IMAPx210 USER's MANUAL

Internet Multimedia Application Processor

March 28, 2010

Rev 0.50

Shanghai InfoTM Microelectronics Co., Ltd

Important Notice

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

InfoTM 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 InfoTM or others.

InfoTM makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does InfoTM 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.

InfoTM 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 InfoTM product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a InfoTM product for any such unintended or unauthorized application, the Buyer shall indemnify and hold InfoTM 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 InfoTM was negligent regarding the design or manufacture of said product.

Internet Multimedia Application Processor (IMAPx210)
User's Manual, Revision 0.50

Shanghai InfoTM Microelectronics Co., Ltd.

Building 11, No. 115, Lane 572, Bibo Rd, Zhangjiang Hi-Tech Park, Pudong Shanghai, 201203, P.R.C

Web: www.infotmic.com.cn

Tel: +86-21-5027-7720 / 7721

Fax: +86-21-5027-7736

Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
0.10	Initial Draft	--	Rongwei Jin	2010-01-05
0.20	Draft	0.10	Jinlin Lin	2010-03-08
0.30	Draft	0.20	Jinlin Lin	2010-03-15
0.40	Draft	0.30	Jinlin Lin	2010-03-23
0.50	Draft	0.40	Jinlin Lin	2010-03-28

Table of Contents

List of Figures

List of Tables

1 Product Overview

1.1 Architectural Overview

The IMAPx210 integrates advanced and power-efficient ARM1136JFS core, which operates at speeds up to 1GHz. The IMAPx210 high-performance, low-power, multifunction processor is a highly-integrated single-chip solution, embedded full-format HD video codec, 2D/3D graphic accelerator and knockout peripheral interface, for variety applications.

Figure 1-1: Application Field of IMAPx210

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/3D graphics, display manipulation and scaling. An integrated Full Format Codec (FFC) supports encoding and decoding of JPG/MPEG4/H.263/H.264 and decoding of VC1, RV8/9/10, DIVX4/5/6, XVID and ON2 VP6. 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 OpenVG 1.0, OpenGL ES 1.1 & 2.0.

The IMAPx210 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 two external memory ports, one DRAM ports and one SRAM port. The DRAM port can be configured to support DDR2, mobile DDR, and DDR. The SRAM port supports NOR-Flash, NAND-Flash, HardDisk/PATA-SSD/Compact-Flash and ROM type external memory.

To reduce total system cost and enhance overall functionality, the IMAPx210 includes many hardware peripherals such as a Camera Interface, TFT 24-bit true color LCD controller, System Manager (power management & clock management, etc.), 4-channel UART, 32-channel private and common DMA, 7-channel 32bit Timers with 2 PWM output, General Purpose I/O Ports, I2S/AC97-Bus audio interface, SSI interface, CF/IDE interface, KeyPad interface, I2C-BUS interface, USB 2.0 Host, USB OTG Device operating at high speed (480Mbps), SD/SDHC/SDIO Host Controller, Ethernet MAC and PLLs for clock generation.

1.2 Features

This section summarizes the features of the IMAP210. Figure 1-2 is an overall block diagram of the IMAPx200/210/220.

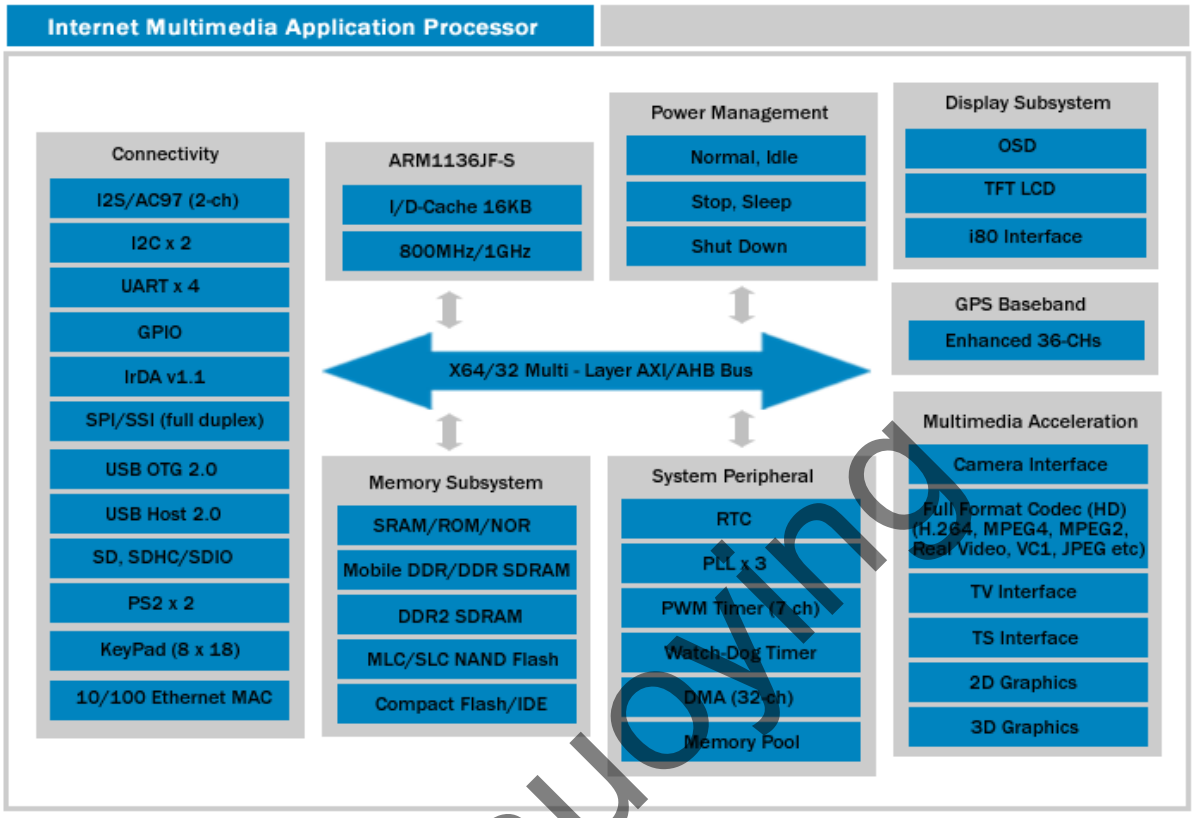


Figure 1-2: IMAPx200/210/220 Block Diagram

1.2.1 Microprocessor

The ARM1136JF-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 ARM1136JF-S processor include:

- ARM1136JF-S Core, 1GHz
- Eight-stage pipeline
- Branch prediction with return stack
- Low-interrupt latency
- Instruction and Data Memory Management units (MMUs), managed by using micro TLB structures backed by a unified main TLB
- Instruction and Data L1 Caches, including a non-blocking Data cache with Hit-Under-Miss
- Virtually indexed/physically addressed L1 caches
- 64-bit interface to both L1 caches
- Write buffer (bypassable)
- High-Speed Advanced Micro Bus architecture (AMBA) L2 interface
- Vector Floating Point coprocessor (VFP) for hardware acceleration of 3D graphics and other floating-point applications
- JTAG-based debug support
- Hyper-L2 instruction accelerator

1.2.2 Memory Subsystem

The IMAPx210 microprocessor provides the following Memory Subsystem features:

- High bandwidth memory matrix subsystem
- One external dram memory port (DDR2/MDDR/DDR)
- One static memory port (Flash/ROM/SRAM type)
- Matrix architecture increases overall bandwidth with the simultaneous access capability

DRAM Memory Port

- DDR / Mobile DDR Interface
  - ► 16/32-bit DDR/Mobile DDR, 266MHz (double data rate)
  - ► 2 chips support, up to 1GB per chip
- DDR2 Interface
  - ► 16/32-bit DDR2, 533MHz (double data rate)
  - ► 2 chips support, up to 1GB per chip

SRAM Memory Port

- 8/16-bit SRAM/ROM/NorFlash
- 26-bit address range support (64MB)
- Up to 4 chips support

NandFlash Interface

- 8/16-bit SLC/MLC NandFlash support
- NandFlash Page Type: 256Words, 512Bytes, 1KWords, 2KBytes, 2KWords and 4Kbytes.
- 1-bit ECC for SLC and 4 burst length (total 36-bits@512Bytes) ECC for MLC NAND flash.
- Auto boot: The boot code is transferred into 8-KBytes IBRAM during reset. After the transfer, the boot code will be executed on the IBRAM.
- Built-in DMA master controller, support two DMA modes: SDMA and ADMA.
- Up to 2 chips support

CF/IDE Interface

- Support HardDisk/PATA-SSD/CompactFlash
- Support PIO MODE0-MODE6
- Support UDMA MODE0-MODE5
- Private DMA & Descriptor support

1.2.3 Multimedia Acceleration

The IMAPx210 microprocessor provides the following Multimedia Acceleration features:

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 8192x4096 in YCbCr 4:2:2 format
- Codec/Preview output image generation (YCbCr 4:2:0/4:2:2 format and RGB 16/24-bit format)
- Codec output YCbCr to planar, semi-planar and interleaved store format
- Image windowing and digital zoom-in function
- H/W Color Space Conversion

Full Format Codec (FFC)

- Multi-Format Encoder
  - ► H.263 Profile 0, Levels 10-70, up to SXGA resolution
  - ► MPEG-4 Simple Profile, Levels 0-6, main profile, level 4, up to SXGA resolution
  - ► H.264 Baseline Profiles, level 1-3.2, up to SXGA resolution
  - ► JPEG baseline DCT mode & JFIF 1.02 file format, up to 16MPixels resolution

- Full-Format HD Decoder
  - ► MPEG-4 Simple & Advanced Profile, up to 1080p resolution
  - ► H.264 High, Main & Baseline Profiles, up to 1080p resolution
  - ► H.263 Profile 0, Levels 10-70, up to D1 resolution
  - ► VC1 Simple, Main & Advanced Profiles, up to 1080p resolution
  - ► MPEG-2 Main Profile, up to 1080p resolution
  - ► JPEG baseline DCT mode, up to 64MPixels resolution
  - ► Sorenson Spark, up to 1080p resolution
  - ► RV8/9/10, up to 1080p resolution
  - ► Divx4/5/6, up to 1080p resolution
  - ► Xvid, up to 1080p resolution
  - ► On2 VP6 E and S Profiles

- Post-Processing
  - ► Input data format
    Any format generated by the decoder in combined mode
    - YCbCr 4:2:0 semi-planar
    - YCbCr 4:2:0 planar
    - YCbYCr 4:2:2
    - YCrYCb 4:2:2
    - CbYCrY 4:2:2
    - CrYCbY 4:2:2
  - ► Input image size (combined mode)
    - 48 x 48 to 8176 x 8176 (66.8 Mpixels)
    - Step size 16 pixels
  - ► Input image size (stand-alone mode)
    - Width from 48 to 8176
    - Height from 48 to 8176
    - Maximum size limited to 16.7 Mpixels
    - Step size 16 pixels
  - ► Output image size
    - 16 x 16 to 1920 x 1088
    - Horizontal step size 8
    - Vertical step size 2
  - ► Image up-scaling
    - Bicubic polynomial interpolation with a four-tap horizontal kernel and a two-tap vertical kernel
    - Arbitrary, non-integer scaling ratio separately for both dimensions
    - Maximum output width is 3x the input width (within the maximum output image size limit)
    - Maximum output height is 3x the input height – 2 pixels (within the maximum output image size limit)
  - ► Image down-scaling
    - Proprietary averaging filter
    - Arbitrary, non-integer scaling ratio separately for both dimensions
    - Unlimited down-scaling ratio (e.g. from 16Mpixel to QVGA)
  - ► YCbCr to RGB color conversion
    - BT.601-5 compliant

- BT.709 compliant
- User definable conversion coefficient

► Dithering, 2x2 ordered spatial dithering for 4, 5 and 6 bit RGB channel precision

► Programmable alpha channel, constant eight bit value can be set to the alpha channel of the 24-bit RGB output data to control the transparency of the output picture. The resulting 32-bit ARGB data can be used as input data for later alpha blending.

► Alpha blending

Output image can be alpha blended with two rectangular areas. YCbCr semi-planar 4:2:0 PP output format is not supported when performing alpha blending.

The supported overlay input formats are following:

- 8 bit alpha value + YCbCr 4:4:4, big endian channel order being A-Y-Cb-Cr, 8 bits each
- 8 bit alpha value + 24 bit RGB, big endian channel order being A-R-G-B, 8 bits each

► Deinterlacing, conditional spatial deinterlace filtering. Supports only YCbCr 4:2:0 input format.

► RGB image contrast adjustment, segmented linear

► RGB image brightness adjustment, linear

► RGB image color saturation adjustment, linear

► De-blocking filter for MPEG-4 simple profile / H.263 / Sorenson, using a modified H.264 in-loop filter as a postprocessing filter. Filtering has to be performed in combined mode.

► Image cropping / digital zoom, user definable start position, height and width. Can be used with scaling to perform digital zoom. Usable only for JPEG or stand-alone mode.

► Picture in picture, output image can be written to any location inside video memory. Up to 1920 x 1088 sized displays supported

► Output image masking, output image writing can be prevented on two rectangular areas in the image . The masking feature is exclusive with alpha blending; however it is possible to have one masking area and one blending area.

► Image rotation

- Rotation 90, 180 or 270 degrees
- Horizontal flip
- Vertical flip

2D/3D Graphics Accelerator

- Supported Screen Resolution
  - ► HD1080p (multiple screens up to 2K x 2K each)
- Full Featured 3D Pipeline
  - ► OpenGL ES 2.0 and OpenGL ES 1.1
  - ► Full 32-bit floating point pipeline including shaders
  - ► Unified vertex and pixel/fragment shader
  - ► High-performance dependent texture operation
  - ► Alpha blending
  - ► Depth and stencil compare
  - ► Support for 8 simultaneous textures
  - ► Cubic environment texture, Projective texture and Depth texture
  - ► Point sample, bi-linear, tri-linear and anisotropic filtering.
- Full Featured 2D Pipeline
  - ► Bit BLT & stretch BLT
  - ► Rectangle fill and clear
  - ► Mono expansion for text rendering
  - ► ROP2, ROP3 and ROP4
  - ► Alpha blending including Java 2 Porter-Duff compositing blending rules
  - ► Supports rendering size of 32K x 32K
  - ► 90 degree rotation

- ► Vertical and Horizontal mirror
- ► Supports up to 255 rectangles per primitive call
- ► Transparency by monochrome mask, chroma key or pattern mask
- ► Color space conversion between YUY and RGB
- ► Clipping
- ► Color Index Input conversion support
- ► Filter Blit
- ► Multi data formats support

1.2.4 Display Subsystem

On Screen Display (OSD)

- 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 24-BPP non-palletized color
  - ► Supports multi-format RGB and YCbCr 4:2:0
  - ► Supports X,Y indexed position
  - ► Supports 4 bit Alpha blending: Plane / Pixel
- Up to 4 layers PIP(OSD) function
- Up to 2Kx2K screen resolution
- Window Source format
  - ► Supports 1, 2, 4 or 8-BPP (bit per pixel) palletized color
  - ► Supports 8, 16, 18 or 24-BPP non-palletized color
- Palette Usage
  - ► 256 x 25(ARGB) bits palette RAM for Window 0
  - ► 256 x 25(ARGB) bits palette RAM for Window1
  - ► 16(entry) x 16 bits palette Register Array for Window 2
  - ► 16(entry) x 16 bits palette Register Array for Window 3
- Soft Scrolling
  - ► Horizontal : 1 pixel resolution
  - ► Vertical : 1 pixel resolution
- Supports transparent overlay
- Supports hardware cursor
- Supports color key function

TFT LCD Interface

- Supports maximum 16M color TFT at 24bit per pixel mode
- Supports multiple screen size
  - ► Typical actual screen size: 1920x1080, 1024x768, 800x480, 640 x 480, 320 x 240, and others
  - ► Maximum virtual screen size is 4Mbytes.
  - ► Maximum virtual screen size in 64K color mode: 2048 x 2048 and others

TV Interface

- Built-in Color Space Conversion matrix
- External or internal clock selection
- Support ITU-R BT 656 8-bit interface (YCbCr422)
- Support ITU-R BT 601 16-bit/8-bit interface (YCbCr422)
- Support ITU-R BT 601 24-bit interface (RGB888)
- Support user defined size resolution up to HDTV

- Support user defined frame and field size

Intel 8080 System Display Interface (I80IF)

- Support 16 normal command by software trigger
- Support 16 auto-command before video data
- Programmable timer
- Support 18/16/9/8 wire connection
- Automatism and Manual operation

1.2.5 Audio Interface

AC97 Controller

- Variable sampling rate (48kHz and below)
- Support single audio master codec
- 1 port stereo inputs/1 port stereo outputs with 16 depth FIFO
- 20/18/16-bit stereo (2-channel) audio
- Interrupt, polled-mode or DMA mode operation

IIS-Bus Interface

- Full duplex communication due to the independence of transmitter and receiver
- Master mode only, support codec clock output
- Audio data resolutions of 8, 12, 16, 20, 24, and 32 bits
- SCLK Gating for multiple clock cycle : 8, 10, 12, 14, 16, 18, 20, 24 and 32
- Independent Transmit and receive buffers(16 depth)
- Programmable FIFO thresholds
- Interrupt, polled-mode or DMA mode operation
- Configurable Clock source of Clock Generator

1.2.6 USB Support

USB OTG

- USB Peripheral Features
  - ► Compliant with the Universal Serial Bus Specification Revision 2.0
  - ► Support High-speed (480Mbit/s) and Full-speed (12Mbit/s) operation as a peripheral
  - ► Support USB transfers: control, bulk, interrupt, and isochronous
  - ► Support all USB standard commands
  - ► Support Class/Vendor commands by passing the commands to the application system
  - ► Up to 4 Physical Interfaces by default and 16 alternate settings per interface
  - ► Up to 7 logical Endpoints by default (including Endpoint0)
- USB Host Features
  - ► Adopt the Enhanced Host Controller Interface Specification for Universal Serial Bus Revision 1.0 at operation of a USB2.0 Host
  - ► Adopt the Universal Host Controller Interface (UHCI) Design Guide Revision 1.1 at operation of a USB1.1 Host
  - ► Support High-speed (480Mbit/s), Full-speed (12Mbit/s), and Low-speed (1.5Mbit/s)
  - ► Support USB transfers: control, bulk, interrupt, and isochronous

- USB OTG Features
  - ▸ Compliant with the On-The-Go Supplement to the USB 2.0 Specification Revision 1.0a
  - ▸ Session Request Protocol (SRP)
  - ▸ Host Negotiation Protocol (HNP)

USB Host

- 3-port USB Host, one port is shared with USB-OTG
- Compatible with OHCI Rev. 1.0
- Compatible with EHCI Rev. 1.0
- Compatible with USB Specification version 2.0

1.2.7 SD/SDHC/SDIO Controller

- Up to 3 SD/SDHC/SDIO controller
- Up to 50MHz SD bus clock support
- SD Standard Host Specification (ver 2.0) compatible
- SD Memory Card Specification (ver 2.0) 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 mode switch support
- Auto CMD12 support
- Suspend / Resume support
- Read Wait operation support
- Card Interrupt support

1.2.8 Ethernet MAC Controller

- Supports 10/100-Mbps data transfer rates with IEEE 802.3-compliant MII PHY interfaces to communicate with an external Ethernet PHY
- Supports both full-duplex and half-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames

1.2.9 Serial Communication

Universal Asynchronous Receiver/Transmitter (UART)

- Four UART support
- Each UART Ch with IrDA 1.0 and RS232
- Each UART Ch with DMA-based or interrupt-based operation
- Each UART Ch with two 64-byte FIFOs (TX/RX FIFO)
- UART Ch 0 and 1 with nRTS0, nCTS0, nRTS1, and nCTS1
- Support Modem Control and Auto Flow Control
- Supports handshake transmit/receive

SSI Interface

- Serial-master or serial-slave operation
- Serial Motorola SPI, Texas Instruments Synchronous Serial Protocol and National Semiconductor Microwire
- Dynamic control of the serial bit rate of the data transfer
- Polling and Interrupt transfer mode

I2C-BUS Interface

- Two independent I2C bus interface
- Two-wire I2C serial interface (SDA and SCL)
- Three speeds:
  - ► Standard mode (100 Kb/s)
  - ► Fast mode (400 Kb/s)
  - ► High-speed mode (3.4 Mb/s)
- Master or Slave I2C operation
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Bulk transmit mode
- Transmit and receive buffers(16 bytes)
- Interrupt or polled-mode operation
- Handles Bit and Byte waiting at all bus speeds

1.2.10 Input Devices

KeyPad interface

- Support 8x18 Key Matrix
- Provides internal debounce filter

PS2 interface

- PS/2 host support standard PS/2 protocol
- Interrupt and polling operation mode
- Programmable clock frequency

1.2.11 GPIO

- 193 Flexibly configurable GPIO
- 117 configurable external Interrupts

1.2.12 System Peripherals

Memory Pool

- Ten piece of 4Kx32 memory blocks
- Multi operation mode for different application
- Multi data interface arbitrating
- Video decode module direct channel
- Internal logic analyzer direct channel
- Internal DMA channel for fast memory operation
- Clock gating and power supply shutdown mode for reduce power consumption

DMA Controller

- 32-channel DMA, 8 general channels and 24 special channels .
- Supports memory to memory, peripheral to memory, memory to peripheral, and peripheral to peripheral
- Burst transfer mode to enhance the transfer rate.
- Support two-layer AHB
- Support Multi-Block transfer

Watchdog Timer

- Interrupt request or system reset at time-out

PWM Timer

- Five 16-bit PWM timers with 4 PWM output
- Two 8-bit prescalers & Two 4-bit divider
- Programmable duty cycle, frequency and polarity of output waveform (PWM)
- Auto reload mode or one-shot pulse mode
- Dead-zone generator

Common Timer

- Two 32-bit common timers
- Support for independent clocking of each timers
- Support for two operation modes: free-running and user-defined count

Real Time Clock (RTC)

- BCD number: second, minute, hour, date, day, month, and year
- Alarm function: alarm interrupt or wake-up interrupt
- 32.768 KHz operation
- Leap year generator
- Independent power pin
- Supports millisecond tick time interrupt for RTOS kernel time tick.

Interrupt Controller

- 56 Maskable Interrupt Sources
- Level/Edge mode on external interrupt source(configuration in GPIO)
- Programmable priority of interrupt
- Supports Fast Interrupt request (FIQ) for very urgent interrupt request

1.2.13 System Management

Clock Generator

- Include three on-chip PLLs called main PLL (APLL) and DPLL and extra PLL (EPLL). APLL generates the system reference clock, DPLL generates the asynchronous clock for CPU, and EPLL generates the clocks for the special functional blocks such as USB, IIS, Camera I/F, OSD, LCD and etc.
- Independent clock ON/OFF control to reduce power consumption for all modules except CPU, including VENC, VDEC, GPU, MPool and etc.
- Support IDLE mode to turn ON/OFF of CPU clock. CPU clock is turned off by software and turned on by IDLE wake-up mechanism.

Reset Controller

The IMAPx210 has five types of reset signals and SYSCON can place the system into one of five resets.

- Hardware reset: It is generated by asserting XnRESET pin. 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.
- Warm reset: It is generated by asserting XnWRESET pin. XnWRESET is used to initialize IMAPx210 and preserve current hardware status.
- 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.
- Software reset: It is generated by setting SW_RESET.
- Wakeup reset: It is generated when IMAPx210 wake up from SLEEP mode. Since internal hardware states are not available any more after SLEEP mode, they must be initialized.

Power Management

- Seven controllable power domains

- Clock-off control for individual module.
- Support four power-down modes, IDLE, STOP, SLEEP, and SHUTDOWN to optimize the power dissipation.
- Wake-up by one of external interrupt (STOP and SLEEP mode), RTC alarm (STOP and SLEEP mode), KeyPad (STOP mode), XPowKey (SHUTDOWN).

1.3 Package & Technology

- 441-Pin TFBGA (18mm x 18mm)
- CMOS 65nm LP

1.4 Pin Assignments

Table 1-1: IMAPx210 Pin Assignments

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
A	XGPSDIN0	XGPSDIN1	XSD1DAT0	XSD1CLK	XVVCLK	XVVD20	XVVD18	XVVD16	XVVD13	XVVD11	XVVD3	XI2CSDA0	XI2CMCL1	XPS2DATA0	XTOUT2	XTOUT1	XTOUT0	XAUBCLK	XAUFSYNC	XAUSDO	XAUSDI	A
B	XGPSCLK	XSD2DAT0	XSDCLK	XSD1DAT1	XVVDEN	XVVD22	XVVD14	XVVD15	XVVD12	XVVD10	XVVD0	XVVD2	XI2CMCL0	XI2CMDA1	XPS2DATA1	XJDTMS	XEINT3	XEINT1	XEXTCLK	XJDTDO	XJDTCK	B
C	XSD2DAT1	XSD2DAT2	XSD1DAT2	XSDDAT0	XSD1DAT3	XVHSYNC	XVVD23	XVVD9	XVVD6	XVVD8	XJCRTCK	XJCTDO	XJCNTRST	XJCTCK	XEINT4	XEINT5	XBPARA1	XXTO	XNWRESET	XTESTMODE	XBPARA3	C
D	XSD2DAT3	XSD2CLK	XSD1CMD	XSDDAT1	XSDDAT2	XVVSYNC	XVVD19	XVVD4	XVVD1	XVVD5	XAUFCLK	XJCTMS	XJDMODE	XEINT2	XEINT0	XPOWIND	XBPARA2	XBPARA0	XJDTDI	XXTI	XNRESET	D
E	XCIVSYNC	XSD2CMD	XSDDAT3	XSDCMD	XVPWREN	XVVD21	XVVD17	XVVD7	XPS2CLK0	XTOUT3	XNRSTOUT	XJCTDI	XPS2CLK1	XBPARA4	GND	XRTCXTO	XRTCXTI	XRTCINT	XPOWKEY	XPOWON	XPADATA8	E
F	XCIHREF	XCIYDAT0	XCIYDAT1	XCIYDAT2	XCIYDAT3	VDDIO_VV	VDDIO_VV	VDDIO_VV	VDDIO_VV	VDDIO_VV	DVDD_EPLL	DVDD_APLL	DVDD_DPLL	VDD_RTC	AVDD_OSC	VDDC	XPACLK1	XPADDR2ODT1	XPADATA10	XPADQM1	XPADATA9	F
G	XCIYDAT4	XCIYDAT5	XCIYDAT6	XCIYDAT7	XCINRST	VDDIO_VV	GND	VDDC	GND	VDDC	AHVDD_EPLL	AHVDD_APLL	AHVDD_DPLL_	VDDIO_RTC	AVSS_OSC	VDDIO_DDR	XPANCLK1	XPADATA11	XPADATA12	XPADQS1	XPADQSN1	G
H	XCICLK	XCIPCLK	XETHMDIO	XETHRXD0	XETHMDC	VDDIO_SD	GND	GND	GND	GND	AHVSS_EPLL	AHVSS_APLL	AHVSS_DPLL	VDDIO_OSC	VDD_OSC	VDDIO_DDR	XPADCSN1	XPADATA13	XPADATA0	XPADATA15	XPADATA14	H
J	XETHTXD2	XETHTXD1	XETHRXD1	XETHRXD2	XETHRXER	VDDIO_CI	VDDC	GND	GND	GND	DVSS_EPLL	DVSS_APLL	DVSS_DRLL	GND	VDDP_DDR	VDDIO_DDR	XPADDROP1	XPADQSN0	XPADQS0	XPADATA2	XPADQM0	J
K	XETHTXEN	XETHTXD3	XETHRXD3	XETHRCLK	XETHCRS	VDDC	GND	GND	GND	GND	GND	GND	GND	GND	GND	VDDIO_DDR	XPADATA1	XPADATA6	XPADATA5	XPADATA4	XPADATA3	K
L	XSSITXD	XETHCOL	XETHTCLK	XETHTXD0	XETHRXDV	VDDIO_ET	GND	GND	GND	GND	GND	GND	GND	GND	VDDP_DDR	VDDIO_DDR	XPADDROP0	XPAADDR3	XPAADDR7	XPADATA7	XPACKE	L
M	XSSIRXD	XSSICSN1	XSSICSN0	XSSICSN2	XSSICSN3	VDDIO_ET	GND	GND	GND	VSDL_USBO	VSDL_USBH0	VSDL_USBH1	GND	GND	VDDR_DDR	VDDR_DDR	GND	XPAADDR0	XPAADDR2	XPADWEN	XPADBANK0	M
N	XSSICLK	XSSI1RXD	XSSI1CLK	XSSI1TXD	XSSI1CSN	VDDC	GND	GND	GND	VSSA_USBO	VSSA_USBH0	VSSA_USBH1	GND	GND	VDDR_DDR	VDDR_DDR	VDDIO_DDR	XPAADDR4	XPAADDR8	XPAADDR10	XPAADDR1	N
P	XURXD0	XUTXD0	XURXD2	XUNCTS0	XIRSDBW	XCFADD1	VDDC	GND	GND	VBUS_USBO	VBUS_USBH0	VBUS_USBH1	GND	GND	GND	VDDIO_DDR	XPAADDR13	XPAADDR5	XPAADDR12	XPAADDR9	XPADBANK1	P
R	XURXD1	XUTXD1	XUTXD3	XURXD3	XCFCS1N	XFHDAT1	VDDIO_ET	VDDIO_ET	VDDC	VDDL_USBO	VDDL_USBH0	VDDL_USBH1	VDDP_DDR	VDDP_DDR	VDDP_DDR	VDDIO_DDR	XPAADDR14	GND	VDDIO_DDR	XPADRASN	XPADCASN	R
T	XUNRTS0	XUDTR	XUNCTS1	XUTXD2	XCFADD0	XCFADD2	XFHDAT6	XKBCOL8	XKBROW0	VDDA_USBO	VDDA_USBH0	VDDA_USBH1	VDDC	VDDC	VDDC	VDDIO_DDR	XPALPCKE	XPADDR2ODT0	XPADBANK2	XPAADDR6	XPAADDR11	T
U	XUNRTS1	XURI	XNDCLE	XNDRNB	XCFCS0N	XFHDAT11	XKBCOL13	XKBCOL17	XKBROW2	XKBROW4	XDRVVBUS	XUSB_OVRES	VDDA_USBH1	VDDIO_DDR	VDDIO_DDR	VDDIO_DDR	XPADDROP3	XPACLK0	XPANCLK0	XPADCSN0	XPADATA24	U
V	XUDCD	XUDSR	XNDCS0N	XCFIORY	XFHDAT12	XFHDAT13	XKBCOL7	XKBCOL1	XKBCOL15	XKBROW3	XKBROW7	XUSB_OID	XUSB_VRES1	XUSB_ID1	GND	XPADDROP2	XPADATA17	XPADATA16	XPADATA28	XPADATA25	XPADQM3	V
W	XNDWEN	XNDCS1N	XNDALE	XCFIORD	XFHDAT3	XFHDAT4	XKBCOL0	XKBCOL5	XKBCOL19	XKBCOL16	XKBROW6	XKBROW5	XUSB_VRES0	XUSB_ID0	GND	XPADATA20	XPADATA21	XPADQM2	XPADATA31	XPADQS3	XPADQSN3	W
Y	XNDREN	XCFRSTN	XCFCD1N	XFHDAT0	XFHDAT5	XFHDAT10	XFHDAT15	XKBCOL4	XKBCOL2	XKBCOL3	XKBCOL14	XKBROW1	XUSB_S1DP	XUSB_H0DP	XUSB_H1DP	VDDIO_DDR	XPADATA22	XPADATA18	XPADQS2	XPADATA26	XPADATA27	Y
AA	XCFINT	XCFPWRN	XCFIOWR	XFHDAT2	XFHDAT7	XFHDAT8	XFHDAT14	XFHDAT9	XKBCOL6	XKBCOL10	XKBCOL11	XKBCOL12	XUSB_S1DM	XUSB_H0DM	XUSB_H1DM	VDDIO_DDR	XPADATA23	XPADATA19	XPADQSN2	XPADATA30	XPADATA29	AA
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	

Note: #A1 is used as index mark.

1.5 Pin Description

1.5.1 External Memory Port

DRAM Memory (DDR2/MDDR/DDR) Port

Signal	I/O	IO Power	Description
XPACLK[1:0]	O	VDDIO_DDR	DRAM port differential clock (positive)
XPAnCLK[1:0]	O	VDDIO_DDR	DRAM port differential clock (negative)
XPADCSN[1:0]	O	VDDIO_DDR	DRAM port chip select, up to 2 chips, active low
XPADRASN	O	VDDIO_DDR	DRAM port row address strobe, active low
XPADCASN	O	VDDIO_DDR	DRAM port column address strobe, active low
XPADWEN	O	VDDIO_DDR	DRAM port write enable, active low
XPACKE	O	VDDIO_DDR	DRAM port DDR2/DDR clock enable
XPALPCKE	O	VDDIO_DDR	DRAM port mDDR clock enable
XPADBANK[2:0]	O	VDDIO_DDR	DRAM port bank address
XPAADDR[14:0]	O	VDDIO_DDR	DRAM port address
XPADDR2ODT[1:0]	O	VDDIO_DDR	DRAM port on-die termination
XPADQM[3:0]	O	VDDIO_DDR	DRAM port data mask
XPADQS[3:0]	IO	VDDIO_DDR	DRAM port differential data strobe (positive)
XPADQSN[3:0]	IO	VDDIO_DDR	DRAM port differential data strobe (negative)
XPADATA[31:0]	IO	VDDIO_DDR	DRAM port bidirectional data bus
XPADDROP[3:0]	IO	VDDIO_DDR	DRAM port feedback signal

Note: DRAM port is dedicated port, not share with other ports.

Shared SRAM Memory Port (SRAM/NAND/CF-IDE/KeyPad)

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XSSICSN[1]	I	O	O	O	O	GPE[8]/EINT4[8]	GPE[8]	MST_SSI_CSN1	SRAM_nBE[0]	3
XSSICSN[2]	I	O	O	O	O	GPE[9]/EINT4[9]	GPE[9]	MST_SSI_CSN2	SRAM_nBE[1]	3
XSSICSN[3]	I	O	O	O	O	GPE[10]/EINT4[10]	GPE[10]	MST_SSI_CSN3	SRAM_nCS[1]	3
XSSICSN[0]	I	O	O	O	I	GPE[7]/EINT4[7]	GPE[7]	MST_SSI_CSN0	SRAM_nCS[2]	0
XSSICLK	I	O	O	O	I	GPE[6]/EINT4[6]	GPE[6]	MST_SSI_CLK	SRAM_nCS[3]	0
XKBROW7	I	O	O	I	O	GPJ[7]/EINT6[25]	GPJ[7]	SRAM_nWE	KB0_ROW7	2
XKBROW6	I	O	O	I	O	GPJ[6]/EINT6[24]	GPJ[6]	SRAM_nOE	KB0_ ROW6	2
XKBROW5	I	O	O	I	O	GPJ[5]/EINT6[23]	GPJ[5]	SRAM_nCS[0]	KB0_ROW5	2
XKBROW[4:0]	I	O	O	I	O	GPJ[4:0]/EINT6[22:18]	GPJ[4:0]	SRAM_ADDR[22:18]	KB0_ROW[4:0]	2
XKBCOL[17:14]	I	O	O	IO	O	GPH[3:0]/EINT6[3:0]	GPH[3:0]	SRAM_ADDR[17:14]	KB0_COL[17:14]	2
XKBCOL[13:0]	I	O	O	IO	O	GPI[13:0]/EINT6[17:4]	GPI[13:0]	SRAM_ADDR[13:0]	KB0_COL[13:0]	2
XCFADD[2:0]	I	O	O	--	O	GPP[2:0]	GPP[2:0]	SRAM_ADDR[25:23]/ CF_ADD[2:0]	--	2
XCFIORD	I	O	O	--	O	GPP[3]	GPP[3]	CF_IORD	--	2
XCFIOWR	I	O	O	--	O	GPP[4]	GPP[4]	CF_IOWR	--	2
XCFRSTN	I	O	O	--	O	GPP[5]	GPP[5]	CF_RESET	--	2
XCFPWRN	I	O	O	--	O	GPP[6]	GPP[6]	CF_PWRN	--	2
XCFCS0N	I	O	O	--	O	GPP[7]	GPP[7]	CF_CS0	--	2
XCFCS1N	I	O	O	--	O	GPP[8]	GPP[8]	CF_CS1	--	2
XCFIORDY	I	O	I	--	I	GPP[9]	GPP[9]	SRAM_nWAIT/ CF_IORDY	--	2
XCFINT	I	O	I	--	I	GPP[10]	GPP[10]	CF_INTRQ	--	2
XCFCD1N	I	O	I	--	I	GPP[11]	GPP[11]	CF_CD1	--	2
XUDSR	I	O	I	O	I	GPO[12]	GPO[12]	UART1_DSR	CF_DMACK	2
XURI	I	O	I	I	I	GPO[14]	GPO[14]	UART1_RI	CF_DMARQ	2

XNDRNB	I	O	I	--	I	GPQ[0]	GPQ[0]	NAND_RnB	--	2
XNDCLE	I	O	O	--	O	GPQ[1]	GPQ[1]	NAND_CLE	--	2
XNDALE	I	O	O	--	O	GPQ[2]	GPQ[2]	NAND_ALE	--	2
XNDWEN	I	O	O	--	O	GPQ[3]	GPQ[3]	NAND_nWE	--	2
XNDREN	I	O	O	--	O	GPQ[4]	GPQ[4]	NAND_nRE	--	2
XNDCS0N	I	O	O	--	O	GPQ[5]	GPQ[5]	NAND_nFCE[0]	--	2
XNDCS1N	I	O	O	--	I	GPE[14]/EINT4[14]	GPE[14]	NAND_nFCE[1]	--	0
XFHDAT[15:0]	I	O	IO	--	I	GPR[15:0]	GPR[15:0]	Flash_DATA[15:0]	--	2

Signal	I/O	IO Power	Description
SRAM_nCS[3:0]	O	VDDIO_ET	SRAM chip select, active low
SRAM_nBE[1:0]	O	VDDIO_ET	SRAM byte enable, active low
SRAM_nWE	O	VDDIO_ET	SRAM write strobe, active low
SRAM_nOE	O	VDDIO_ET	SRAM read strobe, active low
SRAM_ADDR[25:0]	O	VDDIO_ET	SRAM address, bit[25:22] is shared with CF_ADD[2:0]
SRAM_nWAIT	I	VDDIO_ET	SRAM wait signal, active low, shared with CF_IORDY
CF_ADD[2:0]	O	VDDIO_ET	CF-ATA address
CF_IORD	O	VDDIO_ET	CF-ATA IORD signal
CF_IOWR	O	VDDIO_ET	CF-ATA IOWR signal
CF_RESET	O	VDDIO_ET	CF-ATA RESET signal
CF_PWRN	O	VDDIO_ET	CF-ATA Power on/off control signal
CF_CS0	O	VDDIO_ET	CF-ATA CS0 signal
CF_CS1	O	VDDIO_ET	CF-ATA CS1 signal
CF_IORDY	I	VDDIO_ET	CF-ATA IORDY signal
CF_INTRQ	I	VDDIO_ET	CF-ATA INTRQ signal
CF_CD1	I	VDDIO_ET	CF-ATA CD1 signal
CF_DMARQ	I	VDDIO_ET	CF-ATA DMARQ signal
CF_DMACK	I	VDDIO_ET	CF-ATA DMACK signal
NAND_RnB	I	VDDIO_ET	NAND ready/busy signal
NAND_CLE	O	VDDIO_ET	NAND command latch enable signal
NAND_ALE	O	VDDIO_ET	NAND address latch enable signal
NAND_nWE	O	VDDIO_ET	NAND write enable signal
NAND_nRE	O	VDDIO_ET	NAND read enable signal
NAND_nFCE[1:0]	O	VDDIO_ET	NAND chip enable signal
Flash_DATA[15:0]	IO	VDDIO_ET	NAND/SRAM/CF-ATA shared data bus

Note: IMAPx210 has an internal Flash IO arbiter which arbitrates requests from NAND, CF-IDE and SRAM controller.

1.5.2 Serial Communication

Uart/KeyPad/IrDA/SPI

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XURXD0	I	O	I	IO	I	GPA[0]/EINT1[0]	GPA[0]	UART0_RXD	KB2_COL[14]	0
XUTXD0	I	O	O	IO	I	GPA[1]/EINT1[1]	GPA[1]	UART0_TXD	KB2_COL[15]	0
XUCTS0	I	O	I	IO	I	GPA[2]/EINT1[2]	GPA[2]	UART0_nCTS	KB2_COL[16]	0
XURTS0	I	O	O	IO	I	GPA[3]/EINT1[3]	GPA[3]	UART0_nRTS	KB2_COL[17]	0
XURXD1	I	O	I	IO	I	GPA[4]/EINT1[4]	GPA[4]	UART1_RXD	SPI_MISO	0
XUTXD1	I	O	O	IO	I	GPA[5]/EINT1[5]	GPA[5]	UART1_TXD	SPI_CLK	0
XUCTS1	I	O	I	IO	I	GPA[6]/EINT1[6]	GPA[6]	UART1_nCTS	SPI_MOSI	0
XURTS1	I	O	O	I	I	GPA[7]/EINT1[7]	GPA[7]	UART1_nRTS	SPI_nCS	0
XURXD2	I	O	I	IO	I	GPB[0]/EINT2[0]	GPB[0]	UART2_RXD	KB2_COL[10]	0
XUTXD2	I	O	O	IO	I	GPB[1]/EINT2[1]	GPB[1]	UART2_TXD	KB2_COL[11]	0
XURXD3	I	O	I	IO	I	GPB[2]/EINT2[2]	GPB[2]	UART3_RXD	KB2_COL[12]	0
XUTXD3	I	O	O	IO	I	GPB[3]/EINT2[3]	GPB[3]	UART3_TXD	KB2_COL[13]	0
XIRSDBW	I	O	--	IO	I	GPB[4]/EINT2[4]/	GPB[4]	--	KB2_COL[9]	0

						CAM_field				
XUDSR	I	O	I	O	I	GPO[12]	GPO[12]	UART1_DSR	CF_DMACK	2
XUDCD	I	O	I	--	I	GPO[13]	GPO[13]	UART1_DCD	--	2
XURI	I	O	I	I	I	GPO[14]	GPO[14]	UART1_RI	CF_DMARQ	2
XUDTR	I	O	O	--	O	GPO[15]	GPO[15]	UART1_DTR	--	2

Signal	I/O	IO Power	Description
UART0_RXD	I	VDDIO_ET	UART0 receives data input, shared with IrDA0-RXD
UART0_TXD	O	VDDIO_ET	UART0 transmits data output, shared with IrDA0-TXD
UART0_nCTS	I	VDDIO_ET	UART0 clear to send input signal
UART0_nRTS	O	VDDIO_ET	UART0 request to send output signal
UART1_RXD	I	VDDIO_ET	UART1 receives data input, shared with IrDA1-RXD
UART1_TXD	O	VDDIO_ET	UART1 transmits data output, shared with IrDA1-TXD
UART1_nCTS	I	VDDIO_ET	UART1 clear to send input signal
UART1_nRTS	O	VDDIO_ET	UART1 request to send output signal
UART1_DSR	I	VDDIO_ET	UART1 data set ready modem status input signal
UART1_DCD	I	VDDIO_ET	UART1 data carrier detect modem status input signal
UART1_RI	I	VDDIO_ET	UART1 ring indicator modem status input signal
UART1_DTR	O	VDDIO_ET	UART1 modem control data terminal ready output signal
UART2_RXD	I	VDDIO_ET	UART2 receives data input, shared with IrDA2-RXD
UART2_TXD	O	VDDIO_ET	UART2 transmits data output, shared with IrDA2-TXD
UART3_RXD	I	VDDIO_ET	UART3 receives data input, shared with IrDA3-RXD
UART3_TXD	O	VDDIO_ET	UART3 transmits data output, shared with IrDA3-TXD
SPI_MISO	IO	VDDIO_ET	SPI master input slave output signal
SPI_MOSI	IO	VDDIO_ET	SPI master output slave input signal
SPI_CLK	IO	VDDIO_ET	SPI master output clock slave input clock signal
SPI_nCS	I	VDDIO_ET	SPI slave chip select

PS2/I2C

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XI2CMCL0	I	O	IO	--	I	GPC[0]/EINT3[0]	GPC[0]	I2C_SCL0	--	0
XI2CMDA0	I	O	IO	--	I	GPC[1]/EINT3[1]	GPC[1]	I2C_SDA0	--	0
XI2CMCL1	I	O	IO	--	I	GPC[2]/EINT3[2]	GPC[2]	I2C_SCL1	--	0
XI2CMDA1	I	O	IO	--	I	GPC[3]/EINT3[3]	GPC[3]	I2C_SDA1	--	0
XPS2CLK0	I	O	IO	--	I	GPC[4]/EINT3[4]	GPC[4]	PS2_CLK0	--	0
XPS2DATA0	I	O	IO	--	I	GPC[5]/EINT3[5]	GPC[5]	PS2_DATA0	--	0
XPS2CLK1	I	O	IO	--	I	GPC[6]/EINT3[6]	GPC[6]	PS2_CLK1	--	0
XPS2DATA1	I	O	IO	--	I	GPC[7]/EINT3[7]	GPC[7]	PS2_DATA1	--	0

Note: This group of pads is open drain pad, if used as general purpose output pads, the value of pad is inverted with the value of GPC.

Signal	I/O	IO Power	Description
I2C_SCL0	IO	VDDIO_VV	I2C0 clock signal line
I2C_SDA0	IO	VDDIO_VV	I2C0 data signal line
I2C_SCL1	IO	VDDIO_VV	I2C1 clock signal line
I2C_SDA1	IO	VDDIO_VV	I2C1 data signal line
PS2_CLK0	IO	VDDIO_VV	PIC0 serial clock line
PS2_DATA0	IO	VDDIO_VV	PIC0 serial data line
PS2_CLK1	IO	VDDIO_VV	PIC1 serial clock line
PS2_DATA1	IO	VDDIO_VV	PIC1 serial data line

IIS/AC97

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XAUBCLK	I	O	O	I	I	GPD[0]	GPD[0]	I2S_SCLK	AC97_BCLK	0
XAUFCLK	I	O	O	O	I	GPD[1]	GPD[1]	I2S_CDCLK	AC97_RESET	0
XAUFSYNC	I	O	O	O	I	GPD[2]	GPD[2]	I2S_LRCK	AC97_SYNC	0
XAUSDI	I	O	I	I	I	GPD[3]	GPD[3]	I2S_SDI	AC97_SDI	0
XAUSDO	I	O	O	O	I	GPD[4]	GPD[4]	I2S_SDO	AC97_SDO	0

Signal	I/O	IO Power	Description
I2S_SCLK	O	VDDIO_VV	IIS serial clock signal
I2S_CDCLK	O	VDDIO_VV	IIS codec main clock
I2S_LRCK	O	VDDIO_VV	IIS word select signal
I2S_SDI	I	VDDIO_VV	IIS serial data input signal
I2S_SDO	O	VDDIO_VV	IIS serial data output signal
AC97_BCLK	I	VDDIO_VV	AC97 link clock
AC97_RESET	O	VDDIO_VV	AC97 cold reset signal
AC97_SYNC	O	VDDIO_VV	AC97 link frame marker and warm reset.
AC97_SDI	I	VDDIO_VV	AC97 serial data input signal
AC97_SDO	O	VDDIO_VV	AC97 serial data output signal

SSI/SD/SDHC/SDIO

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XSSI1RXD	I	O	I	O	I	GPE[0]/EINT4[0]	GPE[0]	MST_SSI1_RXD	SLV_SSI_TXD	0
XSSI1CLK	I	O	O	I	I	GPE[1]/EINT4[1]	GPE[1]	MST_SSI1_CLK	SLV_SSI_CLK	0
XSSI1TXD	I	O	O	I	I	GPE[2]/EINT4[2]	GPE[2]	MST_SSI1_TXD	SLV_SSI_RXD	0
XSSI1CSN	I	O	O	I	I	GPE[3]/EINT4[3]	GPE[3]	MST_SSI1_CSN	SLV_SSI_CSN	0
XSSITXD	I	O	O	--	I	GPE[4]/EINT4[4]	GPE[4]	MST_SSI_TXD	--	0
XSSIRXD	I	O	I	--	I	GPE[5]/EINT4[5]	GPE[5]	MST_SSI_RXD	--	0
XSSICLK	I	O	O	O	I	GPE[6]/EINT4[6]	GPE[6]	MST_SSI_CLK	SRAM_nCS[3]	0
XSSICSN[0]	I	O	O	O	I	GPE[7]/EINT4[7]	GPE[7]	MST_SSI_CSN0	SRAM_nCS[2]	0
XSSICSN[1]	I	O	O	O	O	GPE[8]/EINT4[8]	GPE[8]	MST_SSI_CSN1	SRAM_nBE[0]	3
XSSICSN[2]	I	O	O	O	O	GPE[9]/EINT4[9]	GPE[9]	MST_SSI_CSN2	SRAM_nBE[1]	3
XSSICSN[3]	I	O	O	O	O	GPE[10]/EINT4[10]	GPE[10]	MST_SSI_CSN3	SRAM_nCS[1]	3
XSDCLK	I	O	O	--	I	GPF[0]	GPF[0]	SD0_CLK	--	0
XSDCMD	I	O	IO	--	I	GPF[1]	GPF[1]	SD0_CMD	--	0
XSDDAT0	I	O	IO	--	I	GPF[2]	GPF[2]	SD0_DAT0	--	0
XSDDAT1	I	O	IO	--	I	GPF[3]	GPF[3]	SD0_DAT1	--	0
XSDDAT2	I	O	IO	--	I	GPF[4]	GPF[4]	SD0_DAT2	--	0
XSDDAT3	I	O	IO	--	I	GPF[5]	GPF[5]	SD0_DAT3	--	0
XSD1CLK	I	O	O	--	I	GPO[0]	GPO[0]	SD1_CLK	--	0
XSD1CMD	I	O	IO	--	I	GPO[1]	GPO[1]	SD1_CMD	--	0
XSD1DAT0	I	O	IO	--	I	GPO[2]/EINT4[24]	GPO[2]	SD1_DAT0	--	0
XSD1DAT1	I	O	IO	--	I	GPO[3]/EINT4[25]	GPO[3]	SD1_DAT1	--	0
XSD1DAT2	I	O	IO	--	I	GPO[4]/EINT4[26]	GPO[4]	SD1_DAT2	--	0
XSD1DAT3	I	O	IO	--	I	GPO[5]/EINT4[27]	GPO[5]	SD1_DAT3	--	0
XSD2CLK	I	O	O	O	I	GPO[6]	GPO[6]	SD2_CLK	MST_SSI2_CLK	0
XSD2CMD	I	O	IO	I	I	GPO[7]	GPO[7]	SD2_CMD	MST_SSI2_RXD	0
XSD2DAT0	I	O	IO	O	I	GPO[8]/EINT4[24]	GPO[8]	SD2_DAT0	MST_SSI2_TXD	0
XSD2DAT1	I	O	IO	--	I	GPO[9]/EINT4[25]	GPO[9]	SD2_DAT1	--	0
XSD2DAT2	I	O	IO	--	I	GPO[10]/EINT4[26]	GPO[10]	SD2_DAT2	--	0
XSD2DAT3	I	O	IO	O	I	GPO[11]/EINT4[27]	GPO[11]	SD2_DAT3	MST_SSI2_CSN	0

XEINT3	I	--	--	--	I	GPG[3]/SD0_nCD	--	--	--	--	0
XEINT4	I	--	--	--	I	GPG[4]/SD1_nCD	--	--	--	--	0
XEINT5	I	--	--	--	I	GPG[5]/SD2_nCD	--	--	--	--	0

Signal	I/O	IO Power	Description
MST_SSI_TXD	O	VDDIO_ET	Master SSI0 transmit data line
MST_SSI_RXD	I	VDDIO_ET	Master SSI0 receive data line
MST_SSI_CLK	O	VDDIO_ET	Master SSI0 serial output clock
MST_SSI_CSN[3:0]	O	VDDIO_ET	Master SSI0 slave select output signal
MST_SSI1_TXD	O	VDDIO_ET	Master SSI1 transmit data line
MST_SSI1_RXD	I	VDDIO_ET	Master SSI1 receive data line
MST_SSI1_CLK	O	VDDIO_ET	Master SSI1 serial output clock
MST_SSI1_CSN	O	VDDIO_ET	Master SSI1 slave select output signal
MST_SSI2_TXD	O	VDDIO_SD	Master SSI2 transmit data line
MST_SSI2_RXD	I	VDDIO_SD	Master SSI2 receive data line
MST_SSI2_CLK	O	VDDIO_SD	Master SSI2 serial output clock
MST_SSI2_CSN	O	VDDIO_SD	Master SSI2 slave select output signal
SLV_SSI_TXD	O	VDDIO_ET	Slave SSI transmit data line
SLV_SSI_RXD	I	VDDIO_ET	Slave SSI receive data line
SLV_SSI_CLK	I	VDDIO_ET	Slave SSI serial input clock
SLV_SS_CSN	I	VDDIO_ET	Slave SSI select input signal, active low
SD0_CLK	O	VDDIO_SD	SD/SDHC/SDIO0 bus clock
SD0_CMD	IO	VDDIO_SD	SD/SDHC/SDIO0 command/response signal
SD0_DAT[3:0]	IO	VDDIO_SD	SD/SDHC/SDIO0 data bus
SD0_nCD	I	VDDIO_VV	SD/SDHC/SDIO0 card detect signal
SD1_CLK	O	VDDIO_SD	SD/SDHC/SDIO1 bus clock
SD1_CMD	IO	VDDIO_SD	SD/SDHC/SDIO1 command/response signal
SD1_DAT[3:0]	IO	VDDIO_SD	SD/SDHC/SDIO1 data bus
SD1_nCD	I	VDDIO_VV	SD/SDHC/SDIO1 card detect signal
SD2_CLK	O	VDDIO_SD	SD/SDHC/SDIO2 bus clock
SD2_CMD	IO	VDDIO_SD	SD/SDHC/SDIO2 command/response signal
SD2_DAT[3:0]	IO	VDDIO_SD	SD/SDHC/SDIO2 data bus
SD2_nCD	I	VDDIO_VV	SD/SDHC/SDIO2 card detect signal

USB Host/OTG

Signal	I/O		Function		
	0	1	0	1	Default
XUSB_S1DP	IO	IO	XUSB_ODP	XUSB_H2DP	0
XUSB_S1DM	IO	IO	XUSB_ODM	XUSB_H2DM	0
XUSB_OID	IO	IO	XUSB_OID	XUSB_ID2	0
XUSB_OVRES	IO	IO	XUSB_OVRES	XUSB_VRES2	0
VBUS_USBO	IO	IO	VBUS_USBO	VBUS_USBH2	0

Note: USB OTG PHY can be multi-used with port2 of USB Host, the control bit is USBOTG_PHYCFG in MEM_CFG register of system management.

Signal	I/O	IO Power	Description
XUSB_H0DP	IO	VDDA_USBH0	USB Host Port 0 DP
XUSB_H0DM	IO	VDDA_USBH0	USB Host Port 0 DM
XUSB_ID0	IO	VDDA_USBH0	USB Host Port 0 ID, Connect to the ID pin on the Mini-type connector
XUSB_VRES0	IO	VDDA_USBH0	USB Host Port 0 VRES, Connect to an external 8.2K Ohm resistor for band-gap reference circuit
VBUS_USBH0	IO	VDDA_USBH0	USB Host Port 0 VBUS, Connect to the VBUS pin on the connector

XUSB_H1DP	IO	VDDA_USBH1	USB Host Port 1 DP
XUSB_H1DM	IO	VDDA_USBH1	USB Host Port 1 DM
XUSB_ID1	IO	VDDA_USBH1	USB Host Port 1 ID, Connect to the ID pin on the Mini-type connector
XUSB_VRES1	IO	VDDA_USBH1	USB Host Port 1 VRES, Connect to an external 8.2K Ohm resistor for band-gap reference circuit
VBUS_USBH1	IO	VDDA_USBH1	USB Host Port 1 VBUS, Connect to the VBUS pin on the connector
XUSB_H2DP	IO	VDDA_USBO	USB Host Port 2 DP
XUSB_H2DM	IO	VDDA_USBO	USB Host Port 2 DM
XUSB_ID2	IO	VDDA_USBO	USB Host Port 2 ID, Connect to the ID pin on the Mini-type connector
XUSB_VRES2	IO	VDDA_USBO	USB Host Port 2 VRES, Connect to an external 8.2K Ohm resistor for band-gap reference circuit
VBUS_USBH2	IO	VDDA_USBO	USB Host Port 2 VBUS, Connect to the VBUS pin on the connector
XUSB_S1DP	IO	VDDA_USBO	USB OTG DP
XUSB_S1DM	IO	VDDA_USBO	USB OTG DM
XUSB_OID	IO	VDDA_USBO	USB OTG ID, Connect to the ID pin on the Mini-type connector
XUSB_OVRES	IO	VDDA_USBO	USB OTG VRES, Connect to an external 8.2K Ohm resistor for band-gap reference circuit
VBUS_USBO	IO	VDDA_USBO	USB OTG VBUS, Connect to the VBUS pin on the connector
XDRVVBUS	O	VDDIO_ET	Drive Vbus for Off-Chip Charge Pump

1.5.3 Parallel Communication

Ethernet MAC/KeyPad

Signal	I/O					Function					
	0	1	2	3	Default	0	1	2	3	Default	
XETHTCLK	I	O	I	IO	I	GPK[0]/EINT5[0]	GPK[0]	ETH_TXC	KB1_COL[0]	0	
XETHRCLK	I	O	I	IO	I	GPK[1]/EINT5[1]	GPK[1]	ETH_RXC	KB1_COL[1]	0	
XETHMDC	I	O	O	IO	I	GPK[2]/EINT5[2]	GPK[2]	ETH_MDC	KB1_COL[2]	0	
XETHMDIO	I	O	IO	IO	I	GPK[3]/EINT5[3]	GPK[3]	ETH_MDIO	KB1_COL[3]	0	
XETHTXEN	I	O	O	IO	I	GPK[4]/EINT5[4]	GPK[4]	ETH_TXEN	KB1_COL[4]	0	
XETHTXD[3:0]	I	O	O	I	I	GPK[8:5]/EINT5[8:5]	GPK[8:5]	ETH_TXD[3:0]	KB1_ROW[7:4]	0	
XETHRXDV	I	O	I	IO	I	GPK[9]/EINT5[9]	GPK[9]	ETH_RXDV	KB1_COL[5]	0	
XETHRXER	I	O	I	IO	I	GPK[10]/EINT5[10]	GPK[10]	ETH_RXER	KB1_COL[6]	0	
XETHRXD[3:0]	I	O	I	I	I	GPK[14:11]/EINT5[14:11]	GPK[14:11]	ETH_RXD[3:0]	KB1_ROW[3:0]	0	
XETHCRS	I	O	I	IO	I	GPK[15]/EINT5[15]	GPK[15]	ETH_CRS	KB1_COL[7]	0	
XETHCOL	I	O	I	IO	I	GPJ[8]/EINT6[26]	GPJ[8]	ETH_COL	KB1_COL[8]	0	

Signal	I/O	IO Power	Description
ETH_TXC	I	VDDIO_ET	**Ethernet PHY transmit clock**. This pin provides a continuous clock as a timing reference for ETH_TXD[3:0] and ETH_TXEN.
ETH_RXC	I	VDDIO_ET	**Ethernet PHY receive clock**. This pin provides a continuous clock reference for ETH_RXDV and ETH_RXD[3:0] signals. ETH_RXC is 25MHz in the 100Mbps mode and 2.5MHz in the 10Mbps mode.
ETH_MDC	O	VDDIO_ET	**Management Data Clock**: This pin provides a clock synchronous to MDIO, which may be asynchronous to the transmit TXC and receive RXC clocks. The clock rate can be up to 2.5MHz.
ETH_MDIO	IO	VDDIO_ET	**Management Data Input/Output**: This pin provides the bi-directional signal used to transfer management information.
ETH_TXEN	O	VDDIO_ET	**Transmit Enable**: The input signal indicates the presence of a valid nibble data on ETH_TXD[3:0].
ETH_TXD[3:0]	O	VDDIO_ET	**Transmit Data:** MAC will source ETH_TXD[3:0] synchronous with ETH_TXC when ETH_TXEN is asserted.
ETH_RXDV	I	VDDIO_ET	**Receive Data Valid:** This pin's signal is asserted high when received data is present on the ETH_RXD[3:0] lines; the signal is deasserted at the end of the packet. The signal is valid on the rising of the RXC.

ETH_RXER	I	VDDIO_ET	**Receive error:** if any 5B decode error occurs, such as invalid J/K, T/R, invalid symbol, this pin will go high.
ETH_RXD[3:0]	I	VDDIO_ET	**Receive Data:** These are the four parallel receive data lines aligned on the nibble boundaries driven synchronously to the RXC for reception by the external physical unit (PHY).
ETH_CRS	I	VDDIO_ET	**Carrier Sense:** This pin's signal is asserted high if the media is not in IDEL state.
ETH_COL	I	VDDIO_ET	**Collision Detected:** COL is asserted high when a collision is detected on the media.

Signal	I/O	IO Power	Description
KB_ROW[7:0]	I	VDDIO_ET	KeyPad row input line use KB0_ROW[7:0]. **Note:** Bit[0] of register PAD_CFG in system management need to set to 0, default value.
	I	VDDIO_ET	KeyPad row input line use KB1_ROW[7:0]. **Note**: Bit[0] of register PAD_CFG in system management need to set to 1.
KB_COL[17:0]	IO	VDDIO_ET	KB0_COL[17:0], KeyPad column output line, using this port, 8x18 keypad can be supported.
	IO	VDDIO_ET	KB1_COL[8:0], KeyPad column output line, using this port, 8x9 keypad can be supported.
	IO	VDDIO_ET	KB1_COL[8:0]/KB2_COL[17:9], KeyPad column output line, using this port, 8x18 keypad can be supported

External Interrupt

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XEINT0	I	--	--	--	I	GPG[0]	--	--	--	0
XEINT1	I	--	--	--	I	GPG[1]	--	--	--	0
XEINT2	I	--	--	--	I	GPG[2]	--	--	--	0
XEINT3	I	--	--	--	I	GPG[3]/SD0_nCD	--	--	--	0
XEINT4	I	--	--	--	I	GPG[4]/SD1_nCD	--	--	--	0
XEINT5	I	--	--	--	I	GPG[5]/SD2_nCD	--	--	--	0

Signal	I/O	IO Power	Description
XEINT[5:0]	I	VDDIO_VV	External interrupts, each interrupt support five trigger type, including high level, low level, rising edge triggered, falling edge triggered and Both edge triggered.

1.5.4 Multimedia Communication

Camera/DVB-TS

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XCICLK	I	O	O	--	I	GPL[0]/EINT6[27]	GPL[0]	CAM_OCLK	--	0
XCIHREF	I	O	I	I	I	GPL[1]/EINT6[28]	GPL[1]	CAM_HREF	TS_DE	0
XCIPCLK	I	O	I	I	I	GPL[2]/EINT6[29]	GPL[2]	CAM_PCLK	TS_CLK	0
XCIVSYNC	I	O	I	I	I	GPL[3]/EINT6[30]	GPL[3]	CAM_VSYNC	TS_SYNC	0
XCINRST	I	O	O	--	I	GPL[4]/EINT6[31]	GPL[4]	CAM_RESET	--	0
XCIYDATA[7:0]	I	O	I	I	I	GPL[12:5]/EINT5[31:24]	GPL[12:5]	CAM_DATA[7:0]	TS_DATA[7:0]	0
XIRSDBW	I	O	--	IO	I	GPB[4]/EINT2[4]/ CAM_FIELD	GPB[4]	--	KB2_COL[9]	0

Signal	I/O	IO Power	Description
CAM_OCLK	O	VDDIO_CI	Sensor reference clock.
CAM_HREF	I	VDDIO_CI	Horizontal reference signal.
CAM_PCLK	I	VDDIO_CI	Sensor pixel clock.
CAM_VSYNC	I	VDDIO_CI	Vertical synchronization signal.
CAM_RESET	O	VDDIO_CI	Sensor reset signal.
CAM_DATA[7:0]	I	VDDIO_CI	Sensor data bus.
CAM_FIELD	I	VDDIO_ET	Interlace or progressive indicate signal
TS_DE	I	VDDIO_CI	DVB-TS data enable signal, active high
TS_CLK	I	VDDIO_CI	DVB-TS data clock.
TS_SYNC	I	VDDIO_CI	DVB-TS data frame sync signal.
TS_DATA[7:0]	I	VDDIO_CI	DVB-TS data bus.

1.5.5 Display Control

TFT LCD/TV/i80 Interface

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XVVD[2:0]	I	O	IO	--	I	GPM[2:0]/EINT5[18:16]	GPM[2:0]	RGB_VD[2:0]/ TV_DAT[2:0]/ I80_DAT[2:0]	--	0
XVVD[7:3]	I	O	IO	--	I	GPM[7:3]	GPM[7:3]	RGB_VD[7:3]/ TV_DAT[7:3]/ I80_DAT[7:3]	--	0
XVVD[9:8]	I	O	IO	--	I	GPM[9:8]/EINT5[20:19]	GPM[9:8]	RGB_VD[9:8]/ TV_DAT[9:8]/ I80_DAT[9:8]	--	0
XVVD[15:10]	I	O	IO	--	I	GPM[15:10]	GPM[15:10]	RGB_VD[15:10]/ TV_DAT[15:10]/ I80_DAT[15:10]	--	0
XVVD[18:16]	I	O	IO	--	I	GPN[2:0]/EINT5[23:21]	GPN[2:0]	RGB_VD[18:16]/ TV_DAT[18:16]/ I80_DAT[17:16]	--	0
XVVD[23:19]	I	O	O	--	I	GPN[7:3]	GPN[7:3]	RGB_VD[23:19]/ TV_DAT[23:19]	--	0
XVVCLK	I	O	O	--	I	GPN[8]	GPN[8]	RGB_VCLK/ TV_PCLK/ I80_NWR	--	0
XVVSYNC	I	O	O	--	I	GPN[9]	GPN[9]	RGB_VSYNC/ TV_VSYNC/ I80_NCS1	--	0
XVHSYNC	I	O	O	--	I	GPN[10]	GPN[10]	RGB_HSYNC/ TV_HSYNC/ I80_NCS0	--	0
XVVDEN	I	O	O	--	I	GPN[11]	GPN[11]	RGB_VDEN/ TV_HREF/ I80_NRS	--	0
XVPWREN	I	O	O	--	I	GPN[12]	GPN[12]	RGB_PWREN/ I80_NRD	--	0
XTOUT2	I	O	O	O	I	GPF[8]/EINT4[22]	GPF[8]	PWM_TOUT2	TV_FIELD	0
XTOUT3	I	O	O	I	I	GPF[9]/EINT4[23]	GPF[9]	PWM_TOUT3	TV_RCLK	0

Signal	I/O	IO Power	Description
RGB_VD[23:0]	O	VDDIO_VV	RGB data bus, the order of RGB is configurable.
RGB_VCLK	O	VDDIO_VV	RGB data clock, polarity is configurable.
RGB_VSYNC	O	VDDIO_VV	RGB vertical sync signal, polarity is configurable.
RGB_HSYNC	O	VDDIO_VV	RGB horizontal sync signal, polarity is configurable.
RGB_VDEN	O	VDDIO_VV	RGB data enable signal, polarity is configurable.
RGB_PWREN	O	VDDIO_VV	Display panel power enable signal.
TV_DAT[23:0]	O	VDDIO_VV	TV data bus, the order of YCbCr/RGB is configurable.
TV_PCLK	O	VDDIO_VV	TV pixel clock, polarity is configurable.
TV_VSYNC	O	VDDIO_VV	TV vertical sync signal, polarity is configurable.
TV_HSYNC	O	VDDIO_VV	TV horizontal sync signal, polarity is configurable.
TV_HREF	O	VDDIO_VV	TV data sync signal, polarity is configurable.
TV_FIELD	O	VDDIO_VV	TV field sync signal, polarity is configurable.
TV_RCLK	I	VDDIO_VV	TV interface input reference clock.
I80_DAT	IO	VDDIO_VV	I80 data bus, support multi-format of data bus.
I80_NWR	O	VDDIO_VV	I80 write strobe signal, active low.
I80_NRD	O	VDDIO_VV	I80 read strobe signal, active low.
I80_NRS	O	VDDIO_VV	I80 register select signal.
I80_NCS0	O	VDDIO_VV	I80 chip 0 select signal.
I80_NCS1	O	VDDIO_VV	I80 chip 1 select signal.

1.5.6 System Management

Reset

Signal	I/O	IO Power	Description
XNRESET	I	VDDIO_RTC	XNRESET suspends any operation in progress and places IMAPX210 into a known reset state.
XNWRESET	I	VDDIO_VV	XnWRESET is used to initialize IMAPx210 and preserve current hardware status.
XNRSTOUT	I	VDDIO_VV	For external device reset control.

Clock

Signal	I/O	IO Power	Description
XRTCXTI	I	VDDIO_OSC	32KHz crystal input for RTC.
XRTCXTO	O	VDDIO_OSC	32KHz crystal output for RTC.
XXTI	I	VDDIO_VV	Crystal Input for internal OSC circuit.
XXTO	O	VDDIO_VV	Crystal output for internal OSC circuit.
XEXTCLK	I	VDDIO_VV	External clock source

JTAG

Signal	I/O	IO Power	Description
XJCNTRST	I	VDDIO_VV	XJCnTRST (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, XJCnTRST pin must be at L or low active pulse.
XJCTCK	I	VDDIO_VV	XJCTCK (TAP Controller Clock) provides the clock input for the JTAG logic.
XJCTMS	I	VDDIO_VV	XJCTMS (TAP Controller Mode Select) controls the sequence of the TAP controller's states.
XJCTDI	I	VDDIO_VV	XJCTDI (TAP Controller Data Input) is the serial input for test instructions and data.
XJCTDO	O	VDDIO_VV	XJCTDO (TAP Controller Data Output) is the serial output for test instructions and data.
XJCRTCK	O	VDDIO_VV	XJCRTCK (TAP Controller Returned Clock) provides the clock output for the JTAG logic.
XJDMODE	I	VDDIO_VV	JTAG mode signal, must be always connected to GND.

PWM Timer/System Clock

Signal	I/O					Function				
	0	1	2	3	Default	0	1	2	3	Default
XTOUT0	I	O	O	O	I	GPF[6]/EINT4[20]	GPF[6]	PWM_TOUT0	SYS_CLKO0	0
XTOUT1	I	O	O	O	I	GPF[7]/EINT4[21]	GPF[7]	PWM_TOUT1	SYS_CLKO1	0
XTOUT2	I	O	O	O	I	GPF[8]/EINT4[22]	GPF[8]	PWM_TOUT2	TV_FIELD	0
XTOUT3	I	O	O	I	I	GPF[9]/EINT4[23]	GPF[9]	PWM_TOUT3	TV_RCLK	0

Signal	I/O	IO Power	Description
PWM_TOUT0	O	VDDIO_VV	PWM timer0 output
PWM_TOUT1	O	VDDIO_VV	PWM timer1 output
PWM_TOUT2	O	VDDIO_VV	PWM timer2 output
PWM_TOUT3	O	VDDIO_VV	PWM timer3 output
SYS_CLKO0	O	VDDIO_VV	Chip reference output clock0
SYS_CLKO1	O	VDDIO_VV	Chip reference output clock1

MISC

Signal	I/O	IO Power	Description
XBPARA[4:0]	I	VDDIO_VV	Boot mode selection
XTESTMODE	I	VDDIO_VV	Test mode signal, must be always connected to GND.
XPOWKEY	I	VDDIO_RTC	Chip power on pad.
XPOWIND	I	VDDIO_RTC	External battery power indication signal
XPOWON	O	VDDIO_RTC	Chip internal logic power on indication signal.

1.6 Power Supply Groups

VDD

Signal	I/O	Description	Number	Voltage (V)
VDDIO_DDR	P	DRAM IO digital Power	15	DDR2=1.8, MDDR=1.8, DDR=2.5
VDDR_DDR	P	DRAM IO Receiver reference voltage	4	DDR2=0.9,DDR=1.25,MDDR:connect to GND
VDDP_DDR	P	DRAM IO core power	5	1.2
VDDIO_VV	P	IO Power for VV field.	6	1.8~3.3
VDDIO_SD	P	IO Power for SD field.	1	1.8~3.3
VDDIO_ET	P	IO Power for ET field.	4	1.8~3.3
VDDIO_CI	P	IO Power for CI field.	1	1.8~3.3
AHVDD_APLL	P	APLL analog power	1	2.5
AHVDD_DPLL	P	DPLL analog power	1	2.5
AHVDD_EPLL	P	EPLL analog power	1	2.5
VDDIO_RTC	P	RTC domain IO Power	1	3.3
VDDIO_OSC	P	32KHz OSC IO Power	1	3.3
VDDA_USBH0	P	USB Host port0 PHY analog power	1	3.3
VDDA_USBH1	P	USB Host port1 PHY analog power	2	3.3
VDDA_USBO	P	USB OTG PHY analog power	1	3.3
DVDD_APLL	P	APLL digital power	1	1.2
DVDD_DPLL	P	DPLL digital power	1	1.2
DVDD_EPLL	P	EPLL digital power	1	1.2
VDD_RTC	P	RTC domain digital power	1	1.2
AVDD_OSC	P	32KHz OSC analog power	1	1.2
VDD_OSC	P	32KHz OSC digital power	1	1.2
VDDL_USBH0	P	USB Host port0 PHY digital power	1	1.2
VDDL_USBH1	P	USB Host port1 PHY digital power	1	1.2

VDDL_USBO	P	USB OTG PHY digital power	1	1.2
VDDC	P	Chip core digital power	11	1.2

VSS

Signal	I/O	Number	Description
AHVSS_APLL	P	1	APLL analog ground
AHVSS_DPLL	P	1	DPLL analog ground
AHVSS_EPLL	P	1	EPLL analog ground
DVSS_APLL	P	1	APLL digital ground
DVSS_DPLL	P	1	DPLL digital ground
DVSS_EPLL	P	1	EPLL digital ground
VSDL_USBH0	P	1	USB Host port0 PHY digital ground
VSDL_USBH1	P	1	USB Host port1 PHY digital ground
VSDL_USBO	P	1	USB OTG PHY digital ground
VSSA_USBH0	P	1	USB Host port0 PHY analog ground
VSSA_USBH1	P	1	USB Host port1 PHY analog ground
VSSA_USBO	P	1	USB OTG PHY analog ground
AVSS_OSC	P	1	32KHz OSC analog ground
GND	P	47	Chip digital ground

Note: GPA, GPB, and so on stands for each general purpose input/output port group. EINT1, EINT2 and etc. stands for each extend interrupt group. Please refer to GPIO module for details.

2 Memory Map

IMAPx210 Memory Map

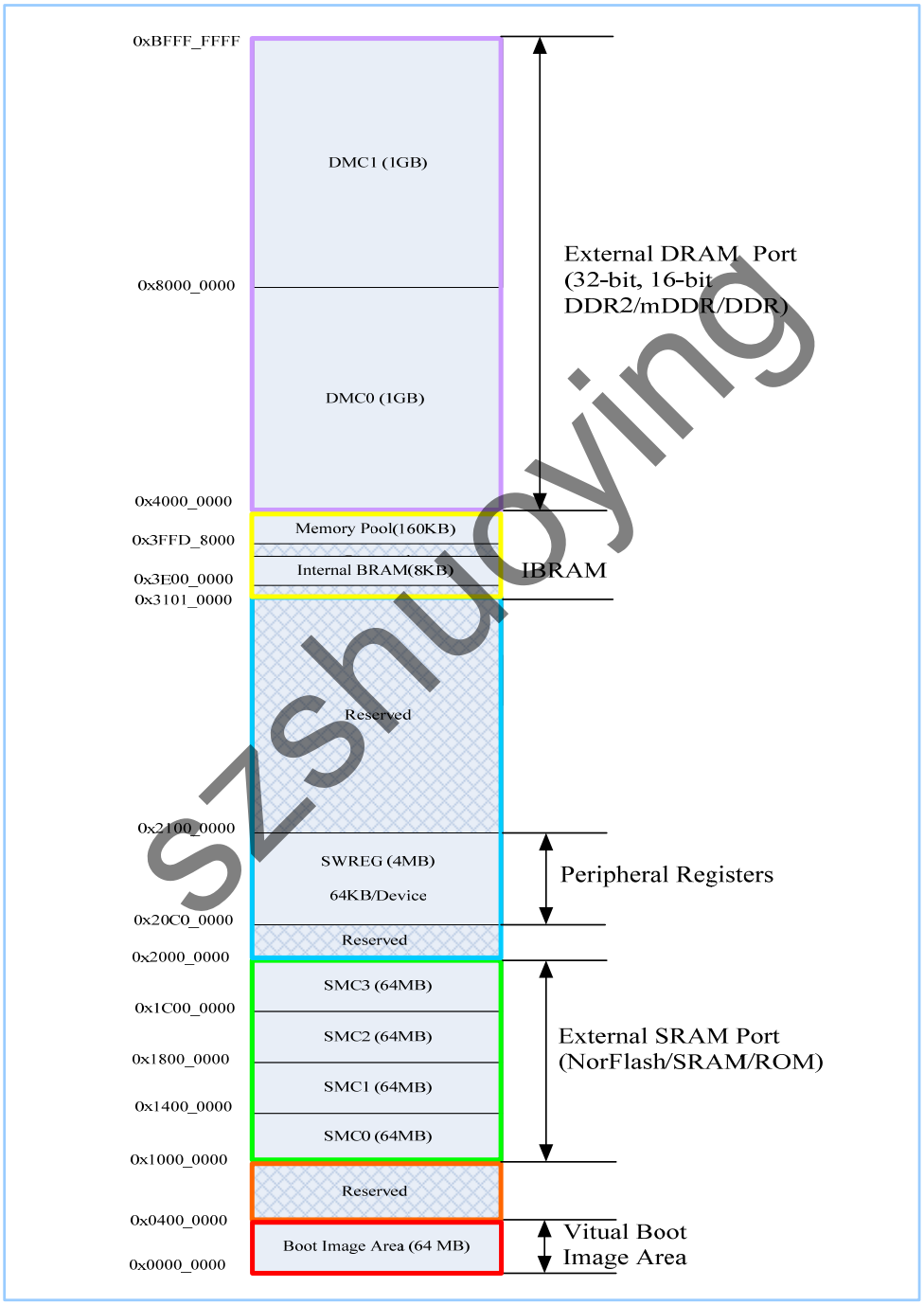


Figure 2-1: IMAPx210 Memory Map

IMAPx210 supports 32-bit physical address field and that address field can be separated into three parts, the first part is for memory, the second part is for peripheral and the last part is for boot image area.

Address range of boot image area is from 0x0000_0000 to 0x03FF_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.

Peripheral Register Address Space

Table 2-1: Peripheral Register Address Space

No	Address Range (bytes)		Size (bytes)	Description
1	0x20C0_0000	0x20C0_FFFF	64KB	System Management
2	0x20C1_0000	0x20C1_FFFF	64KB	SRAM controller
3	0x20C2_0000	0x20C2_FFFF	64KB	DRAM controller
4	0x20C3_0000	0x20C3_FFFF	64KB	Reserved
5	0x20C4_0000	0x20C4_FFFF	64KB	DMA controller
6	0x20C5_0000	0x20C5_FFFF	64KB	NandFlash controller
7	0x20C6_0000	0x20C6_0FFF	4KB	SD/SDHC/SDIO-0 controller
8	0x20C6_1000	0x20C6_1FFF	4KB	SD/SDHC/SDIO-1 controller
9	0x20C6_2000	0x20C6_2FFF	4KB	SD/SDHC/SDIO-2 controller
10	0x20C7_0000	0x20C7_FFFF	64KB	CF-ATA Interface
11	0x20C8_0000	0x20C8_FFFF	64KB	USB2.0 Host
12	0x20C9_0000	0x20C9_FFFF	64KB	USB2.0 OTG
13	0x20CA_0000	0x20CA_FFFF	64KB	MemoryPool
14	0x20CB_0000	0x20CB_FFFF	64KB	Interrupt Controller
15	0x20CC_0000	0x20CC_FFFF	64KB	Camera interface
16	0x20CD_0000	0x20CD_FFFF	64KB	display controller
17	0x20CE_0000	0x20CE_FFFF	64KB	Ethernet MAC
18	0x20F0_0000	0x20FF_FFFF	1MB	2/3D Graphics
19	0x20D0_0000	0x20D0_FFFF	64KB	Video Encoder
20	0x20D1_0000	0x20D1_FFFF	64KB	Video Decoder
21	0x20D3_0000	0x20D3_FFFF	64KB	CMN-Timer
22	0x20D7_0000	0x20D7_FFFF	64KB	PWM-Timer
23	0x20D9_0000	0x20D9_FFFF	64KB	WatchDog Timer
24	0x20DA_0000	0x20DA_0FFF	4KB	I2C-0
25	0x20DA_1000	0x20DA_1FFF	4KB	I2C-1
26	0x20DC_0000	0x20DC_FFFF	64KB	RTC
27	0x20DD_0000	0x20DD_FFFF	64KB	IIS
28	0x20DE_0000	0x20DE_FFFF	64KB	AC97
29	0x20DF_0000	0x20DF_0FFF	4KB	MST-SSI-0
30	0x20DF_1000	0x20DF_1FFF	4KB	MST-SSI-1
31	0x20DF_2000	0x20DF_2FFF	4KB	MST-SSI-2
32	0x20DF_3000	0x20DF_3FFF	4KB	SLV-SSI
33	0x20DF_4000	0x20DF_4FFF	4KB	SPI
34	0x20E1_0000	0x20E1_FFFF	64KB	GPIO
35	0x20E2_0000	0x20E2_0FFF	4KB	UART-0
36	0x20E2_1000	0x20E2_1FFF	4KB	UART-1
37	0x20E2_2000	0x20E2_2FFF	4KB	UART-2
38	0x20E2_3000	0x20E2_3FFF	4KB	UART-3
39	0x20E6_0000	0x20E6_FFFF	64KB	KeyPad
40	0x20E7_0000	0x20E7_0FFF	4KB	PS2-0
41	0x20E7_1000	0x20E7_1FFF	4KB	PS2-1

Memory Address Space

Memory space is separated into four areas, IBRAM, MemoryPool, static memory area, and dynamic memory area.

Address range of dynamic memory area is from 0x4000_0000 to 0xBFFF_FFFF. Dynamic memory area consists of two memory chips, i.e. DMC0 and DMC1. Start address for each chip select is configurable, and depend on the external DRAM size.

Address range of static memory area is from 0x1000_0000 to 0x1FFF_FFFF and separated into four memory chips. SROM, SRAM, NorFlash and asynchronous NOR interface device can be accessed by this address area. Each area stands for a chip select, for example, address range from 0x1000_0000 to 0x13FF_FFFF stands for SRAM_nCS[0]. Start address for each chip is fixed.

IBRAM is internal SRAM, address space range from 0x3E00_0000 to 0x3E00_1FFF. When booting from NandFlash, IBRAM is used for saving bootloader and automatically mapped onto boot image area. After NAND flash booting IBRAM can be used for another purpose.

MemoryPool is also internal SRAM, please refer to Memory Pool controller for more details.

Boot Mode

In IMAPx210, two types of boot mode are supported. One is from NorFlash, the other is from NandFlash. Boot Mode is configured by pin XBPARA[4:0]. The type of NandFlash for boot is also depend on pin XBPARA[4:0].

When booting from NorFlash, NorFlash should be connected to SMC0 bank. SMC0 will be remapped onto boot image area.

When booting from NandFlash, the FSM in NandFlash controller firstly load the first 8KB data in NandFlash into IBRAM, which is mapped onto boot image area. Through resetting CPU, the bootloader program get access to control CPU. 8KB IBRAM buffer can be used for another purpose after NAND flash booting.

In any case, 8KB IBRAM, which is located in 0x3E00_0000 to 0x3E00_1FFF, can be manually remapped to boot image area (i.e. 0x0000_0000 to 0x0000_1FFF). You can set register BOOT_SWAP[0](register address is 0x20C0_0108) high to implement it.

Note: Boot image area is virtual address space, only CPU can access.

Table 2-2: Boot Mode Configuration

XBPARA[1:0] (Flash Type)	XBPARA[2] (Nand Page)	XBPARA[3] (Nand Address Cycle)	XBPARA[4] (Nand Bus Width)
00 : normal NandFlash	0 : 256 words 1: 512 bytes	0: 3 address cycle 1: 4 address cycle	Nandflash: 0: 8-bit-width 1:16-bit-width
01 : advanced NandFlash	0: 1K words 1: 2K bytes	0: 4 address cycle 1: 5 address cycle	
10 : large NandFlash	0: 2K words 1: 4K bytes	0: 4 address cycle 1: 5 address cycle	
11: NorFlash	0	0	0

3 System Controller

3.1 Overview

3.1.1 Clock Generator

- Include three on-chip PLLs called main PLL (APLL) and DPLL and extra PLL (EPLL). APLL generates the system reference clock, DPLL generates the asynchronous clock for CPU, and EPLL generates the clocks for the special functional blocks such as USB, IIS, Camera I/F, OSD, LCD and etc.
- Independent clock ON/OFF control to reduce power consumption for all modules except CPU, including VENC, VDEC, GPU and MEMPL.
- Support IDLE mode to turn ON/OFF of CPU clock. CPU clock is turned off by software and turned on by IDLE wake-up mechanism.

3.1.2 Reset Controller

IMAPx210 has five types of reset signals and SYSCON can place the system into one of five 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.
- Warm reset: It is generated by asserting XnWRESET. XnWRESET is used to initialize IMAPx210 and preserve current hardware status.
- 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, watchdog timer monitors internal hardware status and generates reset signal to escape from this status.
- Software reset: It is generated by setting SW_RESET.
- Wakeup reset: It is generated when IMAPx210 wake up from SLEEP mode. Since internal hardware states are not available any more after SLEEP mode, they must be initialized.

3.1.3 Power Management

- Four controllable power domains
- Clock-off control for individual module.
- Support four power-down modes, IDLE, STOP, SLEEP, and SHUTDOWN to optimize the power dissipation.
- Wake-up by one of external interrupt (STOP and SLEEP mode), RTC alarm (STOP and SLEEP mode), KeyPad (STOP mode), XPowKey (SHUTDOWN).

3.2 Function Description

3.2.1 Clock Architecture

IMAPx210 has three PLL's which are APLL for CPU operating clock, DPLL for CPU synchronous clock, and EPLL for special purpose. The operating clocks are divided into three groups. The first is CPU clock, which are used for operating AXI, AHB, and APB bus operation, which is generated from APLL. DPLL generates the clocks for asynchronous clock for CPU or special functional blocks for peripheral. The last group is generated from EPLL. Mainly, the generated clocks are used for peripheral IP's, i.e., UART, IIS, IIC, USB host, USB slave, Camera, and etc.

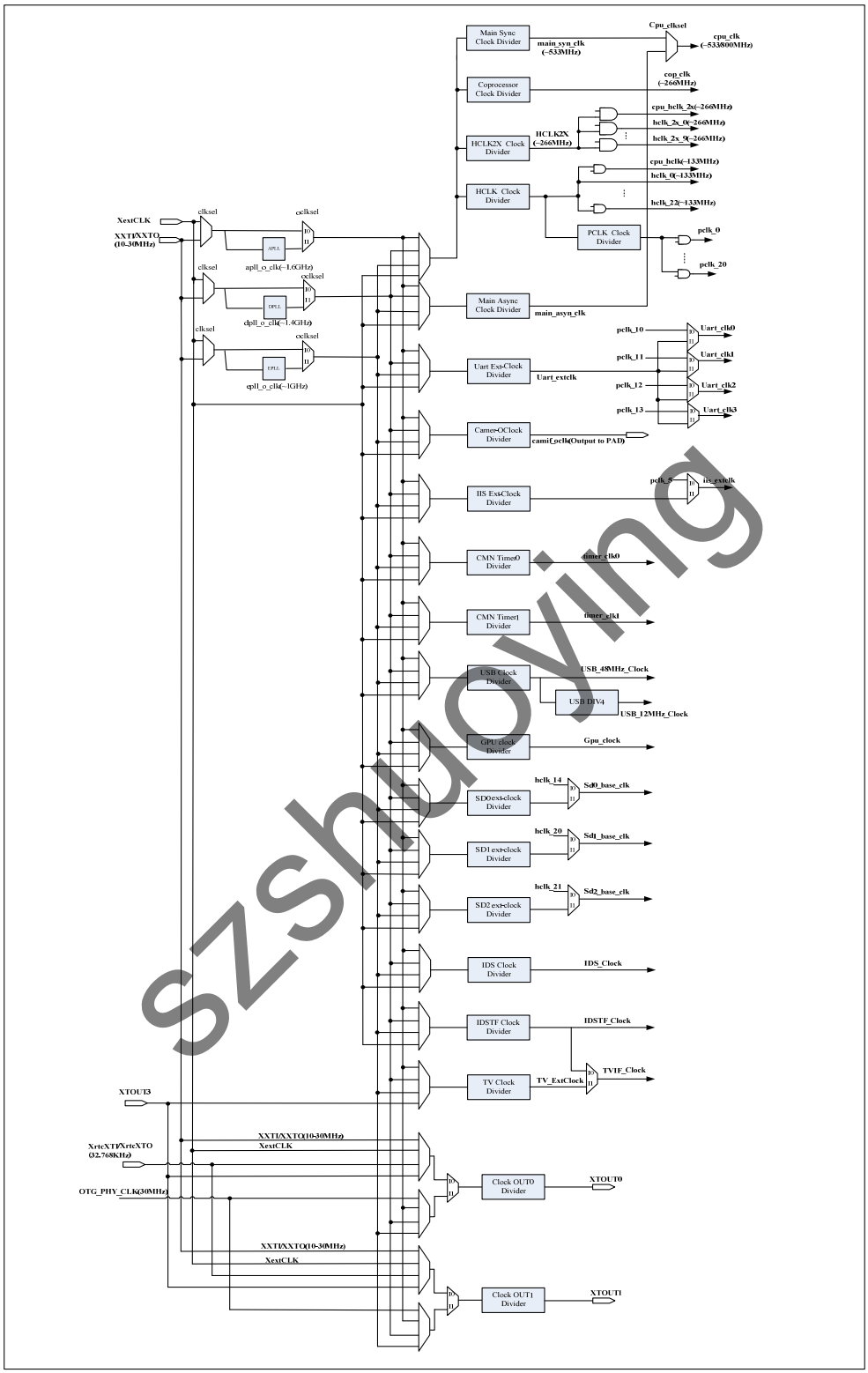


Figure 3-1: System Manager Module Clock Generation

3.2.1.1 Phase Locked Loop (PLL)

Three PLL's within IMAPx210, APLL, DPLL, and EPLL, synchronizes an output signal with a reference input clock in operating frequency and phase.

3.2.1.2 Change PLL Setting In Normal Operation

During the operation of IMAPx210 in NORMAL mode, if the user wants to change the frequency need carry out following steps.

1) Change oclk_sel to reference clock.
2) Powerdown pll by clear pll enable bit [31] in corresponding PLL configuration register.
3) Change PLL configuration
4) Stay in power down mode more than 1 us.
5) Poweron PLL by setting pll enable bit[31] in corresponding PLL configuration register.
6) Wait for PLL locked
7) Set oclk_sel to pll output clock.

3.2.1.3 CPU and AXI/AHB/APB Bus Clock Generation

CPU of IMAPx210 runs at up to 1 GHz. The operating frequency can be controlled by the internal clock divider, DIV_{cpu}, without any change of PLL frequency. The divider ratio varies from 1 to 12. CPU decreases the operating speed to reduce power dissipation.

IMAPx210 consists of AXI bus, AHB bus, and APB bus to optimize the performance requirements. Internal module is connected to appropriate bus systems to meet their I/O bandwidth and operating performance. When they are attached to AHB bus or AXI bus, the operating speed can be up to 266MHz. While they are attached to APB bus, the maximum operating speed can be up to 133MHz. Moreover, the bus speed between AXI, AHB and APB has high dependency to synchronize data transmission.

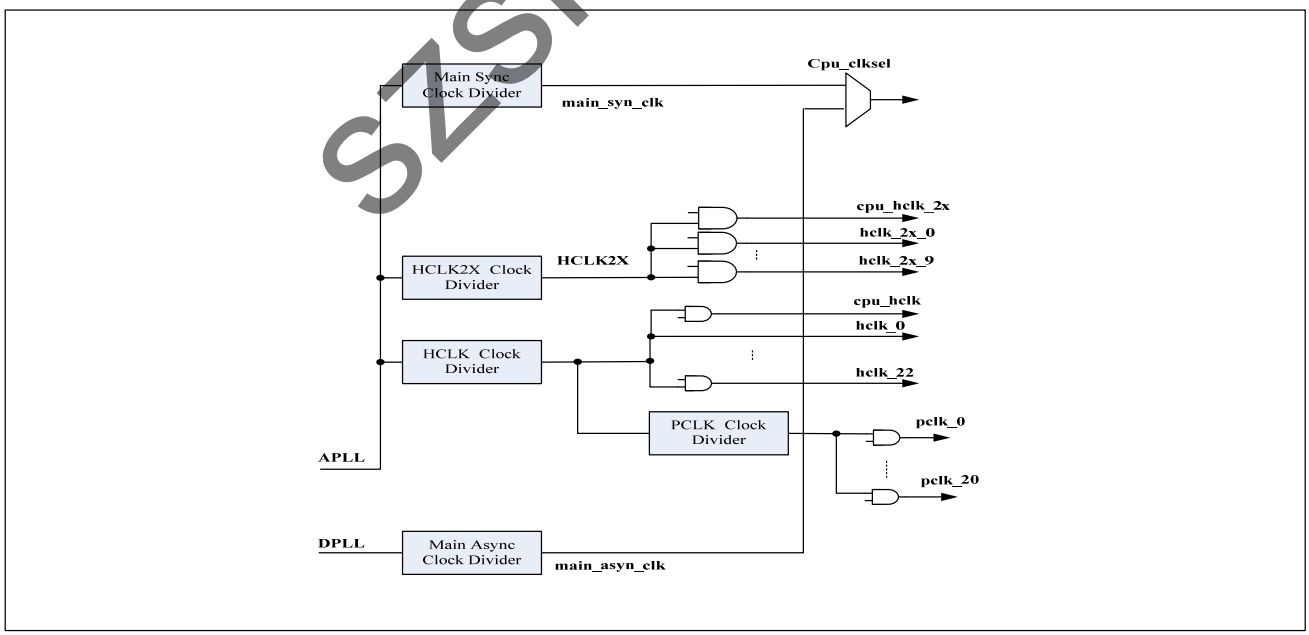


Figure 3-2: CPU and Bus Clock Generation

HCLK2X clocks are supplied to MDDR/DDR/DDR2 controller of IMAPx210. The operating speed can be up to 266MHz to send and to receive data through MDDR/DDR/DDR2 controller.

All AHB bus clocks are generated from DIV_{HCLK} clock divider. The generated clocks can be masked independently to reduce redundant power dissipation. HCLK_MASK register controls the mask operation HCLK.

Low-speed interconnection peripheral 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_MASK register. As described, the frequency ratio between AHB clock and APB clock must be an integer value.

3.2.1.4 USB Clock Generation

Figure 3-3 shows the clock generator for USB OTG and USB host blocks.

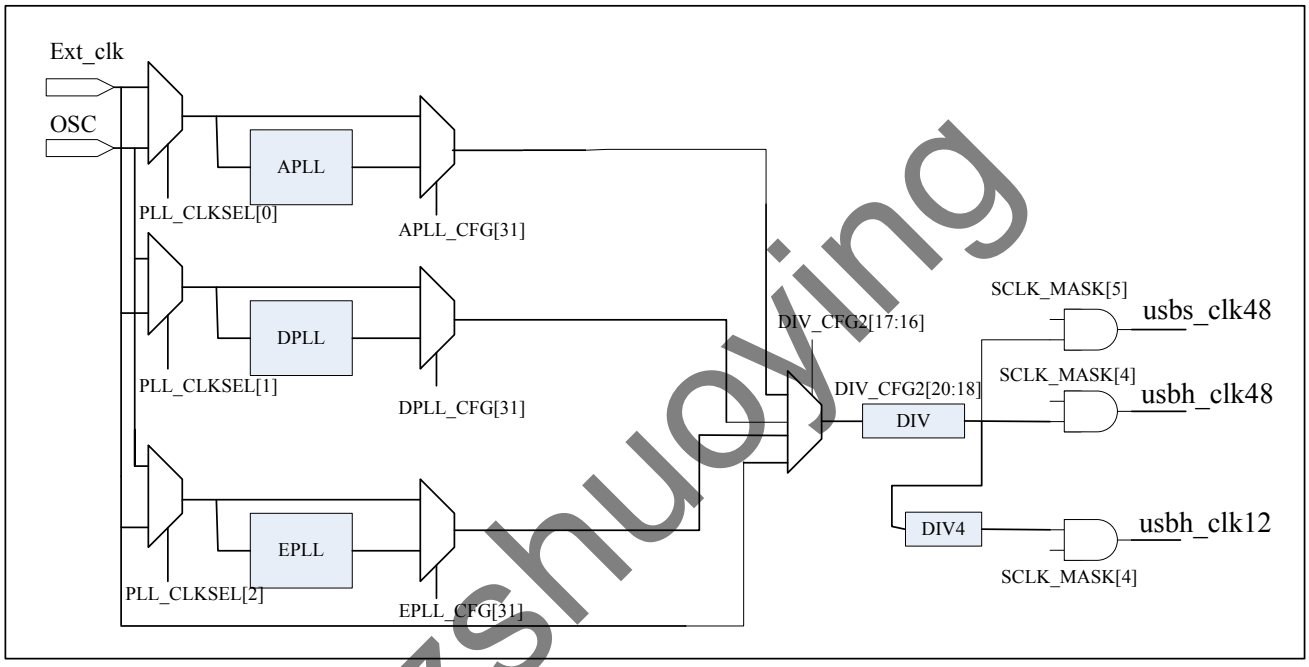


Figure 3-3: Clock Generation for USB

3.2.1.5 Clock Generation For UART,IIS,CAM,TIM0,TIM1

Clock generator for UART, IIS, CAMIF, TIM0, TIM1 is similar to clock generator for USB.

3.2.1.6 Clock Generation For TV Interface

Figure 3-4 shows the clock generator for TV interface. Usually, XTOUT3 provide one special clock for TV interface.

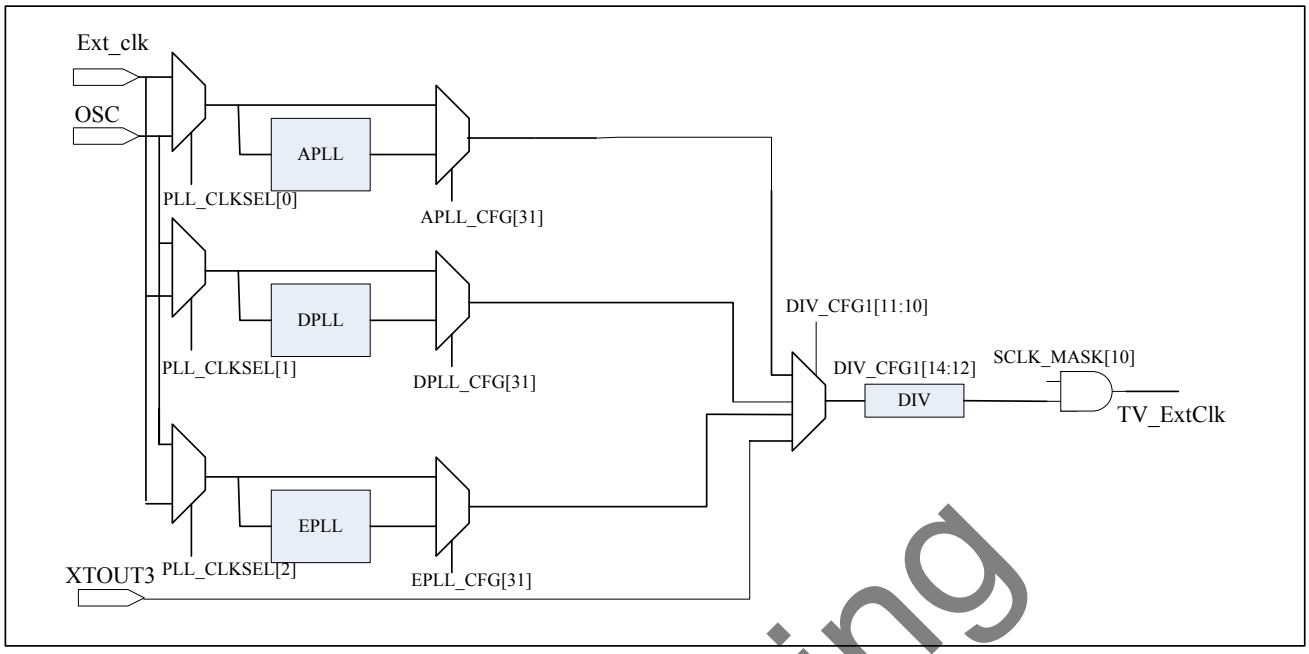


Figure 3-4: Clock Generation for TVIF

3.2.1.7 Clock ON/OFF Control

HCLK_MASK, PCLK_MASK, and SCLK_MASK control the clock operation. If a bit is cleared, the corresponding clock will be supplied through each clock divider. Otherwise, it will be masked.

- HCLK_MASK controls HCLK/HCLK2X domain.
- PCLK_MASK controls PCLK domain.
- Few peripheral require special clocks to operate correctly. The clocks are controlled by SCLK_MASK.

3.2.2 Reset Scheme

3.2.2.1 RESET

IMAPx210 has five types of reset signals and SYSCON can place the system into one of five 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.
- Warm reset: It is generated by asserting XnWRESET. XnWRESET is used to initialize IMAPx210 and preserve current hardware status.
- 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, watchdog timer monitors internal hardware status and generates reset signal to escape from this status.
- Software reset: It is generated by setting SW_RESET.
- Wakeup reset: It is generated when IMAPx210 wake up from SLEEP mode. Since internal hardware states are not available any more after SLEEP mode, they must be initialized.

3.2.2.2 Hardware Reset

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

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

XnRESET cannot be masked and is always enabled. Upon assertion of XnRESET, IMAPx210 enters into reset state regardless of the previous mode. XnRESET must be held long time enough to allow internal stabilization and propagation of the reset state to enter proper reset state. Power regulator for IMAPx210 must be stable prior to the deassertion of XnRESET. Otherwise, it may be damage and the operation is unpredictable.

3.2.2.3 Warm Reset

Warm reset is invoked when XnWRESET pin is asserted for more seven SYSCON work clock in NORMAL, IDLE, and STOP modes. In SLEEP mode, it is treated as a wake-up event. all registers except SYSCON ALIVE and RTC, are initialized. During the warm reset, the following actions occur:

- All blocks except for ALIVE and RTC block go to their pre-defined reset state.
- All pins get their reset state.
- The XnRSTOUT pin is asserted during warm reset.

When XnWRESET signal is asserted as '0', the following sequence occurs:

1) SYSCON requests all AHB bus controllers to finish current bus transactions.
2) Bus controllers send acknowledges to SYSCON after current bus transactions are finished.
3) SYSCON requests external memory controllers to enter into self-refresh mode, since the contents in the external memory must be pReserved while warm reset is asserted.
4) The memory controller's sends acknowledgement when they are in self-refresh mode.
5) SYSCON assert internal reset signals and XnRSTOUT.

Note: Warm reset only valid once until user clear Warm reset flag in RST_ST register by software,otherwise system manager will ignore the following warm reset assert.

3.2.2.4 Software Reset

Software reset is invoked when a software write 0x6565 to SW_RST. The behavior is same as warm reset case.

3.2.2.5 Watchdog Reset

Watchdog reset is invoked when a software hang-up. Then, the software cannot initialize a register within watchdog timer and watchdog timer makes time-out signals for watchdog reset. During the watchdog reset, the following actions take place:

- All blocks except for ALIVE and RTC 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 watchdog can generate time-out signal. It is invoked when watchdog timer and reset are enabled. Then, the following sequence takes place:

1) Watchdog timer generates 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.

3.2.2.6 Wakeup Reset

Wakeup reset is invoked when IMAPx210 is woken-up from SLEEP by a wakeup event. The details are described in SLEEP mode section.

3.2.3 Power Management

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 mode, however, retains previous states and supports fast wake-up time from STOP mode. That is, power is retained, but clock is off.
- Some blocks, which are Video DEC, Video ENC, GPU and Memory Pool, have no state retention feature. They can be power gating to reduce power consumption through internal power switch.
- In SLEEP mode, internal power switch will be OFF to reduce power consumption and IMAPX210 minimizes power consumption and lost all information except ALIVE and RTC block,and IO PAD.
- In SHUTDOWN mode, external power switch will be OFF to reduce power consumption and IMAPX210 minimizes power consumption and lost all information except ALIVE and RTC block.

3.2.3.1 Power Domain in IMAPx210

IMAPx210 consists of six power domains as shown in Figure 3-5. Sub-power domains, VDEC, VENC, GPU and MEMPL, are controlled by NPOW_CFG and GPOW_CFG. When IMAPx210 runs at NORMAL or IDLE mode, NPOW_CFG controls them. If the controlled bit is clear, corresponding block goes power-gating mode and lost previous state. Thus, user software must store internal state before clearing the corresponding bit. When IMAPx210 changes to STOP mode, sub-power domains automatically changes to power-gating mode.

3.2.3.2 Normal/IDLE mode

In NORMAL mode, CPU, GPU, Video Codec, and all peripherals can operate fully. Typical system bus operating frequency is up to 266/133MHz. The clock to each module can be stopped selectively by software to reduce power consumption. The ON/OFF clock gating of the individual clock source of each module is performed by controlling of each corresponding clock enable bit, which is specified by HCLK_MASK, PCLK_MASK, and SCLK_MASK configuration registers.

In IDLE mode, CPU is stopped without any change of other module. Typically, CPU waits a wake-up event to return to NORMAL mode. All modules can run at maximum operating frequency at NORMAL/IDLE mode. When some module are unnecessary to run, IMAPx210 can cut supply power using power-gating circuitry. As shown in Figure 3-5, four power domains can be independently controlled with NORMAL_CFG configuration register. When all functional modules are unnecessary to run, user software can cut supply power of the corresponding power domain, whose color is grey in Figure 3-5.

All internal status of the corresponding domain will be lost after the corresponding power domain is OFF. Thus, user software must store all information, which is required to restore internal state.

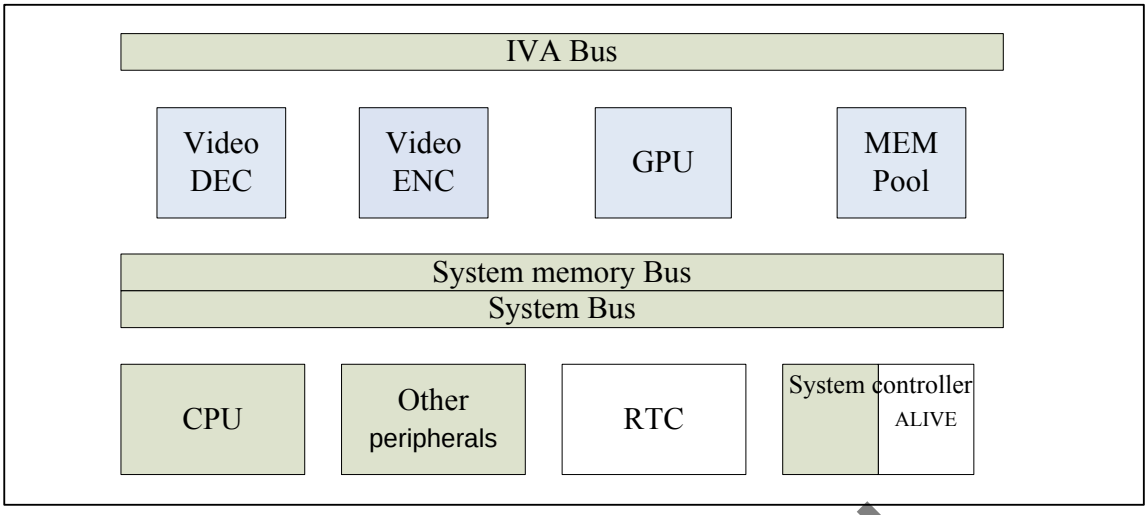


Figure 3-5: Power Domains at IDLE/NORMAL/STOP Mode

Note: Domains highlighted in filemot represent clock-off domain and domains highlighted in blue represent power-gating domain.

3.2.3.3 STOP mode

In STOP mode, ALL blocks (except ALIVE and RTC, system clock generator and DDR2), are retaining the previous state (clock-gating 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.

The following are the STOP mode entering sequence:

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 all bus controller to finish current bus transaction.
4) bus controller sends acknowledge to SYSCON after current bus transaction is completed.

To exit from STOP mode, all wake-up sources excepting normal interrupts are available. The following are the wake-up sequence from STOP mode:

- SYSCON releases system/system memory/IVA bus down request.

3.2.3.4 SLEEP mode

In SLEEP mode, all hardware logics excepting ALIVE, RTC blocks and IO, are power-off using internal 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.

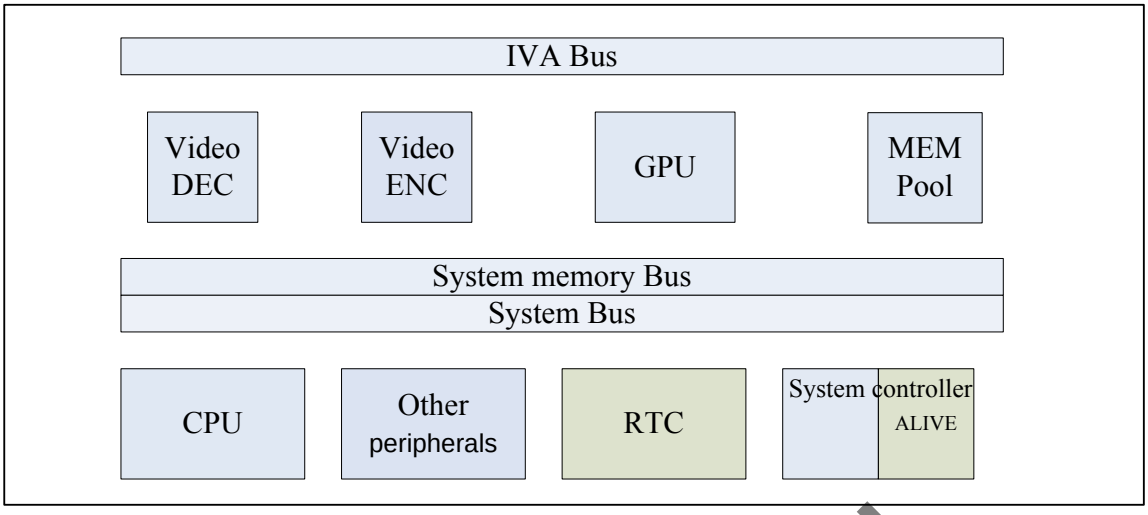


Figure 3-6: Power Domains at SLEEP Mode (only ALIVE and RTC keep internal state)

The following are the SLEEP mode entering sequence:

1) User must confirm the sub power domain is all power off.
2) User software sets GPOW _CFG[6:5] as IDLE/STOP/SLEEP mode.
3) User software generates STANDBYWFI signal by MCR instruction (MCR p15, 0, Rd, c7, c0, 4).
4) Wait CPU assert STANDBYWFI, turn off CPU clock, enter IDLE mode.
5) SYSCON requests all bus to finish current bus transaction.(send pause signal to arbiter)
6) Bus controller sends acknowledge to SYSCON after current bus transaction is completed.(all exist block)
7) SYSCON gate all clock except system manager work clock.
8) SYSCON requests external memory controllers to enter into self-refresh mode, since the contents in the external memory must be preserved during STOP/SLEEP mode.
9) The memory controllers send acknowledges when they are self-refresh mode.
10) disables PLL operations.
11) Change all IO direction to input.
12) Power off common domain.
13) Chip enter SLEEP and wait wake up event.

The following are the SLEEP mode exiting sequence:

1) Turn on module power in order.
2) Reset all modules excepting RTC domain.
3) SYSCON releases system reset signals including HRESETn and PRESETn.
4) NFCON copies boot code from an external NAND device to IBRAM if the boot device is NAND.
5) SYSCON releases CPU reset signals.

3.2.3.5 Wakeup

Table 3-1 shows various wake-up sources from low power state, IDLE, STOP, and SLEEP. According to the low power state, different wake-up sources are available.

Table 3-1: Power Mode Wake-up Sources

Power Mode			Wakeup Source
IDLE			all interrupt source
	STOP		KeyPad
		SLEEP	External Interrupt
			RTC Alarm
			Warm Reset

3.2.3.6 *Power On Sequence*

Figure 3-7: Power On Sequence

Note1: XnRESET must change to high before PowerKey is released.

Note2: PowIND indication the status of external power. If power is stable , PowIND need set high.When power is unstable about 31 ms, SYSCON will send a system interrupt to CPU, SYSCON will shutdown external power when power is unstable morethan 6 seconds.

Note3: When PowerKey is actived more than 2 seconds, SYSCON will send a sleep request to CPU, and SYSCON will shut down external power after PowerKey is actived more than 6 seconds.

Note4: A short time PowerKey actived is a wakeup event for IDLE, STOP and SLEEP.

3.3 System Controller Register Description

3.3.1 System Controller Register Memory Map

Address	Symbol	Direction	Description
0x20C0_0004	PLL_LOCKED	R	Status of A/D/E PLL
0x20C0_0008	PLL_OCLKSEL	R/W	PLL out clock selection
0x20C0_000c	PLL_CLKSEL	R/W	PLL reference clock cfg.
0x20C0_0010	APLL_CFG	R/W	Control PLL output frequency for APLL
0x20C0_0014	DPLL_CFG	R/W	Control PLL output frequency for DPLL
0x20C0_0018	EPLL_CFG	R/W	Control PLL output frequency for EPLL
0x20C0_001c	DIV_CFG0	R/W	Clock divider control 0
0x20C0_0020	DIV_CFG1	R/W	Clock divider control 1
0x20C0_0024	HCLK_MASK	R/W	AHB Clock gating control

0x20C0_0028	PCLK_MASK	R/W	APB Clock gating control
0x20C0_002c	SCLK_MASK	R/W	Special clock gating control
0x20C0_0030	CLKOUT0_CFG	R/W	CLKOUT0 Configuration
0x20C0_0034	CLKOUT1_CFG	R/W	CLKOUT1 Configuration
0x20C0_0038	CPUSYNC_CFG	R/W	CPU sync mode configuration
0x20C0_003C	DIV_CFG2	R/W	Clock divider control 2
0x20C0_0040	USB_SRST	R/W	USB SoftReset Register.
0x20C0_0044	PERSIM_CFG	R/W	Peripheral simulation mode configuration
0x20C0_0048	PAD_CFG	R/W	PAD configuration register
0x20C0_004C	GPU _CFG	R/W	GPU ram clock mode configuration register
0x20C0_005C	DIV_CFG3	R/W	Clock divider control 3
0x20C0_0060	DIV_CFG4	R/W	Clock divider control 4
0x20C0_0100	SW_RST	R/W	Generate software reset
0x20C0_0104	MEM_CFG	R	Memory configuration
0x20C0_0108	MEM_SWAP	R/W	Memory map swap
0x20C0_010c	BOOT_MD	R	Boot mode parameter
0x20C0_0110	RST_ST	R/W	Reset states
0x20C0_0114	PORT_PS_CFG	R/W	Memory port pause enable config
0x20C0_0118	Reserved	R/W	Reserved for future use
0x20C0_011C	Reserved	R/W	Reserved for future use
0x20C0_0120	MPL_MD	R/W	Memory pool mode
0x20C0_0200	GPOW_CFG	W	Global Power mode configuration
0x20C0_0204	WP_MASK	R/W	Wakeup source Mask
0x20C0_0208	POW_STB	R/W	Power on stable counter
0x20C0_020c	WP_ST	R/W	Wakeup states
0x20C0_0210	NPOW_CFG	R/W	Normal power configuration
0x20C0_0214	POW_ST	R/W	Normal power on states
0x20C0_0218	MD_ISO	R/W	Module isolation enable
0x20C0_021c	MD_RST	R/W	Module software reset
0x20C0_0220	AHBP_RST	R/W	AHB peri module software reset
0x20C0_0224	APBP_RST	R/W	APB peri module software reset
0x20C0_0228	AHBP_enable	R/W	AHB peri module output enable
0x20C0_022C	INFO0	R/W	Information register0
0x20C0_0230	INFO1	R/W	Information register1
0x20C0_0234	INFO2	R/W	Information register2
0x20C0_0238	INFO3	R/W	Information register3
0x20C0_0300	SLP_ORST	R/W	system reset out sleep mode register
0x20C0_0304	SLP_GPADAT	R/W	sleep mode GPA data register
0x20C0_0308	SLP_GPAPUD	R/W	sleep mode GPA Pull UP/Down Registers
0x20C0_030c	SLP_GPACON	R/W	sleep mode GPA output enable Registers
0x20C0_0310	SLP_GPBDAT	R/W	sleep mode GPB data register
0x20C0_0314	SLP_GPBPUD	R/W	sleep mode GPB Pull UP/Down Registers
0x20C0_0318	SLP_GPBCON	R/W	sleep mode GPB output enable Registers
0x20C0_031c	SLP_GPODAT	R/W	sleep mode GPO data register
0x20C0_0320	SLP_GPOPUD	R/W	sleep mode GPO Pull UP/Down Registers
0x20C0_0324	SLP_GPOCON	R/W	sleep mode GPO output enable Registers
0x20C0_0328	GPA_SLP_CTRL	R/W	Porta use sleep mode register enable
0x20C0_032c	GPB_SLP_CTRL	R/W	Portb use sleep mode register enable
0x20C0_0330	GPO_SLP_CTRL	R/W	Porto use sleep mode register enable
0x20C0_0334	RTC_INT_CFG	R/W	rtc_int output pad configuration

3.3.2 System Controller Indivdual Register Description

Register 3-1: Status of PLL (PLL_LOCKED,offset=0x0004)

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	R/W	Reserved	0x0
[2]	EPLL_locked	R	EPLL is locked	0x0
[1]	DPLL_locked	R	EPLL is locked	0x0
[0]	APLL_locked	R	APLL is locked	0x0

Register 3-2: PLL Output Selection (PLL_OCLKSEL, offset=0x0008)

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	R/W	Reserved	0x0
[2]	epll_oclk_sel	R/W	EPLL output clock selection 1: PLL out 0: Reference clock out	0
[1]	dpll_oclk_sel	R/W	DPLL output clock selection 1: PLL out 0: Reference clock out	0
[0]	apll_oclk_sel	R/W	APLL output clock selection 1: PLL out 0: Reference clock out	0

Register 3-3: PLL Clock Source Select (PLL_CLKSEL,offset=0x000C)

Field	Symbol	Direction	Description	Default
[31:3]	Reserved	R/W	Reserved	0x0
[2]	EPLL_SEL	R/W	EPLL reference clock source select. 0: External crystal 1: External clock	0x0
[1]	DPLL_SEL	R/W	DPLL reference clock source select. 0: External crystal 1: External clock	0x0
[0]	APLL_SEL	R/W	APLL reference clock source select. 0: External crystal 1: External clock	0x0

Register 3-4: APLL Configuration (APLL_CFG,offset=0x0010)

Field	Symbol	Direction	Description	Default
[31]	pll_en	R/W	PLL enable	0x0
[30:14]	Reserved	RO	Reserved	0x0
[13:12]	od	R/W	Output divider control	0x0
[11:7]	div	R/W	PLL divide value	0x1
[6:0]	mult	R/W	PLL M divide value	0x1

Register 3-5: DPLL Configuration (DPLL_CFG,offset=0x0014)

Field	Symbol	Direction	Description	Default
[31]	pll_en	R/W	PLL enable	0x0
[30:14]	Reserved	RO	Reserved	0x0
[13:12]	od	R/W	Output divider control	0x0
[11:7]	div	R/W	PLL divide value	0x1
[6:0]	mult	R/W	PLL M divide value	0x1

Register 3-6: EPLL Configuration(EPLL_CFG,offset=0x0018)

Field	Symbol	Direction	Description	Default
[31]	pll_en	R/W	PLL enable	0x0
[30:14]	Reserved	RO	Reserved	0x0
[13:12]	od	R/W	Output divider control	0x0
[11:7]	div	R/W	PLL divide value	0x1
[6:0]	mult	R/W	PLL M divide value	0x1

Register 3-7: Clock Divider Configuration 0(DIV_CFG0, offset=0x001C)

Field	Symbol	Direction	Description	Default
[31:22]	Reserved	R/W	Reserved	0x0
[21:20]	SYSCLK_MUX	R/W	System clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[17:16]	PCLK_RATIO	R/W	PCLK clock divider ratio $F_{pclk} = F_{hclk}/(2^{**} PCLK\_RATIO)$	0x1
[15:12]	Reserved	R/W	Reserved	0
[11:8]	HCLKX2_RATIO	R/W	hclk2x clock divider ratio $F_{hclk2x} = F_{pll}$ /HCLKX2_RATIO	0x2
[7:4]	HCLK_RATIO	R/W	HCLK clock divider ratio $F_{hclk} = F_{pll}$/HCLK_RATIO	0x2
[3:0]	CPU_RATIO	R/W	CPU clock divider ratio $F_{cpuclk} = F_{pll}$/CPU_RATIO	0x2

Register 3-8: CLK Divider Configuration 1(DIV_CFG1, offset=0x0020)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	R/W	Reserved	0x0
[28:26]	UART _RATIO	R/W	UART external clock divider ratio(Fuartclk = Fuartsel/(UART _RATIO +1))	0x0
[25:24]	UART_CLK_SEL	R/W	UART external clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[23]	Reserved	R	Reserved	0x0
[22:18]	CAM_RATIO	R/W	Camera output clock divider ratio	0x0
[17:16]	CAM_CLK_SEL	R/W	Camera output clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[15]	Reserved	R/W	Reserved	0x0
[14:12]	TV_RATIO	R/W	TV clock divider ratio	0x0
[11:10]	TV_CLK_SEL	R/W	TV clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External TV reference clock	0x3
[9:7]	TIM _RATIO	R/W	TIMER1 clock divider ratio	0x0
[6:5]	TIM _CLK_SEL	R/W	TIMER1 clock source selection 00: APLL output 01: DPLL output	0x3

			10: EPLL output 11: External clock	
[4:2]	TIM_RATIO	R/W	TIMER0 clock divider ratio	0x0
[1:0]	TIM_CLK_SEL	R/W	TIMER0 clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3

Register 3-9: CLK Divider Configuration 2(DIV_CFG2, offset=0x003C)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	R/W	Reserved	0x0
[28:26]	IIS_RATIO	R/W	IIS clock divider ratio	0x0
[25:24]	IIS_CLK_SEL	R/W	IIS clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[23]	Reserved	R/W	Reserved	0x0
[22:18]	USB_RATIO	R/W	USB clock divider ratio	0x0
[17:16]	USB_CLK_SEL	R/W	USB clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[15:13]	Reserved	R/W	Reserved	0x0
[12:10]	ASYN_RATIO	R/W	Asyn main cpu clock divider ratio	0x0
[9:8]	ASYN_CLK_SEL	R/W	Asyn main cpu clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[7:5]	Reserved	R/W	Reserved	0x0
[4:2]	GPU_RATIO	R/W	GPU clock divider ratio	0x0
[1:0]	GPU_CLK_SEL	R/W	GPU clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3

Register 3-10: CLK Divider Configuration 3(DIV_CFG3, offset=0x005C)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	R/W	Reserved	0x0
[28:21]	Reserved	R/W	Reserved	0x0
[20:18]	SD2_RATIO	R/W	SD2 clock divider ratio	0x0
[17:16]	SD2_CLK_SEL	R/W	SD2 clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[15:13]	Reserved	R/W	Reserved	0x0
[12:10]	SD1_RATIO	R/W	SD1 clock divider ratio	0x0
[9:8]	SD1_CLK_SEL	R/W	SD1 clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3

[7:5]	Reserved	R/W	Reserved	0x0
[4:2]	SD0_RATIO	R/W	SD0 clock divider ratio	0x0
[1:0]	SD0_CLK_SEL	R/W	SD0 clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3

Register 3-11: CLK Divider Configuration 4(DIV_CFG4, offset=0x0060)

Field	Symbol	Direction	Description	Default
[31:13]	Reserved	R/W	Reserved	0x0
[12:10]	IDSTF_RATIO	R/W	IDS external interface clock divider ratio	0x0
[9:8]	IDSTF_CLK_SEL	R/W	IDS external interface clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3
[7:5]	Reserved	R/W	Reserved	0x0
[4:2]	IDS_RATIO	R/W	IDS clock divider ratio	0x0
[1:0]	IDS _CLK_SEL	R/W	IDS clock source selection 00: APLL output 01: DPLL output 10: EPLL output 11: External clock	0x3

Register 3-12: HCLK Gating Control (HCLK_MASK ,offset=0x0024)

Field	Symbol	Direction	Description	Default
[31:23]	Reserved	R/W	Reserved	0x0
[22]	MemoryPool	R/W	Gating hclk for Memory Pool.(0:pass;1:mask)	0x0
[21]	SD2	R/W	Gating hclk for SD/SDHC/SDIO2 controller.(0:pass;1:mask)	0x0
[20]	SD1	R/W	Gating hclk for SD/SDHC/SDIO1 controller.(0:pass;1:mask)	0x0
[19]	Reserved	R/W	Reserved	0x0
[18]	ETH	R/W	Gating hclk for Ethernet MAC controller.(0:pass;1:mask)	0x0
[17]	NAND	R/W	Gating hclk for NandFlash controller.(0:pass;1:mask)	0x0
[16]	NorFlash	R/W	Gating hclk for USB slave.(0:pass;1:mask)	0x0
[15]	INTC	R/W	Gating hclk for Interrupt controller.(0:pass;1:mask)	0x0
[14]	SD0	R/W	Gating hclk for SD/SDHC/SDIO0.(0:pass;1:mask)	0x0
[13]	CF/IDE	R/W	Gating hclk for CF-ATA.(0:pass;1:mask)	0x0
[12]	APB	R/W	Gating hclk for APB arbiter.(0:pass;1:mask)	0x0
[11]	OTG	R/W	Gating hclk for OTG.(0:pass;1:mask)	0x0
[10]	Monitor	R/W	Gating hclk for memory access monitor.(0:pass;1:mask)	0x0
[9]	DRAM	R/W	Gating hclk for DRAM port .(0:pass;1:mask)	0x0
[8]	USBH	R/W	Gating hclk for USB host.(0:pass;1:mask)	0x0
[7]	CAMIF	R/W	Gating hclk for Camera Interface.(0:pass;1:mask)	0x0
[6]	IDS	R/W	Gating hclk for Dispaly Subsystem. (0:pass;1:mask)	0x0
[5]	DMA	R/W	Gating hclk for DMA. (0:pass;1:mask)	0x0
[4]	Reserved	R/W	Reserved	0x0
[3]	GPU	R/W	Gating hclk for 2D/3D graphices. (0:pass;1:mask)	0x0
[2]	VDEC	R/W	Gating hclk for Video decoder. (0:pass;1:mask)	0x0
[1]	VENC	R/W	Gating hclk for Video encoder. (0:pass;1:mask)	0x0
[0]	SYSC/MISC/AHB	R	Gating hclk for system controller.(always pass)	0x0

Register 3-13: Pclk Gating Control (PCLK_MASK, ,offset=0x0028)

Field	Symbol	Direction	Description	Default
[31:21]	Reserved	R/W	Reserved (0:pass;1:mask)	0x0
[20]	RTC	R/W	Gating pclk for RTC. (0:pass;1:mask)	0x0
[19]	SPI	R/W	Gating pclk for SPI. (0:pass;1:mask)	0x0
[18]	Slave SSI	R/W	Gating pclk for Slave SSI (0:pass;1:mask)	0x0
[17]	Master SSI 2	R/W	Gating pclk for Master SSI 2 (0:pass;1:mask)	0x0
[16]	PS2_1	R/W	Gating pclk for PS2-1 (0:pass;1:mask)	0x0
[15]	PS2_0	R/W	Gating pclk for PS2-0 (0:pass;1:mask)	0x0
[14]	KB	R/W	Gating pclk for Keypad. (0:pass;1:mask)	0x0
[13]	UART3	R/W	Gating pclk for UART3. (0:pass;1:mask)	0x0
[12]	UART2	R/W	Gating pclk for UART2. (0:pass;1:mask)	0x0
[11]	UART1	R/W	Gating pclk for UART1. (0:pass;1:mask)	0x0
[10]	UART0	R/W	Gating pclk for UART0. (0:pass;1:mask)	0x0
[9]	GPIO	R/W	Gating pclk for GPIO. (0:pass;1:mask)	0x0
[8]	Master SSI1	R/W	Gating pclk for Master SSI1. (0:pass;1:mask)	0x0
[7]	Master SSI0	R/W	Gating pclk for Master SSI0. (0:pass;1:mask)	0x0
[6]	AC97	R/W	Gating pclk for Ac97. (0:pass;1:mask)	0x0
[5]	IIS	R/W	Gating pclk for Iis. (0:pass;1:mask)	0x0
[4]	IIC1	R/W	Gating pclk for IIC1. (0:pass;1:mask)	0x0
[3]	IIC0	R/W	Gating pclk for IIC0. (0:pass;1:mask)	0x0
[2]	WatchDog	R/W	Gating pclk for Watchdog. (0:pass;1:mask)	0x0
[1]	Pwm-timer	R/W	Gating pclk for PWM timer. (0:pass;1:mask)	0x0
[0]	Cmn-timer	R/W	Gating pclk for Common timer. (0:pass;1:mask)	0x0

Register 3-14: Special Clock Gating Control(SCLK_MASK, offset=0x002C)

Field	Symbol	Direction	Description	Default
[31:18]	Reserved	R	Reserved	0x0
[17]	IBRAM_CLK	R/W	Gating hclk clock for IBRAM	0x0
[16]	VDEC_CLK	R/W	Gating special clock for VDEC core clock. (0:pass;1:mask)	0x0
[15]	IDS_EITF_CLK	R/W	Gating special clock for IDS exteral interface. (0:pass;1:mask)	0x0
[14]	IDS_CLK	R/W	Gating special clock for OSD. (0:pass;1:mask)	0x0
[13]	USB_PHY_CLK	R/W	Gating special clock for usb PHY reference clock. (0:pass;1:mask)	0x0
[12]	SD2_CLK	R/W	Gating special clock for SD2 interface. (0:pass;1:mask)	0x0
[11]	SD1_CLK	R/W	Gating special clock for SD1 interface. (0:pass;1:mask)	0x0
[10]	SD0_CLK	R/W	Gating special clock for SD0 interface. (0:pass;1:mask)	0x0
[9]	TV_CLK	R/W	Gating special clock for TV interface. (0:pass;1:mask)	0x0
[8]	TIM1_CLK	R/W	Gating special clock for Timer1 interface. (0:pass;1:mask)	0x0
[7]	TIM0_CLK	R/W	Gating special clock for Timer0 interface. (0:pass;1:mask)	0x0
[6]	CAM_CLK	R/W	Gating special clock for Camera interface. (0:pass;1:mask)	0x0
[5]	OTG_CLK	R/W	Gating clock for USB OTG. (0:pass;1:mask)	0x0
[4]	USBH_CLK	R/W	Gating 48 MHz /12MHz clock for USB host. (0:pass;1:mask)	0x0
[3]	IIS_CLK	R/W	Gating special clock for IIS. (0:pass;1:mask)	0x0
[2]	GPU_CLK	R/W	Gating GPU clock. (0:pass;1:mask)	0x0
[1]	Reserved	R/W	Reserved	0x0
[0]	UART_CLK	R/W	Gating special clock for UART. (0:pass;1:mask)	0x0

Register 3-15: CLKOUT0 Configuration Register(CLKOUT0_CFG ,offset=0x0030)

Field	Symbol	Direction	Description	Default
[31:4]	CLK_Ratio	R/W	Clkout0 divide ratio	0x0
[3]	Enable	R/W	Clkout0 output enable	0x0

[2:0]	CLK_Sel	R/W	000 : External crystal clock 001 : External clock 010 : RTC clock 011 : TV reference clock 100 : OTG PHY clock 101 : Main sync clock 110 : DPLL out clock 111 : EPLL out clock	0

Register 3-16: CLKOUT1 Configuration Register(CLKOUT1_CFG ,offset=0x0034)

Field	Symbol	Direction	Description	Default
[31:4]	CLK_Ratio	R/W	Clkout1 divide ratio	0x0
[3]	Enable	R/W	Clkout1 output enable	0x0
[2:0]	CLK_Sel	R/W	000 : External crystal clock 001 : External clock 010 : RTC clock 011 : TV reference clock 100 : OTG PHY clock 101 : APLL out clock 110 : Main async clock 111 : EPLL out clock	0

Register 3-17: CPU SYNC Mode Configuration Register (CPUSYNC_CFG, offset=0x0038)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	R/W	Reserved	0x0
[1:0]	CPUSYNC	R/W	00: cpu =main sync clock(asyn mode), 01: cpu= main async clock (asyn mode), 10: cpu= main sync clock (sync mode) 11: Reserved	0x0

Register 3-18: USB Soft Reset (USB_SRST, offset=0x0040)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	R/W	Reserved	0x0
[3]	USBH_PHY_RSTN	R/W	USB host phy soft reset(low level active)	0x1
[2]	USBH_RSTN	R/W	USB host soft reset(low level active)	0x1
[1]	USBO_PHY_RSTN	R/W	USB OTG phy soft reset(low level active)	0x1
[0]	USBO_RSTN	R/W	USB OTG soft reset(low level active)	0x1

Register 3-19: Pad Configuration (PAD_CFG, offset=0x0048)

Field	Symbol	Direction	Description	Default
[31:7]	Reserved	R/W	Reserved	0x0
[6:4]	USB_PLL_CFG	R/W	USB PLL CONFIG [2:1] USB HOST [3] OTG PLL and bandgap standalone feature 0: disable 1: enable	0x0
[3]	Uphy2_suspend	R/W	Force usb phy2 enter suspend mode(host/otg phy) low level active	0x0
[2]	Uphy1_suspend	R/W	Force usb phy1 enter suspend mode low level active	0x0
[1]	Uphy0_suspend	R/W	Force usb phy0 enter suspend mode low level active	0x0
[0]	PAD_CFG	R/W	KEYPAD row line[7:0] use:	0x0

			0:GPIO J 1:GPIO K [8:5] [14:11]	

Register 3-20: GPU Ram Clock Mode Register (GPUCFG, ,offset=0x004C)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	R	Reserved	0x0
[0]	RAMCLKGDSB	R/W	GPU Memory Clock Gate Disable 1 : memory clock gate disable 0 : memory clock gate enable	0x0

Register 3-21: Soft Reset Register (SW_RST,offset=0x0100)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	R	Reserved	0x0
[15:0]	SWRST	R/W	Generate software reset when the value is 0x6565 when software reset is compeleted, SYSCON will clear the value.	0x0

Register 3-22: Memory IO Configuration Register (MEM_CFG,offset=0x0104)

Field	Symbol	Direction	Description	Default
[31:17]	Reserved	R	Reserved	0x0
[16]	Port_no_init	R/W	memory port has been inited	0
[15:9]	Reserved	R	Reserved	0
[8]	USB2_as_host	R/W	USB Host port 2 is used in host application 0: USB port 2 is used as USB OTG 2.0 1: USB port 2 is used as USB host 2.0	0
[7:3]	Reserved	R	Reserved	0
[2]	Reserved	R/W	Must be always set to 1	0x1
[1]	Reserved	R/W	Must be always set to 1	0x1
[0]	Reserved	R/W	Must be always set to 1	0x1

Register 3-23: Memory Map Swap (MEM_SWAP,offset=0x0108)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	R	Reserved	0x0
[0]	BOOT_SWAP	R/W	Indicate adress 0x0 map onto IBRAM or not 1: Boot Image Area map onto IBRAM space. 0: Boot Image Area map onto address space dependent on XBPARA[4:0]	0x0

Register 3-24: Boot Mode Parameter (BOOT_MD,offset=0x010C)

Field	Symbol	Direction	Description	Default
[31:7]	Reserved	R	Reserved	0x0
[6]	NAND_WD	R	NAND bus wodth. 0: 8-bit bus width 1: 16-bit bus width	Pin state
[5]	ADRCYCLE	R	NAND flash address cycles. 0: 3 address cycle(flash_type = 0) or 4 address cycle(flash_type = 1,2) 1: 4 address cycle(flash_type = 0) or 5 address cycle(flash_type = 1,2)	Pin state
[4]	PAGESIZE	R	NAND flash page size. 0: Page=256Words(flash_type = 0) or Page=1KWords(flash_type = 1) or Page=2KWords(flash_type = 2)	Pin state

			1: Page=512Bytes(flash_type = 0) or Page=2KBytes(flash_type = 1) or Page=4KBytes(flash_type = 2)	
[3:2]	NAND_TYPE	R	Boot nand flash type 00: Normal NAND flash(256Words/512Bytes page size, 3/4 address cycle) 01: Advance NAND flash(1KWords/2KBytes page size, 4/5 address cycle) 10: Advance NAND flash (2KWords/4KBytes page size, 4/5 address cycle)	Pin state
[1]	NOT_NBOOT	R	not nand boot	Pin state
[0]	BOOT_IND	R	Boot mode. 0: NAND boot 1: SRAM boot	Pin state

Register 3-25: Reset States (RST_ST,offset=0x0110)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4]	WAKEUP_RESET	R/W	Reset by SLEEP mode wake-up. This is cleared by writing 1	0x0
[3]	SW_RESET	R/W	Software reset by SWRESET. This is cleared by writing 1	0x0
[2]	WARM_RESET	R/W	Warm reset by XnWRESET. This is cleared by writing 1. note : Once warm reset is occured user must clear this flag by software,otherwise system manager will ignore the following warm reset.	0x0
[1]	Reserved	R	Reserved	0x0
[0]	HW_RESET	R/W	External reset by NRESET pin. This is cleared by writing 1	0x1

Note: If reset is occured and no reset flag is asserted, it means a watchdog reset is occured.

Register 3-26: Memory Pool Mode (MP_MD,offset=0x0120)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	R	Reserved	0x0
[1:0]	MP_SEL	R/W	0: Memory mode 1: VDEC mode 2: reserved mode 3: DEBUG mode For more detail information,please reference to memory pool function description.	0x0

Register 3-27: Global Power Mode Configuration (GPOW_CFG,offset=0x0200)

Field	Symbol	Direction	Description	Default
[31:3]	Reserved	R	Reserved	0x0
[2:0]	POW_MODE	W	Chip Power mode configuration. 000: Normal 001: IDLE 010: STOP 011: SLEEP 100: SHUTDOWN	0x0

Register 3-28: Wakeup Source Mask (WP_MASK,offset=0x0204)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	R	Reserved	0x0
[25:16]	PARAINF	R/W	Enable parallel debug information output 1: Enable 0: Disable Note: User must clear These bit in Normal operation.	0x3ff
[15]	DBINF	R/W	Enable debug info output 1: Enable 0: Disable Note: User must clear this bit in Normal operation.	0x1
[14]	KEYP_CFG	R/W	1: key pad share with eth	0

			0: key pad share with nor memory interface	
[13:8]	EINT_Polarity	R/W	External interrupt wakeup level, 0: High level 1 : low level	0x0
[7:2]	EINT_WP	R/W	Mask external interrupt wakeup.(0:pass;1:mask)	0x0
[1]	KEY_WP	R/W	Mask key pad key press wakeup.(0:pass;1:mask)	0x0
[0]	RTC_WP	R/W	Mask RTC alarm wakeup.(0:pass;1:mask)	0

Register 3-29: Power Interrupt States(POW_INT_ST,offset=0x0208)

Field	Symbol	Direction	Description	Default
[31:3]	Reserved	R	Reserved	0x0
[2]	POW_WARN_INT	RW	Power is under stable states,request shut down system power.Write 1 to clear flag	0x0
[1]	SHD_INT	RW	Shut down request on external power key . Write 1 to clear flag	0x0
[0]	SLEEP_INT	RW	Sleep request on external power key . Write 1 to clear flag	0x0

Register 3-30: Wakeup States (WP_ST,offset=0x020c)

Field	Symbol	Direction	Description	Default
[31:10]	Reserved	R	Reserved	0x0
[9]	EINT5_WP	R/W	Wakeup by external interrupt5. This is cleared by writing 1	0x0
[8]	EINT4_WP	R/W	Wakeup by external interrupt4. This is cleared by writing 1	0x0
[7]	EINT3_WP	R/W	Wakeup by external interrupt3. This is cleared by writing 1	0x0
[6]	EINT2_WP	R/W	Wakeup by external interrupt2. This is cleared by writing 1	0x0
[5]	EINT1_WP	R/W	Wakeup by external interrupt1. This is cleared by writing 1	0x0
[4]	EINT0_WP	R/W	Wakeup by external interrupt0. This is cleared by writing 1	0x0
[3]	INT_WP	R/W	Wakeup by internal interrupt. This is cleared by writing 1	0x0
[2]	EINT_WP	R/W	Wakeup by external interrupt(global flag, inculde exint 0 to 5). This is cleared by writing 1	0x0
[1]	RTC_WP	R/W	Wakeup by RTC alarm. This is cleared by writing 1	0x0
[0]	POW_WP	R/W	Wakeup by power mode interrupt . This is cleared by writing 1	0x0

Register 3-31: Normal Power Configuration (NPOW_CFG,offset=0x0210)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4]	GPU_PC	R/W	GPU domain power on control bit,write 1 power on video decode,0 power down	0x0
[3]	VENC_PC	R/W	Video encode power on control bit,write 1 power on Video encode,0 power down	0x0
[2]	MEMPL_PC	R/W	Memory pool power on control bit ,write 1 power on ,0 power down	0x0
[1]	VDEC_PC	R/W	Video decode power on control bit,write 1 power on video decode,0 power down	0x0
[0]	Reserved	R/W	Reserved	0x0

Register 3-32: Normal Power On States (POW_ST,offset=0x0214)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4]	GPU_PON_ACK	R	GPU is ready when this bit is set to 1	0x0
[3]	VENC_PON_ACK	R	Video encode is ready when this bit is set to 1.	0x0
[2]	MEMPL_PON_ACK	R	memory pool is ready when this bit is set to 1.	0x0
[1]	VDEC_PON_ACK	R	Video decode is ready when this bit is set to 1.	0x0
[0]	Reserved	R	Reserved	0x0

Register 3-33: Module Isolation Enable (MD_ISO,offset=0x0218)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4]	GPU_ ISO	R/W	GPU isolation enable. 1:isolation all output, 0:enable output	0x1
[3]	VENC_ ISO	R/W	Video isolation enable. 1:isolation all output, 0:enable output	0x1
[2]	MEMPL_ISO	R/W	memory pool isolation enable. 1:isolation all output, 0:enable output	0x1
[1]	VDEC_ ISO	R/W	Video isolation enable. 1:isolation all output, 0:enable output	0x1
[0]	Reserved	R/W	Reserved	0x1

Note: User must isolate sub-power domain before it's power is stable(power ack valid).

Register 3-34: Module Soft Reset (MD_RST,offset=0x021C)

Field	Symbol	Direction	Description	Default
[6]	GPU_BUS_ RST	R/W	GPU bus software reset.	0x0
[5]	Reserved	R/W	Reserved	0x0
[4]	GPU_ RST	R/W	GPU core software reset.	0x0
[3]	VENC_ RST	R/W	Video software reset.	0x0
[2]	MEMPL_ RST	R/W	memory pool software reset.	0x0
[1]	VDEC_ RST	R/W	Video software reset.	0x0
[0]	Reserved	R/W	Reserved	0x0

Note: It is strongly sugguest reset module domain before release isolation when module is powered on.

Register 3-35: AHB Peripheral Reset (AHBP_RST,offset=0x0220)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	R	Reserved	0x0
[24]	IVA_ARB_RST	R/W	IVA axi bus arbiter software reset.	0x0
[23]	Reserved	R/W	Reserved	0x0
[22]	MP_ITF _RST	R/W	mempool master/slave bridge software reset	0x0
[21]	SD2_RST	R/W	SD2 software reset.	0x0
[20]	SD1_ RST	R/W	SD1 software reset.	0x0
[19]	Reserved	R/W	Reserved	0x0
[18]	ETH	R/W	Ethernet software reset	0x0
[17]	Reserved	R/W	Reserved	0x0
[16]	Reserved	R/W	Reserved	0x0
[15]	INTC	R/W	Interrupt controller software reset	0x0
[14]	SD0	R/W	SD0 software reset	0x0
[13]	CF	R/W	CF/IDE software reset	0x0
[12]	APB	R/W	Ahb to apb bridge software reset	0x0
[11]	Reserved	R/W	Reserved	0x0
[10]	DBG_Monitor	R/W	DRAM Debug moniter software reset	0x0
[9]	Reserved	R/W	Reserved	0x0
[8]	Reserved	R/W	Reserved	0x0
[7]	CAMIF	R/W	Camera interface software reset	0x0
[6]	IDS	R/W	IDS software reset	0x0
[5]	DMA	R/W	DMA software reset	0x0
[4]	Reserved	R/W	Reserved	0x0
[3]	Reserved	R/W	Reserved	0x0
[2]	sys2vdec	R/W	Video decoder ahb bus bridge reset	0x0
[1]	sys2ivenc	R/W	Video encoder ahb bus bridge reset	0x0
[0]	Reserved	R/W	Reserved	

Register 3-36: APB Peripheral Reset (APBP_RST,offset=0x0224)

Field	Symbol	Direction	Description	Default
[19]	SPI_RST	R/W	SPI software reset	0x0
[18]	SLV_SSI_RST	R/W	Slave SSI software reset	0x0

[17]	MS_SSI2_RST	R/W	Master SSI2 software reset	0x0
[16]	PIC1	R/W	PS2-1 software reset	0x0
[15]	PIC0	R/W	PS2-0 software reset	0x0
[14]	KeyPad	R/W	KeyPad software reset	0x0
[13]	UART3	R/W	UART3 software reset	0x0
[12]	UART2	R/W	UART2 software reset	0x0
[11]	UART1	R/W	UART1 software reset	0x0
[10]	UART0	R/W	UART0 software reset	0x0
[9]	GPIO	R/W	GPIO software reset	0x0
[8]	MS_SSI1_RST	R/W	Master SSI1 software reset	0x0
[7]	MS_SSI0_RST	R/W	Master SSI0 software reset	0x0
[6]	AC97	R/W	AC97 software reset	0x0
[5]	IIS	R/W	IIS software reset	0x0
[4]	IIC1	R/W	IIC1 software reset	0x0
[3]	IIC0	R/W	IIC0 software reset	0x0
[2]	Reserved	R/W	Reserved	0x0
[1]	Pwm	R/W	PWM tiner Timer software reset	0x0
[0]	Cmn_time	R/W	common Timer software reset	0x0

Register 3-37: AHB Peripheral Bus Output Enable (AHBP_BOE,offset=0x0228)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	R	Reserved	0x0
[24]	IVA_ARB_BOE	R/W	IVA axi bus arbiter bus output enable.	0x0
[23]	Reserved	R/W	Reserved	0x0
[22]	MP_ITF _BOE	R/W	mempool master/slave bridge output enable	0x0
[21]	SD2_BOE	R/W	SD2 output enable	0x0
[20]	SD1_ BOE	R/W	SD1 output enable	0x0
[19]	Reserved	R/W	Reserved	0x0
[18]	ETH_BOE	R/W	Ethernet software reset	0x0
[17]	NAND_BOE	R/W	NandFlash controller output enable	0x0
[16]	SRAM_BOE	R/W	SRAM port output enable	0x0
[15]	INTC_BOE	R/W	Interrupt controller output enable	0x0
[14]	SD0_BOE	R/W	SD0 output enable	0x0
[13]	CF_BOE	R/W	CF/IDE output enable	0x0
[12]	APB_BOE	R/W	Ahb to apb bridge output enable	0x0
[11]	USBO_BOE	R/W	USB OTG output enable	0x0
[10]	DBG_BOE	R/W	DRAM Debug moniter output enable	0x0
[9]	DRAM_BOE	R/W	DRAM Port Register output enable	0x0
[8]	USBH_BOE	R/W	USB Host output enable	0x0
[7]	CAMIF_BOE	R/W	Camera interface output enable	0x0
[6]	IDS_BOE	R/W	IDS output enable	0x0
[5]	DMA_BOE	R/W	DMA output enable	0x0
[4]	Reserved	R/W	Reserved	0x0
[3]	GPU_BOE	R/W	GPU output enable	0x0
[2]	sys2vdec_BOE	R/W	Video decode ahb bus output enable	0x0
[1]	sys2ivenc_BOE	R/W	Video encode ahb bus output enable	0x0

Register 3-38: Information (0~3)(INFO0~3, offset=0x0228～0x0234)

Field	Symbol	Direction	Description	Default
[31:0]	INFO	R/W	information. User defined information. The content in INFORM0~3 registers only can be cleared by asserting XnRESET pin.	0x0

Register 3-39: Sleep Mode Resetout Control Register(SLP_ORST, offset=0x0300)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	R	Reserved	0x0
[0]	RSTOUT	R/W	Sleep mode system reset out pad value.	0x1

Register 3-40: Sleep Mode GPA Data Register(SLP_GPAD, offset=0x0304)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	R	Reserved	0x0
[7:0]	GPA_DAT	R/W	Sleep mode GPA data.	0x0

Register 3-41: Sleep GPA Pull Configuration Register(SLP_GPAP, offset=0x0308)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	R	Reserved	0x0
[7:0]	GPA_PUD	R/W	Sleep mode GPA pull down/pull up enable. 1 : pull up enable 0 : pull up disable	0x0

Register 3-42: Sleep GPA IO Configuration (SLP_GPAC, offset=0x030C)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	R	Reserved	0x0
[7:0]	GPA_OEN	R/W	Sleep mode GPA output enable. 0: output 1:input	0xff

Register 3-43: Sleep Mode GPB Data Register(SLP_GPBD, offset=0x0310)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4:0]	GPB_DAT	R/W	sleep mode GPB data.	0x0

Register 3-44: Sleep GPB Pull Configuration Register(SLP_GPBP, offset=0x0314)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4:0]	GPB_PUD	R/W	sleep mode GPB pull down/pull up enable.	0x0

Register 3-45: Sleep GPB IO Configuration (SLP_GPBC,0x offset=0318)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	R	Reserved	0x0
[4:0]	GPB_OEN	R/W	sleep mode GPB output enable. 0: output 1:input	0xff

Register 3-46: Sleep Mode GPO Data Register(SLP_GPOD, offset=0x031C)

Field	Symbol	Direction	Description	Default
[15:0]	GPO_DAT	R/W	sleep mode GPO data.	0x0

Register 3-47: Sleep GPO Pull Configuration Register(SLP_GPOP, offset=0x0320)

Field	Symbol	Direction	Description	Default
[15:0]	GPO_PUD	R/W	sleep mode GPO pull down/pull up enable.	0x0

Register 3-48: Sleep GPO IO Configuration (SLP_GPOC, offset=0x0324)

Field	Symbol	Direction	Description	Default
[15:0]	GPO_OEN	R/W	sleep mode GPO output enable. 0: output 1:input	0xff

Register 3-49: GPA Sleep Control (GPA_SLP_CTRL, offset=0x0328)

Field	Symbol	Direction	Description	Default
[7:0]	GPA_USE_SLP	R/W	Use Sleep mode register 1: use sleep mode control register 0: use Normal GPIO register	0x0

Register 3-50: GPB Sleep Control (GPB_SLP_CTRL, offset=0x032C)

Field	Symbol	Direction	Description	Default
[4:0]	GPB_USE_SLP	R/W	Use Sleep mode register 1: use sleep mode control register 0: use Normal GPIO register	0x0

Register 3-51: GPO Sleep Control (GPO_SLP_CTRL, offset=0x0330)

Field	Symbol	Direction	Description	Default
[16]	RSTOUT_USE_SLP	R/W	System reset out pad use sleep configuration	0x0
[15:0]	GPO_USE_SLP	R/W	Use Sleep mode register 1: use sleep mode control register 0: use Normal GPIO register	0x0

Register 3-52: Sleep OSC configuration (RTC_INT_CFG, offset=0x0334)

Field	Symbol	Direction	Description	Default
[31:17]	Reserved	R	Reserved	0x0
[16:1]	STABCNT	R/W	Stable counter.	
[0]	SLP_OSC_C	R/W	sleep mode powerdown oscillator enable 1: In sleep mode, power down crystal oscillator 0: Crystal oscillator is always working	0x0

4 DRAM Controller

4.1 Overview

DRAM controller is an external memory controller and support multi SDRAM – DDR, mobile DDR and DDR2. The SDRAM interface support 16bit/32bit memory data bus and up to 1GB size per chip. DRAM port supports up to 266MHz with using DDR/mobile DDR and up to 533MHz with DDR2 memory chip. There are two dedicated chip select output and ODT.

Features

- Fully pipelined read/write data and command interfaces to the memory controller.
- Advanced bank look-ahead features for high memory throughput.
- Front-end interface to 6 BUS interface.
- A programmable register interface to control memory device parameters and protocols including auto pre-charge.
- Full initialization of memory on memory controller reset.
- Programmable memory data path size of full memory data width or half memory data width.
- Clock frequencies from 80 MHz to 266 MHz supported.

Block Diagram

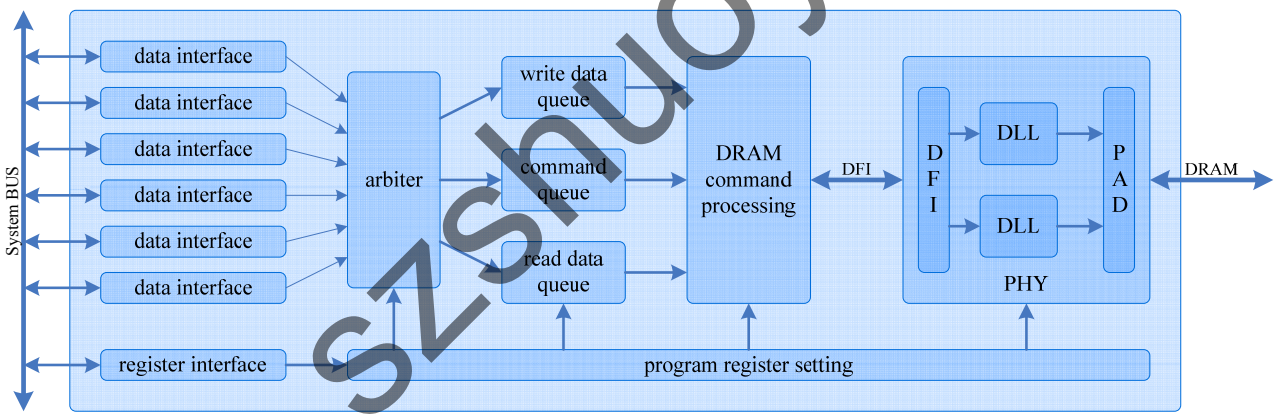


Figure 4-1: DRAM Block Diagram

Figure 4-1 shows the functional block diagram of DRAM, there are several main modules in DRAM. The data interfaces are connect with external system bus and they can transfer data simultaneously. Arbiter will arbitrate transactions from different interface. According to address and data type of transaction, the transactions will be pushed in different queue. Then these transactions will be send to DRAM processing and access the external DRAM through PHY logic.

4.2 Functional Description

4.2.1 Address Mapping

DRAM controller map user addresses to the DRAM memory in a contiguous block. Addressing starts at user address 0 and ends at the highest available address according to the size and number of DRAM devices present. This mapping is dependent on how the memory controller was configured and how the parameters in the internal control registers are programmed. The

mapping of the address space to the internal data storage structure of the DRAM devices is based on the actual size of the DRAM devices available. The size is stored in user-programmable parameters that must be initialized at power up. Certain DRAM devices allow for different mapping options to be chosen, while other DRAM devices depend on the burst length chosen.

The address structure of DDR SDRAM devices contains five fields. Each of these fields can be individually addressed when accessing the DRAM. The address map for this Memory Controller is ordered as follows:

Chip Select -- Row -- Bank -- Column -- Data width

The maximum widths of the fields are based on the configuration settings. The actual widths of the fields may be smaller if the device address width parameters (addr_pins, eight_bank_mode and column_size) are programmed differently.

The maximum user address range is determined by the width of the memory data width, the number of chip select pins, and the address space of the DRAM device. The maximum amount of memory can be calculated by the following formula:

$$MaxMemBytes = ChipSelects * 2^{Address} * NumBanks * DPWidthBytes$$

The maximum values for these fields are as follows:

Chip Selects = 2
Device Address = 15 + 13 (Row + Column)
Number of Banks per Chip Select = 8
Memory Data Width in Bytes = 4 bytes

As a result, the maximum accessible memory area is 16 GB. But for system address limited, DRAM just support 2GB access address totally and 1GB address per chip select.

4.2.2 Core Command Queue with Placement Logic

The Memory Controller core contains a command queue that accepts commands from the Arbiter. This command queue uses a placement algorithm to determine the order that commands will execute in the Memory Controller core. The placement logic follows many rules to determine where new commands should be inserted into the queue, relative to the contents of the command queue at the time. Placement is determined by considering address collisions, source collisions, data collisions, command types and priorities. In addition, the placement logic attempts to maximize efficiency of the Memory Controller core through command grouping and bank splitting. Once placed into the command queue, the relative order of commands is constant. Many of the rules used in placement may be individually enabled or disabled. In addition, the queue may be disabled by clearing the placement_en parameter, resulting in an in-line queue that services requests in the order they are received. If the placement_en parameter is cleared to 'b0, the placement algorithm will be ignored.

Address Collision/Data Coherency Violation

The order in which read and write commands are processed in the memory controller is critical to proper system behavior. While reads and writes to different addresses are independent and may be re-ordered without affecting system performance, reads and writes that access the same address are significantly related. If the port requests a read after a write to the same address, then repositioning the read before the write would return the original data, not the changed data. Similarly, if the read was requested ahead of the write but accidentally positioned after the write, then the read would return the new data, not the original data prior to being overwritten. These are significant data coherency mistakes.

To avoid address collisions, reads or writes that access the same chip select, bank and row as a command already in the command queue will be inserted into the command queue after the original command, even if the new command is of a higher priority.

This factor may be enabled/disabled through the addr_cmp_en parameter and should only be disabled if the system can guarantee coherency of reads and writes.

Write Buffer Collision

Incoming write requests in the command queue are allocated to one of the 8 write buffers of the Memory Controller core automatically based on availability. New write commands will be designated to any available buffer. However, back-to-back write requests from a particular source will be allocated to the same write buffer as the previous command.

Since the Memory Controller core must pull data out of the buffers in the order it was stored, if a write command is linked to

a buffer that is associated with another command in the queue, then the new command will be placed in the command queue after that command, regardless of priority.

This feature will always be enabled.

Priority

Priorities are used to distinguish important commands from less important commands. Each command is given a priority based on the command type through the programmable parameters axiY_r_priority and axiY_w_priority (where Y represents the port number). A priority value of 0 is the highest priority, and a priority value of 3 is the lowest priority for this Memory Controller. The placement algorithm will attempt to place higher priority commands ahead of lower priority commands, as long as they have no source ID, write buffer or address collisions. Higher priority commands will be placed lower in the command queue if they access the same address, are from the same requestor or use the same buffer as lower priority commands already in the command queue.

This feature is enabled through the priority_en parameter.

Bank Splitting

Before accesses can be made to two different rows within the same bank, the first active row must be closed (pre-charged) and the new row must be opened (activated). Both activities require some timing overhead; therefore, for optimization, the placement queue will attempt to insert the new command into the command queue such that commands to other banks may execute during this timing overhead. The placement of the new commands will still follow priority, source ID, write buffer and address collision rules. The placement logic will also attempt to optimize the Memory Controller core by inserting a command to the same bank as an existing command in the command queue immediately after the original command. This reduces the overall timing overhead by potentially eliminating one pre-charging/ activating cycle. This placement will only be possible if there are no priorities, source ID, write buffer or address collisions or conflicts with other commands in the command queue. All bank splitting features are enabled through the bank_split_en parameter.

Read/Write Grouping

The memory suffers a small timing overhead when switching from read to write mode. For efficiency, the placement queue will attempt to place a new read command sequentially with other read commands in the command queue, or a new write command sequentially with other write commands in the command queue. Grouping will only be possible if no priority, source ID, write buffer or address collision rules are violated. This feature is enabled through the rw_same_en parameter.

Command Aging

Once a command has been placed in the command queue, its order relative to the other commands in the queue at that time is fixed. While this provides simplicity in the algorithm, there are drawbacks. For this reason, the Memory Controller offers two options that affect commands once they have been placed in the command queue.

Since commands can be inserted ahead of existing commands in the command queue, the situation could occur where a low priority command remains at the bottom of the queue indefinitely. To avoid such a lockout condition, aging counters have been included in the placement logic that measure the number of cycles that each command has been waiting. If command aging is enabled through the active_aging parameter, then if an aging counter hits its maximum, the priority of the associated command will be decremented by one (lower priority commands are executed first). This increases the likelihood that this command will move to the top of the command queue and be executed. Note that this command does not move relative positions in the command queue when it ages; the new priority will be considered when placing new commands into the command queue.

Aging is controlled through a master aging counter and command aging counters associated with each command in the command queue. The age_count and command_age_count parameters hold the initial values for each of these counters, respectively. When the master counter counts down the age_count value, a signal is sent to the command aging counters to decrement. When the command aging counters have completely decremented, then the priority of the associated command is decremented by one number and the counter is reset. Therefore, a command does not age by a priority level until the total elapsed cycles have reached the product of the age_count and command_age_count values. The maximum number of cycles that any command can wait in the command queue until reaching the top priority level is the product of the age_count value, the command_age_count value, and the number of priority levels in the system.

High-Priority Command Swapping

Commands are assigned priority values to ensure that critical commands are executed more quickly in the memory controller than less important commands. Therefore, it is desirable that high-priority commands pass into the Memory Controller core as soon as possible. The placement algorithm takes priority into account when determining the order of commands, but still allows a scenario in which a high-priority command sits waiting at the top of the command queue while another command, perhaps of a lower priority, is in process. The high-priority command swapping feature allows this new high-priority command to be executed more quickly. If the user has enabled the swapping function through the swap_en parameter, then the entry at the top of the command queue will be compared with the current command in progress. If the command queue's top entry is of a higher priority (not the same priority), and it does not have an address, source ID or write buffer conflict with the current command being executed, then the original command will be interrupted. For this memory controller, an additional check is performed before a read command is interrupted. If the read command in progress and the read command at the top of the command queue are from the same port, then the executing command will only be interrupted if the swap_port_rw_same_en parameter is set to 'b1. If this parameter is cleared to 'b0, a read command from the same port as a read command in progress, even with a higher priority and without any conflicts, would remain at the top of the command queue while the current command completes.

Note: All write commands from a single port, even with different source IDs, will be executed in order. Therefore, two write commands from the same port will never be swapped regardless of the settings of the swap_en and swap_port_rw_same_en parameters.

Note: Priorities are assigned to read commands based on the settings in the axiY_r_priority parameters. While all read commands from a port are assigned the same priority when placed in the command queue, their priorities may change over time through command aging. While uncommon, it is possible that a higher-priority read command may be at the top of the command queue while a lower-priority read command is executing. The behavior of the system in this scenario is based on the value of the swap_en and swap_port_rw_same_en parameters.

Table 4-1: Swapping Behavior

Active Command Priority	New Command Priority	Originating Port for Commands	Conflicts?	Action
Higher	Lower	Same or Different	Yes or No	Current Command continues
Lower	Higher	Same	Yes	Current Command continues
Lower	Higher	Same	No	Will swap IF swap_en = 1 & swap_port_rw_same_en = 1
Lower	Higher	Different	No	Will swap IF swap_en = 1

If the command is to be interrupted, it will be halted after completing the current burst, stored and placed at the top of the queue, and the new command will be executed. As long as the command queue is not full, new commands may continue to be inserted into the command queue based on the placement rules, even at the head of the queue ahead of the interrupted command. The top entry in the command queue will be executed next. Whenever the interrupted command is resumed, it will start from the point at which it was interrupted.

Note that priority 0 commands will never be interrupted, so the user should set any commands that should not be interrupted to priority 0. If supported by the port interfaces, setting the swap_port_rw_same_en parameter will enable interleaving.

Low Power Operation

There are five low power modes available in the Memory Controller. The low power modes are listed from least to most power saving.

Note: It is not possible to exit one low power mode and enter another low power mode simultaneously. The user should plan for a minimum delay between exit and entry between the two low power modes of 15 cycles in which the memory controller must remain stable.

1. Memory Power-Down

The memory controller sets the memory devices into power-down, which reduces the overall power consumption of the system, but has the least effect of all the low power modes. In this mode, the memory controller and memory clocks are fully operational, but the CKE input bit to the memory devices is de-asserted. The memory controller will continue to monitor memory refresh needs and will automatically bring the memory out of power-down to perform these refreshes. When a

refresh is required, the CKE input bit to the memory devices will be re-enabled. This action brings the memory devices out of power-down. Once the refresh has been completed, the memory devices will be returned to power-down by de-asserting the CKE input bit.

2. Memory Power-Down with Memory Clock Gating

The memory controller sets the memory devices into power-down and gates off the clock to the memory devices. Refreshes will be handled as in the Memory Power-Down mode (Mode 1), with the exception that gating on the memory clock will be removed before asserting the CKE pin. After the refresh has been completed, the memory devices will be returned to power-down with the clock gated. Before the memory devices are removed from power-down, the clock will be gated on again. The user should not use this mode for memory devices that do not support memory clock gating. Clock gating is not supported for standard DDR1 and DDR2 devices. When set into this mode, the memory controller will attempt to place the memory devices in power-down and gate off the memory clock. The memory will function unpredictably and may hang.

3. Memory Self-Refresh

The memory controller sets the memory devices into self-refresh. In this mode, the memory controller and memory clocks are fully operational and the CKE input bit to the memory devices is de-asserted. Since the memory automatically refreshes its contents, the memory controller does not need to send explicit refreshes to the memory.

4. Memory Self-Refresh with Memory Clock Gating

The memory controller sets the memory devices into self-refresh and gates off the clock to the memory devices. Before the memory devices are removed from self-refresh, the clock will be gated on again.

5. Memory Self-Refresh with Memory and Controller Clock Gating

This is the deepest low power mode of the memory controller. The memory controller sets the memory devices into self-refresh and gates off the clock to the memory device. In addition, the clock to the memory controller and the programming parameters will be gated off, except to a small portion of the DLL, which must remain active to maintain the lock. Before the memory devices are removed from self-refresh, the memory controller and memory clocks will be gated on.

The memory controller may enter and exit the various low power modes in the following ways:

- Automatic Entry:
  When the memory controller is idle, four timing counters begin counting the cycles of inactivity. If any of the counters expires, the memory controller will enter the low power mode associated with that counter.
- Manual Entry:
  The user may initiate any low power mode by setting the bit of the lowpower_control parameter associated with the desired mode. The memory controller will enter the selected low power mode when it is has completed its current burst.

Automatic and Manual entry methods are both controlled by two parameters: lowpower_control and low power_auto_enable. The lowpower_control parameter contains individual enable/disable bits for each low power mode, and the lowpower_auto_enable parameter controls whether each mode will be entered automatically or manually.

Table 4-2: Low Power Mode Parameters

Low Power Mode	Enable?	Entry?
Memory Power Down (Mode 1)	lowpower_control [4] =’b1	lowpower_auto_enable [4] 0: Manual 1: Automatic
Memory Power-Down with Memory Clock Gating (Mode 2)	lowpower_control [3] =’b1	lowpower_auto_enable [3] 0: Manual 1: Automatic
Memory Self-Refresh (Mode 3)	lowpower_control [2] =’b1	lowpower_auto_enable [2] 0: Manual 1: Automatic
Memory Self-Refresh with Memory Clock Gating (Mode 4)	lowpower_control [1] =’b1	lowpower_auto_enable [1] 0: Manual 1: Automatic

Memory Self-Refresh with Memory and Controller Clock Gating (Mode 5)	lowpower_control [0] =’b1	lowpower_auto_enable [0] 0: Manual 1: Automatic

When a lowpower_control parameter bit is set to ’b1 by the user, the Memory Controller checks the lowpower_auto_enable parameter.

- If the associated bit in the lowpower_auto_enable parameter is set to 1, then the memory controller will watch the associated counter for expiration, and then enter that low power mode. Table 4-3 shows the correlation between the low power modes and the counters that control each mode’s automatic entry.
- If the associated bit in the lowpower_auto_enable parameter is cleared to 0, then the memory controller will complete its current memory burst access and then enter the specified low power mode.

Table 4-3: Low Power Mode Counters

Low Power Mode	Counter
Memory Power Down (Mode 1)	lowpower_power_down_cnt
Memory Power Down with Memory Clock Gating (Mode 2)	lowpower_power_down_cnt
Memory Self Refresh (Mode 3)	lowpower_self_refresh_cnt
Memory Self Refresh with Memory Clock Gating (Mode 4)	lowpower_external_cnt
Memory SelfRefresh with Memory and Controller Clock Gating (Mode 5)	lowpower_internal_cnt

Note that the values in the lowpower_auto_enable parameter are only relevant when the associated lowpower_control bit is set to ’b1.

Multiple bits of the lowpower_control and lowpower_auto_enable parameters can be set to ’b1 at the same time. When this happens, the memory controller always enters the deepest low power mode of all the modes that are enabled. If the memory controller is already in one low power mode when a deeper low power mode is requested automatically or manually, it must first exit the current low power mode, and then enter the deeper low power mode. A minimum 15 cycle delay occurs before the second entry.

The timing for automatic entry into any of the low power modes is based on the number of idle cycles that have elapsed in the memory controller. There are four counters related to the five low power modes to determine when any particular low power mode will be entered if the automatic entry option is chosen. Since the two power-down modes share one counter, if the user wishes to enter Memory Power- Down mode (Mode 1) automatically, then the Memory Power-Down with Memory Clock Gating mode (Mode 2) must not be enabled.

4.3 DRAM Register Description

4.3.1 DRAM Register Memory Map

DRAM’s register mapping address range in system is 0x20C20000~0x20C2FFFF. DRAM register’s system accessing address equal to BASE_ADDRESS (0x20C20000) plus address offset.

R/W = *Read/Write.*
R = *Read Only.*
W = *Write Only.*
R/W+ = Read/Write, where one or more bits of the parameter have additional functionality and require special handling.

Table 4-4: DRAM Register Memory Map

Address	Symbol	Direction	Description
0x000	CTRL_REG_0	R/W+	DRAM control register 0
0x004	CTRL_REG_1	R/W	DRAM control register 1
0x008	CTRL_REG_2	R/W	DRAM control register 2
0x00c	CTRL_REG_3	R/W+	DRAM control register 3
0x010	CTRL_REG_4	R/W+	DRAM control register 4
0x014	CTRL_REG_5	R/W	DRAM control register 5
0x018	CTRL_REG_6	R/W	DRAM control register 6
0x01c	CTRL_REG_7	R/W	DRAM control register 7

0x020	CTRL_REG_8	R/W+	DRAM control register 8
0x024	CTRL_REG_9	R/W+	DRAM control register 9
0x028	CTRL_REG_10	R/W	DRAM control register 10
0x02c	CTRL_REG_11	R/W+	DRAM control register 11
0x030	CTRL_REG_12	R/W	DRAM control register 12
0x034	CTRL_REG_13	R/W	DRAM control register 13
0x038	CTRL_REG_14	R/W	DRAM control register 14
0x03c	CTRL_REG_15	R/W	DRAM control register 15
0x040	CTRL_REG_16	R/W+	DRAM control register 16
0x044	CTRL_REG_17	R/W+	DRAM control register 17
0x048	CTRL_REG_18	R/W	DRAM control register 18
0x04c	CTRL_REG_19	R/W+	DRAM control register 19
0x050	CTRL_REG_20	R/W	DRAM control register 20
0x054	CTRL_REG_21	R/W	DRAM control register 21
0x058	CTRL_REG_22	R/W	DRAM control register 22
0x05c	CTRL_REG_23	R/W	DRAM control register 23
0x060	CTRL_REG_24	R/W	DRAM control register 24
0x064	CTRL_REG_25	R/W	DRAM control register 25
0x068	CTRL_REG_26	R/W+	DRAM control register 26
0x06c	CTRL_REG_27	R/W+	DRAM control register 27
0x070	CTRL_REG_28	R/W+	DRAM control register 28
0x074	CTRL_REG_29	R/W+	DRAM control register 29
0x078	CTRL_REG_30	R/W	DRAM control register 30
0x07c	CTRL_REG_31	R/W	DRAM control register 31
0x080	CTRL_REG_32	R/W	DRAM control register 32
0x084	CTRL_REG_33	R/W+	DRAM control register 33
0x088	CTRL_REG_34	R/W+	DRAM control register 34
0x08c	CTRL_REG_35	R/W+	DRAM control register 35
0x090	CTRL_REG_36	R/W +	DRAM control register 36
0x094	CTRL_REG_37	R/W+	DRAM control register 37
0x098	CTRL_REG_38	R/W	DRAM control register 38
0x09c	CTRL_REG_39	R/W	DRAM control register 39
0x0a0	CTRL_REG_40	R/W+	DRAM control register 40
0x0a4	CTRL_REG_41	R/W+	DRAM control register 41
0x0a8	CTRL_REG_42	R/W	DRAM control register 42
0x0ac	CTRL_REG_43	R/W	DRAM control register 43
0x0b0	CTRL_REG_44	R/W	DRAM control register 44
0x0b4	CTRL_REG_45	R/W	DRAM control register 45
0x0b8	CTRL_REG_46	R/W	DRAM control register 46
0x0bc	CTRL_REG_47	R/W	DRAM control register 47
0x0c0	CTRL_REG_48	R/W	DRAM control register 48
0x0c4	CTRL_REG_49	R/W	DRAM control register 49
0x0c8	CTRL_REG_50	R/W	DRAM control register 50
0x0cc	CTRL_REG_51	R/W	DRAM control register 51
0x0d0	CTRL_REG_52	R/W	DRAM control register 52
0x0d4	CTRL_REG_53	R/W	DRAM control register 53
0x0d8	CTRL_REG_54	R/W	DRAM control register 54
0x0dc	CTRL_REG_55	R/W	DRAM control register 55
0x0e0	CTRL_REG_56	R/W	DRAM control register 56
0x0e4	CTRL_REG_57	R/W	DRAM control register 57
0x0e8	CTRL_REG_58	R/W+	DRAM control register 58
0x0ec	CTRL_REG_59	R/W	DRAM control register 59
0x0f0	CTRL_REG_60	R/W	DRAM control register 60
0x0f4	CTRL_REG_61	R/W	DRAM control register 61
0x0f8	CTRL_REG_62	R/W	DRAM control register 62
0x0fc	CTRL_REG_63	R/W	DRAM control register 63
0x100	CTRL_REG_64	R/W	DRAM control register 64

0x104	CTRL_REG_65	R/W	DRAM control register 65
0x108	CTRL_REG_66	R/W	DRAM control register 66
0x10c	CTRL_REG_67	R/W	DRAM control register 67
0x110	CTRL_REG_68	R/W	DRAM control register 68
0x114	CTRL_REG_69	R/W	DRAM control register 69
0x118	CTRL_REG_70	R	DRAM control register 70
0x11c	CTRL_REG_71	R	DRAM control register 71
0x120	CTRL_REG_72	R	DRAM control register 72
0x124	CTRL_REG_73	R	DRAM control register 73
0x128	CTRL_REG_74	R/W	DRAM control register 74
0x12c	CTRL_REG_75	R/W	DRAM control register 75
0x130	CTRL_REG_76	R/W	DRAM control register 76
0x134	CTRL_REG_77	R/W	DRAM control register 77
0x138	CTRL_REG_78	R/W	DRAM control register 78
0x13c	CTRL_REG_79	R/W	DRAM control register 79
0x140	CTRL_REG_80	R/W	DRAM control register 80
0x144	CTRL_REG_81	R/W	DRAM control register 81
0x148	CTRL_REG_82	R/W	DRAM control register 82
0x14c	CTRL_REG_83	R/W	DRAM control register 83
0x150	CTRL_REG_84	R	DRAM control register 84
0x154	CTRL_REG_85	R	DRAM control register 85
0x158	CTRL_REG_86	R	DRAM control register 86
0x15c	CTRL_REG_87	R	DRAM control register 87
0x170	CTRL_REG_92	R	DRAM control register 92
0x174	CTRL_REG_93	R	DRAM control register 93
0x178	CTRL_REG_94	R	DRAM control register 94
0x17c	CTRL_REG_95	R	DRAM control register 95

4.3.2 DRAM Registers and Field Descriptions

Register 4-1: DRAM control register 0 (CTRL_REG_0)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	AREFRESH	W	Initiates an automatic refresh to the DRAM devices based on the setting of the auto_refresh_mode parameter. If there are any open banks when this parameter is set, the Memory Controller will automatically close these banks before issuing the auto-refresh command. This parameter will always read back as 0x0. ● 0: No action ● 1: Issue refresh to the DRAM devices	0
[23:17]	Reserved	N/A	Reserved	0
[16]	AP	RW	Enables auto pre-charge mode for DRAM devices. Note: This parameter may not be modified after the start parameter has been asserted. ● 0: Auto pre-charge mode disabled. Memory banks will stay open until another request requires this bank, the maximum open time ($t_{ras\ max}$) has elapsed, or a refresh command closes all the banks. ● 1: Auto pre-charge mode enabled. All read and write transactions must be terminated by an auto pre-charge command. If a transaction consists of multiple read or write bursts, only the last command is issued with an auto pre-charge.	0
[15:9]	Reserved	N/A	Reserved	0
[8]	ADDR_CMP_EN	RW	Enables address collision/data coherency detection as a condition when using the placement logic to fill the command queue. ● 0: Disabled ● 1: Enabled	0
[7:1]	Reserved	N/A	Reserved	0
[0]	ACTIVE_AGING	RW	Enables aging of commands in the command queue when using the	0

			placement logic to fill the command queue. The total number of cycles required to decrement the priority value on a command by one is the product of the values in the age_count and command_age_count parameters. ● 0: Disabled ● 1: Enabled	

Register 4-2: DRAM control register 1 (CTRL_REG_1)

field	symbol	Direction	description	default
[31:25]	Reserved	N/A	Reserved	0
[24]	AXI2_BDW_OVFLOW	RW	Determines the behavior of bus interface 2 when the maximum bandwidth has been reached. ● 0: the port does not have access to the memory controller when its bandwidth allocation has been exceeded, even if the memory controller can accept the command. ● 1: the port may exceed the bandwidth allocation if the command queue contains less entries than the value in the arb_cmd_q_threshold parameter and no other requestors (with available bandwidth) are requesting access with commands at the same priority level	0
[23:17]	Reserved	N/A	Reserved	0
[16]	AXI1_BDW_OVFLOW	RW	Determines the behavior of bus interface 1 when the maximum bandwidth has been reached. ● 0: the port does not have access to the memory controller when its bandwidth allocation has been exceeded, even if the memory controller can accept the command. ● 1: The port may exceed the bandwidth allocation if the command queue contains less entries than the value in the arb_cmd_q_threshold parameter and no other requestors (with available bandwidth) are requesting access with commands at the same priority level	0
[15:9]	Reserved	N/A	Reserved	0
[8]	AXI0_BDW_OVFLOW	RW	Determines the behavior of bus interface 0 when the maximum bandwidth has been reached. ● 0: the port does not have access to the memory controller when its bandwidth allocation has been exceeded, even if the memory controller can accept the command. ● 1: The port may exceed the bandwidth allocation if the command queue contains less entries than the value in the arb_cmd_q_threshold parameter and no other requestors (with available bandwidth) are requesting access with commands at the same priority level	0
[7:1]	Reserved	N/A	Reserved	0
[0]	AUTO_REFRESH_MODE	RW	Sets the mode for when the automatic refresh will occur. If the auto_refresh_mode parameter is set and a refresh is required to memory, the memory controller will delay this refresh until the end of the current transaction (if the transaction is fully contained inside a single page), or until the current transaction hits the end of the current page. ● 0: Issue refresh on the next DRAM burst boundary, even if the current command is not complete. ● 1: Issue refresh on the next command boundary.	0

Register 4-3: DRAM control register 2 (CTRL_REG_2)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	BANK_SPLIT_EN	RW	Enables bank splitting as a condition when using the placement logic to fill the command queue.	0

			● 0: Disabled ● 1: Enabled	
[23:17]	Reserved	N/A	Reserved	0
[16]	AXI5_BDW_OVFLOW	RW	Determines the behavior of bus interface 5 when the maximum bandwidth has been reached. ● 0: The port does not have access to the memory controller when its bandwidth allocation has been exceeded, even if the memory controller can accept the command. ● 1: The port may exceed the bandwidth allocation if the command queue contains less entries than the value in the arb_cmd_q_threshold parameter and no other requestors (with available bandwidth) are requesting access with commands at the same priority level	0
[15:9]	Reserved	N/A	Reserved	0
[8]	AXI4_BDW_OVFLOW	RW	Determines the behavior of bus interface 4 when the maximum bandwidth has been reached. ● 0: The port does not have access to the memory controller when its bandwidth allocation has been exceeded, even if the memory controller can accept the command. ● 1: The port may exceed the bandwidth allocation if the command queue contains less entries than the value in the arb_cmd_q_threshold parameter and no other requestors (with available bandwidth) are requesting access with commands at the same priority level	0
[7:1]	Reserved	N/A	Reserved	0
[0]	AXI3_BDW_OVFLOW	RW	Determines the behavior of bus interface 3 when the maximum bandwidth has been reached. ● 0: The port does not have access to the memory controller when its bandwidth allocation has been exceeded, even if the memory controller can accept the command. ● 1: The port may exceed the bandwidth allocation if the command queue contains less entries than the value in the arb_cmd_q_threshold parameter and no other requestors (with available bandwidth) are requesting access with commands at the same priority level	0

Register 4-4: DRAM control register 3 (CTRL_REG_3)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	CONCURRENTAP	RW	Enables concurrent auto pre-charge. Some DRAM devices do not allow one bank to be auto pre-charged while another bank is reading or writing. The JEDEC standard allows concurrent auto pre-charge. Set this parameter for the DRAM device being used. ● 0: Concurrent auto pre-charge disabled. ● 1: Concurrent auto pre-charge enabled.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	BIST_GO	W	Initiates a BIST operation. This parameter will always read back as 0x0. **Note:** All programmable registers must be static for a minimum of 10 clock cycles prior to writing 1 to the *bist_go* parameter. ● 0: No action ● 1: Initiate a BIST operation	0
[15:9]	Reserved	N/A	Reserved	0
[8]	BIST_DATA_CHECK	RW	Enables the data check portion of the BIST algorithm. When the BIST operation is initiated by setting the *bist_go* parameter, the data lines will be tested.	0
[7:1]	Reserved	N/A	Reserved	0
[0]	BIST_ADDR_CHECK	RW	Enables the address check portion of the BIST algorithm. When the BIST operation is initiated by setting the *bist_go* parameter, the address lines	0

			will be tested.	

Register 4-5: DRAM control register 4 (CTRL_REG_4)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	DQS_N_EN	RW	Enables differential data strobe signals from the DRAM. ● 0: Single-ended DQS signal from the DRAM. ● 1: Differential DQS signal from the DRAM.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	DLL_BYPASS_MODE	RW	Defines the behavior of the DLL bypass logic and establishes which set of delay parameters will be used. ● 0: Normal operational mode. ● 1: Bypass the DLL master delay line.	0
[15:9]	Reserved	N/A	Reserved	0
[8]	DLLLOCKREG	R	Shows status of the DLL as locked or unlocked. This parameter is read-only ● 0: DLL is unlocked. ● 1: DLL is locked.	0
[7:1]	Reserved	N/A	Reserved	0
[0]	CONCURRENTAP_WR_ONLY	RW	Limit concurrent auto-precharge by waiting for the write recovery time, tWR, to complete after a write before issuing a read. ● 0: Do not restrict concurrent auto-precharge. ● 1: Wait tWR after a write before issuing a read.	0

Register 4-6: DRAM control register 5 (CTRL_REG_5)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	FAST_WRITE	RW	Controls when the write commands are issued to the DRAM devices. ● 0: The memory controller will issue a write command to the DRAM devices when it has received enough data for one DRAM burst. In this mode, write data can be sent in any cycle relative to the write command. This mode also allows for multi-word write command data to arrive in non-sequential cycles. ● 1: The memory controller will issue a write command to the DRAM devices after the first word of the write data is received by the memory controller. The first word can be sent at any time relative to the write command. In this mode, multi-word write command data must be available to the memory controller in sequential cycles.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	ENABLE_QUICK_SREFRESH	RW	When this bit is set to 1, the memory initialization sequence may be interrupted and the memory may enter self-refresh mode. This is used to place the memory devices into self-refresh mode when a power loss is detected during the initialization process. ● 0: Continue memory initialization. ● 1: Interrupt memory initialization and enter self-refresh mode	0
[15:9]	Reserved	N/A	Reserved	0
[8]	EIGHT_BANK_MODE	RW	Indicates that the memory devices have eight banks. ● 0: Memory devices have 4 banks. ● 1: Memory devices have 8 banks	0
[7:1]	Reserved	N/A	Reserved	0
[0]	DRIVE_DQ_DQS	RW	Selects if the DQ output enables and DQS output enables will be driven active when the memory controller is in an idle state. ● 0: Leave the output enables in their current state when idle. ● 1: Drive the data_byte_disable signal high when idle.	0

Register 4-7: DRAM control register 6 (CTRL_REG_6)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	NO_CMD_INIT	RW	Disables DRAM commands until DLL initialization is complete and *tdll* has expired. ● 0: Issue only REF and PRE commands during DLL initialization of the DRAM devices. If PRE commands are issued before DLL initialization is complete, the command will be executed immediately, and then the DLL initialization will continue. ● 1: Do not issue any type of command during DLL initialization of the DRAM devices. If any other commands are issued during the initialization time, they will be held off until DLL initialization is complete.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	INTRPTWRITEA	RW	Enables interrupting of a combined write with auto pre-charge command with another read or write command to the same bank before the first write command is completed. ● 0: Disable interrupting a combined write with auto pre-charge command with another read or write command to the same bank. ● 1: Enable interrupting a combined write with auto pre-charge command with another read or write command to the same bank.	0
[15:9]	Reserved	N/A	Reserved	0
[8]	INTRPTREADA	RW	Enables interrupting of a combined read with auto pre-charge command with another read command to the same bank before the first read command is completed. ● 0: Disable interrupting the combined read with auto pre-charge command with another read command to the same bank. ● 1: Enable interrupting the combined read with auto pre-charge command with another read command to the same bank.	0
[7:1]	Reserved	N/A	Reserved	0
[0]	INTRPTAPBURST	RW	Enables interrupting an auto pre-charge command with another command for a different bank. If enabled, the current operation will be interrupted. However, the bank will be pre-charged as if the current operation were allowed to continue. ● 0: Disable interrupting an auto pre-charge operation on a different bank. ● 1: Enable interrupting an auto pre-charge operation on a different bank.	0

Register 4-8: DRAM control register 7 (CTRL_REG_7)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	PRIORITY_EN	RW	Enables priority as a condition when using the placement logic to fill the command queue. ● 0: Disabled ● 1: Enabled	0
[23:17]	Reserved	N/A	Reserved	0
[16]	POWER_DOWN	RW	When this parameter is set to 1, the memory controller will complete processing of the current burst for the current transaction (if any), issue a pre-charge all command and then disable the clock enable signal to the DRAM devices. Any subsequent commands in the command queue will be suspended until this parameter is cleared to 0. ● 0: Enable full power state. ● 1: Disable the clock enable and power down the memory controller.	0
[15:9]	Reserved	N/A	Reserved	0
[8]	PLACEMENT_EN	RW	Enables using the placement logic to fill the command queue. ● 0: Placement logic is disabled. The command queue is a straight	0

			FIFO. ● 1: Placement logic is enabled. The command queue will be filled according to the placement logic factors.	
[7:0]	Reserved	N/A	Reserved	0

Register 4-9: DRAM control register 8 (CTRL_REG_8)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	RESYNC_DLL	W	Initiates a re-synchronization of the DLL. This parameter is write-only.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	REG_DIMM_ENABLE	RW	Enables registered DIMM operations to control the address and command pipeline of the memory controller. ● 0: Normal operation ● 1: Enable registered DIMM operation	0
[15:9]	Reserved	N/A	Reserved	0
[8]	REDUC	RW	Controls the width of the memory data width. When enabled, the upper half of the memory buses (DQ, DQS and DM) are unused and relevant data only exists in the lower half of the buses. This parameter expands the Memory Controller for use with memory devices of the configured width or half of the configured width. Note: The entire user data width is used regardless of this setting. When operating in half data width mode, only *bstlen* parameter values of 4 and 8 are supported. ● 0: Standard operation using full memory bus. ● 1: Memory data width is half of the maximum size. The upper half of the data_byte_disable bus will be driven to 1.	0
[7:1]	Reserved	N/A	Reserved	0
[0]	PWRUP_SREFRESH_EXIT	RW	Allows controller to exit power-down mode by executing a self-refresh exit instead of the full memory initialization. This parameter provides a means to skip full initialization when the DRAM devices are in a known self-refresh state. ● 0: Disabled ● 1: Enabled	0

Register 4-10: DRAM control register 9 (CTRL_REG_9)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	START	RW	With this parameter is cleared to 0, the memory controller will not issue any commands to the DRAM devices or respond to any signal activity except for reading and writing parameters. Once this parameter is set to 1, the memory controller will respond to inputs from the ASIC. When set, the memory controller begins its initialization routine. **Note:** The user must wait for the initialization complete bit to be set in the *int_status* parameter and for the dfi_init_complete signal to be asserted from the PHY before submitting any transactions. ● 0: Controller is not in active mode. ● 1: Initiate active mode for the memory controller	0
[23:17]	Reserved	N/A	Reserved	0
[16]	SREFRESH	RW	When this parameter is set to 1, the DRAM device(s) will be placed in self-refresh mode. For this, the current burst for the current transaction (if any) will complete, all banks will be closed, the self-refresh command will be issued to the DRAM, and the clock enable signal will be de-asserted. The system will remain in self-refresh mode until this parameter is cleared to 0. The DRAM devices will return to normal operating mode after the self-refresh exit time (the *txsr* parameter) of the device and any DLL initialization time for the DRAM is reached. The memory controller will resume processing of the commands from the	0

			interruption point. This parameter will be updated with an assertion of the srefresh_enter pin, regardless of the behavior on the register interface. To disable self-refresh again after a srefresh_enter pin assertion, the user will need to clear the parameter to 0. ● 0: Disable self-refresh mode. ● 1: Initiate self-refresh of the DRAM devices.	
[15:9]	Reserved	N/A	Reserved	0
[8]	RW_SAME_EN	RW	Enables read/write grouping as a condition when using the placement logic to fill the command queue. ● 0: Disabled ● 1: Enabled	0
[7:1]	Reserved	N/A	Reserved	0
[0]	RESYNC_DLL_PER_AREF_EN	RW	Enables an automatic re-synchronization of the DLL after every refresh.	0

Register 4-11: DRAM control register 10 (CTRL_REG_10)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	TREF_ENABLE	RW	Enables refresh commands. If command refresh mode is configured, then refresh commands will be automatically issued based on the tref parameter value and any refresh commands sent through the command interface or the register interface. Refreshes will still occur even if the DRAM devices have been placed in power down state by the assertion of the power_down parameter. ● 0: Refresh commands disabled. ● 1: Refresh commands enabled.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	TRAS_LOCKOUT	RW	Defines the tRAS lockout setting for the DRAM device. tRAS lockout allows the memory controller to execute auto pre-charge commands before the *tras_min* parameter has expired. ● 0: tRAS lockout not supported by memory device. ● 1: tRAS lockout supported by memory device.	0
[15:9]	Reserved	N/A	Reserved	0
[8]	SWAP_PORT_RW_SAME_EN	RW	When swapping in enabled (*swap_en* is set to 1), this parameter enables swapping between read commands on the same port from different requestors. Commands from the same requestor cannot be swapped. **Note:** While read data may be interleaved, write data must be accepted in order on a port. The Memory controller logic will not swap write commands from the same port, even if this bit is enabled. **Note:** This situation will be uncommon since all read commands on the same port enter the command queue at the priority defined in *axiY_r_priority*. However, this situation may occur through command aging. 0: Disabled, 1: Enabled	0
[7:1]	Reserved	N/A	Reserved	0
[0]	SWAP_EN	RW	Enables swapping of the active command for a new higher-priority command when using the placement logic. 0: Disabled, 1: Enabled	0

Register 4-12: DRAM control register 11 (CTRL_REG_11)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:24]	AXI0_R_PRIORITY	RW	Sets the priority of read commands from data interface 0. A value of 0 is the highest priority.	0
[23:18]	Reserved	N/A	Reserved	0
[17:16]	AXI0_FIFO_TYPE_REG	RW	Sets the relativity of the clock domains between data interface 0 and the	0

			Memory Controller core clock. It should be set to 11 for a fixed Synchronous mode.	
[15:9]	Reserved	N/A	Reserved	0
[8]	WRITE_MODEREG	W	Writes mode register information into the memory devices. The user should program the appropriate *mrY_data_X* parameters with valid information based on the memory type being used. All of the mode registers that are relevant for the memory type specified in the *dram_class* parameter will be written on each *write_modereg* setting. This parameter will always read back as 0. The mode registers are automatically written at initialization of the memory controller. There is no need to initiate a mode register write after setting the *start* parameter in the memory controller unless some value in these registers needs to be changed after initialization. **Note:** This parameter may not be changed when the memory is in power-down mode (when the CKE input is de-asserted).	0
[7:1]	Reserved	N/A	Reserved	0
[0]	WRITEINTERP	RW	Defines whether the memory controller can interrupt a write burst with a read command. Some memory devices do not allow this functionality. For DDR1 or LPDDR1 memory devices, consult the memory specification for the setting for this parameter. For DDR2memory devices, this parameter must be cleared to 0. 0: The device does not support read commands interrupting write commands. 1: The device does support read commands interrupting write commands.	0

Register 4-13: DRAM control register 12 (CTRL_REG_12)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:24]	AXI1_W_PRIORITY	RW	Sets the priority of write commands from data interface 1. A value of 0 is the highest priority.	0
[23:18]	Reserved	N/A	Reserved	0
[17:16]	AXI1_R_PRIORITY	RW	Sets the priority of read commands from data interface 1. A value of 0 is the highest priority.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	AXI1_FIFO_TYPE_REG	RW	Sets the relativity of the clock domains between data interface 1 and the Memory Controller core clock. It should be set to 11 for a fixed Synchronous mode.	0
[7:2]	Reserved	N/A	Reserved	0
[1:0]	AXI0_W_PRIORITY	RW	Sets the priority of write commands from data interface 0. A value of 0 is the highest priority.	0

Register 4-14: DRAM control register 13 (CTRL_REG_13)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:24]	AXI3_FIFO_TYPE_REG	RW	Sets the relativity of the clock domains between data interface 3 and the Memory Controller core clock. It should be set to 11 for a fixed Synchronous mode.	0
[23:18]	Reserved	N/A	Reserved	0
[17:16]	AXI2_W_PRIORITY	RW	Sets the priority of write commands from data interface 2. A value of 0 is the highest priority.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	AXI2_R_PRIORITY	RW	Sets the priority of read commands from data interface 2. A value of 0 is the highest priority.	0
[7:2]	Reserved	N/A	Reserved	0
[1:0]	AXI2_FIFO_TYPE_REG	RW	Sets the relativity of the clock domains between data interface 2 and the	0

			Memory Controller core clock. It should be set to 11 for a fixed Synchronous mode.	

Register 4-15: DRAM control register 14 (CTRL_REG_14)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:24]	AXI4_R_PRIORITY	RW	Sets the priority of read commands from data interface 4. A value of 0 is the highest priority.	0
[23:18]	Reserved	N/A	Reserved	0
[17:16]	AXI4_FIFO_TYPE_REG	RW	Sets the relativity of the clock domains between data interface 4 and the Memory Controller core clock. It should be set to 11 for a fixed Synchronous mode.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	AXI3_W_PRIORITY	RW	Sets the priority of write commands from data interface 3. A value of 0 is the highest priority.	0
[7:2]	Reserved	N/A	Reserved	0
[1:0]	AXI3_R_PRIORITY	RW	Sets the priority of read commands from data interface 3. A value of 0 is the highest priority.	0

Register 4-16: DRAM control register 15 (CTRL_REG_15)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:24]	AXI5_W_PRIORITY	RW	Sets the priority of write commands from data interface 5. A value of 0 is the highest priority.	0
[23:18]	Reserved	N/A	Reserved	0
[17:16]	AXI5_R_PRIORITY	RW	Sets the priority of read commands from data interface 5. A value of 0 is the highest priority.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	AXI5_FIFO_TYPE_REG	RW	Sets the relativity of the clock domains between data interface 5 and the Memory Controller core clock. It should be set to 11 for a fixed Synchronous mode.	0
[7:2]	Reserved	N/A	Reserved	0
[1:0]	AXI4_W_PRIORITY	RW	Sets the priority of write commands from data interface 4. A value of 0 is the highest priority.	0

Register 4-17: DRAM control register 16 (CTRL_REG_16)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:24]	LOWPOWER_REFRESH_ENABLE	RW	Sets whether refreshes will occur while the memory controller is in any of the low power modes. ● 0: Refreshes still occur ● 1: Refreshes do not occur	0
[23:18]	Reserved	N/A	Reserved	0
[17:16]	DRAM_CLK_DISABLE	RW	Sets value for the DFI output signal dfi_dram_clk_disable. Bit [0] controls CS0, Bit [1] controls CS1. For each bit: ● 0: Memory clock/s should be active. ● 1: Memory clock/s should be disabled.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	CS_MAP	RW	Sets the mask that determines which chip select pins are active, with each bit representing a different chip select. Bit [0] of this parameter corresponds to chip select [0], bit [1] corresponds to chip select [1].	0

[7:2]	Reserved	N/A	Reserved	0
[1:0]	BIST_RESULT	R	Indicates the result of the BIST operation. The meaning of the *bist_result* parameter is dependent on the status of the *bist_addr_check* and *bist_data_check* parameters. This parameter value is only valid if the BIST_DONE interrupt bit in the *int_status* parameter is asserted. The *bist_result* parameter is cleared to 00 on assertion of the *bist_go* parameter. This parameter is read-only. ● Bit [1] = Address Check Indicator -- 0: Address check is not enabled, or it is enabled and it failed. -- 1: Address check is enabled and it passed. ● Bit [0] = Data Check Indicator -- 0: Data check is not enabled, or it is enabled and it failed. -- 1: Data check is enabled and it passed.	0

Register 4-18: DRAM control register 17 (CTRL_REG_17)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	ODT_WR_MAP_CS0	RW	Sets up which (if any) chip(s) will have their ODT termination active while a write occurs on chip select 0. ● Bit [1] = CS1 will have active ODT termination when chip select 0 is performing a write. ● Bit [0] = CS0 will have active ODT termination when chip select 0 is performing a write.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	ODT_RD_MAP_CS1	RW	Sets up which (if any) chip(s) will have their ODT termination active while a read occurs on chip select 1. ● Bit [1] = CS1 will have active ODT termination when chip select 1 is performing a read. ● Bit [0] = CS0 will have active ODT termination when chip select 1 is performing a read.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	ODT_RD_MAP_CS0	RW	Sets up which (if any) chip(s) will have their ODT termination active while a read occurs on chip select 0. ● Bit [1] = CS1 will have active ODT termination when chip select 0 is performing a read. ● Bit [0] = CS0 will have active ODT termination when chip select 0 is performing a read.	0
[7:2]	Reserved	N/A	Reserved	0
[1:0]	MAX_CS_REG	R	Displays the maximum number of chip selects configured for this memory controller. This parameter is read-only。	2

Register 4-19: DRAM control register 18 (CTRL_REG_18)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:24]	ARB_CMD_Q_THRESHOLD	RW	Sets the command queue fullness that determines if ports will be allowed to overflow. This parameter is used in conjunction with the *axiX_bdw_ovflow* parameters	0
[23:19]	Reserved	N/A	Reserved	0
[18:16]	ADDR_PINS	RW	Defines the difference between the maximum number of address pins configured (15) and the actual number of pins being used. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based on the value of this parameter.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	RTT_0	RW	Defines the On-Die termination resistance for all DRAM devices. The Memory Controller cannot be set for different termination values for each	0

			chip select. **Note:** The user must program the same values into the A6/A2 bits of *mr1_data_X* and *rtt_0* parameters. ● 00: Termination Disabled ● 01: 75 Ohm ● 10: 150 Ohm ● 11: 50 Ohm	
[7:2]	Reserved	N/A	Reserved	0
[1:0]	ODT_WR_MAP_CS1	RW	Sets up which (if any) chip(s) will have their ODT termination active while a write occurs on chip select 1. ● Bit [1] = CS1 will have active ODT termination when chip select 0 is performing a write. ● Bit [0] = CS0 will have active ODT termination when chip select 0 is performing a write.	0

Register 4-20: DRAM control register 19 (CTRL_REG_19)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:24]	PORT_DATA_ERROR_TYPE	R	Defines the type of error and the access type that caused the port data error condition. If multiple bits are set to 1, then multiple errors were found. This parameter is read-only. ● Bit [2] = Reserved. ● Bit [1] = Reserved. ● Bit [0] = Reserved.	0
[23:19]	Reserved	N/A	Reserved	0
[18:16]	COLUMN_SIZE	RW	Shows the difference between the maximum column width available and the actual number of column pins being used. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based on the value of this parameter.	0
[15:11]	Reserved	N/A	Reserved	0
[10:8]	CKE_DELAY	RW	Sets the number of additional cycles of delay to include in the CKE signal cke_status for status reporting. The default delay is 0 cycles.	0
[7:3]	Reserved	N/A	Reserved	0
[2:0]	CASLAT	RW	Sets the CAS (Column Address Strobe) latency encoding that the memory uses. The binary value programmed into this parameter is dependent on the memory device, since the same *caslat* value may have different meanings to different memories. This will be programmed into the DRAM devices at initialization. The CAS encoding will be specified in the DRAM spec sheet, and should correspond to the *caslat_lin* parameter.	0

Register 4-21: DRAM control register 20 (CTRL_REG_20)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:24]	R2W_DIFFCS_DLY	RW	Defines the number of additional clocks of delay to insert from a read command to one chip select to a write command to a different chip select.	0
[23:19]	Reserved	N/A	Reserved	0
[18:16]	R2R_SAMECS_DLY	RW	Defines the number of additional clocks of delay to insert between two read commands to the same chip select.	0
[15:11]	Reserved	N/A	Reserved	0
[10:8]	R2R_DIFFCS_DLY	RW	Defines the number of additional clocks of delay to insert from a read command to one chip select to a read command to a different chip select.	0
[7:3]	Reserved	N/A	Reserved	0
[2:0]	Q_FULLNESS	RW	Defines quantity of data that will be considered full for the command queue.	0

Register 4-22: DRAM control register 21 (CTRL_REG_21)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:24]	TDFI_DRAM_CLK_DISABLE	RW	Holds the DFI $t_{dram\ clk\ disable}$ timing parameter. This parameter should be programmed with the number of cycles that the PHY requires to disable the clock after the dfi_dram_clk_disable signal is asserted.	0
[23:19]	Reserved	N/A	Reserved	0
[18:16]	TCKE	RW	Defines the minimum CKE pulse width, in cycles.	0
[15:11]	Reserved	N/A	Reserved	0
[10:8]	TBST_INT_INTERVAL	RW	Defines the burst interrupt interval. This parameter is only relevant if the burst has not completed. This value is loaded into a parameter when a burst is issued and another command may only interrupt the current burst when this counter value hits 0. If the counter value hits 0 and the burst has not completed, the counter will be reset with the *tbst_int_interval* value. If a command is in progress and the burst has not completed, another command may only be issued on cycles after the parameter *tccd* value cycles have elapsed since the last CAS command and this counter value hits 0. For example, if the parameters *bstlen* is 8, *tccd* is 2, *tbst_int_interval* is 2 and a CAS command was issued on cycle 0, another CAS command could interrupt the current burst on cycle 2. After cycle 3, the current burst will complete and this parameter would not be relevant. If instead the *tbst_int_interval* was 1 for the same system, then the command could interrupt on cycles 2 or 3.	0
[7:3]	Reserved	N/A	Reserved	0
[2:0]	R2W_SAMECS_DLY	RW	Defines the number of additional clocks of delay to insert from a read command to a write command to the same chip select.	0

Register 4-23: DRAM control register 22 (CTRL_REG_22)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:24]	W2R_SAMECS_DLY	RW	Defines the number of additional clocks of delay to insert from a write command to a read command to the same chip select.	0
[23:19]	Reserved	N/A	Reserved	0
[18:16]	W2R_DIFFCS_DLY	RW	Defines the number of additional clocks of delay to insert from a write command to one chip select to a read command to a different chip select.	0
[15:11]	Reserved	N/A	Reserved	0
[10:8]	TRTP	RW	For DDR1, this parameter has no meaning. For LPDDR1 or DDR2, defines the DRAM tRTP (read to pre-charge time) parameter, in cycles.	0
[7:3]	Reserved	N/A	Reserved	0
[2:0]	TRRD	RW	Defines the DRAM activate to activate delay for different banks, in cycles.	0

Register 4-24: DRAM control register 23 (CTRL_REG_23)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27:24]	ADD_ODT_CLK_DIFFTYPE_SAMECS	RW	Defines the number of additional clocks of delay to insert between commands of different types (read to write, write to read) to the same chip select to meet ODT timing requirements.	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	ADD_ODT_CLK_DIFFTYPE_DIFFCS	RW	Defines the number of additional clocks of delay to insert between commands of different types (read to write, write to read) to different chip selects to meet ODT timing requirements.	0

[15:11]	Reserved	N/A	Reserved	0
[10:8]	W2W_SAMECS_DLY	RW	Defines the number of additional clocks of delay to insert between two write commands to the same chip select.	0
[7:3]	Reserved	N/A	Reserved	0
[2:0]	W2W_DIFFCS_DLY	RW	Defines the number of additional clocks of delay to insert from a write command to one chip select to a write command to a different chip select.	0

Register 4-25: DRAM control register 24 (CTRL_REG_24)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27:24]	CASLAT_LIN	RW	Sets the CAS latency linear value in 1/2 cycle increments. This sets an internal adjustment for the delay from when the read command is sent from the memory controller to when data will be received back. The window of time in which the data is captured is a fixed length. The *caslat_lin* parameter adjusts the start of this data capture window. Not all linear values will be supported for the memory devices being used. Refer to the specification for the memory devices being used. ● 0000 - 0001: Reserved ● 0010: 1 cycle ● 0011: 1.5 cycles ● 0100: 2 cycles ● 0101: 2.5 cycles ● 0110: 3 cycles ● 0111: 3.5 cycles ● 1000: 4 cycles ● 1001: 4.5 cycles ● 1010: 5 cycles ● 1011: 5.5 cycles ● 1100: 6 cycles ● 1101: 6.5 cycles ● 1110: 7 cycles ● 1111: 7.5 cycles	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	APREBIT	RW	Defines the location of the auto pre-charge bit in the DRAM address in decimal encoding.	0
[15:12]	Reserved	N/A	Reserved	0
[11:8]	AGE_COUNT	RW	Holds the initial value of the master aging-rate counter. When using the placement logic to fill the command queue, the command aging counters will be decremented one each time the master aging-rate counter counts down *age_count* cycles.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	ADD_ODT_CLK_SAME TYPE_DIFFCS	RW	Defines the number of additional clocks of delay to insert between commands of the same type (read to read, write to write) to different chip selects to meet ODT timing requirements.	0

Register 4-26: DRAM control register 25 (CTRL_REG_25)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27:24]	COMMAND_AGE_COU NT	RW	Holds the initial value of the command aging counters associated with each command in the command queue. When using the placement logic to fill the command queue, the command aging counters decrement one each time the master aging-rate counter counts down the number of cycles in the *age_count* parameter.	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	CKSRX	RW	Sets the number of cycles to hold the clock stable before exiting	0

			self-refresh mode. The clock will run for a minimum of *cksrx* cycles before CKE rises.	
[15:12]	Reserved	N/A	Reserved	0
[11:8]	CKSRE	RW	Sets the number of cycles to hold the clock stable after entering self-refresh mode. The clock will run for a minimum of *cksre* cycles after CKE falls.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	CASLAT_LIN_GATE	RW	Adjusts the data capture gate open time by 1/2 cycle increments. This parameter is programmed differently than *caslat_lin* when there are fixed offsets in the flight path between the memories and the memory controller for clock gating. When *caslat_lin_gate* is a larger value than *caslat_lin*, the data capture window will become shorter. A *caslat_lin_gate* value smaller than *caslat_lin* may have no effect on the data capture window, depending on the fixed offsets in the ASIC and the board. ● 0000 - 0001: Reserved ● 0010: 1 cycle ● 0011: 1.5 cycles ● 0100: 2 cycles ● 0101: 2.5 cycles ● 0110: 3 cycles ● 0111: 3.5 cycles ● 1000: 4 cycles ● 1001: 4.5 cycles ● 1010: 5 cycles ● 1011: 5.5 cycles ● 1100: 6 cycles ● 1101: 6.5 cycles ● 1110: 7 cycles ● 1111: 7.5 cycles	0

Register 4-27: DRAM control register 26 (CTRL_REG_26)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27:24]	MAX_ROW_REG	R	Defines the maximum width of the memory address bus (number of row bits) for the memory controller. This value can be used to set the *addr_pins* parameter. This parameter is read-only. *addr_pins* = *max_row_reg* - <number of row bits in memory device>.	f
[23:20]	Reserved	N/A	Reserved	0
[19:16]	MAX_COL_REG	R	Defines the maximum width of column address in the DRAM devices. This value can be used to set the *column_size* parameter. This parameter is read-only. *column_size* = *max_col_reg* - <number of column bits in memory device>.	d
[15:12]	Reserved	N/A	Reserved	0
[11:8]	INITAREF	RW	Defines the number of auto-refresh commands needed by the DRAM devices to satisfy the initialization sequence.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	DRAM_CLASS	RW	Selects the mode of operation for the Memory Controller. ● 0000: DDR1 ● 0001: DDR1 with Mobile (LPDDR1) ● 0100: DDR2 ● All other settings Reserved.	0

Register 4-28: DRAM control register 27 (CTRL_REG_27)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0

[27:24]	TDFI_CTRL_DELAY	RW	Holds the DFI t_{ctrl_delay} timing parameter. This parameter should be programmed with the number of cycles that the PHY requires to send a power-down or self-refresh command to the DRAM devices.	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	TDFI_CTRLUPD_MIN	R	Holds the DFI $t_{ctrlupd\_min}$ timing parameter. This parameter is read-only.	4
[15:12]	Reserved	N/A	Reserved	0
[11:8]	RDLAT_ADJ	RW	Adjusts the relative timing between DFI read commands and the dfi_rddata_en signal to conform to PHY timing requirements. When this parameter is programmed to 0x0, dfi_rddata_en will assert one cycle after the dfi_address.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	PORT_CMD_ERROR_TYPE	R	Defines the type of error and the access type that caused the port command error condition. If multiple bits are set to 1, then multiple errors were found. This parameter is read-only. ● Bit [3] = Narrow transfer requested for a requestor Y whose *axiX_en_size_lt_width_instr* parameter is clear. ● Bit [2] = Reserved. ● Bit [1] = Reserved. ● Bit [0] = Reserved.	0

Register 4-29: DRAM control register 28 (CTRL_REG_28)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27:24]	TDFI_PHY_WRLAT_BASE	RW	Used to adjust the *tdfi_phy_wrlat* parameter if the PHY requires greater delay from write command to write data.	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	TDFI_PHY_WRLAT	R	Holds the calculated value of the t_{phy_wrlat} timing parameter. This equation is dependent on the latency setting for the address / control path of the PHY as set in the *phy_ctrl_reg_2* [25] parameter bit. ● If *phy_ctrl_reg_2* [25] = 0: -- If (*tdfi_phy_wrlat_base* + *wrlat_adj*) = 2: *tdfi_phy_wrlat* = *reg_dimm_enable* -- If (*tdfi_phy_wrlat_base* + *wrlat_adj*) > 2: *tdfi_phy_wrlat* = *tdfi_phy_wrlat_base* + *wrlat_adj* + *reg_dimm_enable* - WRLAT_WIDTH'h3 **Note:** Values of (*tdfi_phy_wrlat_base* + *wrlat_adj*) < 2 are not supported. ● If *phy_ctrl_reg_2* [25] = 1: -- If (*tdfi_phy_wrlat_base* + *wrlat_adj*) < 4: *tdfi_phy_wrlat* = *reg_dimm_enable* -- If (*tdfi_phy_wrlat_base* + *wrlat_adj*) >= 4: *tdfi_phy_wrlat* = *tdfi_phy_wrlat_base* + *wrlat_adj* + *reg_dimm_enable* - WRLAT_WIDTH'h4 This parameter is read-only.	0
[15:12]	Reserved	N/A	Reserved	0
[11:8]	TDFI_PHY_RDLAT	RW	Holds the t_{phy_rdlat} timing parameter.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	TDFI_DRAM_CLK_ENABLE	RW	Holds the DFI $t_{dram\_clk\_enable}$ timing parameter. This parameter is currently unused in the Memory Controller.	0

Register 4-30: DRAM control register 29 (CTRL_REG_29)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27:24]	TRP_AB	RW	Defines the DRAM TRP time for all banks, in cycles.	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	TRP	RW	Defines the DRAM pre-charge command time, in cycles.	0

[15:12]	Reserved	N/A	Reserved	0
[11:8]	TDFI_RDDATA_EN_BASE	RW	Used to adjust the *tdfi_rddata_en* parameter if the PHY requires greater delay from read command to read data enable.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	TDFI_RDDATA_EN	R	Holds the calculated value of the $t_{rddata\ en}$ timing parameter. This equation is dependent on the latency setting for the address / control path of the PHY as set in the *phy_ctrl_reg_2* [25] parameter bit. If *phy_ctrl_reg_2* [25] = 0: -- If (*tdfi_rddata_en_base* + *rdlat_adj*) = 2: *tdfi_rddata_en* = *reg_dimm_enable* -- If (*tdfi_rddata_en_base* + *rdlat_adj*) > 2: *tdfi_rddata_en* = *tdfi_rddata_en_base* + *rdlat_adj* + *reg_dimm_enable* - RDLAT_WIDTH'h3 **Note:** Values of (*tdfi_rddata_en_base* + *rdlat_adj*) < 2 are not supported. If *phy_ctrl_reg_2* [25] = 1: -- If (*tdfi_rddata_en_base* + *rdlat_adj*) < 4: *tdfi_rddata_en* = *reg_dimm_enable* -- If (*tdfi_rddata_en_base* + *rdlat_adj*) >= 4: *tdfi_rddata_en* = *tdfi_rddata_en_base* + *rdlat_adj* + *reg_dimm_enable* - RDLAT_WIDTH'h4 This parameter is read-only.	0

Register 4-31: DRAM control register 30 (CTRL_REG_30)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	N/A	Reserved	0
[28:24]	LOWPOWER_AUTO_ENABLE	RW	Enables automatic entry into the low power modes of the memory controller. ● Bit [4] = Controls memory power-down mode (Mode 1). ● Bit [3] = Controls memory power-down with memory clock gating mode (Mode 2). ● Bit [2] = Controls memory self-refresh mode (Mode 3). ● Bit [1] = Controls memory self-refresh with memory clock gating mode (Mode 4). ● Bit [0] = Controls memory self-refresh with memory and controller clock gating mode (Mode 5). For all bits: -- 0: Automatic entry into this mode is disabled. The user may enter this mode manually by setting the associated *lowpower_control* bit. -- 1: Automatic entry into this mode is enabled. The mode will be entered automatically when the proper counters expire, and only if the associated *lowpower_control* bit is set.	0
[23:20]	Reserved	N/A	Reserved	0
[19:16]	WRLAT_ADJ	RW	Adjusts the relative timing between DFI write commands and the dfi_wrdata_en signal to conform to PHY timing requirements. When this parameter is programmed to 0x0, dfi_wrdata_en will assert on the same cycle as the dfi_address.	0
[15:12]	Reserved	N/A	Reserved	0
[11:8]	WRLAT	RW	Defines the write latency from when the write command is issued to the time the write data is presented to the DRAM devices, in cycles. **Note:** This parameter must be set to 1 when the Memory Controller is used in DDR1 mode.	0
[7:4]	Reserved	N/A	Reserved	0
[3:0]	TWTR	RW	Sets the number of cycles needed to switch from a write to a read operation, as dictated by the DDR SDRAM specification.	0

Register 4-32: DRAM control register 31 (CTRL_REG_31)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	N/A	Reserved	0
[28:24]	TCCD	RW	Defines the minimum delay between CAS commands, in cycles. This value is loaded into a counter when a burst is issued and a new command may be issued when the counter reaches 0.	0
[23:21]	Reserved	N/A	Reserved	0
[20:16]	Reserved	RW	Reserved	0
[15:13]	Reserved	N/A	Reserved	0
[12:8]	Reserved	RW	Reserved	0
[7:5]	Reserved	N/A	Reserved	0
[4:0]	LOWPOWER_CONTROL	RW	Enables the individual low power modes of the device. ● Bit [4] = Controls memory power-down mode (Mode 1). ● Bit [3] = Controls memory power-down with memory clock gating mode (Mode 2). ● Bit [2] = Controls memory self-refresh mode (Mode 3). ● Bit [1] = Controls memory self-refresh with memory clock gating mode (Mode 4). ● Bit [0] = Controls memory self-refresh with memory and controller clock gating mode (Mode 5). For all bits: -- 0: Disabled. -- 1: Enabled.	0

Register 4-33: DRAM control register 32 (CTRL_REG_32)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	N/A	Reserved	0
[28:24]	TWR_INT	RW	Defines the DRAM write recovery time, in cycles.	0
[23:21]	Reserved	N/A	Reserved	0
[20:16]	TMRD	RW	Defines the minimum number of cycles required between two mode register write commands. This is the time required to complete the write operation to the mode register.	0
[15:13]	Reserved	N/A	Reserved	0
[12:8]	TDAL	RW	Defines the auto pre-charge write recovery time when auto pre-charge is enabled (the *ap* parameter is set to 1), in cycles. This is defined internally as tRP (pre-charge time) + auto pre-charge write recovery time. Not all memories use this parameter. If tDAL is defined in the memory specification, then program this parameter to the specified value. If the memory does not specify a tDAL time, then program this parameter to tWR + tRP. DO NOT program this parameter with a value of 0x0 or the memory controller will not function properly when auto pre-charge is enabled.	0
[7:5]	Reserved	N/A	Reserved	0
[4:0]	TCKESR	RW	Defines the minimum number of cycles that CKE must be held low during self-refresh. Pulse width, in cycles. If the memory specification does not define a t_{ckesr}, then the t_{ckesr} parameter should be programmed with the tcke value.	0

Register 4-34: DRAM control register 33 (CTRL_REG_33)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	TRC	RW	Defines the DRAM period between active commands for the same bank, in cycles.	0
[23:17]	Reserved	N/A	Reserved	0

[16]	TFAW	RW	Defines the DRAM tFAW parameter, in cycles.	0
[15:9]	Reserved	N/A	Reserved	0
[8]	OUT_OF_RANGE_TYPE	R	Holds the type of command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.	0
[7:6]	Reserved	N/A	Reserved	0
[5:0]	ADDR_SPACE	RW	Defines the address space in bytes from 0 to 2^{addr_space} that the BIST logic will check. As an example, if *addr_space* was set to 0x1c, then the BIST logic would check 2^{28} bytes = 256 MBytes. **Note:** When performing address checking, the user should program the *addr_space* parameter carefully. If the user is checking addresses to the end of the memory map, the *addr_space* parameter should be programmed to 1 less than the maximum space.	0

Register 4-35: DRAM control register 34 (CTRL_REG_34)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:24]	AXI1_CURRENT_BDW	R	Holds the current bandwidth usage of data interface 0 as calculated by the Arbiter. The percentage will be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100. This parameter is read-only.	0
[23]	Reserved	N/A	Reserved	0
[22:16]	AXI1_BDW	RW	Sets the maximum bandwidth allocation percentage for data interface 1. The percentage must be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100.	0
[15]	Reserved	N/A	Reserved	0
[14:8]	AXI0_CURRENT_BDW	R	Holds the current bandwidth usage of data interface 0 as calculated by the Arbiter. The percentage will be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100. This parameter is read-only.	0
[7]	Reserved	N/A	Reserved	0
[6:0]	AXI0_BDW	RW	Sets the maximum bandwidth allocation percentage for data interface 0. The percentage must be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100.	0

Register 4-36: DRAM control register 35 (CTRL_REG_35)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:24]	AXI3_CURRENT_BDW	R	Holds the current bandwidth usage of data interface 3 as calculated by the Arbiter. The percentage will be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100. This parameter is read-only.	0
[23]	Reserved	N/A	Reserved	0
[22:16]	AXI3_BDW	RW	Sets the maximum bandwidth allocation percentage for data interface 3. The percentage must be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100.	0
[15]	Reserved	N/A	Reserved	0
[14:8]	AXI2_CURRENT_BDW	R	Holds the current bandwidth usage of data interface 2 as calculated by the Arbiter. The percentage will be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100. This parameter is read-only.	0
[7]	Reserved	N/A	Reserved	0
[6:0]	AXI2_BDW	RW	Sets the maximum bandwidth allocation percentage for data interface 2. The percentage must be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100.	0

Register 4-37: DRAM control register 36 (CTRL_REG_36)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:24]	AXI5_CURRENT_BDW	R	Holds the current bandwidth usage of data interface 5 as calculated by the Arbiter. The percentage will be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100. This parameter is read-only.	0
[23]	Reserved	N/A	Reserved	0
[22:16]	AXI5_BDW	RW	Sets the maximum bandwidth allocation percentage for data interface 5. The percentage must be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100.	0
[15]	Reserved	N/A	Reserved	0
[14:8]	AXI4_CURRENT_BDW	R	Holds the current bandwidth usage of data interface 4 as calculated by the Arbiter. The percentage will be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100. This parameter is read-only.	0
[7]	Reserved	N/A	Reserved	0
[6:0]	AXI4_BDW	RW	Sets the maximum bandwidth allocation percentage for data interface 4. The percentage must be specified as a hex value (0x01 - 0x64) representing a decimal percentage value from 1 - 100.	0

Register 4-38: DRAM control register 37 (CTRL_REG_37)

Field	Symbol	Direction	Description	Default
[31:24]	TMOD	RW	Defines the number of cycles of wait time after a mode register write to any non-mode register write command. For write leveling, this is defined as the number of cycles of wait time after a MRS command to the ODT enable.	0
[23:16]	DLL_RST_ADJ_DLY	RW	Specifies the minimum number of cycles after the master delay value is programmed before the DLL reset may be asserted.	0
[15:8]	DLL_LOCK	R	Defines the actual number of delay elements used to capture one full clock cycle. This parameter is automatically updated every time a refresh operation is performed. This parameter is read-only.	0
[7]	Reserved	N/A	Reserved	0
[6:0]	OUT_OF_RANGE_LENGTH	R	Holds the length of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.	0

Register 4-39: DRAM control register 38 (CTRL_REG_38)

Field	Symbol	Direction	Description	Default
[31:24]	Reserved	N/A	Reserved	0
[23:16]	TRFC	RW	Defines the DRAM refresh command time, in cycles.	
[15:8]	TRCD_INT	RW	Defines the DRAM RAS to CAS delay, in cycles.	
[7:0]	TRAS_MIN	RW	Defines the DRAM minimum row activate time, in cycles.	0

Register 4-40: DRAM control register 39 (CTRL_REG_39)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24:16]	PORT_CMD_ERROR_ID	R	Holds the source ID of the command that caused a port command error condition. This parameter is read-only.	0
[15:9]	Reserved	N/A	Reserved	0
[8:0]	OUT_OF_RANGE_SOU	R	Holds the Source ID of the command that caused an out-of-range	0

	RCE_ID		interrupt request to the memory devices. This parameter is read-only.	

Register 4-41: DRAM control register 40 (CTRL_REG_40)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:16]	INT_ACK	W	Controls the clearing of the *int_status* parameter. If any of the *int_ack* bits are set to 1, the corresponding bit in the *int_status* parameter will be cleared to 0. Any *int_ack* bits cleared to 0 will not alter the corresponding bit in the *int_status* parameter. This parameter will always read back as 0x0.	0
[15:9]	Reserved	N/A	Reserved	0
[8:0]	PORT_DATA_ERROR_ID	R	Holds the source ID of the command that caused a port data error condition. This parameter is read-only.	0

Register 4-42: DRAM control register 41 (CTRL_REG_41)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26:16]	INT_STATUS	R	Shows the status of all possible interrupts generated by the memory controller. The MSB is the result of a logical OR of all the lower bits. This parameter is read-only. **Note:** Backwards compatibility is available for register parameters across configurations. However, even with this compatibility, the individual bits, their meaning and the size of the *int_status* parameter may change. The *int_status* bits correspond to these interrupts: ● Bit [11] = Logical OR of all lower bits. ● Bit [10] = User-initiated DLL resync is finished. ● Bit [9] = DLL lock state change condition detected. (i.e. lock to unlock or unlock to lock) ● Bit [8] = Indicates that a read DQS gate error occurred. ● Bit [7] = ODT enabled and CAS Latency 3 programmed error detected. This is an unsupported programming option. ● Bit [6] = Both DDR2 and Mobile modes have been enabled. ● Bit [5] = BIST Complete. ● Bit [4] = DRAM initialization complete. ● Bit [3] = Error was found with command data channel in a port. ● Bit [2] = Error was found with command channel in a port. ● Bit [1] = Multiple accesses outside the defined PHYSICAL memory space detected. ● Bit [0] = A single access outside the defined PHYSICAL memory space detected.	0
[16:11]	Reserved	N/A	Reserved	0
[10:0]	INT_MASK	RW	Active-high mask bits that control the value of the memory controller_int signal on the ASIC interface. Unless the user has suppressed interrupt reporting (by setting bit [11] of this parameter to 1), bits [10:0] of the *int_mask* parameter will be inverted and logically AND'ed with bits [10:0] of the *int_status* parameter and the result is reported on the controller_int signal.	0

Register 4-43: DRAM control register 42 (CTRL_REG_42)

Field	Symbol	Direction	Description	Default
[31:30]	Reserved	N/A	Reserved	0
[29:16]	TDFI_PHYUPD_RESP	RW	Holds the DFI t_{phyupd_resp} timing parameter. This parameter is read-only.	0
[15:14]	Reserved	N/A	Reserved	0
[13:0]	TDFI_CTRLUPD_MAX	RW	Holds the DFI $t_{ctrlupd\_max}$ timing parameter. This parameter is read-only.	0

Register 4-44: DRAM control register 43 (CTRL_REG_43)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	TDFI_PHYUPD_TYPE1	RW	Holds the DFI t_{phyupd_type1} timing parameter. This parameter is read-only.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	TDFI_PHYUPD_TYPE0	RW	Holds the DFI t_{phyupd_type0} timing parameter. This parameter is read-only.	0

Register 4-45: DRAM control register 44 (CTRL_REG_44)

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	TDFI_PHYUPD_TYPE3	RW	Holds the DFI t_{phyupd_type3} timing parameter. This parameter is read-only.	0
[23:17]	Reserved	N/A	Reserved	0
[16]	TDFI_PHYUPD_TYPE2	RW	Holds the DFI t_{phyupd_type2} timing parameter. This parameter is read-only.	0

Register 4-46: DRAM control register 45 (CTRL_REG_45)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:16]	MR0_DATA_0	RW	Holds the memory mode register 0 data for chip select 0 written during memory initialization. Consult the memory specification for the fields of this mode register. This parameter correlates to the memory mode register (MR). The use of this parameter varies based on the memory type connected to this memory controller: ● For DDR1 memories: The memory controller does not support interleaving and therefore the A3 bit should be cleared to 0. In addition, the DLL Reset bit (A8) will be ignored in favor of an internal state machine that sets the DLL Reset bit during initialization. Also, the memory controller only supports a burst length of 4 and therefore the bits A2:A0 should be set to 010.For DDR2 memories: The memory controller does not support interleaving and therefore the A3 bit should be cleared to 0. In addition, the DLL Reset bit (A8) will be ignored in favor of an internal state machine that sets the DLL Reset bit during initialization. Also, the memory controller only supports a burst length of 4 and therefore the bits A2:A0 should be set to 010. ● For LPDDR1 memories: The memory controller does not support interleaving and therefore the A3 bit should be cleared to 0. Also, the memory controller only supports a burst length of 4 and therefore the bits A2:A0 should be set to 010. This data will be programmed into the memory register of the DRAM at initialization or when the *write_modereg* parameter is set to 1.	0
[15:14]	Reserved	N/A	Reserved	0
[13:0]	TREF	RW	Defines the DRAM cycles between refresh commands.	0

Register 4-47: DRAM control register 46 (CTRL_REG_46)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:16]	MR1_DATA_0	RW	Holds the memory mode register 1 data for chip select 0 written during memory initialization. Consult the memory specification for the fields of this mode register. The use of this parameter varies based on the memory type connected to this memory controller:	0

			For DDR1 memories: This parameter correlates to the extended memory mode register (EMR). For DDR2 memories: This parameter correlates to the extended memory mode register 1 (EMR1). The memory controller does not support additive latency and therefore the A5:A3 bits should be cleared to 000. In addition, the user must program the same values into the A6/A2 bits of this parameter and the parameter *rtt_0*. In addition, the user must program the same values into the A6/A2 bits of this parameter and the parameter *rtt_0*.For LPDDR1 memories: This parameter has no meaning for this memory type. This data will be programmed into the appropriate memory register of the DRAM at initialization or when the *write_modereg* parameter is set to 1.	
[15]	Reserved	N/A	Reserved	0
[14:0]	MR0_DATA_1	RW	Holds the memory mode register 0 data for chip select 1 written during memory initialization. Reference MR0_DATA_0.	0

Register 4-48: DRAM control register 47 (CTRL_REG_47)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:16]	MR2_DATA_0	RW	Holds the memory mode register 2 data for chip select 0 written during memory initialization. Consult the memory specification for the fields of this mode register. The use of this parameter varies based on the memory type connected to this memory controller: ● For DDR1 memories: This parameter has no meaning for this memory type. ● For DDR2 memories: This parameter correlates to the extended memory mode register 2 (EMR2). ● For LPDDR1 memories: This parameter correlates to the extended mode register (EMR). This data will be programmed into the appropriate memory register of the DRAM at initialization or when the *write_modereg* parameter is set to 1.	0
[15]	Reserved	N/A	Reserved	0
[14:0]	MR1_DATA_1	RW	Holds the memory mode register 1 data for chip select 0 written during memory initialization. Consult the memory specification for the fields of this mode register. Reference MR1_DATA_0	0

Register 4-49: DRAM control register 48 (CTRL_REG_48)

Field	Symbol	Direction	Description	Default
[31]	Reserved	N/A	Reserved	0
[30:16]	MR3_DATA_0	RW	Holds the memory mode register 3 data for chip select 0 written during memory initialization. Consult the memory specification for the fields of this mode register. The use of this parameter varies based on the memory type connected to this memory controller: ● For DDR1 memories: This parameter has no meaning for this memory type. ● For DDR2 memories: This parameter correlates to the extended memory mode register 3 (EMR3). ● For LPDDR1 memories: This parameter has no meaning for this memory type. This data will be programmed into the appropriate memory register of the DRAM at initialization or when the *write_modereg* parameter is set to 1.	0
[15]	Reserved	N/A	Reserved	0
[14:0]	MR2_DATA_1	RW	Holds the memory mode register 2 data for chip select 1 written during memory initialization. Consult the memory specification for the fields of this mode register. Reference MR2_DATA_0.	0

Register 4-50: DRAM control register 49 (CTRL_REG_49)

Field	Symbol	Direction	Description	Default
[31:16]	AXI0_EN_SIZE_LT_WIDTH_INSTR	RW	Allows the port to accept size less than width transactions on data interface 0 from requestor Z. Each bit Z corresponds to requestor Z, meaning that if bit [0] is set, then requestor 0 will be able to send size less than width transactions. Example: if there are 5 requestors on a port and requestors 1, 3 and 4 were able to send size less than width transactions, then this parameter would be set to 0x001A. ● 0: Requestor Z of port Y may only issue size equal to width instructions. ● 1: Requestor Z of port Y can issue size equal to width or size less than width instructions.	0
[15]	Reserved	N/A	Reserved	0
[14:0]	MR3_DATA_1	RW	Holds the memory mode register 3 data for chip select 1 written during memory initialization. Consult the memory specification for the fields of this mode register. Reference MR3_DATA_0	0

Register 4-51: DRAM control register 50 (CTRL_REG_50)

Field	Symbol	Direction	Description	Default
[31:16]	AXI2_EN_SIZE_LT_WIDTH_INSTR	RW	Allows the port to accept size less than width transactions on data interface 2 from requestor Z. reference AXI0_EN_SIZE_LT_WIDTH_INSTR	0
[15:0]	AXI1_EN_SIZE_LT_WIDTH_INSTR	RW	Allows the port to accept size less than width transactions on data interface 1 from requestor Z. reference AXI0_EN_SIZE_LT_WIDTH_INSTR	0

Register 4-52: DRAM control register 51 (CTRL_REG_51)

Field	Symbol	Direction	Description	Default
[31:16]	AXI4_EN_SIZE_LT_WIDTH_INSTR	RW	Allows the port to accept size less than width transactions on data interface 4 from requestor Z. reference AXI0_EN_SIZE_LT_WIDTH_INSTR	0
[15:0]	AXI3_EN_SIZE_LT_WIDTH_INSTR	RW	Allows the port to accept size less than width transactions on data interface 3 from requestor Z. reference AXI0_EN_SIZE_LT_WIDTH_INSTR	0

Register 4-53: DRAM control register 52 (CTRL_REG_52)

Field	Symbol	Direction	Description	Default
[31:25]	DLL_RST_DELAY	RW	Sets the number of cycles that the reset must be held asserted for the DLL.	0
[15:0]	AXI5_EN_SIZE_LT_WIDTH_INSTR	RW	Allows the port to accept size less than width transactions on data interface 5 from requestor Z. reference AXI0_EN_SIZE_LT_WIDTH_INSTR	0

Register 4-54: DRAM control register 53 (CTRL_REG_53)

Field	Symbol	Direction	Description	Default
[31:16]	LOWPOWER_INTERNAL_CNT	RW	Counts the number of idle cycles before memory self-refresh with memory and controller clock gating low power mode.	0

[15:0]	LOWPOWER_EXTERNAL_CNT	RW	Counts the number of idle cycles before memory self-refresh with memory clock gating low power mode.	0

Register 4-55: DRAM control register 54 (CTRL_REG_54)

Field	Symbol	Direction	Description	Default
[31:16]	LOWPOWER_REFRESH_HOLD	RW	Sets the number of cycles that the Memory Controller will wait before attempting to re-lock the DLL when using the controller clock gating mode low power mode. This counter will ONLY be used in this mode, the deepest low power mode. When this counter expires, the DLL will be un-gated for at least 16 cycles during which the DLL will attempt to re-lock. After 16 cycles have elapsed and the DLL has locked, then the DLL controller clock will be gated again and the counter will reset to this value. If the DLL requires more than 16 cycles to re-lock, then the un-gated time will be longer.	0
[15:0]	LOWPOWER_POWER_DOWN_CNT	RW	Counts the number of idle cycles before memory power-down or power-down with memory clock gating low power mode.	0

Register 4-56: DRAM control register 55 (CTRL_REG_55)

Field	Symbol	Direction	Description	Default
[31:16]	TCPD	RW	Defines the clock enable to pre-charge delay time for the DRAM devices, in cycles.	0
[15:0]	LOWPOWER_SELF_REFRESH_CNT	RW	Counts the number of cycles to the next memory self-refresh low power mode	0

Register 4-57: DRAM control register 56 (CTRL_REG_56)

Field	Symbol	Direction	Description	Default
[31:16]	TPDEX	RW	Defines the DRAM power-down exit command period, in cycles.	0
[15:0]	TDLL	RW	Defines the DRAM DLL lock time, in cycles.	0

Register 4-58: DRAM control register 57 (CTRL_REG_57)

Field	Symbol	Direction	Description	Default
[31:16]	TXSNR	RW	Defines the DRAM time from a self-refresh exit to a command that requires the memory DLL to be locked.	0
[15:0]	TRAS_MAX	RW	Defines the DRAM maximum row active time, in cycles.	0

Register 4-59: DRAM control register 58 (CTRL_REG_58)

Field	Symbol	Direction	Description	Default
[31:16]	VERSION	R	Holds the version number for this controller. This parameter is read-only.	2041
[15:0]	TXSR	RW	Defines the DRAM time from a self-refresh exit to a command that does not require the memory DLL to be locked. (txs)	0

Register 4-60: DRAM control register 59 (CTRL_REG_59)

Field	Symbol	Direction	Description	Default
[31:24]	Reserved	N/A	Reserved	0
[23:0]	TINIT	RW	Defines the DRAM initialization time, in cycles.	0

Register 4-61: DRAM control register 60 (CTRL_REG_60)

field	symbol	Direction	description	default
[31:0]	BIST_DATA_MASK	RW	Sets the data mask that will be applied to the BIST error logic. Each bit in this parameter corresponds to a bit in the memory data path. ● 0: Do not mask error checking on this bit. ● 1: Mask error checking on this bit.	0

Register 4-62: DRAM control register 61 (CTRL_REG_61)

Field	Symbol	Direction	Description	Default
[31:0]	DFT_CTRL_REG	RW	Enables the PHY testing mode. Bits [2:1] are used. ● Bits [2:1] = 10: Normal Mode ● Bits [2:1] = 01: Scan Mode Currently Not Functional	0

Register 4-63: DRAM control register 62 (CTRL_REG_62)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_0_0	RW	There is a separate dll_ctrl_reg_0_0 parameters for the slices 0(7:0) of data sent on the DFI data bus. ● Bit [28] = DLL Bypass Control. -- 0: Normal operational mode. In this mode, the DLL uses dll_ctrl_reg_0_X [14:8] for the read DQS and dll_ctrl_reg_1_X [14:8] for clk_wr. -- 1: Bypass Mode is on. In this mode, the DLL uses dll_ctrl_reg_0_X [23:15] for the read DQS and dll_ctrl_reg_1_X [23:15] for clk_wr. ● Bits [23:15] = Holds the read DQS delay setting when the DLL is operating in bypass mode. (dll_ctrl_reg_0_X [28] = 1) ● Bits [14:8] = Holds the read DQS delay setting when the DLL is operating in normal mode. (dll_ctrl_reg_0_X [28] = 0) Typically, this value is 1/ 4 of a clock cycle. Each increment of this field represents 1/128th of a clock cycle. ● Bits [7:0] = DLL Start Point Control. This value is loaded into the DLL at initialization and is the value at which the DLL will begin searching for a lock. Each increment of this field represents 1/128th of a clock cycle. ● All other bits Reserved.	0

Register 4-64: DRAM control register 63 (CTRL_REG_63)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_0_1	RW	There is a separate dll_ctrl_reg_0_1 parameters for the slices 1(15:8) of data sent on the DFI data bus. Reference DLL_CTRL_REG_0_0	0

Register 4-65: DRAM control register 64 (CTRL_REG_64)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_0_2	RW	There is a separate dll_ctrl_reg_0_2 parameters for the slices 2(23:16) of data sent on the DFI data bus. Reference DLL_CTRL_REG_0_0	0

Register 4-66: DRAM control register 65 (CTRL_REG_65)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_0_3	RW	There is a separate dll_ctrl_reg_0_3 parameters for the slices 0(31:24) of data sent on the DFI data bus. Reference DLL_CTRL_REG_0_0	0

Register 4-67: DRAM control register 66 (CTRL_REG_66)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_1_0	RW	There is a separate dll_ctrl_reg_1_0 parameter for the slices 0 (7:0) of data sent on the DFI data bus. ● Bits [23:15] = Holds the clk_wr delay setting when the DLL is operating in bypass mode. (dll_ctrl_reg_0_X [28] = 1) Each increment of this field represents 1/ 128th of a clock cycle. ● Bits [14:8] = Holds the clk_wr delay setting in normal mode. (dll_ctrl_reg_0_X [28] = 0) Typically, this value is 3/4 of a clock cycle. Each increment of this field represents 1/128th of a clock cycle. ● Bits [7:0] = DLL Increment Value. This sets the increment used by the DLL when searching for a lock. It's better to keep this field small (around 0x4) to keep the steps gradual. ● All other bits Reserved.	0

Register 4-68: DRAM control register 67 (CTRL_REG_67)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_1_1	RW	There is a separate dll_ctrl_reg_1_1 parameter for the slices 1 (15:8) of data sent on the DFI data bus. Reference DLL_CTRL_REG_1_0.	0

Register 4-69: DRAM control register 68 (CTRL_REG_68)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_1_2	RW	There is a separate dll_ctrl_reg_1_2 parameter for the slices 1 (23:16) of data sent on the DFI data bus. Reference DLL_CTRL_REG_1_0.	0

Register 4-70: DRAM control register 69 (CTRL_REG_69)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_CTRL_REG_1_3	RW	There is a separate dll_ctrl_reg_1_3 parameter for the slices 1 (31:24) of data sent on the DFI data bus. Reference DLL_CTRL_REG_1_0.	0

Register 4-71: DRAM control register 70 (CTRL_REG_70)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_OBS_REG_0_0	R	There is a separate dll_obs_reg_0_0 parameter for the slices 0 (7:0) of data sent on the DFI data bus. ● Bits [31:1] = Reports the DLL encoder value from the master DLL to the slave DLL's. The slaves use this value to set up their delays for the clk_wr and read DQS signals. ● Bit [0] = DLL Lock Indicator. -- 0: DLL has not locked. -- 1: DLL is locked.	0

Register 4-72: DRAM control register 71 (CTRL_REG_71)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_OBS_REG_0_1	R	There is a separate dll_obs_reg_0_1 parameter for the slices 1 (15:8) of data sent on the DFI data bus. ● Bits [31:1] = Reports the DLL encoder value from the master DLL	0

			to the slave DLL's. The slaves use this value to set up their delays for the clk_wr and read DQS signals. ● Bit [0] = DLL Lock Indicator. -- 0: DLL has not locked. -- 1: DLL is locked.	

Register 4-73: DRAM control register 72 (CTRL_REG_72)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_OBS_REG_0_2	R	There is a separate dll_obs_reg_0_2 parameter for the slices 2 (23:16) of data sent on the DFI data bus. ● Bits [31:1] = Reports the DLL encoder value from the master DLL to the slave DLL's. The slaves use this value to set up their delays for the clk_wr and read DQS signals. ● Bit [0] = DLL Lock Indicator. -- 0: DLL has not locked. -- 1: DLL is locked.	0

Register 4-74: DRAM control register 73 (CTRL_REG_73)

Field	Symbol	Direction	Description	Default
[31:0]	DLL_OBS_REG_0_3	R	There is a separate dll_obs_reg_0_3 parameter for the slices 3 (31:24) of data sent on the DFI data bus. ● Bits [31:1] = Reports the DLL encoder value from the master DLL to the slave DLL's. The slaves use this value to set up their delays for the clk_wr and read DQS signals. ● Bit [0] = DLL Lock Indicator. -- 0: DLL has not locked. -- 1: DLL is locked.	0

Register 4-75: DRAM control register 74 (CTRL_REG_74)

Field	Symbol	Direction	Description	Default
[31:0]	PAD_CTRL_REG_0	RW	These bit settings are specific to the pad models included with this PHY. The user should alter the programming relative to the particular pads being used in their design. ● Bit[31] = LVCMOS select bit, 1 for 3.3V LVTTL, 1.8V MDDR RX application and 0 for 2.5V SSTL2 class I, class II and 1.8V SSTL18 differential RX application ● Bit[30:28] = Reserved ● Bit[27:26] = CLK/ADR/BNK/CS/ODT/CKE/WE/RAS/CAS signal ODT control, 00=disable, 01=75ohm, 10=150 ohm, 11=Reserved ● Bit[25:24] = Reserved ● Bit[23:22]=CAS/WE/RAS/CKE/ADDR/BNK/ODT/CS output driving strength indicate. For DDR2, 00=Full strength, 10=Half strength. For DDR, 00 =Class II, 10=Class I For MDDR, 00=10mA, 01=8mA, 10=4mA, 11=2mA ● Bit[21:20] = DQ/DQS/DM/CLK output driving strength indicate. For DDR2, 00=Full strength, 10=Half strength. For DDR, 00 =Class II, 10=Class I For MDDR, 00=10mA, 01=8mA, 10=4mA, 11=2mA ● Bit[19:9] = Reserved ● Bit[8] = DS, 1 for SSTL2 differential application, 0 for SSTL18 differential application	0

			• Bit[7:4] = Reserved • Bit[3:2] = DQ/DQS/DM read ODT control, 00=disable, 01=75ohm, 10=150 ohm, 11=Reserved • Bit[1:0] = DQ/DQS/DM write ODT control, 00=disable, 01=75ohm, 10=150 ohm, 11=Reserved	

Register 4-76: DRAM control register 75 (CTRL_REG_75)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_0_0	RW	There is a separate phy_ctrl_reg_0_0 parameter for the slices 0 (7:0) of data sent on the DFI data bus. • Bit [31] = Enables dynamic termination select in the PHY for the DM pads. -- 0: Disabled -- 1: Enabled • Bit [29] = Controls termination enable for the DM pads. Set to 1 to disable termination. -- 0: Enabled -- 1: Disabled • Bits [28] = Echo gate control for data slice X. Default 0x0. -- 0: Uses the dfi_rddata_en signal to create a gate. -- 1: Creates an echo_gate signal. • Bit [27] = Gather FIFO Enable -- 0: Disabled -- 1: Enabled • Bits [26:24] = Defines the read data delay. Holds the number of cycles to delay the dfi_rddata_en signal prior to enabling the read FIFO. After this delay, the read pointers begin incrementing the read FIFO. Default 0x3. • Bit [20] = Sets the pad output enable polarity. Default 0x0. -- 0: OEN pad -- 1: OE pad • Bit [16] = Adjusts the DQS gate by 1/2 cycle. Default 0x0. -- 0: Do not adjust -- 1: Adjust • Bits [14:12] = Adjusts the starting point of the DQS pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Default 0x2. • Bits [10:8] = Adjusts the ending point of the DQS pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Default 0x7. • Bits [6:4] = Adjusts the starting point of the DQ pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Default 0x1. • Bits [2:0] = Adjusts the ending point of the DQ pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Default 0x4. • All other bits Reserved.	0

Register 4-77: DRAM control register 76 (CTRL_REG_76)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_0_1	RW	There is a separate phy_ctrl_reg_0_1 parameter for the slices 1 (15:8) of data sent on the DFI data bus.	0

Register 4-78: DRAM control register 77 (CTRL_REG_77)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_0_2	RW	There is a separate phy_ctrl_reg_0_2 parameter for the slices 2 (23:16) of data sent on the DFI data bus.	0

Register 4-79: DRAM control register 78 (CTRL_REG_78)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_0_3	RW	There is a separate phy_ctrl_reg_0_3 parameter for the slices 3 (31:24) of data sent on the DFI data bus.	0

Register 4-80: DRAM control register 79 (CTRL_REG_79)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_1_0	RW	There is a separate phy_ctrl_reg_1_0 parameter for the slices 0 (7:0) of data sent on the DFI data bus. ● Bits [31:28] = Defines the pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. ● Bits [27:24] = Defines the pad dynamic termination select disable time. Larger values reduce the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. ● Bit [23] = Enables dynamic termination select in the PHY for the DQS pads. -- 0: Disabled -- 1: Enabled ● Bit [22] = Controls the polarity of the tsel signal for the DQS and DM pads. -- 0: Negative Polarity -- 1: Positive Polarity ● Bit [21] = Triggers a data return to the Memory Controller. -- 0: No action -- 1: Sends loopback data on the dfi_rddata signal. ● Bit [20] = Selects data output type for phy_obs_reg_0_X [23:8]. -- 0: Return the expected data. -- 1: Return the actual data. ● Bits [19:18] = Loopback control. -- 00: Normal operational mode. -- 01: Enables loopback write mode. -- 10: Stop loopback to check the error register. -- 11: Clear loopback registers. ● Bits [17] = Controls the loopback read multiplexer. ● Bits [16] = Controls the internal write multiplexer. ● Bits [14:12] = Sets the cycle delay between the LFSR and loopback error check logic. Note that ‘h7 is not a valid selection and will result in a false passing result. ● Bits [10:8] = Stretches the trailing edge of dfi_rddata_en for closing the gate. ● Bits [7:0] = Controls when the DQS gate opens. This value is the number of cycles to delay the dfi_rddata_en signal prior to opening	0

			the gate. All other bits Reserved.	

Register 4-81: DRAM control register 80 (CTRL_REG_80)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_1_1	RW	There is a separate phy_ctrl_reg_1_1 parameter for the slices 1 (15:8) of data sent on the DFI data bus.	0

Register 4-82: DRAM control register 81 (CTRL_REG_81)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_1_2	RW	There is a separate phy_ctrl_reg_1_2 parameter for the slices 2 (23:16) of data sent on the DFI data bus.	0

Register 4-83: DRAM control register 82 (CTRL_REG_82)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_1_3	RW	There is a separate phy_ctrl_reg_1_3 parameter for the slices 3 (31:24) of data sent on the DFI data bus.	0

Register 4-84: DRAM control register 83 (CTRL_REG_83)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_CTRL_REG_2	RW	Controls PHY functionality. ● Bit [25] = Selects the address / command path. -- 0: DFI control signals are captured 1 cycle after being sent from the Memory Controller. -- 1: DFI control signals are captured cycle after being sent from the Memory Controller. ● Bit [24] = Selects the write latency path. -- 0: Write Data is captured 1 cycle after being sent from the Memory Controller -- 1: Write data is captured cycle after being sent from the Memory Controller. ● Bit [23] = DFI control for Mobile Memory Controller. ● Bit [5] = Enables the pad inputs specifically for external loopback. -- 0: Normal Operation -- 1: Loopback Mode ● Bit [4] = Enables the pad outputs specifically for external loopback. -- 0: Normal Operation -- 1: Loopback Mode ● Bits [3:0] = Sets the dfi_rddata_valid delay relative to dfi_rddata_en. ● All other bits Reserved.	0

Register 4-85: DRAM control register 84 (CTRL_REG_84)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_OBS_REG_0_0	R	Reports status for the PHY. ● Bit [24] = Status signal to indicate that the logic gate had to be forced closed. -- 0: Normal operation -- 1: Gate close was forced ● Bits [23:8] = Loopback data. Reports the actual data or expected data, depending on the setting of the phy_ctrl_reg_1_0 [20]	0

			parameter bit. ● Bits [5:4] = Loopback mask data. Reports the actual data mask or the expected data mask, depending on the setting of the phy_ctrl_reg_1_0 [20] parameter bit. ● Bit [1] = Reports status of loopback errors. -- 0: Last Loopback test had no errors. -- 1: Last Loopback test contained data errors. ● Bit [0] = Defines the status of the loopback mode. -- 0: Not in loopback mode. -- 1: In Loopback mode. ● All other bits Reserved.	

Register 4-86: DRAM control register 85 (CTRL_REG_85)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_OBS_REG_0_1	R	Reports status for the PHY. ● Bit [24] = Status signal to indicate that the logic gate had to be forced closed. 0: Normal operation , 1: Gate close was forced ● Bits [23:8] = Loopback data. Reports the actual data or expected data, depending on the setting of the phy_ctrl_reg_1_1 [20] parameter bit. ● Bits [5:4] = Loopback mask data. Reports the actual data mask or the expected data mask, depending on the setting of the phy_ctrl_reg_1_1 [20] parameter bit. ● Bit [1] = Reports status of loopback errors. -- 0: Last Loopback test had no errors. -- 1: Last Loopback test contained data errors. ● Bit [0] = Defines the status of the loopback mode. -- 0: Not in loopback mode. -- 1: In Loopback mode. ● All other bits Reserved.	0

Register 4-87: DRAM control register 86 (CTRL_REG_86)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_OBS_REG_0_2	R	Reports status for the PHY. ● Bit [24] = Status signal to indicate that the logic gate had to be forced closed. -- 0: Normal operation -- 1: Gate close was forced ● Bits [23:8] = Loopback data. Reports the actual data or expected data, depending on the setting of the phy_ctrl_reg_1_2 [20] parameter bit. ● Bits [5:4] = Loopback mask data. Reports the actual data mask or the expected data mask, depending on the setting of the phy_ctrl_reg_1_2 [20] parameter bit. ● Bit [1] = Reports status of loopback errors. -- 0: Last Loopback test had no errors. -- 1: Last Loopback test contained data errors. ● Bit [0] = Defines the status of the loopback mode. -- 0: Not in loopback mode. -- 1: In Loopback mode. ● All other bits Reserved.	0

Register 4-88: DRAM control register 87 (CTRL_REG_87)

Field	Symbol	Direction	Description	Default
[31:0]	PHY_OBS_REG_0_3	R	Reports status for the PHY.	0

			• Bit [24] = Status signal to indicate that the logic gate had to be forced closed. -- 0: Normal operation -- 1: Gate close was forced • Bits [23:8] = Loopback data. Reports the actual data or expected data, depending on the setting of the phy_ctrl_reg_1_3 [20] parameter bit. • Bits [5:4] = Loopback mask data. Reports the actual data mask or the expected data mask, depending on the setting of the phy_ctrl_reg_1_3 [20] parameter bit. • Bit [1] = Reports status of loopback errors. -- 0: Last Loopback test had no errors. -- 1: Last Loopback test contained data errors. • Bit [0] = Defines the status of the loopback mode. -- 0: Not in loopback mode. -- 1: In Loopback mode. • All other bits Reserved.	

Register 4-89: DRAM control register 88 (CTRL_REG_88)

Field	Symbol	Direction	Description	Default
[31:0]	OUT_OF_RANGE_ADDR[31:0]	R	Holds the address of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.	0

Register 4-90: DRAM control register 89 (CTRL_REG_89)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1:0]	OUT_OF_RANGE_ADDR[33:32]	R	Holds the address of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.	0

Register 4-91: DRAM control register 90 (CTRL_REG_90)

Field	Symbol	Direction	Description	Default
[31:0]	PORT_CMD_ERROR_ADDR[31:0]	R	Holds the address of the command that caused a port command error condition. This parameter is read-only.	0

Register 4-92: DRAM control register 91 (CTRL_REG_91)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1:0]	PORT_CMD_ERROR_ADDR[33:32]	R	Holds the address of the command that caused a port command error condition. This parameter is read-only.	0

5 SRAM Controller

5.1 SRAM Controller Overview

The IMAPx210 SRAM Controller (SRAMC) supports external 8, 16-bit NOR Flash, PROM, SRAM memory. SRAM Controller supports 4-bank memory of maximum 64 MB size only.

SRAM Controller features include:

- Supports SRAM, various ROMs and NOR flash memory
- Supports only 8 or 16-bit data bus
- Address space: Up to 64 MB per Bank
- Supports 4 banks.
- Fixed memory bank start address
- External wait to extend the bus cycle
- Support byte and half-word access for external memory

5.2 SRAM Controller Function Descriptions

5.2.1 Architecture

Figure 5-1 illustrates block diagram of SRAM Controller.

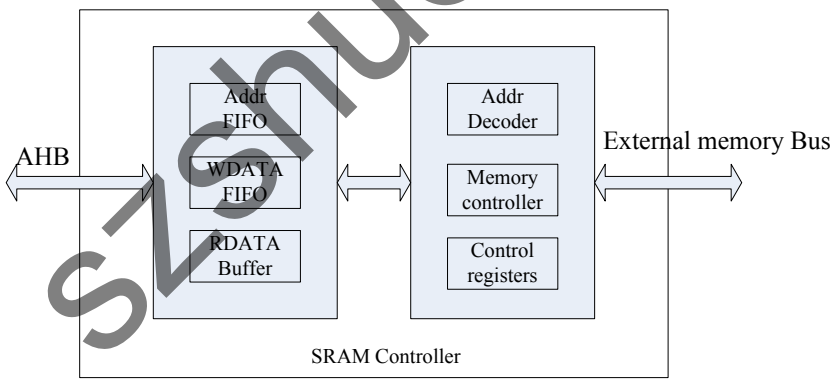


Figure 5-1: SRAM Controller Block Diagram

5.2.2 Interfacing with Asynchronous SRAMs

You can specify SRAM timing parameters by programming the Static Memory Timing Register - Set 0 (SMTR0), Static Memory Timing Register - Set 1 (SMTR1), or Static Memory Timing Register - Set 2 (SMTR2), depending on which one of the three register sets should control the SRAM.

You can use the following SRAM timing parameters:

- Read cycle time (tRC) – Bits 5:0 of SMTR0/1/2 specify the read cycle time;
- Address setup time (tAS) – Bits 7:6 of the Static Memory Timing Register SMTR0/1/2;
- Write pulse width (tWP) – Bits 15:10 of SMTR0/1/2;
- Write recovery time (tWR) – Bits 9:8 of the Static Memory Timing Register SMTR0/1/2;

- Bus turnaround time (tBTA) – Bits 18:16 of SMTR0/1/2;

This parameter forces the IMAPx210 SRAM Controller to insert tBTA number of cycles between back-to-back read/writes. If the address setup time is significant, tBTA is not necessary.

IMAPX210 Static memory controller supports asynchronous SRAMs, asynchronous NORFLASH memories.

The Static memory controller has three sets of timing registers for controlling the Static memory:

- Static Memory Timing Register 0 (SMTR0)
- Static Memory Timing Register 1 (SMTR1)
- Static Memory Timing Register 2 (SMTR2)

The address width of Static Memory controller is 26 bit.

The Static memory controller supports up to four chip selects. The Address Mask Registers (ADRMRn) (addresses 0x54 to 0x60) and the Chip Select Registers (CSRn) (addresses 0x14 to 0x20) control the chip select selection.

Bits 4:0 of the mask register specify the size of the memory assigned to a particular chip select. The IMAPX210 Static memory controller supports Static memory sizes from 64KB to 64MB. Bits 6:5 specify the type of memory connected to that particular chip select. Bits 8:7 specify the Static Memory Timing Register set with which this particular memory is associated. For more details, refer to the register descriptions.

The chip select base address registers hold the base address values that correspond to each chip select.

Figure 5-2 illustrates the timing for a read operation of an SRAM , where

- tRC – read cycle time.

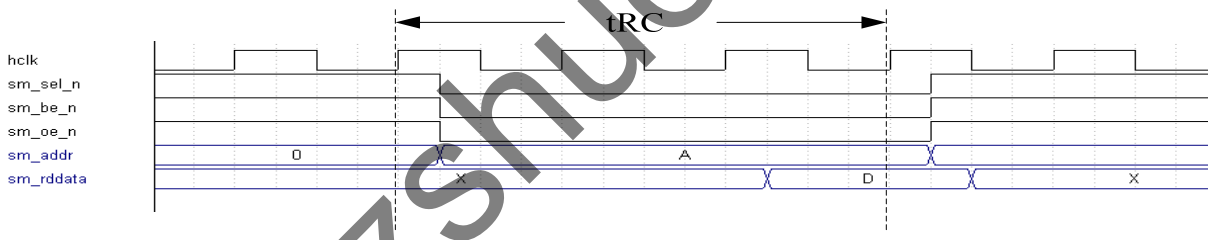


Figure 5-2: Read Timing of SRAM and Flash

Figure 5-3 illustrates the timing for a write operation of an SRAM, where:

- tAS – address setup time
- tWP – write pulse period
- tWR – write recovery time

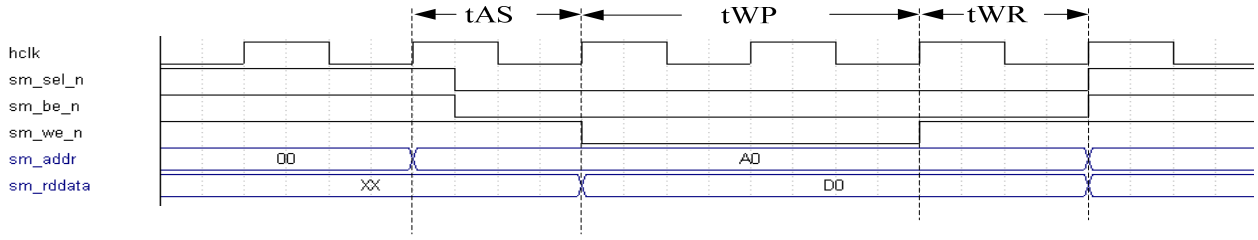


Figure 5-3: Write Timing of SRAM

Figure 5-4 illustrates an example of inserting one idle clock cycle for memory data bus turnaround time, where:

- tBTA – number of idle clock cycles for the external memory data bus turnaround time

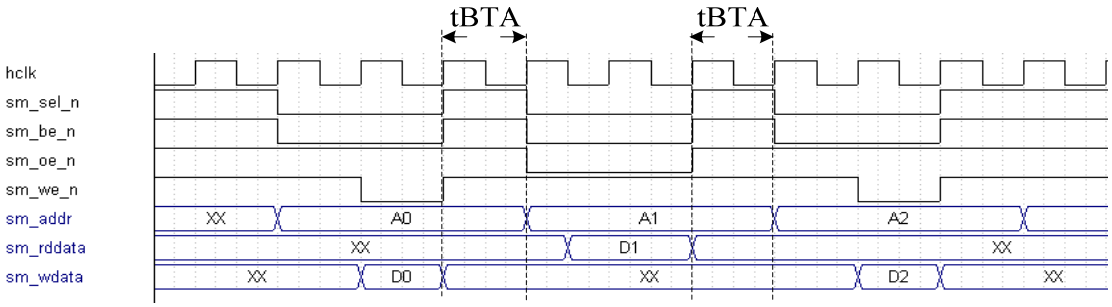


Figure 5-4: External Memory Data Bus Turnaround Timing

5.3 SRAM Controller Register Description

5.3.1 SRAM Controller Register Memory Map

Address	Symbol	Direction	Description
0x20C1_0054	CFGR0	R/W	Configuration register 0
0x20C1_0058	CFGR1	R/W	Configuration register 1
0x20C1_005C	CFGR2	R/W	Configuration register 2
0x20C1_0060	CFGR3	R/W	Configuration register 3
0x20C1_0094	SMTR0	R/W	Static memory timing register 0
0x20C1_0098	SMTR1	R/W	Static memory timing register 1
0x20C1_009C	SMTR2	R/W	Static memory timing register 2
0x20C1_00A4	SMCTLR	R/W	Static memory control register

5.3.2 SRAM Controller Individual Register Description

Register 5-1: Configuration Register0（CFGR0, offset=0x0054）

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0x0
[10:8]	set_select	R/W	Register determines which timing parameters of memory connect to chip select 0; primarily used for specifying static memories 0 – register set 0 1 – register set 1 2 – register set 2	0x0
[7:0]	Reserved	R/W	Must not modify the default value	0x1B

Register 5-2: Configuration Register1（CFGR1, offset=0x0058）

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0x0
[10:8]	set_select	R/W	Register determines which timing parameters of memory connect to chip select 1; primarily used for specifying static memories 0 – register set 0 1 – register set 1 2 – register set 2	0x0
[7:0]	Reserved	R/W	Must not modify the default value	0x1B

Register 5-3: Configuration Register2（CFGR2, offset=0x005C）

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0x0
[10:8]	set_select	R/W	Register determines which timing parameters of memory connect to chip select 2; primarily used for specifying static memories 0 – register set 0 1 – register set 1 2 – register set 2	0x0
[7:0]	Reserved	R/W	Must not modify the default value	0x1B

Register 5-4: Configuration Register3（CFGR3, offset=0x0060）

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0x0
[10:8]	set_select	R/W	Register determines which timing parameters of memory connect to chip select 3; primarily used for specifying static memories 0 – register set 0 1 – register set 1 2 – register set 2	0x0
[7:0]	Reserved	R/W	Must not modify the default value	0x1B

Register 5-5: Static Memory Timing Register0（SMTR0, offset=0x0094）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	R/W	must set 0	0x0
[26]	wait_mode	R/W	Indicates if the static memory associated with register set n is a data-ready device (valid data indicated by a wait signal)	0x0
[25:23]	Reserved	R/W	must set 0	0x0
[22:19]	Reserved	R/W	Reserved	0x8
[18:16]	t_bta	R/W	Static memory idle cycles between “read to write”, or “write to read”, or “read to read when chip-select changes” for memory data bus turn around time; values of 0-7 correspond to 0-7 idle clock cycles.	0x3
[15:10]	t_wp	R/W	Write pulse width; values of 0-63 correspond to write pulse width of 1-64 clock cycles	0x20
[9:8]	t_wr	R/W	Write address/data hold time; values of 0-3 correspond to write address/data hold time of 0-3 clock cycles	0x3
[7:6]	t_as	R/W	Write address setup time; values of 0-3 correspond to address setup time of 0-3 clock cycles;	0x3
[5:0]	t_rc	R/W	Read cycle time; values of 0-63 correspond to read cycle time of 1-64 clock cycles	0x20

Register 5-6: Static Memory Timing Register1（SMTR1, offset=0x0098）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	R/W	must set 0	0x0
[26]	wait_mode	R/W	Indicates if the static memory associated with register set n is a data-ready device (valid data indicated by a wait signal)	0x0
[25:23]	Reserved	R/W	must set 0	0x0
[22:19]	Reserved	R/W	Reserved	0x8
[18:16]	t_bta	R/W	Static memory idle cycles between “read to write”, or “write to read”, or “read to read when chip-select changes” for memory data bus turn around time; values of 0-7 correspond to 0-7 idle clock cycles.	0x3

[15:10]	t_wp	R/W	Write pulse width; values of 0-63 correspond to write pulse width of 1-64 clock cycles	0x20
[9:8]	t_wr	R/W	Write address/data hold time; values of 0-3 correspond to write address/data hold time of 0-3 clock cycles	0x3
[7:6]	t_as	R/W	Write address setup time; values of 0-3 correspond to address setup time of 0-3 clock cycles;	0x3
[5:0]	t_rc	R/W	Read cycle time; values of 0-63 correspond to read cycle time of 1-64 clock cycles	0x20

Register 5-7: Static Memory Timing Register2（SMTR2, offset=0x009C）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	R/W	must set 0	0x0
[26]	wait_mode	R/W	Indicates if the static memory associated with register set n is a data-ready device (valid data indicated by a wait signal)	0x0
[25:23]	Reserved	R/W	must set 0	0x0
[22:19]	Reserved	R/W	Reserved	0x8
[18:16]	t_bta	R/W	Static memory idle cycles between “read to write”, or “write to read”, or “read to read when chip-select changes” for memory data bus turn around time; values of 0-7 correspond to 0-7 idle clock cycles.	0x3
[15:10]	t_wp	R/W	Write pulse width; values of 0-63 correspond to write pulse width of 1-64 clock cycles	0x20
[9:8]	t_wr	R/W	Write address/data hold time; values of 0-3 correspond to write address/data hold time of 0-3 clock cycles	0x3
[7:6]	t_as	R/W	Write address setup time; values of 0-3 correspond to address setup time of 0-3 clock cycles;	0x3
[5:0]	t_rc	R/W	Read cycle time; values of 0-63 correspond to read cycle time of 1-64 clock cycles	0x20

Register 5-8: Static Memory Control Register0（SMTMGR0, offset=0x00A4）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	R/W	Reserved	0x0
[15:13]	sm_data_width2	R/W	Width of Static memory data bus controlled by Static memory register0: 000 - 16 bits 100 - 8 bits Others - Reserved	0x0
[12:10]	sm_data_width1	R/W	Width of Static memory data bus controlled by Static memory register1: 000 - 16 bits 100 - 8 bits Others - Reserved	0x0
[9:7]	sm_data_width0	R/W	Width of Static memory data bus controlled by Static memory register2: 000 - 16 bits 100 - 8 bits Others - Reserved	0x0
[6:0]	Reserved	R/W	Reserved	0x0

6 Interrupt Controller

6.1 Overview

The interrupt controller in the IMAPx210 receives the request from 56 interrupt sources and some interrupts are null. These interrupt sources are provided by internal peripherals such as DMA controller, UART, IIC, external interrupt and others.

When receiving multiple interrupt requests from internal peripherals and external interrupt request pins, the interrupt controller requests FIQ or IRQ interrupt of the CPU after the arbitration procedure.

The arbitration procedure depends on the hardware priority logic and the result is written to the interrupt pending register, which helps users notify which interrupt is generated out of various interrupt sources.

Features

- 56 Maskable Interrupt Sources
- Level/Edge mode on external interrupt source(configure in GPIO)
- Programmable priority of interrupt
- Supports Fast Interrupt request (FIQ) for very urgent interrupt request

Block Diagram

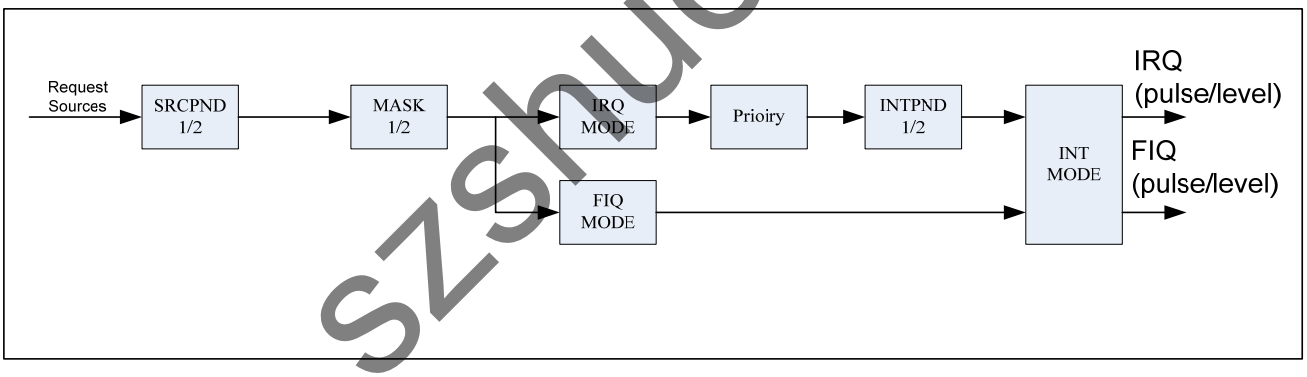


Figure 6-1: Interrupt Controller Block Diagram

6.2 Interrupt Controller Operation

F-bit and I-bit of Program Status Register (PSR)

If the F-bit of PSR in CPU is set to 1, the CPU does not accept the Fast Interrupt Request (FIQ) from the interrupt controller. Likewise, If I-bit of the PSR is set to 1, the CPU does not accept the Interrupt Request (IRQ) from the interrupt controller. So, the interrupt controller can receive interrupts by clearing F-bit or I-bit of the PSR to 0 and setting the corresponding bit of INTMSK1/2 to 0.

Interrupt Mode

The CPU has two types of Interrupt mode: FIQ or IRQ. All the interrupt sources determine which mode is used at interrupt request.

Interrupt Pending Register

The IMAPx210 has four interrupt pending registers: source pending register (SRCPND1 and SRCPND2) and interrupt pending register (INTPND1 and INTPND2). These pending registers indicate whether an interrupt request is pending or not. When the interrupt sources request interrupt the service, the corresponding bits of SRCPND1/SRCPND2 register are set to 1, and at the same time, only one bit of the INTPND1/INTPND2 register is set to 1 automatically after arbitration procedure. If interrupts are masked, then the corresponding bits of the SRCPND1/SRCPND2 register are set to 1. This does not cause the bit of INTPND1/INTPND2 register changed. When a pending bit of INTPND1/2 register is set, the interrupt service routine will start whenever the I-flag or F-flag is cleared to 0. The SRCPND1/2 and INTPND1/2 registers can be read and written, so the service routine can clear the pending condition by writing a 1 to the corresponding bit in the INTPND1/2 register and clear the pending condition in the SRCPND1/2 registers by using the same method.

Interrupt Mask Register

This register indicates that an interrupt has been disabled if the corresponding mask bit is set to 1. If an interrupt mask bit of INTMSK is 0, the interrupt will be serviced normally. If the corresponding mask bit is 1 and the interrupt is generated, the source pending bit will be set.

Interrupt Sources

The interrupt controller supports 56 interrupt sources as shown in the table below.

Table 6-1: CPU Interrupt Source

No	Description	Name	Position
INT0	External interrupt 0	EINT0	SRCPND[0]
INT1	External interrupt 1	EINT1	SRCPND[1]
INT2	External interrupt 2	EINT2	SRCPND[2]
INT3	External interrupt 3	EINT3	SRCPND[3]
INT4	External interrupt 4	EINT4	SRCPND[4]
INT5	External interrupt 5	EINT5	SRCPND[5]
INT6	AC97_Interrupt	INT_AC97	SRCPND[6]
INT7	IIS Interrupt	INT_IIS	SRCPND[7]
INT8	RTC Time tick interrupt	RTC_TICK	SRCPND[8]
INT9	WatchDog Interrupt	INT_WDT	SRCPND[9]
INT10	PWM-Timer0 interrupt	PWM_INT0	SRCPND[10]
INT11	PWM-Timer1 interrupt	PWM_INT1	SRCPND[11]
INT12	PWM-Timer2 interrupt	PWM_INT2	SRCPND[12]
INT13	UART2 Interrupt	INT_UART2	SRCPND[13]
INT14	IIC0 interrupt	INT_IIC0	SRCPND[14]
INT15	PWM-Timer3 interrupt	PWM_INT3	SRCPND[15]
INT16	IDS interrupt	INT_IDS	SRCPND[16]
INT17	Camera Interrupt	INT_CAM	SRCPND[17]
INT18	CNT-timer0_Interrupt	Timer_intr0	SRCPND[18]
INT19	CNT-timer1_Interrupt	Timer_intr1	SRCPND[19]
INT20	DMA Interrupt	INT_DMA	SRCPND[20]
INT21	Master SSI0 interrupt	INT_SSI_MST0	SRCPND[21]
INT22	Master SSI2 interrupt	INT_SSI MST2	SRCPND[22]
INT23	UART1/SPI Interrupt	INT_UART1	SRCPND[23]
INT24	Nand Flash Control Interrupt	INT_NFCON	SRCPND[24]
INT25	USBOTG_Interrupt	INT_USBOTG	SRCPND[25]
INT26	USBHost_OHCI_Interrupt	INT_OHCI	SRCPND[26]
INT27	NULL	NULL	SRCPND[27]
INT28	DRAM_ Interrupt	INT_DRAM	SRCPND[28]
INT29	Monitor_DBG Interrupt	INT_MONDBG	SRCPND[29]
INT30	power_mode_int	INT_PowerMode	SRCPND[30]

INT31	NULL	NULL	SRCPND[31]
INT32	Ethernet Interrupt	INT_ETH	SRCPND2[0]
INT33	UART3 Interrupt	INT_UART3	SRCPND2[1]
INT34	IIC1_ Interrupt	INT_IIC1	SRCPND2[2]
INT35	IDE/CF Interrupt	INT_IDE	SRCPND2[3]
INT36	SDIO 1 Interrupt	INT_SDIO 1	SRCPND2[4]
INT37	SDIO 2 Interrupt	INT_SDIO 2	SRCPND2[5]
INT38	PWM-Timer4 interrupt	PWM_INT4	SRCPND2[6]
INT39	NULL	NULL	SRCPND2[7]
INT40	Video ENC Interrupt	INT_VENC	SRCPND2[8]
INT41	Video DEC Interrupt	INT_VDEC	SRCPND2[9]
INT42	NULL	NULL	SRCPND2[10]
INT43	MemoryPool Master interrupt	INT_MPOOL	SRCPND2[11]
INT44	Usb host ehci interrupt	INT_EHCI	SRCPND2[12]
INT45	PS2-0 Interrupt	INT_PS2_0	SRCPND2[13]
INT46	Keyboard Interrupt	INT_KB	SRCPND2[14]
INT47	GPIO_ Interrupt	INT_GPIO	SRCPND2[15]
INT48	SDIO 0 interrupt	INT_SDIO0	SRCPND2[16]
INT49	NULL	NULL	SRCPND2[17]
INT50	PS2-1 Interrupt	INT_PS2_1	SRCPND2[18]
INT51	UART0 Interrupt	INT_UART0	SRCPND2[19]
INT52	Master SSI1/SlaveSSI Interrupt	INT_SSI_MST1	SRCPND2[20]
INT53	RTC alarm interrupt	INT_RTC	SRCPND2[21]
INT54	2D/3D Graphic Interrupt	INT_GPU	SRCPND2[22]
INT55	NULL	NULL	SRCPND2[23]

Interrupt Priority Generating Block

The priority logic for 56 interrupt requests is composed of 13 rotation based arbiters: ten first-level arbiters, two second-level arbiter and one third-level arbiter as shown in the following figures.

Interrupt Priority

Interrupt controller has three type arbiters, include 2, 4, 6 interrupt requests. There are supported both fix and rotate priority mode. But for six interrupt request arbiter, the priority of REQ0 and REQ5 can't be changed. Note that REQ0 always has the highest priority, and REQ5 has the second one, and the priority of REQ1 to REQ4 can be rotated.

Each arbiter can handle six interrupt requests based on the one bit arbiter mode control (ARB_MODE) and two bits of selection control signals (ARB_SEL) as follows:

— If ARB_SEL bits are 00b, the priority order is REQ0, REQ5, REQ1, REQ2, REQ3 and REQ4.
— If ARB_SEL bits are 01b, the priority order is REQ0, REQ5, REQ2, REQ3, REQ4 and REQ1.
— If ARB_SEL bits are 10b, the priority order is REQ0, REQ5, REQ3, REQ4, REQ1 and REQ2.
— If ARB_SEL bits are 11b, the priority order is REQ0, REQ5, REQ4, REQ1, REQ2 and REQ3.

Here, if ARB_MODE bit is set to 0, ARB_SEL bits doesn't change automatically changed, making the arbiter to operate in the fixed priority mode (note that even in this mode, we can reconfigure the priority by manually changing the ARB_SEL bits). On the other hand, if ARB_MODE bit is 1, ARB_SEL bits are changed in rotation fashion, e.g., if REQ1 is serviced, ARB_SEL bits are changed to 01b automatically so as to put REQ1 into the lowest priority. The detailed rules of ARB_SEL change are as follows:

— If REQ0 or REQ5 is serviced, ARB_SEL bits are not changed at all.
— If REQ1 is serviced, ARB_SEL bits are changed to 01b.
— If REQ2 is serviced, ARB_SEL bits are changed to 10b.
— If REQ3 is serviced, ARB_SEL bits are changed to 11b.
— If REQ4 is serviced, ARB_SEL bits are changed to 00b.

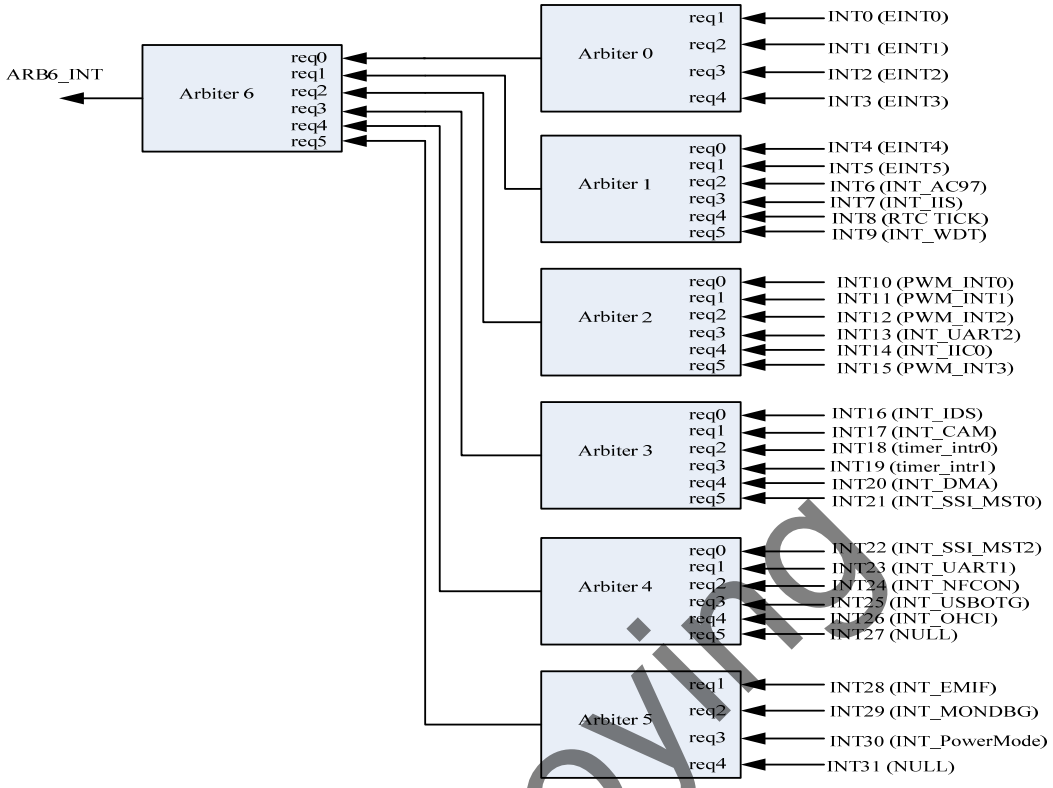


Figure 6-2: CPU Interrupt Priority Generating Block 1

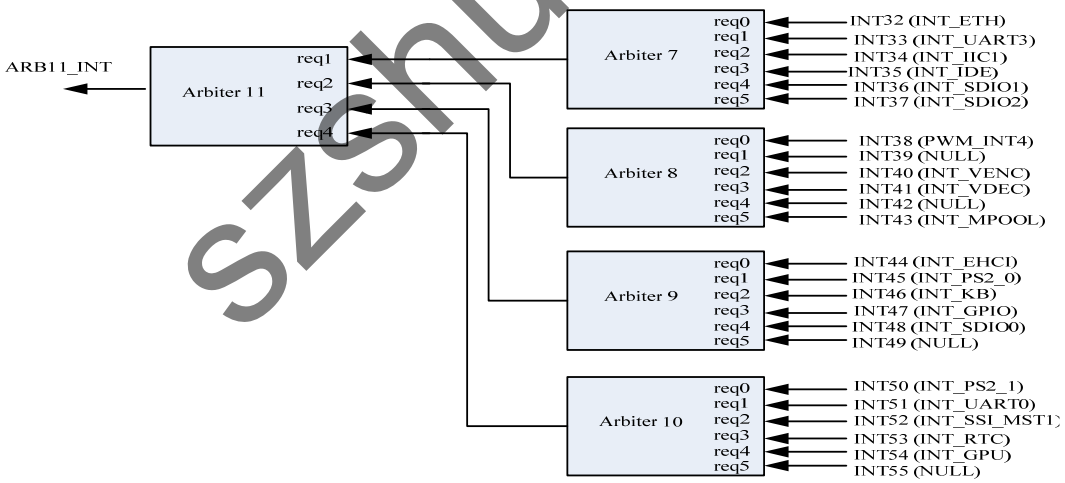


Figure 6-3: CPU Interrupt Priority Generating Block 2

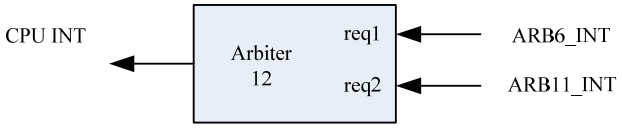


Figure 6-4: CPU Third-Level Interrupt Priority Generating Block

6.3 Interrupt Controller Register Description

6.3.1 Interrupt Controller Register Memory Map

Table 6-2: Interrupt Controller Register Memory Map

Address	Symbol	Direction	Description
0x20CB_0000	SRCPND1	R/W	Interrupt source pending register 1
0x20CB_0004	INTMOD1	R/W	Interrupt mode register 1
0x20CB_0008	INTMSK1	R/W	Interrupt mask register 1
0x20CB_000C	PRIORITY1	R/W	IRQ priority control register 1
0x20CB_0010	INTPND1	R/W	Interrupt pending register 1
0x20CB_0014	INTOFFSET	R	Interrupt offset register
0x20CB_0020	INTCON	R/W	Interrupt control register
0x20CB_0024	SRCPND2	R/W	Interrupt source pending register 2
0x20CB_0028	INTMOD2	R/W	Interrupt mode register 2
0x20CB_002C	INTMSK2	R/W	Interrupt mask register 2
0x20CB_0030	PRIORITY2	R/W	IRQ priority control register 2
0x20CB_0034	INTPND2	R/W	Interrupt pending register 2

6.3.2 Interrupt Controller Registers and Field Descriptions

Register 6-1: Interrupt Source Pending Register 1 （SRCPND1）

Field	Symbol	Direction	Description	Default
[31:0]	SRCPND1	R/W	INT0~INT31 interrupt source 0: no interrupt request 1: interrupt request	0x0

Register 6-2: Interrupt Source Pending Register 2 （SRCPND2）

Field	Symbol	Direction	Description	Default
[23:0]	SRCPND2	R/W	INT32~INT55 interrupt source 0: no interrupt request 1: interrupt request	0x0

Register 6-3: Interrupt Mode Register 1 （INTMOD1）

This register is composed of 32 bits each of which is related to an interrupt source (INT0-31). If a specific bit is set to 1, the corresponding interrupt is processed in the FIQ (fast interrupt) mode. Otherwise, it is processed in the IRQ mode (normal interrupt).

Please note that only one interrupt source can be serviced in the FIQ mode in the interrupt controller (you should use the FIQ mode only for the urgent interrupt). Thus, only one bit of INTMOD1/INTMOD2 can be set to 1.

Field	Symbol	Direction	Description	Default
[31:0]	INTMOD1	R/W	INT0~INT31 interrupt mode 0: IRQ mode 1: FIQ mode	0

Register 6-4: Interrupt Mode Register 2 （INTMOD2）

This register is composed of 24 bits each of which is related to an interrupt source (INT32-55). If a specific bit is set to 1, the corresponding interrupt is processed in the FIQ (fast interrupt) mode. Otherwise, it is processed in the IRQ mode (normal interrupt).

Please note that only one interrupt source can be serviced in the FIQ mode in the interrupt controller (you should use the FIQ mode only for the urgent interrupt). Thus, only one bit of INTMOD1/INTMOD2 can be set to 1.

Field	Symbol	Direction	Description	Default
[23:0]	INTMOD2	R/W	INT32~INT55 interrupt mode 0: IRQ mode 1: FIQ mode	0

Register 6-5: Interrupt Mask Register 1 （INTMSK1）

Field	Symbol	Direction	Description	Default
[31:0]	INTMSK1	R/W	INT0~INT31 interrupt mask register, The masked interrupt source will not be serviced. 0: Interrupt service is available 1: Interrupt service is masked	0xFFF FFFFF

Register 6-6: Interrupt Mask Register 2 （INTMSK2）

Field	Symbol	Direction	Description	Default
[23:0]	INTMSK2	R/W	INT32~INT55 interrupt mask register, The masked interrupt source will not be serviced. 0: Interrupt service is available 1: Interrupt service is masked	0xFFF FFF

Register 6-7: Interrupt Priority Control Register 1（PRIORITY1）

Field	Symbol	Direction	Description	Default
[20:19]	Arb_sel6	R/W	Arbiter 6 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[18:17]	Arb_sel5	R/W	Arbiter 5 group priority order 00: 1-2-3-4 01: 2-3-4-1 10: 3-4-1-2 11: 4-1-2-3	00
[16:15]	Arb_sel4	R/W	Arbiter 4 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[14:13]	Arb_sel3	R/W	Arbiter 3 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[12:11]	Arb_sel2	R/W	Arbiter 2 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[10:9]	Arb_sel1	R/W	Arbiter 1 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[8:7]	Arb_sel0	R/W	Arbiter 0 group priority order 00: 1-2-3-4 01: 2-3-4-1 10: 3-4-1-2 11: 4-1-2-3	00
[6]	Arb_mode6	R/W	Arbiter 6 group priority rotate enable	1

			0: disable 1: enable	
[5]	Arb_mode5	R/W	Arbiter 5 group priority rotate enable 0: disable 1: enable	1
[4]	Arb_mode4	R/W	Arbiter 4 group priority rotate enable 0: disable 1: enable	1
[3]	Arb_mode3	R/W	Arbiter 3 group priority rotate enable 0: disable 1: enable	1
[2]	Arb_mode2	R/W	Arbiter 2 group priority rotate enable 0: disable 1: enable	1
[1]	Arb_mode1	R/W	Arbiter 1 group priority rotate enable 0: disable 1: enable	1
[0]	Arb_mode0	R/W	Arbiter 0 group priority rotate enable 0: disable 1: enable	1

Register 6-8: Interrupt Priority Control Register 2（PRIORITY2）

Field	Symbol	Direction	Description	Default
[17:16]	Arb_sel12	R/W	Arbiter 12 group priority order 00: 1-2 01: 2-1	00
[15:14]	Arb_sel11	R/W	Arbiter 11 group priority order 00: 1-2-3-4 01: 2-3-4-1 10: 3-4-1-2 11: 4-1-2-3	00
[13:12]	Arb_sel10	R/W	Arbiter 10 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[11:10]	Arb_sel9	R/W	Arbiter 9 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[9:8]	Arb_sel8	R/W	Arbiter 8 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[7:6]	Arb_sel7	R/W	Arbiter 7 group priority order 00: 0-5-1-2-3-4 01: 0-5-2-3-4-1 10: 0-5-3-4-1-2 11: 0-5-4-1-2-3	00
[5]	Arb_mode12	R/W	Arbiter 12 group priority rotate enable 0: disable 1: enable	1
[4]	Arb_mode11	R/W	Arbiter 11 group priority rotate enable 0: disable 1: enable	1
[3]	Arb_mode10	R/W	Arbiter 10 group priority rotate enable 0: disable 1: enable	1
[2]	Arb_mode9	R/W	Arbiter 9 group priority rotate enable 0: disable 1: enable	1
[1]	Arb_mode8	R/W	Arbiter 8 group priority rotate enable 0: disable 1: enable	1
[0]	Arb_mode7	R/W	Arbiter 7 group priority rotate enable 0: disable 1: enable	1

Register 6-9: Interrupt Pending Register 1 （INTPND1）

Each of the 32 bits in the interrupt pending register shows whether the corresponding interrupt request, which is unmasked and waits for the interrupt to be serviced, has the highest priority . Since the INTPND1 register is located after the priority logic, only one bit can be set to 1, and that interrupt request generates IRQ to CPU. In interrupt service routine for IRQ, you can read this register to determine which interrupt source is serviced among the INT0-31 interrupt sources.

Field	Symbol	Direction	Description	Default
[31:0]	INTPND1	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0

Note: If the FIQ mode interrupt occurs, the corresponding bit of INTPND1 will not be turned on as the INTPND1 register is available only for IRQ mode interrupt.

Register 6-10: Interrupt Pending Register 2 （INTPND2）

Each of the 24 bits in the interrupt pending register shows whether the corresponding interrupt request, which is unmasked and waits for the interrupt to be serviced, has the highest priority . Since the INTPND2 register is located after the priority logic, only one bit can be set to 1, and that interrupt request generates IRQ to CPU. In interrupt service routine for IRQ, you can read this register to determine which interrupt source is serviced among the INT32-55 interrupt sources.

Field	Symbol	Direction	Description	Default
[23:0]	INTPND2	R/W	Indicate the interrupt request status. 0 = The interrupt has not been requested. 1 = The interrupt source has asserted the interrupt request.	0

Note: If the FIQ mode interrupt occurs, the corresponding bit of INTPND2 will not be turned on as the INTPND2 register is available only for IRQ mode interrupt.

Register 6-11: Interrupt Offset Register （INTOFFSET）

Field	Symbol	Direction	Description	Default
[5:0]	OFFSET	R	Indicate the IRQ interrupt request source The value in the interrupt offset register shows which interrupt request of IRQ mode is in the INTPND1/2 register. This bit can be cleared automatically by clearing SRCPND1/2 and INTPND1/2.	0

Register 6-12: Interrupt Control Register （INTCON）

Field	Symbol	Direction	Description	Default
[1]	Reserved	R/W	Reserved **Note:** This field must not be changed and always set to 1.	1
[0]	DBG_MODE	R/W	Interrupt controller test mode 0: normal mode, 1: test mode **Note:** This field must not be changed and always set to 0.	0

7 NANDFlash Controller

7.1 Overview

IMAPX210 boot code can be executed on an external 8-/16-bit NAND flash memory. In order to support NAND flash boot loader, the IMAPX210 is equipped with an internal SRAM buffer called 'IBRAM'. When booting, the first 8 KBytes of the NAND flash memory will be loaded into IBRAM and the boot code loaded into IBRAM will be executed.

Generally, the boot code will copy NAND flash content to DDR2/MDDR. Using hardware ECC, the NAND flash data validity will be checked. Upon the completion of the copy, the main program will be executed on the DDR2/MDDR.

Features

- Auto boot: The boot code is transferred into 8-kbytes IBRAM during reset. After the transfer, the boot code will be executed on the IBRAM.
- NAND Flash Page Type: 256Words, 512Bytes, 1KWords, 2KBytes, 2KWords and 4Kbytes page.
- Interface: 8/16-bit NAND flash memory interface bus.
- Software mode: User can directly access NAND flash memory, for example this feature can be used in read/erase/program NAND flash memory.
- Hardware ECC generation, detection and indication (Software correction).
- Support both SLC and MLC NAND flash memory : 1-bit ECC for SLC and 4 burst length (total 36-bit@512Bytes) ECC for MLC NAND flash.
- SFR I/F: Byte/half word/word access to Data register, and Word access to other registers
- The IBRAM 8-KB internal SRAM buffer can be used for another purpose after NAND flash booting.
- Built-in DMA master controller, support two DMA modes: SDMA and ADMA.
- Support ADMA descriptor table mechanism.

Block Diagram

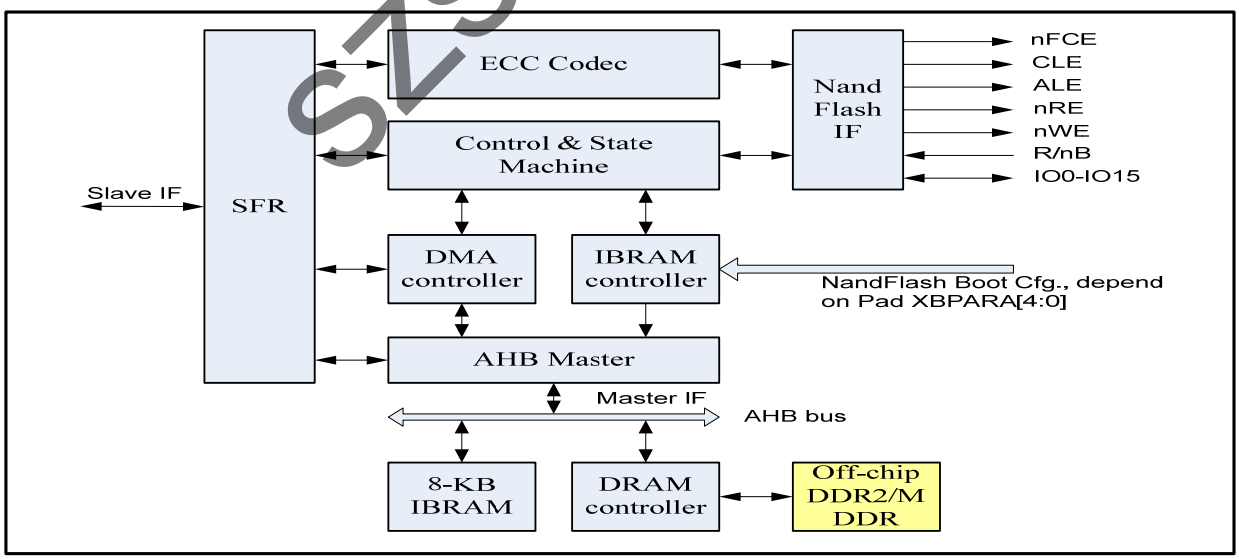


Figure 7-1: NFC Block Diagram

7.2 Functional Description

Boot Loader

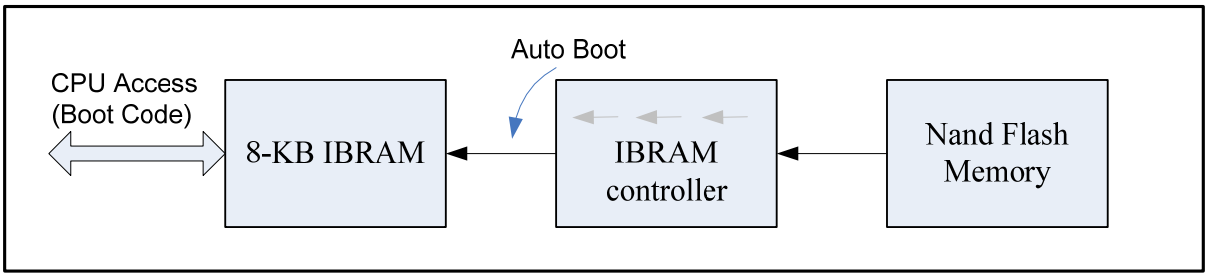


Figure 7-2: NandFlash Bootloader Block Diagram

During reset, Nand flash controller will get information about connected NAND flash through Pin status of XBPARA –refer to Nand Boot Pin Configuration Table below), After power-on or system reset is occurred, the NAND Flash controller load automatically the 8-KBytes boot loader codes. After loading the boot loader codes, the boot loader code in IBRAM is executed. The IBRAM can be used for another purpose after NAND flash booting.

Note: During the auto boot, the ECC is not checked. So, the first 8-KB of NAND flash should have no bit error.

Table 7-1: Nand Boot Pin Configuration Table

XBPARA[1:0] (Nand Type)	XBPARA[2] (Nand Page)	XBPARA[3] (Nand Address Cycle)	XBPARA[4] (Nand Width)
00 : normal nand	0 : 256 words 1: 512 bytes	0: 3 address cycle 1: 4 address cycle	0: 8-bit width 1:16-bit width
01 : advanced nand	0: 1K words 1: 2K bytes	0: 4 address cycle 1: 5 address cycle	
10 : large nand	0: 2K words 1: 4K bytes	0: 4 address cycle 1: 5 address cycle	

Nand Flash Memory Timing

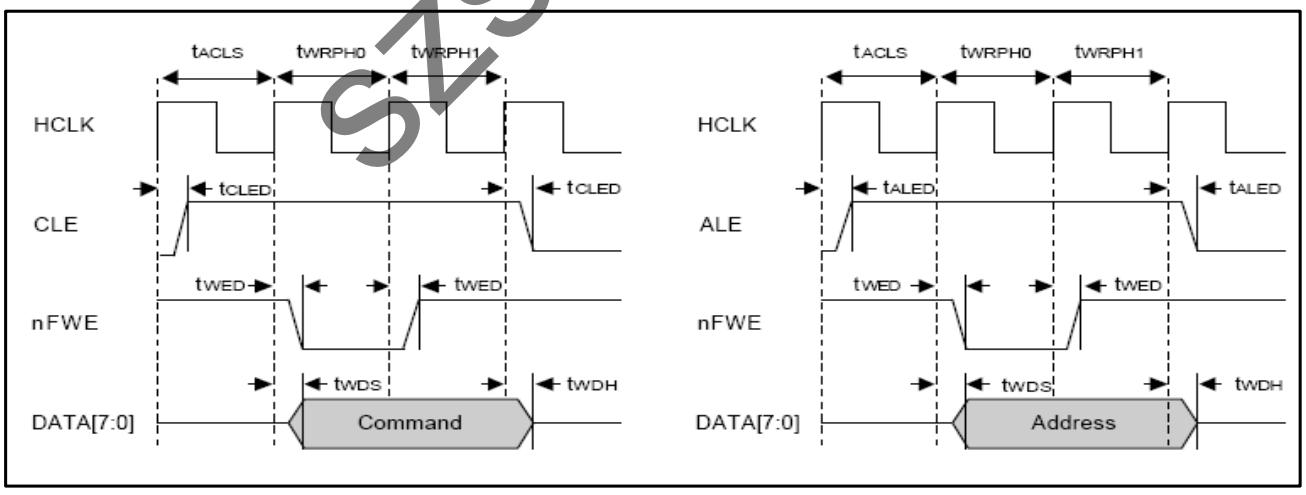


Figure 7-3: NAND Flash Address/Command Timing Diagram (TACLS=1, TWRPH0=0, TWRPH1=0)

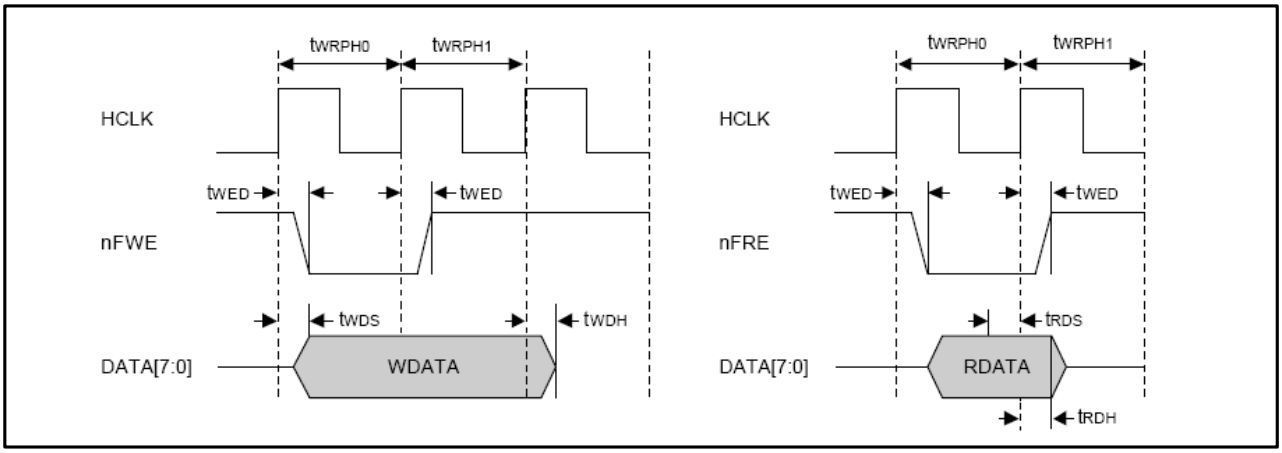


Figure 7-4: NAND Flash Data Timing Diagram (TWRPH0=0, TWRPH1=0)

Note: Timing parameter TACLS, TWRPH0 and TWRPH1 can be programmed by software.

Nand Flash Access

IMAPX210 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)

Note: In NAND flash access, you must check the RnB status input pin by polling the signal or using interrupt. For ADMA based transfer, ADMA controller will automatically check RnB status.

Data Register Configuration

1) 16-bit width Nand Flash

A. Word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	2nd IO[15:8]	2nd IO[7:0]	1st IO[15:8]	1st IO[7:0]

B. Half-word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	--	--	1st IO[15:8]	1st IO[7:0]

2) 8-bit width Nand Flash

A. Word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	4th IO[7:0]	3rd IO[7:0]	2nd IO[7:0]	1st IO[7:0]

B. Half-word Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	--	--	2nd IO[7:0]	1st IO[7:0]

C. Byte Access

Register	Bit [31:24]	Bit [23:16]	Bit [15:8]	Bit [7:0]
NFDATA	--	--	--	1st IO[7:0]

SLC / MLC ECC (Error Correction Code)

NAND flash controller has four ECC (Error Correction Code) modules for SLC NAND flash memory. And has one ECC module for MLC NAND flash memory.

For SLC NAND flash memory interface, NAND Flash controller has four ECC (Error Correction Code) modules. The two ECC modules (one for data[7:0] and the other for data[15:8]) can be used for (up to) 2048 bytes ECC Parity code generation, and the others(one for data[7:0] and the other for data[15:8]) can be used for (up to) 16 bytes ECC Parity code generation.

For MLC NAND flash memory interface, NAND flash controller has one ECC module. This can be used only 512 bytes ECC parity code generation. For 8-bit memory interface, MLC ECC module generate parity code for each 512 byte. And for 16-bit memory interface, MLC ECC module generate parity code for each 256 words (512 bytes). But SLC ECC modules generate parity code per byte lane separately.

MLC ECC module will reorganize 512bytes/256 words to 456 units and each unit has 9 bits except for the last unit. MLC ECC module can correct max 4 units/36bits data error.

ECC generation is controlled by the ECC Lock (MECCLOCK, SECCLOCK) bit of the NFCONT register. When ECCLOCK is Low, ECC codes are generated by the H/W ECC modules

Following ECC parity code and two tables are SLC ECC.

- 28-bit ECC Parity Code = 22-bit Line parity + 6bit Column Parity
- 14-bit ECC Parity Code = 8-bit Line parity + 6bit Column Parity

Table 7-2: SLC 2048 Byte ECC Parity Code Assignment Table

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
MECCDatan_3	P64	P64’	P32	P32’	P16	P16’	P8	P8’
MECCDatan_2	P1024	P1024’	P512	P512’	P256	P256’	P128	P128’
MECCDatan_1	P4	P4’	P2	P2’	P1	P1’	P2048	P2048’
MECCDatan_0	P8192	P8192’	P4096	P4096’	–	–	–	–

Table 7-3: SLC 16 Byte ECC Parity Code Assignment Table

	Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1]	Bit[0]
SECCDatan_1	P16	P16’	P8	P8’	P4	P4’	P2	P2’
SECCDatan_0	P1	P1’	P64	P64’	P32	P32’	–	–

Note: Where n in the above two tables represent byte lane of NAND Flash IO, range from 0 to 1.

SLC ECC Programming Guide

1. To use SLC ECC in software mode, reset the ECCTYPE bit in NFCONF register to ‘0’ (enable SLC ECC). SLC ECC module generates ECC parity code for all read / write data when MECCLOCK (NFCONT[7]) and SECCLOCK (NFCONT[6]) are unlocked (‘0’). So you have to reset ECC value by writing the INITMECC (NFCONT[5]) and INITSECC (NFCON[4]) bit as ‘1’ and have to clear the MECCLOCK (NFCONT[7]) bit to ‘0’(Unlock) before read or write data.

   MECCLOCK (NFCONT[7]) and SECCLOCK (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 (not include spare area data), Set the MECCLOCK bit to ‘1’(Lock). ECC Parity code is locked and the value of the ECC status register will not be changed. For large NandFlash, such as the byte number

of one page is 4096 bytes, you need separate one page into two sub-pages and generate ECC for each sub-page independently.

4. To generate spare area ECC parity code, Clear as '0'(Unlock) SECCLOCK(NFCONT[6]) bit.
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 SECCLOCK bit to '1'(Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.
7. From now, 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 and NFECCERR1 have the valid error status values.

MLC ECC Programming Guide (Encoding)

1. To use MLC ECC in software mode, set the ECCTYPE to '1'(enable MLC ECC). ECC module generates ECC parity code for all write data. So you have to reset ECC value by writing the INITMECC (NFCONT[5]) and INITSECC (NFCONT[4]) bit as '1' and have to clear the MECCLock (NFCONT[7]) bit to '0'(Unlock) before write data.

   MECCLOCK (NFCONT[7]) and SECCLOCK (NFCONT[6]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is written, the MLC ECC module generates ECC parity code internally.
3. After you complete write 512-byte or 256-words (16-bit I/O) (not include spare area data), Set the MECCLock bit to '1'(Lock). ECC Parity code generation is locked and the values are updated to NFMECC0, NFMECC1 and NFMECC2 register when NFSTAT[7] (ECCENCDONE) is set('1'). If you use 512-byte or 256-word (16-bit I/O) NAND flash memory, you can program these values to spare area. But if you use NAND flash memory more than 512-byte or 256-word (16-bit I/O) page, you can't program right now. 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.
4. To generate spare area ECC parity code, you can use SLC ECC module to generate spare area ECC parity code and set the ECCTYPE to '0'(enable SLC ECC).
5. Whenever spare data is written, the SLC ECC module generates ECC parity code on register NFSECC (main area parity code byte number small than 16 bytes) or NFMECC0/1 (main area parity code byte number more than 16 bytes).
6. After you complete write spare area, set the SECCLOCK or MECCLOCK bit to '1'(Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.
7. From now, you can use these values to record to the spare area.

MLC ECC Programming Guide (Decoding)

1. To use MLC ECC in software mode, set the ECCTYPE to '1' (enable MLC ECC). ECC module generates ECC parity code for all read data. So you have to reset ECC value by writing the INITMECC (NFCONT[5]) and INITSECC (NFCONT[4]) bit as '1' and have to clear the MECCLOCK (NFCONT[7]) bit to '0' (Unlock) before read data.

   MECCLOCK (NFCONT[7]) and SECCLOCK(NFCONT[6]) bit controls whether ECC Parity code is generated or not.
2. Whenever data is read, the MLC ECC module generates ECC parity code internally.
3. Before reading main data, you should read MLC ECC parity code in spare area first and write to NFMECCD0, NFMECCD1 and NFMECCD2 register.
4. After you complete read 512-byte or 256-words (16-bit I/O) (not include spare area data), Set the MECCLOCK bit to '1'(Lock). ECC Parity code generation is locked. MLC ECC engine start to search any error internally and NFSTAT[6] can be used to check whether ECC decoding is completed or not.
4. When ECCDECDONE (NFSTAT[7]) is set ('1'), NFECCERR0 indicates whether error bit exist or not. If any error exist, you can fix it by referencing NFECCERR0, NFECCERR1 and NFECCERR2 register.
5. If you have more main data to read, continue to step 1).

6. MLC ECC parity code scheme is used only for main data area to find up to 4-burst length error.
7. Whenever spare data is read, the SLC ECC module generates ECC parity code on register NFSECC (main area parity code byte number small than 16 bytes) or NFMECC0/1 (main area parity code byte number more than 16 bytes).
8. After you complete read spare area, set the SECCLOCK or MECCLOCK bit to '1'(Lock). ECC Parity code is locked and the value of the ECC status register will not be changed.

SDMA Transaction

SDMA transaction is usually used for main data reading/writing, also can be used for spare data reading/writing. The main purpose is to improve data transfer efficient and decrease CPU's load. The figure below shows one page program with using SDMA.

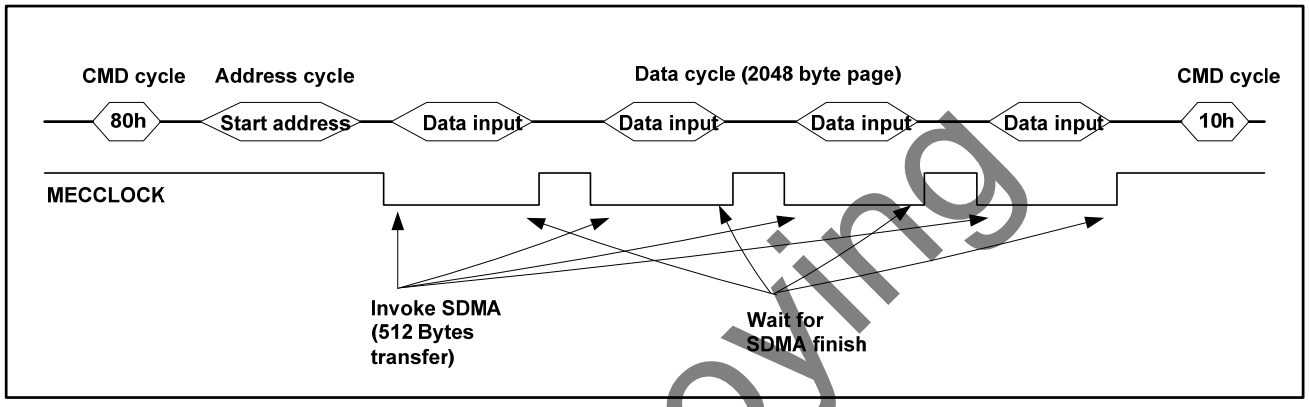


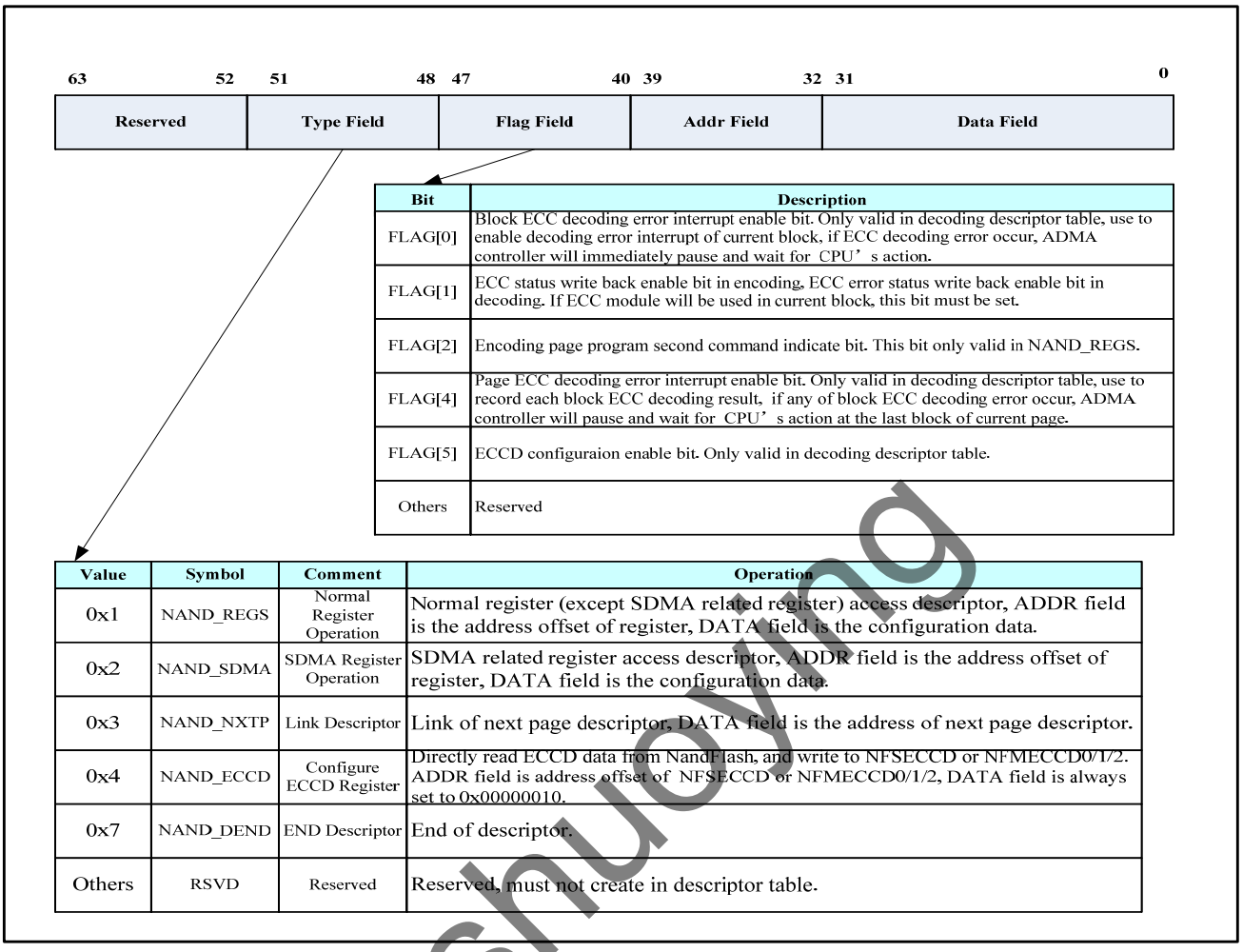
Figure 7-5: Page Program with Using SDMA

ADMA Transaction

ADMA provides data transfer between system memory and NandFlash without interruption of CPU execution. Software can program a list of data transfers between system memory and NandFlash to the Descriptor Table before executing ADMA. ADMA supports one and multi- page descriptor table.

ADMA Descriptor Table

ADMA controller can only access partial registers and fields. There are NFCONF (only ECCTYPE can access, write to other fields will be ignored), NFCONT (only MECCLOCK, SECCLOCK, INITMECC and INITSECC can access, write to other fields will be ignored), NFMECCD0/1/2, NFSECCD, NFCMD, NFADDR, NFDATA and SDMA related registers. The general definition of descriptor line is shown in the following figure, each line consumes 64-bit (8-byte) memory space.



Bit	Description
FLAG[0]	Block ECC decoding error interrupt enable bit. Only valid in decoding descriptor table, use to enable decoding error interrupt of current block, if ECC decoding error occur, ADMA controller will immediately pause and wait for CPU's action.
FLAG[1]	ECC status write back enable bit in encoding, ECC error status write back enable bit in decoding. If ECC module will be used in current block, this bit must be set.
FLAG[2]	Encoding page program second command indicate bit. This bit only valid in NAND_REGS.
FLAG[4]	Page ECC decoding error interrupt enable bit. Only valid in decoding descriptor table, use to record each block ECC decoding result, if any of block ECC decoding error occur, ADMA controller will pause and wait for CPU's action at the last block of current page.
FLAG[5]	ECCD configuraion enable bit. Only valid in decoding descriptor table.
Others	Reserved

Value	Symbol	Comment	Operation
0x1	NAND_REGS	Normal Register Operation	Normal register (except SDMA related register) access descriptor, ADDR field is the address offset of register, DATA field is the configuration data.
0x2	NAND_SDMA	SDMA Register Operation	SDMA related register access descriptor, ADDR field is the address offset of register, DATA field is the configuration data.
0x3	NAND_NXTP	Link Descriptor	Link of next page descriptor, DATA field is the address of next page descriptor.
0x4	NAND_ECCD	Configure ECCD Register	Directly read ECCD data from NandFlash, and write to NFSECCD or NFMECCD0/1/2. ADDR field is address offset of NFSECCD or NFMECCD0/1/2, DATA field is always set to 0x00000010.
0x7	NAND_DEND	END Descriptor	End of descriptor.
Others	RSVD	Reserved	Reserved, must not create in descriptor table.

Figure 7-6: General Definition of Descriptor Line

Five descriptor tables are supported, encoding only uses four types and decoding uses all five types. The following section will give and example for encoding and decoding respectively.

Encoding Descriptor Table

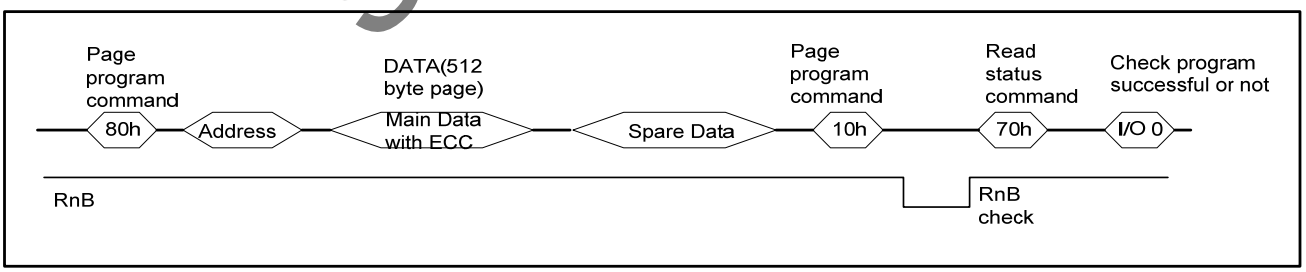


Figure 7-7: 512-byte Page NandFlash Program Diagram

Figure 7-7 shows the page program process for 512-byte page NandFlash. The whole process can be created in one ADMA descriptor table. Once enable ADMA, ADMA controller will execute the page program and interrupt CPU when complete.

Table 7-4: Example of Encoding Descriptor Table

Line No	Reserved(bit[63:52)	Type(bit[51:48)	Flag(bit[47:40)	Addr(bit[39:32)	Data(bit[31:0)
0	0x0	0x1	0x00	0x00	0x01000000
1	0x0	0x1	0x00	0x08	0x00000080
2	0x0	0x1	0x00	0x0C	0x00000000
3	0x0	0x1	0x00	0x0C	0x00000000
4	0x0	0x1	0x00	0x0C	0x00000000
5	0x0	0x1	0x00	0x0C	0x00000000
6	0x0	0x1	0x00	0x04	0x000000F0
7	0x0	0x1	0x00	0x04	0x00000040
8	0x0	0x2	0x00	0x88	0x00000000
9	0x0	0x2	0x00	0x8C	0x00000000
10	0x0	0x2	0x00	0x80	0x40000000
11	0x0	0x2	0x00	0x84	0x80000200
12	0x0	0x2	0x00	0x80	0x41000008
13	0x0	0x2	0x00	0x84	0x8200000C
14	0x0	0x1	0x02	0x04	0x000000C0
15	0x0	0x2	0x00	0x88	0x00000000
16	0x0	0x2	0x00	0x8C	0x00000000
17	0x0	0x2	0x00	0x80	0x41000000
18	0x0	0x2	0x00	0x84	0x80000010
19	0x0	0x2	0x00	0x80	0x00000000
20	0x0	0x2	0x00	0x84	0x00000000
21	0x0	0x1	0x00	0x04	0x000000C0
22	0x0	0x1	0x04	0x08	0x00000010
23	0x0	0x1	0x00	0x08	0x00000070
24	0x0	0x1	0x00	0x10	0x000000FE
25	0x0	0x7	0x00	0x00	0x00000000

1) Line 0, **type =0x1** indicates normal register (except SDMA related register) access, address field is the address offset of each register, data field is the register data, i.e. write 0x01000000 to register NFCONT (set ECCTYPE to MLC).
2) Line 1, write 0x80 to NFCMD register, i.e. issue page program 1command to NandFlash.
3) Line 2 to 5, successively write 0x00 to NFADDR register, i.e. issue 4 cycle address access to NandFlash.
4) Line 6, write 0x000000F0 to register NFCONT, i.e. reset ECC module.
5) Line 7, write 0x00000040 to register NFCONT, i.e. Unlock MECCLOCK bit.
6) Line 8 to 11, **type =0x2** indicates SDMA related register access, i.e. configure SDMA to transfer 512 bytes data from system memory (0x40000000) to main area of NandFlash.

   Note: These four lines must be set together even though you only use one address mode for SDMA.
7) Line 12 to 13, configure SDMA related register, i.e. configure SDMA to transfer ECC status from register NFMECC0/1/2 to system memory (0x41000008).

   Note: These two lines must be set to zero if user doesn't use ECC module to generate ECC status for main data.
8) Line 14, write 0x000000C0 to register NFCONT, i.e. Lock MECCLOCK bit.

   Note: If ECC module is used, FLAG[1] of this descriptor line must be set to 1. When ADMA execute current line, it will invoke SDMA to transfer ECC status from register NFMECC0/1/2 to system memory address 0x41000008. The value of SDMALEN0 is used to decide which ECC status register will be transferred to system memory. The followed table shows the relation of SDMALEN0 and ECC status registers.

SDMALEN0	ECC status registers
0x4	Only NFMECC0 is read.
0x8	NFMECC0 and NFMECC1 are read, NFMECC0 read first.
0xc	NFMECC0, NFMECC1 and NFMECC2 are read, NFMECC0 read first.

9) Like line 6 to 14, line 15 to 21 are used to transfer spare data from system memory (0x41000000) to NandFlash spare area. In current descriptor table, spare data doesn't use ECC module, so FLAG[1] of line 21 is set to 0.

10) Line 22, write 0x10 to NFCMD register, i.e. issue page program second command to NandFlash. In this line, FLAG[2] must be set to 1, use to inform ADMA controller that after issuing current command, need to check RnB signal. When RnB is not busy, controller will execute next descriptor line.

11) Line 23, write 0x70 to NFCMD register, i.e. issue read status command to NandFlash.

12) Line 24, access NFDATA register, followed by read status command, it will be parsed to read status from NandFlash, used to check program successful or not. If page program unsuccessful, controller will issue interrupt and stop, wait for CPU's action.

Note: Data[7:0] is the mask value of read status, data[15:8] is the status reference value. In this line, the value of mask is 0xFE, reference value is 0x00, that is only check IO[0] of NandFlash data bus.

13) Line 25, **type =0x7** indicates current line is the last line of descriptor table. If user want to create multi-page descriptor table, this line can be set to **type=0x3** (indicates link line for next page descriptor table, DATA field is the address of next page descriptor table).

The descriptor table for encoding can be abstracted in the following figure:

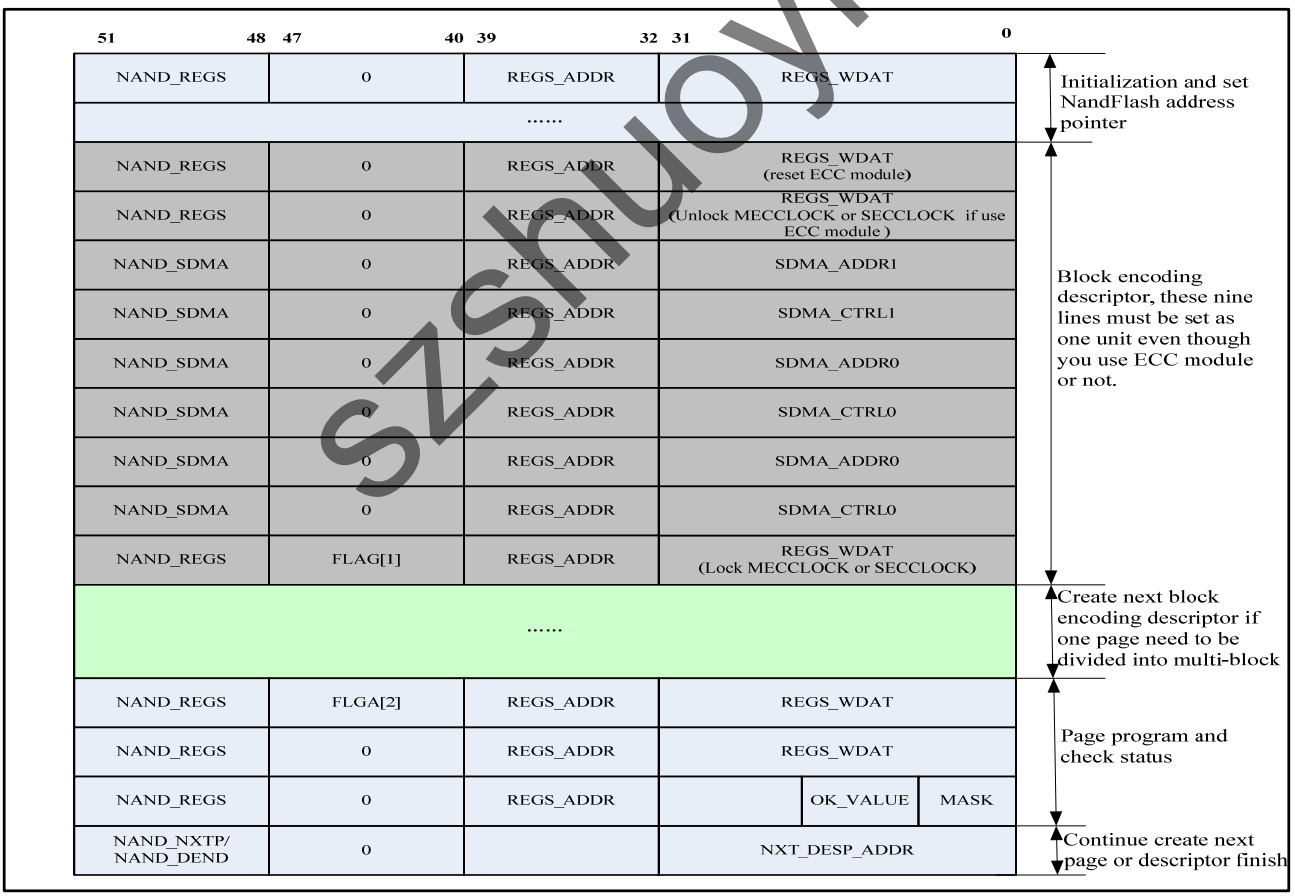


Figure 7-8: Encoding Descriptor Table Diagram

Decoding Descriptor Table

Table 7-5: Example of Decoding Descriptor Table

Line No	Reserved(bit[63:52)	Type(bit[51:48)	Flag(bit[47:40)	Addr(bit[39:32)	Data(bit[31:0)
0	0x0	0x1	0x00	0x00	0x00000000
1	0x0	0x1	0x00	0x08	0x00000050
2	0x0	0x1	0x00	0x0C	0x00000009
3	0x0	0x1	0x00	0x0C	0x00000000
4	0x0	0x1	0x00	0x0C	0x00000000
5	0x0	0x1	0x00	0x0C	0x00000000
6	0x0	0x4	0x00	0x44	0x00000010
7	0x0	0x1	0x00	0x08	0x00000050
8	0x0	0x1	0x00	0x0C	0x00000000
9	0x0	0x1	0x00	0x0C	0x00000000
10	0x0	0x1	0x00	0x0C	0x00000000
11	0x0	0x1	0x00	0x0C	0x00000000
12	0x0	0x1	0x00	0x04	0x000000F0
13	0x0	0x1	0x00	0x04	0x00000080
14	0x0	0x2	0x00	0x88	0x00000000
15	0x0	0x2	0x00	0x8C	0x00000000
16	0x0	0x2	0x00	0x80	0x41000000
17	0x0	0x2	0x00	0x84	0x82000008
18	0x0	0x2	0x11	0x80	0x41200000
19	0x0	0x2	0x00	0x84	0x82000004
20	0x0	0x1	0x02	0x04	0x000000C0
21	0x0	0x2	0x00	0x84	0x00000000
22	0x0	0x2	0x00	0x80	0x00000000
23	0x0	0x1	0x00	0x08	0x00000000
24	0x0	0x1	0x00	0x0C	0x00000000
25	0x0	0x1	0x00	0x0C	0x00000000
26	0x0	0x1	0x00	0x0C	0x00000000
27	0x0	0x1	0x00	0x0C	0x00000000
28	0x0	0x1	0x00	0x04	0x000000F0
29	0x0	0x1	0x00	0x04	0x00000040
30	0x0	0x2	0x00	0x88	0x00000000
31	0x0	0x2	0x00	0x8C	0x00000000
32	0x0	0x2	0x00	0x80	0x40000000
33	0x0	0x2	0x00	0x84	0x82000200
34	0x0	0x2	0x11	0x80	0x41200004
35	0x0	0x2	0x00	0x84	0x82000004
36	0x0	0x1	0x02	0x04	0x000000C0
37	0x0	0x2	0x20	0x84	0x41000000
38	0x0	0x2	0x00	0x80	0x80000004
39	0x0	0x7	0x0	0x0	0x0

1) Line 0, **type=0x1** indicates normal register (except SDMA related register) access, address field is the address offset of each register, data field is the register data, i.e. write 0x00000000 to register NFCONT (set ECCTYPE to SLC, spare data use SLC ECC module).
2) Line 2 to line 5, access NandFlash, set address pointer to 0x9 in spare array.
3) Line 6, **type=0x4** indicates read NFDATA register. Controller will write the data from NFDATA to NFSECCD, i.e. read SECC status from spare area of NandFlash to NFSECCD. Address field used for register NFSECCD address offset and data field used for NFDATA register address offset.

ADMA controller will automatically check RnB signal before executing this line.

4) Line 7 to line 11, access NandFlash, set address pointer to 0x0 in spare array.
5) Line 12, write 0x000000F0 to register NFCONT, i.e. reset ECC module.
6) Line 13, write 0x00000080 to register NFCONT, i.e. Unlock SECCLOCK bit.
7) Line 14 to 17, **type =0x2** indicates SDMA related register access, i.e. configure SDMA to transfer 8 bytes data from spare area of NandFlash to system memory (0x41000000).
8) Line 18 to 19, configure SDMA to transfer 4 bytes data from NFECCERR0 to system memory (0x41200000). FLAG[0] of line 18 is used for issuing ECC decoding error interrupt, controller will pause and wait for CPU's action. FLAG[4] is used for recording each block decoding error status, controller will issuing ECC decoding error interrupt and pause, wait for CPU's action at the last block of page.
9) Line 20, write 0x000000C0 to register NFCONT, i.e. Lock SECCLOCK bit.

Note: If ECC module is used, FLAG[1] of this descriptor line must be set to 1. When ADMA execute current line, it will invoke SDMA to transfer ECC error status from register NFECCERR0/1/2 to system memory address 0x41200000. The value of SDMALEN0 is used to decide which ECC error status register will be transferred to system memory. The followed table shows the relation of SDMALEN0 and ECC error status registers.

SDMALEN0	ECC status registers
0x4	Only NFECCERR0 is read.
0x8	NFECCERR0 and NFECCERR1 are read, NFECCERR0 read first.
0xc	NFECCERR0, NFECCERR1 and NFECCERR2 are read, NFECCERR0 read first.

10) Line 21 to 22, configure SDMA to transfer ECCD from system memory to MECCD0/1/2.

Note: Current block descriptor is used for reading MECCD data to NFMECCD0/1/2 register, for SECCD data, user can use NAND_SECC type. So in current line, you just need to set both FLAG and DATA field to 0.

11) Line 23 to 27, access NandFlash, set address pointer to 0x0 in main array.
12) Line 28, write 0x000000F0 to register NFCONT, i.e. reset ECC module.
13) Line 29, write 0x00000040 to register NFCONT, i.e. Unlock MECCLOCK bit.
14) Line 30 to 33, configure SDMA to transfer 512 bytes data from main area of NandFlash to system memory (0x40000000).
15) Line 34 to 35, configure SDMA to transfer 4 bytes data from NFECCERR0 to system memory (0x41200004). FLAG[0] of line 34 is used for issuing block ECC decoding error interrupt, controller will pause and wait for CPU's action. FLAG[4] is used for recording each block decoding error status, controller will issuing ECC decoding error interrupt and pause, wait for CPU's action at the last block of each page.
16) Line 36, write 0x000000C0 to register NFCONT (Lock MECCLOCK bit) and set FLAG[1] to invoke SDMA to transfer ECC error status from register NFECCERR0/1/2 to system memory address 0x41200004.
17) Line 37 to 38, configure SDMA to transfer ECCD from system memory to MECCD0/1/2.

Note: Current block descriptor is used for reading MECCD data to NFMECCD0/1/2 register. FLAG[5] of this descriptor line must be set to 1, it will invoke SDMA to transfer MECCD data from system memory (0x41000000, the ECCD data had been read from spare area of NandFlash to system memory by line 14 to 17) to register NFMECCD0/1/2. The value of SDMALEN0 is used to decide which MECCD register will be written. The followed table shows the relation of SDMALEN0 and MECCD registers.

SDMALEN0	MECCD registers
0x4	Only NFECCD0 is written.
0x8	NFECCD0 and NFECCD1 are written, NFECCD0 write first.
0xc	NFECCD0, NFECCD1 and NFECCD2 are written, NFECCD0 write first.

18) Line 39, **type =0x7** indicates current line is the last line of descriptor table. If user want to create multi-page descriptor table, this line can be set to **type=0x3** (indicates link line for next page descriptor table, DATA field is the address of next page descriptor table).

The descriptor table for decoding can be abstracted in the following figure:

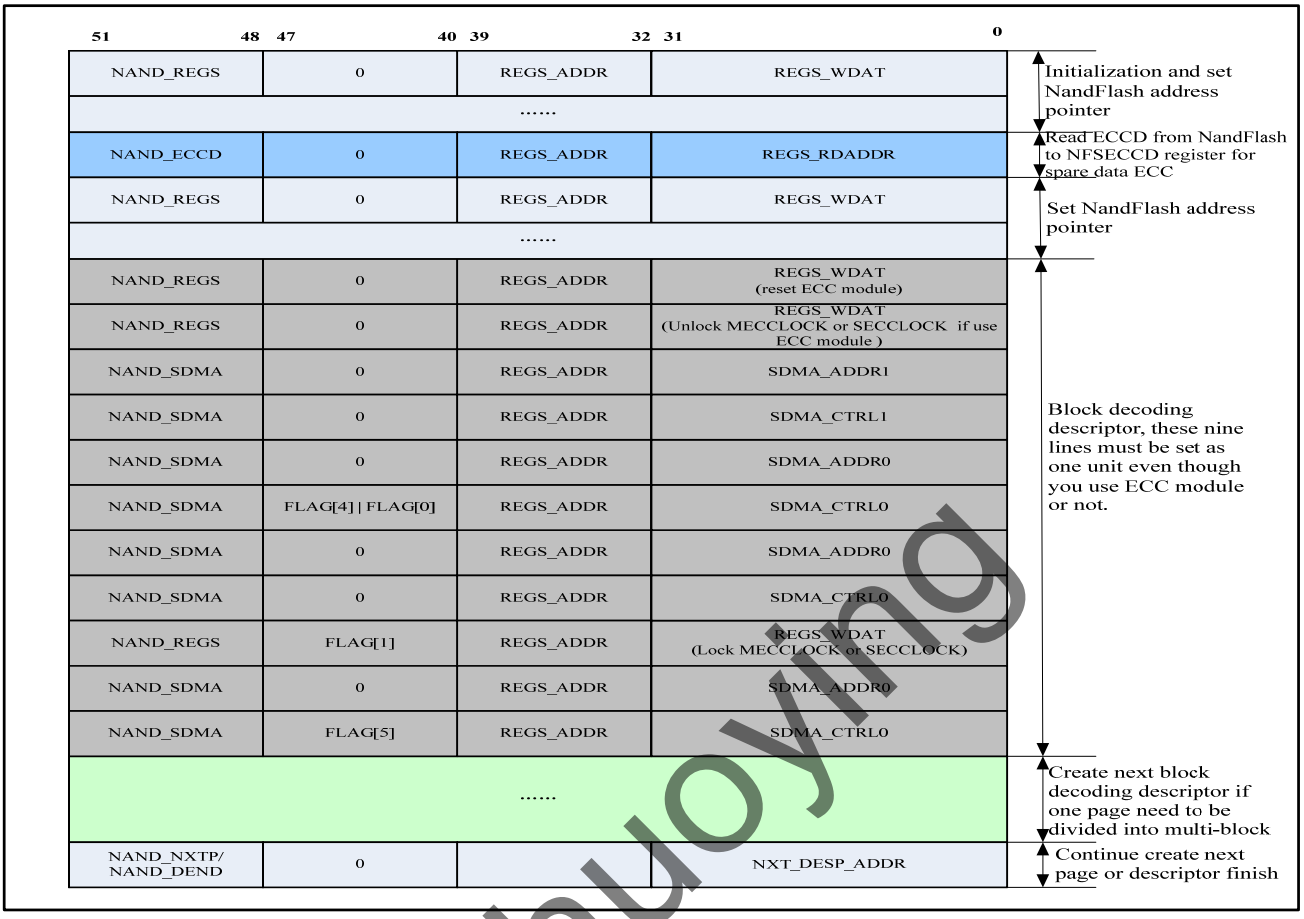


Figure 7-9: Decoding Descriptor Table Diagram

7.3 NFC Register Description

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.

7.3.1 NFC Register Memory Map

Table 7-6: NFC Register Memory Map

Address	Symbol	Direction	Description
0x20C5_0000	NFCONF	R/W	NAND Flash configuration register
0x20C5_0004	NFCONT	R/W	NAND Flash control Register
0x20C5_0008	NFCMD	R/W	NAND Flash command Register
0x20C5_000C	NFADDR	R/W	NAND Flash address Register
0x20C5_0010	NFDATA	R/W	NAND Flash data Register
0x20C5_0014	NFMECCD0	R/W	NAND flash main area ECC0 register
0x20C5_0018	NFMECCD1	R/W	NAND flash main area ECC1 register
0x20C5_001C	NFMECCD2	R/W	NAND flash main area ECC2 register
0x20C5_0020	NFSBLK	R/W	NAND flash start block address register
0x20C5_0024	NFEBLK	R/W	NAND flash end block address register
0x20C5_0028	NFSTAT	R/W	NAND flash operation status register
0x20C5_002C	NFECCERR0	RO	NAND flash ECC error status0 register
0x20C5_0030	NFECCERR1	RO	NAND flash ECC error status1 register
0x20C5_0034	NFMECC0	RO	NAND flash generated ECC status0 register
0x20C5_0038	NFMECC1	RO	NAND flash generated ECC status1 register
0x20C5_003C	NFMECC2	RO	NAND flash generated ECC status2 register
0x20C5_0040	NFECCERR2	RO	NAND flash ECC error status2 register
0x20C5_0044	NFSECCD	R/W	NAND flash spare area ECC register
0x20C5_0048	NFSECC	RO	NAND flash generated Spare area ECC status register
0x20C5_004C	NFADMACFG	R/W	NAND flash ADMA configuration register
0x20C5_0050	NFADMAADDR	R/W	NAND flash ADMA address register
0x20C5_0054	NFADMALINE	R/W	NAND flash ADMA page line number register
0x20C5_0058	NFADMABERR	R/W	NAND flash ADMA block decoding error status register
0x20C5_005C	NFADMACNT	RO	NAND flash ADMA counter status register
0x20C5_0080	NFSDMAADDR0	R/W	NAND flash SDMA Address0 Register
0x20C5_0084	NFSDMACTRL0	R/W	NAND flash SDMA Control0 Register
0x20C5_0088	NFSDMAADDR1	R/W	NAND flash SDMA Address1 Register
0x20C5_008C	NFSDMACTRL1	R/W	NAND flash SDMA Control1 Register

7.3.2 NFC Registers and Field Descriptions

Register 7-1: Nand Flash Configuration Register (NFCONF, offset = 0x0)

Field	Symbol	Direction	Description	Default
[24]	ECCTYPE	R/W	ECC Type Selection 0: SLC (1-bit correction) ECC 1:MLC (4-burst length correction) ECC	0
[14:12]	TACLS	R/W	CLE & ALE duration setting value (0~7) Duration = HCLK x TACLS	1
[10:8]	TWRPH0	R/W	TWRPH0 duration setting value (0~7) Duration = HCLK x (TWRPH0 + 1)	2
[6:4]	TWRPH1	R/W	TWRPH1 duration setting value (0~7) Duration = HCLK x (TWRPH1 + 1)	1
[3:1]	Reserved	R/W	Reserved	0
[0]	BUSWID	R/W	NAND Flash Memory I/O bus width for general access. 0: 8-bit bus 1: 16-bit bus	0

Register 7-2: Nand Flash Control Register (NFCONT, offset = 0x4)

Field	Symbol	Direction	Description	Default
[21]	ENADMADERR	R/W	ADMA ECC decoding error interrupt enable 1=enable, 0=disable	0
[20]	ENADMAEND	R/W	ADMA transfer finish interrupt enable 1=enable, 0=disable	0
[19]	ENADMAPERR	R/W	ADMA page program error interrupt enable. 1=enable, 0=disable	0
[18]	MLCDIR	R/W	4-burst length ECC encoding / decoding control 0: Decoding 4-burst length ECC, It is used for page read 1: Encoding 4-burst length ECC, It is be used for page program	0
[17]	LOCKTIGHT	R/W	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 (can not cleared by software). When it is set to 1, the area setting in SBLKADDR (in NFSBLK register) to EBLKADDR-1 (in NFEBLK register) 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 occur (NFSTAT [3] bit will be set). If the SBLKADDR and EBLKADDR are same, entire area will be locked.	0
[16]	LOCKSOFT	R/W	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 SBLKADDR (in NFSBLK register) to EBLKADDR-1 (in NFEBLK register) 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 occur (NFSTAT [3] bit will be set). If the SBLKADDR and EBLKADDR are same, entire area will be locked.	0
[15]	Reserved	R/W	Reserved,	0
[14]	ENSDMAEND	R/W	SDMA Transfer finish interrupt enable 1=enable, 0=disable	0
[13]	ENECCENCINT	R/W	4-burst length ECC encoding completion interrupt control 1=enable, 0=disable	0
[12]	ENECCDECINT	R/W	4-burst length ECC decoding completion interrupt control 1=enable, 0=disable	0
[10]	ENACCINT	R/W	Illegal access interrupt control Illegal access interrupt is occurs when CPU/DMA tries to program or erase locking area (the area setting in SBLKADDR (in NFSBLK register) to EBLKADDR-1 (in NFEBLK register)). 1=enable, 0=disable	0
[9]	ENRNBINT	R/W	RnB status input signal transition interrupt control 1=enable, 0=disable	0
[8]	RNBTRNMODE	R/W	RnB transition detection configuration 0=Detect rising edge 1=Detect falling edge	0
[7]	MECCLOCK	R/W	Lock Main area ECC generation 0: Unlock Main area ECC 1: Lock Main area ECC Main area ECC status register is NFMECC0/1/2	1

[6]	SECCLOCK	R/W	Lock Spare area ECC generation 0: Unlock Spare area ECC 1: Lock Spare area ECC Spare area ECC status register is NFSECC	1
[5]	INITMECC	RWAC	Initialize main area ECC decoder/encoder, active high	0
[4]	INITSECC	RWAC	Initialize spare area ECC decoder/encoder, active high	0
[2]	NANDCS1N	R/W	Nand flash chip select 1 signal control	1
[1]	NANDCS0N	R/W	Nand flash chip select 0 signal control 0: Force nFCE to low(Enable chip select) 1: Force nFCEto High(Disable chip select) Note: During boot time, it is controlled automatically. This value is only valid while MODE bit is 1	1
[0]	MODE	R/W	NAND Flash controller operating mode 0: NAND Flash Controller Disable (Don't work) 1: NAND Flash Controller Enable	0

Register 7-3: Nand Flash Command Register (NFCMD, offset = 0x8)

Field	Symbol	Direction	Description	Default
[7:0]	NFCMD	R/W	NAND Flash memory command value	0

Register 7-4: Nand Flash Address Register (NFADDR, offset = 0xC)

Field	Symbol	Direction	Description	Default
[7:0]	NFADDR	R/W	NAND Flash memory address value	0

Register 7-5: Nand Flash Data Register (NFDATA, offset = 0x10)

Field	Symbol	Direction	Description	Default
[31:0]	NFDATA	R/W	NAND Flash read/program data value for I/O **Note**: Please refer to Data Register Configuration	0

Register 7-6: Nand Flash Main Area ECC0 Register (NFMECCD0, offset = 0x14)

When 8-bit width SLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	ECCData3	R/W	4th byte ECC for I/O[7:0]	0
[23:16]	ECCData2	R/W	3rd byte ECC for I/O[7:0]	0
[15:8]	ECCData1	R/W	2nd byte ECC for I/O[7:0]	0
[7:0]	ECCData0	R/W	1st byte ECC for I/O[7:0]	0

When 16-bit width SLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	ECCData1_1	R/W	2nd byte ECC for I/O[15:8]	0
[23:16]	ECCData1_0	R/W	2nd byte ECC for I/O[7:0]	0
[15:8]	ECCData0_1	R/W	1st byte ECC for I/O[15:8]	0
[7:0]	ECCData0_0	R/W	1st byte ECC for I/O[7:0]	0

When 8/16-bit width MLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:27]	ECCData3	R/W	4th parity [4:0]	0
[26:18]	ECCData2	R/W	3rd parity [8:0]	0

[17:9]	ECCData1	R/W	2nd parity [8:0]	0
[8:0]	ECCData0	R/W	1st parity [8:0]	0

Register 7-7: Nand Flash Main Area ECC1 Register (NFMECCD1, offset = 0x18)

When 16-bit width SLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	ECCData3_1	R/W	4th byte ECC for I/O[15:8]	0
[23:16]	ECCData3_0	R/W	4th byte ECC for I/O[7:0]	0
[15:8]	ECCData2_1	R/W	3rd byte ECC for I/O[15:8]	0
[7:0]	ECCData2_0	R/W	3rd byte ECC for I/O[7:0]	0

When 8/16-bit width MLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31]	ECCData7	R/W	8th parity [0]	0
[30:22]	ECCData6	R/W	7th parity [8:0]	0
[21:13]	ECCData5	R/W	6th parity [8:0]	0
[12:4]	ECCData4	R/W	5th parity [8:0]	0
[3:0]	ECCData3	R/W	4th parity [8:5]	0

Register 7-8: Nand Flash Main Area ECC2 Register (NFMECCD2, offset = 0x1C)

When 8/16-bit width MLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[7:0]	ECCData7	R/W	8th parity [8:1]	0

Note1: Before reading main area data from Nand Flash, you should configure NFMECCD0/1/2 register first.

Note2: When connect to 8-bit width SLC NandFlash, only NFMECCD0 is used. When connect to 16-bit width SLC NandFlash, NFMECCD0 and NFMECCD1 are used. NFMECCD0/1/2 are used when connect to MCL NandFlash, regardless 8-bit or 16-bit width.

Register 7-9: Nand Flash Start Block Address Register (NFSBLK, offset = 0x20)

Field	Symbol	Direction	Description	Default
[25:24]	PGCOLCYCLE	R/W	Page column address cycle, range from 1 to 3.	1
[23:16]	SBLKADDR2	R/W	The 3rd cycle start block address of programmable block area	0
[15:8]	SBLKADDR1	R/W	The 2nd cycle start block address of programmable block area	0
[7:0]	SBLKADDR0	R/W	The 1st cycle start block address of programmable block area	0

Register 7-10: Nand Flash End Block Address Register (NFEBLK, offset = 0x24)

Field	Symbol	Direction	Description	Default
[23:16]	EBLKADDR2	R/W	The 3rd cycle end block address of programmable block area	0
[15:8]	EBLKADDR1	R/W	The 2nd cycle end block address of programmable block area	0
[7:0]	EBLKADDR0	R/W	The 1st cycle end block address of programmable block area	0

The SBLKADDR and EBLKADDR can be changed while LOCKSOFT bit (NFCONT[12]) is enabled. But cannot be changed when LOCKTIGHT bit (NFCONT[13]) is set. When SBLKADDR is equal to EBLKADDR, the whole NandFlash is locked and read only, as shown in followed figure.

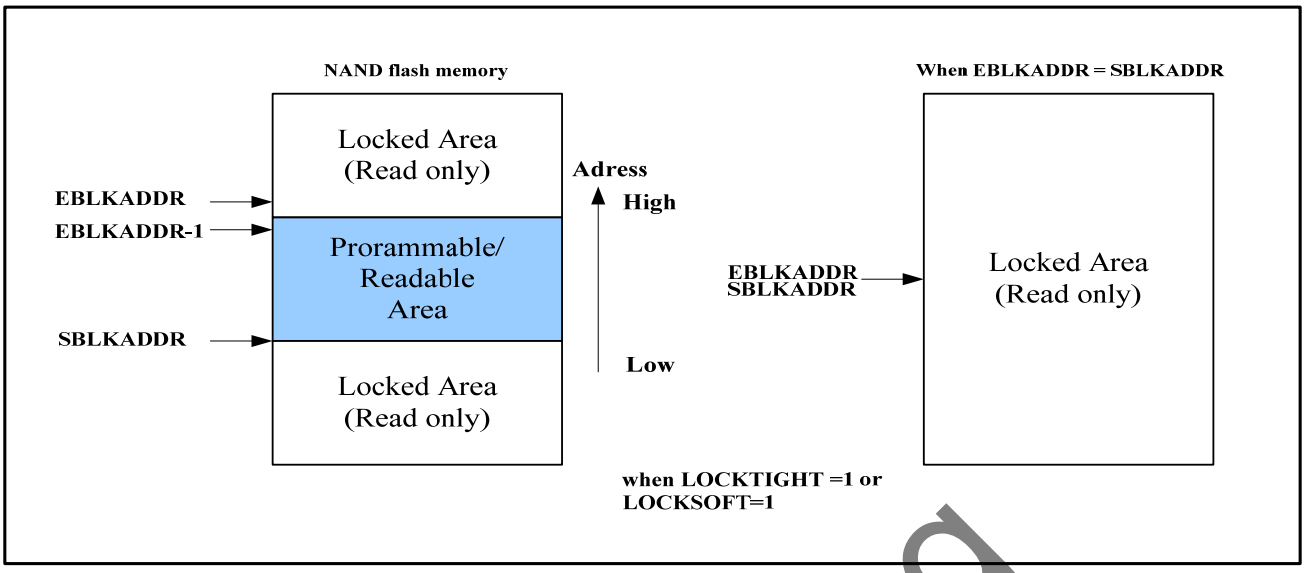


Figure 7-10: NandFlash Locked Area Diagram

Register 7-11: Nand Flash Operation Status Register (NFSTAT, offset = 0x28)

Field	Symbol	Direction	Description	Default
[13]	ABLKDERRSTA	RW1C	ADMA Block ECC Decoding error status. If any of the bits in NFADMABERR register are set, then this bit is set. Therefore user can efficiently test for an error by checking this bit first after ADMA transfer complete. Write this bit to 1, hardware will automatically clear ADMA Block Decoding Error Status register and this bit.	0
[12]	ADMADERRSTA	RW1C	ADMA ECC Decoding error interrupt status. If block ECC decoding error interrupt enable in ADMA descriptor table, ADMA controller will pause when block ECC decoding error occurs and set this bit, then wait for software's action. If ENADMADERR bit in NFCONT register is set, NandFlash controller interrupt will generate. To clear this write to 1.	0
[11]	ADMAENDSTA	RW1C	ADMA transfer finish status. If ENADMAEND bit in NFCONT register is set, NandFlash controller interrupt will generate. To clear this write to 1.	0
[10]	ADMAPERRSTA	RW1C	ADMA page program error status. During ADMA based program transfer, if NandFlash memory doesn't return Successful Program status, this bit will be set. If ENADMAPERR bit in NFCONT register is set, NandFlash controller interrupt will generate. To clear this write to 1.	0
[9]	Reserved	RO	Reserved. This bit must always be set to 0.	0
[8]	SDMAENDSTA	RW1C	SDMA transfer complete status. If ENSDMAEND bit in NFCONT register is set, NandFlash controller interrupt will generate. To clear this write to 1.	0
[7]	ECCENCDONE	RW1C	When 4-burst length ECC encoding is finished, this bit is set and issue interrupt if ENECCENCINT bit in NFCONT register is set. The NFMECC0, NFMECC1 and NFMECC2 have valid encoding	0

			parity values. To clear this write to 1.	
[6]	ECCDECDONE	RW1C	When 4-burst length ECC decoding is finished, this bit is set and issue interrupt if ENECCDECINT bit in NFCONT register is set. The NFECCERR0, NFECCERR1 and NFECCERR2 have valid values and user can check whether decoding error occurs. To clear this write to 1.	0
[5]	ACCINTSTA	RW1C	Once LOCKSOFT or LOCKTIGHT is enabled, the illegal access (program, erase) to the NandFlash memory makes this bit set. If ENACCINT bit in NFCONT register is set, NandFlash controller interrupt will generate. 0: illegal access is not detected 1: illegal access is detected	0
[4]	RNBINTSTA	RW1C	When RnB low to high transition is occurred, this value set and issue interrupt if ENRNBINT bit in NFCONT register is set. To clear this write '1'. 0: RnB transition is not detected 1: RnB transition is detected Transition configuration is set by RNBTRNMODE bit in NFCONT register.	0
[2]	NFCESTA	RO	The status of nFCE output pin	1
[0]	RNBSTA	RO	The status of RnB input pin 0: NAND Flash memory busy 1: NAND Flash memory ready to operate	1

Register 7-12: Nand Flash ECC Error Status0 Register (NFECCERR0, offset = 0x2C)

When SLC NandFlash is selected, this register is used for IO[7:0].

Field	Symbol	Direction	Description	Default
[24:21]	SERRDATANO	RO	In spare area, Indicates which number data is error.	0
[20:18]	SERRBITNO	RO	In spare area, Indicates which bit is error	0
[17:7]	MERRDATANO	RO	In main data area, Indicates which number data is error	0
[6:4]	MERRBITNO	RO	In main data area, Indicates which bit is error	0
[3:2]	SERRSTA	RO	Indicates whether spare area bit fail error occurred. 00: No Error 01: 1-bit error(correctable) 10: Multiple error 11: ECC area error	0
[1:0]	MERRSTA	RO	Indicates whether main area bit fail error occurred. 00: No Error 01: 1-bit error(correctable) 10: Multiple error 11: ECC area error	0

When MLC NandFlash is selected, NFECCERR0, NFECCERR1 and NFECCERR2 are used for decoding result status.

Field	Symbol	Direction	Description	Default
[31]	ECCBUSY	RO	Indicates the 4-burst length ECC decoding engine is searching whether a error exists or not 0=Idle 1=Busy	0
[30]	ECCRDY	RO	Indicates the 4-burst length ECC decoding engine is finished and the other fields of NFECCERR0, NFECCERR1 and NFECCERR2 register have valid values. 1=ready. 0=not ready	0

[29:27]	ECCERRSTA	RO	4-burst length ECC decoding result 000: No error 001: 1-burst error 010: 2-burst error 011: 3-burst error 100: 4-burst error 101: Uncorrectable 11x: Reserved	0
[24:16]	BITERRPAT0	RO	1st burst error bit pattern, each bit corresponding to one bit in current burst error. 1=current bit error, need to correct by software.	0
[8:0]	BITERRLOC0	RO	1st burst error location, range from 0 to 455.	0

Register 7-13: Nand Flash ECC Error Status1 Register (NFECCERR1, offset = 0x30)

When SLC NandFlash is selected, this register is used for IO[15:8].

Field	Symbol	Direction	Description	Default
[24:21]	SERRDATANO	RO	In spare area, Indicates which number data is error.	0
[20:18]	SERRBITNO	RO	In spare area, Indicates which bit is error	0
[17:7]	MERRDATANO	RO	In main data area, Indicates which number data is error	0
[6:4]	MERRBITNO	RO	In main data area, Indicates which bit is error	0
[3:2]	SERRSTA	RO	Indicates whether spare area bit fail error occurred. 00: No Error 01: 1-bit error(correctable) 10: Multiple error 11: ECC area error	0
[1:0]	MERRSTA	RO	Indicates whether main area bit fail error occurred. 00: No Error 01: 1-bit error(correctable) 10: Multiple error 11: ECC area error	0

When MLC NandFlash is selected, NFECCERR0, NFECCERR1 and NFECCERR2 are used for decoding result status.

Field	Symbol	Direction	Description	Default
[26:18]	BITERRLOC4	RO	4th burst error location, range from 0 to 455.	0
[17:9]	BITERRLOC3	RO	3rd burst error location, range from 0 to 455.	0
[8:0]	BITERRLOC2	RO	2nd burst error location, range from 0 to 455.	0

Register 7-14: Nand Flash ECC Error Status2 Register (NFECCERR2, offset = 0x40)

When MLC NandFlash is selected, NFECCERR0, NFECCERR1 and NFECCERR2 are used for decoding result status.

Field	Symbol	Direction	Description	Default
[26:18]	BITERRPAT4	RO	4th burst error bit pattern, each bit corresponding to one bit in current burst error. 1=current bit error, need to correct by software.	0
[17:9]	BITERRPAT3	RO	3rd burst error bit pattern, each bit corresponding to one bit in current burst error. 1=current bit error, need to correct by software.	0
[8:0]	BITERRPAT2	RO	2nd burst error bit pattern, each bit corresponding to one bit in current burst error. 1=current bit error, need to correct by software.	0

Register 7-15: Nand Flash Generated ECC Status0 Register (NFMECC0, offset = 0x34)

When 8-bit width SLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	MECCData0_3	RO	4th byte ECC for I/O[7:0]	0
[23:16]	MECCData0_2	RO	3rd byte ECC for I/O[7:0]	0
[15:8]	MECCData0_1	RO	2nd byte ECC for I/O[7:0]	0
[7:0]	MECCData0_0	RO	1st byte ECC for I/O[7:0]	0

When 16-bit width SLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	MECCData1_1	RO	2nd byte ECC for I/O[15:8]	0
[23:16]	MECCData0_1	RO	2nd byte ECC for I/O[7:0]	0
[15:8]	MECCData1_0	RO	1st byte ECC for I/O[15:8]	0
[7:0]	MECCData0_0	RO	1st byte ECC for I/O[7:0]	0

When 8/16-bit width MLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:27]	ECCData3	RO	4th check parity [4:0]	0
[26:18]	ECCData2	RO	3rd check parity [8:0]	0
[17:9]	ECCData1	RO	2nd check parity [8:0]	0
[8:0]	ECCData0	RO	1st check parity [8:0]	0

Register 7-16: Nand Flash Generated ECC Status1 Register (NFMECC1, offset = 0x38)

When 16-bit width SLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	MECCData1_3	RO	4th byte ECC for I/O[15:8]	0
[23:16]	MECCData0_3	RO	4th byte ECC for I/O[7:0]	0
[15:8]	MECCData1_2	RO	3rd byte ECC for I/O[15:8]	0
[7:0]	MECCData0_2	RO	3rd byte ECC for I/O[7:0]	0

When 8/16-bit width MLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[31]	ECCData7	RO	8th parity [0]	0
[30:22]	ECCData6	RO	7th parity [8:0]	0
[21:13]	ECCData5	RO	6th parity [8:0]	0
[12:4]	ECCData4	RO	5th parity [8:0]	0
[3:0]	ECCData3	RO	4th parity [8:5]	0

Register 7-17: Nand Flash Generated ECC Status2 Register (NFMECC2, offset = 0x3C)

When 8/16-bit width MLC NandFlash is used.

Field	Symbol	Direction	Description	Default
[7:0]	ECCData7	RO	8th parity [8:1]	0

Note1: When writing main area data to Nand Flash, the ECC status can be generated in NFMECC0, NFMECC1 and NFMECC2 register while MECCLOCK in NFCONT register is set to 0 (unlock).

Note2: When connect to 8-bit width SLC NandFlash, only NFMECC0 is used. When connect to 16-bit width SLC NandFlash, NFMECC0 and NFMECC1 are used. NFMECC0/1/2 are used when connect to MCL NandFlash, regardless 8-bit or 16-bit width.

Register 7-18: Nand Flash Spare Area ECC Register (NFSECCD, offset = 0x44)

When 8-bit width NandFlash is used.

Field	Symbol	Direction	Description	Default
[15:8]	SECCData0_1	R/W	2nd byte ECC for I/O[7:0]	0
[7:0]	SECCData0_0	R/W	1st byte ECC for I/O[7:0]	0

When 16-bit width NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	SECCData1_1	R/W	2nd byte ECC for I/O[15:8]	0
[23:16]	SECCData0_1	R/W	2nd byte ECC for I/O[7:0]	0
[15:8]	SECCData1_0	R/W	1st byte ECC for I/O[15:8]	0
[7:0]	SECCData0_0	R/W	1st byte ECC for I/O[7:0]	0

Register 7-19: Nand Flash Spare Area ECC Status Register (NFSECC, offset = 0x48)

When 8-bit width NandFlash is used.

Field	Symbol	Direction	Description	Default
[15:8]	SECCData0_1	RO	2nd byte ECC for I/O[7:0]	0
[7:0]	SECCData0_0	RO	1st byte ECC for I/O[7:0]	0

When 16-bit width NandFlash is used.

Field	Symbol	Direction	Description	Default
[31:24]	SECCData1_1	RO	2nd byte ECC for I/O[15:8]	0
[23:16]	SECCData0_1	RO	2nd byte ECC for I/O[7:0]	0
[15:8]	SECCData1_0	RO	1st byte ECC for I/O[15:8]	0
[7:0]	SECCData0_0	RO	1st byte ECC for I/O[7:0]	0

Note: When writing spare area data to Nand Flash, the ECC status can be generated in NFSECC register while SECCLOCK in NFCONT register is set to 0 (unlock).

Register 7-20: Nand Flash AMDA Configuration Register (NFADMACFG, offset = 0x4C)

Field	Symbol	Direction	Description	Default
[1]	ADMATYP	R/W	Transfer type of ADMA descriptor. 1=Write data to NandFlash, using ECC encoder. 0=Read data from NandFlash, using ECC decoder.	0
[0]	ADMAENABLE	R/W	ADMA controller enable. 1=enable and start ADMA controller, 0=disable and reset ADMA **Note**: Before enable this bit, you should create ADMA descriptor first.	0

Register 7-21: Nand Flash AMDA Address Register (NFADMAADDR, offset = 0x50)

Field	Symbol	Direction	Description	Default
[31:0]	ADMAADDR	R/W	ADMA system address. This register holds byte address of executing command of the Descriptor table. At the start of ADMA, software shall set start address of the	0

			Descriptor table. The ADMA increments this register address, which points to next page, when fetching one page of descriptor table.	

Register 7-22: Nand Flash AMDA Page Line Number Register (NFADMALINE, offset = 0x54)

Field	Symbol	Direction	Description	Default
[5:0]	ADMALINE	R/W	Maximum line of one block descriptor table. ADMA controller will prefetch some descriptor lines, this register is used to inform controller the maximum line of one block descriptor table.	32

Register 7-23: Nand Flash AMDA Block Decoding Error Status Register (NFADMABERR, offset = 0x58)

Field	Symbol	Direction	Description	Default
[31:0]	ADMABERR	R/W	Each bit reflects decoding error status of every block in current corresponding page. Bit 0 correspond to block 0, bit 1 correspond to block 1, and so on. Write ABLKDERRSTA bit in NFSTAT register to 1 will clear this register. **Note1**: Page number can be checked by PAGECNT in NFADMACNT register. **Note2**: If block ECC decoding error interrupt enable in ADMA descriptor table, ADMA controller will pause when block ECC decoding error occurs and set corresponding bit, then wait for software's action. Software need to check which block is error and correct the data according to decoding result. Write 0xA5 to this register, ADMA controller will continue and execute next block transfer or next page descriptor.	0

Register 7-24: Nand Flash AMDA Counter Register (NFADMACNT, offset = 0x5C)

Field	Symbol	Direction	Description	Default
[8:4]	ADMABLKCNT	RO	Block counter of current page transfer. ADMA controller increments this counter after one block of page transfer complete	0
[3:0]	ADMAPGECNT	RO	Page counter of current descriptor. ADMA controller increments this counter after one page transfer complete	0

Register 7-25: Nand Flash SMDA Address0 Register (NFSDMAADDR0, offset = 0x80)

Field	Symbol	Direction	Description	Default
[31:0]	SDMAADDR0	R/W	SDMA system address 0. This register contains the system memory base address for SDMA transfer.	0

Register 7-26: Nand Flash SMDA Control0 Register (NFSDMACTRL0, offset = 0x84)

Field	Symbol	Direction	Description	Default
[31]	SDMAENABLE	R/W	SDMA controller enable bit. Once this bit is set, SDMA controller will transfer data from system memory to NandFlash when SDMADIR bit is set to 0, or transfer data from NandFlash to system memory when SDMADIR bit is set to 1.	0

[30:28]	Reserved	R/W	Reserved. These bits must always be set to 0.	0
[27]	SDMAAMODE	R/W	SDMA address mode. 0=only use one address, i.e. SDMAADDR0. 1=use two address, i.e. SDMAADDR0 and SDMAADDR1.	0
[26]	SDMAAMDEN	R/W	SDMA address mode enable, when SDMAAMODE is set 1, this bit must be set.	0
[25]	SDMADIR	R/W	SDMA transfer direction. 1=transfer data from NandFlash to system memory (reading) 0=transfer data from system memory to NandFlash (writing)	0
[24:0]	SDMALEN0	R/W	SDMA transfer length 0 (byte number). Note: If connect to 16-bit width NandFlash, transfer length should be even.	0

Register 7-27: Nand Flash SMDA Address1 Register (NFSDMAADDR1, offset = 0x88)

Field	Symbol	Direction	Description	Default
[31:0]	SDMAADDR1	R/W	SDMA system address 1. When using two address mode, this register is the next address for SDMA transfer.	0

Register 7-28: Nand Flash SMDA Control1 Register (NFSDMACTRL1, offset = 0x8C)

Field	Symbol	Direction	Description	Default
[27:26]	Reserved	R/W	Reserved, these bits must always be set to 0.	0
[25]	SDMADIR1	R/W	SDMA transfer direction. 1=transfer data from NandFlash to system memory (reading) 0=transfer data from system memory to NandFlash (writing) Note: this bit must be the same as SDMADIR bit in NFSDMAGCFG register.	0
[24:0]	SDMALEN1	R/W	SDMA transfer length 0 (byte number). Note: If connect to 16-bit width NandFlash, transfer length should be even.	0

8 CF/IDE Controller

8.1 Overview

IDEC is an IDE controller. It can be used to connect the external storage device that are compatible IDE or ATA interface such as hard disk and CF card and so on. Through the internal DMA module and enhance ATA protocol support, IDEC of IMAPx210 can transfer the data between external storage devices and internal system memory in a very high speed.

Features

- Support True-IDE PIO(up to mode 6)
- Support True-IDE Ultra-DMA mode(up to mode 5)
- Support ATA/ATAPI-6 spec
- Support CF standard spec. R4.1
- Support Multi-block write and read
- Interrupt, polled-mode or DMA mode operation
- Internal DMA module with a 64 bit AHB master interface

Block Diagram

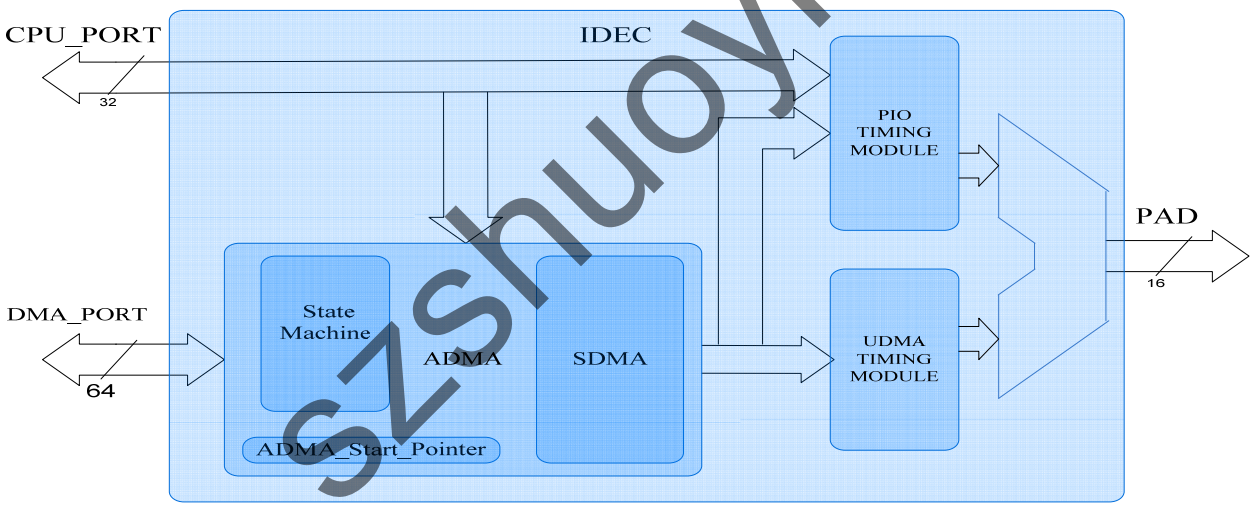


Figure 8-1: IDEC Block Diagram

Figure 8-1 shows the functional block diagram of IDEC. There are three main modules in IDEC – PIO module, UDMA module and ADMA module. PIO is a single mode to access external IDE device. UDMA will get a high speed transfer. ADMA is a internal DMA module to transfer the data to system memory directly and fast the access of IDE debvice.

8.2 Functional Description

8.2.1 IDE Bus Protocol

PIO transfer

Host access the registers of device by using PIO timing model with the parameter for accessing register. Host access data of device by using PIO timing model with the parameter form accessing data.

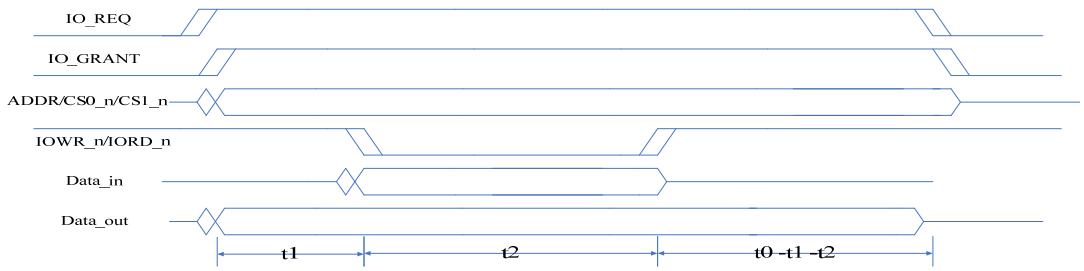


Figure 8-2: PIO Timing

Table 8-1: I/O Decoding

-CS1	-CS0	A2	A1	A0	-DMACK	-IORD = 0	-IORW = 0	Note
1	0	0	0	0	1	PIO RD Data	PIO WR Data	8 or 16 bit
1	1	X	X	X	0	DMA RD Data	DMA WR Data	16 bit
1	0	0	0	1	1	Error Register	Features	8 bit
1	0	0	1	0	1	Sector Count	Sector Count	8 bit
1	0	0	1	1	1	Sector No.	Sector No.	8 bit
1	0	1	0	0	1	Cylinder Low	Cylinder Low	8 bit
1	0	1	0	1	1	Cylinder High	Cylinder High	8 bit
1	0	1	1	0	1	Select Card/Head	Select Card/Head	8 bit
1	0	1	1	1	1	Status	Command	8 bit
0	1	1	1	0	1	Alt Status	Device Control	8 bit

Table 8-2: PIO Register Transfer To/From Device

Register Transfer Timing Parameters		Mode0 (ns)	Mode1 (ns)	Mode2 (ns)	Mode3 (ns)	Mode4 (ns)	Note
t_0	Cycle time (min)	600	383	330	180	120	1,4,5
t_1	Address valid to DIOR-/DIOW- setup (min)	70	50	30	30	25	
t_2	DIOR-/DIOW- pulse width 8-bit (min)	290	290	290	80	70	1
t_{2i}	DIOR-/DIOW- recovery time (min)	-	-	-	70	25	1
t_3	DIOW- data setup (min)	60	45	30	30	20	
t_4	DIOW- data hold (min)	30	20	15	10	10	
t_5	DIOR- data setup (min)	50	35	20	20	20	
t_6	DIOR- data hold (min)	5	5	5	5	5	
t_{6z}	DIOR- data release (min)	30	30	30	30	30	2
t_9	DIOR-/DIOW- to address valid hold (min)	20	15	10	10	10	
t_{RD}	Read Data valid to IORDY active (min)	0	0	0	0	0	
t_A	IORDY Setup time (min)	35	35	35	35	35	3
t_B	IORDY Pulse time (min)	1250	1250	1250	1250	1250	
t_C	IORDY assertion to release (min)	5	5	5	5	5	

Note1: t_0 is the minimum cycle time, t_2 is the minimum DIOR-/DIOW- assertion time, and t_2i is the minimum DIOR-/DIOW- negation time. A host implement shall lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

Note2: This parameter specifies the time from the negation edge of DIOR- to the time that the data bus is released by the device.

Note3: The delay from the activation of DIOR- or DIOW- until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the register transfer cycle is completed. If the device is not driving IORDY negated at tA after the activation of DIOR- or DIOW-, then t5 shall be met and tRD is not applicable. If the device is driving IORDY negated at the time tA after the activation of DIOR- or DIOW-, then tRD shall be met and t5 is not applicable.

Note4: ATA/ATAPI standards prior to ATA/ATAPI-5 inadvertently specified an incorrect value for mode 2 time t0 by utilizing the 16-bit PIO value.

Note5: Mode shall be selected no higher than the highest mode supported by the slowest device.

Table 8-3: PIO Data Transfer To/From Device

Register Transfer Timing Parameters		Mode0 (ns)	Mode1 (ns)	Mode2 (ns)	Mode3 (ns)	Mode4 (ns)	Note
t_0	Cycle time (min)	600	383	240	180	120	1,4
t_1	Address valid to DIOR-/DIOW- setup (min)	70	50	30	30	25	
t_2	DIOR-/DIOW- (min)	165	125	100	80	70	1
t_{2i}	DIOR-/DIOW- recovery time (min)	-	-	-	70	25	1
t_3	DIOW- data setup (min)	60	45	30	30	20	
t_4	DIOW- data hold (min)	30	20	15	10	10	
t_5	DIOR- data setup (min)	50	35	20	20	20	
t_6	DIOR- data hold (min)	5	5	5	5	5	
t_{6z}	DIOR- data release (max)	30	30	30	30	30	2
t_9	DIOR-/DIOW- to address valid hold (min)	20	15	10	10	10	
t_{RD}	Read Data valid to IORDY active (min)	0	0	0	0	0	
t_A	IORDY Setup time (min)	35	35	35	35	35	3
t_B	IORDY Pulse time (max)	1250	1250	1250	1250	1250	
t_C	IORDY assertion to release (max)	5	5	5	5	5	

Note1: t_0 is the minimum cycle time, t_2 is the minimum DIOR-/DIOW- assertion time, and t_2i is the minimum DIOR-/DIOW- negation time. A host implement shall lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the devices IDENTIFY DEVICE data. A device implementation shall support any legal host implementation.

Note2: This parameter specifies the time from the negation edge of DIOR- to the time that the data bus is released by the device.

Note3: The delay from the activation of DIOR- or DIOW- until the state of IORDY is first sampled. If IORDY is inactive then the host shall wait until IORDY is active before the register transfer cycle is completed. If the device is not driving IORDY negated at t_A after the activation of DIOR- or DIOW-, then t_5 shall be met and t_{RD} is not applicable. If the device is driving IORDY negated at the time t_A after the activation of DIOR- or DIOW-, then t_{RD} shall be met and t_5 is not applicable.

Note4: Mode may be selected at the highest mode for the device if CS(1:0) and DA(2:0) do note change between read or write cycles or selected at the highest mode supported by the slowest device if CS(1:0) or DA(2:0) do change between read or write cycles.

Ultra DMA

Ultra DMA data transfer has two operation modes: Burst-In(read) and Burst-Out(write). There are three phases in every mode: initiation phase, transfer phase and termination phase. In transfer phase, both sender and recipient can pause or terminate the current burst transfer.

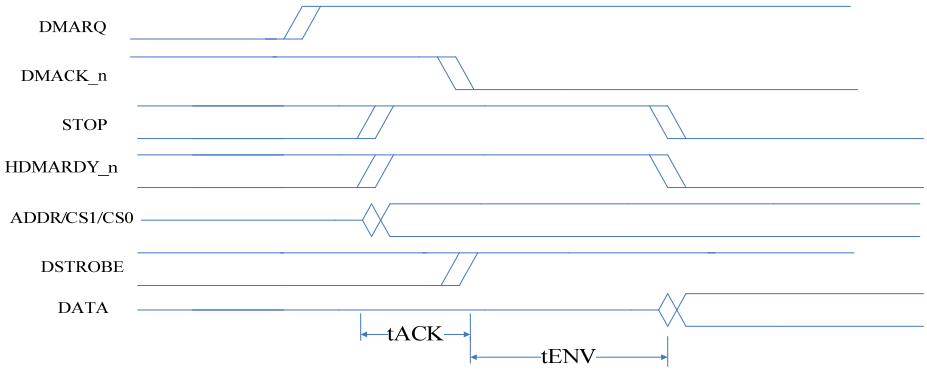


Figure 8-3: Ultra DMA Burst-In Initiation Timing

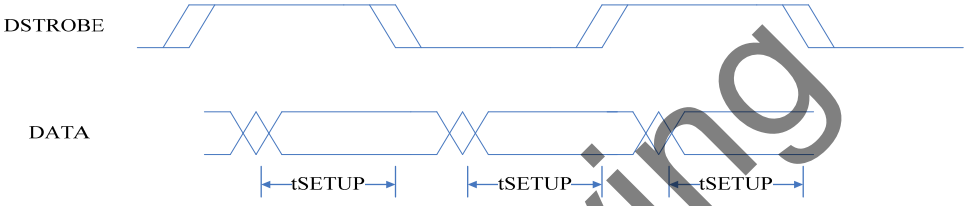


Figure 8-4: Sustain the Burst-In Timing, Clk Cycle Should Less Than t_{SETUP}

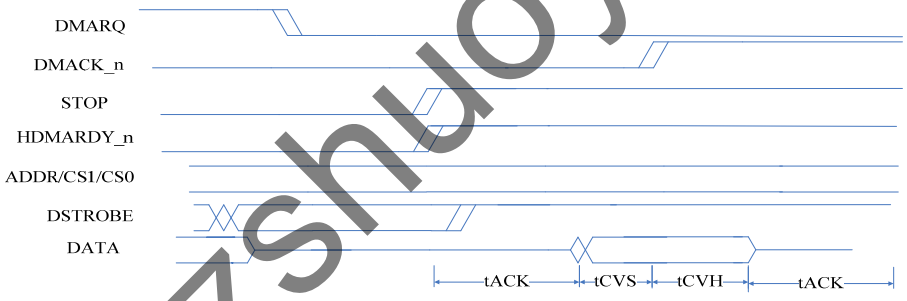


Figure 8-5: Device Terminate Burst-In Timing

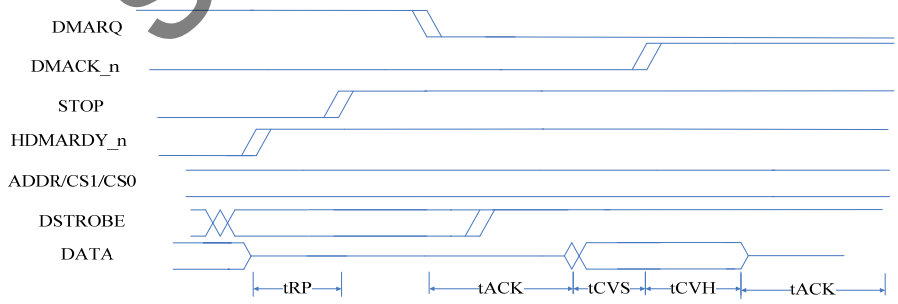


Figure 8-6: Host terminate Burst-In Timing

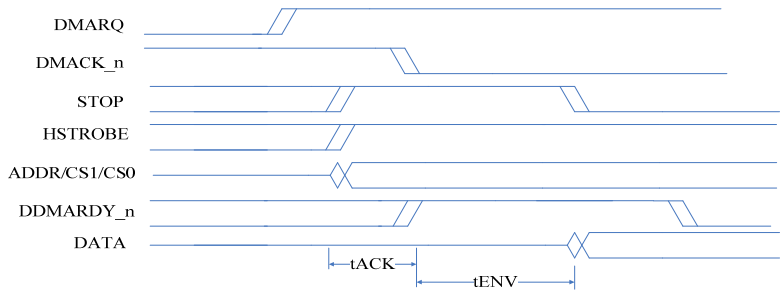


Figure 8-7: Ultra DMA Burst-Out initiation timing

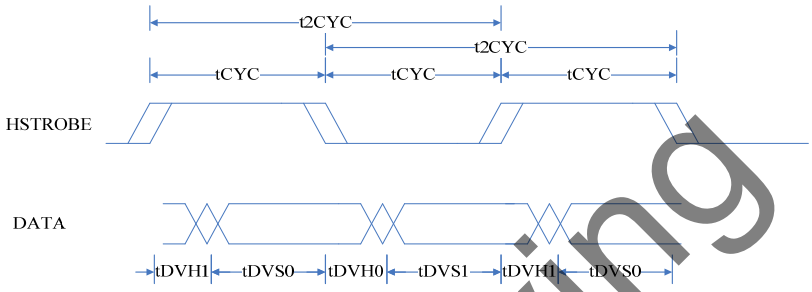


Figure 8-8: Sustain Burst-Out Timing, please set tDVS and tDVH according to tCYC and t2CYC

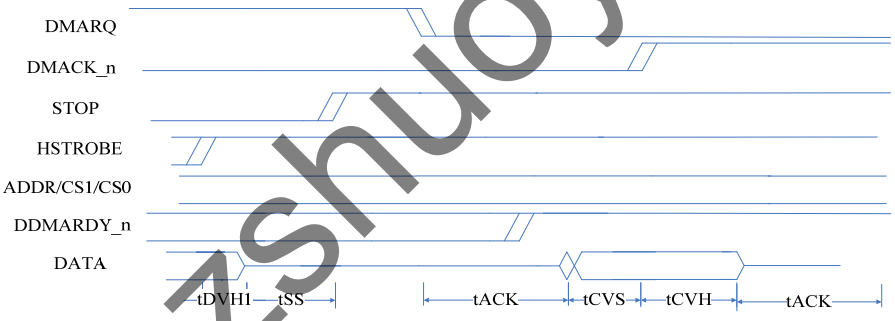


Figure 8-9: Host Terminate Burst-Out Timing

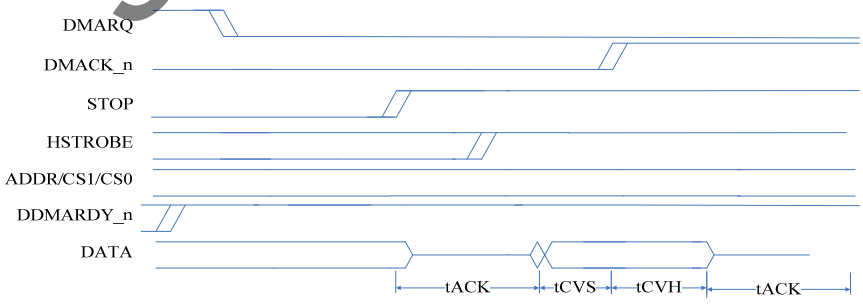


Figure 8-10: Device Terminate Burst-Out Timing

8.2.2 ADMA Descriptor

There is an internal DMA in IDEC and it can be used with description function to speed up the transfer and reduce the usage of CPU.

Table 8-4: IDEC AMDA Descriptor

Field	Symbol	Direction	Description
[63:32]	ADDRESS	RW	Data address in system memory
[31:16]	LENGTH	RW	Data length
[15:6]	Reserved	N/A	Reserved
[5:4]	ACTION	RW	Operation indication 00: NOP, Don't execute current line and go to next line 01: Reserved, same as NOP 10: TRAN, transfer data of current descriptor line 11: LINK, link to another descriptor
[3]	Reserved	N/A	Reserved
[2]	INT	RW	Interrupt indication 0: no interrupt when finish the transfer of current descriptor line 1: generate a interrupt to system when finish the transfer of current descriptor line
[1]	END	RW	End indication 0: descriptor not end 1: current descriptor is the last descriptor, and there will be a Transfer Complete Interrupt when current descriptor line is completed.
[0]	Reserved	N/A	Reserved

8.3 IDEC Register Description

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.
Rsvd or Reserved	Reserved. These bits are initialized to zero, and writes to them are ignored.

8.3.1 IDEC Register Memory Map

IDEC's register mapping address range in system is 0x20C7000~0x20C7FFF. IDEC register's system accessing address equal o BASE_ADDRESS (0x20C70000) plus address offset.

Table 8-5: IDEC Register Memory Map

Address	Symbol	Direction	Description
0x00	IDE_DEV_PIO_DATA	RW	IDE device PIO data register
0x04	IDE_DEV_ERR_FUNC	RW	IDE device error function register
0x08	IDE_DEV_SEC_CNT	RW	IDE device sector count register
0x0C	IDE_DEV_SEC_NO	RW	IDE device sector number register
0x10	IDE_DEV_CYL_LOW	RW	IDE device cylinder low register
0x14	IDE_DEV_CYL_HIGH	RW	IDE device cylinder high register
0x18	IDE_DEV_HEAD_DRV	RW	IDE device head driver register
0x1C	IDE_DEV_CMD_STAT	RW	IDE device command/status register

0x20	IDE_DEV_ALT_STAT	RW	IDE device alterative/status register
0x24	Reserved	RW	Reserved
0x28	IDE_HOST_STAT_FORCE	RW	IDE Host status force register
0x2C	IDE_HOST_CON	RW	IDE Host constant register
0x30	IDE_HOST_STAT_UNMASK	RW	IDE Host status unmask register
0x34	IDE_HOST_STAT_EN	RW	IDE Host status enable register
0x38	IDE_HOST_STAT	RW	IDE Host status register
0x3C	IDE_PIN_STATUS	RW	IDE Pin status register
0x40	IDE_PIO_TPARAM0	RW	IDE PIO timing parameter register 0
0x44	IDE_PIO_TPARAM1	RW	IDE PIO timing parameter register 1
0x48	IDE_PIO_TPARAM2	RW	IDE PIO timing parameter register 2
0x4C	IDE_UDMA_TPARAM0	RW	IDE UDMA timing parameter register 0
0x50	IDE_UDMA_TPARAM1	RW	IDE UDMA timing parameter register 1
0x54	IDE_UDMA_TPARAM2	RW	IDE UDMA timing parameter register 2
0x58	IDE_UDMA_TPARAM3	RW	IDE UDMA timing parameter register 3
0x5C	IDE_UDMA_TPARAM4	RW	IDE UDMA timing parameter register 4
0X60	IDE_UDMA_TPARAM5	RW	IDE UDMA timing parameter register 5
0x64	IDE_UDMA_TRIGGER	RW	IDE UDMA trigger register
0x68	IDE_UDMA_XFER_INFO	RW	IDE UDMA transfer information register
0x6C	IDE_UDMA_FIFO_STATUS	RW	IDE UDMA FIFO status register
0x70	IDE_UDMA_HTBST_CNT	RW	IDE UDMA Host terminate burst register
0x74	IDE_UDMA_BSTI_HP_CNT	RW	IDE UDMA Host pause burst-in register
0x78	IDE_UDMA_BSTO_HP_CNT	RW	IDE UDMA Host pause burst-out register
0x7C	IDE_UDMA_BSTO_DP_CNT	RW	IDE UDMA device pause burst-out register
0x80	IDE_SDMA_BUF0_ADDR	RW	IDE SDMA buffer 0 address register
0x84	IDE_SDMA_BUF0_CTRL	RW	IDE SDMA buffer 0 control register
0x88	IDE_SDMA_BUF1_ADDR	RW	IDE SDMA buffer 1 address register
0x8C	IDE_SDMA_BUF1_CTRL	RW	IDE SDMA buffer 1 control register
0x90	IDE_SDMA_BUF2_ADDR	RW	IDE SDMA buffer 2 address register
0x94	IDE_SDMA_BUF2_CTRL	RW	IDE SDMA buffer 3 control register
0x98	IDE_SDMA_BUF3_ADDR	RW	IDE SDMA buffer 3 address register
0x9C	IDE_SDMA_BUF3_CTRL	RW	IDE SDMA buffer 3 control register
0xA0	IDE_ADMA_DSCPT_POINTER	RW	IDE ADMA descriptor pointer register

8.3.2 IDEC Registers and Field Descriptions

Register 8-1: IDE_DEV_PIO_DATA

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_PIO_DATA	RW	PIO out data transfers are processed by a series of reads to this register each read transferring the data that follows the previous read. PIO in data transfers are processed by a series of writes to this register, each write transferring the data that follows the previous write. The results of a read during a PIO in or a write during a PIO out is indeterminant.	0

Register 8-2: IDE_DEV_ERR_FUNC

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0

[15: 0]	DEV_ERR_FUNC	RW	This register contains status for the current command. Following a power on, a reset, or completion of an EXECUTE DEVICE DIAGNOSTIC command, this register contains a diagnostic code. At the completion of any command except EXECUTE DEVICE DIAGNOSTIC, the contents of this register are valid when the ERR bit is equal to one in the Status register.	0

Register 8-3: IDE_DEV_SEC_CNT

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_SEC_CNT	RW	This register contains the number of sectors of data requested to be transferred on a read or write operation between the host and the device. If the value in this register is zero, a count of 256 sectors is specified.	0

Register 8-4: IDE_DEV_SEC_NO

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_SEC_NO	RW	If the LBA bit is cleared to zero in the Device/Head register, this register contains the starting sector number for any media access. If the LBA bit is set to one in the Device/Head register, this register contains Bits 7-0 of the LBA for any media access. This register is used by some non-media access commands to pass command specific information from the host to the device, or from the device to the host. This register shall be updated to reflect the media address of the error when a media access command is unsuccessfully completed.	0

Register 8-5: IDE_DEV_CYL_LOW

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_SEC_CYL_LOW	RW	If the LBS bit is cleared to zero in the Device/Head register, this register contains the low order bits of the starting cylinder address for any media access. If the LBA bit is set to one in the Device/Head register, this register contains Bits 15-8 of the LBA for any media access. This register shall be updated to reflect the address of the first error when a media access command is unsuccessfully completed.	0

Register 8-6: IDE_DEV_CYL_HIGH

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0

[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_SEC_CYL_HIGH	RW	If the LBA bit is cleared to zero in the Device/Head register, this register contains the high order bits of the starting cylinder address for any media access. If the LBA bit is set to one in the Device/Head register, this register contains Bits 23-16 of the LBA for any media access. This register shall be updated to reflect the address of the first error when a media access command is unsuccessfully completed.	0

Register 8-7: IDE_DEV_HEAD_DRV

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_SEC_HEAD_DRV	RW	This register selects the device, defines address translation as CHS or LBA, and provides the head address if CHS or LBA (27:24) if LBA.	0

Register 8-8: IDE_DEV_CMD_STAT

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 8]	DEV_CMD	RW	This register contains the command code being sent to the device. Command execution begins immediately after this register is written.	0
[7:0]	DEV_STAT	RW	This register contains the device status. The contents of this register are updated to reflect the current state of the device and the progress of any command being executed by the device. When the BSY bit is equal to zero, the other bits in this register are valid. When the BSY bit is equal to one, other bits in this register are not valid.	0

Register 8-9: IDE_DEV_ALT_STAT

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	N/A	Reserved	0
[24]	IOCS16	R	The IDE Device IOCS16_n state	0
[23:17]	Reserved	N/A	Reserved	0
[16]	IORDY	R	The IDE Device IORDY state	0
[15: 0]	DEV_SEC_ALT_STAT	RW	This register contains the same information as the Status register in the command block.	0

Register 8-10: IDE_HOST_STAT_FORCE

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	N/A	Reserved	0
[10]	Force_ADMA_status	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the ADMA interrupt status	0
[9]	Force_UDMA_overflow	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the UDMA mode FIFO overflow status	0
[8]	Force_UDMA_underflow	W	for software engineer debug purpose, only valid in debug	0

			mode(IDE_HOST_CON[0]), set the UDMA mode FIFO underflow status	
[7]	Force_UDMA_autoc_timeout	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the UDMA auto-continue timeout status	0
[6]	Force_UDMA_bsto_timeout	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the UDMA mode burst-out timeout status	0
[5]	Force_UDMA_bsti_timeout	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the UDMA mode burst-in timeout status	0
[4]	Force_PIO_IORDY_timeout	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the PIO mode IORDY timeout status	0
[3]	Force_DMA_finish	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the DMA finish status	0
[2]	Force_card_removal	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the card removal status	0
[1]	Force_card_insert	W	for software engineer debug purpose, only valid in debug mode(IDE_HOST_CON[0]), set the card insert status	0
[0]	Reserved	N/A	Reserved	0

Register 8-11: IDE_HOST_CON

Field	Symbol	Direction	Description	Default
[31]	IDEC_en	R/W	IDE Controller Enable, external device power up	0
[30]	UDMA_mode_en	R/W	IDE UDMA mode enable	0
[29]	Byte_Access_en	R/W	Byte access mode enable, only valid in PIO mode	0
[28]	Device_reset	R/W	Active high	0
[27]	UDMA_FIFO_reset	WO	UDMA mode FIFO reset	0
[26]	UDMA_HTBSTO_cnt_en	R/W	UDMA mode, Host Termination Burst-Out counter enable, only valid in debug mode(IDE_HOST_CON[0])	0
[25]	UDMA_HTBSTI_cnt_en	R/W	UDMA mode, Host Termination Burst-In counter enable, only valid in debug mode(IDE_HOST_CON[0])	0
[24]	UDMA_AUTOC_cnt_en	R/W	UDMA mode, device termination burst-in or burst-out in the middle of transfer counter enable, only valid in debug mode(IDE_HOST_CON[0])	0
[23]	UDMA_BSTI_HP_cnt_en	R/W	UDMA mode, host pause the burst-in clock cycle counter enable, only valid in debug mode(IDE_HOST_CON[0])	0
[22]	UDMA_BSTO_HP_cnt_en	R/W	UDMA mode, host pause the burst-out clock cycle counter enable, only valid in debug mode(IDE_HOST_CON[0])	0
[21]	UDMA_BSTO_DP_cnt_en	R/W	UDMA mode, device pause the burst-out clock cycle counter enable, only valid in debug mode(IDE_HOST_CON[0])	0
[20:4]	Undeifined	N/A	Reserved	0
[3]	Wait_DSTROBE_to_HIGH	R/W	UDMA mode, in Burst-in, host do NOT release the STOP to LOW until the DSTROBE is HIGH, recommended to 0	0
[2]	Inhibit_HSTROBE_STOP_LOW	R/W	UDMA mode, in Burst-out, host inhibit the HSTROBE stop at LOW, recommended to 0	0
[1]	ADMA_mode_en	R/W	AMDA mode enable, support the Descriptor	0
[0]	Debug_mode_en	R/W	Debug mode enable	0

Register 8-12: IDE_HOST_STAT_UNMASK

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	N/A	Reserved	0
[10]	ADMA_int_unmask	R/W	interrupt source unmask bit	0
[9]	UDMA_overflow_unmask	R/W	interrupt source unmask bit	0
[8]	UDMA_underflow_unmask	R/W	interrupt source unmask bit	0

[7]	UDMA_autoc_timeout_unmask	R/W	interrupt source unmask bit	0
[6]	UDMA_bsto_timeout_unmask	R/W	interrupt source unmask bit, recommended to 0, use bit7 instead	0
[5]	UDMA_bsti_timeout_unmask	R/W	interrupt source unmask bit, recommended to 0, use bit7 instead	0
[4]	PIO_IORDY_timeout_unmask	R/W	interrupt source unmask bit	0
[3]	DMA_finish_unmask	R/W	interrupt source unmask bit	0
[2]	card_removal_unmask	R/W	interrupt source unmask bit	0
[1]	card_insert_unmask	R/W	interrupt source unmask bit	0
[0]	DEVICE_INTRQ_unmask	R/W	interrupt source unmask bit, recommended to 1, and mute all the others	0

Register 8-13: IDE_HOST_STAT_EN

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	N/A	Reserved	0
[10]	ADMA_sts_en	R/W	interrupt source status enable	0
[9]	UDMA_overflow_sts_en	R/W	interrupt source status enable	0
[8]	UDMA_underflow_sts_en	R/W	interrupt source status enable	0
[7]	UDMA_autoc_timeout_sts_en	R/W	interrupt source status enable, function enable	0
[6]	UDMA_bsto_timeout_sts_en	R/W	interrupt source status enable, function enable	0
[5]	UDMA_bsti_timeout_sts_en	R/W	interrupt source status enable, function enable	0
[4]	PIO_IORDY_timeout_sts_en	R/W	interrupt source status enable, function enable	0
[3]	DMA_finish_sts_en	R/W	interrupt source status enable	0
[2]	card_removal_sts_en	R/W	interrupt source status enable	0
[1]	card_insert_sts_en	R/W	interrupt source status enable	0
[0]	Reserved	N/A	Reserved	0

Register 8-14: IDE_HOST_STAT

Field	Symbol	Direction	Description	Default
[31:28]	PIO_Message	R	please contact with infoTM for further information	0
[27:24]	UDMA_Message	R	please contact with infoTM for further information	0
[23:20]	ADMA_Message	R	please contact with infoTM for further information	0
[19:17]	Reserved	N/A	Reserved	0
[16]	Card insert	R	Card is in socket	0
[15:11]	Reserved	N/A	Reserved	0
[10]	ADMA_int_status	RW1C	interrupt source, write 1 to clear	0
[9]	UDMA_overflow_status	RW1C	interrupt source, write 1 to clear	0
[8]	UDMA_underflow_status	RW1C	interrupt source, write 1 to clear	0
[7]	UDMA_autoc_timeout_status	RW1C	interrupt source, write 1 to clear	0
[6]	UDMA_bsto_timeout_status	RW1C	interrupt source, write 1 to clear	0
[5]	UDMA_bsti_timeout_status	RW1C	interrupt source, write 1 to clear	0
[4]	PIO_IORDY_timeout_status	RW1C	interrupt source, write 1 to clear	0
[3]	DMA_finish_status	RW1C	Interrupt source, write 1 to clear, in ADMA mode, user should set the INT in the last descriptor line.	0
[2]	card_removal_status	RW1C	interrupt source, write 1 to clear	0
[1]	card_insert_status	RW1C	interrupt source, write 1 to clear	0
[0]	DEVICE_INTRQ_status	RAC	interrupt source, please clear the interrupt source from device	0

Register 8-15: IDE_PIN_STATUS

Field	Symbol	Direction	Description	Default
[31:16]	IDE_DATA_in	R	the current data value in the IDE data	0
[15]	IO_Grant	R	the reused data pad has granted for IDE data	0
[14]	IO_Req	R	the reused data pad has requested for IDE data	0
[13]	IDE_DATA_OE_n	R	the tri-state data pad output enable, active low	0

[12]	DMACK_n	R	the status of DMACK_n	0
[11]	IOWR_n/STOP	R	the status of IOWR_n in PIO mode, and the STOP in UDMA mode	0
[10]	IORD_n/HSTROBE/HDMARDY_n	R	the status of IORD_n inPIO mode, the HSTROBE in UDMA Burst-Out mode, and the HDMARDY_n in UDMA Burst-In mode	0
[9]	CS1_n	R	Chip Select 1 in PIO mode	0
[8]	CS0_n	R	Chip Select 0 in PIO mode	0
[7: 5]	IDE_ADDR	R	the current address value	0
[4]	IOCS16_n	R	the status of IOCS16_n, active low	0
[3]	CD1_n	R	the status of Card Dectect PIN 1, active low	0
[2]	IORDY/DSTROBE/DDMARDY_n	R	the status of IORDY in PIO mode, and the DSTROBE in UDMA Burst-In mode, and the DDMARDY_n in UDMA Burst-Out mode	0
[1]	INTRQ	R	the status of INTRQ from DEVICE	0
[0]	DMARQ	R	the status of DMARQ from DEVICE	0

Register 8-16: IDE_PIO_TPARAM0

Field	Symbol	Direction	Description	Default
[31: 0]	IORDY_TimeOut_Value	RW	the default timer value for IORDY in PIO mode	0

Register 8-17: IDE_ PIO_TPARAM1

Field	Symbol	Direction	Description	Default
[31:24]	Sampe_IORDY_Value	RW	the sample clock ticks for IORDY in PIO mode, recommended to 0	0
[23:16]	Addr_Setup_Value	RW	the address setup time in PIO mode, always be referred as "t1"	0
[15: 8]	Data_Setup_Value	RW	the IORD_n/IOWR_n setup time in PIO mode, always be referred as "t2(d)" for data access	0
[7: 0]	Regs_Setup_Value	RW	the IORD_n/IOWR_n setup time in PIO mode, always be referred as "t2(r)" for 8-bit register access	0

Register 8-18: IDE_ PIO_TPARAM2

Field	Symbol	Direction	Description	Default
[31:16]	Sampe_IORDY_Value	RW	the sample clock ticks for IORDY in PIO mode, recommended to 0	0
[15: 8]	Data_Recovery_Value	RW	the left time for data access, always be referred as "t0 - t1 - t2(d)"	0
[7: 0]	Regs_Recovery_Value	RW	the left time for regs access, always be referred as "t0 - t1 - t2(r)"	0

Register 8-19: IDE_ UDMA_TPARAM0

Field	Symbol	Direction	Description	Default
[31:30]	Reserved	N/A	Reserved	0
[29:24]	param_tACK	RW	the value for tACK timer, always 20ns/clock cycle	0
[23:22]	Reserved	N/A	Reserved	0
[21:16]	param_tENV	RW	the value for tENV timer	0
[15:14]	Reserved	N/A	Reserved	0
[13: 8]	param_tRP	RW	the value for tRP timer	0
[7:6]	Reserved	N/A	Reserved	0
[5:0]	param_tSS	RW	the value for tSS timer	0

Register 8-20: IDE_ UDMA_TPARAM1

Field	Symbol	Direction	Description	Default
[31:30]	Reserved	N/A	Reserved	0
[29:24]	param_tDVS0	RW	the value for tDVS timer, while HSTROBE is HIGH	0
[23:22]	Reserved	N/A	Reserved	0
[21:16]	param_tDVH0	RW	the value for tDVH timer, while HSTROBE is LOW	0
[15:14]	Reserved	N/A	Reserved	0
[13: 8]	param_tDVS1	RW	the value for tDVS timer, while HSTROBE is LOW	0

[7:6]	Reserved	N/A	Reserved	0
[5:0]	param_tDVH1	RW	the value for tDVH timer, while HSTROBE is HIGH	0

Register 8-21: IDE_ UDMA_TPARAM2

Field	Symbol	Direction	Description	Default
[31:14]	Reserved	N/A	Reserved	0
[13: 8]	param_tCVS	RW	the value for tCVS timer	0
[7:6]	Reserved	N/A	Reserved	0
[5:0]	param_tCVH	RW	the value for tCVH timer	0

Register 8-22: IDE_ UDMA_TPARAM3

Field	Symbol	Direction	Description	Default
[31:0]	DDMARDY_TimeOut_Value	RW	the default timer value for DDMARDY_n in UDMA Burst-Out mode	0

Register 8-23: IDE_ UDMA_TPARAM4

Field	Symbol	Direction	Description	Default
[31:0]	DSTROBE_TimeOut_Value	RW	the default timer value for DSTROBE in UDMA Burst-In mode	0

Register 8-24: IDE_ UDMA_TPARAM5

Field	Symbol	Direction	Description	Default
[31:0]	AUTOC_TimeOut_Value	RW	the default timer value for DEVICE Auto-Continue the Burst in UDMA mode	0

Register 8-25: IDE_UDMA_TRIGGER

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	UDMA_Start	RW	Write 1 to trigger the UDMA Burst, IDEC clear it when the Burst is over, please set the IDE_UDMA_XFER_INFO first	0

Register 8-26: IDE_UDMA_XFER_INFO

Field	Symbol	Direction	Description	Default
[31]	UDMA_Burst_In	RW	please set to 1, if IDEC read data from DEVICE, this bit should not be accessed during Burst	0
[30:0]	UDMA_Burst_Lenght	RW	please set the total byte number, these bits should not be accessed during Burst	0

Register 8-27: IDE_UDMA_FIFO_STATUS

Field	Symbol	Direction	Description	Default
[31:16]	UDMA_AUTOC_cnt	R	the numbers that the DEVICE auto-continue the burst	0
[15:8]	UDMA_FIFO_word_cnt	R	the word count in FIFO	0
[7:0]	UDMA_FIFO_free_cnt	R	the free space in FIFO	0

Register 8-28: IDE_UDMA_HTBST_CNT

Field	Symbol	Direction	Description	Default
[31:16]	UDMA_HTBSTI_cnt	R	the numbers that the IDEC terminates the UDMA Burst-In	0

[15:0]	UDMA_HTBSTO_cnt	R	the numbers that the IDEC terminates the UDMA Burst-Out	0

Register 8-29: IDE_UDMA_BSTI_HP_CNT

Field	Symbol	Direction	Description	Default
[31:0]	UDMA_BSTI_HP_cnt	R	the clock numbers that IDEC pause the UDMA Burst-In	0

Register 8-30: IDE_UDMA_BSTO_HP_CNT

Field	Symbol	Direction	Description	Default
[31:0]	UDMA_BSTO_HP_cnt	R	the clock numbers that IDEC pause the UDMA Burst-Out	0

Register 8-31: IDE_UDMA_BSTO_DP_CNT

Field	Symbol	Direction	Description	Default
[31:0]	UDMA_BSTO_DP_cnt	R	the clock numbers that DEVICE pause the UDMA Burst-Out	0

Register 8-32: IDE_SDMA_BUF0_ADDR

Field	Symbol	Direction	Description	Default
[31:0]	SDMA_ADDR	RW	the start address for DMA transfer, this register should not be accessed during AMDA working	0

Register 8-33: IDE_SDMA_BUF0_CTRL

Field	Symbol	Direction	Description	Default
[31]	SDMA_en	RWAC	setting to 1 will start the SDMA, clear itself after finish	0
[30]	BUF_rst	RWAC	setting to 1 will reset the control logic	0
[29:28]	BUF_indicator	R	indicate the current working BUF	0
[27]	ALT_en	RW	alternate change to next BUF	0
[26]	AUTO_en	RW	auto-reload enable	0
[25]	DIR	RW	transfer direction	0
[24: 0]	LEN	RW	Byte length	0

Register 8-34: IDE_SDMA_BUF1_ADDR

Field	Symbol	Direction	Description	Default
[31:0]	SDMA_ADDR	RW	the start address for DMA transfer, this register should not be accessed during AMDA working	0

Register 8-35: IDE_SDMA_BUF1_CTRL

Field	Symbol	Direction	Description	Default
[31]	SDMA_en	RWAC	setting to 1 will start the SDMA, clear itself after finish	0
[30]	BUF_rst	RWAC	setting to 1 will reset the control logic	0
[29:28]	BUF_indicator	R	indicate the current working BUF	0
[27]	ALT_en	RW	alternate change to next BUF	0
[26]	AUTO_en	RW	auto-reload enable	0
[25]	DIR	RW	transfer direction	0
[24: 0]	LEN	RW	Byte length	0

Register 8-36: IDE_SDMA_BUF2_ADDR

Field	Symbol	Direction	Description	Default
[31:0]	SDMA_ADDR	RW	the start address for DMA transfer, this register should not be accessed during AMDA working	0

Register 8-37: IDE_SDMA_BUF2_CTRL

Field	Symbol	Direction	Description	Default
[31]	SDMA_en	RWAC	setting to 1 will start the SDMA, clear itself after finish	0
[30]	BUF_rst	RWAC	setting to 1 will reset the control logic	0
[29:28]	BUF_indicator	R	indicate the current working BUF	0
[27]	ALT_en	RW	alternate change to next BUF	0
[26]	AUTO_en	RW	auto-reload enable	0
[25]	DIR	RW	transfer direction	0
[24: 0]	LEN	RW	Byte length	0

Register 8-38: IDE_SDMA_BUF3_ADDR

Field	Symbol	Direction	Description	Default
[31:0]	SDMA_ADDR	RW	the start address for DMA transfer, this register should not be accessed during AMDA working	0

Register 8-39: IDE_SDMA_BUF3_CTRL

Field	Symbol	Direction	Description	Default
[31]	SDMA_en	RWAC	setting to 1 will start the SDMA, clear itself after finish	0
[30]	BUF_rst	RWAC	setting to 1 will reset the control logic	0
[29:28]	BUF_indicator	R	indicate the current working BUF	0
[27]	ALT_en	RW	alternate change to next BUF	0
[26]	AUTO_en	RW	auto-reload enable	0
[25]	DIR	RW	transfer direction	0
[24: 0]	LEN	RW	Byte length	0

Register 8-40: IDE_ADMA_DSCPT_POINTER

Field	Symbol	Direction	Description	Default
[31:0]	ADMA_DESCRIPTOR_POINTER	R/W	Writing this register will start the ADMA, the current descriptor pointer will be read out. Please set the IDE_UDMA_XFER_INFO before setting this register, and do NOT access the IDE_SDMA_BUFx_ADDR and IDE_SDMA_BUFx_CTRL during the ADMA is working, please start the UDMA later after writing this register.	0

9 DMA Controller

9.1 Overview

DMA controller (GDMA) consists of 32 channels, 8 general channels and 24 special channels. General channel can transfer data from a source peripheral/memory to a destination peripheral/memory over one or more system BUS. Special channel is used to specific application for high performance.

Features

- 8 general channels and 24 special channels
- Support for memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral DMA transfers
- Programmable source and destination addresses with increment, decrement, or no change mode
- Programmable burst transfer size
- Multi-block transfers achieved through
- Support Scatter/Gather transfer mode
- Handshaking interfaces for source and destination peripherals

Block Diagram

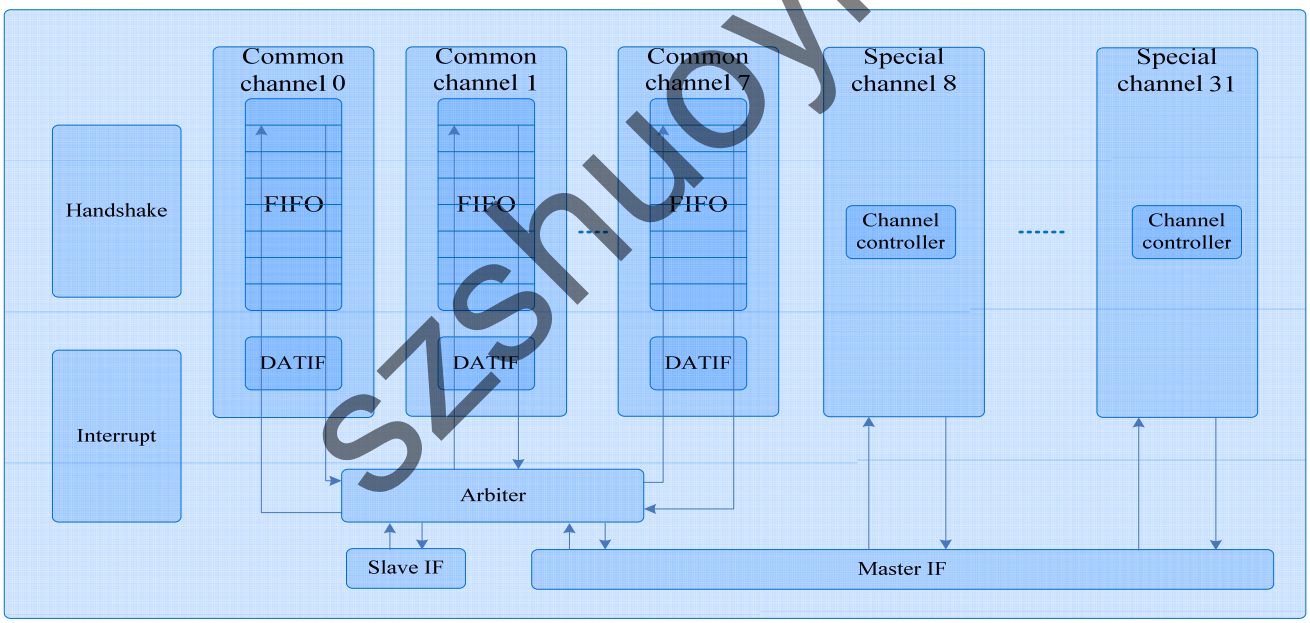


Figure 9-1: GDMA Block Diagram

Figure 9-1 shows the functional block diagram of GDMA, mainly include 32 channel modules, channel arbiter, master IF and Slave IF. One DMA transfer would use one channel and multi Master interface. Slave interface is used to configure the working register and get status of DMA channel. Arbiter gets the request signal from channel control state machine and arbitrates for the master bus interface. Handshake interface support the handshake operation between source and destination peripheral to control transferring a single and a burst transaction. This interface is used to request, acknowledge, and control a GDMA transaction. A channel can receive a request through one of three types of handshaking interface: hardware, software, or peripheral interrupt. Interrupt module control the interrupt generating, masking and clearing.

9.2 Functional Description

GDMA Transfers

GDMA support different transfer type and needed to be program corresponding register to determine whether single- or multi-block transfers occur, and which type of multi-block transfer is used.

Block Flow Controller and Transfer Type

The device that controls the length of a block is known as the flow controller. Either the source peripheral or the destination peripheral must be assigned as the flow controller.

- If the block size is known prior to when the channel is enabled, then the GDMA should be programmed as the flow controller. The block size should be programmed into the register field.
- If the block size is unknown when the GDMA channel is enabled, either the source or destination peripheral must be the flow controller.

Table 9-1: Transfer Types and Flow Control Combinations

Transfer Type	Flow controller
Memory to Memory	GDMA
Memory to Peripheral	GDMA
Memory to Peripheral	Peripheral
Peripheral to Memory	GDMA
Peripheral to Memory	Peripheral
Peripheral to Peripheral	GDMA
Peripheral to Peripheral	Source Peripheral
Peripheral to Peripheral	Destination Peripheral

Handshaking Interface

Handshaking interfaces are used at the transaction level to control the flow of single or burst transactions. The operation of the handshaking interface is different and depends on whether the peripheral or the GDMA is the flow controller.

The peripheral uses the handshaking interface to indicate to the GDMA that it is ready to transfer or accept data over the system data bus.

A non-memory peripheral can request a DMA transfer through the GDMA using one of two types of handshaking interfaces:

- Hardware
- Software

Software selects between the hardware or software handshaking interface on a per-channel basis. Software handshaking is accomplished through memory-mapped registers, while hardware handshaking is accomplished using a dedicated handshaking interface. There are 16 hard handshake interface and connect with different peripheral as following table.

Table 9-2: GDMA Hard Handshake Interface Connections

No	Description
0	UART0 RX Request
1	UART1 RX Request
2	UART2 RX Request
3	UART0 TX Request
4	UART1 TX Request
5	UART2 TX Request
6	IIS/AC97 TX Request
7	IIS/AC97 RX Request
8	UART3 RX Request
9	UART3 TX Request

10	SSI slave RX Request /SSI master1 RX Request
11	SSI slave TX Request /SSI master1 TX Request
12	SSI master0 RX Request
13	SSI master0 TX Request
14	PWM Request/SSI master2 RX Request
15	SSI master2 TX Request

Source and destination peripherals can independently select the handshaking interface type; that is, hardware or software handshaking.

Memory Peripherals

When GDMA is used for a memory peripheral, there is no handshaking interface with the GDMA, and therefore the memory peripheral can never be a flow controller. Once the channel is enabled, the transfer proceeds immediately without waiting for a transaction request.

The alternative to not having a transaction-level handshaking interface is to allow the GDMA to attempt BUS transfers to the peripheral once the channel is enabled. If the peripheral slave cannot accept these transfers, it inserts wait states onto the bus until it is ready; it is not recommended that more than 16 wait states be inserted onto the bus. By using the handshaking interface, the peripheral can signal to the GDMA that it is ready to transmit or receive data, and then the GDMA can access the peripheral without the peripheral inserting wait states onto the bus.

Software Handshaking

When the slave peripheral requires the GDMA to perform a DMA transaction, it communicates this request by sending an interrupt to the CPU or interrupt controller. The interrupt service routine then uses the software registers to initiate and control a DMA transaction. This group of software registers is used to implement the software handshaking interface.

The HS_SEL_SRC/HS_SEL_DST bit in the CFGx channel configuration register must be set to enable software handshaking.

1 Program and enable channel

2 After interrupt, initiate and control DMA transaction between peripheral(s) and GDMA.

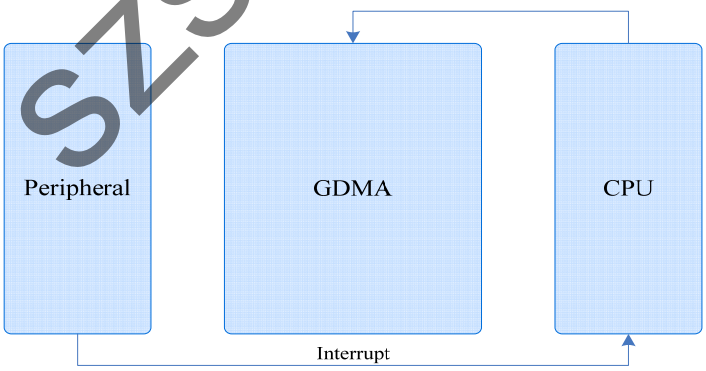


Figure 9-2: GDMA Software Control Diagram

The software handshaking registers are:

- ReqSrcReg – source software transaction request
- ReqDstReg – destination software transaction request

- SglReqSrcReg – single source transaction request
- SglReqDstReg – single destination transaction request
- LstSrcReg – last source transaction request
- LstDstReg – last destination transaction request

Handshaking Interface – Peripheral Is Not Flow Controller

When the peripheral is not the flow controller, the GDMA tries to efficiently transfer the data using as little of the bus bandwidth as possible. Generally, the GDMA tries to transfer the data using burst transactions and, where possible, fill or empty the channel FIFO in single bursts – provided that the software has not limited the burst length. The GDMA can also lock the arbitration for the master bus interface so that a channel is permanently granted the master bus interface.

Before describing the handshaking interface operation when the peripheral is not the flow controller, the following sections define the terms “Single Transaction Region” and “Early-Terminated Burst Transaction.”

Single Transaction Region

There are cases where a DMA block transfer cannot complete using only burst transactions. Typically this occurs when the block size is not a multiple of the burst transaction length. In these cases, the block transfer uses burst transactions up to the point where the amount of data left to complete the block is less than the amount of data in a burst transaction. At this point, the GDMA samples the “single” status flag and completes the block transfer using single transactions.

The peripheral asserts a single status flag to indicate to the GDMA that there is enough data or space to complete a single transaction from or to the source/destination peripheral.

The Single Transaction Region is the time interval where the GDMA uses single transactions to complete the block transfer; burst transactions are exclusively used outside this region.

The Single Transaction Region applies to only a peripheral that is not the flow controller. The precise definition of when this region is entered is dependent on what acts as the flow controller:

- The GDMA is the flow controller – The source peripheral enters the Single Transaction Region when the number of bytes left to complete in the source block transfer is less than src_burst_size_bytes. If:

  blk_size_bytes/src_burst_size_bytes = integer

  Then the source never enters this region, and the source block uses only burst transactions.

  The destination peripheral enters the Single Transaction Region when the number of bytes left to complete in the destination block transfer is less than dst_burst_size_bytes. If:

  blk_size_bytes/dst_burst_size_bytes = integer

  Then the destination never enters this region, and the destination block uses only burst transactions.

  The above conditions cause a peripheral to enter the Single Transaction Region. When the peripheral is outside the Single Transaction Region, then the GDMA responds to only burst transaction requests. Whether the peripheral knows that it is in the Single Transaction Region or not, it must always generate burst requests outside the Single Transaction Region, or the DMA block transfer stalls. Once in the Single Transaction Region, the GDMA can complete the block transfer using single transactions.

- Either the source or destination peripheral is the flow controller – The destination or source peripheral enters the Single Transaction Region when the flow control peripheral – that is, the source or destination – signals the last transaction in the block and when the amount of data left to be transferred in the destination/source block is less than that which is specified by dst_burst_size_bytes/src_burst_size_bytes.

Early-Terminated Burst Transaction

When a source or destination peripheral is in the Single Transaction Region, a burst transaction can still be requested.

However, src_burst_size_bytes/dst_burst_size_bytes is greater than the number of bytes left to complete in the source/destination block transfer at the time that the burst transaction is triggered. In this case, the burst transaction is started and "early-terminated" at block completion without transferring the programmed amount of data – that is, src_burst_size_bytes or dst_burst_size_bytes – but only the amount required completing the block transfer. An Early Terminated Burst Transaction occurs between the GDMA and the peripheral only when the peripheral is not the flow controller.

Hardware Handshaking – Peripheral Is Not Flow Controller

Figure 9-3 illustrates the hardware handshaking interface between a peripheral – whether a destination or source – and the GDMA when the peripheral is not the flow controller.

Hardware Handshaking can support burst transaction, back-to-back burst transaction, and single transaction, and early terminated burst transaction.

Software Handshaking – Peripheral Is Not Flow Controller

When the peripheral is not the flow controller, then the last transaction registers – LstSrcReg and LstDstReg – are not used, and the values in these registers are ignored.

Operation – Peripheral Not In Single Transaction Region

Writing a 1 to the ReqSrcReg[x]/ReqDstReg[x] register is always interpreted as a burst transaction request, where x is the channel number. However, in order for a burst transaction request to start, software must write a 1 to the SglReqSrcReg[x]/ SglReqDstReg[x] register.

You can write a 1 to the SglReqSrcReg[x]/SglReqDstReg[x] and ReqSrcReg[x]/ReqDstReg[x] registers in any order, but both registers must be asserted in order to initiate a burst transaction. Upon completion of the burst transaction, the hardware clears the SglReqSrcReg[x]/SglReqDstReg[x] and ReqSrcReg[x]/ReqDstReg[x] registers.

Operation – Peripheral in Single Transaction Region

Writing a 1 to the SglReqSrcReg/SglReqDstReg initiates a single transaction. Upon completion of the single transaction, both the SglReqSrcReg/SglReqDstReg and ReqSrcReg/ReqDstReg bits are cleared by hardware. Therefore, writing a 1 to the ReqSrcReg/ReqDstReg is ignored while a single transaction has been initiated, and the requested burst transaction is not serviced.

Again, writing a 1 to the ReqSrcReg/ReqDstReg register is always a burst transaction request. However, in order for a burst transaction request to start, the corresponding channel bit in the SglReqSrcReg/SglReqDstReg must be asserted. Therefore, to ensure that a burst transaction is serviced in this region, you must write a 1 to the ReqSrcReg/ReqDstReg before writing a 1 to the SglReqSrcReg/SglReqDstReg register. If the programming order is reversed, a single transaction is started instead of a burst transaction. The hardware clears both the ReqSrcReg/ReqDstReg and the SglReqSrcReg/SglReqDstReg registers after the burst transaction request completes. When a burst transaction is initiated in the Single Transaction Region, then the block completes using an Early-Terminated Burst Transaction.

Software can poll the relevant channel bit in the SglReqSrcReg/SglReqDstReg and ReqSrcReg/ReqDstReg registers. When both are 0, then either the requested burst or single transaction has completed. Alternatively, the IntSrcTran or IntDstTran interrupts can be enabled and unmasked in order to generate an interrupt when the requested source or destination transaction has completed.

Single Transactions – Peripheral Is Not Flow Controller

When the source peripheral is not the flow controller, software will never need to initiate single transactions from the source (software handshaking). This can happen if either of the following is true:

- Block size is a multiple of the burst transaction length.
  - ✧ If GDMA is the flow controller

    blk_size_bytes_dma/src_burst_size_bytes = integer
  - ✧ If the destination peripheral is the flow controller

blk_size_bytes_dst/src_burst_size_bytes = integer

- Block size is not a multiple of the burst transaction length, but the peripheral can dynamically adjust the watermark level that triggers a burst request in order to enable block completion. When the destination peripheral is not a flow controller, then the destination peripheral may hardcode dma_single to an inactive level (hardware handshaking), or software will never need to initiate single transactions to the destination (software handshaking). This can happen when any of the following are true:
- Block size is a multiple of the burst transaction length.
  - ✧ If GDMA is the flow controller

    block_size_bytes_dma/dst_burst_size_bytes = integer
  - ✧ If the source peripheral is flow controller

    block_size_bytes_src/dst_burst_size_bytes = integer
- The destination peripheral can dynamically adjust the watermark level upwards so that a burst request is triggered in order to enable a destination block completion.
- It is guaranteed that data at some point will be extracted from the destination FIFO in the “Single transaction region” in order to trigger a burst transaction

If none of the above is true, then a series of burst transactions followed by single transactions is needed to complete the source/destination block transfer

Hardware Interface – Peripheral Is Flow Controller

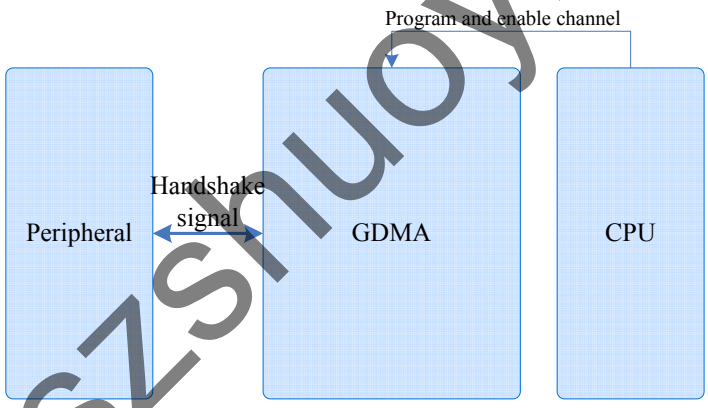


Figure 9-3: Hardware Handshaking Interface

Software Handshaking – Peripheral Is Flow Controller

Writing a 1 to the Source/Destination Software Transaction Request initiates a transaction; refer to “ReqSrcReg” and “ReqDstReg”, respectively. The type of transaction – single or burst – depends on the state of the corresponding channel bit in the Single Source/Destination Transaction Request register; refer to “SglReqSrcReg” or “SglReqDstReg”, respectively.

If SglReqSrcReg[n]/SglReqDstReg[n] = 1 when a 1 is written to the ReqSrcReg[n]/ReqDstReg[n] register, this means that software is requesting a single transaction on channel n, or a burst transaction otherwise.

The request is the last in the block if the corresponding channel bit in the Last Source/Destination Request register is asserted; refer to “LstSrcReg” and “LstDstReg”, respectively.

If LstSrcReg[n]/LstDstReg[n] = 1 when a 1 is written to the ReqSrcReg[n]/ReqDstReg[n] register, this means that software is requesting that this transaction is the last transaction in the block. The SglReqSrcReg/SglReqDstReg and

LstSrcReg/LstDstReg registers must be written to before the ReqSrcReg/ReqDstReg registers.

On completion of the transaction – single or burst – the relevant channel bit in the ReqSrcReg/ReqDstReg register is cleared by hardware. Software can therefore poll this bit in order to determine when the requested transaction has completed. Alternatively, the IntSrcTran or IntDstTran interrupts can be enabled and unmasked in order to generate an interrupt when the requested transaction – single or burst – has completed.

When the peripheral is the flow controller and the block size is not a multiple of the CTLx.SRC_MSIZE/CTLx.DEST_MSIZE, then software must use single transactions to complete the block transfer.

Single Transactions – Peripheral is Flow Controller

When the source peripheral is the flow controller, software will never need to initiate single transactions from the source (software handshaking). This occurs when:

block_size_bytes_src/src_burst_size_bytes = integer

When the destination peripheral is the flow controller, software will never need to initiate single transactions to the destination (software handshaking) when:

block_size_bytes_dst/dst_burst_size_bytes = integer

Flow Control Configurations

For hardware handshake transfer type, there are different flow control types according to different peripheral.

If source/destination is memory or peripheral without flow control function, flow control is performed by GDMA. If source/destination is peripheral with flow control function, flow control can be performed by either GDMA or peripheral.

The flow is same for software handshake transfer type. It is just using software register instead of signals to handshake.

Generating Requests for the Master Bus Interface

Each channel has a source state machine and destination state machine running in parallel. These state machines generate the request inputs to the arbiter, which arbitrates for the master bus interface (one arbiter per master bus interface). When the source/destination state machine is granted control of the master bus interface, and when the master bus interface is granted control of the external system bus, then BUS transfers between the peripheral and the GDMA(on behalf of the granted state machine) can take place. BUS transfers from the source peripheral or to the destination peripheral cannot proceed until the channel FIFO is ready. For burst transaction requests and for transfers involving memory peripherals, the criterion for “FIFO readiness” is controlled by the FIFO_MODE field of the CFGx register.

The channel FIFO is deemed ready when the space/data available is sufficient to complete a single transfer of the specified transfer width. FIFO readiness for source transfers occurs when the channel FIFO contains enough room to accept at least a single transfer of CTLx.SRC_TR_WIDTH width. FIFO readiness for destination transfers occurs when the channel FIFO contains data to form at least a single transfer of CTLx.DST_TR_WIDTH width.

When the source/destination peripheral is not memory, the source/destination state machine waits for a single/burst transaction request. Upon receipt of a transaction request and only if the channel FIFO is “ready” for source/destination BUS transfers; a request for the master bus interface is made by the source/destination state machine.

Scatter/Gather

Scatter is relevant to a destination transfer. The destination address is incremented or decremented by a programmed amount – the scatter increment – when a scatter boundary is reached. Figure 9-4 shows an example destination scatter transfer. The destination address is incremented or decremented by the value stored in the destination scatter increment (DSRx.DSI) field, multiplied by the number of bytes in a single transfer to the destination when a scatter boundary is reached. The number of destination transfers between successive scatter boundaries is programmed into the Destination Scatter Count (DSC) field of the DSRx register.

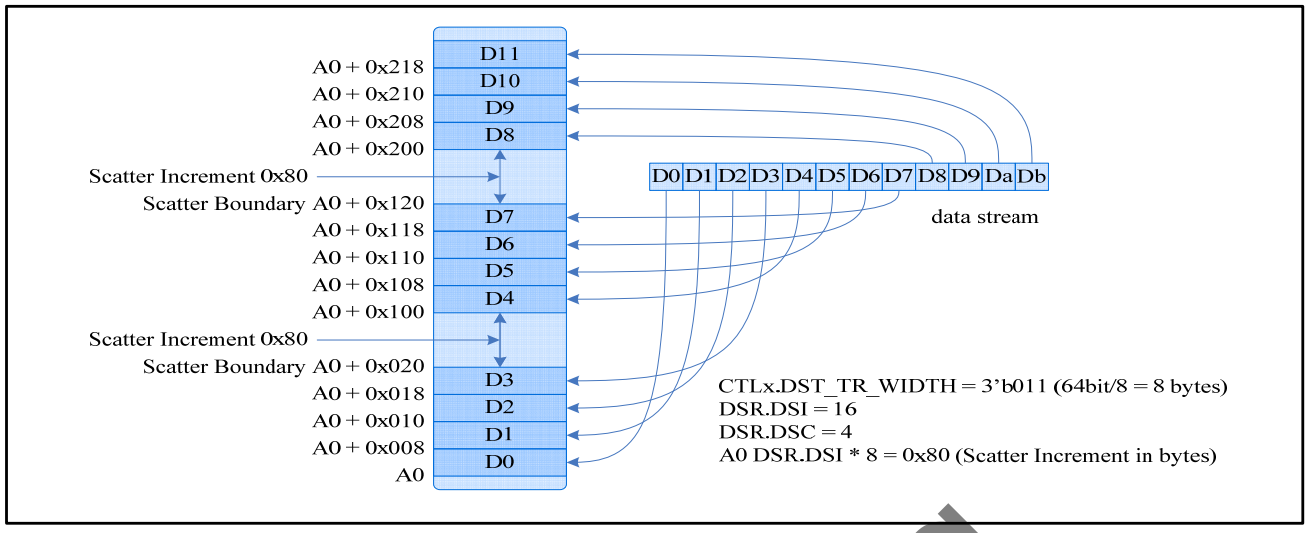


Figure 9-4: Example of a Destination Scatter Transfer

Scatter is enabled by writing a 1 to the CTLx.DST_SCATTER_EN field. The CTLx.DINC field determines if the address is incremented, decremented, or remains fixed when a scatter boundary is reached. If the CTLx.DINC field indicates a fixed-address control throughout a DMA transfer, then the CTLx.DST_SCATTER_EN field is ignored, and the scatter feature is automatically disabled.

Gather is relevant to a source transfer. The source address is incremented or decremented by a programmed amount when a gather boundary is reached. The number of source transfers between successive gather boundaries is programmed into the Source Gather Count (SGRx.SGC) field. The source address is incremented or decremented by the value stored in the source gather increment (SGRx.SGI) field, multiplied by the number of bytes in a single transfer from the source when a gather boundary is reached.

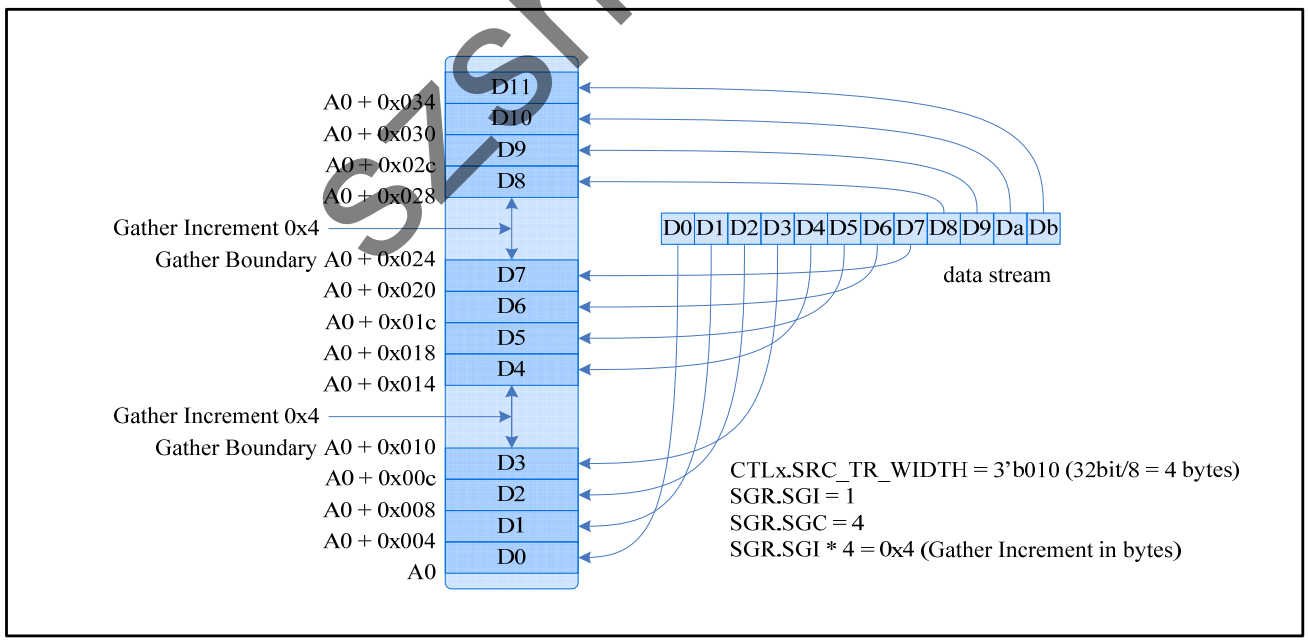


Figure 9-5: Example of a Source Gather Transfer

Gather is enabled by writing a 1 to the CTLx.SRC_GATHER_EN field. The CTLx.SINC field determines if the address is incremented, decremented, or remains fixed when a gather boundary is reached. If the CTLx.SINC field indicates a fixed-address control throughout a DMA transfer, then the CTLx.SRC_GATHER_EN field is ignored, and the gather feature is automatically disabled.

9.3 GDMA Register Description

9.3.1 GDMA Register Memory Map

GDMA's register mapping address range in system is 0x20C40000~0x20C4FFFF. GDMA register's system accessing address equal to BASE_ADDRESS (0x20C40000) plus address offset.

R/W *= Read/Write.*
R *= Read Only.*
W *= Write Only.*
R/W+ *= Read/Write, where one or more bits of the parameter have additional functionality and require special handling.*

Table 9-3: GDMA Register Memory Map

Address	Symbol	Direction	Description
0x000	SAR0	R/W	Channel 0 Source Address Register
0x008	DAR0	R/W	Channel 0 Destination Address Register
0x010	LLP0	R/W	Channel 0 Linked List Pointer Register
0x018	CTL0	R/W	Channel 0 Control Register low 32 bit
0x01c	CTL0_h	R/W	Channel 0 Control Register high 32 bit
0x020	SSTAT0	R/W	Channel 0 Source Status Register
0x028	DSTAT0	R/W	Channel 0 Destination Status Register
0x030	SSTATAR0	R/W	Channel 0 Source Status Address Register
0x038	DSTATAR0	R/W	Channel 0 Destination Status Address Register
0x040	CFG0	R/W	Channel 0 Configuration Register low 32 bit
0x044	CFG0_h	R/W	Channel 0 Configuration Register high 32 bit
0x048	SGR0	R/W	Channel 0 Source Gather Register
0x050	DSR0	R/W	Channel 0 Destination Scatter Register
0x058	SAR1	R/W	Channel 1 Source Address Register
0x060	DAR1	R/W	Channel 1 Destination Address Register
0x068	LLP1	R/W	Channel 1 Linked List Pointer Register
0x070	CTL1	R/W	Channel 1 Control Register low 32 bit
0x074	CTL1_h	R/W	Channel 1 Control Register high 32 bit
0x078	SSTAT1	R/W	Channel 1 Source Status Register
0x080	DSTAT1	R/W	Channel 1 Destination Status Register
0x088	SSTATAR1	R/W	Channel 1 Source Status Address Register
0x090	DSTATAR1	R/W	Channel 1 Destination Status Address Register
0x098	CFG1	R/W	Channel 1 Configuration Register low 32 bit
0x09c	CFG1_h	R/W	Channel 1 Configuration Register high 32 bit
0x0a0	SGR1	R/W	Channel 1 Source Gather Register
0x0a8	DSR1	R/W	Channel 1 Destination Scatter Register
0x0b0	SAR2	R/W	Channel 2 Source Address Register
0x0b8	DAR2	R/W	Channel 2 Destination Address Register
0x0c0	LLP2	R/W	Channel 2 Linked List Pointer Register
0x0c8	CTL2	R/W	Channel 2 Control Register low 32 bit
0x0cc	CTL2_h	R/W	Channel 2 Control Register high 32 bit
0x0d0	SSTAT2	R/W	Channel 2 Source Status Register
0x0d8	DSTAT2	R/W	Channel 2 Destination Status Register
0x0e0	SSTATAR2	R/W	Channel 2 Source Status Address Register
0x0e8	DSTATAR2	R/W	Channel 2 Destination Status Address Register
0x0f0	CFG2	R/W	Channel 2 Configuration Register low 32 bit
0x0f4	CFG2_h	R/W	Channel 2 Configuration Register high 32 bit
0x0f8	SGR2	R/W	Channel 2 Source Gather Register
0x100	DSR2	R/W	Channel 2 Destination Scatter Register

0x108	SAR3	R/W	Channel 3 Source Address Register
0x110	DAR3	R/W	Channel 3 Destination Address Register
0x118	LLP3	R/W	Channel 3 Linked List Pointer Register
0x120	CTL3	R/W	Channel 3 Control Register low 32 bit
0x124	CTL3_h	R/W	Channel 3 Control Register high 32 bit
0x128	SSTAT3	R/W	Channel 3 Source Status Register
0x130	DSTAT3	R/W	Channel 3 Destination Status Register
0x138	SSTATAR3	R/W	Channel 3 Source Status Address Register
0x140	DSTATAR3	R/W	Channel 3 Destination Status Address Register
0x148	CFG3	R/W	Channel 3 Configuration Register low 32 bit
0x14c	CFG3_h	R/W	Channel 3 Configuration Register high 32 bit
0x150	SGR3	R/W	Channel 3 Source Gather Register
0x158	DSR3	R/W	Channel 3 Destination Scatter Register
0x160	SAR4	R/W	Channel 4 Source Address Register
0x168	DAR4	R/W	Channel 4 Destination Address Register
0x170	LLP4	R/W	Channel 4 Linked List Pointer Register
0x178	CTL4	R/W	Channel 4 Control Register low 32 bit
0x17c	CTL4_h	R/W	Channel 4 Control Register high 32 bit
0x180	SSTAT4	R/W	Channel 4 Source Status Register
0x188	DSTAT4	R/W	Channel 4 Destination Status Register
0x190	SSTATAR4	R/W	Channel 4 Source Status Address Register
0x198	DSTATAR4	R/W	Channel 4 Destination Status Address Register
0x1a0	CFG4	R/W	Channel 4 Configuration Register low 32 bit
0x1a4	CFG4_h	R/W	Channel 4 Configuration Register high 32 bit
0x1a8	SGR4	R/W	Channel 4 Source Gather Register
0x1b0	DSR4	R/W	Channel 4 Destination Scatter Register
0x1b8	SAR5	R/W	Channel 5 Source Address Register
0x1c0	DAR5	R/W	Channel 5 Destination Address Register
0x1c8	LLP5	R/W	Channel 5 Linked List Pointer Register
0x1d0	CTL5	R/W	Channel 5 Control Register low 32 bit
0x1d4	CTL5_h	R/W	Channel 5 Control Register high 32 bit
0x1d8	SSTAT5	R/W	Channel 5 Source Status Register
0x1e0	DSTAT5	R/W	Channel 5 Destination Status Register
0x1e8	SSTATAR5	R/W	Channel 5 Source Status Address Register
0x1f0	DSTATAR5	R/W	Channel 5 Destination Status Address Register
0x1f8	CFG5	R/W	Channel 5 Configuration Register low 32 bit
0x1fc	CFG5_h	R/W	Channel 5 Configuration Register high 32 bit
0x200	SGR5	R/W	Channel 5 Source Gather Register
0x208	DSR5	R/W	Channel 5 Destination Scatter Register
0x210	SAR6	R/W	Channel 6 Source Address Register
0x218	DAR6	R/W	Channel 6 Destination Address Register
0x220	LLP6	R/W	Channel 6 Linked List Pointer Register
0x228	CTL6	R/W	Channel 6 Control Register low 32 bit
0x22c	CTL6_h	R/W	Channel 6 Control Register high 32 bit
0x230	SSTAT6	R/W	Channel 6 Source Status Register
0x238	DSTAT6	R/W	Channel 6 Destination Status Register
0x240	SSTATAR6	R/W	Channel 6 Source Status Address Register
0x248	DSTATAR6	R/W	Channel 6 Destination Status Address Register
0x250	CFG6	R/W	Channel 6 Configuration Register low 32 bit
0x254	CFG6_h	R/W	Channel 6 Configuration Register high 32 bit
0x258	SGR6	R/W	Channel 6 Source Gather Register
0x260	DSR6	R/W	Channel 6 Destination Scatter Register
0x268	SAR7	R/W	Channel 7 Source Address Register
0x270	DAR7	R/W	Channel 7 Destination Address Register
0x288	CTL7	R/W	Channel 7 Control Register low 32 bit
0x28c	CTL7_h	R/W	Channel 7 Control Register high 32 bit
0x270	LLP7	R/W	Channel 7 Linked List Pointer Register

0x288	SSTAT7	R/W	Channel 7 Source Status Register
0x290	DSTAT7	R/W	Channel 7 Destination Status Register
0x298	SSTATAR7	R/W	Channel 7 Source Status Address Register
0x2a0	DSTATAR7	R/W	Channel 7 Destination Status Address Register
0x2a8	CFG7	R/W	Channel 7 Configuration Register low 32 bit
0x2ac	CFG7_h	R/W	Channel 7 Configuration Register high 32 bit
0x2b0	SGR7	R/W	Channel 7 Source Gather Register
0x2b8	DSR7	R/W	Channel 7 Destination Scatter Register
0x2c0	RawTfr	R	Raw Status for IntTfr Interrupt
0x2c8	RawBlock	R	Raw Status for IntBlock Interrupt
0x2d0	RawSrcTran	R	Raw Status for IntSrcTran Interrupt
0x2d8	RawDstTran	R	Raw Status for IntDstTran Interrupt
0x2e0	RawErr	R	Raw Status for IntErr Interrupt
0x2e8	StatusTfr	R	Status for IntTfr Interrupt
0x2f0	StatusBlock	R	Status for IntBlock Interrupt
0x2f8	StatusSrcTran	R	Status for IntSrcTran Interrupt
0x300	StatusDstTran	R	Status for IntDstTran Interrupt
0x308	StatusErr	R	Status for IntErr Interrupt
0x310	MaskTfr	R/W	Mask for IntTfr Interrupt
0x314	MaskTfr_h	R/W	Mask for IntTfr Interrupt
0x318	MaskBlock	R/W	Mask for IntBlock Interrupt
0x320	MaskSrcTran	R/W	Mask for IntSrcTran Interrupt
0x328	MaskDstTran	R/W	Mask for IntDstTran Interrupt
0x330	MaskErr	R/W	Mask for IntErr Interrupt
0x338	ClearTfr	W	Clear for IntTfr Interrupt
0x340	ClearBlock	W	Clear for IntBlock Interrupt
0x348	ClearSrcTran	W	Clear for IntSrcTran Interrupt
0x350	ClearDstTran	W	Clear for IntDstTran Interrupt
0x358	ClearErr	W	Clear for IntErr Interrupt
0x360	StatusInt	W	Status for each interrupt type
0x368	ReqSrcReg	R/W	Source Software Transaction Request Register
0x370	ReqDstReg	R/W	Destination Software Transaction Request Register
0x378	SglReqSrcReg	R/W	Single Source Transaction Request Register
0x380	SglReqDstReg	R/W	Single Destination Transaction Request Register
0x388	LstSrcReg	R/W	Last Source Transaction Request Register
0x390	LstDstReg	R/W	Last Destination Transaction Request Register
0x398	DmaCfgReg	R/W	DMA Configuration Register
0x3a0	ChEnReg	R/W	DMA Channel Enable Register

9.3.2 GDMA Registers and Field Descriptions

Register 9-1: GDMA Configuration Registers (DmaCfgReg)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0x0
[0]	DMA_en	RW	**GDMA enable bit** ● 0: GDMA Disable ● 1: GDMA Enabled	0x0

Register 9-2: GDMA Channel Enable Registers (ChEnReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	CH_EN_WE	W	**Channel enable write enable**	0x0
[7:0]	CH_EN	RW	**Enables/Disables the channel**. Setting this bit enables a channel; clearing this bit disables the channel. ● 0: Disable the Channel ● 1: Enable the Channel	0x0

			The ChEnReg.CH_EN bit is automatically cleared by hardware to disable the channel after the last BUS transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.	

Register 9-3: Source Address Register for Channel x (SARx)

Field	Symbol	Direction	Description	Default
[31:0]	SAR	RW	**Current Source Address of DMA transfer**. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.	0x0

Register 9-4: Destination Address Register for Channel x (DARx)

Field	Symbol	Direction	Description	Default
[31:0]	DAR	RW	**Current Destination address of DMA transfer.** Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.	0x0

Register 9-5: Linked List Pointer Register for Channel x (LLPx)

Field	Symbol	Direction	Description	Default
[31:2]	LOC	RW	**Starting Address In Memory of next LLI if block chaining is enabled.** Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary. LLI accesses are always 32-bit accesses and cannot be changed or programmed to anything other than 32-bit.	0x0
[1:0]	LMS	R	**List Master Select.** Identifies the BUS interface where the memory device that stores the next linked list item resides. ● 00: master 1 ● 01: master 2 ● 10: insignificant ● 11: insignificant	0x0

Register 9-6: Control Register for Channel x (CTLx)

Field	Symbol	Direction	Description	Default
[63:45]	Reserved	N/A	Reserved	0x0
[44]	DONE	RW	**Done bit** If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel. LLI accesses are always 32-bit accesses and cannot be changed or programmed to anything other than 32-bit.	0x0
[43:32]	BLOCK_TS	RW	**Block Transfer Size.** When the GDMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single beat.	0x2

			Width: The width of the single transaction is determined by CTLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what the flow controller is. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at 4095, but the actual block size can be greater.	
[31:29]	Reserved	N/A	Reserved	0x0
[28]	LLP_SRC_EN	RW	Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.	0x0
[27]	LLP_DST_EN	RW	Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.	0x0
[26:25]	SMS	RW	**Source Master Select.** Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed. ● 00: master 1 ● 01: master 2 ● 10: insignificant ● 11: insignificant	0x0
[24:23]	DMS	RW	**Destination Master Select.** Identifies the Master Interface layer from which the destination device (peripheral or memory) is accessed. ● 00: master 1 ● 01: master 2 ● 10: insignificant ● 11: insignificant	0x0
[22:20]	TT_FC	RW	**Transfer Type and Flow Control.** The following transfer types are supported. ● 000: Memory to Memory, flow controller is GDMA ● 001: Memory to Peripheral, flow controller is GDMA ● 010: Peripheral to Memory, flow controller is GDMA ● 011: Peripheral to Peripheral, flow controller is GDMA ● 100: Peripheral to Memory, flow controller is Peripheral ● 101: Peripheral to Peripheral, flow controller is Source Peripheral ● 110: Memory to Peripheral, flow controller is Peripheral ● 111: Peripheral to Peripheral, flow controller is Destination Peripheral Flow Control can be assigned to the GDMA, the source peripheral, or the destination peripheral.	0x3
[19]	Reserved	N/A	Reserved	0x0
[18]	DST_SCATTER_EN	RW	**Destination scatter enable bit**: ● 0: Scatter disabled ● 1: Scatter enabled Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.	0x0
[17]	SRC_GATHER_EN	RW	**Source gather enable bit**: ● 0: gather disabled ● 1: gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.	0x0
[16:14]	SRC_SIZE	RW	**Source Burst Transaction Length.** Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. ● 000: Number of data items to be transferred 1 ● 001: Number of data items to be transferred 4 ● 010: Number of data items to be transferred 8 ● 011: Number of data items to be transferred 16 ● 100: Number of data items to be transferred 32 ● 101: Number of data items to be transferred 64	0x1

			● 110: Number of data items to be transferred 128 ● 111: Number of data items to be transferred 256	
[13:11]	DEST_SIZE	RW	**Destination Burst Transaction Length.** Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. ● 000: Number of data items to be transferred 1 ● 001: Number of data items to be transferred 4 ● 010: Number of data items to be transferred 8 ● 011: Number of data items to be transferred 16 ● 100: Number of data items to be transferred 32 ● 101: Number of data items to be transferred 64 ● 110: Number of data items to be transferred 128 ● 111: Number of data items to be transferred 2566	0x1
[10:9]	SINC	RW	**Source Address Increment.** Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change." ● 00: Increment ● 01: Decrement ● 1x = No change	0x0
[8:7]	DINC	RW	**Destination Address Increment.** Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change." ● 00: Increment ● 01: Decrement ● 1x = No change	0x0
[6:4]	SRC_TR_WIDTH	RW	**Source Transfer Width.** Mapped to BUS transfer size. For a non-memory peripheral, typically the peripheral (source) FIFO width. ● 000: size is 8 ● 001: size is 16 ● 010: size is 32 ● 011: size is 64 ● 100: size is 128 ● 101: size is 256 ● 11x = size is 256	0x0
[3:1]	DST_TR_WIDTH	RW	**Destination Transfer Width.** Mapped to BUS transfer size. For a non-memory peripheral, typically the peripheral (destination) FIFO width. ● 000: size is 8 ● 001: size is 16 ● 010: size is 32 ● 011: size is 64 ● 100: size is 128 ● 101: size is 256 ● 11x = size is 256	0x0
0	INT_EN	RW	**Interrupt Enable Bit**. If set, then all interrupt-generating sources are enabled.	0x1

Register 9-7: Source Status Register for Channel x (SSTATx)

Field	Symbol	Direction	Description	Default
[31:0]	SSTAT	RW	Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.	0x0

Register 9-8: Destination Status Register for Channel x (DSTATx)

Field	Symbol	Direction	Description	Default
[31:0]	DSTAT	RW	Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.	0x0

Register 9-9: Source Status Address Register for Channel x (SSTATARx)

Field	Symbol	Direction	Description	Default
[31:0]	SSTATAR	RW	Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.	0x0

Register 9-10: Destination Status Address Register for Channel x (DSTATARx)

Field	Symbol	Direction	Description	Default
[31:0]	DSTATAR	RW	Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.	0x0

Register 9-11: Configuration Register for Channel x (CFGx)

Field	Symbol	Direction	Description	Default
[63:47]	Reserved	N/A	Reserved	0x0
[46:43]	DEST_PER	RW	Assigns a hardware handshaking interface (0 - 15) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	0x0
[42:39]	SRC_PER	RW	Assigns a hardware handshaking interface (0 - 15) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.	0x2
[38]	SS_UPD_EN	RW	**Source Status Update Enable.** Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high.	0x0
[37]	DS_UPD_EN	RW	**Destination Status Update Enable.** Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.	0x0
[36:34]	Reserved	N/A	Reserved	0x1
[33]	FIFO_MODE	RW	**FIFO Mode Select.** Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced. ● 0: Space/data available for single transfer of the specified transfer width. ● 1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers. The exceptions are at the end of a burst transaction request or at the end of a block transfer.	0x0
[32]	FCMODE	RW	**Flow Control Mode.** Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.	0x0

			● 0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled. ● 1: Source transaction requests are not serviced until a destination transaction request occurs. In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.	
[31]	RELOAD_DST	RW	**Automatic Destination Reload.** The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.	0x0
[30]	RELOAD_SRC	RW	**Automatic Source Reload.** The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.	0x0
[29:20]	MAX_ABRST	RW	**Maximum BUS Burst Length.** Maximum BUS burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum BUS burst length for DMA transfers on this channel.	0x0
[19]	SRC_HS_POL	RW	**Source Handshaking Interface Polarity**. ● 0: Active high ● 1: Active low	0x0
[18]	DST_HS_POL	RW	**Destination Handshaking Interface Polarity**. ● 0: Active high ● 1: Active low	0x0
[17:12]	Reserved	N/A	Reserved	0x0
[11]	HS_SEL_SRC	RW	**Source Software or Hardware Handshaking Select**. This register selects which of the handshaking interfaces – hardware or software – is active for source requests on this channel. ● 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. ● 1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the source peripheral is memory, then this bit is ignored.	0x0
[10]	HS_SEL_DST	RW	**Destination Software or Hardware Handshaking Select**. This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel. ● 0: Hardware handshaking interface. Software-initiated transaction requests are ignored. ● 1: Software handshaking interface. Hardware- initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.	0x0
[9]	FIFO_EMPTY	R	Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFG*x*.CH_SUSP to cleanly disable a channel. ● 1: Channel FIFO empty ● 0: Channel FIFO not empty	0x0
8	CH_SUSP	RW	**Channel Suspend**. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFG*x*.FIFO_EMPTY to cleanly disable a channel without losing any data. ● 0: Not suspended. ● 1: Suspend DMA transfer from the source.	0x0
7:5	CH_PRIOR	RW	**Channel priority**. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: 7 A programmed value outside this range will cause erroneous behavior.	Chan Num.
4:0	Reserved	N/A	**Reserved**	0x0

Register 9-12: Source Gather Register for Channel x (SGRx)

Field	Symbol	Direction	Description	Default
[31:20]	SGC	RW	**Source Gather Count.** Source contiguous transfer count between successive gather boundaries.	0x0
19:0	SGI	RW	**Source Gather Interval.**	0x0

Register 9-13: Destination Scatter Register for Channel x (DSRx)

Field	Symbol	Direction	Description	Default
[31:20]	DSC	RW	**Destination Scatter Count.** Destination contiguous transfer count between successive scatter boundaries.	0x0
[19:0]	DSI	RW	**Destination Scatter Interval.**	0x0

Register 9-14: Block Transfer Complete Interrupt Raw Status Registers (RawBlock)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	RAW	RW	Block Transfer Complete Interrupt Raw Status for channel7 to channel 0	0x0

Register 9-15: Destination Transaction Complete Interrupt Raw Status Registers (RawDstTran)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	RAW	RW	Destination Transaction Complete Interrupt Raw Status for channel7 to channel 0	0x0

Register 9-16: Error Interrupt Raw Status Registers (RawErr)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	RAW	RW	Error Interrupt Raw Status for channel7 to channel 0	0x0

Register 9-17: Source Transaction Complete Interrupt Raw Status Registers (RawSrcTran)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	RAW	RW	Source Transaction Complete Interrupt Raw Status for channel7 to channel 0	0x0

Register 9-18: DMA Transfer Complete Interrupt Raw Status Registers (RawTfr)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	RAW	RW	DMA Transfer Complete Interrupt Raw Status for channel7 to channel 0	0x0

Register 9-19: Block Transfer Complete Interrupt Status Registers (StatusBlock)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	STATUS	RW	Block Transfer Complete Interrupt Status for channel7 to channel 0	0x0

Register 9-20: Destination Transaction Complete Interrupt Status Registers (StatusDstTran)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	STATUS	RW	Destination Transaction Complete Interrupt Status for channel7 to channel 0	0x0

Register 9-21: Error Interrupt Status Registers (StatusErr)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	STATUS	RW	Error Interrupt Status for channel7 to channel 0	0x0

Register 9-22: Source Transaction Complete Interrupt Status Registers (StatusSrcTran)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	STATUS	RW	Source Transaction Complete Interrupt Status for channel7 to channel 0	0x0

Register 9-23: DMA Transfer Complete Interrupt Status Registers (StatusTfr)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	STATUS	RW	DMA Transfer Complete Interrupt Status for channel7 to channel 0	0x0

Register 9-24: Block Transfer Complete Interrupt Mask Registers (MaskBlock)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	INT_MASK_WE	RW	Block Transfer Complete Interrupt Mask Write Enable for channel7 to channel 0 ● 0: write disabled ● 1: write enabled	0x0
[7:0]	INT_MASK	RW	Block Transfer Complete Interrupt Mask for channel7 to channel 0 ● 0: masked ● 1: unmasked	0x0

Register 9-25: Destination Transaction Complete Interrupt Mask Registers (MaskDstTran)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	INT_MASK_WE	RW	Destination Transaction Complete Interrupt Mask Write Enable for channel7 to channel 0 ● 0: write disabled ● 1: write enabled	0x0
[7:0]	INT_MASK	RW	Destination Transaction Complete Interrupt Mask for channel7 to channel 0 ● 0: masked ● 1: unmasked	0x0

Register 9-26: Error Interrupt Status Registers (MaskErr)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	INT_MASK_WE	RW	Error Interrupt Mask Write Enable for channel7 to channel 0 ● 0: write disabled ● 1: write enabled	0x0
[7:0]	INT_MASK	RW	Error Interrupt Mask for channel7 to channel 0 ● 0: masked ● 1: unmasked	0x0

Register 9-27: Source Transaction Complete Interrupt Status Registers (MaskSrcTran)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	INT_MASK_WE	RW	Source Transaction Complete Interrupt Mask Write Enable for channel7 to channel 0 ● 0: write disabled ● 1: write enabled	0x0
[7:0]	INT_MASK	RW	Source Transaction Complete Interrupt Mask for channel7 to channel 0 ● 0: masked ● 1: unmasked	0x0

Register 9-28: DMA Transfer Complete Interrupt Status Registers (MaskTfr)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	INT_MASK_WE	RW	DMA Transfer Complete Interrupt Mask Write Enable for channel7 to channel 0 ● 0: write disabled ● 1: write enabled	0x0
[7:0]	INT_MASK	RW	DMA Transfer Complete Interrupt Mask for channel7 to channel 0 ● 0: masked ● 1: unmasked	0x0

Register 9-29: Block Transfer Complete Interrupt Clear Registers (ClearBlock)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	CLEAR	RW	Block Transfer Complete Interrupt Clear for channel7 to channel 0 ● 0: no effect ● 1: clear interrupt	0x0

Register 9-30: Destination Transaction Complete Interrupt Clear Registers (ClearDstTran)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	CLEAR	RW	Destination Transaction Complete Interrupt Clear for channel7 to channel 0 ● 0: no effect ● 1: clear interrupt	0x0

Register 9-31: Error Interrupt Clear Registers (ClearErr)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0

[7:0]	CLEAR	RW	Error Interrupt Clear for channel7 to channel 0 ● 0: no effect ● 1: clear interrupt	0x0

Register 9-32: Source Transaction Complete Interrupt Clear Registers (ClearSrcTran)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	CLEAR	RW	Source Transaction Complete Interrupt Clear for channel7 to channel 0 ● 0: no effect ● 1: clear interrupt	0x0

Register 9-33: DMA Transfer Complete Interrupt Clear Registers (ClearTfr)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0x0
[7:0]	CLEAR	RW	DMA Transfer Complete Interrupt Clear for channel7 to channel 0 ● 0: no effect ● 1: clear interrupt	0x0

Register 9-34: Combined Interrupt Status Register (StatusInt)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	N/A	Reserved	0x0
[4]	ERR	R	OR of the contents of StatusErr register.	0x0
[3]	DSTT	R	OR of the contents of StatusDst register.	0x0
[2]	SRCT	R	OR of the contents of StatusSrcTran register.	0x0
[1]	BLOCK	R	OR of the contents of StatusBlock register.	0x0
[0]	TFR	R	OR of the contents of StatusTfr register.	0x0

Register 9-35: Source Software Transaction Request Register (ReqSrcReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	SRC_REQ_WE	W	Source request write enable ● 0: write disabled ● 1: write enabled	0x0
[7:0]	SRC_REQ	RW	Source request	0x0

Register 9-36: Destination Software Transaction Request Register (ReqDstReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	DST_REQ_WE	W	Destination request write enable ● 0: write disabled ● 1: write enabled	0x0
[7:0]	DST_REQ	RW	Destination request	0x0

Register 9-37: Single Source Transaction Request Register (SglReqSrcReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	SRC_SQLREQ_WE	W	Single Source write enable ● 0: write disabled ● 1: write enabled	0x0
[7:0]	SRC_SQLREQ	RW	Source single request	0x0

Register 9-38: Single Destination Transaction Request Register (SglReqDstReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	DST_SQLREQ_WE	W	Destination write enable ● 0: write disabled ● 1: write enabled	0x0
[7:0]	DST_SQLREQ	RW	Destination single or burst request	0x0

Register 9-39: Last Source Transaction Request Register (LstSrcReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	LSTSRC_WE	W	Source last transaction request write enable ● 0: write disabled ● 1: write enabled	0x0
[7:0]	LSTSRC	RW	Source last transaction request ● 0: Not last transaction in current block ● 1: Last transaction in current block	0x0

Register 9-40: Last Destination Transaction Request Register (LstDstReg)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0x0
[15:8]	LSTDST_WE	W	Destination last transaction request write enable ● 0: write disabled ● 1: write enabled	0x0
[7:0]	LSTDST	RW	Destination last transaction request ● 0: Not last transaction in current block ● 1: Last transaction in current block	0x0

10 Display Subsystem

10.1 Overview

IMAPx210 display subsystem (IDS) consists of two parts, overlay controller and display controller. The main function of the overlay module is window blending, supports 4 window layers' mix. For example, system can use win0 as an OS window, win1 as a video player window with win2 as a menu, win3 as a mouse cursor image information. Win3 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.

Display controller is used to transfer data to external display devices, such as TFT LCD panel, i80 LCD panel, TV and etc. 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.

Features

Overlay Function

- 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 24-BPP non-palletized color
  - Supports multi-format RGB and YCbCr 4:2:0
  - Supports X,Y indexed position
  - Supports 4 bit Alpha blending: Plane / Pixel
- Up to 4 layers PIP(OSD) function
- Up to 2Kx2K screen resolution
- Window Source format
  - Supports 1, 2, 4 or 8-BPP (bit per pixel) palletized color
  - Supports 8, 16, 18 or 24-BPP non-palletized color
- Palette Usage
  - 256 x 25(ARGB) bits palette RAM for Window 0
  - 256 x 25(ARGB) bits palette RAM for Window1
  - 16(entry) x 16 bits palette Register Array for Window 2
  - 16(entry) x 16 bits palette Register Array for Window 3
- Soft Scrolling
  - Horizontal : 1 pixel resolution
  - Vertical : 1 pixel resolution
- Supports Transparent Overlay
- Supports hardware cursor
- Supports Color key function

TFT LCD Displays

- Supports maximum 16M color TFT at 24bit per pixel mode
- Supports multiple screen size

— Typical actual screen size: 1920x1080, 1024x768, 800x480, 640 x 480, 320 x 240, and others
— Maximum virtual screen size is 4Mbytes.
— Maximum virtual screen size in 64K color mode: 2048 x 2048 and others

TV Displays

- Built-in Color Space Conversion matrix
- External or internal clock selection
- Support ITU-R BT 656 8-bit interface (YCbCr422)
- Support ITU-R BT 601 16-bit/8-bit interface (YCbCr422)
- Support ITU-R BT 601 24-bit interface (RGB888)
- Support user defined size resolution up to HDTV
- Support user defined frame and field size

I80 Displays

- Support 16 normal command by software trigger
- Support 16 auto-command before video data
- Programmable timer
- Support 18/16/9/8 wire connection
- Automatism and Manual operation

Block Diagram

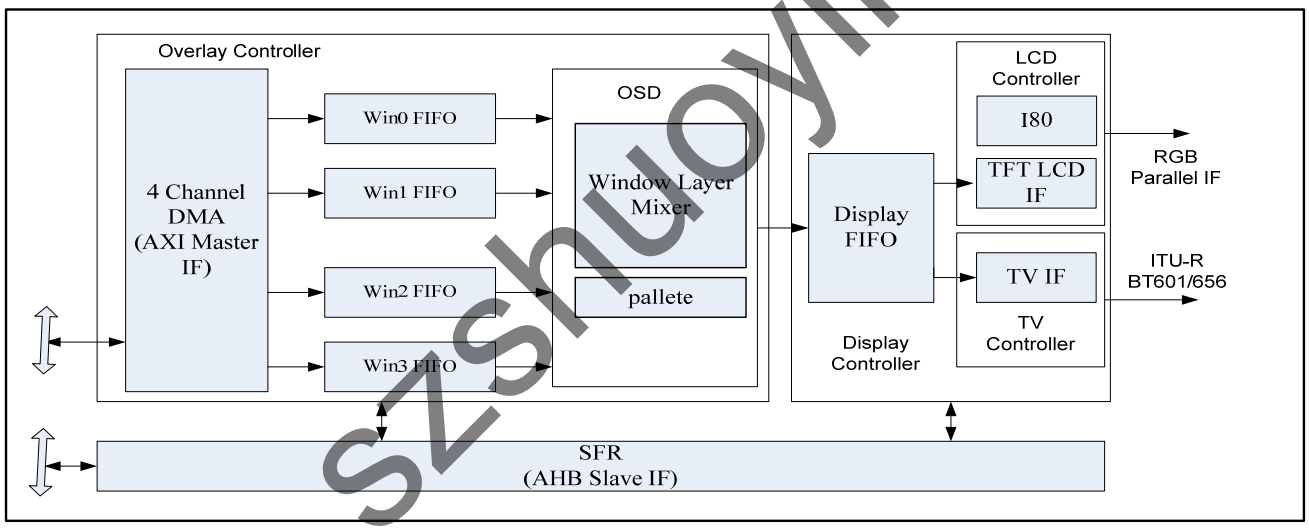


Figure 10-1: IDS Block Diagram

External Interface Signal

Table 10-1: IDS External Interface Signal

Chip PAD	TFT LCD	TV	I80
XvVCLK	RGB_VCLK	TV_PCLK	I80_NWR
XvVSYNC	RGB_VSYNC	TV_VSYNC	I80_NCS1
XvHSYNC	RGB_HSYNC	TV_HSYNC	I80_NCS0
XvVDEN	RGB_VDEN	TV_HREF	I80_NRS
XvPWREN	RGB_PWREN	--	I80_NRD
XvVD[23:0]	RGB_VD[23:0]	TV_DAT[23:0]	I80_DAT
XTOUT3	--	TV_RCLK	--
XTOUT2	--	TV_FIELD	--

10.2 IDS Function Description

10.2.1 Overlay Controller Operation

The overlay operation consists of displaying more than one layer (background, graphics and video layers) using rules based on priority, alpha blending and color keys.

10.2.1.1 Overlay Priority

Rule: Win 3 > Win 2 > Win 1 > Win 0

For instance, the video1 layer is always on top of the graphics layer. The video2 layer is always on top of the video1 and graphics. The overlay controller reads the data for each buffer from the system memory and, depending on the transparency color key values, displays either the pixels in the video layer, the pixels in the graphics layer, or the solid background color.

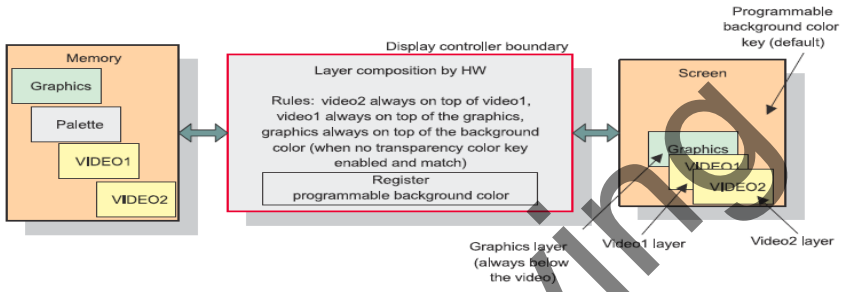


Figure 10-2: Typical Overlay Case

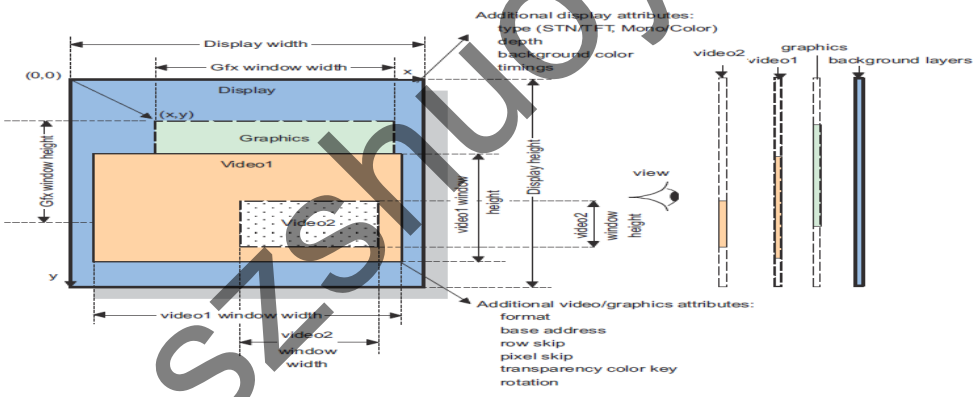


Figure 10-3: Overlay Result

Each layer can have any size up to full-display screen. If there are no graphics or video-encoded pixels at a specific position, the programmable, solid background color appears.

10.2.1.2 Color Key

The COMPKEY of every window Color Key Control Register must be set by 24bit RGB format, defining the encoded pixel data considered as the transparent pixel. Color image of OSD layer, which is specified by Color Key Register, are pixels not visible on the screen, and the underlayer encoded pixel values or solid background color are visible.

Figure 10-4: Color Key Application Example

10.2.1.3 Alpha Blending

Alpha Factor

Display controller can blend 4 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). 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). AR will be selected by applying the below equation.

If WINEN_F is enabled and BLD_ PIX == 1 then

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);

else if WINEN_F is enabled and BLD_PIX==0

AR= ALPHA0_R;

AG= ALPHA0_G;

AB= ALPHA0_B;

The table below shows how to select blending factor in all condition.

Table 10-2: Blending User's Table

<table>
<tr><td colspan="2"></td><td colspan="2">ALPHA_SEL value @ OVCW1/2/3CR[6]</td></tr>
<tr><td colspan="2"></td><td>'0'</td><td>'1'</td></tr>
<tr><td>BLD_PIX value @ OVCW1/2/3CR[6]</td><td>'0'</td><td>Plane blending using ALPHA0</td><td>Plane blending using ALPAH1</td></tr>
<tr><td rowspan="2"></td><td rowspan="2">'1'</td><td>Pixel blending selected by AEN
<table><tr><td colspan="2">AEN value @ Frame Buffer</td></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><td colspan="2">KEYBLENvalue @OVCW1/2/3 CKCR[26]</td></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 value @ OVCW/2/3CR[4:1] is 'b1101'</td></tr>
</table>

The ratios between the main display layer and the alpha blending layer are:

- Alpha value 0000 = 100% main display layer value
- Alpha value 0001 = 15/16 main display layer value + 1/16 alpha blending layer value
- Alpha value 0010 = 14/16 main display layer value + 2/16 alpha blending layer value
- Alpha value 0011 = 13/16 main display layer value + 3/16 alpha blending layer value

- Alpha value 1101 = 3/16 main display layer value + 13/16 alpha blending layer value
- Alpha value 1110 = 2/16 main display layer value + 14/16 alpha blending layer value
- Alpha value 1111 = 100% alpha blending layer value

Blending Equation

The illustration below is described as the example of the R (Red) output using ALPHA_R value of each window.

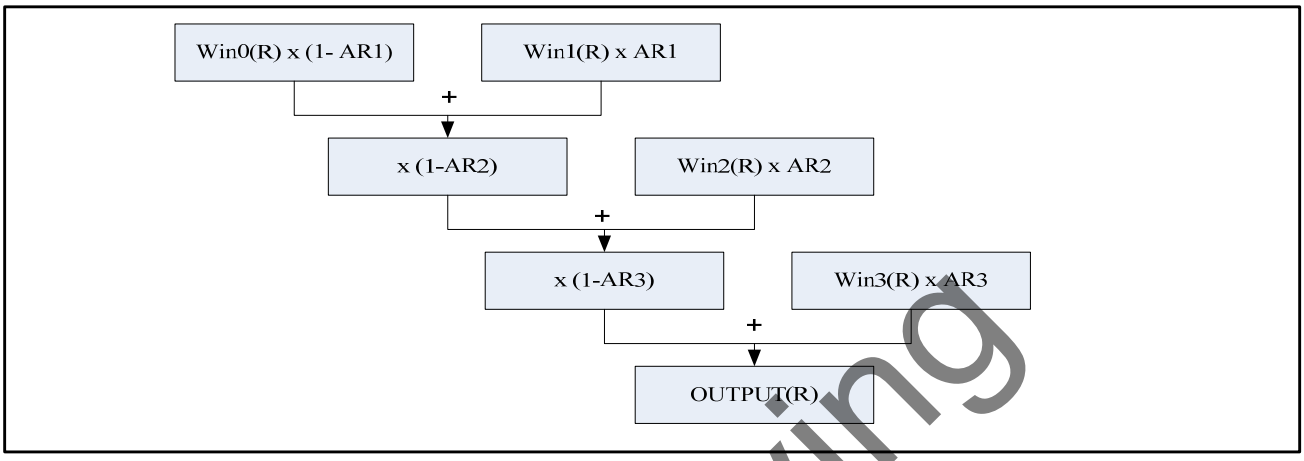


Figure 10-5: Blending Diagram

Blending equation:

Win01(R) = Win0(R) x Beta + Win1(R) x Alpha
Win01(G) = Win0(G) x Beta + Win1(G) x Alpha
Win01(B) = Win0(B) x Beta + Win1(B) x Alpha
Win012(R) = Win01(R) x Beta + Win2(R) x Alpha
Win012(G) = Win01(G) x Beta + Win2(G) x Alpha
Win012(B) = Win01(B) x Beta + Win2(B) x Alpha
Win0123(R) = Win012(R) x Beta + Win3(R) x Alpha
Win0123(G) = Win012(G) x Beta + Win3(G) x Alpha
Win0123(B) = Win012(B) x Beta + Win3(B) x Alpha
WinOut(R) = Win0123(R) x Beta + Win4(R) x Alpha
WinOut(G) = Win0123(G) x Beta + Win4(G) x Alpha
WinOut(B) = Win0123(B) x Beta + Win4(B) x Alpha

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)

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)

In case Win01(R), Alpha and Beta value is determined by

Alpha = AR1/16, Beta = (15-AR1)/16

if AR1 == 0xF then Alpha =1 and Beta =0

if AR1 == 0x0 then Alpha =0 and Beta =1

10.2.1.4 Window Buffer Data Format

The overlay controller requests the specified memory format of frame buffer. The next table shows some examples of each display mode.

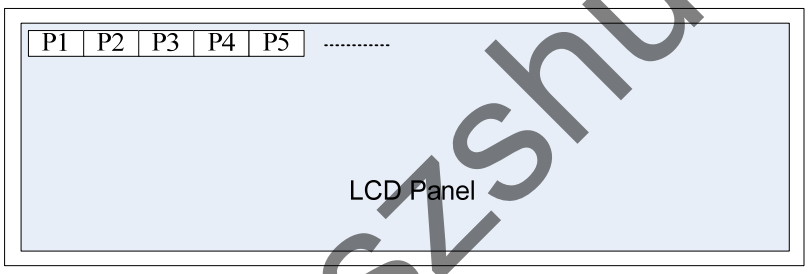
28BPP Mode (A4+RGB888)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	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)

Buffer Address Offset	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
…			

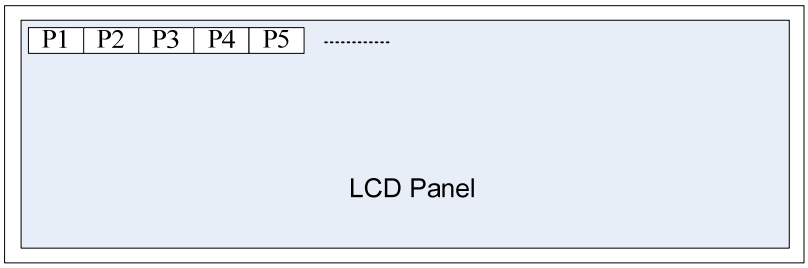


Note: D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data,
In case of BLD_PIX and ALPHA_SEL are set,
D[27:24] = Alpha value, D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

25BPP Mode (A1+RGB888)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:25]	D[24]	D[23:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
…			



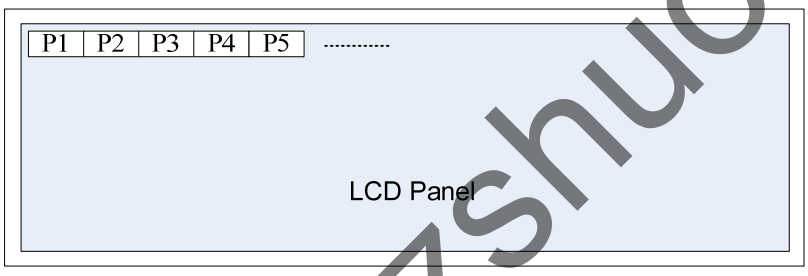
Notes:

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.
2. D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

24BPP Mode (A1+RGB887)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:24]	D[23]	D[22:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
…			



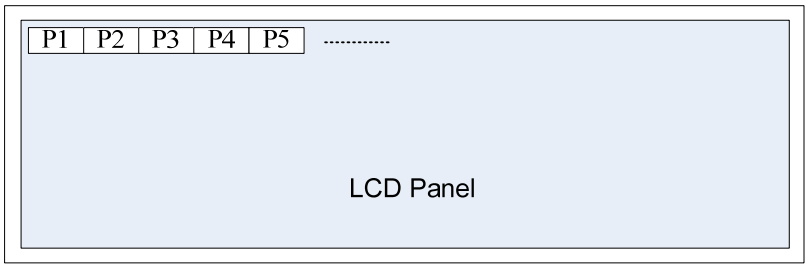
Notes:

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.
2. D[22:15] = Red data, D[14:7] = Green data, D[6:0] = Blue data

24BPP Mode (RGB888)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:24]	D[23:0]
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
…		

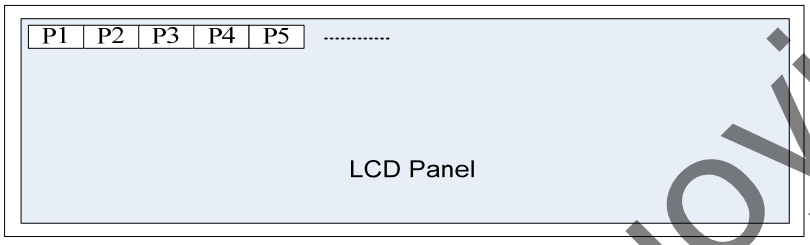


Note: D[23:16] = Red data, D[15:8] = Green data, D[7:0] = Blue data

19BPP Mode (A1+RGB666)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:19]	D[18]	D[17:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
…			



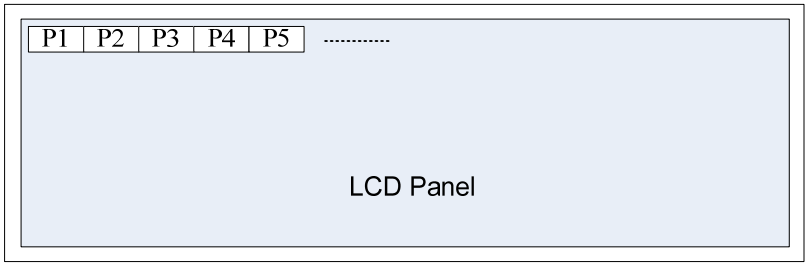
Notes:

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.
2. D[17:12] = Red data, D[11:6] = Green data, D[5:0] = Blue data

18BPP Mode (A1+RGB665)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:18]	D[17]	D[16:0]
000H	Dummy Bit	AEN	P1
004H	Dummy Bit	AEN	P2
008H	Dummy Bit	AEN	P3
…			



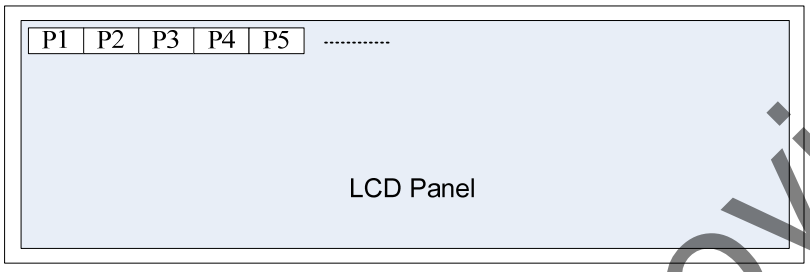
Notes:

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.
2. D[16:11] = Red data, D[10:5] = Green data, D[4:0] = Blue data

18BPP Mode (RGB666)

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:18]	D[17:0]
000H	Dummy Bit	P1
004H	Dummy Bit	P2
008H	Dummy Bit	P3
…		



Note: D[17:12] = Red data, D[11:6] = Green data, D[5:0] = Blue data

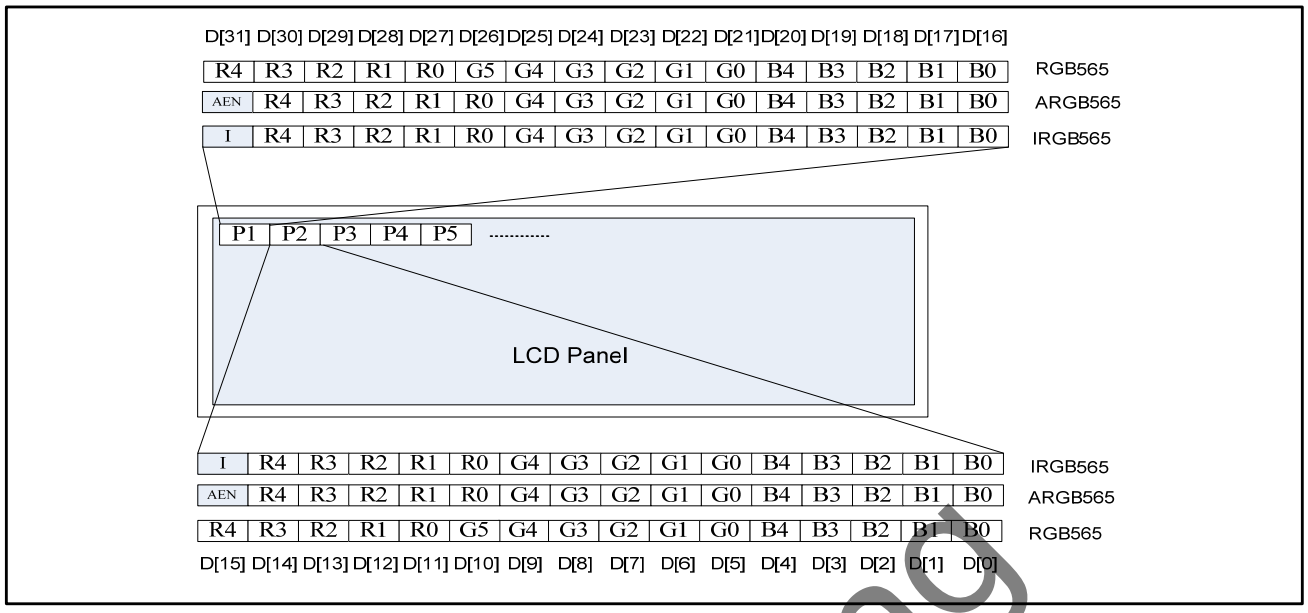
16BPP Mode

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:16]	D[15:0]
000H	P1	P2
004H	P3	P4
008H	P5	P6
…		

(BSWP = 0, HWSWP = 1)

Buffer Address Offset	D[31:16]	D[15:0]
000H	P2	P1
004H	P4	P3
008H	P6	P5
…		



Notes:

1. IRGB555 mode, {D[14:10], D[15] } = Red data, {D[9:5], D[15] } = Green data, {D[4:0], D[15]}= Blue data
2. ARGB565 mode
   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.
   2). D[14:10] = Red data, D[9:5] = Green data, D[4:0] = Blue data
3. RGB565 mode, D[15:11] = Red data, D[10:5] = Green data, D[4:0] = Blue data

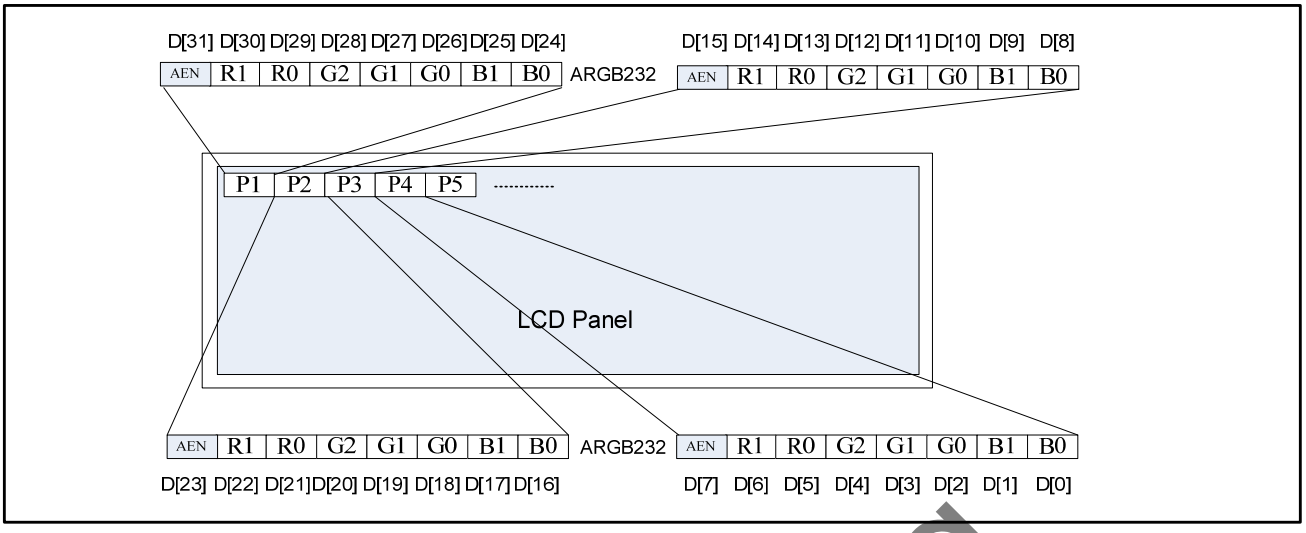
8BPP Mode（A1+RGB232）

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	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)

Buffer Address Offset	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
…				



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.
2. D[6:5] = Red data, D[4:2] = Green data, D[1:0] = Blue data

8BPP Mode（Palette）

(BSWP = 0, HWSWP = 0)

Buffer Address Offset	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)

Buffer Address Offset	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: The values of frame buffer are index of palette memory, and the index is range from 0 to 255.

4BPP Mode（Palette）

(Bits4SWP = 0, BSWP = 0, HWSWP = 0)

Buffer Address Offset	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
…								

(Bits4SWP = 0, BSWP = 1, HWSWP = 0)

Buffer Address Offset	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
…								

(Bits4SWP = 1, BSWP = 1, HWSWP = 0)

Buffer Address Offset	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	P8	P7	P6	P5	P4	P3	P2	P1
004H	P16	P15	P14	P13	P12	P11	P10	P9
008H	P24	P23	P22	P21	P20	P19	P18	P17
…								

Note: The values of frame buffer are index of palette memory, and the index is range from 0 to 15.

2BPP Mode（Palette）

(Bits2SWP = 0, Bits4SWP = 0, BSWP = 0, HWSWP = 0)

Buffer Address Offset	D[31:30]	D[29:28]	D[27:26]	D[25:24]	D[23:22]	D[21:20]	D[19:18]	D[17:16]
000H	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
…								
Buffer Address Offset	D[15:14]	D[13:12]	D[11:10]	D[9:8]	D[7:6]	D[5:4]	D[3:2]	D[1:0]
000H	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
…								

Notes:

1. The values of frame buffer are index of palette memory, and the index is range from 0 to 3.
2. Bits4SWP is used to swap high and low 4 bit of each byte.
3. Bits2SWP is used to swap high and low 2 of each half-byte.

1BPP Mode（Palette）

(BITSWP = 0, Bits2SWP = 0, Bits4SWP = 0, 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	P9	P10	P11	P12	P13	P14	P15	P16
004H	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	P47	P48
…																
D	**[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
…																

Notes:

1. The values of frame buffer are index of palette memory, and the index is range from 0 to 1.
2. Bits4SWP is used to swap high and low 4 bit of each byte.
3. Bits2SWP is used to swap high and low 2 of each half-byte.
4. BITSWP is used to swap bit of each pair bits.

10.2.1.5 Palette Usage

The overlay controller can support the 256 colors palette memory for window 0 and 1, 16 colors palette register array for window 2 and 3.

256 color palette consist of 256(depth) x 25-bit SRAM and support up to eight data format, include ARGB888, RGB888, ARGB666, ARGB665, RGB666, ARGB555 and RGB565. Palette data endian mode is little-endian and the msb is Reserved when source data format is not ARGB888. 16 color palette consist of 16(depth) x 16-bit register array and only support two data format, include ARGB555 and RGB565.

It is important to know that only firstly set UPDATE_PAL of register OVCPCR, CPU can write or read palette.

Table 10-3: 25BPP(ARGB888) Palette Data Format

Index	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
0	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
1	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
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
255	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

Table 10-4: 16BPP(ARGB555) Palette Data Format

Index	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
0	--	--	--	--	--	--	--	--	--	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
1	--	--	--	--	--	--	--	--	--	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
…	--	--	--	--	--	--	--	--	--	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
255	--	--	--	--	--	--	--	--	--	AEN	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

10.2.1.6 Ping-Pong Buffer and Timeslot Interrupt

Overlay controller window 0 and window 1 have two frame buffers and can be used as ping-pong buffer. According to corresponding register configuration of window control register (OVCW0CR and OVCW1CR), the two ping-pong buffers can be switched in auto or manual method. See register description for more details.

Overlay controller can generate one window timeslot interrupt of each window (0 and 1). User can update frame buffer data according to the timeslot interrupt for many application, especially in video play. Usually, the decoded rate of video data is about 30 fps and the LCD refresh rate is about 60-70 fps. In order to avoiding potential risk, overlay controller use two PWM Timer interrupt (PWM_INT2/3, PWM_INT2 for window 0 timeslot interrupt and PWM_INT3 for window 1 timeslot interrupt) to resolve it and the synchronal process is shown in following figure.

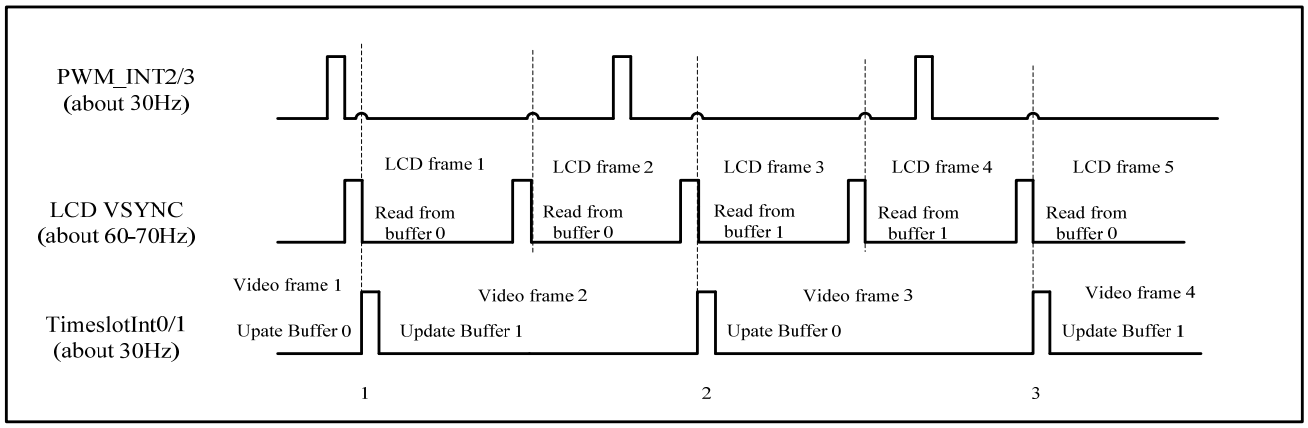


Figure 10-6: Window Timeslot Interrupt

As shown in Figure 10-6, timeslot interrupt is synchronized to PWM_INT2/3 and VSYNC. At time 1, timeslot interrupt is generated and from this time video data of next frame can safely and have enough time to write/decode to buffer 1 because overlay controller is switched to read video data from buffer 0 at the next 1 or 2 LCD frame in auto ping-pong buffer method. Similarly, at time 2, buffer 0 is released and can be updated.

It is important to know that before using window timeslot interrupt, you must set PWM Timer 2 or 3 properly and allow timer interrupt generate. Although PWM Timer is allowed to generate interrupt, you needn't acknowledge (ignore PWM_INT2/3 interrupt and mask it in interrupt controller) as the PWM Timer interrupt output pulse is connected with overlay controller internally.

10.2.1.7 Window Coordinate and Virtual Display

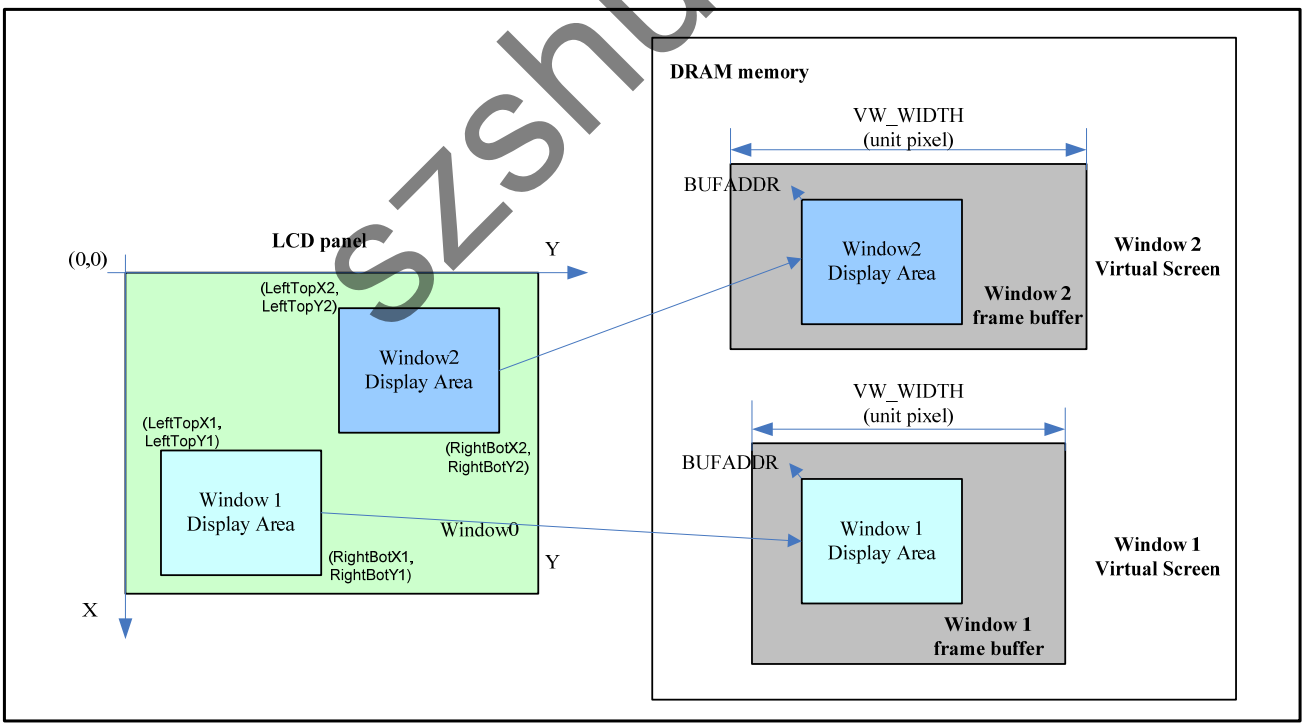


Figure 10-7: Window Coordinate of OSD Image

As shown in Figure 10-7, window coordinate origin is Left top of LCD panel. Each window coordinate position is aligned to the LCD panel coordinate. Buffer Start Address (BUFADDR) of every window is the address of left top of window display area in DRAM memory. Virtual Screen Page Width (VW_WIDTH) is the horizontal pixel size of window virtual screen.

The overlay controller supports the hardware horizontal or vertical scrolling. If the screen is scrolled, you must change the window coordinate and buffer start address and must not change the values of VW_WIDTH.

10.2.2 LCD Controller Operation

LCD controller is used to transfer the video data and to generate the necessary control signals, such as VSYNC, HSYNC, VCLK, VDEN, and so on.

The image data format and 4 windows blending is proceeded in overlay controller. LCD controller transfer 24-bit RGB pixel data to external TFT LCD according to corresponding timing. The LCD controller and overlay controller exchange pixel data by one 16 entry, 24-bit RGB FIFO, which is shown in Figure 10-1.

10.2.2.1 LCD Interface Timing

LCD Controller generates the control signals for the LCD driver, such as VSYNC, HSYNC, VCLK, and VDEN. These control signals are closely related to the configuration on the LCDCON1/2/3/4/5 registers. Based on these programmable configurations on the LCD control registers, controller can generate the programmable control signals suitable to support many different types of LCD drivers.

TFT Timing

The VSYNC signal is asserted to cause the LCD's line pointer to start over at the top of the display.

The VSYNC and HSYNC pulse generation depends on the configurations of both the HOZVAL field and the LINEVAL field in the LCDCON2/3 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 VCLK signal depends on the CLKVAL field in the LCDCON1 register. Table 10-5 defines the relationship of VCLK and CLKVAL. The minimum value of CLKVAL is 0.

VCLK(Hz) = IDSTF_CLK/(CLKVAL+1)

The frame rate is VSYNC signal frequency. The frame rate is related with the field of VSYNC, VBPD, VFPD, LINEVAL, HSYNC, HBPD, HFPD, HOZVAL, and CLKVAL in LCDCON1 and LCDCON2/3/4 registers. Most LCD drivers need 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) / (IDSTF_CLK) }]

Table 10-5: Relation Between VCLK and CLKVAL (60MHz)

CLKVAL	60MHz/X	VCLK
1	60 MHz/2	30.0 MHz
2	60 MHz/3	20.0 MHz
:	:	:
1023	60 MHz/1024	58.6 kHz

Figure 10-8 shows the timing requirements for the TFT LCD driver interface.

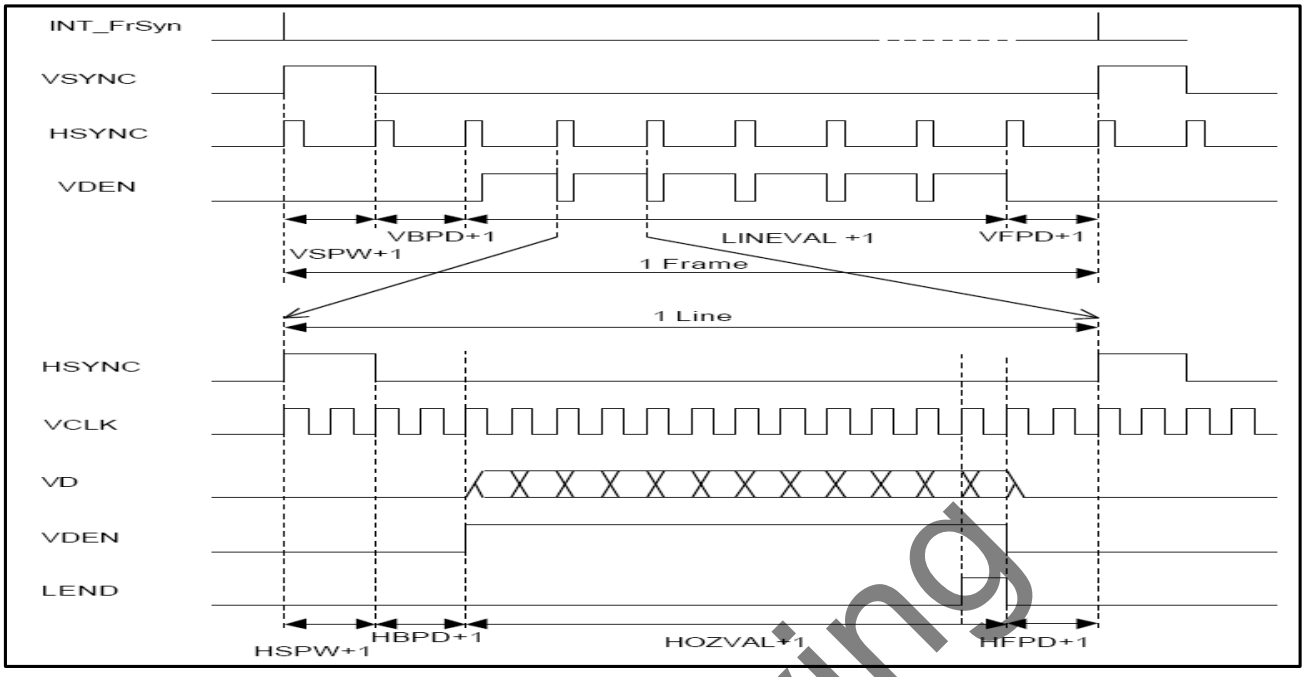


Figure 10-8: TFT LCD Timing

LCD Power Enable (TFT)

The LCD controller provides Power enable (PWREN) function. When PWREN is set to make PWREN signal enabled, the output value of XvPWREN pin is controlled by ENVID. In other words, If XvPWREN pin is connected to the power on/off control pin of the LCD panel, the power of LCD panel is controlled by the setting of ENVID automatically.

The LCD controller also supports INVPWREN bit to invert polarity of the PWREN signal.

This function is available only when LCD panel has its own power on/off control port and when port is connected to XvPWREN pin.

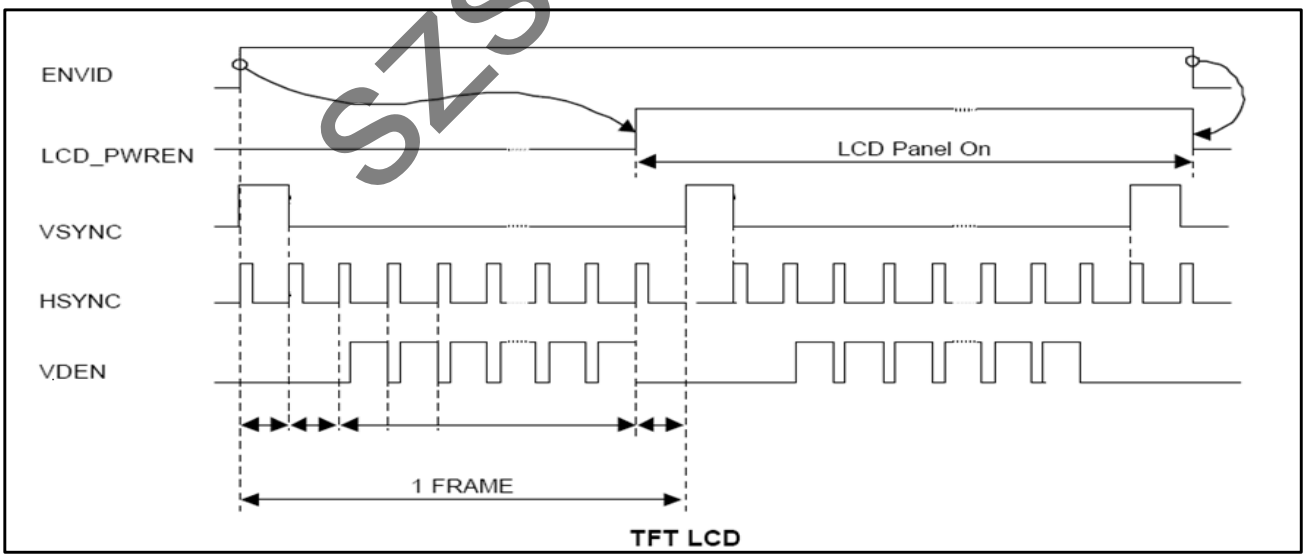


Figure 10-9: Example of PWREN Function (PWREN=1, INVPWREN=0)

10.2.3 TV Controller Operation

The TV Controller convert RGB data of Display RGB FIFO to YCbCr and transfer to external TV Encoder Chip. The YCbCr video data will be packaged with blanking line and blanking data and the interface format compatible with ITU-R BT 601/656 interface. The frequency of interface is variable and depend on the size of image. The interface signals consist of video data bus and necessary control signal, such as VSYNC, HSYNC, HREF, FIELD, and so on. Internal RGB2YCbCr color space conversion matrix coefficients can be configured by software.

10.2.3.1 Interface Timing

The interface timing support two mode, one is line begins with blanking and the other is line begin with EAV. Which mode is selected depend on the value of register TFIF_CFG[0]. The below figure shows the case of line begin with EAV.

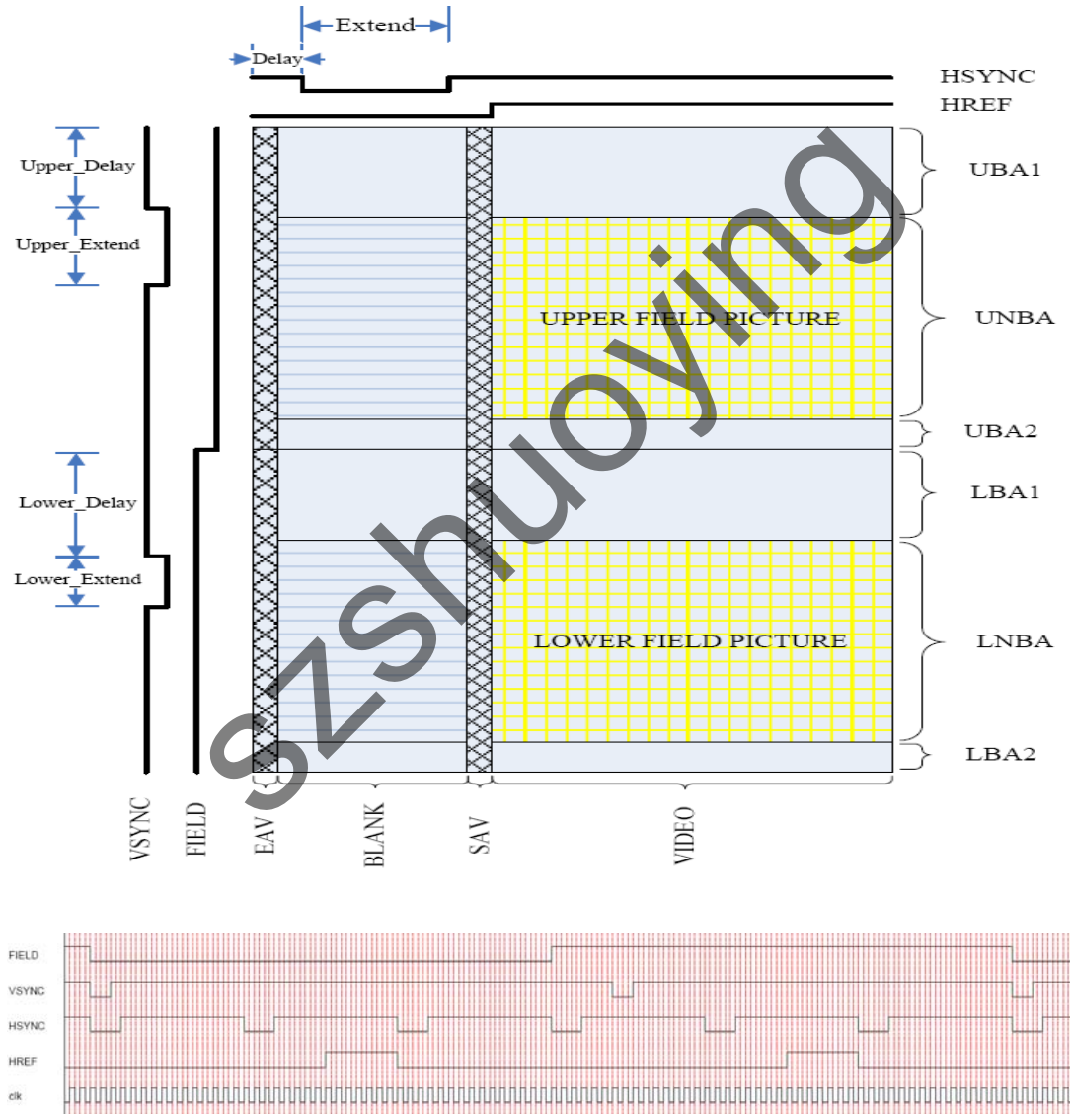


Figure 10-10: Timing of Line Begin with EAV

Note: HREF is only valid in data field, not including blanking field.

10.2.3.2 Data Format

TVIF controller supports 4 type of interface:

1) ITU656 8-bit YCbCr422, when TVIF_CFG[28] = 0, TVIF_CFG[30] = 0, TVIF_CFG[29] = 0, TVIF_MATRIX_CFG[30] = 0.
2) ITU601 8-bit YCbCr422, when TVIF_CFG[28] = 0, TVIF_CFG[30] = 1, TVIF_CFG[29] = 0, TVIF_MATRIX_CFG[30] = 0.
3) ITU601 16-bit YCbCr422, when TVIF_CFG[28] = 0, TVIF_CFG[30] = 1, TVIF_CFG[29] = 1, TVIF_MATRIX_CFG[30] = 0.
4) ITU601 24-bit RGB888, when TVIF_CFG[28] = 1. TVIF_MATRIX_CFG[30] = 1.

When output data width is 8-bit, it is 2 clock cycle to send 1 pixel data, the TVIF_BLANK_LEN and TVIF_VIDEO_LEN should be the double of active data and blanking data. At the same time, TVIF_CLK_CFG should be adjusted.

Data output format and output order are supported as followed:

- **24-bit mode:**

LCD_VD	ORDER	1st	2nd	3rd	4th	5th	6th
[23:16]	00:	[Y0]	[Y1]	[Y2]	[Y3]	[Y4]	[...]
[15: 8]		[Cb0]	[Cb1]	[Cb2]	[Cb3]	[Cb4]	[...]
[7: 0]		[Cr0]	[Cr1]	[Cr2]	[Cr3]	[Cr4]	[...]
[23:16]	01:	[Y0]	[Y1]	[Y2]	[Y3]	[Y4]	[...]
[15: 8]		[Cr0]	[Cr1]	[Cr2]	[Cr3]	[Cr4]	[...]
[7: 0]		[Cb0]	[Cb1]	[Cb2]	[Cb3]	[Cb4]	[...]
[23:16]	10:	[Cb0]	[Cb1]	[Cb2]	[Cb3]	[Cb4]	[...]
[15: 8]		[Cr0]	[Cr1]	[Cr2]	[Cr3]	[Cr4]	[...]
[7: 0]		[Y0]	[Y1]	[Y2]	[Y3]	[Y4]	[...]
[23:16]	11:	[Cr0]	[Cr1]	[Cr2]	[Cr3]	[Cr4]	[...]
[15: 8]		[Cb0]	[Cb1]	[Cb2]	[Cb3]	[Cb4]	[...]
[7: 0]		[Y0]	[Y1]	[Y2]	[Y3]	[Y4]	[...]

Note1: In 24-bit mode, output data format is RGB mode, the relation of RGB and YCbCr is R<->Y, G<->Cb, B<->Cr.
Note2: The control signals of VSYNC, HSYNC, HREF, FIELD and etc. are also output in RGB mode.

- **16-bit mode :**

LCD_VD	ORDER	1st	2nd	3rd	4th	5th	6th
[23:16]	00:	[0]	[0]	[0]	[0]	[0]	[...]
[15: 8]		[Y0]	[Y1]	[Y2]	[Y3]	[Y4]	[...]
[7: 0]		[Cr0]	[Cb0]	[Cr2]	[Cb2]	[Cr4]	[...]
[23:16]	01:	[0]	[0]	[0]	[0]	[0]	[...]

[15: 8]		[Y0][Y1][Y2][Y3][Y4][...]
[7: 0]		[Cb0][Cr0][Cb2][Cr2][Cb4][...]
[23:16]	10:	[0][0][0][0][0][...]
[15: 8]		[Cr0][Cb0][Cr2][Cb2][Cr4][...]
[7: 0]		[Y0][Y1][Y2][Y3][Y4][...]
[23:16]	11:	[0][0][0][0][0][...]
[15: 8]		[Cb0][Cr0][Cb2][Cr2][Cb4][...]
[7: 0]		[Y0][Y1][Y2][Y3][Y4][...]

- **8-bit mode:**

LCD_VD	ORDER	1st	2nd	3rd	4th	5th	6th
[23:16]	00:	[0]	[0]	[0]	[0]	[0]	[...]
[15: 8]		[0]	[0]	[0]	[0]	[0]	[...]
[7: 0]		[Cb0]	[Y0]	[Cr0]	[Y1]	[Cb2]	[...]
[23:16]	01:	[0]	[0]	[0]	[0]	[0]	[...]
[15: 8]		[0]	[0]	[0]	[0]	[0]	[...]
[7: 0]		[Cr0]	[Y0]	[Cb0]	[Y1]	[Cr2]	[...]
[23:16]	10:	[0]	[0]	[0]	[0]	[0]	[...]
[15: 8]		[0]	[0]	[0]	[0]	[0]	[...]
[7: 0]		[Y0]	[Cb0]	[Y1]	[Cr0]	[Y2]	[...]
[23:16]	11:	[0]	[0]	[0]	[0]	[0]	[...]
[15: 8]		[0]	[0]	[0]	[0]	[0]	[...]
[7: 0]		[Y0]	[Cr0]	[Y1]	[Cb0]	[Y2]	[...]

Matrix Convert

Built-in matrix convert implement RGB and YCbCr data format convert, the general matrix formula is shown as followed:

$$\begin{bmatrix} out\_0 \\ out\_1 \\ out\_2 \end{bmatrix} = \frac{1}{256} \times \begin{bmatrix} coef11 & coef12 & coef13 \\ coef21 & coef22 & coef23 \\ coef31 & coef32 & coef33 \end{bmatrix} \left(\begin{bmatrix} in\_0 \\ in\_1 \\ in\_2 \end{bmatrix} - \begin{bmatrix} oft\_b \\ * \\ * \end{bmatrix} \right) + \begin{bmatrix} oft\_a \\ * \\ * \end{bmatrix}$$

The coefficient coef11, coef12, coef13, coef21, coef22, coef23, coef31, coef32 and coef33 are signed numbers, which value are ranged from -1024 to +1023.

Parameter oft_a and oft_b are unsigned numbers, which value are ranged from 0 to 31.

In TVIF controller, only RGB to YCbCr convert is used, the value of oft_a and oft_b are set to 16 and 0 separately. The value

of in_1 and in_2 will be added by 128. If user do NOT want the addition, please set the “inv_MSB_out” in register TVIF_MATRIX_CFG.

The RGB to YCbCr matrix convert formula is shown as following:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{256} \times \begin{bmatrix} 65.7378 & 129.0571 & 25.0940 \\ -37.9452 & -74.4940 & 112.4392 \\ 112.4392 & -94.1538 & -18.2854 \end{bmatrix} \left(\begin{bmatrix} R \\ G \\ B \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

10.2.3.3 ITU-R BT 601/656 Timing

- **ITU-R BT 601 Timing Diagram**

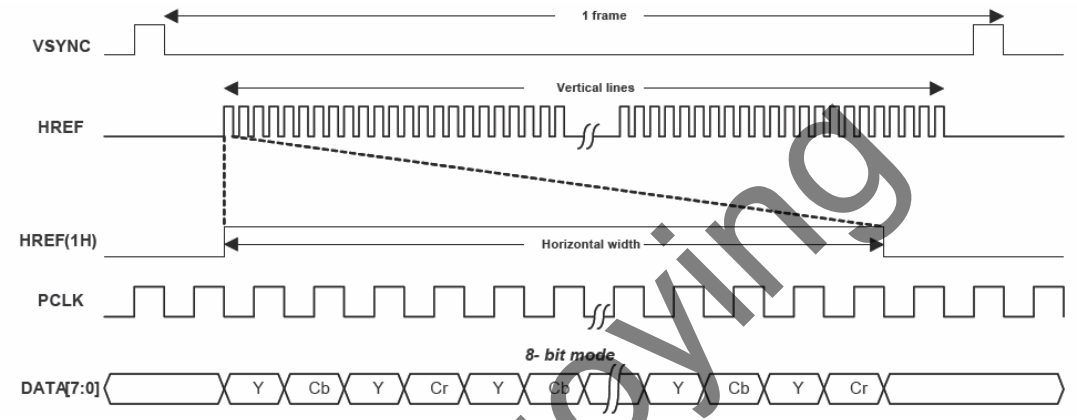


Figure 10-11: ITU-R BT601 8-bit Timing Diagram

- **ITU-R BT 656 Timing Diagram**

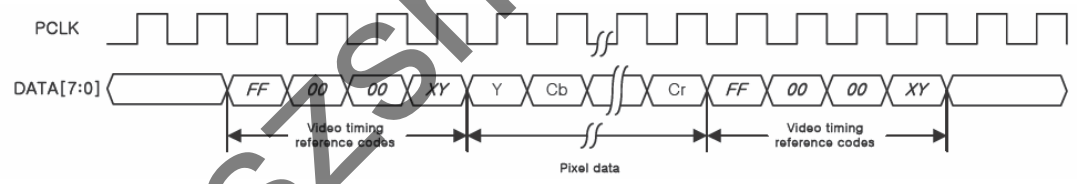


Figure 10-12: ITU-R BT 656 Timing Diagram

10.2.4 I80 Controller Operation

10.2.4.1 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.

I80C 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 = 1,2,4, 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

1) For RS polarity, refer to your LDI specification.

2) LDI_CMD need not to be packed from LDI_CMD0 to LDI_CMD15 contiguously. For example, only the use of LDI_CMD0, LDI_CMD3 and LDI_CMD15 is possible.
3) Maximum 16 auto commands are available.
4) Commands are issued in the order of CMD0 → CMD1 → CMD2 → CMD3 → ….. → CMD14 → CMD15
5) Disabled commands (CMDx_EN = 0x0) are skipped

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

Normal Command

1) Put commands into LDI_CMD0 ~ 15 (maximum 16 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 I80TRIGCON.

10.2.4.2 I80 Interface Timing

Figure 10-13 illustrates the timing for a operation of i80.The parameter LCD_CS_SETUP、LCD_WR_SETUP、LCD_WR_ACT、LCD_WR_HOLD reference to register I80IFCON0.

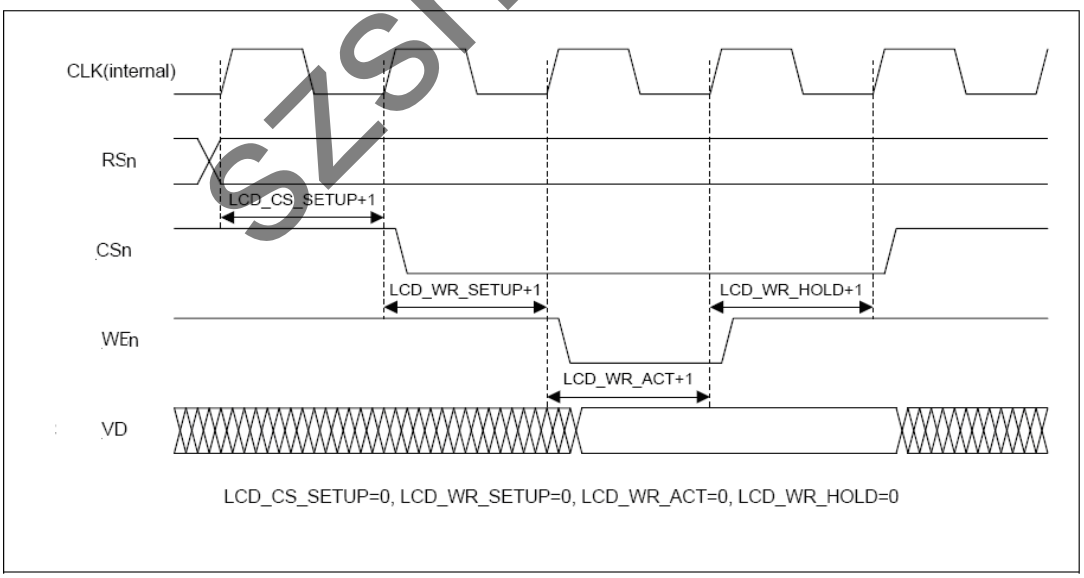
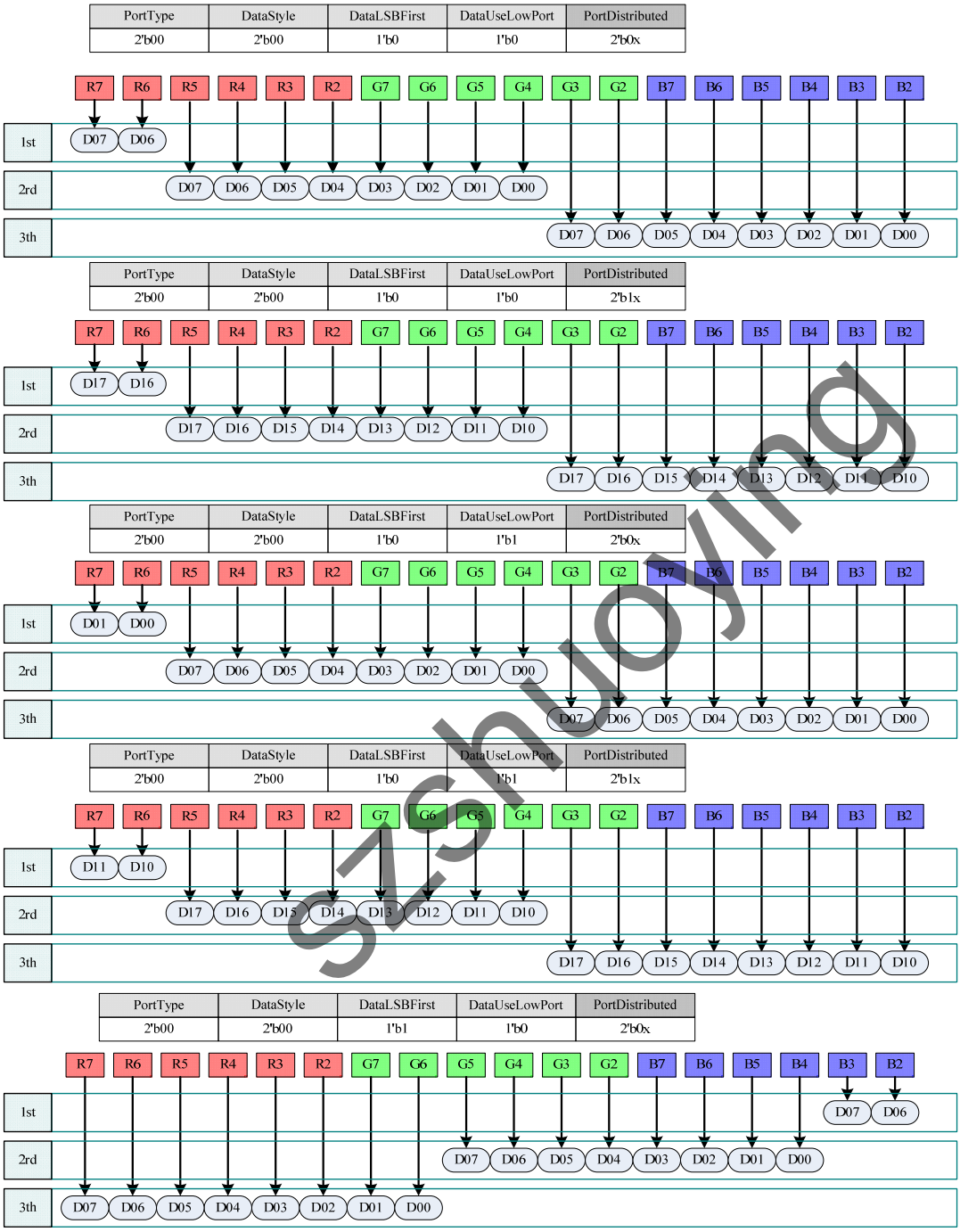
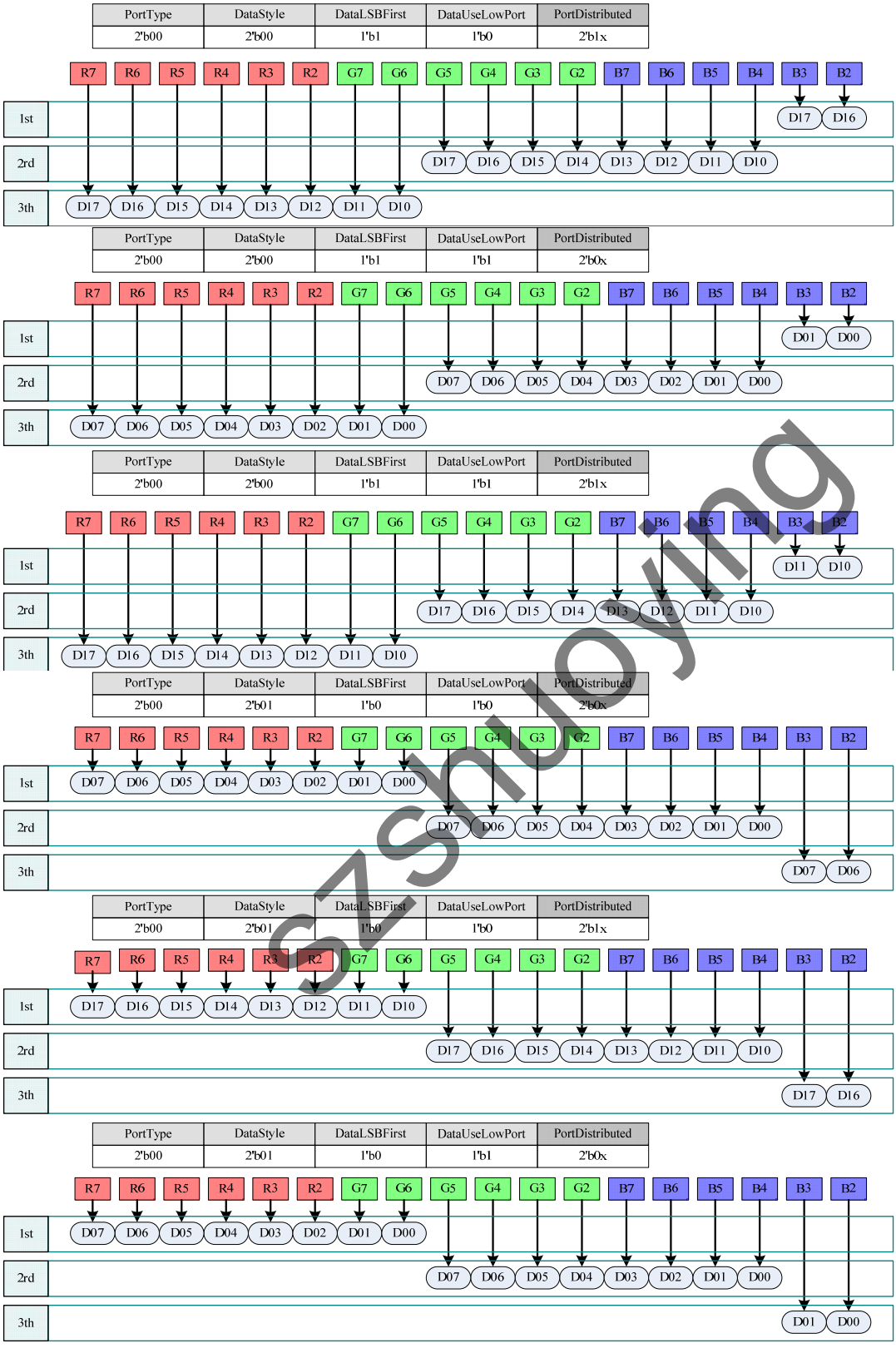


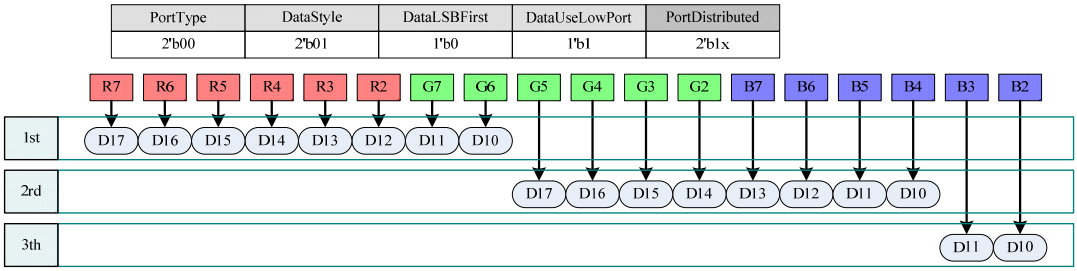
Figure 10-13: I80 Interface Timing

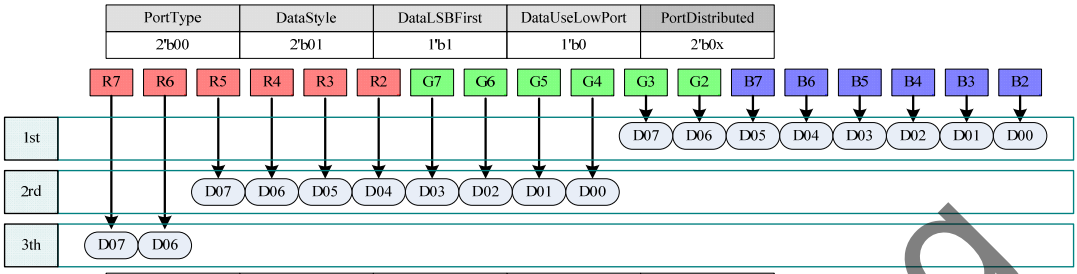
10.2.4.3 I80 Interface Data Structure

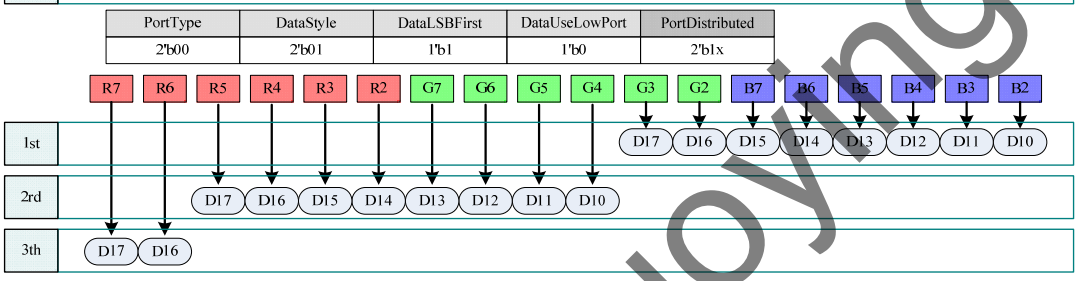
8-Wire Interface

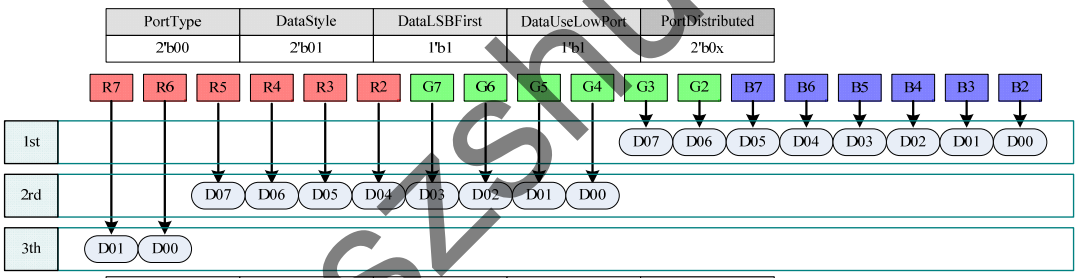


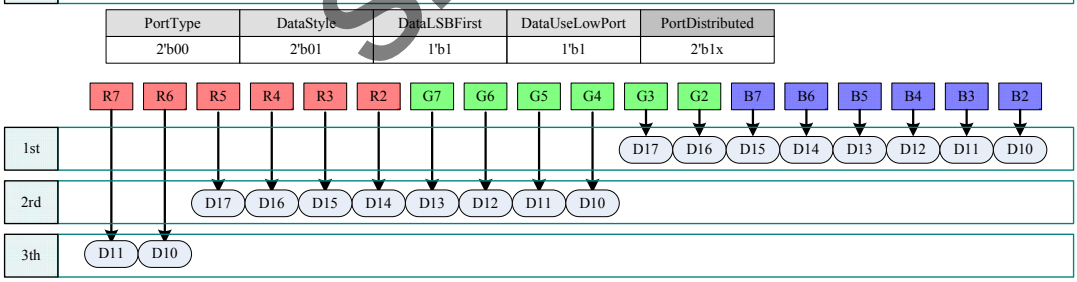

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b00
1'b1
1'b0
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16
2rd
D17 D16 D15 D14 D13 D12 D11 D10
3th
D17 D16 D15 D14 D13 D12 D11 D10
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b00
1'b1
1'b1
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D01 D00
2rd
D07 D06 D05 D04 D03 D02 D01 D00
3th
D07 D06 D05 D04 D03 D02 D01 D00
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b00
1'b1
1'b1
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D11 D10
2rd
D17 D16 D15 D14 D13 D12 D11 D10
3th
D17 D16 D15 D14 D13 D12 D11 D10
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b0
1'b0
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D07 D06 D05 D04 D03 D02 D01 D00
2rd
D07 D06 D05 D04 D03 D02 D01 D00
3th
D07 D06
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b0
1'b0
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10
2rd
D17 D16 D15 D14 D13 D12 D11 D10
3th
D17 D16
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b0
1'b1
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D07 D06 D05 D04 D03 D02 D01 D00
2rd
D07 D06 D05 D04 D03 D02 D01 D00
3th
D01 D00

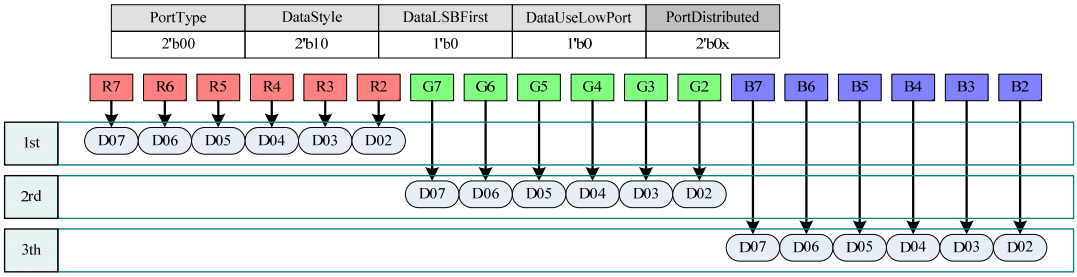

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b0
1'b1
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10
2rd
D17 D16 D15 D14 D13 D12 D11 D10
3th
D11 D10

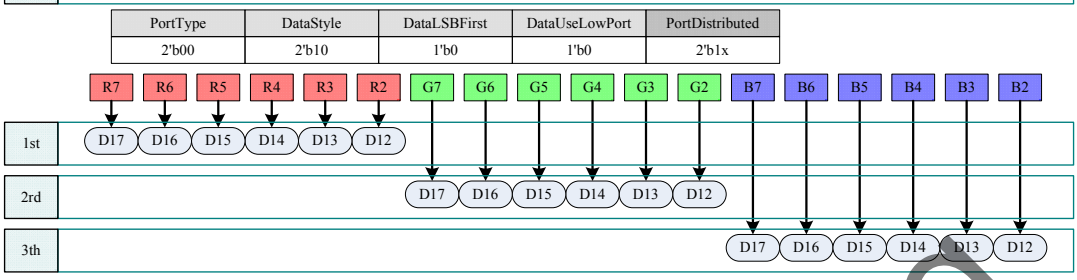

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b1
1'b0
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D07 D06 D05 D04 D03 D02 D01 D00
2rd
D07 D06 D05 D04 D03 D02 D01 D00
3th
D07 D06

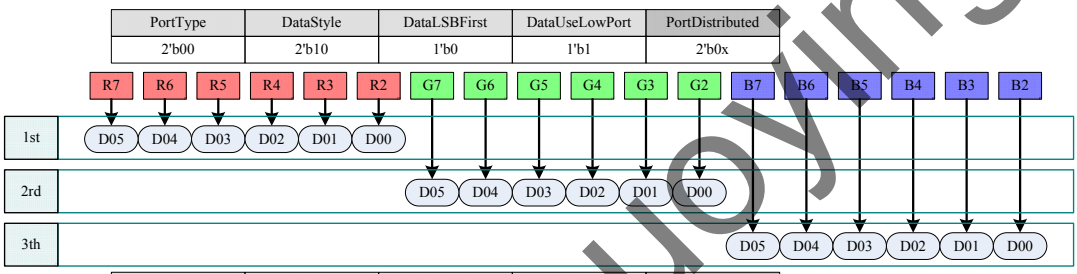

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b1
1'b0
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10
2rd
D17 D16 D15 D14 D13 D12 D11 D10
3th
D17 D16

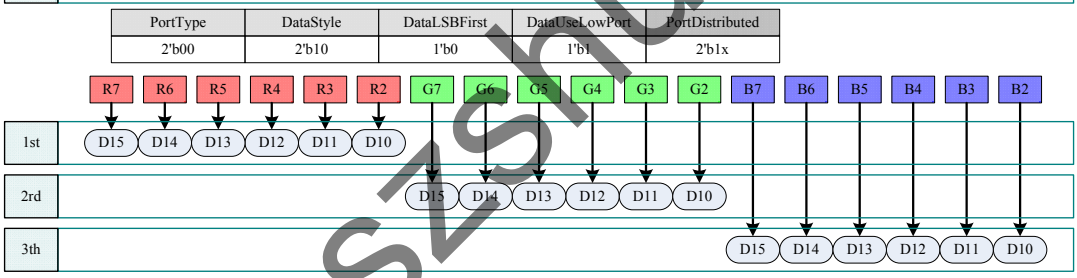

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b1
1'b1
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D07 D06 D05 D04 D03 D02 D01 D00
2rd
D07 D06 D05 D04 D03 D02 D01 D00
3th
D01 D00

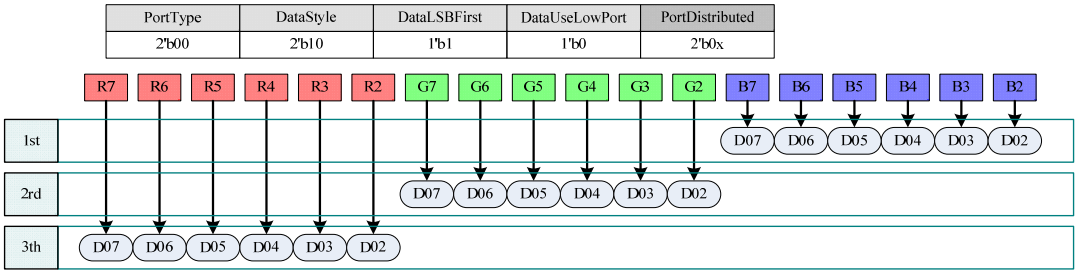

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b01
1'b1
1'b1
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10
2rd
D17 D16 D15 D14 D13 D12 D11 D10
3th
D11 D10

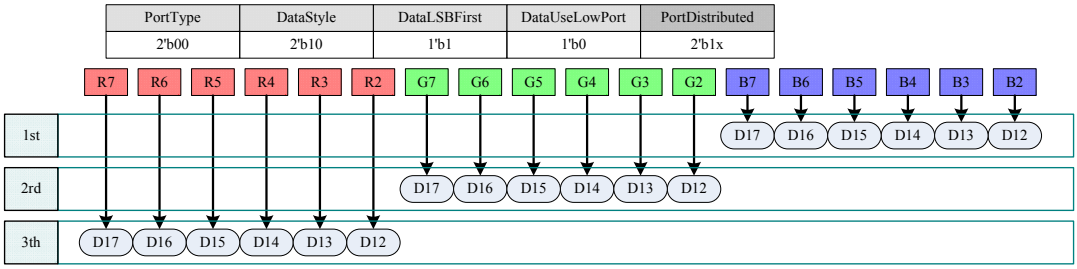

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b0
1'b0
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D07 D06 D05 D04 D03 D02
2rd
D07 D06 D05 D04 D03 D02
3th
D07 D06 D05 D04 D03 D02

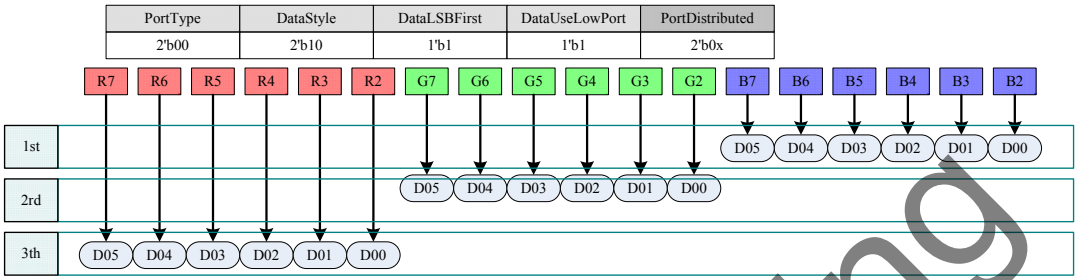

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b0
1'b0
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12
2rd
D17 D16 D15 D14 D13 D12
3th
D17 D16 D15 D14 D13 D12

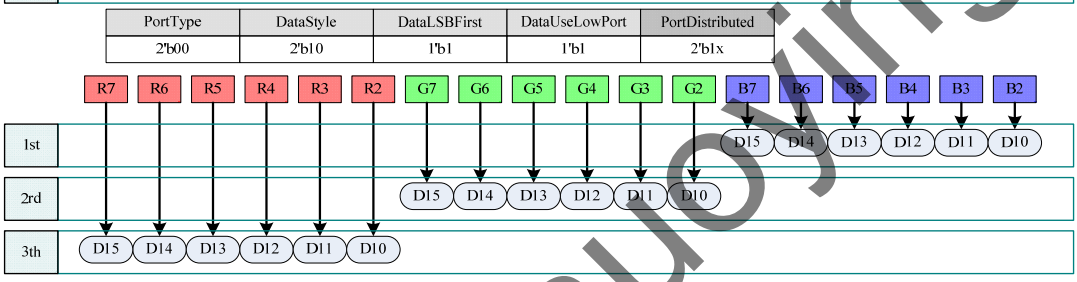

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b0
1'b1
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D05 D04 D03 D02 D01 D00
2rd
D05 D04 D03 D02 D01 D00
3th
D05 D04 D03 D02 D01 D00

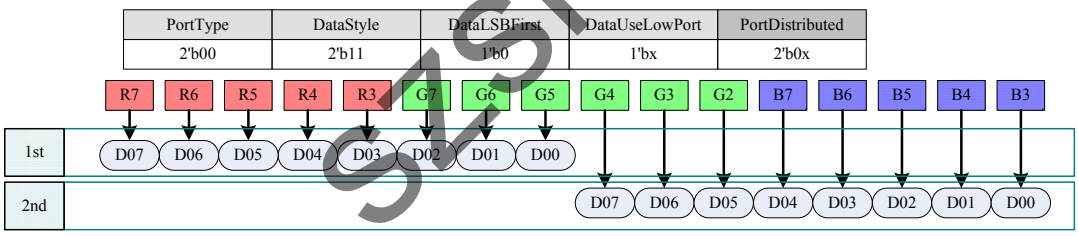

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b0
1'b1
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14 D13 D12 D11 D10
2rd
D15 D14 D13 D12 D11 D10
3th
D15 D14 D13 D12 D11 D10

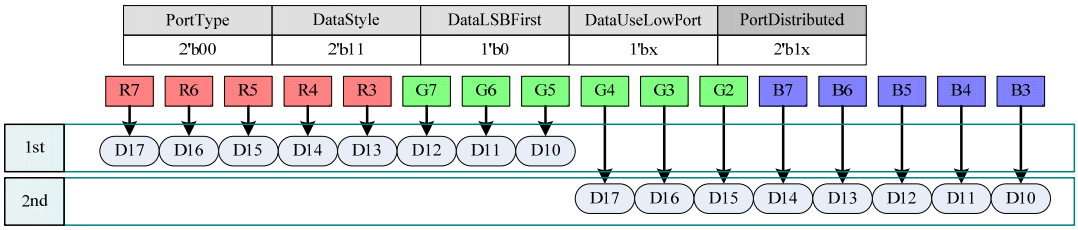

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b1
1'b0
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D07 D06 D05 D04 D03 D02
2rd
D07 D06 D05 D04 D03 D02
3th
D07 D06 D05 D04 D03 D02

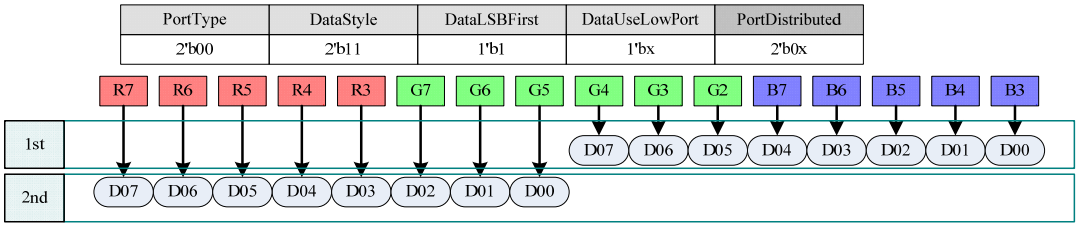

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b1
1'b0
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12
2rd
D17 D16 D15 D14 D13 D12
3th
D17 D16 D15 D14 D13 D12

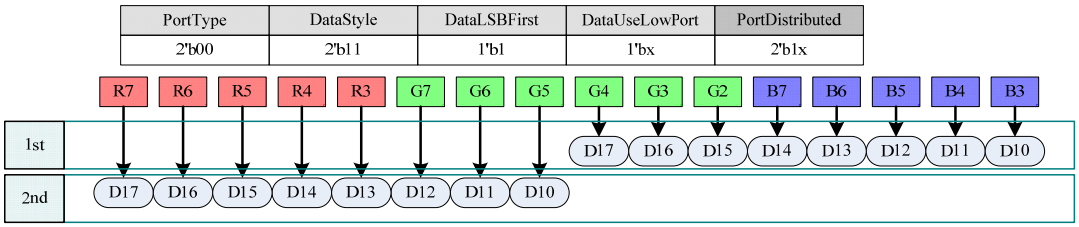

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b1
1'b1
2'b0x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D05 D04 D03 D02 D01 D00
2rd
D05 D04 D03 D02 D01 D00
3th
D05 D04 D03 D02 D01 D00


PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b10
1'b1
1'b1
2'b1x
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14 D13 D12 D11 D10
2rd
D15 D14 D13 D12 D11 D10
3th
D15 D14 D13 D12 D11 D10

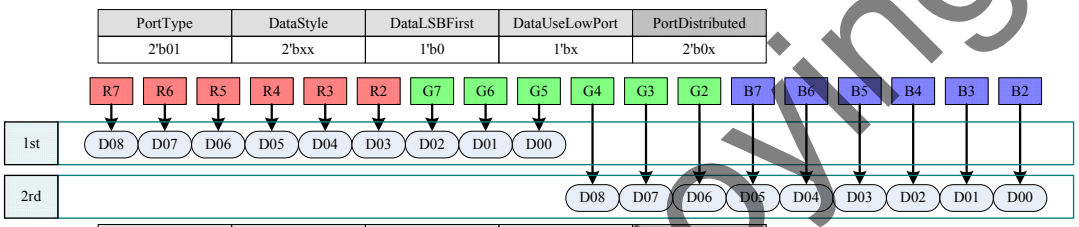

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b11
1'b0
1'bx
2'b0x
R7 R6 R5 R4 R3 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3
1st
D07 D06 D05 D04 D03 D02 D01 D00
2nd
D07 D06 D05 D04 D03 D02 D01 D00

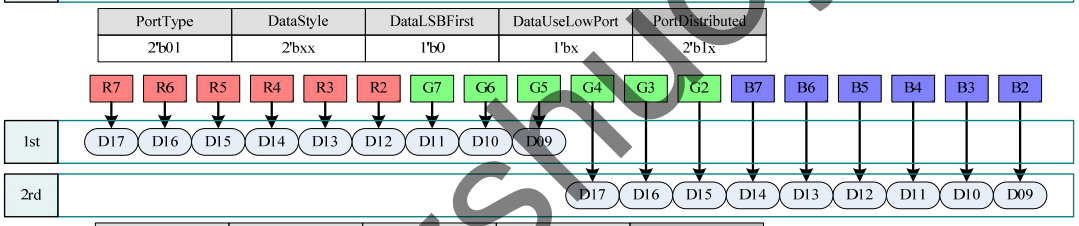

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b00
2'b11
1'b0
1'bx
2'b1x
R7 R6 R5 R4 R3 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3
1st
D17 D16 D15 D14 D13 D12 D11 D10
2nd
D17 D16 D15 D14 D13 D12 D11 D10

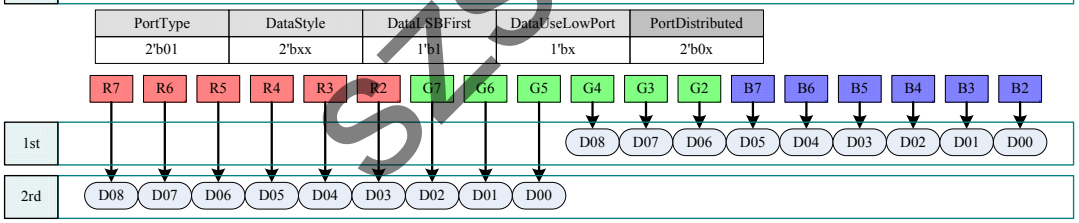


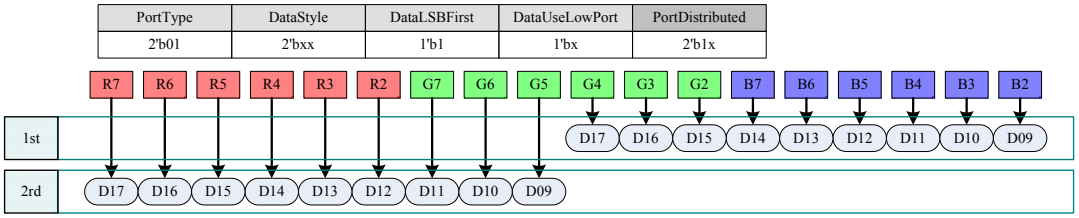


9-Wire Interface



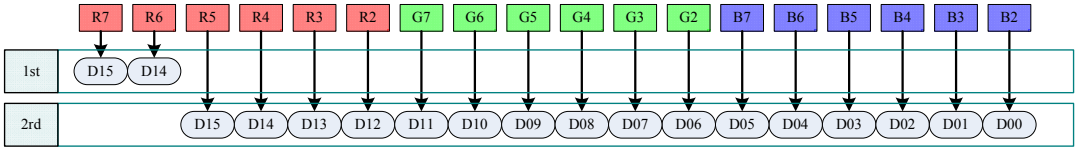




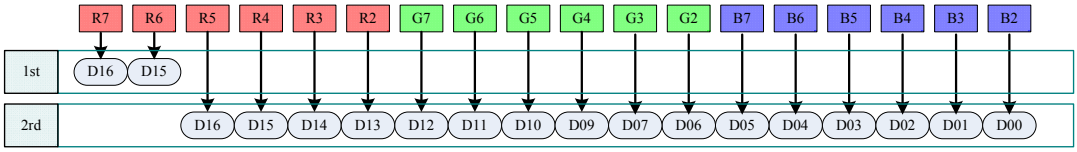


16-Wire Interface

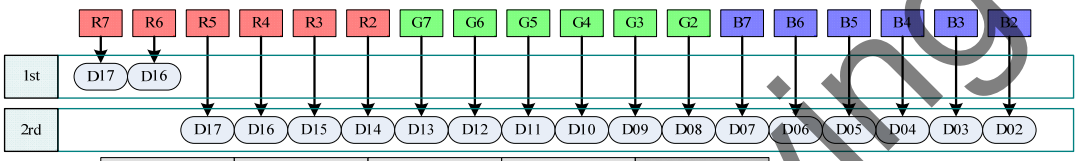
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b0	2'b00



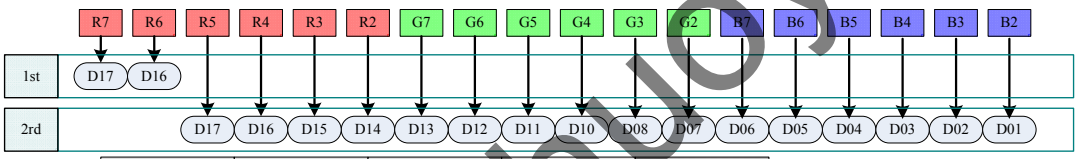
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b0	2'b01



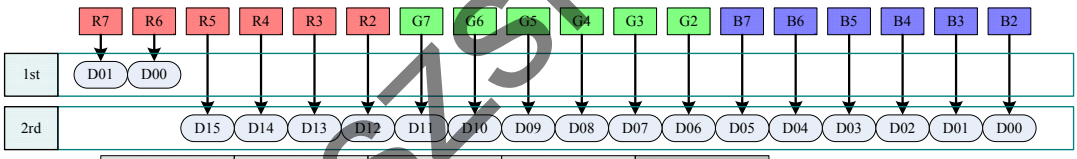
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b0	2'b10



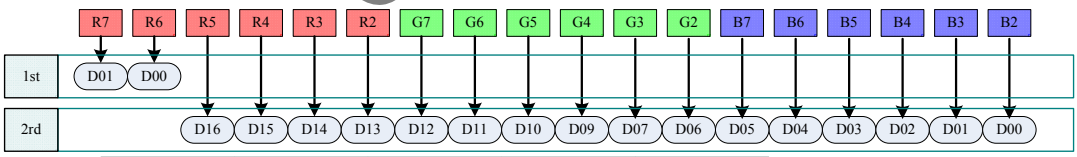
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b0	2'b11



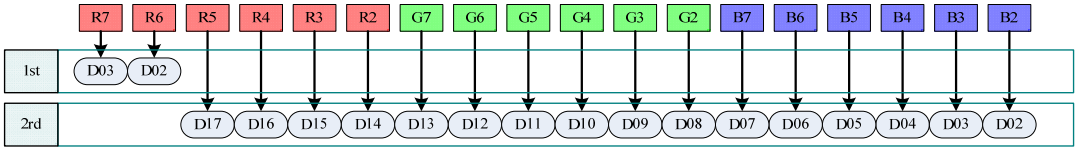
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b1	2'b00

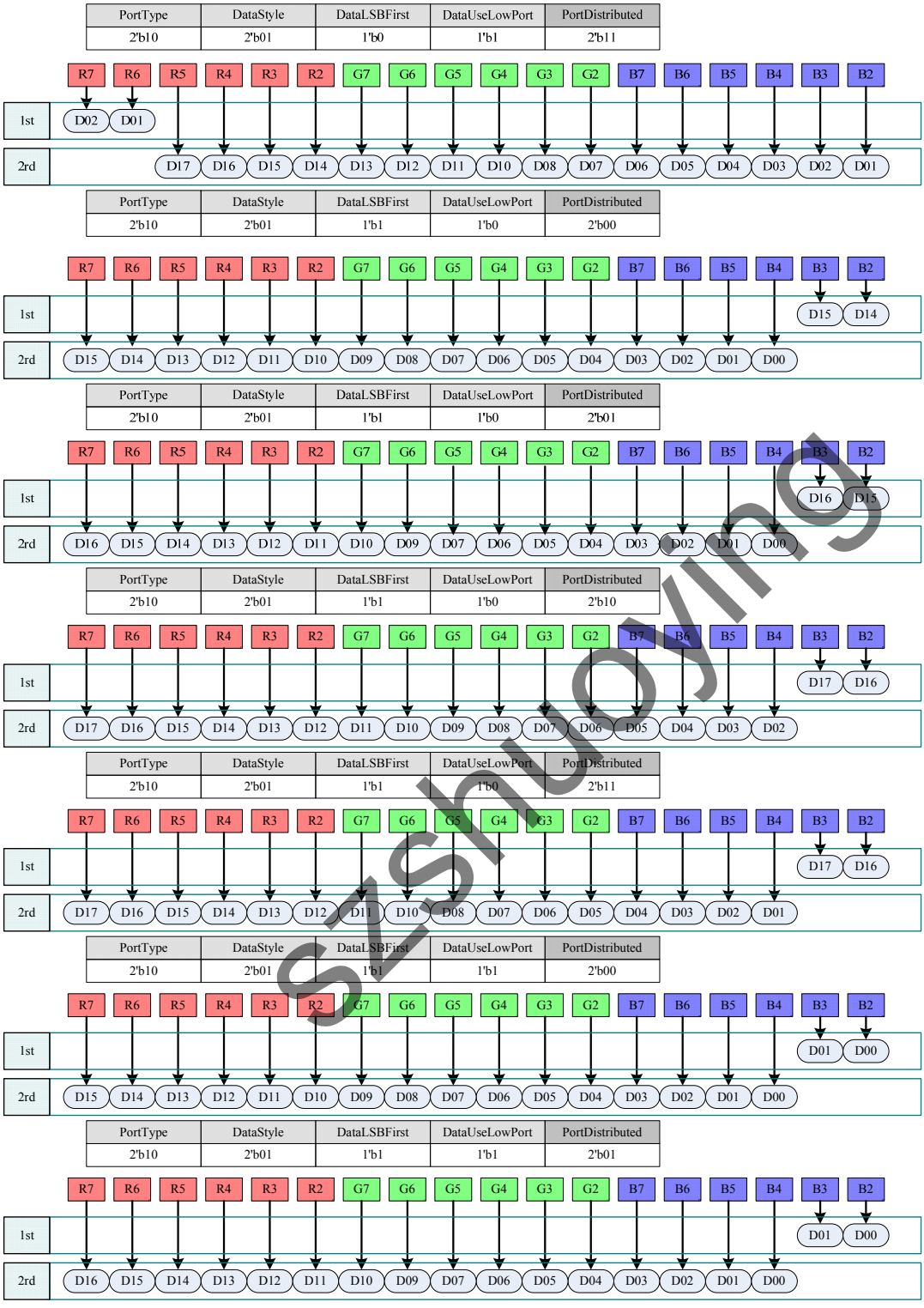


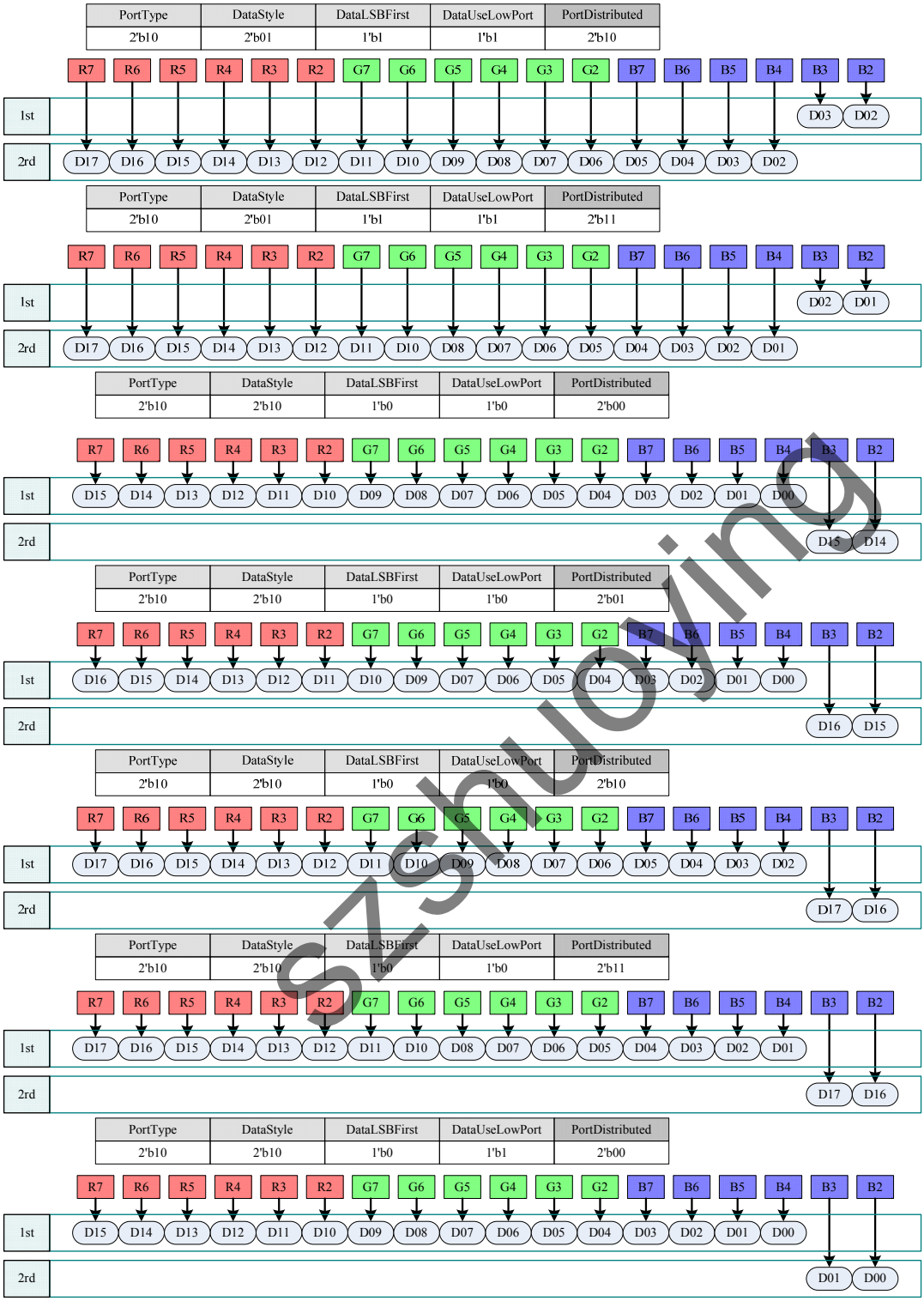
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b1	2'b01

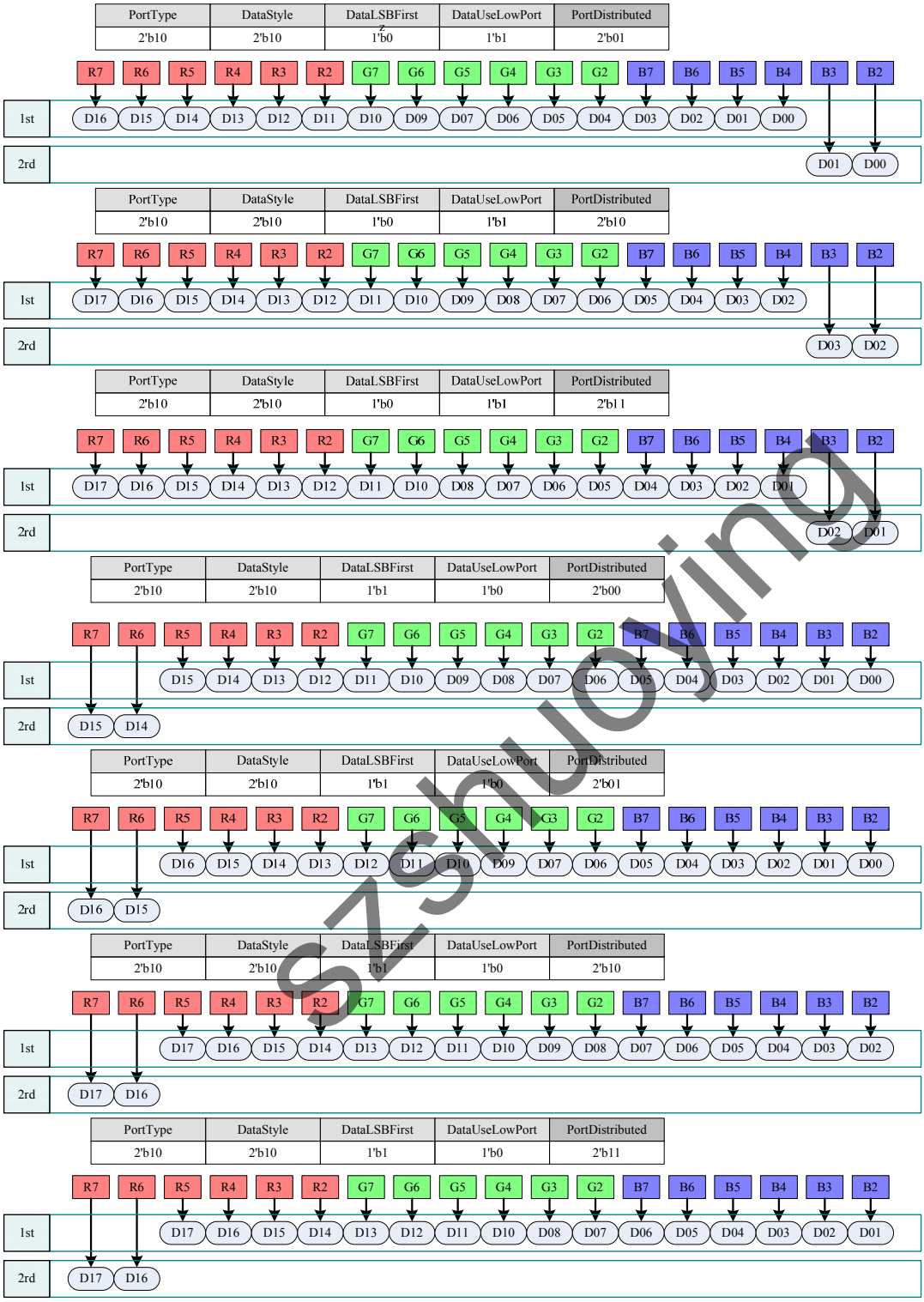


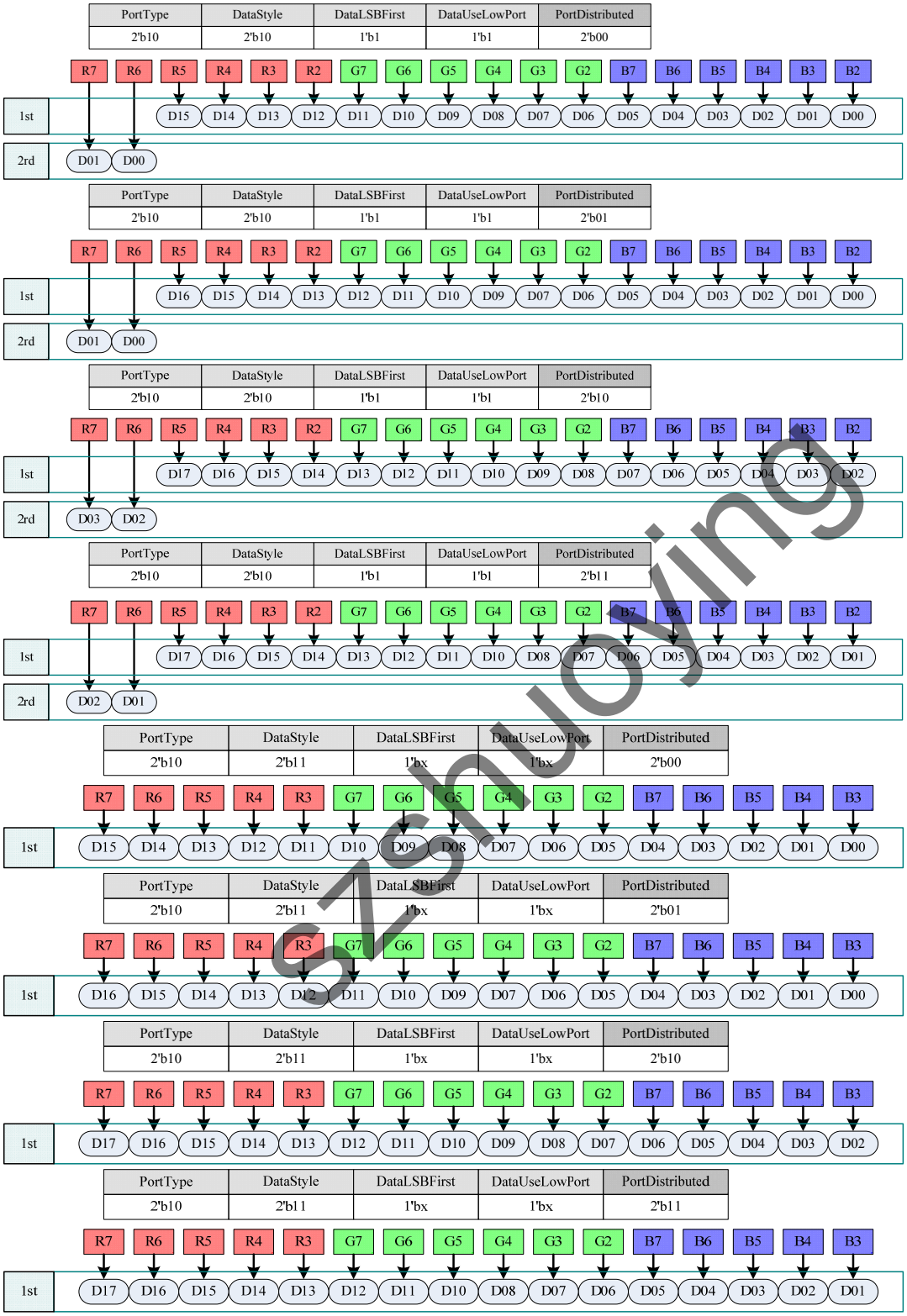
PortType	DataStyle	DataLSBFirst	DataUseLowPort	PortDistributed
2'b10	2'b01	1'b0	1'b1	2'b10



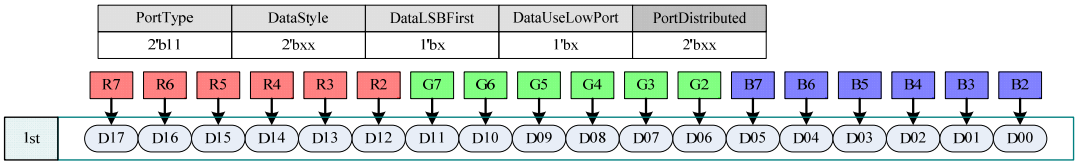

PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b0
1'b1
2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D02 D01
2rd
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b0
2'b00
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14
2rd
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b0
2'b01
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D16 D15
2rd
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b0
2'b10
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16
2rd
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b0
2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16
2rd
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b1
2'b00
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D01 D00
2rd
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b1
2'b01
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D01 D00
2rd
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00


PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b1
2'b10
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D03 D02
2rd
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b01
1'b1
1'b1
2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D02 D01
2rd
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b0
2'b00
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D15 D14
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b0
2'b01
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D16 D15
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b0
2'b10
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
2rd
D17 D16
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b0
2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
2rd
D17 D16
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b1
2'b00
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D01 D00


PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b1
2'b01
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D01 D00
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b1
2'b10
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
2rd
D03 D02
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b0
1'b1
2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
2rd
D02 D01
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b1
1'b0
2'b00
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D15 D14
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b1
1'b0
2'b01
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D16 D15
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b1
1'b0
2'b10
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
2rd
D17 D16
PortType
DataStyle
DataLSBFirst
DataUseLowPort
PortDistributed
2'b10
2'b10
1'b1
1'b0
2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
2rd
D17 D16


PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b10 | 1'b1 | 1'b1 | 2'b00
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D01 D00
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b10 | 1'b1 | 1'b1 | 2'b01
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00
2rd
D01 D00
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b10 | 1'b1 | 1'b1 | 2'b10
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
2rd
D03 D02
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b10 | 1'b1 | 1'b1 | 2'b11
R7 R6 R5 R4 R3 R2 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3 B2
1st
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01
2rd
D02 D01
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b11 | 1'bx | 1'bx | 2'b00
R7 R6 R5 R4 R3 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3
1st
D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b11 | 1'bx | 1'bx | 2'b01
R7 R6 R5 R4 R3 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3
1st
D16 D15 D14 D13 D12 D11 D10 D09 D07 D06 D05 D04 D03 D02 D01 D00
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b11 | 1'bx | 1'bx | 2'b10
R7 R6 R5 R4 R3 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3
1st
D17 D16 D15 D14 D13 D12 D11 D10 D09 D08 D07 D06 D05 D04 D03 D02
PortType | DataStyle | DataLSBFirst | DataUseLowPort | PortDistributed
2'b10 | 2'b11 | 1'bx | 1'bx | 2'b11
R7 R6 R5 R4 R3 G7 G6 G5 G4 G3 G2 B7 B6 B5 B4 B3
1st
D17 D16 D15 D14 D13 D12 D11 D10 D08 D07 D06 D05 D04 D03 D02 D01

18-Wire Interface



10.3 IDS Register Description

10.3.1 IDS Register Memory Map

Table 10-6: IDS Register Memory Map

Address	Symbol	Direction	Description
0x20CD_0000	LCDCON1	R/W	LCD control 1 Register
0x20CD_0004	LCDCON2	R/W	LCD control 2 Register
0x20CD_0008	LCDCON3	R/W	LCD control 3 Register
0x20CD_000C	LCDCON4	R/W	LCD control 4 Register
0x20CD_0010	LCDCON5	R/W	LCD control 5 Register
0x20CD_0030	LCDVCLKFSR	R/W	LCD VCLK Frequency Status Register
0x20CD_0054	IDSINTPND	R/W	IDS interrupt pending
0x20CD_0058	IDSSRCPND	R/W	IDS interrupt source
0x20CD_005C	IDSINTMSK	R/W	IDS interrupt mask
0x20CD_1000	OVCDCR	R/W	Overlay Controller Display Control Register
0x20CD_1004	OVCPCR	R/W	Overlay Controller Palette Control Register
0x20CD_1008	OVCBKCOLOR	R/W	Overlay Controller Background Color Register.
0x20CD_1080	OVCW0CR	R/W	Overlay Controller Window 0 Control Register
0x20CD_1084	OVCW0PCAR	R/W	Overlay Controller Window 0 Position Control A Register
0x20CD_1088	OVCW0PCBR	R/W	Overlay Controller Window 0 Position Control B Register
0x20CD_108C	OVCW0B0SAR	R/W	Overlay Controller Window 0 Buffer 0 Start Address Register
0x20CD_1090	OVCW0B1SAR	R/W	Overlay Controller Window 0 Buffer 1 Start Address Register
0x20CD_1094	OVCW0VSSR	R/W	Overlay Controller Window 0 Virtual Screen Size Register
0x20CD_1098	OVCW0CMR	R/W	Overlay Controller Window 0 Color MAP Register
0x20CD_109C	OVCW0B2SAR	R/W	Overlay Controller Window 0 Buffer 2 Start Address Register
0x20CD_10A0	OVCW0B3SAR	R/W	Overlay Controller Window 0 Buffer 3 Start Address Register
0x20CD_1100	OVCW1CR	R/W	Overlay Controller Window 1 Control Register
0x20CD_1104	OVCW1PCAR	R/W	Overlay Controller Window 1 Position Control A Register
0x20CD_1108	OVCW1PCBR	R/W	Overlay Controller Window 1 Position Control B Register
0x20CD_110C	OVCW1PCCR	R/W	Overlay Controller Window 1 Position Control C Register
0x20CD_1110	OVCW1B0SAR	R/W	Overlay Controller Window 1 Buffer 0 Start Address Register
0x20CD_1114	OVCW1B1SAR	R/W	Overlay Controller Window 1 Buffer 1 Start Address Register
0x20CD_1118	OVCW1BSR	R/W	Overlay Controller Window 1 Buffer Size Register
0x20CD_111C	OVCW1CKCR	R/W	Overlay Controller Window 1 Color Key Control Register
0x20CD_1120	OVCW1CKR	R/W	Overlay Controller Window 1 Color Key Register
0x20CD_1124	OVCW1CM	R/W	Overlay Controller Window 1 Color MAP
0x20CD_1128	OVCW1B2SAR	R/W	Overlay Controller Window 1 Buffer 2 Start Address Register
0x20CD_112C	OVCW1B3SAR	R/W	Overlay Controller Window 1 Buffer 3 Start Address Register
0x20CD_1180	OVCW2CR	R/W	Overlay Controller Window 2 Control Register
0x20CD_1184	OVCW2PCAR	R/W	Overlay Controller Window 2 Position Control A Register
0x20CD_1188	OVCW2PCBR	R/W	Overlay Controller Window 2 Position Control B Register
0x20CD_118C	OVCW2PCCR	R/W	Overlay Controller Window 2 Position Control C Register

0x20CD_1190	OVCW2B0SAR	R/W	Overlay Controller Window 2 Buffer0 Start Address Register
0x20CD_1194	OVCW2B1SAR	R/W	Overlay Controller Window 2 Buffer1 Start Address Register
0x20CD_1198	OVCW2BSR	R/W	Overlay Controller Window 2 Buffer Size Register
0x20CD_119C	OVCW2CKCR	R/W	Overlay Controller Window 2 Color Key Control Register
0x20CD_11A0	OVCW2CKR	R/W	Overlay Controller Window 2 Color Key Register
0x20CD_11A4	OVCW2CM	R/W	Overlay Controller Window 2 Color MAP
0x20CD_11A8	OVCW2B2SAR	R/W	Overlay Controller Window 2 Buffer2 Start Address Register
0x20CD_11AC	OVCW2B3SAR	R/W	Overlay Controller Window 2 Buffer3 Start Address Register
0x20CD_1200	OVCW3CR	R/W	Overlay Controller Window 3 Control Register
0x20CD_1204	OVCW3PCAR	R/W	Overlay Controller Window 3 Position Control A Register
0x20CD_1208	OVCW3PCBR	R/W	Overlay Controller Window 3 Position Control B Register
0x20CD_120C	OVCW3PCCR	R/W	Overlay Controller Window 3 Position Control C Register
0x20CD_1210	OVCW3BSAR	R/W	Overlay Controller Window 3 Buffer Start Address Register
0x20CD_1214	OVCW3BSR	R/W	Overlay Controller Window 3 Buffer Size Register
0x20CD_1218	OVCW3CKCR	R/W	Overlay Controller Window 3 Color Key Control Register
0x20CD_121C	OVCW3CKR	R/W	Overlay Controller Window 3 Color Key Register
0x20CD_1220	OVCW3CM	R/W	Overlay Controller Window 3 Color MAP
0x20CD_1224	OVCW3SABSAR	R/W	Overlay Controller Window 3 Cursor AND Bitmap buffer start address
0x20CD_1300	OVCBRB0SAR	R/W	Overlay Controller CbCr Buffer 0 Start Address Register
0x20CD_1304	OVCBRB1SAR	R/W	Overlay Controller CbCr Buffer 1 Start Address Register
0x20CD_1308	OVCOEF11	R/W	Overlay Controller Color Matrix Coefficient11 Register
0x20CD_130C	OVCOEF12	R/W	Overlay Controller Color Matrix Coefficient12 Register
0x20CD_1310	OVCOEF13	R/W	Overlay Controller Color Matrix Coefficient13 Register
0x20CD_1314	OVCOEF21	R/W	Overlay Controller Color Matrix Coefficient21 Register
0x20CD_1318	OVCOEF22	R/W	Overlay Controller Color Matrix Coefficient22 Register
0x20CD_131C	OVCOEF23	R/W	Overlay Controller Color Matrix Coefficient23 Register
0x20CD_1320	OVCOEF31	R/W	Overlay Controller Color Matrix Coefficient31 Register
0x20CD_1324	OVCOEF32	R/W	Overlay Controller Color Matrix Coefficient32 Register
0x20CD_1328	OVCOEF33	R/W	Overlay Controller Color Matrix Coefficient33 Register
0x20CD_132c	OVCOMC	R/W	Overlay Controller Color Matrix Configure Register
0x20CD_1330	OVCBRB2SAR	R/W	Overlay Controller CbCr Buffer 2 Start Address Register
0x20CD_1334	OVCBRB3SAR	R/W	Overlay Controller CbCr Buffer 3 Start Address Register
0x20CD_1400-0x20CD_17FC	OVCW0PAL0-OVCW0PAL255	R/W	Overlay Controller Window 0 Palette RAM, 256 entries.
0x20CD_1800-0x20CD_1BFC	OVCW1PAL0-OVCW1PAL255	R/W	Overlay Controller Window 1 Palette RAM, 256 entries.
0x20CD_1C00-0x20CD_1C3C	OVCW2PAL0-OVCW2PAL15	R/W	Overlay Controller Window 2 Palette Register, 16 entries.
0x20CD_1E00-0x20CD_1E3C	OVCW3PAL0-OVCW3PAL15	R/W	Overlay Controller Window 3 Palette Register, 16 entries.
0x20CD_2000	TVCCR	R/W	TVIF Clock Control Register
0x20CD_2004	TVICR	R/W	TVIF Interface Control Register
0x20CD_2008	TVCOEF11	R/W	TVIF Color Space Convert Matrix Coefficient 11 Register
0x20CD_200C	TVCOEF12	R/W	TVIF Color Space Convert Matrix Coefficient 12 Register
0x20CD_2010	TVCOEF13	R/W	TVIF Color Space Convert Matrix Coefficient 13 Register
0x20CD_2014	TVCOEF21	R/W	TVIF Color Space Convert Matrix Coefficient 21 Register
0x20CD_2018	TVCOEF22	R/W	TVIF Color Space Convert Matrix Coefficient 22 Register
0x20CD_201C	TVCOEF23	R/W	TVIF Color Space Convert Matrix Coefficient 23 Register
0x20CD_2020	TVCOEF31	R/W	TVIF Color Space Convert Matrix Coefficient 31 Register
0x20CD_2024	TVCOEF32	R/W	TVIF Color Space Convert Matrix Coefficient 32 Register
0x20CD_2028	TVCOEF33	R/W	TVIF Color Space Convert Matrix Coefficient 33 Register
0x20CD_202C	TVCMCR	R/W	TVIF Color Matrix Configure Register

0x20CD_2030	TVUBA1	R/W	TVIF Upper Blanking Area 1 Line Register
0x20CD_2034	TVUNBA	R/W	TVIF Upper Non-blanking Area Line Register
0x20CD_2038	TVUBA2	R/W	TVIF Upper Blanking Area 2 Line Register
0x20CD_203C	TVLBA1	R/W	TVIF Lower Blanking Area 1 Line Register
0x20CD_2040	TVLNBA	R/W	TVIF Lower Non-blanking Area Line Register
0x20CD_2044	TVLBA2	R/W	TVIF Lower Blanking Area 2 Line Register
0x20CD_2048	TVBLEN	R/W	TVIF Line Blanking Length Register
0x20CD_204C	TVVLEN	R/W	TVIF Line Video Length Register
0x20CD_2050	TVHSCR	R/W	TVIF HSync Configure Register
0x20CD_2054	TVVSHCR	R/W	TVIF VSync Upper Configure Register
0x20CD_2058	TVVSLCR	R/W	TVIF VSync Lower Configure Register
0x20CD_205C	TVXSIZE	R/W	TVIF Display Horizontal Size Register
0x20CD_2060	TVYSIZE	R/W	TVIF Display Horizontal Size Register
0x20CD_2064	TVSTSR	R	TVIF Status Register
0x20CD_3000	I80TRIGCON	R/W	I80 Trigger Control/Status Register
0x20CD_3004	I80IFCON0	R/W	I80 Interface Control 0
0x20CD_3008	I80IFCON1	R/W	I80 Interface Control 1
0x20CD_300C	I80CMDCON0	R/W	I80 System Interface Command Control 0
0x20CD_3010	I80CMDCON1	R/W	I80 System Interface Command Control 1
0x20CD_3014	I80CMD15	R/W	I80 Interface Command instruct 15
0x20CD_3018	I80CMD14	R/W	I80 Interface Command instruct 14
0x20CD_301C	I80CMD13	R/W	I80 Interface Command instruct 13
0x20CD_3020	I80CMD12	R/W	I80 Interface Command instruct 12
0x20CD_3024	I80CMD11	R/W	I80 Interface Command instruct 11
0x20CD_3028	I80CMD10	R/W	I80 Interface Command instruct 10
0x20CD_302C	I80CMD09	R/W	I80 Interface Command instruct 9
0x20CD_3030	I80CMD08	R/W	I80 Interface Command instruct 8
0x20CD_3034	I80CMD07	R/W	I80 Interface Command instruct 7
0x20CD_3038	I80CMD06	R/W	I80 Interface Command instruct 6
0x20CD_33C	I80CMD05	R/W	I80 Interface Command instruct 5
0x20CD_3040	I80CMD04	R/W	I80 Interface Command instruct 4
0x20CD_3044	I80CMD03	R/W	I80 Interface Command instruct 3
0x20CD_3048	I80CMD02	R/W	I80 Interface Command instruct 2
0x20CD_304C	I80CMD01	R/W	I80 Interface Command instruct 1
0x20CD_3050	I80CMD00	R/W	I80 Interface Command instruct 0
0x20CD_3054	I80MANCON	R/W	I80 System Interface Manual Command Control
0x20CD_3058	I80MANWDAT	R/W	I80 System Interface Manual Command Data Write Buffer
0x20CD_305C	I80MANRDAT	R/W	I80 System Interface Manual Command Data Read Buffer

10.3.2 IDS Registers and Field Descriptions

Register 10-1: LCD Control 1 Register （LCDCON1,offset=0x0000）

Field	Symbol	Direction	Description	Default
[28:18]	LINECNT	R	Provide the status of the line counter. Down count from LINEVAL to 0	0x0
[17:8]	CLKVAL	R/W	Determine the rates of VCLK and CLKVAL[9:0]. **TFT**: VCLK = IDSTF_CLK / (CLKVAL+1)	0x0
[7]	Reserved	R/W	Must be always set to 0	0x0

[6:5]	Reserved	R	Reserved	0x3
[4:1]	Reserved	R/W	Reserved	0
[0]	ENVID	R/W	LCD video output and the logic enable/disable. 0 = Disable the video output and the LCD control signal. 1 = Enable the video output and the LCD control signal.	0

Register 10-2: LCD Control 2 Register （LCDCON2,offset=0x0004）

Field	Symbol	Direction	Description	Default
[31:24]	VBPD	R/W	**TFT**: Vertical back porch is the number of inactive lines at the start of a frame, after vertical synchronization period.	0
[23:14]	LINEVAL[9:0]	R/W	**TFT**: These bits determine the vertical size of LCD panel. **Note**: In this field, only low 10 bits is set, LINEVAL[10] is set in bit 31 of register LCDCON3	0
[13:6]	VFPD	R/W	**TFT**: Vertical front porch is the number of inactive lines at the end of a frame, before vertical synchronization period.	0
[5:0]	VSPW	R/W	**TFT**: Vertical sync pulse width determines the VSYNC pulse's high level width by counting the number of inactive lines.	0

Register 10-3: LCD Control 3 Register （LCDCON3,offset=0x0008）

Field	Symbol	Direction	Description	Default
[31]	LINEVAL[10]	R/W	**TFT**: This bit determine the vertical size of LCD panel.	0
[25:19]	HBPD	R/W	**TFT**: Horizontal back porch is the number of VCLK periods between the falling edge of HSYNC and the start of active data.	0
[17:8]	HOZVAL	R/W	**TFT**: These bits determine the horizontal size of LCD panel. HOZVAL has to be determined to meet the condition that total bytes of 1 line are 4n bytes. If the x size of LCD is 120 dot in mono mode, x=120 cannot be supported because 1 line consists of 15 bytes. Instead, x=128 in mono mode can be supported because 1 line is composed of 16 bytes (2n). LCD panel driver will discard the additional 8 dot.	0
[7:0]	HFPD	R/W	**TFT**: Horizontal front porch is the number of VCLK periods between the end of active data and the rising edge of HSYNC.	0

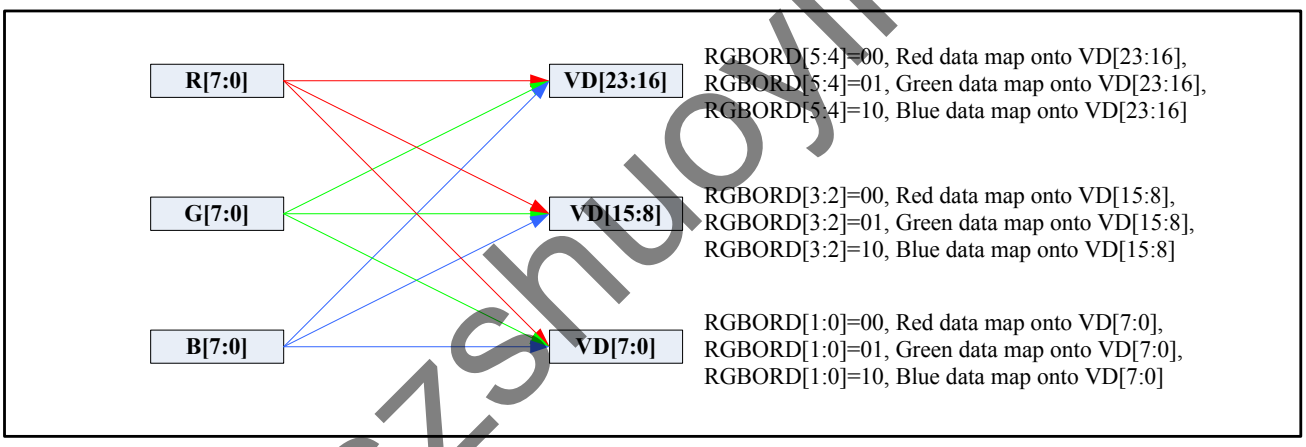
Register 10-4: LCD Control 4 Register （LCDCON4,offset=0x0010）

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	R	Reserved	0
[7:0]	HSPW	R/W	**TFT**: Horizontal sync pulse width determines the HSYNC pulse's high level width by counting the number of the VCLK.	0

Register 10-5: LCD Control 5 Register （LCDCON5,offset=0x0014）

Field	Symbol	Direction	Description	Default
[29:24]	RGBORD	R/W	RGB output order of TFT LCD display interface. **Note** : Please refer to the following figure for details.	0x6
[16:15]	VSTATUS	R	**TFT**: Vertical Status 00 = VSYNC 01 = BACK Porch 10 = ACTIVE 11 = FRONT Porch	0
[14:13]	HSTATUS	R	TFT: Horizontal Status. 00 = HSYNC 01 = BACK Porch	0

			10 = ACTIVE 11 = FRONT Porch	
[12:11]	DSPTYPE	R/W	External Display type indication signal. 00 : TFT LCD display 01 : I80 LCD display 10 : TV display	
[10]	INVVCLK	R/W	**TFT**: This bit controls the polarity of the VCLK active edge. 0 = The video data is fetched at VCLK falling edge 1 = The video data is fetched at VCLK rising edge	0
[9]	INVHSYNC	R/W	**TFT**: This bit indicates the HSYNC pulse polarity. 0 = Normal 1 = Inverted	0
[8]	INVVSYNC	R/W	**TFT**: This bit indicates the VSYNC pulse polarity. 0 = Normal 1 = Inverted	0
[7]	INVVD	R/W	**TFT**: This bit indicates the VD (video data) pulse polarity. 0 = Normal 1 = VD is inverted.	0
[6]	INVVDEN	R/W	**TFT**: This bit indicates the VDEN signal polarity. 0 = normal 1 = inverted	0
[5]	INVPWREN	R/W	**TFT**: This bit indicates the PWREN signal polarity. 0 = normal 1 = inverted	0
[3]	PWREN	R/W	**TFT**: LCD_PWREN output signal enable/disable. 0 = Disable PWREN signal 1 = Enable PWREN signal	0
[2:0]	Reserved	R/W	Reserved	0



Register 10-6: LCD VCLK Frequency Status Register （LCDVCLKFSR,offset=0x0030）

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	R	Reserved	0
[27:24]	CDOWN	R/W	Frequency conversion factor, if continuous CDOWN frames is unsatisfy. Note: If set this field 0xf, The auto frequency conversion will be disabled.	0
[23:22]	Reserved	R	Reserved	0
[21:16]	RFRM_NUM	R/W	Frame number for restore clock divider. When continuous RFRM_NUM frame is satisfy, internal VCLK will be restored.	0x3c
[15:14]	Reserved	R	Reserved	0
[13:4]	CLKVAL	R	Current VCLK divisor. When system bus can't satisfy the LCD refresh rate, display controller will auto increase internal VCLK divisor and set flag in VCLKFAC	0

[3:1]	Reserved	R	Reserved	0
[0]	VCLKFAC	R/C	VCLK frequency auto change flag. When system bus can't satisfy the LCD refresh rate, display controller will auto increase internal VCLK divisor and set 1 to this bit. Write 1 to clear this bit.	0

Register 10-7: IDS Interrupt Pending Register （IDSINTPND,offset=0x0054）

Field	Symbol	Direction	Description	Default
[6]	OSDW2INT	R/C	OSD Win2 Timeslot synchronized interrupt pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[5]	I80INT	R/C	i80 transmission end or control commond sequence end interrupt pending bit. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[4]	OSDERR	R/C	OSD error interrupt pending bit, write 1 to clear this flag. When OSD occur error, need to reset OSD. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[3]	OSDW1INT	R/C	OSD Win1 Timeslot synchronized interrupt pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[2]	OSDW0INT	R/C	OSD Win0 Timeslot synchronized interrupt pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[1]	VCLKINT	R/C	VCLK frequency auto change interrupt pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[0]	LCDINT	R/C	LCD frame synchronized interrupt pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0

Register 10-8: IDS Source Pending Register （IDSSRCPND,0x0058）

Field	Symbol	Direction	Description	Default
[6]	OSDW2INT	R/C	OSD Win2 Timeslot synchronized source pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	
[5]	I80INT	R/C	i80 transmission end or control commond sequence end source pending bit. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[4]	OSDERR	R/C	OSD error interrupt source pending bit, write 1 to clear this flag. When OSD occur error, need to reset OSD. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0

[3]	OSDW1INT	R/C	OSD Win1 Timeslot synchronized interrupt source pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[2]	OSDW0INT	R/C	OSD Win0 Timeslot synchronized interrupt source pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[1]	VCLKINT	R/C	VCLK frequency auto change interrupt source pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0
[0]	LCDINT	R/C	LCD frame synchronized interrupt source pending bit, write 1 to clear this flag. 0 = The interrupt has not been requested. 1 = The frame has asserted the interrupt request.	0

Register 10-9: IDS Interrupt Mask Register （IDSINTMSK,offset=0x005C）

Field	Symbol	Direction	Description	Default
[6]	OSDW2INTMSK	WO	OSD Win2 Timeslot synchronized interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1
[5]	I80INTMSK	R/W	i80 transmission end or control commond sequence end interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1
[4]	OSDERRMSK	R/W	OSD error interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1
[3]	OSDW1INTMSK	R/W	OSD Win1 Timeslot synchronized interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1
[2]	OSDW0INTMSK	R/W	OSD Win0 Timeslot synchronized interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1
[1]	VCLKINTMSK	R/W	VCLK frequency auto change interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1
[0]	LCDINTMSK	R/W	LCD frame synchronized interrupt mask. 0 = The interrupt service is available. 1 = The interrupt service is masked	1

Register 10-10: Overlay Controller Display Control Register （OVCDCR,offset=0x1000）

Field	Symbol	Direction	Description	Default
[31:22]	Reserved	R	Reserved	0
[21:20]	W2_SYN_SEL	R/W	Windows 2 video synchronized source selection. 1 : camera interface C channel synchronized tigger 2 : camera interface preview channel synchronized tigger Others : pwm timer 2 synchronized tigger	0
[19:18]	W1_SYN_SEL	R/W	Windows 1 video synchronized source selection. 1 : camera interface C channel synchronized tigger 2 : camera interface preview channel synchronized tigger	0

			Others : pwm timer 1 synchronized tigger	
[17:16]	W0_SYN_SEL	R/W	Windows 0 video synchronized source selection. 1 : camera interface C channel synchronized tigger 2 : camera interface preview channel synchronized tigger Others : pwm timer 0 synchronized tigger	0
[15:10]	Reserved	R	Reserved	0
[9]	Enrefetch	R/W	Enable refetch image data when DMA occur error. 0:disable 1:emable	0
[8]	Enrelax	R/W	Enable fetch masked data when the masked image data less than 4 DWORD.	0
[7:4]	WaitTime	R/W	DMA error time level. When DMA occur error time reach WaitTime and Enrefetch set to 1, will refetch image data.	0
[3]	SafeBand	R/W	Configure bit for Window blending. 0: No extra data be fetched 1: Extra data will be fetched even after all data for a window has been fetched back.	0
[2]	AllFetch	R/W	Configure bit for Window blending. 0: do not fetch data for window area that is masked by the above window. 1: fetch data for all window area, even though they are partial masked.	0
[1]	IFTYPE	R/W	It determines the output format of IDS Controller. 0: RGB Parallel I/F 1: ITU-R BT601/656	0
[0]	INTERLACE	R/W	Interlace or Progressive for TVIF 0: Progressive, 1: Interlace **Note**: If IFTYPE is set to 0 and INTERLACE must be set to 0.	0

Register 10-11: Overlay Controller Palette Control Register （OVCPCR,offset=0x1004）

Field	Symbol	Direction	Description	Default
[15]	UPDATE_PAL	R/W	0: Normal Mode 1: Enable to update palette memory or registers. **Note**: If you want to update palette memory or registers, you should firstly set 1 to this bit and ensure that controller is disabled.	0
[11:9]	W3PALFM	R/W	This bit determines the palette data format of Window 3 palette. Note that Window 3 palette is 16 entries 16-bit Register array and only the following two formats are supported. 101 = 16bit for ARGB555(A:R5:G5:B5) 110 = 16bit for RGB565(R5:G6:B5)	0
[8:6]	W2PALFM	R/W	This bit determines the palette data format of Window 2 palette. Note that Window 2 palette is 16 entries 16-bit Register array and only the following two formats are supported. 101 = 16bit for ARGB555(A:R5:G5:B5) 110 = 16bit for RGB565(R5:G6:B5)	0
[5:3]	W1PALFM	R/W	This bit determines the palette data format of Window 1 palette. Note that Window 1 palette is 256 entries 25-bit memory and the following formats are supported. 000 = 25bit for ARGB888 (A:R8:G8:B8) 001 = 24 bit for RGB888 (R8:G8:B8) 010 = 19bit for ARGB666 (A:R6:G6:B6) 011 = 18 bit for RGB665 (A:R6:G6:B5)	0

			100 = 18 bit for RGB666 (R6:G6:B6) 101 = 16 bit for ARGB555 (A:R5:G5:B5) 110 = 16 bit for RGB565 (R5:G6:B5)	
[2:0]	W0PALFM	R/W	This bit determines the palette data format of Window 0 palette. Note that Window 0 palette is 256 entries 25-bit memory and the following formats are supported. 000 = 25bit for ARGB888 (A:R8:G8:B8) 001 = 24 bit for RGB888 (R8:G8:B8) 010 = 19bit for ARGB666 (A:R6:G6:B6) 011 = 18 bit for RGB665 (A:R6:G6:B5) 100 = 18 bit for RGB666 (R6:G6:B6) 101 = 16 bit for ARGB555 (A:R5:G5:B5) 110 = 16 bit for RGB565 (R5:G6:B5)	0

Register 10-12: Overlay Controller Background Color Register （OVCBKCOLOR,offset=0x1008）

Field	Symbol	Direction	Description	Default
[23:0]	BKCOLOR	R/W	Background Color. BKCOLOR[23:16] : RED BKCOLOR[15:8] : GREEN BKCOLOR[7:0] : BLUE	0

Register 10-13: Overlay Controller Window 0 Control Register （OVCW0CR,offset=0x1080）

Field	Symbol	Direction	Description	Default
[21]	FIELDSTATUS	R	TVIF Field Fetch Status (Read Only, valid for Interlace mode) 0 = Top Field 1 = Bottom Field	0
[20:19]	BUFSTATUS	R	Window Fetch Buffer Status (Read only, valid when auto ping-pong buffer control bit is enabled.) 0 = fetch buffer 0 1 = fetch buffer 1	0
[18:17]	BUFSEL	R/W	When not use auto ping-pong buffer mode, this bit is used to set fetch buffer by software. 00: fetch buffer 0 01: fetch buffer 1 10: fetch buffer 2 11: fetch buffer 3	0
[16]	BUFAUTOEN	R/W	Auto Ping-pong Buffer control bit 0 = Fixed by BUFSEL 1 = Automatically changed by H/W control signal	0
[15:14]	BUF_NUM	R/W	Auto buffer numbers 00,01:two buffers 10: three buffers 11: four buffers	
[13]	Reserved	RO	Reserved	0
[12]	BITSWP	R/W	Bit in each two bits swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[11]	Bits2SWP	R/W	Two bits in a half byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[10]	Bits4SWP	R/W	Half-Byte swap control bit. 0 = Swap Disable	0

			1 = Swap Enable	
[9]	BYTSWP	R/W	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
[8]	HAWSWP	R/W	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[5:1]	BPPMODE	R/W	Select the BPP (Bits Per Pixel) mode of Window 0 image data. 00000 = 1 BPP (palletized) 00001 = 2 BPP (palletized) 00010 = 4 BPP (palletized) 00011 = 8 BPP (palletized) 00100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 00101 = 16 BPP (non-palletized, R:5-G:6-B:5) 00110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 00111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 01000 = unpacked 18 BPP (non-palletized,R:6-G:6-B:6) 01001 = unpacked 18 BPP (non-palletized,A:1-R:6-G:6-B:5) 01010 = unpacked 19 BPP (non-palletized,A:1-R:6-G:6-B:6) 01011 = unpacked 24 BPP (non-palletized R:8-G:8-B:8) 01100 = unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 01101 = unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 01110 = unpacked 28 BPP (non-palletized A:4-R:8-G:8-B:8) 01111 = unpacked 16 BPP (non-palletized A:4-R:4-G:4-B:4) 10000 = unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) 10001 = YCbCr 4:2:0 semi-planar others = Reserved	0
[0]	ENWIN	R/W	Window 0 enable bit. 0 = Disable 1 = Enable	0

Register 10-14: Overlay Controller Window 0 Position Control A Register （OVCW0PCAR,offset=0x1084）

Field	Symbol	Direction	Description	Default
[26:16]	LeftTopX	R/W	Horizontal screen coordinate for left top pixel of OSD image	0
[10:0]	LeftTopY	R/W	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

Register 10-15: Overlay Controller Window 0 Position Control B Register （OVCW0PCBR,offset=0x1088）

Field	Symbol	Direction	Description	Default
[26:16]	RightBotX	R/W	Horizontal screen coordinate for right bottom pixel of OSD image.	0
[10:0]	RightBotY	R/W	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

Register 10-16: Overlay Controller Window 0 Buffer 0 Start Address Register （OVCW0B0SAR,offset=0x108C）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR0	R/W	Start address of the Window 0's video frame buffer 0. **Note**: BUFADDR0 is the address of left top pixel of window 0 virtual screen in DRAM memory.	0

Register 10-17: Overlay Controller Window 0 Buffer 1 Start Address Register （OVCW0B1SAR,offset=0x1090）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR1	R/W	Start address of the Window 0's video frame buffer 1. **Note**: BUFADDR1 is the address of left top pixel of window 0 virtual screen in DRAM memory.	0

Register 10-18: Overlay Controller Window 0 Virtual Screen Size Register（OVCW0VSSR,offset=0x1094）

Field	Symbol	Direction	Description	Default
[30:28]	BUF3_BITADR	R/W	Start bit address of the Window 0's video frame buffer 3. Value from 0 to 7 Only valid at bppmode = 0,1,2,3	0
[26:24]	BUF2_BITADR	R/W	Start bit address of the Window 0's video frame buffer 2.	0
[23:20]	BUF1_BITADR	R/W	Start bit address of the Window 0's video frame buffer 1. Value from 0 to 7 Only valid at bppmode = 0,1,2,3	0
[19:16]	BUF0_BITADR	R/W	Start bit address of the Window 0's video frame buffer 0.	0
[15:0]	VW_WIDTH	R/W	Virtual screen page width in the unit: pixel	0

Register 10-19: Overlay Controller Window 0 Color Map Register （OVCW0CMR x1098）

Field	Symbol	Direction	Description	Default
[24]	MAPCOLEN	R/W	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
[23:0]	MAPCOLOR	R/W	Map Color. MAPCOLOR[23:16] : RED MAPCOLOR[15:8] : GREEN MAPCOLOR[7:0] : BLUE	0

Register 10-20: Overlay Controller Window 0 Buffer 2 Start Address Register （OVCW0B2SAR,offset=0x109C）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR2	R/W	Start address of the Window 0's video frame buffer 2. **Note**: BUFADDR2 is the address of left top pixel of window 0 virtual screen in DRAM memory.	0

Register 10-21: Overlay Controller Window 0 Buffer 3 Start Address Register （OVCW0B3SAR,offset=0x10A0）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR3	R/W	Start address of the Window 0's video frame buffer 3. **Note**: BUFADDR3 is the address of left top pixel of window 0 virtual screen in DRAM memory.	0

Register 10-22: Overlay Controller Window 1 Control Register （OVCW1CR,offset=0x1100）

Field	Symbol	Direction	Description	Default
[21]	FIELDSTATUS	R	TVIF Field Fetch Status (Read Only, valid for Interlace mode) 0 = Top Field 1 = Bottom Field	0
[20:19]	BUFSTATUS	R	Window Fetch Buffer Status (Read only, valid when auto ping-pong buffer control bit is enabled.)	0

			0 = fetch buffer 0 1 = fetch buffer 1	
[18:17]	BUFSEL	R/W	When not use auto ping-pong buffer mode, this bit is used to set fetch buffer by software. 00: fetch buffer 0 01: fetch buffer 1 10: fetch buffer 2 11: fetch buffer 3	
[16]	BUFAUTOEN	R/W	Auto Ping-pong Buffer control bit 0 = Fixed by BUFSEL 1 = Automatically changed by H/W control signal	
[15:14]	BUF_NUM	R/W	Auto buffer numbers 00,01:two buffers 10: three buffers 11: four buffers	
[12]	BITSWP	R/W	Bit in each two bits swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[11]	Bits2SWP	R/W	Two bits in a half byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[10]	Bits4SWP	R/W	Half-Byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[9]	BYTSWP	R/W	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
[8]	HAWSWP	R/W	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[7]	ALPHA_SEL	R/W	Select Alpha value. 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 (when BPPMODE=5'b01110 & 5'b01111), using DATA[31:24] data (when BPPMODE=5'b10000)	0
[6]	BLD_PIX	R/W	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
[5:1]	BPPMODE	R/W	Select the BPP (Bits Per Pixel) mode of Window 1 image data. 00000 = 1 BPP (palletized) 00001 = 2 BPP (palletized) 00010 = 4 BPP (palletized) 00011 = 8 BPP (palletized) 00100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 00101 = 16 BPP (non-palletized, R:5-G:6-B:5) 00110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 00111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 01000 = unpacked 18 BPP (non-palletized,R:6-G:6-B:6) 01001 = unpacked 18 BPP (non-palletized,A:1-R:6-G:6-B:5) 01010 = unpacked 19 BPP (non-palletized,A:1-R:6-G:6-B:6)	0

			01011 = unpacked 24 BPP (non-palletized R:8-G:8-B:8) 01100 = unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 01101 = unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 01110 = unpacked 28 BPP (non-palletized A:4-R:8-G:8-B:8) 01111 = unpacked 16 BPP (non-palletized A:4-R:4-G:4-B:4) 10000 = unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) 10001 = YCbCr 4:2:0 semi-planar others = Reserved	
[0]	ENWIN	R/W	Window 1 enable bit. 0 = Disable 1 = Enable	0

Register 10-23: Overlay Controller Window 1 Position Control A Register （OVCW1PCAR,offset=0x1104）

Field	Symbol	Direction	Description	Default
[26:16]	LeftTopX	R/W	Horizontal screen coordinate for left top pixel of OSD image	0
[10:0]	LeftTopY	R/W	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

Register 10-24: Overlay Controller Window 1 Position Control B Register （OVCW1PCBR,offset=0x1108）

Field	Symbol	Direction	Description	Default
[26:16]	RightBotX	R/W	Horizontal screen coordinate for right bottom pixel of OSD image.	0
[10:0]	RightBotY	R/W	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

Register 10-25: Overlay Controller Window 1 Position Control C Register （OVCW1PCCR,offset=0x110C）

Field	Symbol	Direction	Description	Default
[23:20]	ALPHA0_R	R/W	Red Alpha value 0	0
[19:16]	ALPHA0_G	R/W	Green Alpha value 0	0
[15:12]	ALPHA0_B	R/W	Blue Alpha value 0	0
[11:8]	ALPHA1_R	R/W	Red Alpha value 1	0
[7:4]	ALPHA1_G	R/W	Green Alpha value 1	0
[3:0]	ALPHA1_B	R/W	Blue Alpha value 1	0

Register 10-26: Overlay Controller Window 1 Buffer 0 Start Address Register（OVCW1B0SAR,offset=0x1110）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR0	R/W	Start address of the Window 1's video frame buffer 0. **Note**: BUFADDR0 is the address of left top pixel of window 1 virtual screen in DRAM memory.	0

Register 10-27: Overlay Controller Window 1 Start Address Register of Odd Frame （OVCW1BAOF,offset=0x1114）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR1	R/W	Start address of the Window 1's video frame buffer 1. **Note**: BUFADDR1 is the address of left top pixel of window 1 virtual screen in DRAM memory.	0

Register 10-28: Overlay Controller Window 1 Virtual Screen Size Register（OVCW1VSSR,offset=0x1118）

Field	Symbol	Direction	Description	Default
[30:28]	BUF3_BITADR	R/W	Start bit address of the Window 1's video frame buffer 3. Value from 0 to 7 Only valid at bppmode = 0,1,2,3	0
[26:24]	BUF2_BITADR	R/W	Start bit address of the Window 1's video frame buffer 2.	0
[23:20]	BUF1_BITADR	R/W	Start bit address of the Window 1's video frame buffer 1. Value from 0 to 7 Only valid at bppmode = 0,1,2,3	0
[19:16]	BUF0_BITADR	R/W	Start bit address of the Window 1's video frame buffer 0.	0
[15:0]	VW_WIDTH	R/W	Virtual screen page width in the unit: pixel	0

Register 10-29: Overlay Controller Window 1 Color Key Control C Register （OVCW1CKCR,offset=0x111C）

Field	Symbol	Direction	Description	Default
[26]	KEYBLEN	R/W	Color Key (Chroma key) Blending control bit 0 = Disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
[25]	KEYEN	R/W	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
[24]	DIRCON	R/W	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
[23:0]	COMPKEY	R/W	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

Register 10-30: Overlay Controller Window 1 Color Key Register （OVCW1CKR,offset=0x1120）

Field	Symbol	Direction	Description	Default
[23:0]	COLVAL	R/W	Color key value for the transparent pixel effect.	0

Register 10-31: Overlay Controller Window 1 Color Map Register （OVCW1CMR,offset=0x1124）

Field	Symbol	Direction	Description	Default
[24]	MAPCOLEN	R/W	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
[23:0]	MAPCOLOR	R/W	Map Color. MAPCOLOR[23:16] : RED MAPCOLOR[15:8] : GREEN MAPCOLOR[7:0] : BLUE	0

Register 10-32: Overlay Controller Window 1 Buffer 2 Start Address Register（OVCW1B2SAR,offset=0x1128）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR2	R/W	Start address of the Window 1's video frame buffer 2. **Note**: BUFADDR2 is the address of left top pixel of window 1 virtual screen in DRAM memory.	0

Register 10-33: Overlay Controller Window 1 Buffer 3 Start Address Register（OVCW1B3SAR,offset=0x112C）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR3	R/W	Start address of the Window 1's video frame buffer 3. **Note**: BUFADDR3 is the address of left top pixel of window 1 virtual screen in DRAM memory.	0

Register 10-34: Overlay Controller Window 2 Control Register （OVCW2CR,offset=0x1180）

Field	Symbol	Direction	Description	Default
[21]	FIELDSTATUS	R	TVIF Field Fetch Status (Read Only, valid for Interlace mode) 0 = Top Field 1 = Bottom Field	0
[20:19]	BUFSTATUS	R	Window Fetch Buffer Status (Read only, valid when auto ping-pong buffer control bit is enabled.) 0 = fetch buffer 0 1 = fetch buffer 1	0
[18:17]	BUFSEL	R/W	When not use auto ping-pong buffer mode, this bit is used to set fetch buffer by software. 00: fetch buffer 0 01: fetch buffer 1 10: fetch buffer 2 11: fetch buffer 3	0
[16]	BUFAUTOEN	R/W	Auto Ping-pong Buffer control bit 0 = Fixed by BUFSEL 1 = Automatically changed by H/W control signal	0
[15:14]	BUF_NUM	R/W	Auto buffer numbers 00,01:two buffers 10: three buffers 11: four buffers	0
[13]	Reserved	RO	Reserved	0
[12]	BITSWP	R/W	Bit in each two bits swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[11]	Bits2SWP	R/W	Two bits in a half byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[10]	Bits4SWP	R/W	Half-Byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[9]	BYTSWP	R/W	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
[8]	HAWSWP	R/W	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[7]	ALPHA_SEL	R/W	Select Alpha value.	0

			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 (when BPPMODE=5'b01110 & 5'b01111), using DATA[31:24] data (when BPPMODE=5'b10000)	
[6]	BLD_PIX	R/W	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
[5:1]	BPPMODE	R/W	Select the BPP (Bits Per Pixel) mode of Window 2 image data. 00000 = 1 BPP (palletized) 00001 = 2 BPP (palletized) 00010 = 4 BPP (palletized) 00011 = 8 BPP (palletized) 00100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 00101 = 16 BPP (non-palletized, R:5-G:6-B:5) 00110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 00111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 01000 = unpacked 18 BPP (non-palletized,R:6-G:6-B:6) 01001 = unpacked 18 BPP (non-palletized,A:1-R:6-G:6-B:5) 01010 = unpacked 19 BPP (non-palletized,A:1-R:6-G:6-B:6) 01011 = unpacked 24 BPP (non-palletized R:8-G:8-B:8) 01100 = unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 01101 = unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 01110 = unpacked 28 BPP (non-palletized A:4-R:8-G:8-B:8) 01111 = unpacked 16 BPP (non-palletized A:4-R:4-G:4-B:4) 10000 = unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) 10001 = Reserved	0
[0]	ENWIN	R/W	Window 1 enable bit. 0 = Disable 1 = Enable	0

Register 10-35: Overlay Controller Window 2 Position Control A Register （OVCW2PCAR,offset=0x1184）

Field	Symbol	Direction	Description	Default
[26:16]	LeftTopX	R/W	Horizontal screen coordinate for left top pixel of OSD image	0
[10:0]	LeftTopY	R/W	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

Register 10-36: Overlay Controller Window 2 Position Control B Register （OVCW2PCBR,offset=0x1188）

Field	Symbol	Direction	Description	Default
[26:16]	RightBotX	R/W	Horizontal screen coordinate for right bottom pixel of OSD image.	0
[10:0]	RightBotY	R/W	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

Register 10-37: Overlay Controller Window 2 Position Control C Register （OVCW2PCCR,offset=0x118C）

Field	Symbol	Direction	Description	Default
[23:20]	ALPHA0_R	R/W	Red Alpha value 0	0
[19:16]	ALPHA0_G	R/W	Green Alpha value 0	0
[15:12]	ALPHA0_B	R/W	Blue Alpha value 0	0
[11:8]	ALPHA1_R	R/W	Red Alpha value 1	0
[7:4]	ALPHA1_G	R/W	Green Alpha value 1	0
[3:0]	ALPHA1_B	R/W	Blue Alpha value 1	0

Register 10-38: Overlay Controller Window 2 Buffer0 Start Address Register（OVCW2B0SAR,offset=0x1190）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR	R/W	Start address of the Window 2's video frame buffer0 **Note**: BUFADDR is the address of left top pixel of window 2 virtual screen in DRAM memory.	0

Register 10-39: Overlay Controller Window 2 Buffer1 Start Address Register（OVCW2B1SAR,offset=0x1194）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR	R/W	Start address of the Window 2's video frame buffer1 **Note**: BUFADDR is the address of left top pixel of window 2 virtual screen in DRAM memory.	0

Register 10-40: Overlay Controller Window 2 Virtual Screen Size Register（OVCW2VSSR,offset=0x1198）

Field	Symbol	Direction	Description	Default
[30:28]	BUF3_BITADR	R/W	Start bit address of the Window 2's video frame buffer 3. Value from 0 to 7 Only valid at bppmode = 0,1,2,3	0
[26:24]	BUF2_BITADR	R/W	Start bit address of the Window 2's video frame buffer 2.	0
[23:20]	BUF1_BITADR	R/W	Start bit address of the Window 2's video frame buffer 1. Value from 0 to 7 Only valid at bppmode = 0,1,2,3	0
[19:16]	BUF0_BITADR	R/W	Start bit address of the Window 2's video frame buffer 0.	0
[15:0]	VW_WIDTH	R/W	Virtual screen page width in the unit: pixel	0

Register 10-41: Overlay Controller Window 2 Color Key Control C Register （OVCW2CKCR,offset=0x119C）

Field	Symbol	Direction	Description	Default
[26]	KEYBLEN	R/W	Color Key (Chroma key) Blending control bit 0 = Disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
[25]	KEYEN	R/W	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
[24]	DIRCON	R/W	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
[23:0]	COMPKEY	R/W	Each bit is correspond to the COLVAL[23:0].	0

			If some position bit is set then that position bit of COLVAL will be ignored in the fore-ground or back-ground match.	

Register 10-42: Overlay Controller Window 2 Color Key Register （OVCW2CKR,offset=0x11A0）

Field	Symbol	Direction	Description	Default
[23:0]	COLVAL	R/W	Color key value for the transparent pixel effect.	0

Register 10-43: Overlay Controller Window 2 Color Map Register （OVCW2CMR,offset=0x11A4）

Field	Symbol	Direction	Description	Default
[24]	MAPCOLEN	R/W	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
[23:0]	MAPCOLOR	R/W	Map Color. MAPCOLOR[23:16] : RED MAPCOLOR[15:8] : GREEN MAPCOLOR[7:0] : BLUE	0

Register 10-44: Overlay Controller Window 2 Buffer2 Start Address Register（OVCW2B2SAR,offset=0x11A8）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR	R/W	Start address of the Window 2's video frame buffer2 **Note**: BUFADDR is the address of left top pixel of window 2 virtual screen in DRAM memory.	0

Register 10-45: Overlay Controller Window 2 Buffer3 Start Address Register（OVCW2B3SAR,offset=0x11AC）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR	R/W	Start address of the Window 2's video frame buffer3 **Note**: BUFADDR is the address of left top pixel of window 2 virtual screen in DRAM memory.	0

Register 10-46: Overlay Controller Window 3 Control Register （OVCW3CR,offset=0x1200）

Field	Symbol	Direction	Description	Default
[31:22]	Reserved	RO	Reserved	0
[21:19]	OPSEL	R/W	Bitblt operation 1: (INV front) xor backcolor 2: front or backcolor 3: front xor (INV backcolor) 4: front and backcolor	
[18]	PALSEL	R/W	Select ram palette 0 = ram palette 0 in channel 0 1 = ram palette 1 in channel 1	0
[17]	RAMPAL	R/W	Use ram palette(256 color)	0
[16]	CURAND	R/W	Cursor and bitmap enable Cursor operation is (backcolor and andbitmap) xor xorbitmap	0
[15:13]	Reserved	RO	Reserved	0
[12]	BITSWP	R/W	Bit in each two bits swap control bit. 0 = Swap Disable	0

			1 = Swap Enable	
[11]	Bits2SWP	R/W	Two bits in a half byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[10]	Bits4SWP	R/W	Half-Byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[9]	BYTSWP	R/W	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
[8]	HAWSWP	R/W	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[7]	ALPHA_SEL	R/W	Select Alpha value. 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 (when BPPMODE=5'b01110 & 5'b01111), using DATA[31:24] data (when BPPMODE=5'b10000)	0
[6]	BLD_PIX	R/W	Select blending category 0 = Per plane blending 1 = Per pixel blending	0
[5:1]	BPPMODE	R/W	Select the BPP (Bits Per Pixel) mode of Window 0 image data. 00000 = 1 BPP (palletized) 00001 = 2 BPP (palletized) 00010 = 4 BPP (palletized) 00011 = 8 BPP (palletized) 00100 = 8 BPP (non-palletized, A: 1-R:2-G:3-B:2) 00101 = 16 BPP (non-palletized, R:5-G:6-B:5) 00110 = 16 BPP (non-palletized, A:1-R:5-G:5-B:5) 00111 = 16 BPP (non-palletized, I :1-R:5-G:5-B:5) 01000 = unpacked 18 BPP (non-palletized,R:6-G:6-B:6) 01001 = unpacked 18 BPP (non-palletized,A:1-R:6-G:6-B:5) 01010 = unpacked 19 BPP (non-palletized,A:1-R:6-G:6-B:6) 01011 = unpacked 24 BPP (non-palletized R:8-G:8-B:8) 01100 = unpacked 24 BPP (non-palletized A:1-R:8-G:8-B:7) 01101 = unpacked 25 BPP (non-palletized A:1-R:8-G:8-B:8) 01110 = unpacked 28 BPP (non-palletized A:4-R:8-G:8-B:8) 01111 = unpacked 16 BPP (non-palletized A:4-R:4-G:4-B:4) 10000 = unpacked 32 BPP (non-palletized A:8-R:8-G:8-B:8) 10001 = Reserved	0
[0]	ENWIN	R/W	Window 1 enable bit. 0 = Disable 1 = Enable	0

Register 10-47: Overlay Controller Window 3 Position Control A Register （OVCW3PCAR,offset=0x1204）

Field	Symbol	Direction	Description	Default
[26:16]	LeftTopX	R/W	Horizontal screen coordinate for left top pixel of OSD image	0
[10:0]	LeftTopY	R/W	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

Register 10-48: Overlay Controller Window 3 Position Control B Register （OVCW3PCBR,offset=0x1208）

Field	Symbol	Direction	Description	Default
[26:16]	RightBotX	R/W	Horizontal screen coordinate for right bottom pixel of OSD image.	0
[10:0]	RightBotY	R/W	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

Register 10-49: Overlay Controller Window 3 Position Control C Register （OVCW3PCCR,offset=0x120C）

Field	Symbol	Direction	Description	Default
[23:20]	ALPHA0_R	R/W	Red Alpha value 0	0
[19:16]	ALPHA0_G	R/W	Green Alpha value 0	0
[15:12]	ALPHA0_B	R/W	Blue Alpha value 0	0
[11:8]	ALPHA1_R	R/W	Red Alpha value 1	0
[7:4]	ALPHA1_G	R/W	Green Alpha value 1	0
[3:0]	ALPHA1_B	R/W	Blue Alpha value 1	0

Register 10-50: Overlay Controller Window 3 Buffer Start Address Register（OVCW3BSAR,offset=0x1210）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR	R/W	Start address of the Window 3's video frame buffer **Note**: BUFADDR is the address of left top pixel of window 3 virtual screen in DRAM memory.	0

Register 10-51: Overlay Controller Window 3 Virtual Screen Size Register（OVCW3VSSR,offset=0x1214）

Field	Symbol	Direction	Description	Default
[23:20]	ANDBUF_BITADR	R/W	Start bit address of the cursor and bitmap buffer 1 Only valid at bppmode = 0,1,2,3	0
[19:16]	BUF_BITADR	R/W	Start bit address of the Window 3's video frame buffer .	0
[15:0]	VW_WIDTH	R/W	Virtual screen page width in the unit: pixel	0

Register 10-52: Overlay Controller Window 3 Color Key Control C Register （OVCW3CKCR,offset=0x1218）

Field	Symbol	Direction	Description	Default
[26]	KEYBLEN	R/W	Color Key (Chroma key) Blending control bit 0 = Disable (blending operation disable) 1 = Blending using ALPHA0_x for non-key area, ALPHA1_x for key area (x=R, G, B)	0
[25]	KEYEN	R/W	Color Key (Chroma key) Enable control 0 = color key disable 1 = color key enable	0
[24]	DIRCON	R/W	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
[23:0]	COMPKEY	R/W	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

Register 10-53: Overlay Controller Window 3 Color Key Register （OVCW3CKR,offset=0x121C）

Field	Symbol	Direction	Description	Default
[23:0]	COLVAL	R/W	Color key value for the transparent pixel effect.	0

Register 10-54: Overlay Controller Window 3 Color Map Register （OVCW3CMR,offset=0x1220）

Field	Symbol	Direction	Description	Default
[24]	MAPCOLEN	R/W	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
[23:0]	MAPCOLOR	R/W	Map Color. MAPCOLOR[23:16] : RED MAPCOLOR[15:8] : GREEN MAPCOLOR[7:0] : BLUE	0

Register 10-55: Overlay Controller Window 3 Sursor And Bitmap Buffer Start Address Register（OVCW3SABSAR,offset=0x1224）

Field	Symbol	Direction	Description	Default
[31:0]	BUFADDR	R/W	Start address of the Window 3's sursor bitmap buffer	0

Register 10-56: Overlay Controller CbCr Buffer 0 Start Address Register （OVCBRB0SAR,offset=0x1300）

Field	Symbol	Direction	Description	Default
[31:0]	OVCBRB0SAR	R/W	Overlay Controller CbCr Buffer 0 Start Address	0

Register 10-57: Overlay Controller CbCr Buffer 1 Start Address Register （OVCBRB1SAR,offset=0x1304）

Field	Symbol	Direction	Description	Default
[31:0]	OVCBRB1SAR	R/W	Overlay Controller CbCr Buffer 1 Start Address	0

Register 10-58: Overlay Controller Color Matrix Coefficient11 Register （OVCOEF11,offset=0x1308）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF11	R/W	Color space convert matrix coefficient coef11, coef11 is signed number and range from -1024 to +1023 Recommend value = +298 (dec)	0

Register 10-59: Overlay Controller Color Matrix Coefficient12 Register （OVCOEF12,offset=0x130C）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF12	R/W	Color space convert matrix coefficient coef12, coef12 is signed number and range from -1024 to +1023 Recommend value = 0	0

Register 10-60: Overlay Controller Color Matrix Coefficient13 Register （CICOEF13,offset=0x1310）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF13	R/W	Color space convert matrix coefficient coef13, coef13 is signed number and range from -1024 to +1023	0

			Recommend value = +409 (dec)	

Register 10-61: Overlay Controller Color Matrix Coefficient21 Register （OVCOEF21,offset=0x1314）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF21	R/W	Color space convert matrix coefficient coef21, coef21 is signed number and range from -1024 to +1023 Recommend value = +298 (dec)	0

Register 10-62: Overlay Controller Color Matrix Coefficient22 Register （OVCOEF22,offset=0x1318）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF22	R/W	Color space convert matrix coefficient coef22, coef22 is signed number and range from -1024 to +1023 Recommend value = -100 (dec)	0

Register 10-63: Overlay Controller Color Matrix Coefficient23 Register （OVCOEF23,offset=0x131C）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF23	R/W	Color space convert matrix coefficient coef23, coef23 is signed number and range from -1024 to +1023 Recommend value = -208 (dec)	0

Register 10-64: Overlay Controller Color Matrix Coefficient31 Register （OVCOEF31,offset=0x1320）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF31	R/W	Color space convert matrix coefficient coef31, coef31 is signed number and range from -1024 to +1023 Recommend value = +298 (dec)	0

Register 10-65: Overlay Controller Color Matrix Coefficient32 Register （OVCOEF32,offset=0x1324）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF32	R/W	Color space convert matrix coefficient coef32, coef32 is signed number and range from -1024 to +1023 Recommend value = +516 (dec)	0

Register 10-66: Overlay Controller Color Matrix Coefficient33 Register （OVCOEF33,offset=0x1328）

Field	Symbol	Direction	Description	Default
[10:0]	OVCOEF33	R/W	Color space convert matrix coefficient coef33, coef33 is signed number and range from -1024 to +1023 Recommend value = 0	0

Register 10-67: Overlay Controller Color Matrix Configure Register （OVCOMC,offset=0x132C）

Field	Symbol	Direction	Description	Default
[31]	ToRGB	R/W	Matrix transform mode and always set to 1	0
[30:25]	Reserved	R/W	Must be set to 0	
[24]	ODD_EVEN	R/W	The line number of display buffer address is a odd line or even line. 0: even line 1: odd line	
[23:21]	Reserved	R/W	Must be set to 0	

[20]	BITSWP	R/W	Bit in each two bits swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[19]	Bits2SWP	R/W	Two bits in a half byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[18]	Bits4SWP	R/W	Half-Byte swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[17]	BYTSWP	R/W	Byte swaps control bit. 0 = Swap Disable 1 = Swap Enable	0
[16]	HAWSWP	R/W	Half-Word swap control bit. 0 = Swap Disable 1 = Swap Enable	0
[12: 8]	oft_b	R/W	Color space convert matrix coefficient oft_b, oft_b is unsigned number and range from 0 to 31 Recommend value = 16	
[4: 0]	oft_a	R/W	Color space convert matrix coefficient oft_a, oft_a is unsigned number and range from 0 to 31 Recommend value = 0	

Register 10-68: Overlay Controller CbCr Buffer 2 Start Address Register （OVCBRB2SAR,offset=0x1330）

Field	Symbol	Direction	Description	Default
[31:0]	OVCBRB2SAR	R/W	Overlay Controller CbCr Buffer 2 Start Address	0

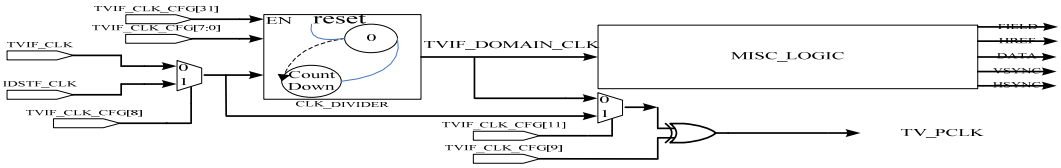
Register 10-69: Overlay Controller CbCr Buffer 3 Start Address Register （OVCBRB3SAR,offset=0x1334）

Field	Symbol	Direction	Description	Default
[31:0]	OVCBRB3SAR	R/W	Overlay Controller CbCr Buffer 3 Start Address	0

Register 10-70: TVIF Clock Configuration Register(TVCCR,offset=0x2000)

Field	Symbol	Direction	Description	Default
[31]	Clock_enable	R/W	TVIF controller clock enable 0: disable clock 1: enable clock	0
[11]	TV_PCLK _mode	R/W	TV_PCLK source select 0: divide clock 1: controller clock(depend on bit[8], which can be external reference clock or internal clock)	0
[10]	Reserved	R/W	Always set to 0	0
[9]	Inv_clock	R/W	Inverse the polarity of clock 0: output data by rising edge 1: output data by falling edge	0
[8]	clock_sel	R/W	Clock selection 0: extern reference clock or special TV clock which is generated by system clock generator 1: internal IDSTF clock, which is generated by system clock generator	0
[7:0]	Clock_div	R/W	Clock divider factor 0: use clock selected clock as pixel clock Others: divider selected clock to (factor+1)	0

Note: TVIF Clock scheme shows as followed:



Register 10-71: TVIF Configuration Register(TVICR,offset=0x2004)

Field	Symbol	Direction	Description	Default
[31]	tvif_enable	R/W	TVIF controller interface enable 0: disable, 1: enable	0
[30]	ITU601_656n	R/W	ITU interface selection, valid when TVIF_CFG[28] = 0 0: ITU656 interface 1: ITU601 interface	0
[29]	Bit16ofITU60	R/W	Data width in ITU601, valid when TVIF_CFG[28] = 0 0: 8-bit data in ITU601 interface 1: 16-bit data in ITU601 interface	0

[28]	Direct_data	R/W	matrix data output directly or not
			0: ITU601 or ITU656(16-bit or 8-bit)
			1: direct output RGB (24-bit)
			Note:
			if(direct_data == 1)
			output_data_width = 24-bit;
			else if((ITU601_656n == 1) && (Bit16ofITU601 == 1))
			output_data_width = 16-bit;
			else
			output_data_width = 8-bit;
				0

[18]	Bitswap	R/W	Bits swap in a byte 0: maintain and MSB and LSB 1: inverse MSB to LSB	0
[17:16]	Data_order	R/W	Output data order, have relationship with follows, TVIF_CFG[30], TVIF_CFG[29] and TVIF_CFG[28]	0
[13]	Inv_vsync	R/W	Inverse the polarity of vsync 0: active low 1: active high	0
[12]	Inv_hsync	R/W	Inverse the polarity of hsync 0: active low 1: active high	0
[11]	Inv_href	R/W	Inverse the polarity of href 0: active high 1: active low	0
[10]	Inv_field	R/W	Inverse the polarity of field 0: 0=odd, 1=even 1: 0=even, 1=odd	0
[0]	Begin_with_EAV	R/W	Line begins with EAV 0: line begins with blanking 1: line begins with EAV. (Recommend)	0

Register 10-72: TVIF Color Space Convert Matrix Coefficient Register(TVCOEFxx,offset=0x2008~0x2028)

Field	Symbol	Direction	Description	Default
[31:0]	Coefficient11	R/W	Color Space Convert Matrix Coefficient 11	0

[31:0]	Coefficient12	R/W	Color Space Convert Matrix Coefficient 12	0
[31:0]	Coefficient13	R/W	Color Space Convert Matrix Coefficient 13	0
[31:0]	Coefficient21	R/W	Color Space Convert Matrix Coefficient 21	0
[31:0]	Coefficient22	R/W	Color Space Convert Matrix Coefficient 22	0
[31:0]	Coefficient23	R/W	Color Space Convert Matrix Coefficient 23	0
[31:0]	Coefficient31	R/W	Color Space Convert Matrix Coefficient 31	0
[31:0]	Coefficient32	R/W	Color Space Convert Matrix Coefficient 32	0
[31:0]	Coefficient33	R/W	Color Space Convert Matrix Coefficient 33	0

Register 10-73: TVIF Color Matrix Configure Register(TVCMCR,offset=0x202C)

Field	Symbol	Direction	Description	Default
[31]	Matrix_mode	R/W	0: convert RGB to YCbCr, invert the MSB of output data 1: convert YCbCr to RGB, invert the MSB of input data **Note**: This field always must be set to 0.	0
[30]	Passby	R/W	Passby the matrix 0: use matrix to convert RGB to YCbCr 1: RGB directly output to YCbCr (note, if bypass the color transfer matrix, R<->Y, G<->Cb, B<->Cr)	0
[29]	Inv_MSB_in	R/W	Invert the MSB of input data of Cb and Cr when Matrix_mode is true, or Inv the MSB of G and B when Matrix_mode is false.	0
[28]	Inv_MSB_out	R/W	Invert the MSB of output data of Cb and Cr when Matrix_mode is false, or Inv the MSB of G and B when Matrix_mode is true.	0
[12:8]	Matrix_oft_b	R/W	Matrix offset a RGB2YCbCr: (dec)0 YCbCr2RGB: (dec)16	0
[4:0]	Matrix_oft_a	R/W	Matrix offset a RGB2YCbCr: (dec)0 YCbCr2RGB: (dec)16	0

Register 10-74: TVIF Upper Blanking Area 1 Line Register(TVUBA1,offset=0x2030)

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0
[10:0]	UBA1_LEN	R/W	length of upper blanking area PAL: 22, NTSC: 19	0

Register 10-75: TVIF Upper Non-blanking Area Line Register(TVUNBA,offset=0x2034)

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0
[10:0]	UNBA_LEN	R/W	length of upper non-blanking area PAL: 288, NTSC: 240	0

Register 10-76: TVIF Upper Blanking Area 2 Line Register (TVUBA2,offset=0x2034)

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0
[10:0]	UBA2_LEN	R/W	length of upper blanking area 2 PAL: 2, NTSC: 3	0

Register 10-77: TVIF Lower Blanking Area 1 Line Register (TVLBA1 ,offset=0x2038)

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0
[10:0]	LBA1_LEN	R/W	length of lower blanking area PAL: 23, NTSC: 20	0

Register 10-78: TVIF Lower Non-blanking Area Line Register (TVLNBA ,offset=0x2040)

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0
[10:0]	LNBA_LEN	R/W	length of lower non-blanking area PAL: 288, NTSC: 240	0

Register 10-79: TVIF Lower Blanking Area 2 Line Register (TVLBA2 ,offset=0x2044)

Field	Symbol	Direction	Description	Default
[31:11]	Reserved	R/W	Reserved	0
[10:0]	LBA2_LEN	R/W	length of lower blanking area 2 PAL: 2 NTSC: 3	0

Register 10-80: TVIF Line Blanking Length Register (TVBLEN,offset=0x2048)

Field	Symbol	Direction	Description	Default
[31:12]	Reserved	R/W	Reserved	0
[11:0]	BLANK_LEN	R/W	length of blank per line PAL: 280 NTSC: 268 (note: total line clock cycle subtract active video clock cycle, and then subtract 8)	0

Register 10-81: TVIF Line Video Length Register (TVVLEN,offset=0x204C)

Field	Symbol	Direction	Description	Default
[31:12]	Reserved	R/W	Reserved	0
[11:0]	VIDEO_LEN	R/W	length of non-blank per line PAL: 1440 NTSC: 1440 (note: active video clock number, in ITU656 and ITU601 8-bit mode, 1 pixel need 2 clock cycle)	0

Register 10-82: TVIF Hsync Configure Register(TVHSCR,offset=0x2050)

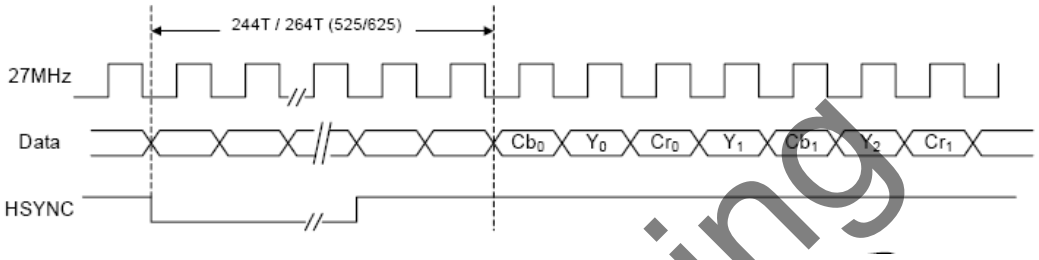
Field	Symbol	Direction	Description	Default
[31:30]	Hsync_VBI_ctrl	R/W	Hsync behavior control in VBI 00: remain the same as active data part 01: hold to 0 10: hold to 1 11: Reserved **Note**: 00 is for ITU601	0
[28:16]	Hsync_delay	R/W	The clock cycle delay of hsync, 0 means no delay of hsync PAL: 24 (1728-1440-264) NTSC: 32 (1716-1440-244)	0

[12:0]	Hsync_extend	R/W	The clock cycle of hsync, 0 means no hsync PAL: 126 NTSC: 124	0

Note1: The recommend PAL and NTSC default number is just for ITU-601/656 8-bit data width and clock frequency is 27MHz

Note2: Because of the flexible configuration of HSYNC, user can use HSYNC as DATA_EN(it is known as HREF), in the mode of ITU-601/656 8-bit data width, 27MHz. The configure parameter are shown as followed:

- TVIF_CFG[0] = 1; // begin with EAV
- TVIF_CFG[12] = 0; // HSYNC active low, it means use inactive of HSYNC as HREF
- TVIF_HSYNC[31:30] = 2; // in the VBI hold HSYNC to 0
- TVIF_HSYNC[12:0] = 287(PAL) or 275(NTSC)// EAV/SAV/BLANKING all hold to 0.



Register 10-83: TVIF Vsync Upper Configure Register(TVVSHCR,offset=0x2054)

Field	Symbol	Direction	Description	Default
[28:16]	Vsync_delay_upper	R/W	The clock cycle delay from hsync to vsync, 0 means no delay. PAL: 24 (the same as hsync_dealy) NTSC: 32 (the same as hsync_dealy)	0
[12:0]	Vsync_extend_upper	R/W	The clock cycle of vsync, 0 means no vsync. PAL: 4320 (2.5 line times) NTSC: 5148 (3 line times)	0

Register 10-84: TVIF Vsync lower Configure Register(TVVSLCR,offset=0x2058)

Field	Symbol	Direction	Description	Default
[28:16]	Vsync_delay_lower	R/W	The clock cycle delay from hsync to vsync, 0 means no delay. PAL: 888 (0.5 line time + hsync_delay) NTSC: 890 (0.5 line time + hsync_delay)	0
[12:0]	Vsync_extend_lower	R/W	The clock cycle of vsync, 0 means no vsync. PAL: 4320 (2.5 line time) NTSC: 5148 (3 line time)	0

Note: The recommend PAL and NTSC default number is just for ITU-601/656 8-bit data width and clock frequency is 27MHz.

Register 10-85: TVIF Display Horizontal Size Register(TVXSIZE,offset=0x205C)

Field	Symbol	Direction	Description	Default
[10:0]	DISP_XSIZE	R/W	Display horizontal size - 1 PAL: 720 – 1 = 719 NTSC: 720 – 1 = 719	0

Register 10-86: TVIF Display Vertical Size Register(TVYSIZE,offset=0x2060)

Field	Symbol	Direction	Description	Default
[10:0]	DISP_YSIZE	R/W	Display horizontal size – 1 PAL : 576 – 1 = 575 NTSC: 480 – 1 = 479	0

Note: The recommend PAL and NTSC default number is just for ITU-601/656 8-bit data width and clock frequency is 27MHz.

Register 10-87: TVIF Status Register(TVSTSR,offset=0x2064)

Field	Symbol	Direction	Description	Default
[0]	Fifo_underflow	R	Asyn fifo underflow indicate, read clear.	0

Register 10-88: I80 Trigger Control(I80TRIGCON,offset=0x3000)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	RO	Reserved	0
[7:5]	Reserved	R/W	Must be '0'	0
[4]	XFER_VIDEO_OVER	ROC	Video Data transfer over status, Read Only, Read Clear.	
[3]	TNORMALCMD	RO	Transmitting Normal Command, Status Indicator. Read Only. (RO)	
[2]	TAUTOCMD	RO	Transmitting Auto Command, Status Indicator. Read Only. (RO)	
[1]	TVIDEO	RO	Transmitting Video Data, Status indicator. Read Only. (RO)	0
[0]	NORMAL_CMD_ST	R/W	W: Trigger the Normal Command R: Waiting the Normal Command to start	0

Register 10-89: I80 Interface Control0(I80IFCON0,offset=0x3004)

Field	Symbol	Direction	Description	Default
[31]	RDPOL	R/W	Inverse the polarity of the RD_n (Read) Signal in non-manual mode 0: RD_n signal is active low 1: RD_n signal is active high	0
[30]	WRPOL	R/W	Inverse the polarity of the WR_n (Write) Signal in non-manual mode 0: WR_n signal is active low 1: WR_n signal is active high	0
[29]	RSPOL	R/W	Inverse the polarity of the RS_n (Register Select) Signal in non-manual mode 0: RS_n signal is active low 1: RS_n signal is active high	
[28]	CS1POL	R/W	Inverse the polarity of the CS1_n(Chip Select) Signal in non-manual mode 0: CS1_n signal is active low 1: CS1_n signal is active high	
[27]	CS0POL	R/W	Inverse the polarity of the CS2_n(Chip Select) Signal in non-manual mode 0: CS2_n signal is active low	

			1: CS2_n signal is active high	
[26:22]	LCD_CS_SETUP	R/W	Numbers of clock cycles for the active period of the address signal enable to the chip select enable.	0
[21:16]	LCD_WR _SETUP	R/W	Numbers of clock cycles for the active period of the CS signal enable to the write signal enable.	0
[15:10]	LCD_WR_ACT	R/W	Numbers of clock cycles for the active period of the chip select enable.	
[9:4]	LCD_WR _HOLD	R/W	Numbers of clock cycles for the active period of the chip select disable to the write signal disable.	
[3:2]	INTMASK	R/W	Interrupt Mask 00: No interrupt 01: Frame over 10: Normal CMD over 11: Frame over and Normal CMD over	
[1]	I80MainConfig	R/W	Main LCD configuration 0: use CS0_n as main LCD (controlled by I80IF) 1: use CS1_n as main LCD (controlled by I80IF)	
[0]	I80IFEN	R/W	LCD I80 Interface Enable 0: Disable 1: Enable	

Register 10-90: I80 Interface Control1(I80IFCON1,offset=0x3008)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	R/W	Reserved	0
[15:14]	PortType	R/W	Data Port Connect Type, 18/16/9/8-Wire. See Note 1	0
[13:12]	PortDistributed	R/W	Video Data Distribute on Data Port. See Note 1	
[11:10]	DataStyle	R/W	Video Data transfer Style on the Data Port. 16/16+2/2+16 in 16-Wire. 8+8/6+6+6/8+8+2/2+8+8 in 8-Wire. See Note 1	
[9]	DataLSBFirst	R/W	Video Data LSB data is Transferred First. See Note 1	
[8]	DataUseLowPort	R/W	Video Data Use Low Select Data Port.	0
[4]	DisAutoCMD	R/W	Send video data only in every frame 0: send auto command before video data, the auto command rate controlled by I80IFCON1[3:0]. 1: only send video data in every frame	0
[3:0]	AUTO_CMD_RATE	R/W	0000 : per 1 Frame 0001 : per 2 Frames 0010 : per 4 Frames 0011 : per 6 Frames … 1111 : per 30 Frames	0

Register 10-91: I80 Command Control0(I80CMDCON0,offset=0x300C)

Field	Symbol	Direction	Description	Default
[31:30]	CMD15_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[29:28]	CMD14_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[27:26]	CMD13_EN	R/W	00 : Disable 01 : Normal Command Enable	

			10 : Auto Command Enable 11 : Normal and Auto Command Enable	
[25:24]	CMD12_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[23:22]	CMD11_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[21:20]	CMD10_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[19:18]	CMD9_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[17:16]	CMD8_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[15:14]	CMD7_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[13:12]	CMD6_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[11:10]	CMD5_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[9:8]	CMD4_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[7:6]	CMD3_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[5:4]	CMD2_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[3:2]	CMD1_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0
[1:0]	CMD0_EN	R/W	00 : Disable 01 : Normal Command Enable 10 : Auto Command Enable 11 : Normal and Auto Command Enable	0

Register 10-92: I80 Command Control1(I80CMDCON1,offset=0x3010)

Field	Symbol	Direction	Description	Default
[30]	CMD15_RS	R/W	Command 15 RS control	0
[28]	CMD14_RS	R/W	Command 14 RS control	0
[26]	CMD13_RS	R/W	Command 13 RS control	0
[24]	CMD12_RS	R/W	Command 12 RS control	0
[22]	CMD11_RS	R/W	Command 11 RS control	0
[20]	CMD10_RS	R/W	Command 10 RS control	0
[18]	CMD9_RS	R/W	Command 9 RS control	0
[16]	CMD8_RS	R/W	Command 8 RS control	0
[14]	CMD7_RS	R/W	Command 7 RS control	0
[12]	CMD6_RS	R/W	Command 6 RS control	0
[10]	CMD5_RS	R/W	Command 5 RS control	0
[8]	CMD4_RS	R/W	Command 4 RS control	0
[6]	CMD3_RS	R/W	Command 3 RS control	0
[4]	CMD2_RS	R/W	Command 2 RS control	0
[2]	CMD1_RS	R/W	Command 1 RS control	0
[0]	CMD0_RS	R/W	Command 0 RS control	0

Register 10-93: I80 Command Instruct X (I80CMDX,offset=0x3014~0x3050)

Field	Symbol	Direction	Description	Default
[17:0]	LDI_CMDx	R/W	LDI command	0

Note: Where x range from 15 downto 0. LDI_CMDx which activate with CMDx_RS are controlled by CMDx_en in Normal Command Mode and Auto Command Mode, LDI_CMDx and CMDx_RS are mapped to PAD of D[17: 0] and RS.

Register 10-94: I80 Manual Control (I80MANCON,offset=0x3054)

Field	Symbol	Direction	Description	Default
[31:7]	Reserved	R/W	Reserved	0
[6]	MAN_DOE	R/W	LCD i80 System Interface Data Enable Signal control. 0: Disable (input) 1: Enable (output)	0
[5]	MAN_RS	R/W	LCD I80 System Interface nRS Signal control. 0: level low 1: level high	
[4]	MAN_CS0	R/W	LCD I80 System Interface nCS0 (main) Signal control 0: level low 1: level high	0
[3]	MAN_CS1	R/W	LCD I80 System Interface nCS1 (sub) Signal control 0: level low 1: level high	0
[2]	MAN_RD	R/W	LCD I80 System Interface nOE Signal control 0: level low 1: level high	0
[1]	MAN_WR	R/W	LCD I80 System Interface nWE Signal control 0: level low 1: level high	0
[0]	MAN_EN	R/W	LCD I80 System Interface Command Mode Enable 0: Disable 1: Enable (Manual Command Mode)	0

Register 10-95: I80 Manual Write Data (I80MANWDAT,offset=0x3058)

Field	Symbol	Direction	Description	Default
[31:18]	Reserved	R/W	Reserved	0
[17:0]	SYS_WDAT	R/W	LCD I80 System Interface Write Data Buffer	0

Register 10-96: I80 Manual Read Data (I80MANRDAT,offset=0x305C)

Field	Symbol	Direction	Description	Default
[31:18]	Reserved	RO	Reserved	0
[17:0]	SYS_RDAT	RO	LCD I80 System Interface Read Data Buffer	0

11 Camera Interface

11.1 Overview

This chapter will explain the specification and defines the camera interface. CAMIF (CAMera InterFace) within the IMAPX210 consists of 5 parts – Catcher, Preview Scaler, Codec Path, DMA and SFR. The CAMIF supports ITU-R BT.601/656 YCbCr 8-bit interface. Maximum input size is 4096x4096 pixels. Preview scaler is dedicated to generate smaller size image like PIP (Picture In Picture). Codec Path can processing image data and generate codec useful image like plane type YCbCr 4:2:0 or 4:2:2. Five channels of DMA can easily transfer image data of each plane to corresponding frame buffer.

Features

- ITU-R BT. 601/656 8-bit mode external interface support
- DZO(Digital Zoom Out) capability of Preview path
- Programmable polarity of video sync signals
- Max. 4096 x 4096 pixel support for CODEC path
- Max. 4096 x 4096 pixel support for PREVIEW path
- PIP and codec input image generation (RGB 16/24-bit format and YCbCr 4:2:0/4:2:2 format)
- Internal FIFO self-reset at the beginning of every frame or field

Block Diagram

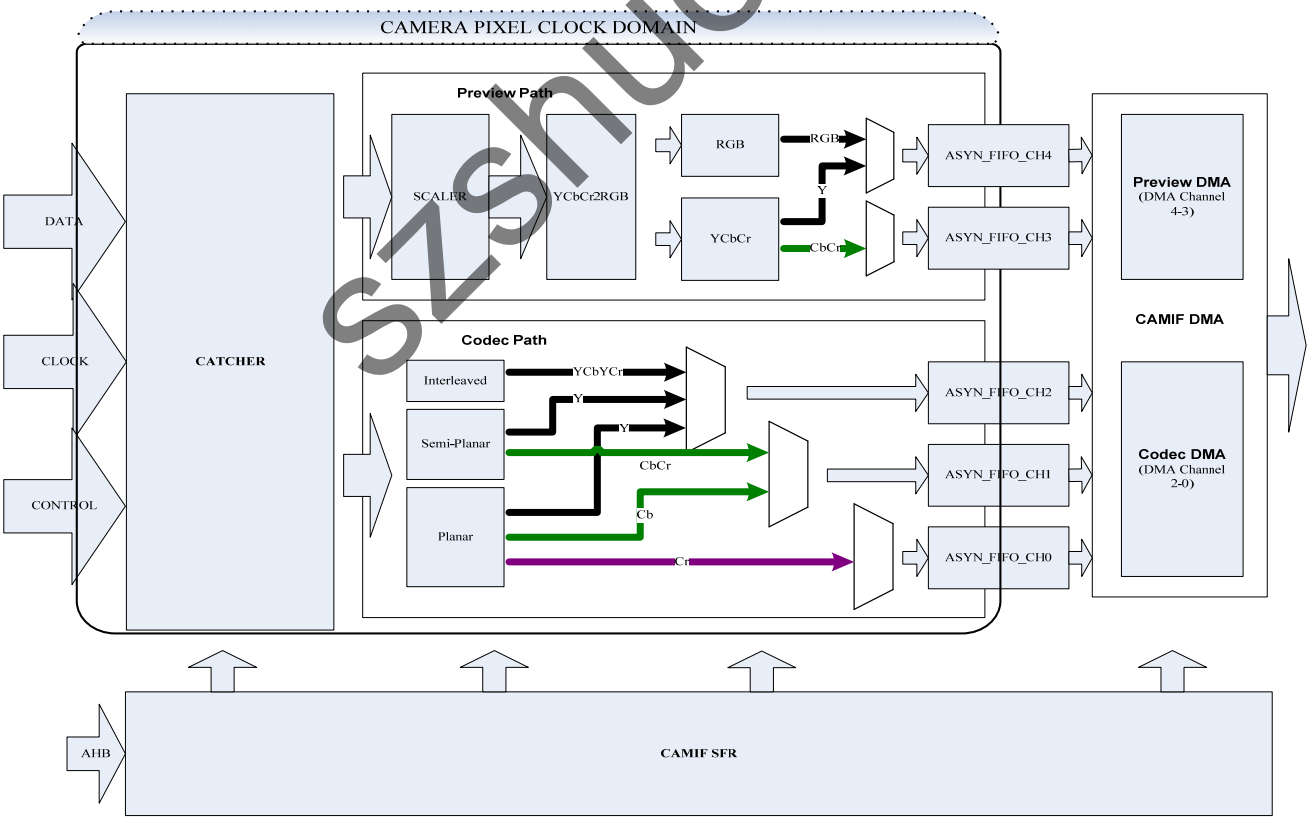


Figure 11-1: Camera Interface Block Diagram

11.2 ITU-R BT 601/656 Protocol

ITU-R BT 601 frame (field) timing format is shown in Figure 11-2, CAMHSYNC is used as field indicate signal, the current field is even or odd field depend on the specific timing in Figure 11-3.

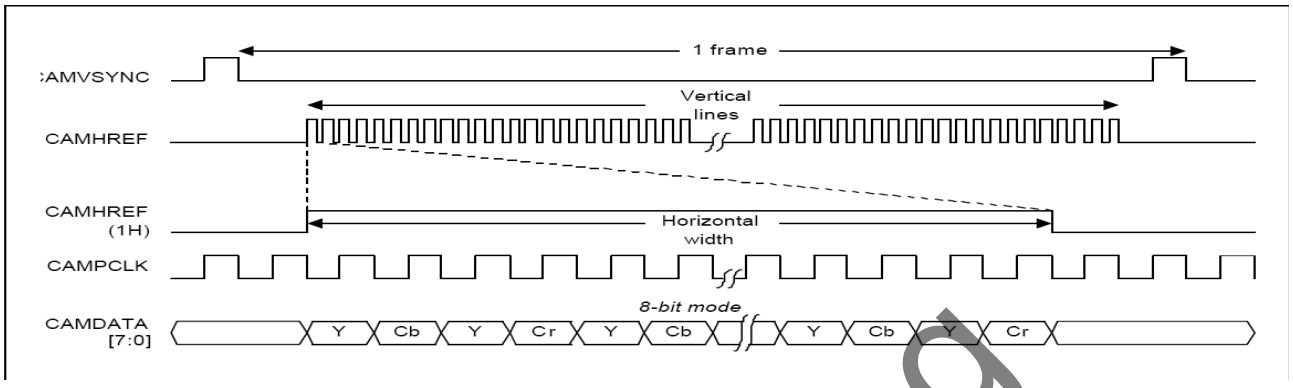


Figure 11-2: ITU-R BT 601 Input Field Timing Diagram

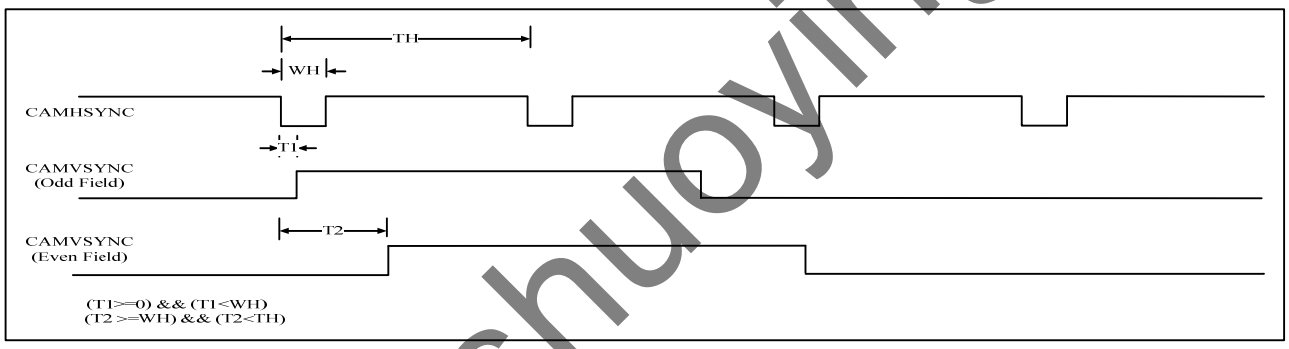


Figure 11-3: ITU-R BT 601 Even and Odd Field Timing Diagram

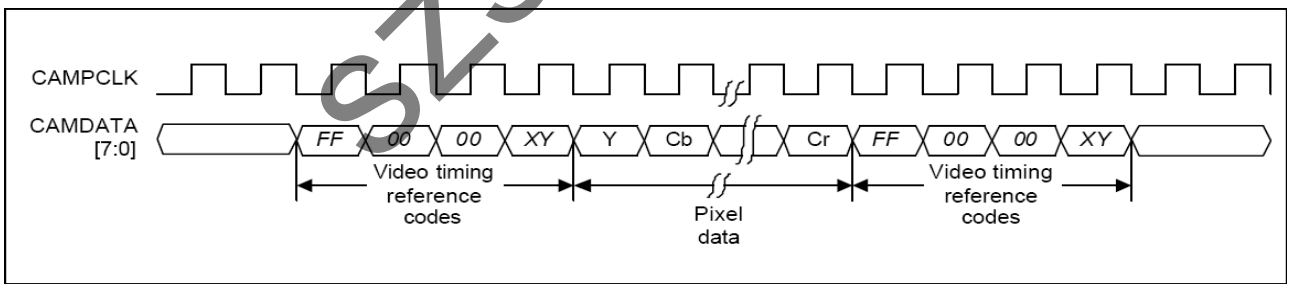


Figure 11-4: ITU-R BT 656 Input Timing Diagram

There are two timing reference signals in ITU-R BT 656 format, one is at the beginning of each video data block (start of active video, SAV) and other is at the end of each video data block (end of active video, EAV) as shown in Figure 11-4 and Table 11-1.

Table 11-1: Video Timing Reference Codes of ITU-656 Format

Data Bit Number	First Word	Second Word	Third Word	Fourth Word
7	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

Where,

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)
P3 = V^H, protection bit
P2 = F^H, protection bit
P1 = F^V, protection bit
P0 = F^V^H, protection bit

Camera interface logic CATCHER can catch the video sync bits like H (SAV, EAV) and V (Frame Sync) after Reserved data as “FF-00-00”.

11.3 Camera Interface Operation

Two DMA Paths

CAMIF has 2 DMA paths. P-path (Preview path) and C-path (Codec path) are separated from each other on the AXI bus. In view of the system bus, both the paths are independent. The P-path stores the RGB image data into memory for PIP. The C-path has three channels and stores the YCbCr 4:2:0 or 4:2:2 image data into memory for Codec as MPEG-4, H.263, etc. These two master paths support the variable applications like DSC (Digital Steel Camera), MPEG-4 video conference, video recording, etc. For example, P-path image can be used as preview image, and C-path image can be used as JPEG image in DSC application. Register setting can separately disable to P-path or C-path.

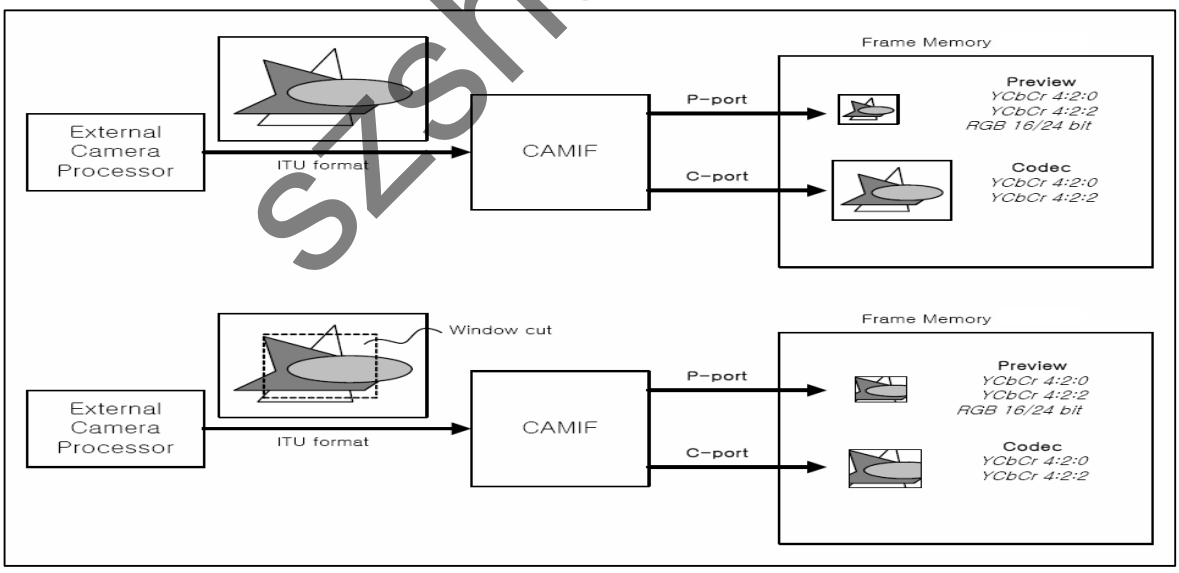


Figure 11-5: Two DMA Paths

Clock Domain

CAMIF has two clock domains. One is the system bus clock, which is HCLK2X. The other is the pixel clock, which is

CAMPCLK. The system clock must be faster than pixel clock. Figure 11-6 shows CAMCLKOUT can be divided from DPLL or EPLL out clock, which is configured in system controller. If external clock oscillator is used, CAMCLKOUT should be floated. It is not necessary for two clock domains to synchronize each other. Other signals such as CAMPCLK should be similarly connected to the Schmitt-triggered level shifter.

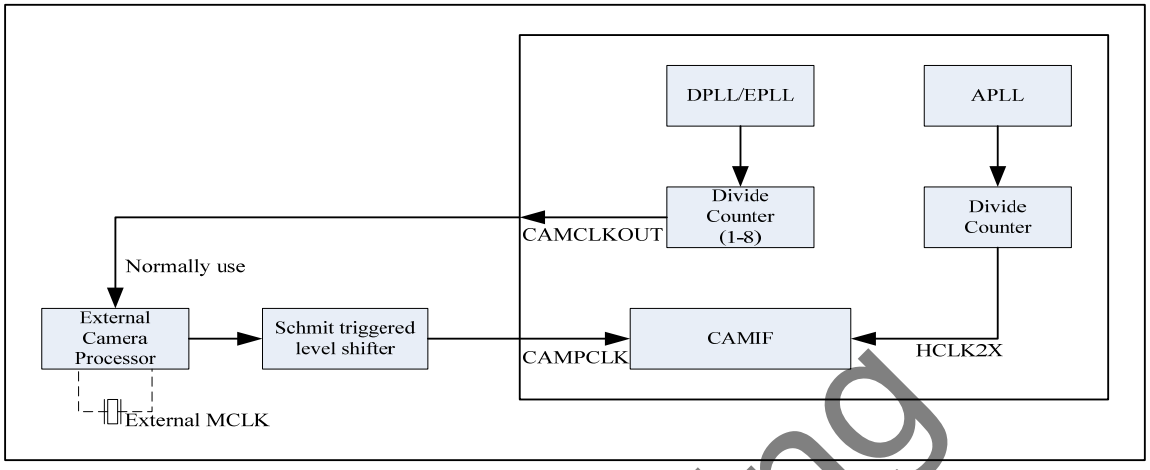


Figure 11-6: CAMIF Clock Generation

Frame Memory Hirerarchy

Frame memories consist of four ping-pong memories for each of P and C paths as shown in the Figure 11-7. C-path ping-pong memories have three element memories – luminance Y, chrominance Cb, and chrominance Cr. Four channels of DMA is responsible to transfer corresponding image data to element memory, respectively. If AXI-bus traffic is not enough for the DMA operation to complete during one horizontal line period, it may lead to malfunctioning.

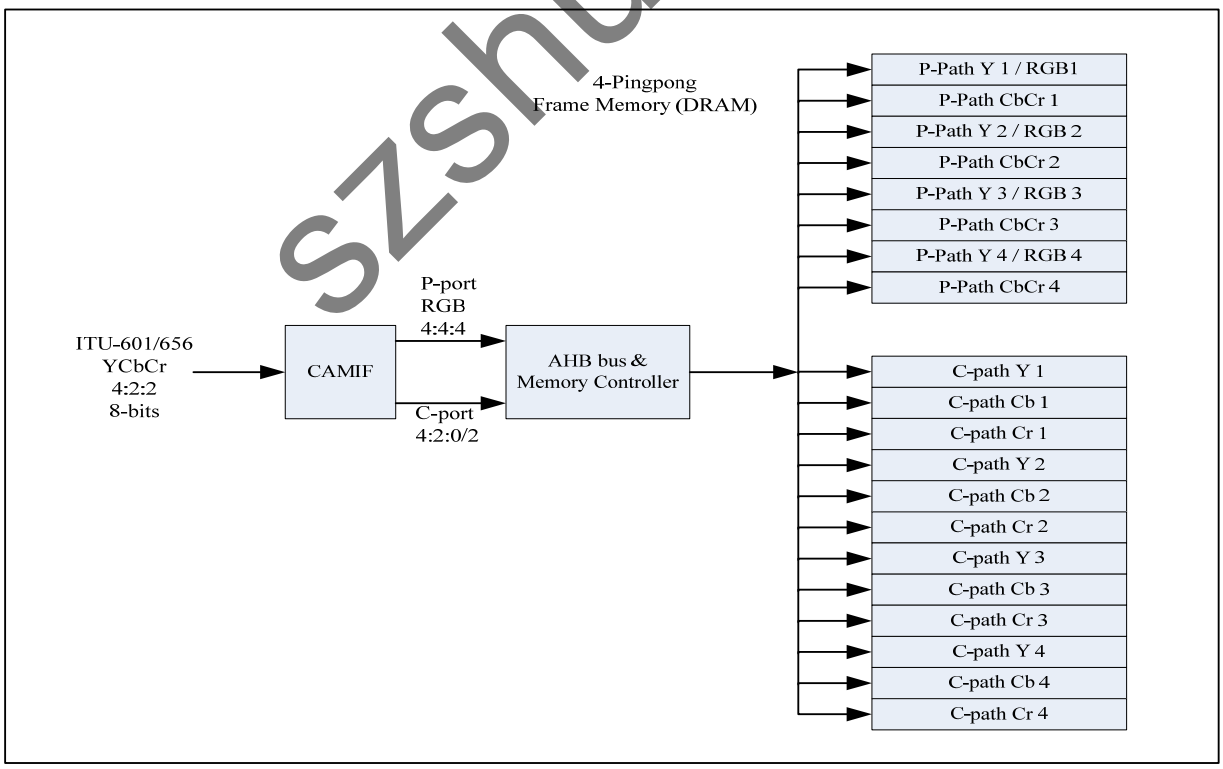


Figure 11-7: Ping-Pong Memory Hierarchy

Memory Store Method and Data Format

The little-endian method in codec path is used to store in the frame memory. The pixels are stored from LSB to MSB side, but you can change the byte order via register bit CICOTRGFMT[30] (HWSWAP) and CICOTRGFMT[29] (BTSWAP), please refer Table 11-2. For Codec path, there are two data store modes, include planer and interleaved mode. CAMIF makes each of the Y-Cb-Cr words in little-endian style in planer mode. In interleaved mode, CAMIF under-sample the incoming data and store to Y frame buffer. Figure 11-8 and Figure 11-9 show the codec path memory store method and under-sample method, respectively.

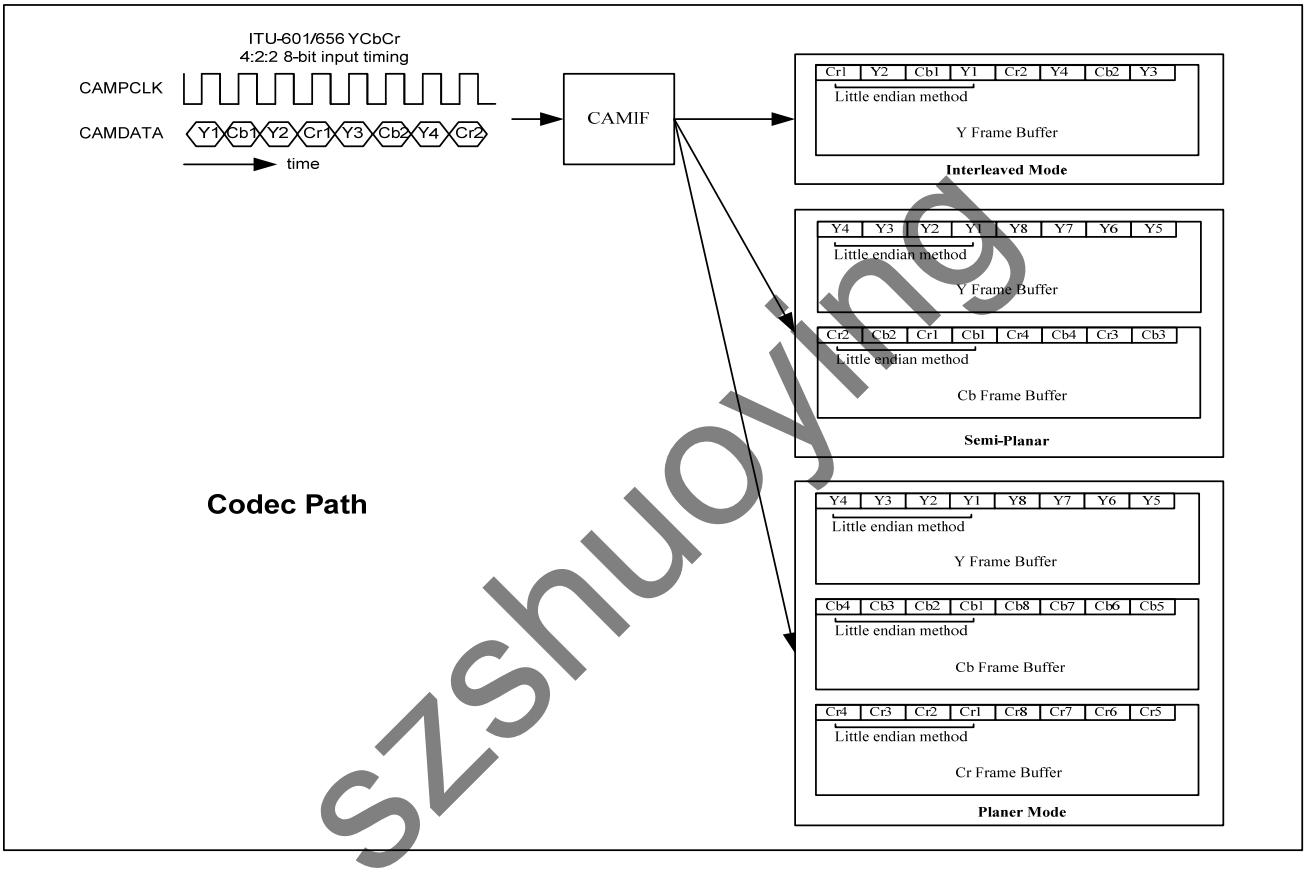


Figure 11-8: Codec Path Memory Store Method

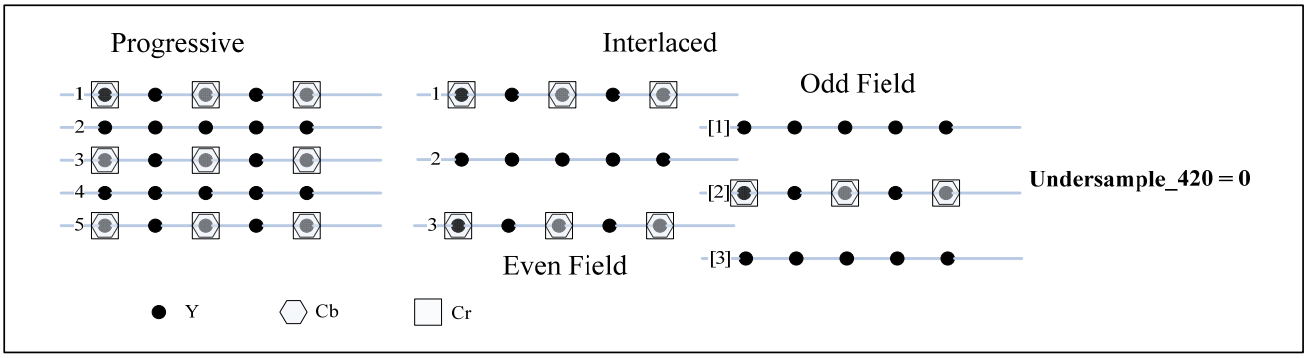


Figure 11-9: Under-Sample Method

Table 11-2: Byte Order Format

HWSWAP	BTSWAP	Byte Order
0	0	B3, B2, B1, B0
1	0	B1, B0, B3, B2
0	1	B2, B3, B0, B1
1	1	B0, B1, B2, B3

For preview path, two different formats exist. One pixel (Color 1 pixel) is one word for RGB 24-bit format. Otherwise, two pixels are one word for RGB 16-bit format (RGB565 and RGB5551).

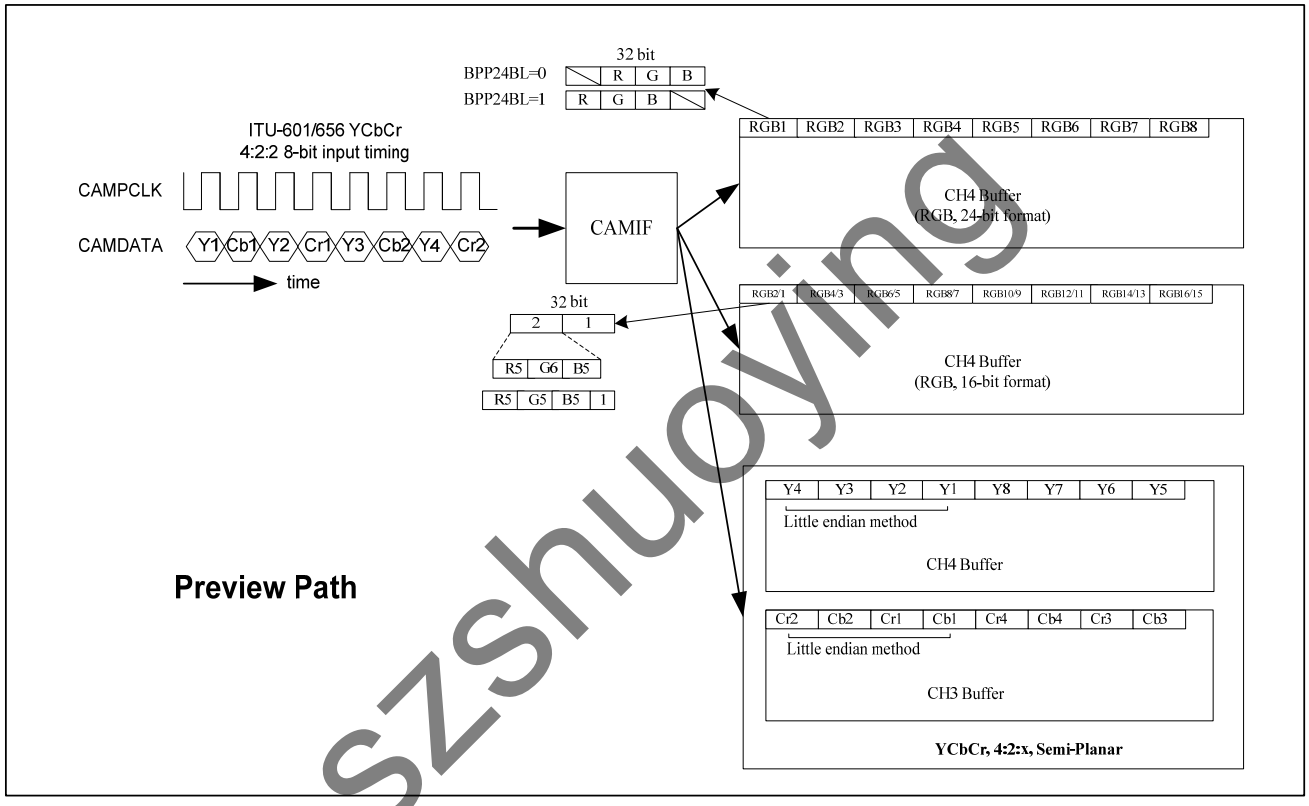


Figure 11-10: Preview Path Memory Store Method

Timing Diagram for Register Setting

The first register setting for frame capture command can occur anywhere in the frame period. But, it is recommended that you set it at the CAMVSYNC "L" state first and the CAMVSYNC information can be read from the status SFR. All command include ImgCptEn, is valid at CAMVSYNC falling edge. But be careful that except for first SFR setting, all command should be programmed in an ISR (Interrupt Service Routine). Especially, capture operation should be disabled when related information for target size are changed.

It's important to know that frame synchronization interrupt is controlled by frame sync interrupt enable, regardless of ImgCptEn enable or disable.

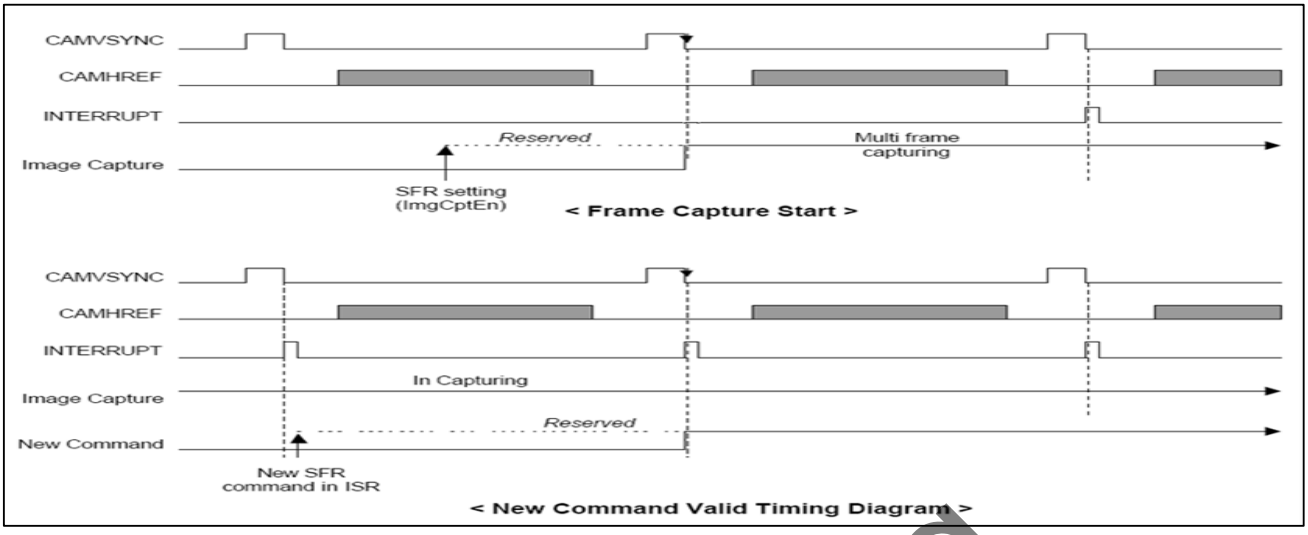


Figure 11-11: Timing Diagram for Register Setting

Internal FIFO self-reset

At the beginning of every frame or field, CAMIF will trigger the DMA to start the data transfer, and before the transfer, CAMIF will reset the internal FIFO. In ISR program, CPU should check the P-Path-DMA-Success(CICPTSTATUS[2]) and C-Path-DMA-Success(CICPTSTATUS[1]), and then write 1 to clear it.

Color Space Matrix

In preview path, the incoming YCbCr image data will be transformed to RGB image data via followed matrix formula:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} \times \begin{bmatrix} coef11 & coef12 & coef13 \\ coef21 & coef22 & coef23 \\ coef31 & coef32 & coef33 \end{bmatrix} \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} oft\_b \\ 128 \\ 128 \end{bmatrix} \right) + \begin{bmatrix} oft\_a \\ 0 \\ 0 \end{bmatrix}$$

Where, matrix coefficient coef11, coef12, coef13, coef21, coef22, coef23, coef31, coef32 and coef33 are signed numbers, and range from -1024 to +1023. Matrix coefficient oft_a and oft_b are unsigned numbers and range from 0 to 31.

Windows Cut and Preview Scale Down

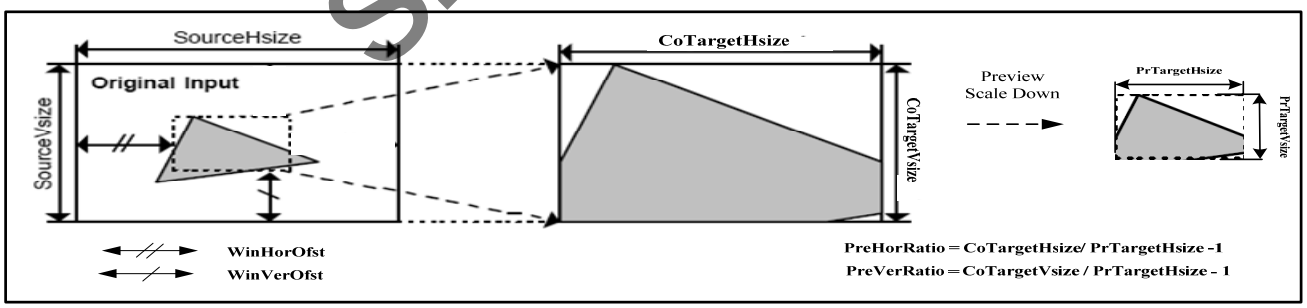


Figure 11-12: Camera Windows Cut and Preview Scale Down

As shown in Figure 11-12 , the image size SourceHsize and SourceVsize of input image is satisfied with following formula:

- SourceHsize > CoTargetHsize + WinHorOfst;
- SourceVsize > CoTargetVsize + WinVerOfst;

So if WinHorOfst and WinVerOfst are both set to zero, image size of codec path is the same as input image.

For preview path, the image can be scaled down, but you should set the proper image size PrTargetHsize and PrTargetVsize in register CIPRTRGFMT. Once CoTargetHsize, CoTargetVsize, PrTargetHsize and PrTargetVsize have been set, PreHorRatio and PreVerRatio can be calculated through the formula in Figure 11-12.

Interrupts

CAMIF interrupt consist of two type interrupt, preview path interrupt and codec path interrupt, which is controlled by separable interrupt enable and shows in Figure 11-13. Both preview path interrupt and codec path interrupt have some independent interrupt sources:

- Preview Path Interrupt
  - Frame synchronization interrupt, once enable will periodically generate frame sync interrupt when ITU-R BT 601/656 input valid.
  - DMA done interrupt, preview path DMA has finish transfer.
  - Smart interrupt, preview path will generate a interrupt between DMA over and frame synchronization.
  - ITU656 bad code interrupt, can't catcher right ITU656 reference code.
  - FIFO overflow interrupt, preview FIFO (CH4 and CH3) occurs overflow.
- Codec Path Interrupt
  - Frame synchronization interrupt, the same interrupt of preview path.
  - DMA done interrupt, codec path DMA has finish transfer.
  - Smart interrupt, the same as interrupt of preview path
  - ITU656 bad code interrupt, the same interrupt of preview path.
  - FIFO overflow interrupt, codec FIFO (CH2, CH1 and CH0) occur overflow.

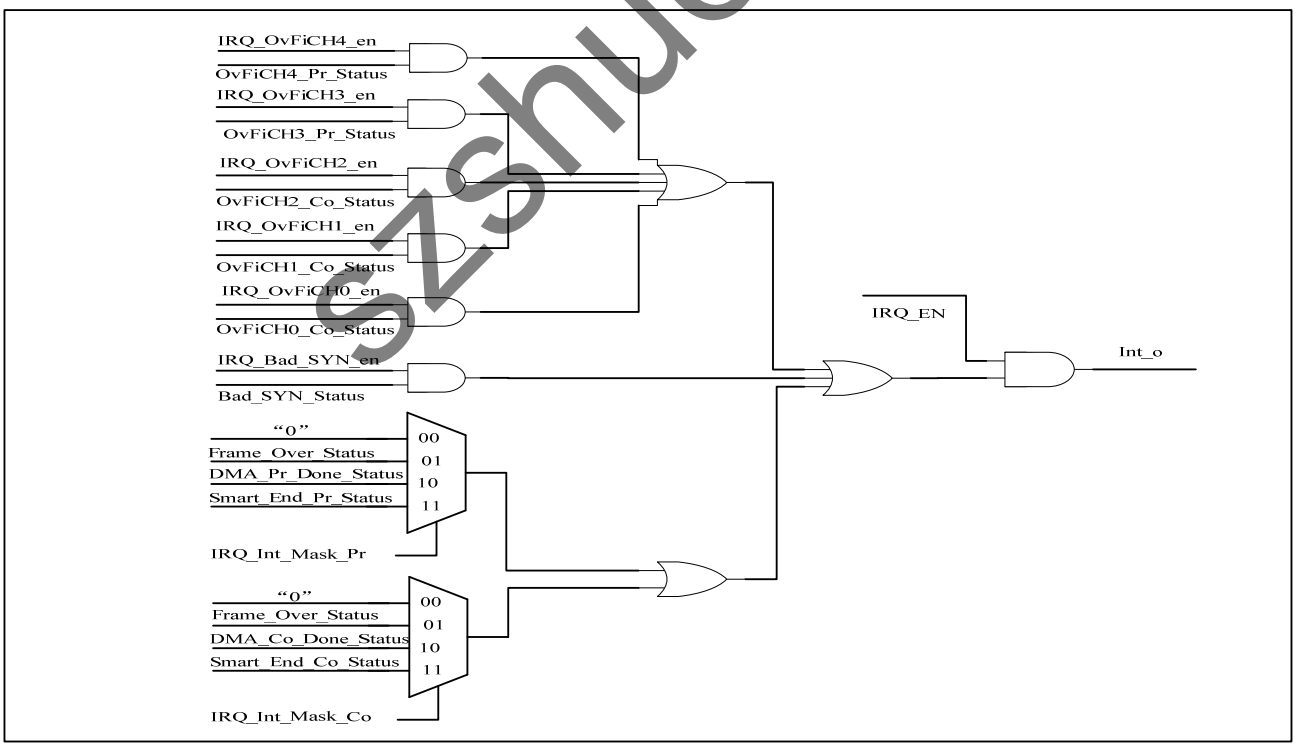


Figure 11-13: Camif Interrupt Generator

11.4 Camera Interface Register Description

11.4.1 Camera Interface Register Memory Map

Table 11-3: Camera Interface Register Memory Map

Address	Symbol	Direction	Description
0x20CC_0000	CISRCFMT	R/W	Input source format register
0x20CC_0004	CIWDOFST	R/W	Window Option Register
0x20CC_0008	CIGCTRL	R/W	Global Control Register
0x20CC_0044	CICOTRGSIZE	R/W	Codec Target Size Register
0x20CC_0048	CICOTRGFMT	R/W	Codec Target Format Register
0x20CC_0078	CIPRTRGSIZE	R/W	Preview Target Format Register
0x20CC_007C	CIPRTRGFMT	R/W	Preview Target Format Register
0x20CC_0084	CIPRSCPRERATIO	R/W	Preview Scaler Control Register
0x20CC_00A0	CIIMGCPT	R/W	Image Capture Enable Register
0x20CC_00A4	CICPTSTATUS	RW1C	Camera Capture Status Register
0x20CC_00C0	CIPRFIFOSTATUS	R	Camera capture Preview Path FIFO Status Register
0x20CC_00C4	CICOFIFOSTATUS	R	Camera capture Codec Path FIFO Status Register
0x20CC_00D0	CICOEF11	R/W	Camera Color Matrix Coefficient11 Register
0x20CC_00D4	CICOEF12	R/W	Camera Color Matrix Coefficient12 Register
0x20CC_00D8	CICOEF13	R/W	Camera Color Matrix Coefficient13 Register
0x20CC_00DC	CICOEF21	R/W	Camera Color Matrix Coefficient21 Register
0x20CC_00E0	CICOEF22	R/W	Camera Color Matrix Coefficient22 Register
0x20CC_00E4	CICOEF23	R/W	Camera Color Matrix Coefficient23 Register
0x20CC_00E8	CICOEF31	R/W	Camera Color Matrix Coefficient31 Register
0x20CC_00EC	CICOEF32	R/W	Camera Color Matrix Coefficient32 Register
0x20CC_00F0	CICOEF33	R/W	Camera Color Matrix Coefficient33 Register
0x20CC_00F4	CICOMC	R/W	Camera Color Matrix Configure Register
0x20CC_00AC	CH4FIFOPOPCNT	RO	CH4 FIFO Pop counter per frame/field, 64-bit
0x20CC_00B0	CH3FIFOPOPCNT	RO	CH3 FIFO Pop counter per frame/field, 64-bit
0x20CC_00B4	CH2FIFOPOPCNT	RO	CH2 FIFO Pop counter per frame/field, 64-bit
0x20CC_00B8	CH1FIFOPOPCNT	RO	CH1 FIFO Pop counter per frame/field, 64-bit
0x20CC_00BC	CH0FIFOPOPCNT	RO	CH0 FIFO Pop counter per frame/field, 64-bit
0x20CC_0050	CH4FIFOPUSHCNT	RO	CH4 FIFO Push counter per frame/field, 64-bit
0x20CC_0054	CH3FIFOPUSHCNT	RO	CH3 FIFO Push counter per frame/field, 64-bit
0x20CC_0058	CH2FIFOPUSHCNT	RO	CH2 FIFO Push counter per frame/field, 64-bit
0x20CC_005C	CH1FIFOPUSHCNT	RO	CH1 FIFO Push counter per frame/field, 64-bit
0x20CC_0060	CH0FIFOPUSHCNT	RO	CH0 FIFO Push counter per frame/field, 64-bit
0x20CC_0064	PPATHPIXELCNT	RO	P-Path pixel counter per frame/field
0x20CC_0068	CPATHPIXELCNT	RO	C-Path pixel counter per frame/field
0x20CC_0100	CH0DMAFB1	R/W	CH0 DMA Channel Frame Buffer 1 Register
0x20CC_0104	CH0DMACC1	R/W	CH0 DMA Channel Control 1 Register
0x20CC_0108	CH0DMAFB2	R/W	CH0 DMA Channel Frame Buffer 2 Register
0x20CC_010C	CH0DMACC2	R/W	CH0 DMA Channel Control 2 Register
0x20CC_0110	CH0DMAFB3	R/W	CH0 DMA Channel Frame Buffer 3 Register
0x20CC_0114	CH0DMACC3	R/W	CH0 DMA Channel Control 3 Register
0x20CC_0118	CH0DMAFB4	R/W	CH0 DMA Channel Frame Buffer 4 Register
0x20CC_011C	CH0DMACC4	R/W	CH0 DMA Channel Control 4 Register
0x20CC_0120	CH1DMAFB1	R/W	CH1 DMA Channel Frame Buffer 1 Register
0x20CC_0124	CH1DMACC1	R/W	CH1 DMA Channel Control 1 Register
0x20CC_0128	CH1DMAFB2	R/W	CH1 DMA Channel Frame Buffer 2 Register

0x20CC_012C	CH1DMACC2	R/W	CH1 DMA Channel Control 2 Register
0x20CC_0130	CH1DMAFB3	R/W	CH1 DMA Channel Frame Buffer 3 Register
0x20CC_0134	CH1DMACC3	R/W	CH1 DMA Channel Control 3 Register
0x20CC_0138	CH1DMAFB4	R/W	CH1 DMA Channel Frame Buffer 4 Register
0x20CC_013C	CH1DMACC4	R/W	CH1 DMA Channel Control 4 Register
0x20CC_0140	CH2DMAFB1	R/W	CH2 DMA Channel Frame Buffer 1 Register
0x20CC_0144	CH2DMACC1	R/W	CH2 DMA Channel Control 1 Register
0x20CC_0148	CH2DMAFB2	R/W	CH2 DMA Channel Frame Buffer 2 Register
0x20CC_014C	CH2DMACC2	R/W	CH2 DMA Channel Control 2 Register
0x20CC_0150	CH2DMAFB3	R/W	CH2 DMA Channel Frame Buffer 3 Register
0x20CC_0154	CH2DMACC3	R/W	CH2 DMA Channel Control 3 Register
0x20CC_0158	CH2DMAFB4	R/W	CH2 DMA Channel Frame Buffer 4 Register
0x20CC_015C	CH2DMACC4	R/W	CH2 DMA Channel Control 4 Register
0x20CC_0160	CH3DMAFB1	R/W	CH3 DMA Channel Frame Buffer 1 Register
0x20CC_0164	CH3DMACC1	R/W	CH3 DMA Channel Control 1 Register
0x20CC_0168	CH3DMAFB2	R/W	CH3 DMA Channel Frame Buffer 2 Register
0x20CC_016C	CH3DMACC2	R/W	CH3 DMA Channel Control 2 Register
0x20CC_0170	CH3DMAFB3	R/W	CH3 DMA Channel Frame Buffer 3 Register
0x20CC_0174	CH3DMACC3	R/W	CH3 DMA Channel Control 3 Register
0x20CC_0178	CH3DMAFB4	R/W	CH3 DMA Channel Frame Buffer 4 Register
0x20CC_017C	CH3DMACC4	R/W	CH3 DMA Channel Control 4 Register
0x20CC_0180	CH4DMAFB1	R/W	CH4 DMA Channel Frame Buffer 1 Register
0x20CC_0184	CH4DMACC1	R/W	CH4 DMA Channel Control 1 Register
0x20CC_0188	CH4DMAFB2	R/W	CH4 DMA Channel Frame Buffer 2 Register
0x20CC_018C	CH4DMACC2	R/W	CH4 DMA Channel Control 2 Register
0x20CC_0190	CH4DMAFB3	R/W	CH4 DMA Channel Frame Buffer 3 Register
0x20CC_0194	CH4DMACC3	R/W	CH4 DMA Channel Control 3 Register
0x20CC_0198	CH4DMAFB4	R/W	CH4 DMA Channel Frame Buffer 4 Register
0x20CC_019C	CH4DMACC4	R/W	CH4 DMA Channel Control 4 Register

11.4.2 Camera Interface Registers and Field Descriptions

Register 11-1: Input Source Format Register （CISRCFMT）

Field	Symbol	Direction	Description	Default
[31]	ITU601_656	R/W	0 = ITU-R BT.656 YCbCr 8-bit mode enable 1 = ITU-R BT.601 YCbCr 8-bit mode enable	0
[30]	UVOFFSET	R/W	Cb,Cr Value Offset Control. 0 = +0 (normally used) - for YCbCr 1 = +128 - for YUV	0
[29]	SCANMODE	R/W	0 = progressive mode 1 = interlaced mode	0
[15:14]	ORDER422	R/W	Input YcbCr format 00 = YCbYCr 01 = YCrYCb 10 = CbYCrY 11 = CrYCbY	0

Register 11-2: Window Option Register （CIWDOFST）

Field	Symbol	Direction	Description	Default
[26:16]	WinHorOfst	R/W	Window Horizontal Offset, if	0
[10:0]	WinVerOfst	R/W	Window Vertical Offset	0

Field	Symbol	Direction	Description	Default
[31]	OVFICH4_EN		Channel 4 FIFO overflow interrupt enable signal. 0 = disable 1 = enable	
[30]	OVFICH3_EN		Channel 3 FIFO overflow interrupt enable signal. 0 = disable 1 = enable	
[29]	OVFICH2_EN		Channel 2 FIFO overflow interrupt enable signal. 0 = disable 1 = enable	
[28]	OVFICH1_EN		Channel 1 FIFO overflow interrupt enable signal. 0 = disable 1 = enable	
[27]	OVFICH0_EN		Channel 0 FIFO overflow interrupt enable signal. 0 = disable 1 = enable	
[26]	DEBUG_EN	R/W	Debug mode. Counter the pops of DMA	
[25]	Reserved	R/W	Always be 0	
[24]	IEN	R/W	Interrupt enable signal, interrupt will be generated when capturing pictures. 0 = disable 1 = enable	0
[23:22]	IPRMASK	R/W	Interrupt mask of preview path 0 = disable 1 = use frame end status as interrupt source 2 = use DMA done status as interrupt source 3 = smart interrupt, use DMA done status or frame end status, interrupt will be generated as early as possible	0
[21:20]	ICOMASK	R/W	Interrupt mask of codec path 0 = disable 1 = use frame end status as interrupt source 2 = use DMA done status as interrupt source 3 = smart interrupt, use DMA done status or frame end status, interrupt will be generated as early as possible	0
[19]	CERR656IEN	R/W	ITU656 input reference code error interrupt enable 0 = disable 1 = enable	0
[05]	INVCAMPCLK	R/W	0 = Normal 1 = Inverse the polarity of CAMPCLK	0
[04]	INVCAMVSYNC	R/W	0 = Normal 1 = Inverse the polarity of CAMVSYNC	0
[03]	INVCAMHREF	R/W	0 = Normal 1 = Inverse the polarity of CAMHREF	0
[02]	INVCAMVSYNC	R/W	0 = Normal 1 = Inverse the polarity of CAMVSYNC	0
[01]	CAMRESET	R/W	External camera processor reset or power down signal.	0
[00]	SWRESET	R/W	Camera interface software reset, active high. Write 1 will reset all register to initial value, clear internal status flag and reset internal logic.	0

Register 11-4: Codec Target Format Register （CICOTRGSIZE）

Field	Symbol	Direction	Description	Default
[28:16]	CoTargetHsize	R/W	Horizontal pixel number of target image for codec DMA (multiple of 4)	0
[12:0]	CoTargetVsize	R/W	Vertical pixel number of target image for codec DMA (multiple of 2)	0

Register 11-5: Codec Target Format Register （CICOTRGFMT）

Field	Symbol	Direction	Description	Default
[31]	CO422	R/W	0 = YcbCr 4:2:0 codec image formats. 1 = YcbCr 4:2:2 codec image formats.	0
[30]	HWSWAP	R/W	Half-word(16 bit) swap in a word(32bit)	0
[29]	BTSWAP	R/W	Byte(8bit) swap in a Half-word(16bit)	0
[16:15]	STFMT	R/W	0 = Planer (Y->CH2, Cb->CH1, Cr-CH0) 1 = Semi-Planar (Y->CH2, CbCr->CH1) 2 = Interleaved(YCbYCr->CH2) (4:2:2 Only) 3 = Reserved	0

Register 11-6: Preview Target Format Register （CIPRTRGSIZE）

Field	Symbol	Direction	Description	Default
[28:16]	PrTargetHsize	R/W	Horizontal pixel number of target image for preview DMA (multiple of 4)	0
[12:0]	PrTargetVsize	R/W	Vertical pixel number of target image for preview DMA (multiple of 2)	0

Register 11-7: Preview Target Format Register （CIPRTRGFMT）

Field	Symbol	Direction	Description	Default
[30]	HWSWAP	R/W	Half-word(16 bit) swap in a word(32bit)	0
[29]	BTSWAP	R/W	Byte(8bit) swap in a Half-word(16bit)	0
[15]	BPP24BL	R/W	RGB 24-bit endian format. 0 : little-endian 1: big-endian	0
[14]	BPP16FMT	R/W	RGB 16-bit format 0 = RGB565 1 = RGB5551 (LSB = '0')	0
[1:0]	STFMT	R/W	0 = 16-bit RGB (RGB->CH4) 1 = 24-bit RGB (RGB->CH4) 2 = Semi-Planar 4:2:0 (Y->CH4, CbCr->CH3) 3 = Semi-Planar 4:2:2 (Y->CH4, CbCr->CH3) (NOTE: if sotre format = 3 or 4, please set MATRIX_CFG[30] to 1.)	0

Register 11-8: Preview Scaler Control Register （CIPRSCR）

Field	Symbol	Direction	Description	Default
[22:16]	PreHorRatio	R/W	Horizontal ratio of preview scaler, the value is dependent on CoTargetHsize and PrTargetHsize. PreHorRatio = CoTargetHsize/ PrTargetHsize -1	0
[6:0]	PreVerRatio	R/W	Vertical ratio of preview scaler, the value is dependent on	0

			CoTargetVsize and PrTargetHsize. PreVerRatio = CoTargetVsize / PrTargetHsize - 1	

Register 11-9: Image Capture Enable Register （CIIMGCPT）

Field	Symbol	Direction	Description	Default
[31]	CAMIFEn	R/W	Camera interface global enable bit.	0
[30]	CoEN	R/W	Codec path enable bit.	0
[29]	PrEN	R/W	Preview path enable bit.	0
[28]	OneShot_Co	R/W	Capture mode for C-PATH 0=capture continuing movie 1=capture a piece of picture	0
[27]	OneShot_Pr	R/W	Capture mode for P-PATH 0=capture continuing movie 1=capture a piece of picture	0

Register 11-10: Camera Capture Status Register （CICPTSTATUS）

Field	Symbol	Direction	Description	Default
[31]	OvFiCH4_Pr	RW1C	Preview Path CH4 FIFO overflow flag bit. When occur overflow, this bit will be set and write 1 to clear. If OVFICH4_EN is enabled, will generate interrupt.	0
[30]	OvFiCH3_Pr	RW1C	Preview Path CH3 FIFO overflow flag bit. When occur overflow, this bit will be set and write 1 to clear. If OVFICH3_EN is enabled, will generate interrupt.	0
[29]	OvFiCH2_Co	RW1C	Codec Path CH2 FIFO overflow flag bit. When occur overflow, this bit will be set and write 1 to clear. If OVFICH2_EN is enabled, will generate interrupt.	0
[28]	OvFiCH1_Co	RW1C	Codec Path CH1 FIFO overflow flag bit. When occur overflow, this bit will be set and write 1 to clear. If OVFICH1_EN is enabled, will generate interrupt.	0
[27]	OvFiCH0_Co	RW1C	Codec Path CH0 FIFO overflow flag bit. When occur overflow, this bit will be set and write 1 to clear. If OVFICH0_EN is enabled, will generate interrupt.	0
[26]	UnFiCH4_Pr	RW1C	Preview Path CH4 FIFO underflow flag bit	0
[25]	UnFiCH3_Pr	RW1C	Preview Path CH4 FIFO underflow flag bit	0
[24]	UnFiCH2_Co	RW1C	Preview Path CH4 FIFO underflow flag bit	0
[23]	UnFiCH1_Co	RW1C	Preview Path CH4 FIFO underflow flag bit	0
[22]	UnFiCH0_Co	RW1C	Preview Path CH4 FIFO underflow flag bit	0
[21]	P_PATH_FIFOS_DIRTY	RW1C	There is some data have not been transferred when the frame end	0
[20]	C_PATH_FIFOS_DIRTY	RW1C	There is some data have not been transferred when the frame end	0
[19]	CERR656	RW1C	CCIR656 input reference code exist error and can't be corrected. When occur error, this bit will be set and clear by read this register. If CERR656IEN is enabled, will generate interrupt.	0
[18]	P_PATH_LEISURE	RO	The time between DMA transfer the hole block and frame end	0
[17]	C_PATH_LEISURE	RO	The time between DMA transfer the hole block and frame end	0
[16]	DMA_CH4_ONCE	RW1C	Preview path DMA CH 4 has done its work once.	0

			Write 1 to clear	
[15]	DMA_CH3_ONCE	RW1C	Preview path DMA CH 3 has done its work once. Write 1 to clear	0
[14]	DMA_CH2_ONCE	RW1C	Codec path DMA CH 2 has done its work once. Write 1 to clear	0
[13]	DMA_CH1_ONCE	RW1C	Codec path DMA CH 1 has done its work once. Write 1 to clear	0
[12]	DMA_CH0_ONCE	RW1C	Codec path DMA CH 0 has done its work once. Write 1 to clear	0
[11]	DMA_CH4_TWICE	RW1C	Preview path DMA CH 4 has done its work once. Write 1 to clear	0
[10]	DMA_CH3_TWICE	RW1C	Preview path DMA CH 3 has done its work once. Write 1 to clear	0
[09]	DMA_CH2_TWICE	RW1C	Codec path DMA CH 2 has done its work once. Write 1 to clear	0
[08]	DMA_CH1_TWICE	RW1C	Codec path DMA CH 1 has done its work once. Write 1 to clear	0
[07]	DMA_CH0_TWICE	RW1C	Codec path DMA CH 0 has done its work once. Write 1 to clear	0
[06]	SMART_STATUS_PR	RW1C	Preview path data should has been transferred over Only valid when CIGCTRL[23:22] = 3, and write 1 to clear Int source	0
[05]	SMART_STATUS_CO	RW1C	Codec path data should has been transferred over Only valid when CIGCTRL[21:20] = 3, and write 1 to clear Int source	0
[04]	FRAME_STATUS_PR	RW1C	Preview path Frame over status Only valid when CIGCTRL[23:22] = 1, and write 1 to clear Int source	0
[03]	FRAME_STATUS_CO	RW1C	Codec path Frame over status Only valid when CIGCTRL[21:20] = 1, and write 1 to clear Int source	0
[02]	P_PATH_DMA_SUCCESS	RW1C	Preview path dma success Int source	0
[01]	C_PATH_DMA_SUCCESS	RW1C	Codec path dma success Int source	0
[00]	DMA_HAS_BEGIN_POP	RO	DMA has begin pop the data in fifos (only valid in DEBUG mode)	

Register 11-11-1: Camera Observe Status Register （CIOBSSTATUS）

Field	Symbol	Direction	Description	Default
[31:24]	Reserved	R	Reserved	0
[23]	CAMVSYNC	R	Camera VSYNC (This bit can be referred by CPU for first SFR setting. And, it can be seen in the ITU-R BT 656 mode, extract frame sync according to reference code.)	0
[22]	Reserved	R	Reserved	0
[21]	CoEN_sync	R	Camera internal valid codec path enable bit, sync to frame synchronization signal	0
[20]	PrEN_sync	R	Camera internal valid preview path enable bit, sync to frame synchronization signal	0

Register 11-12: Camera Capture Preview Path FIFO Status Register （CIPRFIFOSTATUS）

Field	Symbol	Direction	Description	Default
[31:16]	CH4_FIFO_CNT	R	Number of data (word) in FIFO	0
[15:00]	CH3_FIFO_CNT	R	Number of data (word) in FIFO	0

Register 11-13: Camera Capture Codec Path FIFO Status Register （CICOFIFOSTATUS）

Field	Symbol	Direction	Description	Default
[31:16]	CH2_FIFO_CNT	R	Number of data (word) in FIFO	0
[15:08]	CH1_FIFO_CNT	R	Number of data (word) in FIFO	0
[07:00]	CH0_FIFO_CNT	R	Number of data (word) in FIFO	0

Register 11-14: Camera Color Matrix Coefficient11 Register （CICOEF11）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF11	R/W	Color space convert matrix coefficient coef11, coef11 is signed number and range from -1024 to +1023 Recommend value = +298 (dec)	0

Register 11-15: Camera Color Matrix Coefficient12 Register （CICOEF12）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF12	R/W	Color space convert matrix coefficient coef12, coef12 is signed number and range from -1024 to +1023 Recommend value = 0	0

Register 11-16: Camera Color Matrix Coefficient13 Register （CICOEF13）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF13	R/W	Color space convert matrix coefficient coef13, coef13 is signed number and range from -1024 to +1023 Recommend value = +409 (dec)	0

Register 11-17: Camera Color Matrix Coefficient21 Register （CICOEF21）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF21	R/W	Color space convert matrix coefficient coef21, coef21 is signed number and range from -1024 to +1023 Recommend value = +298 (dec)	0

Register 11-18: Camera Color Matrix Coefficient22 Register （CICOEF22）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF22	R/W	Color space convert matrix coefficient coef22, coef22 is signed number and range from -1024 to +1023 Recommend value = -100 (dec)	0

Register 11-19: Camera Color Matrix Coefficient23 Register （CICOEF23）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF23	R/W	Color space convert matrix coefficient coef23, coef23 is signed number and range from -1024 to +1023 Recommend value = -208 (dec)	0

Register 11-20: Camera Color Matrix Coefficient31 Register （CICOEF31）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF31	R/W	Color space convert matrix coefficient coef31, coef31 is signed number and range from -1024 to +1023 Recommend value = +298 (dec)	0

Register 11-21: Camera Color Matrix Coefficient32 Register （CICOEF32）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF32	R/W	Color space convert matrix coefficient coef32, coef32 is signed number and range from -1024 to +1023 Recommend value = +516 (dec)	0

Register 11-22: Camera Color Matrix Coefficient33 Register （CICOEF33）

Field	Symbol	Direction	Description	Default
[10:0]	CICOEF33	R/W	Color space convert matrix coefficient coef33, coef33 is signed number and range from -1024 to +1023 Recommend value = 0	0

Register 11-23: Camera Color Matrix Configure Register （CICOMC）

Field	Symbol	Direction	Description	Default
[31]	ToRGB	R/W	Matrix transform mode and always set to 1	0
[30]	Passby	R/W	Passby the convert matrix 0: use matrix to convert YCbCr to RGB 1: YCbCr directly output to RGB, i.e. R=Y, G=Cb, B=Cr	0
[29:28]	Reserved	R/W	Must be set to 0	0
[12: 8]	oft_b	R/W	Color space convert matrix coefficient oft_b, oft_b is unsigned number and range from 0 to 31 Recommend value = 16	0
[4: 0]	oft_a	R/W	Color space convert matrix coefficient oft_a, oft_a is unsigned number and range from 0 to 31 Recommend value = 0	0

Register 11-24: Chx DMA Channel Frame Buffer 1 Register (ChxDMAFB1)

Field	Symbol	Direction	Description
[31:0]	FrameAddress1	R/W	Chx DMA Channel frame buffer 1 start address.

Register 11-25: Chx DMA Channel Control 1 Register (ChxDMACC1)

Field	Symbol	Direction	Description
[31]	Chx_dma_en	R/W	Chx DMA Channel enable bit 0: disable, 1: enable
[30]	Chx_rst	W	Reset Chx DMA Channel, active high and self-clear by hardware. Note: when reset Chx DMA channel, codec path should also be disabled and restart.
[29:28]	FrameNumber	R	Chx DMA current transferring frame number. 00: Chx DMA transfer data to frame buffer 1; 01: Chx DMA transfer data to frame buffer 2; 10: Chx DMA transfer data to frame buffer 3;

			11: Chx DMA transfer data to frame buffer 4;
[27]	ChxNextFrameEn1	R/W	Next frame buffer start address indication bit and only valid when ChxAutoReload1 is set. If AutoReload1 is set and this bit is also set, Chx DMA will transfer next frame data to frame buffer 2. If AutoReload1 is set and this bit isn't set, Chx DMA still transfer next frame data to current frame buffer 1 and user should ensure processing current frame data in time.
[26]	ChxAutoReload1	R/W	Indicate transfer next frame data after current frame data transfer done, next frame buffer start address is dependent on ChxNextFrameEn1. 1: auto reload 0: stop transfer after current frame data transfer done
[25]	FrameRead	R/W	Indicate DMA Transfer data to frame buffer and this bit always be set to 1.
[23:0]	FrameLen1	R/W	Byte length of frame 1 transfer data.

Note: If users want to use two ping-pong frame buffers, you can set the same buffer address to frame buffer 1 and 3, also set the same buffer address to frame buffer 2 and 4.

Register 11-26: Chx DMA Channel Frame Buffer 2 Register (ChxDMAFB2)

Field	Symbol	Direction	Description
[31:0]	FrameAddress2	R/W	Chx DMA Channel frame buffer 2 start address.

Register 11-27: Chx DMA Channel Control 2 Register (ChxDMACC2)

Field	Symbol	Direction	Description
[27]	ChxNextFrameEn2	R/W	Next frame buffer start address enable bit If ChxAutoReload2 is set and this bit is also set, Chx DMA will transfer next frame data to frame buffer 3. If this bit isn't set, regardless of ChxAutoReload2, Chx DMA will stop transfer after current frame data transfer done.
[26]	ChxAutoReload2	R/W	Indicate transfer next frame data after current frame data transfer done, next frame buffer start address is dependent on ChxNextFrameEn2. 1: auto reload 0: stop transfer after current frame data transfer done.
[25]	FrameRead	R/W	Indicate DMA Transfer data to frame buffer and this bit always be set to 1.
[24:0]	FrameLen2	R/W	Byte length of frame 2 transfer data.

Register 11-28: Chx DMA Channel Frame Buffer 3 Register (ChxDMAFB3)

Field	Symbol	Direction	Description
[31:0]	FrameAddress3	R/W	Chx DMA Channel frame buffer 3 start address.

Register 11-29: Chx DMA Channel Control 3 Register (ChxDMACC3)

Field	Symbol	Direction	Description
[27]	ChxNextFrameEn3	R/W	Next frame buffer start address enable bit If ChxAutoReload3 is set and this bit is also set, Chx DMA will transfer next frame data to frame buffer 4. If this bit isn't set, regardless of ChxAutoReload3, Chx DMA will stop transfer after current frame data transfer done.
[26]	ChxAutoReload3	R/W	Indicate transfer next frame data after current frame data transfer done, next frame buffer start address is dependent on ChxNextFrameEn3. 1: auto reload 0: stop transfer after current frame data transfer done.

[25]	FrameRead	R/W	Indicate DMA Transfer data to frame buffer and this bit always be set to 1.
[24:0]	FrameLen3	R/W	Byte length of frame 3 transfer data.

Register 11-30: Chx DMA Channel Frame Buffer 4 Register (ChxDMAFB4)

Field	Symbol	Direction	Description
[31:0]	FrameAddress4	R/W	Chx DMA Channel frame buffer 4 start address.

Register 11-31: Chx DMA Channel Control 4 Register (ChxDMACC4)

Field	Symbol	Direction	Description
[27]	ChxNextFrameEn4	R/W	Next frame buffer start address enable bit If ChxAutoReload4 is set and this bit is also set, Chx DMA will transfer next frame data to frame buffer 0. If this bit isn't set, regardless of ChxAutoReload4, Chx DMA will stop transfer after current frame data transfer done.
[26]	ChxAutoReload4	R/W	Indicate transfer next frame data after current frame data transfer done, next frame buffer start address is dependent on ChxNextFrameEn4. 1: auto reload 0: stop transfer after current frame data transfer done.
[25]	FrameRead	R/W	Indicate DMA Transfer data to frame buffer and this bit always be set to 1.
[24:0]	FrameLen4	R/W	Byte length of frame 4 transfer data.

Note: FrameLenx in register ChxDMACCy is calculated by Byte_length/8, where Byte_length should be multiple of 8, if not, please modify the image source xsize and ysize.

12 USB host Controller

12.1 USB Host Controller Overview

IMAPx210 supports 3-port USB host interface, which compliance following standards:

- Universal Serial Bus Specification (Revision 2.0, April 27, 2000)
- Enhanced Host Controller Interface Specification for Universal Serial Bus (Revision 1.0, March 12, 2002)
- openHCI: Open Host Controller Interface Specification for USB (Release 1.0a, September 14, 1999)
- EHCI 1.1 Addendum (Revision v0.6, October 2007)
- High-Speed Inter-Chip USB Electrical Specification (Version 1.0, September 27, 2007)
- Engineering Change Notice: USB 2.0 Link Power Management Addendum (July 16, 2007)

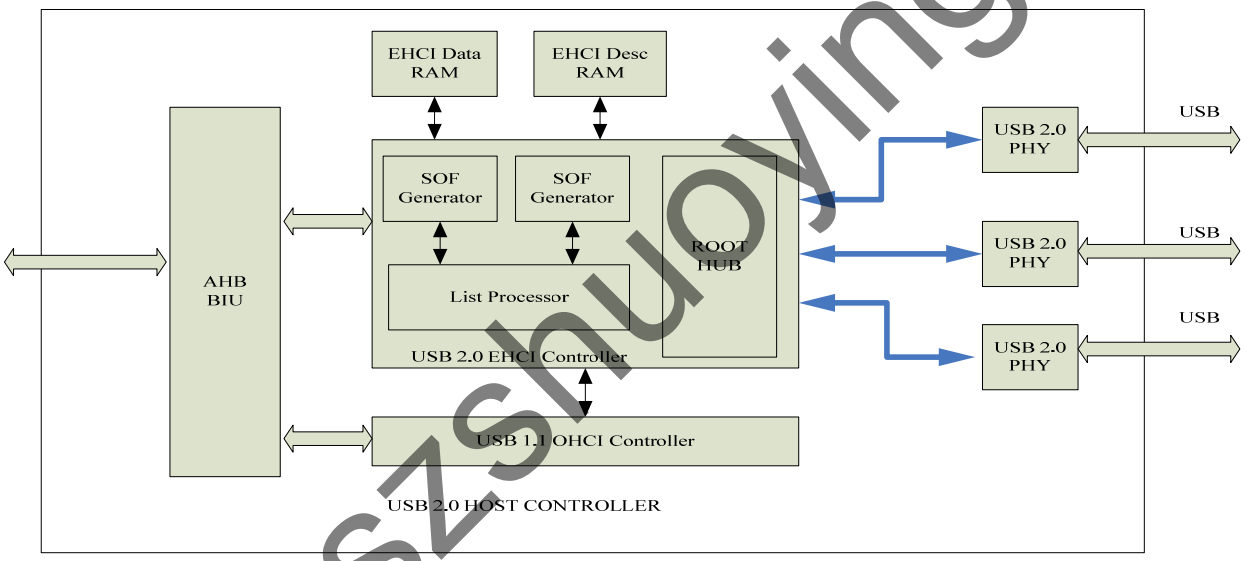


Figure 12-1: USB 2.0 Host Block Diagram

12.2 USB Host Function Descriptions

12.2.1 AHB BIU

The controller supports an AHB interface to the application. The BIU is divided into Master and Slave modules.

The Slave BIU provides the Enhanced Host Controller Driver (EHCD) interface to write to and read from the Operational registers through the AHB. For EHCI register access, two AHB clock cycle wait states are introduced. For OHCI register access, the number of wait states varies with the phase relationship between the AHB and 12 MHz clocks during synchronization.

The Master BIU block receives requests from the List Processor and reads data from or writes data to system memory through the AHB. This block fetches descriptors, data, and status.

12.2.2 List Processor

The List Processor block is the main controller. This block is implemented with multiple state machines to perform the list service flow, which is set up by the Host Controller Driver according to the priority set in the Operational registers. In addition, this block is implemented with a controller that interfaces with the BIU, Packet Buffer, EHCI Operational registers, SOF Generators, and the Root Hub.

12.2.3 Operational Registers

The EHCI Capability and Operational registers are stored in this block. The registers in the Auxiliary Well are part of the Operational registers, but are implemented in a separate module.

12.2.4 Start-of-Frame (SOF) Generator

SOF packets are generated with an SOF counter to generate micro-SOFs for each microframe. Microframe duration is derived from the Frame Length Adjustment (FLADJ) register value, which can be configured through the Application Strap Signals interface. The same FLADJ values must be configured through strap signals. This ensures that the host microframe duration and per-port microframe duration remain the same.

12.2.5 Root Hub

The Root Hub interfaces between the List Processor and the USB PHY. The Root Hub delivers token/data packets to the USB from the List Processor and the data/handshake response packets from the USB to the List Processor. The Root Hub UTMI complies to the USB 2.0 Transceiver Macrocell Interface (UTMI) Specification, Revision 1.05. The Root Hub also provides an interface to the OHCI Host Controller, which is DesignWare USB 1.1 OHCI Host Controller. The OHCI is independent of the EHCI Host Controller and is integrated into the Root Hub with no modification. The Root Hub is implemented with Port Router logic to route the ports to either an EHCI Host Controller or an OHCI Host Controller. The Root Hub is configured to detect if the attached device is high-speed or full/low speed. For example, after a USB reset, if the reset port generates chirp, the Root Hub marks it as a high-speed port and, after enabling the port, the Root Hub routes the port to the EHCI Host Controller. If the reset port does not generate a chirp, the port is marked as a disconnect and routed to the OHCI Host Controller. The OHCI now has ownership of the port and the device is enumerated under the OHCI. Any disconnected port from the OHCI Host Controller is routed back to the EHCI Host Controller. The Root Hub also generates start-of-microframe operations at every 125-µs interval.

12.2.6 DataRAM/DescRAM

The data RAM and DescRAM store IN/OUT data and control descriptor.

12.3 USB Host Register Description

12.3.1 USB Host Register Memory Map

Address	Symbol	Direction	Description
0x20C8_0000	HcRevision	RO	Host Controller Revision Register[1)]
0x20C8_0004	HcControl	R/W	Host Controller Control Register[1)]
0x20C8_0008	HcCommandStatus	R	Host Controller Command and Status Register[1)]
0x20C8_000C	HcInterruptStatus	R/W	Host Controller Interrupt Status Register[1)]
0x20C8_0010	HcInterruptEnable	R/WS	Host Controller Interrupt Enable Register[1)]
0x20C8_0014	HcInterruptDisable	R/WS	Host Controller Interrupt Disable Register[1)]
0x20C8_0018	HcHCCA	R/W	Host Controller Communication Area Register[1)]
0x20C8_001C	HcPeriodCurrentED	R/W	Host Controller Current Endpoint Descriptor address Register[1)]
0x20C8_0020	HcControlHeadED	R/W	Host Controller First Endpoint Descriptor of the Control list[1)]

0x20C8_0024	HcControlCurrentED	R/W	Host Controller Current Endpoint Descriptor of the Control list[1)]
0x20C8_0028	HcBulkHeadED	R/W	Host Controller First Endpoint Descriptor of the Bulk list [1)]
0x20C8_002C	HcBulkCurrentED	R/W	Host Controller Current Endpoint of the Bulk list[1)]
0x20C8_0030	HcDoneHead	R/W	Host Controller Last Completed Transfer Descriptor[1)]
0x20C8_0034	HcFmInterval	R/W	Host Controller Frame interval [1)]
0x20C8_0038	HcFmRemaining	R/W	Host Controller Frame remaining [1)]
0x20C8_003C	HcFmNumber	R/W	Host Controller Frame Number[1)]
0x20C8_0040	HcPeriodicStart	R/W	Host Controller Periodic Start[1)]
0x20C8_0044	HcLSThreshold	R/W	Host Controller Low Speed Threshold[1)]
0x20C8_0048	HcRhDescriptorA	R/W	Host Controller Root Hub DescriptorA[1)]
0x20C8_004C	HcRhDescriptorB	R/W	Host Controller Root Hub DescriptorB[1)]
0x20C8_0050	HcRhStatus	R/W	Host Controller Root Hub Status[1)]
0x20C8_0054	HcRhPortStatus[1]	R/W	Host Controller Root Hub Downstream Post Status 1[1)]
0x20C8_0058	HcRhPortStatus[2]	R/W	Host Controller Root Hub Downstream Post Status 2[1)]
0x20C8_0060	HcRhPortStatus[3]	R/W	Host Controller Root Hub Downstream Post Status 3[1)]
0x20C8_1000	HCCAPBASE	R/W	USB Host EHC Capability register
0x20C8_1004	HCSPARAMS	R/W	USB Host EHC Structural Parameters
0x20C8_1008	HCCPARAMS	R/W	USB Host EHC Capability Parameters
0x20C8_1400	HUSBCMD	R/W	USB Host EHC Command
0x20C8_1404	HUSBSTS	R/W	USB Host EHC Status
0x20C8_1408	HUSBINTR	R/W	USB Host EHC Interrupt Enable
0x20C8_140C	HFRINDEX	R/(C)	USB Host EHC Frame Index
0x20C8_1414	HPERIODICLISTBASE	R/W	USB Host EHC Periodic Frame List Base Address
0x20C8_1418	HASYNCLISTADDR	R/W	USB Host EHC Current Asynchronous List Address
0x20C8_1440	HCONFIGFLAG	R/W	USB Host EHC Configured Flag
0x20C8_1444	HPORTSC1	R/W	USB Host EHC Port Status and Control1
0x20C8_1448	EPORTSC2	R/W	USB Host EHC Port Status and Control2
0x20C8_144C	EPORTSC3	R/W	USB Host EHC Port Status and Control3

1) Refer to the *Open Host Controller Interface Specification for USB* (Release 1.0a, September 14, 1999), for more information on this registers.

12.3.2 USB Host Individual Register Description

Register 12-1: USB HOST EHC Command（USBHCMD, offset=0xC000）

Field	Symbol	Direction	Description	Default
[31:24]	Reserved	RO	Reserved	0x0
[23:16]	InthCtrl	R/W	**Interrupt Threshold Control** This field is used by system software to select the maximum rate at which the Host Controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are Reserved. **Value Maximum Interrupt Interval** 00h Reserved 01h 1 micro-frame 02h 2 micro-frames 04h 4 micro-frames 08h 8 micro-frames (default, equates to 1 ms) 10h 16 micro-frames (2 ms) 20h 32 micro-frames (4 ms) 40h 64 micro-frames (8 ms) Refer to Section 4.15 for interrupts affected by this register. Any other value in this register yields Reserved results. Software modifications to this bit while HCHalted bit is equal to zero	0x08

			results in Reserved behavior.	
[15:12]	Reserved	RO	Reserved	0x0
[11]	ASPME	R/W	**Asynchronous Schedule Park Mode Enable (OPTIONAL)** If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled.	0x0
[10]	Reserved	RO	Reserved	0x0
[9:8]	ASPMC	R/W	**Asynchronous Schedule Park Mode Count (OPTIONAL).** It contains a count of the number of successive transactions the Host Controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. See Section 4.10.3.2 for full operational details. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in Reserved behavior.	0x0
[7]	Reserved	RO	Reserved	0x0
[6]	Doorbell	R/W	**Interrupt on Async Advance Doorbell** This bit is used as a doorbell by software to tell the Host Controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to *ring* the doorbell. When the Host Controller has evicted all appropriate cached schedule state, it sets the *Interrupt on Async Advance* status bit in the USBSTS register. If the *Interrupt on Async Advance Enable* bit in the USBINTR register is a one then the Host Controller will assert an interrupt at the next interrupt threshold. The Host Controller sets this bit to a zero after it has set the *Interrupt on Async Advance* status bit in the USBSTS register to a one. Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield Reserved results.	0x0
[5]	ASEnable	R/W	**Asynchronous Schedule Enable** This bit controls whether the Host Controller skips processing the Asynchronous Schedule. 0b Do not process the Asynchronous Schedule 1b Use the ASYNCLISTADDR register to access the Asynchronous Schedule	0x0
[4]	PSEnable	R/W	**Periodic Schedule Enable** This bit controls whether the Host Controller skips processing the Periodic Schedule. 0b Do not process the Periodic Schedule 1b Use the PERIODICLISTBASE register to access the Periodic Schedule	0x0
[3:2]	Reserved	RO	Reserved	0x0
[1]	HCRESET	R/W	**Host Controller Reset** This control bit is used by software to reset the Host Controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. All operational registers, including port registers and port state machines are set to their initial values. Software must reinitialize the Host Controller in order to return the Host Controller to an	0x0

			operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software should not set this bit to a one when the *HCHalted* bit in the USBSTS register is a zero. Attempting to reset an actively running HostController will result in Reserved behavior.	
[0]	RS	R/W	**Run/Stop** 1=Run. 0=Stop. When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to a 1. When this bit is set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a one to this field unless the Host Controller is in the Halted state (i.e. *HCHalted* in the USBSTS register is a one). Doing so will yield Reserved results.	0x0

Register 12-2: USB Host EHC Status Register（USBHSTS, offset=0xC004）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	RO	Reserved	0x0
[15]	ASStatus	RO	**Asynchronous Schedule Status** The bit reports the current real status of the Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the *Asynchronous Schedule Enable* bit in the USBCMD register. When this bit and the *Asynchronous Schedule Enable* bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).	0x0
[14]	PSStatus	RO	**Periodic Schedule Status** The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit in the USBCMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).	0x0
[13]	Recl	RO	**Reclamation** This is a read-only status bit, which is used to detect an empty asynchronous schedule.	0x0
[12]	HCHalt	RO	**HCHalted** This bit is a zero whenever the Run/Stop bit is a one. The Host Controller sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g. internal error).	0x0
[11:6]	Reserved	RO	Reserved	0x0
[5]	AsynAdvInt	R/WC	**Interrupt on Asynchronous Advance** System software can force the Host Controller to issue an interrupt	0x0

			the next time the Host Controller advances the asynchronous schedule by writing a one to the *Interrupt on Async Advance Doorbell* bit in the USBCMD register. This status bit indicates the assertion of that interrupt source.	
[4]	SysErrInt	R/WC	**Host System Error** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs.	0x0
[3]	FLRInt	R/WC	**Frame List Rollover** The Host Controller sets this bit to a one when the *Frame List Index* rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the *Frame List Size* field of the USBCMD register) is 1024, the *Frame Index Register* rolls over every time FRINDEX[13] toggles.	0x0
[2]	PortChgInt	R/WC	**Port Change Detect** The Host Controller sets this bit to a one when any port for which the *Port Owner* bit is set to zero has a change bit transition from a zero to a one or a *Force Port Resume* bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the *Connect Status Change* being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's *Port Owner* bit.	0x0
[1]	USBERRInt	R/WC	**USB Error Interrupt** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and USBINT bit are set.	0x0
[0]	USBInt	R/WC	**USB Interrupt** The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes).	0x0

Register 12-3: USB Host EHC Interrupt Enable（USBHIER, offset=0xC008）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	RO	Reserved	0x0
[5]	AsynAdvInt_E	R/W	**Interrupt on Async Advance Enable.** 1= Enabled. 0=Disabled.	0x0
[4]	SysErrInt_E	R/W	**Host System Error Enable.** 1= Enabled. 0=Disabled	0x0
[3]	FLRInt_E	R/W	**Frame List Rollover Enable.** 1= Enabled. 0=Disabled.	0x0
[2]	PortChgInt_E	R/W	**Port Change Interrupt Enable** 1= Enabled. 0=Disabled.	0x0
[1]	USBERRInt_E	R/W	**USB Error Interrupt Enable** 1= Enabled. 0=Disabled.	0x0
[0]	USBInt_E	R/W	**USB Interrupt Enable** 1= Enabled. 0=Disabled.	0x0

Register 12-4: USB Host EHC Frame Index（HFRINDEX, offset=0xC00C）

Field	Symbol	Direction	Description	Default
[31:14]	Reserved	RO	Reserved	0x0
[13:0]	Findx	R/W	**Frame Index** The value in this register increments at the end of each time frame (e.g. micro-frame). Bits [N:3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index.	0x0

Register 12-5: USB Host EHC Periodic Frame List Base Address（HPERIODICLISTBASE, offset=0xC014）

Field	Symbol	Direction	Description	Default
[31:12]	PLBase	R/W	**Base Address (Low)** These bits correspond to memory address signals [31:12], respectively.	0x0
[11:0]	Reserved	RO	Reserved	0x0

Register 12-6: USB Host EHC Current Asynchronous List Address（HASYNCLISTADDR, offset=0xC018）

Field	Symbol	Direction	Description	Default
[31:5]	ALBase	R/W	**Link Pointer Low** These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).	0x0
[4:0]	Reserved	RO	Reserved	0x0

Register 12-7: USB Host EHC Configured Flag（HCONFIGFLAG, offset=0xC040）

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	RO	Reserved	0x0
[0]	CF	R/W	**Configure Flag** Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below. 0b Port routing control logic default-routes each port to an implementation dependent classic Host Controller. 1b Port routing control logic default-routes all ports to this Host Controller	0x0

Register 12-8: USB Host EHC Portx Status and Control（HPORTSCx, offset=0xC044~0xC04C）

Field	Symbol	Direction	Description	Default
[31:23]	Reserved	RO	Reserved	0x0
[22]	WKOC_E	R/W	**Wake on Over-current Enable** Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events.	0x0
[21]	WKDSCNNT_E	RO	**Wake on Disconnect Enable** Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events.	0x0
[20]	WKCNNT_E	R/W	**Wake on Connect Enable** Writing this bit to a one enables the port to be sensitive to device connects as wake-up events.	0x0

[29:16]	TestCtrl	R/WC	**Port Test Control** When this field is zero, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. The encoding of the test mode bits are (0110b - 1111b are Reserved): **Bits Test Mode** 0000b Test mode not enabled 0001b Test J_STATE 0010b Test K_STATE 0011b Test SE0_NAK 0100b Test Packet 0101b Test FORCE_ENABLE	0x0
[15:14]	Reserved	RO	Reserved	0x0
[13]	PortOwner	R/W	**Port Owner** This bit unconditionally goes to a 0b when the *Configured* bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the *Configured* bit is zero. System software uses this field to release ownership of the port to a selected Host Controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion Host Controller owns and controls the port.	0x1
[12]	Reserved	RO	Reserved	0x1
[11:10]	Line	RO	**Line Status** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits are: **Bits[11:10] USB State Interpretation** 00b SE0 Not Low-speed device, perform EHCI reset 10b J-state Not Low-speed device, perform EHCI reset 01b K-state Low-speed device, release ownership of port 11b Reserved Not Low-speed device, perform EHCI reset	0x0
[9]	Reserved	RO	Reserved	0x0
[8]	PortReset	R/W	**Port Reset** 1=Port is in Reset. 0=Port is not in Reset. When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Note: when software writes this bit to a one, it must also write a zero to the *Port Enable* bit. Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the Host Controller will automatically enable this port (e.g. set the *Port Enable* bit to a one). A Host Controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the Host Controller must have the port in the enabled state within 2ms of software writing this bit to a zero. The *HCHalted* bit in the USBSTS register should be a zero before	0x0

			software attempts to use this bit. The Host Controller may hold Port Reset asserted to a one when the *HCHalted* bit is a one.	
[7]	PortSusp	R/W	**Suspend** 1=Port in suspend state. 0=Port not in suspend state. Port Enabled Bit and Suspend bit of this register define the port states as follows: **Bits [Port Enabled, Suspend] Port State** 0X Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. A write of zero to this bit is ignored by the Host Controller. The Host Controller will unconditionally set this bit to a zero when: Software sets the *Force Port Resume* bit to a zero (from a one). Software sets the *Port Reset* bit to a one (from a zero). If host software sets this bit to a one when the port is not enabled (i.e. Port enabled bit is a zero) the results are Reserved.	0x0
[6]	PortResume	R/W	**Force Port Resume R/W** 1=Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. This functionality defined for manipulating this bit depends on the value of the *Suspend* bit. For example, if the port is not suspended (*Suspend* and *Enabled* bits are a one) and software transitions this bit to a one, then the effects on the bus are Reserved. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to a one. If software sets this bit to a one, the Host Controller must not set the *Port Change Detect* bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to the high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The Host Controller must complete this transition within 2 milliseconds of software setting this bit to a zero.	0x0
[5]	OverCurrentChg	R/WC	**Over-current Change** 1=This bit gets set to a one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position.	0x0
[4]	OverCurrent	RO	**Over-current Active** 1=This port currently has an over-current condition. 0=This port does not have an over-current condition. This bit will automatically transition from a one to a zero when the over current condition is removed.	0x0
[3]	PortEnChg	R/WC	**Port Enable/Disable Change**	0x0

			1=Port enabled/disabled status has changed. 0=No change. For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it.	
[2]	PortEn	RO	**Port Enabled/Disabled** 1=Enable. 0=Disable. Ports can only be enabled by the Host Controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The Host Controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other Host Controller and bus events. When the port is disabled (0b) downstream propagation of data is blocked on this port, except for reset.	0x0
[1]	ConnChg	R/WC	**Connect Status Change** 1=Change in Current Connect Status. 0=No change. Indicates a change has occurred in the port's Current Connect Status. The Host Controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). Software sets this bit to 0 by writing a 1 to it.	0x0
[0]	Conn	RO	**Current Connect Status** 1=Device is present on port. 0=No device is present. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.	0x0

13 USB OTG Controller

13.1 USB OTG Overview

USB On-The-Go (OTG) is a Dual-Role Device (DRD) controller, which is compliant with Universal Serial Bus Specification Revision 2.0 and On-The-Go Supplement to the USB 2.0 Specification Revision 1.0a. An OTG Dual-Role Device is a USB device which supports both host and peripheral functions. It also supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP). It supports high-speed (HS, 480-Mbps), full-speed (FS, 12-Mbps), and low-speed (LS, 1.5-Mbps, Host only) transfers. KeyPad Function descriptions

USB Peripheral Features

- Compliant with the Universal Serial Bus Specification Revision 2.0
- Support High-speed (480Mbit/s) and Full-speed (12Mbit/s) operation as a peripheral
- Configurable architecture
- Support USB transfers: control, bulk, interrupt, and isochronous
- Support all USB standard commands
- Support Class/Vendor commands by passing the commands to the application system
- Up to 16 alternate settings per interface
- Up to 7 logical Endpoints by default (including Endpoint0)

USB Host Features

- Adopt the Enhanced Host Controller Interface Specification for Universal Serial Bus Revision
- at operation of a USB2.0 Host
- Adopt the Universal Host Controller Interface (UHCI) Design Guide Revision 1.1 at operation of
- a USB1.1 Host
- Support High-speed (480Mbit/s), Full-speed (12Mbit/s), and Low-speed (1.5Mbit/s)
- Support USB transfers: control, bulk, interrupt, and isochronous
- Support split transaction

USB OTG Features

- Compliant with the On-The-Go Supplement to the USB 2.0 Specification Revision 1.0a
- UTMI+ level 3 interface compliant with the UTMI+ Specification Revision 1.0
- Targeted Peripheral List
- Session Request Protocol (SRP)
- Host Negotiation Protocol (HNP)
- Support single USB port

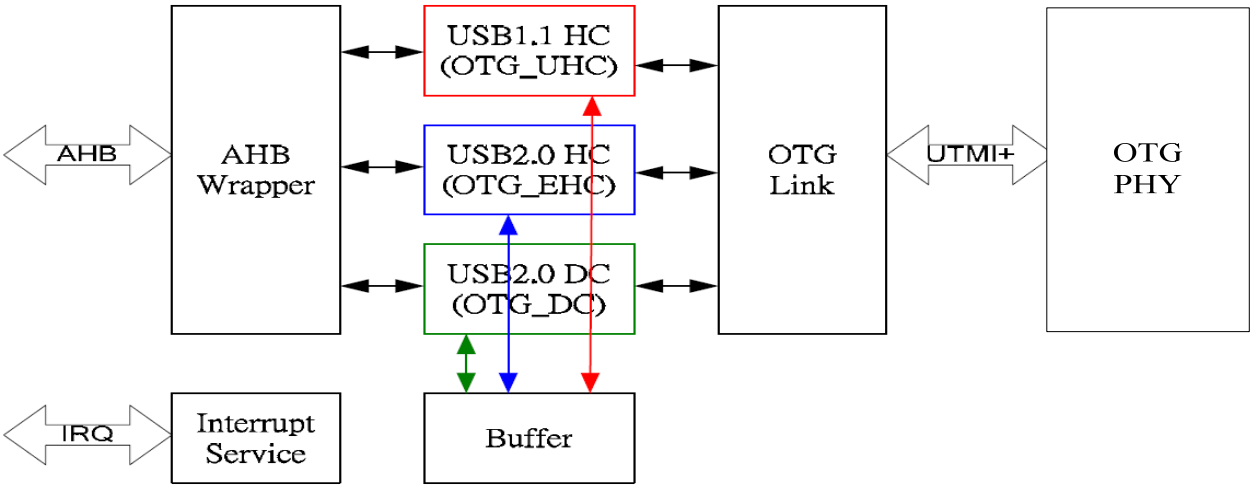


Figure 13-1: USB OTG 2.0 Block Diagram

13.2 USB2.0 OTG Function Descriptions

13.2.1 USB2.0 Host (OTG_EHC) Function

The OTG_DRD is a dual-role device containing USB2.0 Host function, USB1.1 Host function, USB2.0 Peripheral function, and OTG Link function. The OTG_EHC is an embedded USB2.0 Host Controller of the OTG dual-role device. It follows the Enhanced Host Controller Interface Specification for Universal Serial Bus Revision 1.0 (EHCI specification). The OTG_EHC is enabled when the OTG_DRD is operating at USB High-speed Host mode, and disabled at other modes. This section gives a function overview of the OTG_EHC core. Some contents are extracted from the EHCI specification in order to explain the OTG_EHC functions. Users can ignore those sections if they are familiar with the specification.

13.2.1.1 OTG_EHC Interrupt Events

The OTG_EHC provides interrupt capability based on a number of sources. There are several general groups of interrupt sources:

- Interrupts as a result of executing transactions from the schedule (success and error conditions)
- Host controller events (Port change events, etc.)
- Host Controller error events

All transaction-based sources are maskable through the Host Controller's Interrupt Enable register (USBINTR). Additionally, individual transfer descriptors can be marked to generate an interrupt on completion. This section describes each interrupt source and the processing that occurs in response to the interrupt.

During normal operation, interrupts may be immediate or deferred until the next interrupt threshold occurs. The interrupt threshold is a tunable parameter via the Interrupt Threshold Control field in the USBCMD register. The value of this register controls when the host controller will generate an interrupt on behalf of normal transaction execution. When a transaction completes during an interrupt interval period, the interrupt signaling the completion of the transfer will not occur until the end of interrupt threshold. For example, the default value is eight micro-frames. This means that the host controller will not generate interrupts any more frequently than once every eight micro-frames.

If an interrupt has been scheduled to be generated for the current interrupt threshold interval, the interrupt is not signaled until after the status for the last complete transaction in the interval has been written back to host memory. This may sometimes result in the interrupt not being signaled until the next interrupt threshold. Below lists the OTG_EHC interrupt events. Please see the EHCI Specification for details.

Transfer/Transaction base interrupts:

- Transaction Error (ex: CRC Error, Time-out, Bad PID, and Babble)
- Interrupt On Completion (IOC)
- Short Packet

Host Controller Event interrupts:

- Port Change Events (Connect Status Change, Port Enable/Disable Change, Over-current Change,Force Port Resume)
- Frame List Rollover
- Interrupt on Asynchronous Advance
- Host System Error

13.2.2 USB1.1 Host (OTG_UHC) Function Description

The OTG_UHC follows the Universal Host Controller Interface (UHCI) Design Guide Revision 1.1 (UHCI specification). The OTG_UHC is enabled when the OTG_DRD is operating at USB Full/Low-speed Host mode, and disabled at other modes. This section gives a function overview of the OTG_UHC. Some contents are extracted from the UHCI specification in order to explain the OTG_UHC functions.

13.2.2.1 OTG_UHC Interrupt Events

Interrupts are the direct communication method for OTG_UHC communication with the processor. There are two general groups of interrupt sources, those resulting from execution of transactions in the schedule, and those resulting from a Host Controller operation error. All transaction-based sources are maskable by the Host Controller Driver (HCD) through the Host Controller's Interrupt Enable register.Additionally, individual transfer descriptors can be marked to generate an interrupt on completion. This section describes each interrupt source and the processing that occurs in response to the interrupt. During normal operation, interrupts are deferred until the last transaction in the frame is complete. This results in the normal runtime interrupts being processed in a batch fashion. If an HC process error or host system error (both considered fatal) occur, the core halts and immediately issues a hardware interrupt to the system.

If an interrupt has been scheduled to be generated for the frame, the interrupt is not signaled until after the status for the last complete transaction in the frame has been written back to host memory. This may sometimes result in the interrupt not being signaled until after the Start Of Frame (SOF) for the next frame has been sent. This guarantees that the software can safely process through Frame List Current Index -1 when it is servicing an interrupt. Below lists the OTG_UHC interrupt events. Please see the UHCI Specification for details.

Transaction base interrupts:

- CRC Error / Time-out
- Interrupt On Completion (IOC)
- Short Packet
- Serial Bus Babble
- Stalled
- Data Buffer Error
- Bit Stuff Error

Non-Transaction base interrupts:

- Resume Received
- Host Controller Process Error
- Host System Error

13.2.3 USB2.0 Peripheral (OTG_DC) Function

The OTG_DC follows the Universal Serial Bus Specification Revision 2.0. The OTG_ core is enabled when the OTG_DRD is operating at USB High/Full-speed peripheral mode, and disabled at other modes. This section gives a function overview of the OTG_DC. Some contents are extracted from the USB2.0 specification in order to explain the OTG_DC functions.

13.2.3.1 OTG_DC InterruptEvents

Interrupt is a direct communication channel between the OTG_DC and the processer. There are three interrupt related registers:

- Interrupt Status Register (ISR)

The core reflects interrupt events in this register by setting '1' in the corresponding bits.
Software clears interrupt events by writing '1' to the same bits.

- Interrupt Enable Register (IER)

Software enables a specific event by writing the corresponding bit to '1' in this register.
Writing '0' has no effect to this register.

- Interrupt Disable Register (IDR)

Software disables a specific event by writing the corresponding bit to '1' in this register.
Writing '0' has no effect to this register.

When an interrupt event is detected by the core, it sets a specific bit to '1' in the Interrupt Status Register (ISR) to reflect the event. The interrupt signal will be asserted if the corresponding enable bit in the Interrupt Enable Register (IER) is also '1'. If software IRQ Handler detects the interrupt, it should read the ISR to know which event is triggered, and service it. After the interrupt event service is done, software should clear the event by writing '1' to the corresponding bit in ISR. Software can disable some events from asserting interrupt by setting the corresponding bits in the IDR. Those events will still be reflected in the ISR, but never causing an interrupt unless they are enabled by software through IER later.

13.2.3.2 USB Event Interrupt

Connect Event & Disconnect Event

The core monitors the USB VBUS status to detect a connection to a USB Host/Hub port. Once the VBUS status transits from low to high (VBUS is powered), the Connect Event bit in the ISR will be set. On the other hand, the transition of the VBUS status from high to low (VBUS is loss) means a disconnection, and causes the core to set the Disconnect Event bit in the ISR.

Reset Event

The core detects and processes the USB reset (High-speed detection). After the whole USB reset process is completed, the core decides its operation speed (High-speed or Full-speed) based on the High-speed detection result, and sets the Reset Event bit in the ISR.

Suspend Event

The core detects USB bus idle and processes the entry to suspend mode. Before the device goes into suspend mode, it sets Suspend Event bit in the ISR to inform software. Software should response to this by clearing the Suspend Event bit (write a '1'). The core regards this response as an acknowledgement to enter suspend mode. Once in suspend mode, the core de-asserts suspend signal (active low), and the PHY will stop its output clock. This handshake mechanism gives software enough time to finish its jobs safely before it authorizes the core to enter suspend mode and stop the clock.

Resume Event

The USB bus is at Full-speed idle state (J state) while the link is suspended. The core monitors USB bus status to detect USB resume or reset. The transition from J state to K state means a resume signal. The transition from J state to SE0 state means a reset signal. If a resume signal is detected during suspend, the core exits suspend mode and completes the resume process. At the same time, it sets the Resume Event bit in the ISR to inform software. The whole resume process (~20ms) gives both software and hardware enough time to wakeup.
If a reset signal is detected during suspend, the core processes High-speed detection procedure. The core will set the Resume Event bit first, and then set the Reset Event bit after the completion of the USB reset process.

SOF Event

A High-speed USB device receives SOF packet every microframe (125us) and a Full-speed USB device receives SOF packet every frame (1ms) in order to synchronize the frame boundary with the Host. If enabled, the core asserts interrupt every time it receives a SOF packet. For applications which do not need to do synchronization with the Host every frame, it is

recommended to disable this event from triggering interrupt.

DMA Done Event

DMA Done Event indicates that a DMA transfer is completed without any error. Software can start another DMA transfer when the previous transfer is done.

DME Error Event

DMA Error Event indicates that a DMA transfer encounters system bus error. How to detect and handle the system bus error is implementation specific.

SyncFrame Standard Command

This event indicates that the core receives a SyncFrame command.

SetFeature(b_hnp_enable) Standard Command

This event indicates that the core receives a SetFeature(b_hnp_enable) command. This command is used by OTG devices. For non-OTG devices, it is recommended to disable this event.

SetFeature(a_hnp_support) Standard Command

This event indicates that the core receives a SetFeature(a_hnp_support) command. This command is used by OTG devices. For non-OTG devices, it is recommended to disable this event.

SetFeature(a_alt_hnp_support) Standard Command

This event indicates that the core receives a SetFeature(a_alt_hnp_support) command. This command is used by OTG devices. For non-OTG devices, it is recommended to disable this event.

SetInterface Standard Command

This event indicates that the core receives a SetInterface command. Some applications require this event to inform software that the device interface alternate setting has been changed by the Host. Upon receiving this interrupt event, software can check the Current Configuration Register (CCR) to know the current settings.

Control Endpoint 0 SETUP Transaction Event

This event indicates that the core receives a USB command SETUP packet which requires software's cooperation to process it. For most USB standard commands which can be processed by the core, this event will not be set.

Control Endpoint 0 OUT Transaction Event

This event indicates that the core receives a USB command OUT data packet. Software receives this event should enable DMA to transfer data from the Rx Buffer to system memory.

Control Endpoint 0 IN Transaction Event

This event indicates that the core receives a USB command IN packet to ask data. Software receives this event should enable DMA to transfer data from system memory to the Tx Buffer.

Control Endpoint 0 Query Transaction Event

This event indicates that the core receives a USB command STATUS packet. Software receives this event should write the Flush, Handshake and Halt Register (FHHR) Handshake[1:0] to make the core response to the STATUS query. The following table describes the behavior of writing different values into this field. Before software writing this field, the core keeps NAKing the STATUS query.

Handshanke[1:0]	Device response handshake to Host STATUS query
00	No effect(Default)
01	ACK
10	STALL
11	No response

If the Application Control Register (ACR) bit 1 QueryACK is set to '1', the core ACKs all STATUS query automatically. In

this case, this event will not be triggered.

NORMAL ENDPOINT EVENT INTERRUPT

There are six Physical Endpoint Transaction Events for normal operation endpoints from physical endpoint 1 to physical endpoint 6. This event has different meanings for different endpoint types.

For OUT endpoints, this event is set when the core receives an OUT data packet. Software receiving this interrupt should do DMA to move data from the Rx Buffer to system memory.

For Post-buffering IN endpoints, this event is set when the core receives an IN packet. Software receiving this interrupt should do DMA to move data from system memory to the Tx Buffer. For Pre-buffering IN endpoints and Isochronous endpoints, since the data should be prepared in the Tx Buffer in advance before receiving an IN packet, this event is set after the IN data transaction is done on USB bus. Software receiving this interrupt knows that one data in the Tx Buffer has been transferred.

13.3 USB OTG Register Description

13.3.1 USB OTG Register Memory Map

Address	Symbol	Direction	Description
0x20C9_0000	UUSBCMD	R/W	USB UHC Command
0x20C9_0004	UUSBSTS	R/W	USB UHC Status
0x20C9_0008	UUSBINTR	R/W	USB UHC Interrupt Enable
0x20C9_000C	UFRBASEADD	R/(C)	USB UHC Frame List Based Address
0x20C9_0080	UFRNUM	R/W	USB UHC Frame Number
0x20C9_0084	USOFMOD	R/W	USB UHC Start of Frame Modify
0x20C9_0088	UPORTSC	R/W	USB UHC Port Status and Control
0x20C9_4000	ACR	R/W	Application Control Register
0x20C9_4004	MDAR	R/W	Memory Destination Address Register
0x20C9_4008	UDCR	R/W	USB and Device Control Register
0x20C9_400C	FNCR	R/W	Frame Number Control Register
0x20C9_4010	FHHR	R/W	Flush, Handshake and Halt bit Register
0x20C9_4014	PRIR	R/W	Post Request Information Register
0x20C9_4018	STR0	R/W	Setup Transaction0 Register
0x20C9_401C	STR1	R/W	Setup Transaction1 Register
0x20C9_4020	BFCR	R/W	Buffer Flush Controller Register
0x20C9_4030	TBCR0	R/W	TxBuffer0 Control Register
0x20C9_4034	TBCR1	R/W	TxBuffer1 Control Register
0x20C9_4038	TBCR2	R/W	TxBuffer2 Control Register
0x20C9_403C	TBCR3	R/W	TxBuffer3 Control Register
0x20C9_4050	IER	R/W	Interrupt Enable Register
0x20C9_4054	IDR	R/W	Interrupt Disable Register
0x20C9_4058	ISR	R/W	Interrupt Status Register
0x20C9_4070	CCR	R/W	Current Configuration Register
0x20C9_4074	PIR0	R/W	Physical Interface Register0
0x20C9_4078	PIR1	R/W	Physical Interface Register1
0x20C9_4080	EDR0	R/W	Endpoint Descriptor Register
0x20C9_4084	EDR1	R/W	Endpoint Descriptor Register
0x20C9_4088	EDR2	R/W	Endpoint Descriptor Register
0x20C9_408C	EDR3	R/W	Endpoint Descriptor Register
0x20C9_4090	EDR4	R/W	Endpoint Descriptor Register
0x20C9_4094	EDR5	R/W	Endpoint Descriptor Register

0x20C9_4098	EDR6	R/W	Endpoint Descriptor Register
0x20C9_8000	BCWR1	R/W	OTG Link Write Register
0x20C9_8010	BCSR1	R/W	OTG Link Status Register
0x20C9_8020	ISR1	R/W	Interrupt Status Register
0x20C9_8030	IER1	R/W	Interrupt Enable Register
0x20C9_8040	IDR1	R/W	Interrupt Disable Register
0x20C9_8050	IPCR	R/W	IP Control Register
0x20C9_C000	EUSBCMD	R/W	USB EHC Command
0x20C9_C004	EUSBSTS	R/W	USB EHC Status
0x20C9_C008	EUSBINTR	R/W	USB EHC Interrupt Enable
0x20C9_C00C	EFRINDEX	R/(C)	USB EHC Frame Index
0x20C9_C014	EPERIODICLISTBASE	R/W	USB EHC Periodic Frame List Base Address
0x20C9_C018	EASYNCLISTADDR	R/W	USB EHC Current Asynchronous List Address
0x20C9_C040	ECONFIGFLAG	R/W	USB EHC Configured Flag
0x20C9_C044	EPORTSC	R/W	USB EHC Port Status and Control

13.3.2 USB OTG Individual Register Description

Register 13-1: USB UHC Command Register（UUSBCMD, offset=0x0000）

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	R/W	Reserved	0x0
[7]	MAXP	R/W	**Max Packet (MAXP)** 1=64 bytes. 0=32 bytes. This bit selects the maximum packet size that can be used for full speed bandwidth reclamation at the end of a frame. This value is used by the Host Controller to determine whether it should initiate another transaction based on the time remaining in the SOF counter. Use of reclamation packets larger than the programmed size will cause a Babble error if executed during the critical window at frame end. The Babble error results in the offending endpoint being stalled. Software is responsible for ensuring that any packet which could be executed under bandwidth reclamation be within this size limit.	0x0
[6]	CF	R/W	**Configure Flag (CF)** HCD software sets this bit as the last action in its process of configuring the Host Controller. This bit has no effect on the hardware. It is provided only as a semaphore service for software.	0x0
[5]	SWDBG	R/W	**Software Debug (SWDBG)** 1=Debug mode. 0=Normal Mode. In SW Debug mode, the Host Controller clears the Run/Stop bit after the completion of each USB transaction. The next transaction is executed when software sets the Run/Stop bit back to 1. The SWDBG bit must only be manipulated when the controller is in the stopped state. This can be determined by checking the HCHalted bit in the USBSTS register..	0x0
[4]	FGR	R/W	**Force Global Resume (FGR)** 1=Host Controller sends the Global Resume signal on the USB. Software sets this bit to 0 after 20 ms has elapsed to stop sending the Global Resume signal. At that time all USB devices should be ready	0x0

			for bus activity. The Host Controller sets this bit to 1 when a resume event (connect, disconnect, or K-state) is detected while in global suspend mode. Software resets this bit to 0 to end Global Resume signaling. The 1 to 0 transition causes the port to send a low speed EOP signal. This bit will remain a 1 until the EOP has completed.	
[3]	EGSM	R/W	**Enter Global Suspend Mode (EGSM)** 1=Host Controller enters the Global Suspend mode. No USB transaction occurs during this time. The Host Controller is able to receive resume signals from USB and interrupt the system. Software resets this bit to 0 to come out of Global Suspend mode. Software writes this bit to 0 at the same time that Force Global Resume (bit 4) is written to 0 or after writing bit 4 to 0. Software must also ensure that the Run/Stop bit (bit 0) is cleared prior to setting this bit.	0x0
[2]	GRESET	R/W	**Global Reset (GRESET)** When this bit is set, the Host Controller sends the global reset signal on the USB and then resets all its logic, including the internal hub registers. The hub registers are reset to their power on state. This bit is reset by the software after a minimum of 10 ms has elapsed as specified in Chapter 7 of the USB Specification. Note: Chip Hardware Reset has the same effect as Global Reset (bit 2), except that the Host Controller does not send the Global Reset on USB.	0x0
[1]	HCRESET	R/W	**Host Controller Reset (HCRESET)** When this bit is set, the Host Controller module resets its internal timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. This bit is reset by the Host Controller when the reset process is complete. The HCReset effects on Hub registers are slightly different from Chip Hardware Reset and Global USB Reset. The HCReset affects bits [8,3:0] of the Port Status and Control Register (PORTSC) of each port. HCReset resets the state machines of the Host Controller including the Connect/Disconnect state machine (one for each port). When the Connect/Disconnect state machine is reset, the output that signals connect/disconnect are negated to 0, effectively signaling a disconnect even if a device is attached to the port. This virtual disconnect causes the port to be disabled. This disconnect and disabling of the port causes bit 1 (connect status change) and bit 3 (port enable/disable change) of the PORTSC to get set. The disconnect also causes bit 8 of PORTSC to reset. About 64 bit times after HCReset goes to 0, the connect and low-speed detect will take place and bits 0 and 8 of the PORTSC will change accordingly.	0x0
[0]	RS	R/W	**Run/Stop (RS)** 1=Run. 0=Stop. When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Host Controller has finished the transaction and has entered the stopped state. The Host Controller clears this bit when the following fatal errors occur: consistency check failure, AHB Bus errors.	0x0

Register 13-2: USB UHC Status（UUSBSTS, offset=0x0004）

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	RO	Reserved	0x0
[5]	HCHalted	R/WC	**HCHalted bit** The Host Controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (debug mode or an internal error).	0x1
[4]	HCPErr	R/WC	**Host Controller Process Error** The Host Controller sets this bit to 1 when it detects a fatal error and indicates that the Host Controller suffered a consistency check failure while processing a Transfer Descriptor. An example of a consistency check failure would be finding an illegal PID field while processing the packet header portion of the TD. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further schedule execution. A hardware interrupt is generated to the system.	0x0
[3]	HSErr	R/WC	**Host System Error** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. In an AHB system, conditions that set this bit to 1 include ERROR response and 4G boundary detection. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system.	0x0
[2]	RD	R/WC	**Resume Detect** The Host Controller sets this bit to 1 when it receives a “RESUME” signal from a USB device. This is only valid if the Host Controller is in a global suspend state (bit 3 of Command register = 1).	0x0
[1]	USBErr	R/WC	**USB Error Interrupt** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set.	0x0
[0]	USBInt	R/WC	**USB Interrupt (USBINT)** The Host Controller sets this bit to 1 when the cause of an interrupt is a completion of a USB transaction whose Transfer Descriptor had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual length field in TD is less than maximum length field in TD), and short packet detection is enabled in that TD.	0x0

Register 13-3 USB UHC Interrupt Enable（UUSBINTR, offset=0x0008）

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	RO	Reserved	0x0
[3]	SPIE	R/W	**Short Packet Interrupt Enable** 1=Enabled. 0=Disabled.	0x0
[2]	IOCE	R/W	**Interrupt On Complete (IOC) Enable** 1= Enabled. 0=Disabled.	0x0
[1]	RIE	R/W	**Resume Interrupt Enable** 1= Enabled. 0=Disabled.	0x0
[0]	CRCE	R/W	**Timeout/CRC Interrupt Enable**	0x0

			1= Enabled. 0=Disabled.	

Register 13-4: USB UHC Frame List Based Address（UFRBASEADD, offset=0x000C）

Field	Symbol	Direction	Description	Default
[31:12]	FLBASADD	R/W	**Base Address** These bits correspond to memory address signals [31:12], respectively	0x0
[11:0]	Reserved	R/W	Reserved	0x0

Register 13-5: USB UHC Frame Number（UFRNUM, offset=0x0080）

Field	Symbol	Direction	Description	Default
[31]	FrameTestEnable	R/W	**Frame Test Enable** When HIGH represents the SOF counter is running in test mode. When LOW represents the SOF count is running normally. When write '1' to this bit, the SOF counter is reset to 14'h0030 immediately. And reload value of 14'h0300 at each following frame boundary. In normal operation mode, the reload value will be 14'h2ee0.	0x0
[31:11]	Reserved	RO	Reserved	0x0
[10:0]	FrameNum	R/W	**Frame List Current Index/Frame Number** Bits [10:0] provide the frame number in the SOF Frame. The value in this register increments at the end of each time frame (approximately every 1 ms). In addition, bits [9:0] are used for the Frame List current index and correspond to memory address signals [11:2].	0x0

Register 13-6:USB UHC Start of Frame Modify Register（USOFMOD, offset=0x0084）

Field	Symbol	Direction	Description	Default
[31:7]	Reserved	R/W	Reserved	0x0
[6:0]	SOFMOD	R/W	**SOF Timing Value** Guidelines for the modification of frame time are contained in Chapter 7 of the USB Specification. The SOF cycle time (number of SOF counter clock periods to generate a SOF frame length) is equal to 11936 + value in this field. The default value is decimal 64 which gives a SOF cycle time of 12000. For a 12 MHz SOF counter clock input, this produces a 1 ms Frame period. The following table indicates what SOF Timing Value to program into this field for a certain frame period. **Frame Length (# 12Mhz Clocks) (decimal)** — **SOF Reg. Value (decimal)** 11936 — 0 11937 — 1 . . 11999 — 63 12000 — 64 12001 — 65 . . 12062 — 126 12063 — 127	0x40

Register 13-7:USB UHC Port Status/Control Register（UPORTSC, offset=0x0088）

Field	Symbol	Direction	Description	Default
[31:13]	Reserved	RO	Reserved	0x0
[12]	PortSuspend	R/W	**Suspend** 1=Port in suspend state. 0=Port not in suspend state. This bit should not be written to a 1 if global suspend is active (bit 3=1 in the USBCMD register). Bit 2 and bit 12 of this register define the hub states as follows: **Bits [12,2] Hub State** x0 Disable 01 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for single ended 0 resets (global reset and port reset). The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.	0x0
[11:10]	Reserved	RO	Reserved	0x0
[9]	PortReset	R/W	**Port Reset** 1=Port is in Reset. 0=Port is not in Reset. When in the Reset State, the port is disabled and sends the USB Reset signaling. Note that host software must guarantee that the RESET signaling is active for the proper amount of time as specified in the USB Specification.	0x0
[8]	LSDA	RO	**Low Speed Device Attached** 1=Low speed device is attached to this port. 0=Full speed device. Writes have no effect.	0x0
[7]	Reserved	RO	Reserved	0x0
[6]	PortResume	R/W	**Resume Detect** 1= Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. Note that when this bit is 1, a K-state is driven on the port as long as this bit remains 1 and the port is still in suspend state. Writing a 0 (from 1) causes the port to send a low speed EOP. This bit will remain a 1 until the EOP has completed.	0x0
[5:4]	Linestate[1:0]	RO	**Line Status** These bits reflect the D+ (bit 4) and D- (bit 5) signals lines' logical levels. These bits are used for fault detect and recovery as well as for USB diagnostics. This field is updated at EOF2 time (See Chapter 11 of the USB Specification).	0x0
[3]	PortEnableChange	R/WC	**Port Enable/Disable Change** 1=Port enabled/disabled status has changed. 0=No change. For the root hub, this bit gets set only when a port is disabled due to disconnect on the that port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). Software clears this bit by writing a 1 to it.	0x0
[2]	PortEnable	R/W	**Port Enabled/Disabled** 1=Enable. 0=Disable. Ports can be enabled by host software only. Ports can be disabled by either a fault condition (disconnect event or	0x0

			other fault condition) or by host software. Note that the bit status does not change until the port state actually changes and that there may be a delay in disabling or enabling a port if there is a transaction currently in progress on the USB.	
[1]	PortConnectChange	R/WC	**Connect Status Change** 1=Change in Current Connect Status. 0=No change. Indicates a change has occurred in the port's Current Connect Status (see bit 0). The hub device sets this bit for any changes to the port device connect status, even if system software has not cleared a connect status change. If, for example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be setting" an already-set bit (i.e., the bit will remain set). However, the hub transfers the change bit only once when the Host Controller requests a data transfer to the Status Change endpoint. System software is responsible for determining state change history in such a case. Software sets this bit to 0 by writing a 1 to it.	0x0
[0]	PortConnect	RO	**Current Connect Status** 1=Device is present on port. 0=No device is present. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.	0x0

Register 13-8: Application Control Register（ACR, offset=0x4000）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	RO	Reserved	0x0
[26:16]	ReqLength[10:0]	R/W	**Master Request Length of IN** The DMA transfer data length, align with byte. This field is only used by IN transaction (Tx). The DMA length of OUT transaction (Rx) is decided by the core according to the actual byte count it received from the Host.	0x0
[15:12]	Reserved	RO	Reserved	0x0
[11]	ReqError	R/WC	**Master Request Error** Indicate error response arrived during master transfer. Write '0' is no effect. Write '1' to clear this bit. 0: No Error (default) 1: Error response	0x0
[10:8]	TxBufSel[2:0]	R/W	**TxBuffer Selection** For IN transfer, indicate the DMA Tx sub-buffer number. 000b: TxBuffer0 (default) 001b: TxBuffer1 010b: TxBuffer2 011b: TxBuffer3 others: Reserved	0x0
[7:6]	DataToken[1:0]	R/W	**Data Token of High Bandwidth Isochronous IN Transaction** The data token of high bandwidth isochronous IN transaction. Software needs to write a valid data token in this field before it starts to DMA data into TxBuffer. 00: DATA0 (default) 01: DATA1 10: DATA2 11: MDATA	0x0
[5:4]	DMAEnable[1:0]	R/W	**Master Request Enable** Enable a DMA transfer, and indicate transfer direction. This field	0x0

			also shows the status of Master Transfer. Once DMA transfer completes, this field will return to '00b' automatically. 00b No Transfer (DMA idle) 01b Transfer from Rx Buffer to Memory (DMA busy) 10b Transfer from Memory to Tx Buffer (DMA busy) 11b Reserved	
[3]	Reserved	RO	Reserved	0x0
[2]	SuspEnSW	R/W	**PHY suspend by software control** Normal operation device should not use this bit to enter suspend mode.This bit provides a way to force PHY go into suspend mode for special application. 1: Set PHY suspend 0: Disable PHY suspend	0x0
[1]	QueryACK	R/W	**QUERY Transaction AutoACK** Make the core to response Control STATUS query with ACK automatically. 0: disable (default) 1: enabled	0x0
[0]	INPRESEL	R/W	**IN Data Pre-buffering/Post-buffering Selection** Select a Tx data buffering mechanism. This selection will apply to all Interrupt IN and Bulk IN endpoints. Control endpoint is always a Post-buffering. Isochronous IN endpoint is always a Pre-buffering. 0: Post-buffering mechanism (default) 1: Pre-buffering mechanism	0x0

Register 13-9: emory Destination Address Register（MDAR, offset=0x4004）

Field	Symbol	Direction	Description	Default
[31:0]	MemAddr	R/W	**Memory Destination Address** 32 bits width memory address for DMA transfer memory start address.	0x0

Register 13-10: USB and Device Control Register（UDCR, offset=0x4008）

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	RO	Reserved	0x0
[28]	UTMS	R/WC	**USB Test Mode Status** Hardware set UTMS at each SetFeature() with TEST_MODE feature. Write '1' to clear test mode. Write '0' has no effect. 0: disabled (default) 1: enabled	0x0
[27]	UTME	R/W	**USB Test Mode Enable** Write '1' to enable USB test mode. Write '0' has no effect. Read this bit always get a '0'.	0x0
[26:24]	USBTestMode[2:0]	RO	**USB Test Mode** 00h Reserved 01h Test_J 02h Test_K 03h Test_SE0_NAK 04h Test_Packet 05-07h Reserved	0x0
[23:19]	Reserved	RO	Reserved	0x0
[18]	DSI	RO	**Device Speed Indication** Indicate the device operation speed.	0x0

			0b Class Speed 1b High Speed	
[17]	UTD	R/W	**USB Timeout Disable** USB transaction IPG and EED timeout disable bit. Write '1' to disable IPG and EED time interval checking by the core. 0b Enable (default) 1b Disable	0x0
[16]	SyncFrame Enable	R/W	**Enable of SyncFrame Standard Command** If set, the core responses SyncFrame command with the frame number in FNCR. Otherwise, it responses with STALL.	0x0
[25:11]	Reserved	RO	Reserved	0x0
[10]	HNPS	R/W	**Host Negotiation Protocol (HNP) Support** Indicate the application support HNP. This field is for OTG device using. 0: disabled (default) 1: enabled	0x0
[9]	RWS	R/W	**Remote Wakeup Support** Indicate the application support remote wakeup. 0: disabled (default) 1: enabled	0x0
[8]	SPI	R/W	**Self Powered Indication** Indicate the device is bus-powered or self-powered. 0: bus powered (default) 1: self powered	0x0
[7]	Reserved	RO	Reserved	0x0
[6]	SoftDisc	R/W	**Software Disconnect** One-shot-pulse signal. Write '1' to do soft disconnection. The *op_mode[1:0]* of UTMI will be changed to "Non-Driving" mode (2'b01) to generate a disconnect event. Write '0' has no effect. Normal operation device should not use this field.	0x0
[5]	SoftConn	R/W	**Software Connect** One-shot-pulse signal. User writes '1' to do soft connection. The *op_mode[1:0]* will be changed to "Disable" mode (2'b10) to generate a connect event. Write '0' has no effect. Normal operation device should not use this field.	0x0
[4]	LTE	R/W	**LCB Short Period Enable for Simulation** Simulation use only. Most USB events, e.g., Reset, Suspend, and Resume, takes tens of milliseconds to complete. This makes the simulation time too long and waveform size too big. Setting this bit shortens USB events timing. Normal operation device should not use this field. 0: Disabled (default) 1: Enabled	0x0
[3:1]	MTMS	R/W	**Test Mode Select** Simulation use only. When enabled, the core enters selected state directly without standard USB enumeration procedure. Normal operation device should not use this field. 000: Disable test mode (default) 001: Enter Default State (High-Speed) 010: Enter Address State (High-Speed) 011: Enter Configuration State (High-Speed) 100: Reserved 101: Enter Default State (Full-Speed) 110: Enter Address State (Full-Speed) 111: Enter Configuration State (Full-Speed)	0x0

[0]	MCE	R/W	**Core Operation Enable** Setting this bit to enable the core operation. Software should enable the core after the initialization and configuration is done to make the core operating properly. 0: disable (default) 1: enable	0x0

Register 13-11: Frame Number Control Register（FNCR, offset=0x400C）

Field	Symbol	Direction	Description	Default
[31]	Reserved	RO	Reserved	0x0
[30]	hs_reset	W	**Internal Use Only: Software set High-speed Reset OK** Write '1' causes the core enter High-speed mode. Read this bit will always get "0". This bit is for internal use only. Normal operation device should not change this field.	0x0
[29]	fs_reset	W	**Internal Use Only: Software set Full-speed Reset OK** Write '1' causes the core enter Full-speed mode. Read this bit will always get "0". This bit is for internal use only. Normal operation device should not change this field.	0x0
[28]	fs_only	R/W	**Internal Use Only: Full-speed Only Device** Write '1' causes the core regards itself as a Full-speed device while processing High-speed detection. This bit is for internal use only. Normal operation device should not change this field.	0x0
[27]	susp_chk	R/W	**Internal Use Only: USB Suspend Enable/Disable** Write "0" disables USB Suspend mode. In this case, the core will never enter USB Suspend mode. This bit is for internal use only. Normal operation device should not change this field.	0x1
[26:16]	SyncFrameNumber	R/W	**Sync Frame Number** This field is used to respond the SyncFrame() standard command. Software can update this filed if necessary.	0x0
[15:11]	Reserved	RO	Reserved	0x0
[10:0]	FrameNumber[10:0]	R/W	**Current Frame Number** This field is updated at each frame boundary based on the SOF packet.	0x0

Register 13-12: Flush Handshake and Halt bit Register（FHHR, offset=0x4010）

Field	Symbol	Direction	Description	Default
[31:23]	Reserved	RO	Reserved	0x0
[22]	Endpoint6 Halt	R/WS	**Physical Endpoint6 Halt bit** The HALT bit of physical endpoint6. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by Clear_Feature() standard command.	0x0
[21]	Endpoint5 Halt	R/WS	**Physical Endpoint5 Halt bit** The HALT bit of physical endpoint5. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by Clear_Feature() standard command.	0x0
[20]	Endpoint4 Halt	R/WS	**Physical Endpoint4 Halt bit** The HALT bit of physical endpoint4. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by Clear_Feature() standard command.	0x0
[19]	Endpoint3 Halt	R/WS	**Physical Endpoint3 Halt bit** The HALT bit of physical endpoint3. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by	0x0

			Clear_Feature() standard command.	
[18]	Endpoint2 Halt	R/WS	**Physical Endpoint2 Halt bit** The HALT bit of physical endpoint2. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by Clear_Feature() standard command.	0x0
[17]	Endpoint1 Halt	R/WS	**Physical Endpoint1 Halt bit** The HALT bit of physical endpoint1. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by Clear_Feature() standard command.	0x0
[16]	Endpoint0 Halt	R/WS	**Physical Endpoint0 Halt bit** The HALT bit of physical endpoint0. Write '1' to set this bit, and write '0' has no effect. The Host Controller clears this bit by Clear_Feature() standard command.	0x0
[15:2]	Reserved	RO	Reserved	0x0
[1:0]	Handshake[1:0]	R/W	**Handshake Register Bits** Indicate the handshake that core responds to the Control transfer STATUS query. 00: No Effect (default) 01: ACK 10: STALL 11: No Response Note: The DC responds NAK with the STATUS query before this field is written.	0x0

Register 13-13: Post Request Information Register（PRIR, offset=0x4014）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	RO	Reserved	0x0
[26]	HB_ISO	RO	**High Bandwidth Isochronous OUT Transaction** Indicate a receiving of a High Bandwidth Isochronous OUT transaction request.	0x0
[25:24]	HB_ISO_Tkn[1:0]	RO	**Data Token of High Bandwidth Isochronous OUT Transaction** Indicate the data token of the receiving High Bandwidth Isochronous OUT transaction. 00: DATA0 (default) 01: DATA1 10: DATA2 11: MDATA	0x0
[23:16]	PEP[7:0]	RO	Physical Endpoint Number Indicate current request endpoint number.	0x0
[15:11]	Reserved	RO	Reserved	0x0
[11:0]	ByteCnt[10:0]	RO	**Request Byte Count** Indicate current request packet size byte count. For OUT transaction, this field shows the actual byte count the DC received from the Host. For IN transaction, this field always shows the endpoint max packet size.	0x0

Register 13-14: Setup Transaction0 Register（STR0, offset=0x4018）

Field	Symbol	Direction	Description	Default
[31:16]	wValue	RO	**wValue** USB command request value	0x0
[15:8]	bRequest	RO	**bRequest**	0x0

			USB command request	
[7:0]	bmRequestType[7:0]	RO	**bmRequestType** USB Command request type	0x0

Register 13-15: Setup Transaction1 Register（STR1, offset=0x401C）

Field	Symbol	Direction	Description	Default
[31:16]	wLength	RO	**wLength** USB command request length	0x0
[15:8]	wIndex	RO	**wIndex** USB command request index	0x0

Register 13-16: Buffer Flush Controller Register（BFCR, offset=0x4020）

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	RO	Reserved	0x0
[7]	TXB3 Flush	R/WC	**Flush TX Buffer3** One-Shot-Pulse. Only Tx Buffer3 will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0
[6]	TXB2 Flush	R/W	**Flush TX Buffer2** One-Shot-Pulse. Only Tx Buffer2 will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0
[5]	TXB1 Flush	R/W	**Flush TX Buffer1** One-Shot-Pulse. Only Tx Buffer1 will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0
[4]	TXB0 Flush	R/W	**Flush TX Buffer0** One-Shot-Pulse. Only Tx Buffer0 will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0
[3]	Reserved	RO	Reserved	0x0
[2]	TXB Flush	R/W	**Flush TX Buffer** One-Shot-Pulse. Only Tx Buffer will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0
[1]	RXB Flush	R/W	**Flush RX Buffer** One-Shot-Pulse. Only Rx Buffer will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0
[0]	Flush	R/W	**Flush All Data Buffer** One-Shot-Pulse. All of data buffers, includes Rx Buffer and TX Buffer, will be flushed. The flush set will reset all of read and write pointers to default value. And the post request information table is cleared too. Make sure the DMA is idle while setting this bit, or the behavior is Reserved.	0x0

Register 13-17: TxBuffer0 Control Register（TBCR0, offset=0x4030）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	RO	Reserved	0x0
[26:16]	BufferResidue[10:0]	RO	**Post-IN Data Buffer Residue Count** TxBuffer residue buffer size. It updates on DMA transfer done and USB IN packet transfer ok. This field is only valid for POST-IN mechanism.	0x0
[15]	Reserved	RO	Reserved	0x0
[14:12]	PreINQueue[2:0]	RO	**Pre-IN Queue Count** For TxBuffer1/2/3, this field indicates the current Pre-IN Queue Count. For TxBuffer0, this field always shows 3'b000 due to it is used for control endpoint, does not support pre-in mechanism, The '3'b000' for TxBuffer1/2/3 means queue is empty, software could sends IN data to TxBuffer anytime if the pre-in mechanism enables. Once the queue is full, default is '3'b100', software should not send any IN data to current TxBuffer. Otherwise, the TxBuffer overrun would occur.	0x0
[11:9]	Reserved	RO	Reserved	0x0
[8:0]	BufferOffset0[8:0]	R/W	**Data Buffer0 Offset** Word (4-Byte) base. Indicate the Tx sub-buffer start address.	0x0

Register 13-18: TxBuffer1 Control Register（TBCR1, offset=0x4034）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	RO	Reserved	0x0
[26:16]	BufferResidue[10:0]	RO	**Post-IN Data Buffer Residue Count** TxBuffer residue buffer size. It updates on DMA transfer done and USB IN packet transfer ok. This field is only valid for POST-IN mechanism.	0x0
[15]	Reserved	RO	Reserved	0x0
[14:12]	PreINQueue[2:0]	RO	**Pre-IN Queue Count** For TxBuffer1/2/3, this field indicates the current Pre-IN Queue Count. For TxBuffer0, this field always shows 3'b000 due to it is used for control endpoint, does not support pre-in mechanism, The '3'b000' for TxBuffer1/2/3 means queue is empty, software could sends IN data to TxBuffer anytime if the pre-in mechanism enables. Once the queue is full, default is '3'b100', software should not send any IN data to current TxBuffer. Otherwise, the TxBuffer overrun would occur.	0x0
[11:9]	Reserved	RO	Reserved	0x0
[8:0]	BufferOffset1[8:0]	R/W	**Data Buffer0 Offset** Word (4-Byte) base. Indicate the Tx sub-buffer start address.	0x0

Register 13-19: TxBuffer2 Control Register（TBCR2, offset=0x4038）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	RO	Reserved	0x0
[26:16]	BufferResidue[10:0]	RO	**Post-IN Data Buffer Residue Count** TxBuffer residue buffer size. It updates on DMA transfer done and USB IN packet transfer ok. This field is only valid for POST-IN mechanism.	0x0
[15]	Reserved	RO	Reserved	0x0
[14:12]	PreINQueue[2:0]	RO	**Pre-IN Queue Count** For TxBuffer1/2/3, this field indicates the current Pre-IN Queue	0x0

			Count. For TxBuffer0, this field always shows 3'b000 due to it is used for control endpoint, does not support pre-in mechanism, The '3'b000' for TxBuffer1/2/3 means queue is empty, software could sends IN data to TxBuffer anytime if the pre-in mechanism enables. Once the queue is full, default is '3'b100', software should not send any IN data to current TxBuffer. Otherwise, the TxBuffer overrun would occur.	
[11:9]	Reserved	RO	Reserved	0x0
[8:0]	BufferOffset2[8:0]	R/W	**Data Buffer0 Offset** Word (4-Byte) base. Indicate the Tx sub-buffer start address.	0x0

Register 13-20: TxBuffer3 Control Register（TBCR3, offset=0x403C）

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	RO	Reserved	0x0
[26:16]	BufferResidue[10:0]	RO	**Post-IN Data Buffer Residue Count** TxBuffer residue buffer size. It updates on DMA transfer done and USB IN packet transfer ok. This field is only valid for POST-IN mechanism.	0x0
[15]	Reserved	RO	Reserved	0x0
[14:12]	PreINQueue[2:0]	RO	**Pre-IN Queue Count** For TxBuffer1/2/3, this field indicates the current Pre-IN Queue Count. For TxBuffer0, this field always shows 3'b000 due to it is used for control endpoint, does not support pre-in mechanism, The '3'b000' for TxBuffer1/2/3 means queue is empty, software could sends IN data to TxBuffer anytime if the pre-in mechanism enables. Once the queue is full, default is '3'b100', software should not send any IN data to current TxBuffer. Otherwise, the TxBuffer overrun would occur.	0x0
[11:9]	Reserved	RO	Reserved	0x0
[8:0]	BufferOffset3[8:0]	R/W	**Data Buffer0 Offset** Word (4-Byte) base. Indicate the Tx sub-buffer start address.	0x0

Register 13-21: Interrupt Enable Register（IER, offset=0x4050）

Field	Symbol	Direction	Description	Default
[31:30]	Reserved	RO	Reserved	0x0
[29]	PEP6 Transaction	R/WS	**PEP6 Transaction Interrupt Enable** Write '1' to enable PEP6 Transaction interrupt. Write '0' has no effect.	0x0
[28]	PEP5 Transaction	R/WS	**PEP5 Transaction Interrupt Enable** Write '1' to enable PEP5 Transaction interrupt. Write '0' has no effect.	0x0
[27]	PEP4 Transaction	R/WS	**PEP4 Transaction Interrupt Enable** Write '1' to enable PEP4 Transaction interrupt. Write '0' has no effect.	0x0
[26]	PEP3 Transaction	R/WS	**PEP3 Transaction Interrupt Enable** Write '1' to enable PEP3 Transaction interrupt. Write '0' has no effect.	0x0
[25]	PEP2 Transaction	R/WS	**PEP2 Transaction Interrupt Enable** Write '1' to enable PEP2 Transaction interrupt. Write '0' has no effect.	0x0
[24]	PEP1 Transaction	R/WS	**PEP1 Transaction Interrupt Enable** Write '1' to enable PEP1 Transaction interrupt. Write '0' has no	0x0

			effect.	
[23:20]	Reserved	RO	Reserved	0x0
[19]	Control Endpoint Query Transaction	R/WS	**Control Endpoint Query Interrupt Enable** Write '1' to enable Control Endpoint Query Transaction interrupt. Write '0' has no effect.	0x0
[18]	Control Endpoint IN Transaction	R/WS	**Control Endpoint IN Interrupt Enable** Write '1' to enable Control Endpoint IN Transaction interrupt. Write '0' has no effect.	0x0
[17]	Control Endpoint OUT Transaction	R/WS	**Control Endpoint OUT Interrupt Enable** Write '1' to enable Control Endpoint OUT Transaction interrupt. Write '0' has no effect.	0x0
[16]	Control Endpoint SETUP Transaction	R/WS	**Control Endpoint SETUP Interrupt Enable** Write '1' to enable Control Endpoint SETUP Transaction interrupt. Write '0' has no effect.	0x0
[15]	Reserved	RO	Reserved	0x0
[14]	Set_Interface	R/WS	**Set_Interface Interrupt Enable** Write '1' to enable Set_Interface interrupt. Write '0' has no effect.	0x0
[13]	Set_Feature (a_alt_hnp_support)	R/WS	**Set_Feature(a_alt_hnp_support) Interrupt Enable** Write '1' to enable Set_Feature(a_alt_hnp_support) interrupt. Write '0' has no effect.	0x0
[12]	Set_Feature (a_hnp_support)	R/WS	**Set_Feature(a_hnp_support) Interrupt Enable** Write '1' to enable Set_Feature(a_hnp_support) interrupt. Write '0' has no effect.	0x0
[11]	Set_Feature (b_hnp_enable)	R/WS	**Set_Feature(b_hnp_enable) Interrupt Enable** Write '1' to enable Set_Feature(b_hnp_enable) interrupt. Write '0' has no effect.	0x0
[10]	Sync_Frame	R/WS	**Sync_Frame Interrupt Enable** Write '1' to enable Sync_Frame interrupt. Write '0' has no effect.	0x0
[9]	DMA Error	R/WS	**DMA Error Done Interrupt Enable** Write '1' to enable DMA Error Done interrupt. Write '0' has no effect.	0x0
[8]	DMA Done	R/WS	**DMA OK Done Interrupt Enable** Write '1' to enable DMA OK Done interrupt. Write '0' has no effect.	0x0
[7:6]	Reserved	RO	Reserved	0x0
[5]	SOF	R/WS	**SOF Interrupt Enable** Write '1' to enable SOF interrupt. Write '0' has no effect.	0x0
[4]	Resume	R/WS	**Resume Interrupt Enable** Write '1' to enable Resume interrupt. Write '0' has no effect.	0x0
[3]	Suspend	R/WS	**Suspend Interrupt Enable** Write '1' to enable Suspend interrupt. Write '0' has no effect.	0x0
[2]	Reset	R/WS	**Reset Interrupt Enable** Write '1' to enable Reset interrupt. Write '0' has no effect.	0x0
[1]	Disconnect	R/WS	**Disconnect Interrupt Enable** Write '1' to enable Disconnect interrupt. Write '0' has no effect.	0x0
[0]	Connect	R/WS	**Connect Interrupt Enable** Write '1' to enable Connect interrupt. Write '0' has no effect.	0x0

Register 13-22: Interrupt Disable Register（IDR, offset=0x4054）

Field	Symbol	Direction	Description	Default
[31:30]	Reserved	RO	Reserved	0x0
[29]	PEP6 Transaction	R/WS	**PEP6 Transaction Interrupt Disable** Write '1' to clear the enable bit of PEP6 Transaction interrupt. Write '0' has no effect.	0x0

[28]	PEP5 Transaction	R/WS	**PEP5 Transaction Interrupt Disable** Write '1' to clear the enable bit of PEP5 Transaction interrupt. Write '0' has no effect.	0x0
[27]	PEP4 Transaction	R/WS	**PEP4 Transaction Interrupt Disable** Write '1' to clear the enable bit of PEP4 Transaction interrupt. Write '0' has no effect.	0x0
[26]	PEP3 Transaction	R/WS	**PEP3 Transaction Interrupt Disable** Write '1' to clear the enable bit of PEP3 Transaction interrupt. Write '0' has no effect.	0x0
[25]	PEP2 Transaction	R/WS	**PEP2 Transaction Interrupt Disable** Write '1' to clear the enable bit of PEP2 Transaction interrupt. Write '0' has no effect.	0x0
[24]	PEP1 Transaction	R/WS	**PEP1 Transaction Interrupt Disable** Write '1' to clear the enable bit of PEP1 Transaction interrupt. Write '0' has no effect.	0x0
[23:20]	Reserved	RO	Reserved	0x0
[19]	Control Endpoint Query Transaction	R/WS	**Control Endpoint Query Interrupt Disable** Write '1' to clear the enable bit of Control Endpoint Query Transaction interrupt. Write '0' has no effect.	0x0
[18]	Control Endpoint IN Transaction	R/WS	**Control Endpoint IN Interrupt Disable** Write '1' to clear the enable bit of Control Endpoint IN Transaction interrupt. Write '0' has no effect.	0x0
[17]	Control Endpoint OUT Transaction	R/WS	**Control Endpoint OUT Interrupt Disable** Write '1' to clear the enable bit of Control Endpoint OUT Transaction interrupt. Write '0' has no effect.	0x0
[16]	Control Endpoint SETUP Transaction	R/WS	**Control Endpoint SETUP Interrupt Disable** Write '1' to clear the enable bit of Control Endpoint SETUP Transaction interrupt. Write '0' has no effect.	0x0
[15]	Reserved	RO	Reserved	0x0
[14]	Set_Interface	R/WS	**Set_Interface Interrupt Disable** Write '1' to clear the enable bit of Set_Interface interrupt. Write '0' has no effect.	0x0
[13]	Set_Feature (a_alt_hnp_support)	R/WS	**Set_Feature(a_alt_hnp_support) Interrupt Disable** Write '1' to clear the enable bit of Set_Feature(a_alt_hnp_support) interrupt. Write '0' has no effect.	0x0
[12]	Set_Feature (a_hnp_support)	R/WS	**Set_Feature(a_hnp_support) Interrupt Disable** Write '1' to clear the enable bit of Set_Feature(a_hnp_support) interrupt. Write '0' has no effect.	0x0
[11]	Set_Feature (b_hnp_enable)	R/WS	**Set_Feature(b_hnp_enable) Interrupt Disable** Write '1' to clear the enable bit of Set_Feature(b_hnp_enable) interrupt. Write '0' has no effect.	0x0
[10]	Sync_Frame	R/WS	**Sync_Frame Interrupt Disable** Write '1' to clear the enable bit of Sync_Frame interrupt. Write '0' has no effect.	0x0
[9]	DMA Error	R/WS	**DMA Error Done Disable** Write '1' to clear the enable bit of DMA Error Done interrupt. Write '0' has no effect.	0x0
[8]	DMA Done	R/WS	**DMA OK Done Disable** Write '1' to clear the enable bit of DMA OK Done interrupt. Write '0' has no effect.	0x0
[7:6]	Reserved	RO	Reserved	0x0
[5]	SOF	R/WS	**SOF Interrupt Disable** Write '1' to clear the enable bit of SOF interrupt. Write '0' has no effect.	0x0

[4]	Resume	R/WS	**Resume Interrupt Disable** Write '1' to clear the enable bit of Resume interrupt. Write '0' has no effect.	0x0
[3]	Suspend	R/WS	**Suspend Interrupt Disable** Write '1' to clear the enable bit of Suspend interrupt. Write '0' has no effect.	0x0
[2]	Reset	R/WS	**Reset Interrupt Disable** Write '1' to clear the enable bit of Reset interrupt. Write '0' has no effect.	0x0
[1]	Disconnect	R/WS	**Disconnect Interrupt Disable** Write '1' to clear the enable bit of Disconnect interrupt. Write '0' has no effect.	0x0
[0]	Connect	R/WS	**Connect Interrupt Disable** Write '1' to clear the enable bit of Connect interrupt. Write '0' has no effect.	0x0

Register 13-23: Interrupt Status Register（ISR, offset=0x4058）

Field	Symbol	Direction	Description	Default
[31:30]	Reserved	RO	Reserved	0x0
[29]	PEP6 Transaction	R/WC	**Physical Endpoint6 Transaction Event** Indicate an endpoint6 transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[28]	PEP5 Transaction	R/WC	**Physical Endpoint5 Transaction Event** Indicate an endpoint5 transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[27]	PEP4 Transaction	R/WC	**Physical Endpoint4 Transaction Event** Indicate an endpoint4 transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[26]	PEP3 Transaction	R/WC	**Physical Endpoint3 Transaction Event** Indicate an endpoint3 transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[25]	PEP2 Transaction	R/WC	**Physical Endpoint2 Transaction Event** Indicate an endpoint2 transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[24]	PEP1 Transaction	R/WC	**Physical Endpoint1 Transaction Event** Indicate an endpoint1 transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[23:20]	Reserved	RO	Reserved	0x0
[19]	Control Endpoint Query Transaction	R/WC	**Control Endpoint0 Query Transaction Event** Indicate a Control endpoint0 STATUS query transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[18]	Control Endpoint IN Transaction	R/WC	**Control Endpoint0 IN Transaction Event** Indicate a Control endpoint0 DATA IN transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[17]	Control Endpoint OUT Transaction	R/WC	**Control Endpoint0 OUT Transaction Event** Indicate a Control endpoint0 DATA OUT transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[16]	Control Endpoint SETUP Transaction	R/WC	**Control Endpoint0 SETUP Transaction Event** Indicate a Control endpoint0 SETUP transaction event. Write '1' to clear this bit. Write '0' has no effect.	0x0
[15]	Reserved	RO	Reserved	0x0
[14]	Set_Interface	R/WC	**Set_Interface Standard Command** Asserted when a Set_Interface command is completed. Write '1' to	0x0

			clear this bit. Write '0' has no effect.	
[13]	Set_Feature (a_alt_hnp_support)	R/WC	**Set_Feature(a_alt_hnp_support) Standard Command** Asserted when a Set_Feature(a_alt_hnp_support) command is completed. Write '1' to clear this bit. Write '0' has no effect.	0x0
[12]	Set_Feature (a_hnp_support)	R/WC	**Set_Feature(a_hnp_support) Standard Command** Asserted when a Set_Feature(a_hnp_support) command is completed. Write '1' to clear this bit. Write '0' has no effect.	0x0
[11]	Set_Feature (b_hnp_enable)	R/WC	**Set_Feature(b_hnp_enable) Standard Command** Asserted when a Set_Feature(b_hnp_enable) command is completed. Write '1' to clear this bit. Write '0' has no effect.	0x0
[10]	Sync_Frame	R/WC	**Sync_Frame Standard Command** Asserted when a Sync_Frame command is completed. Write '1' to clear this bit. Write '0' has no effect.	0x0
[9]	DMA Error	R/WC	**DMA Error Done Event** Asserted when DMA operation is done with error. Write '1' to clear this bit. Write '0' has no effect.	0x0
[8]	DMA Done	R/WC	**DMA OK Done Event** Asserted when DMA operation is done without any error. Write '1' to clear this bit. Write '0' has no effect.	0x0
[7:6]	Reserved	RO	Reserved	0x0
[5]	SOF	R/WC	**SOF Packet Event** Indicate a receiving of a SOF packet. Write '1' to clear this bit. Write '0' has no effect.	0x0
[4]	Resume	R/WC	**USB Resume Event** Indicate a USB resume event is detected during suspend. Write '1' to clear this bit. Write '0' has no effect.	0x0
[3]	Suspend	R/WC	**USB Suspend Event** Indicate a USB suspend event is detected. Write '1' to clear this bit. Write '0' has no effect. The core would not enter suspend mode until software clear this bit as an acknowledgement.	0x0
[2]	Reset	R/WC	**USB Reset Event** Indicate a USB reset event is detected. Write '1' to clear this bit. Write '0' has no effect.	0x0
[1]	Disconnect	R/WC	**Disconnect Event** Indicate a disconnect event is detected. Write '1' to clear this bit. Write '0' has no effect.	0x0
[0]	Connect	R/WC	**Connection Event** Indicate a connect event is detected. Write '1' to clear this bit. Write '0' has no effect.	0x0

Register 13-24: Current Configuration Register（CCR, offset=0x4070）

Field	Symbol	Direction	Description	Default
[31:12]	Reserved	RO	Reserved	0x0
[11:8]	ConfigNumber[3:0]	RO	**Current Active Configuration Number** Indicate the core current operating configuration. This field is updated by the core at each SetConfiguration() command. 0000: Un-configured (default) 0001: The core operates in Configuration 1 0010: The core operates in Configuration 2 1111: Reserved	0x0
[7]	Reserved	RO	Reserved	0x0
[6:0]	DevAddress[6:0]	RO	**Device Address** Indicate the core current operating device address. This field is	0x0

			updated by the core at each SetAddress() command.	

Register 13-25: Physical Interface Register0（PIR0, offset=0x4074）

Field	Symbol	Direction	Description	Default
[31:28]	MaxAltNum1[3:0]	R/W	**Physical Interface1 Maximum Alternate Setting Number** The maximum alternate setting number of which the physical interface supports. The maximum value is '1111'. The default value is '0000'	0x0
[27:24]	ConfigurationNum1[3:0]	R/W	**Physical Interface1 Configuration Number** The configuration number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[23:20]	LogicalInterfaceNum1[3:0]	R/W	**Physical Interface1 Logic Interface Number** The logical interface number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[19:16]	CurrentActiveAlt1[3:0]	R/W	**Physical Interface1 Current Active Alternate Setting Number** The current active alternate setting number of which the physical interface contains. The maximum value is '1111'. The default value is '0000'.	0x0
[15:12]	MaxAltNum[3:0]	R/W	**Physical Interface0 Maximum Alternate Setting Number** The maximum alternate setting number of which the physical interface supports. The maximum value is '1111'. The default value is '0000'	0x0
[11:8]	ConfigurationNum[3:0]	R/W	**Physical Interface0 Configuration Number** The configuration number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[7:4]	LogicalInterfaceNum[3:0]	R/W	**Physical Interface0 Logic Interface Number** The logical interface number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[3:0]	CurrentActiveAlt[3:0]	RO	**Physical Interface0 Current Active Alternate Setting Number** The current active alternate setting number of which the physical interface contains. The maximum value is '1111'. The default value is '0000'.	0x0

Register 13-26: Physical Interface Register1（PIR1, offset=0x4078）

Field	Symbol	Direction	Description	Default
[31:28]	MaxAltNum1[3:0]	R/W	**Physical Interface3 Maximum Alternate Setting Number** The maximum alternate setting number of which the physical interface supports. The maximum value is '1111'. The default value is '0000'	0x0
[27:24]	ConfigurationNum1[3:0]	R/W	**Physical Interface3 Configuration Number** The configuration number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[23:20]	LogicalInterfaceNum1[3:0]	R/W	**Physical Interface3vLogic Interface Number** The logical interface number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[19:16]	CurrentActiveAlt1[3:0]	R/W	**Physical Interface3 Current Active Alternate Setting Number** The current active alternate setting number of which the physical interface contains. The maximum value is '1111'. The default value is '0000'.	0x0
[15:12]	MaxAltNum[3:0]	R/W	**Physical Interface2 Maximum Alternate Setting Number** The maximum alternate setting number of which the physical	0x0

			interface supports. The maximum value is '1111'. The default value is '0000'	
[11:8]	ConfigurationNum[3:0]	R/W	**Physical Interface2 Configuration Number** The configuration number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[7:4]	LogicalInterfaceNum[3:0]	R/W	**Physical Interface2 Logic Interface Number** The logical interface number to which the physical interface belongs. The maximum value is '1111'. The default value is '0000'	0x0
[3:0]	CurrentActiveAlt[3:0]	RO	**Physical Interface2 Current Active Alternate Setting Number** The current active alternate setting number of which the physical interface contains. The maximum value is '1111'. The default value is '0000'.	0x0

Register 13-27: Endpoint Descriptor RegisterX（EDRX, offset=0x4080~0x4098）

Field	Symbol	Direction	Description	Default
[31]	Halt	R/W	**Halt Bit** Indicate current status of the endpoint. This bit is set/clear by hardware only. 0: Normal function 1: Halt	0x0
[30:28]	EP_InBuffer[3:0]	R/W	**IN buffer entry point select** Define the Tx sub-buffer this endpoint uses if it is an IN endpoint. 000b: TBCR0 select 001b: TBCR1 select 010b: TBCR2 select 011b: TBCR3 select others: Reserved	0x0
[27]	Reserved	RO	Reserved	0x0
[26:16]	EP_MaxSize[10:0]	R/W	**Maximum Packet Size** Define maximum packet size for each data transaction of this endpoint. Note: The maximum value defined in USB2.0 is 1024.	0x0
[15:12]	EP_PhyInterface[3:0]	R/W	**Physical Interface Number** Define the Physical Interface number this endpoint belongs to	0x0
[11:8]	EP_Alter[3:0]	R/W	**Alternate Setting Number** Define the alternate setting number this endpoint belongs to.	0x0
[7:4]	EP_No[3:0]	R/W	**Logic Endpoint Number** Define the endpoint number of this endpoint. Two physical endpoints may have the same logic endpoint number if they are in different Configurations. Note: 4'h0 (endpoint 0) is Reserved for Control endpoint.	0x0
[3]	EP_Dir	R/W	**Transfer direction of Endpoint** Define the direction of this endpoint. This bit is invalid for Control endpoints. 0: OUT endpoint 1: IN endpoint	0x0
[2:0]	EP_Type[2:0]	R/W	**Transfer type of Endpoint.** Define the endpoint type. 000: Control 001: Bulk 010: Interrupt 100: Isochronous 101: High Bandwidth ISO with one maximum transaction	0x0

			110: High Bandwidth ISO with two maximum transaction 111: High Bandwidth ISO with three maximum transaction	

Register 13-28: OTG Link Write Register（BCWR1, offset=0x8000）

Field	Symbol	Direction	Description	Default
[31]	PHY_Susp	R/W	**PHY Suspend** Set this bit to "1" will force the PHY to enter suspend mode and stop the UTMI clock. This bit can be used in Host suspend control, or application specific purpose.	0x0
[30]	EHC_En	R/W	**EHCI (USB2.0 Host Controller) Enable** 1'b1: Enable OTG_EHC. The OTG_DRD operates at High-speed Host mode if a High-speed capable peripheral is connected. 1'b0: Disable OTG_EHC. The OTG_DRD operates at Full-speed Host mode if a High-speed capable peripheral is connected..	0x0
[29:28]	Reserved	RO	Reserved	0x0
[27:26]	Vbus_Timer	R/W	**Vbus Glitch-Free Timer** A timer setting to filter VBUS glitch. 00 = 4us 01 = 8us 10 = 16us 11 = 32us	0x0
[25:24]	Dp_Timer	R/WC	**Data-line Glitch-Free Timer** A timer setting to filter DP/DM glitch. 00 = 4us 01 = 8us 10 = 16us 11 = 32us	0x0
[23:21]	Reserved	RO	Reserved	0x0
[20:16]	Test_Mode	R/W	**Simulation Test Mode** Must set to 5'b00000	0x0
[15]	B_HNP_Fail	R/W	**B-Device HNP: HNP Fail** During B-Device HNP process, if driver detects the HNP failed, it sets this bit to force the core to return to the peripheral mode (B_PERI). After the core returning to peripheral mode, driver should clear this bit. 1'b1: Software sets HNP fail during HNP process 1'b0: Clear event	0x0
[14]	B_HNP_OK	R/W	**B-Device HNP: HNP Success** During B-Device HNP process, if driver detects the HNP success, it sets this bit to force the core to enter host mode (B_HOST). After the core transitioning to host mode, driver should clear this bit. 1'b1: Software sets HNP success during HNP process 1'b0: Clear event	0x0
[13]	B_HNP_En	R/W	**B-Device HNP: HNP Enable** Enable B-Device HNP capability. Driver should set this bit according to the host Set_Feature(b_hnp_en) command. 1'b1: B-Device is able to request HNP to A-Device 1'b0: B-Device is not able to request HNP to A-Device	0x0
[12]	B_HNP_Req	R/W	**B-Device HNP: Request HNP** B-Device requests HNP to A-Device. After setting to "1", this bit should be maintained during all HNP process unless the driver is willing to terminate the HNP process. The B-Device will return to peripheral mode if this bit is cleared. 1'b1: B-Device requests HNP 1'b0: Terminate HNP process and B-Device returns to peripheral mode.	0x0
[11]	B_Conn_Evt	R/W	**B-Device: Connect Event** Setting this bit to "1" will cause B-Device enter peripheral mode. Driver sets this bit when it detects connect event by monitoring the	0x0

			VBUS status in BCSR. After B-Device transitioning to peripheral mode, driver should clear this bit. 1'b1: Connect event and B-Device enters peripheral mode 1'b0: Clear event	
[10]	B_Chrg_Vbus	R/W	**B-Device SRP: Charge VBUS** B-Device issues SRP charge VBUS during SRP process. 1'b1: Charging VBUS 1'b0: Stop charging VBUS	0x0
[9]	B_Chrg_Dp	R/W	**B-Device SRP: Charge DP** B-Device issues SRP charge DP during SRP process. 1'b1: Charging DP 1'b0: Stop charging DP	0x0
[8]	B_Dschg_Vbus	R/W	**B-Device SRP: Discharge VBUS** B-Device issues SRP discharge VBUS during SRP process. 1'b1: Discharging VBUS 1'b0: Stop discharging VBUS	0x0
[7]	A_HNP_En	R/W	**A-Device HNP: HNP Enable** Enable A-Device HNP capability. 1'b1: A-Device is able to accept HNP request from B-Device 1'b0: A-Device is not able to accept HNP request from B-Device	0x0
[6:4]	Reserved	RO	Reserved	0x0
[3]	A_Idle	R/W	**Return to IDLE request** Setting this bit to "1" to force A-Device transition to idle state (A_IDLE). Driver should clear this bit after A-Device transitioning to idle state. 1'b1: A-Device returns to idle state 1'b0: Clear event	0x0
[2]	A_Conn_Evt	R/W	**Driver detects A Connect Event** Setting this bit to "1" will cause A-Device enter host mode. Driver sets this bit when it detects connect event by monitoring the linestate status in BCSR. After A-Device transitioning to host mode, driver should clear this bit. 1'b1: Connect event and A-Device enters host mode 1'b0: Clear event	0x0
[1]	A_Vbus_Req	R/W	**Drive Vbus request** Setting this bit to "1" will cause A-Drive to drive VBUS to 5V. Setting this bit to "0" will cause A-Device to stop driving VBUS. This bit should be remained to high during A-Device operation. 1'b1: Driving VBUS 1'b0: Stop driving VBUS	0x0
[0]	ID_Pullup	R/W	**Driver asserts ID Pull-up** Drive IdPullup to get ID Pin value. 1'b1: Driving IdPullup 1'b0: Stop driving IdPullup	0x0

Register 13-29: OTG Link Status Register（BCSR1, offset=0x8010）

Field	Symbol	Direction	Description	Default
[31:17]	Reserved	RO	Reserved	0x0
[16]	A_HNP_Start	RO	**A-Device HNP Start (Read Pulse Clear)** During A-Device suspend state (A_SUSP), the core set this bit if it detects that the B-Device disconnects itself to start HNP process. Software should monitor this bit if HNP is expected. Read this bit will clear this event.	0x0

[15]	Sess_End	RO	**Session End** VBUS status is Session End for an interval (Vbus_Timer in BCWR). 1'b1: VBUS < 0.2V 1'b0: VBUS > 0.8V	0x0
[14]	Vbus_Valid	RO	**VBUS Valid** VBUS status is VBUS Valid for an interval (Vbus_Timer in BCWR). 1'b1: VBUS > 4.75V 1'b0: VBUS < 4.4V	0x0
[13]	B_Valid	RO	**B Session Valid** VBUS status is B Session Valid for an interval (Vbus_Timer in BCWR). 1'b1: VBUS > 4V 1'b0: VBUS < 0.8V	0x0
[12]	A_Valid	RO	**A Session Valid** VBUS status is A Session Valid for an interval (Vbus_Timer in BCWR). 1'b1: VBUS > 2V 1'b0: VBUS < 0.8V	0x0
[11]	Line_SE1	RO	**Linestate SE1** Linestate status is SE1 (DP is high, DM is high) for an interval (DP_Timer in BCWR).	0x0
[10]	Line_SE0	RO	**Linestate SE0** Linestate status is SE0 (DP is low, DM is low) for an interval (DP_Timer in BCWR).	0x0
[9]	Line_K	RO	**Linestate K** Linestate status is K (DP is low, DM is high) for an interval (DP_Timer in BCWR).	0x0
[8]	Line_J	RO	**Linestate J** Linestate status is J (DP is high, DM is low) for an interval (DP_Timer in BCWR).	0x0
[7]	ID_Pin	RO	**Current ID Pin** ID Pin status. This bit is only valid when ID_Pullup in BCWR is set to "1". 1'b1: Connected plug is a mini-B 1'b0: Connected plug is a mini-A	0x0
[6]	Reserved	RO	Reserved	0x0
[5:4]	DRDB_CS	RO	**B-Device FSM Current State** 2'b00: B_IDLE 2'b01: B_PERI 2'b10: B_WACON 2'b11: B_HOST	0x0
[3]	Reserved	RO	Reserved	0x0
[2:0]	DRDA_CS	RO	**A-Device FSM Current State** 3'b000: A_IDLE 3'b001: A_WVRISE 3'b010: A_WBCON 3'b011: A_HOST 3'b100: A_SUSP 3'b101: A_PERI 3'b110: A_WVFALL	0x0

Register 13-30: OTG Link Interrupt Status Register（BCISR, offset=0x8020）

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	RO	Reserved	0x0
[5]	B_HNP_End	R/WC	**B-Device HNP Process End IRQ Event** This bit is set to "1" by the B-Device core when it completes HNP process and enters host mode. Driver clears this event by writing a "1".	0x0
[4]	B_HNP_Start	R/WC	**B-Device Starts HNP Process IRQ Event** This bit is set to "1" by the B-Device core when it starts HNP process. Driver clears this event by writing a "1".	0x0
[3]	B_Disc_Evt	R/WC	**B-Device Disconnect IRQ Event** This bit is set to "1" by the B-Device core when it detects a disconnect event. Driver clears this event by writing a "1".	0x0
[2]	A_HNP_End	R/WC	**A-Device HNP End and Wait B-Device Connect IRQ Event** This bit is set to "1" by the A-Device core when it detects a HNP end due to USB bus idle. Driver clears this event by writing a "1".	0x0
[1]	A_Disc_Evt	R/WC	**A-Device Disconnect IRQ Event** This bit is set to "1" by the A-Device core when it detects a disconnect event.Driver clears this event by writing a "1".	0x0
[0]	A_Low_Pwr	R/WC	**A-Device Low Power IRQ Event** This bit is set to "1" by the A-Device core when it detects a low power condition (cannot drive VBUS to A_Valid). Driver clears this event by writing a "1".	0x0

Register 13-31: OTG Link Interrupt Enable Register（BCIER1, offset=0x8030）

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	RO	Reserved	0x0
[5]	B_HNP_End_En	R/W	**B-Device HNP Process End IRQ Event Enable** Write '1' to enable this event interrupt. Write '0' has no effect.	0x0
[4]	B_HNP_Start_En	R/W	**B-Device starts HNP Process IRQ Event Enable** Write '1' to enable this event interrupt. Write '0' has no effect.	0x0
[3]	B_Disc_Evt_En	R/W	**B-Device Disconnect IRQ Event Enable** Write '1' to enable this event interrupt. Write '0' has no effect.	0x0
[2]	A_HNP_End_En	R/W	**A-Device HNP End and Wait B-Device Connect IRQ Event Enable** Write '1' to enable this event interrupt. Write '0' has no effect.	0x0
[1]	A_Disc_Evt_En	R/W	**A-Device Disconnect IRQ Event Enable** Write '1' to enable this event interrupt. Write '0' has no effect.	0x0
[0]	A_Low_Pwr_En	R/W	**A-Device Low Power IRQ Event Enable** Write '1' to enable this event interrupt. Write '0' has no effect.	0x0

Register 13-32: OTG Link Interrupt Disable Register（BCIDR1, offset=0x8040）

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	RO	Reserved	0x0
[5]	B_HNP_End_Dis	R/W	**B-Device HNP Process End IRQ Event Disable** Write '1' to clear the enable bit of this event in IER. Write '0' has no effect.	0x0
[4]	B_HNP_Start_Dis	R/W	**B-Device starts HNP Process IRQ Event Disable** Write '1' to clear the enable bit of this event in IER. Write '0' has no effect.	0x0

[3]	B_Disc_Evt_Dis	R/W	**B-Device Disconnect IRQ Event Disable** Write '1' to clear the enable bit of this event in IER. Write '0' has no effect.	0x0
[2]	A_HNP_End_Dis	R/W	**A-Device HNP End and Wait B-Device Connect IRQ Event Disable** Write '1' to clear the enable bit of this event in IER. Write '0' has no effect.	0x0
[1]	A_Disc_Evt_Dis	R/W	**A-Device Disconnect IRQ Event Disable** Write '1' to clear the enable bit of this event in IER. Write '0' has no effect.	0x0
[0]	A_Low_Pwr_Dis	R/W	**A-Device Low Power IRQ Event Disable** Write '1' to clear the enable bit of this event in IER. Write '0' has no effect.	0x0

Register 13-33: OTG Link IP Control Register（BCIPCR, offset=0x8050）

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	RO	Reserved	0x0
[5]	A_HNP_Proc	R/W	**A-Device HNP Process Ongoing** Set this bit to "1" to prevent Disconnect Event detection at A-Device HNP process. HNP process contains B-Device disconnection itself to transition from peripheral mode to host mode. A disconnect IRQ during the process will cause HNP failed. Driver should set this bit to ensure the HNP process from disconnect event interfering. 1'b1: A-Device HNP Process is ongoing and disable the disconnect event detection 1'b0: No effect	0x0
[4]	Reserved	RO	Reserved	0x0
[3]	OTG_IRQ_En	R/W	**OTG Link IRQ Enable** 1'b1: Enable 1'b0: Disable	0x1
[2]	EHC_IRQ_En	R/W	**OTG_EHC IRQ Enable** 1'b1: Enable 1'b0: Disable	0x1
[1]	UHC_IRQ_En	R/W	**OTG_UHC IRQ Enable** 1'b1: Enable 1'b0: Disable	0x1
[0]	DC_IRQ_En	R/W	**OTG_DC IRQ Enable** 1'b1: Enable 1'b0: Disable	0x1

Register 13-34: OTG USB EHC Command（EUSBCMD, offset=0xC000）

Field	Symbol	Direction	Description	Default
[31:24]	Reserved	RO	Reserved	0x0
[23:16]	InthCtrl	R/W	**Interrupt Threshold Control** This field is used by system software to select the maximum rate at which the Host Controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are Reserved. **Value Maximum Interrupt Interval** 00h Reserved 01h 1 micro-frame 02h 2 micro-frames 04h 4 micro-frames	0x0

			08h 8 micro-frames (default, equates to 1 ms) 10h 16 micro-frames (2 ms) 20h 32 micro-frames (4 ms) 40h 64 micro-frames (8 ms) Refer to Section 4.15 for interrupts affected by this register. Any other value in this register yields Reserved results. Software modifications to this bit while HCHalted bit is equal to zero results in Reserved behavior.	
[15:12]	Reserved	RO	Reserved	0x0
[11]	ASPME	R/W	**Asynchronous Schedule Park Mode Enable (OPTIONAL)** If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled.	0x0
[10]	Reserved	RO	Reserved	0x0
[9:8]	ASPMC	R/W	**Asynchronous Schedule Park Mode Count (OPTIONAL).** It contains a count of the number of successive transactions the Host Controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. See Section 4.10.3.2 for full operational details. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in Reserved behavior.	0x0
[7]	Reserved	RO	Reserved	0x0
[6]	Doorbell	R/W	**Interrupt on Async Advance Doorbell** This bit is used as a doorbell by software to tell the Host Controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to *ring* the doorbell. When the Host Controller has evicted all appropriate cached schedule state, it sets the *Interrupt on Async Advance* status bit in the USBSTS register. If the *Interrupt on Async Advance Enable* bit in the USBINTR register is a one then the Host Controller will assert an interrupt at the next interrupt threshold. The Host Controller sets this bit to a zero after it has set the *Interrupt on Async Advance* status bit in the USBSTS register to a one. Software should not write a one to this bit when the asynchronous schedule is disabled. Doing so will yield Reserved results.	0x0
[5]	ASEnable	R/W	**Asynchronous Schedule Enable** This bit controls whether the Host Controller skips processing the Asynchronous Schedule. 0b Do not process the Asynchronous Schedule 1b Use the ASYNCLISTADDR register to access the Asynchronous Schedule	0x0
[4]	PSEnable	R/W	**Periodic Schedule Enable** This bit controls whether the Host Controller skips processing the Periodic Schedule. 0b Do not process the Periodic Schedule 1b Use the PERIODICLISTBASE register to access the Periodic Schedule	0x0
[3:2]	Reserved	RO	Reserved	0x0
[1]	HCRESET	R/W	**Host Controller Reset** This control bit is used by software to reset the Host Controller. The effects of this on Root Hub registers are similar to a Chip Hardware Reset.When software writes a one to this bit, the Host Controller	0x0

			resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. All operational registers, including port registers and port state machines are set to their initial values. Software must reinitialize the Host Controller in order to return the Host Controller to an operational state. This bit is set to zero by the Host Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Software should not set this bit to a one when the *HCHalted* bit in the USBSTS register is a zero. Attempting to reset an actively running HostController will result in Reserved behavior.	
[0]	RS	R/W	**Run/Stop** 1=Run. 0=Stop. When set to a 1, the Host Controller proceeds with execution of the schedule. The Host Controller continues execution as long as this bit is set to a 1. When this bit is set to 0, the Host Controller completes the current and any actively pipelined transactions on the USB and then halts. The Host Controller must halt within 16 micro-frames after software clears the Run bit. The HC Halted bit in the status register indicates when the Host Controller has finished its pending pipelined transactions and has entered the stopped state. Software must not write a one to this field unless the Host Controller is in the Halted state (i.e. *HCHalted* in the USBSTS register is a one). Doing so will yield Reserved results.	0x0

Register 13-35: OTG EHC Status Register（EUSBSTS, offset=0xC004）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	RO	Reserved	0x0
[15]	ASStatus	RO	**Asynchronous Schedule Status** The bit reports the current real status of the Asynchronous Schedule. If this bit is a zero then the status of the Asynchronous Schedule is disabled. If this bit is a one then the status of the Asynchronous Schedule is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the *Asynchronous Schedule Enable* bit in the USBCMD register. When this bit and the *Asynchronous Schedule Enable* bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0).	0x0
[14]	PSStatus	RO	**Periodic Schedule Status** The bit reports the current real status of the Periodic Schedule. If this bit is a zero then the status of the Periodic Schedule is disabled. If this bit is a one then the status of the Periodic Schedule is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit in the USBCMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0).	0x0
[13]	Recl	RO	**Reclamation** This is a read-only status bit, which is used to detect an empty asynchronous schedule.	0x0
[12]	HCHalt	RO	**HCHalted**	0x0

			This bit is a zero whenever the Run/Stop bit is a one. The Host Controller sets this bit to one after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g. internal error).	
[11:6]	Reserved	RO	Reserved	0x0
[5]	AsynAdvInt	R/WC	**Interrupt on Asynchronous Advance** System software can force the Host Controller to issue an interrupt the next time the Host Controller advances the asynchronous schedule by writing a one to the *Interrupt on Async Advance Doorbell* bit in the USBCMD register. This status bit indicates the assertion of that interrupt source.	0x0
[4]	SysErrInt	R/WC	**Host System Error** The Host Controller sets this bit to 1 when a serious error occurs during a host system access involving the Host Controller module. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs.	0x0
[3]	FLRInt	R/WC	**Frame List Rollover** The Host Controller sets this bit to a one when the *Frame List Index* rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the *Frame List Size* field of the USBCMD register) is 1024, the *Frame Index Register* rolls over every time FRINDEX[13] toggles.	0x0
[2]	PortChgInt	R/WC	**Port Change Detect** The Host Controller sets this bit to a one when any port for which the *Port Owner* bit is set to zero has a change bit transition from a zero to a one or a *Force Port Resume* bit transition from a zero to a one as a result of a J-K transition detected on a suspended port. This bit will also be set as a result of the *Connect Status Change* being set to a one after system software has relinquished ownership of a connected port by writing a one to a port's *Port Owner* bit.	0x0
[1]	USBERRInt	R/WC	**USB Error Interrupt** The Host Controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and USBINT bit are set.	0x0
[0]	USBInt	R/WC	**USB Interrupt** The Host Controller sets this bit to 1 on the completion of a USB transaction, which results in the retirement of a Transfer Descriptor that had its IOC bit set. The Host Controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes).	0x0

Register 13-36: OTG EHC Interrupt Enable（EUSBIER, offset=0xC008）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	RO	Reserved	0x0
[5]	AsynAdvInt_E	R/W	**Interrupt on Async Advance Enable.** 1= Enabled. 0=Disabled.	0x0
[4]	SysErrInt_E	R/W	**Host System Error Enable.** 1= Enabled. 0=Disabled	0x0
[3]	FLRInt_E	R/W	**Frame List Rollover Enable.** 1= Enabled. 0=Disabled.	0x0

[2]	PortChgInt_E	R/W	**Port Change Interrupt Enable** 1= Enabled. 0=Disabled.	0x0
[1]	USBERRInt_E	R/W	**USB Error Interrupt Enable** 1= Enabled. 0=Disabled.	0x0
[0]	USBInt_E	R/W	**USB Interrupt Enable** 1= Enabled. 0=Disabled.	0x0

Register 13-37: OTG EHC Frame Index（EFRINDEX, offset=0xC00C）

Field	Symbol	Direction	Description	Default
[31:14]	Reserved	RO	Reserved	0x0
[13:0]	Findx	R/W	**Frame Index** The value in this register increments at the end of each time frame (e.g. micro-frame). Bits [N:3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index.	0x0

Register 13-38: OTG EHC Periodic Frame List Base Address（EPERIODICLISTBASE, offset=0xC014）

Field	Symbol	Direction	Description	Default
[31:12]	PLBase	R/W	**Base Address (Low)** These bits correspond to memory address signals [31:12], respectively.	0x0
[11:0]	Reserved	RO	Reserved	0x0

Register 13-39: OTG EHC Current Asynchronous List Address（EASYNCLISTADDR, offset=0xC018）

Field	Symbol	Direction	Description	Default
[31:5]	ALBase	R/W	**Link Pointer Low** These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH).	0x0
[4:0]	Reserved	RO	Reserved	0x0

Register 13-40: OTG EHC Configured Flag（ECONFIGFLAG, offset=0xC040）

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	RO	Reserved	0x0
[0]	CF	R/W	**Configure Flag** Host software sets this bit as the last action in its process of configuring the Host Controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below. 0b Port routing control logic default-routes each port to an implementation dependent classic Host Controller. 1b Port routing control logic default-routes all ports to this Host Controller	0x0

Register 13-41: OTG EHC Port Status and Control（EPORTSC, offset=0xC044）

Field	Symbol	Direction	Description	Default
[31:23]	Reserved	RO	Reserved	0x0
[22]	WKOC_E	R/W	**Wake on Over-current Enable** Writing this bit to a one enables the port to be sensitive to	0x0

			over-current conditions as wake-up events.	
[21]	WKDSCNNT_E	RO	**Wake on Disconnect Enable** Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events.	0x0
[20]	WKCNNT_E	R/W	**Wake on Connect Enable** Writing this bit to a one enables the port to be sensitive to device connects as wake-up events.	0x0
[29:16]	TestCtrl	R/WC	**Port Test Control** When this field is zero, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. The encoding of the test mode bits are (0110b - 1111b are Reserved): **Bits Test Mode** 0000b Test mode not enabled 0001b Test J_STATE 0010b Test K_STATE 0011b Test SE0_NAK 0100b Test Packet 0101b Test FORCE_ENABLE	0x0
[15:14]	Reserved	RO	Reserved	0x0
[13]	PortOwner	R/W	**Port Owner** This bit unconditionally goes to a 0b when the *Configured* bit in the CONFIGFLAG register makes a 0b to 1b transition. This bit unconditionally goes to 1b whenever the *Configured* bit is zero. System software uses this field to release ownership of the port to a selected Host Controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that a companion Host Controller owns and controls the port.	0x1
[12]	Reserved	RO	Reserved	0x1
[11:10]	Line	RO	**Line Status** These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is zero and the current connect status bit is set to a one. The encoding of the bits are: **Bits[11:10] USB State Interpretation** 00b SE0 Not Low-speed device, perform EHCI reset 10b J-state Not Low-speed device, perform EHCI reset 01b K-state Low-speed device, release ownership of port 11b Reserved Not Low-speed device, perform EHCI reset	0x0
[9]	Reserved	RO	Reserved	0x0
[8]	PortReset	R/W	**Port Reset** 1=Port is in Reset. 0=Port is not in Reset. When software writes a one to this bit (from a zero), the bus reset sequence as defined in the USB Specification Revision 2.0 is started. Software writes a zero to this bit to terminate the bus reset sequence. Software must keep this bit at a one long enough to ensure the reset sequence, as specified in the USB Specification Revision 2.0, completes. Note: when software writes this bit to a one, it must also write a zero to the *Port Enable* bit. Note that when software writes a zero to this bit there may be a delay before the bit status changes to a zero. The bit status will not read as a zero until after the reset has completed. If the port is in high-speed mode after reset is complete, the Host Controller will automatically	0x0

			enable this port (e.g. set the *Port Enable* bit to a one). A Host Controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from a one to a zero. For example: if the port detects that the attached device is high-speed during reset, then the Host Controller must have the port in the enabled state within 2ms of software writing this bit to a zero. The *HCHalted* bit in the USBSTS register should be a zero before software attempts to use this bit. The Host Controller may hold Port Reset asserted to a one when the *HCHalted* bit is a one.	
[7]	PortSusp	R/W	**Suspend** 1=Port in suspend state. 0=Port not in suspend state. Port Enabled Bit and Suspend bit of this register define the port states as follows: **Bits [Port Enabled, Suspend] Port State** 0X Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. A write of zero to this bit is ignored by the Host Controller. The Host Controller will unconditionally set this bit to a zero when: Software sets the *Force Port Resume* bit to a zero (from a one). Software sets the *Port Reset* bit to a one (from a zero). If host software sets this bit to a one when the port is not enabled (i.e. Port enabled bit is a zero) the results are Reserved.	0x0
[6]	PortResume	R/W	**Force Port Resume R/W** 1=Resume detected/driven on port. 0=No resume (K-state) detected/driven on port. This functionality defined for manipulating this bit depends on the value of the *Suspend* bit. For example, if the port is not suspended (*Suspend* and *Enabled* bits are a one) and software transitions this bit to a one, then the effects on the bus are Reserved. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to a one. If software sets this bit to a one, the Host Controller must not set the *Port Change Detect* bit. Note that when the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. Software must appropriately time the Resume and set this bit to a zero when the appropriate amount of time has elapsed. Writing a zero (from one) causes the port to return to the high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a one until the port has switched to the high-speed idle. The Host Controller must complete this transition within 2 milliseconds of software setting this bit to a zero.	0x0
[5]	OverCurrentChg	R/WC	**Over-current Change** 1=This bit gets set to a one when there is a change to Over-current	0x0

			Active. Software clears this bit by writing a one to this bit position.	
[4]	OverCurrent	RO	**Over-current Active** 1=This port currently has an over-current condition. 0=This port does not have an over-current condition. This bit will automatically transition from a one to a zero when the over current condition is removed.	0x0
[3]	PortEnChg	R/WC	**Port Enable/Disable Change** 1=Port enabled/disabled status has changed. 0=No change. For the root hub, this bit gets set to a one only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a Port Error). Software clears this bit by writing a 1 to it.	0x0
[2]	PortEn	RO	**Port Enabled/Disabled** 1=Enable. 0=Disable. Ports can only be enabled by the Host Controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. The Host Controller will only set this bit to a one when the reset sequence determines that the attached device is a high-speed device. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other Host Controller and bus events. When the port is disabled (0b) downstream propagation of data is blocked on this port, except for reset.	0x0
[1]	ConnChg	R/WC	**Connect Status Change** 1=Change in Current Connect Status. 0=No change. Indicates a change has occurred in the port's Current Connect Status. The Host Controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). Software sets this bit to 0 by writing a 1 to it.	0x0
[0]	Conn	RO	**Current Connect Status** 1=Device is present on port. 0=No device is present. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.	0x0

14 SD/SDHC/SDIO Controller

14.1 Overview

The SD/SDHC/SDIO host controller (SDHC) is a host for SD memory cards, or SDIO cards and combo cards. This host is compatible for SD Association's (SDA) Host Standard Specification. The performance of this host is very powerful, you would get up to 50MHz clock rate.

The IMAPx210 processor contains three SD/SDHC/SDIO host controllers, each controller can be used as DVB-TS receiver. The DVB-TS interface is multi-used with camera interface.

Features

- SD Standard Host Specification (ver 2.0) compatible
- SD Memory Card Specification (ver 2.0) 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 mode switch support
- Auto CMD12 support
- Suspend / Resume support
- Read Wait operation support
- Card Interrupt support
- DVB-TS receiver

Block Diagram

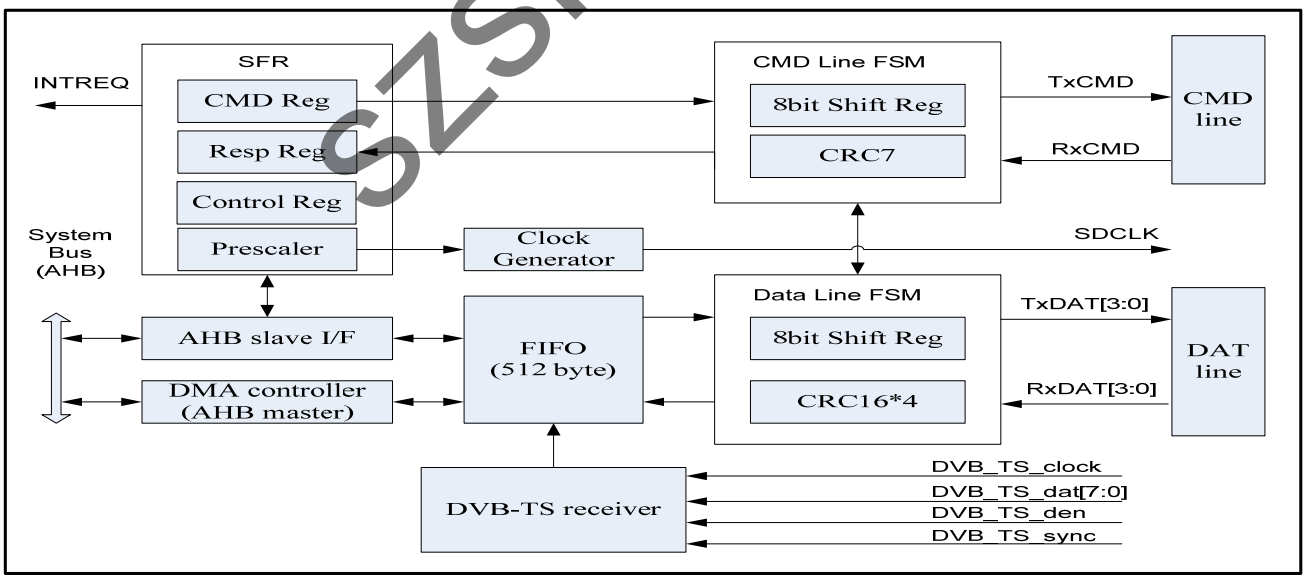


Figure 14-1: SDHC Block Diagram

Figure 14-1 shows the functional block diagram of SDHC, controller deals with SD/SDIO protocol at transmission level, data packing, adding cyclic redundancy checks (CRC), start/end bit, and checking for syntactical correctness. When SDHC used as DVB-TS receiver, only part of module will work, as shown in figure below.

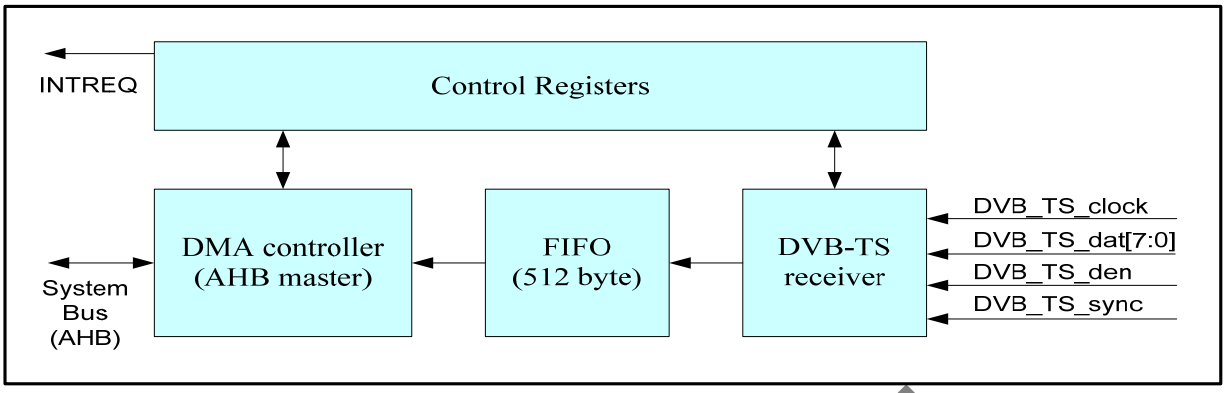


Figure 14-2: DVB-TS Receiver Block Diagram

14.2 Functional Description

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.

SD Card Detection Sequence

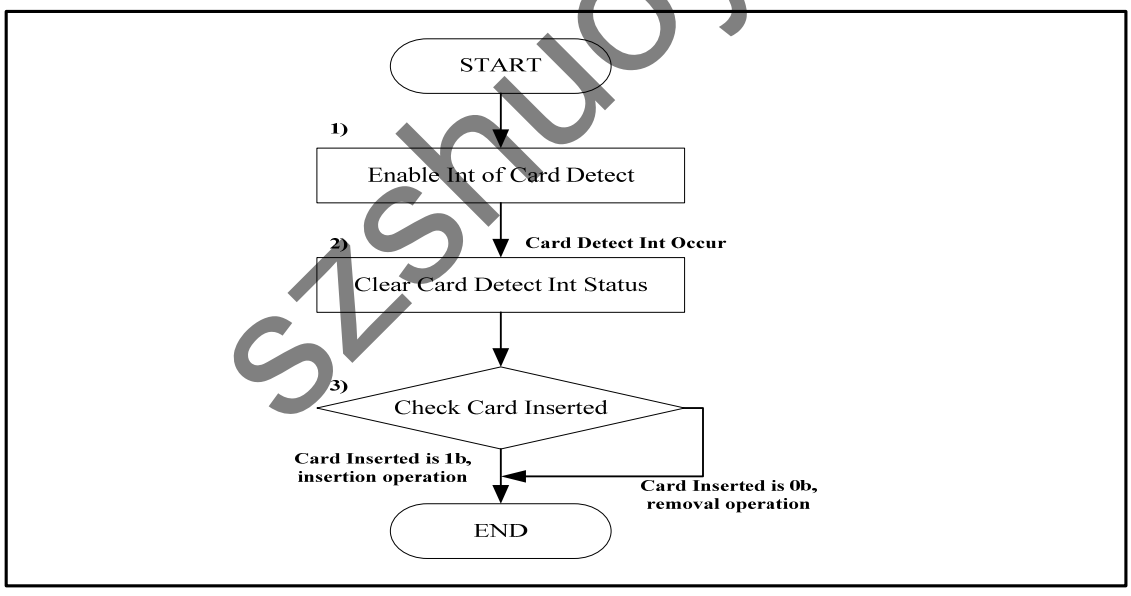


Figure 14-3: SD Card Detect Sequence

The flow chart for detecting a SD card is shown in Figure 14-3. 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.

SD Clock Supply Sequence

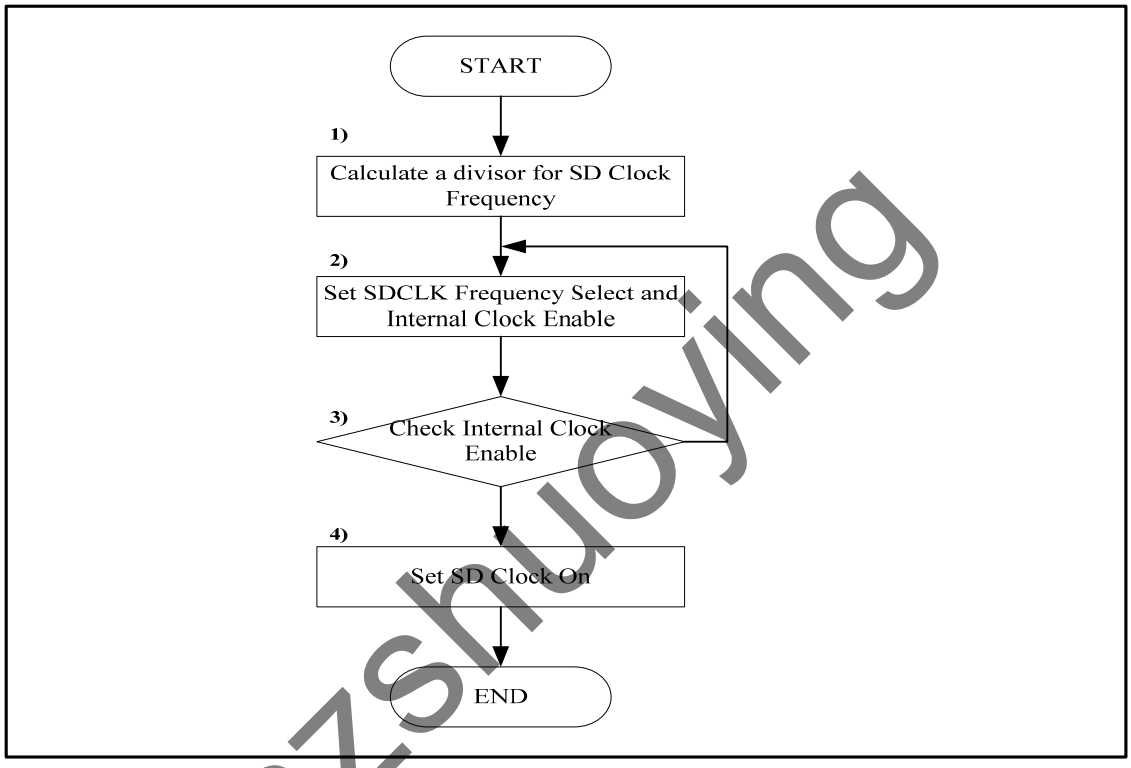


Figure 14-4: SD Clock Supply Sequence

The sequence for supplying SD Clock to a SD card is described in Figure 14-4. 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. The base clock comes from AHB bus clock or system clock generator.

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.

Timeout Setting for DAT Line

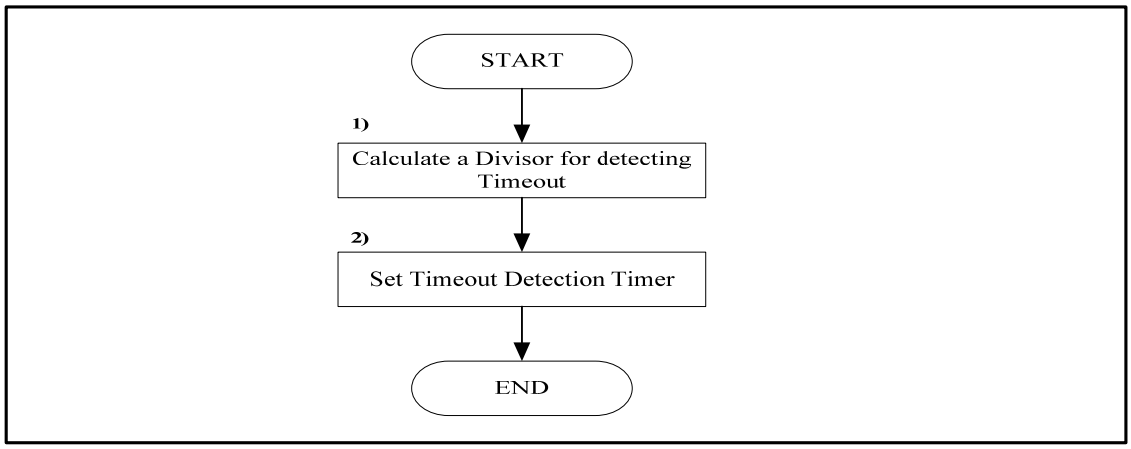


Figure 14-5: Timeout Setting Sequence

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. Timeout counter is counting by SD clock.

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

SD Command Issue Sequence

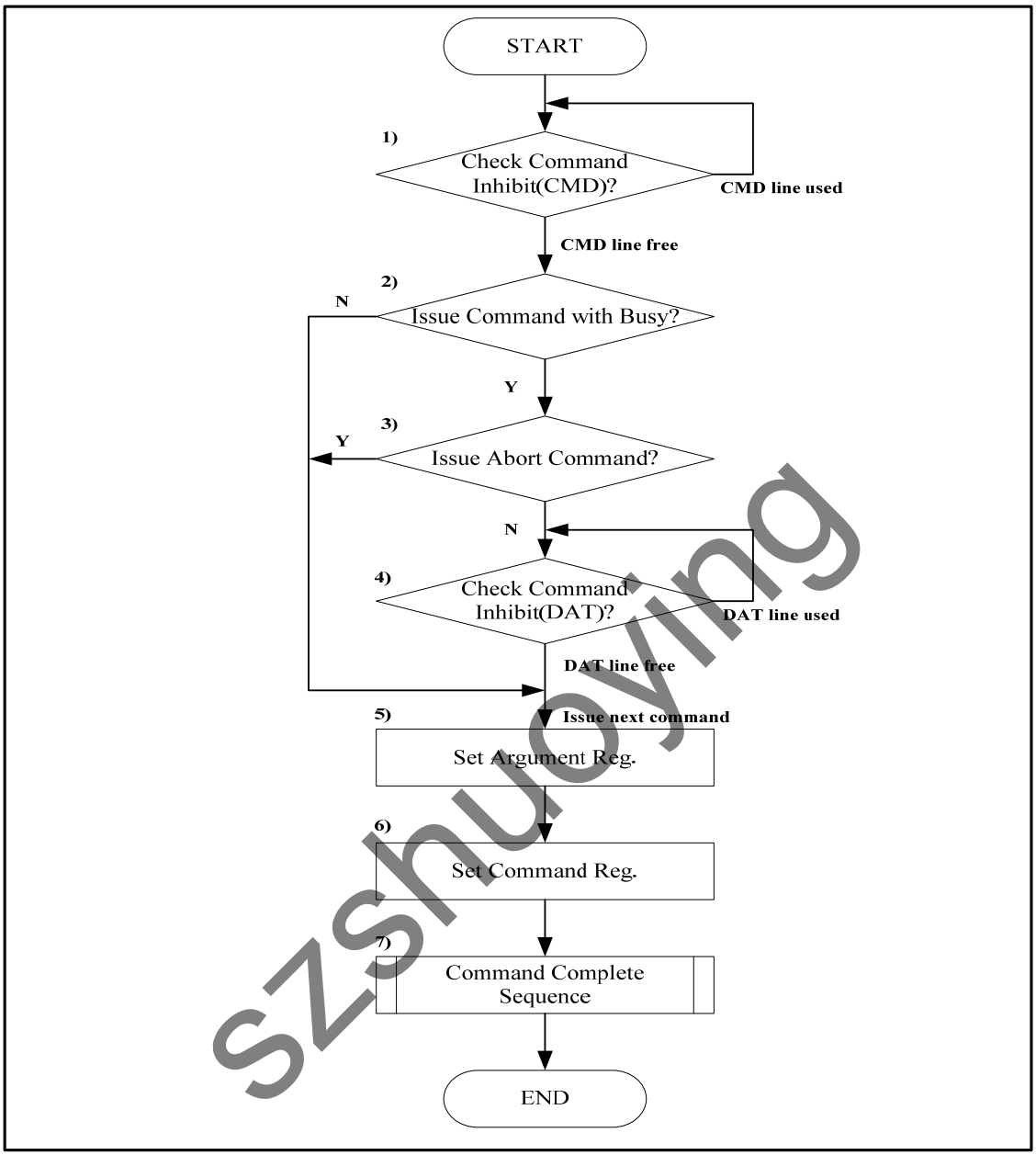


Figure 14-6: Command Issue Sequence

Take the following steps for new command issuing:

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

Command Complete Sequence

The sequence for completing the SD Command is shown in Figure 14-7. 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 Content 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.

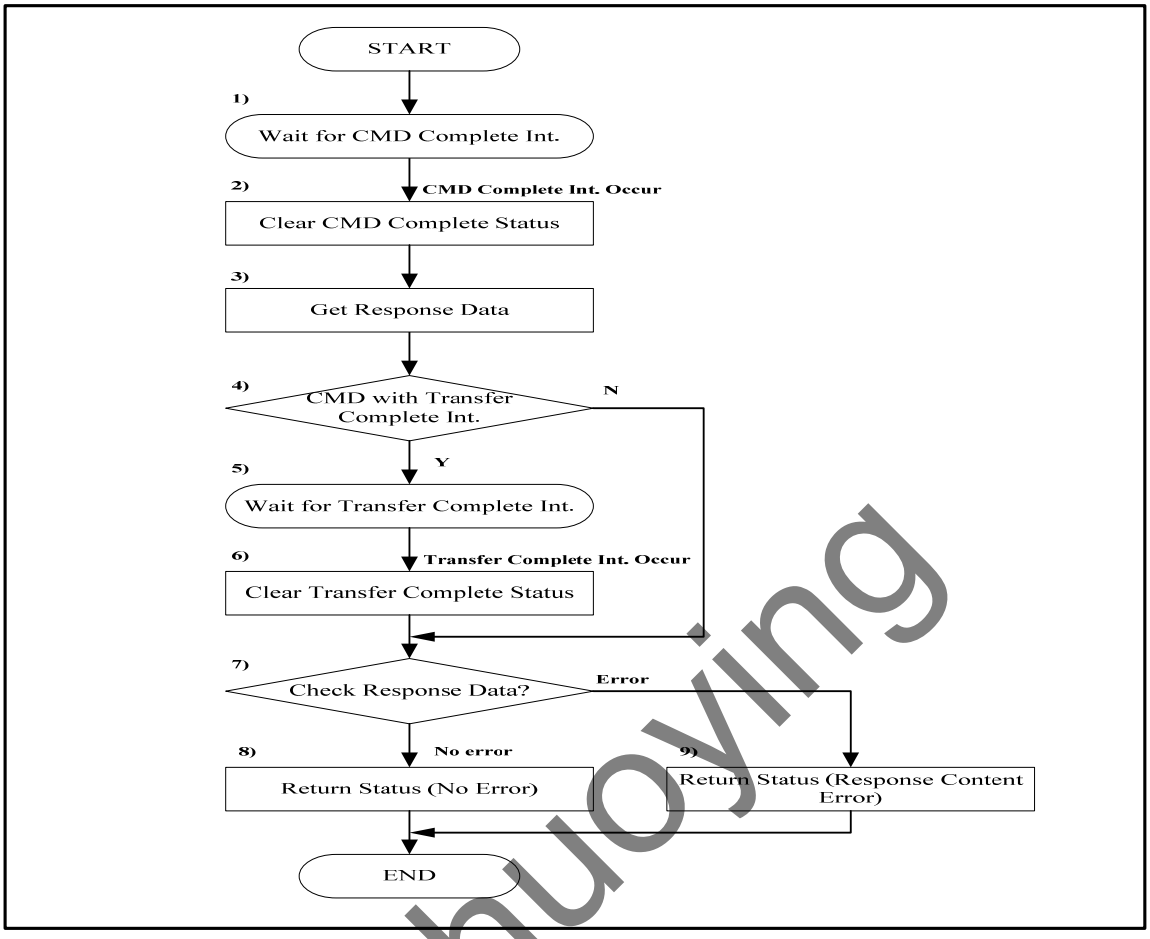


Figure 14-7: Command Complete Sequence

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 14-8 and the sequence using DMA is shown in Figure 14-9.

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.

Transfer without Using DMA

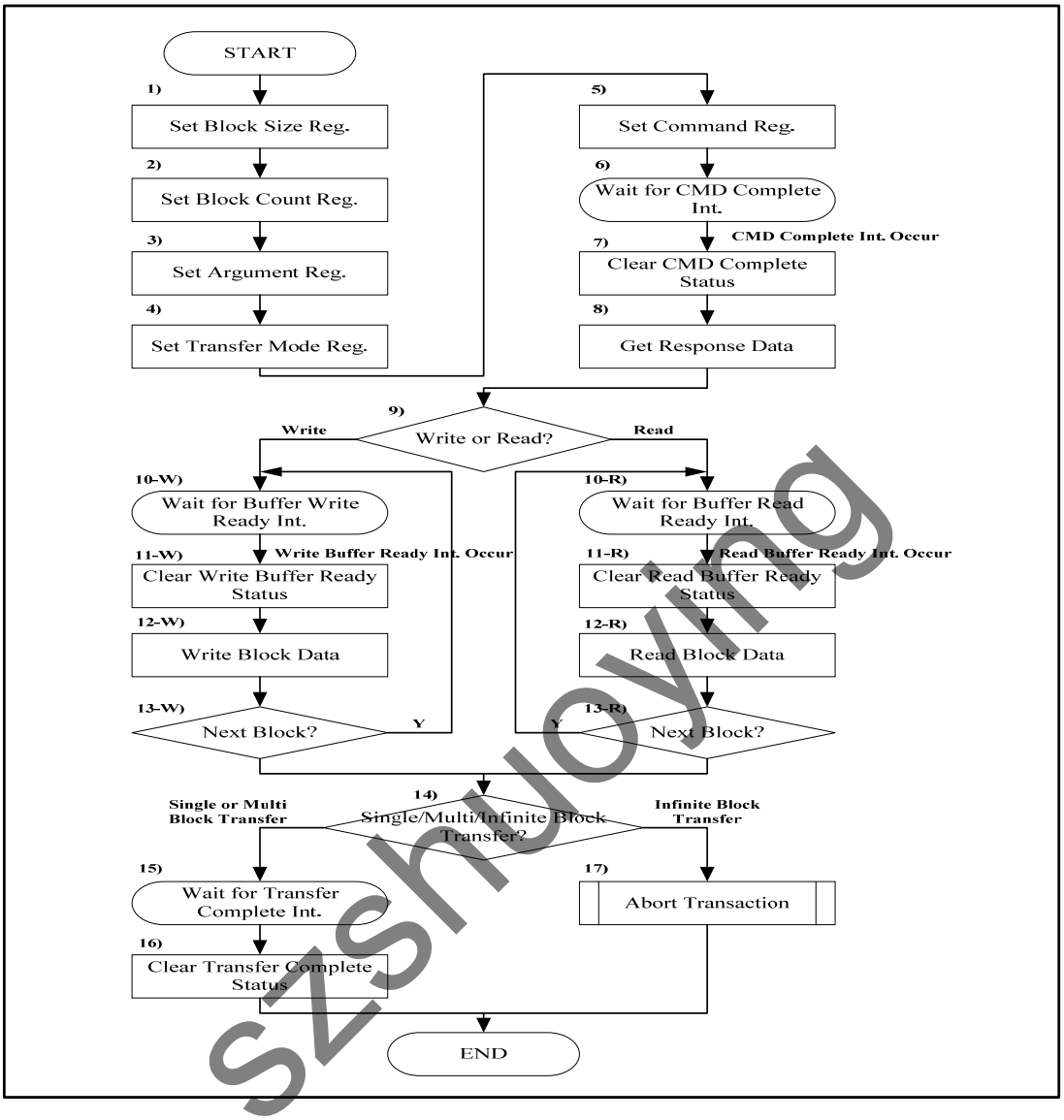


Figure 14-8: Transaction Control with Data Transfer Using DAT Line Sequence (Not Using DMA)

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.

Transfer with Using DMA

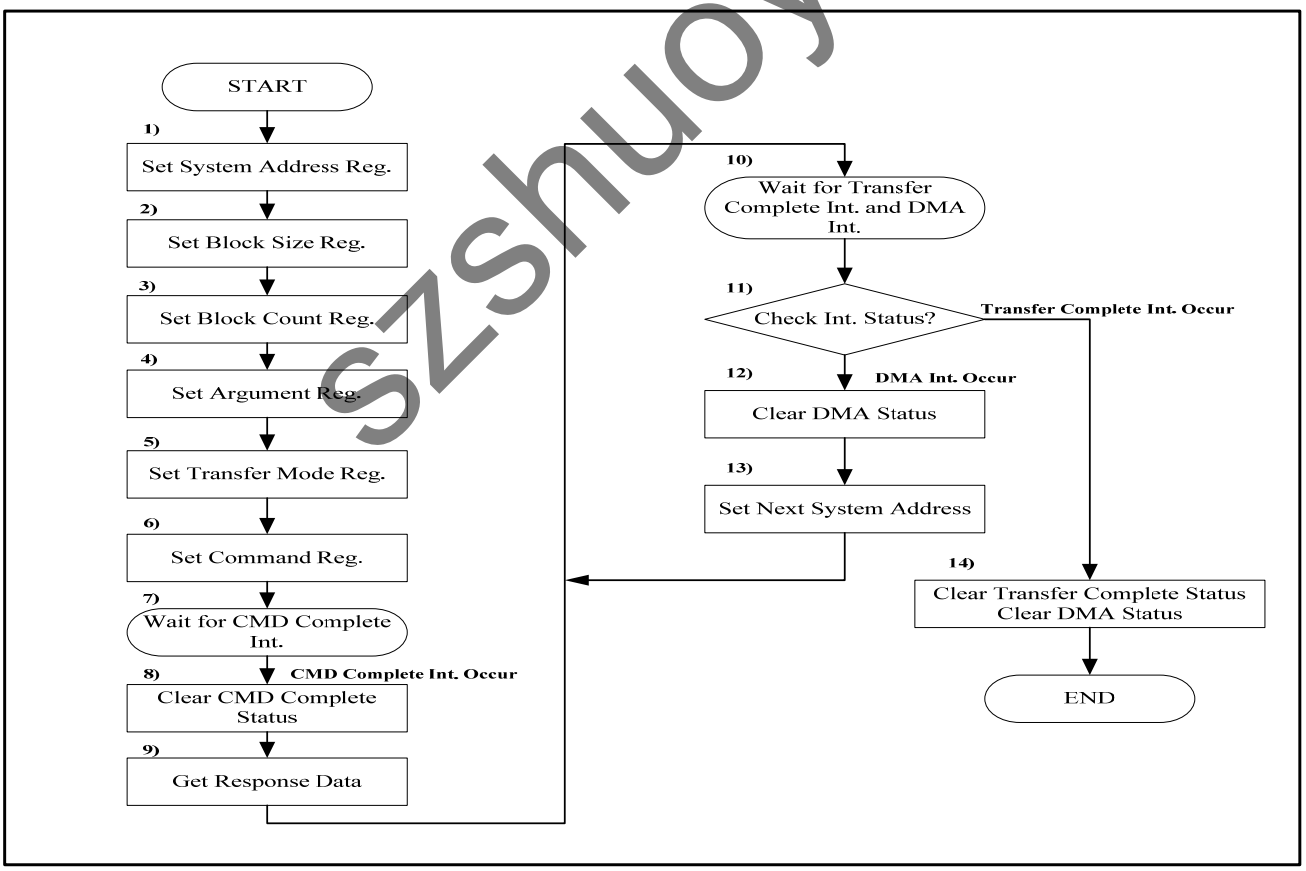


Figure 14-9: Transaction Control with Data Transfer Using DAT Line Sequence (Using DMA)

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.

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.

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.

DVB-TS Receiver

DVB-TS Receiver transfers external DVB-TS data stream to system memory. There are two transfer modes, one is async mode, the other is sync mode. In async mode, once receiver is enable and **DVBTSMST** bit in *DVBTSCTRL* register is set, any valid data will be received. Whereas in sync mode, receiver will wait for sync signal coming. As shown in Figure 14-10,

receiver start up at time t0, from time t1 data D0, D1 and so on is received in async mode. Whereas in sync mode, from time t2 data D4, D5 and so on will be received.

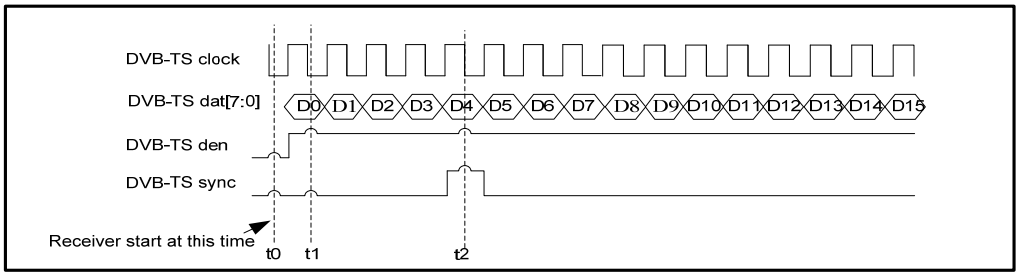


Figure 14-10: DVB-TS Receiver Timing

When controller used as DVB-TS receiver, single block, multi-block or infinite block transfer can be used. User can easily transfer continuous data stream to one, two or more ping-pong buffer through setting corresponding ADMA descriptor. One descriptor line is corresponding to one buffer. The figure below shows one four ping-pong buffer receiver.

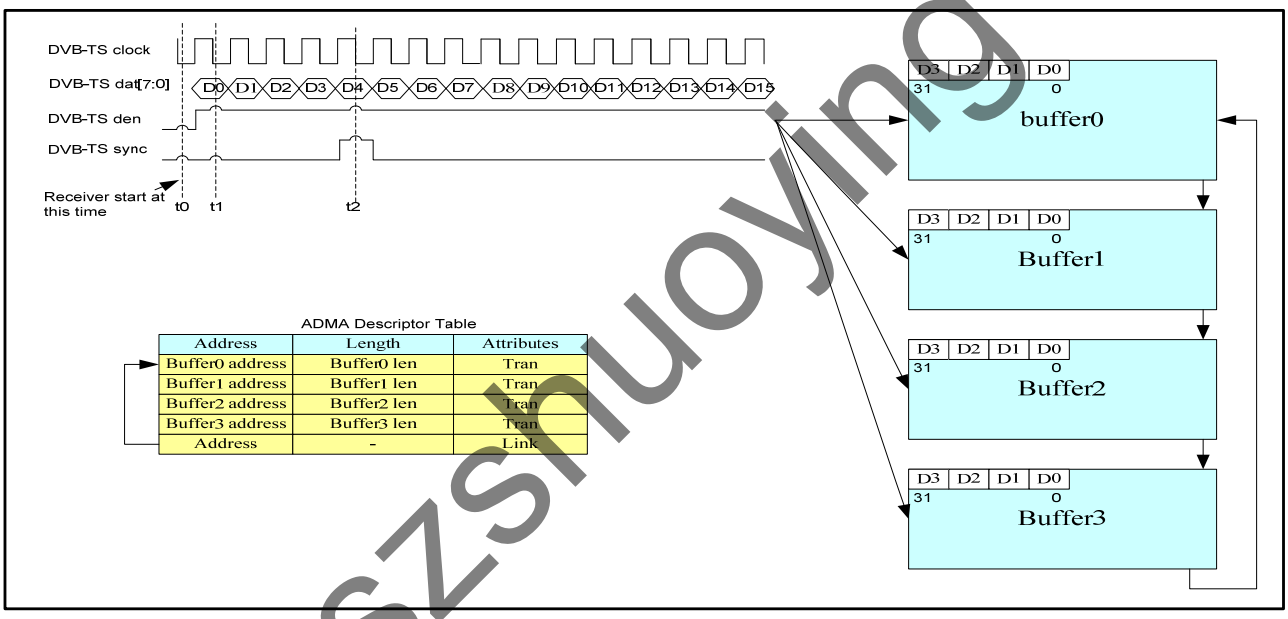


Figure 14-11: Example of Four Ping-pong Buffer DVB-TS Receiver

14.3 SDHC Register Description

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.

14.3.1 SDHC Register Memory Map

Table 14-1: SDHC Register Memory Map

Address	Symbol	Direction	Description
0x20C6_0000+n*0x1000	SDMASYSAD_n	R/W	SDMA System Address register
0x20C6_0004+n*0x1000	BLKSIZE_n	R/W	Host DMA Buffer Boundary and Transfer Block Size Register
0x20C6_0006+n*0x1000	BLKCNT_n	R/W	Blocks Count For Current Transfer Register
0x20C6_0008+n*0x1000	ARGUMENT_n	R/W	Command Argument Register
0x20C6_000C+n*0x1000	TRNMOD_n	R/W	Transfer Mode Setting Register
0x20C6_000E+n*0x1000	CMDREG_n	R/W	Command Register
0x20C6_0010+n*0x1000	RSPREG0_n	ROC	Response Register 0
0x20C6_0014+n*0x1000	RSPREG1_n	ROC	Response Register 1
0x20C6_0018+n*0x1000	RSPREG2_n	ROC	Response Register 2
0x20C6_001C+n*0x1000	RSPREG3_n	ROC	Response Register 3
0x20C6_0020+n*0x1000	BUFDAT_n	R/W	Buffer Data Port Register
0x20C6_0024+n*0x1000	PRNSTS_n	R/W	Present State Register
0x20C6_0028+n*0x1000	HOSTCTL_n	R/W	Host Control Register
0x20C6_0029+n*0x1000	PWRCON_n	R/W	Power Control Register
0x20C6_002A+n*0x1000	BLKGAP_n	R/W	Block Gap Control Register
0x20C6_002B+n*0x1000	WAKCON_n	R/W	Wakeup Control Register
0x20C6_002C+n*0x1000	CLKCON_n	R/W	Clock Control Register
0x20C6_002E+n*0x1000	TIMEOUTCON_n	R/W	Timeout Control Register
0x20C6_002F+n*0x1000	SWRST_n	R/W	Software Reset Register
0x20C6_0030+n*0x1000	NORINTSTS_n	ROC/RW1C	Normal Interrupt Status Register
0x20C6_0032+n*0x1000	ERRINTSTS_n	ROC/RW1C	Error Interrupt Status Register
0x20C6_0034+n*0x1000	NORINTSTSEN_n	R/W	Normal Interrupt Status Enable Register
0x20C6_0036+n*0x1000	ERRINTSTSEN_n	R/W	Error Interrupt Status Enable Register
0x20C6_0038+n*0x1000	NORINTSIGEN_n	R/W	Normal Interrupt Signal Enable Register
0x20C6_003A+n*0x1000	ERRINTSIGEN_n	R/W	Error Interrupt Signal Enable Register
0x20C6_003C+n*0x1000	ACMD12ERRSTS_n	ROC	Auto CMD12 Error Status Register
0x20C6_0040+n*0x1000	CAPAREG_n	HWInit	Capabilities Register
0x20C6_0048+n*0x1000	MAXCURR_n	HWInit	Maximum Current Capabilities Register
0x20C6_0050+n*0x1000	FEAER_n	WO	Force Event Auto CMD12 Error Interrupt Register
0x20C6_0052+n*0x1000	FEERR_n	WO	Force Event Error Interrupt Register Error Interrupt
0x20C6_0054+n*0x1000	ADMAERR_n	R/W	ADMA Error Status Register
0x20C6_0058+n*0x1000	ADMASYSADDR_n	R/W	ADMA System Address Register
0x20C6_0080+n*0x1000	CONTROL0_n	R/W	Controller control0 register
0x20C6_0088+n*0x1000	CONTROL2_n	R/W	Controller control2 register
0x20C6_0090+n*0x1000	DVBTSCTRL_n	R/W	DVB TS Control Register
0x20C6_0094+n*0x1000	DVBTSSTA_n	R/W	DVB TS Status Register
0x20C6_00FC+n*0x1000	SLTINTSTA_n	ROC	Slot Interrupt Status Register
0x20C6_00FE+n*0x1000	HCVER_n	RO	Host Controller Version register

Note: Where n is set to 0 to 2, corresponding to SDHC channel 0, 1 and 2.

14.3.2 SDHC Registers and Field Descriptions

Register 14-1: SDMA System Address Register (SDMASYSAD_n, offset = 0x0)

Field	Symbol	Direction	Description	Default
[31:0]	SDMASYSAD	R/W	**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.	0

Register 14-2: Host DMA Buffer Boundary and Transfer Block Size Register (BLKSIZE, offset=0x4)

Field	Symbol	Direction	Description	Default
[14:12]	BUFBOUND	R/W	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)	0

			101b = 128K Bytes (Detects A16 carry out) 110b = 256K Bytes (Detects A17 carry out) 111b = 512K Bytes (Detects A18 carry out)	
[11:0]	BLKSIZE	R/W	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

Register 14-3: Blocks Count for Current Transfer Register (BLKCNT_n, offset=0x6)

Field	Symbol	Direction	Description	Default
[15:0]	BLKCNT	R/W	**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

Register 14-4: Command Argument Register (ARGUMENT_n, offset=0x8)

Field	Symbol	Direction	Description	Default
[31:0]	ARGUMENT	R/W	**Command Argument** The SD Command Argument is specified as bit[39:8] of Command-Format in the SD Memory Card Physical Layer Specification.	0

Register 14-5: Transfer Mode Setting Register (TRNMOD_n, offset=0xC)

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.

Field	Symbol	Direction	Description	Default
[5]	MUL1SIN0	R/W	**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
[4]	RD1WT0	R/W	**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
[2]	ENACMD12	R/W	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
[1]	ENBLKCNT	R/W	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
[0]	ENDMA	R/W	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 14-2: 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

Register 14-6: Command Register (CMDREG_n, offset=0xE)

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.

Field	Symbol	Direction	Description	Default
[13:8]	CMDIDX	R/W	**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.	0
[7:6]	CMDTYP	R/W	**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 software reset. (Refer to Abort Transaction) 11b = Abort CMD12, CMD52 for writing "I/O Abort" in CCCR 10b = Resume CMD52 for writing "Function Select" in CCCR 01b = Suspend CMD52 for writing "Bus Suspend" in CCCR 00b = Normal Other commands	0
[5]	DATAPRNT	R/W	**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	0

			DAT[0] line (R1b or R5b ex. CMD38) (3) Resume command 1 = Data Present 0 = No Data Present	
[4]	ENCMDIDX	R/W	**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	0
[3]	ENCMDCRC	R/W	**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	0
[1:0]	RSPTYP	R/W	**Response Type Select** 00 = No Response 01 = Response Length 136 10 = Response Length 48 11 = Response Length 48 check Busy after response	0

Table 14-3: 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,R5,R6
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.

Register 14-7: Response Register (RSPREG0/1/2/3_n, offset=0x10,0x14,0x18,0x1C)

Field	Symbol	Direction	Description	Default
[31:0]	RSPREG3	ROC	CMD Response[127:96]	0
[31:0]	RSPREG2	ROC	CMD Response[95:64]	0
[31:0]	RSPREG1	ROC	CMD Response[63:32]	0
[31:0]	RSPREG0	ROC	CMD Response[31:0]	0

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.

Table 14-4: 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.	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.

Register 14-8: Buffer Data Port Register (BUFDAT_n, offset=0x20)

Field	Symbol	Direction	Description	Default
[31:0]	BUFDAT	R/W	**Buffer Data** The Host Controller internal buffer can be accessed by this data port register. **Note**: Support byte, half-word and word access.	0

Register 14-9: Present State Register (PRNSTS_n, offset=0x24)

Field	Symbol	Direction	Description	Default
[24]	PRNTCMD	RO	CMD Line Signal Level This status is used to check the CMD line level to recover from errors, and for debugging.	Line state
[23:20]	PRNTDAT	RO	DAT[3:0] Line Signal Level 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].	Line state
[19]	Reserved	RO	Reserved	1
[18]	PRNTCD	RO	Card Detect Pin Level 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)	Line state
[17]	STBLCARD	RO	**Card State Stable** 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**	1

			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	
[16]	INSCARD	RO	Card Inserted 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
[11]	BUFRDRDY	ROC	Buffer Read Enable 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
[10]	BUFWTRDY	ROC	Buffer Write Enable 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
[9]	RDTRANACT	ROC	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	0

			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	
[8]	WTTRANACT	ROC	Write Transfer Active 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
[2]	DATLINEACT	ROC	DAT Line Active 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	0

			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	
[1]	CMDINHDAT	ROC	Command Inhibit (DAT) 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
[0]	CMDINHCMD	ROC	Command Inhibit (CMD) 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.

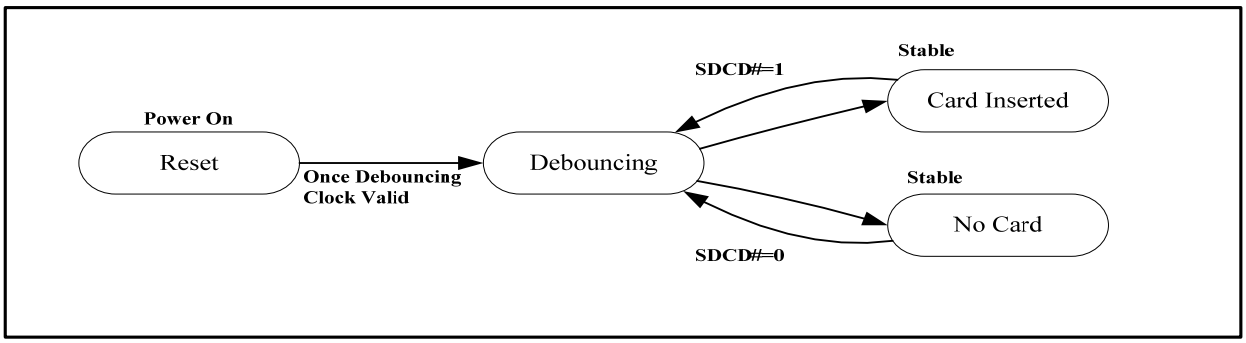


Figure 14-12: Card Detect State

The above figure shows the state definitions of hardware that handles "Debouncing".

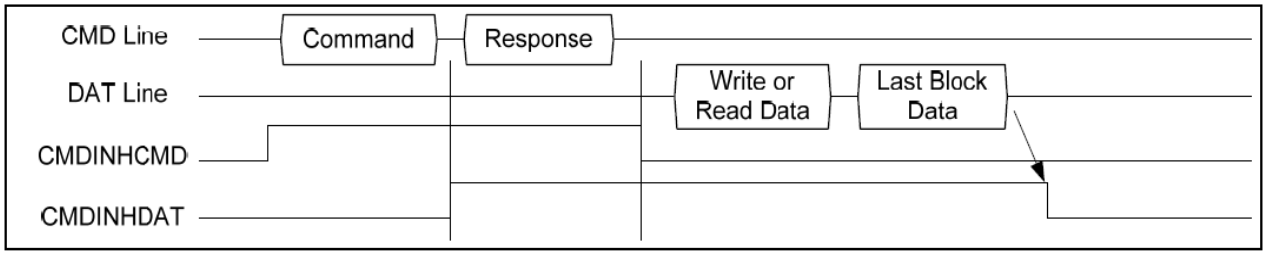


Figure 14-13: Timing of Command Inhibit (DAT) and Command Inhibit (CMD) with Data Transfer

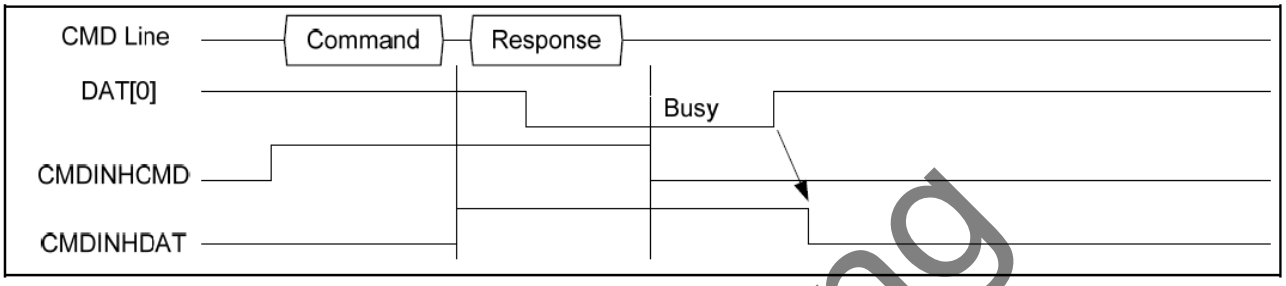


Figure 14-14: Timing of Command Inhibit (DAT) for the case of Response With Busy

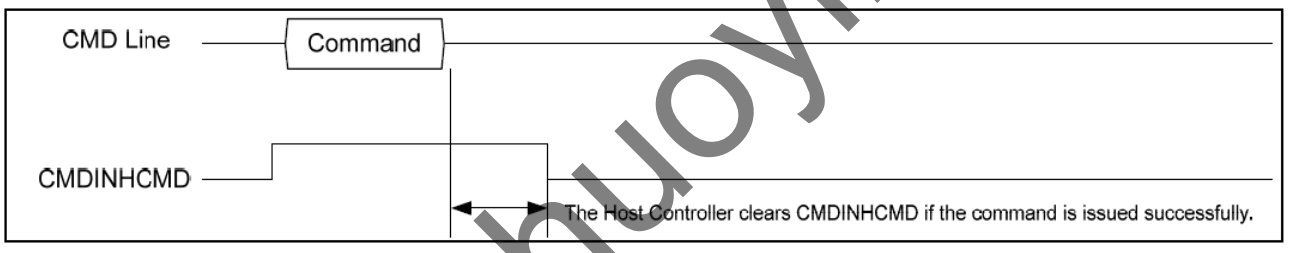


Figure 14-15: Timing of Command Inhibit (CMD) for the case of No Response Command

Register 14-10: Host Control Register (HOSTCTL_n, offset=0x28)

Field	Symbol	Direction	Description	Default
[7]	SDSIGSEL	R/W	**Card Detect Signal Selection** This bit selects source for the card detection: 1=The Card Detect Test Level is selected, i.e. use SDCDTL for test purpose. 0=SDCD# is selected (for normal use) When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.	0
[6]	SDCDTL	R/W	**Card Detect Test Level** This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not 1=Card Inserted 0=No Card	0
[5]	Reserved	RO	**Reserved**	0

[4:3]	DMASEL	R/W	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
[2]	HSEN	R/W	High Speed Enable 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. If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock 1=High Speed Mode 0=Normal Speed Mode	0
[1]	WIDE4	R/W	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	RO	Reserved	0

Register 14-11: Power Control Register (PWRCON_n, offset=0x29)

Field	Symbol	Direction	Description	Default
[3:1]	SELPWRLVL	R/W	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 shall check the Voltage Support bits in the Capabilities register. If an unsupported voltage is selected, the Host System shall not supply SD Bus voltage. '111b' = 3.3V (Typ.) '110b' = 3.0V (Typ.) '101b' = 1.8V (Typ.) '100b' – '000b' = Reserved	0
[0]	PWRON	R/W	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 **Note**: This bit is under control by **SDSIGPC** bit in *CONTRL0* register. When **SDSIGPC** is set to 0, Host Controller will always power on SD bus.	0

Register 14-12: Block Gap Control Register (BLKGAP_n, offset=0x2A)

Field	Symbol	Direction	Description	Default
[3]	ENINTBGAP	R/W	Interrupt At Block Gap This bit is valid only in 4-bit mode of the SDIO card and selects a	0

			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. '1' = Enabled, '0' = Disabled **Note**: Interrupt at Block Gap operation is not supported in IMAPX210 controller, it should be fixed to 0.	
[2]	ENRWAIT	R/W	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. '1' = Enable Read Wait Control, '0' = Disable Read Wait Control	0
[1]	CONTREQ	RWAC	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. '1' = Restart, '0' = Not affect	0
[0]	STOPBGAP	R/W	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	0

			register. Regarding detailed control of bits D01 and D00. '1' = Stop '0' = Transfer	

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 Block 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.

Register 14-13: Clock Control Register (CLKCON_n, offset=0x2C)

Field	Symbol	Direction	Description	Default
[15:8]	SELFREQ	R/W	**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. 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, from AHB bus clock or system clock generator depending on **SELBASECLK** bit in *CONTROL0* registesr.	0
[2]	ENSDCLK	R/W	**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. '1' = Enable '0' = Disable	0
[1]	STBLINTCLK	ROC	**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.	0

			Note: This is useful when using PLL for a clock oscillator that requires setup time. '1' = Ready '0' = Not Ready	
[0]	ENINTCLK	R/W	**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. '1' = Oscillate '0' = stop	0

Register 14-14: Timout Control Register (TIMEOUTCON_n, offset=0x2E)

Field	Symbol	Direction	Description	Default
[3:0]	TIMEOUTCON	R/W	**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} Note: TMCLK is equal to SDCLK.	0

Register 14-15: Software Reset Register (SWRST_n, offset=0x2F)

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.

Field	Symbol	Direction	Description	Default
[2]	RSTDAT	RWAC	**Software Reset For DAT Line** Only part of data circuit is reset. DMA circuit is also reset. 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	0

			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	
[1]	RSTCMD	RWAC	**Software Reset For CMD Line** Only part of command circuit is reset. 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
[0]	RSTALL	RWAC	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. '1' = Reset '0' = Work	0

Register 14-16: Normal Interrupt Status Register (NORINTSTS_n, offset=0x30)

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.

Field	Symbol	Direction	Description	Default
[15]	STAERR	ROC	**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. '0' = No Error '1' = Error	0
[10]	STARWAIT	RW1C	Read Wait Interrupt Status '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
[8]	STACARDINT	RW1C	Card Interrupt	0

			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. '1' = Generate Card Interrupt '0' = No Card Interrupt **Note1**: Card Interrupt status bit keeps previous value until next card interrupt period (level interrupt) and can be cleared when write to 1. **Note2**: SDHC Controller of the IMAPX210 does not support "card interrupt at block gap" used when the multiple block 4-bit operation.	
[7]	STACARDREM	RW1C	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. '1' = Card removed '0' = Card state stable or Debouncing	0
[6]	STACARDINS	RW1C	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. '1' = Card inserted '0' = Card state stable or Debouncing	0
[5]	STABUFRDRDY	RW1C	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. '1' = Ready to read buffer '0' = Not ready to read buffer	0
[4]	STABUFWTRDY	RW1C	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. '1' = Ready to write buffer '0' = Not ready to write buffer	0

<table>
<tr><td>[3]</td><td>STADMAINT</td><td>RW1C</td><td>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.
‘1’ = DMA Interrupt is generated
‘0’ = No DMA Interrupt</td><td>0</td></tr>
<tr><td>[2]</td><td>STABLKGAP</td><td>RW1C</td><td>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.
(1) In the case of a Read Transaction
This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing. The Read Wait must be supported in order to use this function.
(2) Case of Write Transaction
This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing).
‘1’ = Transaction stopped at block gap
‘0’ = No Block Gap Event</td><td>0</td></tr>
<tr><td>[1]</td><td>STATRANCMPLT</td><td>RW1C</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).
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><td>Transfer Complete</td><td>Data Timeout Error</td><td>Meaning of the status</td></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>
</td><td>0</td></tr>
</table>

			'1' = Data Transfer Complete '0' = No transfer complete	
[0]	STACMDCMPLT	RW1C	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. Command Complete / Command Timeout Error / Meaning of the status 0 / 0 / Interrupted by another factor Don't care / 1 / Response not received within 64 SDCLK cycles 1 / 0 / Response received '1' = Command Complete '0' = No command complete	0

Register 14-17: Error Interrupt Status Register (ERRINTSTS_n, offset=0x32)

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.

Field	Symbol	Direction	Description	Default
[9]	STAADMAERR	RW1C	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
[8]	STAACMDERR	RW1C	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
[6]	STADENDERR	RW1C	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
[5]	STADATCRCERR	RW1C	Data CRC Error Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC status	0

			having a value of other than "010". '1' = Error '0' = No Error	
[4]	STADATTOUTERR	RW1C	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. '1' = Timeout '0' = No Error	0
[3]	STACMDIDXERR	RW1C	Command Index Error Occurs if a Command Index error occurs in the command response. '1' = Error '0' = No Error	0
[2]	STACMDEBITERR	RW1C	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	0
[1]	STACMDCRCERR	RW1C	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
[0]	STACMDTOUTERR	RW1C	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 below, 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.

Table 14-5: 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

Register 14-18: Normal Interrupt Status Enable Register (NORINTSTSEN_n, offset=0x34)

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.

Field	Symbol	Direction	Description	Default
[15]	Reserved	RO	Fixed to 0 The Host Driver shall control error interrupts using the Error Interrupt Status Enable register.	0
[10]	ENSTARWAIT	R/W	Read Wait Interrupt Status enable 1=enable, 0=masked	0
[8]	ENSTACARDINT	R/W	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
[7]	ENSTACARDREM	R/W	Card Removal status enable 1=enable, 0=masked	0
[6]	ENSTACARDINS	R/W	Card Insertion status enable 1=enable, 0=masked	0
[5]	ENSTABUFRDRDY	R/W	Buffer Read Ready status enable 1=enable, 0=masked	0
[4]	ENSTABUFWTRDY	R/W	Buffer Write Ready status enable 1=enable, 0=masked	0
[3]	ENSTADMA	R/W	DMA Interrupt status enable 1=enable, 0=masked	0
[2]	ENSTABLKGAP	R/W	Block Gap Event status enable 1=enable, 0=masked	0
[1]	ENSTATRANCMPLT	R/W	Transfer Complete status enable 1=enable, 0=masked	0
[0]	ENSTACMDCMPLT	R/W	Command Complete status enable 1=enable, 0=masked	0

Register 14-19: Error Interrupt Status Enable Register (ERRINTSTSEN_n, offset=0x36)

Field	Symbol	Direction	Description	Default
[9]	ENSTAADMAERR	R/W	ADMA Error status enable 1=enable, 0=masked	0
[8]	ENSTAACMDERR	R/W	Auto CMD12 Error status enable 1=enable, 0=masked	0
[6]	ENSTADENDERR	R/W	Data End Bit Error status enable 1=enable, 0=masked	0
[5]	ENSTADATCRCERR	R/W	Data CRC Error status enable 1=enable, 0=masked	0
[4]	ENSTADATTOUTERR	R/W	Data Timeout Error status enable 1=enable, 0=masked	0

[3]	ENSTACMDIDXERR	R/W	Command Index Error status enable 1=enable, 0=masked	0
[2]	ENSTACMDEBITERR	R/W	Command End Bit Error status enable 1=enable, 0=masked	0
[1]	ENSTACMDCRCERR	R/W	Command CRC Error status enable 1=enable, 0=masked	0
[0]	ENSTACMDTOUTERR	R/W	Command Timeout Error status enable 1=enable, 0=masked	0

Register 14-20: Normal Interrupt Signal Enable Register (NORINTSIGEN_n, offset=0x38)

This register is used to select which interrupt status is indicated 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.

Field	Symbol	Direction	Description	Default
[15]	Reserved	RO	Fixed to 0 The Host Driver shall control error interrupts using the Error Interrupt Signal Enable register.	0
[10]	ENSIGRWAIT	R/W	Read Wait Interrupt Signal enable 1=enable, 0=masked	0
[8]	ENSIGCARDINT	R/W	Card Interrupt Signal Enable 1=enable, 0=masked	0
[7]	ENSIGCARDREM	R/W	Card Removal Signal enable 1=enable, 0=masked	0
[6]	ENSIGCARDINS	R/W	Card Insertion Signal enable 1=enable, 0=masked	0
[5]	ENSIGBUFRDRDY	R/W	Buffer Read Ready Signal enable 1=enable, 0=masked	0
[4]	ENSIGBUFWTRDY	R/W	Buffer Write Ready Signal enable 1=enable, 0=masked	0
[3]	ENSIGDMA	R/W	DMA Interrupt Signal enable 1=enable, 0=masked	0
[2]	ENSIGBLKGAP	R/W	Block Gap Event Signal enable 1=enable, 0=masked	0
[1]	ENSIGTRANCMPLT	R/W	Transfer Complete Signal enable 1=enable, 0=masked	0
[0]	ENSIGCMDCMPLT	R/W	Command Complete Signal enable 1=enable, 0=masked	0

Register 14-21: Error Interrupt Signal Enable Register (ERRINTSIGEN_n, offset=0x3A)

This register is used to select which interrupt status is indicated 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.

Field	Symbol	Direction	Description	Default
[9]	ENSIGADMAERR	R/W	ADMA Error Signal enable 1=enable, 0=masked	0
[8]	ENSIGACMDERR	R/W	Auto CMD12 Error Signal enable 1=enable, 0=masked	0
[6]	ENSIGDENDERR	R/W	Data End Bit Error Signal enable 1=enable, 0=masked	0
[5]	ENSIGDATCRCERR	R/W	Data CRC Error Signal enable 1=enable, 0=masked	0

[4]	ENSIGDATTOUTERR	R/W	Data Timeout Error Signal enable 1=enable, 0=masked	0
[3]	ENSIGCMDIDXERR	R/W	Command Index Error Signal enable 1=enable, 0=masked	0
[2]	ENSIGCMDEBITERR	R/W	Command End Bit Error Signal enable 1=enable, 0=masked	0
[1]	ENSIGCMDCRCERR	R/W	Command CRC Error Signal enable 1=enable, 0=masked	0
[0]	ENSIGCMDTOUTERR	R/W	Command Timeout Error Signal enable 1=enable, 0=masked	0

Register 14-22: AUTOCMD12 Error Status Register (ACMD12ERRSTS_n, offset=0x3C)

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.

Field	Symbol	Direction	Description	Default
[7]	STANCMDAER	ROC	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
[4]	STACMDIDXAER	ROC	Auto CMD12 Index Error Occurs if the Command Index error occurs in response to a command. '1' = Error '0' = No Error	0
[3]	STACMDEBITAER	ROC	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
[2]	STACMDCRCAER	ROC	Auto CMD12 CRC Error Occurs when detecting a CRC error in the command response. '1' = CRC Error Generated '0' = No Error	0
[1]	STACMDTOUTAER	ROC	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
[0]	STANACMDAER	ROC	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 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.

Register 14-23: Capabilities Register (CAPAREG_n, offset=0x40)

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.

Field	Symbol	Direction	Description	Default
[26]	CAPAV18	HWInit	Voltage Support 1.8V '1'=1.8V Supported '0'=1.8V Not Supported	1
[25]	CAPAV30	HWInit	Voltage Support 3.0V '1'=3.0V Supported '0'=3.0V Not Supported	1
[24]	CAPAV33	HWInit	Voltage Support 3.3V '1'=3.3V Supported '0'=3.3V Not Supported	1
[23]	CAPASUSRES	HWInit	Suspend/Resume Support 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
[22]	CAPADMA	HWInit	DMA Support 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
[21]	CAPAHSPD	HWInit	High Speed Support 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
[19]	CAPAADMA2	HWInit	ADMA2 Support This bit indicates whether the Host Controller is capable of using ADMA2. '1'=ADMA2 Support	1

			'0'=ADMA2 not Support	
[17:16]	CAPAMAXBLKLEN	HWInit	Max Block Length 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	00
[13:8]	CAPABASECLK	HWInit	Base Clock Frequency For SD Clock 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	00000
[7]	CAPATOUTUNIT	HWInit	Timeout Clock Unit This bit shows the unit of base clock frequency used to detect Data Timeout Error. '0'=KHz, '1'=MHz	1
[5:0]	CAPATOUTCLK	HWInit	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 **Note**: Timeout Clock of SDHC controller in IMAPX210 is equal to SDCLK.	000000

Register 14-24: Maximum Current Capabilities Register (MAXCURR_n, offset=0x48)

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.

Field	Symbol	Direction	Description	Default
[23:16]	MAXCURR18	HWInit	Maximum Current for 1.8V	0
[15:8]	MAXCURR30	HWInit	Maximum Current for 3.0V	0
[7:0]	MAXCURR33	HWInit	Maximum Current for 3.3V	0

Register 14-25: Force Event AutoCMD12 Error Interrupt Register (FEAER_n, offset=0x50)

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

Field	Symbol	Direction	Description	Default
[7]	FENCMDAER	WO	Force Event for Command Not Issued By Auto CMD12 Error 1=Interrupt is generated	0

			0=No Interrupt	
[4]	FECMDIDXERR	WO	Force Event for Auto CMD12 Index Error 1=Interrupt is generated 0=No Interrupt	0
[3]	FECMDEBITAER	WO	Force Event for Auto CMD12 End Bit Error 1=Interrupt is generated 0=No Interrupt	0
[2]	FECMDCRCAER	WO	Force Event for Auto CMD12 CRC Error 1=Interrupt is generated 0=No Interrupt	0
[1]	FECMDTOUTAER	WO	Force Event for Auto CMD12 Timeout Error 1=Interrupt is generated 0=No Interrupt	0
[0]	FENACMDAER	WO	Force Event for Auto CMD12 Not Executed 1=Interrupt is generated 0=No Interrupt	0

Register 14-26: Force Event Error Interrupt Register (FEERR_n, offset=0x52)

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.

Field	Symbol	Direction	Description	Default
[9]	FEADMAERR	WO	Force Event for ADMA Error 1=Interrupt is generated 0=No Interrupt	0
[8]	FEACMDERR	WO	Force Event for Auto CMD12 Error 1=Interrupt is generated 0=No Interrupt	0
[7]	FECURERR	WO	Force Event for Current Limit Error 1=Interrupt is generated 0=No Interrupt	0
[6]	FEDENDERR	WO	Force Event for Data End Bit Error 1=Interrupt is generated 0=No Interrupt	0
[5]	FEDATCRCERR	WO	Force Event for Data CRC Error 1=Interrupt is generated 0=No Interrupt	0
[4]	FEDATTOUTERR	WO	Force Event for Data Timeout Error 1=Interrupt is generated 0=No Interrupt	0
[3]	FECMDIDXERR	WO	Force Event for Command Index Error 1=Interrupt is generated 0=No Interrupt	0
[2]	FECMDEBITERR	WO	Force Event for Command End Bit Error 1=Interrupt is generated 0=No Interrupt	0
[1]	FECMDCRCERR	WO	Force Event for Command CRC Error	0

			1=Interrupt is generated 0=No Interrupt	
[0]	FECMDTOUTERR	WO	Force Event for Command Timeout Error 1=Interrupt is generated 0=No Interrupt	0

Register 14-27: ADMA Error Status Register (ADMAERR_n, offset=0x54)

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.

Field	Symbol	Direction	Description	Default
[10]	STAADMAFINBLK	ROC	ADMA Final Block Transferred 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
[8]	ADMASTAINT	RW1C	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
[2]	ADMALENMISERR	RW1C	ADMA Length Mismatch Error This error occurs in the following 2 cases. (1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. (2) Total data length can not be divided by the block length. '0' = No Error '1'= Error	0
[1:0]	ADMAERRST	RO	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	00

Register 14-28: ADMA System Address Register (ADMASYSADDR_n, offset=0x58)

Field	Symbol	Direction	Description	Default
[31:0]	ADMASYSAD	R/W	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).	0

Register 14-29: Controller Control0 Register (CONTROL0_n, offset=0x80)

Field	Symbol	Direction	Description	Default
[30]	ENCMDCNFMSK	R/W	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 HSEN field in the Host Control Register is set (High Speed mode enable), this field should be enabled to prevent from command conflict status alarm.	0
[29]	CDINVRXD3	R/W	Card Detect signal inversion for RX_DAT[3] 0=Disable, 1=Enable	0
[28]	SELCARDOUT	R/W	Card Removed Condition Selection 0= Card Removed condition is "Not Card Insert" State (When the transition from "Card Inserted" state to "Debouncing" state in) 1= Card Removed state is "Card Out" State (When the transition from "Debouncing state to "No Card" state in Figure 14-12)	0
[19:16]	LVLDAT	RO	DAT[3:0] Line level	Line state
[13]	SDCDSEL	R/W	SD Card Detect Signal Selection Card Detect Pin Level does not simply reflect SDCD# pin, but chooses from SDCD, DAT[3], or SDCDTL 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
[12]	SDSIGPC	R/W	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

			'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	
[6]	DISBUFRD	R/W	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
[5]	SELBASECLK	R/W	Base Clock Source Select 0=HCLK 1=System Clock Generator	0
[0]	HWINITFIN	RO	SD Host Controller Hardware Initialization Finish 0=Not Finish, 1=Finish	0

Register 14-30: Control2 Register (CONTROL2_n, offset=0x88)

Field	Symbol	Direction	Description	Default
[23]	SDIOINTFEN	R/W	SDIO Card Interrupt Sample Filter Enable 1=enable	0
[22:20]	SDIOINTSTYP	R/W	SDIO Card Interrupt Sample Type 000 = Low level 001 = High level, 01x = Falling edge triggered, 10x = Rising edge triggered, 11x = Both edge triggered	0
[19:16]	SDIOINTFCNT	R/W	SDIO Card Interrupt Sample Filter Count Select. **Note**: Internal sample clock is hclk and filter counter is equal to 4* **SDIOINTFCNT.**	0
[15]	SDCDFEN	R/W	Card Detect signal sample filter enable 1=enable	0
[14]	SDCDSTYP	R/W	Card Detect signal sample level select. 1=low level, 0=high level.	0
[13:0]	SDCDFCNT	R/W	Card Detect signal sample filter count select. **Note**: Internal sample clock is hclk and filter counter is equal to 4* **SDCDFCNT.**	0

Register 14-31: DVB TS Control Register (DVBTSCTRL_n, offset=0x90)

Field	Symbol	Direction	Description	Default
[6]	RVSD	R/W	Reserved, always set to 0	0
[5]	DVBTSINVCLK	R/W	DVB-TS receiver input clock invert select. 1=invert input clock.	0
[4]	ENDVBTSINT	R/W	DVB-TS receiver error signal enable, enable this bit will generate DVB-TS receiver error interrupt. 1=enable, 0=masked. **Note**: DVB-TS receiver error is generated when internal buffer overflow occurs.	0
[3]	ENDVBTSSTA	R/W	DVB-TS receiver error status enable	0
[2]	DVBTSMODE	R/W	DVB-TS receiver sample mode. 0=async mode, 1=sync mode	0
[1]	DVBTSENABLE	R/W	DVB-TS receiver enable.	0

			1=enable, DVB-TS receiver 0=disable, normal SD/SDIO controller	
[0]	DVBTSMST	R/W	DVB-TS sample start. 1=receiver sample start. 0=receiver sample stop.	0

Register 14-32: DVB TS Status Register (DVBTSSTA_n, offset=0x94)

Field	Symbol	Direction	Description	Default
[0]	DVBTSERR	RW1C	DVB-TS receiver overflow status. Write 1 to clear this bit.	0

Register 14-33: Slot Interrupt Status Register (SLTINTSTA_n, offset=0xFC)

Field	Symbol	Direction	Description	Default
[7:0]	SLTINTSTA	ROC	Interrupt Signal For Each Slot These status bits indicate the logical OR of Interrupt Signal and Wakeup Signal for each slot. A maximum of 8 slots can be defined. If one interrupt signal is associated with multiple slots, the Host Driver can know which interrupt is generated by reading these status bits. By a power on reset or by setting Software Reset For All, the interrupt signal shall be de-asserted and this status shall read 00h Bit[0]=Slot1 Bit[1]=Slot2 Bit[2]=Slot3 Bit[3]=Slot4 Bit[4]=Slot5 Bit[5]=Slot6 Bit[6]=Slot7 Bit[7]=Slot8	0

Register 14-34: Host Controller Version Register (HCVER_n, offset=0xFE)

Field	Symbol	Direction	Description	Default
[15:8]	VENVER	HWInit	Vendor Version Number This status is Reserved for the vendor version number. The Host Driver should not use this status.	0x10
[7:0]	SPECVER	HWInit	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

15 Ethernet MAC Controller

15.1 Overview

The Ethernet MAC 10/100 Universal enables a host to transmit and receive data over Ethernet in compliance with the IEEE 802.3-2002 standard. Ethernet MAC supports Media Independent Interface (MII) defined in the IEEE 802.3 specifications. Ethernet MAC support AXI interface with master DMA.

Features

- **MAC Core Features**
  - Supports 10/100-Mbps data transfer rates with IEEE 802.3-compliant MII PHY interfaces to communicate with an external Ethernet PHY
  - Supports both full-duplex and half-duplex operation
  - Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
  - Automatic CRC and pad generation controllable on a per-frame basis
  - Options for Automatic Pad/CRC Stripping on receive frames
  - Programmable frame length to support Standard or Jumbo Ethernet frames with sizes up to 16 KB
  - Programmable InterFrameGap (40-96 bit times in steps of 8)
  - Supports a variety of flexible address filtering modes
  - Separate 32-bit status returned for transmission and reception packets
  - Supports IEEE 802.1Q VLAN tag detection for reception frames
- **DMA Block Features**
  - 64bit data transfers
  - Single-channel Transmit and Receive engines
  - Fully synchronous design operating on a single system clock (except for CSR module, when a separate CSR clock is configured)
  - Optimization for packet-oriented DMA transfers with frame delimiters
  - Byte-aligned addressing for data buffer support
  - Dual-buffer (ring) or linked-list (chained) descriptor chaining
  - Descriptor architecture, allowing large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 8 KB of data
  - Comprehensive status reporting for normal operation and transfers with errors
  - Individual programmable burst size for Transmit and Receive DMA Engines for optimal host bus utilization
  - Programmable interrupt options for different operational conditions
  - Per-frame Transmit/Receive complete interrupt control
  - Round-robin or fixed-priority arbitration between Receive and Transmit engines
  - Start/Stop modes
- **Transaction Layer (MTL) Features**
  - Single-channel Transmit and Receive engines
  - Data transfers executed using simple FIFO-protocol
  - Synchronization for all clocks in the design (Transmit, Receive and system clocks)
  - Optimization for packet-oriented transfers with frame delimiters

- Two 2-port RAM-based asynchronous FIFOs with synchronous/asynchronous Read and Write operation with respect to the Read and Write clocks (one for transmission and one for reception)
- Supports 2KB receive FIFO depths on reception.
- Programmable burst-length support for starting a burst up to half the size of the MTL Rx and Tx FIFO in the MAC-MTL configuration
- Configurable Receive FIFO threshold in Cut-Through mode
- Supports 2KB FIFO depth on transmission
- Supports Store and Forward mechanism for transmission to the MAC core
- Supports threshold control for transmit buffer management
- Supports configurable number of frames to be stored in FIFO at any time. The default is 2 frames (fixed) with internal DMA, and up to 8 frames in MAC-MTL configuration.
- Automatic generation of PAUSE frame control or backpressure signal to the MAC core based on Receive FIFO-fill (threshold configurable) level.
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- Software control to flush Tx FIFO
- Data FIFO RAM chip-select disabled when inactive, to reduce power consumption

Block Diagram

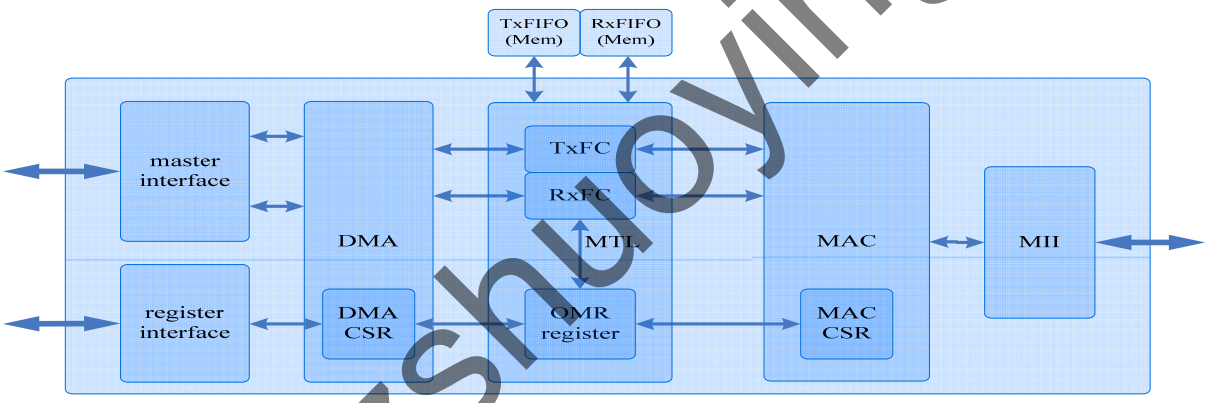


Figure 15-1: Ethernet MAC block diagram

Figure 15-1 shows the functional block diagram of ETH. The mainly module include BUS interface, DMA, Transaction Layer and MAC core. Ether MAC core need a external PHY chip. Master interface is used to transfer data by DMA module. Register interface is slave BUS interface to be used to configured Ether MAC internal register by system. The DMA has independent Transmit and Receive engines to transfer the data between the system memory and transaction layer. The MAC Transaction Layer provides FIFO memory to buffer and regulate the frames between the application system memory and the MAC core.

15.2 Functional Description

15.2.1 DMA Controller

The DMA has independent Transmit and Receive engines, and a CSR space. The Transmit Engine transfers data from system memory to the device port (MTL), while the Receive Engine transfers data from the device port to system memory. The controller utilizes descriptors to efficiently move data from source to destination with minimal Host CPU intervention. The DMA is designed for packet-oriented data transfers such as frames in Ethernet. The controller can be programmed to interrupt the Host CPU for situations such as Frame Transmit and Receive transfer completion, and other normal/error conditions. The DMA and the Host driver communicate through two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into RDLAR and TDLAR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1 [24] and TDES1 [24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

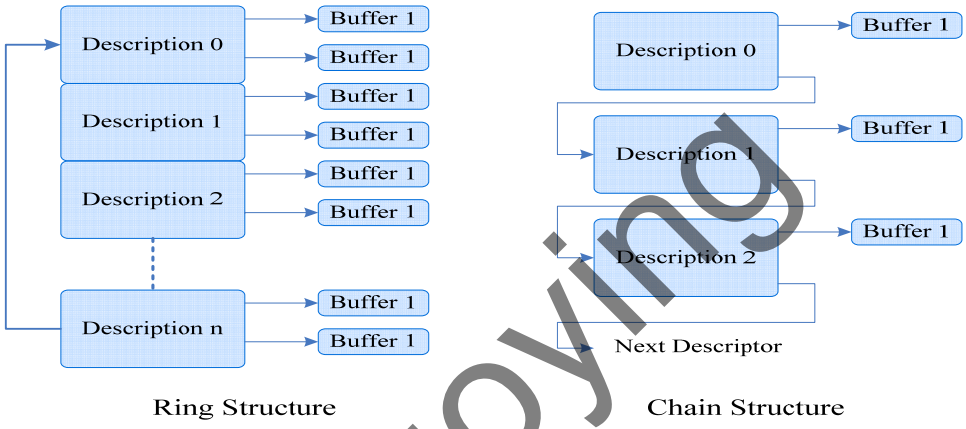


Figure 15-2: Ethernet MAC Descriptor Structure

A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled.

Initialization

Initialization of ETH is ready to transfer data. It is mainly to set the basal registers. It is necessary to set the Host BUS access parameters, interrupt registers, transmit and receive descriptor lists, filtering registers and so on.

The Transmit DMA will initiate a data transfer only when sufficient space to accommodate the configured burst is available in MTL Transmit FIFO or the number of bytes till the end of frame (when it is less than the configured burst length). The DMA will indicate the start address and the number of transfers required to the AXI Master Interface. When the AXI Interface is configured for fixed-length burst, then it will transfer data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise (no fixed-length burst), it will transfer data using INCR (Reserved length) and SINGLE transactions.

The Receive DMA will initiate a data transfer only when sufficient data to accommodate the configured burst is available in MTL Receive FIFO or when the end of frame (when it is less than the configured burst length) is detected in the Receive FIFO. The DMA will indicate the start address and the number of transfers required to the AXI Master Interface. When the AXI Interface is configured for fixed-length burst, then it will transfer data using the best combination of INCR4/8/16 and SINGLE transactions. If the end-of frame is reached before the fixed-burst ends on the AXI interface, then dummy transfers are performed in-order to complete the fixed-burst. Otherwise (FB bit of BMR is reset), it will transfer data using INCR (Reserved length) and SINGLE transactions.

When the AXI interface is configured for address-aligned beats, both DMA engines ensure that the first burst transfer the AXI initiates is less than or equal to the size of the configured PBL. Thus, all subsequent beats start at an address that is aligned to the configured PBL. The DMA can only align the address for beats up to size 16 (for PBL > 16), because the AHB/AXI interface does not support more than INCR16.

Transmission

While in the Run state, the transmit process can simultaneously acquire two frames without closing the Status descriptor of the first (if the OSF bit is set in OPR [2]). As the transmit process finishes transferring the first frame, it immediately polls the Transmit Descriptor list for the second frame. If the second frame is valid, the transmit process transfers this frame before writing the first frames status information.

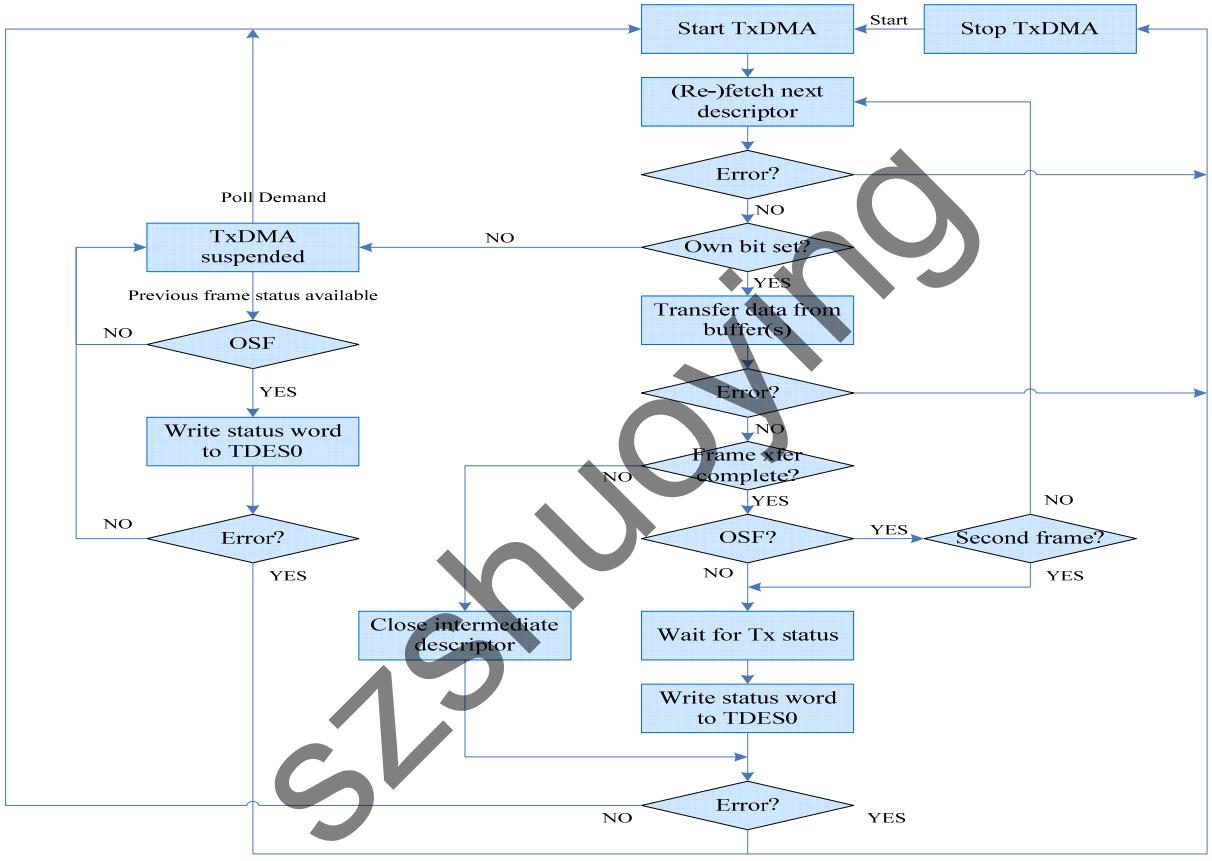


Figure 15-3: TxDMA Operation Mode Flowchart

The Transmit DMA expects that the data buffers contain complete Ethernet frames, excluding preamble, pad bytes, and FCS fields. The DA, SA, and Type/Len fields contain valid data. If the Transmit Descriptor indicates that the MAC core must disable CRC or PAD insertion, the buffer must have complete Ethernet frames (excluding preamble), including the CRC bytes.

Frames can be data-chained and can span several buffers. Frames must be delimited by the First Descriptor (TDES1 [29]) and the Last Descriptor (TDES1 [30]), respectively.

As transmission starts, the First Descriptor must have (TDES1 [29]) set. When this occurs, frame data transfers from the Host buffer to the MTL Transmit FIFO. Concurrently, if the current frame has the Last Descriptor (TDES1 [30]) clear, the Transmit Process attempts to acquire the Next Descriptor. The Transmit Process expects this descriptor to have TDES1 [29]

clear. If TDES1 [30] is clear, it indicates an intermediary buffer. If TDES1 [30] is set, it indicates the last buffer of the frame.

After the last buffer of the frame has been transmitted, the DMA writes back the final status information to the Transmit Descriptor 0 (TDES0) word of the descriptor that has the last segment set in Transmit Descriptor 1 (TDES1 [30]). At this time, if Interrupt on Completion (TDES1 [31]) was set, Transmit Interrupt (SR [0]) is set, the Next Descriptor is fetched, and the process repeats.

Actual frame transmission begins after the MTL Transmit FIFO has reached either a programmable transmit threshold (OMR [16:14]), or a full frame is contained in the FIFO. There is also an option for Store and Forward Mode (OMR [21]). Descriptors are released (Own bit TDES0 [31] clears) when the DMA finishes transferring the frame.

Transmit polling can be suspended by either of the following conditions:

- The DMA detects a descriptor owned by the Host (TDES0 [31] =0). To resume, the driver must give descriptor ownership to the DMA and then issue a Poll Demand command.
- A frame transmission is aborted when a transmit error due to underflow is detected. The appropriate Transmit Descriptor 0 (TDES0) bit is set.

If the second condition occurs, both Abnormal Interrupt Summary (SR [15]) and Transmit Underflow bits (SR [5]) are set, and the information is written to Transmit Descriptor 0, causing the suspension. If the DMA goes into SUSPEND state due to the first condition, and then both Normal Interrupt Summary (SR [16]) and Transmit Buffer Unavailable (SR [2]) are set. In both cases, the position in the Transmit List is retained. The retained position is that of the descriptor following the Last Descriptor closed by the DMA. The driver must explicitly issue a Transmit Poll Demand command after rectifying the suspension cause.

Reception

Reception has a similar process and just one mode.

The DMA does not acknowledge accepting the status from the MTL until it has completed the time stamp write-back and is ready to perform status write-back to the descriptor. If software has enabled time stamping through CSR, when a valid time stamp value is not available for the frame (for example, because receive FIFO was full before the time stamp could be written to it), the DMA writes all-ones to RDES2 and RDES3. Otherwise (that is, if time stamping is not enabled), the RDES2 and RDES3 remain unchanged.

The Receive Engine always attempts to acquire an extra descriptor in anticipation of an incoming frame. Descriptor acquisition is attempted if any of the following conditions is satisfied:

- The receive Start/Stop bit (Register 6[1]) has been set immediately after being placed in the Run state.
- The data buffer of current descriptor is full before the frame ends for the current transfer.
- The controller has completed frame reception, but the current Receive Descriptor is not yet closed.
- The receive process has been suspended because of a host-owned buffer (RDES0 [31] = 0) and a new frame is received.
- A Receive poll demand has been issued.

The ETH transfers the received frames to the Host memory only when the frame passes the address filter and frame size is greater than or equal to configurable threshold bytes set for the Receive FIFO of MTL, or when the complete frame is written to the FIFO in Store-and-Forward mode.

If the frame fails the address filtering, it is dropped in the ETH block itself (unless Receive All MCR [31] bit is set). Frames that are shorter than 64 bytes, because of collision or premature termination, can be purged from the MTL Receive FIFO.

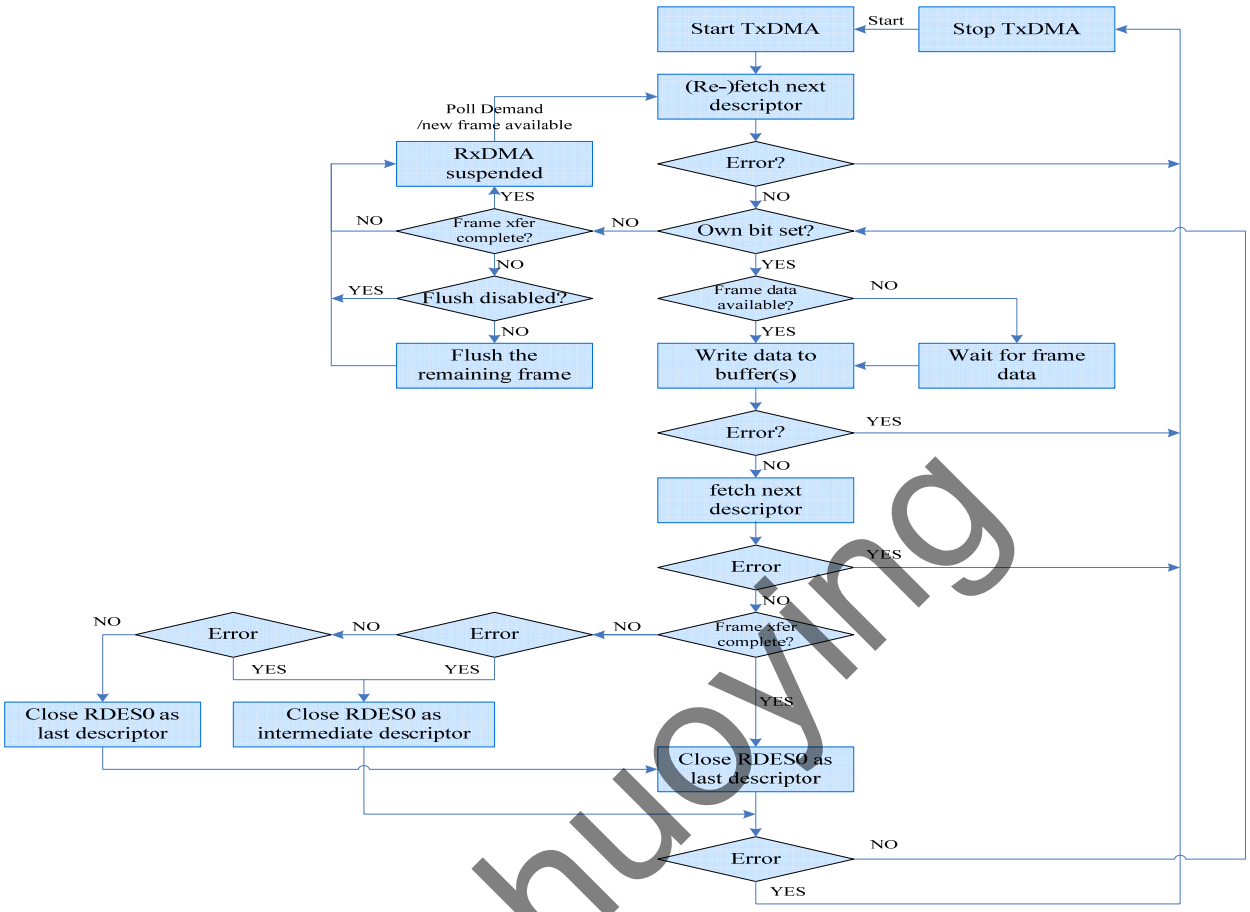


Figure 15-4: RxDMA Operation Mode Flowchart

After 64 (configurable threshold) bytes have been received, the MTL block requests the DMA block to begin transferring the frame data to the Receive Buffer pointed to by the current descriptor. The DMA sets First Descriptor (RDES0 [9]) after the DMA Host Interface becomes ready to receive a data transfer (if DMA is not fetching transmit data from the host), to delimit the frame. The descriptors are released when the Own (RDES [31]) bit is reset to 1'b0, either as the Data buffer fills up or as the last segment of the frame is transferred to the Receive buffer. If the frame is contained in a single descriptor, both Last Descriptor (RDES [8]) and First Descriptor (RDES [9]) are set.

The DMA fetches the next descriptor, sets the Last Descriptor (RDES [8]) bit, and releases the RDES0 status bits in the previous frame descriptor. Then the DMA sets Receive Interrupt (SR [6]). The same process repeats unless the DMA encounters a descriptor flagged as being owned by the host. If this occurs, the Receive Process sets Receive Buffer Unavailable (SR [7]) and then enters the Suspend state. The position in the receive list is retained.

If a new Receive frame arrives while the Receive Process is in Suspend state, the DMA re-fetches the current descriptor in the Host memory. If the descriptor is now owned by the DMA, the Receive Process re-enters the Run state and starts frame reception. If the descriptor is still owned by the host, by default, the DMA discards the current frame at the top of the MTL Rx FIFO and increments the missed frame counter. If more than one frame is stored in the MTL Rx FIFO, the process repeats. The discarding or flushing of the frame at the top of the MTL Rx FIFO can be avoided by setting Operation Mode register bit 24 (DFF). In such conditions, the receive process sets the Receive Buffer Unavailable status and returns to the Suspend state.

Interrupts

Interrupts can be generated as a result of various events. SR contains all the bits that might cause an interrupt. IER contains an enable bit for each of the events that can cause an interrupt. There are two groups of interrupts Normal and Abnormal, as described in SR. Interrupts are cleared by writing 1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal sbd_intr_o is deasserted. If the MAC core is the cause for assertion of the interrupt, then any of the GLI, GMI, or GPI bits of SR will be set high.

Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt (SR [6]) indicates that one or more frames were transferred to the Host buffer. The driver must scan all descriptors, from the last recorded position to the first one owned by the DMA.

An interrupt is generated only once for simultaneous, multiple events. The driver must scan SR for the cause of the interrupt. The interrupt is not generated again unless a new interrupting event occurs, after the driver has cleared the appropriate bit in SR. For example, the controller generates a Receive interrupt (SR [6]) and the driver begins reading SR. Next, Receive Buffer Unavailable (SR [7]) occurs. The driver clears the Receive interrupt. Even then, the sbd_intr_o signal is not de-asserted, due to the active or pending Receive Buffer Unavailable interrupt.

An interrupt timer is given for flexible control of Receive Interrupt (SR [6]). When this Interrupt timer is programmed with a non-zero value, it will get activated as soon as the RxDMA completes a transfer of a received frame to system memory without asserting the Receive Interrupt because it is not enabled in the corresponding Receive Descriptor. When this timer runs out as per the programmed value, RI bit is set and the interrupt is asserted if the corresponding RI is enabled in IER. This timer gets disabled before it runs out, when a frame is transferred to memory and the RI is set because it is enabled for that descriptor.

15.2.2 MAC Transaction Layer (MTL)

Transmit Path

In ETH, the DMA controls all transactions for the transmit path through the ATI. Ethernet frames read from the system memory are pushed into the FIFO by the DMA. The frame is then popped out and transferred to the MAC core when triggered. When the end-of-frame is transferred, the status of the transmission is taken from the MAC core and transferred back to the DMA.

The Transmit FIFO has a default depth of 2K bytes. FIFO-fill level is indicated to the DMA so that it can initiate a data fetch in required bursts from the system memory, using the AXI interface. The data from the AXI Master interface is pushed into the FIFO with the appropriate byte lanes qualified by the DMA. The DMA also indicates the start-of-frame (SOF) and end-of-frame (EOF) transfers along with a few sideband signals controlling the pad-insertion/CRC generation for that frame in the MAC core.

Per-frame control bits, such as Automatic Pad/CRC Stripping disable and so forth are taken as sideband control inputs on the ATI, stored in a separate register FIFO, and passed on to the core transmitter when the corresponding frame data is read from the Transmit FIFO.

There are two modes of operation for popping data towards the MAC core. In Threshold mode, as soon as the number of bytes in the FIFO crosses the configured threshold level (or when the end-of-frame is written before the threshold is crossed) the data is ready to be popped out and forwarded to the MAC core. The threshold level is configured using the TTC bits of DMA Register 0.

In Store-and-Forward mode, only after a complete frame is stored in the FIFO, the MTL pops the frame towards the MAC core. If the Tx FIFO size is smaller than the Ethernet frame to be transmitted (such as Jumbo frame), then the MTL pops the

frame towards the MAC core when the Tx FIFO becomes almost full or when the ATI watermark becomes low. The watermark becomes low when the requested FIFO does not have space to accommodate the requested burst-length on the ATI. Therefore, the MTL never stalls in Store and Forward mode even if the Ethernet frame length is bigger than the Tx FIFO depth.

The application can flush the Transmit FIFO of all contents by setting the FTF (OMR [20]) bit. This bit is self-clearing and initializes the FIFO pointers to the default state. If the FTF bit is set during a frame transfer from the MTL to the MAC core, then the MTL stops further transfer as the FIFO is considered to be empty. Hence an underflow event occurs at the MAC transmitter and the corresponding Status word is forwarded to the DMA.

Receive Path

This module receives the frames given out by the MAC core and pushes them into the Rx FIFO. The status (fill level) of this FIFO is indicated to the DMA once it crosses the configured Receive threshold (RTC of OMR). The MTL also indicates the FIFO fill level so that the DMA can initiate pre-configured burst transfers towards the AXI interface.

During an Rx operation, the MTL is slave to the MAC. When the MAC receives a frame, it pushes in data along with byte enables. The MAC also indicates the SOF and EOF. The MTL accepts the data and pushes it into the Rx FIFO. After the EOF is transferred, the MAC drives the status word, which is also pushed into the same Rx FIFO by the MTL.

The MTL_RX engine takes the data out of the FIFO and sends it to the DMA. In the default Cut- Through mode, when 64 bytes or a full packet of data are received into the FIFO, the MTL_RX engine pops out the data and indicates its availability to the DMA. Once the DMA initiates the transfer to the AXI interface, the MTL_RX engine continues to transfer data from the FIFO until a complete packet has been transferred. Upon completion of the EOF frame transfer, the MTL pops out the status word and send it to the DMA controller.

In Rx FIFO Store-and-Forward mode, a frame is read out only after being written completely into the Receive FIFO. In this mode, all error frames are dropped (if the core is configured to do so) such that only valid frames are read out and forwarded to the application. In Cut-Through mode, some error frames are not dropped, because the error status is received at the end-of-frame, by which time the start of that frame has already been read out of the FIFO.

15.2.3 MAC Core

Transmission

Transmission is initiated when the MTL Application pushes in data with the SOF (mti_sof_i) signal asserted. When the SOF signal is detected, the MAC accepts the data and begins transmitting to the MII. The time required to transmit the frame data to the MII after the Application initiates transmission is variable, depending on delay factors like IFG delay, time to transmit preamble/SFD, and any back-off delays for Half-Duplex mode. Until then, the MAC does not accept the data received from MTL by deasserting the mti_rdy_o signal. After the EOF is transferred to the MAC Core, the core complete normal transmission and then gives the Status of Transmission back to the MTL. If a normal collision (in Half-duplex mode) occurs during transmission, the MAC core makes valid the Transmit Status to the MTL. It then accepts and drops all further data until the next SOF is received. The MTL block should retransmit the same frame from SOF on observing a Retry request (in the Status) from the MAC. The MAC issues an underflow status if the MTL is not able to provide the data continuously during the transmission. During the normal transfer of a frame from MTL, if the MAC receives a SOF without getting an EOF for the previous frame, then it (the SOF) is ignored and the new frame is considered as continuation of the previous frame.

- Transmit Bus Interface Module (TBU)
- Transmit Frame Controller Module (TFC)
- Transmit Protocol Engine Module (TPE)

- Transmit Scheduler Module (STX)
- Transmit CRC Generator Module (CTX)
- Transmit Flow Control Module (FTX)

Reception

A receive operation is initiated when the MAC detects an SFD on the GMII/MII. The core strips the preamble and SFD before proceeding to process the frame. The header fields are checked for the filtering and the FCS field used to verify the CRC for the frame. The received frame is stored in a shallow buffer until the address filtering is performed. The frame is dropped in the core if it fails the address filter. The following are the functional blocks in the Receive path of the MAC core.

- Receive Protocol Engine Module (RPE)
- Receive CRC Module (CRX)
- Receive Frame Controller Module (RFC)
- Receive Flow Control Module (FRX)
- Receive IP Checksum checker (IPC)
- Receive Bus Interface Unit Module (RBU)
- Address Filtering Module (AFM)

15.2.4 Descriptors

The DMA in the Ethernet subsystem transfers data based on a linked list of descriptors. Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes.

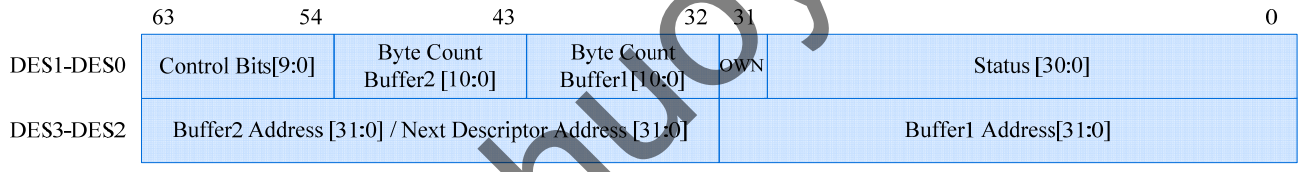


Figure 15-5: Rx/Tx Descriptors

For ring structure, DES3 is Buffer 2 address. For chain structure, DES3 is Next Descriptor address.

Receive Descriptor

The MAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMA (in the MAC Subsystem) always attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received. In a single descriptor (receive) system, the subsystem will generate a descriptor error if receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Table 15-1: Receive Descriptor 0

Field	Symbol	Description
[31]	OWN	**Own Bit** When set, this bit indicates that the descriptor is owned by the DMA of the MAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.

[30]	AFM	**Destination Address Filter Fail** When set, this bit indicates a frame that failed in the DA Filter in the MAC Core.
[29:16]	FL	**Frame Length** These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0 [8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0 [8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
[15]	ES	**Error Summary** Indicates the logical OR of the following bits: ● RDES0[0]: Payload Checksum Error ● RDES0[1]: CRC Error ● RDES0[3]: Receive Error ● RDES0[4]: Watchdog Timeout ● RDES0[6]: Late Collision ● RDES0[7]: IPC Checksum (Type 2) / Giant Frame ● RDES0[11]: Overflow Error ● RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0 [8]) is set.
[14]	DE	**Descriptor Error** When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0 [8]) is set.
[13]	SAF	**Source Address Filter Fail** When set, this bit indicates that the SA field of frame failed the SA Filter in the MAC Core.
[12]	LE	**Length Error** When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0 [5]) bit is reset. Length error status is not valid when CRC error is present.
[11]	OE	**Overflow Error** When set, this bit indicates that the received frame was damaged due to buffer overflow in MTL.
[10]	VLAN	**VLAN Tag** When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the MAC Core.
[9]	FS	**First Descriptor** When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
[8]	LS	**Last Descriptor** When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame
[7]	IPCCE	**IPC Checksum Error/Giant Frame** When IP Checksum Engine (Type 1) is enabled, this bit, when set, indicates that the 16-bit IPv4 Header checksum calculated by the core did not match

		the received checksum bytes. The Error Summary bit [15] is NOT set when this bit is set in this mode. If this bit is set when Full Checksum Offload Engine (Type 2) is enabled, it indicates an error in the IPv4 or IPv6 header. This error can be due to inconsistent Ethernet Type field and IP header Version field values, a header checksum mismatch in IPv4, or an Ethernet frame lacking the expected number of IP header bytes. If you do not select IP Checksum Module during core configuration, this bit, when set, indicates that the received frame was a Giant Frame. Giant frames are larger-than-1,518-byte (or 1,522-byte for VLAN) normal frames and larger-than-9,018-byte (9,022-byte for VLAN) frame when Jumbo Frame processing is enabled.
[6]	LC	**Late Collision** When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.
[5]	FT	**Frame Type** When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
[4]	RWT	**Receive Watchdog Timeout** When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
[3]	RE	**Receive Error** When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension or error (rxd $\neq$ 0f) during extension.
[2]	DE	**Dribble Bit Error** When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.
[1]	CE	**CRC Error** When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0 [8]) is set.
[0]	RMAE	**Rx MAC Address/Payload Checksum Error** When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.

Table 15-2: Receive Descriptor 1

Field	Symbol	Description
[31]	DIC	**Disable Interrupt on Completion** When set, this bit will prevent the setting of the RI (CSR5 [6]) bit of the Status Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
[30:26]	Reserved	Reserved

[25]	RER	**Receive End of Ring** When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
[24]	RCH	**Second Address Chained** When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1 [24] is set, RBS2 (RDES1 [21-11]) is a "don't care" value. RDES1 [25] takes precedence over RDES1 [24].
[23:22]	Reserved	Reserved
[21:11]	RBS2	**Receive Buffer 2 Size** These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 4/8/16 depending upon the bus width, even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 4/8/16, the resulting behavior is Reserved. This field is not valid if RDES1 [24] is set.
[10:0]	RBS1	**Receive Buffer 1 Size** Indicates the first data buffer size in bytes. The buffer size must be a multiple of 4/8/16 depending upon the bus width, even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 4/8/16, the resulting behavior is Reserved. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Table 15-3: Receive Descriptor 2

Field	Symbol	Description
[31:0]	BAP1	**Buffer 1 Address Pointer** These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2 [2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2 [2:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

Table 15-4: Receive Descriptor 3

Field	Symbol	Description
[31:0]	BAP2	**Buffer 2 Address Pointer (Next Descriptor Address)** These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1 [24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1 [24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned. However, when RDES1 [24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3 [2:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

Transmit Descriptor

Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.

Table 15-5: Transmit Descriptor 0

Field	Symbol	Description
[31]	OWN	**Own Bit** When set, this bit indicates that the descriptor is owned by the DMA. When this bit is

		reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
[30:16]	Reserved	Reserved
[15]	ES	**Error Summary** Indicates the logical OR of the following bits: ● TDES0[14]: Jabber Timeout ● TDES0[13]: Frame Flush ● TDES0[11]: Loss of Carrier ● TDES0[10]: No Carrier ● TDES0[9]: Late Collision ● TDES0[8]: Excessive Collision ● TDES0[2]: Excessive Deferral ● TDES0[1]: Underflow Error
[14]	JT	**Jabber Timeout** When set, this bit indicates the MAC transmitter has experienced a jabber time-out. This bit is only set when the MAC configuration register's JD bit is not set.
[13]	FF	**Frame Flushed** When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
[12]	Reserved	Reserved
[11]	LC	**Loss of Carrier** When set, this bit indicates that Loss of Carrier occurred during frame transmission (that is, the gmii_crs_i signal was inactive for one or more transmit clock periods during frame transmission). This is valid only for the frames transmitted without collision and when the MAC operates in Half-Duplex Mode.
[10]	NC	**No Carrier** When set, this bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
[9]	LC	**Late Collision** When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window. Not valid if Underflow Error is set.
[8]	EC	**Excessive Collision** When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable Retry) bit in the MAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
[7]	VF	**VLAN Frame** When set, this bit indicates that the transmitted frame was a VLAN-type frame.
[6:3]	CC	Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0 [8]) is set.
[2]	ED	**Excessive Deferral** When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times if the Deferral Check (DC) bit is set high in the MAC Control Register.
[1]	UF	**Underflow Error** When set, this bit indicates that the MAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (SR [5]) and Transmit Interrupt (SR [0]).
[0]	DB	**Deferred Bit** When set, this bit indicates that the MAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

Table 15-6: Transmit Descriptor 1

Field	Symbol	Description
[31]	IC	**Interrupt on Completion** When set, this bit sets Transmit Interrupt (SR [0]) after the present frame has been transmitted.
[30]	LS	**Last Segment** When set, this bit indicates that the buffer contains the last segment of the frame.
[29]	FS	**First Segment** When set, this bit indicates that the buffer contains the first segment of a frame.
[28:27]	CIC	**Checksum Insertion Control** These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. ● 00: Do nothing. Checksum Engine is bypassed ● 01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. ● 10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. ● 11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.
[26]	DC	**Disable CRC** When set, the MAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).
[25]	TER	**Transmit End of Ring** When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
[24]	TCH	**Second Address Chained** When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1 [24] is set, TBS2 (TDES1 [21–11]) are "don't care" values. TDES1 [25] takes precedence over TDES1 [24].
[23]	DP	**Disable Padding** When set, the MAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1 [29]) is set.
[22]	TTSE	**Transmit Time Stamp Enable** When set, this bit enables IEEE1588 hardware time stamping for the transmit frame referenced by the descriptor. This field is valid only when the First Segment control bit (TDES1 [29]) is set.
[21:11]	TBS2	**Transmit Buffer 2 Size** These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1 [24] is set.
[10:0]	TBS1	**Transmit Buffer 1 Size** These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Table 15-7: Transmit Descriptor 2

Field	Symbol	Description
[31:0]	BAP1	**Buffer 1 Address Pointer** These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Table 15-8: Transmit Descriptor 3

Field	Symbol	Description
[31:0]	BAP2	**Buffer 2 Address Pointer (Next Descriptor Address)** Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1 [24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1 [24] is set. (LSBs are ignored internally.)

15.3 Ethernet MAC Register Description

Configuration register fields are assigned to one of the attributes described below:

Register Attribute	Description
R	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 registers: 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.
RC	Register field can be read by the application (Read), can be set to 1'b1 by the core on a certain internal event (Self Set), and is automatically cleared to 1'b0 on a register read. A register write of 1'b0 has no effect on this field. The conditions under which the core sets this field are explained in detail in the field's description.

15.3.1 Ethernet MAC Register Memory Map

ETH's register mapping address range in system is 0x20EC0000~0x20EFFFFF. ETH register's system accessing address equal to BASE_ADDRESS (0x20EC0000) plus address offset.

Table 15-9: Ethernet MAC Register Memory Map

Address	Symbol	Direction	Description
0x0000	MCR	RW	This is the operation mode register for the MAC.
0x0004	MFF	RW	Contains the frame filtering controls.
0x0008	HTHR	RW	Contains the higher 32 bits of the Multicast Hash table.
0x000c	HTLR	RW	Contains the lower 32 bits of the Multicast Hash table.
0x0010	Reserved	N/A	Reserved
0x0014	Reserved	N/A	Reserved
0x0018	FCR	RW	Controls the generation of control frames.
0x001c	VTR	RW	Identifies IEEE 802.1Q VLAN type frames.
0x0020	Reserved	N/A	Reserved
0x0024	DBGR	R	Gives the status of various internal blocks for debugging
0x0028	Reserved	N/A	Reserved

0x002c	Reserved	N/A	Reserved
0x0030	Reserved	N/A	Reserved
0x0034	Reserved	N/A	Reserved
0x0038	Reserved	N/A	Reserved
0x003c	Reserved	N/A	Reserved
0x0040	MAHR0	RW	Contains the higher 16 bits of the MAC address 0.
0x0044	MALR0	RW	Contains the lower 32 bits of the MAC address 0.
0x0048	MAHR1	RW	Contains the higher 16 bits of the MAC address 1.
0x004c	MALR1	RW	Contains the lower 32 bits of the MAC address 1.
0x0050	MAHR2	RW	Contains the higher 16 bits of the MAC address 2.
0x0054	MALR2	RW	Contains the lower 32 bits of the MAC address 2.
0x0058	MAHR3	RW	Contains the higher 16 bits of the MAC address 3.
0x005c	MALR3	RW	Contains the lower 32 bits of the MAC address 3.
0x0060	MAHR4	RW	Contains the higher 16 bits of the MAC address 4.
0x0064	MALR4	RW	Contains the lower 32 bits of the MAC address 4.
0x0068	MAHR5	RW	Contains the higher 16 bits of the MAC address 5.
0x006c	MALR5	RW	Contains the lower 32 bits of the MAC address 5.
0x0070	MAHR6	RW	Contains the higher 16 bits of the MAC address 6.
0x0074	MALR6	RW	Contains the lower 32 bits of the MAC address 6.
0x0078	MAHR7	RW	Contains the higher 16 bits of the MAC address 7 .
0x007c	MALR7	RW	Contains the lower 32 bits of the MAC address 7.
0x0080	MAHR8	RW	Contains the higher 16 bits of the MAC address 8.
0x0084	MALR8	RW	Contains the lower 32 bits of the MAC address 8.
0x0088	MAHR9	RW	Contains the higher 16 bits of the MAC address 9.
0x008c	MALR9	RW	Contains the lower 32 bits of the MAC address 9.
0x0090	MAHR10	RW	Contains the higher 16 bits of the MAC address 10.
0x0094	MALR10	RW	Contains the lower 32 bits of the MAC address 10.
0x0098	MAHR11	RW	Contains the higher 16 bits of the MAC address 11.
0x009c	MALR11	RW	Contains the lower 32 bits of the MAC address 11.
0x00a0	MAHR12	RW	Contains the higher 16 bits of the MAC address 12.
0x00a4	MALR12	RW	Contains the lower 32 bits of the MAC address 12.
0x00a8	MAHR13	RW	Contains the higher 16 bits of the MAC address 13.
0x00ac	MALR13	RW	Contains the lower 32 bits of the MAC address 13.
0x00b0	MAHR14	RW	Contains the higher 16 bits of the MAC address 14.
0x00b4	MALR14	RW	Contains the lower 32 bits of the MAC address 14.
0x00b8	MAHR15	RW	Contains the higher 16 bits of the MAC address 15.
0x00bc	MALR15	RW	Contains the lower 32 bits of the MAC address 15.
0x00c0~0x07fc	Reserved	N/A	Reserved
0x0800	MAHR16	RW	Contains the higher 16 bits of the MAC address 16.
0x0804	MALR16	RW	Contains the lower 32 bits of the MAC address 16.
0x0808	MAHR17	RW	Contains the higher 16 bits of the MAC address 17.
0x080c	MALR17	RW	Contains the lower 32 bits of the MAC address 17.
0x0810	MAHR18	RW	Contains the higher 16 bits of the MAC address 18.
0x0814	MALR18	RW	Contains the lower 32 bits of the MAC address 18.
0x0818	MAHR19	RW	Contains the higher 16 bits of the MAC address 19.
0x081c	MALR19	RW	Contains the lower 32 bits of the MAC address 19.
0x0820	MAHR20	RW	Contains the higher 16 bits of the MAC address 20.
0x0824	MALR20	RW	Contains the lower 32 bits of the MAC address 20.
0x0828	MAHR21	RW	Contains the higher 16 bits of the MAC address 21.
0x082c	MALR21	RW	Contains the lower 32 bits of the MAC address 21.
0x0830	MAHR22	RW	Contains the higher 16 bits of the MAC address 22.
0x0834	MALR22	RW	Contains the lower 32 bits of the MAC address 22.
0x0838	MAHR23	RW	Contains the higher 16 bits of the MAC address 23.
0x083c	MALR23	RW	Contains the lower 32 bits of the MAC address 23.
0x0840	MAHR24	RW	Contains the higher 16 bits of the MAC address 24.
0x0844	MALR24	RW	Contains the lower 32 bits of the MAC address 24.
0x0848	MAHR25	RW	Contains the higher 16 bits of the MAC address 25.

0x084c	MALR25	RW	Contains the lower 32 bits of the MAC address 25.
0x0850	MAHR26	RW	Contains the higher 16 bits of the MAC address 26.
0x0854	MALR26	RW	Contains the lower 32 bits of the MAC address 26.
0x0858	MAHR27	RW	Contains the higher 16 bits of the MAC address 27.
0x085c	MALR27	RW	Contains the lower 32 bits of the MAC address 27.
0x0860	MAHR28	RW	Contains the higher 16 bits of the MAC address 28.
0x0864	MALR28	RW	Contains the lower 32 bits of the MAC address 28.
0x0868	MAHR29	RW	Contains the higher 16 bits of the MAC address 29.
0x086c	MALR29	RW	Contains the lower 32 bits of the MAC address 29.
0x0870	MAHR30	RW	Contains the higher 16 bits of the MAC address 30.
0x0874	MALR30	RW	Contains the lower 32 bits of the MAC address 30.
0x0878	MAHR31	RW	Contains the higher 16 bits of the MAC address 31.
0x087c	MALR31	RW	Contains the lower 32 bits of the MAC address 31.
0x0880~0x0ffc	Reserved	N/A	Reserved
0x1000	BMR	RW	Controls the Host Interface Mode.
0x1004	TPDR	RW	Used by the host to instruct the DMA to poll the Transmit Descriptor List.
0x1008	RPDR	RW	Used by the Host to instruct the DMA to poll the Receive Descriptor list.
0x100c	RDLAR	RW	Points the DMA to the start of the Receive Descriptor list.
0x1010	TDLAR	RW	Points the DMA to the start of the Transmit Descriptor List.
0x1014	SR	RW	The Software driver (application) reads this register during interrupt service routine or polling to determine the status of the DMA.
0x1018	OMR	RW	Establishes the Receive and Transmit operating modes and command.
0x101c	IER	RW	Enables the interrupts reported by the Status Register.
0x1020	MFBOCR	RW	Contains the counters for discarded frames because no host Receive Descriptor was available, and discarded frames because of Receive FIFO Overflow.
0x1024	RIWTR	RW	Watchdog time-out for Receive Interrupt (RI) from DMA
0x1028	ABMR	RW	Controls AXI Master behavior (mainly controls burst splitting and number of outstanding requests).
0x102c	AXISR	RW	Gives the idle status of the AXI master's read/write channels.
0x1030~0x1044	Reserved	N/A	Reserved
0x1048	CHTDR	RW	Points to the start of current Transmit Descriptor read by the DMA.
0x104c	CHRDR	RW	Points to the start of current Receive Descriptor read by the DMA.
0x1050	CHTBAR	RW	Points to the current Transmit Buffer address read by the DMA.
0x1054	CHRBAR	RW	Points to the current Receive Buffer address read by the DMA.

15.3.2 Ethernet MAC Registers and Field Descriptions

Register 15-1: MAC Configuration Register (MCR)

Field	Symbol	Direction	Description	Default
[31:27]	Reserved	N/A	Reserved	0
[26]	Reserved	RW	Reserved	0
[25]	CST	RW	**CRC stripping for Type frames** When set, the last 4 bytes (FCS) of all frames of Ether type (type field greater than 0x0600) will be stripped and dropped before forwarding the frame to the application. This function is not valid when the MAC receiver has IP Checksum Engine (Type 1) enabled.	0
[24]	Reserved	RW	Reserved	0
[23]	WD	RW	**Watchdog Disable** When this bit is set, the MAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the MAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that.	0
[22]	JD	RW	**Jabber Disable** When this bit is set, the MAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.	0

[21]	Reserved	RW	Reserved	0
[20]	JE	RW	**Jumbo Frame Enable** When this bit is set, MAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.	0
[19:17]	IFG	RW	**Inter-Frame Gap** These bits control the minimum IFG between frames during transmission. ● 000: 96 bit times ● 001: 88 bit times ● 010: 80 bit times ● 011: 72 bit times ● 100: 64 bit times ● 101: 56 bit times ● 110: 48 bit times ● 111: 40 bit times Note that in Half-Duplex mode, the minimum IFG can be configured for 64 bit times (IFG = 100) only. Lower values are not considered.	0
[16]	DCRS	RW	**Disable Carrier Sense During Transmission** When set high, this bit makes the MAC transmitter ignore the MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.	0
[15]	Reserved	N/A	Reserved	1
[14]	Reserved	N/A	Reserved	0
[13]	DO	RW	**Disable Receive Own** When this bit is set, the MAC disables the reception of frames when the txen_o is asserted in Half-Duplex mode. When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting.	0
[12]	LM	RW	**Loopback Mode** When this bit is set, the MAC operates in loopback mode at /MII. The MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally.	0
[11]	DM	RW	**Duplex Mode** When this bit is set, the MAC operates in a Full-Duplex mode where it can transmit and receive simultaneously.	0
[10]	Reserved	N/A	Reserved	0
[9]	DR	RW	**Disable Retry** When this bit is set, the MAC will attempt only 1 transmission. When a collision occurs on the MII, the MAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the MAC will attempt retries based on the settings of BL. This bit is applicable only to Half-Duplex mode and is Reserved (RO with default value) in Full- Duplex-only configuration.	0
[8]	Reserved	N/A	Reserved	0
[7]	ACS	RW	**Automatic Pad/CRC Stripping** When this bit is set, the MAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the MAC will pass all incoming frames to the Host unmodified.	0
[6:5]	BL	RW	**Back-Off Limit** The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is Reserved (RO) in Full-Duplex-only configuration. ● 00: k = min (n, 10) ● 01: k = min (n, 8) ● 10: k = min (n, 4)	0

			● 11: k = min (n, 1), Where n = retransmission attempt. The random integer r takes the value in the range 0 ≤ r < 2k	
[4]	DC	RW	**Deferral Check** When this bit is set, the deferral check function is enabled in the MAC. The MAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmit state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Jumbo frame mode is enabled in 10/100-Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in Half-Duplex mode.	0
[3]	TE	RW	**Transmitter Enable** When this bit is set, the transmitting state machine of the MAC is enabled for transmission on the MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.	0
[2]	RE	RW	**Receiver Enable** When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the MII.	0
[1:0]	Reserved	N/A	Reserved	0

Register 15-2: MAC Frame Filter (MFF)

Field	Symbol	Direction	Description	Default
[31]	RA	RW	**Receive All** When this bit is set, the MAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter.	0
[30:11]	Reserved	N/A	Reserved	0
[10]	HPF	RW	**Hash or Perfect Filter** When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter.	0
[9]	SAF	RW	**Source Address Filter** Enable The MAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the MAC drops the frame. When this bit is reset, then the MAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison.	0
[8]	SAIF	RW	**SA Inverse Filtering** When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.	0

[7:6]	PCF	RW	**Pass Control Frames** These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Flow Control Register [2]. ● 00: MAC filters all control frames from reaching the application. ● 01: MAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. ● 10: MAC forwards all control frames to application even if they fail the Address Filter. ● 11: MAC forwards control frames that pass the Address Filter.	0
[5]	DBF	RW	**Disable Broadcast Frames** When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames.	0
[4]	PM	RW	**Pass All Multicast** When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.	0
[3]	DAIF	RW	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.	0
[2]	HMC	RW	**Hash Multicast** When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.	0
[1]	HUC	RW	**Hash Unicast** When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.	0
[0]	PR	RW	**Promiscuous Mode** When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.	0

Register 15-3: Hash Table Low Register (HTLR)

Field	Symbol	Direction	Description	Default
[31:0]	HTL	RW	**Hash Table Low** This field contains the lower 32 bits of Hash table.	0

Register 15-4: Flow Control Register (FCR)

Field	Symbol	Direction	Description	Default
[30:16]	PT	RW	**Pause Time** This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.	0
[15:8]	Reserved	N/A	Reserved	0
[7]	DZPQ	RW	**Disable Zero-Quanta Pause** When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the deassertion of the flow-control signal from the FIFO layer When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.	0

[6]	Reserved	N/A	Reserved	0
[5:4]	PLT	RW	**Pause Low Threshold** This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256 – 28) slot-times after the first PAUSE frame is transmitted. **Selection** **Threshold** 00 Pause time minus 4 slot times 01 Pause time minus 28 slot times 10 Pause time minus 144 slot times 11 Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the MII interface.	0
[3]	UP	RW	**Unicast Pause Frame Detect** When this bit is set, the MAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the MAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.	0
[2]	RFE	RW	**Receive Flow Control Enable** When this bit is set, the MAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled.	0
[1]	TFE	RW	**Transmit Flow Control Enable** In Full-Duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the MAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled.	0
[0]	FCB/BPA	RW	**Flow Control Busy/Backpressure Activate** This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the Flow Control register. To initiate a Pause control frame, the Application must set this bit to 1'b1. During a transfer of the Control Frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the MAC will reset this bit to 1'b0. The Flow Control register should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC Core. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured to Full- Duplex mode, the BPA is automatically disabled.	0

Register 15-5: VLAN Tag Register (VTR)

Field	Symbol	Direction	Description	Default
[30:17]	Reserved	N/A	Reserved	0
[16]	ETV	RW	**Enable 12-Bit VLAN Tag Comparison** When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame.	0

			When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.	
[15:0]	VL	RW	**VLAN Tag Identifier for Receive Frames** This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits [15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits [11:0]) is used for comparison. If VL (VL [11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.	0

Register 15-6: Debug Register (DBGR)

Field	Symbol	Direction	Description	Default
[30:26]	Reserved	N/A	Reserved	0
[25]	FLAG	R	When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission.	0
[24]	FLAG	R	When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission.	0
[23]	Reserved	R	Reserved	0
[22]	FLAG	R	When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO.	0
[21:20]	FLAG	R	This indicates the state of the TxFIFO read Controller: ● 00: IDLE state ● 01: READ state (transferring data to MAC transmitter) ● 10: Waiting for TxStatus from MAC transmitter ● 11: Writing the received TxStatus or flushing the TxFIFO	0
[19]	FLAG	R	When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission	0
[18:17]	FLAG	R	This indicates the state of the MAC Transmit Frame Controller module (section 3.7.1.2): ● 00: IDLE ● 01: Waiting for Status of previous frame or IFG/backoff period to be over ● 10: Generating and transmitting a PAUSE control frame (in full duplex mode) ● 11: Transferring input frame for transmission	0
[16]	FLAG	R	When high, it indicates that the MAC MII transmit protocol engine is actively transmitting data and not in IDLE state.	0
[15:10]	Reserved	N/A	Reserved	0
[9:8]	FLAG	R	This gives the status of the RxFIFO Fill-level: ● 00: RxFIFO Empty ● 01: RxFIFO fill-level below flow-control de-activate threshold ● 10: RxFIFO fill-level above flow-control activate threshold ● 11: RxFIFO Full	0
[7]	Reserved	N/A	Reserved	0
[6:5]	FLAG	R	It gives the state of the RxFIFO read Controller: ● 00: IDLE state ● 01: Reading frame data ● 10: Reading frame status (or time-stamp) ● 11: Flushing the frame data and Status	0
[4]	FLAG	R	When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO.	0
[3]	Reserved	N/A	Reserved	0
[2:1]	FLAG	R	When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module	0
[0]	FLAG	R	When high, it indicates that the MAC MII receive protocol engine is actively	

			receiving data and not in IDLE state.	

Register 15-7: MAC Address0 High Register (MAHR0)

Field	Symbol	Direction	Description	Default
[31]	MO	R	Always 1	0
[30:16]	Reserved	N/A	Reserved	0
[15:0]	A[47:32]	RW	**MAC Address0 [47:32]** This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.	FFFF

Register 15-8: MAC Address0 Low Register (MALR0)

Field	Symbol	Direction	Description	Default
[31:0]	A[31:0]	RW	**MAC Address0 [31:0]** This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.	FFFFF FFF

Register 15-9: MAC Address1 High Register (MAHR1)

Field	Symbol	Direction	Description	Default
[31]	AE	RW	**Address Enable** When this bit is set, the Address filter module uses the second MAC address for perfect filtering. When reset, the address filter module will ignore the address for filtering.	0
[30]	SA	RW	**Source Address** When this bit is set, the MAC Address 1 [47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address 1 [47:0] is used to compare with the DA fields of the received frame.	0
[29:24]	MBC	RW	**Mask Byte Control** These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the MAC core does not compare the corresponding byte of received DA/SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: ● Bit [29]: MAHR1[15:8] ● Bit [28]: MAHR1 [7:0] ● Bit [27]: MALR1 [31:24] ● Bit [26]: MALR1 [23:16] ● Bit [25]: MALR1 [15:8] ● Bit [24]: MALR1 [7:0]	0
[23:16]	Reserved	N/A	Reserved	0
[15:0]	A[47:32]	RW	**MAC Address1 [47:32]** This field contains the upper 16 bits (47:32) of the 6-byte second MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.	FFFF

Register 15-10: MAC Address1 Low Register (MALR1)

Field	Symbol	Direction	Description	Default
[31:0]	A[31:0]	RW	**MAC Address1 [31:0]** This field contains the lower 32 bits of the 6-byte second MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.	FFFFF FFF

MAHR2, MAHR3 … MAHR31 have same description with MAHR1. MALR2, MALR3 … MALR31 have same description with MALR1.

Register 15-11: BUS Mode Register (BMR)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25]	AAL	RW	**Address-Aligned Beats** When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.	
[24]	8xPBL Mode	RW	When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.	0
[23]	USP	RW	**Use Separate PBL** When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines.	0
[22:17]	RPBL	RW	**RxDMA PBL** These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in Reserved behavior. These bits are valid and applicable only when USP is set high.	0
[16]	FB	RW	**Fixed Burst** This bit controls whether the AXI Master interface performs fixed burst transfers or not. When reset, the AXI will use SINGLE and INCR burst transfer operations.	0
[15:14]	PR	RW	**Rx:Tx priority ratio** RxDMA requests given priority over TxDMA requests in the following ratio. This is valid only when the DA bit is reset. ● 00: 1:1 ● 01: 2:1 ● 10: 3:1 ● 11: 4:1	0
[16]	PBL	RW	**Programmable Burst Length** These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in Reserved behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only.	0
[7]	Reserved	N/A	Reserved	1
[6:2]	DSL	RW	**Descriptor Skip Length** This bit specifies the number of Word/Dword/Lword (depending on 32/64/128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.	0
[1]	Reserved	N/A	Reserved	0
[0]	SWR	RW	**Software Reset** When this bit is set, the MAC DMA Controller resets all MAC Subsystem internal registers and logic. It is cleared automatically after the reset	0

			operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that the PHY inputs clocks are present for software reset completion.	

Register 15-12: Transmit Poll Demand Register (TPDR)

Field	Symbol	Direction	Description	Default
[31:0]	TPD	RW	**Transmit Poll Demand** When these bits are written with any value, the DMA reads the current descriptor pointed to by Register 18. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register 5[2] is asserted. If the descriptor is available, transmission resumes.	0

Register 15-13: Receive Poll Demand Register (RPDR)

Field	Symbol	Direction	Description	Default
[31:0]	RPD	RW	**Receive Poll Demand** When these bits are written with any value, the DMA reads the current descriptor pointed to by Register 19. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register 5[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state.	0

Register 15-14: Receive Descriptor List Address Register (RPDR)

Field	Symbol	Direction	Description	Default
[31:0]	SRL	RW	**Start of Receive List** This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.	0

Register 15-15: Transmit Descriptor List Address Register (TPDR)

Field	Symbol	Direction	Description	Default
[31:0]	STL	RW	**Start of Transmit List** This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.	0

Register 15-16: Status Register (SR)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25:23]	EB	R	**Error Bits** These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (SR [13]) set. This field does not generate an interrupt. ● Bit 23 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA ● Bit 24 1'b1 Error during read transfer 1'b0 Error during write transfer ● Bit 25 1'b1 Error during descriptor access 1'b0 Error during data buffer access	0

[22:20]	TS	R	**Transmit Process State** These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. ● 000: Stopped; Reset or Stop Transmit Command issued. ● 001: Running; Fetching Transmit Transfer Descriptor. ● 010: Running; Waiting for status. ● 011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). ● 100: TIME_STAMP write state. ● 101: Reserved for future use. ● 110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. ● 111: Running; Closing Transmit Descriptor.	0
[19:17]	RS	R	**Receive Process State** These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. ● 000: Stopped: Reset or Stop Receive Command issued. ● 001: Running: Fetching Receive Transfer Descriptor. ● 010: Reserved for future use. ● 011: Running: Waiting for receive packet. ● 100: Suspended: Receive Descriptor Unavailable. ● 101: Running: Closing Receive Descriptor. ● 110: TIME_STAMP write state. ● 111: Running: Transferring the receive packet data from receive buffer to host memory.	0
[16]	NIS	RW1C	**Normal Interrupt Summary** Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in IER: ● SR[0]: Transmit Interrupt ● SR[2]: Transmit Buffer Unavailable ● SR[6]: Receive Interrupt ● SR[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.	0
[15]	AIS	RW1C	**Abnormal Interrupt Summary** Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in IER: ● SR[1]:Transmit Process Stopped ● SR[3]:Transmit Jabber Timeout ● SR[4]: Receive FIFO Overflow ● SR[5]: Transmit Underflow ● SR[7]: Receive Buffer Unavailable ● SR[8]: Receive Process Stopped ● SR[9]: Receive Watchdog Timeout ● SR[10]: Early Transmit Interrupt ● SR[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.	0
[14]	ERI	RW1C	**Early Receive Interrupt** This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt SR[6] automatically clears this bit.	0
[13]	FBI	RW1C	**Fatal Bus Error Interrupt** This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses.	0
[12:11]	Reserved	N/A	Reserved	0
[10]	ETI	RW1C	**Early Transmit Interrupt** This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.	0
[9]	RWT	RW1C	**Receive Watchdog Timeout**	0

			This bit is asserted when a frame with a length greater than 2,048 bytes is received (10,240 when Jumbo Frame mode is enabled).	
[8]	RPS	RW1C	**Receive Process Stopped** This bit is asserted when the Receive Process enters the Stopped state.	0
[7]	RU	RW1C	**Receive Buffer Unavailable** This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. SR [7] is set only when the previous Receive Descriptor was owned by the DMA.	0
[6]	RI	RW1C	**Receive Interrupt** This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.	0
[5]	UNF	RW1C	**Transmit Underflow** This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0 [1] is set.	0
[4]	OVF	RW1C	**Receive Overflow** This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0 [11].	0
[3]	TJT	RW1C	**Transmit Jabber Timeout** This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0 [14] flag to assert.	0
[2]	TU	RW1C	**Transmit Buffer Unavailable** This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits [22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command.	0
[1]	TPS	RW1C	**Transmit Process Stopped** This bit is set when the transmission is stopped.	0
[0]	TI	RW1C	**Transmit Interrupt** This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor.	0

Register 15-17: Operation Mode Register (OMR)

Field	Symbol	Direction	Description	Default
[31:26]	Reserved	N/A	Reserved	0
[25]	RSF	RW	**Receive Store and Forward** When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.	0
[24]	DFF	RW	**Disable Flushing of Received Frames** When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.	0
[23:22]	Reserved	N/A	Reserved	0
[21]	TSF	RW	**Transmit Store and Forward** When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in OMR[16:14] are ignored.	0

			This bit should be changed only when transmission is stopped.	
[20]	FTF	RW	**Flush Transmit FIFO** When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.	0
[19:17]	Reserved	N/A	Reserved	0
[16:14]	TTC	RW	**Transmit Threshold Control** These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. ● 000: 64 ● 001: 128 ● 010: 192 ● 011: 256 ● 100: 40 ● 101: 32 ● 110: 24 ● 111: 16	0
[13]	ST	RW	**Start/Stop Transmission Command** When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by TDLAR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (GDR[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting TDLAR, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state.	0
[12:8]	Reserved	N/A	Reserved	0
[7]	FEF	RW	**Forward Error Frames** When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, and overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is wriiten, then such frames are dropped even when FEF is set.	0
[6]	FUF	RW	**Forward Undersized Good Frames** When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01).	0

[5]	Reserved	N/A	Reserved	0
[4:3]	RTC	RW	**Receive Threshold Control** These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. ● 00: 64 ● 01: 32 ● 10: 96 ● 11: 128	0
[2]	OSF	RW	**Operate on Second Frame** When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained.	0
[1]	SR	RW	**Start/Stop Receive** When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by RDLAR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register 5[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting RDLAR, DMA behavior is unpredictable. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.	0
[0]	Reserved	N/A	Reserved	0

Register 15-18: Interrupt Enable Register (SR)

Field	Symbol	Direction	Description	Default
[31:17]	Reserved	N/A	Reserved	0
[16]	NIE	RW	**Normal Interrupt Summary Enable** When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: ● SR[0]: Transmit Interrupt ● SR[2]: Transmit Buffer Unavailable ● SR[6]: Receive Interrupt ● SR[14]: Early Receive Interrupt	0
[15]	AIE	RW	**Abnormal Interrupt Summary Enable** When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits ● SR[1]: Transmit Process Stopped ● SR[3]: Transmit Jabber Timeout ● SR[4]: Receive Overflow ● SR[5]: Transmit Underflow ● SR[7]: Receive Buffer Unavailable ● SR[8]: Receive Process Stopped ● SR[9]: Receive Watchdog Timeout ● SR[10]: Early Transmit Interrupt ● SR[13]: Fatal Bus Error	0
[14]	ERE	RW	**Early Receive Interrupt Enable** When this bit is set with Normal Interrupt Summary Enable (IER [16]),	0

			Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled.	
[13]	FBE	RW	**Fatal Bus Error Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.	0
[12:11]	Reserved	N/A	Reserved	0
[10]	ETE	RW	**Early Transmit Interrupt Enable** When this bit is set with an Abnormal Interrupt Summary Enable (IER [15]), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.	0
[9]	RWE	RW	**Receive Watchdog Timeout Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.	0
[8]	RSE	RW	**Receive Stopped Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.	0
[7]	RUE	RW	**Receive Buffer Unavailable Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.	0
[6]	RIE	RW	**Receive Interrupt Enable** When this bit is set with Normal Interrupt Summary Enable (IER [16]), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.	0
[5]	UNE	RW	**Underflow Interrupt Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.	0
[4]	OVE	RW	**Overflow Interrupt Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled	0
[3]	TJE	RW	**Transmit Jabber Timeout Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.	0
[2]	TUE	RW	**Transmit Buffer Unavailable Enable** When this bit is set with Normal Interrupt Summary Enable (IER [16]), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.	0
[1]	TSE	RW	**Transmit Stopped Enable** When this bit is set with Abnormal Interrupt Summary Enable (IER [15]), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.	0
[0]	TIE	RW	**Transmit Interrupt Enable** When this bit is set with Normal Interrupt Summary Enable (IER [16]), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.	0

Register 15-19: Missed Frame and Buffer Overflow Counter Register (MFBOCR)

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	N/A	Reserved	0
[28]	OVF	RC	Overflow bit for FIFO Overflow Counter	0
[27:17]	NFMA	RC	Indicates the number of frames missed by the application. This counter is incremented each time the MTL asserts the sideband signal	0

			mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.	
[16]	MOVF	RC	Overflow bit for Missed Frame Counter	0
[15:0]	NFMC	RC	Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.	0

Register 15-20: Receive Interrupt Watchdog Timer Register (RIWTR)

Field	Symbol	Direction	Description	Default
[31:8]	Reserved	N/A	Reserved	0
[7:0]	RIWT	RW	**RI Watchdog Timer count** Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame.	0

Register 15-21: AXI Bus Mode Register (ABMR)

Field	Symbol	Direction	Description	Default
[31]	EN_LPI	RW	**Enable LPI (Low Power Interface)** When set to 1, enable the LPI (Low Power Interface) and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller.	0
[30]	UNLCK_ON_MGK_RWK	RW	**Unlock on Magic Packet or Remote Wake Up** When set to 1, enables to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enable to requests to come out of Low Power mode when any frame is received.	0
[29:22]	Reserved	N/A	Reserved	0
[21:20]	WR_OSR_LMT	RW	**AXI Maximum Write Out Standing Request Limit** This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1	1
[19:18]	Reserved	N/A	Reserved	0
[17]	RD_OSR_LMT	RW	**AXI Maximum Read Out Standing Request Limit** This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1	1
[15:13]	Reserved	N/A	Reserved	0
[12]	AXI_AAL	R	**Address-Aligned Beats** This bit is read-only bit and reflects the AAL bit Register0 (Bus Mode Register [25]). When this bit set to 1, ETH performs address-aligned burst transfers on both read and write channels.	0
[11:4]	Reserved	N/A	Reserved	0
[3]	BLEN16	RW	**AXI Burst Length 16** When this bit is set to 1, or when UNDEF is set to 1, the ETH is allowed to select a burst length of 16.	0
[2]	BLEN8	RW	**AXI Burst Length 8** When this bit is set to 1, or when UNDEF is set to 1, the ETH is allowed to select a burst length of 8.	0
[1]	BLEN4	RW	**AXI Burst Length 4** When this bit is set to 1, or when UNDEF is set to 1, the ETH is allowed to select a burst length of 4.	0
[0]	UNDEF	R	**AXI Reserved Burst Length** This bit is read-only bit and indicates the complement (invert) value of FB	0

			bit in BMR (Bus Mode Register [16]). When this bit is set to 1, the ETH is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1] When this bit is set to 0, the ETH is allowed to perform only fixed burst lengths as indicated by BLEN16/8/4, or a burst length of 1.	

Register 15-22: AXI Status Register (AXISR)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1]	AMRCH	R	When high, it indicates that AXI Master's read channel is active and transferring data.	0
[0]	AMWCH	R	When high, it indicates that AXI Master's write channel is active and transferring data.	0

Register 15-23: Current Host Transmit Descriptor Register (CHTDR)

Field	Symbol	Direction	Description	Default
[31:0]	HTDAP	R	**Host Transmit Descriptor Address Pointer** Cleared on Reset. Pointer updated by DMA during operation.	0

Register 15-24: Current Host Transmit Buffer Address register (CHRDR)

Field	Symbol	Direction	Description	Default
[31:0]	HRDAP	R	**Host Receive Descriptor Address Pointer** Cleared on Reset. Pointer updated by DMA during operation.	0

Register 15-25: Current Host Transmit Buffer Address Register (CHTBAR)

Field	Symbol	Direction	Description	Default
[31:0]	HTBAP	R	**Host Transmit Buffer Address Pointer** Cleared on Reset. Pointer updated by DMA during operation.	0

Register 15-26: Current Host Receive Buffer Address Register (CHRBAR)

Field	Symbol	Direction	Description	Default
[31:0]	HRBAP	R	**Host Receive Buffer Address Pointer** Cleared on Reset. Pointer updated by DMA during operation.	0

16 Video Decoder

16.1 Video Decoder Overview

IMAPx210 video decoder supports video playback up to full HD (1080p) resolution at 30 frames per second (fps) in multiple formats including MPEG-1, MPEG-2, MPEG-4, Sorenson Spark(R), H.263, H.264, VC-1 and REALVIDEO(R) 8, 9 & 10, as well as up to 66 megapixel JPEG still images.

The video decoder core is also accompanied by a comprehensive suite of image processing functions including color space conversion, alpha blending, deinterlacing and image scaling.

The supported standards, profiles and levels are presented in Table 16-1. Possible deviations from the tools specified by these levels, and other points to notice are listed in Table 16-2.

Table 16-1: Supported Standards, Profiles and Levels

Standard	Decoder support
H.264 profile and level	• Baseline Profile, levels 1 - 4.1 • Main Profile, levels 1 - 4.1 • High Profile, levels 1 - 4.1
MPEG-4 visual profile and level	• Simple Profile, levels 0 - 6 1) • Advanced Simple Profile, levels 0 - 5 2)
MPEG-2 profile and level	Main Profile, low, medium and high levels
H.263 profile and level	• Profile 0, levels 10-70. Image size up to 720x576
Sorenson Spark profile and level	• Bitstream version 0 and 1
VC-1 profile and level	• Simple Profile, low, medium and high levels • Main Profile, low, medium and high levels • Advanced Profile, levels 0-3
JPEG profile and level	• Baseline interleaved
RV profile and level	• RV8 • RV9 • RV10

Note 1: MPEG-4 main profile streams that use simple profile tools only are also supported. MPEG-4 SP streams that have larger than 720p resolution conform to main profile.

Note 2: MPEG-4 main profile streams that use advanced simple profile tools only are also supported. MPEG-4 ASP streams that have larger than D1 resolution conform to main profile.

Table 16-2: Deviation From The Supported Profiless and Levels

Standard	Tool	Decoder support
H.263	Time code extensions	Not supported
H.264	Slice groups	If more than one slice group used, SW performs entropy decoding
H.264	Arbitrary slice order	Supported, SW performs entropy decoding
H.264	Redundant slices	Supported, but not utilized; redundant slices are skipped by SW
H.264	Image cropping	Not performed by the decoder, cropping parameters are returned to the application
MPEG-4	Data partitioning	Supported, SW performs entropy decoding
MPEG-4	Global motion compensation	Not supported

VC-1	Multi-resolution	Supported, upscaling will be performed by the post-processor
VC-1	Range mapping	Supported, range mapping will be performed by the post-processor
JPEG	Non-interleaved data order	Not supported
RV	Interlacing	Not supported

Block Diagram

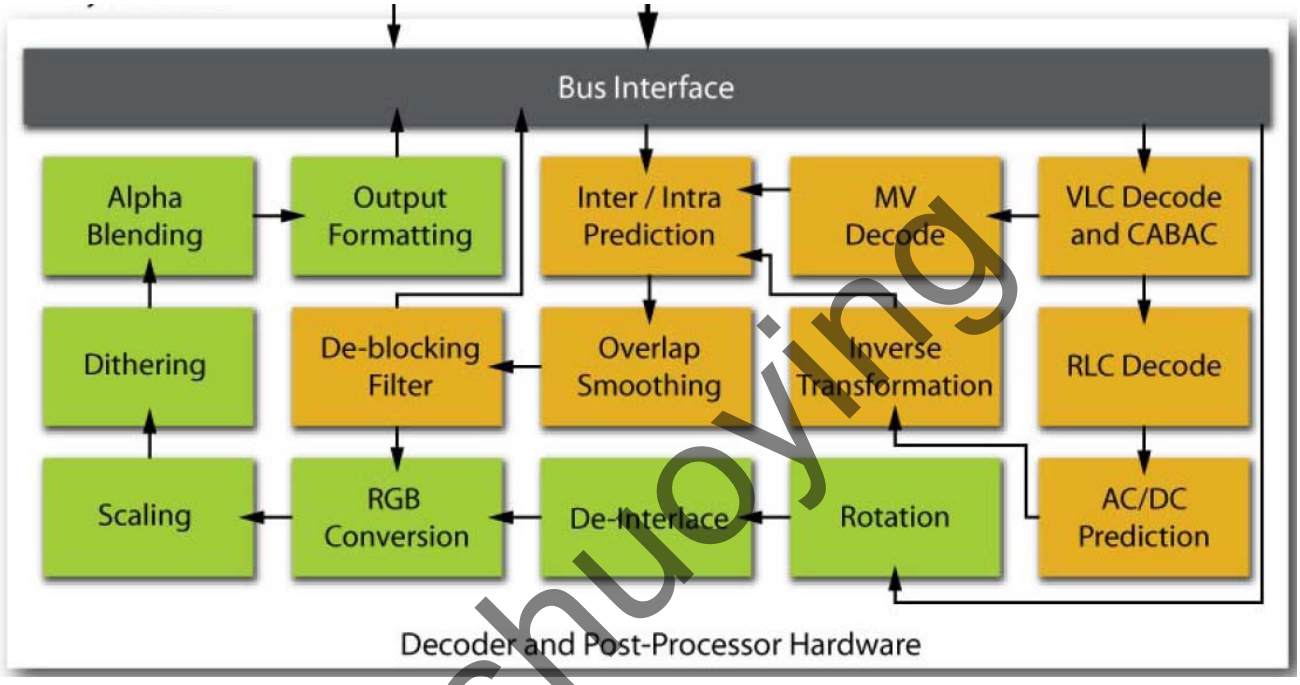


Figure 16-1: Video Decoder Block Diagram

16.2 Video Decoder Function Descriptions

16.2.1 Full-format Decoder Features

H.264

Feature	Decoder Support
Input data format	H.264 byte or NAL unit stream
Decoding scheme	• Frame by frame (or field by field) • Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	• 48 x 48 to 1920 x 1088 • Step size 16 pixels
Maximum frame rate	30 fps at 1920 x 1088
Maximum bit rate	As specified by H.264 HP level 4.1
Error detection and concealment	Supported

MPEG-4/H.263/SORENSON SPARK

Feature	Decoder Support
Input data format	MPEG-4 / H.263 / Sorenson Spark elementary video stream
Decoding scheme	• Frame by frame (or field by field) • Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	• 48 x 48 to 1920 x 1088 (MPEG-4, Sorenson Spark) • 48 x 48 to 720 x 576 (H.263) • Step size 16 pixels
Maximum frame rate	30 fps at 1920 x 1088
Maximum bit rate	As specified by MPEG-4 ASP level 5
Error detection and concealment	Supported

MPEG-2/ MPEG-1

Feature	Decoder Support
Input data format	MPEG-2 / MPEG-1 elementary video stream
Decoding scheme	• Frame by frame (or field by field) • Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	• 48 x 48 to 1920 x 1088 1) • Step size 16 pixels 2)
Maximum frame rate	30 fps at 1920 x 1088 3)
Maximum bit rate	As specified by MPEG-2 MP, high level
Error detection and concealment	Supported

JPEG

Feature	Decoder support
Input data format	• JFIF file format 1.02 • YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 sampling formats
Decoding scheme	• Input: buffer by buffer, from 5kB to 8MB at a time • Output: from 1 MB row to 16 Mpixels at a time
Output data format	YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
Supported image size	• 48 x 48 to 8176 x 8176 (66.8 Mpixels) • Step size 8 pixels
Maximum frame rate	Up to 76 million pixels per second
Maximum bit rate	JPEG compressed thumbnails supported
Error detection and concealment	Supported

VC-1

Feature	Decoder support
Input data format	VC-1 stream
Decoding scheme	• Frame by frame (or field by field) • Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	• 48 x 48 to 1920 x 1088 • Step size 16 pixels
Maximum frame rate	30 fps at 1920 x 1088
Maximum bit rate	As specified by VC-1 AP level 3
Error detection and concealment	Supported

Real video

Feature	Decoder support
Input data format	RV8 or RV9 (RV10) stream
Decoding scheme	• Frame by frame • Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	• 48 x 48 to 1920 x 1088 • Step size 16 pixels
Maximum frame rate	30 fps at 1920 x 1088
Maximum bit rate	As specified by RV specification
Error detection and concealment	Supported

16.2.2 Post-Processor

The post-processor (PP) features are described in Table 16-3. It is possible to run the post-processor combined with the decoder, or as a stand-alone IP block, when it can process image data from any external source. Typical examples of an external source are a camera interface or a 3rd party video decoder.

Using combined mode reduces bus bandwidth, as PP can read its input data directly from the decoder output without accessing external memory.

The post-processor output image can be alpha blended with two rectangular areas. If alpha blending is used in combined mode, the currently decoded image will be set as the background image. Alpha blending can be used for creating transparent menus, subtitles and logos on top of the video playback. These overlay regions must be in the same color space, YCbCr or RGB, as the target format of the post-processor output image. If the two areas for alpha blending overlap, the second area overrides the first (the first area content is discarded). Alpha blending increases the bus load.

Table 16-3 : Post-processor Features

Feature	Decoder support
Input data format	Any format generated by the decoder in combined mode • YCbCr 4:2:0 semi-planar • YCbCr 4:2:0 planar • YCbYCr 4:2:2 • YCrYCb 4:2:2 • CbYCrY 4:2:2 • CrYCbY 4:2:2
Post-processing scheme	• Frame by frame. Post-processor handles the image macroblock by macroblock, also in standalone mode.
Input image source	• Internal source (combined mode): IMAPx210 decoder • External source (standalone mode): e.g. a software decoder or camera interface
Output data format	• YCbCr 4:2:0 semi-planar • YCbYCr 4:2:2 • YCrYCb 4:2:2 • CbYCrY 4:2:2 • CrYCbY 4:2:2 • Fully configurable ARGB channel lengths and locations inside 32 bits, e.g. ARGB 32-bit (8-8-8-8), RGB 16-bit (5-6-5), ARGB 16-bit (4-4-4-4)
Input image size (combined mode)	• 48 x 48 to 8176 x 8176 (66.8 Mpixels) • Step size 16 pixels
Input image size (stand-alone mode)	• Width from 48 to 8176 • Height from 48 to 8176 • Maximum size limited to 16.7 Mpixels • Step size 16 pixels
Output image size	• 16 x 16 to 1920 x 1088 • Horizontal step size 8

	• Vertical step size 2
Image up-scaling	• Bicubic polynomial interpolation with a fourtap horizontal kernel and a two-tap vertical kernel • Arbitrary, non-integer scaling ratio separately for both dimensions • Maximum output width is 3x the input width (within the maximum output image size limit) • Maximum output height is 3x the input height – 2 pixels (within the maximum output image size limit)
Image down-scaling	• Proprietary averaging filter • Arbitrary, non-integer scaling ratio separately for both dimensions • Unlimited down-scaling ratio (e.g. from 16Mpixel to QVGA)
YCbCr to RGB color conversion	• BT.601-5 compliant • BT.709 compliant • User definable conversion coefficient
Dithering	2x2 ordered spatial dithering for 4, 5 and 6 bit RGB channel precision
Programmable alpha channel	Constant eight bit value can be set to the alpha channel of the 24-bit RGB output data to control the transparency of the output picture. The resulting 32-bit ARGB data can be used as input data for later alpha blending.
Alpha blending	Output image can be alpha blended with two rectangular areas 4). YCbCr semi-planar 4:2:0 PP output format is not supported when performing alpha blending. The supported overlay input formats are following: 1. 8 bit alpha value + YCbCr 4:4:4, big endian channel order being A-Y-Cb-Cr, 8 bits each 2. 8 bit alpha value + 24 bit RGB, big endian channel order being A-R-G-B, 8 bits each
Deinterlacing	Conditional spatial deinterlace filtering. Supports only YCbCr 4:2:0 input format.
RGB image contrast adjustment	Segmented linear
RGB image brightness adjustment	Linear
RGB image color saturation adjustment	Linear
De-blocking filter for MPEG-4 simple profile / H.263 / Sorenson	Using a modified H.264 in-loop filter as a postprocessing filter. Filtering has to be performed in combined mode.
Image cropping / digital zoom	User definable start position, height and width. Can be used with scaling to perform digital zoom. Usable only for JPEG or stand-alone mode.
Picture in picture	Output image can be written to any location inside video memory. Up to 1920 x 1088 sized displays supported
Output image masking	Output image writing can be prevented on two rectangular areas in the image . The masking feature is exclusive with alpha blending; however it is possible to have one masking area and one blending area.
Image rotation	• Rotation 90, 180 or 270 degrees • Horizontal flip • Vertical flip

16.2.3 Interrupt

The decoder interrupt register contains critical information for the integration such as the decoder enable bit, the interrupt bit and the interrupt status flags. When the software has initialized decoder control registers it will set the enable bit high. This will start the hardware decoding.

When the decoder hardware wants the software's attention, it sets the interrupt bit high with one of the status flags providing information about the reason for the interrupt. When the software has handled the interrupt it must reset all status flags to zero. The interrupt bit stays high until software has reset it.

The interrupt method can be set to interrupting or polling.

16.2.4 Device Configuration

In the device configuration register for example the DRAM accessing related items of the hardware can be adjusted. Adjusting these values will affect the performance of the hardware.

Decoder output picture format is used for selecting the storage order of the pixels of the decoded reference pictures. This does not affect JPEG decoding or the post-processed output picture.

Note: The reference picture pixel order does not affect the semi-planarity of the picture. Luminance and interleaved chrominance are written to their own planes in both formats.

The decoder core internal bus service priority selector (bits 7:5) controls a finite state machine inside decoder that decides which internal decoder request for external memory is served first if there are several waiting to be served. The default value 0 should usually give the best performance, but depending on target environment wait states and latencies, it might be useful to test the performance with all priority modes. The performance differences between the modes are in a few percent range.

16.3 Video Decoder Register Description

16.3.1 Video Decoder Register Memory Map

Address	Symbol	Direction	Description
0x20D1_0000	IDR	RO	ID register
0x20D1_0004	INTR	R/W	Interrupt register
0x20D1_0008	DCR	R/W	Device configuration register
0x20D1_000C	DCON0	R/W	Decoder control register 0
0x20D1_0010	DCON1	R/W	Decoder control register 1
0x20D1_0014	DCON2	R/W	Decoder control register 2
0x20D1_0018	DCON3	R/W	Decoder control register 3
0x20D1_001C	DCON4	R/W	Decoder control register 4
0x20D1_0020	DCON5	R/W	Decoder control register 5
0x20D1_0024	DCON6	R/W	Decoder control register 6/ Base address for MB-control/ VC-1 intensity control
0x20D1_0028	DCON7	R/W	Decoder control register 7/ Base address for differential motion vectors/ VC-1 intensity control 1
0x20D1_002C	DCON8	R/W	Decoder control register 8 / Base address for H.264 intra prediction 4x4 modes / Base address for MPEG-4 DC components
0x20D1_0030	RLCBA	R/W	Base address for RLC data / H.264/MPEG-4/H.263/JPEG stream start address/ Decoded stream end address register
0x20D1_0034	DOPBA	R/W	Base address for decoded picture / Base address for JPEG decoder output luminance picture
0x20D1_0038	RPBA0	R/W	Base address for H.264 reference picture index 0 / Base address for JPEG decoder output chrominance picture
0x20D1_003C	RPBA1	R/W	Base address for reference picture index 1 / MPEG-4 and JPEG control
0x20D1_0040	RPBA2	R/W	Base address for reference picture index 2 / List of VLC code lengths in first JPEG AC table
0x20D1_0044	RPBA3	R/W	Base address for reference picture index 3 / List of VLC code lengths in first JPEG AC table
0x20D1_0048	RPBA4	R/W	Base address for reference picture index 4 /

			List of VLC code lengths in first JPEG AC table
0x20D1_004C	RPBA5	R/W	Base address for reference picture index 5 / List of VLC code lengths in first JPEG AC table
0x20D1_0050	RPBA6	R/W	Base address for reference picture index 6 / List of VLC code lengths in first JPEG AC table
0x20D1_0054	RPBA7	R/W	Base address for reference picture index 7 / List of VLC code lengths in first JPEG AC table
0x20D1_0058	RPBA8	R/W	Base address for reference picture index 8 / List of VLC code lengths in first JPEG AC table
0x20D1_005C	RPBA9	R/W	Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table
0x20D1_0060	RPBA10	R/W	Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table
0x20D1_0064	RPBA11	R/W	Base address for reference picture index 11 / List of VLC code lengths in first JPEG DC table
0x20D1_0068	RPBA12	R/W	Base address for reference picture index 12 / List of VLC code lengths in first JPEG DC table
0x20D1_006C	RPBA13	R/W	Base address for reference picture index 13 / Base address for JPEG AC, DC and QP
0x20D1_0070	RPBA14	R/W	Base address for reference picture index 14
0x20D1_0074	RPBA15	R/W	Base address for reference picture index 15
0x20D1_0078	RPN01	R/W	Reference picture numbers for index 0 and 1
0x20D1_007C	RPN23	R/W	Reference picture numbers for index 2 and 3
0x20D1_0080	RPN45	R/W	Reference picture numbers for index 4 and 5
0x20D1_0084	RPN67	R/W	Reference picture numbers for index 6 and 7
0x20D1_0088	RPN89	R/W	Reference picture numbers for index 8 and 9
0x20D1_008C	RPNAB	R/W	Reference picture numbers for index 10 and 11
0x20D1_0090	RPNCD	R/W	Reference picture numbers for index 12 and 13
0x20D1_0094	RPNEF	R/W	Reference picture numbers for index 14 and 15
0x20D1_0098	RPLTVF	R/W	Reference picture long term and valid flags
0x20D1_009C	RPVF	R/W	Reference picture valid flags
0x20D1_00A0	SDTBA	R/W	Base address for standard dependent tables
0x20D1_00A4	DMMVBA	R/W	Base address for direct mode motion vectors
0x20D1_00A8	IRPL0	R/W	H.264 initial reference picture list register
0x20D1_00AC	IRPL1	R/W	H.264 initial reference picture list register
0x20D1_00B0	IRPL2	R/W	H.264 initial reference picture list register
0x20D1_00B4	IRPL3	R/W	H.264 initial reference picture list register
0x20D1_00B8	IRPL4	R/W	H.264 initial reference picture list register
0x20D1_00BC	IRPL5	R/W	H.264 initial reference picture list register
0x20D1_00C0	ERRCON	R/W	Error concealment register
0x20D1_00C4	PFTR	R/W	Prediction filter tap register
0x20D1_00C8	SCR	RO	Synthesis configuration register
0x20D1_00CC	RPBCR0	R/W	Reference picture buffer control register 0
0x20D1_00D0	RPBCR1	R/W	Reference picture buffer control register 1
0x20D1_00D4	RPBCR2	R/W	Reference picture buffer control register 2
0x20D1_00D8	Reserved		
0x20D1_00DC	RPBCR3	R/W	Reference picture buffer control register 3
0x20D1_00E8	DBGR0	R/W	Decoder debug register 0
0x20D1_00EC	DBGR1	R/W	Decoder debug register 1
0x20D1_00F0	PPINTR	R/W	Interrupt register, post-processor
0x20D1_00F4	PPDCR	R/W	Device configuration register, post-processor
0x20D1_00F8	PPDEICR	R/W	Deinterlace control register

0x20D1_00FC	PPIYBA	R/W	Base address for post-processor input Y picture
0x20D1_0100	PPICBBA	R/W	Base address for post-processor input Cb picture
0x20D1_0104	PPICRBA	R/W	Base address for post-processor input Cr picture
0x20D1_0108	PPOYBA	R/W	Base address for post-processor output Y picture
0x20D1_010C	PPOCBA	R/W	Base address for post-processor output C picture
0x20D1_0110	PPCADJR	R/W	Contrast adjustment register
0x20D1_0114	PPCCACADJ	R/W	Colour conversion and contrast adjustment
0x20D1_0118	PPCCONV0	R/W	Colour conversion register 0
0x20D1_011C	PPCCARM	R/W	Colour conversion and rotation mode register
0x20D1_0120	PPIPSIZE	R/W	Post-processing input picture size and cropping register
0x20D1_0124	PPIYBFBA	R/W	Base address for post-processing input Y bottom field
0x20D1_0128	PPICBFBA	R/W	Base address for post-processing input C bottom field
0x20D1_013C	PPSCALERG	R/W	Scaling register 0 ratio, padding for R and G
0x20D1_0140	PPSCALEB	R/W	Scaling ratio register 1, padding for B
0x20D1_0144	PPSCALE2	R/W	Scaling ratio register 2
0x20D1_0148	RCMR	R/W	Red channel mask register
0x20D1_014C	GCMR	R/W	Green channel mask register
0x20D1_0150	BCMR	R/W	Blue channel mask register
0x20D1_0154	PPCON	R/W	Post-processor control register
0x20D1_0158	M1SCR	R/W	Mask 1 start coordinate register
0x20D1_015C	M2SCR	R/W	Mask 2 start coordinate register
0x20D1_0160	M1SAW	R/W	Mask 1 size and PP original width register
0x20D1_0164	M2SIZE	R/W	Mask 2 size register
0x20D1_0168	PIPR0	R/W	Picture-in-picture register0
0x20D1_016C	PIPR1	R/W	Picture-in-picture register 1 and dithering control
0x20D1_0170	DISPW	R/W	Display width register
0x20D1_0174	ABGCBA1	R/W	Base address for alpha blending GUI component 1
0x20D1_0178	ABGCBA2	R/W	Base address for alpha blending GUI component 2
0x20D1_0190	PPSCR	RO	Synthesis configuration register, post-processor

16.3.2 Video Decoder Individual Register Description

Register 16-1: Decoder ID Register（IDR, offset=0x0000）

Field	Symbol	Direction	Description	Default
[31:16]	PID	RO	Product ID	0x9170
[25:12]	MAJVN	RO	Major version number	0x0
[11:4]	MINVN	RO	Minor version number	0x96
[3:0]	BVN	RO	Build version number	0x0

Register 16-2: Decoder Interrrupt Register（INTR, offset=0x0004）

Field	Symbol	Direction	Description	Default
[31:19]	Reserved	RO	Reserved	0x0
[18]	TimeOut	R/W	Interrupt status bit timeout. When high, decoder has made no bus transactions in 2^{18}- 1 clock cycles and has not set an interrupt. Possible only if timeout interrupting is enabled. This should be considered as an encountered error in the input stream. Note: Post-processor transactions affect this feature; running stand-alone post-processing while decoding mayprevent decoder timeout interrupts.	0x0
[17]	JpegsliceDecoded	R/W	Interrupt status flag JPEG slice decoded. Hardware has decoded the	0x0

			requested amount of macroblock rows of a JPEG picture. When high, the SW must set a new base addresses for the decoder output. Hardware will not self-reset.	
[16]	InputStreamError	R/W	Interrupt status flag input stream error. When high, hardware has encountered a bit error in the input stream data, and software must perform error concealment. Hardware will self-reset.	0x0
[15]	ASODetected		Interrupt status flag ASO detected. When high, hardware has encountered Arbitrary Slice Order tool in the input H.264 stream data, and software must perform entropy decoding. Hardware will self-reset.	0x0
[14]	StreamBufEmpty		Interrupt status flag stream buffer empty. When high, the input stream buffer is empty but the picture is not ready. Software must provide a new stream pointer to hardware. Hardware will not self-reset.	0x0
[13]	BusError		Interrupt status flag bus error. When high, hardware has received an error response from the bus while accessing external memory. This is a fatal error possibly caused by the incorrect allocation of decoder linear memory. Hardware will self-reset.	0x0
[12]	DecoderReady		Interrupt status flag decoder ready. When high, the hardware has decoded a picture. Hardware will self-reset.	0x0
[11:9]	Reserved	RO	Reserved	0x0
[8]	Interrupt		Decoder interrupt bit. This bit drives the interrupt line, OR gated with the post-processor interrupt bit. Software will reset this after the interrupt is handled. The interrupt line is not used for the decoder if the interrupt disable bit for decoder is high.	0x0
[7:5]	Reserved	RO	Reserved	0x0
[4]	InterruptDisable		Interrupt disable for decoder. When high, there will be no interrupts issued by the decoder. Polling must be used to see the hardware status.	0x0
[3:1]	Reserved	RO	Reserved	0x0
[0]	DecoderEnable		Decoder enable. Setting this bit high will start the decoding operation. Setting this bit low will reset all the control flip-flops. Hardware will reset this bit when a picture is processed, a stream error or ASO is detected or a bus error is received.	0x0

Register 16-3: Decoder Post Processor Interrrupt Register（PPINTR, offset=0x00F0）

Field	Symbol	Direction	Description	Default
[31:14]	Reserved	RO	Reserved	0x0
[13]	BusError	R/W	Interrupt status flag bus error. When high, hardware has received an error response from the bus while accessing external memory. This is a fatal error possibly caused by the incorrect allocation of postprocessor linear memory. Hardware will self-reset.	0x0
[12]	PPReady	R/W	Interrupt status flag post-processor ready. If decoder and post-processor are running in combined mode, this bit is not used. When high, the hardware has post-processed a picture. Hardware will self-reset.	0x0
[11:9]	Reserved	RO	Reserved	0x0

[8]	PPInterrrupt	R/W	Post-processor interrupt bit. This bit drives the interrupt line, OR gated with the decoder interrupt bit. Software will reset this after the interrupt is handled. The interrupt line is not used if the interrupt disable bit for postprocessor is high.	0x0
[7:5]	Reserved	RO	Reserved	0x0
[4]	InterruptDisable	R/W	Interrupt disable for post-processor. When high, there will be no interrupts issued by the postprocessor. Polling must be used to see the hardware status.	0x0
[3:2]	Reserved	RO	Reserved	0x0
[1]	PPCombineModeEN	R/W	Decoder–post-processing combined mode enable. When high, post-processing is performed in pipeline with decoder. When low, post-processor is processing different picture than the decoder.	0x0
[0]	PPStandAloneEN	R/W	Post-processor standalone enable bit. This bit is not used if the combined mode is enabled. Setting this bit high will start the post-processing operation. Hardware will reset this bit when a picture is processed	0x0

Register 16-4: Decoder Device configuration Register（DCR, offset=0x0008）

Field	Symbol	Direction	Description	Default
[31:24]	AXIRID	R/W	AXI read ID value. Used in all AXI read transfers.	0x0
[23]	TimeOutEN	R/W	Enable hardware timeout interrupting.	0x0
[22]	IStreamSwap	R/W	32-bit swap for decoder input stream data. 1)	0x0
[21]	InputEndianness	R/W	Decoder input endian mode for stream data. 0 = Big endian 1 = Little endian	0x0
[20]	IDataSwap	R/W	32-bit swap for decoder input data through the SW/HW interface. 1)	0x0
[19]	OSwap	R/W	32-bit swap for decoder output data. 1)	0x0
[18:17]	Reserved	RO	Reserved	0x0
[16:11]	IdleLatency	R/W	Decoder external memory access context switching idle latency. This setting defines how many clock cycles the master interface shall wait before changing data context when accessing external memory. Other devices may get their turn at accessing memory during this wait. Range: [0-63], the actual applied latency in clock cycles is the defined value multiplied by 8.	0x0
[10]	ClockGate	R/W	Decoder dynamic clock gating enable. When high, clock is gated for decoder structures that are not used. When low, clock is running for all structures. Note: Clock gating value should be changed only when the decoder is disabled.	0x0
[9]	IDataEndianness	R/W	Decoder input endian mode for data through the SW/HW interface. 0 = Big endian 1 = Little endian	0
[8]	ODataEndianness	R/W	Decoder output endian mode. 0 = Big endian 1 = Little endian	0
[7:5]	Reserved	RO	Reserved	0x0
[4:0]	MaxBurstlength	R/W	Maximum burst length for decoder bus transactions. Valid values for	0

			AXI are 1-16.	

1) When high, 64-bit little endian mode results in byte order 7-6-5-4-3-2-1-0, and big endian in 4-5-6-7-0-1-2-3.
When low, 64-bit little endian mode results in byte order 3-2-1-0-7-6-5-4 and big endian in 0-1-2-3-4-5-6-7.

Register 16-5: Decoder Post Processor Device configuration Register（PPDCR, offset=0x00F4）

Field	Symbol	Direction	Description	Default
[31:24]	AXIRID	R/W	AXI read ID value. Used in all AXI read transfers.	0x0
[23:16]	AXIWID	R/W	AXI write ID value. Used in all AXI write transfers.	0x0
[15:11]	Reserved	RO	Reserved	0x0
[10]	IDataSwap	R/W	32-bit swap for post-processor input data. Has no effect if post-processor is pipelined with a decoder. 1)	0x0
[9]	Reserved	RO	Reserved	0x0
[8]	ClockGateEN	R/W	Post-processor dynamic clock gating enable. When high, clock is gated from PP structures that are not used. When low, clock is running for all PP structures. Note: Clock gating value should be changed only when the PP is disabled	0x1
[7]	IEndianness	R/W	Post-processor input picture byte endian mode. This bit is used only if the post-processor is used in standalone mode. 0 = Big endian 1 = Little endian	0x0
[6]	OEndianness	R/W	PP output picture endian mode for YCbCr data. 0 = Big endian 1 = Little endian Note: For 16-bit RGB data this bit swaps the two 16-bit pixels. For 32-bit RGB this bit has no effect	0x0
[5]	ODataSwap	R/W	32-bit swap for post-processor output data. 1)	0x0
[4:0]	MaxBurstlength	R/W	Maximum burst length for PP bus transactions. Valid values of AXI are 1-16.	0x0

1) When high, 64-bit little endian mode results in byte order 7-6-5-4-3-2-1-0, and big endian in 4-5-6-7-0-1-2-3.
When low, 64-bit little endian mode results in byte order 3-2-1-0-7-6-5-4 and big endian in 0-1-2-3-4-5-6-7.

17 Video Encoder

17.1 Video Encoder Overview

IMAPx210 Multi-format video encoder processes up to 1280 x 1024 resolution video at 30 frames per second. IMAPx210 video encoder support MPEG-4,, H.263, H.264, as well as up to 16 megapixel JPEG still images.

Table 17-1: Supported Standards, Profiles and Levels

Standard	Encoder Support
H.264 Profile and level	Baseline Profile, levels 1-3.2
MPEG-4 Visual profile and level	Simple Profile, levels 0-6 Main Profile, level 4
H.263 profile and level	Profile 0, levels 10-70. Image size up to 720x576, time code extensions not supported
JPEG profile and level	Baseline

Table 17-2: Supported H.264 Tools

Tool	Encoder Support
Slices	I and P slices
Entropy encoding	CAVLC
Basic	Constrained intra prediction Maximum MV range +-16 pixels MV accuracy ¼ pixels All block sizes from 4x4 to 16x16 supported
Number of reference frames	1
Maximum number of slice groups	1

Table 17-3: Supported MPEG-4 Visual Tools

Tool	Encoder support
Basic	I and P-VOPs Maximum MV range +-16 pixels MV accuracy ½ pixels 1 or 4 MV/Macroblock DC prediction
Error resilience	Video packets (SW performs entropy encoding) Data partitioning (SW performs entropy encoding) Reversible VLC
Number of reference frames	1
Quantization	Method 2
Number of visual objects	1
Short Video Header	Yes

Table 17-4: Supported H.263 Visual Tools

Tool	Encoder support
Basic	I and P-VOPs Maximum MV range +-16 pixels MV accuracy ½ pixels 1 MV/Macroblock
Error resilience	GOB
Number of reference frames	1

Block Diagram

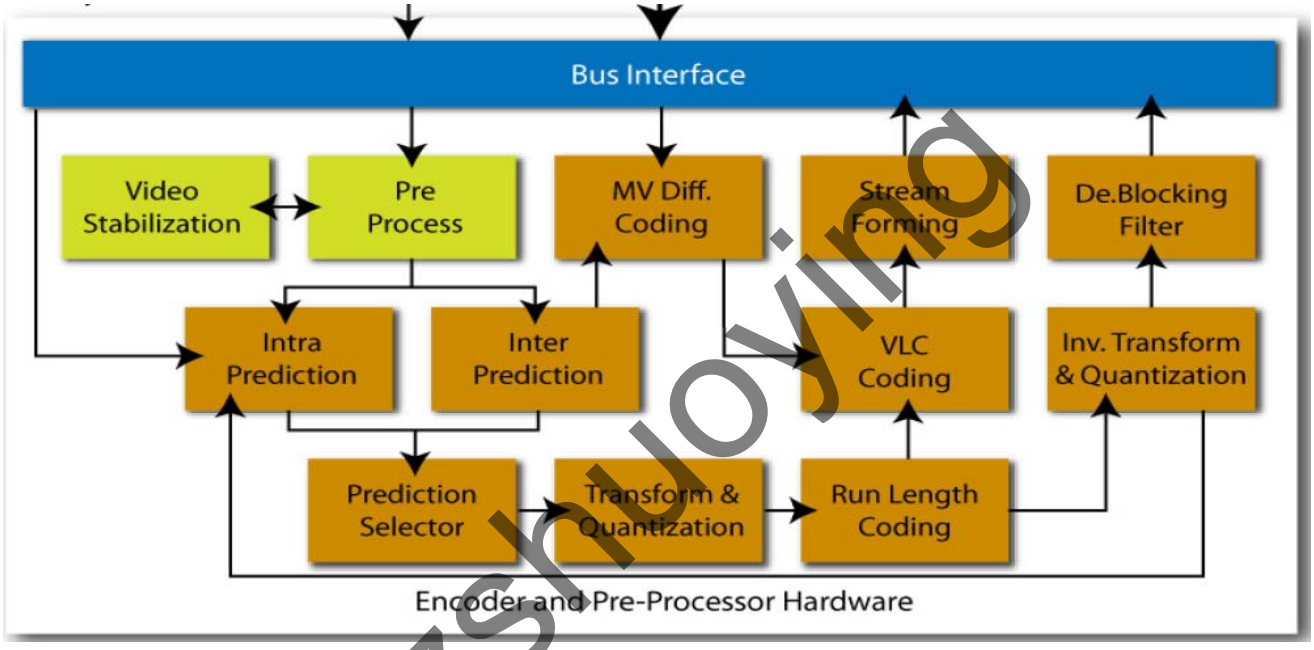


Figure 17-1: Video Encoder Block Diagram

17.2 Video Encoder Function Descriptions

17.2.1 Supported Standards and Tools

H.264

Feature	Encoder Support
Input data format	YCbCr 4:2:0 planar or semi-planar YCbYCr and CbYCrY 4:2:2 Interleaved
Output data format	H.264 byte or NAL unit stream
Supported image size	96 x 96 to 1280 x 1024 (SXGA) Step size 4 pixels
Maximum frame rate	25 fps at 720x576, or 30 fps at 720x480 or 30 fps at 1280x720 or 30 fps at 1280x1024
Maximum bit rate	20 Mbps

MPEG-4/H.263

Feature	Encoder Support
Input data format	YCbCr 4:2:0 planar or semi-planar YCbYCr and CbYCrY 4:2:2 Interleaved
Output data format	MPEG-4 / H.263 elementary video stream
Supported image size	96 x 96 to 1280 x 1024 (SXGA) Step size 4 pixels
Maximum frame rate	25 fps at 720x576, or 30 fps at 720x480 30 fps at 1280x720 or 30 fps at 1280x1024
Maximum bit rate	10 Mbps

JPEG

Feature	Encoder Support
Input data format	YCbCr 4:2:0 planar or semi-planar YCbYCr and CbYCrY 4:2:2 Interleaved
Output data format	JFIF file format 1.02 Non-progressive JPEG
Supported image size	80x16 to 4672 x 3504 (16.4 million pixels) Step size 4 pixels
Maximumdata rate	Up to 90 million pixels per second
Thumbnail encoding	JPEG compressed thumbnails supported

Pre-processing

Feature	Encoder Support
Color space conversion	YCbYCr or CbYCrY 4:2:2 Interleaved or semi-planar 4:2:0 to YCbCr 4:2:0
Cropping	Video - from 4672 x 3504 to any supported encoding size
Rotation	90 or 270 degrees

Video stabilization

Feature	Encoder Support
Maximum stabilization displacement in pixels for two sequential input video pictures	+-16 pixels
Adaptive motion compensation filter	From 6 to 40 sequential video pictures noticed in unwanted and wanted movement separation
Offset around stabilized picture	Minimum 8 pixels in standalone mode Minimum 16 pixels when pipelined with video encoder Recommended 64 pixels Maximum not limited

17.2.2 Encoder Data Flow

Figure 17-2 illustrates the encoder data flow in H.264 encoding mode. Some of the transfers are omitted in MPEG-4 and JPEG encoding modes which are thus causing less bus load than H.264 encoding. Otherwise the encoder behaves in a similar way in each mode.

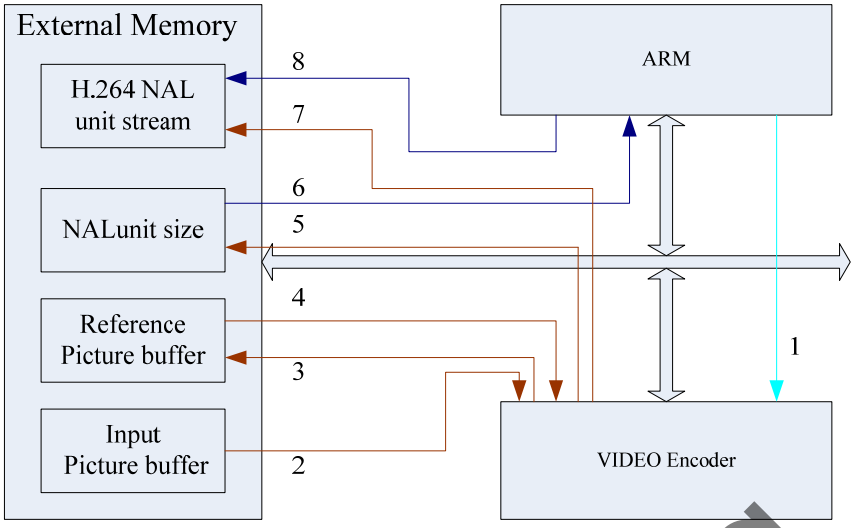


Figure 17-2: Encoder and External Memory Data Flow

The numbers presented in the above figure represent the following transactions:

1. Memory mapped register writes and reads
2. Input image read
3. Reference image write (not included in JPEG)
4. Reference image read (not included in JPEG)
5. NAL sizes write from HW
6. NAL sizes read to SW
7. Output byte or NAL unit stream write from HW
8. Output byte or NAL unit stream headers write from SW

Note 1: The encoder can be run in slice mode when encoding multimillion pixel JPEG images. In this case each slice behaves from the hardware's point of view as it was a separate picture. This approach keeps the required SW/HW interface memory in reasonable limits.

Note 2: If Video Stabilization is enabled two input picture buffers are needed. Reading is simultaneous with encoding read process. Stand alone Video Stabilization mode only two input picture buffers are required and encoder returns address offset for stabilized picture.

The encoder software starts encoding the first picture by initializing hardware and writing the stream headers. After HW has encoded the image, SW calculates new quantization values for HW, and initializes HW again.

17.3 Video Encoder Register Description

17.3.1 Video Encoder Register Memory Map

Address	Symbol	Direction	Description
0x20D0_0000	IDR	RO	Encoder ID register
0x20D0_0004	INTR	R/W	Encoder interrupt register
0x20D0_0008	DCR	R/W	Encoder device configuration register
0x20D0_000C	ITR	R/W	Encoder integration test register
0x20D0_0010	IIR	R/W	Encoder interrupt interval register
0x20D0_0014	ODBA	R/W	Encoder base address register for output data
0x20D0_0018	OCBA	R/W	Encoder base address register for output control

0x20D0_001C	LRPBA	R/W	Encoder reference picture base address register for luminance
0x20D0_0020	CRPBA	R/W	Encoder reference picture base address register for chrominance
0x20D0_0024	LNRPBA	R/W	Encoder new reference picture base address register for luminance
0x20D0_0028	CNRPBA	R/W	Encoder new reference picture base address register for chrominance
0x20D0_002C	LIPBA	R/W	Encoder input picture base address register for luminance
0x20D0_0030	CbIPBA	R/W	Encoder input picture base address register for Cb
0x20D0_0034	CrIPBA	R/W	Encoder input picture base address register for Cr
0x20D0_0038	CCR1	R/W	Encoder common control register 1
0x20D0_003C	CCR2	R/W	Encoder common control register 2
0x20D0_0040	H264CR1	R/W	Encoder H.264 control register 1
0x20D0_0044	H264CR2	R/W	Encoder H.264 control register 2
0x20D0_0048	H264CR3	R/W	Encoder H.264 control register 3
0x20D0_004C	MP4CR	R/W	Encoder MPEG-4 / H.263 control register
0x20D0_0050	JPCR	R/W	Encoder JPEG control register
0x20D0_0054	FAVCR	R/W	Encoder favor control register
0x20D0_0058	SDBCR1	R/W	Encoder stream data buffer control register 1
0x20D0_005C	SDBCR2	R/W	Encoder stream data buffer control register 2
0x20D0_0060	DCLR	R/W	Encoder data counter / buffer limit register
0x20D0_0064	RCPR	RO	Encoder rate control parameter register
0x20D0_0068		RO	Read only register
0x20D0_006C	QPCR	R/W	Encoder quantization parameter control register
0x20D0_0070	RCTCR1	R/W	Encoder rate control target count register for checkpoints 1,2
0x20D0_0074	RCTCR3	R/W	Encoder rate control target count register for checkpoints 3,4
0x20D0_0078	RCTCR5	R/W	Encoder rate control target count register for checkpoints 5,6
0x20D0_007C	RCTCR7	R/W	Encoder rate control target count register for checkpoints 7,8
0x20D0_0080	RCTCR9	R/W	Encoder rate control target count register for checkpoints 9,10
0x20D0_0084	RCCET1	R/W	Encoder rate control count error table 1 register
0x20D0_0088	RCCET2	R/W	Encoder rate control count error table 2 register
0x20D0_008C	RCCET3	R/W	Encoder rate control count error table 3 register
0x20D0_0090	RCDQPT	R/W	Encoder rate control delta QP table register
0x20D0_0094	RLCSUM	RO	Encoder RLC sum register
0x20D0_0098	MBCR	RO	Encoder MB counter register
0x20D0_009C	LNIPBA	R/W	Encoder next input picture base address register for luminance
0x20D0_00A0	STAB1	R/W	Stabilization register 1
0x20D0_00A4	STAB2	RO	Stabilization register 2
0x20D0_00A8	TLMV	RO	Top-left matrix value around full-pixel minimum
0x20D0_00AC	TMMV	RO	Top-middle matrix value around full-pixel minimum
0x20D0_00B0	TRMV	RO	Top-right matrix value around full-pixel minimum
0x20D0_00B4	MLMV	RO	Middle-left matrix value around full-pixel minimum
0x20D0_00B8	MMV	RO	Middle matrix value around full-pixel minimum
0x20D0_00BC	MRV	RO	Middle-right matrix value around full-pixel minimum
0x20D0_00C0	BLMV	RO	Bottom-left matrix value around full-pixel minimum
0x20D0_00C4	BMMV	RO	Bottom-middle matrix value around full-pixel minimum
0x20D0_00C8	BRMV	RO	Bottom-right matrix value around full-pixel minimum
0x20D0_00FC	CFG	RO	Encoder config register
0x20D0_0100	JPGTQ1	WO	Encoder JPEG Q table write register 1
0x20D0_0104	JPGTQ2	WO	Encoder JPEG Q table write register 2
0x20D0_0108	JPGTQ3	WO	Encoder JPEG Q table write register 3
0x20D0_010C	JPGTQ4	WO	Encoder JPEG Q table write register 4
0x20D0_0110	JPGTQ5	WO	Encoder JPEG Q table write register 5

0x20D0_0114	JPGTQ6	WO	Encoder JPEG Q table write register 6
0x20D0_0118	JPGTQ7	WO	Encoder JPEG Q table write register 7
0x20D0_011C	JPGTQ8	WO	Encoder JPEG Q table write register 8
0x20D0_0120	JPGTQ9	WO	Encoder JPEG Q table write register 9
0x20D0_0124	JPGTQ10	WO	Encoder JPEG Q table write register 10
0x20D0_0128	JPGTQ11	WO	Encoder JPEG Q table write register 11
0x20D0_012C	JPGTQ12	WO	Encoder JPEG Q table write register 12
0x20D0_0130	JPGTQ13	WO	Encoder JPEG Q table write register 13
0x20D0_0134	JPGTQ14	WO	Encoder JPEG Q table write register 14
0x20D0_0138	JPGTQ15	WO	Encoder JPEG Q table write register 15
0x20D0_013C	JPGTQ16	WO	Encoder JPEG Q table write register 16
0x20D0_0140	JPGTQ17	WO	Encoder JPEG Q table write register 17
0x20D0_0144	JPGTQ18	WO	Encoder JPEG Q table write register 18
0x20D0_0148	JPGTQ19	WO	Encoder JPEG Q table write register 19
0x20D0_014C	JPGTQ20	WO	Encoder JPEG Q table write register 20
0x20D0_0150	JPGTQ21	WO	Encoder JPEG Q table write register 21
0x20D0_0154	JPGTQ22	WO	Encoder JPEG Q table write register 22
0x20D0_0158	JPGTQ23	WO	Encoder JPEG Q table write register 23
0x20D0_015C	JPGTQ24	WO	Encoder JPEG Q table write register 24
0x20D0_0160	JPGTQ25	WO	Encoder JPEG Q table write register 25
0x20D0_0164	JPGTQ26	WO	Encoder JPEG Q table write register 26
0x20D0_0168	JPGTQ27	WO	Encoder JPEG Q table write register 27
0x20D0_016C	JPGTQ28	WO	Encoder JPEG Q table write register 28
0x20D0_0170	JPGTQ29	WO	Encoder JPEG Q table write register 29
0x20D0_0174	JPGTQ30	WO	Encoder JPEG Q table write register 30
0x20D0_0178	JPGTQ31	WO	Encoder JPEG Q table write register 31
0x20D0_017C	JPGTQ32	WO	Encoder JPEG Q table write register 32

17.3.2 Video Encoder Individual Register Description

Register 17-1: Encoder ID Register（IDR, offset=0x0000）

Field	Symbol	Direction	Description	Default
[31:16]	PID	RO	Product ID	0x7280
[25:12]	MAJVN	RO	Major version number	0x2
[11:4]	MINVN	RO	Minor version number	0x2
[3:0]	BVN	RO	Build version number	0x0

Register 17-2: Encoder Interrupt Register（INTR, offset=0x0004）

Field	Symbol	Direction	Description	Default
[31:9]	Reserved	RO	Reserved	0x0
[8]	INTERVAL	R/W	Interrupt status bit for interrupt interval reached. Active high. When this bit is set then encoder is processed the desired amount of MBs.	0x0
[7]	MCCO	R/W	Interrupt status bit for memory cache coherency test. Active high. When this bit is set, hw has copied data from input buffer to output buffer.	0x0
[6]	INTTST	R/W	Interrupt status bit for interrupt test. Active high. When interrupt test bit (bit0) in integration test register(ITR) is set, status of this is activated.	0x0
[5]	SDBF	R/W	Interrupt status bit for stream data buffer full. Active high. When this bit is high encoding is waiting until this bit is lowered.	0x0
[4]	SFRST	R/W	Interrupt status bit for software reset. Active high. If software resets the hardware while encoding then this bit is written high.	0x0
[3]	BUSERR	R/W	Interrupt status bit for bus error. Active high.	0x0

[2]	FRMRDY	R/W	Interrupt status bit for encoder frame ready. Active high. When this bit is high encoder has coded a picture.	0x0
[1]	INTDIS	R/W	Interrupt disable. Active high. When high, there are no interrupts (HINTenc) concerning encoder from HW. Polling must be used to see the interrupt.	0x0
[0]	HINTenc	R/W	Interrupt HINTenc. SW will reset this after interrupt is handled. HINTenc is not used if IRQ disable is high (swreg1 bit 1).	0x0

Register 17-3: Encoder Device Configuration Register（DCR, offset=0x0008）

Field	Symbol	Direction	Description	Default
[31:24]	AXIWID	R/W	AXI write ID value. Used in all AXI write transfers	0x0
[23:16]	AXIRID	R/W	AXI read ID value. Used in all AXI read transfers.	0x0
[15:14]	Reserved	RO	Reserved	0x0
[13:8]	MAXLEN	RO	Maximum burst length for encoder bus transactions.Valid values 1-16 for AXI.	0x0
[7:5]	Reserved	RO	Reserved	0x0
[4]	CGE	R/W	Clock gating enable. 0 = Clock gating is disabled. 1 = Clock gating is enabled.	0x1
[3]	SWPO	RO	Swap of 32-bit words in 64-bit environment. For encoder output data. 0 = Swap disabled 1 = Swap enabled	0x0
[2]	SWPI	R/W	Swap of 32-bit words in 64-bit environment. For encoder input picture. 0 = Swap disabled 1 = Swap enabled	0x0
[1]	ENDO	R/W	Endian mode for encoder output data. 0 = Big endian 1 = Little endian	0x0
[0]	ENDI	R/W	Endian mode for encoder input picture. 0 = Big endian 1 = Little endian	0x0

Register 17-4: Encoder Integration Test Register（DCR, offset=0x0008）

Field	Symbol	Direction	Description	Default
[31:28]	CNT	R/W	Counter	0x0
[27:21]	Reserved	RO	Reserved	0x0
[20:3]	MCCTLEN	R/W	Data length for memory cache coherency test. Amount of 64 bit words.	0x0
[2]	MCCTE	R/W	Input/Output memory cache coherency test enable. When this bit is set to one HW copies data from the location pointed by the base address of ODBA to the location pointed by the base address of OCBA. Data length is specified in bits 20:3.	0x0
[1]	RCCTE	R/W	Register cache coherency test enable. When this bit is set to 1, HW reads bits 31:28 from register, increases value by one and writes result back to the same bits. Only once per enable HW self-reset this enable.	0x0
[0]	INTTST	R/W	Interrupt test register, when this bit is set to 1, HW gives an interrupt	0x0

Register 17-5: Encoder Config Register（CFG, offset=0x00FC）

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	RO	Reserved	0x0

[27]	H264	RO	Encoding format support, H.264 '0' = not supported '1' = supported	0x1
[11:4]	MP4/H263	RO	Encoding format support, MPEG-4/H.263 '0' = not supported '1' = supported	0x2
[3:0]	JPEG	RO	Encoding format support, JPEG '0' = not supported '1' = supported	0x0
[10]	STAB	RO	Stabilization support '0' = not supported '1' = supported	0x0
[23:11]	Reserved	RO	Reserved	0x322
[10:0]	MVER	RO	Maximum video encoding resolution in macroblocks	0x0

18 Graphics Processing Unit

18.1 Graphics Processing Unit Overview

The graphics processing unit (GPU) is a high-performance graphics core, capable of rendering 2D and 3D graphics primitives. A block diagram of the complete graphics pipeline is given in Figure 18-1.

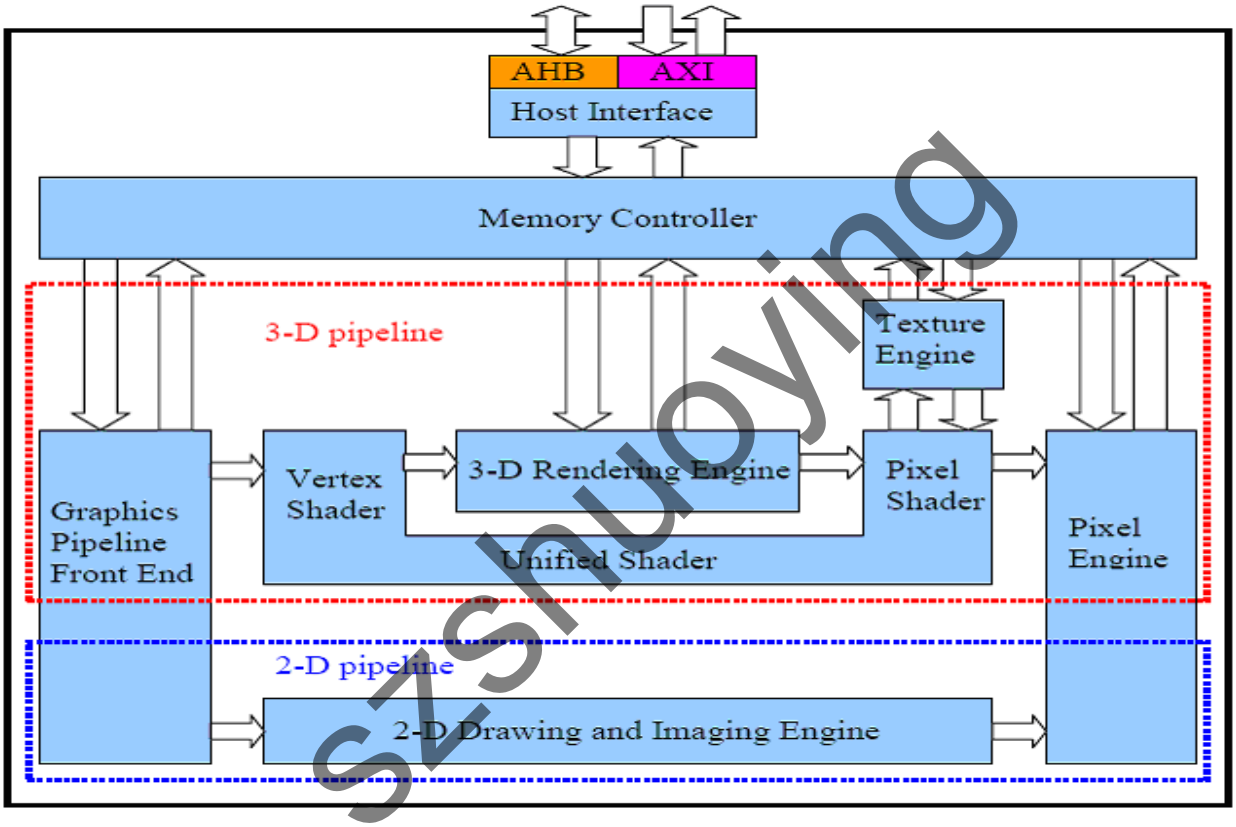


Figure 18-1: GPU Block Diagram

The main functional blocks of the GPU are:

Host Interface: It allows the GPU to communicate with external memory and host CPU through the AXI and AHB buses. In this block data crosses clock domain boundaries.

Memory Controller: Internal memory controller that arbitrates internal memory requests before sending them to external memory through host interface.

Graphics Pipeline Front End: Interprets GPU commands and send them into the graphics pipeline.

Shader (Unified): Unified SIMD processor used for vertex and pixel/fragment processing. As a vertex shader it performs geometry transformation, lighting and other computations on vertices. As a pixel/fragment shader it shades each pixel/fragment.

3-D Rendering Engine: Forms triangles, lines and points from vertices. Computes necessary parameters for primitive rasterization. Performs clipping, culling and hidden surface removal. Computes coordinates and attributes for each pixel.

Texture Engine: Fetches Texels from memory as per shades request. Performs filtering and transfers the computed values to the shaders.

Pixel Engine: Performs alpha blending, hidden surface removal, format conversion and memory read/write operations.

2D Drawing Engine: Converts primitives (rectangles and lines) to pixels before sending to pixel engine.

18.1.1 GPU Feature

Full Featured 3D Pipeline

- OpenGL ES 2.0 and OpenGL ES 1.1
- Full 32-bit floating point pipeline including shaders
- Unified vertex and pixel/fragment shader
- High-performance dependent texture operation
- Alpha blending
- Depth and stencil compare
- Support for 8 simultaneous textures
- Cubic environment texture, Projective texture and Depth texture
- Point sample, bi-linear, tri-linear and anisotropic filtering.

Full Featured 2D Pipeline

- Bit BLT & stretch BLT
- Rectangle fill and clear
- Mono expansion for text rendering
- ROP2, ROP3 and ROP4
- Alpha blending including Java 2 Porter-Duff compositing blending rules
- Supports rendering size of 32K x 32K
- 90 degree rotation
- Vertical and Horizontal mirror
- Supports up to 255 rectangles per primitive call
- Transparency by monochrome mask, chroma key or pattern mask
- Color space conversion between YUY and RGB
- Clipping
- Color Index Input conversion support
- Filter Blit
- Multi data formats support

Full Scene Anti-aliasing Support

- Support full scene anti-aliasing 2x and 4x MSAA
- Anti-aliasing algorithm achieves greater than 16x equivalent quality without additional memory and processing.

Hardware Shaders

- Unified Vertex and Fragment/Pixel Shader
- Full support of OpenGL ES 2.0 shading language
- Supports transform and lighting features of OpenGL ES 1.1
- Supports fixed function texture blending features of OpenGL ES 1.1
- IEEE 32-bit single precision floating point support
- 128-bit pipeline
- Up to 256vertex and/or fragment threads

Texture Mapping

- 1D/2D texture
- Cubic environment texture
- Projective texture
- Bump map
- Depth map up to 24-bit
- High dynamic range (HDR) textures
- Dependent texture operation with high performance
- Point sample, Bi-linear, Tri-linear and anisotropic filters
- Support 8 resident textures simultaneously
- Supports texture size up to 8k x 8k

Video Post Processing and Peripherals

- Color space conversion
- Camera preview
- High-quality image and video scalar
- Programmable kernel any size from 1x1 up to 9x9
- Programmable filter coefficient
- Supports 32 filter phases
- Programmable format conversion
- 10 standard source formats
- 7 standard destination formats

19 Memory Pool

19.1 Overview

Memory Pool(MEMPL) is a special module in IMAPx210 system for a flexible internal usage. Memory Pool is a memory concentration and includes a set of memories for external module accessing. Many functional modules can access the memories of Memory Pool directly by a direct access channel or by a system bus interface. Modules can exchange the data with each other by accessing directly and Memory Pool internal DMA.

Features

- Ten piece of 4Kx32 memory blocks
- Multi operation mode for different application
- Multi data interface arbitrating
- Video decode module direct channel
- Internal logic analyzer direct channel
- Internal DMA channel for fast memory operation
- Clock gating and power supply shutdown mode for reduce power consumption

Block Diagram

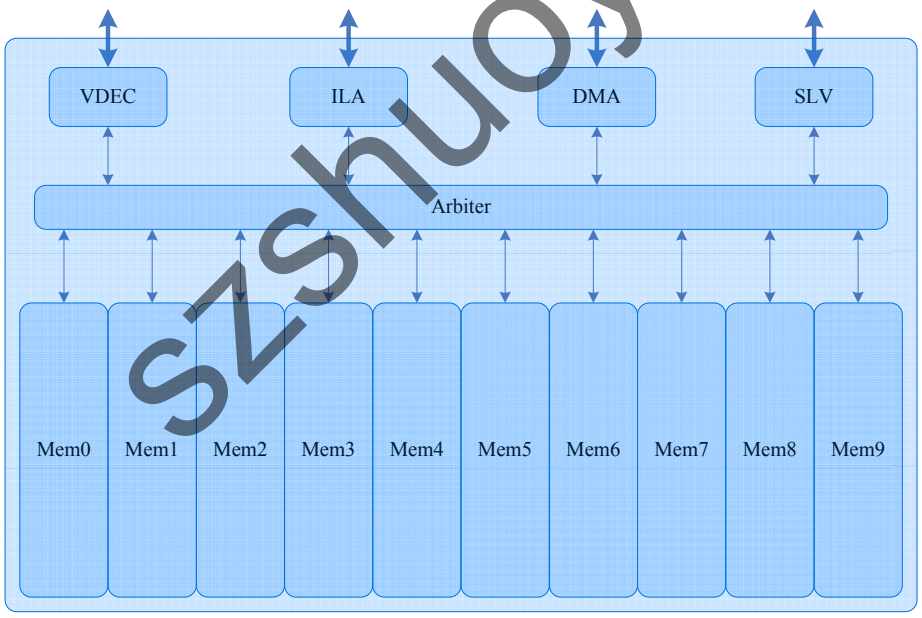


Figure 19-1: Memory Pool Block Diagram

Figure 19-1 shows the functional block diagram of memory pool， main module include data interfaces, arbiter and memory blocks. Data interface are used to connect with other function modules. Arbiter will distribute the memory blocks to different interfaces by different working mode and bus accessing order.

19.2 Functional Description

19.2.1 Interface

There are five data interfaces in memory pool and three of them are direct data channel and one is a master system bus and one is a slave system bus.

VDEC interface is a direct data interface for Video decoder module to access memory pool. It's a standard 64bit memory interface.

ILA interface is direct data interface for internal logic analyzer to access memory pool. It's a standard 128bit memory interface. Please refer the relational document of ILA if you want to know the operation detail of this port.

DMA interface is a master system bus interface. Its backend interface connects with the memory pool arbiter and master interface is an AHB interface and connect to the system bus. By configuring the control register, memory pool internal DMA can transfer the data between system memory and memory in memory pool.

SLV interface is slave system bus interface and connect with system AHB bus. Host processer can access all memory in memory pool through this interface and configure the control register of memory pool. Similarly, the other master module in system, such as GDMA, GPU and so on, can access memory pool by this slave interface. SLV interface has two address spaces. One is memory space and another is register space. Memory space is mapped to access memories of memory. Register space is mapped to access memory pool control register.

At different operation mode, these interfaces will work at different mode.

19.2.2 Operation Mode

There are four working mode with memory pool by configuring the control register.

Mode 0: Memory share mode. At this mode, except VDEC interface and ILA interface, all the other interface including DMA and SLV can access memory in memory pool. And these interfaces can access full space of memory pool.

Mode 1: VDEC mode. At this mode, memories in memory pool will be divided into two blocks. There six memories will just be accessed by video decode module and the last four memories will be access by the other interfaces except ILA.

Mode 2: Reserved mode.

Mode 3: Debug mode. At this mode, ILA module will access memory pool alone and the other interfaces cannot access memory pool.

19.2.3 DMA

There is an internal DMA module in memory pool for batch transfer. By configuring control registers, internal DMA can transfer the data between the memory pool and system memory. It will use system bus maximum efficiency to transfer data.

DMA is most used to perform memory block transfer such as memory copy, memory move and so on. For transferring data from/to system external memory (DRAM) to/from memory pool, DMA will be the best module to perform the transaction with high-speed system bus interface.

19.2.4 Address Mapping

For different data width and operation mode of different data interface, every data interface has a special address mapping format. According to these mapping formats, different interfaces can exchange data by each other.

For video decode module, usage of memory pool is special. It's not necessary to access all memories of memory pool. VDEC

just can access six memory blocks in Mode 1 and cannot access any memory in rest modes. VDEC interface has a 64bit data width and address is 64 bit aligned.

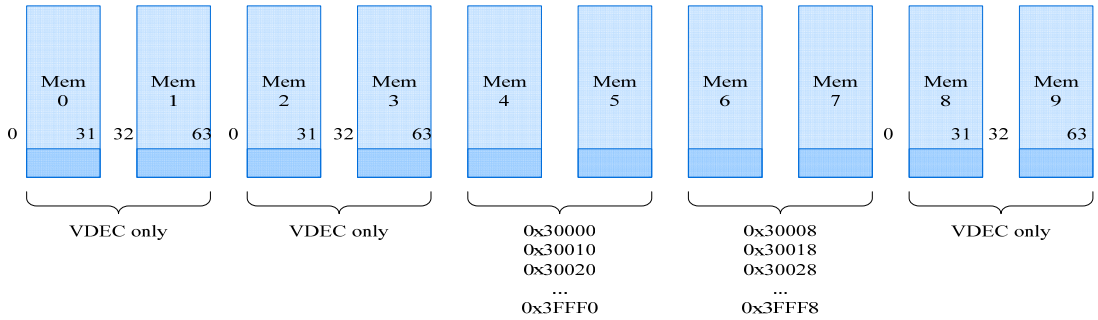


Figure 19-2: VDEC Interface Address Mapping

ILA interface just can access memories in memory pool in mode 3. In the other modes, ILA can't and needn't access the memory pool. ILA interface's data width is 128bit. For uniform data format with host processor accessing, ILA address mapping have a special order and just access eight memories of ten.

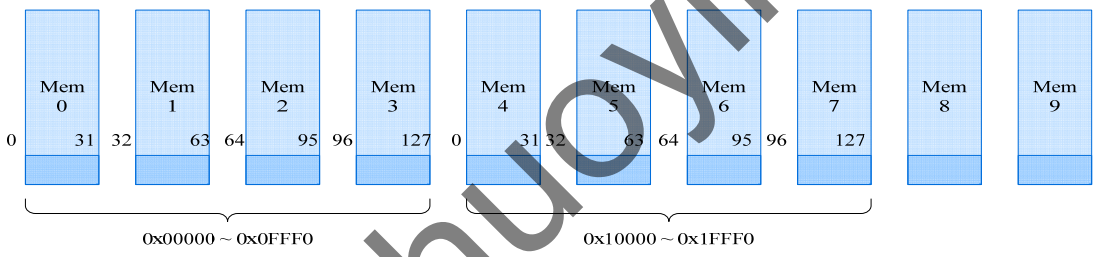


Figure 19-3: ILA Interface Address Mapping

DMA interface with memories is an internal data path. DMA normally is used to transfer the data between the system external memories, so DMA interface address mapping must be uniform with the system slave interface. Internal DMA has 64 bit data width

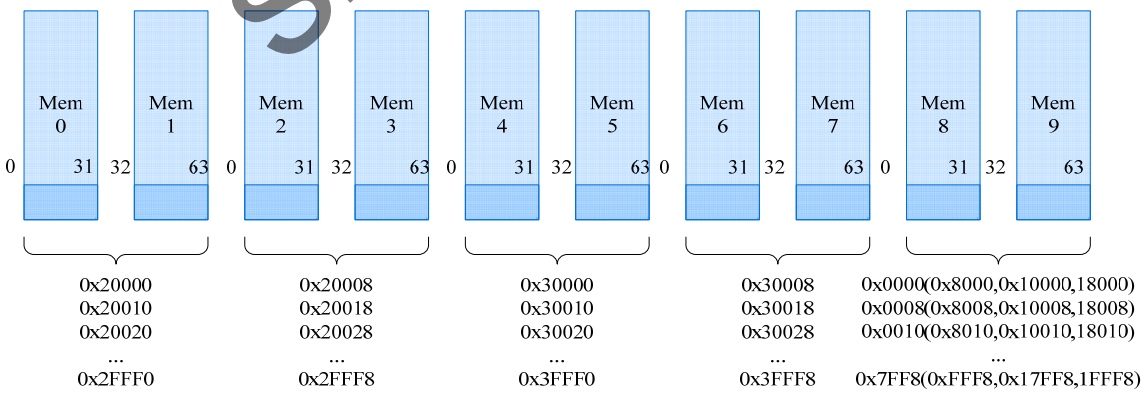


Figure 19-4: DMA Interface Address Mapping

SLV interface is used to access memory pool by all other host modules in system. Main devices include CPU, GDMAC, GPU, block devices and so on. When these devices access memory pool at same time, there will be an arbiter to decide the accessing order. SLV interface is a 64 bit data width interface and connected with several system buses.

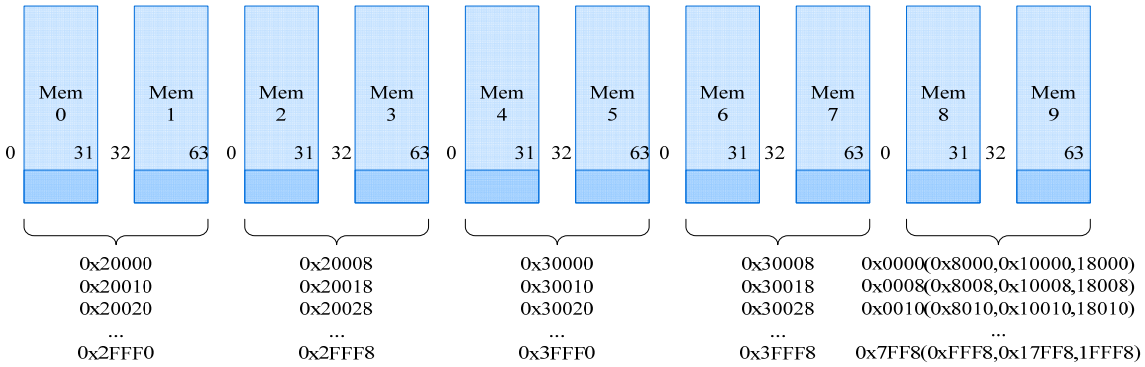


Figure 19-5: SLV Interface Address Mapping

It is just SLV interface that has a common system address in these data interfaces. Last interfaces' address is private in the modules. The SLV interface's memory space address range is 0x3FFD8000 ~ 0x3FFFFFFF. You can refer the module document for more detail of other interfaces' address and operation.

19.3 Memory Pool Register Description

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 registers: 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.

19.3.1 Memory Pool Register Memory Map

There are two type registers for memory pool, one is mode control register. Mode control register is located in system manage module and its address is part of system manage module address range. Another is internal DMA registers. It is memory pool proper address and the address range of memory is 0x20CA0000 ~ 0x 20CAFFFF. It is register space and different from SLV interface memory space.

Table 19-1: Memory Pool Mode Control Register Memory Map

Address	Symbol	Direction	Description
0x20c00120	MP_MCR	RW	Memory pool mode selection

Table 19-2: Memory Pool DMA Register Memory Map

Address	Symbol	Direction	Description
0x0000	CH_0_MADDR	RW	channel 0 memory address register
0x0004	CH_1_MADDR	RW	channel 1 memory address register
0x0008	CH_2_MADDR	RW	channel 2 memory address register
0x000C	CH_3_MADDR	RW	channel 3 memory address register
0x0010	DMA_EN	RW	DMA enable register
0x0014	INT_EN	RW	Interrupt enable register
0x0018	INT_STAT	RW	Interrupt status register
0x000C ~ 0x007C	Reserved	N/A	Reserved
0x0080	CH_0_SADDR	RW	Channel 0 system address register
0x0084	CH_0_CTRL	RW	Channel 0 control register
0x0088	CH_1_SADDR	RW	Channel 1 system address register
0x008C	CH_1_CTRL	RW	Channel 1 control register
0x0090	CH_2_SADDR	RW	Channel 2 system address register
0x0094	CH_2_CTRL	RW	Channel 2 control register
0x0098	CH_3_SADDR	RW	Channel 3 system address register
0x009C	CH_3_CTRL	RW	Channel 3 control register

19.3.2 MEMPL Registers and Field Descriptions

Register 19-1: Memory Pool Mode Control Register (MP_MCR)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1:0]	MP_SEL	RW	Memory Pool Mode Configuration. ● 00: Normal mode. Most master can access. ● 01: VDEC mode ● 10: Reserved mode ● 11: Debug Mode. ● Others: Reserved	0

Register 19-2: Memory Pool DMA Channel 0 Memory Address (CH0_MADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH0_MADDR	RW	DMA transfer address of channel 0 for internal memory space.	0

Register 19-3: Memory Pool DMA Channel 1 Memory Address (CH1_MADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH1_MADDR	RW	DMA transfer address of channel 1 for internal memory space.	0

Register 19-4: Memory Pool DMA Channel 2 Memory Address (CH2_MADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH2_MADDR	RW	DMA transfer address of channel 2 for internal memory space.	0

Register 19-5: Memory Pool DMA Channel 3 Memory Address (CH3_MADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH3_MADDR	RW	DMA transfer address of channel 3 for internal memory space.	0

Register 19-6: Memory Pool DMA Enable Register (DMA_EN)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	DMA_EN	RW	DMA transfer enable bit ● 0: disable DMA transfer ● 1: enable DMA transfer	0

Register 19-7: Memory Pool Interrupt Enable Register (INT_EN)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1]	DMA_COMP_EN	RW	DMA transfer complete interrupt enable ● 0: disable DMA transfer complete interrupt ● 1: enable DMA transfer complete interrupt This bit field just control to generate an interrupt signal to system host processor when the according interrupt status bit is active, enable/disable this bit cannot affect the interrupt status register.	0
[0]	BLK_END_EN	RW	Block transfer complete interrupt enable ● 0: disable block transfer complete interrupt ● 1: enable block transfer complete interrupt This bit field just control to generate an interrupt signal to system host processor when the according interrupt status bit is active, enable/disable this bit cannot affect the interrupt status register.	0

Register 19-8: Memory Pool Interrupt Status Register (INT_STAT)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1]	DMA_COMP_STAT	RW1C	DMA transfer complete interrupt status ● 0: DMA transfer not complete ● 1: DMA transfer complete This bit field is cleared by written 1.	0
[0]	BLK_END_STAT	RW1C	Block transfer complete interrupt status ● 0: block transfer not complete ● 1: block transfer complete This bit field is cleared by written 1.	0

Register 19-9: Memory Pool DMA Channel 0 System Address (CH0_SADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH0_SADDR	RW	DMA transfer address of channel 0 for external system space.	0

Register 19-10: DMA Channel 0 Control Register (CH0_CTRL)

Field	Symbol	Direction	Description	Default
[31]	DMAEN	RW	DMA transfer enable bit ● 0: disable DMA ● 1: enable DMA This bit must be active with DMA_EN[0] at same time, both them asserted will start DMA transfer.	0
[30]	RESET	RW	Reset DMA channel control ● 0: normal operation ● 1: reset DMA channel	0
[29:28]	WORK_FLAG	RW	Manually control the alternative channel ● 00: control channel 0 to transfer ● 01: control channel 1 to transfer ● 10: control channel 2 to transfer	0

			● 11: control channel 3 to transfer	
[27]	ALTEN	RW	Alternative transfer control, decide whether start the DMA channel 1 transfer ● 0: normal operation ● 1: alternate enable	0
[26]	AUTOLOAD	RW	Automatic load address control，decide either complete transfer or start again automatically when transfer finish ● 0: normal mode ● 1: auto load mode	0
[25]	DIR	RW	DMA transfer direction ● 0: transfer data from external system memory to Memory Pool internal integrate memory ● 1: transfer data from Memory Pool internal integrate memory to external system memory	0
[24:0]	LENGTH	RW	DMA transfer length	0

Register 19-11: Memory Pool DMA Channel 1 System Address (CH1_SADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH1_SADDR	RW	DMA transfer address of channel 1 for external system space.	0

Register 19-12: DMA Channel 1 Control Register (CH1_CTRL)

Field	Symbol	Direction	Description	Default
[31]	END_MODE	RW	Indicate the endianess mode ● 0: little endian mode ● 1: big endian mode	0
[30:28]	Reserved	N/A	Reserved	0
[27]	ALTEN	RW	Alternative transfer control, decide whether start the DMA channel 2 transfer ● 0: normal operation ● 1: alternate enable	0
[26]	AUTOLOAD	RW	Automatic load address control，decide either complete transfer or start again automatically when transfer finish ● 0: normal mode ● 1: auto load mode	0
[25]	DIR	RW	DMA transfer direction ● 0: transfer data from external system memory to internal integrate memory ● 1: transfer data from internal integrate memory to external system memory	0
[24:0]	LENGTH	RW	DMA transfer length	0

Register 19-13: Memory Pool DMA Channel 2 System Address (CH2_SADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH2_SADDR	RW	DMA transfer address of channel 2 for external system space.	0

Register 19-14: DMA Channel 2 Control Register (CH2_CTRL)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27]	ALTEN	RW	Alternative transfer control, decide whether start the DMA channel 3 transfer ● 0: normal operation ● 1: alternate enable	0
[26]	AUTOLOAD	RW	Automatic load address control，decide either complete transfer or start again automatically when transfer finish	0

			● 0: normal mode ● 1: auto load mode	
[25]	DIR	RW	DMA transfer direction ● 0: transfer data from external system memory to internal integrate memory ● 1: transfer data from internal integrate memory to external system memory	0
[24:0]	LENGTH	RW	DMA transfer length	0

Register 19-15: Memory Pool DMA Channel 3 System Address (CH3_SADDR)

Field	Symbol	Direction	Description	Default
[31:0]	CH3_SADDR	RW	DMA transfer address of channel 3 for external system space.	0

Register 19-16: DMA Channel 3 Control Register (CH3_CTRL)

Field	Symbol	Direction	Description	Default
[31:28]	Reserved	N/A	Reserved	0
[27]	ALTEN	RW	Alternative transfer control, decide whether start the DMA channel 0 transfer ● 0: normal operation ● 1: alternate enable	0
[26]	AUTOLOAD	RW	Automatic load address control, decide either complete transfer or start again automatically when transfer finish ● 0: normal mode ● 1: auto load mode	0
[25]	DIR	RW	DMA transfer direction ● 0: transfer data from external system memory to internal integrate memory ● 1: transfer data from internal integrate memory to external system memory	0
[24:0]	LENGTH	RW	DMA transfer length	0

20 AC97

20.1 Overview

The AC97 Controller Unit of the IMAPx210 supports AC97 revision 2.1 features. AC97 Controller communicates with AC97 Codec using an audio controller link (AC-link). Controller sends the stereo PCM data to Codec. The external digital-to-analog converter (DAC) in the Codec then converts the audio sample to an analog audio waveform. Also, the Controller receives the stereo PCM data from the Codec and then stores them in the memories. This chapter describes the programming model for the AC97 Controller Unit. The information in this chapter requires an understanding of the AC97 revision 2.1 specifications

Features

- AC97 revision 2.1 specifications compatible
- Independent channels for stereo PCM In, stereo PCM Out
- DMA-based operation and interrupt based operation.
- All of the channels support 16/18/20-bit samples.
- Variable sampling rate AC97 Codec interface (48 KHz and below)
- 20-bit, 16 entry FIFOs per channel
- Only Primary CODEC support

Block Diagram

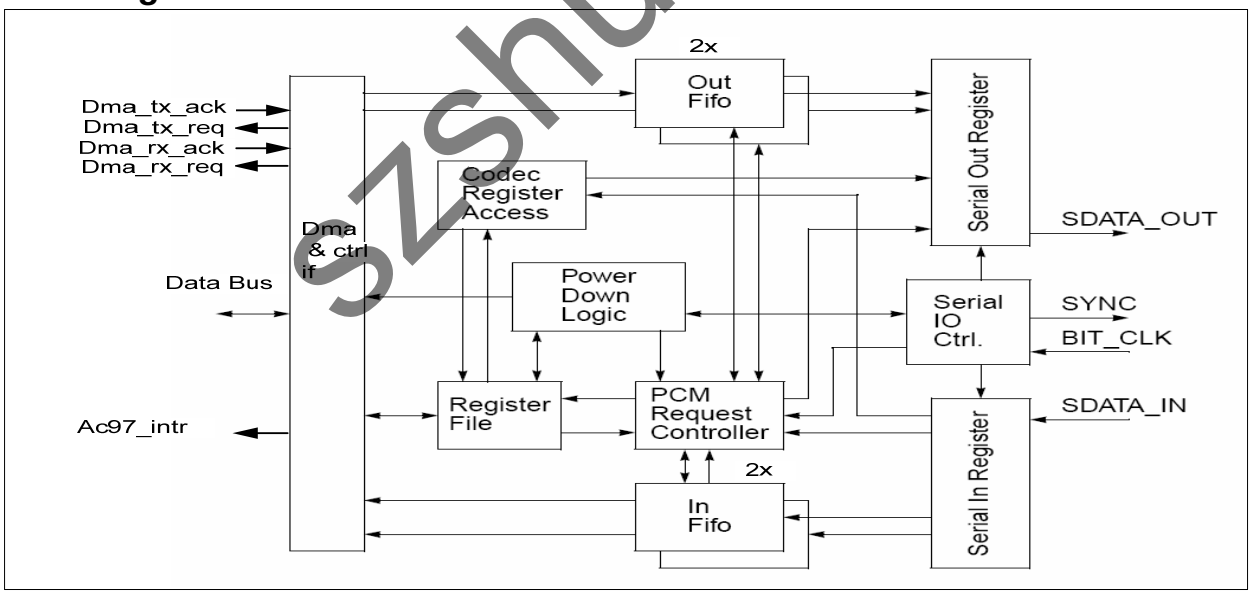


Figure 20-1: AC97 Block Diagram

Figure 20-1 shows the functional block diagram of the IMAPx210 AC97 Controller. The AC97 signals form the AC-link, which is a point-to-point synchronous serial interconnect that supports full-duplex data transfers. All digital audio streams and command/status information are communicated over the AC-link.

20.2 AC97 Controller Operation

Internal Data Path and Format

Figure 20-2 shows the internal data path of the IMAPx210 AC97 Controller. It has stereo Pulse Code Modulated (PCM) In and Stereo PCM Out, which consist of 20-bit, 16 entries buffer. Also it has a 20-bit I/O shift register via AC-link.

Figure 20-3 shows the sample data format in buffer at three different bit sample resolutions.

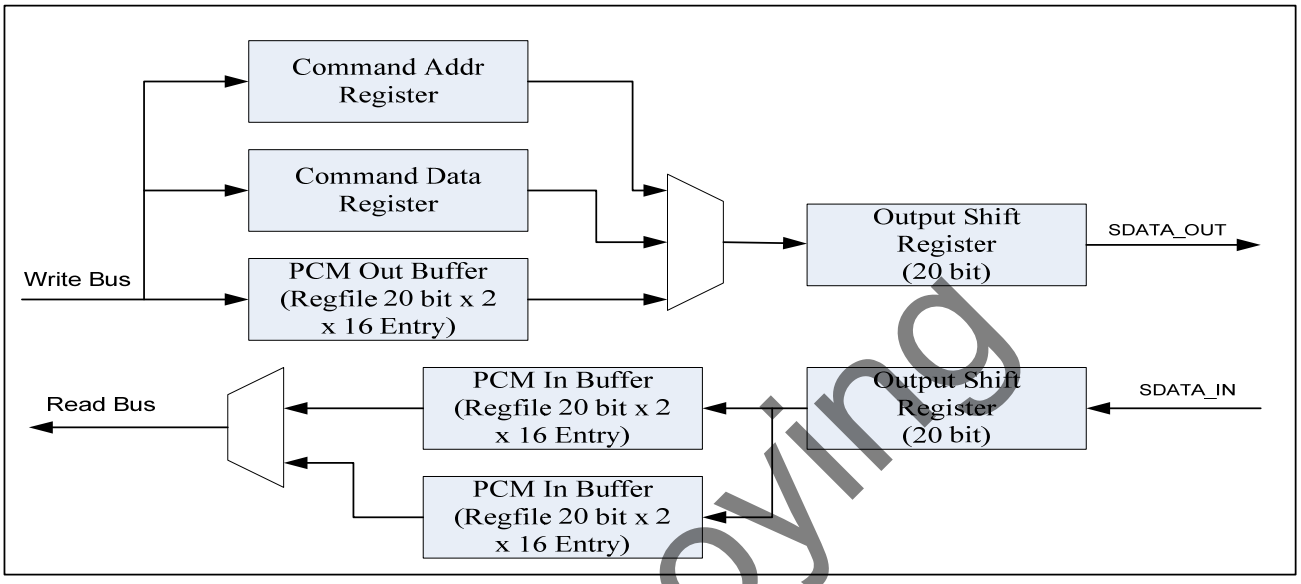


Figure 20-2: Internal Data Path

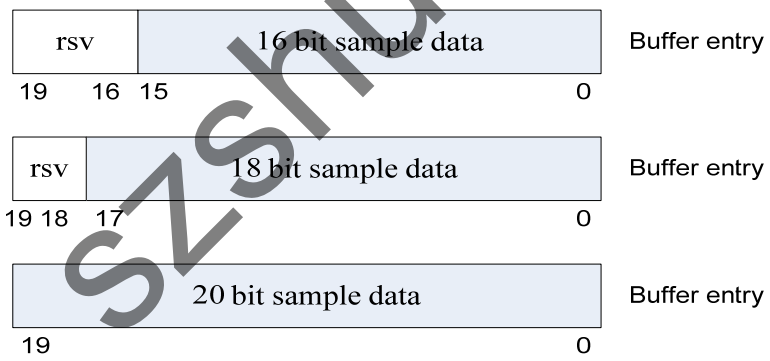


Figure 20-3: Sample Data Format

Operation Flow Chart

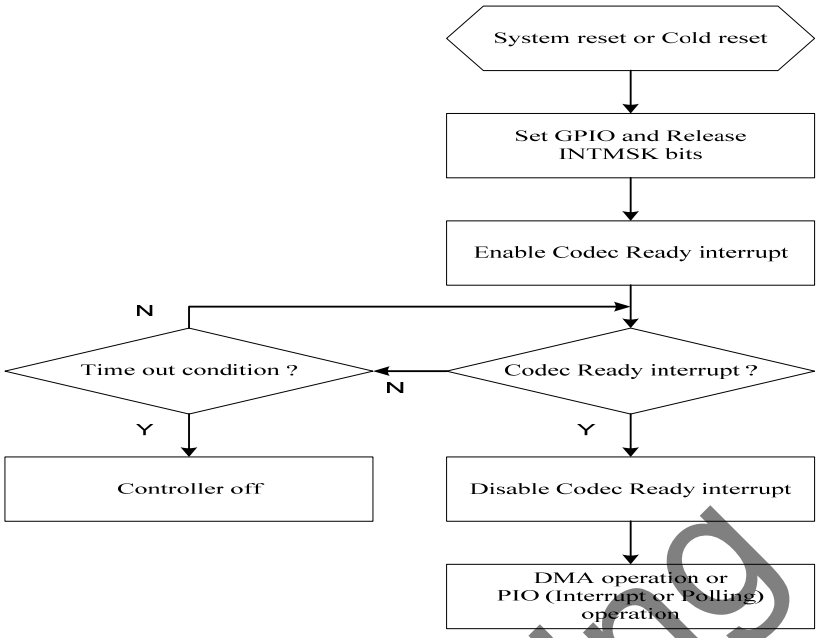


Figure 20-4: AC97 Operation Flow Chart

AC-Link Digital Interface Protocol

Each AC97 Codec incorporates a five-pin digital serial interface that links it to the IMAPx210 AC97 Controller, which is illustrated in Figure 20-5. The AC-link is a full-duplex, fixed-clock, PCM digital stream. It employs a time division multiplexing (TDM) scheme to handle control register access 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 a 20-bit sample resolution and requires a DAC and an analog-to-digital converter (ADC) with a minimum of 16-bit resolution.

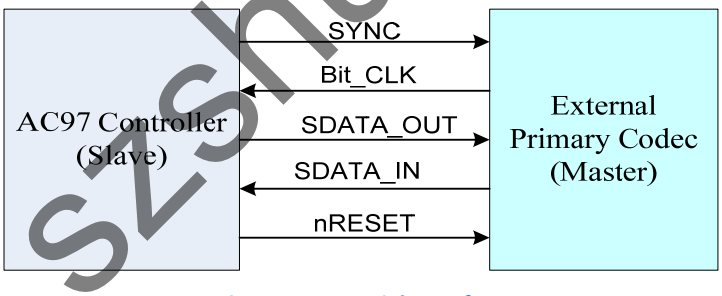


Figure 20-5: AC-Link Interface

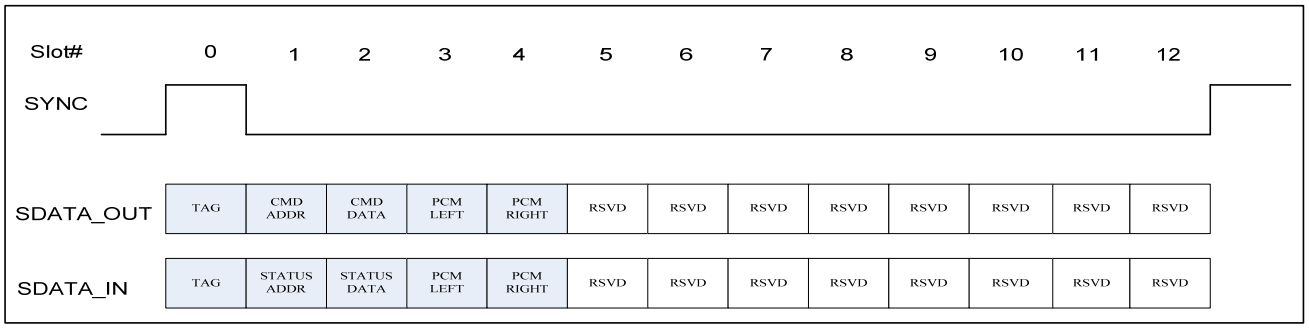


Figure 20-6: Bi-directional AC-link Frame with Slot Assignments

Figure 20-6 shows the slot definitions supported by the AC97 Controller. The 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 as Tag Phase and it is 16-bits long. The remaining 12 time slots are called as Data Phase. The Tag Phase contains a bit that identifies a valid frame and 12-bits that identify the time slots in the Data Phase that contain a valid data. Each time slot in the Data Phase is 20-bits long. A frame begins when the SYNC goes high. The amount of time the 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, BIT_CLK. The controller and the CODEC use the SYNC and BIT_CLK to determine when to send the transmit data and when to sample the received data. A transmitter transitions the serial data stream on each rising edge of BIT_CLK and a receiver samples the serial data stream on falling edges of BIT_CLK. 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 from MSB to LSB. The Tag Phase's first bit is bit 15 and the first bit of each slot in the Data Phase is bit 19. The last bit in any slot is bit 0.

AC-Link Output Frame (SDATA_OUT)

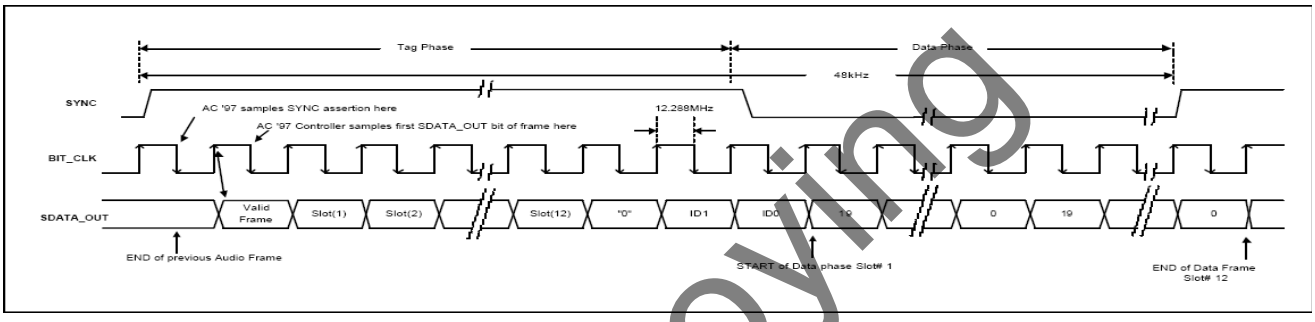


Figure 20-7: AC-link Output Frame

AC-Link Input Frame (SDATA_IN)

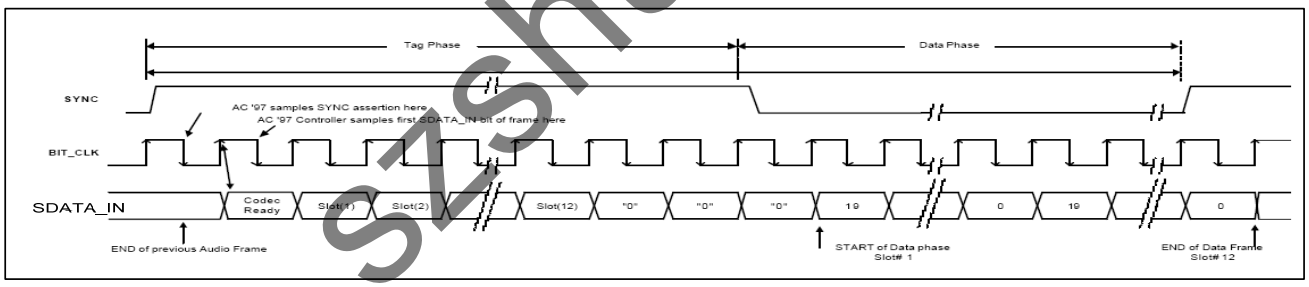


Figure 20-8: AC-link Input Frame

AC97 Powerdown

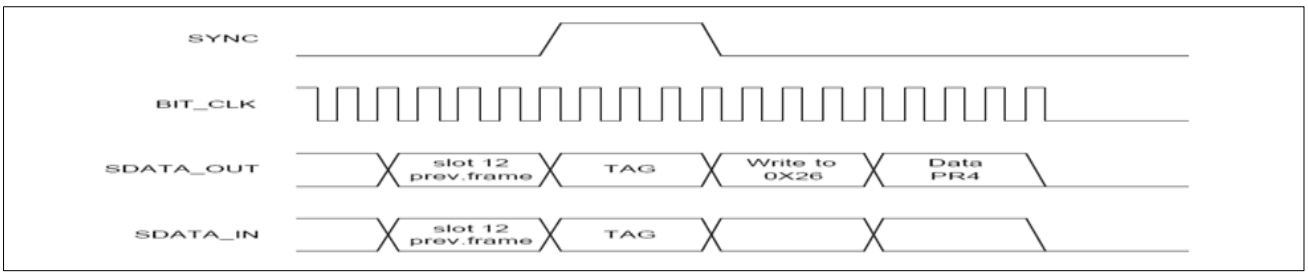


Figure 20-9: AC97 Powerdown Timing Diagram

Powering Down the AC-link

The AC-link signals enter a low power mode when the AC97 CODEC Powerdown register (0x26) bit PR4 is set to 1 (i.e. by writing 0x1000). Then the Primary CODEC drives both the BITCLK and SDATA_IN to a logic low voltage level. The sequence follows the timing diagram shown above in the Figure 20-9.

The AC97 Controller transmits the write to Powerdown 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 Powerdown 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 BIT_CLK and SDATA_IN to a logic low level. The AC97 Controller drives the 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 is provided for 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 time 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).

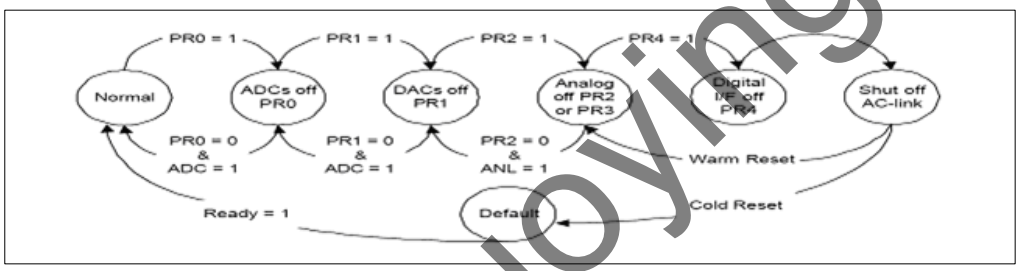


Figure 20-10: AC97 Power Down/Power Up Flow

Cold AC97 Reset

A cold reset is generated when an nRESET pin is asserted through the AC_GLBCTRL. Asserting and de-asserting nRESET activates the BIT_CLK and SDATA_OUT. All the 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 BIT_CLK is absent and SYNC is driven high. In normal audio frames, SYNC is a synchronous AC97 input. When BIT_CLK is absent, SYNC is treated as an asynchronous input used to generate a warm reset to AC97.The AC97 Controller must not activate BIT_CLK until it samples the SYNC to low again. This prevents a new audio frame from being falsely detected.

Time Reference

AC97 Controller has one time reference register for AC-link Power Down and Warm Up, you must configure it firstly.

20.3 AC97 Register Description

20.3.1 AC97 Register Memory Map

Table 20-1: AC97 Register Memory Map

Address	Symbol	Direction	Description
0x20DE_0000	AC_GLBCTRL	R/W	AC97 global control register
0x20DE_0004	AC_GLBSTAT	RC	AC97 global status register
0x20DE_0008	AC_CODEC_CMD	R/W	AC97 codec command register
0x20DE_000C	AC_CODEC_STAT	R	AC97 codec status register
0x20DE_0010	AC_OFS	R	AC97 PCM out channel FIFO status
0x20DE_0014	AC_IFS	R	AC97 PCM in channel FIFO status
0x20DE_0020	AC_PCMDR	RW	AC97 PCM Data Register, PCM out channel data(write), PCM in channel data(read)
0x20DE_0034	AC_TIMER	R/W	AC97 Timer reference Register

20.3.2 AC97 Registers and Field Descriptions

Register 20-1: AC97 Global Control Register （AC_GLBCTRL）

Field	Symbol	Direction	Description	Default
[29]	CRRD_INTREN	R/W	Codec register read done interrupt enable bit 0: disable 1: enable	0
[28]	CWRD_INTREN	R/W	Codec register write done interrupt enable bit 0: disable 1: enable	0
[22]	CRDY_INTREN	R/W	Codec ready interrupt enable bit 0: disable 1: enable	0
[21]	POUR_INTREN	R/W	PCM out channel FIFO underrun (FIFO is empty) interrupt enable bit 0: disable 1: enable	0
[20]	PIOV_INTREN	R/W	PCM in channel FIFO overrun (FIFO is full) interrupt enable bit 0: disable 1: enable	0
[18]	POTH_INTREN	R/W	PCM out channel FIFO threshold (FIFO is half empty) interrupt enable bit 0: disable 1: enable	0
[17]	PITH_INTREN	R/W	PCM in channel FIFO threshold (FIFO is half full) interrupt enable bit 0: disable 1: enable	0
[13:12]	PCMO_TM	R/W	PCM out channel transfer mode configuration bits 00: Off 01: PIO 10: DMA 11: Reserved	00
[11:10]	PCMI_TM	R/W	PCM in channel transfer mode configuration bits 00: Off 01: PIO 10: DMA 11: Reserved	00
[6]	SMRT	R/W	Sample rate configuration bit. 0: fixed sample rate(48KHz) 1: variable sample rate(below 48KHz)	0
[5:4]	SMSIZE	R/W	Sample bit data resolution configuration bits. 00: 16 bit 01: 18 bit	00

			10: 20 bit	
[3]	CHDTEN	R/W	PCM channel data transfer enable bit. Only this field set, PCM in/out channel FIFO data will be transmit/received to/from AC-link. 0: disable 1: enable	0
[2]	ACLINKON	R/W	AC-Link set up bit. 0: Off 1: SYNC signal transfer to Codec	0
[1]	WMRST	R/W	Warm Reset enable bit. 0: Normal 1: Wake up codec from power down (this is a self clearing bit)	0
[0]	CDRST	R/W	Cold Reset enable bit. 0: Normal 1: Reset Codec and Controller logic (this is a self clearing bit)	0

Register 20-2: AC97 Global Status Register （AC_GLBSTAT）

Field	Symbol	Direction	Description	Default
[29]	CRRD_INTR	RC	Codec register read done interrupt, indicate controller had finish reading corresponding Codec Register and can be read in AC_CODEC_CMD. 0: Not requested 1: Requested	0
[28]	CWRD_INTR	RC	Codec register write done interrupt, indicate controller had finish writing data to corresponding Codec Register. 0: Not requested 1: Requested	0
[22]	CRDY_INTR	RC	Codec ready interrupt 0: Not requested 1: Requested	0
[11]	POUR_INTR	RC	PCM out channel FIFO underrun (FIFO is empty) interrupt 0: Not requested 1: Requested	0
[10]	PIOV_INTR	RC	PCM in channel FIFO overrun (FIFO is full) interrupt 0: Not requested 1: Requested	0
[9]	POTH_INTR	RC	PCM out channel FIFO threshold (FIFO is half empty) interrupt 0: Not requested 1: Requested	0
[8]	PITH_INTR	RC	PCM in channel FIFO threshold (FIFO is half full) interrupt 0: Not requested 1: Requested	0
[2:0]	AC97_STAT	R	AC97 Controller State 000: IDLE 011: Active, AC97 normal transceiver 100: LP, AC97-Link Powerdown Else: Reserved	000

Register 20-3: AC97 Codec Status Register （AC_CODEC_STAT）

Field	Symbol	Direction	Description	Default
[22:16]	CMD_ADDR	R	CODEC command address	
[15:0]	CRDY_INTR	R	CODEC command data	0

Note: If you want to read data from AC97 codec register via the AC_CODEC_STAT register, you should follow these steps.

1. Write command address and data on the AC_CODEC_CMD register with Bit [23] =1.
2. Have a enough delay time or wait for CRRD_INTR interrupt.
3. Read command address and data from AC_CODEC_STAT register.

Register 20-4: AC97 Codec Command Register （AC_CODEC_CMD）

Field	Symbol	Direction	Description	Default
[23]	CMD_TYPE	R/W	Codec command operation type 0: command write 1: status read **Note**: When the commands are written on the AC_CODDEC_CMD register, it is recommended that the delay time between the command and the next command is more than 1/48 KHz. Also, the delay time of two successive status read should more than 2/48 KHz. Safely, you can use Codec Register Write/Read done interrupt to configure external codec.	0

Register 20-5: AC97 PCM Out Channel FIFO Status Register （AC_OFS）

Field	Symbol	Direction	Description	Default
[9:5]	RFREE	R	The free space of data in PCM out right channel FIFO.	0x10
[4:0]	LFREE	R	The free space of data in PCM out left channel FIFO	0x10

Register 20-6: AC97 PCM In Channel FIFO Status Register （AC_IFS）

Field	Symbol	Direction	Description	Default
[9:5]	RNUM	R	The number of data in PCM in right channel FIFO.	0x0
[4:0]	LNUM	R	The number of data in PCM in left channel FIFO	0x0

Register 20-7: AC97 PCM Data Register （AC_PCMDR）

Field	Symbol	Direction	Description	Default
[19:0]	PCMDATA	R/W	PCM out/in channel FIFO data register. Write: PCM data in PCM out channel FIFO, the order of write data must be Left Data, Right Data, Left Data, Right Data and so on. Read: PCM data in PCM in channel FIFO, the order of return read data is Left Data, Right Data, Left Data, Right Data and so on.	0x0

Register 20-8: AC97 Timer Reference Register （AC_TIMER）

Field	Symbol	Direction	Description	Default
[13:8]	LPDET	R/W	AC-link Powerdown Time Reference parameter (count of PCLK), the minimum of time is 163ns. **Note** : If PCLK = 100MHz, LPDET must larger than 17(163ns/10ns);	0x5
[5:0]	WPDET	R/W	AC-link Warm Up Time Reference parameter (count of PCLK), the minimum of time is 260ns. **Note** : If PCLK = 100MHz, WPDET must larger than 26(260ns/10ns);	0x6

21 IIS

21.1 Overview

Currently, many digital audio systems are attracting the consumers on the market, in the form of compact discs, digital audio tapes, digital sound processors, and digital TV sound. The IMAPx210 Inter-IC Sound (IIS) bus interface can be used to implement a CODEC interface to an external multiple bit-resolutions stereo audio CODEC IC for mini-disc and portable applications. The IIS bus interface supports IIS bus data format. It can transmit and receive data simultaneously as well as transmit or receive data alternatively at a time.

Features

- Full duplex communication due to the independence of transmitter and receiver
- I2S transmitter and/or receiver based on the Philips I2S serial protocol
- Master mode only, support codec clock output
- Audio data resolutions of 8, 12, 16, 20, 24, and 32 bits
- SCLK Gating for multiple clock cycle : 8, 10, 12, 14, 16, 18, 20, 24 and 32
- Independent Transmit and receive buffers(16 depth)
- Programmable FIFO thresholds
- Interrupt, polled-mode or DMA mode operation
- Configurable Clock source of Clock Generator

Block Diagram

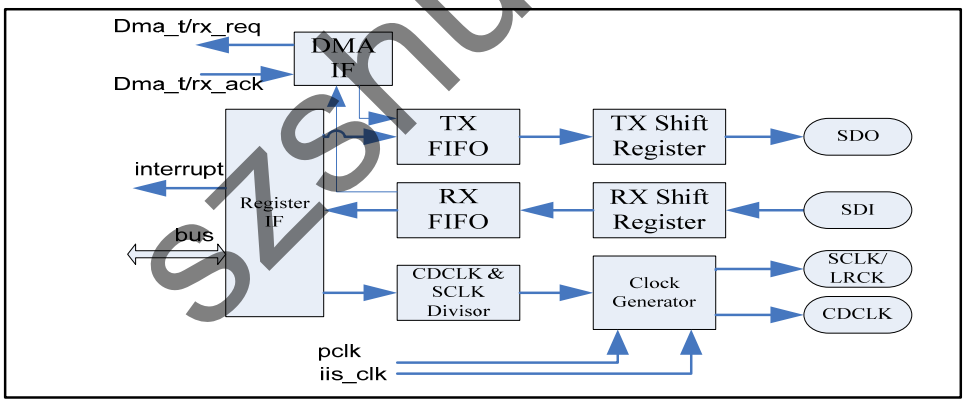


Figure 21-1: IIS Block Diagram

21.2 Audio Serial Interface Format

IIS-Bus Format

The IIS bus has four lines including serial data input (SDI), serial data output (SDO), left/right channel select(LRCK), and serial bit clock (SCLK); the device generating LRCK and SCLK is the master.

Serial data is transmitted in 2's complement with the MSB first. The MSB is transmitted first because the transmitter and receiver may have different word lengths. The transmitter does not have to know how many bits the receiver can handle, nor does the receiver need to know how many bits are being transmitted.

When the system word length is greater than the transmitter word length, the word is truncated (least significant data bits are set to '0') for data transmission. If the receiver gets more bits than its word length, the bits after the LSB are ignored. On the other hand, if the receiver gets fewer bits than its word length, the missing bits are set to zero internally. And therefore, the MSB has 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 whenever the LRCK is changed.

The word select line LRCK is used to time the multiplexed data streams. For instance, when LRCK is low, the word being transferred is left stereo data; when LRCK is high, the word being transferred is right stereo data. This format is illustrated in Figure 21-2. For standard I2S formats, the MSB of a word is sent one SCLK cycle after a LRCK change. Serial data sent by the transmitter can be synchronized with either the negative edge or positive edge of the SCLK signal. However, the receiver must latch the serial data on the rising edge of SCLK.

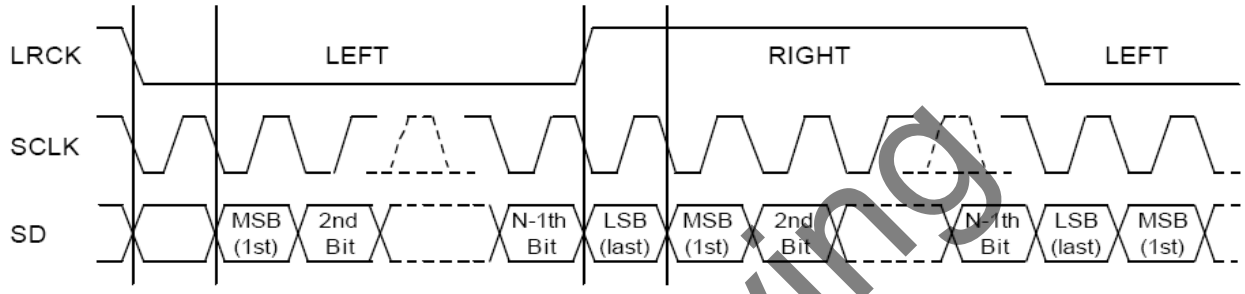


Figure 21-2: I2S Stereo Frame Format

21.3 IIS Operation Modes

IIS Enable

You must enable the IIS component before any data can be received or transmitted into the FIFOs. To enable the component, set the I2S Enable (IEN) bit of the I2S Enable Register (IER) to 1. When you disable the device, it acts as a global disable. To disable IIS, set IER[0] to 0.

After disable, the following events occur:

- TX and RX FIFOs are cleared, and read/write pointers are reset;
- Any data in the process of being transmitted or received is lost;
- All other programmable enables (such as transmitter/receiver block enables and TX/RX channel enables) in the component are overridden;

IIS always starts in the left stereo data cycle (LRCK = 0), and one SCLK cycle later transitions to the right stereo data cycle (LRCK = 1). This allows for half a frame of SCLK to write data to the TX FIFOs and to ensure that any connected slave receivers do not miss the start of the data frame (for instance, the LRCK 1-to-0 transition) once the SCLK restarts.

On reset, the IER[0] is set to 0 (disable).

IIS as Transmitter

Stereo data pairs (such as, left and right audio data) written to a TX channel via the APB bus are shifted out serially on the appropriate serial data out line (SDO). The shifting is timed with respect to the serial clock (SCLK) and the word select line (LRCK).

The basic usage flow of transmitter is illustrated in followed figure.

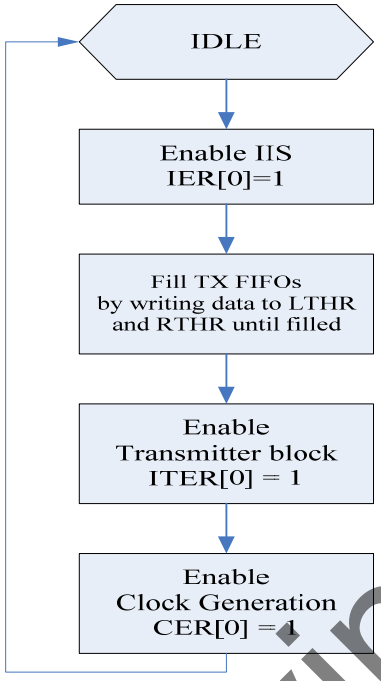


Figure 21-3: Basic Usage Flow – IIS as Transmitter

Transmitter Block Enable

To enable the transmitter block, set ITER[0] to 1. To disable the block, set ITER[0] to 0.

When the transmitter block is disabled, the following events occur:

- Outgoing data is lost and the channel outputs are held low;
- Data in the TX FIFOs are pReserved and the FIFOs can be written to;
- Any previous programming (like changes in word size, threshold levels, and so on) of the TX channels is pReserved; and
- TX channel enables are overridden.

  When the transmitter block is enabled, if there is data in the TX FIFOs, the channel resumes transmission on the next left stereo data cycle (such as when the LRCK line goes low).

When the block is disabled, you can perform any of the following procedures:

- Program (or further program) TX channel registers
- Flush the TX FIFOs by programming the Transmitter FIFOs Reset bit of the Transmitter FIFO Flush Register (TXFFR[0] = 1)

On reset, the ITER[0] is set to 0 (disable).

Transmit Channel Enable

Transmit channel enable/disable is controlled by bit 0 of the Transmitter Enable Register.

When a TX channel is disabled, the following occurs:

- Outgoing stereo data is lost;
- Channel output is held low;
- Data in the TX FIFO is pReserved, and the FIFO can be written to; and

- Any previous programming of the TX channel's registers is pReserved, and the registers can be further reprogrammed.

When TX channel is disabled, you can flush the channel's TX FIFO by programming the Transmit Channel FIFO Reset (TXCHFR) bit of the Transmit FIFO Flush (TFF[0] = 1). When the TX channel is enabled, if there is data in the TX FIFO, the channel resumes transmission on the next left stereo data cycle (such as, when the LRCK line goes low).

On reset, the TFF[0] is set to 1 (enable).

Transmit Channel Audio Data Resolution

TX channel is initially configured with a maximum audio data resolution (32 bits), and can be reprogrammed during operation to any supported audio data resolution that is less than 32 bits, such as 8, 12, 16, 20 or 24 bits.

Reprogramming of the audio resolution ensures that the MSB of the data is still transmitted first if the resolution of the data to be sent is reduced. Changes to the resolution are programmed via the Word Length (WLEN) bits of the Transmitter Configuration Registers (TCR[2:0]). The channel must be disabled prior to any resolution changes.

Transmit Channel FIFOs

Transmit Channel has two FIFO banks for left and right stereo data. The depth of FIFOs is 16(FIFO_DEPTH). The FIFO width is 32 bits (maximum audio data resolution).

There are several ways to clear the TX FIFOs and reset the read/write pointers as described as follows;

- on reset
- by disabling IIS (IER[0] = 0)
- by flushing the transmitter block (TXFFR[0] = 1)
- by flushing TX channel (TFF[0] = 1)

You must disable the transmitter block/channel before the transmitter block and channel FIFO can be flushed.

The TX FIFO Empty Threshold Trigger Level can be set to any value in the range of 0 to FIFO_DEPTH-1, which correspond to trigger levels of 1 to FIFO_DEPTH (for example, Trigger Level = Configured Value + 1). When this level is reached, a transmit channel empty interrupt is generated. This level can be reprogrammed during operation by writing to the Transmit Channel Empty Trigger (TXCHET) bits of the Transmit FIFO Configuration Register (TFCR [3:0]).

You must disable the TX channel prior to changing the trigger level.

Transmit Channel Interrupts

TX channel generates two interrupts: TX FIFO Empty and Data Overrun.

- TX FIFO Empty interrupt – This interrupt is asserted when the empty trigger threshold level for the TX FIFO is reached. A TX FIFO Empty interrupt is cleared by writing data to the TX FIFO to bring its level above the empty trigger threshold level for the channel.
- Data Overrun interrupt –This interrupt is asserted when an attempt is made to write to a full TX FIFO (any data being written is lost while data in the FIFO is pReserved). A Data Overrun interrupt is cleared by reading the Transmit Channel Overrun (TXCHO) bit [0] of the Transmit Overrun Register.

The interrupt status of TX channel can be determined by polling the Interrupt Status Register (ISR). The TXFE bit [4] indicates the status of the TX FIFO Empty interrupt, while the TXFO bit [5] indicates the status of the Data Overrun interrupt.

Both the TX FIFO Empty and Data Overrun interrupts can be masked off by writing a 1 in the Transmit Empty Mask (TXFEM) and Transmit Overrun Mask (TXFOM) bits of the Interrupt Mask Register (IMR), respectively. This prevents the interrupts from driving their output lines, however, the ISR always shows the current status of the interrupts regardless of any masking.

Writing to a Transmit Channel

The stereo data pairs to be transmitted by TX channel are written to the TX FIFOs via the Left Transmit Holding Register

(LTHR) and the Right Transmit Holding Register (RTHR). All stereo data pairs must be written using the following two-stage process:

1. Write left stereo data to LTHR
2. Write right stereo data to RTHR.

You must write stereo data to the device in this order, otherwise, the interrupt and status lines values will be invalid, and the left/right stereo pairs might be transmitted out of sync.

When IIS is enabled, if the TX FIFO is empty and data is not written to the FIFOs before the next left cycle, the channel outputs zeros for a full frame (left and right cycle). Transmission only commences if there is data in the TX FIFO prior to the transition to the left data cycle. In other words, if the start of the frame is missed, the channel output idles until the next available frame.

Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and a Data Overrun interrupt being generated.

IIS as Receiver

Stereo data pairs (such as, left and right audio data) are received serially from a data input line (SDI). These data words are stored in RX FIFOs until they are read via the APB bus. The receiving is timed with respect to the serial clock (SCLK) and the word select line (LRCK).

The basic usage flow of receiver is illustrated in followed figure.

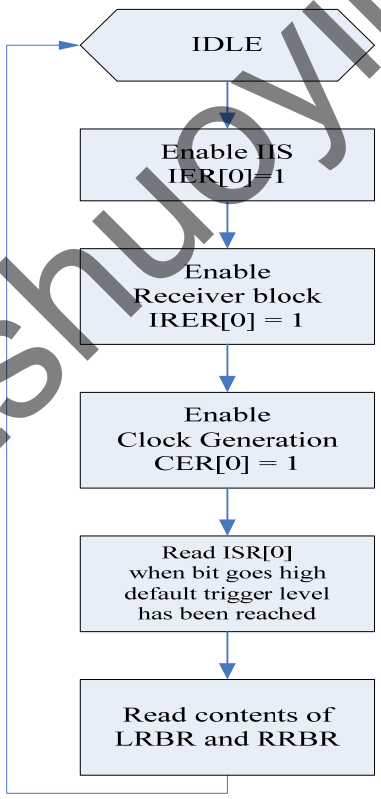


Figure 21-4: Basic Usage Flow – IIS as Receiver

Receiver Block Enable

To enable the receiver block, set IRER[0] to '1.' To disable the block, set this bit to '0.'

When the receiver block is disabled, the following events occur:

- Incoming data is lost;
- Data in the RX FIFOs is pReserved and the FIFOs can be read;
- Any previous programming (such as changes in word size, threshold levels, and so on) of the RX channels is pReserved; and
- RX channel enable is overridden. Enabling the channel resumes receiving on the next left stereo data cycle (for instance, when LRCK goes low).

When the block is disabled, you can perform any of the following procedures:

- Program (or further program) the RX channel registers;
- Flush the RX FIFOs by programming the Receiver FIFOs Reset (RXFR) bit of the Receiver FIFO Flush Register (RXFFR[0] = 1).
- Flush channel's RX FIFO by programming the Receive Channel FIFO Reset (RXCHFR) bit of the Receive FIFO Flush Register (RFF [0] = 1).

On reset, IRER[0] is set to 0 (disable).

Receive Channel Enable

RX channel enable/disable is controlled by bit 0 of the Receiver Enable Register (RER[0])

When the RX channel is disabled, the following occurs:

- Incoming data is lost;
- Data in the RX FIFO is pReserved;
- FIFO can be read;
- Previous programming of the RX channel is pReserved; and
- RX channel can be further programmed.

When the RX channel or block is disabled, you can flush the channel's RX FIFO by writing 1 in bit 0 of the Receive FIFO Flush Register (RFF). When the channel is enabled, it resumes receiving on the next left stereo data cycle (for instance. when LRCK line goes low).

On reset, the RFF[0] is set to 1 (enable).

Receive Channel Audio Data Resolution

RX channel is initially configured with a maximum audio data resolution (32 bits), and can be reprogrammed during operation to any supported audio data resolution that is less than 32 bits, such as 8, 12, 16, 20 or 24 bits.

Reprogramming of the audio resolution ensures that the LSB of the received data is placed in the LSB position of the RX FIFO if the resolution of the data being received is reduced. Changes to the resolution are programmed via the Word Length (WLEN) bits of the Receive Configuration registers (RCR[3:0]). The channel must be disabled prior to any resolution changes.

The RX channel also supports unknown data resolutions. If the received word is greater than the configured channel resolution, the least significant bits are ignored. If the received word is less than the configured/programmed channel resolution, the least significant bits are padded with zeros.

Receive Channel FIFOs

Receive Channel has two FIFO banks for left and right stereo data. The depth of FIFOs is 16(FIFO_DEPTH). The FIFO width is 32 bits (maximum audio data resolution).

The RX FIFOs can be cleared and the read/write pointers reset in a number ways, as described as follows:

- on reset

- by disabling IIS (IER[0] = 0)
- by flushing the receiver block (RXFFR[0] = 1)
- by flushing an individual RX channel (RFF[0] = 1)

Before you flush the receiver block or channel, you must disable the receiver block or channel.

The RX FIFO Data Available Trigger Level can be set to any value in the range of 0 to FIFO_DEPTH-1, which correspond to trigger levels of 1 to FIFO_DEPTH (for example, Trigger Level = Configured Value + 1). When this level is reached, a RX channel data available interrupt is generated. This level can be reprogrammed during operation via the Receive Channel Data Trigger (RXCHDT) bits of the Receive FIFO Configuration Register (RFCR[3:0]). The RX channel needs to be disabled prior to any changes in the trigger level.

Receive Channel Interrupts

RX channel generates two interrupts: RX FIFO Data Available and Data Overrun.

- RX FIFO Data Available interrupt – This interrupt is asserted when the trigger level for the RX FIFO is reached. This interrupt is cleared by reading data from the RX FIFO until its level drops below the data available trigger level for the channel.
- Data Overrun interrupt – This interrupt is asserted when an attempt is made to write received data to a full RX FIFO (any data being written is lost while data in the FIFO is pReserved). This interrupt is cleared by reading the Receive Channel Overrun (RXCHO) bit [0] of the Receive Overrun Register (ROR).

The interrupt status of any RX channel can be determined by polling the Interrupt Status Register (ISR). The RXDA bit [0] indicates the status of the RX FIFO Data Available interrupt; the RXFO bit [1] indicates the status of the RX FIFO Data Overrun interrupt.

Both the Receive Empty Threshold and Data Overrun interrupts can be masked by writing a 1 in the Receive Empty Threshold Mask (RDM) and Receive Overrun Mask (ROM) bits of the Interrupt Mask Register (IMR), respectively. This prevents the interrupts from driving their output line, however, the ISR always shows the current status of the interrupts regardless of any masking.

Reading from a Receive Channel

The stereo data pairs received by RX channel are written to the left and right RX FIFOs. These FIFOs can be read via the Left Receive Buffer Register (LRBR) and the Right Receive Buffer Register (RRBR). All stereo data pairs must be read using the following two-stage process:

1. Read the left stereo data from LRBR.
2. Read the right stereo data from RRBR.

You must read the stereo data in this order to avoid the status and interrupt lines becoming out of sync.

Clock Generation

Clock Generation Enable

The Clock Generation Enable (CLKEN) bit of the Clock Enable Register (CER) enables and disables the master mode clock signals: LRCK, SCLK. To enable these signals, set CER[0] to 1; to disable them, set this bit to 0, in which case LRCK and SCLK are held low.

When the CLKEN bit is disabled, any incoming or outgoing data is lost. However, data already in the RX and TX FIFOs are pReserved. After this bit is enabled, transmission recommences at the start of the next stereo frame.

On enabling CER[0], LRCK always starts in the left stereo data cycle (LRCK = 0). One SCLK cycle later, it transitions to the right stereo data cycle (LRCK = 1); hence—a 0-to-1 transition. This allows for half a frame of sclks to write data to the TX FIFOs and ensures that any connected slave receivers do not miss the start of the data frame (the LRCK 1-to-0 transition) once the sclk restarts.

SCLK Gating

When IIS is configured as a master and the audio data resolution of the receive and transmit channels is less than the current word select size, the sclk can be gated off for the remainder of the left/ right cycle, as illustrated in Figure 21-5. This can be reprogrammed during operation of the component by setting the SCLK Gating (SCLKG) bits [2:0] of the Clock Configuration Register (CCR).

The Clock Generation Block must be disabled prior to any changes to the sclk gating value. The actual gating of the sclk needs to be done externally by AND'ing the generated sclk_gate signal with sclk.

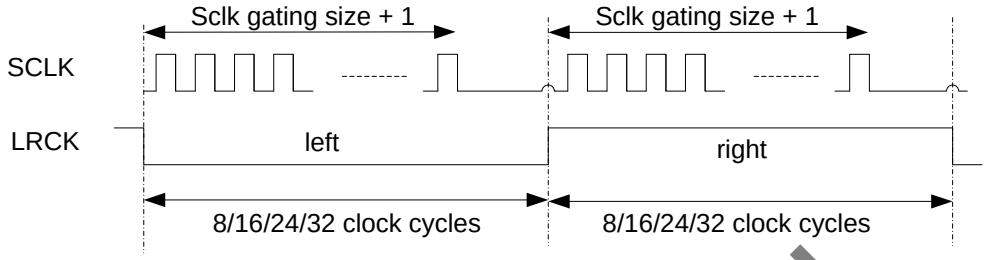


Figure 21-5: Gating of SCLK

21.4 IIS Register Description

21.4.1 IIS Register Memory Map

Table 21-1: IIS Register Memory Map

Address	Symbol	Direction	Description
0x20DD_0000	IER	R/W	IIS Enable Register
0x20DD_0004	IRER	R/W	IIS Receiver Block Enable Register
0x20DD_0008	ITER	R/W	IIS Transmitter Block Enable Register
0x20DD_000C	CER	R/W	IIS Clock Enable Register
0x20DD_0010	CCR	R/W	IIS Clock Configuration Register
0x20DD_0014	RXFFR	W	IIS Receiver Block FIFO Register
0x20DD_0018	TXFFR	W	IIS Transmitter Block FIFO Register
0x20DD_001C	CDR	R/W	IIS Clock Divisor Register
0x20DD_0020	LRBR	R	IIS Left Receive Buffer
0x20DD_0020	LTHR	W	IIS Left Transmit Holding Register
0x20DD_0024	RRBR	R	IIS Right Receive Buffer
0x20DD_0024	RTHR	W	IIS Right Transmit Holding Register
0x20DD_0028	RER	R/W	IIS Receive Enable Register
0x20DD_002C	TER	R/W	IIS Transmit Enable Register
0x20DD_0030	RCR	R/W	IIS Receive Configuration Register
0x20DD_0034	TCR	R/W	IIS Transmit Configuration Register
0x20DD_0038	ISR	R	IIS Interrupt Status Register
0x20DD_003C	IMR	R/W	IIS nterrupt Mask Register
0x20DD_0040	ROR	R	IIS Receive Overrun Register
0x20DD_0044	TOR	R	IIS Transmit Overrun Register
0x20DD_0048	RFCR	R/W	IIS Receive FIFO Configuration Register
0x20DD_004C	TFCR	R/W	IIS Transmit FIFO Configuration Register
0x20DD_0050	RFF	W	IIS Receive FIFO Flush
0x20DD_0054	TFF	W	IIS Transmit FIFO Flush
0x20DD_01C0	RXDMA	R	IIS Receive Block DMA Register
0x20DD_01C4	RRXDMA	W	IIS Reset Receiver Block DMA Register

0x20DD_01C8	TXDMA	W	IIS Transmitter Block DMA Register
0x20DD_01CC	RTXDMA	W	IIS Reset Transmitter Block DMA Register

21.4.2 IIS Registers and Field Descriptions

Register 21-1: IIS Enable Register （IER）

Field	Symbol	Direction	Description	Default
[0]	IEN	R/W	IIS global enable/disable bit 1: IIS is enabled 0: IIS is disabled	0

Register 21-2: IIS Receiver Block Enable Register （IRER）

Field	Symbol	Direction	Description	Default
[0]	RXEN	R/W	Receiver block enable. A disable on this bit overrides any individual receive channel enables. 1: enable receiver 0: disable receiver	0

Register 21-3: IIS Transmitter Block Enable Register （ITER）

Field	Symbol	Direction	Description	Default
[0]	TXEN	R/W	Transmitter block enable. A disable on this bit overrides any individual transmit channel enables. 1: enable transmitter 0: disable transmitter	0

Register 21-4: IIS Clock Enable Register （CER）

Field	Symbol	Direction	Description	Default
[0]	CLKEN	R/W	Clock generation enable/disable. This bit enables/disables the clock generation signals: SCLK, LRCK. 1: enable 0: disable	0

Register 21-5: IIS Clock Configuration Register （CCR）

Field	Symbol	Direction	Description	Default
[4:3]	WSS	R/W	These bits are used to program the number of sclk cycles for which the word select line (LRCK) stays in the left or right sample mode: 0: 16 clock cycles 1: 24 clock cycles 2: 32 clock cycles 3: 8 clock cycles The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value.	0
[2:0]	SCLKG	R/W	These bits are used to program the gating of sclk: 0: No clock gating 1: Gate after 12 clock cycles 2: Gate after 8 clock cycles 3: Gate after 20 clock cycles 4: Gate after 24 clock cycles 5: Gate after 18 clock cycles 6: Gate after 10 clock cycles 7: Gate after 14 clock cycles The programmed gating value should be less than or equal to the largest configured/programmed audio resolution to prevent the truncating of RX/TX data.	0

			The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value.	

Register 21-6: IIS Receiver Block FIFO Reset Register （RXFFR）

Field	Symbol	Direction	Description	Default
[0]	RXFFR	W	Receiver FIFO Reset. Writing a 1 to this register flushes all the RX FIFOs (this is a self clearing bit). Receiver Block must be disabled prior to writing this bit.	0

Register 21-7: IIS Transmitter Block FIFO Reset Register （TXFFR）

Field	Symbol	Direction	Description	Default
[0]	TXFFR	W	Transmitter FIFO Reset. Writing a 1 to this register flushes all the TX FIFOs (this is a self clearing bit). The Transmitter Block must be disabled prior to writing this bit.	0

Register 21-8: IIS Clock Divisor Register （CDR）

Field	Symbol	Direction	Description	Default
[27:26]	CDCLKSEL	R/W	Select CDCLK generator source clock(sel_cdclk) 00 : pclk 01 : iis_clk, which is generated in system clock generator. Else : Reserved	0
[25:24]	SCLKSEL	R/W	Select SCLK generator source clock(sel_sclk) 00 : pclk 01 : iis_clk, which is generated in system clock generator Else : Reserved	0
[23:16]	CDCLKDIV	R/W	CDCLK divisor value(>0) determine CDCLK clock rate. CDCLK = sel_cdclk / (CDCLKDIV + 1)	0
[15:0]	SCLKDIV	R/W	SCLK divisor value(>0) determine SCLK clock rate. SCLK = sel_sclk / (SCLKDIV + 1)	0

Register 21-9: IIS Left Receive Buffer Register （LRBR）

Field	Symbol	Direction	Description	Default
[31:0]	LRBR	R	The left stereo data received serially from the receive channel input (SDI) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBR followed by a read from RRBR) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs **Note:** Before reading this register again, the right stereo data MUST be read from RRBR, or the status/interrupts will not be valid.	0

Register 21-10: IIS Left Transmit Holding Register （LTHR）

Field	Symbol	Direction	Description	Default
[31:0]	LTHR	W	The left stereo data to be transmitted serially through the transmit channel output (SDO) is written through this register. Writing is a two-stage process: (1) A write to this register passes the left stereo sample to the	0

			transmitter. (2) This MUST be followed by writing the right stereo sample to the RTHR register. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.	

Register 21-11: IIS Right Receive Buffer Register （RRBR）

Field	Symbol	Direction	Description	Default
[31:0]	RRBR	R	The right stereo data received serially from the receive channel input (SDI) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBR followed by a read from RRBR) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is pReserved.) NOTE: Prior to reading this register, the left stereo data MUST be read from LRBR, or the status/interrupts will not be valid.	0

Register 21-12: IIS Right Transmit Holding Register （RTHR）

Field	Symbol	Direction	Description	Default
[31:0]	RTHR	W	The right stereo data to be transmitted serially through the transmit channel output (SDO) is written through this register. Writing is a two-stage process: (1) A left stereo sample MUST first be written to the LTHR register. (2) A write to this register passes the right stereo sample to the transmitter. Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.	0

Register 21-13: IIS Receive Enable Register （RER）

Field	Symbol	Direction	Description	Default
[0]	RXCHEN	R/W	Receive channel enable. On enable, the channel begins receiving on the next left stereo cycle. A global disable of DW_apb_i2s (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value. 1: Enable 0: Disable	0

Register 21-14: IIS Transmit Enable Register （TER）

Field	Symbol	Direction	Description	Default
[0]	TXCHEN	R/W	Transmit channel enable. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of DW_apb_i2s (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. 1: Enable 0: Disable	0

Register 21-15: IIS Receive Configuration Register （RCR）

Field	Symbol	Direction	Description	Default
[2:0]	WLEN	R/W	These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR (or RRBR) register. 000 = 8 bit resolution 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Else = Reserved	0

Register 21-16: IIS Transmit Configuration Register （TCR）

Field	Symbol	Direction	Description	Default
[2:0]	WLEN	R/W	These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. 000 = 8 bit resolution 001 = 12 bit resolution 010 = 16 bit resolution 011 = 20 bit resolution 100 = 24 bit resolution 101 = 32 bit resolution Else = Reserved	0

Register 21-17: IIS Interrupt Status Register （ISR）

Field	Symbol	Direction	Description	Default
[5]	TXFO	R	Status of Data Overrun interrupt for the TX channel. Attempt to write to full TX FIFO. 0: TX FIFO write valid 1: TX FIFO write overrun	0
[4]	TXFE	R	Status of Transmit Empty Trigger interrupt. TX FIFO is empty 1: trigger level reached 0: trigger level not reached	0
[1]	RXFO	R	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. 0: RX FIFO write valid 1: RX FIFO write overrun	0
[0]	RXDA	R	Status of Receive Data Available interrupt. RX FIFO data available. 1: trigger level reached 0: trigger level not reached	0

Register 21-18: IIS Interrupt Mask Register （IMR）

Field	Symbol	Direction	Description	Default
[5]	TXFOM	R/W	Masks TX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt	0
[4]	TXFEM	R/W	Masks TX FIFO Empty interrupt. 1: masks interrupt 0: unmasks interrupt	0
[1]	RXFOM	R/W	Masks RX FIFO Overrun interrupt. 1: masks interrupt 0: unmasks interrupt	0
[0]	RXDAM	R/W	Masks RX FIFO Data Available interrupt. 1: masks interrupt 0: unmasks interrupt	0

Register 21-19: IIS Receive Overrun Register （ROR）

Field	Symbol	Direction	Description	Default
[0]	RXCHO	R	Read this bit to clear the RX FIFO Data Overrun interrupt. 0: RX FIFO write valid 1: RX FIFO write overrun	0

Register 21-20: IIS Transmit Overrun Register （TOR）

Field	Symbol	Direction	Description	Default
[0]	TXCHO	R	Read this bit to clear the TX FIFO Data Overrun interrupt. 0: TX FIFO write valid 1: TX FIFO write overrun	0

Register 21-21: IIS Receive FIFO Configuration Register （RFCR）

Field	Symbol	Direction	Description	Default
[3:0]	RXCHDT	R/W	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1	0

Register 21-22: IIS Transmit FIFO Configuration Register （TFCR）

Field	Symbol	Direction	Description	Default
[3:0]	TXCHET	R/W	Transmit Channel Empty Trigger. These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Trigger Level = TXCHET	0

Register 21-23: IIS Receive FIFO Flush Register （RFF）

Field	Symbol	Direction	Description	Default
[0]	RXCHFR	W	Receive Channel FIFO Reset. Writing a 1 to this register flushes RX FIFO. (This is a self clearing bit.) RX channel or block must be disabled prior to writing to this bit.	0

Register 21-24: IIS Transmit FIFO Flush Register （TFF）

Field	Symbol	Direction	Description	Default
[0]	TXCHFR	W	Transmit Channel FIFO Reset. Writing a 1 to this register flushes TX FIFO. (This is a self clearing bit.) TX channel or block must be disabled prior to writing to this bit.	0

Register 21-25: IIS Receiver Block DMA Register （RXDMA）

The RXDMA register allows access to receive channel via a single point rather than through the LRBR and RRBR registers. The order of returned read data: Left Data, Right Data, Left Data, Right Data and so on.

Field	Symbol	Direction	Description	Default
[31:0]	RXDMA	R	Receiver Block DMA Register. Used to cycle repeatedly reading stereo data pairs.	0

Register 21-26: IIS Reset Receiver Block DMA Register （RRXDMA）

The RRXDMA register can be written to at any stage of the RXDMA's read cycle, however, it has no effect when the component is in the middle of a stereo pair read.

The order of returned read data:

1. Left Data,
2. Right Data,
3. Left Data,
4. RRXDMA Reset - No effect (read not complete)
5. Right Data,
6. RRXDMA Reset
7. Left Data,
8. Right Data

Field	Symbol	Direction	Description	Default
[0]	RRXDMA	W	Reset Receiver Block DMA Register. Write 1 enable reset and self-clearing.	0

Register 21-27: IIS Transmitter Block DMA Register （TXDMA）

The TXDMA register functions similar to the RXDMA register and allows write accesses to transmit channel via a single point rather than through the LTHR and RTHR registers.

Field	Symbol	Direction	Description	Default
[31:0]	TXDMA	W	Transmitter Block DMA Register	0

Register 21-28: IIS Reset Transmitter Block DMA Register （RTXDMA）

The RTXDMA register provides the same functionality as the RRXDMA register but targets TXDMA instead.

Field	Symbol	Direction	Description	Default
[0]	RTXDMA	W	Reset Transmitter Block DMA Register. Write 1 enable reset and self-clearing.	0

22 I2C

22.1 Overview

IMAPx210 I2C bus is a two-wire serial interface, consisting of a serial data line (SDA) and a serial clock (SCL). These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. Devices can also be considered as masters or slaves when performing data transfers. A master is a device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

I2C bus interface requires external hardware components as support in order to be compliant in an I2C system. It must also be noted that the I2C should only be operated either as (but not both):

- A sole master in an I2C system and programmed only as a Master; OR
- A slave in an I2C system and programmed only as a Slave.

I2C can operate in standard mode (with data rates up to 100 Kb/s), fast mode (with data rates up to 400 Kb/s), and high-speed mode (with data rates up to 3.4 Mb/s). The I2C can communicate with devices only of these modes as long as they are attached to the bus. Additionally, high-speed mode and fast mode devices are downward compatible. For instance, high-speed mode devices can communicate with fast mode and standard mode devices in a mixed speed bus system; fast mode devices can communicate with standard mode devices in 0 to 100 Kb/s I2C bus system. However, standard mode devices are not upward compatible and should not be incorporated in a fast-mode I2C bus system as they cannot follow the higher transfer rate and unpredictable states would occur.

Features

- Two independent I2C bus interface
- Two-wire I2C serial interface (SDA and SCL)
- Three speeds:
  - Standard mode (100 Kb/s)
  - Fast mode (400 Kb/s)
  - High-speed mode (3.4 Mb/s)
- Clock synchronization
- Master or Slave I2C operation
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Bulk transmit mode
- Transmit and receive buffers(16 bytes)
- Interrupt or polled-mode operation
- Handles Bit and Byte waiting at all bus speeds

Block Diagram

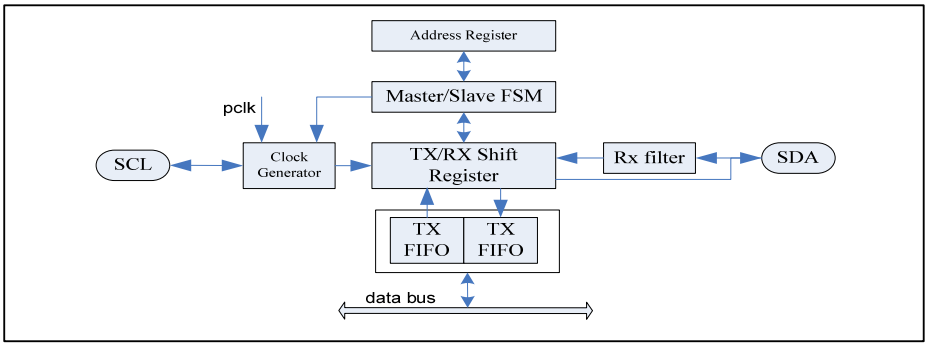


Figure 22-1: I2C Block Diagram

22.2 I2C Terminology

The following terms are used throughout this manual and are defined as follows:

I2C Bus Terms

The following terms relate to how the role of the I2C device and how it interacts with other I2C devices on the bus.

Transmitter – the device that sends data to the bus. A transmitter can either be a device that initiates the data transmission to the bus (a *master-transmitter*) or responds to a request from the master to send data to the bus (a *slave-transmitter*).

Receiver – the device that receives data from the bus. A receiver can either be a device that receives data on its own request (a master-receiver) or in response to a request from the master (a slave-receiver).

Master – the component that initializes a transfer (START command), generates the clock (SCL) signal and terminates the transfer (STOP command). A master can be either a transmitter or a receiver.

Slave – the device addressed by the master. A slave can be either receiver or transmitter.

These concepts are illustrated in the followed figure.

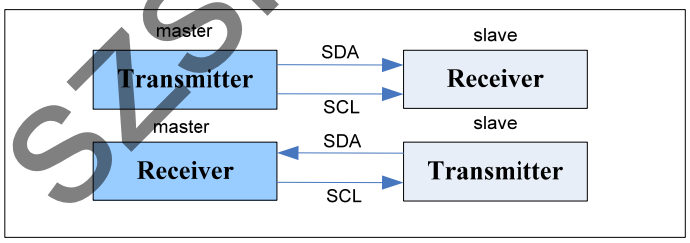


Figure 22-2: Master/Slave and Transmitter/Receiver Relationships

Multi-master – the ability for more than one master to co-exist on the bus at the same time without collision or data loss.

Arbitration – the predefined procedure that authorizes only one master at a time to take control of the bus.

Synchronization – the predefined procedure that synchronizes the clock signals provided by two or more masters.

SDA – data signal line (Serial DAta)

SCL – clock signal line (Serial CLock)

Bus Transfer Terms

The following terms are specific to data transfers that occur to/from the I2C bus.

START (**RESTART**) – data transfer begins with a START or RESTART condition. The level of the SDA data line changes from high to low, while the SCL clock line remains high. When this occurs, the bus becomes busy.

STOP – data transfer is terminated by a STOP condition. This occurs when the level on the SDA data line passes from the low state to the high state, while the SCL clock line remains high. When the data transfer has been terminated, the bus is free or idle once again. The bus stays busy if a RESTART is generated instead of a STOP condition.

22.3 I2C Behavior

IMAPx210 I2C can be controlled via software to be either:

- The sole I2C master only, communicating with other I2C slaves; or
- An I2C slave only, communicating with one more I2C masters.

The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The acknowledgement of data is sent by the device that is receiving data, which can be either a master or a slave. As mentioned previously, the I2C protocol also allows multiple masters to reside on the I2C bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledge (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in following figure.

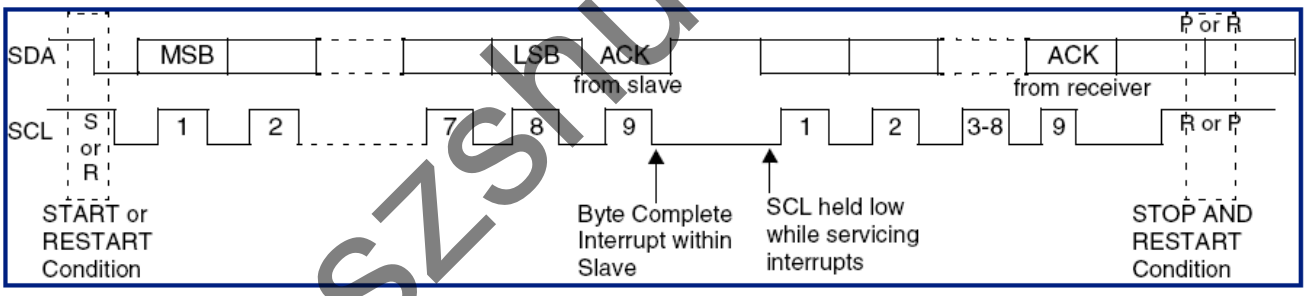


Figure 22-3: Data Transfer on the I2C Bus

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain to perform wire-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.

22.4 I2C Protocols

START and STOP Conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master wants to terminate the transmission, the master issues a STOP condition. This is defined to be a low-to-high transition of the SDA line while SCL is 1. Figure 22-4 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

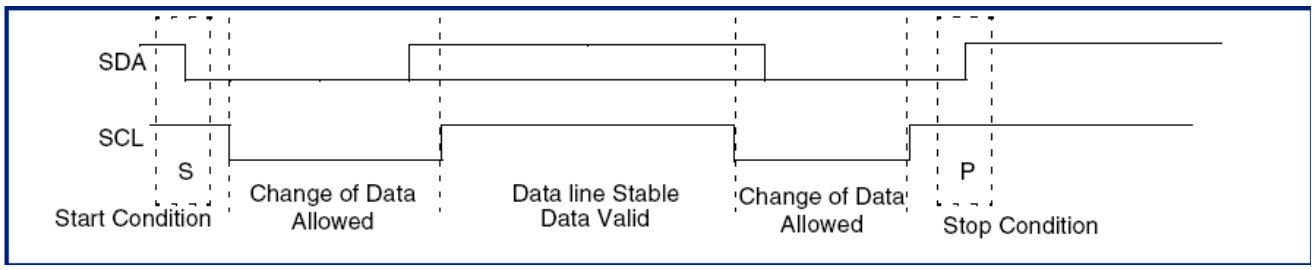


Figure 22-4: START and STOP Condition

Addressing Slave Protocol

There are two address formats: the 7-bit address format and the 10-bit address format.

7-bit Address Format

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in Figure 22-5. When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads from the slave.

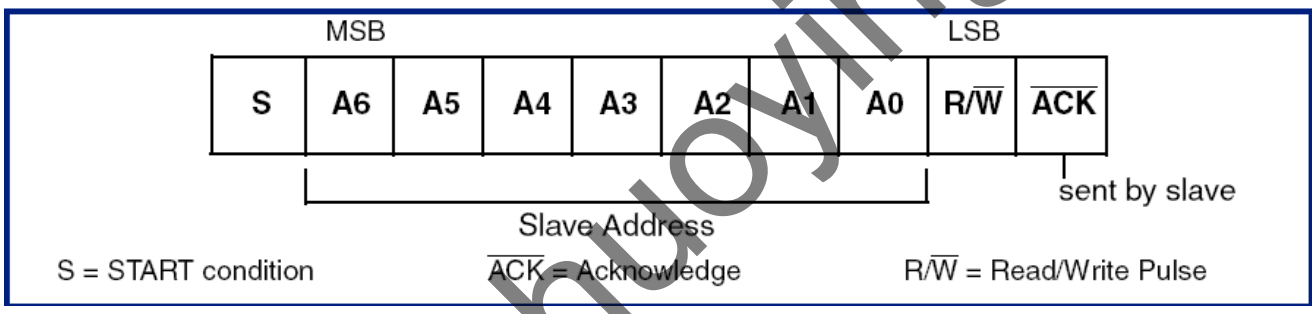


Figure 22-5: 7-bit Address Format

10-bit Address Format

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (bit 0) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address. The followed figure shows the 10-bit address format, and Table 22-1 defines the special purpose and Reserved first byte addresses.

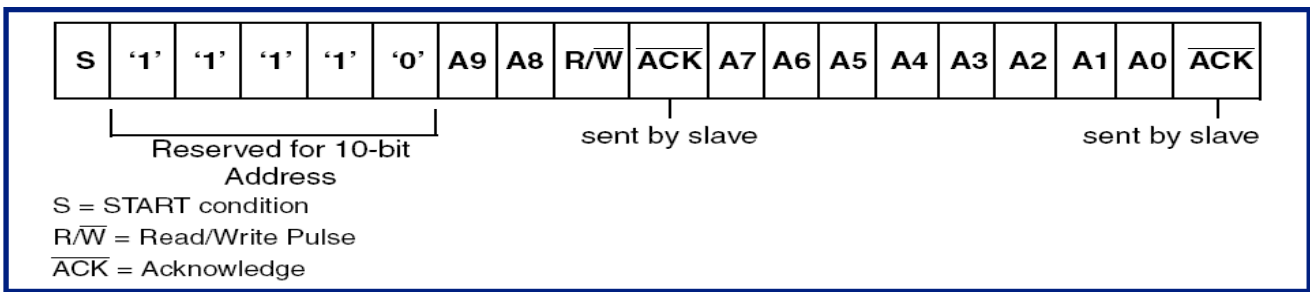


Figure 22-6: 10-bit Address Format

Table 22-1: I2C Definition of Bits in First Byte

Slave Address	R/W bit	Description
0000 000	0	General Call Address. I2C places the data in the receive buffer and issues a General Call interrupt.
0000 000	1	START byte.
0000 001	X	CBUS address. I2C ignores these accesses.
0000 010	X	Reserved
0000 011	X	Reserved
0000 1XX	X	High-speed master code
1111 1XX	X	Reserved.
1111 0XX	X	10-bit slave addressing.

Transmitting and Receiving Protocol

The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

If the master-transmitter is transmitting data as shown in Figure 22-7, then the slave-receiver responds to the master-transmitter with an acknowledge pulse after every byte of data is received.

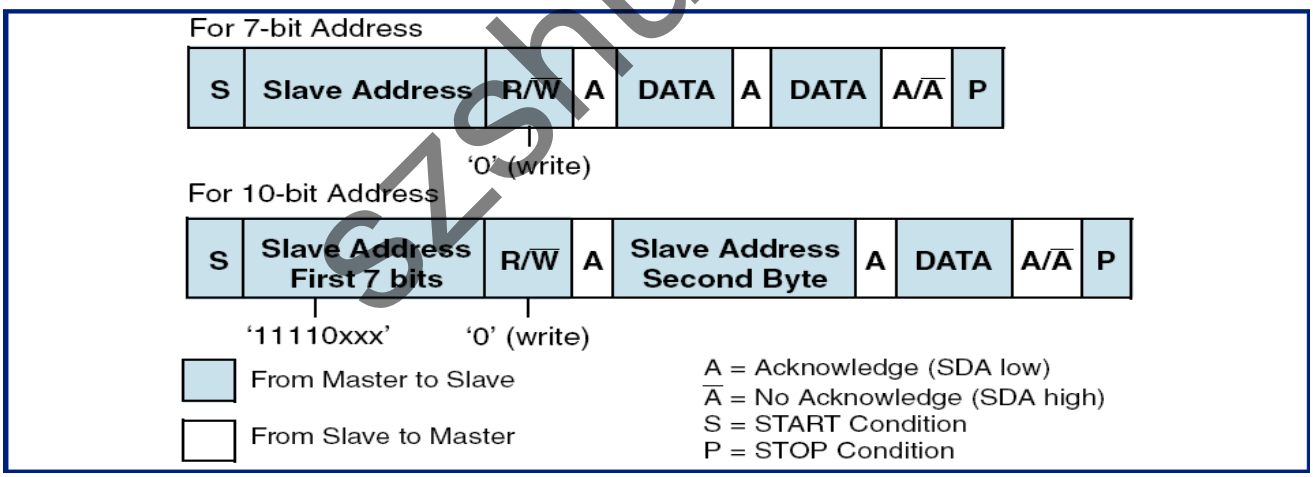


Figure 22-7: Master-Transmitter Protocol

Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 22-8, then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This is the way the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

When a master does not want to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. The master can then communicate with the same slave or a different slave.

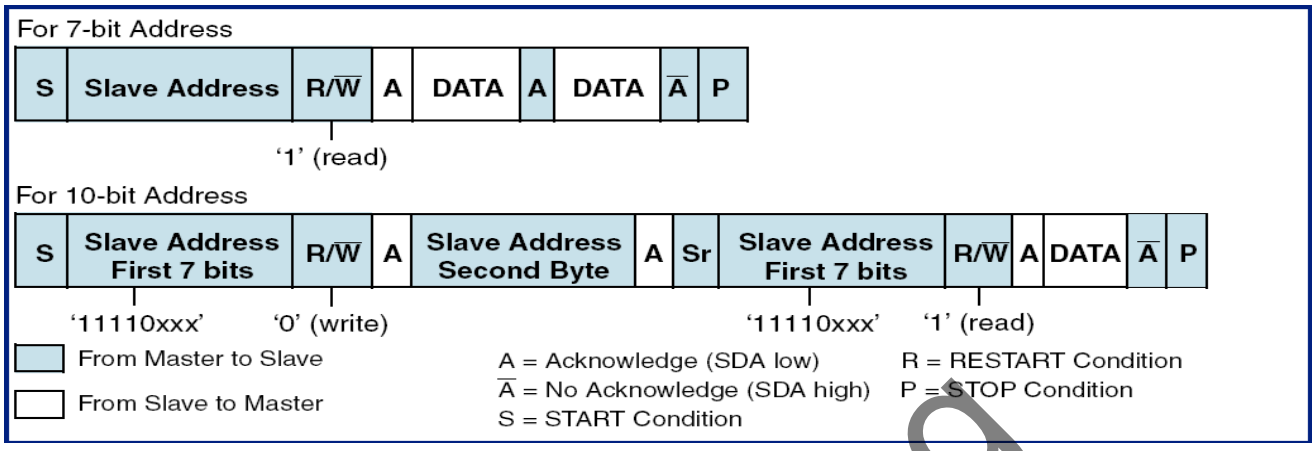


Figure 22-8: Master-Receiver Protocol

START BYTE Transfer Protocol

The START BYTE transfer protocol is set up for systems that do not have an on-board dedicated I2C hardware module. When the I2C is addressed as a slave, it always samples the I2C bus at the highest speed supported so that it never requires a START BYTE transfer. However, when I2C is a master, it supports the generation of START BYTE transfers at the beginning of every transfer in case a slave device requires it. This protocol consists of seven zeros being transmitted followed by a 1, as illustrated in Figure 22-9. This allows the processor that is polling the bus to under-sample the address phase until 0 is detected. Once the microcontroller detects a 0, it switches from the under sampling rate to the correct rate of the master.

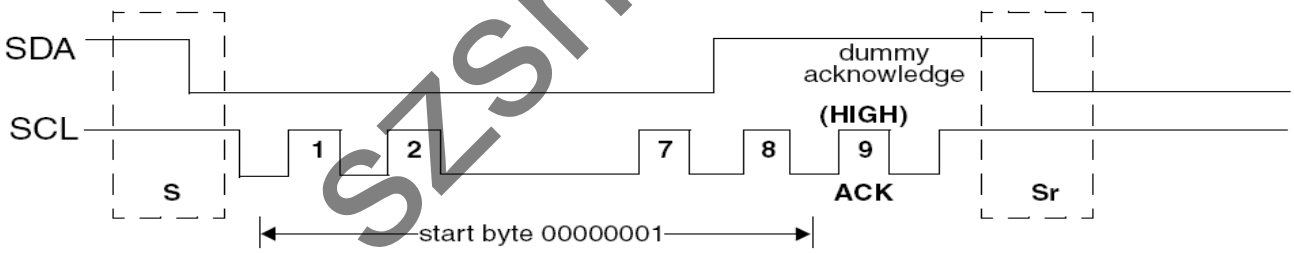


Figure 22-9: START BYTE Transfer

The START BYTE procedure is as follows:

1. Master generates a START condition.
2. Master transmits the START byte (0000 0001).
3. Master transmits the ACK clock pulse. (Present only to conform with the byte handling format used on the bus)
4. No slave sets the ACK signal to 0.
5. Master generates a RESTART (R) condition.

A hardware receiver does not respond to the START BYTE because it is a Reserved address and resets after the RESTART condition is generated.

Multiple Master Arbitration

The I2C bus protocol allows multiple masters to reside on the same bus. If there are two masters on the same I²C-bus, there is an arbitration procedure if both try to take control of the bus at the same time by generating a START condition at the same time. Once a master (for example, a microcontroller) has control of the bus, no other master can take control until the first master sends a STOP condition and places the bus in an idle state.

Arbitration takes place on the SDA line, while the SCL line is 1. The master, which transmits a 1 while the other master transmits 0, loses arbitration and turns off its data output stage. The master that lost arbitration can continue to generate clocks until the end of the byte transfer. If both masters are addressing the same slave device, the arbitration could go into the data phase. Figure 22-10 illustrates the timing of when two masters are arbitrating on the bus.

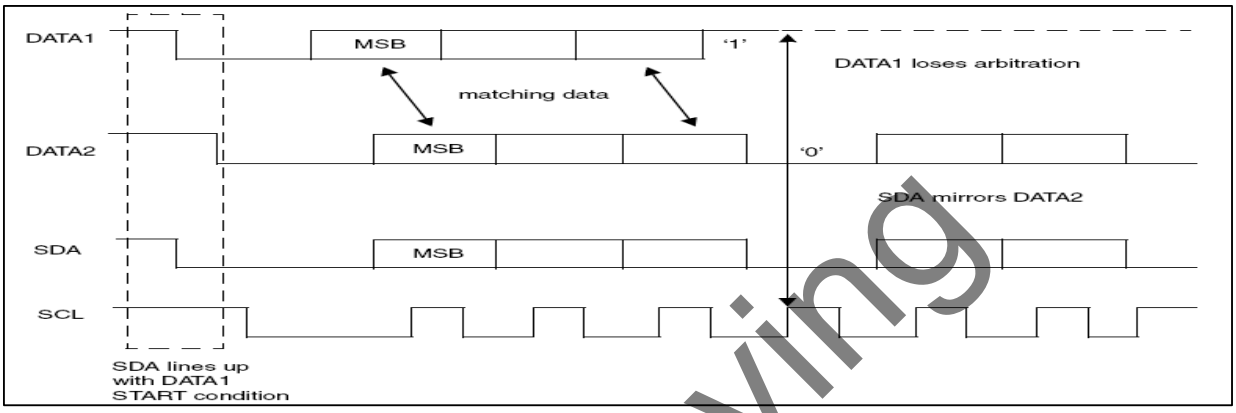


Figure 22-10: Multiple Master Arbitration

For high-speed mode, the arbitration cannot go into the data phase because each master is programmed with a unique high-speed master code. This 8-bitcode is defined by the system designer and is set by writing to the High Speed Master Mode Code Address Register, I2C_HS_MADDR. Because the codes are unique, only one master can win arbitration, which occurs by the end of the transmission of the high-speed master code.

Control of the bus is determined by address or master code and data sent by competing masters, so there is no central master nor any order of priority on the bus.

Arbitration is not allowed between the following conditions:

- A RESTART condition and a data bit
- A STOP condition and a data bit
- A RESTART condition and a STOP condition

Slaves are not involved in the arbitration process.

Clock Synchronization

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is valid only during the high period of SCL clock. Clock synchronization is performed using the wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL clock and transitions the SCL clock signal to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master goes into a HIGH wait state until the SCL clock line transitions to 1.

All masters then count off their high time, and the master with the shortest high time transitions the SCL line to 0. The masters then counts out their low time and the one with the longest low time forces the other master into a HIGH wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in Figure 22-11. Optionally, slaves may hold the SCL line low to slow down the timing on the I2C bus.

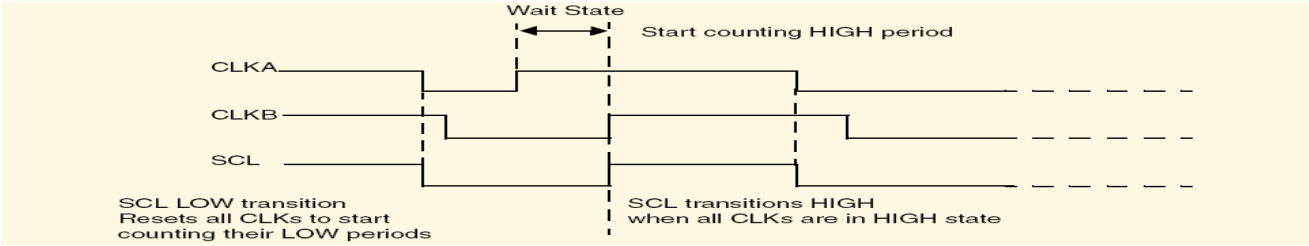


Figure 22-11: Multi-Master Clock Synchronization

22.5 Operation Modes

It is important to note that the I2C should only be set to operate as an I2C Master, or I2C Slave, but not both simultaneously. This is achieved by ensuring that bit 6 (I2C_SLAVE_DISABLE) and 0 (I2C_MASTER_MODE) of the I2C_CON register are never set to 0 and 1, respectively.

Slave Mode Operation

Initial Configuration

To use the I2C as a slave, perform the following steps:

1. Disable the I2C by writing a '0' to bit 0 of the I2C_ENABLE register.
2. Write to the I2C_SAR register (bits 9:0) to set the slave address. This is the address to which the I2C responds.
3. Write to the I2C_CON register to specify which type of addressing is supported (7- or 10-bit by setting bit 3). Enable the I2C in slave-only mode by writing a '0' into bit 6 (I2C_SLAVE_DISABLE) and a '0' to bit 0 (MASTER_MODE).
4. Enable the I2C by writing a '1' in bit 0 of the I2C_ENABLE register.

Slave-Transmitter Operation for a Single Byte

When another I2C master device on the bus addresses the I2C and requests data, the I2C acts as a slave-transmitter and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the slave address in the I2C_SAR register of the I2C.
2. The I2C acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.
3. The I2C asserts the RD_REQ interrupt (bit 5 of the I2C_RAW_INTR_STAT register) and holds the SCL line low. It is in a wait state until software responds.

   If the RD_REQ interrupt has been masked, due to I2C_INTR_MASK[5] register (M_RD_REQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the I2C_RAW_INTR_STAT register.
   1) Reads that indicate I2C_RAW_INTR_STAT[5] (R_RD_REQ bit field) being set to 1 must be treated as the equivalent of the RD_REQ interrupt being asserted.
   2) Software must then act to satisfy the I2C transfer.
   3) The timing interval used should be in the order of 10 times the fastest SCL clock period the I2C can handle. For example, for 400 kb/s, the timing interval is 25us.
4. If there is any data remaining in the TX FIFO before receiving the read request, then the I2C asserts a TX_ABRT interrupt (bit 6 of the I2C_RAW_INTR_STAT register) to flush the old data from the TX FIFO.

If the TX_ABRT interrupt has been masked, due to of I2C_INTR_MASK[6] register (M_TX_ABRT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the I2C_RAW_INTR_STAT register.

1) Reads that indicate bit 6 (R_TX_ABRT) being set to 1 must be treated as the equivalent of the TX_ABRT interrupt being asserted.
2) There is no further action required from software.
3) The timing interval used should be similar to that described in the previous step for the I2C_RAW_INTR_STAT[5] register.

5. Software writes to the I2C_DATA_CMD register with the data to be written (by writing a '0' in bit 8).
6. Software must clear the RD_REQ and TX_ABRT interrupts (bits 5 and 6, respectively) of the I2C_RAW_INTR_STAT register before proceeding.

   If the RD_REQ and/or TX_ABRT interrupts have been masked, then clearing of the I2C_RAW_INTR_STAT register will have already been performed when either the R_RD_REQ or R_TX_ABRT bit has been read as 1.
7. The I2C releases the SCL and transmits the byte.
8. The master may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Slave-Receiver Operation for a Single Byte

When another I2C master device on the bus addresses the I2C and is sending data, the I2C acts as a slave-receiver and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the I2C's slave address in the I2C_SAR register.
2. The I2C acknowledges the sent address and recognizes the direction of the transfer to indicate that the I2C is acting as a slave-receiver.
3. I2C receives the transmitted byte and places it in the receive buffer.

   If the RX FIFO is completely filled with data when a byte is pushed, then an overflow occurs and the I2C continues with subsequent I2C transfers. Because a NACK is not generated, software must recognize the overflow when indicated by the I2C (by the R_RX_OVER bit in the I2C_INTR_STAT register) and take appropriate actions to recover from lost data. Hence, there is a real time constraint on software to service the RX FIFO before the latter overflow as there is no way to re-apply pressure to the remote transmitting master. You must select a deep enough RX FIFO depth to satisfy the interrupt service interval of their system.
4. I2C asserts the RX_FULL interrupt (I2C_RAW_INTR_STAT[2] register).

   If the RX_FULL interrupt has been masked, due to setting I2C_INTR_MASK[2] register to 0 or setting I2C_TX_TL to a value larger than 0, then it is recommended that a timing routine (described in "Slave-Transmitter Operation for a Single Byte") be implemented for periodic reads of the I2C_STATUS register. Reads of the I2C_STATUS register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX_FULL interrupt being asserted.
5. Software may read the byte from the I2C_DATA_CMD register (bits 7:0).
6. The other master device may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Slave-Transfer Operation For Bulk Transfers

In the standard I2C protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave's TX FIFO. When a slave (slave-transmitter) is issued with a read request (RD_REQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter's TX FIFO.

I2C is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when I2C is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave's TX FIFO, the I2C holds the I2C SCL line low while it raises the read request interrupt (RD_REQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RD_REQ interrupt is masked, due to bit 5 (M_RD_REQ) of the I2C_INTR_STAT register being set to 0, then it is recommended that a timing routine be used to activate periodic reads of the I2C_RAW_INTR_STAT register. Reads of I2C_RAW_INTR_STAT that return bit 5 (R_RD_REQ) set to 1 must be treated as the equivalent of the RD_REQ interrupt referred to in this section. This timing routine is similar to that described in "Slave-Transmitter Operation for a Single Byte".

The RD_REQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows you to either write 1 byte or more than 1 byte into the TX FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte, then the slave must raise the RD_REQ again because the master is requesting for more data.

If the programmer knows in advance that the remote master is requesting a packet of n bytes, then when another master addresses I2C and requests data, the TX FIFO could be written with n number bytes and the remote master receives it as a continuous stream of data. For example, the I2C slave continues to send data to the remote master as long as the remote master is acknowledging the data sent and there is data available in the TX FIFO. There is no need to hold the SCL line low or to issue RD_REQ again.

If the remote master is to receive n bytes from the I2C but the programmer wrote a number of bytes larger than n to the TX FIFO, then when the slave finishes sending the requested n bytes, it clears the TX FIFO and ignores any excess bytes.

The I2C generates a transmit abort (TX_ABRT) event to indicate the clearing of the TX FIFO in this example. At the time an ACK/NACK is expected, if a NACK is received, then the remote master has all the data it wants. At this time, a flag is raised within the slave's state machine to clear the leftover data in the TX FIFO. This flag is transferred to the processor bus clock domain where the FIFO exists and the content of the TX FIFO is cleared at that time.

Master Mode Operation

Initial Configuration

To use the I2C as a master, perform the following steps:

1. Disable the I2C by writing 0 to the I2C_ENABLE register.
2. Write to the I2C_CON register to set the maximum speed mode supported (bits 2:1) and the desired speed of the I2C master-initiated transfers, either 7-bit or 10-bit addressing (bit 4). Ensure that bit 6 (I2C_SLAVE_DISABLE) is written with a '1' and bit 0 (MASTER_MODE) is written with a '1'.
3. Write to the I2C_TAR register the address of the I2C device to be addressed (bits 9:0). This register also indicates whether a General Call or a START BYTE command is going to be performed by I2C.
4. Only applicable for high-speed mode transfers. Write to the I2C_HS_MADDR register the desired master code for the I2C. The master code is programmer-defined.
5. Enable the I2C by writing a '1' in bit 0 of the I2C_ENABLE register.
6. Now write transfer direction and data to be sent to the I2C_DATA_CMD register. If the I2C_DATA_CMD register is written before the I2C is enabled, the data and commands are lost as the buffers are kept cleared when I2C is disabled.

   This step generates the START condition and the address byte on the I2C. Once I2C is enabled and there is data in the TX FIFO, I2C starts reading the data.

Master Transmit and Master Receive

The I2C supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I2C Rx/Tx Data Buffer and Command Register (I2C_DATA_CMD). The CMD bit [8] should be written to 0 for I2C write operations. Subsequently, a read command may be issued by writing "don't cares" to the lower byte of the I2C_DATA_CMD register, and a 1 should be written to the CMD bit.

Clock Frequency Configuration

When the I2C is configured as a master, the *CNT registers must be set before any I2C bus transaction can take place to ensure proper I/O timing. The *CNT registers are:

- I2C_SS_SCL_HCNT
- I2C_SS_SCL_LCNT
- I2C_FS_SCL_HCNT
- I2C_FS_SCL_LCNT
- I2C_HS_SCL_HCNT
- I2C_HS_SCL_LCNT

Setting the *_LCNT registers configures the number of pclk signals that are required for setting the low time of the SCL clock in each speed mode. Setting the *_HCNT* registers configures the number of pclk signals that are required for setting the high time of the SCL clock in each speed mode.

The table below shows some sample I2C_SS_SCL_HCNT calculations and it's similar to other five *CNT registers. These values apply only if the pclk is set to the given frequency in the table.

I2C Data Rate (kbps)	PCLK (MHz)	SCL High Required min (µs)	Minimum H_CNT	Actual SCL High Time (µs)
100	15	4	52	4.00
100	20	4	72	4.00
100	50	4	192	4.00
100	100	4	392	4.00

22.6 I2C Register Description

22.6.1 I2C Register Memory Map

Table 22-2: I2C Register Memory Map

Address	Symbol	Direction	Description
0x20DA_0000	I2C_CON0	R/W	I2C channel 0 control register
0x20DA_0004	I2C_TAR0	RC	I2C channel 0 Target Address
0x20DA_0008	I2C_SAR0	R/W	I2C channel 0 Slave Address
0x20DA_000C	I2C_HS_MADDR0	R/W	I2C cannel 0 HS Master Mode Code Address
0x20DA_0010	I2C_DATA_CMD0	R/W	I2C channel 0 Rx/Tx Data Buffer and Command
0x20DA_0014	I2C_SS_SCL_HCNT0	R/W	I2C channel 0 Standard speed I2C Clock SCL High Count
0x20DA_0018	I2C_SS_SCL_LCNT0	R/W	I2C channel 0 Standard speed I2C Clock SCL Low Count
0x20DA_001C	I2C_FS_SCL_HCNT0	R/W	I2C channel 0 Fast speed I2C Clock SCL High Count
0x20DA_0020	I2C_FS_SCL_LCNT0	R/W	I2C channel 0 Fast speed I2C Clock SCL Low Count
0x20DA_0024	I2C_HS_SCL_HCNT0	R/W	I2C channel 0 High speed I2C Clock SCL High Count
0x20DA_0028	I2C_HS_SCL_LCNT0	R/W	I2C channel 0 High speed I2C Clock SCL Low Count
0x20DA_002C	I2C_INTR_STAT0	R	I2C channel 0 Interrupt Status
0x20DA_0030	I2C_INTR_MASK0	R/W	I2C channel 0 Interrupt Mask
0x20DA_0034	I2C_RAW_INTR_STAT0	R	I2C channel 0 Raw Interrupt Status
0x20DA_0038	I2C_RX_TL0	R/W	I2C channel 0 Receive FIFO Threshold
0x20DA_003C	I2C_TX_TL0	R/W	I2C channel 0 Transmit FIFO Threshold

0x20DA_0040	I2C_CLR_INTR0	R	I2C channel 0 Clear Combined and Individual Interrupts
0x20DA_0044	I2C_CLR_RX_UNDER0	R	I2C channel 0 Clear RX_UNDER Interrupt
0x20DA_0048	I2C_CLR_RX_OVER0	R	I2C channel 0 Clear RX_OVER Interrupt
0x20DA_004C	I2C_CLR_TX_OVER0	R	I2C channel 0 Clear TX_OVER Interrupt
0x20DA_0050	I2C_CLR_RD_REQ0	R	I2C channel 0 Clear RD_REQ Interrupt
0x20DA_0054	I2C_CLR_TX_ABRT0	R	I2C channel 0 Clear TX_ABRT Interrupt
0x20DA_0058	I2C_CLR_RX_DONE0	R	I2C channel 0 Clear RX_DONE Interrupt
0x20DA_005C	I2C_CLR_ACTIVITY0	R	I2C channel 0 Clear ACTIVITY Interrupt
0x20DA_0060	I2C_CLR_STOP_DET0	R	I2C channel 0 Clear STOP_DET Interrupt
0x20DA_0064	I2C_CLR_START_DET0	R	I2C channel 0 Clear START_DET Interrupt
0x20DA_0068	I2C_CLR_GEN_CALL0	R	I2C channel 0 Clear GEN_CALL Interrupt
0x20DA_006C	I2C_ENABLE0	R/W	I2C channel 0 Enable
0x20DA_0070	I2C_STATUS0	R	I2C channel 0 Status Register
0x20DA_0074	I2C_TXFLR0	R	I2C channel 0 Transmit FIFO Level Registers
0x20DA_0078	I2C_RXFLR0	R	I2C channel 0 Receive FIFO Level Register
0x20DA_0080	I2C_TX_ABRT_SOURCE0	R/W	I2C channel 0 Transmit Abort Status Register
0x20DA_0098	I2C_ACK_GENERAL_CALL0	R/W	I2C channel 0 ACK General Call Register
0x20DA_009C	I2C_ENABLE_STATUS0	R	I2C channel 0 Enable Status Register
0x20DA_00A0	I2C_SDA_CFG0	R/W	I2C channel 0 SDA Configuration Register
0x20DA_1000	I2C_CON1	R/W	I2C channel 1 control register
0x20DA_1004	I2C_TAR1	RC	I2C channel 1 Target Address
0x20DA_1008	I2C_SAR1	R/W	I2C channel 1 Slave Address
0x20DA_100C	I2C_HS_MADDR1	R/W	I2C cannel 1 HS Master Mode Code Address
0x20DA_1010	I2C_DATA_CMD1	R/W	I2C channel 1 Rx/Tx Data Buffer and Command
0x20DA_1014	I2C_SS_SCL_HCNT1	R/W	I2C channel 1 Standard speed I2C Clock SCL High Count
0x20DA_1018	I2C_SS_SCL_LCNT1	R/W	I2C channel 1 Standard speed I2C Clock SCL Low Count
0x20DA_101C	I2C_FS_SCL_HCNT1	R/W	I2C channel 1 Fast speed I2C Clock SCL High Count
0x20DA_1020	I2C_FS_SCL_LCNT1	R/W	I2C channel 1 Fast speed I2C Clock SCL Low Count
0x20DA_1024	I2C_HS_SCL_HCNT1	R/W	I2C channel 1 High speed I2C Clock SCL High Count
0x20DA_1028	I2C_HS_SCL_LCNT1	R/W	I2C channel 1 High speed I2C Clock SCL Low Count
0x20DA_102C	I2C_INTR_STAT1	R	I2C channel 1 Interrupt Status
0x20DA_1030	I2C_INTR_MASK1	R/W	I2C channel 1 Interrupt Mask
0x20DA_1034	I2C_RAW_INTR_STAT1	R	I2C channel 1 Raw Interrupt Status
0x20DA_1038	I2C_RX_TL1	R/W	I2C channel 1 Receive FIFO Threshold
0x20DA_103C	I2C_TX_TL1	R/W	I2C channel 1 Transmit FIFO Threshold
0x20DA_1040	I2C_CLR_INTR1	R	I2C channel 1 Clear Combined and Individual Interrupts
0x20DA_1044	I2C_CLR_RX_UNDER1	R	I2C channel 1 Clear RX_UNDER Interrupt
0x20DA_1048	I2C_CLR_RX_OVER1	R	I2C channel 1 Clear RX_OVER Interrupt
0x20DA_104C	I2C_CLR_TX_OVER1	R	I2C channel 1 Clear TX_OVER Interrupt
0x20DA_1050	I2C_CLR_RD_REQ1	R	I2C channel 1 Clear RD_REQ Interrupt
0x20DA_1054	I2C_CLR_TX_ABRT1	R	I2C channel 1 Clear TX_ABRT Interrupt
0x20DA_1058	I2C_CLR_RX_DONE1	R	I2C channel 1 Clear RX_DONE Interrupt
0x20DA_105C	I2C_CLR_ACTIVITY1	R	I2C channel 1 Clear ACTIVITY Interrupt
0x20DA_1060	I2C_CLR_STOP_DET1	R	I2C channel 1 Clear STOP_DET Interrupt
0x20DA_1064	I2C_CLR_START_DET1	R	I2C channel 1 Clear START_DET Interrupt
0x20DA_1068	I2C_CLR_GEN_CALL1	R	I2C channel 1 Clear GEN_CALL Interrupt
0x20DA_106C	I2C_ENABLE1	R/W	I2C channel 1 Enable
0x20DA_1070	I2C_STATUS1	R	I2C channel 1 Status Register
0x20DA_1074	I2C_TXFLR1	R	I2C channel 1 Transmit FIFO Level Registers
0x20DA_1078	I2C_RXFLR1	R	I2C channel 1 Receive FIFO Level Register
0x20DA_1080	I2C_TX_ABRT_SOURCE1	R/W	I2C channel 1 Transmit Abort Status Register
0x20DA_1098	I2C_ACK_GENERAL_CALL1	R/W	I2C channel 1 ACK General Call Register
0x20DA_109C	I2C_ENABLE_STATUS1	R	I2C channel 1 Enable Status Register
0x20DA_10A0	I2C_SDA_CFG1	R/W	I2C channel 1 SDA Configuration Register

22.6.2 I2C Registers and Field Descriptions

Register 22-1: I2C Control Register （I2C_CONn）

Field	Symbol	Direction	Description	Default
[6]	SLAVE_DISABLE	R/W	If this bit is set (slave is disabled), only use as a master and does not perform any action that requires a slave. 0: slave is enabled 1: slave is disabled	0
[5]	RESTART_EN	R/W	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several I2C operations. 0: disable 1: enable When RESTART is disabled, the master is prohibited from performing the following functions: ● Change direction within a transfer (split) ● Send a START BYTE ● High-speed mode operation ● Combined format transfers in 7-bit addressing modes ● Read operation with a 10-bit address ● Send multiple bytes per transfer By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple I2C transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the I2C_RAW_INTR_STAT register.	1
[4]	10BITADDR_MASTER	R/W	Control whether the I2C starts its transfers in 7- or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing	1
[3]	10BITADDR_SLAVE	R/W	When acting as a slave, this bit controls whether the I2C responds to 7- or 10-bit addresses.. 0: 7-bit addressing 1: 10-bit addressing	1
[2:1]	SPEED	R/W	Operate speed mode 1: standard mode (100 kbit/s) 2: fast mode (400 kbit/s) 3: high speed mode (3.4 Mbit/s)	3
[0]	MASTER_MODE	R/W	This bit controls whether the I2C master is enabled. 0: master disabled 1: master enabled NOTE: Software should ensure that if this bit is written with ‘1,’ then bit 6 should also be written with a ‘1’.	1

Note: This register can be written only when the I2C is disabled, which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.

Register 22-2: I2C Target Address Register （I2C_TARn）

Field	Symbol	Direction	Description	Default
[11]	SPECIAL	R/W	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use I2C_TAR normally 1: perform special I2C command as specified in GC_OR_START bit	0

[10]	GC_OR_START	R/W	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the I2C_RAW_INTR_STAT register. The I2C remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1: START BYTE	0
[9:0]	I2C_TAR	R/W	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits.	0x55

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-3: I2C Slave Address Register （I2C_SARn）

Field	Symbol	Direction	Description	Default
[9:0]	I2C_SAR	R/W	The I2C_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only I2C_SAR[6:0] is used.	0x55

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-4: I2C High Speed Master Mode Code Address Register （I2C_HS_MADDR）

Field	Symbol	Direction	Description	Default
[2:0]	I2C_HS_MAR	R/W	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are Reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0's if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2).	1

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-5: I2C RX/TX Data Buffer and Command Register （I2C_DATA_CMDn）

I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

Field	Symbol	Direction	Description	Default
[8]	CMD	R/W	This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C acts as a slave. It controls only the direction when it acts as a master. 1 : read 0 : write When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a "don't care" because writes to this register are not required. In slave-transmitter mode, a "0" indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the I2C_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the	0

			I2C_TAR register has been cleared. If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. NOTE: It is possible that while attempting a master I2C read transfer on I2C, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing I2C. In this type of scenario, I2C ignores the I2C_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt.	
[7:0]	DAT	R/W	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the I2C. However, when you read this register, these bits return the value of data received on the I2C interface.	0

Register 22-6: Standard Speed I2C Clock SCL High Count Register （I2C_SS_SCL_HCNTn）

Field	Symbol	Direction	Description	Default
[15:0]	I2C_SS_SCL_HCNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed.	0x190

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-7: Standard Speed I2C Clock SCL Low Count Register （I2C_SS_SCL_LCNTn）

Field	Symbol	Direction	Description	Default
[15:0]	I2C_SS_SCL_LCNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low-period count for standard speed.	0x1d6

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-8: Fast Speed I2C Clock SCL High Count Register （I2C_FS_SCL_HCNT）

Field	Symbol	Direction	Description	Default
[15:0]	I2C_FS_SCL_HCNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed.	0x3c

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-9: Fast Speed I2C Clock SCL Low Count Register （I2C_FS_SCL_LCNTn）

Field	Symbol	Direction	Description	Default
[15:0]	I2C_FS_SCL_LCNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low-period count for fast speed.	0x82

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-10: High Speed I2C Clock SCL High Count Register （I2C_HS_SCL_HCNTn）

Field	Symbol	Direction	Description	Default
[15:0]	I2C_HS_SCL_HCNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for high speed.	0x6

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-11: High Speed I2C Clock SCL Low Count Register （I2C_HS_SCL_LCNTn）

Field	Symbol	Direction	Description	Default
[15:0]	I2C_HS_SCL_LCNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low-period count for high speed.	0x10

Note: This register can be written to only when I2C_ENABLE is set to 0.

Register 22-12: I2C Interrupt Status Register （I2C_INTR_STATn）

Each bit in this register has a corresponding mask bit in the I2C_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the I2C_RAW_INTR_STAT register.

Field	Symbol	Direction	Description	Default
[11]	R_GEN_CALL	R	See I2C_RAW_INTR_STAT for a detailed description of these bits.	0
[10]	R_START_DET	R		0
[9]	R_STOP_DET	R		0
[8]	R_ACTIVITY	R		0
[7]	R_RX_DONE	R		0
[6]	R_TX_ABRT	R		0
[5]	R_RD_REQ	R		0
[4]	R_TX_EMPTY	R		0
[3]	R_TX_OVER	R		0
[2]	R_RX_FULL	R		0
[1]	R_RX_OVER	R		0
[0]	R_RX_UNDER	R		0

Register 22-13: I2C Interrupt Mask Register （I2C_INTR_MASKn）

These bits mask their corresponding interrupt status bits. They are active high; a value of 0 prevents a bit from generating an interrupt.

Field	Symbol	Direction	Description	Default
[11]	M_GEN_CALL	R/W	This bit should be set to “1” when the IC_ACK_GENERAL_CALL register is set to “0”.	1
[10]	M_START_DET	R/W	These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register.	0
[9]	M_STOP_DET	R/W		0
[8]	M_ACTIVITY	R/W		0
[7]	M_RX_DONE	R/W		1
[6]	M_TX_ABRT	R/W		1
[5]	M_RD_REQ	R/W		1
[4]	M_TX_EMPTY	R/W		1
[3]	M_TX_OVER	R/W		1
[2]	M_RX_FULL	R/W		1
[1]	M_RX_OVER	R/W		1
[0]	M_RX_UNDER	R/W		1

Register 22-14: I2C Raw Interrupt Status Register （I2C_RAW_INTR_STATn）

Unlike the I2C_INTR_STAT register, these bits are not masked so they always show the true status of the I2C.

Field	Symbol	Direction	Description	Default
[11]	GEN_CALL	R	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. I2C stores the received data in the Rx buffer.	0

[10]	START_DET	R	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.	0
[9]	STOP_DET	R	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.	0
[8]	ACTIVITY	R	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: ● Disabling the I2C ● Reading the IC_CLR_ACTIVITY register ● Reading the IC_CLR_INTR register ● System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus.	0
[7]	RX_DONE	R	When the I2C is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.	0
[6]	TX_ABRT	R	This bit indicates if I2C, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a "transmit abort". When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. **Note**: The I2C flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register I2C_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.	0
[5]	RD_REQ	R	This bit is set to 1 when I2C is acting as a slave and another I2C master is attempting to read data from I2C. The I2C holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the I2C_DATA_CMD register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register.	0
[4]	TX_EMPTY	R	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the I2C_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with I2C_en=0, this bit is set to 0.	0
[3]	TX_OVER	R	Set during transmit if the transmit buffer is filled to I2C_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the I2C_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when I2C_en goes to 0, this interrupt is cleared.	0
[2]	RX_FULL	R	Set when the receive buffer reaches or goes above the RX_TL threshold in the I2C_RX_TL register. It is automatically cleared by	0

			hardware when buffer level goes below the threshold. If the module is disabled (I2C_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the I2C_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.	
[1]	RX_OVER	R	Set if the receive buffer is completely filled to I2C_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when I2C_en goes to 0, this interrupt is cleared.	0
[0]	RX_UNDER	R	Set if the processor attempts to read the receive buffer when it is empty by reading from the I2C_DATA_CMD register. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when I2C_en goes to 0, this interrupt is cleared.	0

Register 22-15: I2C Receive FIFO Threshold Register （I2C_RX_TLn）

Field	Symbol	Direction	Description	Default
[7:0]	RX_TL	R/W	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in I2C_RAW_INTR_STAT register). The valid range is 0-15, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 15 sets the threshold for 15 entries. Note: RX_TL[7:4] should be set to “0000”	0x0

Register 22-16: I2C Transmit FIFO Threshold Register （I2C_TX_TLn）

Field	Symbol	Direction	Description	Default
[7:0]	TX_TL	R/W	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in I2C_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 15 sets the threshold for 15 entries. Note: TX_TL[7:4] should be set to “0000”	0x0

Register 22-17: I2C Clear Combined and Individual Interrupt Register （I2C_CLR_INTRn）

Field	Symbol	Direction	Description	Default
[0]	CLR_INTR	R	Read this register to clear the combined interrupt, all individual interrupts, and the I2C_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the I2C_TX_ABRT_SOURCE register for an exception to clearing I2C_TX_ABRT_SOURCE.	0

Register 22-18: I2C Clear RX_UNDER Interrupt Register （I2C_CLR_RX_UNDERn）

Field	Symbol	Direction	Description	Default
[0]	CLR_RX_UNDER	R	Read this register to clear the RX_UNDER interrupt (bit 0) of the I2C_RAW_INTR_STAT register.	0

Register 22-19: I2C Clear RX_OVER Interrupt Register （I2C_CLR_RX_OVERn）

Field	Symbol	Direction	Description	Default
[0]	CLR_RX_OVER	R	Read this register to clear the RX_OVER interrupt (bit 1) of the I2C_RAW_INTR_STAT register.	0

Register 22-20: I2C Clear TX_OVER Interrupt Register （I2C_CLR_TX_OVERn）

Field	Symbol	Direction	Description	Default
[0]	CLR_TX_OVER	R	Read this register to clear the TX_OVER interrupt (bit 3) of the I2C_RAW_INTR_STAT register.	0

Register 22-21: I2C Clear RD_REQ Interrupt Register （I2C_CLR_RD_REQn）

Field	Symbol	Direction	Description	Default
[0]	CLR_RD_REQ	R	Read this register to clear the RD_REQ interrupt (bit 5) of the I2C_RAW_INTR_STAT register.	0

Register 22-22: I2C Clear TX_ABRT Interrupt Register （I2C_CLR_TX_ABRTn）

Field	Symbol	Direction	Description	Default
[0]	CLR_TX_ABRT	R	Read this register to clear the TX_ABRT interrupt (bit 6) of the I2C_RAW_INTR_STAT register, and the I2C_TX_ABRT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the I2C_TX_ABRT_SOURCE register for an exception to clearing I2C_TX_ABRT_SOURCE.	0

Register 22-23: I2C Clear RX_DONE Interrupt Register （I2C_CLR_RX_DONEn）

Field	Symbol	Direction	Description	Default
[0]	CLR_RX_DONE	R	Read this register to clear the RX_DONE interrupt (bit 7) of the I2C_RAW_INTR_STAT register.	0

Register 22-24: I2C Clear ACTIVITY Interrupt Register （I2C_CLR_ACTIVITYn）

Field	Symbol	Direction	Description	Default
[0]	CLR_ACTIVITY	R	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the I2C_RAW_INTR_STAT register.	0

Register 22-25: I2C Clear STOP_DET Interrupt Register （I2C_CLR_STOP_DETn）

Field	Symbol	Direction	Description	Default
[0]	CLR_STOP_DET	R	Read this register to clear the STOP_DET interrupt (bit 9) of the I2C_RAW_INTR_STAT register.	0

Register 22-26: I2C Clear START_DET Interrupt Register （I2C_CLR_START_DETn）

Field	Symbol	Direction	Description	Default
[0]	CLR_START_DET	R	Read this register to clear the START_DET interrupt (bit 10) of the I2C_RAW_INTR_STAT register.	0

Register 22-27: I2C Clear GEN_CALL Interrupt Register （I2C_CLR_GEN_CALLn）

Field	Symbol	Direction	Description	Default
[0]	CLR_GEN_CALL	R	Read this register to clear the GEN_CALL interrupt (bit 11) of the I2C_RAW_INTR_STAT register.	0

Register 22-28: I2C Enable Register （I2C_ENABLEn）

Field	Symbol	Direction	Description	Default
[0]	ENABLE	R/W	Controls whether the I2C is enabled. 0: Disables I2C (TX and RX FIFOs are held in an erased state) 1: Enables I2C Software can disable I2C while it is active. However, it is important that care be taken to ensure that I2C is disabled properly. When I2C is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed. • Status bits in the IC_INTR_STAT register are still active until I2C goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the I2C. For a detailed description on how to disable I2C, refer to “Disabling I2C” on page 62.	0

Register 22-29: I2C Status Register （I2C_STATSn）

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

Field	Symbol	Direction	Description	Default
[6]	SLV_ACTIVITY	R	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set.	0
[5]	MST_ACTIVITY	R	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set.	0
[4]	RFF	R	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.	0
[3]	RFNE	R	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty.	0
[2]	TFE	R	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.	0
[1]	TFNF	R	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.	0
[0]	ACTIVITY	R	I2C Activity Status. ACTIVITY bit—is the OR of SLV_ACTIVITY and MST_ACTIVITY bits.	0

Register 22-30: I2C Transmit FIFO Level Register （I2C_TXFLRn）

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the I2C_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Field	Symbol	Direction	Description	Default
[4:0]	TXFLR	R	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.	0

Register 22-31: I2C Receive FIFO Level Register （I2C_RXFLRn）

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in I2C_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Field	Symbol	Direction	Description	Default
[4:0]	RXFLR	R	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.	0

Register 22-32: I2C Transmit Abort Source Register （I2C_TX_ABRT_SOURCEn）

This register has 16 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the I2C_CLR_TX_ABRT register or the I2C_CLR_INTR register is read. To clear Bit 9, the source of the

ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (I2C_CON[5]=1), the SPECIAL bit must be cleared (I2C_TAR[11]), or the GC_OR_START bit must be cleared (I2C_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Field	Symbol	Direction	Description	Role of I2C
[15]	ABRT_SLVRD_INTX	R/W	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register.	Slave-Transmitter
[14]	ABRT_SLV_ARBLOST	R/W	1: Slave lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never “owns” the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then I2C no longer own the bus.	Slave-Transmitter
[13]	ABRT_SLVFLUSH_TXFIFO	R/W	1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.	Slave-Transmitter
[12]	ARB_LOST	R/W	1: Master has lost arbitration, or if I2C_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration.	Master-Transmitter or Slave-Transmitter
[11]	ABRT_MASTER_DIS	R/W	1: User tries to initiate a Master operation with the Master mode disabled.	Master-Transmitter or Master-Receiver
[10]	ABRT_10B_RD_NORSTRT	R/W	1: The restart is disabled (I2C_RESTART_EN bit (I2C_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.	Master-Receiver
[9]	ABRT_SBYTE_NORSTRT	R/W	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (I2C_CON[5]=1), the SPECIAL bit must be cleared (I2C_TAR[11]), or the GC_OR_START bit must be cleared (I2C_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. 1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte.	Master
[8]	ABRT_HS_NORSTRT	R/W	1: The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode.	Master-Transmitter or Master-Receiver
[7]	*ABRT_SBYTE_ACKDET*	R/W	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).	Master
[6]	ABRT_HS_ACKDET	R/W	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).	Master
[5]	ABRT_GCALL_READ	R/W	1: I2C in master mode sent a General Call but the user programmed the byte following the General Call to be a	Master-Transmitter

			read from the bus (I2C_DATA_CMD[9] is set to 1).	
[4]	ABRT_GCALL_NOACK	R/W	1: I2C in master mode sent a General Call and no slave on the bus acknowledged the General Call.	Master-Transmitter
[3]	ABRT_TXDATA_NOACK	R/W	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).	Master-Transmitter
[2]	ABRT_10ADDR2_NOACK	R/W	1: Master is in 10-bit address mode and the second address byte of the 10-bit addres	Master-Transmitter or Master-Receiver
[1]	ABRT_10ADDR1_NOACK	R/W	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.	Master-Transmitter or Master-Receiver
[0]	ABRT_7B_ADDR_NOACK	R/W	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.	Master-Transmitter or Master-Receiver

Register 22-33: I2C ACK General Call Register （I2C_ACK_GENERAL_CALLn）

Field	Symbol	Direction	Description	Default
[0]	ACK_GEN_CALL	R/W	ACK General Call. When set to 1, I2C responds with a ACK (by asserting I2C_data_oe) when it receives a General Call. Otherwise, I2C responds with a NACK (by negating I2C_data_oe).	1

Register 22-34: I2C Enable Status Register （I2C_ENABLE_STATSn）

The register is used to report the I2C hardware status when the I2C_ENABLE register is set from 1 to 0; that is, when I2C is disabled.

If I2C_ENABLE has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If I2C_ENABLE has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Field	Symbol	Direction	Description	Default
[2]	SLV_RX_DATA_LOST	R	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of I2C_ENABLE from 1 to 0. When read as 1, I2C is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and I2C_ENABLE has been set to 0, then this bit is also set to 1. When read as 0, I2C is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when I2C_EN (bit 0) is read as 0.	0
[1]	SLV_DISABLED_WHILE_BUSY	R	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the I2C_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the I2C_ENABLE register while: (a) I2C is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, I2C is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address	0

			matches the slave address set in I2C (I2C_SAR register) OR if the transfer is completed before I2C_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and I2C_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, I2C is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when I2C_EN (bit 0) is read as 0.	
[0]	IC_EN	R	I2C_en Status. This bit always reflects the value driven on the output port I2C_en. When read as 1, I2C is deemed to be in an enabled state. When read as 0, I2C is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).	0

Register 22-35: I2C SDA Configure Register （I2C_SDA_CFGn）

Field	Symbol	Direction	Description	Default
[8]	IGNORE_MST_TX_ACK	R/W	When this bit is set, Master Transmitter ignore Slave ACK, Master FSM auto-generate internal ACK.	0
[7:0]	SDA_HDLY_NUM	R/W	This register controls the amount of SDA hold time(tHD:DAT, in terms of number of PCLK clock periods)	2

23 PWM Timer

23.1 Overview

The IMAPx210 PWM-Timer has five 16-bit timers. Timer 0, 1, 2, and 3 have Pulse Width Modulation (PWM) function. Timer 4 has an internal timer only with no output pins. The timer 0 has a dead-zone generator, which is used with a large current device.

The timer 0 and 1 share an 8-bit prescaler, while the timer 2, 3 and 4 share other 8-bit prescaler. Each timer has a clock divider, which generates 4 different divided signals (1/2, 1/4, 1/8, 1/16). Each timer block receives its own clock signals from the clock divider, which receives the clock from the corresponding 8-bit prescaler. The 8- bit prescaler is programmable and divides the PCLK according to the loading value, which is stored in TCFG0 and TCFG1 registers.

The timer count buffer register (TCNTBn) has an initial value which is loaded into the down-counter when the timer is enabled. The timer compare buffer register (TCMPBn) has an initial value which is loaded into the compare register to be compared with the down-counter value. This double buffering feature of TCNTBn and TCMPBn makes the timer generate a stable output when the frequency and duty ratio are changed.

Each timer has its own 16-bit down counter, which is driven by the timer clock. When the down counter reaches zero, the timer interrupt request is generated to inform the CPU that the timer operation has been completed. When the timer counter reaches zero, the value of corresponding TCNTBn is automatically loaded into the down counter to continue the next operation. 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 value of TCMPBn is used for pulse width modulation (PWM). 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.

Features

- Five 16-bit timers
- Two 8-bit prescalers & Two 4-bit divider
- Programmable duty control of output waveform (PWM)
- Auto reload mode or one-shot pulse mode
- Dead-zone generator

Block Diagram

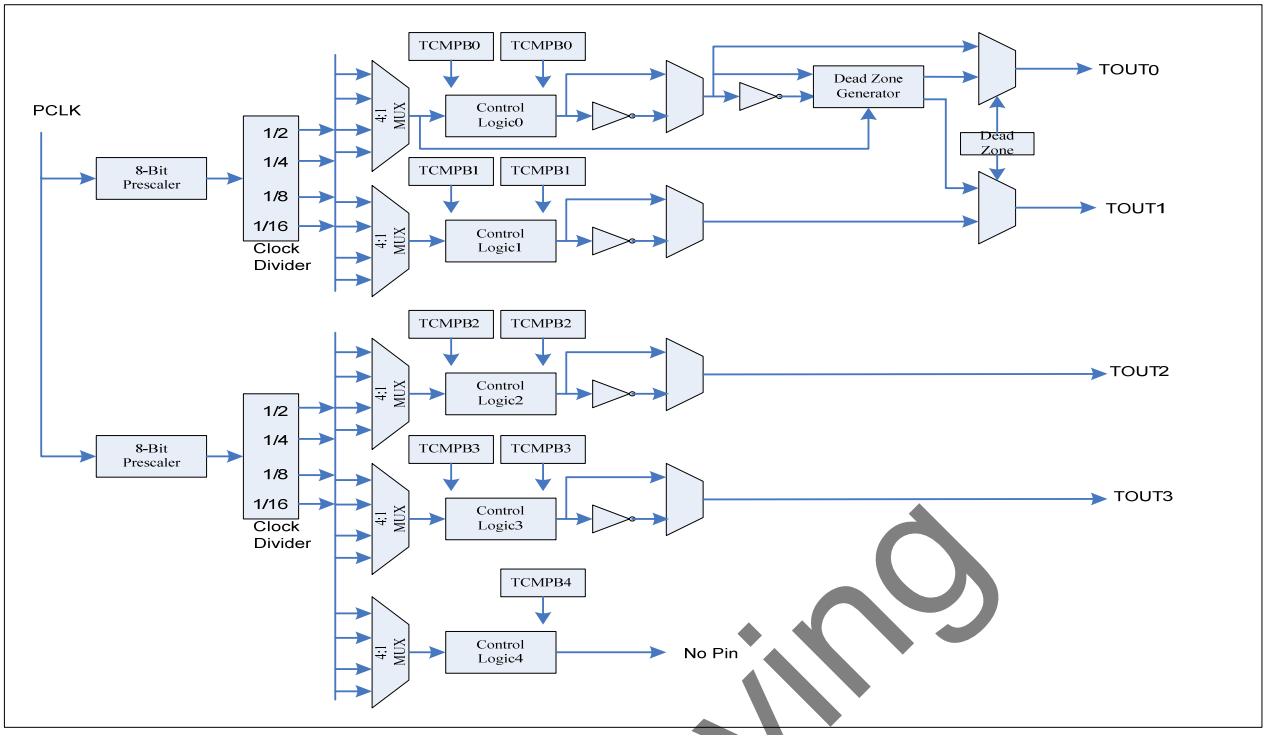


Figure 23-1: PWM Timer Block Diagram

23.2 PWM Timer Operation

Prescaler Divider

An 8-bit prescaler and a 4-bit divider make the following output frequencies:

4-bit Divider Settings	Minimum Resolution (prescaler = 0)	Maximum Resolution (prescaler = 255)	Maximum Interval (TCNTBn = 65535)
1/2 (PCLK = 50 MHz)	0.0400 us (25.0000 MHz)	10.2400 us (97.6562 kHz)	0.6710 sec
1/4 (PCLK = 50 MHz)	0.0800 us (12.5000 MHz)	20.4800 us (48.8281 kHz)	1.3421 sec
1/8 (PCLK = 50 MHz)	0.1600 us (6.2500 MHz)	40.9601 us (24.4140 kHz)	2.6843 sec
1/16 (PCLK = 50 MHz)	0.3200 us (3.1250 MHz)	81.9188 us (12.2070 kHz)	5.3686 sec

Basic Timer Operation

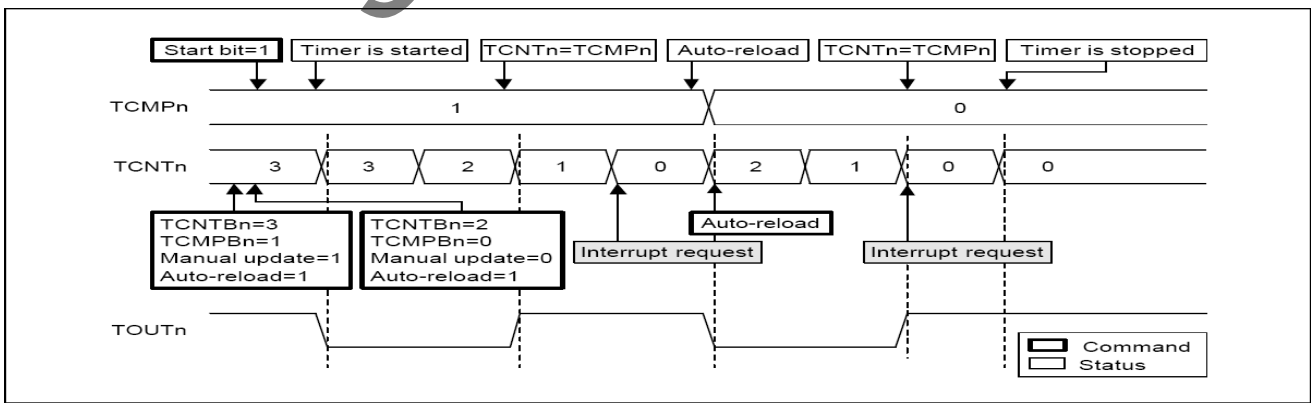


Figure 23-2: Timer Operations

A timer (except the timer ch-5) has TCNTBn, TCNTn, TCMPBn and TCMPn. (TCNTn and TCMPn are the names of the internal registers. The TCNTn register can be read from the TCNTOn register) The TCNTBn and the TCMPBn are loaded into the TCNTn and the TCMPn when the timer reaches 0. When the TCNTn reaches 0, an interrupt request will occur if the interrupt is enabled.

Auto Reload & Double Buffering

IMAPX210 PWM Timers have a double buffering function, enabling the reload value changed for the next timer operation without stopping the current timer operation. So, although the new timer value is set, a current timer operation is completed successfully.

The timer value can be written into Timer Count Buffer register (TCNTBn) and the current counter value of the timer can be read from Timer Count Observation register (TCNTOn). If the TCNTBn is read, the read value does not indicate the current state of the counter but the reload value for the next timer duration.

The auto-reload operation copies the TCNTBn into TCNTn when the TCNTn reaches 0. The value, written into the TCNTBn, is loaded to the TCNTn only when the TCNTn reaches 0 and auto reload is enabled. If the TCNTn becomes 0 and the auto reload bit is 0, the TCNTn does not operate any further.

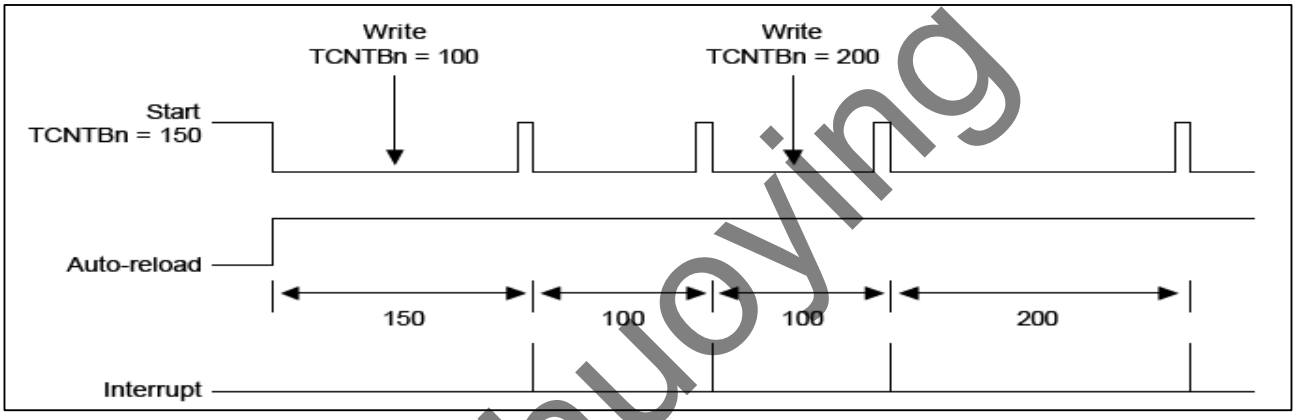


Figure 23-3: Example of Double Buffering Function

Timer Initialization Using Manual Update Bit and Inverter Bit

An auto reload operation of the timer occurs when the down counter reaches 0. So, a starting value of the TCNTn has to be defined by the user in advance. In this case, the starting value has to be loaded by the manual update bit. The following steps describe how to start a timer:

1) Write the initial value into TCNTBn and TCMPBn.

2) Set the manual update bit of the corresponding timer. It is recommended that you configure the inverter on/off bit. (Whether use inverter or not).

3) Set start bit of the corresponding timer to start the timer (and clear the manual update bit).

Note: Whenever TOUT inverter on/off bit is changed, the TOUTn logic value will also be changed whether the timer runs. Therefore, it is desirable that the inverter on/off bit is configured with the manual update bit.

Timer Operation

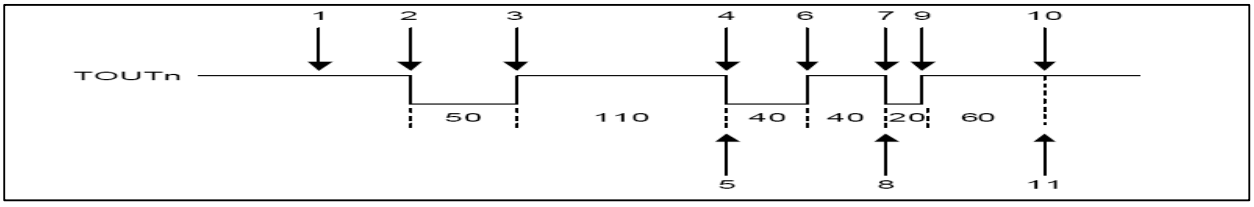


Figure 23-4: Example of a Timer Operation

The above figure shows the result of the following procedure:

1. Enable the auto re-load function. Set the TCNTBn to 160 (50+110) and the TCMPBn to 110. Set the manual update bit and configure the inverter bit (on/off). The manual update bit sets TCNTn and TCMPn to the values of TCNTBn and TCMPBn, respectively.
   And then, set the TCNTBn and the TCMPBn to 80 (40+40) and 40, respectively, to determine the next reload value.
2. Set the start bit, provided that manual_update is 0 and the inverter is off and auto reload is on. The timer starts counting down after latency time within the timer resolution.
3. When the TCNTn has the same value as that of the TCMPn, the logic level of the TOUTn is changed from low to high.
4. When the TCNTn reaches 0, the interrupt request is generated and TCNTBn value is loaded into a temporary register. At the next timer tick, the TCNTn is reloaded with the temporary register value (TCNTBn).
5. In Interrupt Service Routine (ISR), the TCNTBn and the TCMPBn are set to 80 (20+60) and 60, respectively, for the next duration.
6. When the TCNTn has the same value as the TCMPn, the logic level of TOUTn is changed from low to high.
7. When the TCNTn reaches 0, the TCNTn is reloaded automatically with the TCNTBn, triggering an interrupt request.
8. In Interrupt Service Routine (ISR), auto reload and interrupt request are disabled to stop the timer.
9. When the value of the TCNTn is same as the TCMPn, the logic level of the TOUTn is changed from low to high.
10. Even when the TCNTn reaches 0, the TCNTn is not any more reloaded and the timer is stopped because auto reload has been disabled.
11. No more interrupt requests are generated.

Note: It's important to note that timer start control bit is just a timer startup signal and self-clear after configuration, so if you want to stop timer, you can clear auto reload bit and clear manual update bit for next operation.

Pulse Width Modulation (PWM)

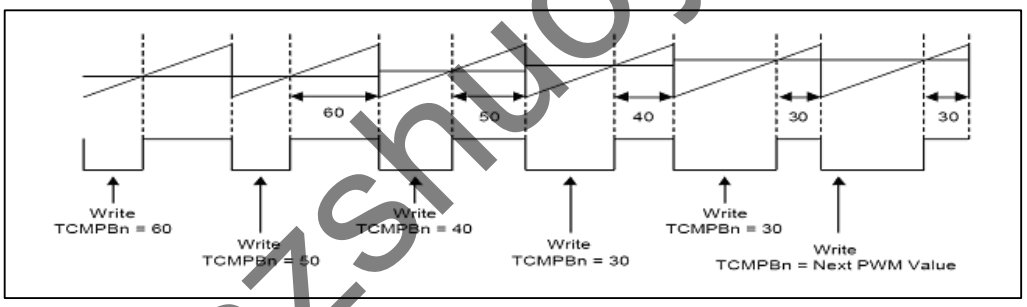


Figure 23-5: Example of PWM

PWM function can be implemented by using the TCMPBn. PWM frequency is determined by TCNTBn. Figure 23-5 shows a PWM value determined by TCMPBn.

For a higher PWM value, decrease the TCMPBn value. For a lower PWM value, increase the TCMPBn value. If an output inverter is enabled, the increment/decrement may be reversed.

The double buffering function allows the TCMPBn, for the next PWM cycle, written at any point in the current PWM cycle by ISR or other routine.

Dead Zone Generator

The Dead Zone is for the PWM control in a power device. This function enables the insertion of the time gap between a turn-off of a switching device and a turn on of another switching device. This time gap prohibits the two switching devices from being turned 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 wave form of TOUT0 and nTOUT0 will be TOUT0_DZ and nTOUT0_DZ, respectively. nTOUT0_DZ is routed to the TOUT1 pin.

In the dead zone interval, TOUT0_DZ and nTOUT0_DZ can never be turned on simultaneously.

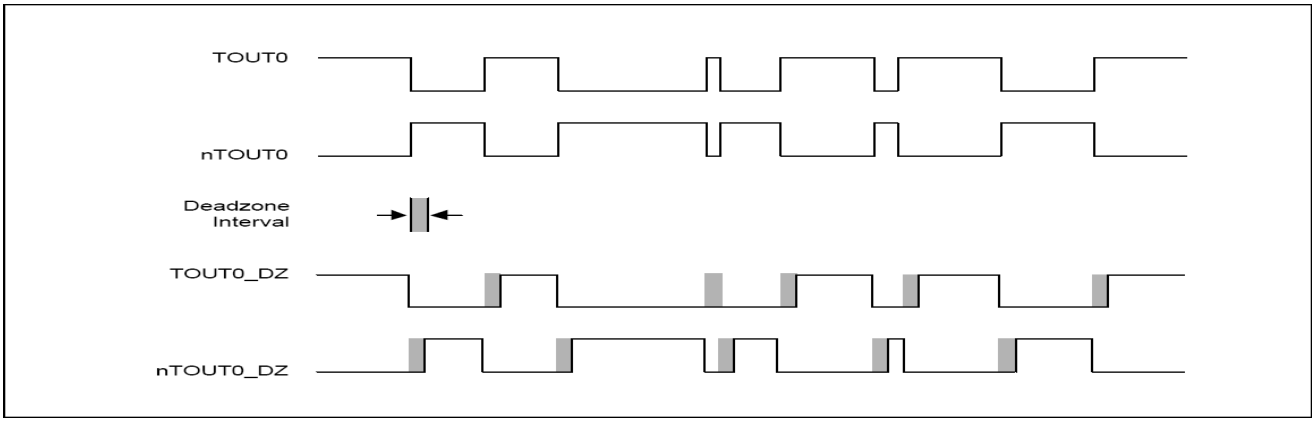


Figure 23-6: The Wave Form When a Dead Zone Feature is Enabled

DMA Request Mode

The PWM timer can generate a DMA request at every specific time. The timer keeps DMA request signals (DMA_REQ) high until the timer receives an ACK signal. When the timer receives the ACK signal, it makes the request signal inactive. The timer, which generates the DMA request, is determined by setting DMA mode bits (in TCFG1 register). If one of timers is configured as DMA request mode, that timer does not generate an interrupt request. The others can generate interrupt normally.

Table 23-1: DMA Mode Configuration and DMA / Interrupt Operation

DMA Mode	DMA Request	Timer0 INT	Timer1 INT	Timer2 INT	Timer3 INT	Timer4 INT
0000	No select	ON	ON	ON	ON	ON
0001	Timer0	OFF	ON	ON	ON	ON
0010	Timer1	ON	OFF	ON	ON	ON
0011	Timer2	ON	ON	OFF	ON	ON
0100	Timer3	ON	ON	ON	OFF	ON
0101	Timer4	ON	ON	ON	ON	OFF
0111	No select	OFF	OFF	OFF	OFF	OFF

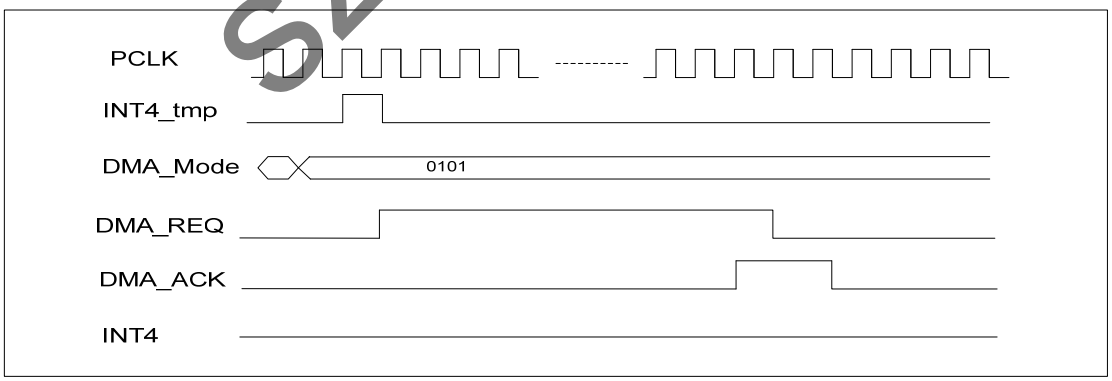


Figure 23-7: Timer4 DMA Mode Operation

23.3 PWM Timer Register Description

23.3.1 PWM Timer Register Memory Map

Table 23-2: PWM Timer Register Memory Map

Address	Symbol	Direction	Description
0x20D7_0000	TCFG0	R/W	Timer configuration 0 register
0x20D7_0004	TCFG1	R/W	Timer configuration 1 register
0x20D7_0008	TCON	R/W	Timer control register
0x20D7_000C	TCNTB0	R/W	Timer count buffer 0 register
0x20D7_0010	TCMPB0	R/W	Timer compare buffer 0 register
0x20D7_0014	TCNTO0	R	Timer count observation 0 register
0x20D7_0018	TCNTB1	R/W	Timer count buffer 1 register
0x20D7_001C	TCMPB1	R/W	Timer compare buffer 1 register
0x20D7_0020	TCNTO1	R	Timer count observation 1 register
0x20D7_0024	TCNTB2	R/W	Timer count buffer 2 register
0x20D7_0028	TCMPB2	R/W	Timer compare buffer 2 register
0x20D7_002C	TCNTO2	R	Timer count observation 2 register
0x20D7_0030	TCNTB3	R/W	Timer count buffer 3 register
0x20D7_0034	TCMPB3	R/W	Timer compare buffer 3 register
0x20D7_0038	TCNTO3	R	Timer count observation 3 register
0x20D7_003C	TCNTB4	R/W	Timer count buffer 4 register
0x20D7_0040	TCNTO4	R	Timer count observation 4 register

23.3.2 PWM Timer Registers and Field Descriptions

Register 23-1: PWM Timer Configuration 0 Register （TCFG0）

Field	Symbol	Direction	Description	Default
[23:16]	Dead zone length	R/W	These 8 bits determine the dead zone length. The 1 unit time of the dead zone length is equal to that of timer 0.	0x0
[15:8]	Prescaler 1	R/W	These 8 bits determine prescaler value for Timer 2, 3 and 4.	0x0
[7:0]	Prescaler 0	R/W	These 8 bits determine prescaler value for Timer 0 and 1	0x0

Timer internal clock Frequency = PCLK / {prescaler value+1} / {divider value}

{prescaler value} = 0~255

{divider value} = 2, 4, 8, 16

Register 23-2: PWM Timer Configuration 1 Register （TCFG1）

Field	Symbol	Direction	Description	Default
[23:20]	DMA mode	R/W	Select DMA request channel 0000 = No select (all interrupt) 0001 = Timer0 0010 = Timer1 0011 = Timer2 0100 = Timer3 0101 = Timer4 0111 = All Off ELSE = Reserved Note: For detail of DMA Mode, please see DMA Request Mode section.	0
[19:16]	MUX 4	R/W	Select MUX input for PWM Timer4. 1000 = 1/2	0

			1001 = 1/4 1010 = 1/8 1011 = 1/16 0xxx = shut down clk Else = Reserved	
[15:12]	MUX 3	R/W	Select MUX input for PWM Timer3. 1000 = 1/2 1001 = 1/4 1010 = 1/8 1011 = 1/16 0xxx = shut down clk Else = Reserved	0
[11:8]	MUX 2	R/W	Select MUX input for PWM Timer2. 1000 = 1/2 1001 = 1/4 1010 = 1/8 1011 = 1/16 0xxx = shut down clk Else = Reserved	0
[7:4]	MUX 1	R/W	Select MUX input for PWM Timer1. 1000 = 1/2 1001 = 1/4 1010 = 1/8 1011 = 1/16 0xxx = shut down clk Else = Reserved	0
[3:0]	MUX 0	R/W	Select MUX input for PWM Timer0. 1000 = 1/2 1001 = 1/4 1010 = 1/8 1011 = 1/16 0xxx = shut down clk Else = Reserved	0

Register 23-3: PWM Timer Control Register （TCON）

Field	Symbol	Direction	Description	Default
[22]	Timer4AROF	R/W	Determine auto reload on/off for Timer 4. 0 = One-shot 1 = Interval mode (auto reload)	0
[21]	Timer4MU *	W	Start manual update for Timer 4, This field is self-clear bit. 0 = No operation 1 = Update TCNTB4	0
[20]	Timer4ST	W	When set will start Timer 4 and this field is self-clear bit.	0
[19]	Timer3AROF	R/W	Determine auto reload on/off for Timer 3. 0 = One-shot 1 = Interval mode (auto reload)	0
[18]	Timer3OINV	R/W	Determine output inverter on/off for Timer 3. 0 = Inverter off 1 = Inverter on for TOUT3	1
[17]	Timer3MU *	W	Start manual update for Timer 3, This field is self-clear bit. 0 = No operation 1 = Update TCNTB3	0
[16]	Timer3ST	W	When set will start Timer 3 and this field is self-clear bit.	0
[15]	Timer2AROF	R/W	Determine auto reload on/off for Timer 2. 0 = One-shot 1 = Interval mode (auto reload)	0
[14]	Timer2OINV	R/W	Determine output inverter on/off for Timer 2. 0 = Inverter off 1 = Inverter on for TOUT2	1
[13]	Timer2MU *	W	Start manual update for Timer 2, This field is self-clear bit.	0

			0 = No operation 1 = Update TCNTB2	
[12]	Timer2ST	W	When set will start Timer 2 and this field is self-clear bit.	0
[11]	Timer1AROF	R/W	Determine auto reload on/off for Timer 1. 0 = One-shot 1 = Interval mode (auto reload)	0
[10]	Timer1OINV	R/W	Determine output inverter on/off for Timer 1. 0 = Inverter off 1 = Inverter on for TOUT1	1
[9]	Timer1MU *	W	Start manual update for Timer 1, This field is self-clear bit. 0 = No operation 1 = Update TCNTB1	0
[8]	Timer1ST	W	When set will start Timer 1 and this field is self-clear bit.	0
[7:5]	Reserved			
[4]	DeadZoneEnable	R/W	Determine the dead zone operation. 0 = Disable 1 = Enable	0
[3]	Timer0AROF	R/W	Determine auto reload on/off for Timer 0. 0 = One-shot 1 = Interval mode (auto reload)	0
[2]	Timer0OINV	R/W	Determine output inverter on/off for Timer 0. 0 = Inverter off 1 = Inverter on for TOUT0	1
[1]	Timer0MU *	W	Start manual update for Timer 1, This field is self-clear bit. 0 = No operation 1 = Update TCNTB0	0
[0]	Timer0ST	W	When set will start Timer 0 and this field is self-clear bit.	0

[*]: The bits have to be cleared at next writing.

Register 23-4: PWM Timer 0-4 Count Buffer Register （TCNTB0-4）

Field	Symbol	Direction	Description	Default
[15:0]	Count Buffer Register	R/W	count buffer value for Timer 0-4	0

Register 23-5: PWM Timer 0-3 Compare Buffer Register （TCMPB0-3）

Field	Symbol	Direction	Description	Default
[15:0]	Compare Buffer Register	R/W	compare buffer value for Timer 0-3	0

Register 23-6: PWM Timer 0-4 Count Observation Register （TCNTO0-4）

Field	Symbol	Direction	Description	Default
[15:0]	Count Observation Register	R	count observation for Timer 0-4	0

24 UART

24.1 Overview

The IMAPx210 Universal Asynchronous Receiver and Transmitter (UART) provide four independent asynchronous serial I/O (SIO) ports, each of which 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 CPU and the UART. The UART can support bit rates up to 115.2K bps using PCLK clock. If an external device provides the UART with UEXTCLK, then the UART can operate at higher speed. Each UART channel contains two 64-byte FIFOs for receiver and transmitter.

The IMAPx210 UART includes programmable baud rates, UART and infrared (IR) transmit/receive, one or two stop bit insertion, 5-bit, 6-bit, 7-bit or 8-bit data width and parity checking.

Each UART contains a baud-rate generator, transmitter, receiver and a control unit, as shown in Figure 24-1. The baud-rate generator can be clocked by PCLK or UEXTCLK (system generated clock). 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 (nTXD). Meanwhile, received data is shifted from the receive data pin (nRXD), and then copied to FIFO from the shifter.

Features

- Each UART Ch with IrDA 1.0 and RS232
- Each UART Ch with DMA-based or interrupt-based operation
- Each UART Ch with two 64-byte FIFOs (TX/RX FIFO)
- UART Ch 0 and 1 with nRTS0, nCTS0, nRTS1, and nCTS1
- Supports handshake transmit/receive

Block Diagram

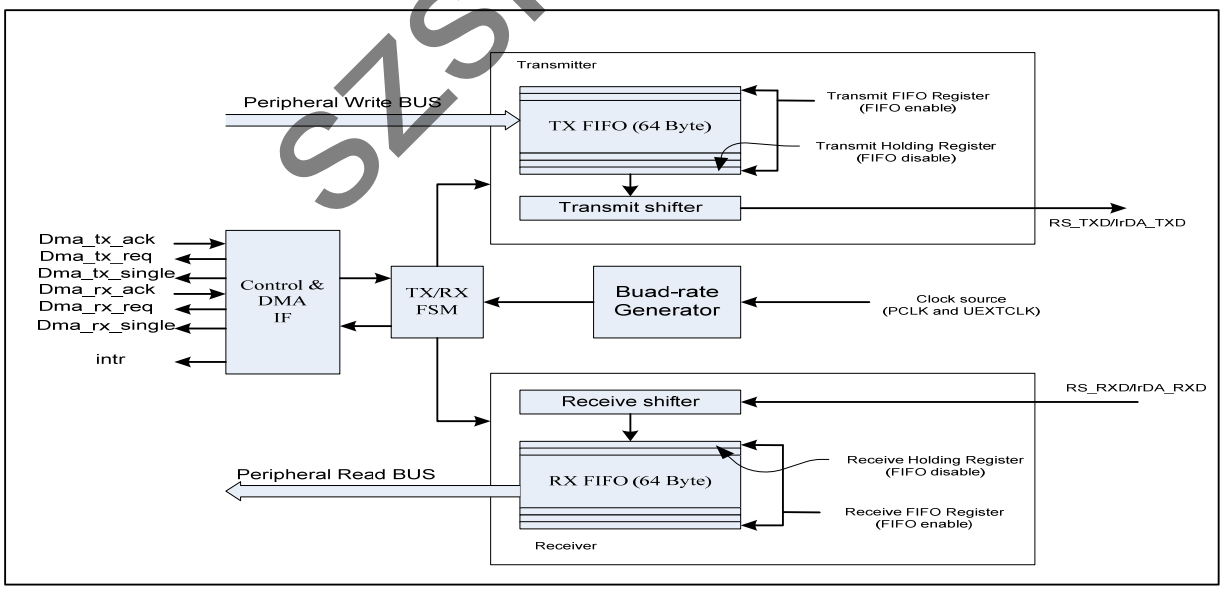


Figure 24-1: UART Block Diagram

24.2 UART Operation Modes

UART (RS232) Serial Protocol

Because the serial communication between the UART and the selected device is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. Utilizing these bits allows two devices to be synchronized. This structure of serial data accompanied by start and stop bits is referred to as a character, as shown in following figure.

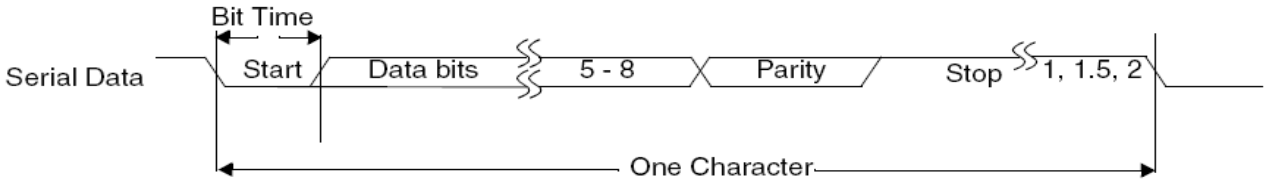


Figure 24-2: Serial Data Format

An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to provide the UART with the ability to perform simple error checking on the received data.

The UART Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the start bit, starting with the least-significant bit (LSB). These are followed by the optional parity bit, followed by the stop bit(s), which can be 1, 1.5 or 2.

All the bits in the transmission (with exception to the half stop bit when 1.5 stop bits are used) are transmitted for exactly the same time duration. This is referred to as a Bit Period or Bit Time. One Bit Time equals 16 baud clocks. To ensure stability on the line the receiver samples the serial input data at approximately the mid point of the Bit Time once the start bit has been detected. As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid point for sampling is not difficult, that is every 16 baud clocks after the mid point sample of the start bit. Figure 24-3 shows the sampling points of the first couple of bits in a serial character.

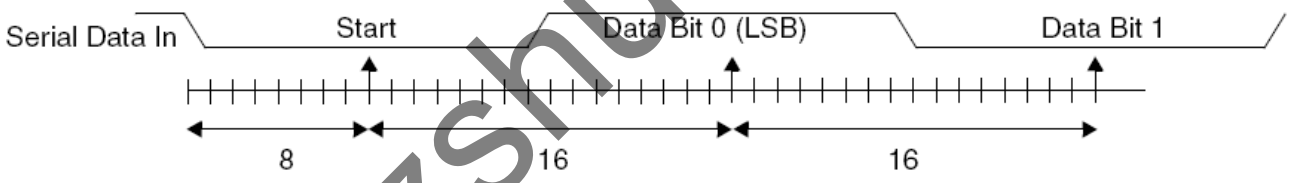


Figure 24-3: Receiver Serial Data Sample Points

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 K.

The data format is similar to the standard serial (RS TXD and RS RXD) data format. Each data character is sent serially, beginning with a start bit, followed by 8 data bits, and ending with at least one stop bit. Thus, the number of data bits that can be sent is fixed. No parity information can be supplied and only one stop bit is used while in this mode.

Trying to adjust the number of data bits sent or enable parity with the Line Control Register (LCR) has no effect. When the UART is configured to support IrDA 1.0 SIR it can be enabled with Mode Control Register (MCR) bit 6. When the UART is not configured to support IrDA SIR mode, none of the logic is implemented and the mode cannot be activated, reducing total gate counts. When SIR mode is enabled and active, serial data is transmitted and received on the IrDA TXD and IrDA RXD ports, respectively.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Thus, each new character begins with an infrared pulse for the start bit. However, received data is inverted from transmitted data due to the infrared pulses energizing the photo transistor base of the IrDA receiver, pulling its output low. This inverted transistor output is then fed to the UART IrDA RXD port, which then has

correct UART polarity. Figure 24-4 shows the timing diagram for the IrDA SIR data format in comparison to the standard serial format.

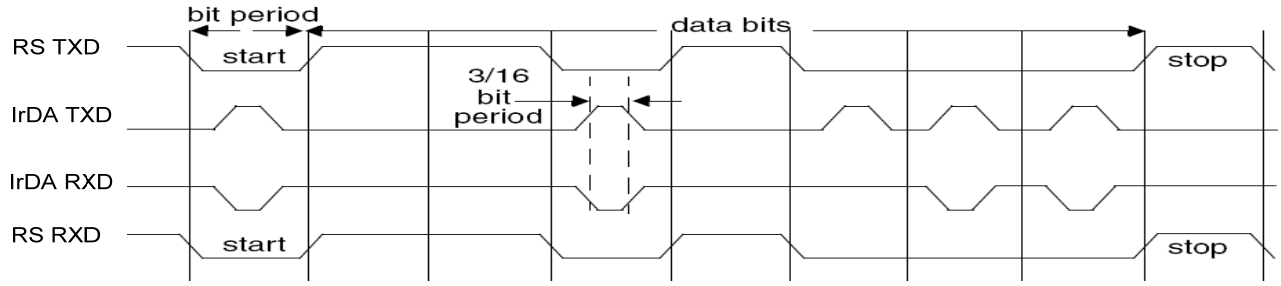


Figure 24-4: IrDA SIR Data Format

As detailed in the IrDA 1.0 SIR, the UART support a low-power reception mode, the reception of SIR pulses of 1.41 microseconds (minimum pulse duration) is possible, as well as nominal 3/16 of a normal serial bit time. Using this low-power reception mode requires programming the Low Power Divisor Latch (LPDLL/LPDLH) registers. It should be noted that for all sclk frequencies greater than or equal to 7.37MHz (and obey the requirements of the Low Power Divisor Latch registers), pulses of 1.41uS are detectable. However there are several values of sclk that do not allow the detection of such a narrow pulse and these are as follows:

SCLK	Low Power Divisor Latch Register Value	Min Pulse Width for Detection *
1.84MHz	1	3.77uS
3.69MHz	2	2.086uS
5.53MHz	3	1.584uS
* 10% has been added to the internal pulse width signal to cushion the effect of pulse reduction due to the synchronization and data integrity logic so that a pulse slightly narrower than these may be detectable.		

When IrDA SIR mode is enabled, the UART operation is similar to when the mode is disabled, with one exception; data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled. This is because the IrDA SIR physical layer specifies a minimum of 10ms delay between transmission and reception. This 10ms delay must be generated by software.

Interrupts

The assertion of the UART interrupt output signal (intr) occurs whenever one of the several prioritized interrupt types are enabled and active. The following interrupt types can be enabled with the IER register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)
- Busy Detect Indication

These interrupt types are covered in more detail in the register description section.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control is not selected, none of the corresponding logic is implemented and the mode cannot be enabled, reducing overall gate counts. When Auto Flow Control mode has been selected it can be enabled with the Mode Control Register (MCR[5]). Figure 24-5 shows a block diagram of the Auto Flow Control functionality.

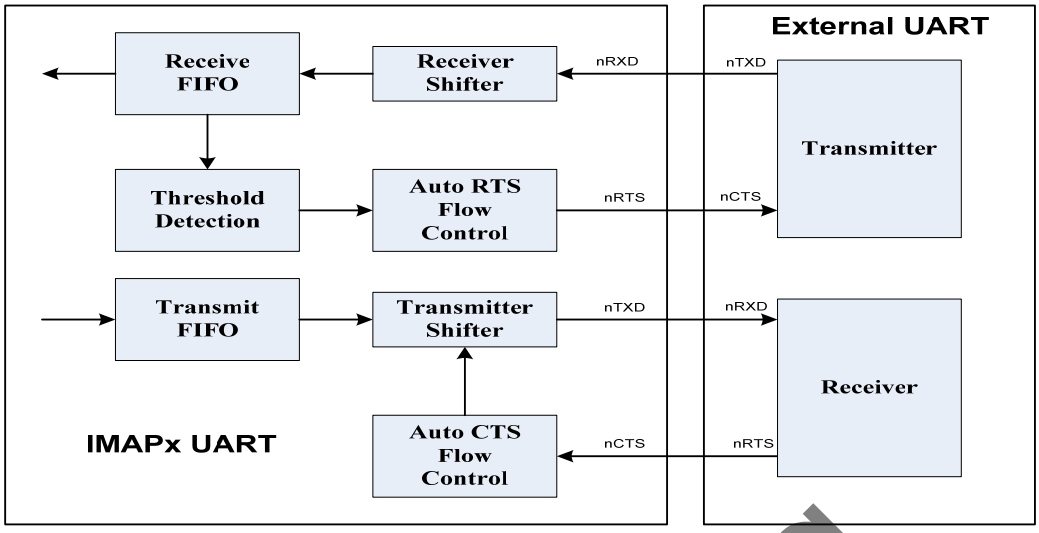


Figure 24-5: Auto Flow Control Block Diagram

Auto RTS and Auto CTS are described as follows:

Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

When Auto RTS is enabled (active), the nRTS output is forced inactive (high) when the receiver FIFO level reaches the threshold set by FCR[7:6]. When nRTS is connected to the nCTS input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space (until it is completely empty).

The selectable receiver FIFO threshold values are: 1, ¼, ½, and "2 less than full". Since one additional character may be transmitted to the UART after nRTS has become inactive (due to data already having entered the transmitter block in the other UART), setting the threshold to "2 less than full" allows maximum use of the FIFO with a safety zone of one character.

Once the receiver FIFO becomes completely empty by reading the Receiver Buffer Register (RBR), nRTS again becomes active (low), signalling the other UART to continue sending data.

It is important to note that even if everything else is selected and the correct MCR bits are set, if the FIFOs are disabled through FCR[0] or the UART is in SIR mode (MCR[6] is set to one), Auto Flow Control is also disabled. When Auto RTS is not implemented or disabled, nRTS is controlled solely by MCR[1]. **Error! Reference source not found.** shows a timing diagram of Auto RTS operation.

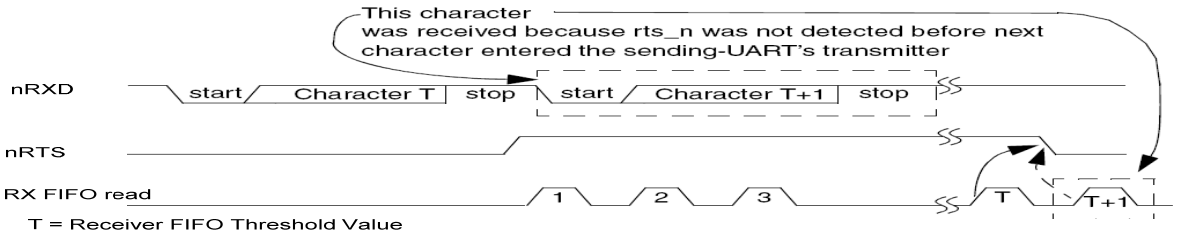


Figure 24-6: Auto RTS Timing

Auto CTS – becomes active when the following occurs:

- ❍ Auto Flow Control is selected during configuration
- ❍ FIFOs are implemented
- ❍ AFCE (MCR[5] bit is set)
- ❍ FIFOs are enabled through FIFO Control Register FCR[0] bit
- ❍ SIR mode is disabled (MCR[6] bit is not set)

When Auto CTS is enabled (active), the UART transmitter is disabled whenever the nCTS input becomes inactive (high). This prevents overflowing the FIFO of the receiving UART.

If the nCTS input is not inactivated before the middle of the last stop bit, another character is transmitted before the transmitter is disabled. While the transmitter is disabled, the transmitter FIFO can still be written to, and even overflowed.

Therefore, when using this mode, the following happens:

- The UART status register can be read to check if the transmit FIFO is full (USR[1] set to zero),
- The current FIFO level can be read via the TFL register, or
- The Programmable THRE Interrupt mode must be enabled to access the "FIFO full" status via the Line Status Register (LSR).

When using the "FIFO full" status, software can poll this before each write to the Transmitter FIFO. When the nCTS input becomes active (low) again, transmission resumes. It is important to note that even if everything else is selected, if the FIFOs are disabled via FCR[0], Auto Flow Control is also disabled. When Auto CTS is not implemented or disabled, the transmitter is unaffected by nCTS. A Timing Diagram showing Auto CTS operation can be seen in Figure 24-7.

Figure 24-7: Auto CTS Timing

Programmable THRE Interrupt

The UART can be configured to have a Programmable THRE Interrupt mode available to increase system performance. If FIFOs are not implemented, then this mode cannot be selected. When Programmable THRE Interrupt mode is not selected, none of the logic is implemented and the mode cannot be enabled, reducing the overall gate counts.

When Programmable THRE Interrupt mode is selected it can be enabled via the Interrupt Enable Register (IER[7]). When FIFOs and the THRE Mode are implemented and enabled, THRE Interrupts (when also enabled) and dma_tx_req are active at, and below, a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in followed Figure.

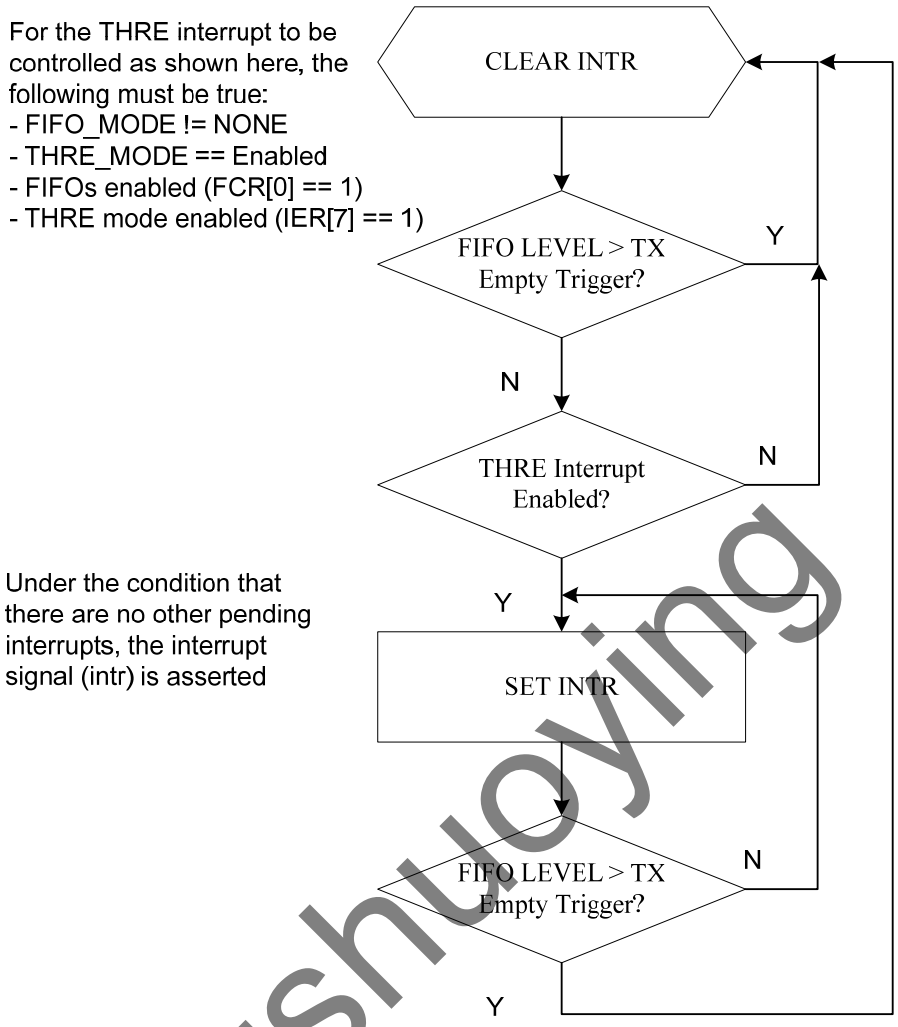


Figure 24-8: Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode

This threshold level is programmed into FCR[5:4]. The available empty thresholds are: empty, 2, ¼ and ½. See Register description section for threshold setting details. Selection of the best threshold value depends on the system's ability to begin a new transmission sequence in a timely manner. However, one of these thresholds should prove optimum in increasing system performance by preventing the transmitter FIFO from running empty.

In addition to the interrupt change, Line Status Register (LSR[5]) also switches function from indicating transmitter FIFO empty, to FIFO full. This allows software to fill the FIFO each transmit sequence by polling LSR[5] before writing another character. The flow then becomes, “fill transmitter FIFO whenever an interrupt occurs and there is data to transmit”, instead of waiting until the FIFO is completely empty. Waiting until the FIFO is empty causes a performance hit whenever the system is too busy to respond immediately. Further system efficiency is achieved when this mode is enabled in combination with Auto Flow Control.

Even if everything else is selected and enabled, if the FIFOs are disabled via FCR[0], the Programmable THRE Interrupt mode is also disabled. When not selected or disabled, THRE interrupts and LSR[5] function normally (both reflecting an empty THR or FIFO). The flowchart of THRE interrupt generation when not in programmable THRE interrupt mode is shown in following Figure.

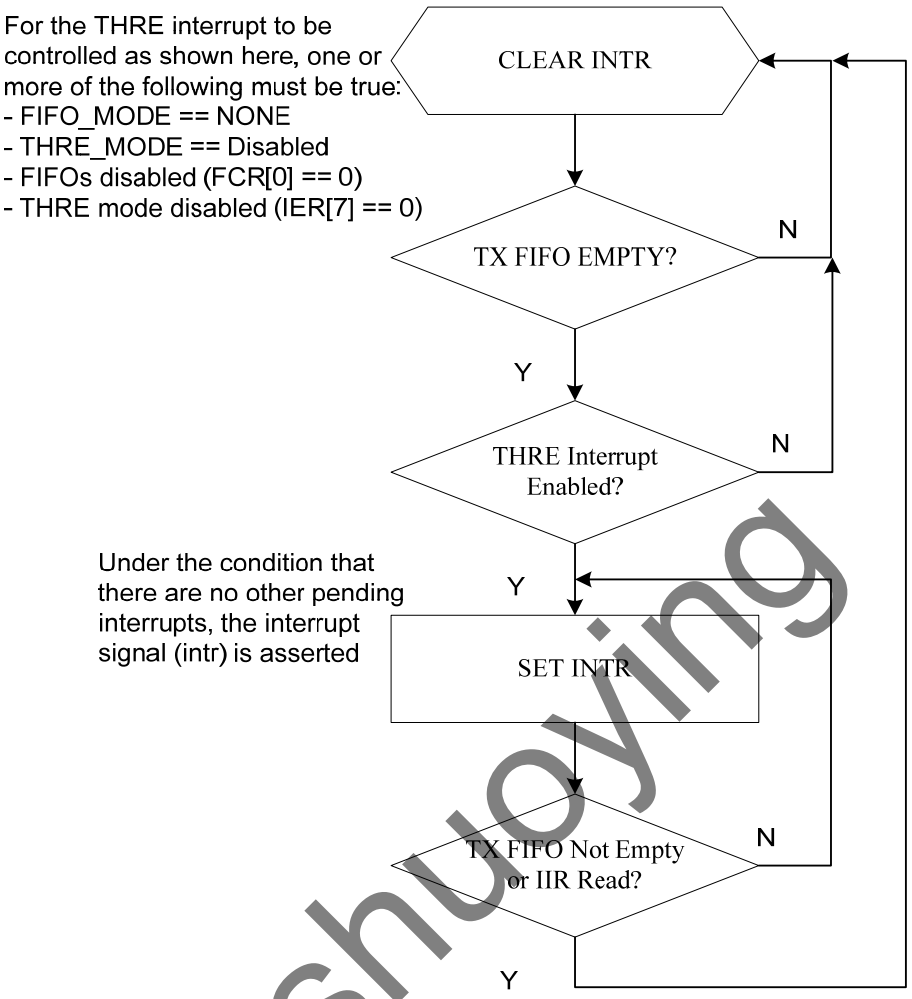


Figure 24-9: Flowchart of Interrupt Generation When not in Programmable THRE Interrupt Mode

DMA mode

The UART has special DMA interface signals connected with DMA controller. There are dma_tx_req, dma_tx_single, dma_tx_ack, dma_rx_req, dma_rx_single and dma_rx_ack, which can be seen in Figure 24-1. The dma_tx_req signal is active high and can be asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

Also, the dma_rx_req signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode

With the presence of the special handshaking signals the UART does not have to rely on internal status and level values to recognize the completion of a request and hence remove the request. Instead, the de-assertion of the DMA transmit and receive request is controlled by the assertion of the DMA transmit and receive acknowledge respectively.

24.3 UART Register Description

24.3.1 UART Register Memory Map

Table 24-1: UART Register Memory Map

Note: IMAPx210 has four UART, where n is the number of UART and range from 0 to 3.

Address	Symbol	Direction	Description
0x20E2_0000+n*0x1000	RBRn	R	Receive Buffer Register when LCR[7] bit = 0
	THRn	W	Transmit Holding Register when LCR[7] bit = 0
	DLLn	R/W	Divisor Latch (Low) when LCR[7] bit = 1
0x20E2_0004+n*0x1000	DLHn	R/W	Divisor Latch (High) when LCR[7] bit = 1
	IERn	R/W	Interrupt Enable Register when LCR[7] bit = 0
0x20E2_0008+n*0x1000	IIRn	R	Interrupt Identification Register
	FCRn	W	FIFO Control Register
0x20E2_000C+n*0x1000	LCRn	R/W	Line Control Register
0x20E2_0010+n*0x1000	MCRn	R/W	Mode Control Register
0x20E2_0014+n*0x1000	LSRn	R	Line Status Register
0x20E2_0018+n*0x1000	CSRn	R	CTS Status Register
0x20E2_0020+n*0x1000	LPDLLn	R/W	Low Power Divisor Latch (Low) Register
0x20E2_0024+n*0x1000	LPDLHn	R/W	Low Power Divisor Latch (High) Register
0x20E2_007C+n*0x1000	USRn	R	UART Status Register
0x20E2_0080+n*0x1000	TFLn	R	Transmit FIFO Level
0x20E2_0084+n*0x1000	RFLn	R	Receive FIFO Level
0x20E2_00A4+n*0x1000	HTXn	R/W	UART Halt TX Register
0x20E2_00A8+n*0x1000	DMASAn	W	UART DMA Software Acknowledge Register
0x20E2_0100+n*0x1000	CKSRn	R/W	UART Clock Source Register

24.3.2 UART Registers and Field Descriptions

Register 24-1: UART Receive Buffer Register （RBRn）

Field	Symbol	Direction	Description	Default
[7:0]	RBR	R	Data byte received on the serial input port in UART mode or infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is pReserved, but any incoming data are lost and an overrun error occurs. **Note:** This register only can be accessed when LCR[7] bit = 0.	0

Register 24-2: UART Transmit Hold Register （THRn）

Field	Symbol	Direction	Description	Default
[7:0]	THR	W	Data to be transmitted on the serial output port in UART mode or infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single	0

			character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost. **Note:** This register only can be accessed when LCR[7] bit = 0.	

Register 24-3: UART Divisor Latch High Register （DLHn）

Field	Symbol	Direction	Description	Default
[7:0]	DLH	R/W	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero); The output baud rate is equal to the select serial clock (PCLK or UEXTCLK) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). When the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data. **Note:** This register only can be accessed when LCR[7] bit = 1.	0

Register 24-4: UART Divisor Latch Low Register （DLLn）

Field	Symbol	Direction	Description	Default
[7:0]	DLL	R/W	Low 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero); The output baud rate is equal to the select serial clock (PCLK or UEXTCLK) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). When the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data. **Note:** This register only can be accessed when LCR[7] bit = 1.	0

Register 24-5: UART Interrupt Enable Register （IERn）

Field	Symbol	Direction	Description	Default
[7]	PTIME	R/W	This bit is used to enable/disable the generation of THRE Interrupt when Programmable THRE Interrupt Mode Enable. 0 = disabled 1 = enabled	0

[2]	ELSI	R/W	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled	0
[1]	ETBEI	R/W	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled	0
[0]	ERBFI	R/W	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled	0

Note: This register only can be accessed when LCR[7] bit = 0.

Register 24-6: UART Interrupt Identity Register （IIRn）

Field	Symbol	Direction	Description	Default
[7:6]	FIFOSE	R	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled	0
[3:0]	IID	R	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout The interrupt priorities are split into four levels that are detailed in followed table. Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.	0001

Table 24-2: Interrupt Control Functions

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0001	--	None	None	--
0110	Highest	Receiver line status	Overrun/parity/framing errors or break interrupt	Reading the line status register
0100	Second	Received data available	Receiver data available(FIFOs disable) or RCVR FIFO trigger level reached (FIFOs enabled)	Reading the receiver buffer register (FIFOs disabled) or the FIFO drops below the trigger level (FIFOs enabled)
1100	Second	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time	Reading the receiver buffer register
0010	Third	Transmit holding Register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode disabled) or XMIT

			XMIT FIFO at or below threshold (Prog. THRE Mode enabled)	FIFO above threshold (FIFOs and THRE Mode enabled).
0111	Fourth	Busy detect indication	master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one).	Reading the UART status register

Register 24-7: UART FIFO Control Register （FCRn）

Field	Symbol	Direction	Description	Default
[7:6]	RT	W	This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the nRTS signal is de-asserted. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full	0
[5:4]	TET	W	This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full	0
[2]	XFIFOR	W	XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.	0
[1]	RFIFOR	W	RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.	0
[0]	FIFOE	W	FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.	0

Register 24-8: UART Line Control Register （LCRn）

Field	Symbol	Direction	Description	Default
[7]	DLAB	R/W	Divisor Latch Access Bit, writeable only when UART is not busy (USR[0] is zero), otherwise always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.	0
[6]	Break	R/W	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the RS nTXD line is forced low until the Break bit is cleared.	0

			If SIR_MODE == Enabled and active (MCR[6] set to one) the IrDA nTXD line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the IrDA nTXD line is forced low.	
[4]	EPS	R/W	Even Parity Select, writeable only when UART is not busy (USR[0] is zero), otherwise always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.	0
[3]	PEN	R/W	Parity Enable, writeable only when UART is not busy (USR[0] is zero), otherwise always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled	0
[2]	STOP	R/W	Number of stop bits, writeable only when UART is not busy (USR[0] is zero), otherwise always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit	0
[1:0]	DLS	R/W	Data Length Select, writeable only when UART is not busy (USR[0] is zero), otherwise always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	0

Register 24-9: UART Mode Control Register （MCRn）

Field	Symbol	Direction	Description	Default
[6]	SIRE	R/W	SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode features as described in “IrDA 1.0 SIR Protocol”. 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled	0
[5]	ACFE	R/W	Auto Flow Control Enable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in “Auto Flow Control”. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled	0
[4]	LB	R/W	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode(MCR[6] set to zero), data on the RS nTXD line is held high, while serial data output is looped back to the nRXD line, internally. In this mode all the interrupts are fully functional.	0

			If operating in infrared mode (MCR[6] set to one), data on the IrDA nTXD line is held low, while serial data output is inverted and looped back to the nRXD line.	
[1]	RTS	R/W	Request to Send. This is used to directly control the Request to Send (nRTS) output. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the nRTS signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control and FIFOs enable (FCR[0] set to one), the nRTS output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (nRTS is inactive high when above the threshold). The nRTS signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the nRTS output is held inactive high while the value of this location is internally looped back to an input.	0

Register 24-10: UART Line Status Register （LSRn）

Field	Symbol	Direction	Description	Default
[7]	RFE	R	Receiver FIFO Error bit when FIFO enable. This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.	0
[6]	TEMT	R	Transmitter Empty bit. If FIFOs are enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.	1
[5]	THRE	R	Transmit Holding Register Empty bit If THRE mode is disabled (IER[7] set to zero), this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur. If the THRE Interrupt is enabled and also FIFO enable, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting. For more details, see “Programmable THRE Interrupt”.	1
[4]	BI	R	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, RS nRXD, is held in a logic '0' state for longer than the sum of start time + data bits +parity + stop bits. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, IrDA nRXD, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits.	0

			A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.	
[3]	FE	R	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit.	0
[2]	PE	R	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit.	0
[1]	OE	R	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.	0
[0]	DR	R	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.	0

Register 24-11: UART CTS Status Register （CSRn）

Field	Symbol	Direction	Description	Default
[4]	CTS	R	Clear to Send. This is used to indicate the current state of input control line nCTS. This bit is the complement of nCTS. When the Clear to Send input (nCTS) is asserted it is an indication that external UART is ready to exchange data with the UART. 0 = nCTS input is de-asserted (logic 1) 1 = nCTS input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).	0

Register 24-12: UART Low Power Divisor Latch Low Register （LPDLLn）

Field	Symbol	Direction	Description	Default
[7:0]	LPDLL	R/W	This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero) The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. **NOTE:** When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.	0

Register 24-13: UART Low Power Divisor Latch High Register （LPDLHn）

Field	Symbol	Direction	Description	Default
[7:0]	LPDLH	R/W	This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero) The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. **Note:** When the Low Power Divisor Latch registers (LPDLL and LPDLH)	0

			are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.	

Register 24-14: UART Status Register （USRn）

Field	Symbol	Direction	Description	Default
[4]	RFF	R	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.	0
[3]	RFNE	R	Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.	0
[2]	TFE	R	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.	1
[1]	TFNF	R	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.	1
[0]	BUSY	R	UART Busy. This bit is valid when a serial transfer is in progress; when cleared, indicates that the UART is idle or inactive. 0 = UART is idle or inactive 1 = UART is busy (actively transferring data) NOTE: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the UART has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the UART. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If UEXTCLK is selected, the assertion of this bit is also delayed by several cycles of UEXTCLK.	0

Register 24-15: UART Transmit FIFO Level Register （TFLn）

Field	Symbol	Direction	Description	Default
[5:0]	TFL	R	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.	0

Register 24-16: UART Receive FIFO Level Register （RFLn）

Field	Symbol	Direction	Description	Default
[5:0]	RFL	R	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.	0

Register 24-17: UART Halt TX Register （HTXn）

Field	Symbol	Direction	Description	Default
[0]	HTX	R/W	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are enabled. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.	0

Register 24-18: UART DMA Software Acknowledge Register （DMASAn）

Field	Symbol	Direction	Description	Default
[0]	DMASA	W	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the UART should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.	0

Register 24-19: UART Clock Source Register （CKSRn）

Field	Symbol	Direction	Description	Default
[1:0]	CKSEL	R/W	UART serial clock (sclk) source configuration register. 00 : sclk is PCLK 01 : sclk is UEXTCLK (from system clock generator)	0

25 SPI

25.1 Overview

The IMAPx210 Serial Peripheral Interface (SPI) can interface with the serial data transfer. SPI has two 8-bit shift registers for transmission and receiving, respectively. During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). 8-bit serial data at a frequency is determined by its corresponding control register settings. If you only want to transmit, receive data can be kept dummy. Otherwise, if you only want to receive, you should transmit dummy '1' data.

There are 4 I/O pin signals associated with SPI transfers: SCK (SPICLK0,1), MISO (SPIMISO0,1) data line, MOSI (SPIMOSI0,1) data line and active low /SS (SPICS0,1) pin (input).

Features

— SPI Protocol (ver. 2.11) compatible
— 8-bit Shift Register for transmit
— 8-bit Shift Register for receive
— 8-bit Prescaler logic
— Polling and Interrupt transfer mode

25.2 Block Diagram

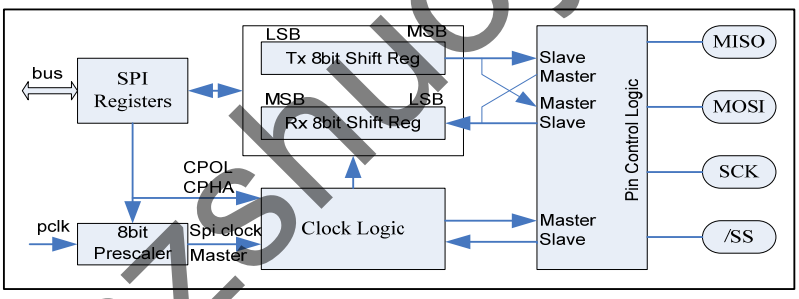


Figure 25-1: SPI Block Diagram

25.3 SPI Operation

Using the SPI interface, IMAPx210 can send/receive 8-bit data simultaneously with an external device. A serial clock line is synchronized with the two data lines for shifting and sampling of the information. When the SPI is the master, transmission frequency can be controlled by setting the appropriate bit in SPPREn register. You can modify its frequency to adjust the baud rate data register value. When the SPI is a slave, other master supplies the clock. When the programmer writes byte data to SPTDATn register, SPI transmit/receive operation will start simultaneously. In some cases, nSS should be activated before writing byte data to SPTDATn.

Programming Procedure

When a byte data is written into the SPTDATn register, SPI starts to transmit if ENSCK and MSTR of SPCONn register are set. You can use a typical programming procedure to operate an external SPI card.

To program the SPI modules, follow these basic steps:

1. Set Baud Rate Prescaler Register (SPPREn).
2. Set SPCONn to configure properly the SPI module.

3. Write data 0xFF to SPTDATn 10 times in order to initialize external card.
4. Set a GPIO pin, which acts as nSS, low to activate external card.
5. Tx data: Check the status of Transfer Ready flag (REDY=1), and then write data to SPTDATn.
6. Rx data(1): SPCONn's TAGD bit disable = normal mode
7. Write 0xFF to SPTDATn, then confirm REDY to set, and then read data from Read Buffer.
8. Rx data(2): SPCONn's TAGD bit enable = Tx Auto Garbage Data mode
9. Confirm REDY to set, and then read data from Read Buffer (then automatically start to transfer).
10. Set a GPIO pin, which acts as nSS, high to deactivate external card.

25.4 SPI Transfer Format

The IMAPx210 SPI supports 4 different formats to transfer data. The followed figure shows the four waveforms for SPICLK.

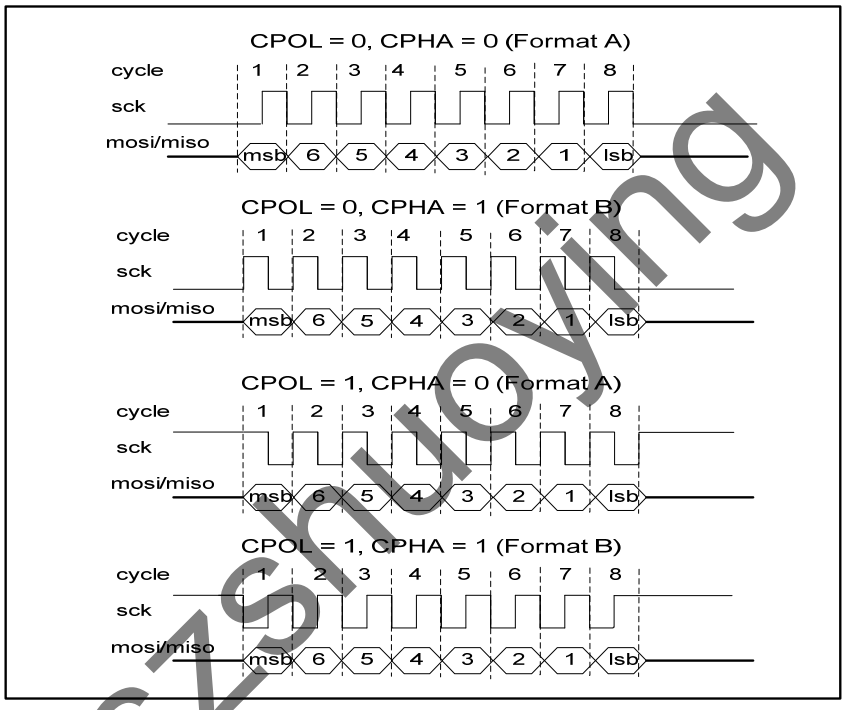


Figure 25-2: SPI Transfer Format

25.5 Register Description

25.5.1 SPI Register Memory Map

Table 25-1: SPI Register Memory Map

Address	Symbol	Direction	Description
0x20DF_4000	SPCON	R/W	SPI control register
0x20DF_4004	SPSTA	RC	SPI status register
0x20DF_4008	SPPIN	R/W	SPI pin control register
0x20DF_400C	SPPRE	R/W	SPI baud rate prescaler register
0x20DF_4010	SPTDAT	R/W	SPI Tx data register
0x20DF_4014	SPRDAT	R	SPI Rx data register

25.5.2 SPI Register and Field Descriptions

Register 25-1: SPI Control Register（SPICON）

Field	Symbol	Direction	Description	Default
[6:5]	SMOD	R/W	Determine how SPTDAT is read/written 00 = polling mode 01 = interrupt mode 1x = Reserved	00
[4]	ENSCK	R/W	Determine whether you want SCK enabled or not (master only). 0 = disable 1 = enable	0
[3]	MSTR	R/W	Determine the desired mode (master or slave). 0 = slave 1 = master Note: In slave mode, there should be set up time for master to initiate Tx/Rx.	0
[2]	CPOL	R/W	Determine an active high or active low clock. 0 = active high 1 = active low	0
[1]	CPHA	R/W	Select one of the two fundamentally different transfer format 0 = format A 1 = format B	0
[0]	TAGD	R/W	Decide whether the receiving data is required or not. 0 = normal mode 1 = Tx auto garbage data mode Note: In normal mode, if you only want to receive data, you should transmit dummy 0xFF data.	0

Register 25-2: SPI Status Register（SPSTA）

Field	Symbol	Direction	Description	Default
[2]	DCOL	RC	This flag is set if the SPTDATn is written or SPRDATn is read while a transfer is in progress and cleared by reading the SPSTAn. 0 = Not detect 1 = Collision error detect	0
[1]	MULF	RC	This flag is set if the nSS signal goes to active low while the SPI is configured as a master, and SPPINn's ENMUL bit is multi master error detect mode. MULF is cleared by reading SPSTAn. 0 = Not detect 1 = Multi master error detect	0
[0]	REDY	R	This bit indicates that SPTDATn or SPRDATn is ready to transmit or receive. This flag is automatically cleared by writing data to SPTDATn or reading SPRDATn in tx auto garbage mode. 0 = Not ready 1 = Data Tx/Rx ready	1

Register 25-3: SPI Pin Control Register（SPPIN）

Field	Symbol	Direction	Description	Default
[2]	ENMUL	R/W	The nSS pin is used as an input to detect multi master error when the SPI system is a master. 0 = Disable (general purpose), 1 = Multi master error detect enable	0
[0]	KEEP	R/W	Determine MOSI drive or release when 1byte transmit is completed (master only). 0 = Release, 1 = Drive the previous level	0

Register 25-4: SPI Baud Rate Prescaler Register（SPPRE）

Field	Symbol	Direction	Description	Default
[7:0]	PV	R/W	prescaler value(PV) determine SPI clock rate.Baud rate = PCLK / 2 / (PV + 1)	0

Register 25-5: SPI Tx Data Register（SPTDAT）

Field	Symbol	Direction	Description	Default
[7:0]	TDAT	R/W	This field contains the data to be transmitted over the SPI channel	0

Register 25-6: SPI Rx Data Register（SPRDAT）

Field	Symbol	Direction	Description	Default
[7:0]	RDAT	R	This field contains the data to be received over the SPI channel	0

26 SSSI

26.1 Overview

SSSI is a slave-synchronous serial interface. CPU accesses data, control, and status information on the SSSI through the APB interface. The SSSI can connect to any serial-master peripheral device using one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI)
- Texas Instruments Serial Protocol (SSP)
- National Semiconductor Microwire

Features

- Serial-slave operation – Enables serial communication with serial-master peripheral devices.
- DMA Controller Interface – SSSI interface to a DMA controller over the system bus using a handshaking interface for transfer requests.
- Independent masking of interrupts – Transmit FIFO overflow, transmit FIFO empty, receive FIFO full, receive FIFO underflow, and receive FIFO overflow interrupts can all be masked independently.

Block Diagram

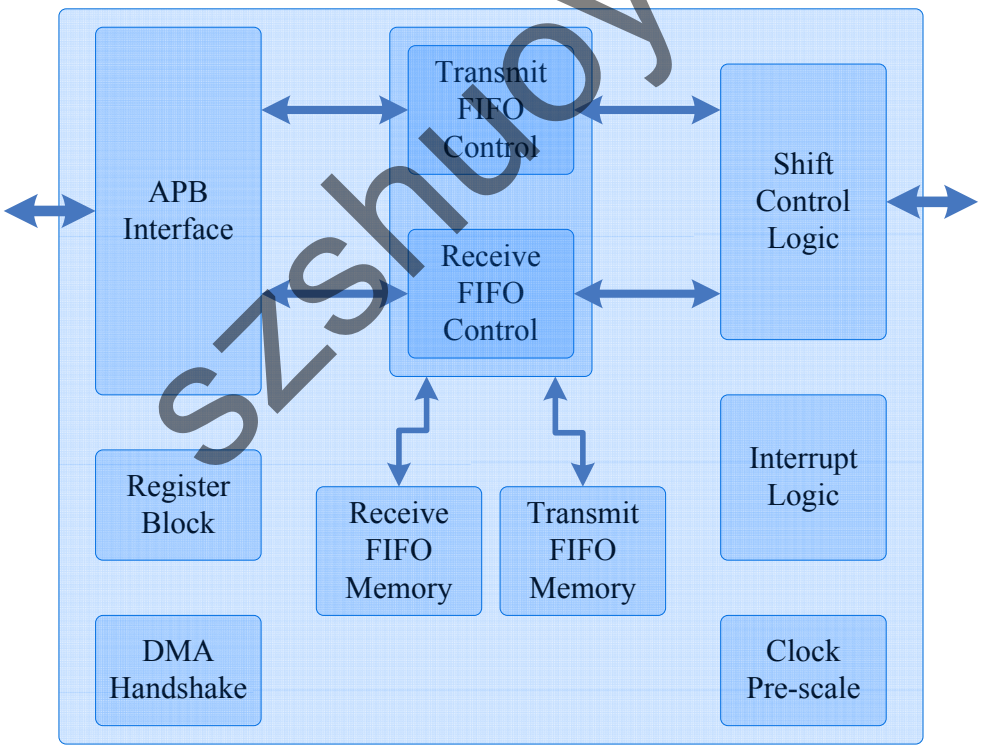


Figure 26-1: SSSI Block Diagram

Figure 26-1 shows the functional block diagram of SSSI, The interface connected with system bus is APB BUS, with Transmit FIFO and Receive FIFO controlled, SSSI can transfer data from/to system and external peripheral device. SSSI has single DMA hard handshake interface and can communicate with DMA directly.

26.2 Functional Description

In order for the SSSI to connect to a serial-master peripheral device, the peripheral must have at least one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI) - A four-wire, full-duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of the slave select signal or the first edge of the serial clock. The slave select line is held high when the SSSI is idle or disabled.

- Texas Instruments Serial Protocol (SSP) - A four-wire, full-duplex serial protocol. The slave select line used for SPI and Microwire protocols doubles as the frame indicator for the SSP protocol.

- National Semiconductor Microwire – A half-duplex serial protocol, which uses a control word transmitted from the serial master to the target serial slave.

User can program the FRF (frame format) bit field in the Control Register 0 (CTRLR0) to select which protocol is used.

The serial protocols supported by the SSSI allow for serial slaves to be selected or addressed using either hardware or software. When implemented in hardware, serial slaves are selected under the control of dedicated hardware select lines. The number of select lines generated from the serial master is equal to the number of serial slaves present on the bus. The serial-master device asserts the select line of the target serial slave before data transfer begins.

When implemented in software, the input select line of each serial slave can originate either from a single-slave select output signal on the serial master (user must configure the master to have one slave select output) or be permanently grounded. The main program in the software domain controls selection of the target slave device; Software would use the SSIENR register in all slaves in order to control which slave is to respond to the serial transfer request from the master device.

26.2.1 Transfer Modes

The SSSI operates in the following four modes when transferring data on the serial bus:

- “Transmit and Receive”
- “Transmit Only”
- “Receive Only”

The transfer mode (TMOD) is set by writing to control register 0 (CTRLR0), as described in “CTRLR0”.

Transmit and Receive

When TMOD = 2‘b00, both transmit and receive logic are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.

Transmit Only

When TMOD = 2‘b01, the receive data are invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

Receive Only

When TMOD = 2‘b10, the transmit data are invalid. The transmit FIFO is never popped in Receive Only mode. Data from a previous transfer is retransmitted from the shift register. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.

26.2.2 Operation Mode

This mode enables serial communication with master peripheral devices. All serial transfers are initiated and controlled by the serial bus master.

When the SSSI is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line, driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

When the SSSI serial slave is not selected, it must not interfere with data transfers between the serial-master and other serial-slave devices. When the serial slave is not selected, its rxd output is buffered, resulting in a high impedance drive onto the serial master rxd line.

The serial clock that regulates the data transfer is generated by the serial-master device and input to the SSSI slave on sclk_in. The slave remains in an idle state until selected by the bus master. When not actively transmitting data, the slave must hold its txd line in a high impedance state to avoid interference with serial transfers to other slave devices. The ssi_oe_n line is available for use to control the txd output buffer. The slave continues to transfer data to and from the master device as long as it is selected. If the master transmits to all serial slaves, a control bit (SLV_OE) in the SSSI control register 0 (CTRLR0) can be programmed to inform the slave if it should respond with data from its txd line.

Slave SPI and SSP Serial Transfers

If the SSSI is Receive only (TMOD=10), the transmit FIFO need not contain valid data because the data currently in the transmit shift register is resent each time the slave device is selected.

The TXE error flag in the status register (SR) is not set when TMOD=01. You should mask the transmit FIFO empty interrupt when this mode is used.

If the SSSI transmits data to the master, you must ensure that data exists in the transmit FIFO before a transfer is initiated by the serial-master device. If the master initiates a transfer to the SSSI when no data exists in the transmit FIFO, an error flag (TXE) is set in the SSSI status register, and the previously transmitted data frame is resent on txd. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data have been transmitted. The transmit FIFO threshold level register (TXFTLR) can be used to early interrupt the processor, indicating that the transmit FIFO buffer is nearly empty. When a DMA Controller is used for APB accesses, the DMA transmit data level register (DMATDLR) can be used to early request the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer.

The receive FIFO buffer should be read each time receive FIFO generates a FIFO full interrupt request to prevent an overflow. The receive FIFO threshold level register (RXFTLR) can be used to give early indication that the receive FIFO is nearly full. When a DMA Controller is used for APB accesses, the DMA receive data level register (DMARDLR) can be used to early request the DMA controller, indicating that the receive FIFO is nearly full.

A typical software flow for completing a continuous serial transfer from a serial master to the SSSI is described as follows:

1. If the SSSI is enabled, disable it by writing 0 to SSIENR.

2. Set up the SSSI control registers for the transfer. These registers can be set in any order.
   - Write CTRLR0 (for SPI transfers SCPH and SCPOL must be set identical to the master device).
   - Write TXFTLR and RXFTLR to set FIFO threshold levels.
   - Write the IMR register to set up interrupt masks.
3. Enable the SSSI by writing 1 to the SSIENR register.
4. If the transfer mode is transmit and receive (TMOD=2'b00) or transmit only (TMOD=2'b01), write data for transmission to the master into the transmit FIFO (Write DR).

   If the transfer mode is receive only (TMOD=2'b10), there is no need to write data into the transmit FIFO; the current value in the transmit shift register is retransmitted.
5. The SSSI slave is now ready for the serial transfer. The transfer begins when the SSSI slave is selected by a serial-master device.
6. When the transfer is underway, the BUSY status can be polled to return the transfer status. If a transmit FIFO empty interrupt request is made, write the transmit FIFO (write DR). If a receive FIFO full interrupt request is made, read the receive FIFO (read DR).
7. The transfer ends when the serial master removes the select input to the SSSI. When the transfer is completed, the BUSY status is reset to 0.
8. If the transfer mode is not transmit only (TMOD $\neq$ 01), read the receive FIFO until empty.
9. Disable the SSSI by writing 0 to SSIENR.

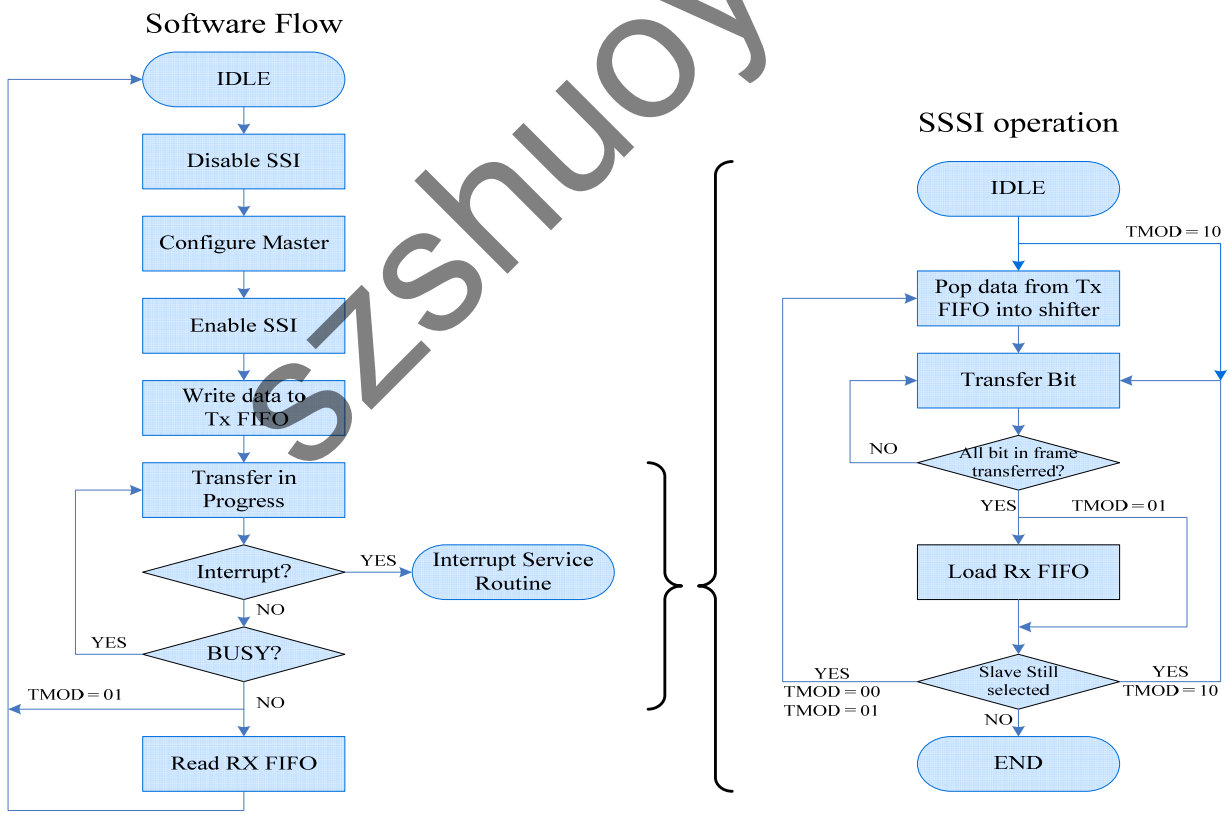


Figure 26-2: SSSI Slave SPI/SSP Transfer Flow

Slave Microwire Serial Transfers

When the SSSI is connected as a Microwire protocol slave device, the Microwire protocol operates in much the same way as the SPI protocol. There is no decoding of the control frame by the SSSI device.

26.2.3 DMA Controller Interface

The SSSI has built-in DMA capability. It has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA. While the SSSI DMA operation is designed in a generic way to fit DMA controller as easily as possible, it is designed to work seamlessly, and best used, with the GDMA. The settings of the GDMA that are relevant to the operation of the SSSI are discussed here, mainly bit fields in the GDMA channel control register, CTLx, where x is the channel number.

The SSSI uses two DMA channels, one for the transmit data and one for the receive data. The SSSI has these DMA registers:

- "DMACR" – Control register to enable DMA operation.
- "DMATDLR"– Register to set transmit the FIFO level at which a DMA request is made.
- "DMARDLR" – Register to set the receive FIFO level at which a DMA request is made.

As a block flow control device, the DMA Controller is programmed by the processor with the number of data items (block size) that are to be transmitted or received by the SSSI; this is programmed into the BLOCK_TS field of the CTLx register.

The block is broken into a number of transactions, each initiated by a request from the SSSI.

The DMA Controller must also be programmed with the number of data items (in this case, SSSI FIFO entries) to be transferred for each DMA request. This is also known as the burst transaction length, and is programmed into the SRC_MSIZE/DEST_MSIZE fields of the GDMA CTLx register for source and destination, respectively.

Transmit Watermark Level and Transmit FIFO Underflow

During SSSI transfers, transmit FIFO requests are made to the GDMA whenever the number of entries in the transmit FIFO is less than or equal to the DMA Transmit Data Level Register (DMATDLR) value; this is known as the watermark level. The GDMA responds by writing a burst of data to the transmit FIFO buffer, of length CTLx.DEST_MSIZE.

Data should be fetched from the DMA often enough for the transmit FIFO to perform serial transfers continuously; that is, when the FIFO begins to empty another DMA request should be triggered. Otherwise the FIFO will run out of data (underflow). To prevent this condition, the user must set the watermark level correctly.

Receive Watermark Level and Receive FIFO Overflow

During SSSI serial transfers, receive FIFO requests are made to the GDMA whenever the number of entries in the receive FIFO is at or above the DMA Receive Data Level Register; that is, DMARDLR+1. This is known as the watermark level. The GDMA responds by writing a burst of data to the transmit FIFO buffer of length CTLx.SRC_MSIZE.

Data should be fetched by the DMA often enough for the receive FIFO to accept serial transfers continuously; that is, when the FIFO begins to fill, another DMA transfer is requested. Otherwise, the FIFO will fill with data (overflow). To prevent this condition, the user must correctly set the watermark level.

26.3 SSSI Register Description

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 registers: 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.

26.3.1 SSSI Register Memory Map

SSSI's register mapping address range in system is 0x20DF3000~0x20DF3FFF. SSSI register's system accessing address equal to BASE_ADDRESS (0x20DF3000) plus address offset.

Table 26-1: SSSI Register Memory Map

Address	Symbol	Direction	Description
0x0	CTRLR0	RW	Control Register 0
0x04	Reserved	N/A	Reserved
0x08	SSSIENR	RW	SSSI Enable Register
0x0C	MWCR	RW	Microwire Control Register
0x10	Reserved	N/A	Reserved
0x14	BAUDR	RW	Baud Rate Select
0x18	TXFTLR	RW	Transmit FIFO Threshold Level
0x1C	RXFTLR	RW	Receive FIFO Threshold Level
0x20	TXFLR	RW	Transmit FIFO Level Register Width: TX_ABW+1
0x24	RXFLR	RW	Receive FIFO Level Register Width: RX_ABW+1
0x28	SR	R	Status Register
0x2C	IMR	RW	Interrupt Mask Register
0x30	ISR	R	Interrupt Status Register
0x34	RISR	R	Raw Interrupt Status Register
0x38	TXOICR	R	Transmit FIFO Overflow Interrupt Clear Register
0x3C	RXOICR	R	Receive FIFO Overflow Interrupt Clear Register
0x40	RXUICR	R	Receive FIFO Underflow Interrupt Clear Register
0x44	MSTICR	R	Multi-Master Interrupt Clear Register
0x48	ICR	R	Interrupt Clear Register
0x4C	DMACR	RW	DMA Control Register
0x50	DMATDLR	RW	DMA Transmit Data Level
0x54	DMARDLR	RW	DMA Receive Data Level
0x60-0x9C	DR	RW	Data Register

26.3.2 SSSI Registers and Field Descriptions

Register 26-1: Control Register 0 (CTRLR0)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:12]	CFS	RW	**Control Frame Size.** Selects the length of the control word for the Microwire frame format. ● 0000 : Reserved – Reserved operation ● 0001 : Reserved – Reserved operation ● 0010 : Reserved – Reserved operation ● 0011 : 4-bit serial data transfer ● 0100 : 5-bit serial data transfer ● 0101 : 6-bit serial data transfer ● 0110 : 7-bit serial data transfer ● 0111 : 8-bit serial data transfer ● 1000 : 9-bit serial data transfer ● 1001 : 10-bit serial data transfer ● 1010 : 11-bit serial data transfer ● 1011 : 12-bit serial data transfer ● 1100 : 13-bit serial data transfer ● 1101 : 14-bit serial data transfer ● 1110 : 15-bit serial data transfer ● 1111 : 16-bit serial data transfer	0
[11]	SRL	RW	**Shift Register Loop.** Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. ● 0 : Normal Mode Operation ● 1 : Test Mode Operation	0
[10]	SLV_OE	RW	**Slave Output Enable.** Relevant only when the SSSI is configured as a serial-slave device. When configured as a serial master, this bit field has no functionality. This bit enables or disables the setting of the ssi_oe_n output from the SSSI serial slave. When SLV_OE = 1, the ssi_oe_n output can never be active. When the ssi_oe_n output controls the tri-state buffer on the txd output from the slave, a high impedance state is always present on the slave txd output when SLV_OE = 1. This is useful when the master transmits in broadcast mode (master transmits data to all slave devices). Only one slave may respond with data on the master rxd line. This bit is enabled after reset and must be disabled by software (when broadcast mode is used), if you do not want this device to respond with data. ● 0 : Slave txd is enabled ● 1 : Slave txd is disabled	0
[9:8]	TMOD	RW	**Transfer Mode.** Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. ● 00 : Transmit & Receive ● 01 : Transmit Only ● 10 : Receive Only ● 11 : Reserved	0

[7]	SCPOL	RW	**Serial Clock Polarity.** Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SSSI master is not actively transferring data on the serial bus. ● 0 : Inactive state of serial clock is low ● 1 : Inactive state of serial clock is high	0
[7]	SCPH	RW	**Serial Clock Phase.** Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. ● 0: Serial clock toggles in middle of first data bit ● 1: Serial clock toggles at start of first data bit	0
[5:4]	FRF	RW	**Frame Format.** Selects which serial protocol transfers the data. ● 00 : Motorola SPI ● 01 : Texas Instruments SSP ● 10 : National Semiconductors Microwire ● 11 : Reserved	0
[4]	DFS	RW	**Data Frame Size.** Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. ● 0000 : 1-bit control word ● 0001 : 2-bit control word ● 0010 : 3-bit control word ● 0011 : 4-bit control word ● 0100 : 5-bit control word ● 0101 : 6-bit control word ● 0110 : 7-bit control word ● 0111 : 8-bit control word ● 1000 : 9-bit control word ● 1001 : 10-bit control word ● 1010 : 11-bit control word ● 1011 : 12-bit control word ● 1100 : 13-bit control word ● 1101 : 14-bit control word ● 1110 : 15-bit control word ● 1111 : 16-bit control word	7

Register 26-2: SSSI Enable Register (SSSIENR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	SSSI_EN	RW	**SSSI Enable.** Enables and disables all SSSI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the SSSI control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system.	

Register 26-3: Microwire Control Register (MWCR)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1]	MDD	RW	**Microwire Control.** Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the MSSI Macrocell from the external serial device. When this bit is set to 1, the data word is transmitted from the MSSI Macrocell to the external serial device.	0
[0]	MWMOD	RW	**Microwire Transfer Mode.** Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. ● 0 : non-sequential transfer ● 1 : sequential transfer	0

Register 26-4: Baud Rate Select Register (BAUR)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:0]	SCKDV	RW	**SSSI Clock Divider.** The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fssi_clk/SCKDV Where SCKDV is any even value between 2 and 65534.	0

Register 26-5: Transmit FIFO Threshold Level Register (TXFTLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	TFT	RW	**Transmit FIFO Threshold.** Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 16; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. ● 0000 : ssi_txe_intr is asserted when 0 or less data entries are present in transmit FIFO ● 0001 : ssi_txe_intr is asserted when 1 or less data entries are present in transmit FIFO ● 0010 : ssi_txe_intr is asserted when 2 or less data entries are present in transmit FIFO ● 0011 : ssi_txe_intr is asserted when 3 or less data entries are present in transmit FIFO ● 0100 : ssi_txe_intr is asserted when 4 or less data entries are present in transmit FIFO ● 0101 : ssi_txe_intr is asserted when 5 or less data entries are present in transmit FIFO ● 0110 : ssi_txe_intr is asserted when 6 or less data entries are present in transmit FIFO	0

			● 0111 : ssi_txe_intr is asserted when 7 or less data entries are present in transmit FIFO ● 1000 : ssi_txe_intr is asserted when 8 or less data entries are present in transmit FIFO ● 1001 : ssi_txe_intr is asserted when 9 or less data entries are present in transmit FIFO ● 1010 : ssi_txe_intr is asserted when 10 or less data entries are present in transmit FIFO ● 1011 : ssi_txe_intr is asserted when 11 or less data entries are present in transmit FIFO ● 1100 : ssi_txe_intr is asserted when 12 or less data entries are present in transmit FIFO ● 1101 : ssi_txe_intr is asserted when 13 or less data entries are present in transmit FIFO ● 1110 : ssi_txe_intr is asserted when 14 or less data entries are present in transmit FIFO ● 1111 : ssi_txe_intr is asserted when 15 or less data entries are present in transmit FIFO	

Register 26-6: Receive FIFO Threshold Level Register (RXFTLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	RFT	RW	**Receive FIFO Threshold.** Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is 16. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered. ● 0000 : ssi_txe_intr is asserted when 0 or more data entries are present in receive FIFO ● 0001 : ssi_txe_intr is asserted when 1 or more data entries are present in receive FIFO ● 0010 : ssi_txe_intr is asserted when 2 or more data entries are present in receive FIFO ● 0011 : ssi_txe_intr is asserted when 3 or more data entries are present in receive FIFO ● 0100 : ssi_txe_intr is asserted when 4 or more data entries are present in receive FIFO ● 0101 : ssi_txe_intr is asserted when 5 or more data entries are present in receive FIFO ● 0110 : ssi_txe_intr is asserted when 6 or more data entries are present in receive FIFO ● 0111 : ssi_txe_intr is asserted when 7 or more data entries are present in receive FIFO ● 1000 : ssi_txe_intr is asserted when 8 or more data entries are present in receive FIFO ● 1001 : ssi_txe_intr is asserted when 9 or more data entries are present in receive FIFO ● 1010 : ssi_txe_intr is asserted when 10 or more data entries are present in receive FIFO ● 1011 : ssi_txe_intr is asserted when 11 or more data entries are present in receive FIFO ● 1100 : ssi_txe_intr is asserted when 12 or more data entries are present in receive FIFO ● 1101 : ssi_txe_intr is asserted when 13 or more data entries are present in receive FIFO	0

			● 1110 : ssi_txe_intr is asserted when 14 or more data entries are present in receive FIFO ● 1111 : ssi_txe_intr is asserted when 15 or more data entries are present in receive FIFO	

Register 26-7: Transmit FIFO Level Register (TXFLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	TXTFL	RW	**Transmit FIFO Level.** Contains the number of valid data entries in the transmit FIFO.	0

Register 26-8: Receive FIFO Level Register (RXFLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	RXTFL	RW	**Receive FIFO Level.** Contains the number of valid data entries in the receive FIFO.	0

Register 26-9: Status Register (SR)

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	N/A	Reserved	0
[5]	TXE	R	**Transmission Error.** Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read. ● 0 : No error ● 1 : Transmission error	0
[4]	RFF	R	**Receive FIFO Full.** When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. ● 0 : Receive FIFO is not full ● 1 : Receive FIFO is full	0
[3]	RFNE	R	**Receive FIFO Not Empty.** Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. ● 0 : Receive FIFO is empty ● 1 : Receive FIFO is not empty	0
[2]	TFE	R	**Transmit FIFO Empty.** When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. ● 0 : Transmit FIFO is not empty ● 1 : Transmit FIFO is empty	0
[1]	TFNF	R	**Transmit FIFO Not Full.** Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. ● 0 : Transmit FIFO is full ● 1 : Transmit FIFO is not full	0
[0]	BUSY	R	**SSSI Busy Flag.** When set, indicates that a serial transfer is in progress; when cleared indicates that the SSSI is idle or disabled. ● 0 : SSSI is idle or disabled ● 1 : SSSI is actively transferring data	0

Register 26-10: Interrupt Mask Register (IMR)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	N/A	Reserved	0
[4]	RXFIM	RW	**Receive FIFO Full Interrupt Mask** ● 0 – ssi_rxf_intr interrupt is masked ● 1 – ssi_rxf_intr interrupt is not masked	0
[3]	RXOIM	RW	**Receive FIFO Overflow Interrupt Mask** ● 0 – ssi_rxo_intr interrupt is masked ● 1 – ssi_rxo_intr interrupt is not masked	0
[2]	RXUIM	RW	**Receive FIFO Underflow Interrupt Mask** ● 0 – ssi_rxu_intr interrupt is masked ● 1 – ssi_rxu_intr interrupt is not masked	0
[1]	TXOIM	RW	**Transmit FIFO Overflow Interrupt Mask** ● 0 – ssi_txo_intr interrupt is masked ● 1 – ssi_txo_intr interrupt is not masked	0
[0]	TXEIM	RW	**Transmit FIFO Empty Interrupt Mask** ● 0 – ssi_txe_intr interrupt is masked ● 1 – ssi_txe_intr interrupt is not masked	0

Register 26-11: Interrupt Status Register (ISR)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	N/A	Reserved	0
[4]	RXFIS	R	**Receive FIFO Full Interrupt Status** ● 0 = ssi_rxf_intr interrupt is not active after masking ● 1 = ssi_rxf_intr interrupt is full after masking	0
[3]	RXOIS	R	**Receive FIFO Overflow Interrupt Status** ● 0 = ssi_rxo_intr interrupt is not active after masking ● 1 = ssi_rxo_intr interrupt is active after masking	0
[2]	RXUIS	R	**Receive FIFO Underflow Interrupt Status** ● 0 = ssi_rxu_intr interrupt is not active after masking ● 1 = ssi_rxu_intr interrupt is active after masking	0
[1]	TXOIS	R	**Transmit FIFO Overflow Interrupt Status** ● 0 = ssi_txo_intr interrupt is not active after masking ● 1 = ssi_txo_intr interrupt is active after masking	0
[0]	TXEIS	R	**Transmit FIFO Empty Interrupt Status** ● 0 = ssi_txe_intr interrupt is not active after masking ● 1 = ssi_txe_intr interrupt is active after masking	0

Register 26-12: RAW Interrupt Status Register (RISR)

Field	Symbol	Direction	Description	Default
[31:5]	Reserved	N/A	Reserved	0
[4]	RXFIR	R	**Receive FIFO Full Raw Interrupt Status** ● 0 = ssi_rxf_intr interrupt is not active prior to masking ● 1 = ssi_rxf_intr interrupt is active prior to masking Reset	0
[3]	RXOIR	R	**Receive FIFO Overflow Raw Interrupt Status** ● 0 = ssi_rxo_intr interrupt is not active prior to masking ● 1 = ssi_rxo_intr interrupt is active prior masking Reset	0
[2]	RXUIR	R	**Receive FIFO Underflow Raw Interrupt Status** ● 0 = ssi_rxu_intr interrupt is not active prior to masking ● 1 = ssi_rxu_intr interrupt is active prior to masking Reset	0
[1]	TXOIR	R	**Transmit FIFO Overflow Raw Interrupt Status** ● 0 = ssi_txo_intr interrupt is not active prior to masking ● 1 = ssi_txo_intr interrupt is active prior masking Reset	0
[0]	TXEIR	R	**Transmit FIFO Empty Raw Interrupt Status** ● 0 = ssi_txe_intr interrupt is not active prior to masking ● 1 = ssi_txe_intr interrupt is active prior masking Reset	0

Register 26-13: Transmit FIFO Overflow Interrupt Clear Register (TXOICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	TXOICR	R	**Clear Transmit FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.	0

Register 26-14: Receive FIFO Overflow Interrupt Clear Register (RXOICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	RXOICR	R	**Clear Receive FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.	0

Register 26-15: Receive FIFO Underflow Interrupt Clear Register (RXUICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	RXUICR	R	**Clear Receive FIFO Underflow Interrupt.** This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.	0

Register 26-16: Multi-Master Interrupt Clear Register (MSTICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	MSTICR	R	**Clear Multi-Master Contention Interrupt.** This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.	0

Register 26-17: Interrupt Clear Register (ICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	ICR	R	**Clear Interrupts**. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.	0

Register 26-18: DMA Control Register (DMACR)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1]	TDMAE	RW	**Transmit DMA Enable.** This bit enables/disables the transmit FIFO DMA channel. ● 0 = Transmit DMA disabled ● 1 = Transmit DMA enabled	0
[0]	RDMAE	RW	**Receive DMA Enable.** This bit enables/disables the receive FIFO DMA channel ● 0 = Receive DMA disabled ● 1 = Receive DMA enabled	0

Register 26-19: DMA Transmit Data Level Register (DMATDLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0

[3:0]	DMATDL	RW	**Transmit Data Level.** This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the DMA request signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. ● 0000 : DMA request signal is asserted when 0 or less data entries are present in the transmit FIFO ● 0001 : DMA request signal is asserted when 1 or less data entries are present in the transmit FIFO ● 0010 : DMA request signal is asserted when 2 or less data entries are present in the transmit FIFO ● 0011 : DMA request signal is asserted when 3 or less data entries are present in the transmit FIFO ● 0100 : DMA request signal is asserted when 4 or less data entries are present in the transmit FIFO ● 0101 : DMA request signal is asserted when 5 or less data entries are present in the transmit FIFO ● 0110 : DMA request signal is asserted when 6 or less data entries are present in the transmit FIFO ● 0111 : DMA request signal is asserted when 7 or less data entries are present in the transmit FIFO ● 1000 : DMA request signal is asserted when 8 or less data entries are present in the transmit FIFO ● 1001 : DMA request signal is asserted when 9 or less data entries are present in the transmit FIFO ● 1010 : DMA request signal is asserted when 10 or less data entries are present in the transmit FIFO ● 1011 : DMA request signal is asserted when 11 or less data entries are present in the transmit FIFO ● 1100 : DMA request signal is asserted when 12 or less data entries are present in the transmit FIFO ● 1101 : DMA request signal is asserted when 13 or less data entries are present in the transmit FIFO ● 1110 : DMA request signal is asserted when 14 or less data entries are present in the transmit FIFO ● 1111 : DMA request signal is asserted when 15 or less data entries are present in the transmit FIFO	0

Register 26-20: DMA Receive Data Level Register (DMARDLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	DMARDL	RW	**Receive Data Level.** This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, DMA request signal is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1. ● 0000 : DMA request signal is asserted when 0 or more data entries are present in the receive FIFO ● 0001 : DMA request signal is asserted when 1 or more data entries are present in the receive FIFO ● 0010 : DMA request signal is asserted when 2 or more data entries are present in the receive FIFO ● 0011 : DMA request signal is asserted when 3 or more data entries are present in the receive FIFO ● 0100 : DMA request signal is asserted when 4 or more data entries are present in the receive FIFO ● 0101 : DMA request signal is asserted when 5 or more data entries are present in the receive FIFO ● 0110 : DMA request signal is asserted when 6 or more data entries are present in the receive FIFO	0

			● 0111 : DMA request signal is asserted when 7 or more data entries are present in the receive FIFO ● 1000 : DMA request signal is asserted when 8 or more data entries are present in the receive FIFO ● 1001 : DMA request signal is asserted when 9 or more data entries are present in the receive FIFO ● 1010 : DMA request signal is asserted when 10 or more data entries are present in the receive FIFO ● 1011 : DMA request signal is asserted when 11 or more data entries are present in the receive FIFO ● 1100 : DMA request signal is asserted when 12 or more data entries are present in the receive FIFO ● 1101 : DMA request signal is asserted when 13 or more data entries are present in the receive FIFO ● 1110 : DMA request signal is asserted when 14 or more data entries are present in the receive FIFO ● 1111 : DMA request signal is asserted when 15 or more data entries are present in the receive FIFO	

Register 26-21: Data Register (DR)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:0]	DR	RW	**Data Register.** When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer	0

27 MSSI

27.1 Overview

MSSI is a full-duplex master serial interface. CPU processor accesses data, control, and status information on the MSSI through the APB interface. The MSSI can connect to any serial-slave peripheral device using one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI)
- Texas Instruments Serial Protocol (SSP)
- National Semiconductor Microwire

Features

- Serial-master operation – Enables serial communication with serial-master peripheral devices.
- DMA Controller Interface – MSSI interface to a DMA controller over the system bus using a handshaking interface for transfer requests.
- Independent masking of interrupts – Master collision, transmit FIFO overflow, transmit FIFO empty, receive FIFO full, receive FIFO underflow, and receive FIFO overflow interrupts can all be masked independently.
- Multi-master contention detection – Informs the processor of multiple serial-master accesses on the serial bus.
- Clock bit-rate – Dynamic control of the serial bit rate of the data transfer.

Block Diagram

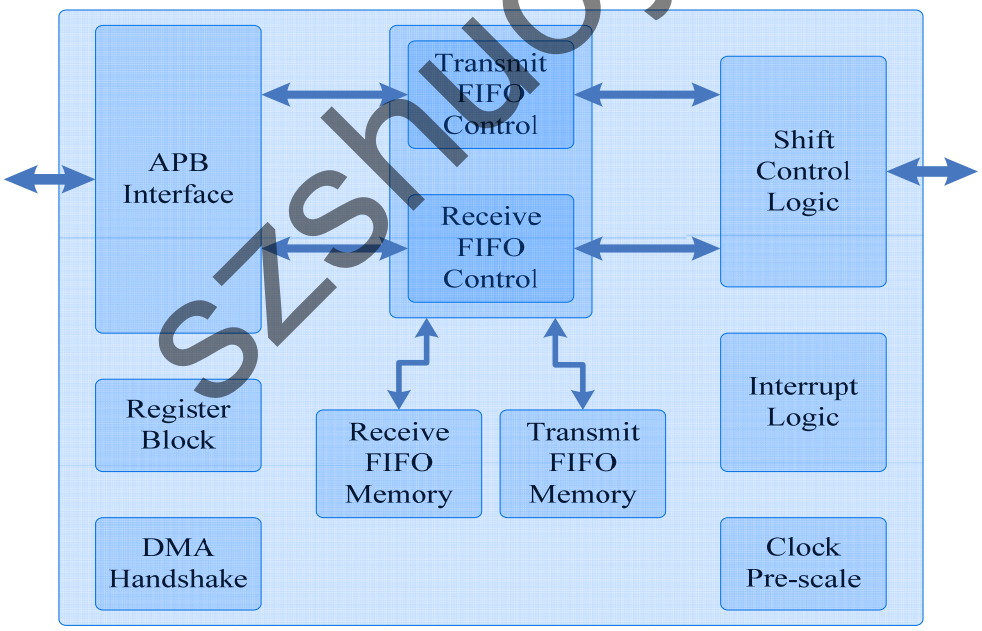


Figure 27-1: MSSI Block Diagram

Figure 27-1 shows the functional block diagram of MSSI, The interface connected with system BUS is APB BUS, with Transmit FIFO and Receive FIFO controlled, MSSI can transfer data from/to system and external peripheral device. MSSI has single DMA hard handshake interface and can communicate with DMA directly.

27.2 Functional Description

In order for the MSSI to connect to a serial-slave peripheral device, the peripheral must have at least one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI) - A four-wire, full-duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of the slave select signal or the first edge of the serial clock. The slave select line is held high when the MSSI is idle or disabled.
- Texas Instruments Serial Protocol (SSP) - A four-wire, full-duplex serial protocol. The slave select line used for SPI and Microwire protocols doubles as the frame indicator for the SSP protocol.
- National Semiconductor Microwire – A half-duplex serial protocol, which uses a control word transmitted from the serial master to the target serial slave.

User can program the FRF (frame format) bit field in the Control Register 0 (CTRLR0) to select which protocol is used.

The serial protocols supported by the MSSI allow for serial slaves to be selected or addressed using either hardware or software. When implemented in hardware, serial slaves are selected under the control of dedicated hardware select lines. The number of select lines generated from the serial master is equal to the number of serial slaves present on the bus. The serial-master device asserts the select line of the target serial slave before data transfer begins.

When implemented in software, the input select line of each serial slave can originate either from a single-slave select output signal on the serial master (user must configure the master to have one slave select output) or be permanently grounded. The main program in the software domain controls selection of the target slave device; Software would use the SSIENR register in all slaves in order to control which slave is to respond to the serial transfer request from the master device.

Clock Ratios

MSSI's maximum frequency of the bit-rate clock (sclk_out) is one-half the frequency of PCLK. This allows the shift control logic to capture data on one clock edge of sclk_out and propagate data on the opposite edge. The sclk_out line toggles only when an active transfer is in progress. At all other times it is held in an inactive state, as defined by the serial protocol under which it operates.

The frequency of sclk_out can be derived from the following equation:

Fsclk_out = PCLK/SCKDV

SCKDV is a bit field in the programmable register BAUDR, holding any even value in the range 0 to 65,534. If SCKDV is 0, then sclk_out is disabled.

27.2.1 Transfer Modes

The MSSI operates in the following four modes when transferring data on the serial bus:

- "Transmit and Receive"
- "Transmit Only"
- "Receive Only"
- "EEPROM Read"

The transfer mode (TMOD) is set by writing to control register 0 (CTRLR0), as described in "CTRLR0".

Transmit and Receive

When TMOD = 2‘b00, both transmit and receive logic are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.

Transmit Only

When TMOD = 2‘b01, the receive data are invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

Receive Only

When TMOD = 2‘b10, the transmit data are invalid. Data from a previous transfer is retransmitted from the shift register. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.

EEPROM Read

When TMOD = 2‘b11, the transmit data is used to transmit an opecode and/or an address to the EEPROM device. Typically this takes three data frames (8-bit opecode followed by 8-bit upper address and 8-bit lower address). During the transmission of the opecode and address, no data is captured by the receive logic (as long as the MSSI master is transmitting data on its txd line, data on the rxd line is ignored). The MSSI master continues to transmit data until the transmit FIFO is empty. Therefore, you should ONLY have enough data frames in the transmit FIFO to supply the opecode and address to the EEPROM. If more data frames are in the transmit FIFO than are needed, then read data is lost. When the transmit FIFO becomes empty (all control information has been sent), data on the receive line (rxd) is valid and is stored in the receive FIFO. The serial transfer continues until the number of data frames received by the MSSI master matches the value of the NDF field in the CTRLR1 register + 1.

27.2.2 Operation Mode

This mode enables serial communication with serial-slave peripheral devices. The MSSI initiates and controls all serial transfers.

The serial bit-rate clock, generated and controlled by the MSSI, is driven out on the sclk_out line. When the MSSI is disabled (MSSI_EN = 0), no serial transfers can occur and sclk_out is held in “inactive” state, as defined by the serial protocol under which it operates.

The MSSI has a serial slave select input, ss_in_n that can be used to inform the MSSI that another serial master is active on the bus. When this input is active—the active level depends on the serial protocol—the MSSI remains in an IDLE state and holds off any pending serial transfer until the ss_in_n input is returned to an in-active level.

Data transfers are started by the serial-master device. When the MSSI is enabled (MSSI_EN=1), at least one valid data entry is present in the transmit FIFO and a serial-slave device is selected. When actively transferring data, the busy flag (BUSY) in the status register (SR) is set. You must wait until the busy flag is cleared before attempting a new serial transfer.

Master SPI and SSP Serial Transfers

When the transfer mode is “transmit and receive” or “transmit only” (TMOD = 2'b00 or TMOD = 2'b01, respectively), transfers are terminated by the shift control logic when the transmit FIFO is empty. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data have been transmitted. The transmit FIFO threshold level (TXFTLR) can be used to early interrupt (ssi_txe_intr) the processor indicating that the transmit FIFO buffer is nearly empty.

When a DMA is used for APB accesses, the transmit data level (DMATDLR) can be used to early request the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. The user may also write a block of data (at least two FIFO entries) into the transmit FIFO before enabling a serial slave. This ensures that serial transmission does not begin until the number of data-frames that make up the continuous transfer are present in the transmit FIFO.

When the transfer mode is “receive only” (TMOD = 2'b10), a serial transfer is started by writing one “dummy” data word into the transmit FIFO when a serial slave is selected. If the serial transfer is continuous, this same data word is retransmitted until the serial transfer is completed. The transmit FIFO is popped only once at the beginning and may remain empty for the duration of the serial transfer.

The end of the serial transfer is controlled by the “number of data frames” (NDF) field in control register 1 (CTRLR1).

If, for example, you want to receive 24 data frames from a serial-slave peripheral, you should program the NDF field with the value 23; the receive logic terminates the serial transfer when the number of frames received is equal to the NDF value + 1. This transfer mode increases the bandwidth of the APB bus as the transmit FIFO never needs to be serviced during the transfer. The receive FIFO buffer should be read each time receive FIFO generates a FIFO full interrupt request to prevent an overflow.

When the transfer mode is “EEPROM_READ” (TMOD = 2'b11), a serial transfer is started by writing the opecode and/or address into the transmit FIFO when a serial slave (EEPROM) is selected. The opecode and address are transmitted to the EEPROM device, after which read data is received from the EEPROM device and stored in the receive FIFO. The end of the serial transfer is controlled by the NDF field in the control register 1 (CTRLR1). The receive FIFO threshold level (RXFTLR) can be used to give early indication that the receive FIFO is nearly full. When a DMA is used for APB accesses, the receive data level (DMARDLR) can be used to early request the DMA Controller, indicating that the receive FIFO is nearly full.

A typical software flow for completing an SPI or SSP serial transfer from the MSSI is outlined as follows:

1. If the MSSI is enabled, disable it by writing 0 to the MSSI Enable register (SSIENR).
2. Set up the MSSI control registers for the transfer; these registers can be set in any order.

- Write Control Register 0 (CTRLR0). For SPI transfers, the serial clock polarity and serial clock phase parameters must be set identical to target slave device.
- If the transfer mode is Receive only, write CTRLR1 (Control Register 1) with the number of frames in the transfer minus 1; for example, if you want to receive four data frames, write this register with 3.
- Write the Baud Rate Select Register (BAUDR) to set the baud rate for the transfer.
- Write the Transmit and Receive FIFO Threshold Level registers (TXFTLR and RXFTLR, respectively) to set FIFO threshold levels.
- Write the IMR register to set up interrupt masks.
- The Slave Enable Register (SER) register can be written here to enable the target slave for selection. If a slave is enabled here, the transfer begins as soon as one valid data entry is present in the transmit FIFO. If no slaves are enabled prior to writing to the Data Register (DR), the transfer does not begin until a slave is enabled.

3. Enable the MSSI by writing 1 to the SSIENR register.

4. Write data for transmission to the target slave into the transmit FIFO (write DR).

   If no slaves were enabled in the SER register at this point, enable it now to begin the transfer.

5. Poll the BUSY status to wait for completion of the transfer. The BUSY status cannot be polled immediately

   If a transmit FIFO empty interrupt request is made, write the transmit FIFO (write DR). If a receive FIFO full interrupt request is made, read the receive FIFO (read DR).

6. The transfer is stopped by the shift control logic when the transmit FIFO is empty. If the transfer mode is receive only (TMOD = 2'b10), the transfer is stopped by the shift control logic when the specified number of frames have been received. When the transfer is done, the BUSY status is reset to 0.

7. If the transfer mode is not transmit only (TMOD $\neq$ 01), read the receive FIFO until it is empty.

8. Disable the MSSI by writing 0 to SSIENR.

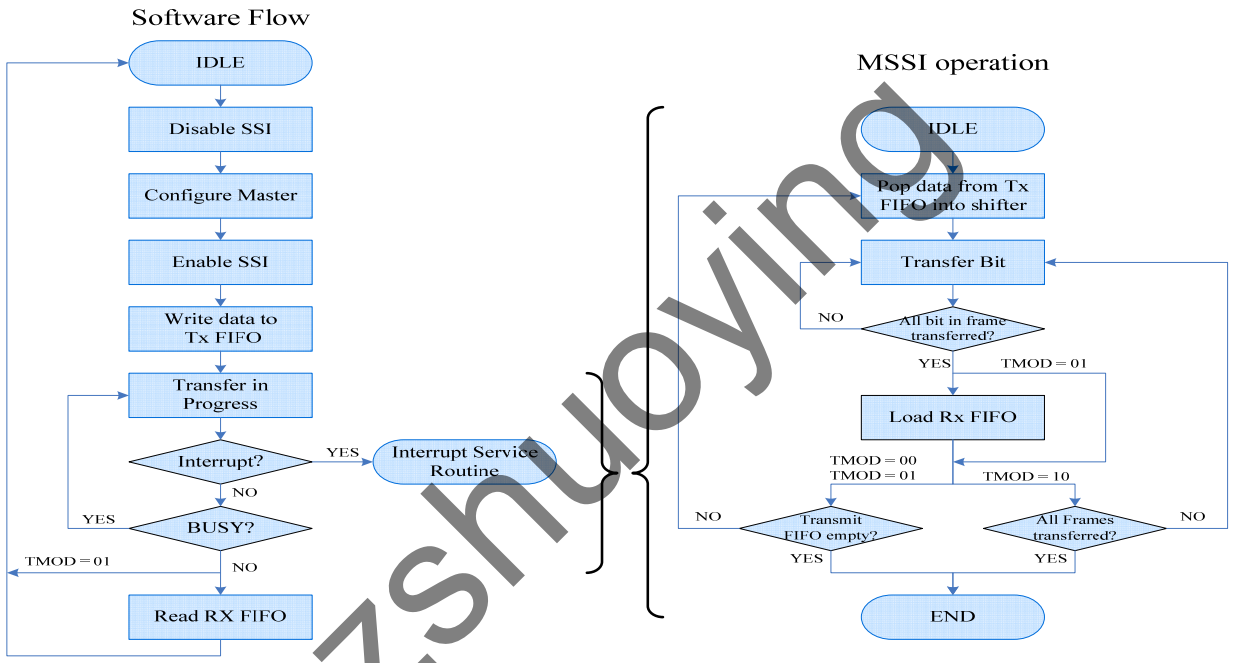


Figure 27-2: MSSI Master SPI/SSP Transfer Flow

Master Microwire Serial Transfers

Microwire serial transfers from the MSSI are controlled by the Microwire Control Register (MWCR). The MWHS bit field enables and disables the Microwire handshaking interface.

The MDD bit field controls the direction of the data frame (the control frame is always transmitted by the master and received by the slave). The MWMOD bit field defines whether the transfer is sequential or non-sequential.

All Microwire transfers are started by the MSSI when there is at least one control word in the transmit FIFO and a slave is enabled. When the MSSI transmits the data frame (MDD = 1), the transfer is terminated by the shift logic when the transmit FIFO is empty. When the MSSI receives the data frame (MDD = 1), the termination of the transfer depends on the setting of the MWMOD bit field. If the transfer is non-sequential (MWMOD = 0), it is terminated when the transmit FIFO is empty after shifting in the data frame from the slave. When the transfer is sequential (MWMOD = 1), it is terminated by the shift logic when the number of data frames received is equal to the value in the CTRLR1 register + 1.

When the handshaking interface on the MSSI is enabled (MWHS =1), the status of the target slave is polled after transmission. Only when the slave reports a ready status does the MSSI complete the transfer and clear its BUSY status. If the transfer is continuous, the next control/data frame is not sent until the slave device returns a ready status.

A typical software flow for completing a Microwire serial transfer from the MSSI is outlined as follows:

1. If the MSSI is enabled, disable it by writing 0 to SSIENR.
2. Set up the MSSI control registers for the transfer. These registers can be set in any order. Write CTRLR0 to set transfer parameters.
   - If the transfer is sequential and the MSSI receives data, write CTRLR1 with the number of frames in the transfer minus 1; for instance, if you want to receive four data frames, write this register with 3.
   - Write BAUDR to set the baud rate for the transfer.
   - Write TXFTLR and RXFTLR to set FIFO threshold levels.
   - Write the IMR register to set up interrupt masks.

   You can write the SER register to enable the target slave for selection. If a slave is enabled here, the transfer begins as soon as one valid data entry is present in the transmit FIFO. If no slaves are enabled prior to writing to the DR register, the transfer does not begin until a slave is enabled.
3. Enable the MSSI by writing 1 to the SSIENR register.
4. If the MSSI transmits data, write the control and data words into the transmit FIFO (write DR). If the MSSI receives data, write the control word(s) into the transmit FIFO.

   If no slaves were enabled in the SER register at this point, enable now to begin the transfer.
5. Poll the BUSY status to wait for completion of the transfer. The BUSY status cannot be polled immediately.

   If a transmit FIFO empty interrupt request is made, write the transmit FIFO (write DR). If a receive FIFO full interrupt request is made, read the receive FIFO (read DR).
6. The transfer is stopped by the shift control logic when the transmit FIFO is empty. If the transfer mode is sequential and the MSSI receives data, the transfer is stopped by the shift control logic when the specified number of data frames is received. When the transfer is done, the BUSY status is reset to 0.
7. If the MSSI receives data, read the receive FIFO until it is empty.
8. Disable the MSSI by writing 0 to SSIENR.

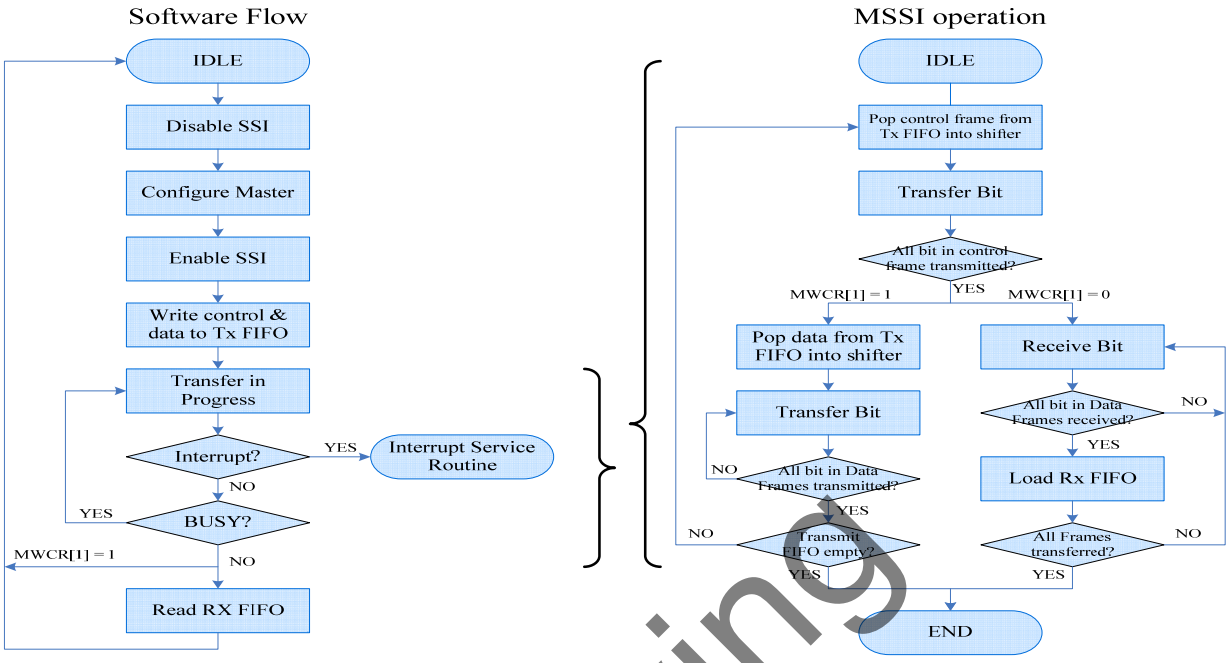


Figure 27-3: MSSI Master Microwire Transfer Flow

27.2.3 DMA Controller Interface

The MSSI has optional built-in DMA capability which can be selected at configuration time; it has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA. While the MSSI DMA operation is designed in a generic way to fit any DMA controller as easily as possible, it is designed to work seamlessly, and best used, with the GDMA. The settings of the GDMA that are relevant to the operation of the MSSI are discussed here, mainly bit fields in the GDMA channel control register, CTLx, where x is the channel number.

The MSSI uses two DMA channels, one for the transmit data and one for the receive data. The MSSI has these DMA registers:

- "DMACR" – Control register to enable DMA operation.
- "DMATDLR"– Register to set transmit the FIFO level at which a DMA request is made.
- "DMARDLR" – Register to set the receive FIFO level at which a DMA request is made.

As a block flow control device, the DMA Controller is programmed by the processor with the number of data items (block size) that are to be transmitted or received by the MSSI; this is programmed into the BLOCK_TS field of the CTLx register.

The block is broken into a number of transactions, each initiated by a request from the MSSI.

The DMA Controller must also be programmed with the number of data items (in this case, MSSI FIFO entries) to be transferred for each DMA request. This is also known as the burst transaction length, and is programmed into the SRC_MSIZE/DEST_MSIZE fields of the GDMA CTLx register for source and destination, respectively.

27.3 MSSI Register Description

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 registers: 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.

27.3.1 MSSI Register Memory Map

There are three MSSI in IMAPX210.

MSSI0 register's system accessing address equal to BASE_ADDRESS (0x20DF0000) plus address offset.
MSSI1 register's system accessing address equal to BASE_ADDRESS (0x20DF1000) plus address offset.
MSSI2 register's system accessing address equal to BASE_ADDRESS (0x20DF2000) plus address offset.

Table 27-1: MSSI Register Memory Map

Address	Symbol	Direction	Description
0x00	CTRLR0	RW	Control Register 0
0x04	CTRLR1	RW	Control Register 1
0x08	MSSIENR	RW	MSSI Enable Register
0x0C	MWCR	RW	Microwire Control Register
0x10	SER	RW	Slave Enable Register Width: MSSI_NUM_SLAVES
0x14	BAUDR	RW	Baud Rate Select
0x18	TXFTLR	RW	Transmit FIFO Threshold Level
0x1C	RXFTLR	RW	Receive FIFO Threshold Level
0x20	TXFLR	RW	Transmit FIFO Level Register Width: TX_ABW+1
0x24	RXFLR	RW	Receive FIFO Level Register Width: RX_ABW+1
0x28	SR	R	Status Register
0x2C	IMR	RW	Interrupt Mask Register
0x30	ISR	R	Interrupt Status Register
0x34	RISR	R	Raw Interrupt Status Register
0x38	TXOICR	R	Transmit FIFO Overflow Interrupt Clear Register
0x3C	RXOICR	R	Receive FIFO Overflow Interrupt Clear Register
0x40	RXUICR	R	Receive FIFO Underflow Interrupt Clear Register
0x44	MSTICR	R	Multi-Master Interrupt Clear Register
0x48	ICR	R	Interrupt Clear Register
0x4C	DMACR	RW	DMA Control Register
0x50	DMATDLR	RW	DMA Transmit Data Level
0x54	DMARDLR	RW	DMA Receive Data Level
0x60-0x9C	DR	RW	Data Register

27.3.2 MSSI Registers and Field Descriptions

Register 27-1: Control Register 0 (CTRLR0)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:12]	CFS	RW	**Control Frame Size.** Selects the length of the control word for the Microwire frame format. ● 0000 : Reserved – Reserved operation ● 0001 : Reserved – Reserved operation ● 0010 : Reserved – Reserved operation ● 0011 : 4-bit serial data transfer ● 0100 : 5-bit serial data transfer ● 0101 : 6-bit serial data transfer ● 0110 : 7-bit serial data transfer ● 0111 : 8-bit serial data transfer ● 1000 : 9-bit serial data transfer ● 1001 : 10-bit serial data transfer ● 1010 : 11-bit serial data transfer ● 1011 : 12-bit serial data transfer ● 1100 : 13-bit serial data transfer ● 1101 : 14-bit serial data transfer ● 1110 : 15-bit serial data transfer ● 1111 : 16-bit serial data transfer	0
[11]	SRL	RW	**Shift Register Loop.** Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. ● 0 : Normal Mode Operation ● 1 : Test Mode Operation	0
[10]	Reserved	N/A	**Reserved**	0
[9:8]	TMOD	RW	**Transfer Mode.** Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. ● 00 : Transmit & Receive ● 01 : Transmit Only ● 10 : Receive Only ● 11 : EEPROM Read	0
[7]	SCPOL	RW	**Serial Clock Polarity.** Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the MSSI is not actively transferring data on the serial bus. ● 0 : Inactive state of serial clock is low ● 1 : Inactive state of serial clock is high	0
[6]	SCPH	RW	**Serial Clock Phase.** Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal.	0

			When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. ● 0 : Serial clock toggles in middle of first data bit ● 1 : Serial clock toggles at start of first data bit	
[5:4]	FRF	RW	**Frame Format.** Selects which serial protocol transfers the data. ● 00 : Motorola SPI ● 01 : Texas Instruments SSP ● 10 : National Semiconductors Microwire ● 11 : Reserved	0
[3:0]	DFS	RW	**Data Frame Size.** Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. ● 0000 : 1-bit control word ● 0001 : 2-bit control word ● 0010 : 3-bit control word ● 0011 : 4-bit control word ● 0100 : 5-bit control word ● 0101 : 6-bit control word ● 0110 : 7-bit control word ● 0111 : 8-bit control word ● 1000 : 9-bit control word ● 1001 : 10-bit control word ● 1010 : 11-bit control word ● 1011 : 12-bit control word ● 1100 : 13-bit control word ● 1101 : 14-bit control word ● 1110 : 15-bit control word ● 1111 : 16-bit control word	7

Register 27-2: Control Register 1 (CTRLR1)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:0]	NDF	RW	**Number of Data Frames.** When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the MSSI. The MSSI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.	

Register 27-3: MSSI Enable Register (MSSIENR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	MSSI_EN	RW	**MSSI Enable.** Enables and disables all MSSI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the MSSI control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system.	0

Register 27-4: Microwire Control Register (MWCR)

Field	Symbol	Direction	Description	Default
[31:3]	Reserved	N/A	Reserved	0
[2]	MHS	RW	**Microwire Handshaking.** Used to enable and disable the "busy/ready" handshaking interface for the Microwire protocol. When enabled, the MSSI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. ● 0: handshaking interface is disabled ● 1: handshaking interface is enabled	0
[1]	MDD	RW	**Microwire Control.** Defines the direction of the data word when the Microwire serial protocol is used. ● When this bit is set to 0, the data word is received by the MSSI Macrocell from the external serial device. ● When this bit is set to 1, the data word is transmitted from the MSSI Macrocell to the external serial device.	0
[0]	MWMOD	RW	**Microwire Transfer Mode.** Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. ● 0 : non-sequential transfer ● 1 : sequential transfer	0

Register 27-5: Slave Enable Register (SER)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	SER	RW	**Slave Select Enable Flag.** Each bit in this register corresponds to a slave select line (ss_x_n) from the MSSI. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set. ● 1: Selected ● 0: Not Selected	

Register 27-6: Baud Rate Select Register (BAUR)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:0]	SCKDV	RW	**MSSI Clock Divider.** The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fssi_clk/SCKDV Where SCKDV is any even value between 2 and 65534.	0

Register 27-7: Transmit FIFO Threshold Level Register (TXFTLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	TFT	RW	**Transmit FIFO Threshold.** Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 16; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. ● 0000 : ssi_txe_intr is asserted when 0 or less data entries are present in transmit FIFO ● 0001 : ssi_txe_intr is asserted when 1 or less data entries are present in transmit FIFO ● 0010 : ssi_txe_intr is asserted when 2 or less data entries are present in transmit FIFO ● 0011 : ssi_txe_intr is asserted when 3 or less data entries are present in transmit FIFO ● 0100 : ssi_txe_intr is asserted when 4 or less data entries are present in transmit FIFO ● 0101 : ssi_txe_intr is asserted when 5 or less data entries are present in transmit FIFO ● 0110 : ssi_txe_intr is asserted when 6 or less data entries are present in transmit FIFO ● 0111 : ssi_txe_intr is asserted when 7 or less data entries are present in transmit FIFO ● 1000 : ssi_txe_intr is asserted when 8 or less data entries are present in transmit FIFO ● 1001 : ssi_txe_intr is asserted when 9 or less data entries are present in transmit FIFO ● 1010 : ssi_txe_intr is asserted when 10 or less data entries are present in transmit FIFO ● 1011 : ssi_txe_intr is asserted when 11 or less data entries are present in transmit FIFO ● 1100 : ssi_txe_intr is asserted when 12 or less data entries are present in transmit FIFO ● 1101 : ssi_txe_intr is asserted when 13 or less data entries are present in transmit FIFO ● 1110 : ssi_txe_intr is asserted when 14 or less data entries are present in transmit FIFO ● 1111 : ssi_txe_intr is asserted when 15 or less data entries are present in transmit FIFO	0

Register 27-8: Receive FIFO Threshold Level Register (RXFTLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	RFT	RW	**Receive FIFO Threshold.** Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is 16. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered. ● 0000 : ssi_txe_intr is asserted when 0 or more data entries are present in receive FIFO ● 0001 : ssi_txe_intr is asserted when 1 or more data entries are present	0

			in receive FIFO ● 0010 : ssi_txe_intr is asserted when 2 or more data entries are present in receive FIFO ● 0011 : ssi_txe_intr is asserted when 3 or more data entries are present in receive FIFO ● 0100 : ssi_txe_intr is asserted when 4 or more data entries are present in receive FIFO ● 0101 : ssi_txe_intr is asserted when 5 or more data entries are present in receive FIFO ● 0110 : ssi_txe_intr is asserted when 6 or more data entries are present in receive FIFO ● 0111 : ssi_txe_intr is asserted when 7 or more data entries are present in receive FIFO ● 1000 : ssi_txe_intr is asserted when 8 or more data entries are present in receive FIFO ● 1001 : ssi_txe_intr is asserted when 9 or more data entries are present in receive FIFO ● 1010 : ssi_txe_intr is asserted when 10 or more data entries are present in receive FIFO ● 1011 : ssi_txe_intr is asserted when 11 or more data entries are present in receive FIFO ● 1100 : ssi_txe_intr is asserted when 12 or more data entries are present in receive FIFO ● 1101 : ssi_txe_intr is asserted when 13 or more data entries are present in receive FIFO ● 1110 : ssi_txe_intr is asserted when 14 or more data entries are present in receive FIFO ● 1111 : ssi_txe_intr is asserted when 15 or more data entries are present in receive FIFO	

Register 27-9: Transmit FIFO Level Register (TXFLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	TXTFL	RW	**Transmit FIFO Level.** Contains the number of valid data entries in the transmit FIFO.	0

Register 27-10: Receive FIFO Level Register (RXFLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	RXTFL	RW	**Receive FIFO Level.** Contains the number of valid data entries in the receive FIFO.	0

Register 27-11: Status Register (SR)

Field	Symbol	Direction	Description	Default
[31:7]	Reserved	N/A	Reserved	0
[6]	DCOL	R	**Data Collision Error.** This bit is set if the MSSI is actively transmitting when another master selects this device as a slave. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. ● 0 : No error ● 1 : Transmit data collision error	0
[5]	Reserved	N/A	Reserved	0
[4]	RFF	R	**Receive FIFO Full.** When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. ● 0 : Receive FIFO is not full ● 1 : Receive FIFO is full	0

[3]	RFNE	R	**Receive FIFO Not Empty.** Set when the Receive FIFO contains one or more entries and is cleared when the Receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. ● 0 : Receive FIFO is empty ● 1 : Receive FIFO is not empty	0
[2]	TFE	R	**Transmit FIFO Empty.** When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. ● 0 : Transmit FIFO is not empty ● 1 : Transmit FIFO is empty	0
[1]	TFNF	R	**Transmit FIFO Not Full.** Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. ● 0 : Transmit FIFO is full ● 1 : Transmit FIFO is not full	0
[0]	BUSY	R	**MSSI Busy Flag.** When set, indicates that a serial transfer is in progress; when cleared indicates that the MSSI is idle or disabled. ● 0 : MSSI is idle or disabled ● 1 : MSSI is actively transferring data	0

Register 27-12: Interrupt Mask Register (IMR)

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	N/A	Reserved	0
[5]	MSTIM	RW	**Multi-Master Contention Interrupt Mask.** ● 0 : ssi_mst_intr interrupt is masked ● 1 : ssi_mst_intr interrupt is not masked	0
[4]	RXFIM	RW	**Receive FIFO Full Interrupt Mask** ● 0 – ssi_rxf_intr interrupt is masked ● 1 – ssi_rxf_intr interrupt is not masked	0
[3]	RXOIM	RW	**Receive FIFO Overflow Interrupt Mask** ● 0 – ssi_rxo_intr interrupt is masked ● 1 – ssi_rxo_intr interrupt is not masked	0
[2]	RXUIM	RW	**Receive FIFO Underflow Interrupt Mask** ● 0 – ssi_rxu_intr interrupt is masked ● 1 – ssi_rxu_intr interrupt is not masked	0
[1]	TXOIM	RW	**Transmit FIFO Overflow Interrupt Mask** ● 0 – ssi_txo_intr interrupt is masked ● 1 – ssi_txo_intr interrupt is not masked	0
[0]	TXEIM	RW	**Transmit FIFO Empty Interrupt Mask** ● 0 – ssi_txe_intr interrupt is masked ● 1 – ssi_txe_intr interrupt is not masked	0

Register 27-13: Interrupt Status Register (ISR)

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	N/A	Reserved	0
[5]	MSTIS	R	**Multi-Master Contention Interrupt Status.** ● 0 = ssi_mst_intr interrupt not active after masking ● 1 = ssi_mst_intr interrupt is active after masking.	0
[4]	RXFIS	R	**Receive FIFO Full Interrupt Status** ● 0 = ssi_rxf_intr interrupt is not active after masking ● 1 = ssi_rxf_intr interrupt is full after masking	0
[3]	RXOIS	R	**Receive FIFO Overflow Interrupt Status** ● 0 = ssi_rxo_intr interrupt is not active after masking ● 1 = ssi_rxo_intr interrupt is active after masking	0
[2]	RXUIS	R	**Receive FIFO Underflow Interrupt Status**	0

			● 0 = ssi_rxu_intr interrupt is not active after masking ● 1 = ssi_rxu_intr interrupt is active after masking	
[1]	TXOIS	R	**Transmit FIFO Overflow Interrupt Status** ● 0 = ssi_txo_intr interrupt is not active after masking ● 1 = ssi_txo_intr interrupt is active after masking	0
[0]	TXEIS	R	**Transmit FIFO Empty Interrupt Status** ● 0 = ssi_txe_intr interrupt is not active after masking ● 1 = ssi_txe_intr interrupt is active after masking	0

Register 27-14: RAW Interrupt Status Register (RISR)

Field	Symbol	Direction	Description	Default
[31:6]	Reserved	N/A	Reserved	0
[5]	MSTIR	R	**Multi-Master Contention Raw Interrupt Status.** ● 0 = ssi_mst_intr interrupt is not active prior to masking ● 1 = ssi_mst_intr interrupt is active prior masking Reset	0
[4]	RXFIR	R	**Receive FIFO Full Raw Interrupt Status** ● 0 = ssi_rxf_intr interrupt is not active prior to masking ● 1 = ssi_rxf_intr interrupt is active prior to masking Reset	0
[3]	RXOIR	R	**Receive FIFO Overflow Raw Interrupt Status** ● 0 = ssi_rxo_intr interrupt is not active prior to masking ● 1 = ssi_rxo_intr interrupt is active prior masking Reset	0
[2]	RXUIR	R	**Receive FIFO Underflow Raw Interrupt Status** ● 0 = ssi_rxu_intr interrupt is not active prior to masking ● 1 = ssi_rxu_intr interrupt is active prior to masking Reset	0
[1]	TXOIR	R	**Transmit FIFO Overflow Raw Interrupt Status** ● 0 = ssi_txo_intr interrupt is not active prior to masking ● 1 = ssi_txo_intr interrupt is active prior masking Reset	0
[0]	TXEIR	R	**Transmit FIFO Empty Raw Interrupt Status** ● 0 = ssi_txe_intr interrupt is not active prior to masking ● 1 = ssi_txe_intr interrupt is active prior masking Reset	0

Register 27-15: Transmit FIFO Overflow Interrupt Clear Register (TXOICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	TXOICR	R	**Clear Transmit FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.	0

Register 27-16: Receive FIFO Overflow Interrupt Clear Register (RXOICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	RXOICR	R	**Clear Receive FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.	0

Register 27-17: Receive FIFO Underflow Interrupt Clear Register (RXUICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	RXUICR	R	**Clear Receive FIFO Underflow Interrupt.** This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.	0

Register 27-18: Multi-Master Interrupt Clear Register (MSTICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	MSTICR	R	**Clear Multi-Master Contention Interrupt.** This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.	0

Register 27-19: Interrupt Clear Register (ICR)

Field	Symbol	Direction	Description	Default
[31:1]	Reserved	N/A	Reserved	0
[0]	ICR	R	**Clear Interrupts**. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.	0

Register 27-20: DMA Control Register (DMACR)

Field	Symbol	Direction	Description	Default
[31:2]	Reserved	N/A	Reserved	0
[1]	TDMAE	RW	**Transmit DMA Enable.** This bit enables/disables the transmit FIFO DMA channel. ● 0 = Transmit DMA disabled ● 1 = Transmit DMA enabled	0
[0]	RDMAE	RW	**Receive DMA Enable.** This bit enables/disables the receive FIFO DMA channel ● 0 = Receive DMA disabled ● 1 = Receive DMA enabled	0

Register 27-21: DMA Transmit Data Level Register (DMATDLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	DMATDL	RW	**Transmit Data Level.** This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the DMA request signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. ● 0000 : DMA request signal is asserted when 0 or less data entries are present in the transmit FIFO ● 0001 : DMA request signal is asserted when 1 or less data entries are present in the transmit FIFO ● 0010 : DMA request signal is asserted when 2 or less data entries are present in the transmit FIFO ● 0011 : DMA request signal is asserted when 3 or less data entries are present in the transmit FIFO ● 0100 : DMA request signal is asserted when 4 or less data entries are present in the transmit FIFO ● 0101 : DMA request signal is asserted when 5 or less data entries are present in the transmit FIFO ● 0110 : DMA request signal is asserted when 6 or less data entries are present in the transmit FIFO ● 0111 : DMA request signal is asserted when 7 or less data entries are present in the transmit FIFO ● 1000 : DMA request signal is asserted when 8 or less data entries are present in the transmit FIFO ● 1001 : DMA request signal is asserted when 9 or less data entries are present in the transmit FIFO	0

			● 1010 : DMA request signal is asserted when 10 or less data entries are present in the transmit FIFO ● 1011 : DMA request signal is asserted when 11 or less data entries are present in the transmit FIFO ● 1100 : DMA request signal is asserted when 12 or less data entries are present in the transmit FIFO ● 1101 : DMA request signal is asserted when 13 or less data entries are present in the transmit FIFO ● 1110 : DMA request signal is asserted when 14 or less data entries are present in the transmit FIFO ● 1111 : DMA request signal is asserted when 15 or less data entries are present in the transmit FIFO	

Register 27-22: DMA Receive Data Level Register (DMARDLR)

Field	Symbol	Direction	Description	Default
[31:4]	Reserved	N/A	Reserved	0
[3:0]	DMARDL	RW	**Receive Data Level.** This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, DMA request signal is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1. ● 0000 : DMA request signal is asserted when 0 or more data entries are present in the receive FIFO ● 0001 : DMA request signal is asserted when 1 or more data entries are present in the receive FIFO ● 0010 : DMA request signal is asserted when 2 or more data entries are present in the receive FIFO ● 0011 : DMA request signal is asserted when 3 or more data entries are present in the receive FIFO ● 0100 : DMA request signal is asserted when 4 or more data entries are present in the receive FIFO ● 0101 : DMA request signal is asserted when 5 or more data entries are present in the receive FIFO ● 0110 : DMA request signal is asserted when 6 or more data entries are present in the receive FIFO ● 0111 : DMA request signal is asserted when 7 or more data entries are present in the receive FIFO ● 1000 : DMA request signal is asserted when 8 or more data entries are present in the receive FIFO ● 1001 : DMA request signal is asserted when 9 or more data entries are present in the receive FIFO ● 1010 : DMA request signal is asserted when 10 or more data entries are present in the receive FIFO ● 1011 : DMA request signal is asserted when 11 or more data entries are present in the receive FIFO ● 1100 : DMA request signal is asserted when 12 or more data entries are present in the receive FIFO ● 1101 : DMA request signal is asserted when 13 or more data entries are present in the receive FIFO ● 1110 : DMA request signal is asserted when 14 or more data entries are present in the receive FIFO ● 1111 : DMA request signal is asserted when 15 or more data entries are present in the receive FIFO	0

Register 27-23: Data Register (DR)

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	N/A	Reserved	0
[15:0]	DR	RW	**Data Register.** When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer	0

28 PIC

28.1 Overview

PS2 Interface Controller (PIC) is a two-wire serial interface, consisting of a serial data line (PS2DAT) and a serial clock (PS2CLK). These wires carry information between host and device. PIC is used as host and connected with external device, such as PS-2 keyboard, mouse and touchpad etc.

PIC has five registers, including PICCR, PICSTAT, PICDATA, PICCLKDV and PICIR. Register PICCLKDV is used to configure PIC internal reference clock. It's important to note that the reference clock should better lower than 8MHz to satisfy PS2 device timing requirement. PICDATA is the transmit/receive data register. It is 8-bits wide. When PICDATA is read, the data values received are accessed. When PICDATA is written to, it is loaded into the transmit register and then serially shifted out onto a pin. The other three registers, PICCR, PICSTAT and PICIR, are used to control PIC operation mode, interrupt and status polling mode.

28.2 PS2 Serial Protocol

The PS/2 device interface, used by many modern mice and keyboards, was developed by IBM and originally appeared in the IBM Technical Reference Manual. The PS/2 mouse and keyboard implement a bidirectional synchronous serial protocol. In other words, Data is sent one bit at a time on the Data line and is read on each time Clock is pulsed. The keyboard/mouse can send data to the host and the host can send data to the device, but the host always has priority over the bus and can inhibit communication from the keyboard/mouse at any time by holding Clock low.

Data sent from the keyboard/mouse to the host is read on the falling edge of the clock signal (when Clock goes from high to low); data sent from the host to the keyboard/mouse is read on the rising edge (when Clock goes from low to high.) Regardless of the direction of communication, the keyboard/mouse always generates the clock signal. If the host wants to send data, it must first tell the device to start generating a clock signal (that process is described in the next section.) The maximum clock frequency is 33 kHz and most devices operate within 10-20kHz.

All data is arranged in bytes with each byte sent in a frame consisting of 11-12 bits. These bits are:

- 1 start bit. This is always 0.
- 8 data bits, LSB first.
- 1 parity bit (odd parity).
- 1 stop bit. This is always 1.
- 1 acknowledge bit (Host-to-device communication only)

  When the host is sending data to the keyboard/mouse, a handshaking bit is sent from the device to acknowledge the packet was received. This bit is not present when the device sends data to the host.

Device-to-Host Communication

The Data and Clock lines are both open drain (normally held at a high logic level). When the keyboard or mouse wants to send information, it first checks Clock to make sure it's at a high logic level. If it's not, the host is inhibiting communication and the device must buffer any to-be-sent data until it regains control of the bus (the keyboard has a 16-byte buffer and the mouse's buffer stores only the last packet sent.) If the Clock line is high, the device can begin to transmit its data. The following figure shows the frame format and each bit is read by the host on the falling edge of the clock.

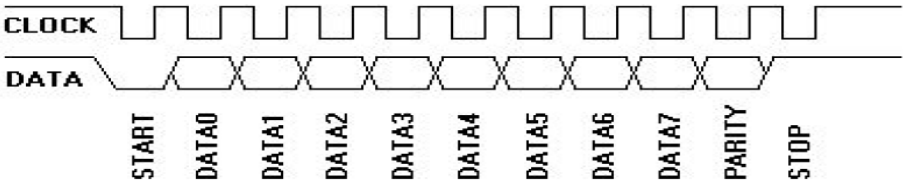


Figure 28-1: Device-to-Host Communication

Host to Device Communication

First of all, the PS/2 device always generates the clock signal. If the host wants to send data, it must first put the Clock and Data lines in a "Request-to-send" state as follows:

1. Inhibit communication by pulling Clock low for at least 100 microseconds.
2. Apply "Request-to-send" by pulling Data low, then release Clock.

The device should check for this state at intervals not to exceed 10 milliseconds. When the device detects this state, it will begin generating Clock signals and clock in eight data bits and one stop bit. The host changes the Data line only when the Clock line is low, and data is latched on the rising edge of the clock pulse. This is opposite of what occurs in device-to-host communication.

After the stop bit is sent, the device will acknowledge the received byte by bringing the Data line low and generating one last clock pulse. If the host does not release the Data line after the 11th clock pulse, the device will continue to generate clock pulses until the Data line is released (the device will then generate an error.)

The Host may abort transmission at time before the 11th clock pulse (acknowledge bit) by holding Clock low for at least 100 microseconds.

To make this process a little easier to understand, here's the steps the host must follow to send data to a PS2 device:

1. Bring the Clock line low for at least 100 microseconds.
2. Bring the Data line low.
3. Release the Clock line.
4. Wait for the device to bring the Clock line low.
5. Set/reset the Data line to send the first data bit
6. Wait for the device to bring Clock high.
7. Wait for the device to bring Clock low.
8. Repeat steps 5-7 for the other seven data bits and the parity bit
9. Release the Data line.
10. Wait for the device to bring Data low.
11. Wait for the device to bring Clock low.
12. Wait for the device to release Data and Clock

Figure 28-2 shows this graphically and Figure 28-3 separates the timing to show which signals are generated by the host, and which are generated by the PS2 device. Notice the change in timing for the Ack bit--the data transition occurs when the Clock line is high (rather than when it is low as is the case for the other 11 bits.)

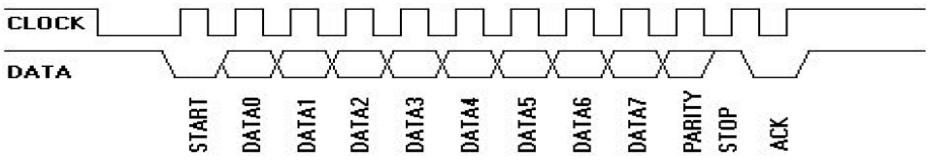


Figure 28-2: Host-to-Device Communication

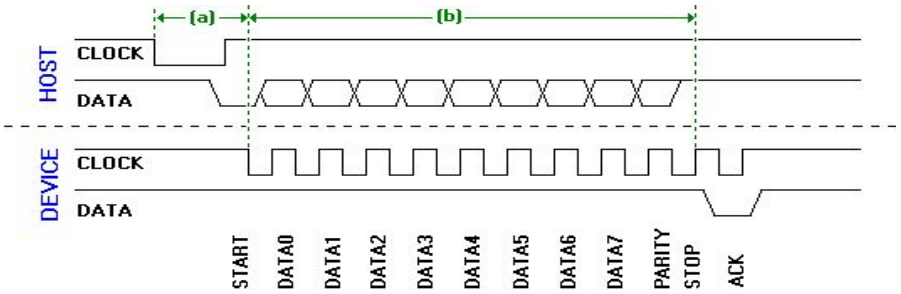


Figure 28-3: Detailed Host-to-Device Communication

28.3 PIC Register Description

28.3.1 PIC Register Memory Map

Table 28-1: PIC Register Memory Map

Address	Symbol	Direction	Description
0x20E7_0000	PICCR	R/W	Control Register
0x20E7_0004	PICSTAT	R	Status register.
0x20E7_0008	PICDATA	R/W	Received data (read) / Data to be transmitted (write).
0x20E7_000C	PICCLKDIV	R/W	Clock divisor register.
0x20E7_0010	PICIR	R	Interrupt status register.

28.3.2 PIC Registers and Field Descriptions

Register 28-1: PIC Control Register （PICCR）

Field	Symbol	Direction	Description	Default
[5]	TYPE	R/W	0 = PS2/AT mode, 1 = No line control bit mode. No line control bit mode is only used by keyboards that do not issue a line control bit to acknowledge data from the system.	0
[4]	RXINTREN	R/W	Enable receiver interrupt. This bit field is used to enable the PIC receiver interrupt. If RXINTREN = 1, the receiver interrupt is enabled.	0
[3]	TXINTREN	R/W	Enable transmitter interrupt. This bit field is used to enable the PIC transmitter interrupt. If TXINTREN = 1, the transfer interrupt is enabled.	0
[2]	PICEN	R/W	The enable PIC bit field is used to enable the PIC. If PICEN = 1, the PIC is enabled.	0
[1]	FPICD	R/W	The force PIC data LOW bit field is used to force the PIC data pad LOW regardless of the state of the PIC finite state machine (FSM). If FPICD = 1, the PS2DAT pad is forced LOW.	0
[0]	FPICC	R/W	The force PIC clock LOW bit field is used to force the PIC clock pad LOW regardless of the state of the PIC FSM. If FPICC = 1, the PS2CLK pad is forced LOW.	0

Register 28-2: PIC Status Register （PICSTAT）

Field	Symbol	Direction	Description	Default
[6]	TXEMPTY	R	This bit indicates that the transmit register is empty and ready to transmit. 0 = Transmit register full. 1 = Transmit register empty, ready to be written.	0
[5]	TXBUSY	R	This bit indicates that the PIC is currently sending data. 0 = Idle. 1 = Currently sending data.	0
[4]	RXFULL	R	This bit indicates that the receiver register is full and ready to be read. 0 = Receive register empty. 1 = Receive register full, ready to be read.	0
[3]	RXBUSY	R	This bit indicates that the PIC is currently receiving data. 0 = Idle. 1 = Currently receiving data.	0
[2]	RXPARITY	R	This bit reflects the parity bit for the last received data byte (odd parity).	0
[1]	PICC	R	This bit reflects the status of the PS2CLK line after synchronizing and sampling.	1
[0]	PICD	R	This bit reflects the status of the PS2DAT line after synchronizing.	1

Register 28-3: PIC Data Register （PICDATA）

Field	Symbol	Direction	Description	Default
[7:0]	PICDATA	R/W	PICDATA is the transmit/receive data register. It is 8-bits wide. When PICDATA is read, the data values received are accessed. When PICDATA is written to, it is loaded into the transmit register and then serially shifted out onto a pin.	0

Register 28-4: PIC Clock Divisor Register （PICCLKDIV）

PICCLKDIV is the clock divisor register which is used to specify the division factor by PCLK should be internally divided before further use. This register can be programmed with the appropriate divisor value in order to have an internal 8MHz signal. The value programmed into this register should be between 0 and 15.

Field	Symbol	Direction	Description	Default
[3:0]	PICCLKDIV	R/W	Clock divisor register. The divide ratio is given as: 1/(1 + PICCLKDIV). The internal nominal 8MHz clock signal will have a frequency F: $F = \frac{F_{PCLK}}{(1 + PICCLKDIV)}$	0

Register 28-5: PIC Interrupt Status Register （PICIR）

Field	Symbol	Direction	Description	Default
[1]	TXINTR	R	This bit is set to 1 if the PIC transmit interrupt is asserted and cleared by write data to PICDATA register.	0
[0]	RXINTR	R	This bit is set to 1 if the PIC receive interrupt is asserted and read data from PICDATA register.	0

29 KeyPad

29.1 KeyPad Overview

The Key Pad Interface block in IMAPx210 facilitates communication with external keypad devices. The ports multiplexed with GPIO ports provide up to 8 rows and 18 columns. The Key Pad can work in two modes, in manual mode, 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. In auto scan mode, the events of key press or key release are detected, hardware will scan the column lines to detect one or multiple key press or release. CPU will be informed by interrupt. 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.

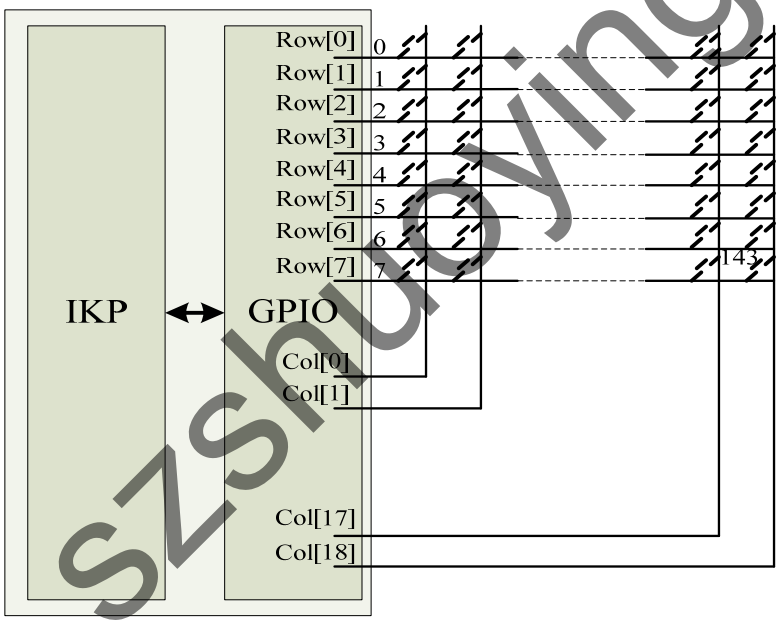


Figure 29-1: Key Matrix Interface External Connection Guide

29.2 KeyPad Function Descriptions

29.2.1 Auto and Manual Scan Mode

The key pad interface can work in two mode, auto scan mode and manual scan mode. In man manual mode, At initial state, all column lines (outputs) are low level. When column data output tri-state enable bits are all high,and the tri-state enable mode is not used, these bits must be written to zeros. 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 shortened 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 KPCOLD register. In each write time, the CPU reads the value of the KPROWD0 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.

In auto scan mode, when any key is pressed or released, the Key Pad Interface will scan each column one by one dependent on value of KPCOEN[24:20] and store column data in KPROWD0 to KPROWD4. When complete scan all column, Key Pad provide a data ready interrupt to CPU. In auto scan mode, the Key Pad Interface can segment one consecutive key pressed into several key pressed by set repeat scan.

29.2.2 Debouncing Filter

The debouncing filter is supported for keypad interrupt of any key input. The filtering width dependent on the value of KBDCNT. The key pressed interrupt to the CPU is an ANDed signal of the all row input lines after filtering. The key released interrupt to the CPU is an ORed signal of the all row input lines after filtering.

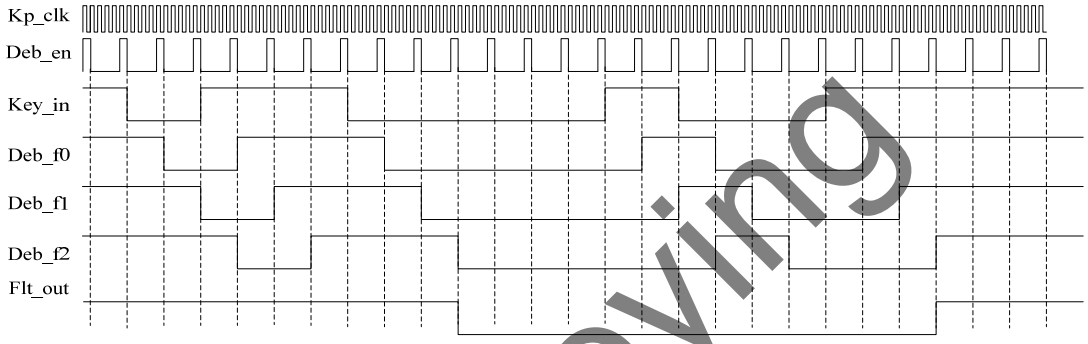


Figure 29-2: Internal Debouncing Filter

29.3 KeyPad Register Description

29.3.1 KeyPad Register Memory Map

Address	Symbol	Direction	Description
0x20E6_0000	KPCON	R/W	KB Control Rregister
0x20E6_0004	KPCKD	R/W	KB Clock Divider Register
0x20E6_0008	KPDCNT	R/W	Debouncing Filter Counter
0x20E6_000C	KPINT	R/(C)	Interrupt States Register
0x20E6_0010	KPCOLD	R/W	Key Column Data Register
0x20E6_0014	KPCOEN	R/W	Column Output Enable Register
0x20E6_0018	KPRPTC	R/W	repeat scan period Register
0x20E6_001c	KPROWD0	R/W	Row Data Register 0
0x20E6_0020	KPROWD1	R/W	Row Data Register 1
0x20E6_0024	KPROWD2	R/W	Row Data Register 2
0x20E6_0028	KPROWD3	R/W	Row Data Register 3
0x20E6_002c	KPROWD4	R/W	Row Data Register 4

29.3.2 KeyPad Individual Register Description

Register 29-1: KeyPad Control Register（KPCON, offset=0x00）

Field	Symbol	Direction	Description	Default
[31:29]	Reserved	R/W	Reserved	0x0
[8]	manual_scan	R/W	control column enable and column value by software	0x0
[7]	rpt_scan_en	R/W	enable repeat scan	0x0
[6]	scan_st	R/W	column scan start ,clear by hardware when data ready.	0x0

			when manual scan , hardware get rowdata accord column.	
[5]	Reserved	R/W	must set '0'.	0x0
[4]	fc_en	R/W	divid clock count enable	0x0
[3]	df_en	R/W	debouncing filter enable	0x0
[2]	drdy_int_en	R/W	scan data ready interrupt enable	0x0
[1]	fint_en	R/W	release key interrupt enable	0x0
[0]	pint_en	R/W	press key interrupt enable	0x0

Register 29-2: KeyPad Clock Divider Register（KPCKD, offset=0x04）

Field	Symbol	Direction	Description	Default
[31:12]	Reserved	R/W	Reserved	0x0
[11:0]	Prescaler	R/W	Determines scan clock(KPCLK) rate as above equation. KPCLK = PCLK / (Prescaler value + 1)	0x00

Register 29-3: KeyPad Debouncing Filter Counter（KPDCNT, offset=0x08）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	R/W	Reserved	0x0
[15:0]	deb_cnt	R/W	debouncing filter counter	0x00

Register 29-4: KeyPad Interrupt States Register（KPINT, offset=0x0C）

Field	Symbol	Direction	Description	Default
[31:3]	Reserved	R/W	Reserved	0x0
[16]	drdy_int	R/W	scan data ready(write '1'to clear flag)	0x0
[15:8]	fint	R/W	release key interrup for each row(write '1'to clear flag)	0x0
[7:0]	pint	R/W	press key interrup for each row(write '1'to clear flag)	0x0

Register 29-5: KeyPad Key Column Data Register（KPCOLD, offset=0x10）

Field	Symbol	Direction	Description	Default
[31:18]	Reserved	R/W	Reserved	0x0
[17:0]	col_line	R/W	column scan enable， '0' scan column	0x00

Register 29-6: KeyPad Column Output Enable Register（KPCOEN, offset=0x14）

Field	Symbol	Direction	Description	Default
[31:25]	Reserved	R/W	Reserved	0x0
[24:20]	col_number	R/W	column number for auto scan. in manual scan mode,set 0.	0x0
[17:0]	col_oen	R/W	column scan output enable， '0' scan enable	0x0

Register 29-7: KeyPad repeat scan period Register（KPRPTC, offset=0x18）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	R/W	Reserved	0x0
[15:0]	rtp_cnt	R/W	repeat scan period counter,clocked by KBCLK	0x00

Register 29-8: KeyPad Row Data Register 0（KPROWD0, offset=0x1C）

Field	Symbol	Direction	Description	Default
[31:24]	row3_line_reg	R/W	row data accord column 3	0x0
[23:16]	row2_line_reg	R/W	row data accord column 2	0x0
[15:8]	row1_line_reg	R/W	row data accord column 1	0x0

[7:0]	row0_line_reg	R/W	row data accord column 0，reflect row port，without debouncing filter in manual mode,only this field valid.	0x0

Register 29-9: KeyPad Row Data Register 1（KPROWD1, offset=0x20）

Field	Symbol	Direction	Description	Default
[31:24]	row7_line_reg	R/W	row data accord column 7	0x0
[23:16]	row6_line_reg	R/W	row data accord column 6	0x0
[15:8]	row5_line_reg	R/W	row data accord column 5	0x0
[7:0]	row4_line_reg	R/W	row data accord column 4，reflect row port，without debouncing filter	0x0

Register 29-10: KeyPad Row Data Register 2（KPROWD2, offset=0x24）

Field	Symbol	Direction	Description	Default
[31:24]	row11_line_reg	R/W	row data accord column 11	0x0
[23:16]	row10_line_reg	R/W	row data accord column 10	0x0
[15:8]	row9_line_reg	R/W	row data accord column 9	0x0
[7:0]	row8_line_reg	R/W	row data accord column 8，reflect row port，without debouncing filter	0x0

Register 29-11: KeyPad Row Data Register 3（KPROWD3, offset=0x28）

Field	Symbol	Direction	Description	Default
[31:24]	row15_line_reg	R/W	row data accord column 15	0x0
[23:16]	row14_line_reg	R/W	row data accord column 14	0x0
[15:8]	row13_line_reg	R/W	row data accord column 13	0x0
[7:0]	row12_line_reg	R/W	row data accord column 12，reflect row port，without debouncing filter	0x0

Register 29-12: KeyPad Row Data Register 4（KPROWD4, offset=0x2C）

Field	Symbol	Direction	Description	Default
[31:16]	Reserved	R/W	Reserved	0x0
[15:8]	row17_line_reg	R/W	row data accord column 13	0x0
[7:0]	row16_line_reg	R/W	row data accord column 12，reflect row port，without debouncing filter	0x0

30 RTC

30.1 RTC Overview

The Real Time Clock (RTC) unit can be operated by the backup battery while the system power is off. The RTC can transmit 8-bit data to CPU as Binary Coded Decimal (BCD) values using the STRB/LDRB ARM operation. The data include the time by second, minute, hour, date, day, month, and year. The RTC unit works with an external 32.768KHz crystal and also can perform the alarm function.

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
- Year 2000 problem is removed.
- Independent power pin (RTCVDD)
- Supports millisecond tick time interrupt for RTOS kernel time tick.

Block Diagram

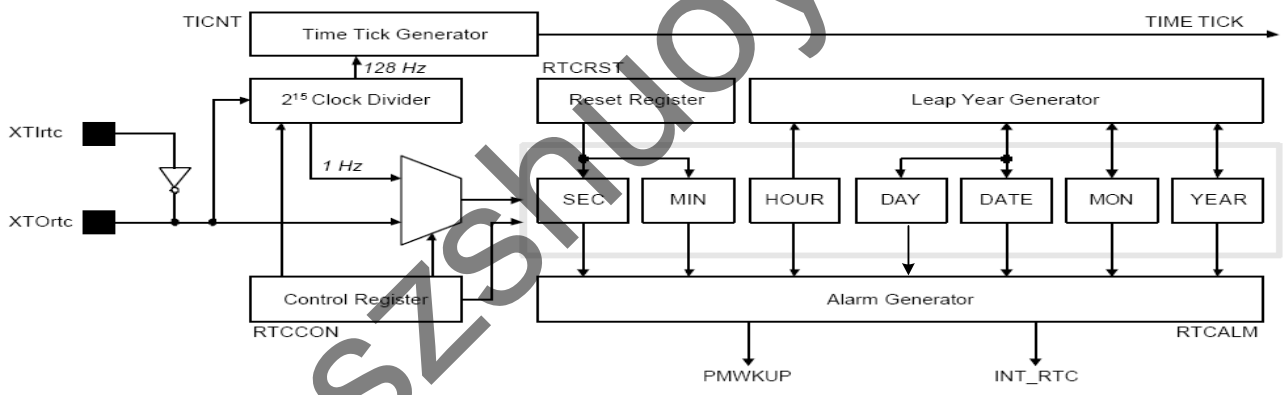


Figure 30-1: RTC Block Diagram

30.2 RTC Function Descriptions

30.2.1 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 BCDDATE, BCDMON, and BCDYEAR. This block considers leap year in deciding on the last date. An 8-bit counter can only represent 2 BCD digits, so 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 IMAPX210 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 IMAPX210 denote 2000, not 1900.

30.2.2 Read/Write Registers

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, date, month, and year, the CPU should read the data in BCDSEC, BCDMIN, BCDHOUR, BCDDAY, BCDDATE, 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 should re-read from BCDYEAR to BCDSEC if BCDSEC is zero.

Note: It's strongly recommend that user set RTC time in the follow order, BCDSEC, BCDMIN, BCDHOUR, BCDDATE, BCDDAY, BCDMON and BCDYEAR, and read RTC time in the reverse order. If the read data of BCDSEC is 00, you should re-read RTC time.

30.2.3 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 should be blocked, and the backup battery only drives the oscillation circuit and the BCD counters to minimize power dissipation.

Note: Before power off, the RTCEN bit should be cleared to 0 to prevent inadvertent writing into RTC registers.

30.2.4 Alarm Function

The RTC generates an alarm signal at a specified time in the power-off mode or normal operation mode. In normal operation mode, the alarm interrupt (INT_RTC) is activated. In the power-off mode, the power management wakeup (PMWKUP) signal is activated as well as the INT_RTC. The RTC alarm register (RTCALM) determines the alarm enable/disable status and the condition of the alarm time setting.

30.2.5 Tick Time Interrupt

The RTC tick time is used for interrupt request. The TICNT register has an interrupt enable bit and the count value for the interrupt. The count value reaches '0' when the tick time interrupt occurs. Then the period of interrupt is as follows:

- Period = (n+1) / 128 second
- n: Tick time count value (1~127)

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.

To avoid any risk of mistaking "Tick Time Interrupt", user should follow these steps:

1. Enable RTCEN (RTCCON[0]) after power on or cold reset.
2. Wait for at least 2 cycle of RTC Clock (32.768 KHz)
3. Set TICK_CNT (TICNT[6:0]).
4. Enable TICK_INTREN (TICNT[7]).

30.3 RTC Registers Descriptions

30.3.1 RTC Register Memory Map

Table 30-1: RTC Register Memory Map

Address	Symbol	Direction	Description
0x20DC_0040	RTCCON	R/W	RTC control register
0x20DC_0044	TICNT	R/W	RTC Tick time count register
0x20DC_0050	RTCALM	R/W	RTC alarm control register
0x20DC_0054	ALMSEC	R/W	RTC Alarm second register
0x20DC_0058	ALMMIN	R/W	RTC Alarm minute register
0x20DC_005C	ALMHOUR	R/W	RTC Alarm hour register
0x20DC_0060	ALMDATE	R/W	RTC alarm date register
0x20DC_0064	ALMMON	R/W	RTC Alarm month register
0x20DC_0068	ALMYEAR	R/W	Alarm year register
0x20DC_0070	BCDSEC	R/W	BCD second register
0x20DC_0074	BCDMIN	R/W	BCD minute register
0x20DC_0078	BCDHOUR	R/W	BCD hour register
0x20DC_007C	BCDDATE	R/W	BCD day register
0x20DC_0080	BCDDAY	R/W	BCD date register
0x20DC_0084	BCDMON	R/W	BCD month register
0x20DC_0088	BCDYEAR	R/W	BCD year register
0x20DC_008C	ALMDAY	R/W	Alarm day register

30.3.2 RTC Registers and Field Descriptions

Register 30-1: RTC Control Register （RTCCON, offset=0x40）

Field	Symbol	Direction	Description	Default
[3]	RCLK_RST	R/W	Real time counter clock disable. 0 = Normal 1 =disable BCD clock	0
[2]	Reserved	R/W	Must be set to 0	0
[1]	CLKSEL	R/W	BCD clock select. 0 = XTAL $1/2^{15}$ divided clock(1Hz) 1 = Reserved (XTAL clock only for test)	0
[0]	RTCEN	R/W	RTC control enable. 0 = Disable 1 = Enable RTCEN bit can control all interfaces between the CPU and the RTC, so it should be set to 1 in an RTC control routine to enable data read/write after a system reset. Also before power off, the RTCEN bit should be cleared to 0 to prevent inadvertent writing into RTC registers. **Note1:** Only BCD time count and read operation can be performed when it is invalid. **Note2**: You should wait for a delay time(2 RTC_CLK clock period, about 61us) when modify RTCEN valued for internal synchronization between PCK and RTC_CLK.	0

Register 30-2: RTC Tick Time Count Register （TICNT, offset=0x44）

Field	Symbol	Direction	Description	Default
[7]	TICK_INTREN	R/W	Tick time interrupt enable. 0 = Disable 1 = Enable	0
[6:0]	TICK_CNT	R/W	Tick time count value (1~127). This counter value decreases internally, and users cannot read this counter value in working.	0

Register 30-3: RTC Alarm Control Register （RTCALM, offset=0x50）

The RTCALM register determines the alarm enable and the alarm time. Please note that the RTCALM register generates the alarm signal through both INT_RTC and PMWKUP in power down mode, but only through INT_RTC in the normal operation mode.

Field	Symbol	Direction	Description	Default
[7]	DAYEN	R/W	Day alarm enable, alarm every week and not be set with YEAREN, MONREN and DATEEN at the same time. 0 = Disable, 1 = Enable	0
[6]	ALMEN	R/W	Alarm global enable. 0 = Disable, 1 = Enable	0
[5]	YEAREN	R/W	Year alarm enable. 0 = Disable, 1 = Enable	
[4]	MONREN	R/W	Month alarm enable. 0 = Disable, 1 = Enable	
[3]	DATEEN	R/W	Date alarm enable. 0 = Disable, 1 = Enable	
[2]	HOUREN	R/W	Hour alarm enable. 0 = Disable, 1 = Enable	
[1]	MINEN	R/W	Minute alarm enable. 0 = Disable, 1 = Enable	
[0]	SECEN	R/W	Second alarm enable. 0 = Disable, 1 = Enable	

Register 30-4: RTC Alarm Second Data Register （ALMSEC, offset=0x54）

Field	Symbol	Direction	Description	Default
[6:4]	ALMSEC	R/W	BCD value for alarm second. 0 ~ 5	0
[3:0]		R/W	0 ~ 9	0

Register 30-5: RTC Alarm Minute Data Register （ALMMIN, offset=0x58）

Field	Symbol	Direction	Description	Default
[6:4]	ALMMIN	R/W	BCD value for alarm minute. 0 ~ 5	0
[3:0]		R/W	0 ~ 9	0

Register 30-6: RTC Alarm Hour Data Register （ALMHOUR, offset=0x5C）

Field	Symbol	Direction	Description	Default
[5:4]	ALMHOUR	R/W	BCD value for alarm hour. 0 ~ 2	0
[3:0]		R/W	0 ~ 9	0

Register 30-7: RTC Alarm Date Data Register （ALMDATE, offset=0x60）

Field	Symbol	Direction	Description	Default
[5:4]	ALMDATE	R/W	BCD value for alarm date, from 0 to 28, 29, 30, 31. 0 ~ 3	0
[3:0]		R/W	0 ~ 9	1

Register 30-8: RTC Alarm Month Data Register （ALMMON, offset=0x64）

Field	Symbol	Direction	Description	Default
[4]	ALMMON	R/W	BCD value for alarm month. 0 ~ 1	0
[3:0]		R/W	0 ~ 9	1

Register 30-9: RTC Alarm Year Data Register （ALMYEAR, offset=0x68）

Field	Symbol	Direction	Description	Default
[7:0]	ALMYEAR	R/W	BCD value for alarm year. 00~ 99	0

Register 30-10: RTC Alarm Day Data Register （ALMDAY, offset=0x8C）

Field	Symbol	Direction	Description	Default
[2:0]	ALMDAY	R/W	BCD value for alarm day. 0~ 6	0

Register 30-11: RTC BCD Second Register （BCDSEC, offset=0x70）

Field	Symbol	Direction	Description	Default
[6:4]	BCDSEC	R/W	BCD value for second. 0 ~ 5	-
[3:0]		R/W	0 ~ 9	-

Register 30-12: RTC BCD Minute Register （BCDMIN, offset=0x74）

Field	Symbol	Direction	Description	Default
[6:4]	BCDMIN	R/W	BCD value for minute. 0 ~ 5	-
[3:0]		R/W	0 ~ 9	-

Register 30-13: RTC BCD Hour Register （BCDHOUR, offset=0x78）

Field	Symbol	Direction	Description	Default
[5:4]	BCDHOUR	R/W	BCD value for hour. 0 ~ 2	-
[3:0]		R/W	0 ~ 9	-

Register 30-14: RTC BCD Date Register （BCDDATE, offset=0x7C）

Field	Symbol	Direction	Description	Default
[5:4]	BCDDATE	R/W	BCD value for date, from 0 to 28, 29, 30, 31. 0 ~ 3	-
[3:0]		R/W	0 ~ 9	-

Register 30-15: RTC BCD Month Register （BCDMON, offset=0x80）

Field	Symbol	Direction	Description	Default
[4]	BCDMON	R/W	BCD value for month. 0 ~ 1	-
[3:0]		R/W	0 ~ 9	-

Register 30-16: RTC BCD Year Register （BCDYEAR, offset=0x84）

Field	Symbol	Direction	Description	Default
[7:0]	BCDYEAR	R/W	BCD value for year. 00~ 99	0

Register 30-17: RTC BCD Day Register （BCDDAY, offset=0x88）

Field	Symbol	Direction	Description	Default
[2:0]	BCDDAY	R/W	BCD value for day. 0~ 6	0

31 WatchDog Timer

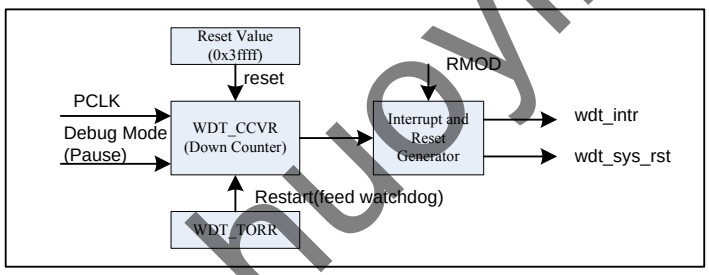
31.1 Overview

The IMAPx210 watchdog timer(WDT) is used to resume the controller operation whenever it is disturbed by malfunctions such as noise and system errors. It can be used as a normal 32-bit interval timer to request interrupt service (interrupt must be cleared each time). The watchdog timer generates the reset signal for 128 PCLK cycles.

Features

- Internal reset signal is activated for 128 PCLK cycles when the timer count value reaches 0 (time-out).
- If a timeout occurs the WDT can perform one of the following operations:
  - Generate a system reset
  - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Debug mode with pause count decrement.

Block Diagram



Register 31-1: Watchdog Timer Block Diagram

31.2 WatchDog Operation Modes

Counter

The WDT counts from a preset (timeout) value in descending order to zero. When the counter reaches zero, depending on the output response mode selected, either a system reset or an interrupt occurs. When the counter reaches zero, it wraps to the selected timeout value and continues decrementing. The user can restart the counter to its initial value. This is programmed by writing to the restart register at any time. The process of restarting the watchdog counter is sometimes referred to as kicking the dog. As a safety feature to prevent accidental restarts, the value 0x76 must be written to the Current Counter Value Register (WDT_CRR).

The initial value of counter after power on reset is 0x3fffff. But you can change it through configure register WDT_TORR.

Interrupts

The WDT can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When a 1 is written to the response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates an interrupt. If it is not cleared by the time a second timeout occurs, then it generates a system reset. If a restart occurs at the same time the watchdog counter reaches zero, an interrupt is not generated.

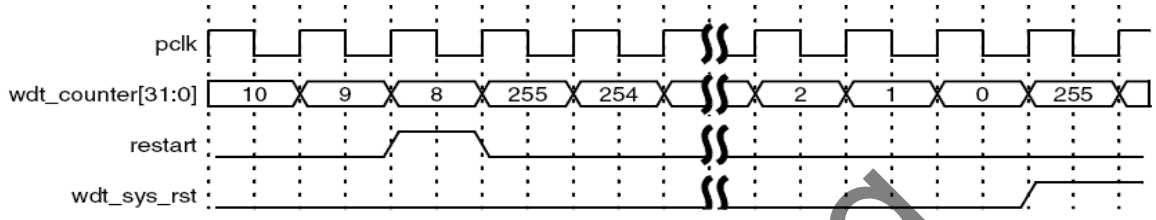
The interrupt is cleared by reading the Watchdog Timer Interrupt Clear register (WDT_EOI) in which no kick is required.

The interrupt can also be cleared by a “kick” (watchdog counter restart).

System Resets

When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs. The WDT can be configured so that it is always enabled upon reset of the WDT. If this is the case, it overrides whatever has been written in bit 0 of the WDT_CR register (the WDT enable field).

The followed figure shows the timing diagram of a counter restart and the generation of a system reset. If a restart occurs at the same time the watchdog counter reaches zero, a system reset is not generated.



Register 31-2: Counter Restart and System Reset

The reset pulse length is 128 PCLK cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length or until the system is reset (by the Reset Controller). A counter restart has no effect on the system reset once it has been asserted.

Debug Mode

When CPU run in emulator mode (Debug Mode), watchdog counter will be freeze and pause counting. During debug mode, if the counter is frozen at the zero count, no interrupt or system reset is generated. When system is run out of debug mode, the interrupt or system reset is asserted on the next rising edge of the clock. If the counter is not zero when exiting debug mode, the interrupt or system reset is not generated.

31.3 WatchDog Register Description

31.3.1 WatchDog Register Memory Map

Table 31-1: WatchDog Register Memory Map

Address	Symbol	Direction	Description
0x20D9_0000	WDT_CR	R/W	Control register
0x20D9_0004	WDT_TORR	R/W	Timeout range register
0x20D9_0008	WDT_CCVR	R	Current counter value register
0x20D9_000C	WDT_CRR	W	Counter restart register
0x20D9_0010	WDT_STAT	R	Interrupt status register
0x20D9_0014	WDT_EOI	R	Interrupt clear register

31.3.2 WatchDog Registers and Field Descriptions

Register 31-3: WDT Control Register （WDT_CR）

Field	Symbol	Direction	Description	Default
[1]	RMOD	R/W	Response mode. Selects the output response generated to a timeout. 0 = Generate a system reset. 1 = First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.	0

[0]	WDT_EN	R/W	WDT enable. This bit is used to enable and disable the WDT. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. 0 = WDT disabled. 1 = WDT enabled.	1

Register 31-4: WDT Timeout Range Register （WDT_TORR）

Field	Symbol	Direction	Description	Default
[3:0]	TOP	R/W	Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: Where i = TOP and t = timeout period For i = 0 to 15 $t = 2^{(16+i)}$	0

Register 31-5: WDT Current Counter Value Register （WDT_CCVR）

Field	Symbol	Direction	Description	Default
[31:0]	CCVR	R	This register, when read, is the current value of the internal counter.	0x3ffff

Register 31-6: WDT Counter Restart Register （WDT_CRR）

Field	Symbol	Direction	Description	Default
[7:0]	CRR	W	This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.	0

Register 31-7: WDT Interrupt Status Register （WDT_STAT）

Field	Symbol	Direction	Description	Default
[0]	ISR	R	This register shows the interrupt status of the WDT. 1 = Interrupt is active 0 = Interrupt is inactive	0

Register 31-8: WDT Interrupt Clear Register （WDT_EOI）

Field	Symbol	Direction	Description	Default
[0]	ICR	R	Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.	0

32 Common Timer

32.1 Overview

The IMAPx210 Common Timer (CMN Timer) has two separately-programmable timers. Each timer has an independent clock input, timer_N_clk, which are configured in system management unit and the frequency must be slower than PCLK.

Timers count down from a programmed value and generate an interrupt when the count reaches zero. The initial value for each timer – that is, the value from which it counts down – is loaded into the timer using the appropriate load count register (TimerNLoadCount). Two events can cause a timer to load the initial count from its TimerNLoadCount register:

- Timer is enabled after being reset or disabled
- Timer counts down to 0

All interrupt status registers and end-of-interrupt registers can be accessed at any time.

Features

- two 32-bit timers
- Support for independent clocking of timers
- Support for two operation modes: free-running and user-defined count

32.2 Common Timer Operation

Timer Usage Flow

The procedure illustrated in followed figure is a basic flow to follow when programming the CMN Timer.

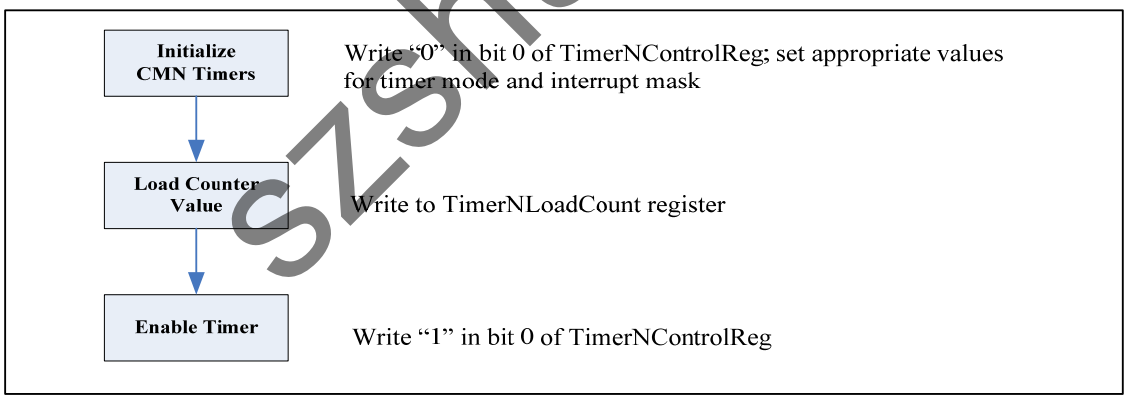


Figure 32-1: Common Timer Usage Flow

1. Initialize the timer through the TimerNControlReg register (where N is in the range 1 to 2):
   a. Disable the timer by writing a "0" to the timer enable bit (bit 0); accordingly, the timer_en output signal is de-asserted.

      Before writing to a TimerNLoadCount register, you must disable the timer by writing a "0" to the timer enable bit of TimerNControlReg in order to avoid potential synchronization problems.

   b. Program the timer mode—user-defined or free-running—by writing a "0" or "1," respectively, to the timer mode bit (bit 1).

You must set the TimerNLoadCount register to all 1s before enabling the timer in free-running mode.

c. Set the interrupt mask as either masked or not masked by writing a "0" or "1," respectively, to the timer interrupt mask bit (bit 2).

2. Load the timer counter value into the TimerNLoadCount register
3. Enable the timer by writing a "1" to bit 0 of TimerNControlReg.

Enabling and Disabling a Timer

You use bit 0 of the TimerNControlReg, where N is in the range 1 to 2, to either enable or disable a timer.

Enabling a Timer

If you want to enable a timer, you write a "1" to bit 0 of its TimerNControlReg register.

Disabling a Timer

To disable a timer, write a "0" to bit 0 of its TimerNControlReg register.

When a timer is enabled and running, its counter decrements on each rising edge of its clock signal, timer_N_clk. When a timer transitions from disabled to enabled, the current value of its TimerNLoadCount register is loaded into the timer counter on the next rising edge of timer_N_clk.

When the timer enable bit is de-asserted and the timer stops running, the timer counter and any associated registers in the timer clock domain, such as the toggle register, are asynchronously reset.

When the timer enable bit is asserted, then a rising edge on the timer_en signal is used to load the initial value into the timer counter. A "0" is always read back when the timer is not enabled; otherwise, the current value of the timer (TimerNCurrentValue register) is read back.

Loading a Timer Countdown Value

The initial value for each timer—that is, the value from which it counts down—is loaded into the timer using the appropriate load count register (TimerNLoadCount). Two events can cause a timer to load the initial count from its TimerNLoadCount register:

- Timer is enabled after being reset or disabled
- Timer counts down to 0

When a timer counts down to 0, it loads one of two values, depending on the timer operating mode:

- User-defined count mode – Timer loads the current value of the TimerNLoadCount register. Use this mode if you want a fixed, timed interrupt. Designate this mode by writing a "1" to bit 1 of TimerNControlReg.
- Free-running mode – Timer loads the maximum value, all bits are loaded with 1s (232-1). The timer counter wrapping to its maximum value allows time to reprogram or disable the timer before another interrupt occurs. Use this mode if you want a single timed interrupt. Designate this mode by writing a "0" to bit 1 of TimerNControlReg.

Set the TimerNLoadCount register to all 1s before enabling the timer in free running mode

32.3 CMN Timer Register Description

32.3.1 CMN Timer Register Memory Map

Table 32-1: CMN Timer Register Memory Map

Address	Symbol	Direction	Description
0x20D3_0000	Timer0loadCount	R/W	CMN Timer 0 Load Count Register
0x20D3_0004	Timer0CurrentValue	R	CMN Timer 0 Current Value Register

0x20D3_0008	Timer0ControlReg	R/W	CMN Timer 0 Control Register
0x20D3_000C	Timer0EOI	R	CMN Timer 0 End-of-Interrupt Register
0x20D3_0010	Timer0IntStatus	R	CMN Timer 0 Interrupt Status Register
0x20D3_0014	Timer1loadCount	R/W	CMN Timer 1 Load Count Register
0x20D3_0018	Timer1CurrentValue	R	CMN Timer 1 Current Value Register
0x20D3_001C	Timer1ControlReg	R/W	CMN Timer 1 Control Register
0x20D3_0020	Timer1EOI	R	CMN Timer 1 End-of-Interrupt Register
0x20D3_0024	Timer1IntStatus	R	CMN Timer 1 Interrupt Status Register

32.3.2 CMN Timer Registers and Field Descriptions

Register 32-1: CMN TimerN Load Count Register （TimerNloadCount, where N is 0 and 1）

Field	Symbol	Direction	Description	Default
[31:0]	TimerN Load Count Register	R/W	Value to be loaded into TimerN. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.	0x0

Register 32-2: CMN TimerN Current Value Register （TimerNCurrentValue, where N is 0 and 1）

Field	Symbol	Direction	Description	Default
[31:0]	TimerN Current Value	R	Current Value of TimerN. **Note:** This register can't exactly reflect the current value of TimerN because timer use independent internal count clock timer_N_clk.	0x0

Register 32-3: CMN TimerN Control Register （TimerNControlReg, where N is 0 and 1）

Field	Symbol	Direction	Description	Default
[2]	Timer Interrupt Mask	R/W	Timer interrupt mask for TimerN. 0: not masked 1: masked	0
[1]	Timer Mode	R/W	Timer mode for TimerN. 0: free-running mode 1: user-defined count mode NOTE: You must set the TimerNLoadCount register to all 1s before enabling the timer in free-running mode.	0
[0]	Timer Enable	R/W	Timer enable bit for TimerN. 0: disable 1: enable	0

Register 32-4: CMN TimerN End-of-Interrupt Register （TimerNEOI, where N is 0 and 1）

Field	Symbol	Direction	Description	Default
[0]	TimerN Endof-Interrupt Register	R	Reading from this register returns all zeroes (0) and clears the interrupt from TimerN.	0

Register 32-5: CMN TimerN Interrupt Status Register （TimerNIntStatus, where N is 0 and 1）

Field	Symbol	Direction	Description	Default
[0]	TimerN Interrupt Status Register	R	Contains the interrupt status for TimerN.	0

33 GPIO

33.1 Overview

IMAPx210 includes 193 multi-functional input/output port pins. There are 18 ports as listed below:

• GPA : 8 in/out port – UART0/1, IrDA0/1, SPI, Keyboard COL[17:14], EINT1
• GPB : 5 in/out port – UART2/3, IrDA2/3, Camera I/F, Keyboard COL[13:9], EINT2
• GPC : 8 in/out port(open drain) – I2C-0/1,PIC-0/1 , EINT3
• GPD : 5 in/out port – I2S, AC97
• GPE : 16 in/out port – Master SSI0/1,Slave SSI, SRAM nCS/nBE, OTG DrvVbus, Nand CS1, EINT4
• GPF : 10 in/out port – SDIO0, PWM, ClockOut, TVIF Control signals, EINT4
• GPG : 6 in port – EINT, SDIO nCD
• GPH : 4 in/out port – Keyboard COL[17:14], SRAM Addr[17:14], EINT6
• GPI : 14 in/out port –Keyboard COL[13:0], SRAM Addr[13:0], EINT6
• GPJ : 9 in/out port –Keyboard ROW, SRAM Control Signals, Ethernet, EINT6
• GPK : 16 in/out port – Ethernet, Keyboard COL and ROW, EINT5
• GPL : 13 in/out port – Camera I/F, DVB-TS, EINT5/6
• GPM : 16 in/out port – IDS VD[15:0], Keyboard COL, EINT5
• GPN : 13 in/out port – IDS VD[23:16], IDS Control signals, Keyboard COL, EINT5
• GPO : 16 in/out port – SDIO1/2, Master SSI2, Uart1 Modem, IDE UDMA, USB Host Port3, EINT4
• GPP : 12 in/out port – CF/IDE control signals, SRAM addr
• GPQ : 6 in/out port – Nand control signals
• GPR : 16 in/out port – Flash Data

Features

The GPIO provides the following features:

- Controls 117 External Interrupts
- 193 multi-functional input/output ports

33.2 Register Description

33.2.1 GPIO Register Memory Map

Address	Symbol	Direction	Description
0x20E10000	GPADAT	R/W	Port A Data Register
0x20E10004	GPACON	R/W	Port A Configuration Register
0x20E10008	GPAPUD	R/W	Port A Pull-up/down Register
0x20E10010	GPBDAT	R/W	Port B Data Register
0x20E10014	GPBCON	R/W	Port B Configuration Register
0x20E10018	GPBPUD	R/W	Port B Pull-up/down Register
0x20E10020	GPCDAT	R/W	Port C Data Register
0x20E10024	GPCCON	R/W	Port C Configuration Register
0x20E10028	GPCPUD	R/W	Port C Pull-up/down Register
0x20E10030	GPDDAT	R/W	Port D Data Register

0x20E10034	GPDCON	R/W	Port D Configuration Register
0x20E10038	GPDPUD	R/W	Port D Pull-up/down Register
0x20E10040	GPEDAT	R/W	Port E Data Register
0x20E10044	GPECON	R/W	Port E Configuration Register
0x20E10048	GPEPUD	R/W	Port E Pull-up/down Register
0x20E10050	GPFDAT	R/W	Port F Data Register
0x20E10054	GPFCON	R/W	Port F Configuration Register
0x20E10058	GPFPUD	R/W	Port F Pull-up/down Register
0x20E10060	GPGDAT	R	Port G Data Register
0x20E10070	GPHDAT	R/W	Port H Data Register
0x20E10074	GPHCON	R/W	Port H Configuration Register
0x20E10078	GPHPUD	R/W	Port H Pull-up/down Register
0x20E10080	GPIDAT	R/W	Port I Data Register
0x20E10084	GPICON	R/W	Port I Configuration Register
0x20E10088	GPIPUD	R/W	Port I Pull-up/down Register
0x20E10090	GPJDAT	R/W	Port J Data Register
0x20E10094	GPJCON	R/W	Port J Configuration Register
0x20E10098	GPJPUD	R/W	Port J Pull-up/down Register
0x20E100A0	GPKDAT	R/W	Port K Data Register
0x20E100A4	GPKCON	R/W	Port K Configuration Register
0x20E100A8	GPKPUD	R/W	Port K Pull-up/down Register
0x20E100B0	GPLDAT	R/W	Port L Data Register
0x20E100B4	GPLCON	R/W	Port L Configuration Register
0x20E100B8	GPLPUD	R/W	Port L Pull-up/down Register
0x20E100C0	GPMDAT	R/W	Port M Data Register
0x20E100C4	GPMCON	R/W	Port M Configuration Register
0x20E100C8	GPMPUD	R/W	Port M Pull-up/down Register
0x20E100D0	GPNDAT	R/W	Port N Data Register
0x20E100D4	GPNCON	R/W	Port N Configuration Register
0x20E100D8	GPNPUD	R/W	Port N Pull-up/down Register
0x20E100E0	GPODAT	R/W	Port O Data Register
0x20E100E4	GPOCON	R/W	Port O Configuration Register
0x20E100E8	GPOPUD	R/W	Port O Pull-up/down Register
0x20E100F0	GPPDAT	R/W	Port P Data Register
0x20E100F4	GPPCON	R/W	Port P Configuration Register
0x20E100F8	GPPPUD	R/W	Port P Pull-up/down Register
0x20E10100	GPQDAT	R/W	Port Q Data Register
0x20E10104	GPQCON	R/W	Port Q Configuration Register
0x20E10108	GPQPUD	R/W	Port Q Pull-up/down Register
0x20E10110	GPRDAT	R/W	Port R Data Register
0x20E10114	GPRCON	R/W	Port R Configuration Register
0x20E10118	GPRPUD	R/W	Port R Pull-up/down Register
0x20E10200	EINTCON	R/W	External Interrupt configuration Register
0x20E10204	EINTFLTCON0	R/W	External Interrupt Filter Control Register 0
0x20E10208	EINTFLTCON1	R/W	External Interrupt Filter Control Register 1
0x20E10210	EINTGCON	R/W	External Interrupt Group configuration Register
0x20E10214	EINTGFLTCON0	R/W	External Interrupt Group Filter Control Register 0
0x20E10218	EINTGFLTCON1	R/W	External Interrupt Group Filter Control Register 1
0x20E1021C	EINTG1MASK	R/W	External Interrupt Group 1 Mask register
0x20E10220	EINTG2MASK	R/W	External Interrupt Group 2 Mask register
0x20E10224	EINTG3MASK	R/W	External Interrupt Group 3 Mask register
0x20E10228	EINTG4MASK	R/W	External Interrupt Group 4 Mask register
0x20E1022C	EINTG5MASK	R/W	External Interrupt Group 5 Mask register
0x20E10230	EINTG6MASK	R/W	External Interrupt Group 6 Mask register
0x20E10234	EINTG1PEND	R/W	External Interrupt Group 1 Pending register
0x20E10238	EINTG2PEND	R/W	External Interrupt Group 2 Pending register

0x20E1023C	EINTG3PEND	R/W	External Interrupt Group 3 Pending register
0x20E10240	EINTG4PEND	R/W	External Interrupt Group 4 Pending register
0x20E10244	EINTG5PEND	R/W	External Interrupt Group 5 Pending register
0x20E10248	EINTG6PEND	R/W	External Interrupt Group 6 Pending register

33.2.2 GPIO Individual Register Description

Register 33-1: PortA Data Registers(GPADAT, offset=0x00)

Field	Symbol	Direction	Description	Default
[7:0]	GPA	R/W	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 Reserved value will be read.	Pin state

Register 33-2: PortA Configuration Registers(GPACON, offset=0x04)

Field	Symbol	Direction	Description	Default
[1:0]	GPA0	R/W	00=port input / eint group 1[0], 01=output, 10= RS232/IrDA RXD[0], 11= KB COL[14]	00
[3:2]	GPA1	R/W	00=port input / eint group 1[1], 01=output, 10= RS232/IrDA TXD[0], 11= KB COL[15]	00
[5:4]	GPA2	R/W	00=port input / eint group 1[2], 01=output, 10= UART CTS[0], 11= KB COL[16]	00
[7:6]	GPA3	R/W	00=port input / eint group 1[3], 01=output, 10= UART RTS[0], 11= KB COL[17]	00
[9:8]	GPA4	R/W	00=port input / eint group 1[4], 01=output, 10= RS232/IrDA RXD[1], 11= SPI MISO	00
[11:10]	GPA5	R/W	00=port input / eint group 1[5], 01=output, 10= RS232/IrDA TXD[1], 11= SPI CLK	00
[13:12]	GPA6	R/W	00=port input / eint group 1[6], 01=output, 10= UART CTS[1], 11= SPI MOSI	00
[15:14]	GPA7	R/W	00=port input / eint group 1[7], 01=output, 10= UART RTS[1], 11= SPI CS[1]	00

Register 33-3: PortA Pull UP/Down Registers(GPAPUD, offset=0x08)

Field	Symbol	Direction	Description	Default
[0]	GPA0	R/W	1 : pull up enable, 0 : pull up disable	1
[1]	GPA1	R/W	1 : pull up enable, 0 : pull up disable	1
[2]	GPA2	R/W	1 : pull down enable, 0 : pull down disable	1
[3]	GPA3	R/W	1 : pull down enable, 0 : pull down disable	1
[4]	GPA4	R/W	1 : pull up enable, 0 : pull up disable	1
[5]	GPA5	R/W	1 : pull up enable, 0 : pull up disable	1
[6]	GPA6	R/W	1 : pull down enable, 0 : pull down disable	1
[7]	GPA7	R/W	1 : pull down enable, 0 : pull down disable	1

Register 33-4: PortB Data Registers(GPBDAT, offset=0x10)

Field	Symbol	Direction	Description	Default
[4:0]	GPB	R/W	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 Reserved value will be read.	Pin state

Register 33-5: PortB Configuration Registers(GPBCON, offset=0x14)

Field	Symbol	Direction	Description	Default
[1:0]	GPB0	R/W	00=port input / eint group 2[0], 01=output, 10= RS232/IrDA RXD[2], 11= KB COL[10]	00
[3:2]	GPB1	R/W	00=port input / eint group 2[1], 01=output, 10= RS232/IrDA TXD[2], 11= KB COL[11]	00
[5:4]	GPB2	R/W	00=port input / eint group 2[2], 01=output, 10= RS232/IrDA RXD[3], 11= KB COL[12]	00
[7:6]	GPB3	R/W	00=port input / eint group 2[3], 01=output, 10= RS232/IrDA TXD[3], 11= KB COL[13]	00
[9:8]	GPB4	R/W	00=port input/ eint group 2[4] /cam_field 01=output 10= Reserved, 11= KB COL[9]	00

Register 33-6: PortB Pull UP/Down Registers(GPBPUD, offset=0x18)

Field	Symbol	Direction	Description	Default
[4:0]	GPB[n]	R/W	1 : pull up enable, 0 : pull up disable	0x1f

Register 33-7: PortC Data Registers(GPCDAT, offset=0x20)

Field	Symbol	Direction	Description	Default
[7:0]	GPC	R/W	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 Reserved value will be read.	pin state

Register 33-8: PortC Configuration Registers(GPCCON, offset=0x24)

Field	Symbol	Direction	Description	Default
[1:0]	GPC0	R/W	00=port input / eint group 3[0], 01=output, 10= IIC SCL[0], 11=Reserved	00
[3:2]	GPC1	R/W	00=port input / eint group 3[1], 01=output, 10= IIC SDA[0], 11=Reserved	00
[5:4]	GPC2	R/W	00=port input / eint group 3[2], 01=output, 10= IIC SCL[1] , 11= Reserved	00
[7:6]	GPC3	R/W	00=port input / eint group 3[3], 01=output, 10= IIC SDA[1], 11= Reserved	00
[9:8]	GPC4	R/W	00=port input / eint group 3[4], 01=output, 10= PIC CLK[0], 11=Reserved	00
[11:10]	GPC5	R/W	00=port input / eint group 3[5], 01=output, 10= PIC DATA[0], 11=Reserved	00
[13:12]	GPC6	R/W	00=port input / eint group 3[6], 01=output, 10= PIC CLK[1], 11= Reserved	00
[15:14]	GPC7	R/W	00=port input / eint group 3[7], 01=output, 10= PIC DATA[1], 11= Reserved	00

Note : GPC pads are open drain pads.

Register 33-9: PortC Pull UP/Down Registers(GPCPUD, offset=0x28)

Field	Symbol	Direction	Description	Default
[7:0]	GPC[n]	R/W	1 : pull up enable, 0 : pull up disable	0xFF

Register 33-10: PortD Data Registers(GPDDAT, offset=0x30)

Field	Symbol	Direction	Description	Default
[4:0]	GPD	R/W	When the port is configured as input port, the corresponding bit is the pin state.	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 Reserved value will be read.	

Register 33-11: PortD Configuration Registers(GPDCON, offset=0x34)

Field	Symbol	Direction	Description	Default
[1:0]	GPD0	R/W	0=port input, 01=output, 10= IIS SCLK, 11= AC97 BCLK	00
[3:2]	GPD1	R/W	00=port input 01=output, 10= IIS CDCLK, 11== AC97 RESET	00
[5:4]	GPD2	R/W	0=port input 01=output, 10= IIS LRCK 11== AC97 SYNC	00
[7:6]	GPD3	R/W	00=port input 01=output, 10= IIS SDI 11== AC97 SDIN	00
[9:8]	GPD4	R/W	0=port input 01=output, 10= IIS SDO, 11=AC97 SDOUT	00

Register 33-12: PortD Pull UP/Down Registers(GPDPUD, offset=0x38)

Field	Symbol	Direction	Description	Default
[4:0]	GPD[n]	R/W	1 : pull up enable, 0 : pull up disable	0x1f

Register 33-13: PortE Data Registers(GPEDAT, offset=0x40)

Field	Symbol	Direction	Description	Default
[15:0]	GPE	R/W	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 Reserved value will be read.	pin state

Register 33-14: PortE Configuration Registers(GPECON, offset=0x44)

Field	Symbol	Direction	Description	Default
[1:0]	GPE0	R/W	00=port input/eint group 4[0] 01=output, 10= Master SSI1 RXD 11= Slave SSI TXD	00
[3:2]	GPE1	R/W	00=port input/eint group 4[1] 01=output, 10= Master SSI1 CLK 11= Slave SSI CLK	00
[5:4]	GPE2	R/W	00=port input/eint group 4[2] 01=output, 10= Master SSI1 TXD 11= Slave SSI RXD	00
[7:6]	GPE3	R/W	00=port input/eint group 4[3] 01=output, 10= Master SSI1 CSN0 11= Slave SSI CSN	00
[9:8]	GPE4	R/W	00=port input/eint group 4[4] 01=output, 10= Master SSI0 TXD, 11= Reserved	00
[11:10]	GPE5	R/W	00=port input/eint group 4[5] 01=output, 10= Master SSI0 RXD, 11= Reserved	00
[13:12]	GPE6	R/W	00=port input/eint group 4[6] 01=output, 10= Master SSI0 CLK, 11= SRAM CSN3	00
[15:14]	GPE7	R/W	00=port input/eint group 4[7] 01=output, 10= Master SSI0 CSN[0], 11= SRAM CSN2	00
[17:16]	GPE8	R/W	00=port input/eint group 4[8] 01=output, 10= Master SSI0 CSN[1], 11= SRAM BEN0	00
[19:18]	GPE9	R/W	00=port input/eint group 4[9] 01=output, 10= Master SSI0 CSN[2], 11= SRAM BEN1	00
[21:20]	GPE10	R/W	00=port input/eint group 4[10] 01=output, 10= Master SSI0 CSN[3], 11= SRAM CSN1	00
[23:22]	GPE11	R/W	00=port input/eint group 4[11] 01=output,	00

			10= Reserved, 11= Reserved	
[25:24]	GPE12	R/W	00=port input/eint group 4[12] 01=output, 10= Reserved, 11= Reserved	00
[27:26]	GPE13	R/W	00=port input/eint group 4[13] 01=output, 10= Reserved, 11= Reserved	00
[29:28]	GPE14	R/W	00=port input/eint group 4[14] 01=output, 10= NAND CSN1, 11= Reserved	00
[31:30]	GPE15	R/W	00=port input/eint group 4[15] 01=output, 10= OTG DrvVBUS, 11= Reserved	00

Register 33-15: PortF Data Registers(GPFDAT, offset=0x50)

Field	Symbol	Direction	Description	Default
[9:0]	GPF	R/W	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 Reserved value will be read.	pin state

Register 33-16: PortF Configuration Registers(GPFCON, offset=0x54)

Field	Symbol	Direction	Description	Default
[1:0]	GPF0	R/W	00=port input 01=output, 10= SDIO0 CLK 11=Reserved	00
[3:2]	GPF1	R/W	00=port input 01=output, 10= SDIO0 CMD 11=Reserved	00
[5:4]	GPF2	R/W	00=port input/eint group4[16] 01=output, 10= SDIO0 DAT[0] 11=Reserved	00
[7:6]	GPF3	R/W	00=port input/eint group4[17] 01=output, 10= SDIO0 DAT[1] 11=Reserved	00
[9:8]	GPF4	R/W	00=port input/eint group4[18] 01=output, 10= SDIO0 DAT[2] 11=Reserved	00
[11:10]	GPF5	R/W	00=port input/eint group4[19] 01=output, 10= SDIO0 DAT[3] 11=Reserved	00
[13:12]	GPF6	R/W	00=port input/eint group4[20] , 01=output, 10= TOUT[0], 11= clkout0	00
[15:14]	GPF7	R/W	00=port input/eint group4[21] , 01=output, 10= TOUT[1], 11= clkout1	00
[17:16]	GPF8	R/W	00=port input/eint group4[22] , 01=output, 10= TOUT[2], 11=tv_field	00
[19:18]	GPF9	R/W	00=port input/eint group4[23], 01=output, 10= TOUT[3], 11= tvif_clk_i	00

Register 33-17: PortF Pull UP/Down Registers(GPFPUD, offset=0x58)

Field	Symbol	Direction	Description	Default
[9:0]	GPF[n]	R/W	1 : pull up enable, 0 : pull up disable	0x3ff

Register 33-18: PortG Data Registers(GPGDAT, offset=0x60)

Field	Symbol	Direction	Description	Default
[2:0]	GPG	R	Only input port, corresponding bit is the pin state.	pin state
[3]	GPG	R	Only input port, corresponding bit is the pin state. This pad used for SD/SDHC/SDIO0 nCD.	pin state
[4]	GPG	R	Only input port, corresponding bit is the pin state. This pad used for SD/SDHC/SDIO1 nCD.	pin state
[5]	GPG	R	Only input port, corresponding bit is the pin state. This pad used for SD/SDHC/SDIO2 nCD.	pin state

Register 33-19: PortH Data Registers(GPHDAT, offset=0x70)

Field	Symbol	Direction	Description	Default
[3:0]	GPH	R/W	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 Reserved value will be read.	pin state

Register 33-20: PortH Configuration Registers(GPHCON, offset=0x74)

Field	Symbol	Direction	Description	Default
[1:0]	GPH0	R/W	00=port input/eint group6[0], 01=output, 10= SRAM ADDR[14] 11= Keyboard COL[14]	10
[3:2]	GPH1	R/W	00=port input/eint group6[1], 01=output, 10= SRAM ADDR[15] 11= Keyboard COL[15]	10
[5:4]	GPH2	R/W	00=port input/eint group6[2], 01=output, 10= SRAM ADDR[16] 11= Keyboard COL[16]	10
[7:6]	GPH3	R/W	00=port input/eint group6[3], 01=output, 10= SRAM ADDR[17] 11= Keyboard COL[17]	10

Register 33-21: PortH Pull UP/Down Registers(GPHPUD, offset=0x78)

Field	Symbol	Direction	Description	Default
[3:0]	GPH[n]	R/W	1 : pull up enable, 0 : pull up disable	0xf

Register 33-22: PortI Data Registers(GPIDAT, offset=0x80)

Field	Symbol	Direction	Description	Default
[13:0]	GPI	R/W	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 Reserved value will be read.	pin state

Register 33-23: PortI Configuration Registers(GPICON, offset=0x84)

Field	Symbol	Direction	Description	Default
[1:0]	GPI0	R/W	00=port input/eint group6[4], 01=output, 10= SRAM ADDR[0] 11= Keyboard COL[0]	10
[3:2]	GPI1	R/W	00=port input/eint group6[5], 01=output, 10= SRAM ADDR[1] 11= Keyboard COL[1]	10
[5:4]	GPI2	R/W	00=port input/eint group6[6], 01=output, 10= SRAM ADDR[2] 11= Keyboard COL[2]	10
[7:6]	GPI3	R/W	00=port input/eint group6[7], 01=output, 10= SRAM ADDR[3] 11= Keyboard COL[3]	10
[9:8]	GPI4	R/W	00=port input/eint group6[8], 01=output, 10= SRAM ADDR[4] 11= Keyboard COL[4]	10
[11:10]	GPI5	R/W	00=port input/eint group6[9], 01=output, 10= SRAM ADDR[5] 11= Keyboard COL[5]	10
[13:12]	GPI6	R/W	00=port input/eint group6[10], 01=output, 10= SRAM ADDR[6] 11= Keyboard COL[6]	10
[15:14]	GPI7	R/W	00=port input/eint group6[11], 01=output, 10= SRAM ADDR[7] 11= Keyboard COL[7]	10
[17:16]	GPI8	R/W	00=port input/eint group6[12], 01=output, 10= SRAM ADDR[8] 11= Keyboard COL[8]	10
[19:18]	GPI9	R/W	00=port input/eint group6[13], 01=output, 10= SRAM ADDR[9] 11= Keyboard COL[9]	10
[21:20]	GPI10	R/W	00=port input/eint group6[14], 01=output,	10

			10= SRAM ADDR[10] 11= Keyboard COL[10]	
[23:22]	GPI11	R/W	00=port input/eint group6[15], 01=output, 10= SRAM ADDR[11] 11= Keyboard COL[11]	10
[25:24]	GPI12	R/W	00=port input/eint group6[16], 01=output, 10= SRAM ADDR[12] 11= Keyboard COL[12]	10
[27:26]	GPI13	R/W	00=port input/eint group6[17], 01=output, 10= SRAM ADDR[13] 11= Keyboard COL[13]	10

Register 33-24: PortI Pull UP/Down Registers(GPIPUD, offset=0x88)

Field	Symbol	Direction	Description	Default
[13:0]	GPI[n]	R/W	1 : pull up enable, 0 : pull up disable	0x3fff

Register 33-25: PortJ Data Registers(GPJDAT, offset=0x90)

Field	Symbol	Direction	Description	Default
[8:0]	GPJ	R/W	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 Reserved value will be read.	pin state

Register 33-26: PortJ Configuration Registers(GPJCON, offset=0x94)

Field	Symbol	Direction	Description	Default
[1:0]	GPJ0	R/W	00=port input/eint group6[18], 01=output, 10= SRAM ADDR[18] 11= Keyboard ROW[0]	10
[3:2]	GPJ1	R/W	00=port input/eint group6[19], 01=output, 10= SRAM ADDR[19] 11= Keyboard ROW[1]	10
[5:4]	GPJ2	R/W	00=port input/eint group6[20], 01=output, 10= SRAM ADDR[20] 11= Keyboard ROW[2]	10
[7:6]	GPJ3	R/W	00=port input/eint group6[21], 01=output, 10= SRAM ADDR[21] 11= Keyboard ROW[3]	10
[9:8]	GPJ4	R/W	00=port input/eint group6[22], 01=output, 10= SRAM ADDR[22] 11= Keyboard ROW[4]	10
[11:10]	GPJ5	R/W	00=port input/eint group6[23], 01=output, 10= SRAM CSN0 11= Keyboard ROW[5]	10
[13:12]	GPJ6	R/W	00=port input/eint group6[24], 01=output, 10= SRAM OEN 11= Keyboard ROW[6]	10
[15:14]	GPJ7	R/W	00=port input/eint group6[25], 01=output, 10= SRAM WEN 11= Keyboard ROW[7]	10
[17:16]	GPJ8	R/W	00=port input/eint group6[26], 01=output, 10= Ethernet COL 11= Keyboard COL[8]	10

Register 33-27: PortJ Pull UP/Down Registers(GPJPUD, offset=0x98)

Field	Symbol	Direction	Description	Default
[8]	GPJ[8]	R/W	1 : pull down enable, 0 : pull down disable	0x1
[7:0]	GPJ[n]	R/W	1 : pull up enable, 0 : pull up disable	0xff

Register 33-28: PortK Data Registers(GPKDAT, offset=0xa0)

Field	Symbol	Direction	Description	Default
[15:0]	GPK	R/W	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 Reserved value will be read.	pin state

Register 33-29: PortK Configuration Registers(GPKCON, offset=0xa4)

Field	Symbol	Direction	Description	Default
[1:0]	GPK0	R/W	00=port input/eint group5[0], 01=output, 10= ETH TCLK 11= KB COL[0]	00
[3:2]	GPK1	R/W	00=port input/eint group5[1] 01=output, 10=ETH RCLK 11= KB COL[1]	00
[5:4]	GPK2	R/W	00=port input/eint group5[2] 01=output, 10= ETH MDC 11= KB COL[2]	00
[7:6]	GPK3	R/W	00=port input/eint group5[3] 01=output, 10=ETH MDIO 11= KB COL[3]	00
[9:8]	GPK4	R/W	00=port input/eint group5[4] 01= output 10=ETH TXEN 11= KB COL[4]	00
[11:10]	GPK5	R/W	00=port input/eint group5[5] 01=output, 10=ETH TXD[0] 11= KB ROW[4]	00
[13:12]	GPK6	R/W	00=port input/eint group5[6] 01=output, 10=ETH TXD[1] 11= KB ROW[5]	00
[15:14]	GPK7	R/W	00=port input/eint group5[7] 01=output, 10=ETH TXD[2] 11= KB ROW[6]	00
[17:16]	GPK8	R/W	00=port input/eint group5[8] 01=output, 10=ETH TXD[3] 11= KB ROW[7]	00
[19:18]	GPK9	R/W	00=port input/eint group5[9] 01=output, 10=ETH RXDV 11= KB COL[5]	00
[21:20]	GPK10	R/W	00=port input/eint group5[10] 01=output, 10=ETH RXER 11= KB COL[6]	00
[23:22]	GPK11	R/W	00=port input/eint group5[11] 01=output, 10=ETH RXD[0] 11= KB ROW[0]	00
[25:24]	GPK12	R/W	00=port input/eint group5[12] 01=output, 10=ETH RXD[1] 11= KB ROW[1]	00
[27:26]	GPK13	R/W	00=port input/eint group5[13] 01=output, 10=ETH RXD[2] 11= KB ROW[2]	00
[29:28]	GPK14	R/W	00=port input/eint group5[14] 01=output, 10=ETH RXD[3] 11= KB ROW[3]	00
[31:30]	GPK15	R/W	00=port input/eint group5[15] 01=output, 10=ETH CRS 11= KB COL[7]	00

Register 33-30: PortK Pull UP/Down Registers(GPKPUD, offset=0xa8)

Field	Symbol	Direction	Description	Default
[15:0]	GPK[n]	R/W	1 : pull down enable, 0 : pull down disable	0xffff

Register 33-31: PortL Data Registers(GPLDAT, offset=0xb0)

Field	Symbol	Direction	Description	Default
[12:0]	GPL	R/W	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 Reserved value will be read.	pin state

Register 33-32: PortL Configuration Registers(GPLCON, offset=0xb4)

Field	Symbol	Direction	Description	Default
[1:0]	GPL0	R/W	00=port input/eint group6[27] 01=output, 10=CAMIF OCLK 11=Reserved	00
[3:2]	GPL1	R/W	00=port input/eint group6[28] 01=output, 10=CAMIF HREF 11= ts enable	00
[5:4]	GPL2	R/W	00=port input/eint group6[29] 01=output,	00

			10= CAMIF PCLK 11= ts_clk	
[7:6]	GPL3	R/W	00=port input/eint group6[30] 01=output, 10=CAMIF VSYNC 11= ts_sync	00
[9:8]	GPL4	R/W	00=port input/eint group6[31] 01=output, 10=CAMIF nRST 11= Reserved	00
[11:10]	GPL5	R/W	00=port input/eint group5[24] 01=output, 10=CAMIF DAT[0] 11= ts_dat[0]	00
[13:12]	GPL6	R/W	00=port input/eint group5[25] 01=output, 10= CAMIF DAT[1] 11= ts_dat[1]	00
[15:14]	GPL7	R/W	00=port input/eint group5[26] 01=output, 10= CAMIF DAT[2] 11= ts_dat[2]	00
[17:16]	GPL8	R/W	00=port input/eint group5[27] 01=output, 10= CAMIF DAT[3] 11= ts_dat[3]	00
[19:18]	GPL9	R/W	00=port input/eint group5[28] 01=output, 10= CAMIF DAT[4] 11= ts_dat[4]	00
[21:20]	GPL10	R/W	00=port input/eint group5[29] 01=output, 10= CAMIF DAT[5] 11= ts_dat[5]	00
[23:22]	GPL11	R/W	00=port input/eint group5[30] 01=output, 10= CAMIF DAT[6] 11= ts_dat[6]	00
[25:24]	GPL12	R/W	00=port input/eint group5[31] 01=output, 10= CAMIF DAT[7] 11= ts_dat[7]	00

Register 33-33: PortL Pull UP/Down Registers(GPLPUD, offset=0xb8)

Field	Symbol	Direction	Description	Default
[12:0]	GPL[n]	R/W	1 : pull down enable, 0 : pull down disable	0x1fff

Register 33-34: PortM Data Registers(GPMDAT, offset=0xc0)

Field	Symbol	Direction	Description	Default
[15:0]	GPM	R/W	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 Reserved value will be read.	pin state

Register 33-35: PortM Configuration Registers(GPMCON, offset=0xc4)

Field	Symbol	Direction	Description	Default
[1:0]	GPM0	R/W	00=port input/eint group 5[16] 01=output, 10=IDS VD[0] 11= KB COL[9]	00
[3:2]	GPM1	R/W	00=port input/eint group 5[17] 01=output, 10= IDS VD[1] 11= KB COL[10]	00
[5:4]	GPM2	R/W	00=port input/eint group 5[18] 01=output, 10= IDS VD[2] 11= KB COL[11]	00
[7:6]	GPM3	R/W	00=port input 01=output, 10= IDS VD[3] 11=Reserved	00
[9:8]	GPM4	R/W	00=port input 01=output, 10= IDS VD[4] 11= Reserved	00
[11:10]	GPM5	R/W	00=port input 01=output, 10= IDS VD[5] 11= Reserved	00
[13:12]	GPM6	R/W	00=port input 01=output, 10= IDS VD[6] 11= Reserved	00
[15:14]	GPM7	R/W	00=port input 01=output, 10= IDS VD[7] 11= Reserved	00
[17:16]	GPM8	R/W	00=port input/eint group 5[19] 01=output, 10= IDS VD[8] 11= KB COL[12]	00
[19:18]	GPM9	R/W	00=port input/eint group 5[20] 01=output,	00

			10= IDS VD[9]	11= KB COL[13]	
[21:20]	GPM10	R/W	00=port input 10= IDS VD[10]	01=output, 11= Reserved	00
[23:22]	GPM11	R/W	00=port input 10= IDS VD[11]	01=output, 11= Reserved	00
[25:24]	GPM12	R/W	00=port input 10= IDS VD[12]	01=output, 11= Reserved	00
[27:26]	GPM13	R/W	00=port input 10= IDS VD[13]	01=output, 11= Reserved	00
[29:28]	GPM14	R/W	00=port input 10= IDS VD[14]	01=output, 11= Reserved	00
[31:30]	GPM15	R/W	00=port input 10= IDS VD[15]	01=output, 11= Reserved	00

Register 33-36: PortM Pull UP/Down Registers(GPMPUD, offset=0xc8)

Field	Symbol	Direction	Description	Default
[15:0]	GPM[n]	R/W	1 : pull down enable, 0 : pull down disable	0xffff

Register 33-37: PortN Data Registers(GPNDAT, offset=0xd0)

Field	Symbol	Direction	Description	Default
[12:0]	GPN	R/W	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 Reserved value will be read.	pin state

Register 33-38: PortN Configuration Registers(GPNCON, offset=0xd4)

Field	Symbol	Direction	Description	Default
[1:0]	GPN0	R/W	00=port input/eint group 5[21] 10=IDS VD[16]	01=output, 11= KB COL[14]	00
[3:2]	GPN1	R/W	00=port input/eint group 5[22] 10= IDS VD[17]	01=output, 11= KB COL[15]	00
[5:4]	GPN2	R/W	00=port input/eint group 6[23] 10= IDS VD[18]	01=output, 11= KB COL[16]	00
[7:6]	GPN3	R/W	00=port input 10= IDS VD[19]	01=output, 11=Reserved	00
[9:8]	GPN4	R/W	00=port input 10= IDS VD[20]	01=output, 11= Reserved	00
[11:10]	GPN5	R/W	00=port input 10= IDS VD[21]	01=output, 11= Reserved	00
[13:12]	GPN6	R/W	00=port input 10= IDS VD[22]	01=output, 11= Reserved	00
[15:14]	GPN7	R/W	00=port input 10= IDS VD[23]	01=output, 11= Reserved	00
[17:16]	GPN8	R/W	00=port input 10= IDS VCLK	01=output, 11= Reserved	00
[19:18]	GPN9	R/W	00=port input 10= IDS VSYNC	01=output, 11= Reserved	00
[21:20]	GPN10	R/W	00=port input 10= IDS HSYNC	01=output, 11= Reserved	00
[23:22]	GPN11	R/W	00=port input 10= IDS VDEN	01=output, 11= Reserved	00
[25:24]	GPN12	R/W	00=port input 10= IDS PWREN	01=output, 11= Reserved	00

Register 33-39: PortN Pull UP/Down Registers(GPNPUD, offset=0xd8)

Field	Symbol	Direction	Description	Default
[12:0]	GPN[n]	R/W	1 : pull down enable, 0 : pull down disable	0x1fff

Register 33-40: PortO Data Registers(GPODAT, offset=0xe0)

Field	Symbol	Direction	Description	Default
[15:0]	GPO	R/W	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 Reserved value will be read.	0x780

Register 33-41: PortO Configuration Registers(GPOCON, offset=0xe4)

Field	Symbol	Direction	Description	Default
[1:0]	GPO0	R/W	00=port input 01=output, 10=SDIO1 CLK 11=Reserved	00
[3:2]	GPO1	R/W	00=port input 01=output, 10= SDIO1 CMD 11= Reserved	00
[5:4]	GPO2	R/W	00=port input/eint group4[24] 01=output, 10= SDIO1 DAT[0] 11= Reserved	00
[7:6]	GPO3	R/W	00=port input/eint group4[25] 01=output, 10= SDIO1 DAT[1] 11= Reserved	00
[9:8]	GPO4	R/W	00=port input/eint group4[26] 01=output, 10= SDIO1 DAT[2] 11= Reserved	00
[11:10]	GPO5	R/W	00=port input/eint group4[27] 01=output, 10= SDIO1 DAT[3] 11= Reserved	00
[13:12]	GPO6	R/W	00=port input 01=output, 10=SDIO2 CLK 11=Master SSI2 CLK	00
[15:14]	GPO7	R/W	00=port input 01=output, 10= SDIO2 CMD 11=Master SSI2 RXD	00
[17:16]	GPO8	R/W	00=port input/eint group4[28] 01=output, 10= SDIO2 DAT[0] 11= Master SSI2 TXD	00
[19:18]	GPO9	R/W	00=port input/eint group4[29] 01=output, 10= SDIO2 DAT[1] 11=Reserved	00
[21:20]	GPO10	R/W	00=port input/eint group4[30] 01=output, 10= SDIO2 DAT[2] 11= Reserved	00
[23:22]	GPO11	R/W	00=port input/eint group4[31] 01=output, 10= SDIO2 DAT[3] 11= Master SSI2 CSN	00
[25:24]	GPO12	R/W	00=port input 01=output, 10= Uart1_DSR 11= CF DMA_ACK	00
[27:26]	GPO13	R/W	00=port input 01=output, 10= Uart1_DCD 11= Reserved	00
[29:28]	GPO14	R/W	00=port input 01=output, 10= Uart1_RI 11= CF DMA_REQ	00
[31:30]	GPO15	R/W	00=port input 01=output, 10= Uart1_DTR 11= Reserved	00

Register 33-42: PortO Pull UP/Down Registers(GPOPUD, offset=0xe8)

Field	Symbol	Direction	Description	Default
[10:0]	GPO[n]	R/W	1 : pull up enable, 0 : pull up disable	0x0

Register 33-43: PortP Data Registers(GPPDAT, offset=0xf0)

Field	Symbol	Direction	Description	Default
[11:0]	GPP	R/W	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 Reserved value will be read.	0xdbf

Register 33-44: PortP Configuration Registers(GPPCON, offset=0xF4)

Field	Symbol	Direction	Description	Default
[1:0]	GPP0	R/W	00= port input 01=output, 10= SRAM addr[23]/CF_ADDR[0] 11=Reserved	10
[3:2]	GPP1	R/W	00= port input 01=output, 10= SRAM addr[24] /CF_ADDR[1] 11=Reserved	10
[5:4]	GPP2	R/W	00= port input 01=output, 10= SRAM addr[25] /CF_ADDR[2] 11= Reserved	10
[7:6]	GPP3	R/W	00= port input 01=output, 10= CF IORD 11=Reserved	10
[9:8]	GPP4	R/W	00= port input 01=output, 10= CF IOWR 11= Reserved	10
[11:10]	GPP5	R/W	00= port input 01=output, 10= CF Reset 11= Reserved	10
[13:12]	GPP6	R/W	00= port input 01=output, 10= CF Power 11= Reserved	10
[15:14]	GPP7	R/W	00= port input 01=output, 10= CF CS0N 11= Reserved	10
[17:16]	GPP8	R/W	00= port input 01=output, 10= CF CS1N 11= Reserved	10
[19:18]	GPP9	R/W	00= port input 01=output, 10= CF IORDY/SRAM nWAIT 11= Reserved	10
[21:20]	GPP10	R/W	00= port input 01=output, 10= CF INT 11= Reserved	10
[23:22]	GPP11	R/W	00= port input 01=output, 10= CF CD1N 11= Reserved	10

Register 33-45: PortP Pull UP/Down Registers(GPPPUD, offset=0xF8)

Field	Symbol	Direction	Description	Default
[11:0]	GPP[n]	R/W	1 : pull up enable 0 : pull up disable	0xfff

Register 33-46: PortQ Data Registers(GPQDAT, offset=0x100)

Field	Symbol	Direction	Description	Default
[5:0]	GPQ	R/W	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 Reserved value will be read.	0x19

Register 33-47: PortQ Configuration Registers(GPQCON, offset=0x104)

Field	Symbol	Direction	Description	Default
[1:0]	GPQ0	R/W	00=port input 01=output, 10=Xnd RNB 11=Reserved	10
[3:2]	GPQ1	R/W	00=port input 01=output, 10= Xnd CLE 11=Reserved	10
[5:4]	GPQ2	R/W	00=port input 01=output, 10= Xnd ALE 11= Reserved	10
[7:6]	GPQ3	R/W	00=port input 01=output,	10

			10= Xnd WEN 11=Reserved	
[9:8]	GPQ4	R/W	00=port input 01=output, 10= Xnd REN 11= Reserved	10
[11:10]	GPQ5	R/W	00=port input 01=output, 10= Xnd CSN0 11=Reserved	10

Register 33-48: PortQ Pull UP/Down Registers(GPQPUD, offset=0x108)

Field	Symbol	Direction	Description	Default
[5]	GPQ5	R/W	1 : pull up enable, 0 : pull up disable	1
[4]	GPQ4	R/W	1 : pull up enable, 0 : pull up disable	1
[3]	GPQ3	R/W	1 : pull up enable, 0 : pull up disable	1
[2]	GPQ2	R/W	1 : pull down enable, 0 : pull down disable	1
[1]	GPQ1	R/W	1 : pull down enable, 0 : pull down disable	1
[0]	GPQ0	R/W	1 : pull up enable, 0 : pull up disable	1

Register 33-49: PortR Data Registers(GPRDAT, offset=0x110)

Field	Symbol	Direction	Description	Default
[15:0]	GPR	R/W	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 Reserved value will be read.	0x19

Register 33-50: PortQ Configuration Registers(GPRCON, offset=0x114)

Field	Symbol	Direction	Description	Default
[1:0]	GPR0	R/W	00=port input 01=output, 10=Flash DAT[0] 11=Reserved	10
[3:2]	GPR1	R/W	00=port input 01=output, 10=Flash DAT[1] 11=Reserved	10
[5:4]	GPR2	R/W	00=port input 01=output, 10=Flash DAT[2] 11=Reserved	10
[7:6]	GPR3	R/W	00=port input 01=output, 10=Flash DAT[3] 11=Reserved	10
[9:8]	GPR4	R/W	00=port input 01=output, 10=Flash DAT[4] 11=Reserved	10
[11:10]	GPR5	R/W	00=port input 01=output, 10=Flash DAT[5] 11=Reserved	10
[13:12]	GPR6	R/W	00=port input 01=output, 10=Flash DAT[6] 11=Reserved	10
[15:14]	GPR7	R/W	00=port input 01=output, 10=Flash DAT[7] 11=Reserved	10
[17:16]	GPR8	R/W	00=port input 01=output, 10=Flash DAT[8] 11=Reserved	10
[19:18]	GPR9	R/W	00=port input 01=output, 10=Flash DAT[9] 11=Reserved	10
[21:20]	GPR10	R/W	00=port input 01=output, 10=Flash DAT[10] 11=Reserved	10
[23:22]	GPR11	R/W	00=port input 01=output, 10=Flash DAT[11] 11=Reserved	10
[25:24]	GPR12	R/W	00=port input 01=output, 10=Flash DAT[12] 11=Reserved	10
[27:26]	GPR13	R/W	00=port input 01=output, 10=Flash DAT[13] 11=Reserved	10
[29:28]	GPR14	R/W	00=port input 01=output, 10=Flash DAT[14] 11=Reserved	10
[31:30]	GPR15	R/W	00=port input 01=output, 10=Flash DAT[15] 11=Reserved	10

Register 33-51: PortR Pull UP/Down Registers(GPRPUD, offset=0x118)

Field	Symbol	Direction	Description	Default
[15:0]	GPR	R/W	1 : pull up enable, 0 : pull up disable	1

Register 33-52: External Interrupt configuration Register (EINTCON, offset=0x200)

Field	Symbol	Direction	Description	Default
[2:0]	EINT[0]	R/W	Setting the signaling method of the EINT[0] 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[6:4]	EINT[1]	R/W	Setting the signaling method of the EINT[1] 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[10:8]	EINT[2]	R/W	Setting the signaling method of the EINT[2] 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[14:12]	EINT[3]	R/W	Setting the signaling method of the EINT[3] 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[18:16]	EINT[4]	R/W	Setting the signaling method of the EINT[4] 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[22:20]	EINT[5]	R/W	Setting the signaling method of the EINT[5] 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

Register 33-53: External Interrupt Filter Control Register (EINTFLTCON0, offset=0x204)

Field	Symbol	Direction	Description	Default
[6:0]	EINT[0]	R/W	Filtering width of EINT[0], digital filter(clock count of PCLK), Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[7]	FLTWN	R/W	Filter Enable for EINT[0] 0 = disables 1 = enabled	0
[14:8]	EINT[1]	R/W	Filtering width of EINT[1], digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[15]	FLTWN	R/W	Filter Enable for EINT[1] 0 = disables 1 = enabled	0
[22:16]	EINT[2]	R/W	Filtering width of EINT[2], digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[23]	FLTWN	R/W	Filter Enable for EINT[2] 0 = disables 1 = enabled	0
[30:24]	EINT[3]	R/W	Filtering width of EINT[3], digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[31]	FLTWN	R/W	Filter Enable for EINT[3] 0 = disables 1 = enabled	0

The debounce filter is supported for external interrupt and illustrated as following figure.

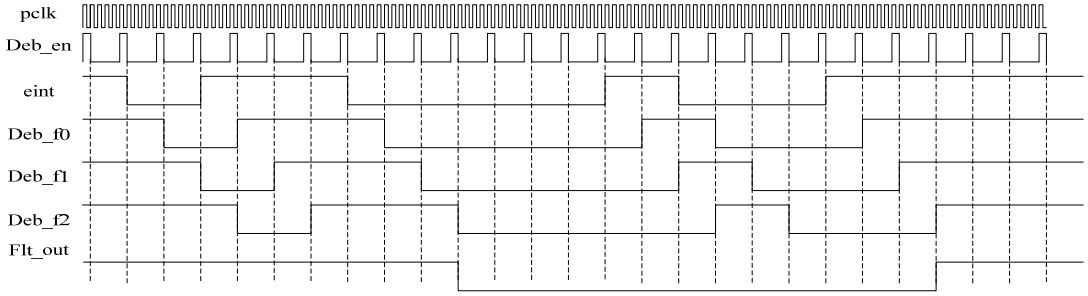


Figure 33-1: Debounce Filter Timing

Register 33-54: External Interrupt Filter Control Register (EINTFLTCON1, offset=0x208)

Field	Symbol	Direction	Description	Default
[6:0]	EINT[4]	R/W	Filtering width of EINT[4], digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[7]	FLTWN	R/W	Filter Enable for EINT[4] 0 = disables 1 = enabled	0
[14:8]	EINT[5]	R/W	Filtering width of EINT[5], digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[15]	FLTWN	R/W	Filter Enable for EINT[1] 0 = disables 1 = enabled	0

Register 33-55: External Interrupt Group configuration Register (EINTGCON, offset=0x210)

Field	Symbol	Direction	Description	Default
[2:0]	EINT1	R/W	Setting the signaling method of the EINT Group 1 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[6:4]	EINT2	R/W	Setting the signaling method of the EINT Group 2 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[10:8]	EINT3	R/W	Setting the signaling method of the EINT Group 3 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[14:12]	EINT4	R/W	Setting the signaling method of the EINT Group 4 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[18:16]	EINT5	R/W	Setting the signaling method of the EINT Group 5 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000
[22:20]	EINT6	R/W	Setting the signaling method of the EINT Group 6 000 = Low level 001 = High level 01x = Falling edge triggered 10x = Rising edge triggered 11x = Both edge triggered	000

Register 33-56: External Interrupt Group Filter Control Register (EINTGFLTCON0, offset=0x214)

Field	Symbol	Direction	Description	Default
[6:0]	EINT 1	R/W	Filtering width of EINT Group 1, digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[7]	FLTWN	R/W	Filter Enable for EINT Group 1 0 = disables 1 = enabled	0
[14:8]	EINT 2	R/W	Filtering width of EINT Group 2, digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[15]	FLTWN	R/W	Filter Enable for EINT Group 2 0 = disables 1 = enabled	0
[22:16]	EINT 3	R/W	Filtering width of EINT Group 3, digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[23]	FLTWN	R/W	Filter Enable for EINT Group 3 0 = disables 1 = enabled	0
[30:24]	EINT 4	R/W	Filtering width of EINT Group 4, digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[31]	FLTWN	R/W	Filter Enable for EINT Group 4 0 = disables 1 = enabled	0

Register 33-57: External Interrupt Group Filter Control Register (EINTGFLTCON1, offset=0x218)

Field	Symbol	Direction	Description	Default
[6:0]	EINT 5	R/W	Filtering width of EINT Group 5, digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[7]	FLTWN	R/W	Filter Enable for EINT Group 5 0 = disables 1 = enabled	0
[14:8]	EINT 6	R/W	Filtering width of EINT Group 6, digital filter(clock count of PCLK) Note : Internal Filter use debounce filter, and actual filter count is four times of filtering width.	00000
[15]	FLTWN	R/W	Filter Enable for EINT Group 6 0 = disables 1 = enabled	0

Register 33-58: External Interrupt Group 1 Mask register (EINTG1MASK, offset=0x21C)

Field	Symbol	Direction	Description	Default
[n] n = 0~7	EINTMASK[n]	R/W	0 = Enable Interrupt 1= Masked	0x1

Register 33-59: External Interrupt Group 2 Mask register (EINTG2MASK, offset=0x220)

Field	Symbol	Direction	Description	Default
[n] n = 0~4	EINTMASK[n]	R/W	0 = Enable Interrupt 1= Masked	0x1

Register 33-60: External Interrupt Group 3 Mask register (EINTG3MASK, offset=0x224)

Field	Symbol	Direction	Description	Default
[n] n = 0~7	EINTMASK[n]	R/W	0 = Enable Interrupt 1= Masked	0x1

Register 33-61: External Interrupt Group 4 Mask register (EINTG4MASK, offset=0x228)

Field	Symbol	Direction	Description	Default
[n] n = 0~31	EINTMASK[n]	R/W	0 = Enable Interrupt 1= Masked	0x1

Register 33-62: External Interrupt Group 5 Mask register (EINTG5MASK, offset=0x22C)

Field	Symbol	Direction	Description	Default
[n] n = 0~31	EINTMASK[n]	R/W	0 = Enable Interrupt 1= Masked	0x1

Register 33-63: External Interrupt Group 6 Mask register (EINTG6MASK, offset=0x230)

Field	Symbol	Direction	Description	Default
[n] n = 0~31	EINTMASK[n]	R/W	0 = Enable Interrupt 1= Masked	0x1

Register 33-64: External Interrupt Group 1 Pending register (EINTG1PEND, offset=0x234)

Field	Symbol	Direction	Description	Default
[n] n = 0~7	EINTPEND[n]	R/W	0 = Not occur 1= Occur interrupt Each bit is cleared by writing "1".	0x0

Register 33-65: External Interrupt Group 2 Pending register (EINTG2PEND, offset=0x238)

Field	Symbol	Direction	Description	Default
[n] n = 0~4	EINTPEND[n]	R/W	0 = Not occur 1= Occur interrupt Each bit is cleared by writing "1".	0x0

Register 33-66: External Interrupt Group 3 Pending register (EINTG3PEND, offset=0x23C)

Field	Symbol	Direction	Description	Default
[n] n = 0~7	EINTPEND[n]	R/W	0 = Not occur 1= Occur interrupt Each bit is cleared by writing "1".	0x0

Register 33-67: External Interrupt Group 4 Pending register (EINTG4PEND, offset=0x240)

Field	Symbol	Direction	Description	Default
[n] n = 0~31	EINTPEND[n]	R/W	0 = Not occur 1= Occur interrupt Each bit is cleared by writing "1".	0x0

Register 33-68: External Interrupt Group 5 Pending Register (EINTG5PEND, offset=0x244)

Field	Symbol	Direction	Description	Default
[n] n = 0~31	EINTPEND[n]	R/W	0 = Not occur 1= Occur interrupt Each bit is cleared by writing "1".	0x0

Register 33-69: External Interrupt Group 6 Pending Register (EINTG6PEND, offset=0x248)

Field	Symbol	Direction	Description	Default
[n] n = 0~31	EINTPEND[n]	R/W	0 = Not occur 1= Occur interrupt Each bit is cleared by writing "1".	0x0

34 Electrical Data

34.1 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.

Parameter	Symbol	Min	Max	Unit
DC Supply Voltage	VDDIO_VV, VDDIO_SD, VDDIO_CI, VDDIO_ET, VDDIO_OSC, VDDIO_RTC	0	3.6	V
	AHVDD_DPLL, AHVDD_APLL, AHVDD_EPLL	0	2.75	V
	DVDD_DPLL, DVDD_APLL, DVDD_EPLL	0	1.32	V
	VDDA_USBH1, VDDA_USBH0, VDDA_USBO	0	3.6	V
	VDDL_USBH1, VDDL_USBH0 , VDDL_USBO	0	1.32	V
	VDDP_DDR	0	1.32	V
	VDDIO_DDR	0	3.6	V
	VDDR_DDR	0	3.6	V
	AVDD_OSC	0	1.32	V
DC Input Voltage	1.8V input	0	2.5	V
	2.5V input	0	2.75	V
	3.3V input	0	5.0	V
DC Output Voltage	1.8V output buffer	0	1.98	V
	2.5V output buffer	0	2.75	V
	3.3V output buffer	0	3.6	V
Storage Temperature	TSTG	-60	150	ºC

34.2 Recommended Operating Conditions

Parameter	Symbol	Min	Typical	Max	Unit
DC Supply Voltage for RTC	AVDD_OSC VDD_RTC	1.08	1.2	1.32	V
	VDDIO_RTC VDDIO_OSC	3.0	3.3	3.6	V
DC Supply Voltage for USB	VDDA_USBH1 VDDA_USBH0 VDDA_USBO	3.0	3.3	3.6	V
	VDDL_USBH1 VDDL_USBH0 VDDL_USBO	1.08	1.2	1.32	V
DC Supply Voltage for PLL	AHVDD_DPLL AHVDD_APLL AHVDD_EPLL	2.25	2.5	2.75	V
	DVDD_DPLL DVDD_APLL DVDD_EPLL	1.08	1.2	1.32	V
DC Supply Voltage for IO	VDDIO_VV VDDIO_SD VDDIO_CI VDDIO_ET	3.0	3.3	3.6	V
DC Supply Voltage for MDDR	VDDP_DDR	1.08	1.2	1.32	V
	VDDIO_DDR	1.7	1.8	1.9	V
	VDDR_DDR	GND	GND	GND	--
DC Supply Voltage for DDR2	VDDP_DDR	1.08	1.2	1.32	V
	VDDIO_DDR	1.7	1.8	1.9	V
	VDDR_DDR	0.85	0.9	0.95	V

DC Supply Voltage for DDR1	VDDP_DDR	1.08	1.2	1.32	V
	VDDIO_DDR	2.3	2.5	2.7	V
	VDDR_DDR	1.15	1.25	1.35	V
Input low Voltage	V_{IL}	-0.3		0.8	V
Input high Voltage	V_{IH}	2		3.6	V
Output low Voltage	V_{OL}			0.4	V
Output high Voltage	V_{OH}	2.4			V
Pull-up Resister	R_{PU}	26K	38K	60K	Ω
Pull-down Resister	R_{PD}	34K	48K	82K	Ω
Junction Temperature	Tj	-40		125	°C
Operating Temperature	TA	-25		85	°C

34.3 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.

Normal IO DC Electrical Characteristics

Parameter	Symbol	Min	Typical	Max	Unit
Input low Voltage	V_{IL}	-0.3		0.8	V
Input high Voltage	V_{IH}	2		3.6	V
Threshold Point	VT	1.3	1.40	1.48	V
Schmitt Trig. Low to High Threshold Point	VT^+	1.56	1.68	1.77	V
Schmitt Trig. High to Low Threshold Point	VT^-	1.15	1.23	1.31	V
Input Leakage Current @*VI*=3.3V or 0V				±10μ	A
Tri-state Output Leakage Current @ *VO*=3.3V or 0V				±10μ	A
Output low Voltage	V_{OL}			0.4	V
Output high Voltage	V_{OH}	2.4			V
Pull-up Resister	R_{PU}	26K	38K	60K	Ω
Pull-down Resister	R_{PD}	34K	48K	82K	Ω
Junction Temperature	Tj	-40		125	°C
Operating Temperature	T_{OPT}	-25		85	°C

Special Memory I/O PAD DC Electrical Characteristics (DDR2)

Parameter	Symbol	Min	Typical	Max	Unit
DDR2 IO core voltage	VDDP_DDR	1.08	1.2	1.32	V
DDR2 IO voltage	VDDIO_DDR	1.7	1.8	1.9	V
DDR2 reference voltage	VDDR_DDR	0.85	0.9	0.95	V
Input Voltage	V_I	0	VDDIO_DDR	1.9	V
Output Voltage	V_O	0	VDDIO_DDR	1.9	V
Core Input Voltage	V_{IC}	0	VDDP_DDR	1.32	V
Core Output Voltage	V_{OC}	0	VDDP_DDR	1.32	V
Operating Temperature	T_{OPT}	-25	25	85	°C

Special Memory I/O PAD DC Electrical Characteristics (DDR1)

Parameter	Symbol	Min	Typical	Max	Unit
DDR1 IO core voltage	VDDP_DDR	1.08	1.2	1.32	V
DDR1 IO voltage	VDDIO_DDR	2.3	2.5	2.7	V
DDR1 reference voltage	VDDR_DDR	1.15	1.25	1.35	V
Input Voltage	V_I	0	VDDIO_DDR	2.7	V
Output Voltage	V_O	0	VDDIO_DDR	2.7	V

Core Input Voltage	V_{IC}	0	VDDP_DDR	1.32	V
Core Output Voltage	V_{OC}	0	VDDP_DDR	1.32	V
Operating Temperature	T_{OPT}	-25	25	85	°C

Special Memory I/O PAD DC Electrical Characteristics (MDDR)

Parameter	Symbol	Min	Typical	Max	Unit
mDDR IO core voltage	VDDP_DDR	1.08	1.2	1.32	V
mDDR IO voltage	VDDIO_DDR	1.7	1.8	1.9	V
mDDR reference voltage	VDDR_DDR	GND	GND	GND	V
Input Low Voltage	V_{IL}	-0.3		0.3* VDDIO_DDR	V
Input High Voltage	V_{IH}	0.7* VDDIO_DDR		VDDIO_DDR+ 0.3	V
Output High Voltage	V_{OH}	0.9* VDDIO_DDR			V
Output Low Voltage	V_{OL}			0.1* VDDIO_DDR	V
Operating Temperature	T_{OPT}	-25	25	85	°C

USB/USB OTG IO DC Electrical Characteristics(LOW/FULL speed)

Parameter	Symbol	Min	Typical	Max	Unit
USB Analog supply voltage	VDDA_USBH1 VDDA_USBH0 VDDA_USBO	3.0	3.3	3.6	V
USB core supply voltage	VDDL_USBH1 VDDL_USBH0 VDDL_USBO	1.08	1.2	1.32	V
PAD Input Low Voltage	V_{IL}			0.8	V
PAD Input High Voltage	V_{IH}	2.0			V
Output High Voltage	V_{OH}	2.8		3.6	V
Output Low Voltage	V_{OL}	0		0.3	V
Differential input sensitivity	V_{DI}(|PADP-PADM|)	0.2			V
Common mode voltage range	V_{CM}	0.8		2.5	V
Differential output signal cross-point voltage	V_{CRS}	1.3		2.0	V
Pull-up Resister	R_{PU}(IDLE)	900		1575	Ω
	R_{PU}(Recieving)	1425		3090	Ω
Pull-down Resister	R_{PD}	14.25K		28.20K	Ω
Driver output resistance	Z_{DRV}		10		Ω
Transceiver pad capacitance	C_{IN}			20	pf

USB/USB OTG IO DC Electrical Characteristics(high speed)

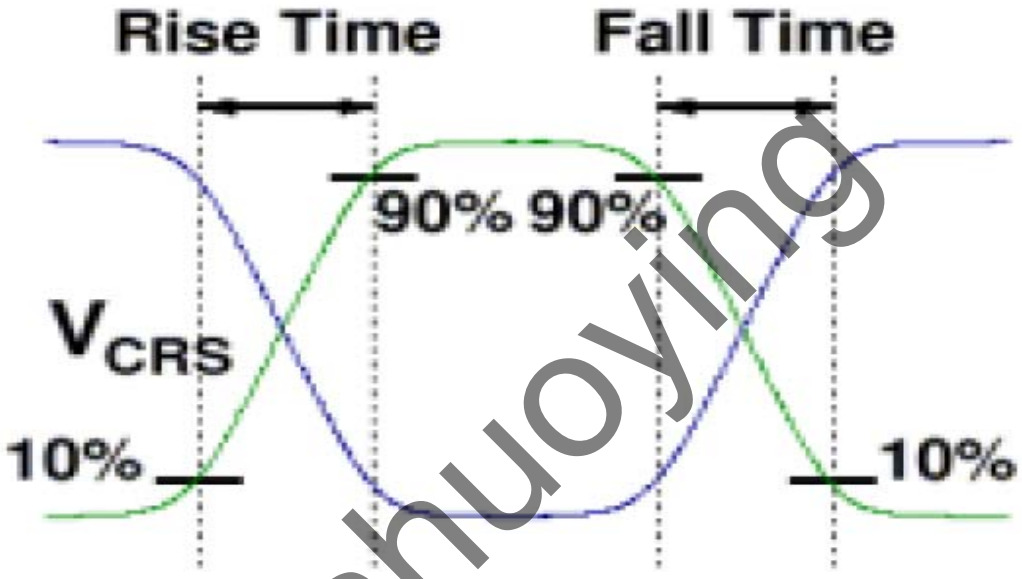
Parameter	Symbol	Min	Typical	Max	Unit
USB Analog supply voltage	VDDA_USBH1 VDDA_USBH0 VDDA_USBO	3.0	3.3	3.6	V
USB core supply voltage	VDDL_USBH1 VDDL_USBH0 VDDL_USBO	1.08	1.2	1.32	V
High-speed differential input signal level	V_{HSDI}(|PADP-PADM|)	150m			V
High-speed SQ detection threshold	V_{HSSQ}	100m		150m	V

High-speed common mode voltage range	V_{HSCM}	-50m		500m	V
High-speed data signaling high	V_{SHOH}	360m		440m	V
High-speed data signaling low	V_{SHOL}	-10m		10m	V
Chirp J level	V_{CHIRPJ}	700m		1100m	V
Chirp K level	V_{CHIRPK}	-900m		-500m	V
High-speed driver output resistance	Z_{HDRV}	40.5		49.5	Ω

34.4 A.C. Electrical Characteristics

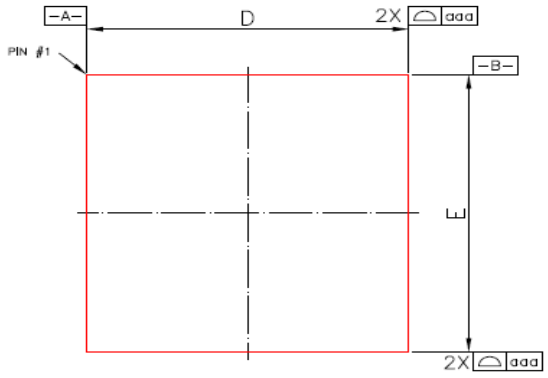
USB AC Electrical characteristics

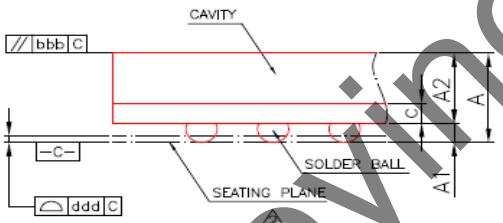
The figure below shows the definition of USB signal rise time and fall time.



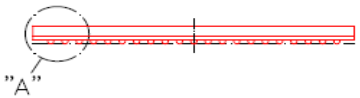
Parameter	Symbol	Condition	Min	Typical	Max	Unit
		LOW SPEED				
Low-speed driver rise time	T_{LR}	CL=100pF	75		300	ns
Low-speed driver fall time	T_{LF}	CL=100pF	75		300	ns
Low-speed rise/fall time matching	TL_{RFM}	$TL_{RFM} = T_{LR} / T_{LF}$	80%		125%	
		FULL SPEED				
Full-speed driver rise time	T_{FR}	CL=50pF	4		20	ns
Full-speed driver fall time	T_{FF}	CL=50pF	4		20	ns
Full-speed rise/fall time matching	TF_{RFM}	$TF_{RFM} = T_{FR} / T_{FF}$	90%		111%	
		HIGH SPEED				
High-speed driver rise time	T_{HSR}	Z_{HSDRV}=45Ω	500		900	ps
High-speed driver fall time	T_{HSF}	Z_{HSDRV}=45Ω	500		900	ps

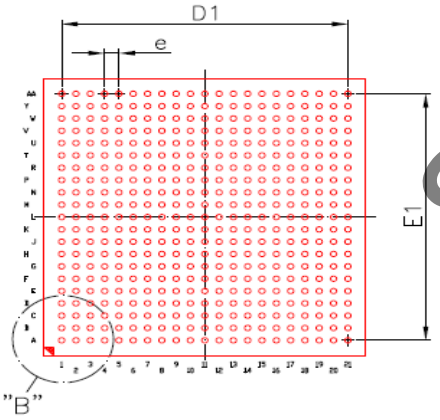
35 Mechanical Data

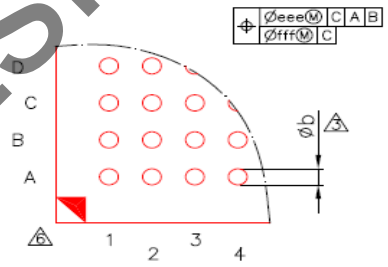




DETAIL : "A"







DETAIL : "B"

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	---	---	1.30	---	---	0.0512
A1	0.20	0.25	0.30	0.0079	0.0098	0.0118
A2	0.86	0.91	0.96	0.0339	0.0358	0.0378
c	0.22	0.26	0.30	0.0087	0.0102	0.0118
D	17.90	18.00	18.10	0.7047	0.7087	0.7126
E	17.90	18.00	18.10	0.7047	0.7087	0.7126
D1	---	16.00	---	---	0.6299	---
E1	---	16.00	---	---	0.6299	---
e	---	0.80	---	---	0.0315	---
b	0.30	0.35	0.40	0.0118	0.0138	0.0157
aaa		0.15			0.0059	
bbb		0.20			0.0079	
ddd		0.20			0.0079	
eee		0.15			0.0059	
fff		0.08			0.0031	
MD/ME		21/21			21/21	

NOTE :

1. CONTROLLING DIMENSION : MILLIMETER.
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. THERE SHALL BE A MINIMUM CLEARANCE OF 0.25mm BETWEEN THE EDGE OF THE SOLDER BALL AND THE BODY EDGE.
5. REFERENCE DOCUMENT : JEDEC MO-205
6. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY .
7. SPECIAL CHARACTERISTICS C CLASS: bbb, ddd

COPY
CONTROLLED