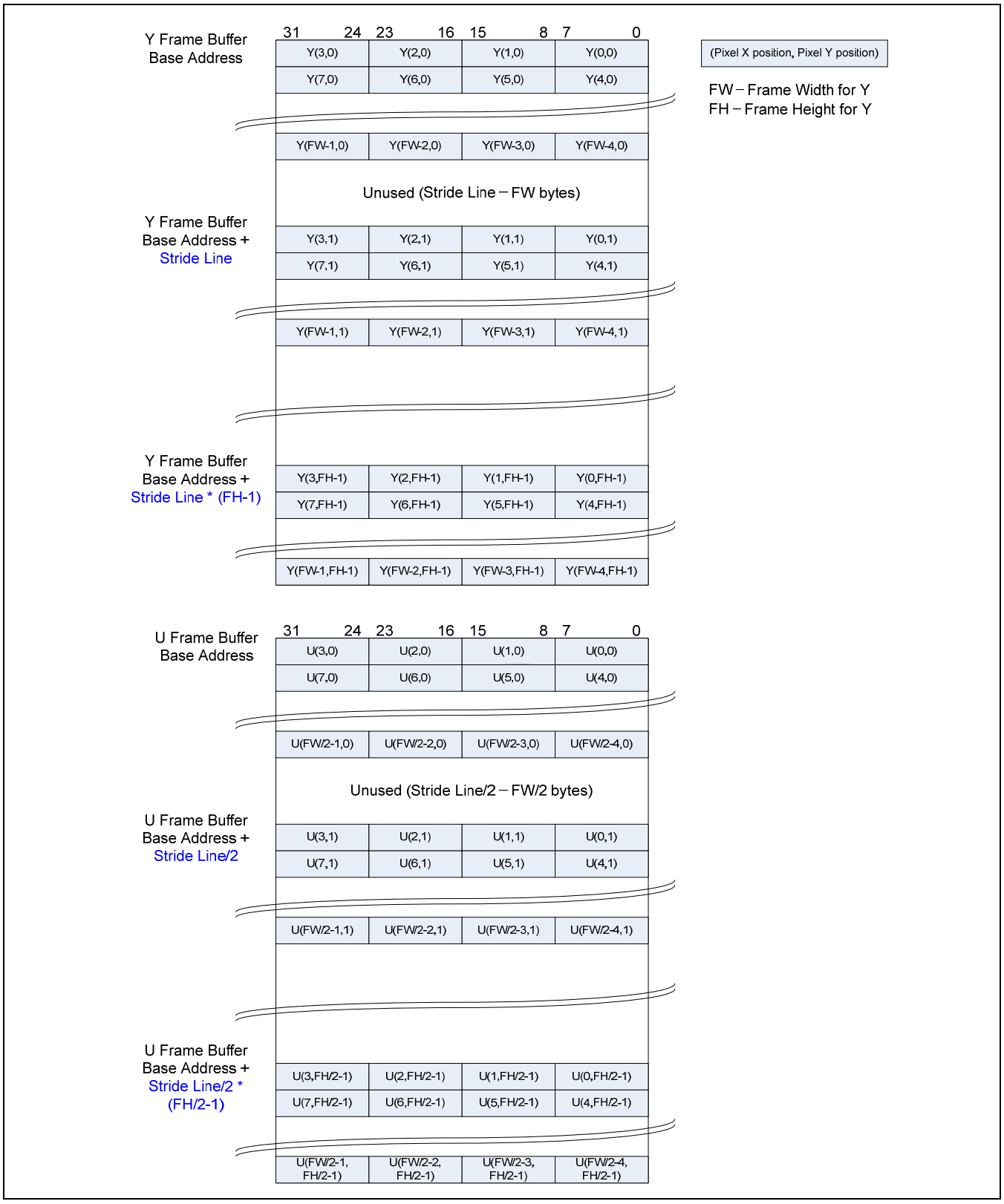


Figure 21-10. Frame buffer address map in little endian

For encoding case, typically, 4 frame buffers are required. These buffers are used for storing incoming image from camera or pre-processor, encoding, storing the currently reconstructed image, and previously reconstructed frame.

21.4.3 ROTATION/MIRRORING

FIMV-MFC V1.0 supports rotation together with mirroring function for both incoming image for encoding and output image of the decoder for display. The former is done by the PrP rotator module and the latter by the PP rotator module.

21.4.3.1 PrP rotator module

FIMV-MFC V1.0 uses output from the rotator module as input to the encoder without additional bandwidth consumption on the external SDRAM for rotation itself. The rotated image is sent to the local buffer of the motion estimation module, and re-used in the intra-prediction module for intra-mode decision and residual computation module. The PrP rotator module does not work in decoding process.

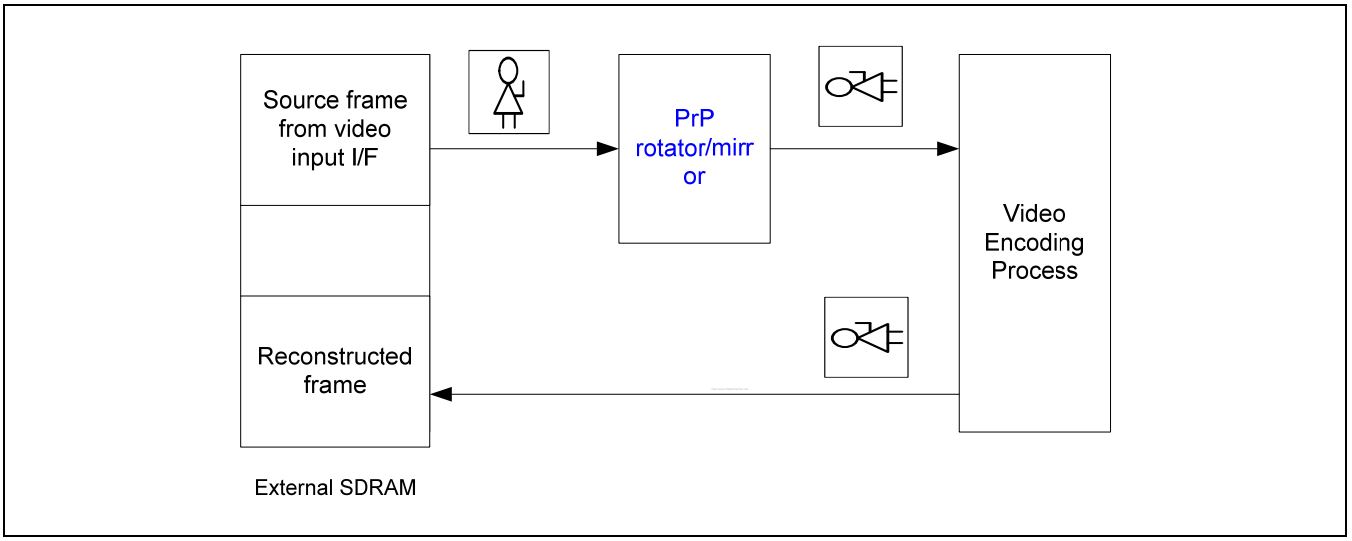


Figure 21-11. PrP rotator data flow

21.4.3.2 PP rotator module

The rotation/mirroring process in decoding process requires additional bandwidth because the video codec IP has to re-use the un-rotated image for decoding the next image. Therefore, the rotated image is written to other memory space. In this scheme, the display I/F has not to change memory space for displaying the decoded image because subsequent rotated image is written to the same space. Of course, change of target frame buffer is possible by setting registers that specify its base addresses. The PP rotator module does not work in encoding process.

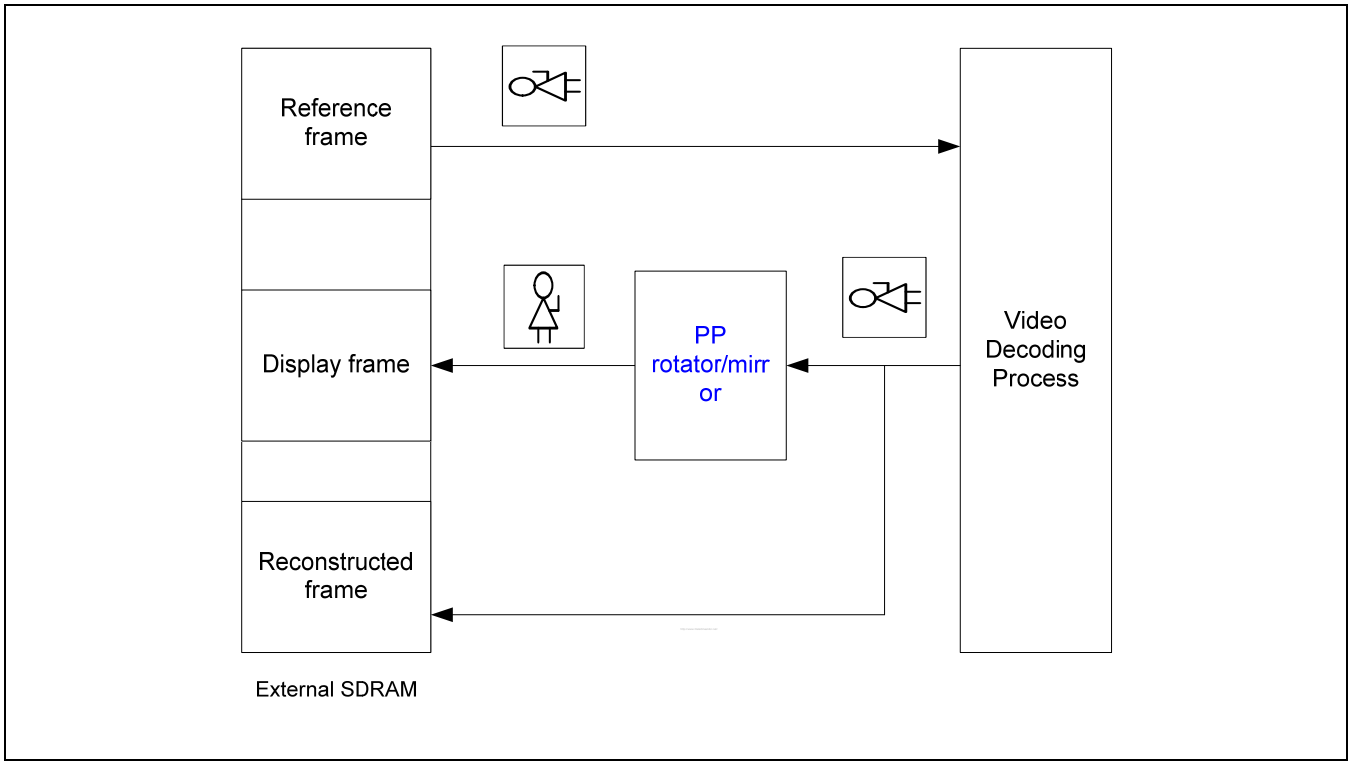


Figure 21-12. PP rotator data flow

21.4.3.3 Rotation/mirroring modes

The Rotator modules support 8-types mode of 90 x n degree(n=0, 1, 2, 3) rotating and mirroring simultaneously. The following table is the supporting rotating /mirroring lists and these represents all possible combinations of rotating and mirroring. There are two register sets. One is for the PrP module and another for PP module.

Table 21-1. Rotation/mirroring modes

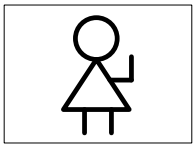
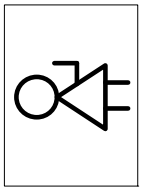
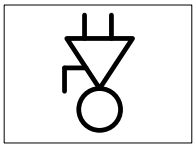
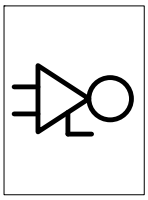
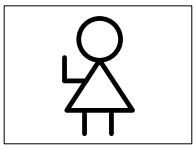
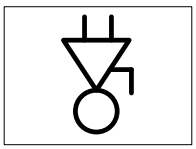
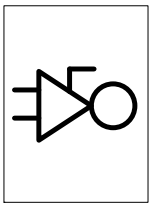
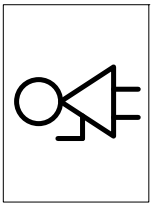
MODE	Rotated/mirrored Image	Descriptions
NONE_ROTATE		Original Image(No ratoting/mirroring) Example Image Size : 720x480
ROT_LEFT_90		Rotate Left 90 (Rotate Right 270) Example Image Size : 480x720
ROT_LEF_180		Rotate Left 180 (Rotate Right180) Example Image Size : 720x480
ROT_LEFT_270		Rotate Left 270 (Rotate Right 90) Example Image Size : 480x720
MIR_HORIZ		Horizontal mirroring Example Image Size : 720x480
MIR_VERT		Vertical mirroring Example Image Size : 720x480
MIR_HOR_ROT_RIGHT_90		Horizontal mirroring and rotate right 90 Example Image Size : 480x720
MIR_HOR_ROT_LEFT_90		Horizontal mirroring and rotate left 90 Example Image Size : 480x720

Table 21-1 highlights rotation/mirroring modes. In above example, input to all rotation/mirroring modes is the same as the output image of the NONE_ROTATE mode. A host processor can select one of those modes by setting dedicated register. Practically, you call an API to configure the rotation mode and the information is sent to the BIT processor. The BIT processor sets a register of the rotation/mirroring module that specify the mode.

21.4.4 MOTION ESTIMATION

The Motion Estimation Block uses full-search algorithm, and the search range is +- 16 pixel or +- 8 pixel. The following features are supported:

- UMV (Unrestricted Motion Vector) mode
- Up to quarter-pel search for H.264-BP
- Up to half-pel search for MPEG4-SP:
- Support 16x16/8x8 block for MPEG4-SP
- Support 16x16/16x8/8x16/8x8 block for H.264-BP
- Full-search algorithm

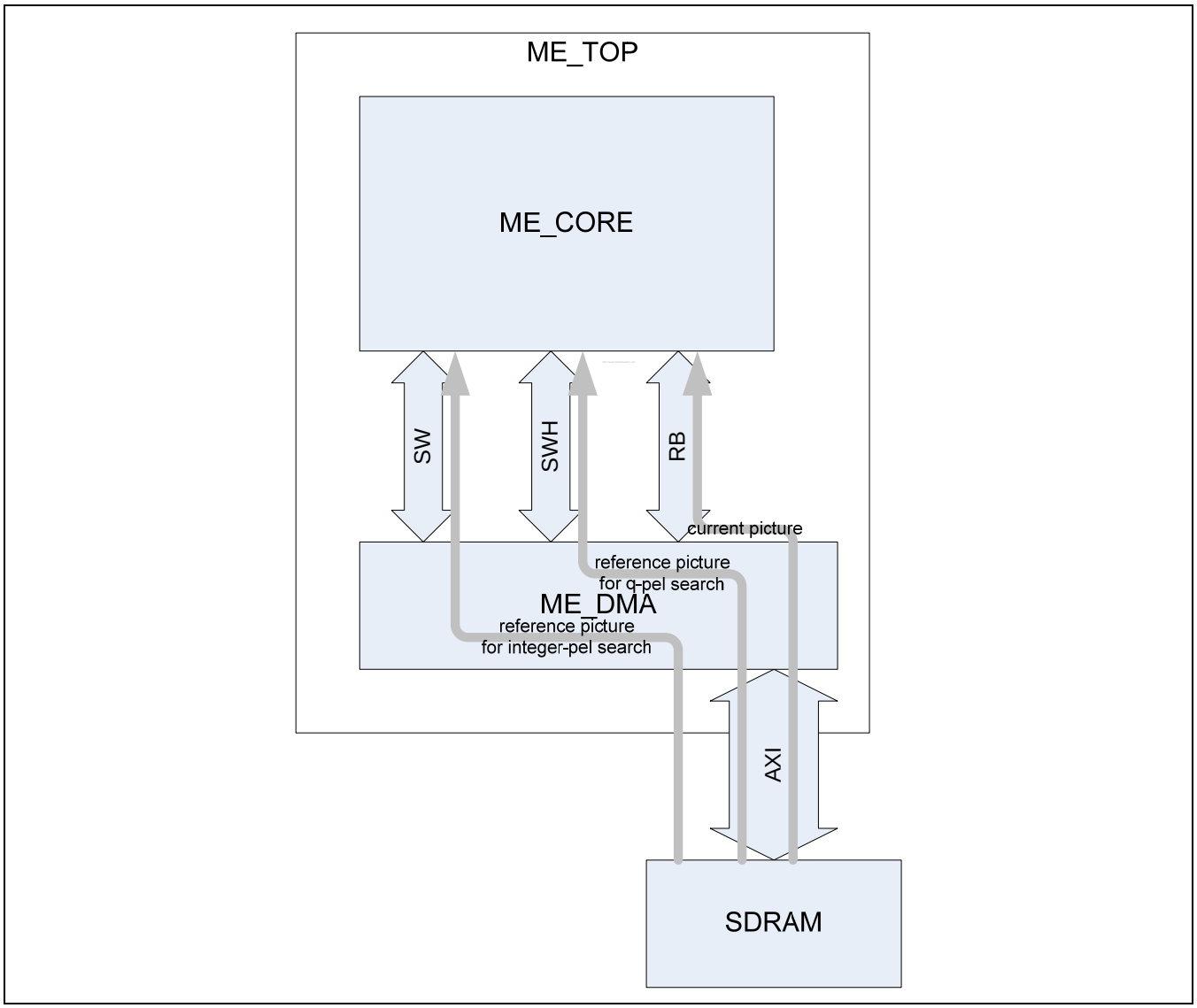


Figure 21-13. Motion Estimation Block Diagram

The ME block can impose high-priority on zero motion vector by subtracting calculated SAD by user-defined register value. The prediction block size (16x16, 16x8, 8x16, 8x8) decision can be prioritized in similar ways. There are 3 offset registers for 8x8 block, 8x16 block and 16x8 block. You can impose high-priority on large block size by adding calculated SAD by user-defined registers.

The above figure highlights the Motion Estimation Block top diagram.

The Motion Estimation Block reads current frame and reference frame. ME_DMA handles all memory operations. ME_CORE requests to ME_DMA for current and reference frame images, and ME_DMA reads the data and send it to ME_CORE.

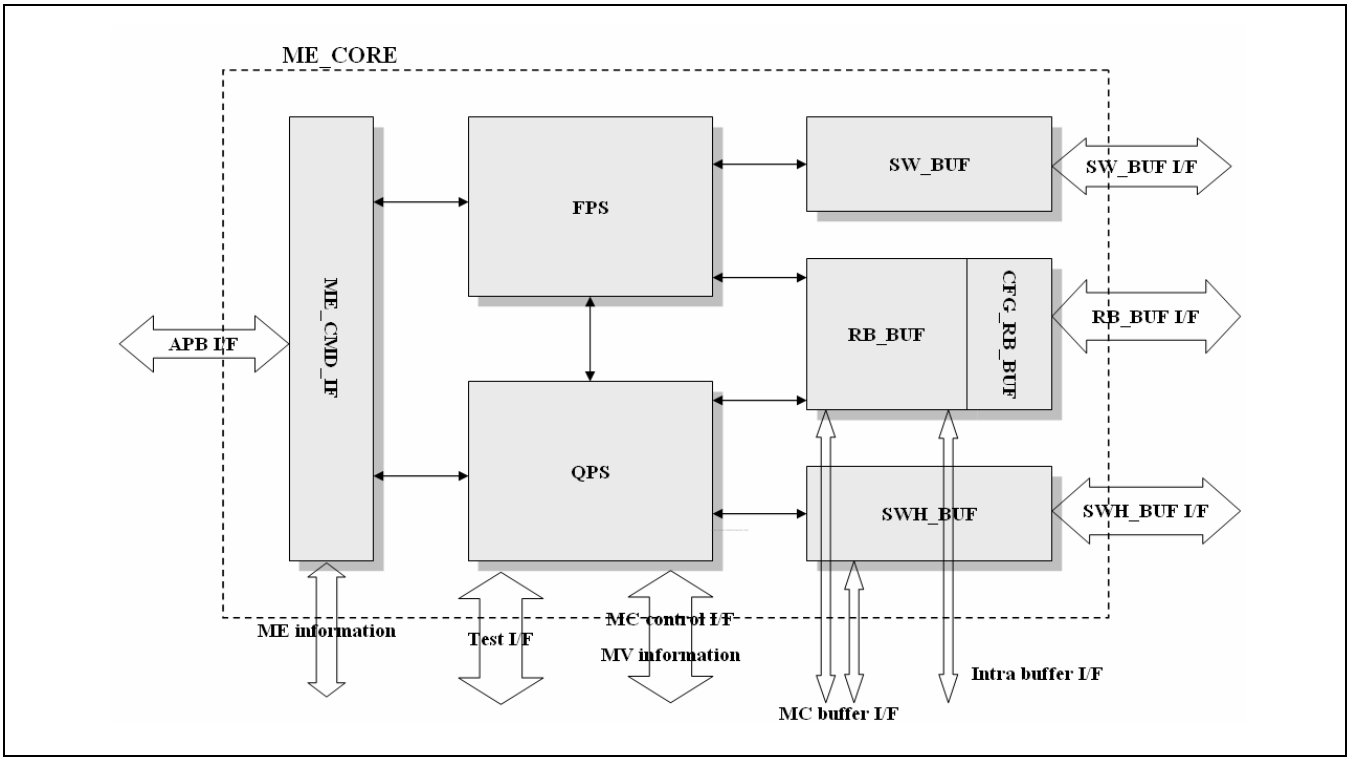


Figure 21-14. Motion Estimation Core Block Diagram

ME_CORE consist of command interface (ME_CMD_IF), integer-pel search block (FPS), quarter/half-pel search block (QPS) and internal buffers(SW_BUF, RB_BUF, SWH_BUF).

ME_CMD_IF block interfaces with BIT processor or CPU. Current frame image is stored in RB_BUF, and reference frame image is stored in SW_BUF and SWH_BUF. Interger-pel search block (FPS) uses RB_BUF for current frame and SW_BUF for reference frame. Quarter/half-pel search block (QPS) uses RB_BUF for current frame and SWH_BUF for reference frame.

21.4.4.1 ME DMA block

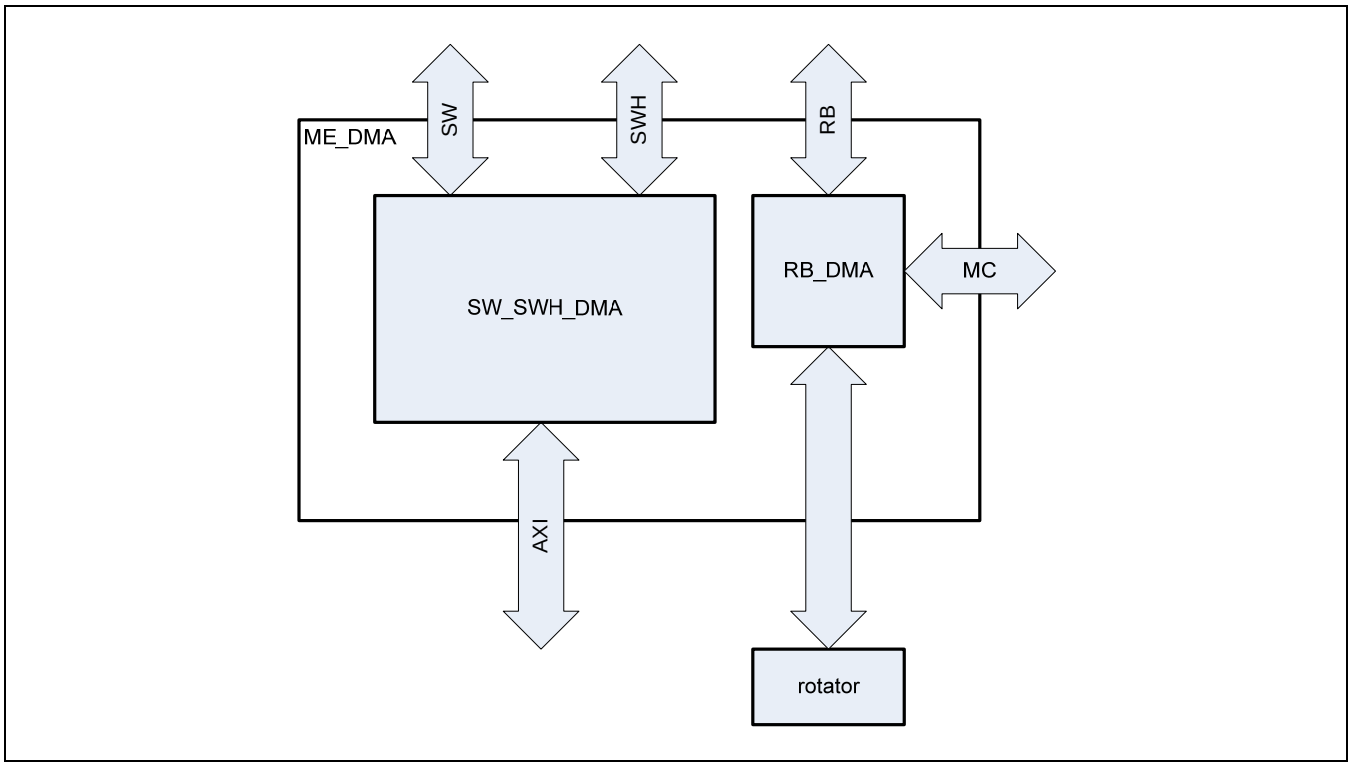


Figure 21-15. Motion Estimation DMA Block Diagram

ME_DMA services current and reference frame images to ME_CORE.

SW_DMA and SWH_DMA is reference search window data request channel. SW_DMA and SWH_DMA read data from SDRAM. RB_DMA reads current frame data through Pre-rotator block.

21.4.5 INTER-PREDICTION

The Inter-Predictor uses reconstructed motion vector that represents the displacement between the block currently being decoded and the corresponding location in the reference frame, to calculate interpolated pixel data for motion compensation.

The Inter-Predictor consists of main controller, interpolator, DMA and local memory. The role of each sub-block is listed below:

Main controller	It is controlled by the BIT processor that writes control information (motion vector, block mode, reference picture index, picture size and run command etc.) into the control register in main controller. It includes data register for motion vector and reference picture base address .
Interpolator	It interpolates reference picture data with 2-pel/4-pel resolution. It uses temporal memory for calculating H.264 quarter-pels.
DMA	Reads reference picture data from SDRAM(decoding) or ME local memory(encoding) and write it into local buffer memory

Inter-predictor supports all H.264 block mode (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4), half/quarter pixel resolution, 16 reference frame in H.264 BP and padding pixel when the motion vector points to an area that is outside the image.

The figure 21-16 is the block diagram of Inter-predictor.

When the BIT processor set the registers (motion vectors, reference frame index, macroblock mode etc) of main controller, main controller controls DMA and interpolator.

The DMA read reference pixel data from SDRAM in decoding and from ME local memory in encoding to reduce SDRAM bandwidth. If the motion vectors used to locate the predicted blocks in the reference frame may include pixel locations that are outside the boundary of the reference frame in decoding mode. In these cases, the out-of-bounds pixel values are the replicated values of the edge pixel by DMA.

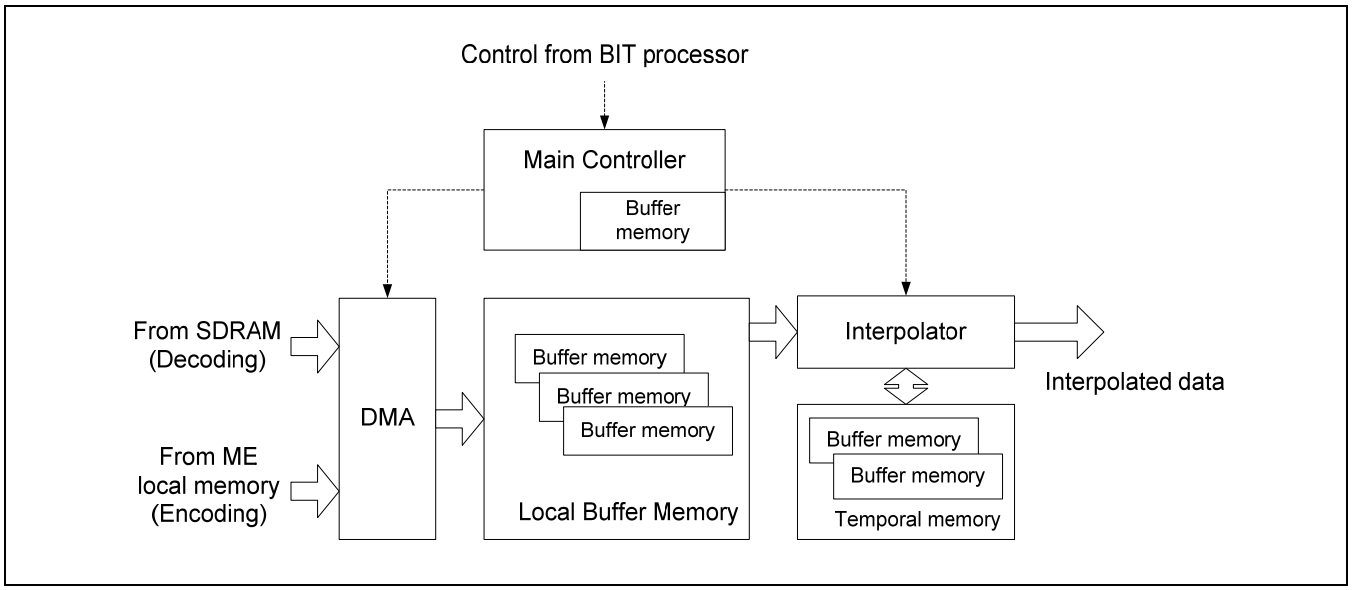


Figure 21-16. The Block Diagram of Inter-prediction

The interpolator block of inter-predictor calculates half or quarter pixel using reference pixel data in the local buffer memory stored by DMA. The interpolator uses the temporal memory to interpolate pixel in (1/4, 1/4), (1/4, 1/2), (1/2, 1/4) position indicated by reconstructed motion vector. The result interpolated pixel is written in the buffer to add the error residual and current pixel.

Followings features are supported by inter-prediction module.

- MPEG-4/H.263P3
  - UMC(unrestricted motion compensation)
  - 4MV (Both 16x16 and 8x8 block sizes are supported.)
  - 1/2-pel motion compensation
- H.264
  - 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4
  - 16 reference frames
  - 1/4-pel motion compensation

21.4.6 INTRA-PREDICTION

FIMV-MFC V1.0 includes two intra-prediction modules. One is for the MPEG-4/H.263P3 AC/DC prediction and the other for the H.264 intra-prediction.

Figure 21-17 highlights the data flow of intra-prediction in the MPEG-4 and the H.264. In case of H.263P3 AIC (Advanced Intra Coding) mode, the AC/DC prediction is performed for the transformed coefficient, not for quantized coefficient.

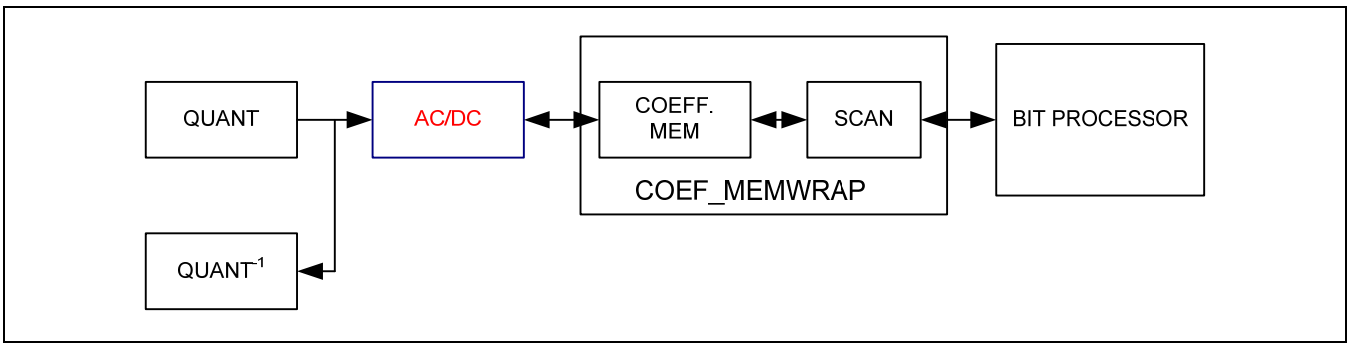


Figure 21-17 (a). MPEG-4 AC/DC prediction data flow

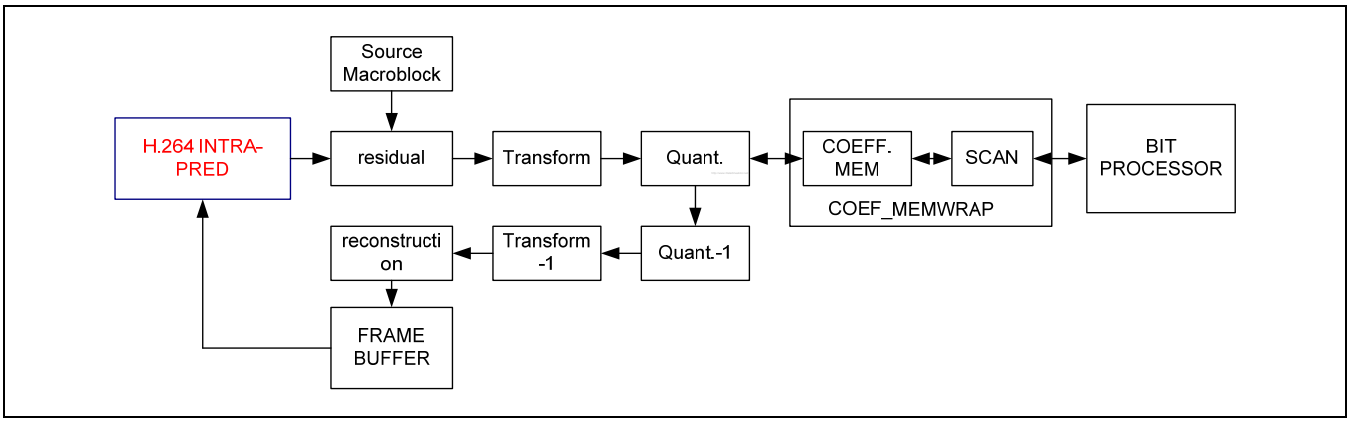


Figure 21-17 (b). H.264 Intra prediction data flow

Large parts of modules are shared in both encoding and decoding. This module loads and stores the neighboring pixel or coefficients through the DMA controller interface with the AMBA AXI bus.

An internal buffer is shared between both AC/DC prediction and intra-prediction module. It has dual-port because the prediction module is operated in the core clock and the bus interface in the bus clock, which is asynchronous to the core clock.

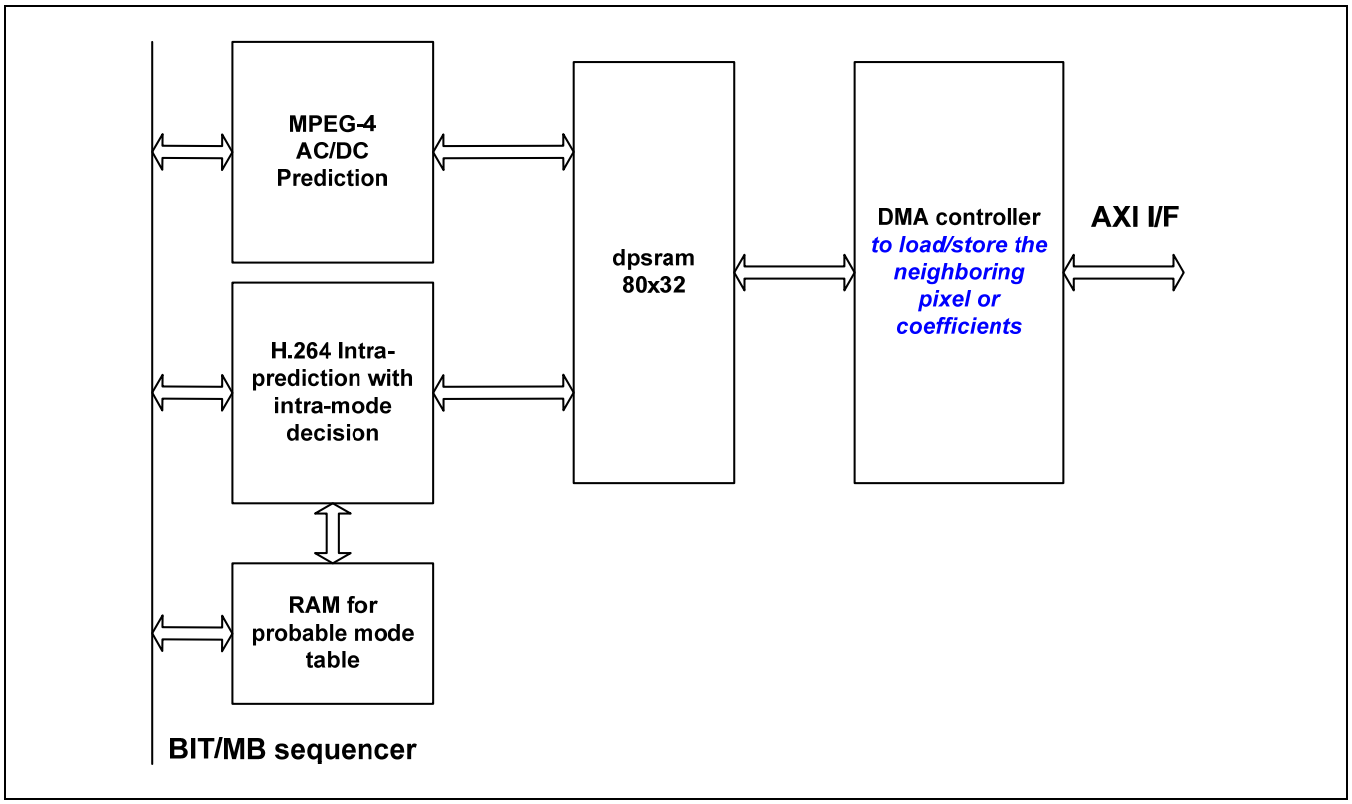


Figure 21-18. Intra-prediction block diagram

In case of MPEG-4 encoding, the hardwired prediction mode decision is used. It brings high performance and low power consumption. The coefficient data of both encoding and decoding is re-ordered automatically based on the detected prediction mode.

For the H.264 intra-prediction mode in encoding, hardwired mode decision or software-based mode decision is used. In case of the hardwired mode decision, a probable mode table is downloaded by the BIT processor. This table is used for the intra-prediction mode decision logic to search the best mode in some candidates. The number of candidates is configurable. For maximum quality of the encoding, all possible prediction modes could be estimated.

The availabilities of the neighboring macroblock or block are managed by the BIT processor. The availability is determined by the slice, macroblock type, and location of current macroblock (or block).

21.4.7 TRANSFORM/QUANTIZATION

FIMV-MFC V1.0 has two transform/quantization modules. One is for the MPEG-4/H.263P3, the other for the H.264.

FIMV-MFC V1.0 H.264 T/Q block processes transform and quantization (or inverse transform and inverse quantization) of residual data. T/Q block compresses the residual data and send to other blocks (Coefficient buffer, Motion compensation block).

21.4.7.1 Overview

The following picture highlights the encoding and decoding flow of FIMV-MFC V1.0 H.264 T/Q. chroma_dc block shared in encoding and decoding process. The trans and itrans block processes not only residual transform, but also 4x4 luma dc transform.

In case of encoding, the quantized coefficients are written to the coefficient buffer, and the BIT processor reads them that are reordered in the coefficient buffer interface module. In the same time, the quantized coefficients are processed in the decoding loop of the encoding process.

For decoding processor, the T/Q module reads coefficients that are decoded by the BIT processor and reordered in the coefficient buffer interface.

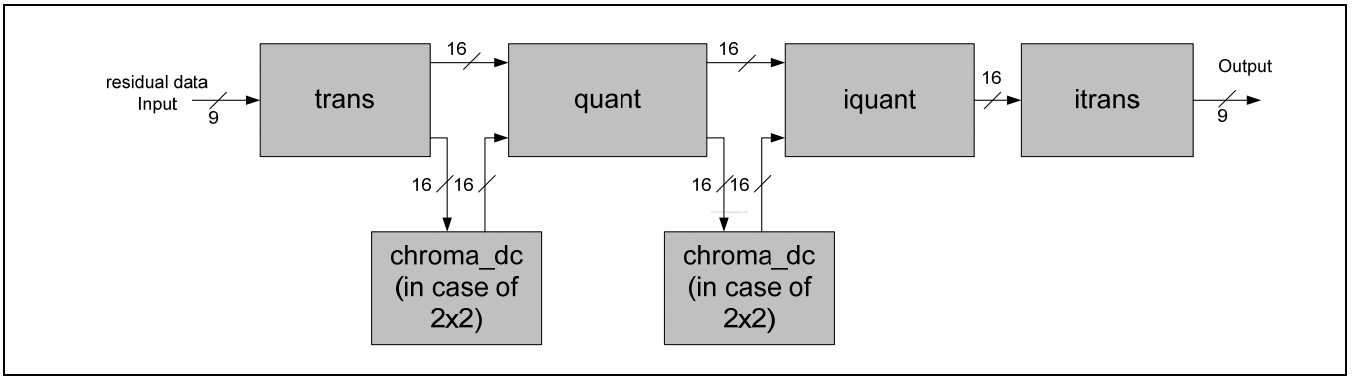


Figure 21-19. H.264 transform/quantization data flow

21.4.7.2 Block diagram

Figure 21-20 displays block diagram of FIMV-MFC V1.0 H.264 T/Q block.

TRANS block sends transform coefficient data to QUANT block by TQ_CTRL block. TRANS block can process residual 4x4 block transform and Hadamard transform (in case of luminance). In case Hadamard transform of chrominance, CHROMA_DC block processes chroma DC transform. CHROMA_DC processes both forward and inverse chroma DC transform.

QUANT block transfer the quantization result of coefficient data to coefficient memory or IQUANT block.

TQ_CTRL controls encode and decode processing of the H.264 T/Q block.

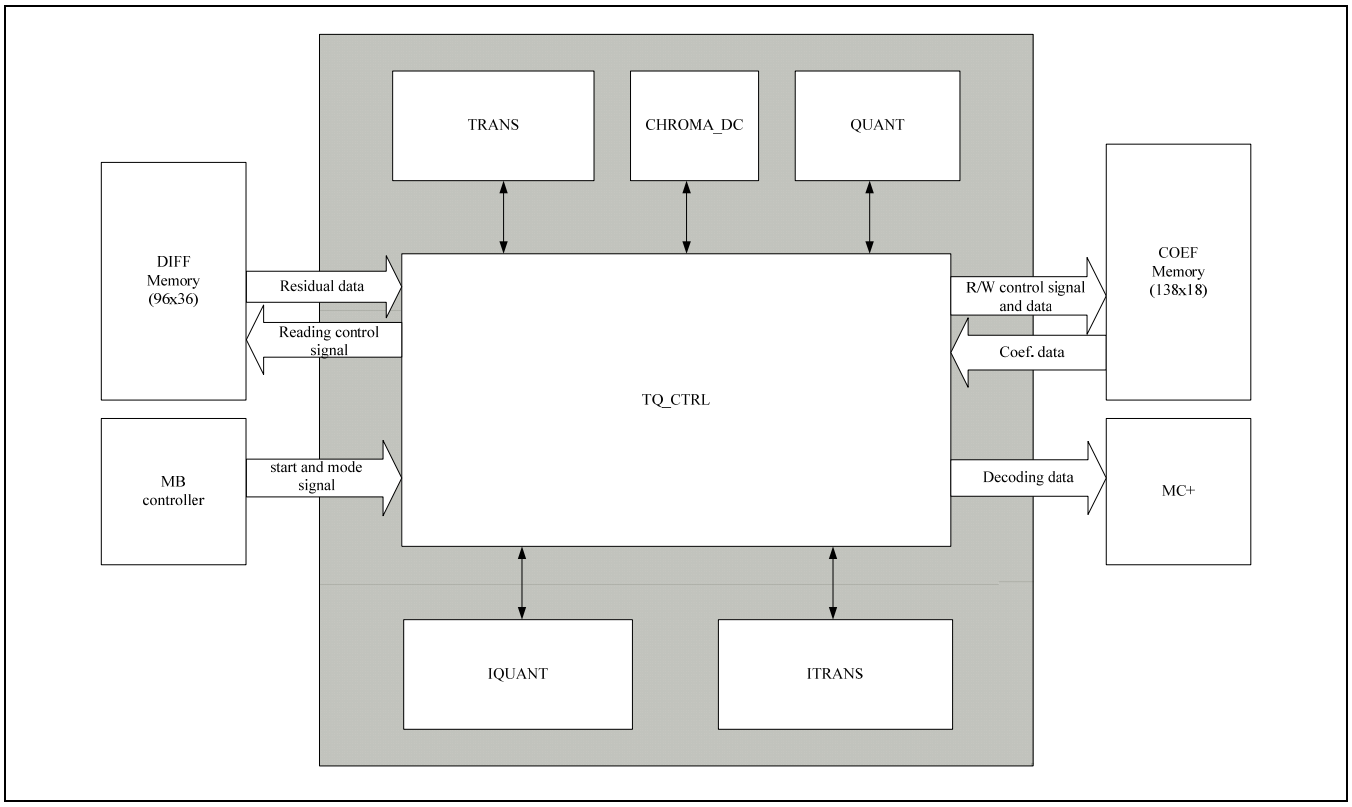


Figure 21-20. H.264 tranform/quantization block diagram

21.4.7.3 MPEG-4/H.263

The MPEG-4/H.263P3 transform/quantization module supports only the method1 quantization mode for MPEG-4 bitstream. It can process the AIC (Advanced Intra Coding) and the modified quantization mode for H.263P3 bitstream.

To process one macroblock in the MPEG-4/H.263P3, FIMV-MFC V1.0 transform/quantization requires about 500 cycles.

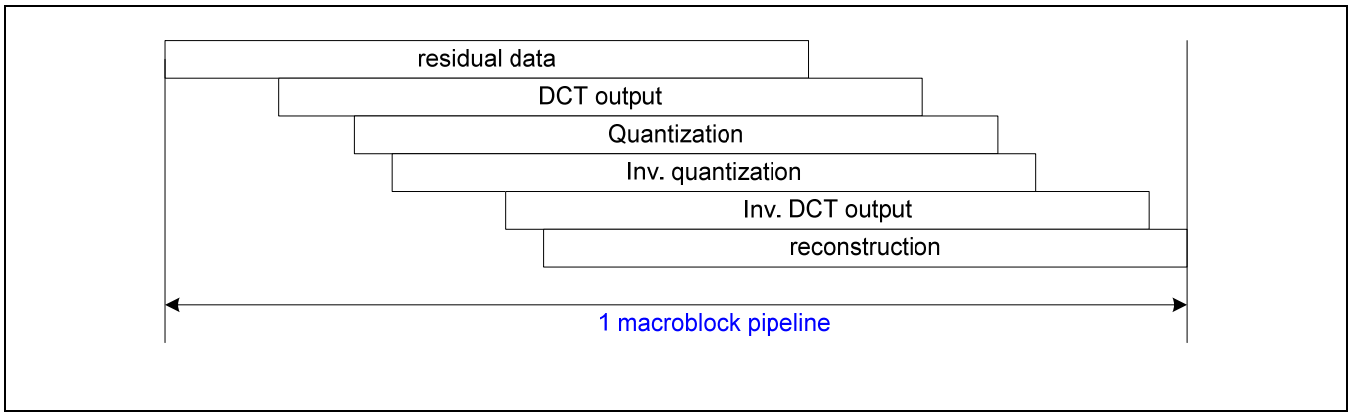


Figure 21-21. MPEG-4/H.263 transform/quantization process

Figure 21-21 highlights the MPEG-4/H.263 transform/quantization process in a macroblock pipeline. In the encoding process, the quantized coefficients are sent to the inverse quantization module and the coefficient buffer interface at the same time. The coefficients in the coefficient buffer are processed again in the AC/DC prediction module.

NOTE:

The H.263 annex I-advanced intra-coding mode-is supported for the H.263P3 decoding. It is not supported in the encoding process.

121.4.7.4 H.264

There are many macroblock types in the H.264 standard such as INTRA_4X4, INTRA_16X16, inter macroblocks with various block sizes, and I_PCM. The I_PCM macroblock does not have any transform and quantization on its pixel data. Therefore, in case of I_PCM macroblock, the T/Q module by-passes incoming data to the coefficient buffer memory and the inverse quantization module without any processing. In case of inter-macroblock and INTRA_16X16 macroblock, the operation is very similar except transformation of the DC coefficient is performed for INTRA_16X16. For the H.264 codec, the most complex macroblock type in view point of performance and control is the INTRA_4X4. Except the INTRA_4X4, all processing such as transform and quantization can be pipelined efficiently. However, in case of encoding INTRA_4X4-type macroblock, before transform of current 4x4 block, reconstruction of previous 4x4 block has to be finished. This is because the previously reconstructed 4x4 block is used to do intra-prediction of current 4x4 block that is input to residual computation module.

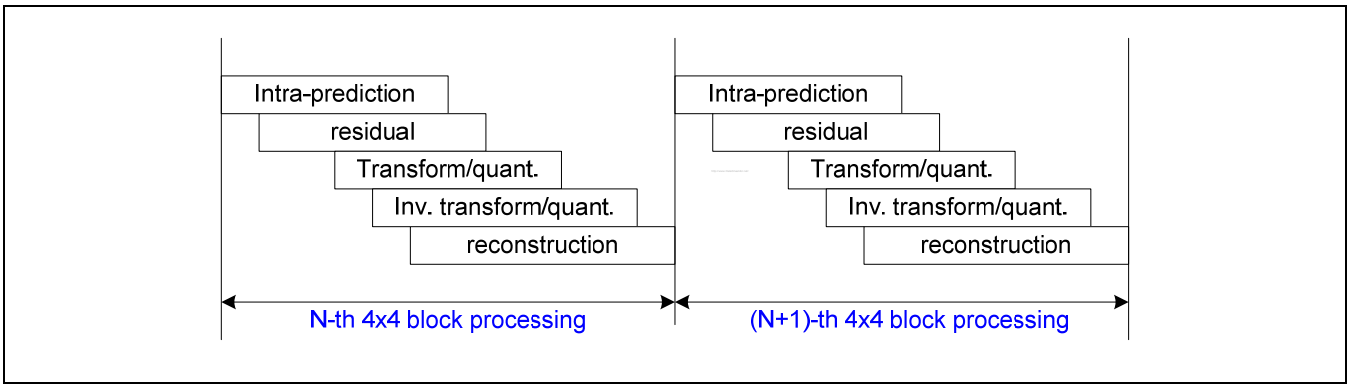


Figure 21-22 H.264 encoding pipeline

21.4.8 OVERLAP-SMOOTHING/DEBLOCKING FILTER

21.4.8.1 Overview

Deblocking filter removes blocking artifacts resulted from quantization, different motion vectors. The filter processing is applied in both decoder and encoder. FIMV-MFC V1.0 deblocking filter supports H.264/H.263/MPEG4. For H.264 and H263, the deblocking filter operates within coding loop. Filtered frames are used as reference frames for motion compensation of subsequent coded frames. But for MPEG4, the deblocking filter operates outside coding loop for only display.

The basic coding structure of H.264 is shown in Figure 21-23.

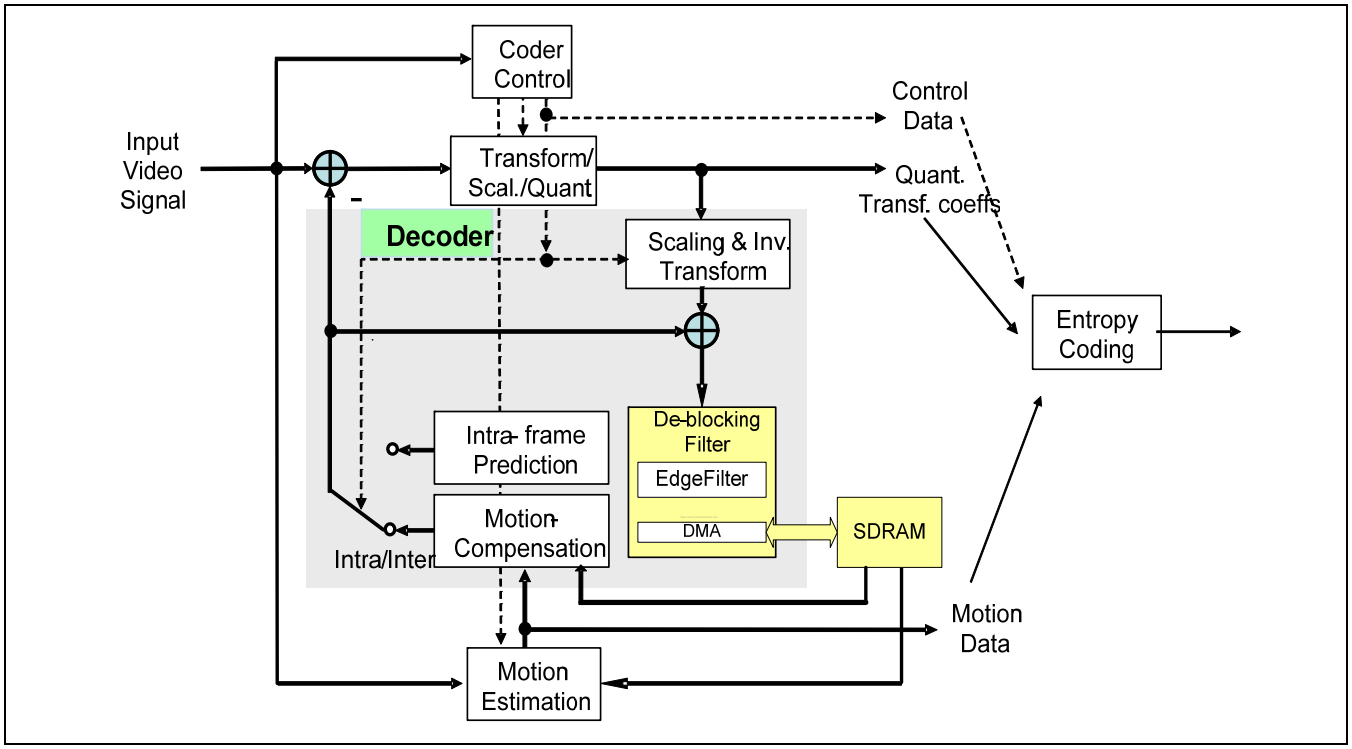


Figure 21-23. H.264 deblocking filter data flow

21.4.8.2 Processing modes

FIMV-MFC V1.0 supports 2 operating modes for each standard regarding to whether the filtering process is applied to the reconstructed image in the macroblock pipeline or not.

On-the-fly mode

In the on-the-fly mode, the macroblock output from the reconstruction is immediately filtered without saving the entire reconstructed output to external SDRAM. The DMA controller transfer only part of the reconstructed macroblock that cannot be filtered due to absence of reconstruction pixel data of neighboring macroblock. This mode improves the bandwidth efficiency.

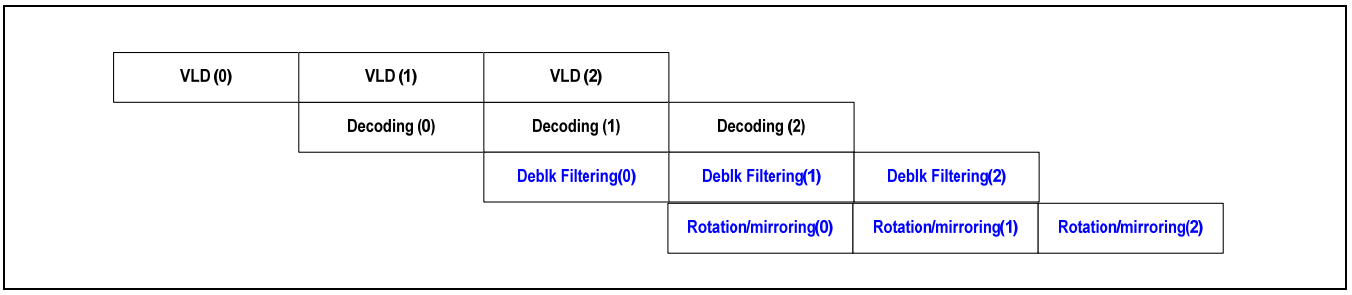


Figure 21-24. The pipeline structure of the decoding case(for H.263 and H.264)

Figure 21-24 illustrates the pipeline structure of the decoding case that is applied to both the H.263 and the H.264. The number between brackets means the macroblock address is being processed.

Stand-alone mode

Another mode, called as ‘standalone mode’, performs filtering process after making reconstructed frame, not macroblock. In the stand-alone mode, the deblocking filter module reads an entire frame decoded already, and writes it to specified frame buffer after filtering. It needs more bandwidth of external memory bus compared to the on-the-fly mode. In the on-the-fly mode, the input to the filter is from a buffer in the codec, not from external memory.

While stand-alone deblocking filtering, other modules of the codec are in idle state. Therefore, total time for processing complete 1 frame is longer than the on-the-fly mode.

The filtered frame can be stored to a specified frame buffer or overwritten to the source frame buffer where the input frame to the filter exists.

21.4.8.3 Block diagram

The figure 21-25 highlights the overlap/deblocking filter architecture. The filtered pixel data are stored in working buffer. The output data is store to output buffer. The rotator/mirror block will read this output buffer. The input buffer and DMA buffer is used to store/load the intermediate data for processing deblock/overlap filter.

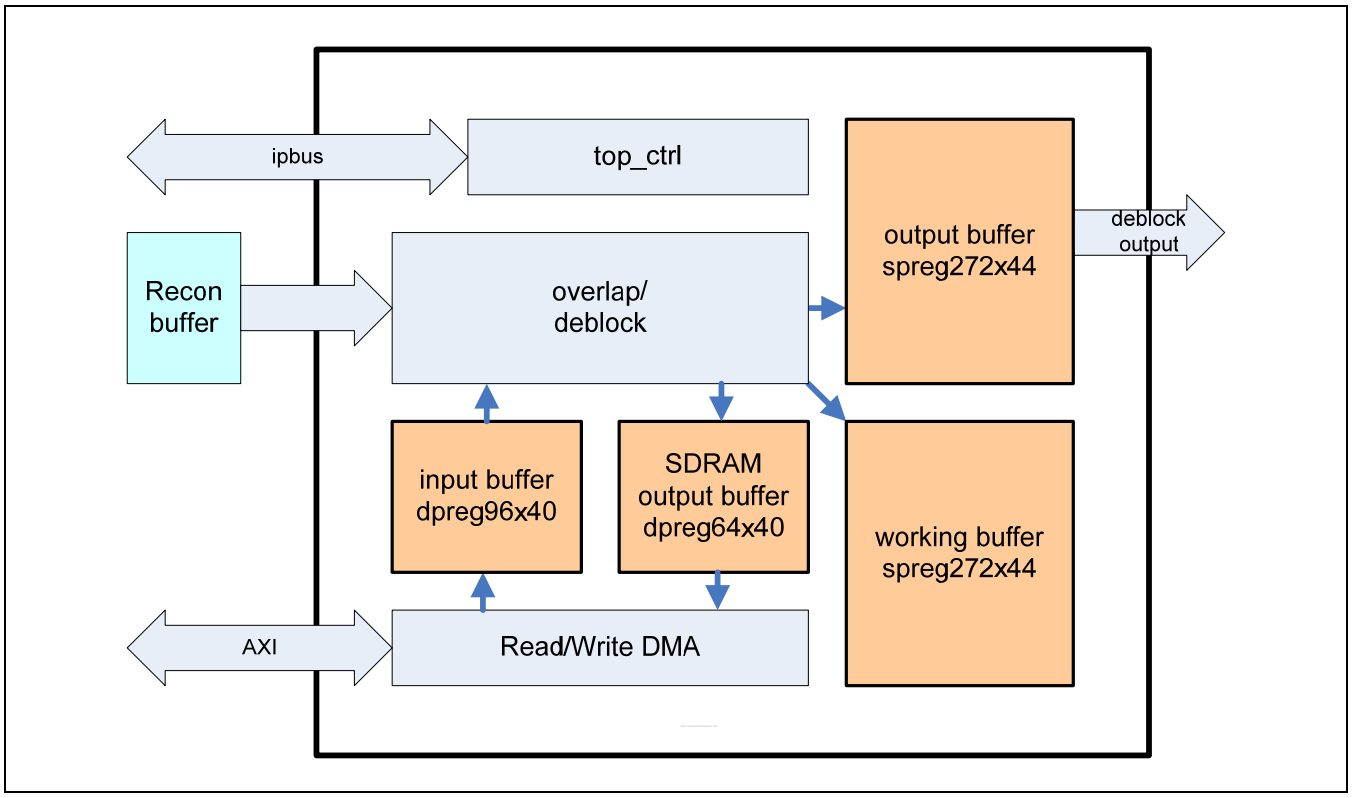


Figure 21-25. Overlap/deblocking filter architecture

The main controller (top_ctrl) is controlled through ip-bus by BIT processor. The BIT processor writes control information (processing mode, filter mode (H.263/H.264/MPEG-4/VC-1), run command etc.) into the control register in main controller.

Overlap-smoothing filter is enabled in VC-1 mode. In VC-1 mode, overlap-smoothing filtering is processed before deblocking filter. Deblock/overlap-smoothing filter reads the data to filter from reconstructed buffer. In stand-alone mode, the reconstructed data is loaded from SDRAM. Deblock/overlap-smoothing filter uses working buffer as temporal working buffer for filtering process. After filtering process is finished, the output data is moved to output buffer for next pipe-line stage. For filtering macroblock edge, the neighbor macroblock data is stored in working buffer.

21.4.8.4 H.264 Deblocking filter

A filtering shall be applied to all 4x4 block edges of a picture, except edges at the boundary of the picture. The filtering is performed on a macroblock basis and processed in order of increasing macroblock addresses. For each macroblock, vertical edges are filtered first, from left to right, and then horizontal edges are filtered from top to bottom. Sample values modified during filtering of vertical edges are used as input for the filtering of the horizontal edges for the same macroblock.

The filter adjusts its strength depending upon compression mode of a macroblock (Intra or Inter), the quantization parameter, motion vector and pixel values. Parameters for filtering operation such as boundary strength, indexA, indexB are set by the BIT processor.

21.4.8.5 H.263 Annex J Deblocking filter

The filtering is performed on 8x8 block edges, except across picture edge. 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 need not be influenced by previous filtering across a vertical edge. Parameters for filtering operation are set by the BIT processor.

21.4.8.6 VC-1 Overlap-smoothing filter

VC-1 overlap-smoothing filtering shall be performed subsequent to decode the frame, and prior to deblocking filter. The edges of an 8x8 block that separate two intra blocks are filtered.

Vertical edges will be filtered first, followed by the horizontal edges. Subsequent to filtering, the constant value of 128 will be added to each pixel of the block, which will be clamped to the range [0 255] to produce the reconstructed output.

The BIT processor writes the neighboring block information for overlap-smoothing filter

21.4.8.7 VC-1 DEBLOCKING FILTER

VC-1 deblocking filtering process operates on the pixels that border neighboring blocks. The block boundaries may occur at every 4^{th}, 8^{th}, 12^{th}, etc pixel row or column in P pictures. Filtering the I pictures occurs at every 8^{th}, 16^{th}, 24^{th}, etc pixel row and column.

The horizontal boundary lines will be filtered first followed by the vertical lines. All blocks and subblocks that have a horizontal boundary along the 8^{th}, 16^{th}, 24^{th}, etc horizontal lines will be filtered. Next, all subblocks that have a horizontal boundary along the 4^{th} 12^{th}, 20^{th}, etc horizontal lines will be filtered. Next, all blocks and subblocks that have a vertical boundary along the 8^{th}, 16^{th}, 24^{th}, etc vertical lines will be filtered. Next, all subblocks that have a vertical boundary along the 4^{th} 12^{th}, 20^{th}, etc vertical lines will be filtered

21.4.8.8 MPEG-4 Deblocking filter for post-processing

FIMV-MFC V1.0 can apply deblocking filtering for the MPEG-4 decoded image using either H.264-like filtering or H.263-like filtering operation. The BIT processor generates parameters suitable for selected mode based on the result from the MPEG-4 decoding process.

21.4.9 COEFFICIENT BUFFER INTERFACE

The coefficient buffer interface provides a channel for the BIT processor to read quantized coefficients resulted from encoding process or to send variable-length-decoded coefficients to the video codec module for decoding process. The coefficient buffer interface also performs reordering of coefficients based on the scan type.

21.4.9.1 Block diagram

Figure 21-26 illustrates the block diagram of the coefficient buffer interface. All sub-modules related to processing coefficient as input or as output are connected to the coefficient buffer interface. The BIT processor sets scan type based on the result from encoding process or from decoding process. To remove reading coefficient with zero-value, there are flag registers-6x64 bits- to indicate it.

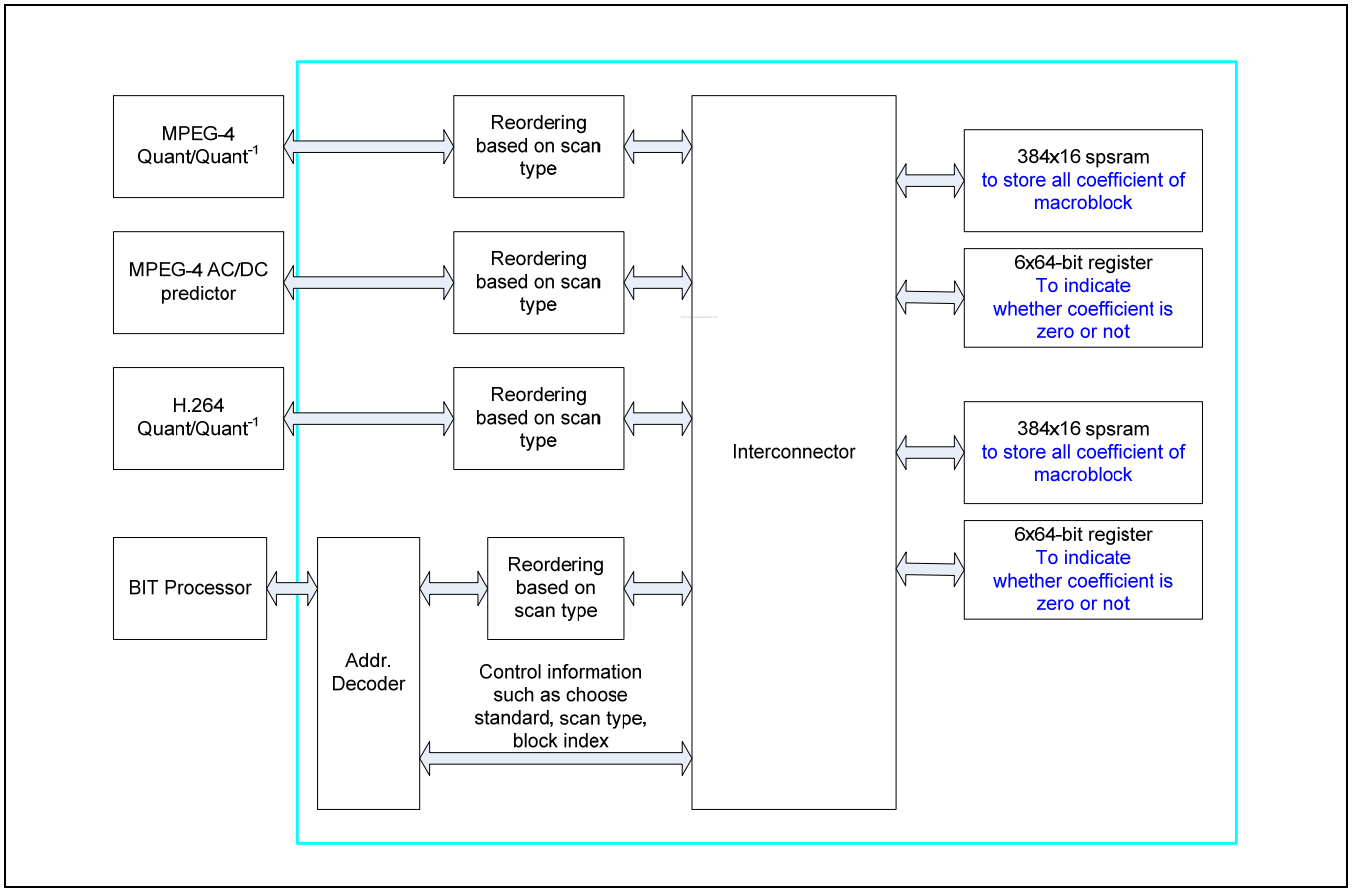


Figure 21-26 Coefficient memory interface block diagram

21.4.9.2 Reordering coefficients

Reordering is performed based on scan types set by the BIT processor. The scan types are listed below.

- H.264

  The H.264 baseline profile uses only the zig-zag scan type.

- MPEG-4/H.263P3

The MPEG-4/H.263P3 uses 3 scan types zig-zag, alternative horizontal, and alternative vertical scan type. Which scan type is selected based on the AC/DC prediction flag and macroblock type. In case of inter-macroblock type, only zig-zag scan type is used. For intra-macroblock, if AC prediction flag is '1', the scan type is dependent on the prediction direction of DC coefficient. If AC prediction flag is '0', only zig-zag scan type is applied.

21.4.9.3 Accessing the coefficient memory

The BIT processor reads or writes coefficient in the coefficient memory by specifying the block index, scan type, and the corresponding coefficient index. Figure 21-27 illustrates how the BIT processor accesses the coefficient memory.

- Block index : In case of the MPEG-4/H.263P3, each 8x8 block has its own index in range from 0 to 6. For the H.264, the range for the block index- the size of block is 4x4- is from -1 to 25. The assignment of the index is based on the standard.
- Scan type : The BIT processor does not have to consider it should write or read coefficient to which position of the coefficient memory for both encoding and decoding. The coefficient memory interface writes or reads coefficients after reordering them based on the specified scan type.

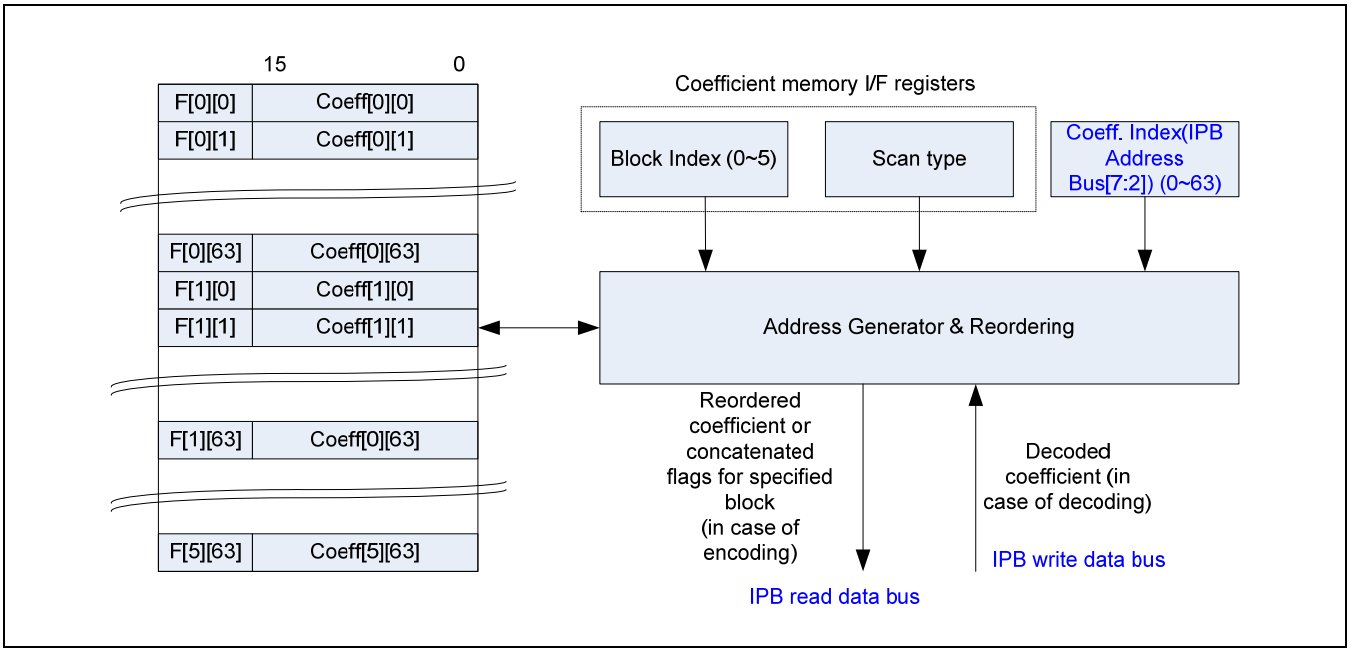


Figure 21-27. Coefficient memory access

21.4.9.4 Encoder operation

In case of encoding, quantized coefficients are written to the internal memory of the coefficient buffer interface without any reordering. When the BIT processor reads the coefficients, they are reordered and the flags that indicate corresponding coefficient has a non-zero value are sent to the BIT processor to check if there is coefficient to be encoded. The BIT processor can know the number of the coefficients that have non-zero value by simply counting number of bit that is not zero. Therefore, the BIT processor read only the non-zero coefficients and computes coded block pattern without consuming any computation power.

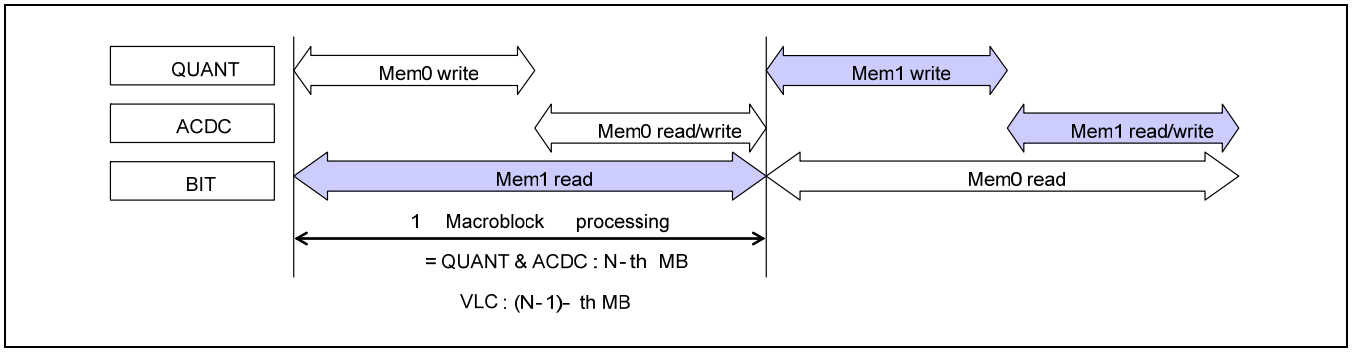


Figure 21-28. MPEG-4 encoding case

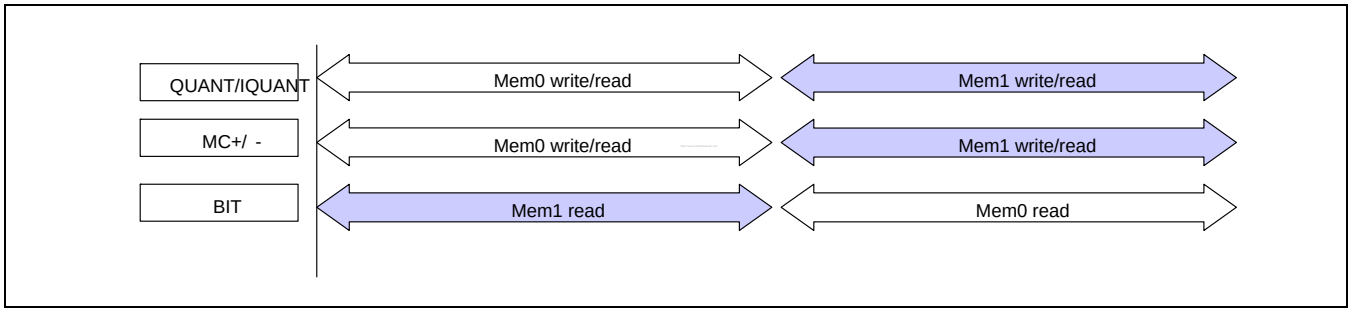


Figure 21-29. H.264 encoding case

21.4.9.5 Decoder operation

The BIT processor writes decoded coefficient without reordering-inverse zig-zag scanning. The reordering process is performed when the inverse quantization module reads coefficients from the coefficient buffer interface. The sub-modules such as the quantizer and AC/DC predictor handles zero if corresponding flag of coefficient is zero.

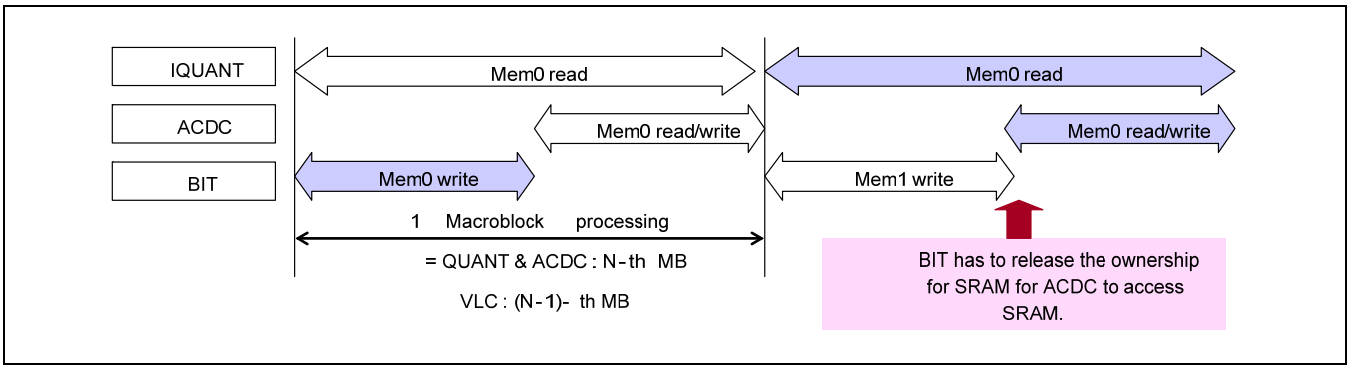


Figure 21-30 (a). MPEG-4 decoding case

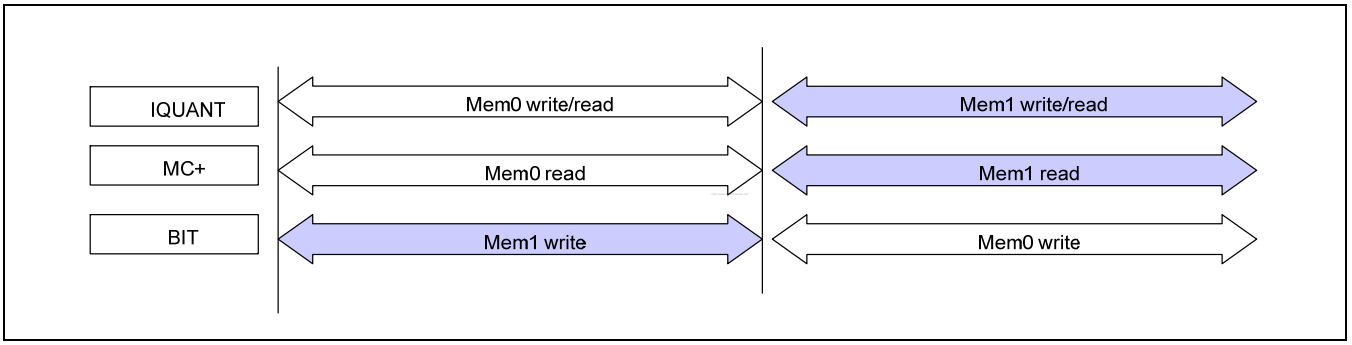


Figure 21-30 (b). H.264 decoding case

NOTE:

In the Figure 21-27, Figure 21-28, Figure 21-29, and Figure 21-30, MC+ means reconstruction, and MC- computing residual.

21.4.10 MACROBLOCK CONTROLLER

FIMV-MFC V1.0 has a complex and large number of pipeline for high-performance. To manage it wholly by the BIT processor is not suitable. Therefore, FIMV-MFC V1.0 embeds the macroblock controller to control all sub module of the video codec based on the configuration of pipelining by the BIT processor. This scheme reduces the load on the BIT processor and guarantees the programmability of the IP.

Before the video codec encode or decode a macroblock, the BIT processor configures how the pipeline of the codec is structured. If all processes are completed for encoding/decoding a macroblock, the macroblock controller indicates its completion.

To summarize, the BIT processor configures which sub-modules are enabled for current macroblock processing, and the macroblock controller manages corresponding sub-modules based on the configuration.

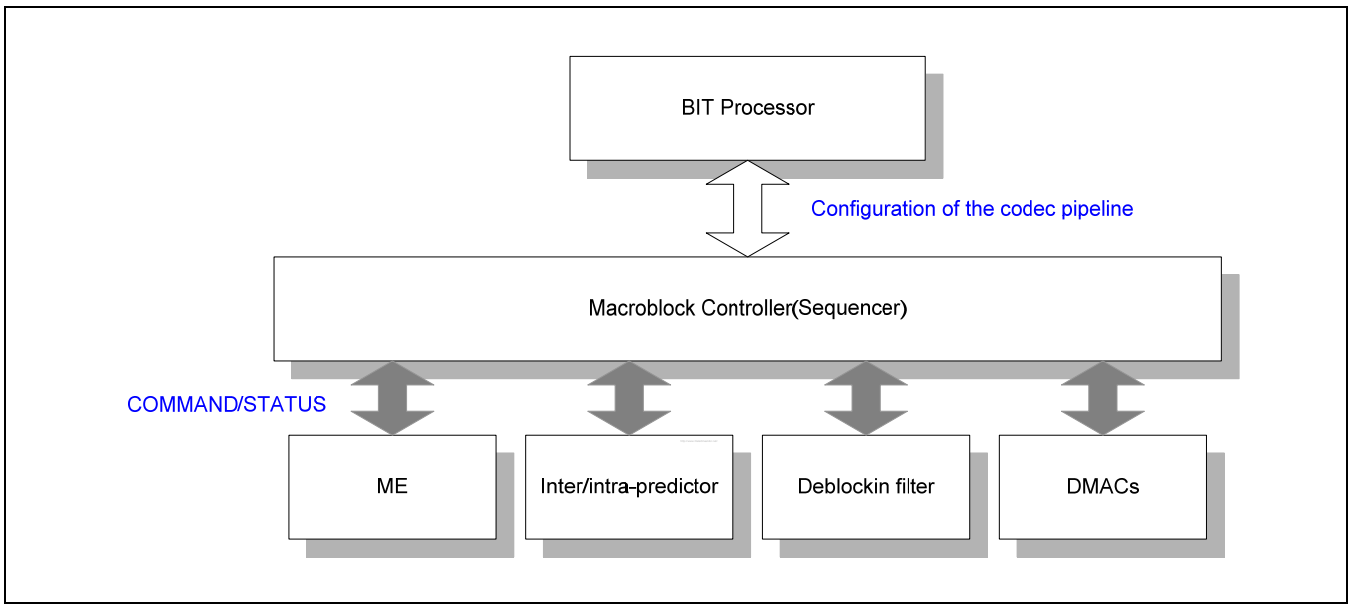


Figure 21-31. Macroblock controller connectivity

21.5 FIMV-MFC V1.0 PROGRAMMING MODEL (SPECIAL FUNCTION REGISTER)

The FIMV-MFC V1.0 is communicated with a host processor through the APB bus interface. Table 21-2 illustrates the address map of the region that could be accessed via the APB.

Table 21-2. Internal Register Address Map

PADDR[11:9]	Module	Description
3'b000	Host interface of the BIT processor	
3'b001	Macroblock controller(sequencer)	These registers cannot be accessed by a host processor in normal operation.
3'b010	Coefficient memory interface	
3'b011	Deblocking filter	
3'b100	Motion estimation	
3'b101	Inter-predictor	
3'b110	VC-1 scaler	
3'b111	S/W RESET	Soft-ware reset module

21.5.1 BIT PROCESSOR OPERATIONS

21.5.1.1 Description of BIT Processor Registers

These host interface registers can be partitioned into three categories according to their usage as listed below:

- BIT Processor Control Registers:
  Host interface registers in this category will be used to update or show BIT processor status to host processors. Most of these registers will be used for initializing BIT processor during boot-up.
- BIT Processor Global Registers:
  Host interface registers in this category will be used to store all the global variables, which will be kept even while active instance is changed. Basically all the buffer addresses and some global options will be safely stored into these registers.
- BIT Processor Command I/O Registers:
  Host interface registers in this category will be overwritten or updated whenever new command is transmitted from Host processor. All the commands with input arguments and all the corresponding responses with return values will be handled using these registers.

Address 0x000 ~ 0x0FC (64 registers address space) are H/W registers. This register has reset values and the function is fixed (not configurable), and Address 0x100 ~ 0x1FC (64 registers) are general purpose S/W registers. They have no reset values and are configurable by BIT firmware. They are mainly used for interface between host and BIT processor.

Upper 32 registers (address 0x100 ~ 0x17C) are used as static parameters. The meanings or functions of those registers are not changed for all kinds of run commands (SEQ_INIT, SEQ_END, PICTURE_RUN, ...) and applied to whole commands and processes. Lower 32 registers (address 0x180 ~ 0x1FC) are used as temporal command arguments. The meanings or functions of those registers may be changed for each run command.

For example BitStreamCtrl register is applied to whole processes. It means all encoding/decoding operations are affected by BitStreamCtrl register. But CMD_DEC_SEQ_STRIDE register is applied to only current running process of executing DEC_SEQ_INIT command. So that register (address 0x18C) will be used as different function for other commands. The value of [LineStride] is applied to only current process so all processes may have different line stride offset because BIT processor reads the CMD_DEC_SEQ_STRIDE register during executing DEC_SEQ_INIT command and stores [LineStride] value to internal memory.

21.5.1.2 BIT Processor Code Download

BIT processor has 6144 words internal code memory (word : 16 bit). The internal code memory is used as the instruction cache controlled by BIT firmware. The total code image may be bigger than 6144 words and must reside in SDRAM. BIT firmware loads appropriate code image from SDRAM at run time. Host must inform the start SDRAM byte address of code image to the BIT processor by set [CodeBufAddr] register.

For example, two processes (MPEG4 Decoder, H.264 Encoder) run simultaneously (full duplex case). If host executes ENC_PIC_RUN command after DEC_PIC_RUN command, BIT processor needs H.264 encoding code image to execute H.264 encoding a picture, therefore it loads automatically H.264 encoding code image from SDRAM (context switching). For the initial BIT processor executing (initial start after hardware reset), initial booting code image must be downloaded directly by host. Before executing the BIT processor by set [CodeRun] register, host must directly downloads booting code image (some amount of uppermost part of code image) to the lowest code memory (address 0) by set [CodeDownLoad] register.

The total byte size of code image of current version of BIT firmware is 80 KB (40K words) and the booting code image is 1024 byte (512 words).

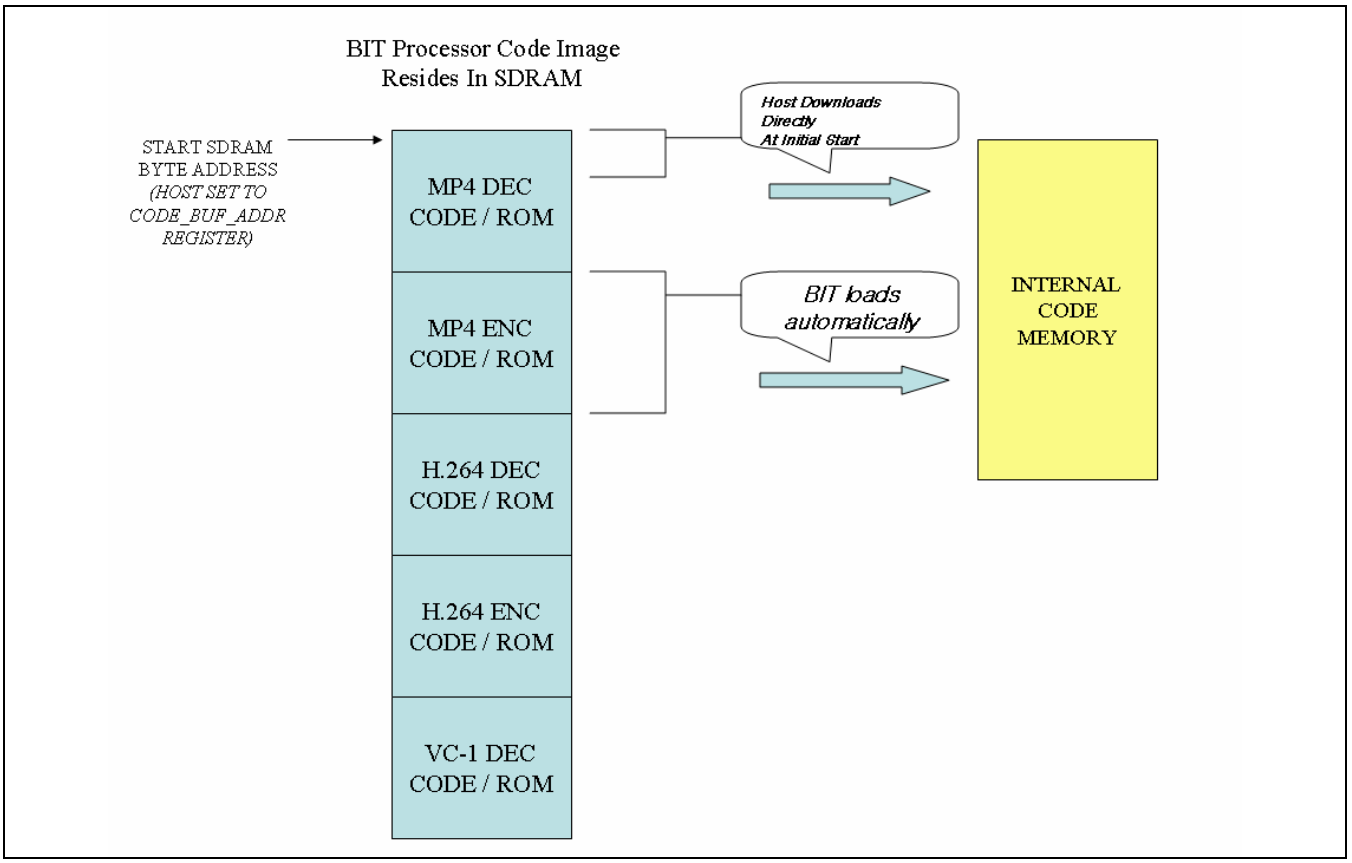


Figure 21-32. How to Download Code image into the BIT Processor

21.5.1.3 Bit Stream Buffer management

External bit stream buffer resides in SDRAM and is composed of ring buffer. The start address of ring buffer and buffer size must be written by host to BIT processor. The current read or write address of ring buffer is automatically wrapped-around.

In decoder case, host writes the bit stream to be decoded then BIT processor reads the bit stream. In this case, the bit stream overwriting or underflow may occur and if it occurs, decoding will fail. To prevent overwriting or underflow, current bit stream read/write pointer must be exchanged between host and BIT processor. BIT processor writes current read pointer of ring buffer to BitStreamRdPtr register and host must write current write pointer of ring buffer to BitStreamWrPtr register. BIT processor checks the bit buffer empty(undeflow) status by comparing current BitStreamRdPtr and BitStreamWrPtr. If no more bit stream data is available to be decoded (buffer empty status), BIT processor stops bit stream decoding to prevent mis-reading the bit stream and waits until host write more bit stream data and updates BitStreamWrPtr. Host must check the current BitStreamRdPtr and BitStreamWrPtr before writing more bit stream data to ring buffer to prevent overwriting bit stream data.

In encoder case, BIT processor and host change the roll, e.g. BIT processor writes the bit stream and host read the encoded bit stream. Therefore BIT processor writes current write pointer of ring buffer to BitStreamWrPtr register and host must write current read pointer of ring buffer to BitStreamRdPtr register. BIT processor will wait writing encoded bit stream data if the ring buffer is full. Therefore next writing will be overwriting the pre-stored data, which is not read by host. Host will wait till reading encoded bit stream data if the ring buffer is empty so next reading will be reading again pre-read data.

If [BufStsCheckDis] flag in BitStreamCtrl register is “1”, the option of bit stream overfow/underflow check by BIT processor is disabled. Therefore the encoding/decoding operation never stops because of bit stream buffer status. In this case, the host processor does not need the writing current read/write pointer to BitStreamRdPtr/BitStreamWrPtr but host guarantee that bit stream overflow/underflow will never occur.

If [BufPicReset] flag is in BitStreamCtrl register is “1”, the external bit stream buffer does not operate as a ring buffer for encoding case only. The encoded bit stream is always written to start of external bit stream buffer at every end of picture encoding. Therefore, host must read whole encoded bit stream of one picture at completion of ENC_PIC_RUN command.

In addition to [BufPicReset] flag, if [EncDynBufAllocEn] flag is “1” when [BufPicReset] flag is “1”, the external buffer start address and size can be changed frame by frame and The bit stream buffer operates like when [BufPicReset] flag is “1.” So host can double buffering by setting [EncDynBufAllocEn] flag.

If [BufPicFlush] flag in BitStreamCtrl register is “1”, all of encoded bit stream data of a picture is written to the external bit buffer at every end of encoding a picture. The exact byte address is written to BitStreamWrPtr.

The internal bit stream buffer also exists in the BIT processor and is used as a cache between BIT processor core and external bit stream buffer in SDRAM. Incase of decoding, BIT processor reads the bit stream data from external bit stream buffer in SDRAM and store to internal bit stream buffer. The real parsing operation is executed in internal buffer (not external buffer directly) for efficiency. In case of encoding, the encoded bit stream data stored to internal buffer first then written to external buffer later by chunk. The internal bit stream buffer is composed of 512 bytes. Therefore, the bit stream data is read from (or written to, for encoding case) external bit buffer by 512 bytes. The BitStreamRdPtr or BitStreamWrPtr is increased by 512 bytes because internal bit stream data is read/written by 512 byte chunk only. Therefore at the end of encoding a picture, not all of encoded bit stream of a picture is written to the external bit buffer. The some amount of last encoded stream data (less than 512 byte) reside in internal bit buffer only and will be flushed later when accumulated encoded data is greater than 512 byte. To get the remaining encoded data at end of encoding sequence, host must execute ENC_SEQ_END command and BIT processor flush remaining encoded bit stream to external bit buffer. After ENC_SEQ_END command, BitStreamWrPtr is increased to the exact byte address and host may calculate exact byte size of encoded bit stream.

21.5.1.4 Description of Run Commands

The command arguments registers must be set by host prior to executing the command. After completion of command, return parameter registers are available to host. Host must set command at BusyFlag register is “0”. After BIT processor acknowledges new command, BusyFlag register is “1” and after the command completion BusyFlag register is “0”. Host may acknowledge the command completion by polling the BusyFlag register or interrupt by BIT processor.

DEC_SEQ_INIT

This command initiates a decoding process. All of the DEC_SEQ_INIT command arguments register’s value are stored to internal memory during executing DEC_SEQ_INIT command by BIT processor and applied to current process permanently. Therefore the value of DEC_SEQ_INIT command register will never be changed during followed command (for example DEC_PIC_RUN)

At DEC_SEQ_INIT command, BIT processor finds sequence header and parsing the header to extract the bit stream information such as picture size. Then the information is reported to DEC_SEQ_INIT return registers.

ENC_SEQ_INIT

This command initiates an encoding process. At ENC_SEQ_INIT command, BIT processor reads encoding parameter from command argument registers and encodes sequence header. The encoding parameters which exist in ENC_SEQ_INIT command argument register are applied to followed ENC_PIC_RUN commands and

never be changed. For example, [SliceMode] in CMD_ENC_SLICE_MODE register is applied to entire pictures but [PictureQs] in CMD_ENC_PIC_QS register is changed at every ENC_PIC_RUN command. If invalid encoding parameter is set by host, BIT processor reports error flag to RET_ENC_SEQ_SUCCESS register.

DEC_SEQ_END

This command terminates decoding process. After DEC_SEQ_END command, no further DEC_PIC_RUN command will be accepted.

ENC_SEQ_END

This command terminates encoding process. BIT processor transfer remaining internal encoded bit stream data to external bit stream buffer resides in SDRAM if [BufPicFlush] flag is “0” and [BufPicReset] flag is “0”. After ENC_SEQ_END command, no further ENC_PIC_RUN command will be accepted.

DEC_PIC_RUN

This command decodes one picture. After decoding one picture, BIT processor reports successful decoding flag to RET_DEC_PIC_ERR_MB_NUM register and display frame buffer index to RET_DEC_PIC_IDX register.

ENC_PIC_RUN

This command encodes one picture. The SDRAM address of encoded source frame buffer must be set to command argument register.

SET_FRAME_BUF

This command informs frame buffer addresses to be used as a decoding/reconstructing image to BIT processor. Total 63 frame buffers may be used for decoding/reconstructing. The minimal number of frame buffers required for successful decoding is informed by RET_DEC_SEQ_FRAME_NEED register. In encoding case, two frame buffers are sufficient. The number of total frame buffer is set to CMD_SET_FRAME_BUF_NUM register by host.

The decoding (decode case) / reconstructing (encode case) image must be reserved for motion compensation reference until not used for reference. Therefore the decoded/reconstructed frame buffer is re-used carefully. BIT processor receives the whole frame buffer address by this command before start of encoding/decoding pictures then manages the frame buffer allocation for next storage area for decoding/reconstructing.

The frame buffer addresses are stored to SDRAM of address [ParaBufAddr]. The Luminance and two chrominance buffer address for each frame index must be stored. The format of addresses is illustrated in the following diagram:

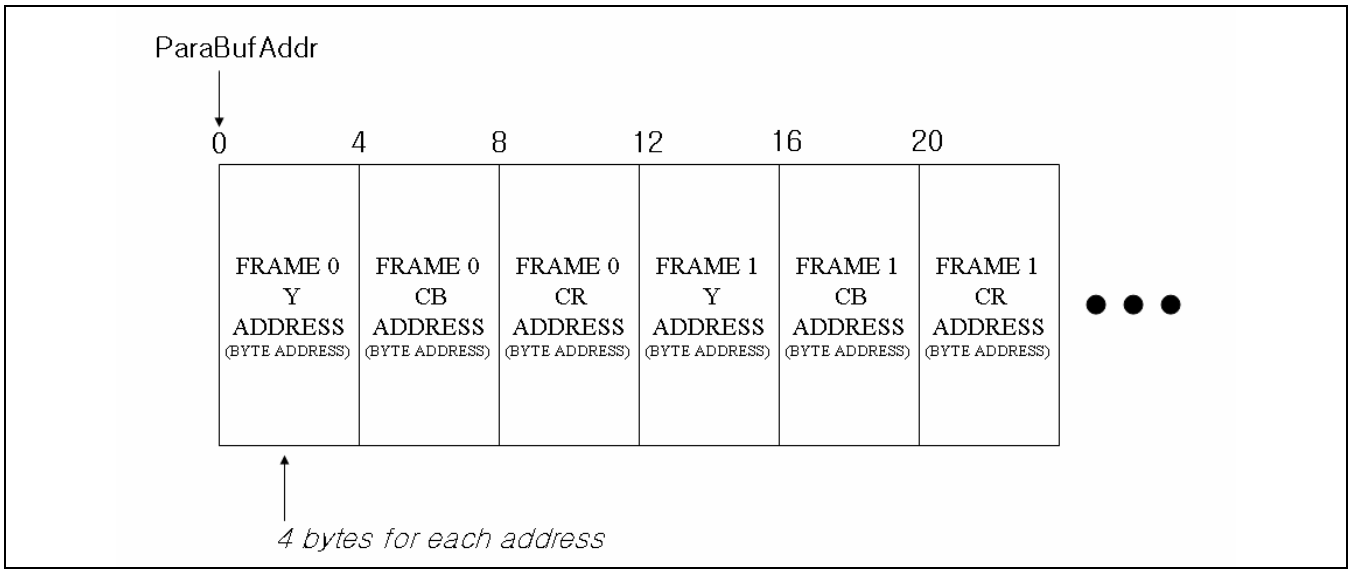


Figure 21-33. How to Store Frame buffer address in the Parameter buffer area

ENC_HEADER

This command encodes the specific header. The sequence headers must be encoded at start of bit stream and BIT processor encodes the sequence header at ENC_SEQ_INIT command. In some application like video telephony the sequence headers must be transferred for error resilience or decoder refreshment. In this case host set ENC_HEADER command between ENC_PIC_RUN commands to insert the specific header in the bit stream. The header code of the input header syntax must be set to CMD_ENC_HEADER_CODE register.

DEC_PARA_SET

This command adds sequence parameter set or picture parameter set to decoder in H.264 case

The sequence parameter set or picture parameter set may be conveyed via “out-of-band”. In that case host must transfer the sequence parameter set or picture parameter set to decoder by this command. The sequence parameter set or picture parameter set must be written to Parameter Buffer of BIT processor in RBSP format prior to executing this command. BIT processor decodes the transferred sequence/picture parameter and stores decoded contents. The decoded sequence/picture parameter set is activated at decoding slice header with the matched sequence/picture parameter set id.

Multiple sequence/picture parameter sets may be delivered to decoder. They are distinguished by different sequence/picture parameter set id. The sequence parameter set id is encoded as 5 bits (0~31) in sequence parameter set RBSP. The picture parameter set id is encoded as 8 bits (0 ~ 255). BIT processor can process 32 sequence parameter sets and 256 picture parameter sets. The type (sequence or picture) and size (byte count) of conveyed sequence/picture parameter set must be delivered to BIT processor by command argument register.

ENC_PARA_SET

This command encodes sequence parameter set or picture parameter set and delivers to host via Parameter Buffer in H.264 case. In application that conveys sequence/picture parameter sets in “out-of-band”, host can obtain sequence/picture parameter set RBSP by this command.

BIT processor encodes sequence/picture parameter set and encoded RBSP stream is stored to Parameter Buffer instead of normal bitstream buffer. The encoded byte count of sequence/picture parameter set is returned by command return register.

MFC_SLEEP

This command saves the current status of VPU to work buffer area. So host can power down after saving environment variables of VPU to get power efficiency.
Note. VPU(Video Processor Unit) means MFC and host means ARM1176 CPU in the S3C6410.

MFC_WAKEUP

This command loads the former status of VPU which saved by host to work buffer area. So host can power on from sleep status after loading environment variables of VPU to get power efficiency.

21.5.1.5 Parameter Buffer Management

Parameter buffer is used for input command arguments or output return data at certain commands. The parameter buffer resides in SDRAM and the start byte address must be set to ParaBufAddr register.

Frame Buffer Address

This is the input command arguments for SET_FRAME_BUF command. For the detailed description refer to SET_FRAME_BUF command description.

Encoded Macro Block Bit Number

This is the output return data for ENC_PIC_RUN command. If [MbBitReport] flag in CMD_ENC_SEQ_OPTION register is “1”, BIT processor stores the start bit position of each macro block from the beginning of picture to the parameter buffer after ENC_PIC_RUN command completion. The start bit position is stored as unsigned 16 bit for each macro block. The macro block row stride offset is 128 bytes so the last 128-720/16*2 = 38 bytes of each macro block row area is never used. The byte address of macro block position <MbX, MbY> is {ParaBufAddr + MbY * 128 + MbX * 2}. The maximum encoding source picture height is 576 so the maximum size of the macro block bit number buffer is 128 * 576/16 = 4608 byte (4.5 KB). The detailed format is illustrated in the following diagram.

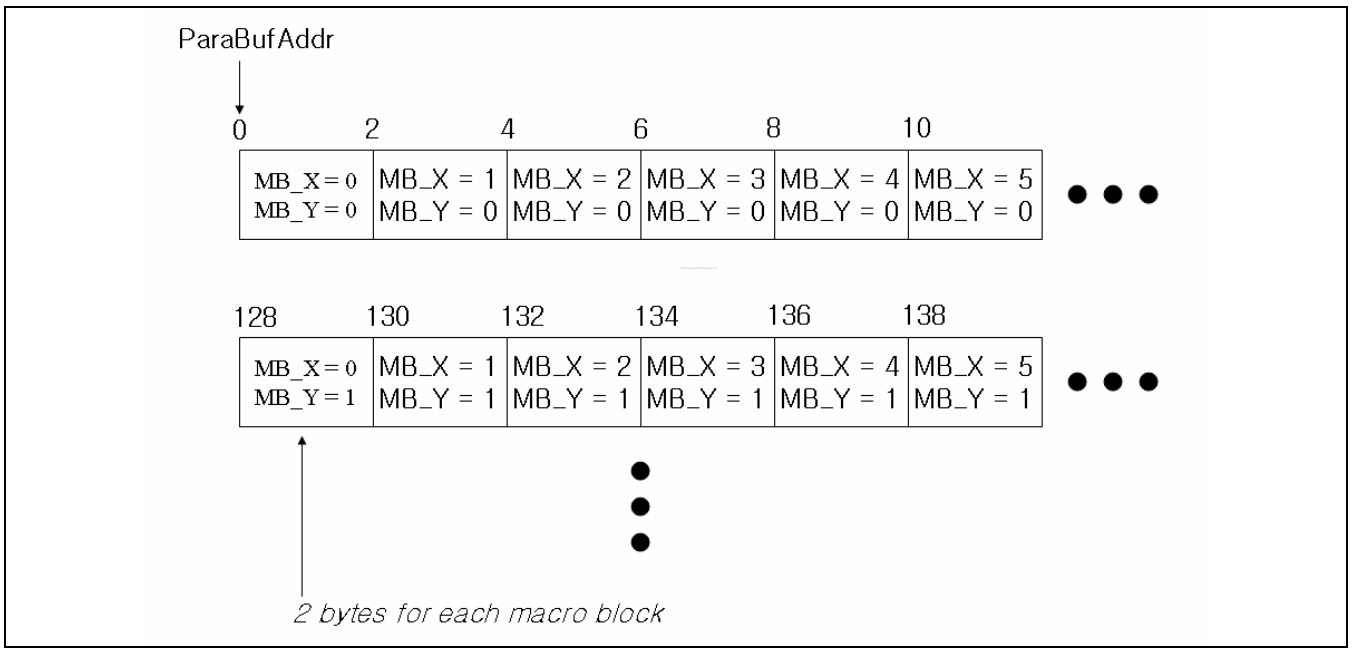


Figure 21-34. How to Store the start bit position of each macro block in the Parameter buffer when an MbBitReport flag is set

Encoded Slice Information

This is the output return data for ENC_PIC_RUN command. If [SliceInfoReport] flag in CMD_ENC_SEQ_OPTION register is “1”, BIT processor stores the end SDRAM address in the external bit stream buffer of each encoded slice to the parameter buffer after ENC_PIC_RUN command completion. The total number of generated slices is stored to RET_ENC_PIC_SLICE_NUM register and the SDRAM address of the end position of each slice is stored to parameter buffer. From the end position of each slice, host may calculate exact byte count size of each slice. The start address of slice position buffer is {ParaBufAddr + 4608(0x1200)} just below macro block bit number buffer. The SDRAM address of slice end position is unsigned 32 bit for each slice. The detailed format is illustrated in the following diagram.

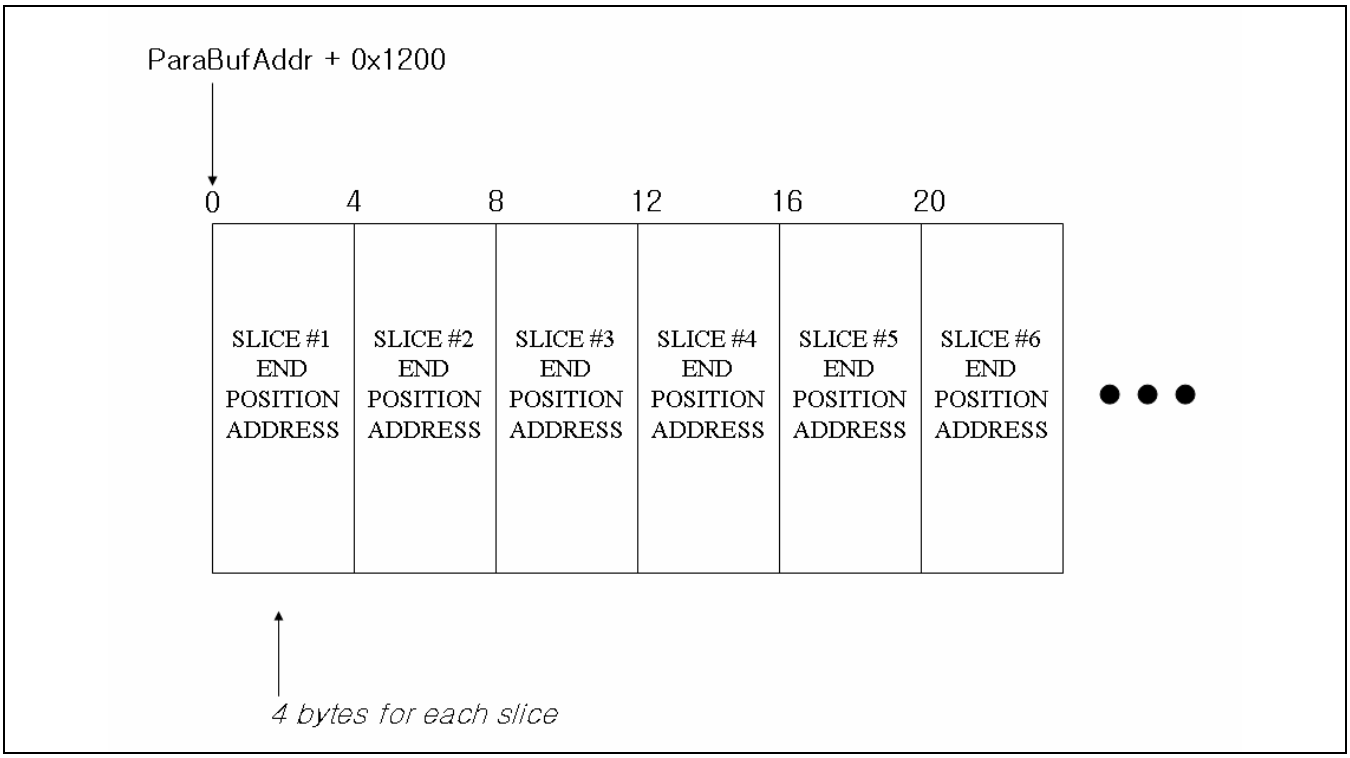


Figure 21-35. How to Store the end address of each encoded slice in the Parameter buffer when a SliceInfoReport flag is set

Sequence/Picture parameter set RBSP

This is the input argument for DEC_PARA_SET command or the output encoded RBSP stream for ENC_PARA_SET.

21.5.1.6 Working Buffer Management

This area is used for internal working buffer in SDRAM for encoding/decoding operation. For example, the reconstructed pixel row buffer for MPEG4 AC/DC prediction or H.264 intra prediction, context saving buffer for running multiple processes, bit stream re-ordering buffer for MPEG4 data partition or H.264 FMO and so on. The required working buffer size is varied according to decode/encode size and capability. For example, AC/DC prediction buffer size is determined by picture width and maximum bit stream re-ordering buffer for data partition is determined by the maximum bit stream size of one picture.

The current version of firmware supports picture decoding and encoding of full D1 size picture (720 x 576) with bit-rate up to 10 Mbps. If working buffer configuration option is disabled in720 x 576 case, working buffer will be organized as follows:

The static buffer is used commonly in whole processes/codecs and temporal picture buffer is re-used by each process (codec).

Table 21-3. Internal Register Address Map

Type	Name	Description	Size (KB)
STATIC	STATIC_PRC_DMEM	BIT processor data memory of each process for context switching	40[1]
	STATIC_PRC_SEQ	Static data storage of each sequence	160[2]
TEMP_PIC MP4_DEC	MP4_DEC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	picWidth*8
	MP4_DEC_DP1	Bit stream reordering buffer for data partition	48
	MP4_DEC_DP2	Bit stream reordering buffer for data partition	48
TEMP_PIC MP4_ENC	MP4_ENC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	picWidth*8
	MP4_ENC_DP1	Bit stream reordering buffer for data partition	48
	MP4_ENC_DP2	Bit stream reordering buffer for data partition	48
TEMP_PIC AVC_DEC	AVC_DEC_IP	Intra prediction buffer of Y/Cb/Cr	72
	AVC_DEC_FMO	FMO group status buffer	5.5
	AVC_DEC_SLICE_INFO	Slice information buffer Maximum 1620 slices per picture	12.66
	AVC_DEC_NAL_BUF	Save NAL Unit Buffer	0.5
	AVC_DEC_SLICE	Slice data RBSP buffer All slice data RBSP of one picture is stored. Picture raw data size in worst case.	XSIZE*YSIZE*1.5/1024
TEMP_PIC VC1_DEC	VC1_DEC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	6
	VC1_DEC_DEBLK	Overlap/deblock filter working buffer	8
	VC1_DEC_DIRECTMV	DirectMV working buffer	32
TEMP_PIC AVC_ENC	AVC_ENC_IP	Intra prediction buffer of Y/Cb/Cr	72
	AVC_ENC_FMO	FMO group status buffer	**256**

When application enables Work Buffer configuration option, application must carefully configure it in order to avoid unexpected malfunction of encoder. Work buffer can be divided into three parts, fixed static buffer, configurable static buffer and configurable temporal buffer, respectively. Fixed static buffer is used for context switching and static data storage of each instance. Size of fixed static buffer is 78 KB. Configurable static buffer is process buffer defined in register description section and is used when decode AVC and VC-1. Configurable static buffer must be allocated each instance and not be used as other usage before each instance. Configurable temporal buffer is temporary buffer defined in register description section and must be allocated explained. Configurable temporal buffer can be used after one picture processing end and reused by each instance. For example, if application performs full D1 processing of one MPEG4 decoding, two AVC decoding, one VC-1 decoding, one MPEG4 encoding and 1 AVC encoding simultaneously, then the recommended configuration of Work Buffer will be as follows.

[1] 5KB for each instance.

[2] 8 KB for each instance and 96 KB for worst case test such as Allegro test streams.

Table 21-4. Internal Register Address Map

Type	Name	Description	Size (KB)
STATIC	STATIC_PRC_DMEM	BIT processor data memory of each process for context switching	40
	STATIC_PRC_SEQ	Static data storage of each sequence	32
	STATIC_AVC_DEC_FMO	Dec FMO group buffer	5.5
	STATIC_AVC_DEC_NAL_BUF	Save NAL Unit Buffer	0.5
STATIC(configurable)	AVC_DEC_PS	PS Data save buffer	128
	VC1_DEC_DirectMV	Direct MV prediction buffer	6.33
TEMP_PIC MP4_DEC	MP4_DEC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	picWidth*8
	MP4_DEC_DP1	Bit stream reordering buffer for data partition	48
	MP4_DEC_DP2	Bit stream reordering buffer for data partition	48
TEMP_PIC MP4_ENC	MP4_ENC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	picWidth*8
	MP4_ENC_DP1	Bit stream reordering buffer for data partition	48
	MP4_ENC_DP2	Bit stream reordering buffer for data partition	48
TEMP_PIC AVC_DEC	AVC_DEC_IP_Y	Intra prediction buffer of Y	(FrameBufferStride*picHieght/16)/1024
	AVC_DEC_IP_Cb	Intra prediction buffer of Cb	[{FrameBufferStride/2*(picHieght/2)}/8]/1024
	AVC_DEC_IP_Cr	Intra prediction buffer of Cr	[{FrameBufferStride/2*(picHieght/2)}/8]/1024
	AVC_DEC_SLICE_INFO	Slice information buffer Maximum 1620 slices per picture	12.66
	AVC_DEC_SLICE	Slice data RBSP buffer All slice data RBSP of one picture is stored. Picture raw data size in worst case.	XSIZE*YSIZE*1.5/1024
TEMP_PIC AVC_ENC	AVC_ENC_IP_Y	Intra prediction buffer of Y	(FrameBufferStride*picHieght/16)/1024
	AVC_ENC_IP_Cb	Intra prediction buffer of Cb	[{FrameBufferStride/2*(picHieght/2)}/8]/1024
	AVC_ENC_IP_Cr	Intra prediction buffer of Cr	[{FrameBufferStride/2*(picHieght/2)}/8]/1024
	AVC_ENC_FMO	Enc FMO group buffer	32 * SliceGroupNum
TEMP_PIC VC1_DEC	VC1_DEC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	6
	VC1_DEC_DEBLK	Overlap/deblock filter working buffer	8

According to the requirements of target application, application could reserve Work Buffer space by selecting the sub-set of the buffers presented in this table.

21.5.1.7 Description of Run Process

BIT processor can execute maximum 8 processes simultaneously. Each process may have different Codec Standard (MPEG4 DECODE, H.264 ENCODE, etc.). Each process has different bit stream buffer and related read/write pointer register (BitStreamRdPtr0, BitStreamWrPtr0, etc.). The process is generated by DEC_SEQ_INIT/ENC_SEQ_INIT command and terminated by DEC_SEQ_END/ENC_SEQ_END command. Each process is identified with RunIndex register at executing command. Bit processor schedules every process by host executing command (picture by picture basis). Switching to different process requires a little cycle/bandwidth overhead for context switching of BIT processor. At each context switching about 8 KB data memory of each process is read from and written to SDRAM and additional 8 KB code memory is read from SDRAM if codec standard is different between previous and current process. Therefore worst-case context switching overhead SDRAM bandwidth is about {8KB(R)+ 8KB(W) + 8KB(R)} * 30 Hz * 8 process = 5760 KB / sec (at 30 frame / sec).

BIT processor requires about 0.75 cycles / byte for reading or writing SDRAM data at burst mode. Therefore worst case context switching overhead cycles are about {8K + 8K + 8K} * 0.75 * 30 Hz * 8 process ~= 4.4 M Cycles

21.5.1.8 Description of Pre/Post Rotation

Encoding source image may be rotated prior to encoding process (pre-rotation) and decoded output image may be rotated after decoding process. Host may inform the pre/post rotation mode at every encoding/decoding a picture.

In pre-rotation case, the rotation is performed at reading source image process. Therefore no further SDRAM bandwidth is needed for pre-rotation. After reading the rotated source image, remaining encoding process is same with no pre-rotation case.

In post-rotation case, the rotation is performed at storing decoded image process. But the decoded image (prior to rotation) is used for future frame decoding for reference of motion compensation. Therefore the decoded image must be stored when post-rotation is enabled and further SDRAM bandwidth is needed for additional post-rotated image store.

The pre/post rotation mode is composed of 4 bit field. The first bit (most significant bit) is horizontal mirroring, the next bit is vertical mirroring and last two bits (least significant two bits) are counterclockwise rotation degree. Each mirroring/rotation field (horizontal, vertical, rotation) is applied independently.

In case of the pre-rotator, the mirroring fields are applied prior to rotating then a counterclockwise rotating is applied to horizontal/vertical flipped image.

In case of the post-rotator, the rotation field is applied prior to horizontal/vertical mirroring then mirroring is applied to counterclockwise rotated image.

If counterclockwise rotation field is 1(90 degree) or 3(270 degree), the rotated image picture width/height will be exchanged. For example CIF size image (352 x 288) will be 288 x 352 after rotation and the decoding image and rotated image will be stored different format at post rotation case.

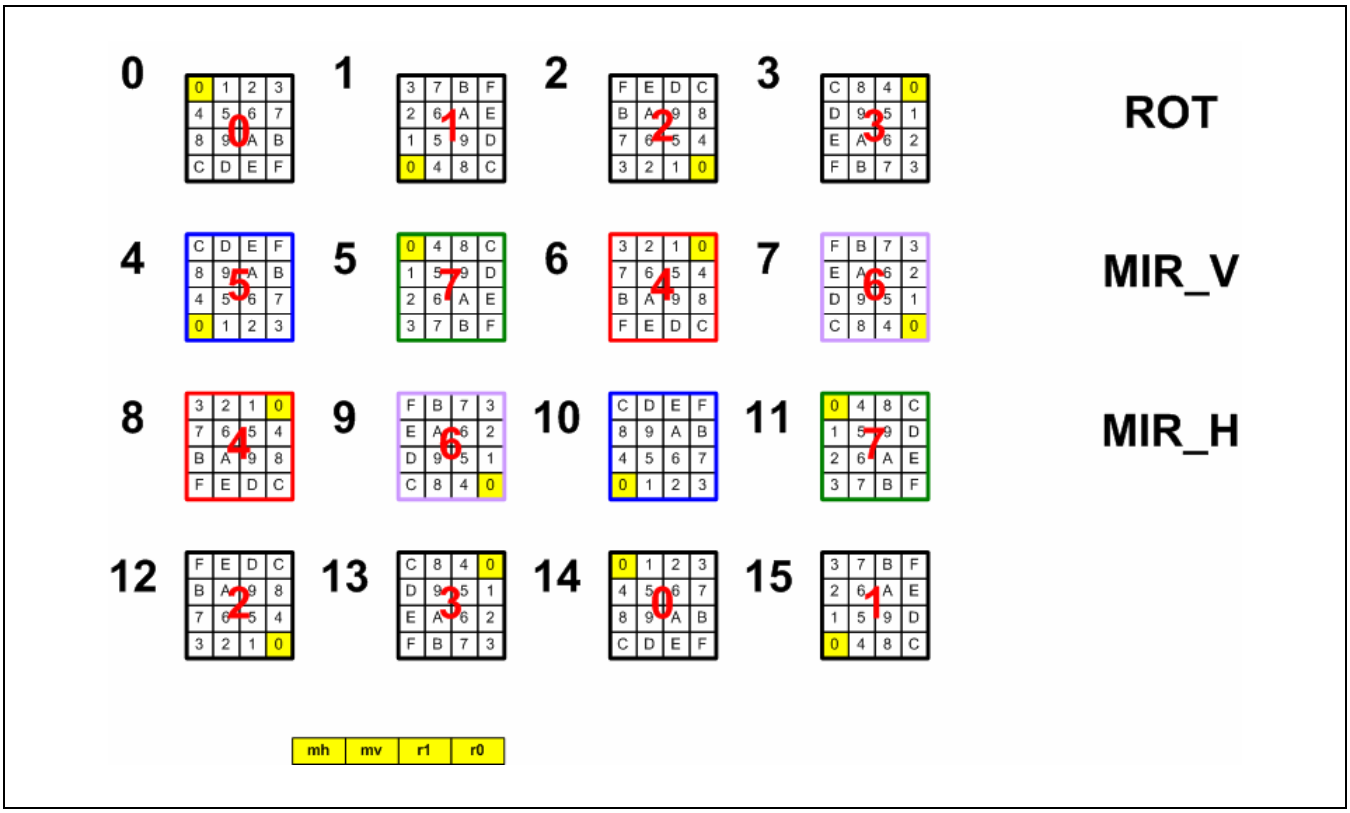


Figure 21-36. Description of a Rotation/Mirror operation

21.5.1.9 Sample operation flows

The following flowcharts are the simple examples of executing BIT processor for specific running scenario.

Example of Single MPEG4 Decoder

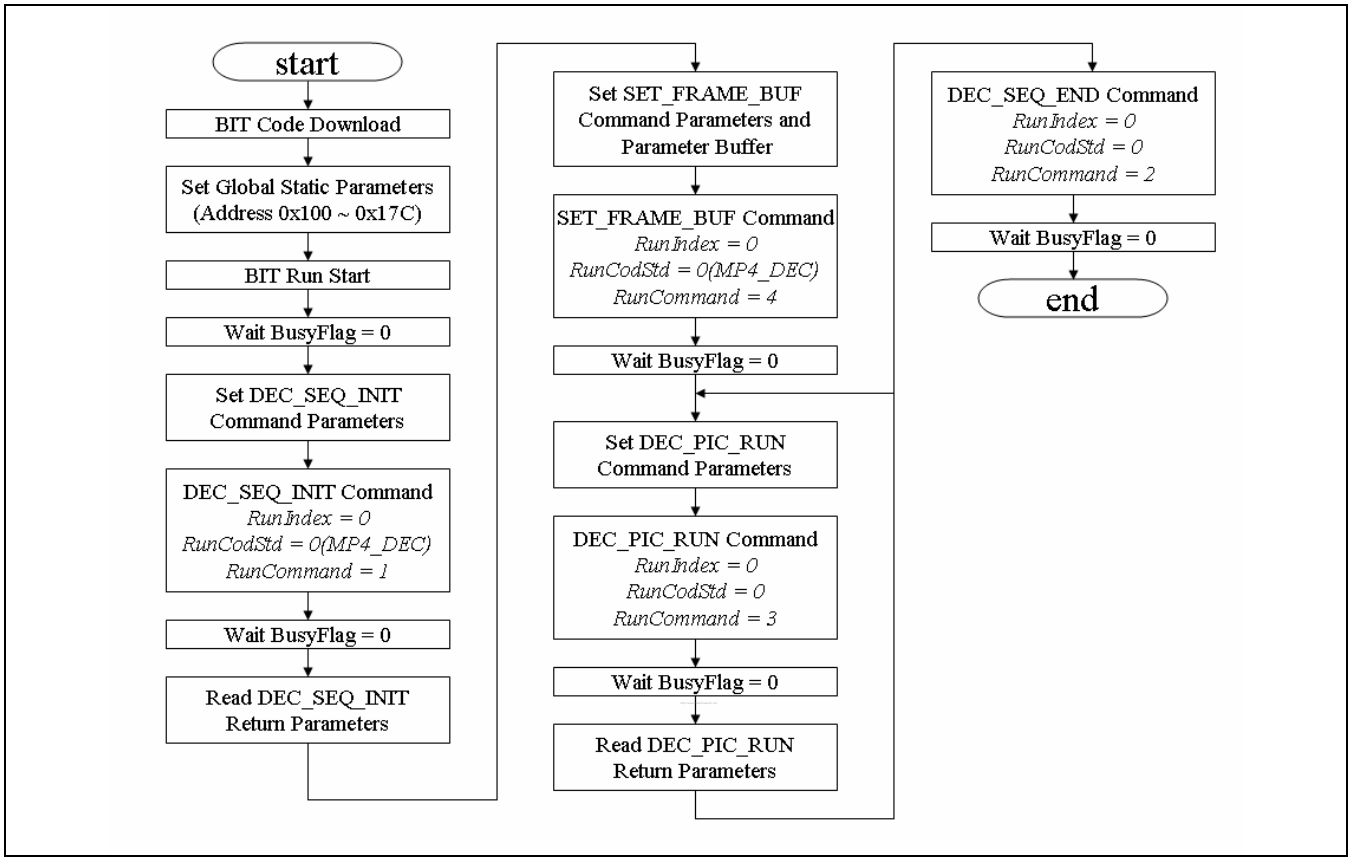


Figure 21-37. Example of Single MPEG4 Decoder

Example of Single H.264 Encoder

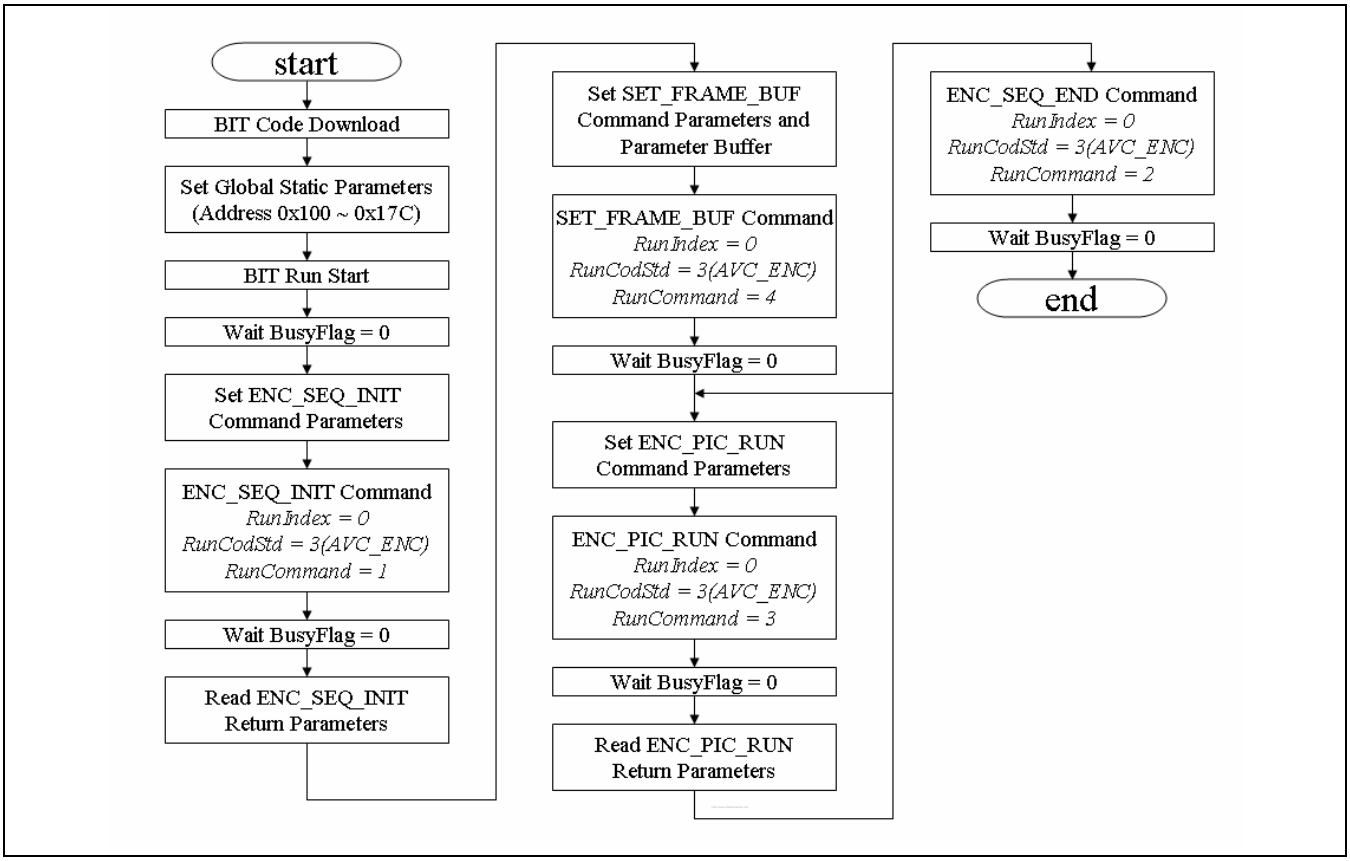


Figure 21-38. Example of Single H.264 Encoder

Example of H.264 Full Duplex

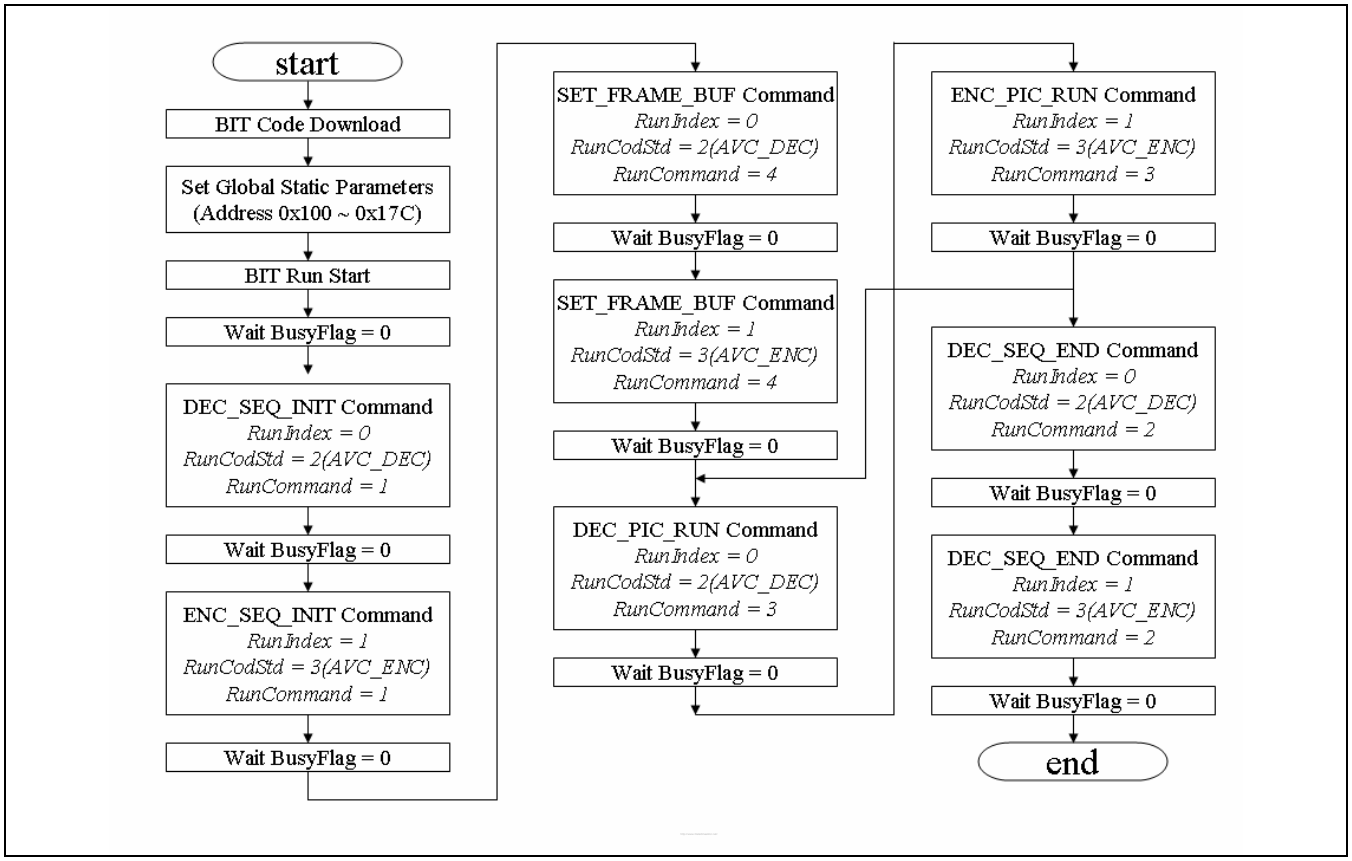


Figure 21-39. Example of H.264 Full Duplex

21.5.2 THE HOST INTERFACE REGISTERS

BIT Processor's registers are divided into 2 categories.

Address 0x000 ~ 0x0FC (64 registers address space) are H/W registers. These registers have reset values and the functions are fixed (not configurable).

Address 0x100 ~ 0x1FC (64 registers) are general purpose S/W registers. They have no reset values and are configurable by BIT firmware. They are used as interface between host and BIT processor.

Upper 32 registers (address 0x100 ~ 0x17C) are used as static parameters. The meanings or functions of those registers are not changed for all kinds of run commands (SEQ_INIT, SEQ_END, PICTURE_RUN) and applied to all commands. Lower 32 registers (address 0x180 ~ 0x1FC) are used as temporal command arguments. The meanings or functions of those registers may be changed for each run commands.

21.5.2.1 Summary of host interface registers

Table 21-5. BIT Processor Common Register Summary (BASE = 0x7E002000)

Address	Type	Width	Reset value	Name	Description
BASE+0x000	W		0	CodeRun	BIT run start
BASE+0x004	W		0	CodeDownLoad	Code Download Data register
BASE+0x008	W		0	HostIntReq	Host Interrupt Request to BIT
BASE+0x00C	W		0	BitIntClear	BIT Interrupt Clear
BASE+0x010	R		0	BitIntSts	BIT Interrupt Status
BASE+0x014	W		0	BitCodeReset	BIT Code Reset
BASE+0x018	R		0	BitCurPc	BIT Current PC
Protected for Internal Use					
BASE+0x100	R/W		N/A	CodeBufAdddr	CODE Table SDRAM Address
BASE+0x104	R/W		N/A	WorkBufAddr	Working Buffer SDRAM Address
BASE+0x108	R/W		N/A	ParaBufAddr	Argument/Return Parameter Buffer SDRAM Address
BASE+0x10C	R/W		N/A	BitStreamCtrl	Bit Stream Buffer Control
BASE+0x110	R/W		N/A	FrameMemCtrl	Frame Memory Control
BASE+0x114	R/W		N/A	DecFuncCtrl	Decoder Function Control[3]
BASE+0x11C	R/W		N/A	BitWorkBufCtrl	Work Buf Control[4]
BASE+0x120	R/W		N/A	BitStreamRdPtr0	Bit Stream Buffer Read Address of Run Index 0
BASE+0x124	R/W		N/A	BitStreamWrPtr0	Bit Stream Buffer Write Address of Run Index 0
BASE+0x128	R/W		N/A	BitStreamRdPtr1	Bit Stream Buffer Read Address of Run Index 1
BASE+0x12C	R/W		N/A	BitStreamWrPtr1	Bit Stream Buffer Write Address of Run Index 1
BASE+0x130	R/W		N/A	BitStreamRdPtr2	Bit Stream Buffer Read Address of Run Index 2
BASE+0x134	R/W		N/A	BitStreamWrPtr2	Bit Stream Buffer Write Address of Run Index 2
BASE+0x138	R/W		N/A	BitStreamRdPtr3	Bit Stream Buffer Read Address of Run Index 3
BASE+0x13C	R/W		N/A	BitStreamWrPtr3	Bit Stream Buffer Write Address of Run Index 3
BASE+0x140	R/W		N/A	BitStreamRdPtr4	Bit Stream Buffer Read Address of Run Index 4
BASE+0x144	R/W		N/A	BitStreamWrPtr4	Bit Stream Buffer Write Address of Run Index 4
BASE+0x148	R/W		N/A	BitStreamRdPtr5	Bit Stream Buffer Read Address of Run Index 5
BASE+0x14C	R/W		N/A	BitStreamWrPtr5	Bit Stream Buffer Write Address of Run Index 5
BASE+0x150	R/W		N/A	BitStreamRdPtr6	Bit Stream Buffer Read Address of Run Index 6
BASE+0x154	R/W		N/A	BitStreamWrPtr6	Bit Stream Buffer Write Address of Run Index 6

[3] This function control register is newly added to provide an escape control scheme by host processor.
[4] This Work Buffer control register is newly added to support Work Buffer configuration by host processor.

BASE+0x158	R/W		N/A	BitStreamRdPtr7	Bit Stream Buffer Read Address of Run Index 7
Address	**Type**	**Width**	**Reset value**	**Name**	**Description**
BASE+0x15C	R/W		N/A	BitStreamWrPtr7	Bit Stream Buffer Write Address of Run Index 7[5]
BASE+0x160	R		N/A	BusyFlag	Processor Busy Flag
BASE+0x164	R/W		N/A	RunCommand[6]	Run Command
BASE+0x168	R/W		N/A	RunIndex	Run Process Index
BASE+0x16C	R/W		N/A	RunCodStd	Run Codec Standard
BASE+0x170	R/W		N/A	IntEnable	Interrupt Enable
BASE+0x174	R/W		N/A	IntReason	Interrupt Reason
Protected for Internal Use					
BASE+0x180 ~ BASE + 1D8	R/W		N/A	Command I/O Reg	Command I/O registers

[5] Register from 0x140 to 0x15C are newly added to support maximum eight instances.

[6] A new command for checking F/W version is newly added just for convenience. And new command for sleep/wakeup and encoder parameter change are added.

Table 21-6. DEC_SEQ_INIT Parameter Register Summary

DEC_SEQ_INIT				
	Address	**Type**	Name	Description
INPUT ARGUMENT	BASE+0x180	R/W	CMD_DEC_SEQ_BIT_BUF_START	Bitstream Buffer Address
	BASE+0x184	R/W	CMD_DEC_SEQ_BIT_BUF_SIZE	Bitstream Buffer Size
	BASE+0x188	R/W	CMD_DEC_SEQ_OPTION[7][8]	Decoding sequence option
	BASE+0x18C	R/W	CMD_DEC_SEQ_PRO_BUF	Process Buffer Address
	BASE+0x190	R/W	CMD_DEC_SEQ_TMP_BUF_1	Temporary Buffer1 Address
	BASE+0x194	R/W	CMD_DEC_SEQ_TMP_BUF_2	Temporary Buffer2 Address
	BASE+0x198	R/W	CMD_DEC_SEQ_TMP_BUF_3	Temporary Buffer3 Address
	BASE+0x19C	R/W	CMD_DEC_SEQ_TMP_BUF_4	Temporary Buffer4 Address
	BASE+0x1A0	R/W	CMD_DEC_SEQ_TMP_BUF_5	Temporary Buffer5 Address[9]
	BASE+0x1A4	R/W	CMD_DEC_SEQ_START_BYTE	Start byte of valid stream data
OUTPUT RETURN	BASE+0x1C0	R	RET_DEC_SEQ_SUCCESS	Command executing result status
	BASE+0x1C4	R	RET_DEC_SEQ_SRC_SIZE	Decoded source picture size
	BASE+0x1C8	R	RET_DEC_SEQ_SRC_F_RATE	Decoded source frame rate
	BASE+0x1CC	R	RET_DEC_SEQ_FRAME_NEED	Required minimum decoded frame buffer
	BASE+0x1D0	R	RET_DEC_SEQ_FRAME_DELAY	Maximum display frame buffer delay
	BASE+0x1D4	R	RET_DEC_SEQ_INFO[10]	Decoded sequence information

[7] Bits for representing File Play mode, Dynamic Buffer Allocation enable are added to this register.
[8] Bits for ignoring VUI decoding is added to this register.
[9] From 18C to 1A0, this registers are added to provide a way to use work buffer configurable.
[10] A bit for representing Annex-J indication is added to this register.

Table 21-7. ENC_SEQ_INIT Parameter Register Summary

ENC_SEQ_INIT				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_SEQ_BIT_BUF_START	Bitstream Buffer Address
	BASE+0x184	R/W	CMD_ENC_SEQ_BIT_BUF_SIZE	Bitstream Buffer Size
	BASE+0x188	R/W	CMD_ENC_SEQ_OPTION	Encoding sequence option[11]
	BASE+0x18C	R/W	CMD_ENC_SEQ_COD_STD	Encode Coding Standard
	BASE+0x190	R/W	CMD_ENC_SEQ_SRC_SIZE	Encode Source Frame Size
	BASE+0x194	R/W	CMD_ENC_SEQ_SRC_F_RATE	Encode Source Frame Rate
	BASE+0x198	R/W	CMD_ENC_SEQ_MP4_PARA	Encode MPEG4 Parameter[12]
	BASE+0x19C	R/W	CMD_ENC_SEQ_263_PARA	Encode H.263 Parameter
	BASE+0x1A0	R/W	CMD_ENC_SEQ_264_PARA	Encode H.264 Parameter
	BASE+0x1A4	R/W	CMD_ENC_SEQ_SLICE_MODE	Encode Slice Mode
	BASE+0x1A8	R/W	CMD_ENC_SEQ_GOP_NUM	Encode GOP Number
	BASE+0x1AC	R/W	CMD_ENC_SEQ_RC_PARA	Encode Rate Control Parameter
	BASE+0x1B0	R/W	CMD_ENC_SEQ_RC_BUF_SIZE	Encode Rate Control Buffer Size
	BASE+0x1B4	R/W	CMD_ENC_SEQ_INTRA_MB	Encode Intra MB Refresh Number
	BASE+0x1B8	R/W	CMD_ENC_SEQ_FMO	FMO configuration in H.264 Enc.[13]
	BASE+0x1BC	R/W	CMD_ENC_SEQ_INTRA_QP	I-frame Qp value. This parameter is valid when 5th bit of CMD_ENC_SEQ_OPTION register is set.
	BASE+0x1D0	R/W	CMD_ENC_SEQ_TMP_BUF_1	Temporary Buffer1 Address
	BASE+0x1D4	R/W	CMD_ENC_SEQ_TMP_BUF_2	Temporary Buffer2 Address
	BASE+0x1D8	R/W	CMD_ENC_SEQ_TMP_BUF_3	Temporary Buffer3 Address[14]
	BASE+0x1DC	R/W	CMD_ENC_SEQ_TMP_BUF_4	Temporary Buffer4 Address[15]
OUTPUT	BASE+0x1C0	R	RET_ENC_SEQ_SUCCESS	Command executing result

[11] ConstIntraQpEn bit is added to this register.
[12] HEC enable and Version ID 1 enable bits are added to this register.
[13] This register controls FMO options in H.264 encoder. FMO slice save buffer size is added to this register.
[14] From 1D0 to 1D8, this registers are added to provide a way to use work buffer configurable.
[15] This temporal buffer is dedicated buffer for FMO operation in H.264 encoder

RETURN				status

Table 21-8. DEC_PIC_RUN Parameter Register Summary

DEC_PIC_RUN				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_DEC_PIC_ROT_MODE	Display frame post-rotator mode
	BASE+0x184	R/W	CMD_DEC_PIC_ROT_ADDR_Y	Post-rotated frame store Y address
	BASE+0x188	R/W	CMD_DEC_PIC_ROT_ADDR_CB	Post-rotated frame store CB address
	BASE+0x18C	R/W	CMD_DEC_PIC_ROT_ADDR_CR	Post-rotated frame store CR address
	BASE+0x190	R/W	CMD_DEC_PIC_DBK_ADDR_Y	Deblocked frame store Y address
	BASE+0x194	R/W	CMD_DEC_PIC_DBK_ADDR_CB	Deblocked frame store CB address
	BASE+0x198	R/W	CMD_DEC_PIC_DBK_ADDR_CR	Deblocked frame store CR address
	BASE+0x19C	R/W	CMD_DEC_PIC_ROT_STRIDE	Post-rotated frame stride[16]
	BASE+0x1A8	R/W	CMD_DEC_PIC_CHUNK_SIZE	Frame chunk size
	BASE+0x1AC	R/W	CMD_DEC_PIC_BB_START	4-byte aligned start address of picture stream buffer
	BASE+0x1B0	R/W	CMD_DEC_PIC_START_BYTE	Start byte of valid stream data
OUTPUT RETURN	BASE+0x1C0	R	RET_DEC_PIC_FRAME_NUM	Decoded frame number
	BASE+0x1C4	R	RET_DEC_PIC_IDX	Display frame index
	BASE+0x1C8	R	RET_DEC_PIC_ERR_MB_NUM	Error MB number in decoded picture
	BASE+0x1CC	R	RET_DEC_PIC_TYPE	Decoded picture type
	BASE+0x1D8	R	RET_DEC_PIC_SUCCESS	Command executing result status
	BASE+0x1DC	R	RET_DEC_PIC_CUR_IDX	Decode frame index

[16] This stride register is added to provide a way to use same size frame buffer as rotated output.

Table 21-9. ENC_PIC_RUN Parameter Register Summary

ENC_PIC_RUN				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_PIC_SRC_ADDR_Y	Input source frame buffer Y SDRAM address
	BASE+0x184	R/W	CMD_ENC_PIC_SRC_ADDR_CB	Input source frame buffer CB SDRAM address
	BASE+0x188	R/W	CMD_ENC_PIC_SRC_ADDR_CR	Input source frame buffer CR SDRAM address
	BASE+0x18C	R/W	CMD_ENC_PIC_QS	Encode picture quantization step
	BASE+0x190	R/W	CMD_ENC_PIC_ROT_MODE	Input frame pre-rotator mode
	BASE+0x194	R/W	CMD_ENC_PIC_OPTION	Encode picture option
	BASE+0x198	R/W	CMD_ENC_PIC_BB_START	Start address of picture stream buffer
	BASE+0x19C	R/W	CMD_ENC_PIC_BB_SIZE	Size of picture stream buffer
OUTPUT RETURN	BASE+0x1C0	R	RET_ENC_PIC_FRAME_NUM	Encoded frame number
	BASE+0x1C4	R	RET_ENC_PIC_TYPE	Encoded picture type
	BASE+0x1C8	R	RET_ENC_PIC_IDX	Reconstructed frame index
	BASE+0x1CC	R	RET_ENC_PIC_SLICE_NUM	Encoded slice number of picture
	BASE+0x1D0	R	RET_ENC_PIC_FLAG	Picture encoding status flag

Table 21-10. SET FRAME BUFFER Parameter Register Summary

SET_FRAME_BUF				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_SET_FRAME_BUF_NUM	Frame buffer number to be used by codec
	BASE+0x184	R/W	CMD_SET_FRAME_BUF_STRIDE	Frame Buffer Line Stride
OUTPUT RETURN				

Table 21-11. ENC HEADER Parameter Register Summary

ENC_HEADER				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_HEADER_CODE	Header code to be encoded
	BASE+0x184	R/W	CMD_ENC_HEADER_BB_START	Start address of header syntax stream buffer
	BASE+0x188	R/W	CMD_ENC_HEADER_BB_SIZE	Size of header syntax stream buffer
OUTPUT RETURN				

Table 21-12. DEC PARA SET Parameter Register Summary

DEC_PARA_SET				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_DEC_PARA_SET_TYPE	Sequence/Picture Parameter Set type
	BASE+0x184	R/W	CMD_DEC_PARA_SET_SIZE	Sequence/Picture Parameter Set RBSP byte size
OUTPUT RETURN				

Table 21-13. ENC PARA SET Parameter Register Summary

ENC_PARA_SET				
	Address	**Type**	**Name**	**Description**
INPUT ARGUMENT	BASE+0x180	R/W	CMD_ENC_PARA_SET_TYPE	Sequence/Picture Parameter Set type
OUTPUT RETURN	BASE+0x1C0	R	RET_ENC_PARA_SET_SIZE	Encoded Sequence/Picture Parameter Set RBSP byte size

Table 21-14. MFC_SLEEP Parameter Register Summary

MFC_SLEEP				
	Address	Type	Name	Description
INPUT ARGUMENT				
OUTPUT RETURN				

Table 21-15. MFC_WAKEUP Parameter Register Summary

MFC_WAKEUP				
	Address	Type	Name	Description
INPUT ARGUMENT				
OUTPUT RETURN				

Table 21-16. GET F/W VER Parameter Register Summary

GET_F/W_VER				
	Address	Type	Name	Description
INPUT ARGUMENT				
OUTPUT RETURN	BASE+0x1C0	R	RET_GET_FW_VER	Returned Version Code with following format: [31:16]: Product No (0xF202) [15:0]: Ver. No. (0xMmrr) for M.m.rr

21.5.2.2 Detailed Description of BIT Processor Registers

CodeRun (0x000)

Bit	Name	Type	Function	Reset Value
0	CodeRun	W	0 – BIT Processor stop execution 1 – BIT Processor start execution	0

CodeDownLoad (0x004)

Bit	Name	Type	Function	Reset Value
15:0	CodeData	W	16-bit BIT code download data	0
28:16	CodeAddr	W	13-bit BIT code download address BIT code word address (16-bit address) * Current design has 4 K code word space (8 KB). Therefore CodeAddr[12:0] must be less than 4095	0

HostIntReq (0x008)

Bit	Name	Type	Function	Reset Value
0	IntReq	W	Interrupt request to BIT processor. Host can write ‘1’ to this register to request interrupt to BIT * Current firmware version does not use Interrupt from Host to BIT. Therefore this register is not used	0

BitIntClear (0x00C)

Bit	Name	Type	Function	Reset Value
0	IntClear	W	Writing ‘1’ to this register clear BIT interrupt to host	0

BitIntSts (0x010)

Bit	Name	Type	Function	Reset Value
0	IntSts	R	1 means that BIT interrupts to host is asserted. This bit is cleared when Host writes IntClear Register ‘1’	0

BitCodeReset (0x014)

Bit	Name	Type	Function	Reset Value
0	CodeReset	W	If host write ‘1’ to this register, program counter of BIT is set to “0” Therefore restart at initial routine	0

BitCurPc (0x018)

Bit	Name	Type	Function	Reset Value
13:0	CurPc	R	Current program counter of BIT processor. This register may be used for only debugging purpose	0

CodeBufAddr (0x100)

Bit	Name	Type	Function	Reset Value
31:0	CodeBufAddr	R/W	BIT firmware code image start byte address, which resides in SDRAM. Host must set start SDRAM byte address of BIT code image to this register before start executing BIT processor * Current design uses 80 KB for code buffer	N/A

WorkBufAddr (0x104)

Bit	Name	Type	Function	Reset Value
31:0	WorkBufAddr	R/W	BIT processor working buffer SDRAM byte address. Host must reserve working buffer in SDRAM for BIT processor encoding/decoding.	N/A

ParaBufAddr (0x108)

Bit	Name	Type	Function	Reset Value
31:0	ParaBufAddr	R/W	BIT processor parameter buffer SDRAM byte address. Host must reserve parameter buffer in SDRAM for BIT processor command execution argument and return data. * Current design uses 8 KB for parameter buffer	N/A

BitStreamCtrl (0x10C)

Bit	Name	Type	Function	Reset Value
0	SelBigEndian	R/W	0 – bit stream buffer is 4 byte little endian format 1 – bit stream buffer is big endian	N/A
1	BufStsCheckDis	R/W	0 – bit stream buffer overflow/underflow check enable 1 – bit stream buffer overflow/underflow check disable BIT processor stop bit stream loading if bit stream buffer underflow occurs in decoding case and stop bit stream saving if bit stream buffer overflow occurs in encoding case. If this flag is “1”, BIT does not check bit stream buffer overflow/underflow status.	N/A
2	BufPicFlush	R/W	The value of “1” means that bit stream buffer is flushed at every end of encoding picture. In encoding case, after encoding one picture internal bit stream buffer is flushed to external SDRAM. Therefore entire encoded bit stream data is available to host. If this flag is “0”, internal bit stream buffer is flushed only when internal bit stream buffer is filled to its maximal size (512 byte). Therefore at the end of encoding one picture, the some (less than 512 byte) last encoded data is not flushed to external SDRAM and only resides in internal bit stream buffer. To flush remaining encoded data residing in internal bit stream buffer, host must execute ENC_SEQ_END command. This flag is valid only when [BufPicReset] flag is “0” In decoding case, this flag is ignored.	N/A
3	BufPicReset	R/W	The value of “1” means that bit stream buffer is reset at every picture encoding/decoding command. In encoding case, after encoding one picture bit stream buffer is flushed to external SDRAM and next picture encoded data is over-written to the start of bit stream buffer. Therefore host must get encoded data at every end of encoding picture. If this flag is “1”, [BufPicFlush] bit is ignored. In decoding case, this flag is ignored.	N/A
4	EncDynBufAllocEn	R/W	Enable dynamic picture stream buffer allocation in encoder operations. BufPicReset should also be enabled to use this option. If not, this value will be ignored. When this option is enabled, encoder stream buffer can be dynamically allocated at every picture encoding stage. This option will be helpful to achieve higher efficiency of buffering encoded picture stream.	

FrameMemCtrl (0x110)

Bit	Name	Type	Function	Reset Value
0	SelBigEndian	R/W	0 – frame memory is 4 byte little endian format 1 – frame memory is big endian	N/A

DecFuncCtrl (0x114)

Bit	Name	Type	Function	Reset Value
0	StreamEnd	R/W	0 – There is more bitstream to be given to the decoder. 1 – The whole bitstream has been given to the decoder. In Picture Run state, BIT processor can know the end of bitstream by checking this bit. Host must set this flag after writing the whole bitstream to get the last picture of bitstream. Host also can clear busy state while BIT is waiting for the rest of bitstream corresponding to one picture and get one picture by setting this flag. Host can set/clear this flag at any stage in decoding process after BIT is initialized. Once this bit is set, the decoder will not accept more streams; therefore HOST must clear this flag to 0 before starting decoding and set this flag to 1 after writing the whole bitstream. In Seq init state, this flag is used for signaling to BIT processor to escape from SEQ_INIT stall state. When this bit is set by 1, then BIT processor gives up parsing more stream data and escape from stall state with return value 0 (Fail). This flag is ignored in encoding case.	N/A

BitWorkBufCtrl (0x11C)

Bit	Name	Type	Function	Reset Value
0	WorkBufConfig	R/W	0 – Work Buffer configurable setting disable 1 – Work Buffer configurable setting enable	N/A

BitStreamRdPtr0 (0x120)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr0	R/W	In decode case, current external SDRAM Bit Stream Buffer read address of process index 0 is set to this register by BIT processor. In encode case; host must set current external Bit Stream Buffer read address of process index 0 to this register. This register is updated at every bit stream data load by BIT processor and wrapped around by automatically. * Current design load 512 bytes to internal buffer for each transfer. Therefore Bit Stream Read Pointer is increased to 512 after loading data completion.	N/A

BitStreamWrPtr0 (0x124)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr0	R/W	In decode case; host must set current external Bit Stream Buffer write address of process index 0 to this register. In encode case, current external SDRAM Bit Stream Buffer write address of process index 0 is set to this register by BIT processor. This register is updated at every bit stream data save by BIT processor and wrapped around by automatically. * Current design save 512 bytes from internal buffer for each transfer. Therefore Bit Stream Write Pointer is increased to 512 after saving data completion.	N/A

BitStreamRdPtr1 (0x128)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr1	R/W	External SDRAM Bit Stream Buffer read address of process index 1.	N/A

BitStreamWrPtr1 (0x12C)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr1	R/W	External SDRAM Bit Stream Buffer write address of process index 1.	N/A

BitStreamRdPtr2 (0x130)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr2	R/W	External SDRAM Bit Stream Buffer read address of process index 2.	N/A

BitStreamWrPtr2 (0x134)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr2	R/W	External SDRAM Bit Stream Buffer write address of process index 2.	N/A

BitStreamRdPtr3 (0x138)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr3	R/W	External SDRAM Bit Stream Buffer read address of process index 3.	N/A

BitStreamWrPtr3 (0x13C)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr3	R/W	External SDRAM Bit Stream Buffer write address of process index 3.	N/A

BitStreamRdPtr4 (0x140)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr4	R/W	External SDRAM Bit Stream Buffer read address of process index 4.	N/A

BitStreamWrPtr4 (0x144)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr4	R/W	External SDRAM Bit Stream Buffer write address of process index 4.	N/A

BitStreamRdPtr5 (0x148)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr5	R/W	External SDRAM Bit Stream Buffer read address of process index 5.	N/A

BitStreamWrPtr5 (0x14C)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr5	R/W	External SDRAM Bit Stream Buffer write address of process index 5.	N/A

BitStreamRdPtr6 (0x150)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr6	R/W	External SDRAM Bit Stream Buffer read address of process index 6.	N/A

BitStreamWrPtr6 (0x154)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr6	R/W	External SDRAM Bit Stream Buffer writes address of process index 6.	N/A

BitStreamRdPtr7 (0x158)

Bit	Name	Type	Function	Reset Value
31:0	StreamRdPtr7	R/W	External SDRAM Bit Stream Buffer read address of process index 7.	N/A

BitStreamWrPtr7 (0x15C)

Bit	Name	Type	Function	Reset Value
31:0	StreamWrPtr7	R/W	External SDRAM Bit Stream Buffer writes address of process index 7.	N/A

BusyFlag (0x160)

Bit	Name	Type	Function	Reset Value
0	BusyFlag	R	The value of '0' means BIT processor is ready for host command. The value of '1' means BIT processor is executing host command and not completed yet. Host must check this bit before write RunCommand register. If this bit is '1', host must wait until the value of '0' to set command.	N/A

RunCommand (0x164)

Bit	Name	Type	Function	Reset Value
3:0	RunCommand	R/W	Host writes the command code to this register. Command code 3'b001 (SEQ_INIT): Encode/Decode sequence initialize. In encode case, BIT processor analysis encoding parameter and encode sequence header. In decode case, BIT processor decode sequence header and report sequence header information. 3'b010 (SEQ_END): Terminates Encode/ Decode sequence. In encode case, BIT processor flush internal bit stream buffer to external bit stream buffer. In decode case, BIT processor terminates process 3'b011 (PICURE_RUN): Encode/Decode one picture. BIT processor encodes/decodes one picture. 3'b100 (SET_FRAME_BUF): Set decoded/ reconstructed frame buffer SDRAM address and maximum frame buffer number. Before encode/decode picture run command, host must inform frame buffer SDRAM address to BIT processor then BIT processor arrange frame buffer for decoded/reconstructed image and return frame buffer index to host at end of encoding/decoding picture. 3'b101 (ENCODE HEADER): Encode header. For example in H.264 case, SPS (Sequence Parameter Set), PPS (Picture Parameter Set) may be inserted between picture boundary by this command. 3'b110 (ENC PARA SET): Encode SPS, PPS to BIT processor's parameter set buffer. In H.264, host can obtain SPS / PPS by this command 3'b111 (DEC PARA SET): Add SPS, PPS to BIT processor's parameter set buffer. In H.264, multiple SPS / PPS is allowed and host may inform one of the parameter set to BIT processor for use in decoding process. 4'b1001 Reserved 4'b1010 (MFC_SLEEP): Sleep VPU. After sending this command and checking command done, host must save environment variables related VPU. Ex) host I/F registers, workbuffer memory area data, codebuffer area data and etc. 4'b1011 (MFC_WAKEUP): Wake up VPU form sleep state. Before sending this command, host must download VPU F/W to internal program memory and load data which might be saved after completion of sleep VPU command. Using these commands, host control sleep/wakeup of VPU. 4'b1111 (GET F/W VER): A command to check F/W version. This command is only applicable to F/W version later than 1.2.5.	N/A

RunIndex (0x168)

Bit	Name	Type	Function	Reset Value
1:0	RunIndex	R/W	Host writes the codec process index to this register before every writing run command. BIT processor can execute max 4 encoding/decoding processes simultaneously. If more than one process is running, each process must be assigned different process index by this register. For example, when 1 MPEG4 Decoder + 1 AVC Decoder + 1 AVC Encoder are running simultaneously, MPEG4 Decoder is assigned process index '0', AVC Decoder is assigned process index '1' and AVC Encoder is assigned process index '2'.	N/A

RunCodStd (0x16C)

Bit	Name	Type	Function	Reset Value
2:0	CodStd	R/W	Host writes the codec standard index code to this register before every writing run command 3'b000 : MPEG4/H.263 DECODER 3'b001 : MPEG4/H.263 ENCODER 3'b010 : H.264 DECODER 3'b011 : H.264 ENCODER 3'b100 : VC-1 DECODER	N/A

IntEnable (0x170)

Bit	Name	Type	Function	Reset Value
15:0	IntEnable	R/W	Interrupt Enable Flag register. Each bit of this register is interrupt enable flag of various interrupt. "1" means interrupt enable so BIT generates interrupt and "0" means interrupt disable 0th bit (LSB) : Initialize complete. This interrupt is generated at once after BIT run 1st bit : SEQ_INIT command execution complete 2nd bit: SEQ_END command execution complete 3rd bit : PIC_RUN command execution complete 4th bit: SET_FRAME_BUF command complete 5th bit: ENC_HEADER command complete 6th bit: ENC_PARA_SET command complete 7th bit: DEC_PARA_SET command complete 8th ~ 9th bit: Reserved 10th bit : MFC_SLEEP command complete 11th bit : MFC_WAKEUP command complete 12th ~ 13th bits: Reserved 14th bit: External bit stream buffer is empty status in decoding case 15th bit: External bit stream buffer is full status in encoding case	N/A

IntReason (0x174)

Bit	Name	Type	Function	Reset Value
15:0	IntReason	R/W	Interrupt Reason Flag register. Each bit of this register is interrupt report flag of each interrupt. “1” means interrupt is generated and “0” means not generated. BIT writes “1” to the bit of each interrupt when generates interrupt request and host may acknowledge which interrupt is generated by reading this register at interrupt service routine. Host is responsible for resetting this register to “0” for next interrupt The interrupt matching of each bit field is same with IntEnable register.	N/A

CMD_DEC_SEQ_BIT_BUF_START (0x180)

Bit	Name	Type	Function	Command
31:0	BitBufAddr	R/W	Bitstream buffer SDRAM byte address Bitstream buffer must be 512 byte-aligned. Host must writes this register before executing DEC_SEQ_INIT command	DEC_SEQ_INIT

CMD_DEC_SEQ_BIT_BUF_SIZE (0x184)

Bit	Name	Type	Function	Command
13:0	BitBufSize	R/W	Bitstream buffer size in kilo bytes count Host must writes this register before executing DEC_SEQ_INIT command Maximal bitstream buffer size is 4G byte	DEC_SEQ_INIT

CMD_DEC_SEQ_OPTION (0x188)

Bit	Name	Type	Function	Command
0	Mp4DbkOn	R/W	Enable MPEG4 De-blocking filter If this flag is “1”, MPEG-4 de-blocking filter is enabled and de-blocking filtered image stored to DbkAddrY, DbkAddrCb, DbkAddrCr address. The decoded image (prior to de-blocking filter) is also stored for future motion compensation reference. This flag is valid only for MPEG4 / H.263 case. When H.263 case, if Annex J is turned on, this flag is ignored and H.263+ Annex J de-blocking filter is performed. If Annex J is turned off, MPEG-4 de-blocking filter is enabled (same with MPEG-4)	DEC_SEQ_ INIT
1	ReorderEn	R/W	Enable display buffer reordering in H.264 decode case. In H.264 case output decoded picture may be re-ordered if pic_order_cnt_type is “0” or “1”. In that case, decoder must delay output display for re-ordering but some applications (ex. Video telephony) don’t want such display delay. Host may set this flag to “0” to disable output display buffer reordering. Then BIT processor does not re-order output buffer when pic_order_cnt_type is “0” or “1”. In pic_order_cnt_type is “2” or MPEG4/H.263 case, this flag is ignored because output display buffer reordering is not allowed. If this flag is “1”, BIT processor perform output decoded picture reordering and output display is delayed in the amount of [RET_DEC_SEQ_FRAME_DELAY] register’s value.	
2	FilePlayEn	R/W	Enable file-play mode in decoder operation with frame-based streaming.	
3	DecDynBufAl locEn	R/W	Enable dynamic picture stream buffer allocation in file-play mode If this option is enabled in file-play mode, stream buffer address given with DEC_PIC_RUN command will be used instead of stream buffer address given with DEC_SEQ_INIT. By using this dynamic allocation feature, application can achieve higher efficiency in streaming.	
4	IgnoreVUI	R/W	Enable decoding with ignoring VUI syntax element. If this option is enabled, H.264 decoder will decode streams without decoding VUI syntax. In some case of decoding erroneous stream with invalid VUI, this option will be useful.	

5	Set to 0			
6[17]	VC1_Reoder _Disable	R/W	Re-ordering disable in VC-1 decoding This is for test only.	

[17] This bit is shifted

CMD_DEC_SEQ_PRO_BUF (0x18C)

Bit	Name	Type	Function	Command
31:0	ProcessBufAddr	R/W	Process buffer SDRAM byte address Process buffer must be 256 byte-aligned. Host must write this register before executing DEC_SEQ_INIT command Process buffer is used as PS data save buffer for AVC and MV direct prediction buffer for VC1 and not used for mpeg4. Process buffer must be larger than MB number * 4 for VC1. In case of AVC, process buffer size is not estimated. Therefore, host process allocates some amount temporarily. Process buffer must be maintained before instance closed. Therefore, host processor must allocate process buffer as instance number.	DEC_SEQ_INIT

CMD_DEC_SEQ_TMP_BUF_1 (0x190)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must writes this register before executing DEC_SEQ_INIT command Temporary buffer 1 is used as ACDC prediction buffer for mpeg4 and Intra prediction Y buffer for AVC and ACDC prediction buffer for VC1. Temporary buffer 1 must be larger than picWidth*8 for mpeg4, stride*picHeight/16 for AVC, picWidth*8 for VC1.	DEC_SEQ_INIT

CMD_DEC_SEQ_TMP_BUF_2 (0x194)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must writes this register before executing DEC_SEQ_INIT command Temporary buffer 2 is used as data partition part 1 save buffer for mpeg4 and Intra prediction Cb buffer for AVC and deblocking buffer for VC1. Temporary buffer 2 must be larger than stride/2*picHeight/2/16 for AVC, picWidth*10 for VC1. In case of mpeg4, temporary buffer 2 size is not estimated.	DEC_SEQ_INIT

CMD_DEC_SEQ_TMP_BUF_3 (0x198)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must write this register before executing DEC_SEQ_INIT command Temporary buffer 3 is used as data partition part 2 save buffer for mpeg4 and Intra prediction Cr buffer for AVC and not used for VC1. Temporary buffer 3 must be larger than stride/2*picHeight/2/16 for AVC. In case of mpeg4, temporary buffer 2 size is not estimated.	DEC_SEQ_INIT

CMD_DEC_SEQ_TMP_BUF_4 (0x19C)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must write this register before executing DEC_SEQ_INIT command Temporary buffer 4 is used as slice information save buffer for AVC and not used for VC1 and mpeg4. Max size of temporary buffer 4 is MB number * 8 for AVC.	DEC_SEQ_INIT

CMD_DEC_SEQ_TMP_BUF_5 (0x1A0)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must write this register before executing DEC_SEQ_INIT command Temporary buffer 5 is used as slice save buffer for AVC and not used for VC1 and mpeg4. Max size of temporary buffer 5 may be picwidth*picheight*1.5.	DEC_SEQ_INIT

CMD_DEC_SEQ_START_BYTE (0x1A4)

Bit	Name	Type	Function	Command
1:0	DecSeqValidByteStart	R/W	Byte Address of valid bitstream in input stream buffer	DEC_SEQ_INIT

RET_DEC_SEQ_SUCCESS (0x1C0)

Bit	Name	Type	Function	Command
0	RetStatus	R	0 – DEC_SEQ_INIT command executed with error 1 – DEC_SEQ_INIT command executed successfully.	DEC_SEQ_INIT

RET_DEC_SEQ_SRC_SIZE (0x1C4)

Bit	Name	Type	Function	Command
9:0	PictureHeight	R	Decoded picture height size in pixel	DEC_SEQ_INIT
19:10	PictureWidth	R	Decoded picture width size in pixel	

RET_DEC_SEQ_SRC_F_RATE (0x1C8)

Bit	Name	Type	Function	Command
15:0	FrameRateRes	R	Decoded picture frame rate residual Number of time units of a clock operating at the frequency [FrameRateDiv] Hz For example, [FrameRateDiv] = 30000 and [FrameRateRes] = 1001 then video frame rate = 30000 / 1001 = 29.97 Hz [FrameRateDiv] = 1 and [FrameRateRes] = 15 then video frame rate = 15 / 1 = 15 Hz	DEC_SEQ_INIT
31:16	FrameRateDivMinus1	R	Decoded picture frame rate unit number in Hz minus 1 [FrameRateDiv] is derived by adding this value to 1	

RET_DEC_SEQ_FRAME_NEED (0x1CC)

Bit	Name	Type	Function	Command
4:0	FrameBufNeed	R	Minimum decoded frame buffer need to decode stream successfully. In MPEG4/H.263 case, this value will be 2 (one for motion compensation reference, one for current frame store). In H.264 case, this value may be bigger than 2 and maximal value may be 18 (16 for reference, 1 for current, 1 for display). Host must reserve frame buffer with the amount of minimum this value. In VC-1 case, the rotated output frame is included in this value. Therefore additional rotated frame is not required when rotator is enabled in VC-1. For example, BIT returns [FrameBufNeed] with 5 and host prepares 7 frame buffers, informs frame buffer address by SET_FRAME_BUF command. BIT processor arranges 7 frame buffers and allocates appropriate frame buffer address to decoded image data store. If no MMCO (Memory Management Control Operation) and output reordering, BIT processor will allocate decoded frame buffer 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, … and reference frame data over-writing will not be occurred.	DEC_SEQ_INIT

ET_DEC_SEQ_FRAME_DELAY (0x1D0)

Bit	Name	Type	Function	Command
4:0	FrameBufDelay	R	Maximum display frame buffer delay for buffering decoded picture reorder. BIT processor may delay decoded picture display for display reordering when H.264, pic_order_cnt_type “0” or “1” case or VC-1 decode case. For example, BIT processor return [FrameBufDelay] to “5”. Then BIT processor returns after 5 picture decoding at the first DEC_PIC_RUN command because during first 5 frames, there is no decoded picture to be displayed. Maximum [FrameBufDelay] value may be 16. This value is “0” if [ReorderEn] flag is “0” at H.264 case. In VC-1 decode case, this value is “0” if there is no B picture, “1” if there is a B picture regardless [ReorderEn] flag. In MPEG4/H.263 case, this value will be 0 (no delay)	DEC_SEQ_INIT

RET_DEC_SEQ_INFO (0x1D4)

Bit	Name	Type	Function	Command
0	DataPartEn	R	0 – Data Partition Disable 1 – Data Partition Enable After executing completion of decode DEC_SEQ_INIT command, BIT write to this register with data partition enable flag from decoded sequence header information. In encode case, this register is not used.	DEC_SEQ_INIT
1	RevVlcEn	R	0 – Normal VLC table used 1 – Reversible VLC table used This bit is ignored if DataPartEn bit is ‘0’	
2	ShortVideoHeader	R	0 – Normal MPEG4 Stream 1 – Short Video Header Stream	
3	H.263 Annex J	R	0 – Annex J off 1 – Annex J On	

CMD_ENC_SEQ_BIT_BUF_START (0x180)

Bit	Name	Type	Function	Command
31:0	BitBufAddr	R/W	Bitstream buffer SDRAM byte address Bitstream buffer must be 512 byte-aligned. Host must write this register before executing NC_SEQ_INIT command	ENC_SEQ_INIT

CMD_ENC_SEQ_BIT_BUF_SIZE (0x184)

Bit	Name	Type	Function	Command
13:0	BitBufSize	R/W	Bitstream buffer size in kilo bytes count Host must write this register before executing ENC_SEQ_INIT command Maximal bitstream buffer size is 4G byte.	ENC_SEQ_INIT

CMD_ENC_SEQ_OPTION (0x188)

Bit	Name	Type	Function	Command
0	MbBitReport	R/W	Bit position of every MB is stored to SDRAM buffer. If this flag is “1”, BIT processor store the start bit position of every MB to SDRAM. Host may access this bit position value after encoding one picture. The MB BIT buffer resides in [ParaBufAddr].. The bit position is counted from the start of picture	ENC_SEQ_INIT
1	SliceInfoReport	R/W	Enable encoded Slice number and position in SDRAM bit buffer store to SDRAM buffer. If this flag is “1”, BIT processor store encoded slice end position of every slice to SDRAM. Host may access this slice position after encoding one picture. The encoded slice number is stored to [EncSliceNum] of RET_ENC_PIC_SLICE_NUM register. This flag is ignored for H.263 with Annex K disable because H.263 without Annex K stream has no slice structure.	
2	AUDEnable	R/W	Encode H.264 Access Unit Delimiter RBSP enable If this flag is “1”, BIT encodes Access Unit Delimiter RBSP at every start of picture. Access Unit Delimiter RBSP is used to simplify the detection of the picture boundary This flag is ignored at MPEG4/H.263 encode case.	
3	MbQpReport	R/W	Enable MB QP Store to SDRAM buffer. If this flag is “1”, BIT processor stores QP(Quantization Parameter) of every MB to SDRAM. Host may access this QP value after encoding one picture. The MB QP buffer resides in [ParaBufAddr+0x1300]. This flag is valid only for MPEG-4/H.263 case.	
4	Reserved	-	-	-
5	ConstIntraQp	R/W	Enable I-frame Quantization parameter value setting	

CMD_ENC_SEQ_COD_STD (0x18C)

Bit	Name	Type	Function	Command
1:0	EncCodStd	R/W	Encode Coding Standard 0 – MPEG4 Simple Profile 1 – MPEG4 Short Video Header / H.263+ 2 – H.264 Host must writes this register before executing SEQ_INIT command	ENC_SEQ_INIT

CMD_EMC_SEQ_SRC_SIZE (0x190)

Bit	Name	Type	Function	Command
9:0	PictureHeight	R	Encode source picture height size in pixel Source picture height must be a multiple of 16, less than or equal to 576	ENC_SEQ_INIT
19:10	PictureWidth	R	Encode source picture width size in pixel Source picture width must be a multiple of 16, less than or equal to 720	

CMD_ENC_SEQ_SRC_F_RATE (0x194)

Bit	Name	Type	Function	Command
15:0	FrameRateRes	R	Encode source frame rate residual Number of time units of a clock operating at the frequency [FrameRateDiv] Hz Frame rate = [FrameRateRes] / [FrameRateDiv] If [EncCodStd] = 1 and no Annex (I,J,K,T) turns on (H.263 without PLUSPTYPE), encode source frame rate must be 29.97 ([FrameRateRes] = 30000, [FrameRateDiv] = 1001) because H.263 without PLUSPTYPE supports only 29.97 Hz source frame rate.	ENC_SEQ_INIT
31:16	FrameRateDivMinus1	R	Encode source frame rate unit number in Hz minus 1 [FrameRateDiv] is derived by adding this value to 1	

CMD_ENC_SEQ_MP4_PARA (0x198)

Bit	Name	Type	Function	Command
0	DataPartEn	R/W	0 – Data Partition Disable 1 – Data Partition Enable	ENC_SEQ_INIT
1	RevVlcEn	R/W	0 – Normal VLC table used 1 – Reversible VLC table used This bit is ignored if DataPartEn bit is '0'	
4:2	IntraDcVlcThr	R/W	MPEG4 Intra DC VLC Threshold code The allowed range is [0 ~ 7]	

CMD_ENC_SEQ_263_PARA (0x19C)

Bit	Name	Type	Function	Command
0	Annex T	R/W	0 – Annex T off 1 – Annex T on	ENC_SEQ_INIT
1	Annex K	R/W	0 – Annex K off 1 – nnex K on	
2	Annex J	R/W	0 – Annex J off 1 – Annex J on	
3	Annex I	R/W	0 – Annex I off 1 – Annex I on * Current design does not supports Annex I for encoding mode. Therefore this flag must be set to 0	

CMD_ENC_SEQ_AVC_PARA (0x1A0)

Bit	Name	Type	Function	Command
4:0	ChromaQpOffset	R/W	chroma_qp_index_offset in Picture Parameter set range -12 to +12 2's complement signed 5 bit 1_0100 : -12 1_0101 : -11 … 0_1100 : +12	ENC_SEQ_INIT
5	ConstIntraFlag	R/W	constrained_intra_pred_flag in Picture Parameter set 0 – intra prediction use inter MB data 1 – intra prediction does not use inter MB data	
7:6	DisableDeblk	R/W	disable_deblocking_filter_idc in slice header 0 – enable deblocking filter 1 – disable deblocking filter 2 – enable deblocking filter except slice boundary	
11:8	DeblkAlphaOffset	R/W	slice_alpha_c0_offset_div2 in slice header range -6 to +6 2's complement signed 4 bit	
15:12	DeblkBetaOffset	R/W	slice_beta_offset_div2 in slice header range -6 to +6 2's complement signed 4 bit	

CMD_ENC_SEQ_SLICE_MODE (0x1A4)

Bit	Name	Type	Function	Command
0	SliceMode	R/W	0 – one slice per picture 1 – multiple slices per picture MPEG4 mode, re-sync marker and packet header is inserted between slice boundary H.263 mode with Annex K = 0, GOB header is inserted at every GOB layer start H.263 mode with Annex K = 1, multiple slices are generated H.264 mode, multiple slice layer RBSP is generated	ENC_SEQ_INIT
1	SliceSizeMode	R/W	0 – Slice is changed by encoded slice bit number 1 – Slice is changed by encoded macro-block number This bit is ignored if SliceMode bit is 0 In H.263 Mode with Annex K = 0, this bit is ignored	
15:2	SliceSizeNum	R/W	If SliceSizeMode is 0, macro-block number of one slice must be set to this register If SliceSizeMode is 1, encoded bit count of one slice must be set to this register This bit is ignored if SliceMode bit is 0 In H.263 Mode with Annex K = 0, this bit is ignored	

CMD_ENC_SEQ_GOP_NUM (0x1A8)

Bit	Name	Type	Function	Command
5:0	EncGopNum	R/W	Encode GOP number I picture is inserted at every GOP picture number Maximum GOP number is 60. 0 – I, P, P, P, … (only first picture is I) 1 – I, I, I, … (no P picture) 2 – I, P, I, P, … 3 – I, P, P, I, P, P, I, …	ENC_SEQ_INIT

CMD_ENC_SEQ_RC_PARA (0x1AC)

Bit	Name	Type	Function	Command
0	RcEnable	R/W	Rate Control Enable If this flag is set to 0, the value of register PictureQs is used as a Quantization Step in whole sequence.	ENC_SEQ_INIT
15:1	BitRate	R/W	Target Bit Rate in kilo bit per seconds (kbps) This value is ignored if RcEnable = 0 Maximum allowed value is 32767 (0x7FFF)	
30:16	InitDelay	R/W	Reference Decoder initial buffer removal delay in mili-second (ms) This value is ignored if RcEnable = 0 Maximum allowed value is 32767 (0x7FFF) 0 : do not check Reference decoder buffer delay constraint	
31	SkipDisable	R/W	Rate control automatic skip disable If this flag is “0”, BIT processor may skip one picture if available bits are insufficient for accommodate the bit budget. If this flag is “1”, BIT processor never skip the picture but encoded bitstream may be overflow than target bit rate at hard-to-encode sequences. This flag is ignored if [RcEnable] is “0”.	

CMD_ENC_SEQ_RC_BUF_SIZE (0x1B0)

Bit	Name	Type	Function	Command
31:0	VbvBufSize	R/W	Reference Decoder buffer size in bits This value is ignored if RcEnable = 0 or InitDelay = 0 Maximum allowed value is 0x7FFF_FFFF 0 : do not check Reference decoder buffer size constraint	ENC_SEQ_INIT

CMD_ENC_SEQ_INTRA_MB (0x1B4)

Bit	Name	Type	Function	Command
15:0	IntraMbRefreshNum	R/W	Intra MB refresh number Must be less than encoded (PictureHeight*PictureWidt/256) 0 – Intra MB refresh is not used N – At least N number of MBs are encoded as Intra mode at every picture	ENC_SEQ_INIT

CMD_ENC_SEQ_FMO (0x1B8)

Bit	Name	Type	Function	Command
0	FmoEnable	R/W	0: FMO disable 1: fmo enable	ENC_SEQ_INIT
4:1	FmoSliceNr	R/W	Number of Slice Groups (It must be a value between 2 and 8.	
5	FmoType[18]	R/W	0: Type0 (interleaved) 1: Type1 (Dispersed)	
23:8	FmoSliceBufSize	R/W	Fmo slice save work buffer size. This bits means work buffer size for 1 slice in KB size. So, if this bits value is 2, total fmo slice work buffer is 2*8=16KB.	

CMD_ENC_INTRA_QP (0x1BC)

Bit	Name	Type	Function	Command
31:0	IntraQp	R/W	Intra frame picture quantized step parameter for encoding process. In MPEG-4/H.263 mode, allowed range is 1 to 31. In H.264 mode, allowed range is 0 to 51. If rate control disabled, this value is ignored.	ENC_SEQ_INIT

CMD_ENC_SEQ_TMP_BUF_1 (0x1D0)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must write this register before executing ENC_SEQ_INIT command. Temporary buffer 1 is used as ACDC prediction Buffer for Mpeg4 and Intra Prediction Y buffer for AVC. Temporary buffer 1 must be larger than picwidth*8 for mpeg4, stride*picheight/16 for AVC.	ENC_SEQ_INIT

[18] We now provide two FMO types, interleaved (Type 0) and dispersed (Type 1).

CMD_ENC_SEQ_TMP_BUF_2 (0x1D4)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must write this register before executing ENC_SEQ_INIT command. Temporary buffer 2 is used as data partition part 2 save Buffer for Mpeg4 and Intra Prediction Cb buffer for AVC. Temporary buffer 2 must be larger than (stride/2)*(picheight/2)/8 for AVC. In case of mpeg4, temporary buffer 2 size is not estimated.	ENC_SEQ_INIT

CMD_ENC_SEQ_TMP_BUF_3 (0x1D8)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 256 byte-aligned. Host must write this register before executing ENC_SEQ_INIT command Temporary buffer 3 is used as data partition part 3 save Buffer for Mpeg4 and Intra Prediction Cr buffer for AVC. Temporary buffer 3 must be larger than (stride/2)*(picheight/2)/8 for AVC. In case of mpeg4, temporary buffer 3 size is not estimated.	ENC_SEQ_INIT

CMD_ENC_SEQ_TMP_BUF_4 (0x1DC)

Bit	Name	Type	Function	Command
31:0	TempBufAddr	R/W	Temporary buffer SDRAM byte address Temporary buffer must be 4 byte-aligned. Host must write this register before executing ENC_SEQ_INIT command Temporary buffer 4 is used when FMO option is enabled in H.264 encoder. The size of this buffer must be given by (32 KB * FmoSliceNr). For example, if the number of slice group is 8, this buffer size should be 32x8 KB.	ENC_SEQ_INIT

RET_ENC_SEQ_SUCCESS (0x1C0)

Bit	Name	Type	Function	Command
0	RetStatus	R	0 – ENC_SEQ_INIT command executed with error 1 – ENC_SEQ_INIT command executed successfully.	ENC_SEQ_INIT

CMD_DEC_PIC_ROT_MODE (0x180)

Bit	Name	Type	Function	Command
3:0	PostRotMode	R/W	Post rotation mode PostRotMode[3:0] = {HorMir, VerMir, RotAng[1:0]} HorMir : Horizontal mirroring VerMir : Vertical mirroring RotAng[1:0] 0 : 0 degree counterclockwise rotate 1 : 90 degree counterclockwise rotate 2 : 180 degree counterclockwise rotate 3 : 270 degree counterclockwise rotate If this field is 4'b0000, post rotation is enabled but just copy the decoded image	DEC_PIC_RUN
4	PostRotEn	R/W	Post rotation enable If this field is "1", the rotated image is stored to DecPicRotAddrY, DecPicRotAddrCb, DecPicRotAddrCr address addition to decoded image store for future reference. If this field is "0", the post rotation is disabled and PostRotMode field is ignored.	

CMD_DEC_PIC_ROT_ADDR_Y (0x184)

Bit	Name	Type	Function	Command
31:0	DecRotAddrY	R/W	Rotated display frame address of luminance If PostRotEn field is "1", the rotated image is stored to this address In VC-1 mode, this register is not used. The rotated output will be one of the frames which previously allocated by RET_DEC_SEQ_FRAME_NEED(0x1CC).	DEC_PIC_RUN

CMD_DEC_PIC_ROT_ADDR_CB (0x188)

Bit	Name	Type	Function	Command
31:0	DecRotAddrCb	R/W	Rotated display frame address of Cb In VC-1 mode, this register is not used. The rotated output will be one of the frames which previously allocated by RET_DEC_SEQ_FRAME_NEED(0x1CC).	DEC_PIC_RUN

CMD_DEC_PIC_ROT_ADDR_CR (0x18C)

Bit	Name	Type	Function	Command
31:0	DecRotAddrCr	R/W	Rotated display frame address of Cr In VC-1 mode, this register is not used. The rotated output will be one of the frames which previously allocated by RET_DEC_SEQ_FRAME_NEED(0x1CC).	DEC_PIC_RUN

CMD_DEC_PIC_DBK_ADDR_Y (0x190)

Bit	Name	Type	Function	Command
31:0	DecDbkAddrY	R/W	Deblocked display frame address of luminance If Mp4DbkOn field is “1”, the deblocked image is stored to this address	DEC_PIC_RUN

CMD_DEC_PIC_DBK_ADDR_CB (0x194)

Bit	Name	Type	Function	Command
31:0	DecDbkAddrCb	R/W	Deblocked display frame address of Cb	DEC_PIC_RUN

CMD_DEC_PIC_DBK_ADDR_CR (0x198)

Bit	Name	Type	Function	Command
31:0	DecDbkAddrCr	R/W	Deblocked display frame address of Cr	DEC_PIC_RUN

CMD_DEC_PIC_ROT_STRIDE (0x19C)

Bit	Name	Type	Function	Command
10:0	DecRotStride	R/W	Rotated display frame Stride	DEC_PIC_RUN

CMD_DEC_PIC_CHUNK_SIZE (0x1A8)

Bit	Name	Type	Function	Command
31:0	DecPicChunkSize	R/W	Byte size of picture stream data This value will be only valid in file-play mode, and it will be used as size of stream data when BIT processor update write pointer of picture stream buffer.	DEC_PIC_RUN

CMD_DEC_PIC_BB_START (0x1AC)

Bit	Name	Type	Function	Command
31:0	DecPicBitBufStart	R/W	4-byte aligned byte address of the decoder input picture stream buffer This value will only be valid if decoder dynamic buffer allocation option is enabled as well as file-play mode option. In this case, this address will be used as the start address of the bitstream data when BIT processor update write pointer of picture stream buffer.	DEC_PIC_ RUN

CMD_DEC_PIC_START_BYTE (0x1B0)

Bit	Name	Type	Function	Command
1:0	DecPicValidByteStart	R/W	Byte Address of valid bitstream in input picture stream buffer	DEC_PIC_ RUN

RET_DEC_PIC_FRAME_NUM (0x1C0)

Bit	Name	Type	Function	Command
15:0	DecFrameNum	R/W	Decoded frame number. After BIT decodes one frame, BIT increase frame number and then stores frame number to this register.	DEC_PIC_ RUN

RET_DEC_PIC_IDX (0x1C4)

Bit	Name	Type	Function	Command
15:0	DecPicIdx	R/W	Display frame index After BIT decodes one frame, BIT return display frame index to this register. The frame index is the index of array of frame buffer address that host informs by SET_FRAME_BUF command If BIT return -1(0xFFFF), it means that all pictures of given bitstream have already been decoded. For example, if host has given a bitstream containing 20 pictures, host will get -1 value from 21th PICTURE_RUN. In file play mode, BIT return -3(0xFFFD) when BIT does not have a picture to be displayed. For example, BIT does not have a picture to be displayed before decodes maximum 16 pictures when BIT decodes H.264 stream with reorderEn bit enable. And BIT may not have picture to be displayed before decodes 2 picture when BIT decodes VC-1 streams which have B-frames.	DEC_PIC_ RUN

RET_DEC_PIC_ERR_MB_NUM (0x1C8)

Bit	Name	Type	Function	Command
15:0	ErrMbNum	R/W	Error MB number in current decoded picture If BIT recognizes the stream error, BIT performs error concealment in MB basis and returns concealed MB number in whole picture. If this value is “0”, the frame is decoded with no error.	DEC_PIC_RUN

RET_DEC_PIC_TYPE (0x1CC)

Bit	Name	Type	Function	Command
0	DecPicType	R/W	The picture type of current decoded picture. 0 – I (Intra) picture 1 – P (Inter) picture.	DEC_PIC_RUN

RET_DEC_PIC_SUCCESS (0x1D8)

Bit	Name	Type	Function	Command
0	RetStatus	R	0 – DEC_PIC_RUN command executed with error 1 – DEC_PIC_RUN command executed successfully.	DEC_PIC_RUN
1	Invalid PPS	R	Invalid PPS This bit is set when there is no valid PPS in given stream. This bit is set only in file play mode and valid when RetStatus bit is 0.	

RET_DEC_PIC_CUR_IDX (0x1DC)

Bit	Name	Type	Function	Command
15:0	DecodedPicIdx	R/W	Decoded frame index After BIT decodes one frame, BIT return decoded frame index to this register. The frame index is the index of array of frame buffer address that host informs by SET_FRAME_BUF command. BIT returns -1 (0xFFFF), if BIT doesn’t decode picture at this picture run command.	DEC_PIC_RUN

CMD_ENC_PIC_SRC_ADDR_Y (0x180)

Bit	Name	Type	Function	Command
31:0	SrcAddrY	R/W	Encoding source frame address of luminance Host must write this register before executing ENC_PIC_RUN command Host must set SDRAM frame buffer start address to this register before every encoding picture	ENC_PIC_RUN

CMD_ENC_PIC_SRC_ADDR_CB (0x184)

Bit	Name	Type	Function	Command
31:0	SrcAddrCb	R/W	Encoding source frame address of Cb	ENC_PIC_RUN

CMD_ENC_PIC_SRC_ADDR_CR (0x188)

Bit	Name	Type	Function	Command
31:0	SrcAddrCr	R/W	Encoding source frame address of Cr	ENC_PIC_RUN

CMD_ENC_PIC_QS (0x18C)

Bit	Name	Type	Function	Command
31:0	PictureQs	R/W	Picture quantized step parameter for encoding process. In MPEG4/H.263 mode, allowed range is 1 to 31. In H.264 mode, allowed range is 0 to 51. If rate control is enabled, this register is ignored. If rate control is disabled, BIT encodes whole MBs in current picture with this value. Host may apply its own picture-level rate control algorithm by regulating this value picture-by-picture basis.	ENC_PIC_RUN

CMD_ENC_PIC_ROT_MODE (0x190)

Bit	Name	Type	Function	Command
3:0	PreRotMode	R/W	Pre-rotation mode PreRotMode[3:0] = {HorMir, VerMir, RotAng[1:0]} HorMir : Horizontal mirroring VerMir : Vertical mirroring RotAng[1:0] 0 : 0 degree counterclockwise rotate 1 : 90 degree counterclockwise rotate 2 : 180 degree counterclockwise rotate 3 : 270 degree counterclockwise rotate If this field is 4'b0000, pre-rotation is disabled.	ENC_PIC_RUN
4	PreRotEn	R/W	Pre-rotation enable If this field is "1", the source image is rotated prior to encoding, If this field is "0", the pre-rotation is disabled and PreRotMode field is ignored.	

CMD_ENC_PIC_OPTION (0x194)

Bit	Name	Type	Function	Command
0	PicSkipEn	R/W	Picture skip flag If this field is “1”, EncSrcAddrY, EncSrcAddrCb, EncSrcAddrCr are ignored and one skipped picture is encoded. In that case, the reconstructed image at decoder side is a copy of previous picture The skipped picture is encoded as P type (Inter) picture regardless of [EncGopNum]. Host may set this field as “1” when next source frame to be encoded is not available For example of encoding frame rate 5 Hz case, if camera output is not available at 4th picture, host must set ENC_PIC_RUN command 5 times during one seconds and set PicSkipEn flag “0, 0, 0, 1, 0”	ENC_PIC_RUN
1	IdrPic	R/W	If this field is “1”, the source image is encoded as IDR(Instantaneous Decoding Refresh) picture at H.264 or I(Intra) picture at MPEG-4/H.263 regardless of [EncGopNum] value The IDR picture is I(Intra) picture with zero frame_num value and all of decoding status(ex. reference picture list) are reset. The first frame in bit stream is encoded IDR picture automatically. After encoding IDR picture, I picture period calculation is reset to initial state. For example, if host set [IdrPic] flag set 18th frame and [EncGopNum] is 15, encoded picture types are 1st frame : I (IDR - automatically) 2nd frame : P ... 14th frame : P 15th frame : I 16th frame : P 17th frame : P 18th frame : I (IDR – set by host) 19th frame : P ... 32nd frame : I 33rd frame : P ... In MPEG-4/H.263 case, I(Intra) picture is sufficient for decoder refresh Host must set this filed as “1” periodically for inserting decoder refresh point in encoded bit stream.	

CMD_ENC_PIC_BB_START (0x198)

Bit	Name	Type	Function	Command

15:0	EncPicBitBufStart	R/W	Byte address of the encoder output picture stream buffer This value will only be valid if encoder dynamic buffer allocation option is enabled as well as stream buffer reset option. In this case, this address will be used as the start address of the bitstream data.	ENC_PIC_RUN

CMD_ENC_PIC_BB_SIZE (0x19C)

Bit	Name	Type	Function	Command
15:0	EncPicBitBufSize	R/W	Byte size of encoded picture stream buffer This value will only be valid if encoder dynamic buffer allocation option is enabled as well as stream buffer reset option. In this case, it will be used as the pointer of the end of picture stream.	ENC_PIC_RUN

RET_ENC_PIC_FRAME_NUM (0x1C0)

Bit	Name	Type	Function	Command
15:0	EncFrameNum	R/W	Encoded frame number. After BIT encodes one frame, BIT increase frame number and then stores frame number to this register.	ENC_PIC_RUN

RET_ENC_PIC_TYPE (0x1C4)

Bit	Name	Type	Function	Command
0	EncPicType	R/W	The picture type of current encoded picture. 0 – I (Intra) picture 1 – P (Inter) picture.	ENC_PIC_RUN

RET_ENC_PIC_IDX (0x1C8)

Bit	Name	Type	Function	Command
15:0	RecPicIdx	R/W	Reconstructed frame index After BIT encodes one frame, BIT return reconstructed frame index to this register Reconstructed frame is used for reference of future frame	ENC_PIC_RUN

RET_ENC_PIC_SLICE_NUM (0x1CC)

Bit	Name	Type	Function	Command
14:0	EncSliceNum	R/W	If [SliceInfoReport] flag in CMD_ENC_SEQ_OPTION register, BIT returns encoded slice number to this register. The encoded slice end position of every slice is stored to SDRAM.	ENC_PIC_RUN

RET_ENC_PIC_FLAG (0x1D0)

Bit	Name	Type	Function	Command
0	BitstreamBufferWrapArounded	R/W	This bit is set if BitstreamBuffer wrptr is wrap arounded in reset mode. If this bit set, bitstream generated might be crashed.	ENC_PIC_RUN

CMD_SET_FRAME_BUF_NUM (0x180)

Bit	Name	Type	Function	Command
4:0	FrameBufNum	R/W	Number of frames used for reference or output reodering. This value must be equal or greater than [FrameBufNeed] in RET_DEC_SEQ_FRAME_NEED register at decoding case and “2” at encoding case Host must set the associated frame buffer SDRAM address (Y, Cb, Cr) to SDRAM buffer of address [ParaBufAddr]	SET_FRAME_BUF

CMD_SET_FRAME_BUF_STRIDE (0x184)

Bit	Name	Type	Function	Command
10:0	LineStride	R/W	Line stride offset of picture frame memory Stride number is byte count	SET_FRAME_BUF

CMD_ENC_HEADER_CODE (0x180)

Bit	Name	Type	Function	Command
2:0	HeaderCode	R/W	Encode header code In MPEG4, 3'b000 – VOL header 3'b001 – VOS header 3'b010 – VIS header In H.264, 3'b000 – SPS rbsp 3'b001 – PPS rbsp In H.263, ENC_HEADER command is ignored	ENC_HEADER

CMD_ENC_HEADER_BB_START (0x184)

Bit	Name	Type	Function	Command
15:0	EncHdrBitBufStart	R/W	Byte address of the encoder output header syntax buffer This value will only be valid if encoder dynamic buffer allocation option is enabled as well as stream buffer reset option. In this case, this address will be used as the start address of the bitstream data.	ENC_HEADER

CMD_ENC_HEADER_BB_SIZE (0x188)

Bit	Name	Type	Function	Command
15:0	EncHdrBitBufSize	R/W	Byte size of encoded header stream buffer This value will only be valid if encoder dynamic buffer allocation option is enabled as well as stream buffer reset option. In this case, it will be used as the pointer of the end of picture stream.	ENC_HEADER

CMD_DEC_PARA_SET_TYPE (0x180)

Bit	Name	Type	Function	Command
0	DecParaSetType	R/W	Parameter set type 0 – Sequence Parameter Set 1 – Picture Parameter Set	DEC_PARA_SET

CMD_DEC_PARA_SET_SIZE (0x184)

Bit	Name	Type	Function	Command
8:0	DecParaSetSize	R/W	Sequence/Picture parameter set RBSP byte size Maximum allowed RBSP size is 511 byte	DEC_PARA_SET

CMD_ENC_PARA_SET_TYPE (0x180)

Bit	Name	Type	Function	Command
0	EncParaSetType	R/W	Parameter set type 0 – Sequence Parameter Set 1 – Picture Parameter Set	ENC_PARA_SET

RET_ENC_PARA_SET_SIZE (0x1C0)

Bit	Name	Type	Function	Command
8:0	EncParaSetSize	R/W	Encoded Sequence/Picture parameter set RBSP byte size	ENC_PARA_SET

RET_VER_NUM (0x1C0)

Bit	Name	Type	Function	Command
31:16	ProductID	R/W	Product ID with 16bit hexacode	GET_FW_V
15:0	VersionID	R/W	Version ID with 16 bit hexacode	ERSION

22 JPEG CODEC

This chapter describes the functions and usage of JPEG CODEC in S3C6410X.

22.1 OVERVIEW

JPEG is a commonly used method of compression for photographic images. The name JPEG stands for **Joint Photographic Experts Group**, the name of the committee that created the standard. The JPEG standard specifies both the codec, which defines how an image is compressed into a stream of bytes and decompressed back into an image, and the file format used to contain that stream.

The JPEG codec core is composed of control circuit, DCT/quantization, Huffman coder, marker process block, and AHB slave interface control as shown in Figure 22-1. Both input/output image data bus and compressed data bus are 8-bits. It has control registers inside. It is possible to set the operation modes, specify the Huffman table number and DRI value into these registers.

22.2 FEATURE

The JPEG CODEC includes the following features:

- Compression/decompression up to (4096 x 4096).
- Encoding Output format : YCbCr4:2:2 or YCbCr4:2:0
- Decoding Input format : YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0 or Gray.
- Support the compression of memory data in YCbCr4:2:2(interleaving only) or RGB565 format.(Input raw data)
- Support the general-purpose color converter.

22.3 BLOCK DIAGRAM

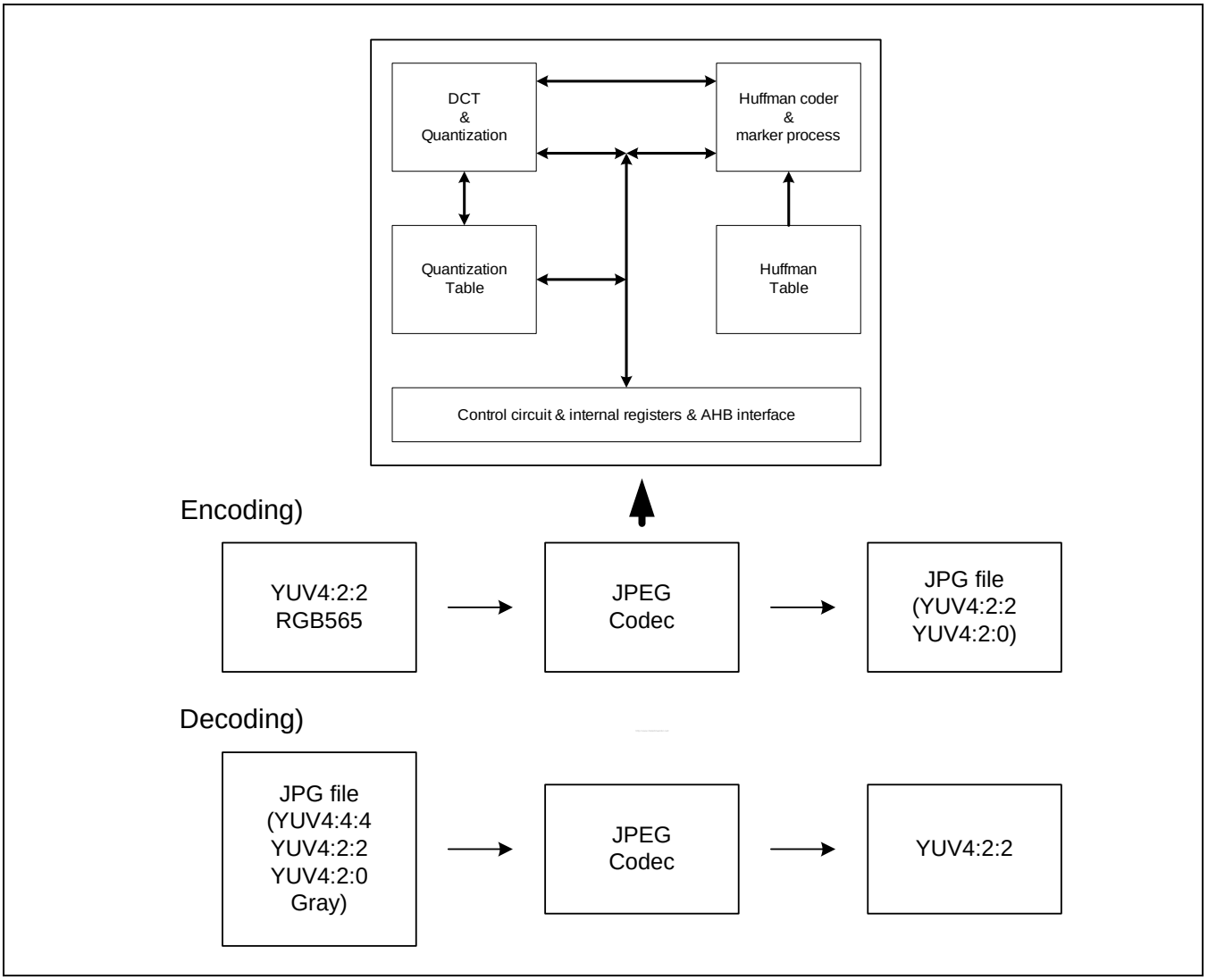


Figure 22-1. JPEG Codec Block Diagram

22.4 FUNCTIONAL DESCRIPTIONS

22.4.1 CONTROL CIRCUIT & AHB INTERFACE

This block sets and initializes the operation mode. It consists of in built registers. It is possible to set the operation modes, quantify the Huffman table number and DRI value into these registers.

22.4.2 DCT/QUANTIZATION

During encoding, JPEG Codec transforms the 8x8 image data to DCT coefficients. Then the quantization process is performed over the DCT coefficients by utilizing the quantization tables. During decoding, dequantization will be done and then DCT coefficients will be transformed into the image data.

22.4.3 HUFFMAN CODER AND MARKER PROCESS

Variable-length encoding and decoding are done based on Huffman table.

22.4.4 QUANTIZATION TABLE

It is the place to store quantization table. This is the RAM area, where users can allocate.

22.4.5 HUFFMAN TABLE

It is the place to store Huffman tables; this is the RAM area where users can allocate

22.4.6 REGISTER ACCESS

The registers can be modified:

1. After reset, until a new job starts.

   Or

2. After process completion interruption signal is generated, until a new job starts.

Other conditions indicate that the core is in the normal operation, therefore no modification is allowed. For some registers, either writing or reading is prohibited.

22.4.7 TABLE ACCESS

Four Huffman tables (AC & DC, 2 tables each) and four quantizer tables must be configured before compression. To set any Quantizer table and Huffman table, the corresponding entry register must be accessed first. Then burst write transfers must follow. For better understanding of the burst write transfer, refer to Figure 22-2. 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	29	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 22-2. Access Order in Quantizer Table

22.4.8 INTERRUPT SIGNAL

Interrupt signal will be generated under the following conditions (the register, **JPGIRQ**, which identifies causes)

1. Compression or decompression process for one frame is completed,
2. **JPGIRQ[3]** is set high during decompression when the registers storing the image size and sampling factor are ready to be read out after the marker analysis.

For interrupt 1, the normal process is finished. To clear the pending interrupt request, read the **JPGSTS** register. If there no encoding or decoding error, **JPGIRQ** will be read as 0x40.

Canceling the interrupt 2 is also done by reading **JPGIRQ**. If there is no header parsing error, it will be read as 0x08. The interrupt 2 indicates that the decompression process is paused.

3-interrupt signals (which are ‘result status, bit stream parsing error and header parsing status) are used to distinguish interrupt occurred circumstance.

22.4.9 INTERRUPT SETTING REGISTER

This register sets whether the interrupt is allowed or not, when it is ready to decompress an image. To allow the interrupt, set **JPGIRQS**[3] to high, before starting the decompression process. When this interrupt occurs, JPEG codec pauses the process and drives **JPGSTS**[0] high as it is. Cancellation of the interrupt by this register setting is done by reading **JPGIRQ.**

22.4.10 MARKER PROCESS

The following markers are generated during compression.

Table 22-1. Markers at JPEG CODEC

Marker	Codes(Hex)	Description
SOI	FFD8	Start of image
SOF0	FFC0	Baseline DCT
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 Table 22-1 are subject to process during decompression. The other markers except SOF1~SOFF and JPG will be ignored.

22.4.1 BITSTREAM OF COMPRESSED FILE

The created JPEG Bit Stream is shown below.

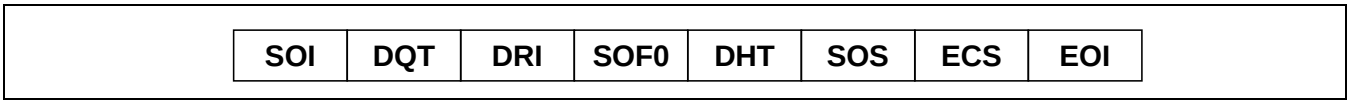


Figure 22-3. Bit stream of JPEG file Block Diagram

22.5 PROGRAMMER'S MODEL

Process start command instructs to start the encoding or decoding process of one frame, after setting various registers. It is set by writing 0x1 on the **SW_JSTART** register. Then, the core processing will start and then **JPGSTS** will be high. Operation can't be guaranteed when this command is issued again during processing. Do not set like that.

Table 22-2. Registers that must be configured before start processing

Register	Description	At Encoding process	At Decoding process
JPGMOD	Process mode register	Essential	Essential
JPGQHNO	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 entry register.	Essential	--
QTBL1	Quantizer table1 entry register	Essential	--
QTBL2	Quantizer table2 entry register	Essential	--
QTBL3	Quantizer table3 entry 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	--

NOTE: "Essential" means that the each register has to be configured.

The contents of any register in Table 22-2 will not be changed unless they are written again or they are reset Therefore it is possible to process the next frame by only performing the process start command, after a frame process is completed. It is restarted by writing 0x1 on the **SW_JSTART** register.

22.6 JPEC CODEC PROGRAMMING GUIDE

JPEG engine has its own internal pending flag that is cleared by reading a **JPGIRQ** by software.

If all processes are finished, read **JPGSTS** to clear all internal interrupt pending flags.

22.6.1 BASIC JPEG ENCODING SEQUENCE

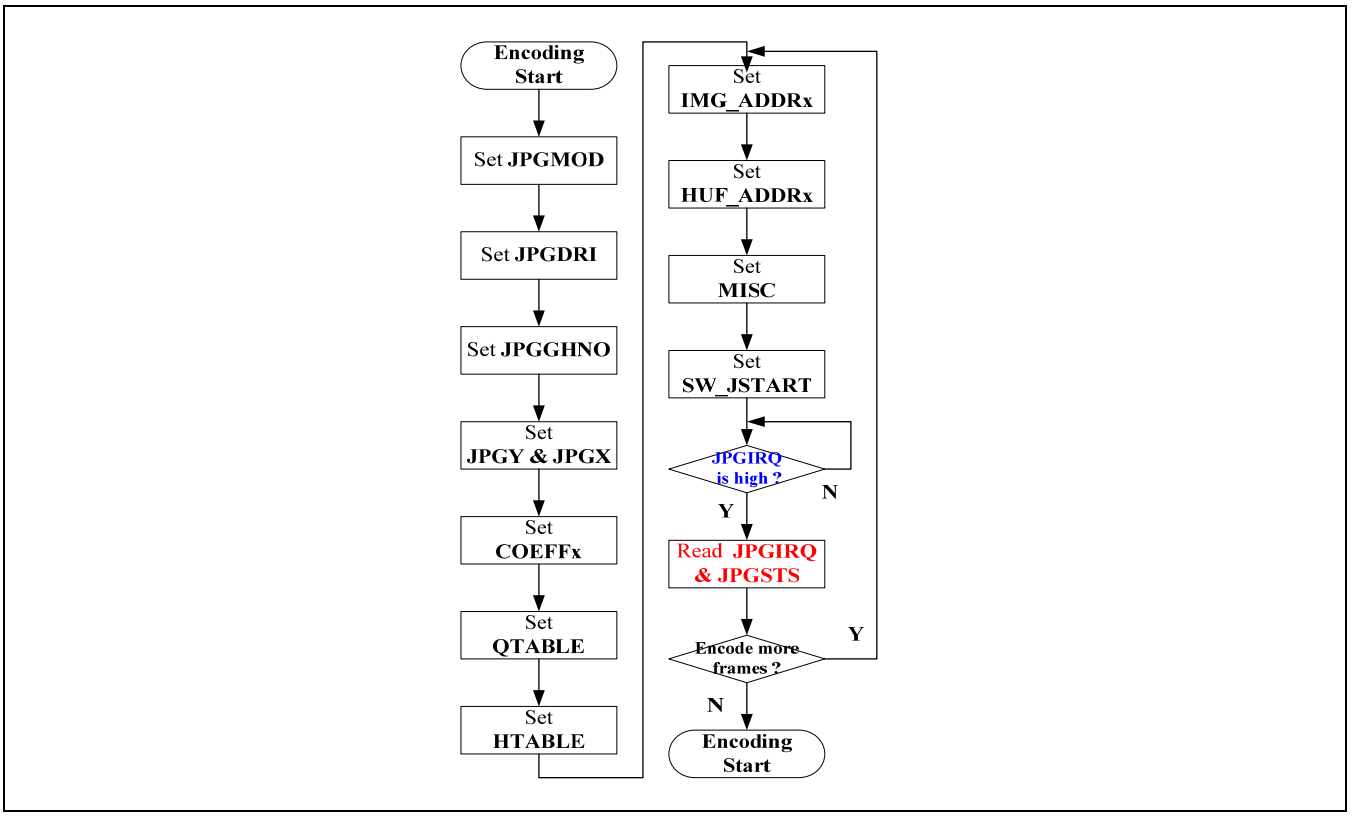


Figure 22-4. Example flow chart of basic encoding

Take the following steps for basic JPEG encoding:

1. Set the process mode to encoding process and sub-sampling mode in **JPGMOD.**
2. Set the MCU and RST marker register **JPGDRI.**
3. Set the Q and H table number register **JPGQHNO.**
4. Set the **JPGY** and **JPGX** registers.
5. Set the Coefficient register **COEFF1, COEFF2, COEFF3** for color space conversion.
6. Set the **QTABLE** and **HTABLE.**
7. Set the source image data of 1st frame Address register **IMG_ADDR0.**
8. Set the destination JPEG file Address register **HUFADDR0.**
9. Set the Miscellaneous register **MISC**
10. Set the **SW_JSTART** to high.
11. You must read **JPGIRQ** and **JPGSTS** registers to clear internal pended IRQs.

22.6.2 JPEG DECODING SEQUENCE

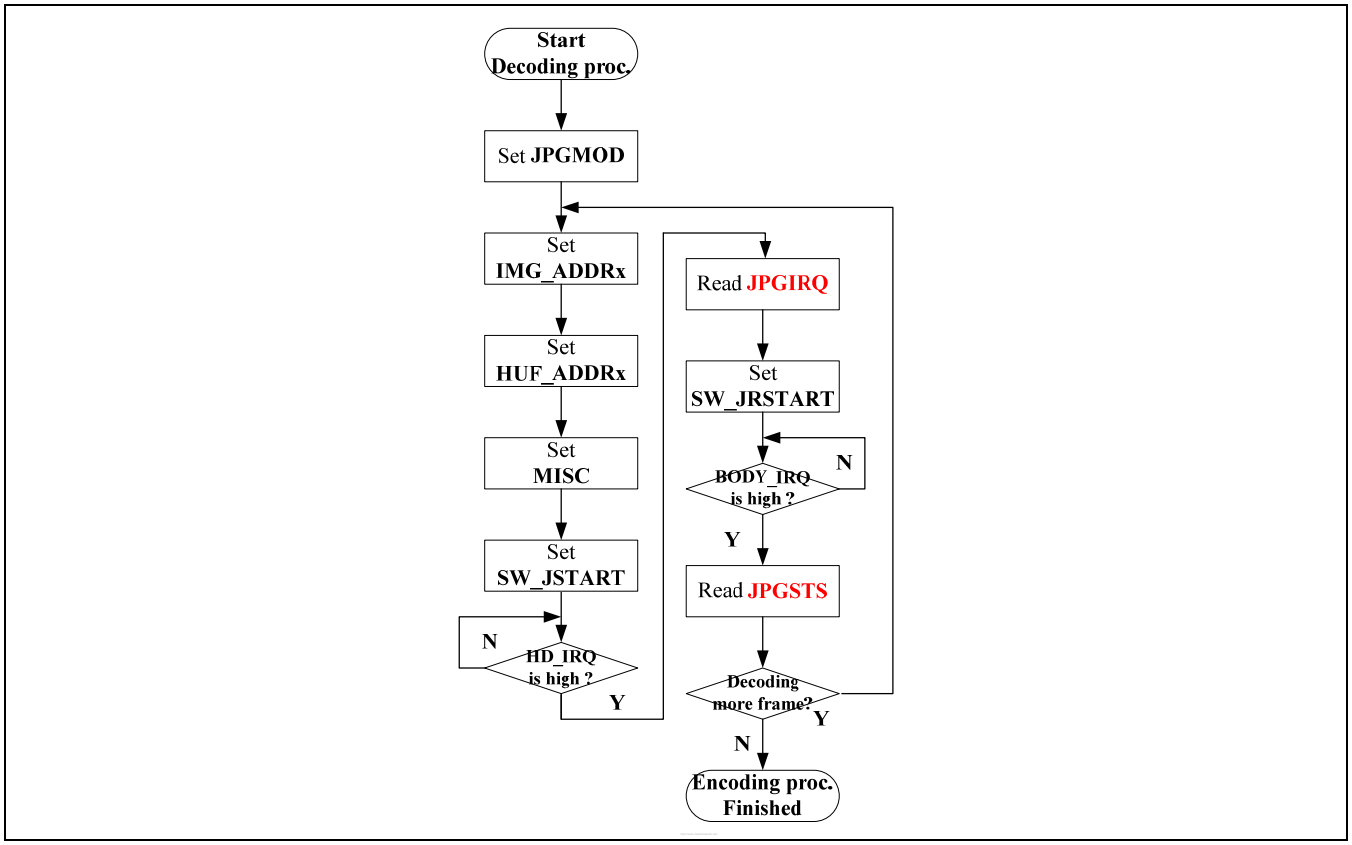


Figure 22-5. Example flow chart of software controlled decoding

Take the following steps for software controlled JPEG decoding:

1. Set the process mode to decoding process in **JPGMOD**
2. Set the destination address of 1st decoded image data I**MG_ADDR0**.
3. Set the source address of 1st JPEG file **HUFADDR0.**
4. Set the Miscellaneous register **MISC (set MODE_SEL to 0x1 or 0x2).**
5. Set the **SW_JSTART** to high.
6. If HD_IRQ is high and ERR_IRQ is low, read **JPGIRQ** register to clear internal pended IRQ**.**
7. Set the **SW_JRSTART** to high.
8. If MAIN_IRQ is high and ERR_IRQ is low, read frame size (in byte) from **JPGCNT** register
9. You must read **JPGIRQ** and **JPGSTS** registers to clear internal pended IRQs.

22.7 JPEC CODEC SPECIAL REGISTERS

Table 22-3. Register summary of JPEG Codec

Register	Address	R/W	Description	Reset Value
JPGMOD	0x78800000	R/W	Process mode register.	0x00000000
JPGSTS	0x78800004	R	Operation status registers.	0x00000000
JPGQHNO	0x78800008	R/W	Quantization table number register and Huffman table number register.	0x00000000
JPGDRI	0x7880000C	R/W	MCU, which inserts RST marker.	0x00000000
JPGY	0x78800010	R/W	Vertical resolution.	0x00000000
JPGX	0x78800014	R/W	Horizontal resolution	0x00000000
JPGCNT	0x78800018	R	The amount of the compressed data in bytes	-
JPGIRQS	0x7880001C	R/W	Interrupt setting register	0x00000000
JPGIRQ	0x78800020	R	Interrupt status register	-
QTBL0	0x78800400 ~ 0x788004FC	W	8-bit Quantization of table number 0 (64 data with the distance of 4 on address)	-
QTBL1	0x78800500 ~ 0x788005FC	W	8-bit Quantization of table number 1 (64 data with the distance of 4 on address)	-
QTBL2	0x78800600 ~ 0x788006FC	W	8-bit Quantization of table number 2 (64 data with the distance of 4 on address)	-
QTBL3	0x78800700 ~ 0x788007FC	W	8-bit Quantization of table number 3 (64 data with the distance of 4 on address)	-
HDCTBL0	0x78800800 ~ 0x7880083C	W	The number of code per code length (16 data with the distance of 4 on address)	-
HDCTBLG0	0x78800840 ~ 0x7880086C	W	Group number of the order for occurrence (12 data with the distance of 4 on address)	-
HACTBL0	0x78800880 ~ 0x788008BC	W	The number of code per code length (16 data with the distance of 4 on address)	-
HACTBLG0	0x788008C0 ~ 0x78800B44	W	Group number of the order for occurrence/Group number (162 data with the distance of 4 on address)	-

Register	Address	R/W	Description	Reset Value
HDCTBL1	0x78800C00 ~ 0x78800C3C	W	The number of code per code length (16 data with the distance of 4 on address) 8-bits register	-
HDCTBLG1	0x78800C40 ~ 0x78800C6C	W	Group number of the order for occurrence (12 data with the distance of 4 on address) 8-bits register	-
HACTBL1	0x78800C80 ~ 0x78800C8C	W	The number of code per code length (16 data with the distance of 4 on address) 8-bits register	-
HACTBLG1	0x78800CC0 ~ 0x78800F44	W	Group number of the order for occurrence /Group number (162 data with the distance of 4 on address) 8-bits register	-
IMG_ADDR0	0x78801000	R/W	Source or destination image address 0	0x00000000
IMG_ADDR1	0x78801004	R/W	Source or destination image address 1	0x00000000
HUFADDR0	0x78801008	R/W	Source or destination JPEG file address 0	0x00000000
HUFADDR1	0x7880100C	R/W	Source or destination JPEG file address 1	0x00000000
SW_JSTART	0x78801010	R/W	Start JPEG process	0x00000000
SW_JRSTART	0x78801014	R/W	Restart JPEG process	0x00000000
SW_RESET_CON	0x78801018	R/W	SW Reset JPEG	0x00000001
COEF1	0x78801020	R/W	Coefficient values for RGB ↔ YCbCr converter	0x00000000
COEF2	0x78801024	R/W	Coefficient values for RGB ↔ YCbCr converter	0x00000000
COEF3	0x78801028	R/W	Coefficient values for RGB ↔ YCbCr converter	0x00000000
MISC	0x7880102C	R/W	Miscellaneous	0x00000000

22.7.1 JPGMOD

Register	Address	Description	Reset Value
JPGMOD	0x78800000	Process mode register.	0x00000000

JPGMOD	Bit	R/W	Description
Reserved	[31:4]	-	Reserved
Process_Mode	[3]	R/W	Process mode. 0: Encoding Process. 1: Decoding Process.
Sub_sampling _Mode	[2:0]	R/W	Sub sampling Mode During decoding process, this is reading only. And it means following as 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 prohibited. During encoding process, this is readable and writable. And it only supports following as, 0x1: chroma 4:2:2 format. 0x2: chroma 4:2:0 format Others are prohibited.

22.7.2 JPGSTS

Register	Address	Description	Reset Value
JPGSTS	0x78800004	Operation status registers.	0x00000000

JPGSTS	Bit	R/W	Description
Reserved	[31:1]	-	Reserved
Status	[0]	R	Process mode. It is low (or ‘0’) while JPEG engine is running. Otherwise, it is high (or ‘1’) whether operation has normally finished or errors have occurred.

22.7.3 JPGQHNO

Register	Address	Description	Reset Value
JPGQHNO	0x78800008	Quantization table number register and Huffman table number register.	0x00000000

JPGQHNO	Bit	R/W	Description
Reserved	[31:14]	-	Reserved
Q_table_num3	[13:12]	R/W	Quantization table number for the 3rd color component.
Q_table_num2	[11:10]	R/W	Quantization table number for the 2nd color component
Q_table_num1	[9:8]	R/W	Quantization table number for the 1st color component.
Reserved	[7:6]	R/W	Reserved.
H_table_num3_ac	[5]	R/W	Huffman table number for the 3rd color component AC.
H_table_num3_dc	[4]	R/W	Huffman table number for the 3rd color component DC.
H_table_num2_ac	[3]	R/W	Huffman table number for the 2nd color component AC.
H_table_num2_dc	[2]	R/W	Huffman table number for the 2nd color component DC.
H_table_num1_ac	[1]	R/W	Huffman table number for the 1st color component AC.
H_table_num1_dc	[0]	R/W	Huffman table number for the 1st color component DC.

NOTE: When Gray mode, let the 1st to 3rd color components have the same value. (Both Quantization tables number and Huffman table number). There are three Q table and each of three ac/dc table. The table is selected when it is high. The first of all table is automatically updated by JPEG stream while decoding is running.

22.7.4 JPGDRI

Register	Address	Description	Reset Value
JPGDRI	0x7880000C	MCU, which inserts RST marker.	0x00000000

JPGDRI	Bit	R/W	Description
Reserved	[31:16]	-	Reserved
Restart_Interval	[15:0]	R/W	It is the reset interval that identifies the distance between two adjacent RST markers in terms of MCU

NOTE: It is valid only in compression. When JPGDRI is set zero, DRI and RST markers will not be inserted

22.7.5 JPGY

Register	Address	Description	Reset Value
JPGY	0x78800010	Vertical resolution.	0x00000000

JPGY	Bit	R/W	Description
Reserved	[31:16]	-	Reserved
VSIZE	[15:0]	R/W	It defines the image size value in the vertical direction. It is only valid during encoding mode. Note : It has limitation of MCU size. Refer to application note.

22.7.6 JPGX

Register	Address	Description	Reset Value
JPGX	0x78800014	Horizontal resolution	0x00000000

JPGX	Bit	R/W	Description
Reserved	[31:16]	-	Reserved
HSIZE	[15:0]	R/W	It defines the image size value in the horizontal direction. It is only valid during encoding mode. Note : It has limitation of MCU size. Refer to application note.

22.7.7 JPGCNT

Register	Address	Description	Reset Value
JPGCNT	0x78800018	The amount of the compressed data in bytes	0x00000000

JPGCNT	Bit	R/W	Description
Reserved	[31:24]	-	Reserved
B_COUNT	[23:0]	R	It defines the byte of compressed data count in the width of **24** bits.

22.7.8 JPGIRQS

Register	Address	Description	Reset Value
JPGIRQS	0x7880001C	Interrupt setting register	0x00000000

JPGIRQS	Bit	R/W	Description
Reserved	[31:4]	-	Reserved
Irq_enable	[3]	R/W	Interrupt enable control. 0: Disable to read the image size and sampling factor value in the result of compressed data analysis during decompression. 1: Enable to read the image size and sampling factor value in the result of compressed data analysis during decompression.
Irq_control	[2:0]	R/W	Write 0x0 to set interrupt.

22.7.9 JPGIRQ

Register	Address	Description	Reset Value
JPGIRQ	0x78800020	Interrupt status register	0x00000000

JPGIRQ	Bit	R/W	Description
Reserved	[31:7]	-	Reserved
Result_status	[6]	R	Result status. 0: The processing was finished abnormally. 1: The processing was done normally.
Reserved	[5]	-	Reserved
Bitstre_err_status	[4]	R	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.
Header_status	[3]	R	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.
Reserved	[2:0]	-	Reserved

NOTE: The value of this register will be clear after reading.

22.7.10 QTBL0

Register	Address	Description	Reset Value
QTBL0	0x78800400 ~ 0x788004FC	8-bit Quantization of table number 0 (64 data with the distance of 4 on address)	-

QTBL0	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
Q_val0	[7:0]	W	It defines the quantizer table 0. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.11 QTBL1

Register	Address	Description	Reset Value
QTBL1	0x78800500 ~ 0x788005FC	8-bit Quantization of table number 1 (64 data with the distance of 4 on address)	-

QTBL1	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
Q_val1	[7:0]	W	It defines the quantizer table 1. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.12 QTBL2

Register	Address	Description	Reset Value
QTBL2	0x78800600 ~ 0x788006FC	8-bit Quantization of table number 2 (64 data with the distance of 4 on address)	-

QTBL2	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
Q_val2	[7:0]	W	It defines the quantizer table 2. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.13 QTBL3

Register	Address	Description	Reset Value
QTBL3	0x78800700 ~ 0x788007FC	8-bit Quantization of table number 3 (64 data with the distance of 4 on address)	-

QTBL3	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
Q_val3	[7:0]	W	It defines the quantizer table 3. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.14 HDCTBL0

Register	Address	Description	Reset Value
HDCTBL0	0x78800800 ~ 0x7880083C	The number of code per code length (16 data with the distance of 4 on address)	-

HDCTBL0	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_DC_val0	[7:0]	W	It defines the number of code per code length in DC Huffman table 0. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.15 HDCTBLG0

Register	Address	Description	Reset Value
HDCTBLG0	0x78800840 ~ 0x7880086C	Group number of the order for occurrence (12 data with the distance of 4 on address)	-

HDCTBLG0	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_DC_G_val0	[7:0]	W	It defines the group number of the order for occurrence in DC Huffman table 0. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.16 HACTBL0

Register	Address	Description	Reset Value
HACTBL0	0x78800880 ~ 0x788008BC	The number of code per code length (16 data with the distance of 4 on address)	-

HACTBL0	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_AC_val0	[7:0]	W	It defines the number of code per code length in AC Huffman table 0. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.17 HACTBLG0

Register	Address	Description	Reset Value
HACTBLG0	0x788008C0 ~ 0x78800B44	Group number of the order for occurrence/Group number (162 data with the distance of 4 on address)	-

HACTBLG0	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_AC_G_val0	[7:0]	W	It defines the group number of the order for occurrence in AC Huffman table 0. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.18 HDCTBL1

Register	Address	Description	Reset Value
HDCTBL1	0x78800C00 ~ 0x78800C3C	The number of code per code length (16 data with the distance of 4 on address) 8-bits register	-

HDCTBL1	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_DC_val1	[7:0]	W	It defines the number of code per code length in DC Huffman table 1. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.19 HDCTBLG1

Register	Address	Description	Reset Value
HDCTBLG1	0x78800C40 ~ 0x78800C6C	Group number of the order for occurrence (12 data with the distance of 4 on address) 8-bits register	-

HDCTBLG1	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_DC_G_val1	[7:0]	W	It defines the group number of the order for occurrence in DC Huffman table 1. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.20 HACTBL1

Register	Address	Description	Reset Value
HACTBL1	0x78800C80 ~ 0x78800CBC	The number of code per code length (16 data with the distance of 4 on address) 8-bits register	-

HDCTBL1	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_AC_val1	[7:0]	W	It defines the number of code per code length in AC Huffman table 1. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.21 HACTBLG1

Register	Address	Description	Reset Value
HACTBLG1	0x78800CC0 ~ 0x78800F44	Group number of the order for occurrence /Group number (162 data with the distance of 4 on address) 8-bits register	-

HDCTBLG1	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
H_AC_G_val1	[7:0]	W	It defines the group number of the order for occurrence in AC Huffman table 1. The user must write some value in this.

NOTE: Address offset is increased as 0x04(word addressing).

22.7.22 IMGADDR0

Register	Address	Description	Reset Value
IMGADDR0	0x78801000	Source or destination image address 0	0x0000_0000

IMGADDR0	Bit	R/W	Description
Image_addr0	[31:0]	R/W	Source or destination Image address 0 It is source during encoding. Otherwise, it is destination during decoding

22.7.23 IMGADDR1

Register	Address	Description	Reset Value
IMGADDR1	0x78801004	Source or destination image address 1	0x0000_0000

IMGADDR1	Bit	R/W	Description
Image_addr1	[31:0]	R/W	Source or destination Image address 1 It is source during encoding. Otherwise, it is destination during decoding

IMGADDR0 and **IMGADDR1** are the start address of an image data. The data before compression are read from this address on encode mode, and decompressed data are stored from this address. It is needed when **MISC**[7:5] is 0x1 or 0x2.

If process (encoding or decoding) is ended, **IMG_ADDR0** or **IMG_ADDR1** are swapped.

22.7.24 HUFADDR0

Register	Address	Description	Reset Value
HUFADDR0	0x78801008	Source or destination JPEG file address 0	0x0000_0000

HUFADDR0	Bit	R/W	Description
Huff_addr0	[31:0]	R/W	Source or destination JPEG file address 0. It is source during decoding. Otherwise, it is destination during encoding

22.7.25 HUFADDR1

Register	Address	Description	Reset Value
HUFADDR1	0x7880100C	Source or destination JPEG file address 1	0x0000_0000

HUFADDR1	Bit	R/W	Description
Huff_addr1	[31:0]	R/W	Source or destination JPEG file address 1 It is source during decoding. Otherwise, it is destination during encoding

HUFADDR0 and **HUFADDR1** are the start address for JPEG data. Both encode mode and decode mode, JPEG data are stored or read from this address.

If process (encoding or decoding) is ended, **HUFADDR0** or **HUFADDR1** are swapped**.**

22.7.26 SW_JSTART

Register	Address	Description	Reset Value
SW_JSTART	0x78801010	Start JPEG process	0x0000_0000

SW_JSTART	Bit	R/W	Description
Reserved	[31:1]	-	Reserved
S_JSTART	[0]	R/W	0: Don't start JPEG process. 1: Start JPEG process

22.7.27 SW_JRSTART

Register	Address	Description	Reset Value
SW_JRSTART	0x78801014	Restart JPEG process	0x0000_0000

SW_JRSTART	Bit	R/W	Description
Reserved	[31:1]	-	Reserved
S_JRSTART	[0]	R/W	This value is available in decoding process.(This bit is valid when HW_DEC is high.) 0: Don't start main decoding process. 1: Start main decoding process.

JPEG Decoding Process in S3C6410X **S_JSTART** command initiate JPEG decoding process and if JPEG engine get header information of JPEG file, (this mean high HD_IRQ value are detected) **S_JRSTART** command is set high to start main decoding process.

22.7.28 SW_RESET_CON

Register	Address	Description	Reset Value
SW_RESET_CON	0x78801018	SW Reset JPEG	0x0000_0001

SW_RESET_CON	Bit	R/W	Description
Reserved	[31:1]	-	Reserved
S_RESET	[0]	R/W	0: Soft Reset enable. 1: Soft Reset disable.

If S_RESET flag is low, all registers except **SW_RESET_CON** are soft-reset. So, it has to reset when JPEG is initialized

22.7.29 OEF1

Register	Address	Description	Reset Value
COEF1	0x78801020	Coefficient values for RGB ↔ YCbCr converter	0x0000_0000

COEF1	Bit	R/W	Description
Reserved	[31:24]	-	Reserved
COEF11	[23:16]	R/W	Coefficient value of COEF11
Reserved	[15:8]	-	Reserved
COEF13	[7:0]	R/W	Coefficient value of COEF13

22.7.30 COEF2

Register	Address	Description	Reset Value
COEF2	0x78801024	Coefficient values for RGB ↔ YCbCr converter	0x0000_0000

COEF2	Bit	R/W	Description
Reserved	[31:24]	-	Reserved
COEF21	[23:16]	R/W	Coefficient value of COEF21
COEF22	[15:8]	R/W	Coefficient value of COEF22
COEF23	[7:0]	R/W	Coefficient value of COEF23

22.7.31 COEF3

Register	Address	Description	Reset Value
COEF3	0x78801028	Coefficient values for RGB ↔ YCbCr converter	0x0000_0000

COEF3	Bit	R/W	Description
Reserved	[31:24]	-	Reserved
COEF31	[23:16]	R/W	Coefficient value of COEF21
COEF32	[15:8]	R/W	Coefficient value of COEF22
Reserved	[7:0]	R/W	Reserved

The calculation in YCbCr converter is based on the following matrix. YCbCr converter is used when **SWSEL** = 0x2 (RGB565) on encode mode.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} COEF11 & 0 & COEF13 \\ COEF21 & COEF22 & COEF23 \\ COEF31 & COEF32 & 0 \end{pmatrix} * \begin{pmatrix} Y - c1 \\ Cb - 128 \\ Cr - 128 \end{pmatrix}$$

Figure 22-6. Relationship matrix between RGB and YCbCr format

22.7.32 MISC

Register	Address	Description	Reset Value
MISC	0x7880102C	Miscellaneous	0x0000_0000

MISC	Bit	R/W	Description
Reserved	[31:8]	-	Reserved
MODE_SEL	[7:5]	R/W	Mode selector 0x1 : Memory (YCbCr4:2:2) 0x2 : Memory (RGB 565) Others are prohibited.
Reserved	[4:2]	-	Reserved. It has to be zero.
MODE_Y16	[1]	R/W	MODE_Y16 0 : c1 = 0 1 : c1 = 16 Refer to the explanation of COEF11 – COEF33
Reserved	[0]	-	Reserved

23 MODEM INTERFACE

23.1 OVERVIEW

This specification defines the interface between the Base-band Modem and the Application Processor for the data-exchange of these two devices (Refer Figure 23-). For the data-exchange, the AP (Application Processor, S3C6410X) has a DPSRAM(Dual Port SRAM) buffer (on-chip) and the Modem chip can access that DPSRAM buffer using a typical asynchronous-SRAM interface.

The size of the SRAM buffer is 8KByte. For the buffer status and Interrupt Requests, this specification also specifies a few of pre-defined special addresses.

The Modem chip can write data in the data buffer and write interrupt control-data to the interrupt-port address for the interrupt request to the AP. The AP reads that data when an interrupt request is accepted and the interrupt is cleared when the AP accesses the interrupt-port address. In the same manner, the AP can write data in the data buffer and write interrupt control-data to the interrupt-port address for the interrupt request to the Modem chip.

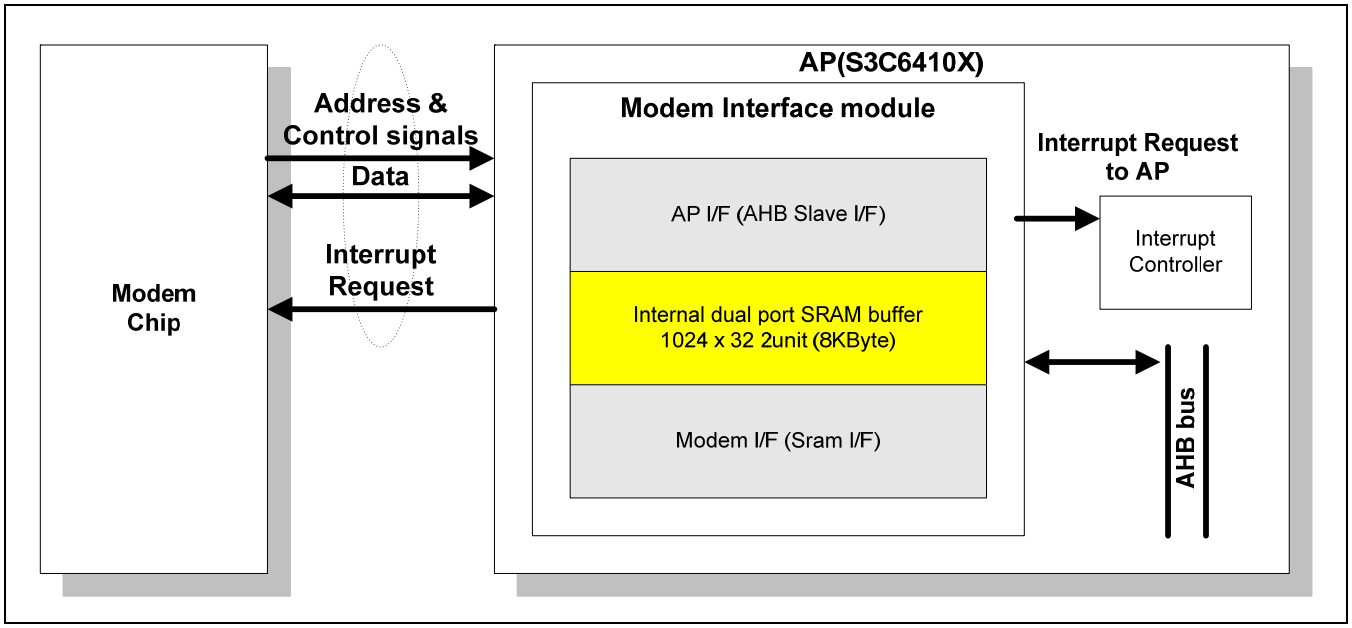


Figure 23-1. The interface with the Modem chip and the MODEM I/F block diagram

23.2 FEATURES

- Asynchronous SRAM interface style interface
- 16-bit parallel bus for data transfer
- 8K bytes internal dual-port SRAM buffer
- Interrupt request for data exchange
- Programmable interrupt port address
- Support from 1.8V to 3.3V I/O voltage range
- AP Booting for Modem procedure provides a dual-port memory as a Modem boot memory.

23.3 EXTERNAL INTERFACES

Name	Type	Description
XhiADR[12:0]	Input	Address bus, driven by the Modem chip
XhiCSn	Input	Chip select, driven by the Modem chip
XhiCSn_main	Input	Chip select for LCD bypass main, driven by the Modem chip
XhiCSn_sub	Input	Chip select for LCD bypass sub, driven by the Modem chip
XhiWEn	Input	Write enable, driven by the Modem chip
XhiOEn	Input	Read enable, driven by the Modem chip
XhiDATA[17:0]	Input/Output	Data bus, driven by the Modem chip (Note : XhiDATA[17:16] are used only LCD Bypass mode)
XhiINTR	Output	Interrupt request to the Modem chip

23.4 MEMORY MAP

Register	Address	R/W	Description	Reset Value
MSBM	0x74100000 ~ 0x74101FFF	R/W	MODEM I/F SRAM buffer memory (AP side)	0x00000000

Register	Address	R/W	Description	Reset Value
INT2AP	0x74108000	R/W	Interrupt request to AP Register	0x00001FFE
INT2MODEM	0x74108004	R/W	Interrupt request to MODEM modem Register	0x00001FFC
MIFCON	0x74108008	R/W	Modem Interface Control Register	0x00000008
MIFPCON	0x7410800C	R/W	Modem Interface Port Control register	0x00000008
MODEMINTCLR	0x74108010	W	Modem Interface Pending Interrupt Request Clear	-

23.5 INTERRUPT PORTS

Interrupts are requested or cleared if the Modem chip or the AP accesses the interrupt-port (predefined special addresses). That special address can be configured by the AP and the default address-map is shown in the Table23-1.

Table 23-1. Interrupt request and clear conditions

Interrupt	Address[1)]	An Interrupt is requested, when	The Interrupt is cleared, when
To AP	0x1FFE	Modem chip writes	AP writes to MODEMINTCLR register in MODEM interface block [2)]
To Modem	0x1FFC	AP writes (at least one bit “High”) or after AP Initialization process [3)]	Modem chip writes High to the corresponding bits [4)]

NOTES:

1) This address is a default value. It can be set to the other value by the SFR (INT2AP register INT2AP_ADR field, INT2MODEM register INT2MODEM_ADR field).

2) Modem interface block has one Interrupt Clear Registers; MODEMINTCLR. 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.

3) There are 3 cases that draw out AP initialization process. They are H/W reset, STOP mode wakeup, and SLEEP mode wakeup. After AP initialization process has been occurred, the system clock would be stable and the reset signal would be released. Besides, AP makes an interrupt signal (active low) to the modem. This automatic interrupt makes a bit[0] of he ‘INT2MODEM_REG’ high. So, a bit[0] of the ‘INT2MODEM_REG’ should not be written by interrupt handler.

4) The level of XhiINTR sustains low because AP writes an interrupt signal to the modem right after system reset. So, modem should clear AP to Modem interrupt first in order to communicate each other.

Modem chip or AP(S3C6410X) can read 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.

23.6 AP BOOTING FOR MODEM

AP Booting for MODEM means that AP provides the boot area (8Kbyte memory) for MODEM.

MODEM can boot using internal Dual-Port SRAM memory inside MODEM_IF block. In this case, AP should provide **'MODEM Reset pin'** and AP should complete boot operation. Also, AP should download the MODEM boot code into its Dual-Port SRAM. If the chip select of the MODEM is connected to 'XhiCSn', MODEM can boot without its external Boot Memory (NAND).

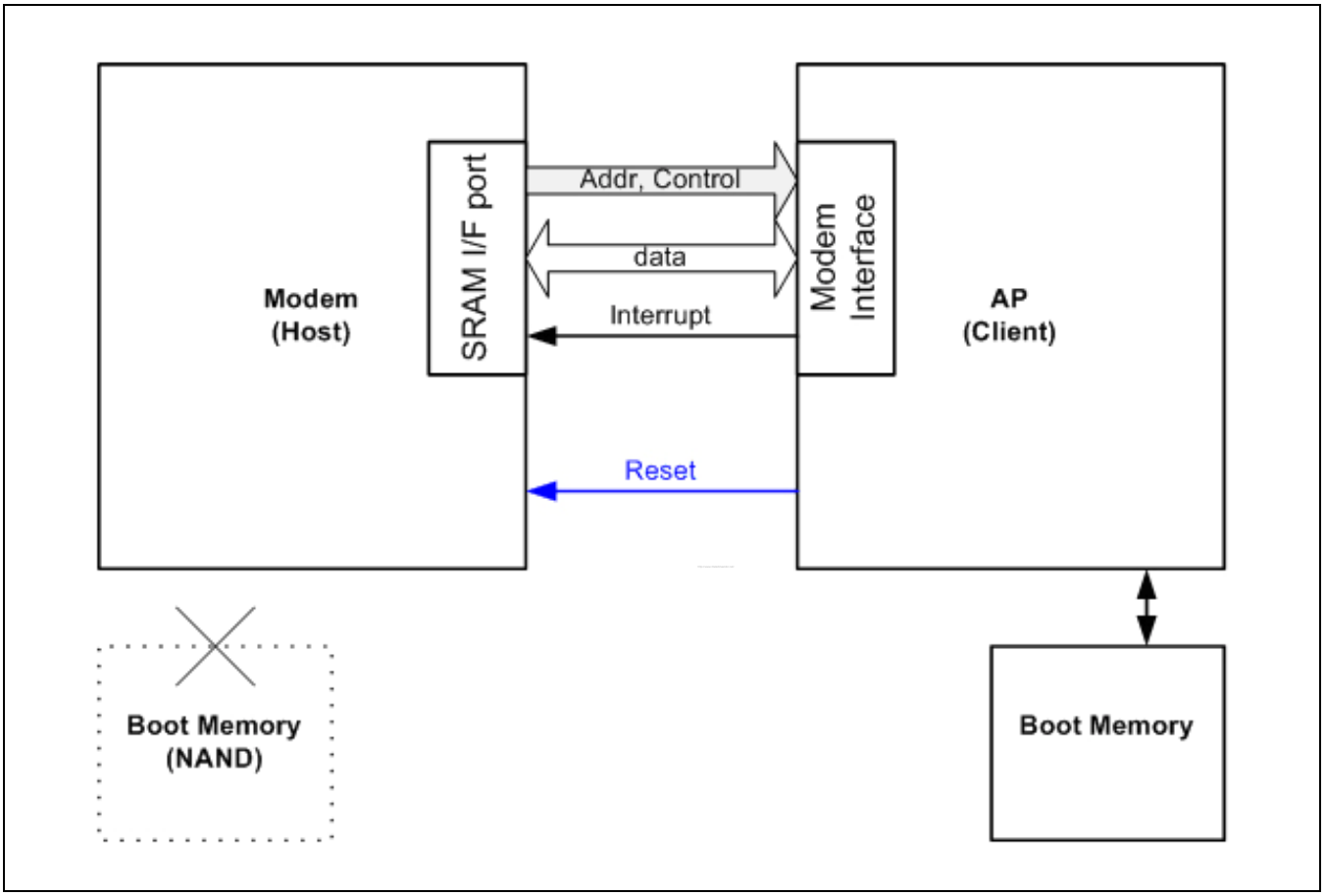


Figure 23-2. MODEM interface for AP Booting for MODEM

In this mode, AP should control the "H/W Reset pin of the Modem" with AP's GPIO output port. This port should be pulled-down at initial state.

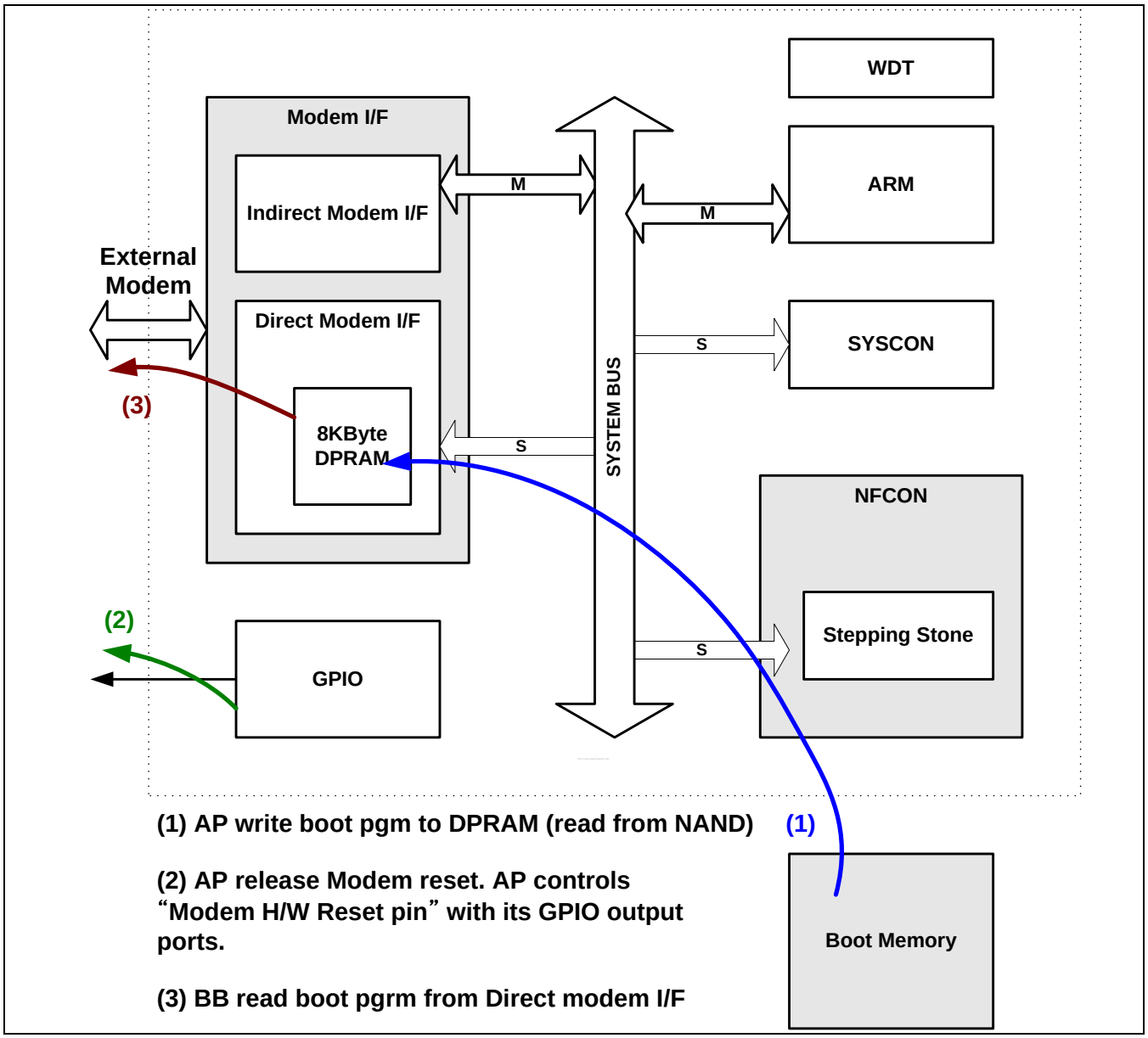


Figure 23-3. AP Booting for MODEM boot procedure

NOTES:

1: MODEM Reset pin (Reset pin of the MODEM) is controlled by the GPIO pin of the AP. The pin must be in pull-down mode when the initial power-on-reset state.
2: The boot address Chip Select pin of the MODEM should be connected "XhiCSn".

23.7 ADDRESS MAPPING

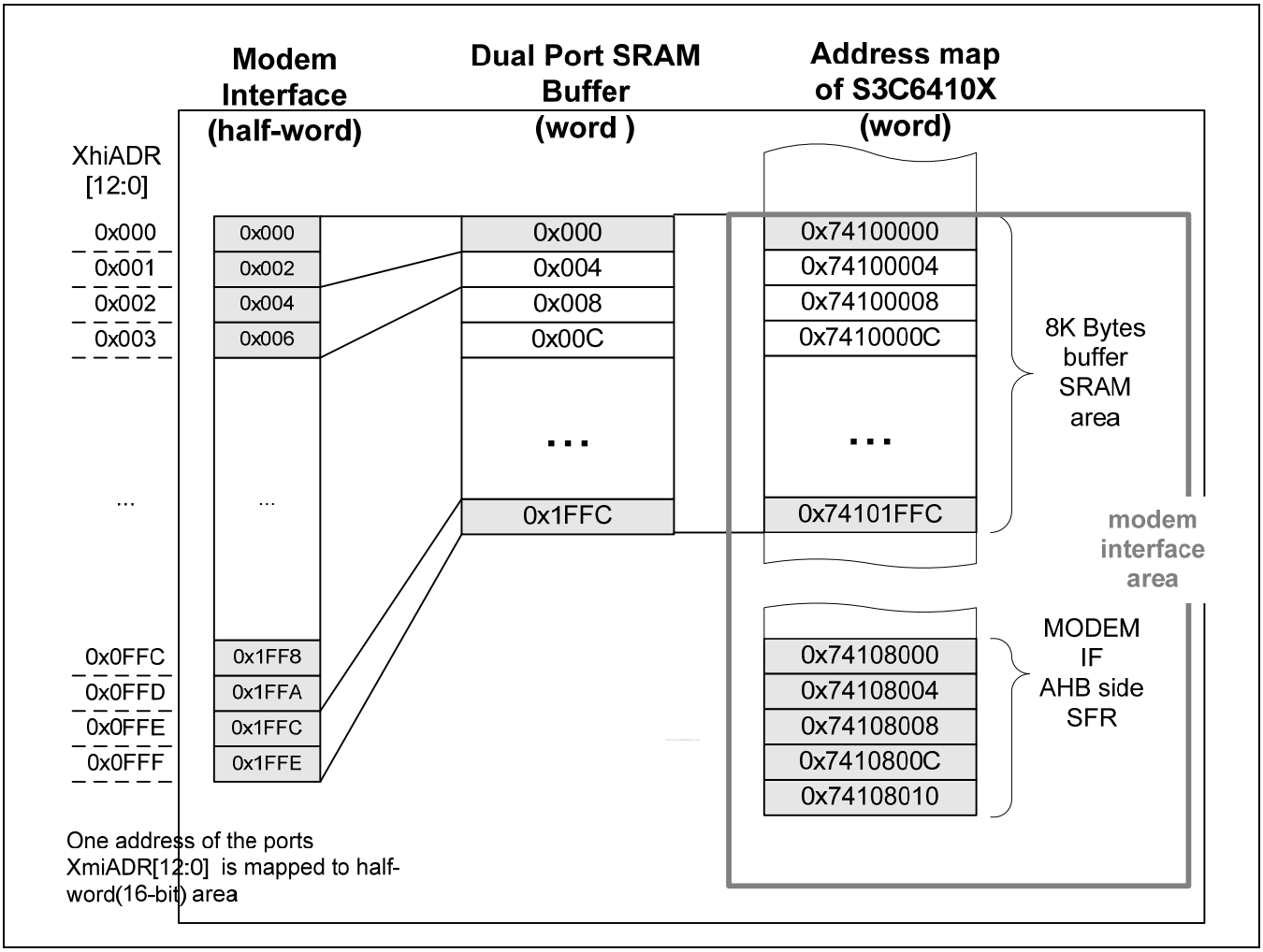


Figure 23-4. MODEM interface address mapping

When XhiCSn = '0', Address mapping of the Modem Side is following

XhiADDR			Host/Modem Interface select	description
[12]	[11:8]	[7]		
0	XXX	X	Modem Interface (Direct interface)	
1	0000	X	Host Interface (Indirect interface)	Xhi_ADDR[2]='0' (Fixed)
1	0001	0	SLEEP/STOP mode Wakeup assert 1)	Write Operation
1	0001	1	SLEEP/STOP mode Wakeup clear	Write Operation 2)
1	100X	X	LCD Bypass3) main	Modem to external main LCD driver bypass mode
1	101X	X	LCD Bypass3) sub	Modem to external sub LCD driver bypass mode

NOTES:

1: Modem Interrupt source(INT_MSM) in VIC should be enabled before AP enters STOP mode.

2: SLEEP/STOP mode wakeup clear procedure is required for the next SLEEP/STOP mode wakeup operation. After STOP/SLEEP mode wakeup operation, auto interrupt to Modem would be generated. XhiINTR output would be asserted to Low and "INT2MODEM_REG" bit[0] would be set to High. So, interrupt clear (write to INT2MODEM_REG bit[0] to 1) should be done by Modem. Auto interrupt to Modem tells the modem that the state of AP is stable.

3: Xhi_CSn_main, Xhi_CSn_sub are used for main LCD and sub LCD chip select signals when Modem uses LCD Bypass mode. About LCD Bypass mode, refer to the LCD Controller datasheet.

23.8 TIMING DIAGRAM

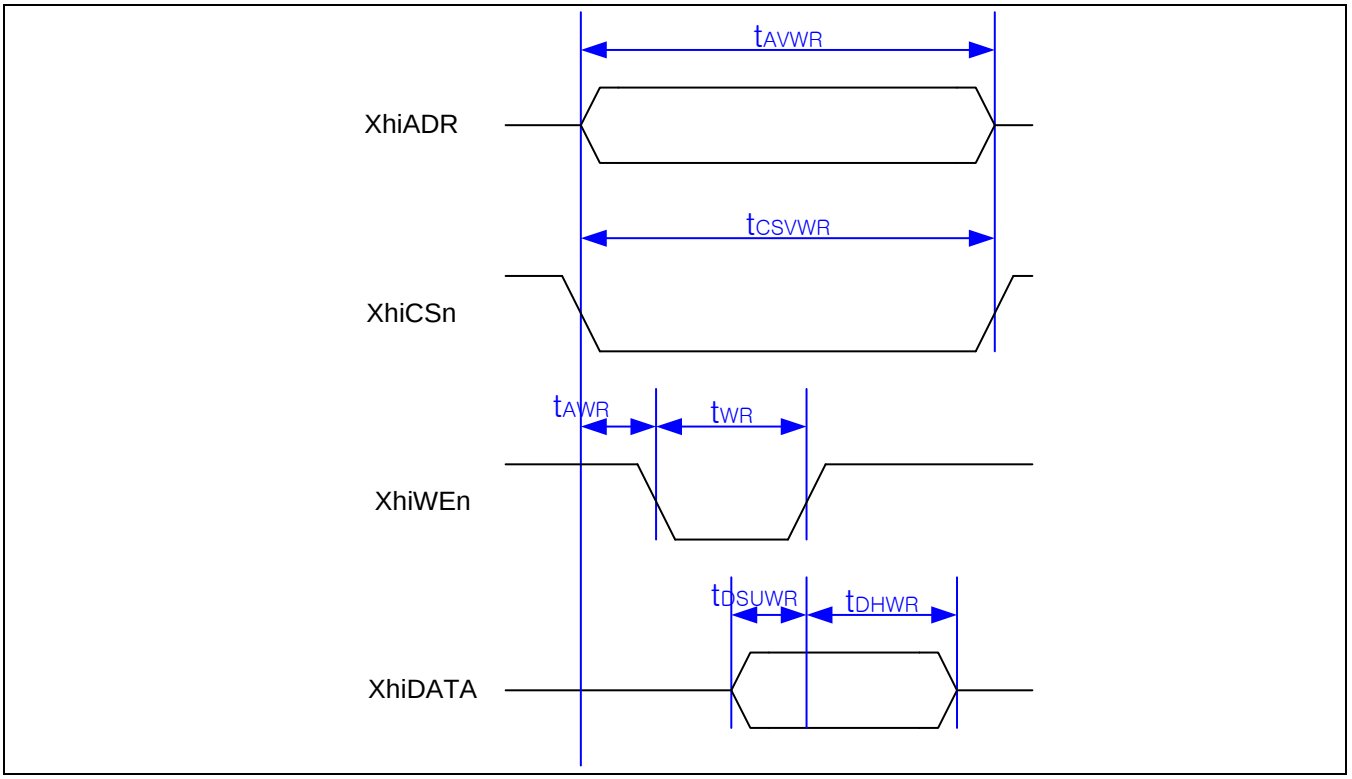


Figure 23-5. Modem interface write timing diagram

Table 23-2. Modem interface write timing

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	-	

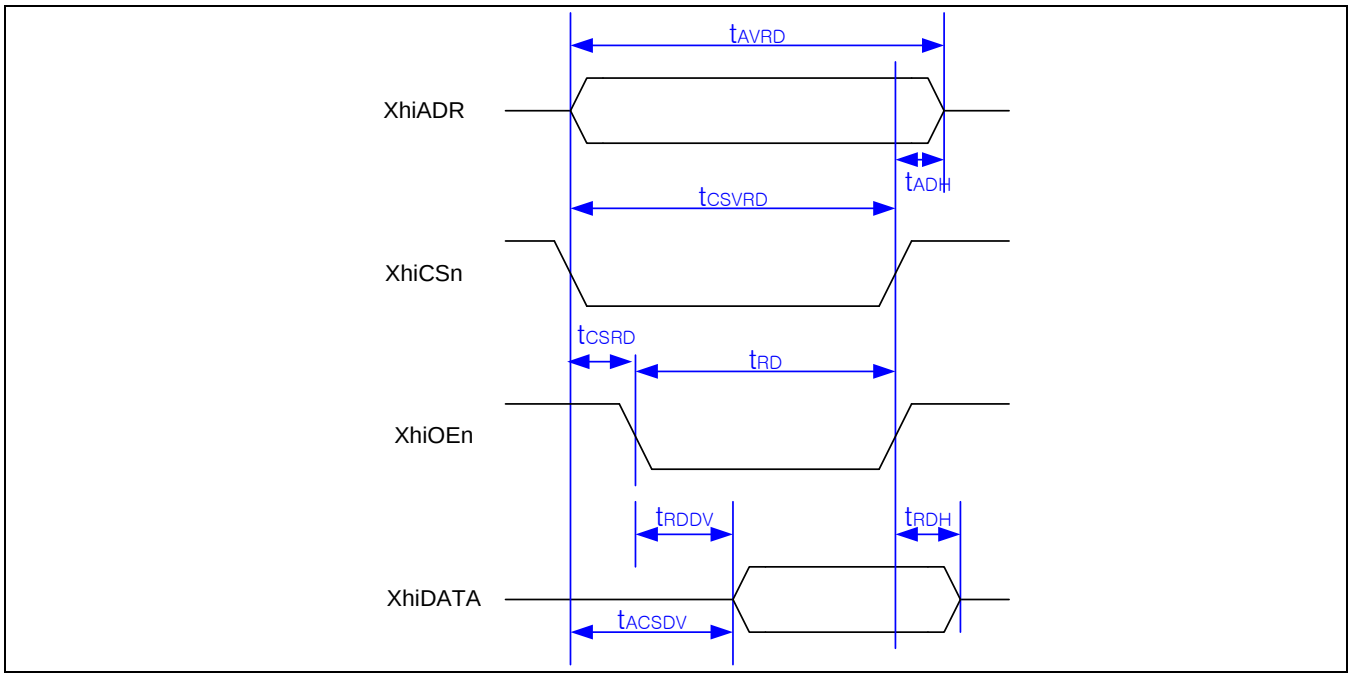


Figure 23-6. Modem interface read timing diagram

Table 23-3. Modem interface read timing

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°C)

23.9 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 other required functionalities for the data exchange between the modem chip and the AP such as the data exchange protocol, the data buffer managements, and etc.

23.10 MODEM INTERFACE SPECIAL FUNCTION REGISTERS

23.10.1 INTERRUPT REQUEST TO AP REGISTER (INT2AP)

Register	address	R/W	Description	Reset Value
INT2AP	0x74108000	R/W	Interrupt request to AP register	0x00001FFE

INT2AP	Bit	Description	Initial State
Reserved	[31:13]	Reserved	0
INT2AP_ADR	[12:0]	Modem interface requests the interrupt to AP when modem chip writes this address. This interrupt is cleared by the interrupt controller of AP and write access to the MODEMINTCLR register.	1FFE

23.10.2 INTERRUPT REQUEST TO MODEM REGISTER (INT2MODEM)

Register	Address	R/W	Description	Reset Value
INT2MODEM	0x74108004	R/W	Interrupt request to Modem register	0x00001FFC

INT2MODEM	Bit	Description	Initial State
Reserved	[31:13]	Reserved	0
INT2MODEM_ADR	[12:0]	AP requests the interrupt to MODEM by writing non-zero value to this address(INT2MODEM_ADR contents). MODEM clears the interrupt by writing High to the corresponding bit field.	1FFC

NOTE: It is recommended that AP writes data with half-word access on the interrupt port because AP can overwrite the data in INT2AP if there are INT2AP and INT2MODEM sharing the same word.

23.10.3 MODEM INTERFACE CONTROL REGISTER (MIFCON)

Register	Address	R/W	Description	Reset Value
MIFCON	0x74108008	R/W	Modem Interface Control register	0x00000008

MIFCON	Bit	Description	Initial State
Reserved	[31:20]	-	0
DMARXREQEN_1	[19]	Modem Write DMA Request (RX 1) to AP(DMA Controller) Enable	0
DMARXREQEN_0	[18]	Modem Write DMA Request (RX 0) to AP(DMA Controller) Enable	0
DMATXREQEN_1	[17]	Modem Read DMA Request (TX 1) to AP(DMA Controller) Enable	0
DMATXREQEN_0	[16]	Modem Read DMA Request (TX 0) to AP(DMA Controller) Enable	0
Reserved	[15:4]	-	0
INT2MODEMEN	[3]	Interrupt to Modem Enable : MODEM_nIRQ(XhiINTR) is interrupt signal enable. ‘0’=Disable, ‘1’=Enable	1
INT2APEN	[2]	MODEM(Modem) write interrupt to AP Enable ‘0’=Disable, ‘1’=Enable	0
Reserved	[1]	Reserved	0
Fixed	[0]	Fixed to 0	0

23.10.4 MODEM INTERFACE PORT CONTROL REGISTER (MIFPCON)

Register	Address	R/W	Description	Reset Value
MIFPCON	0x7410800C	R/W	Modem Interface Port Control register	0x00000008

MIFCON	Bit	Description	Initial State
Reserved	[31:6]	-	0
Reserved	[5]	Fixed to 0	0
INT2M_LEVEL	[4]	Interrupt to Modem Active High : MODEM_nIRQ(XhiINTR) interrupt signal make active high when this bit is set to High. ‘0’=Active Low, ‘1’=Active High Note : “INV_INTR” field of the HOST I/F block and “INT2M_LEVEL” of the MODEM I/F block should have same polarity.	0
SEL_BYPASS	[3]	Select (mux) control for LCD bypass (from Modem/Host Interface to LCD i80 interface) ‘0’ = Normal mode ‘1’ = Bypass mode (initial value)	1

SEL_RS	[2:0]	RS selection for LCD Bypass path 3'b000 : XhiADDR[8] -> bypass_RS 3'b001 : XhiADDR[7] -> bypass_RS 3'b010 : XhiADDR[6] -> bypass_RS 3'b011 : XhiADDR[5] -> bypass_RS 3'b100 : XhiADDR[4] -> bypass_RS 3'b101 : XhiADDR[3] -> bypass_RS Ohters : XhiADDR[8] -> bypass_RS	0

NOTE: SEL_BYPASS field initial value is 1. But, when the Sleep mode wakeup case, if LCD Bypass is not used and Automatic SLP_EN is enabled (GPIO SLPEN register field 'SLPEN_CFG'), the SEL_BYPASS field shows High after Sleep mode wakeup. In this case, the SEL_BYPASS field should be cleared to LOW for SFR and operation coherency.

23.10.5 MODEM INTERRUPT CLEAR REGISTER (MODEMINTCLR)

Register	Address	R/W	Description	Reset Value
MODEMINTCLR	0x74108010	W	Modem Interface Pending Interrupt Request Clear	-

MODEMINTCLR	Bit	Description	Initial State
-	[31:0]	Write access to this register with any data will clear the interrupt pending register of Modem interface.	-

NOTE: The interrupt controllers of AP(S3C6410X), VIC, receive level-triggered type interrupt requests. Therefore, interrupt requests from MODEM_IF block are maintained until ARM(the interrupt service routine S/W) clears this register by writing to HIGH.

23.10.6 DMA REQUEST TX ADDRESS REGISTER (DMAREQ_TX_ADR)

Register	address	R/W	Description	Reset Value
DMA_TX_ADR	0x74108014	R/W	DMA TX request Address register	0x07FE03FE

INT2AP	Bit	Description	Initial State
Reserved	[31:29]	Reserved	0
DMA_TX_ADR_1	[28:16]	Modem interface requests the DMA to AP(DMA Controller) when modem chip **reads** this address. Source : MODEM_IF TX 1 Request	07FE
Reserved	[15:13]	Reserved	0
DMA_TX_ADR_0	[12:0]	Modem interface requests the DMA to AP(DMA Controller) when modem chip **reads** this address. Source : MODEM_IF TX 0 Request	03FE

NOTE: DMA TX Request signal will be generated when the MODEM

23.10.7 DMA REQUEST RX ADDRESS REGISTER (DMAREQ_RX_ADR)

Register	address	R/W	Description	Reset Value
DMA_RX_ADR	0x74108018	R/W	DMA RX request Address register	0x0FFE0BFE

INT2AP	Bit	Description	Initial State
Reserved	[31:29]	Reserved	0
DMA_RX_ADR_1	[28:16]	Modem interface requests the DMA to AP(DMA Controller) when modem chip **writes** this address. Source : MODEM_IF RX 1 Request	0FFE
Reserved	[15:13]	Reserved	0
DMA_RX_ADR_0	[12:0]	Modem interface requests the DMA to AP(DMA Controller) when modem chip **writes** this address. Source : MODEM_IF RX 0 Request	0BFE

24 HOST INTERFACE

24.1 OVERVIEW

The Host Interface block in the S3C6410X supports indirect access of the external host device (Ex: Modem Chip).

By the selected host interface protocol, the following operations are supported:

- 16-bit protocol register
- Single R/W on the SFR/memory in the system memory map
- Burst R/W on the SFR/memory in the system memory map
- Repeated Burst Write on the SFR/memory in the system memory map
- Supports Modem Booting that enables HOST to control AP boot.

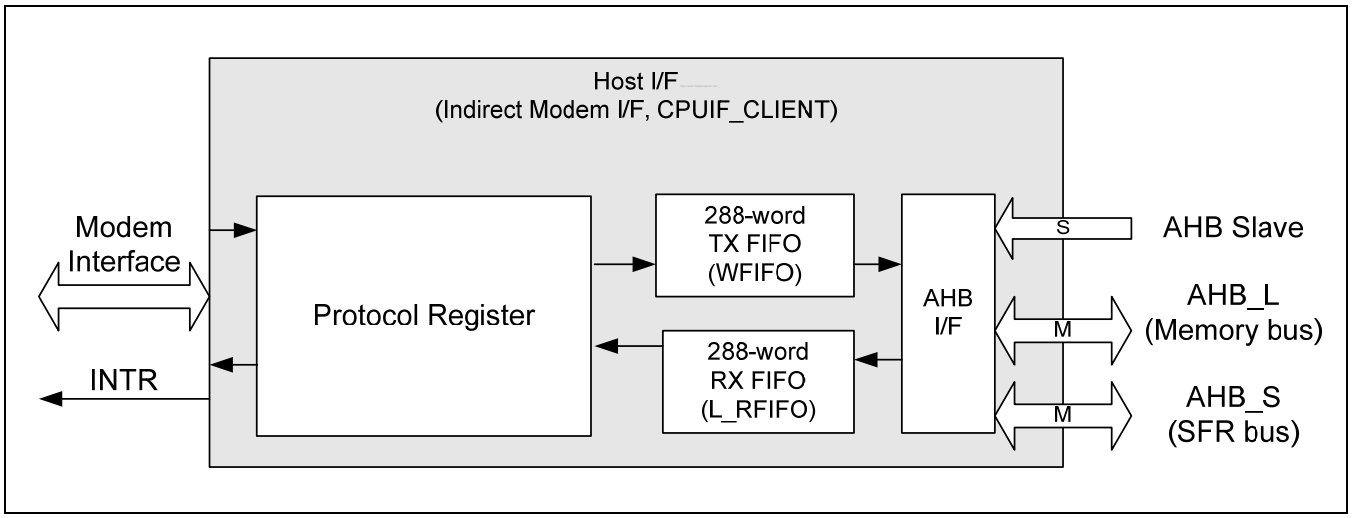


Figure 24-1. HOST I/F Block Diagram

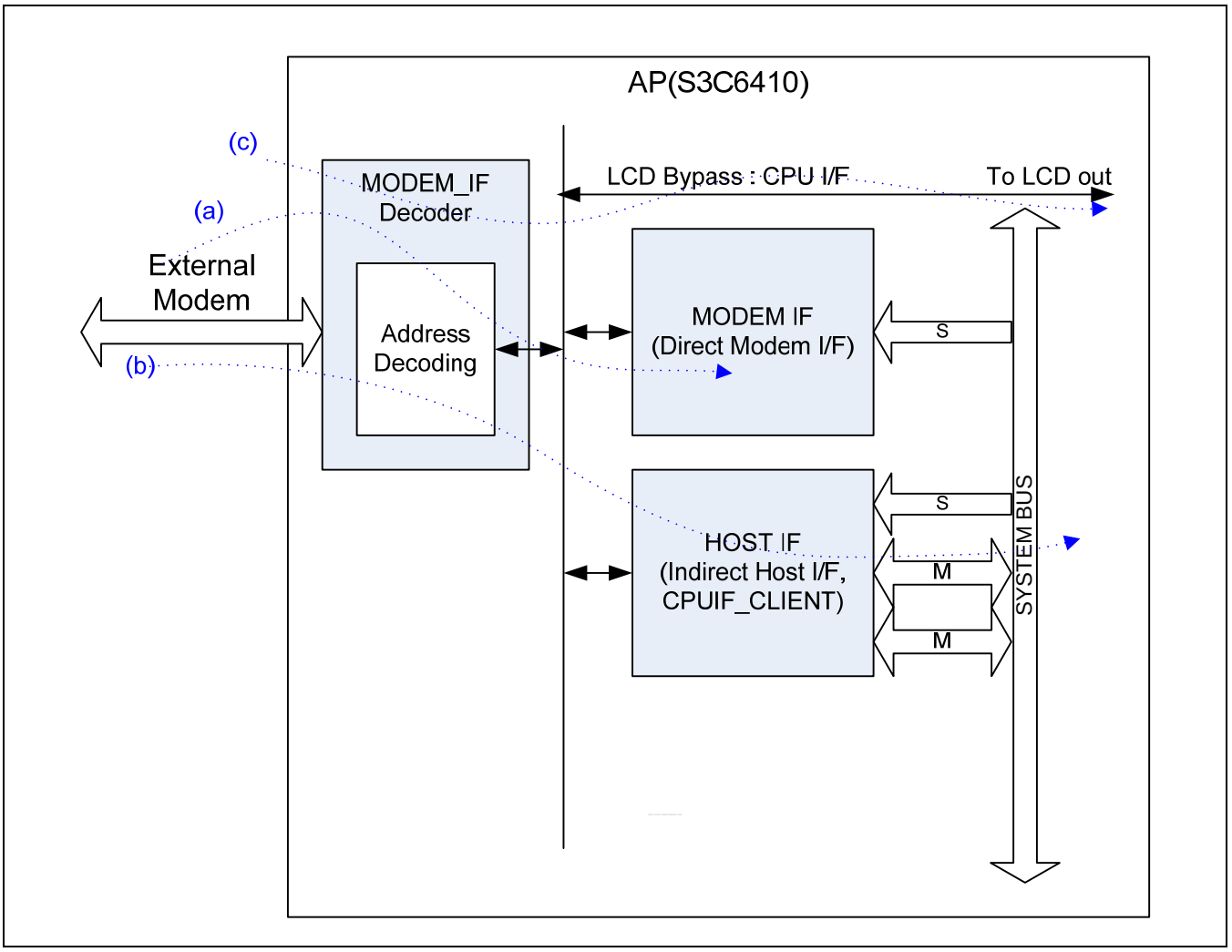


Figure 24-2. Data flow of the the External Host device (MODEM) and the AP

The External Host Interface of the AP supports (a) the Direct Modem Interface path (MODEM_IF), (b) Indirect Modem Interface path (HOST_IF) and (c) the LCD bypass path. (Figure 24-2)

When XhiCSn = '0', Address mapping of the Modem Side is following

XhiADDR			Host/Modem Interface select	Description
[12]	[11:8]	[7]		
0	XXX	X	Modem Interface (Direct interface)	
1	0000	X	Host Interface (Indirect interface)	Xhi_ADDR[2]='0' : Indirect Host I/F, Xhi_ADDR[2]='1' : Reserved
1	0001	0	SLEEP/STOP mode Wakeup assert	Write Operation
1	0001	1	SLEEP/STOP mode Wakeup clear	Write Operation
1	100X	X	LCD Bypass main	
1	101X	X	LCD Bypass sub	

NOTE: Xhi_CSn_main, Xhi_CSn_sub : Another Chip Select signals for LCD Bypass main and sub

25.2 FEATURES

Host Interface (Indirect Modem Interface) Features

- Asynchronous indirect 16-bit SRAM-style host interface (i80)
- Banked 16-bit protocol registers for host interface
- Write-FIFO and Read-FIFO to support burst write/read transfer up to 1 Kbytes
- A 32-bit in-mailbox register and a 32-bit out-mailbox register for data exchange

24.3 FUNCTIONAL DESCRIPTION

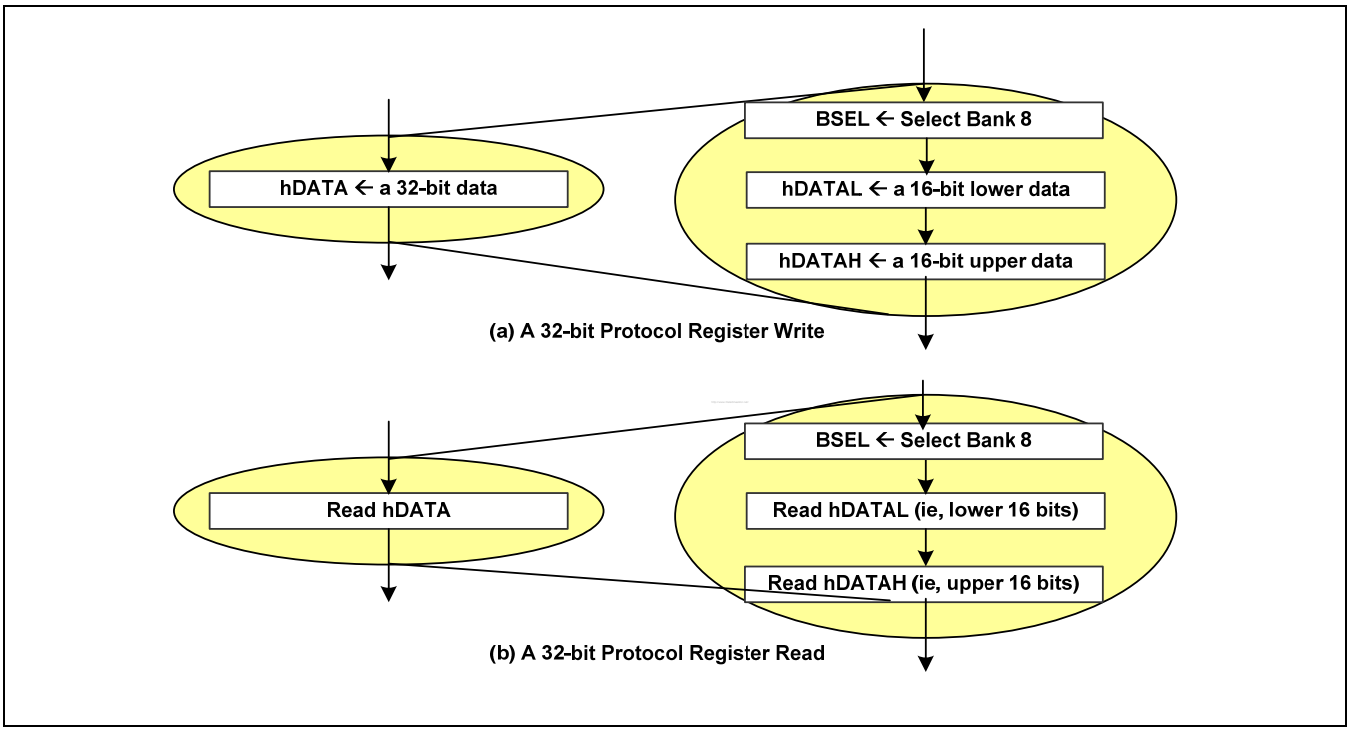


Figure24-3. A 32-bit Protocol Register Access by Using Two 16-bit Read/Write

NOTE:

hDATAL is “Host Interface Data Low Register” and hDATAH is “Host Interface Data High Register”. For more information refer to the Register Description section.

24.3.1 READ AND WRITE OF A 16-BIT PROTOCOL REGISTER

To access a 16-bit protocol register take the following steps:

1. First select a corresponding bank by writing BSEL, then
2. Read or write the protocol register.

24.3.2 READ AND WRITE OF TWO 16-BIT PROTOCOL REGISTERS IN THE SAME BANK

To access two 16-bit protocol registers in the same bank take the following steps:

1. First select a corresponding bank by writing BSEL,
2. Read or write the lower 16-bit protocol register, then
3. Read or write the upper 16-bit protocol register.

24.3.3 SINGLE WRITE

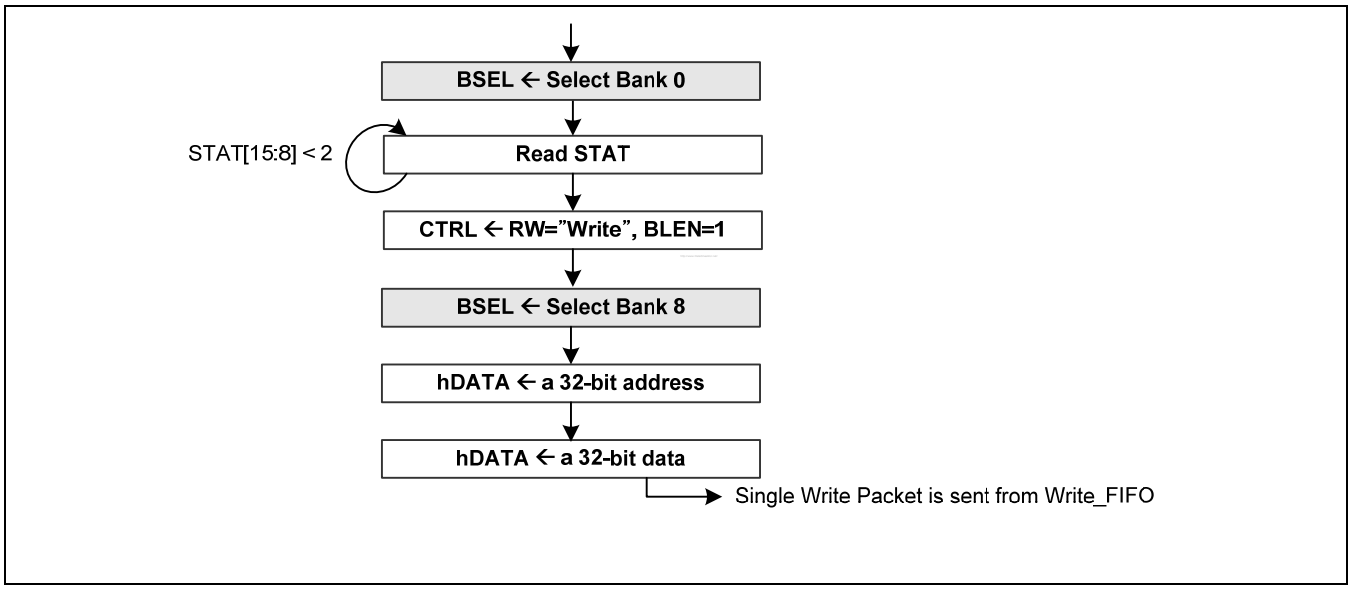


Figure 24-4. Single Write Procedure

24.3.4 SINGLE READ

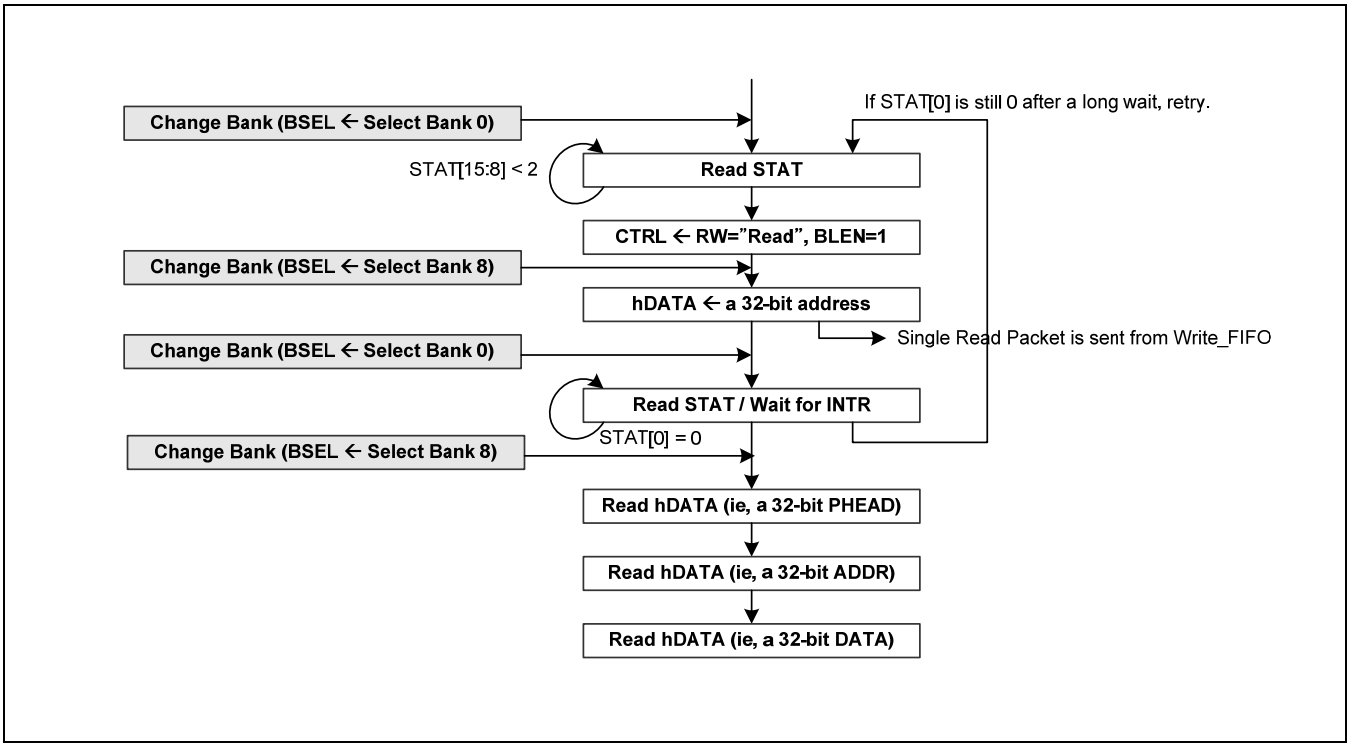


Figure 24-5. Single Read Procedure

The name "HOST I/F" means HOST Interface block and the name "HOST" means External Host device (ex. MODEM) in this section.

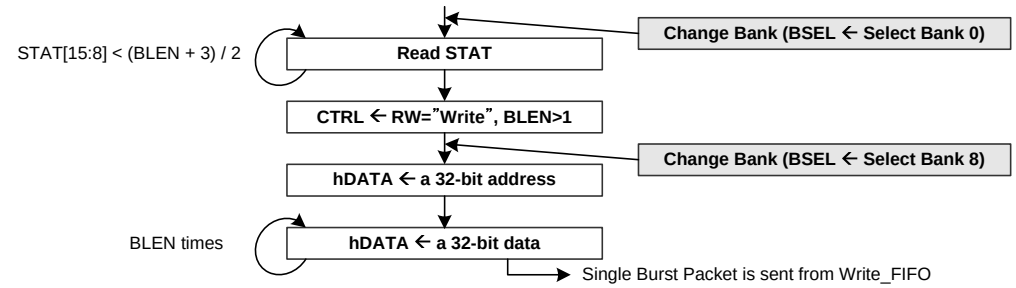
The HOST I/F reads a result from AP (S3C6410) and it can start a new read operation before the completion of the previous one. There is no information on the source of the result. Therefore, the HOST must make sure that the results of multiple read operations are not mixed. For example, if Burst Read from a "Source area A" is issued and Single Read from a "Source area B" is issued without waiting for the result of Burst Read, the result of Single Read can arrive at the read buffer earlier. As a result, the address of the received result must be compared with the requested address.

However, for multiple Read operations to the same "Source area", the orders of the results are maintained.

The simple way to avoid the above situation (ie, Out-order Execution Issue) is that the HOST makes sure that the previous result is completely read by the HOST before issuing another Read command (i.e., Single Read or Burst Read).

The HOST can know the status of the 32-bit read buffer by polling a 16-bit status register (i.e., STAT [0]) or by using an interrupt scheme (i.e., INTR). If an interrupt scheme is used (i.e., the corresponding bits of INTE [15:0] are set high), the HOST must read the status register in ISR (Interrupt Service Routine) in order to know the cause of the interrupt.

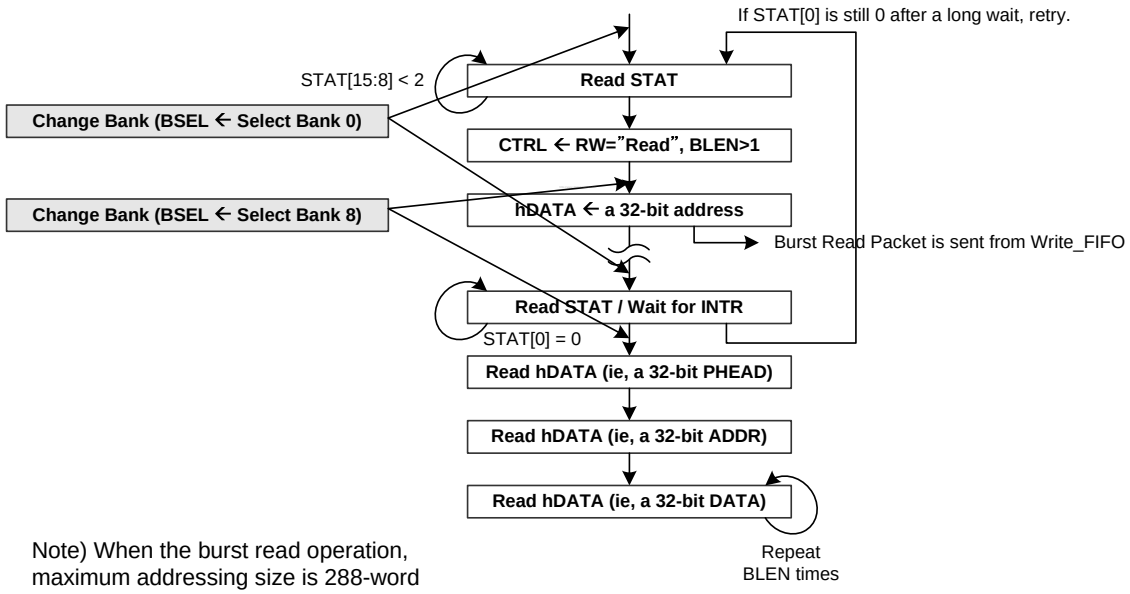
24.3.5 BURST WRITE



Note) When the burst write operation, maximum addressing size is 288-word

Figure 24-6. Burst Write Procedure

24.3.6 BURST READ



Note) When the burst read operation, maximum addressing size is 288-word

Figure 24-7. Burst Read Procedure

24.3.7 REPEATED BURST WRITE TO REDUCE THE HOST (EX. MODEM) OVERHEAD

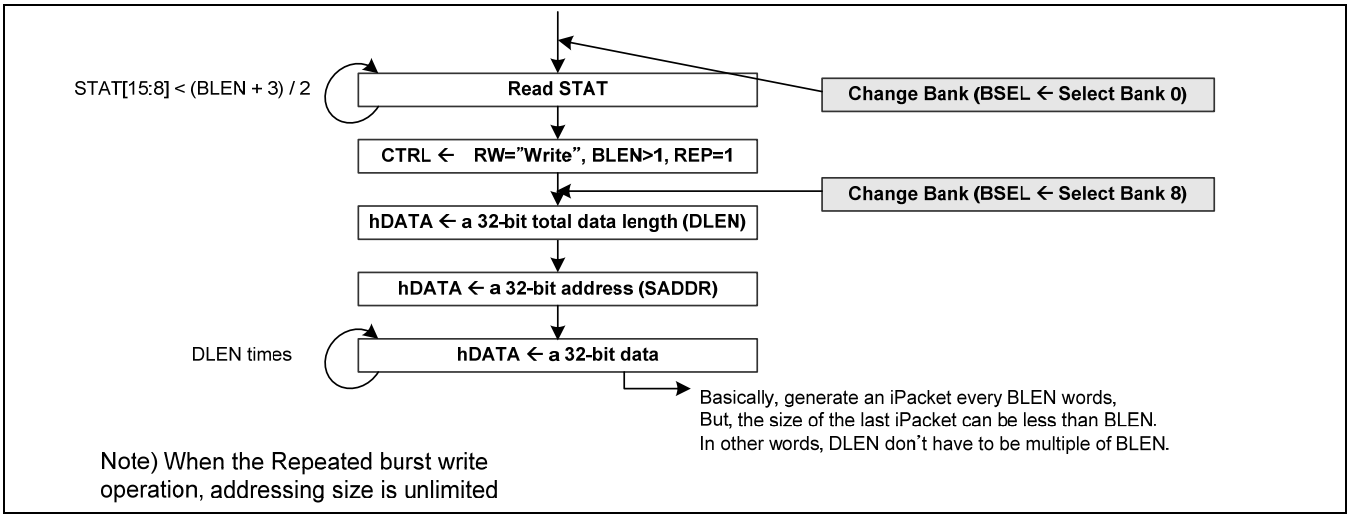


Figure24-8. Repeated Burst Write Procedure

24.3.8 MODEM BOOTING

MODEM Booting means that the Host (Modem) controls AP booting including Reset. In this case, AP does not require an External Boot Memory. Modem downloads the AP boot code from its Boot Memory to the Stepping Stone memory area (4Kbyte) inside AP through HOST I/F (indirect modem I/F) block. Then Modem assert the boot done (bit[0] field in the SYS_CTRL register) signal to release AP operation.

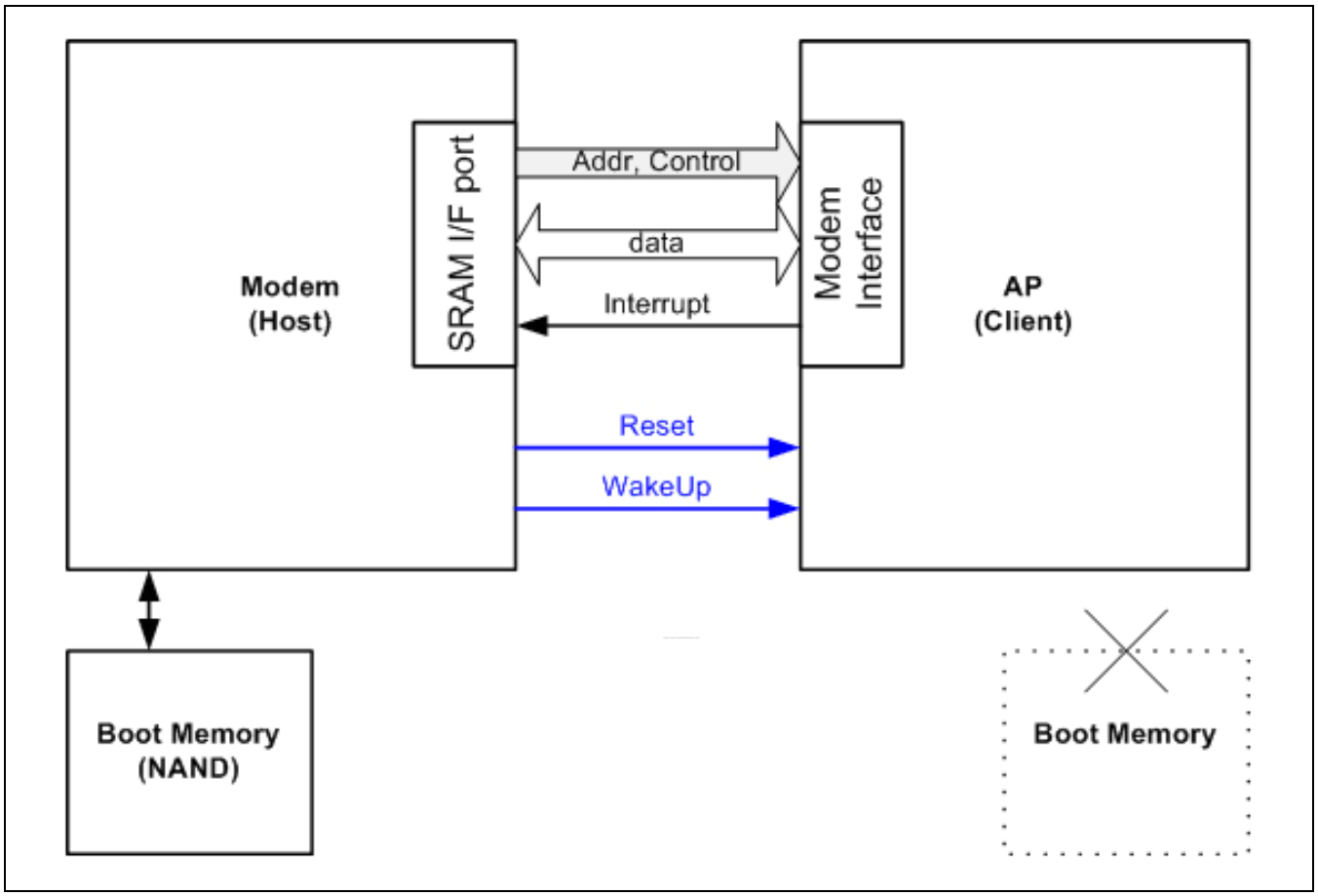


Figure24-9. Modem Boot connection diagram

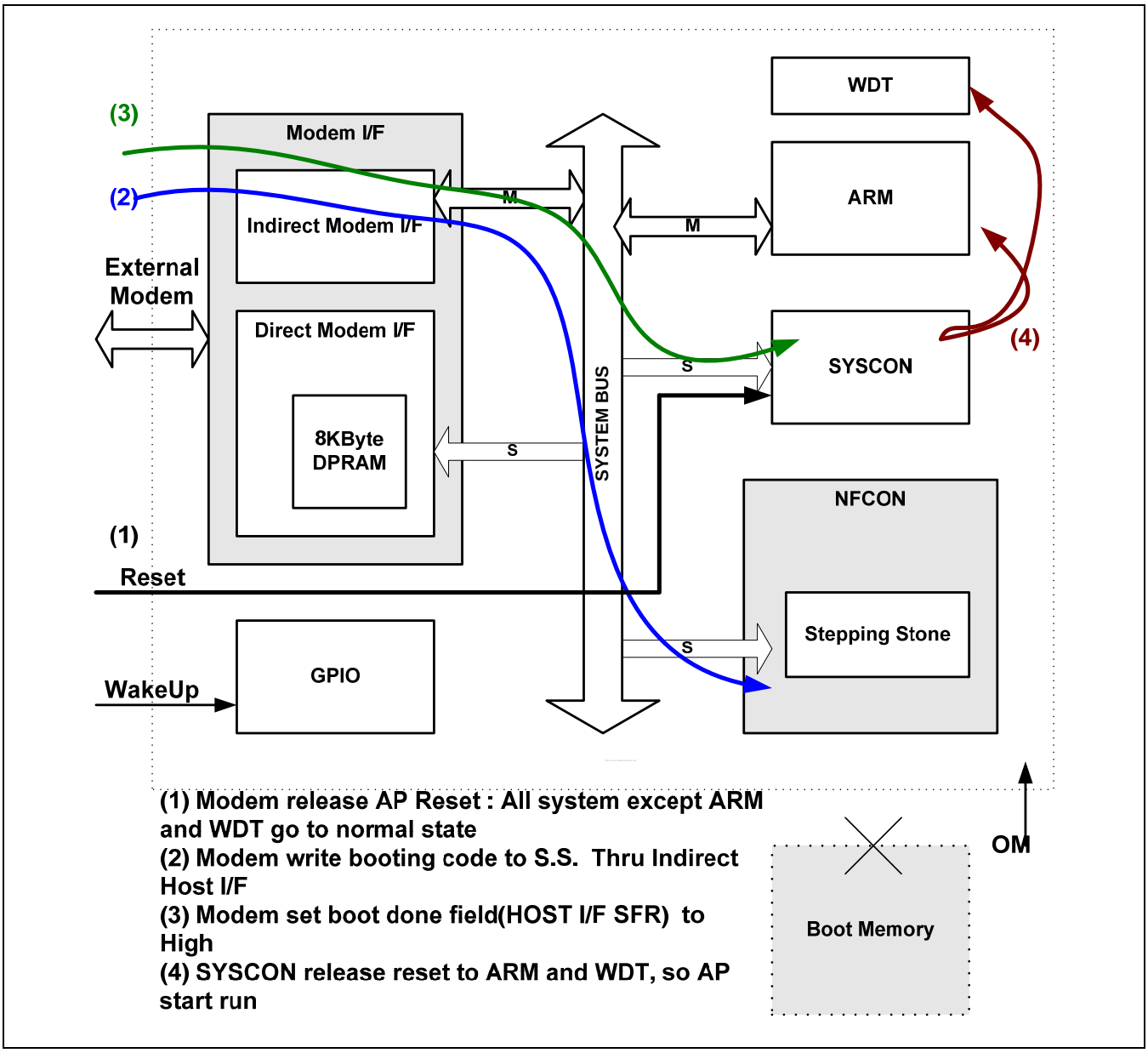


Figure 24-10. Modem Boot procedure

Note) When the Modem Booting procedure is used, XOM setting should be set as a Modem Booting mode initially.

24.3.9 MAILBOX INTERFACE

The modem can use a 32-bit in-mailbox and a 32-bit out-mailbox for IPC (Inter Process Communication) between the modem and the AP (S3C6410).

24.3.10 MAILBOX BASIC OPERATION

When the modem writes a 32-bit data into the in-mailbox, the HOST I/F generates an interrupt to the AP (S3C6410) so that the AP (S3C6410) will read the in-mailbox to know the requests from the modem. The format of the in-mailbox can be freely defined by the application. The in-mailbox flag in the HOST I/F is automatically cleared when the AP (S3C6410) reads the in-mailbox.

When the AP (S3C6410) writes a 32-bit data into the out-mailbox, the HOST I/F generates an interrupt to the modem so that the modem will read the out-mailbox to know the requests from the AP (S3C6410). The format of the out-mailbox can be freely defined by the application. The out-mailbox flag (ie, STAT1 [0]) in the HOST I/F is automatically cleared when the modem reads the out-mailbox.

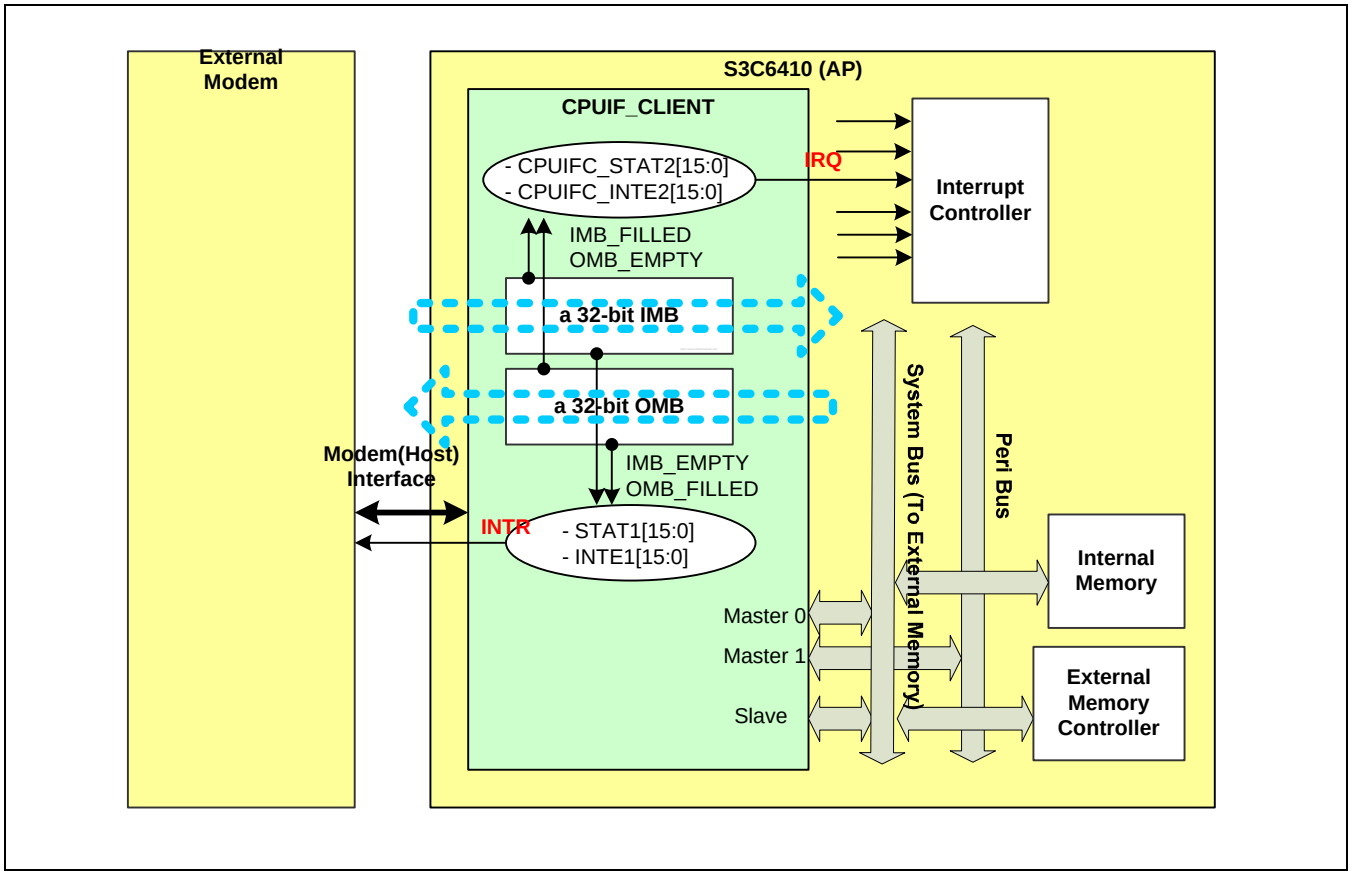


Figure 24-11. In/Out Mailbox between Modem and AP

24.3.11 MAILBOX OPERATION WITH A BULK DATA

When a 32-bit mailbox is not enough to transfer information, any memory of AP(S3C6410) can be used as buffer memory.

For example, in order to transfer a bulk data from the host to AP(S3C6410), the host first writes the data into a memory in AP(S3C6410) by using Burst Write of the HOST I/F. After checking the completion of Burst Write by reading the status register (ie, STAT[1] = 0), the host writes a message into the in-mailbox.

In order to transfer a bulk data from the AP(S3C6410) to the host, AP(S3C6410) first prepares the data in its memory and writes a message into the out-mailbox by using SFR access. The write will generate an interrupt to the host.

Table 24-1. Registers for Mailbox Interface

Group	Operation	Registers
Protocol register	Registers for mail-box	IMBL, IMBH, OMBL, OMBH
SFR	Mailbox	HOSTIFC_IMB, HOSTIFC_OMB

24.4 PROGRAMMER'S MODEL

24.4.1 THE REGISTERS OF HOST INTERFACE ARE CLASSIFIED INTO:

1. Protocol Registers that are accessed through 16-bit Host Interface by modem, and
2. Special Function Registers that are accessed through SYSTEM bus by the bus master.

By using protocol registers, modem can execute the following operations:

1. Single transfer (Host Interface)
2. Burst transfer (Host Interface)
3. Reading of all address areas of the S3C6410 including Special Function Registers
4. Writing of all address areas of the S3C6410 including Special Function Registers
5. Write a 32-bit message into In-Mail box
6. Read a 32-bit message from Out-Mail box

By using SFR's the following operations are supported:

1. Write a 32-bit message into Out-Mail box
2. Read a 32-bit message from In-Mail box

24.5 REGISTER DESCRIPTIONS (PROTOCOL REGISTERS)

Register	Bank	MP_A[1:0]	R/W	Description	Reset Value
CTRL	0x0	00	R/W	Control Register	0x0000
INTE		01	R/W	Interrupt Enable Register	0x2000
STAT		10	R	Status Register	0x90A2
CTRL1	0x1	00	R/W	Control1 Register	0x0000
INTE1		01	R/W	Interrupt Enable1 Register	0x0000
STAT1		10	R	Status1 Register	0x0002
IMBL	0x2	00	R/W	In-Mail Box Low Register	0x0000
IMBH		01	R/W	In-Mail Box High Register	0x0000
OMBL	0x3	00	R	Out-Mail Box Low Register	0x0000
OMBH		01	R	Out-Mail Box High Register	0x0000
hDATAL	0x8	00	R/W	Host Interface Data Low Register	-
hDATAH		01	R/W	Host Interface Data High Register	-
SYS_CTRL	0xB	00	R/W	System Control Register	0x0000
Reserved		01	R/W	Reserved	0x0005
Reserved		10	R	Reserved	-
BSEL	all banks	11	R/W	Bank Selection Register	0x0000

NOTE: MP_A[1:0] are input pin names “XhiADDR[1:0]”, MP means Modem Processor and MP_A means MP Address”.

24.5.1 PROTOCOL REGISTER MATRIX

As shown in Table 24-2 and Table 24-3, protocol registers are classed into 16 banks so that BSEL[3:0] must be properly set before access.

Table 24-2. Protocol Register Matrix (Bank0 ~ Bank7)

MP_A[1:0]	Protocol Register (Selected by BSEL[3:0])							
	Bank0 (0000)	Bank1 (0001)	Bank2 (0010)	Bank3 (0011)	Bank4 (0100)	Bank5 (0101)	Bank6 (0110)	Bank7 (0111)
00	CTRL	CTRL1	IMBL	OMBL	Reserved	Reserved	Reserved	Reserved
01	INTE	INTE1	IMBH	OMBH	Reserved	Reserved	Reserved	Reserved
10	STAT	STAT1	reserved	reserved	Reserved	Reserved	Reserved	Reserved
11	BSEL[3:0]							

Table 24-3. Protocol Register Matrix (Bank8 ~ Bank15)

MP_A[1:0]	Protocol Register (Selected by BSEL[3:0])							
	Bank8 (1000)	Bank9 (1001)	Bank10 (1010)	Bank11 (1011)	Bank12 (1100)	Bank13 (1101)	Bank14 (1110)	Bank15 (1111)
00	hDATAL	Reserved	Reserved	SYS_CTRL	Reserved	Reserved	Reserved	Reserved
01	hDATAH	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
10	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
11	BSEL[3:0]							

SFR-mirrored registers: INTE, INTE1, STAT, STAT1, IMBH, IBML, OMBH, OMBL
Mirrored-register means the register can be accessed by both AP and MODEM side.

24.5.2 CONTROL REGISTER (CTRL)

BSEL[3:0] = 0000, MP_A[1:0] = 00, R/W, Reset value = 0x0000

Field	Bit	Description	Initial State
Reserved	[15:13]		00
BLEN[8:0]	[12:4]	Burst length for transfer The basic unit is a 32-bit word. Maximum burst length is 256 words. 0 = No transaction 1 = Single Write or Single Read N = N-word * Note) BLEN must be 0 not to issue a new HOST I/F command.	0_0000_0000
REP_WRITE	[3]	Repeated Burst Write Enable 0 = No 'Repeated Burst Write' 1 = If set, "Burst Write" is treated as "Repeated Burst Write"	0
HOSTIF_RESET	[2]	Reset of the HOST I/F 0 = This bit is used a soft reset signal of the HOST I/F. 1 = Therefore, this bit must be de-asserted by software.	0
Reserved	[1]	This field must be fixed to '0'	0
READ_WRITE	[0]	Read or Write 0 = Write operation, 1 = Read operation	0

24.5.3 INTERRUPT ENABLE REGISTER (INTE)

BSEL[3:0] = 0000, MP_A[1:0] = 01, R/W, Reset value = 0x2000

Field	Bit	Description	Initial State
WFIFO_THRES	[15:8]	Threshold of the empty elements in WFIFO This value specifies the threshold of the partial emptiness. For example, if WFIFO_THRES is 8, STAT[7] becomes 1 when 16 and more elements in WFIFO are empty.	0x20
WFIFO_PEMPTY	[7]	WFIFO partial empty interrupt enable Interrupt occurs (i.e., INTR becomes 1) when INTE [7] = 1 and STAT [7] = 1.	0
Reserved	[6:2]	-	0
WFIFO	[1]	WFIFO interrupt enable Interrupt occurs when INTE [1] = 1 and STAT [1] = 1.	0
RFIFO	[0]	RFIFO interrupt enable Interrupt occurs when INTE [0] = 1 and STAT [0] = 1.	0

24.5.4 STATUS REGISTER (STAT)

BSEL [3:0] = 0000, MP_A[1:0] = 10, R, Reset value = 0x90A2

Field	Bit	Description	Initial State
WRITABLE_CNT	[15:8]	Writable word counts in wfifo: Read only This field shows how many words can be written into WFIFO without checking of the WFIFO fullness. For example, if this field is 0x08, there are currently 16 (8 x 2) entries of wfifo so that the HOST can write 16 words immediately. Note that the reset value is 0x90 so that 288 words can be written without overflow. Practically, since the packet head and address must be written into WFIFO, the recommend maximum data payload size is 256-word.	0x90
WFIFO_PEMPTY	[7]	WFIFO partial empty flag (from HOST I/F itself) This flag becomes 1 when the number of empty elements in WFIFO is larger than or equal to a threshold (ie, 2 x INTE [15:8]).	1
Reserved	[6:2]	-	0
WFIFO	[1]	WFIFO empty flag. (from HOST I/F itself) This flag becomes 1 when Write Buffer is empty so that the maximum capacity of Write Buffer is available right now. At least a single write into wfifo resets this flag.	1
RFIFO	[0]	RFIFO ready flag (from HOST I/F itself) This flag becomes 1 when Read Buffer is ready to read. The HOST read of DATAH resets this flag.	0

24.5.5 INTERRUPT ENABLE1 REGISTER (INTE1)

BSEL[3:0] = 0001, MP_A[1:0] = 01, R/W, Reset value = 0x0000

Field	Bit	Description	Initial State
Reserved	[15:2]		0
IMB_EMPTY	[1]	IMB empty interrupt enable Interrupt occurs when INTE1 [1] = 1 and STAT1 [1] = 1.	0
OMB_FILLED	[0]	OMB filled interrupt enable Interrupt occurs when INTE1 [0] = 1 and STAT1 [0] = 1.	0

24.5.6 STATUS1 REGISTER (STAT1)

BSEL[3:0] = 0001, MP_A[1:0] = 10, R/W, Reset value = 0x0002

Field	Bit	Description	Initial State
Reserved	[15:2]		0
IMB_EMPTY	[1]	IMB (In-Mail Box) empty flag This flag is an inversion of the IMB_FILLED (ie, HOSTIFC_STAT2[17]).	1
OMB_FILLED	[0]	OMB (Out-Mail Box) filled flag This flag is set when the out-mailbox is written by SFR access. In order to clear this flag, HIGH value must be written in this bit.	0

24.5.7 IN-MAIL BOX LOW REGISTER (IMBL)

BSEL[3:0] = 0010, MP_A[1:0] = 00, R/W, Reset value = 0x0000

Field	Bit	Description	Initial State
IMBL	[15:0]	Lower 16 bits of In-Mail Box register HOST writes a 16-bit data into IMBL.	0x0000

24.5.8 IN-MAIL BOX HIGH REGISTER (IMBH)

BSEL[3:0] = 0010, MP_A[1:0] = 01, R/W, Reset value = 0x0000

Field	Bit	Description	Initial State
IMBH	[15:0]	Upper 16 bits of In-Mail Box register After HOST writes a 16-bit data into IMBH, HOST I/F asserts 'IMB_flag' (an internal signal) in order to notify that a 32-bit IMB contains a new value. IMB_flag is automatically cleared when IMB is read by software.	0x0000

24.5.9 OUT-MAIL BOX LOW REGISTER (OMBL)

BSEL[3:0] = 0011, MP_A[1:0] = 00, R, Reset value = 0x0000

Field	Bit	Description	Initial State
OMBL	[15:0]	Lower 16 bits of Out-Mail Box register HOST reads a lower 16-bit data of a 32-bit OMB.	0x0000

24.5.10 OUT-MAIL BOX HIGH REGISTER (OMBH)

BSEL[3:0] = 0011, MP_A[1:0] = 01, R, Reset value = 0x0000

Field	Bit	Description	Initial State
OMBH	[15:0]	Upper 16 bits of Out-Mail Box register When the host reads an upper 16-bit data of a 32-bit OMB, STAT1[0] (ie, OMB flag) is automatically cleared.	0x0000

24.5.11 HOST INTERFACE DATA LOW REGISTER (HDATAL)

BSEL[3:0] = 1000, MP_A[1:0] = 00, R/W, Reset value = (undefined)

Field	Bit	Description	Initial State
DATAL	[15:0]	Data Register	-

24.5.12 HOST INTERFACE DATA HIGH REGISTER (HDATAH)

BSEL[3:0] = 1000, MP_A[1:0] = 01, R/W, Reset value = (undefined)

Field	Bit	Description	Initial State
DATAH	[15:0]	Data Register	-

24.5.13 SYSTEM CONTROL REGISTER (SYS_CTRL)

BSEL[3:0] = 1011, MP_A[1:0] = 00, R/W, Reset value = 0x0000

Field	Bit	Description	Initial State
Reserved	[15:1]	Data Register	0
BOOTDONE	[0]	Boot Done for Modem Booting. When this bit is set to High, boot done signal is asserted to System Controller. S3C6410(AP) boot operation starts. Host (Modem) must clear this bit to Low after Modem boot.	0

24.5.14 BANK SELECTION REGISTER (BSEL)

MP_A[1:0] = 11, R/W, Reset value = 0x0000

Field	Bit	Description	Initial State
Reserved	[15:4]		0x000
BSEL	[3:0]	Bank selection 0000 = One of the protocol registers in Bank0 will be selected. 0001 = One of the protocol registers in Bank1 will be selected. … 1111 = One of the protocol registers in Bank15 will be selected.	0000

24.6 REGISTER DESCRIPTIONS (SPECIAL FUNCTION REGISTERS)

Base address : 0x7400_0000				
Register	**Offset**	**R/W**	**Description**	**Reset Value**
HOSTIFC_CTRL	0x000	R/W	HOST I/F Control Register	0x20FF_0100
Reserved	0x004	R/W	Reserved	0x0000_0006
HOSTIFC_TMP	0x008	R/W	HOST I/F Temporary Register	0x0000_0000
Reserved	0x00C	--	Reserved	0x0000_0000
HOSTIFC_IMB	0x010	R	HOST I/F IMB Register	0x0000_0000
HOSTIFC_OMB	0x014	R/W	HOST I/F OMB Register	0x0000_0000
HOSTIFC_MR_STAT	0x020	R	HOST I/F Status Mirrored Register	0x0000_90A2
HOSTIFC_MR_STAT1	0x024	R	HOST I/F Status1 Mirrored Register	0x0000_0002
HOSTIFC_STAT2	0x028	R/W	HOST I/F Status2 Register	0x0001_0000
Reserved	0x02C	--	Reserved	0x0000_0000
HOSTIFC_MR_INTE	0x030	R/W	HOST I/F Interrupt Enable Mirrored Register	0x0000_2000
HOSTIFC_MR_INTE1	0x034	R	HOST I/F Interrupt Enable1 Mirrored Register	0x0000_0000
HOSTIFC_INTE2	0x038	RW	HOST I/F Interrupt Enable2 Register	0x0000_0000
Reserved	0x03C	--	Reserved	0x0000_0000

24.6.1 HOST I/F CONTROL REGISTER (HOSTIFC_CTRL)

Register	address	R/W	Description	Reset Value
HOSTIFC_CTRL	0x74000000	R/W	HOST I/F Control Register	0x20FF0100

HOSTIFC_CTRL	Bit	Description	Initial State
Reserved	[31:30]	-	0
INV_INTR	[29]	Polarity inversion of INTR 0: INTR is active high so that INTR becomes HIGH when an interrupt occurs. 1: INTR is active low so that INTR becomes LOW when an interrupt occurs. **Note:** "INV_INTR" field of the HOST I/F block and "INT2M_LEVEL" of the MODEM I/F block must have same polarity.	1
Reserved	[28:24]	-	0x0
Reserved	[23:16]	-	0xFF
Reserved	[15:9]	-	0x0
Reserved	[8:0]	-	0x100

24.6.2 HOST I/F TEMPORARY REGISTER (HOSTIFC_TMP)

Register	address	R/W	Description	Reset Value
HOSTIFC_ TMP	0x74000008	R/W	HOST I/F Temporary Register	0x00000000

HOSTIFC_ TMP	Bit	Description	Initial State
DATA	[31:0]	Temporary register This register can be used for the design revision or the verification.	0x0000_0000

24.6.3 HOST I/F IMB REGISTER (HOSTIFC_IMB)

Register	address	R/W	Description	Reset Value
HOSTIFC_IMB	0x74000010	R	HOST I/F IMB Register	0x00000000

HOSTIFC_IMB	Bit	Description	Initial State
IMB	[31:0]	A 32-bit In-MailBox Shadow register	0x0000_0000

24.6.4 HOST I/F OMB REGISTER (HOSTIFC_OMB)

Register	Address	R/W	Description	Reset Value
HOSTIFC_OMB	0x74000014	R/W	HOST I/F OMB Register	0x00000000

HOSTIFC_OMB	Bit	Description	Initial State
OMB	[31:0]	A 32-bit Out-MailBox register	0x0000_0000

24.6.5 HOST I/F STATUS MIRRORED REGISTER (HOSTIFC_MR_STAT)

Register	Address	R/W	Description	Reset Value
HOSTIFC_MR_STAT	0x74000020	R	HOST I/F Status Mirrored Register	0x000090A2

HOSTIFC_MR_STAT	Bit	Description	Initial State
Reserved	[31:16]	-	0x0000
STAT	[15:0]	Mirrored Protocol Register of STAT[15:0]	0x90A2

24.6.6 HOST I/F STATUS1 MIRRORED REGISTER (HOSTIFC_MR_STAT1)

Register	Address	R/W	Description	Reset Value
HOSTIFC_MR_STAT1	0x74000024	R	HOST I/F Status1 Mirrored Register	0x00000002

HOSTIFC_MR_STAT1	Bit	Description	Initial State
Reserved	[31:16]	-	0x0000
STAT1	[15:0]	Mirrored Protocol Register of STAT1[15:0]	0x0002

24.6.7 HOST I/F STATUS2 REGISTER (HOSTIFC_STAT2)

Register	Address	R/W	Description	Reset Value
HOSTIFC_STAT2	0x74000028	R/W	HOST I/F Status2 Register	0x00010000

HOSTIFC_STAT2	Bit	Description	Initial State
Reserved	[31:20]	-	0x000
Reserved	[19]	-	0
RBURST_DONE	[18]	Repeated burst write done flag This flag is set when the repeated burst write is done. In order to clear the flag, write HIGH value.	0
IMB_FILLED	[17]	IMB (In-Mail Box) filled flag This flag is set when the in-mailbox is written by the modem. In order to clear this flag, HIGH value should be written in this bit.	0
OMB_EMPTY	[16]	OMB (Out-Mail Box) empty flag This flag is an inversion of the OMB_FILLED (ie, STAT1[0]).	1
Reserved	[15:8]	Reserved	0x00
RX_FIFO_OVER_RUN	[7]	The over-run flag of Local RX FIFO(L_RFIFO) in the HOST I/F	0
RX_FIFO_UNDER_RUN	[6]	The under-run flag of Local RX FIFO(L_RFIFO) in the HOST I/F	0
TX_FIFO_OVER_RUN	[5]	The over-run flag of TX FIFO(WFIFO) in the HOST I/F	0
TX_FIFO_UNDER_RUN	[4]	The under-run flag of TX FIFO(WFIFO) in the HOST I/F	0
Reserved	[3]	-	0
Reserved	[2]	-	0
Reserved	[1]	-	0
Reserved	[0]	-	0

24.6.8 HOST I/F INTERRUPT ENABLE MIRRORED REGISTER (HOSTIFC_MR_INTE)

Register	Address	R/W	Description	Reset Value
HOSTIFC_MR_INTE	0x74000030	R	HOST I/F Interrupt Enable Mirrored Register	0x00002000

HOSTIFC_MR_INTE	Bit	Description	Initial State
Reserved	[31:16]		0x0000
INTE	[15:0]	Mirrored Protocol Register of INTE[15:0]	0x2000

24.6.9 HOST I/F INTERRUPT ENABLE1 MIRRORED REGISTER (HOSTIFC_MR_INTE1)

Register	Address	R/W	Description	Reset Value
HOSTIFC_MR_INTE1	0x74000034	R	HOST I/F Interrupt Enable1 Mirrored Register	0x00000000

HOSTIFC_MR_INTE1	Bit	Description	Initial State
Reserved	[31:16]		0x0000
INTE1	[15:0]	Mirrored Protocol Register of INTE1[15:0]	0x0000

24.6.10 HOST I/F INTERRUPT ENABLE2 REGISTER (HOSTIFC_INTE2)

Register	Address	R/W	Description	Reset Value
HOSTIFC_INTE2	0x74000038	R/W	HOST I/F Interrupt Enable2 Register	0x00000000

HOSTIFC_INTE2	Bit	Description	Initial State
Reserved	[31:16]		0x0000
INTE2	[15:0]	Each bit is an interrupt enable control bit of the corresponding bit of HOSTIF_STAT2	0x0000

25 USB HOST CONTROLLER

This chapter describes the Universal Serial Bus host controller (USB) implemented in S3C6410X RISC microprocessor.

25.1 OVERVIEW

S3C6410X supports 2-port USB host interface, which are as follows:

- OHCI Rev 1.0 compatible
- USB Rev1.1 compatible
- Two down stream ports
- Support for both LowSpeed and FullSpeed USB devices

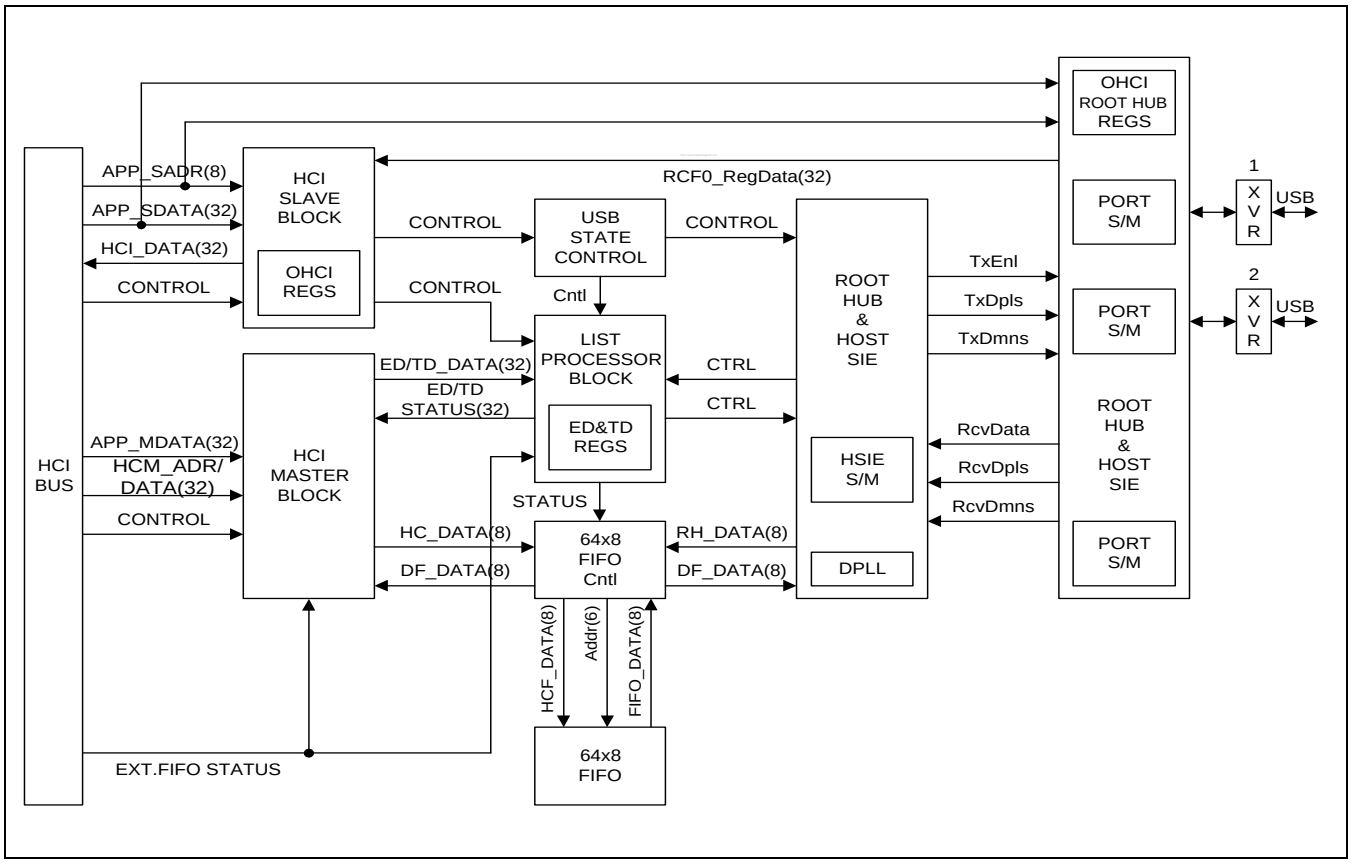


Figure 25-1. USB Host Controller Block Diagram

25.2 USB HOST CONTROLLER SPECIAL REGISTERS

The S3C6410X USB host controller complies with OHCI Rev 1.0. Refer to *Open Host Controller Interface Rev 1.0* specification for detailed information.

Table 25-1. OHCI Registers for USB Host Controller

Register	Base Address	R/W	Description	Reset Value
HcRevision	0x74300000	R	USB Host Controller Revision Register	0x0000_0010
HcControl	0x74300004	R/W	USB Host Controller Control Register	0x0000_0000
HcCommonStatus	0x74300008	R/W	USB Host Controller Command Status Register	0x0000_0000
HcInterruptStatus	0x7430000C	R/W	USB Host Controller Interrupt Status Register	0x0000_0000
HcInterruptEnable	0x74300010	R/W	USB Host Controller Interrupt Enable Register	0x0000_0000
HcInterruptDisable	0x74300014	R/W	USB Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	0x74300018	R/W	USB Host Controller HCCA Register	0x0000_0000
HcPeriodCuttentED	0x7430001C	R	USB Host Controller Period Current ED Register	0x0000_0000
HcControlHeadED	0x74300020	R/W	USB Host Controller Control Head ED Register	0x0000_0000
HcControlCurrentED	0x74300024	R/W	USB Host Controller Control Current ED Register	0x0000_0000
HcBulkHeadED	0x74300028	R/W	USB Host Controller Bulk Head ED Register	0x0000_0000
HcBulkCurrentED	0x7430002C	R/W	USB Host Controller Bulk Current ED Register	0x0000_0000
HcDoneHead	0x74300030	R	USB Host Controller Done Head Register	0x0000_0000
HcRmInterval	0x74300034	R/W	USB Host Controller FmInterval Register	0x0000_2EDF
HcFmRemaining	0x74300038	R	USB Host Controller Frame Remaining Register	0x0000_0000
HcFmNumber	0x7430003C	R	USB Host Controller Frame Number Register	0x0000_0000
HcPeriodicStart	0x74300040	R/W	USB Host Controller Periodic Start Register	0x0000_0000
HcLSThreshold	0x74300044	R/W	USB Host Controller Low-Speed Threshold Register	0x0000_0628
HcRhDescriptorA	0x74300048	R/W	USB Host Controller Root Hub Descriptor A Register	0x0200_1202
HcRhDescriptorB	0x7430004C	R/W	USB Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhStatus	0x74300050	R/W	USB Host Controller Root Hub Status Register	0x0000_0000
HcRhPortStatus1	0x74300054	R/W	USB Host Controller Root Hub Port Status 1 Register	0x0000_0100
HcRhPortStatus2	0x74300058	R/W	USB Host Controller Root Hub Port Status 2 Register	0x0000_0100

26 USB2.0 HS OTG

26.1 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.

26.2 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 SRP (Session Request Protocol) and HNP (Host Negotiation Protocol)
- 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
- 16 Host channels supportable
- Supports packet-based, dynamic FIFO memory allocation of 6,144 depths (35-bit width)

26.3 BLOCK DIAGRAM

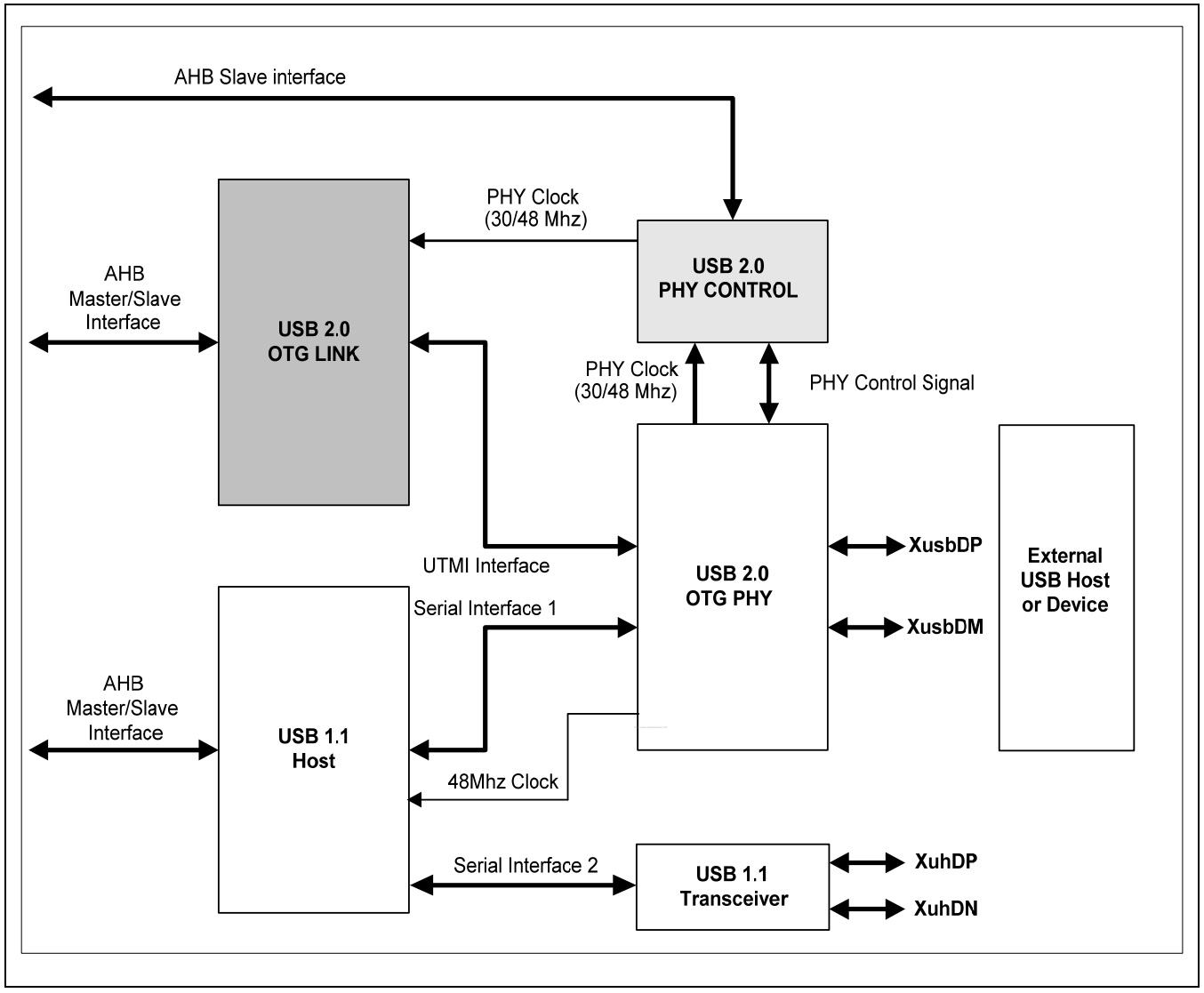


Figure 26-1. System Level Block Diagram

HS OTG controller is composed of two independent blocks, 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 S3C6410X USB system shown in Figure 26-1 can be configured as following:

1. USB 1.1 Host 1 Port & USB 2.0 OTG 1 Port
2. USB 1.1 Host 2 Ports

To enable Serial Interface 1 and use 2 ports of Host 1.1, set the OPHYCLK.serial_mode register bit to 1.

26.4 MODES OF OPERATION

The application can operate the Link either in DMA mode or in Slave mode. The application cannot operate the core using DMA and Slave modes simultaneously.

26.4.1 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.

26.4.2 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 can program the OTG to perform multiple transactions. The advantage of pipelined operation is that the application is not interrupted on packet basis.

26.5 SYSTEM CONTROLLER SETTING

A register in SYSTEM CONTROLLER has to be set for USB to work appropriately.

OTHERS	Bit	R/W	Description	Initial State
USB_SIG_MASK	[16]	R_W	USB signal mask to prevent unwanted leakage (This bit must be set before USB PHY is used.)	1'b0

The 16th bit of OTHER CONTROL register based on address 7E00_F900h is guided to be set differently depending on the system operation mode:

26.5.1 NORMAL MODE

Initial state of USB_SIG_MASK is 1'b0. It must be set to 1'b1 in order to start USB transaction.

In this mode, USB PHY power can be cut off if USBOTG function is not used.

26.5.2 STOP/DEEP STOP/SLEEP MODE

In these operation modes, USB PHY power can be cut off.

Therefore to prevent unwanted leakage current , USB_SIG_MASK must be set to 1'b0 before entering these modes.

26.6 REGISTER MAP

26.6.1 OVERVIEW

The OTG PHY control registers based on address 7C10_0000h must be accessed to control and observe the OTG PHY.

The OTG Link Core registers based on address 7C00_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

Only the Core Global and Host Port registers can be accessed in both Host and Device modes. When 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. When 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.

26.6.2 OTG LINK CSR MEMORY MAP

Figure 26-2 shows the OTG link CSR address map. Host and Device mode registers occupy different addresses. All registers are implemented in the AHB Clock domain.

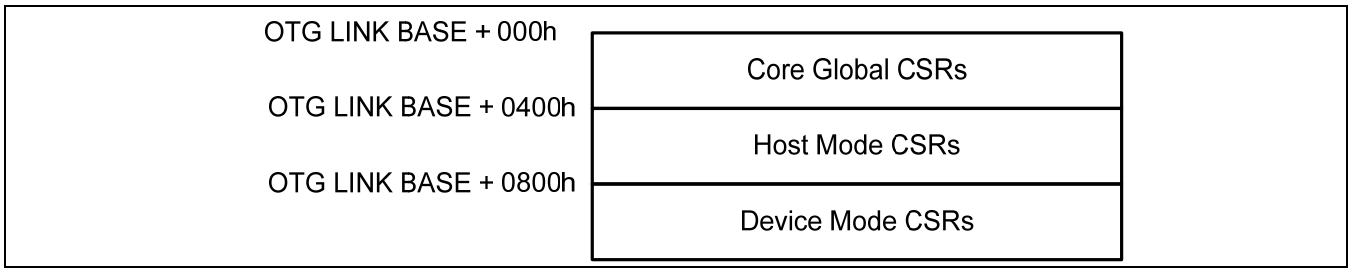


Figure 26-2. OTG Link CSR Memory Map

26.6.3 OTG FIFO ADDRESS MAPPING

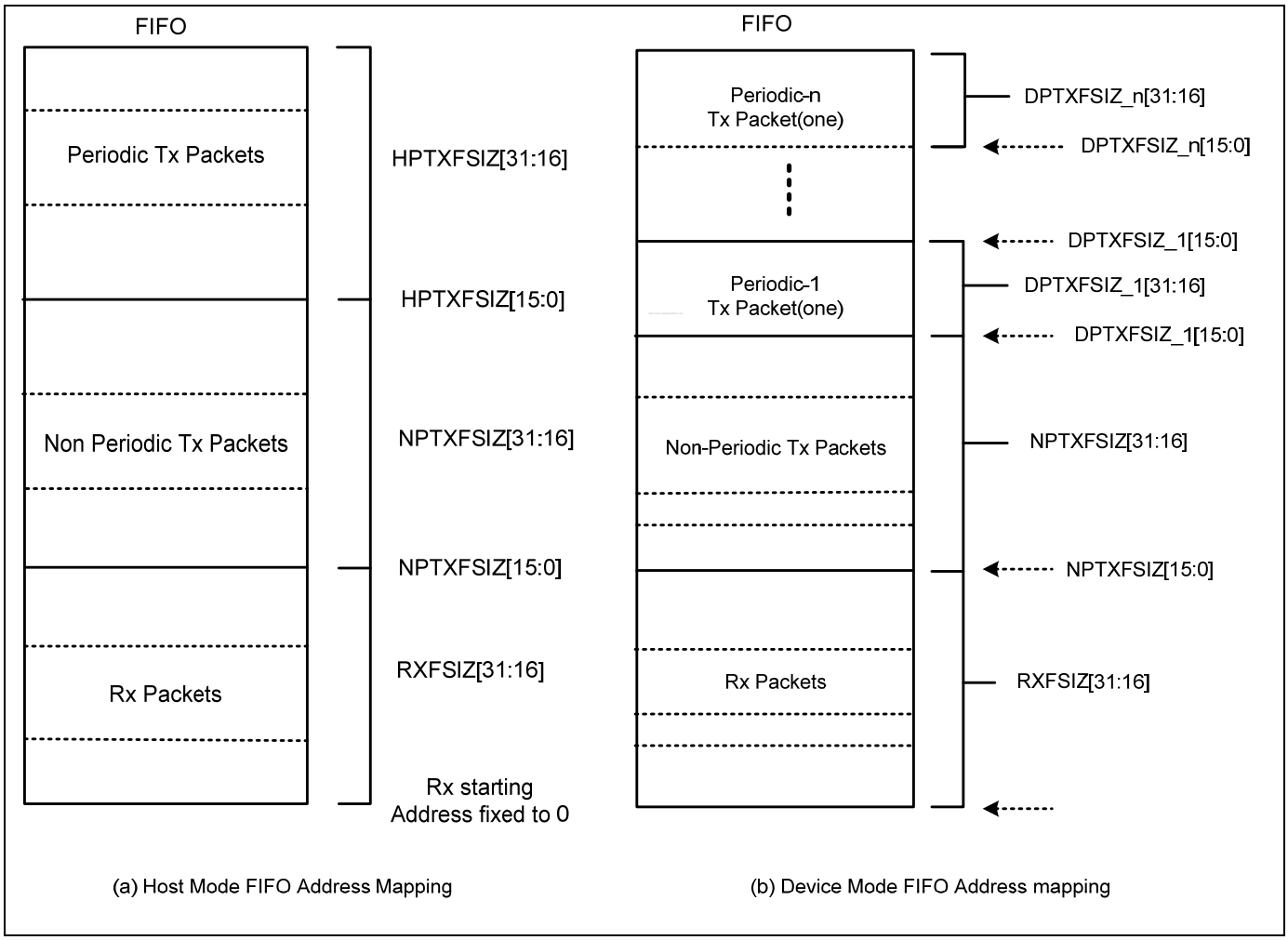


Figure 26-3. OTG FIFO Mapping

Figure 26-3 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

26.6.4 APPLICATION ACCESS TO THE CSRS

The Access column of each register description that follows specifies how the application and the core can access the register fields of the CSRs. The following conventions are used.

Read Only	RO	Register field can only be read by the application. Writes to read-only fields have no effect.
Write Only	WO	Register field can only be written by the application.
READ and Write	R_W	Register field can be read and written by the application. The application can set this field by writing 1'b1 and can clear it by writing 1'b0.
Read, Write, and Self Clear	R_W_SC	Register field can be 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 can be 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 can be read by the application (Read), can be set to 1'b1 by the core on certain internal or USB or AHB event (Self Set), and can be 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 can be read by the application (Read), can be 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 can be read by the application (Read), can be set to 1'b1 by the core on certain internal or USB or AHB events (Self Set), and can be 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.

26.7 HS OTG CONTROLLER SPECIAL REGISTERS

Table 26-1. Register summary of HS OTG Controller

Register	Offset	R/W	Description	Reset Value
OTG PHY CONTROL REGISTERS (Base address : 0x7C10_0000)				
OPHYPWR	0x000	R/W	OTG PHY Power Control Register	0x0000_0019
OPHYCLK	0x004	R/W	OTG PHY Clock Control Register	0x0000_0000
ORSTCON	0x008	R/W	OTG Reset Control Register	0x0000_0001
OPHYTUNE	0x020	R/W	OTG PHY Tuning Register	0x0027_1B93
OTG LINK CORE REGISTERS (Base address : 0x7C00_0000)				
Core Global Registers				
GOTGCTL	0x000	R/W	OTG Control and Status Register	0x0001_0000
GOTGINT	0x004	R/W	OTG Interrupt Register	0x0000_0000
GAHBCFG	0x008	R/W	Core AHB Configuration Register	0x0000_0000
GUSBCFG	0x00C	R/W	Core USB Configuration Register	0x0000_1400
GRSTCTL	0x010	R/W	Core Reset Register	0x8000_0000
GINTSTS	0x014	R/W	Core Interrupt Register	0x0400_1020
GINTMSK	0x018	R/W	Core Interrupt Mask Register	0x0000_0000
GRXSTSR	0x01C	R	Receive Status Debug Read Register	-
GRXSTSP	0x020	R	Receive Status Read/Pop Register	-
GRXFSIZ	0x024	R/W	Receive FIFO Size Register	0x0000_1800
GNPTXFSIZ	0x028	R/W	Non-Periodic Transmit FIFO Size Register	0x1800_1800
GNPTXSTS	0x02C	R	Non-Periodic Transmit FIFO/Queue Status Register	0x0008_1800
HPTXFSIZ	0x100	R/W	Host Periodic Transmit FIFO Size Register	0x0300_5A00
DPTXFSIZ1	0x104	R/W	Device Periodic Transmit FIFO-1 Size Register	0x0300_1000
DPTXFSIZ2	0x108	R/W	Device Periodic Transmit FIFO-2 Size Register	0x0300_3300
DPTXFSIZ3	0x10C	R/W	Device Periodic Transmit FIFO-3 Size Register	0x0300_3600
DPTXFSIZ4	0x110	R/W	Device Periodic Transmit FIFO-4 Size Register	0x0300_3900
DPTXFSIZ5	0x114	R/W	Device Periodic Transmit FIFO-5 Size Register	0x0300_3C00
DPTXFSIZ6	0x118	R/W	Device Periodic Transmit FIFO-6 Size Register	0x0300_3F00
DPTXFSIZ7	0x11C	R/W	Device Periodic Transmit FIFO-7 Size Register	0x0300_4200
DPTXFSIZ8	0x120	R/W	Device Periodic Transmit FIFO-8 Size Register	0x0300_4500
DPTXFSIZ9	0x124	R/W	Device Periodic Transmit FIFO-9 Size Register	0x0300_4800
DPTXFSIZ10	0x128	R/W	Device Periodic Transmit FIFO-10 Size Register	0x0300_4B00
DPTXFSIZ11	0x12C	R/W	Device Periodic Transmit FIFO-11 Size Register	0x0300_4E00
DPTXFSIZ12	0x130	R/W	Device Periodic Transmit FIFO-12 Size Register	0x0300_5100
DPTXFSIZ13	0x134	R/W	Device Periodic Transmit FIFO-13 Size Register	0x0300_5400

DPTXFSIZ14	0x138	R/W	Device Periodic Transmit FIFO-14 Size Register	0x0300_5700
DPTXFSIZ15	0x13C	R/W	Device Periodic Transmit FIFO-15 Size Register	0x0300_5A00
Host Mode Registers				
Host Global Registers				
HCFG	0x400	R/W	Host Configuration Register	0x0020_0000
HFIR	0x404	R/W	Host Frame Interval Register	0x0000_17D7
HFNUM	0x408	R	Host Frame Number/Frame Time Remaining Register	0x0000_0000
HPTXSTS	0x410	R	Host Periodic Transmit FIFO/Queue Status Register	0x0008_0100
HAINT	0x414	R	Host All Channels Interrupt Register	0x0000_0000
HAINTMSK	0x418	R/W	Host All Channels Interrupt Mask Register	0x0000_0000
Host Port Control and Status Registers				
HPRT	0x440	R/W	Host Port Control and Status Register	0x0000_0000
Host Channel-Specific Registers				
HCCHAR0	0x500	R/W	Host Channel 0 Characteristics Register	0x0000_0000
HCSPLT0	0x504	R/W	Host Channel 0 Spilt Control Register	0x0000_0000
HCINT0	0x508	R/W	Host Channel 0 Interrupt Register	0x0000_0000
HCINTMSK0	0x50C	R/W	Host Channel 0 Interrupt Mask Register	0x0000_0000
HCTSIZ0	0x510	R/W	Host Channel 0 Transfer Size Register	0x0000_0000
HCDMA0	0x514	R/W	Host Channel 0 DMA Address Register	0x0000_0000
HCCHAR1	0x520	R/W	Host Channel 1 Characteristics Register	0x0000_0000
HCSPLT1	0x524	R/W	Host Channel 1 Spilt Control Register	0x0000_0000
HCINT1	0x528	R/W	Host Channel 1 Interrupt Register	0x0000_0000
HCINTMSK1	0x52C	R/W	Host Channel 1 Interrupt Mask Register	0x0000_0000
HCTSIZ1	0x530	R/W	Host Channel 1 Transfer Size Register	0x0000_0000
HCDMA1	0x534	R/W	Host Channel 1 DMA Address Register	0x0000_0000
HCCHAR2	0x540	R/W	Host Channel 2 Characteristics Register	0x0000_0000
HCSPLT2	0x544	R/W	Host Channel 2 Spilt Control Register	0x0000_0000
HCINT2	0x548	R/W	Host Channel 2 Interrupt Register	0x0000_0000
HCINTMSK2	0x54C	R/W	Host Channel 2 Interrupt Mask Register	0x0000_0000
HCTSIZ2	0x550	R/W	Host Channel 2 Transfer Size Register	0x0000_0000
HCDMA2	0x554	R/W	Host Channel 2 DMA Address Register	0x0000_0000
HCCHAR3	0x560	R/W	Host Channel 3 Characteristics Register	0x0000_0000
HCSPLT3	0x564	R/W	Host Channel 3 Spilt Control Register	0x0000_0000
HCINT3	0x568	R/W	Host Channel 3 Interrupt Register	0x0000_0000
HCINTMSK3	0x56C	R/W	Host Channel 3 Interrupt Mask Register	0x0000_0000
HCTSIZ3	0x570	R/W	Host Channel 3 Transfer Size Register	0x0000_0000
HCDMA3	0x574	R/W	Host Channel 3 DMA Address Register	0x0000_0000

HCCHAR4	0x580	R/W	Host Channel 4 Characteristics Register	0x0000_0000
HCSPLT4	0x584	R/W	Host Channel 4 Spilt Control Register	0x0000_0000
HCINT4	0x588	R/W	Host Channel 4 Interrupt Register	0x0000_0000
HCINTMSK4	0x58C	R/W	Host Channel 4 Interrupt Mask Register	0x0000_0000
HCTSIZ4	0x580	R/W	Host Channel 4 Transfer Size Register	0x0000_0000
HCDMA4	0x584	R/W	Host Channel 4 DMA Address Register	0x0000_0000
HCCHAR5	0x5A0	R/W	Host Channel 5 Characteristics Register	0x0000_0000
HCSPLT5	0x5A4	R/W	Host Channel 5 Spilt Control Register	0x0000_0000
HCINT5	0x5A8	R/W	Host Channel 5 Interrupt Register	0x0000_0000
HCINTMSK5	0x5AC	R/W	Host Channel 5 Interrupt Mask Register	0x0000_0000
HCTSIZ5	0x5B0	R/W	Host Channel 5 Transfer Size Register	0x0000_0000
HCDMA5	0x5B4	R/W	Host Channel 5 DMA Address Register	0x0000_0000
HCCHAR6	0x5C0	R/W	Host Channel 6 Characteristics Register	0x0000_0000
HCSPLT6	0x5C4	R/W	Host Channel 6 Spilt Control Register	0x0000_0000
HCINT6	0x5C8	R/W	Host Channel 6 Interrupt Register	0x0000_0000
HCINTMSK6	0x5CC	R/W	Host Channel 6 Interrupt Mask Register	0x0000_0000
HCTSIZ6	0x5D0	R/W	Host Channel 6 Transfer Size Register	0x0000_0000
HCDMA6	0x5D4	R/W	Host Channel 6 DMA Address Register	0x0000_0000
HCCHAR7	0x5E0	R/W	Host Channel 7 Characteristics Register	0x0000_0000
HCSPLT7	0x5E4	R/W	Host Channel 7 Spilt Control Register	0x0000_0000
HCINT7	0x5E8	R/W	Host Channel 7 Interrupt Register	0x0000_0000
HCINTMSK7	0x5EC	R/W	Host Channel 7 Interrupt Mask Register	0x0000_0000
HCTSIZ7	0x5F0	R/W	Host Channel 7 Transfer Size Register	0x0000_0000
HCDMA7	0x5F4	R/W	Host Channel 7 DMA Address Register	0x0000_0000
HCCHAR8	0x600	R/W	Host Channel 8 Characteristics Register	0x0000_0000
HCSPLT8	0x604	R/W	Host Channel 8 Spilt Control Register	0x0000_0000
HCINT8	0x608	R/W	Host Channel 8 Interrupt Register	0x0000_0000
HCINTMSK8	0x60C	R/W	Host Channel 8 Interrupt Mask Register	0x0000_0000
HCTSIZ8	0x610	R/W	Host Channel 8 Transfer Size Register	0x0000_0000
HCDMA8	0x614	R/W	Host Channel 8 DMA Address Register	0x0000_0000
HCCHAR9	0x620	R/W	Host Channel 9 Characteristics Register	0x0000_0000
HCSPLT9	0x624	R/W	Host Channel 9 Spilt Control Register	0x0000_0000
HCINT9	0x628	R/W	Host Channel 9 Interrupt Register	0x0000_0000
HCINTMSK9	0x62C	R/W	Host Channel 9 Interrupt Mask Register	0x0000_0000
HCTSIZ9	0x630	R/W	Host Channel 9 Transfer Size Register	0x0000_0000
HCDMA9	0x634	R/W	Host Channel 9 DMA Address Register	0x0000_0000
HCCHAR10	0x640	R/W	Host Channel 10 Characteristics Register	0x0000_0000

HCSPLT10	0x644	R/W	Host Channel 10 Spilt Control Register	0x0000_0000
HCINT10	0x648	R/W	Host Channel 10 Interrupt Register	0x0000_0000
HCINTMSK10	0x64C	R/W	Host Channel 10 Interrupt Mask Register	0x0000_0000
HCTSIZ10	0x650	R/W	Host Channel 10 Transfer Size Register	0x0000_0000
HCDMA10	0x654	R/W	Host Channel 10 DMA Address Register	0x0000_0000
HCCHAR11	0x660	R/W	Host Channel 11 Characteristics Register	0x0000_0000
HCSPLT11	0x664	R/W	Host Channel 11 Spilt Control Register	0x0000_0000
HCINT11	0x668	R/W	Host Channel 11 Interrupt Register	0x0000_0000
HCINTMSK11	0x66C	R/W	Host Channel 11 Interrupt Mask Register	0x0000_0000
HCTSIZ11	0x670	R/W	Host Channel 11 Transfer Size Register	0x0000_0000
HCDMA11	0x674	R/W	Host Channel 11 DMA Address Register	0x0000_0000
HCCHAR12	0x680	R/W	Host Channel 12 Characteristics Register	0x0000_0000
HCSPLT12	0x684	R/W	Host Channel 12 Spilt Control Register	0x0000_0000
HCINT12	0x688	R/W	Host Channel 12 Interrupt Register	0x0000_0000
HCINTMSK12	0x68C	R/W	Host Channel 12 Interrupt Mask Register	0x0000_0000
HCTSIZ12	0x690	R/W	Host Channel 12 Transfer Size Register	0x0000_0000
HCDMA12	0x694	R/W	Host Channel 12 DMA Address Register	0x0000_0000
HCCHAR13	0x6A0	R/W	Host Channel 13 Characteristics Register	0x0000_0000
HCSPLT13	0x6A4	R/W	Host Channel 13 Spilt Control Register	0x0000_0000
HCINT13	0x6A8	R/W	Host Channel 13 Interrupt Register	0x0000_0000
HCINTMSK13	0x6AC	R/W	Host Channel 13 Interrupt Mask Register	0x0000_0000
HCTSIZ13	0x6B0	R/W	Host Channel 13 Transfer Size Register	0x0000_0000
HCDMA13	0x6B4	R/W	Host Channel 13 DMA Address Register	0x0000_0000
HCCHAR14	0x6C0	R/W	Host Channel 14 Characteristics Register	0x0000_0000
HCSPLT14	0x6C4	R/W	Host Channel 14 Spilt Control Register	0x0000_0000
HCINT14	0x6C8	R/W	Host Channel 14 Interrupt Register	0x0000_0000
HCINTMSK14	0x6CC	R/W	Host Channel 14 Interrupt Mask Register	0x0000_0000
HCTSIZ14	0x6D0	R/W	Host Channel 14 Transfer Size Register	0x0000_0000
HCDMA14	0x6D4	R/W	Host Channel 14 DMA Address Register	0x0000_0000
HCCHAR15	0x6E0	R/W	Host Channel 15 Characteristics Register	0x0000_0000
HCSPLT15	0x6E4	R/W	Host Channel 15 Spilt Control Register	0x0000_0000
HCINT15	0x6E8	R/W	Host Channel 15 Interrupt Register	0x0000_0000
HCINTMSK15	0x6EC	R/W	Host Channel 15 Interrupt Mask Register	0x0000_0000
HCTSIZ15	0x6F0	R/W	Host Channel 15 Transfer Size Register	0x0000_0000
HCDMA15	0x6F4	R/W	Host Channel 15 DMA Address Register	0x0000_0000
Device Mode Registers				
Device Global Registers				

DCFG	0x800	R/W	Device Configuration Register	0x0020_0000
DCTL	0x804	R/W	Device Control Register	0x0000_0000
DSTS	0x808	R	Device Status Register	0x0000_0002
DIEPMSK	0x810	R/W	Device IN Endpoint Common Interrupt Mask Register	0x0000_0000
DOEPMSK	0x814	R/W	Device OUT Endpoint Common Interrupt Mask Register	0x0000_0000
DAINT	0x818	R	Device ALL Endpoints Interrupt Register	0x0000_0000
DAINTMSK	0x81C	R/W	Device ALL Endpoints Interrupt Mask Register	0x0000_0000
DTKNQR1	0x820	R	Device IN Token Sequence Learning Queue Read Register 1	0x0000_0000
DTKNQR2	0x824	R	Device IN Token Sequence Learning Queue Read Register 2	0x0000_0000
DVBUSDIS	0x828	R/W	Device VBUS Discharge Time Register	0x0000_17D7
DVBUSPULSE	0x82C	R/W	Device VBUS Pulsing Time Register	0x0000_05B8
DTKNQR3	0x830	R	Device IN Token Sequence Learning Queue Read Register 3	0x0000_0000
DTKNQR4	0x834	R	Device IN Token Sequence Learning Queue Read Register 4	0x0000_0000
Device Logical IN Endpoint-Specific Registers				
DIEPCTL0	0x900	R/W	Device Control IN Endpoint 0 Control Register	0x0000_8000
DIEPINT0	0x908	R/W	Device IN Endpoint 0 Interrupt Register	0x0000_0000
DIEPTSIZ0	0x910	R/W	Device IN Endpoint 0 Transfer Size Register	0x0000_0000
DIEPDMA0	0x914	R/W	Device IN Endpoint 0 DMA Address Register	0x0000_0000
DIEPCTL1	0x920	R/W	Device Control IN Endpoint 1 Control Register	0x0000_0000
DIEPINT1	0x928	R/W	Device IN Endpoint 1 Interrupt Register	0x0000_0080
DIEPTSIZ1	0x930	R/W	Device IN Endpoint 1 Transfer Size Register	0x0000_0000
DIEPDMA1	0x934	R/W	Device IN Endpoint 1 DMA Address Register	0x0000_0000
DIEPCTL2	0x940	R/W	Device Control IN Endpoint 2 Control Register	0x0000_0000
DIEPINT2	0x948	R/W	Device IN Endpoint 2 Interrupt Register	0x0000_0080
DIEPTSIZ2	0x950	R/W	Device IN Endpoint 2 Transfer Size Register	0x0000_0000
DIEPDMA2	0x954	R/W	Device IN Endpoint 2 DMA Address Register	0x0000_0000
DIEPCTL3	0x960	R/W	Device Control IN Endpoint 3 Control Register	0x0000_0000
DIEPINT3	0x968	R/W	Device IN Endpoint 3 Interrupt Register	0x0000_0080
DIEPTSIZ3	0x970	R/W	Device IN Endpoint 3 Transfer Size Register	0x0000_0000
DIEPDMA3	0x974	R/W	Device IN Endpoint 3 DMA Address Register	0x0000_0000
DIEPCTL4	0x980	R/W	Device Control IN Endpoint 0 Control Register	0x0000_0000
DIEPINT4	0x988	R/W	Device IN Endpoint 4 Interrupt Register	0x0000_0080
DIEPTSIZ4	0x990	R/W	Device IN Endpoint 4 Transfer Size Register	0x0000_0000
DIEPDMA4	0x994	R/W	Device IN Endpoint 4 DMA Address Register	0x0000_0000

DIEPCTL5	0x9A0	R/W	Device Control IN Endpoint 5 Control Register	0x0000_0000
DIEPINT5	0x9A8	R/W	Device IN Endpoint 5 Interrupt Register	0x0000_0080
DIEPTSIZ5	0x9B0	R/W	Device IN Endpoint 5 Transfer Size Register	0x0000_0000
DIEPDMA5	0x9B4	R/W	Device IN Endpoint 5 DMA Address Register	0x0000_0000
DIEPCTL6	0x9C0	R/W	Device Control IN Endpoint 6 Control Register	0x0000_0000
DIEPINT6	0x9C8	R/W	Device IN Endpoint 6 Interrupt Register	0x0000_0080
DIEPTSIZ6	0x9D0	R/W	Device IN Endpoint 6 Transfer Size Register	0x0000_0000
DIEPDMA6	0x9D4	R/W	Device IN Endpoint 6 DMA Address Register	0x0000_0000
DIEPCTL7	0x9E0	R/W	Device Control IN Endpoint 7 Control Register	0x0000_0000
DIEPINT7	0x9E8	R/W	Device IN Endpoint 7 Interrupt Register	0x0000_0080
DIEPTSIZ7	0x9F0	R/W	Device IN Endpoint 7 Transfer Size Register	0x0000_0000
DIEPDMA7	0x9F4	R/W	Device IN Endpoint 7 DMA Address Register	0x0000_0000
DIEPCTL8	0xA00	R/W	Device Control IN Endpoint 8 Control Register	0x0000_0000
DIEPINT8	0xA08	R/W	Device IN Endpoint 8 Interrupt Register	0x0000_0080
DIEPTSIZ8	0xA10	R/W	Device IN Endpoint 8 Transfer Size Register	0x0000_0000
DIEPDMA8	0xA14	R/W	Device IN Endpoint 8 DMA Address Register	0x0000_0000
DIEPCTL9	0xA20	R/W	Device Control IN Endpoint 9 Control Register	0x0000_0000
DIEPINT9	0xA28	R/W	Device IN Endpoint 9 Interrupt Register	0x0000_0080
DIEPTSIZ9	0xA30	R/W	Device IN Endpoint 9 Transfer Size Register	0x0000_0000
DIEPDMA9	0xA34	R/W	Device IN Endpoint 9 DMA Address Register	0x0000_0000
DIEPCTL10	0xA40	R/W	Device Control IN Endpoint 10 Control Register	0x0000_0000
DIEPINT10	0xA48	R/W	Device IN Endpoint 10 Interrupt Register	0x0000_0080
DIEPTSIZ10	0xA50	R/W	Device IN Endpoint 10 Transfer Size Register	0x0000_0000
DIEPDMA10	0xA54	R/W	Device IN Endpoint 10 DMA Address Register	0x0000_0000
DIEPCTL11	0xA60	R/W	Device Control IN Endpoint 11 Control Register	0x0000_0000
DIEPINT11	0xA68	R/W	Device IN Endpoint 11 Interrupt Register	0x0000_0080
DIEPTSIZ11	0xA70	R/W	Device IN Endpoint 11 Transfer Size Register	0x0000_0000
DIEPDMA11	0xA74	R/W	Device IN Endpoint 11 DMA Address Register	0x0000_0000
DIEPCTL12	0xA80	R/W	Device Control IN Endpoint 12 Control Register	0x0000_0000
DIEPINT12	0xA88	R/W	Device IN Endpoint 12 Interrupt Register	0x0000_0080
DIEPTSIZ12	0xA90	R/W	Device IN Endpoint 12 Transfer Size Register	0x0000_0000
DIEPDMA12	0xA94	R/W	Device IN Endpoint 12 DMA Address Register	0x0000_0000
DIEPCTL13	0xAA0	R/W	Device Control IN Endpoint 13 Control Register	0x0000_0000
DIEPINT13	0xAA8	R/W	Device IN Endpoint 13 Interrupt Register	0x0000_0080
DIEPTSIZ13	0xAB0	R/W	Device IN Endpoint 13 Transfer Size Register	0x0000_0000
DIEPDMA13	0xAB4	R/W	Device IN Endpoint 13 DMA Address Register	0x0000_0000
DIEPCTL14	0xAC0	R/W	Device Control IN Endpoint 14 Control Register	0x0000_0000

DIEPINT14	0xAC8	R/W	Device IN Endpoint 14 Interrupt Register	0x0000_0080
DIEPTSIZ14	0xAD0	R/W	Device IN Endpoint 14 Transfer Size Register	0x0000_0000
DIEPDMA14	0xAD4	R/W	Device IN Endpoint 14 DMA Address Register	0x0000_0000
DIEPCTL15	0xAE0	R/W	Device Control IN Endpoint 15 Control Register	0x0000_0000
DIEPINT15	0xAE8	R/W	Device IN Endpoint 15 Interrupt Register	0x0000_0080
DIEPTSIZ15	0xAF0	R/W	Device IN Endpoint 15 Transfer Size Register	0x0000_0000
DIEPDMA15	0xAF4	R/W	Device IN Endpoint 15 DMA Address Register	0x0000_0000
Device Logical OUT Endpoint-Specific Registers				
DOEPCTL0	0xB00	R/W	Device Control OUT Endpoint 0 Control Register	0x0000_8000
DOEPINT0	0xB08	R/W	Device OUT Endpoint 0 Interrupt Register	0x0000_0000
DOEPTSIZ0	0xB10	R/W	Device OUT Endpoint 0 Transfer Size Register	0x0000_0000
DOEPDMA0	0xB14	R/W	Device OUT Endpoint 0 DMA Address Register	0x0000_0000
DOEPCTL1	0xB20	R/W	Device Control OUT Endpoint 1 Control Register	0x0000_0000
DOEPINT1	0xB28	R/W	Device OUT Endpoint 1 Interrupt Register	0x0000_0000
DOEPTSIZ1	0xB30	R/W	Device OUT Endpoint 1 Transfer Size Register	0x0000_0000
DOEPDMA1	0xB34	R/W	Device OUT Endpoint 1 DMA Address Register	0x0000_0000
DOEPCTL2	0xB40	R/W	Device Control OUT Endpoint 2 Control Register	0x0000_0000
DOEPINT2	0xB48	R/W	Device OUT Endpoint 2 Interrupt Register	0x0000_0000
DOEPTSIZ2	0xB50	R/W	Device OUT Endpoint 2 Transfer Size Register	0x0000_0000
DOEPDMA2	0xB54	R/W	Device OUT Endpoint 2 DMA Address Register	0x0000_0000
DOEPCTL3	0xB60	R/W	Device Control OUT Endpoint 3 Control Register	0x0000_0000
DOEPINT3	0xB68	R/W	Device OUT Endpoint 3 Interrupt Register	0x0000_0000
DOEPTSIZ3	0xB70	R/W	Device OUT Endpoint 3 Transfer Size Register	0x0000_0000
DOEPDMA3	0xB74	R/W	Device OUT Endpoint 3 DMA Address Register	0x0000_0000
DOEPCTL4	0xB80	R/W	Device Control OUT Endpoint 4 Control Register	0x0000_0000
DOEPINT4	0xB88	R/W	Device OUT Endpoint 4 Interrupt Register	0x0000_0000
DOEPTSIZ4	0xB90	R/W	Device OUT Endpoint 4 Transfer Size Register	0x0000_0000
DOEPDMA4	0xB94	R/W	Device OUT Endpoint 4 DMA Address Register	0x0000_0000
DOEPCTL5	0xBA0	R/W	Device Control OUT Endpoint 5 Control Register	0x0000_0000
DOEPINT5	0xBA8	R/W	Device OUT Endpoint 5 Interrupt Register	0x0000_0000
DOEPTSIZ5	0xBB0	R/W	Device OUT Endpoint 5 Transfer Size Register	0x0000_0000
DOEPDMA5	0xBB4	R/W	Device OUT Endpoint 5 DMA Address Register	0x0000_0000
DOEPCTL6	0xBC0	R/W	Device Control OUT Endpoint 6 Control Register	0x0000_0000
DOEPINT6	0xBC8	R/W	Device OUT Endpoint 6 Interrupt Register	0x0000_0000
DOEPTSIZ6	0xBD0	R/W	Device OUT Endpoint 6 Transfer Size Register	0x0000_0000
DOEPDMA6	0xBD4	R/W	Device OUT Endpoint 6 DMA Address Register	0x0000_0000
DOEPCTL7	0xBE0	R/W	Device Control OUT Endpoint 7 Control Register	0x0000_0000

DOEPINT7	0xBE8	R/W	Device OUT Endpoint 7 Interrupt Register	0x0000_0000
DOEPTSIZ7	0xBF0	R/W	Device OUT Endpoint 7 Transfer Size Register	0x0000_0000
DOEPDMA7	0xBF4	R/W	Device OUT Endpoint 7 DMA Address Register	0x0000_0000
DOEPCTL8	0xC00	R/W	Device Control OUT Endpoint 8 Control Register	0x0000_0000
DOEPINT8	0xC08	R/W	Device OUT Endpoint 8 Interrupt Register	0x0000_0000
DOEPTSIZ8	0xC10	R/W	Device OUT Endpoint 8 Transfer Size Register	0x0000_0000
DOEPDMA8	0xC14	R/W	Device OUT Endpoint 8 DMA Address Register	0x0000_0000
DOEPCTL9	0xC20	R/W	Device Control OUT Endpoint 9 Control Register	0x0000_0000
DOEPINT9	0xC28	R/W	Device OUT Endpoint 9 Interrupt Register	0x0000_0000
DOEPTSIZ9	0xC30	R/W	Device OUT Endpoint 9 Transfer Size Register	0x0000_0000
DOEPDMA9	0xC34	R/W	Device OUT Endpoint 9 DMA Address Register	0x0000_0000
DOEPCTL10	0xC40	R/W	Device Control OUT Endpoint 10 Control Register	0x0000_0000
DOEPINT10	0xC48	R/W	Device OUT Endpoint 10 Interrupt Register	0x0000_0000
DOEPTSIZ10	0xC50	R/W	Device OUT Endpoint 10 Transfer Size Register	0x0000_0000
DOEPDMA10	0xC54	R/W	Device OUT Endpoint 10 DMA Address Register	0x0000_0000
DOEPCTL11	0xC60	R/W	Device Control OUT Endpoint 11 Control Register	0x0000_0000
DOEPINT11	0xC68	R/W	Device OUT Endpoint 11 Interrupt Register	0x0000_0000
DOEPTSIZ11	0xC70	R/W	Device OUT Endpoint 11 Transfer Size Register	0x0000_0000
DOEPDMA11	0xC74	R/W	Device OUT Endpoint 11 DMA Address Register	0x0000_0000
DOEPCTL12	0xC80	R/W	Device Control OUT Endpoint 12 Control Register	0x0000_0000
DOEPINT12	0xC88	R/W	Device OUT Endpoint 12 Interrupt Register	0x0000_0000
DOEPTSIZ12	0xC90	R/W	Device OUT Endpoint 12 Transfer Size Register	0x0000_0000
DOEPDMA12	0xC94	R/W	Device OUT Endpoint 12 DMA Address Register	0x0000_0000
DOEPCTL13	0xCA0	R/W	Device Control OUT Endpoint 13 Control Register	0x0000_0000
DOEPINT13	0xCA8	R/W	Device OUT Endpoint 13 Interrupt Register	0x0000_0000
DOEPTSIZ13	0xCB0	R/W	Device OUT Endpoint 13 Transfer Size Register	0x0000_0000
DOEPDMA13	0xCB4	R/W	Device OUT Endpoint 13 DMA Address Register	0x0000_0000
DOEPCTL14	0xCC0	R/W	Device Control OUT Endpoint 14 Control Register	0x0000_0000
DOEPINT14	0xCC8	R/W	Device OUT Endpoint 14 Interrupt Register	0x0000_0000
DOEPTSIZ14	0xCD0	R/W	Device OUT Endpoint 14 Transfer Size Register	0x0000_0000
DOEPDMA14	0xCD4	R/W	Device OUT Endpoint 14 DMA Address Register	0x0000_0000
DOEPCTL15	0xCE0	R/W	Device Control OUT Endpoint 15 Control Register	0x0000_0000
DOEPINT15	0xCE8	R/W	Device OUT Endpoint 15 Interrupt Register	0x0000_0000
DOEPTSIZ15	0xCF0	R/W	Device OUT Endpoint 15 Transfer Size Register	0x0000_0000
DOEPDMA15	0xCF4	R/W	Device OUT Endpoint 15 DMA Address Register	0x0000_0000
Power and Clock Gating Register				
PCGCCTL	0xE00	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.

26.8 OTG PHY CONTROL REGISTERS

OTG PHY Power Control Register (OPHYPWR)

Register	Address	R/W	Description	Reset Value
OPHYPWR	0x7C10_0000	R/W	OTG PHY Power Control Register	0x00000019

OPHYPWR	Bit	R/W	Description	Initial State
Reserved	[31:5]		Reserved	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, you can set this input high to save power.	1'b1
analog_powerdown	[3]	R_W	Analog block power down in PHY2.0 • 1'b0 : Analog block power up (Normal Operation) • 1'b1 : Analog block power down	1'b1
Reserved	[2:1]		Reserved	2'b00
force_suspend	[0]	R_W	Apply Suspend signal for power save • 1'b0 : disable (Normal Operation) • 1'b1 : enable	1'b1

OTG PHY Clock Control Register (OPHYCLK)

Register	Address	R/W	Description	Reset Value
OPHYCLK	0x7C10_0004	R/W	OTG PHY Control Register	0x00000000

OPHYCLK	Bit	R/W	Description	Initial State
Reserved	[31:7]		Reserved	25'h0
serial_mode	[6]	R_W	UTMI/Serial Interface Select When 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.	1'b0
xo_ext_clk_enb	[5]	R_W	Reference Clock Select for XO Block • 1'b0: external crystal • 1'b1: external clock/oscillator	1'b0
common_on_n	[4]	R_W	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 when 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.	1'b0
Reserved	[3]		Reserved	1'b0
id_pullup	[2]	R_W	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 which type of plug is connected.)	1'b0
clk_sel	[1:0]	R_W	Reference Clock Frequency Select for PLL • 2'b00 : 48MHz • 2'b01 : Reserved • 2'b10 : 12MHz • 2'b11 : 24MHz	2'b00

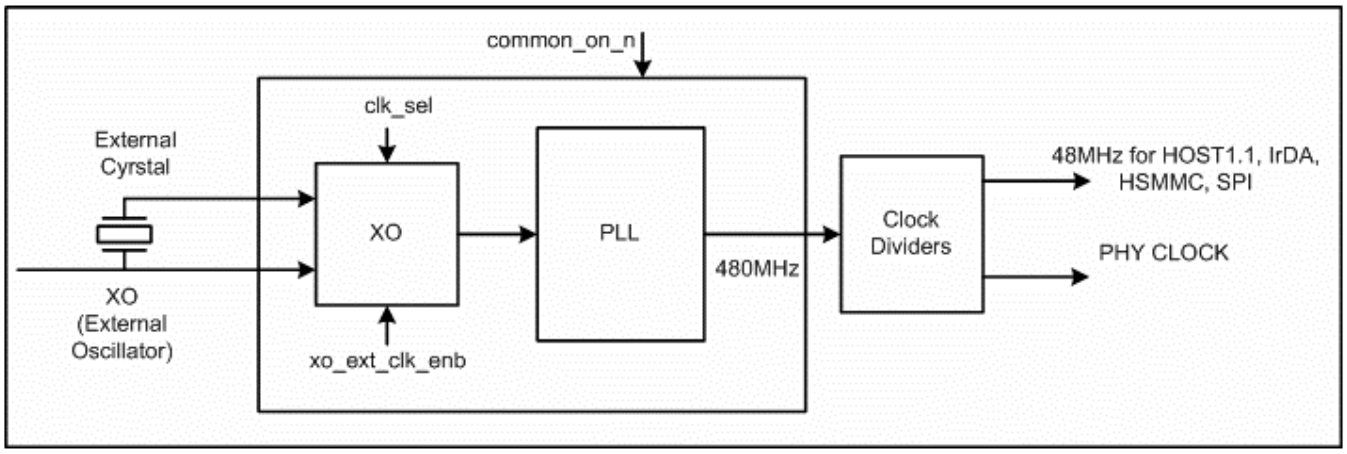


Figure 26-4. OTG PHY Clock Path

OTG Reset Control Register (ORSTCON)

Register	Address	R/W	Description	Reset Value
ORSTCON	0x7C10_0008	R/W	OTG Reset Control Register	0x00000001

ORSTCON	Bit	R/W	Description	Initial State
Reserved	[31:3]		Reserved	29'h0
phylnk_sw_rst	[2]	R_W	OTG Link Core phy_clock domain S/W Reset	1'b0
link_sw_rst	[1]	R_W	OTG Link Core hclk domain S/W Reset	1'b0
phy_sw_rst	[0]	R_W	OTG PHY 2.0 S/W Reset The phy_sw_rst signal must be asserted for at least 10us	1'b1

PHY Tune Register(PHY_TUNE)

Register	Address	R/W	Description	Reset Value
OPHYTUNE	0x7C10_0020	R_W	OTG PHY Tuning Register	0x00271B93

OPHYTUNE	Bit	R/W	Description	Initial State
Reserved	[31:21]		Reserved	11'h1
txpreemphasistune	[20]	R_W	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'b0

			• 1 : The HS Transmitter pre-emphasis is enabled. • 0 : The HS Transmitter pre_empahsis is disabled.	
compdistune	[19:17]	R_W	Disconnect Threshold Adjustment. This bus 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%	3'b011
otgtune	[16:14]	R_W	VBUS Valid Threshold Adjustment. This bus adjusts the voltage level for the VBUS Valid threshold. • 111 : +9% • 110 : +6% • 101 : +3% • 100 : Design default • 011 : -3% • 010 : -6% • 001 : -9% • 000 : -12%	3'b100
sqrxtune	[13:11]	R_W	Squelch Threshold Tune. This bus 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%	3'b011
txfslstune	[10:7]	R_W	FS/LS Pull-Up Resistance Adjustment. This bus 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%	4'b0111
txrisetune	[6]	R_W	HS Transmitter Rise/Fall Time Adjustment. This bus	1'b0

			adjusts the rise/fall times of the high speed waveform. • 1 : -8% • 0 : Design default	
txhsxvtune	[5:4]	R_W	Transmitter High-Speed Crossover adjustment. This bus 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	2'b01
txvreftune	[3:0]	R_W	HS DC Voltage Level Adjustment. This bus adjusts the voltage 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 :	4'b0011

26.9 OTG LINK CORE REGISTERS

26.9.1 OTG GLOBAL REGISTERS

These registers are available in both Host and Device modes, and not required to be reprogrammed when switching between these modes.

OTG Control and Status Register (GOTGCTL)

The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.

Register	Address	R/W	Description	Reset Value
GOTGCTL	0x7C00_0000	R/W	OTG Control and Status Register	0x00010000

GOTGCTL	Bit	R/W	Description	Initial State
Reserved	[31:20]		Reserved	12'h0
BSesVld	[19]	RO	B-Session Valid Indicates the Device mode transceiver status. • 1'b0 : B-session is not valid • 1'b1 : B-session is valid	1'b0
ASesVld	[18]	RO	A-Session Valid Indicates the Host mode transceiver status. • 1'b0 : A-session is not valid • 1'b1 : A-session is valid	1'b0
DbncTime	[17]	RO	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	1'b0
ConIDSts	[16]	RO	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	1'b1
Reserved	[15:12]		Reserved	4'h0
DevHNPEn	[11]	R_W	Device HNP Enable The application sets the bit when it successfully receives a SetFeature. • 1'b0 : HNP is not enabled in the application • 1'b1 : HNP is enabled in the application	1'b0
HstSetHNPEn	[10]	R_W	Host Set HNP Enable The application sets this bit when it has successfully enabled HNP on the connected device. • 1'b0 : Host Set HNP is not enabled • 1'b1 : Host Set HNP is enabled	1'b0
HNPReq	[9]	R_W	HNP Request The application sets this bit to initiate an HNP request to the connected USB host. The core clears this bit when the HstNegSucStsChng bit is cleared. • 1'b0 : No HNP request • 1'b1 : HNP request	1'b0

GOTGCTL	Bit	R/W	Description	Initial State
HstNegScs	[8]	RO	Host Negotiation Success The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. • 1'b0 : Host negotiation failure • 1'b1 : Host negotiation success	1'b0
Reserved	[7:2]		Reserved	6'h0
SesReq	[1]	R_W	Session Request The application sets this bit to initiate a session request on the USB. The core clears this bit when the HstNegSucStsChng bit is cleared. • 1'b0 : No session request • 1'b1 : Session request	1'b0
SesReqScs	[0]	RO	Session Request Success The core sets this bit when a session request initiation is successful. • 1'b0 : Session request failure • 1'b1 : Session request success	1'b0

OTG Interrupt Register (GOTGINT)

The application reads this register whenever there is an OTG interrupt and clears the bits in this register to clear the OTG interrupt.

Register	Address	R/W	Description	Reset Value
GOTGINT	0x7C00_0004	R/W	OTG Interrupt Register	0x00000000

GOTGINT	Bit	R/W	Description	Initial State
Reserved	[31:20]		Reserved	12'h0
DbnceDone	[19]	R_SS_WC	Debounce Done The core sets this bit when the debounce is completed after the device connects. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register.	1'b0
ADevTOUTChg	[18]	R_SS_WC	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.	1'b0
HstNegDet	[17]	R_SS_WC	Host Negotiation Detected. The core sets this bit when it detects a host negotiation request on the USB.	1'b0
Reserved	[16:10]		Reserved	7'h0
HstnegSucStsChng	[9]	R_SS_WC	Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request.	1'b0
SesReqSucStsChng	[8]	R_SS_WC	Session Request Success Status Change The core sets this bit on the success or failure of a session request.	1'b0
Reserved	[7:3]		Reserved	5'h0
SesEndDet	[2]	R_SS_WC	Session End Detected The core sets this bit when the b_valid signal is deasserted.	1'b0
Reserved	[1:0]		Reserved	2'h0

OTG AHB Configuration Register (GAHBCFG)

This register can be used to configure 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.

Register	Address	R/W	Description	Reset Value
GAHBCFG	0x7C00_0008	R/W	Core AHB Configuration Register	0x00000000

GAHBCFG	Bit	R/W	Description	Initial State
Reserved	[31:9]		Reserved	23'h0
PTxFEmpLvl	[8]	R_W	Periodic TxFIFO Empty Level Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (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	1'b0
NPTxFEmpLvl	[7]	R_W	Non-Periodic TxFIFO Empty Level Indicates when the Non-Periodic TxFIFO Empty Interrupt bit 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	1'b0
Reserved	[6]		Reserved	1'b0
DMAEn	[5]	R_W	DMA Enable • 1'b0 : Core operates in Slave mode • 1'b1 : Core operates in a DMA mode	1'b0
HBstLen	[4:1]	R_W	Burst Length/Type Internal DMA Mode – AHB Master burst type: • 4'b0000 : Single • 4'b0001 : INCR • 4'b0011 : INCR4 • 4'b0101 : INCR8 • 4'b0111 : INCR16 • Others : Reserved	4'b0
GlblIntrMsk	[0]	R_W	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	1'b0

OTG USB Configuration Register (GUSBCFG)

This register can be used to configure 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.

Register	Address	R/W	Description	Reset Value
GUSBCFG	0x7C00_000C	R/W	Core USB Configuration Register	0x00001400

GUSBCFG	Bit	R/W	Description	Initial State
Reserved	[31:16]		Reserved	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 can 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]		Reserved	5'h5
HNPCap	[9]	R_W	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	1'b0
SRPCap	[8]	R_W	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	1'b0
Reserved	[7:4]		Reserved	4'h0
PHYIf	[3]	R_W	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	1'b0
TOutCal	[2:0]	R_W	HS/FS Timeout Calibration Set this bit to 3'h7.	3'h0

Core Reset Register (GRSTCTL)

The application uses this register to reset various hardware features inside the core.

Register	Address	R/W	Description	Reset Value
GRSTCTL	0x7C00_0010	R/W	Core Reset Register	0x80000000

GRSTCTL	Bit	R/W	Description	Initial State
AHBIdle	[31]	RO	AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.	1'b1
DMAReq	[30]	RO	DMA Request Signal Indicates that the DMA request is in progress. Used for debug.	1'b0
Reserved	[29:11]		Reserved	19'h0
TxFNum	[10:6]	R_W	TxFIFO Number This is the FIFO number that must be flushed using the TxFIFO Flush bit. 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	5'h0
TxFFlsh	[5]	R_W S_SC	TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst 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.	1'b0
RxFFlsh	[4]	R_W S_SC	RxFIFO Flush The application can flush 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 will take 8 clocks to clear.	1'b0
INTknQFlsh	[3]	R_W S_SC	IN Token Sequence Learning Queue Flush The application writes this bit to flush the IN Token Sequence Learning Queue.	1'b0
FrmCntrRst	[2]	R_W S_SC	Host Frame Counter Reset The application writes this bit to reset the (micro) frame number counter inside the core. When the (micro) frame counter is reset, the subsequent SOF sent out by the core will have a (micro)frame number of 0.	1'b0

GRSTCTL	Bit	R/W	Description	Initial State
HSftRst	[1]	R_WS_SC	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 can get the status of any core events that occurred after it set this bit. 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.	1'b0
CSftRst	[0]	R_WS_SC	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 also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When 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.	1'b0

Core Interrupt Register (GINTSTS)

This register interrupts the application for system-level events in the current mode of operation (Device mode or Host mode).

Register	Address	R/W	Description	Reset Value
GINTSTS	0x7C00_0014	R/W	Core Interrupt Register	0x04001020

GINTSTS	Bit	R/W	Description	Initial State
WkUpInt	[31]	R_SS_WC	Resume/Remote Wakeup Detected Interrupt In Device mode, this interrupt is asserted when a resume is detected on the USB. In Host mode, this interrupt is asserted when a remote wakeup is detected on the USB.	1'b0
SessReqInt	[30]	R_SS_WC	Session Request/New Session Detected Interrupt In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the b_valid signal goes high.	1'b0
DisconnInt	[29]	R_SS_WC	Disconnect Detected Interrupt Asserted when a device disconnect is detected.	1'b0
ConIDSts Chng	[28]	R_SS_WC	Connector ID Status Change The core sets this bit when there is a change in connector ID status.	1'b0
Reserved	[27]		Reserved	1'b0
PTxFEmp	[26]	RO	Periodic TxFIFO Empty Asserted when 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.	1'b1
HChInt	[25]	RO	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.	1'b0
PrtInt	[24]	RO	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.	1'b0
Reserved	[23]		Reserved	1'b0

GINTSTS	Bit	R/W	Description	Initial State
FetSusp	[22]	R_SS_WC	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 can check the GINSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the “IN token received when FIFO empty” interrupt when clearing a global IN NAK handshake.	1’b0
incompIP	[21]	R_SS_WC	Incomplete Periodic Transfer. In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe.	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 completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	
Incompl SOIN	[20]	R_SS_WC	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 completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	1’b0
OEPInt	[19]	RO	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.	1’b0

GINTSTS	Bit	R/W	Description	Initial State
IEPInt	[18]	RO	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.	1'b0
EPMis	[17]	R_SS_WC	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.	1'b0
Reserved	[16]		Reserved	1'b0
EOPF	[15]	R_SS_WC	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.	1'b0
ISOutDrop	[14]	R_SS_WC	Isochronous OUT Packet Dropped Interrupt The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO doesn't have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	1'b0
EnumDone	[13]	R_SS_WC	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.	1'b0
USBRst	[12]	R_SS_WC	USB Reset The core sets this bit to indicate that a reset is detected on the USB.	1'b1
USBSusp	[11]	R_SS_WC	USB Suspend The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state when there is no activity on the line_state signal for an extended period of time.	1'b0
ErlySusp	[10]	R_SS_WC	Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.	1'b0
Reserved	[9]		Reserved	1'b0
Reserved	[8]		Reserved	1'b0
GOUTNakEff	[7]	RO	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 can be cleared by writing the Clear Global OUT NAK bit in the Device Control register.	1'b0

GINTSTS	Bit	R/W	Description	Initial State
GINNakEff	[6]	RO	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 can be cleared by clearing the Clear Global Non-Periodic IN NAK bit set by the application. This bit can be 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.	1'b0
NPTxFEmp	[5]	RO	Non-Periodic TxFIFO Empty This interrupt is asserted when 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).	1'b1
RxFLvl	[4]	RO	RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.	1'b0
Sof	[3]	R_SS_WC	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 can read the Device Status register to get the current (micro) frame number. This interrupt is seen only when the core is operating at either HS or FS.	1'b0
OTGInt	[2]	RO	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.	1'b0
ModeMis	[1]	R_SS_WC	Mode Mismatch Interrupt The core sets this bit when the application is trying to access: • A Host mode register, when the core is operating in Device mode • A Device mode register, when the core is operating in Host mode	1'b0
CurMod	[0]	RO	Current Mode Of Operation Indicates the current mode of operation. • 1'b0 : Device mode • 1'b1 : Host mode	1'b0

Core Interrupt Mask Register (GINTMSK)

This register works with the Core Interrupt register to interrupt the application. When 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

Register	Address	R/W	Description	Reset Value
GINTMSK	0x7C00_0018	R/W	Core Interrupt Mask Register	0x00000000

GINTMSK	Bit	R/W	Description	Initial State
WkUpIntMsk	[31]	R_W	Resume/Remote Wakeup Detected Interrupt Mask	1'b0
SessReqIntMsk	[30]	R_W	Session Request/New Session Detected Interrupt Mask	1'b0
DisconnIntMsk	[29]	R_W	Disconnect Detected Interrupt Mask	1'b0
ConIDStsChngMsk	[28]	R_W	Connector ID Status Change Mask	1'b0
Reserved	[27]		Reserved	1'b0
PTxFEmpMsk	[26]	R_W	Periodic TxFIFO Empty Mask	1'b0
HChIntMsk	[25]	R_W	Host Channels Interrupt Mask	1'b0
PrtIntMsk	[24]	R_W	Host Port Interrupt Mask	1'b0
Reserved	[23]		Reserved	1'b0
FetSuspMsk	[22]	R_W	Data Fetch Suspended Mask	1'b0
incomplPMsk	[21]	R_W	Incomplete Periodic Transfer Mask	1'b0
incompISOOUTMsk			Incomplete Isochronous OUT Transfer Mask	
incompISOINMsk	[20]	R_W	Incomplete Isochronous IN Transfer Mask	1'b0
OEPIntMsk	[19]	R_W	OUT Endpoints Interrupt Mask	1'b0
INEPIntMsk	[18]	R_W	IN Endpoints Interrupt Mask	1'b0
EPMisMsk	[17]	R_W	Endpoint Mismatch Interrupt Mask	1'b0
Reserved	[16]		Reserved	1'b0
EOPFMsk	[15]	R_W	End of Periodic Frame Interrupt Mask	1'b0
ISOOutDropMsk	[14]	R_W	Isochronous OUT Packet Dropped Interrupt Mask	1'b0
EnumDoneMsk	[13]	R_W	Enumeration Done Mask	1'b0
USBRstMsk	[12]	R_W	USB Reset Mask	1'b0
USBSuspMsk	[11]	R_W	USB Suspend Mask	1'b0
ErlySuspMsk	[10]	R_W	Early Suspend Mask	1'b0
Reserved	[9]		Reserved	1'b0
Reserved	[8]		Reserved	1'b0
GOUTNakEffMsk	[7]	R_W	Global OUT NAK Effective Mask	1'b0
GINNakEffMsk	[6]	R_W	Global Non-Periodic IN NAK Effective Mask	1'b0
NPTxFEmpMsk	[5]	R_W	Non-Periodic TxFIFO Empty Mask	1'b0
RxFLvlMsk	[4]	R_W	Receive FIFO Non-Empty Mask	1'b0
SofMsk	[3]	R_W	Start of (micro)Frame Mask	1'b0
OTGIntMsk	[2]	R_W	OTG Interrupt Mask	1'b0
ModeMisMsk	[1]	R_W	Mode Mismatch Interrupt Mask	1'b0
Reserved	[0]		Reserved	1'b0

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 when the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

Host Mode Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

Register	Address	R/W	Description	Reset Value
GRXSTSR/ GRXSTSP	0x7C00_001C 0x7C00_0020	R	Host Mode Receive Status Debug Read/ Status Read and Pop Registers	-

GRXSTSR/ GRXSTSP	Bit	R/W	Description	Initial State
Reserved	[31:21]		Reserved	-
PktSts	[20:17]	RO	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	-
DPID	[16:15]	RO	Data PID Indicates the Data PID of the received packet. • 2'b00 : DATA0 • 2'b10 : DATA1 • 2'b01 : DATA2 • 2'b11 : MDATA	-
BCnt	[14:4]	RO	Byte Count Indicates the byte count of the received IN data packet.	-
ChNum	[3:0]	RO	Channel number Indicates the channel number to which the current received packet belongs.	-

Device Mode Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

Register	Address	R/W	Description	Reset Value
GRXSTSR/ GRXSTSP	0x7C00_001C 0x7C00_0020	R	Device Mode Receive Status Debug Read/Status Read and Pop Registers	0xFFFFFFFF

GRXSTSR/ GRXSTSP	Bit	R/W	Description	Initial State
Reserved	[31:25]		Reserved	7'h3F
FN	[24:21]	RO	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 only when isochronous OUT endpoints are supported.	4'hF
PktSts	[20:17]	RO	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	4'b1111
DPID	[16:15]	RO	Data PID Indicates the Data PID of the received OUT data packet. • 2'b00 : DATA0 • 2'b10 : DATA1 • 2'b01 : DATA2 • 2'b11 : MDATA	2'b11
BCnt	[14:4]	RO	Byte Count Indicates the byte count of the received data packet.	11'h3FF
EPNum	[3:0]	RO	Endpoint number Indicates the endpoint number to which the current received packet belongs.	4'hF

Receive FIFO Size Register (GRXFSIZ)

The application can program the RAM size that must be allocated to the RxFIFO.

Register	Address	R/W	Description	Reset Value
GRXFSIZ	0x7C00_0024	R/W	Receive FIFO Size Register	0x00001800

GRXFSIZ	Bit	R/W	Description	Initial State
Reserved	[31:16]		Reserved	16'h0
RxFDep	[15:0]	R_W	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.	16'h1800

Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ)

The application can program the RAM size and the memory start address for the Non-Periodic TxFIFO.

Register	Address	R/W	Description	Reset Value
GNPTXFSIZ	0x7C00_0028	R/W	Non-periodic Transmit Size FIFO Register	0x18001800

GNPTXFSIZ	Bit	R/W	Description	Initial State
NPTxFDep	[31:16]	R_W	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.	16'h1800
NPTxFStAddr	[15:0]	R_W	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.	16'h1800

Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS)

This read-only register contains the free space information for the Non-Periodic TxFIFO and the Non-Periodic Transmit Request Queue.

Register	Address	R/W	Description	Reset Value
GNPTXSTS	0x7C00_002C	R	Non-periodic Transmit FIFO/Queue Status Register	0x00081800

GNPTXSTS	Bit	R/W	Description	Initial State
Reserved	[31]		Reserved	1'b0
NPTxQTop	[30:24]	RO	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)	7'h0
NPTxQSpcAvail	[23:16]	RO	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	8'h08
NPTxFSpcAvail	[15:0]	RO	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	16'h1800

Host Periodic Transmit FIFO Size Register (HPTXFSIZ)

This register holds the size and the memory start address of the Periodic TxFIFO.

Register	Address	R/W	Description	Reset Value
HPTXFSIZ	0x7C00_0100	R/W	Host Periodic Transmit FIFO SIZE Register	0x03005A00

HPTXFSIZ	Bit	R/W	Description	Initial State
PTxFSize	[31:16]	R_W	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.	16'h0300
PTxFStAddr	[15:0]	R_W	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, you can write their sum in this field. Programmed values must not exceed the power-on value.	16'h5A00

Device Periodic Transmit FIFO-N Size Register (DPTXFSIZN)

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.

Register	Address	R/W	Description	Reset Value
DPTXFSIZn	0x7C00_0104 +(n-1)*04h	R/W	Device Periodic Transmit FIFO - n Size Register	0x0300XXXX

DPTXFSIZn	Bit	R/W	Description	Initial State
DPTxFSize	[31:16]	R_W	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. You can write a new value to this field.	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]	R_W	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 you can write their sum in this field. Programmed values must not exceed the power-on value set.	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)

26.9.2 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 can be categorized as follows:

- Host Global registers
- Host Port Control and Status registers
- Host Channel-Specific registers

26.9.2.1 HOST GLOBAL REGISTERS

Host Configuration Register (HCFG)

This register configures the core after power-on. Do not make changes to this register after initializing the host.

Register	Address	R/W	Description	Reset Value
HCFG	0x7C00_0400	R/W	OTG Control and Status Register	0x00200000

HCFG	Bit	R/W	Description	Initial State
Reserved	[31:3]		Reserved	29'h0040000
FSLSSupp	[2]	R_W	FS- and LS- Only Support The application uses this bit to control the core's enumeration speed. Using this bit, the application can make 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	1'b0
FSLSPclkSel	[1:0]	R_W	FS/LS PHY Clock Select When the core is in FS Host mode • 2'b00 : PHY clock is 30/60 MHz • 2'b01 : PHY clock is 48 MHz • Others : Reserved When 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	2'b0

Host Frame Interval Register (HFIR)

This register stores the frame interval information for the current speed to which the core has enumerated

Register	Address	R/W	Description	Reset Value
HFIR	0x7C00_0404	R/W	Host Frame Interval Register	0x000017D7

HFNUM	Bit	R/W	Description	Initial State
Reserved	[31:16]		Reserved	16'h0
FrInt	[15:0]	R/W	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 when 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)	16'h17D7

Host Frame Number/Frame Time Remaining Register (HFNUM)

This register indicates the current frame number. It also indicates the time remaining in the current frame.

Register	Address	R/W	Description	Reset Value
HFNUM	0x7C00_0408	R	Host Frame Number/Frame Time Remaining Register	0x00000000

HFNUM	Bit	R/W	Description	Initial State
FrRem	[31:16]	RO	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. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.	16'h0
FrNum	[15:0]	RO	Frame Number This field increment when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF.	16'h0

Host Periodic Transmit FIFO/QUEUE Status Register (HPTXSTS)

This read-only register contains the free space information for the Periodic TxFIFO and the Periodic Transmit Request Queue.

Register	Address	R/W	Description	Reset Value
HPTXSTS	0x7C00_0410	R	Host Periodic Transmit FIFO/Queue Status Register	0x00080100

HPTXSTS	Bit	R/W	Description	Initial State
PTxQTop	[31:24]	RO	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	8'h0
PTxQSpcAvail	[23:16]	RO	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	8'h8
PTxFSpcAvail	[15:0]	RO	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	16'h0100

Host All Channels Interrupt Register (HAINT)

When 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 when the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

Register	Address	R/W	Description	Reset Value
HAINT	0x7C00_0414	R	Host All Channels Interrupt Register	0x00000000

HAINT	Bit	R/W	Description	Initial State
Reserved	[31:16]		Reserved	16'h0
HAINT	[15:0]	RO	Channel Interrupts One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	16'h0

Host All Channels Interrupt Mask Register (HAINTMSK)

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application when 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

Register	Address	R/W	Description	Reset Value
HAINTMSK	0x7C00_0418	R/W	Host All Channels Interrupt Mask Register	0x00000000

HAINTMSK	Bit	R/W	Description	Initial State
Reserved	[31:16]		Reserved	16'h0
HAINTMsk	[15:0]	R_W	Channel Interrupt Mask One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	16'h0

26.9.2.2 HOST Port Control and Status RegiSters

Host Port Control and Status Register (HPRT)

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 can trigger 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.

Register	Address	R/W	Description	Reset Value
HPRT	0x7C00_0440	R/W	Host Port Control and Status Register	0x00000000

HPRT	Bit	R/W	Description	Initial State
Reserved	[31:19]		Reserved	13'h0
PrtSpd	[18:17]	RO	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	2'b0
PrtTstCtl	[16:13]	R_W	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	4'h0
PrtPwr	[12]	R_W_SC	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	1'b0
PrtLnSts	[11:10]	RO	Port Line Status Indicates the current logic level USB data lines • Bit [10] : Logic level of D– • Bit [11] : Logic level of D+	2'b0
Reserved	[9]		Reserved	1'b0

HPRT	Bit	R/W	Description	Initial State
PrtRst	[8]	R_W	Port Reset When 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	1'b0
prtSusp	[7]	R_W S_SC	Port Suspend The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which will assert 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	1'b0
PrtRes	[6]	R_W_ SS_S C	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 when 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	1'b0
PrtOvr CurrChng	[5]	R_SS _WC	Port Overcurrent Change The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.	1'b0
PrtOvr CurrAct	[4]	RO	Port Overcurrent Active Indicates the overcurrent condition of the port. • 1'b0 : No overcurrent condition • 1'b1 : Overcurrent condition	1'b0
PrtEnChng	[3]	R_SS _WC	Port Enable/Disable Change The core sets this bit when the status of the Port Enable bit [2] of this register changes.	1'b0

HPRT	Bit	R/W	Description	Initial State
PrtEna	[2]	R_SS_SC_WC	Port Enable A port is enabled only 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 can only clear it to disable the port. This bit does not trigger any interrupt to the application. • 1'b0 : Port disabled • 1'b1 : Port enabled	1'b0
PrtConnDet	[1]	R_SS_WC	Port Connect Detected The core sets this bit when 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.	1'b0
PrtConnSts	[0]	RO	Port Connect Status • 1'b0 : No device is attached to the port • 1'b1 : A device is attached to the port	1'b0

26.9.2.3 HOST CHANNEL-SPECIFIC RegiSters

Host Channel-N Characteristics Register (HCCHARN)

Channel_number: 0≤n≤15

Register	Address	R/W	Description	Reset Value
HCCHARn	0x7C00_0500 +n*20h	R/W	Host Channel-n Characteristics Register	0x00000000

HCCHARn	Bit	R/W	Description	Initial State
ChEna	[31]	R_WS_SC	Channel Enable This field is set by the application and cleared by the OTG host. • 1'b0 : Channel disabled • 1'b1 : Channel enabled	1'b0
ChDis	[30]	R_WS_SC	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.	1'b0

HCCHARn	Bit	R/W	Description	Initial State
OddFrm	[29]	R_W	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	1'b0
DevAddr	[28:22]	R_W	Device Address This field selects the specific device serving as the data source or sink.	7'h0
MC/EC	[21:20]	R_W	Multi Count/Error Count When 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 When 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.	2'b0
EPType	[19:18]	R_W	Endpoint Type Indicates the transfer type selected. • 2'b00 : Control • 2'b01 : Isochronous • 2'b10 : Bulk • 2'b11 : Interrupt	2'b0
LSpdDev	[17]	R_W	Low-Speed Device This field is set by the application to indicate that this channel is communicating to a low-speed device.	1'b0
	[16]		Reserved	1'b0
EPDir	[15]	R_W	Endpoint Direction Endpoint Type Indicates the transfer type selected. • 1'b0 : OUT • 1'b1 : IN	1'b0
EPNum	[14:11]	R_W	Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.	4'h0
MPS	[10:0]	R_W	Maximum Packet Size Indicates the maximum packet size of the associated endpoint.	11'h0

Host Channel-n Split Register (HCSPLTn)

Channel_number : 0≤n≤15

Register	Address	R/W	Description	Reset Value
HCSPLTn	0x7C00_0504 +n*20h	R/W	Host Channel-n Split Register	0x00000000

HCSPLTn	Bit	R/W	Description	Initial State
SpltEna	[31]	R_W	Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.	1'b0
Reserved	[30:17]		Reserved	14'h0
CompSplt	[16]	R_W	Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.	1'b0
XactPos	[15:14]	R_W	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.	2'h0
HubAddr	[13:7]	R_W	Hub Address This field holds the device address of the transaction translator's hub.	7'h0
PrtAddr	[6:0]	R_W	Port Address This field is the port number of the recipient transaction translator.	7'h0

Host Channel-n interrupt Register (HCINTn)

Channel_number : 0≤n≤15

This register indicates the status of a channel with respect to USB- and AHB-related events. The application must read this register when the Host Channels Interrupt bit of the Core Interrupt register is set. Before the application can read 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.

Register	Address	R/W	Description	Reset Value
HCINTn	0x7C00_0508 +n*20h	R/W	Host Channel-n interrupt Register	0x00000000

HCINTn	Bit	R/W	Description	Initial State
Reserved	[31:11]		Reserved	21'h0
DataTglErr	[10]	R_SS_WC	Data Toggle Error	1'b0
FrmOvrun	[9]	R_SS_WC	Frame Overrun	1'b0
BblErr	[8]	R_SS_WC	Babble Error	1'b0
XactErr	[7]	R_SS_WC	Transaction Error	1'b0
NYET	[6]	R_SS_WC	NYET Response Received Interrupt	1'b0
ACK	[5]	R_SS_WC	ACK Response Received Interrupt	1'b0
NAK	[4]	R_SS_WC	NAK Response Received Interrupt	1'b0
STALL	[3]	R_SS_WC	STALL Response Received Interrupt	1'b0
AHBErr	[2]	R_SS_WC	AHB Error This is generated only in Internal DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address.	1'b0
ChHltd	[1]	R_SS_WC	Channel Halted Indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application.	1'b0
XferCompl	[0]	R_SS_WC	Transfer Completed Transfer completed normally without any errors.	1'b0

Host Channel-n interrupt Mask Register (HCINTMSKn)

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

Register	Address	R/W	Description	Reset Value
HCINTMSKn	0x7C00_050C+n*20h	R/W	Host Channel-n interrupt Mask Register	0x00000000

HCINTMSKn	Bit	R/W	Description	Initial State
Reserved	[31:11]		Reserved	21'h0
DataTglErrMsk	[10]	R_W	Data Toggle Error Mask	1'b0
FrmOvrunMsk	[9]	R_W	Frame Overrun Mask	1'b0
BblErrMsk	[8]	R_W	Babble Error Mask	1'b0
XactErrMsk	[7]	R_W	Transaction Error Mask	1'b0
NyetMsk	[6]	R_W	NYET Response Received Interrupt Mask	1'b0
AckMsk	[5]	R_W	ACK Response Received Interrupt Mask	1'b0
NakMsk	[4]	R_W	NAK Response Received Interrupt Mask	1'b0
StallMsk	[3]	R_W	STALL Response Received Interrupt Mask	1'b0
AHBErrMsk	[2]	R_W	AHB Error Mask	1'b0
ChHltdMsk	[1]	R_W	Channel Halted Mask	1'b0
XferComplMsk	[0]	R_W	Transfer Completed Mask	1'b0

Host Channel-n Transfer Size Register (HCTSIZn)

Channel_number : 0≤n≤15

Register	Address	R/W	Description	Reset Value
HCTSIZn	0x7C00_0510 +n*20h	R/W	Host Channel-n Transfer Size Register	0x00000000

HCTSIZn	Bit	R/W	Description	Initial State
DoPng	[31]	R_W	Do Ping Setting this field to 1 directs the host to do PING protocol.	1'h0
Pid	[30:29]	R_W	PID The application programs this field with the type of PID to use for the initial transaction. The host will maintain this field for the rest of the transfer. • 2'b00: DATA0 • 2'b01: DATA1 • 2'b10: DATA2 • 2'b11: MDATA(non-control)/SETUP(control)	2'b0
PktCnt	[28:19]	R_W	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.	10'b0
XferSize	[18:0]	R_W	Transfer Size For an OUT, this field is the number of data bytes the host will send 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.	19'b0

NOTE: Transfer Size for a Host Channel must equal [Packet Count * Max Packet Size] for accurate data transfer.

Host Channel-n DMA Address Register (HCDMAn)

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.

Register	Address	R/W	Description	Reset Value
HCDMAn	0x7C00_0514 +n*20h	R/W	Host Channel-n DMA Address Register	0x00000000

HCDMAn	Bit	R/W	Description	Initial State
DMAAddr	[31:0]	R_W	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.	32'h0

26.9.3 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

26.9.3.1 DEVICE GLOBAL REGISTERS

Device Configuration Register (DCFG)

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.

Register	Address	R/W	Description	Reset Value
DCFG	0x7C00_0800	R/W	Device Configuration Register	0x00200000

DCFG	Bit	R/W	Description	Initial State
Reserved	[31:23]		Reserved	9'h0
EPMisCnt	[22:18]	R_W	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 when the counter expires. The width of this counter depends on the depth of the Token Queue.	5'h8

DCFG	Bit	R/W	Description	Initial State
	[17:13]		Reserved	5'h0
PerFrInt	[12:11]	R_W	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%	2'h0
DevAddr	[10:4]	R_W	Device Address The application must program this field after every SetAddress control command.	7'h0
Reserved	[3]		Reserved	1'b0
NZSts OUTHShk	[2]	R_W	Non-Zero-Length Status OUT Handshake The application can use 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: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. • 1'b1: Send 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.	1'b0
DevSpd	[1:0]	R_W	Device Speed. Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However the actual bus speed is determined only after the chirp sequence is completed, 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).	2'b0

Device Control Register (DCTL)

Register	Address	R/W	Description	Reset Value
DCTL	0x7C00_0804	R/W	Device Control Register	0x00000000

DCTL	Bit	R/W	Description	Initial State
Reserved	[31:12]		Reserved	20'h0
PWROnPrgDone	[11]	R_W	Power-On Programming Done The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.	1'b0
CGOUTNak	[10]	WO	Clear Global OUT NAK A write to this field clears the Global OUT NAK.	1'b0
SGOUTNak	[9]	WO	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 the his bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register is cleared.	1'b0
CGNPInNAK	[8]	WO	Clear Global Non-Periodic IN NAK A write to this field clears the Global Non-Periodic IN NAK.	1'b0
SGNPInNAK	[7]	WO	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 can also set this bit when 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.	1'b0
TstCtl	[6:4]	R_W	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	3'b0
GOUTNakSts	[3]	RO	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.	1'b0

DCTL	Bit	R/W	Description	Initial State
GNPINNakSts	[2]	RO	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.	1'b0
SftDiscon	[1]	R_W	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 will 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. When 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. When 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.	1'b0
RmtWkUpSig	[0]	R_W	Remote Wakeup Signaling When 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.	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

Device Status Register (DSTS)

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.

Register	Address	R/W	Description	Reset Value
DSTS	0x7C00_0808	R	Device Status Register	0x00000002

DSTS	Bit	R/W	Description	Initial State
Reserved	[31:22]		Reserved	10'h0
SOFFN	[21:8]	RO	Frame or Microframe Number of the Received SOF When the core is operating at high speed; this field contains a microframe number. When the core is operating at full or low speed, this field contains a frame number.	14'h0
Reserved	[7:4]		Reserved	4'h0
ErrticErr	[3]	RO	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 can only perform a soft disconnect recover.	1'b0
EnumSpd	[2:1]	RO	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.	2'b01
SuspSts	[0]	RO	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 when there is no activity on the line_state signal for an extended period of time. The core comes out of the suspend: • When there is any activity on the line_state signal • When the application writes to the Remote Wakeup Signaling bit in the Device Control register.	1'b0

Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)

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 can be masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt : 1'b0
- Unmask interrupt : 1'b1

Register	Address	R/W	Description	Reset Value
DIEPMSK	0x7C00_0810	R/W	Device IN Endpoint Common Interrupt Mask	0x00000000

DIEPMSK	Bit	R/W	Description	Initial State
Reserved	[31:7]		Reserved	25'h0
INEPNakEffMsk	[6]	R_W	IN Endpoint NAK Effective Mask	1'b0
INTknEPMisMsk	[5]	R_W	IN Token received with EP Mismatch Mask	1'b0
INTknTXFEmpMsk	[4]	R_W	IN Token received with TxFIFO Empty mask	1'b0
TimeOUTMsk	[3]	R_W	Timeout Condition Mask	1'b0
AHBErrMsk	[2]	R_W	AHB Error Mask	1'b0
EPDisbldMsk	[1]	R_W	Endpoint Disabled Interrupt Mask	1'b0
XferComplMsk	[0]	R_W	Transfer Completed Interrupt Mask	1'b0

Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)

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 can be masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt : 1'b0
- Unmask interrupt : 1'b1

Register	Address	R/W	Description	Reset Value
DOEPMSK	0x7C00_0814	R/W	Device OUT Endpoint Common Interrupt Mask	0x00000000

DOEPMSK	Bit	R/W	Description	Initial State
Reserved	[31:7]		Reserved	27'h0
Back2BackSETup	[6]	R_W	Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.	1'b0
Reserved	[5]		Reserved	1'b0
OUTTknEPdisMsk	[4]	R_W	OUT Token Received When Endpoint Disabled Applies to control OUT endpoints only.	1'b0
SetUPMsk	[3]	R_W	SETUP Phase Done Mask Applies to control endpoints only.	1'b0
AHBErrMsk	[2]	R_W	AHB Error	1'b0
EPDisbldMsk	[1]	R_W	Endpoint Disabled Interrupt Mask	1'b0
XferComplMsk	[0]	R_W	Transfer Completed Interrupt Mask	1'b0

Device ALL Endpoints Interrupt Register (DAINT)

When 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 when the application sets and clears bits in the corresponding Device Endpoint – n Interrupt register.

Register	Address	R/W	Description	Reset Value
DAINT	0x7C00_0818	R	Device ALL Endpoints Interrupt Register	0x00000000

DAINT	Bit	R/W	Description	Initial State
OutEPInt	[31:16]	RO	OUT Endpoint Interrupt Bits One bit per OUT endpoint : Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	16'h0
InEpInt	[15:0]	RO	IN Endpoint Interrupt Bits One bit per IN endpoint : Bit 0 for IN endpoint 0, bit 15 for endpoint 15	16'h0

Device ALL Endpoints Interrupt Mask Register (DAINTMSK)

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device all Endpoints Interrupt register bit corresponding to that interrupt will still be set.

- Mask interrupt : 1'b0
- Unmask interrupt : 1'b1

Register	Address	R/W	Description	Reset Value
DAINTMSK	0x7C00_081C	R/W	Device ALL Endpoints Interrupt Mask Register	0x00000000

DAINTMSK	Bit	R/W	Description	Initial State
OutEPMsk	[31:16]	R_W	OUT EP Interrupt Mask Bits One bit per OUT endpoint : Bit 16 for OUT EP 0, bit 31 for OUT EP 15	16'h0
InEpMsk	[15:0]	R_W	IN EP Interrupt Mask Bits One bit per IN endpoint : Bit 0 for IN EP 0, bit 15 for IN EP 15	16'h0

Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)

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. When the Queue is full, the new token is pushed into the queue and oldest token is discarded.

Register	Address	R/W	Description	Reset Value
DTKNQR1	0x7C00_0820	R	Device IN Token Sequence Learning Queue Read Register 1	0x00000000

DTKNQR1	Bit	R/W	Description	Initial State
EPTkn	[31:8]	RO	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	24'h0
WrapBit	[7]	RO	Wrap Bit This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.	1'b0
	[6:5]	RO	Reserved	2'h0
INTKnWPtr	[4:0]	RO	IN Token QUEUE Write Pointer	5'h0

Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)

Read from this register returns the next 8 endpoint entries of the learning queue.

Register	Address	R/W	Description	Reset Value
DTKNQR2	0x7C00_0824	R	Device IN Token Sequence Learning Queue Read Register 2	0x00000000

DTKNQR2	Bit	R/W	Description	Initial State
EPTkn	[31:0]	RO	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	32'h0

Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)

Read from this register returns the next 8 endpoint entries of the learning queue.

Register	Address	R/W	Description	Reset Value
DTKNQR3	0x7C00_0830	R	Device IN Token Sequence Learning Queue Read Register 3	0x00000000

DTKNQR3	Bit	R/W	Description	Initial State
EPTkn	[31:0]	RO	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	32'h0

Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)

Read from this register returns the next 8 endpoint entries of the learning queue.

Register	Address	R/W	Description	Reset Value
DTKNQR4	0x7C00_0834	R	Device IN Token Sequence Learning Queue Read Register 4	0x00000000

DTKNQR4	Bit	R/W	Description	Initial State
EPTkn	[31:0]	RO	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	32'h0

Device VBUS Discharge Time Register (DVBUSDIS)

This register specifies the Vbus discharge time after Vbus pulsing during SRP.

Register	Address	R/W	Description	Reset Value
DVBUSDIS	0x7C00_0828	R/W	Device VBUS Discharge Time Register	0x000017D7

DVBUSDIS	Bit	R/W	Description	Initial State
Reserved	[31:16]		Reserved	16'h0
DVBUSDis	[15:0]	R_W	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	16'h17D7

Device VBUS Pulsing Time Register (DVBUSPULSE)

This register specifies the Vbus discharge time during SRP.

Register	Address	R/W	Description	Reset Value
DVBUSPULSE	0x7C00_082C	R/W	Device VBUS Discharge Time Register	0x000005B8

DVBUSPULSE	Bit	R/W	Description	Initial State
Reserved	[31:12]		Reserved	16'h0
DVBUSPulse	[11:0]	R_W	Device Vbus Pulsing Time Specifies the Vbus pulsing time during SRP. This value equals : Vbus pulse time in PHY clocks /1,024	12'h5B8

26.9.3.2 Device Logical Endpoint-Specific Registers

A logical endpoint is unidirectional: it can be 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.

Device Control IN Endpoint 0 Control Register (DIEPCTL0)

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

Register	Address	R/W	Description	Reset Value
DIEPCTL0	0x7C00_0900	R/W	Device Control IN Endpoint 0 Control Register	0x00008000

DIEPCTL0	Bit	R/W	Description	Initial State
EPEna	[31]	R_WS_SC	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	1'b0
EPDis	[30]	R_WS_SC	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.	1'b0
Reserved	[29:28]		Reserved	2'b0
SetNAK	[27]	WO	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.	1'b0
CNAK	[26]	WO	Clear NAK A write to this bit clears the NAK bit for the endpoint.	1'b0
TxFNum	[25:22]	RO	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.	4'h0
Stall	[21]	R_WS_SC	STALL Handshake The application can only set this bit, and the core clears it, when 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.	1'b0
Reserved	[20]		Reserved	1'b0

DIEPCTL0	Bit	R/W	Description	Initial State
EPType	[19:18]	RO	Endpoint Type Hardcoded to 00 for control	2'h0
NAKsts	[17]	RO	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 When 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.	1'b0
Reserved	[16]		Reserved	1'b0
USBActEP	[15]	RO	USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	1'b1
NextEp	[14:11]	R_W	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 can access this field, even when the Endpoint Enable bit is not set. This field is not valid in Slave mode operation.	4'b0
Reserved	[10:2]		Reserved	9'h0
MPS	[1:0]	R_W	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	2'h0

Device Control OUT Endpoint 0 Control Register (DOEPCTL0)

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

Register	Address	R/W	Description	Reset Value
DOEPCTL0	0x7C00_0B00	R/W	Device Control OUT Endpoint 0 Control Register	0x00008000

DOEPCTL0	Bit	R/W	Description	Initial State
EPEna	[31]	R_WS_SC	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.	1'b0
EPDis	[30]	RO	Endpoint Disable The application cannot disable control OUT endpoint 0.	1'b0
Reserved	[29:28]		Reserved	2'b0
SetNAK	[27]	WO	Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	1'b0
CNAK	[26]	WO	Clear NAK A write to this bit clears the NAK bit for the endpoint.	1'b0
Reserved	[25:22]		Reserved	4'h0
Stall	[21]	R_WS_SC	STALL Handshake The application can only set this bit, and the core clears it, when 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.	1'b0
Snp	[20]	R_W	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.	1'b0
EPType	[19:18]	RO	Endpoint Type Hardcoded to 2'b00 for control.	2'h0

DOEPCTL0	Bit	R/W	Description	Initial State
NAKsts	[17]	RO	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 When either the 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.	1'b0
Reserved	[16]		Reserved	1'b0
USBActEP	[15]	RO	USB Active Endpoint This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.	1'b1
Reserved	[14:2]		Reserved	13'h0
MPS	[1:0]	RO	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	2'h0

Device Endpoint-N Control Register (DIEPCTLN/DOEPCTLN)

Endpoint_number :1≤n≤15

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

Register	Address	R/W	Description	Reset Value
DIEPCTLn / DOEPCTLn	0x7C00_0900+ n*20h /0x7C00_0B00+ n*20h	R/W	Device Endpoint-n Control Register	0x00000000

DIEPCTLn/ DOEPCTLn	Bit	R/W	Description	Initial State
EPEna	[31]	R_W S_SC	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.	1'b0
EPDis	[30]	R_W S_SC	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.	1'b0
SetD1PID	[29]	WO	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.	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]	WO	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.	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.	

DIEPCTLn/ DOEPCTLn	Bit	R/W	Description	Initial State
SNAK	[27]	WO	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 can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP packet is received on that endpoint.	1'b0
CNAK	[26]	WO	Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.	1'b0
TxFNum	[25:22]	R_W	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.	4'h0
Stall	[21]	R_W	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 can clear this bit, never the core.	1'b0
		R_WS _SC	Applies to control endpoints only The application can only set this bit, and the core clears it, when 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.	
Snp	[20]	R_W	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.	1'b0
EPType	[19:18]	RO	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	2'h0

DIEPCTLn/ DOEPCTLn	Bit	R/W	Description	Initial State
NAKsts	[17]	RO	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 When either the application or the core sets this bit: • 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.	1'b0
DPID	[16]	RO	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	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]	R_W_SC	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.	1'b0

DIEPCTLn/ DOEPCTLn	Bit	R/W	Description	Initial State
NextEp	[14:11]	R_W	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 can access this field, even when the Endpoint Enable bit is low. This field is not valid in Slave mode operation.	4'h0
MPS	[10:0]	R_W	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.	11'h0

Device Endpoint-N Interrupt Register (DIEPINTN/DOEPINTN)

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 when the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register is set. Before the application can read 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.

Register	Address	R/W	Description	Reset Value
DIEPINTn/ DOEPINTn	0x7C00_0908 +n*20h / 0x7C00_0B08 +n*20h	R/W	Device Endpoint-n Interrupt Register	0x00000080

DIEPINTn/ DOEPINTn	Bit	R/W	Description	Initial State
EPEna	[31:7]		Reserved	25'h1
INEPNakEff	[6]	RO	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 can be cleared when 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.	1'b0
Back2BackSETup		R_W	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.	

DIEPINTn/ DOEPINTn	Bit	R/W	Description	Initial State
INTknEPMis	[5]	R_SS_WC	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.	1'b0
			For OUT endpoints, this bit is reserved.	
INTknTXFEmp	[4]	R_SS_WC	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.	1'b0
OUTTknEPdis			OUT Token Received When Endpoint Disabled Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.	
TimeOUT	[3]	R_SS_WC	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.	1'b0
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 can decode the received SETUP data packet.	
AHBErr	[2]	R_SS_WC	AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.	1'b0
EPDisbld	[1]	R_SS_WC	Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.	1'b0
XferCompl	[0]	R_SS_WC	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.	1'b0

Device Endpoint 0 Transfer Size Register (DIEPTSIZ0)

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 can only read this register once the core has cleared the Endpoint Enable bit.

Register	Address	R/W	Description	Reset Value
DIEPTSIZ0	0x7C00_0910	R/W	Device IN Endpoint 0 Transfer Size Register	0x00000000

DIEPTSIZ0	Bit	R/W	Description	Initial State
Reserved	[31:21]		Reserved	11'h0
PktCnt	[20:19]	R_W	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.	2'b0
Reserved	[18:7]		Reserved	12'h0
XferSize	[6:0]	R_W	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.	7'h0

Device OUT Endpoint 0 Transfer Size Register (DOEPTSIZ0)

Register	Address	R/W	Description	Reset Value
DOEPTSIZ0	0x7C00_0B10	R/W	Device OUT Endpoint 0 Transfer Size Register	0x00000000

DOEPTSIZ0	Bit	R/W	Description	Initial State
Reserved	[31]		Reserved	1'b0
SUPCnt	[30:29]	R_W	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	2'h0
Reserved	[28:20]		Reserved	9'h0
PktCnt	[19]	R_W	Packet Count This field is decremented to zero after a packet is written into the RxFIFO.	1'b0
Reserved	[18:7]		Reserved	12'h0
XferSize	[6:0]	R_W	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.	7'h0

Device Endpoint-N Transfer Size Register (DIEPTSIZN/DOEPTSIZN)

Endpoint_number: 1≤n≤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 can only read this register once the core has cleared the Endpoint Enable bit. This register is used only for endpoints other than Endpoint 0.

Register	Address	R/W	Description	Reset Value
DIEPTSIZn/ DOEPTSIZn	0x7C00_0910 +n*20h / 0x7C00_0B10 +n*20h	R/W	Device Endpoint-n Transfer Size Register	0x00000000

DIEPTSIZn/ DOEPTSIZn	Bit	R/W	Description	Initial State
Reserved	[31]		Reserved	1'b0
MC	[30:29]	R_W	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	2'b0
		RO	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.	
RxDPID		RO	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	
SUPCnt		R_W	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	

DIEPTSIZn/ DOEPTSIZn	Bit	R/W	Description	Initial State
PktCnt	[28:19]	R_W	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	10'h0
XferSize	[18:0]	R_W	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 can be 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.	19'h0

NOTE: Transfer Size for a Device Endpoint must equal [Packet Count * Max Packet Size] for accurate data transfer.

Device Endpoint-N DMA Address (DIEPDMAN/DOEPDMAN)

Endpoint_number : 0≤n≤15

The starting DMA address must be DWORD-aligned.

Register	Address	R/W	Description	Reset Value
DIEPDMAn/ DOEPDMAn	0x7C00_0914 +n*20h / 0x7C00_0B14 +n*20h	R/W	Device Endpoint-n DMA Address	0x00000000

DIEPDMAn/ DOEPDMAn	Bit	R/W	Description	Initial State
DMAAddr	[31:0]	R_W	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.	32'h0

26.9.3 POWER AND CLOCK GATING REGISTER

Power and Clock Gating control register (PCGCCTL)

The application can use this register to control OTG's clock gating.

Register	Address	R/W	Description	Reset Value
PCGCCTL	0x7C00_0E00	R/W	Power and Clock Gating Control Register	0x00000000

DIEPTSIZ0	Bit	R/W	Description	Initial State
Reserved	[31:1]		Reserved	31'h0
StopPclk	[0]	R_W	STOP Pclk The application sets this bit to stop the PHY clock when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts.	1'b0

27 SD/MMC HOST CONTROLLER

This chapter describes the SD/SDIO/MMC/CE-ATA host controller and related registers supported by S3C6410X RISC microprocessor.

27.1 OVERVIEW

The SD/MMC host controller is a combo host for Secure Digital card and MultiMediaCard. This host is compatible for 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, you would get 50Mhz clock rate and access 8-bit data pin simultaneously.

27.2 FEATURES

The High-Speed MMC controller supports:

- SD Standard Host Specification (ver 2.0) compatible
- SD Memory Card Specificatio (ver 2.0) / High Speed MMC Spec(4.2) compatible
- SDIO Card Specification (Ver 1.0) compatible
- 512 bytes FIFO for data Tx/Rx
- CPU Interface and DMA data transfer mode
- 1-bit / 4-bit / 8-bit mode switch support
- Auto CMD12 support
- Suspend / Resume support
- Read Wait operation support
- Card Interrupt support
- CE-ATA mode support

27.3 BLOCK DIAGRAM

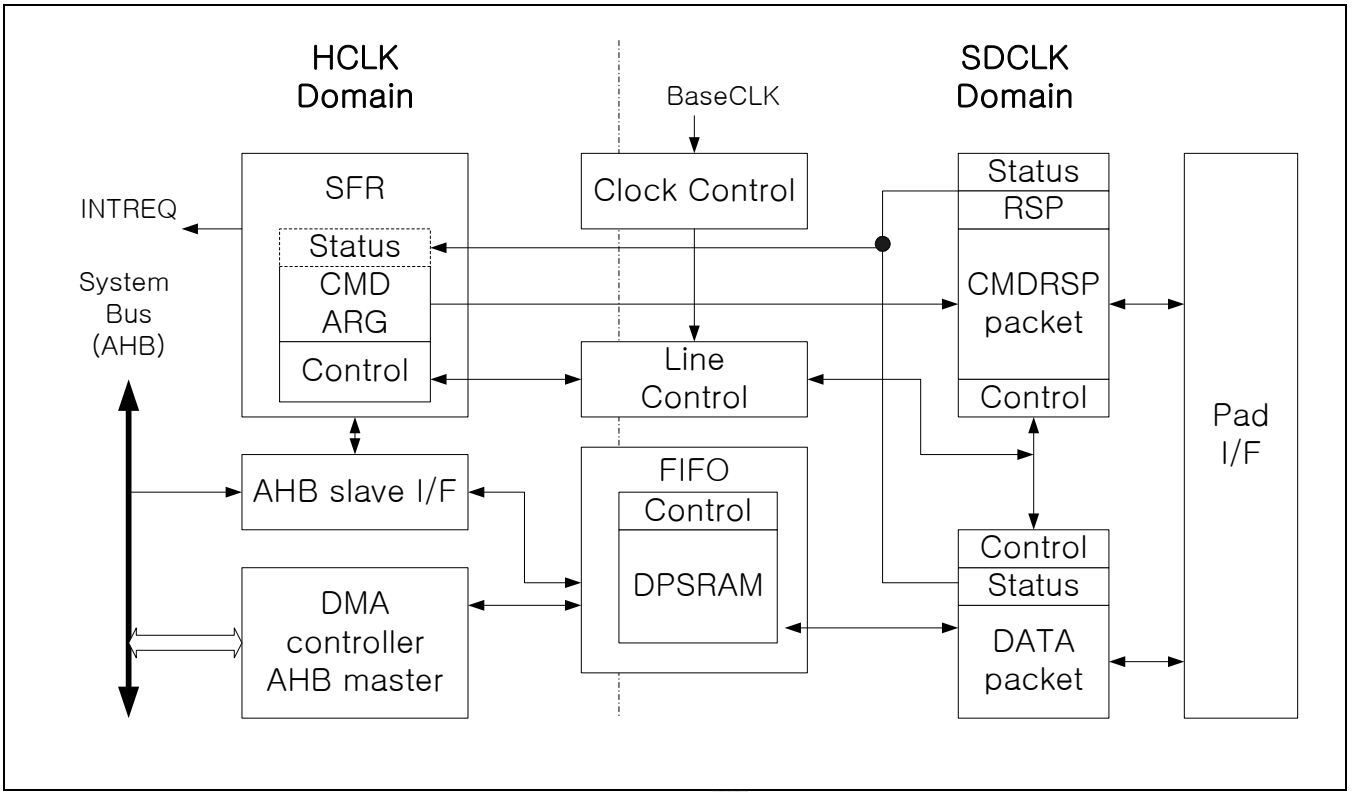


Figure 27-1. HSMMC block diagram

27.4 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.

27.4.1 SD CARD DETECTION SEQUENCE

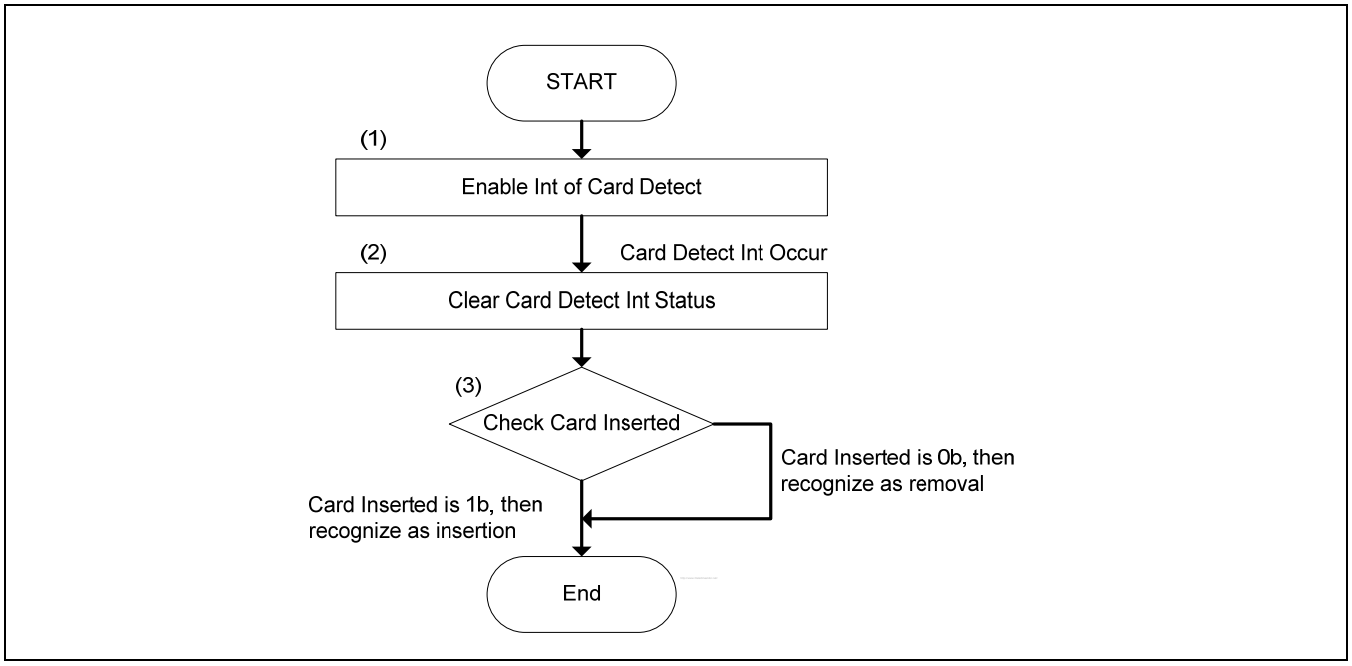


Figure 27-2. SD Card Detect Sequence

The flow chart for detecting a SD card is shown in Figure 27-2. Each step is executed as follows:

(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) When the 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. In this case where Card Inserted(INSCARD) is 1, the Host Driver can supply the power and the clock to the SD card. In this case where Card Inserted is 0, the other executing processes of the Host Driver shall be immediately closed.

27.4.2 SD CLOCK SUPPLY SEQUENCE

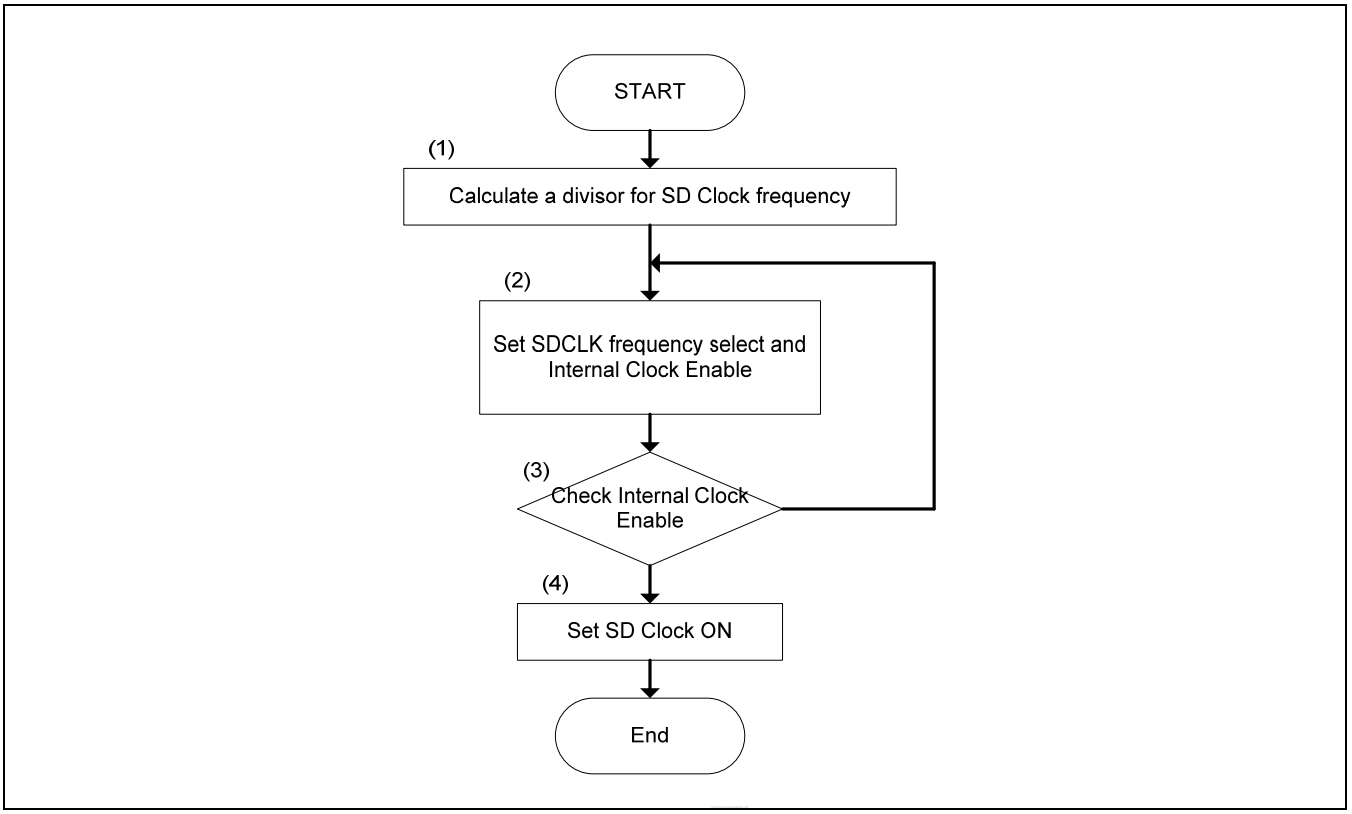


Figure 27-3. SD Clock Supply Sequence

The sequence for supplying SD Clock to a SD card is described in Figure 27-3. The clock shall be supplied to the card 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.

(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. Then, the Host Controller starts to supply the SD Clock.

27.4.3 SD CLOCK STOP SEQUENCE

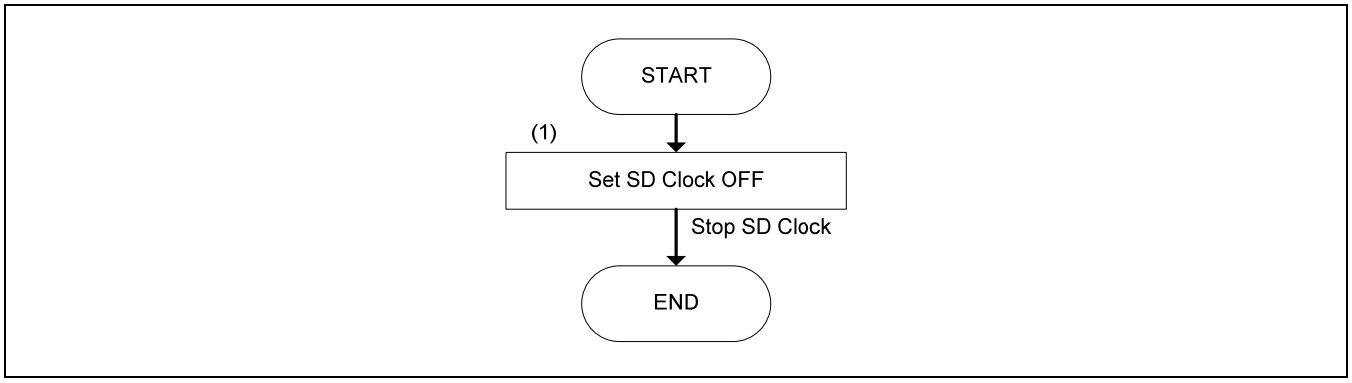


Figure 27-4. SD Clock Stop Sequence

The flow chart for stopping the SD Clock is shown in Figure 27-4. The Host Driver does not stop the SD Clock when 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. Then the Host Controller stops supplying the SD Clock.

27.4.4 SD CLOCK FREQUENCY CHANGE SEQUENCE

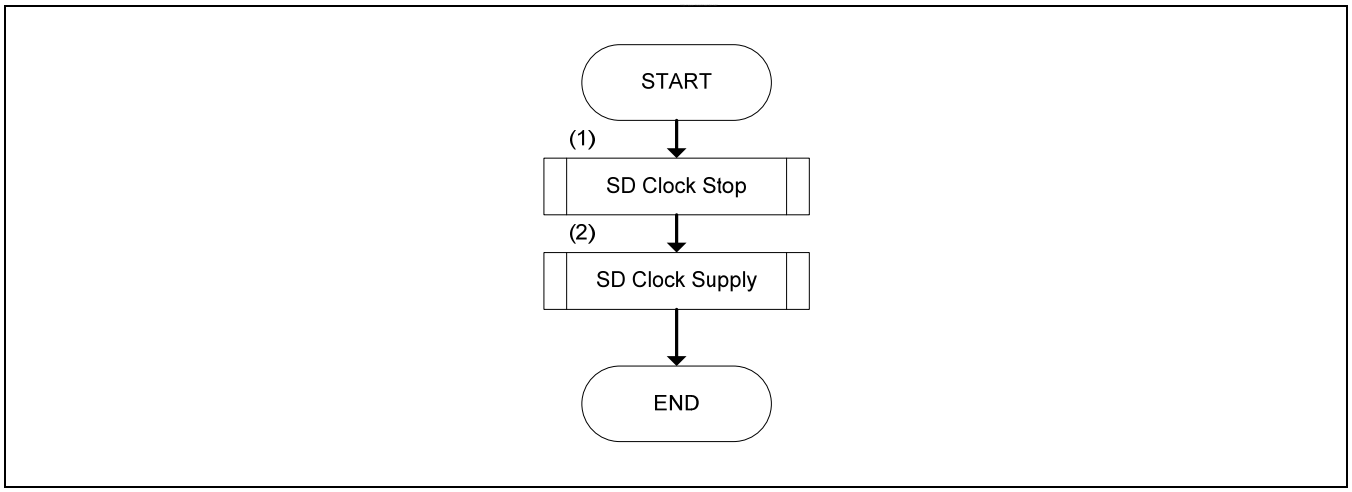


Figure 27-5. SD Clock Change Sequence

The sequence for changing SD Clock frequency is shown in Figure 27-5. When SD Clock is still off, step (1) is omitted.

(1) Perform SD Clock Stop Sequence. Refer to 27.4.2

(2) Perform SD Clock Supply Sequence. Refer to 27.4.3

27.4.5 SD BUS POWER CONTROL SEQUENCE

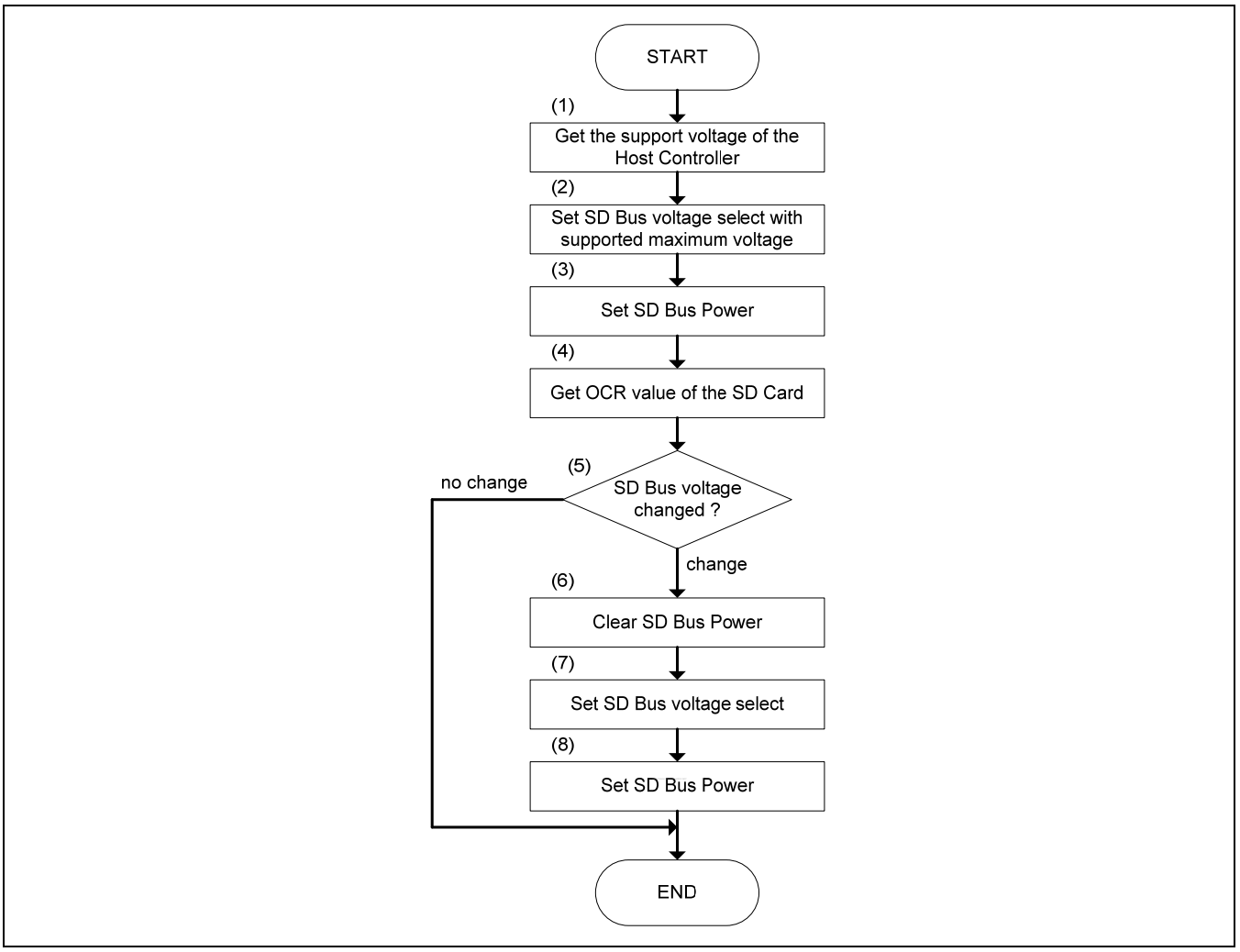


Figure 27-6. SD Bus Power Control Sequence

The sequence for controlling the SD Bus Power is described in Figure 27-6 and steps are described below:

(1) By reading the Capabilities register, 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, go to step (6). If SD Bus voltage does 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 shall clear 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.

27.4.6 CHANGE BUS WIDTH SEQUENCE

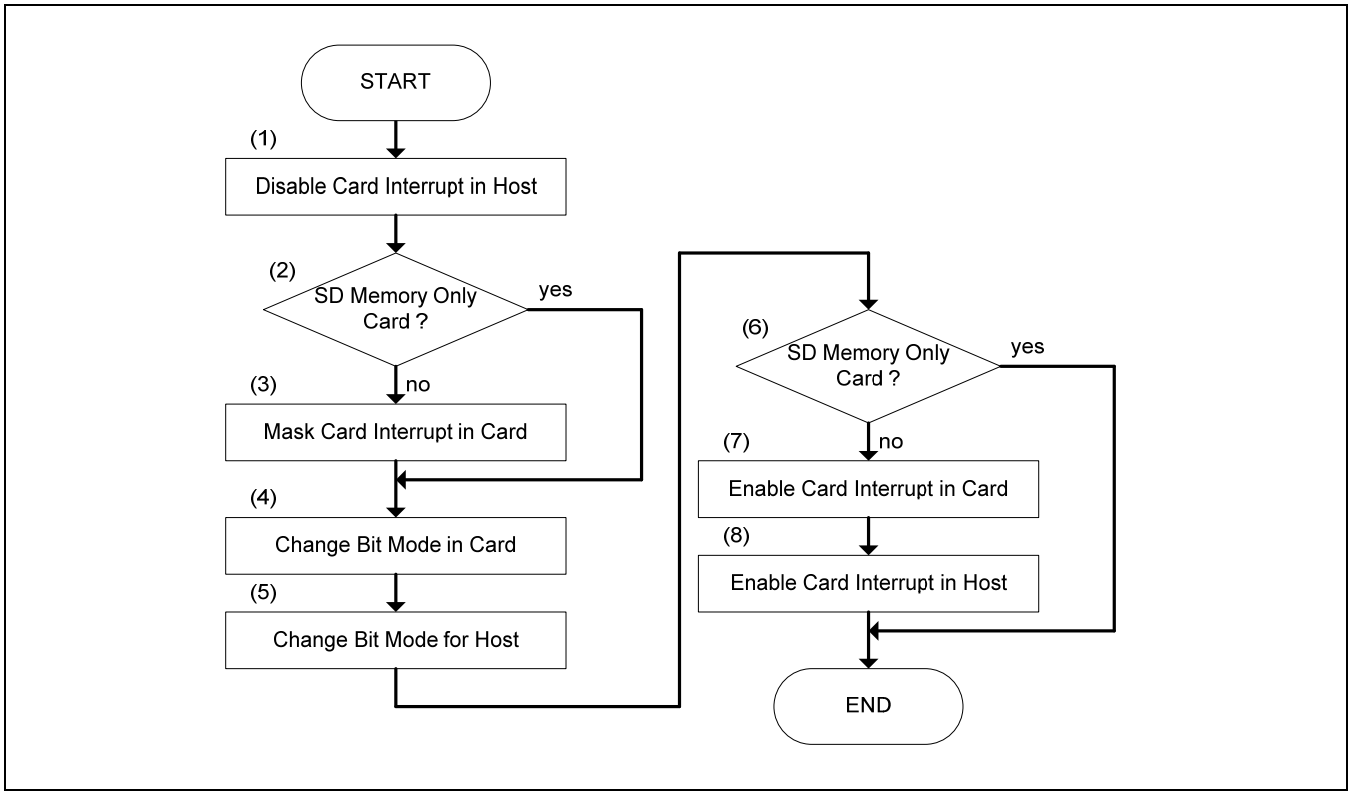


Figure 27-7. Change Bus Width Sequence

The sequence for changing bit mode on SD Bus is shown in Figure 27-7.

(1) Set Card Interrupt Status Enable(STACARDINT) in the Normal Interrupt Status Enable register to 0 for masking incorrect interrupts that may occur while changing the bus width.

(2) In case of SD memory card, 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. Change SD memory card bus width by ACMD6(Set bus width) and SDIO card bus width by setting Bus Width of Bus Interface Control register in CCCR.

(5) In case of you want to change to 4-bit mode, set Data Transfer Width(WIDE4) in the Host Control register to 1. In another case (1-bit mode), set this bit to 0.

(6) In case of SD memory card, 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 in the Normal Interrupt Status Enable register to 1.

27.4.7 TIMEOUT SETTING FOR DAT LINE

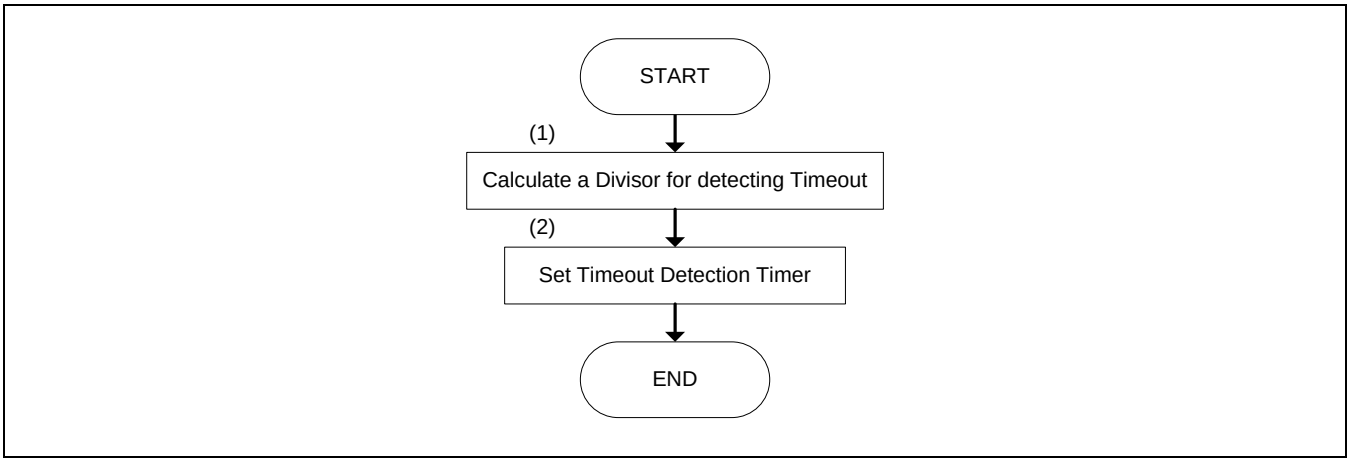


Figure 27-8. Timeout Setting Sequence

In order to detect timeout errors on DAT line, the Host Driver will execute 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 from step (1) above.

27.4.8 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 only 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

27.4.9 SD COMMAND ISSUE SEQUENCE

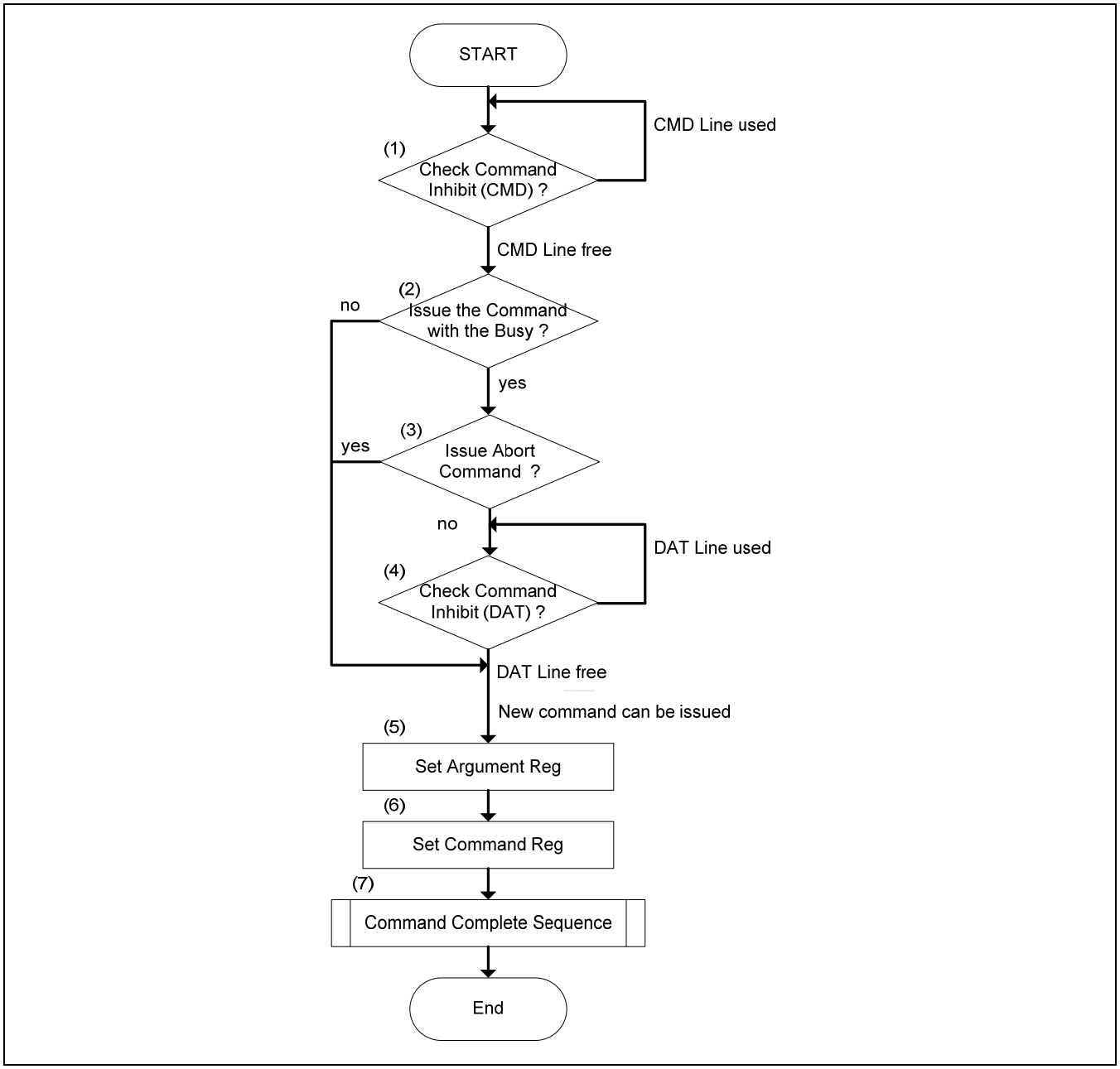


Figure 27-9. Timeout Setting Sequence

Take the following steps for Timeout Setting:

(1) Check Command Inhibit(CMD) in the Present State register. Repeat this step until Command Inhibit(CMD) is 0. That is, when Command Inhibit(CMD) is 1, the Host Driver will 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). In the case of no abort command, 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: Writing the upper byte in the Command register causes a SD command to be issued.

(7) Perform Command Complete Sequence

27.4.10 COMMAND COMPLETE SEQUENCE

The sequence for completing the SD Command is shown in Figure 27-10. There is a possibility that the errors (Command Index/End bit/CRC/Timeout Error) occur during this sequence.

(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 has occurred, 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 will only issue commands that do not use the busy signal.
2. The Host Driver judges the Auto CMD12(Stop Command) complete by monitoring Transfer Complete.
3. When 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 will appear in the response of Auto CMD12 or in the response of the next memory command.

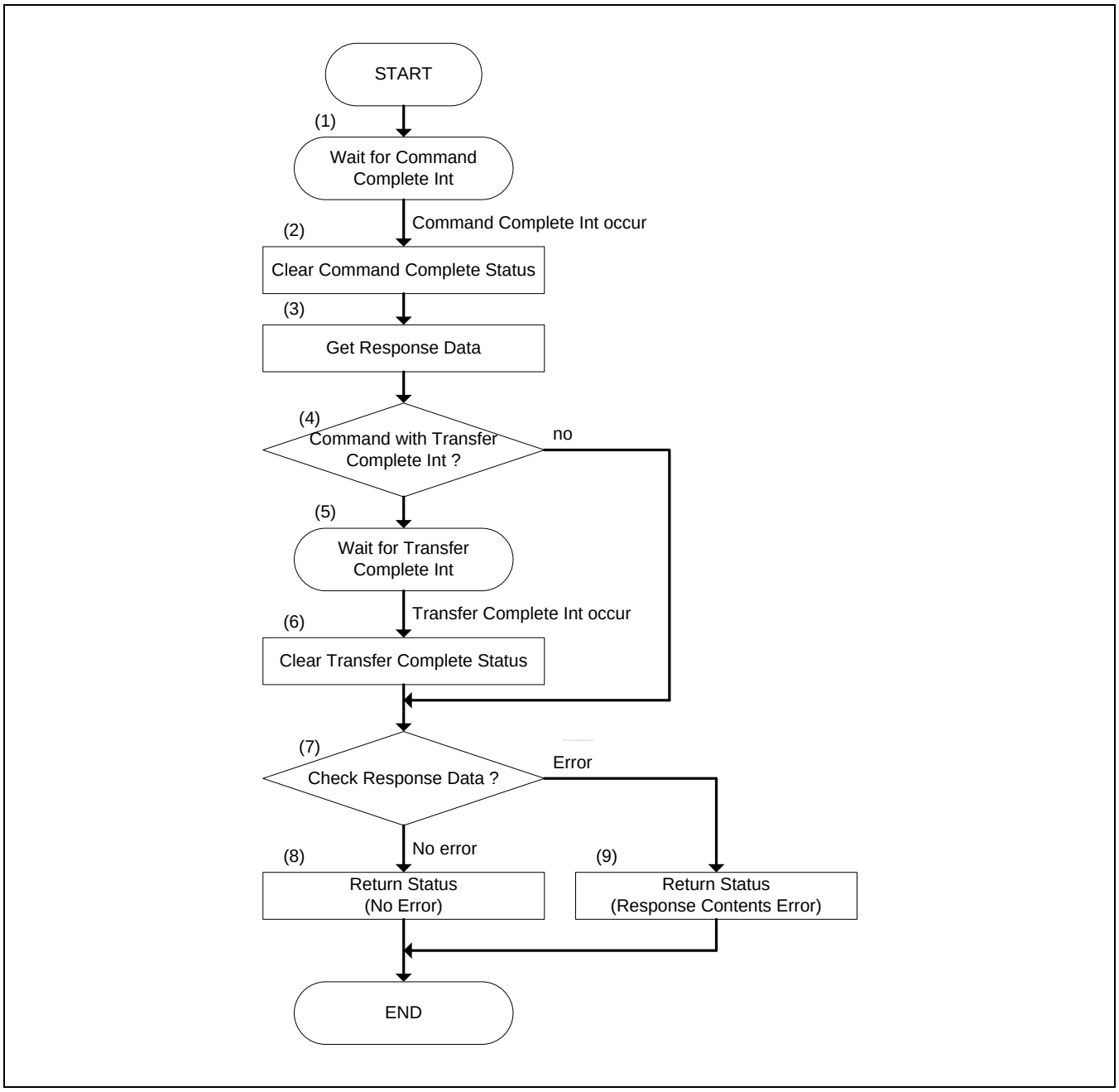


Figure 27-10. Command Complete Sequence

27.4.11 TRANSACTION CONTROL WITH DATA TRANSFER USING DAT LINE

Depending on whether DMA (optional) is used or not, there are two execution methods. The sequence not using DMA is shown in Figure 27-11 and the sequence using DMA is shown in Figure 27-12.

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 shall be 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.

27.4.12 NOT USING DMA

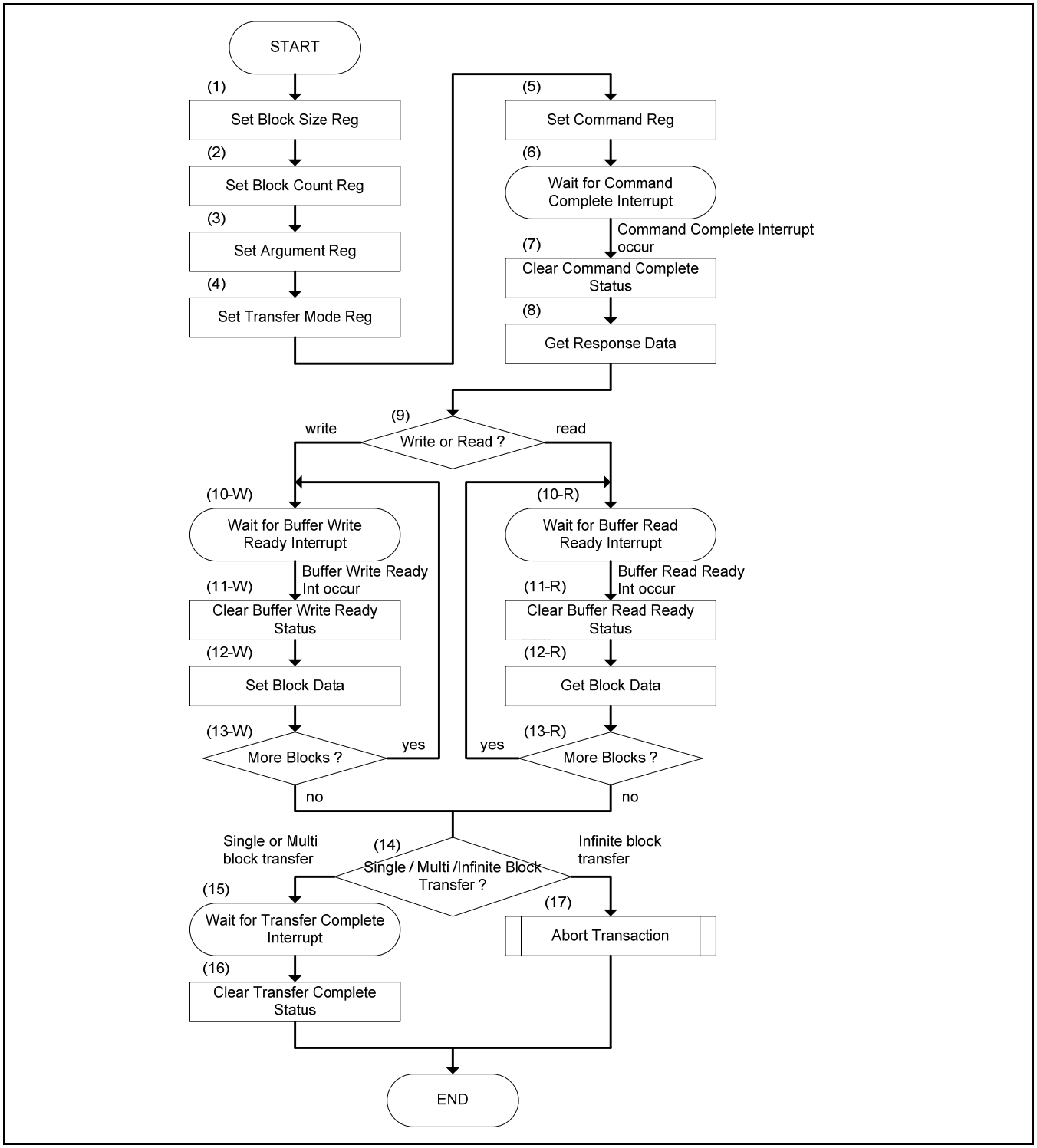


Figure 27-11. Transaction Control with Data Transfer Using DAT Line Sequence (Not using DMA)

(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. And 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: When writing the upper byte of Command register, SD command is issued.

(6) Wait for the Command Complete Interrupt.

(7) Write 1 to the Command Complete(STACMDCMPLT) in the Normal Interrupt Status register for clearing 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). In case of 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 for clearing 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 for clearing 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 for clearing 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

27.4.13 USING DMA

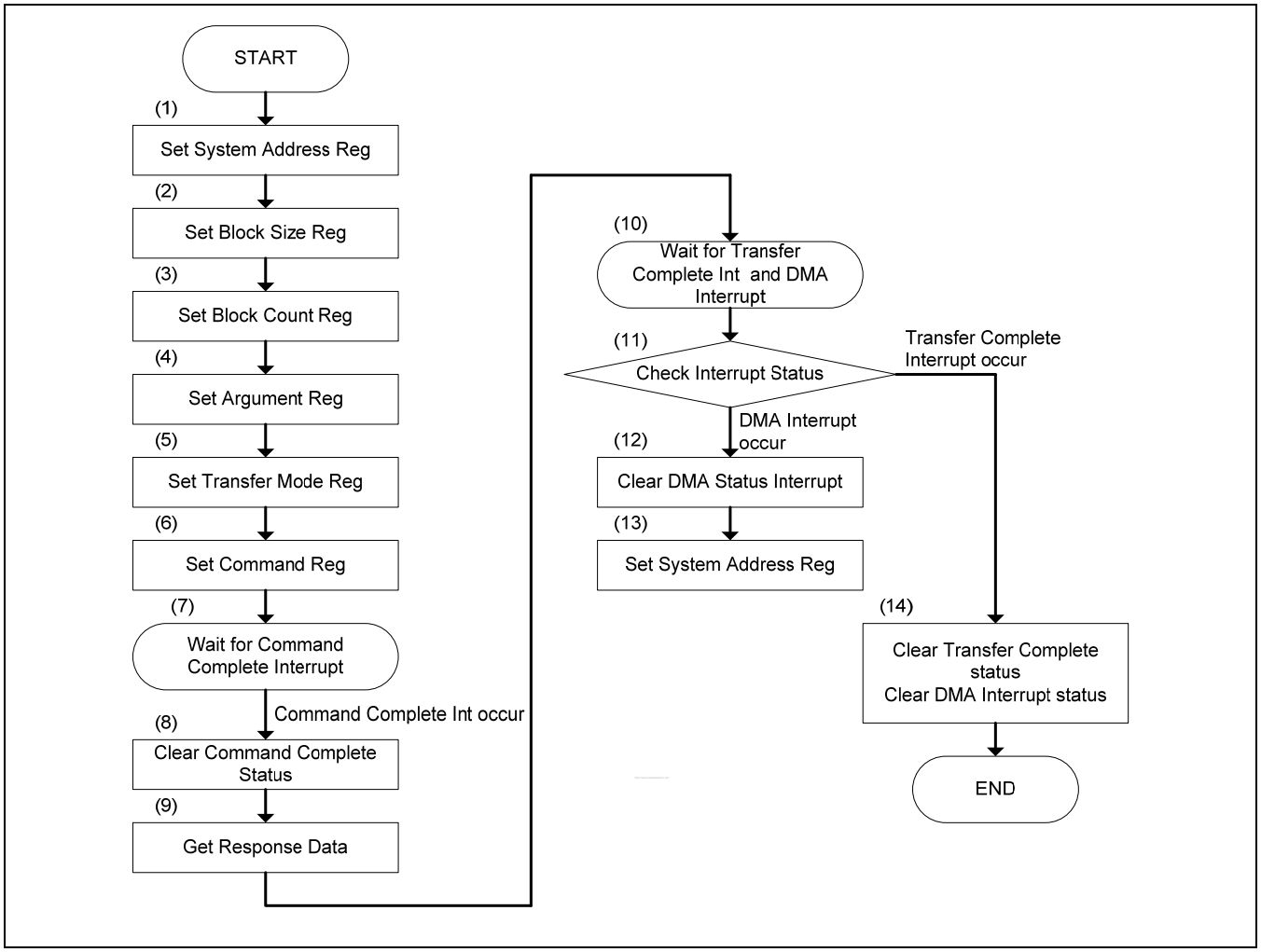


Figure 27-12. Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)

(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.

And 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: When writing to the upper byte of the Command register, the SD command is issued 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.

27.5 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 when the Host Driver stops Infinite Block Transfers. The second case is when 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 can issue 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 shall issue an Abort Command after the data transfer stopped by using **Stop At Block Gap Request** in the *Block Gap Control* register.

27.6 DMA TRANSACTION

DMA allows a peripheral to read and write memory without intervention from the CPU. Only one SD command transaction can be executed by DMA. Host Controllers that support DMA shall support 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 shall remain accessible for issuing non-DAT line commands during a DMA transfer. The result of a DMA transfer shall be the same regardless of the system bus transaction method used. DMA shall not support infinite transfers.

DMA transfers can be stopped and restarted using control bits in the Block Gap Control register. When the Stop At Block Gap Request is set, DMA transfers shall be suspended. When the Continue Request is set or a Resume Command is issued, DMA shall continue to execute transfers. Refer to the Block Gap Control register for details. If SD Bus errors occur, SD Bus transfers shall be stopped and DMA transfers shall be stopped. Setting the Software Reset For DAT Line in the Software Reset register shall abort DMA transfers.

27.7 SD/MMC HOST CONTROLLER SPECIAL REGISTERS

27.7.1 CONFIGURATION REGISTER TYPES

Configuration register fields are assigned to one of the attributes described below:

Register Attribute	Description
RO	Read-only register: Register bits are read-only and cannot be altered by software or any reset operation**.** Writes to these bits are ignored.
ROC	Read-only status: These bits are initialized to zero at reset. Writes to these bits are ignored.
RW or R/W	Read-write register: Register bits are read-write and may be either set or cleared by software to the desired state.
RW1C	Read-only status, Write-1-to-clear status: Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a 1. Writing a 0 to RW1C bits has no effect.
RWAC	Read-Write, automatic clear register: The Host Driver requests a Host Controller operation by setting the bit. The Host Controllers shall clear the bit automatically when the operation is complete. Writing a 0 to RWAC bits has no effect.
HWInit	Hardware Initialized: Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. Bits are read-only after initialization, and writes to these bits are ignored.
Rsvd or Reserved	Reserved. These bits are initialized to zero, and writes to them are ignored.

27.7.2 SDMA SYSTEM ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
SDMASYSAD0	0x7C200000	R/W	SDMA System Address register (Channel 0)	0x0
SDMASYSAD1	0x7C300000	R/W	SDMA System Address register (Channel 1)	0x0
SDMASYSAD2	0x7C400000	R/W	SDMA System Address register (Channel 2)	0x0

This register contains the physical system memory address used for DMA transfers.

Name	Bit	Description	Initial Value
SDMASYSAD	[30:0]	**SDMA System Address** This register contains the system memory address for a DMA transfer. When the Host Controller stops a DMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the next system address of the next contiguous data position can be 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. When the most upper byte of this register (003h) is written, the Host Controller restarts the DMA transfer. When restarting DMA by the Resume command or by setting **Continue Request** in the *Block Gap Control* register, the Host Controller shall start at the next contiguous address stored here in the *System Address* register.	0x00

27.7.3 BLOCK SIZE REGISTER

This register is used to configure the number of bytes in a data block.

Register	Address	R/W	Description	Reset Value
BLKSIZE0	0x7C200004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 0)	0x0
BLKSIZE1	0x7C300004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 1)	0x0
BLKSIZE2	0x7C400004	R/W	Host DMA Buffer Boundary and Transfer Block Size Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15]	Reserved	0
BUFBOUND	[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 shall be updated at every system memory boundary during SDMA transfer. These bits specify the size of contiguous buffer in the system memory. The SDMA transfer shall wait 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 may issue or may not issue **DMA Interrupt**. In particular, **DMA Interrupt** shall not be issued after **Transfer Complete Interrupt** is issued. In case of 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 when the Host Controller detects carry out of the address from bit 11 to 12. These bits shall be supported when the **SDMA Support** in the *Capabilities* register is set to 1 and this function is active when the **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

Name	Bit	Description	Initial Value
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 the maximum buffer size can be set. In case of memory, it shall be set up to 512 bytes. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations shall be ignored. 0200h = 512 Bytes 01FFh = 511 Bytes … … 0004h = 4 Bytes 0003h = 3 Bytes 0002h = 2 Bytes 0001h = 1 Byte 0000h = No data transfer	0

27.7.4 BLOCK COUNT REGISTER

This register is used to configure the number of data blocks.

Register	Address	R/W	Description	Reset Value
BLKCNT0	0x7C200006	R/W	Blocks Count For Current Transfer (Channel 0)	0x0
BLKCNT1	0x7C300006	R/W	Blocks Count For Current Transfer (Channel 1)	0x0
BLKCNT2	0x7C400006	R/W	Blocks Count For Current Transfer (Channel 2)	0x0

Name	Bit	Description	Initial Value
BLKCNT	[15:0]	**Blocks Count For Current Transfer** This register is enabled when **Block Count Enable** in the *Transfer Mode* register is set to 1 and is valid only for multiple block transfers. The Host Driver shall set 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 when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register must be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer context as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the Host Driver shall restore the previously saved block count. FFFFh = 65535 blocks … … 0002h = 2 blocks 0001h = 1 block 0000h = Stop Count	0

27.7.5 ARGUMENT REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
ARGUMENT0	0x7C200008	R/W	Command Argument Register (Channel 0)	0x0
ARGUMENT1	0x7C300008	R/W	Command Argument Register (Channel 1)	0x0
ARGUMENT2	0x7C400008	R/W	Command Argument Register (Channel 2)	0x0

Name	Bit	Description	Initial 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

27.7.6 TRANSFER MODE REGISTER

This register is used to control the operation of data transfers. The Host Driver shall set 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 shall save the value of this register when the 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 shall implement write protection for this register during data transactions. Writes to this register shall be ignored when the **Command Inhibit (DAT)** in the *Present State* register is 1.

Register	Address	R/W	Description	Reset Value
TRNMOD0	0x7C20000C	R/W	Transfer Mode Setting Register (Channel 0)	0x0
TRNMOD1	0x7C30000C	R/W	Transfer Mode Setting Register (Channel 1)	0x0
TRNMOD2	0x7C40000C	R/W	Transfer Mode Setting Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:10]	**Reserved**	0
CCSCON	[9:8]	**Command Completion Signal Control** ‘00’ = No CCS Operation (Normal operation, Not 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
	[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 shall be set to 0. If this bit is 0, it is not necessary to set the *Block Count* register. (Refer to the Table below "**Determination of Transfer Type"**) 1 = Multiple Block 0 = Single Block	0
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
-	[3]	**Reserved**	0

Name	Bit	Description	Initial Value
ENACMD12	[2]	**Auto CMD12 Enable** Multiple block transfers for memory require CMD12 to stop the transaction. When this bit is set to 1, the Host Controller shall issue CMD12 automatically when last block transfer is completed. The Host Driver shall 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. When 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 can be enabled only if it is supported as indicated in the **DMA Support** in the *Capabilities* register. If DMA is not supported, this bit is meaningless and shall always read 0. If this bit is set to 1, a DMA operation shall begin when 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.

Table 27-1. 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.

27.7.7 COMMAND REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
CMDREG0	0x7C20000E	R/W	Command Register (Channel 0)	0x0
CMDREG1	0x7C30000E	R/W	Command Register (Channel 1)	0x0
CMDREG2	0x7C40000E	R/W	Command Register (Channel 2)	0x0

The Host Driver shall check 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 for writing when **Command Inhibit (CMD)** is set.

Name	Bit	Description	Initial Value
	[15:14]	**Reserved**	
CMDIDX	[13:8]	**Command Index** These bits shall be 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 **shall** be set to 00b for all other commands. • Suspend Command If the Suspend command succeeds, the Host Controller shall assume the SD Bus has been released and that it is possible to issue the next command, which uses the ***DAT*** line. The Host Controller shall de-assert Read Wait for read transactions and stop checking busy for write transactions. The interrupt cycle shall start, in 4-bit mode. If the Suspend command fails, the Host Controller shall maintain its current state, and the Host Driver shall restart 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 shall check for busy before starting write transfers. • Abort Command If this command is set when executing a read transfer, the Host Controller shall stop reads to the buffer. If this command is set when executing a write transfer, the Host Controller shall stop driving the ***DAT*** line. After issuing the Abort command, the Host Driver must issue a softwares 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	

Name	Bit	Description	Initial Value
DATAPRNT	[5]	**Data Present Select** This bit is set to 1 to indicate that data is present and shall be 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 shall check 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	
ENCMDCRC	[3]	**Command CRC Check Enable** If this bit is set to 1, the Host Controller shall check 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	
	[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	

Table 27-2. 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 shall check busy after receiving response. For example, usually CMD52 is used as R5 but I/O abort command shall be used as R5b.
2. For CMD52 to read BS after writing "Bus Suspend," Command Type must be "Suspend" as well.

27.7.8 RESPONSE REGISTER

This register is used to store responses from SD cards.

Register	Address	R/W	Description	Reset Value
RSPREG0_0	0x7C200010	ROC	Response Register 0 (Channel 0)	0x0
RSPREG1_0	0x7C200014	ROC	Response Register 1 (Channel 0)	0x0
RSPREG2_0	0x7C200018	ROC	Response Register 2 (Channel 0)	0x0
RSPREG3_0	0x7C20001C	ROC	Response Register 3 (Channel 0)	0x0

Register	Address	R/W	Description	Reset Value
RSPREG0_1	0x7C300010	ROC	Response Register 0 (Channel 1)	0x0
RSPREG1_1	0x7C300014	ROC	Response Register 1 (Channel 1)	0x0
RSPREG2_1	0x7C300018	ROC	Response Register 2 (Channel 1)	0x0
RSPREG3_1	0x7C30001C	ROC	Response Register 3 (Channel 1)	0x0

Register	Address	R/W	Description	Reset Value
RSPREG0_2	0x7C400010	ROC	Response Register 0 (Channel 2)	0x0
RSPREG1_2	0x7C400014	ROC	Response Register 1 (Channel 2)	0x0
RSPREG2_2	0x7C400018	ROC	Response Register 2 (Channel 2)	0x0
RSPREG3_2	0x7C40001C	ROC	Response Register 3 (Channel 2)	0x0

Name	Bit	Description	Initial 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 (upper) 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 shall check R[47:1], and if the response length is 136 the Host Controller shall check 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.

When the Host Controller modifies part of the *Response* register, as shown in the Table above, it shall preserve the unmodified bits.

27.7.9 BUFFER DATA PORT REGISTER

32-bit data port register to access internal buffer.

Register	Address	R/W	Description	Reset Value
BDATA0	0x7C200020	R/W	Buffer Data Register (Channel 0)	0x0
BDATA1	0x7C300020	R/W	Buffer Data Register (Channel 1)	0x0
BDATA2	0x7C400020	R/W	Buffer Data Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
BUFDAT		**Buffer Data** The Host Controller buffer can be accessed through this 32-bit single port SRAM memory. Write and Read memories are separated.	0

Detailed documents are to be copied from SD Host Standard Specification.

27.7.10 PRESENT STATE REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
PRNSTS0	0x7C200024	RO/ROC	Present State Register (Channel 0)	0x000A0000
PRNSTS1	0x7C300024	RO/ROC	Present State Register (Channel 1)	0x000A0000
PRNSTS2	0x7C400024	RO/ROC	Present State Register (Channel 2)	0x000A0000

Name	Bit	Description	Initial Value
	[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
-	[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 may be valid when **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 it 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 can be detected by this bit is set to 1 and **Card Inserted** is set to 0. The **Software Reset For All** in the *Software Reset* register shall not affect this bit. 1 = No Card or Inserted 0 = Reset or Debouncing	1 (After Reset)

Name	Bit	Description	Initial Value
INSCARD	[16]	**Card Inserted** (RO) This bit indicates whether a card has been inserted. The Host Controller shall debounce this signal so that the Host Driver will not need 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 *Software Reset* register will not affect this bit. If a card is removed when its power is on and its clock is oscillating, the Host Controller shall clear **SD Bus Power** in the *Power Control* register and **SD Clock Enable** in the *Clock Control* register. When this bit is changed from 1 to 0, the Host Controller shall immediately stop 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	0
	[15:12]	**Reserved**	
BUFRDRDY	[11]	**Buffer Read Enable** (ROC) This status is used for non-DMA read transfers. The Host Controller may implement 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 when all the block data is read from the buffer. A change of this bit from 0 to 1 occurs when block data is ready in the buffer and generates the **Buffer Read Ready** interrupt. 1 = Read enable 0 = Read disable	0
BUFWTRDY	[10]	**Buffer Write Enable (ROC)** **This status is used for non-DMA write transfers. The Host Controller can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer. A change of this bit from 1 to 0 occurs when all the block data is written to the buffer. A change of this bit from 0 to 1 occurs when top of block data can be written to the buffer and generates the Buffer Write Ready interrupt.** **1 = Write enable** **0 = Write disable**	0

Name	Bit	Description	Initial Value
RDTRANACT	[9]	**Read Transfer Active** (ROC) This status is used for detecting 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) When 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) When the last data block as specified by block length is transferred to the System. (2) When 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 when this bit changes to 0. 1 = Transferring data 0 = No valid data	0
WTTRANACT	[8]	**Write Transfer Active** (ROC) This status indicates 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) When writing a 1 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, a **Block Gap Event** interrupt is generated when this bit is changed to 0, as result of the **Stop At Block Gap Request** being set. This status is useful for the Host Driver in determining when to issue commands during write busy. 1 = Transferring data 0 = No valid data	0

Name	Bit	Description	Initial Value
	[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 can be set in either of the following cases: (1) After the end bit of the read command. (2) When writing a 1 to **Continue Request** in the *Block Gap Control* register to restart a read transfer. This bit can be cleared in either of the following cases: (1) When 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 will wait 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 can wait for current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the 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. This bit can be set in either of the following cases: (1) After the end bit of the write command. (2) When writing to 1 to **Continue Request** in the *Block Gap Control* register to continue a write transfer. This bit can be cleared in either of the following cases: (1) When the SD card releases write busy of the last data block the Host Controller will detect if output is not busy. If SD card does not drive busy signal for 8 SD Clocks, the Host Controller will consider the card drive “Not Busy”. (2) When 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	0

Name	Bit	Description	Initial Value
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 can save 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 = Can issue 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 can issue a SD Command using the ***CMD*** line. This bit is set immediately after the *Command* register (00Fh) is written. This bit is cleared when the command response is received. Even if the **Command Inhibit (DAT)** is set to 1, Commands using only the ***CMD*** line can be 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 shall remain 1 and the **Command Complete** is not set. Status issuing Auto CMD12 is not read from this bit. 1 = Cannot issue command 0 = Can issue 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

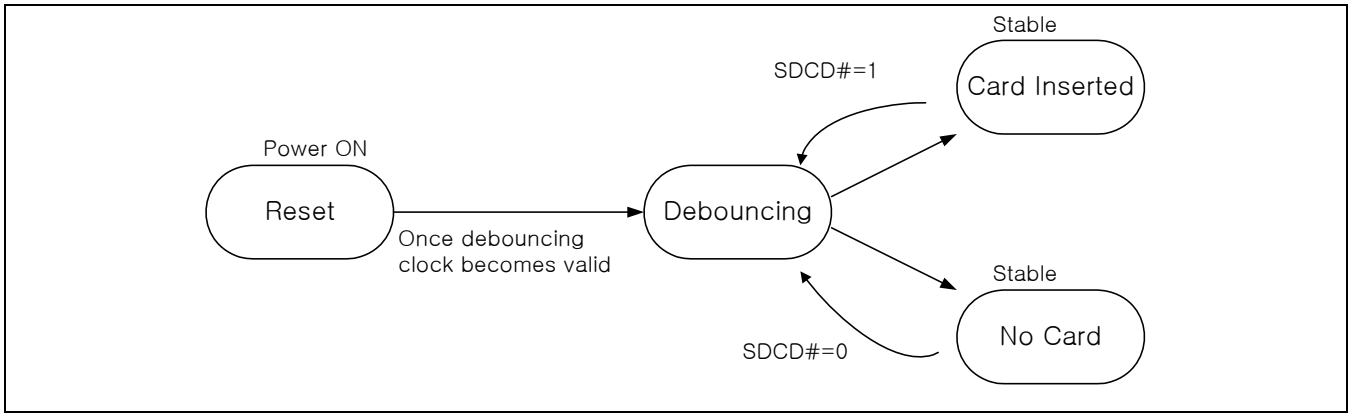


Figure 27-13. Card Detect State

The above Figure 27-13 shows the state definitions of hardware that handles “Debouncing”.

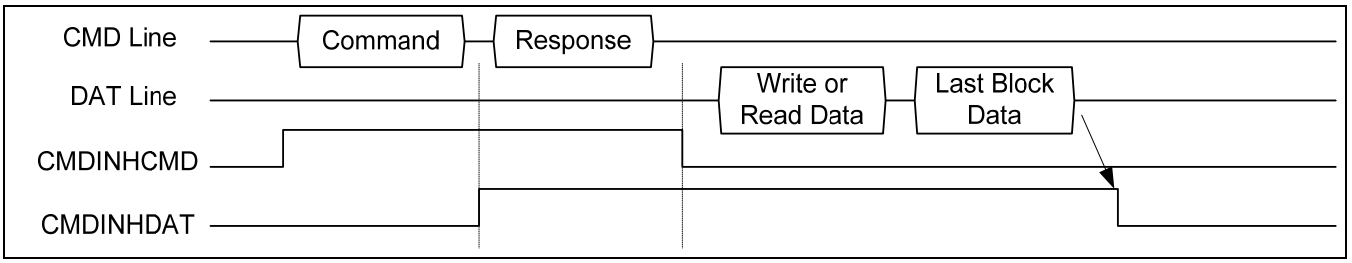


Figure 27-14. Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with data transfer

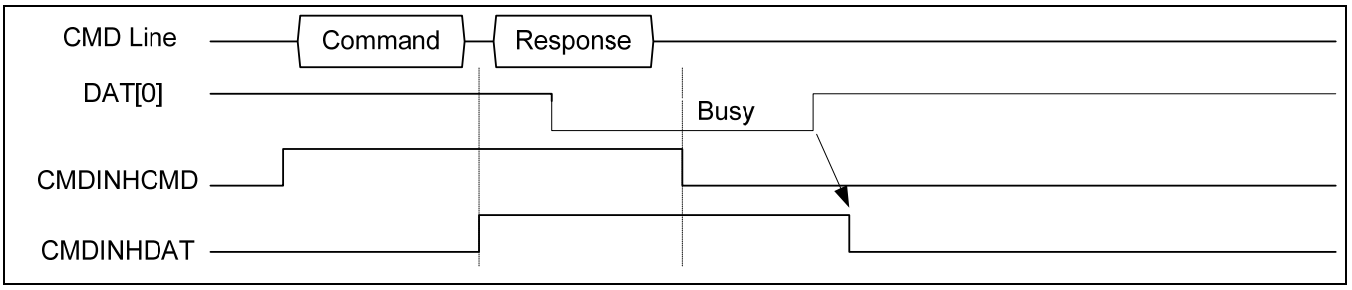


Figure 27-15. Timing of Command Inhibit (DAT) for the case of response with busy

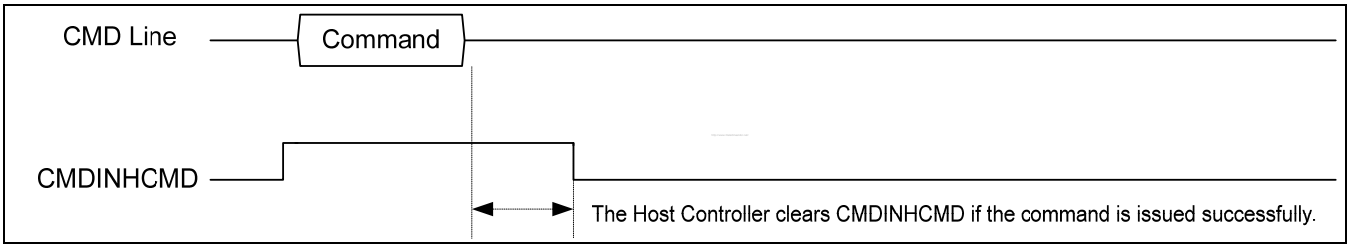


Figure 27-16. Timing of Command Inhibit (CMD) for the case of no response command

27.7.11 HOST CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
HOSTCTL0	0x7C200028	R/W	Present State Register (Channel 0)	0x0
HOSTCTL1	0x7C300028	R/W	Present State Register (Channel 1)	0x0
HOSTCTL2	0x7C400028	R/W	Present State Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
-	[7]	**Reserved** This field should be fixed to LOW	0
-	[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' = the 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 shall check support of DMA modes by referring the *Capabilities* register. Use of selected DMA is determined by **DMA Enable** of the *Transfer Mode* register. 00 = SDMA is selected 01 = Reserved 10 = 32-bit Address ADMA2 is selected 11 = 64-bit Address ADMA2 is selected (Not supported)	0
ENHIGHSPD	[2]	**High Speed Enable** This bit is optional. Before setting this bit, the Host Driver shall check 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 shall set it to match the data width of the SD card. '1' = 4-bit mode '0' = 1-bit mode	0
-	[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.

27.7.12 POWER CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
PWRCON0	0x7C200029	R/W	Present State Register (Channel 0)	0x0
PWRCON1	0x7C300029	R/W	Present State Register (Channel 1)	0x0
PWRCON2	0x7C400029	R/W	Present State Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	
SELPWRLVL	[3:1]	**SD Bus Voltage Select** By setting these bits, the Host Driver selects the voltage level for the SD card. Before setting this register, the Host Driver will check the **Voltage Support** bits in the *Capabilities* register. If an unsupported voltage is selected, the Host System will 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 will set **SD Bus Voltage Select**. If the Host Controller detects the No Card state, this bit will be cleared. If this bit is cleared, the Host Controller will immediately stop driving ***CMD*** and ***DAT[3:0]*** (tri-state) and drive ***SDCLK*** to low level. ‘1’ = Power on ‘0’ = Power off	0

27.7.13 BLOCK GAP CONTROL REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
BLKGAP0	0x7C20002A	R/W	Block Gap Control Register (Channel 0)	0x0
BLKGAP1	0x7C30002A	R/W	Block Gap Control Register (Channel 1)	0x0
BLKGAP2	0x7C40002A	R/W	Block Gap Control Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:4]	Reserved	0
ENINTBGAP	[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. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 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. When the Host Driver detects an SD card insertion, it will set this bit according to the CCCR of the SDIO card. (RW) '1' = Enabled, '0' = Disabled 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. When the Host Driver detects an SD card insertion, it will set 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 may occur. If this bit is set to 0, Suspend/Resume cannot be supported. (RW) '1' = Enable Read Wait Control, '0' = Disable 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

Name	Bit	Description	Initial Value
STOP BGAP	[0]	**Stop At Block Gap Request** This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the **Transfer Complete** is set to 1, indicating a transfer completion the Host Driver will leave this bit set to 1. Clearing both the **Stop At Block Gap Request** and **Continue Request** will not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The Host Controller shall honour **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	0

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate 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** can be used to restart the transfer.

(2) If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.

(3) If the Host Driver issues a Suspend command and the SD card does not accept it, the **Continue Request** is used to restart the transfer.

Any time **Stop At B**lock Gap Request stops the data transfer, the Host Driver will wait for Transfer Complete (in the *Normal Interrupt Status* register) before attempting to restart the transfer. When 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.

27.7.14 WAKEUP CONTROL REGISTER

This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver maintains voltage on the SD Bus, by setting **SD Bus Power** to 1 in the *Power Control* register, when wakeup event via Card Interrupt is desired.

Register	Address	R/W	Description	Reset Value
WAKCON0	0x7C20002B	R/W	Wakeup Control Register (Channel 0)	0x0
WAKCON1	0x7C30002B	R/W	Wakeup Control Register (Channel 1)	0x0
WAKCON2	0x7C40002B	R/W	Wakeup Control Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[7:3]	Reserved	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' = Enable '0' = Disable	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 '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 can be set to 1 if **FN_WUS** (Wake Up Support) in CIS is set to 1. (RW) '1' = Enable '0' = Disable	0

27.7.15 CLOCK CONTROL REGISTER

At the initialization of the Host Controller, the Host Driver sets the **SDCLK Frequency Select** according to the *Capabilities* register.

Register	Address	R/W	Description	Reset Value
CLKCON0	0x7C20002C	R/W	Command Register (Channel 0)	0x0
CLKCON1	0x7C30002C	R/W	Command Register (Channel 1)	0x0
CLKCON2	0x7C40002C	R/W	Command Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
SELFREQ	[15:8]	**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. 80h: base clock divided by 256 40h: base clock divided by 128 20h: base clock divided by 64 10h: base clock divided by 32 08h: base clock divided by 16 04h: base clock divided by 8 02h: base clock divided by 4 01h: base clock divided by 2 00h: base clock (10MHz-63MHz) 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 can be 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 will yield 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.	0

Name	Bit	Description	Initial Value
	[7:4]	**Reserved**	

-	[3]	**Reserved**	0
ENSDCLK	[2]	**SD Clock Enable** The Host Controller stops ***SDCLK*** when writing this bit to 0. **SDCLK Frequency Select** can be changed when this bit is 0. Then, the Host Controller shall maintain the same clock frequency until ***SDCLK*** is stopped (Stop at ***SDCLK***=0). If the **Card Inserted** in the *Present State register* is cleared, this bit will be cleared. **(RW)** '1' = Enable '0' = Disable	0
STBLINTCLK	[1]	**Internal Clock Stable** This bit is set to 1 when SD Clock is stable after writing to **Internal Clock Enable** in this register to 1. The SD Host Driver shall wait to set **SD Clock Enable** until this bit is set to 1. **Note:** This is useful when using PLL 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 when 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 very low power state. Still, registers shall be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, the Host Controller can be set **Internal Clock Stable** in this register to 1. This bit shall not affect card detection. **(RW)** '1' = Oscillate '0' = Stop	

27.7.16 TIMEOUT CONTROL REGISTER

At the initialization of the Host Controller, the Host Driver can set the **Data Timeout Counter Value** according to the *Capabilities* register.

Register	Address	R/W	Description	Reset Value
TIMEOUTCON0	0x7C20002E	R/W	Timeout Control Register (Channel 0)	0x0
TIMEOUTCON1	0x7C30002E	R/W	Timeout Control Register (Channel 1)	0x0
TIMEOUTCON2	0x7C40002E	R/W	Timeout Control Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[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 will be generated by dividing the base clock TMCLK value by this value. When 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

27.7.17 SOFTWARE RESET REGISTER

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 some time to complete software reset, the SD Host Driver shall confirm that these bits are 0.

Register	Address	R/W	Description	Reset Value
SWRST0	0x7C20002F	R/W	Software Reset Register (Channel 0)	0x0
SWRST1	0x7C30002F	R/W	Software Reset Register (Channel 1)	0x0
SWRST2	0x7C40002F	R/W	Software Reset Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[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

Name	Bit	Description	Initial Value
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 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 when capabilities registers are valid and the Host Driver can read them. If this bit is set to 1, the SD card shall reset itself and must be reinitialized by the Host Driver. (RWAC) '1' = Reset '0' = Work	0

27.7.18 NORMAL INTERRUPT STATUS REGISTER

The *Normal Interrupt Status Enable* affects reads of this register, but *Normal Interrupt Signal Enable* does not affect these reads. An interrupt is generated when 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 to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. The **Card Interrupt** is cleared when the card stops asserting the interrupt; that is, when the Card Driver services the interrupt condition.

Register	Address	R/W	Description	Reset Value
NORINTSTS0	0x7C200030	ROC/RW1C	Normal Interrupt Status Register (Channel 0)	0x0
NORINTSTS1	0x7C300030	ROC/RW1C	Normal Interrupt Status Register (Channel 1)	0x0
NORINTSTS2	0x7C400030	ROC/RW1C	Normal Interrupt Status Register (Channel 2)	0x0

Name	Bit	Description	Initial 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 can efficiently test for an error by checking this bit first. 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 When 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 When the FIFO Address of the SD clock side reaches the FIFO Interrupt Address register 2 values, this status bit is asserted.	0

Name	Bit	Description	Initial Value
STAFIA1	[12]	**FIFO SD Address Pointer Interrupt 1 Status (RW1C)** '0' = Occurred '1' = Not Occurred When 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 When 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 Note1) After checking response for the suspend command, release Read Wait interrupt status manually if BS = 0 (BS means 'Bus Status' field 'Bus Suspend' register in the SDIO card spec) Note2) Read Wait operation procedure is started after 4-SDCLK from the end of the block data read transfer.	0
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 shall detect the Card Interrupt without SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so 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. When this status has been 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 be remain set to high. (**RW1C**) Note2,3 '1' = Generate Card Interrupt '0' = No Card Interrupt	0

Name	Bit	Description	Initial Value
STACARDREM	[7]	**Card Removal** This status is set if the Card Inserted in the Present State register changes from 1 to 0. When 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 when 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. When 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 when the Host Driver clear this bit and interrupt event may not be generated. (RW1C) '1' = Card inserted '0' = Card state stable or Debouncing	0
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. Suppose that it is used for debugging. This interrupt shall not be generated after the **Transfer Complete**. (RW1C) '1' = **DMA Interrupt** is generated '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 when both a 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).	0

<table>
<tr><td></td><td></td><td>‘1’ = Transaction stopped at block gap
‘0’ = No Block Gap Event</td><td></td></tr>
<tr><td>STATRAN CMPLT</td><td>[1]</td><td>Transfer Complete
This bit is set when a read / write transfer is completed.
(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 when a data transfer is completed as specified by data length (After the last data has been read to the Host System). The second is when data has stopped at the block gap and completed 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 is when the last data is written to the SD card as specified by data length and the busy signal released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request in the Block Gap Control register and data transfers completed. (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 can be 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>0</td></tr>
<tr><td>STACMD CMPLT</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 can be 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 may check 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 can be cleared when write to 1 (RW1C).
3. SD/MMC Controller of the S3C6410 does not support “card interrupt at block gap” used when the multiple block 4-bit

operation.

27.7.19 ERROR INTERRUPT STATUS REGISTER

Signals defined in this register can be enabled by the *Error Interrupt Status Enable* register, but not by the *Error Interrupt Signal Enable* register. The interrupt is generated when 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 can be cleared at the one register write.

Register	Address	R/W	Description	Reset Value
ERRINTSTS0	0x7C200032	ROC/RW1C	Error Interrupt Status Register (Channel 0)	0x0
ERRINTSTS1	0x7C300032	ROC/RW1C	Error Interrupt Status Register (Channel 1)	0x0
ERRINTSTS2	0x7C400032	ROC/RW1C	Error Interrupt Status Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:10]	**Reserved**	0
STAADMAERR	[9]	**ADMA Error** This bit is set when 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 when 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 when detecting that one of the bits in *Auto CMD12 Error Status* register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur but also when 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 either when detecting 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 when detecting CRC error when transferring read data which uses the ***DAT*** line or when detecting the Write CRC status having a value of other than "010". '1' = Error '0' = No Error	0
STADATTOUTERR	[4]	**Data Timeout Error** Occurs when detecting 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.	0

		'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 when detecting 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 when detecting a CRC error in the command response. (2) The Host Controller detects a ***CMD*** line conflict by monitoring the ***CMD*** line when 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 will abort the command (Stop driving ***CMD*** line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish ***CMD*** line conflict. '1' = CRC Error generated '0' = No Error	0
STACMDTOUTER R	[0]	**Command Timeout Error** Occurs only 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 shall** also be set as shown in Table 33, this bit **shall** be set without waiting for 64 ***SDCLK*** cycles because the command will be aborted by the Host Controller. '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

27.7.20 NORMAL INTERRUPT STATUS ENABLE REGISTER

Setting to 1 enables Interrupt Status.

Register	Address	R/W	Description	Reset Value
NORINTSTSEN0	0x7C200034	R/W	Normal Interrupt Status Enable Register (Channel 0)	0x0
NORINTSTSEN1	0x7C300034	R/W	Normal Interrupt Status Enable Register (Channel 1)	0x0
NORINTSTSEN2	0x7C400034	R/W	Normal Interrupt Status Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15]	**Fixed to 0** The Host Driver shall control 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 when this bit is cleared and restarted when 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

Name	Bit	Description	Initial Value
ENSTACARDREM	[7]	**Card Removal Status Enable** ‘1’ = Enabled ‘0’ = Masked	0
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

27.7.21 ERROR INTERRUPT STATUS ENABLE REGISTER

Setting to 1 enables Error Interrupt Status.

Register	Address	R/W	Description	Reset Value
ERRINTSTSEN0	0x7C200036	R/W	Error Interrupt Status Enable Register (Channel 0)	0x0
ERRINTSTSEN1	0x7C300036	R/W	Error Interrupt Status Enable Register (Channel 1)	0x0
ERRINTSTSEN2	0x7C400036	R/W	Error Interrupt Status Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[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

27.7.22 NORMAL INTERRUPT SIGNAL ENABLE REGISTER

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same1 bit interrupt line. To enable interrupt generate set any of this bit to 1.

Register	Address	R/W	Description	Reset Value
NORINTSIGEN0	0x7C200038	R/W	Normal Interrupt Signal Enable Register (Channel 0)	0x0
NORINTSIGEN1	0x7C300038	R/W	Normal Interrupt Signal Enable Register (Channel 1)	0x0
NORINTSIGEN2	0x7C400038	R/W	Normal Interrupt Signal Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15]	**Fixed to 0** The Host Driver shall control 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 Singal 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

Name	Bit	Description	Initial Value
ENSIGCARDREM	[7]	**Card Removal Signal Enable** '1' = Enabled '0' = Masked	0
ENSIGCARDNS	[6]	**Card Insertion Signal Enable** '1' = Enabled '0' = Masked	0
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

27.7.23 ERROR INTERRUPT SIGNAL ENABLE REGISTER

This register is used to select which interrupt status is notified to the Host System as the interrupt. These status bits all share the same 1 bit interrupt line. To enable interrupt generate set any of this bit to 1.

Register	Address	R/W	Description	Reset Value
ERRINTSIGEN0	0x7C20003A	R/W	Error Interrupt Signal Enable Register (Channel 0)	0x0
ERRINTSIGEN1	0x7C30003A	R/W	Error Interrupt Signal Enable Register (Channel 1)	0x0
ERRINTSIGEN2	0x7C40003A	R/W	Error Interrupt Signal Enable Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[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

Detailed documents are to be copied from SD Host Standard Specification.

27.7.24 AUTOCMD12 ERROR STATUS REGISTER

When *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 only when the **Auto CMD12 Error** is set.

Register	Address	R/W	Description	Reset Value
ACMD12ERRSTS0	0x7C20003C	ROC	Auto CMD12 Error Status Register (Channel 0)	0x0
ACMD12ERRSTS1	0x7C30003C	ROC	Auto CMD12 Error Status Register (Channel 1)	0x0
ACMD12ERRSTS2	0x7C40003C	ROC	Auto CMD12 Error Status Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[15:8]	Reserved	0
STANCMDAER	[7]	**Command Not Issued By Auto CMD12 Error** Setting this bit to 1 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
	[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 when detecting 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 when detecting 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. Setting this bit to 1 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.

Table 27-3. 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* can be classified in three scenarios:

(1) When 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 of generating the **Auto CMD12 Error** and writing to the *Command* register are asynchronous. Then D07 are sampled when driver never writing to the *Command* register. So just before reading the *Auto CMD12 Error Status* register set the D07 status bit. An Auto CMD12 Error Interrupt is generated when 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.

27.7.25 CAPABILITIES REGISTER

This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to **Software Reset For All** in the *Software Reset* register for loading from flash memory and completion timing control.

Register	Address	R/W	Description	Reset Value
CAPAREG0	0x7C200040	HWInit	Capabilities Register (Channel 0)	0x05E00080
CAPAREG1	0x7C300040	HWInit	Capabilities Register (Channel 1)	0x05E00080
CAPAREG2	0x7C400040	HWInit	Capabilities Register (Channel 2)	0x05E00080

Name	Bit	Description	Initial Value
	[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
	[20]	**Reserved**	0
CAPAADMA2	[19]	**ADMA2 Support** This bit indicates whether the Host Controller is capable of using ADMA2. '1'=ADMA2 Support '0'=ADMA2 not Support	1
	[28]	**Reserved**	0

Name	Bit	Description	Initial Value
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 can be defined as indicated below. '00'=512-byte, '01'=1024-byte, '10'=2048-byte, '11'=Reserved	0
	[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
	[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

27.7.26 MAXIMUM CURRENT CAPABILITIES REGISTER

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.

Register	Address	R/W	Description	Reset Value
MAXCURR0	0x7C200048	HWInit	Maximum Current Capabilities Register (Channel 0)	0x0
MAXCURR1	0x7C300048	HWInit	Maximum Current Capabilities Register (Channel 1)	0x0
MAXCURR2	0x7C400048	HWInit	Maximum Current Capabilities Register (Channel 2)	0x0

Name	Bit	Description	Initial Value
	[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.

Table 27-4. Maximum Current Value Definition

Register Value	Current Value
0	Get information via another method
1	4mA
2	8mA
3	12mA
…	…
255	1020mA

27.7.27 CONTROL REGISTER 2

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
CONTROL2_0	0x7C200080	R/W	Control register 2 (Channel 0)	0x0
CONTROL2_1	0x7C300080	R/W	Control register 2 (Channel 1)	0x0
CONTROL2_2	0x7C400080	R/W	Control register 2 (Channel 2)	0x0

Name	Bit	Description	Initial Value
ENSTAASYNCCLR	[31]	**Write Status Clear Async Mode Enable** This bit can make 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) When 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
CDINVRXD3	[29]	**Card Detect signal inversion for RX_DAT[3]** 0=Disable, 1=Enable	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 27-13) 1= Card Removed state is "Card Out" State (When the transition from "Debouncing state to "No Card" state in Figure 27-13)	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

Name	Bit	Description	Initial Value
SDCDSEL	[13]	**SD Card Detect Signal Selection** Card Detect Pin Level does not simply reflect SDCD# pin, but chooses from SDCD, DAT[3], or CDTestlvl depending on CDSigSel and this field (SDCDSel) values '0'=nSDCD is used for SD Card Detect Signal '1'=DAT[3] is used for SD Card Detect Signal	0
SDSIGPC	[12]	**SD Output Signal Power Control Support** This field is used to enable output CMD and DAT referencing SD Bus Power bit in the "PWRCON register", when being set. '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
ENBUSYCHKTXSTART	[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
ENCLKOUTHOLD	[8]	**SDCLK Hold Enable** The enter and exit of the SDCLK Hold state is done by Host Controller. 0=Disable, 1=Enable	0
RWAITMODE	[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 only can be read through memory area. (Debug purpose)	0
SELBASECLK	[5:4]	**Base Clock Source Select** 00 or 01 =HCLK, 10=EPLL out Clock (from SYSCON), 11=External Clock source (XTI or XEXTCLK)	00
PWRSYNC	[3]	**SD OP Power Sync Support with SD Card** This field is used to enable input CMD and DAT referencing SD Bus Power bit in the "PWRCON register", when being set. '0'=No Sync, no switch input enable signal (Command, Data) '1'=Sync, control input enable signal (Command, Data)	0
-	[2]	**Reserved**	0
ENCLKOUTMSKCON	[1]	**SDCLK output clock masking when Card Insert cleared** This field when High is used not to stop SDCLK when No Card state.	0

		’0’=Disable, ‘1’=Enable	
HWINITFIN	[0]	**SD Host Controller Hardware Initialization Finish** 0=Not Finish, 1=Finish	0

NOTES:

1. Ensure to always 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 when SDCLK Hold Enable is set.

27.2.28 CONTROL REGISTERS 3 REGISTER

Register	Address	R/W	Description	Reset Value
CONTROL3_0	0x7C200084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 0)	0x7F5F3F1F
CONTROL3_1	0x7C300084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 1)	0x7F5F3F1F
CONTROL3_2	0x7C400084	R/W	FIFO Interrupt Control (Control Register 3) (Channel 2)	0x7F5F3F1F

Name	Bit	Description	Initial 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 when 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 when 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).

27.2.29 CONTROL REGISTER 4

Register	Address	R/W	Description	Reset Value
CONTROL4_0	0x7C20008C	R/W	Control register 4 (Channel 0)	0x0
CONTROL4_1	0x7C30008C	R/W	Control register 4 (Channel 1)	0x0
CONTROL4_2	0x7C40008C	R/W	Control register 4 (Channel 2)	0x0

Name	Bit	Description	Initial Value
-	[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 only 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	
STABUSY	[0]	**Status Busy** This bit is “High” when the clock domain crossing (HCLK to SDCLK) operation is processing. This bit is status bit and Read Only (RO)	0

27.2.30 FORCE EVENT REGISTER FOR AUTO CMD12 ERROR STATUS

Register	Address	R/W	Description	Reset Value
FEAER0	0x7C200050	WO	Force Event Auto CMD12 Error Interrupt Register (Channel 0)	0x0000
FEAER1	0x7C300050	WO	Force Event Auto CMD12 Error Interrupt Register (Channel 1)	0x0000
FEAER2	0x7C400050	WO	Force Event Auto CMD12 Error Interrupt Register (Channel 2)	0x0000

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

Name	Bit	Description	Initial Value
	[15:8]	-	0x0
FENCMDAER	[7]	**Force Event for Command Not Issued By Auto CMD12 Error** 1=Interrupt is generated 0=No Interrupt	0
	[6:5]	-	0
FECMDIDXERR	[4]	**Force Event for Auto CMD12 Index Error** 1=Interrupt is generated 0=No Interrupt	0
FECMDEBITAER	[3]	**Force Event for Auto CMD12 End Bit Error** 1=Interrupt is generated 0=No Interrupt	0
FECMDCRCAER	[2]	**Force Event for Auto CMD12 CRC Error** 1=Interrupt is generated 0=No Interrupt	0
FECMDTOUTAER	[1]	**Force Event for Auto CMD12 Timeout Error** 1=Interrupt is generated 0=No Interrupt	0
FENACMDAER	[0]	**Force Event for Auto CMD12 Not Executed** 1=Interrupt is generated 0=No Interrupt	0

27.2.31 FORCE EVENT REGISTER FOR ERROR INTERRUPT STATUS

Register	Address	R/W	Description	Reset Value
FEERR0	0x7C200052	WO	Force Event Error Interrupt Register Error Interrupt (Channel 0)	0x0000
FEERR1	0x7C300052	WO	Force Event Error Interrupt Register Error Interrupt (Channel 1)	0x0000
FEERR2	0x7C400052	WO	Force Event Error Interrupt Register Error Interrupt (Channel 2)	0x0000

The *Force Event* Register is not a physically implemented register. Rather, it is an address at which the *Error Interrupt Status* register can be written. The effect of a write to this address will be 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.

D15 D12

Name	Bit	Description	Initial Value
	[15:12]	**Force Event for Vendor Specific Error Status** Additional status bits can be defined in this register by the vendor. 1=Interrupt is generated 0=No Interrupt	0x0
	[11:10]		
FEADMAERR	[9]	**Force Event for ADMA Error** 1=Interrupt is generated 0=No Interrupt	0
FEACMDERR	[8]	**Force Event for Auto CMD12 Error** 1=Interrupt is generated 0=No Interrupt	0
FECURERR	[7]	**Force Event for Current Limit Error** 1=Interrupt is generated 0=No Interrupt	0
FEDENDERR	[6]	**Force Event for Data End Bit Error** 1=Interrupt is generated 0=No Interrupt	0
FEDATCRCERR	[5]	**Force Event for Data CRC Error** 1=Interrupt is generated 0=No Interrupt	0
FEDATTOUTERR	[4]	**Force Event for Data Timeout Error** 1=Interrupt is generated 0=No Interrupt	0
FECMDID	[3]	**Force Event for Command Index Error**	0

XERR		1=Interrupt is generated 0=No Interrupt	
FECMDEBITERR	[2]	**Force Event for Command End Bit Error** 1=Interrupt is generated 0=No Interrupt	0
FECMDCRCERR	[1]	**Force Event for Command CRC Error** 1=Interrupt is generated 0=No Interrupt	0
FECMDTOUTERR	[0]	**Force Event for Command Timeout Error** 1=Interrupt is generated 0=No Interrupt	0

27.2.32 ADMA ERROR STATUS REGISTER

When **ADMA Error** Interrupt is occurred, 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 when 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 may find that the Valid bit is not set in the error descriptor.

Register	Address	R/W	Description	Reset Value
ADMAERR0	0x7C200054	R/W	ADMA Error Status Register (Channel 0)	0x00
ADMAERR1	0x7C300054	R/W	ADMA Error Status Register (Channel 1)	0x00
ADMAERR2	0x7C400054	R/W	ADMA Error Status Register (Channel 2)	0x00

Name	Bit	Description	Initial Value
	[31:11]	Reserved	0x00
STAADMAFINBLK	[10]	**ADMA Final Block Transferred (ROC)** In ADMA operation mode, this field is set to High when the Transfer Complete condition and the block is final (no block transfer remains). If this bit is Low when the Transfer Complete condition, 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)** When the stop state by ADMA Interrupt, ADMA operation continues by setting this bit to HIGH.	0
ADMASTAINT	[8]	**ADMA Interrupt Status (RW1C)** This bit is set to HIGH when 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	00

		the Block Count and Block Length. (2) Total data length can not be divided by the block length. ‘0’ = No Error ‘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

27.2.33 ADMA SYSTEM ADDRESS REGISTER

This register contains the physical Descriptor address used for ADMA data transfer.

Register	Address	R/W	Description	Reset Value
ADMASYSADDR0	0x7C200058	R/W	ADMA System Address Register (Channel 0)	0x00
ADMASYSADDR1	0x7C300058	R/W	ADMA System Address Register (Channel 1)	0x00
ADMASYSADDR2	0x7C400058	R/W	ADMA System Address Register (Channel 2)	0x00

Name	Bit	Description	Initial 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 shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program 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

27.7.34 HOST CONTROLLER VERSION REGISTER

This register contains the SD Command Argument.

Register	Address	R/W	Description	Reset Value
HCVER0	0x7C2000FE	HWInit	Host Controller Version Register (Channel 0)	0x0401
HCVER1	0x7C3000FE	HWInit	Host Controller Version Register (Channel 1)	0x0401
HCVER2	0x7C4000FE	HWInit	Host Controller Version Register (Channel 2)	0x0401

Name	Bit	Description	Initial 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 Spec. 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

28 MIPI HSI INTERFACE CONTROLLER

28.1 OVERVIEW

MIPI HSI interface is a kind of high speed synchronous serial interface.

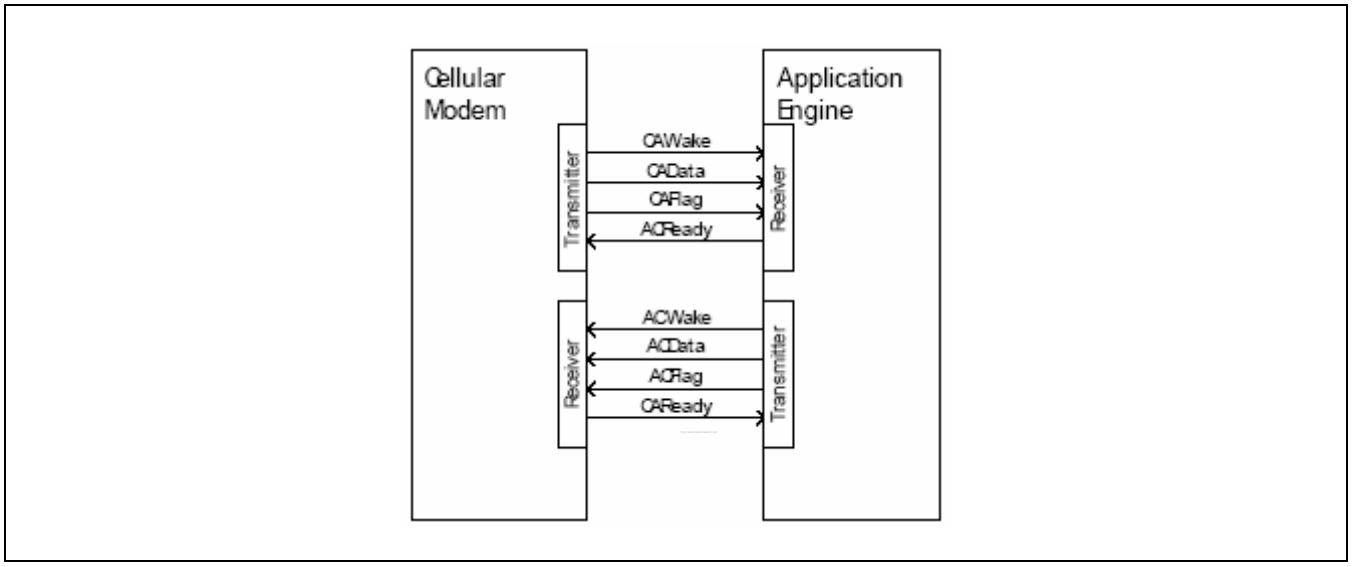


Figure 28-1. MIPI HSI signal definition Block Diagram

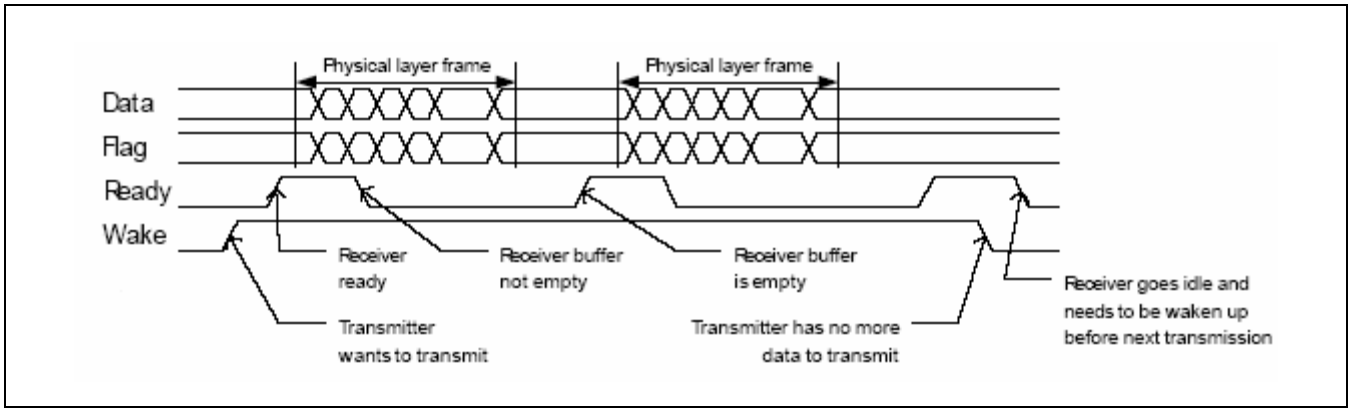


Figure 28-2. MIPI HSI transmitting example Block Diagram

28.2 FEATURES

The MIPI HSI Rx/Tx controller features:

The MIPI HSI interface is a uni-direction interface.

MIPI HSI Rx maximum bandwidth is 100Mbps. MIPI HSI TX controller uses PCLK for data transmitting.

28.2.1 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
  - Operation mode select (stream mode or frame mode)
  - Fixed channel ID mode
  - Number of channel
  - Generated Error clear
  - TxHOLD state timer & enable
  - TxIDLE state timer & enable
  - TxREQ state timer & 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 (128Byte)

28.2.3 RX 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 0
  - Operation mode select (stream mode or frame mode)
  - Fixed channel ID mode
  - Number of channel
  - Generated Error clear
  - RxACK state timer & enable
  - Rx state timer
- Configuration register 1
  - Rx FIFO clear
  - Rx FIFO timer & 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 (256Byte)

28.3 BLOCK DIAGRAM

28.3.1 TOP-LEVEL BLOCK DIAGRAM

Basic architectures of the Rx module part & the Tx module part are similar.

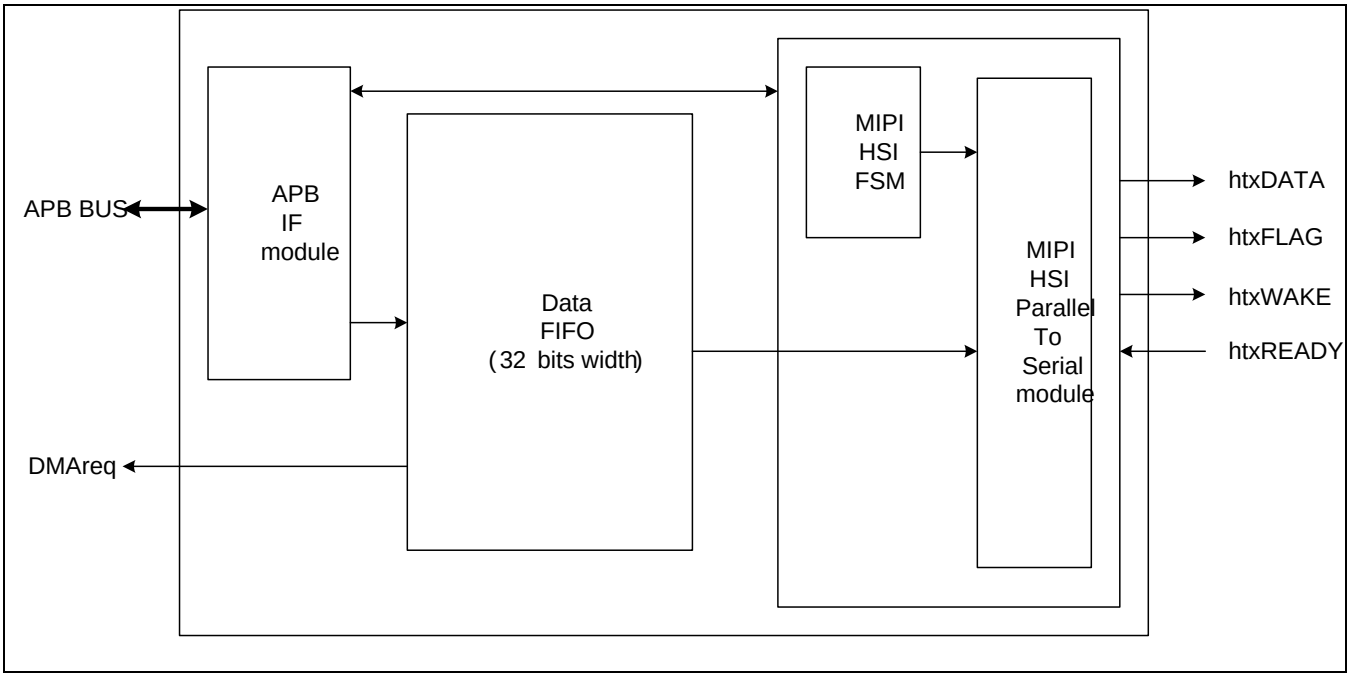


Figure 28-3. MIPI HSI interface controller Tx module Top Block Diagram

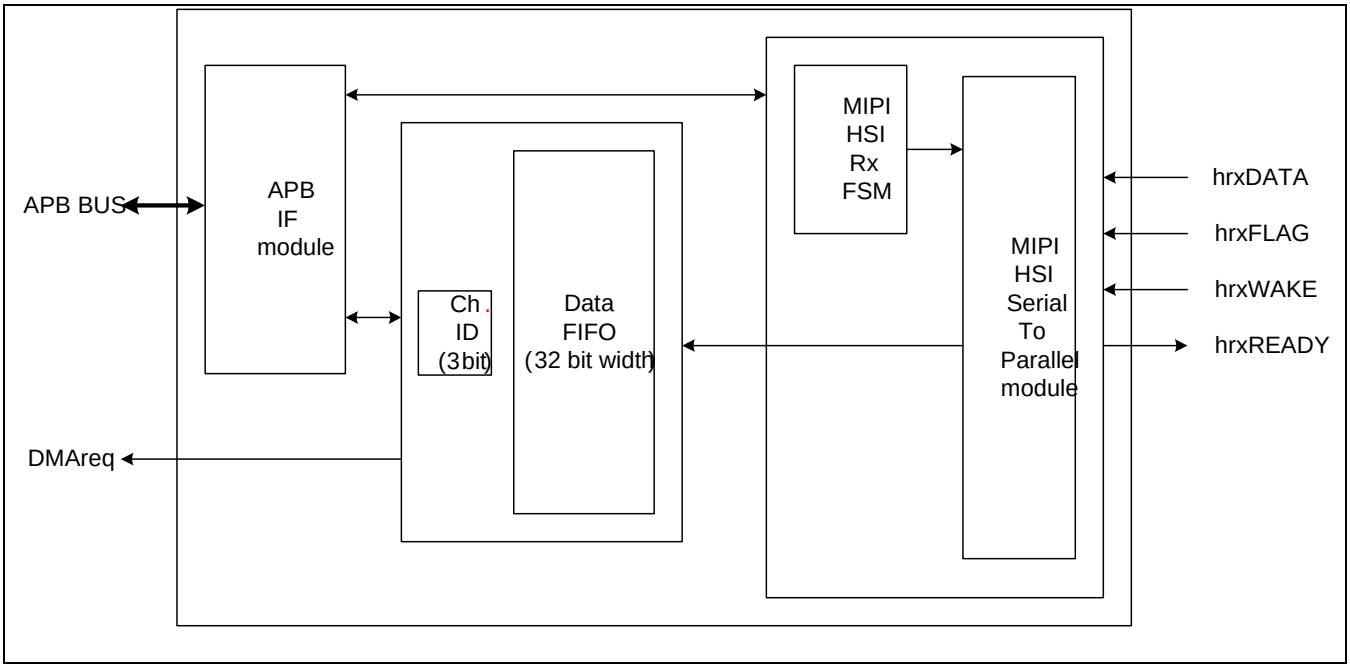


Figure 28-4. MIPI HSI interface controller Rx module Top Block Diagram

Tx module part parallel-to-serial block

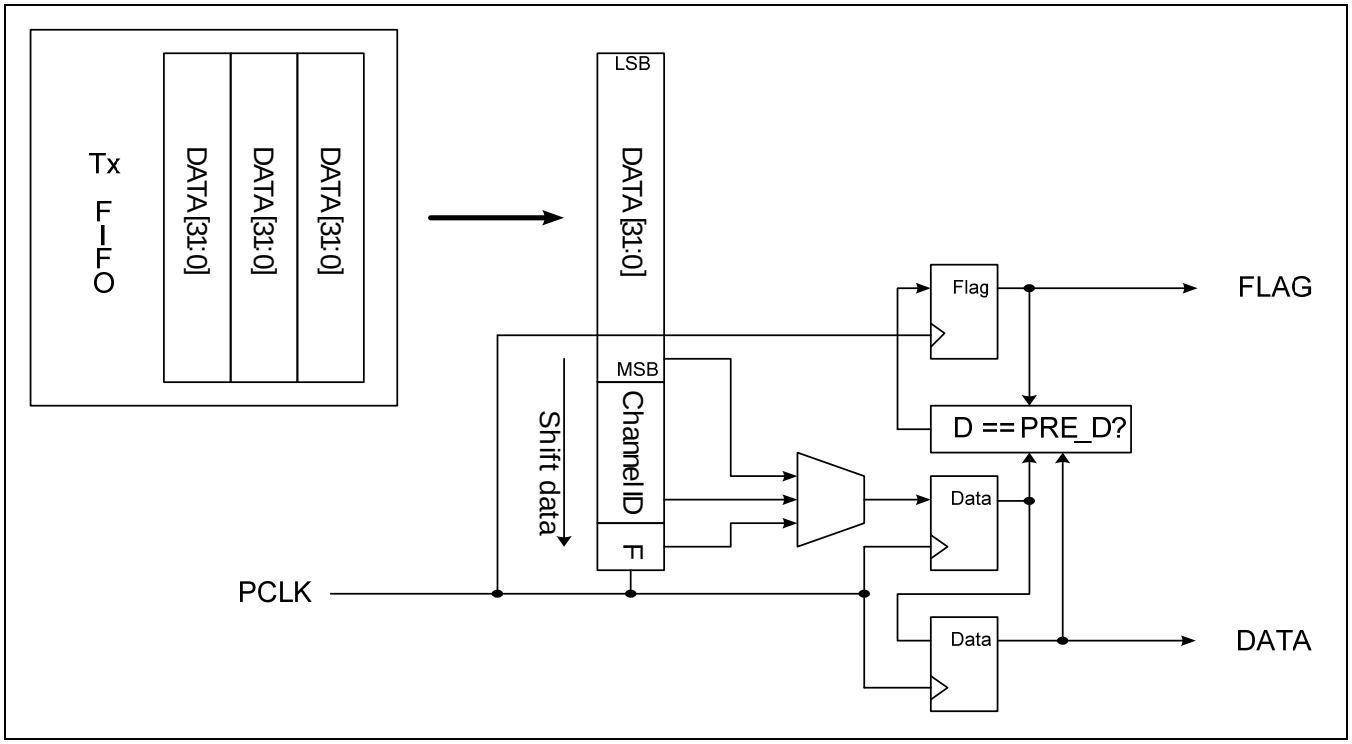


Figure 28-5. Parallel -to- Serial block (Tx module Part)

Rx module part serial-to-parallel block

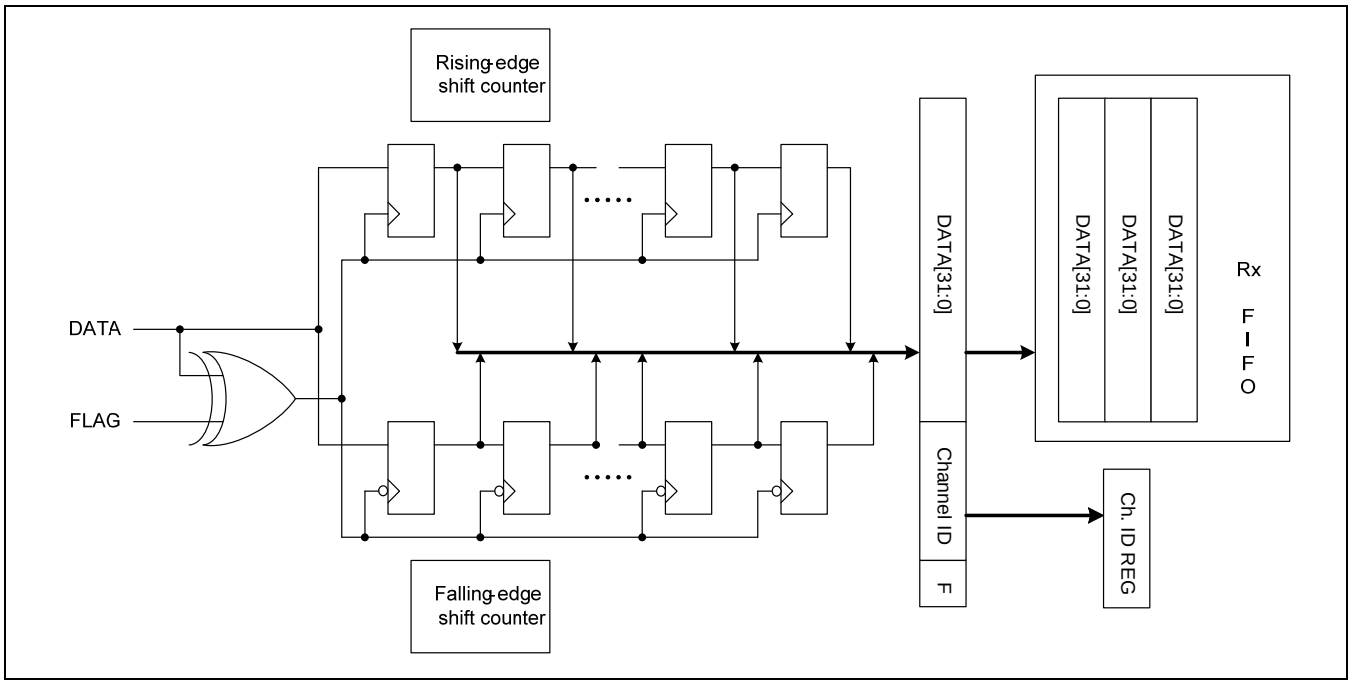


Figure 28-6. Serial-to-Parallel block (Rx module Part)

28.4 I/O DESCRIPTION

28.4.1 TX MODULE I/O LIST

Table 28-1. Tx I/O Description

Name	# bits	I/O	Function
MIPI HSI interface Signals (Tx)			
TX_DATA	1	O	MIPI HSI data line
TX_FLAG	1	O	MIPI HSI flag line
TX_WAKE	1	O	MIPI HSI wake up line to the other side Rx
TX_READY	1	I	MIPI HSI ready line from the other side Rx

If DMA request enable bit at interrupt & DMA request mask register is ‘enable’, Tx module will request DMA when FIFO is empty.

28.4.2 RX MODULE I/O LIST

Table 28-2. RX I/O Description

Name	# bits	I/O	Function
MIPI HSI interface Signals (Rx)			
RX_DATA	1	I	MIPI HSI data line
RX_FLAG	1	I	MIPI HSI flag line
RX_WAKE	1	I	MIPI HSI wake up line from the other side Tx
RX_READY	1	O	MIPI HSI ready line to the other side Tx

If DMA request enable bit at interrupt & DMA request mask register is ‘enable’ and DMA request threshold value bits at configuration register0 is set(0x00~0x11), Rx module will request DMA operation when the number of data in Rx FIFO is more than threshold value in configuration register0.

28.5 TIMING DIAGRAM

28.5.1 WAVEFORM

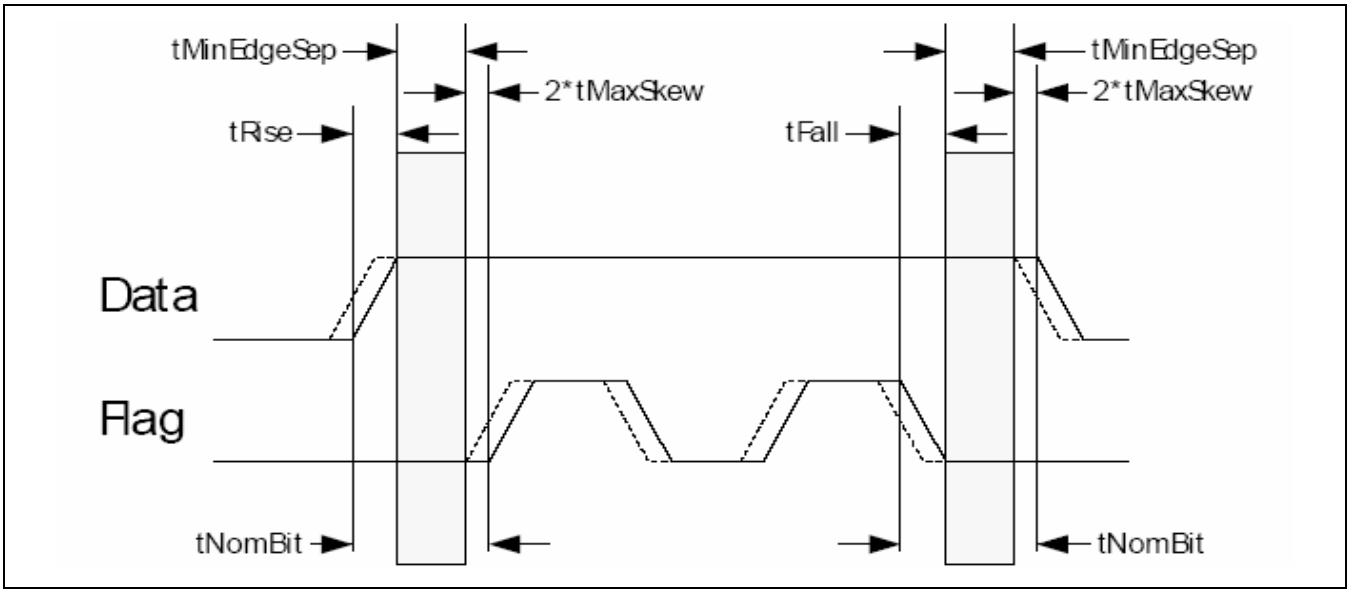


Figure 28-7. Waveform Block Diagram

28.5.2 SIGNAL TIMINGS

Table 28-3. Signal timings

Parameter	Description	1 Mbit/s	100 Mbit/s
TNomBit	Nominal bit time	1000 ns	10 ns
TMinEdgeSep	Minimum allowed separation of DATA and FLAG signal transitions	500 ns	5 ns
TMaxSkew	Maximum allowed time for combined skew and jitter	249 ns	1.5 ns
tRise and tFall	Minimum allowed signal rise and fall time	2 ns	2 ns

28.5.3 SINGLE/BURST CHANNEL ID MODE

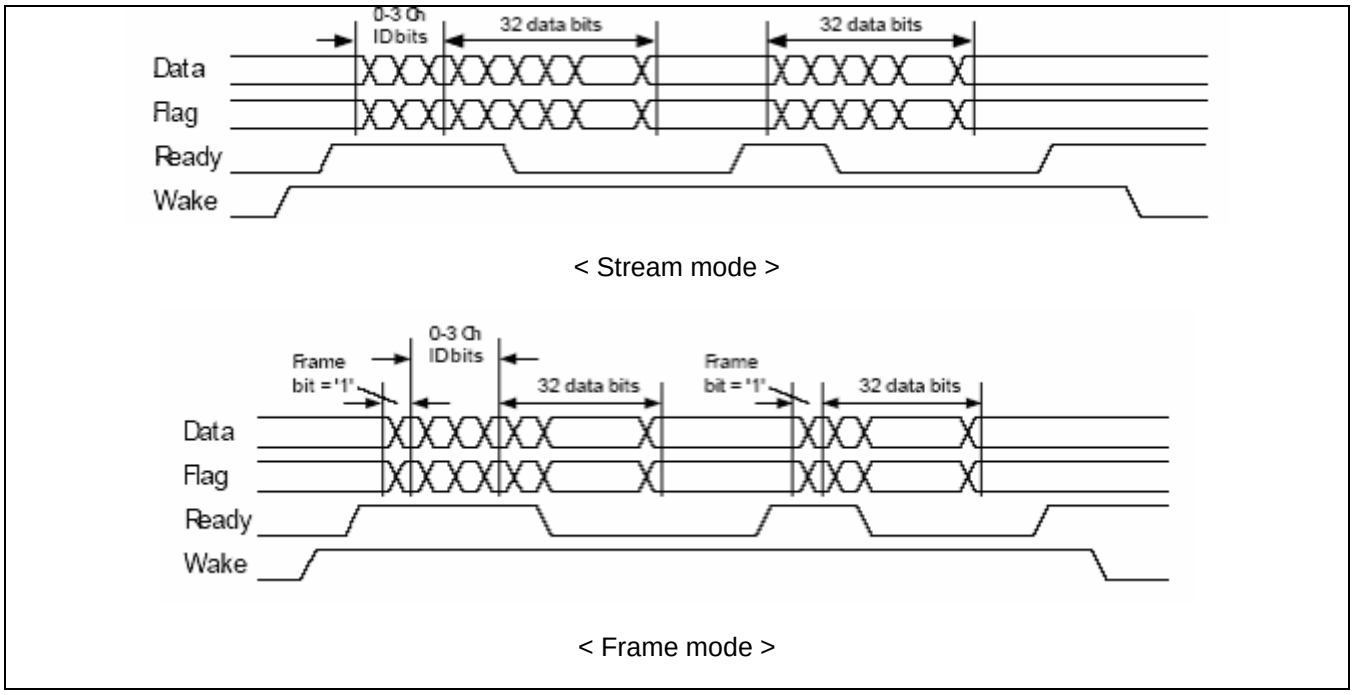


Figure 28-8. Example of Burst channel ID mode Block Diagram

In the Single channel ID mode, channel ID is attached in front of each data to send. In the Burst channel ID mode, channel ID is attached only in front of the first data frame which is sent after the IDLE state, and then only 32bit data are sent until going into IDLE mode again. The variable channel mode has a wider bandwidth in transferring a large data compared to the fixed channel ID mode, because the number of transferring channel ID is reduced.

28.5.4 STREAM MODE

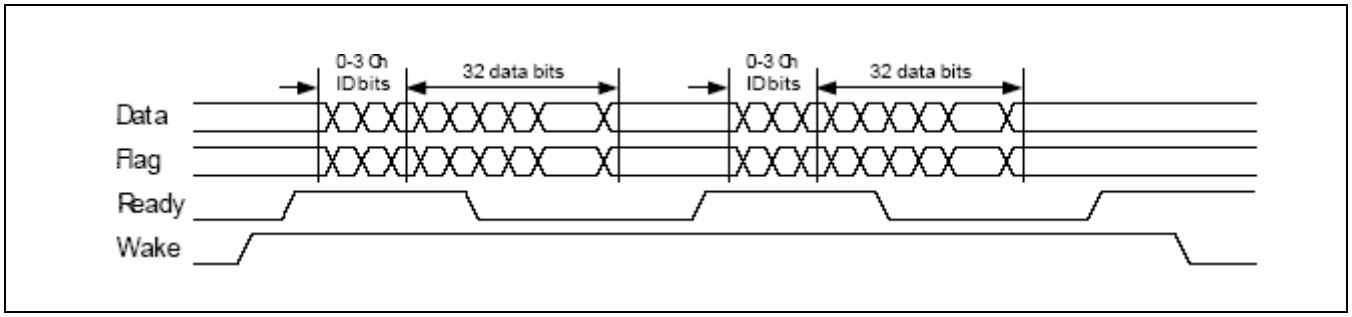


Figure 28-9. Example of Stream mode Block Diagram

28.5.5 FRAME MODE

Normal mode

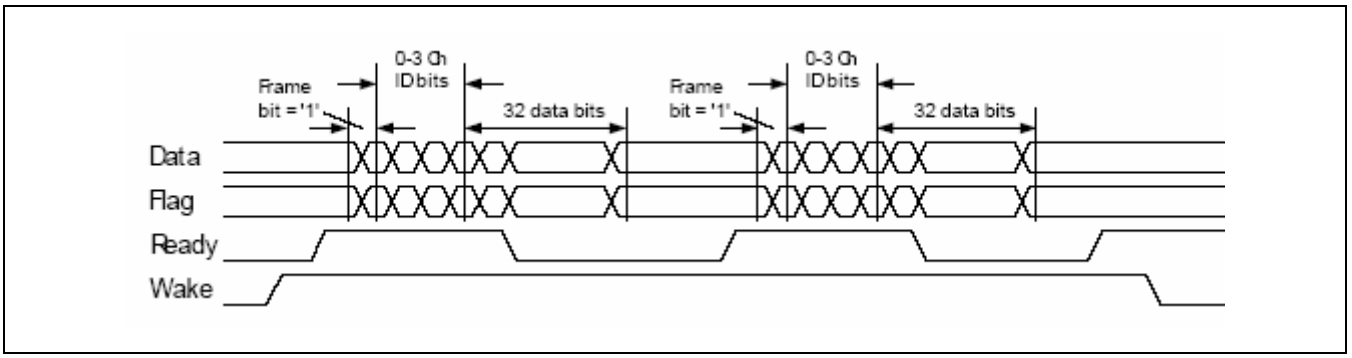


Figure 28-10. Example of Frame mode (normal mode) Block Diagram

Break frame

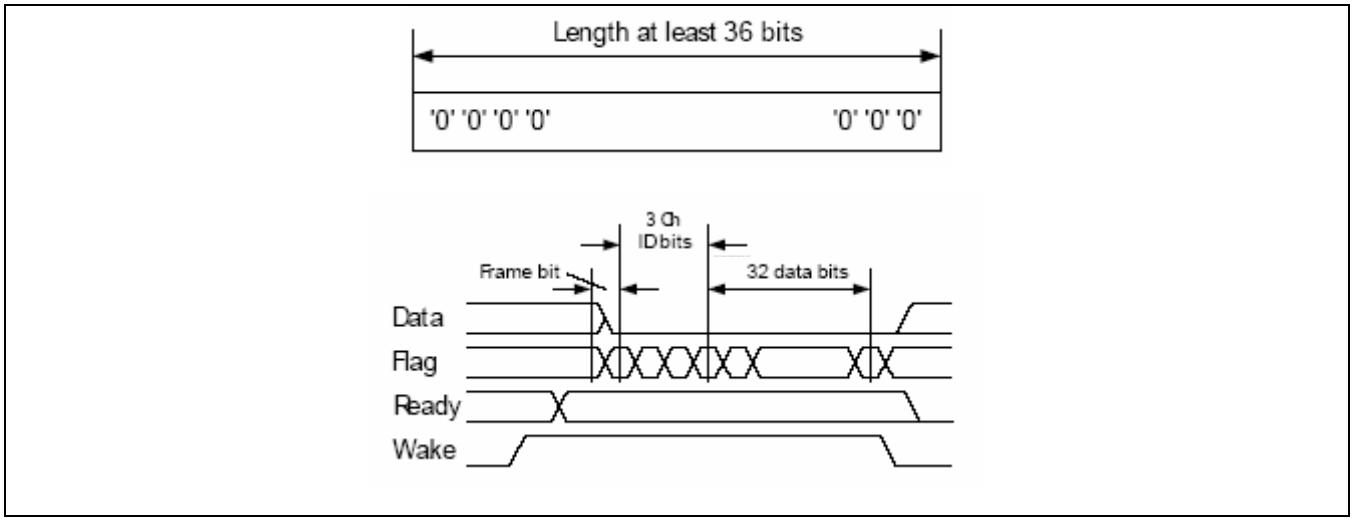


Figure 28-11. Break frame Block Diagram

The flag signal continues to toggle until the transfer is finished, because the break frame is transferring more than 36 ‘0’s. Tx module does not monitor the ready signal while it transfers the break frame, which is different from the normal mode. Therefore, ready signal is ignored as shown in Figure 28-11.

28.6 FUNCTIONAL DESCRIPTION

28.6.1 MIPI HSI TX CONTROLLER PART

Finite State Machine

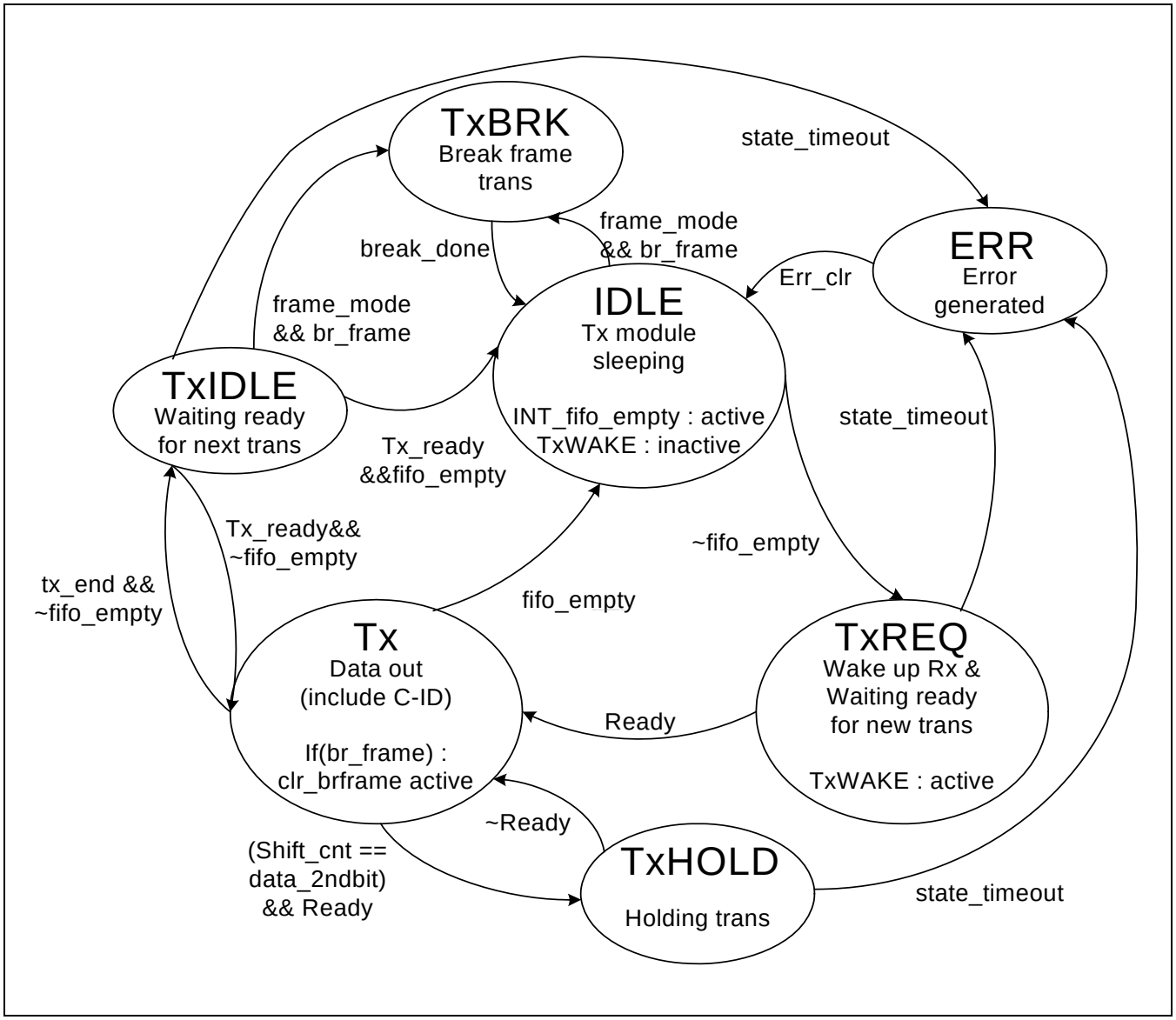


Figure 28-12. FSM of Tx module Part Block Diagram

28.6.2 MIPI HSI RX CONTROLLER PART

Finite State Machine

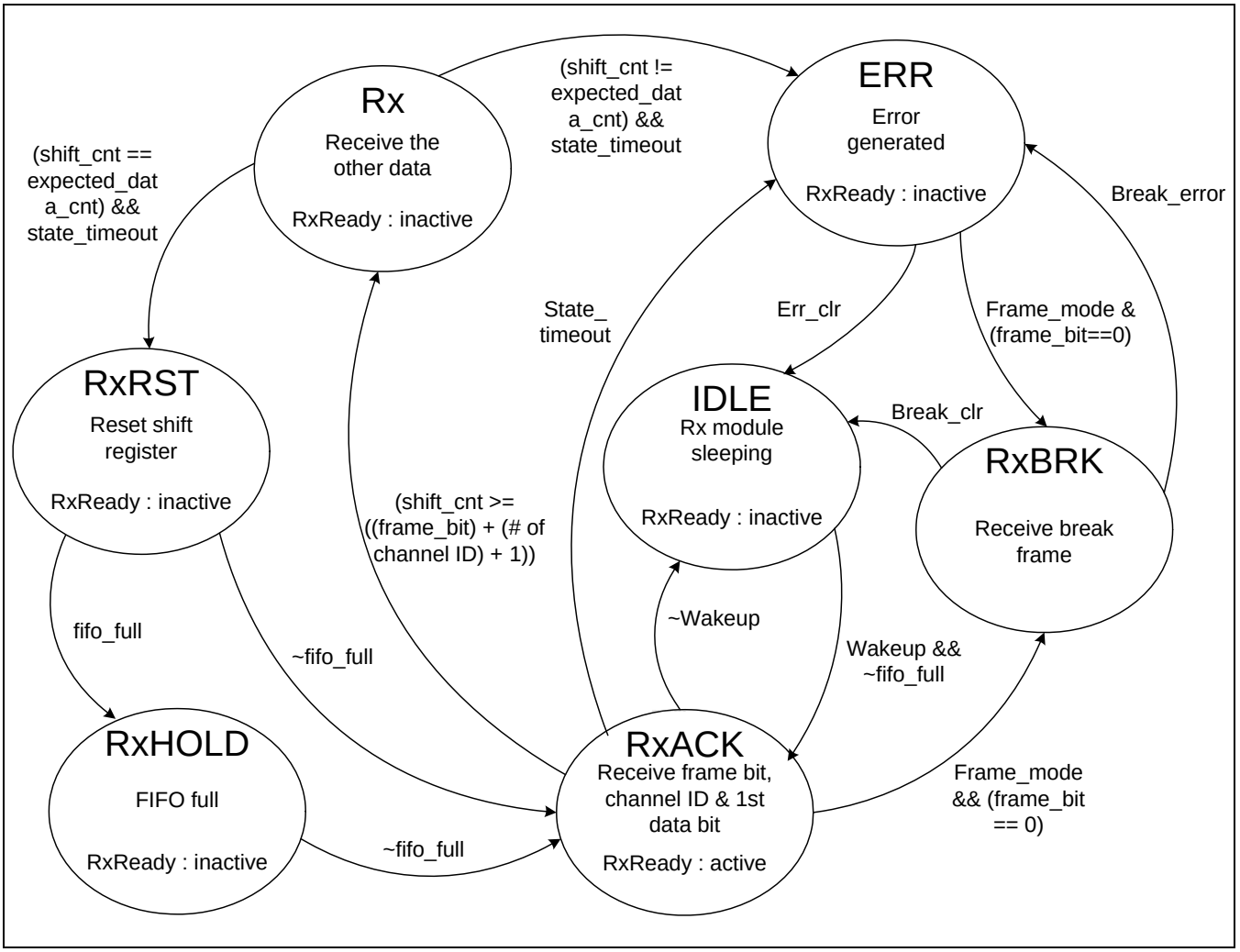


Figure 28-13. FSM of Rx module Part

Rx state is operated by a timer, which is set as the clock frequency input through MIPI HSI interface, the operation mode and the number of channel IDs. For example, let's assume that the clock frequency of MIPI is 100MHz, the number of channel IDs is 3, the operation mode is frame mode and the operating clock frequency of Rx module is 133MHz. One date frame is 10ns x 36 cycles, which requires 360ns to transfer. Dividing this value by 7.5ns (1/133MHz) yields 27cycles to transfer. By setting SFR as the value of 27-1, the Rx state timer can operate in optimal case. RxRST state of the FSM is a state to prepare for the next operation by reset the shift registers used for MIPI HSI.

The break frame can be fed even in the error state. The state goes to RxBREAK state if the frame bit is '0' when the clock toggles. (Ready signal is disabled in error state.)

28.7 SPECIAL FUNCTION REGISTERS

28.7.1 REGISTER MAP

MIPI HSI Tx Controller Register Map Table

Table 28-4. Tx Controller Register Map Table

Register	Address	Description	Reset Value
TX_STATUS_REG	0x7E006000	MIPI HSI Tx controller status register	0x00010000
TX_CONFIG_REG	0x7E006004	MIPI HSI Tx controller configuration register	0xFFFFFF02
Reseved	0x7E006008	Reserved register area	0x00000000
TX_INTSRC_REG	0x7E00600C	MIPI HSI Tx controller interrupt source register	0x00000000
TX_INTMSK_REG	0x7E006010	MIPI HSI Tx controller interrupt mask register	0x8000001F
TX_SWRST_REG	0x7E006014	Tx controller software reset	0x00000000
TX_CHID_REG	0x7E006018	MIPI HSI Tx controller channel ID register	0x00000000
TX_DATA_REG	0x7E00601C	MIPI HSI Tx controller data register (FIFO input)	0x00000000

MIPI HSI Rx Controller Register Map Table

Table 28-5. Rx Controller Register Map Table

Register	Address	Description	Reset Value
RX_STATUS_REG	0x7E007000	MIPI HSI Rx controller status register	0x00010000
RX_CONFIG0_REG	0x7E007004	MIPI HSI Rx controller configuration register	0x0FFFFF02
RX_CONFIG1_REG	0x7E007008	MIPI HSI Rx controller configuration register	0x00FFFFFF
RX_INTSRC_REG	0x7E00700C	MIPI HSI Rx controller interrupt source register	0x00000000
RX_INTMSK_REG	0x7E007010	MIPI HSI Rx controller interrupt mask register	0x800001FF
RX_SWRST_REG	0x7E007014	Rx controller software reset	0x00000000
RX_CHID_REG	0x7E007018	MIPI HSI Rx controller channel ID register	0x00000000
RX_DATA_REG	0x7E00701C	MIPI HSI Rx controller data register (FIFO output)	0x00000000

28.8 INDIVIDUAL REGISTER DESCRIPTIONS (TX CONTROLLER)

28.8.1 TX_STATUS_REG

TX_STATUS_REG is an internal logic monitoring window.

Table 28-6. TX_STATUS_REG register description

Register	Address	R/W	Description	Reset Value
TX_STATUS_REG	0x7E00_6000	R	MIPI HSI Tx controller status register	0x00010000

Bits	Name	Description	R/W	Reset Value
[31]	reserved	Reserved bit	R	0x0
[30:28]	next_state	Next state*	R	0x0
[27]	reserved	Reserved bit	R	0x0
[26:24]	current state	Current state*	R	0x0
[23:18]	reserved	Reserved bits	R	0x00
[17]	FIFO_full	TxFIFO full 0 : FIFO not full 1 : FIFO full	R	0x0
[16]	FIFO_empty	TxFIFO empty 0 : FIFO not empty 1 : FIFO empty	R	0x1
[15:13]	reserved	Reserved bits	R	0x0
[12:8]	tx_rd_point	TxFIFO read point	R	0x00
[7:5]	reserved	Reserved bits	R	0x0
[4:0]	tx_wr_point	TxFIFO write point	R	0x00

- State register value

  000 : IDLE 001 : TxREQ

  010 : Tx 011 : TxHOLD

  100 : TxIDLE 101 : Reserved state

  110 : TxBRK 111 : TxERR

28.8.2 TX_CONFIG_REG

CONFIG_REG is used to set the configuration of Tx controller.

Table 28-7. TX_CONFIG_REG register description

Register	Address	R/W	Description	Reset Value
TX_CONFIG_REG	0x7E00_6004	R/W	MIPI HSI Tx controller configuration register	0xFFFFFF02

Bits	Name	Description	R/W	Reset Value
[31:24]	TxHOLD time	TxHOLD state timer setting value	R/W	0xFF
[23:16]	TxIDLE time	TxIDLE state timer setting value	R/W	0xFF
[15:8]	TxREQ time	TxREQ state timer setting value	R/W	0xFF
[7]	TxHOLD time_en	TxHOLD state timer enabler 0 : disable 1 : enable	R/W	0x0
[6]	TxIDLE time_en	TxIDLE state timer enabler 0 : disable 1 : enable	R/W	0x0
[5]	TxREQ time_en	TxREQ state timer enabler 0 : disable 1 : enable	R/W	0x0
[4]	Err_clr	Generated Error clear 0 : stay 1 : clear	R/W	0x0
[3:2]	Width of CHID	Width of channel ID	R/W	0x0
[1]	Burst_mode	Fixed channel ID mode 0 : Burst ch ID mode 1 : Single ch ID mode	R/W	0x1
[0]	Frame_mode	Frame mode 0 : Stream mode 1 : Frame mode	R/W	0x0

28.8.3 TX_INTSRC_REG

INTSRC_REG is interrupt source pending register.

Table 28-8. TX_INTSRC_REG register description

Register	Address	R/W	Description	Reset Value
TX_INTSRC_REG	0x7E00_600C	R/W	MIPI HSI Tx controller interrupt source register	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:5]	Reserved	Reserved bits	R	0x0000001
[4]	TxH_timeout	TxHOLD state timeout interrupt (set ‘1’ for clearing)	R/W	0x0
[3]	TxI_timeout	TxIDLE state timeout interrupt (set ‘1’ for clearing)	R/W	0x0
[2]	TxR_timeout	TxREQ state timeout interrupt (set ‘1’ for clearing)	R/W	0x0

Bits	Name	Description	R/W	Reset Value
[1]	Brframe_end	Break frame transfer–done in Frame mode (set '1' for clearing)	R/W	0x0
[0]	TxFIFO_empty	TxFIFO empty interrupt (set '1' for clearing)	R/W	0x1

28.8.4 TX_INTMSK_REG

INTMSK_REG is interrupt mask & DMA request enabler register.

Table 28-9. TX_INTMSK_REG register description

Register	Address	R/W	Description	Reset Value
TX_INTMSK_REG	0x7E00_6010	R/W	MIPI HSI Tx controller interrupt mask register	0x8000001F

Bits	Name	Description	R/W	Reset Value
[31]	DMA_req_en	DMA request signal enable 0: enable 1: disable	R/W	0x1
[30:5]	Reserved	Reserved bits	R	0x0000000
[4]	TxH_timeout_msk	TxHOLD state timeout interrupt mask 0 : unmask 1 : mask	R/W	0x1
[3]	TxI_timeout_msk	TxIDLE state timeout interrupt mask 0 : unmask 1 : mask	R/W	0x1
[2]	TxR_timeout_msk	TxREQ state timeout interrupt mask 0 : unmask 1 : mask	R/W	0x1
[1]	Brframe_end_msk	Break frame transfer–done mask (in Frame mode) 0 : unmask 1 : mask	R/W	0x1
[0]	TxFIFO_empty_msk	TxFIFO empty interrupt mask 0 : unmask 1 : mask	R/W	0x1

28.8.5 TX_SWRST_REG

SWRST_REG is software reset.

Table 28-10. TX_SWRST_REG register description

Register	Address	R/W	Description	Reset Value
TX_SWRST_REG	0x7E00_6014	R/W	Tx controller software reset	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:1]	Reserved	Reserved bits	R	0x00000000
[0]	Sw_rst	Software reset 0 : set 1 : reset	R/W	0x0

28.8.6 TX_CHID_REG

CHID_REG is used to transfer channel ID.

Table 28-11. TX_CHID_REG register description

Register	Address	R/W	Description	Reset Value
TX_CHID_REG	0x7E00_6018	R/W	MIPI HSI Tx controller channel ID register	0x00000000

Bits	Name	Description	R/W	Reset Value
[31]	Break_frame	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.	R/W/C	0x0
[30]	Auto_clr	Break frame auto clear bit 0 : auto clear & TxBRK state end 1 : auto clear disable & TxBRK state continue	R/W	0x0
[29]	Br_frame_clr	Stop break frame continuing transfer	W	0x0
[28:3]	Reserved	Reserved bits	R	0x0000000
[2:0]	CHID	Channel ID	R/W	0x0

NOTE: In order to send data , the TX_CHID_REG must be set first and then the data to be transfered are pushed into the data fifo. The same channel ID is attached to each of the data when it sent through TxDATA. If the channel ID is different from the previous one, you must set the new channel ID into the TX_CHID_REG before pushing data into the data fifo. Break frame is sent to Rx side when ‘1’ is inputted for the Break_frame bit in the frame mode. This bit is automatically cleared after the transmission is completed (36 ‘0’s are transffered) in the auto clear mode (auto clear bit =0). In this case, the internal state goes to IDLE state. If not in the auto clear mode, the TxDATA continues to transfer ‘0’ while the internal state maintained in TxBRK state. Meanwhile, if the br_frame_clr bit is written as ‘1’, the state changes from RxBRK to IDLE and the TxDATA stops to transfer ‘0’s.

28.8.7 TX_DATA_REG

TX_DATA_REG is TxFIFO input.

Table 28-12. DATA_REG register description

Register	Address	R/W	Description	Reset Value
TX_DATA_REG	0x7E00_601C	R/W	MIPI HSI Tx controller data register (FIFO input)	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:0]	TxFIFO in	TxFIFO data input for transmitting	W	0x0

NOTE: As soon as data are loaded on TxFIFO, data in TxFIFO are transferred to the other side's RX through MIPI HSI Tx controller until TxFIFO is fully empty.

28.9 INDIVIDUAL REGISTER DESCRIPTIONS (RX CONTROLLER)

28.9.1 RX_STATUS_REG

RX_STATUS_REG is an internal logic monitoring window.

Table 28-13. RX_STATUS_REG register description

Register	Address	R/W	Description	Reset Value
RX_STATUS_REG	0x7E00_7000	R/W	MIPI HSI Rx controller status register	0x00010000

Bits	Name	Description	R/W	Reset Value
[31]	Reserved	Reserved bit	R	0x0
[30:28]	Next_state	Next state*	R	0x0
[27]	Reserved	Reserved bit	R	0x0
[26:24]	Curr_state	Current state*	R	0x0
[23:19]	Reserved	Reserved bits	R	0x00
[18]	FIFO_timeout	RxFIFO read timeout 0 : in time 1 : time out	R	0x0
[17]	FIFO_full	RxFIFO full 0 : FIFO not full 1: FIFO full	R	0x0
[16]	FIFO_empty	RxFIFO empty 0 : FIFO not empty 1: FIFO empty	R	0x1
[15:14]	Reserved	Reserved bits	R	0x0
[13:8]	Rx_rd_point	RxFIFO read point	R	0x00
[7:6]	Reserved	Reserved bits	R	0x0
[5:0]	Rx_wr_point	RxFIFO write point	R	0x00

- State register value

  000 : IDLE 001 : RxACK

  010 : Rx 011 : RxHOLD

  100 : RxBREAK 101 : Reserved state

  110 : RxRST 111 : RxERR

28.9.2 RX_CONFIG0_REG

RX_CONFIG0_REG is used to set the configuration of Rx controller.

Table 28-14. RX_CONFIG0_REG register description

Register	Address	R/W	Description	Reset Value
RX_CONFIG0_REG	0x7E00_7004	R/W	MIPI HSI Rx controller configuration register	0x0FFFFF02

Bits	Name	Description	R/W	Reset Value
[31:30]	Reserved	Reserved bits	R	0x0
[29:28]	DREQ_thres_val	DMA request threshold value DMA request signal is active when valid data in FIFO is 0x00 : full 0x01 : more than 4word 0x10 : more than 8word 0x11 : more than 16word	R/W	0x00
[27:16]	Rx_state time	Rx state timer setting value	R/W	0xFFF
[15:8]	RxACK time	RxACK state timer setting value	R/W	0xFF
[7]	Reserved	Reserved bit	R	0x0
[6]	RxACK time_en	RxACK state timer enabler 0 : disable 1 : enable	R/W	0x0
[5]	Break_clr	RxBREAK state clear bit 0 : disable 1 : enable	R/W	0x0
[4]	Err_clr	Generated Error clear	R/W	0x0
[3:2]	Width of CHID	Width of channel ID	R/W	0x0
[1]	Burst_mode	Fixed channel ID mode 0 : Burst ch ID mode 1 : Single ch ID mode	R/W	0x1
[0]	Frame_mode	Frame mode 0 : Stream mode 1 : Frame mode	R/W	0x0

28.9.3 RX_CONFIG1_REG

RX_CONFIG1_REG is used to set the configuration of Rx FIFO.

Table 28-15. RX_CONFIG1_REG register description

Register	Address	R/W	Description	Reset Value
RX_CONFIG1_REG	0x7E00_7008	R/W	MIPI HSI Rx controller configuration register	0x00FFFFFF

Bits	Name	Description	R/W	Reset Value
[31]	RxFIFO_clr	Break frame receiving timer setting value	R/W	0x0
[30:28]	Reserved	Reserved bits	R	0x0
[27]	RxFIFO_timer_en	RxFIFO timer enabler	R/W	0x0
[26:24]	Reserved	Reserved bits	R	0x0
[23:0]	RxFIFO_time	RxFIFO timer setting value	R/W	0xFFFFFF

28.9.4 RX_INTSRC_REG

INTSRC_REG is interrupt source panding register.

Table 28-16. RX_INTSRC_REG register description

Register	Address	R/W	Description	Reset Value
RX_INTSRC_REG	0x7E00_700C	R/W	MIPI HSI Rx controller interrupt source register	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:8]	Reserved	Reserved bits	R	0x000000
[7]	Break_done	Received Break frame in Frame mode (set ‘1’ for clearing)	R/W	0x0
[6]	Added_clock	Added clock input (set ‘1’ for clearing)	R/W	0x0
[5]	Missed_clock	Missed clock input interrupt (set ‘1’ for clearing)	R/W	0x0
[4]	RxACK_timeout	RxACK state timeout interrupt (set ‘1’ for clearing)	R/W	0x0
[3]	Brframe_err	Received data is not break frame. (set ‘1’ for clearing)	R/W	0x0
[2]	RxDONE	Data receiving is Done. (set ‘1’ for clearing)	R/W	0x0
[1]	RxFIFO_timeout	RxFIFO timeout but RxFIFO is not empty. (set ‘1’ for clearing)	R/W	0x0
[0]	RxFIFO_full	RxFIFO full interrupt (set ‘1’ for clearing)	R/W	0x0

28.9.5 RX_INTMSK_REG

RX_INTMSK_REG is interrupt mask & DMA reqeust enabler register.

Table 28-17. RX_INTMSK_REG register description

Register	Address	R/W	Description	Reset Value
RX_INTMSK_REG	0x7E00_7010	R/W	MIPI HSI Rx controller interrupt mask register	0x800001FF

Bits	Name	Description	R/W	Reset Value
[31]	DMA_req_en	DMA request signal enable 0: enable 1: disable	R/W	0x1
[30:9]	Reserved	Reserved bits	R	0x0000000
[8]	wakeup_enn	MIPI wake up enabler 0 : enable 1 : disable	R/W	0x1
[7]	Break_done_msk	Break frame done interrupt mask 0 : unmask 1 : mask	R/W	0x1
[6]	Added_clock_msk	Added clock input interrupt mask 0 : unmask 1 : mask	R/W	0x1
[5]	Missed_clock_msk	Missed clock input interrupt mask 0 : unmask 1 : mask	R/W	0x1
[4]	RxACK_timeout_msk	RxACK state timeout interrupt mask 0 : unmask 1 : mask	R/W	0x1
[3]	Brframe_err_msk	Break frame err interrupt mask 0 : unmask 1 : mask	R/W	0x1
[2]	RxDONE_msk	RxDONE interrupt mask 0 : unmask 1 : mask	R/W	0x1
[1]	RxFIFO_timeout_msk	RxFIFO_timeout interrupt mask 0 : unmask 1 : mask	R/W	0x1
[0]	RxFIFO_full_msk	RxFIFO full interrupt mask 0 : unmask 1 : mask	R/W	0x1

28.9.6 RX_SWRST_REG

SWRST_REG is software reset.

Table 28-18. RX_SWRST_REG register description

Register	Address	R/W	Description	Reset Value
RX_SWRST_REG	0x7E00_7014	R/W	Rx controller software reset	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:1]	Reserved	Reserved bits	R	0x00000000
[0]	Sw_rst	Software reset 0 : set 1 : reset	R/W	0x0

28.9.7 RX_CHID_REG

RX_CHID_REG is channel ID RxFIFO output.

Table 28-19. RX_CHID_REG register description

Register	Address	R/W	Description	Reset Value
RX_CHID_REG	0x7E00_7018	R/W	MIPI HSI Rx controller channel ID register	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:3]	Reserved	Reserved bits	R	0x0000000
[2:0]	CURR_ID	Current Channel ID	R	0x0

NOTE: Because Data in RxFIFO have the same channel ID, Channel ID should be changed after making RxFIFO empty,.

28.9.8 RX_DATA_REG

RX_DATA_REG is RxFIFO output.

Table 28-20. RX_DATA_REG register description

Register	Address	R/W	Description	Reset Value
RX_DATA_REG	0x7E00_701C	R	MIPI HSI Rx controller data register (FIFO output)	0x00000000

Bits	Name	Description	R/W	Reset Value
[31:0]	RxFIFO out	RxFIFO data output	R	0x0

29 SPI

29.1 OVERVIEW

The Serial Peripheral Interface (SPI) can interface the serial data transfer. SPI includes two 8, 16, 32-bit shift registers for transmission and receiving, respectively. During 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.

29.2 FEATURES

The SPI supports the following features:

- Full duplex
- 8/16/32-bit shift register for TX/RX
- 8-bit prescale logic
- 3 clocks source
- 8-bit/16-bit/32-bit bus interface
- The Motorola SPI protocol and National Semiconductor Microwire
- Two independent transmit and receive FIFOs, each 16 samples deep by 32-bits wide
- Master-mode and Slave-mode
- Receive-without-transmit operation
- Tx/ Rx maximum frequency at up to 50MHz

29.3 SIGNAL DESCRIPTIONS

The following table lists the external signals between the SPI and external device. All ports of the SPI can be used as General Purpose I/O ports when disable. For more information refer to “General Purpose I/O” chapter.

Table 29-1. External signals description

Name	Direction	Description
XspiCLK	Inout	XspiCLK is the serial clock used to control time to transfer data.
XspiMISO	Inout	In Master mode, this port is the input port. Input mode is used to get data from slave output port. Data are transmitted to master through this port in slave mode.
XspiMOSI	Inout	In Master mode, this port is the output port. This port is used to transfer data from master output port. Data are received from master through this port in slave mode.
XspiCS	Inout	Slave selection signal, all data TX/RX sequences are executed when XspiCS is low.

29.4 OPERATION

The SPI in S3C6410x transfers 1-bit serial data between S3C6410x and external device. The SPI in S3C6410x 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 then data are automatically sent to the register SPI_RX_DATA.

29.4.1 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 starts to be transmitted or received.

29.4.2 FIFO ACCESS

The SPI in S3C6410x supports CPU access and DMA access to FIFOs. Data size of CPU access and DMA access to FIFOs can be selected from 8-bit/16-bit/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, which is user defined. The trigger level of each FIFOs 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 only 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.

29.4.3 TRAILING BYTES IN THE RX FIFO

When the number of samples in Rx FIFO is less than the 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 can be 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.

29.4.4 PACKET NUMBER CONTROL

SPI can control the number of packets to be received in master mode. If there is any number of packets to be received, just set the SFR (Packet_Count_reg). SPI stops generating SPICLK when the number of packets is the same as what you 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.)

29.4.5 CHIP SELECT CONTROL

XspiCS# can be selected auto control or manual control.

29.4.5.1 MANUAL CONTROL MODE

AUTO_N_MANUAL must be clear (default value is 0). XspiCS# level is controlled by NSSOUT bit.

29.4.5.2 AUTO CONTROL MODE

AUTO_N_MANUAL must be set. XspiCS toggled between packet and packet automatically. XspiCS inactive period is controlled by NCS_TIME_COUNT. NSSOUT is not available at this time.

29.4.6 SPI TRANSFER FORMAT

The S3C6410X supports 4 different formats to transfer the data. Figure 29-1 describes four waveforms for SPICLK.

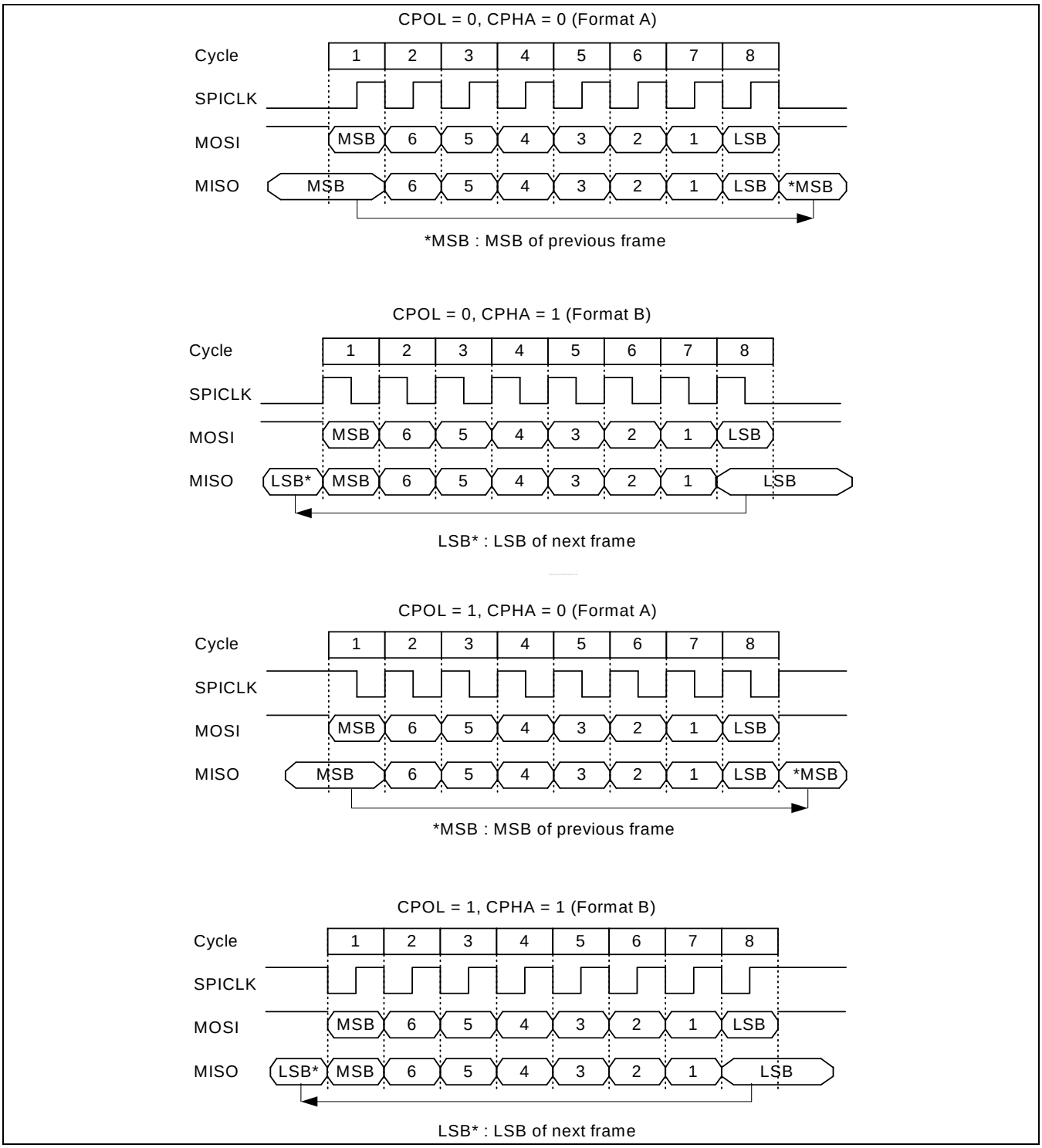


Figure 29-1. SPI Transfer Format

29.6 Special Function Register Descriptions

29.6.1 SETTING SEQUENCE OF SPECIAL FUNCTION REGISTER

Special Function Register must be set in 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.

29.6.2 SPECIAL FUNCTION REGISTER

Register	Address	R/W	Description	Reset Value
CH_CFG(Ch0)	0x7F00B000	R/W	SPI configuration register	0x40
CH_CFG(Ch1)	0x7F00C000	R/W	SPI configuration register	0x40

CH_CFG	Bit		Description	Initial State
HIGH_SPEED_EN	[6]	R/W	Slave TX output time control bit. If this bit is set in CPHA 0, Slave TX output time is reduced as much as half period of SPICLKout period. SPI clock is less than 30MHz, this bit should be clear. 0: disable 1: enable	1
SW_RST	[5]	R/W	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]	R/W	Whether SPI Channel is Master or Slave 0: Master 1: Slave	0
CPOL	[3]	R/W	Determine an active high or active low clock 0: active high 1: active low	0
CPHA	[2]	R/W	Select one of the two fundamentally different transfer format 0: format A 1: format B	0
RX_CH_ON	[1]	R/W	SPI Rx Channel On 0: Channel Off 1: Channel On	0
TX_CH_ON	[0]	R/W	SPI Tx Channel On 0: Channel Off 1: Channel On	0

Register	Address	R/W	Description	Reset Value
CLK_CFG(Ch0)	0x7F00B004	R/W	Clock configuration register	0x0
CLK_CFG(Ch1)	0x7F00C004	R/W	Clock configuration register	0x0

CLK_CFG	Bit		Description	Initial State
SPI_CLKSEL	[10:9]	R/W	Clock source selection to generate SPI clock-out 00 : PCLK 01 : USBCLK 10 : Epll clock 11 : reserved * For using USBCLK source, The USB_SIG_MASK at system controller must be set. *Epll clock is from System Controller and has 4 sources: $MOUT_{EPLL}$, $DOUT_{MPLL}$, FIN_{EPLL}, 27MHz	0
ENCLK	[8]	R/W	Clock enable/disable 0 : disable 1 : enable	0
SPI_SCALER	[7:0]	R/W	SPI clock-out division rate SPI clock-out = Clock source / (2 x (Prescaler value +1))	0

Register	Address	R/W	Description	Reset Value
MODE_CFG(Ch0)	0x7F00B008	R/W	SPI FIFO control register	0x0
MODE_CFG(Ch1)	0x7F00C008	R/W	SPI FIFO control register	0x0

MODE_CFG	Bit		Description	Initial State
CH_WIDTH	[30:29]	R/W	00 : Byte 01 : Halfword 10 : Word 11 : reserved	0
TRAILING_CNT	[28:19]	R/W	Count value from writing the last data in RX FIFO to flush trailing bytes in FIFO	0
BUS_WIDTH	[18:17]	R/W	00: Byte 01: Halfword 10 : Word 11:reserved	0
RX_RDY_LVL	[16:11]	R/W	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]	R/W	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]	-	-	-
RX_DMA_SW	[2]	R/W	Rx DMA mode enable/disable 0 : DMA mode disable 1 : DMA mode enable	0
TX_DMA_SW	[1]	R/W	Tx DMA mode on/off 0 : DMA mode disable 1 : DMA mode enable	0
DMA_TYPE	[0]	R/W	DMA transfer type, single or 4 busts. 0 : single 1 : 4 burst DMA transfer size must be set as the same size in DMA as it is in SPI.	0

Note 1. CH_WIDTH is shift-register width.

Note 2. BUS_WIDTH is SPI FIFO width, transfer data size should be aligned at BUS_WIDTH.

Note 3. CH_WIDTH must be smaller than BUS_WIDTH or the same as.

Register	Address	R/W	Description	Reset Value
CS_REG(Ch0)	0x7F00B00C	R/W	Slave selection signal control register	0x1
CS_REG(Ch1)	0x7F00C00C	R/W	Slave selection signal control register	0x1

CS_REG	Bit		Description	Initial State
NCS_TIME_COUNT	[9:4]	R/W	NSSOUT inactive time = ((nCS_time_count+3)/2) x SPICLKout	0
reserved	[3:2]	-	reserved -	-
AUTO_N_MANUAL	[1]	R/W	Chip select toggle manual or auto selection 0: manual 1: Auto	0
NSSOUT	[0]	R/W	Slave selection signal (manual only) 0: active 1: inactive	1

Note. If AUTO_N_MANUAL is set, NSSOUT is controlled by SPI controller and data transfer is performed discontinuously. Unit data size depends on CH_WIDTH.

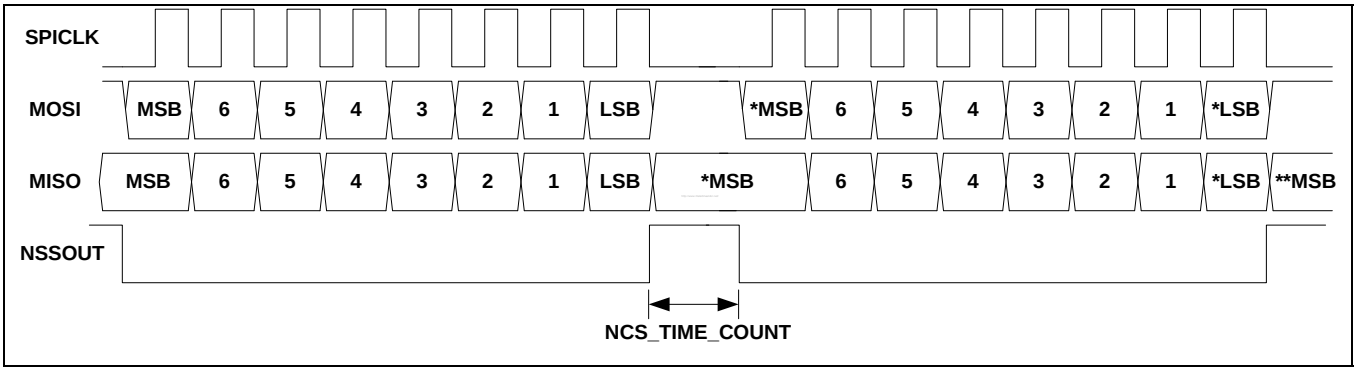


Figure 29-2. Auto chip select mode waveform (CPOL=0, CPHA=0, CH_WIDTH=Byte)

Register	Address	R/W	Description	Reset Value
SPI_INT_EN(Ch0)	0x7F00B010	R/W	SPI Interrupt Enable register	0x0
SPI_INT_EN(Ch1)	0x7F00C010	R/W	SPI Interrupt Enable register	0x0

SPI_INT_EN	Bit		Description	Initial State
INT_EN_TRAILING	[6]	R/W	Interrupt Enable for trailing count to be zero 0: Disable 1:Enable	0
INT_EN_RX_OVERRUN	[5]	R/W	Interrupt Enable for RxOverrun 0: Disable 1:Enable	0
INT_EN_RX_UNDERRUN	[4]	R/W	Interrupt Enable for RxUnderrun 0: Disable 1:Enable	0
INT_EN_TX_OVERRUN	[3]	R/W	Interrupt Enable for TxOverrun 0: Disable 1:Enable	0
INT_EN_TX_UNDERRUN	[2]	R/W	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]	R/W	Interrupt Enable for RxFifoRdy(INT mode) 0: Disable 1:Enable	0
INT_EN_TX_FIFO_RDY	[0]	R/W	Interrupt Enable for TxFifoRdy(INT mode) 0: Disable 1:Enable	0

Register	Address	R/W	Description	Reset Value
SPI_STATUS(Ch0)	0x7F00B014	R	SPI status register	0x0
SPI_STATUS(Ch1)	0x7F00C014	R	SPI status register	0x0

SPI_STATUS	Bit		Description	Initial State
TX_DONE	[21]	R	Indication of transfer done in Shift register(master mode only) 0 : all case except blow case 1 : when tx fifo and shift register are empty	0
TRAILING_BYTE	[20]	R	Indication that trailing count is zero	0
RX_FIFO_LVL	[19:13]	R	Data level in RX FIFO 0 ~ 64 byte	0
TX_FIFO_LVL	[12:6]	R	Data level in TX FIFO 0 ~ 64 byte	0
RX_OVERRUN	[5]	R	Rx Fifo overrun error 0: no error, 1: overrun error	0
RX_UNDERRUN	[4]	R	Rx Fifo underrun error 0: no error, 1: underrun error	0
TX_OVERRUN	[3]	R	Tx Fifo overrun error 0: no error, 1: overrun error	0
TX_UNDERRUN	[2]	R	Tx Fifo underrun error Tx Fifo underrun error is occurred if TX Fifo is empty in slave mode.(no emoty state in slave Tx mode) 0: no error, 1: underrun error	0
RX_FIFO_RDY	[1]	R	0 : data in FIFO less than trigger level 1 : data in FIFO more than trigger level	0
TX_FIFO_RDY	[0]	R	0 : data in FIFO more than trigger level 1 : data in FIFO less than trigger level	0

Register	Address	R/W	Description	Reset Value
SPI_TX_DATA(Ch0)	0x7F00B018	W	SPI TX DATA register	0x0
SPI_TX_DATA(Ch1)	0x7F00C018	W	SPI TX DATA register	0x0

SPI_TX_DATA	Bit		Description	Initial State
TX_DATA	[31:0]	W	This field contains the data to be transmitted over the SPI channel.	0

Register	Address	R/W	Description	Reset Value
SPI_RX_DATA(Ch0)	0x7F00B01C	R	SPI RX DATA register	0x0
SPI_RX_DATA(Ch1)	0x7F00C01C	R	SPI RX DATA register	0x0

SPI_RX_DATA	Bit		Description	Initial State
RX_DATA	[31:0]	R	This field contains the data to be received over the SPI channel.	0

Register	Address	R/W	Description	Reset Value
PACKET_CNT_REG(Ch0)	0x7F00B020	R/W	Count how many data master gets	0x0
PACKET_CNT_REG(Ch1)	0x7F00C020	R/W	Count how many data master gets	0x0

PACKET_CNT_REG	Bit		Description	Initial State
PACKET_CNT_EN	[16]	R/W	Enable bit for packet count 0: Disable 1:Enable	1’b0
COUNT_VALUE	[15:0]	R/W	Packet count value	0

Register	Address	R/W	Description	Reset Value
PENDING_CLR_REG(Ch0)	0x7F00B024	R/W	Status pending clear register	0x0
PENDING_CLR_REG(Ch1)	0x7F00C024	R/W	Status pending clear register	0x0

PENDING_CLR_REG	Bit	R/W	Description	Initial State
TX_UNDERRUN_CLR	[4]	R/W	TX underrun pending clear bit 0: non-clear 1:clear	0
TX_OVERRUN_CLR	[3]	R/W	TX overrun pending clear bit 0: non-clear 1:clear	0
RX_UNDERRUN_CLR	[2]	R/W	RX underrun pending clear bit 0: non-clear 1:clear	0
RX_OVERRUN_CLR	[1]	R/W	RX overrun pending clear bit 0: non-clear 1:clear	0
TRAILING_CLR	[0]	R/W	Trailing pending clear bit 0: non-clear 1:clear	0

Register	Address	R/W	Description	Reset Value
SWAP_CFG(Ch0)	0x7F00B028	R/W	SWAP config register	0x0
SWAP_CFG (Ch1)	0x7F00C028	R/W	SWAP config register	0x0

SWAP_CFG	Bit		Description	Initial State
RX_HWORD_SWAP	[7]	R/W	0: off 1: swap	0
RX_BYTE_SWAP	[6]	R/W	0: off 1: swap	0
RX_BIT_SWAP	[5]	R/W	0: off 1: swap	0
RX_SWAP_EN	[4]	R/W	Swap enable 0 : normal 1 : swap	0
TX_HWORD_SWAP	[3]	R/W	0: off 1: swap	0
TX_BYTE_SWAP	[2]	R/W	0: off 1: swap	0
TX_BIT_SWAP	[1]	R/W	0: off 1: swap	0
TX_SWAP_EN	[0]	R/W	Swap enable 0 : normal 1 : swap	0

** Data size must be larger than swap size.

Register	Address	R/W	Description	Reset Value
FB_CLK_SEL (Ch0)	0x7F00B02C	R/W	Feedback clock selecting register.	0x3
FB_CLK_SEL (Ch1)	0x7F00C02C	R/W	Feedback clock selecting register.	0x3

FB_CLK_SEL	Bit		Description	Initial State
FB_CLK_SEL	[1:0]	R/W	00 : 0nS additional delay 01 : 3nS additional delay 10 : 6nS additional delay 11 : 9nS additional delay * Delay based on typical condition.	0x3

29.6.3 SPECIAL FUNCTION REGISTER

PAD driving strength of SPI can be controlled by setting SPCON register in GPIO.

30 IIC-BUS INTERFACE

This chapter describes the functions and usage of IIC-BUS Interface in S3C6410X RISC microprocessor.

30.1 OVERVIEW

The S3C6410 RISC microprocessor can support 2-ch multi-master IIC-bus serial interface. A serial data line (SDA) and a serial clock line (SCL) carry information between bus masters and peripheral devices, which are connected to the IIC-bus. The SDA and SCL lines are bi-directional.

In multi-master IIC-bus mode, multiple S3C6410 RISC microprocessors can receive or transmit serial data to or from slave devices. The master S3C6410 can initiate and terminate a data transfer over the IIC-bus. The IIC-bus in the S3C6410 uses Standard bus arbitration procedure.

To control multi-master IIC-bus operations, values must be written to the following registers:

- Multi-master IIC-bus control register, IICCON
- Multi-master IIC-bus control/status register, IICSTAT
- Multi-master IIC-bus Tx/Rx data shift register, IICDS
- Multi-master IIC-bus address register, IICADD

When the IIC-bus is free, the SDA and SCL lines must be both at High level. A High-to-Low transition of SDA can initiate a Start condition. A Low-to-High transition of SDA can initiate a Stop condition while SCL remains steady at High Level.

The Start and Stop conditions can always be generated by the master devices. After the Start condition has been initiated, the master selects the slave device by writing its 7-bit address in the first outgoing data byte. The 8th bit determines the direction of the transfer (read or write).

Every data byte put onto the SDA line must be eight bits in total. There is no limit to send or receive bytes during the bus transfer operation. Data is always sent first from most-significant bit (MSB) and every byte must be immediately followed by acknowledge (ACK) bit.

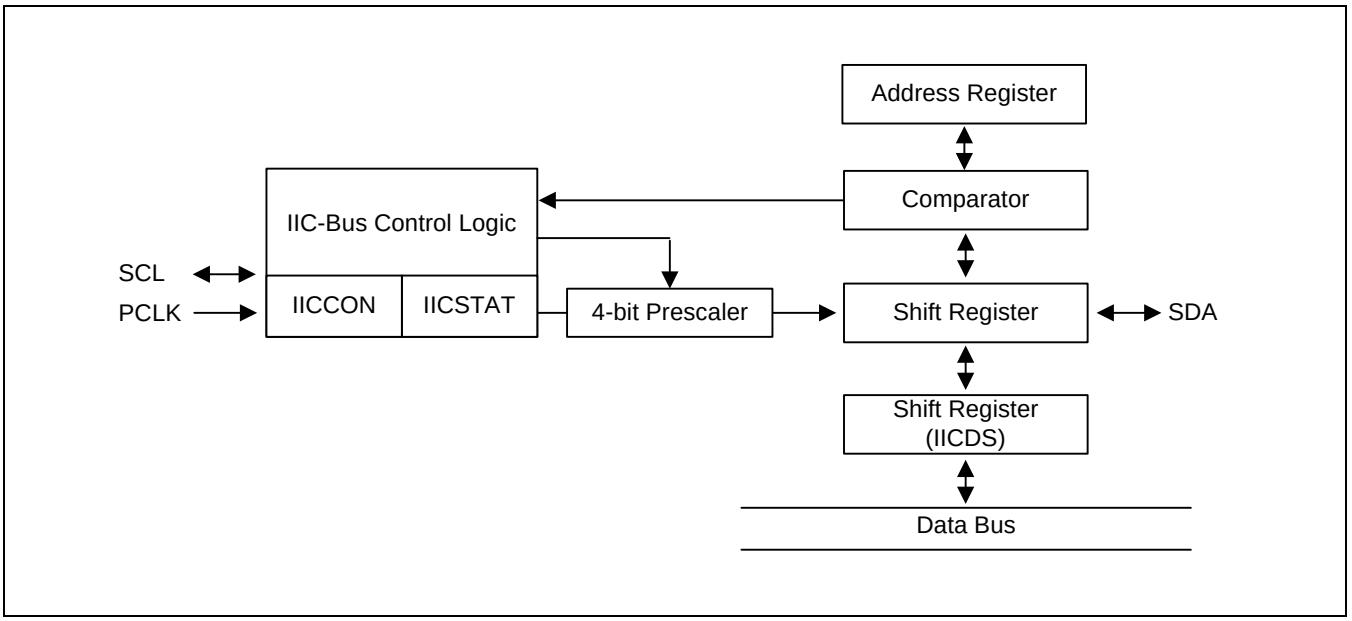


Figure 30-1. IIC-Bus Block Diagram

30.2 IIC-BUS INTERFACE

The S3C6410 IIC-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.

30.3 START AND STOP CONDITIONS

When the IIC-bus interface is inactive, it is usually in Slave mode. In other words, the interface must be in Slave mode before detecting a Start condition on the SDA line (a Start condition can be initiated with a High-to-Low transition of the SDA line while the clock signal of SCL is high). When 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 can transfer a one-byte serial data over the SDA line and a Stop condition can terminate the data transfer. A Stop condition is a Low-to-High transition of the SDA line while SCL is High. Start and Stop conditions are always generated by the master. The IIC-bus gets busy when a Start condition is generated. A Stop condition will make the IIC-bus free.

When a master initiates a Start condition, it will 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 will complete the transfer operation by transmitting a Stop condition. If the master wants to continue the data transmission to the bus, it will generate another Start condition and a slave address. In this way, the read-write operation can be performed in various formats.

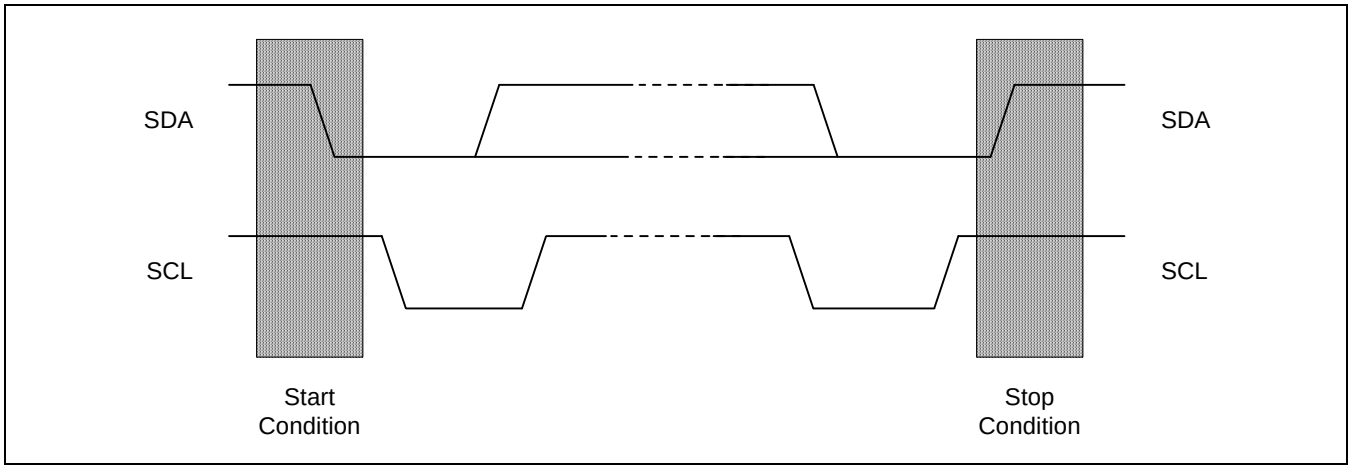


Figure 30-2. Start and Stop Condition Block Diagram

30.4 DATA TRANSFER FORMAT

Every byte placed on the SDA line must be eight bits in length. The bytes can be unlimitedly transmitted per transfer. The first byte following a Start condition will have the address field. The address field can be transmitted by the master when the IIC-bus is operating in Master mode. Each byte is followed by an acknowledgement (ACK) bit. The MSB bit of the serial data and addresses are always sent first.

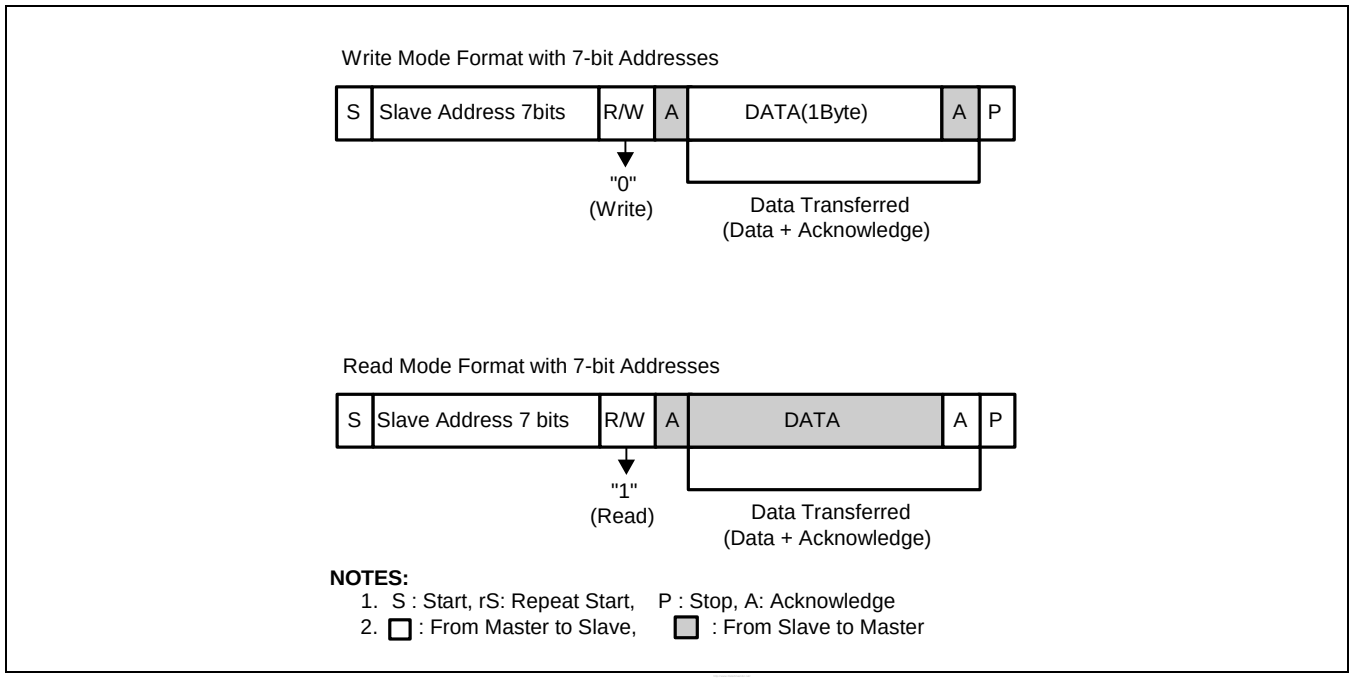


Figure 30-3. Data Transfer on the IIC-Bus Block Diagram

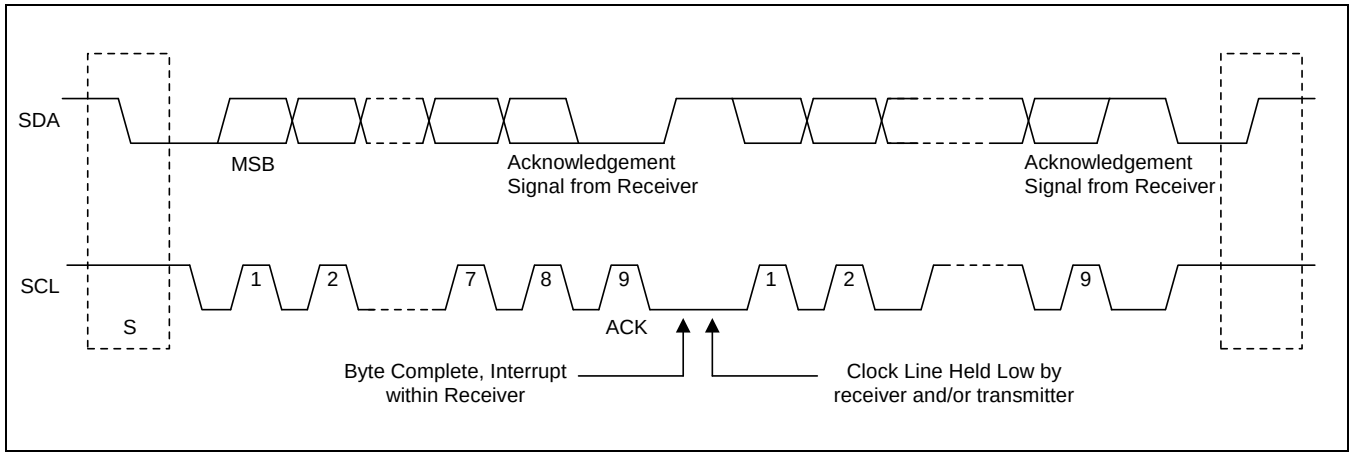


Figure 30-4. Data Transfer on the IIC-Bus Block Diagram

30.5 ACK SIGNAL TRANSMISSION

To complete a one-byte transfer operation, the receiver must send 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 will generate the clock pulse required to transmit the ACK bit.

The transmitter releases the SDA line by making the SDA line High when 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 ACK bit transmit function can be enabled or disabled by software (IICSTAT). However, the ACK pulse on the ninth clock of SCL is required to complete the one-byte data transfer operation.

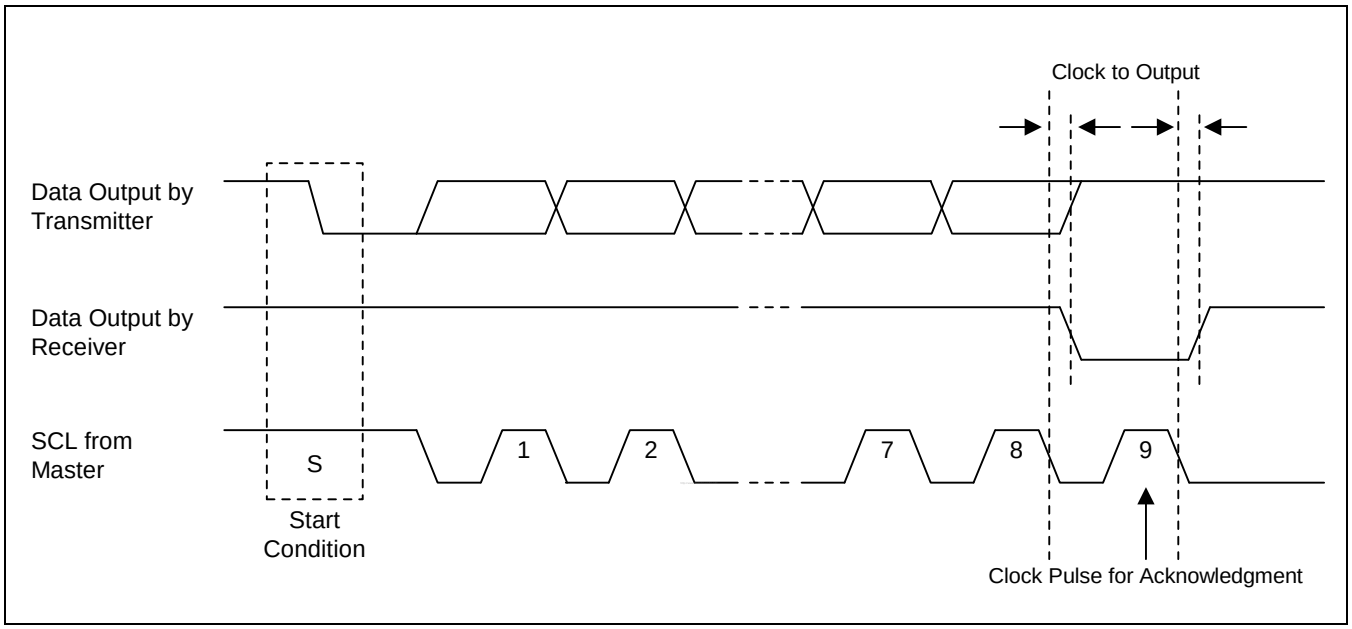


Figure 30-5. Acknowledge on the IIC-Bus Block Diagram

30.6 READ-WRITE OPERATION

When the data is transferred in the Transmitter mode, the IIC-bus interface will wait until IIC-bus Data Shift (IICDS) register receives a new data. Before the new data is written into the register, the SCL line will be held low and then released after it is written. The S3C6410 holds the interrupt to identify the completion of current data transfer. After the CPU receives the interrupt request, it writes a new data again into the IICDS register.

When data is received in the Receive mode, the IIC-bus interface will wait until IICDS register is read. Before the new data is read out, the SCL line will be held low and then released after it is read. The S3C6410 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 IICDS register.

30.7 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 will not initiate a data transfer because the current level on the bus does not correspond to its own. The arbitration procedure will be extended until the SDA line turns High.

However, when 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 must detect the address bits. While each master generates the slave address, it also 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 will detect Low on the bus because the Low status is superior to the High status in power. When this happens, Low (as the first bit of address) generating master will get the mastership while High (as the first bit of address) generating master will withdraw the mastership. If both masters generate Low as the first bit of address, there will be arbitration again for the second address bit. This arbitration will continue to the end of last address bit. Assume that both masters are each trying to address the same device, there will be arbitration again for the data-bits.

30.8 ABORT CONDITIONS

If a slave receiver cannot acknowledge the confirmation of the slave address, it will hold the level of the SDA line High. In this case, the master generates a Stop condition 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.

30.9 CONFIGURING IIC-BUS

To control the frequency of the serial clock (SCL), the 4-bit prescaler value can be programmed in the IICCON register. The IIC-bus interface address is stored in the IIC-bus address (IICADD) register. (By default, the IIC-bus interface address has an unknown value.)

30.10 FLOWCHARTS OF OPERATIONS IN EACH MODE

The following steps must be executed before any IIC Tx/Rx operations.

1. Write own slave address on IICADD register, if needed.
2. Set IICCON register.
   a) Enable interrupts
   b) Define SCL period
3. Set IICSTAT to enable Serial Output

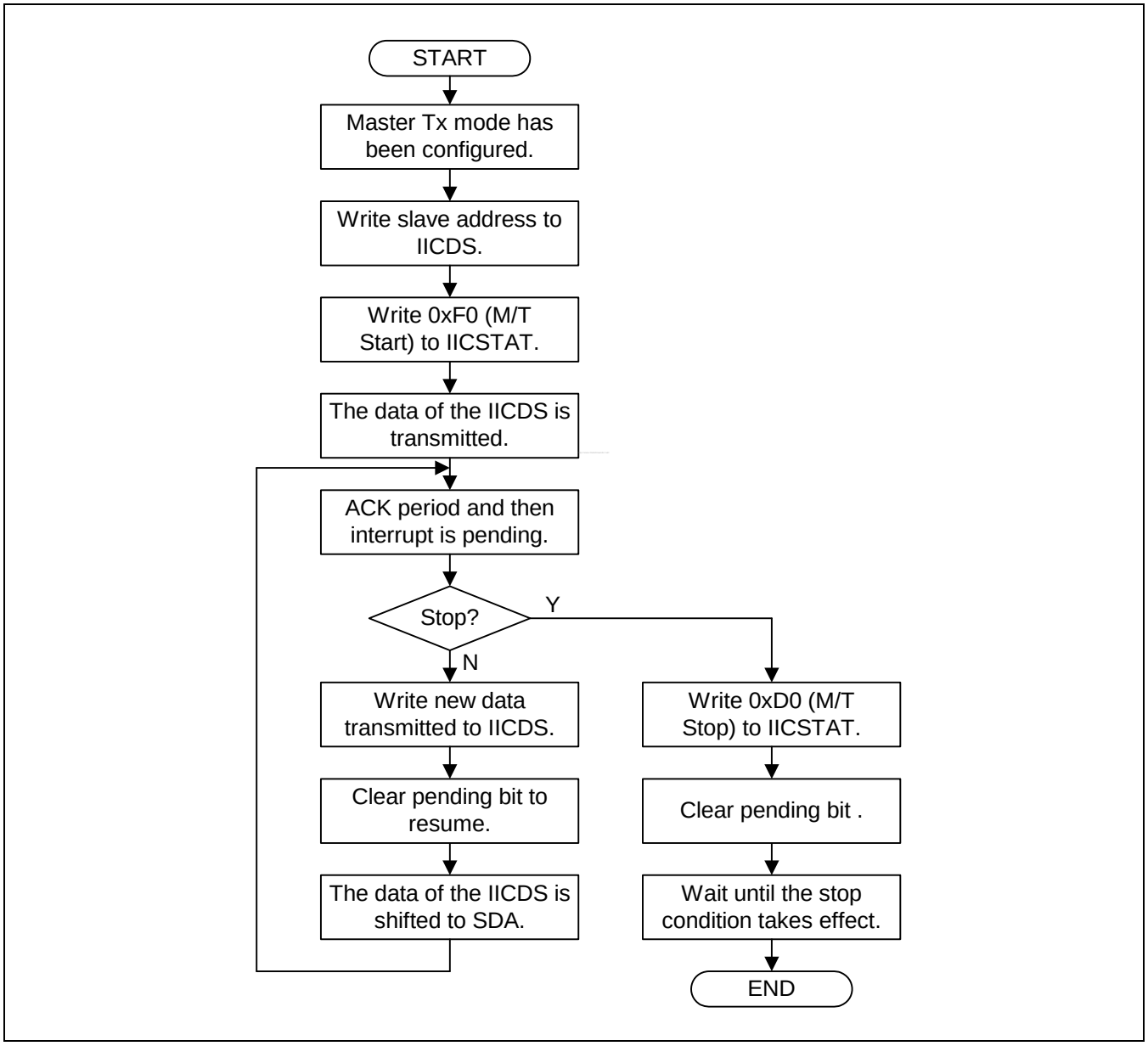


Figure 30-6. Operations for Master/Transmitter Mode

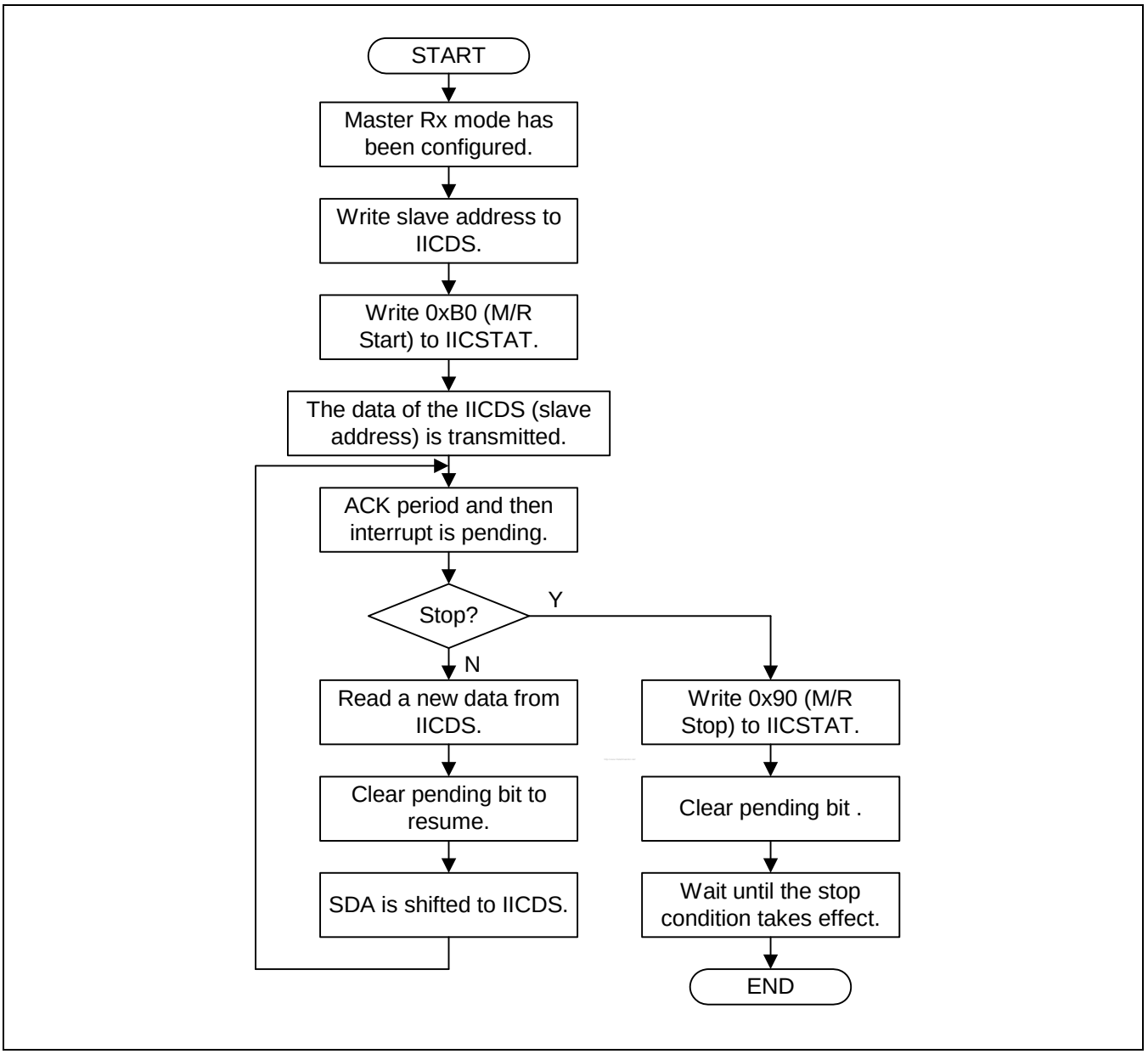


Figure 30-7. Operations for Master/Receiver Mode

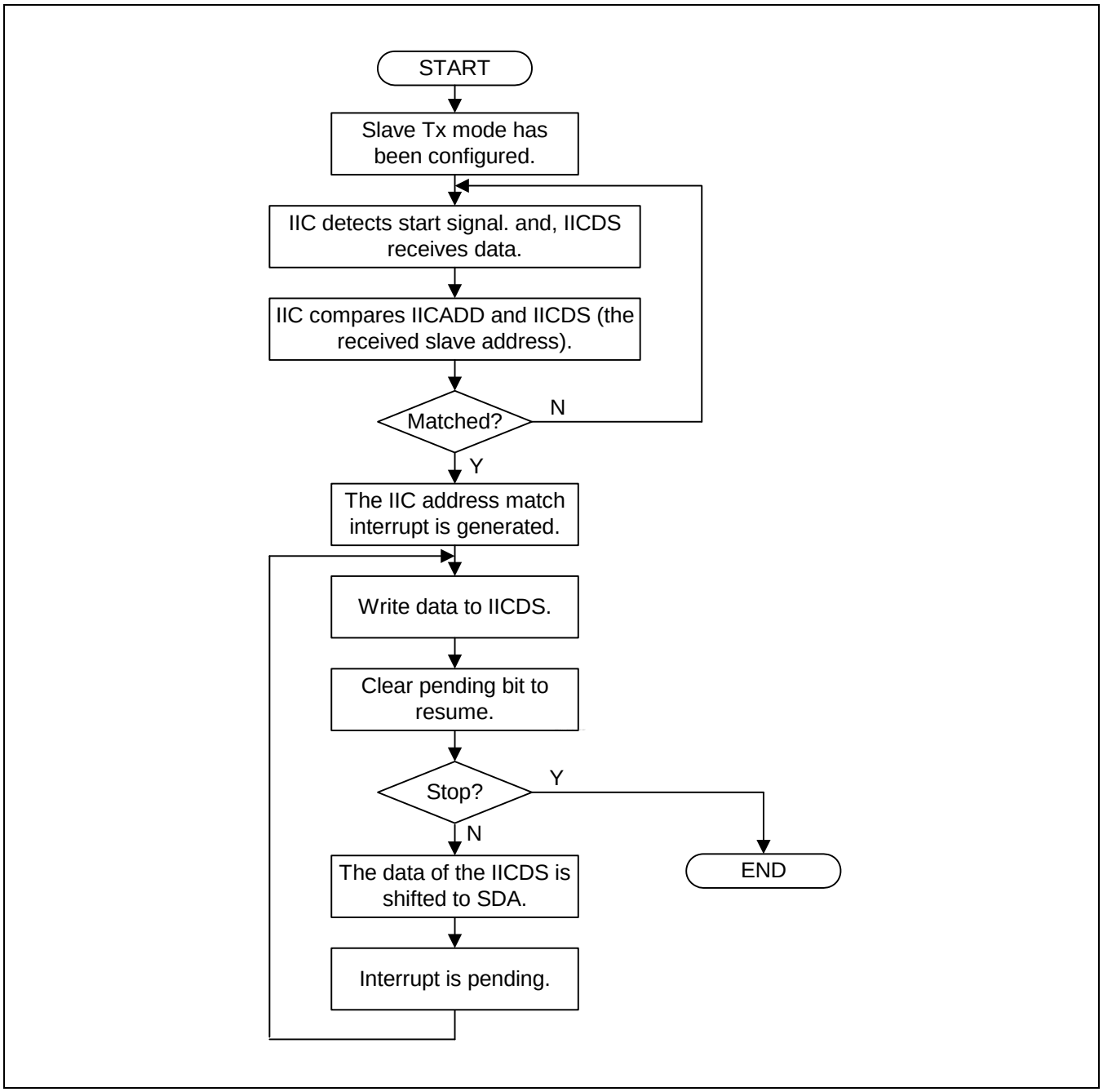


Figure 30-8. Operations for Slave/Transmitter Mode

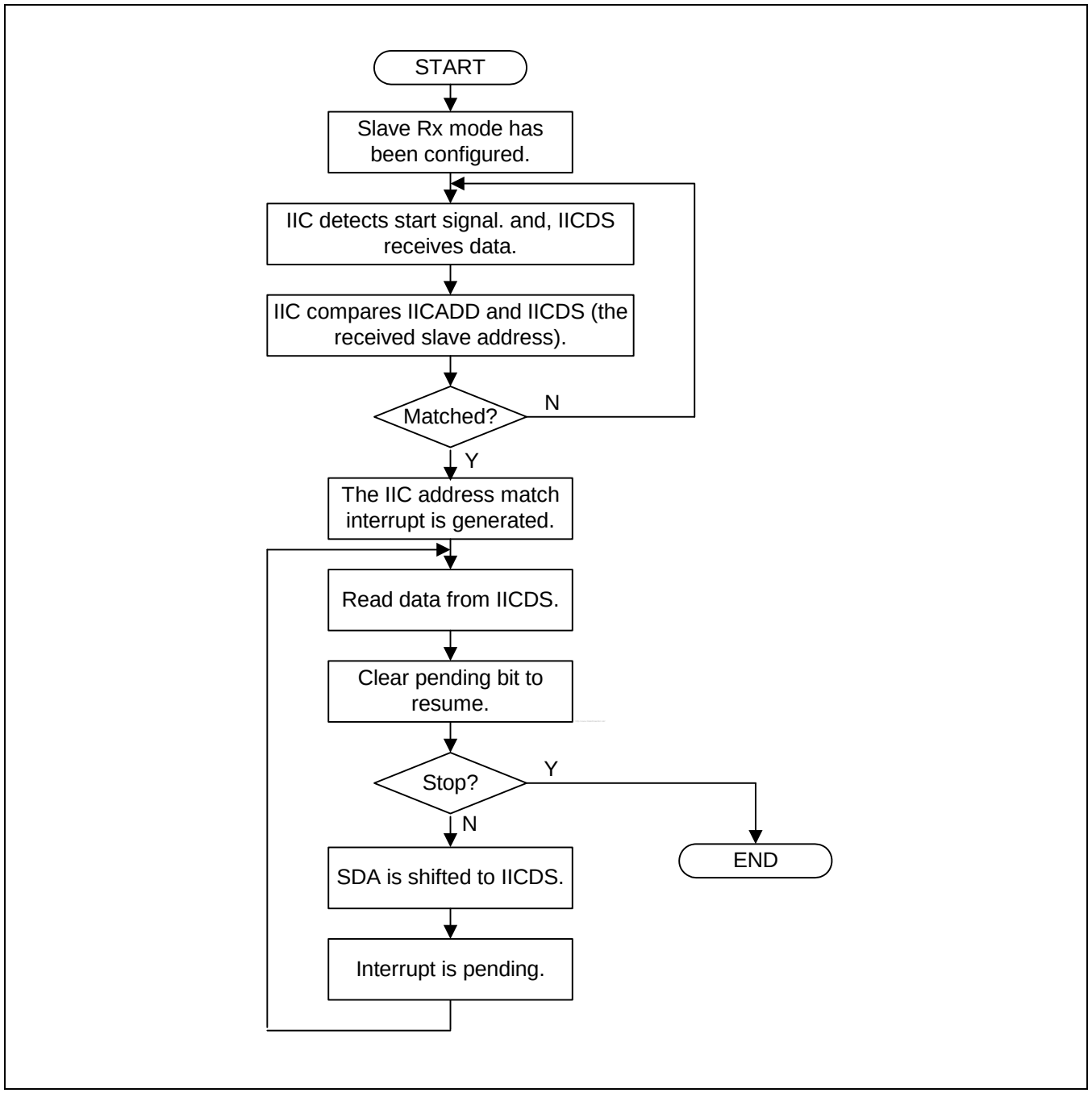


Figure 30-9. Operations for Slave/Receiver Mode

30.11 IIC-BUS INTERFACE SPECIAL REGISTERS

30.11.1 MULTI-MASTER IIC-BUS CONTROL (IICCON) REGISTER

Register	Address	R/W	Description	Reset Value
IICCON	0x7F004000	R/W	IIC Channel 0 Bus control register	0x0X
	0x7F00F000	R/W	IIC Channel 1 Bus control register	0x0X

IICCON	Bit	Description	Initial State
Acknowledge generation (1)	[7]	IIC-bus acknowledge (ACK) enable bit. 0: Disable 1: Enable In Tx mode, the IICSDA is free in the ACK time. In Rx mode, the IICSDA is L in the ACK time.	0
Tx clock source selection	[6]	Source clock of IIC-bus transmit clock prescaler selection bit. 0: IICCLK = PCLK /16 1: IICCLK = PCLK /512	0
Tx/Rx Interrupt (5)	[5]	IIC-Bus Tx/Rx interrupt enable/disable bit. 0: Disable, 1: Enable	0
Interrupt pending flag (2) (3)	[4]	IIC-bus Tx/Rx interrupt pending flag. This bit cannot be written to 1. When this bit is read as 1, the IICSCL is tied to L and the IIC is stopped. To resume the operation, clear this bit as 0. 0: 1) No interrupt pending (when read). 2) Clear pending condition & Resume the operation (when write). 1: 1) Interrupt is pending (when read) 2) N/A (when write)	0
Transmit clock value (4)	[3:0]	IIC-Bus transmit clock prescaler. IIC-Bus transmit clock frequency is determined by this 4-bit prescaler value, according to the following formula: Tx clock = IICCLK/(IICCON[3:0]+1).	Undefined

NOTES:

1. Interfacing with EEPROM, the acknowledge (ACK) generation may be disabled before reading the last data in order to generate the STOP condition in Rx mode.
2. An IIC-bus interrupt occurs 1) when a 1-byte transmits or a receive operation is completed, 2) when 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, IICDS has to be written before clearing the IIC interrupt pending bit.
4. IICCLK is determined by IICCON [6].
   Tx clock can vary by SCL transition time.
   When IICCON[6]=0, IICCON[3:0]=0x0 or 0x1 is not available.
5. If the IICCON[5]=0, IICCON[4] does not operate correctly.
   So, It is recommended that you set IICCON[5]=1, although you does not use the IIC interrupt.

30.11.2 MULTI-MASTER IIC-BUS CONTROL/STATUS (IICSTAT) REGISTER

Register	Address	R/W	Description	Reset Value
IICSTAT	0x7F004004	R/W	Channel 0 control/status register	0x00
	0x7F00F004	R/W	Channel 1 control/status register	0x00

IICSTAT	Bit	Description	Initial State
Mode selection	[7:6]	IIC-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]	IIC-Bus busy signal status bit. 0: Read) Not busy (when read) Write) STOP signal generation 1: Read) Busy (when read) Write) START signal generation. The data in IICDS will be transferred automatically just after the start signal.	0
Serial output	[4]	IIC-bus data output enable/disable bit. 0: Disable Rx/Tx, 1: Enable Rx/Tx	0
Arbitration status flag (Read Only)	[3]	IIC-bus arbitration procedure status flag bit. 0: Bus arbitration successful 1: Bus arbitration failed during serial I/O	0
Address-as-slave status flag (Read Only)	[2]	IIC-bus address-as-slave status flag bit. 0: Cleared after reading of IICSTAT register 1: Received slave address matches the address value in the IICADD	0
Address zero status flag (Read Only)	[1]	IIC-bus addresses zero status flag bit. 0: Cleared when START/STOP condition was detected 1: Received slave address is 00000000b.	0
Last-received bit status flag (Read Only)	[0]	IIC-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

30.11.3 MULTI-MASTER IIC-BUS ADDRESS (IICADD) REGISTER

Register	Address	R/W	Description	Reset Value
IICADD	0x7F004008	R/W	Channel 0 address register	Undefined
	0x7F00F008	R/W	Channel 1 address register	Undefined

IICADD	Bit	Description	Initial State
Slave address	[7:0]	7-bit slave address, latched from the IIC-bus. When IICSTAT [4] = 0 (Serial Output Disable), IICADD is write-enabled. The IICADD value can be read any time, regardless of the current serial output enable bit (IICSTAT) setting. Slave address : [7:1] Not mapped : [0]	Undefined

30.11.4 MULTI-MASTER IIC-BUS TRANSMIT/RECEIVE DATA SHIFT (IICDS) REGISTER

Register	Address	R/W	Description	Reset Value
IICDS	0x7F00400C	R/W	Channel 0 transmit/receive data shift register	Undefined
	0x7F00F00C	R/W	Channel 1 transmit/receive data shift register	Undefined

IICDS	Bit	Description	Initial State
Data shift	[7:0]	8-bit data shift register for IIC-bus Tx/Rx operation. When IICSTAT [4] = 1 (Serial Output Enable), IICDS is write-enabled. The IICDS value can be read any time, regardless of the current serial output enable bit (IICSTAT) setting.	Undefined

30.11.5 MULTI-MASTER IIC-BUS LINE CONTROL (IICLC) REGISTER

Register	Address	R/W	Description	Reset Value
IICLC	0x7F004010	R/W	Channel 0 multi-master line control register	0x00
	0x7F00F010	R/W	Channel 1 multi-master line control register	0x00

IICLC	Bit	Description	Initial State
Filter enable	[2]	IIC-bus filter enable bit. When SDA port is operating as input, this bit should be set 1. This filter can prevent from occurred error by a glitch during double of PCLK time. 0: Filter disable 1: Filter enable	0
SDA output delay	[1:0]	IIC-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

31 UART

This chapter describes the universal asynchronous receiver/transmitter (UART) serial ports included in the S3C6410X RISC microprocessor.

31.1 OVERVIEW

The S3C6410X UART provide four independent asynchronous serial I/O (SIO) ports. Each of asynchronous serial I/O (SIO) ports can operate in interrupt-based or DMA-based mode. In other words, the UART can generate an interrupt or a DMA request to transfer data between MEM and the UART. The UART can support bit rates of maximum 3Mbps. Each UART channel contains two 64-byte FIFO's for both reception and transmission.

The S3C6410X UART includes programmable baud rates, infra-red (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 31-1. The baud-rate generator can be clocked by PCLK, EXT_UCLK0 or EXT_UCLK1. The transmitter and the receiver contain 64-byte FIFOs and data shifters. Data is written to FIFO and then copied to the transmit shifter before being transmitted. The data is then shifted out by the transmit data pin (TxDn). Meanwhile, received data is shifted from the receive data pin (RxDn), and then copied to FIFO from the shifter.

31.2 FEATURES

The UART features include:

- 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 and 1 with nRTS0, nCTS0, nRTS1, and nCTS1
- Supports high-speed operation in UART
- Supports handshake transmit/receive

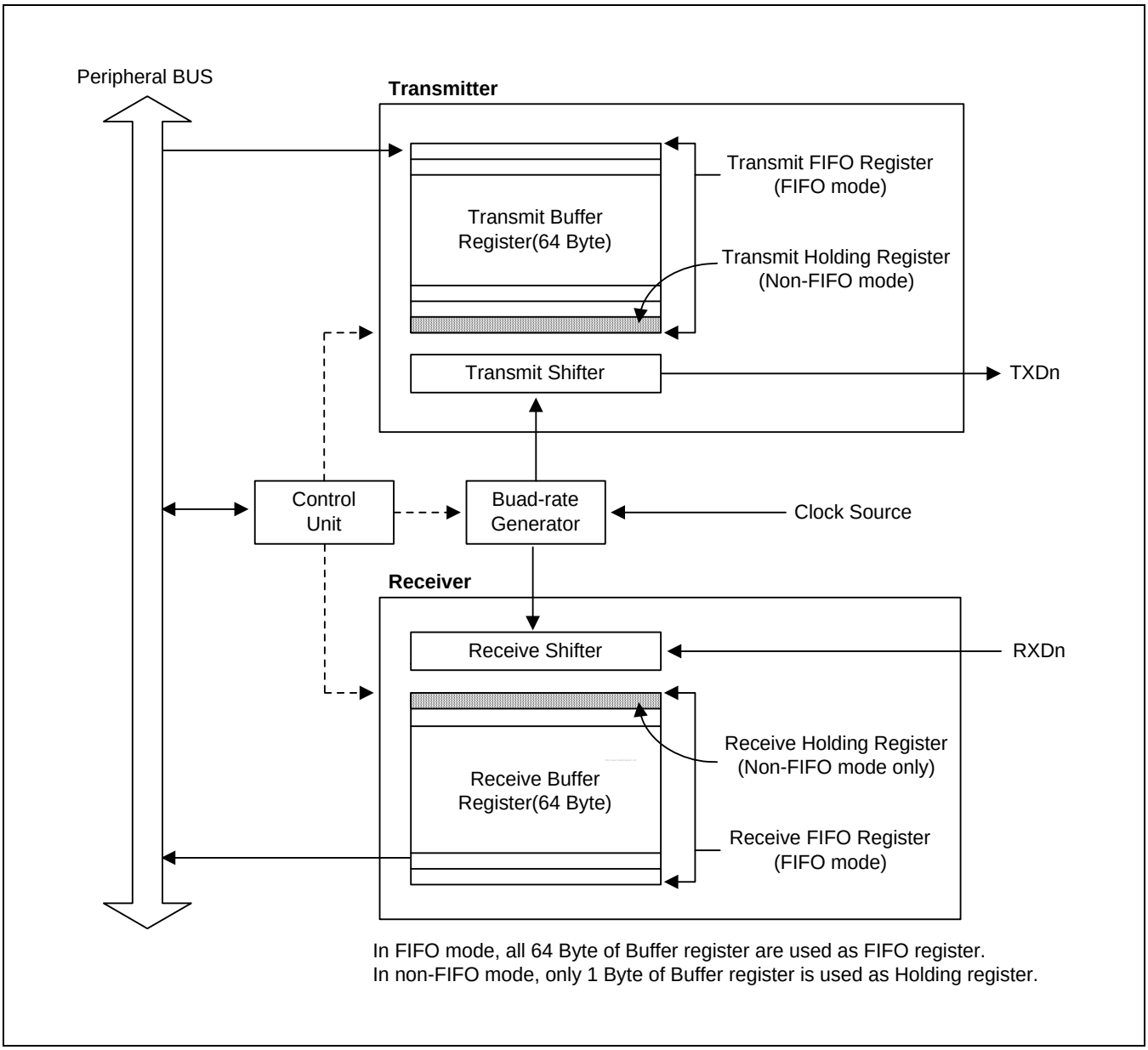


Figure 31-1. UART Block Diagram

31.3 DESCRIPTION

The following sections describe the UART operations that include data transmission, data reception, interrupt generation, baud-rate generation, Loopback mode, Infra-red mode, and auto flow control.

31.3.1 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 can be 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 the case of Non-FIFO mode).

31.3.2 DATA RECEPTION

Like the 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 can detect overrun error, parity error, frame error and break condition, each of which can set an error flag.

- The overrun error indicates that new data has overwritten the old data before the old data has been read.
- The parity error indicates that the receiver has detected an unexpected parity condition.
- The frame error indicates that the received data does not have a valid stop bit.
- The break condition indicates that the RxDn input is held in the logic 0 state for a duration longer than one frame transmission time.

Receive time-out condition occurs when it does not receive any data 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.

31.3.3 AUTO FLOW CONTROL(AFC)

The S3C6410X UART 0 and UART 1 support auto flow control with nRTS and nCTS signals. In case, it can be connected to external UARTs. If you want to connect a UART to a Modem, you must 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 only when nCTS signals are activated (in AFC, nCTS means that other UART's FIFO is ready to receive data). Before the UART receives data, nRTS has to be activated when its receive FIFO has a spare of more than 2-byte and has to be inactivated when its receive FIFO has a spare under 1-byte (in AFC, nRTS means that its own receive FIFO is ready to receive data).

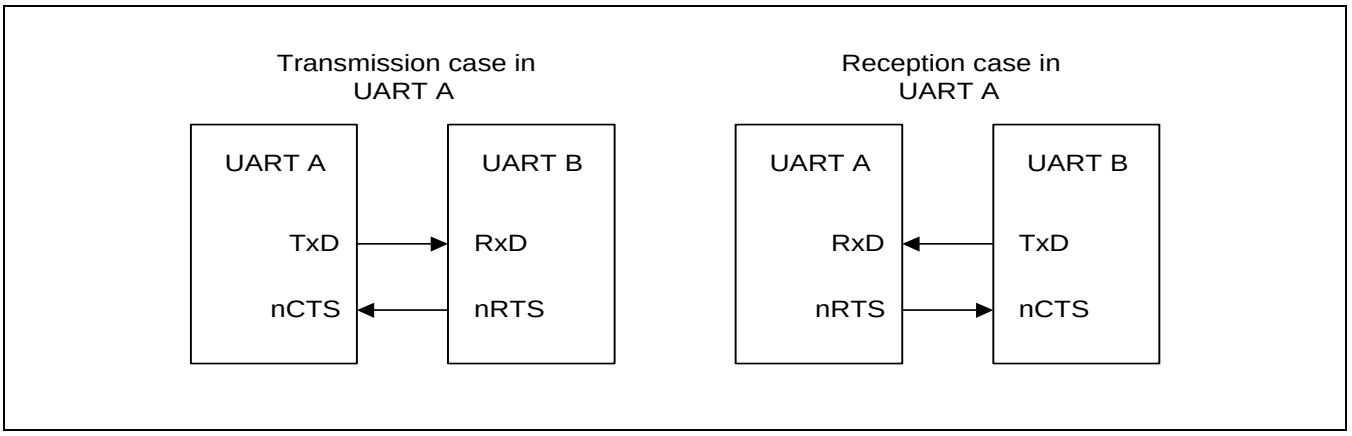


Figure 31-2. UART AFC interface

31.3.3.1 Example of Non Auto-Flow control (controlling nRTS and nCTS by software)

— Rx operation with FIFO

1. Select receive mode (Interrupt or DMA mode).
2. Select Rx FIFO trigger level to 16

   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' (activating nRTS), and if it is equal or larger than 16 you must set the value to '0' (inactivating nRTS).
3. Repeat the Step 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' (activating nCTS), you must write the data to Tx FIFO register.
3. Repeat the Step 2.

31.3.4 RS-232C INTERFACE

To connect the UART to modem interface (instead of null modem), nRTS, nCTS, nDSR, nDTR, DCD and nRI signals are needed. But, UART of S3C6410 supports nRTS and nCTS. 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.

31.3.5 INTERRUPT/DMA REQUEST GENERATION

Each UART of the S3C6410X has seven status (Tx/Rx/Error) signals: Overrun error, Parity error, Frame error, Break, Receive buffer data ready, Transmit buffer empty, and Transmit shifter empty. All seven status signals 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. Each of which can cause the receive error status interrupt request, if the receive-error-status-interrupt-enable bit is set to one in the control register, UCONn. When a receive-error-status-interrupt-request is detected, the signal causing the request can be identified by reading the value of UERSTSTn.

When 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. Rx interrupt is generated if Receive mode in control register (UCONn) is selected as 1 (Interrupt request or polling mode).

In the Non-FIFO mode, transferring the data of the receive shifter to the receive holding register will cause Rx interrupt under the Interrupt request and polling mode.

When 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. Tx interrupt is generated if Transmit mode in control register is selected as Interrupt request or polling mode.
In the Non-FIFO mode, transferring data from the transmit holding register to the transmit shifter will cause Tx interrupt under the Interrupt request and polling mode.

Note that the Tx interrupt is always requested whenever 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 S3C6410X are level-triggered type. You must set the interrupt type as ‘Level’ whenever you program the UART control registers.

If the Receive mode and Transmit mode in control register are selected as the DMAn request mode, then DMAn request occurs instead of Rx or Tx interrupt in the situation mentioned above.

Table 31-1. Interrupts in Connection with FIFO

Type	FIFO Mode	Non-FIFO Mode
Rx interrupt	Generated whenever receive data reaches the trigger level of receive FIFO. Generated when 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 the receive holding register whenever receive buffer becomes full.
Tx interrupt	Generated whenever transmit data reaches the trigger level of transmit FIFO (Tx FIFO trigger Level).	Generated by the transmit holding register whenever transmit buffer is empty.
Error interrupt	Generated when frame error, parity error, or break signal are detected. Generated when 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.

31.3.6 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. The error interrupt will be issued only when 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 will not generate any error interrupt because the character, which was received with an error, has not been read yet. The error interrupt will occur when the character is read out.

Figure 31-3 shows the UART receiving the five characters including the two errors.

Time	Sequence Flow	Error Interrupt	Note
#0	When no character is read out	-	
#1	A, B, C, D, and E is received	-	
#2	After A is read out	The 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	The 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	-	

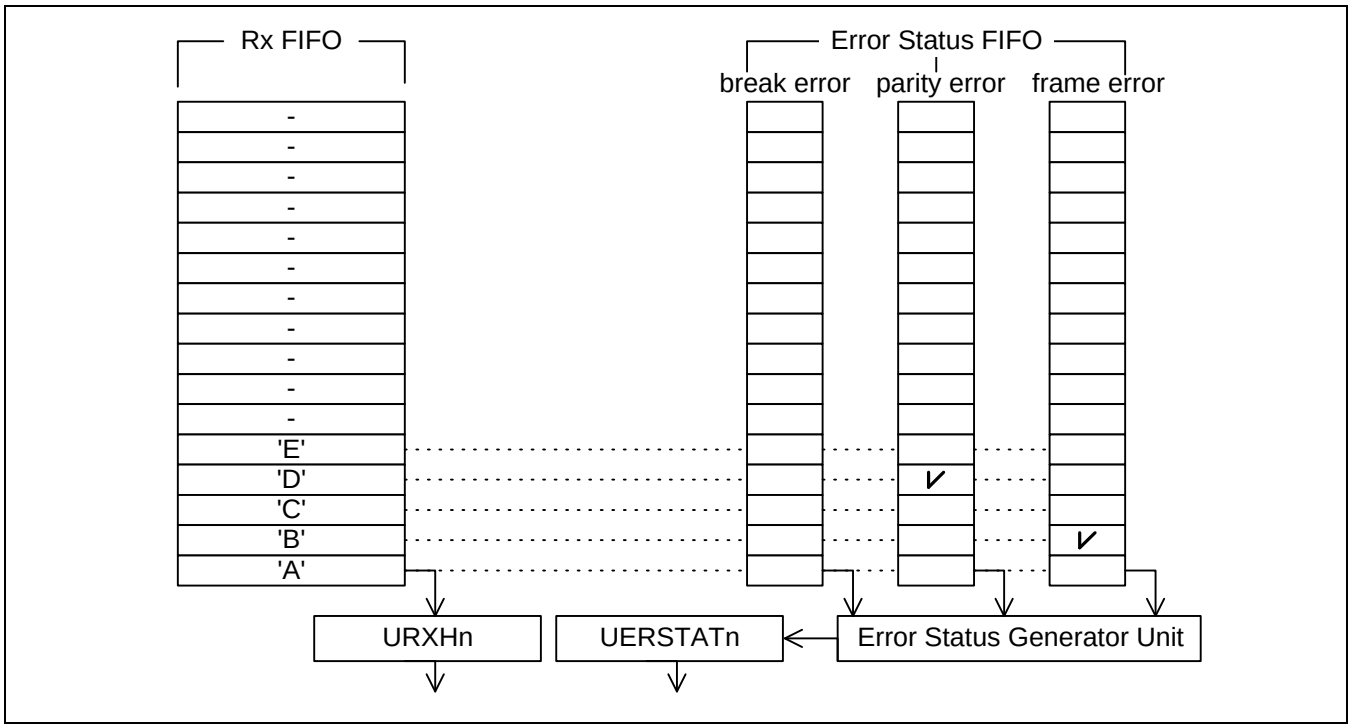


Figure 31-3. Example showing UART Receiving 5 Characters with 2 Errors

31.3.7 INFRA-RED (IR) MODE

The S3C6410X UART block supports infra-red (IR) transmission and reception, which can be selected by setting the Infra-red-mode bit in the UART line control register (ULCONn). Figure 31-4 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 (when the transmit data bit is zero); In IR receive mode, the receiver must detect the 3/16 pulsed period to recognize a zero value (see the frame timing diagrams shown in Figure 31-6 and Figure 31-7).

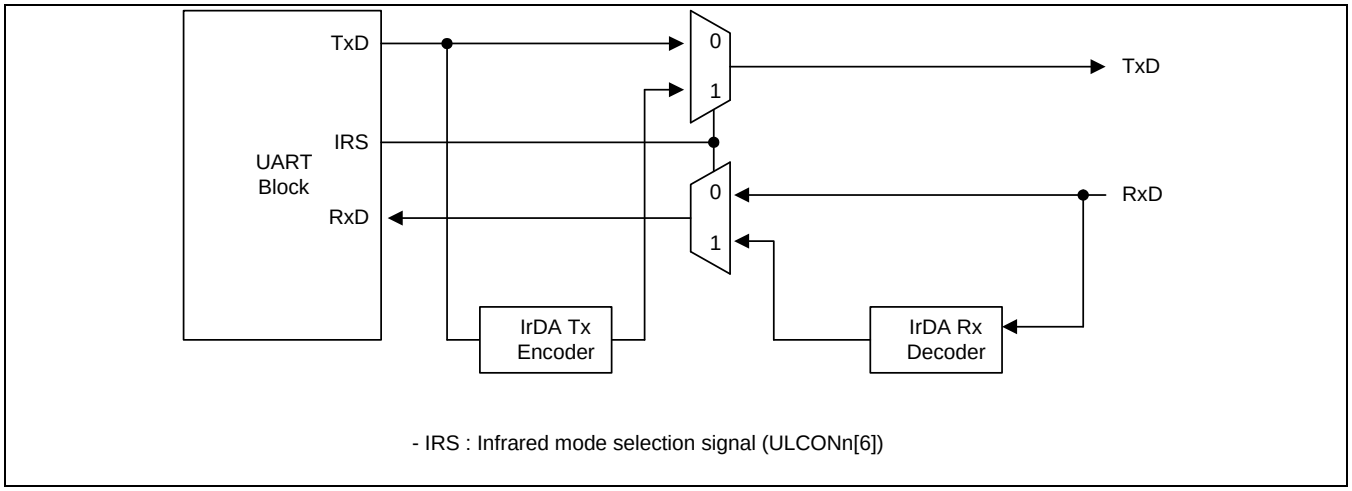


Figure 31-4. IrDA Function Block Diagram

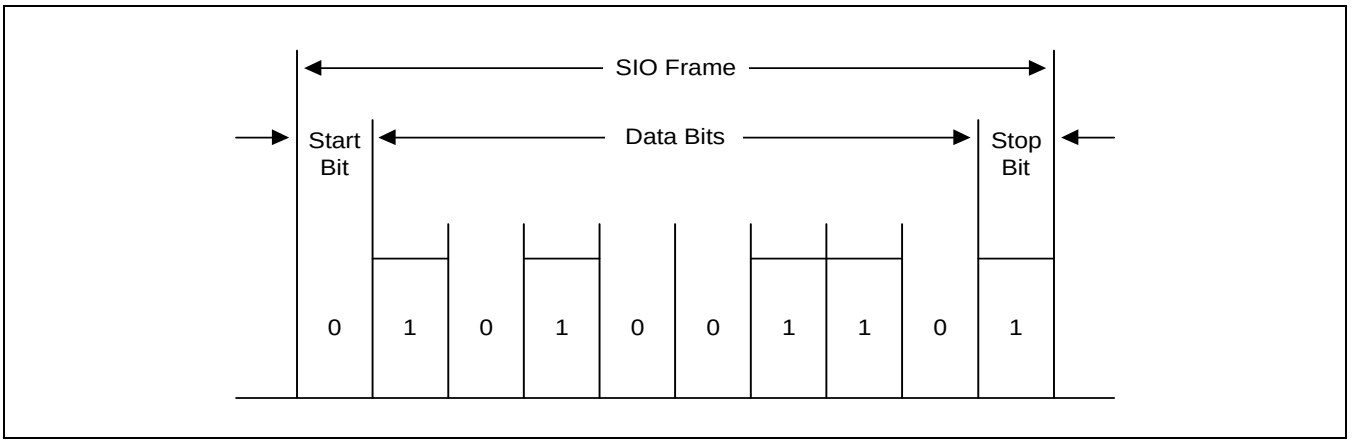


Figure 31-5. Serial I/O Frame Timing Diagram (Normal UART)

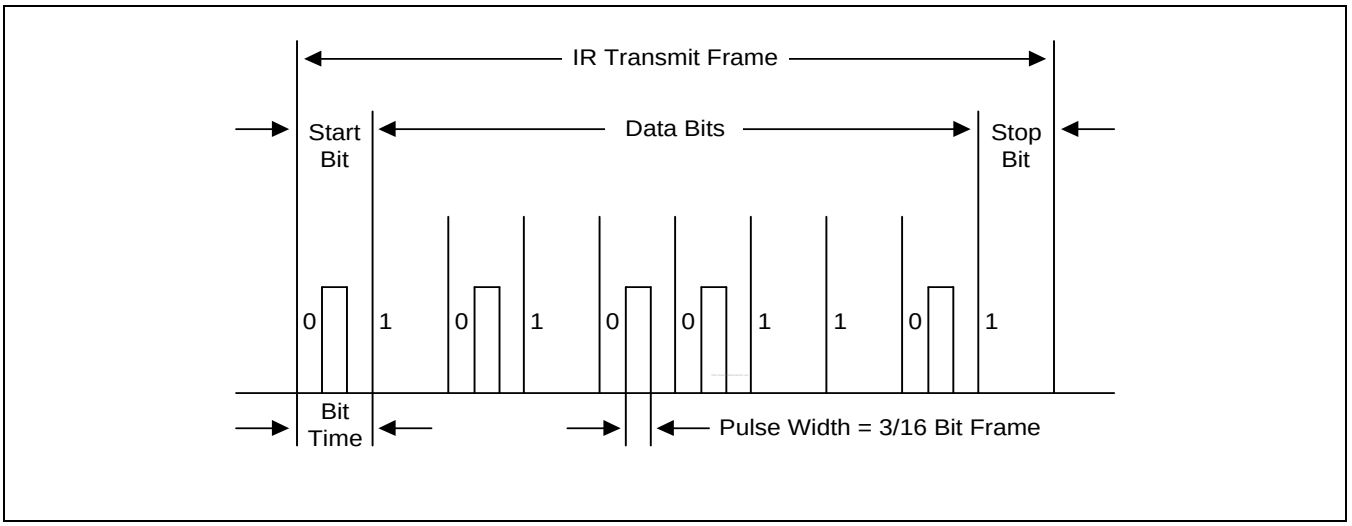


Figure 31-6. Infra-Red Transmit Mode Frame Timing Diagram

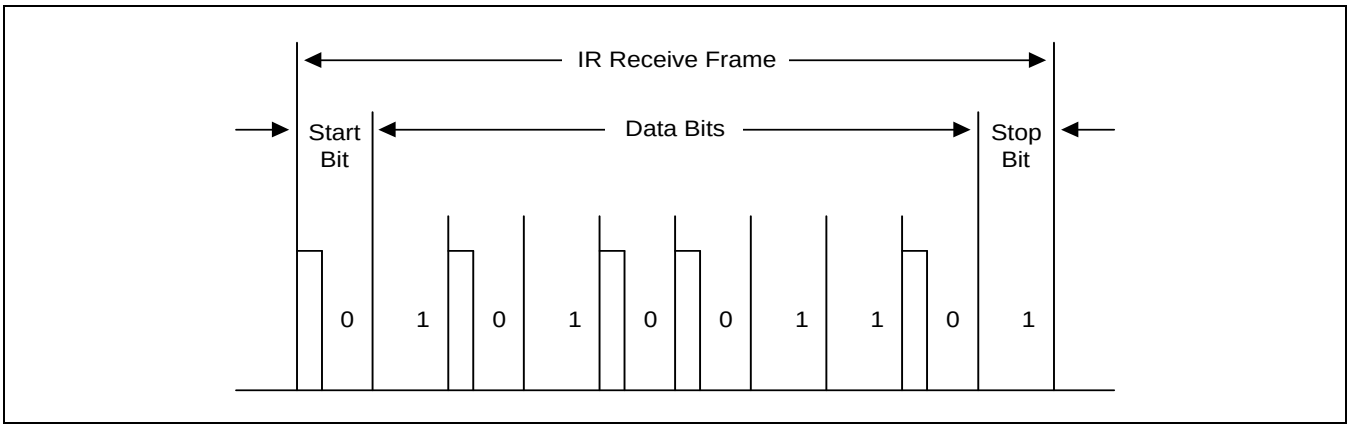


Figure 31-7. Infra-Red Receive Mode Frame Timing Diagram

31.4 EXTERNAL INTERFACES

Name	Type	Description
XuRXD[0]	Input	Receive data for UART0
XuTXD[0]	Output	Transmit data for UART0
XuCTSn[0]	Input	Clear to Send(active low) for UART0
XuRTSn[0]	Output	Request to Send(active low) for UART0
XuRXD[1]	Input	Receive data for UART1
XuTXD[1]	Output	Transmit data for UART1
XuCTSn[1]	Input	Clear to Send(active low) for UART1
XuRTSn[1]	Output	Request to Send(active low) for UART1
XuRXD[2]	Input	Receive data for UART2
XuTXD[2]	Output	Transmit data for UART2
XuRXD[3]	Input	Receive data for UART3
XuTXD[3]	Output	Transmit data for UART3

NOTE: UART external pin are shared with other Controllers like CFCON, IrDA and etc. In order to use these pads for UART, GPIO must be set before the UART started. For more information refer to the GPIO chapter of this manual for proper GPIO settings.

31.5 REGISTER DESCRIPTION

31.5.1 MEMORY MAP

Register	Address	R/W	Description	Reset Value
ULCON0	0x7F005000	R/W	UART channel 0 line control register	0x00
UCON0	0x7F005004	R/W	UART channel 0 control register	0x00
UFCON0	0x7F005008	R/W	UART channel 0 FIFO control register	0x0
UMCON0	0x7F00500C	R/W	UART channel 0 Modem control register	0x0
UTRSTAT0	0x7F005010	R	UART channel 0 Tx/Rx status register	0x6
UERSTAT0	0x7F005014	R	UART channel 0 Rx error status register	0x0
UFSTAT0	0x7F005018	R	UART channel 0 FIFO status register	0x00
UMSTAT0	0x7F00501C	R	UART channel 0 Modem status register	0x0
UTXH0	0x7F005020	W	UART channel 0 transmit buffer register	-
URXH0	0x7F005024	R	UART channel 0 receive buffer register	0x00
UBRDIV0	0x7F005028	R/W	UART channel 0 Baud rate divisior register	0x0000
UDIVSLOT0	0x7F00502C	R/W	UART channel 0 Dividing slot register	0x0000
UINTP0	0x7F005030	R/W	UART channel 0 Interrupt Pending Register	0x0
UINTSP0	0x7F005034	R/W	UART channel 0 Interrupt Source Pending Register	0x0
UINTM0	0x7F005038	R/W	UART channel 0 Interrupt Mask Register	0x0
ULCON1	0x7F005400	R/W	UART channel 1 line control register	0x00
UCON1	0x7F005404	R/W	UART channel 1 control register	0x00
UFCON1	0x7F005408	R/W	UART channel 1 FIFO control register	0x0
UMCON1	0x7F00540C	R/W	UART channel 1 Modem control register	0x0
UTRSTAT1	0x7F005410	R	UART channel 1 Tx/Rx status register	0x6
UERSTAT1	0x7F005414	R	UART channel 1 Rx error status register	0x0
UFSTAT1	0x7F005418	R	UART channel 1 FIFO status register	0x00
UMSTAT1	0x7F00541C	R	UART channel 1 Modem status register	0x0
UTXH1	0x7F005420	W	UART channel 1 transmit buffer register	-
URXH1	0x7F005424	R	UART channel 1 receive buffer register	0x00
UBRDIV1	0x7F005428	R/W	UART channel 1 Baud rate divisior register	0x0000
UDIVSLOT1	0x7F00542C	R/W	UART channel 1 Dividing slot register	0x0000
UINTP1	0x7F005430	R/W	UART channel 1 Interrupt Pending Register	0x0
UINTSP1	0x7F005434	R/W	UART channel 1 Interrupt Source Pending Register	0x0
UINTM1	0x7F005438	R/W	UART channel 1 Interrupt Mask Register	0x0

Register	Address	R/W	Description	Reset Value
ULCON2	0x7F005800	R/W	UART channel 2 line control register	0x00
UCON2	0x7F005804	R/W	UART channel 2 control register	0x00
UFCON2	0x7F005808	R/W	UART channel 2 FIFO control register	0x0
UTRSTAT2	0x7F005810	R	UART channel 2 Tx/Rx status register	0x6
UERSTAT2	0x7F005814	R	UART channel 2 Rx error status register	0x0
UFSTAT2	0x7F005818	R	UART channel 2 FIFO status register	0x00
UTXH2	0x7F005820	W	UART channel 2 transmit buffer register	-
URXH2	0x7F005824	R	UART channel 2 receive buffer register	0x00
UBRDIV2	0x7F005828	R/W	UART channel 2 Baud rate divisior register	0x0000
UDIVSLOT2	0x7F00582C	R/W	UART channel 2 Dividing slot register	0x0000
INTP2	0x7F005830	R/W	UART channel 2 Interrupt Pending Register	0x0
UINTSP2	0x7F005834	R/W	UART channel 2 Interrupt Source Pending Register	0x0
UINTM2	0x7F005838	R/W	UART channel 2 Interrupt Mask Register	0x0
ULCON3	0x7F005C00	R/W	UART channel 3 line control register	0x00
UCON3	0x7F005C04	R/W	UART channel 3 control register	0x00
UFCON3	0x7F005C08	R/W	UART channel 3 FIFO control register	0x0
UTRSTAT3	0x7F005C10	R	UART channel 3 Tx/Rx status register	0x6
UERSTAT3	0x7F005C14	R	UART channel 3 Rx error status register	0x0
UFSTAT3	0x7F005C18	R	UART channel 3 FIFO status register	0x00
UTXH3	0x7F005C20	W	UART channel 3 transmit buffer register	-
URXH3	0x7F005C24	R	UART channel 3 receive buffer register	0x00
UBRDIV3	0x7F005C28	R/W	UART channel 3 Baud rate divisior register	0x0000
UDIVSLOT3	0x7F005C2C	R/W	UART channel 3 Dividing slot register	0x0000
INTP3	0x7F005C30	R/W	UART channel 3 Interrupt Pending Register	0x0
UINTSP3	0x7F005C34	R/W	UART channel 3 Interrupt Source Pending Register	0x0
UINTM3	0x7F005C38	R/W	UART channel 3 Interrupt Mask Register	0x0

31.6 INDIVIDUAL REGISTER DESCRIPTIONS

31.6.1 UART LINE CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
ULCON0	0x7F005000	R/W	UART channel 0 line control register	0x00
ULCON1	0x7F005400	R/W	UART channel 1 line control register	0x00
ULCON2	0x7F005800	R/W	UART channel 2 line control register	0x00
ULCON3	0x7F005C00	R/W	UART channel 3 line control register	0x00

There are three UART line control registers including ULCON0, ULCON1, ULCON2 and ULCON3 in the UART block.

ULCONn	Bit	Description	Initial State
Reserved	[7]		0
Infra-Red Mode	[6]	Determine whether or not to use the Infra-Red mode. 0 = Normal mode operation 1 = Infra-Red Tx/Rx mode	0
Parity Mode	[5:3]	Specify the type of parity generation 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]	Specify how many stop bits are to be used for end-of-frame signal. 0 = One stop bit per frame 1 = Two stop bit per frame	0
Word Length	[1:0]	Indicate 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

31.6.2 UART CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
UCON0	0x7F005004	R/W	UART channel 0 control register	0x00
UCON1	0x7F005404	R/W	UART channel 1 control register	0x00
UCON2	0x7F005804	R/W	UART channel 2 control register	0x00
UCON3	0x7F005C04	R/W	UART channel 3 control register	0x00

There are three UART control registers including UCON0, UCON1, UCON2 and UCON3 in the UART block.

UCONn	Bit	Description	Initial State
Clock Selection	[11:10]	Select PCLK or EXT_UCLK0[4] or EXT_UCLK1[4] clock for the UART baud rate. x0 = PCLK : DIV_VAL[1] = (PCLK / (bps x 16)) -1 01 = EXT_UCLK0: DIV_VAL[1] = (EXT_UCLK0 / (bps x 16)) –1 11 = EXT_UCLK1: DIV_VAL[1] = (EXT_UCLK1 / (bps x 16)) –1	0
Tx Interrupt Type	[9]	Interrupt request type. 0 = Pulse (Interrupt is requested as soon as the Tx buffer becomes 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]	Enable/Disable Rx time-out interrupts when UART FIFO is enabled. The interrupt is a receive interrupt. [3] 0 = Disable 1 = Enable	0
Rx Error Status Interrupt Enable	[6]	Enable 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 = Generate receive error status interrupt.	0
Loop-back Mode	[5]	Setting loop-back bit to 1 cause 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 causes 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

UCONn	Bit	Description	Initial State
Transmit Mode	[3:2]	Determine which function is currently able to write Tx data to the UART transmit buffer register. 00 = Disable 01 = Interrupt request or polling mode 10 = DMA request (DMA_UART0) 11 = DMA request (DMA_UART1)	00
Receive Mode	[1:0]	Determine which function is currently able to read data from UART receive buffer register. 00 = Disable 01 = Interrupt request or polling mode 10 = DMA request (DMA_UART0) 11 = DMA request (DMA_UART1)	00

NOTES:

1) DIV_VAL = UBRDIVn + (num of 1's in UDIVSLOTn)/16. Refer to UART Baud Rate Configure Registers.
2) RX interrupt type must be set to pulse for every transfer in S3C6410.
3) When 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 will be generated (receive time out), and then you must check the FIFO status and read out the rest.
4) EXT_UCLK0 clock is external clock.(XpwmECLK PAD input)
   EXT_UCLK1 clock is generated clock by SYSCON. SYSCON generates EXT_UCLK1 for dividing EPLL or MPLL output.
   When you want to change EXT_UCLK0 to PCLK for UART baudrate , clock selection field must be set to 2'b00.
   But, when you want to change EXT_UCLK1 to PCLK for UART baudrate , clock selection field must be set to 2'b10.

31.6.3 UART FIFO CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
UFCON0	0x7F005008	R/W	UART channel 0 FIFO control register	0x0
UFCON1	0x7F005408	R/W	UART channel 1 FIFO control register	0x0
UFCON2	0x7F005808	R/W	UART channel 2 FIFO control register	0x0
UFCON3	0x7F005C08	R/W	UART channel 3 FIFO control register	0x0

There are three UART FIFO control registers including UFCON0, UFCON1, UFCON2 and UFCON3 in the UART block.

UFCONn	Bit	Description	Initial State
Tx FIFO Trigger Level	[7:6]	Determine the trigger level of transmit FIFO. 00 = Empty 01 = 16-byte 10 = 32-byte 11 = 48-byte	00
Rx FIFO Trigger Level	[5:4]	Determine the trigger level of receive FIFO.1) 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

NOTES:

1) 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.

31.6.4 UART MODEM CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
UMCON0	0x7F00500C	R/W	UART channel 0 Modem control register	0x0
UMCON1	0x7F00540C	R/W	UART channel 1 Modem control register	0x0
Reserved	0x7F00580C	-	Reserved	Undef
Reserved	0x7F005C0C	-	Reserved	Undef

There are two UART MODEM control registers including UMCON0 and UMCON1 in the UART block.

UMCONn	Bit	Description	Initial State
RTS trigger Level	[7:5]	When AFC bit is enabled, these bits determine when to inactivate nRTS signal. 000 = When RX FIFO contains 63 bytes. 001 = When RX FIFO contains 56 bytes. 010 = When RX FIFO contains 48 bytes. 011 = When RX FIFO contains 40 bytes. 100 = When RX FIFO contains 32 bytes. 101 = When RX FIFO contains 24 bytes. 110 = When RX FIFO contains 16 bytes. 111 = When RX FIFO contains 8 bytes.	000
Auto Flow Control (AFC)	[4]	0 = Disable 1 = Enable	0
Modem Interrupt enable	[3]	Modem interrupt enable 0 = Disable 1 = Enable	000
Reserved	[2:1]	These bits must be 0's	000
Request to Send	[0]	If AFC bit is enabled, this value will be ignored. In this case the S3C6410 will control 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 does not support AFC function, because the S3C6410 has no nRTS2 and nCTS2.
UART 3 does not support AFC function, because the S3C6410 has no nRTS3 and nCTS3.

31.6.5 UART TX/RX STATUS REGISTER

Register	Address	R/W	Description	Reset Value
UTRSTAT0	0x7F005010	R	UART channel 0 Tx/Rx status register	0x6
UTRSTAT1	0x7F005410	R	UART channel 1 Tx/Rx status register	0x6
UTRSTAT2	0x7F005810	R	UART channel 2 Tx/Rx status register	0x6
UTRSTAT3	0x7F005C10	R	UART channel 3 Tx/Rx status register	0x6

There are three UART Tx/Rx status registers including UTRSTAT0, UTRSTAT1, UTRSTAT2 and UTRSTAT3 in the UART block.

UTRSTATn	Bit	Description	Initial State
Transmitter empty	[2]	Set to 1 automatically when 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]	Set to 1 automatically when transmit buffer register is empty. 0 =The buffer register is not empty 1 = Empty (In Non-FIFO mode, Interrupt or DMA is requested. In FIFO mode, Interrupt or DMA is requested, when Tx FIFO Trigger Level is set to 00 (Empty)) If the UART uses the FIFO, you must check Tx FIFO Count bits and Tx FIFO Full bit in the UFSTAT register instead of this bit.	1
Receive buffer data ready	[0]	Set to 1 automatically whenever receive buffer register contains valid data, received over the RXDn port. 0 = Empty 1 = The buffer register has a received data (In Non-FIFO mode, Interrupt or DMA is requested) If the UART uses the FIFO, you must check Rx FIFO Count bits and Rx FIFO Full bit in the UFSTAT register instead of this bit.	0

31.6.6 UART ERROR STATUS REGISTER

Register	Address	R/W	Description	Reset Value
UERSTAT0	0x7F005014	R	UART channel 0 Rx error status register	0x0
UERSTAT1	0x7F005414	R	UART channel 1 Rx error status register	0x0
UERSTAT2	0x7F005814	R	UART channel 2 Rx error status register	0x0
UERSTAT3	0x7F005C14	R	UART channel 3 Rx error status register	0x0

There are three UART Rx error status registers including UERSTAT0, UERSTAT1, UERSTAT2 and UERSTAT3 in the UART block.

UERSTATn	Bit	Description	Initial State
Break Detect	[3]	Set to 1 automatically to indicate that a break signal has been received. 0 = Send Break signal is not received 1 = Send Break (signal is received Interrupt is requested.)	0
Frame Error	[2]	Set to 1 automatically whenever a frame error occurs during reception operation. 0 = No frame error during reception 1 = Frame error (Interrupt is requested.)	0
Parity Error	[1]	Set to 1 automatically whenever a parity error occurs during receive operation. 0 = No parity error during reception 1 = Parity error (Interrupt is requested.)	0
Overrun Error	[0]	Set to 1 automatically whenever an overrun error occurs during receive operation. 0 = No overrun error during reception 1 = Overrun error (Interrupt is requested.)	0

NOTE: These bits (UERSATn[3:0]) are automatically cleared to 0 when the UART error status register is read.

31.6.7 UART FIFO STATUS REGISTER

Register	Address	R/W	Description	Reset Value
UFSTAT0	0x7F005018	R	UART channel 0 FIFO status register	0x00
UFSTAT1	0x7F005418	R	UART channel 1 FIFO status register	0x00
UFSTAT2	0x7F005818	R	UART channel 2 FIFO status register	0x00
UFSTAT3	0x7F005C18	R	UART channel 3 FIFO status register	0x00

There are three UART FIFO status registers including UFSTAT0, UFSTAT1, UFSTAT2 and UFSTAT3 in the UART block.

UFSTATn	Bit	Description	Initial State
Reserved	[15]		0
Tx FIFO Full	[14]	Set to 1 automatically whenever transmit 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]	Set to 1 automatically whenever receive 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

31.6.8 UART MODEM STATUS REGISTER

Register	Address	R/W	Description	Reset Value
UMSTAT0	0x7F00501C	R	UART channel 0 Modem status register	0x0
UMSTAT1	0x7F00541C	R	UART channel 1 Modem status register	0x0
Reserved	0x7F00581C	-	Reserved	Undef
Reserved	0x7F005C1C	-	Reserved	Undef

There are two UART modem status registers including UMSTAT0 and UMSTAT1 in the UART block.

UMSTAT0	Bit	Description	Initial State
Reserved	[7:5]	reserved	000
Delta CTS	[4]	Indicate that the nCTS input to the S3C6410 has changed state since the last time it was read by CPU. (Figure 31-8) 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

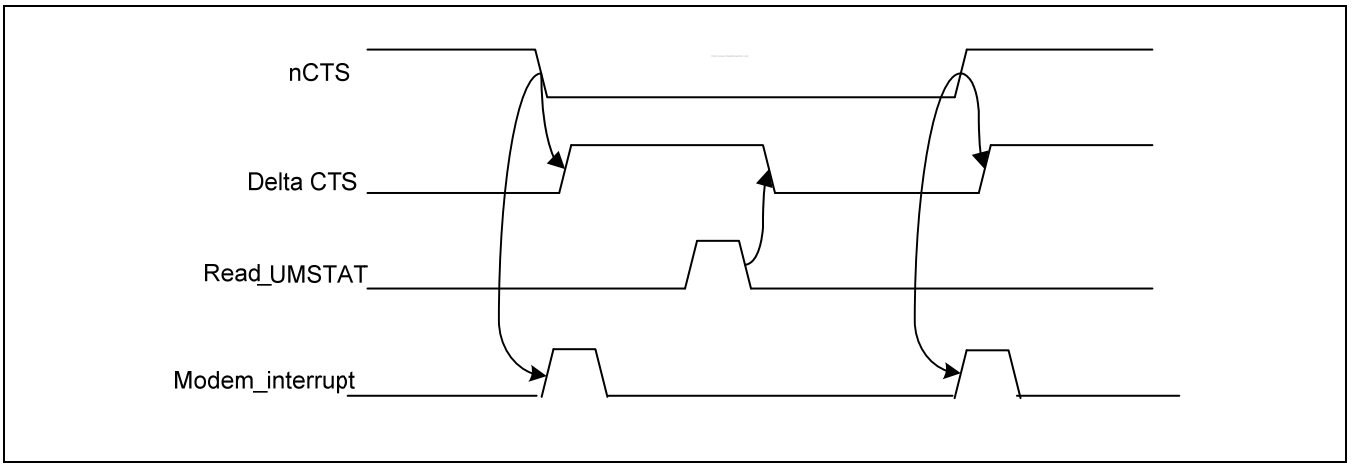


Figure 31-8. nCTS and Delta CTS Timing Diagram

31.6.9 UART TRANSMIT BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

Register	Address	R/W	Description	Reset Value
UTXH0	0x7F005020	W	UART channel 0 transmit buffer register	-
UTXH1	0x7F005420	W	UART channel 1 transmit buffer register	-
UTXH2	0x7F005820	W	UART channel 2 transmit buffer register	-
UTXH3	0x7F005C20	W	UART channel 3 transmit buffer register	-

There are four UART transmitting buffer registers including UTXH0, UTXH1, UTXH2 and UTXH3 in the UART block. UTXHn has an 8-bit data for transmitting data.

UTXHn	Bit	Description	Initial State
TXDATAn	[7:0]	Transmit data for UARTn	-

31.6.10 UART RECIVE BUFFER REGISTER (HOLDING REGISTER & FIFO REGISTER)

There are four UART receiving buffer registers including URXH0, URXH1, URXH2 and URXH3 in the UART block. URXHn has an 8-bit data for received data.

Register	Address	R/W	Description	Reset Value
URXH0	0x7F005024	R	UART channel 0 receive buffer register	0x00
URXH1	0x7F005424	R	UART channel 1 receive buffer register	0x00
URXH2	0x7F005824	R	UART channel 2 receive buffer register	0x00
URXH3	0x7F005C24	R	UART channel 3 receive buffer register	0x00

URXHn	Bit	Description	Initial State
RXDATAn	[7:0]	Receive data for UARTn	0x00

NOTE: When an overrun error occurs, the URXHn must be read. If not, the next received data will also make an overrun error, even though the overrun bit of UERSTATn had been cleared.

31.6.11 UART BAUD RATE CONFIGURE REGISTER

There are four UART baud rate divisor registers including 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 = (EXT_UCLK0 / (bps x 16)) –1

or

DIV_VAL = (EXT_UCLK1 / (bps x 16)) –1

Where, the divisor must be from 1 to (2^{16}-1)
Using UDIVSLOT, you can make more accurate baud rate.
For example, if the baud-rate is 115200 bps and EXT_UCLK0 is UART baud-rate clock and 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)

31.6.12 BAUD-RATE ERROR TOLERANCE

UART Frame error should be less than 1.87%(3/160)

tUPCLK = (UBRDIVn + 1) x 16 x 1Frame / (PCLK ,EXT_UCLK0 or EXT_UCLK1) tUPCLK: Real UART Clock

tEXTUARTCLK = 1Frame / baud-rate tEXTUARTCLK: Ideal UART Clock

UART error = (tUPCLK – tEXTUARTCLK) / tEXTUARTCLK x 100%

NOTE: 1Frame = start bit + data bit + parity bit + stop bit.

31.6.13 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

Register	Address	R/W	Description	Reset Value
UBRDIV0	0x7F005028	R/W	Baud rate divisior register 0	0x0000
UBRDIV1	0x7F005428	R/W	Baud rate divisior register 1	0x0000
UBRDIV2	0x7F005828	R/W	Baud rate divisior register 2	0x0000
UBRDIV3	0x7F005C28	R/W	Baud rate divisior register 3	0x0000

UBRDIV n	Bit	Description	Initial State
UBRDIV	[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.

Register	Address	R/W	Description	Reset Value
UDIVSLOT0	0x7F00502C	R/W	Baud rate divisior register 0	0x0000
UDIVSLOT1	0x7F00542C	R/W	Baud rate divisior register 1	0x0000
UDIVSLOT2	0x7F00582C	R/W	Baud rate divisior register 2	0x0000
UDIVSLOT3	0x7F005C2C	R/W	Baud rate divisior register 3	0x0000

UDIVSLOT n	Bit	Description	Initial State
UDIVSLOT	[15:0]	Select the slot where clock generator divide clock source	-

31.6.14 UART INTERRUPT PENDING REGISTER

Register	Address	R/W	Description	Reset Value
UINTP0	0x7F005030	R/W	Interrupt Pending Register for UART channel 0	0x0
UINTP1	0x7F005430	R/W	Interrupt Pending Register for UART channel 1	0x0
UINTP2	0x7F005830	R/W	Interrupt Pending Register for UART channel 2	0x0
UINTP3	0x7F005C30	R/W	Interrupt Pending Register for UART channel 3	0x0

Interrupt pending register contains the information of the interrupts, which are generated.

UINTPn	Bit	Description	Initial State
MODEM	[3]	Modem interrupt generated.	0
TXD	[2]	Transmit interrupt generated.	0
ERROR	[1]	Error interrupt generated.	0
RXD	[0]	Receive interrupt generated.	0

Whenever one of above 4 bits is logical high (‘1’), each UART channel generates interrupt.
This register has to be cleared in the interrupt service routine.
You can clear specific bits of UINTP register by writing 1’s to the bits that you want to clear.

31.6.15 UART INTERRUPT SOURCE PENDING REGISTER

Interrupt Source Pending Register contains the information which interrupt are generated regardless of the value of Interrupt Mask Register

You can clear specific bits of UINTSP register by writing 1’s to the bits that you want to clear.

Register	Address	R/W	Description	Reset Value
UINTSP0	0x7F005034	R/W	Interrupt Source Pending Register 0	0x0
UINTSP1	0x7F005434	R/W	Interrupt Source Pending Register 1	0x0
UINTSP2	0x7F005834	R/W	Interrupt Source Pending Register 2	0x0
UINTSP3	0x7F005C34	R/W	Interrupt Source Pending Register 3	0x0

UINTSPn	Bit	Description	Initial State
MODEM	[3]	Modem interrupt generated.	0
TXD	[2]	Transmit interrupt generated.	0
ERROR	[1]	Error interrupt generated.	0
RXD	[0]	Receive interrupt generated.	0

31.6.16 UART INTERRUPT MASK REGISTER

Interrupt mask register contains the information of the masked interrupt. If a specific bit is set to 1, interrupt request signal to Interrupt Controller is not generated even though corresponding interrupt is generated. (Note that even in such a case, the corresponding bit of UINTSPn register is set to 1). If the mask bit is 0, the interrupt request can be serviced from the corresponding interrupt source

(Note that even in such a case, the corresponding bit of UINTSPn register is set to 1).

Register	Address	R/W	Description	Reset Value
UINTM0	0x7F005038	R/W	Interrupt Mask Register for UART channel 0	0x0
UINTM1	0x7F005438	R/W	Interrupt Mask Register for UART channel 1	0x0
UINTM2	0x7F005838	R/W	Interrupt Mask Register for UART channel 2	0x0
UINTM3	0x7F005C38	R/W	Interrupt Mask Register for UART channel 3	0x0

UINTMn	Bit	Description	Initial State
MODEM	[3]	Mask Modem interrupt.	0
TXD	[2]	Mask Transmit interrupt.	0
ERROR	[1]	Mask Error interrupt.	0
RXD	[0]	Mask Receive interrupt.	0

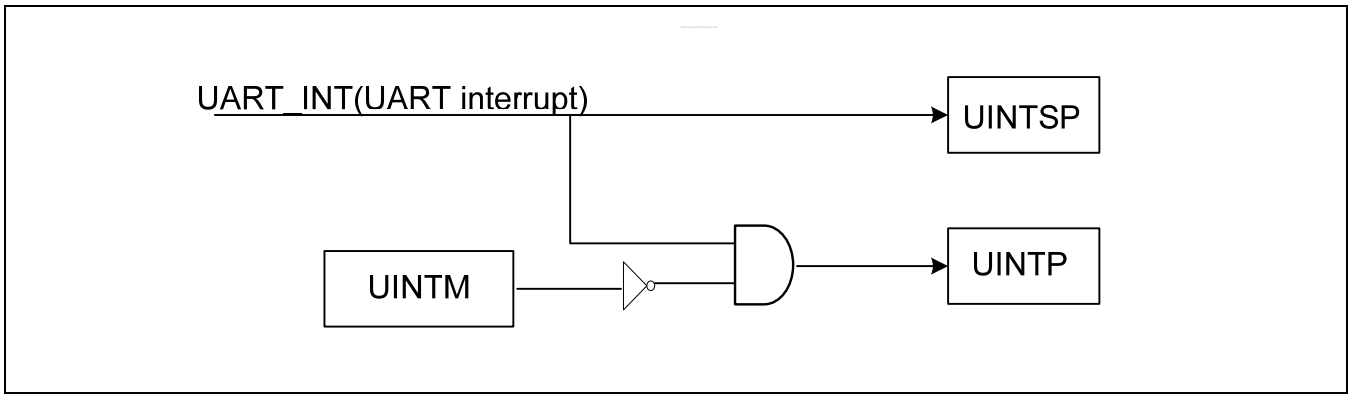


Figure 31-9. UINTSP,UINTP and UINTM block diagram

NOTES

32 PWM TIMER

This chapter describes the functions and usage of PWM TIMER in S3C6410X RISC microprocessor.

32.1 OVERVIEW

The S3C6410X RISC microprocessor comprises of five 32-bit timers. These timers are used to generate internal interrupts to the ARM subsystem. In addition, Timers 0 and 1 include a PWM function (Pulse Width Modulation), which can drive an external I/O signal. The PWM for timer 0 and 1 have an optional dead-zone generator capability, which can be utilized to support a large current device. Timer 2, 3 and 4 are internal timers with no output pins.

32.2 FEATURES

The Features supported by the PWM are as follows:

- 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 External Clocks.
- Programmable Clock Select Logic for individual PWM Channels.
- Two Independent PWM Channels with Programmable Duty Control and Polarity.
- Supports Auto-Reload Mode and One-Shot Pulse Mode.
- Supports for external inputs to start PWM.
- Dead Zone Generator on two PWM Outputs.
- Supports DMA Transfers.
- Optional Pulse or 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, 16 special function registers are provided. The PWM is a programmable output. The 16 special function registers within PWM are accessed via APB transactions.

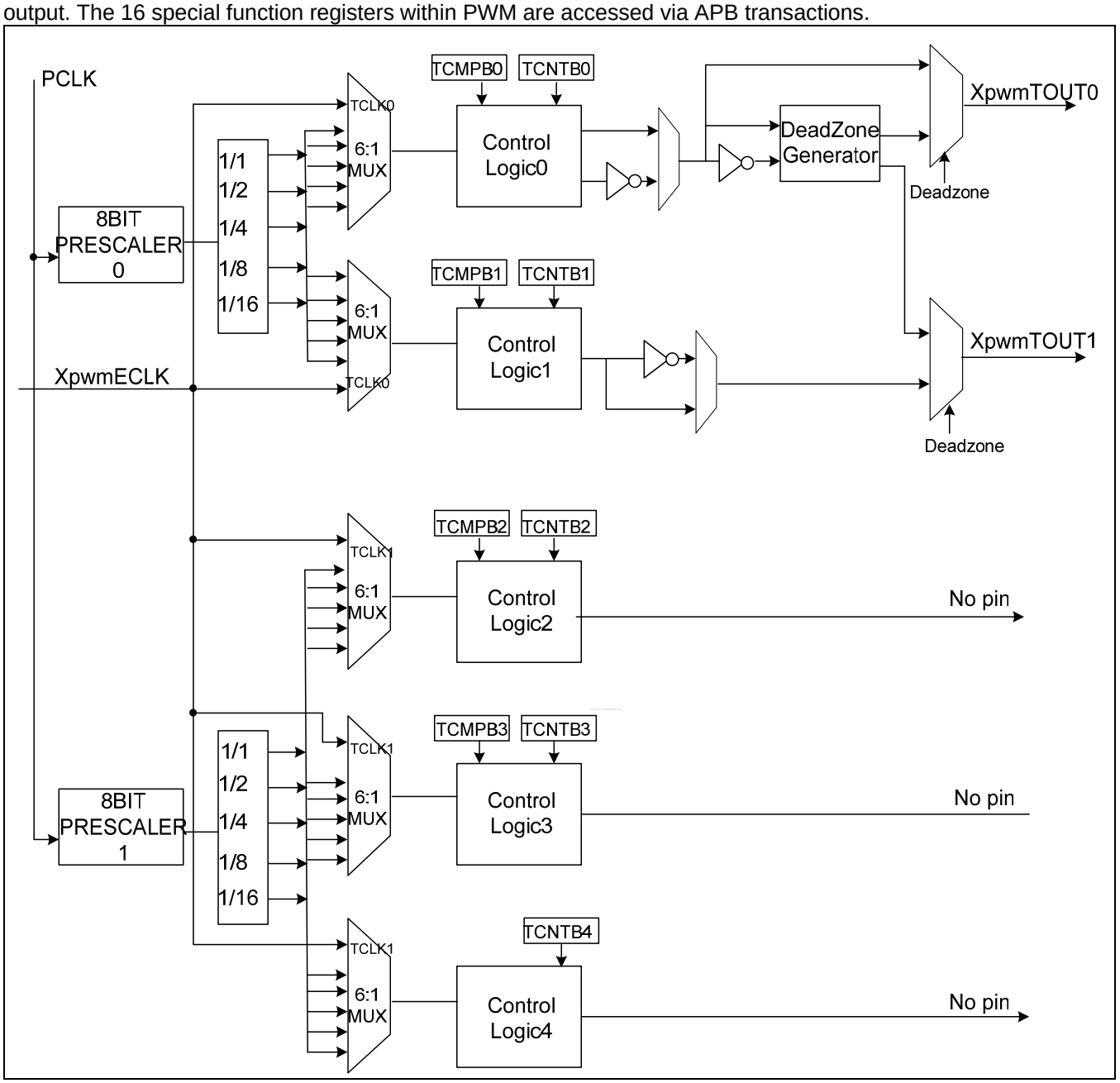


Figure 32-1. PWMTIMER Clock Tree Diagram

The Figure 32-1 depict the clock generation scheme for individual PWM Channels.

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 select a clock source from an external pin. Timers 0 and 1 can select the external clock TCLK0. Timers 2, 3, and 4 can select the external clock TCLK1. 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). When the down-counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation is completed. When the timer down-counter reaches zero, the value of corresponding TCNTBn can be automatically reloaded into the down-counter to start the 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 will not be 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 when the down-counter value matches the value of the compare register in the 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 will not take effect until the current timer cycle completes.
A simple example of a PWM cycle is shown in the figure below.

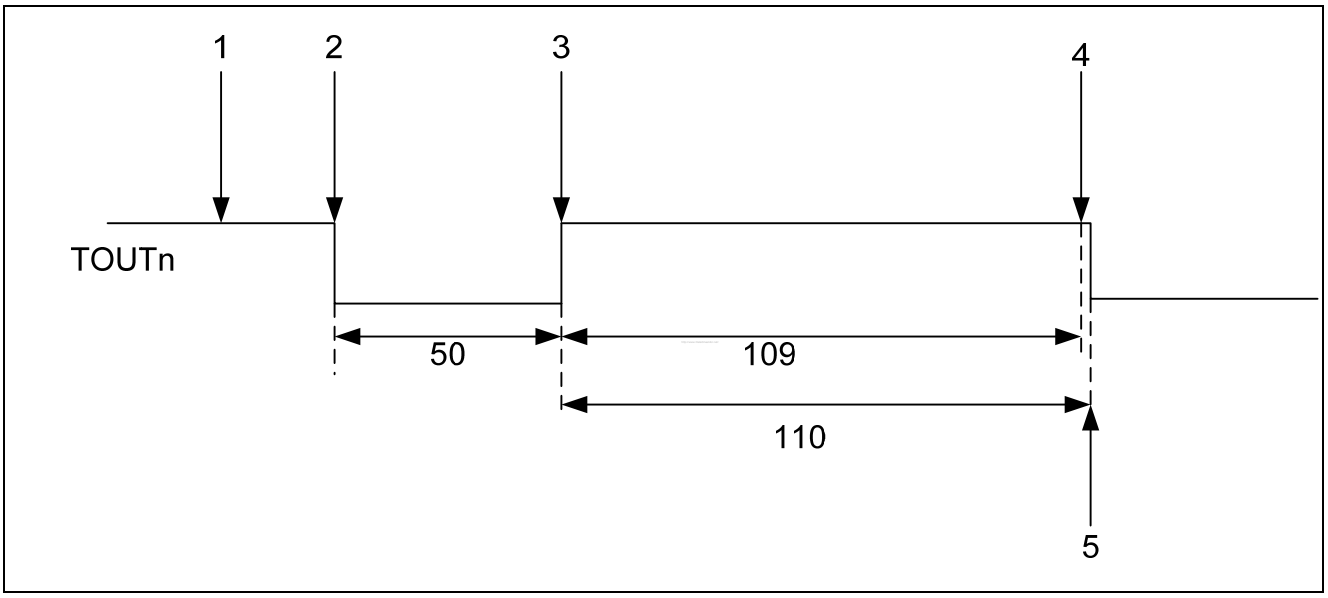


Figure 32-2. Simple Example of PWM Cycle Block Diagram

1. Initialize the TCNTBn with 159(50+109) and the TCMPBn with 109.
2. Start Timer by setting the start bit and manual update bit off.
   The TCNTBn value of 159 is loaded into the down-counter, the output is driven low.
3. When down-counter counts down to the value in the TCMPBn register 109,
   the output is changed from low to high
4. When the down-counter reaches 0, the interrupt request is generated.
5. The down-counter is automatically reloaded with TCNTBn, which restarts the cycle.

The Figure 32-3 depicts the block diagram for PWM Timer.

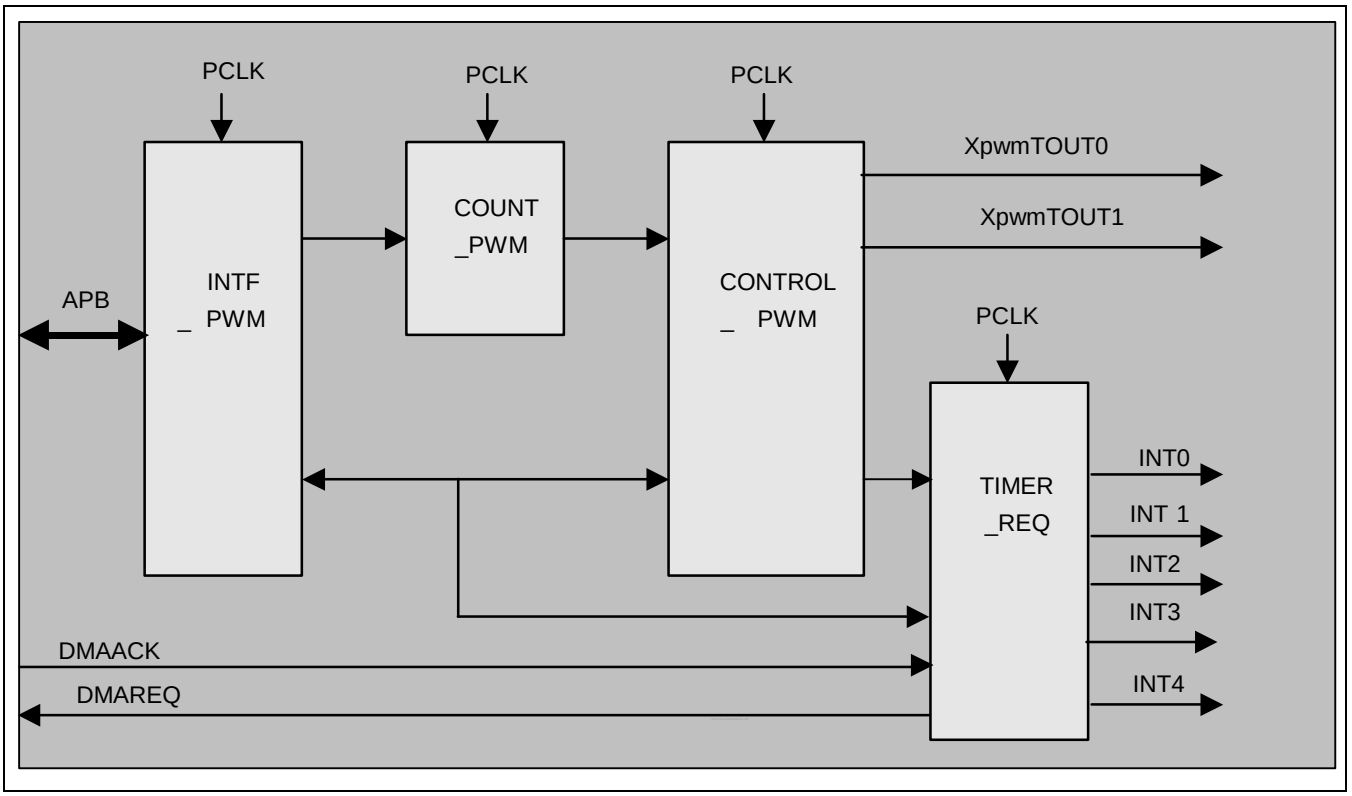


Figure 32-3. PWMTIMER Block Diagram

32.3 PWM OPERATION

32.3.1 PRESCALER & DIVIDER

An 8-bit prescaler and 4-bit divider makes the following output frequencies:

Table 32-1. Min. and Max. Resolution based on Prescaler and Clock Divider Values

4-bit divider settings	Minimum resolution (prescaler=1)	Maximum resolution (prescaler=255)	Maximum interval (TCNTBn=4294967295)
1/1(PCLK=66 MHz)	0.030us (33.0 MHz)	3.87us(258 kHz)	16621.52s
1/2 (PCLK=66 MHz)	0.060us (16.5 MHz)	7.75us (129 kHz)	33243.05s
1/4 (PCLK=66 MHz)	0.121us (8.25 MHz)	15.5us (64.5 kHz)	66486.09s
1/8 (PCLK=66 MHz)	0.242us (4.13 MHz)	31.0us (32.2 kHz)	132972.19s
1/16 (PCLK=66 MHz)	0.484us (2.07 MHz)	62.1us (16.1 kHz)	265944.37s

32.3.2 BASIC TIMER OPERATION

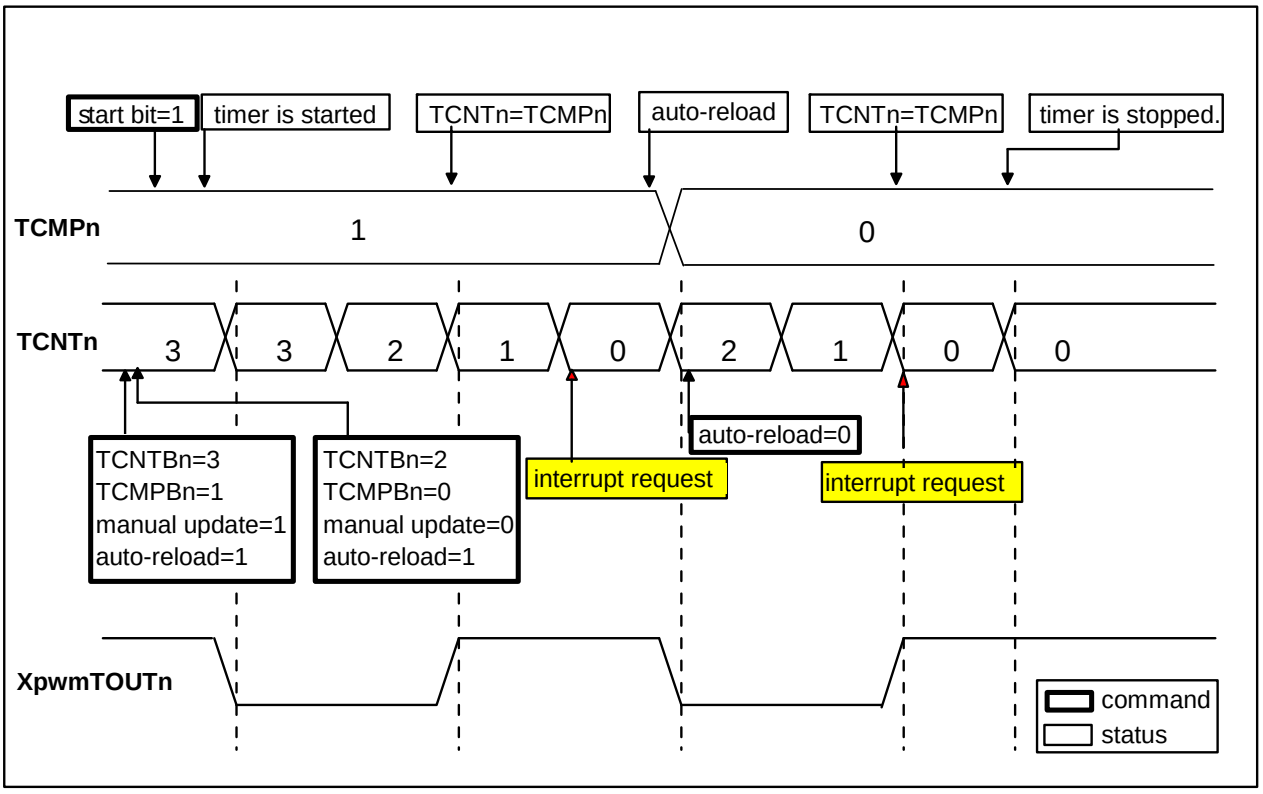


Figure 32-4. Basic Timer operations example

A timer (except the timer channel 4) has TCNTBn, TCNTn, TCMPBn and TCMPn. TCNTBn and TCMPBn are loaded into TCNTn and TCMPn when the timer reaches 0. When TCNTn reaches 0, the interrupt request will occur if the interrupt is enabled. (TCNTn and TCMPn are the names of the internal registers. The TCNTn register can be read from the TCNTOn register)
If you want to generate interrupt at intervals 3cycle of XpwmTOUTn, set TCNTBn, TCMPBn and TCON register like Figure 32-4.
That is:

1. Set TCNTBn=3 and TCMPBn=1.
2. Set auto-reload=1 and manual update=1.
   When 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.
   So, 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.
   When TCNTn is 0, interrupt is generated and if auto-reload is enable, TCNTn is loaded 2(TCNTBn value) and TCMPn is loade 0(TCMPn value).
6. Before stop, TCNTn is down counting.

32.3.3 AUTO-RELOAD AND DOUBLE BUFFERING

The Timers have a double buffering feature, which can change the reload value for the next timer operation without stopping the current timer operation. Though the new timer value is set, current timer operation is completed successfully.
The timer value can be written into TCNTBn (Timer Count Buffer register) and the current counter value of the timer can be read from TCNTOn (Timer Count Observation register). If TCNTBn is read, the read value is the reload value for the next timer duration not the current state of the counter.
The auto-reload is the operation, which copies the TCNTBn into TCNTn when TCNTn reaches 0. The value, written into TCNTBn, is loaded to TCNTn only when 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.

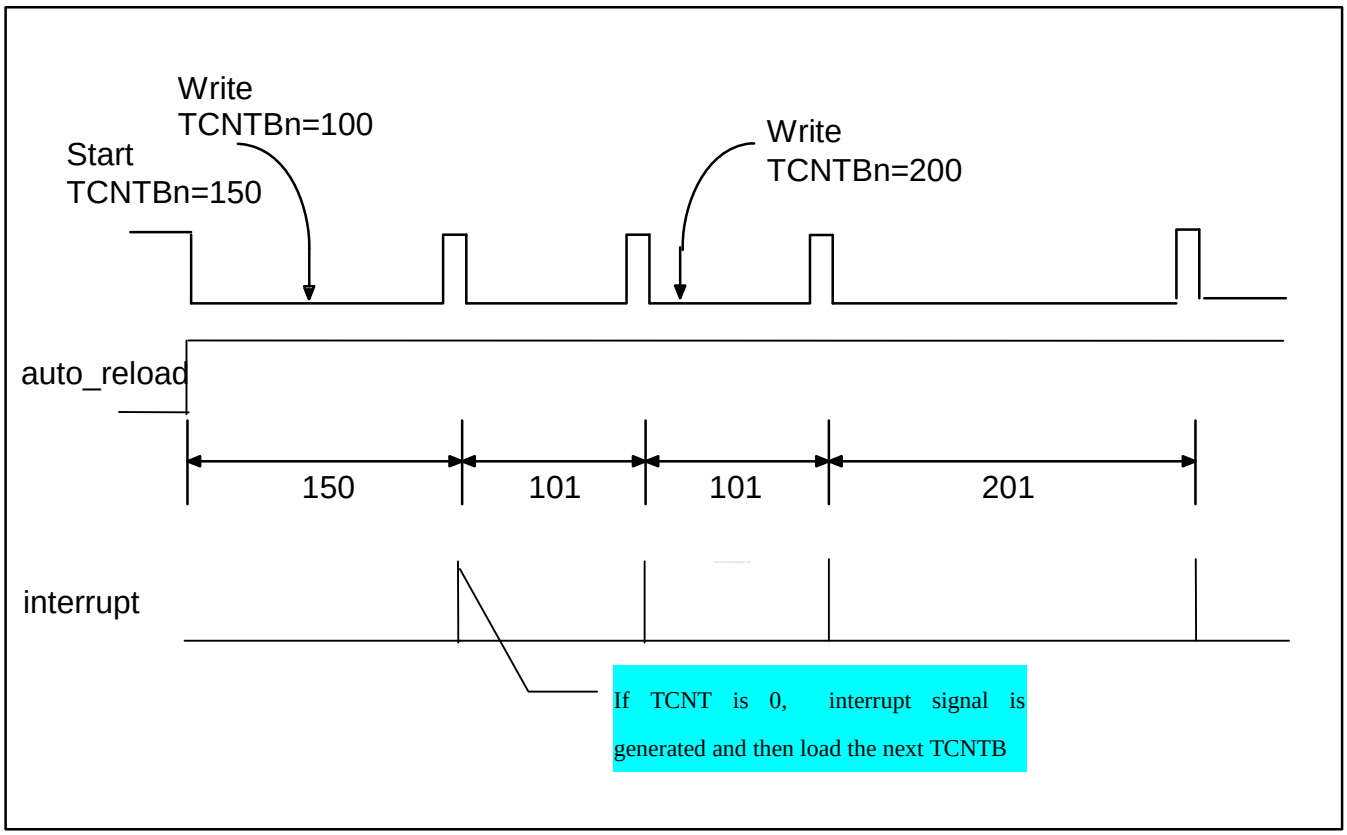


Figure 32-5. Example of Double Buffering Feature Block Diagram

32.3.4 TIMER OPERATION EXAMPLE

The result of the following procedure is shown in Figure 32-6.

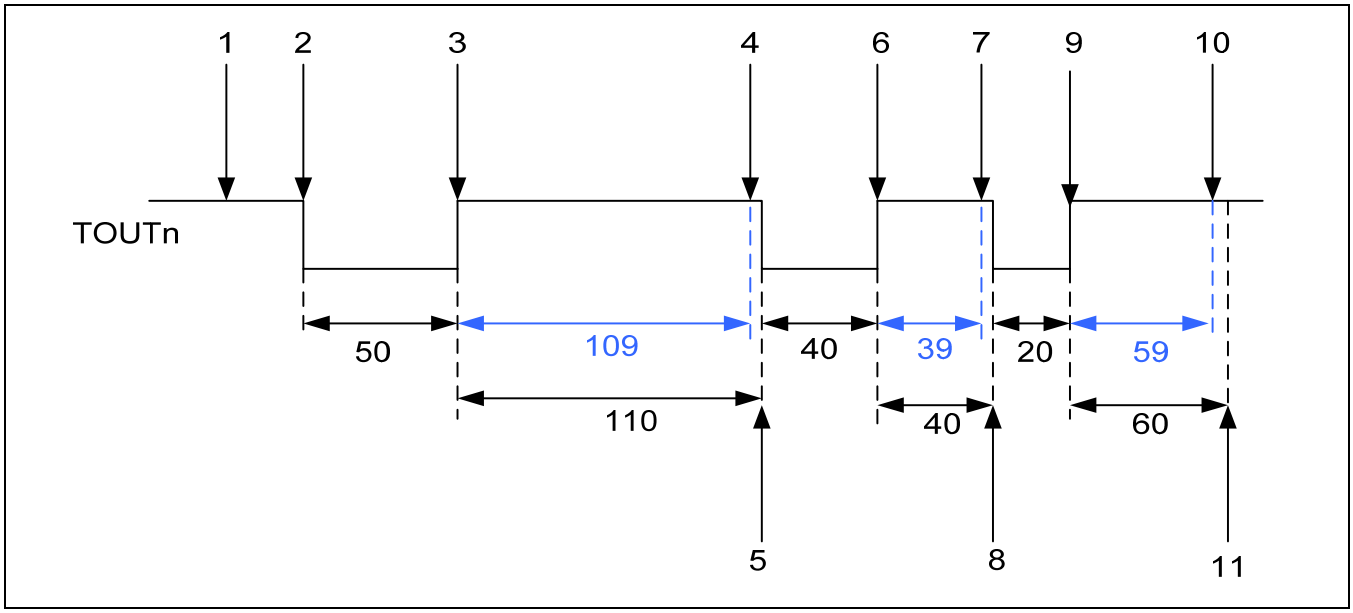


Figure 32-6. Example of a Timer Operation

1. Enable the auto-reload feature. Set the TCNTBn as 159(50+109) and the TCMPBn as 109. Set the manual update bit and inverter bit(on/off). The manual update bit sets the TCNTn,TCMPn to the value of TCNTBn,TCMPBn.

   And then, set TCNTBn,TCMPBn as 79(40+39) and 39.
2. Start Timer by setting the start bit and manual update bit off.
3. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
4. As soon as TCNTn reaches to 0, the interrupt request is generated.
5. TCNTn and TCMPn are reloaded automatically with TCNTBn,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. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
7. As soon as TCNTn reaches to 0, the interrupt request is generated.
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. When TCNTn has the same value with TCMPn, the logic level of TOUTn is changed from low to high
10. Even when TCNTn reaches to 0, No interrupt request is generated.
11. TCNTn is not any more reloaded and the timer is stopped because auto-reload is disabled.

32.3.5 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 has to be loaded by the manual update bit. Take the following steps to start a Timer;

1) Write the initial value into TCNTBn and TCMPBn.
2) Set the manual update bit of the corresponding timer.
   (Recommended setting the inverter on/off bit (whether using inverter or not)).
3) Set the start bit of the corresponding timer to start the timer and clear only manual update bit.

32.3.6 PWM (PULSE WIDTH MODULATION)

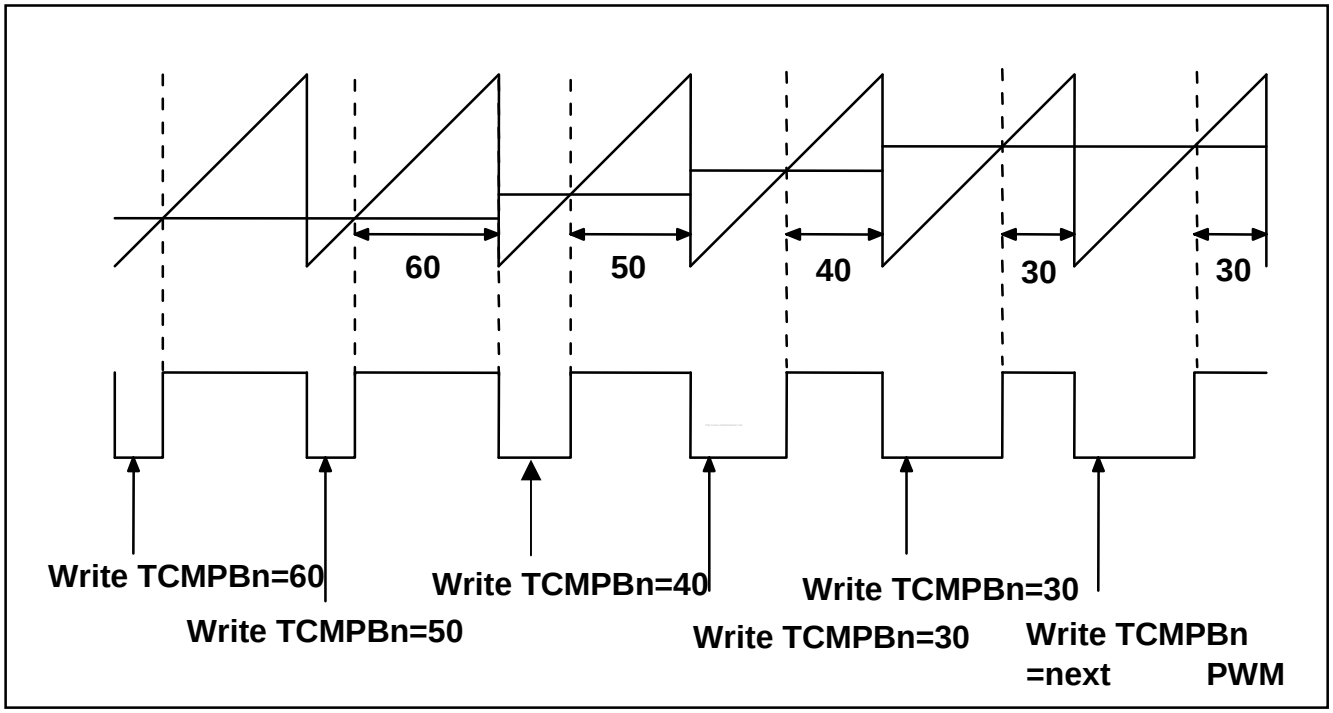


Figure 32-7. Example of PWM

NOTE:

PWM feature can implement by using the TCMPBn. PWM frequency is determined by TCNTBn. A PWM value is determined by TCMPBn as shown in the Figure 32-7.

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, can be written in any point of current PWM cycle by ISR.

32.3.7 OUTPUT LEVEL CONTROL

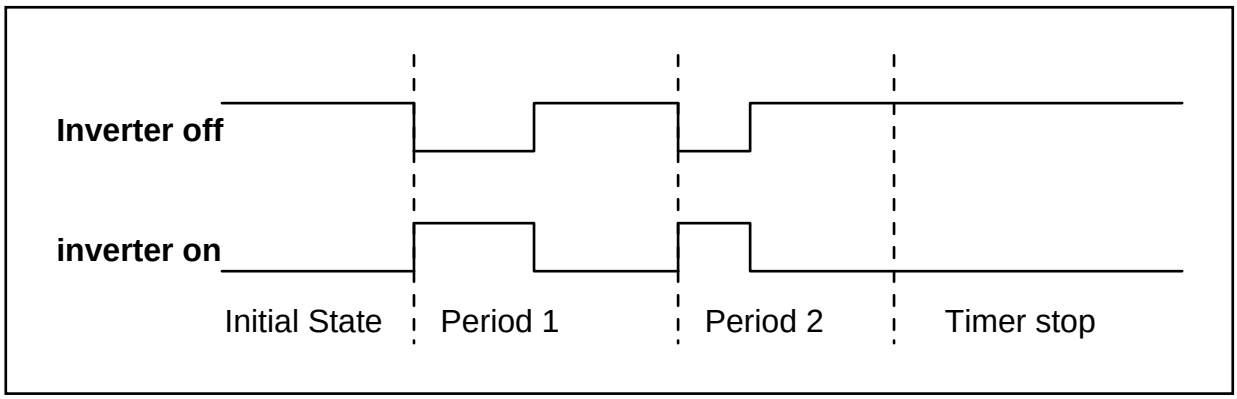


Figure 32-8. Inverter on/off

The following methods can be used to maintain TOUT as high or low. (Assuming the 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 can be inverted by the inverter on/off bit in TCON. The inverter removes the additional circuit to adjust the output level.

32.3.8 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 prohibit the two switching device turning on simultaneously even for a very short time.

TOUT0 is the PWM output. nTOUT0 is the inversion of the TOUT0. If the dead-zone is enabled, the output waveform of TOUT0, nTOUT0 will be TOUT0_DZ and nTOUT0_DZ. TOUT0_DZ and nTOUT0_DZ never can be turned on simultaneously by the dead zone interval. For functional correctness, the deadzone length must be set smaller than compare counter value.

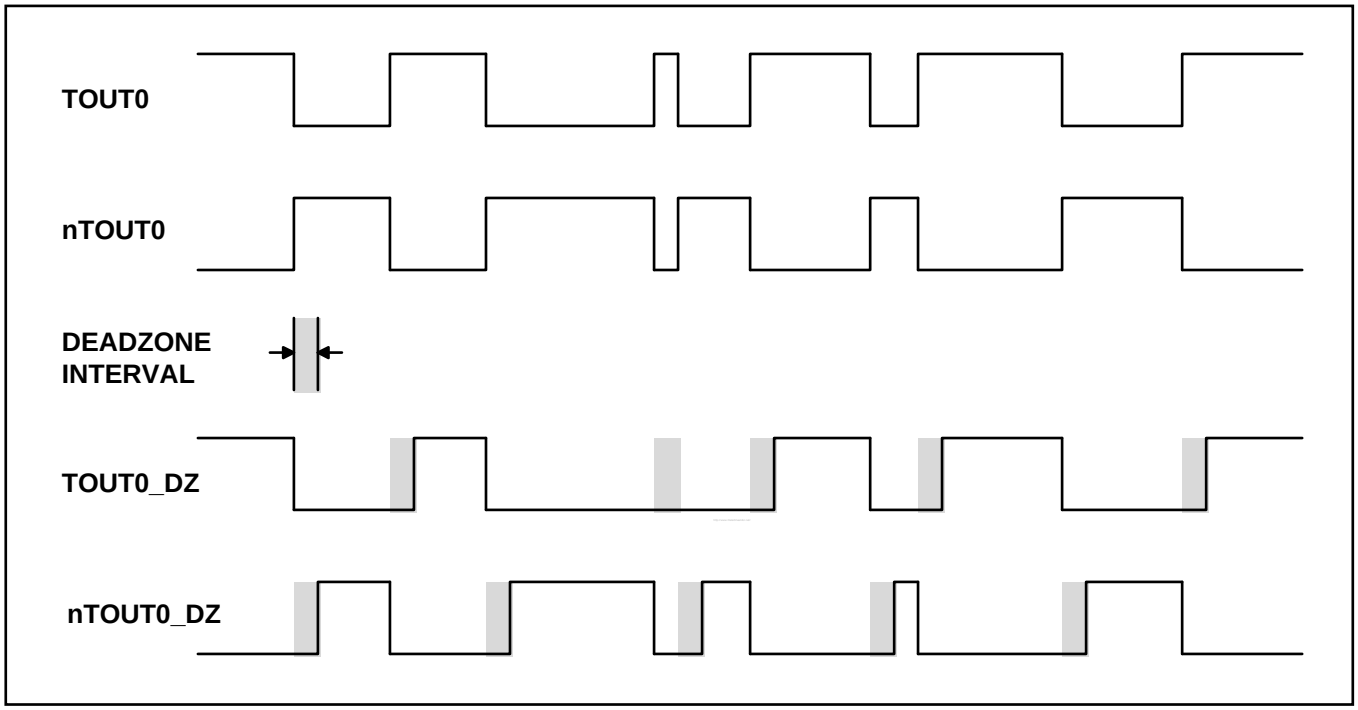


Figure 32-9. The waveform when a deadzone feature is enabled.

32.3.9 DMA REQUEST MODE

Instead of sending an Interrupt at the end of the down-counter cycle, a Timer can be configured to send a DMA request signal to one of the DMA channels. This mode is intended to allow DMA transfers between a source and destination to occur at regular intervals.
The timer will keep the DMA request signal (DMA_REQ) active (high) until the timer receives the ACK signal from the DMA unit. When the timer receives the ACK signal, it makes the request signal inactive.

Only one Timer at a time can be configured as the DMA request source. The timer that generates the DMA request is determined by setting DMA mode bits (in TCFG1 register). If a particular timer is configured as DMA request mode, that timer sends the DMA request and generate the normal ARM interrupt request simultaneously. However the other Timers, which are not configured for DMA mode, can still generate ARM interrupts normally. If none of the Timers are configured in DMA mode, then all of them can generate normal interrupts.

The Figure 32-10 shows how the DMA request will remain active until the DMA sends the ACK signal.

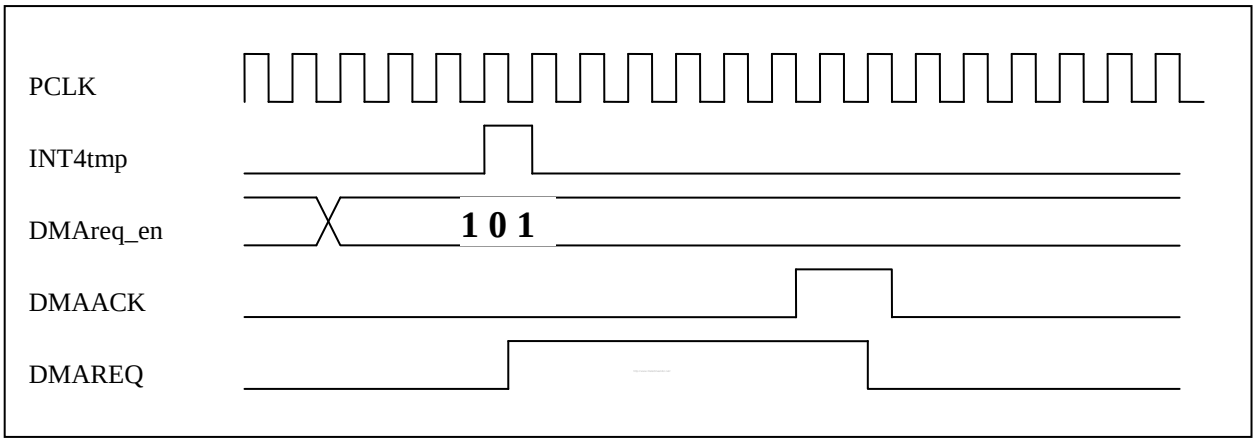


Figure 32-10. DMA Operation

The table below shows the action of each Timer at the completion of the down-count as a function of the DMA mode configuration:

32.3.10 TIMER INTERRUPT GENERATION

The PWMTIMER provides flexibility to generate Pulse and Level Interrupts by controlling the 'INTRGEN_SEL' port status. When the port 'INTRGEN_SEL' is tied to logic 1, optional level interrupts will be generated or optional pulse interrupts will be generated. The interrupt generation is controlled by writing specific values to the 'TINT_CSTAT' register within PWMTIMER. Interrupt generation is optional based on programmed value in 'TINT_CSTAT' register.

32.4 SPECIAL FUNCTION REGISTERS

32.4.1 REGISTER MAP

Register	Offset	R/W	Description	Reset Value
TCFG0	0x7F006000	R/W	Timer Configuration Register 0 that configures the two 8-bit Prescaler and DeadZone Length	0x0000_0101
TCFG1	0x7F006004	R/W	Timer Configuration Register 1 that controls 5 MUX and DMA Mode Select Bit	0x0000_0000
TCON	0x7F006008	R/W	Timer Control Register	0x0000_0000
TCNTB0	0x7F00600C	R/W	Timer 0 Count Buffer Register	0x0000_0000
TCMPB0	0x7F006010	R/W	Timer 0 Compare Buffer Register	0x0000_0000
TCNTO0	0x7F006014	R	Timer 0 Count Observation Register	0x0000_0000
TCNTB1	0x7F006018	R/W	Timer 1 Count Buffer Register	0x0000_0000
TCMPB1	0x7F00601c	R/W	Timer 1 Compare Buffer Register	0x0000_0000
TCNTO1	0x7F006020	R	Timer 1 Count Observation Register	0x0000_0000
TCNTB2	0x7F006024	R/W	Timer 2 Count Buffer Register	0x0000_0000
TCNTO2	0x7F00602c	R	Timer 2 Count Observation Register	0x0000_0000
TCNTB3	0x7F006030	R/W	Timer 3 Count Buffer Register	0x0000_0000
TCNTO3	0x7F006038	R	Timer 3 Count Observation Register	0x0000_0000
TCNTB4	0x7F00603c	R/W	Timer 4 Count Buffer Register	0x0000_0000
TCNTO4	0x7F006040	R	Timer 4 Count Observation Register	0x0000_0000
TINT_CSTAT	0x7F006044	R/W	Timer Interrupt Control and Status Register	0x0000_0000

32.4.1.1 TCFG0 (Timer Configuration Register)

Register	Offset	R/W	Description	Reset Value
TCFG0	0x7F006000	R/W	Timer Configuration Register 0 that configures the two 8-bit Prescaler and DeadZone Length	0x0000_0101

Timer input clock Frequency = PCLK / ({prescaler value + 1}) / {divider value}
{prescaler value} = 1~255
{divider value} = 1, 2, 4, 8, 16, TCLK

TCFG0	Bit	R/W	Description	Initial State
Reserved	[31:24]	R	Reserved Bits	0x00
Dead zone length	[23:16]	R/W	Dead zone length	0x00
Prescaler 1	[15:8]	R/W	Prescaler 1 value for Timer 2, 3 and 4	0x01
Prescaler 0	[7:0]	R/W	Prescaler 0 value for timer 0 & 1	0x01

32.4.1.2 TCFG1 (Timer Configuration Register)

Register	Offset	R/W	Description	Reset Value
TCFG1	0x7F006004	R/W	Timer Configuration Register 1 that controls 5 MUX and DMA Mode Select Bit	0x0000_0000

TCFG1	Bit	R/W	Description	Initial State
Reserved	[31:24]	R	Reserved Bits	0x00
DMA mode	[23:20]	R/W	Select DMA Request Channel Select Bit 0000: No select 0001: INT0 0010: INT1 0011: INT2 0100: INT3 0101: INT4 0110: No select 0111: No select	0x0
Divider MUX4	[19:16]	R/W	Select Mux input for PWM Timer 4 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	0x0
Divider MUX3	[15:12]	R/W	Select Mux input for PWM Timer 3 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	0x0
Divider MUX2	[11:8]	R/W	Select Mux input for PWM Timer 2 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK1 0110: External TCLK1 0111: External TCLK1	0x0
Divider MUX1	[7:4]	R/W	Select Mux input for PWM Timer 1	0x0

			0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK0 0110: External TCLK0 0111: External TCLK0	
Divider MUX0	[3:0]	R/W	Select Mux input for PWM Timer 0 0000:1/1 0001:1/2 0010:1/4 0011:1/8 0100: 1/16 0101: External TCLK0 0110: External TCLK0 0111: External TCLK0	0x0

32.4.1.3 TCON (Timer Control Register)

Register	Offset	R/W	Description	Reset Value
TCON	0x7F006008	R/W	Timer Control Register	0x0000_0000

TCON	Bit	R/W	Description	Initial State
Reserved	[31:23]	R	Reserved Bits	0x000
Timer 4 Auto Reload on/off	[22]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Timer 4 Manual Update	[21]	R/W	0: No Operation 1: Update TCNTB4	0
Timer 4 Start/Stop	[20]	R/W	0: Stop 1: Start Timer 4	0
Timer 3 Auto Reload on/off	[19]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Reserved	[18]	R/W	Reserved Bits	0
Timer 3 Manual Update	[17]	R/W	0: No Operation 1:Update TCNTB3,TCMPB3	0
Timer 3 Start/Stop	[16]	R/W	0: Stop 1: Start Timer 3	0
Timer 2 Auto Reload on/off	[15]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Reserved	[14]	R/W	Reserved Bits	0
Timer 2 Manual Update	[13]	R/W	0: No Operation 1: Update TCNTB2,TCMPB2	0
Timer 2 Start/Stop	[12]	R/W	0: Stop 1: Start Timer 2	0
Timer 1 Auto Reload on/off	[11]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Timer 1 Output Inverter on/off	[10]	R/W	0: Inverter Off 1: TOUT1 Inverter-On	0
Timer 1 Manual Update	[9]	R/W	0: No Operation 1: Update TCNTB1,TCMPB1	0
Timer 1 Start/Stop	[8]	R/W	0: Stop 1: Start Timer 1	0
Reserved	[7:5]	R/W	Reserved Bits	000
Dead zone enable/disable	[4]	R/W	Deadzone Generator Enable/Disable	0
Timer 0 Auto Reload on/off	[3]	R/W	0: One-Shot 1: Interval Mode(Auto-Reload)	0
Timer 0 Output Inverter on/off	[2]	R/W	0: Inverter Off 1: TOUT0 Inverter-On	0
Timer 0 Manual Update	[1]	R/W	0: No Operation 1: Update TCNTB0,TCMPB0	0
Timer 0 Start/Stop	[0]	R/W	0: Stop 1: Start Timer 0	0

32.4.1.4 TCNTB0 (Timer0 Counter Register)

Register	Offset	R/W	Description	Reset Value
TCNTB0	0x7F00600C	R/W	Timer 0 Count Buffer Register	0x0000_0000

TCNTB0	Bit	R/W	Description	Initial State
Timer 0 Count Buffer	[31:0]	R/W	Timer 0 Count Buffer Register	0x00000000

32.4.1.5 TCMPB0 (Timer0 Compare Register)

Register	Offset	R/W	Description	Reset Value
TCMPB0	0x7F006010	R/W	Timer 0 Compare Buffer Register	0x0000_0000

TCMPB0	Bit	R/W	Description	Initial State
Timer 0 Compare Buffer	[31:0]	R/W	Timer 0 Compare Buffer Register	0x00000000

32.4.1.6 TCNTO0 (Timer0 Observation Register)

Register	Offset	R/W	Description	Reset Value
TCNTO0	0x7F006014	R	Timer 0 Count Observation Register	0x0000_0000

TCNTO0	Bit	R/W	Description	Initial State
Timer 0 Count Observation	[31:0]	R	Timer 0 Count Observation Register	0x00000000

32.4.1.7 TCNTB1 (Timer1 Counter Register)

Register	Offset	R/W	Description	Reset Value
TCNTB1	0x7F006018	R/W	Timer 1 Count Buffer Register	0x0000_0000

TCNTB1	Bit	R/W	Description	Initial State
Timer 1 Count Buffer	[31:0]	R/W	Timer 1 Count Buffer Register	0x00000000

32.4.1.8 TCMPB1 (TIMER1 COMPARE REGISTER)

Register	Offset	R/W	Description	Reset Value
TCMPB1	0x7F00601c	R/W	Timer 1 Compare Buffer Register	0x0000_0000

TCMPB1	Bit	R/W	Description	Initial State
Timer 1 Compare Buffer	[31:0]	R/W	Timer 1 Compare Buffer Register	0x00000000

32.4.1.9 TCNTO1 (Timer1 Observation Register)

Register	Offset	R/W	Description	Reset Value
TCNTO1	0x7F006020	R	Timer 1 Count Observation Register	0x0000_0000

TCNTO1	Bit	R/W	Description	Initial State
Timer 1 Count Observation	[31:0]	R	Timer 1 Count Observation Register	0x00000000

32.4.1.10 TCNTB2 (Timer2 Counter Register)

Register	Offset	R/W	Description	Reset Value
TCNTB2	0x7F006024	R/W	Timer 2 Count Buffer Register	0x0000_0000

TCNTB2	Bit	R/W	Description	Initial State
Timer 2 Count Buffer	[31:0]	R/W	Timer 2 Count Buffer Register	0x00000000

32.4.1.11 TCNTO2 (Timer2 Observation Register)

Register	Offset	R/W	Description	Reset Value
TCNTO2	0x7F00602C	R	Timer 2 Count Observation Register	0x0000_0000

TCNTO2	Bit	R/W	Description	Initial State
Timer 2 Count Observation	[31:0]	R	Timer 2 Count Observation Register	0x00000000

32.4.1.12 TCNTB3 (Timer3 Counter Register)

Register	Offset	R/W	Description	Reset Value
TCNTB3	0x7F006030	R/W	Timer 3 Count Buffer Register	0x0000_0000

TCNTB3	Bit	R/W	Description	Initial State
Timer 3 Count Buffer	[31:0]	R/W	Timer 3 Count Buffer Register	0x00000000

32.4.1.13 TCNTO3 (Timer3 Observation Register)

Register	Offset	R/W	Description	Reset Value
TCNTO3	0x7F006038	R	Timer 3 Count Observation Register	0x0000_0000

TCNTO3	Bit	R/W	Description	Initial State
Timer 3 Count Observation	[31:0]	R	Timer 3 Count Observation Register	0x00000000

32.4.1.14 TCNTB4 (Timer4 Counter Register)

Register	Offset	R/W	Description	Reset Value
TCNTB4	0x7F00603C	R/W	Timer 4 Count Buffer Register	0x0000_0000

TCNTB4	Bit	R/W	Description	Initial State
Timer 4 Count Buffer	[31:0]	R/W	Timer 4 Count Buffer Register	0x00000000

32.4.1.15 TCNTO4 (Timer4 Observation Register)

Register	Offset	R/W	Description	Reset Value
TCNTO4	0x7F006040	R	Timer 4 Count Observation Register	0x0000_0000

TCNTO4	Bit	R/W	Description	Initial State
Timer 4 Count Observation	[31:0]	R	Timer 4 Count Observation Register	0x00000000

32.4.1.16 TINT_CSTAT (Interrupt Control And Status Register)

Register	Offset	R/W	Description	Reset Value
TINT_CSTAT	0x7F006044	R/W	Timer Interrupt Control and Status Register	0x0000_0000

TINT_CSTAT	Bit	R/W	Description	Initial State
Reserved	[31:10]	R	Reserved Bits	0x00000
Timer 4 Interrupt Status	[9]	R/W	Timer 4 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 3 Interrupt Status	[8]	R/W	Timer 3 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 2 Interrupt Status	[7]	R/W	Timer 2 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 1 Interrupt Status	[6]	R/W	Timer 1 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 0 Interrupt Status	[5]	R/W	Timer 0 Interrupt Status Bit. Clears by writing '1' on this bit.	0x0
Timer 4 interrupt Enable	[4]	R/W	Timer 4 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 3 interrupt Enable	[3]	R/W	Timer 3 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 2 interrupt Enable	[2]	R/W	Timer 2 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 1 interrupt Enable	[1]	R/W	Timer 1 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0
Timer 0 interrupt Enable	[0]	R/W	Timer 0 Interrupt Enable. 1 – Enabled 0 – Disabled	0x0

33 REAL TIME CLOCK (RTC)

This chapter describes the functions and usage of Real Time Clock (RTC) in S3C6410X RISC microprocessor.

33.1 OVERVIEW

The Real Time Clock (RTC) unit can be operated by the backup battery when 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 can perform the alarm function.

33.2 FEATURES

The Real Time Clock includes the following 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.

33.3 REAL TIME CLOCK OPERATION DESCRIPTION

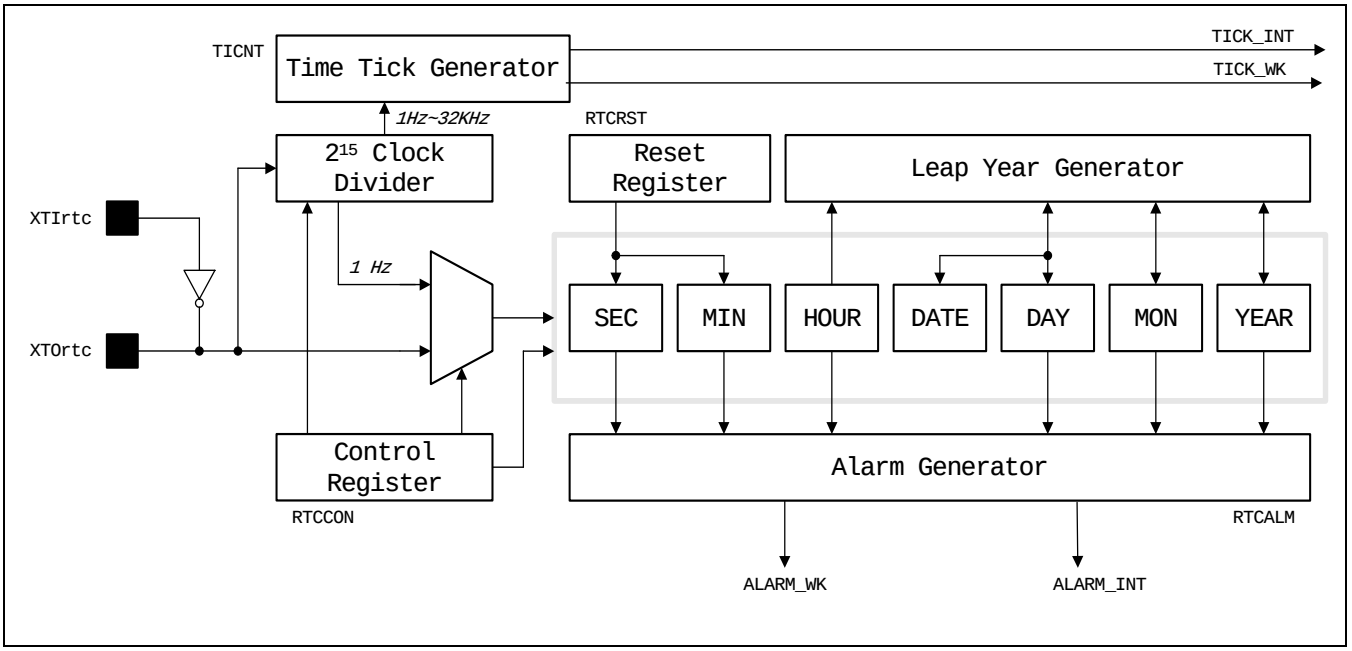


Figure 33-1. Real Time Clock Block Diagram

33.3.1 LEAP YEAR GENERATOR

The leap year generator can determine 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 on the last date. An 8-bit counter can only represent 2 BCD digits, therefore it cannot decide whether “00” year (the 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 S3C6410 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 S3C6410 denote 2000, not 1900.
So, RTC in S3C6410 supports from 1901 to 2099.

33.3.2 READ/WRITE REGISTER

Bit 0 of the RTCCON register must be set high in order to write the BCD register in RTC block. To display the second, minute, hour, day, date, month, and year, the CPU must 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, when the user reads the registers from BCDYEAR to BCDMIN, the result is assumed to be 2059 (Year), 12 (Month), 31 (Date), 23 (Hour) and 59 (Minute). When the user 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, the user must re-read from BCDYEAR to BCDSEC if BCDSEC is zero.

33.3.3 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. When the system is off, the interfaces of the CPU and RTC logic must be blocked, and the backup battery only drives the oscillation circuit and the BCD counters to minimize power dissipation.

33.3.4 ALARM FUNCTION

The RTC generates ALARM_INT(alarm interrupt) and ALARM_WK(alarm wake-up) at a specified time in the power-down mode, power off mode or normal operation mode. In normal operation mode, If ALARM register value is a same to BCD register, ALARM_INT is activated as well as the ALARM_WK. In the power-off and power-down, If ALARM register value is a same to BCD register,ALARM_WK is activated. The RTC alarm register (RTCALM) determines the alarm enable/disable status and the condition of the alarm time setting.

33.3.5 TICK TIME INTERRUPT

The RTC tick time is used for interrupt request. The RTCCON[8] register has an interrupt enable bit. The count value reaches '0' when the tick time interrupt occurs. Then the period of interrupt is as follows:

Period = (n+1)/32768 second (n= tick counter value)

Table 33-1 Tick interrupt resolution

Tick counter clock source selection	Tick clock source frequency(Hz)	Clock range (s)	Resolution (ms)
4'b0000	32768 (2^15)	0 ~ 2	0.03
4'b0001	16384 (2^14)	0 ~ 4	0.06
4'b0010	8192 (2^13)	0 ~ 8	0.12
4'b0011	4096 (2^12)	0 ~ 16	0.24
4'b0100	2048 (2^11)	0 ~ 32	0.49
4'b0101	1024 (2^10)	0 ~ 64	0.97
4'b0110	512 (2^9)	0 ~ 128	1.95
4'b0111	256 (2^8)	0 ~ 256	3.90
4'b1000	128 (2^7)	0 ~ 512	7.81
4'b1001	64 (2^6)	0 ~ 1024	15.62
4'b1010	32 (2^5)	0 ~ 2048	31.25
4'b1011	16 (2^4)	0 ~ 4096	62.50
4'b1100	8 (2^3)	0 ~ 8192	125

4’b1101	4 (2^2)	0 ~ 16384	250
4’b1110	2	0 ~ 32768	500
4’b1111	1	0 ~ 65535	1000

NOTES:

1. Tick time resolution can be extended by selecting the appropriate tick time clock source.
2. 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 will always synchronized in real time.

33.3.6 32.768 KHZ X-TAL CONNECTION EXAMPLE

The Figure 33-2 shows a circuit of the RTC unit oscillation at 32.768 Khz.

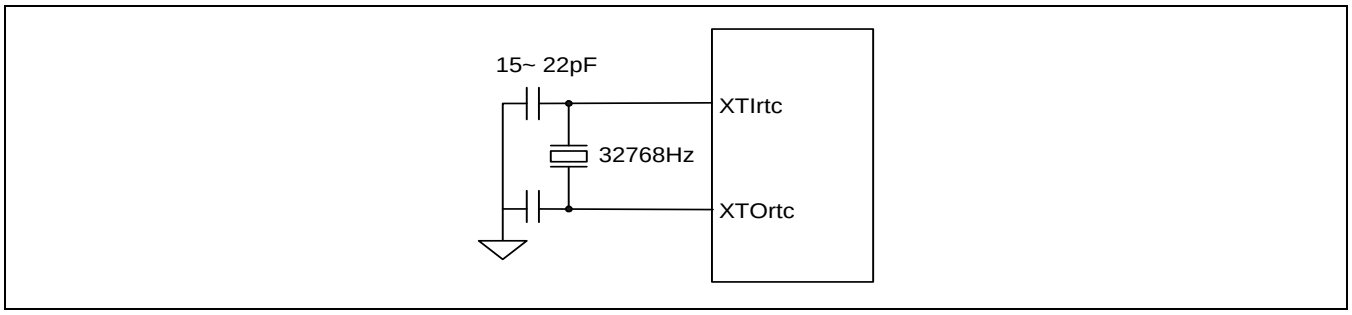


Figure 33-2. Main Oscillator Circuit Example

3.7 EXTERNAL INTERFACE

Name	Direction	Description
XTI	Input	32 KHz RTC Oscillator Clock Input
XTO	Input	32 KHz RTC Oscillator Clock output

33.8 REGISTER DESCRIPTION

33.8.1 MEMORY MAP

Table 33-2. RTC Register summary

Register	Address	R/W	Description	Reset Value
INTP	0x7E005030	R/W	Interrupt pending Register	0x0
RTCCON	0x7E005040	R/W	RTC control Register	0x0
TICCNT	0x7E005044	R/W	Tick time count Register	0x0
RTCALM	0x7E005050	R/W	RTC alarm control Register	0x0
ALMSEC	0x7E005054	R/W	Alarm second data Register	0x0
ALMMIN	0x7E005058	R/W	Alarm minute data Register	0x00
ALMHOUR	0x7E00505C	R/W	Alarm hour data Register	0x0
ALMDATE	0x7E005060	R/W	Alarm date data Register	0x01
ALMMON	0x7E005064	R/W	Alarm month data Register	0x01
ALMYEAR	0x7E005068	R/W	Alarm year data Register	0x0
BCDSEC	0x7E005070	R/W	BCD second Register	Undefined
BCDMIN	0x7E005074	R/W	BCD minute Register	Undefined
BCDHOUR	0x7E005078	R/W	BCD hour Register	Undefined
BCDDATE	0x7E00507C	R/W	BCD date Register	Undefined
BCDDAY	0x7E005080	R/W	BCD day Register	Undefined
BCDMON	0x7E005084	R/W	BCD month Register	Undefined
BCDYEAR	0x7E005088	R/W	BCD year Register	Undefined
CURTICCNT	0x7E005090	R	Current Tick time counter Register	0x0

33.9 INDIVIDUAL REGISTER DESCRIPTIONS

33.9.1 REAL TIME CLOCK CONTROL (RTCCON) REGISTER

The RTCCON register consists of 9 bits . It controls the read/write enable of the CLKSEL, CNTSEL and CLKRST for testing.
RTCEN bit can control all interfaces between the CPU and the RTC, Therefore it must be set to 1 in an RTC control routine to enable data read/write after a system reset. Before power off, the RTCEN bit is cleared to 0 to prevent inadvertent writing into BCD counter register.
CLKRST is counter reset for 2^{15} Clock divider.(reference to Figure 33-1)
Before RTC clock setting, 2^{15} Clock divider must be reset for exact RTC operation.

Register	Address	R/W	Description	Reset Value
RTCCON	0x7E005040	R/W	RTC control Register	0x0

RTCCON	Bit	Description	Initial State
TICEN	[8]	Tick timer enable 0 = Disable 1 = Enable	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 = RTC counter(2^{15} Clock divider) enable, 1 = RTC counter reset and disable. Note: When RTCEN is enable, CLKRST affects RTC.	0
CNTSEL	[2]	BCD count select. 0 = Merge BCD counters 1 = Reserved (Separate BCD counters) Note: When RTCEN is enable, CNTSEL affects RTC.	0
CLKSEL	[1]	BCD clock select. 0 = XTAL 1/2^{15} divided clock 1 = Reserved (XTAL clock only for test) Note: When RTCEN is enable, CLKSEL affects RTC.	0
RTCEN	[0]	RTC control enable. 0 = Disable 1 = Enable Note: When RTCEN is enable, BCD time count setting, 2^{15} Clock divider reset, BCD counter select and BCD clock selecting.	0

33.9.2 TICK TIME COUNT REGISTER (TICNT)

Register	Address	R/W	Description	Reset Value
TICNT	0x7E005044	R/W	Tick time count register	0x0

TICNT	Bit	Description	Initial State
TICK TIME COUNT	[31:0]	32 bit tick time count value	0x0

33.9.3 RTC ALARM CONTROL (RTCALM) REGISTER

The RTCALM register determines the alarm enable and the alarm time. Note that the RTCALM register generates the alarm signal through both ALMINT and ALARM_WK as power mode.
For using ALMIN and ALARM_WK, ALMEN must be enable.
If compare value is year, ALMEN and YEAREN must be enable.
If compare values are year,mon,date,hour,min and sec, ALMEN,YEAREN,MONEN,DATEEN,HOUREN,MINEN and SECEN must be enable.

Register	Address	R/W	Description	Reset Value
RTCALM	0x7E005050	R/W	RTC alarm control Register	0x0

RTCALM	Bit	Description	Initial State
Reserved	[7]		0
ALMEN	[6]	Alarm global enable 0 = Disable, 1 = Enable Note: For using ALMIN and ALARM_WK, set ALMEN=1'b1	0
YEAREN	[5]	Year alarm enable 0 = Disable, 1 = Enable	0
MONEN	[4]	Month alarm enable 0 = Disable, 1 = Enable	0
DATEEN	[3]	Date alarm enable 0 = Disable, 1 = Enable	0
HOUREN	[2]	Hour alarm enable 0 = Disable, 1 = Enable	0
MINEN	[1]	Minute alarm enable 0 = Disable, 1 = Enable	0
SECEN	[0]	Second alarm enable 0 = Disable, 1 = Enable	0

33.9.4 ALARM SECOND DATA (ALMSEC) REGISTER

Register	Address	R/W	Description	Reset Value
ALMSEC	0x7E005054	R/W	Alarm second data Register	0x0

ALMSEC	Bit	Description	Initial State
Reserved	[7]		0
SECDATA	[6:4]	BCD value for alarm second. 0 ~ 5	000
	[3:0]	0 ~ 9	0000

33.9.5 ALARM MIN DATA (ALMMIN) REGISTER

Register	Address	R/W	Description	Reset Value
ALMMIN	0x7E005058	R/W	Alarm minute data Register	0x00

ALMMIN	Bit	Description	Initial State
Reserved	[7]		0
MINDATA	[6:4]	BCD value for alarm minute. 0 ~ 5	000
	[3:0]	0 ~ 9	0000

33.9.6 ALARM HOUR DATA (ALMHOUR) REGISTER

Register	Address	R/W	Description	Reset Value
ALMHOUR	0x7E00505C	R/W	Alarm hour data Register	0x0

ALMHOUR	Bit	Description	Initial State
Reserved	[7:6]		00
HOURDATA	[5:4]	BCD value for alarm hour. 0 ~ 2	00
	[3:0]	0 ~ 9	0000

33.9.7 ALARM DATE DATA (ALMDATE) REGISTER

Register	Address	R/W	Description	Reset Value
ALMDATE	0x7E005060	R/W	Alarm day data Register	0x01

ALMDATE	Bit	Description	Initial State
Reserved	[7:6]		00
DATEDATA	[5:4]	BCD value for alarm date, from 0 to 28, 29, 30, 31. 0 ~ 3	00
	[3:0]	0 ~ 9	0001

33.9.8 ALARM MONTH DATA (ALMMON) REGISTER

Register	Address	R/W	Description	Reset Value
ALMMON	0x7E005064	R/W	Alarm month data Register	0x01

ALMMON	Bit	Description	Initial State
Reserved	[7:5]		00
MONDATA	[4]	BCD value for alarm month. 0 ~ 1	0
	[3:0]	0 ~ 9	0001

33.9.10 ALARM YEAR DATA (ALMYEAR) REGISTER

Register	Address	R/W	Description	Reset Value
ALMYEAR	0x7E005068	R/W	Alarm year data Register	0x0

ALMYEAR	Bit	Description	Initial State
YEARDATA	[7:4]	BCD value for year. 0 ~ 9	0x0
	[3:0]	0 ~ 9	0x0

33.9.11 BCD SECOND (BCDSEC) REGISTER

Register	Address	R/W	Description	Reset Value
BCDSEC	0x7E005070	R/W	BCD second Register	Undefined

BCDSEC	Bit	Description	Initial State
SECDATA	[6:4]	BCD value for second. 0 ~ 5	-
	[3:0]	0 ~ 9	-

33.9.12 BCD MINUTE (BCDMIN) REGISTER

Register	Address	R/W	Description	Reset Value
BCDMIN	0x7E005074	R/W	BCD minute Register	Undefined

BCDMIN	Bit	Description	Initial State
MINDATA	[6:4]	BCD value for minute. 0 ~ 5	-
	[3:0]	0 ~ 9	-

33.9.13 BCD HOUR(BCDHOUR) REGISTER

Register	Address	R/W	Description	Reset Value
BCDHOUR	0x7E005078	R/W	BCD hour Register	Undefined

BCDHOUR	Bit	Description	Initial State
Reserved	[7:6]		-
HOURDATA	[5:4]	BCD value for hour. 0 ~ 2	-
	[3:0]	0 ~ 9	-

33.9.14 BCD DATE (BCDDATE) REGISTER

Register	Address	R/W	Description	Reset Value
BCDDATE	0x7E00507C	R/W	BCD DATE Register	Undefined

BCDDAY	Bit	Description	Initial State
Reserved	[7:6]		-
DATEDATA	[5:4]	BCD value for date. 0 ~ 3	-
	[3:0]	0 ~ 9	-

33.9.15 BCD DAY (BCDDAY) REGISTER

Register	Address	R/W	Description	Reset Value
BCDDAY	0x7E005080	R/W	BCD DAY Register	Undefined

BCDDAY	Bit	Description	Initial State
Reserved	[7:3]		-
DAYDATA	[2:0]	BCD value for a day of the week. 1 ~ 7	-

33.9.16 BCD MONTH (BCDMON) REGISTER

Register	Address	R/W	Description	Reset Value
BCDMON	0x7E005084	R/W	BCD month Register	Undefined

BCDMON	Bit	Description	Initial State
Reserved	[7:5]		-
MONDATA	[4]	BCD value for month. 0 ~ 1	-
	[3:0]	0 ~ 9	-

33.9.17 BCD YEAR (BCDYEAR) REGISTER

Register	Address	R/W	Description	Reset Value
BCDYEAR	0x7E005088	R/W	BCD year Register	Undefined

BCDYEAR	Bit	Description	Initial State
YEARDATA	[7:4]	BCD value for year. 0 ~ 9	-
	[3:0]	0 ~ 9	-

NOTE: For setting BCD registers, RTCEN(RTCCON[0] bit) must be enable.
But at no setting BCD registers, RTCEN must be disable for reducing power comsumption.

33.9.18 TICK COUNTER REGISTER

Register	Address	R/W	Description	Reset Value
CURTICCNT	0x7E005090	R	Current tick count value	32'h0

CURTICCNT	Bit	Description	Initial State
Tick counter observation	[31:0]	Current tick count value	-

33.9.19 INTERRUPT PENDING REGISTER

Register	Address	R/W	Description	Reset Value
INTP	0x7E005030	R/W	Interrupt pending register	Undefined

You can clear specific bits of INTP register by writing 1's to the bits that you want to clear regardless of the value of RTCEN.

INTP	Bit	Description	Initial State
Reserved	[7:2]	Reserved.	00
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

34 WATCHDOG TIMER

This chapter describes the functions and usage of Watchdog Timer in S3C6410X RISC microprocessor.

34.1 OVERVIEW

The S3C6410X RISC microprocessor watchdog timer is used to resume the controller operation whenever it is disturbed by malfunctions such as noise and system errors. The watchdog timer generates the reset signal. It can be used as a normal 16-bit interval timer to request interrupt service.

Advantage in using WDT instead of PWM timer is that WDT generates the reset signal.

34.2 FEATURES

The Watchdog Timer includes the following features:

- Normal interval timer mode with interrupt request.
- Internal reset signal is activated when the timer count value reaches 0 (time-out).
- Level-triggered Interrupt mechanism.

34.3 FUNCTIONAL DESCRIPTION

34.3.1 WATCHDOG TIMER OPERATION

Figure 34-1 shows the functional block diagram of the watchdog timer. The watchdog timer uses only PCLK as its source clock. The PCLK frequency is prescaled to generate the corresponding watchdog timer clock, and the resulting frequency is divided again.

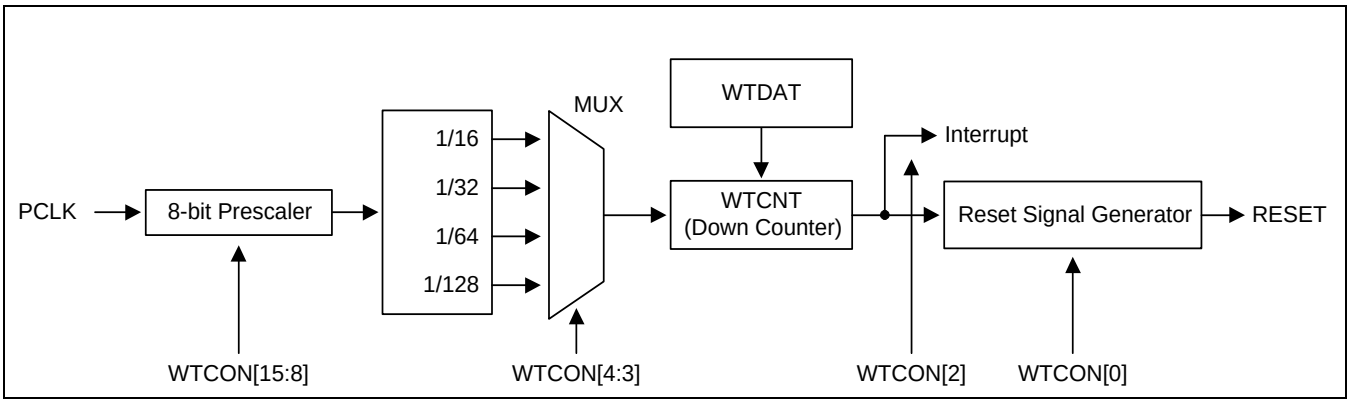


Figure 34-1. Watchdog Timer Block Diagram

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)

34.3.2 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.

34.3.3 CONSIDERATION OF DEBUGGING ENVIRONMENT

When the S3C6410 is in debug mode using Embedded ICE, the watchdog timer must not operate.

The watchdog timer can determine whether or not 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.

34.4 SPECIAL FUNCTION REGISTER

34.4.1 MEMORY MAP

Register	Address	R/W	Description	Reset Value
WTCON	0x7E004000	R/W	Watchdog timer control register	0x8021
WTDAT	0x7E004004	R/W	Watchdog timer data register	0x8000
WTCNT	0x7E004008	R/W	Watchdog timer count register	0x8000
WTCLRINT	0x7E00400C	W	Watchdog timer interrupt clear register	-

34.5 INDIVIDUAL REGISTER DESCRIPTION

34.5.1 WATCHDOG TIMER CONTROL (WTCON) REGISTER

The WTCON register allows the user 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 S3C6410 restart on mal-function after its power on. At this time, disable the interrupt generation and enable the Watchdog timer output for reset signal.

If controller restart is not desired and if the user wants to use the normal timer only, which is provided by the Watchdog timer, enable the interrupt generation and disable the Watchdog timer output for reset signal.

Register	Address	R/W	Description	Reset Value
WTCON	0x7E004000	R/W	Watchdog timer control register	0x8021

WTCON	Bit	Description	Initial State
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 bit of Watchdog timer. 0 = Disable 1 = Enable	1
Clock select	[4:3]	Determine the clock division factor. 00: 16 01 : 32 10: 64 11 : 128	00
Interrupt generation	[2]	Enable or disable bit of the interrupt. 0 = Disable 1 = Enable	0
Reserved	[1]	Reserved. This bit must be 0 in normal operation.	0
Reset enable/disable	[0]	Enable or disable bit of Watchdog timer output for reset signal. 1: Assert reset signal of the S3C6410 at watchdog time-out 0: Disable the reset function of the watchdog timer.	1

Note: Initial state of 'Reset enable/disable' is 1(reset enable). If user do not disable this bit, S3C6410 will be rebooted in about 5.63sec (In the case of PCLK is 12MHz). So at boot loader, this bit should be disabled before under control of Operating System, or Firmware.

34.5.2 WATCHDOG TIMER DATA (WTDAT) REGISTER

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 of WTCNT) will drive the first time-out. Then, the value of WTDAT will be automatically reloaded into WTCNT.

Register	Address	R/W	Description	Reset Value
WTDAT	0x7E004004	R/W	Watchdog timer data register	0x8000

WTDAT	Bit	Description	Initial State
Count reload value	[15:0]	Watchdog timer count value for reload.	0x8000

34.5.3 WATCHDOG TIMER COUNT (WTCNT) REGISTER

The WTCNT register contains the current count values for the watchdog timer during normal operation.

NOTE:

The content of the WTDAT register cannot be automatically loaded into the timer count register when the watchdog timer is enabled initially, so the WTCNT register must be set to an initial value before enabling it.

Register	Address	R/W	Description	Reset Value
WTCNT	0x7E004008	R/W	Watchdog timer count register	0x8000

WTCNT	Bit	Description	Initial State
Count value	[15:0]	The current count value of the watchdog timer	0x8000

34.5.4 WATCHDOG TIMER INTERRUPT CLEAR (WTCLRINT) REGISTER

The WTCLRINT register is used to clear the interrupt. Interrupt service routine is responsible for clearing the relevant interrupt after the interrupt service is completed. Writing any values on this register clears the interrupt. Reading this register is not allowed.

Register	Address	R/W	Description	Reset Value
WTCLRINT	0x7E00400C	W	Watchdog timer interrupt clear register	-

WTCLRINT	Bit	Description	Initial State
Interrupt clear	[0]	Write any values clears the interrupt	-

35 AC97 CONTROLLER

This chapter describes the functions and usage of AC97 Controller in SC6410X66 RISC microprocessor.

35.1 OVERVIEW

The AC97 Controller Unit of the S3C6410 supports the AC97 revision 2.0 features. AC97 Controller communicates with AC97 Codec using audio controller link (AC-link). 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. Controller receives the stereo PCM data and the mono Mic data from Codec then store in memories. This chapter describes the programming model for the AC97 Controller Unit. The prerequisite in this chapter requires an understanding of the AC97 revision 2.0 specifications.

35.2 FEATURES

The AC97 Controller includes the following features:

- Independent channels for stereo PCM In(Slot3, Slot4), mono MIC In(Slot 6), stereo PCM Out(Slot3, Slot4).
- DMA-based operation and interrupt based operation.
- All of the channels support only 16-bit samples.
- Variable sampling rate AC97 Codec interface (48KHz and below)
- 16-bit, 16 depth FIFOs per channel
- Only primary Codec support

35.3 SIGNALS

Name	Direction	Description
X97RESETn	Output	AC_RESETn : Active-low CODEC reset.
X97BITCLK	Input	AC_BIT_CLK : 12.288MHz bit-rate clock
X97SYNC	Output	AC_SYNC : 48 kHz frame indicator and synchronizer
X97SDO	Output	AC_SDO : Serial audio output data.
X97SDI	Input	AC_SDI : Serial audio input data.

35.4 AC97 CONTROLLER OPERATION

This section explains the AC97 Controller operation such as AC-Link, Power-down sequence and Wake-up sequence.

35.4.1 BLOCK DIAGRAM

Figure 35-1 shows the functional block diagram of S3C6410 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 over the AC-link.

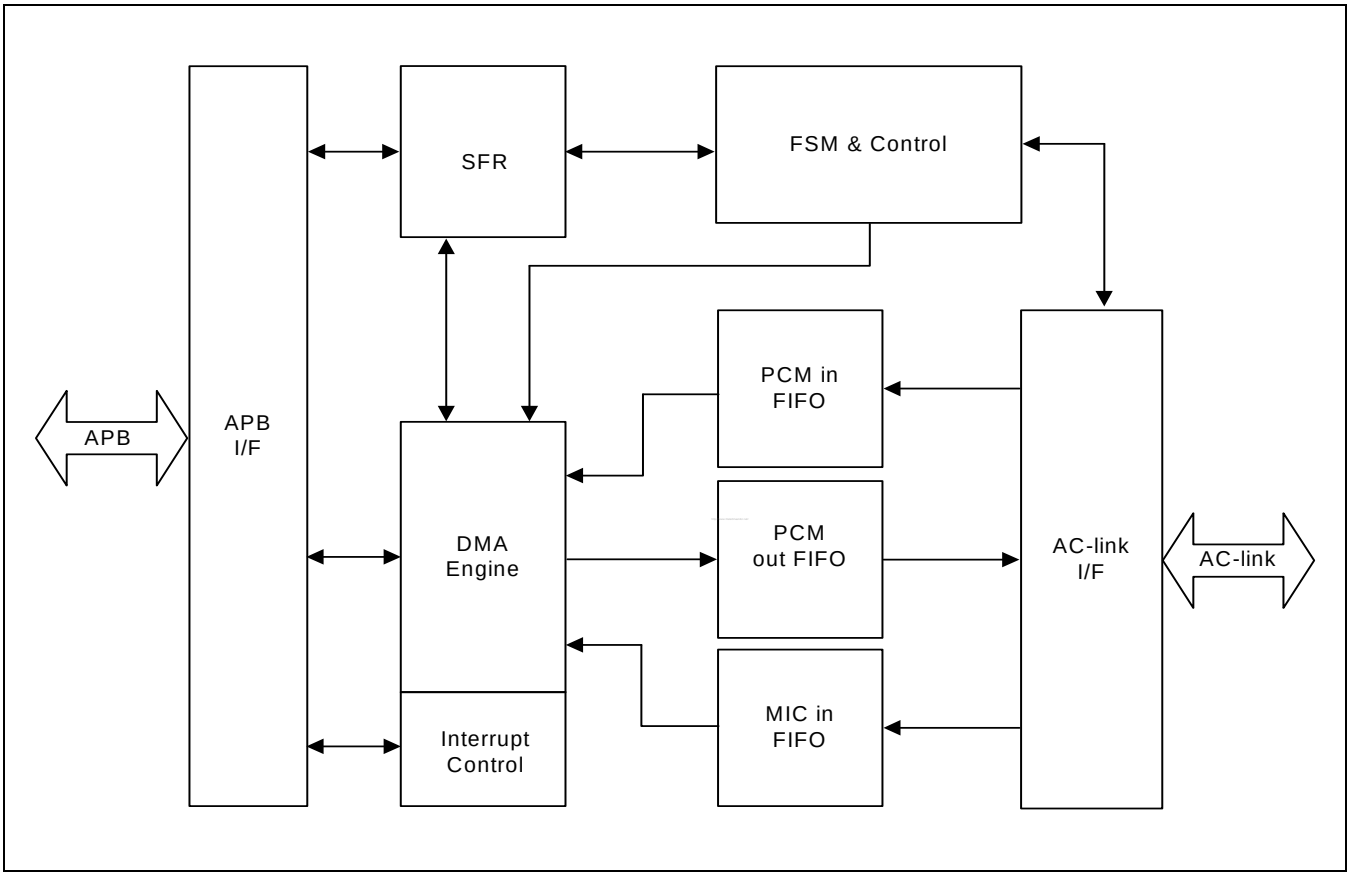


Figure 35-1. AC97 Block Diagram

35.4.2 INTERNAL DATA PATH

Figure 35-2 shows the internal data path of S3C6410 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.

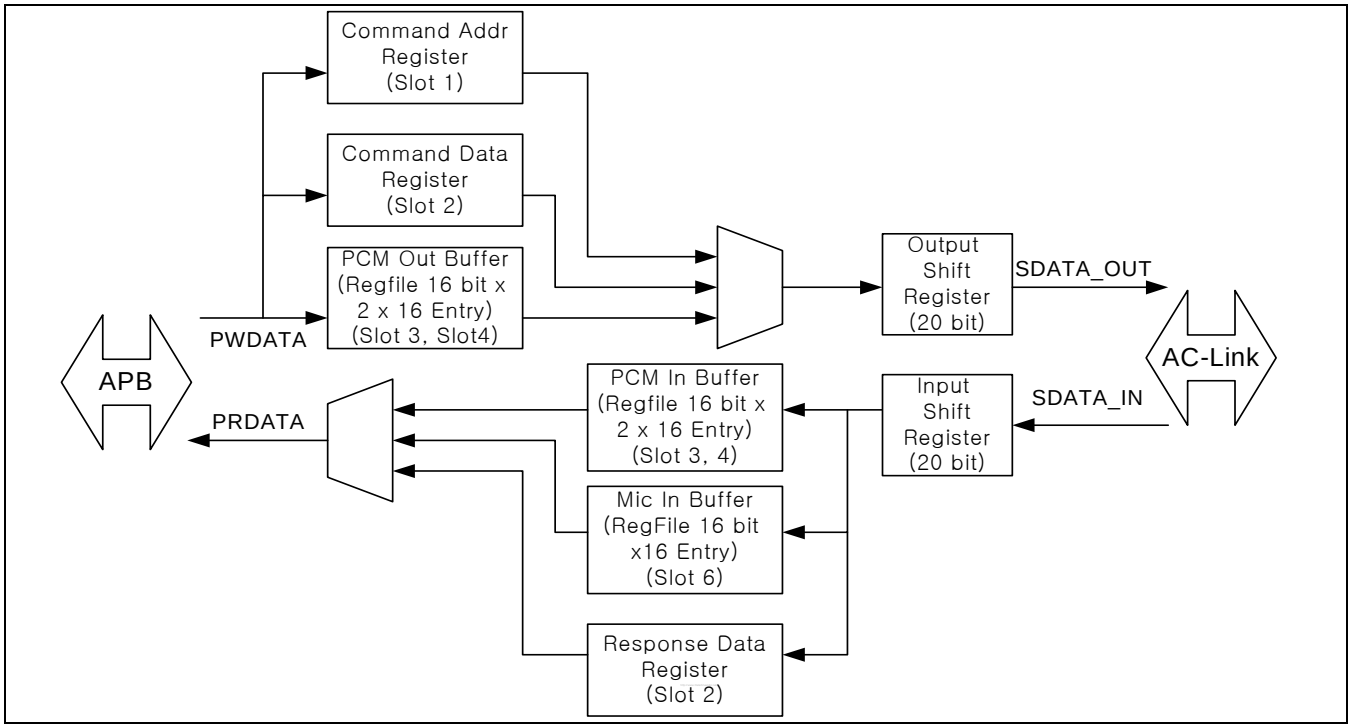


Figure 35-2. Internal Data Path

35.4.3 OPERATION FLOW CHART

When you initialize the AC97 controller, you must assert system reset or cold reset, because we don't know the previous state of the external AC97 audio-codec. This assures that GPIO is already ready. Then you enable the codec ready interrupt. You can check codec ready interrupt by polling or interrupt. When interrupt is occurred, you must de-assert codec ready interrupt. You can 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.

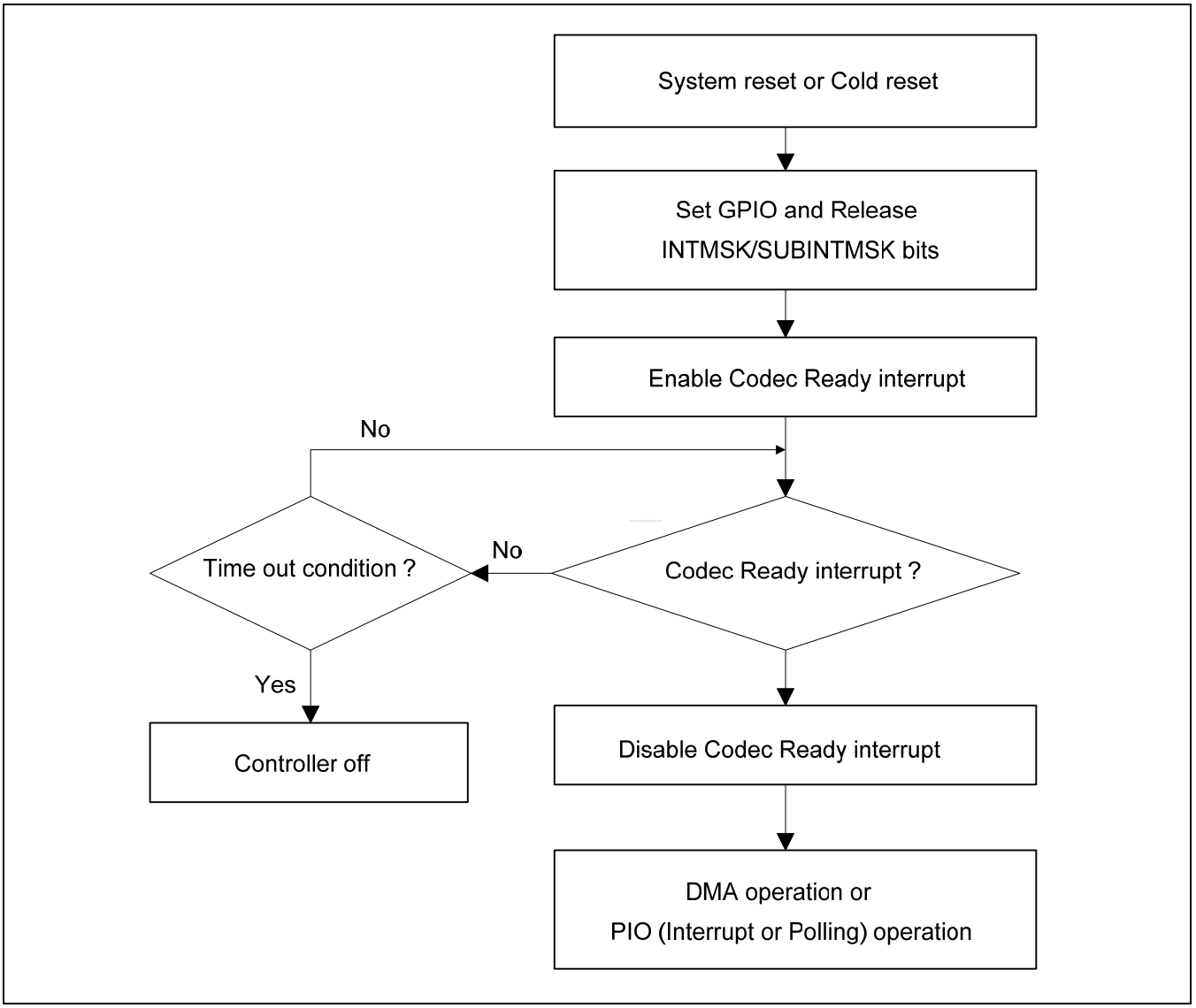


Figure 35-3. AC97 Operation Flow Chart

35.4.4 AC-LINK DIGITAL INTERFACE PROTOCOL

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the S3C6410 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.

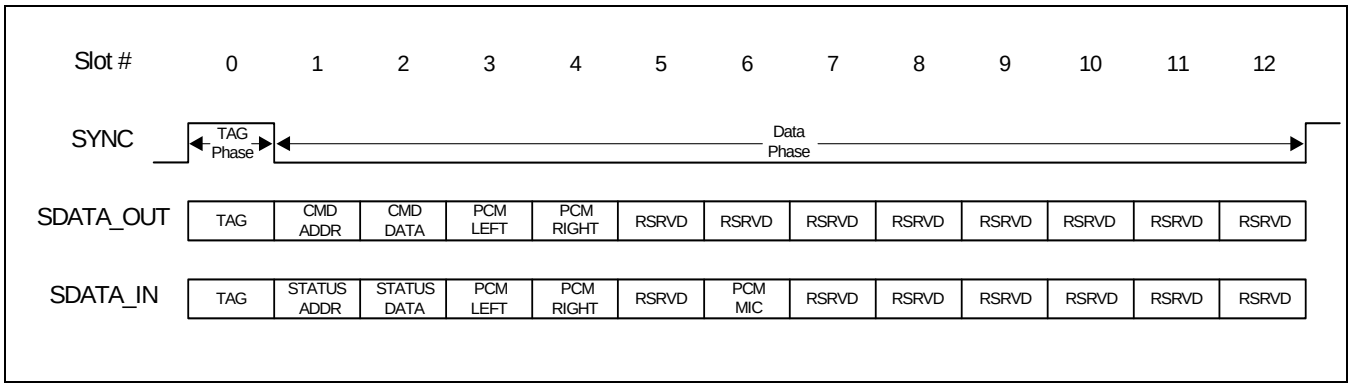


Figure 35-4. Bi-directional AC-link Frame with Slot Assignments

Figure 35-4 shows the slot definitions supported by S3C6410 AC97 Controller. The S3C6410 AC97 Controller provides synchronization for all data transaction on the AC-link.

A data transaction is made up of 256 bits of information broken up 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.

35.4.5 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 times 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 can be transmitted across AC-link at its fixed 48kHz 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. When software accesses the primary CODEC, the hardware configures the frame as follows :

- In slot 0, the valid bit for 1, 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.

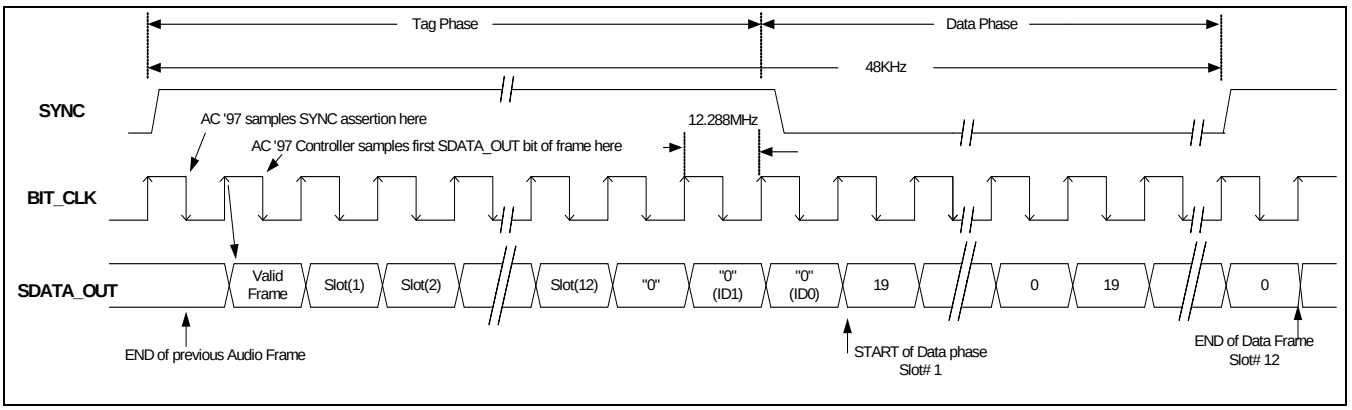


Figure 35-5. AC-link Output Frame

35.4.6 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 indicate 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 35-1. 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 Record Right channel

Slot 4, which is audio input frame is the right 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 6: Microphone Record Data

The AC97 Controller only supports 16-bit resolution for the MIC-in channel.

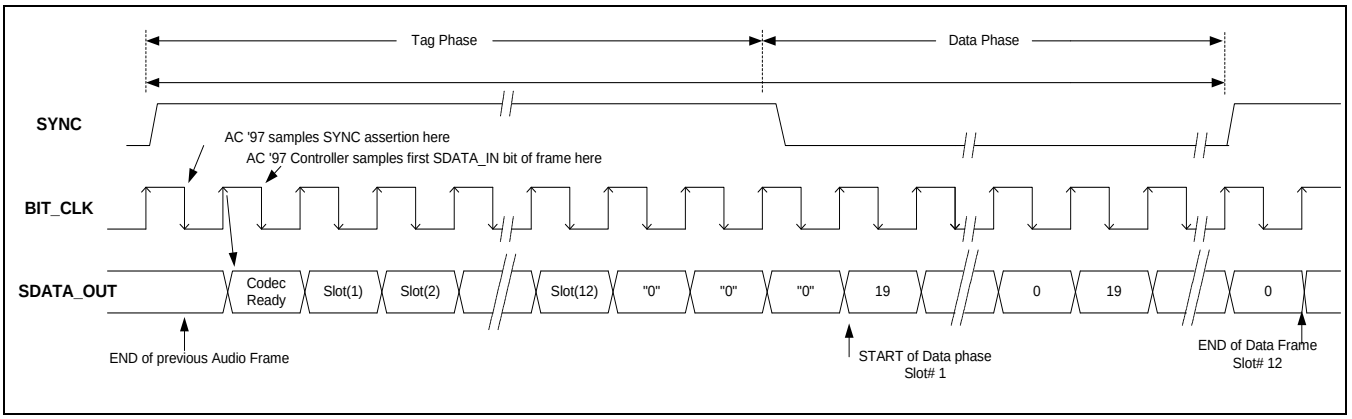


Figure 35-6. AC-link Input Frame

35.4.7 AC97 POWER-DOWN

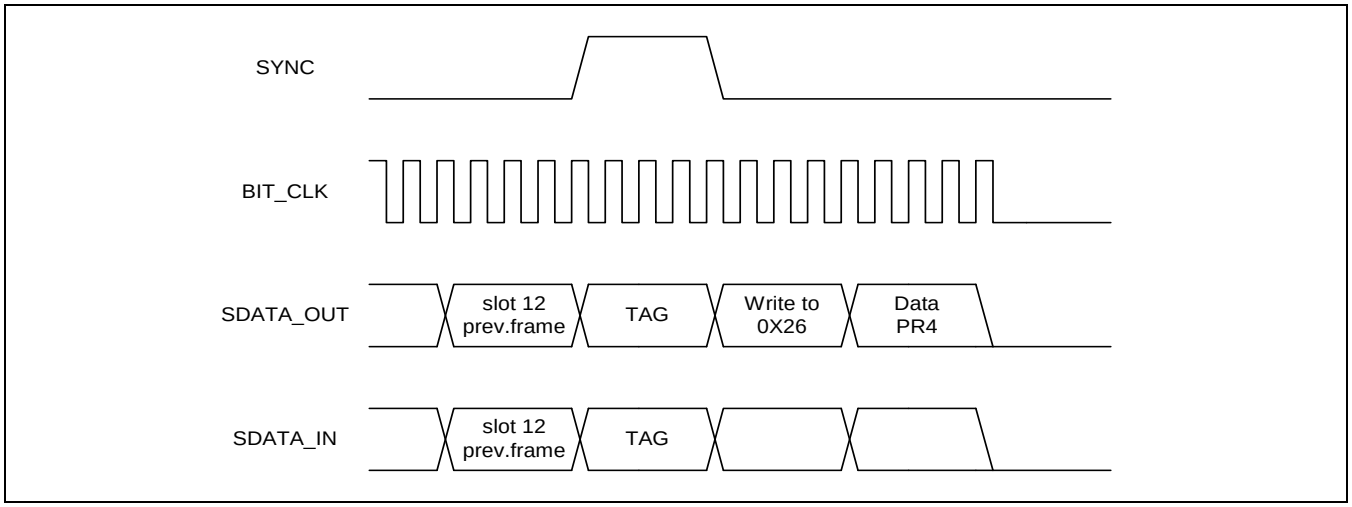


Figure 35-7. AC97 Power-down Timing

Powering Down the AC-link

The AC-link signals enter a low power mode when 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 35-7.

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. When 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.

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. When AC-link powers up, it indicates readiness through the Codec ready bit (input slot 0, bit 15).

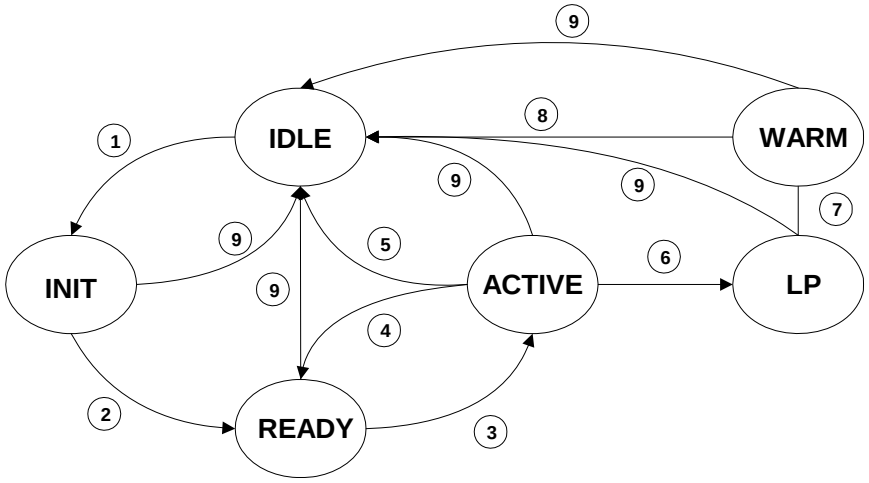
Cold AC97 Reset

A cold reset is generated when 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.

Warm AC97 Reset

A Warm AC97 reset reactivates the AC-link without altering the current AC97 register values. A warm reset is generated when BITCLK is absent and SYNC is driven high. In normal audio frames, SYNC is a synchronous AC97 input. When 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.

35.4.8 AC97 STATE DIAGRAM



1 : PCLK rising

2 : ACLINK_ON

3 : CODEC_READY & TRANS_DATA & NORMAL_SYNC

4 : ~CODEC_READY | ~TRANS_DATA

5 : !ACLINK_ON

6 : POWER_DOWN

7 : WARM_RESET

8 : CODEC_WAKEUP

9 : COLD_RESET | ~PRESETn

Figure 35-8. 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.

35.5 AC97 CONTROLLER SPECIAL REGISTERS

AC97 Special function register summary

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0x7F001000	R/W	AC97 Global Control Register	0x00000000
AC_GLBSTAT	0x7F001004	R	AC97 Global Status Register	0x00000001
AC_CODEC_CMD	0x7F001008	R/W	AC97 Codec Command Register	0x00000000
AC_CODEC_STAT	0x7F00100C	R	AC97 Codec Status Register	0x00000000
AC_PCMADDR	0x7F001010	R	AC97 PCM Out/In Channel FIFO Address Register	0x00000000
AC_MICADDR	0x7F001014	R	AC97 Mic In Channel FIFO Address Register	0x00000000
AC_PCMDATA	0x7F001018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000
AC_MICDATA	0x7F00101C	R	AC97 MIC In Channel FIFO Data Register	0x00000000

35.5.1 AC97 GLOBAL CONTROL REGISTER (AC_GLBCTRL)

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.

Register	Address	R/W	Description	Reset Value
AC_GLBCTRL	0x7F001000	R/W	AC97 Global Control Register	0x000000

AC_GLBCTRL	Bit	Description	Initial State
-	[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
-	[23]	Reserved	0
Codec ready interrupt enable	[22]	0 : Disable 1 : Enable	0
PCM out channel underrun interrupt enable	[21]	0 : Disable 1 : Enable (FIFO is empty)	0

AC_GLBCTRL	Bit	Description	Initial State
PCM in channel overrun interrupt enable	[20]	0 : Disable 1 : Enable (FIFO is full)	0
Mic in channel overrun interrupt enable	[19]	0 : Disable 1 : Enable (FIFO is full)	0
PCM out channel threshold interrupt enable	[18]	0 : Disable 1 : Enable (FIFO is half empty)	0
PCM in channel threshold interrupt enable	[17]	0 : Disable 1 : Enable (FIFO is half full)	0
MIC in channel threshold interrupt enable	[16]	0 : Disable 1 : Enable (FIFO is half full)	0
-	[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
-	[7:4]	Reserved.	0000
Transfer data enable using AC-link	[3]	0 : Disable 1 : Enable	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 Registers	0

35.5.2 AC97 GLOBAL STATUS REGISTER (AC_GLBSTAT)

This is the status register. When the interrupt is occurs, you can check what the interrupt source is.

Register	Address	R/W	Description	Reset Value
AC_GLBSTAT	0x7F001004	R	AC97 Global Status Register	0x00000001

AC_GLBSTAT	Bit	Description	Initial State
-	[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
-	[15:3]	Reserved.	0x000
Controller main state	[2:0]	000 : Idle 001 : Init 010 : Ready 011 : Active 100 : LP 101 : Warm	001

35.5.3 AC97 CODEC COMMAND REGISTER (AC_CODEC_CMD)

When you control writing or reading, you must set the Read enable bit. If you want to write data to the AC97 Codec, you set the index (or address) of the AC97 Codec and data.

Register	Address	R/W	Description	Reset Value
AC_CODEC_CMD	0x7F001008	R/W	AC97 Codec Command Register	0x00000000

AC_CODEC_CMD	Bit	Description	Initial State
-	[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: When the commands are written on the AC_CODDEC_CMD register, It is recommended to have the delay time between the command and the next command is more than 1 / 48kHz.

35.5.4 AC97 CODEC STATUS REGISTER (AC_CODEC_STAT)

If the Read enable bit is 1 and Codec command address is valid, Codec status data is also valid.

Register	Address	R/W	Description	Reset Value
AC_CODEC_STAT	0x7F00100C	R	AC97 Codec Status Register	0x00000000

AC_CODEC_STAT	Bit	Description	Initial State
-	[31:23]	Reserved.	0x00
Address	[22:16]	Codec status address	0x00
Data	[15:0]	Codec status data	0x0000

NOTES: If you want to read data from AC97 codec register via the AC_CODDEC_STAT register, you must follow the following steps.

1. Write command address and data on the AC_CODEC_CMD register with Bit[23] =1.
2. Have a proper delay. It depends on Codec type
3. Read command address and data from AC_CODEC_STAT register.

35.5.5 AC97 PCM OUT/IN CHANNEL FIFO ADDRESS REGISTER (AC_PCMADDR)

To index the internal PCM FIFOs address.

Register	Address	R/W	Description	Reset Value
AC_PCMADDR	0x7F001010	R	AC97 PCM Out/In Channel FIFO Address Register	0x00000000

AC_PCMADDR	Bit	Description	Initial State
-	[31:28]	Reserved.	0000
Out read address	[27:24]	PCM out channel FIFO read address	0000
-	[23:20]	Reserved.	0000
In read address	[19:16]	PCM in channel FIFO read address	0000
-	[15:12]	Reserved.	0000
Out write address	[11:8]	PCM out channel FIFO write address	0000
-	[7:4]	Reserved.	0000
In write address	[3:0]	PCM in channel FIFO write address	0000

35.5.6 AC97 MIC IN CHANNEL FIFO ADDRESS REGISTER (AC_MICADDR)

To index the internal MIC-in FIFO address.

Register	Address	R/W	Description	Reset Value
AC_MICADDR	0x7F001014	R	AC97 MIC In Channel FIFO Address Register	0x00000000

AC_MICADDR	Bit	Description	Initial State
-	[31:20]	Reserved.	0000
Read address	[19:16]	MIC in channel FIFO read address	0000
-	[15:4]	Reserved.	0x000
Write address	[3:0]	MIC in channel FIFO write address	0000

35.5.7 AC97 PCM OUT/IN CHANNEL FIFO DATA REGISTER (AC_PCMDATA)

This is PCM out/in channel FIFO data register.

Register	Address	R/W	Description	Reset Value
AC_PCMDATA	0x7F001018	R/W	AC97 PCM Out/In Channel FIFO Data Register	0x00000000

AC_PCMDATA	Bit	Description	Initial State
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

35.5.8 AC97 MIC IN CHANNEL FIFO DATA REGISTER (AC_MICDATA)

This is MIC-in channel FIFO data register.

Register	Address	R/W	Description	Reset Value
AC_MICDATA	0x7F00101C	R	AC97 MIC In Channel FIFO Data Register	0x00000000

AC_MICDATA	Bit	Description	Initial State
-	[31:16]	Reserved	0x0000
Mono data	[15:0]	MIC in mono channel FIFO data	0x0000

36 IIS-BUS INTERFACE

36.1 OVERVIEW

IIS (Inter-IC Sound) is one of the popular digital audio interface. The bus only handles audio data, while the other signals, such as sub-coding and control, are transferred separately. It is possible to transmit data between two IIS bus. To minimize the number of pins required and to keep wiring simple, a 3-line serial bus is used consisting of a line for two time-multiplexed data channels, a word select line and a clock line..

IIS interface transmits or receives sound data from external stereo audio codec. For transmitting and receiving data, two 16x32bit FIFOs (First-In-First-Out) data structures are included. DMA transfer mode for transmitting or receiving samples can be supported.

36.2 FEATURE

The IIS-BUS interface includes the following features:

- 2-ports stereo IIS bus for audio interface with DMA-based operation
- Serial, 8/16/24 bit per channel data transfers
- Supports sampling rate from 8kHz to 192kHz
- Supports IIS, MSB-justified and LSB-justified data format
- 64 Bytes Tx FIFO/64 Bytes Rx FIFO per each port

36.3 SIGNAL DESCRIPTIONS

IIS external pads are shared with other IPs like PCM, AC97 and etc. In order to use these pads for IIS, GPIO must be set before the IIS started. For more information, refer to the GPIO chapter of this manual for proper GPIO setting.

Name	Type	Source/Destination	Description
Xi2sCLK[0]	Input/Output	Pad	IIS-bus0 serial clock(bit clock)
Xi2sCDCLK[0]	Input/Output	Pad	IIS0 Codec system clock or IISEXTCLK source
Xi2sLRCK[0]	Input/Output	Pad	IIS-bus0 channel select(word select) clock
Xi2sDI[0]	Input	Pad	IIS-bus0 serial data input
Xi2sDO[0]	Output	Pad	IIS-bus0 serial data output
Xi2sCLK[1]	Input/Output	Pad	IIS-bus1 serial clock(bit clock)
Xi2sCDCLK[1]	Input/Output	Pad	IIS1 Codec system clock or IISEXTCLK source
Xi2sLRCK[1]	Input/Output	Pad	IIS-bus1 channel select(word select) clock
Xi2sDI[1]	Input	Pad	IIS-bus1 serial data input
Xi2sDO[1]	Output	Pad	IIS-bus1 serial data output

36.4 BLOCK DIAGRAM

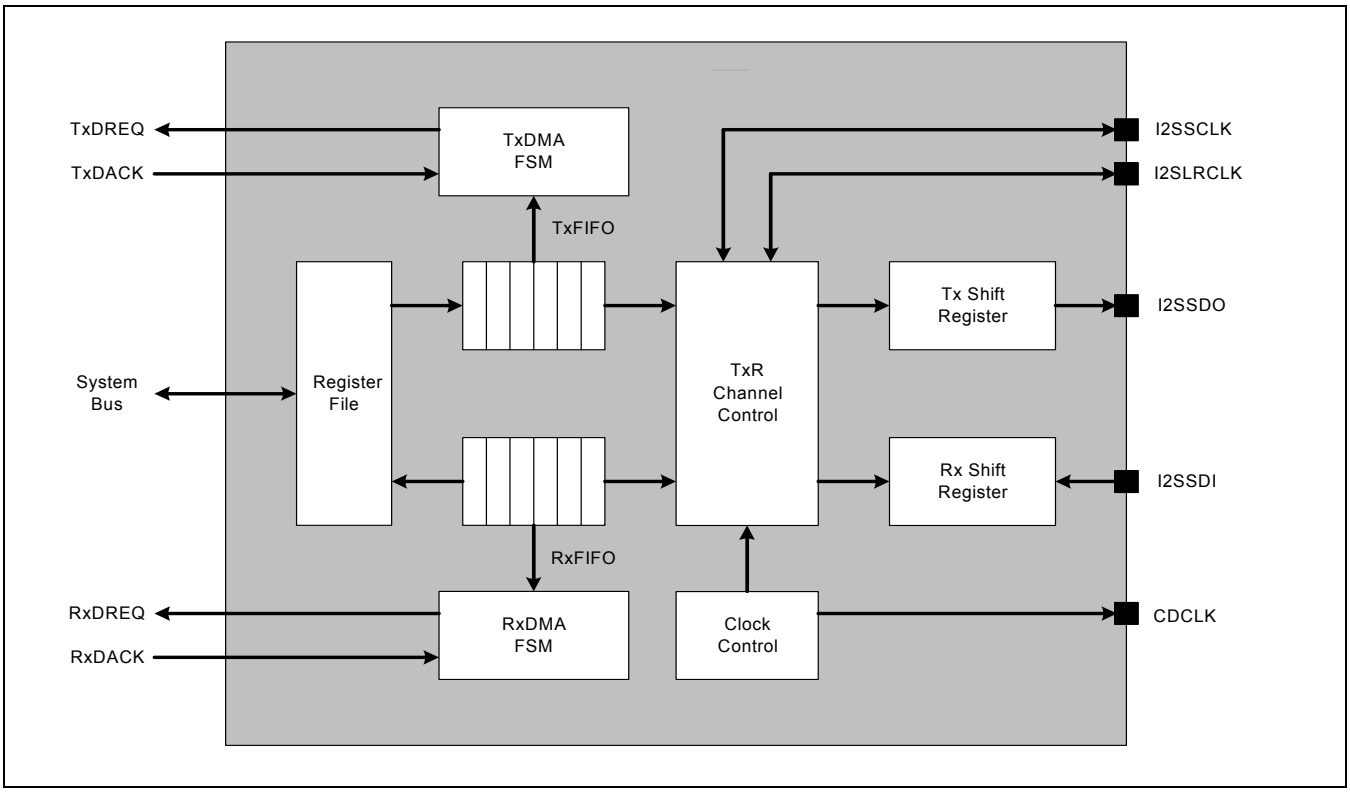


Figure 36-1. IIS-Bus Block Diagram

36.5 FUNCTIONAL DESCRIPTIONS

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as shown in Figure 36-1. Note that each FIFO has 32-bit width and 16 depths structure, which contains left/right channel data. Therefore FIFO access and data transfer are handled with left/right pair unit. Figure 36-1 shows the functional block diagram of IIS interface.

36.5.1 MASTER/SLAVE MODE

Master or slave mode can be selected by setting IMS bit of IISMOD register. In master mode, I2SSCLK(serial clock or bit clock) and I2SLRCLK(word select or channel select) are generated internally and supplied to external device. Therefore a I2SCDCLK(System clock) is needed for generating I2SSCLK and I2SLRCLK by dividing. The IIS pre-scaler (clock divider) is employed for generating a I2SCDCLK with divided frequency from internal system clock. In external master mode, the I2SCDCLK can be fed from IIS external. 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. Direction of I2SCDCLK (This is only auxiliary.) is not important. If IIS bus interface transmits clock signals to IIS codec, IIS bus is in master mode. But if IIS bus interface receives clock signal from IIS codec, IIS bus is in slave mode. TX/RX mode indicates the direction of data flow. If IIS bus interface transmits data to IIS codec, this is TX mode. Conversely, IIS bus interface receives data from IIS codec that is RX mode. Let's distinguish Master/Slave mode from TX/RX mode.

Figure 36-2 shows the route of the I2SCDCLK with internal master or external master mode setting in IIS clock control block and system controller. Note that RCLK indicates root clock and this clock can be supplied to external IIS codec chip at internal master mode.

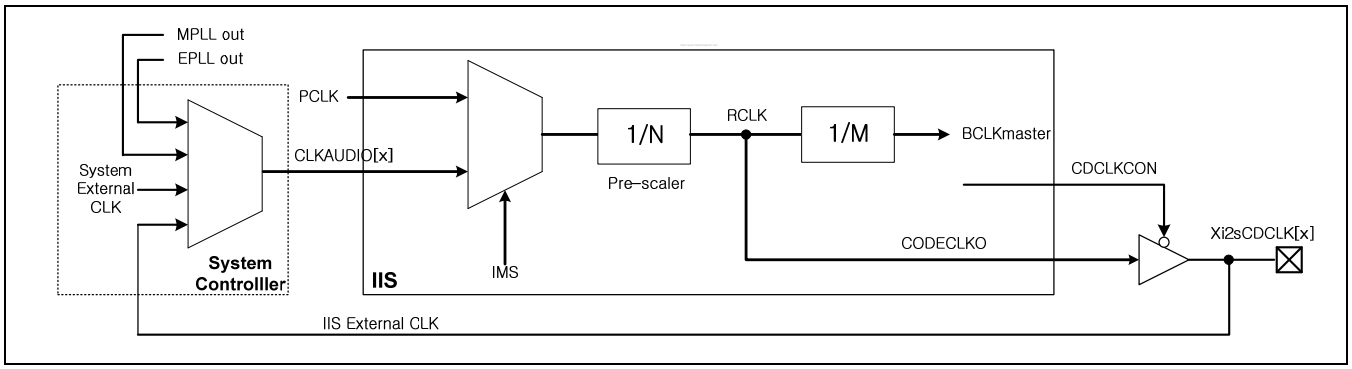


Figure 36-2. IIS Clock Control Block Diagram

36.5.2 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 when TXFIFO is not empty and the receiver DMA service request is activated when 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 must be performed.

* DMA request point

– TX mode : (FIFO is not full) & (TXDMACTIVE is active)

– RX mode : (FIFO is not empty) & (RXDMACTIVE is active)

* Note : It only supports single transfer in DMA mode.

36.6 AUDIO SERIAL DATA FORMAT

36.6.1 IIS-BUS FORMAT

The IIS 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 transmitting data that is synchronized with the leading edge has some restrictions.

The LR channel select line indicates the channel being transmitted. I2SLRCLK may be changed either on a trailing or leading edge of the serial clock, but it is not mandatory 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 clear the input for the next word.

36.6.2 MSB (LEFT) JUSTIFIED

MSB-Justified (Left-Justified) format is similar to IIS 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.

36.6.3 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 36-3 shows the audio serial format of IIS, 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).

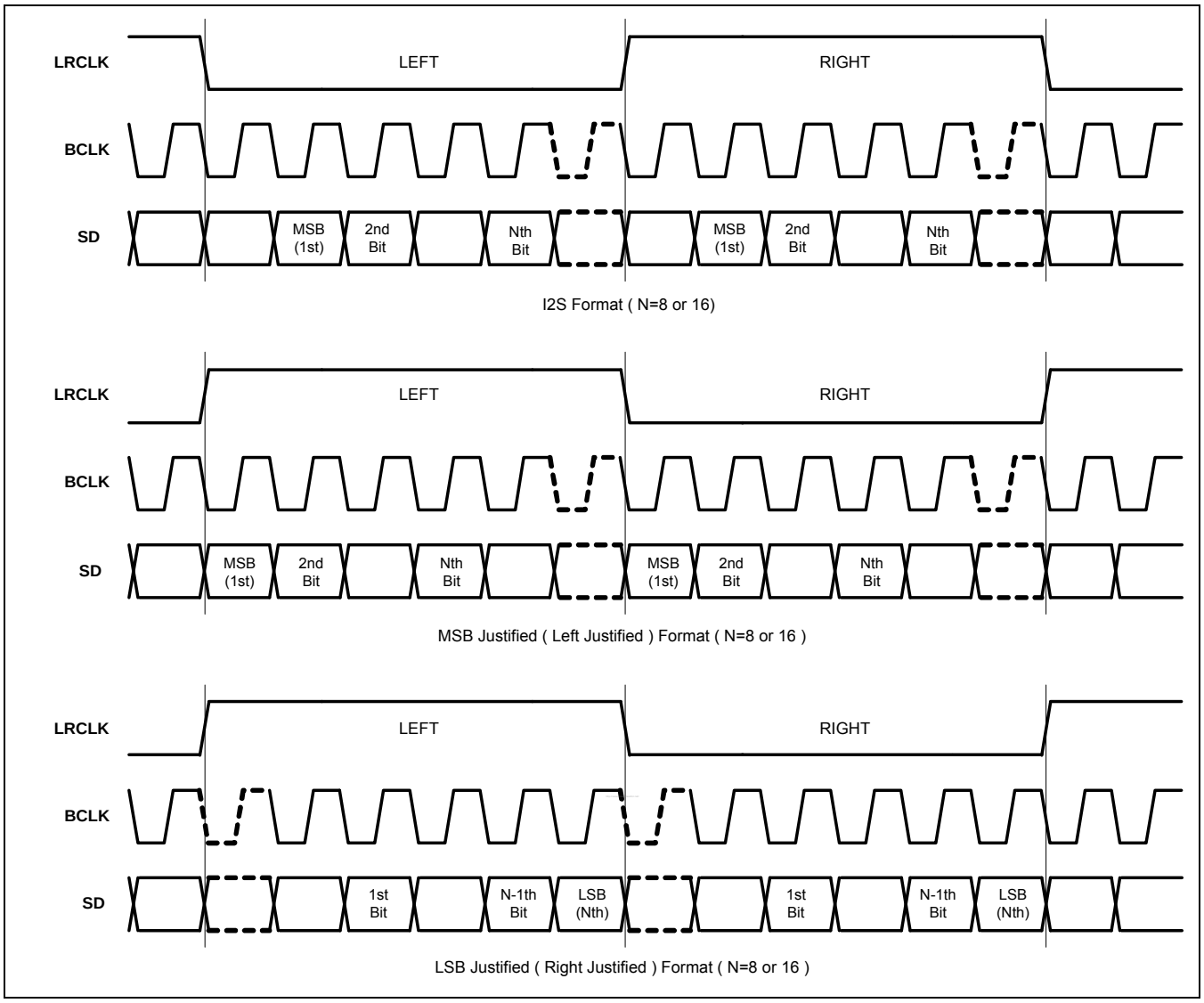


Figure 36-3. IIS Audio Serial Data Formats

36.7 SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (RCLK) can be selected by sampling frequency as shown in Table 36-1. Because RCLK is made by IIS pre-scaler, the pre-scaler value and RCLK type (256fs or 384fs or 512fs or 768fs) must be determined properly.

Table 36-1. CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)

IISLRCK (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	-	-

Note : fs represents sampling frequency.

CODECLK Frequency = fs * (256 or 384 or 512 or 768)

36.8 IIS CLOCK MAPPING TABLE

On selecting BFS, RFS, and BLC bits of I2SMOD register, you must refer to the following table. Table 36-2 shows the allowable clock frequency mapping relations.

Table 36-2. IIS 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 when BLC is 8-bit (b) Allowed when BLC is 16-bit (c) Allowed when BLC is 24-bit			

Note : Bit Clock Frequency ≥ fs * (bit length * 2). The codec clock is a multiple of the bit clock.

36.9 PROGRAMMING GUIDE

The IIS bus interface can be accessed either by the processor using programmed I/O instructions or by the DMA controller.

36.9.1 INITIALIZATION

1. Before you use IIS bus interface, you have to configure GPIOs to IIS mode and check signal's direction. I2SLRCLK, I2SSCLK and I2SCDCLK is inout-type. The each of I2SSDI and I2SSDO is a input and output.
2. You must select a clock source. S3C6410 has five clock sources. Those are MPLL, EPLL, PCLK, system external clock and IIS external clock. For more information please refer to Figure 36-2.

36.9.2 PLAY MODE (TX MODE) WITH DMA

1. TXFIFO is flushed before operation. If you don't distinguish Master/Slave mode from TX/RX mode, you must study Master/Slave mode and TX/RX mode. For more information please refer to Master/Slave chapter.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register) correctly.
3. To operate system in stability, the internal TXFIFO must be nearly full before transmission. DMA starts because of this reason.
4. IIS bus doesn't support the interrupt. You can only check the state by polling through accessing SFR.
5. If TXFIFO is full, you make I2SACTIVE be asserted.

36.9.3 RECORDING MODE (RX MODE) WITH DMA

1. RXFIFO is flushed before operation. If you don't distinguish between Master/Slave mode and TX/RX mode, you must study Master/Slave mode and TX/RX mode. For more information please refer to Master/Slave chapter.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register) correctly.
3. To operate system in stability, the internal RXFIFO must have at least one data before DMA operation. Because of this reason, you make I2SACTIVE be asserted.
4. Check RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, start the RXDMACTIVE.

36.9.4 EXAMPLE CODE

TX CHANNEL

The I2S TX channel provides a single stereo compliant output. The transmit channel can operate 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 can write upto 16 left/right data samples After enabling the channel for transmission.
An Example sequence is as the following.
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**

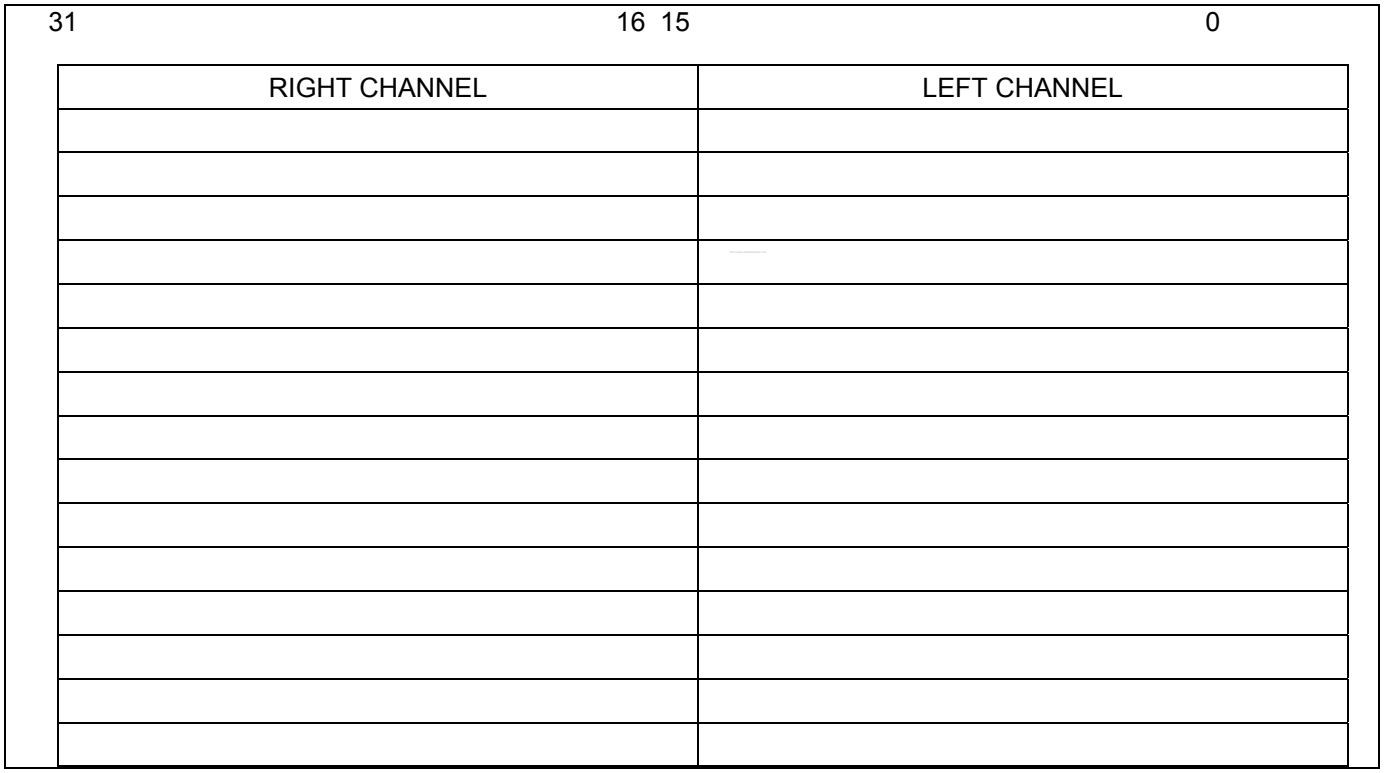


Figure 36-4. TX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the TX FIFO for 24-bits/channel BLC as shown

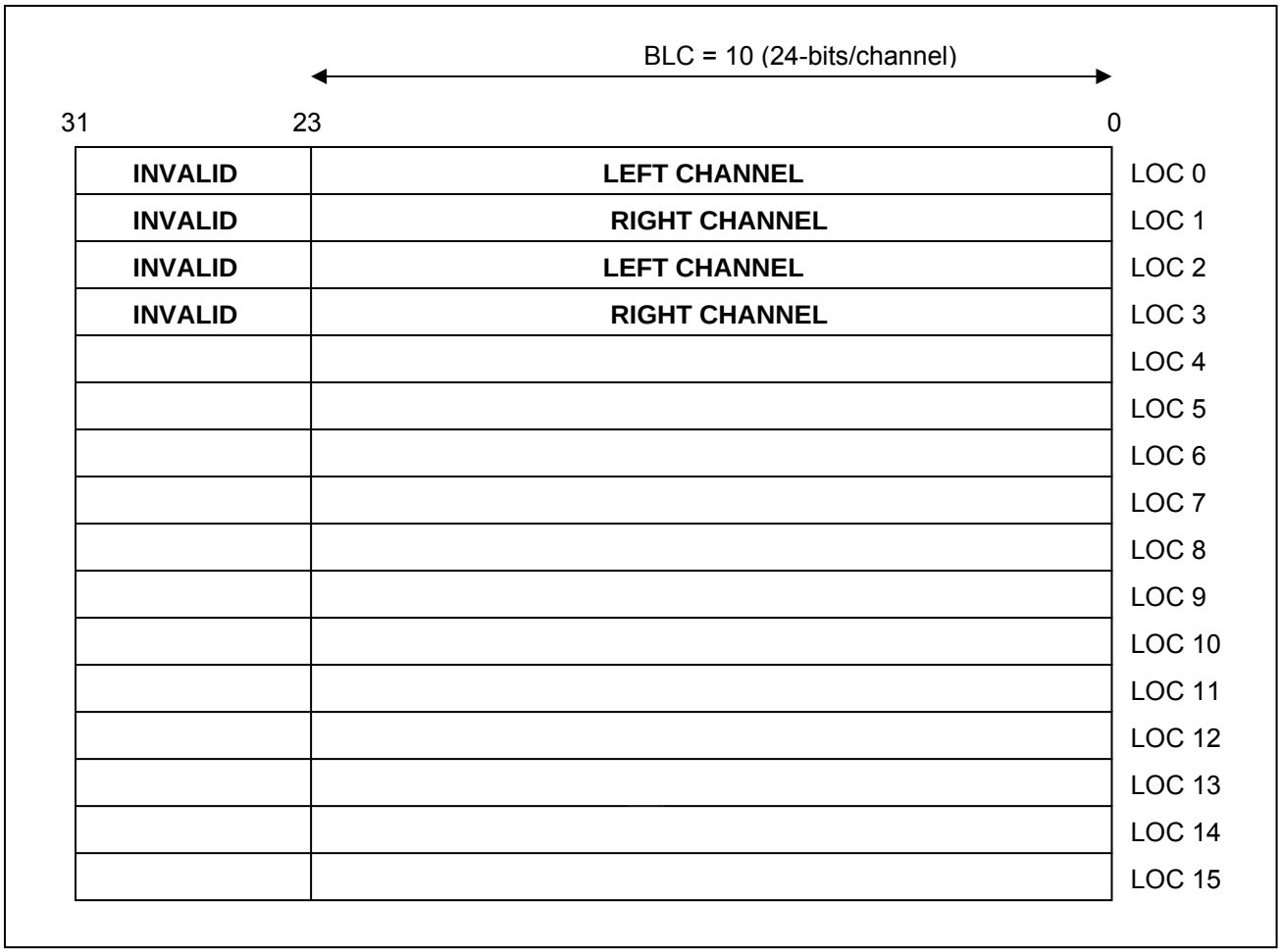


Figure 36-5. TX FIF0 Structure for BLC = 10 (24-bits/channel)

RX CHANNEL

The I2S RX channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from the input line and transferred into the RX FIFO. The processor can then read 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 can read upto 16 left/right data samples after enabling the channel for reception.

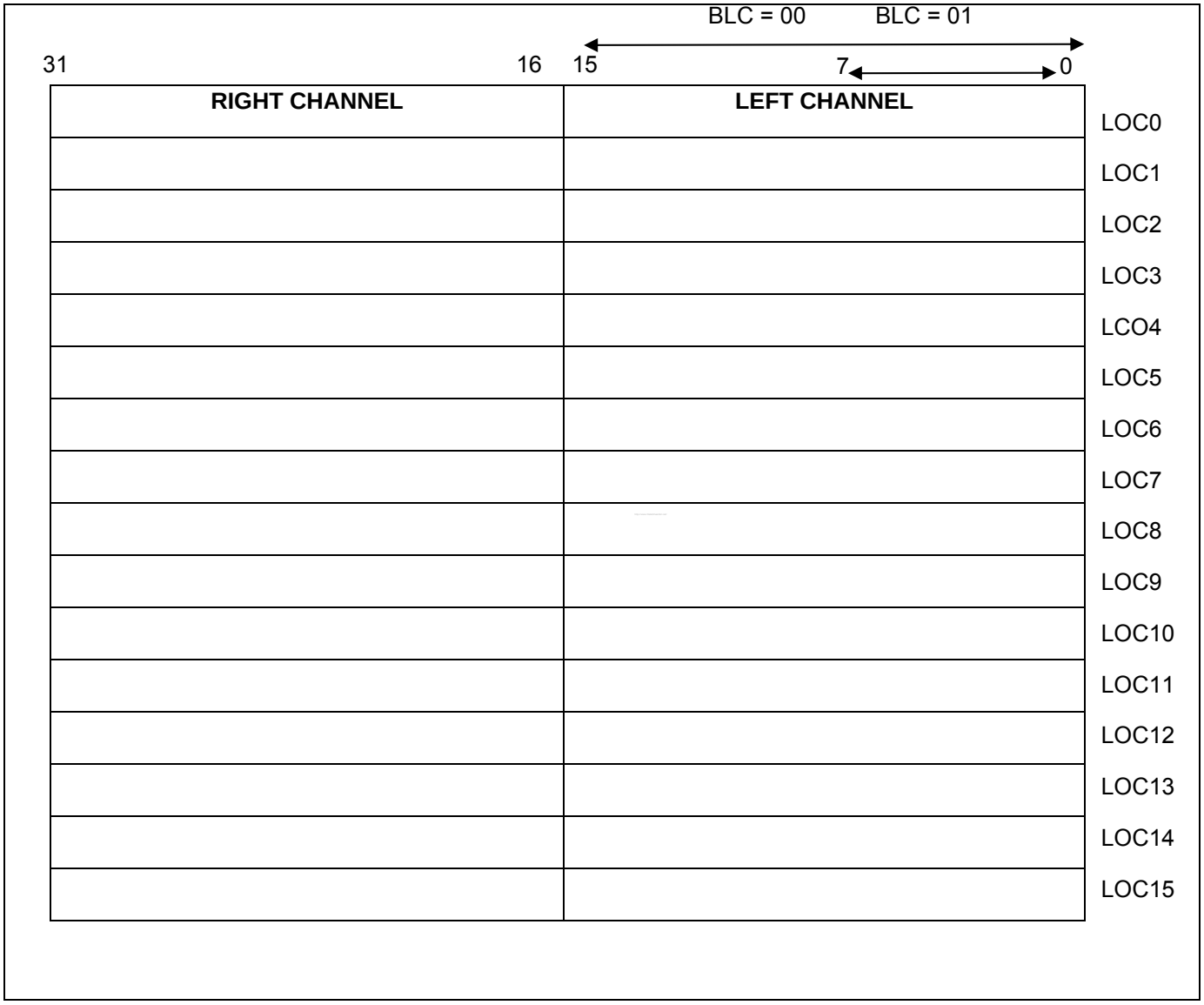


Figure 36-6. RX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the RX FIFO for 24-bits/channel BLC as shown

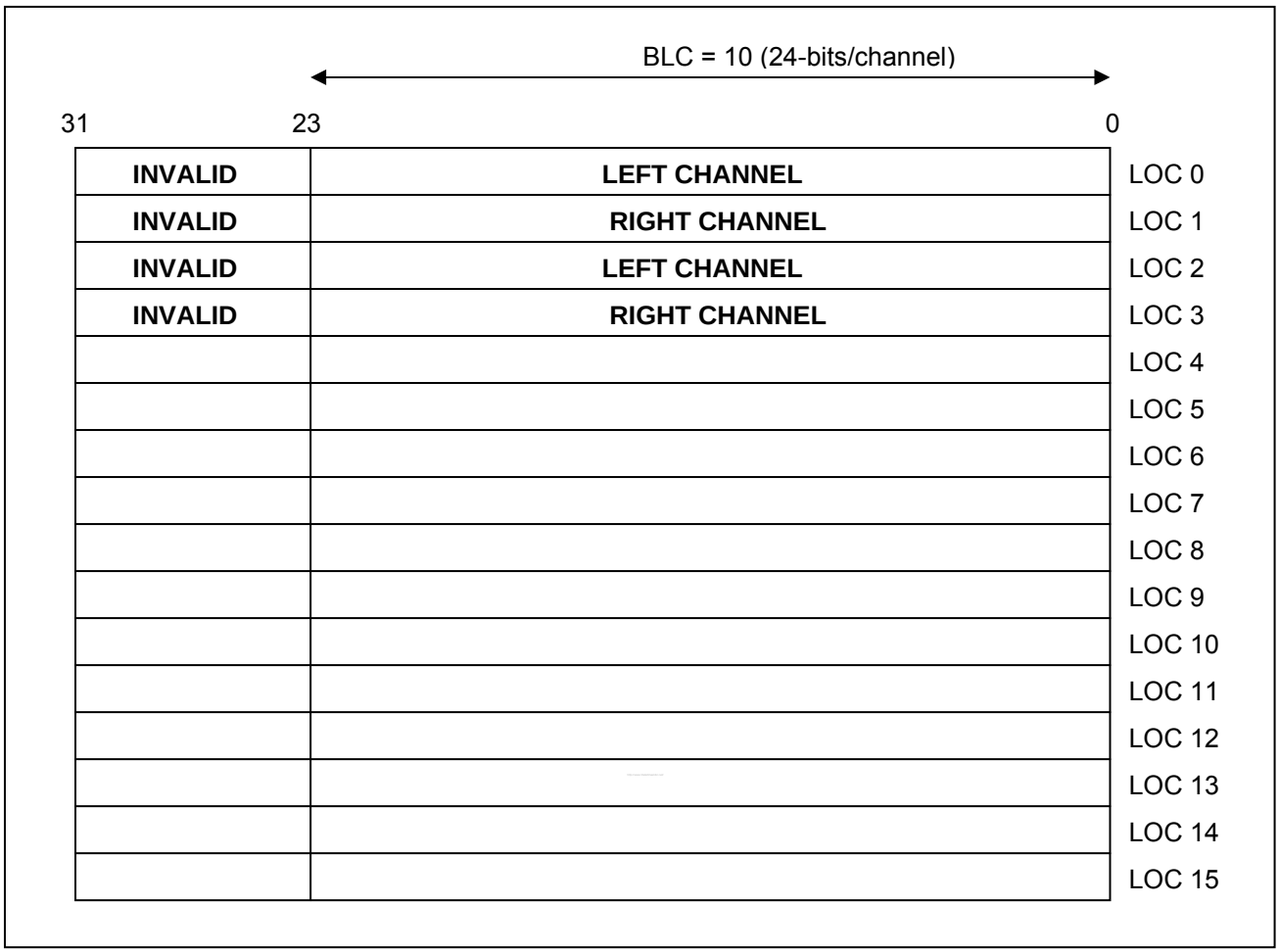


Figure 36-7. RX FIF0 Structure for BLC = 10 (24-bits/channel)

36.10 IIS-BUS INTERFACE SPECIAL REGISTERS

Table 36-3. Register summary of IIS interface

Register	Address	R/W	Description	Reset Value
IISCON	0x7F002000 0x7F003000	R/W	IIS interface control register	0xE00
IISMOD	0x7F002004 0x7F003004	R/W	IIS interface mode register	0x0
IISFIC	0x7F002008 0x7F003008	R/W	IIS interface FIFO control register	0x0
IISPSR	0x7F00200C 0x7F00300C	R/W	IIS interface clock divider control register	0x0
IISTXD	0x7F002010 0x7F003010	W	IIS interface transmit data register	0x0
IISRXD	0x7F002014 0x7F003014	R	IIS interface receive data register	0x0

36.10.1 IISCON

Register	Address	Description	Reset Value
IISCON	0x7F002000 0x7F003000	IIS interface control register	0x0000_0E00

IISCON	Bit	R/W	Description
Reserved	[31:20]	R/W	Reserved. Program to zero.
FRXORSTATUS	[19]	R/W	RX FIFO over-run interrupt status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing ‘1’. 0 : Interrupt didn’t be occurred. 1 : Interrupt was occurred.
FRXORINTEN	[18]	R/W	RX FIFO Over-run Interrupt Enable 0: RXFIFO Over-run INT disable 1: RXFIFO Over-run INT enable
FTXURSTATUS	[17]	R/W	TX FIFO under-run interrupt status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing ‘1’. 0 : Interrupt didn’t be occurred. 1 : Interrupt was occurred.
FTXURINTEN	[16]	R/W	TX FIFO Under-run Interrupt Enable 0: TXFIFO Under-run INT disable 1: TXFIFO Under-run INT enable
Reserved	[15:12]	R/W	Reserved. Program to zero.
LRI	[11]	R	Left/Right channel clock indication. Note that LRI meaning is dependent on the value of LRP bit of I2SMOD register.

			0: Left (when LRP bit is low) or right (when LRP bit is high) 1: Right (when LRP bit is low) or left (when LRP bit is high)
FTXEMPT	[10]	R	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)
FRXEMPT	[9]	R	Rx FIFO empty status indication. 0: FIFO is not empty 1: FIFO is empty
FTXFULL	[8]	R	Tx FIFO full status indication. 0: FIFO is not full 1: FIFO is full
FRXFULL	[7]	R	Rx FIFO full status indication. 0: FIFO is not full (ready for receive data from channel) 1: FIFO is full (not ready for receive data from channel)
TXDMAPAUSE	[6]	R/W	Tx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0: No pause DMA operation 1: Pause DMA operation
RXDMAPAUSE	[5]	R/W	Rx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0: No pause DMA operation 1: Pause DMA operation
TXCHPAUSE	[4]	R/W	Tx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0: No pause operation 1: Pause operation
RXCHPAUSE	[3]	R/W	Rx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0: No pause operation 1: Pause operation
TXDMACTIVE	[2]	R/W	Tx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0: Inactive, 1: Active
RXDMACTIVE	[1]	R/W	Rx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0: Inactive, 1: Active
I2SACTIVE	[0]	R/W	IIS interface active (start operation). 0: Inactive, 1:Active

36.10.2 IISMOD

Register	Address	Description	Reset Value
IISMOD	0x7F002004 0x7F003004	IIS interface mode register	0x0000_0000

IISMOD	Bit	R/W	Description
Reserved	[31:15]	R/W	Reserved. Program to zero.
BLC	[14:13]	R/W	Bit Length Control Bit Which decides transmission of 8/16 bits per audio channel 00:16 Bits per channel 01:8 Bits Per Channel 10:24 Bits Per Channel 11:Reserved
CDCLKCON	[12]	R/W	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 36-2)
IMS	[11:10]	R/W	IIS master (internal/external) or slave mode select. 00: Master mode (, using PCLK) 01: Master mode (, using CLKAUDIO[x]) 10: Slave mode (divide mode, using PCLK) 11: Slave mode (bypass mode, using I2SCLK) (For more information refer to Figure 36-2)
TXR	[9:8]	R/W	Transmit or receive mode select. 00: Transmit only mode 01: Receive only mode 10: Transmit and receive simultaneous mode 11: Reserved
LRP	[7]	R/W	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
SDF	[6:5]	R/W	Serial data format. 00: IIS format 01: MSB-justified (left-justified) format 10: LSB-justified (right-justified) format 11: Reserved
RFS	[4:3]	R/W	IIS root clock (codec clock) frequency select. 00: 256 fs, where fs is sampling frequency 01: 512 fs 10: 384 fs 11: 768 fs
BFS	[2:1]	R/W	Bit clock frequency select. 00: 32 fs, where fs is sampling frequency 01: 48 fs 10:16 fs 11: 24 fs
Reserved	[0]	R/W	Reserved. Program to zero.

36.10.3 IISFIC

Register	Address	Description	Reset Value
IISFIC	0x7F002008 0x7F003008	IIS interface FIFO control register	0x0000_0000

IISFIC	Bit	R/W	Description
Reserved	[31:16]	R/W	Reserved. Program to zero.
TFLUSH	[15]	R/W	TX FIFO flush command. 0: No flush, 1: Flush
Reserved	[14:13]	R/W	Reserved. Program to zero.
FTXCNT	[12:8]	R	TX FIFO data count. FIFO has 16 depth, so value ranges from 0 to 16. N: Data count N of FIFO
RFLUSH	[7]	R/W	RX FIFO flush command. 0: No flush, 1: Flush
Reserved	[6:5]	R/W	Reserved. Program to zero.
FRXCNT	[4:0]	R	RX FIFO data count. FIFO has 16 depth, so value ranges from 0 to 16. N: Data count N of FIFO

36.10.4 IISPSR

Register	Address	Description	Reset Value
IISPSR	0x7F00200C 0x7F00300C	IIS interface clock divider control register	0x0000_0000

IISPSR	Bit	R/W	Description
Reserved	[31:16]	R/W	Reserved. Program to zero.
PSRAEN	[15]	R/W	Pre-scaler (Clock divider) active. 0: Inactive, 1: Active
Reserved	[14]	R/W	Reserved. SBZ
PSVALA	[13:8]	R/W	Pre-scaler (Clock divider) division value. N: Division factor is N+1
Reserved	[7:0]	R/W	Reserved. Program to zero.

36.10.5 IISTXD

Register	Address	Description	Reset Value
IISTXD	0x7F002010 0x7F003010	IIS interface transmit data register	0x0000_0000

IISTXD	Bit	R/W	Description
IISTXD	[31:0]	W	TX FIFO write data. Note that the left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

36.10.6 IISRXD

Register	Address	Description	Reset Value
IISRXD	0x7F002014 0x7F003014	IIS interface receive data register	0x0000_0000

IISRXD	Bit	R/W	Description
IISRXD	[31:0]	R	RX FIFO read data. Note that the left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

37 PCM AUDIO INTERFACE

This chapter describes the functions and usage of PCM Audio interface in S3C6410X RISC microprocessor.

37.1 OVERVIEW

The S3C6410X has two ports of PCM Audio Interface module. Each port provides PCM bi-directional serial interface to an external Codec.

37.2 FEATURE

The PCM Audio interface includes the following features:

- Mono, 16bit PCM, 2 ports audio interface.
- Master mode: This block always sources the main serial clock.
- The sources of PCM clock are based on an internal PCLK or Audio clock from system contoller.
- It is possible that external PCM Audio clock(from PCMEXTCLK pad).is used as extra clock of PCMs.
- Input(16bit 32depth) and output(16bit 32depth) FIFO to buffer data.
- Optional DMA interface for TX and/or RX.

37.3 SIGNALS

Name	Direction	Description
XpcmSCLK0 XpcmSCLK1	Output	Serial shift clock.
XpcmFSYNC0 XpcmFSYNC1	Output	Serial data indicator and synchronizer
XpcmSIN0 XpcmSIN1	Input	Serial PCM Input data
XpcmSOUT0 XpcmSOUT1	Output	Serial PCM output data.
XpcmEXTCLK0 XpcmEXTCLK1	Input	Optional External clock source

37.4 PCM AUDIO INTERFACE

The PCM Audio Interface provides a serial interface to an external Codec. The PCM module receives an input PCMSOURCE_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 synchronized to the serial shift clock.

The serial shift clock, PCMSCLK, is generated from a programmable divide of the input PCMSOURCE_CLK. The sync signal PCMFSYNC, is generated based upon a programmable number of serial clocks and is one serial data clock wide.

The PCM data words are 16-bits wide and serially shifted out 1-bit per PCMSCLK. Only one 16-bit word is shifted out for each PCMFSYNC. The PCMSCLK will continue to toggle even after all 16-bits have been shifted out. The PCMSOUT data will be undefined after the 16-bit word is completed. The next PCMFSYNC will signal 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 PCM serial output data PCMSOUT is clocked out using the rising edge of the PCMSCLK. The MSB bit position relative to the PCMFSYNC is programmable to be either coincident with the PCMFSYNC or one PCMSCLK later. After all 16-bits have been shifted out, an interrupt can optionally be generated indicating the end of the transfer.

When the 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 in the falling edge of PCMSCLK. The position of the first bit is programmable to be coincident with the PCMFSYNC or one PCMSCLK later.

The first 16-bits are serially shifted into the PCM_DATAIN register which is later loaded into the RX FIFO. Subsequent bits are ignored until the next PCMFSYNC.

Various Interrupts are available to indicate the status of the RX and TX FIFO. Each FIFO has a programmable flag to indicate when the CPU needs to service the FIFO. For the RX FIFO there is an interrupt, which will be raised when the FIFO exceeds a certain programmable ALMOST_FULL depth. Similarly there is a programmable ALMOST_EMPTY interrupt for the TX FIFO.

37.4.1 PCM TIMING

The following figures show the timing relationship for the PCM transfers.

Figure 37-1 shows a PCM transfer with the MSB configured to be coincident with the PCMFSYNC. This MSB positioning corresponds to setting the TX_MSB_POS and RX_MSB_POS bits in PCMCTL register to be 0.

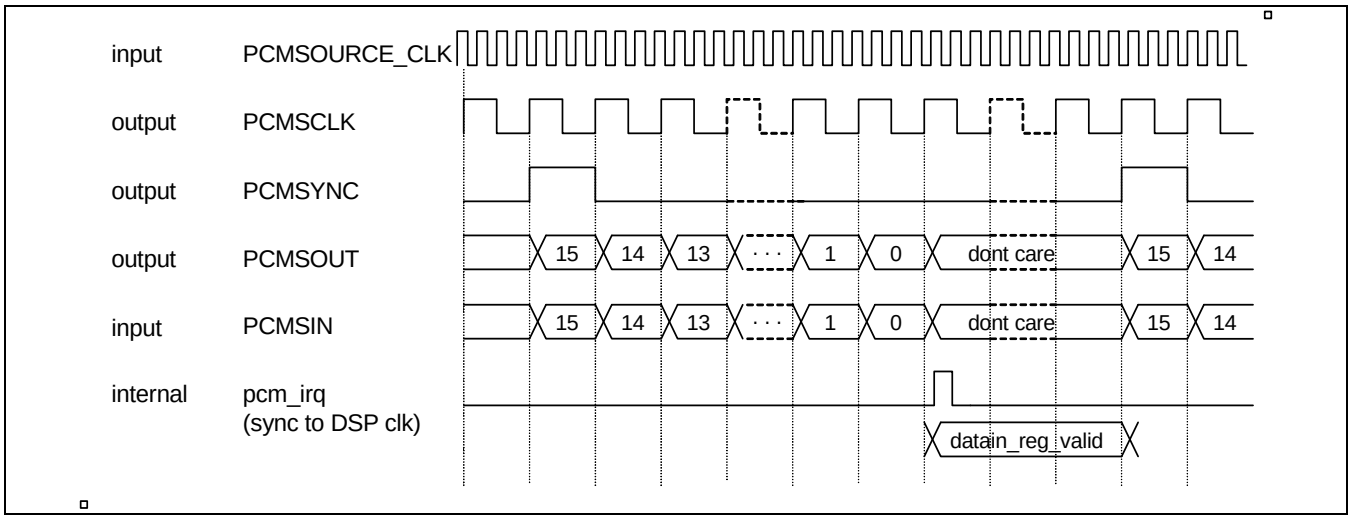


Figure 37-1. PCM timing, TX_POS_MSB/ RX_POS_MSB = 0

Figure 37-2 shows a PCM transfer with the MSB configured one shift clock after the PCMFSYNC. This MSB positioning corresponds to setting the TX_MSB_POS and RX_MSB_POS bits in PCMCTL register to be 1.

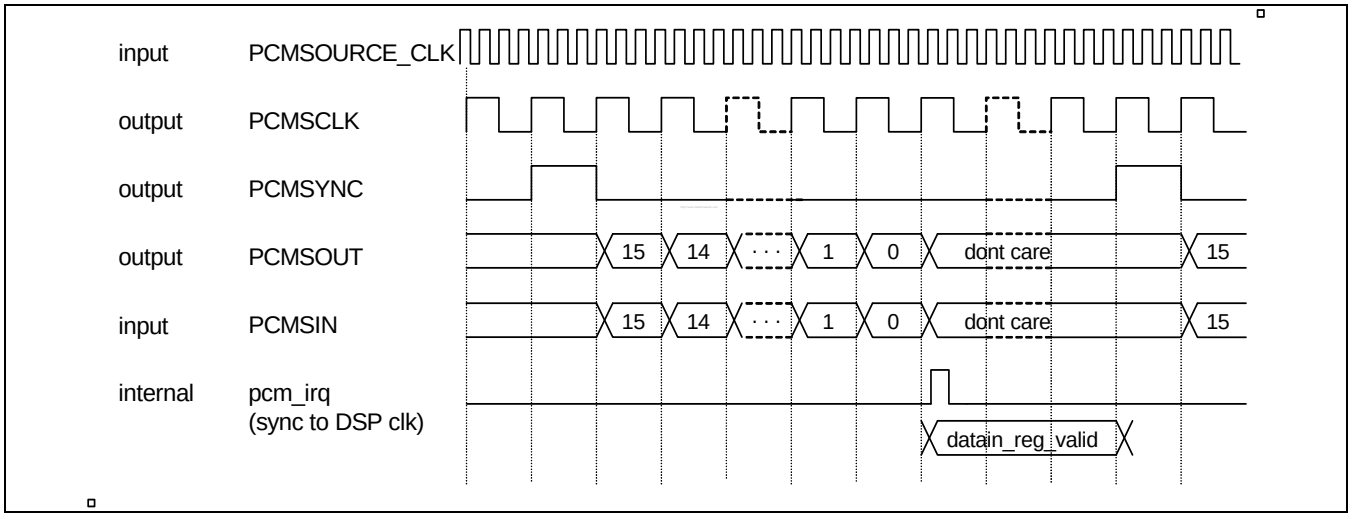


Figure 37-2. PCM timing, TX_POS_MSB/ RX_POS_MSB = 1

Note In all cases, the PCM shift timing is derived by dividing the input clock, PCMSOURCE_CLK. While the timing is based upon the PCMSOURCE_CLK, there is no attempt to realign the rising edge of the output PCMSCLK with the original PCMSOURCE_CLK input clock. These edges will be 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. If the PCMSCLK output is not used, the skew will be significantly less than the period of the PCMSOURCE_CLK. It should not represent a problem since most PCM interfaces capture data on the falling edge of the clock.

37.4.2 PCM INPUT CLOCK DIAGRAM

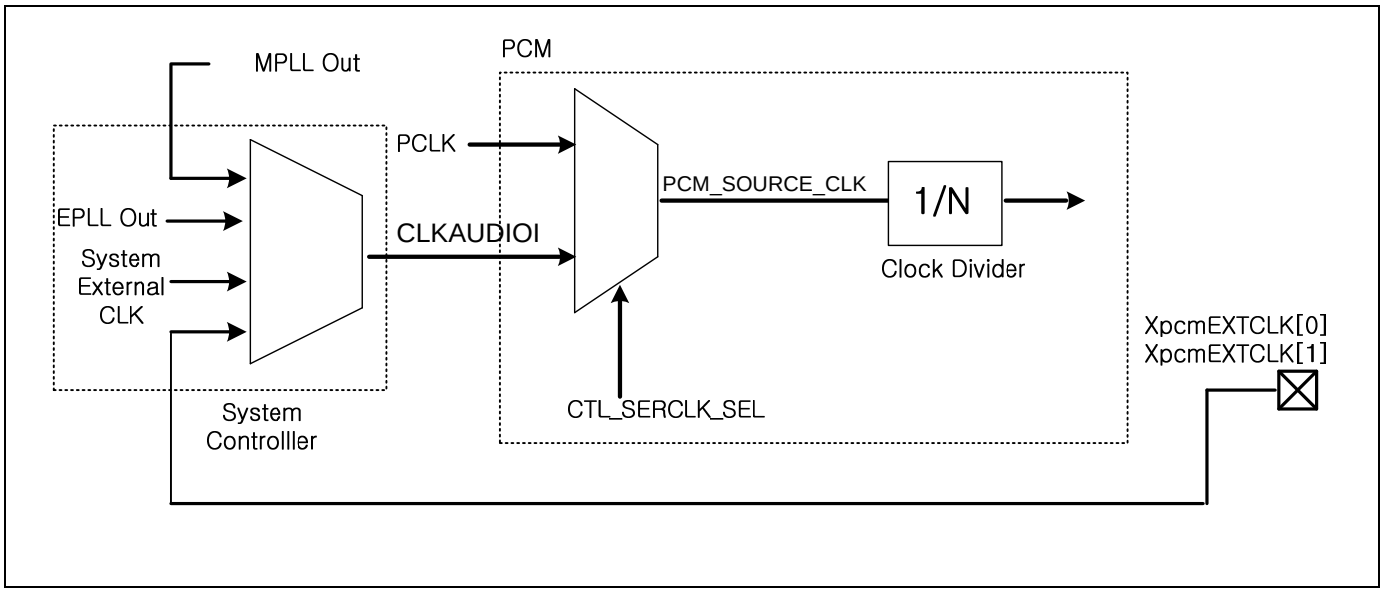


Figure 37-3. Input Clock Diagram for PCM

S3C6410X can provide PCM with a variety of clock. As described in the Figure 37-3, PCM interface is able to select clock from two PCLK or AUDIO clock(CLKAUDIOI) which is from system controller. We can also select CLKAUDIO among PLL or external input clock(PCMEXTCLK). To select CLKAUDIO, please refer to the SYSCON (chapter 6).

37.5 PCM REGISTERS

There are 8 control special function registers (SFRs) for each PCM Port. Since there are two ports, the total number of SFR is 16. Base Address of Port 0 is 0x7F00900 and Port 1 is 0x7F00A000.

37.5.1 PCM REGISTER SUMMARY

Register	Address	R/W	Description	Reset Value
PCM_CTL	0x7F009000 0x7F00A000	R/W	PCM Main Control	0x00000000
PCM_CLKCTL	0x7F009004 0x7F00A004	R/W	PCM Clock Control	0x00000000
PCM_TX_FIFO	0x7F009008 0x7F00A008	R/W	PCM TX FIFO write port	0x00000000
PCM_RX_FIFO	0x7F00900C 0x7F00A00C	R/W	PCM RX FIFO read port	0x00000000
PCM_IRQ_CTL	0x7F009010 0x7F00A010	R/W	PCM Interrupt Control	0x00000000
PCM_IRQ_STAT	0x7F009014 0x7F00A014	R	PCM Interrupt Status	0x00000000
PCM_FIFO_STAT	0x7F009018 0x7F00A018	R	PCM FIFO Status	0x00000000
PCM_CLRINT	0x7F009020 0x7F00A020	W	PCM INTERRUPT CLEAR	0x00000000

37.5.2 PCM CONTROL REGISTER

The PCM_CTL register is used to control the various aspects of the PCM module. It also provides a status bit to provide the option to use polling instead of interrupt based control.

Register	Address	R/W	Description	Reset Value
PCM_CTL	0x7F009000 0x7F00A000	R/W	Control the PCM Audio Inteface	0x00000000

The bit definitions for the PCM_CTL Control Register are described below:

PCM_CTL	Bit	Description	Initial State
Reserved	[31:19]	Reserved	
TX_FIFO_DIPSTICK	[18:13]	Determines when the TX_ALMOST_FULL, TX_ALMOST_EMPTY flags go active for the TX FIFO TX_ALMOST_EMPTY: TX_FIFO_DEPTH < TX_FIFO_DIPSTICK TX_ALMOST_FULL: TX_FIFO_DEPTH > (0x20 – TX_FIFO_DIPSTICK) NOTE: When TX_FIFO_DIPSTICK is 0, TX_ALMOST_EMPTY, TX_ALMOST_FULL are invalid NOTE: For DMA loading of TX FIFO, TX_FIFO_DIPSTICK is 0x2 or more. This is required since the PCM_TXDMA uses TX_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 TX_ALMOST_FULL flag mostly go active with at least space for one extra word in the FIFO	0

PCM_CTL	Bit	Description	Initial State
RX_FIFO_DIPSTICK	[12:7]	Determines when the RX_ALMOST_FULL, RX_ALMOST_EMPTY flags go active for the RX FIFO RX_ALMOST_EMPTY: RX_FIFO_DEPTH < RX_FIFO_DIPSTICK RX_ALMOST_FULL: RX_FIFO_DEPTH > (0x20 – RX_FIFO_DIPSTICK) NOTE: When RX_FIFO_DIPSTICK is 0x0, RX_ALMOST_EMPTY, RX_ALMOST_FULL are invalid NOTE: for DMA, RX_FIFO_DIPSTICK is a don't care DMA unloading of RX FIFO uses the RX_ALMOST_EMPTY flag as the DMA request NOTE: non-DMA IRQ/polling RXFIFO_DIPSTICK must 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 must operate in the demand mode. DMA_TX request will occur whenever the TX FIFO is not TX_ALMOST_FULL	0
PCM_RX_DMA_EN	[5]	Enable the DMA interface for the RXFIFO DMA must operate in the demand mode. DMA_RX request will occur whenever the RXFIFO is not empty.	0
TX_MSB_POS	[4]	Controls the position of the MSB bit in the serial output stream relative to the PCMFSYNC signal 0: MSB sent during the same clock that PCMFSYNC is high 1: MSB sent on the next PCMSCLK cycle after PCMFSYNC is high	0
RX_MSB_POS	[3]	Controls the position of the MSB bit in the serial input stream relative to the PCMFSYNC signal 0: MSB is captured on the falling edge of PCMSCLK during the same cycle that PCMFSYNC is high 1: MSB is captured on the falling edge of PCMSCLK during the cycle after the PCMFSYNC is high	0
PCM_TXFIFO_EN	[2]	Enable the TXFIFO	0
PCM_RXFIFO_EN	[1]	Enable the RXFIFO	0

PCM_CTL	Bit	Description	Initial State
PCM_PCM_ENABLE	[0]	PCM enable signal. 1: Enables the serial shift state machines. The enable must be set HIGH for the PCM to operate. 0: The PCMSOUT will not toggle.	0

37.5.3 PCM CLK CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
PCM_CLKCTL	0x7F009004 0x7F00A004	R/W	Control the PCM Audio Inteface	0x00000000

The bit definitions for the PCM_CTL Control Register are described below:

PCM_CLKCTL	Bit	Description	Initial State
Reserved	[31:20]	Reserved	
CTL_SERCLK_EN	[19]	Enable the serial clock division logic. Must be HIGH for the PCM to operate (if it is high, PCMSCLK and PCMFSYNC is operated.)	0
CTL_SERCLK_SEL	[18]	Select the source of the PCMSCLK 0 - CLKAUDIOI 1 - PCLK	0
SCLK_DIV	[17:9]	Controls the divider used to create the PCMSCLK based on the PCMSOURCE_CLK PCMSCLK will be PCMSOURCE_CLK / 2*(SCLK_DIV+1)	000
SYNC_DIV	[8:0]	Controls the frequency of the PCMFSYNC signal based on the PCMSCLK. PCMFSYNC will be PCMSCLK/(SYNC_DIV+1)	000

37.5.4 THE PCM TX FIFO REGISTER

Register	Address	R/W	Description	Reset Value
PCM_TXFIFO	0x7F009008 0x7F00A008	R/W	Control the PCM Audio Inteface	0x00000000

The bit definitions for the PCM_TX_FIFO Register are described below:

PCM_TX_FIFO	Bit	Description	Initial State
Reserved	[31:17]	Reserved	
TX_FIFO_DVALID	[16]	TX FIFO data is valid Write: Not Valid Read: TX FIFO read data valid 1: valid 0: invalid (probably read an empty FIFO)	0
TX_FIFO_DATA	[15:0]	TX FIFO DATA Write: TX_FIFO_DATA is written into the TX FIFO Read: TX FIFO is read using the APB interface **NOTE:** reading the TX FIFO is meant to support debugging. Online the TX FIFO is read by the PCM serial shift engine, not the APB	0

37.5.5 PCM RX FIFO REGISTER

Register	Address	R/W	Description	Reset Value
PCM_RXFIFO	0x7F00900C 0x7F00A00C	R/W	Control the PCM Audio Inteface	0x00000000

The bit definitions for the PCM_RXFIFO Register are described below:

PCM_RXFIFO	Bit	Description	Initial State
Reserved	[31:17]	Reserved	
RXFIFO_DVALID	[16]	RX FIFO data is valid Write: Not Valid Read: RX FIFO read data valid 1: valid 0: invalid (probably read an empty fifo)	0
RXFIFO_DATA	[15:0]	RX FIFO DATA Write: RX_FIFO_DATA is written into the RX FIFO **NOTE:** writing the RX FIFO is meant to support debugging. Online the RXFIFO is written by the PCM serial shift engine, not the APB Read: RX FIFO is read using the APB interface	0

37.5.6 PCM INTERRUPT CONTROL REGISTER

The PCM_IRQ_CTL register is used to control the various aspects of the PCM interrupts.

Register	Address	R/W	Description	Reset Value
PCM_IRQ_CTL	0x7F009010 0x7F00A010	R/W	Control the PCM Interrupts	0x00000000

The bit definitions for the PCM_IRQ_CTL Control Register are described below:

PCM_IRQ_CTL	Bit	Description	Initial State
Reserved	[31:15]	Reserved	
EN_IRQ_TO_ARM	[14]	Controls whether the PCM interrupt is sent to the ARM or not 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 16bit PCM Data word completes 1: IRQ source enabled 0: IRQ source disabled	0
TX_FIFO_EMPTY	[11]	Interrupt is generated whenever the TX FIFO is EMPTY 1: IRQ source enabled 0: IRQ source disabled	0
TX_FIFO_ALMOST_EMPTY	[10]	Interrupt is generated whenever the TX FIFO is ALMOST_EMPTY. ALMOST_EMPTY is defined as TX_FIFO_DEPTH < TX_FIFO_DIPSTICK 1: IRQ source enabled 0: IRQ source disabled	0
TX_FIFO_FULL	[9]	Interrupt is generated whenever the TX FIFO is FULL 1: IRQ source enabled 0: IRQ source disabled	0
TX_FIFO_ALMOST_FULL	[8]	Interrupt is generated whenever the TX FIFO is ALMOST_FULL. ALMOST_FULL is defined as TX_FIFO_DEPTH > (32 – TX_FIFO_DIPSTICK) 1: IRQ source enabled 0: IRQ source disabled	0

PCM_IRQ_CTL	Bit	Description	Initial State
TX_FIFO_ERROR _STARVE	[7]	Interrupt is generated for TX FIFO starve ERROR. This occurs whenever the TX FIFO is read when it is still empty. This is considered as a ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0
TX_FIFO_ERROR _OVERFLOW	[6]	Interrupt is generated for TX FIFO overflow ERROR. This occurs whenever the TX FIFO is written when it is already full. This is considered as a ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_EMPTY	[5]	Interrupt is generated whenever the RX FIFO is empty 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_ALMOST _EMPTY	[4]	Interrupt is generated whenever the RX FIFO is ALMOST_EMPTY. ALMOST_EMPTY is defined as RX_FIFO_DEPTH < RX_FIFO_DIPSTICK 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_FULL	[3]	Interrupt is generated whenever the RX FIFO is full 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_ALMOST _FULL	[2]	Interrupt is generated whenever the RX FIFO is ALMOST_FULL. ALMOST_FULL is defined as RX_FIFO_DEPTH > (32 – RX_FIFO_DIPSTICK) 1: IRQ source enabled 0: IRQ source disabled	0
RX_FIFO_ERROR _STARVE	[1]	Interrupt is generated for RX FIFO starve ERROR. This occurs whenever the RX FIFO is read when it is still empty. This is considered as a ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0

PCM_IRQ_CTL	Bit	Description	Initial State
RX_FIFO_ERROR _OVERFLOW	[0]	Interrupt is generated for RX FIFO overflow ERROR. This occurs whenever the RX FIFO is written when it is already full. This is considered as a ERROR and will have unexpected results 1: IRQ source enabled 0: IRQ source disabled	0

37.5.7 PCM INTERRUPT STATUS REGISTER

The PCM_IRQ_STAT register is used to report IRQ status.

Register	Address	R/W	Description	Reset Value
PCM_IRQ_STAT	0x7F009014 0x7F00A014	R	PCM Interrupt Status	0x00000000

The bit definitions for the PCM_IRQ_STATUS Register are described below:

PCM_IRQ_STAT	Bit	Description	Initial State
Reserved	[31:14]	Reserved	
IRQ_PENDING	[13]	Monitoring PCM IRQ. 1: PCM IRQ is occurred. 0: PCM IRQ is not occurred.	0
TRANSFER_DONE	[12]	Interrupt is generated every time the serial shift for a word completes 1: IRQ is occurred. 0: IRQ is not occurred.	0
TXFIFO_EMPTY	[11]	Interrupt is generated whenever the TX FIFO is empty 1: IRQ is occurred. 0: IRQ is not occurred.	0
TXFIFO_ALMOST_EMPTY	[10]	Interrupt is generated whenever the TXFIFO is ALMOST empty. Almost empty is defined as FIXME words remaining 1: IRQ is occurred. 0: IRQ is not occurred.	0
TXFIFO_FULL	[9]	Interrupt is generated whenever the TX FIFO is full 1: IRQ is occurred. 0: IRQ is not occurred.	0
TXFIFO_ALMOST_FULL	[8]	Interrupt is generated whenever the TX FIFO is ALMOST full. Almost full is defined as FIXME words remaining 1: IRQ is occurred. 0: IRQ is not occurred.	0
TXFIFO_ERROR_STARVE	[7]	Interrupt is generated for TX FIFO starve ERROR. This occurs whenever the TX FIFO is read when it is still empty. This is considered as a ERROR and will have unexpected results 1: IRQ is occurred. 0: IRQ is not occurred.	0

PCM_IRQ_STAT	Bit	Description	Initial State
TXFIFO_ERROR _OVERFLOW	[6]	Interrupt is generated for TX FIFO overflow ERROR. This occurs whenever the TX FIFO is written when it is already full. This is considered as a ERROR and will have unexpected results 1: IRQ is occurred. 0: IRQ is not occurred.	0
RXFIFO_EMPTY	[5]	Interrupt is generated whenever the RX FIFO is empty 1: IRQ is occurred. 0: IRQ is not occurred.	0
RXFIFO_ALMOST _EMPTY	[4]	Interrupt is generated whenever the RX FIFO is ALMOST empty. Almost empty is defined as FIXME words remaining 1: IRQ is occurred. 0: IRQ is not occurred.	0
RX_FIFO_FULL	[3]	Interrupt is generated whenever the RX FIFO is full 1: IRQ is occurred. 0: IRQ is not occurred.	0
RX_FIFO_ALMOST _FULL	[2]	Interrupt is generated whenever the RX FIFO is ALMOST full. Almost full is defined as FIXME words remaining 1: IRQ is occurred. 0: IRQ is not occurred.	0
RXFIFO_ERROR _STARVE	[1]	Interrupt is generated for RX FIFO starve ERROR. This occurs whenever the RX FIFO is read when it is still empty. This is considered as a ERROR and will have unexpected results 1: IRQ is occurred. 0: IRQ is not occurred.	0
RXFIFO_ERROR _OVERFLOW	[0]	Interrupt is generated for RX FIFO overflow ERROR. This occurs whenever the RX FIFO is written when it is already full. This is considered as a ERROR and will have unexpected results 1: IRQ is occurred. 0: IRQ is not occurred.	0

37.4.8 PCM FIFO STATUS REGISTER

The PCM_FIFO_STAT register is used to report FIFO status.

Register	Address	R/W	Description	Reset Value
PCM_FIFO_STAT	0x7F009018 0x7F00A018	R	PCM FIFO Status	0x00000000

The bit definitions for the PCM_FIFO_STATUS Register are described below:

PCM_FIFO_STAT	Bit	Description	Initial State
Reserved	[31:20]	Reserved	
TXFIFO_COUNT	[19:14]	TX FIFO data count(0x0 ~ 0x20).	0
TXFIFO_EMPTY	[13]	Indicate whether TXFIFO is empty or not. 0 : not empty 1: empty	0
TXFIFO_ALMOST_EMPTY	[12]	To indicate whether TXFIFO is ALMOST_EMPTY or not. 0 : not ALMOST_EMPTY 1: ALMOST_EMPTY	0
TXFIFO_FULL	[11]	To indicate whether TXFIFO is full or not. 0 : not full 1: full	0
TXFIFO_ALMOST_FULL	[10]	To indicate whether TXFIFO is ALMOST_FULL. 0 : not ALMOST_FULL 1: ALMOST_FULL	0
RXFIFO_COUNT	[9:4]	RX FIFO data count(0x0 ~ 0x 20)	
RXFIFO_EMPTY	[3]	Indicate whether RX FIFO is empty or not. 0: not empty 1: empty	0
RXFIFO_ALMOST_EMPTY	[2]	Indicate whether RX FIFO is ALMOST_EMPTY or not 0: not ALMOST_FULL 1: ALMOST_FULL	0
RX_FIFO_FULL	[1]	Indicate whether RXFIFO is full or not. 0: not full 1: full	0
RX_FIFO_ALMOST_FULL	[0]	Indicate whether RXFIFO is ALMOST_FULL or not 0: not ALMOST_FULL 1: ALMOST_FULL	0

37.4.9 PCM INTERRUPT CLEAR REGISTER

The PCM_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 ARM. Reading this register is not allowed. Clearing interrupt must be prior to resolving the interrupt condition; otherwise another interrupt that would occur after this interrupt may be ignored.

Register	Address	R/W	Description	Reset Value
PCM_CLRINT	0x7F009020 0x7F00A020	W	PCM INTERRUPT CLEAR	-

The bit definitions for the PCM_CLRINT Register are described below:

PCM_CLRINT	Bit	Description	Initial State
Reserved	[31:1]	Reserved	0
CLRINT	[0]	Interrupt register clear	0

NOTES

38 IRDA CONTROLLER

This chapter describes the functions and usage of IrDA Controller in S3C6410X RISC microprocessor.

38.1 OVERVIEW

The Samsung IrDA Core is a wireless serial communication controller. The Samsung IrDA Core supports two different types of IrDA speed (MIR, FIR). This core can transmit Ir(Infrared) pulses up to 4 Mbps speed. It includes configurable FIFO feature to reduce the CPU burden. This makes it easy to adjust the internal FIFO sizes.

You can program the core by accessing 16 internal registers. When receiving the Ir pulses, this core detects three kinds of line errors such as CRC-error, PHY-error and payload length error.

38.2 FEATURES

The IrDA Controller supports:

- IrDA specification compliant
  IrDA 1.1 physical layer specification
- FIFO operation in the MIR and FIR mode (4Mbps, 1.152Mpbs and 0.576Mbps)
- 64-byte FIFO size
- Back-to-Back Transactions
- Software in selecting Temic-IBM or HP transceiver

38.3 BLOCK DIAGRAM

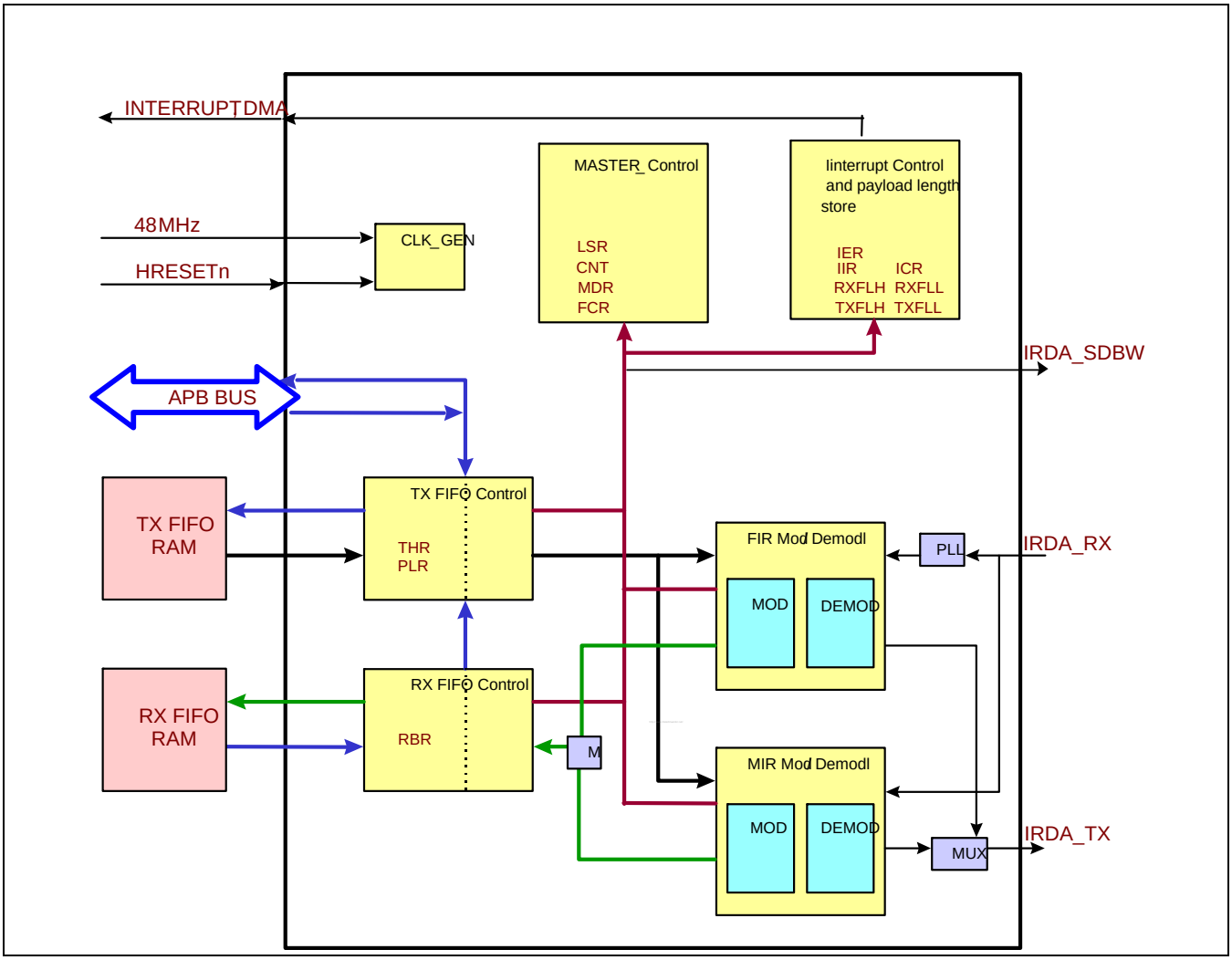


Figure 38-1. Block Diagram

38.4 EXTERNAL INTERFACE SIGNALS

IRDA_TX : IrDA Tx signal (output)
IRDA_RX : IrDA Rx signal (input)
IRDA_SDBW : IrDA Transceiver control (Shutdown, Bandwidth) (output)
MCLK : IrDA operation clock; Must set up IrDA Clock in SYSCON as 48MHz

Table 38-1. IrDA signals

Group	Name	Bit	Direction	Source/Destination
IrDA special signal	MCLK	1	IN	SYSCON (IrDA operation clock : must be 48MHz) Source Clock : USB clock, PLL clock
	IrDA_Rx	1	IN	PAD
	IrDA_Tx	1	OUT	PAD
	IrDA_SDBW	1	OUT	PAD

38.5 FUNCTION DESCRIPTION

38.5.1 FAST-SPEED INFRARED (FIR) MODE (IRDA 1.1)

In FIR mode, data communicates at the baud rate speed of 4 Mbps. In the data transmission mode, 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, when Ir 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 the 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 errors are the Phy-Error, the Frame-Length Error and the CRC error. The CRC error is checked when the entire payload data is received. The micro-controller can monitor the error status of the received frame by reading the Line Status Register (IRDA_LSR) at the end of the each received frame.

The below diagram shows the frame structure of the fir data frame (The specific information of the each field can be found 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

The number of preambles is 16.

* **Note:** 4 PPM Coding

Data Bit Pair(DBP)	4PPM Data Symbol(DD)
00	1000
01	0100
10	0010
11	0001

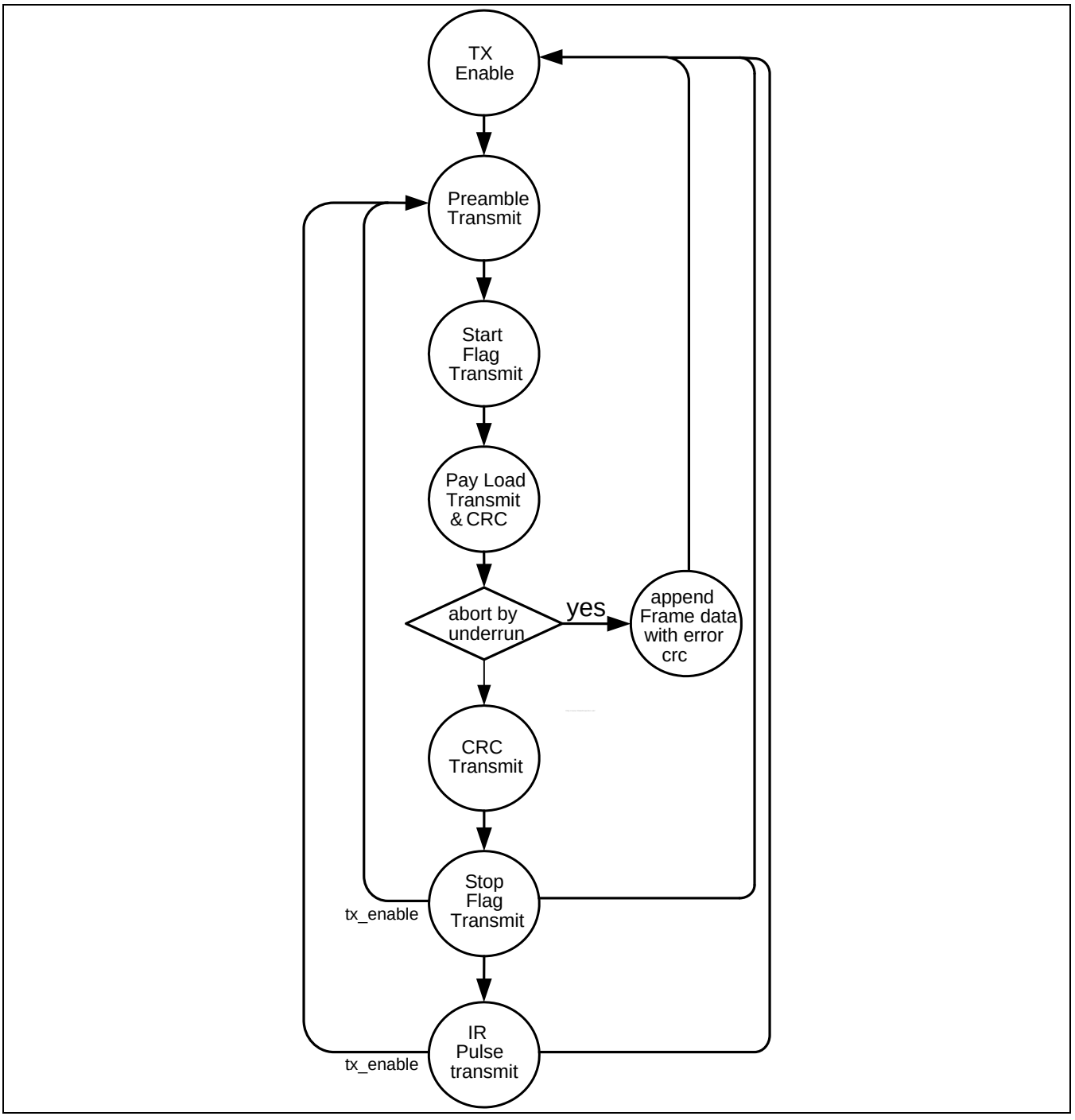


Figure 38-2. FIR modulation process

Figure 38-2 shows the FIR modulation state machine. The FIR transmission mode can be selected by programming IRDA_CNT register. If an under run condition occurs, the state machine appends the payload with error crc data and terminates the transmission.

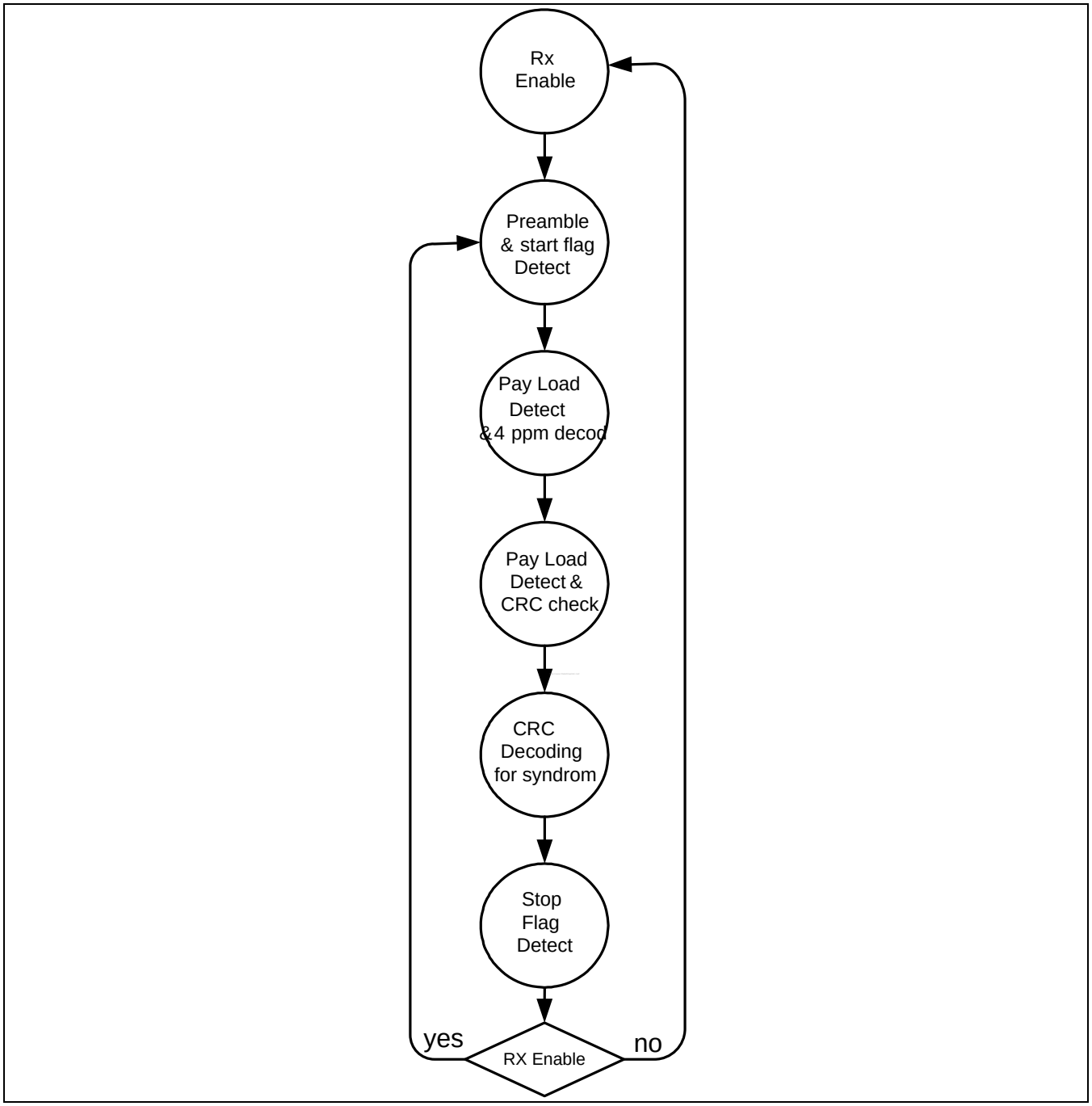


Figure 38-3. FIR demodulation process

Figure 38-3 shows FIR demodulation state machine. The state machine starts when IRDA_CNT register bit 6 is set to logic high. The incoming data will be depacketized by removing preamble and start flag and stop flag. 4PPM decoding and CRC decoding is carried out.

38.5.2 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 must be at least 3 bytes. In transmitting and receiving process, the basic wrapping and de-wrapping processes are same as the FIR mode. The MIR mode needs a 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 must be removed. Like the fir mode case, three different kinds of errors (crc, phy and frame length error) can be reported to the microcontroller in receiving mode by reading the IRDA_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 the pulse generation.

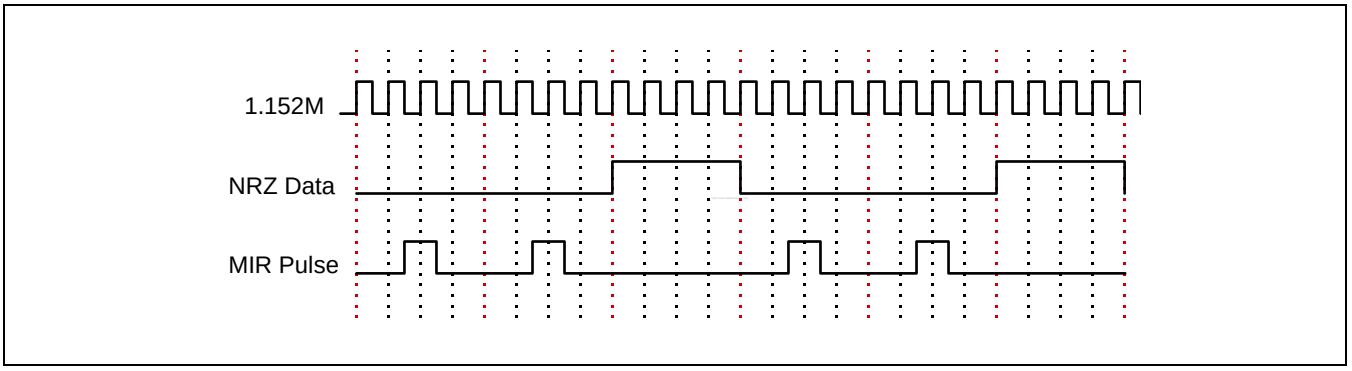


Figure 38-4. Pulse modulation in MIR mode

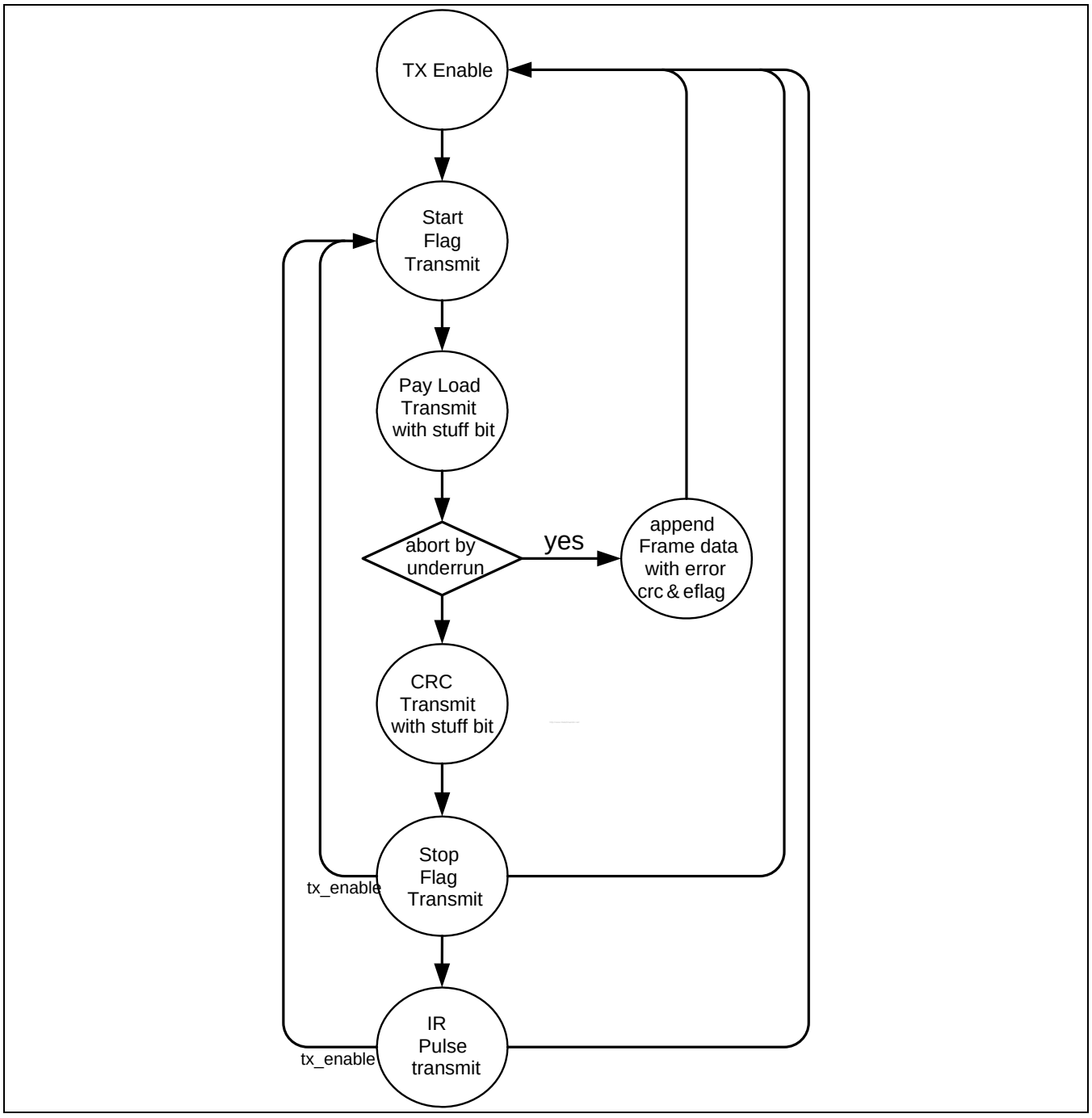


Figure 38-5. MIR modulation process

Figure 38-5 describes MIR modulation state machine. This machine works very similarly as FIR modulation state machine. The major difference is that the MIR data transmission needs bit stuffing. After the every 5 consecutive ones, a zero data should be stuffed in MIR payload data. The state machine for this bit-stuffing is not presented here.

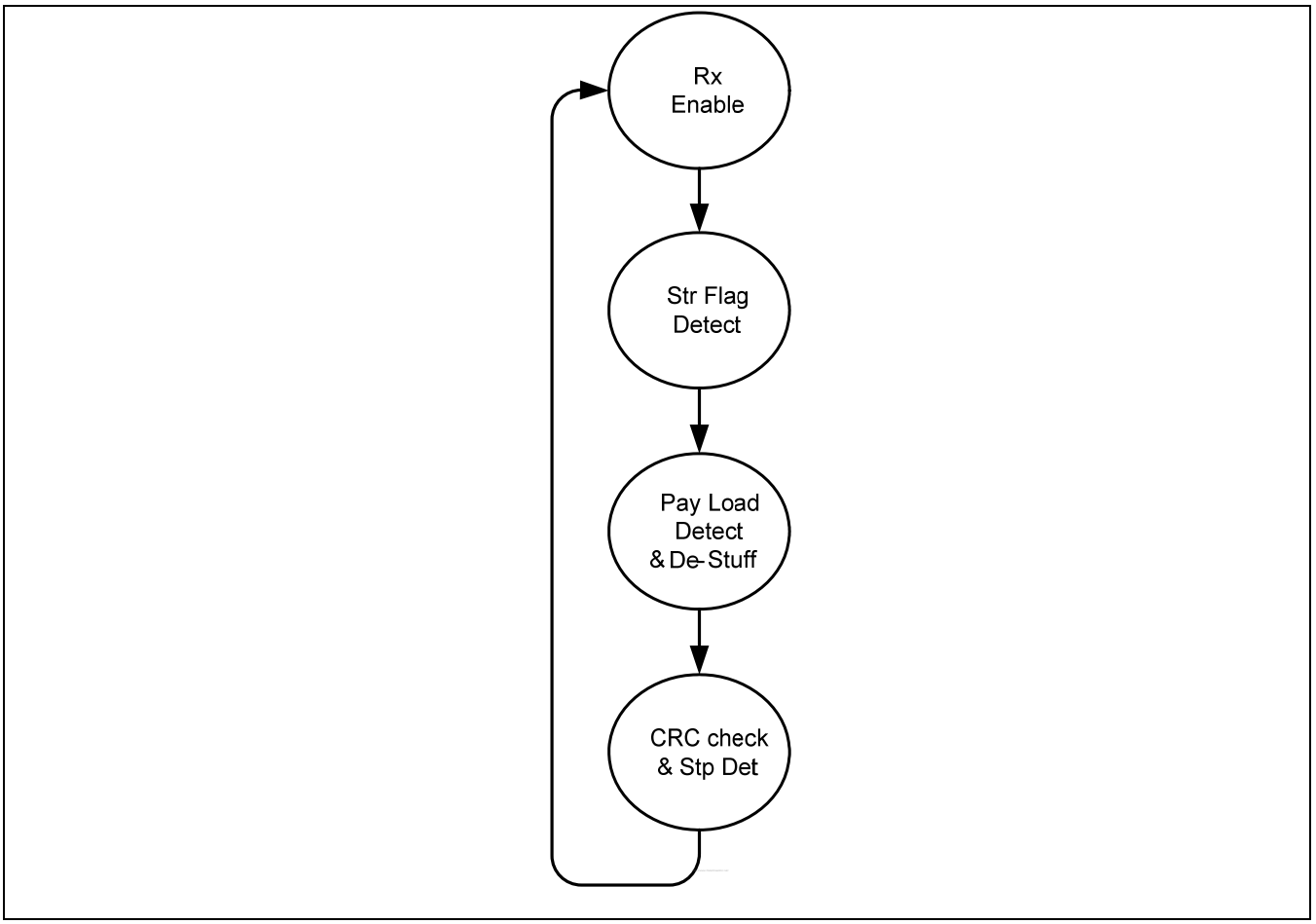


Figure 38-6. MIR demodulation process

Figure 38-6 shows the MIR demodulation state machine. It has similar structure as FIR demodulation state machine. Instead of 4 PPM demodulation phase, it has the stage of removing stuffed bits from payload data stream. Since the MIR data stream doesn't have preamble data, the preamble/start flag data detection stage in MIR demodulation is simplified to start flag detection state.

38.5.3 CORE INITIALIZATION PROCEDURE

MIR/FIR Mode Initialization Operation

1. Program the IRDA_MDR register to select the MIR/FIR mode.
2. Program the IRDA_CNT register to select the transceiver type.
   - For the Temic-IBM type transceiver, program twice in IRDA_CNT[0] = 1'b0 and IRDA_CNT[0] = 1'b1.
   - For the HP type transceiver, program just once in IRDA_CNT[0] = 1'b0 to FIR/MIR mode.
3. Program the IRDA_PLR register to select the number of start flag, and TX threshold level.
4. Program the IRDA_RXFLL and IRDA_RXFLH register (maximum available receive bytes in frame).
5. Program the IRDA_TXFLL and IRDA_TXFLH register (transmit bytes in transmission frame).
6. Program the IRDA_FCR register (FIFO size and RX threshold level).
7. Program the IRDA_IER register (the types of interrupt).
8. Program the IRDA_CNT register (TX enable or RX enable).
9. Program the IRDA_IER register (interrupt enable).
10. Service Interrupt signal from the core.

38.6 SPECIAL FUNCTION REGISTERS

38.6.1 IRDA CONTROL REGISTER(IRDA_CNT)

Register	Address	R/W	Description	Reset Value
IRDA _CNT	0x7F00_7000	R/W	IrDA Control Register	0x00

IRDA _CNT	Bit	Description	Initial State
TX enable	[7]	TX enabled. Bit 7 must be set to '1' to enable data transmission in MIR/FIR Ir modes.	0
RX enable	[6]	RX enabled. Bit 6 must be set to '1' to enable data receive in all MIR/FIR Ir 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. When 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]	Send 1.6-us IR pulse. When the IRDA_MDR[4] 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]	Reserved	0
Frame abort	[1]	Frame abort. The CPU can intentionally abort 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 will find 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 can be 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

38.6.2 IRDA MODE DEFINITION REGISTER(IRDA_MDR)

Register	Address	R/W	Description	Reset Value
IRDA_MDR	0x7F00_7004	R/W	IrDA Mode Definition Register	0x00

IRDA _MDR	Bit	Description	Initial State
Reserved	[7:5]	Reserved	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, when 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' doesn't help to generate SIP. Along with IRDA_CNT[3] bit, the way of SIP generation can be controlled.	0
Temic select	[3]	Bit 3 is Temic transceiver select bit. When bit 3 is clear to "0", core automatically selects in Temic transceiver mode.	0
Reserved	[2:1]	Reserved	00
Mode select	[0]	select the mode of operation as 0 : FIR Mode 1 : MIR Mode	0

38.6.3 IRDA INTERRUPT / DMA CONFIGURATION REGISTER (IRDA_CNF)

Register	Address	R/W	Description	Reset Value
IRDA_CNF	0x7F00_7008	R/W	IrDA Interrupt / DMA Configuration Register	0x00

IRDA _CNF	Bit	Description	Initial State
Reserved	[7:4]	Reserved	0
DMA Enable	[3]	0 : DMA Disable 1 : DMA Enable	0
DMA Mode	[2]	0 : Tx DMA 1 : Rx DMA	0
Reserved	[1]	Reserved	0
Interrupt Enable	[0]	The bit 0 enables Interrupt output signal. 0 : Disable 1 : Enable	0

38.6.4 IRDA INTERUPT ENALBLE REGISTER(IRDA_IER)

Register	Address	R/W	Description	Reset Value
IRDA _IER	0x7F00_700C	R/W	IrDA Interrupt Enable Register	0x00

IRDA_IER	Bit	Description	Initial State
Last byte to Rx FIFO	[7]	Enables state indication interrupt. When 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 will be activated when the last byte of the received data frame comes into the demodulation block and then 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 when the microcontroller reads the last byte of the frame from the RX FIFO.	0
Tx FIFO below threshold	[1]	Bit 1 enables an TX FIFO below threshold level interrupt when 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 when the RX FIFO is equal to or above the threshold level.	0

38.6.5 IRDA INTERUPT IDENTIFICATION REGISTER(IRDA_IIR)

Register	Address	R/W	Description	Reset Value
IRDA _IIR	0x7F00_7010	R	IrDA Interrupt Identification Register	0x00

IRDA _IIR	Bit	Description	Initial State
Last byte to Rx FIFO	[7]	Last byte write to RX FIFO interrupt pending. When 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 when 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 when the source of the error is cleared.	0
Tx Underrun	[5]	Transmit under-run interrupt pending. When 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' when the demodulation block detects the last byte of a received frame and the CRC decoding is finished. Bit 4 is cleared when it is read.	0
Rx overrun	[3]	RX FIFO over-run interrupt. When corresponding interrupt enable bit is set, bit3 is active, bit 3 is set to '1' when 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. When corresponding interrupt enable bit is active, it is set to '1' when the CPU reads the last byte of a frame from the RX FIFO. It is cleared when it is read.	0
Tx FIFO below threshold	[1]	TX FIFO below threshold interrupt pending. Bit 1 is set to '1' when 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' when the receiver FIFO level is equal to or above its threshold level.	0

38.6.6 IRDA LINE STATUS REGISTER(IRDA_LSR)

Register	Address	R/W	Description	Reset Value
IRDA _LSR	0x7F00_7014	R	IrDA Line Status Register	0x83

IRDA_LSR	Bit	Description	Initial State
Tx empty	[7]	Transmitter empty. This bit is set to '1' when TX FIFO is empty and the transmitter front-end is idle.	1
Reserved	[6]	Reserved	0
Received last byte from Rx FIFO	[5]	Last byte received from RX FIFO. It is set to a '1' when the microcontroller reads the last byte of a frame from the RX FIFO and cleared when the MCU reads the IRDA_LSR register.	0
Frame length error	[4]	Frame length error. It is set to '1' when a frame exceeding the maximum frame length predefined by IRDA_RXFLL and IRDA_RXFLH register is received. This bit is cleared when the microcontroller reads the IRDA_LSR register. When 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' when the IRDA_LSR register is read by the microcontroller.	0
PHY error	[3]	PHY error. In FIR mode, It is set to a '1' when 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 when microcontroller reads the IRDA_LSR register.	0
CRC error	[2]	CRC error. Bit 2 is set to '1' when a bad IrDA CRC is detected on data receive. It is cleared to '0' when microcontroller reads the LSR register.	0
Reserved	[1]	Reserved	1
Rx FIFO empty	[0]	RX FIFO empty. It indicates that the RX FIFO is empty. When the state of RX FIFO turns into empty, it is set to '1'. When the RX FIFO is not empty, it is set to '0'.	1

38.6.7 IRDA FIFO CONTROL REGISTER(IRDA_FCR)

Register	Address	R/W	Description	Reset Value
IRDA _FCR	0x7F00_7018	R/W	IrDA FIFO Control Register	0x00

IRDA _FCR	Bit	Description	Initial State
Rx FIFO Trigger level select	[7:6]	Receiver FIFO triggers level selection. Bit 7 / Bit 6 / 64-byte RX FIFO 0 / 0 / 01 0 / 1 / 16 1 / 0 / 32 1 / 1 / 56	00
Reserved	[5]	Must set ‘1’	0
TX FIFO Clear Notification	[4]	This bit will be activated when the FIFO clear is over. This bit is cleared by the CPU reads this register.	0
RX FIFO Clear Notification	[3]	This bit will be activated when the FIFO clear is over. This bit is cleared by the CPU reads this register.	0
Tx FIFO reset	[2]	TX FIFO reset. When 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.	0
Rx FIFO reset	[1]	RX FIFO reset. When 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.	0
FIFO enable	[0]	FIFO enabled. When 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.	0

38.6.8 IRDA PREAMBLE LENGTH REGISTER(IRDA_PLR)

Register	Address	R/W	Description	Reset Value
IRDA _PLR	0x7F00_701C	R/W	IrDA Preamble Length Register	0x12

REG_PLR	Bit	Description	Initial State
Reserved	[7:6]	Reserved	00
TX FIFO trigger level select	[5:4]	Transceiver FIFO triggers level selection. Bit 5 / Bit 4 / 64-byte FIFO 0 / 0 / Reserved 0 / 1 / 48 1 / 0 / 32 1 / 1 / **08** **Note:** Tx Trigger level value means total number data empty.	01
Number of start flags in MIR mode	[3:0]	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**.	0010

38.6.9 IRDA RECEIVER & TRANSMITTER BUFFER REGISTER(IRDA_RBR)

Register	Address	R/W	Description	Reset Value
IRDA_RBR IRDA_THR	0x7F00_7020	R/W	IrDA Receiver & Transmitter Buffer Register	0x00

IRDA _RBR	Bit	Description	Initial State
Rx/Tx data	[7:0]	Received data (When read data) Data to transmit (When write data)	0x00

38.6.10 IRDA TOTAL NUMBER OF DATA BYTES REMAINED IN TX FIFO(IRDA_TXNO)

Register	Address	R/W	Description	Reset Value
IRDA _TXNO	0x7F00_7024	R	The total number of data bytes remained in Tx FIFO	0x00

IRDA _TXNO	Bit	Description	Initial State
Tx data total number	[7:0]	The total number of data bytes remained in Tx FIFO	0x00

38.6.11 IRDA TOTAL NUMBER OF DATA BYTES REMAINED IN RX FIFO(IRDA_RXNO)

Register	Address	R/W	Description	Reset Value
IRDA _RXNO	0x7F00_7028	R	The total number of data bytes remained in Rx FIFO	0x00

IRDA _RXNO	Bit	Description	Initial State
Rx data total number	[7:0]	The total number of data bytes remained in Rx FIFO.	00

38.6.12 IRDA TRANSMIT FRAME-LENGTH REGISTER LOW(IRDA_TXFLL)

Register	Address	R/W	Description	Reset Value
IRDA _TXFLL	0x7F00_702C	R/W	IrDA Transmit Frame-Length Register Low	0x00

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

38.6.13 IRDA TRANSMIT FRAME-LENGTH REGISTER HIGH(IRDA_TXFLH)

Register	Address	R/W	Description	Reset Value
IRDA _TXFLH	0x7F00_7030	R/W	IrDA Transmit Frame-Length Register High	0x00

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

38.6.14 IRDA RECEIVER FRAME-LENGTH REGISTER LOW(IRDA_RXFLL)

Register	Address	R/W	Description	Reset Value
IRDA _RXFLL	0x7F00_7034	R/W	IrDA Receive Frame-Length Register Low	0x00

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

38.6.15 IRDA RECEIVER FRAME-LENGTH REGISTER HIGH(IRDA_RXFLH)

Register	Address	R/W	Description	Reset Value
IRDA _RXFLH	0x7F00_7038	R/W	IrDA Receive Frame-Length Register High	0x00

IRDA _RXFLH	Bit	Description	Initial State
Reserved	[7:6]	Reserved	00
Rx frame length high	[5:0]	TXFLL stores the upper 6 bits of the maximum byte number of the frame to be received.	

38.6.16 IRDA INTERRUPT CLEAR REGISTER (IRDA_INTCLR)

Register	Address	R/W	Description	Reset Value
IRDA _INTCLR	0x7E00_903C	W	IrDA Interrupt Clear Register	

IRDA _INTCLR	Bit	Description	Initial State
Interrupt Clear	[31:0]	Read undefined. Write any value to make IrDA interrupt clear.	

39 ADC & TOUCH SCREEN INTERFACE

This chapter describes the functions and usage of ADC & Touch Screen interface S3C6410X RISC microprocessor.

39.1 OVERVIEW

The 10-bit/12-bit CMOS ADC (Analog to Digital Converter) is a recycling type device with 8-channel analog inputs. It converts the analog input signal into 10-bit/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 can control input pads (XP, XM, YP, and YM) to obtain X/Y-position on the external touch screen device. Touch Screen Interface contains three main blocks; these are touch screen pads control logic, ADC interface logic and interrupt generation logic.

39.2 FEATURES

The ADC & Touch Screen interface includes the following features:

- Resolution: 10-bit/12-bit
- Differential Nonlinearity: ± 2.0 LSB
- Integral Nonlinearity: ± 4.0 LSB
- Maximum Conversion Rate: 1MSPS
- 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

39.3 ADC & TOUCH SCREEN INTERFACE OPERATION

39.3.1 BLOCK DIAGRAM

Figure 39-1 shows the functional block diagram of A/D converter and Touch Screen Interface.

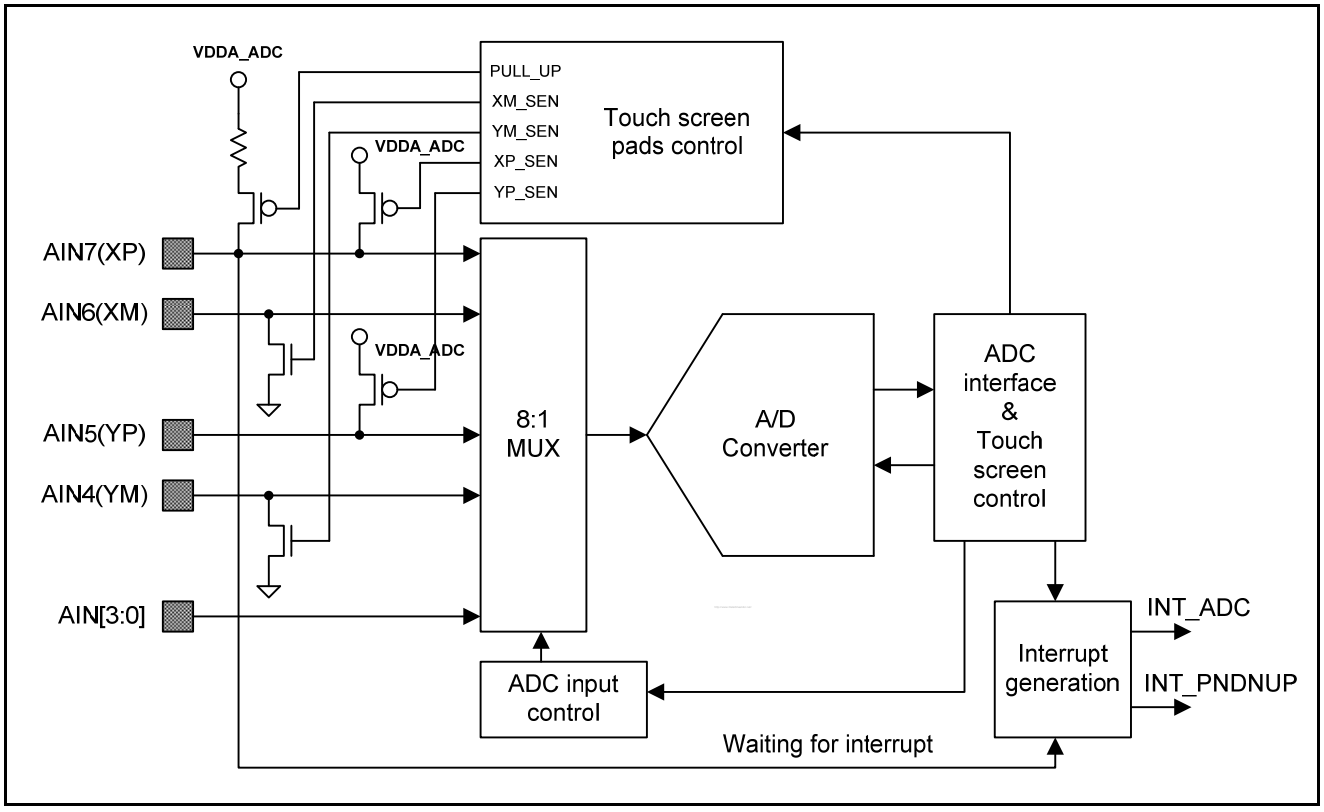


Figure 39-1. ADC and Touch Screen Interface Functional Block Diagram

NOTE

When Touch Screen device is used, XM or YM is only connected to ground for Touch Screen I/F.

When Touch Screen device is not used, XM or YM is connecting Analog Input Signal for Normal ADC conversion.

39.4 FUNCTION DESCRIPTIONS

39.4.1 A/D CONVERSION TIME

When the PCLK frequency is 50MHz and the prescaler value is 49, total 10-bit or 12-bit conversion time is as follows.

A/D converter freq. = 50MHz/(49+1) = 1MHz

Conversion time = 1/(1MHz / (5cycles)) = 1/200KHz =5 us

Note:
This A/D converter was designed to operate at maximum 5MHz clock, so the conversion rate can go up to 1MSPS.

39.4.2 TOUCH SCREEN INTERFACE MODE

1. Normal Conversion Mode (AUTO_PST=0, XY_PST=0)

The operation of this mode is identical with AIN0~AIN3'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	AIN5	Hi-z
Y-position measurement	AIN7	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

39.4.3 STANDBY MODE

Standby mode is activated when ADCCON [2] is set to '1'. In this mode, A/D conversion operation is halted and ADCDAT0, ADCDAT1 register contains the previous converted data.

39.5 PROGRAMMING NOTES

1. 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.

2. 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.

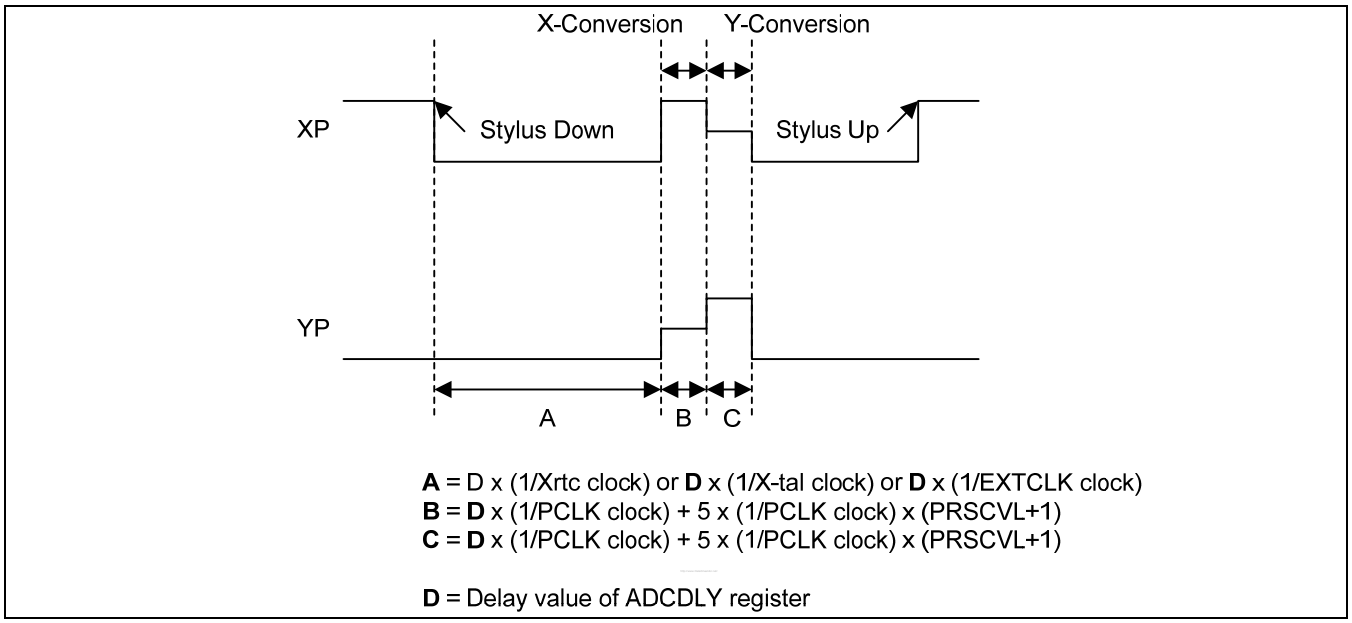


Figure 39-2. ADC and Touch Screen Operation signal (Touch Normal Operation)

3. If pen down/up interrupt is used as a wakeup source in STOP mode, XY_PST bit (ADCTSC [1:0]) should be set to waiting for interrupt mode (2b'11). To choose stylus pen up or pen down wakeup, UD_SEN bit (ADCTSC [8]) is used.

39.6 ADC AND TOUCH SCREEN INTERFACE SPECIAL REGISTERS

39.6.1 REGISTER MAP

Register	Address	R/W	Description	Reset Value
ADCCON	0x7E00_B000	R/W	ADC Control Register	0x0000_3FC4
ADCTSC	0x7E00_B004	R/W	ADC Touch Screen Control Register	0x0000_0058
ADCDLY	0x7E00_B008	R/W	ADC Start or Interval Delay Register	0x0000_00FF
ADCDAT0	0x7E00_B00C	R	ADC Conversion Data Register	-
ADCDAT1	0x7E00_B010	R	ADC Conversion Data Register	-
ADCUPDN	0x7E00_B014	R/W	Stylus Up or Down Interrupt Register	0x0000_0000
ADCCLRINT	0x7E00_B018	W	Clear ADC Interrupt	-
Reserved	0x7E00_B01C	-	reserved	-
ADCCLRINTPNDNUP	0x7E00_B020	W	Clear Pen Down/Up Interrupt	-

39.6.2 ADC CONTROL REGISTER (ADCCON)

Register	Address	R/W	Description	Reset Value
ADCCON	0x7E00B000	R/W	ADC Control Register	0x3FC4

ADCCON	Bit	Description	Initial State
RESSEL	[16]	A/D converter 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 = Disable 1 = Enable	0
PRSCVL	[13:6]	A/D converter prescaler value Data value: 5 ~ 255 **NOTE:** Note that division factor is (N+1) when the prescaler value is N. 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
SEL_MUX	[5:3]	Analog input channel select 000 = AIN 0 001 = AIN 1 010 = AIN 2 011 = AIN 3 100 = YM 101 = YP 110 = XM 111 = XP	0
STDBM	[2]	Standby mode select 0 = Normal operation mode 1 = Standby mode	1
READ_ START	[1]	A/D conversion start by read 0 = Disable start by read operation 1 = Enable 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 cleared after the start-up.	0

39.6.3 ADC TOUCH SCREEN CONTROL REGISTER (ADCTSC)

Register	Address	R/W	Description	Reset Value
ADCTSC	0x7E00B004	R/W	ADC Touch Screen Control Register	0x58

ADCTSC	Bit	Description	Initial State
Reserved	[11:9]		000
UD_SEN	[8]	Detect Stylus Up or Down status. 0 = Detect Stylus Down Interrupt Signal. 1 = Detect Stylus Up Interrupt Signal.	0
YM_SEN	[7]	YM to GND Switch Enable 0 = Switch disable (YM = AIN4, Hi-z) 1 = Switch enable (YM = VSSA_ADC)	0
YP_SEN	[6]	YP to VDD Switch Enable 0 = Switch enable (YP=VDDA_ADC) 1 = Switch disable (YP=AIN5, Hi-z)	1
XM_SEN	[5]	XM to GND Switch Enable 0 = Switch disable (XM = AIN6, Hi-z) 1 = Switch enable (XM=VSSA_ADC)	0
XP_SEN	[4]	XP to VDD Switch Enable 0 = Switch enable (XP=VDDA_ADC) 1 = Switch disable (XP=AIN7, Hi-z)	1
PULL_UP	[3]	Pull-up Switch Enable 0 = XP Pull-up Enable. 1 = XP Pull-up Disable.	1
AUTO_PST	[2]	Automatic sequencing conversion of X-Position and Y-Position 0 = Normal ADC conversion. 1 = Auto Sequential measurement of X-position, 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

Note: 1) While waiting for Touch screen Interrupt, XP_SEN bit must be set to '1', namely 'Switch 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 & Sequential X/Y Position conversion.
3) If you don't use AIN[7], you must tie AIN [7] to VDDA_ADC or ADCTSC register must be setting to 0xd3.

Touch screen pin conditions in X/Y position conversion.

	XP	XM	YP	YM	ADC ch. select
X Position	Vref	GND	AIN[5]	Hi-Z	YP
Y Position	AIN[7]	Hi-Z	Vref	GND	XP

39.6.4 ADC START DELAY REGISTER (ADCDLY)

Register	Address	R/W	Description	Reset Value
ADCDLY	0x7E00B008	R/W	ADC Start or Interval Delay Register	0x00ff

ADCDLY	Bit	Description	Initial State
FILCLKsrc	[16]	ADCDLY clock source. In waiting for interrupt mode, FILCLKsrc is used as delay filter clock source. 0 = External input clock. (XXTI or XEXTCLK) 1 = RTC clock. (XrtcXTI)	0
DELAY	[15:0]	1) In case of ADC conversion mode (Normal, Separate, Auto conversion); ADC conversion is delayed by counting this value. Counting clock is PCLK. → ADC conversion delay value. 2) In case of waiting for Interrupt mode; when stylus down occurs in waiting for interrupt mode, it generates interrupt signal (INT_PNDNUP) at interval of several ms for Auto X/Y position conversion. If this interrupt occurs in STOP mode, it generates Wake-Up signal, having interval (several ms), for Exiting STOP MODE. **Note:** Do not use Zero value(0x0000)	0x00ff

Note: Before ADC conversion, Touch screen uses X-tal clock (3.68MHz).
During ADC conversion PCLK (Max. 50MHz) is used.

39.6.5 ADC CONVERSION DATA REGISTER (ADCDAT0)

Register	Address	R/W	Description	Reset Value
ADCDAT0	0x7E00B00C	R	ADC Conversion Data Register	-

ADCDAT0	Bit	Description	Initial State
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_12	[11:10]	When A/D resolution is 12bit, this is X-position conversion data [11:0] value.	-
XPDATA (Normal ADC)	[9:0]	X-Position Conversion data value (Includes Normal ADC Conversion data value) Data value : 0x0 ~ 0x3FF	-

39.6.6 ADC CONVERSION DATA REGISTER (ADCDAT1)

Register	Address	R/W	Description	Reset Value
ADCDAT1	0x7E00B010	R	ADC Conversion Data Register	-

ADCDAT1	Bit	Description	Initial State
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_12	[11:10]	When A/D resolution is 12bit, this is Y-position conversion data [11:0] value.	-
YPDATA	[9:0]	Y-Position Conversion data value Data value : 0x0 ~ 0x3FF	-

39.6.7 ADC TOUCH SCREEN UP-DOWN REGISTER (ADCUPDN)

Register	Address	R/W	Description	Reset Value
ADCUPDN	0x7E00B014	R/W	Stylus Up or Down Interrupt Register	0x0

ADCUPDN	Bit	Description	Initial State
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 been 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 been occurred.	0

39.6.8 ADC TOUCH SCREEN INTERRUPT CLEAR REGISTER

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 will clear up the relevant interrupts asserted. When it is read, undefined value will be returned.

Register	Address	R/W	Description	Reset Value
ADCCLRINT	0x7E00B018	W	Clear ADC Interrupt	-

ADCCLRINT	Bit	Description	Initial State
INT_ADC_CLR	[0]	INT_ADC interrupt clear	-

Register	Address	R/W	Description	Reset Value
ADCCLRINTPNDNUP	0x7E00B020	W	Clear Pen Down/Up Interrupt	-

ADCCLRINTPNDNUP	Bit	Description	Initial State
INT_PNDNUP_CLR	[0]	INT_PNDNUP interrupt clear	-

40 KEYPAD INTERFACE

40.1 OVERVIEW

The Key Pad Interface block in S3C6410X facilitates 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. When any of the interrupt from row lines occurs, the software will scan the column lines using the proper procedure to detect one or multiple key press or release.

It provides interrupt status register bits when key pressed or key released or both cases (when two interrupt conditions are enabled). To prevent the switching noises, internal debouncing filter is provided.

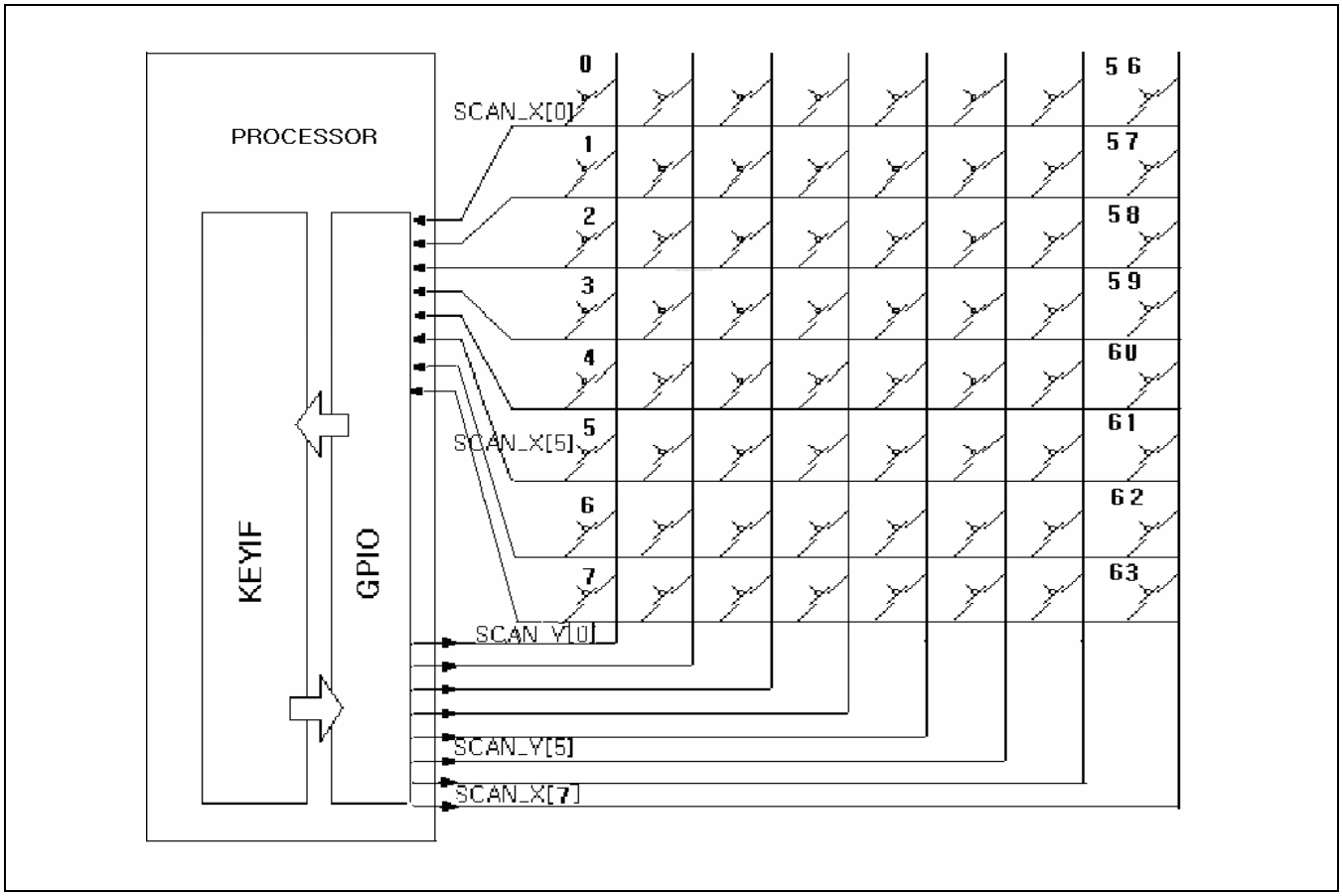


Figure 40-1. Key Matrix Interface External Connection Guide

40.2 DEBOUNCING FILTER

The debouncing filter is supported for keypad interrupt of any key input. The filtering width is approximately 62.5usec (“FLT_CLK” two-clock, when the FLT_CLK is 32 kHz). The keypad interrupt (key pressed or key released) to the CPU is an ANDed signal of the all row input lines after filtering.

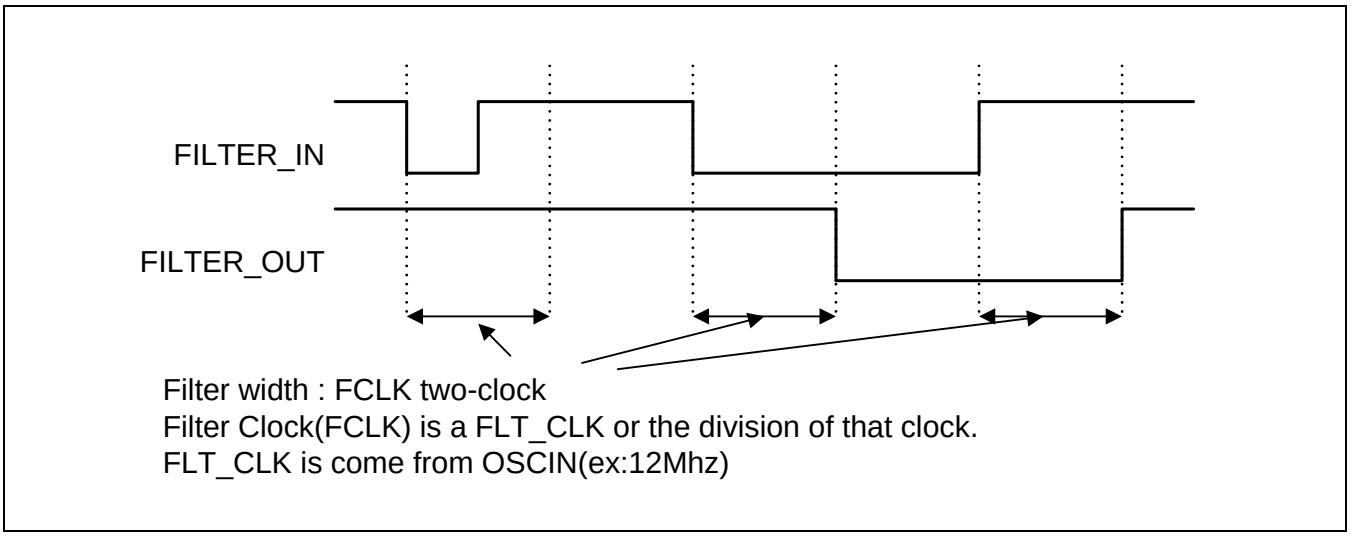


Figure 40-2. Internal Debouncing Filter Operation

40.2.1 FILTER CLOCK

KEYPAD interface debouncing filter clock (FCLK) is divided from FLT_CLK that is OSC_IN. User can set compare value for 10-bit up-counter (KEYIFFC). When 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 .

40.3 PIN MULTIPLEXING

In S3C6410X, the Keypad interface input, output ports are multiplexed with GPIO K and L ports. The input and output port counts are controlled by bit up to 8 inputs and 8 outputs.

Table 40-1. Keypad interface pin multiplexing

GPIO port 1	GPIO port 2	Keypad interface port	I/O
GPIOK[8]	GPION[0]	ROW_IN[0]	I
GPIOK[9]	GPION[1]	ROW_IN[1]	I
GPIOK[10]	GPION[2]	ROW_IN[2]	I
GPIOK[11]	GPION[3]	ROW_IN[3]	I
GPIOK[12]	GPION[4]	ROW_IN[4]	I
GPIOK[13]	GPION[5]	ROW_IN[5]	I
GPIOK[14]	GPION[6]	ROW_IN[6]	I
GPIOK[15]	GPION[7]	ROW_IN[7]	I
GPIOL[0]	GPIOH[0]	COL_OUT [0]	O
GPIOL[1]	GPIOH[1]	COL_OUT [1]	O
GPIOL[2]	GPIOH[2]	COL_OUT [2]	O
GPIOL[3]	GPIOH[3]	COL_OUT [3]	O
GPIOL[4]	GPIOH[4]	COL_OUT [4]	O
GPIOL[5]	GPIOH[5]	COL_OUT [5]	O
GPIOL[6]	GPIOH[6]	COL_OUT [6]	O
GPIOL[7]	GPIOH[7]	COL_OUT [7]	O

40.4 WAKEUP SOURCE

When the Key input is used for wakeup source from STOP or SLEEP mode, KEYPAD I/F register setting is not required. GPIO register setting (GPKCON) for KEYPAD I/F and SYSCON register (PWR_CFG) for masking are required for wakeup.

40.5 SOFTWARE KEY SCANNING PROCEDURE

At initial state, all column lines (outputs) are low level. When no key pressed state, all row lines (inputs) are high (used pull-up pads). When any key is pressed, the corresponding row and column lines are connected 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. In 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.

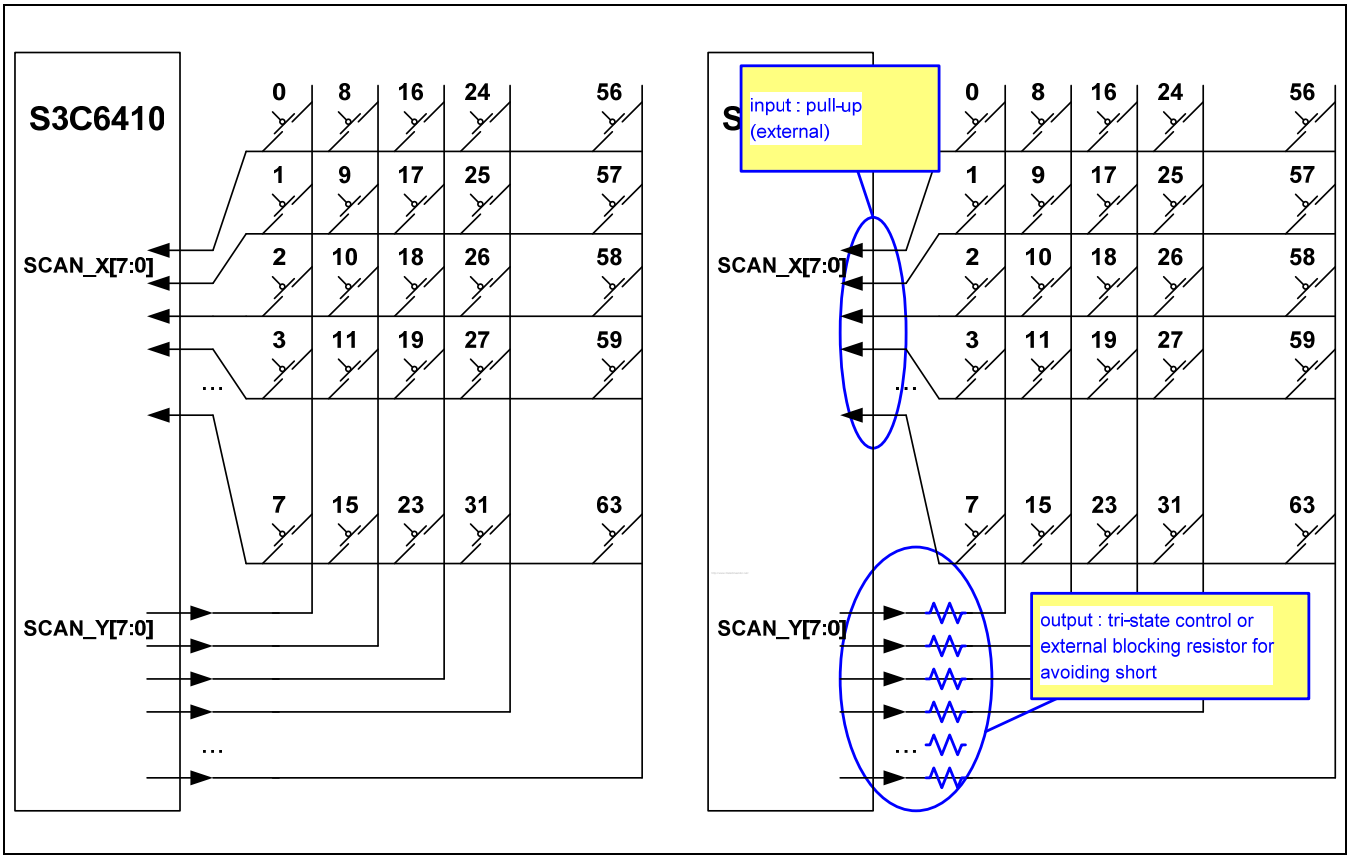


Figure 40-3. Keypad scanning procedure I

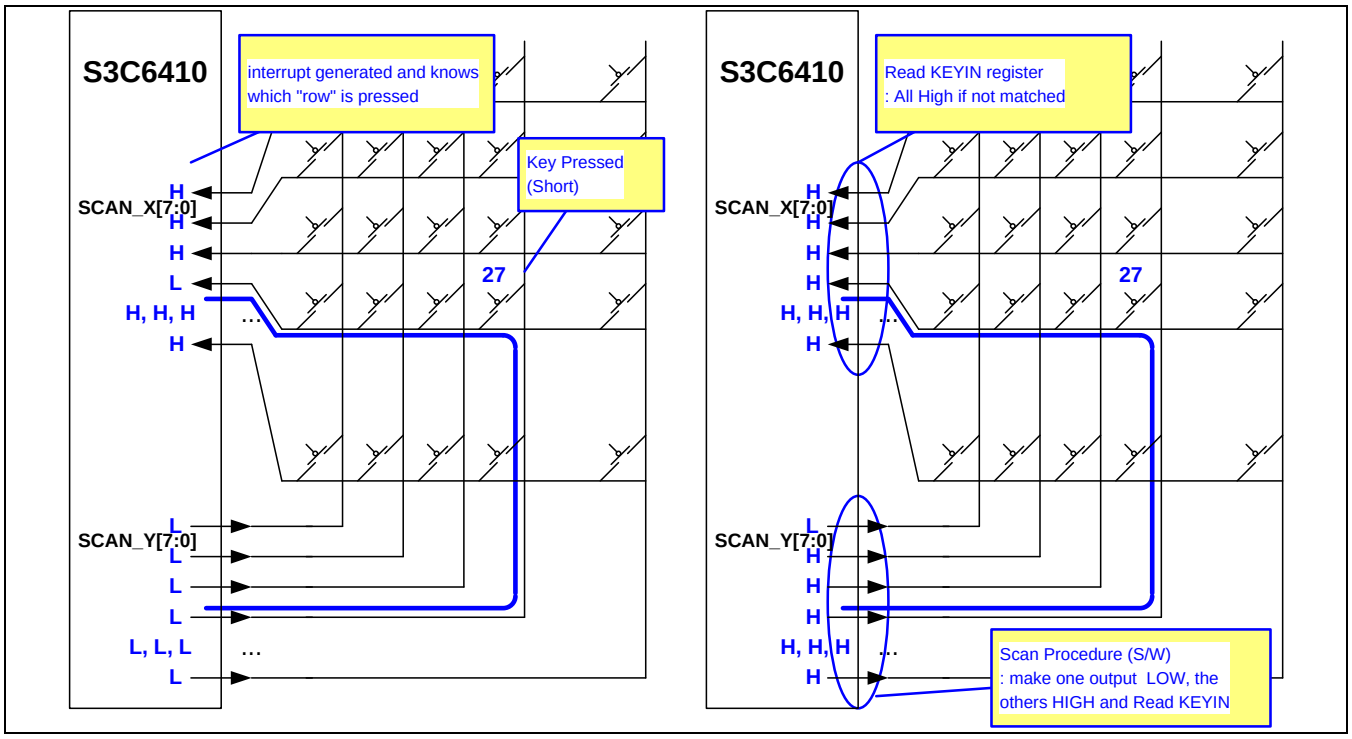


Figure 40-4. Keypad scanning procedure II

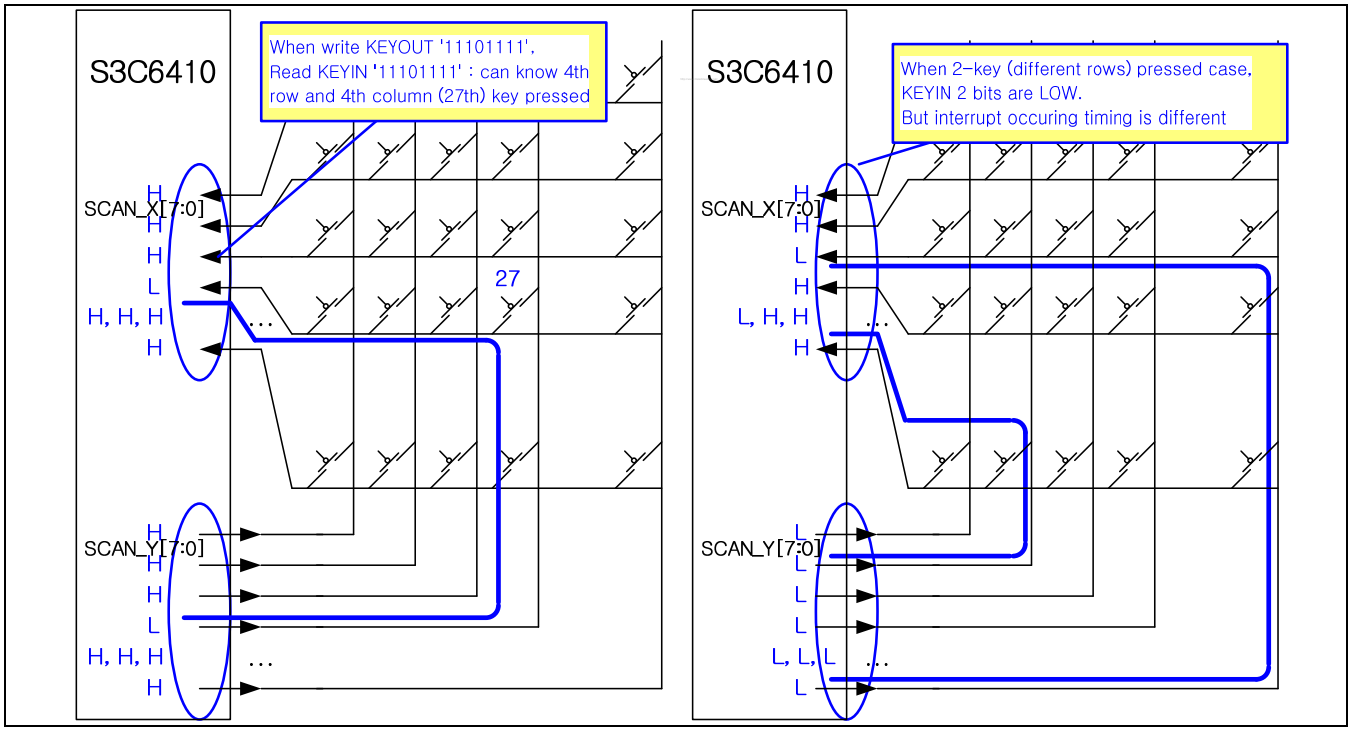


Figure 40-5.Keypad scanning procedure III

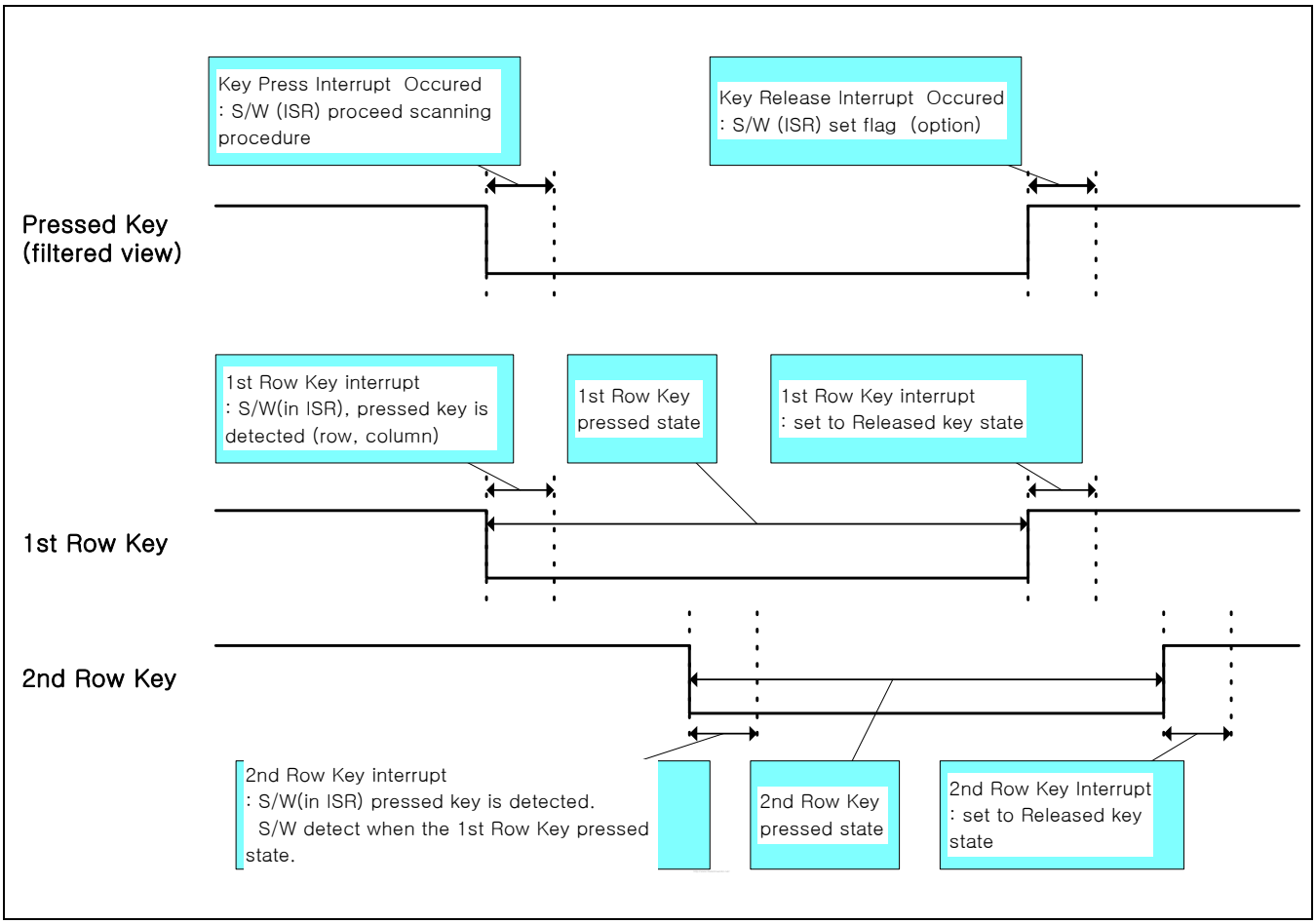


Figure 40-6. Keypad Scanning Procedure when the two-key pressed with different row

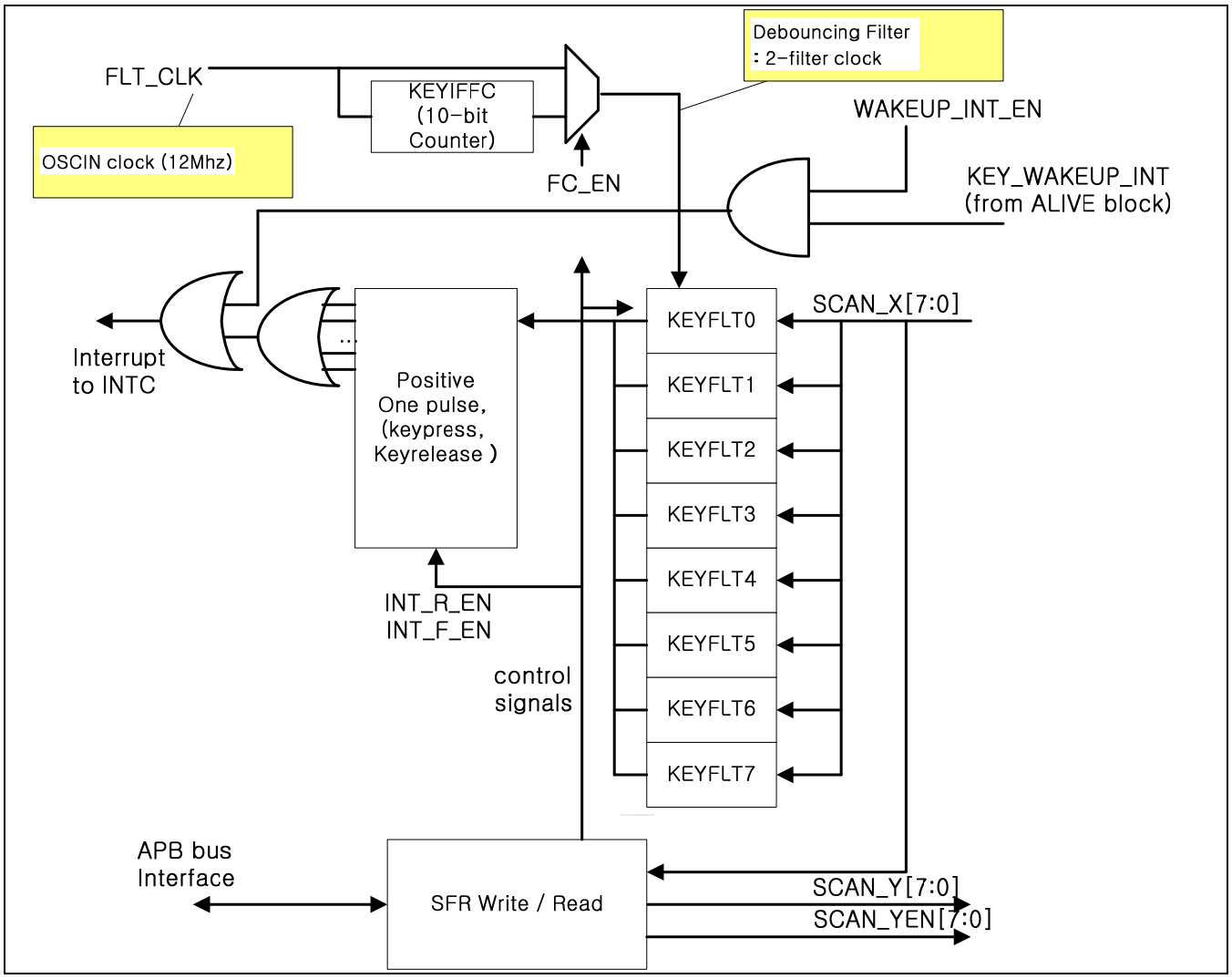


Figure 40-7. Keypad I/F block diagram

40.6 KEYPAD INTERFACE REGISTER

40.6.1 MEMORY MAP

Register	Address	R/W	Description	Reset Value
KEYIFCON	0x7E00A000	R/W	KEYPAD interface control register	0x00000000
KEYIFSTSCLR	0x7E00A004	R/W	KEYPAD interface status and clear register	0x00000000
KEYIFCOL	0x7E00A008	R/W	KEYPAD interface column data output register	0x0000FF00
KEYIFROW	0x7E00A00C	R	KEYPAD interface row data input register	Reflects input ports
KEYIFFC	0x7E00A010	R/W	KEYPAD interface debouncing filter clock division register	0x00000000

40.7 REGISTER DESCRIPTION

40.7.1 KEYPAD INTERFACE CONTROL REGISTERS (KEYIFCON)

Register	address	R/W	Description	Reset Value
KEYIFCON	0x7E00A000	R/W	KEYPAD interface control register	0x00

KEYIFCON	Bit	Description
Reserved	[31:5]	-
WAKEUP_INT_EN	[4]	KEYPAD input Stop/ Sleep/Idle mode wakeup interrupt enable. Wakeup Interrupt signal is to VIC. ‘0’=Disable, ‘1’=Key input Low Level (while key-pressed) wakeup interrupt enable-
FC_EN	[3]	10-bit counter (for Debouncing digital filter clock) enable ‘0’=Disable : Disable use division counter ‘1’=Enable : use division counter
DF_EN	[2]	KEYPAD input port debouncing filter enable ‘0’=Disable, ‘1’=Enable
INT_R_EN	[1]	KEYPAD input port rising edge (key-released) interrupt ‘0’=Disable, ‘1’=Enable
INT_F_EN	[0]	KEYPAD input port falling edge (key-pressed) interrupt ‘0’=Disable, ‘1’=Enable

NOTE: Both edge interrupt is selected when both INT_F_EN and INT_R_EN are set.

40.7.2 KEYPAD INTERRUPT STATUS AND CLEAR REGISTER

Register	address	R/W	Description	Reset Value
KEYIFSTSCLR	0x7E00A004	R/W	KEYPAD interface status and clear register	0x0

KEYIFSTSCLR	Bit	Description
P_INT	[7:0]	KEYPAD input “press” interrupts (falling edge) status(read) and clear(write) Read : ‘1’=Pressed interrupt occurred, ‘0’=Does not occur Write : Pressed interrupt is cleared when write data is ‘1’ The P_INT[7:0] indicate that each key pressed from 0 to 7 has a dedicated interrupt to it from P_INT[0] to P_INT[7]
R_INT	[15:8]	KEYPAD input “release” interrupts (rising edge) status(read) and clear(write) Read : ‘1’=Released interrupt occurred, ‘0’=Does not occur Write : Released interrupt is cleared when write data is ‘1’ The R_INT[15:8] indicate that each key released from 0 to 7 has a dedicated interrupt to it from R_INT[8] to R_IN T[15]
Reserved	[31:16]	-

NOTE: Keypad wakeup interrupt is also cleared when the write access to the KEYIFSTSCLR.

40.7.3 KEYPAD INTERFACE COLUMN DATA OUTPUT REGISTER

Register	Address	R/W	Description	Reset Value
KEYIFCOL	0x7E00A008	R/W	KEYPAD interface column data output register	0x0000FF00

KEYIFCOL	Bit	Description
KEYIFCOL	[7:0]	KEYPAD interface column data output register
KEYIFCOLEN	[15:8]	KEYPAD interface column data output tri-state enable register Each bit is for each KEYIFCOL bit. ‘0’= Output pad tri-state buffer enable(Normal output), ‘1’= Output pad Tri-state buffer disable(High-Z output) (@ reset)
Reserved	[31:16]	-

40.7.4 KEYPAD INTERFACE ROW DATA INPUT REGISTER

Register	address	R/W	Description	Reset Value
KEYIFROW	0x7E00A00C	R	KEYPAD interface row data input register	Reflects input ports

KEYIFROW	Bit	Description
KEYIFROW	[7:0]	KEYPAD interface row data input register (read only)
Reserved	[31:8]	-

40.7.5 KEYPAD INTERFACE DEBOUNCING FILTER CLOCK DIVISION REGISTER

Register	address	R/W	Description	Reset Value
KEYIFFC	0x7E00A010	R/W	KEYPAD interface debouncing filter clock division register	0x0

KEYIFFC	Bit	Description
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 freq = FLT_CLK freq / ((KEYIFFC[9:0] + 1) x 2) (FLT_CLK is OSC_IN (10 ~ 12MHz))
Reserved	[31:10]	-

서식 있음: 글꼴 색: 자동

41 IIS MULTI AUDIO INTERFACE

41.1 OVERVIEW

IIS (Inter-IC Sound) is one of the popular digital audio interface. The bus only handles audio data, while the other signals, such as sub-coding and control, are transferred separately. It is possible to transmit data between two IIS bus. To minimize the number of pins required and to keep wiring simple, a 3-line serial bus is used consisting of a line for two time-multiplexed data channels, a word select line and a clock line.

IIS interface transmits or receives sound data from external stereo audio codec. For transmitting and receiving data, two 16x32-bit FIFOs (First-In-First-Out) data structures are included. DMA transfer mode for transmitting or receiving samples can be supported.

41.2 FEATURE

The IIS-BUS interface includes the following features:

- Up to 5.1ch IIS-bus for audio interface with DMA-based operation
- Serial, 8/16/24 bit per channel data transfers
- Supports sampling rate from 8kHz to 192kHz
- Supports IIS, MSB-justified and LSB-justified data format
- 64 Bytes Tx FIFO/64 Bytes Rx FIFO per each port

41.3 SIGNAL DESCRIPTIONS

IIS external pads are shared with other IPs like PCM, AC97 and etc. In order to use these pads for IIS, GPIO must be set before the IIS started. For more information, refer to the GPIO chapter of this manual for proper GPIO setting.

Name	Type	Source/Destination	Description
XmmcDATA1[4]	Input/Output	Pad	IIS Multi Audio serial clock(bit clock)
XmmcDATA1[5]	Input/Output	Pad	IIS Multi Audio Codec system clock or IISMultiEXTCLK source
XmmcDATA1[6]	Input/Output	Pad	IIS Multi Audio channel select(word select) clock
XmmcDATA[7]	Input	Pad	IIS Multi Audio serial data input
XspiMISO[1]	Output	Pad	IIS Multi Audio serial data output 0
XspiCLK[1]	Output	Pad	IIS Multi Audio serial data output 1
XspiCSn[1]	Output	Pad	IIS Multi Audio serial data output 2

41.4 BLOCK DIAGRAM

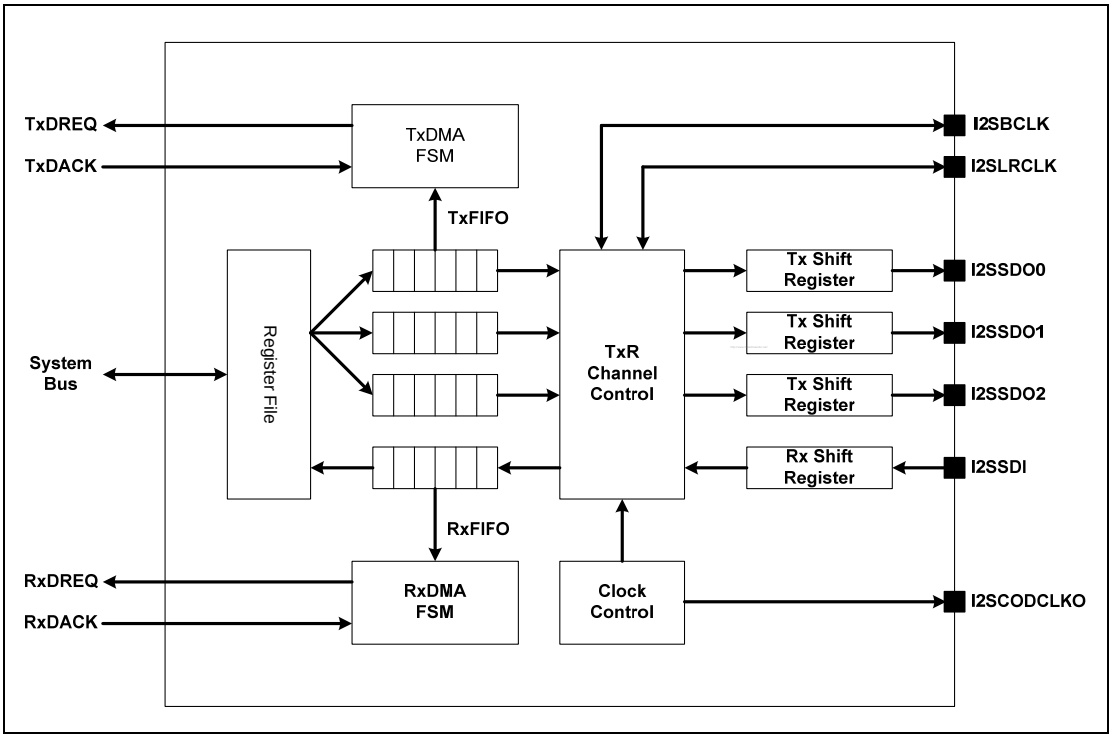


Figure 41-1. IIS-Bus Block Diagram

41.5 FUNCTIONAL DESCRIPTIONS

IIS interface consists of register bank, FIFOs, shift registers, clock control, DMA finite state machine, and channel control block as shown in Figure 41-1. Note that each FIFO has 32-bit width and 16 depths structure, which contains left/right channel data. Therefore FIFO access and data transfer are handled with left/right pair unit. Figure 41-1 shows the functional block diagram of IIS interface.

41.5.1 MASTER/SLAVE MODE

Master or slave mode can be selected by setting IMS bit of IISMOD register. In master mode, I2SSCLK(serial clock or bit clock) and I2SLRCLK(word select or channel select) are generated internally and supplied to external device. Therefore a I2SCDCLK(System clock) is needed for generating I2SSCLK and I2SLRCLK by dividing. The IIS pre-scaler (clock divider) is employed for generating a I2SCDCLK with divided frequency from internal system clock. In external master mode, the I2SCDCLK can be fed from IIS external. 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. Direction of I2SCDCLK (This is only auxiliary.) is not important. If IIS bus interface transmits clock signals to IIS codec, IIS bus is in master mode. But if IIS bus interface receives clock signal from IIS codec, IIS bus is in slave mode. TX/RX mode indicates the direction of data flow. If IIS bus interface transmits data to IIS codec, this is TX mode. Conversely, IIS bus interface receives data from IIS codec that is RX mode. Let's distinguish Master/Slave mode from TX/RX mode.

Figure 41-2 shows the route of the I2SCDCLK with internal master or external master mode setting in IIS clock control block and system controller. Note that RCLK indicates root clock and this clock can be supplied to external IIS codec chip at internal master mode.

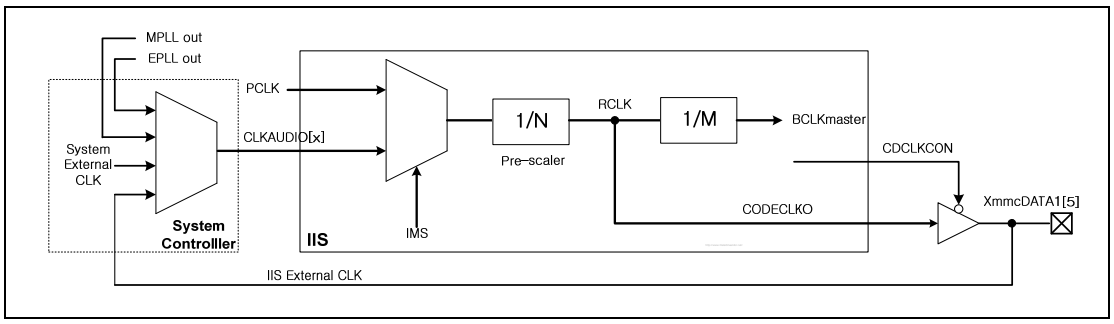


Figure 41-2. IIS Clock Control Block Diagram

41.5.2 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 when TXFIFO is not empty and the receiver DMA service request is activated when 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 must be performed.

* DMA request point

– TX mode : (FIFO is not full) & (TXDMACTIVE is active)

– RX mode : (FIFO is not empty) & (RXDMACTIVE is active)

NOTE:

It only supports single transfer in DMA mode.

41.6 AUDIO SERIAL DATA FORMAT

41.6.1 IIS-BUS FORMAT

The IIS 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 transmitting data that is synchronized with the leading edge has some restrictions.

The LR channel select line indicates the channel being transmitted. I2SLRCLK may be changed either on a trailing or leading edge of the serial clock, but it is not mandatory 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 clear the input for the next word.

41.6.2 MSB (LEFT) JUSTIFIED

MSB-Justified (Left-Justified) format is similar to IIS 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.

41.6.3 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 41-3 shows the audio serial format of IIS, 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).

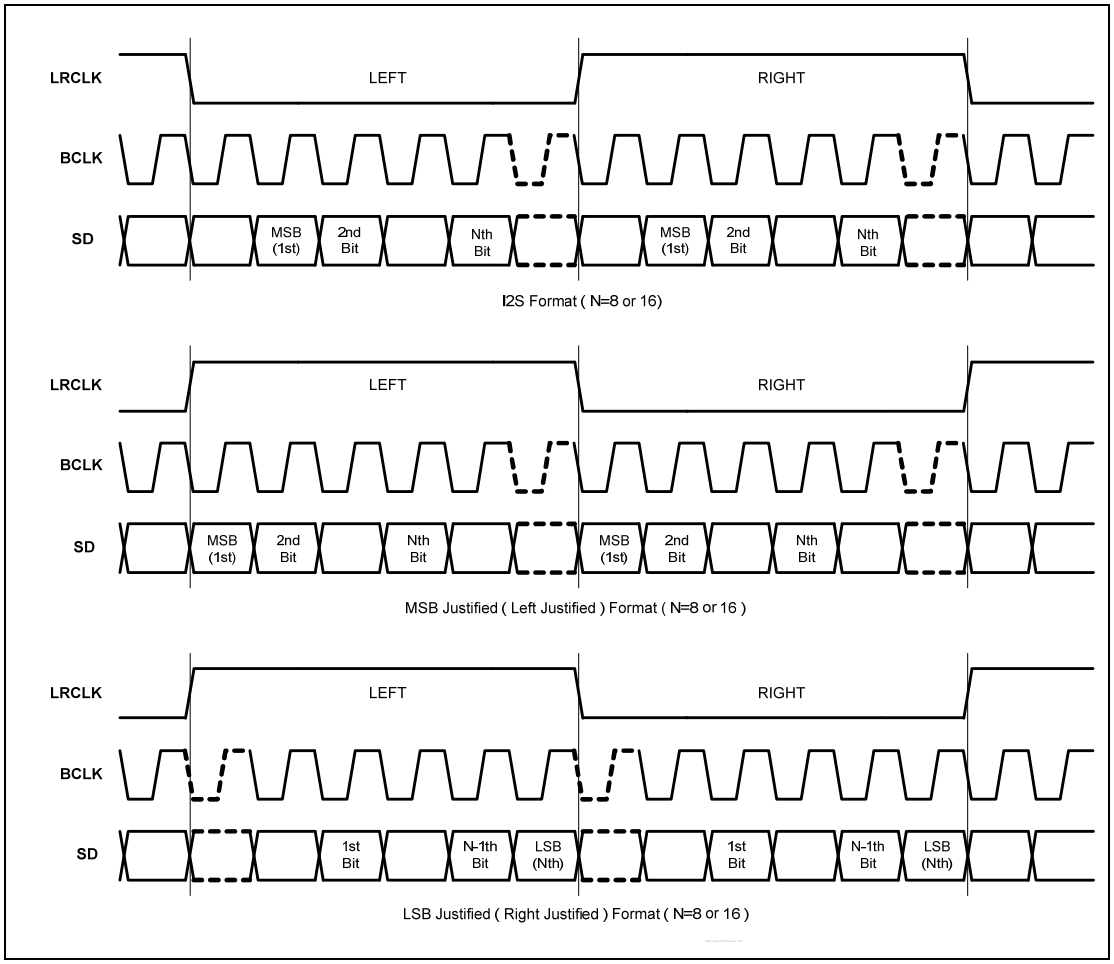


Figure 41-3. IIS Audio Serial Data Formats

41.7 SAMPLING FREQUENCY AND MASTER CLOCK

Master clock frequency (RCLK) can be selected by sampling frequency as shown in Table 41-1. Because RCLK is made by IIS pre-scaler, the pre-scaler value and RCLK type (256fs or 384fs or 512fs or 768fs) must be determined properly.

Table 41-1. CODEC clock (CODECLK = 256fs, 384fs, 512fs, 768fs)

IISLRCK (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	-	-

NOTE: *: fs represents sampling frequency. CODEC clock is fs * (256, 384, 512 or 768)

41.8 IIS CLOCK MAPPING TABLE

On selecting BFS, RFS, and BLC bits of I2SMOD register, you must refer to the following table. Table 41-2 shows the allowable clock frequency mapping relations.

Table 41-2. IIS 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 when BLC is 8-bit (b) Allowed when BLC is 16-bit (c) Allowed when BLC is 24-bit			

NOTE: Bit Clock Frequency ≥ fs * (bit length * 2). The codec clock is a multiple of the bit clock.

41.9 PROGRAMMING GUIDE

The IIS bus interface can be accessed either by the processor using programmed I/O instructions or by the DMA controller.

41.9.1 INITIALIZATION

1. Before you use IIS bus interface, you have to configure GPIOs to IIS mode and check signal's direction. I2SLRCLK, I2SSCLK and I2SCDCLK is inout-type. The each of I2SSDI and I2SSDO is a input and output.
2. You must select a clock source. S3C6410 has five clock sources. Those are MPLL, EPLL, PCLK, system external clock and IIS external clock. For more information please refer to Figure 41-2.

41.9.2 PLAY MODE (TX MODE) WITH DMA

1. TXFIFO is flushed before operation. If you don't distinguish Master/Slave mode from TX/RX mode, you must study Master/Slave mode and TX/RX mode. For more information please refer to Master/Slave chapter.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register) correctly.
3. To operate system in stability, the internal TXFIFO must be nearly full before transmission. DMA starts because of this reason.
4. IIS bus doesn't support the interrupt. You can only check the state by polling through accessing SFR.
5. If TXFIFO is full, you make I2SACTIVE be asserted.

41.9.3 RECORDING MODE (RX MODE) WITH DMA

1. RXFIFO is flushed before operation. If you don't distinguish between Master/Slave mode and TX/RX mode, you must study Master/Slave mode and TX/RX mode. For more information please refer to Master/Slave chapter.
2. Configure I2SMOD register and I2SPSR (IIS pre-scaler register) correctly.
3. To operate system in stability, the internal RXFIFO must have at least one data before DMA operation. Because of this reason, you make I2SACTIVE be asserted.
4. Check RXFIFO state by polling through accessing SFR.
5. If RXFIFO is not empty, start the RXDMACTIVE.

41.9.4 EXAMPLE CODE

TX CHANNEL

The I2S TX channel provides a single stereo compliant output. The transmit channel can operate 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 can write upto 16 left/right data samples After enabling the channel for transmission.
An Example sequence is as the following.
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

31 16 15 0

RIGHT CHANNEL	LEFT CHANNEL

Figure 41-4. TX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the TX FIFO for 24-bits/channel BLC as shown

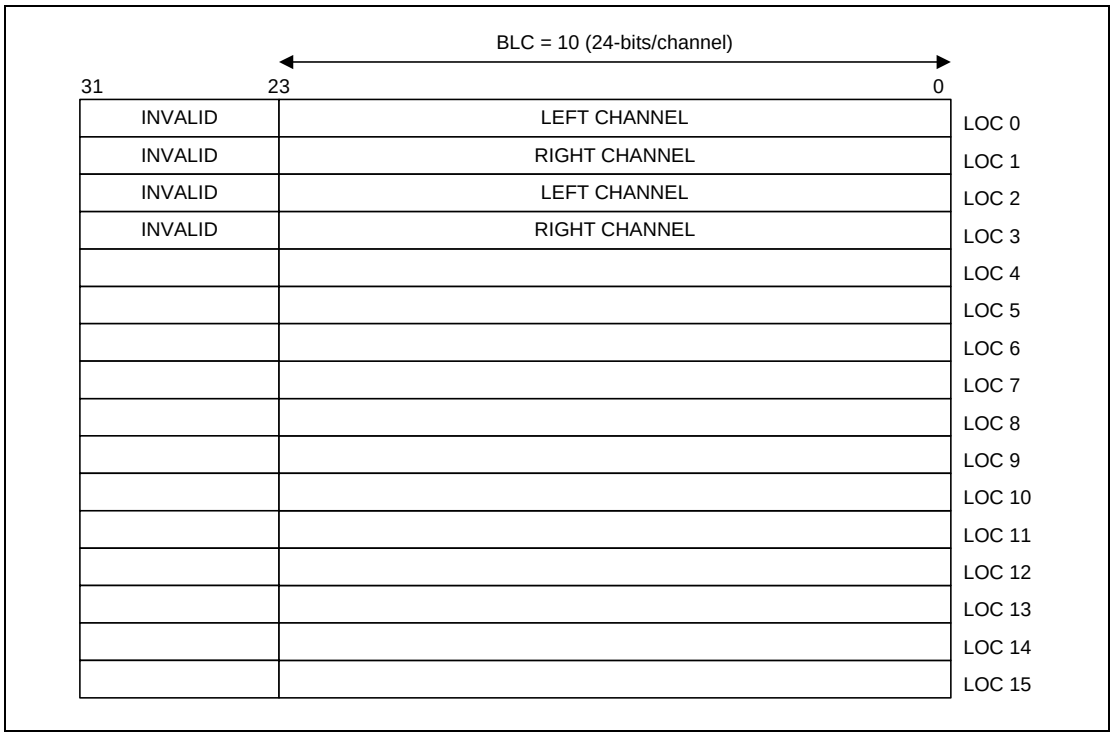


Figure 41-5. TX FIF0 Structure for BLC = 10 (24-bits/channel)

RX CHANNEL

The I2S RX channel provides a single stereo compliant output. The receive channel can operate in master or slave mode. Data is received from the input line and transferred into the RX FIFO. The processor can then read 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 can read upto 16 left/right data samples after enabling the channel for reception.

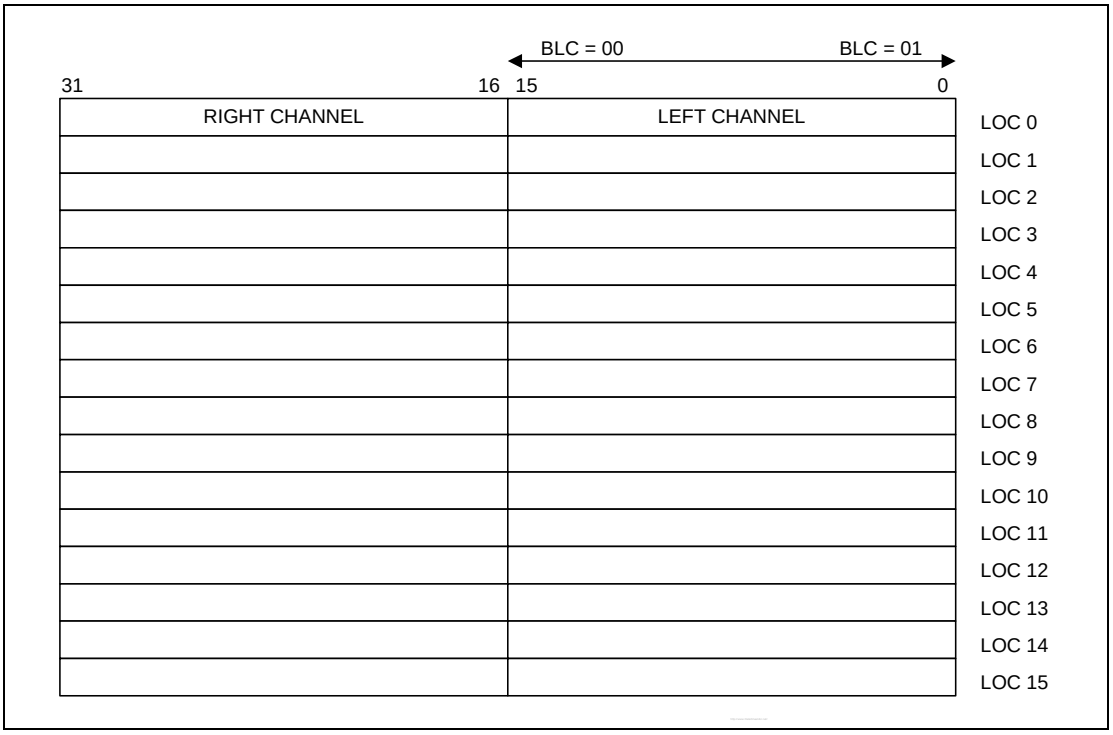


Figure 41-6. RX FIFO Structure for BLC = 00 or BLC = 01

The Data is aligned in the RX FIFO for 24-bits/channel BLC as shown

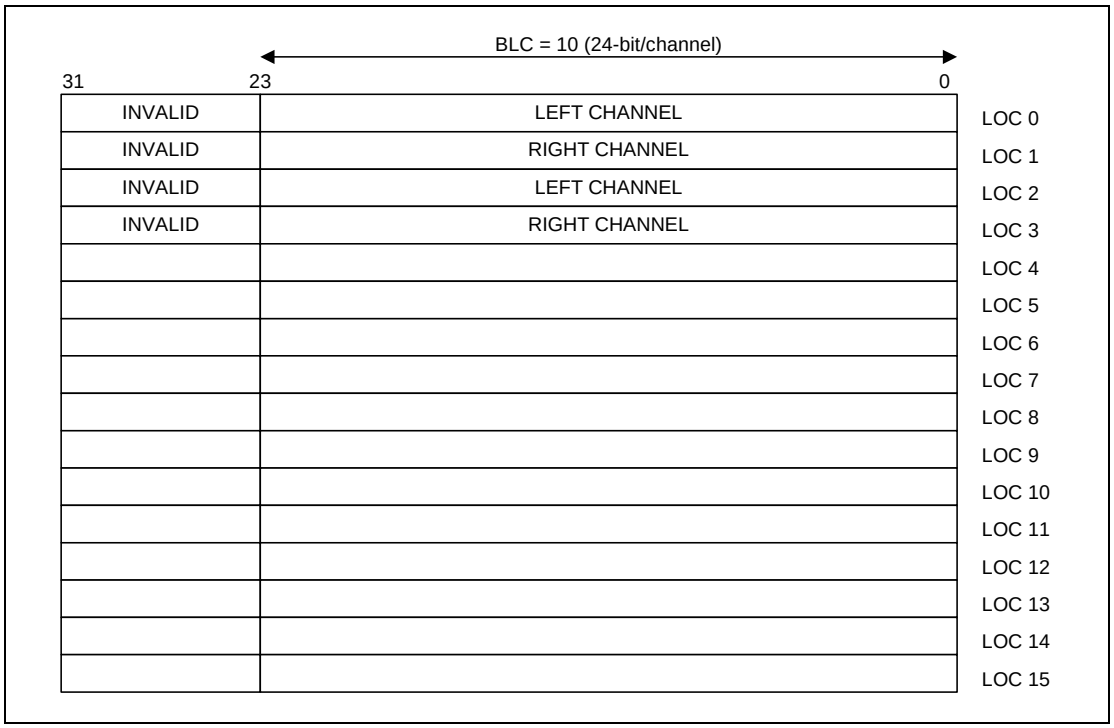


Figure 41-7. RX FIF0 Structure for BLC = 10 (24-bits/channel)

41.10 IIS-BUS INTERFACE SPECIAL REGISTERS

Table 41-3. Register summary of IIS interface

Register	Address	R/W	Description	Reset Value
IISCON	0x7F00D000	R/W	IIS interface control register	0xE00
IISMOD	0x7F00D004	R/W	IIS interface mode register	0x0
IISFIC	0x7F00D008	R/W	IIS interface FIFO control register	0x0
IISPSR	0x7F00D00C	R/W	IIS interface clock divider control register	0x0
IISTXD	0x7F00D010	W	IIS interface transmit data register	0x0
IISRXD	0x7F00D014	R	IIS interface receive data register	0x0

41.10.1 IISCON

Register	Address	Description	Reset Value
IISCON	0x7F00D000	IIS interface control register	0x0000_0E00

IISCON	Bit	R/W	Description
Reserved	[31:18]	R/W	Reserved. Program to zero.
FTXURSTATUS	[17]	R/W	TX FIFO under-run interrupt status. And this is used by interrupt clear bit. When this is high, you can do interrupt clear by writing ‘1’. 0: Interrupt didn’t be occurred. 1: Interrupt was occurred.
FTXURINTEN	[16]	R/W	TX FIFO Under-run Interrupt Enable 0: TXFIFO Under-run INT disable 1: TXFIFO Under-run INT enable
FTX2EMPT	[15]	R	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)
FTX1EMPT	[14]	R	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)
FTX2FULL	[13]	R	TX FIFO2 full Status Indication 0:TX FIFO2 is not full 1:TX FIFO2 is full
FTX1FULL	[12]	R	TX FIFO1 full Status Indication 0:TX FIFO1 is not full 1:TX FIFO1 is full
LRI	[11]	R	Left/Right channel clock indication. Note that LRI meaning is dependent on the value of LRP bit of I2SMOD register. 0: Left (when LRP bit is low) or right (when LRP bit is high) 1: Right (when LRP bit is low) or left (when LRP bit is high)

IISCON	Bit	R/W	Description
FTX0EMPT	[10]	R	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)
FRXEMPT	[9]	R	Rx FIFO empty status indication. 0: FIFO is not empty 1: FIFO is empty
FTX0FULL	[8]	R	Tx FIFO0 full status indication. 0: FIFO is not full 1: FIFO is full
FRXFULL	[7]	R	Rx FIFO full status indication. 0: FIFO is not full (ready for receive data from channel) 1: FIFO is full (not ready for receive data from channel)
TXDMAPAUSE	[6]	R/W	Tx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0: No pause DMA operation 1: Pause DMA operation
RXDMAPAUSE	[5]	R/W	Rx DMA operation pause command. Note that when this bit is activated at any time, the DMA request will be halted after current on-going DMA transfer is completed. 0: No pause DMA operation 1: Pause DMA operation
TXCHPAUSE	[4]	R/W	Tx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0: No pause operation 1: Pause operation
RXCHPAUSE	[3]	R/W	Rx channel operation pause command. Note that when this bit is activated at any time, the channel operation will be halted after left-right channel data transfer is completed. 0: No pause operation 1: Pause operation
TXDMACTIVE	[2]	R/W	Tx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0: Inactive, 1: Active
RXDMACTIVE	[1]	R/W	Rx DMA active (start DMA request). Note that when this bit is set from high to low, the DMA operation will be forced to stop immediately. 0: Inactive, 1: Active
I2SACTIVE	[0]	R/W	IIS interface active (start operation). 0: Inactive, 1:Active

41.10.2 IISMOD

Register	Address	Description	Reset Value
IISMOD	0x7F00D004	IIS interface mode register	0x0000_0000

IISMOD	Bit	R/W	Description
Reserved	[31:22]	R/W	Reserved. Program to zero.
CDD2	[21:20]	R/W	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
CDD1	[19:18]	R/W	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
DCE	[17:16]	R/W	Data Channel Enable. [17] : SD2 channel enable [16] : SD1 channel enable
Reserved	[15]	R/W	Reserved, Program to Zero
BLC	[14:13]	R/W	Bit Length Control Bit Which decides transmission of 8/16 bits per audio channel 00:16 Bits per channel 01:8 Bits Per Channel 10:24 Bits Per Channel 11:Reserved
CDCLK CON	[12]	R/W	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 41-2)
IMS	[11:10]	R/W	IIS master (internal/external) or slave mode select. 00: Master mode (, using PCLK) 01: Master mode (, using CLKAUDIO[x]) 10: Slave mode (divide mode, using PCLK) 11: Slave mode (bypass mode, using I2SCLK) (For more information refer to Figure 41-2)
TXR	[9:8]	R/W	Transmit or receive mode select. 00: Transmit only mode 01: Receive only mode 10: Transmit and receive simultaneous mode 11: Reserved

IISMOD	Bit	R/W	Description
LRP	[7]	R/W	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
SDF	[6:5]	R/W	Serial data format. 00: IIS format 01: MSB-justified (left-justified) format 10: LSB-justified (right-justified) format 11: Reserved
RFS	[4:3]	R/W	IIS root clock (codec clock) frequency select. 00: 256 fs, where fs is sampling frequency 01: 512 fs 10: 384 fs 11: 768 fs
BFS	[2:1]	R/W	Bit clock frequency select. 00: 32 fs, where fs is sampling frequency 01: 48 fs 10:16 fs 11: 24 fs
Reserved	[0]	R/W	Reserved. Program to zero.

41.10.3 IISFIC

Register	Address	Description	Reset Value
IISFIC	0x7F00D008	IIS interface FIFO control register	0x0000_0000

IISFIC	Bit	R/W	Description
Reserved	[31:29]	R/W	Reserved. Program to zero.
FTX2CNT	[28:24]	R	TX FIFO2 data count. FIFO has 16 depth, so value ranges from 0 to 15. N: Data count N of FIFO
Reserved	[23:21]	R/W	Reserved. Program to zero.
FTX1CNT	[20:16]	R	TX FIFO1 data count. FIFO has 16 depth, so value ranges from 0 to 15. N: Data count N of FIFO
TFLUSH	[15]	R/W	TX FIFO flush command. 0: No flush, 1: Flush
Reserved	[14:13]	R/W	Reserved. Program to zero.
FTX0CNT	[12:8]	R	TX FIFO0 data count. FIFO has 16 depth, so value ranges from 0 to 16. N: Data count N of FIFO
RFLUSH	[7]	R/W	RX FIFO flush command. 0: No flush, 1: Flush
Reserved	[6:5]	R/W	Reserved. Program to zero.
FRXCNT	[4:0]	R	RX FIFO data count. FIFO has 16 depth, so value ranges from 0 to 16. N: Data count N of FIFO

41.10.4 IISPSR

Register	Address	Description	Reset Value
IISPSR	0x7F00D00C	IIS interface clock divider control register	0x0000_0000

IISPSR	Bit	R/W	Description
Reserved	[31:16]	R/W	Reserved. Program to zero.
PSRAEN	[15]	R/W	Pre-scaler (Clock divider) active. 0: Inactive, 1: Active
Reserved	[14]	R/W	Reserved. SBZ
PSVALA	[13:8]	R/W	Pre-scaler (Clock divider) division value. N: Division factor is N+1
Reserved	[7:0]	R/W	Reserved. Program to zero.

41.10.5 IISTXD

Register	Address	Description	Reset Value
IISTXD	0x7F00D010	IIS interface transmit data register	0x0000_0000

IISTXD	Bit	R/W	Description
IISTXD	[31:0]	W	TX FIFO write data. Note that the left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

41.10.6 IISRXD

Register	Address	Description	Reset Value
IISRXD	0x7F002014 0x7F003014	IIS interface receive data register	0x0000_0000

IISRXD	Bit	R/W	Description
IISRXD	[31:0]	R	RX FIFO read data. Note that the left/right channel data is allocated as the following bit fields. R[31:16], L[15:0] when 16-bit BLC R[23:16], L[7:0] when 8-bit BLC

NOTES

42 GRAPHIC 3D

42.1 OVERVIEW

Graphic 3D (hereinafter 3D Engine) is a 3D Graphics Hardware Accelerator which can accelerate OpenGL ES 1.1 & 2.0 rendering. This 3D Engine is mainly targeting for mobile handsets and its key features are as follows. This 3D Engine includes two programmable shaders: one vertex shader and one pixel shader. Also, maximum 8 attributes (color or texture) can be supported in single rendering pass. In addition, high quality images can be obtained since this 3D engine is designed using 32-bit Floating-Point pipeline. And, hierarchical texture caching and texture compression technique are used for low memory bandwidth requirement. This 3D Engine uses one AHB channel for Host Interface and two AXI channels for frame buffer accesses follows.

42.1.1 FEATURES

- 4M triangles/s @133MHz (Transform Only)
- 75.8M 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 surfaces (max. 8 user-defined textures)
- API Support: OpenGL ES 1.1 & 2.0, D3D Mobile
- Intelligent Host Interface
  - 15 input data-types, Vertex Buffer & Vertex Cache
- H/W Clipping (Near & Far)
- 8-stage five-threaded Shader architecture
- Primitive assembly & hard-wired triangle setup engine
- One pixels/cycle hard-wired rasterizer

- One texturing engine (one bilinear-filtered texel/cycle each)
  - Nearest/bilinear/trilinear filtering
  - 8-layered multi-texturing support
- Fragment processing: Alpha/Stencil/Z/Dither/Mask/ROP
- Memory bandwidth optimization through hierarchical caching
  - L1/L2 Texture-caches, Z/Color caches
- System bus interface
  - Host interface: 32-bit AHB (AMBA 2.0)
  - Memory Interface: two 64-bit AXI (AMBA 3.0) channels

42.1.2 BLOCK DIAGRAM

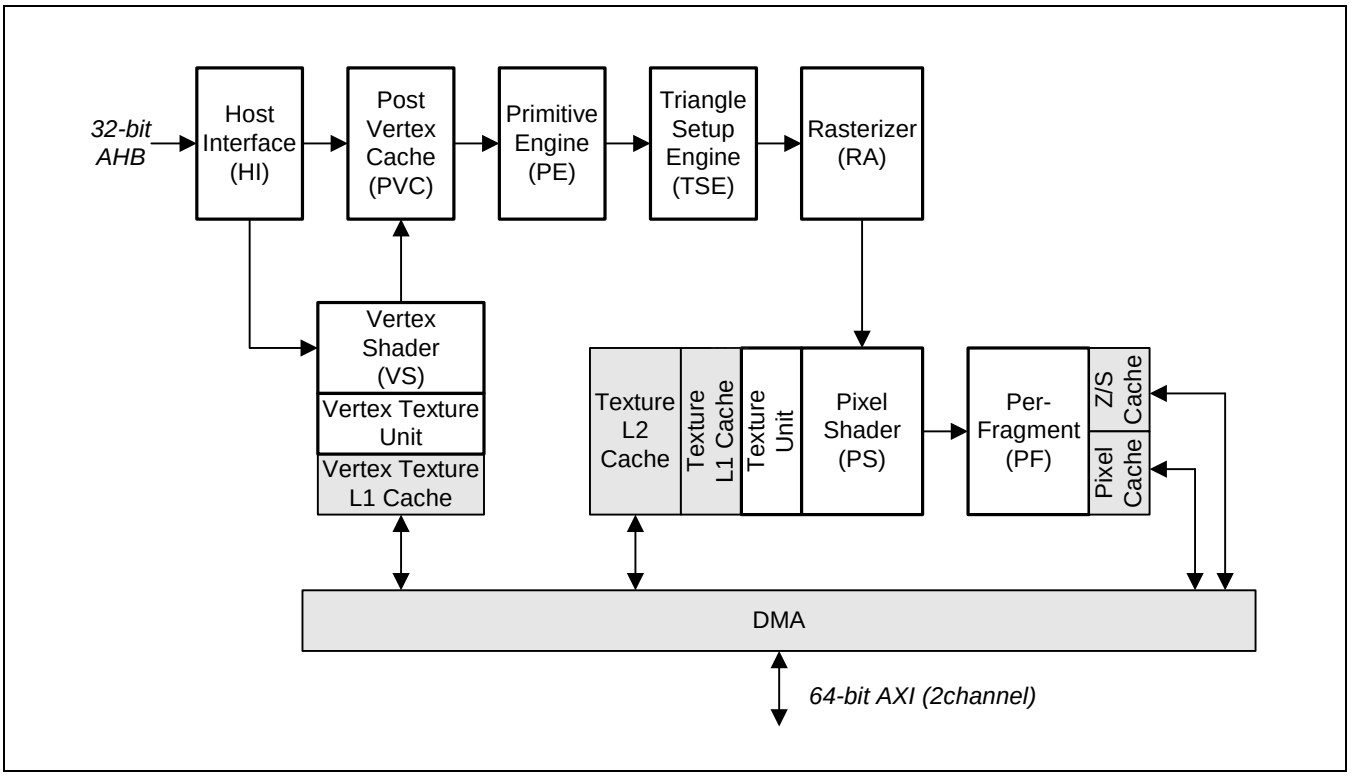


Figure 42-1. Block Diagram

42.2 SPECIAL FUNCTION REGISTERS

42.2.1 GLOBAL REGISTERS

42.2.1.1 Overview

Global registers is a set of overall states in GRAPHIC 3D.

THE ROLE OF FGGB_PIPESTATE SFRS

Each bit field in FGGB_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. FGGB_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 FGGB_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 GRAPHIC 3D 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 FGGB_PIPESTATE exists.

DATA TRANSFER USING GRAPHIC 3D'S INTERRUPT

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 GRAPHIC 3D’s pipeline-state can be used to know when to change SFR values for a GRAPHIC 3D 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 GRAPHIC 3D 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 GRAPHIC 3D’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 GRAPHIC 3D’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 GRAPHIC 3D.

The following shows an example illustrating how to transfer geometry data.

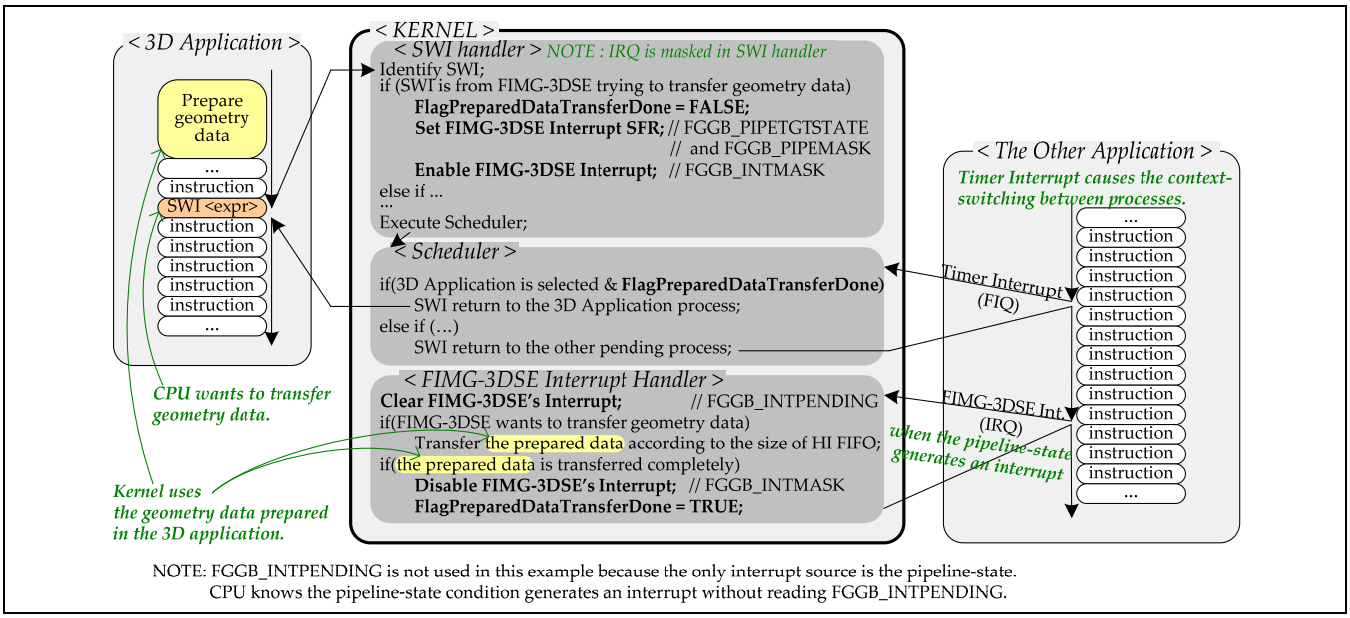


Figure 42-2. USING GRAPHIC 3D'S INTERRUPT with Geometry data

Note that when GRAPHIC 3D is empty, the pipeline-state generates an interrupt right after GRAPHIC 3D interrupt is enabled. However, the GRAPHIC 3D interrupt handler is executed when CPU executes the other user-mode application program (IRQ interrupt in enabled); in the above figure.

The following illustrates how to change SFR values. SFR can be changed when all the pipe-state becomes empty.

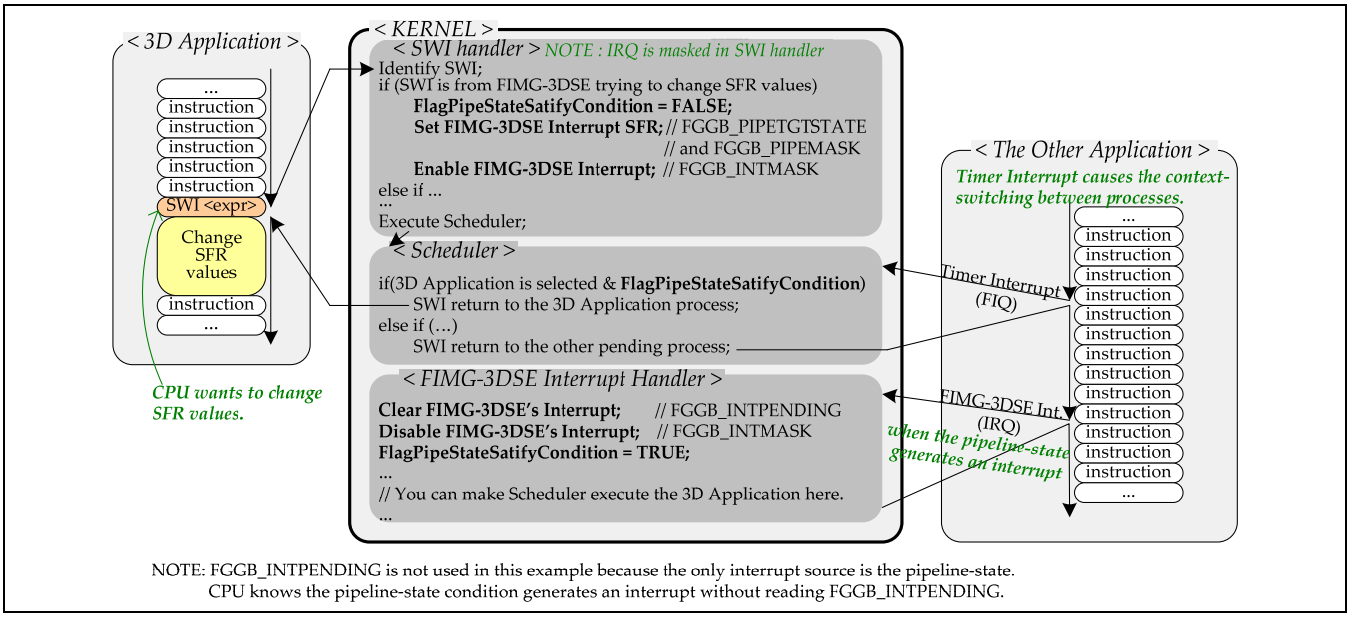


Figure 42-3. USING GRAPHIC 3D'S INTERRUPT

Note that the above scheme can be used also to transfer geometry data.

42.3 GLOBAL SPECIAL REGISTERS

42.3.1 PIPELINE STATUS REGISTER (FGGB_PIPESTATE)

Register	Address	R/W	Description	Reset Value
FGGB_PIPESTATE	0x72000000	R	The status of pipeline	0x00000000

FGGB_PIPESTATE	Bit	Description	Initial State
reserved	[31:19]	reserved	0
CCache0	[18]	0b = color cache0 is empty. 1b = color cache0 is not empty (busy).	0b
reserved	[17]	reserved	0
PF0	[16]	0b = per-fragment unit 0 is empty. 1b = per-fragment unit 0 is not empty (busy).	0b
reserved	[15:13]	reserved	0
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

42.3.2 CACHE CONTROL REGISTER (FGGB_CACHECTL)

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 FGGB_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.

Register	Address	R/W	Description	Reset Value
FGGB_CACHECTL	0x72000004	R/W	Cache control register	0x00000000

FGGB_CACHECTL	Bit	Description	Initial State
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; texture cache unchanged. 11: texture cache start invalidation	00b
reserved	[7:6]	reserved	0
CCFLUSH	[5:4]	Color cache flush (Automatically set to 00b after flushing) 00: color cache flush end 11: color cache flush start	00b
reserved	[3:2]	reserved	0
ZCFLUSH	[1:0]	Z cache flush (Automatically set to 00b after flushing) 00b = Z cache0 flush end 11b = Z cache0 flush start	00b

42.3.3 SOFTWARE RESET REGISTER (FGGB_RST)

You can reset the core of GRAPHIC 3D with FGGB_RST register. However, the SFR values are not affected by FGGB_RST. The reset bit of FGGB_RST is not recovered to 0 automatically. You must set FGGB_RST to 0 for the GRAPHIC 3D's operation.

Register	Address	R/W	Description	Reset Value
FGGB_RST	0x72000008	W	The SW reset control	0xbad1ff00

FGGB_RST	Bit	Description	Initial State
reserved	[31:8]	reserved	0
reset	[0]	Reset signal for GRAPHIC 3D core (logic and internal memory) 1 = Reset, 0 = Work	0b

42.3.4 VERSION INFORMATION REGISTER (FGGB_VERSION)

By reading FGGB_INFO register, you can identify which GRAPHIC 3D is implemented in a system.

Register	Address	R/W	Description	Reset Value
FGGB_VERSION	0x72000010	R	Version Information	0x01050000

FGGB_VERSION	Bit	Description	Initial State
major	[31:24]	Major version.	0x01
minor	[23:0]	Minor version. Ex) Version 1.2.1 : FGGB_INFO = 0x01020100	0x050000

42.3.5 INTERRUPT PENDING REGISTER (FGGB_INTPENDING)

When CPU receives an interrupt from GRAPHIC 3D, CPU must investigate which functional block in GRAPHIC 3D generates an interrupt. CPU can figure out the interrupt-generating block by reading FGGB_INTPENDING.

Any value must be written into FGGB_INTPENDING in the interrupt service routine to clear interrupts from GRAPHIC 3D. By writing any value into FGGB_INTPENDING, FGGB_INTPENDING is automatically cleared and GRAPHIC 3D can generate another interrupt. The written value into FGGB_INTPENDING is not important; the write operation into FGGB_INTPENDING clears its value.

Currently, FGGB_PIPESTATE(Pipeline-State) in HI can only generate an interrupt. Once GRAPHIC 3D generates an interrupt, CPU knows that FGGB_PIPESTATE is the interrupt source without reading FGGB_INTPENDING.

Register	Address	R/W	Description	Reset Value
FGGB_INTPENDING	0x72000040	R/W	Interrupt Pending Register	0x00000000

FGGB_INTPENDING	Bit	Description	Initial State
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

42.3.6 INTERRUPT MASK REGISTER (FGGB_INTMASK)

FGGB_INTMASK can enable or disable interrupts from GRAPHIC 3D. Currently, interrupts can be generated only by HI (Pipeline-State). Hence, LSB of FGGB_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 FGGB_PIPEMASK. FGGB_INTMASK is the global control while FGGB_PIPEMASK is the bit-wise control.

Register	Address	R/W	Description	Reset Value
FGGB_INTMASK	0x72000044	R/W	Enables of Disables interrupts.	0x00000000

FGGB_INTMASK	Bit	Description	Initial State
reserved	[31:1]	reserved	0
Pipeline State	[0]	“Pipeline State” generates an interrupt. 1 = Enable Interrupt, 0 = Disable Interrupt.	0b

42.3.7 PIPELINE MASK REGISTER (FGGB_PIPEMASK)

FGGB_PIPEMASK specifies the interesting GRAPHIC 3D block for interrupt generation. The GRAPHIC 3D blocks having value one in FGGB_PIPEMASK are candidates for interrupts; the blocks having zero value are ignored during interrupt generation.

Register	Address	R/W	Description	Reset Value
FGGB_PIPEMASK	0x72000048	R/W	Specifies the blocks in GRAPHIC 3D which are candidates to generate interrupts. The bit position of each block is as same as that of FGGB_PIPESTATE.	0x00000000

FGGB_PIPEMASK	Bit	Description	Initial State
reserved	[31:17]	reserved	0
PF0	[16]	0b = don't care 1b = used to generate interrupts	0b
reserved	[15:13]	reserved	0
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

42.3.8 PIPELINE TARGET STATE REGISTER (FGGB_PIPETGTSTATE)

As mentioned before, FGGB_PIPEMASK specifies the interesting GRAPHIC 3D block for interrupt generation. FGGB_PIPETGTSTATE specifies the value of pipeline-state when interrupts occur. Note that the FGGB_PIPETGTSTATE value for a block with 0 value in FGGB_PIPEMASK is ignored.

Register	Address	R/W	Description	Reset Value
FGGB_PIPETGTSTATE	0x7200004C	R/W	Specifies the value of pipeline-state when interrupts are to occur.	0x00000000

FGGB_PIPETGTSTATE	Bit	Description	Initial State
reserved	[31:16]	reserved	0
PF0	[16]	0b = interrupts when the PF0 is not working. (empty) 1b = interrupts when the PF0 is working. (not-empty)	0b
reserved	[15:13]	Reserved	0
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

42.3.9 PIPELINE INTERRUPT STATE REGISTER (FGGB_PIPEINTSTATE)

FGGB_PIPEINTSTATE captures the pipeline-state when interrupts occur. When several interrupts occur, the FGGB_PIPEINTSTAE holds the first pipeline-state.

Note that FGGB_PIPEINTSTATE depends on FGGB_PIPEMASKE.

Register	Address	R/W	Description	Reset Value
FGGB_PIPEINTSTATE	0x72000050	R	Captures the first pipeline-state when several interrupts occur.	0x00000000

FGGB_PIPEINTSTATE	Bit	Description	Initial State
reserved	[31:17]	reserved	0
PF0	[16]	0b = the PF0 was empty when an interrupt occurred. 1b = the PF0 was not empty when an interrupt occurred.	0b
reserved	[15:13]	reserved	0
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

42.4 HOST INTERFACE

42.4.1 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 GRAPHIC 3D 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 GRAPHIC 3D. GRAPHIC 3D 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 GRAPHIC 3D. At this moment, the previous geometry data can be affected by the new state. Hence, state data can be transferred to the GRAPHIC 3D only when the state data does not affect the previous geometry. PIPELINE STATUS REGISTER (FGGB_PIPESTATE), which is explained the previous chapter, is used to query where the geometry data is processed in GRAPHIC 3D. (CPU decides when to update the states using FGGB_PIPESTATE.) Suppose there are blocks, A, B, C, and D in GRAPHIC 3D 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.

42.4.2 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.

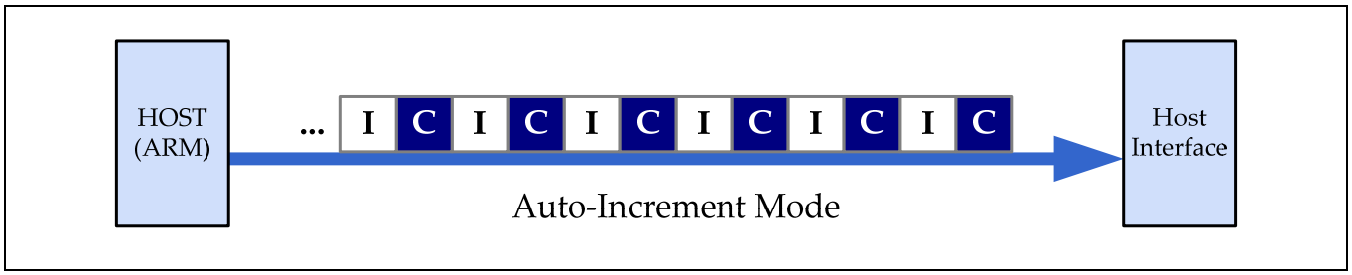
Index mode: After CPU stores the geometry data (input attributes to vertex shader) into the Vertex Buffer in Host Interface, CPU transfers the indices to the stored geometry. This scheme consumes low bus bandwidth.

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.

42.4.2.1 INDEX MODE

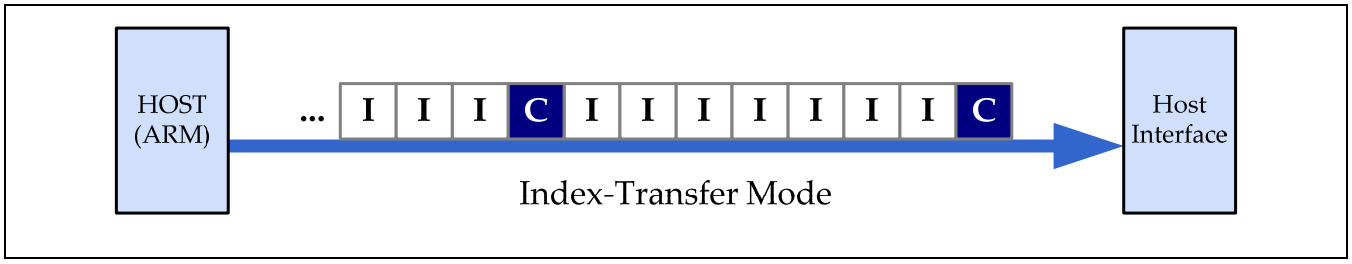
Index mode **uses the Vertex Buffer** in Host Interface. There are two operation modes: auto-increment mode and index-transfer mode. NOTE: All of the geometry data must be in the Vertex Buffer. Also, the range of used indices is deeply related to the VB size: the VB address calculated with indices must be the available VB address. Hence the indices must be carefully controlled by the application program (or the device driver).

- **Auto-increment mode (FGHI_CONTROL.EnVB=1, FGHI_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; FGHI_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.



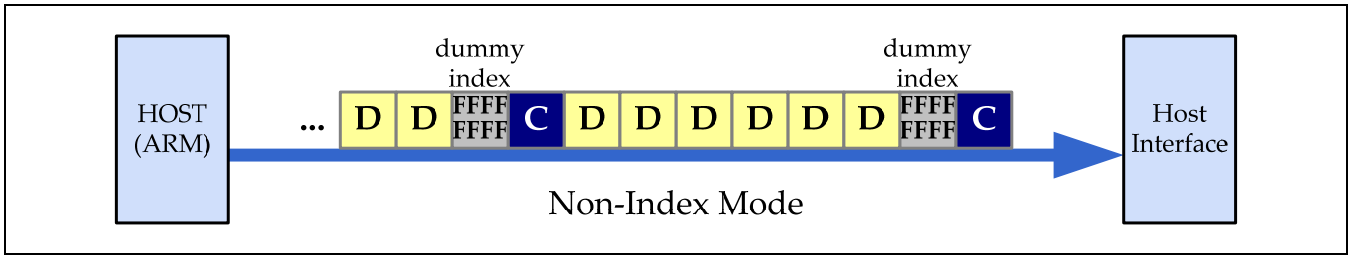
In the above figure, 6 sets of the geometries are transferred from CPU.

- **Index-transfer mode (FGHI_CONTROL.EnVB=1, FGHI_CONTROL.AutoInc=0)**: In index-transfer mode, PU 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: FGHI_IDXOFFSET can be used to bias the sent index; (sent index + FGHI_IDXOFFSET) is used as a new index}.



NON-INDEX MODE (FGHI_CONTROL.EnVB=0, FGHI_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.



The transferred data are decoded using FGHI_ATTRIB0~FGHI_ATTRIB9.

Note that when float or half-float format data is transferred to the Host-Interface, it must not be NaN or Infinite.

42.4.3 HOW TO SET SPECIAL REGISTER FOR THE HOST INTERFACE

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 FGHI_ATTRIB0~FGHI_ATTRIB9. The number of components in each attributes is determined by FGHI_ATTRIB*n*.NumComp. The LastAttr bit in FGHI_ATTRIB*n* represents whether the FGHI_ATTRIB*n* is used or not. For example, FGHI_ATTRIB0.LastAttr = FGHI_ATTRIB1.LastAttr = FGHI_ATTRIB2.LastAttr = 0 and FGHI_ATTRIB3.LastAttr = 1 mean that a vertex is composed of 4 attributes. (FGHI_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 FGHI_DWENTRY of the Host Interface.) Then, CPU can transfer indices or raw geometry data into the same FGHI_DWENTRY. However, the Vertex Buffer can feed only geometry data to the Host Interface. The usage of the Vertex Buffer is determined by FGHI_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 FGHI_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 FGHI_CONTROL.AutoInc. In Non-Index mode, 0xFFFFFFFF is used.
3. Add FGHI_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(FGHI_CONTROL.EnVB == 1 && FGHI_ATTRIB[n]_VBCTRL.Range != 0 && *index* < FGHI_ATTRIB[n]_VBCTRL.Range)
6. Use DWORDs fetched from VertexBuffer[**FGHI_ATTRIB**[*n*]_**VBBASE**.**Addr** + *index****FGHI_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 **FGHI_ATTRIB**[*n*].**Dt** and send them to vertex shader.
10. if (**FGHI_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 FGHI_ATTRIB*n*.Dt (data type of transferred DWORD) and FGHI_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, FGHI_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 FGHI_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.

42.4.4 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 GRAPHIC 3D 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 FGHI_DWSPACE register is used to ease this situation. FGHI_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 FGHI_DWSPACE value and transfer DWORDs as much as the value of FGHI_DWSPACE*.**

Generally speaking, FGHI_DWSPACE does not tell CPU the exact empty space because the clock signals fed into GRAPHIC 3D and AMBA bus can be different. (If clock signals are same, FGHI_DWSPACE has the exact value.) Hence, FGHI_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 FGHI_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 FGHI_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 FGHI_DWSPACE value and the actual value is usually small.

After CPU reading FGHI_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 GRAPHIC 3D. 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.)

42.4.5 THE TYPE OF INDEX TRANSFERRED FROM CPU

The IdxType field in FGHI_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.

42.4.6 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 FGHI_VBADDR. And then, a serise of DWORD written into FGHI_VBDATA is stored into the Vertex Buffer (Burst writes are possible). The address in FGHI_VBADDR is automatically incremented by 16 (in bytes) whenever 4 DWORDs are written into FGHI_VBDATA. Therefore, the destination address does not need to be updated for every DWORD written into FGHI_VBDATA. Note that the number of DWORD written into FGHI_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 FGHI_ATTRIB*n*_VBCT-RL.Range. The value of actual index range must be written in FGHI_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.”

42.4.7 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 FGHI_DWSPACE polling, CPU has to waste lots of cycles reading FGHI_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 GRAPHIC 3D’s interrupt scheme making the interrupt-unit send an interrupt to CPU when the values of FGGB_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 GRAPHIC 3D. If an interrupt from GRAPHIC 3D 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 GRAPHIC 3D 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 FGHI_PIPESTATE to be zero, an interrupt from GRAPHIC 3D never occurs under this situation because the Primitive-Engine’s value of FGHI_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.

42.4.8 VERTEX CACHE CONTROL

The EnVC and NumOutAttrib fields in FGHI_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 FGHI_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 FGHI_CONTROL is written by CPU. Although the FGHI_CONTROL value is not changed, FGHI_CONTROL can be rewritten with the same value to clear the contents of Vertex Cache.

42.5 HOST INTERFACE SPECIAL REGISTERS

Note: Host Interface is abbribiated to HI.

42.5.1 HOST-FIFO'S FREE DWORD SPACE REGISTER (FGHI_DWSPACE)

Register	Address	R/W	Description	Reset Value
FGHI_DWSPACE	0x72008000	R	The number of empty slots of the Host-FIFO in HI.	0x00000000

FES	Bit	Description	Initial State
VAL	[31:0]	A number of empty slots	0x0

42.5.2 HOST-FIFO ENTRY PORT REGISTER (FGHI_DWENTRY)

Register	Address	R/W	Description	Reset Value
FGHI_DWENTRY	0x7200C000 ~ 0x7200DFFF	W	The input port of the Host-FIFO in HI. Burst writes are possible. DWORDs written in 0x0000C000 ~ 0x0000DFFF are stored into Host-FIFO of Host Interface	-

FGHI_DWENTRY	Bit	Description	Initial State
DATA	[31:0]	The number of vertices, indices, and the geometry data is transferred into this register.	-

42.5.3 HOST INTERFACE CONTROL REGISTER (FGHI_CONTROL)

Register	Address	R/W	Description	Reset Value
FGHI_CONTROL	0x72008008	R/W	Host Interface control register. Note that VC is initialized automatically when FGHI_CONGROL is written.	0x00010000

FGHI_CONTROL	Bit	Description	Initial State
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

42.5.4 INDEX OFFSET REGISTER (FGHI_IDXOFFSET)

Register	Address	R/W	Description	Reset Value
FGHI_IDXOFFSET	0x7200800C	R/W	Index offset register (signed value)	0x00000001

FGHI_IDXOFFSET	Bit	Description	Initial State
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 FGHI_ATTRIB*n*_VBCTRL.Range, the geometry data in Vertex Buffer is used.	0x00000001

42.5.5 VERTEX BUFFER ADDRESS REGISTER (FGHI_VBADDR)

Register	Address	R/W	Description	Reset Value
FGHI_VBADDR	0x72008010	R/W	*Write*: Set the destination address register into which attributes are transferred in VB. (Address must be 16-byte-aligned) *Read*: The address which will be used for the next transferred geometry data into VB is read. This value can be used to calculate how many data is transferred. FGHI_VBADDR is automatically updated whenever four DWORDs are written into Vertex Buffer.	0x00000000

FGHI_VBADDR	Bit	Description	Initial State
VAL	[31:0]	Start address of attribute to copy the geometry data.	0x0

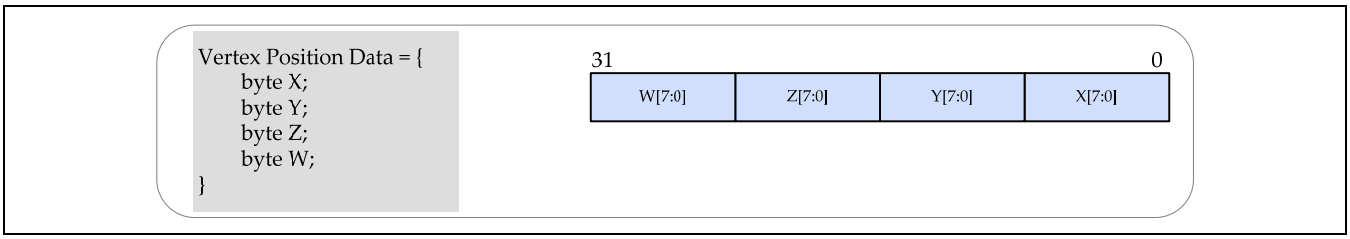
42.5.6 VERTEX BUFFER ENTRY PORT REGISTER (FGHI_VBDATA)

Register	Address	R/W	Description	Reset Value
FGHI_VBDATA	0x7200E000 ~ 0x7200FFFF	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 FGHI_VBDATA is read.	0x00000000

VBD	Bit	Description	Initial State
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

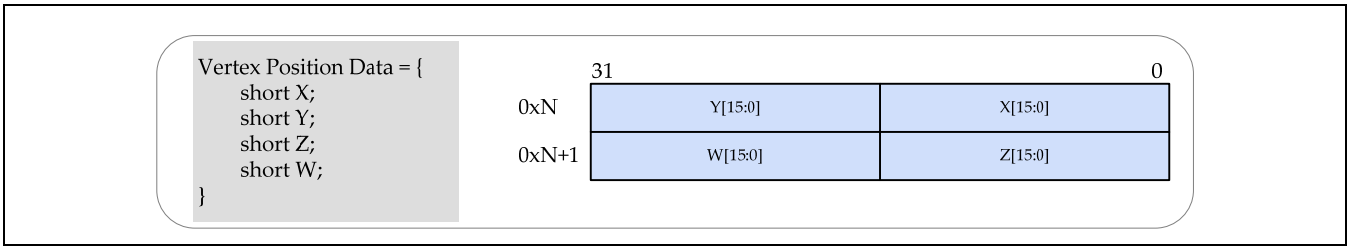
42.5.7 ATTRIBUTE CONTROL REGISTER (FGHI_ATTRIB0~FGHI_ATTRIB9)

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.



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, 16-bit w).



Note that when DWORDs are stored in the Vertex Buffer, the above rule is also applied.

Register	Address	R/W	Description	Reset Value
FGHI_ATTRIB0	0x72008040	R/W	Input attribute 0 control register	0x800000E4
FGHI_ATTRIB1	0x72008044	R/W	Input attribute 1 control register	0x800000E4
FGHI_ATTRIB2	0x72008048	R/W	Input attribute 2 control register	0x800000E4
FGHI_ATTRIB3	0x7200804C	R/W	Input attribute 3 control register	0x800000E4
FGHI_ATTRIB4	0x72008050	R/W	Input attribute 4 control register	0x800000E4
FGHI_ATTRIB5	0x72008054	R/W	Input attribute 5 control register	0x800000E4
FGHI_ATTRIB6	0x72008058	R/W	Input attribute 6 control register	0x800000E4
FGHI_ATTRIB7	0x7200805C	R/W	Input attribute 7 control register	0x800000E4
FGHI_ATTRIB8	0x72008060	R/W	Input attribute 8 control register	0x800000E4
FGHI_ATTRIB9	0x72008064	R/W	Input attribute 9 control register	0x800000E4

FGHI_ATTRIBn	Bit	Description	Initial State
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) FGHI_ATTRIB0[31] = 0, FGHI_ATTRIB1[31] = 0, FGHI_ATTRIB2[31] = 1, FGHI_ATTRIB3~9[31] = don’t care → FGHI_ATTRIB0 ~ 2 are used.	1b
reserved	[30:16]	reserved	0

FGHI_ATTRIBn	Bit	Description	Initial State
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}) (a, b, c, d) is used to select (X, Y, Z, W) generally	00b
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 are defined in NumComp field.	11b

FGHI_ATTRIBn	Bit	Description	Initial State
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 are 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

42.5.8 VERTEX BUFFER CONTROL REGISTER (FGHI_ATTRIB0_VBCTRL ~ FGHI_ATTRIB9_VBCTRL)

FGHI_ATTRIBn_VBCTRL.Stride represents the number of bytes to the next input attribute in the Vertex Buffer.
FGHI_ATTRIBn_VBCTRL.num represnets how many the input attributes are in the Vertex Buffer.

Register	Address	R/W	Description	Reset Value
FGHI_ATTRIB0_VBCTRL	0x72008080	R/W	Vertex buffer control of input attribute 0	0x00000000
FGHI_ATTRIB1_VBCTRL	0x72008084	R/W	Vertex buffer control of input attribute 1	0x00000000
FGHI_ATTRIB2_VBCTRL	0x72008088	R/W	Vertex buffer control of input attribute 2	0x00000000
FGHI_ATTRIB3_VBCTRL	0x7200808C	R/W	Vertex buffer control of input attribute 3	0x00000000
FGHI_ATTRIB4_VBCTRL	0x72008090	R/W	Vertex buffer control of input attribute 4	0x00000000
FGHI_ATTRIB5_VBCTRL	0x72008094	R/W	Vertex buffer control of input attribute 5	0x00000000
FGHI_ATTRIB6_VBCTRL	0x72008098	R/W	Vertex buffer control of input attribute 6	0x00000000
FGHI_ATTRIB7_VBCTRL	0x7200809C	R/W	Vertex buffer control of input attribute 7	0x00000000
FGHI_ATTRIB8_VBCTRL	0x720080A0	R/W	Vertex buffer control of input attribute 8	0x00000000
FGHI_ATTRIB9_VBCTRL	0x720080A4	R/W	Vertex buffer control of input attribute 9	0x00000000

FGHI_ATTRIBn_VBCTRL	Bit	Description	Initial State
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. This value is used to decide whether the geometry data for an index is in the Vertex Buffer or not.	0x0

42.5.9 VERTEX BUFFER BASE ADDRESS REGISTER (FGHI_ATTR0_VBBASE~FGHI_ATTR9_VBBASE)

Register	Address	R/W	Description	Reset Value
FGHI_ATTRIB0_VBBASE	0x720080C0	R/W	Vertex buffer base address of input attribute 0	0x00000000
FGHI_ATTRIB1_VBBASE	0x720080C4	R/W	Vertex buffer base address of input attribute 1	0x00000000
FGHI_ATTRIB2_VBBASE	0x720080C8	R/W	Vertex buffer base address of input attribute 2	0x00000000
FGHI_ATTRIB3_VBBASE	0x720080CC	R/W	Vertex buffer base address of input attribute 3	0x00000000
FGHI_ATTRIB4_VBBASE	0x720080D0	R/W	Vertex buffer base address of input attribute 4	0x00000000
FGHI_ATTRIB5_VBBASE	0x720080D4	R/W	Vertex buffer base address of input attribute 5	0x00000000
FGHI_ATTRIB6_VBBASE	0x720080D8	R/W	Vertex buffer base address of input attribute 6	0x00000000
FGHI_ATTRIB7_VBBASE	0x720080DC	R/W	Vertex buffer base address of input attribute 7	0x00000000
FGHI_ATTRIB8_VBBASE	0x720080E0	R/W	Vertex buffer base address of input attribute 8	0x00000000
FGHI_ATTRIB9_VBBASE	0x720080E4	R/W	Vertex buffer base address of input attribute 9	0x00000000

FGHI_ATTRIBn_VBBASE	Bit	Description	Initial State
reserved	[31:16]	reserved	0
Addr	[15:0]	Base address of input attribute n in Vertex Buffer	0x0

42.6 VERTEX SHADER

42.6.1 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.

42.6.2 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.

42.6.3 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.

42.6.3.1 INSTRUCTION MEMORY

The instruction memory has 512 slots and 1 slot is composed of 4 words.

Register	Address	R/W	Description	Reset Value
FGVS_INSTMEM	0x72010000 ~ 0x72011FFF	R/W	Instruction memory of the vertex shader	0xX

See the Programmable Shader S/W Developer's Guide for more detail.

42.6.3.2 CONSTANT FLOAT REGISTER

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.)

Register	Address	R/W	Description	Reset Value
FGVS_CFLOAT	0x72014000 ~ 0x72014FFF	R/W	Constant float register of the vertex shader	

Word 3 (0x72014XXC)

WORD3	Bit	Description	Initial State
W	[127:96]	Constant float W component value	0xXXXXXXXX

Word 2 (0x72014XX8)

WORD2	Bit	Description	Initial State
Z	[95:64]	Constant float Z component value	0xXXXXXXXX

Word 1 (0x72014XX4)

WORD1	Bit	Description	Initial State
Y	[63:32]	Constant float Y component value	0xXXXXXXXX

Word 0 (0x72014XX0)

WORD0	Bit	Description	Initial State
X	[31:0]	Constant float X component value	0xXXXXXXXX

♣ IEEE single precision floating-point format

	Bit	Description	Initial State
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

42.6.3.3 CONSTANT INTEGER REGISTER

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.

Register	Address	R/W	Description	Reset Value
FGVS_CINT	0x72018000 ~ 0x7201803F	R/W	A constant integer register of vertex shader	0xXXXXXXXX

WORD (0x72018XX0)

WORD	Bit	Description	Initial State
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

42.6.3.4 CONSTANT BOOL REGISTER

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’.

Register	Address	R/W	Description	Reset Value
FGVS_CBOOL	0x72018400	R/W	A constant boolean register of vertex shader	0xXXXXXXXX

CBOOL	Bit	Description	Initial State
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

42.6.3.5 CONFIGURATION REGISTER FOR VERTEX SHADER

Global register contains various configurations and environment setting for global operation.

Register	Address	R/W	Description	Reset Value
FGVS_Config	0x7201C800	W	Configuration register of vertex shader	0x00000000
FGVS_Status	0x7201C804	R	Internal status register (Reserved)	0x00000000
FGVS_PCRange	0x72020000	R/W	Vertex shader program start and end address	0x01FF0000
FGVS_AttributeNum	0x72020004	R/W	The number of attributes of current context	0x00010001

FGVS_Config	Bit	Description	Initial State
reserved	[31:1]	reserved	0
ClrStatus	[1]	When this bit is set to '1', the all values of FGVS_Status register are cleared. This bit is automatically cleared to '0' after clear operation.	0b
CopyPC	[0]	When this bit is set to '1', FGVS_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 FGVS_PCRange is not used, and the previous values are used for program start and end address.	0b

FGVS_PCRange	Bit	Description	Initial State
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

FGVS_AttributeNum	Bit	Description	Initial State
reserved	[31:4]	reserved	0x0001000
InAttributeNum	[3:0]	The number of attributes for the vertex shader input. This should be match the number of input registers in current shader program.	0x1

42.6.3.6 INDEX REGISTER FOR INPUT ATTRIBUTE

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.

Register	Address	R/W	Description	Reset Value
FGVS_InAttrIndex0	0x72020008	W	Index of input attributes 0~3	0x03020100
FGVS_InAttrIndex1	0x7202000C	W	Index of input attributes 4~7	0x07060504
FGVS_InAttrIndex2	0x72020010	W	Index of input attributes 8~9	0x0B0A0908

FGVS_InAttrIndex0	Bit	Description	Initial State
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

FGVS_InAttrIndex1	Bit	Description	Initial State
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

FGVS_InAttrIndex2	Bit	Description	Initial State
reserved	[31:12]	reserved	0x0B0A0
Attrib9	[11:8]	Index of input attribute 9	0x9
reserved	[7:4]	reserved	0
Attrib8	[3:0]	Index of input attribute 8	0x8

42.6.3.7 INDEX REGISTER FOR OUTPUT ATTRIBUTE

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.

Register	Address	R/W	Description	Reset Value
FGVS_OutAttrIndex0	0x72020014	W	Index of output attributes 0~3	0x03020100
FGVS_OutAttrIndex1	0x72020018	W	Index of output attributes 4~7	0x07060504
FGVS_OutAttrIndex2	0x7202001C	W	Index of output attributes 8~9	0x0B0A0908

FGVS_OutAttrIndex0	Bit	Description	Initial State
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

FGVS_OutAttrIndex1	Bit	Description	Initial State
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

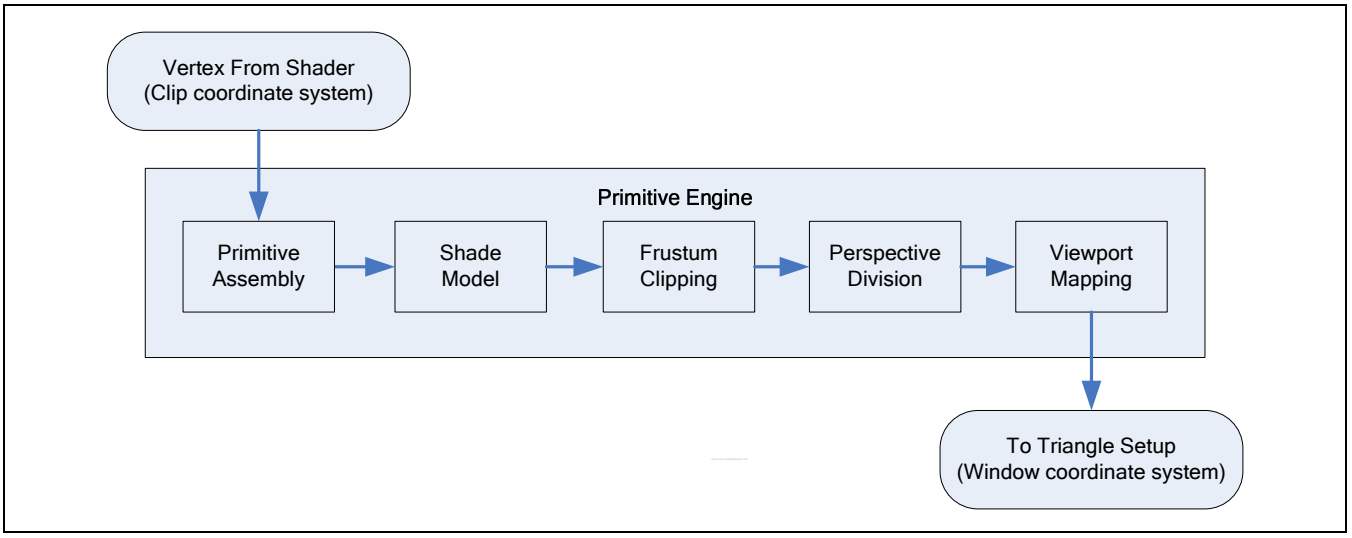
FGVS_OutAttrIndex2	Bit	Description	Initial State
reserved	[31:12]	reserved	0x0B0A0
Attrib9	[11:8]	Index of output attribute 9	0x9
reserved	[7:4]	reserved	0
Attrib8	[3:0]	Index of output attribute 8	0x8

42.7 PRIMITIVE ENGINES

42.7.1 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.



42.7.2 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.

42.7.2.1 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.

$$VSOUT = \begin{cases} \left\lfloor \frac{(\text{the number of varying variables})}{4} \right\rfloor + 1 & \text{, when vertex program point size mode} \\ \left\lfloor \frac{(\text{the number of varying variables})}{4} \right\rfloor & \text{, otherwise} \end{cases}$$

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 GRAPHIC 3D, the vertex shader supports the number of input attributes up to 8 and that of varying variables up to 32.

Register	Address	R/W	Description	Reset Value
FGPE_VERTEX_CONTEXT	0x72030000	R/W	Vertex context format definition	0x00000000

VCTX	Bit	Description	Initial State
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
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

VCTX	Bit	Description	Initial State
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

42.7.2.2 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

$$\begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} = \begin{pmatrix} (p_x/2)x_d + o_x \\ (p_y/2)y_d + o_y \\ [(f-n)/2]z_d + (n+f)/2 \end{pmatrix}$$

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 GRAPHIC 3D, 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.

Register	Address	R/W	Description	Reset Value
FGPE_VIEWPORT_OX	0x72030004	R/W	The x-coordinate of viewport center	0xX
FGPE_VIEWPORT_OY	0x72030008	R/W	The y-coordinate of viewport center	0xX
FGPE_VIEWPORT_HALF_PX	0x7203000C	R/W	Half of viewport width	0xX
FGPE_VIEWPORT_HALF_PY	0x72030010	R/W	Half of viewport height	0xX
FGPE_DEPTHRANGE_HALF_F_SUB_N	0x72030014	R/W	Half of depth range far minus near	0x3F000000
FGPE_DEPTHRANGE_HALF_F_ADD_N	0x72030018	R/W	Half of depth range far plus near	0x3F000000

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	Initial State
FGPE_VIEWPORT_OX	[31:0]	The x-coordinate of viewport center $x_0 + \frac{p_x}{2}$	0xX

	Bit	Description	Initial State
FGPE_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	Initial State
FGPE_VIEWPORT_HALF_PX	[31:0]	The half value of viewport width $\frac{p_x}{2}$	0xX

	Bit	Description	Initial State
FGPE_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	Initial State
FGPE_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	Initial State
FGPE_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

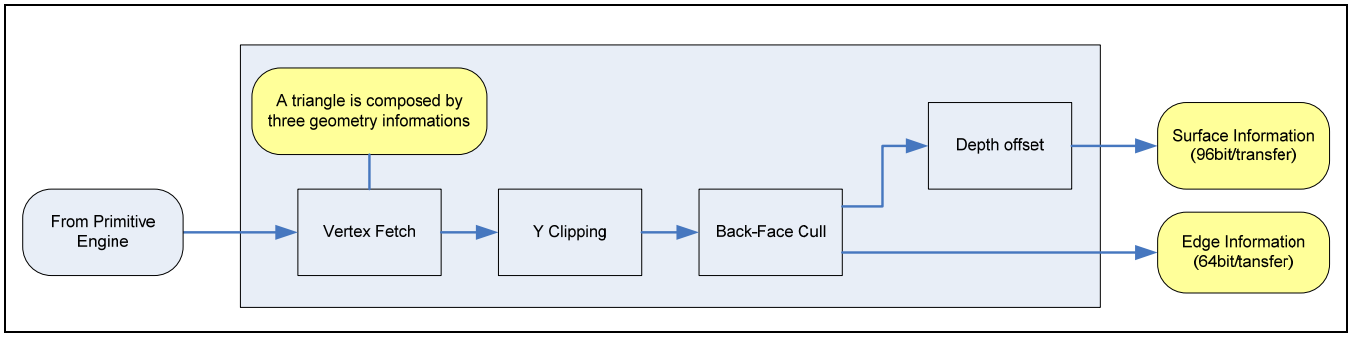
42.8 RASTER ENGINES

42.8.1 OVERVIEW

Raster engine include triangle setup engine and rasterizer.

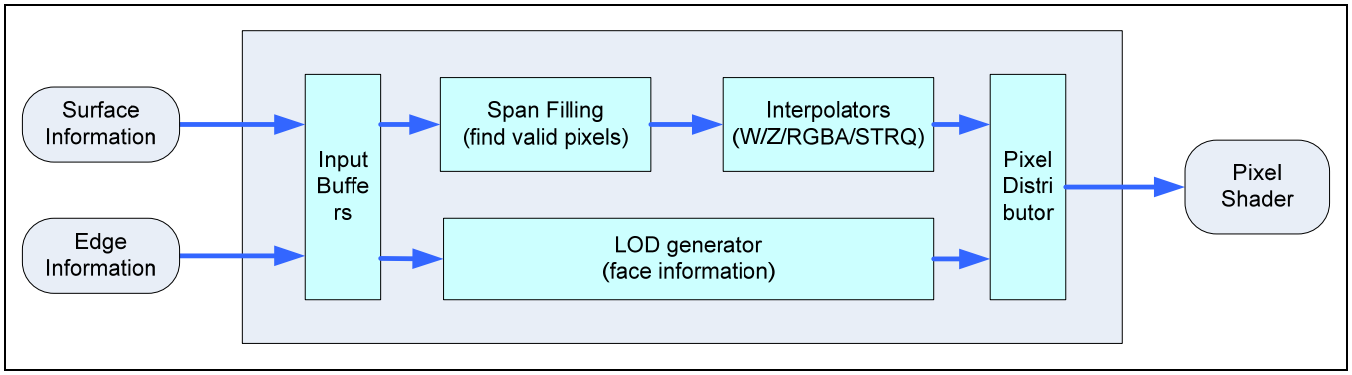
Triangle setup engine overview

- Primitive y-value bounding
- Back face culling
- Edge interpolation
- Triangle gradients calculation
- Depth offset calculation



Rasterizer overview

- Triangle, Line, Point, and Point Sprite are supported.(aliased only)
- 1-fragment is generated per cycle by a 1x1 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.



42.8.2 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 properly. 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 attribute generated in TSE is 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 attribute. ***And, the number of attributes should be set to the number increased by one for the attached texture coordinate attribute. (That is, NumOutAttrib in FGHI_CONTROL register and VSOUT in FGPE_VERTEX_CONTEXT register should be increased by one respectively.)***

Point Width control register: Point size value can be set by the SFR, PointWidth. Or, Point size value can be given from Vertex Shader. For this, the VPPS (VERTEXPROGRAMPOINTSIZE) SFR in Primitive Engine should be set. And then, the selected Point size is clamped by both PointSize_Min and PointSize_Max.

Line Width control register: It is used to assign Line Width value.

42.9 RASTER ENGINE SPECIAL REGISTERS

42.9.1 SAMPLING POSITION REGISTER

Register	Address	R/W	Description	Reset Value
FGRA_PixSamp	0x72038000	RW	Indicate pixel sampling position register	0x00000000

FGRA_PixSamp	Bit	Description	Initial State
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

42.9.2 DEPTH OFFSET REGISTERS

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* (0x33800002) 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.

Register	Address	R/W	Description	Reset Value
FGRA_DOffEn	0x72038004	RW	Depth offset enable register	0x00000000
FGRA_DOffFactor	0x72038008	RW	Depth offset factor register	0x00000000
FGRA_DOffUnits	0x7203800C	RW	Depth offset units register	0x00000000
FGRA_DOffRIn	0x72038010	RW	Depth offset implementation constant r value	0x34000001

FGRA_DOffEn	Bit	Description	Initial State
reserved	[31:1]	reserved	0
DOffEn	[0]	Depth offset usage control register 0b = Disable 1b = Enable	0b

FGRA_DOffFactor	Bit	Description	Initial State
DOffFactor	[31:0]	Factor scales the max depth slope of the polygon. Used to calculate the depth offset value.	0x0

FGRA_DOffUnits	Bit	Description	Initial State
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

FGRA_DOffRIn	Bit	Description	Initial State
DOffRIn	[31:0]	An **implementation dependent constant** that relates to the usable resolution of the depth buffer. (Do not change.) Used to calculate the depth offset value of a polygon.	0x33800002

42.9.3 BACK-FACE CULL CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGRA_BFCULL	0x72038014	RW	Back-face culling control register	0x00000000

FGRA_BFCULL	Bit	Description	Initial State
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

42.9.4 WINDOW Y CLIPPING REGION REGISTER

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

Register	Address	R/W	Description	Reset Value
FGRA_YCLIP	0x72038018	RW	Y clipping coordinate register	0x00000000

FGRA_YCLIP	Bit	Description	Initial State
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

42.9.5 LEVEL OF DETAIL CONTROL REGISTER

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

Register	Address	R/W	Description	Reset Value
FGRA_LODCTL	0x7203C000	RW	Indicate LOD calculation control register	0x00000000

FGRA_LODCTL	Bit	Description	Initial State
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

42.9.6 WINDOW X CLIPPING REGION REGISTER

Register	Address	R/W	Description	Reset Value
FGRA_CLIPX	0x7203C004	RW	X clip coordinate register	0x00000000

FGRA_CLIPX	Bit	Description	Initial State
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

42.9.7 POINT WIDTH CONTROL REGISTERS

Point width value is clamped by Point Width Min and Point Width Max values.

Register	Address	R/W	Description	Reset Value
FGRA_PWIDTH	0x7203801C	RW	Point Width control register	0x3F800000
FGRA_PSIZE_MIN	0x72038020	RW	Point Width Min value control register	0x3F800000
FGRA_PSIZE_MAX	0x72038024	RW	Point Width Max value control register	0x45000000

FGRA_PWIDTH	Bit	Description	Initial State
PointWidth	[31:0]	Specify point width value (floating point).	0x3F800000

FGRA_PSIZE_MIN	Bit	Description	Initial State
PointSize_Min	[31:0]	Specify point width minimum value (floating point).	0x3F800000

FGRA_PSIZE_MAX	Bit	Description	Initial State
PointSize_Max	[31:0]	Specify point width maximum value (floating point).	0x45000000

42.9.8 COORD REPLACE CONTROL REGISTER

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.

Register	Address	R/W	Description	Reset Value
FGRA_COORDREPLACE	0x72038028	RW	Coord Replace control register	0x00000000

FGRA_COORDREPLAC	Bit	Description	Initial State
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

42.9.9 LINE WIDTH CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGRA_LWIDTH	0x7203802C	RW	Line Width control register	0x3F800000

FGRA_LWIDTH	Bit	Description	Initial State
LineWidth	[31:0]	Specify line width value (floating point).	0x3F800000

42.10 PIXEL SHADER

42.10.1 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 6-1. 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.

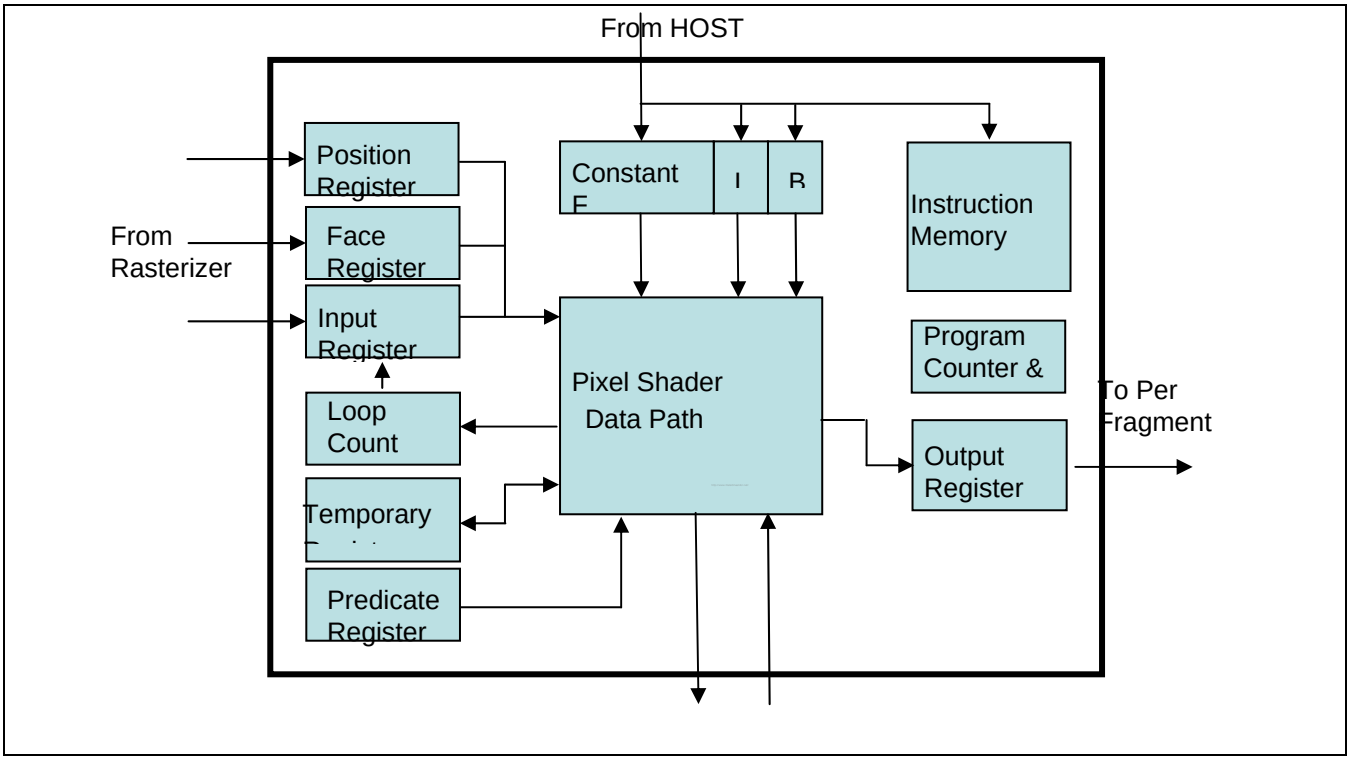


Figure 6-1 Pixel Shader Block Diagram

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”.

42.10.2 INSTRUCTION MEMORY

The instruction memory has 512 slots and 1 slot is composed of 4 words.

Register	Address	R/W	Description	Reset Value
INSTMEM	0x72040000 ~ 0x72041FFF	W	Instruction memory of pixel shader	0xX

See the Programmable Shader S/W Developer's Guide for more detail.

42.10.3 CONSTANT FLOAT REGISTER

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.)

Register	Address	R/W	Description	Reset Value
FGPS_CFLOAT	0x72044000 ~ 0x72044FFF	W	Constant float register of pixel shader	

Word 3 (0x72044XXC)

WORD3	Bit	Description	Initial State
W	[127:96]	Constant float W component value	0xXXXXXXXX

Word 2 (0x72044XX8)

WORD2	Bit	Description	Initial State
Z	[95:64]	Constant float Z component value	0xXXXXXXXX

Word 1 (0x72044XX4)

WORD1	Bit	Description	Initial State
Y	[63:32]	Constant float Y component value	0xXXXXXXXX

Word 0 (0x72044XX0)

WORD0	Bit	Description	Initial State
X	[31:0]	Constant float X component value	0xXXXXXXXX

♣ IEEE single precision floating-point format

	Bit	Description	Initial State
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

42.10.4 CONSTANT INTEGER REGISTER

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.

Register	Address	R/W	Description	Reset Value
FGPS_CINT	0x72048000 ~ 0x7204803F	W	A constant integer register of pixel shader	0xXXXXXXXX

Word (0x72048XX0)

WORD	Bit	Description	Initial State
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

42.10.5 CONSTANT BOOL REGISTER

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’.

Register	Address	R/W	Description	Reset Value
FGPS_CBOOL	0x72048400	W	Constant bool register of the pixel shader	0x0

FGPS_CBOOL	Bit	Description	Initial State
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

42.10.6 SPECIAL FUNCTION REGISTER FOR HW CONFIGURATION

Register	Address	R/W	Description	Reset Value
FGPS_ExeMode	0x7204C800	R/W	Pixel shader execution mode control register	0x0
FGPS_PCStart	0x7204C804	R/W	Start address of pixel shader program	0x0
FGPS_PCEnd	0x7204C808	R/W	End address of pixel shader program	0x0
FGPS_PCCopy	0x7204C80C	R/W	Copy PSPCS_ADDR value to program counter	0x0
FGPS_AttributeNum	0x7204C810	R/W	Number of attribute of current context	0x0
FGPS_IBStatus	0x7204C814	R	Stauts signal of PS Input Buffer initialization is NotReady 0 : Ready 1 : Not Ready	0x0

FGPS_ExeMode	Bit	Description	Initial State
reserved	[31:1]	reserved	0
FGPS_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 FGPS_IBStatus and FGPS_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 FGPS_IBStatus is '0'. Assert FGPS_ExeMode to '1'. PSExeMode → PSHostMode : Confirm IsNotEmpty_PS is '0' for all pixel shader. Assert FGPS_ExeMode to '0'	0b

FGPS_PCStart	Bit	Description	Initial State
reserved	[31:9]	reserved	0
FGPS_PCStart	[8:0]	When the pixel shader start operation, the first instruction, which is stored at FGPS_PCStart, is fetched from instruction memory. This register value should be copied to pixel shader program counter by FGPS_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

FGPS_PCEnd	Bit	Description	Initial State
reserved	[31:10]	reserved	0
FGPS_IgnorePCEnd	[9]	When this bit is set to 1, FGPS_PCEnd value is ignored and the only program counter stack empty condition is used for program termination.	0
FGPS_PCEnd	[8:0]	When the program counter is reached at the value of FGPS_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

FGPS_PCCopy	Bit	Description	Initial State
reserved	[31:1]	reserved	0
FGPS_PCCopy	[0]	When this bit is set to ‘1’, FGPS_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 FGPS_PCStart is not used, and the previous value is used for program start.	0b

FGPS_Attribute Num	Bit	Description	Initial State
reserved	[31:4]	reserved	0
FGPS_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, FGPS_ 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

FGPS_IBStatus	Bit	Description	Initial State
reserved	[31:1]	reserved	0
FGPS_IBStatus	[0]	This is read only register for monitoring pixel shader input buffer IsNotReady. When FGPS_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

42.11 TEXTURE UNITS

42.11.1 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 32-bit texture data.

Both compress format and paletted texture are not supported.

42.11.2 TEXTURE UNITS SPECIAL REGISTERS

42.11.2.1 Texture Status Register

Register	Address	R/W	Description	Reset Value
FGTU_TSTA0	0x72060000	R/W	Texture 0's status	0x08000000
FGTU_TSTA1	0x72060050	R/W	Texture 1's status	0x08000000
FGTU_TSTA2	0x720600A0	R/W	Texture 2's status	0x08000000
FGTU_TSTA3	0x720600F0	R/W	Texture 3's status	0x08000000
FGTU_TSTA4	0x72060140	R/W	Texture 4's status	0x08000000
FGTU_TSTA5	0x72060190	R/W	Texture 5's status	0x08000000
FGTU_TSTA6	0x720601E0	R/W	Texture 6's status	0x08000000
FGTU_TSTA7	0x72060230	R/W	Texture 7's status	0x08000000

FGTU_TSTAn	Bit	Description	Initial State
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

FGTU_TSTAn	Bit	Description	Initial State
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) 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	00000b
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]	reserved	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

42.11.2.2 TEXTURE U SIZE REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_USIZE0	0x72060004	R/W	Texture 0's U Size	0x00000000
FGTU_USIZE1	0x72060054	R/W	Texture 1's U Size	0x00000000
FGTU_USIZE2	0x720600A4	R/W	Texture 2's U Size	0x00000000
FGTU_USIZE3	0x720600F4	R/W	Texture 3's U Size	0x00000000
FGTU_USIZE4	0x72060144	R/W	Texture 4's U Size	0x00000000
FGTU_USIZE5	0x72060194	R/W	Texture 5's U Size	0x00000000
FGTU_USIZE6	0x720601E4	R/W	Texture 6's U Size	0x00000000
FGTU_USIZE7	0x72060234	R/W	Texture 7's U Size	0x00000000

FGTU_USIZEn	Bit	Description	Initial State
reserved	[31:11]	reserved	0
U_SIZE	[10:0]	U Size of Level 0 Texture	0x0

42.11.2.3 TEXTURE V SIZE REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_VSIZE0	0x72060008	R/W	Texture 0's V Size	0x00000000
FGTU_VSIZE1	0x72060058	R/W	Texture 1's V Size	0x00000000
FGTU_VSIZE2	0x720600A8	R/W	Texture 2's V Size	0x00000000
FGTU_VSIZE3	0x720600F8	R/W	Texture 3's V Size	0x00000000
FGTU_VSIZE4	0x72060148	R/W	Texture 4's V Size	0x00000000
FGTU_VSIZE5	0x72060198	R/W	Texture 5's V Size	0x00000000
FGTU_VSIZE6	0x720601E8	R/W	Texture 6's V Size	0x00000000
FGTU_VSIZE7	0x72060238	R/W	Texture 7's V Size	0x00000000

FGTU_VSIZEn	Bit	Description	Initial State
reserved	[31:11]	reserved	0
V_SIZE	[10:0]	V Size of Level 0 Texture	0x0

42.11.2.4 TEXTURE P SIZE REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_PSIZE0	0x7206000C	R/W	Texture 0's P Size	0x00000000
FGTU_PSIZE1	0x7206005C	R/W	Texture 1's P Size	0x00000000
FGTU_PSIZE2	0x720600AC	R/W	Texture 2's P Size	0x00000000
FGTU_PSIZE3	0x720600FC	R/W	Texture 3's P Size	0x00000000
FGTU_PSIZE4	0x7206014C	R/W	Texture 4's P Size	0x00000000
FGTU_PSIZE5	0x7206019C	R/W	Texture 5's P Size	0x00000000
FGTU_PSIZE6	0x720601EC	R/W	Texture 6's P Size	0x00000000
FGTU_PSIZE7	0x7206023C	R/W	Texture 7's P Size	0x00000000

FGTU_PSIZEn	Bit	Description	Initial State
reserved	[31:11]	reserved	0
P_SIZE	[10:0]	P Size of Level 0 Texture (the Depth of 3D Textures)	0x0

42.11.2.5 TEXTURE L1 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L1_0	0x72060010	R/W	Texture 0's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_1	0x72060060	R/W	Texture 1's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_2	0x720600B0	R/W	Texture 2's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_3	0x72060100	R/W	Texture 3's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_4	0x72060150	R/W	Texture 4's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_5	0x720601A0	R/W	Texture 5's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_6	0x720601F0	R/W	Texture 6's Level 1 Texture Offset	0x00000000
FGTU_TOFFS_L1_7	0x72060240	R/W	Texture 7's Level 1 Texture Offset	0x00000000

FGTU_TOFFS_L1n	Bit	Description	Initial State
reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 1 Texture Offset	0x0

42.11.2.6 TEXTURE L2 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L2_0	0x72060014	R/W	Texture 0's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_1	0x72060064	R/W	Texture 1's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_2	0x720600B4	R/W	Texture 2's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_3	0x72060104	R/W	Texture 3's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_4	0x72060154	R/W	Texture 4's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_5	0x720601A4	R/W	Texture 5's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_6	0x720601F4	R/W	Texture 6's Level 2 Texture Offset	0x00000000
FGTU_TOFFS_L2_7	0x72060244	R/W	Texture 7's Level 2 Texture Offset	0x00000000

FGTU_TOFFS_L2n	Bit	Description	Initial State
reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 2 Texture Offset	0x0

42.11.2.7 TEXTURE L3 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L3_0	0x72060018	R/W	Texture 0's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_1	0x72060068	R/W	Texture 1's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_2	0x720600B8	R/W	Texture 2's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_3	0x72060108	R/W	Texture 3's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_4	0x72060158	R/W	Texture 4's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_5	0x720601A8	R/W	Texture 5's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_6	0x720601F8	R/W	Texture 6's Level 3 Texture Offset	0x00000000
FGTU_TOFFS_L3_7	0x72060248	R/W	Texture 7's Level 3 Texture Offset	0x00000000

FGTU_TOFFS_L3n	Bit	Description	Initial State
Reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 3 Texture Offset	0x0

42.11.2.8 TEXTURE L4 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L4_0	0x7206001C	R/W	Texture 0's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_1	0x7206006C	R/W	Texture 1's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_2	0x720600BC	R/W	Texture 2's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_3	0x7206010C	R/W	Texture 3's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_4	0x7206015C	R/W	Texture 4's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_5	0x720601AC	R/W	Texture 5's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_6	0x720601FC	R/W	Texture 6's Level 4 Texture Offset	0x00000000
FGTU_TOFFS_L4_7	0x7206024C	R/W	Texture 7's Level 4 Texture Offset	0x00000000

FGTU_TOFFS_L4n	Bit	Description	Initial State
reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 4 Texture Offset	0x0

42.11.2.9 TEXTURE L5 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L5_0	0x72060020	R/W	Texture 0's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_1	0x72060070	R/W	Texture 1's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_2	0x720600C0	R/W	Texture 2's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_3	0x72060110	R/W	Texture 3's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_4	0x72060160	R/W	Texture 4's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_5	0x720601B0	R/W	Texture 5's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_6	0x72060200	R/W	Texture 6's Level 5 Texture Offset	0x00000000
FGTU_TOFFS_L5_7	0x72060250	R/W	Texture 7's Level 5 Texture Offset	0x00000000

FGTU_TOFFS_L5n	Bit	Description	Initial State
reserved	[31:23]	Reserved	0
OFFSET	[22:0]	Level 5 Texture Offset	0x0

42.11.2.10 TEXTURE L6 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L6_0	0x72060024	R/W	Texture 0's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_1	0x72060074	R/W	Texture 1's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_2	0x720600C4	R/W	Texture 2's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_3	0x72060114	R/W	Texture 3's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_4	0x72060164	R/W	Texture 4's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_5	0x720601B4	R/W	Texture 5's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_6	0x72060204	R/W	Texture 6's Level 6 Texture Offset	0x00000000
FGTU_TOFFS_L6_7	0x72060254	R/W	Texture 7's Level 6 Texture Offset	0x00000000

FGTU_TOFFS_L6n	Bit	Description	Initial State
reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 6 Texture Offset	0x0

42.11.2.11 TEXTURE L7 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L7_0	0x72060028	R/W	Texture 0's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_1	0x72060078	R/W	Texture 1's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_2	0x720600C8	R/W	Texture 2's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_3	0x72060118	R/W	Texture 3's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_4	0x72060168	R/W	Texture 4's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_5	0x720601B8	R/W	Texture 5's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_6	0x72060208	R/W	Texture 6's Level 7 Texture Offset	0x00000000
FGTU_TOFFS_L7_7	0x72060258	R/W	Texture 7's Level 7 Texture Offset	0x00000000

FGTU_TOFFS_L7n	Bit	Description	Initial State
Reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 7 Texture Offset	0x0

42.11.2.12 TEXTURE L8 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L8_0	0x7206002C	R/W	Texture 0's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_1	0x7206007C	R/W	Texture 1's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_2	0x720600CC	R/W	Texture 2's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_3	0x7206011C	R/W	Texture 3's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_4	0x7206016C	R/W	Texture 4's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_5	0x720601BC	R/W	Texture 5's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_6	0x7206020C	R/W	Texture 6's Level 8 Texture Offset	0x00000000
FGTU_TOFFS_L8_7	0x7206025C	R/W	Texture 7's Level 8 Texture Offset	0x00000000

FGTU_TOFFS_L8n	Bit	Description	Initial State
Reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 8 Texture Offset	0x0

42.11.2.13 TEXTURE L9 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L9_0	0x72060030	R/W	Texture 0's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_1	0x72060080	R/W	Texture 1's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_2	0x720600D0	R/W	Texture 2's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_3	0x72060120	R/W	Texture 3's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_4	0x72060170	R/W	Texture 4's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_5	0x720601C0	R/W	Texture 5's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_6	0x72060210	R/W	Texture 6's Level 9 Texture Offset	0x00000000
FGTU_TOFFS_L9_7	0x72060260	R/W	Texture 7's Level 9 Texture Offset	0x00000000

FGTU_TOFFS_L9n	Bit	Description	Initial State
reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 9 Texture Offset	0x0

42.11.2.14 TEXTURE L10 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L10_0	0x72060034	R/W	Texture 0's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_1	0x72060084	R/W	Texture 1's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_2	0x720600D4	R/W	Texture 2's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_3	0x72060124	R/W	Texture 3's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_4	0x72060174	R/W	Texture 4's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_5	0x720601C4	R/W	Texture 5's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_6	0x72060214	R/W	Texture 6's Level 10 Texture Offset	0x00000000
FGTU_TOFFS_L10_7	0x72060264	R/W	Texture 7's Level 10 Texture Offset	0x00000000

FGTU_TOFFS_L10n	Bit	Description	Initial State
reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 10 Texture Offset	0x0

42.11.2.15 TEXTURE L11 OFFSET REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TOFFS_L11_0	0x72060038	R/W	Texture 0's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_1	0x72060088	R/W	Texture 1's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_2	0x720600D8	R/W	Texture 2's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_3	0x72060128	R/W	Texture 3's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_4	0x72060178	R/W	Texture 4's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_5	0x720601C8	R/W	Texture 5's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_6	0x72060218	R/W	Texture 6's Level 11 Texture Offset	0x00000000
FGTU_TOFFS_L11_7	0x72060268	R/W	Texture 7's Level 11 Texture Offset	0x00000000

FGTU_TOFFS_L11n	Bit	Description	Initial State
reserved	[31:23]	reserved	0
OFFSET	[22:0]	Level 11 Texture Offset	0x0

42.11.2.16 TEXTURE MIN LEVEL REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_T_MIN_L0	0x7206003C	R/W	Texture 0's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 1	0x7206008C	R/W	Texture 1's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 2	0x720600DC	R/W	Texture 2's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 3	0x7206012C	R/W	Texture 3's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 4	0x7206017C	R/W	Texture 4's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 5	0x720601CC	R/W	Texture 5's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 6	0x7206021C	R/W	Texture 6's Mipmap Min Level	0x00000000
FGTU_T_MIN_L 7	0x7206026C	R/W	Texture 7's Mipmap Min Level	0x00000000

FGTU_T_MIN_Ln	Bit	Description	Initial State
reserved	[31:4]	reserved	0
MIN_LEVEL	[3:0]	Texture Mipmap Min level	0x0

42.11.2.17 TEXTURE MAX LEVEL REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_T_MAX_L0	0x72060040	R/W	Texture 0's Mipmap Max Level	0x00000000
FGTU_T_MAX_L1	0x72060090	R/W	Texture 1's Mipmap Max Level	0x00000000
FGTU_T_MAX_L2	0x720600E0	R/W	Texture 2's Mipmap Max Level	0x00000000
FGTU_T_MAX_L3	0x72060130	R/W	Texture 3's Mipmap Max Level	0x00000000
FGTU_T_MAX_L4	0x72060180	R/W	Texture 4's Mipmap Max Level	0x00000000
FGTU_T_MAX_L5	0x720601D0	R/W	Texture 5's Mipmap Max Level	0x00000000
FGTU_T_MAX_L6	0x72060220	R/W	Texture 6's Mipmap Max Level	0x00000000
FGTU_T_MAX_L7	0x72060270	R/W	Texture 7's Mipmap Max Level	0x00000000

FGTU_T_MAX_Ln	Bit	Description	Initial State
reserved	[31:4]	reserved	0
MAX_LEVEL	[3:0]	Texture Mipmap Max Level	0x0

42.11.2.18 TEXTURE BASE ADDRESS REGISTER 0 ~ 7

Register	Address	R/W	Description	Reset Value
FGTU_TBADD0	0x72060044	R/W	Texture 0's base address	0x00000000
FGTU_TBADD1	0x72060094	R/W	Texture 1's base address	0x00000000
FGTU_TBADD2	0x720600E4	R/W	Texture 2's base address	0x00000000
FGTU_TBADD3	0x72060134	R/W	Texture 3's base address	0x00000000
FGTU_TBADD4	0x72060184	R/W	Texture 4's base address	0x00000000
FGTU_TBADD5	0x720601D4	R/W	Texture 5's base address	0x00000000
FGTU_TBADD6	0x72060224	R/W	Texture 6's base address	0x00000000
FGTU_TBADD7	0x72060274	R/W	Texture 7's base address	0x00000000

FGTU_TBADDn	Bit	Description	Initial State
ADDR	[31:0]	Texture Base Address for Level 0	0xXXXXXXXX

42.11.2.19 TEXTURE COLOR KEY REGISTER

Register	Address	R/W	Description	Reset Value
FGTU_CKEY1	0x72060280	R/W	3D color key1 register	0x00000000
FGTU_CKEY2	0x72060284	R/W	3D color key2 register	0x00000000
FGTU_CKYUV	0x72060288	R/W	3D color key YUV register	0x00000000
FGTU_CKMASK	0x7206028C	R/W	3D color key mask register	0x00000000
FGTU_PALLETTE_ADDR	0x72060290	W	Palette address for indexed texture	0x00000000
FGTU_PALLETTE_IN	0x72060294	W	Palette data in	0x00000000

FGTU_CKEY1	Bit	Description	Initial State
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

FGTU_CKEY2	Bit	Description	Initial State
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

FGTU_CKYUV	Bit	Description	Initial State
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

FGTU_CKMASK	Bit	Description	Initial State
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

FGTU_PALLETTE_ADDR	Bit	Description	Initial State
reserved	[31:8]	Reserved	0
ADDR	[7:0]	Palette address	0x0

FGTU_PALLETTE_IN	Bit	Description	Initial State
DATA	[31:0]	Palette data in	0x0

42.11.2.20 VERTEX TEXTURE STATUS REGISTER

Register	Address	R/W	Description	Reset Value
FGVTU_VTSTA0	0x720602C0	R/W	Vertex texture 0's status	0x00000000
FGVTU_VTSTA1	0x720602C8	R/W	Vertex texture 1's status	0x00000000
FGVTU_VTSTA2	0x720602D0	R/W	Vertex texture 2's status	0x00000000
FGVTU_VTSTA3	0x720602D8	R/W	Vertex texture 3's status	0x00000000

FGVTU_VTSTAn	Bit	Description	Initial State
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

42.11.2.21 VERTEX TEXTURE BASE ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
FGVTU_VTBADDR0	0x720602C4	R/W	Vertex texture 0's base address	0x00000000
FGVTU_VTBADDR1	0x720602CC	R/W	Vertex texture 1's base address	0x00000000
FGVTU_VTBADDR2	0x720602D4	R/W	Vertex texture 2's base address	0x00000000
FGVTU_VTBADDR3	0x720602DC	R/W	Vertex texture 3's base address	0x00000000

FGVTU_VTBADDRn	Bit	Description	Initial State
ADDR	[31:0]	Vertex texture bass address	0xXXXXXXXX

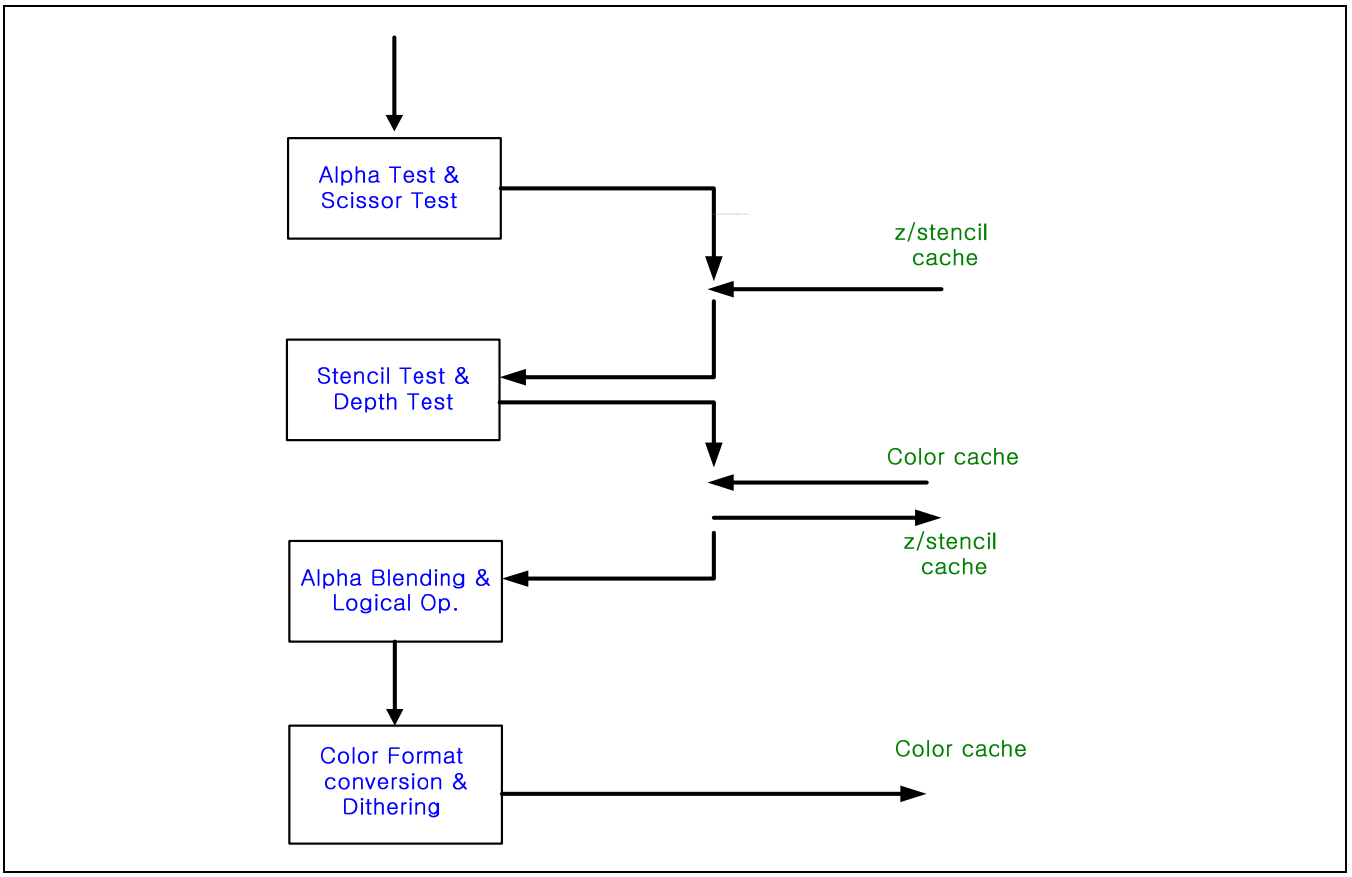
42.12 PER-FRAGMENT UNIT

42.12.1 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.



Above Figure is Per-fragment unit's functional block diagram

42.12.2 PER-FRAGMENT UNIT SPECIAL REGISTERS

42.12.2.1 SCISSOR TEST CONTROL REGISTER

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.

Register	Address	R/W	Description	Reset Value
FGPF_SCISSOR_X	0x72070000	R/W	X coordinate pixel clipping region (X coordinate's range is from 0 to 2047)	0x00000000
FGPF_SCISSOR_Y	0x72070004	R/W	Y coordinate pixel clipping region (Y coordinate's range is from 0 to 2047)	0x00000000

FGPF_SCISSOR_X	Bit	Description	Initial State
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

FGPF_SCISSOR_Y	Bit	Description	Initial State
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

42.12.2.2 ALPHA TEST CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_ALPHAT	0x72070008	R/W	Alpha test control register	0x00000000

FGPF_ALPHAT	Bit	Description	Initial State
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

42.12.2.3 STENCIL TEST CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_FRONTST	0x7207000C	R/W	Front face stencil test control register	0x0000000
FGPF_BACKST	0x72070010	R/W	Back face stencil test control register	0x0000000

FGPF_FRONTST	Bit	Description	Initial State
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

FGPF_BACKST	Bit	Description	Initial State
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

42.12.2.4 DEPTH TEST CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_DEPTHT	0x72070014	R/W	Depth test control register	0x00000002

FGPF_DEPTHT	Bit	Description	Initial State
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

42.12.2.5 BLENDING CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_CCLR	0x72070018	R/W	Blend constant color	0x00000000
FGPF_BLEND	0x7207001C	R/W	Blending control register	0x00000000

FGPF_CCLR	Bit	Description	Initial State
VAL	[31:0]	Blend constant RGBA color	0x0

FGPF_BLEND	Bit	Description	Initial State
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

42.12.2.6 LOGICAL OPERATION CONTROL REGISTER

Object의 color와 alpha값을 의미하는 source 와 현재 frame buffer의 color와 alpha 값을 의미하는 destination 간의 binary logic operation을 수행한다.

Register	Address	R/W	Description	Reset Value
FGPF_LOGOP	0x72070020	R/W	RGBA color logical operation enable & function	0x00000000

FGPF_LOGOP	Bit	Description	Initial State
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

42.12.2.7 COLOR BUFFER WRITE MASK REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_CBMSK	0x72070024	R/W	Color write mask in RGBA mode	0x00000000

FGPF_CBMSK	Bit	Description	Initial State
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 initial state, 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

42.12.2.8 DEPTH/STENCIL BUFFER WRITE MASK REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_DBMSK	0x72070028	R/W	Depth/Stencil buffer write mask	0x00000000

FGPF_DBMSK	Bit	Description	Initial State
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 initial state, 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 initial state, 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 initial state, the depth buffer is enabled for writing. 0b = depth buffer writing 1b = no depth buffer writing	0b

42.12.2.9 FRAME BUFFER CONTROL REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_FBCTL	0x7207002C	R/W	Frame buffer write control register	0x00000000

FGPF_FBCTL	Bit	Description	Initial State
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

42.12.2.10 DEPTH BUFFER BASE ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_DBADDR	0x72070030	R/W	Depth buffer offset address	0x00000000

FGPF_DBADDR	Bit	Description	Initial State
FbDepthOffset	[31:0]	Depth buffer offset address	0x00000000

42.12.2.11 COLOR BUFFER BASE ADDRESS REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_CBADDR	0x72070034	R/W	Color buffer offset address	0x0000000

FGPF_CBADDR	Bit	Description	Initial State
FbColorOffset	[31:0]	Color buffer offset address	00000000h

42.12.2.12 FRAME BUFFER WIDTH REGISTER

Register	Address	R/W	Description	Reset Value
FGPF_FBW	0x72070038	R/W	Frame buffer width	0x000007FF

FGPF_FBW	Bit	Description	Initial State
reserved	[31:11]	reserved	0
FbWidth	[11:0]	Frame buffer width (0~2048)	800h

42.13 AXI ARBITER & AXI DMA

42.13.1 OVERVIEW

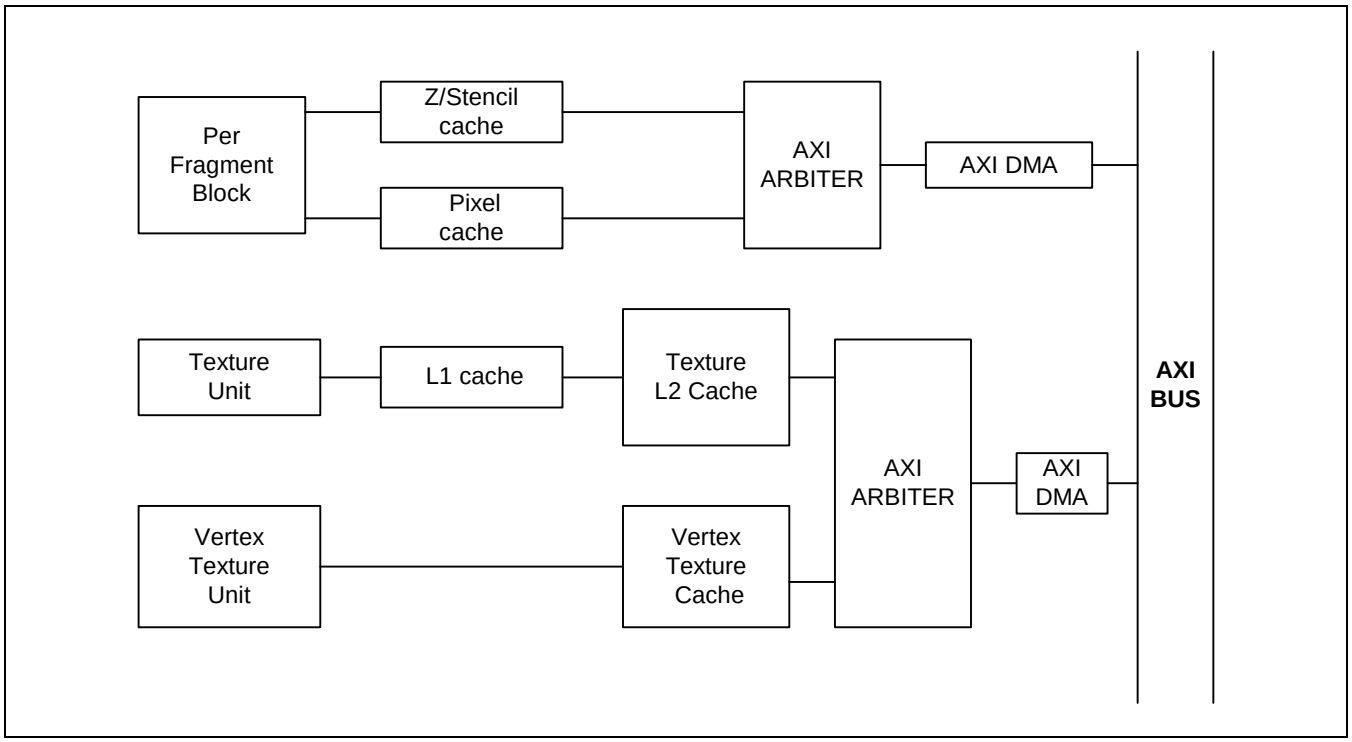


Figure ???

- 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

- 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 (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 Arbter1 (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

- GRAPHIC 3D AXI DMA support AMBA AXI BUS protocol

43 AXI BUSES

43.1 OVERVIEW

S3C6410X holds following three AXI buses, which interconnects different building blocks together to form a single SoC.

- **AXI_SYS** serves as a system bus.
- **AXI_PERI** implements a channel for SFR access.
- **AXI_SFR** supplements AXI_PERI for SFR accesses to AHB sub-systems.

43.2 FEATURES

- Sparse connection options to reduce gate count and improve security
- Support for AXI at 32-bit or 64-bit data widths
- Support for AMBA 2 APB and AMBA 3 APB at 32-bit data width
- Programmable QoS scheme
- APB interface to provide access to programming registers

43.3 ARBITRATION SCHEME

All AXI buses, i.e., AXI_SYS, AXI_PERI, and AXI_SFR, implement arbiter, which makes it possible to arbitrate between multiple bus masters. Arbitration scheme can be any of the followings.

- Fixed-priority
- Round-robin
- Combination of fixed-priority and round-robin

The default arbitration scheme uses both fixed-priority and round-roubind scheme but users can change the scheme anytime during boot or run-time by updating arbitration control registers you can access through an APB port. Each AXI bus implements a single shared arbiter for both read and write channels. This means that a slave can only accept simultaneous read and write transactions from the same master.

43.3.1 PRIORITY GROUPS

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 of which a member is trying to win access, and within that group, to the highest master at that time. When a 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.

43.3.2 FIXED PRIORITY OPERATION

If all master arbitration order values are the same, then a round-robin style priority scheme is implemented. The reason that it behaves like a round-robin style priority scheme 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 Least Recently Granted (LRG) master at the top, to the Most Recently Granted (MRG) at the bottom.

43.3.3 CONCURRENT OPERATION

The round-robin and fixed priority modes concurrently exist when the control registers are programmed with a combination of identical and unique arbitration order 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.

43.4 PROGRAMMABLE QUALITY-OF-SERVICE (QOS)

43.4.1 QOS TIDEMARK REGISTER

You can program this with the number of slots of the combined issuing capability of the master interface that you want to reserve for QoS. The maximum value that you can write to this location is one less than the combined issuing capability of the master interface. This ensures that there is always at least one unreserved slot so that deadlock does not occur. If you attempt to write a greater value, then afterwards the register contains the maximum permitted value, and no error is given. This permits the following process to discover the maximum number of reserved slots for the master interface:

1. Attempt to write 31, that is 0x1F, to the qos_tidemark register.
2. Read back the register contents.

The value obtained is the maximum possible number of reserved slots.

43.4.2 QOS ACCESS CONTROL REGISTER

A 1 in any bit of this register indicates that the slave interface corresponding to the bit position is permitted to use the reserved slots of the connected combined acceptance capability of the slaves.

The maximum value that you can write to this register is:

(2^<total number of slave interfaces>) -1

If you attempt to write a greater value, then afterwards the register contains the maximum permitted value, and no error is given. Changes to these values occur on the first possible arbitration time after they are written.

43.5 BUS INTERCONNECTION

This section shows interconnection of AXI buses.

43.5.1 SLAVE INTERFACE (MASTER PORT) OF AXI_SYS

Port ID	Interconnection	Initial arbitration order
S0	AHB_I	1
S1	AHB_F	0
S2	AHB_P	2
S3	AXI_V	7
S4	AHB_X	8
S5	AHB_T	9
S6	AHB_M	3
S7	AHB_S	4
S8	ARM_I	11
S9	ARM_RW	12
S10	ARM_DMA	13
S11	AHB_CF	10
S12	G3D_PF0	5
S13	G3D_TC	6
S14	G2D	2

43.5.2 MASTER INTERFACE (SLAVE PORT) OF AXI_SYS

Port ID	Interconnection
M0	AHB_IROM
M1	AHB_SMC
M2	RESERVED
M3	DMC1

43.5.3 SLAVE INTERFACE (MASTER PORT) OF AXI_PERI

Port ID	Interconnection	Initial arbitration order
S0	ARM_P	0
S1	AHB_M	1
S2	AHB_T	2
S3	AHB_S	3

43.5.4 MASTER INTERFACE (SLAVE PORT) OF AXI_PERI

Port ID	Interconnection
M0	AHB_SMC
M1	AHB_VICTZIC
M2	AHB_G3D
M3	AXI_SFR
M4	APB0
M5	APB1

43.5.5 SLAVE INTERFACE (MASTER PORT) OF AXI_SFR

Port ID	Interconnection	Initial arbitration order
S0	AXI_PERI	0

43.5.6 MASTER INTERFACE (SLAVE PORT) OF AXI_SFR

Port ID	Interconnection
M0	ETB
M1	AHB_T
M2	AHB_M
M3	AHB_P
M4	AHB_F
M5	AHB_I
M6	AHB_X
M7	AHB_S

43.6 SPARSE INTERCONNECTION

PL301 implements spare interconneciton.

43.6.1 SPARE INTERCONNECTION OF AXI_SYS

Following table shows sparse interconnection of AXI_SYS bus.

	M0	M1	M2	M3
S0	X	O	O	O
S1	X	O	O	O
S2	X	O	O	O
S3	X	O	O	O
S4	X	O	O	O
S5	X	O	O	O
S6	X	O	O	O
S7	O	O	O	O
S8	O	O	O	O
S9	O	O	O	O
S10	O	O	O	O
S11	X	O	O	O
S12	X	O	O	O
S13	X	O	O	O
S14	X	O	O	O

43.6.2 SPARE INTERCONNECTION OF AXI_PERI

Following table shows sparse interconnection of AXI_PERI bus.

	M0	M1	M2	M3	M4	M5
S0	O	O	O	O	O	O
S1	O	X	O	O	O	O
S2	O	O	O	O	O	O
S3	O	X	O	O	O	O

43.6.3 SPARE INTERCONNECTION OF AXI_SFR

AXI_SFR does not implement sparse interconnection. And all bus masters can access all bus slaves.

43.7 CONNECTING 32BIT BUS MASTER TO 64BIT AXI_SYS

AXI_SYS is implemented with AXI-compliant backbone bus with 64-bit data width. When bus master with 32bit data width is accessing AXI_SYS with 64bit data width, there are two choices how to generate 64-bit backbone data from 32-bit data.

1. Just use lower 32-bit of AXI_SYS backbone channel and invalid the remaining upper 32-bit. This is inefficient in terms of bus usage but it can be applied to any combinations of bus master and slaves without limitations.
2. To minimize the traffic of AXI_SYS, two 32-bit bus transfers can be packed together to become a single 64-bit transfer on AXI_SYS. This is efficient use of AXI_SYS bus channel, but this is not applicable to all bus master and slave pairs.

Following table shows whether packing of two 32-bit bus transactions are possible for each bus master and slave pair.

	M0	M1	M2	M3
S0	No channel	NO-PACKING	**PACKING**	**PACKING**
S1	No channel	NO-PACKING	**PACKING**	**PACKING**
S2	No channel	NO-PACKING	**PACKING**	**PACKING**
S3	No channel	NO-PACKING	**PACKING**	**PACKING**
S4	No channel	NO-PACKING	**PACKING**	**PACKING**
S5	No channel	NO-PACKING	**PACKING**	**PACKING**
S6	No channel	NO-PACKING	**PACKING**	**PACKING**
S7	NO-PACKING	NO-PACKING	**PACKING**	**PACKING**
S8	64bit bus master	64bit bus master	64bit bus master	64bit bus master
S9	64bit bus master	64bit bus master	64bit bus master	64bit bus master
S10	64bit bus master	64bit bus master	64bit bus master	64bit bus master
S11	No channel	NO-PACKING	**PACKING**	**PACKING**
S12	No channel	64bit bus master	64bit bus master	64bit bus master
S13	No channel	64bit bus master	64bit bus master	64bit bus master
S14	No channel	64bit bus master	64bit bus master	64bit bus master

Each partition in the upper table has following meanings.

* No channel: There is no channel between bus master and slave due to spare interconnection.

* NO-PACKING: Packing of two 32-bit transactions to a single 64-bit transaction is not allowed.

* PACKING: Packing of two 32-bit transactions to a single 64-bit transaction is possible.

* 64-bit bus master: The bus master is 64bit-wide, so no packing is needed.

The decision whether packing is used or not is decided by BUS_CACHEABLE_CON residing at address 0x7E00_F834. Refer to system controller manual for details.

43.8 AXI BUS MASTER IDS

On AXI bus, each bus mater has a dedicated unique ID for use for bus slave.

AXI ID	Master bus name	Related IPs
0000_0000	I Block	Camera, JPEG
0000_0001	F Block	Display Controller
0000_0010	P Block	2D, TV Encoder, TV Scaler
XXXX_00111	V Block	MFC
0000_0100	X Block	HSMMC, USB OTG
0000_0101	T Block	Host I/F
0000_0110	M Block	DMA0, DMA1
0000_0111	S Block	Security Sub Block, SDMA0, SDMA1
0000_1000	ARM Instruction	ARM Core Instruction
0000_1001	ARM Data	ARM Core Data
0000_1010	ARM DMA	ARM Core DMA
0000_1011	CF	CFCON
000X_11001	G block	G3D
000X_11011	G block	G3D
0000_1110	G2D	G2D

[1] X refers to ID that can be dynamically changed by bus master.

43.9 REGISTER DESCRIPTION

43.9.1 ARBITRATION ORDER NUMBER FOR AXI_SYS

Register	Address	R/W	Description	Reset Value
AXI_SYS_ARBIT_CTRL_REG0	0x7E003000	RW	Arbitration order number for slave interface 0	0x0000_0001
AXI_SYS_ARBIT_CTRL_REG1	0x7E003020	RW	Arbitration order number for slave interface 1	0x0000_0000
AXI_SYS_ARBIT_CTRL_REG2	0x7E003040	RW	Arbitration order number for slave interface 2	0x0000_0002
AXI_SYS_ARBIT_CTRL_REG3	0x7E003060	RW	Arbitration order number for slave interface 3	0x0000_0007
AXI_SYS_ARBIT_CTRL_REG4	0x7E003080	RW	Arbitration order number for slave interface 4	0x0000_0008
AXI_SYS_ARBIT_CTRL_REG5	0x7E0030A0	RW	Arbitration order number for slave interface 5	0x0000_0009
AXI_SYS_ARBIT_CTRL_REG6	0x7E0030C0	RW	Arbitration order number for slave interface 6	0x0000_0003
AXI_SYS_ARBIT_CTRL_REG7	0x7E0030E0	RW	Arbitration order number for slave interface 7	0x0000_0004
AXI_SYS_ARBIT_CTRL_REG8	0x7E003100	RW	Arbitration order number for slave interface 8	0x0000_000B
AXI_SYS_ARBIT_CTRL_REG9	0x7E003120	RW	Arbitration order number for slave interface 9	0x0000_000C
AXI_SYS_ARBIT_CTRL_REG10	0x7E003140	RW	Arbitration order number for slave interface 10	0x0000_000D
AXI_SYS_ARBIT_CTRL_REG11	0x7E003160	RW	Arbitration order number for slave interface 11	0x0000_000A
AXI_SYS_ARBIT_CTRL_REG12	0x7E003180	RW	Arbitration order number for slave interface 12	0x0000_0005
AXI_SYS_ARBIT_CTRL_REG13	0x7E0031A0	RW	Arbitration order number for slave interface 13	0x0000_0006
AXI_SYS_ARBIT_CTRL_REG14	0x7E0031C0	RW	Arbitration order number for slave interface 14	0x0000_0002

AXI_SYS_ARBIT_CTRL_REGn	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x0000_000
order value	[7:0]	Arbitration order number for slave interface n	-

43.9.2 QOS TIDEMARK FOR AXI_SYS

Register	Address	R/W	Description	Reset Value
AXI_SYS_QOS_CTRL_REG0	0x7E003400	RW	QoS Tiemark for master interface 0.	0x0000_0000
AXI_SYS_QOS_CTRL_REG1	0x7E003404	RW	QoS Access Control for master interface 0.	0x0000_0000
AXI_SYS_QOS_CTRL_REG2	0x7E003420	RW	QoS Tiemark for master interface 1.	0x0000_0000
AXI_SYS_QOS_CTRL_REG3	0x7E003424	RW	QoS Access Control for master interface 1.	0x0000_0000
AXI_SYS_QOS_CTRL_REG4	0x7E003440	RW	QoS Tiemark for master interface 2.	0x0000_0000
AXI_SYS_QOS_CTRL_REG5	0x7E003444	RW	QoS Access Control for master interface 2.	0x0000_0000
AXI_SYS_QOS_CTRL_REG6	0x7E003460	RW	QoS Tiemark for master interface 3.	0x0000_0000
AXI_SYS_QOS_CTRL_REG7	0x7E003464	RW	QoS Access Control for master interface 3.	0x0000_0000

AXI_SYS_ARBIT_CTRL_REG0	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x0000_0000

AXI_SYS_ARBIT_CTRL_REG1	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x0000_0000

AXI_SYS_ARBIT_CTRL_REG2	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x0000_0000

AXI_SYS_ARBIT_CTRL_REG3	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x0000_0000

AXI_SYS_ARBIT_CTRL_REG4	Bit	Description	Initial State
Reserved	[31:6]	Reserved	0x0
QoS Tidemark	[5:0]	QoS Tidemark for master interface 2.[2]	0x0

[2] QoS Tidemark can be set between 0 and 7. If write data bigger than 7 is written to this location, it is set to 7.

AXI_SYS_ARBIT_CTRL_REG5	Bit	Description	Initial State
Reserved	[31:15]	Reserved	0x0
QoS Tidemark	[14:0]	QoS Access Control for master interface 2.[3]	0x0

AXI_SYS_ARBIT_CTRL_REG6	Bit	Description	Initial State
Reserved	[31:6]	Reserved	0x0
QoS Tidemark	[5:0]	QoS Tidemark for master interface 3.[2]	0x0

AXI_SYS_ARBIT_CTRL_REG7	Bit	Description	Initial State
Reserved	[31:15]	Reserved	0x0
QoS Tidemark	[14:0]	QoS Access Control for master interface 3.[3]	0x0

43.5.3 ARBITRATION ORDER NUMBER FOR AXI_PERI

Register	Address	R/W	Description	Reset Value
AXI_PERI_ARBIT_CTRL_REG0	0x7E008000	RW	Arbitration order number for slave interface 0	0x0000_0000
AXI_PERI_ARBIT_CTRL_REG1	0x7E008020	RW	Arbitration order number for slave interface 1	0x0000_0001
AXI_PERI_ARBIT_CTRL_REG2	0x7E008040	RW	Arbitration order number for slave interface 2	0x0000_0002
AXI_PERI_ARBIT_CTRL_REG3	0x7E008060	RW	Arbitration order number for slave interface 3	0x0000_0003

AXI_PERI_ARBIT_CTRL_REGn	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x0000_000
order value	[7:0]	Arbitration order number for slave interface n	-

[3] Each bit position corresponds to slave interface of AXI_SYS.

43.9.4 QOS TIDEMARK FOR AXI_PERI

Register	Address	R/W	Description	Reset Value
AXI_PERI_QOS_CTRL_REG0	0x7E008400	RW	QoS Tiemark for master interface 0.	0x0000_0000
AXI_PERI_QOS_CTRL_REG1	0x7E008404	RW	QoS Access Control for master interface 0.	0x0000_0000
AXI_PERI_QOS_CTRL_REG2	0x7E008420	RW	QoS Tiemark for master interface 1.	0x0000_0000
AXI_PERI_QOS_CTRL_REG3	0x7E008424	RW	QoS Access Control for master interface 1.	0x0000_0000
AXI_PERI_QOS_CTRL_REG4	0x7E008440	RW	QoS Tiemark for master interface 2.	0x0000_0000
AXI_PERI_QOS_CTRL_REG5	0x7E008444	RW	QoS Access Control for master interface 2.	0x0000_0000
AXI_PERI_QOS_CTRL_REG6	0x7E008460	RW	QoS Tiemark for master interface 3.	0x0000_0000
AXI_PERI_QOS_CTRL_REG7	0x7E008464	RW	QoS Access Control for master interface 3.	0x0000_0000
AXI_PERI_QOS_CTRL_REG8	0x7E008480	RW	QoS Tiemark for master interface 4.	0x0000_0000
AXI_PERI_QOS_CTRL_REG9	0x7E008484	RW	QoS Access Control for master interface 4.	0x0000_0000
AXI_PERI_QOS_CTRL_REG10	0x7E0084A0	RW	QoS Tiemark for master interface 5.	0x0000_0000
AXI_PERI_QOS_CTRL_REG11	0x7E0084A4	RW	QoS Access Control for master interface 5.	0x0000_0000

AXI_PERI_ARBIT_CTRL_REGn	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x0000_0000

43.9.5 ARBITRATION ORDER NUMBER FOR AXI_SFR

Register	Address	R/W	Description	Reset Value
AXI_SFR_ARBIT_CTRL_REG0	0x7E009000	RW	Arbitration order number for slave interface 0	0x0000_0000

AXI_SFR_ARBIT_CTRL_REGn	Bit	Description	Initial State
Reserved	[31:8]	Reserved	0x0000_000
order value	[7:0]	Arbitration order number for slave interface n	-

43.9.6 QOS TIDEMARK FOR AXI_SFR

Register	Address	R/W	Description	Reset Value
AXI_SFR_QOS_CTRL_REG0	0x7E009400	RW	QoS Tiemark for master interface 0.	0x0000_0000
AXI_SFR_QOS_CTRL_REG1	0x7E009404	RW	QoS Access Control for master interface 0.	0x0000_0000
AXI_SFR_QOS_CTRL_REG2	0x7E009420	RW	QoS Tiemark for master interface 1.	0x0000_0000
AXI_SFR_QOS_CTRL_REG3	0x7E009424	RW	QoS Access Control for master interface 1.	0x0000_0000
AXI_SFR_QOS_CTRL_REG4	0x7E009440	RW	QoS Tiemark for master interface 2.	0x0000_0000
AXI_SFR_QOS_CTRL_REG5	0x7E009444	RW	QoS Access Control for master interface 2.	0x0000_0000
AXI_SFR_QOS_CTRL_REG6	0x7E009460	RW	QoS Tiemark for master interface 3.	0x0000_0000
AXI_SFR_QOS_CTRL_REG7	0x7E009464	RW	QoS Access Control for master interface 3.	0x0000_0000
AXI_SFR_QOS_CTRL_REG8	0x7E009480	RW	QoS Tiemark for master interface 4.	0x0000_0000
AXI_SFR_QOS_CTRL_REG9	0x7E009484	RW	QoS Access Control for master interface 4.	0x0000_0000
AXI_SFR_QOS_CTRL_REG10	0x7E0094A0	RW	QoS Tiemark for master interface 5.	0x0000_0000
AXI_SFR_QOS_CTRL_REG11	0x7E0094A4	RW	QoS Access Control for master interface 5.	0x0000_0000
AXI_SFR_QOS_CTRL_REG12	0x7E0094C0	RW	QoS Tiemark for master interface 6.	0x0000_0000
AXI_SFR_QOS_CTRL_REG13	0x7E0094C4	RW	QoS Access Control for master interface 6.	0x0000_0000
AXI_SFR_QOS_CTRL_REG14	0x7E0094E0	RW	QoS Tiemark for master interface 7.	0x0000_0000
AXI_SFR_QOS_CTRL_REG15	0x7E0094E4	RW	QoS Access Control for master interface 7.	0x0000_0000

AXI_SFR_ARBIT_CTRL_REGn	Bit	Description	Initial State
Reserved	[31:0]	Reserved	0x0000_0000

44 ELECTRICAL DATA

44.1 ABSOLUTE MAXIMUM RATINGS

The following list of absolute maximum ratings is specified over operating junction temperature range. Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum rated conditions for extended periods may affect device reliability.

Table 44-1. Absolute Maximum Rating

Parameter	Symbol	Min	Max	Unit
DC Supply Voltage	VDDapll, VDDepll, VDDmpll, VDDotgi, VDDINT, VDDalive	-0.5	1.8	V
	VDDarm	-0.5	1.8	
	VDDmm, VDDhi, VDDlcd, VDDpcm, VDDext, VDDsys	-0.5	4.6	
	VDDADC, VDDDAC	-0.5	4.6	
	VDDm0, VDDss, VDDm1	-0.5	3.6	
	VDDotg, VDDuh	-0.5	4.6	
	VDDrtc	-0.5	4.6	
DC Input Voltage	VIN(1.8v Input buffer)	-0.5	2.5	
	VIN(2.5v Input buffer)	-0.5	3.6	
	VIN(3.3v Input buffer)	-0.5	4.6	
DC Output Voltage	VOUT(1.8v output buffer)	-0.5	2.5	
	VOUT(2.5v output buffer)	-0.5	3.6	
	VOUT(3.3v output buffer)	-0.5	4.6	

DC Input Current	IIN	± 20	mA
Storage Temperature	TSTG	– 65 to 150	°C

44.2 RECOMMENDED OPERATING CONDITIONS

S3C6410X must be used under the operating conditions listed in Table 44−2.

Table 44-2. Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
DC Supply Voltage for Alive Block	VDDalive	1.15	1.2	1.25	V
DC Supply Voltage for PLL	VDDepll, VDDmpll, VDDapll,	1.15	1.2	1.25	
DC Supply Voltage for Internal @ ARM Core Frequency:533Mhz	VDDint	1.15	1.2	1.25	
DC Supply Voltage for Internal @ ARM Core Frequency:667Mhz	VDDint	1.25	1.3	1.35	
DC Supply Voltage for USB OTG Logic	VDDotgi,	1.2-5%	1.2	1.2+5%	
DC Supply Voltage for ARM Core @ ARM Core Frequency:533Mhz	VDDarm	1.05	1.1	1.15	
DC Supply Voltage for ARM Core @ ARM Core Frequency:667Mhz	VDDarm	1.15	1.2	1.25	
DC Supply Voltage for I/O Block	VDDmm	1.7	1.8~3.3	3.6	
	VDDhi	1.7	1.8~3.3	3.6	
	VDDlcd	1.7	1.8~3.3	3.6	
	VDDpcm	1.7	1.8~3.3	3.6	
	VDDext	1.7	1.8~3.3	3.6	
	VDDsys	1.7	1.8~3.3	3.6	
	VDDss	1.7	1.8~3.3	3.6	

DC Supply Voltage for Memory Interface	VDDm0	1.7	1.8~3.3	3.6	
	VDDm1	1.75	1.8/2.5	2.7	
DC Supply Voltage for RTC	VDDrtc	1.7	3.0	3.3	
DC Supply Voltage for USB	VDDotg,	3.3-5%	3.3	3.3+5%	
DC Supply Voltage for USB Host	VDDuh	3.0	3.3	3.6	
DC Supply Voltage for ADC	VDDADC	3.0	3.3	3.6	
DC Supply Voltage for DAC	VDDDAC	3.0	3.3	3.6	
DC Input Voltage	VIN	3.0	3.3	3.6	
		2.3	2.5	2.7	
		1.7	1.8	1.9	
DC Output Voltage	VOUT	3.0	3.3	3.6	
		2.3	2.5	2.7	
		1.7	1.8	1.9	
Operating Temperature	TA	Industrial	-40 to 85		°C
		Extended	-20 to 70		

44.3 D.C. ELECTRICAL CHARACTERISTICS

All The DC characteristics for each pin include input sense levels, output drive levels, and currents. These parameters can be used to determine maximum DC loading and to determine maximum transition times for a given load. Table 44-3 and 44.4 shows the DC operating conditions for the high- and low-strength input, output, and I/O pins. .

Table 44-3. 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			0.7VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	LVCMOS Interface			-0.3		0.3VDD	V
ΔV	Hysteresis Voltage			0.1VDD			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 44-4. 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			0.7VDD		VDD+0.3	V
Vil	Low Level Input Voltage						
	LVCMOS Interface			-0.3		0.3VDD	V
ΔV	Hysteresis Voltage			0.1VDD			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 44-5. 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
VDDINT	Internal Voltage	1.15	1.2	1.25	V
Temp	Ambient Temperature	-40	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=2.5V)	-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 44-6. Special Memory DDR I/O PAD DC Electrical Characteristics (Memory Port 1)

Symbol	Parameter	Min	Typ	Max	Unit
VDDm1	Output supply voltage	1.75	1.8/2.5	2.7	V
VDDINT	Internal Voltage	1.15	1.2	1.25	V
Temp	Ambient Temperature	-40	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=2.5V)	-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 44-7. USB DC Electrical Characteristics

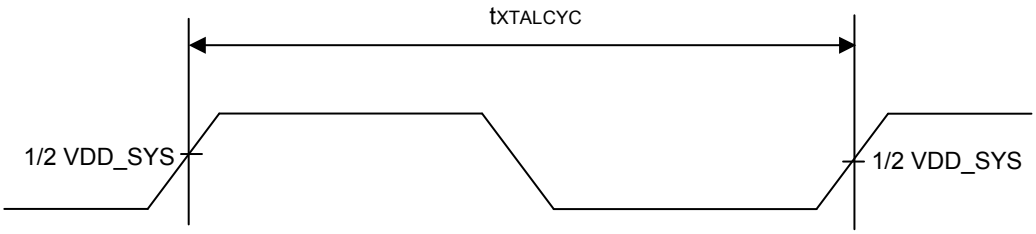
Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	High level input voltage		2.0		V
V_{IL}	Low level input voltage			0.8	V
I_{IH}	High level input current	Vin = 3.3V	-10	10	μA
I_{IL}	Low level input current	Vin = 0.0V	-10	10	μA
V_{OH}	Static Output High	15Kohm to GND	2.8	3.6	V
V_{OL}	Static Output Low	1.5Kohm to 3.6V		0.3	V

Table 44-8. RTC OSC DC Electrical Characteristics

Symbol	Parameter	Min	Type	Max	Unit
VDDrtc	Output supply voltage	1.7	1.8~3.0	3.3	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

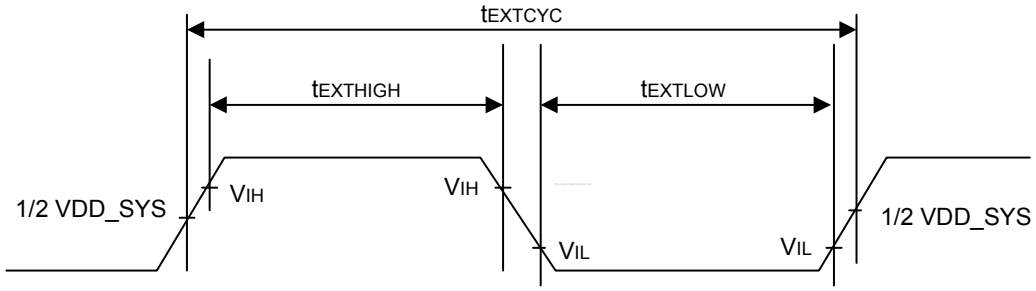
44.4 CLK A.C. ELECTRICAL CHARACTERISTICS

A 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 44-1. XTIpll Clock Timing



NOTE: The clock input from the EXTCLK pin.

Figure 44-2. EXTCLK Clock Input Timing

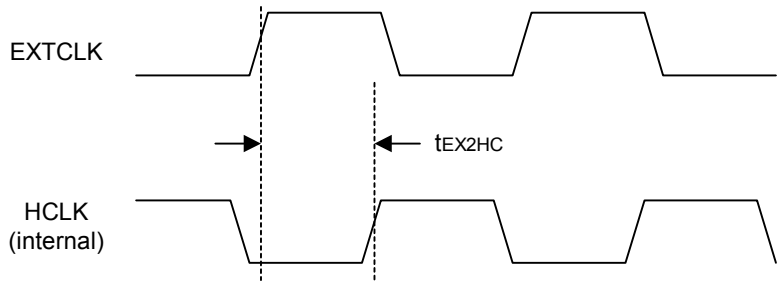


Figure 44-3. EXTCLK/HCLK in case that EXTCLK is used without the PLL

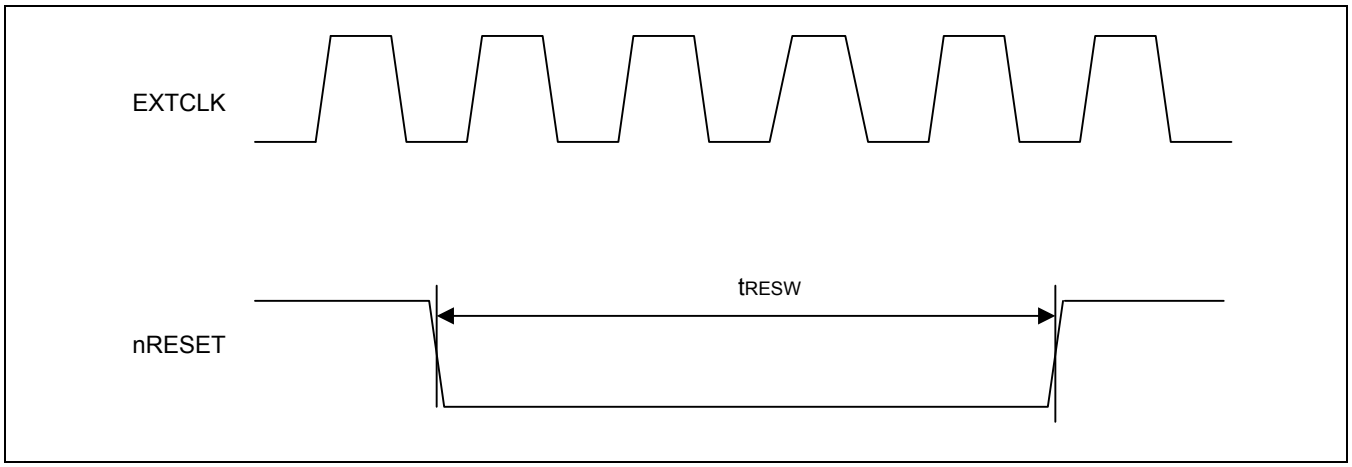


Figure 44-4. Manual Reset Input Timing

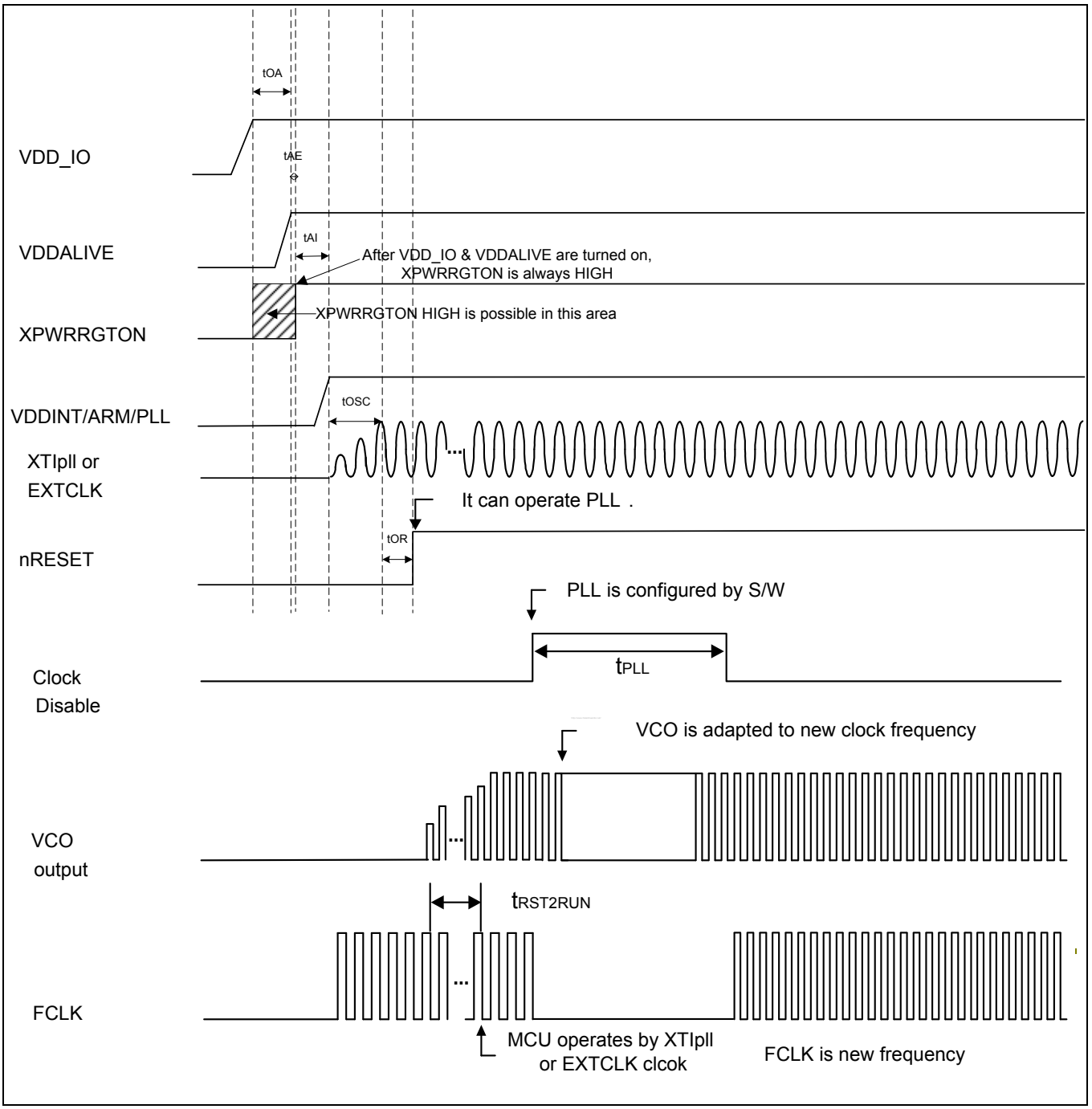


Figure 44-5. Power-On Oscillation Setting Timing

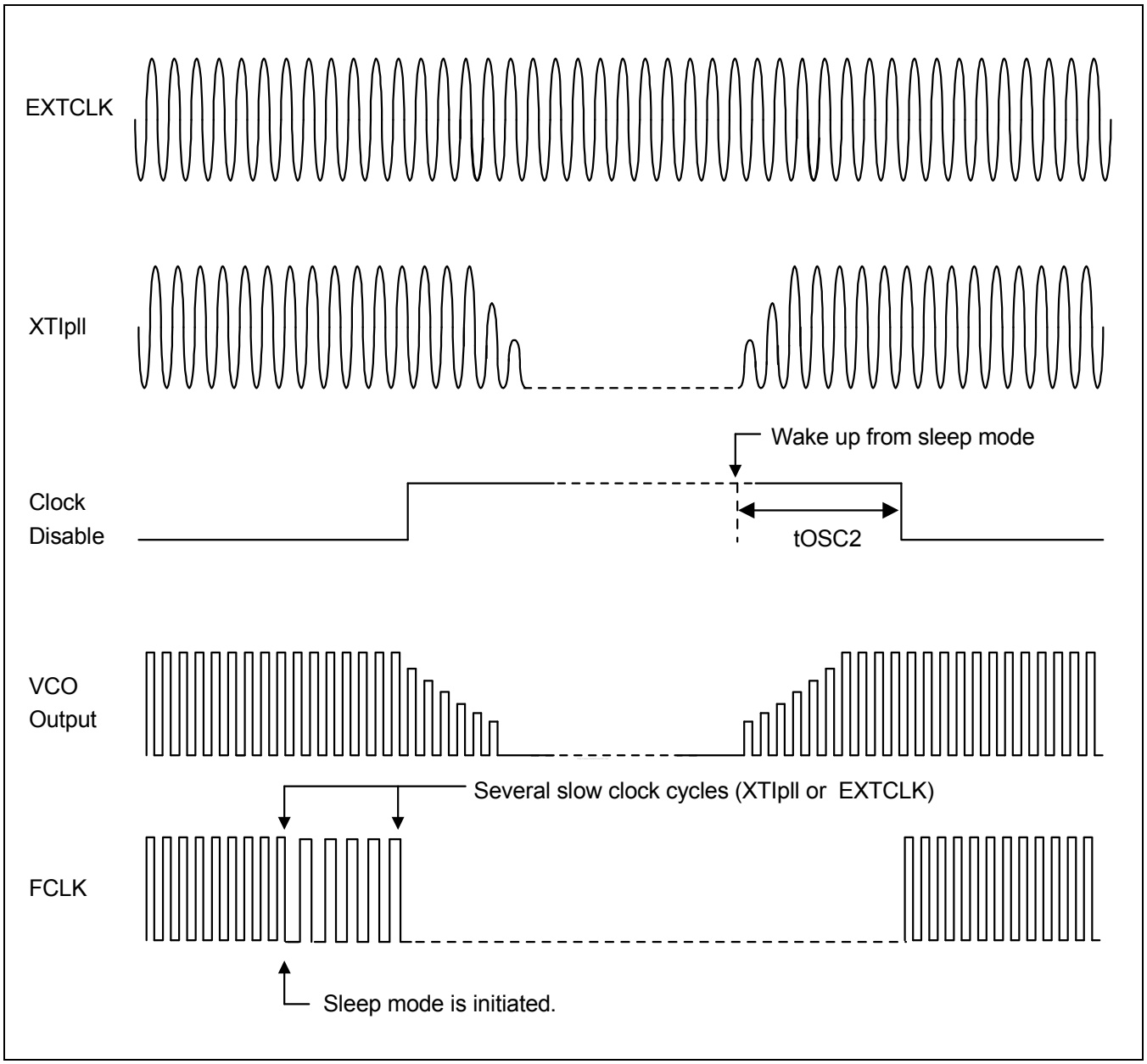


Figure 44-6. Sleep Mode Return Oscillation Setting Timing

Table 44-9. Clock Timing Constants

(VDDINT= 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 VDDINT/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 & 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^{16}	XTIpll or EXTCLK
The interval before CPU runs after nRESET is released.	$t_{RST2RUN}$	5		-	XTIpll or EXTCLK

44.5 ROM/SRAM AC ELECTRICAL CHARACTERISTICS

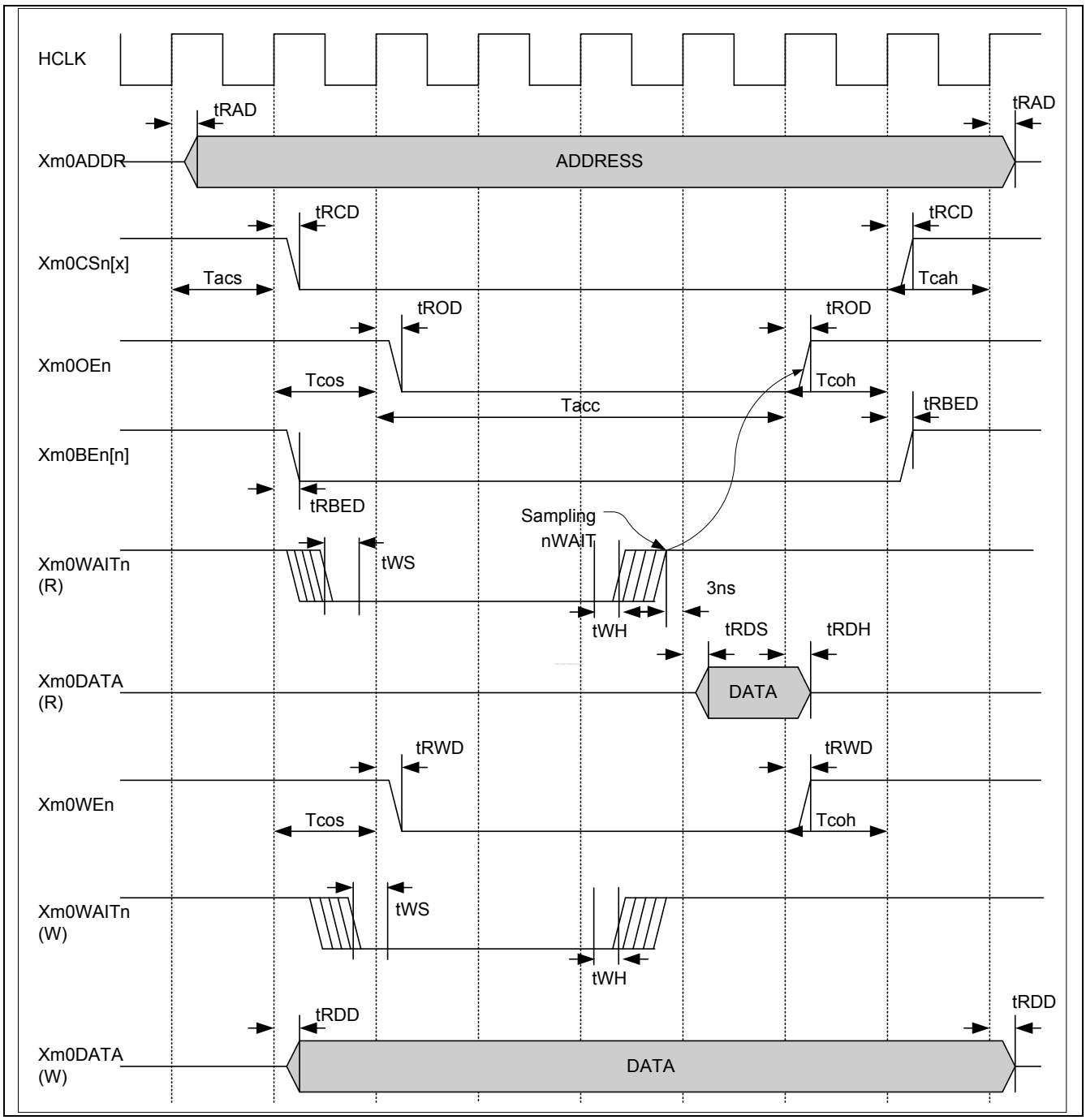


Figure 44-7. ROM/SRAM Timing
(Tacs = 0, Tcos = 0, Tacc = 2, Tcoh = 0, Tcah = 0, PMC = 0, ST = 0, DW = 16-bit)

Table 44-10. ROM/SRAM Bus Timing Constants

(VDDINT = 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}	1.2823	7.8220	ns
ROM/SRAM Chip Select 0 Delay	t_{RCD}	1.9564	6.6403	ns
ROM/SRAM Chip Select 1 Delay	t_{RCD}	1.8722	6.6967	ns
ROM/SRAM Chip Select 2 Delay	t_{RCD}	1.8775	6.1614	ns
ROM/SRAM Chip Select 3 Delay	t_{RCD}	1.7831	6.0382	ns
ROM/SRAM Chip Select 4 Delay	t_{RCD}	1.7790	6.1450	ns
ROM/SRAM Chip Select 5 Delay	t_{RCD}	1.8434	6.4550	ns
ROM/SRAM nOE(Output Enable) Delay	t_{ROD}	1.8143	6.4113	ns
ROM/SRAM nWE(Write Enable) Delay	t_{RWD}	1.7700	6.2336	ns
ROM/SRAM Byte Enable Delay	t_{RBED}	1.8072	6.5093	ns
ROM/SRAM Output Data Delay	t_{RDD}	1.1940	8.2706	ns
ROM/SRAM Read Data Setup Time	t_{RDS}	2.0000	-	ns
ROM/SRAM Write Data Hold Time	t_{RDH}	1.0000	-	ns

44.6 ONENAND AC ELECTRICAL CHARACTERISTICS

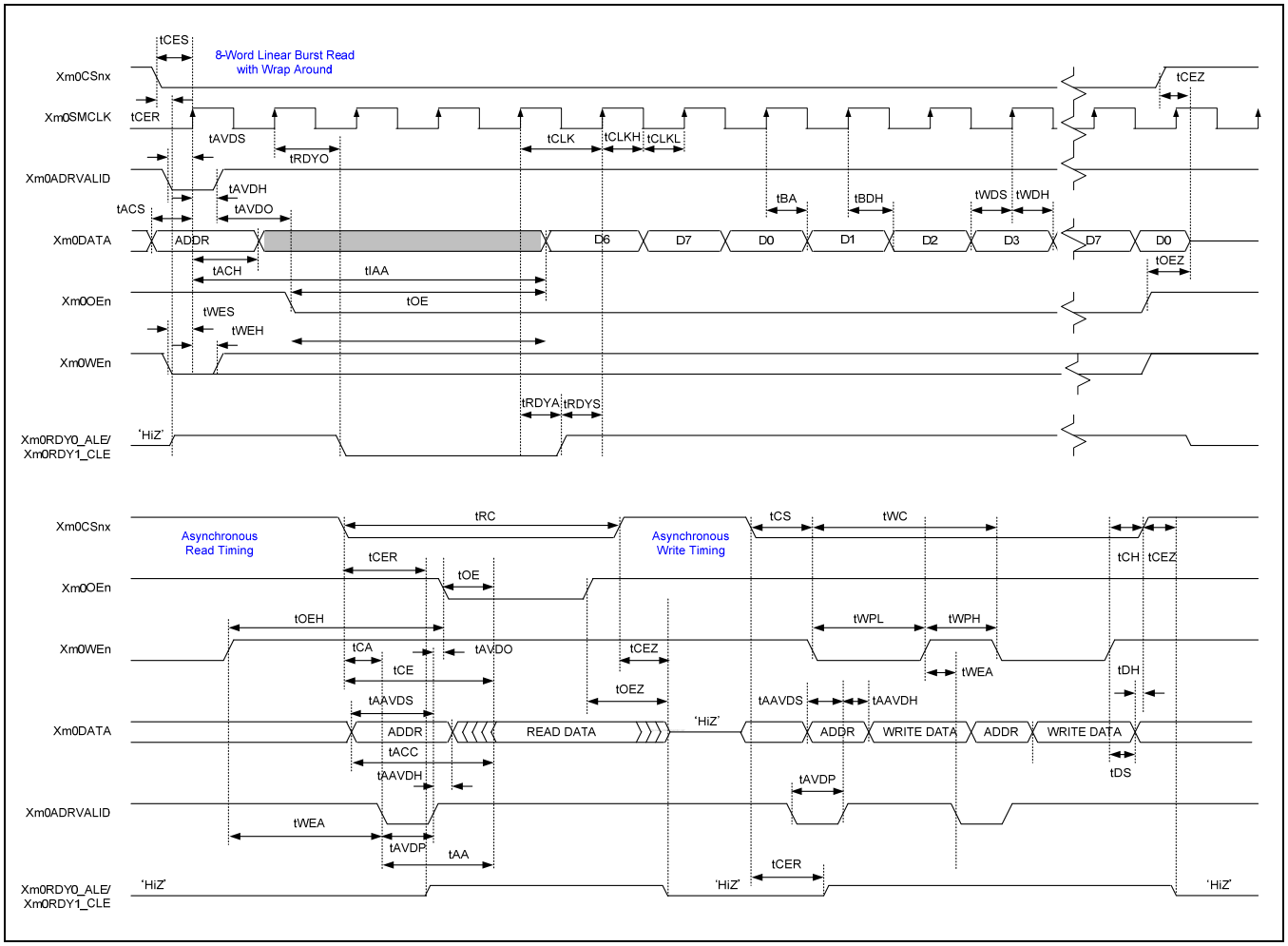


Figure 44-8. OneNand Flash Timing

Table 44-11. OneNAND Bus Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDm0 = 1.7V – 1.9V)

Parameter	Symbol	Min	Max	Unit
OneNAND SMCLK cycle	t_{CLK}	15		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}	6		ns
OneNAND Initial Access time	t_{IAA}		70	ns
OneNAND Burst Access time valid SMCLK to Output delay	t_{BA}		11.5	ns
OneNAND Data Hold time from next clock cycle	t_{BDH}	3.5		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}	5		ns
OneNAND Address Hold time to SMCLK	t_{ACH}	6		ns
OneNAND ADRVALID Setup time to SMCLK	t_{AVDS}	5		ns
OneNAND ADRVALID Hold time to SMCLK	t_{AVDH}	6		ns
OneNAND Write Data Setup time to SMCLK	t_{WDS}	5		ns
OneNAND Write Data Hold time to SMCLK	t_{WDH}	2		ns
OneNAND WEn Setup time to SMCLK	t_{WES}	5		ns
OneNAND WEn Hold time to SMCLK	t_{WEH}	6		ns
OneNAND ADRVALID high to OEn low	t_{AVDO}	0		ns
OneNAND SMCLK to RDY valid	t_{RDYO}		11.5	ns
OneNAND SMCLK to RDY Setup time	t_{RDYA}		11.5	ns
OneNAND RDY Setup time to SMCLK	t_{RDYS}		11.5	ns
OneNAND CSn low to RDY valid	t_{CER}		15	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
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

44.7 NFCON AC ELECTRICAL CHARACTERISTICS

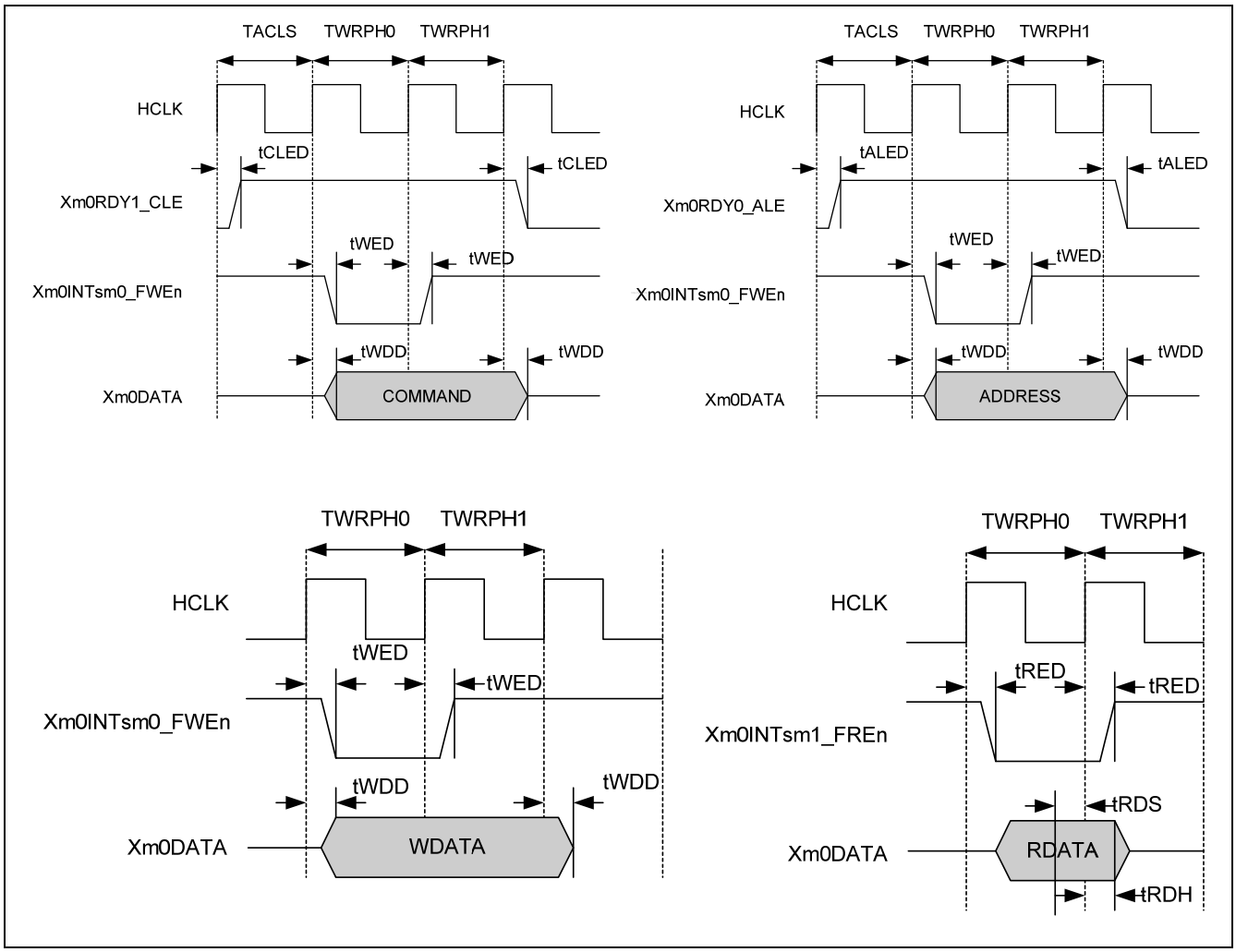


Figure 44-9. Nand Flash Timing

Table 44-12. NFCON Bus Timing Constants

(VDDINT= 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}	-	6.85	ns
NFCON CLE delay	t_{CLED}	-	7.72	ns
NFCON ALE delay	t_{ALED}	-	7.73	ns
NFCON Write Enable delay	t_{WED}	-	8.61	ns
NFCON Read Enable delay	t_{RED}	-	7.91	ns
NFCON Write Data delay	t_{WDD}	-	8.42	ns
NFCON Read Data Setup requirement time	t_{RDS}	1.00	-	ns
NFCON Read Data Hold requirement time	t_{RDH}	0.20	-	ns

44.8 SDRAM AC ELECTRICAL CHARACTERISTICS

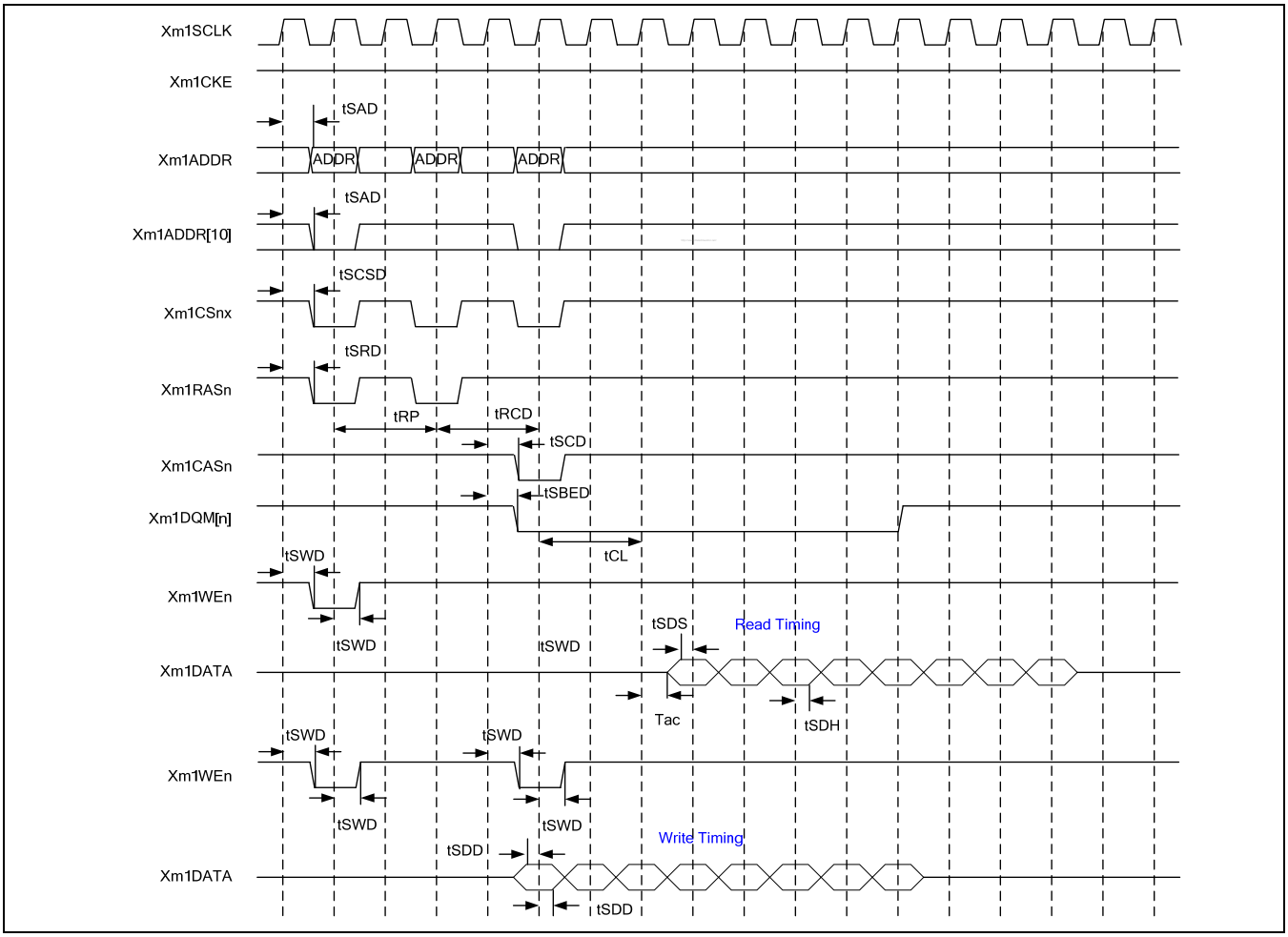


Figure 44-10. SDR SDRAM READ / WRITE Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit)

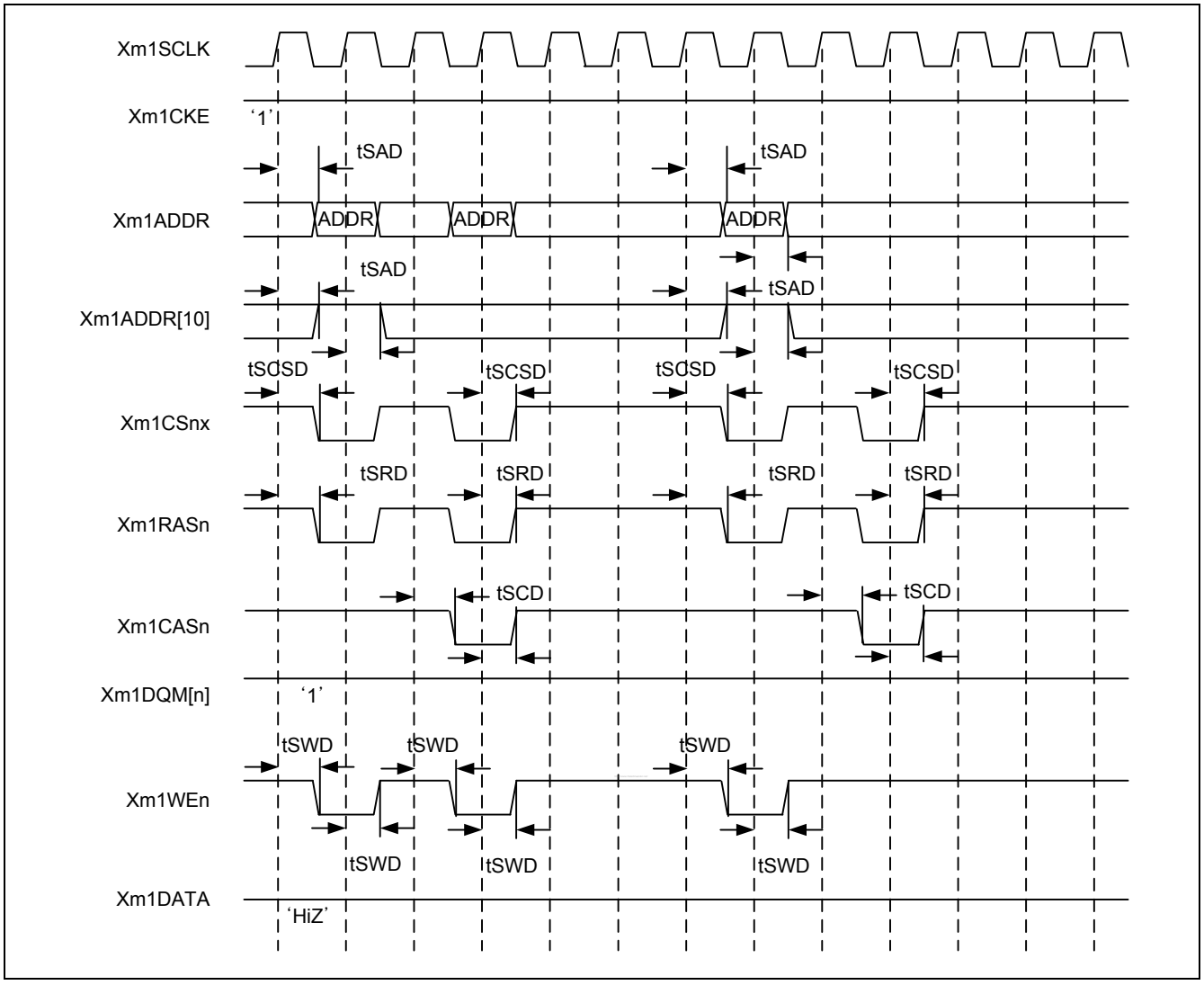


Figure 44-11. SDRAM MRS Timing and Auto Refresh Timing (Trp = 2, Trc = 4)

NOTE: Before executing auto/self refreshing command, all banks must be in idle state.

Table 44-13. Memory Port 1 Interface Timing Constants (SDR SDRAM)

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDm1 = 1.75V – 2.70V)

Parameter	Symbol	Min	Max	Unit
SDRAM Address Delay	t_{SAD}	1.3640	3.8910	ns
SDRAM Chip Select Delay	t_{SCSD}	1.2760	3.7090	ns
SDRAM Row active Delay	t_{SRD}	1.2460	3.6590	ns
SDRAM Column active Delay	t_{SCD}	1.2860	3.6890	ns
SDRAM Byte Enable Delay	t_{SBED}	2.3360	5.0760	ns
SDRAM Write enable Delay	t_{SWD}	1.2290	3.6100	ns
SDRAM read Data Setup time	t_{SDS}	3.0000	-	ns
SDRAM read Data Hold time	t_{SDH}	1.5000	-	ns
SDRAM output Data Delay	t_{SDD}	2.2380	5.1940	ns
SDRAM Clock Enable Delay	t_{CKED}	1.2070	3.6990	ns
SDRAM Access time from Clock	t_{AC}	-	6.0000	ns

Load Capacitance	
Xm1*	< 15pF

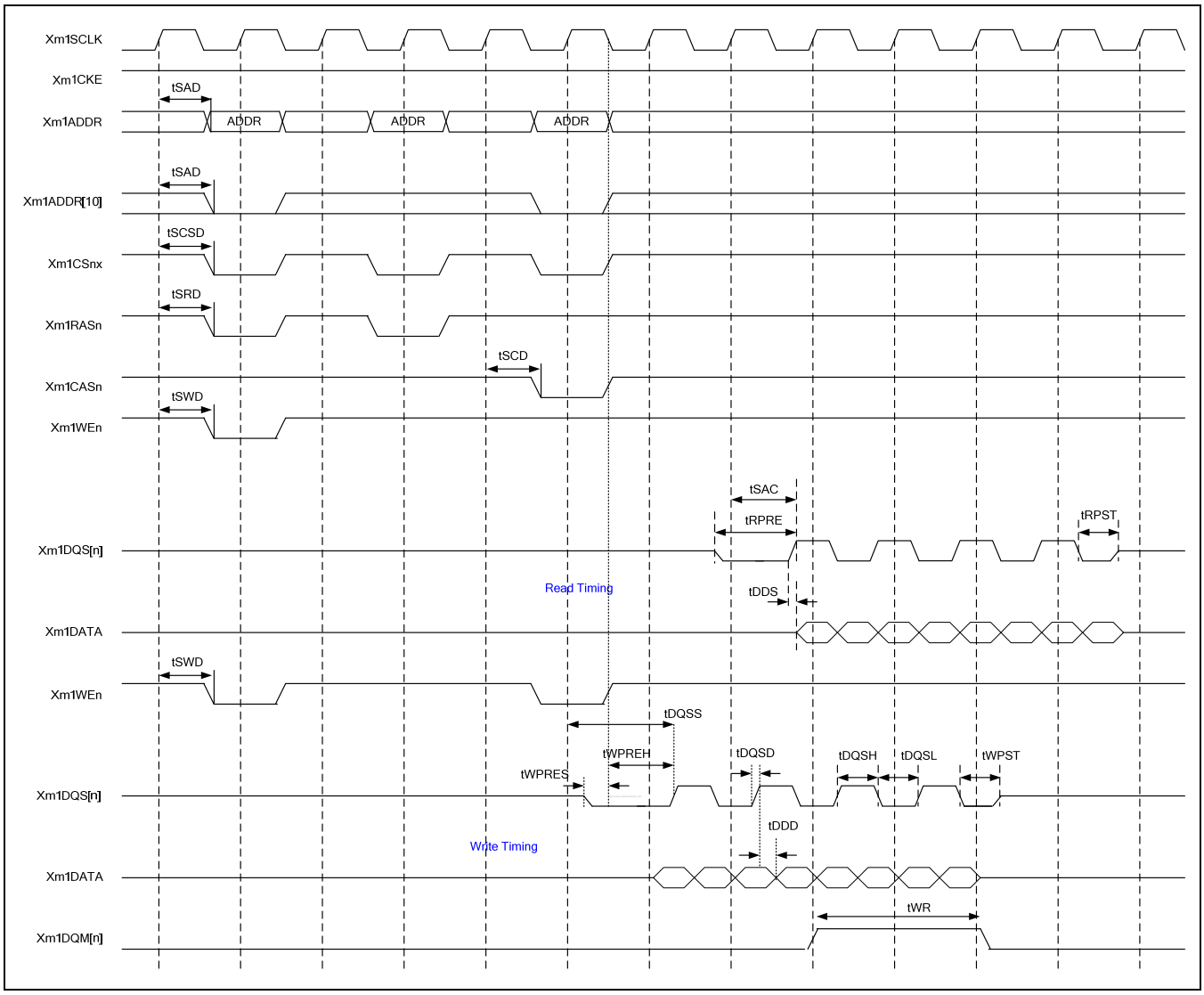


Figure 44-12. DDR SDRAM READ / WRITE Timing (Trp = 2, Trcd = 2, Tcl = 2, DW = 16-bit)

Table 44-14. Memory Port 1 Interface Timing Constants (DDR SDRAM)

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDm1 = 1.75V – 2.7V)

Parameter	Symbol	Min	Max	Unit
DDR SDRAM Address Delay	tSAD	1.3640	3.8910	ns
DDR SDRAM Chip Select Delay	tSCSD	1.2760	3.7090	ns
DDR SDRAM Row active Delay	tSRD	1.2460	3.6590	ns
DDR SDRAM Column active Delay	tSCD	1.2860	3.6890	ns
DDR SDRAM Byte Enable Delay	tSBED	2.3360	5.0760	ns
DDR SDRAM Write enable Delay	tSWD	1.2290	3.6100	ns
DDR SDRAM Output data access time from CK	tSAC	2.0000	5.5000	ns
DDR SDRAM Row Precharge time	tRP	22.5000	-	ns
DDR SDRAM RAS to CAS delay	tRCD	22.5000	-	ns
DDR SDRAM Write recovery time	tWR	12.0000	-	ns
DDR SDRAM Clock low level width	tCL	3.4751	3.6220	ns
DDR SDRAM Read Preamble	tRPRE	6.7500	8.2500	ns
DDR SDRAM Read Postamble	tRPST	3.0000	4.5000	ns
DDR SDRAM Write Postamble time	tWPST	3.0000	4.5000	ns
DDR SDRAM Clock to valid DQS-In	tDQSS	9.1423	9.3750	ns
DDR SDRAM DQS-In Setup time	tWPRES	1.3000	-	ns
DDR SDRAM DQS-In Hold time	tWPREH	1.3000	-	ns
DDR SDRAM DQS-In high level width	tDQSH	3.0000	4.5000	ns
DDR SDRAM DQS-In low level width	tDQSL	3.0000	4.5000	ns
DDR SDRAM read Data Setup time	tDDS	-	0.5000	ns

Load Capacitance	
Xm1*	< 15pF

44.9 LCD CONTROLLER AC ELECTRICAL CHARACTERISTICS

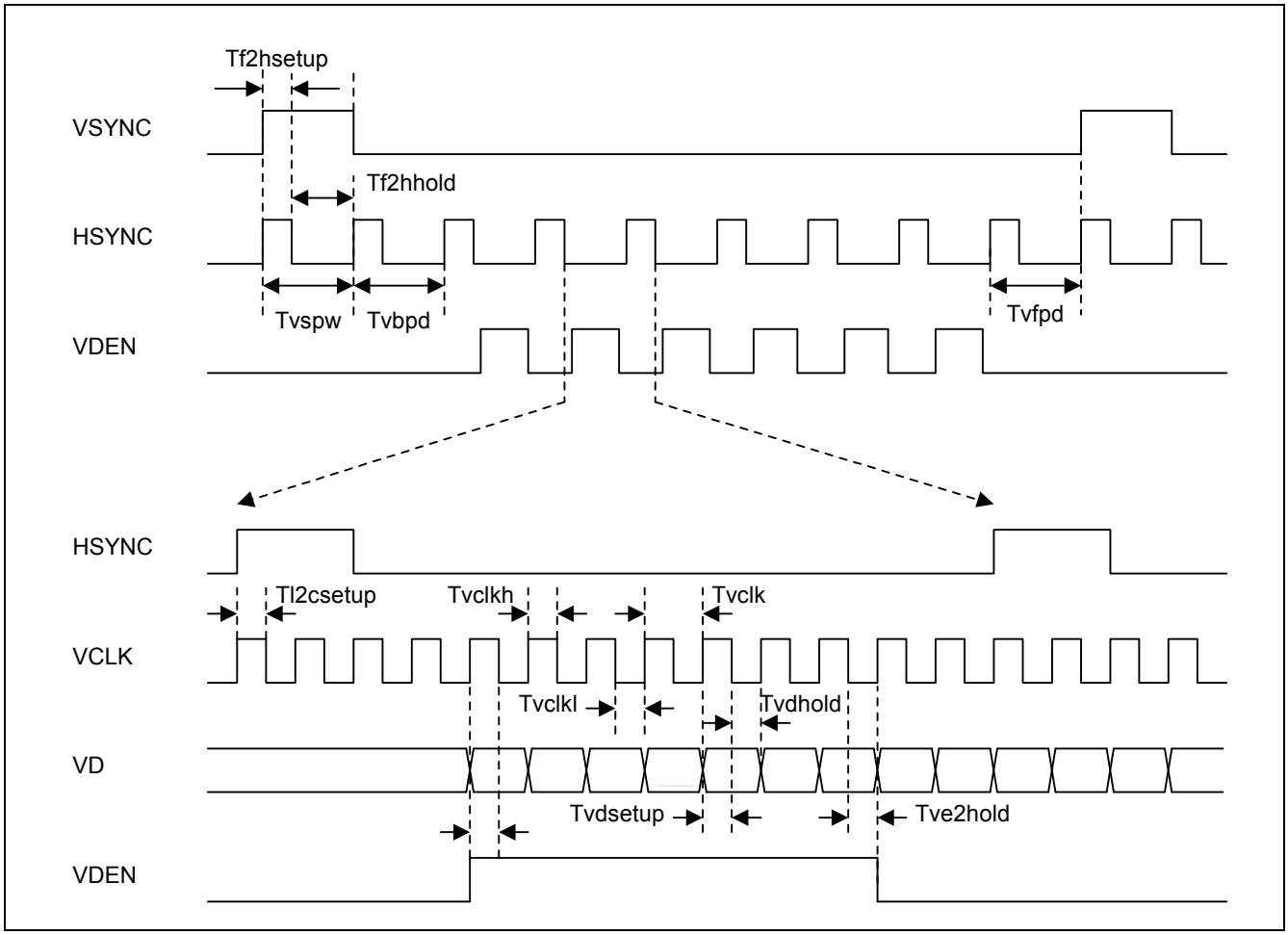


Figure 44-13. TFT LCD Controller Timing

Table 44-15. TFT LCD Controller Module Signal Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDlcd = 1.7V - 3.6V)

Parameter	Symbol	Min	Type	Max	Units
VCLK pulse width	Tvclk	18	200	–	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 set up 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

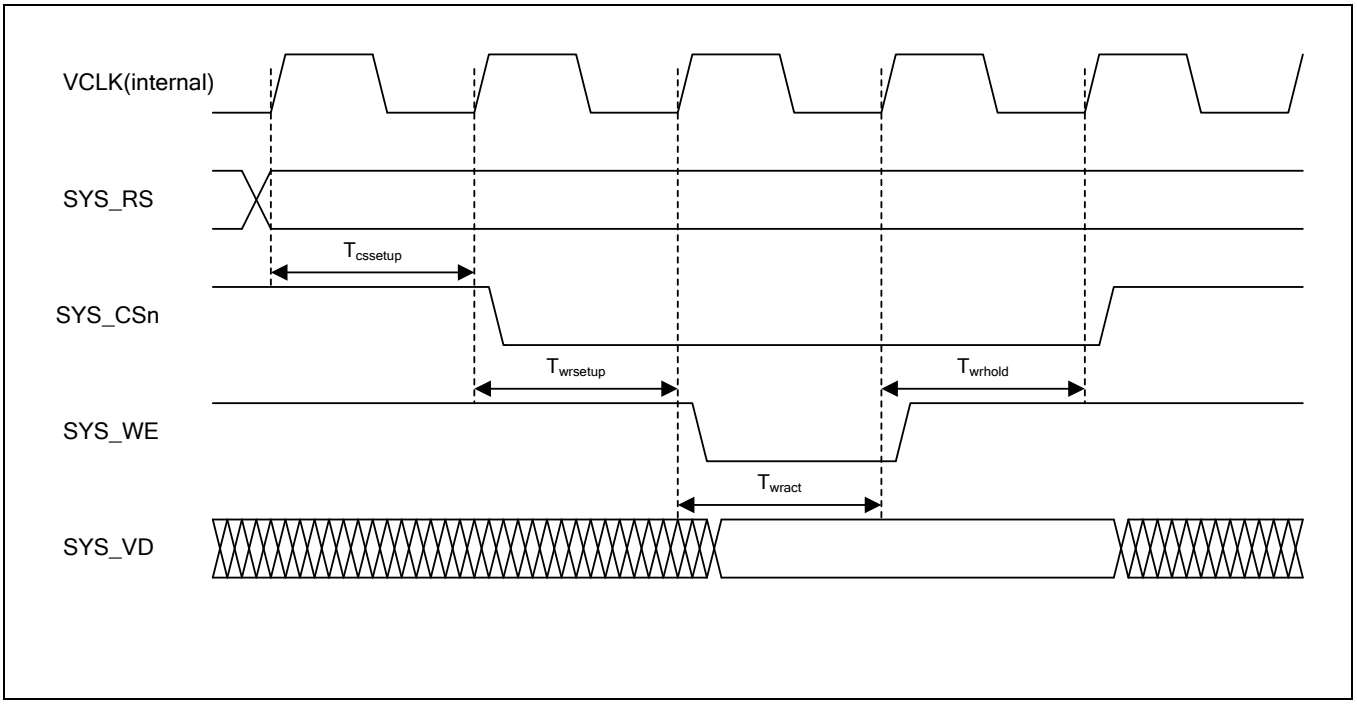


Figure 44-14. LCD I80 InterfaceTiming

Table 44-16. LCD I80 Interface Signal Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDlcd = 1.7V - 3.6V)

Parameter	Symbol	Min	Type	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

44.10 MODEMIF AC ELECTRICAL CHARACTERISTICS

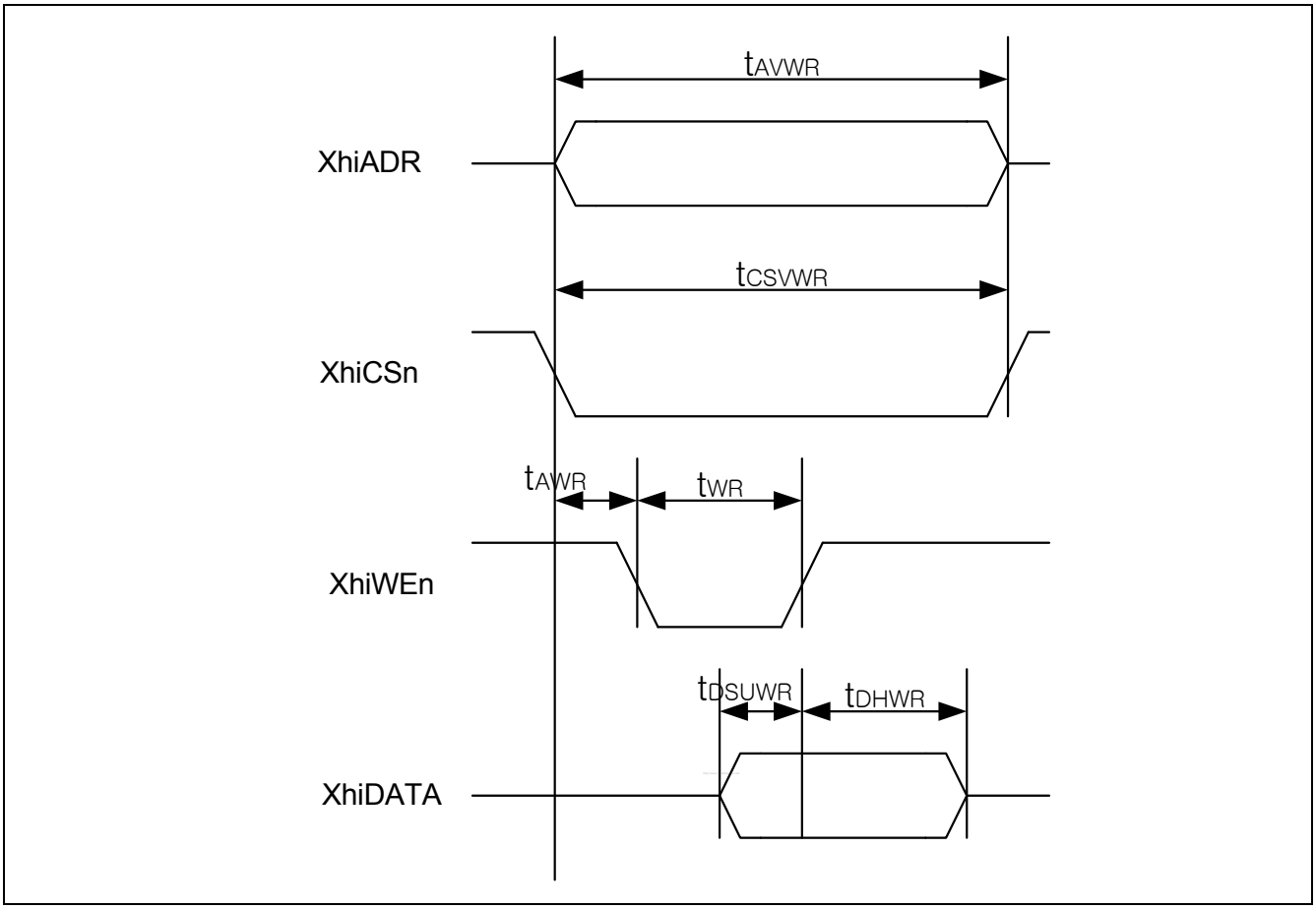


Figure 44-15. Modem interface write timing diagram

Table 44-17. Modem interface write timing

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDhi = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVWR}	Address valid to address invalid	20 ns	-	
t_{CSVWR}	Chip select active	20 ns	-	
t_{AWR}	Address valid to write active	5 ns	-	
t_{WR}	Write active	10 ns	-	
t_{DSUWR}	Write data setup	10 ns	-	
t_{DHWR}	Write data hold	5 ns	-	

NOTE: Output load is 30pF at room temperature (25°C)

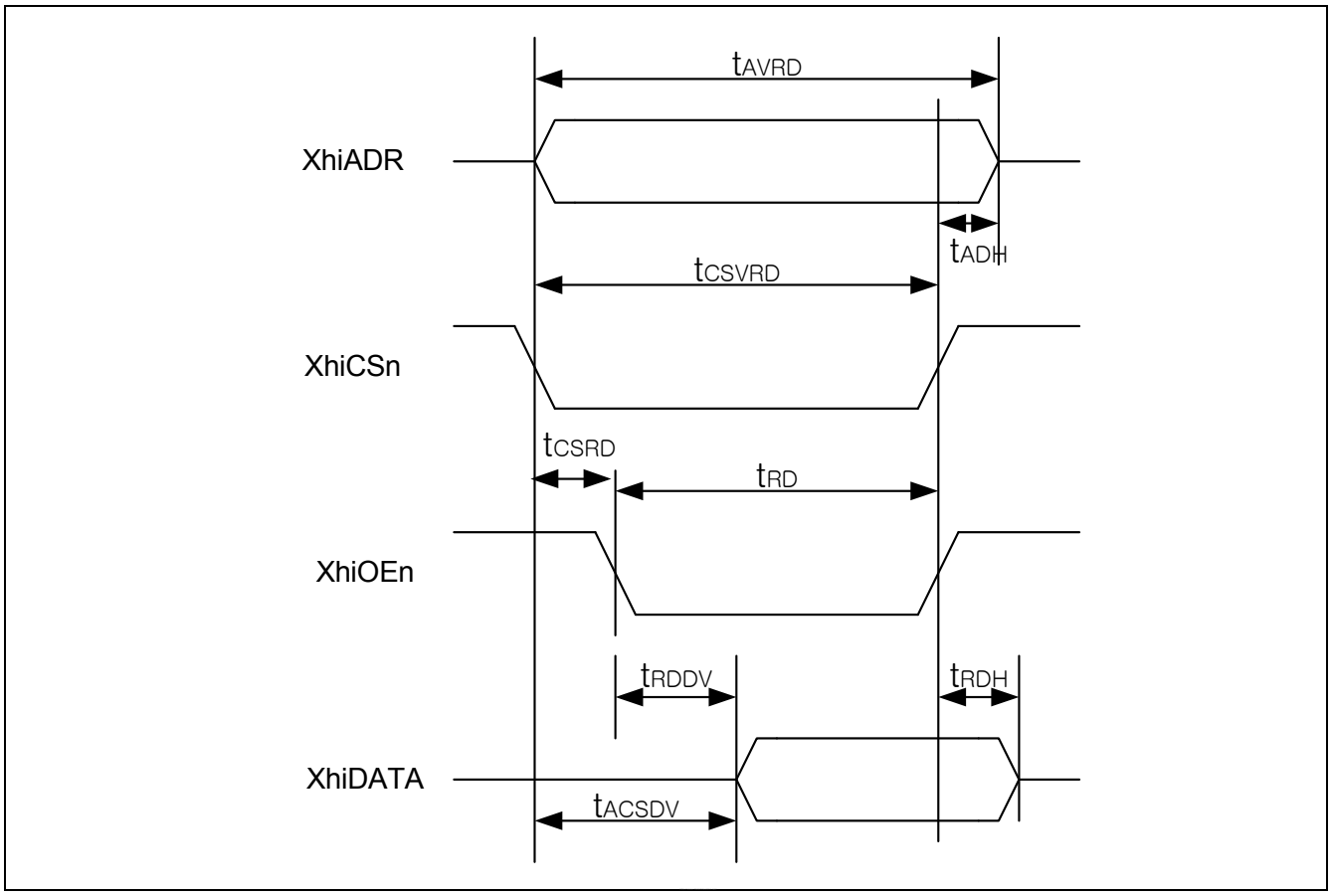


Figure 44-16. Modem interface read timing diagram

Table 44-18. Modem interface read timing

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDhi = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Description	Min (ns)	Max (ns)	Notes
t_{AVRD}	Address valid to address invalid	40 ns	-	
t_{ADH}	Address hold	0 ns		
t_{CSVRD}	Chip select active	40 ns	-	
t_{CSRD}	Chip select active to Read active	10 ns	-	
t_{RD}	Read active	20 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°C)

44.11 LCD BYPASS AC ELECTRICAL CHARACTERISTICS

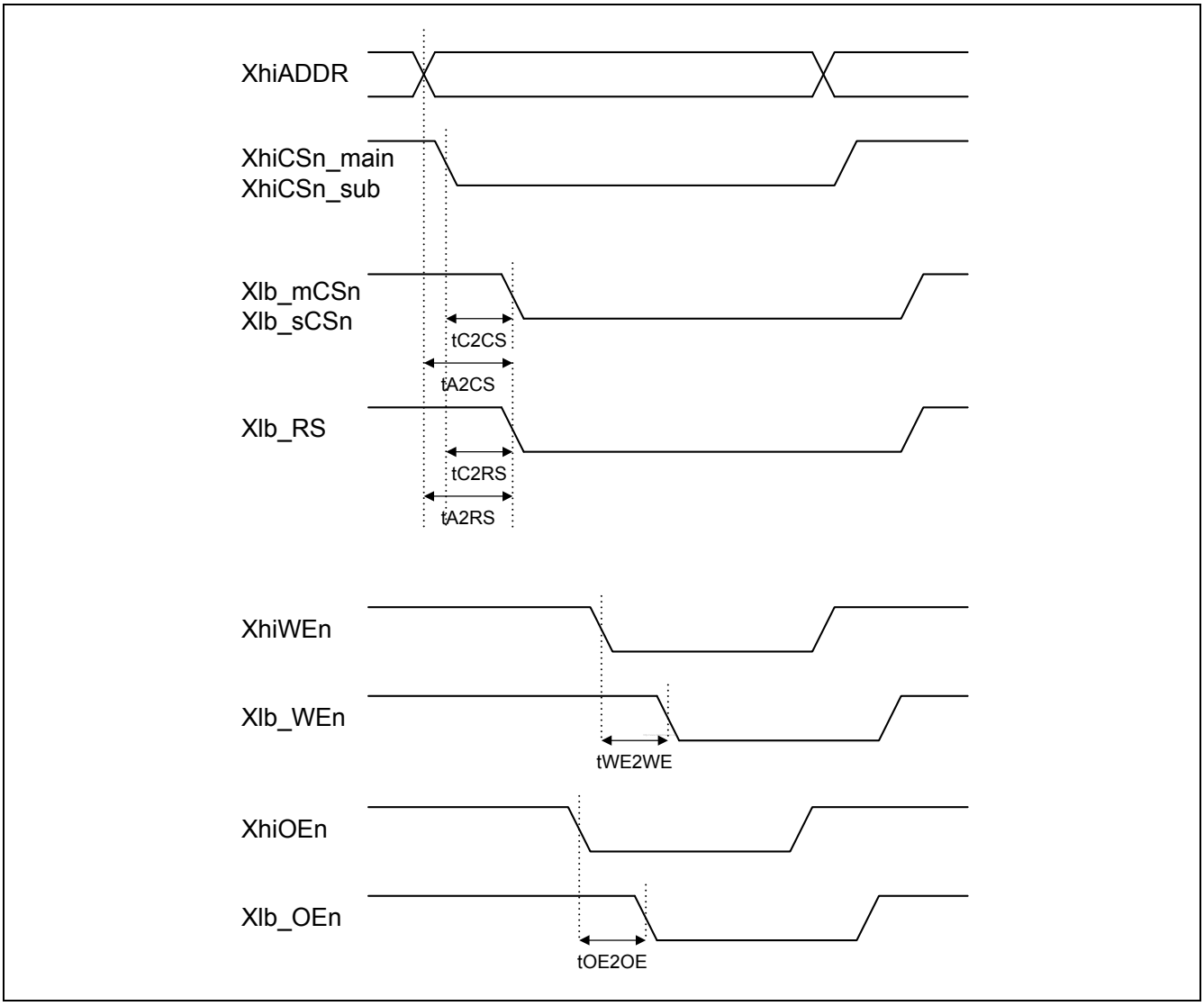


Figure 44-17. LCD Bypass Control signal timing diagram

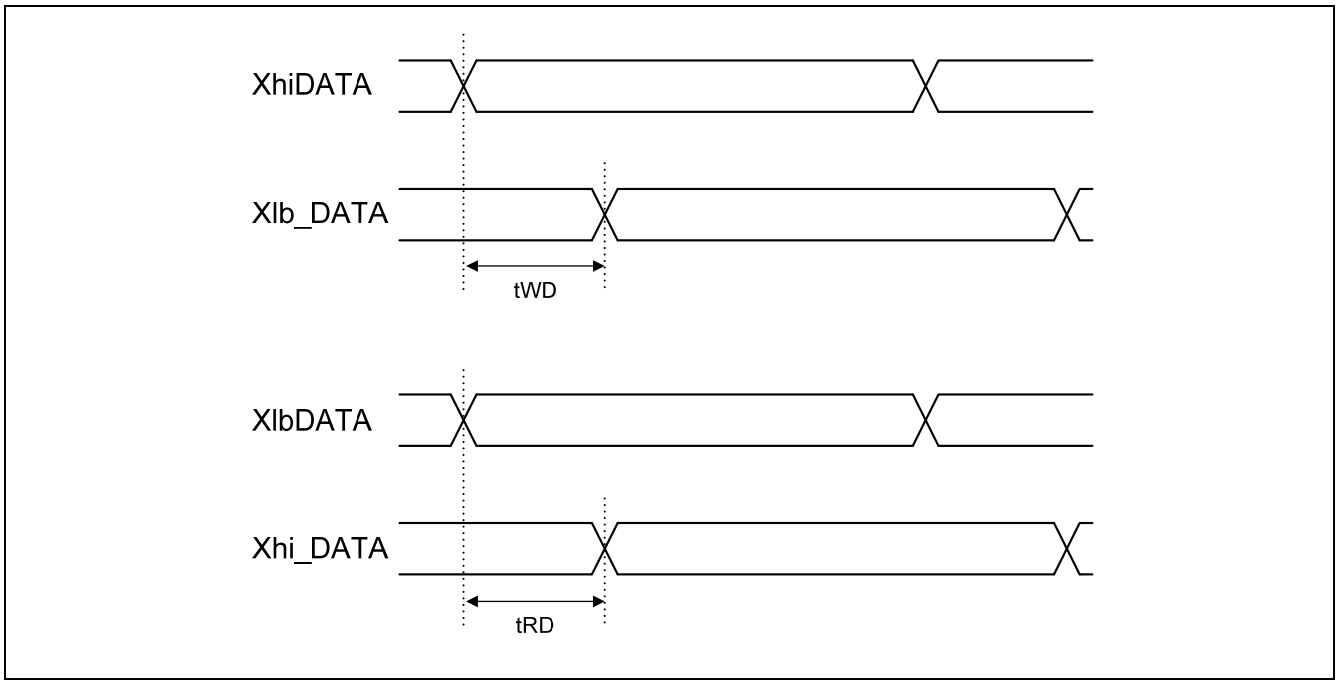


Figure 44-18. LCD Bypass Data for write and read timing diagram

Table 44-19. LCD Bypass timing Timing

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDhi = 1.7V - 3.6V)

Parameter	Description	Min (ns)	Max (ns)	Unit
tA2CS	Host Address to LCD Chip Select	5	26	ns
tC2CS	Host Chip Select to LCD Chip Select	5	24	ns
tA2RS	Host Address to LCD Read Select	6	25	ns
tWE2WE	Host Write Enable to LCD Write Enable	3	12	ns
tOE2OE	Host Read Enable to LCD Read Enable	5	18	ns
tWD	Host Write Data to LCD Data	5	23	ns
tRD	Host Read Data from LCD Data	4	22	ns

NOTE: Output load is 30pF at room temperature (25°C)

44.12 CAMERA INTERFACE AC ELECTRICAL CHARACTERISTICS

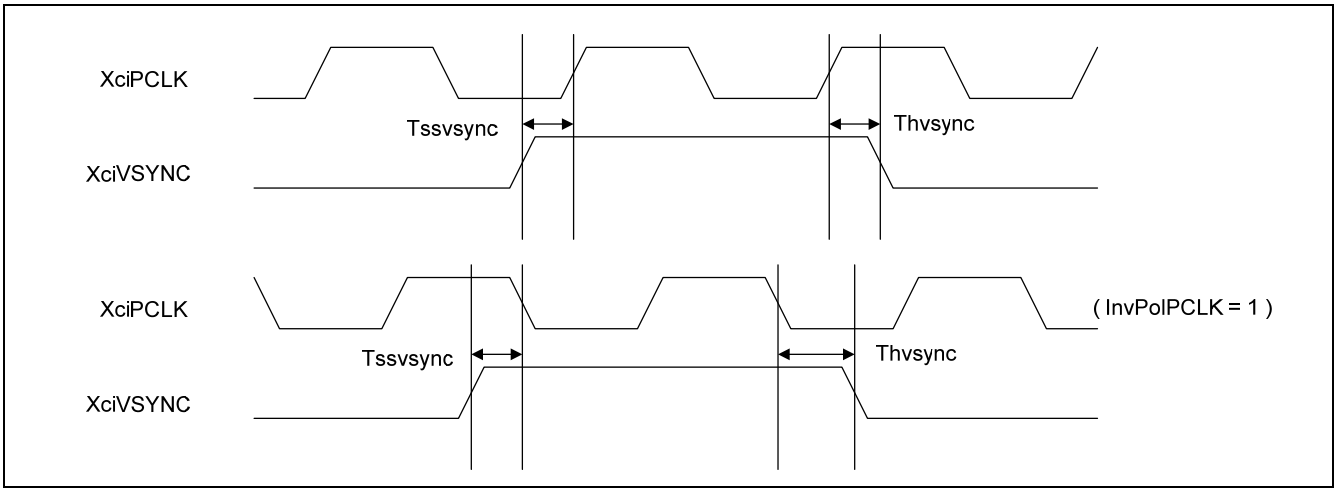


Figure 44-19. Camera Interface VSYNC Timing

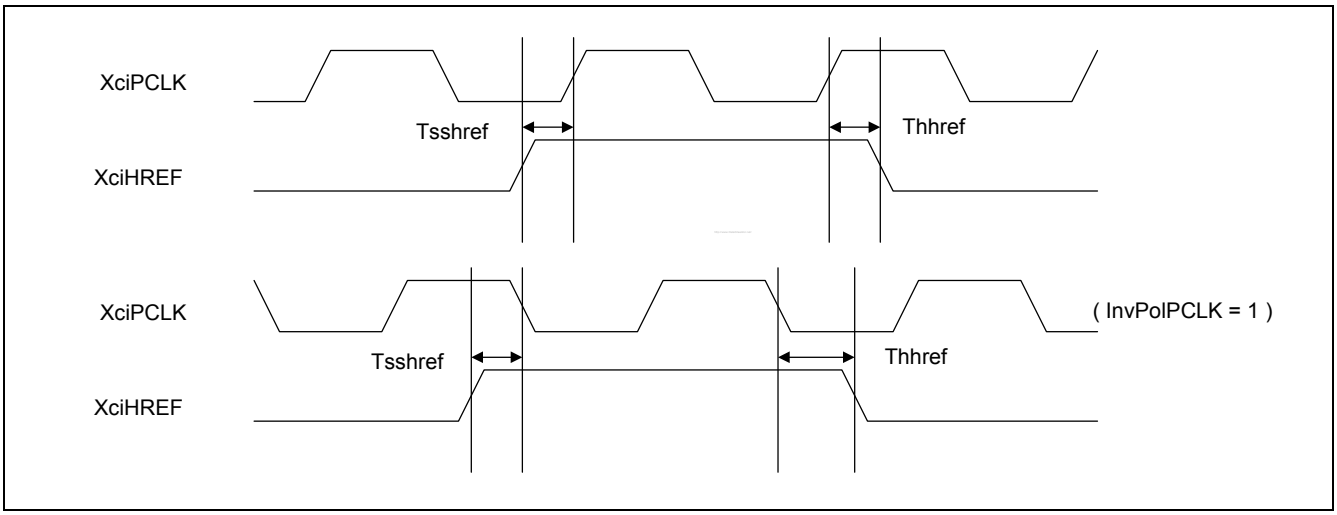


Figure 44-20. Camera Interface HREF Timing

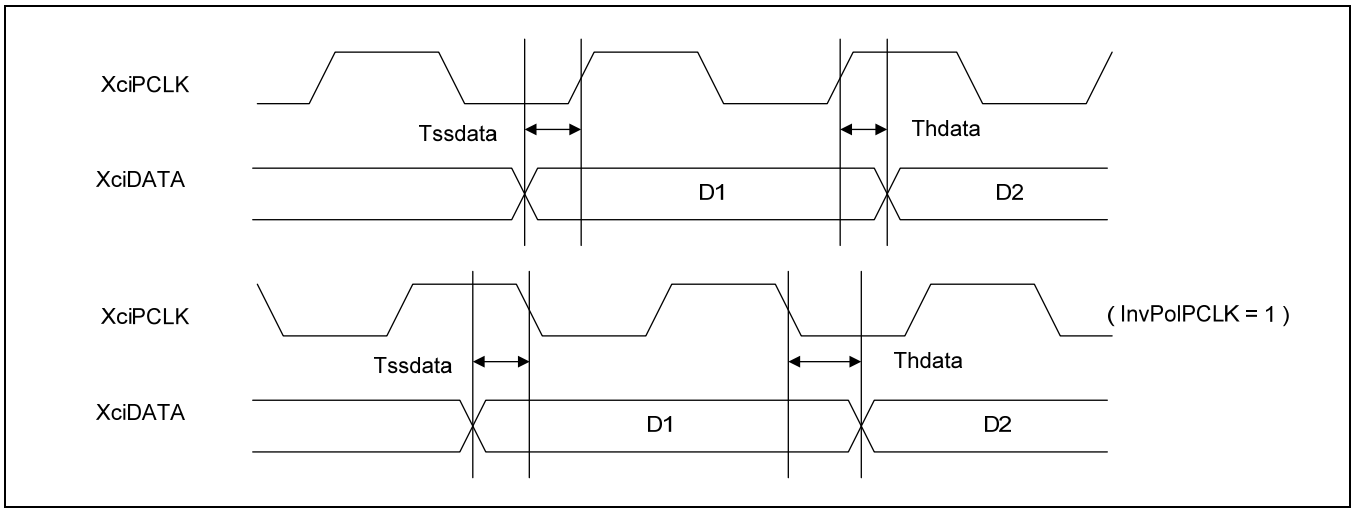


Figure 44-21. Camera Interface Data Timing

Table 44-20. Camera Controller Module Signal Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDext = 1.7V - 3.6V)

Parameter	Symbol	Min	Type	Max	Units
XciVSYNC input Setup time	$T_{ssvsync}$	2.4	-	-	ns
XciVSYNC input Hold time	T_{hvsync}	2	-	-	ns
XciHREF input Setup time	T_{sshref}	2.5	-	-	ns
XciHREF input Hold time	T_{hhref}	2	-	-	ns
XciDATA input Setup time	T_{ssdata}	1.8	-	-	ns
XciDATA input Hold time	T_{hdata}	3	-	-	ns

44.13 SDMMC AC ELECTRICAL CHARACTERISTICS

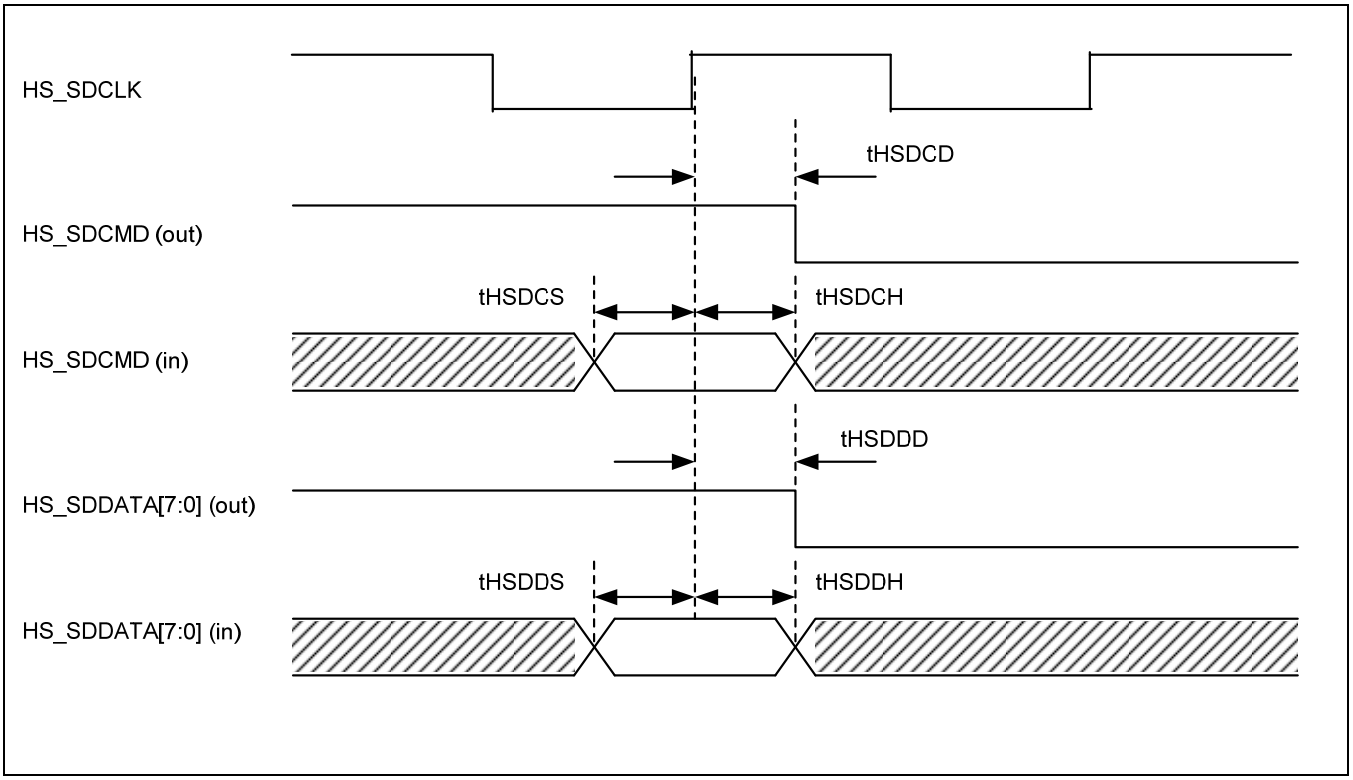


Figure 44-22. High Speed SDMMC Interface Timing

Table 44-21. High Speed SDMMC Interface Transmit/Receive Timing Constants

(VDDINT= 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 is increased to 14ns (this setting should be used when low speed mode).

44.14 SPI AC ELECTRICAL CHARACTERISTICS

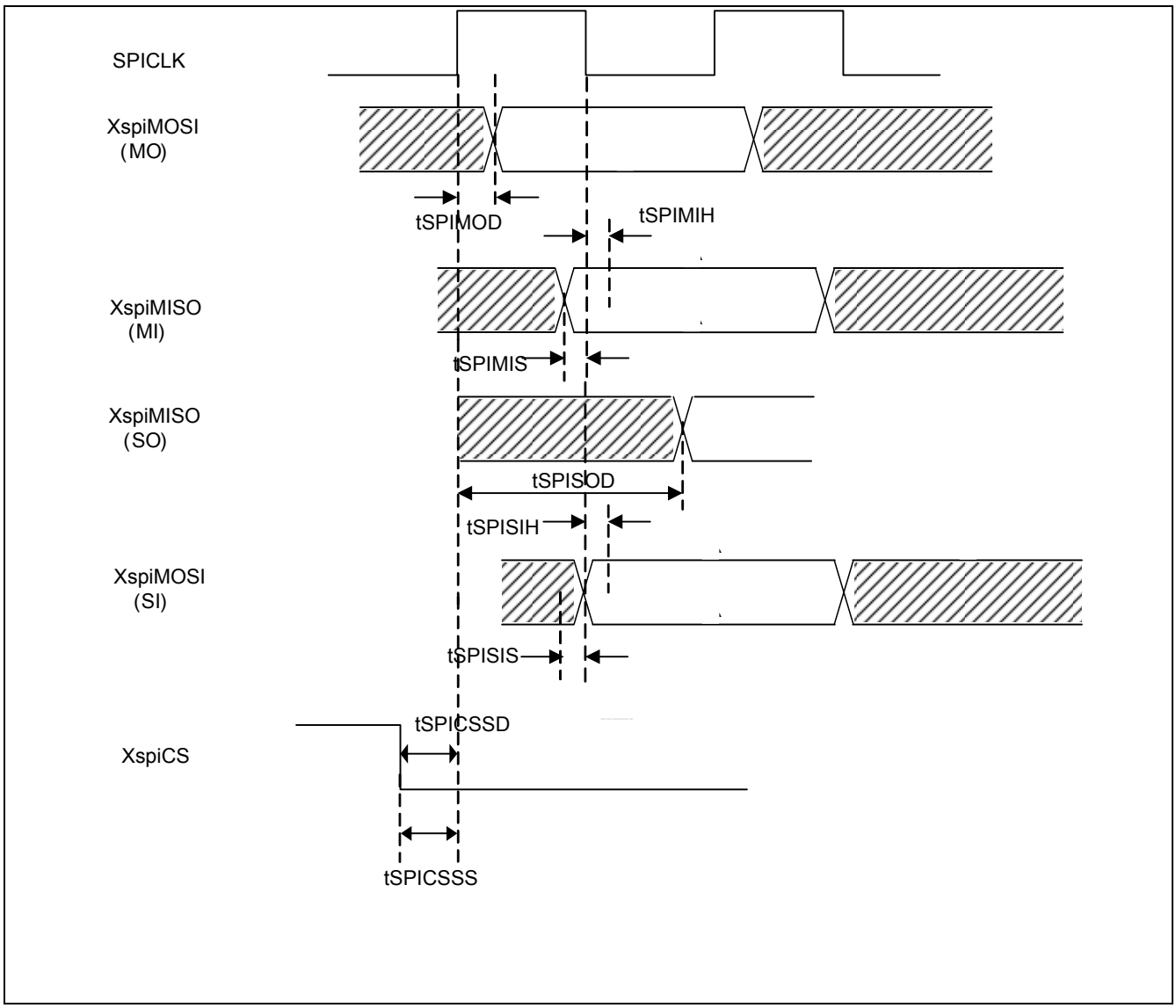


Figure 44-23. SPI Interface Timing (CPHA = 0, CPOL = 1)

Table 44-22. SPI Interface Transmit/Receive Timing Constants

(VDDINT= 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	2	-	-	ns
	SPI MISO Master Input Setup time(Feedback Delay- 2nS)		1	-	-	ns
	SPI MISO Master Input Setup time(Feedback Delay- 4nS)		1	-	-	ns
	SPI MISO Master Input Setup time(Feedback Delay- 6nS)		0	-	-	ns
	SPI MISO Master Input Hold time(Feedback Delay- 0nS)	tSPIMIH	3	-	-	ns
	SPI MISO Master Input Hold time(Feedback Delay- 2nS)		6	-	-	ns
	SPI MISO Master Master Input Hold time(Feedback Delay- 4nS)		9	-	-	ns
	SPI MISO Master Input Hold time(Feedback Delay- 6nS)		11	-	-	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	-	-	19	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		1	-	-	ns

	time(Feedback Delay- 4nS)					
	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	-	-	9	ns
	SPI nSS Master Output Delay time	tSPICSSD	-	-	23	ns
	SPI nSS Slave Input Setup time	tSPICSSS	-	-	20	ns

NOTE: SPICLKout = 50MHz
VDDEXT connects to SPI0
VDDmmc connects to SPI1

44.15 MIPI-HSI AC ELECTRICAL CHARACTERISTICS

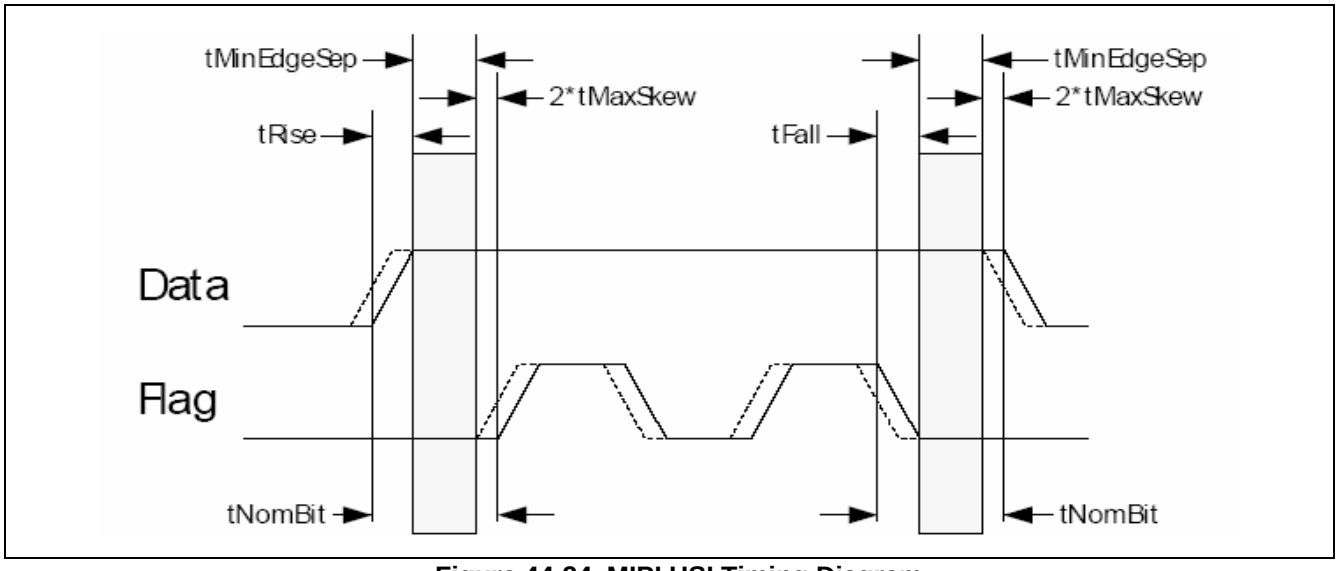


Figure 44-24. MIPI HSI Timing Diagram

Table 44-23. MIPI HSI Interface Transmit/Receive Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDhi = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Description	1 Mbit/s	100 Mbit/s
TNomBit	Nominal bit time	1000 ns	10 ns
TMinEdgeSep	Minimum allowed separation of DATA and FLAG signal transitions	500 ns	5 ns
TMaxSkew	Maximum allowed time for combined skew and jitter	249 ns	1.5 ns
tRise and tFall	Minimum allowed signal rise and fall time	2 ns	2 ns

44.16 IIS AC ELECTRICAL CHARACTERISTICS

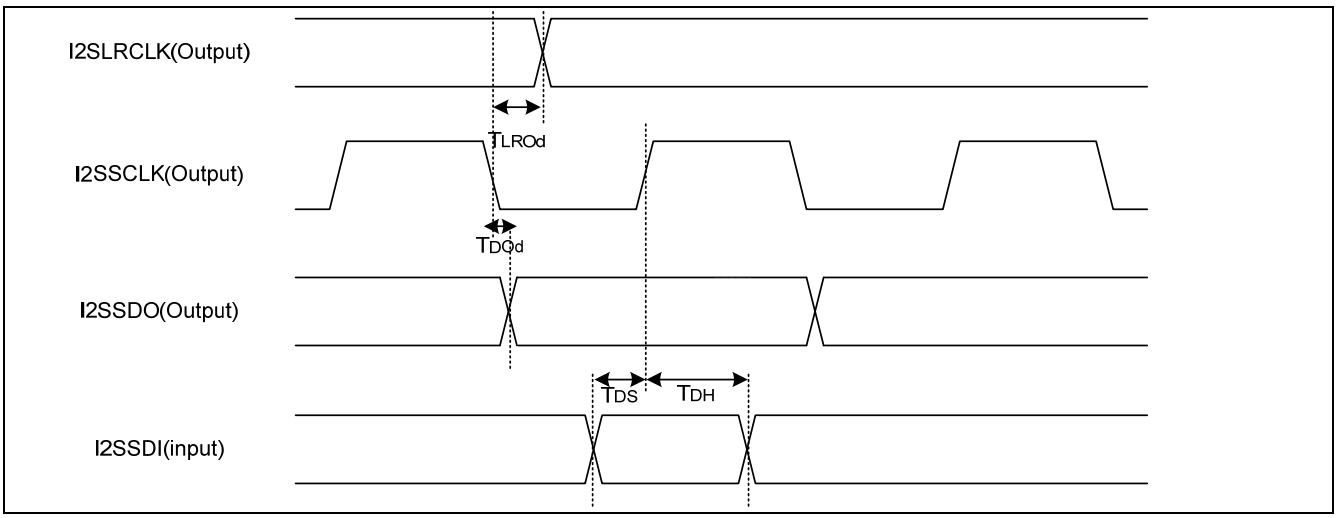


Figure 44-25. IIS Interface Timing (Master)

Table 44-24. IIS Controller Module Signal Timing Constants(I2S Master / TX)

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDpcm = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min.	Type.	Max	Unit
LR Clock Out Delay	TLROd	0.2	-	2.5	ns
Serial Data Out Delay	TDOd	0	-		ns
Serial Data In Setup Time	TDS	15	-		ns
Serial Data In Hold Time	TDH	5	-		ns

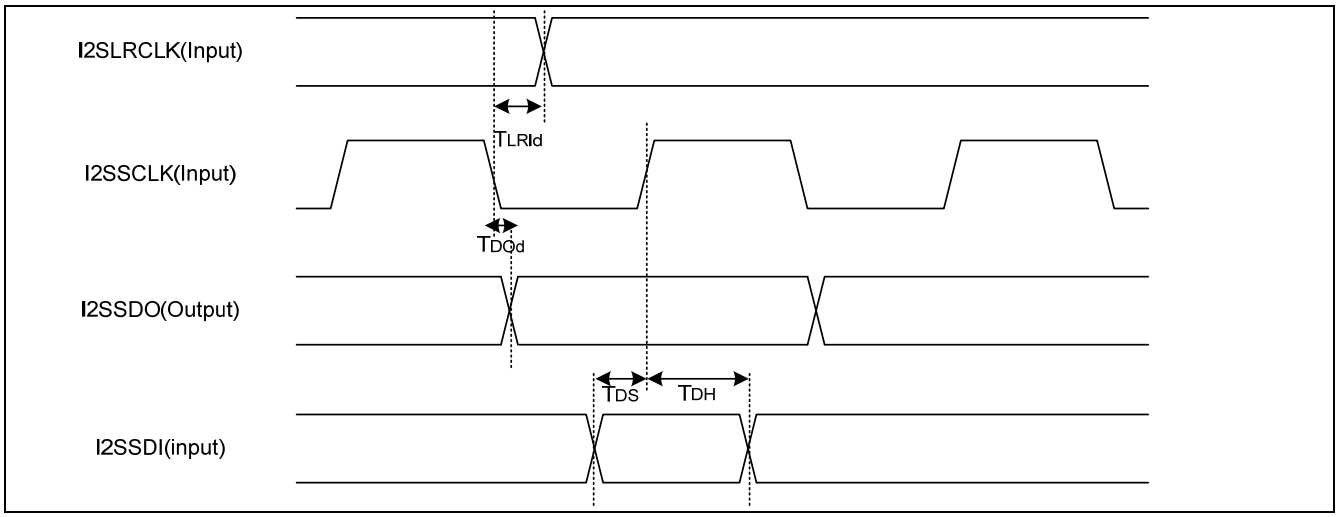


Figure 44-26. IIS Interface Timing (Slave)

Table 44-25. IIS Controller Module Signal Timing Constants(I2S Slave / RX)

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDpcm = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min.	Type.	Max	Unit
LR Clock Input Delay	TLRId	0	-		ns
Serial Data Out Delay	TDOd	7	-		ns
Serial Data Setup Time	TDS	2	-		ns
Serial Data Hold Time	TDH	5	-		ns

44.17 USB HOST1.1 AC ELECTRICAL CHARACTERISTICS

Table 44-26. AC Electrical Characteristics (Driver@FS)

Parameter	Symbol	Conditions	Min	Type	Max	Unit
Rise Time	Tr	CL = 50 pF	4		20	ns
Fall Time	Tf	CL = 50 pF	4		20	ns
Differential Rise and Fall Timing Matching	TRFM		90		111.11	%
Output Signal Crossover Voltage	VCRS		1.3		2.0	V
Driver Output Resistance	ZDRV	Steady State Drive External Resistance 39 Ω	39		44	Ω

Table 44-27. AC Electrical Characteristics (Driver@LS)

Parameter	Symbol	Conditions	Min	Type	Max	Unit
Rise Time	Tr	CL = 200 ~450pF	75		300	ns
Fall Time	Tf	CL = 200 ~450pF	75		300	ns
Differential Rise and Fall Timing Matching	TRFM		80		125	%
Output Signal Crossover Voltage	VCRS		1.3		2.0	V

44.18 PCM AC ELECTRICAL CHARACTERISTICS

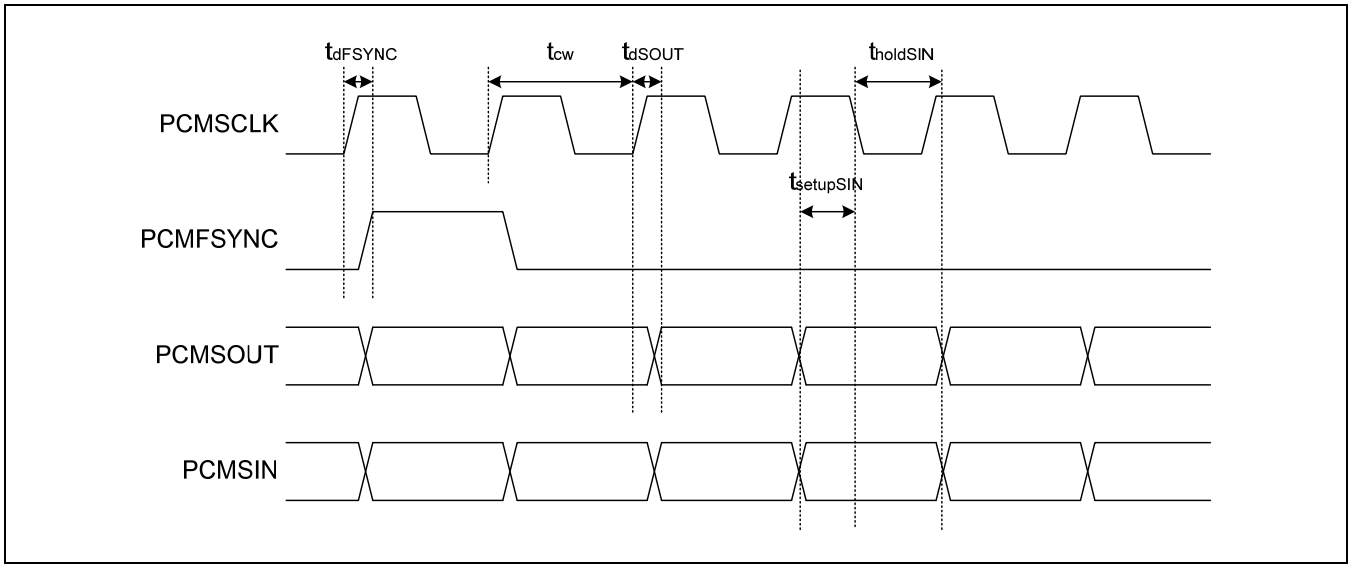


Figure 44-27. PCM Interface Timing

Table 44-28. PCM Module Signal Timing Constants

(VDDINT= 1.2V± 0.05V, TA = -40 to 85°C, VDDpcm = 3.3V ± 0.3V, 2.5V ± 0.2V, 1.8V ± 0.1V)

Parameter	Symbol	Min.	Typ.	Max	Unit
PCMSCLK clock width	1/tCW	0.128	-	8.192	MHz
PCMSCLK to PCMFSYNC delay	tdFSYNC	0.5	-		ns
PCMSCLK to PCMSOUT delay	tdSOUT	-0.5	-		ns
PCMSIN setup time	tsetupSIN	15	-		ns
PCMSIN hold time	tholdSIN	5	-		ns

NOTE: This table is applied to PCM0 and PCM1, respectively

44.19 I2C AC ELECTRICAL CHARACTERISTICS

Table 44-29. IIC BUS controller Module Signal Timing

(VDDINT, 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	-	-	ns
STOP setup time	tSTOPH	std. 4.0 fast 0.6	-	-	μs

NOTES : std. means Standard Mode and fast means Fast Mode.

1. The IIC data hold time(tSDAH) is minimum 0ns.

   (IIC data hold time is minimum 0ns for standard/fast bus mode IIC specification v2.1)

   Please check the data hold time of your IIC device if it's 0 ns or not.

2. The IIC controller supports only IIC bus device(standard/fast bus mode), not C bus device.

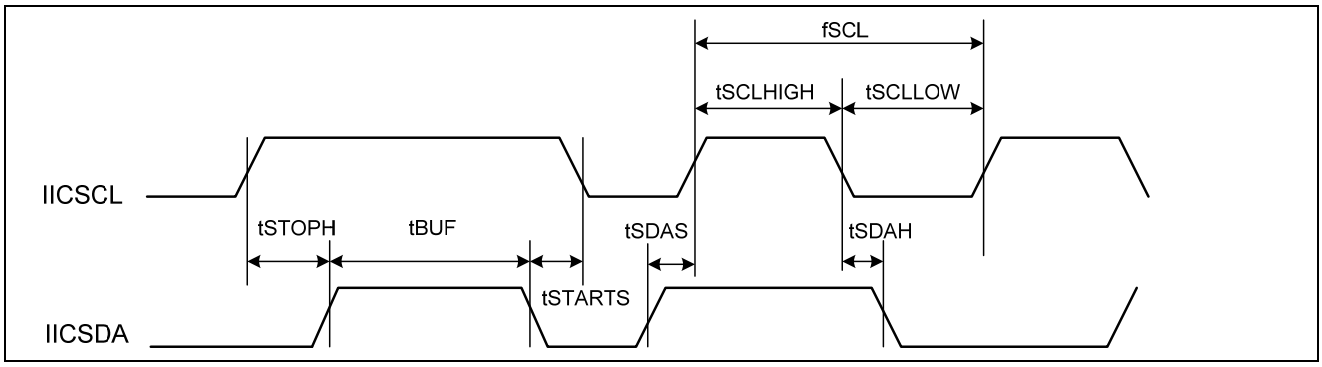
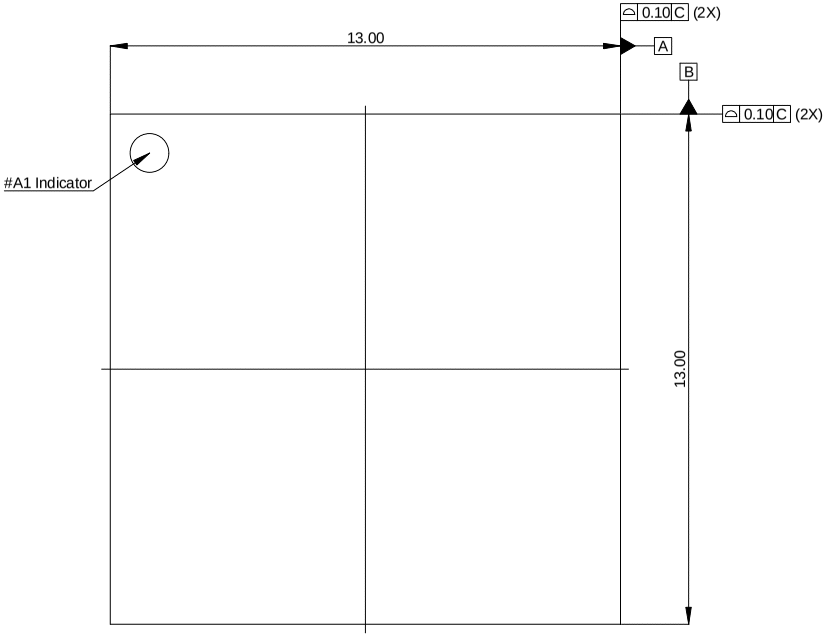


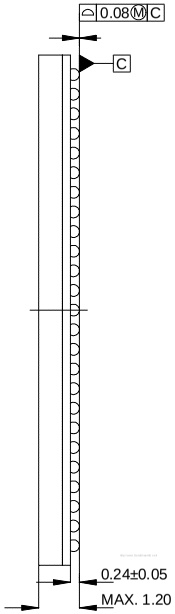
Figure 44-28. IIC Interface Timing

45 MECHANICAL DATA

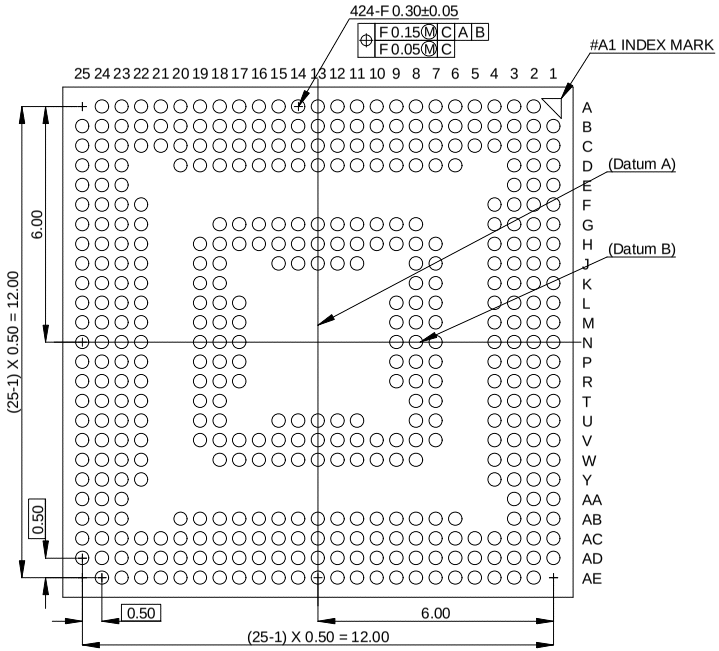
REV.	DATE	ECN#
000	2006.12.19	



TOP VIEW



SIDE VIEW



BOTTOM VIEW

NOTE

1. ALL DIMENSIONS AND TOLERANCES CONFIRM
   ASME Y14.5M-1994
2. REFERENCE SPECIFICATIONS:
   A.THIS DRAWING CONFIRMS TO THE JEDEC REGISTERED OUTLINE MO-195

TITLE: **424-FBGA-13.0X13.0 PACKAGE OUTLINE**			DESIGNER: B.Y.CHO
UNIT: mm	TOLERANCE: ±0.10	SCALE: N/A	SHEET: 1/1
DWG. NO.: SK-06206-O	SPEC NO.: **ADSXXXXX**	SAMSUNG ELECTRONICS	