

ARM926EJ-S Based 32-bit Microcontroller

N9H20 Series Technical Reference Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Publication Release Date: April 19, 2018

Revision V1.0

TABLE OF CONTENTS

1	GENERAL DESCRIPTION.....	11
2	FEATURES	12
3	PARTS INFORMATION LIST AND PIN CONFIGURATION.....	17
3.1	N9H20 Series Part Number Naming Guide.....	17
3.2	N9H20 Series Part Selection Guide	18
3.3	Pin Configuration	19
3.3.1	N9H20 series Pin Diagram	19
3.4	Pin Description.....	20
4	BLOCK DIAGRAM.....	29
4.1	N9H20 Series Block Diagram.....	29
5	FUNCTIONAL DESCRIPTION.....	30
5.1	ARM® ARM926EJ-S CPU Core.....	30
5.2	System Manager.....	31
5.2.1	Overview	31
5.2.2	System Memory Map	31
5.2.3	Power-On Settings	34
5.2.4	Bus Arbitration Mode.....	35
5.2.5	Power Management Mode	37
5.2.6	IBR (Internal Boot ROM) Sequence	38
5.2.7	System Management Control Registers	39
5.3	Clock Controller(CLK).....	78
5.3.1	Overview	78
5.3.2	Block diagram	78
5.3.3	Clock Controller Overview.....	79
5.3.4	Control Registers.....	89
5.4	SDRAM Controller (SDRAM).....	115
5.4.1	Overview	115
5.4.2	Block Diagram.....	116
5.4.3	SDRAM Control Timing	116
5.4.4	SDRAM Power-Up Sequence	117
5.4.5	SDRAM Interface Signals	119
5.4.6	SDRAM Components Supported	119
5.4.7	AHB Bus Address Mapping to SDRAM Bus	120
5.4.8	SDRAM Control Registers Map.....	122
5.4.9	Register Details	123
5.5	BitBlit Blitting Accelerator(BitBLT).....	144

5.5.1 Overview	144
5.5.2 Features	144
5.5.3 Architecture	145
5.5.4 Register Map	147
5.5.5 Register Description	148
5.6 JPEG Codec(JPEG).....	163
5.6.1 Overview	163
5.6.2 Features	163
5.6.3 JPEG Encode.....	164
5.6.4 JPEG Decode	167
5.6.5 JPEG Codec Interrupt.....	168
5.6.6 Register Map	169
5.6.7 Register Description	171
5.7 Display Interface Controller (VPOST)	220
5.7.1 Overview	220
5.7.2 Features	220
5.7.3 POST Controller Interface	221
5.7.4 VPOST Controller Block Diagram	223
5.7.5 Functional Description	224
5.7.6 Registers Map	225
5.7.7 Registers Description	227
5.8 Sound Processing Unit (SPU)	266
5.8.1 Overview	266
5.8.2 Features	266
5.8.3 SPU Block Diagram	267
5.8.4 Pin descriptions	267
5.8.5 Register Map	268
5.8.6 Register Description	269
5.9 I2S Controller(I2S).....	291
5.9.1 Overview	291
5.9.2 Features	291
5.9.3 Block diagram	292
5.9.4 I2S Interface	293
5.9.5 Register Map	294
5.9.6 Register Description	295
5.10 Storage Interface Controller(SIC).....	313
5.10.1 Overview	313
5.10.2 Features	313
5.10.3 Block Diagram and Card Pad Assignment	314
5.10.4 Registers Map	316

5.11	SIC DMA Controller	318
5.11.1	Overview	318
5.11.2	Features	318
5.11.3	Flash Memory Interface Controller (FMI)	327
5.11.4	Block Diagram	330
5.11.5	Function Description	331
5.11.6	Register Map	334
5.11.7	Register Description	336
5.12	USB 2.0 Device Controller (USBD)	381
5.12.1	Overview	381
5.12.2	Features	381
5.12.3	Internal Block Diagram	382
5.12.4	USB Device Register Group Summary	382
5.12.5	USB Device Control Registers Map	383
5.12.6	Registers Description	385
5.13	USB Host Controller (UHC)	433
5.13.1	Overview	433
5.13.2	Features	433
5.13.3	Block Diagram	434
5.13.4	Operation	435
5.13.5	Register Mapping	437
5.13.6	Register Details	438
5.14	Enhanced DMA Controller	469
5.14.1	Features	469
5.14.2	Symbol Diagram	470
5.14.3	Block Diagram	471
5.14.4	Function Description	471
5.14.5	Register Map	475
5.14.6	Register Description	477
5.15	Advanced Interrupt Controller	515
5.15.1	Overview	515
5.15.2	Features	515
5.15.3	Interrupt Sources	516
5.15.4	AIC Block Diagram	518
5.15.5	AIC Functional Descriptions	518
5.15.6	AIC Registers Map	521
5.15.7	AIC Control Registers	522
5.16	General Purpose I/O	536
5.16.1	Overview and Features	536
5.16.2	Register Map	537

5.16.3	Register Description.....	539
5.17	TIMER Controller	559
5.17.1	General Timer Controller	559
5.17.2	Features	559
5.17.3	Watchdog Timer.....	559
5.17.4	Register Map	561
5.17.5	Register Description.....	562
5.18	Real Time Clock (RTC).....	569
5.18.1	Overview	569
5.18.2	RTC Features.....	569
5.18.3	RTC Block Diagram	570
5.18.4	RTC Function Description	570
5.18.5	System Power Control Flow.....	572
5.18.6	Force system Power Off Control Flow.....	573
5.18.7	Register Map.....	575
5.18.8	Register Description.....	576
5.19	I2C Synchronous Serial Interface Controller	591
5.19.1	Overview	591
5.19.2	Features	591
5.19.3	I2C Serial Interface Block Diagram.....	591
5.19.4	I2C Protocol.....	592
5.19.5	START or Repeated START signal.....	593
5.19.6	I2C Programming Examples.....	595
5.19.7	1.5 Software I2C Operation	596
5.19.8	Registers Map	598
5.19.9	Register Description.....	599
5.20	PWM-Timer.....	606
5.20.1	Overview	606
5.20.2	Features	606
5.20.3	PWM Timer Start Procedure	607
5.20.4	PWM Architecture	608
5.20.5	Basic Timer Operation	610
5.20.6	PWM Double Buffering and Automatic Reload	610
5.20.7	Modulate Duty Ratio.....	612
5.20.8	Dead-Zone Generator.....	612
5.20.9	PWM Timer Start Procedure.....	613
5.20.10	PWM Timer Stop Procedure.....	613
5.20.11	Register Map	615
5.20.12	Register Description	616
5.21	UART Interface Controller	635

5.21.1 Overview	635
5.21.2 Features	635
5.21.3 Block Diagram	636
5.21.4 Functional Blocks Descriptions	636
5.21.5 Finite State Machine	638
5.21.6 Registers Map	641
5.21.7 Register Description	642
5.22 SPI0/1 Serial Interface Controller (Master/Slave)	662
5.22.1 Overview	662
5.22.2 Feature	662
5.22.3 SPI (MICROWIRE) Block Diagram (Master/Slave)	663
5.22.4 SPI (MICROWIRE) Timing Diagram (Master/Slave)	665
5.22.5 SPI (MICROWIRE) Programming Example	667
5.22.6 Wireless Joystick SPI Programming Example	669
5.22.7 SPI0 (MICROWIRE) Serial Interface Control Register Map	671
5.22.8 SPI 0 (MICROWIRE) Control Register Description	672
5.22.9 SPI1 (MICROWIRE) Serial Interface Control Register Map	682
5.22.10 SPI 1 (MICROWIRE) Control Register Description	683
5.23 Analog to Digital Controller (ADC)	693
5.23.1 Overview	693
5.23.2 Features	693
5.23.3 ADC Functional Description	694
5.23.4 Waiting for Trigger Mode	694
5.23.5 Interface Mode	694
5.23.6 Standby mode	695
5.23.7 Recording path and AGC	695
5.23.8 Analog Macro Control Register Map	697
5.23.9 Register Description	698
5.24 Keypad Interface (KPI)	736
5.24.1 Overview	736
5.24.2 Features:	736
5.24.3 KPI Block Diagram	738
5.24.4 Keypad Interface Register Map	739
5.24.5 Timing Diagram	755
6 ELECTRICAL CHARACTERISTICS.....	756
6.1 Absolute Maximum Ratings	756
6.2 DC Electrical Characteristics	757
6.2.1 N9H20 Series DC Electrical Characteristics	757
6.2.2 ADC Characteristics	757
6.2.3 Audio DAC Characteristics	758

6.3 AC Electrical Characteristics	759
6.3.1 External 12 MHz Crystal.....	759
6.3.2 External 32.768 kHz Low Speed Crystal	760
6.3.3 Power-on Sequence & RESET	761
6.3.4 I2C Interface.....	762
6.3.5 I2S Interface.....	763
6.3.6 LCD/Display Interface	764
6.3.7 SPI Interface.....	766
6.3.8 NAND Interface.....	767
6.3.9 SD Card Interface.....	768
6.3.10 USB PHY Specifications.....	769
6.3.11 Thermal characteristics of N9H20 series LQFP-128 Package	771
7 PACKAGE DIMENSIONS.....	772
7.1 LQFP 128L (14x14x1.4mm footprint 2.0mm).....	772
7.2 PCB Reflow Profile Suggestion.....	773
8 REVISION HISTORY	775

LIST OF FIGURES

Figure 3.1-1 N9H20 Series Part Number Naming Guide.....	17
Figure 3.3-1 N9H20xxN LQFP 128 Pin Diagram	19
Figure 4.1-1 N9H20 Series Block Diagram	29
Figure 5.5-1 Architecture Diagram	145
Figure 5.5-2 Architecture Diagram	146
Figure 5.5-3 Architecture Diagram	146
Figure 5.6-1 1 JPEG Codec block diagram	163
Figure 5.6-2.3 Primary and thumbnail encode	166
Figure 5.6-3. 4 Single mode and continue mode	167
Figure 5.6-4. Specified window decode mode	168
Figure 5.7-1 VPOST Controller Interface Diagram	222
Figure 5.7-2 VPOST LCD Data Output Pin Definition.....	222
Figure 5.7-3 VPOST Controller Block Diagram	223
Figure 5.10-1 SIC Controller Block Diagram.....	314
Figure 5.11-1 FMI Controller Block Diagram	330
Figure 5.11-2 Timing Effect of Above 3 Registers	353
Figure 5.14-1 EDMA Controller Symbol Diagram	470
Figure 5.14-2 EDMA Controller Block Diagram	471
Figure 5.14-3 Peripheral interface signal	472
Figure 5.15-1 AIC Functional Block Diagram.....	518
Figure 5.16-1 Type I GPIO: Input/Output Port with Program Controlled Weakly Pull-High.....	536
Figure 5.17-1 TIMER Block Diagram.....	559
Figure 5.17-2 Watchdog Timer Block Diagram	560
Figure 5.17-3 Watchdog Timer Timing Diagram	561
Figure 5.19-1 I ² C Block Diagram	592
Figure 5.19-2 Data transfer on the I ² C-bus	592
Figure 5.19-3 START and STOP conditions.....	593
Figure 5.19-4 The first byte after the START procedure.....	594
Figure 5.19-5 Bit transfer on the I ² C-bus	594
Figure 5.19-6 Acknowledge on the I ² C-bus	594
Figure 5.19-7 Implementation of Software I ² C	597
Figure 5.20-1 PWM Architecture Diagram	609
Figure 5.20-2 Basic Timer Operation Timing	610
Figure 5.20-3 PWM Double Buffering Illustration.....	611
Figure 7.11-5.20-4 PWM Controller Output Duty Ratio	612

Figure 5.20-5 Dead Zone Generation Operation	613
Figure 5.21-1 UART Block Diagram.....	636
Figure 5.21-2 Auto Flow Control Block Diagram.....	637
Figure 5.22-1 MICROWIRE/SPI Block Diagram (Master/Slave)	663
Figure 5.22-2 MICROWIRE/SPI Timing (Master)	665
Figure 5.22-3 Alternate Phase SCLK Clock Timing (Master)	665
Figure 5.22-4 MICROWIRE/SPI Timing (Slave)	666
Figure 5.22-5 Alternate Phase SCLK Clock Timing (Slave)	666
Figure 5.23-1 ADC Block Diagram.....	693
Figure 5.23-2 Interface for 4-wire Touch Screen	694
Figure 5.24-1 Keypad Interface.....	736
Figure 5.24-2 Keypad architecture.....	737
Figure 5.24-3 Keypad Controller Block Diagram.....	738
Figure 5.24-4 16x8 Keypad Scan Timing Diagram	755
Figure 6.3-1 Typical LXT Crystal Application Circuit.....	760

LIST OF TABLE

Table 5.10.3-1 SD/SDHC/SDIO/MMC Card Pad Assignment 314

Table 5.11.2-1 MA Controller Symbol Diagram 319

Table 5.11.2-2 DMA Controller Block Diagram 320

1 GENERAL DESCRIPTION

The N9H20 series is built on the ARM926EJ-S CPU@200MHz, synchronous DRAM, 2D BitBLT accelerator, LCD panel interface, USB 1.1 Host & USB2.0 HS Device core and integrated with JPEG codec, 32-channel SPU (Sound Processing Unit), ADC, DAC, for meeting various kinds of application needs for saving the BOM cost and makes the best choice for LCD application of cost/performance products.

Maximum resolution is XVGA (1,024x768)@TFT LCD panel. The 2D BitBLT accelerator accelerates the graphic computation to make the rendering smooth and off-load CPU to save power consumption.

The open source code environment provides the product development more flexibility. For reducing system complexity while cutting the BOM cost, the N9H20 series provides MCP (Multi-Chip Package) to ensure higher performance and to minimize the system design efforts. Total BOM cost could be cut by employing 2-layer PCB along with the elimination of external parts, EMI prevention components and saving board design space.

2 FEATURES

- CPU
 - ARM926EJ-S 32-bit RISC CPU with 8KB I-Cache & 8KB D-Cache
 - Frequency up to [200MHz@1.8V](#) core power operation voltage
 - JTAG interface supported for development and debugging
- Internal SRAM & ROM
 - 8KB internal SRAM and 16KB IBR internal booting ROM supported
 - IBR booting messages displayed by UART console for debugging supported
 - Different system booting modes supported:
 - ◆ Memory card
 - SD card or eMMC Flash device
 - SD-to-NAND flash bridge
 - ◆ Raw NAND Flash
 - ◆ SPI Flash
 - ◆ USB
- EDMA (Enhanced DMA)
 - Totally 5 DMA channels supported
 - ◆ 4 peripheral DMA channels for transfer between memory and on-chip peripherals, such as ADC, UART and SPI
 - ◆ One dedicated channel for memory-to-memory transfer
 - Byte, half-word and word data width types supported
 - Single and burst transfer modes supported
 - Block transfer supported in memory-to-memory transfer channel
 - Color format transformation supported in memory-to-memory transfer channel
 - ◆ Source color format could be RGB555, RGB565 and YCbCr422
 - ◆ Destination color format could be RGB555, RGB565 and YCbCr422
 - Auto reload supported for continuous data transfer
 - Interrupt generation supported in the half-of-transfer or end-of-transfer
- JPEG Codec
 - Baseline Sequential mode JPEG codec function compliant with ISO/IEC 10918-1 international JPEG standard supported.
 - Planar Format
 - ◆ Support to encode interleaved YCbCr 4:2:2/4:2:0 and gray-level (Y only) format image
 - ◆ Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0/4:1:1 and gray-level (Y only) format image
 - ◆ Support to decode YCbCr 4:2:2 transpose format
 - ◆ Support arbitrary width and height image encode and decode
 - ◆ Support three programmable quantization-tables
 - ◆ Support standard default Huffman-table and programmable Huffman-table for decode
 - ◆ Support arbitrarily 1X~8X image up-scaling function for encode mode
 - ◆ Support down-scaling function for encode and decode modes
 - ◆ Support specified window decode mode
 - ◆ Support quantization-table adjustment for bit-rate and quality control in encode mode
 - ◆ Support rotate function in encode mode
 - Packet Format
 - ◆ Support to encode interleaved YUYV format input image, output bitstream 4:2:2 and 4:2:0 format
 - ◆ Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0 format image

- ◆ Support decoded output image RGB555, RGB565 and RGB888 formats.
 - ◆ The encoded JPEG bit-stream format is fully compatible with JFIF and EXIF standards
 - ◆ Support arbitrary width and height image encode and decode
 - ◆ Support three programmable quantization-tables
 - ◆ Support standard default Huffman-table and programmable Huffman-table for decode
 - ◆ Support arbitrarily 1X~8X image up-scaling function for encode mode
 - ◆ Support down-scaling function 1X~ 16X for Y422 and Y420, 1X~ 8X for Y444 for decode mode
 - ◆ Support specified window decode mode
 - ◆ Support quantization-table adjustment for bit-rate and quality control in encode mode
- 2D Accelerator
 - BitBLT operation
 - ◆ 2x2 transform matrix with effects:
 - Scale
 - Translate
 - Rotate
 - Shear
 - ◆ Alpha blending and color transformation supported
 - ◆ Source format for operations: supported color format of source bitmap
 - Fill
 - ◆ Rectangle Fill with single color – ARGB8888
 - ◆ Fill with blending effect supported
 - Supported color formats
 - ◆ Source
 - 16 bits/pixel – RGB565
 - 32 bits/pixel – ARGB8888
 - 1 bit/pixel, 2 bits/pixel, 4 bits/pixel, 8 bits/pixel with RGB color palette
 - ◆ Destination
 - 16 bits/pixel – RGB565
 - 32 bits/pixel – ARGB8888
- VPOST
 - 8/16/18/24-bit SYNC type and 8/9/16/18/24-bit MPU type TFT LCD supported
 - Color format supported:
 - ◆ YCbCr422, RGB565, RGB555, and RGB888 color formats supported for data in
 - ◆ YCbCr422, RGB565, RGB555, and RGB888 color formats supported for data out
 - XGA (1024x768), SVGA (800x600), WVGA (800x480), D1 (720x480), VGA (640x480), WQVGA (480x272), QVGA (320x240) and HVGA (640x240) resolution supported
 - ◆ The maximum resolution is up to 1024X768 for TFT LCD panel for still image displaying
 - ◆ The maximum resolution is up to 480x272 for TFT LCD panel for MJPEG video displaying up to 25fps.
 - Display scaler – to fit different size of LCD panels
 - ◆ Horizontal: At most 4.0x scale
 - ◆ Vertical: At most 3.0x scale
 - For SYNC type LCD:
 - ◆ For 8-bit bus
 - CCIR601 YCbCr422 packet mode (NTSC/PAL) supported
 - CCIR601 RGB Dummy mode (NTSC/PAL) supported
 - CCIR656 interface supported
 - RGB Through mode supported

- ◆ For 16/18/24-bit bus
 - Parallel pixel data output mode (1-pixel/1-clock)
 - NTSC/PAL interlace & non-interlace output supported
 - Color format transform supported:
 - ◆ Color format transform between YCbCr422 and RGB565
 - ◆ Color format transform from YCbCr422 to RGB888
 - Support OSD function to overlap system information like battery life, brightness tuning, volume tuning or muting, etc.
- Frame Switch Controller
 - Frame relation controlled between VPOST and Capture supported
 - 2 modes supported to switch Frame Buffer Base
 - ◆ Frame Ratio Mode (16 selectable ratio)
 - ◆ Frame sync mode
 - Double/triple buffers supported
- SPU (Sound Processing Unit)
 - 32 stereo channels supported
 - PCM8/PCM16/4-bit MDPCM/TONE source format supported
 - 7-bit volume control supported for each of 32 channels
 - 5-bit pan control supported for each L/R of 32 channels
 - 10-band equalizer supported
 - Special code supported for loop playing and event detection
- Audio DAC
 - 16-bit stereo DAC supported with headphone driver output
 - H/W volume control supported
- I2S Controller
 - I2S interface supported to connect external audio codec
 - 16/18/20/24-bit data format supported
- Storage Interface Controller
 - Interface to NAND Flash:
 - ◆ 8-bit data bus width supported
 - ◆ SLC and MLC type NAND Flash supported
 - ◆ 512B, 2KB, 4KB, and 8KB page size NAND Flash supported
 - ◆ ECC4, ECC8, ECC12 and ECC15 algorithm supported for ECC generation, error detection and error correction
 - ◆ PBA-NAND flash supported
 - Interface to SD/MMC/SDIO/SDHC/micro-SD cards/ eMMC Flash device supported
 - ◆ SD-to-NAND flash bridge supported
 - DMA function supported to accelerate the data transfer between system memory and NAND Flash or SD/MMC/SDIO/SDHC/micro-SD/eMMC Flash device
- USB Device Controller
 - USB2.0 HS (High-Speed) x 1 port
 - 6 configurable endpoints supported
 - Control, Bulk, Interrupt and Isochronous transfers supported
 - Suspend and remote wakeup supported
- USB Host Controller
 - USB1.1 Host one H/W Engine, two pin locations.

- Fully compliant with USB Revision 1.1 specification
- Open Host Controller Interface (OHCI) Revision 1.0 compatible
- Full-speed (12Mbps) and low-speed (1.5Mbps) USB devices supported
- Control, Bulk, Interrupt and Isochronous transfers supported

- Timer & Watch-Dog Timer
 - Two 32-bit with 8-bit pre-scalar timers supported
 - One programmable 24-bit Watch-Dog Timer supported

- PWM
 - 4 PWM channel outputs supported
 - 16-bit counter supported for each PWM channel
 - Two 8-bit pre-scalars supported and each pre-scalar shared by two PWM channels
 - Two clock-dividers supported and each divider shared by two PWM channels
 - Two Dead-Zone generators supported and each generator shared by two PWM channels
 - Auto reloaded mode and one-shot pulse mode supported
 - Capture function supported

- UART
 - A high speed UART supported:
 - ◆ Baud rate is up to 1M bps
 - ◆ 4 signals TX, RX, CTS and RTS supported
 - A normal UART supported:
 - ◆ Baud rate is up to 115.2K bps
 - ◆ 2 signals TX and RX supported only

- SPI
 - Two SPI controller is supported
 - ◆ Both master and slave mode are supported in SPI interface
 - ◆ Two chip selection signals for two SPI devices

- I2C
 - One I2C channel supported
 - Compatible with Philips's I²C standard and only master mode supported
 - Multi-master operation supported

- Advanced Interrupt Controller
 - Total 32 interrupt source supported
 - Configurable interrupt type:
 - ◆ Low-active level triggered interrupt
 - ◆ High-active level triggered interrupt
 - ◆ Low-active edge (falling edge) triggered interrupt
 - ◆ High-active edge (rising edge) triggered interrupt
 - Individual interrupt mask bit for each interrupt source
 - 8 different priority levels supported
 - Daisy-chain priority mechanism supported for interrupts with same priority level
 - Low priority interrupt automatic masking supported for interrupt nesting

- RTC
 - Independent power plane supported
 - 32.768 KHz crystal oscillation circuit supported

- Time counter (second, minute, hour) and Calendar counter (day, month, year) supported
- Alarm supported (second, minute, hour, day, month and year)
- 12/24-hour mode and Leap year supported
- Alarm to wake chip up from Standby mode or from Power-down mode supported
- Wake chip up from Power-down mode by input pin supported
- Power-off chip by register setting supported
- Power-on timeout is supported for low battery protection

- GPIO
 - 70 programmable general purpose I/Os supported and separated into 5 groups
 - Individual configuration supported for each I/O signal
 - Configurable interrupt control functions supported
 - Configurable de-bounce circuit supported for interrupt function

- ADC
 - Multi-channel, 10-bit ADC supported
 - ◆ 2 channels dedicated for 4-wire resistive touch sensor inputs
 - ◆ 2 channels dedicated for Audio ADC with Microphone pre-Amp & AGC
 - ◆ 3 channels reserved for various purposes, like LVD (Low Voltage Detection), keypad input, and light sensor
 - ◆ Input voltage range from 0V ~ 3.3V supported
 - Maximum 25MHz input clock supported
 - Maximum 400K/s conversion rate supported
 - LVR (Low Voltage Reset) supported

- Power Management
 - Advanced power management including Power Down, Deep Standby, CPU Standby, and Normal Operating modes
 - ◆ Normal Operating Mode
 - Core power is 1.8V and chip is in normal operation
 - ◆ CPU Standby Mode
 - Core power is 1.8V and only ARM CPU clock is turned OFF
 - ◆ Deep Standby Mode
 - Core power is 1.8V and all IP clocks are turned OFF
 - ◆ Power Down Mode
 - Only the RTC power is ON. Other 3.3V and 1.8V power are OFF

- Operating Voltage
 - I/O: 3.3V
 - Core: 1.8V for 200MHz

- Package
 - LQFP-128 MCP

3 PARTS INFORMATION LIST AND PIN CONFIGURATION

3.1 N9H20 Series Part Number Naming Guide

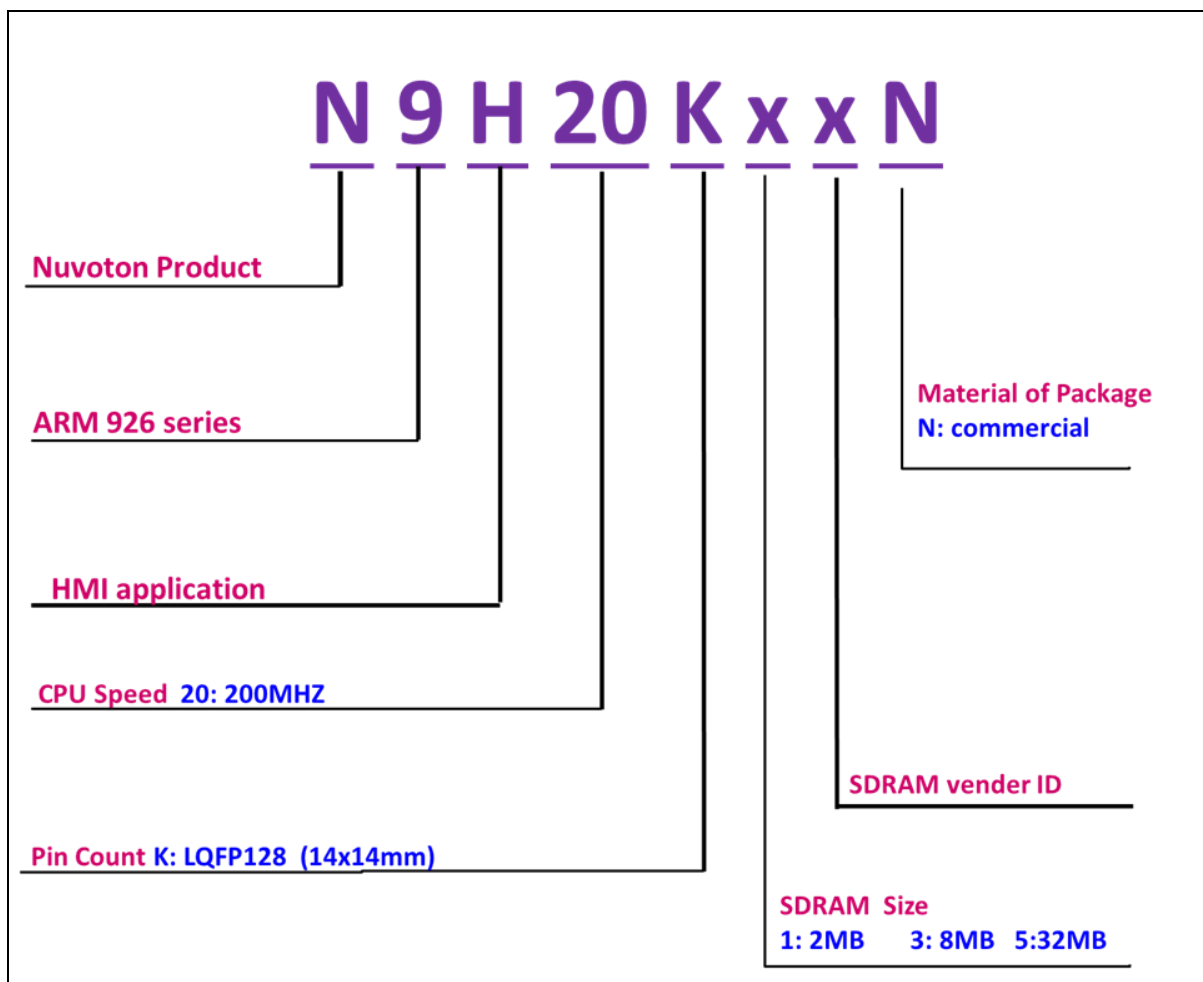


Figure 3.1-1 N9H20 Series Part Number Naming Guide

3.2 N9H20 Series Part Selection Guide

N9H20K series																																	
Part No.	Core				Memory				USB		H/W Accelerator		LCD		Analog		Peripheral							Power			PKG		Status ²				
	Max Speed (MHz)	ARM CPU	I Cache (KB)	D Cache (KB)	SRAM (KB)	Stacked DRAM	SPI Flash I/F	Raw NAND I/F, ECC bits	eMMC/SD / SDIO	1.1 Host (12 Mbps)	Device (FS / HS)	2D BitBLT GFX	JPEG Codec	Video Codec	RGB Color (bits)	Max. Resolution ¹	10/12-bit SAR ADC	4/5-wire TP ADC	Audio Stereo DAC	JTAG	GPIO (Max)	UART	I2C	SPI booting	SPI	RTC	PWM	I2S		Core Voltage (V)	DRAM Voltage (V)	I/O Voltage (V)	Package
N9H20K51N	200	926	8	8	8	32MB	√	15	3	1	HS	√	√	JPEG	24	XGA ³	10	4W	√	√	70	2	1	√	2	√	4	√	1.8	1.8	3.3	LQFP-128	MP
N9H20K31N	200	926	8	8	8	8MB	√	15	3	1	HS	√	√	JPEG	24	XGA ³	10	4W	√	√	70	2	1	√	2	√	4	√	1.8	1.8	3.3	LQFP-128	MP
N9H20K11N	200	926	8	8	8	2MB	√	15	3	1	HS	√	√	JPEG	24	QVGA	10	4W	√	√	70	2	1	√	2	√	4	√	1.8	3.3	3.3	LQFP-128	MP
¹ Resolution: QVGA (320x240), VGA (640x480), SVGA (800x600), XGA (1024 x 768).																																	
² Status: MP - Mass Production, ES - Engineering Sample, UD - Under Development.																																	
³ XGA is for still image only.																																	

3.3 Pin Configuration

3.3.1 N9H20 series Pin Diagram

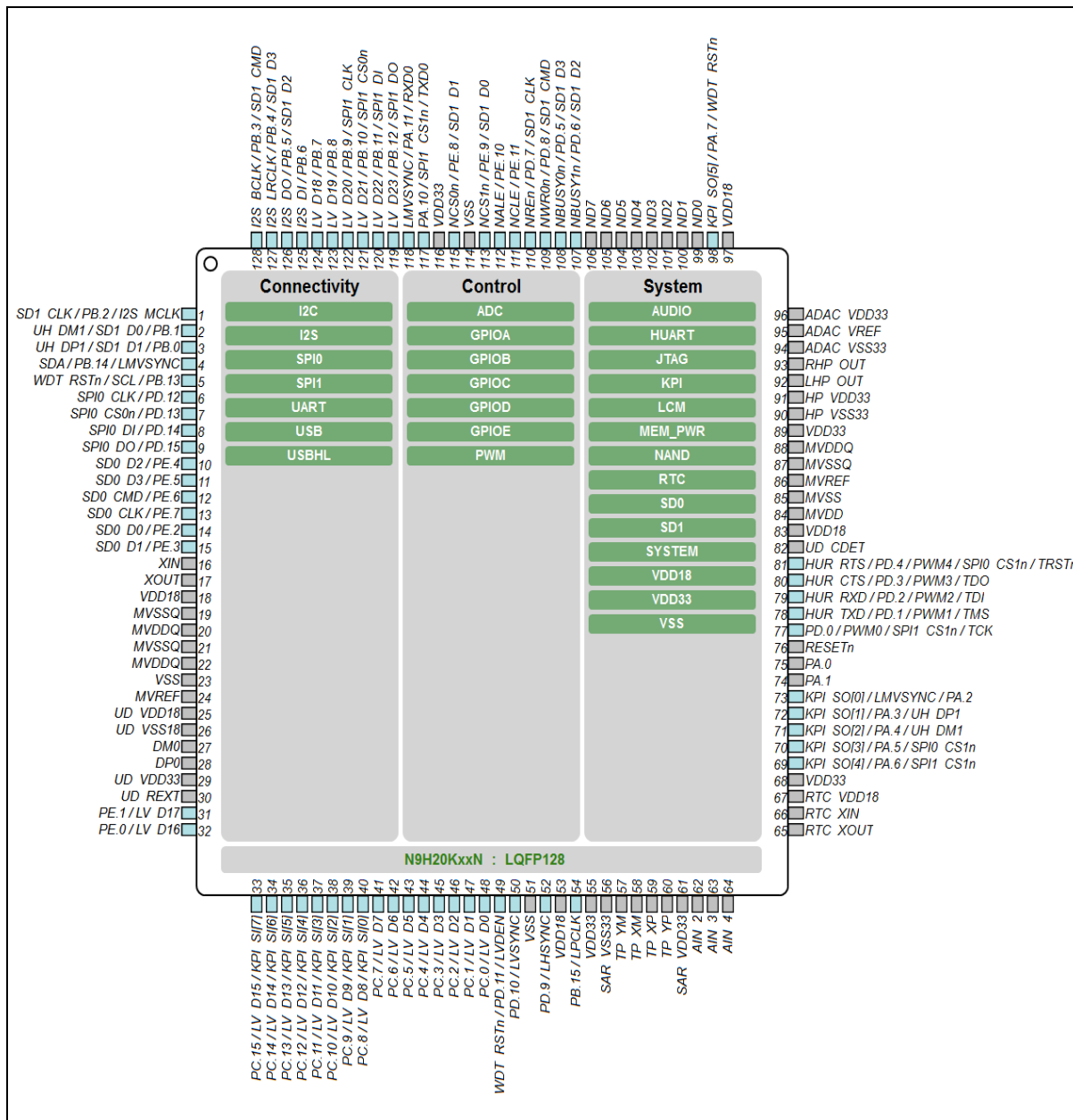


Figure 3.3-1 N9H20xxN LQFP 128 Pin Diagram

3.4 Pin Description

Pin No	Name	Type	Group	Description
1	GPB[2]	IOU	GPIOB	GPIO Port B Bit 2
	I2S_MCLK	OU	I ² S	Clock to I2S Codec, Output
	SD1_CLK	OU	SD1	SD Port 1 Clock, Output
2	GPB[1]	IOU	GPIOB	GPIO Port B Bit 1
	SD1_DAT[0]	IOU	SD1	SD Port 1 Data Bit 0
	UHL_DM1	IOU	USB	USB Host 1.0 Lite Port 1, D-E154
3	GPB[0]	IOU	JTAG	GPIO Port B Bit 0
	SD1_DAT[1]	IOU	SD1	SD Port 1 Data Bit 1
	UHL_DP1	IOU	USB	USB Host 1.0 Lite Port 1, D+
4	ISDA	IOU	I2C	I2C Interface Data
	GPB[14]	IOU	GPIOB	GPIO Port B Bit 14
	LFMARK	IU	LCD	MPU LCD Interface Frame Mark, Input
	LMVSYNC	OU	LCD	MPU LCD Interface Mode VSYNC, Output
5	ISCK	OU	I2C	I2C Interface Clock, Output
	GPB[13]	IOU	GPIOB	GPIO Port B Bit 13
6	SPI0_CLK	IOU	SPI0	SPI Port 0 Clock;Output in Master Mode;Input in Slave Mode
	GPD[12]	IOU	GPIOD	GPIO Port D Bit 12
7	SPI0_CS0_	IOU	SPI0	SPI Port 0 Device Select 0, Low Active;Output in Master Mode;Input in Slave Mode
	GPD[13]	IOU	GPIOD	GPIO Port D Bit 13
8	SPI0_DI	IU	SPI0	SPI Port 0 Data Input
	GPD[14]	IOU	GPIOD	GPIO Port D Bit 14
9	SPI0_DO	OU	SPI0	SPI Port 0 Data Output
	GPD[15]	IOU	GPIOD	GPIO Port D Bit 15
10	SD0_DAT[2]	IOU	SD0	SD Port 0 Data Bit 2
	GPE[4]	IOU	GPIOE	GPIO Port E Bit 4
11	SD0_DAT[3]	IOU	SD0	SD Port 0 Data Bit 3
	GPE[5]	IOU	GPIOE	GPIO Port E Bit 5
12	SD0_CMD	IOU	SD0	SD Port 0 Command/Response
	GPE[6]	IOU	GPIOE	GPIO Port E Bit 6
13	SD0_CLK	OU	SD0	SD Port 0 Clock, Output
	GPE[7]	IOU	GPIOE	GPIO Port E Bit 7
14	SD0_DAT[0]	IOU	SD0	SD Port 0 Data Bit 0

	GPE[2]	IOU	GPIOE	GPIO Port E Bit 2
15	SD0_DAT[1]	IOU	SD0	SD Port 0 Data Bit 1
	GPE[3]	IOU	GPIOE	GPIO Port E Bit 3
16	XIN	I	XTAL	12MHz Crystal Input
17	XOUT	O	XTAL	12MHz Crystal Output
18	VDD18	P	Core	Core Logic Power (1.8V)
19	VSSQ	G	MVDD	SDRAM I/F Ground (0V)
20	VDDQ	P	MVDD	SDRAM I/F Power
21	VSSQ	G	MVDD	SDRAM Ground (0V)
22	MVDD	P	MVDD	SDRAM Power
23	VSS	G	GND	Ground (0V)
24	VSS	G	GND	Ground (0V), when NUC9H20K11N is installation
	MVREF	P	MVDD	1/2 MVDD (0.9V) for DDR_VREF, when NUC9H20K31N is installation
25	PLL_UD_VDD18	P	PLL & USB	PLL & USB2.0 Device Core Power (1.8V)
26	UD_VSS	G	USB	USB2.0 Device Ground (0V)
27	UD_DM	I/O	USB	USB 2.0 Device D-.
28	UD_DP	I/O	USB	USB 2.0 Device D+.
29	UD_VDD33	P	USB	USB 2.0 Device PHY 3.3V
30	UD_REXT	O	USB	External Resistor 12.1K resistor connected to ground
31	LVDATA[17]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 17
	GPE[1]	IOU	GPIOE	GPIO Port E Bit 1
32	LVDATA[16]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 16
	GPE[0]	IOU	GPIOE	GPIO Port E Bit 0
33	LVDATA[15]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 15
	GPC[15]	IOU	GPIOC	GPIO Port C Bit 15
	KPI_SI[7]	IU	KPI	KPI Scan In Bit 7
34	LVDATA[14]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 14
	GPC[14]	IOU	GPIOC	GPIO Port C Bit 14
	KPI_SI[6]	IU	KPI	KPI Scan In Bit 6
35	LVDATA[13]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 13
	GPC[13]	IOU	GPIOC	GPIO Port C Bit 13
	KPI_SI[5]	IU	KPI	KPI Scan In Bit 5
36	LVDATA[12]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 12
	GPC[12]	IOU	GPIOC	GPIO Port C Bit 12

	KPI_SI[4]	IU	KPI	KPI Scan In Bit 4
37	LVDATA[11]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 11
	GPC[11]	IOU	GPIOC	GPIO Port C Bit 11
	KPI_SI[3]	IU	KPI	KPI Scan In Bit 3
38	LVDATA[10]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 10
	GPC[10]	IOU	GPIOC	GPIO Port C Bit 10
	KPI_SI[2]	IU	KPI	KPI Scan In Bit 2
39	LVDATA[9]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 9
	GPC[9]	IOU	GPIOC	GPIO Port C Bit 9
	KPI_SI[1]	IU	KPI	KPI Scan In Bit 1
40	LVDATA[8]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 8
	GPC[8]	IOU	GPIOC	GPIO Port C Bit 8
	KPI_SI[0]	IU	KPI	KPI Scan In Bit 0
41	LVDATA[7]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 7
	GPC[7]	IOU	GPIOC	GPIO Port C Bit 7
42	LVDATA[6]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 6
	GPC[6]	IOU	GPIOC	GPIO Port C Bit 6
	CFG[10]	IU	SYSTEM	Chip Power-On Configuration Bit [10], Input
43	LVDATA[5]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 5
	GPC[5]	IOU	GPIOC	GPIO Port C Bit 5
	CFG[9]	IU	SYSTEM	Chip Power-On Configuration Bit [9], Input
44	LVDATA[4]	IOU	LCD	GPIO Port C Bit 4
	GPC[4]	IOU	GPIOC	GPIO Port C Bit 4
	CFG[8]	IU	SYSTEM	Chip Power-On Configuration Bit [8], Input
45	LVDATA[3]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 3
	GPC[3]	IOU	GPIOC	GPIO Port C Bit 3
46	LVDATA[2]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 2
	GPC[2]	IOU	GPIOC	GPIO Port C Bit 2
47	LVDATA[1]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 1
	GPC[1]	IOU	GPIOC	GPIO Port C Bit 1
48	LVDATA[0]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 0
	GPC[0]	IOU	GPIOC	GPIO Port C Bit 0
49	LVDE	OU	LCD	SYNC LCD Interface Data Enable, Output, High Active
	LRS	OU	LCD	MPU LCD Interface Register Select
	GPD[11]	IOU	GPIOD	GPIO Port D Bit 11

				Note. By design limitation, GPD[11] function will be disable when SPI1 engine is used.
50	LVSYNC	OU	LCD	SYNC LCD Interface VSYNC, Output, High Active
	LRD	OU	LCD	MPU I80 mode LCD Interface Read, active low
	LEN	OU	LCD	MPU M68 mode LCD Interface Read Write Enable/Disable , High Enable
	GPD[10]	IOU	GPIOD	LCD Interface Data Enable, High Active.
51	VSS	G	GND	Ground (0V)
52	LHSYNC	OU	LCD	SYNC LCD Interface HSYNC, Output, High Active
	LWR	OU	LCD	MPU I80 mode LCD Interfacer Write, active low
	LR/W	OU	LCD	MPU M68 mode LCD interface R/W, High is read command and Low is write instruction
	GPD[9]	IOU	GPIOD	GPIO Port D Bit 9
53	VDD18	P	Core	Core Logic Power (1.8V)
54	LPCLK	OU	LCD	SYNC LCD Interface Pixel Clock, Output
	LCS	OU	LCD	MPU LCD Interface Chip Enable, active low
	GPB[15]	IOU	GPIOB	GPIO Port B Bit 15
55	VDD33	P	I/O	I/O Power (3.3V)
56	ADC_VSS33	G	ADC	ADC Ground (0V)
57	ADC_TP_YM	I	ADC	Touch Panel YM
58	ADC_TP_XM	I	ADC	Touch Panel XM
59	ADC_TP_XP	I	ADC	Touch Panel XP
60	ADC_TP_YP	I	ADC	Touch Panel YP
61	ADC_VDD33	P	ADC	ADC Power (3.3V)
62	ADC_AIN[2]	I	ADC	ADC Analog Input Channel 2
63	ADC_AIN[3]	I	ADC	ADC Analog Input Channel 3
64	ADC_AIN[4]	I	ADC	ADC Analog Input Channel 4
65	RTC_XOUT	O	RTC	32768Hz Crystal Output
66	RTC_XIN	I	RTC	32768Hz Crystal Input
67	RTC_VDD1.8V	P	RTC	RTC Power (1.8V)
68	VDD33	P	I/O	I/O Power (3.3V)
69	GPA[6]	IOU	GPIOA	GPIO Port A Bit 6
	SPI1_CS1_	OU	SPI1	SPI Port 1 Device Select 1, Output, Low Active
	KPI_SO[4]	OU	KPI	KPI Scan Out Bit 4
70	GPA[5]	IOU	GPIOA	GPIO Port A Bit 5

	SPI0_CS1_	OU	SPI0	SPI Port 0 Device Select 1, Output, Low Active
	KPI_SO[3]	OU	KPI	KPI Scan Out Bit 3
71	GPA[4]	IOU	GPIOA	GPIO Port A Bit 4
	UHL_DM1	IOU	USB	USB Host 1.0 Lite Port 1, D-
	KPI_SO[2]	OU	KPI	KPI Scan Out Bit 2
72	GPA[3]	IOU	GPIOA	GPIO Port A Bit 3
	UHL_DP1	IOU	USB	USB Host 1.0 Lite Port 1, D+
	KPI_SO[1]	OU	KPI	KPI Scan Out Bit 1
73	GPA[2]	IOU	GPIOA	GPIO Port A Bit 2
	LMVSYNC	OU	LCD	MPU LCD Interface VSYNC, Output
	LFMARK	IU	LCD	MPU LCD Interface Frame Mark, Input
	KPI_SO[0]	OU	KPI	KPI Scan Out Bit 0
74	GPA[1]	IOU	GPIOA	GPIO Port A Bit 1
	SD_CD_	IU	SD	SD Card Detect, Input, Low Active
75	GPA[0]	IOU	GPIOA	GPIO Port A Bit 0
76	RST_	IU	SYSTEM	System Reset, Input, Low Active
	WDT_RST_	OU	SYSTEM	Watch-Dog Reset, Output, Low Active
77	TCK	ID	JTAG	JTAG Interface Test Clock, Input
	GPD[0]	IOD	GPIOD	GPIO Port D Bit 0
	SPI1_CS1_	OD	SPI1	SPI Port 1 Device Select 1, Output, Low Active
	PWM0	OD	PWM	PWM Channel 0
78	TMS	IU	JTAG	JTAG Interface Test Mode Select, Input
	GPD[1]	IOU	GPIOD	GPIO Port D Bit 1
	HUR_TXD	OU	UART	High-Speed UART TX Data, Output
	PWM1	OU	PWM	PWM Channel 1
79	TDI	IU	JTAG	JTAG Interface Test Data In, Input
	GPD[2]	IOU	GPIOD	GPIO Port D Bit 2
	HUR_RXD	IU	UART	High-Speed UART RX Data, Input
	PWM2	OU	PWM	PWM Channel 2
80	TDO	OU	JTAG	JTAG Interface Test Data Out, Output
	GPD[3]	IOU	GPIOD	GPIO Port D Bit 3
	HUR_CTS	IU	UART	High-Speed UART Clear-To-Send, Input, Low Active
	PWM3	OU	PWM	PWM Channel 3
81	TRST_	IU	JTAG	JTAG Interface Test Reset, Input, Low Active
	GPD[4]	IOU	GPIOD	GPIO Port D Bit 4

	HUR_RTS	OU	UART	High-Speed UART Reset-To-Send, Output, Low Active
	SPI0_CS1_	OU	SPI0	SPI Port 0 Device Select 1, Output, Low Active
82	UD_CDET	I	USB	USB Device Connect Detect, Input, High Active
83	VDD18	P	Core	Core Logic Power (1.8V)
84	MVDD	P	MVDD	SDRAM I/O Power
85	MVSS	G	MVDD	SDRAM I/F Ground (0V)
86	MVREF	P	MVDD	1/2 MVDD (0.9V) for DDR_VREF, when NUC9H20K31N is installation
	VSS	G	GND	Ground (0V), when NUC9H20K11N is installation
87	VSSQ	G	MVDD	SDRAM Ground (0V)
88	VDDQ	P	MVDD	SDRAM Power
89	VDD33	P	I/O	I/O Power (3.3V)
90	ADAC_HPVS33	G	AUDIO	Audio DAC Headphone Driver Ground (0V)
91	ADAC_HPVD33	P	AUDIO	Audio DAC Headphone Driver Power (3.3V)
92	ADAC_HPOUT_L	O	AUDIO	Audio Headphone Left Channel Output
93	ADAC_HPOUT_R	O	AUDIO	Audio Headphone Right Channel Output
94	ADAC_AVSS33	G	AUDIO	Audio DAC Ground (0V)
95	ADAC_VREF	O	AUDIO	Audio DAC Reference Voltage Output, please connect 1uF capacitor to DAC ground
96	ADAC_AVDD33	P	AUDIO	Audio DAC Power (3.3V)
97	VDD18	P	Core	Core Logic Power (1.8V)
98	GPA[7]	IOU	GPIOA	GPIO Port A Bit 7
	KPI_SO[5]	OU	KPI	KPI Scan Out Bit 5
99	ND[0]	IOU	NAND	NAND Interface Data Bit [0]
	CFG[0]	IU	SYSTEM	Chip Power-On Configuration Bit [0], Input
100	ND[1]	IOU	NAND	NAND Interface Data Bit [1]
	CFG[1]	IU	SYSTEM	Chip Power-On Configuration Bit [1], Input
101	ND[2]	IOU	NAND	NAND Interface Data Bit [2]
	CFG[2]	IU	SYSTEM	Chip Power-On Configuration Bit [2], Input
102	ND[3]	IOU	NAND	NAND Interface Data Bit [3]
	CFG[3]	IU	SYSTEM	Chip Power-On Configuration Bit [3], Input
103	ND[4]	IOU	NAND	NAND Interface Data Bit [4]
	CFG[4]	IU	SYSTEM	Chip Power-On Configuration Bit [4], Input
104	ND[5]	IOU	NAND	NAND Interface Data Bit [5]
	CFG[5]	IU	SYSTEM	Chip Power-On Configuration Bit [5], Input
105	ND[6]	IOU	NAND	NAND Interface Data Bit [6]

	CFG[6]	IU	SYSTEM	Chip Power-On Configuration Bit [6], Input
106	ND[7]	IOU	NAND	NAND Interface Data Bit [7]
	CFG[7]	IU	SYSTEM	Chip Power-On Configuration Bit [7], Input
107	NBUSY1_	IU	NAND	NAND Interface Busy 1, Input, Low Active
	GPD[6]	IOU	GPIO	GPIO Port D Bit 6
	SD2_DAT[2]	IOU	SD2	SD Port 2 Data Bit 2
108	NBUSY0_	IU	NAND	NAND Interface Busy 0, Input, Low Active
	GPD[5]	IOU	GPIO	GPIO Port D Bit 5
	SD2_DAT[3]	IOU	SD2	SD Port 2 Data Bit 3
109	NWR_	OU	NAND	NAND Interface Write Enable, Output, Low Active
	GPD[8]	IOU	GPIO	GPIO Port D Bit 8
	SD2_CMD	IOU	SD2	SD Port 2 Command/Response
110	NRE_	OU	NAND	NAND Interface Read Enable, Output, Low Active
	GPD[7]	IOU	GPIO	GPIO Port D Bit 7
	SD2_CLK	OU	SD2	SD Port 2 Clock, Output
111	NCLE	OU	NAND	NAND Interface Command-Latch-Enable, Output, High Active
	GPE[11]	IOU	GPIOE	GPIO Port E Bit 11
112	NALE	OU	NAND	NAND Interface Address-Latch-Enable, Output, High Active
	GPE[10]	IOU	GPIOE	GPIO Port E Bit 10
113	NCS1_	OU	NAND	NAND Interface Chip Select 1, Output, Low Active
	GPE[9]	IOU	GPIOE	GPIO Port E Bit 9
	SD2_DAT[0]	IOU	SD2	SD Port 2 Data Bit 0
114	VSS	G	GND	Ground (0V)
115	NCS0_	IU	NAND	NAND Interface Chip Select 0, Output, Low Active
	GPE[8]	IOU	GPIOE	GPIO Port E Bit 8
	SD2_DAT[1]	IOU	SD2	SD Port 2 Data Bit 1
116	VDD33	P	I/O	I/O Power (3.3V)
117	URTXD	OU	UART	UART TX Data, Output
	GPA[10]	IOU	GPIOA	GPIO Port A Bit 10
	SPI1_CS1_	OU	SPI1	SPI Port 1 Device Select 1, Output, Low Active
118	URRXD	IU	UART	UART RX Data, Input
	GPA[11]	IOU	GPIOA	GPIO Port A Bit 11
	LMVSYNC	OU	LCD	MPU LCD Interface VSYNC, Output
	LFMARK	IU	LCD	MPU LCD Interface Frame Mark, Input
119	GPB[12]	IOU	GPIOB	GPIO Port B Bit 12

	SPI1_DO	OU	SPI1	SPI Port 1 Data Output
	LVDATA[23]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 23
120	GPB[11]	IOU	GPIOB	GPIO Port B Bit 11
	SPI1_DI	IU	SPI1	SPI Port 1 Data Input
	LVDATA[22]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 22
121	GPB[10]	IOU	GPIOB	GPIO Port B Bit 10
	SPI1_CS0_	IOU	SPI1	SPI Port 1 Device Select 0, Low Active;Output in Master Mode;Input in Slave Mode
	LVDATA[21]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 21
122	GPB[9]	IOU	GPIOB	GPIO Port B Bit 9
	SPI1_CLK	IOU	SPI1	SPI Port 1 Clock;Output in Master Mode;Input in Slave Mode
	LVDATA[20]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 20
123	GPB[8]	IOU	GPIOB	GPIO Port B Bit 8
	LVDATA[19]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 19
124	GPB[7]	IOU	GPIOB	GPIO Port B Bit 7
	LVDATA[18]	IOU	LCD	SYNC/MPU LCD Interface Data Bit 18
125	GPB[6]	IOU	GPIOB	GPIO Port B Bit 6
	I2S_DIN	IU	I ² S	I2S Interface Data Input
126	GPB[5]	IOU	GPIOB	GPIO Port B Bit 5
	I2S_DOUT	OU	I2S	I2S Interface Data Output
	SD1_DAT[2]	IOU	SD1	SD Port 1 Data Bit 2
127	GPB[4]	IOU	GPIOB	GPIO Port B Bit 4
	I2S_WS	OU	I ² S	I2S Interface Word Select, Output
	SD1_DAT[3]	IOU	SD1	SD Port 1 Data Bit 3
128	GPB[3]	IOU	GPIOB	GPIO Port B Bit 3
	I2S_BCLK	IU	I ² S	I2S Interface Clock, Input
	SD1_CMD	IOU	SD1	SD Port 1 Command/Response

Note:

TYPE	DESCRIPTION
I	Input
O	Output
I/O	Input / Output
IU	Input with internal pull high (50K)
OU	Output with internal pull high (50K)
IOU	Bi-direction with internal pull high (50K)
ID	Input with internal pull low (50K)
OD	Output with internal pull low (50K)
IOD	Bi-direction with internal pull low (50K)
P	Analog Power or Digital Power
G	Analog GND or Digital GND

4 BLOCK DIAGRAM

4.1 N9H20 Series Block Diagram

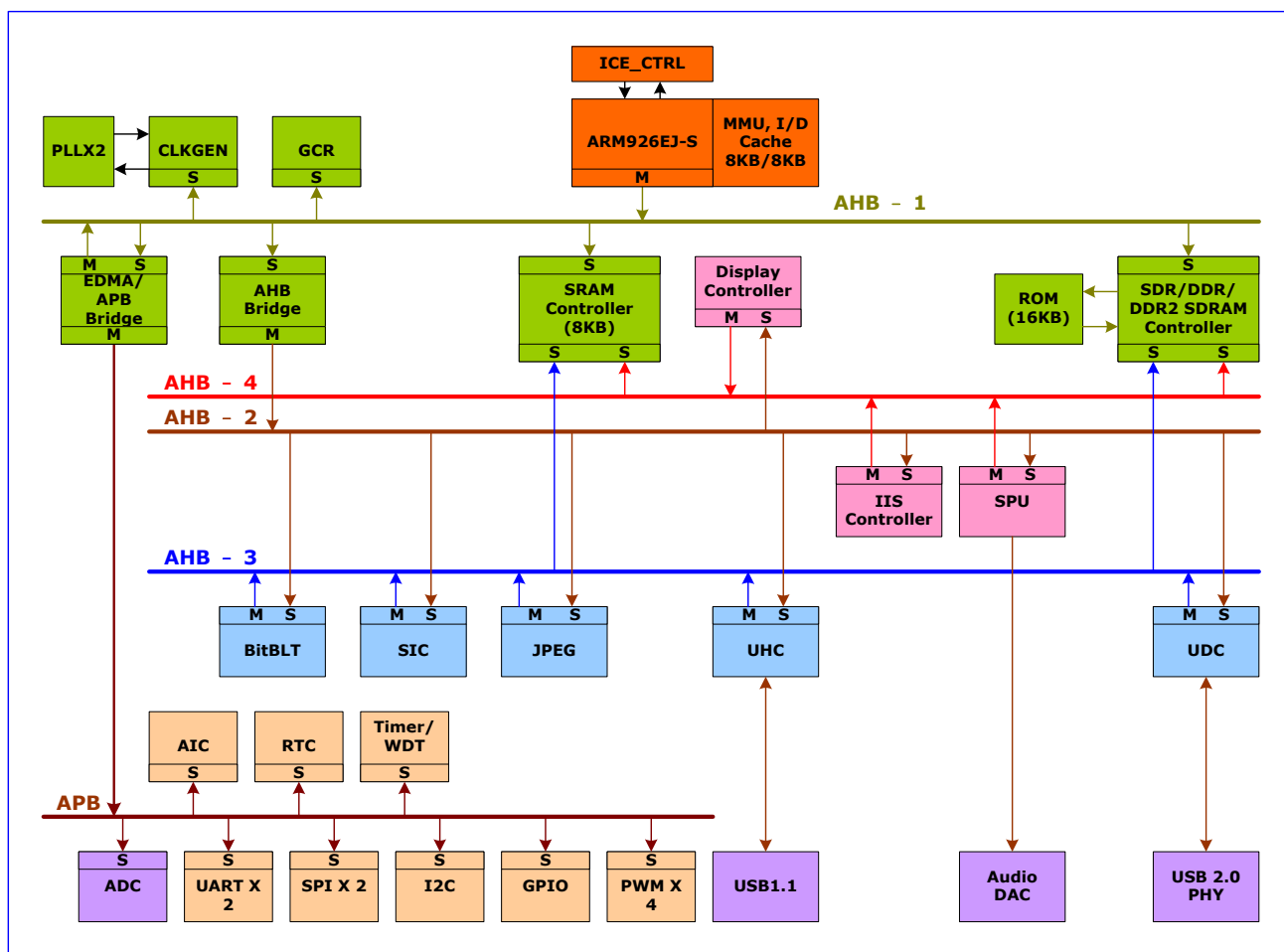


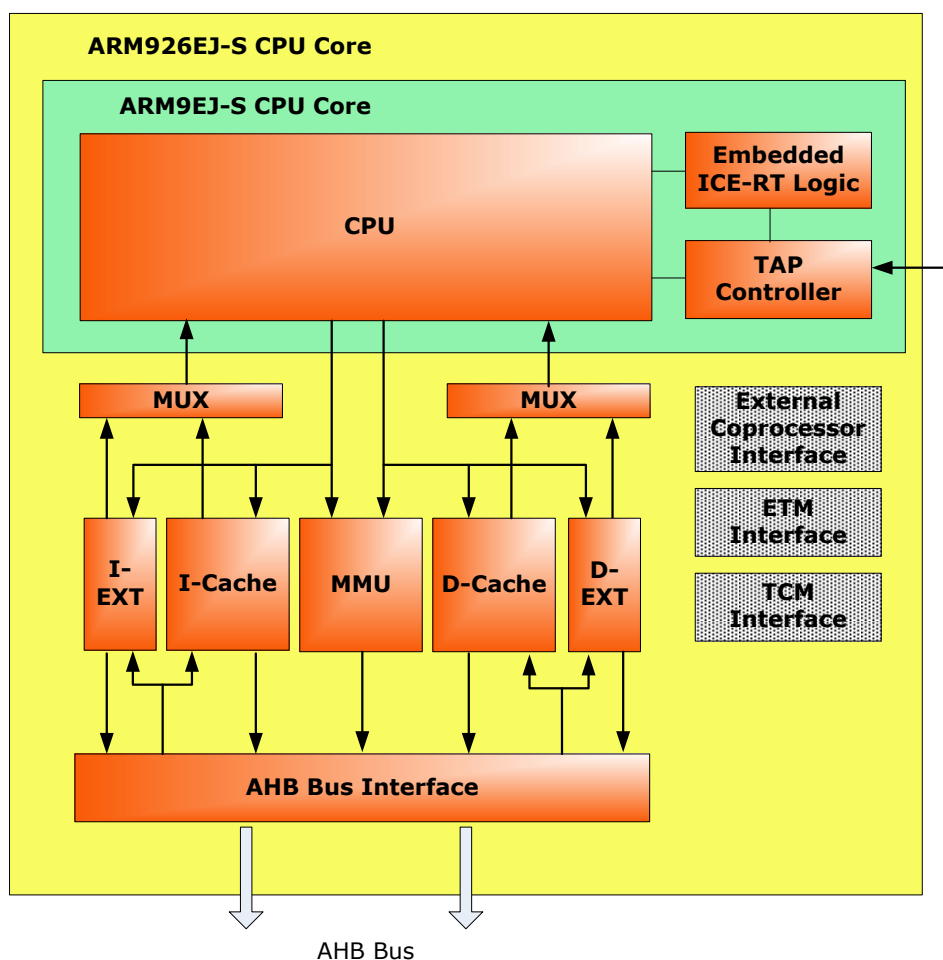
Figure 4.1-1 N9H20 Series Block Diagram

5 FUNCTIONAL DESCRIPTION

5.1 ARM® ARM926EJ-S CPU Core

The ARM926EJ-S CPU core is a member of the ARM9 family of general-purpose microprocessors. The ARM926EJ-S CPU core is targeted at multi-tasking applications where full memory management, high performance, low die size, and low power are all important. The ARM926EJ-S CPU core supports the 32-bit ARM and 16-bit Thumb instruction sets, enabling the user to trade off between high performance and high code density. The ARM926EJ-S CPU core includes features for efficient execution of Java byte codes, providing Java performance similar to JIT, but without the associated code overhead. The ARM926EJ-S processor has a Harvard cached architecture and provides a complete high-performance processor subsystem, including:

- An ARM9EJ-S integer core
- A Memory Management Unit (MMU)
- Separate instruction and data AMBA AHB bus interfaces



5.2 System Manager

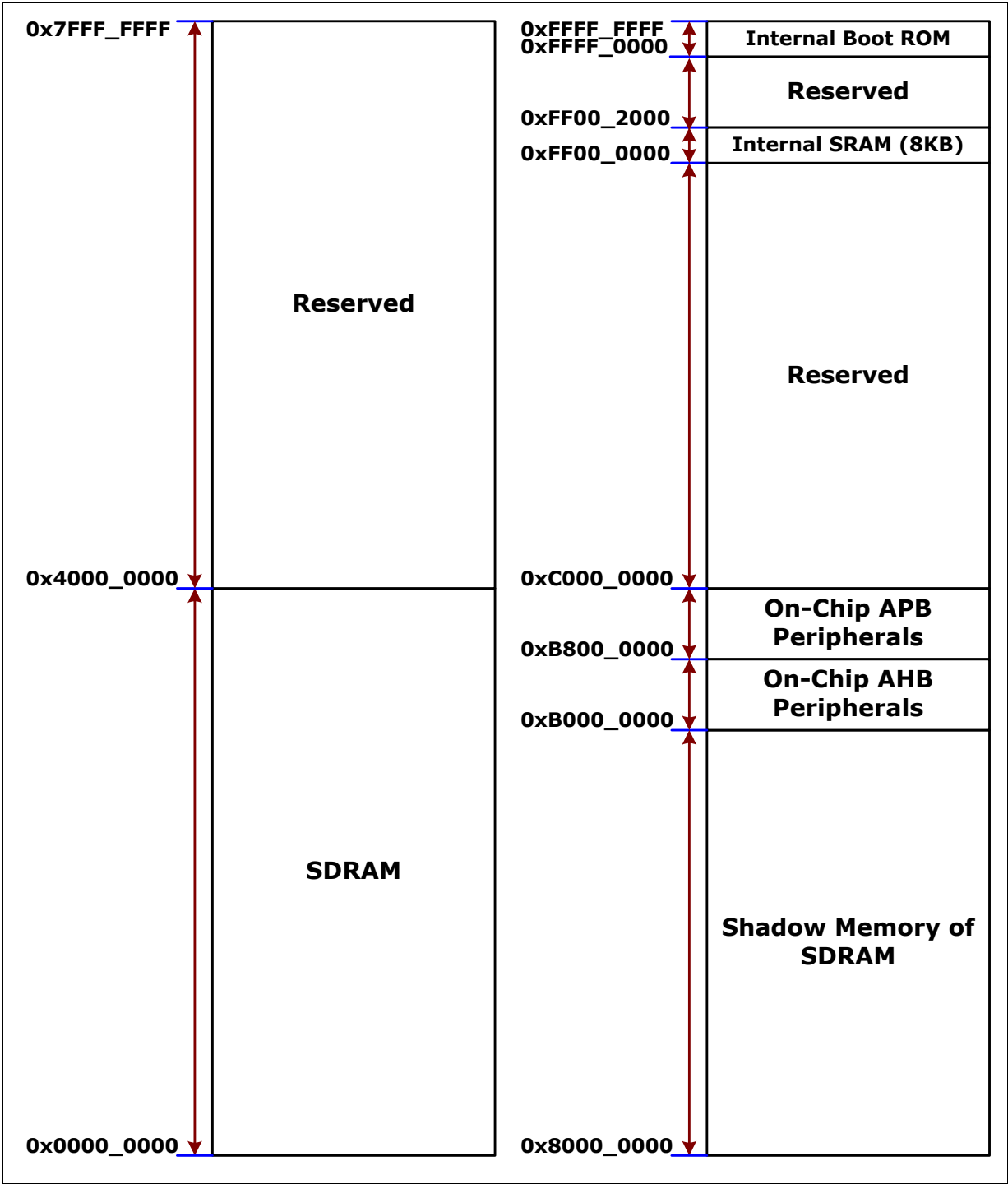
5.2.1 Overview

The following functions are included in system manager section

- System Memory Map
- Power-On Setting
- Bus Arbitration Mode
- Power Management
- IBR (Internal Boot ROM) Sequence
- System management registers for product ID, functional reset and multi-function pin control.

5.2.2 System Memory Map

This chip provides a 4G-byte address space for programmers. The memory locations assigned to each on-chip modules are shown as follow. The detailed register and memory addressing and programming will be described in the following sections for individual on-chip modules.



Memory Space	Base Address	Alias	Descriptions
0x0000_0000 – 0x7FFF_FFFF		SDRAM_BA	SDRAM Memory Space
0x8000_0000 – 0xAFFF_FFFF			Shadow Space for SDRAM
0xB000_0000 – 0xB000_01FF	0xB000_0000	GCR_BA	System and Global Control Registers
0xB000_0200 – 0xB000_02FF	0xB000_0200	CLK_BA	Clock Control Registers
0xB000_1000 – 0xB000_1FFF	0xB000_1000		Reserved
0xB000_2000 – 0xB000_2FFF	0xB000_2000		Reserved
0xB000_3000 – 0xB000_3FFF	0xB000_3000	SDIC_BA	SDRAM Interface Control Registers
0xB000_4000 – 0xB000_4FFF	0xB000_4000		Reserved
0xB000_5000 – 0xB000_5FFF	0xB000_5000		Reserved
0xB000_8000 – 0xB000_8FFF	0xB000_8000	EDMA_BA	EDMA Control Registers
0xB100_0000 – 0xB100_0FFF	0xB100_0000	SPU_BA	SPU Control Registers
0xB100_1000 – 0xB100_1FFF	0xB100_1000	I2S_BA	I2S Control Registers
0xB100_2000 – 0xB100_2FFF	0xB100_2000	LCD_BA	VPOST (Display) Control Registers
0xB100_3000 – 0xB100_3FFF	0xB100_3000	CAP_BA	Video-In (Capture) Control Registers
0xB100_4000 – 0xB100_4FFF	0xB100_4000		Reserved
0xB100_5000 – 0xB100_5FFF	0xB100_5000	FSC_BA	Frame Switch Control Registers
0xB100_6000 – 0xB100_6FFF	0xB100_6000	SIC_BA	SIC Control Registers
0xB100_8000 – 0xB100_8FFF	0xB100_8000	UDC_BA	USB Device Control Registers
0xB100_9000 – 0xB100_9FFF	0xB100_9000	UHC_BA	USB Host Control Registers
0xB100_A000 – 0xB100_AFFF	0xB100_A000	JPG_BA	JPEG Codec Control Registers
0xB100_C000 – 0xB100_CFFF	0xB100_C000		Reserved
0xB100_D000 – 0xB100_DFFF	0xB100_D000	BLT_BA	BitBLT Accelerator Control Register
0xB800_0000 – 0xB800_0FFF	0xB800_0000	AIC_BA	AIC Control Registers
0xB800_1000 – 0xB800_1FFF	0xB800_1000	GP_BA	GPIO Control Registers
0xB800_2000 – 0xB800_2FFF	0xB800_2000	TMR_BA	Timer/WDT Control Registers
0xB800_3000 – 0xB800_3FFF	0xB800_3000	RTC_BA	RTC Control Registers
0xB800_4000 – 0xB800_4FFF	0xB800_4000	I2C_BA	I2C Control Registers
0xB800_5000 – 0xB800_5FFF	0xB800_5000	KPI_BA	KPI Control Registers
0xB800_7000 – 0xB800_7FFF	0xB800_7000	PWM_BA	PWM Control Registers
0xB800_8000 – 0xB800_8FFF	0xB800_8000	UA_BA	UART Control Registers
0xB800_C000 – 0xB800_C3FF	0xB800_C000	SPIMS0_BA	SPIMS 0 Control Registers
0xB800_C400 – 0xB800_C7FF	0xB800_C400	SPIMS1_BA	SPIMS 1 Control Registers
0xB800_E000 – 0xB800_EFFF	0xB800_E000	ADC_BA	ADC Control Registers
0xFF00_0000 – 0xFF00_1FFF	0xFF00_0000	SRAM_BA	SRAM Memory Space (8KB)

0xFFFF_0000 – 0xFFFF_FFFF	0xFFFF_0000	IBR_BA	Internal Boot ROM Memory Space

5.2.3 Power-On Settings

The power-on setting value is used to configure the chip to enter a specific state after power-up or reset. The power-on setting value will be kept in power-on setting control register for reference.

Pin Name	Descriptions	Register Bit Mapping										
LVDATA[6]	NAND Flash Type Selection This power-on setting value define the which NAND flash type is used for NAND booting. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0</td><td>EF (Error Free) NAND flash memory</td></tr><tr><td>0x1</td><td>Raw NAND flash memory</td></tr></table>	Value	Description	0x0	EF (Error Free) NAND flash memory	0x1	Raw NAND flash memory	CHIPCFG[10]				
Value	Description											
0x0	EF (Error Free) NAND flash memory											
0x1	Raw NAND flash memory											
LVDATA[5:4]	NAND Page Size This power-on setting value define the page size of NAND for NAND booting. <table><tr><th>LVDATA[5:4]</th><th>Description</th></tr><tr><td>0x00</td><td>NAND page size is 2KB</td></tr><tr><td>0x01</td><td>NAND page size is 4KB</td></tr><tr><td>0x10</td><td>NAND page size is 8KB</td></tr><tr><td>0x11</td><td>Ignore NAND power-on setting</td></tr></table>	LVDATA[5:4]	Description	0x00	NAND page size is 2KB	0x01	NAND page size is 4KB	0x10	NAND page size is 8KB	0x11	Ignore NAND power-on setting	CHIPCFG[9:8]
LVDATA[5:4]	Description											
0x00	NAND page size is 2KB											
0x01	NAND page size is 4KB											
0x10	NAND page size is 8KB											
0x11	Ignore NAND power-on setting											
ND[7]	NAND Command Cycle This 1-bit power-on setting value is to define the NAND address cycle count for NAND booting. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0</td><td>The NAND address cycle is 4.</td></tr><tr><td>0x1</td><td>The NAND address cycle is 5.</td></tr></table>	Value	Description	0x0	The NAND address cycle is 4.	0x1	The NAND address cycle is 5.	CHIPCFG[7]				
Value	Description											
0x0	The NAND address cycle is 4.											
0x1	The NAND address cycle is 5.											
ND[6]	Reserved	CHIPCFG[6]										

ND[5:4]	SDRAM Type Selection This 2-bit power-on setting value is to indicate which type of SDRAM is used for system memory. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x00</td><td>SDR SDRAM (normal SDRAM) is used.</td></tr><tr><td>0x01</td><td>Low Power DDR (mDDR) SDRAM is used.</td></tr><tr><td>0x10</td><td>DDR2 SDRAM is used.</td></tr><tr><td>0x11</td><td>DDR SDRAM is used.</td></tr></table>	Value	Description	0x00	SDR SDRAM (normal SDRAM) is used.	0x01	Low Power DDR (mDDR) SDRAM is used.	0x10	DDR2 SDRAM is used.	0x11	DDR SDRAM is used.	CHIPCFG[5:4]																								
Value	Description																																			
0x00	SDR SDRAM (normal SDRAM) is used.																																			
0x01	Low Power DDR (mDDR) SDRAM is used.																																			
0x10	DDR2 SDRAM is used.																																			
0x11	DDR SDRAM is used.																																			
ND[3:0]	Chip Operation Mode Selection This 4-bit power-on setting value is to configure the chip to enter the normal operation. In normal operation mode, this field will configure the chip to boot from (IBR) Internal Boot ROM. In addition, power-on setting also used to select the UART debug port message disable or enable. The following is the truth table for different chip operation mode described above: <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0000</td><td>Reserved</td></tr><tr><td>0x0001</td><td>Reserved</td></tr><tr><td>0x0010</td><td>Reserved</td></tr><tr><td>0x0011</td><td>Reserved</td></tr><tr><td>0x0100</td><td>Reserved</td></tr><tr><td>0x0101</td><td>Reserved</td></tr><tr><td>0x0110</td><td>Reserved</td></tr><tr><td>0x0111</td><td>Reserved</td></tr><tr><td>0x1000</td><td>Reserved</td></tr><tr><td>0x1001</td><td>Reserved</td></tr><tr><td>0x1010</td><td>Boot at USB Recover Mode.</td></tr><tr><td>0x1011</td><td>Boot at Normal Mode.</td></tr><tr><td>0x1100</td><td>Reserved.</td></tr><tr><td>0x1101</td><td>Reserved.</td></tr><tr><td>0x1110</td><td>Boot at USB Recover Mode & UART debug port message disable.</td></tr><tr><td>0x1111</td><td>Boot at Normal Mode & UART debug port message disable.</td></tr></table> Please note if any of the reserved setting is used, CHIP's behavior is undefined.	Value	Description	0x0000	Reserved	0x0001	Reserved	0x0010	Reserved	0x0011	Reserved	0x0100	Reserved	0x0101	Reserved	0x0110	Reserved	0x0111	Reserved	0x1000	Reserved	0x1001	Reserved	0x1010	Boot at USB Recover Mode.	0x1011	Boot at Normal Mode.	0x1100	Reserved.	0x1101	Reserved.	0x1110	Boot at USB Recover Mode & UART debug port message disable.	0x1111	Boot at Normal Mode & UART debug port message disable.	CHIPCFG[3:0]
Value	Description																																			
0x0000	Reserved																																			
0x0001	Reserved																																			
0x0010	Reserved																																			
0x0011	Reserved																																			
0x0100	Reserved																																			
0x0101	Reserved																																			
0x0110	Reserved																																			
0x0111	Reserved																																			
0x1000	Reserved																																			
0x1001	Reserved																																			
0x1010	Boot at USB Recover Mode.																																			
0x1011	Boot at Normal Mode.																																			
0x1100	Reserved.																																			
0x1101	Reserved.																																			
0x1110	Boot at USB Recover Mode & UART debug port message disable.																																			
0x1111	Boot at Normal Mode & UART debug port message disable.																																			

5.2.4 Bus Arbitration Mode

The internal bus of this chip is the share bus architecture and a bus arbiter is implemented to decide which bus master could get the ownership of this share bus. The bus arbiter provides a choice of two arbitration algorithms for simultaneous requests. These two arbitration algorithms are the *fixed-priority mode* and the *round-robin-priority (rotate) mode*. The selection of arbitration modes is determined by **PRTMOD0** and **PRTMOD1** of the *Arbitration Control Register*. These fields control the different priority group respectively. Each priority includes several functional circuits. The mapping between these fields and priority groups is as following table.

PRTMOD0 and PRTMOD1 are for AHB1 and AHB3 bus arbitration control respectively. In AHB4 bus, there is only *fixed-priority* mode.

Priority Modes	Priority Group
PRTMOD0	ARM926EJ-S Master I/F for Instruction (ARM926-I), ARM926EJ-S Master I/F for Data (ARM926-D) and EDMA.
PRTMOD1	BitBLT Accelerator, JPEG, SIC, UHC and UDC.
	SPU, I2S Controller, Display Controller and Capture.

The bus arbiter also provides a mechanism to limit the maximum burst length of the bus transfer. When current bus transfer count is equal to the maximum burst length, the access of current bus owner will be stop. The maximum burst length is fixed at 16.

5.2.4.1 Fixed Priority Mode

Fixed priority mode is selected if **PRTMODx** = 0. The order of priorities on the AHB mastership among the on-chip master modules, listed in. If two or more master modules request to access AHB bus at the same time, the higher priority request will get the permission to access AHB bus.

Priority Group	Group 0	Group 1	Group 2
Inter-Group Priority	1 (Lowest)	2	3 (Highest)
Intra-Group Priority			
1 (Lowest)	ARM-I	Reserved	Capture
2	ARM-D	UDC	
3	EDMA	UHC	Display Controller
4		Reserved	I2S Controller
5		SIC	SPU
6		JPEG	
7 (Highest)		BitBLT Accelerator	

The ARM core normally has the lowest priority under the fixed priority mode. N9H20 provides a mechanism to raise the priority of CPU request to the highest. If the **IPEN** bit (bit-4 of *AHB Control Register*) is set to 1, the **IPACT** bit (bit-5 of *AHB Control Register*) will be automatically set to 1 while an unmasked external FIQ or IRQ occurs. Under this circumstance, the ARM core will become the highest priority to access AHB bus.

The programmer can recover the original priority order by directly writing “1” to clear the **IPACT** bit. For example, this can be done that at the end of an interrupt service routine. Note that **IPACT** only can be automatically set to 1 by an external interrupt when **IPEN** = 1. It will not take effect for a programmer to directly write 1 to **IPACT** to raise ARM core's AHB priority.

5.2.4.2 Round Robin Priority Mode

Round-robin priority mode is selected if **PRTMODx** = 1. The AHB bus arbiter uses a round robin arbitration scheme for every master module to gain the bus ownership in turn. That is the requestor

having the highest priority becomes the lowest-priority requestor after it has been granted access.

5.2.4.3 Priority algorithm for AHB-4 bus

There are many real time interfaces attached on AHB4-DMA bus and fixed priority is used. The priority from high to low is SPU > I2S Controller > Display Controller > Capture.

5.2.4.4 Rotate rule Example:

In the default sequence of AHB-3 DMA Master bus, the priority is BitBLT Accelerator > JPEG > SIC > UHC > UDC.

If the UHC has been granted, the next priority order will be UDC > BitBLT Accelerator > JPEG > SIC > UHC.

5.2.5 Power Management Mode

N9H20 provides several power management scenarios to reduce power consumption. The operating clock of each different functional block could be enabled or disabled individually by controlling the corresponding bit of CLKEN control register. Software can turn-off the clock of unused blocks dynamically to save the power consumption. N9H20 also implements power management bit to disabled all system clock simultaneously. Besides, two different power islands (the core-power island and the RTC power island) are implemented in this chip. The core-power island could be power-off individually while all functional blocks are not used except the RTC.

There are three different power consumption levels defined in this chip. The following is the detail description for each power consumption level:

Normal Operation Mode

In this power consumption level, all functional blocks could work normally. Core power keeps in 1.8V. S/W could disable the clock of unused functional blocks dynamically to keep the minimum power consumption. Besides, S/W also could make the ARM926EJ-S CPU to enter power saving mode dynamically by disabling the CPU's clock. The clock of ARM926EJ-S CPU will be enabled automatically once the nIRQ or nFIQ is active.

Standby Mode

In this power consumption level, the clocks for all functional blocks including the ARM926EJ-S CPU are disabled. The crystal is also disabled. And the PLL is in power saving mode, too.

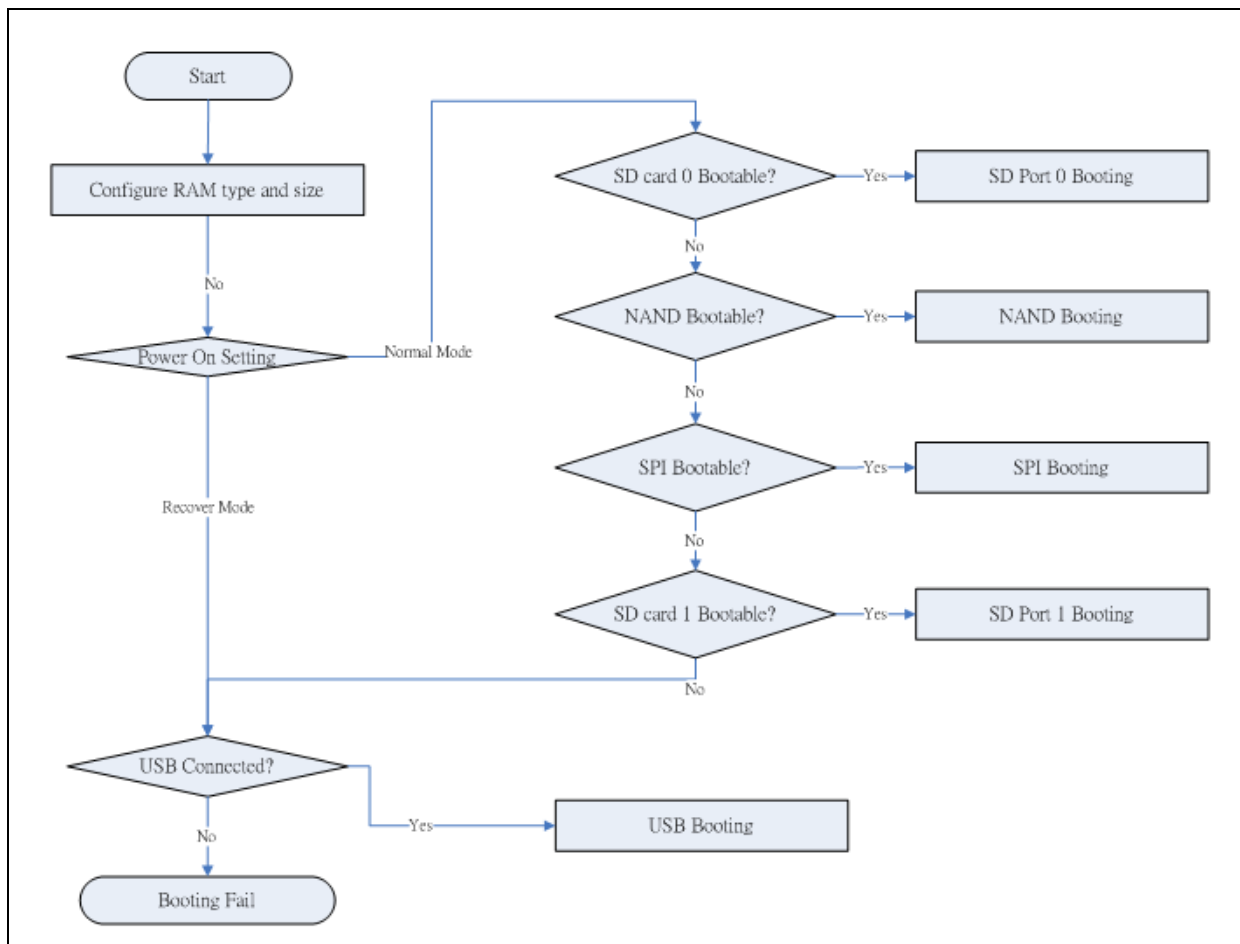
If chip is in Standby Mode, it could be wake up by many different wake-up events. These wake-up events could from external (GPIO) interrupt, USB host connect/disconnect/remote-wakeup event, USB device connect/disconnect event, UART trigger, SD Card interface, Touch Panel, KPI event and RTC alarm.

Power-Down Mode

In N9H20, there are two different power domains. One is for RTC function only and called RTC power domain while the other is for all other functions and called core power domain. By programming the bit PWR_ON of RTC register PWRON to low, the power of core power domain will be turned off and N9H20 will be in Power-Down Mode.

In this mode, only the power of RTC function is on and the power of all other functions will be turned off. By setting bit PWR_ON of RTC register PWRON to high, the core power will be turned on.

5.2.6 IBR (Internal Boot ROM) Sequence



5.2.7 System Management Control Registers

Register	Address	R/W	Description	Reset Value
GCR_BA = 0xB000_0000				
CHIPID	GCR_BA+0x00	R	Chip Identification Register	0x00FA_5C30
CHIPCFG	GCR_BA+0x04	R/W	Chip Power-On Configuration Register	0x0003_0XXX
AHBCTL	GCR_BA+0x10	R/W	AHB Bus Arbitration Control Register	0x0000_0000
AHBIPRST	GCR_BA+0x14	R/W	AHB IP Reset Control Resister	0x0000_0000
APBIPRST	GCR_BA+0x18	R/W	APB IP Reset Control Resister	0x0000_0000
MISCR	GCR_BA+0x20	R/W	Miscellaneous Control Register	0x0000_0000
CRBIST	GCR_BA+0x28	R/W	Cache RAM BIST Control & Status Register	0x0000_0000
EDSSR	GCR_BA+0x2C	R/W	EDMA Service Selection Register	0x7720_4270
MISSR	GCR_BA+0x30	R/W	Miscellaneous Status Register	0x0000_0000
GPAFUN	GCR_BA+0x80	R/W	GPIO A Multi Function Control Register	0x00X0_0000
GPBFUN	GCR_BA+0x84	R/W	GPIO B Multi Function Control Register	0x0000_0000
GPCFUN	GCR_BA+0x88	R/W	GPIO C Multi Function Control Register	0x0000_0000
GPDFUN	GCR_BA+0x8C	R/W	GPIO D Multi Function Control Register	0xXX00_03FF
GPEFUN	GCR_BA+0x90	R/W	GPIO E Multi Function Control Register	0x0000_0XX0
MISFUN	GCR_BA+0x94	R/W	Miscellaneous Multi Function Control Register	0x0000_0001F
MISCPKR	GCR_BA+0xA0	R/W	Miscellaneous Pins Control Register	0x0000_0000

Chip Identification Register (CHIPID)

This register provides specific read-only information for software to identify the revision and ID of chip.

Register	Address	R/W	Description	Reset Value
CHIPID	GCR_BA+0x00	R	Chip Identification Register	0x00FA_5C30

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CHIP_ID							
15	14	13	12	11	10	9	8
CHIP_ID							
7	6	5	4	3	2	1	0
CHIP_ID							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	CHIP_ID	Chip Identification Chip identification is "0xFA_5C30" to indicate the chip product ID is "FA5C30".

Chip Power-on Configuration Register (CHIPCFG)

This register provides information for software to identify chip's power-on setting. Bits [7:0] are the status of the power-on setting pins. These configuration bits could be modified by software programming.

Register	Address	R/W	Description	Reset Value
CHIPCFG	GCR_BA+0x04	R/W	Chip Power-On Configuration Register	0x0003_0XXX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						HVMODE	Reserved
15	14	13	12	11	10	9	8
Reserved					NTYPE	NPAGE	
7	6	5	4	3	2	1	0
NADDR	Reserved	SDRAMSEL			COPMODE		

Bits	Descriptions											
[31:18]	Reserved	Reserved										
[17]	HVMODE	ARM High Vector Mode This bit indicates the ARM926EJ-S CPU is in high vector mode. While ARM926EJ-S CPU is in high vector mode, the CPU will boot from 0xFFFF_0000 of 4GB system memory space. In this chip, the internal boot ROM is mapped to 0xFFFF_0000. 1'b0: ARM926EJ-S CPU boots from 0x0000_0000 1'b1: ARM926EJ-S CPU boots from 0xFFFF_0000										
[16:11]	Reserved	Reserved										
[10]	NTYPE	NAND Flash Type Selection This power-on setting value define the which NAND flash type is used for NAND booting. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0</td><td>EF (Error Free) NAND flash memory</td></tr><tr><td>0x1</td><td>Raw NAND flash memory</td></tr></table>	Value	Description	0x0	EF (Error Free) NAND flash memory	0x1	Raw NAND flash memory				
Value	Description											
0x0	EF (Error Free) NAND flash memory											
0x1	Raw NAND flash memory											
[9:8]	NPAGE	NAND Page Size This power-on setting value define the page size of NAND for NAND booting. <table><tr><th>NPAGE</th><th>Description</th></tr><tr><td>0x00</td><td>NAND page size is 2KB</td></tr><tr><td>0x01</td><td>NAND page size is 4KB</td></tr><tr><td>0x10</td><td>NAND page size is 8KB</td></tr><tr><td>0x11</td><td>Ignore NAND power-on setting</td></tr></table>	NPAGE	Description	0x00	NAND page size is 2KB	0x01	NAND page size is 4KB	0x10	NAND page size is 8KB	0x11	Ignore NAND power-on setting
NPAGE	Description											
0x00	NAND page size is 2KB											
0x01	NAND page size is 4KB											
0x10	NAND page size is 8KB											
0x11	Ignore NAND power-on setting											

[7]	NADDR	NAND Address Cycle This 1-bit power-on setting value is to define the NAND address cycle count for NAND booting. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0</td><td>The NAND address cycle is 4.</td></tr><tr><td>0x1</td><td>The NAND address cycle is 5.</td></tr></table>	Value	Description	0x0	The NAND address cycle is 4.	0x1	The NAND address cycle is 5.																												
Value	Description																																			
0x0	The NAND address cycle is 4.																																			
0x1	The NAND address cycle is 5.																																			
[6]	Reserved	Reserved																																		
[5:4]	SDRAMSEL	SDRAM Type Selection This field reflects the power-on setting value about which type of SDRAM is used for system memory. 2'b00: SDR SDRAM (normal SDRAM) is used. 2'b01: Low Power DDR (mDDR) SDRAM is used. 2'b10: DDR2 SDRAM is used. 2'b11: DDR SDRAM is used.																																		
[3:0]	COPMODE	Chip Operation Mode Selection This field reflects power-on setting value about the chip is in which operation mode. <table><tr><th>Value</th><th>Description</th></tr><tr><td>0x0000</td><td>Reserved</td></tr><tr><td>0x0001</td><td>Reserved</td></tr><tr><td>0x0010</td><td>Reserved</td></tr><tr><td>0x0011</td><td>Reserved</td></tr><tr><td>0x0100</td><td>Reserved</td></tr><tr><td>0x0101</td><td>Reserved</td></tr><tr><td>0x0110</td><td>Reserved.</td></tr><tr><td>0x0111</td><td>Reserved</td></tr><tr><td>0x1000</td><td>Reserved.</td></tr><tr><td>0x1001</td><td>Reserved.</td></tr><tr><td>0x1010</td><td>Boot from IBR Recover Mode with crystal input is 12MHz.</td></tr><tr><td>0x1011</td><td>Boot From IBR Normal Mode with crystal input is 12MHz.</td></tr><tr><td>0x1100</td><td>Reserved.</td></tr><tr><td>0x1101</td><td>Reserved.</td></tr><tr><td>0x1110</td><td>Boot from IBR Recover Mode with crystal input is 27MHz.</td></tr><tr><td>0x1111</td><td>Boot From IBR Normal Mode with crystal input is 27MHz.</td></tr></table> Please note if any of the reserved setting is used, CHIP's behavior is undefined.	Value	Description	0x0000	Reserved	0x0001	Reserved	0x0010	Reserved	0x0011	Reserved	0x0100	Reserved	0x0101	Reserved	0x0110	Reserved.	0x0111	Reserved	0x1000	Reserved.	0x1001	Reserved.	0x1010	Boot from IBR Recover Mode with crystal input is 12MHz.	0x1011	Boot From IBR Normal Mode with crystal input is 12MHz.	0x1100	Reserved.	0x1101	Reserved.	0x1110	Boot from IBR Recover Mode with crystal input is 27MHz.	0x1111	Boot From IBR Normal Mode with crystal input is 27MHz.
Value	Description																																			
0x0000	Reserved																																			
0x0001	Reserved																																			
0x0010	Reserved																																			
0x0011	Reserved																																			
0x0100	Reserved																																			
0x0101	Reserved																																			
0x0110	Reserved.																																			
0x0111	Reserved																																			
0x1000	Reserved.																																			
0x1001	Reserved.																																			
0x1010	Boot from IBR Recover Mode with crystal input is 12MHz.																																			
0x1011	Boot From IBR Normal Mode with crystal input is 12MHz.																																			
0x1100	Reserved.																																			
0x1101	Reserved.																																			
0x1110	Boot from IBR Recover Mode with crystal input is 27MHz.																																			
0x1111	Boot From IBR Normal Mode with crystal input is 27MHz.																																			

AHB Bus Arbitration Control Register (AHBCTL)

Register	Address	R/W	Description	Reset Value
AHBCTL	GCR_BA+0x10	R/W	AHB Bus Arbitration Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		IPACT	IPEN	Reserved		PRTMOD1	PRTMOD0

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5]	IPACT	Interrupt Active Status When IPEN="1", this bit is set high when nIRQ or nFIQ is active. Note: This bit is available only when the PRTMOD1=0 and PRTMOD0=0.
[4]	IPEN	CPU Priority Raising Enable This bit is used to enable the CPU priority raising function. If this bit is set high, the CPU will get higher AHB1 bus access priority while nIRQ or nFIQ is active. 0: Disable CPU priority raising function. 1: Enable CPU priority raising function. Note: This bit is available only when the PRTMOD1=0 and PRTMOD0=0.
[3]	Reserved	Reserved
[2]	Reserved	Reserved
[1]	PRTMOD1	Priority Mode Control 1 This bit control the priority mode used in AHB3 bus. 0: The fix priority mode is used. 1: The round-robin priority mode is used.
[0]	PRTMOD0	Priority Mode Control 0 This bit control the priority mode used in AHB1 bus. 0: The fix priority mode is used. 1: The round-robin priority mode is used.

AHB IP Reset Control Register (AHBIPRST)

Each bit of this register is to reset its corresponding functional circuit. By writing 1'b1 to any reset bit, the corresponding functional circuit will be reset, and all operating state and control registers of that functional circuit will return to their default power-on state. By writing 1'b0 to any reset bit, the reset was terminated.

Register	Address	R/W	Description	Reset Value
AHBIPRST	GCR_BA+0x14	R/W	AHB IP Reset Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						JPGRST	BLTRST
15	14	13	12	11	10	9	8
Reserved	FSCRST	Reserved	Reserved	CAPRST	VPOSTST	I2SRST	SPURST
7	6	5	4	3	2	1	0
UHRST	UDCRST	SICRST	TICRST	EDMARST	SRAMRST	Reserved	SDICRST

Bits	Descriptions	
[31:18]	Reserved	Reserved
[17]	JPGRST	JPEG Reset 1'b0: JPEG reset is no active. 1'b1: JPEG reset is active.
[16]	BLTRST	BitBLT Accelerator Reset 1'b0: BitBLT Accelerator reset is no active. 1'b1: BitBLT Accelerator reset is active.
[15]	Reserved	Reserved
[14]	FSCRST	FSC Reset 1'b0: FSC reset is no active. 1'b1: FSC reset is active.
[13:12]	Reserved	Reserved
[11]	CAPRST	Capture Reset 1'b0: Capture reset is no active. 1'b1: Capture reset is active.
[10]	VPOSTRST	VPOST Reset 1'b0: VPOST reset is no active. 1'b1: VPOST reset is active.
[9]	I2SRST	I2S Reset 1'b0: I2S reset is no active. 1'b1: I2S reset is active.

[8]	SPURST	SPU Reset 1'b0: SPU reset is no active. 1'b1: SPU reset is active.
[7]	UHCIRST	UHC Reset 1'b0: UHC reset is no active. 1'b1: UHC reset is active.
[6]	UDCRST	UDC Reset 1'b0: UDC reset is no active. 1'b1: UDC reset is active.
[5]	SICRST	SIC Reset 1'b0: SIC reset is no active. 1'b1: SIC reset is active.
[4]	TICRST	TIC Reset 1'b0: TIC reset is no active. 1'b1: TIC reset is active.
[3]	EDMARST	EDMA Reset 1'b0: EDMA reset is no active. 1'b1: EDMA reset is active.
[2]	SRAMRST	SRAM Controller Reset 1'b0: SRAM controller reset is no active. 1'b1: SRAM controller reset is active.
[1]	Reserved	Reserved
[0]	SDICRST	SDIC Reset 1'b0: SDIC reset is no active. 1'b1: SDIC reset is active.

APB IP Reset Control Register (APBIPRST)

Each bit of this register is to reset its corresponding functional circuit. By writing 1'b1 to any reset bit, the corresponding functional circuit will be reset, and all operating state and control registers of that functional circuit will return to their default power-on state. By writing 1'b0 to any reset bit, the reset was terminated.

Register	Address	R/W	Description	Reset Value
APBIPRST	GCR_BA+0x18	R/W	APB IP Reset Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
KPIRST	ADCRST	SPI1RST	SPI0RST	Reserved	PWMRST	Reserved	I2CRST
7	6	5	4	3	2	1	0
UART1RST	UART0RST	TMR1RST	TMR0RST	WDTRST	Reserved	GPIORST	Reserved

Bits	Descriptions	
[31:15]	Reserved	Reserved
[15]	KPIRST	KPI Reset 1'b0: KPI reset is no active. 1'b1: KPI reset is active.
[14]	ADCRST	ADC Reset 1'b0: ADC reset is no active. 1'b1: ADC reset is active.
[13]	SPI1RST	SPIMS 1 Reset 1'b0: SPIMS 1 reset is no active. 1'b1: SPIMS 1 reset is active.
[12]	SPI0RST	SPIMS 0 Reset 1'b0: SPIMS 0 reset is no active. 1'b1: SPIMS 0 reset is active.
[11]	Reserved	Reserved
[10]	PWMRST	PWM Reset 1'b0: PWM reset is no active. 1'b1: PWM reset is active.
[9]	Reserved	Reserved
[8]	I2CRST	I2C Reset 1'b0: I2C reset is no active. 1'b1: I2C reset is active.

[7]	UART1RST	UART 1 Reset 1'b0: UART 1 reset is no active. 1'b1: UART 1 reset is active.
[6]	UART0RST	UART 0 Reset 1'b0: UART 0 reset is no active. 1'b1: UART 0 reset is active.
[5]	TMR1RST	Timer 1 Reset 1'b0: Timer 1 reset is no active. 1'b1: Timer 1 reset is active.
[4]	TMR0RST	UART 0 Reset 1'b0: Timer 0 reset is no active. 1'b1: Timer 0 reset is active.
[3]	WDTRST	Watch-dog Timer Reset 1'b0: Watch-dog Timer reset is no active. 1'b1: Watch-dog Timer reset is active.
[2]	Reserved	Reserved
[1]	GPORST	GPIO Controller Reset 1'b0: GPIO reset is no active. 1'b1: GPIO reset is active.
[0]	Reserved	Reserved

Miscellaneous Control Register (MISCR)

Register	Address	R/W	Description	Reset Value
MISCR	GCR_BA+0x20	R/W	Miscellaneous Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							WDTRSTEN
23	22	21	20	19	18	17	16
Reserved	Reserved			Reserved		Reserved	
15	14	13	12	11	10	9	8
Reserved						LVR_RDY	LVR_EN
7	6	5	4	3	2	1	0
Reserved						CPURSTON	CPURST

Bits	Descriptions	
[31:25]	Reserved	Reserved
[24]	WDTRSTEN	WatchDog Timer Reset Connection Enable This bit is use to enable the function that connect watch-dog timer reset to RST_ pin. If this bit is disabled, the watch-dog timer reset is connected to other multi-functional pins. Please refer GPAFUN, GPBFUN and GPDFUN register for detail setting. If this bit is enabled, the watch-dog timer reset is connected to RST_ pin internally. 1'b0: Watch-dog timer reset is connected to multi-functional pins. (Default) 1'b1: Watch-dog timer reset is connected to RST_ pin internally.
[22:20]	Reserved	Reserved
[15:10]	Reserved	Reserved
[9]	LVR_RDY	Low Voltage Reset Function Ready After low voltage reset function is enabled, it needs at least 5us to make its output signal be stable. Before the output of low voltage reset function is stable, it has to be masked. Or the N9H20 will be affected during this 5us period. This bit is to gate the output of low voltage reset function. If this bit is low (1'b0), the output of low voltage reset function will be gated and N9H20 would never be affected by low voltage reset function. While this bit is set high (1'b1), N9H20 would affected by low voltage reset function. 1'b0: Low Voltage Reset function is gated and would never affect the N9H20. 1'b1: Low Voltage Reset function is ready to use. NOTE1: Before setting this bit to '1', it's necessary to be sure that LVR_EN has been enabled for 5us. Otherwise, an unexpected reset may follow. NOTE2: Before setting LVD_EN to '0', it's necessary to be sure this bit has been set to '0'. Otherwise, an unexpected reset will follow.

[8]	LVR_EN	Low Voltage Reset Function Enable This bit is used to enable the low voltage reset function. After low voltage reset function is enabled, it needs at least 5us to make its output signal be stable. Before the output of low voltage reset function is stable, it has to be masked by keeping the <i>LVR_RDY</i> in low. Or the N9H20 will be affected during this 5us period. 0 = Low Voltage Reset function is disabled. 1 = Low Voltage Reset function is enabled. NOTE1: When this bit is set high (1'b1), the LVR functional block will always consumes a current of about several tens micro Amps. NOTE2: Do not disable this bit when <i>LVR_RDY</i> is still set as '1'. Otherwise, an immediate reset follows the ill setting. And be sure to enable <i>LVR_RDY</i> bit after <i>LVD_EN</i> is enabled for at least 5us.
[7:2]	Reserved	Reserved
[1]	CPURSTON	CPU Reset ON (cleared by register write 0) This bit is to reset CPU continuously. By writing 1'b1 to this bit, the CPU will be reset permanently. Writing "0" to this bit clear the reset and make CPU to return the normal operation state. 1'b0: CPU reset is no active. 1'b1: CPU reset is active.
[0]	CPURST	CPU Reset (automatically cleared) This bit is to reset CPU. By writing 1'b1 to this bit, the CPU will be reset. After reset completion, the reset bit will be cleared automatically. Writing "0" to any reset bit wouldn't take any effect. 1'b0: CPU reset is no active. 1'b1: CPU reset is active.

Cache RAM BIST Control & Status Register (CRBIST)

Register	Address	R/W	Description	Reset Value
CRBIST	GCR_BA+0x28	R/W	Cache RAM BIST Control & Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ICV_F	ICT_F	ICD3_F	ICD2_F	ICD1_F	ICD0_F
23	22	21	20	19	18	17	16
MMU_F	DCDIR_F	DCV_F	DCT_F	DCD3_F	DCD2_F	DCD1_F	DCD0_F
15	14	13	12	11	10	9	8
BISTEN	Reserved	ICV_R	ICT_R	ICD3_R	ICD2_R	ICD1_R	ICD0_R
7	6	5	4	3	2	1	0
MMU_R	DCDIR_R	DCV_R	DCT_R	DCD3_R	DCD2_R	DCD1_R	DCD0_R

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29]	ICV_F	I-Cache Valid RAM BIST Failed Flag This bit indicates if the I-Cache valid RAM BIST test is O.K. or Failed. This value could only be referred while <i>ICV_R</i> is 1'b0. 1'b0: I-Cache valid RAM BIST test is O.K. 1'b1: I-Cache valid RAM BIST test failed.
[28]	ICT_F	I-Cache Tag RAM BIST Failed Flag This bit indicates if the I-Cache tag RAM BIST test is O.K. or Failed. This value could only be referred while <i>ICT_R</i> is 1'b0. 1'b0: I-Cache tag RAM BIST test is O.K. 1'b1: I-Cache tag RAM BIST test failed.
[27]	ICD3_F	I-Cache Data RAM 3 BIST Failed Flag This bit indicates if the I-Cache data RAM 3 BIST test is O.K. or Failed. This value could only be referred while <i>ICD3_R</i> is 1'b0. 1'b0: I-Cache data RAM 3 BIST test is O.K. 1'b1: I-Cache data RAM 3 BIST test failed.
[26]	ICD2_F	I-Cache Data RAM 2 BIST Failed Flag This bit indicates if the I-Cache data RAM 2 BIST test is O.K. or Failed. This value could only be referred while <i>ICD2_R</i> is 1'b0. 1'b0: I-Cache data RAM 2 BIST test is O.K. 1'b1: I-Cache data RAM 2 BIST test failed.
[25]	ICD1_F	I-Cache Data RAM 1 BIST Failed Flag This bit indicates if the I-Cache data RAM 1 BIST test is O.K. or Failed. This value could only be referred while <i>ICD1_R</i> is 1'b0. 1'b0: I-Cache data RAM 1 BIST test is O.K. 1'b1: I-Cache data RAM 1 BIST test failed.
[24]	ICD0_F	I-Cache Data RAM 0 BIST Failed Flag This bit indicates if the I-Cache data RAM 0 BIST test is O.K. or Failed. This value could only be referred while <i>ICD0_R</i> is 1'b0. 1'b0: I-Cache data RAM 0 BIST test is O.K. 1'b1: I-Cache data RAM 0 BIST test failed.

[23]	MMU_F	MMU RAM BIST Failed Flag This bit indicates if the MMU RAM BIST test is O.K. or Failed. This value could only be referred while <i>MMU_R</i> is 1'b0. 1'b0: MMU RAM BIST test is O.K. 1'b1: MMU RAM BIST test failed.
[22]	DCDIR_F	D-Cache Dirty RAM BIST Failed Flag This bit indicates if the D-Cache dirty RAM BIST test is O.K. or Failed. This value could only be referred while <i>DCDIR_R</i> is 1'b0. 1'b0: D-Cache dirty RAM BIST test is O.K. 1'b1: D-Cache dirty RAM BIST test failed.
[21]	DCV_F	D-Cache Valid RAM BIST Failed Flag This bit indicates if the D-Cache valid RAM BIST test is O.K. or Failed. This value could only be referred while <i>DCV_R</i> is 1'b0. 1'b0: D-Cache valid RAM BIST test is O.K. 1'b1: D-Cache valid RAM BIST test failed.
[20]	DCT_F	D-Cache Tag RAM BIST Failed Flag This bit indicates if the D-Cache tag RAM BIST test is O.K. or Failed. This value could only be referred while <i>DCT_R</i> is 1'b0. 1'b0: D-Cache tag RAM BIST test is O.K. 1'b1: D-Cache tag RAM BIST test failed.
[19]	DCD3_F	D-Cache Data RAM 3 BIST Failed Flag This bit indicates if the D-Cache data RAM 3 BIST test is O.K. or Failed. This value could only be referred while <i>DCD3_R</i> is 1'b0. 1'b0: D-Cache data RAM 3 BIST test is O.K. 1'b1: D-Cache data RAM 3 BIST test failed.
[18]	DCD2_F	D-Cache Data RAM 2 BIST Failed Flag This bit indicates if the D-Cache data RAM 2 BIST test is O.K. or Failed. This value could only be referred while <i>DCD2_R</i> is 1'b0. 1'b0: D-Cache data RAM 2 BIST test is O.K. 1'b1: D-Cache data RAM 2 BIST test failed.
[17]	DCD1_F	D-Cache Data RAM 1 BIST Failed Flag This bit indicates if the D-Cache data RAM 1 BIST test is O.K. or Failed. This value could only be referred while <i>DCD1_R</i> is 1'b0. 1'b0: D-Cache data RAM 1 BIST test is O.K. 1'b1: D-Cache data RAM 1 BIST test failed.
[16]	DCD0_F	D-Cache Data RAM 0 BIST Failed Flag This bit indicates if the D-Cache data RAM 0 BIST test is O.K. or Failed. This value could only be referred while <i>DCD0_R</i> is 1'b0. 1'b0: D-Cache data RAM 0 BIST test is O.K. 1'b1: D-Cache data RAM 0 BIST test failed.
[15]	BISTEN	Cache RAM BIST Test Enable This bit is to enable the cache RAM BIST test mode. When this is set high, the cache and MMU RAM will enter BIST test mode. This bit has to be kept high and should not be cleared during the BIST test mode. The behavior is undefined if this bit is cleared during BIST test mode. Besides, it's important to note that program should never enable this bit while CPU is operating on cache-on mode, or the CPU's behavior is undefined. 1'b0: Cache & MMU RAM BIST test mode is disabled. 1'b1: Cache & MMU RAM BIST test mode is enabled.
[14]	Reserved	Reserved

[13]	ICV_R	I-Cache Valid RAM BIST Running Flag This bit indicates if the I-Cache valid RAM BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache valid RAM BIST test has completed. 1'b1: I-Cache valid RAM BIST test is running.
[12]	ICT_R	I-Cache Tag RAM BIST Running Flag This bit indicates if the I-Cache tag RAM BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache tag RAM BIST test has completed. 1'b1: I-Cache tag RAM BIST test is running.
[11]	ICD3_R	I-Cache Data RAM 3 BIST Running Flag This bit indicates if the I-Cache data RAM 3 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 3 BIST test has completed. 1'b1: I-Cache data RAM 3 BIST test is running.
[10]	ICD2_R	I-Cache Data RAM 2 BIST Running Flag This bit indicates if the I-Cache data RAM 2 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 2 BIST test has completed. 1'b1: I-Cache data RAM 2 BIST test is running.
[9]	ICD1_R	I-Cache Data RAM 1 BIST Running Flag This bit indicates if the I-Cache data RAM 1 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 1 BIST test has completed. 1'b1: I-Cache data RAM 1 BIST test is running.
[8]	ICD0_R	I-Cache Data RAM 0 BIST Running Flag This bit indicates if the I-Cache data RAM 0 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 0 BIST test has completed. 1'b1: I-Cache data RAM 0 BIST test is running.
[7]	MMU_R	MMU RAM BIST Running Flag This bit indicates if the MMU RAM BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: MMU RAM BIST test has completed. 1'b1: MMU RAM BIST test is running.
[6]	DCDIR_R	D-Cache Dirty RAM BIST Running Flag This bit indicates if the D-Cache dirty RAM BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache dirty RAM BIST test has completed. 1'b1: I-Cache dirty RAM BIST test is running.
[5]	DCV_R	D-Cache Valid RAM BIST Running Flag This bit indicates if the D-Cache valid RAM BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache valid RAM BIST test has completed. 1'b1: I-Cache valid RAM BIST test is running.
[4]	DCT_R	D-Cache Tag RAM BIST Running Flag This bit indicates if the D-Cache tag RAM BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache tag RAM BIST test has completed. 1'b1: I-Cache tag RAM BIST test is running.

[3]	DCD3_R	D-Cache Data RAM 3 BIST Running Flag This bit indicates if the D-Cache data RAM 3 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 3 BIST test has completed. 1'b1: I-Cache data RAM 3 BIST test is running.
[2]	DCD2_R	D-Cache Data RAM 2 BIST Running Flag This bit indicates if the D-Cache data RAM 2 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 2 BIST test has completed. 1'b1: I-Cache data RAM 2 BIST test is running.
[1]	DCD1_R	D-Cache Data RAM 1 BIST Running Flag This bit indicates if the D-Cache data RAM 1 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 1 BIST test has completed. 1'b1: I-Cache data RAM 1 BIST test is running.
[0]	DCD0_R	D-Cache Data RAM 0 BIST Running Flag This bit indicates if the D-Cache data RAM 0 BIST test is running or has completed. This value could only be referred while <i>BISTEN</i> is 1'b1. 1'b0: I-Cache data RAM 0 BIST test has completed. 1'b1: I-Cache data RAM 0 BIST test is running.

EDMA Service Selection Control Register (EDSSR)

Register	Address	R/W	Description	Reset Value
EDSSR	GCR_BA+0x2C	R/W	EDMA Service Selection Control Register	0x7720_4270

31	30	29	28	27	26	25	24
Reserved	CH4_TXSEL			Reserved	CH3_TXSEL		
23	22	21	20	19	18	17	16
Reserved	CH2_TXSEL			Reserved	CH1_TXSEL		
15	14	13	12	11	10	9	8
Reserved	CH4_RXSEL			Reserved	CH3_RXSEL		
7	6	5	4	3	2	1	0
Reserved	CH2_RXSEL			Reserved	CH1_RXSEL		

Bits	Descriptions	
[31]	Reserved	Reserved
[30:28]	CH4_TXSEL	EDMA Channel 4 Tx Selection This filed defined EDMA channel 4 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the TX request of EDMA channel 4 is disabled. 3'b000: SPIMS 0 3'b001: SPIMS 1 3'b010: UART 0 3'b011: UART 1 3'b111: Disable (Default) Others: Reserved
[27]	Reserved	Reserved
[26:24]	CH3_TXSEL	EDMA Channel 3 Tx Selection This filed defined EDMA channel 3 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the TX request of EDMA channel 3 is disabled. 3'b000: SPIMS 0 3'b001: SPIMS 1 3'b010: UART 0 3'b011: UART 1 3'b111: Disable (Default) Others: Reserved
[23]	Reserved	Reserved

[22:20]	CH2_TXSEL	EDMA Channel 2 Tx Selection This filed defined EDMA channel 2 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the TX request of EDMA channel 2 is disabled. 3'b000: SPIMS 0 3'b001: SPIMS 1 3'b010: UART 0 (Default) 3'b011: UART 1 3'b111: Disable Others: Reserved
[19]	Reserved	Reserved
[18:16]	CH1_TXSEL	EDMA Channel 1 Tx Selection This filed defined EDMA channel 1 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the TX request of EDMA channel 1 is disabled. 3'b000: SPIMS 0 (Default) 3'b001: SPIMS 1 3'b010: UART 0 3'b011: UART 1 3'b111: Disable Others: Reserved
[15]	Reserved	Reserved
[14:12]	CH4_RXSEL	EDMA Channel 4 Rx Selection This filed defined EDMA channel 4 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the RX request of EDMA channel 4 is disabled. 3'b000: SPIMS 0 3'b001: SPIMS 1 3'b010: UART 0 3'b011: UART 1 3'b100: ADC Controller (Default) 3'b111: Disable Others: Reserved
[11]	Reserved	Reserved
[10:8]	CH3_RXSEL	EDMA Channel 3 Rx Selection This filed defined EDMA channel 3 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the RX request of EDMA channel 3 is disabled. 3'b000: SPIMS 0 3'b001: SPIMS 1 3'b010: UART 0 (Default) 3'b011: UART 1 3'b100: ADC Controller 3'b111: Disable Others: Reserved
[7]	Reserved	Reserved
[6:4]	CH2_RXSEL	EDMA Channel 2 Rx Selection This filed defined EDMA channel 2 is to service which on-chip peripherals. If this filed is configured to be 3'b111, the RX request of EDMA channel 2 is disabled. 3'b000: SPIMS 0 3'b001: SPIMS 1 3'b010: UART 0 3'b011: UART 1 3'b100: ADC Controller 3'b111: Disable (Default) Others: Reserved
[3]	Reserved	Reserved

[2:0]	CH1_RXSEL	EDMA Channel 1 Rx Selection This field defined EDMA channel 1 is to service which on-chip peripherals. If this field is configured to be 3'b111, the RX request of EDMA channel 1 is disabled. 3'b000: SPIMS 0 (Default) 3'b001: SPIMS 1 3'b010: UART 0 3'b011: UART 1 3'b100: ADC Controller 3'b111: Disable Others: Reserved

Miscellaneous Status Register (MISSR)

Register	Address	R/W	Description	Reset Value
MISSR	GCR_BA+0x30	R/W	Miscellaneous Status Register	0x0000_0000

31	30	29	28	27	26	25	24
KPI_WS	ADC_WS	UHC_WS	UDC_WS	UART_WS	SDH_WS	RTC_WS	GPIO_WS
23	22	21	20	19	18	17	16
KPI_WE	ADC_WE	UHC_WE	UDC_WE	UART_WE	SDH_WE	RTC_WE	GPIO_WE
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
			CPU_RST	WDT_RST	KPI_RST	LVR_RST	EXT_RST

Bits	Descriptions	
[31]	KPI_WS	KPI Wake-Up Status This bit indicates if there is a wake-up event from KPI. If KPI_WE is set high and a wake-up event from KPI, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from KPI 1: A wake-up event from KPI
[30]	ADC_WS	ADC Wake-Up Status This bit indicates if there is a wake-up event from ADC. If ADC_WE is set high and a wake-up event from ADC, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from ADC 1: A wake-up event from ADC
[29]	UHC_WS	UHC Wake-Up Status This bit indicates if there is a wake-up event from USB 1.1 host controller. If UHC_WE is set high and a wake-up event from USB 1.1 host controller, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from USB 1.1 host controller 1: A wake-up event from USB 1.1 host controller

[28]	UDC_WS	UDC Wake-Up Status This bit indicates if there is a wake-up event from USB 2.0 device controller. If UDC_WE is set high and a wake-up event from USB 2.0 device controller, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from USB 2.0 device controller 1: A wake-up event from USB 2.0 device controller
[27]	UART_WS	UART Wake-Up Status This bit indicates if there is a wake-up event from UART. If UART_WE is set high and a wake-up event from UART, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from UART 1: A wake-up event from UART
[26]	SDH_WS	SD Controller Wake-Up Status This bit indicates if there is a wake-up event from SD controller. If SDH_WE is set high and a wake-up event from SD controller, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from SD controller 1: A wake-up event from SD controller
[25]	RTC_WS	RTC Wake-Up Status This bit indicates if there is a wake-up event from RTC. If RTC_WE is set high and a wake-up event from RTC, the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from RTC 1: A wake-up event from RTC
[24]	GPIO_WS	GPIO Wake-Up Status This bit indicates if there is a wake-up event from GPIO. If GPIO_WE is set high and a wake-up event from GPIO the chip will be wake-up from Standby Mode and this bit will be set high. Write 1 to this bit to clear the status. 0: No wake-up event from GPIO 1: A wake-up event from GPIO
[23]	KPI_WE	KPI Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by KPI wake-up event. 0: Chip couldn't be wake-up from Standby Mode by KPI wake-up event. 1: Chip could be wake-up from Standby Mode by KPI wake-up event.
[22]	ADC_WE	ADC Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by ADC wake-up event. 0: Chip couldn't be wake-up from Standby Mode by ADC wake-up event. 1: Chip could be wake-up from Standby Mode by ADC wake-up event.

[21]	UHC_WE	UHC Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by USB 1.1 host controller wake-up event. 0: Chip couldn't be wake-up from Standby Mode by USB 1.1 host wake-up event. 1: Chip could be wake-up from Standby Mode by USB 1.1 host wake-up event.
[20]	UDC_WE	UDC Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by USB 2.0 device controller wake-up event. 0: Chip couldn't be wake-up from Standby Mode by USB 2.0 device wake-up event. 1: Chip could be wake-up from Standby Mode by USB 2.0 device wake-up event.
[19]	UART_WE	UART Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by UART wake-up event. 0: Chip couldn't be wake-up from Standby Mode by UART wake-up event. 1: Chip could be wake-up from Standby Mode by UART wake-up event.
[18]	SDH_WE	SD Controller Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by SD Controller wake-up event. 0: Chip couldn't be wake-up from Standby Mode by SD Controller wake-up event. 1: Chip could be wake-up from Standby Mode by SD Controller wake-up event.
[17]	RTC_WE	RTC Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by RTC wake-up event. 0: Chip couldn't be wake-up from Standby Mode by RTC wake-up event. 1: Chip could be wake-up from Standby Mode by RTC wake-up event.
[16]	GPIO_WE	GPIO Wake-Up Enable This bit is to control if the chip could be wake-up from Standby Mode by GPIO wake-up event. 0: Chip couldn't be wake-up from Standby Mode by GPIO wake-up event. 1: Chip could be wake-up from Standby Mode by GPIO wake-up event.
[15:5]	Reserved	Reserved
[4]	CPU_RST	CPU Reset Active Status This bit is to record if the CPU reset is active during operation. Write 1 to this bit to clear the status. 0: CPU reset is not active during operation. 1: CPU reset is active during operation.

[3]	WDT_RST	WDT Reset Active Status This bit is to record if the WDT (Watch Dog Timer) reset is active during operation. Write 1 to this bit to clear the status. 0: WDT reset is not active during operation. 1: WDT reset is active during operation.
[2]	KPI_RST	KPI Reset Active Status This bit is to record if the KPI reset is active during operation. Write 1 to this bit to clear the status. 0: KPI reset is not active during operation. 1: KPI reset is active during operation.
[1]	LVR_RST	LVR Reset Active Status This bit is to record if the LVR reset is active during operation. Write 1 to this bit to clear the status. 0: LVR reset is not active during operation. 1: LVR reset is active during operation.
[0]	EXT_RST	External Reset Pin Active Status This bit is to record if the external reset pin (RST_) is active during operation. Write 1 to this bit to clear the status. 0: External reset pin (RST_) is not active during operation. 1: External reset pin (RST_) is active during operation.

GPIO A Multi Function Control Register (GPAFUN)

This register defines the multi function description for GPIO A.

Note: The pull-up enable of the pins shared with GPIOA is controlled by register GPIOA_PUEN directly. Users have to set the GPIOA_PUEN properly before the chip entered the standby mode.

Register	Address	R/W	Description	Reset Value
GPAFUN	GCR_BA+0x80	R/W	GPIO A Multi Function Control Register	0x00X0_0000

31	30	29	28	27	26	25	24
Reserved		Reserved		Reserved		Reserved	
23	22	21	20	19	18	17	16
MF_GPA11		MF_GPA10		MF_GPA9		MF_GPA8	
15	14	13	12	11	10	9	8
MF_GPA7		MF_GPA6		MF_GPA5		MF_GPA4	
7	6	5	4	3	2	1	0
MF_GPA3		MF_GPA2		MF_GPA1		MF_GPA0	

Bits	Descriptions			
[31:24]	Reserved	Reserved This field is reserved and keep all these bits in zero is necessary.		
[23:22]	MF_GPA11	GPIOA[11] Multi Function		
		Pin Name	MF_GPA11	
		URRXD	2'b00 GPIOA[11]	2'b01 LMVSYNC
[21:20]	MF_GPA10	GPIOA[10] Multi Function		
		Pin Name	MF_GPA10	
		URTXD	2'b00 GPIOA[10]	2'b01 SPI1_CS1_
[19:18]	MF_GPA9	GPIOA[9] Multi Function		
		Pin Name	MF_GPA9	
		GPA[9]	2'b00 GPIOA[9]	2'b01 KPI_SO[7]
[17:16]	MF_GPA8	GPIOA[8] Multi Function		
		Pin Name	MF_GPA8	
		GPA[8]	2'b00 GPIOA[8]	2'b01 KPI_SO[6]

[15:14]	MF_GPA7	GPIOA[7] Multi Function				
		Pin Name	MF_GPA7			
		GPA[7]	2'b00	2'b01	2'b10	2'b11
GPIOA[7]	KPI_SO[5]		Reserved	Reserved		
[13:12]	MF_GPA6	GPIOA[6] Multi Function				
		Pin Name	MF_GPA6			
		GPA[6]	2'b00	2'b01	2'b10	2'b11
GPIOA[6]	KPI_SO[4]		SPI1_CS1_	SPI1_CS1_		
[11:10]	MF_GPA5	GPIOA[5] Multi Function				
		Pin Name	MF_GPA5			
		GPA[5]	2'b00	2'b01	2'b10	2'b11
GPIOA[5]	KPI_SO[3]		SPI0_CS1_	SPI0_CS1_		
[9:8]	MF_GPA4	GPIOA[4] Multi Function				
		Pin Name	MF_GPA4			
		GPA[4]	2'b00	2'b01	2'b10	2'b11
GPIOA[4]	KPI_SO[2]		UHL_DM1	UHL_DM1		
[7:6]	MF_GPA3	GPIOA[3] Multi Function				
		Pin Name	MF_GPA3			
		GPA[3]	2'b00	2'b01	2'b10	2'b11
GPIOA[3]	KPI_SO[1]		UHL_DP1	UHL_DP1		
[5:4]	MF_GPA2	GPIOA[2] Multi Function				
		Pin Name	MF_GPA2			
		GPA[2]	2'b00	2'b01	2'b10	2'b11
GPIOA[2]	KPI_SO[0]		LMVSYNC	LMVSYNC		
[3:2]	MF_GPA1	GPIOA[1] Multi Function				
		Pin Name	MF_GPA1			
		GPA[1]	2'b00	2'b01	2'b10	2'b11
GPIOA[1]	Reserved		SD_CD_	SD_CD_		
[1:0]	MF_GPA0	GPIOA[0] Multi Function				
		Pin Name	MF_GPA0			
		GPA[0]	2'b00	2'b01	2'b10	2'b11
GPIOA[0]	Reserved		Reserved	Reserved		

GPIO B Multi Function Control Register (GPBFUN)

Note: The pull-up enable of the pins shared with GPIOB is controlled by register GPIOB_PUEN directly. Users have to set the GPIOB_PUEN properly before the chip entered the standby mode.

Register	Address	R/W	Description	Reset Value
GPBFUN	GCR_BA+0x84	R/W	GPIO B Multi Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MF_GPB15		MF_GPB14		MF_GPB13		MF_GPB12	
23	22	21	20	19	18	17	16
MF_GPB11		MF_GPB10		MF_GPB9		MF_GPB8	
15	14	13	12	11	10	9	8
MF_GPB7		MF_GPB6		MF_GPB5		MF_GPB4	
7	6	5	4	3	2	1	0
MF_GPB3		MF_GPB2		MF_GPB1		MF_GPB0	

Bits	Descriptions				
[31:30]	MF_GPB15	GPIOB[15] Multi Function			
		Pin Name	MF_GPB15		
		LPCLK	2'b00	2'b01	2'b10
			GPIOB[15]	Reserved	LPCLK
[29:28]	MF_GPB14	GPIOB[14] Multi Function			
		Pin Name	MF_GPB14		
		ISDA	2'b00	2'b01	2'b10
			GPIOB[14]	LMVSYNC	ISDA
[27:26]	MF_GPB13	GPIOB[13] Multi Function			
		Pin Name	MF_GPB13		
		ISCK	2'b00	2'b01	2'b10
			GPIOB[13]	Reserved	ISCK
[25:24]	MF_GPB12	GPIOB[12] Multi Function			
		Pin Name	MF_GPB12		
		SPDATA[7]	2'b00	2'b01	2'b10
			GPIOB[12]	SPI1_DO	LVDATA[23]
[23:22]	MF_GPB11	GPIOB[11] Multi Function			
		Pin Name	MF_GPB11		
		SPDATA[6]	2'b00	2'b01	2'b10
			GPIOB[11]	SPI1_DI	LVDATA[22]

[21:20]	MF_GPB10	GPIOB[10] Multi Function				
		Pin Name	MF_GPB10			
		SPDATA[5]	2'b00	2'b01	2'b10	2'b11
			GPIOB[10]	SPI1_CS0_	LVDATA[21]	SPDATA[5]
[19:18]	MF_GPB9	GPIOB[9] Multi Function				
		Pin Name	MF_GPB9			
		SPDATA[4]	2'b00	2'b01	2'b10	2'b11
			GPIOB[9]	SPI1_CLK	LVDATA[20]	SPDATA[4]
[17:16]	MF_GPB8	GPIOB[8] Multi Function				
		Pin Name	MF_GPB8			
		SPDATA[3]	2'b00	2'b01	2'b10	2'b11
			GPIOB[8]	Reserved	LVDATA[19]	SPDATA[3]
[15:14]	MF_GPB7	GPIOB[7] Multi Function				
		Pin Name	MF_GPB7			
		SPDATA[2]	2'b00	2'b01	2'b10	2'b11
			GPIOB[7]	Reserved	LVDATA[18]	SPDATA[2]
[13:12]	MF_GPB6	GPIOB[6] Multi Function				
		Pin Name	MF_GPB6			
		SPDATA[1]	2'b00	2'b01	2'b10	2'b11
			GPIOB[6]	I2S_DIN	SPDATA[1]	SPDATA[1]
[11:10]	MF_GPB5	GPIOB[5] Multi Function				
		Pin Name	MF_GPB5			
		SPDATA[0]	2'b00	2'b01	2'b10	2'b11
			GPIOB[5]	I2S_DOUT	SDDAT1[2]	SPDATA[0]
[9:8]	MF_GPB4	GPIOB[4] Multi Function				
		Pin Name	MF_GPB4			
		SFIELD	2'b00	2'b01	2'b10	2'b11
			GPIOB[4]	I2S_WS	SDDAT1[3]	SFIELD
[7:6]	MF_GPB3	GPIOB[3] Multi Function				
		Pin Name	MF_GPB3			
		SVSYNC	2'b00	2'b01	2'b10	2'b11
			GPIOB[3]	I2S_BCLK	SDCMD1	SVSYNC
[5:4]	MF_GPB2	GPIOB[2] Multi Function				
		Pin Name	MF_GPB2			
		SHSYNC	2'b00	2'b01	2'b10	2'b11
			GPIOB[2]	I2S_MCLK	SDCLK1	SHSYNC

[3:2]	MF_GPB1	GPIOB[1] Multi Function				
		Pin Name	MF_GPB1			
		SPCLK	2'b00	2'b01	2'b10	2'b11
GPIOB[1]	UHL_DM1		SDDAT1[0]	SPCLK		
[1:0]	MF_GPB0	GPIOB[0] Multi Function				
		Pin Name	MF_GPB0			
		SCLKO	2'b00	2'b01	2'b10	2'b11
GPIOB[0]	UHL_DP1		SDDAT1[1]	SCLKO		

GPIO C Multi Function Control Register (GPCFUN)

Note: The pull-up enable of the pins shared with GPIOC is controlled by register GPIOC_PUEN directly. Users have to set the GPIOC_PUEN properly before the chip entered the standby mode.

Register	Address	R/W	Description	Reset Value
GPCFUN	GCR_BA+0x88	R/W	GPIO C Multi Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MF_GPC15		MF_GPC14		MF_GPC13		MF_GPC12	
23	22	21	20	19	18	17	16
MF_GPC11		MF_GPC10		MF_GPC9		MF_GPC8	
15	14	13	12	11	10	9	8
MF_GPC7		MF_GPC6		MF_GPC5		MF_GPC4	
7	6	5	4	3	2	1	0
MF_GPC3		MF_GPC2		MF_GPC1		MF_GPC0	

Bits	Descriptions				
[31:30]	MF_GPC15	GPIOC[15] Multi Function			
		Pin Name	MF_GPC15		
		LVDATA[15]	2'b00	2'b01	2'b10
			GPIOC[15]	SPDATA[7]	KPI_SI[7]
[29:28]	MF_GPC14	GPIOC[14] Multi Function			
		Pin Name	MF_GPC14		
		LVDATA[14]	2'b00	2'b01	2'b10
			GPIOC[14]	SPDATA[6]	KPI_SI[6]
[27:26]	MF_GPC13	GPIOC[13] Multi Function			
		Pin Name	MF_GPC13		
		LVDATA[13]	2'b00	2'b01	2'b10
			GPIOC[13]	SPDATA[5]	KPI_SI[5]
[25:24]	MF_GPC12	GPIOC[12] Multi Function			
		Pin Name	MF_GPC12		
		LVDATA[12]	2'b00	2'b01	2'b10
			GPIOC[12]	SPDATA[4]	KPI_SI[4]
[23:22]	MF_GPC11	GPIOC[11] Multi Function			
		Pin Name	MF_GPC11		
		LVDATA[11]	2'b00	2'b01	2'b10
			GPIOC[11]	SPDATA[3]	KPI_SI[3]

[21:20]	MF_GPC10	GPIOC[10] Multi Function				
		Pin Name	MF_GPC10			
		LVDATA[10]	2'b00	2'b01	2'b10	2'b11
			GPIOC[10]	SPDATA[2]	KPI_SI[2]	LVDATA[10]
[19:18]	MF_GPC9	GPIOC[9] Multi Function				
		Pin Name	MF_GPC9			
		LVDATA[9]	2'b00	2'b01	2'b10	2'b11
			GPIOC[9]	SPDATA[1]	KPI_SI[1]	LVDATA[9]
[17:16]	MF_GPC8	GPIOC[8] Multi Function				
		Pin Name	MF_GPC8			
		LVDATA[8]	2'b00	2'b01	2'b10	2'b11
			GPIOC[8]	SPDATA[0]	KPI_SI[0]	LVDATA[8]
[15:14]	MF_GPC7	GPIOC[7] Multi Function				
		Pin Name	MF_GPC7			
		LVDATA[7]	2'b00	2'b01	2'b10	2'b11
			GPIOC[7]	Reserved	LVDATA[7]	LVDATA[7]
[13:12]	MF_GPC6	GPIOC[6] Multi Function				
		Pin Name	MF_GPC6			
		LVDATA[6]	2'b00	2'b01	2'b10	2'b11
			GPIOC[6]	Reserved	LVDATA[6]	LVDATA[6]
[11:10]	MF_GPC5	GPIOC[5] Multi Function				
		Pin Name	MF_GPC5			
		LVDATA[5]	2'b00	2'b01	2'b10	2'b11
			GPIOC[5]	Reserved	LVDATA[5]	LVDATA[5]
[9:8]	MF_GPC4	GPIOC[4] Multi Function				
		Pin Name	MF_GPC4			
		LVDATA[4]	2'b00	2'b01	2'b10	2'b11
			GPIOC[4]	Reserved	LVDATA[4]	LVDATA[4]
[7:6]	MF_GPC3	GPIOC[3] Multi Function				
		Pin Name	MF_GPC3			
		LVDATA[3]	2'b00	2'b01	2'b10	2'b11
			GPIOC[3]	Reserved	LVDATA[3]	LVDATA[3]
[5:4]	MF_GPC2	GPIOC[2] Multi Function				
		Pin Name	MF_GPC2			
		LVDATA[2]	2'b00	2'b01	2'b10	2'b11
			GPIOC[2]	Reserved	LVDATA[2]	LVDATA[2]

[3:2]	MF_GPC1	GPIOC[1] Multi Function				
		Pin Name	MF_GPC1			
		LVDATA[1]	2'b00	2'b01	2'b10	2'b11
	GPIOC[1]		Reserved	LVDATA[1]	LVDATA[1]	
[1:0]	MF_GPC0	GPIOC[0] Multi Function				
		Pin Name	MF_GPC0			
		LVDATA[0]	2'b00	2'b01	2'b10	2'b11
	GPIOC[0]		Reserved	LVDATA[0]	LVDATA[0]	

GPIO D Multi Function Control Register (GPDFUN)

Note: The pull-up enable of the pins shared with GPIO D is controlled by register GPIO D_PUEN directly. Users have to set the GPIO D_PUEN properly before the chip entered the standby mode.

Register	Address	R/W	Description	Reset Value
GPDFUN	GCR_BA+0x8C	R/W	GPIO D Multi Function Control Register	0xxx00_03FF

31	30	29	28	27	26	25	24
MF_GPD15		MF_GPD14		MF_GPD13		MF_GPD12	
23	22	21	20	19	18	17	16
MF_GPD11		MF_GPD10		MF_GPD9		MF_GPD8	
15	14	13	12	11	10	9	8
MF_GPD7		MF_GPD6		MF_GPD5		MF_GPD4	
7	6	5	4	3	2	1	0
MF_GPD3		MF_GPD2		MF_GPD1		MF_GPD0	

Bits	Descriptions				
[31:30]	MF_GPD15	GPIO D[15] Multi Function			
		Pin Name	MF_GPD15		
		SPI0_DO	2'b00	2'b01	2'b10
			GPIO D[15]	Reserved	SPI0_DO
[29:28]	MF_GPD14	GPIO D[14] Multi Function			
		Pin Name	MF_GPD14		
		SPI0_DI	2'b00	2'b01	2'b10
			GPIO D[14]	Reserved	SPI0_DI
[27:26]	MF_GPD13	GPIO D[13] Multi Function			
		Pin Name	MF_GPD13		
		SPI0_CS0_	2'b00	2'b01	2'b10
			GPIO D[13]	Reserved	SPI0_CS0_
[25:24]	MF_GPD12	GPIO D[12] Multi Function			
		Pin Name	MF_GPD12		
		SPI0_CLK	2'b00	2'b01	2'b10
			GPIO D[12]	Reserved	SPI0_CLK
[23:22]	MF_GPD11	GPIO D[11] Multi Function			
		Pin Name	MF_GPD11		
		LVDE	2'b00	2'b01	2'b10
			GPIO D[11]	Reserved	LVDE
[21:20]	MF_GPD10	GPIO D[10] Multi Function			
		Pin Name	MF_GPD10		
		LVSYNC	2'b00	2'b01	2'b10
			GPIO D[10]	Reserved	LVSYNC

[19:18]	MF_GPD9	GPIO[9] Multi Function				
		Pin Name	MF_GPD9			
		LHSYNC	2'b00	2'b01	2'b10	2'b11
			GPIO[9]	Reserved	LHSYNC	LHSYNC
[17:16]	MF_GPD8	GPIO[8] Multi Function				
		Pin Name	MF_GPD8			
		NWR_	2'b00	2'b01	2'b10	2'b11
			GPIO[8]	SDCMD2	NWR_	NWR_
[15:14]	MF_GPD7	GPIO[7] Multi Function				
		Pin Name	MF_GPD7			
		NRE_	2'b00	2'b01	2'b10	2'b11
			GPIO[7]	SDCLK2	NRE_	NRE_
[13:12]	MF_GPD6	GPIO[6] Multi Function				
		Pin Name	MF_GPD6			
		NBUSY1_	2'b00	2'b01	2'b10	2'b11
			GPIO[6]	SDDAT2[2]	NBUSY1_	NBUSY1_
[11:10]	MF_GPD5	GPIO[5] Multi Function				
		Pin Name	MF_GPD5			
		NBUSY0_	2'b00	2'b01	2'b10	2'b11
			GPIO[5]	SDDAT2[3]	NBUSY0_	NBUSY0_
[9:8]	MF_GPD4	GPIO[4] Multi Function				
		Pin Name	MF_GPD4			
		TRST_	2'b00	2'b01	2'b10	2'b11
			GPIO[4]	HUR_RTS	SPI0_CS1_	TRST_
[7:6]	MF_GPD3	GPIO[3] Multi Function				
		Pin Name	MF_GPD3			
		TDO	2'b00	2'b01	2'b10	2'b11
			GPIO[3]	HUR_CTS	PWM3	TDO
[5:4]	MF_GPD2	GPIO[2] Multi Function				
		Pin Name	MF_GPD2			
		TDI	2'b00	2'b01	2'b10	2'b11
			GPIO[2]	HUR_RXD	PWM2	TDI
[3:2]	MF_GPD1	GPIO[1] Multi Function				
		Pin Name	MF_GPD1			
		TMS	2'b00	2'b01	2'b10	2'b11
			GPIO[1]	HUR_TXD	PWM1	TMS
[1:0]	MF_GPD0	GPIO[0] Multi Function				
		Pin Name	MF_GPD0			
		TCK	2'b00	2'b01	2'b10	2'b11
			GPIO[0]	SPI1_CS1_	PWM0	TCK

GPIO E Multi Function Control Register (GPEFUN)

Note: The pull-up enable of the pins shared with GPIOE is controlled by register GPIOE_PUEN directly. Users have to set the GPIOE_PUEN properly before the chip entered the standby mode.

Register	Address	R/W	Description	Reset Value
GPEFUN	GCR_BA+0x90	R/W	GPIO E Multi Function Control Register	0x0000_0XX0

31	30	29	28	27	26	25	24
Reserved		Reserved		Reserved		Reserved	
23	22	21	20	19	18	17	16
MF_GPE11		MF_GPE10		MF_GPE9		MF_GPE8	
15	14	13	12	11	10	9	8
MF_GPE7		MF_GPE6		MF_GPE5		MF_GPE4	
7	6	5	4	3	2	1	0
MF_GPE3		MF_GPE2		MF_GPE1		MF_GPE0	

Bits	Descriptions					
[31:24]	Reserved	Reserved This field is reserved and keep all these bits in zero is necessary.				
[23:22]	MF_GPE11	GPIOE[11] Multi Function				
		Pin Name	MF_GPE11			
		NCLE	2'b00	2'b01	2'b10	2'b11
	GPIOE[11]		Reserved	NCLE	NCLE	
[21:20]	MF_GPE10	GPIOE[10] Multi Function				
		Pin Name	MF_GPE10			
		NALE	2'b00	2'b01	2'b10	2'b11
	GPIOE[10]		Reserved	NALE	NALE	
[19:18]	MF_GPE9	GPIOE[9] Multi Function				
		Pin Name	MF_GPE9			
		NCS1_	2'b00	2'b01	2'b10	2'b11
	GPIOE[9]		SDDAT2[0]	NCS1_	NCS1_	
[17:16]	MF_GPE8	GPOE[8] Multi Function				
		Pin Name	MF_GPE8			
		NCS0_	2'b00	2'b01	2'b10	2'b11
	GPIOE[8]		SDDAT2[1]	NCS0_	NCS0_	
[15:14]	MF_GPE7	GPIOE[7] Multi Function				
		Pin Name	MF_GPE7			
		SDCLK	2'b00	2'b01	2'b10	2'b11
	GPIOE[7]		Reserved	SDCLK	SDCLK	

[13:12]	MF_GPE6	GPIOE[6] Multi Function				
		Pin Name	MF_GPE6			
		SDCMD	2'b00	2'b01	2'b10	2'b11
GPIOE[6]	Reserved		SDCMD	SDCMD		
[11:10]	MF_GPE5	GPIOE[5] Multi Function				
		Pin Name	MF_GPE5			
		SDDAT[3]	2'b00	2'b01	2'b10	2'b11
GPIOE[5]	Reserved		SDDAT[3]	SDDAT[3]		
[9:8]	MF_GPE4	GPIOE[4] Multi Function				
		Pin Name	MF_GPE4			
		SDDAT[2]	2'b00	2'b01	2'b10	2'b11
GPIOE[4]	Reserved		SDDAT[2]	SDDAT[2]		
[7:6]	MF_GPE3	GPIOE[3] Multi Function				
		Pin Name	MF_GPE3			
		SDDAT[1]	2'b00	2'b01	2'b10	2'b11
GPIOE[3]	Reserved		SDDAT[1]	SDDAT[1]		
[5:4]	MF_GPE2	GPIOE[2] Multi Function				
		Pin Name	MF_GPE2			
		SDDAT[0]	2'b00	2'b01	2'b10	2'b11
GPIOE[2]	Reserved		SDDAT[0]	SDDAT[0]		
[3:2]	MF_GPE1	GPIOE[1] Multi Function				
		Pin Name	MF_GPE1			
		LVDATA[17]	2'b00	2'b01	2'b10	2'b11
GPIOE[1]	SVSYNC		LVDATA[17]	LVDATA[17]		
[1:0]	MF_GPE0	GPIOE[0] Multi Function				
		Pin Name	MF_GPE0			
		LVDATA[16]	2'b00	2'b01	2'b10	2'b11
GPIOE[0]	SHSYNC		LVDATA[16]	LVDATA[16]		

Miscellaneous Multi Function Control Register (MISFUN)

Register	Address	R/W	Description	Reset Value
MISFUN	GCR_BA+0x94	R/W	Miscellaneous Multi Function Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							MF_I2S

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	MF_I2S	I2S I/F Functional Selection This field is to control the I2S interface is used by I2S Controller or SPU.
		Pin Name MF_I2S
		I2S Interface 1'b0 1'b1
		I2S SPU

Miscellaneous Pins Control Register (MISCP CR)

Register	Address	R/W	Description	Reset Value
MISCP CR	GCR_BA+0xA0	R/W	Miscellaneous Pins Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SL_MD	SL_MA	SL_MCTL	SL_MCLK	DS_MD	DS_MA	DS_MCTL	DS_MCLK

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7]	SL_MD	MD Pins Slew Rate Control This bit control the output slew rate of 16 MD pins. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[6]	SL_MA	MA Pins Slew Rate Control This bit control the output slew rate of 13 MA and 3 MBA pins. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[5]	SL_MCTL	Memory I/F Control Pins Slew Rate Control This bit control the output slew rate of MCKE, MCS0_, MCS1_, MRAS_, MCAS_, MWE_, MDQM, MDQS0 and MDQS1 pins. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[4]	SL_MCLK	MCLK Pin Slew Rate Control This bit control the output slew rate of MCLK and MCLK_ pins. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[3]	DS_MD	MD Pins Driving Strength Control This bit controls the output driving strength rate of 16 MD pins. 1'b0: Output driving strength is 8mA (Class I Buffer). 1'b1: Output driving strength is 24mA (Class II Buffer).
[2]	DS_MA	MA Pins Driving Strength Control This bit controls the output driving strength of 13 MA and 3 MBA pins. 1'b0: Output driving strength is 8mA (Class I Buffer). 1'b1: Output driving strength is 24mA (Class II Buffer).
[1]	DS_MCTL	Memory I/F Control Pins Driving Strength Control This bit controls the output driving strength of MCKE, MCS0_, MCS1_, MRAS_, MCAS_, MWE_, MDQM, MDQS0 and MDQS1 pins. 1'b0: Output driving strength is 8mA (Class I Buffer). 1'b1: Output driving strength is 24mA (Class II Buffer).

[0]	DS_MCLK	MCLK Pin Driving Strength Control This bit controls the output driving strength of MCLK and MCLK_ pins. 1'b0: Output driving strength is 8mA (Class I Buffer). 1'b1: Output driving strength is 24mA (Class II Buffer).

Multi Function Pin Note

In this chip, some functional signals could be assigned to multiple pins. Normally, each functional signal could only be assigned to a pin. This section describes how the functional signal is assigned while the setting makes a functional signal be assigned to multiple pins simultaneously.

While the direction of functional signal is **output** and setting makes this functional signal is assigned to multiple pins simultaneously, this functional signal will be outputted to multiple pins directly.

While the direction of functional signal is **input** and setting makes this functional signal is from multiple pins simultaneously, a hardwired priority will be used to choose which pin is used for functional signal.

The following is the detail information for the functional signals.

Output Functional Signals

Functional Signal	SPI0_CS1_	
Direction	Output	
Pin Name	TRST_	GPA[8]
Priority	Equal	Equal

Functional Signal	SPI1_CS1_		
Direction	Output		
Pin Name	URTXD	TRST_	GPA[10]
Priority	Equal	Equal	Equal

Functional Signal	WDT_RST_		
Direction	Output		
Pin Name	ISCK	LVDE	GPA[7]
Priority	Equal	Equal	Equal

Input Functional Signals

Functional Signal	LMVSYNC		
Direction	Input		
Pin Name	URRXD	ISDA	GPA[2]
Priority	High	Middle	Low

Bi-Direction Functional Signals

Functional Signal	PWM0	
Direction	In/Out	
Pin Name	TCK	SPCLK
Out Priority	Equal	Equal
In Priority	High	Low

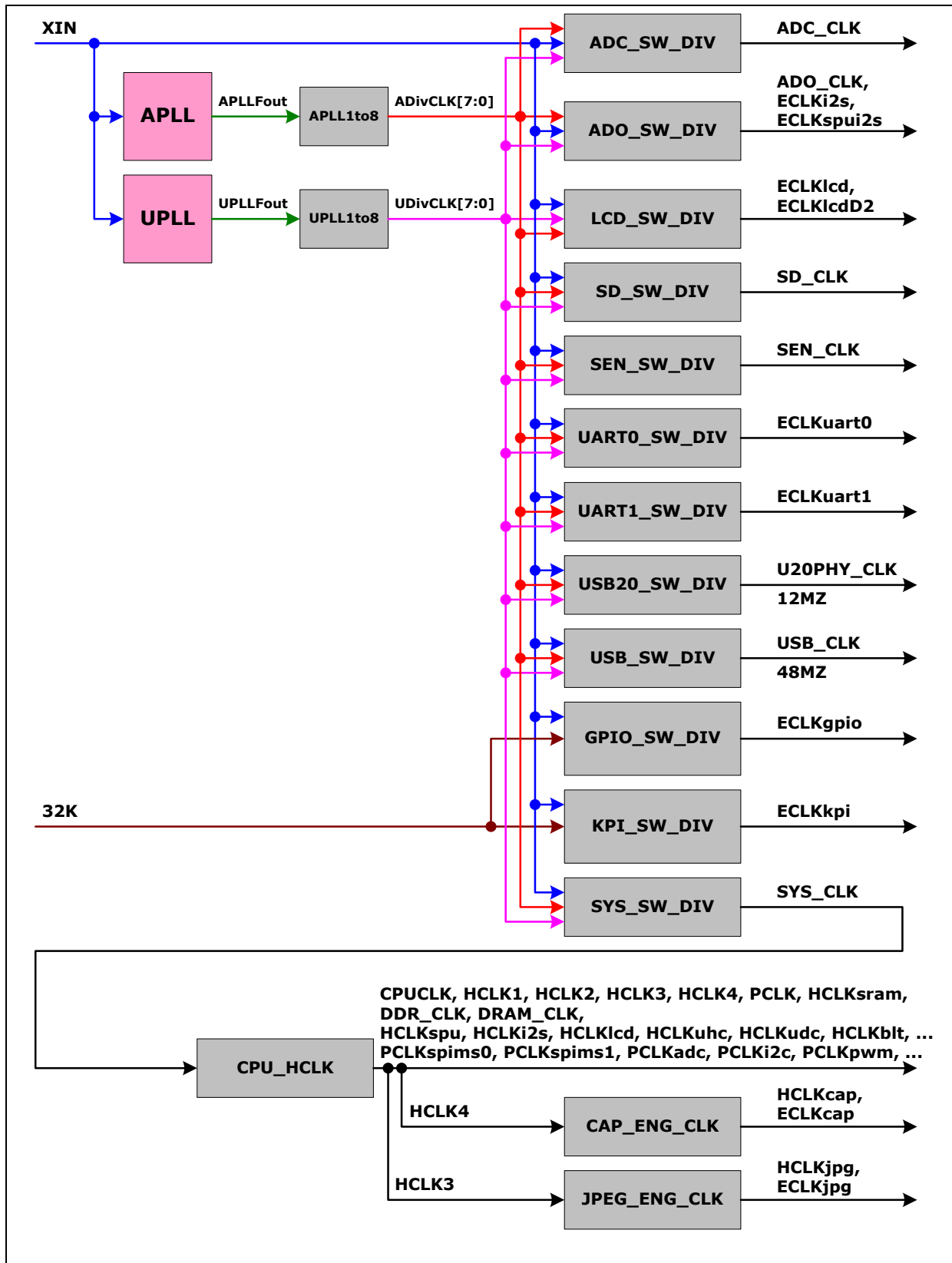
5.3 Clock Controller(CLK)

5.3.1 Overview

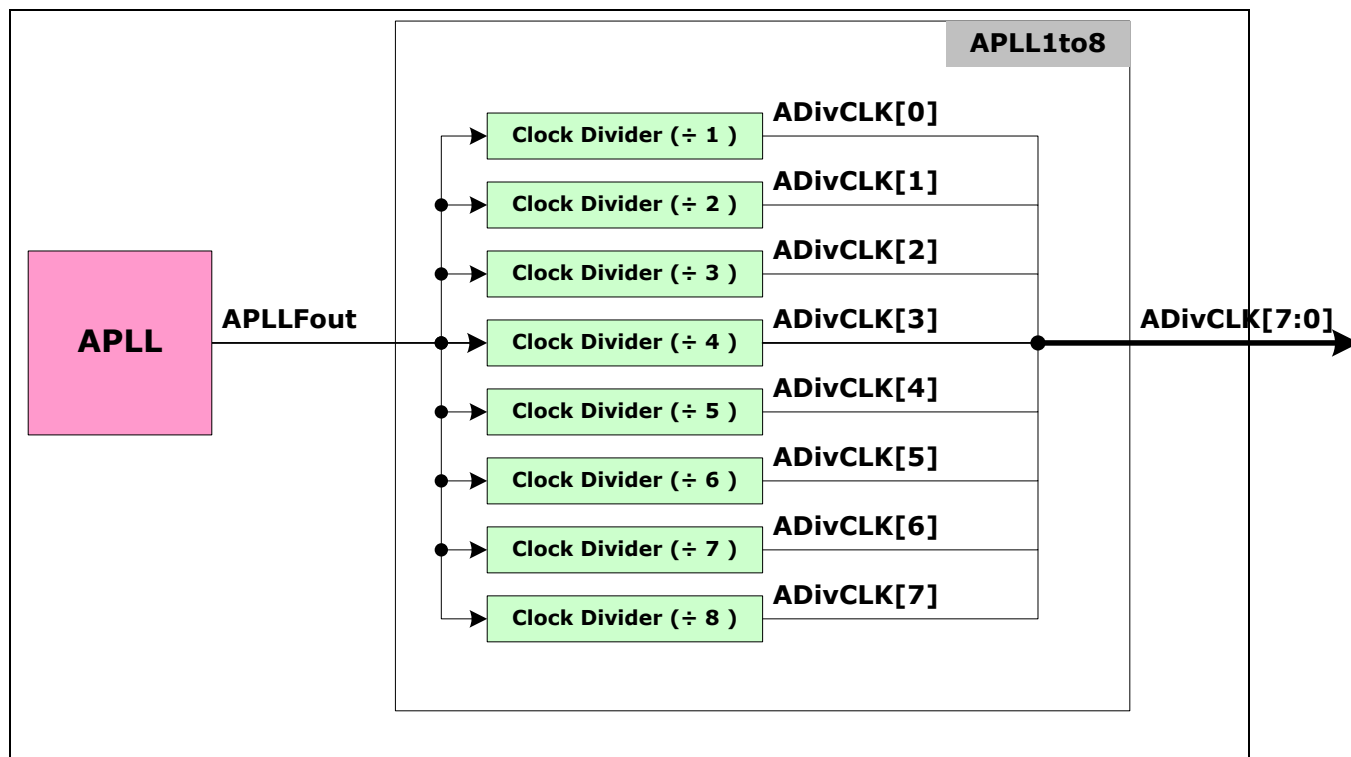
The clock controller generates the clocks for the whole chip, it include all of IPs on AHB, APB and engine clock like USB, UART and so on. There are a PLL in this chip, and the PLL clock source is from the external crystal input. It also implements the power control function, include the individually clock on or off control register, clock source selector and divider. These functions minimize the extra power consumption and the chip run on the only just condition. On the power down mode the controller turn off the crystal oscillator to minimize the chip power consumption.

5.3.2 Block diagram

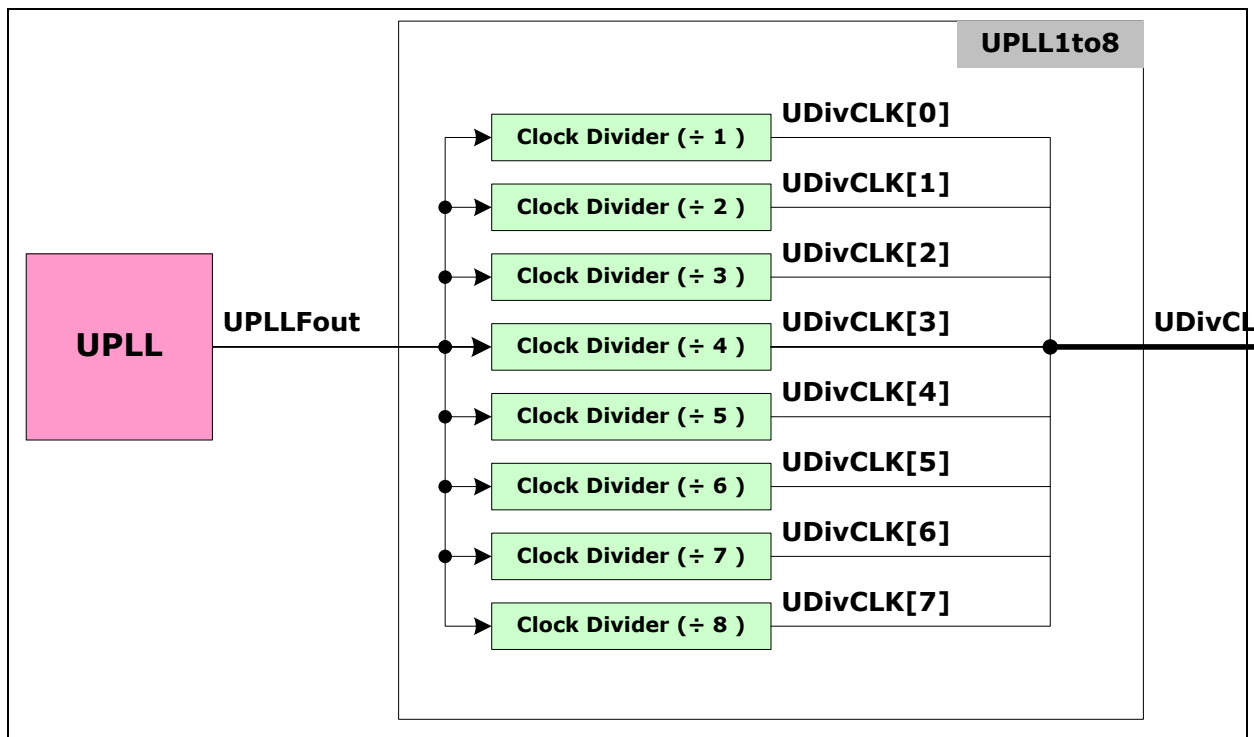
5.3.3 Clock Controller Overview



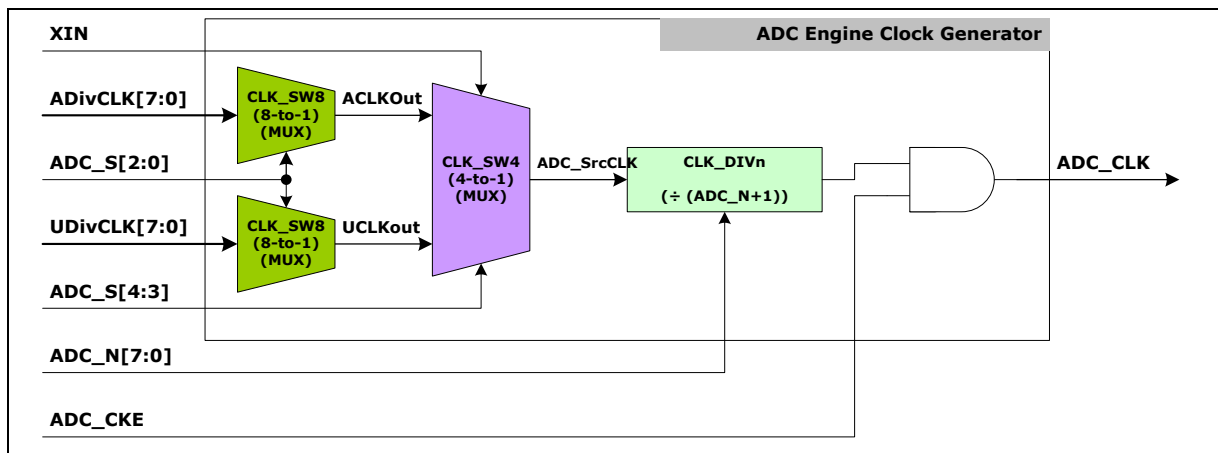
5.3.3.1 APLL1to8



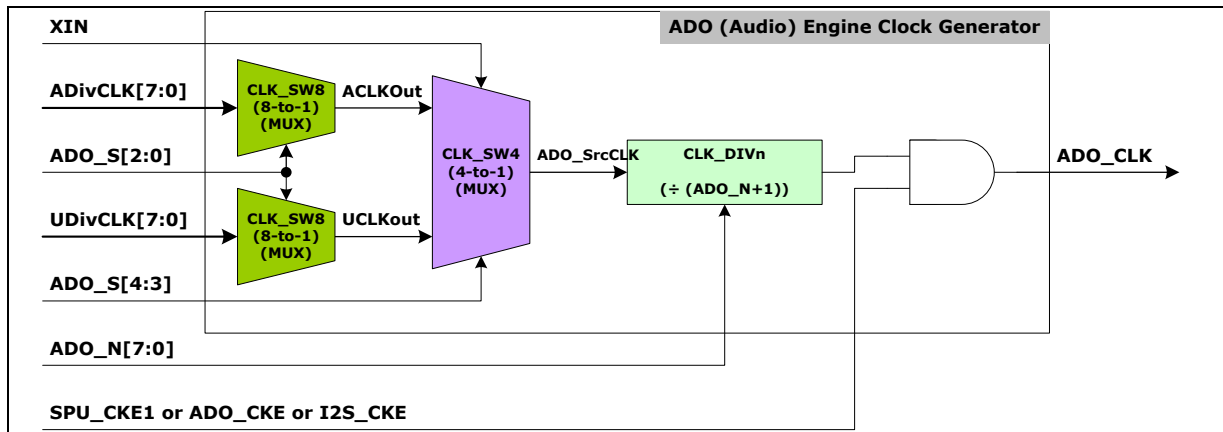
5.3.3.2 UPLL1to8



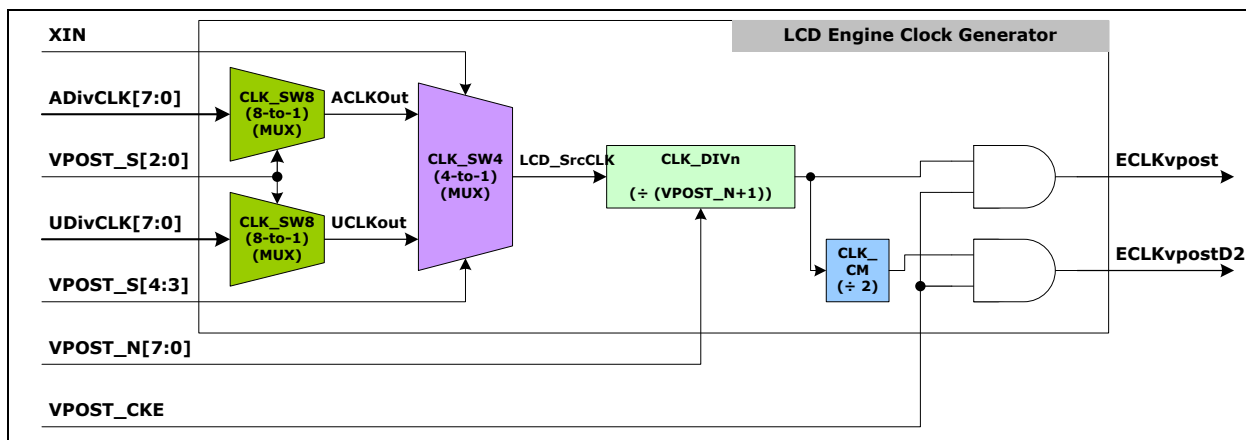
5.3.3.3 ADC_SW_DIV



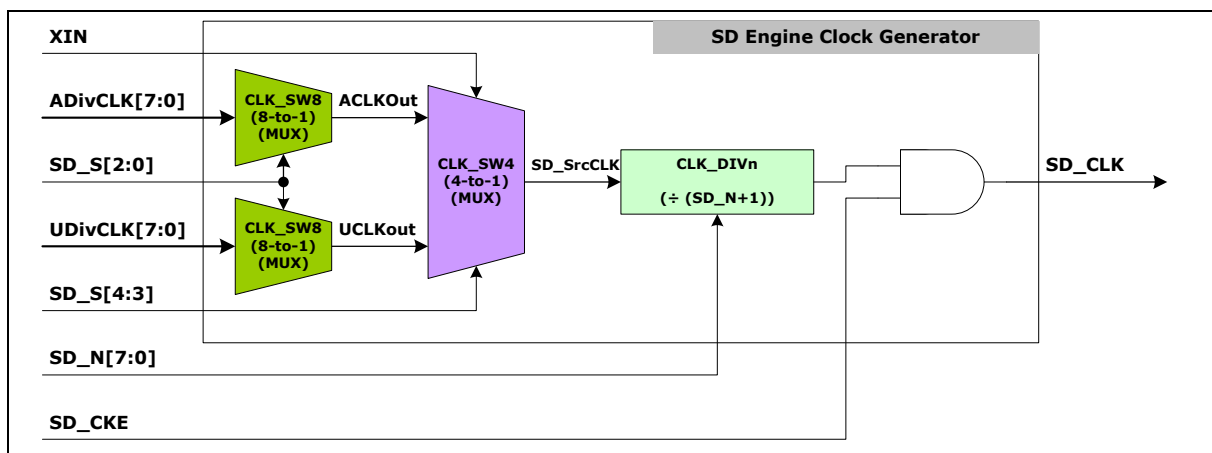
5.3.3.4 ADO_SW_DIV



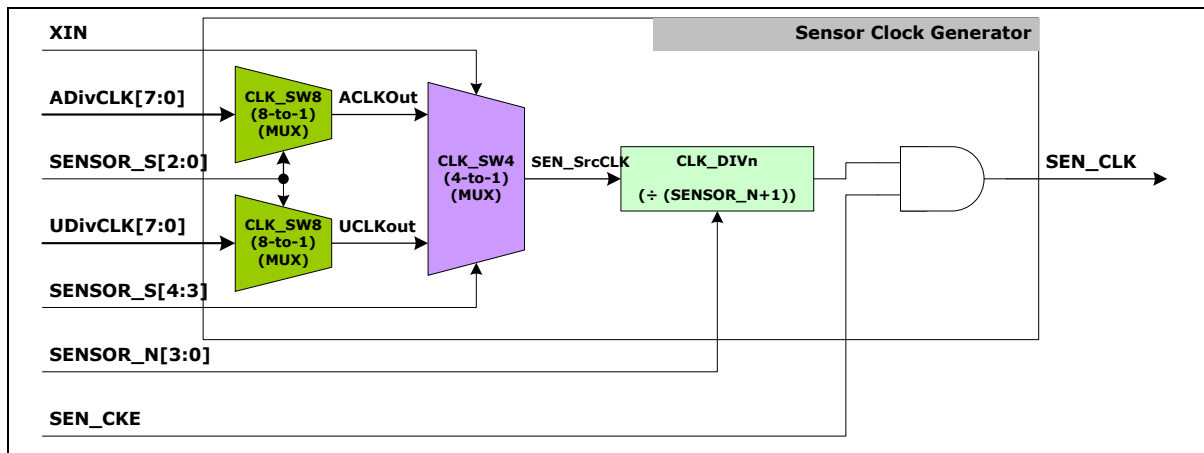
5.3.3.5 LCD_SW_DIV



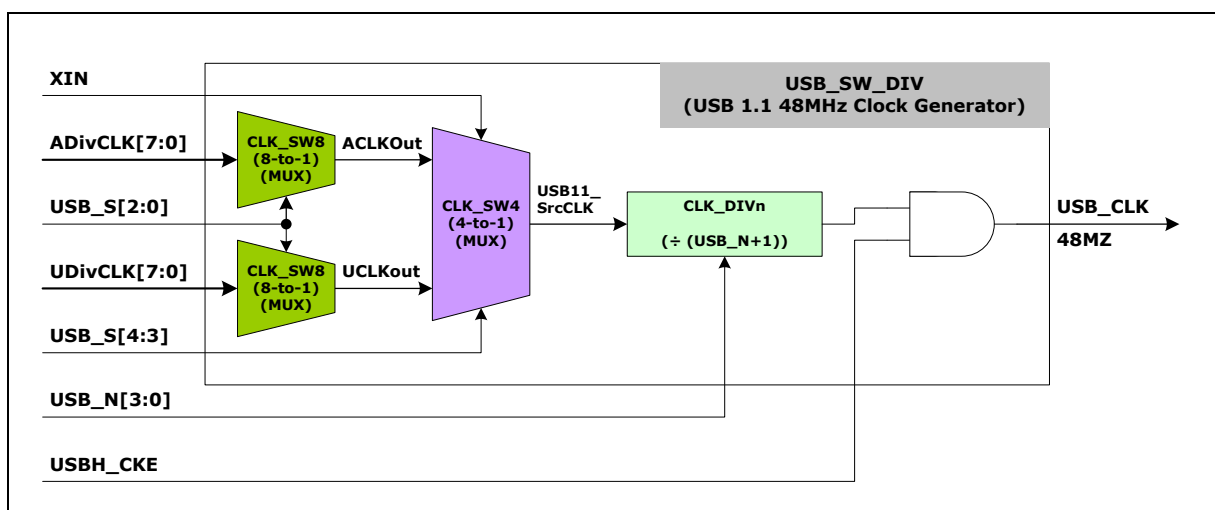
5.3.3.6 SD_SW_DIV



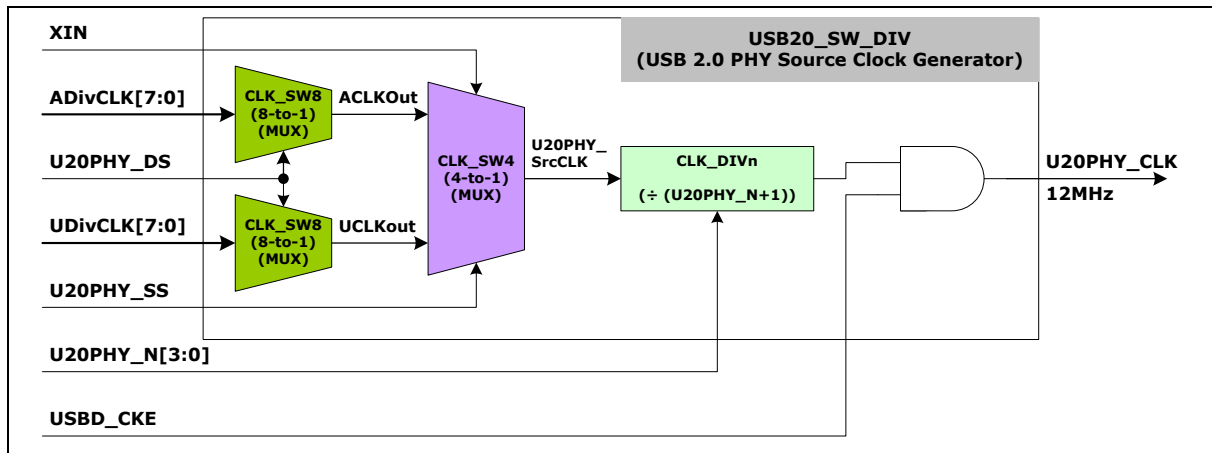
5.3.3.7 SEN_SW_DIV



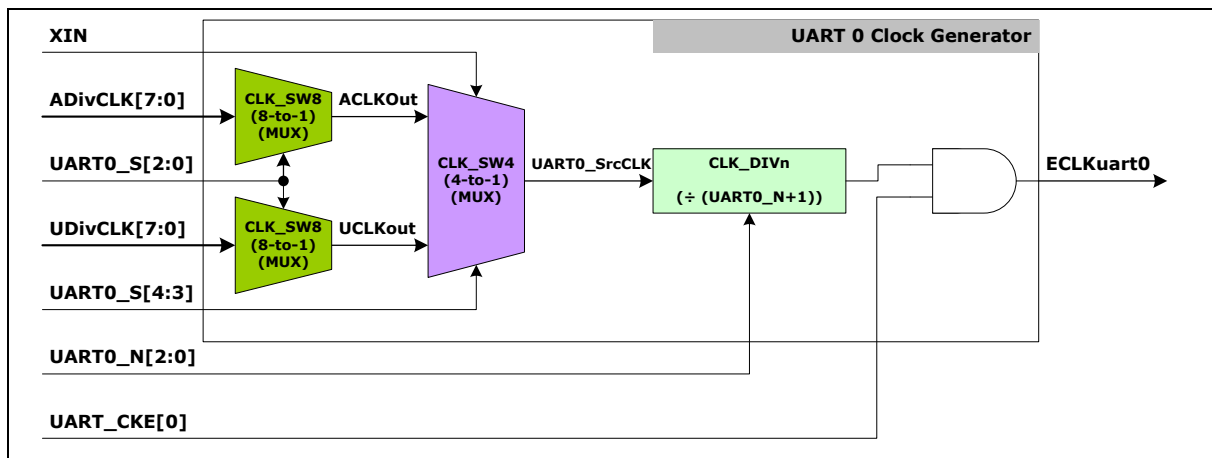
5.3.3.8 USB_SW_DIV



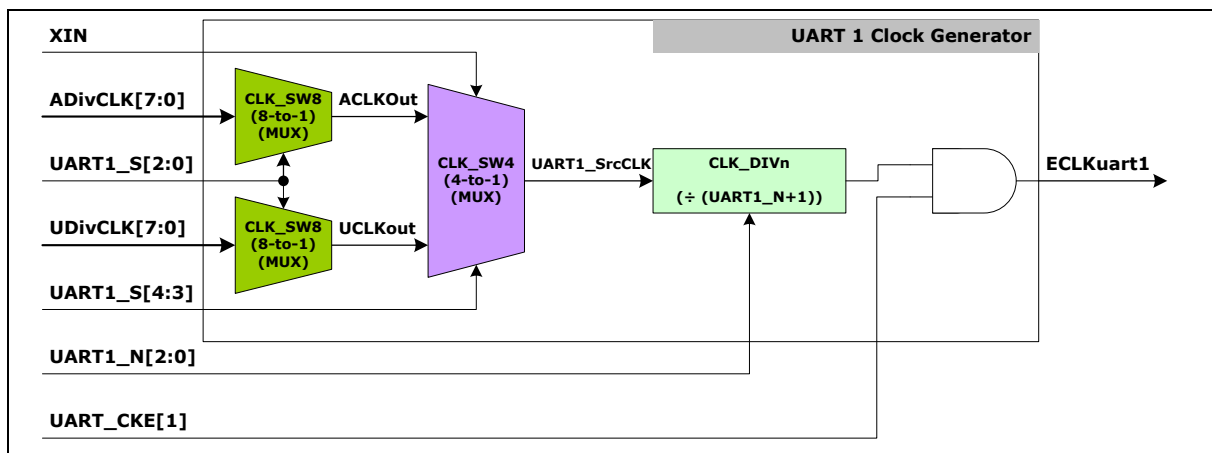
5.3.3.9 USB20_SW_DIV



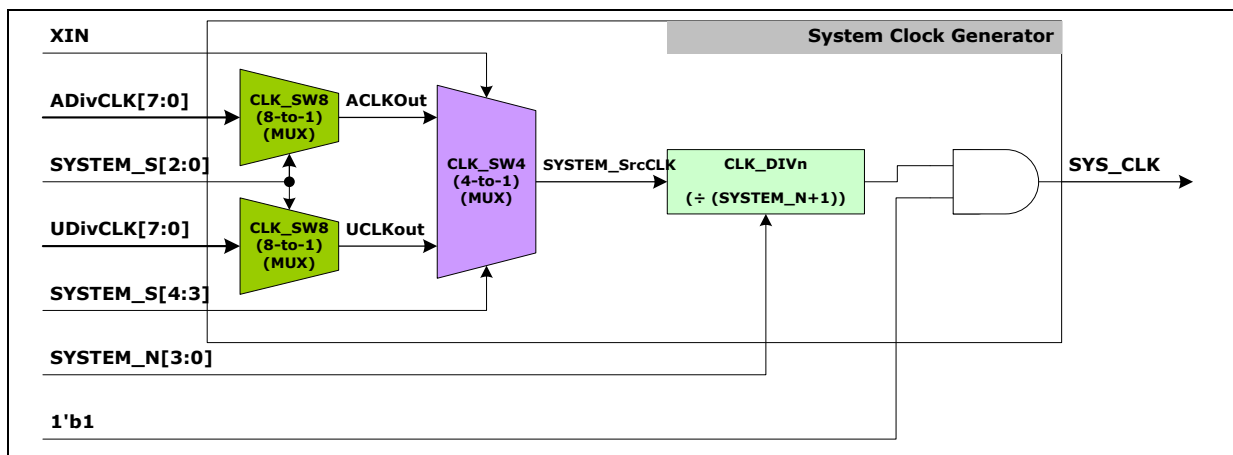
5.3.3.10 UART0_SW_DIV



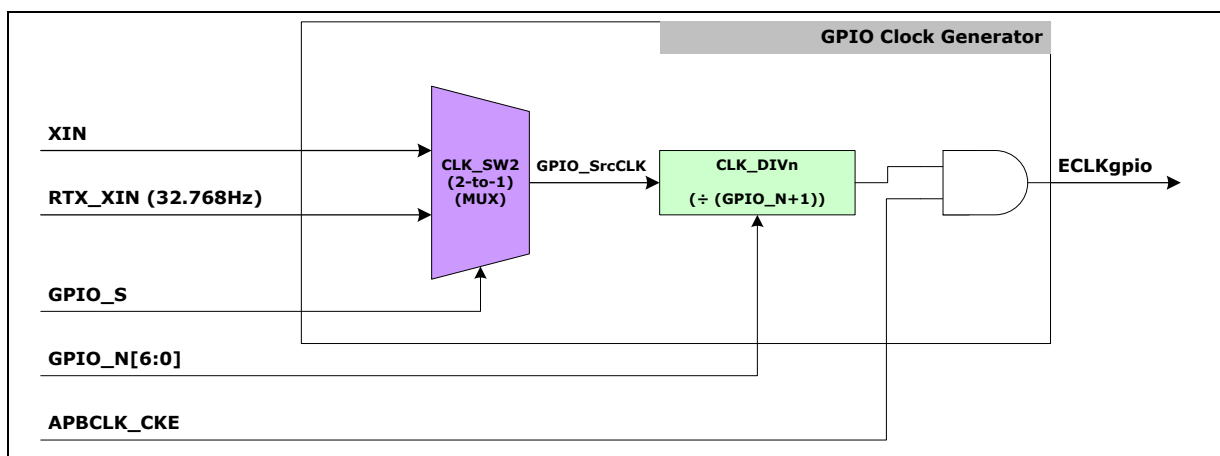
5.3.3.11 UART1_SW_DIV



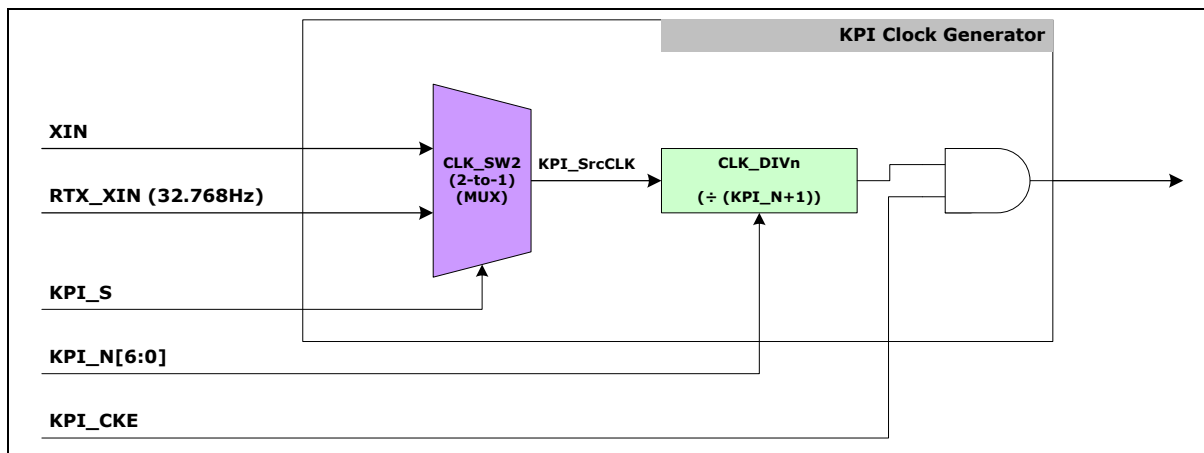
5.3.3.12 SYS_SW_DIV

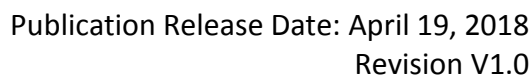


5.3.3.13 GPIO_SW_DIV



5.3.3.14 KPI_SW_DIV





The System Clock is the source of the AMBA bus, CPU and synchronous engine modules. The AMBA clock includes 5 AHB clock and 1 APB clock, HCLK, HCLK1, HCLK2, HCLK3, HCLK4 and the PCLK. The HCLK is divided by two from the system clock and the clock rate of HCLK is equal to or faster than the HCLK1 and HCLK2, HCLK3 and HCLK4. The HCLK is used for the AHB-bridge, to synchronize the AHB1 and AHB2. The CPU clock is used for ARM CPU and the frequency can be double higher or slower than the HCLK.

The DRAM_CLK is used for clock output to external SDRAM device. The DDR_CLK is used for SDRAM controller to sample the data of DDR/DDR2/LPDDR SDRAM device.

The HCLK1 clock source is from system clock and controlled by CPU clock divider (CPU_N) to keep the HCLK1 lower than CPU clock. If the CPU_N is 0x1, the HCLK1 will be the same with CPU clock. If CPU_N is not 0x1, the HCLK1 will be the half of CPU clock.

The HCLK2, HCLK3 and HCLK4 with the same frequency but with dependent on/off control register. They are divided from the HCLK and used for the AHB bus controller.

The PCLK is divided from the HCLK1 and is used as the APB bus clock. The clock of each APB bus peripherals are from PCLK with individual on/off control bit.

5.3.4 Control Registers

Register	Address	R/W	Description	Reset Value
PWRCON	CLK_BA + 00	R/W	System Power Down Control Register	0x00FF_FF03
AHBCLK	CLK_BA + 04	R/W	Clock Enable Control Register	0x0000_011F
APBCLK	CLK_BA + 08	R/W	Clock Enable Control Register	0x0000_8100
CLKDIV0	CLK_BA + 0C	R/W	Clock Divider Number Register 0	0x0000_0000
CLKDIV1	CLK_BA + 10	R/W	Clock Divider Number Register 1	0x0000_0000
CLKDIV2	CLK_BA + 14	R/W	Clock Divider Number Register 2	0x0000_0000
CLKDIV3	CLK_BA + 18	R/W	Clock Divider Number Register 3	0x0000_0000
CLKDIV4	CLK_BA + 1C	R/W	Clock Divider Number Register 4	0x0000_0000
APLLCON	CLK_BA + 20	R/W	APLL Control Register	0x0001_C02E
UPLLCON	CLK_BA + 24	R/W	UPLL Control Register	0x0001_C02E
CLK_TREG	CLK_BA + 30	R/W	TEST Clock Control Register	0x0000_0000

Power Down Control Register (PWRCON)

The chip clock source is from an external crystal. The crystal oscillator can be control on/off by the register XTAL_EN. When turn off the crystal, the chip into power down state. To avoid outputting an unstable clock to system, clock controller implements a pre-scalar counter. After the clock counter count pre-scalar x 256 crystal cycle, the clock controller starts to output the clock to system.

Register	Address	R/W	Description	Reset Value
PWRCON	CLK_BA + 00	R/W	System Power Down Control Register	0x00FF_FF03

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Pre-Scalar[15:8]							
15	14	13	12	11	10	9	8
Pre-Scalar[7:0]							
7	6	5	4	3	2	1	0
Reserved			SEN_OFF_ST	INT_EN	INTSTS	XIN_CTL	XTAL_EN

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:8]	Pre-Scalar	Pre-Scalar counter Assume the crystal is stable after the Pre-Scalar x 256 crystal cycles. Clock controller wouldn't output clock to system before the counter reaching (pre-scalar x 256).
[7:5]	Reserved	Reserved
[4]	SEN_OFF_ST	Sensor clock level on clock off state 0 = sensor clock keep on low level 1 = sensor clock keep on high level
[3]	INT_EN	Power On Interrupt Enable 0 = Disable 1 = Enable. The interrupt will occur when the Crystal enable signal (XTAL_EN) change from LOW to HIGH.
[2]	INTSTS	Power Down interrupt status Read 0 = Normal 1 = Indicate crystal enable change from low to high, the chip is resume from power down state. Write 0 = No action. 1 = Clear interrupt

[1]	XIN_CTL	Crystal pre-divide control for Wake-up from power down mode. The chip will delay 256 x pre-scalar cycles after the reset signal to wait the Crystal to stable. 0 = Disable the pre-scalar, assume the crystal is stable 1 = Enable the pre-scalar counter
[0]	XTAL_EN	Crystal (Power Down) Control 0: Crystal off (Power down) 1: Crystal on (Normal operation)

AHB Devices Clock Enable Control Register (AHBCLK)

These register bits are used to enable/disable clock for AMBA clock, AHB engine and peripheral

Register	Addressca	R/W	Description	Reset Value
AHBCLK	CLK_BA + 04	R/W	AHB IPs Clock Enable Control Register	0x0000_011F

31	30	29	28	27	26	25	24
SPU_CKE1	ADO_CKE	SEN_CKE	CAP_CKE	VPOST_CKE	I2S_CKE	SPU_CKE	HCLK4_CKE
23	22	21	20	19	18	17	16
SD_CKE	NAND_CKE	SIC_CKE	Reserved	Reserved	USB_D_CKE	USBH_CKE	HCLK3_CKE
15	14	13	12	11	10	9	8
Reserved	EDMA4_CKE	EDMA3_CKE	EDMA2_CKE	EDMA1_CKE	EDMA0_CKE	Reserved	HCLK1_CKE
7	6	5	4	3	2	1	0
JPG_CKE	FSC_CKE	BLT_CKE	DRAM_CKE	SRAM_CKE	HCLK_CKE	APBCLK_CKE	CPU_CKE

Bits	Descriptions	
[31]	SPU_CKE1	SPU's I2S Interface Clock Enable Control This bit is to enable the clock for I2S interface of SPU. 0 = Disable 1 = Enable
[30]	ADO_CKE	Audio DAC Engine Clock Enable Control 0 = Disable 1 = Enable
[29]	SEN_CKE	Sensor Interface Clock Enable Control 0 = Disable 1 = Enable
[28]	CAP_CKE	Capture Clock Enable Control (Also is Capture engine clock enable control) 0 = Disable 1 = Enable
[27]	VPOST_CKE	VPOST Clock Enable Control (Also is VPOST engine clock enable control) 0 = Disable 1 = Enable
[26]	I2S_CKE	I2S Controller Clock Enable Control 0 = Disable 1 = Enable
[25]	SPU_CKE	SPU Clock Enable Control 0 = Disable 1 = Enable

[24]	HCLK4_CKE	HCLK4 Clock Enable Control 0 = Disable 1 = Enable
[23]	SD_CKE	SD Card Controller Engine Clock Enable Control 0 = Disable 1 = Enable
[22]	NAND_CKE	NAND Controller Clock Enable Control 0 = Disable 1 = Enable
[21]	SIC_CKE	SIC Clock Enable Control 0 = Disable 1 = Enable
[20]	Reserved	Reserved
[19]	Reserved	Reserved
[18]	USBD_CKE	USB Device Clock Enable Control 0 = Disable 1 = Enable
[17]	USBH_CKE	USB Host Controller Clock Enable Control 0 = Disable 1 = Enable
[16]	HCLK3_CKE	HCLK3 Clock Enable Control 0 = Disable 1 = Enable
[15]	Reserved	Reserved
[14]	EDMA4_CKE	EDMA Controller Channel 4 Clock Enable Control 0 = Disable 1 = Enable
[13]	EDMA3_CKE	EDMA Controller Channel 3 Clock Enable Control 0 = Disable 1 = Enable
[12]	EDMA2_CKE	EDMA Controller Channel 2 Clock Enable Control 0 = Disable 1 = Enable
[11]	EDMA1_CKE	EDMA Controller Channel 1 Clock Enable Control 0 = Disable 1 = Enable
[10]	EDMA0_CKE	EDMA Controller Channel 0 Clock Enable Control 0 = Disable 1 = Enable
[9]	Reserved	Reserved

[8]	HCLK1_CKE	HCLK1 Clock Enable Control. 0 = Disable 1 = Enable
[7]	JPG_CKE	JPEG Codec Clock Enable Control This bit is the clock enabling control for both the system clock (HCLKjpg) and the engine clock (ECLKjpg) of JPEG codec. 0 = Disable 1 = Enable
[6]	FSC_CKE	Frame Switch Controller Clock Enable Control 0 = Disable 1 = Enable
[5]	BLT_CKE	BitBLT Accelerator Clock Enable Control This bit is the clock enabling control for clock (HCLKblt) 0 = Disable 1 = Enable
[4]	DRAM_CKE	SDRAM and SDRAM Controller Clock Enable Control. 0 = Disable 1 = Enable
[3]	SRAM_CKE	SRAM Controller Clock Enable Control. 0 = Disable 1 = Enable
[2]	HCLK_CKE	HCLK Clock Enable Control. (This clock is used for DRAM controller, SRAM controller and AHB-to-AHB bridge) 0 = Disable 1 = Enable
[1]	APBCLK_CKE	APB Clock Enable Control. 0 = Disable 1 = Enable
[0]	CPU_CKE	CPU Clock Enable Control 0 = Disable 1 = Enable

APB Devices Clock Enable Control Register (APBCLK)

These register bits are used to enable/disable clock for APB engine and peripheral.

Register	Address	R/W	Description	Reset Value
APBCLK	CLK_BA + 08	R/W	APB IPs Clock Enable Control Register	0x0000_8100

31	30	29	28	27	26	25	24
Reserved						KPI_CKE	TIC_CKE
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
WDCLK_CKE	Reserved					TIMER_CKE	
7	6	5	4	3	2	1	0
SPIMS1_CKE	SPIMS0_CKE	PWM_CKE	UART_CKE		RTC_CKE	I2C_CKE	ADC_CKE

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25]	KPI_CKE	KPI Clock Enable Control 0 = Disable 1 = Enable
[24]	TIC_CKE	TIC Clock Enable Control 0 = Disable 1 = Enable
[23:16]	Reserved	Reserved
[15]	WDCLK_CKE	Watch Dog Clock Enable Control (Also is Watch Dog engine clock enable control) 0 = Disable 1 = Enable Note: The Watch Dog engine clock source ONLY is from the external crystal input.
[14:10]	Reserved	Reserved
[9]	TIMER_CKE	Timer1 Clock Enable Control 0 = Disable 1 = Enable
[8]		Timer0 Clock Enable Control 0 = Disable 1 = Enable Note: 1. The Timer clock engine source ONLY is from the external crystal input. 2. Timer APB clock will be enabled when Timer0 or Timer1 is enabled.
[7]	SPIMS1_CKE	SPIMS1 (Master / Slave) Clock Enable Control 0 = Disable 1 = Enable

[6]	SPIMS0_CKE	SPIMS0 (Master / Slave) Clock Enable Control 0 = Disable 1 = Enable
[5]	PWM_CKE	PWM Clock Enable Control 0 = Disable 1 = Enable
[4]	UART_CKE	UART1 Clock Enable Control 0 = Disable 1 = Enable
[3]		UART0 Clock Enable Control 0 = Disable 1 = Enable Note: UART APB clock will be enabled when UART0 or UART1 is enabled.
[2]	RTC_CKE	RTC Clock Enable Control (NOT X32K clock enable control) 0 = Disable 1 = Enable
[1]	I2C_CKE	I2C Clock Enable Control 0 = Disable 1 = Enable
[0]	ADC_CKE	ADC Clock Enable Control (Also is ADC engine clock enable control) 0 = Disable 1 = Enable

Clock Divider Register 0 (CLKDIV0)

Before clock switch the related clock sources (pre-select and new-select) must be turn on.

Register	Address	R/W	Description	Reset Value
CLKDIV0	CLK_BA + 0C	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				SENSOR_N			
23	22	21	20	19	18	17	16
KPI_N[6:4]				SENSOR_S			
15	14	13	12	11	10	9	8
KPI_N[3:0]				SYSTEM_N			
7	6	5	4	3	2	1	0
Reserved		KPI_S		SYSTEM_S			

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27:24]	SENSOR_N	Sensor Clock Divide This field defines the clock divide number for clock divider to generate the sensor clock. The actual clock divide number is (SENSOR_N + 1). So, $SEN\_CLK = SEN\_SrcCLK / (SENSOR\_N + 1)$
[23:21]	KPI_N[6:4]	KPI Engine Clock Divider Bits [6:4] This field defines the bits [6:4] of clock divide number for clock divider to generate the engine clock for KPI. The actual clock divide number is ({KPI_N [6:5], KPI_N [3:0]} + 1). So, $ECLKkpi = KPI\_SrcCLK / ({KPI\_N [6:5], KPI\_N [3:0]} + 1)$
[20:19]	SENSOR_S	Sensor Clock Source Selection This field selects which clock is used to be the source of sensor clock. 00: SEN_SrcCLK = XIN 01: SEN_SrcCLK = Reserved 10: SEN_SrcCLK = ACLKOut 11: SEN_SrcCLK = UCLKOut

[18:16]		Sensor Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. And, this field only takes effect while the <i>SENSOR_S[4:3]</i> is 2'b10 (APLL) or 2'b11 (UPLL). 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8
[15:12]	KPI_N[3:0]	KPI Engine Clock Divider Bits [3:0] This field defines the bits [3:0] of clock divide number for clock divider to generate the engine clock for KPI. The actual clock divide number is ({KPI_N [6:5], KPI_N [3:0]} + 1). So, $ECLK_{kpi} = KPI\_SrcCLK / (\{KPI\_N [6:5], KPI\_N [3:0]\} + 1)$
[11:8]	SYSTEM_N	System Clock Divide This field defines the clock divide number for clock divider to generate the system clock SYS_CLK. The actual clock divide number is (SYSTEM_N + 1). So, $SYS\_CLK = SYSTEM\_SrcCLK / (SYSTEM\_N + 1)$
[7:6]	Reserved	Reserved
[5]	KPI_S	KPI Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for KPI. 0: KPI_SrcCLK = XIN 1: KPI_SrcCLK = X32K
[4:3]	SYSTEM_S	System Clock Source Selection This field selects which clock is used to be the source of system clock SYS_CLK. 00: SYSTEM_SrcCLK = XIN 01: SYSTEM_SrcCLK = Reserved 10: SYSTEM_SrcCLK = ACLKOut 11: SYSTEM_SrcCLK = UCLKOut
[2:0]		System Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. And, this field only takes effect while the <i>SYSTEM_S[4:3]</i> is 2'b10 (APLL) or 2'b11 (UPLL). 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8

Clock Divider Register 1 (CLKDIV1)

Register	Address	R/W	Description	Reset Value
CLKDIV1	CLK_BA_ + 10	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
ADO_N							
23	22	21	20	19	18	17	16
Reserved			ADO_S				
15	14	13	12	11	10	9	8
VPOST_N							
7	6	5	4	3	2	1	0
Reserved			VPOST_S				

Bits	Descriptions	
[31:24]	ADO_N	Audio DAC Engine Clock Divide This field defines the clock divide number for clock divider to generate the engine clock for Audio-DAC. The actual clock divide number is (ADO_N + 1). So, $ADO\_CLK = ADO\_SrcCLK / (ADO\_N + 1)$
[23:21]	Reserved	Reserved
[20:19]	ADO_S	Audio-DAC Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for Audio-DAC. 00: ADO_SrcCLK = XIN 01: ADO_SrcCLK = Reserved 10: ADO_SrcCLK = ACLKOut 11: ADO_SrcCLK = UCLKOut
[18:16]		Audio-DAC Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. And, this field only takes effect while the ADO_S[4:3] is 2'b10 (APLL) or 2'b11 (UPLL). 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8
[15:8]	VPOST_N	VPOST Engine Clock Divide This field defines the clock divide number for clock divider to generate the engine clock for VPOST. The actual clock divide number is (VPOST_N + 1). So, $ECLKvpost = LCD\_SrcCLK / (VPOST\_N + 1)$

[7:5]	Reserved	Reserved
[4:3]	VPOST_S	VPOST Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for VPOST. 00: LCD_SrcCLK = XIN 01: LCD_SrcCLK = Reserved 10: LCD_SrcCLK = ACLKOut 11: LCD_SrcCLK = UCLKOut
[2:0]		VPOST Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. <i>And, this field only takes effect while the VPOST_S[4:3] is 2'b10 (APLL) or 2'b11 (UPLL).</i> 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8

Clock Divider Register 2 (CLKDIV2)

Register	Address	R/W	Description	Reset Value
CLKDIV2	CLK_BA_ + 14	R/W	Clock Divider Register 2	0x0000_0000

31	30	29	28	27	26	25	24
SD_N							
23	22	21	20	19	18	17	16
Reserved	U20PHY_SS		SD_S				
15	14	13	12	11	10	9	8
U20PHY_N				USB_N			
7	6	5	4	3	2	1	0
U20PHY_DS			USB_S				

Bits	Descriptions	
[31:24]	SD_N	SD Engine Clock Divide This field defines the clock divide number for clock divider to generate the engine clock for SD controller. The actual clock divide number is (SD_N + 1). So, $SD\_CLK = SD\_SrcCLK / (SD\_N + 1)$
[23]	Reserved	Reserved
[22:21]	U20PHY_SS	USB20 PHY Source Clock Selection This field selects which clock is used to be the source of 12MHz clock for embedded USB 2.0 PHY. 00: U20PHY_SrcCLK = XIN 01: U20PHY_SrcCLK = Reserved 10: U20PHY_SrcCLK = ACLKOut 11: U20PHY_SrcCLK = UCLKOut
[20:19]	SD_S	SD Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for SD controller. 00: SD_SrcCLK = XIN 01: SD_SrcCLK = Reserved 10: SD_SrcCLK = ACLKOut 11: SD_SrcCLK = UCLKOut

[18:16]		SD Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. And, this field only takes effect while the <i>SD_S[4:3]</i> is 2'b10 (APLL) or 2'b11 (UPLL). 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8
[15:12]	U20PHY_N	USB20 PHY Clock Divider This field defines the clock divide number for clock divider to generate the 12MHz clock for embedded USB 2.0 PHY. The actual clock divide number is (U20PHY_N + 1). So, $U20PHY\_CLK = U20PHY\_SrcCLK / (U20PHY\_N + 1)$ Note: The U20PHY_CLK must be 12MHz.
[11:8]	USB_N	USB 1.1 Host Controller Engine Clock Divider This field defines the clock divide number for clock divider to generate the 48MHz clock for USB 1.1 host controller. The actual clock divide number is (USB_N + 1). So, $USB\_CLK = USB11\_SrcCLK / (USB\_N + 1)$ Note: The USB_CLK must be 48MHz.
[7:5]	U20PHY_DS	USB20 PHY Source Clock Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. And, this field only takes effect while the <i>U20PHY_SS</i> is 2'b10 (APLL) or 2'b11 (UPLL). 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8
[4:3]	USB_S	USB 1.1 Engine Clock Source Selection This field selects which clock is used to be the source of 48MHz clock for USB 1.1 host controller. 00: USB11_SrcCLK = XIN 01: USB11_SrcCLK = Reserved 10: USB11_SrcCLK = ACLKOut 11: USB11_SrcCLK = UCLKOut

[2:0]	USB 1.1 Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. And, this field only takes effect while the USB_S[4:3] is 2'b10 (APLL) or 2'b11 (UPLL). 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8

Clock Divider Register 3 (CLKDIV3)

Register	Address	R/W	Description	Reset Value
CLKDIV3	CLK_BA_ + 18	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
ADC_N							
23	22	21	20	19	18	17	16
Reserved			ADC_S				
15	14	13	12	11	10	9	8
UART1_N			UART1_S				
7	6	5	4	3	2	1	0
UART0_N			UART0_S				

Bits	Descriptions	
[31:24]	ADC_N	ADC Engine Clock Divider This field defines the clock divide number for clock divider to generate the engine clock for ADC. The actual clock divide number is (ADC_N + 1). So, $ADC\_CLK = ADC\_SrcCLK / (ADC\_N + 1)$
[23:21]	Reserved	Reserved
[20:19]	ADC_S	ADC Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for ADC controller. 00: ADC_SrcCLK = XIN 01: ADC_SrcCLK = Reserved 10: ADC_SrcCLK = ACLKOut 11: ADC_SrcCLK = UCLKOut
[18:16]		ADC Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. <i>And, this field only takes effect while the ADC_S[4:3] is 2'b10 (APLL) or 2'b11 (UPLL).</i> 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8
[15:13]	UART1_N	UART1 Engine Clock Divider This field defines the clock divide number for clock divider to generate the engine clock for UART1. The actual clock divide number is (UART1_N + 1). So, $ECLKuart1 = UART1\_SrcCLK / (UART1\_N + 1)$

[12:11]	UART1_S	UART1 Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for UART1 controller. 00: UART1_SrcCLK = XIN 01: UART1_SrcCLK = Reserved 10: UART1_SrcCLK = ACLKOut 11: UART1_SrcCLK = UCLKOut
[10:8]		UART1 Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. <i>And, this field only takes effect while the UART1_S[4:3] is 2'b10 (APLL) or 2'b11 (UPLL).</i> 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8
[7:5]	UART0_N	UART0 Engine Clock Divider This field defines the clock divide number for clock divider to generate the engine clock for UART0. The actual clock divide number is (UART0_N + 1). So, $ECLKuart0 = UART0\_SrcCLK / (UART0\_N + 1)$
[4:3]	UART0_S	UART0 Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for UART0 controller. 00: UART0_SrcCLK = XIN 01: UART0_SrcCLK = Reserved 10: UART0_SrcCLK = ACLKOut 11: UART0_SrcCLK = UCLKOut
[2:0]		UART0 Engine Clock Source Divide Selection This field selects the source clock divide number while the source clock is from APLL or UPLL. <i>And, this field only takes effect while the UART0_S[4:3] is 2'b10 (APLL) or 2'b11 (UPLL).</i> 000: ACLKOut(or UCLKOut) = APLL(or UPLL) 001: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 2 010: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 3 011: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 4 100: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 5 101: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 6 110: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 7 111: ACLKOut(or UCLKOut) = APLL(or UPLL) ÷ 8

Clock Divider Register 4 (CLKDIV4)

Register	Address	R/W	Description	Reset Value
CLKDIV4	CLK_BA_ + 1C	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					JPG_N		
23	22	21	20	19	18	17	16
GPIO_N							GPIO_S
15	14	13	12	11	10	9	8
Reserved	CAP_N			APB_N			
7	6	5	4	3	2	1	0
HCLK234_N				CPU_N			

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:24]	JPG_N	JPEG Engine Clock Divider This field defines the clock divide number for clock divider to generate the engine clock for JPEG codec. The actual clock divide number is (JPG_N + 1). So, $ECLK_{jpg} = HCLK3 / (JPG\_N + 1)$
[23:17]	GPIO_N	GPIO Engine Clock Divider This field defines the clock divide number for clock divider to generate the engine clock for GPIO controller. The actual clock divide number is (GPIO_N + 1). So, $ECLK_{gpio} = GPIO\_SrcCLK / (GPIO\_N + 1)$
[16]	GPIO_S	GPIO Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for GPIO controller. 0 = XIN 1 = X32K
[15]	Reserved	Reserved
[14:12]	CAP_N	Capture Engine Clock Divider This field defines the clock divide number for clock divider to generate the engine clock for capture. The actual clock divide number is (CAP_N + 1). So, $ECLK_{cap} = HCLK4 / (CAP\_N + 1)$
[11:8]	APB_N	APB Clock Divider This field defines the clock divide number for clock divider to generate the PCLK for APB bus and controllers in APB bus. The actual clock divide number is (APB_N + 1). So, $PCLK = HCLK1 / (APB\_N + 1)$

[7:4]	HCLK234_N	AHB234 Clock Divider This field defines the clock divide number for clock divider to generate the HCLK for AHB2, AHB3, AHB4 bus and controllers in AHB2, AHB3 and AHB4 bus. The actual clock divide number is (APB_N + 1). So, $\text{HCLK2} = \text{HCLK} / (\text{HCLK234\_N} + 1)$ $\text{HCLK3} = \text{HCLK} / (\text{HCLK234\_N} + 1)$ $\text{HCLK4} = \text{HCLK} / (\text{HCLK234\_N} + 1)$
[3:0]	CPU_N	CPU Clock Divider This field defines the clock divide number for clock divider to generate the CPUCLK for ARM926EJ-S CPU. The actual clock divide number is (CPU_N + 1). So, $\text{CPUCLK} = \text{SYS\_CLK} / (\text{CPU\_N} + 1)$ Note: The CPU_N only could be set as 0x0, 0x1. Other values are prohibited.

APLL Control Register (APLLCON)

Register	Address	R/W	Description	Reset Value
APLLCON	CLK_BA + 20	R/W	APLL Control Register	0x0001_C02E

31	30	29	28	27	26	25	24
Reserved		WAIT_CYCLE[9:4]					
23	22	21	20	19	18	17	16
WAIT_CYCLE[3:0]				PLL_CHANGE_Fen	OE	BP	PD
15	14	13	12	11	10	9	8
OUT_DV		IN_DV					FB_DV
7	6	5	4	3	2	1	0
FB_DV							

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:20]	WAIT_CYCLE	Wait Cycle Used together with PLL_CHANGE_Fen, please refer to the PLL_CHANGE_Fen description on next field.
[19]	PLL_CHANGE_Fen	APLL Change Frequency Control 0 = APLL updates parameters (bit[15:0]) on next cycle (default) 1 = APLL updates parameters (bit[15:0]) on next cycle (on the condition HCLK/PCLK = 2/1), and DRAM controller will do nothing until internal counter counts to the same value with WAIT_CYCLE[9:0] for MCLK re-lock for DLL of DDR-SDRAM.
[18]	OE	PLL OE (Fout enable) pin Control 0 = PLL FOUT enable 1 = PLL FOUT is fixed low
[17]	BP	PLL ByPass Control 0 = PLL at Normal mode (Default) 1 = By pass Fin (i.e. Fout = XIN)
[16]	PD	Power Down Mode 0 = PLL at Normal mode (Default) 1 = PLL at power-down mode
[15:14]	OUT_DV	PLL Output Divider Control This field controls the output divider (OD) value for PLL. The formula between this field and output divider is as following: $OutputDivider = 2^{OUT\_DV}$ This field connected to pin OD of PLL directly.

[13:9]	IN_DV	PLL Input Divider Control This field controls the input divider value for PLL. The formula between this field and input divider is as following: $InputDivider = IN\_DV + 2$ This field connected to pin N of PLL directly.
[8:0]	FB_DV	PLL Feedback Divider Control This field controls the feedback divider value for PLL. The formula between this field and feedback divider is as following: $FeedbackDivider = DB\_DV + 2$ This field connected to pin M of PLL directly.

Output Clock Frequency Setting

$$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

Constrain:

1. $1MHz < \frac{FIN}{NR} < 15MHz$
3. $100MHz \leq FOUT \times NO \leq 500MHz$

FOUT	Output Clock Frequency
FIN	Input (Reference) Clock Frequency
NR	Input Divider (IN_DV + 2)
NF	Feedback Divider (FB_DV + 2)
NO	OUT_DV = "00" : NO = 1 OUT_DV = "01" : NO = 2 OUT_DV = "10" : NO = 2 OUT_DV = "11" : NO = 4

Default Setting

The default value: 0xC02E

FIN = 12 MHz

NR = (0+2) = 2

NF = (46+2) = 48

NO = 4

FOUT = 12 x 48/2 x 1/4 = 72 MHz

UPLL Control Register (UPLLCON)

Register	Address	R/W	Description	Reset Value
UPLLCON	CLK_BA + 24	R/W	UPLL Control Register	0x0001_C02E

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					OE	BP	PD
15	14	13	12	11	10	9	8
OUT_DV		IN_DV					FB_DV
7	6	5	4	3	2	1	0
FB_DV							

Bits	Descriptions	
[31:19]	Reserved	Reserved
[18]	OE	PLL OE (Fout enable) pin Control 0 = PLL FOUT enable 1 = PLL FOUT is fixed low
[17]	BP	PLL ByPass Control 0 = PLL at Normal mode (Default) 1 = By pass Fin (i.e. Fout = XIN)
[16]	PD	Power Down Mode 0 = PLL at Normal mode (Default) 1 = PLL at power-down mode
[15:14]	OUT_DV	PLL Output Divider Control This field controls the output divider (OD) value for PLL. The formula between this field and output divider is as following: $OutputDivider = 2^{OUT\_DV}$ This field connected to pin OD of PLL directly.
[13:9]	IN_DV	PLL Input Divider Control This field controls the input divider value for PLL. The formula between this field and input divider is as following: $InputDivider = IN\_DV + 2$ This field connected to pin N of PLL directly.
[8:0]	FB_DV	PLL Feedback Divider Control This field controls the feedback divider value for PLL. The formula between this field and feedback divider is as following: $FeedbackDivider = DB\_DV + 2$ This field connected to pin M of PLL directly.

Output Clock Frequency Setting

$$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

Constrain:

1. $1MHz < \frac{FIN}{NR} < 15MHz$
3. $100MHz \leq FOUT \times NO \leq 500MHz$

<i>FOUT</i>	Output Clock Frequency
<i>FIN</i>	Input (Reference) Clock Frequency
<i>NR</i>	Input Divider (IN_DV + 2)
<i>NF</i>	Feedback Divider (FB_DV + 2)
<i>NO</i>	OUT_DV = "00" : NO = 1 OUT_DV = "01" : NO = 2 OUT_DV = "10" : NO = 2 OUT_DV = "11" : NO = 4

Default Setting

The default value of this register is according to the XIN frequency selection from power-on setting

The default value: 0xC02E

FIN = 12 MHz

NR = (0+2) = 2

NF = (46+2) = 48

NO = 4

$FOUT = 12 \times 48/2 \times 1/4 = 72 \text{ MHz}$

Test Clock Register (CLK_TREG)

Register	Address	R/W	Description	Reset Value
CLK_TREG	CLK_BA + 30	R/W	Test Clock Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TEST_CKE	SW_CLK	TEST_SEL					

Bits	Descriptions	
[31:18]	Reserved	Reserved
[7]	TEST_CKE	Test clock output enable 1 = Test clock output enable 0 = Disable the test clock Note: The test clock is output to the SEN_CLK pin.
[6]	SW_CLK	Software Generated Clock This bit is used to generate clock by writing this bit high then writing this bit low repeatedly. This bit is for test only and the generated clock will be output to test clock while TEST_CKE is enabled and TEST_SEL is set to 0x35.
[5:0]	TEST_SEL	00_0000 = 0 00_0001 = APLLFout ÷ 8 00_0010 = UPLLFout ÷ 8 00_0011 = XIN 00_0100 = CPUCLK 00_0101 = HCLK 00_0110 = HCLK1 00_0111 = HCLK2 00_1000 = HCLK3 00_1001 = HCLK4 00_1010 = PCLK 00_1011 = HCLK1EN 00_1100 = HCLK2EN 00_1101 = HCLKcpuEN 00_1110 = PCLKEN 00_1111 = HCLKjpg

		01_0000 = HCLKcap 01_0001 = PCLKkpi 01_0010 = HCLKsic 01_0011 = HCLKnand 01_0100 = HCLKusbd 01_0101 = HCLKi2s 01_0110 = HCLKspu 01_0111 = HCLKvpost 01_1000 = HCLKtic 01_1001 = HCLKblt 01_1010 = Reserved 01_1011 = HCLKedma0 01_1100 = HCLKusbh 01_1101 = HCLKsram 01_1110 = ECLKi2s 01_1111 = ECLKspui2s
		10_0000 = PCLKspims 10_0001 = PCLKspim 10_0010 = PCLKadc 10_0011 = PCLKuart0 10_0100 = PCLKpwm 10_0101 = PCLKrtc 10_0110 = PCLKwdg 10_0111 = PCLKtm0 10_1000 = ECLKcap 10_1001 = ECLKwdg 10_1010 = ECLKtm0 10_1011 = ECLKtm1 10_1100 = ECLKuart0 10_1101 = ECLKuart1 10_1110 = ECLKvpost 10_1111 = ECLKvpost ÷ 2

		11_0000 = DDR_CLK 11_0001 = DRAM_CLK 11_0010 = ADC_CLK 11_0011 = ADO_CLK 11_0100 = USB_CLK 11_0101 = USB20_CLK 11_0110 = SD_CLK 11_0111 = ECLKgpio 11_1000 = PCLKuart1 11_1001 = PCLKi2c 11_1010 = PCLKtm1 11_1011 = HCLKedma1 11_1100 = HCLKedma2 11_1101 = HCLKedma3 11_1110 = HCLKedma4 11_1111 = SW_CLK

5.4 SDRAM Controller (SDRAM)

5.4.1 Overview

The SDRAM Controller support SDR, DDR, Low-Power DDR and DDR2 type SDRAM. The memory device size type can be from 16M bit and up to 1G bits. Only 16-bit data bus width is supported. The total system memory size can be from 2M bytes and up to 256M bytes for different SDRAM configuration.

The SDRAM controller interface to three isolated AHB. All these AHB masters can access the memory independent. Except the memory access, the masters of AHB also could access the SDRAM control registers.

For performance and function issue, the SDRAM controller also supports the proprietary Enhanced-AHB. The EAHB add the down-count address mode, byte-enable signal and explicit burst access number. The explicit access number function is reached by modify the HBURST signal to EHBURST and it represent the access number. The maximum EAHB access number is 16. The SDRAM controller also builds a BIST module to test the external memory device.

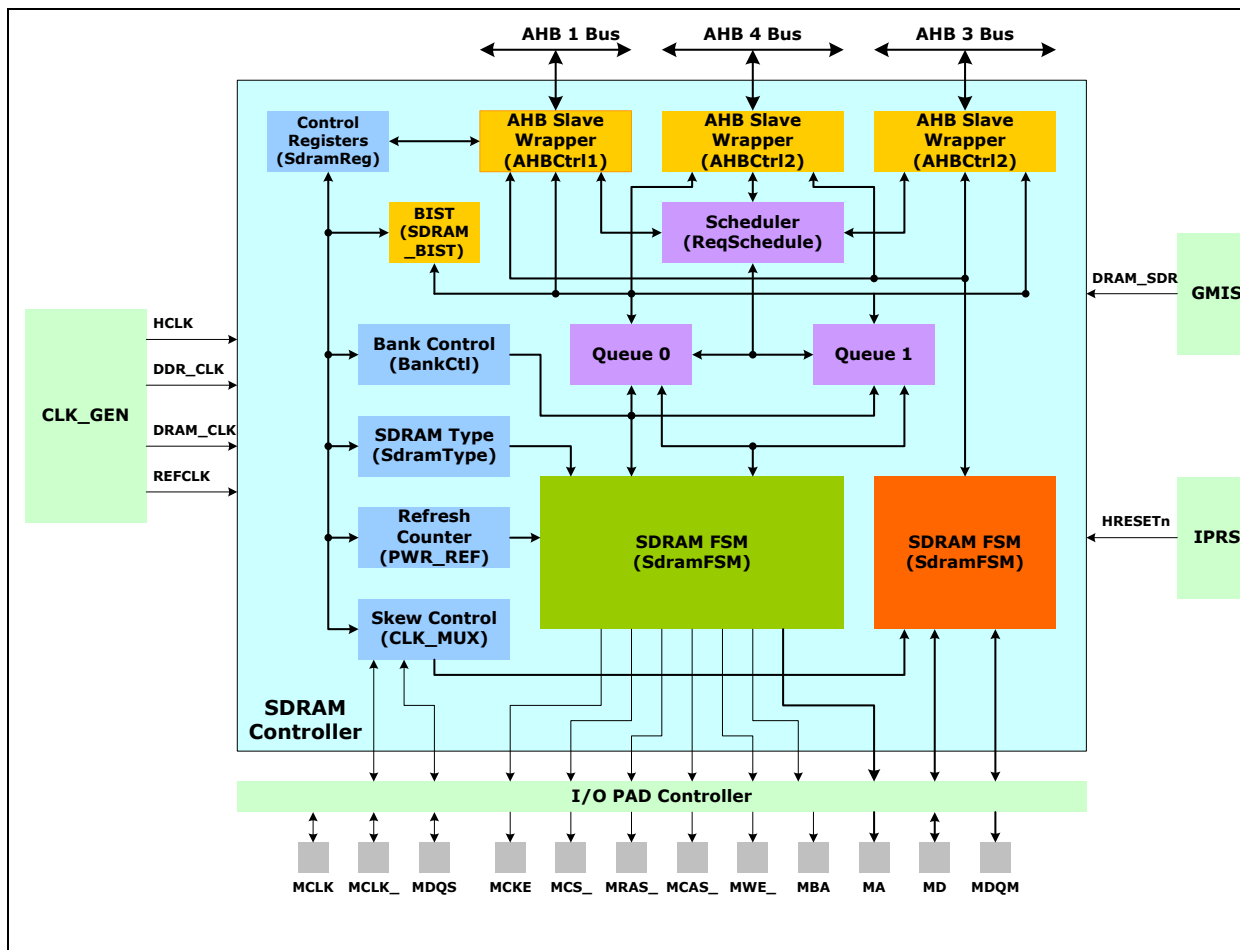
An internal arbiter is used to schedule the access from the masters and the BIST request, the BIST request with the highest priority and the then the AHB3 master, AHB2 master and AHB1 master.

The SDRAM controller uses 3 pipe queues to improve the SDRAM command and data bus efficiency. The request in queue0 is the SDRAM active data access request. Simultaneous, the requests in queue1 can request the controller to issue the ACTIVE or PRECHARGE command to reduce the access latency for the later command. The queue1 also can issue the READ or WRITE command to close the SDRAM command when advance pipe queue

The SDRAM refresh rate is programmable. The Refresh and Power-on control module generate the refresh request signal and SDRAM power on sequence. The SDRAM controller also supports software reset, SDRAM self refresh and auto power down function.

Note. By MCP limitation, N9H20 family just supported three kinds of SDRAM size 2MB, 8MB and 32MB solution, about the detail please refer the ordering information of datasheet.

5.4.2 Block Diagram



5.4.3 SDRAM Control Timing

The SDRAM Controller supports programmable CAS Latency and Refresh Rate control. The SDR SDRAM initial sequence is also automatic trigger when the system power up. It also can control the SDRAM to enter self-refresh mode to reduce the power consumption in power-down mode.

The SDRAM controller provides the fixed sequential burst type and some other programmable controls for the SDRAM operations include:

- SDRAM Type: SDR, DDR, Low-Power DDR and DDR2 type SDRAM.
- SDRAM Size: 16Mbits, 64Mbits, 128Mbits, 256Mbits, 512Mbits and 1Gbits with 16 bits configurations.
- SDRAM Timing: adjustable tWR, tRP, tRCD, tRAS, tRFC, tXSR, tRC, tRRD and tWTR timings.
- SDRAM Read Latency: 2 ~ 4 clocks and don't support DDR 2.5 latency.
- SDRAM Burst Length: support Burst length 4 only.
- SDRAM Refresh: Normally Auto Refresh or power save mode Self Refresh.

5.4.4 SDRAM Power-Up Sequence

Before the SDRAM can be accessed for read or writing after power on, or when exiting deep power-down mode, an SDRAM device must be initialized by software to progress an initialization sequence.

Because the SDR SDRAM, DDR, Low-Power DDR and DDR2 SDRAM require different initialization sequences and different parameters, the sequence is driven by software manually by using the registers SDCMD, SDMR, SDEMR, SDEMR2 and SDEMR3.

For SDR SDRAM device, the initialization procedure is:

1. Wait for 200us after power up.
2. Set the SDRAM controller in initialization state. This is accomplished by writing 1 to InitState of register SDCMD.
3. Set the CKE_H of register SDCMD to be 1 to force the CKE at high state.
4. Apply a PRECHARGE ALL command. This is accomplished by writing 1 to PALL_CMD of register SDCMD. The PALL_CMD bit will auto clear after the PRECHARGE command completed.
5. Apply two or more AUTOREFRESH commands. This is accomplished by writing 1 to REF_CMD of register SDCMD twice or more. The REF_CMD is auto cleared after SDRAM controller completes each CAS-BEFORE-RAS refresh command.
6. Apply an MRS (Mode Register Set) command to MR (Mode Register). This is accomplished by writing appropriate value to the register SDMR (SDRAM MODE Register).
7. SDRAM initialization sequence completed and it's necessary to make SDRAM controller to exit initialization state and into normal operating mode. This is accomplished by writing 0 to both InitState and CKE_H of register SDCMD.

The sequence listed above is also suitable for the low power (mobile) SDRAM device and the DDR SDRAM device does not include delay-locked loop technology (DLL).

For Standard DDR, the initialization procedure is:

1. Wait for 200us after power up.
2. Set the SDRAM type is DDR. This is accomplished by setting 2'b10 to SD_TYPE of register SDOPM.
3. Set the SDRAM controller in initialization state. This is accomplished by writing 1 to InitState of register SDCMD.
4. Set the CKE_H of register SDCMD to be 1 to force the CKE at high state.
5. Apply a PRECHARGE ALL command. This is accomplished by writing 1 to PALL_CMD of register SDCMD. The PALL_CMD bit will auto clear after the PRECHARGE command completed.
6. Apply a MRS (Mode Register Set) command to EMR (Extended Mode Register) to enable DLL. This is accomplished by writing appropriate value to the register SDEMR (SDRAM EXTEND MODE Register).
7. Apply a MRS (Mode Register Set) command to MR (Mode Register) with A8 high to set DDR SDRAM in normal operation with resetting the DLL. This is accomplished by writing appropriate value with bit [8] high to the register SDMR (SDRAM MODE Register).
8. Apply a PRECHARGE ALL command. This is accomplished by writing 1 to PALL_CMD of register SDCMD. The PALL_CMD bit will auto clear after the PRECHARGE command completed.
9. Apply two or more AUTOREFRESH commands. This is accomplished by writing 1 to REF_CMD of register SDCMD twice or more. The REF_CMD is auto cleared after SDRAM

controller completes each CAS-BEFORE-RAS refresh command.

10. Apply a MRS (Mode Register Set) command to MR (Mode Register) with A8 low to set DDR SDRAM in normal operation without resetting the DLL. This is accomplished by writing appropriate value with bit [8] low to the register SDMR (SDRAM MODE Register).
11. Apply 200 dummy clocks to meet minimum latency delay between MRS and normal operation command (ACTIVE, READ, WRITE ...). This is accomplished by inserting a period of delay.
12. SDRAM initialization sequence completed and it's necessary to make SDRAM controller to exit initialization state and into normal operating mode. This is accomplished by writing 0 to both InitState and CKE_H of register SDCMD.
13. Doing system memory remap to map SDRAM to address 0x0000_0000 of system memory. This is accomplished by writing 0x60000 to register SYSCFG.

For Standard DDR2, the initialization procedure is:

1. Wait for 200us after power up.
2. Apply NOP or DESELECT commands for a minimum 400 ns.
3. Set the SDRAM type is DDR. This is accomplished by setting 2'b11 to SD_TYPE of register SDOPM.
4. Set the SDRAM controller in initialization state. This is accomplished by writing 1 to InitState of register SDCMD.
5. Set the CKE_H of register SDCMD to be 1 to force the CKE at high state.
6. Apply a PRECHARGE ALL command. This is accomplished by writing 1 to PALL_CMD of register SDCMD. The PALL_CMD bit will auto clear after the PRECHARGE command completed.
7. Apply a MRS (Mode Register Set) command to EMR2 (Extended Mode Register 2). This is accomplished by writing appropriate value to the register SDEMR2 (SDRAM EXTEND MODE Register 2).
8. Apply a MRS (Mode Register Set) command to EMR3 (Extended Mode Register 3). This is accomplished by writing appropriate value to the register SDEMR3 (SDRAM EXTEND MODE Register 3).
9. Apply a MRS (Mode Register Set) command to EMR (Extended Mode Register) to enable DLL. This is accomplished by writing appropriate value to the register SDEMR (SDRAM EXTEND MODE Register). When writing SDEMR register, writing 3'b000 to bit [9:7] is recommended.
10. Apply a MRS (Mode Register Set) command to MR (Mode Register) with A8 high to set DDR SDRAM in normal operation with resetting the DLL. This is accomplished by writing appropriate value with bit [8] high to the register SDMR (SDRAM MODE Register).
11. Apply a PRECHARGE ALL command. This is accomplished by writing 1 to PALL_CMD of register SDCMD. The PALL_CMD bit will auto clear after the PRECHARGE command completed.
12. Apply two or more AUTOREFRESH commands. This is accomplished by writing 1 to REF_CMD of register SDCMD twice or more. The REF_CMD is auto cleared after SDRAM controller completes each CAS-BEFORE-RAS refresh command.
13. Apply a MRS (Mode Register Set) command to MR (Mode Register) with A8 low to set DDR SDRAM in normal operation without resetting the DLL. This is accomplished by writing appropriate value with bit [8] low to the register SDMR (SDRAM MODE Register).

14. Apply a MRS (Mode Register Set) command to EMR (Extended Mode Register) to enable OCD default state. This is accomplished by writing appropriate value with 3'b111 in bit [9:7] to the register SDEMR (SDRAM EXTEND MODE Register).
15. Apply a MRS (Mode Register Set) command to EMR (Extended Mode Register) to enable OCD exit state. This is accomplished by writing appropriate value with 3'b000 in bit [9:7] to the register SDEMR (SDRAM EXTEND MODE Register).
16. Apply 200 dummy clocks to meet minimum latency delay between MRS and normal operation command (ACTIVE, READ, WRITE ...). This is accomplished by inserting a period of delay.
17. SDRAM initialization sequence completed and it's necessary to make SDRAM controller to exit initialization state and into normal operating mode. This is accomplished by writing 0 to both InitState and CKE_H of register SDCMD.
18. Doing system memory remap to map SDRM to address 0x0000_0000 of system memory. This is accomplished by writing 0x60000 to register SYSCFG.

5.4.5 SDRAM Interface Signals

Pin Name	DDR2 (16-Bit)	DDR (16-Bit)	SDR (16-Bit)
MD[15:0]	MD[15:0]	MD[15:0]	MD[15:0]
MA[12:0]	MA[12:0]	MA[12:0]	MA[12:0]
BA0	BA0	BA0	BA0
BA1	BA1	BA1	BA1
BA2	BA2	MA[13]	
CS#	CS#	CS#	CS#
DQM0	DQML	DQML	DQML
DQM1	DQMH	DQMH	DQMH
CKE	CKE	CKE	CKE
WE#	WE#	WE#	WE#
RAS#	RAS#	RAS#	RAS#
CAS#	CAS#	CAS#	CAS#
CK	CK	CK	CLK
CK#	CK#	CK#	
LDQS	LDQS	LDQS	
UDQS	UDQS	UDQS	

5.4.6 SDRAM Components Supported

- 16M-bit SDRAM Devices
 - 1Mx16 with 2 banks: RA0 ~ RA10, CA0 ~ CA7
- 64M-bit SDRAM Devices
 - 4Mx16 with 4 banks: RA0 ~ RA11, CA0 ~ CA7
- 128M-bit SDRAM

- 8Mx16 with 4 banks: RA0 ~ RA11, CA0 ~ CA8
- 256M-bit SDRAM
 - 16Mx16 with 4 banks: RA0 ~ RA12, CA0 ~ CA8
- 512M-bit SDRAM
 - 32Mx16 with 4 banks: RA0 ~ RA12, CA0 ~ CA9
- 1G-bit SDRAM
 - 64Mx16 with 4 banks: RA0 ~ RA13, CA0 ~ CA9
 - 64Mx16 with 8 banks: RA0 ~ RA12, CA0 ~ CA9

5.4.7 AHB Bus Address Mapping to SDRAM Bus

The following table indicates how the 32-bit AHB bus address be mapped to SDRAM address. All the SDRAM devices listed below are 16-bit data bus width.

5.4.7.1 For SDR SDRAM

Type	R X C	R/C	BA1	BA0	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
16M 1Mx16	11x8	R		9			12	13	11	10	20	19	18	17	16	15	14
		C					AP			8	7	6	5	4	3	2	1
64M 4Mx16	12x8	R	10	9		11	12	13	22	21	20	19	18	17	16	15	14
		C					AP			8	7	6	5	4	3	2	1
128M 8Mx16	12x9	R	11	10		23	12	13	22	21	20	19	18	17	16	15	14
		C					AP		9	8	7	6	5	4	3	2	1
256M 16Mx16	13x9	R	11	10	24	23	12	13	22	21	20	19	18	17	16	15	14
		C					AP		9	8	7	6	5	4	3	2	1
512M 32Mx16	13x10	R	12	11	25	23	24	13	22	21	20	19	18	17	16	15	14
		C					AP	10	9	8	7	6	5	4	3	2	1

5.4.7.2 For DDR SDRAM

Type	R X C	R/C	BA1	BA0	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
128M 8Mx16	12x9	R	11	10			23	12	13	22	21	20	19	18	17	16	15	14
		C					AP			9	8	7	6	5	4	3	2	1
256M 16Mx16	13x9	R	11	10		24	23	12	13	22	21	20	19	18	17	16	15	14
		C					AP			9	8	7	6	5	4	3	2	1

512M 32Mx16	13x10	R	12	11		25	23	24	13	22	21	20	19	18	17	16	15	14
		C						AP	10	9	8	7	6	5	4	3	2	1
1G 64Mx16	14x10	R	12	11	26	25	23	24	13	22	21	20	19	18	17	16	15	14
		C						AP	10	9	8	7	6	5	4	3	2	1

5.4.7.3 For DDR2 SDRAM

Type	R X C	R/C	BA2	BA1	BA0	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
128M 8Mx16	12x9	R		11	10		23	12	13	22	21	20	19	18	17	16	15	14
		C						AP		9	8	7	6	5	4	3	2	1
256M 16Mx16	13x9	R		11	10	24	23	12	13	22	21	20	19	18	17	16	15	14
		C						AP		9	8	7	6	5	4	3	2	1
512M 32Mx16	13x10	R		12	11	25	23	24	13	22	21	20	19	18	17	16	15	14
		C						AP	10	9	8	7	6	5	4	3	2	1
1G 64Mx16	13x10	R	13	12	11	26	23	25	24	22	21	20	19	18	17	16	15	14
		C						AP	10	9	8	7	6	5	4	3	2	1

Note: The AHB bus address HADDR prefixes have been omitted on the following tables.

A13 ~ A00 are the Address pins of the SDRAM interface.

BA2, BA1 and BA0 are the Bank Selected Signal of SDRAM.

5.4.8 SDRAM Control Registers Map

Register	Address	R/W	Description	Reset Value
SDIC_BA = 0xB000_3000				
SDOPM	SDRAM_BA + 00	R/W	SDRAM Controller Operation Mode Control Register	0x0003_04x6
SDCMD	SDRAM_BA + 04	R/W	SDRAM Command Register	0x0000_0021
SDREF	SDRAM_BA + 08	R/W	SDRAM Controller Refresh Control Register	0x0000_80FF
	SDRAM_BA + 0C	R/W	Reserved	0xFFFF_FFFF
SDSIZE0	SDRAM_BA + 10	R/W	SDRAM 0 Size Register	0x0000_000X
SDSIZE1	SDRAM_BA + 14	R/W	SDRAM 1 Size Register	0x1000_0000
SDMR	SDRAM_BA + 18	R/W	SDRAM Mode Register	0x0000_0032
SDEMR	SDRAM_BA + 1C	R/W	SDRAM Extended Mode Register	0x0000_4000
SDEMR2	SDRAM_BA + 20	R/W	SDRAM Extended Mode Register 2	0x0000_8000
SDEMR3	SDRAM_BA + 24	R/W	SDRAM Extended Mode Register 3	0x0000_C000
SDTIME	SDRAM_BA + 28	R/W	SDRAM Timing Control Register	0x2BDE_9649
	SDRAM_BA + 2C	R/W	Reserved	0xFFFF_FFFF
DQSODS	SDRAM_BA + 30	R/W	DQS Output Delay Selection Register	0x0000_1010
CKDQSDS	SDRAM_BA + 34	R/W	Clock and DQS Delay Selection Register	0x0044_4400
	SDRAM_BA + 38	R/W	Reserved	0xFFFF_FFFF
	SDRAM_BA + 3C	R/W	Reserved	0xFFFF_FFFF

5.4.9 Register Details

SDRAM Controller Operation Mode Control Register (SDOPM)

Register	Address	R/W	Description	Reset Value
SDOPM	SDRAM_BA + 00	R/W	SDRAM Controller Operation Mode Control Register	0x0003_04x6

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			RD2WR_CTL	OEDelay	LowFreq	PreActBnk	AutoPDn
15	14	13	12	11	10	9	8
Reserved					RDBUFTH		
7	6	5	4	3	2	1	0
Reserved	SD_TYPE		PchMode	OPMode	MCLKMode	DRAM_EN	Reserved

Bits	Descriptions	
[31:19]	Reserved	Reserved
[20]	RD2WR_CTL	Read-To-Write Turn Around Control This bit is to insert one more turn around cycle between memory read and memory write access to SDRA'M device. 0 = Default turn-around cycle is used. 1 = One more turn-around cycle is inserted between memory read and write access.
[19]	OEDelay	Output Enable Delay Half MCLK This bit control the data output enable signal. If set high, the data output enable will be turned off half MCLK earlier. 0 = Default data output enable timing. 1 = Turn off data output enable half MCLK earlier.
[18]	LowFreq	Low Frequency Mode For low power DDR (LPDDR) SDRAM, the valid read data outputted by LPDDR SDRAM is not ready at clock edge. If this bit is enabled, the SDRAM controller will sampled read data based on the following timing: ● If CL is 2, the read data output latency will be $2 \cdot t_{CK} + t_{AC}$. ● If CL is 3, the read data output latency will be $t_{CK} + t_{AC}$. CL: CAS Latency. t_{CK}: Clock cycle time for LPDDR SDRAM. t_{AC}: Data output latency from clock for LPDDR SDRAM. This bit only takes effect when the SD_TYPE is selected in DDR or DDR2 SDRAM. 0 = SDRAM controller sampled read data based on the DDR/DDR2 standard. (Default) 1 = SDRAM controller sampled read data based on the LPDDR standard.

[17]	PreActBnk	Pre-Active Bank If this bit is enabled, the SDRAM controller will open request bank early to get better performance. It means maybe more than one bank active and consumes more power. There are several bus requests in N9H20 and the SDRAM controller checks all these requests simultaneous. If request in queue access bank is different with current bank, the SDRAM controller will open the new bank early to reduce the access latency to get better performance. The mode takes effect for Close-Page mode (OPMode is 0) only. In Open-Page mode, SDRAM controller always opens bank early. 0 = Disable Pre-Active-Bank mode. 1 = Enable Pre-Active-Bank mode. (Default)										
[16]	AutoPDn	Auto Power Down Mode If this bit is enabled, the SDRAM controller will make SDRAM to enter power down mode (CKE low) automatically while the memory request is stop. Otherwise, the SDRAM is in IDLE state (CKE = high). 0 = Disable auto power down mode. 1 = Enable auto power down mode. (Default)										
[15:11]	Reserved	Reserved										
[10:8]	RDBUFTH	The AHB read SDRAM read buffer threshold control Due to the SDRAM working clock may be higher than the AHB BUS clock, the SDRAM controller contains a read buffer for each AHB interface and they are used for data pre-read. The controller read the data to buffer full or till AHB read request end. When the data buffer full, the controller stop the read request and to service another AHB request. The RDBUFTH is used to control the buffer threshold level for the SDRAM re-start the memory request. This function can minimize the redundant memory read. <table><tr><td>[10:8]</td><td>RDBUFTH</td></tr><tr><td>0,0,0</td><td>Reserved</td></tr><tr><td>others</td><td>Re-start memory read when the data buffer remain data level is equal to RDBUFTH</td></tr></table>	[10:8]	RDBUFTH	0,0,0	Reserved	others	Re-start memory read when the data buffer remain data level is equal to RDBUFTH				
[10:8]	RDBUFTH											
0,0,0	Reserved											
others	Re-start memory read when the data buffer remain data level is equal to RDBUFTH											
[7]	Reserved	Reserved										
[6:5]	SD_TYPE	SDRAM Type This file indicates which type of SDRAM is used. The reset value is decided by chip's system power-on setting. <table><tr><td>SD_TYPE[1:0]</td><td>SDRAM TYPE</td></tr><tr><td>00</td><td>SDR SDRAM Type. (Single Data Rate SDRAM)</td></tr><tr><td>01</td><td>LPDDR SDRAM Type.</td></tr><tr><td>10</td><td>DDR SDRAM Type. (Double-Data-Rate SDRAM)</td></tr><tr><td>11</td><td>DDR2 SDRAM Type.</td></tr></table>	SD_TYPE[1:0]	SDRAM TYPE	00	SDR SDRAM Type. (Single Data Rate SDRAM)	01	LPDDR SDRAM Type.	10	DDR SDRAM Type. (Double-Data-Rate SDRAM)	11	DDR2 SDRAM Type.
SD_TYPE[1:0]	SDRAM TYPE											
00	SDR SDRAM Type. (Single Data Rate SDRAM)											
01	LPDDR SDRAM Type.											
10	DDR SDRAM Type. (Double-Data-Rate SDRAM)											
11	DDR2 SDRAM Type.											

[4]	PchMode	Auto Pre-Charge Mode This bit controls if SDRAM controller will pre-charge all active banks while there is no new memory request. The SDRAM power consumption increases with the active bank number. If no new memory access request, the active bank can be pre-charge to save power, but the SDRAM controller may lose some performance. 0 = The SDRAM controller keeps bank active. 1 = Pre-charge all bank if there is no new memory request. (Default) Note: This bit only take effect while OPMODE is high.
[3]	OPMODE	Open Page Mode This bit controls if the SDRAM controller will send pre-charge command to close the active bank page after SDRAM access. If this bit doesn't be enabled, the SDRAM controller will pre-charge bank after each burst read or write cycle. This could make the SDRAM consume less power. If set this bit high, the state machine will keep on the bank-active state until a page missed read/write request or at a period refresh request. This makes SDRAM controller to get better performance, but SDRAM will consume more power. 0 = Pre-charge after each read/write request. (Default) 1 = No auto pre-charge. The bank page keeps in active state after read/write.
[2]	MCLKMODE	MCLK Mode This bit controls the SDRAM clock (MCLK) is always enabled, or is enabled and disabled by SDRAM controller automatically. 0 = The MCLK is enabled and disabled by SDRAM controller automatically. The MCLK will keep low when the SDRAM is in the power down state. 1 = MCLK is always enabled. (Default)
[1]	SDRAM_EN	SDRAM Controller Enable Set this bit 0 will disable the SDRAM controller function. 0 = Disable the SDRAM controller. 1 = Enable the SDRAM controller. (Default)
[0]	Reserved	Reserved

SDRAM Command Register (SDCMD)

Register	Address	R/W	Description	Reset Value
SDCMD	SDRAM_BA + 04	R/W	SDRAM Command Register	0x0000_0021

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		AutoExSelfRef	SELF_REF	REF_CMD	PALL_CMD	CKE_H	InitState

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5]	AutoExSelfRef	Auto Exit Self-Refresh This controls if the SDRAM will exit self refresh mode automatically while the system interrupt occurred. 0 = Disable auto exit self-refresh function. The SDRAM keep in self-refresh state when the interrupt occur. 1 = Enable auto exit self-refresh function. The SDRAM will exit self-refresh state when the interrupt occur. (Default)
[4]	SELF_REF	Self-Refresh Command Set this bit high, the SDRAM controller will make SDRAM to enter self-refresh mode. SDRAM controller will not have response to any read, write or refresh request until this bit is cleared. If the bit 5 (AutoExSelfRef) is set high, this bit will be cleared automatically when the system interrupt occurred. 0 = SDRAM in normal operation mode. (Default) 1 = Set the SDRAM enter the Self Refresh power saving state
[3]	REF_CMD	Auto Refresh Command Set this bit high the SDRAM controller will issue an auto refresh command to SDRAM. This bit will be cleared by SDRAM controller automatically after the auto refresh command is end. 0 = No operation. (Default) 1 = Issue an auto refresh command to the SDRAM.
[2]	PALL_CMD	Pre-Charge All Bank Command Set this bit high, the SDRAM controller will issue a pre-charge all bank command to the SDRAM. This bit will be cleared by SDRAM controller automatically after the pre-chare all bank command is end 0 = No operation. (Default) 1 = Issue a pre-charge all bank command to the SDRAM.

[1]	CKE_H	CKE High This bit indicates the CKE is controlled by SDRAM controller state machine or always keeps high. 0 = Set the CKE signal in normal state and controlled by the SDRAM controller state machine. (Default) 1 = Set the CKE signal keep in "high" state.
[0]	InitState	Initial State This bit indicates if the SDRAM is in the Initialize State. When the SDRAM is in the initialize state, SDRAM controller will not accept any SDRAM read or write request. The logical state of the internal circuit of the SDRAM is undefined after power on. The SDRAM must be initialized to set the SDRAM into the right operation. The SDR SDRAM, LP SDRAM, DDR SDRAM and DDR2 SDRAM have different initialization sequence, and the users must set the right sequence to initialize the SDRAM. This bit is default high and means the SDRAM is not initialized yet. After the initialization, user must set this bit low to set the SDRAM controller in correct mode. 0 = The SDRAM is in normal state 1 = The SDRAM is in initialization state, the SDRAM initialization doesn't complete yet. (Default)

SDRAM Controller Refresh Control Register (SDREF)

Register	Address	R/W	Description	Reset Value
SDREF	SDRAM_BA + 08	R/W	SDRAM Controller Refresh Control Register	0x0000_80FF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
REF_EN	REFRAT						
7	6	5	4	3	2	1	0
REFRAT							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[15]	REF_EN	Refresh Period Counter Enable This bit controls if the refresh period counter is enabled. If refresh period counter is disabled, the SDRAM controller would never issue auto-refresh command to SDRAM automatically. However, if refresh period counter is enabled, the SDRAM controller will issue auto-refresh command to SDRAM automatically once the refresh period counter is equal to REFRATE. 0 = Refresh period counter is disabled. 1 = Refresh period counter is enabled to trigger SDRAM controller to issue auto-refresh command to SDRAM periodically. (Default)
[14:0]	REFRATE	Refresh Count Value This field defines the period for SDRAM controller to generate the auto-refresh command to SDRAM. The SDRAM controller will issue an auto-refresh cycle to SDRAM automatically for every period programmed in the REFRAT field when the REF_EN bit of is set. The refresh period is calculated as $Period = REFRAT / f_{SCLK}$. The f_{SCLK} is the frequency of external crystal for chip.

SDRAM Size Register 0/1 (SDSIZE0/1)

Register	Address	R/W	Description	Reset Value
SDSIZE0	SDRAM_BA + 10	R/W	SDRAM 0 Size Register	0x0000_000X
SDSIZE1	SDRAM_BA + 14	R/W	SDRAM 1 Size Register	0x1000_0000

31	30	29	28	27	26	25	24
Reserved			BASADDR				
23	22	21	20	19	18	17	16
BASADDR			Reserved				
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				BUSWD	DRAMSIZE		

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28:21]	BASADDR	Base Address This field defines the memory space where the SDRAM is mapped. In N9H20, the SDRAM could be mapped to address 0x0000_0000 ~ 0x1fff_ffff of system memory, and shadow address on 0x8000_0000 ~ 0x9fff_ffff of system memory. The minimum supported SDRAM size is 2M bytes. Based on the above criteria, the bit [28:21] is used to define the base address. For example, if [28:21] is set as 0x01, the address 0 of SDRAM memory will be mapped to 0x00200000 of system memory.
[20:4]	Reserved	Reserved
[3]	BUSWD	SDRAM Data Bus width This bit defines if the data bus width of SDRAM is 16 bit or 32 bit. The DDR and DDR2 type SDRAM only support 16 bit data bus width and this bit will be 1'b0. In N9H20, SDRAM controller only support 16 bit SDRAM. So, this bit will be fixed at 1'b0. Write 1 to this bit doesn't take any effect. 0 = 16bits SDRAM data BUS width (Default) 1 = Reserved Note: In register SDSIZE1, this field is reserved.

[2:0]	DRAMSIZE	Size of SDRAM Device			
		This field indicates the size of SDRAM device.			
		The default memory size is 2MB or 16MB depend on power on setting value. If the power on setting value indicates the SDRAM type is DDR/DDR2, the default size is 16MB (8Mx16). Otherwise, the default size is 2MB (1Mx16).			
		Note: In register SDSIZE1, this field is reserved.			
[2:0]			SIZE of SDRAM (Byte)		
0	0	0	SDRAM disable		
0	0	1	2M		
0	1	0	4M		
0	1	1	8M		
1	0	0	16M		
1	0	1	32M		
1	1	0	64M		
1	1	1	128M		

SDRAM TYPE (Byte) table

SDRAM SIZE	SDR SDRAM	DDR SDRAM	DDR2 SDRAM
	16 bits	16 bits	16 bits
2MB	1Mx16 (16Mbits)	Reserved	Reserved
4MB	Reserved	Reserved	Reserved
8MB	4Mx16 (64Mbits)	Reserved	Reserved
16MB	8Mx16 (128Mbits)	8Mx16 (128Mbits)	8Mx16 (128Mbits)
32MB	16Mx16 (256Mbits)	16Mx16 (256Mbits)	16Mx16 (256Mbits)
64MB	32Mx16 (512Mbits)	32Mx16 (512Mbits)	32Mx16 (512Mbits)
128MB	Reserved	64Mx16 (1Gbits)	64Mx16 (1Gbits)

SDRAM Mode Register (SDMR)

The SDRAM mode registers is used to configure the Mode Register of SDRAM device. This Mode Register value will be applied to both SDRAM 0 and SDRAM 1 devices.

Write this register, the SDRAM controller will generate a Load Mode Register (LMR) command to the SDRAM device.

Register	Address	R/W	Description	Reset Value
SDMR	SDRAM_BA + 18	R/W	SDRAM Mode Register	0x0000_0032

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		Configure					
7	6	5	4	3	2	1	0
Configure	LATENCY			BrstType	BrstLength		

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13:7]	Configure	SDRAM Dependent Configuration The value of this field is SDRAM type dependent. The definition of bits in this field is different between SDR SDRAM, DDR SDRAM and DDR2 SDRAM. Please refer the SDRAM initial sequence and related SDRAM specification to know what value should be configured in this field.

[6:4]	LATENCY	CAS Latency This field defines the CAS latency parameter of external SDRAM device. In N9H20, SDRAM controller doesn't support the mode CAS latency is 2.5. Setting CAS latency to be 2.5 is inhibited. For DDR2 SDRAM, SDRAM controller only support CAS latency is 3 or 4. Setting CAS latency to be 5 or 6 is inhibited.																																																																					
		<table><tr><th colspan="3">LATENCY</th><th>SDR</th><th>DDR</th><th>DDR400</th><th>DDR2</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr><tr><td>0</td><td>0</td><td>1</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr><tr><td>0</td><td>1</td><td>0</td><td>2</td><td>2</td><td>2</td><td>Reserved</td></tr><tr><td>0</td><td>1</td><td>1</td><td>3</td><td>3</td><td>Reserved</td><td>3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>4</td><td>Reserved</td><td>Reserved</td><td>4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>5 (Inhibit)</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Reserved</td><td>2.5 (Inhibit)</td><td>2.5 (Inhibit)</td><td>6 (Inhibit)</td></tr><tr><td>1</td><td>1</td><td>1</td><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr></table>							LATENCY			SDR	DDR	DDR400	DDR2	0	0	0	Reserved	Reserved	Reserved	Reserved	0	0	1	Reserved	Reserved	Reserved	Reserved	0	1	0	2	2	2	Reserved	0	1	1	3	3	Reserved	3	1	0	0	4	Reserved	Reserved	4	1	0	1	Reserved	Reserved	Reserved	5 (Inhibit)	1	1	0	Reserved	2.5 (Inhibit)	2.5 (Inhibit)	6 (Inhibit)	1	1	1	Reserved	Reserved	Reserved	Reserved
		LATENCY			SDR	DDR	DDR400	DDR2																																																															
		0	0	0	Reserved	Reserved	Reserved	Reserved																																																															
		0	0	1	Reserved	Reserved	Reserved	Reserved																																																															
		0	1	0	2	2	2	Reserved																																																															
		0	1	1	3	3	Reserved	3																																																															
		1	0	0	4	Reserved	Reserved	4																																																															
		1	0	1	Reserved	Reserved	Reserved	5 (Inhibit)																																																															
		1	1	0	Reserved	2.5 (Inhibit)	2.5 (Inhibit)	6 (Inhibit)																																																															
1	1	1	Reserved	Reserved	Reserved	Reserved																																																																	
The CAS latency setting listed above is for reference. Before configuring SDRAM CAS latency, it's necessary to confirm the CAS latency value supported by SDRAM device.																																																																							
[3]	BrstType	Burst Type This bit indicates the burst type of SDRAM device is sequential or interleaved. In N9H20, the SDRAM controller only support sequential burst type and this bit will be fixed at 1'b0. 0 = Sequential burst type. (Default) 1 = Reserved.																																																																					

[2:0]

BrstLength

Burst Length

This field defines the burst length of external SDRAM device.

SDRAM controller only supports the burst length 4. Setting burst length to be other value is inhibited.

Burst Length			SDR	DDR	DDR2
0	0	0	1 (Inhibit)	Reserved	Reserved
0	0	1	2 (Inhibit)	2 (Inhibit)	Reserved
0	1	0	4	4	4
0	1	1	8 (Inhibit)	8 (Inhibit)	8 (Inhibit)
1	0	0	Reserved	Reserved	Reserved
1	0	1	Reserved	Reserved	Reserved
1	1	0	Reserved	Reserved	Reserved
1	1	1	Full Page	Reserved	Reserved

SDRAM Extended Mode Register (SDEMR)

The SDRAM Extended Mode Register is used to configure SDRAM Extend Mode Register. This Extended Mode Register value will be applied to both SDRAM 0 and SDRAM 1 devices.

Write this register, the SDRAM controller will generate a Load Mode Register (LMR) command to the SDRAM. This Extended Mode Register is only used for DDR and DDR2 SDRAM.

Register	Address	R/W	Description	Reset Value
SDEMR	SDRAM_BA + 1C	R/W	SDRAM Extended Mode Register	0x0000_4000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		Configure					
7	6	5	4	3	2	1	0
Configure						DrvStrength	DLLLEN

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13:2]	Configure	SDRAM Dependent Configuration The value of this field is SDRAM type dependent. The definition of bits in this field is different between DDR SDRAM and DDR2 SDRAM. Please refer the SDRAM initial sequence and related SDRAM specification to know what value should be configured in this field.
[1]	DrvStrength	Output Drive Strength This bit sets the SDRAM output drive strength. 0 = Normal drive strength. 1 = Reduced drive strength.
[0]	DLLLEN	DLL Enable This bit is to enable or disable the DLL of SDRAM device. 0 = Enable DLL of SDRAM device. 1 = Disable DLL of SDRAM device.

SDRAM Extended Mode Register 2 (SDEMR2)

The SDRAM Extended Mode Register 2 is used to configure SDRAM Extend Mode Register 2. This Extended Mode Register 2 value will be applied to both SDRAM 0 and SDRAM 1 devices.

Write this register, the SDRAM controller will generate a Load Mode Register (LMR) command to the SDRAM. This Extended Mode Register is only used for DDR2 SDRAM.

Register	Address	R/W	Description	Reset Value
SDEMR2	SDRAM_BA + 20	R/W	SDRAM Extended Mode Register 2	0x0000_8000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		Configure					
7	6	5	4	3	2	1	0
Configure							

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13:0]	Configure	SDRAM Dependent Configuration The value of this field is SDRAM type dependent and only available for DDR2 SDRAM. Please refer the SDRAM initial sequence and related SDRAM specification to know what value should be configured in this field.

SDRAM Extended Mode Register 3 (SDEMR3)

The SDRAM Extended Mode Register 3 is used to configure SDRAM Extend Mode Register 3. This Extended Mode Register 3 value will be applied to both SDRAM 0 and SDRAM 1 devices.

Write this register, the SDRAM controller will generate a Load Mode Register (LMR) command to the SDRAM. This Extended Mode Register is only used for DDR2 SDRAM.

Register	Address	R/W	Description	Reset Value
SDEMR3	SDRAM_BA + 24	R/W	SDRAM Extended Mode Register 3	0x0000_C000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		Configure					
7	6	5	4	3	2	1	0
Configure							

Bits	Descriptions	
[31:18]	Reserved	Reserved
[17:15]	MR_DEF	Mode Register Definition For Extended Mode Register 3, this field is fixed at 3'b011.
[14:0]	Configure	SDRAM Dependent Configuration The value of this field is SDRAM type dependent and only available for DDR2 SDRAM. Please refer the SDRAM initial sequence and related SDRAM specification to know what value should be configured in this field.

SDRAM Timing Control Register (SDTIME)

This timing control register defines some SDRAM timing parameters that should be followed during SDRAM access. These timing parameters are SDRAM dependent. Refer SDRAM device's data sheet to set these timing parameters is recommended.

Register	Address	R/W	Description	Reset Value
SDTIME	SDRAM_BA + 28	R/W	SDRAM Timing Control Register	0x2BDE_9649

31	30	29	28	27	26	25	24
Reserved	tWTR		tRRD		tRC		
23	22	21	20	19	18	17	16
tRC		tXSR					tRFC
15	14	13	12	11	10	9	8
tRFC				tRAS			
7	6	5	4	3	2	1	0
tRCD			tRP			tWR	

Bits	Descriptions	
[31]	Reserved	Reserved
[30:29]	tWTR	Internal Write to Read Command Delay This timing defines the minimum delay latency from last write data to next new valid READ command and only takes effect while SDRAM type is DDR or DDR2. $tWTR = t_{HCLK} * (tWTR + 1)$ HCLK: It's the operating clock of SDRAM controller.
[28:27]	tRRD	Active Bank a to Active Bank b Command Delay This timing defines the minimum delay latency between SDRAM bank a ACTIVE command to SDRAM bank B ACTIVE command. $tRRD = t_{HCLK} * (tRRD + 1)$ HCLK: It's the operating clock of SDRAM controller.
[26:22]	tRC	Active to Active Command Delay This timing defines the minimum delay latency between two ACTIVE commands. $tRC = t_{HCLK} * (tRC + 1)$ HCLK: It's the operating clock of SDRAM controller.

[21:17]	tXSR	Exit SELF REFRESH to ACTIVE Command Delay This timing defines the minimum delay latency from SDRAM exiting self refresh mode to next valid ACTIVE command. $t_{XSR} = t_{HCLK} * (tXSR+1)$ HCLK: It's the operating clock of SDRAM controller.
[16:12]	tRFC	AUTO REFRESH Period This timing defines the minimum delay latency from AUTO-REFRESH command to any other command. $t_{RFC} = t_{HCLK} * (tRFC+1)$ HCLK: It's the operating clock of SDRAM controller.
[11:8]	tRAS	ACTIVE to PRECHARGE Command Delay This timing defines the minimum delay latency from a valid ACTIVE command to PRECHARGE command. $t_{RAS} = t_{HCLK} * (tRAS+1)$ HCLK: It's the operating clock of SDRAM controller.
[7:5]	tRCD	Active to READ or WRITE Delay This timing defines the minimum delay latency from a ACTIVE command to READ or WRITE command. $t_{RCD} = t_{HCLK} * (tRCD+1)$ HCLK: It's the operating clock of SDRAM controller.
[4:2]	tRP	PRECHARGE Command Period This timing defines the minimum delay latency from PRECHARGE command to any other command. $t_{RP} = t_{HCLK} * (tRP+1)$ HCLK: It's the operating clock of SDRAM controller.
[1:0]	tWR	WRITE Recovery Time This timing defines the minimum delay latency from last valid write data to PRECHARGE command. $t_{WR} = t_{HCLK} * (tWR+1)$ HCLK: It's the operating clock of SDRAM controller.

DQS Output Delay Selection Register (DQSODS)

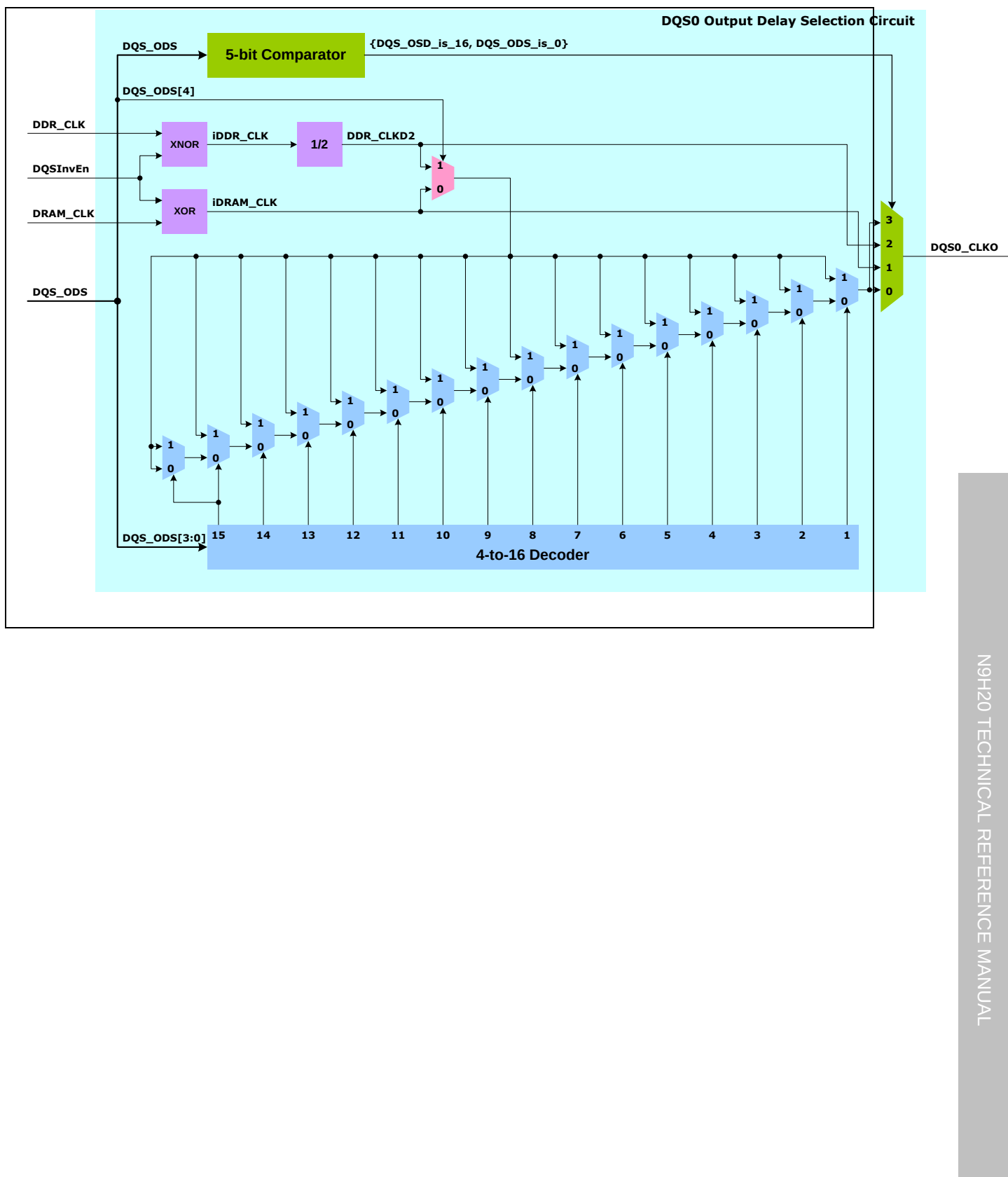
This register controls the DQS output delay and source selection circuit for DQS0 and DQS1 output generation. This control register only takes effect while SDRAM type is DDR or DDR2. The function equivalent circuit for DQS output delay selection is shown below. There are two same circuits in the SDRAM controller. One is for DQS0 generation and the other is for DQS1 generation.

Register	Address	R/W	Description	Reset Value
DQSODS	SDRAM_BA + 30	R/W	DQS Output Delay Selection Register	0x0000_1010

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		DQSInvEn	DQS1_ODS				
7	6	5	4	3	2	1	0
Reserved			DQS0_ODS				

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13]	DQSInvEn	DQS Invert Enable This bit controls if the clock DRAM_CLK or DDR_CLK/2 is inverted for DQS0 output generation while DQS0_ODS [3:0] is 4'b0000. This control bit takes the same effect for DQS1 output generation while DQS1_ODS [11:8] is 4'b0000. 0 = DRAM_CLK and DDR_CLK/2 is not inverted for DQS0/DQS1 output generation. (Default) 1 = DRAM_CLK and DDR_CLK/2 is inverted for DQS0/DQS1 output generation.

[12:8]	DQS1_ODS	DQS1 Output Delay Selection	
		This field controls the DQS1 output delay value and source selection circuit for DQS1 output generation. The brief circuit for this function is listed below.	
		DQS1_ODS	DQS1 Output
		5'b0_0000	The DQS1 is generated from DRAM_CLK.
		5'b0_0001 ~ 5'b0_1111	The DQS1 is generated from DRAM_CLK with a delay value. This delay value is controlled by the following equation: $DQS1\ delay = DQS1\_ODS[11:8] * Delay_{CLKMUX}$ Delay _{CLKMUX} : It's the gate delay of a CLKMUX gate.
		5'b1_0000	The DQS1 is generated from DDR_CLK/2.
		5'b1_0001 ~ 5'b1_1111	The DQS1 is generated from DDR_CLK/2 with a delay value. This delay value is controlled by the following equation: $DQS1\ delay = DQS1\_ODS[11:8] * Delay_{CLKMUX}$ Delay _{CLKMUX} : It's the gate delay of a CLKMUX gate.
[7:5]	Reserved	Reserved	
[4:0]	DQS0_ODS	DQS0 Output Delay Selection	
		This field controls the DQS0 output delay value and source selection circuit for DQS0 output generation. The brief circuit for this function is listed below.	
		DQS0_ODS	DQS0 Output
		5'b0_0000	The DQS0 is generated from DRAM_CLK.
		5'b0_0001 ~ 5'b0_1111	The DQS0 is generated from DRAM_CLK with a delay value. This delay value is controlled by the following equation: $DQS1\ delay = DQS0\_ODS[3:0] * Delay_{CLKMUX}$ Delay _{CLKMUX} : It's the gate delay of a CLKMUX gate.
		5'b1_0000	The DQS0 is generated from DDR_CLK/2.
		5'b1_0001 ~ 5'b1_1111	The DQS1 is generated from DDR_CLK/2 with a delay value. This delay value is controlled by the following equation: $DQS1\ delay = DQS0\_ODS[3:0] * Delay_{CLKMUX}$ Delay _{CLKMUX} : It's the gate delay of a CLKMUX gate.



Clock and DQS Delay Selection Register (CKDQSDS)

Register	Address	R/W	Description	Reset Value
CKDQSDS	SDRAM_BA + 34	R/W	Clock and DQS Delay Selection Register	0x0044_4400

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DQS1_DS1				DQS1_DS0			
15	14	13	12	11	10	9	8
DQS0_DS1				DQS0_DS0			
7	6	5	4	3	2	1	0
DCLK_DS				DCLKSrcSel	MCLK_ODS		

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:20]	DQS1_DS1	DQS1 Input Delay Selection 1 This field controls the DQS1 input delay selection circuit to generate a clock signal DQS11_CLKIn. DQS11_CLKIn is used to sample the data bits [15:12] outputted by SDRAM device. This field only takes effect while the SDRAM type is DDR or DDR2. This delay value is controlled by the following equation: $DQS11\_CLKIn\ delay = DQS1\_DS1 * Delay_{CLKMUX}$ Delay_{CLKMUX}: It's the gate delay of a CLKMUX gate.
[19:16]	DQS1_DS0	DQS1 Input Delay Selection 0 This field controls the DQS1 input delay selection circuit to generate a clock signal DQS10_CLKIn. DQS10_CLKIn is used to sample the data bits [11:8] outputted by SDRAM device. This field only takes effect while the SDRAM type is DDR or DDR2. This delay value is controlled by the following equation: $DQS10\_CLKIn\ delay = DQS1\_DS0 * Delay_{CLKMUX}$ Delay_{CLKMUX}: It's the gate delay of a CLKMUX gate.
[15:12]	DQS0_DS1	DQS0 Input Delay Selection 1 This field controls the DQS0 input delay selection circuit to generate a clock signal DQS01_CLKIn. DQS01_CLKIn is used to sample the data bits [7:4] outputted by SDRAM device. This field only takes effect while the SDRAM type is DDR or DDR2. This delay value is controlled by the following equation: $DQS01\_CLKIn\ delay = DQS0\_DS1 * Delay_{CLKMUX}$ Delay_{CLKMUX}: It's the gate delay of a CLKMUX gate.

[11:8]	DQS0_DS0	DQS0 Input Delay Selection 0 This field controls the DQS0 input delay selection circuit to generate a clock signal DQS00_CLKIn. DQS00_CLKIn is used to sample the data bits [3:0] outputted by SDRAM device. This field only takes effect while the SDRAM type is DDR or DDR2. This delay value is controlled by the following equation: $DQS00\_CLKIn\ delay = DQS0\_DS0 * Delay_{CLKMUX}$ Delay_{CLKMUX}: It's the gate delay of a CLKMUX gate.
[7:4]	DCLK_DS	Data Clock Delay Selection This field controls the delay selection circuit to generate a clock signal DataCLK. If SDRAM type is DDR or DDR2, the DataCLK is used to sample the data registered by {DQS11_CLKIn, DQS10_CLKIn, DQS01_CLKIn, DQS00_CLKIn}. Or, the DataCLK is used to sample the data outputted by SDRAM device. This control field only takes effect while DCLKSrcSel is set low. The delay value is controlled by the following equation: $DataCLK\ delay = DCLK\_DS * Delay_{CLKMUX}$ Delay_{CLKMUX}: It's the gate delay of a CLKMUX gate.
[4]	DCLKSrcSel	Data Clock Source Selection This bit controls if the DataCLK source is from HCLK or MCLK (from SDRAM clock MCLK I/O buffer). 0 = DataCLK is from MCLK. (Default) 1 = DataCLK is from HCLK.
[3:0]	MCLK_ODS	MCLK Output Delay Selection This field controls the delay selection circuit for SDRAM clock MCLK generation. The delay value is controlled by the following equation: $MCLK\ delay = MCLK\_ODS * Delay_{CLKMUX}$ Delay_{CLKMUX}: It's the gate delay of a CLKMUX gate.

5.5 BitBlt Blitting Accelerator(BitBLT)

5.5.1 Overview

The BitBLT accelerator features are built on top of the FlashLite Bitmap rendering feature. It improves rendering performance of bitmap objects (source image) onto the frame buffer (destination image).

There are two functions support. First is BitBLT function with effects of Scale, Rotate, Shear and Reflect. The second is Fill function to fill a rectangle in the frame buffer.

5.5.2 Features

- Blit Transformation for source image
- Fill operation to a rectangle in the frame buffer
- Blending and Blitting operation of a source bitmap to the frame buffer
- Support source bitmap format ARGB8888, RGB565, clutRGB_bpp1, bpp2, bpp4 and bpp8
- Support display format ARGB8888, RGB555 and RGB565
- Support effects of Scale, Rotate, Shear and Reflect

5.5.3 Architecture

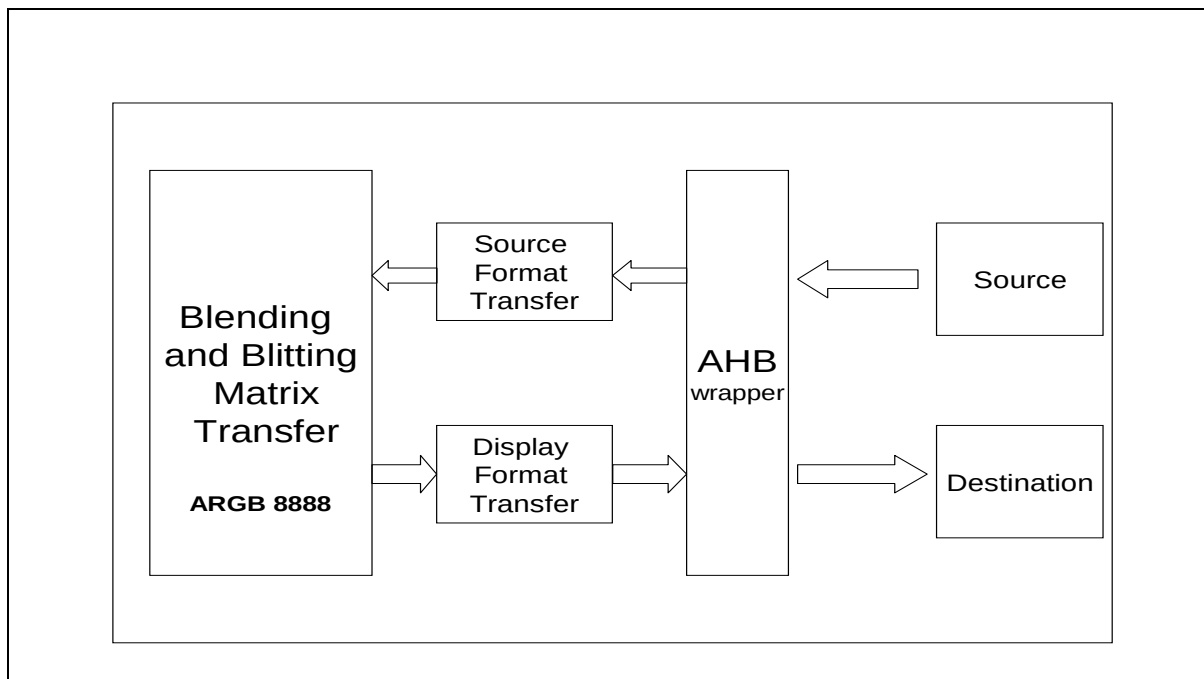


Figure 5.5-1 Architecture Diagram

Data Flow and Structure (2)

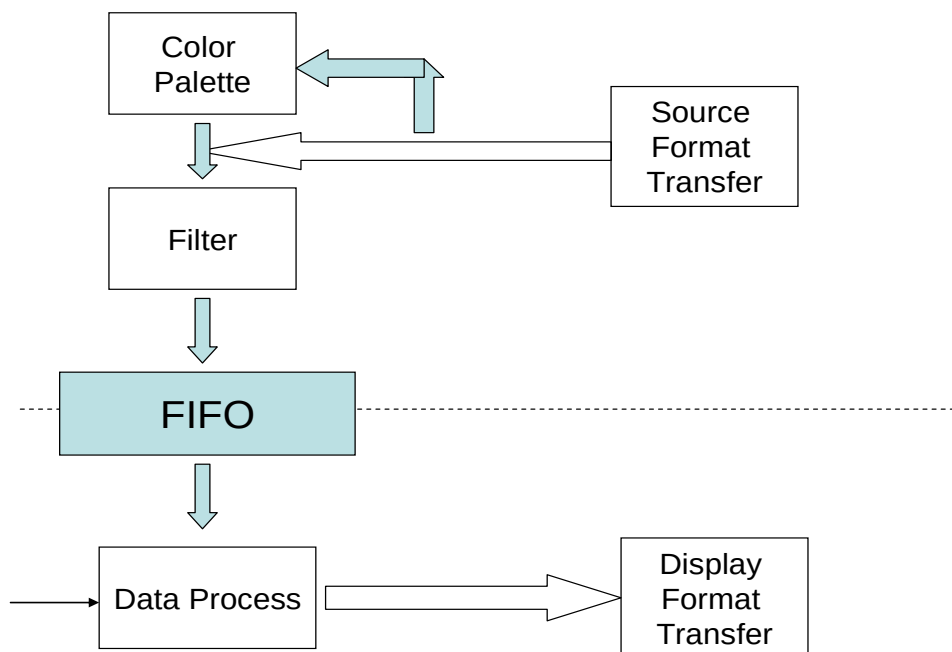


Figure 5.5-2 Architecture Diagram

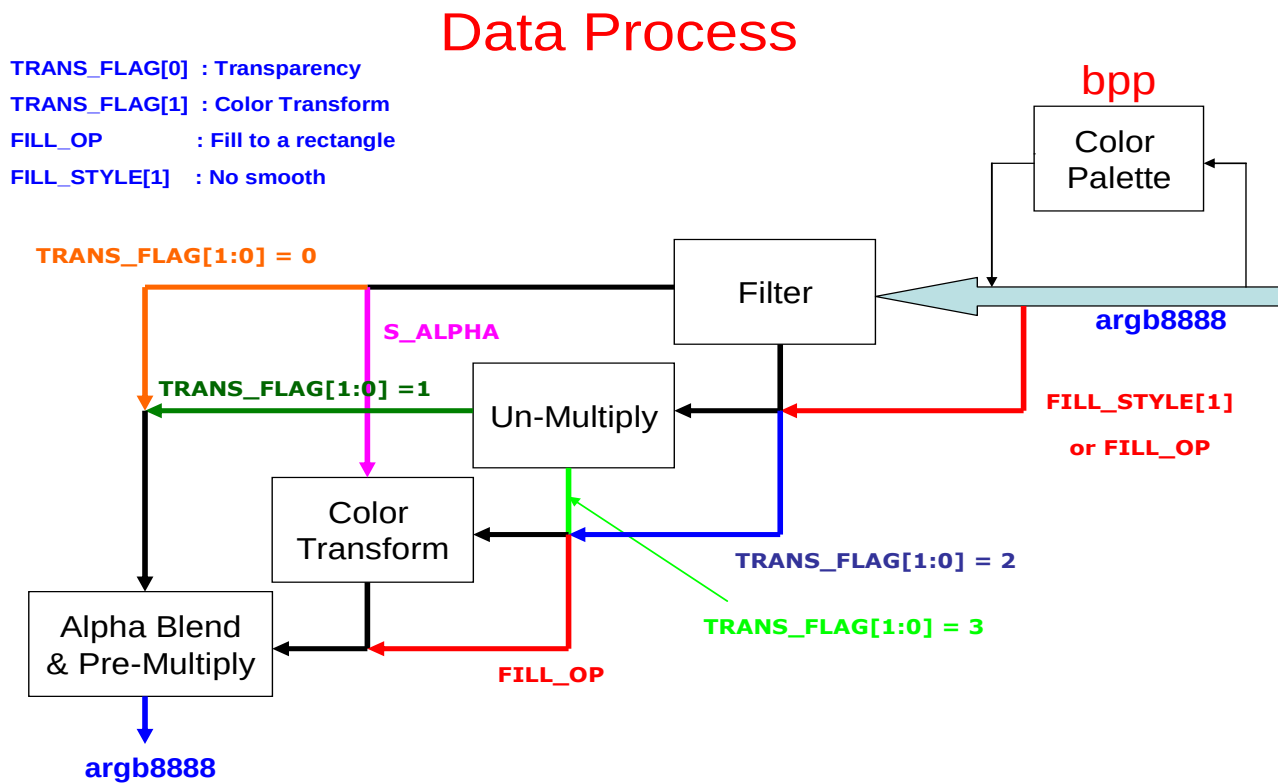


Figure 5.5-3 Architecture Diagram

5.5.4 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W	Description	Reset Value
BLT_BA = 0xB100_D000				
SET2DA	BLT_BA+0x000	R/W	BitBLT Accelerator Enable Set Up Register	0x0000_0000
SFMT	BLT_BA+0x004	R/W	Pixel Format of Source Bitmap Register	0x0000_0000
DFMT	BLT_BA+0x008	R/W	Pixel Format of Destination Bitmap Register	0x0000_0000
BLTINTCR	BLT_BA+0x00C	R/W	BLT Interrupt and Status Control Register	0x0000_0000
MLTA	BLT_BA+0x010	R/W	Alpha Multiplier and Offset Register	0x0000_0000
MLTR	BLT_BA+0x014	R/W	Red Multiplier and Offset Register	0x0000_0000
MLTG	BLT_BA+0x018	R/W	Green Multiplier and Offset Register	0x0000_0000
MLTB	BLT_BA+0x01C	R/W	Blue Multiplier and Offset Register	0x0000_0000
SWIDTH	BLT_BA+0x020	R/W	Width of Source Register	0x0000_0000
SHEIGHT	BLT_BA+0x024	R/W	Height of Source Register	0x0000_0000
DWIDTH	BLT_BA+0x028	R/W	Width of Destination Register	0x0000_0000
DHEIGHT	BLT_BA+0x02C	R/W	Height of Destination Register	0x0000_0000
ELEMENTA	BLT_BA+0x030	R/W	Transform Element A Register	0x0000_0000
ELEMENTB	BLT_BA+0x034	R/W	Transform Element B Register	0x0000_0000
ELEMENTC	BLT_BA+0x038	R/W	Transform Element C Register	0x0000_0000
ELEMENTD	BLT_BA+0x03C	R/W	Transform Element D Register	0x0000_0000
SADDR	BLT_BA+0x040	R/W	Source Remap Start Address Register	0x0000_0000
DADDR	BLT_BA+0x044	R/W	Frame Buffer Address Register	0x0000_0000
SSTRIDE	BLT_BA+0x048	R/W	Source Stride Register	0x0000_0000
DSTRIDE	BLT_BA+0x04C	R/W	Destination Stride Register	0x0000_0000
OFFSETSX	BLT_BA+0x050	R/W	Offset of Source X Register	0x0000_0000
OFFSETS Y	BLT_BA+0x054	R/W	Offset of Source Y Register	0x0000_0000
TRCOLOR	BLT_BA+0x058	R/W	RGB565 Transparent Color Register	0x0000_0000
FILLARGB	BLT_BA+0x060	R/W	ARGB Color Values for Fill Operation Register	0x0000_0000
PALETTE	BLT_BA+0x400 ~ BLT_BA+0x7FF	R/W	Color Palette Register	Un-Define

5.5.5 Register Description

BitBLT Accelerator Enable Set Up Register (SET)

Register	Offset	R/W	Description	Reset Value
SET	BLT_BA+0x000	R/W	Enable Set Up Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				FILL_OP	FILL_STYLE		
7	6	5	4	3	2	1	0
TRCOLOR	TRANS_FLAG			S_ALPHA	FILL_BLEND	L_ENDIAN	BLIT_EN

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	FILL_OP	1 : Fill operation to a rectangle in the frame buffer is request 0 : No Fill operation to a rectangle
[10:8]	FILL_STYLE	0x1 : the bitmap should be clipped to its edge, otherwise a repeating texture. 0x2 : the bitmap should not be smoothed. 0x4 : none fill while destination pixel map outside of source image.
[7]	TRCOLOR	1 : RGB565 Transparent Color Enable 0 : RGB565 Transparent Color Disable
[6:4]	TRANS_FLAG	0x4 : AlphaOnly Only alphaMultiplier needs to be used and rest of the multipliers and offsets should be ignored New alpha value = (old alpha value * alphaMultiplier) + alphaOffset
		0x2 : ColorTransform Compute the color values by the following formula New red value = (old red value * redMultiplier) + redOffset New green value = (old green value * greenMultiplier) + greenOffset New blue value = (old blue value * blueMultiplier) + blueOffset New alpha value = (old alpha value * alphaMultiplier) + alphaOffset
		0x1 : Transparency Per-Pixel Transparency in the Source
[3]	S_ALPHA	Reveal Source Image Alpha during Transparency. 1 : Alpha has no effective Source image has no pre-multiply, no need un-multiply.

		0 : Alpha has effective Source image default has pre-multiply, need un-multiply.
[2]	FILL_BLEND	Alpha blending for Fill Operation, used while Fill_OP = 1 1 : Color to be blended. 0 : Color not to be blended.
[1]	L_ENDIAN	Source format data order for palette index 1 : Little Endian data order 0 : Big Endian data order
[0]	BLIT_EN	Blit a bitmap to the frame buffer using the hardware accelerator 1 : Accelerator operation enable 0 : Accelerator operation disable

Pixel Format of Source Bitmap Register (SFMT)

Register	Offset	R/W	Description	Reset Value
SFMT	BLT_BA+0x004	R/W	Pixel Format of Source Bitmap Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		RGB_bpp8	RGB_bpp4	RGB_bpp2	RGB_bpp1	RGB565	ARGB8888

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5]	RGB_bpp8	BitmapPixelFormat_bpp8_clutRGB 1 : Format select 0 : Format without select
[4]	RGB_bpp4	BitmapPixelFormat_bpp4_clutRGB 1 : Format select 0 : Format without select
[3]	RGB_bpp2	BitmapPixelFormat_bpp2_clutRGB 1 : Format select 0 : Format without select
[2]	RGB_bpp1	BitmapPixelFormat_bpp1_clutRGB 1 : Format select 0 : Format without select
[1]	RGB565	BitmapPixelFormat_bpp16_rgb565 1 : Format select 0 : Format without select
[0]	ARGB8888	BitmapPixelFormat_bpp32_argb8888 1 : Format select 0 : Format without select

Note : To avoid the typo to turn on more than two formats at the same time, the LSB will be the highest priority, as following, the MSB will be the lowest priority.

Pixel Format of Destination Bitmap Register (DFMT)

Register	Offset	R/W	Description	Reset Value
DFMT	BLT_BA+0x008	R/W	Pixel Format of Destination Bitmap Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					DRGB555	DRGB565	DARGB

Bits	Descriptions	
[31:3]	Reserved	Reserved
[2]	DRGB555	DisplayFormat_RGB555_16 (16 bit RGB 0555 format) 1: Format select 0: Format without select
[1]	DRGB565	DisplayFormat_RGB565_16 (16 bit RGB 0565 format) 1: Format select 0: Format without select
[0]	DARGB8888	DisplayFormat_ARGB8888_32 (32 bit ARGB 8888 format) 1: Format select 0: Format without select

Note : To avoid the typo to turn on more than two formats at the same time, the LSB will be the highest priority, as following, the MSB will be the lowest priority.

BLT Interrupt Control and Status Register

Register	Offset	R/W	Description	Reset Value
BLTINTCR	BLT_BA+0x00C	R/W	BLT Interrupt Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					BLT_ERR	BLT_INTE	BLT_INTS

Bits	Descriptions	
[31:3]	Reserved	Reserved
[2]	BLT_ERR	Blitting Error Status 0 = No Error 1 = Error Generated Note: When write value “1” to this bit, the error will be clear.
[1]	BLT_INTE	Blitting Complete Interrupt Enable 0 = Disable 1 = Enable
[0]	BLT_INTS	Blitting Complete Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.

Multiplier and Offset Register (MLT)

Register	Offset	R/W	Description	Reset Value
MLTA	BLT_BA+0x010	R/W	Alpha Multiplier and Offset Register	0x0000_0000
MLTR	BLT_BA+0x014	R/W	Red Multiplier and Offset Register	0x0000_0000
MLTG	BLT_BA+0x018	R/W	Green Multiplier and Offset Register	0x0000_0000
MLTB	BLT_BA+0x01c	R/W	Blue Multiplier and Offset Register	0x0000_0000

31	30	29	28	27	26	25	24
Offset [15:8]							
23	22	21	20	19	18	17	16
Offset [7:0]							
15	14	13	12	11	10	9	8
Multiplier [15:8]							
7	6	5	4	3	2	1	0
Multiplier [7:0]							

Bits	Descriptions	
[31:16]	Offset	Fixed point sign 16 ARGB color offset value
[15:0]	Multiplier	Fixed point sign 8.8 ARGB multiplier value

(Width, Height) of (Source, Destination) Register

Register	Offset	R/W	Description	Reset Value
SWIDTH	BLT_BA+0x020	R/W	Width of Source Register	0x0000_0000
SHEIGHT	BLT_BA+0x024	R/W	Height of Source Register	0x0000_0000
DWIDTH	BLT_BA+0x028	R/W	Width of Destination Register	0x0000_0000
DHEIGHT	BLT_BA+0x02C	R/W	Height of Destination Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
WIDTH[15:8] / HEIGHT[15:8]							
7	6	5	4	3	2	1	0
WIDTH[7:0] / HEIGHT[7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	WIDTH[15:0] /HEIGHT[15:0]	The Width or Height of Source or Destination Register

Transform Element Register

Register	Offset	R/W	Description	Reset Value
ELEMENTA	BLT_BA+0x030	R/W	Transform Element A Register	0x0000_0000
ELEMENTB	BLT_BA+0x034	R/W	Transform Element B Register	0x0000_0000
ELEMENTC	BLT_BA+0x038	R/W	Transform Element C Register	0x0000_0000
ELEMENTD	BLT_BA+0x03C	R/W	Transform Element D Register	0x0000_0000

31	30	29	28	27	26	25	24
ELEMENT [31:24]							
23	22	21	20	19	18	17	16
ELEMENT [23:16]							
15	14	13	12	11	10	9	8
ELEMENT [15:8]							
7	6	5	4	3	2	1	0
ELEMENT [7:0]							

Bits	Descriptions	
[31:0]	ELEMENT[A,B,C,D]	Transform Elements are fixed point 16.16 values.

Source Remap Start Address Register

Register	Offset	R/W	Description	Reset Value
SADDR	BLT_BA+0x040	R/W	Source Remap Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
SADDR [31:24]							
23	22	21	20	19	18	17	16
SADDR [23:16]							
15	14	13	12	11	10	9	8
SADDR [15:8]							
7	6	5	4	3	2	1	0
SADDR [7:0]							

Bits	Descriptions	
[31:0]	SADDR	Source Remap Start Address

Frame Buffer Start Address Register

Register	Offset	R/W	Description	Reset Value
DADDR	BLT_BA+0x044	R/W	Frame Buffer Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
DADDR [31:24]							
23	22	21	20	19	18	17	16
DADDR [23:16]							
15	14	13	12	11	10	9	8
DADDR [15:8]							
7	6	5	4	3	2	1	0
DADDR [7:0]							

Bits	Descriptions	
[31:0]	DADDR	Frame Buffer Start Address

Stride Register

Register	Offset	R/W	Description	Reset Value
SSTRIDE	BLT_BA+0x048	R/W	Source Stride Register	0x0000_0000
DSTRIDE	BLT_BA+0x04C	R/W	Destination Stride Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
STRIDE [15:8]							
7	6	5	4	3	2	1	0
STRIDE [7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	STRIDE	Stride of row bytes. Word alignment is needed for rgb565 and color palette 1, 2, 4 and 8 bits.

Offset of Source (X/Y) Register

Register	Offset	R/W	Description	Reset Value
OFFSETSX	BLT_BA+0x050	R/W	Offset of Source X Register	0x0000_0000
OFFSETSXY	BLT_BA+0x054	R/W	Offset of Source Y Register	0x0000_0000

31	30	29	28	27	26	25	24
OFFSETXY [31:24]							
23	22	21	20	19	18	17	16
OFFSETXY [23:16]							
15	14	13	12	11	10	9	8
OFFSETXY [15:8]							
7	6	5	4	3	2	1	0
OFFSETXY [7:0]							

Bits	Descriptions	
[31:0]	OFFSETXY	The X,Y offset into the source to start rendering from Fixed point sign 16.16 related to original point.

RGB565 Transparent Color Register

Register	Offset	R/W	Description	Reset Value
TRCOLOR	BLT_BA+0x058	R/W	RGB565 Transparent Color Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TRCOLOR [15:8]							
7	6	5	4	3	2	1	0
TRCOLOR [7:0]							

Bits	Descriptions	
[15:0]	TRCOLOR	RGB565 Transparent Color

ARGB Color Values for Fill Operation Register

Register	Offset	R/W	Description	Reset Value
FILLARGB	BLT_BA+0x060	R/W	ARGB Color Values for Fill Operation Register	0x0000_0000

31	30	29	28	27	26	25	24
FILL_A							
23	22	21	20	19	18	17	16
FILL_R							
15	14	13	12	11	10	9	8
FILL_G							
7	6	5	4	3	2	1	0
FILL_B							

Bits	Descriptions	
[31:24]	FILL_A	Alpha Values for Fill to a rectangle Operation.
[23:16]	FILL_R	Red Color Values for Fill to a rectangle Operation.
[15:8]	FILL_G	Green Color Values for Fill to a rectangle Operation.
[7:0]	FILL_B	Blue Color Values for Fill to a rectangle Operation.

Color Palette Index Register

Register	Offset	R/W	Description	Reset Value
PALETTE	BLT_BA+0x400, ~ BLT_BA+0x7FF	R/W	Color Palette Index Register	0x0000_0000

31	30	29	28	27	26	25	24
P_A							
23	22	21	20	19	18	17	16
P_R							
15	14	13	12	11	10	9	8
P_G							
7	6	5	4	3	2	1	0
P_B							

Bits	Descriptions	
[31:24]	P_A	Alpha value of Color Palette
[23:16]	P_R	Red value of Color Palette
[15:8]	P_G	Green value of Color Palette
[7:0]	P_B	Blue value of Color Palette

5.6 JPEG Codec(JPEG)

5.6.1 Overview

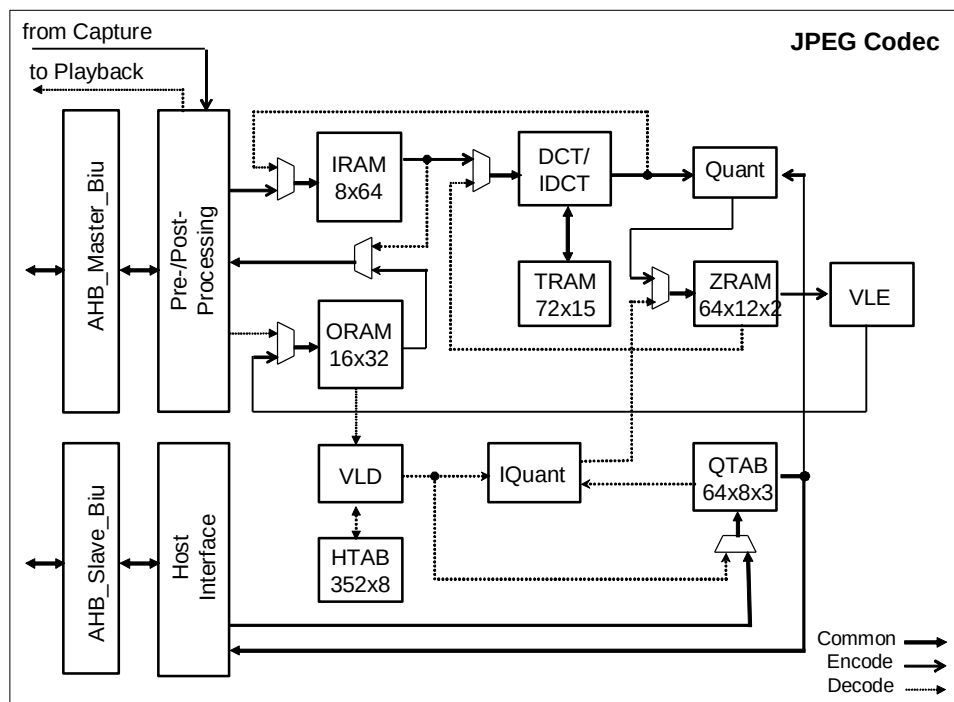


Figure 5.6-1 1 JPEG Codec block diagram

The JPEG Codec supports Baseline Sequential Mode JPEG still image compression and decompression that is fully compliant with ISO/IEC International Standard 10918-1 (T.81). The features and capability of the JPEG codec are listed below.

5.6.2 Features

If image data input or output by planar format (PLANAR_ON= 1), the features are as following:

- Support to encode interleaved YCbCr 4:2:2/4:2:0 and gray-level (Y only) format image
- Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0/4:1:1 and gray-level (Y only) format image
- Support to decode YCbCr 4:2:2 transpose format
- The encoded JPEG bit-stream format is fully compatible with JFIF and EXIF standards
- Support arbitrary width and height image encode and decode
- Support three programmable quantization-tables
- Support standard default Huffman-table and programmable Huffman-table for decode
- Support arbitrarily 1X~8X image up-scaling function for encode mode
- Support down-scaling function for encode and decode modes(Thumbnail/Primary)
- Support specified window decode mode

- Support quantization-table adjustment for bit-rate and quality control in encode mode
- Support rotate function in encode mode
-
- If image data input or output by packet format (PLANAR_ON= 0), the feature are as following:
- Support to encode interleaved YUYV format input image, output bitstream 4:2:2 and 4:2:0 format
- Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0 format image
- Support decoded output image YUYV422, RGB555, RGB565, RGB888 format (ORDER= 1).
- The encoded JPEG bit-stream format is fully compatible with JFIF and EXIF standards
- Support arbitrary width and height image encode and decode(Primary Only)
- Support three programmable quantization-tables
- Support standard default Huffman-table and programmable Huffman-table for decode
- Support arbitrarily 1X~8X image up-scaling function for encode mode
- Support down-scaling function 1X~ 16X for Y422 and Y420, 1X~ 8X for Y444 for decode mode
- Support specified window decode mode
- Support quantization-table adjustment for bit-rate and quality control in encode mode

5.6.3 JPEG Encode

5.6.3.1 Introduction

The JPEG Codec supports Baseline Sequential Mode JPEG still image compression and decompression that is fully compliant with ISO/IEC International Standard 10918-1 (T.81). The JPEG codec also supports the thumbnail image compression for EXIF (Exchangeable image file format for digital still camera, JEIDA). The following description describes the feature of the JPEG encoder. For the DCT-based sequential mode, 8x8 blocks are typically input block-by-block from left to right, and block-row by block-row from up to bottom. Each block is transformed by the forward DCT (FDCT) into a set of 64 values referred to as DCT coefficients. Each of these 64 coefficients is quantized by one of 64 corresponding values selected from the quantization-table. After quantization, the DC coefficient is coded by DPCM algorithm and the 63 AC coefficients are converted into one-dimension zig-zag sequence. Then the run-size symbols are passed to a Huffman encoder for entropy coding and the compressed JPEG bit-stream is generated by variable-length-encoder (VLE).

The JPEG encoder supports interleaved YUV422, YUV420 format and non-interleaved Y-component only format if planar format set and packet YUYV format if packet format set. Besides the standard compression, the JPEG encoder integrates a pre-processing unit that can scale-up or scale-down the source image in horizontal and vertical directions.

5.6.3.2 JPEG Encode Operation

The JPEG encoder supports single compression mode and continuous compression mode. In single mode, the programmer can use dual-buffer or fix-buffer to store JPEG bit-stream. In continue mode, the programmer can store neighbor JPEG bit-stream in Frame Memory continuously. The JPEG encoder also supports thumbnail image encode.

The JPEG Codec can encode the image with three components (Y, Cb, Cr) or Y component only, where Y component represents the luminance information, and Cb & Cr represent the chrominance information in planar format. It also can encode packet YUYV in packet format. The three components

are stored in frame memory separately. The JPEG Codec can compress YUV 420 or YUV 422 format by programming the control register bit **EY422**. The control registers **JYADDR0**, **JUADDR0**, **JVADDR0**, **JYADDR1**, **JUADDR1**, **JVADDR1** specify the memory starting address (buffer-0 & buffer-1) of Y, Cb and Cr components and packet data. The control registers **JYSTRIDE**, **JUSTRIDE**, **JVSTRIDE** specify the stride that is the address distance between adjacent lines for each component. The control registers **IO_IADDR0**, **IO_IADDR1** specify the memory starting address (buffer-0 & buffer-1) for the JPEG bit-stream. The following figure depicts the source image starting address and stride.

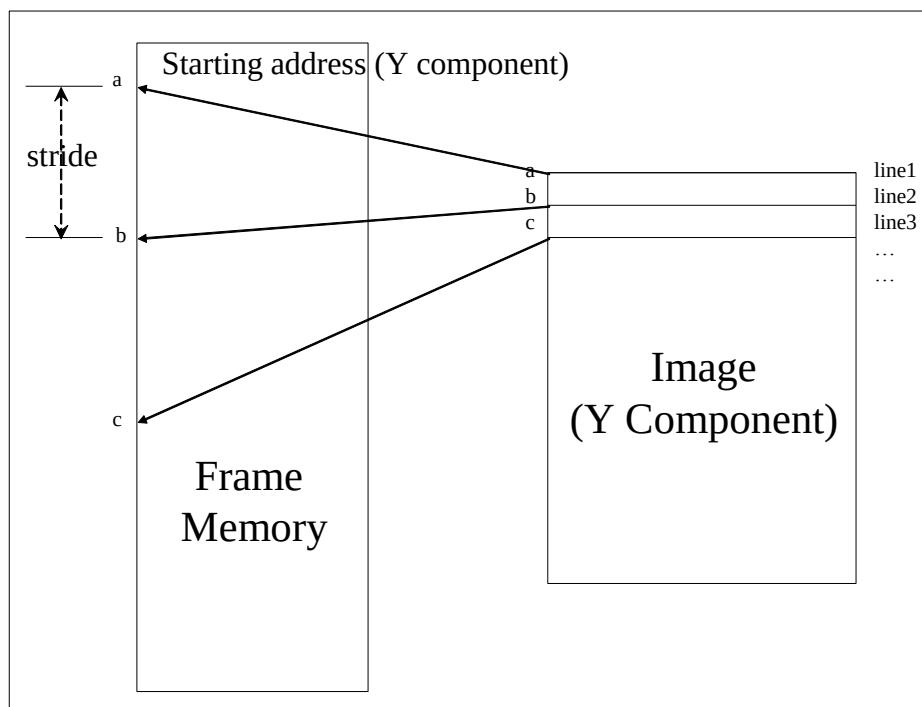


Figure 5.6.2 Image starting address and stride

The JPEG Codec supports thumbnail encode, the programmer can set the register bit **THB** for thumbnail encode. It will automatically trigger JPEG engine twice, where the first encode operation is for thumbnail image, and the second one is for primary image. The following figure specifies the encode path. When the programmer turns on thumbnail encode, the JPEG engine supports an option for inserting one buffer region into primary JPEG bit-stream. This option can be turned on by setting the register bit **A_JUMP**. The buffer size can be programmed by specifying the registers **JRESERVE**. In general, the buffer is used to store the thumbnail JPEG bit-stream and some information about this encoded image. The starting address of the JPEG bit-stream for thumbnail image is equal to **IO_IADDR** plus an offset size specified by register **JOFFSET**. When the encode operation is completed, the programmer can get the size information of the encoded JPEG bit-stream by reading the registers **JPRI_SIZE** and **JTHB_SIZE**.

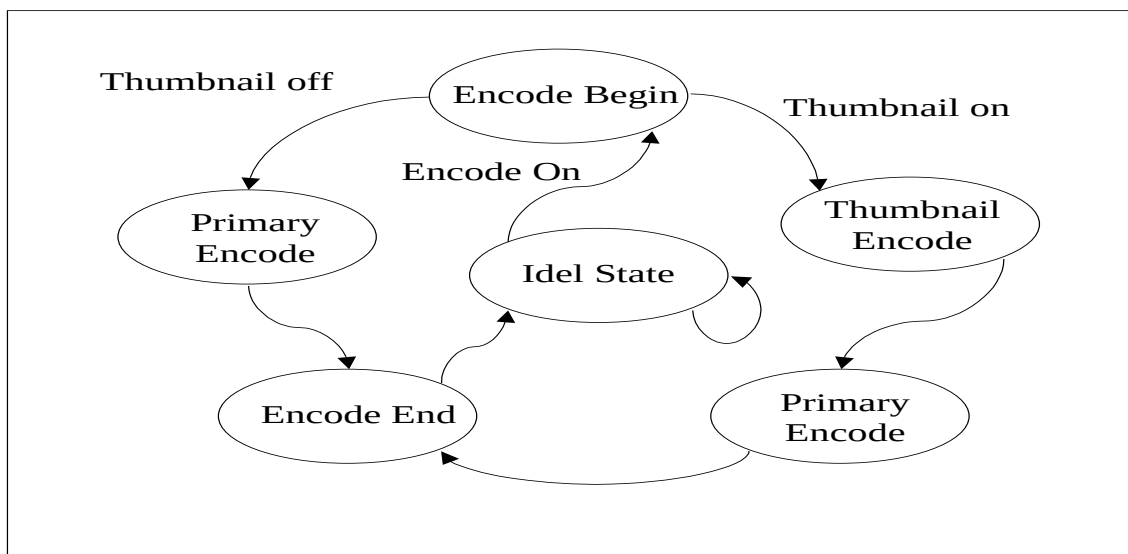


Figure 5.6-2.3 Primary and thumbnail encode

The JPEG Codec supports the planar format up-scaling and down-scaling function to adjust the encoded image size. The primary image encode supports 1~8X arbitrarily up-scaling in horizontal and vertical direction. But the thumbnail image encode doesn't support the up-scaling function. The JPEG Codec also supports the planar format down-scaling function with the 1/2, 1/3, 1/4, ..., 1/64 ratio in vertical direction, and 1/2, 1/4, 1/6, 1/8, ..., 1/62, 1/64 ratio in horizontal direction. The programmer can set the registers **JPSCALU**, **JPSCALD**, **JTSCALD** for image scaling up or down. The registers **JPRIWH** and **JTHBWH** specify the width and height of the encoded image after scaling. For planar format up-scaling mode, the programmer needs to specify the up-scale ratio by registers **JUPRAT** and the source image height register **JSRCH**. The rotation function is only supported for planar format. The source image can be encoded by rotate left or right 90°. It should be mentioned that the rotation function can only be applied only when encode YCbCr 4:2:0 source.

The standard JPEG bit-stream format includes some headers that specify some information, For example, Quantization-table (QTAB) and Huffman-table (HTAB) belong to the part of the header. The JPEG Codec supports some options whether you can insert these headers into the JPEG bit-stream or not. These options are defined in the control register **JHEADER**. The JPEG codec also supports quantization-table adjustment to control the size of JPEG bit-stream. When the bit-stream size is too large, it can set or adjust the value of quantization-table to reduce the bit-stream size. The adjustment control is defined in registers **JPRIQC** and **JTHBQC**. In addition, three programmable quantization-tables are provided. The programmer can specify using two or three tables for encoding by register bit **E3QTAB**. The Quantization-table registers are defined in **JQTAB0~JQTAB2**.

The JPEG Codec supports two coding modes: single mode and continue mode. For single mode, the programmer can use dual-buffer or fix-buffer to store JPEG bit-stream in frame memory. For continue mode, the programmer can store the neighbor JPEG bit-streams into frame memory continuously or store each JPEG bit-stream into frame memory by creating same buffer size that is specified in register **JFSTRIDE**. The following figure specifies these two encode modes.

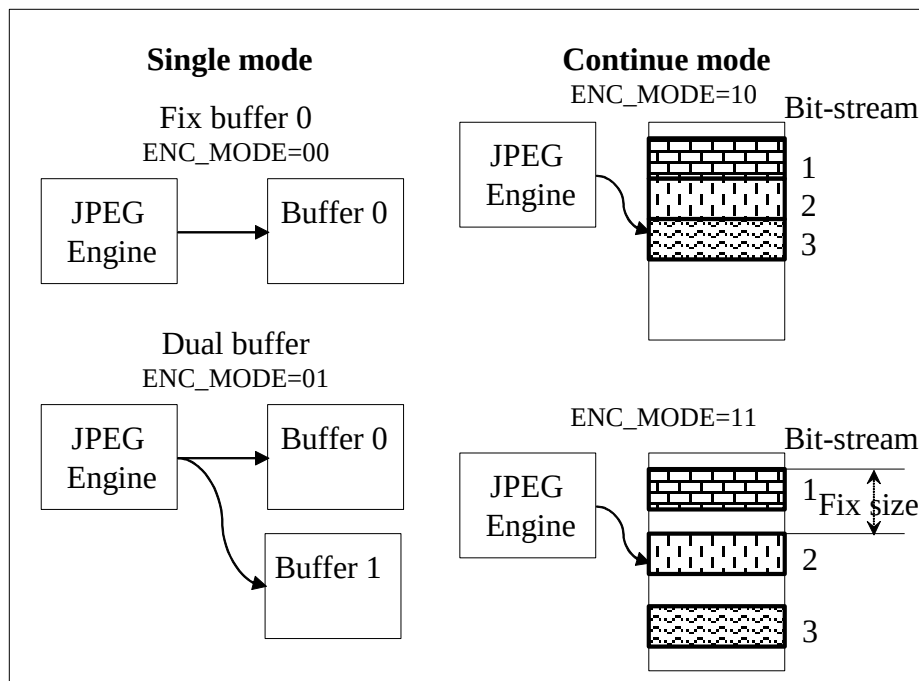


Figure 5.6-3. 4 Single mode and continue mode

5.6.3.3 JPEG Encode Programming

The programming flow for JPEG encode operation is described as follows:

The programmer needs to program all the required control registers parameters, like image format, width, height, header format, quantization-table, scaling-factor, memory address, etc. before triggering the JPEG engine. Apply engine soft reset by setting reset bit **ENG_RST** to 1 and then to 0. For planar format, **PLANAR_ON** set to 1. For single mode, the programmer can trigger the JPEG engine once by setting the engine enable bit **JPG_EN** to 1 and then to 0. For continue mode, the JPEG engine will continually operate if the engine enable bit **JPG_EN** is kept in 1, and will stop operate when **JPG_EN** is set to 0 and the current picture is encoded completely. When the encode operation for one picture (with both primary and thumbnail images if thumbnail encode is enabled) is complete, the JPEG codec will issue an interrupt to host. For packet format, **PLANAR_ON** set to 0.

5.6.4 JPEG Decode

5.6.4.1 Introduction

The JPEG decoding operation is very similar to the feedback loop of the JPEG encoding. After operation is triggered, the JPEG bit-stream is fetched from frame memory and processed by the variable-length decoder (VLD). The decoded data are parsed, inverse zig-zag scanned (IZZ), inverse quantization (IQ) and inverse DCT (IDCT). An offset value is added to the output data of IDCT to become the reconstruction picture.

The JPEG decoder also integrates a post-processing unit that can apply some post-processing to the decoded image. It can scale-down the image in horizontal and vertical directions.

5.6.4.2 JPEG Decode Operation

The JPEG decoder can decode the JPEG bit-stream that is baseline JPEG format. When the programmer want to decode JPEG bit-stream, he just needs to specify the starting address of the

JPEG bit-stream in frame memory by registers **JIOADDR0** or **JIOADDR1** and set JPEG decode mode by register bit **ENC_DEC**, and then trigger the JPEG engine. When the decode operation is complete, the JPEG engine will issue an interrupt to host. The status register **DYUVMODE** reports the image color format (4:4:4, 4:2:2, 4:2:0, or 4:1:1) that has been decoded. The register **JDECWH** reports the original width & height of the decoded JPEG image. The register **JPRIWH** should be written the real size image width & height after scaling or not for packet format. The programmer can also get the quantization-table of the decoded JPEG bit-stream by reading the registers **JQTAB0~JQTAB2**. For RGB555 packet output, set ORDER to 1 in the register **JITCR**.

For 4:4:4 color format images, if the original width is not multiple-of-8, the decoded image will be padded to multiple-of-8. For 4:2:2 and 4:2:0 color format images, if the original width is not multiple-of-16, the decoded image will be padded to multiple-of-16. For 4:1:1 color format images, if the original width is not multiple-of-32, the decoded image will be padded to multiple-of-32.

The JPEG decoder supports the programmable Huffman-table function and can decode the bitstream that is coded by the user-defined Huffman-table. The programmer can choose to turn-on the programmable Huffman-table function or directly use the default Huffman-table to decode the JPEG bitstream by setting the register bit **PDHTAB**.

The JPEG decoder supports 1/2, 1/4 and 1/8 planar format down-scaling in horizontal and vertical direction to adjust the decoded image size. The programmer can specify the register **JPSCALD** for image planar format down-scaling function. The JPEG decoder also supports 1~16X arbitrarily packet format down-scaling in horizontal and vertical direction. For planar format down-scaling mode, the programmer needs to specify the down-scale ratio by registers **JUPRAT**.

The JPEG decoder supports specified window decode mode. This function allows user to specify a sub-window region within the whole image to be decoded as shown in the following figure. Only the specified window region image will be decoded and stored to frame memory. This function can be enabled by setting register bit **WIN_DEC**, and the window region can be specified in registers **JWINDEC0~JWINDEC2**.

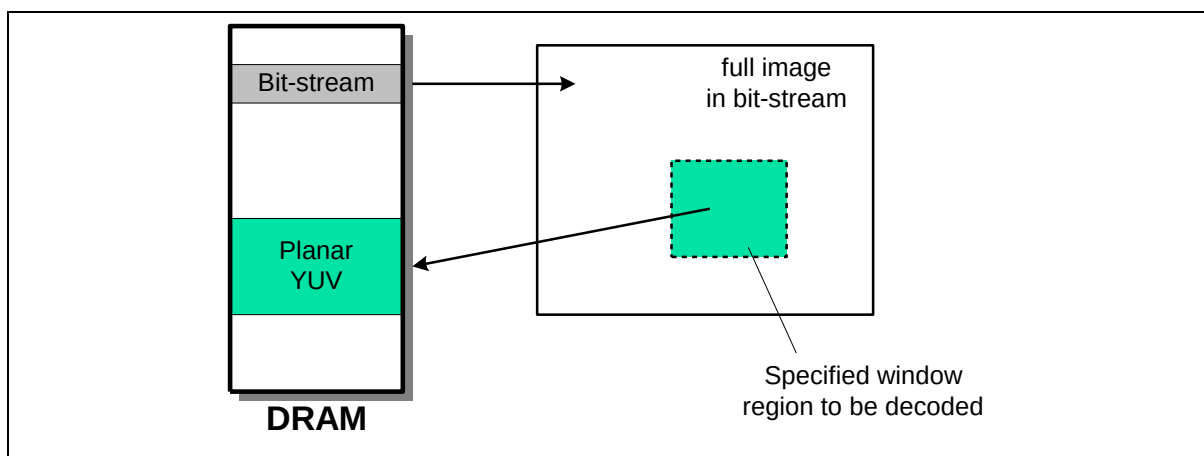


Figure 5.6-4.Specified window decode mode

When the “header decode complete wait” bit **DHEAD** is set, JPEG will enter the pending state after the header information has been decoded, and will resume the decode operation after the interrupt status is cleared.

5.6.5 JPEG Codec Interrupt

The JPEG codec supports encode complete, decode complete, encode error, decode error and header decode complete interrupts. When the encode or decode operation is complete with no error, an encode or decode complete interrupt will be issued to host. When decode bitstream and some error occurs, a decode error interrupt will be issued to host. When DHEAD is set and the bitstream

header has been decoded, a header decode complete interrupt will be issued to host. The interrupt status is reflected on register **JINTCR**.

5.6.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W/C	Description	Reset Value
JPG_BA = 0xB100_A000				
JMCR	JPG_BA + 000	R/W	JPEG Engine Mode Control Register	0x0000_0000
JHEADER	JPG_BA + 004	R/W	JPEG Encode Header Control Register	0x0000_0000
JITCR	JPG_BA + 008	R/W	JPEG Image Type Control Register	0x0000_0000
RESERVED	JPG_BA + 00C	R/W	Reserved	0x0000_0000
JPRIQC	JPG_BA + 010	R/W	JPEG Encode Primary Q-Table Control Register	0x0000_00F4
JTHBQC	JPG_BA + 014	R/W	JPEG Encode Thumbnail Q-Table Control Register	0x0000_00F4
JPRIWH	JPG_BA + 018	R/W	JPEG Primary Width/Height Register	0x0000_0000
JTHBWH	JPG_BA + 01C	R/W	JPEG Encode Thumbnail Width/Height Register(For Planar Format Only)	0x0000_0000
JPRST	JPG_BA + 020	R/W	JPEG Encode Primary Restart Interval Register	0x0000_0004
JTRST	JPG_BA + 024	R/W	JPEG Encode Thumbnail Restart Interval Register	0x0000_0004
JDECWH	JPG_BA + 028	R	JPEG Decode Image Width/Height Register	0x0000_0000
JINTCR	JPG_BA + 02C	R/W	JPEG Interrupt Control and Status Register	0x0020_0000
RESERVED	JPG_BA + 030 ~ JPG_BA + 03C	R/W	Reserved	0x0000_0000
JTEST	JPG_BA + 040	R/W	JPEG Test Control Register	0x0000_0000
JWINDEC0	JPG_BA + 044	R/W	JPEG Window Decode Mode Control Register 0	0x0000_0000
JWINDEC1	JPG_BA + 048	R/W	JPEG Window Decode Mode Control Register 1	0x0000_0000
JWINDEC2	JPG_BA + 04C	R/W	JPEG Window Decode Mode Control Register 2	0x0000_0000
JMACR	JPG_BA + 050	R/W	JPEG Memory Address Mode Control Register	0x0000_0000
JPSCALU	JPG_BA + 054	R/W	JPEG Primary Scaling-Up Control Register	0x0000_0000
JPSCALD	JPG_BA + 058	R/W	JPEG Primary Scaling-Down Control Register	0x0000_0000
JTSCALD	JPG_BA + 05C	R/W	JPEG Thumbnail Scaling-Down Control Register	0x0000_0000
JDBCR	JPG_BA + 060	R/W	JPEG Dual-Buffer Control Register	0x0000_0000
RESERVED	JPG_BA + 064 ~ JPG_BA + 06C	R/W	Reserved	0x0000_0000
JRESERVE	JPG_BA + 070	R/W	Primary Encode Bit-stream Reserved Size Register	0x0000_0000
JOFFSET	JPG_BA + 074	R/W	Address Offset Between Primary/Thumbnail Register	0x0000_0000
JFSTRIDE	JPG_BA + 078	R/W	JPEG Encode Bit-stream Frame Stride Register	0x0000_0000

JYADDR0	JPG_BA + 07C	R/W	Y Component or Packet Format Frame Buffer-0 Start Address Register	0x0000_0000
JUADDR0	JPG_BA + 080	R/W	U Component Frame Buffer-0 Start Address Register	0x0000_0000
JVADDR0	JPG_BA + 084	R/W	V Component Frame Buffer-0 Start Address Register	0x0000_0000
JYADDR1	JPG_BA + 088	R/W	Y Component or Packet Format Frame Buffer-1 Start Address Register	0x0000_0000
JUADDR1	JPG_BA + 08C	R/W	U Component Frame Buffer-1 Start Address Register	0x0000_0000
JVADDR1	JPG_BA + 090	R/W	V Component Frame Buffer-1 Start Address Register	0x0000_0000
JYSTRIDE	JPG_BA + 094	R/W	Y Component Frame Buffer Stride Register	0x0000_0000
JUSTRIDE	JPG_BA + 098	R/W	U Component Frame Buffer Stride Register	0x0000_0000
JVSTRIDE	JPG_BA + 09C	R/W	V Component Frame Buffer Stride Register	0x0000_0000
JIOADDR0	JPG_BA + 0A0	R/W	Bit-stream Frame Buffer-0 Start Address Register	0x0000_0000
JIOADDR1	JPG_BA + 0A4	R/W	Bit-stream Frame Buffer-1 Start Address Register	0x0000_0000
JPRI_SIZE	JPG_BA + 0A8	R	JPEG Encode Primary Bit-stream Size Register	0x0000_0000
JTHB_SIZE	JPG_BA + 0AC	R	JPEG Encode Thumbnail Bit-stream Size Register	0x0000_0000
JUPRAT	JPG_BA + 0B0	R/W	JPEG Planar Format Encode Up-Scale Ratio and Packet Format Decode Down-Scale Ratio	0x0000_0000
JBSFIFO	JPG_BA + 0B4	R/W	JPEG Bit-stream FIFO Control Register	0x0000_0032
JSRCH	JPG_BA + 0B8	R/W	JPEG Encode Source Image Height	0x0000_0FFF
RESERVED	JPG_BA + 0BC ~ JPG_BA + 0FC	R/W	Reserved	0x0000_0000
JQTAB0	JPG_BA + 100 ~ JPG_BA + 13F	R/W	JPEG Quantization-Table 0	0x0000_0000
JQTAB1	JPG_BA + 140 ~ JPG_BA + 17F	R/W	JPEG Quantization-Table 1	0x0000_0000
JQTAB2	JPG_BA + 180 ~ JPG_BA + 1BF	R/W	JPEG Quantization-Table 2	0x0000_0000
RESERVED	JPG_BA + 1C0 ~ JPG_BA + 1FC	R/W	Reserved	0x0000_0000

5.6.7 Register Description

JPEG Engine Mode Control Register (JMCR)

Register	Address	R/W	Description	Default Value
JMCR	JPG_BA + 000	R/W	JPEG Engine Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						RESUMEI	RESUMEEO
7	6	5	4	3	2	1	0
ENC_DEC	WIN_DEC	PRI	THB	EY422	QT_BUSY	ENG_RST	JPG_EN

Bits	Descriptions	
[31:9]	Reserved	Reserved
[9]	RESUMEI	Resume JPEG Operation for Input On-the-Fly Mode Write a "1" to this bit to restart JPEG from a pending state after an input wait interrupt event occurs. This bit will be automatically set to "0" after JPEG receives it. Note: This bit can be set when the next source data is filled into frame buffer by host no matter JPEG is pending or not.
[8]	RESUMEEO	Resume JPEG Operation for Output On-the-Fly Mode Write a "1" to this bit to restart JPEG from a pending state after an output wait interrupt event occurs. This bit will be automatically set to "0" after JPEG receives it. Note: This bit can be set when the JPEG generated data is fetched from frame buffer by host no matter JPEG is pending or not.
[7]	ENC_DEC	JPEG Encode/Decode Mode 0 = Decode 1 = Encode
[6]	WIN_DEC	JPEG Window Decode Mode 0 = Disable, decode full image 1 = Enable, only the window region defined by registers JWINDEC of the whole image will be decoded Note: This bit is only valid when JPEG engine is operates in decode mode. If window decode mode is enabled, the up-scaling and down-scaling functions are not allowed.
[5]	PRI	Encode Primary Image 0 = Disable encoding primary image 1 = Enable encoding primary image
[4]	THB	Encode Thumbnail Image 0 = Disable encoding thumbnail image 1 = Enable encoding thumbnail image

[3]	EY422	Encode Image Format 0 = YUV 4:2:0 1 = YUV 4:2:2
[2]	QT_BUSY	Quantization-Table Busy Status (Read-Only) 0 = Quantization-Table is ready for host access 1 = Quantization-Table is busy and can't be accessed
[1]	ENG_RST	Soft Reset JPEG Engine (Except JPEG Control Registers) 0 = Disable. Normal operation 1 = Reset JPEG engine, but the value of control registers keep no change
[0]	JPG_EN	JPEG Engine Operation Control 0 = Disable 1 = Enable

JPEG Encode Header Control Register (JHEADER)

Register	Address	R/W	Description	Default Value
JHEADER	JPG_BA + 004	R/W	JPEG Encode Header Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
P_JFIF	P_HTAB	P_QTAB	P_DRI	T_JFIF	T_HTAB	T_QTAB	T_DRI

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7]	P_JFIF	Primary JPEG Bit-stream Include JFIF Header 0 = Not Include 1 = Include
[6]	P_HTAB	Primary JPEG Bit-stream Include Huffman-Table 0 = Not Include 1 = Include
[5]	P_QTAB	Primary JPEG Bit-stream Include Quantization-Table 0 = Not Include 1 = Include
[4]	P_DRI	Primary JPEG Bit-stream Include Restart Interval 0 = Not Include 1 = Include
[3]	T_JFIF	Thumbnail JPEG Bit-stream Include JFIF Header 0 = Not Include 1 = Include
[2]	T_HTAB	Thumbnail JPEG Bit-stream Include Huffman-Table 0 = Not Include 1 = Include
[1]	T_QTAB	Thumbnail JPEG Bit-stream Include Quantization-Table 0 = Not Include 1 = Include

[0]	T_DRI	Thumbnail JPEG Bit-stream Include Restart Interval 0 = Not Include 1 = Include

JPEG Image Type Control Register (JITCR)

Register	Address	R/W	Description	Default Value
JITCR	JPG_BA + 008	R/W	JPEG Image Type Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							ARGB8888
15	14	13	12	11	10	9	8
PLANAR_ON	ORDER	RGB_555_565	ROTATE		DYUV_MODE		
7	6	5	4	3	2	1	0
EXIF	EY_ONLY	DHEND	DTHB	E3QTAB	D3QTAB	ERR_DIS	PDHTAB

Bits	Descriptions	
[31:17]	Reserved	Reserved
[16]	ARGB8888	1 = ARGB8888 while ORDER = 1
[15]	PLANAR_ON	0 = Packet format 1 = Planar format
[14]	ORDER	Decode Packet format Output Data Order (low byte first) 0 = Y0 U0 Y1 V0 1 = RGB
[13]	RGB_555_565	1 = RGB565, while ORDER = 1 0 = RGB555, while ORDER = 1
[12:11]	ROTATE	Encode Image Rotate(For Planar Format only) 00 = normal encode 10 = The encoded image is rotated left from source image 11 = The encoded image is rotated right from source image Note: The JPRIWH and JTHBWH specify the image width and height after rotation. However the JPSCALD, JTSCALD and JUPRAT specify the scale ratio before rotation.
[10:8]	DYUV_MODE	Decoded Image YUV Color Format (Read-Only) 000 = The format of decoded image is YUV 4:2:0 001 = The format of decoded image is YUV 4:2:2 010 = The format of decoded image is YUV 4:4:4 011 = The format of decoded image is YUV 4:1:1 100 = The format of decoded image is gray-level (Y only) 101 = The format of decoded image is YUV 4:2:2 transpose

[7]	EXIF	Encode Quantization-Table & Huffman-Table Header Format Selection 0 = General format. The header QTAB/HTAB for each component is defined in separated DQT/DHT marker. 1 = EXIF compatible format. Three QTAB are defined for Y, Cb, and Cr, header QTAB/HTAB is defined in only one DQT/DHT marker.
[6]	EY_ONLY	Encode Gray-level (Y-component Only) Image 0 = Encode normal Y/Cb/Cr color image 1 = Encode gray-level image
[5]	DHEND	Header Decode Complete Stop Enable 0 = JPEG engine will not stop after the header information of JPEG bitstream has been decoded 1 = JPEG engine will enter the pending state after the header information of JPEG bitstream has been decoded. Clear header-decode-end interrupt status can resume JPEG decoding operation
[4]	DTHB	Decode Thumbnail Image Only 0 = Decode primary image 1 = Decode thumbnail image only Note: If the JPEG bit-stream contains thumbnail, the programmer can select to decode the primary image or decode thumbnail image only by this bit.
[3]	E3QTAB	Numbers of Quantization-Table are Used For Encode 0 = Two QTAB, one for Y and another for Cb & Cr. 1 = Three QTAB, one for Y, one for Cb, and one for Cr. Note: If EXIF is enable, three QTAB are always used for Y, Cb, and Cr.
[2]	D3QTAB	Numbers of Quantization-Table are Used For Decode (Read-Only) 0 = Two QTAB 1 = Three QTAB
[1]	ERR_DIS	Decode Error Engine Abort 0 = JPEG decode operation will abort if decode error occurs 1 = JPEG decode operation will continue if decode error occurs
[0]	PDHTAB	Programmable Huffman-Table Function For Decode 0 = Disable. Use default huffman-table for JPEG decode 1 = Enable. Allow user-defined Huffman-table in JPEG bit-stream

Register	Address	R/W	Description	Default Value
JPRIQC	JPG_BA + 010	R/W	JPEG Primary Q-Table Control Register	0x0000_00F4

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
P_QADJUST				P_QVS			

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:4]	P_QADJUST	Primary Quantization-Table Adjustment If the sum of the position (x, y) of quantization-table is greater than P_QADJUST, the quantization value will be set to 127. Otherwise the value will keep as the original. 8x8 DCT block: x = 0~7, y = 0~7 if ((x+y) > P_QADJUST) => Q' = 127 else => Q' = Q
[3:0]	P_QVS	Primary Quantization-Table Scaling Control $Q' = (P\_QVS[3]*2*Q)+(P\_QVS[2]*Q)+(P\_QVS[1]*Q/2)+(P\_QVS[0]*Q/4)$

JPEG Encode Thumbnail Quantization-table Control Register (JTHBQC)

Register	Address	R/W	Description	Default Value
JTHBQC	JPG_BA + 014	R/W	JPEG Thumbnail Q-Table Control Register	0x0000_00F4

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
T_QADJUST				T_QVS			

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:4]	T_QADJUST	Thumbnail Quantization-Table Adjustment If the sum of the position (x, y) of quantization-table is greater than T_QADJUST, the quantization value will be set to 127. Otherwise the value will keep as the original. 8x8 DCT block: x = 0~7, y = 0~7 if ((x+y) > T_QADJUST) => Q' = 127 else => Q' = Q
[3:0]	T_QVS	Thumbnail Quantization-Table Scaling Control $Q' = (T\_QVS[3]*2*Q) + (T\_QVS[2]*Q) + (T\_QVS[1]*Q/2) + (T\_QVS[0]*Q/4)$

JPEG Primary Image Width/Height Register (JPRIWH)

Register	Address	R/W	Description	Default Value
JPRIWH	JPG_BA + 018	R/W	JPEG Primary Width/Height Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				P_HEIGHT[11:8]			
23	22	21	20	19	18	17	16
P_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved				P_WIDTH[11:8]			
7	6	5	4	3	2	1	0
P_WIDTH[7:0]							

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27:16]	P_HEIGHT	Primary Encode and Packet Format Decode Image Height A 12-bit value specifies the height of encoded and decoded JPEG image. The value is equal to the size after scaling-up or scaling-down. Note: The JPEG engine supports horizontal and vertical arbitrarily up-scaling 1X~8X in planar format encode mode. When the vertical up-scaling mode (Y2) is enabled, the height of source image needs to be specified by JCRCH. When the vertical down-scaling mode is enable in packet format decode, the size is $\text{ceil}((\text{YSF}/1024) * (\text{height of image}))$
[15:12]	Reserved	Reserved
[11:0]	P_WIDTH	Primary Encode and Packet Format Decode Image Width A 12-bit value specifies the width of encoded and decoded JPEG image. The value is equal to the size after scaling-up or scaling-down. When the down-scaling mode is enable in packet format decode, the size is $\text{ceil}((\text{XSF}/1024) * (\text{width of image})/16) * (16 + \text{Block1})$, while $\text{Block1} = 1$ when $\text{mod}((\text{XSF}/1024) * (\text{width of image})/16)$ is 0

JPEG Encode Thumbnail Image Width/Height Register (JTHBWH) (for Planar Format Only)

Register	Address	R/W	Description	Default Value
JTHBWH	JPG_BA + 01C	R/W	JPEG Encode Thumbnail Width/Height Register (For Planar Format Only)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				T_HEIGHT[11:8]			
23	22	21	20	19	18	17	16
T_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved				T_WIDTH[11:8]			
7	6	5	4	3	2	1	0
T_WIDTH[7:0]							

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27:16]	T_HEIGHT	Thumbnail Encode Image Height (For Planar Format Only) A 12-bit value specifies the height of encoded JPEG thumbnail image. The value is equal to the size after scaling-down.
[15:12]	Reserved	Reserved
[11:0]	T_WIDTH	Thumbnail Encode Image Width (For Planar Format Only) A 12-bit value specifies the width of encoded JPEG thumbnail image. The value is equal to the size after scaling-down.

JPEG Encode Primary Restart Interval Value Register (JPRST)

Register	Address	R/W	Description	Default Value
JPRST	JPG_BA + 020	R/W	JPEG Encode Primary Restart Interval Register	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
P_RST[7:0]							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	P_RST	Primary Encode Restart Interval Value An 8-bit value specifies the restart interval for encoding primary JPEG image.

JPEG Encode Thumbnail Restart Interval Value Register (JTRST)

Register	Address	R/W	Description	Default Value
JTRST	JPG_BA + 024	R/W	JPEG Encode Thumbnail Restart Interval	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
T_RST[7:0]							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	T_RST	Thumbnail Encode Restart Interval Value An 8-bit value specifies the restart interval for encoding thumbnail JPEG image.

JPEG Decode Image Width/Height Register (JDECWH)

Register	Address	R/W	Description	Default Value
JDECWH	JPG_BA + 028	R	JPEG Decode Image Width/Height Register	0x0000_0000

31	30	29	28	27	26	25	24
DEC_HEIGHT[15:8]							
23	22	21	20	19	18	17	16
DEC_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
DEC_WIDTH[15:8]							
7	6	5	4	3	2	1	0
DEC_WIDTH[7:0]							

Bits	Descriptions	
[31:16]	DEC_HEIGHT	Decode Image Height A 16-bit value reports the height of decoded JPEG image.
[15:0]	DEC_WIDTH	Decode Image Width A 16-bit value reports the width of decoded JPEG image. Note: 1. The value of width and height are extracted from bitstream header and are not the width and height after up-scaling or down-scaling. 2. The real decoded image width (stored to DRAM) will be aligned to multiple of 16 for 4:2:2/4:2:0, and multiple of 8 for 4:4:4/y-only. The real decoded image height (stored to DRAM) will be aligned to multiple of 16 for 4:2:0, and multiple of 8 for 4:4:4/4:2:2/y-only. 3. If up-scaling or down-scaling function is enabled, the real image width/height (stored to DRAM) is equal to aligned width/height (described above) * scaling-factor.

JPEG Interrupt Control and Status Register (JINTCR)

Register	Address	R/W	Description	Default Value
JINTCR	JPG_BA + 02C	R/W	JPEG Interrupt Control and Status Register	0x0020_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
JPG_WAITI	JPG_WAITO	Reserved					BAabort
15	14	13	12	11	10	9	8
Reserved	DHE_INTE	IPW_INTE	OPW_INTE	ENC_INTE	DEC_INTE	DER_INTE	EER_INTE
7	6	5	4	3	2	1	0
Reserved	DHE_INTS	IPW_INTS	OPW_INTS	ENC_INTS	DEC_INTS	DER_INTS	EER_INTS

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23]	JPG_WAITI	JPEG Input Wait Status (Read-Only) 0 = JPEG is operating or idle 1 = JPEG is pending and is waiting for a input resume
[22]	JPG_WAITO	JPEG Output Wait Status (Read-Only) 0 = JPEG is operating or idle 1 = JPEG is pending and is waiting for a output resume
[21:17]	Reserved	Reserved
[16]	BAabort	JPEG Memory Access Error Status (Read-Only) 0 = Normal operation 1 = Wrong frame memory space is accessed
[15]	Reserved	Reserved
[14]	DHE_INTE	JPEG Header Decode End Wait Interrupt Enable 0 = Disable 1 = Enable
[13]	IPW_INTE	Input Wait Interrupt Enable 0 = Disable 1 = Enable
[12]	OPW_INTE	Output Wait Interrupt Enable 0 = Disable 1 = Enable
[11]	ENC_INTE	Encode Complete Interrupt Enable 0 = Disable 1 = Enable

[10]	DEC_INTE	5.6.7.1 Decode Complete Interrupt Enable 0 = Disable 1 = Enable
[9]	DER_INTE	Decode Error Interrupt Enable 0 = Disable 1 = Enable
[8]	EER_INTE	Encode (On-The-Fly) Error Interrupt Enable 0 = Disable 1 = Enable
[7]	Reserved	Reserved
[6]	DHE_INTS	JPEG Header Decode End Wait Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear and JPEG will resume operating from a pending state.
[5]	IPW_INTS	Input Wait Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear.
[4]	OPW_INTS	Output Wait Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear.
[3]	ENC_INTS	Encode Complete Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear.
[2]	DEC_INTS	Decode Complete Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear.
[1]	DER_INTS	Decode Error Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear.
[0]	EER_INTS	Encode (On-The-Fly) Error Interrupt Status 0 = No Interrupt 1 = Interrupt Generated Note: When write value "1" to this bit, the interrupt will be clear.

JPEG TEST Control Register (JTEST)

Register	Address	R/W	Description	Default Value
JTEST	JPG_BA + 040	R/W	JPEG Test Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BIST_ST[9:8]	
23	22	21	20	19	18	17	16
BIST_ST[7:0]							
15	14	13	12	11	10	9	8
TEST_DOUT[7:0]							
7	6	5	4	3	2	1	0
TEST_ON	BIST_ON	BIST_FINI	BIST_FAIL	TEST_SEL[3:0]			

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	BIST_ST	Internal SRAM BIST Status (Read-Only) The 8 bits indicate which one of the internal SRAM macros is fail after BIST.
[15:8]	TEST_DOUT	Test Data Output (Read-Only) The JPEG internal operation status can be read from this register by selecting TEST_SEL. Note: This control register is used only for debugging.
[7]	TEST_ON	Test Enable 0 = Disable 1 = Enable
[6]	BIST_ON	Internal SRAM BIST Mode Enable 0 = Disable 1 = Enable
[5]	BIST_FINI	Internal SRAM BIST Mode Finish (Read-Only) 0 = SRAM BIST is not finish 1 = SRAM BIST is finish
[4]	BIST_FAIL	Internal SRAM BIST Mode Fail (Read-Only) 0 = SRAM BIST is OK 1 = SRAM BIST is fail

[3:0]	TEST_SEL	Test Data Selection 0000 = exif_st[3:0] 0001 = iopt_st[3:0] 0010 = ioay_st[4:0] 0011 = cycle_st[2:0] 0100 = vld_cycle_st[1:0] 0101 = vle_st[4:0] 0110 = blk_st[2:0] 0111 = vld_st[4:0] 1000 = dec_st[2:0] 1001 = jmi_st[3:0] 1010 = addr_st[3:0]

JPEG Window Decode Mode Control Register 0 (JWINDEC0)

Register	Address	R/W	Description	Default Value
JWINDEC0	JPG_BA + 044	R/W	JPEG Window Decode Mode Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							MCU_S_Y[8]
23	22	21	20	19	18	17	16
MCU_S_Y[7:0]							
15	14	13	12	11	10	9	8
Reserved							MCU_S_X[8]
7	6	5	4	3	2	1	0
MCU_S_X[7:0]							

Bits	Descriptions	
[31:25]	Reserved	Reserved
[24:16]	MCU_S_Y	MCU(Minimum Coded Unit) Start Position Y For Window Decode Mode A 9-bit value specifies the MCU start position y of the window region within the whole image to be decoded when the window decode mode (WIN_DEC) is enabled. The position y is started from 0. Note: The MCU size is fixed to 16x16.
[15:9]	Reserved	Reserved
[8:0]	MCU_S_X	MCU Start Position X For Window Decode Mode A 9-bit value specifies the MCU start position x of the window region within the whole image to be decoded when the window decode mode (WIN_DEC) is enabled. The position x is started from 0.

JPEG Window Decode Mode Control Register 1 (JWINDEC1)

Register	Address	R/W	Description	Default Value
JWINDEC1	JPG_BA + 048	R/W	JPEG Window Decode Mode Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							MCU_E_Y[8]
23	22	21	20	19	18	17	16
MCU_E_Y[7:0]							
15	14	13	12	11	10	9	8
Reserved							MCU_E_X[8]
7	6	5	4	3	2	1	0
MCU_E_X[7:0]							

Bits	Descriptions	
[31:25]	Reserved	Reserved
[24:16]	MCU_E_Y	MCU End Position Y For Window Decode Mode A 9-bit value specifies the MCU end position y of the window region within the whole image to be decoded when the window decode mode (WIN_DEC) is enabled.
[15:9]	Reserved	Reserved
[8:0]	MCU_E_X	MCU End Position X For Window Decode Mode A 9-bit value specifies the MCU end position x of the window region within the whole image to be decoded when the window decode mode (WIN_DEC) is enabled.

JPEG Window Decode Mode Control Register 2 (JWINDEC2)

Register	Address	R/W	Description	Default Value
JWINDEC2	JPG_BA + 04C	R/W	JPEG Window Decode Mode Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				WD_WIDTH[11:8]			
7	6	5	4	3	2	1	0
WD_WIDTH[7:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	WD_WIDTH	Image Width (Y-Stride) For Window Decode Mode A 12-bit value specifies the memory line space (Y-Stride) for the window image within the whole image to be decoded when the window decode mode (WIN_DEC) is enabled: byte address of word aligned.

JPEG Memory Address Mode Control Register (JMACR)

Register	Address	R/W	Description	Default Value
JMACR	JPG_BA + 050	R/W	JPEG Memory Address Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FLY_SEL					
23	22	21	20	19	18	17	16
FLY_TYPE		Reserved				BSF_SEL[9:8]	
15	14	13	12	11	10	9	8
BSF_SEL[7:0]							
7	6	5	4	3	2	1	0
FLY_ON	Reserved			IP_SF_ON	OP_SF_ON	ENC_MODE	

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:24]	FLY_SEL	Hardware Memory On-the-Fly Access Image Buffer-Size Selection for Encode The numbers of lines used as frame-buffer = (FLY_SEL+1) * 16. Note: This setting is only valid when FLY_TYPE = 2'b01. Otherwise the buffer-size is always fixed. The minimum buffer-size is 32-line for 4:2:0 format image and is 16-line for 4:2:2 format image. If down-scaling in vertical direction is applied, the buffer-size must larger than (16 x down-scaling-factor) in 4:2:2 and (32 x down-scaling-factor) in 4:2:0 image. Ex: For scaling-down 1/3 in vertical direction, 4:2:2 image, the minimum buffer-size must be 48-line (= 16 x 3). If down-scaling in vertical direction is applied, the source image height (S_HEIGHT) needs to be specified.
[23:22]	FLY_TYPE	01 = Dual buffer on-the fly 10 = Single buffer on-the-fly
[21:18]	Reserved	Reserved
[17:8]	BSF_SEL	Memory On-the-Fly Access Bitstream Buffer-Size Selection A 10-bit value specifies the memory space for JPEG bitstream. The unit of this register is 2K Bytes. Note: The buffer region is used in a dual-buffer manner. For example, if the buffer-size is 2KB (BSF_SEL = 1), host needs to fill/remove 1KB bitstream data into/from one of the half buffer region before triggering or resuming JPEG operation, and JPEG will issue an input-wait interrupt while 1KB bitstream data stored in one of the half buffer region is processed or an output-wait interrupt while 1KB encoded bitstream data is stored into one of the half buffer region in decode and encode modes individually.
[7]	FLY_ON	Hardware Memory On-the-Fly Access Mode 0 = Disable 1 = Enable, the buffer size for source image data in encode mode is defined by FLY_SEL
[6:4]	Reserved	Reserved

[3]	IP_SF_ON	Software Memory On-the-Fly Access Mode for Data Input 0 = Disable, JPEG can only be triggered after the whole image or bitstream data is stored in frame-buffer in encode or decode mode individually 1 = Enable, JPEG can encode partial image or decode partial bitstream data by re-using a small size frame-buffer; the buffer size for the image data to be encoded is fixed, and the buffer size for the bitstream to be decoded is defined by BSF_SEL
[2]	OP_SF_ON	Software Memory On-the-Fly Access Mode for Data Output 0 = Disable, JPEG will continue to write the whole encoded bitstream or decoded image data into frame-buffer 1 = Enable, JPEG can write partial encoded bitstream or decoded image data by re-using a small size frame-buffer; the buffer size for the encoded bitstream is defined by BSF_SEL, and the buffer size for the decoded image data is fixed
[1:0]	ENC_MODE	JPEG Memory Address Mode Control 00 = Still image encode mode, the encoded bit-stream is always placed into output buffer-0 01 = Still image encode mode, the output dual-buffer is controlled by Register <i>JDBCR[0]</i> 10 = Continue image encode mode, the encoded bit-stream is placed into the continuous memory address 11 = Continue image encode mode, the distance of memory address for adjacent image is fixed and is specified by JPEG Encode Bit-stream Frame Stride Register (F_STRIDE).

JPEG Primary Scaling-up Control Register (JPSCALU)

Register	Address	R/W	Description	Default Value
JPSCALU	JPG_BA + 054	R/W	JPEG Primary Scaling-Up Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	8X	Reserved			A_JUMP	Reserved	

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6]	8X (X2 and Y2 are the same register)	Primary Image Up-Scaling For Encode 0 = No up-scaling, original size 1 = In encode mode, the image is arbitrarily up-scaled 1X~8X; Note: When FLY_TYPE = 2'b1x, the up-scaling function is not supported.
[5:3]	Reserved	Reserved
[2]	A_JUMP	Reserve Buffer Size In JPEG Bit-stream For Software Application 0 = Disable 1 = Enable, only primary encode supports this function Note: When this bit is enabled, H/W will reserve the specified size (RES_SIZE) for S/W usage to fill some vendor specified information after the Start-Of-Image (SOI) marker.
[1:0]	Reserved	Reserved

JPEG Primary Scaling-down Control Register (JPSCALD)

Register	Address	R/W	Description	Default Value
JPSCALD	JPG_BA + 058	R/W	JPEG Primary Scaling-Down Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PSX_ON	PS_LPF_ON	Reserved	PSCALX_F				
7	6	5	4	3	2	1	0
Reserved		PSCALY_F					

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	PSX_ON	Primary Image Horizontal Down-Scaling For Encode/Decode 0 = Disable, no horizontal down-scale 1 = Enable Note: When FLY_TYPE = 2'b1x, the down-scaling function is not supported. In packet format decode mode, the image is arbitrarily down-scaled 1X~16X for Y422 and Y440, 1X~8X for Y444.
[14]	PS_LPF_ON	Primary Image Down-Scaling Low Pass Filter For Decode • 0 = Disable, no down-scale low pass filter • 1 = Enable
[13]	Reserved	Reserved
[12:8]	PSCALX_F	Primary Image Horizontal Down-Scaling Factor A 5-bit value specifies the horizontal down-scaling factor. The scaling factor is equal to $2^{(1+SCALX\_F)}$. For example, if SCALX_F = 1, the image will shrink 4 times in horizontal direction. Note: 1. For planar format encode mode, SCALX_F can be any value from 0 to 31. For planar format decode mode, the value of SCALX_F can only be 0, 1, 3 i.e. scaling-down 1/2, 1/4 and 1/8. 2. For planar format decode mode, the image width after down-scaling needs to be multiple of 4. 3. For packet format, SCALX_F is reserved.
[7:6]	Reserved	Reserved

[5:0]	PSCALY_F	Primary Image Vertical Down-Scaling Factor A 6-bit value specifies the vertical down-scaling factor. The scaling factor is equal to $(1+SCALY\_F)$. For example, if $SCALY\_F = 3$, the image will shrink 4 times in vertical direction. Note: For planar format encode mode, $SCALY\_F$ can be any value from 0 to 63. For planar format decode mode, the value of $SCALY\_F$ can only be 0, 1, 3, 7, i.e. scaling-down 1/1, 1/2, 1/4 and 1/8. For packet format, $SCALY\_F$ is reserved.

JPEG Thumbnail Scaling-down Control Register (JTSCALD)

Register	Address	R/W	Description	Default Value
JTSCALD	JPG_BA + 05C	R/W	JPEG Thumbnail Scaling-Down Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TSX_ON	Reserved		TSCALX_F				
7	6	5	4	3	2	1	0
Reserved	TSCALY_F						

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	TSX_ON	Thumbnail Image Horizontal Down-Scaling For Encode 0 = Disable, no horizontal down-scale 1 = Enable
[14:13]	Reserved	Reserved
[12:8]	TSCALX_F	Thumbnail Image Horizontal Down-Scaling Factor A 5-bit value specifies the horizontal down-scaling factor. The scaling factor is equal to $2^{(1+SCALX\_F)}$. EX: If SCALX_F = 1, the image will shrink 4 times in horizontal direction.
[7:6]	Reserved	Reserved
[5:0]	TSCALY_F	Thumbnail Image Vertical Down-Scaling Factor A 6-bit value specifies the vertical down-scaling factor. The scaling factor is equal to $(1+SCALY\_F)$. EX: If SCALY_F = 3, the image will shrink 4 times in vertical direction.

JPEG Dual-buffer Control Register (JDBCR)

Register	Address	R/W	Description	Default Value
JDBCR	JPG_BA + 060	R/W	JPEG Dual-Buffer Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DBF_EN	Reserved		IP_BUF	Reserved			OP_BUF

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7]	DBF_EN	Dual Buffering Control 0 = Disable dual buffering 1 = Enable dual buffering
[6:5]	Reserved	5.6.7.2 <i>Reserved</i>
[4]	IP_BUF	Input Dual Buffer Control 0 = Input data from buffer-0 1 = Input data from buffer-1 Note: If DBF_EN is disabled, this bit is unused.
[3:1]	Reserved	5.6.7.3 <i>Reserved</i>
[0]	OP_BUF	Output Dual Buffer Control 0 = Output data to buffer-0 1 = Output data to buffer-1 Note: If DBF_EN is disabled, this bit is unused.

JPEG Encode Primary Bit-stream Reserved Size Register (JRESERVE)

Register	Address	R/W	Description	Default Value
JRESERVE	JPG_BA + 070	R/W	JPEG Encode Primary Bit-stream Reserved Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
RES_SIZE[15:8]							
7	6	5	4	3	2	1	0
RES_SIZE[7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	RES_SIZE	Primary Encode Bit-stream Reserved Size A 16-bit value specifies the reserved size (<i>byte address</i>) in encoded primary JPEG bit-stream. Note: When the function of reserved size (A_JUMP) is enabled, the value of reserved size must greater than zero, be multiple of 2 but can't be multiple of 4. The actual byte counts reserved in bit-stream is equal to (RES_SIZE - 2).

JPEG Offset Between Primary/Thumbnail Start Address Register (JOFFSET)

Register	Address	R/W	Description	Default Value
JOFFSET	JPG_BA + 074	R/W	JPEG Offset Between Primary & Thumbnail Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
OFFSET_SIZE[23:16]							
15	14	13	12	11	10	9	8
OFFSET_SIZE[15:8]							
7	6	5	4	3	2	1	0
OFFSET_SIZE[7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	OFFSET_SIZE	Primary/Thumbnail Starting Address Offset Size A 24-bit value specifies the offset size (<i>byte address</i>) between the starting address of primary and thumbnail bit-stream. Note: When thumbnail encode is enabled, the value of offset size must greater than zero and be multiple of 4.

JPEG Encode Bit-stream Frame Stride Register (JFSTRIDE)

Register	Address	R/W	Description	Default Value
JFSTRIDE	JPG_BA + 078	R/W	JPEG Encode Bit-stream Frame Stride Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
F_STRIDE[23:16]							
15	14	13	12	11	10	9	8
F_STRIDE[15:8]							
7	6	5	4	3	2	1	0
F_STRIDE[7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	F_STRIDE	JPEG Encode Bit-stream Frame Stride A 24-bit value specifies the memory distance between neighbor JPEG bit-stream (<i>byte address of word aligned</i>).

JPEG Y Component or Packet Format Frame Buffer-0 Starting Address Register (JYADDR0)

Register	Address	R/W	Description	Default Value
JYADDR0	JPG_BA + 07C	R/W	JPEG Y Component or Packet Format Frame Buffer-0 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Y_IADDR0[31:24]							
23	22	21	20	19	18	17	16
Y_IADDR0[23:16]							
15	14	13	12	11	10	9	8
Y_IADDR0[15:0]							
7	6	5	4	3	2	1	0
Y_IADDR0[7:0]							

Bits	Descriptions	
[31:0]	Y_IADDR0	JPEG Y Component Frame Buffer-0 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-0 for Y component or packet format (<i>byte address of word aligned</i>).

JPEG U Component Frame Buffer-0 Starting Address Register (JUADDR0)

Register	Address	R/W	Description	Default Value
JUADDR0	JPG_BA + 080	R/W	JPEG U Component Frame Buffer-0 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
U_IADDR0[31:24]							
23	22	21	20	19	18	17	16
U_IADDR0[23:16]							
15	14	13	12	11	10	9	8
U_IADDR0[15:8]							
7	6	5	4	3	2	1	0
U_IADDR0[7:0]							

Bits	Descriptions	
[31:0]	U_IADDR0	JPEG U Component Frame Buffer-0 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-0 for U component (byte address of word aligned).

JPEG V Component Frame Buffer-0 Starting Address Register (JVADDR0)

Register	Address	R/W	Description	Default Value
JVADDR0	JPG_BA + 084	R/W	JPEG V Component Frame Buffer-0 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
V_IADDR0[31:24]							
23	22	21	20	19	18	17	16
V_IADDR0[23:16]							
15	14	13	12	11	10	9	8
V_IADDR0[15:8]							
7	6	5	4	3	2	1	0
V_IADDR0[7:0]							

Bits	Descriptions	
[31:0]	V_IADDR0	JPEG V Component Frame Buffer-0 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-0 for V component (byte address of word aligned).

JPEG Y Component or Packet Format Frame Buffer-1 Starting Address Register (JYADDR1)

Register	Address	R/W	Description	Default Value
JYADDR1	JPG_BA + 088	R/W	JPEG Y Component or Packet Format Frame Buffer-1 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Y_IADDR1[31:24]							
23	22	21	20	19	18	17	16
Y_IADDR1[23:16]							
15	14	13	12	11	10	9	8
Y_IADDR1[15:8]							
7	6	5	4	3	2	1	0
Y_IADDR1[7:0]							

Bits	Descriptions	
[31:0]	Y_IADDR1	JPEG Y Component or Packet Format Frame Buffer-1 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-1 for Y component or packet format (<i>byte address of word aligned</i>)

JPEG U Component Frame Buffer-1 Starting Address Register (JUADDR1)

Register	Address	R/W	Description	Default Value
JUADDR1	JPG_BA + 08C	R/W	JPEG U Component Frame Buffer-1 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
U_IADDR1[31:24]							
23	22	21	20	19	18	17	16
U_IADDR1[23:16]							
15	14	13	12	11	10	9	8
U_IADDR1[15:8]							
7	6	5	4	3	2	1	0
U_IADDR1[7:0]							

Bits	Descriptions	
[31:0]	U_IADDR1	JPEG U Component Frame Buffer-1 Starting Address A32-bit value specifies the starting address bits 31 to 0 of frame buffer-1 for U component (byte address of word aligned).

JPEG V Component Frame Buffer-1 Starting Address Register (JVADDR1)

Register	Address	R/W	Description	Default Value
JVADDR1	JPG_BA + 090	R/W	JPEG V Component Frame Buffer-1 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
V_IADDR1[31:24]							
23	22	21	20	19	18	17	16
V_IADDR1[23:16]							
15	14	13	12	11	10	9	8
V_IADDR1[15:8]							
7	6	5	4	3	2	1	0
V_IADDR1[7:0]							

Bits	Descriptions	
[31:0]	V_IADDR1	JPEG V Component Frame Buffer-1 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-1 for V component (byte address of word aligned).

JPEG Y Component Frame Buffer Stride Register (JYSTRIDE)

Register	Address	R/W	Description	Default Value
JYSTRIDE	JPG_BA + 094	R/W	JPEG Y Component Frame Buffer Stride Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				Y_STRIDE[11:8]			
7	6	5	4	3	2	1	0
Y_STRIDE[7:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	Y_STRIDE	JPEG Y Component Frame Buffer Stride A 12-bit value specifies the <i>byte</i> offset of memory address of vertical adjacent line for Y component (<i>byte address of word aligned</i>). For packet format, the stride is the difference between the final output width and the input image width after scaling.

JPEG U Component Frame Buffer Stride Register (JUSTRIDE)

Register	Address	R/W	Description	Default Value
JUSTRIDE	JPG_BA + 098	R/W	JPEG U Component Frame Buffer Stride Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				U_STRIDE[11:8]			
7	6	5	4	3	2	1	0
U_STRIDE[7:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	U_STRIDE	JPEG U Component Frame Buffer Stride A 12-bit value specifies the <i>byte</i> offset of memory address of vertical adjacent line for U component (<i>byte address of word aligned</i>).

JPEG V Component Frame Buffer Stride Register (JVSTRIDE)

Register	Address	R/W	Description	Default Value
JVSTRIDE	JPG_BA + 09C	R/W	JPEG V Component Frame Buffer Stride Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				V_STRIDE[11:8]			
7	6	5	4	3	2	1	0
V_STRIDE[7:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	V_STRIDE	JPEG V Component Frame Buffer Stride A 12-bit value specifies the <i>byte</i> offset of memory address of vertical adjacent line for V component (<i>byte address of word aligned</i>).

JPEG Bit-stream Frame Buffer-0 Starting Address Register (JIOADDR0)

Register	Address	R/W	Description	Default Value
JIOADDR0	JPG_BA + 0A0	R/W	JPEG Bit-stream Frame Buffer-0 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
IO_IADDR0[31:24]							
23	22	21	20	19	18	17	16
IO_IADDR0[23:16]							
15	14	13	12	11	10	9	8
IO_IADDR0[15:8]							
7	6	5	4	3	2	1	0
IO_IADDR0[7:0]							

Bits	Descriptions	
[31:0]	IO_IADDR0	JPEG Bit-stream Frame Buffer-0 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-0 for JPEG bit-stream (<i>byte address of word aligned</i>).

JPEG Bit-stream Frame Buffer-1 Starting Address Register (JIOADDR1)

Register	Address	R/W	Description	Default Value
JIOADDR1	JPG_BA + 0A4	R/W	JPEG Bit-stream Frame Buffer-1 Starting Address Register	0x0000_0000

31	30	29	28	27	26	25	24
IO_IADDR1[31:24]							
23	22	21	20	19	18	17	16
IO_IADDR1[23:16]							
15	14	13	12	11	10	9	8
IO_IADDR1[15:8]							
7	6	5	4	3	2	1	0
IO_IADDR1[7:0]							

Bits	Descriptions	
[31:0]	IO_IADDR1	JPEG Bit-stream Frame Buffer-1 Starting Address A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-1 for JPEG bit-stream (<i>byte address of word aligned</i>).

JPEG Encode Primary Image Bit-stream Size Register (JPRI_SIZE)

Register	Address	R/W	Description	Default Value
JPRI_SIZE	JPG_BA + 0A8	R	JPEG Encode Primary Image Bit-stream Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PRI_SIZE[23:16]							
15	14	13	12	11	10	9	8
PRI_SIZE[15:8]							
7	6	5	4	3	2	1	0
PRI_SIZE[7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	PRI_SIZE	JPEG Primary Image Encode Bit-stream Size A 24-bit value reports the bit-stream <i>byte</i> size of encoded primary image.

JPEG Encode Thumbnail Image Bit-stream Size Register (JTHB_SIZE)

Register	Address	R/W	Description	Default Value
JTHB_SIZE	JPG_BA + 0AC	R	JPEG Encode Thumbnail Image Bit-stream Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
THB_SIZE[15:8]							
7	6	5	4	3	2	1	0
THB_SIZE[7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	THB_SIZE	JPEG Thumbnail Image Encode Bit-stream Size A 16-bit value reports the bit-stream <i>byte</i> size of encoded thumbnail image.

JPEG Planar Format Encode Up-scale and Packet Format Decode Down-scale Ratio (JUPRAT)

Register	Address	R/W	Description	Default Value
JUPRAT	JPG_BA + 0B0	R/W	JPEG Planar Format Encode Up-Scale and Packet Format Decode Down-Scale Ratio Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		S_HEIGHT[12:8]					
23	22	21	20	19	18	17	16
S_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved		S_WIDTH[12:8]					
7	6	5	4	3	2	1	0
S_WIDTH[7:0]							

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:16]	S_HEIGHT	JPEG Image Height Planar Format Encode Up-Scale or Packet Format Decode Down-Scale Ratio A 14-bit value specifies image height planar format encode up-scale or packet format decode down-scale ratio. For planar format, the first 4 bits are integer part and the others are decimal part. For packet format, 13 bits are decimal part. The JPEG engine supports vertical arbitrarily up-scaling in planar format encode mode and arbitrarily down-scaling in packet format decode mode. This value needs to be specified only when vertical up-scaling (Y2) or down-scaling (PSX_ON) is enabled. Note : if up-scale from 128 to 256, the up-scale ratio is $(256-1)/(128-1)$ instead of 2. if down-scale from 256 to 128, the down-scale ratio is $\text{ceil}(4096/8192)$. The height is $\text{floor}[(256 - 1) * (4096/8192)] + 1$.
[15:14]	Reserved	Reserved
[13:0]	S_WIDTH	JPEG Image Width Planar Format Encode Up-Scale or Packet Format Decode Down-Scale Ratio A 14-bit value specifies source image width planar format encode up-scale or packet format decode down-scale ratio. For planar format, the first 4 bits are integer part and the others are decimal part. For packet format, 13 bits are decimal part. The JPEG engine supports horizontal arbitrarily up-scaling in planar format encode mode and arbitrarily down-scaling in packet format decode mode. This value needs to be specified only when horizontal up-scaling (X2) or down-scaling (PSX_ON) is enabled. Note : if up-scale from 128 to 256, the up-scale ratio is $(256-1)/(128-1)$ instead of 2. If down-scale from 256 to 128, the down-scale ratio is $\text{ceil}(4096/8192)$. The width is $\text{floor}[(256 - 1) * (4096/8192)] + 1$;

JPEG Bit-stream FIFO Control Register (JBSFIFO)

Register	Address	R/W	Description	Default Value
JBSFIFO	JPG_BA + 0B4	R/W	JPEG Bit-stream FIFO Control Register	0x0000_0032

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	BSFIFO_HT			Reserved	BSFIFO_LT		

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6:4]	BSFIFO_HT	Bit-stream FIFO High-Threshold Control While the fullness of bit-stream output FIFO is higher than the high-threshold in encode mode, the priority for output will become higher than input. 000 = 2 words 001 = 4 words 010 = 6 words 011 = 8 words 100 = 10 words 101 = 12 words 110 = 14 words 111 = 16 words
[3]	Reserved	Reserved
[2:0]	BSFIFO_LT	Bit-stream FIFO Low-Threshold Control The JPEG engine may start to request memory access while the fullness of bit-stream output FIFO in encode mode or emptiness of bit-stream input FIFO in decode mode is higher than the low-threshold. 000 = 1 word 001 = 2 words 010 = 4 words 011 = 6 words 100 = 8 words 101 = 10 words 110 = 12 words 111 = 14 words

JPEG Encode Source Image Height (JSRCH)

Register	Address	R/W	Description	Default Value
JSRCH	JPG_BA + 0B8	R/W	JPEG Bit-stream FIFO Control Register	0x0000_0FFF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				JSRCH[11:8]			
7	6	5	4	3	2	1	0
JSRCH[7:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	JSRCH	JPEG Encode Source Image Height A 12-bit value specifies source image height. The JPEG engine supports vertical arbitrarily up-scaling in encode mode. This value needs to be specified only when vertical up-scaling (Y2) is enabled.

JPEG Quantization-table 0 Register (JQTAB0)

Register	Address	R/W	Description	Default Value
JQTAB0_0/1/2/3 ~ JQTAB0_60/61/62/63	JPG_BA + 100~ JPG_BA + 13F	R/W	JPEG Quantization-Table 0 Register	Undefined

31	30	29	28	27	26	25	24
QTAB0_3							
23	22	21	20	19	18	17	16
QTAB0_2							
15	14	13	12	11	10	9	8
QTAB0_1							
7	6	5	4	3	2	1	0
QTAB0_0							

Bits	Descriptions	
[31:24]	QTAB0_3	JPEG Quantization-Table 0 – 3 An 8-bit value specifies one element (3, 7, 11, ..., 59, 63) of the Quantization-Table 0. Note: 1. The sequence order of QTAB is from the left to right and from the top to bottom. 2. You need to read the same address twice to get the correct data
[23:16]	QTAB0_2	JPEG Quantization-Table 0 – 2 An 8-bit value specifies one element (2, 6, 10, ..., 58, 62) of the Quantization-Table 0.
[15:8]	QTAB0_1	JPEG Quantization-Table 0 – 1 An 8-bit value specifies one element (1, 5, 9, ..., 57, 61) of the Quantization-Table 0.
[7:0]	QTAB0_0	JPEG Quantization-Table 0 – 0 An 8-bit value specifies one element (0, 4, 8, ..., 56, 60) of the Quantization-Table 0.

JPEG Quantization-table 1 Register (JQTAB1)

Register	Address	R/W	Description	Default Value
JQTAB1_0/1/2/3 ~ JQTAB1_60/61/62/63	JPG_BA + 140~ JPG_BA + 17F	R/W	JPEG Quantization-Table 1 Register	Undefined

31	30	29	28	27	26	25	24
QTAB1_3							
23	22	21	20	19	18	17	16
QTAB1_2							
15	14	13	12	11	10	9	8
QTAB1_1							
7	6	5	4	3	2	1	0
QTAB1_0							

Bits	Descriptions	
[31:24]	QTAB1_3	JPEG Quantization-Table 1 – 3 An 8-bit value specifies one element of the Quantization-Table 1. Note: When three-QTAB mode (E3QTAB) is enabled, JPEG encoder uses this table for coding Cb component. Otherwise JPEG encoder uses this table for coding both Cb and Cr component. <i>The other requirements are the same as QTAB0 described above.</i>
[23:16]	QTAB1_2	5.6.7.4 JPEG Quantization-Table 1 – 2 An 8-bit value specifies one element of the Quantization-Table 1.
[15:8]	QTAB1_1	JPEG Quantization-Table 1 – 1 An 8-bit value specifies one element of the Quantization-Table 1.
[7:0]	QTAB1_0	JPEG Quantization-Table 1 – 0 An 8-bit value specifies one element of the Quantization-Table 1.

JPEG Quantization-table 2 Register (JQTAB2)

Register	Address	R/W	Description	Default Value
JQTAB2_0/1/2/3 ~ JQTAB2_60/61/62/63	JPG_BA + 180~ JPG_BA + 1BF	R/W	JPEG Quantization-Table 2 Register	Undefined

31	30	29	28	27	26	25	24
QTAB2_3							
23	22	21	20	19	18	17	16
QTAB2_2							
15	14	13	12	11	10	9	8
QTAB2_1							
7	6	5	4	3	2	1	0
QTAB2_0							

Bits	Descriptions	
[31:24]	QTAB2_3	JPEG Quantization-Table 2 – 3 An 8-bit value specifies one element of the Quantization-Table 2. Note: When three-QTAB mode (E3QTAB) is enabled, JPEG encoder uses this table for coding Cr component. Otherwise this table is unused. <i>The other requirements are the same as QTAB0 described above.</i>
[23:16]	QTAB2_2	5.6.7.5 JPEG Quantization-Table 2 – 2 An 8-bit value specifies one element of the Quantization-Table 2.
[15:8]	QTAB2_1	JPEG Quantization-Table 2 – 1 An 8-bit value specifies one element of the Quantization-Table 2.
[7:0]	QTAB2_0	5.6.7.6 JPEG Quantization-Table 2 – 0 An 8-bit value specifies one element of the Quantization-Table 2.

5.7 Display Interface Controller (VPOST)

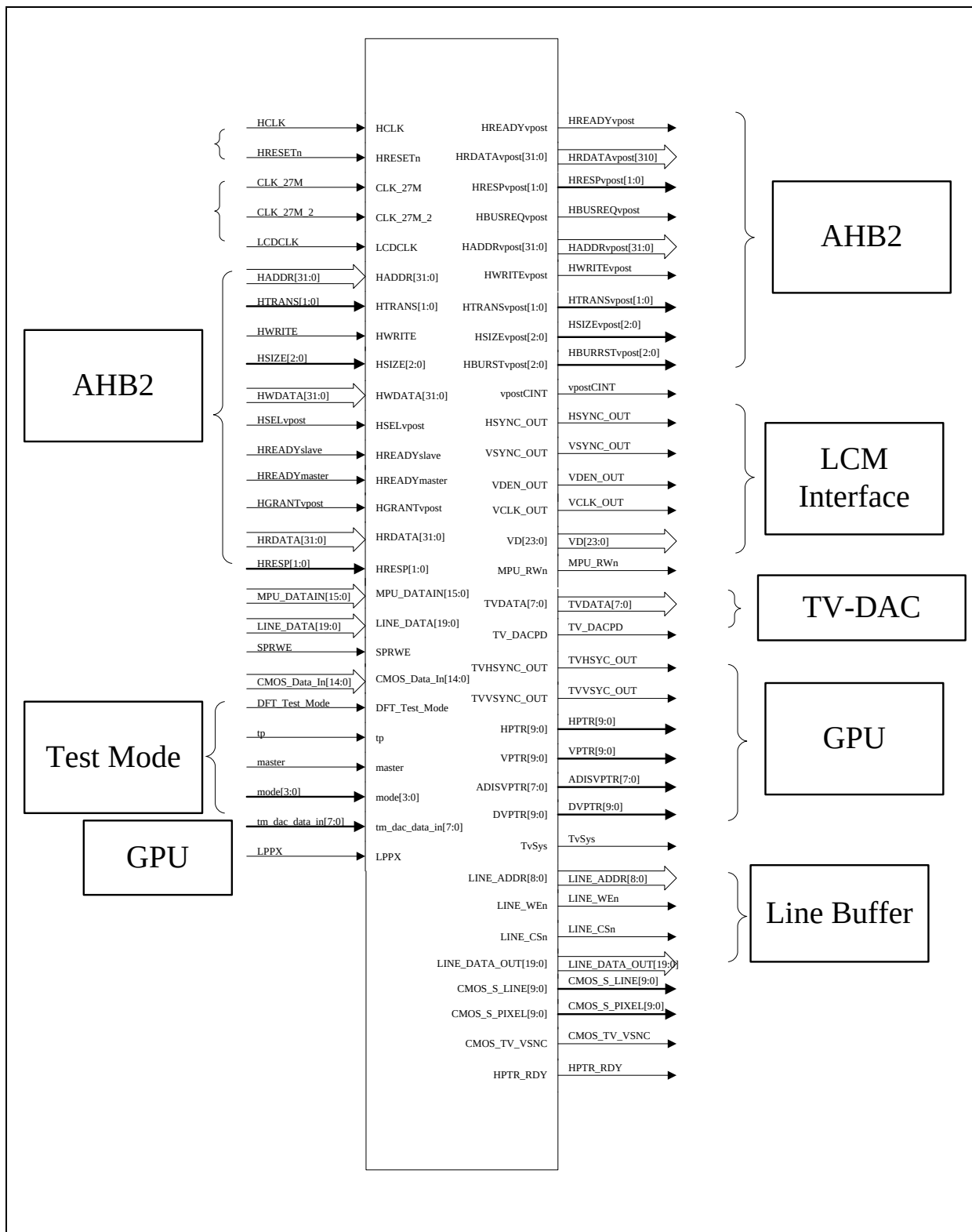
5.7.1 Overview

The main purpose of VPOST Controller (include LCD Controller & TVEncoder Controller) is used to display the video/image data to LCD device or to generate the composite signal to the TV system. The LCD timing can be synchronize with TV (NTSC/PAL non-interlace/interlace timing) or set by the LCD timing control register. The TV picture and LCD picture can display same image source simultaneously when the timing is synchronized with TV timing. The video/image data source comes from the frame buffer which stored in system memory (SDRAM).

5.7.2 Features

- Supports 2 types LCD
 - 8/16/18/24 bit Sync-Type TFT LCD
 - 8/9/16/18/24 bit MPU-Type LCD
- 8bit Sync-Type TFT LCD
 - Supports CCIR601 4 :2 :2 YCbCr packet mode (NTSC/PAL)
 - Supports CCIR601 RGB Dummy mode (NTSC/PAL)
 - Supports CCIR656 720Y/640Y Interface
 - Support Serial RGB delta/strip mode
- Support NTSC/PAL interlace & non-interlace system
- LCD Timing Setting Method
 - Resolution can up to 1024x768
 - Sync with TV(NTSC/PAL) Mode
 - Timing Control Register Setting
 - MPU type access timing can be configurable
- Support Frame Buffer solution to D1/VGA/HVGA for the TV Timing Mode
- Support Frame Buffer Format RGB555/RGB565/RGB888/YCbCr422
- Support configurable size OSD(on screen display) function
 - Format: ARGB888/RGB565/RGB555/YUV422
- OSD approach:
 - ◆ Block Overlapping
 - ◆ 2 Vertical Bar Overlapping
 - ◆ 2 Horizontal Bar Overlapping
 - ◆ 4 block
- Support rainbow effect remove filter(Notch Filter)

5.7.3 POST Controller Interface



LCD Device Pin Name	Sync-Type High Color TFT LCD (16 Bit Data Bus)	Sync-Type TFT LCD (8 Bit Data Bus)	MPU-Type LCD	
			80 Mode	68 Mode
LCD_VSYNC (GPD10)	VSYNC	VSYNC	RD	EN
LCD_HSYNC (GPD9)	HSYNC	HSYNC	WR	RW
LCD_PIXCLK (GPB15)	PCLK	PCLK	CS	CS
LCD_VDEN (GPD11)	DE	DE	RS	RS
LCD_VDATA[15:0] (GPC[15:0])	DATA[15:0]	DATA[7:0]	DATA[15:0]	DATA[15:0]

Figure 5.7-1 VPOST Controller Interface Diagram

LCD DAT A	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I/O Pin	GP B 12	GP B 11	GP B 10	GP B 9	GP B 8	GP B 7	GP E 1	GP E 0	GP C 15	GP C 14	GP C 13	GP C 12	GP C 11	GP C 10	GP C 9	GP C 8	GP C 7	GP C 6	GP C 5	GP C 4	GP C 3	GP C 2	GP C 1	GP C 0
RGB 565	1b0	1b0	1b0	1b0	1b0	1b0	1b0	1b0	R7	R6	R5	R4	R3	G7	G6	G5	G4	G3	G2	B7	B6	B5	B4	B3
RGB 666	1b0	1b0	1b0	1b0	1b0	1b0	R7	R6	R5	R4	R3	R2	G7	G6	G5	G4	G3	G2	B7	B6	B5	B4	B3	B2
RGB 888	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

Note:

1. By N9H20 pin limitation, LCD RGB888 data type cannot be support.
2. How to set the I/O pin as LCD related pin out, please define these control registers.
 - GPBPINFUN(0xB000_0084)
 - GPCPINFUN(0xB000_0088)
 - GPEPINFUN(0xB000_0090)

Figure 5.7-2 VPOST LCD Data Output Pin Definition

5.7.4 VPOST Controller Block Diagram

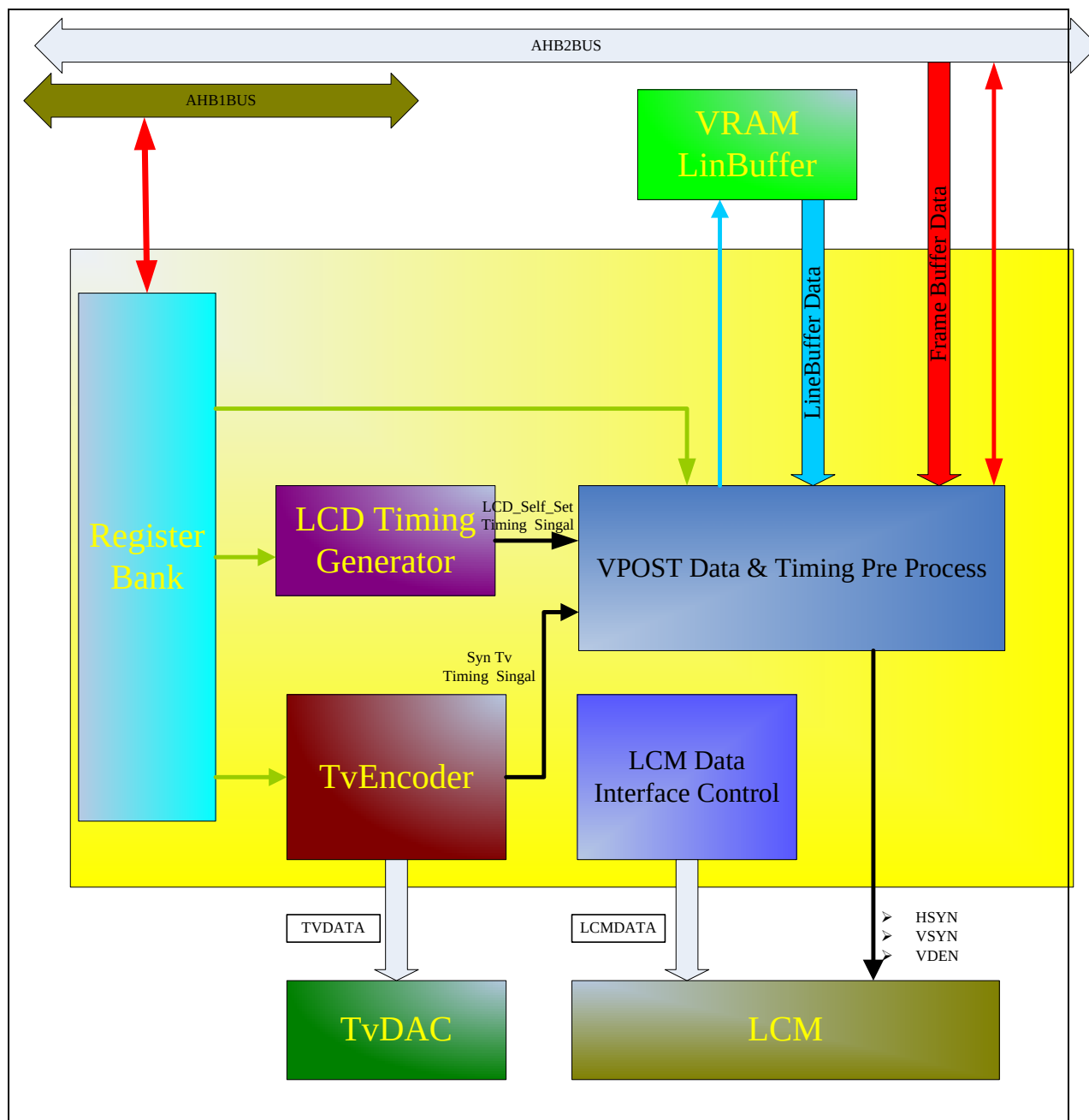


Figure 5.7-3 VPOST Controller Block Diagram

5.7.5 Functional Description

- TVEncoder
 - Generate TV NTST/PAL data signal to current DAC.
 - Generate timing control signal to LCD device.
- LCD Timing Generator
 - user-self control timing
- Register Bank
 - AHB Slave interface on AHB1.
 - A bridge that CPU control and observe the state of LCD Controller.
- VPOST Data & Timing Pre-Process
 - To generate the data request signal
 - To modify hsyn & syn signal for TFT LCM requirement
 - To generate MPU control signal
- LCM Data Interface Control
 - Transfer the CRTIC data and Frame Buffer Data to fit different types of LCD device.

5.7.6 Registers Map

R: read only, W: write only, R/W: both read and write, C: Only value 0 can be written

Register	Address	R/W	Description	Reset Value
VPOST_BA = 0xB100_2000				
LCDCCTL	0x00	R/W	LCD Controller Control Register	0x0000_0000
LCDCPrm	0x04	R/W	LCD Controller Parameter Register	0x4384_8900
LCDCInt	0x08	R/W	LCD Controller Interrupt Register	0x0000_0000
Reserved	0x0C	R/W	Reserved	0x0000_0000
TCON1	0x10	R/W	Timing Control Register 1	0x0000_0000
TCON2	0x14	R/W	Timing Control Register 2	0x0000_0000
TCON3	0x18	R/W	Timing Control Register 3	0x0000_0000
TCON4	0x1C	R/W	Timing Control Register 4	0x0140_0000
MPUCMD	0x20	R/W	MPU-type LCD Command Register	0x0000_0000
MPUTS	0x24	R/W	MPU Type timing control	0x0000_0000
OSD_CTL	0x28	R/W	OSD Control Register	0x0000_0000
OSD_SIZE	0x2C	R/W	OSD Picture SIZE	0x0000_0000
OSD_SP	0x30	R/W	OSD Start Position	0x0001_001
OSD_1BEP	0x34	R/W	OSD Bar End Position	0x0000_0000
OSD_BO	0x38	R/W	OSD Bar Offset	0x0000_0000
CBAR	0x3C	R/W	Color Burst Active Region	0x0000_0000
TVCTL	0x40	R/W	TvControl Register	0x0000_0321
IIRA	0x44	R/W	IIR notch filter Denominator Coefficient	0x0058_0c00
IIRB	0x48	R/W	IIR notch filter Numerator Coefficient	0x042C_0d0b
COLORSET	0x4C	R/W	Backdraw Color Setting Register	0x0000_0000
FSADDR	0x50	R/W	Frame Buffer Start Address Register	0x0000_0000
TvDisCtl	0x54	R/W	TV Display Control Register	0x10F0_12A5
CBACtl	0x58	R/W	Color Burst Amplitude Control Register	0x0100_100b
OSD_ADDR	0x5C	R/W	OSD Frame Buffer Start Address	0xFFFF_FFFF
TV_FFFSET1	0x60	R/W	TV Flick Free Filter Setting	0x0000_0002
TvContrast	0x64	R/W	Tv Contrast adjust setting register	0x0080_8080
TvBright	0x68	R/W	Tv Bright adjust setting register	0x0000_0000
TV_FFFSET2	0x6c	R/W	TV Flick Free Filter Setting	0x1212_2121
LINE_STRIPE	0x70	R/W	Line Stripe Offset	0x0000_EF00
RGBin	0x74	R/W	RGB 888 Data Input for RGB2YCbCr equation	0x000_0000
YCbCrout	0x78	R	YCbCr Data Output for RGB2YCbCr equation	0x0010_8080

YCbCrIn	0x7c	R/W	YCbCr Data Input for YCbCr2RGB equation	0x0010_8080
RGBout	0x80	R	RGB Data Output for YCbCr2RGB equation	0x0010_1010
LBTestCtl	0x88	R	Line Buffer Test Control	0x0000_0000

5.7.7 Registers Description

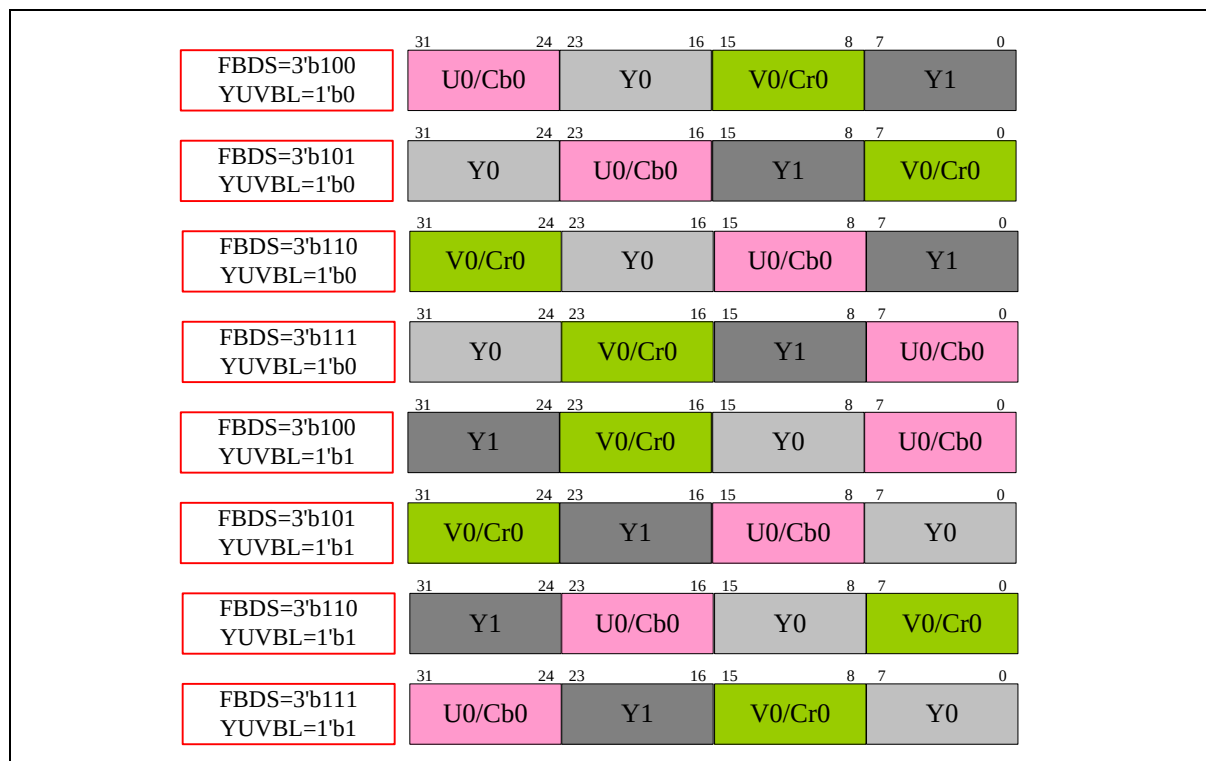
LCD Controller Control Register

Register	Address	R/W	Description	Reset Value
LCDCCtl	0x00	R/W	LCD Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
FSADDR_SEL	HAW_656	POLY_EN	Reserved				
23	22	21	20	19	18	17	16
Reserved		PRDB_SEL		Reserved			YUVBL
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				FBDS			LCDRUN

Bits	Descriptions	
[31]	FSADDR_SEL	Frame Buffer Address Selection 1'b0: Refer Internal FSADDR register 1'b1: Refer the Frame Switch Control Buffer Address
[30]	HAW_656	Horizontal Active Width for the CCIR656 1'b0: 720 Y 360CbCr mode 1'b1: 640 Y 320CbCr mode
[29]	POLY_EN	Polyphase filter Enable 1'b0: disable 1'b1: enable
[28:22]	Reserved	Reserved
[21:20]	PRDB_SEL	Parallel RGB Data Bus Selection For the High Resolution Mode 2'b00: 16 Pin (RGB565 Output) 2'b01: 18 Pin (RGB666 Output) 2'b10: 24 Pin (RGB888 Output) 2'b11: Reserved
[19:17]	Reserved	Reserved
[16]	YUVBL (1'b0)	The Y Cb(U) Cr(V) Big Endian & Little Endian selection 0: Big Endian 1: Little Endian Note: The bit was combined with FBDS for YCbCr packet data format selection
[15:4]	Reserved	Reserved
[3:1]	FBDS	Frame Buffer Data Selection

	(3'b000)	000 = RGB555 001 = RGB565 010 = RGB888_Mode0 [Dummy,R,G,B] 011 = RGB888_Mode1 [R,G,B,Dummy] 100 = Cb0 Y0 Cr0 Y1 101 = Y0 Cb0 Y1 Cr0 110 = Cr0 Y0 Cb0Y1 111 = Y0 Cr0 Y1 Cb0 Note: For the YCbCr packet data format, the packet data sequence is decided by YUVBL & FBDS register. The table list below
[0]	LCDRUN (1'b0)	LCD Controller Run 0 = Disable 1 = Enable



LCD Controller Parameter Register

Register	Address	R/W	Description	Reset Value
LCDCPrm	0x04	R/W	LCD Controller Parameter Register	0x4384_8900

31	30	29	28	27	26	25	24
Even_Field_AL				Odd_Field_AL			
23	22	21	20	19	18	17	16
F1_EL[8:1]							
15	14	13	12	11	10	9	8
F1_EL[0]	F1_SL						LCDSynTv
7	6	5	4	3	2	1	0
SRGB_EL_SEL		SRGB_OL_SEL		LCDDataSel		LCDTYPE	

Bits	Descriptions	
[31:28]	Odd_Field_AL (4'h4)	CCIR656 Odd Field Dummy Active Line Odd Filed Total Active Line=240+ Odd_Field_AL
[27:24]	Even_Field_AL (4'h3)	CCIR656 Even Field Dummy Active Line Even Filed Total Active Line=240+ Even_Field_AL
[23:15]	F1_EL (9'h109)	CCIR656 Filed1(Odd Field) Ending Line
[14:9]	F1_SL (6'h4)	CCIR656 Filed1(Odd Field) Start Line
[8]	LCDSynTv (1'b1)	LCD timing Synch with TV 0 = disable 1 = enable Note1: To set the bit for Sync Type LCD Panel which adopts UPS052 mode, it can synchronize the TV timing, So for the dual screen application, the "LCD panel screen" & "TV composite out" can show same or different image which depends on the image source setting(refer the TvCtl register setting) Note2: When this bit is clear. The Sync Type LCD panel timing is decided by TCON1, TCON2, TCON3, TCON4 register.
[7:6]	SRGB_EL_SEL (2'b00)	Serial RGB Even Line Selection 00 = Even line data is RGB 01 = Even line data is BGR 10 = Even line data is GBR 11 = Even line data is RBG
[5:4]	SRGB_OL_SEL (2'b00)	LCD data interface Select 00 = odd line data is RGB

		01 = odd line data is BGR 10 = odd line data is GBR 11 = odd line data is RBG
[3:2]	LCDDataSel (2'b00)	8bit LCD data interface Select 00 = YUV422(CCIR601) 01 = dummy serial (R, G, B Dummy) 10 = CCIR656 Output 11= Serial RGB interface
[1:0]	LCDTYPE (2'b00)	LCD device Type Select 00 = High Resolution mode(parallel output data pin 1pixel/1clock) 01 = Sync-type TFT LCD(8 bit output data pin) 10 = Reserved 11 = MPU-type LCD

LCD Controller Interrupt Register

Register	Address	R/W	Description	Reset Value
LCDCInt	0x08	R/W	LCD Controller Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Clr_ReqUnderRun	Reserved			MPUCPLEN		TVFILED_INT EN	HINTEN
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
ReqUnderRun	Reserved			MPUCPL		TVFIELD_INT	HINT

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23]	Clr_ReqUnderRun	To write 1 clear ReqUnderRun flag register
[22:21]	Reserved	Reserved
[20]	MPUCPLEN	MPU Frame Complete Enable
[19:18]	Reserved	Reserved
[17]	VINTEN (1'b0)	LCD VSYNC Interrupt Enable 0 = Disable 1 = Enable
[16]	HINTEN (1'b0)	LCD HSYNC Interrupt Enable 0 = Disable 1 = Enable
[15:8]	Reserved	Reserved
[7]	ReqUnderRun	Internal Fifo request is underun flag 1'b0: Internal fifo request is underrun 1'b1: Internal fifo request is not underrun
[6:5]	Reserved	Reserved
[4]	MPUCPL	MPU Frame Complete
[3:2]	TVFIELD_INT	T
[1]	VINT (1'b0)	LCD VSYNC End Interrupt For Sync-type LCD, if read this bit shows 1, LCDC send a frame to LCD device complete. Write 0 to clear it.
[0]	HINT	LCD HSYNC End Interrupt

	(1'b0)	For Sync-type LCD, if read this bit shows 1, LCDC send a line to LCD device complete. Write 0 to clear it.

Timing Control Register 1

Register	Address	R/W	Description	Reset Value
TCON1	0x10	R/W	Timing Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
HSPW							
15	14	13	12	11	10	9	8
HBPD							
7	6	5	4	3	2	1	0
HFPD							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	HSPW	Horizontal sync pulse width To determines the HSYNC pulse's high level width by counting the number of the LCD Pixel Clock.
[15:8]	HBPD	Horizontal back porch The number of LCD Pixel Clock periods between the falling edge of HSYNC and the start of active data.
[7:0]	HFPD	Horizontal front porch The number of LCD Pixel Clock periods between the end of active data and the rising edge of HSYNC.

(**Note:** The register is for LCD timing which do not synchronize the TV CCIR601 timing. When set the register, The **LCDCPrm[8] (LCDSynTv)** should be in clear status.)

Timing Control Register 2

Register	Address	R/W	Description	Reset Value
TCON2	0x14	R/W	Timing Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
VSPW							
15	14	13	12	11	10	9	8
VBPD							
7	6	5	4	3	2	1	0
VFPD							

Bits	Descriptions	
[31:25]	Reserved	Reserved
[23:16]	VSPW	Vertical sync pulse width To determines the VSYNC pulse's high level width by counting the number of inactive lines.
[15:8]	VBPD	Vertical back porch The number of inactive lines at the start of a frame, after vertical synchronization period.
[7:0]	VFPD	Vertical front porch The number of inactive lines at the end of a frame, before vertical synchronization period.

(**Note:** The register is for LCD timing which do not synchronize the TV CCIR601 timing. When set the register, The **LCDCPrm[8]** (**LCDSynTv**) should be in clear status.)

Timing Control Register 3

Register	Address	R/W	Description	Reset Value
TCON3	0x18	R/W	Timing Control Register 3	0x0000_0000

31	30	29	28	27	26	25	24
PPL[15:8]							
23	22	21	20	19	18	17	16
PPL[7:0]							
15	14	13	12	11	10	9	8
LPP[15:8]							
7	6	5	4	3	2	1	0
LPP[7:0]							

Bits	Descriptions	
[31:16]	PPL	Active Data Count Per-Line The PPL bit field specifies the number of output data in each line or row of screen.
[15:0]	LPP	Lines Per-Panel The LPP bit field specifies the number of active lines per screen.

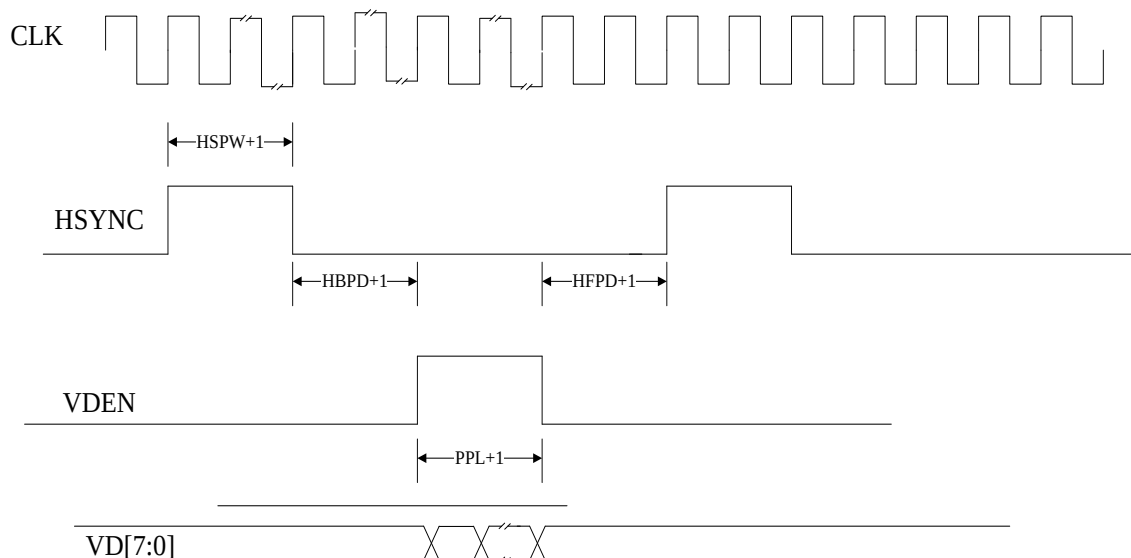
(**Note:** The register is for LCD timing which do not synchronize the TV CCIR601 timing. When set the register, The **LCDCPrm[8]** (**LCDSynTv**) should be in clear status.)

Timing Control Register 4

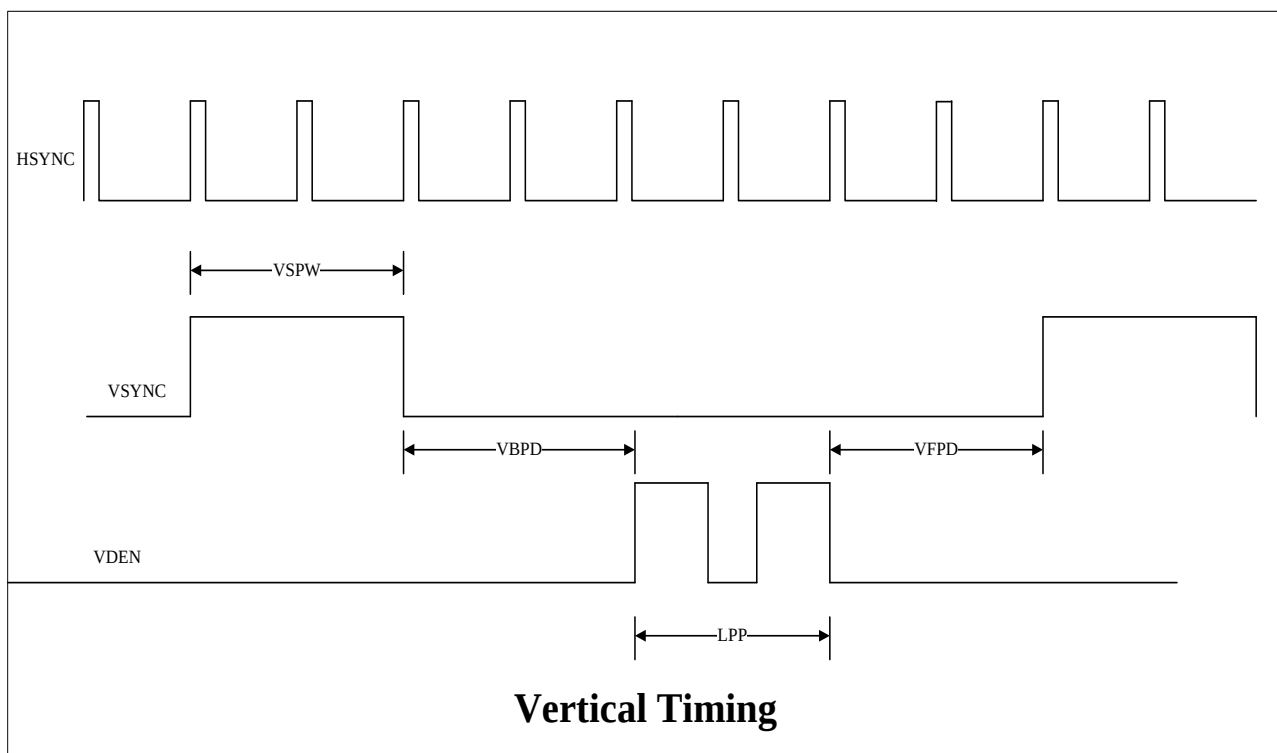
Register	Address	R/W	Description	Reset Value
TCON4	0x1c	R	Timing Control Register 4	0x0140_0000

31	30	29	28	27	26	25	24
Reserved						TAPN[9:8]	
23	22	21	20	19	18	17	16
TAPN[7:0]							
15	14	13	12	11	10	9	8
MVPW							
7	6	5	4	3	2	1	0
Reserved		MPU_FMARK P	MPU_VSYNC P	VSP	HSP	DEP	PCLKP

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	TAPN	Total Active Pixel Number
[15:8]	MVPW	MPU Vsync Pulse Width
[7:6]	Reserved	Reserved
[5]	MPU_FMARK P	MPU Type LCD FMARK Polarity 0=Wait FMARK Input rising edge(0->1) to trigger frame start 1=Wait FMARK Input falling edge(1->0) to trigger frame start
[4]	MPU_VSYNC P	MPU Type LCD Vsync Polarity 0=Sync Pulse Active Low 1=Sync Pulse Active High
[3]	VSP	Sync Type LCD VSYNC Polarity 0 = Active Low 1 = Active High
[2]	HSP	Sync Type LCD HSYNC Polarity 0 = Active Low 1 = Active High
[1]	DEP	Sync LCD VDEN Polarity 0 = Active Low 1 = Active High
[0]	PCLKP	Sync LCD Pixel Clock Polarity 0 = output video data and signals are released by rising edge of Pixel Clock 1 = output video data and signals are released by falling edge of Pixel Clock



Horizontal line timing



MPU-type LCD Command Register

Register	Address	R/W	Description	Reset Value
MPUCMD	0x20	R/W	MPU-type LCD Command Register	0x0000_0000

31	30	29	28	27	26	25	24
MPU_VFPIN_SEL	DIS_SEL	CMD_DISn	MPU_CS	MPU_ON	BUSY	WR_RS	MPU_RWn
23	22	21	20	19	18	17	16
MPU68	FMARK	Reserved		MPU_SI_SEL			
15	14	13	12	11	10	9	8
MPU_CMD[15:8]							
7	6	5	4	3	2	1	0
MPU_CMD[7:0]							

Bits	Descriptions	
[31]	MPU_VFPIN_SEL	MPU VSYNC/FMARK Pin Selection 1'b0: MPU VSYNC/FMARK as Output Pin 1'b1: MPU VSYNC/FMARK as Input Pin
[30]	DIS_SEL	1'b0: Single Mode 1'b1: Continuous Mode
[29]	CMD_DISn (1'b0)	Command or Display Mode Selection for 18/16/9/8 bit data output 1'b0 = Normal video display mode 1'b1 = Turn-on command mode for sending LCD command or parameter data;
[28]	MPU_CS (1'b0)	Command Mode 0 = Output pin CS=0 1 = Output pin CS=1
[27]	MPU_ON (1'b0)	MPU Turn On Only 0 → 1 = Trigger Enable 0 → 0, 1 → 0, 1 → 1 = Trigger Disable Note: In command mode , it will send out 16 bit command In display single mode, it will send 1 frame video data. And for the continuous display mode, it will send video data until the bit is clear
[26]	MPU_BUSY (1'b0)	Command interface is busy(only read) 0 = Command interface is ready for next command 1 = Command interface is busy for writing/reading pending command
[25]	WR_RS (1'b0)	Write/Read RS Setting 0 = Output pin RS = 0 1 = Output pin RS = 1

[24]	MPU_RWn (1'b0)	Write/Read Status or data 0 = Write command/parameter 1 = Read status/data from LCD
[23]	MPU68 (1'b0)	MPU interface Selection 0 = 80-series MPU interface 1 = 68-series MPU interface
[22]	FMARK	Frame Mark Detection Disable/Enable 1'b0: To ignore the Frame Mark Input Signal 1'b1: To update display data after Frame Mark signal
[21:20]	Reserved	Reserved
[19:16]	MPU_SI_SEL (2'h0)	MPU System Interface Selection (Refer the List Table)
[15:0]	MPU_CMD (16'h0)	MPU-type LCD command/parameter data, read data

MPU System Interface Selection						
Mode	MPU_SI_SEL	Bus Width	Pixel Color Depth	Transfer Method		Note
MPU8_Mode0	4'h0	8bit	16bit	1 st transfer	8 bit	
				2 nd Transfer	8 bit	
MPU8_Mode1	4'h1	8bit	18bit	1 st transfer	2 bit	
				2 nd Transfer	8 bit	
				3 rd Transfer	8 bit	
MPU8_Mode2	4'h2	8bit	18bit	1 st transfer	6 bit	
				2 nd Transfer	6 bit	
				3 rd Transfer	6 bit	
MPU8_Mode3	4'h3	8bit	24bit	1 st transfer	8 bit	
				2 nd Transfer	8 bit	
				3 rd Transfer	8 bit	
MPU9_Mode0	4'h4	9bit	18bit	1 st transfer	8 bit	
				2 nd Transfer	8 bit	

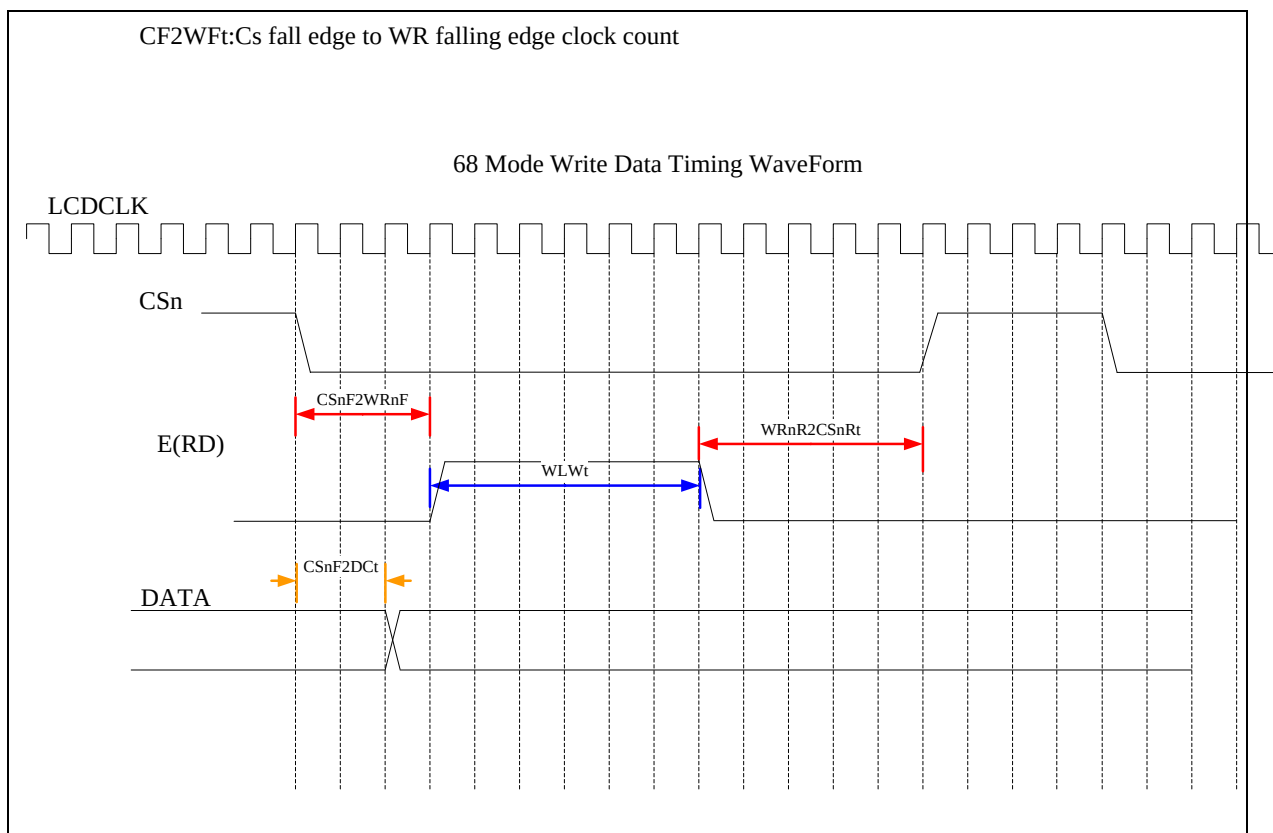
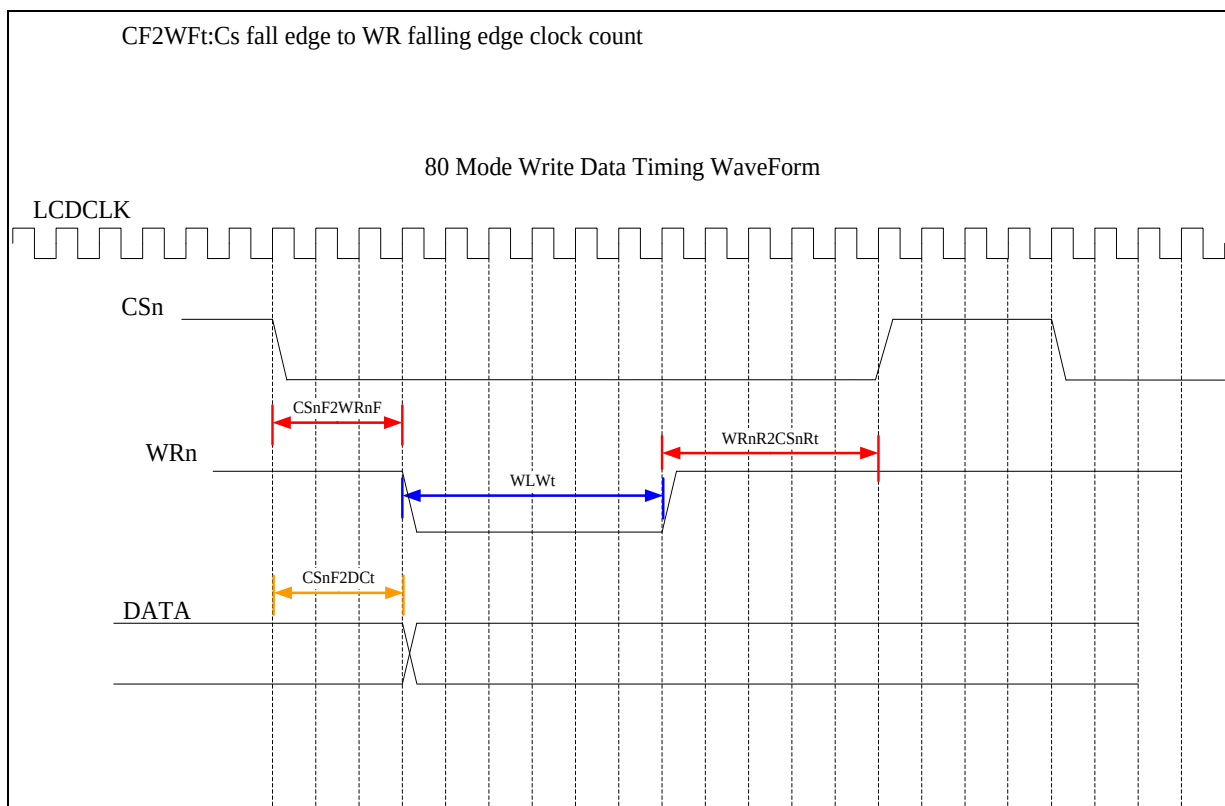
Mode	MPU_SI_SEL	Bus Width	Pixel Color Depth	Transfer Method		Note
MPU16_Mode0	4'h5	16bit	16bit	1 st transfer	16 bit	
MPU16_Mode1	4'h6	16bit	18bit	1 st transfer	16 bit	
				2 nd Transfer	2 bit	
MPU16_Mode2	4'h7	16bit	18bit	1 st transfer	2 bit	
				2 nd Transfer	16 bit	
MPU16_Mode3	4'h8	16bit	24bit	1 st transfer	16 bit	
				2 nd Transfer	8 bit	
MPU18_Mode0	4'h9	18bit	18bit	1 st transfer	18 bit	
MPU18_Mode1	4'ha	18bit	24bit	1 st transfer	18 bit	
				2 nd Transfer	6 bit	
MPU24_Mode0	4'hb	24bit	24bit	1 st transfer	24 bit	

MPU-type LCD Timing Setting Register

Register	Address	R/W	Description	Reset Value
MPUTS	0x24	R/W	MPU type LCD Timing Setting	0x0000_0000

31	30	29	28	27	26	25	24
CSnF2DCt							
23	22	21	20	19	18	17	16
WRnR2CSnRt							
15	14	13	12	11	10	9	8
WRnLWt							
7	6	5	4	3	2	1	0
CSnF2WRnFt							

Bits	Descriptions	
[31:24]	CSnF2DCt	CSn fall edge to Data change clock counter
[23:16]	WRnR2CSnRt	WRn rising edge to CSn rising clock counter
[15:8]	WRnLWt	WR Low pulse clock counter
[7:0]	CSnF2WRnFt	CSn fall edge To WR falling edge clock counter ** CSnF2WRnFt must be larger than or equal to CSnF2DCt, otherwise, except "16+0" and "24+0" modes, all other modes can't work well. **



OSD Control Register

Register	Address	R/W	Description	Reset Value
OSD_CTL	0x28	R/W	OSD Control Register	0x0000_0000

31	30	29	28	27	26	25	24
OSD_EN	Reserved		OSD_TPEN	OSD_FSEL			
23	22	21	20	19	18	17	16
			OSD_TC[23:16]				
15	14	13	12	11	10	9	8
			OSD_TC [15:8]				
7	6	5	4	3	2	1	0
			OSD_TC[7:0]				

Bits	Descriptions	
[31]	OSD_EN	OSD Enable register 1'b0: Disable 1'b1: Enable
[30:29]	Reserved	Reserved
[28]	OSD_TPEN	OSD Transparent Disable/Enable 1'b0: Disable 1'b1: Enable
[27:24]	OSD_FSEL	OSD Format Selection 4'b1000 RGB555 4'b1001 RGB565 4'b1010 = RGB888_Mode0 [Dummy,R,G,B] 4'b1011 = RGB888_Mode1 [R,G,B,Dummy] 4'b1100 = ARGB888 4'b0000 = Cb0 Y0 Cr0 Y1 4'b0001= Y0 Cb0 Y1 Cr0 4'b0010= Cr0 Y0 Cb0 Y1 4'b0011= Y0 Cr0 Y1 Cb0 4'b0100= Y1 Cr0 Y0 Cb0 4'b0101= Cr0 Y1 Cb0 Y0 4'b0110= Y1 Cb0 Y0 Cr0 4'b0111= Cb0 Y1 Cr0 Y0
[23:0]	OSD_TC	OSD Transparent Color Setting for RGB565 & YUV422 format RGB565: R5=OSD_TC[15:11] G6=OSD_TC[10:5] B5=OSD_TC[4:0]

		YUV: Y=OSD_TC[23:16]; Cb=OSD_TC[15:8] Cr=OSD_TC[7:0] RGB888: R=OSD_TC[23:16]; G=OSD_TC[15:8] B=OSD_TC[7:0] 7

OSD Size Setting Register

Register	Address	R/W	Description	Reset Value
OSD_SIZE	0x2c	R/W	OSD SIZE	0x0000_0000

31	30	29	28	27	26	25	24
						OSD_VSIZE[9:8]	
23	22	21	20	19	18	17	16
OSD_VSIZE[7:0]							
15	14	13	12	11	10	9	8
Reserved						OSD_HSIZE[9:8]	
7	6	5	4	3	2	1	0
OSD_HSIZE[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	OSD_VSIZE	OSD Vertical Size (Line)
[15:10]	Reserved	Reserved
[9:0]	OSD_HSIZE	OSD Horizontal Size (Pixel)

OSD Start Position Register

Register	Address	R/W	Description	Reset Value
OSD_SP	0x30	R/W	OSD Start Position on the background picture	0x0001_0001

31	30	29	28	27	26	25	24
						OSD_SY[9:8]	
23	22	21	20	19	18	17	16
OSD_SY[7:0]							
15	14	13	12	11	10	9	8
Reserved						OSD_SX[9:8]	
7	6	5	4	3	2	1	0
OSD_SX[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[23:16]	OSD_SY	OSD Vertical Start Position (Line) on the background picture
[15:10]	Reserved	Reserved
[9:0]	OSD_SX	OSD Horizontal Start Position (Pixel) on the background picture

OSD 1st Bar End Position Register

Register	Address	R/W	Description	Reset Value
OSD_1BEP	0x34	R/W	OSD 1 st Bar End Position on the background picture	0x0001_0001

31	30	29	28	27	26	25	24
						OSD_1BEY[9:8]	
23	22	21	20	19	18	17	16
OSD_1BEY[7:0]							
15	14	13	12	11	10	9	8
Reserved						OSD_1BEX[9:8]	
7	6	5	4	3	2	1	0
OSD_1BEX[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[23:16]	OSD_1BEY	OSD Vertical 1 st Bar End Position (Line) on the background picture
[15:10]	Reserved	Reserved
[9:0]	OSD_1BEX	OSD Horizontal 1 st Bar End Position (Pixel) on the background picture

OSD Bar Offset Setting Register

Register	Address	R/W	Description	Reset Value
OSD_BO	0x38	R/W	OSD Bar Offset	0x0000_0000

31	30	29	28	27	26	25	24
						OSD_BOY[9:8]	
23	22	21	20	19	18	17	16
OSD_BOY[7:0]							
15	14	13	12	11	10	9	8
Reserved						OSD_BOX[9:8]	
7	6	5	4	3	2	1	0
OSD_BOX[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	OSD_BOY	OSD Vertical Bar Offset (Line)
[15:10]	Reserved	Reserved
[9:0]	OSD_BOX	OSD Horizontal Bar Offset (Pixel)

Color Burst Active Region Control Register

Register	Address	R/W	Description	Reset Value
CBAR_CTL	0x3c	R/W	Color Burst Active Region Control	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved	EQ6SEL	Reserved		HCBEP[9:8]	
23	22	21	20	19	18	17	16
HCBEP[7:0]							
15	14	13	12	11	10	9	8
Reserved						HCBBP[9:8]	
7	6	5	4	3	2	1	0
HCBBP[7:0]							

Bits	Descriptions	
[31:30]	VLCBAR	Vertical Line Color Burst Active Region 2'b00:7~309 2'b01:7~310 2'b10:6~309 2'b11:6~310 (Ref Value 2'b00)
[29]	Reserved	Reserved
[28]	EQ6Sel	Equalization Pulse Selection for non-interlace PAL mode 1'b0:4 EQ Pulse 1'b1:6 EQ Pulse (Ref Value 1'b0)
[27:26]	Reserved	Reserved
[25:16]	HCBEP	Horizontal Color Burst Begin End Counter for non-interlace PAL mode (Ref Value 10'h7F)
[15:10]	Reserved	Reserved
[9:0]	HCBBP	Horizontal Color Burst Begin Pixel Counter for non-interlace PAL mode (Ref Value 10'h43)

TvControl Register

Register	Address	R/W	Description	Reset Value
TVCtl	0x40	R/W	TvControl Register	0x0000_0321

31	30	29	28	27	26	25	24
TvField	Reserved			GPU_SIZE	TvFFFE	TvFFFS	
23	22	21	20	19	18	17	16
Reserved		TvR2			TvR1		TvMSel
15	14	13	12	11	10	9	8
FBSIZE		FBMLM		LCDSrc		TvSrc	
7	6	5	4	3	2	1	0
TvOld	TvLBSA	NotchE	TvDac	TvInter	TvSys	TvColor	TvSleep

Bits	Descriptions	
[31]	TvField (1'b0)	Tv Field Status (only read) 1'b0 = Even Field 1'b1 = Odd Field
[30:28]	Reserved	Reserved
[27]	GPU_SIZE	Dummy Register
[26]	TvFFFE	Tv Flick Free Filter Enable REGISTER 1'b0=disable 1'b1=enable
[25:24]	TvFFFS	Dymmy Register
[23:22]	Reserved	Reserved
[21:19]	TvR2	27MHz Modulation delay adjust(Internal using)
[18:17]	TvR1	27MHz Line Ending delay adjust(Internal using)
[16]	TvMSel (1'h0)	TV Color Modulation Method 0 = 13.5MHz 1 = 27MHz
[15:14]	FBSIZE (2'b00)	Frame Buffer Size in TV Timing Mode 2'b00 = 320x240(QVGA) 2'b01= 640x240(HVGA) 2'b10 = 640x480(VGA) 2'b11= 720x480(D1)
[13:12]	FBMLM	Dummy Register
[11:10]	LCDSrc	LCD Source Selection

	(2'h0)	00 = Reserved 01 = Frame Buffer 10 = Register Setting Color 11 = Internal Color Bar
[9:8]	TvSrc (2'h0)	Tv Source Selection 00 = Reserved 01 = Frame Buffer 10 = Register Setting Color 11 = Internal Color Bar
[7]	TvOld	
[6]	TvLBSA (1'h0)	Tv Line Buffer Scaling Algorithm(320->640) 0 = Duplicate 1 = Interpolation
[5]	NotchE (1'h1)	Notch Filter Enable/Disable 0 = Disable 1 = Enable
[4]	TvDac (1'b0)	Analog Tv DAC Enable/Disable 0 = Enable (Normal Run) 1 = Disable (Entering Power Down Mode) Note: The bit just controls the Analog TV DAC,
[3]	TvInter (1'b0)	Interlace or Non Interlace 0 = Non-interlance 1 = Interlace
[2]	TvSys (1'b0)	TV System Selection 0 = NTSC 1 = PAL
[1]	TvColor (1'b0)	TV Color Selection Color/Black 0 = Color 1 = Black
[0]	TvSleep (1'b1)	Digital TV Encoder Enable/Disable 1 = Digital TV Timing Enable 0 = Digital TV Timing Disable Note: The bit just controls the Digital TV Timing

IIR Denominator Coefficient Register

Register	Address	R/W	Description	Reset Value
IIRA	0x44	R/W	IIR Denominator Coefficient Register	0x0058_0C00

31	30	29	28	27	26	25	24
Reserved					IIRA3[8:6]		
23	22	21	20	19	18	17	16
IIRA3[5:0]						IIRA2[8:7]	
15	14	13	12	11	10	9	8
	IIRA2[6:0]						IIRA1[8]
7	6	5	4	3	2	1	0
IIRA1[7:0]							

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:18]	IIRA3	IIR Denominator A3 Coefficient
[17:9]	IIRA2	IIR Denominator A2 Coefficient
[8:0]	IIRA1	IIR Denominator A1 Coefficient

IIR Numerator Coefficient Register

Register	Address	R/W	Description	Reset Value
IIRB	0x48	R/W	IIR Numerator Coefficient Register	0x042C_0D0B

31	30	29	28	27	26	25	24
Reserved					IIRB3[8:6]		
23	22	21	20	19	18	17	16
IIRB3[5:0]						IIRB2[8:7]	
15	14	13	12	11	10	9	8
	IIRB2[6:0]						IIRB1[8]
7	6	5	4	3	2	1	0
IIRB1[7:0]							

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:18]	IIRB3	IIR Denominator B3 Coefficient
[17:9]	IIRB2	IIR Denominator B2 Coefficient
[8:0]	IIRB1	IIR Denominator B1 Coefficient

Color Setting Register

Register	Address	R/W	Description	Reset Value
COLORSET	0x4C	R/W	RGB888 Single Color Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						VD_SWAP_MODE	VD_SWAP_EN
23	22	21	20	19	18	17	16
Color_R							
15	14	13	12	11	10	9	8
Color_G							
7	6	5	4	3	2	1	0
Color_B							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25]	VD_SWAP_MODE	0 = LCM SWAP_16BIT 1 = LCM SWAP_18BIT
[24]	VD_SWAP_EN	0 = LVDATA no swap 1 = LVDATA swap
[23:16]	Color_R	Color R Value
[15:8]	Color_G	Color G Value
[7:0]	Color_B	Color B Value

VD_SWAP_MODE = 1 (LCM SWAP_18BIT)	User function with 18-bits LCM	VD_SWAP_MODE = 0 (LCM SWAP_16BIT)	User function with 16-bits LCM

B0	LVDATA[17]/R5		TBD
B1	LVDATA[16]/R4		TBD
B2	LVDATA[15]/R3	B0	LVDATA[15]/R4
B3	LVDATA[14]/R2	B1	LVDATA[14]/R3
B4	LVDATA[13]/R1	B2	LVDATA[13]/R2
B5	LVDATA[12]/R0	B3	LVDATA[12]/R1
G0	LVDATA[11]/G5	B4	LVDATA[11]/R0
G1	LVDATA[10]/G4	G0	LVDATA[10]/G5
G2	LVDATA[9]/G3	G1	LVDATA[9]/G4
G3	LVDATA[8]/G2	G2	LVDATA[8]/G3
G4	LVDATA[7]/G1	G3	LVDATA[7]/G2
G5	LVDATA[6]/G0	G4	LVDATA[6]/G1
R0	LVDATA[5]/ B5	G5	LVDATA[5]/ G0
R1	LVDATA[4]/B4	R0	LVDATA[4]/B4
R2	LVDATA[3]/B3	R1	LVDATA[3]/B3
R3	LVDATA[2]/B2	R2	LVDATA[2]/B2
R4	LVDATA[1]/B1	R3	LVDATA[1]/B1
R5	LVDATA[0]/B0	R4	LVDATA[0]/B0

Frame Buffer Start Address Register

Register	Address	R/W	Description	Reset Value
FSADDR	0x50	R/W	Frame Buffer Start Address	0x0000_0000

31	30	29	28	27	26	25	24
FSADDR[31:24]							
23	22	21	20	19	18	17	16
FSADDR[23:16]							
15	14	13	12	11	10	9	8
FSADDR[15:8]							
7	6	5	4	3	2	1	0
FSADDR[7:0]							

Bits	Descriptions	
[31:0]	FSADDR	Frame Buffer Start Address

Color Burst Amplitude Control Register

Register	Address	R/W	Description	Reset Value
CBACtl	0x58	R/W	Color Burst Amplitude Control	0x0100_100bh

31	30	29	28	27	26	25	24
Reserved		CBA_CB4					
23	22	21	20	19	18	17	16
Reserved		CBA_CB3					
15	14	13	12	11	10	9	8
Reserved		CBA_CB2					
7	6	5	4	3	2	1	0
Reserved		CBA_CB1					

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:24]	CBA_CB4	Color Burst Amplitude Adjust factor 4 in non-interlace PAL mode (Ref Value 6'25)
[23:22]	Reserved	Reserved
[21:16]	CBA_CB3	Color Burst Amplitude Adjust factor3 in non-interlace PAL mode (Ref Value 6'1A)
[15:4]	Reserved	Reserved
[13:8]	CBA_CB2	Color Burst Amplitude Adjust factor 2 in non-interlace PAL mode (Ref Value 6'10)
[7:6]	Reserved	Reserved
[5:0]	CBA_CB1	Color Burst Amplitude Adjust factor 1 in non-interlace PAL mode (Ref Value 6'h04)

OSD Start Address Register

Register	Address	R/W	Description	Reset Value
OSD_ADDR	0x5C	R/W	OSD Frame Buffer Start Address	0x0000_0000

31	30	29	28	27	26	25	24
OSD_ADDR[31:24]							
23	22	21	20	19	18	17	16
OSD_ADDR[23:16]							
15	14	13	12	11	10	9	8
OSD_ADDR[15:8]							
7	6	5	4	3	2	1	0
OSD_ADDR[7:0]							

Bits	Descriptions	
[31:0]	OSD_ADDR	OSD Frame Buffer Start Address

Line Stripe Offset Register

Register	Address	R/W	Description	Reset Value
LINE_STRIPE	0x70	R/W	Line Stripe Offset Register	0x0000_0000

31	30	29	28	27	26	25	24
OSD_LSD[31:24]							
23	22	21	20	19	18	17	16
OSD_LSL[23:16]							
15	14	13	12	11	10	9	8
F1_LSL[15:8]							
7	6	5	4	3	2	1	0
F1_LSL[7:0]							

Bits	Descriptions	
[31:0]	OSD_LSL	OSD Buffer Line Stripe Offset Register
[15:0]	F1_LSL	Frame Buffer Line Stripe Offset Register

RGB88 Data Input

Register	Address	R/W	Description	Reset Value
RGBin	0x74	R/W	RGB 888 Data Input for RGB2YCbCr equation	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Rin							
15	14	13	12	11	10	9	8
Gin							
7	6	5	4	3	2	1	0
Bin							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Rin	Red Byte Data Input
[15:8]	Gin	Green Byte Data Input
[7:0]	Bin	Blue Byte Data Input

YCbCr Data Output

Register	Address	R/W	Description	Reset Value
YCbCrout	0x78	R	YCbCr Data Output for RGB2YCbCr equation	0x0010_8080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Yout							
15	14	13	12	11	10	9	8
Cbout							
7	6	5	4	3	2	1	0
Crout							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Yout	Red Byte Data Output
[15:8]	Cbout	Green Byte Data Output
[7:0]	Crout	Blue Byte Data Output

YCbCr Data Input

Register	Address	R/W	Description	Reset Value
YCbCrIn	0x7C	R/W	YCbCr Data Input for YCbCr2RGB equation	0x0010_8080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Yin							
15	14	13	12	11	10	9	8
Cbin							
7	6	5	4	3	2	1	0
Crin							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Yin	Red Byte Data Input
[15:8]	Cbin	Green Byte Data Input
[7:0]	Crin	Blue Byte Data Input

RGB Data Output

Register	Address	R/W	Description	Reset Value
RGBout	0x80	R	RGB Data Output for YCbCr2RGB equation	0x0010_1010

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Rout							
15	14	13	12	11	10	9	8
Gout							
7	6	5	4	3	2	1	0
Bout							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Rout	Red Byte Data Output
[15:8]	Gout	Green Byte Data Output
[7:0]	Bout	Blue Byte Data Output

Line Buffer Test Control

Register	Address	R/W	Description	Reset Value
LBTestCtl	0x88	R	Line Buffer Test Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				FinishR			
15	14	13	12	11	10	9	8
Reserved				BistFailR			
7	6	5	4	3	2	1	0
Reserved				BistModeR			

Bits	Descriptions	
[31:19]	Reserved	Reserved
[18:16]	FinishR	Bist Ram Finish Flag
[15:11]	Reserved	Reserved
[10:8]	BistFailR	Bist Ram Finish Fail Flag
[7:3]	Reserved	Reserved
[2:0]	BistModeR	Bist Ram Selection 3'b001: Line Buffer 1 3'b010: Line Buffer 2 3'b100: Line Buffer 3

5.8 Sound Processing Unit (SPU)

5.8.1 Overview

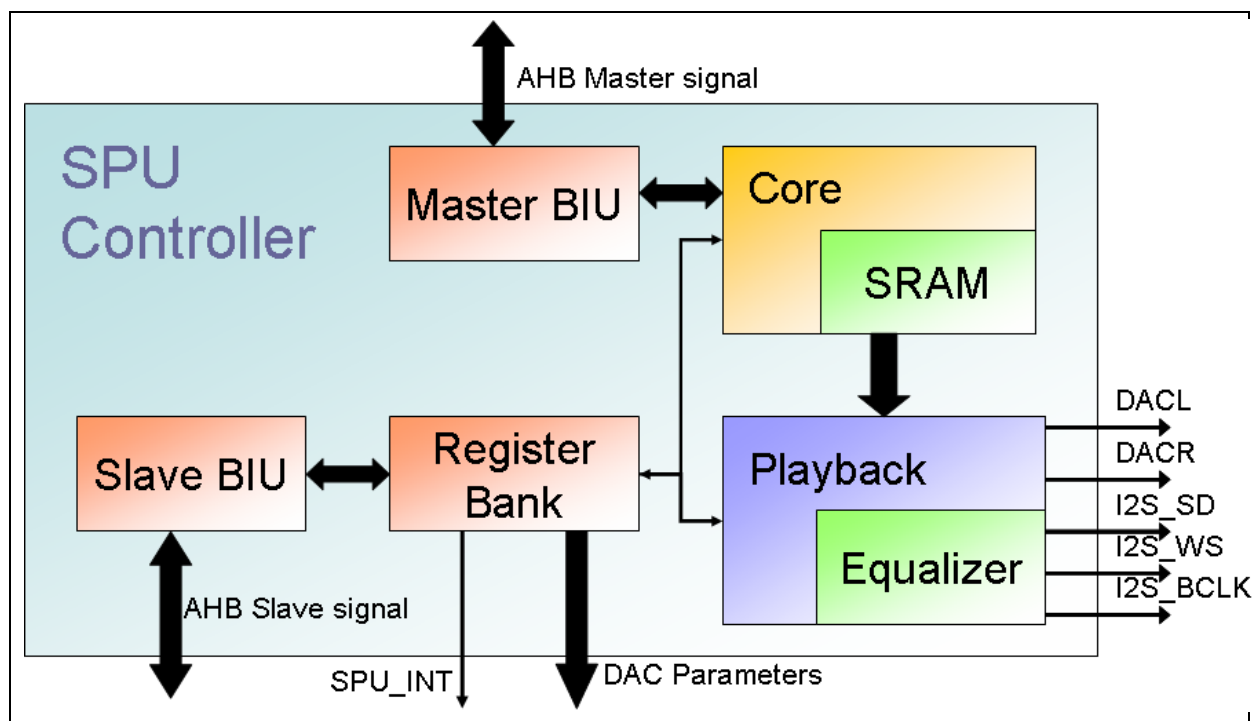
The SPU performs 32 channels audio input and 16-bit stereo output to DAC and I2S. SPU support 3 data-types (E-MDPCM (4bit), PCM16, LP8) with event and raw PCM16 mono/stereo and Tone.

5.8.2 Features

- Left / Right 16-bit stereo output to DAC with 6-bit master Volume Control
- I2S output interface
- Support 32 channels sources
- Support E-MDPCM (4bit) / PCM16 / LP8 data type with event
- Special code in source bitstream for loop playing, silence, end and user event.
- Support raw PCM16 mono/stereo type
- Dual buffer addresses
- 7-bit channel volume control
- Pan control left/right 5-bit
- DFA 13-bit for source sampling rate
- Partial update register setting (Volume , Pan , DFA)

5.8.3 SPU Block Diagram

The block diagram of SPU is shown as following:



5.8.4 Pin descriptions

- DACL : Left channel PCM output to DAC
- DACR : Right channel PCM output to DAC
- I2S_SD : I2S Data out
- I2S_WS : I2S Word Select
- I2S_BCLK : I2S Bit Clock
- SPU_INT : SPU interrupt signal

5.8.5 Register Map

R: read only, W: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W/C	Description	Reset Value
Base Address: 0xB100_0000				
SPU_CTRL	SPU_BA + 0x00	R/W	Control and Status Register	0x0000_0000
DAC_PAR	SPU_BA + 0x04	R/W	DAC Parameter Register	0x0000_1030
DAC_VOL	SPU_BA + 0x08	R/W	DAC Volume Register	0x00ff_0000
EQGain0	SPU_BA + 0x0c	R/W	Equalizer Band Gain Register 0	0x7777_7777
EQGain1	SPU_BA + 0x10	R/W	Equalizer Band Gain Register 1	0x000d_0077
CH_EN	SPU_BA + 0x14	R/W	Channel Enable	0x0000_0000
CH_IRQ	SPU_BA + 0x18	R/W	Channel Interrupt Flag	0x0000_0000
CH_PAUSE	SPU_BA + 0x1c	R/W	Channel Pause	Reserved
CH_CTRL	SPU_BA + 0x20	R/W	Channel Control Register	0x0000_0000
S_ADDR	SPU_BA + 0x24	R/W	Source Start Address	0x0000_0000
M_ADDR TONE_PULSE	SPU_BA + 0x28	R/W	Threshold Address Tone Pulse Length	0x0000_0000
E_ADDR TONE_AMP	SPU_BA + 0x2C	R/W	Source End Address Tone Amplitude	0x0000_0000
CH_PAR_1	SPU_BA + 0x30	R/W	Channel Parameter 1 Register	0x3F1F_1F00
CH_PAR_2	SPU_BA + 0x34	R/W	Channel Parameter 2 Register	0x0000_0400
CH_EVENT	SPU_BA + 0x38	R/W	Channel Event Register	0x0000_0080
Reserved	SPU_BA + 0x3c	R/W	Reserved	Reserved
CUR_ADDR	SPU_BA + 0x40	R	Channel current address Register	0x0000_0000
LP_ADDR PA_ADDR	SPU_BA + 0x44	R/W	Loop Start Address Pause Address for mono/stereo PCM16	0x0000_0000

5.8.6 Register Description

Control and Status Register

Register	Address	R/W/C	Description	Reset Value
SPU_CTRL	SPU_BA + 0x00	R/W	Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			FIFO_SIZE				
23	22	21	20	19	18	17	16
Reserved							SPU_RST
15	14	13	12	11	10	9	8
Reserved					I2S_JUSTIFIED	Reserved	I2S_EN
7	6	5	4	3	2	1	0
Reserved			SRAM_CLK_GATED	Reserved			SPU_EN

Bits	Descriptions	
[28:24]	FIFO_SIZE	Output FIFO Size Set this register to configure the output fifo size between 0~31.
[16]	SPU_RST	SPU Reset 0 = Normal 1 = Reset the whole SPU except register value.
[10]	I2S_JUSTIFIED	I2S Output Mode 0 = I2S mode 1 = Justified mode
[8]	I2S_EN	I2S Output Enable 0 = Disable 1 = Enable
[4]	SRAM_CLK_GATED	SPU SRAM Clock gated enable 0 = Disable 1 = Enable
[0]	SPU_EN	SPU Core Enable 0 = Disable 1 = Enable

DAC Parameter Register

Register	Address	R/W	Description	Reset Value
DAC_PAR	SPU_BA + 0x04	R/W	DAC Parameter Register	0x0000_1030

31	30	29	28	27	26	25	24
Reserved			DAC_ZERO_EN	Reserved		EQ_ZERO_EN	EQU_EN
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	DISCHARGE_EN	DISCHARGE_CON		Reserved			
7	6	5	4	3	2	1	0
Reserved	HP_POP_CLEAR_EN	POP_CON		Reserved			

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28]	DAC_ZERO_EN	DAC PCM Zero cross detection for DAC Volume change 0 = Disable 1 = Enable
[27:26]	Reserved	Reserved
[25]	EQ_ZERO_EN	Zero cross detection enable 0 = Disable 1 = Enable
[24]	EQU_EN	Equalizer Enable 0 : Disable 1 : Enable
[23:15]	Reserved	Reserved
[14]	DISCHARGE_EN	Enable the discharging path for output coupling capacitor. 1 : enable 0 : disable (This bit is write-only)
[13:12]	DISCHARGE_CON	Control the resistor on the discharging path. "00"=set R for 10uF C load, "01"=for 47uF, "10"=for 100uF, "11"=for 220uF. (The two bits are write-only)
[11:7]	Reserved	Reserved
[6]	HP_POP_CLEAR_EN	Headphone output pop sound clear Enable
[5:4]	POP_CON	Pop noise control register
[3:0]	Reserved	Reserved

DAC Volume Register

Register	Address	R/W	Description	Reset Value
DAC_VOL	SPU_BA + 0x08	R/W	DAC Volume Register	0x00ff_0000

31	30	29	28	27	26	25	24
DWA_SEL		Reserved	LOW_POWER_EN	Reserved			
23	22	21	20	19	18	17	16
ANA_PD[7:0]							
15	14	13	12	11	10	9	8
Reserved		LHPVL					
7	6	5	4	3	2	1	0
Reserved		RHPVL					

Bits	Descriptions	
[31:30]	DWA_SEL	Delta Sigma Mode SEL
[29]	Reserved	Reserved
[28]	LOW_POWER_EN	Low power enable for volume control OP, and it can down 0.6mA 0 : no low power model
[27:24]	Reserved	Reserved
[23:16]	ANA_PD	Audio DAC Power Down ANA_PD[0] : power down of ramp ANA_PD[1] : power down of DAC_L ANA_PD[2] : power down of DAC_R ANA_PD[3] : power down of volume control L ANA_PD[4] : power down of volume control R ANA_PD[5] : power down of headphone amp L and headphone amp R ANA_PD[6] : power down of reference voltage ANA_PD[7] : power down of current bias
[15:14]	Reserved	Reserved
[13:8]	LHPVL	Headphone Left Channel Volume Volume(dB) = (LHPVL – 0x3F) * 0.5 , if LHPVL = 0, Volume = Mute.
[7:6]	Reserved	Reserved
[5:0]	RHPVL	Headphone Right Channel Volume Volume(dB) = (RHPVL – 0x3F) * 0.5 , if RHPVL = 0, Volume = Mute.

LHPVL (RHPVL)	Volume(DB)	LHPVL (RHPVL)	Volume(DB)	LHPVL (RHPVL)	Volume(DB)	LHPVL (RHPVL)	Volume(DB)
00H	Mute	08H	-23	10H	-15	18H	-7
01H	-30	09H	-22	11H	-14	19H	-6
02H	-29	0AH	-21	12H	-13	1AH	-5
03H	-28	0BH	-20	13H	-12	1BH	-4
04H	-27	0CH	-19	14H	-11	1CH	-3
05H	-26	0DH	-18	15H	-10	1DH	-2
06H	-25	0EH	-17	16H	-9	1EH	-1
07H	-24	0FH	-16	17H	-8	1FH	-0

The volume setting of **RHPVL** is the same with that of **LHPVL**.

Equalizer Band Gain Register 0

Register	Address	R/W	Description	Reset Value
EQGain0	SPU_BA+0x0c	R/W	Equalizer Band Gain Register 0	0x7777_7777

31	30	29	28	27	26	25	24
Gain08				Gain07			
23	22	21	20	19	18	17	16
Gain06				Gain05			
15	14	13	12	11	10	9	8
Gain04				Gain03			
7	6	5	4	3	2	1	0
Gain02				Gain01			

Bits	Descriptions	
[31:28]	Gain08	Equalizer Band 08 Gain control (4000Hz @48k)
[27:24]	Gain07	Equalizer Band 07 Gain control (2000Hz @48k)
[23:20]	Gain06	Equalizer Band 06 Gain control (1000Hz @48k)
[19:16]	Gain05	Equalizer Band 05 Gain control (500Hz @48k)
[15:12]	Gain04	Equalizer Band 04 Gain control (250Hz @48k)
[11:8]	Gain03	Equalizer Band 03 Gain control (125Hz @48k)
[7:4]	Gain02	Equalizer Band 02 Gain control (62Hz @48k)
[3:0]	Gain01	Equalizer Band 01 Gain control (31Hz @48k)

Gain Setting	Amplification (DB)	Gain Setting	Amplification (DB)
00H	-7	08H	1
01H	-6	09H	2
02H	-5	0AH	3
03H	-4	0BH	4
04H	-3	0CH	5
05H	-2	0DH	6
06H	-1	0EH	7
07H	0	0FH	8

Equalizer Band Gain Register 1

Register	Address	R/W	Description	Reset Value
EQGain1	SPU_BA+0x10	R/W	Equalizer Band Gain Register 1	0x000d_0077

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				Gaindc			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Gain10				Gain09			

Bits	Descriptions	
[31:20]	Reserved	Reserved
[19:16]	Gaindc	Equalizer PassThrough Gain control
[15:8]	Reserved	Reserved
[7:4]	Gain10	Equalizer Band 10 Gain control (16000Hz @48k)
[3:0]	Gain09	Equalizer Band 09 Gain control (8000Hz @48k)

Channel Enable

Register	Address	R/W/C	Description	Reset Value
CH_EN	SPU_BA + 0x14	R/W	Channel Enable	0x0000_0000

31	30	29	28	27	26	25	24
CH_EN [31:24]							
23	22	21	20	19	18	17	16
CH_EN [23:16]							
15	14	13	12	11	10	9	8
CH_EN [15:8]							
7	6	5	4	3	2	1	0
CH_EN [7:0]							

Bits	Descriptions	
[31:0]	CH_EN	Channel Enable Register CH_EN Bit 31:0 map to Channel number 31:0 0 = Disable 1 = Enable

Channel Interrupt Flag

Register	Address	R/W/C	Description	Reset Value
CH_IRQ	SPU_BA + 0x18	R/W	Channel Interrupt Flag	0x0000_0000

31	30	29	28	27	26	25	24
CH_IRQ [31:24]							
23	22	21	20	19	18	17	16
CH_IRQ [23:16]							
15	14	13	12	11	10	9	8
CH_IRQ [15:8]							
7	6	5	4	3	2	1	0
CH_IRQ [7:0]							

Bits	Descriptions	
[31:0]	CH_IRQ	Channel Interrupt Flag CH_IRQ Bit 31:0 map to Channel number 31:0 0 = Normal 1 = Interrupt occurred, write 1 to clear.

Channel Pause

Register	Address	R/W/C	Description	Reset Value
CH_PAUSE	SPU_BA + 0x1C	R/W	Channel PAUSE	0x0000_0000

31	30	29	28	27	26	25	24
CH_PAUSE [31:24]							
23	22	21	20	19	18	17	16
CH_PAUSE [23:16]							
15	14	13	12	11	10	9	8
CH_PAUSE [15:8]							
7	6	5	4	3	2	1	0
CH_PAUSE [7:0]							

Bits	Descriptions	
[31:0]	CH_PAUSE	Channel Pause Register CH_PAUSE Bit [31:0] map to Channel number [31:0] 0 = Normal 1 = Pause NOTE: The bit [n] will set to 0 if CH_EN[n]==0

Channel Control Register

Register	Address	R/W/C	Description	Reset Value
CH_CTRL	SPU_BA + 0x20	R/W	Channel Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			CH_NO[4:0]				
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			FN_IRQ_FG	Reserved			FN_IRQ_EN
7	6	5	4	3	2	1	0
UP_IRQ	UP_DFA	UP_PAN	UP_VOL	UP_PAS_ADD R	Reserved	CH_FN[1:0]	

Bits	Descriptions	
[28:24]	CH_NO	Select Channel Number (5bits) Set target channel.
[23:13]	Reserved	Reserved
[12]	FN_IRQ_FG	Channel Function Done Interrupt Flag Only Raise to 1 when FN_IRQ_EN=1 Write 1 clear
[11:9]	Reserved	Reserved
[8]	FN_IRQ_EN	Channel Function Done Interrupt Enable 0 = Disable FN interrupt 1 = Enable FN interrupt
[7]	UP_IRQ	Interrupt Update in Partial Update Function 0 = Keep IRQ setting 1 = Update IRQ setting
[6]	UP_DFA	DFA Update in Partial Update Function 0 = Keep DFA setting 1 = Update DFA setting
[5]	UP_PAN	Pan Update in Partial Update Function 0 = Keep Pan setting 1 = Update Pan setting
[4]	UP_VOL	Volume Update in Partial Update Function 0 = Keep Volume setting 1 = Update Volume setting

[3]	UP_PAS_ADDR	Pass Address Update in Partial Update Function 0 = Keep address setting 1 = Update address setting
[2]	Reserved	Reserved
[1:0]	CH_FN	Channel Function Register 00 = no function, default 01 = load select channel register 10 = update all register setting to selected channel configuration 11 = partial update register setting to selected channel configuration NOTE: After operation complete, this register will automatically set to 00.

Source Start Address

Register	Address	R/W/C	Description	Reset Value
S_ADDR	SPU_BA + 0x24	R/W	Source Start Address	0x0000_0000

31	30	29	28	27	26	25	24
S_ADDR [31:24]							
23	22	21	20	19	18	17	16
S_ADDR [23:16]							
15	14	13	12	11	10	9	8
S_ADDR [15:8]							
7	6	5	4	3	2	1	0
S_ADDR [7:0]							

Bits	Descriptions	
[31:0]	S_ADDR	Source Start Address The channel source starting address, word boundary.

Threshold Address – Share Register with Tone Pulse Length

Register	Address	R/W/C	Description	Reset Value
M_ADDR	SPU_BA + 0x28	R/W	Threshold Address	0x0000_0000

31	30	29	28	27	26	25	24
M_ADDR [31:24]							
23	22	21	20	19	18	17	16
M_ADDR [23:16]							
15	14	13	12	11	10	9	8
M_ADDR [15:8]							
7	6	5	4	3	2	1	0
M_ADDR [7:0]							

Bits	Descriptions	
[31:0]	M_ADDR	Threshold Address When the channel current address cross this threshold address, the threshold interrupt flag will up, word boundary.

Source End Address – Share Register with Tone Amplitude

Register	Address	R/W/C	Description	Reset Value
E_ADDR	SPU_BA + 0x2C	R/W	Source End Address	0x0000_0000

31	30	29	28	27	26	25	24
E_ADDR [31:24]							
23	22	21	20	19	18	17	16
E_ADDR [23:16]							
15	14	13	12	11	10	9	8
E_ADDR [15:8]							
7	6	5	4	3	2	1	0
E_ADDR [7:0]							

Bits	Descriptions	
[31:0]	E_ADDR	Source End Address The channel source end address, word boundary. When the channel current address meet this end address, the end address interrupt flag will up, and the current address will return to start address. NOTE: This END address is word alignment and this address would not be read. Example: S_ADDR = 0x100, data buffer = 1k (0x400). E_ADDR => 0x100+0x400 = 0x500. not 0x49f or 0x49c.

Tone Pulse Length – Share Register with Threshold Address

Register	Address	R/W/C	Description	Reset Value
TONE_PULSE	SPU_BA + 0x28	R/W	Tone Pulse Length	0x0000_0000

31	30	29	28	27	26	25	24
TONE_P1 [15:8]							
23	22	21	20	19	18	17	16
TONE_P1 [7:0]							
15	14	13	12	11	10	9	8
TONE_P0 [15:8]							
7	6	5	4	3	2	1	0
TONE_P0 [7:0]							

Bits	Descriptions	
[15:0]	TONE_P1	Tone Pulse Length 1 Tone Pulse Length, Unit is output sampling period. example: output sampling rate = 22k, the period is 1/22k if TONE_P1 = 100, the P0 output will keep 100/22k seconds
[15:0]	TONE_P0	Tone Pulse Length 0 Tone Pulse Length, Unit is output sampling period. example: output sampling rate = 22k, the period is 1/22k if TONE_P0 = 100, the P0 output will keep 100/22k seconds

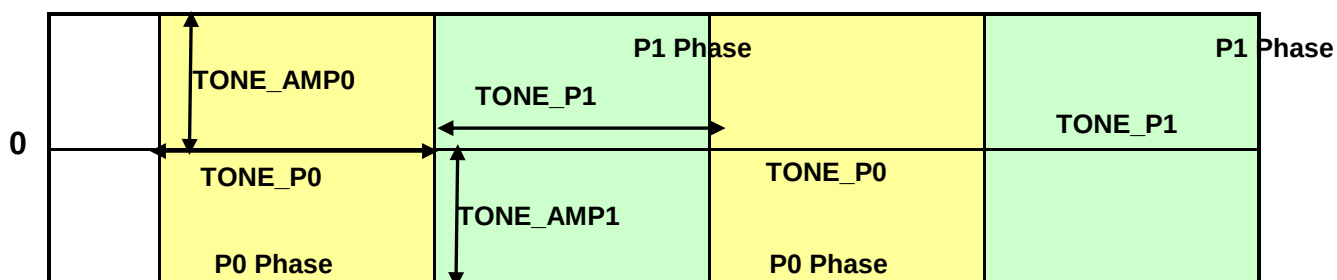
Tone Amplitude – Share Register with End Address

Register	Address	R/W/C	Description	Reset Value
TONE_AMP	SPU_BA + 0x2C	R/W	Tone Amplitude	0x0000_0000

31	30	29	28	27	26	25	24
TONE_AMP1 [15:8]							
23	22	21	20	19	18	17	16
TONE_AMP1 [7:0]							
15	14	13	12	11	10	9	8
TONE_AMP0 [15:8]							
7	6	5	4	3	2	1	0
TONE_AMP0 [7:0]							

Bits	Descriptions	
[15:0]	TONE_AMP1	Tone Amplitude 1 In the Tone Pulse Length 1 phase, output TONE_AMP1, 2's complement
[15:0]	TONE_AMP0	Tone Amplitude 0 In the Tone Pulse Length 0 phase, output TONE_AMP0, 2's complement

Tone Waveform diagram



Channel Parameter 1

Register	Address	R/W/C	Description	Reset Value
CH_PAR_1	SPU_BA + 0x30	R/W	Channel Parameter 1 Register	0x3F1F_1F00

31	30	29	28	27	26	25	24
Reserved	CH_VOL [6:0]						
23	22	21	20	19	18	17	16
Reserved			PAN_L [4:0]				
15	14	13	12	11	10	9	8
Reserved			PAN_R [4:0]				
7	6	5	4	3	2	1	0
Reserved					SRC_TYPE [2:0]		

Bits	Descriptions	
[31]	Reserved	Reserved
[30:24]	CH_VOL	Channel Volume (7bit)
[23:21]	Reserved	Reserved
[20:16]	PAN_L	Output Left Channel PAN (5bit)
[15:13]	Reserved	Reserved
[12:8]	PAN_R	Output Right Channel PAN (5bit)
[7:3]	Reserved	Reserved
[2:0]	SRC_TYPE	Channel Sound Type Sign-Magnitude, special code embedded 000 = 4-bit MDPCM, default 001 = LP8 011 = PCM16 100 = Tone Two's complement, Raw PCM Data 101 = Mono PCM16 110 = Stereo PCM16 Left ([15:0]) 111 = Stereo PCM16 Right ([31:16])

Channel Parameter 2

Register	Address	R/W/C	Description	Reset Value
CH_PAR_2	SPU_BA + 0x34	R/W	Channel Parameter 2 Register	0x0000_0400

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			DFA [12:8]				
7	6	5	4	3	2	1	0
DFA [7:0]							

Bits	Descriptions	
[31:13]	Reserved	Reserved
[12:0]	DFA	DFA (13bit) Set this register for pitch shifting and changing source sampling rate. DFA is divided into 3-bit integral part (DFA[12:10]) and 10-bit fractional part (DFA[9:0]).

Channel Event Register

Register	Address	R/W/C	Description	Reset Value
CH_EVENT	SPU_BA + 0x38	R/W	Channel Event Register	0x0000_0080

31	30	29	28	27	26	25	24
SUB_IDX [7:0]							
23	22	21	20	19	18	17	16
EVENT_IDX [7:0]							
15	14	13	12	11	10	9	8
Reserved		EV_USR_FG	EV_SLN_FG	EV_LP_FG	EV_END_FG	END_FG	TH_FG
7	6	5	4	3	2	1	0
AT_CLR_EN	Reserved	EV_USR_EN	EV_SLN_EN	EV_LP_EN	EV_END_EN	END_EN	TH_EN

Bits	Descriptions	
[31:24]	SUB_IDX	Sub-index of user event (6bit)
[23:16]	EVENT_IDX	Index of user event (8bit)
[15:14]	Reserved	Reserved
[13]	EV_USR_FG	User Event Interrupt Flag, write 1 to clear
[12]	EV_SLN_FG	Silent Event Interrupt Flag, write 1 to clear
[11]	EV_LP_FG	Loop Start Event Interrupt Flag, write 1 to clear
[10]	EV_END_FG	End Event Interrupt Flag, write 1 to clear
[9]	END_FG	End Address Interrupt Flag, write 1 to clear
[8]	TH_FG	Threshold Address Interrupt Flag, write 1 to clear
[7]	AT_CLR_EN	Auto interrupt flag clear after read event register. 0 = Disable 1 = Enable, default
[6]	Reserved	Reserved
[5]	EV_USR_EN	User Event Interrupt Enable
[4]	EV_SLN_EN	Silent Event Interrupt Enable
[3]	EV_LP_EN	Loop Start Event Interrupt Enable
[2]	EV_END_EN	End Event Interrupt Enable
[1]	END_EN	End Address Interrupt Enable
[0]	TH_EN	Threshold Address Interrupt Enable

Channel Current Address

Register	Address	R/W/C	Description	Reset Value
CUR_ADDR	SPU_BA + 0x40	R	Channel current address Register	0x0000_0000

31	30	29	28	27	26	25	24
CUR_ADDR [31:24]							
23	22	21	20	19	18	17	16
CUR_ADDR [23:16]							
15	14	13	12	11	10	9	8
CUR_ADDR [15:8]							
7	6	5	4	3	2	1	0
CUR_ADDR [7:0]							

Bits	Descriptions	
[31:0]	CUR_ADDR	Channel current address

Loop Start Address

Register	Address	R/W/C	Description	Reset Value
LP_ADDR	SPU_BA + 0x44	R/W	Loop Start Address	0x0000_0000

31	30	29	28	27	26	25	24
LP_ADDR [31:24]							
23	22	21	20	19	18	17	16
LP_ADDR [23:16]							
15	14	13	12	11	10	9	8
LP_ADDR [15:8]							
7	6	5	4	3	2	1	0
LP_ADDR [7:0]							

Bits	Descriptions	
[31:0]	LP_ADDR	Loop Start Address When loop start event occur, the address will keep in this register.

Pause Address for Mono/Stereo PCM16 – Share Register with Loop Start Address

Register	Address	R/W/C	Description	Reset Value
PA_ADDR	SPU_BA + 0x44	R/W	Pause Address for mono/stereo PCM16 format	0x0000_0000

31	30	29	28	27	26	25	24
PA_ADDR [31:24]							
23	22	21	20	19	18	17	16
PA_ADDR [23:16]							
15	14	13	12	11	10	9	8
PA_ADDR [15:8]							
7	6	5	4	3	2	1	0
PA_ADDR [7:0]							

Bits	Descriptions	
[31:0]	PA_ADDR	Pause Address for mono/stereo PCM16 format When the channel current address meet this pause address, the current channel will auto Pause.

5.9 I2S Controller(I2S)

5.9.1 Overview

The audio controller consists of I2S protocols to interface with external audio CODEC. The I2S interface supports 16, 18, 20 and 24-bit left/right precision in record and playback. When operating in 18/20/24-bit precision, each left/right-channel sample is stored in a 32-bit word. Each left/right-channel sample has 24/20/18 MSB bits of valid data and other LSB bits are the padding zeros. When operating in 16-bit precision, right-channel sample is stored in MSB of a 32-bit word and left-channel sample is stored in LSB of a 32-bit word.

5.9.2 Features

- When 16-bit precision, the DMA always 8-beat incrementing burst (FIFO_TH = 0) or 4-beat incrementing burst (FIFO_TH = 1).
- When 24/20/18-bit precision, the DMA always 16-beat incrementing burst (FIFO_TH = 0) or 8-beat incrementing burst (FIFO_TH = 1).
- Always bus lock when 4-beat or 8-beat or 16-beat incrementing burst.
- When reach eighth, quarter, middle and end address of destination address, a DMA_IRQ is triggered to CPU automatically.

An AHB master port and an AHB slave port are offered in audio controller.

5.9.3 Block diagram

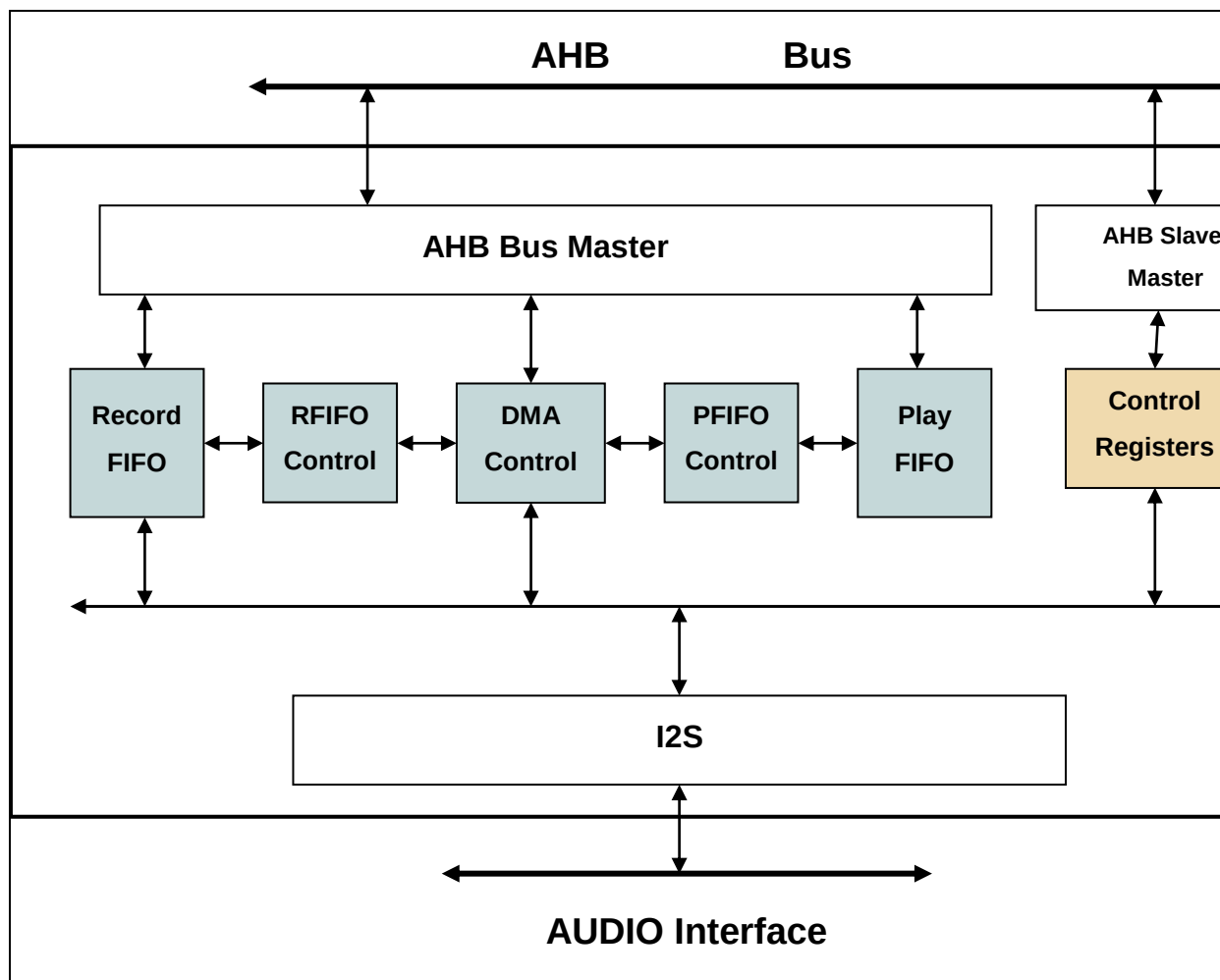


Figure 5.9.1 Block diagram of Audio Controller

5.9.4 I2S Interface

The I2S interface signals are shown as the following figure

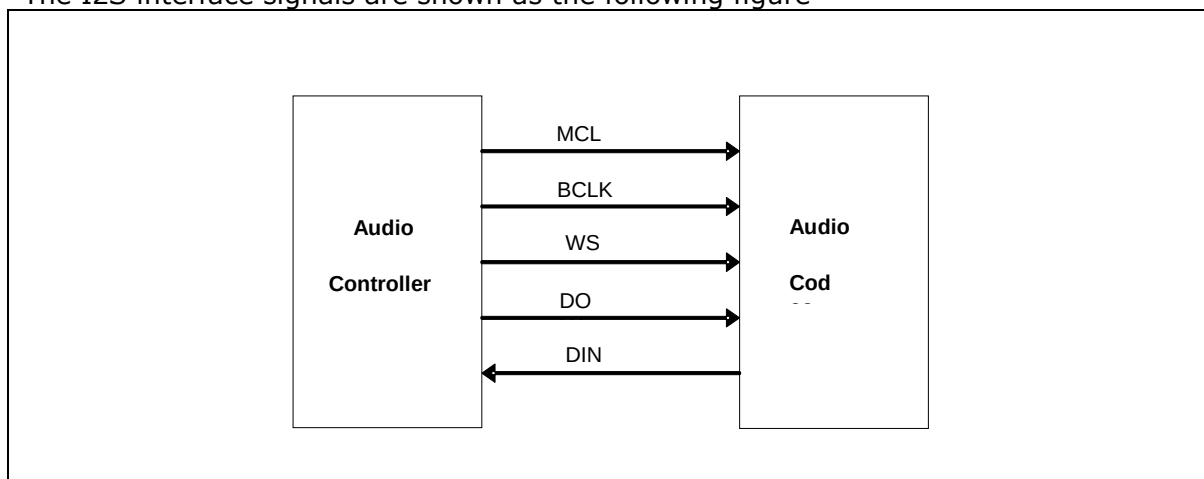


Figure 5.9.2 The interface signal of I2S

The I2S and MSB-justified format are supported; the timing diagram is shown as following figure.

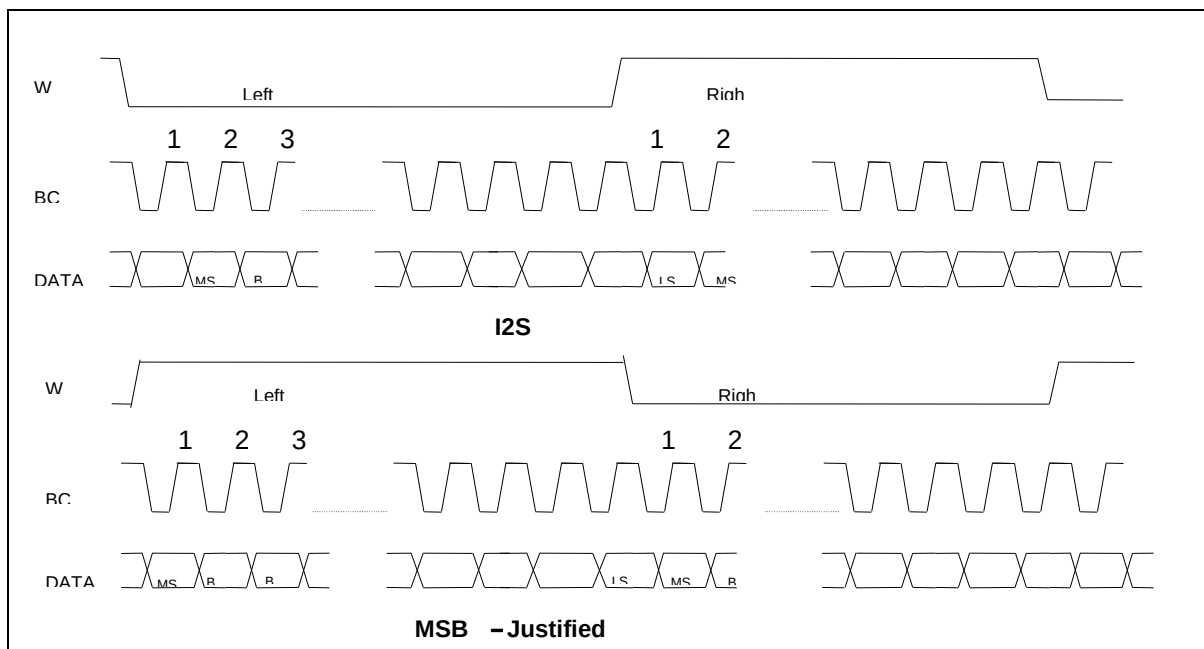


Figure 5.9.3 The format of I2S

The sampling rate, bit shift clock frequency could be set by the control register ACTL_I2SCON.

5.9.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write, **C:** Only value 1 can be written

Register	Address	R/W	Description	Reset Value
I2S_BA = 0xB100_1000				
ACTL_CON	I2S_BA + 0x00	R/W	Audio control register	0x0000_0000
ACTL_RESET	I2S_BA + 0x04	R/W	Sub block reset control register	0x0000_0000
ACTL_RDSTB	I2S_BA + 0x08	R/W	DMA destination base address register for record	0x0000_0000
ACTL_RDST_LENGTH	I2S_BA + 0x0C	R/W	DMA destination length register for record	0x0000_0000
ACTL_RDSTC	I2S_BA + 0x10	R	DMA destination current address register for record	0x0000_0000
ACTL_PDSTB	I2S_BA + 0x14	R/W	DMA destination base address register for play	0x0000_0000
ACTL_PDST_LENGTH	I2S_BA + 0x18	R/W	DMA destination length register for play	0x0000_0000
ACTL_PDSTC	I2S_BA + 0x1C	R	DMA destination current address register for play	0x0000_0000
ACTL_RSR	I2S_BA + 0x20	R/W	Record status register	0x0000_0000
ACTL_PSR	I2S_BA + 0x24	R/W	Play status register	0x0000_0000
ACTL_I2SCON	I2S_BA + 0x28	R/W	I2S control register	0x0000_0000
ACTL_COUNTER	I2S_BA + 0x2C	R/W	DMA counter down values	0xFFFF_FFFF

5.9.6 Register Description

Audio Controller Control Register (ACTL_CON)

Register	Address	R/W	Description	Reset Value
ACTL_CON	I2S_BA + 0x00	R/W	Audio controller control register	0x0000_0000

The ACTL_CON register control the basic operation of audio controller.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		R_DMA_IRQ_EN	P_DMA_IRQ_EN	R_FIFO_FULL_IRQ_EN	R_FIFO_EMPTY_IRQ_EN	P_FIFO_FULL_IRQ_EN	P_FIFO_EMPTY_IRQ_EN
15	14	13	12	11	10	9	8
R_DMA_IRQ_SEL		P_DMA_IRQ_SEL		R_DMA_IRQ	P_DMA_IRQ	I2S_BITS_16_24	Reserved
7	6	5	4	3	2	1	0
FIFO_TH	Reserved		IRQ_DMA_CNTER_EN	IRQ_DMA_DATA_ZERO_EN	Reserved	I2S_EN	Reserved

Bits	Descriptions	
[31:20]	RESERVED	RESERVED
[21]	R_DMA_IRQ_EN	Record DMA Interrupt Request Enable bit 0: not allowed to generation R_DMA_IRQ 1: allowed to generation R_DMA_IRQ The R_DMA_IRQ_EN bit is read/write
[20]	P_DMA_IRQ_EN	Playback DMA Interrupt Request Enable bit 0: not allowed to generation P_DMA_IRQ 1: allowed to generation P_DMA_IRQ The P_DMA_IRQ_EN bit is read/write
[19]	R_FIFO_FULL_IRQ_EN	Record FIFO Full Interrupt Request Enable bit 0: not allowed to generation R_FIFO_FULL_IRQ 1: allowed to generation R_FIFO_FULL_IRQ The R_FIFO_FULL_IRQ_EN bit is read/write
[18]	R_FIFO_EMPTY_IRQ_EN	Record FIFO Empty Interrupt Request Enable bit 0: not allowed to generation R_FIFO_EMPTY_IRQ 1: allowed to generation R_FIFO_EMPTY_IRQ The R_FIFO_EMPTY_IRQ_EN bit is read/write
[17]	P_FIFO_FULL_IRQ_EN	Playback FIFO Full Interrupt Request Enable bit 0: not allowed to generation P_FIFO_FULL_IRQ 1: allowed to generation P_FIFO_FULL_IRQ The P_FIFO_FULL_IRQ_EN bit is read/write
[16]	P_FIFO_EMPTY_IRQ_EN	Playback FIFO Empty Interrupt Request Enable bit 0: not allowed to generation P_FIFO_EMPTY_IRQ

		1: allowed to generation P_FIFO_EMPTY_IRQ The P_FIFO_EMPTY_IRQ_EN bit is read/write
[15:14]	R_DMA_IRQ_SEL[1:0]	Record DMA Interrupt Request Selection bits 00: When record DMA address reach DMA record destination end address, the R_DMA_RIA_IRQ will be issued. 01: When record DMA address reach each half of DMA record destination end address, the R_DMA_RIA_IRQ will be issued. 10: When record DMA address reach each quarter of DMA record destination end address, the R_DMA_RIA_IRQ will be issued. 11: When record DMA address reach each eighth of DMA record destination end address, the R_DMA_RIA_IRQ will be issued. The R_DMA_IRQ_SEL bits are read/write
[13:12]	P_DMA_IRQ_SEL[1:0]	Play DMA Interrupt Request Selection bits 00: When play DMA address reach DMA play destination end address, the P_DMA_RIA_IRQ will be issued. 01: When play DMA address reach each half of DMA play destination end address, the P_DMA_RIA_IRQ will be issued. 10: When play DMA address reach each quarter of DMA play destination end address, the P_DMA_RIA_IRQ will be issued. 11: When play DMA address reach each eighth of DMA play destination end address, the P_DMA_RIA_IRQ will be issued. The P_DMA_IRQ_SEL bits are read/write
[11]	R_DMA_IRQ	Record DMA Interrupt Request bit When R_DMA_RIA_IRQ or R_FIFO_FULL or R_FIFO_EMPTY is set to "1" in record and these corresponding interrupt enable bits are set to "1", the R_DMA_IRQ bit will be set to 1 automatically, and this bit could be cleared to 0 by CPU writing "1". The bit is hardwired to ARM926 as interrupt request signal with an inverter. The R_DMA_IRQ bit is read/write
[10]	P_DMA_IRQ	Playback DMA Interrupt Request bit When P_DMA_RIA_IRQ or DMA_DATA_ZERO_IRQ or DMA_CNTER_IRQ or P_FIFO_FULL or P_FIFO_EMPTY is set to 1 in playback and these corresponding interrupt enable bits are set to "1", the bit P_DMA_IRQ will be set to 1, and this bit could be clear to 0 by CPU writing "1". And the bit is hardwired to ARM926 as interrupt request signal with an inverter. The P_DMA_IRQ bit is read/write
[9]	I2S_BITS_16_24	I2S_BITS_16_24 bit 0: I2S data format is 16-bits of a channel. 1: I2S data format is 24-bits of a channel. The I2S_BITS_16_24 bit is read/write
[8]	RESERVED	RESERVED
[7]	FIFO_TH	FIFO Threshold Control bit 0: The FIFO threshold is 8 levels. 1: The FIFO threshold is 4 levels. The FIFO_TH bit is read/write
[6:5]	RESERVED	RESERVED
[4]	IRQ_DMA_CNTER_EN	IRQ_DMA counter function enable Bit 0: not allowed to generation P_DMA_IRQ 1: allowed to generation P_DMA_IRQ

		The IRQ_DMA_CNTER_EN bit is read/write
[3]	IRQ_DMA_DATA_ZERO_EN	IRQ_DMA_DATA zero and sign detect enable bit 0: not allowed to generation P_DMA_IRQ 1: allowed to generation P_DMA_IRQ The IRQ_DMA_DATA_ZERO_EN bit is read/write
[2]	RESERVED	RESERVED
[1]	I2S_EN	I2S Enable bit 0: The I2S interface is disabled. 1: The I2S interface is enabled. The I2S bits are read/write
[0]	RESERVED	RESERVED

Sub-block Reset Control Register (ACTL_RESET)

Register	Address	R/W	Description	Reset Value
ACTL_RESET	I2S_BA + 0x04	R/W	Sub block reset control	0x0000_0000

The ACTL_RESET register control the reset operation in each sub block.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							ACTL_RESET
15	14	13	12	11	10	9	8
RECORD_SINGLE[1:0]		Reserved	PLAY_STEREO	Reserved			
7	6	5	4	3	2	1	0
Reserved	I2S_RECORD	I2S_PLAY	DMA_CNTER_EN	DMA_DATA_ZERO_EN	Reserved		I2S_RESET

Bits	Descriptions	
[31:17]	RESERVED	RESERVED
[16]	ACTL_RESET	Audio Controller Reset Control bit 1: The whole audio controller is reset. 0: The audio controller is normal operation. The ACTL_RESET bit is read/write
[15:14]	RECORD_SINGLE [1:0]	Record Single/Dual Channel Select bits 11: The record is dual channel. 01: The record only selects left channel. 10: The record only selects right channel. 00: RESERVED. The PLAY_SINGLE[1:0] bits are read/write
[13]	RESERVED	RESERVED
[12]	PLAY_STEREO	Playback Stereo bit 1: The playback is in stereo mode. 0: The playback is in mono mode. The PLAY_Stereo bit is read/write
[11:7]	RESERVED	RESERVED
[6]	I2S_RECORD	I2S Record Control bit 0: The record path of I2S is disabled. 1: The record path of I2S is enabled. The I2S_RECORD bit is read/write
[5]	I2S_PLAY	I2S Playback Control bit 0: The playback path of I2S is disabled. 1: The playback path of I2S is enabled. The I2S_PLAY bit is read/write

[4]	DMA_CNTER_EN	DMA counter function enable Bit This function is supported to count playback data for software monitoring. When one playback data is transferred to codec, the DMA counter subtracts 1. When the ACTL_COUNTER [31:0] register is Zero that set DMA_CNTER_IRQ bit =1. 0: The DMA counter function is disabled. 1: The DMA counter function is enabled. The DMA_CNTER_EN bit is read/write
[3]	DMA_DATA_ZERO_EN	DMA_DATA zero and sign detect enable bit 0: The DMA_DATA zero and sign detect function is disabled. 1: The DMA_DATA zero and sign detect function is enabled. The DMA_DATA_ZERO_EN bit is read/write
[2:1]	RESERVED	RESERVED
[0]	I2S_RESET	I2S RESET Control bit 0: Release the I2S function block from reset mode. 1: Force the I2S function block to reset mode. The I2S_RESET bit is read/write

DMA Record Destination Base Address (ACTL_RDESB)

Register	Address	R/W	Description	Reset Value
ACTL_RDESB	I2S_BA + 0x08	R/W	DMA record destination base address	0x0000_0000

The value in ACTL_RDESB register is the record destination base address of DMA, and only could be changed by CPU.

31	30	29	28	27	26	25	24
AUDIO_RDESB[31:24]							
23	22	21	20	19	18	17	16
AUDIO_RDESB[23:16]							
15	14	13	12	11	10	9	8
AUDIO_RDESB[15:8]							
7	6	5	4	3	2	1	0
AUDIO_RDESB[7:0]							

Bits	Descriptions	
[31:0]	AUDIO_RDESB[31:0]	32-bit Record Destination Base Address The AUDIO_RDESB [31:0] bits are read/write.

DMA Record Destination Address Length (ACTL_RDES_LENGTH)

Register	Address	R/W	Description	Reset Value
ACTL_RDES_LENGTH	I2S_BA + 0x0C	R/W	DMA record destination address length	0x0000_0000

The value in ACTL_RDES_LENGTH register is the record destination address length of DMA, and the register could only be changed by CPU.

31	30	29	28	27	26	25	24
AUDIO_RDES_L[31:24]							
23	22	21	20	19	18	17	16
AUDIO_RDES_L[23:16]							
15	14	13	12	11	10	9	8
AUDIO_RDES_L[15:8]							
7	6	5	4	3	2	1	0
AUDIO_RDES_L[7:0]							

Bits	Descriptions	
[31:0]	AUDIO_RDES_L[31:0]	32-bit Record Destination Address Length The AUDIO_RDES_L [31:0] bits are read/write. The minimum value for 16-bits mode is 0x20 and for 24-bits mode is 0x40.

DMA Record Destination Current Address (ACTL_RDESC)

Register	Address	R/W	Description	Reset Value
ACTL_RDESC	I2S_BA + 0x10	R	DMA record destination current address	0x0000_0000

The value in ACTL_RDESC is the DMA record destination current address; this register could only be read by CPU.

31	30	29	28	27	26	25	24
AUDIO_RDESC[31:24]							
23	22	21	20	19	18	17	16
AUDIO_RDESC[23:16]							
15	14	13	12	11	10	9	8
AUDIO_RDESC[15:8]							
7	6	5	4	3	2	1	0
AUDIO_RDESC[7:0]							

Bits	Descriptions	
[31:0]	AUDIO_RDESC[31:0]	32-bit Record Destination Current Address The AUDIO_RDESC [31:0] bits are read only.

DMA Play Destination Base Address (ACTL_PDESB)

Register	Address	R/W	Description	Reset Value
ACTL_PDESB	I2S_BA + 0x14	R/W	DMA play destination base address	0x0000_0000

The value in ACTL_PDESB register is the play destination base address of DMA, and only could be changed by CPU.

31	30	29	28	27	26	25	24
AUDIO_PDESB[31:24]							
23	22	21	20	19	18	17	16
AUDIO_PDESB[23:16]							
15	14	13	12	11	10	9	8
AUDIO_PDESB[15:8]							
7	6	5	4	3	2	1	0
AUDIO_PDESB[7:0]							

Bits	Descriptions	
[31:0]	AUDIO_PDESB[31:0]	32-bit play destination base address The AUDIO_PDESB [31:0] bits are read/write.

DMA Play Destination Address Length (ACTL_PDES_LENGTH)

Register	Address	R/W	Description	Reset Value
ACTL_PDES_LENGTH	I2S_BA + 0x18	R/W	DMA play destination address length	0x0000_0000

The value in ACTL_PDES_LENGTH register is the play destination address length of DMA, and the register could only be changed by CPU.

31	30	29	28	27	26	25	24
AUDIO_PDES_L[31:24]							
23	22	21	20	19	18	17	16
AUDIO_PDES_L[23:16]							
15	14	13	12	11	10	9	8
AUDIO_PDES_L[15:8]							
7	6	5	4	3	2	1	0
AUDIO_PDES_L[7:0]							

Bits	Descriptions	
[31:0]	AUDIO_PDES_L[31:0]	32-bit play destination address length The AUDIO_PDES_L [31:0] bits are read/write. The minimum value for 16-bits mode is 0x20 and for 24-bits mode is 0x40.

DMA Play Destination Current Address (ACTL_PDESC)

Register	Address	R/W	Description	Reset Value
ACTL_PDESC	I2S_BA + 0x1C	R	DMA play destination current address	0x0000_0000

The value in ACTL_PDESC is the play destination current address of DMA; this register could only be read by CPU.

31	30	29	28	27	26	25	24
AUDIO_PDESC[31:24]							
23	22	21	20	19	18	17	16
AUDIO_PDESC[23:16]							
15	14	13	12	11	10	9	8
AUDIO_PDESC[15:8]							
7	6	5	4	3	2	1	0
AUDIO_PDESC[7:0]							

Bits	Descriptions	
[31:0]	AUDIO_PDESC[31:0]	32-bit Play Destination Current Address The AUDIO_PDESC [31:0] bits are read only.

Audio Controller Record Status Register (ACTL_RSR)

Register	Address	R/W	Description	Reset Value
ACTL_RSR	I2S_BA + 0x20	R/W	Audio controller FIFO and DMA status register for record	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
R_DMA_RIA_SN[2:0]			Reserved		R_FIFO_FULL	R_FIFO_EMPTY	R_DMA_RIA_IRQ

Bits	Descriptions	
[31:8]	RESERVED	RESERVED
[7:5]	R_DMA_RIA_SN[2:0]	Record DMA Reach indicative Address Section number bit R_DMA_IRQ_SEL = 01, R_DMA_RIA_SN[2:0] = 1, 0 R_DMA_IRQ_SEL = 10, R_DMA_RIA_SN[2:0] = 1, 2, 3, 0 R_DMA_IRQ_SEL = 11, R_DMA_RIA_SN[2:0] = 1, 2, 3, 4, 5, 6, 7, 0 The R_DMA_RIA_SN[2:0] bits are read only
[4:3]	RESERVED	RESERVED
[2]	R_FIFO_FULL	Record FIFO Full Indicator bit When record FIFO is full and the record data is written in R_FIFO, the R_FIFO_FULL bit is set to 1. This bit indicates the FIFO full error is happened. 0: the R_FIFO full error is not happened. 1: the R_FIFO full error is happened. The R_FIFO_FULL bit is readable, and only can be clear by write "1" to this bit.
[1]	R_FIFO_EMPTY	Record FIFO EMPTY Indicator bit When record FIFO is empty and the record data is read from FIFO, the R_FIFO_EMPTY bit is set to 1. This bit indicates the FIFO empty error is happened. 0: the R_FIFO empty error is not happened. 1: the R_FIFO empty error is happened. The R_FIFO_EMPTY bit is readable, and only can be clear by write "1" to this bit.
[0]	R_DMA_RIA_IRQ	Record DMA Reach indicative Address Interrupt Request bit 0: Record DMA address does not reach the indicative address by R_DMA_IRQ_SEL bits. 1: Record the DMA address reach the indicative address by R_DMA_IRQ_SEL bits. The R_DMA_RIA_IRQ bit is readable, and only can be clear by write "1" to

		this bit

Audio Controller Playback Status Register (ACTL_PSR)

Register	Address	R/W	Description	Reset Value
ACTL_PSR	I2S_BA + 0x24	R/W	Audio controller FIFO and DMA status register for playback	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
P_DMA_RIA_SN[2:0]			DMA_CNTER_IRQ	DMA_DATA_ZERO_IRQ	P_FIFO_FULL	P_FIFO_EMPTY	P_DMA_RIA_IRQ

Bits	Descriptions	
[31:8]	RESERVED	RESERVED
[7:5]	P_DMA_RIA_SN[2:0]	Play DMA Reach indicative Address Section Number bit P_DMA_IRQ_SEL = 01, P_DMA_RIA_SN[2:0] = 1, 0 P_DMA_IRQ_SEL = 10, P_DMA_RIA_SN[2:0] = 1, 2, 3, 0 P_DMA_IRQ_SEL = 11, P_DMA_RIA_SN[2:0] = 1, 2, 3, 4, 5, 6, 7, 0 The P_DMA_RIA_SN[2:0] bits are read only
[4]	DMA_CNTER_IRQ	DMA counter IRQ 0: not found DMA_COUNTER to zero 1: DMA_COUNTER counter down to zero The DMA_CNTER_IRQ bit is readable , and only can be clear by write "1" to clear this bit
[3]	DMA_DATA_ZERO_IRQ	DMA_DATA zero IRQ 0: not found DMA DATA is zero or sign change(two channel) 1: found DMA DATA is zero or sign change (two channel) The DMA_DATA_ZERO_IRQ bit is readable , and only can be clear by write "1" to clear this bit
[2]	P_FIFO_FULL	Playback FIFO Full Indicator bit When playback FIFO is empty and the playback data is read from FIFO, the P_FIFO_FULL bit is set to 1. This bit indicates the FIFO full error is happened. 0: the P_FIFO full error is not happened. 1: the P_FIFO full error is happened. The TP_FIFO_FULL bit is readable, and only can be clear by write "1" to this bit.
[1]	P_FIFO_EMPTY	Playback FIFO EMPTY Indicator bit When playback FIFO is empty and the playback data is read from FIFO, the P_FIFO_EMPTY bit is set to 1. This bit indicates the FIFO empty error is happened.

		0: the P_FIFO empty error is not happened. 1: the P_FIFO empty error is happened. The P_FIFO_EMPTY bit is readable, and only can be clear by write "1" to this bit.
[0]	P_DMA_RIA_IRQ	Playback DMA Reach Indicative Address Interrupt Request bit 0: Playback DMA address does not reach the specific address by P_DMA_IRQ_SEL bits. 1: Playback DMA address reach the indicative address by P_DMA_IRQ_SEL bits. The P_DMA_RIA_IRQ bit is readable, and only can be clear by write "1" to this bit

Play (3004 Bit7,5)	DMA_DATA_Zero_EN (3004 Bit 3)	DMA_DATA_Zero_IRQ (3024 Bit 3)	
1	0	0	play
1	0	0	Play
1	1	0	Play
1	1	1	Play(output 0,DMA not stop)
0	0	0	stop
0	0	0	Stop
0	1	0	Play
0	1	1	Stop(DMA stop and output 0 after output data is zero)

I2S Control Register (ACTL_I2SCON)

Register	Address	R/W	Description	Reset Value
ACTL_I2SCON	I2S_BA + 0x28	R/W	I2S control register	0x0000_0000

The ACTL_I2SCON is the I2S basic operation control register.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PRS[3:0]			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
BCLK_SEL[1:0]		WS_SEL	MCLK_SEL	FORMAT	Reserved		

Bits	Descriptions	
[31:20]	RESERVED	RESERVED
[19:16]	PRS[3:0]	I2S Frequency PRE_SCALER Selection bits. (FPLL is the input PLL frequency, MCLK is the output main clock) 0000: MCLK=FPLL/1 0001: MCLK=FPLL/2 0010: MCLK=FPLL/3 0011: MCLK=FPLL/4 0100: MCLK=FPLL/5 0101: MCLK=FPLL/6 0110: MCLK=FPLL/7 0111: MCLK=FPLL/8 1000: RESERVED 1001: MCLK=FPLL/10 1010: RESERVED 1011: MCLK=FPLL/12 1100: RESERVED 1101: MCLK=FPLL/14 1110: RESERVED 1111: MCLK=FPLL/16 (when the division factor is 3/5/7, the duty cycle of MCLK is not 50%, the high duration is 0.5*FPLL) The PSR[3:0] bits are read/write
[15:8]	RESERVED	RESERVED
[7:6]	BCLK_SEL [1:0]	I2S Serial Data Clock Frequency Selection bit 00: The frequency of bit clock (BCLK) is MCLK/8. 01: The frequency of bit clock (BCLK) is MCLK/12. 48fs is selected (only when FS_SEL=1, this term could be selection), when FS_SEL=1, the frequency of bit clock is MCLK/8.

		1x: RESERVED The BCLK_SEL[1:0] bits are read/write																				
[5]	WS_SEL	I2S Sampling Word Selection Bit If BCLK_SEL[1:0]=00, and WS_SEL=0, 32ws is selected, the word selection (WS) = $MCLK/(8*32) = MCLK/(256)$ If BCLK_SEL[1:0]=00, and WS_SEL=1, 48ws is selected, the word selection (WS) = $MCLK/(8*48) = MCLK/(384)$ If BCLK_SEL[1:0]=01, this bit is ignored, 32ws is selected, the word selection (WS) = $MCLK/(12*32) = MCLK/(384)$ (WS is sampling rate) The WS_SEL bit is read/write Example: <table><tr><td>MCLK</td><td>Sample Rate</td><td>Sample Freq.</td><td>BCLK_SEL</td><td>WS_SEL</td></tr><tr><td>12.288MHz</td><td>32ws</td><td>48.0KHz</td><td>00</td><td>0</td></tr><tr><td>16.934MHz</td><td>32ws</td><td>44.1KHz</td><td>01</td><td>0</td></tr><tr><td>16.934MHz</td><td>48ws</td><td>44.1KHz</td><td>00</td><td>1</td></tr></table>	MCLK	Sample Rate	Sample Freq.	BCLK_SEL	WS_SEL	12.288MHz	32ws	48.0KHz	00	0	16.934MHz	32ws	44.1KHz	01	0	16.934MHz	48ws	44.1KHz	00	1
MCLK	Sample Rate	Sample Freq.	BCLK_SEL	WS_SEL																		
12.288MHz	32ws	48.0KHz	00	0																		
16.934MHz	32ws	44.1KHz	01	0																		
16.934MHz	48ws	44.1KHz	00	1																		
[4]	MCLK_SEL	MCLK Clock Selection bit 0: I2S MCLK output will follow the PRS [3:0] setting. 1: I2S MCLK output will be the same with FPLL. The MCLK_SEL bit is read/write																				
[3]	FORMAT	I2S Format Selection bit 0: I2S compatible format is selected. 1: MSB-justified format is selected. The FORMAT bit are read/write																				
[2:0]	RESERVED	RESERVED																				

DOWN_COUNTER Control Register (ACTL_COUNTER)

Register	Address	R/W	Description	Reset Value
ACTL_COUNTER	I2S_BA + 0x2C	R/W	DMA down counter register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
ACTL_COUNTER[31:24]							
23	22	21	20	19	18	17	16
ACTL_COUNTER[23:16]							
15	14	13	12	11	10	9	8
ACTL_COUNTER[15:8]							
7	6	5	4	3	2	1	0
ACTL_COUNTER[7:0]							

Bits	Descriptions	
[31:0]	ACTL_COUNTER[31:0]	ACTL_COUNTER is Read and Write Data. The ACTL_COUNTER [31:0] bits are read and write. When the register is Zero that set DMA_CNTER_IRQ bit =1.

5.10 Storage Interface Controller(SIC)

5.10.1 Overview

The Storage Interface Controller (SIC) of N9H20 Chip has SIC_DMA unit and SIC_FMI unit. The SIC_DMAM unit provides a DMA (Direct Memory Access) function for FMI to exchange data between system memory (ex. SDRAM) and shared buffer (128 bytes), and the SIC_FMI unit control the interface of SD/SDHC/SDIO/MMC or NAND/SM. The serial interface controller can support SD/SDHC/SDIO/MMC card and NAND-type flash and the FMI is cooperated with DMAC to provide a fast data transfer between system memory and cards.

5.10.2 Features

- AMBA AHB master/slave interface compatible, for data transfer and register read/write.
- Support single DMA channel and address in non-word boundary.
- Support hardware Scatter-Gather function.
- Using single 128Bytes shared buffer for data exchange between system memory and cards. (Separate into two 64 bytes ping-pong FIFO).
- Synchronous design for SIC_DMA with single clock domain, AHB bus clock (HCLK).
- Interface with DMAC for register read/write and data transfer.
- Support SD/SDHC/SDIO/MMC card.
- Supports SLC and MLC NAND type Flash.
- Adjustable NAND page sizes. (512B+spare area, 2048B+spare area, 4096B+spare area and 8192B+spare area).
- Support up to 4bit/8bit/12bit/15bit hardware ECC calculation circuit to protect data communication.
- Programmable NAND/SM timing cycle.
- Synchronous design for NAND-type flash interface with single clock domain, AHB bus clock (HCLK)
- Completely asynchronous design for Secure-Digital with two clock domains, HCLK and Engine clock, note that frequency of HCLK should be higher than the frequency of engine clock.

5.10.3 Block Diagram and Card Pad Assignment

The block diagram and Card Pad Assignment of SIC Controller is shown as following.

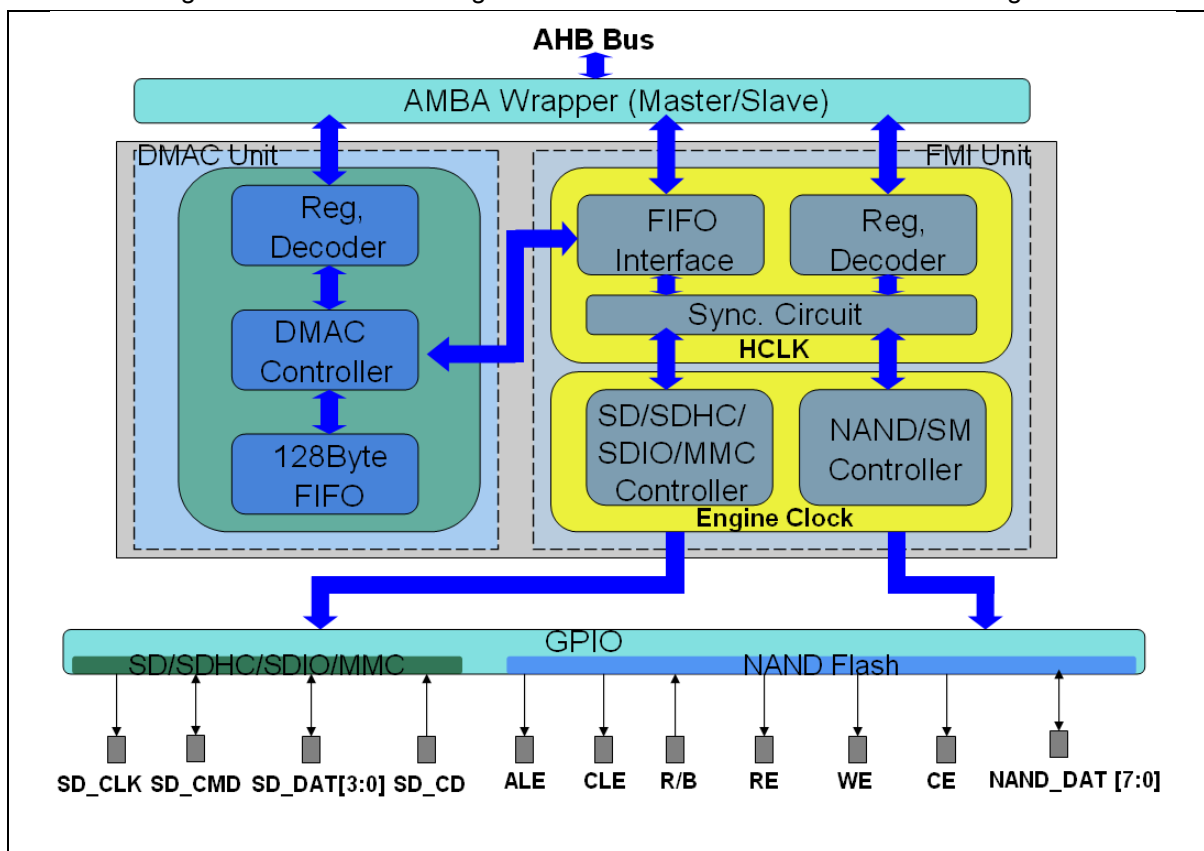


Figure 5.10-1 SIC Controller Block Diagram

	Name	Description	Note
1.	DAT0	Data Line 0 (bit0)	None.
2.	DAT1	Data Line 1 (bit1)	None.
3.	DAT2	Data Line 2 (bit2)	None.
4.	DAT3	Data Line 3 (bit3)	None.
5.	CMD	Command/Response	None.
6.	CLK	Clock pin	None.
7.	CD	Card Detect	The source can be GPIO or DAT3 (ref SDIER)

Table 5.10.3-1 SD/SDHC/SDIO/MMC Card Pad Assignment

	Name	Description	Note
1.	DAT0	Data Line 0 (bit0)	Note.
2.	DAT1	Data Line 1 (bit1)	Note.
3.	DAT2	Data Line 2 (bit2)	Note.
4.	DAT3	Data Line 3 (bit3)	Note.
5.	DAT4	Data Line 4 (bit4)	Note.
6.	DAT5	Data Line 5 (bit5)	Note.
7.	DAT6	Data Line 6 (bit6)	Note.
8.	DAT7	Data Line 7 (bit7)	Note.
9.	ALE	Address Latch Enable	Note.
10.	CLE	Command Latch Enable	Note.
11.	R/B	Ready/Busy Enable	Note.
12.	RE	Read Enable	Note.
13.	WE	Write Enable	Note.
14.	CE	Chip Enable	Note.

NAND/SM Card Pad Assignment

5.10.4 Registers Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
Shared Buffer (DMAC_BA = 0xB100_6000)				
FMI_FB_0 FMI_FB_32	DMAC_BA+0x000 DMAC_BA+0x07C	R/W	Shared Buffer (FIFO)	0x0000_0000
DMAC Registers (DMAC_BA = 0xB100_6400)				
DMACCSR	DMAC_BA+0x00	R/W	DMAC Control and Status Register	0x0000_0000
DMAC SAR	DMAC_BA+0x08	R/W	DMAC Transfer Starting Address Register	0x0000_0000
DMACBCR	DMAC_BA+0x0C	R	DMAC Transfer Byte Count Register	0x0000_0000
DMACIER	DMAC_BA+0x10	R/W	DMAC Interrupt Enable Register	0x0000_0001
DMACISR	DMAC_BA+0x14	R/W	DMAC Interrupt Status Register	0x0000_0000
FMI Global Registers (FMI_BA = 0xB100_6800)				
FMICR	FMI_BA + 0x000	R/W	Global Control and Status Register	0x0000_0000
FMIER	FMI_BA + 0x004	R/W	Global Interrupt Control Register	0x0000_0001
FMIISR	FMI_BA + 0x008	R/W	Global Interrupt Status Register	0x0000_0000
Secure-Digital Registers				
SDCR	FMI_BA + 0x020	R/W	SD Control and Status Register	0x0101_0000
SDARG	FMI_BA + 0x024	R/W	SD Command Argument Register	0x0000_0000
SDIER	FMI_BA + 0x028	R/W	SD Interrupt Control Register	0x0000_0A00
SDISR	FMI_BA + 0x02C	R/W	SD Interrupt Status Register	0x000X_008C
SDRSP0	FMI_BA + 0x030	R	SD Receiving Response Token Register 0	0x0000_0000
SDRSP1	FMI_BA + 0x034	R	SD Receiving Response Token Register 1	0x0000_0000
SDBLEN	FMI_BA + 0x038	R/W	SD Block Length Register	0x0000_01FF
SDTMOUT	FMI_BA + 0x03C	R/W	SD Response/Data-in Time-out Register	0x0000_0000
Smart-Media Registers				
SMCR	FMI_BA + 0x0A0	R/W	Smart-Media Control and Status Register	0x1E88_0090
SMTCR	FMI_BA + 0x0A4	R/W	Smart-Media Timing Control Register	0x0001_0105
SMIER	FMI_BA + 0x0A8	R/W	Smart-Media Interrupt Control Register	0x0000_0000
SMISR	FMI_BA + 0x0AC	R/W	Smart-Media Interrupt Status Register	0x000X_0000
SMCMD	FMI_BA + 0x0B0	W	Smart-Media Command Port Register	N/A
SMADDR	FMI_BA + 0x0B4	W	Smart-Media Address Port Register	N/A
SMDATA	FMI_BA + 0x0B8	R/W	Smart-Media Data Port Register	N/A
SMREAREA_	FMI_BA + 0xBC	R/W	Smart-Media Redundant Area Control Register	0x0000_0000

CTL				
SM_ECC_ST0	FMI_BA + 0xD0	R	Smart-Media ECC Error Status 0	0x0000_0000
SM_ECC_ST1	FMI_BA + 0xD4	R	Smart-Media ECC Error Status 1	0x0000_0000
SM_ECC_ST2	FMI_BA + 0xD8	R	Smart-Media ECC Error Status 2	0x0000_0000
SM_ECC_ST3	FMI_BA + 0xDC	R	Smart-Media ECC Error Status 3	0x0000_0000
Smart-Media BCH Error Address Register (ECC_ADD_BA = 0xB100_6900)				
BCH_ECC_ADD_R0	ECC_BA + 0x00	R	BCH error byte address 0	0x0000_0000
BCH_ECC_ADD_R1	ECC_BA + 0x04	R	BCH error byte address 1	0x0000_0000
BCH_ECC_ADD_R2	ECC_BA + 0x08	R	BCH error byte address 2	0x0000_0000
BCH_ECC_ADD_R3	ECC_BA + 0x0C	R	BCH error byte address 3	0x0000_0000
BCH_ECC_ADD_R4	ECC_BA + 0x10	R	BCH error byte address 4	0x0000_0000
BCH_ECC_ADD_R5	ECC_BA + 0x14	R	BCH error byte address 5	0x0000_0000
BCH_ECC_ADD_R6	ECC_BA + 0x18	R	BCH error byte address 6	0x0000_0000
BCH_ECC_ADD_R7	ECC_BA + 0x1C	R	BCH error byte address 7	0x0000_0000
Smart-Media BCH Error Data Register				
BCH_ECC_DAT_A0	ECC_BA + 0x60	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DAT_A1	ECC_BA + 0x64	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DAT_A2	ECC_BA + 0x68	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DAT_A3	ECC_BA + 0x6C	R	BCH ECC Error Data 0	0x8080_8080
Smart-Media Redundant Area Register (SMRA_BA = 0xB100_6A00)				
SM_RA0 ... SM_RA117	SMRA_BA + 0x000 ... SMRA_BA + 0x1D4	R/W	Smart-Media Redundant Area Register	N/A

5.11 SIC DMA Controller

5.11.1 Overview

The SIC_DMA Controller provides a DMA (Direct Memory Access) function for FMI controller to exchange data between system memory (ex. SDRAM) and shared buffer (128 bytes). Arbitration of DMA request between FMI is done by DMAC's bus master. Software just simply fills in the starting address and enables DMAC, and then you can let DMAC to handle the data transfer automatically.

There is a 128 bytes shared buffer inside DMAC, separate into two 64 bytes ping-pong FIFO (total 128 bytes). It can provide multi-block transfers using ping-pong mechanism for FMI. Software can access these shared buffers directly when FMI is not in busy.

5.11.2 Features

- AMBA AHB master/slave interface compatible, for data transfer and register read/write.
- Support single DMA channel and address in non-word boundary.
- Support SD/SDHC/SDIO/MMC cards in byte-access.
- Support hardware Scatter-Gather function.
- One 128 bytes shared buffer is embedded.
- Synchronous design for SIC_DMA with single clock domain, AHB bus clock (HCLK).

5.11.2.1 Symbol Diagram

The symbol diagram of DMA Controller is shown as following.

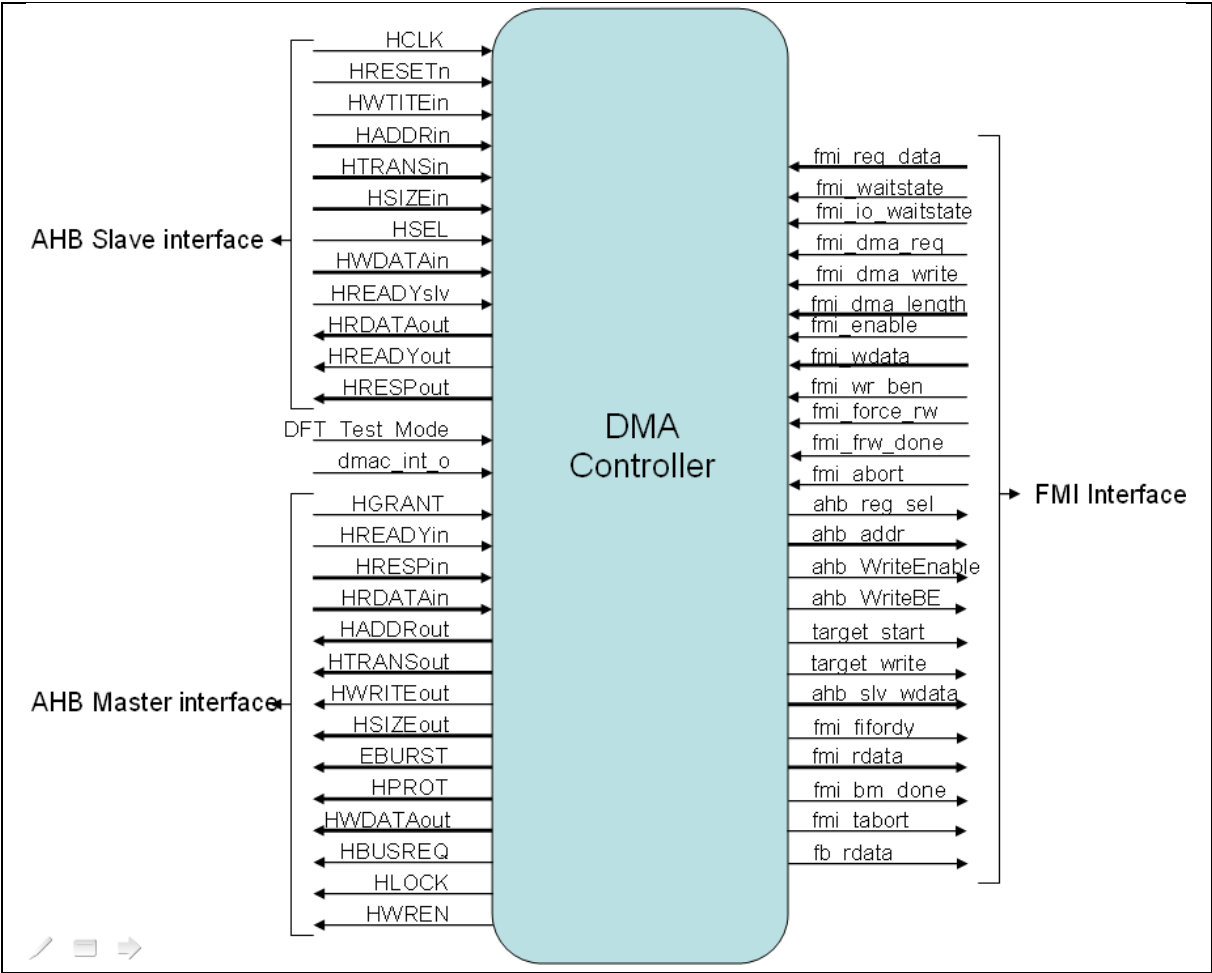


Table 5.11.2-1 MA Controller Symbol Diagram

5.11.2.2 Block Diagram

The block diagram of DMA Controller is shown as following.

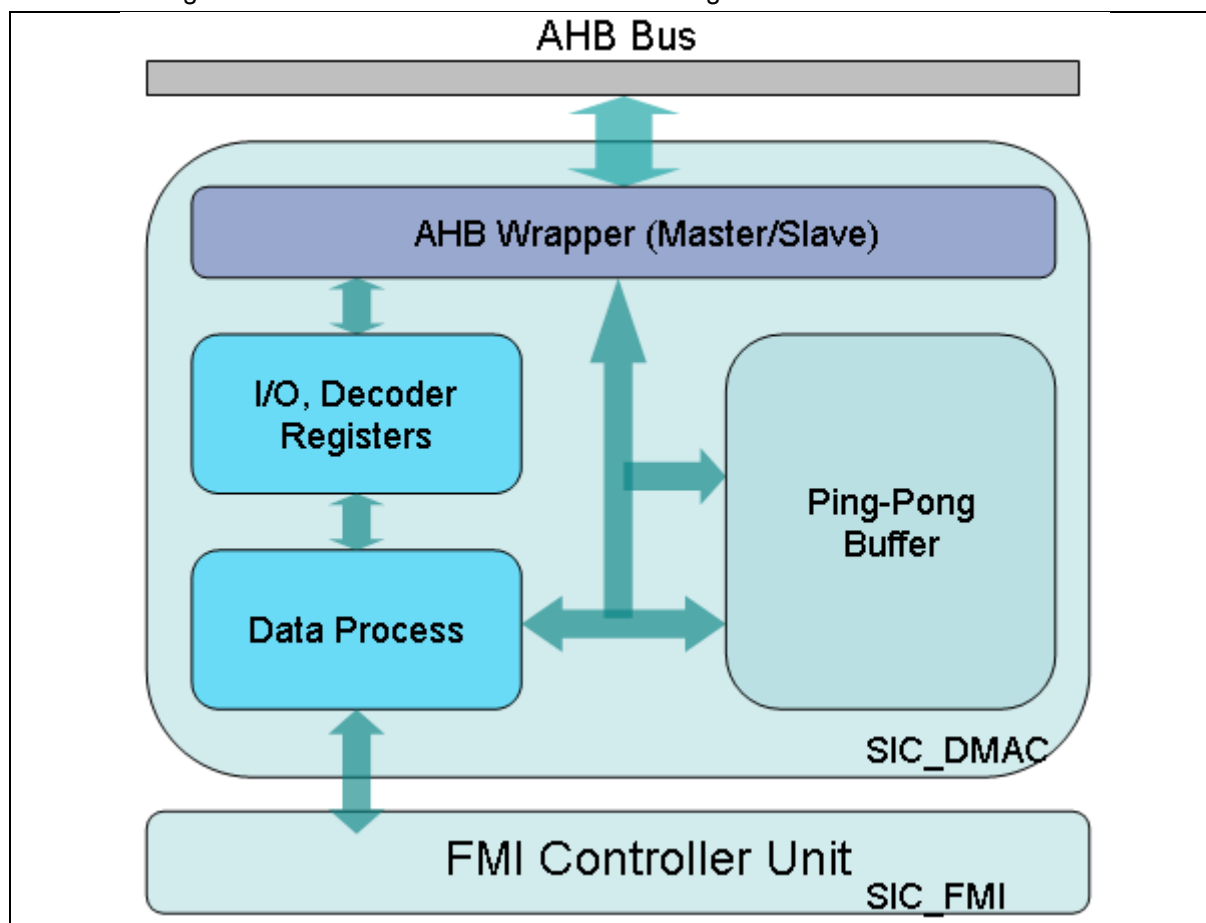


Table 5.11.2-2 DMA Controller Block Diagram

5.11.2.3 Programming Flow

Here is a simple example programming flow without DMA Scatter-Gather enable.

1. Set DMACCSR [DMACEN] to enable DMAC.
2. Fill corresponding starting address in DMACSAR for FMI.
3. Enable IP to start DMA transfer.
4. Wait IP finished, software doesn't need to take care of DMAC.

Here is a simple example programming flow with DMA Scatter-Gather enable.

1. Set DMACCSR [DMACEN] to enable DMAC and DMACCSR [SG_EN] to enable Scatter-Gather function.
2. Fill corresponding starting address of Physical Address Descriptor (PAD) table in DMACSAR for FMI.
3. When bit-0 of DMACSAR is 1, the PAD will fetch in out of order, otherwise, it's fetched in order from PAD. The first time of writing bit-0 with 1 or not is not available for this function. The bits will be available in PAD table.
4. Enable IP to start DMA transfer.
5. Wait IP finished, software doesn't need to take care of DMAC.

5.11.2.4 DMA Controller Registers Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
Shared Buffer (DMAC_BA = 0xB100_6000)				
FMI_FB_0 FMI_FB_32	DMAC_BA+0x000 DMAC_BA+0x07C	R/W	Shared Buffer (FIFO)	0x0000_0000
DMAC Registers (DMAC_BA = 0xB100_6400)				
DMACCSR	DMAC_BA+0x00	R/W	DMAC Control and Status Register	0x0000_0000
DMACSAR	DMAC_BA+0x08	R/W	DMAC Transfer Starting Address Register 2	0x0000_0000
DMACBCR	DMAC_BA+0x0C	R	DMAC Transfer Byte Count Register	0x0000_0000
DMACIER	DMAC_BA+0x10	R/W	DMAC Interrupt Enable Register	0x0000_0001
DMACISR	DMAC_BA+0x14	R/W	DMAC Interrupt Status Register	0x0000_0000

5.11.2.5 DMAC Register Detail

DMAC Control and Status Register (DMACCSR)

Register	Offset	R/W	Description	Reset Value
DMACCSR	0x00	R/W	DMAC Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						FMI_BUSY	Reserved
7	6	5	4	3	2	1	0
Reserved				SG_EN2	Reserved	SW_RST	DMACEN

Bits	Descriptions	
[31:10]	Reserved	Reserved
[9]	FMI_BUSY	FMI DMA Transfer is in progress This bit indicates if FMI is granted and doing DMA transfer or not. 0 = FMI DMA transfer is not in progress. 1 = FMI DMA transfer is in progress.
[8:4]	Reserved	Reserved
[3]	SG_EN2	Enable Scatter-Gather Function for FMI Enable DMA scatter-gather function or not. 0 = Normal operation. DMAC will treat the starting address in DMACSSAR as starting pointer of a single block memory. 1 = Enable scatter-gather operation. DMAC will treat the starting address in DMACSSAR as a starting address of Physical Address Descriptor (PAD) table. The format of these Pads' will be described later.
[2]	Reserved	Reserved
[1]	SW_RST	Software Engine Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after few clock cycles. NOTE: The software reset DMA region.
[0]	DMACEN	DMAC Engine Enable Setting this bit to 1 enables DMAC's operation. If this bit is cleared, DMAC will ignore all DMA request from ATAPI or FMI and force Bus Master into IDLE state. 0 = Disable DMAC. 1 = Enable DMAC. NOTE: If target abort is occurred, DMACEN will be cleared.

DMAC Transfer Starting Address Register (DMAC SAR)

Register	Offset	R/W	Description	Reset Value
DMAC SAR	0x08	R/W	DMAC Transfer Starting Address Register 2	0x0000_0000

31	30	29	28	27	26	25	24
DMACSA[31:24]							
23	22	21	20	19	18	17	16
DMACSA[23:16]							
15	14	13	12	11	10	9	8
DMACSA[15:8]							
7	6	5	4	3	2	1	0
DMACSA[7:0]							

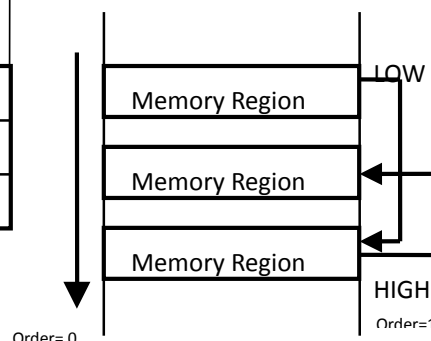
Bits	Descriptions
[31:0]	DMACSA DMA Transfer Starting Address for FMI This field indicates a 32-bit starting address of system memory (SRAM/SDRAM) for DMAC to retrieve or fill in data (for FMI engine). If DMAC is not in normal mode, this field will be interpreted as a starting address of Physical Address Descriptor (PAD) table.
[0]	ORDER Determined to the PAD table fetching is in order or out of order 0 = PAD table is fetched in order 1 = PAD table is fetched out of order Note: the bit0 is valid in scatter-gather mode when SG_EN2 = 1.

NOTE: Starting address of the SDRAM can be on word aligned or non word aligned, for example, 0x0000_0000, 0x0000_0001...

The format of PAD table must like below. Note that the total byte count of all Pads must be equal to the byte count filled in FMI engine. EOT should be set to 1 in the last descriptor.

byte 3	byte 2	byte 1	byte 0
SDRAM Physical Base			
Next Descriptor Physical Base			
EOT	Reserved		Byte Count

Physical Base Address: 32-bit
Byte Count: any byte, 65536 bytes
Bytes (bit 15~0)
EOT: End of PAD Table (bit 31)



DMAC Transfer Byte Count Register (DMACBCR)

Register	Offset	R/W	Description	Reset Value
DMACBCR	0x0C	R	DMAC Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BCNT[25:24]	
23	22	21	20	19	18	17	16
BCNT[23:16]							
15	14	13	12	11	10	9	8
BCNT[15:8]							
7	6	5	4	3	2	1	0
BCNT[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:0]	BCNT	DMA Transfer Byte Count (Read Only) This field indicates the remained byte count of DMAC transfer. The value of this field is valid only when ATAPI or FMI is busy; otherwise, it is zero.

DMAC Interrupt Enable Register (DMACIER)

Register	Offset	R/W	Description	Reset Value
DMACIER	0x10	R/W	DMAC Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						WEOT_IE	TABORT_IE

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1]	WEOT_IE	Wrong EOT Encountered Interrupt Enable 0 = Disable interrupt generation when wrong EOT is encountered. 1 = Enable interrupt generation when wrong EOT is encountered.
[0]	TABORT_IE	DMA Read/Write Target Abort Interrupt Enable 0 = Disable target abort interrupt generation during DMA transfer. 1 = Enable target abort interrupt generation during DMA transfer.

DMAC Interrupt Status Register (DMACISR)

Register	Offset	R/W	Description	Reset Value
DMACISR	0x14	R/W	DMAC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						WEOT_IF	TABORT_IF

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1]	WEOT_IF	Wrong EOT Encountered Interrupt Flag When DMA Scatter-Gather function is enabled, and EOT of the descriptor is encountered before DMA transfer finished (that means the total sector count of all PAD is less than the sector count of ATAPI or FMI), this bit will be set. 0 = No EOT encountered before DMA transfer finished. 1 = EOT encountered before DMA transfer finished. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[0]	TABORT_IF	DMA Read/Write Target Abort Interrupt Flag 0 = No bus ERROR response received. 1 = Bus ERROR response received. NOTE: This bit is read only, but can be cleared by writing '1' to it.

NOTE: When DMAC's bus master received ERROR response, it means that target abort is happened. DMAC will stop transfer and respond this event to software, ATAPI and FMI; then go to IDLE state. When target abort occurred or WEOT_IF is set, software must reset DMAC and IP, and then transfer those data again.

5.11.3 Flash Memory Interface Controller (FMI)

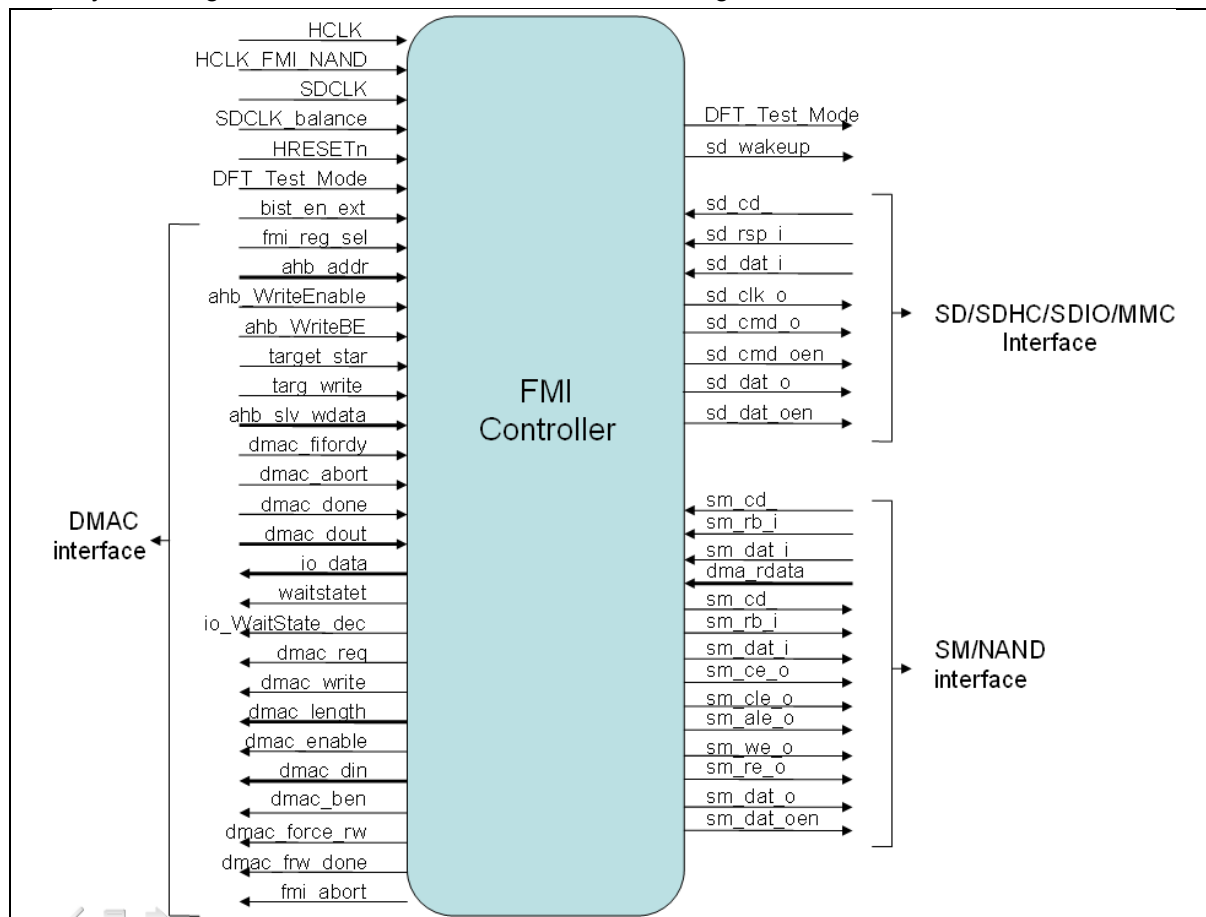
The Flash Memory Interface of N9H20 Chip supports Secure-Digital SD/SDHC/SDIO/MMC and NAND-type flash. FMI is cooperated with DMAC to provide a fast data transfer between system memory and cards. There is a single 128 bytes buffer embedded in DMAC for temporary data storage (separate into two 64 bytes ping-pong FIFO). Due to DMAC only has single channel, that means only one interface can be active at one time.

5.11.3.1 Features:

- Interface with DMAC for register read/write and data transfer
- Support SD/SDHC/SDIO/MMC card.
- Support SD/SDHC/SDIO/MMC programmable timing cycle.
- Supports SLC and MLC NAND type Flash.
- Adjustable NAND page sizes. (512B+spare area, 2048B+spare area, 4096B+spare area and 8192+spare area).
- Support up to 4bit/8bit/12bit/15bit hardware ECC calculation circuit to protect data communication.
- Support NAND/SM programmable timing cycle.
- Using single 128Bytes shared buffer for data exchange between system memory and cards. (Separate into two 64 bytes ping-pong FIFO)
- Synchronous design for NAND-type flash interface with single clock domain, AHB bus clock (HCLK)
- Completely asynchronous design for Secure-Digital with two clock domains, HCLK and Engine clock, note that frequency of HCLK should be higher than the frequency of engine clock.

5.11.3.2 Symbol Diagram

The symbol diagram of FMI Controller is shown as following.



FMI Controller Symbol Diagram

5.11.4 Block Diagram

A simple block diagram of FMI Controller is shown as following.

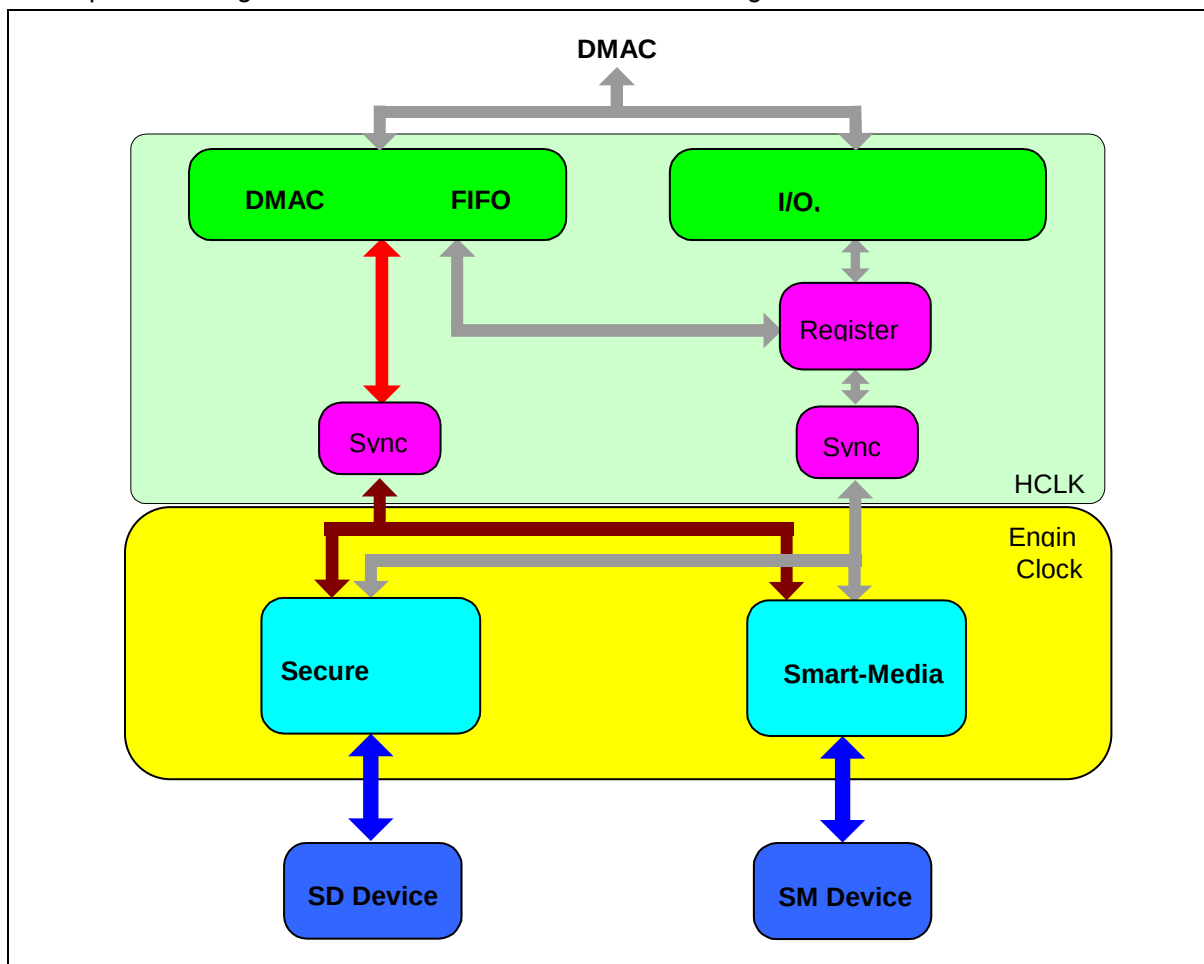


Figure 5.11-1 FMI Controller Block Diagram

5.11.5 Function Description

5.11.5.1 Secure-Digital (SD)

FMI provides an interface for SD/SDHC/SDIO/MMC card access. This SD controller provides 3 SD ports –port0, port1 and port2. Each port can provide 1-bit/4-bit data bus mode for SD, but only port0 have card detect function and SDIO interrupt.

SD controller uses an independent clock source named SDCLK as engine clock. SDCLK can be completely asynchronous with system clock HCLK, software can change SD clock arbitrary. Note that HCLK should be faster than SDCLK.

This SD controller can generate all types of 48-bit command to SD card and retrieve all types of response from SD card. After response in, the content of response will be stored at SDRSP0 and SDRSP1. SD controller will calculate CRC-7 and check its correctness for response. If CRC-7 is error, SDISR [CRC_IF] will be set and SDISR [CRC-7] will be '0'. For response R1b, software should notice that after response in, SD card will put busy signal on data line DAT0; software should check this status with clock polling until it became high. For response R3, CRC-7 is invalid; but SD controller will still calculate CRC-7 and get an error result, software should ignore this error and clear SDISR [CRC_IF] flag.

This SD controller is composed of two state machines – command/response part and data part. For command/response part, the trigger bits are CO_EN, RI_EN, R2_EN, CLK74_OE and CLK8_OE in SDCR. If software enables all of these bits, the execution priority will be CLK74_OE → CO_EN → RI_EN/R2_EN → CLK8_OE, note that RI_EN and R2_EN can't be triggered at the same time. For data part, there are DI_EN and DO_EN for choose. Software can only trigger one of them at one time. If DI_EN is triggered, SD controller waits start bit from data line DAT0 immediately, and then get specified amount data from SD card. After data-in, SD controller will check CRC-16 correctness; if it is error, SDISR [CRC_IF] will be set and SDISR [CRC-16] will be '0'. If DO_EN is triggered, SD controller will wait response in finished, and then send specified amount data to SD card. After data-out, SD controller will get CRC status from SD card and check its correctness; it should be '010', otherwise SDISR [CRC_IF] will be set and SDISR [CRCSTAT] will be the value it received.

If R2_EN is triggered, SD controller will receive response R2 (136 bits) from SD card, CRC-7 and end bit will be dropped. The receiving data will be placed at DMAC's buffer, starting from address offset 0x0.

This SD controller also provides multiple block transfer function (change SDBLEN to change the block length). Software can use this function to accelerate data transfer throughput. If CRC-7, CRC-16 or CRC status is error, SD controller will stop transfer and set SDISR [CRC_IF], software should do engine reset when this situation occurred.

There is a hardware time-out mechanism for response in and data in inside SD engine. Software can specify a 24-bit time-out value at SDTMOUT, and then SD controller will decide when to time-out according to this value.

NAND-type Flash/Smart-Media Controller.

FMI provides an interface for NAND-type Flash/Smart-Media access. It supports 512bytes/page, 2048bytes/page, 4096bytes/page and 8192bytes/page NAND. This NAND-type Flash controller provides all required signals for NAND flash, including R/-B, -CE, CLE, ALE, -WE, -RE and data pins. It has direct command port, address port and data port for software to use. When software writes to command port, NAND controller will generate appropriate signal to NAND. When software writes to address port without SMADDR [EOA] set, NAND controller will generate an address cycle to NAND, but do not clear ALE until software writes the last address cycle with SMADDR [EOA] set. In this way, software can generate address cycle arbitrarily. For example, if software wants to write 4 address cycles to NAND, you should write 3 addresses without SMADDR [EOA] set, and then write the last one address with SMADDR [EOA] set. NAND controller also provides a status and an interrupt flag of R/-B (READY/-BUSY) pin. The interrupt flag will be set only when rising edge is encountered on R/-B pin.

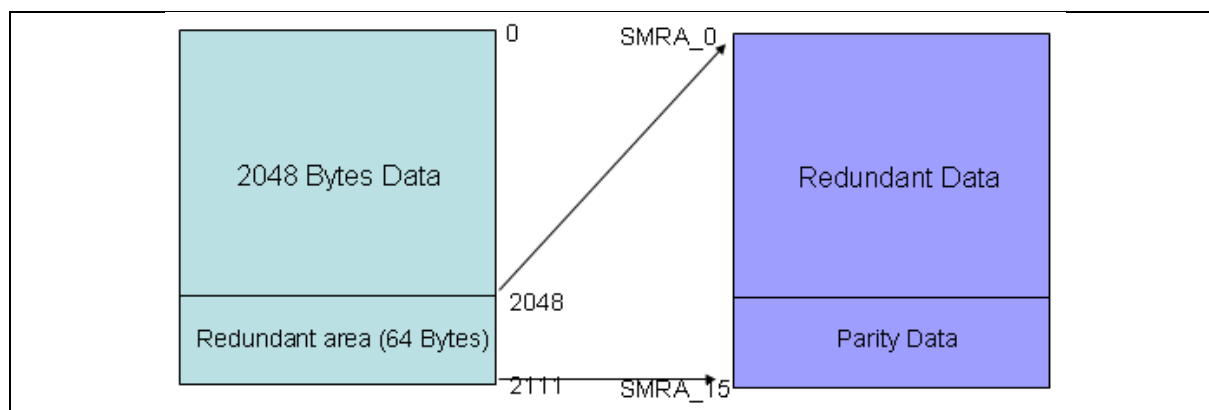
There are four page sizes for choose, 512bytes/page, 2048bytes/page, 4096bytes/page and 8192bytes/page. Using SMCR [PSIZE] to select your NAND type. Software can use DMA function for data transfer to increase your performance. For different model of NAND, software should adjust the timing parameter at SMTCR to meet its specification. Adjust timing parameter can also improve performance of data transfer.

In error recovery part, there is a BCH algorithm inside this NAND controller. The BCH algorithm can correct up to 4 bits errors or 8 bits errors or 12 bits errors or 15 bits errors. Software can read SMISR [ECC_FLD_IF] to judge the error occurrence and read SM_ECC_ST0/SM_ECC_ST1/SM_ECC_ST2/SM_ECC_ST3 to judge how many error occurrence and judge those errors are correctable or not. If those errors are correctable error, software can read BCH_ECC_ADDRx and BCH_ECC_DATAx to correct those errors.

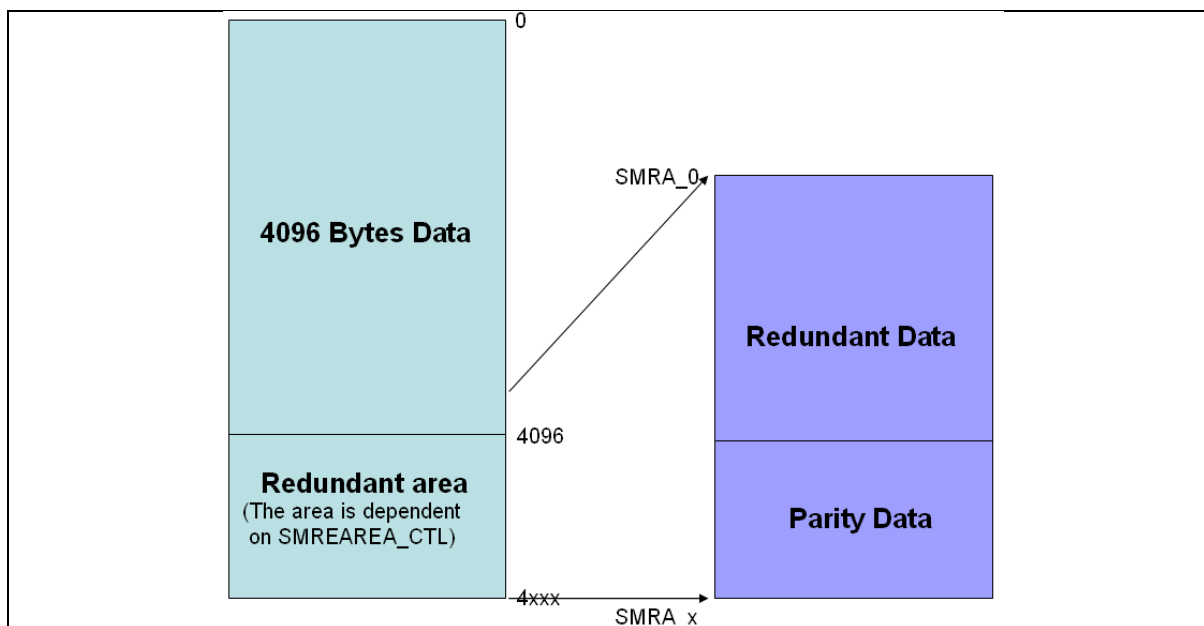
For 512/2K/4K/8K Page size NAND flash with BCH algorithm, T can be t4 or t8 or t12 or t15 which parity number and redundant number is shown in table 1.3.4-a, and the data arrangement of redundant area is shown in fig 1.3.4-a ~ 1.3.4-c.

BCH algorithm	Parity (Byte) 512 Page Size	Parity (Byte) 2048 Page Size	Parity (Byte) 4096 Page Size	Parity (Byte) 8192 Page Size
BCH_T4	8 Byte	32 Byte	64 Byte	128 Byte
BCH_T8	15 Byte	60 Byte	120 Byte	240 Byte
BCH_T12	23 Byte	92 Byte	184 Byte	368 Byte
BCH_T15	29 Byte	116 Byte	232 Byte	464 Byte

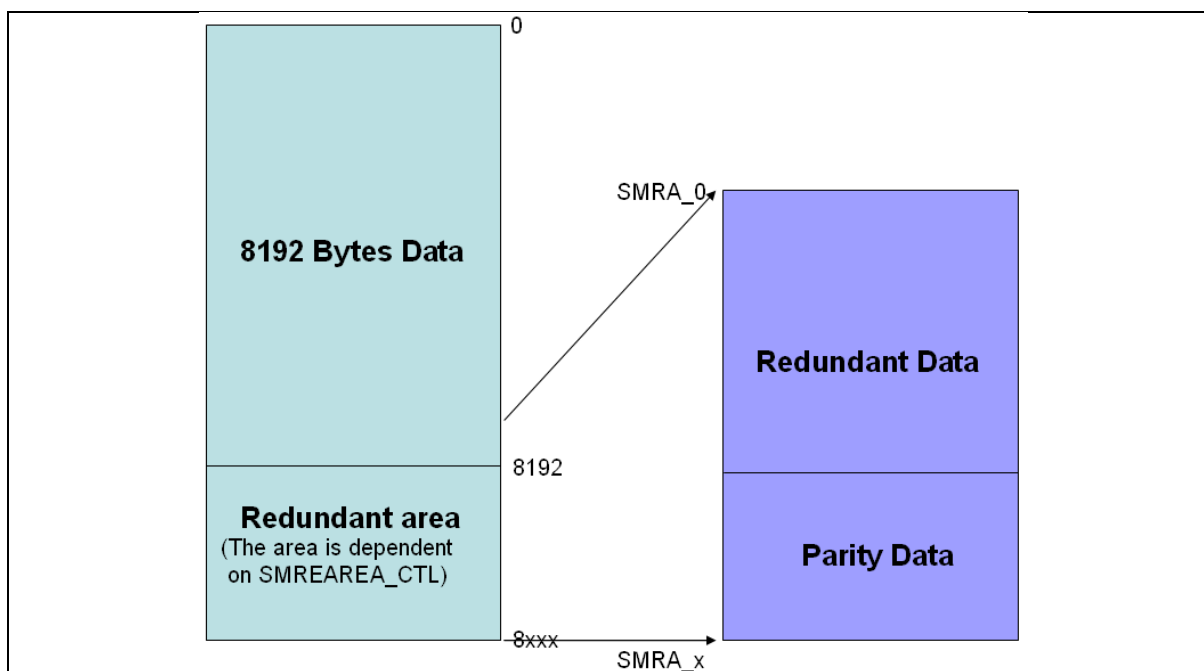
1.3.4-a Parity/Redundant number of BCH algorithm



1.3.4-a Data arrangement for 2k page size and BCH algorithm



1.3.4-b Data arrangement for 4k page size NAND flash



1.3.4-c Data arrangement for 8k page size NAND flash

5.11.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
FMI Global Registers (FMI_BA = 0XB100_6800)				
FMICR	FMI_BA + 0x00	R/W	Global Control and Status Register	0x0000_0000
FMIER	FMI_BA + 0x04	R/W	Global Interrupt Control Register	0x0000_0001
FMIISR	FMI_BA + 0x08	R/W	Global Interrupt Status Register	0x0000_0000
Secure Digital Registers (SD_BA = 0xB100_6820)				
SDCR	FMI_BA + 0x020	R/W	SD Control and Status Register	0x0101_0000
SDARG	FMI_BA + 0x024	R/W	SD Command Argument Register	0x0000_0000
SDIER	FMI_BA + 0x028	R/W	SD Interrupt Control Register	0x0000_0A00
SDISR	FMI_BA + 0x02C	R/W	SD Interrupt Status Register	0x000X_008C
SDRSP0	FMI_BA + 0x030	R	SD Receiving Response Token Register 0	0x0000_0000
SDRSP1	FMI_BA + 0x034	R	SD Receiving Response Token Register 1	0x0000_0000
SDBLEN	FMI_BA + 0x038	R/W	SD Block Length Register	0x0000_01FF
SDTMOUT	FMI_BA + 0x03C	R/W	SD Response/Data-in Time-out Register	0x0000_0000
Smart-Media Registers				
SMCR	FMI_BA + 0x0A0	R/W	Smart-Media Control and Status Register	0x0200_0080
SMTCR	FMI_BA + 0x0A4	R/W	Smart-Media Timing Control Register	0x0001_0105
SMIER	FMI_BA + 0x0A8	R/W	Smart-Media Interrupt Control Register	0x0000_0000
SMISR	FMI_BA + 0x0AC	R/W	Smart-Media Interrupt Status Register	0x000X_0000
SMCMD	FMI_BA + 0x0B0	W	Smart-Media Command Port Register	N/A
SMADDR	FMI_BA + 0x0B4	W	Smart-Media Address Port Register	N/A
SMDATA	FMI_BA + 0x0B8	R/W	Smart-Media Data Port Register	N/A
SMREAREA_CTL	FMI_BA + 0xBC	R/W	Smart-Media Redundant Area Control Register	0x0000_0000
SM_ECC_ST0	FMI_BA + 0xD0	R	Smart-Media ECC Error Status 0	0x0000_0000
SM_ECC_ST1	FMI_BA + 0xD4	R	Smart-Media ECC Error Status 1	0x0000_0000
SM_ECC_ST2	FMI_BA + 0xD8	R	Smart-Media ECC Error Status 2	0x0000_0000
SM_ECC_ST3	FMI_BA + 0xDC	R	Smart-Media ECC Error Status 3	0x0000_0000
Smart-Media BCH Error Address Register (ECC_ADD_BA = 0xB100_6900)				
BCH_ECC_ADDR0	ECC_BA + 0x00	R	BCH error byte address 0	0x0000_0000
BCH_ECC_ADDR1	ECC_BA + 0x04	R	BCH error byte address 1	0x0000_0000
BCH_ECC_ADDR2	ECC_BA + 0x08	R	BCH error byte address 2	0x0000_0000
BCH_ECC_ADDR3	ECC_BA + 0x0C	R	BCH error byte address 3	0x0000_0000

BCH_ECC_ADDR4	ECC_BA + 0x10	R	BCH error byte address 4	0x0000_0000
BCH_ECC_ADDR5	ECC_BA + 0x14	R	BCH error byte address 5	0x0000_0000
BCH_ECC_ADDR6	ECC_BA + 0x18	R	BCH error byte address 6	0x0000_0000
BCH_ECC_ADDR7	ECC_BA + 0x1C	R	BCH error byte address 7	0x0000_0000
Smart-Media BCH Error Data Register				
BCH_ECC_DATA 0	ECC_BA + 0x60	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DATA 1	ECC_BA + 0x64	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DATA 2	ECC_BA + 0x68	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DATA 3	ECC_BA + 0x6C	R	BCH ECC Error Data 0	0x8080_8080
Smart-Media Redundant Area Register (SMRA_BA = 0xB100_6A00)				
SM_RA0	SMRA_BA + 0x000	R/W	Smart-Media Redundant Area Register	N/A
...	...			
SM_RA117	SMRA_BA + 0x1D4			

5.11.7 Register Description

Global Control and Status Register (FMICR)

Register	Offset	R/W	Description	Reset Value
FMICR	0x000	R/W	Global Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			Reserved	SM_EN	Reserved	SD_EN	SW_RST

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	Reserved	Reserved
[3]	SM_EN	Smart-Media Functionality Enable 0 = Disable SM functionality of FMI. 1 = Enable SM functionality of FMI.
[2]	Reserved	Reserved
[1]	SD_EN	Secure-Digital Functionality Enable 0 = Disable SD functionality of FMI. 1 = Enable SD functionality of FMI.
[0]	SW_RST	Software Engine Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset all FMI engines. The contents of control register will not be cleared. This bit will auto clear after few clock cycles.

NOTE: Software should only enable one engine at one time, or FMI will work abnormal.

Global Interrupt Control Register (FMIER)

Register	Offset	R/W	Description	Reset Value
FMIER	0x004	R/W	Global Interrupt Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							DTA_IE

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	DTA_IE	DMAC READ/WRITE Target Abort Interrupt Enable 0 = Disable DMAC READ/WRITE target abort interrupt generation. 1 = Enable DMAC READ/WRITE target abort interrupt generation.

Global Interrupt Status Register (FMIISR)

Register	Offset	R/W	Description	Reset Value
FMIISR	0x008	R/W	Global Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							DTA_IF

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	DTA_IF	DMAC READ/WRITE Target Abort Interrupt Flag (Read Only) This bit indicates DMAC received an ERROR response from internal AHB bus during DMA read/write operation. When Target Abort is occurred, please reset all engine. 0 = No bus ERROR response received. 1 = Bus ERROR response received. NOTE: This bit is read only, but can be cleared by writing '1' to it.

NOTE: No matter interrupt enable is turn on or not, the interrupt flag will be set when target condition is occurred.

SD Control and Status Register (SDCR)

Register	Offset	R/W	Description	Reset Value
SDCR	0x020	R/W	SD Control and Status Register	0x0101_0000

31	30	29	28	27	26	25	24
CLK_KEEP1	SDPORT		CLK_KEEP2	SDNWR			
23	22	21	20	19	18	17	16
BLK_CNT							
15	14	13	12	11	10	9	8
DBW	SW_RST	CMD_CODE					
7	6	5	4	3	2	1	0
CLK_KEEP0	CLK8_OE	CLK74_OE	R2_EN	DO_EN	DI_EN	RI_EN	CO_EN

Bits	Descriptions	
[31]	CLK_KEEP1	SD Clock Enable for Port 1 0 = Disable SD clock generation. 1 = SD clock always keeps free running.
[30:29]	SDPORT	SD Port Selection 00 = Port 0 is selected. 01 = Port 1 is selected. 10 = Port 2 is selected.
[28]	CLK_KEEP2	SD Clock Enable for Port 2 0 = Disable SD clock generation. 1 = SD clock always keeps free running.
[27:24]	SDNWR	N _{WR} Parameter for Block Write Operation This value indicates the N _{WR} parameter for data block write operation in SD clock counts. The actual clock cycle will be SDNWR+1.
[23:16]	BLK_CNT	Block Counts to Be Transferred or Received This field contains the block counts for data-in and data-out transfer. For READ_MULTIPLE_BLOCK and WRITE_MULTIPLE_BLOCK command, software can use this function to accelerate data transfer and improve performance. Note that only when SDBLEN=0x1FF, value 0x0 in this field means 256. Otherwise, don't fill 0x0 to this field. Note: For READ_MULTIPLE_BLOCK and WRITE_MULTIPLE_BLOCK command, the actual total length is BLK_CNT * BLKLEN [10:0].
[15]	DBW	SD Data Bus Width (For 1-bit / 4-bit Selection) 0 = Data bus width is 1-bit. 1 = Data bus width is 4-bit.

[14]	SW_RST	Software Engine Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and counters. The contents of control register will not be cleared (but RI_EN, DI_EN, DO_EN and R2_EN will be cleared). This bit will be auto cleared after few clock cycles.
[13:8]	CMD_CODE	SD Command Code This register contains the SD command code (0x00 – 0x3F).
[7]	CLK_KEEPO	SD Clock Enable for Port 0 0 = Disable SD clock generation. 1 = SD clock always keeps free running.
[6]	CLK8_OE	Generating 8 Clock Cycles Output Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will output 8 clock cycles. NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[5]	CLK74_OE	Initial 74 Clock Cycles Output Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will output 74 clock cycles to SD card. NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[4]	R2_EN	Response R2 Input Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will wait to receive a response R2 from SD card and store the response data into DMAC's flash buffer (exclude CRC-7). NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[3]	DO_EN	Data Output Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will transfer block data and the CRC-16 value to SD card. NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[2]	DI_EN	Data Input Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will wait to receive block data and the CRC-16 value from SD card. NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[1]	RI_EN	Response Input Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will wait to receive a response from SD card. NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).
[0]	CO_EN	Command Output Enable 0 = No effect. (Please use SDCR [SW_RST] to clear this bit.) 1 = Enable, SD host will output a command to SD card. NOTE: When operation is finished, this bit will be cleared automatically, so don't write 0 to this bit (the controller will be abnormal).

SD Command Argument Register (SDARG)

Register	Offset	R/W	Description	Reset Value
SDARG	0x024	R/W	SD Command Argument Register	0x0000_0000

31	30	29	28	27	26	25	24
SD_CMD_ARG							
23	22	21	20	19	18	17	16
SD_CMD_ARG							
15	14	13	12	11	10	9	8
SD_CMD_ARG							
7	6	5	4	3	2	1	0
SD_CMD_ARG							

Bits	Descriptions	
[31:0]	SD_CMD_ARG	SD Command Argument This register contains a 32-bit value specifies the argument of SD command from host controller to SD card. Before trigger SDCR [CO_EN], software should fill argument in this field.

SD Interrupt Control Register (SDIER)

Register	Offset	R/W	Description	Reset Value
SDIER	0x028	R/W	SD Interrupt Control Register	0x0000_0A00

31	30	29	28	27	26	25	24
Reserved	CD0SRC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	WKUP_EN	DITO_IE	RITO_IE	Reserved	SDIO0_IE	Reserved	CD0_IE
7	6	5	4	3	2	1	0
Reserved						CRC_IE	BLKD_IE

Bits	Descriptions	
[31]	Reserved	Reserved
[30]	CD0SRC	SD0 Card Detect Source Selection 0 = From SD0 card's DAT3 pin. 1 = From GPIO pin.
[29:15]	Reserved	Reserved
[14]	WKUP_EN	Wake-Up Signal Generating Enable Enable/Disable wake-up signal generating of SD host when SDIO card (current using) issues an interrupt (wake-up) via DAT [1] to host. 0 = Disable. 1 = Enable.
[13]	DITO_IE	Data Input Time-out Interrupt Enable Enable/Disable interrupts generation of SD controller when data input time-out. Time-out value is specified at SDTMOUT. 0 = Disable. 1 = Enable.
[12]	RITO_IE	Response Time-out Interrupt Enable Enable/Disable interrupts generation of SD controller when receiving response or R2 time-out. Time-out value is specified at SDTMOUT. 0 = Disable. 1 = Enable.
[11]	Reserved	Reserved
[10]	SDIO0_IE	SDIO Interrupt Enable for Port 0 Enable/Disable interrupts generation of SD host when SDIO card 0 issues an interrupt via DAT [1] to host. 0 = Disable. 1 = Enable.

[9]	Reserved	Reserved
[8]	CD0_IE	SD0 Card Detection Interrupt Enable Enable/Disable interrupts generation of SD controller when card 0 is inserted or removed. 0 = Disable. 1 = Enable.
[7:2]	Reserved	Reserved
[1]	CRC_IE	CRC-7, CRC-16 and CRC Status Error Interrupt Enable 0 = SD host will not generate interrupt when CRC-7, CRC-16 and CRC status is error. 1 = SD host will generate interrupt when CRC-7, CRC-16 and CRC status is error.
[0]	BLKD_IE	Block Transfer Done Interrupt Enable 0 = SD host will not generate interrupt when data-in (out) transfer done. 1 = SD host will generate interrupt when data-in (out) transfer done.

SD Interrupt Status Register (SDISR)

Register	Offset	R/W	Description	Reset Value
SDISR	0x02C	R/W	SD Interrupt Status Register	0x000x_008C

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					SD0DAT1	Reserved	CDPS0
15	14	13	12	11	10	9	8
Reserved		DITO_IF	RITO_IF	Reserved	SDIO0_IF	Reserved	CD0_IF
7	6	5	4	3	2	1	0
SDDAT0	CRCSTAT			CRC-16	CRC-7	CRC_IF	BLKD_IF

Bits	Descriptions	
[31:19]	Reserved	Reserved
[18]	SD0DAT1	DAT1 Pin Status of SD0 (Read Only) This bit is the DAT1 pin status of SD0.
[17]	Reserved	Reserved
[16]	CDPS0	Card Detect Status of SD0 (Read Only) This bit is the card detect pin status of SD0, and it is using for card detection. When there is a card inserted in or removed from SD0, software should check this bit to confirm if there is really a card insertion or remove.
[15:14]	Reserved	Reserved
[13]	DITO_IF	Data Input Time-out Interrupt Flag (Read Only) This bit indicates that SD host counts to time-out value when receiving data (waiting start bit). 0 = Not time-out. 1 = Data input time-out. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[12]	RITO_IF	Response Time-out Interrupt Flag (Read Only) This bit indicates that SD host counts to time-out value when receiving response or R2 (waiting start bit). 0 = Not time-out. 1 = Response time-out. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[11]	Reserved	Reserved

[10]	SDIO0_IF	SDIO 0 Interrupt Flag (Read Only) This bit indicates that SDIO card 0 issues an interrupt to host. This interrupt is designed to level sensitive. Before clear it, turn off SDIER [SDIO0_IE] first. 0 = No interrupt is issued by SDIO card 0. 1 = an interrupt is issued by SDIO card 0. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[9]	Reserved	Reserved
[8]	CD0_IF	SD0 Card Detection Interrupt Flag (Read Only) This bit indicates that SD card 0 is inserted or removed. Only when SDIER [CD0_IE] is set to 1, this bit is active. 0 = No card is inserted or removed. 1 = There is a card inserted in or removed from SD0. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[7]	SDDAT0	DAT0 Pin Status of Current Selected SD Port (Read Only) This bit is the DAT0 pin status of current selected SD port.
[6:4]	CRCSTAT	CRC Status Value of Data-out Transfer (Read Only) SD host will record CRC status of data-out transfer. Software could use this value to identify what type of error is during data-out transfer. 010 = Positive CRC status. 101 = Negative CRC status 111 = SD card programming error occurs.
[3]	CRC-16	CRC-16 Check Status of Data-in Transfer (Read Only) SD host will check CRC-16 correctness after data-in transfer. 0 = Fault. 1 = OK.
[2]	CRC-7	CRC-7 Check Status (Read Only) SD host will check CRC-7 correctness during each response in. If that response does not contain CRC-7 information (ex. R3), then software should turn off SDIER [CRC_IE] and ignore this bit. 0 = Fault. 1 = OK.
[1]	CRC_IF	CRC-7, CRC-16 and CRC Status Error Interrupt Flag (Read Only) This bit indicates that SD host has occurred CRC error during response in, data-in or data-out (CRC status error) transfer. When CRC error is occurred, software should reset SD engine. Some response (ex. R3) doesn't have CRC-7 information with it; SD host will still calculate CRC-7, get CRC error and set this flag. In this condition, software should ignore CRC error and clears this bit manually. 0 = No CRC error is occurred. 1 = CRC error is occurred. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[0]	BLKD_IF	Block Transfer Done Interrupt Flag (Read Only) This bit indicates that SD host has finished all data-in or data-out block transfer. If there is a CRC-16 error or incorrect CRC status during multiple block data transfer, the transfer will be broken and this bit will also be set. 0 = Not finished yet. 1 = Done. NOTE: This bit is read only, but can be cleared by writing '1' to it.

SD Receiving Response Token Register 0 (SDRSP0)

Register	Offset	R/W	Description	Reset Value
SDRSP0	0x030	R	SD Receiving Response Token Register 0	0x0000_0000

31	30	29	28	27	26	25	24
SD_RSP_TK0							
23	22	21	20	19	18	17	16
SD_RSP_TK0							
15	14	13	12	11	10	9	8
SD_RSP_TK0							
7	6	5	4	3	2	1	0
SD_RSP_TK0							

Bits	Descriptions	
[31:0]	SD_RSP_TK0	SD Receiving Response Token 0 SD host controller will receive a response token for getting a reply from SD card when SDCR [RI_EN] is set. This field contains response bit 47-16 of the response token.

SD Receiving Response Token Register 1 (SDRSP1)

Register	Offset	R/W	Description	Reset Value
SDRSP1	0x034	R	SD Receiving Response Token Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SD_RSP_TK1							

Bits	Descriptions	
[7:0]	SD_RSP_TK1	SD Receiving Response Token 1 SD host controller will receive a response token for getting a reply from SD card when SDCR [RI_EN] is set. This register contains the bit 15-8 of the response token.

SD Block Length Register (SDBLEN)

Register	Offset	R/W	Description	Reset Value
SDBLEN	0x038	R/W	SD Block Length Register	0x0000_01FF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					SDBLEN		
7	6	5	4	3	2	1	0
SDBLEN							

Bits	Descriptions	
[10:0]	SDBLEN	SD BLOCK LENGTH in Byte Unit An 11-bit value specifies the SD transfer byte count of a block. The actual byte count is equal to SDBLEN+1. Note : The default SD block length is 512 bytes

SD Response/Data-in Time-out Register (SDTMOUT)

Register	Offset	R/W	Description	Reset Value
SDTMOUT	0x03C	R/W	SD Response/Data-in Time-out Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
SDTMOUT							
15	14	13	12	11	10	9	8
SDTMOUT							
7	6	5	4	3	2	1	0
SDTMOUT							

Bits	Descriptions	
[23:0]	SDTMOUT	SD Response/Data-in Time-out Value A 24-bit value specifies the time-out counts of response and data input. SD host controller will wait start bit of response or data-in until this value reached. The time period is depended on SD engine clock frequency. Do not write a small number into this field, or you may never get response or data due to time-out. NOTE: Fill 0x0 into this field will disable hardware time-out function.

Smart-media Control and Status Register (SMCR)

Register	Address	R/W	Description	Reset Value
SMCR	FMI_BA+0x0A0	R/W	Smart-Media Control and Status Register	0x1E88_0090

31	30	29	28	27	26	25	24
FAILED	FINISH	BIST_EN	Reserved		SM_CS1	SM_CS0	Reserved
23	22	21	20	19	18	17	16
ECC_EN	BCH_TSEL				Reserved	PSIZE	
15	14	13	12	11	10	9	8
Reserved						SRAM_INT	PROT_3BEN
7	6	5	4	3	2	1	0
ECC_CHK	Reserved		REDUN_AUT O_WEN	REDUN_REN	DWR_EN	DRD_EN	SW_RST

Bits	Descriptions	
[31]	FAILED	BIST Failed This bit indicates the BIST test of the embedded SRAM was failed or not. If it is 0 at the end of BIST, that means shared buffer has passed the test; otherwise, it is faulty. The FAILED field will be set to 1 once the BIST controller detected an error and remain high during BIST operation.
[30]	FINISH	BIST Operation Finish This bit indicates the end of BIST operation of the embedded SRAM. When BIST controller finishes all operations, this bit will be set high. 0 = No BIST operation or BIST operation is in progress. 1 = BIST operation finished.
[29]	BIST_EN	BIST Enable This bit is used to enable the BIST (Build-in Self Test) operation of embedded SRAM. Set this bit high to enable the BIST test. This bit should be enabled and cleared by software. 0 = Disable BIST operation. 1 = Enable BIST operation.
[28:27]	Reserved	Reserved
[26]	SM_CS1	Smart-Media Card1 Enable 0 = card enable. (chip select enable) 1 = card disable. (chip select disable)
[25]	SM_CS0	Smart-Media Card0 Enable 0 = card enable. (chip select enable) 1 = card disable. (chip select disable)
[24]	Reserved	Reserved

[23]	ECC_EN	ECC Algorithm Enable This field is used to select the ECC algorithm for data protecting. The BCH algorithm can correct 4 or 8 or 12 or 15 bits. 1 = Enable BCH code encode/decode. 0 = Disable BCH code encode/decode. Note: If disable ECC_EN and when read data from NAND, NAND controller will ignore its ECC check result. When write data to NAND, NAND controller will write out 0xFF to every parity field. Note: The ECC algorithm only protects data area and hardware ECC parity code, software can choose protect software redundant data first 3 bytes by setting SMCR [PROT_3BEN] or not protect software redundant data.
[22:19]	BCH_TSEL	BCH Correct Bit Selection This field is used to select BCH correct bits for data protecting. For BCH algorithm, T can be 4 or 8 or 12 or 15 for choosing (correct 4 or 8 or 12 or 15 bits). 1000 = Using BCH T15 to encode/decode (T15). 0100 = Using BCH T12 to encode/decode (T12). 0010 = Using BCH T8 to encode/decode (T8). 0001 = Using BCH T4 to encode/decode (T4).
[18]	Reserved	Reserved
[17:16]	PSIZE	Page Size of NAND This bit indicates the page size of NAND. There are four page sizes for choose, 512bytes/page, 2048bytes/page, 4096bytes/page and 8192bytes/page. 00 = Page size is 512bytes/page. 01 = Page size is 2048bytes/page. 10 = Page size is 4096bytes/page. 11 = Page size is 8192bytes/page.
[15:10]	Reserved	Reserved
[9]	SRAM_INT	SRAM Initial 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal SN_RA0~SM_RA1 to 0xFFFF_FFFF. The contents of control register will not be cleared. This bit will be auto cleared after few clock cycles.
[8]	PROT_3BEN	Protect_3Byte Software Data Enable The ECC algorithm only protects data area and hardware ECC parity code, software can choose protect software redundant data first 3 bytes by setting SMCR [PROT_3BEN] or not protect software redundant data. 0 = Not protect software redundant data. 1 = Protect software redundant data first 3 bytes.
[7]	ECC_CHK	None Used Field ECC Check After Read Page Data 0 = Disable. NAND controller will always check ECC result for each field, no matter it is used or not. 1 = Enable. NAND controller will check 1's count for byte 2, 3 of redundant data of the ECC in each field. If count value is greater than 8, NAND controller will treat this field as none used field; otherwise, it's used. If that field is none used field, NAND controller will ignore its ECC check result.
[6:5]	Reserved	Reserved

[4]	REDUN_AUTO_WEN	Redundant Area Auto Write Enable This field is used to auto write redundant data out to NAND Flash card. The redundant data area is dependent on SMREAREA_CTL register. 0 = Enable auto write redundant data out to NAND flash. 1 = Disable auto write redundant data out to NAND flash.
[3]	REDUN_REN	Redundant Area Read Enable This bit enables NAND controller to transfer redundant data from Smart-Media card or NAND type flash into SMRA, the data size is dependent on SMREAREA_CTL register. 0 = No effect. 1 = Enable read redundant data transfer. NOTE: When transfer completed, this bit will be cleared automatically.
[2]	DWR_EN	DMA Write Data Enable This bit enables NAND controller to transfer data (1 page) from DMAC's embedded frame buffer into Smart-Media card or NAND type flash. 0 = No effect. 1 = Enable DMA write data transfer. NOTE: When DMA transfer completed, this bit will be cleared automatically.
[1]	DRD_EN	DMA Read Data Enable This bit enables NAND controller to transfer data (1 page) from Smart-Media card or NAND type flash into DMAC's embedded frame buffer. 0 = No effect. 1 = Enable DMA read data transfer. NOTE: When DMA transfer completed, this bit will be cleared automatically.
[0]	SW_RST	Software Engine Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and counters (include SMCR [DWR_EN] and SMCR [DRD_EN]). The contents of control register will not be cleared. This bit will be auto cleared after few clock cycles.

Smart-media Timing Control Register (SMTCR)

Register	Offset	R/W	Description	Reset Value
SMTCR	0x0A4	R/W	Smart-Media Timing Control Register	0x0001_0105

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	CALE_SH						
15	14	13	12	11	10	9	8
HI_WID							
7	6	5	4	3	2	1	0
LO_WID							

Bits	Descriptions	
[31:23]	Reserved	Reserved
[22:16]	CALE_SH	CLE/ALE Setup/Hold Time This field controls the CLE/ALE setup/hold time to $\overline{\text{WE}}$. The setup/hold time can be calculated using following equation: $t_{\text{CLS}} = (\text{CALE\_SH} + 1) * T_{\text{AHB}}$ $t_{\text{CLH}} = ((\text{CALE\_SH} * 2) + 2) * T_{\text{AHB}}$ $t_{\text{ALS}} = (\text{CALE\_SH} + 1) * T_{\text{AHB}}$ $t_{\text{ALH}} = ((\text{CALE\_SH} * 2) + 2) * T_{\text{AHB}}$
[15:8]	HI_WID	Read/Write Enable Signal High Pulse Width This field controls the high pulse width of signals $\overline{\text{RE}}$ and $\overline{\text{WE}}$ while H/W mode page access is enabled. The pulse width is a multiple of period of AHB bus clock. (The actual width time will be $[\text{clock period} * (\text{HI\_WID} + 1)]$)
[7:0]	LO_WID	Read/Write Enable Signal Low Pulse Width This field controls the low pulse width of signals $\overline{\text{RE}}$ and $\overline{\text{WE}}$ while H/W mode page access is enabled. The pulse width is a multiple of period of AHB bus clock. (The actual width time will be $[\text{clock period} * (\text{LO\_WID} + 1)]$)

NOTE1: The reset value is calculated base on 100MHz AHB Clock.

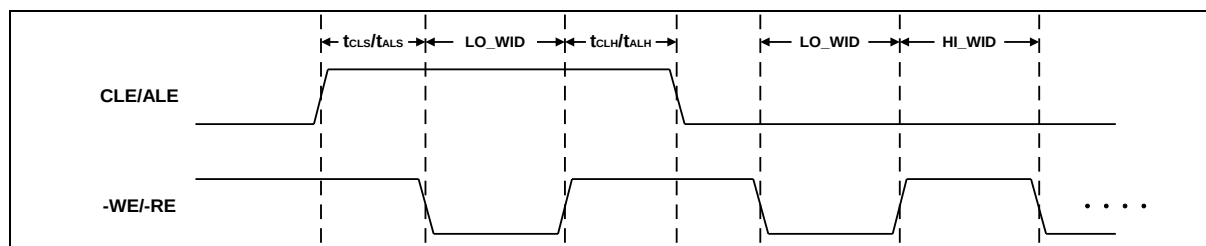


Figure 5.11-2 Timing Effect of Above 3 Registers

Smart-media Interrupt Control Register (SMIER)

Register	Address	R/W	Description	Reset Value
SMIER	FMI_BA+0x0A8	R/W	Smart-Media Interrupt Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				RB1_IE	RB0_IE	Reserved	
7	6	5	4	3	2	1	0
Reserved					ECC_FLD_IE	Reserved	DMA_IE

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	RB1_IE	Ready/-Busy 1 Rising Edge Detect Interrupt Enable 0 = Disable R/-B rising edge detect interrupt generation. 1 = Enable R/-B rising edge detect interrupt generation.
[10]	RB0_IE	Ready/-Busy Rising Edge Detect Interrupt Enable 0 = Disable R/-B rising edge detect interrupt generation. 1 = Enable R/-B rising edge detect interrupt generation.
[9:8]	Reserved	Reserved
[7:3]	Reserved	Reserved
[2]	ECC_FLD_IE	ECC Field Check Error Interrupt Enable This bit can check the ECC error on each field (512bytes) of data transfer. Software can enable this bit to detect error and do error correction. 0 = Disable. 1 = Enable
[1]	Reserved	Reserved
[0]	DMA_IE	DMA Read/Write Data Complete Interrupt Enable 0 = Disable DMA read/write data complete interrupt generation. 1 = Enable DMA read/write data complete interrupt generation.

Smart-media Interrupt Status Register (SMISR)

Register	Address	R/W	Description	Reset Value
SMISR	FMI_BA+0x0AC	R/W	Smart-Media Interrupt Status Register	0x000X_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				RB1_Status	RB0_Status	Reserved	
15	14	13	12	11	10	9	8
Reserved				RB1_IF	RB0_IF	Reserved	
7	6	5	4	3	2	1	0
Reserved					ECC_FLD_IF	Reserved	DMA_IF

Bits	Descriptions	
[31:19]	Reserved	Reserved
[19]	RB1_	Ready/-Busy 1 Pin Status (Read Only) This bit reflects the Ready/-Busy pin status of Smart-Media card.
[18]	RB0_	Ready/-Busy 0 Pin Status (Read Only) This bit reflects the Ready/-Busy pin status of Smart-Media card.
[17:16]	Reserved	Reserved
[15:11]	Reserved	Reserved
[10]	RB_IF	Ready/-Busy Rising Edge Detect Interrupt Flag (Read Only) 0 = R/-B rising edge is not detected. 1 = R/-B rising edge is detected. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[9:3]	Reserved	Reserved
[2]	ECC_FLD_IF	ECC Field Check Error Interrupt Flag (Read Only) This bit can check the ECC error on each field (512bytes) of data transfer. Software can read this bit to judge the error occurrence. 0 = Disable. 1 = Enable NOTE: This bit is read only, but can be cleared by writing '1' to it.
[1]	Reserved	Reserved
[0]	DMA_IF	DMA Read/Write Data Complete Interrupt Flag (Read Only) 0 = DMA read/write transfer is not finished yet. 1 = DMA read/write transfer is done. NOTE: This bit is read only, but can be cleared by writing '1' to it.

Smart-media Command Port Register (SMCMD)

Register	Address	R/W	Description	Reset Value
SMCMD	FMI_BA+0x0B0	W	Smart-Media Command Port Register	N/A

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SMCMD							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	SMCMD	Smart-Media Command Port When CPU writes to this port, SM H/W circuit will send a command to Smart-Media card.

Smart-media Address Port Register (SMADDR)

Register	Address	R/W	Description	Reset Value
SMADDR	FMI_BA+0x0B4	W	Smart-Media Address Port Register	N/A

31	30	29	28	27	26	25	24
EOA	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SMADDR							

Bits	Descriptions	
[31]	EOA	End of Address Writing this bit to tell SM host if this address is the last one or not. When software first writes to address port with this bit cleared, SM host will set ALE pin to active (HIGH). After the last address is written (with this bit set), SM host will set ALE pin to inactive (LOW). 0 = Not the last address cycle. 1 = the last one address cycle.
[30:8]	Reserved	Reserved
[7:0]	SMADDR	Smart-Media Address Port When CPU writes to this port, SM H/W circuit will send an address to Smart-Media card.

Smart-media Data Port Register (SMDATA)

Register	Address	R/W	Description	Reset Value
SMDATA	FMI_BA+0x0B8	R/W	Smart-Media Data Port Register	N/A

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
SMDATA							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	SMDATA	Smart-Media Data Port CPU can access NAND's memory array through this data port. When CPU WRITE, the lower 8-bit data from CPU will appear on the data bus of NAND controller. When CPU READ, NAND controller will get 8-bit data from data bus.

Smart-media Redundant Area Control Register (SMREAREA_CTL)

Register	Address	R/W	Description	Reset Value
SMREAREA_CTL	FMI_BA+0x0BC	R/W	Smart-Media Redundant Area Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MECC							
23	22	21	20	19	18	17	16
MECC							
15	14	13	12	11	10	9	8
Reserved							REA128_ext
7	6	5	4	3	2	1	0
REA128_ext							

Bits	Descriptions	
[31:16]	MECC	Mask ECC During Write Page Data. These 16 bits registers indicate NAND controller to write out ECC parity or just 0xFF for each field (every 512 bytes) the real parity data will be write out to SMRAx. 0x00 = Do not mask the ECC parity for each field. 0x01 = Mask ECC parity and write out FF to NAND ECC parity for 512 Bytes page size or 2K/4K/8K page size first 512 field. 0x02 = Mask ECC parity and write out FF to NAND ECC parity for 512 Bytes page size or 2K/4K/8K page size second 512 field. 0xxx = Mask ECC parity and write out FF to NAND ECC parity for 512 Bytes page size or 2K/4K/8K page size each 512 field.
[15:9]	Reserved	Reserved
[8:0]	REA128_ext	Redundant Area 128Byte Enable These bits indicate NAND flash extended redundant area. If SMCR [PSIZE] = 2'b00, this field will be set 0x10(16bytes) automatically. If SMCR [PSIZE] = 2'b01, this field will be set 0x40(64bytes) automatically. If SMCR [PSIZE] = 2'b10, this field will be set 0x80(128 bytes) automatically. If SMCR [PSIZE] = 2'b11, this field will be set 0x100 (256bytes) automatically. Note: The REA128_ext must be 4 byte aligned, so bit1 and bit0 can't be filled 1 to it. The maximum redundant area of the controller is 472Bytes.

Smart-media ECC Error Status 0 (SM_ECC_ST0)

Register	Address	R/W	Description	Reset Value
SM_ECC_ST0	FMI_BA+0x0D0	R	Smart-Media ECC Error Status 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		F4_ECNT				F4_STAT	
23	22	21	20	19	18	17	16
Reserved		F3_ECNT				F3_STAT	
15	14	13	12	11	10	9	8
Reserved		F2_ECNT				F2_STAT	
7	6	5	4	3	2	1	0
Reserved		F1_ECNT				F1_STAT	

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:26]	F4_ECNT	Error Count of ECC Field 4 This field contains the error counts after ECC correct calculation of Field 4. For this ECC core (BCH algorithm), only when F4_STAT equals to 0x01, the value in this field is meaningful. F4_ECNT means how many errors depending on which ECC is used.
[25:24]	F4_STAT	ECC Status of Field 4 This field contains the ECC correction status (BCH algorithm) of ECC-field 4. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[23:22]	Reserved	Reserved
[21:18]	F3_ECNT	Error Count of ECC Field 3 This field contains the error counts after ECC correct calculation of Field 3. For this ECC core (BCH algorithm), only when F3_STAT equals to 0x01, the value in this field is meaningful. F4_ECNT means how many errors depending on which ECC is used.
[17:16]	F3_STAT	ECC Status of Field 3 This field contains the ECC correction status (BCH algorithm) of ECC-field 3. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[15:14]	Reserved	Reserved
[13:10]	F2_ECNT	Error Count of ECC Field 2 This field contains the error counts after ECC correct calculation of Field 2. For this ECC core (BCH algorithm), only when F2_STAT equals to 0x01, the value in this field is meaningful. F4_ECNT means how many errors depending on which ECC is used.

[9:8]	F2_STAT	ECC Status of Field 2 This field contains the ECC correction status (BCH algorithm) of ECC-field 2. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[7:6]	Reserved	Reserved
[5:2]	F1_ECNT	Error Count of ECC Field 1 This field contains the error counts after ECC correct calculation of Field 1. For this ECC core (BCH algorithm), only when F1_STAT equals to 0x01, the value in this field is meaningful. F1_ECNT means how many errors depending on which ECC is used.
[1:0]	F1_STAT	ECC Status of Field 1 This field contains the ECC correction status (BCH algorithm) of ECC-field 1. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.

Smart-media ECC Error Status 1 (SM_ECC_ST1)

Register	Address	R/W	Description	Reset Value
SM_ECC_ST1	FMI_BA+0x0D4	R	Smart-Media ECC Error Status 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		F8_ECNT				F8_STAT	
23	22	21	20	19	18	17	16
Reserved		F7_ECNT				F7_STAT	
15	14	13	12	11	10	9	8
Reserved		F6_ECNT				F6_STAT	
7	6	5	4	3	2	1	0
Reserved		F5_ECNT				F5_STAT	

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:26]	F8_ECNT	Error Count of ECC Field 8 This field contains the error counts after ECC correct calculation of Field 8. For this ECC core (BCH algorithm), only when F8_STAT equals to 0x01, the value in this field is meaningful. F8_ECNT means how many errors depending on which ECC is used.
[25:24]	F8_STAT	ECC Status of Field 8 This field contains the ECC correction status (BCH algorithm) of ECC-field 8. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[23:22]	Reserved	Reserved
[21:18]	F7_ECNT	Error Count of ECC Field 7 This field contains the error counts after ECC correct calculation of Field 7. For this ECC core (BCH algorithm), only when F7_STAT equals to 0x01, the value in this field is meaningful. F7_ECNT means how many errors depending on which ECC is used.
[17:16]	F7_STAT	ECC Status of Field 7 This field contains the ECC correction status (BCH algorithm) of ECC-field 7. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[15:14]	Reserved	Reserved
[13:10]	F6_ECNT	Error Count of ECC Field 6 This field contains the error counts after ECC correct calculation of Field 6. For this ECC core (BCH algorithm), only when F6_STAT equals to 0x01, the value in this field is meaningful. F6_ECNT means how many errors depending on which ECC is used.

[9:8]	F6_STAT	ECC Status of Field 6 This field contains the ECC correction status (BCH algorithm) of ECC-field 6. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[7:6]	Reserved	Reserved
[5:2]	F5_ECNT	Error Count of ECC Field 5 This field contains the error counts after ECC correct calculation of Field 5. For this ECC core (BCH algorithm), only when F5_STAT equals to 0x01, the value in this field is meaningful. F5_ECNT means how many errors depending on which ECC is used.
[1:0]	F5_STAT	ECC Status of Field 5 This field contains the ECC correction status (BCH algorithm) of ECC-field5. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.

Smart-media ECC Error Status 2 (SM_ECC_ST2)

Register	Address	R/W	Description	Reset Value
SM_ECC_ST2	FMI_BA+0x0D8	R	Smart-Media ECC Error Status 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		F11_ECNT				F11_STAT	
23	22	21	20	19	18	17	16
Reserved		F10_ECNT				F10_STAT	
15	14	13	12	11	10	9	8
Reserved		F9_ECNT				F9_STAT	
7	6	5	4	3	2	1	0
Reserved		F8_ECNT				F8_STAT	

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:26]	F11_ECNT	Error Count of ECC Field 11 This field contains the error counts after ECC correct calculation of Field 11. For this ECC core (BCH algorithm), only when F11_STAT equals to 0x01, the value in this field is meaningful. F11_ECNT means how many errors depending on which ECC is used.
[25:24]	F11_STAT	ECC Status of Field 11 This field contains the ECC correction status (BCH algorithm) of ECC-field 11. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[23:22]	Reserved	Reserved
[21:18]	F10_ECNT	Error Count of ECC Field 10 This field contains the error counts after ECC correct calculation of Field 10. For this ECC core (BCH algorithm), only when F10_STAT equals to 0x01, the value in this field is meaningful. F10_ECNT means how many errors depending on which ECC is used.
[17:16]	F10_STAT	ECC Status of Field 10 This field contains the ECC correction status (BCH algorithm) of ECC-field 10. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[15:14]	Reserved	Reserved
[13:10]	F9_ECNT	Error Count of ECC Field 9 This field contains the error counts after ECC correct calculation of Field 9. For this ECC core (BCH algorithm), only when F9_STAT equals to 0x01, the value in this field is meaningful. F9_ECNT means how many errors depending on which ECC is used.

[9:8]	F9_STAT	ECC Status of Field 9 This field contains the ECC correction status (BCH algorithm) of ECC-field 9. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[7:6]	Reserved	Reserved
[5:2]	F8_ECNT	Error Count of ECC Field 8 This field contains the error counts after ECC correct calculation of Field 8. For this ECC core (BCH algorithm), only when F8_STAT equals to 0x01, the value in this field is meaningful. F8_ECNT means how many errors depending on which ECC is used.
[1:0]	F8_STAT	ECC Status of Field 8 This field contains the ECC correction status (BCH algorithm) of ECC-field 8. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.

Smart-media ECC Error Status 3 (SM_ECC_ST3)

Register	Address	R/W	Description	Reset Value
SM_ECC_ST3	FMI_BA+0x0DC	R	Smart-Media ECC Error Status 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		F15_ECNT				F15_STAT	
23	22	21	20	19	18	17	16
Reserved		F14_ECNT				F14_STAT	
15	14	13	12	11	10	9	8
Reserved		F13_ECNT				F13_STAT	
7	6	5	4	3	2	1	0
Reserved		F12_ECNT				F12_STAT	

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:26]	F15_ECNT	Error Count of ECC Field 15 This field contains the error counts after ECC correct calculation of Field 15. For this ECC core (BCH algorithm), only when F15_STAT equals to 0x01, the value in this field is meaningful. F15_ECNT means how many errors depending on which ECC is used.
[25:24]	F15_STAT	ECC Status of Field 15 This field contains the ECC correction status (BCH algorithm) of ECC-field 15. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[23:22]	Reserved	Reserved
[21:18]	F14_ECNT	Error Count of ECC Field 14 This field contains the error counts after ECC correct calculation of Field 14. For this ECC core (BCH algorithm), only when F14_STAT equals to 0x01, the value in this field is meaningful. F14_ECNT means how many errors depending on which ECC is used.
[17:16]	F14_STAT	ECC Status of Field 14 This field contains the ECC correction status (BCH algorithm) of ECC-field 14. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[15:14]	Reserved	Reserved
[13:10]	F13_ECNT	Error Count of ECC Field 13 This field contains the error counts after ECC correct calculation of Field 13. For this ECC core (BCH algorithm), only when F13_STAT equals to 0x01, the value in this field is meaningful. F13_ECNT means how many errors depending on which ECC is used.

[9:8]	F13_STAT	ECC Status of Field 13 This field contains the ECC correction status (BCH algorithm) of ECC-field 13. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.
[7:6]	Reserved	Reserved
[5:2]	F12_ECNT	Error Count of ECC Field 12 This field contains the error counts after ECC correct calculation of Field 12. For this ECC core (BCH algorithm), only when F12_STAT equals to 0x01, the value in this field is meaningful. F12_ECNT means how many errors depending on which ECC is used.
[1:0]	F12_STAT	ECC Status of Field 12 This field contains the ECC correction status (BCH algorithm) of ECC-field 12. 00 = No error. 01 = Correctable error. 10 = Uncorrectable error.

BCH Error Address 0 (BCH_ERR_ADDR0)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR0	FMI_BA+0x100	R	BCH Error Byte Address0.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_1	
23	22	21	20	19	18	17	16
E_ADDR_FF_1							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_0	
7	6	5	4	3	2	1	0
E_ADDR_FF_0							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_1	ECC Error Address First Field of Error 1 This field contains a 10-bit ECC error address 1 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF1 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_0	ECC Error Address First Field of Error 0 This field contains a 10-bit ECC error address 0 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF0 for correcting this error.

BCH Error Address 1 (BCH_ERR_ADDR1)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR1	FMI_BA+0x104	R	BCH Error Byte Address1.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_3	
23	22	21	20	19	18	17	16
E_ADDR_FF_3							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_2	
7	6	5	4	3	2	1	0
E_ADDR_FF_2							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_3	ECC Error Address First Field of Error 3 This field contains a 10-bit ECC error address 3 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF3 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_2	ECC Error Address First Field of Error 2 This field contains a 10-bit ECC error address 2 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF2 for correcting this error.

BCH Error Address 2 (BCH_ERR_ADDR2)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR2	FMI_BA+0x108	R	BCH Error Byte Address2.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_5	
23	22	21	20	19	18	17	16
E_ADDR_FF_5							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_4	
7	6	5	4	3	2	1	0
E_ADDR_FF_4							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_5	ECC Error Address First Field of Error 5 This field contains a 10-bit ECC error address 5 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF5 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_4	ECC Error Address First Field of Error 4 This field contains a 10-bit ECC error address 4 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF4 for correcting this error.

BCH Error Address 3 (BCH_ERR_ADDR3)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR3	FMI_BA+0x10C	R	BCH Error Byte Address3.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_7	
23	22	21	20	19	18	17	16
E_ADDR_FF_7							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_6	
7	6	5	4	3	2	1	0
E_ADDR_FF_6							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_7	ECC Error Address First Field of Error 7 This field contains a 10-bit ECC error address 7 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF7 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_6	ECC Error Address First Field of Error 6 This field contains a 10-bit ECC error address 6 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF6 for correcting this error.

BCH Error Address 4 (BCH_ERR_ADDR4)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR4	FMI_BA+0x110	R	BCH Error Byte Address4.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_9	
23	22	21	20	19	18	17	16
E_ADDR_FF_9							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_8	
7	6	5	4	3	2	1	0
E_ADDR_FF_8							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_9	ECC Error Address First Field of Error 9 This field contains a 10-bit ECC error address 9 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF9 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_8	ECC Error Address First Field of Error 8 This field contains a 10-bit ECC error address 8 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF8 for correcting this error.

BCH Error Address 5 (BCH_ERR_ADDR5)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR5	FMI_BA+0x114	R	BCH Error Byte Address5.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_11	
23	22	21	20	19	18	17	16
E_ADDR_FF_11							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_10	
7	6	5	4	3	2	1	0
E_ADDR_FF_10							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_11	ECC Error Address First Field of Error 11 This field contains a 10-bit ECC error address 11 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF11 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_10	ECC Error Address First Field of Error 10 This field contains a 10-bit ECC error address 10 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF10 for correcting this error.

BCH Error Address 6 (BCH_ERR_ADDR6)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR6	FMI_BA+0x118	R	BCH Error Byte Address6.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						E_ADDR_FF_13	
23	22	21	20	19	18	17	16
E_ADDR_FF_13							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_12	
7	6	5	4	3	2	1	0
E_ADDR_FF_12							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	E_ADDR_FF_13	ECC Error Address First Field of Error 13 This field contains a 10-bit ECC error address 13 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF13 for correcting this error.
[15:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_12	ECC Error Address First Field of Error 12 This field contains a 10-bit ECC error address 12 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF12 for correcting this error.

BCH Error Address 7 (BCH_ERR_ADDR7)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR7	FMI_BA+0x11C	R	BCH Error Byte Address7.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						E_ADDR_FF_14	
7	6	5	4	3	2	1	0
E_ADDR_FF_14							

Bits	Descriptions	
[31:10]	Reserved	Reserved
[9:0]	E_ADDR_FF_14	ECC Error Address First Field of Error 14 This field contains a 10-bit ECC error address 14 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF14 for correcting this error.

BCH ECC Error Data (BCH ECC_DATA0)

Register	Address	R/W	Description	Reset Value
BCH_ECC_DATA0	FMI_BA+0x160	R	BCH ECC Error Data 0	0x8080_8080

31	30	29	28	27	26	25	24
E_DATA_FF_3							
23	22	21	20	19	18	17	16
E_DATA_FF_2							
15	14	13	12	11	10	9	8
E_DATA_FF_1							
7	6	5	4	3	2	1	0
E_DATA_FF_0							

Bits	Descriptions	
[31:24]	E_DATA_FF_3	ECC Error Data Of First Field 3 This field contains an 8-bit BCH ECC error data 3 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_3; the result will be the correct data.
[23:16]	E_DATA_FF_2	ECC Error Data Of First Field 2 This field contains an 8-bit BCH ECC error data 2 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_2; the result will be the correct data.
[15:8]	E_DATA_FF_1	ECC Error Data Of First Field 1 This field contains an 8-bit BCH ECC error data 1 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_1; the result will be the correct data.
[7:0]	E_DATA_FF_0	ECC Error Data Of First Field 0 This field contains an 8-bit BCH ECC error data 0 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_0; the result will be the correct data.

BCH ECC Error Data (BCH ECC DATA1)

Register	Address	R/W	Description	Reset Value
BCH_ECC_DATA1	FMI_BA+0x164	R	BCH ECC Error Data 1	0x8080_8080

31	30	29	28	27	26	25	24
E_DATA_FF_7							
23	22	21	20	19	18	17	16
E_DATA_FF_6							
15	14	13	12	11	10	9	8
E_DATA_FF_5							
7	6	5	4	3	2	1	0
E_DATA_FF_4							

Bits	Descriptions	
[31:24]	E_DATA_FF_7	ECC Error Data Of First Field 7 This field contains an 8-bit BCH ECC error data 7 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_7; the result will be the correct data.
[23:16]	E_DATA_FF_6	ECC Error Data Of First Field 6 This field contains an 8-bit BCH ECC error data 6 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_6; the result will be the correct data.
[15:8]	E_DATA_FF_5	ECC Error Data Of First Field 5 This field contains an 8-bit BCH ECC error data 5 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_5; the result will be the correct data.
[7:0]	E_DATA_FF_4	ECC Error Data Of First Field 4 This field contains an 8-bit BCH ECC error data 4 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_4; the result will be the correct data.

BCH ECC Error Data (BCH ECC DATA2)

Register	Address	R/W	Description	Reset Value
BCH_ECC_DATA2	FMI_BA+0x168	R	BCH ECC Error Data 2	0x8080_8080

31	30	29	28	27	26	25	24
E_DATA_FF_11							
23	22	21	20	19	18	17	16
E_DATA_FF_10							
15	14	13	12	11	10	9	8
E_DATA_FF_9							
7	6	5	4	3	2	1	0
E_DATA_FF_8							

Bits	Descriptions	
[31:24]	E_DATA_FF_11	ECC Error Data Of First Field 11 This field contains an 8-bit BCH ECC error data 11 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_11; the result will be the correct data.
[23:16]	E_DATA_FF_10	ECC Error Data Of First Field 10 This field contains an 8-bit BCH ECC error data 10 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_10; the result will be the correct data.
[15:8]	E_DATA_FF_9	ECC Error Data Of First Field 9 This field contains an 8-bit BCH ECC error data 9 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_9; the result will be the correct data.
[7:0]	E_DATA_FF_8	ECC Error Data Of First Field 8 This field contains an 8-bit BCH ECC error data 8 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_8; the result will be the correct data.

BCH ECC Error Data (BCH ECC DATA3)

Register	Address	R/W	Description	Reset Value
BCH_ECC_DATA3	FMI_BA+0x16C	R	BCH ECC Error Data 3	0x0080_8080

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
E_DATA_FF_14							
15	14	13	12	11	10	9	8
E_DATA_FF_13							
7	6	5	4	3	2	1	0
E_DATA_FF_12							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	E_DATA_FF_14	ECC Error Data Of First Field 14 This field contains an 8-bit BCH ECC error data 14 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_14; the result will be the correct data.
[15:8]	E_DATA_FF_13	ECC Error Data Of First Field 13 This field contains an 8-bit BCH ECC error data 13 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_13; the result will be the correct data.
[7:0]	E_DATA_FF_12	ECC Error Data Of First Field 12 This field contains an 8-bit BCH ECC error data 12 of first field. If it is a correctable error, software can read out the error data in this field and doing bitwise XOR with received data locating at address E_ADDR_FF_12; the result will be the correct data.

Smart-media Redundant Area Register (SMRA)

Register	Address	R/W	Description	Reset Value
SM_RA0	FMI_BA+0x200	R/W	Smart-Media Redundant Area Register	N/A
...	...			
SM_RA117	FMI_BA+0x3D4			

31	30	29	28	27	26	25	24
SM_RA							
23	22	21	20	19	18	17	16
SM_RA							
15	14	13	12	11	10	9	8
SM_RA							
7	6	5	4	3	2	1	0
SM_RA							

Bits	Descriptions	
[31:0]	SM_RA	Smart-Media Redundant Area This field is parity data buffer. Note: The SRMA reset value is undefined.

5.12 USB 2.0 Device Controller (USBD)

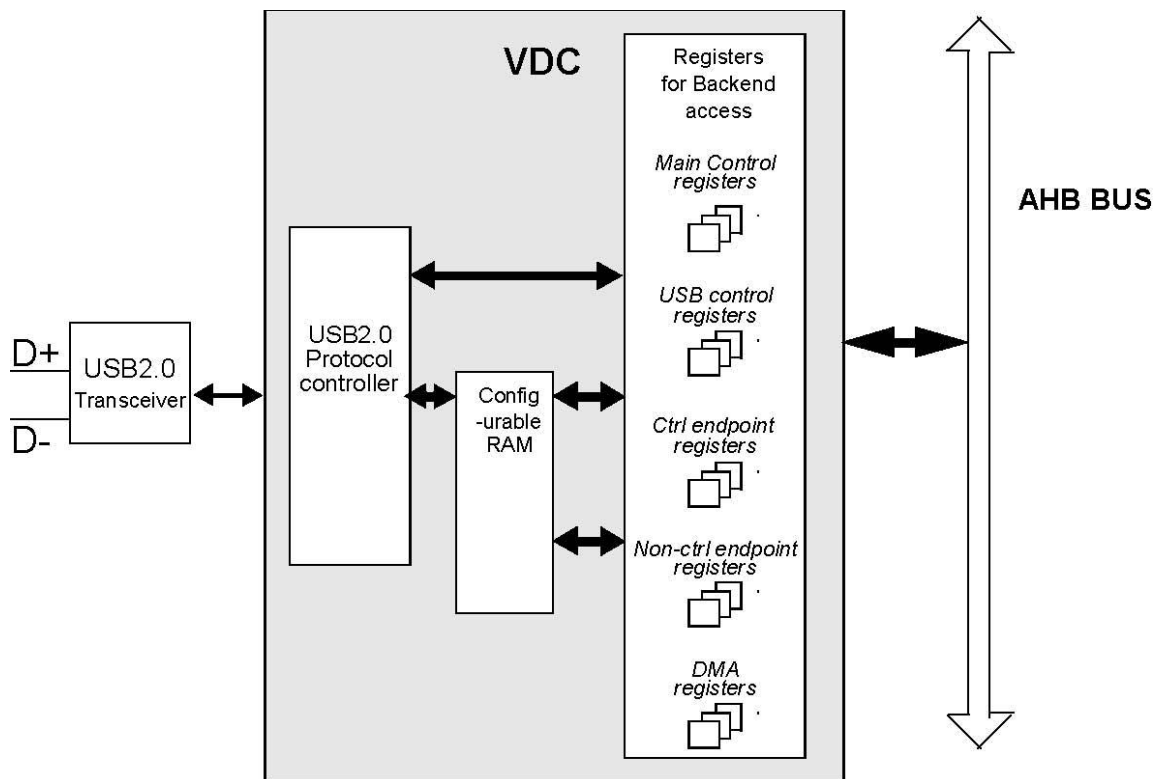
5.12.1 Overview

The USB device controller interfaces the AHB bus and the UTMI bus. The USB controller contains both the AHB master interface and AHB slave interface. CPU programs the USB controller registers through the AHB slave interface. For IN or OUT transfer, the USB device controller needs to write data to memory or read data from memory through the AHB master interface. The USB device controller is compliant with USB 2.0 specification and it contains four configurable endpoints in addition to control endpoint. These endpoints could be configured to BULK, INTERRUPT or ISO. The USB device controller has a built-in DMA to relieve the load of CPU.

5.12.2 Features

- USB Specification version 2.0 compliant.
- Interfaces between USB 2.0 bus and the AHB bus.
- Supports 16-bit UTMI Interface to USB2.0 Transceiver.
- Support direct register addressing for all registers from the AHB bus.
- Software control for device remote-wakeup.
- AHB bus facilitates connection to common micro controllers and DMA controllers.
- Supports 4 configurable endpoints in addition to Control Endpoint
- Each of these endpoints can be Isochronous, Bulk or Interrupt and they can be either of IN or OUT direction.
- Three different modes of operation of an in-endpoint (Auto validation mode, manual validation mode, Fly mode.)
- DP RAM is used as end point buffer.
- DMA operation is carried out by AHB master
- Supports Endpoint Maximum Packet Size up to 1024 bytes.

5.12.3 Internal Block Diagram



5.12.4 USB Device Register Group Summary

Register Groups	Description
Main Control Registers	These set of registers control the global enable of interrupts and maintain the status of the interrupts
USB Control Registers	These set of registers control the USB related events to/from the USB host and hold the status of the USB events.
Control Endpoint Registers	These set of registers direct the control endpoint in handling the USB requests from the host and hold the status information of the transactions.
Non control Endpoint Registers	These set of registers configure, control and exhibit the status of the non-control endpoints' operation
DMA Registers	These registers are responsible for the DMA related operations

5.12.5 USB Device Control Registers Map

Register	Address	R/W	Description	Reset Value
USBD_BA = B100_8000				
IRQ	USBD_BA+0x00	R	Interrupt Register	0x0000_0000
Reserved	USBD_BA+0x04			
IRQ_ENB_L	USBD_BA+0x08	R/W	Interrupt Enable Low Register	0x0000_0001
Reserved	USBD_BA+0x0C			
USB_IRQ_STAT	USBD_BA+0x10	R/W	USB Interrupt Status register	0x0000_0000
USB_IRQ_ENB	USBD_BA+0x14	R/W	USB Interrupt Enable register	0x0000_0040
USB_OPER	USBD_BA+0x18	R/W	USB operational register	0x0000_0002
USB_FRAME_CNT	USBD_BA+0x1C	R	USB frame count register	0x0000_0000
USB_ADDR	USBD_BA+0x20	R/W	USB address register	0x0000_0000
USB_TEST	USBD_BA+0x24	R/W	USB test mode register	0x0000_0000
CEP_DATA_BUF	USBD_BA+0x28	R/W	Control-ep Data Buffer	0x0000_0000
CEP_CTRL_STAT	USBD_BA+0x2C	R/W	Control-ep Control and Status	0x0000_0000
CEP_IRQ_ENB	USBD_BA+0x30	R/W	Control-ep Interrupt Enable	0x0000_0000
CEP_IRQ_STAT	USBD_BA+0x34	R/W	Control-ep Interrupt Status	0x0000_1800
IN_TRNSFR_CNT	USBD_BA+0x38	R/W	In-transfer data count	0x0000_0000
OUT_TRNSFR_CNT	USBD_BA+0x3C	R	Out-transfer data count	0x0000_0000
CEP_CNT	USBD_BA+0x40	R	Control-ep data count	0x0000_0000
SETUP1_0	USBD_BA+0x44	R	Setupbyte1 & byte0	0x0000_0000
SETUP3_2	USBD_BA+0x48	R	Setupbyte3 & byte2	0x0000_0000
SETUP5_4	USBD_BA+0x4C	R	Setupbyte5 & byte4	0x0000_0000
SETUP7_6	USBD_BA+0x50	R	Setupbyte7 & byte6	0x0000_0000
CEP_START_ADDR	USBD_BA+0x54	R/W	Control EP' s RAM start address	0x0000_0000
CEP_END_ADDR	USBD_BA+0x58	R/W	Control EP' s RAM end address	0x0000_0000
DMA_CTRL_STS	USBD_BA+0x5C	R/W	DMA control and status register	0x0000_0000
DMA_CNT	USBD_BA+0x60	R/W	DMA count register	0x0000_0000
EPA_DATA_BUF	USBD_BA+0x64	R/W	Endpoint Adata register	0x0000_0000
EPA_IRQ_STAT	USBD_BA+0x68	R/W	Endpoint A Interrupt status register	0x0000_0003
EPA_IRQ_ENB	USBD_BA+0x6C	R/W	Endpoint A Interrupt enable register	0x0000_0000
EPA_DATA_CNT	USBD_BA+0x70	R	Data count available in endpoint Abuffer	0x0000_0000
EPA_RSP_SC	USBD_BA+0x74	R/W	Endpoint Aresponse register set/clear	0x0000_0000
EPA_MPS	USBD_BA+0x78	R/W	Endpoint Amaximum packet size register	0x0000_0000

EPA_CNT	USBD_BA+0x7C	R/W	Endpoint Atransfer count register	0x0000_0000
EPA_CFG	USBD_BA+0x80	R/W	Endpoint Aconfiguration register	0x0000_0012
EPA_START_ADDR	USBD_BA+0x84	R/W	EndpointA' s RAM start address	0x0000_0000
EPA_END_ADDR	USBD_BA+0x88	R/W	EndpointA' s RAM end address	0x0000_0000
EPB_DATA_BUF	USBD_BA+0x8C	R/W	EndpointB data register	0x0000_0000
EPB_IRQ_STAT	USBD_BA+0x90	R/W	EndpointB Interrupt status register	0x0000_0003
EPB_IRQ_ENB	USBD_BA+0x94	R/W	EndpointB Interrupt enable register	0x0000_0000
EPB_DATA_CNT	USBD_BA+0x98	R	Data count available in endpointB buffer	0x0000_0000
EPB_RSP_SC	USBD_BA+0x9C	R/W	EndpointB response register set/clear	0x0000_0000
EPB_MPS	USBD_BA+0xA0	R/W	EndpointB maximum packet size register	0x0000_0000
EPB_TRF_CNT	USBD_BA+0xA4	R/W	EndpointB transfer count register	0x0000_0000
EPB_CFG	USBD_BA+0xA8	R/W	EndpointB configuration register	0x0000_0022
EPB_START_ADDR	USBD_BA+0xAC	R/W	EndpointB' s RAM start address	0x0000_0000
EPB_END_ADDR	USBD_BA+0xB0	R/W	EndpointB' s RAM end address	0x0000_0000
EPC_DATA_BUF	USBD_BA+0xB4	R/W	EndpointC data register	0x0000_0000
EPC_IRQ_STAT	USBD_BA+0xB8	R/W	EndpointC Interrupt status register	0x0000_0003
EPC_IRQ_ENB	USBD_BA+0xBC	R/W	EndpointC Interrupt enable register	0x0000_0000
EPC_DATA_CNT	USBD_BA+0xC0	R	Data count available in endpointC buffer	0x0000_0000
EPC_RSP_SC	USBD_BA+0xC4	R/W	EndpointC response register set/clear	0x0000_0000
EPC_MPS	USBD_BA+0xC8	R/W	EndpointC maximum packet size register	0x0000_0000
EPC_TRF_CNT	USBD_BA+0xCC	R/W	EndpointC transfer count register	0x0000_0000
EPC_CFG	USBD_BA+0xD0	R/W	EndpointC configuration register	0x0000_0032
EPC_START_ADDR	USBD_BA+0xD4	R/W	EndpointC' s RAM start address	0x0000_0000
EPC_END_ADDR	USBD_BA+0xD8	R/W	EndpointC' s RAM end address	0x0000_0000
EPD_DATA_BUF	USBD_BA+0xDC	R/W	Endpoint Ddata register	0x0000_0000
EPD_IRQ_STAT	USBD_BA+0xE0	R/W	Endpoint D Interrupt status register	0x0000_0003
EPD_IRQ_ENB	USBD_BA+0xE4	R/W	Endpoint D Interrupt enable register	0x0000_0000
EPD_DATA_CNT	USBD_BA+0xE8	R	Data count available in endpoint Dbuffer	0x0000_0000
EPD_RSP_SC	USBD_BA+0xEC	R/W	Endpoint Dresponse register set/clear	0x0000_0000
EPD_MPS	USBD_BA+0xF0	R/W	Endpoint Dmaximum packet size register	0x0000_0000
EPD_CNT	USBD_BA+0xF4	R/W	Endpoint Dtransfer count register	0x0000_0000
EPD_CFG	USBD_BA+0xF8	R/W	Endpoint Dconfiguration register	0x0000_0042
EPD_START_ADDR	USBD_BA+0xFC	R/W	EndpointD' s RAM start address	0x0000_0000
EPD_END_ADDR	USBD_BA+0x100	R/W	EndpointD' s RAM end address	0x0000_0000

USB MEM TEST	USBD_BA+0x154	R/W	USB memory test	0x0000_0000
USB HEAD WORD0	USBD_BA+0x158	R/W	USB header word0	0x0000_0000
USB HEAD WORD1	USBD_BA+0x15C	R/W	USB header word1	0x0000_0000
USB HEAD WORD2	USBD_BA+0x160	R/W	USB header word2	0x0000_0000
EPA HEAD CNT	USBD_BA+0x164	R/W	Endpoint A header count	0x0000_0000
EPB HEAD CNT	USBD_BA+0x168	R/W	Endpoint B header count	0x0000_0000
EPC HEAD CNT	USBD_BA+0x16C	R/W	Endpoint C header count	0x0000_0000
EPD HEAD CNT	USBD_BA+0x170	R/W	Endpoint D header count	0x0000_0000
AHB_DMA_ADDR	USBD_BA+0x700	R/W	AHB bus DMA address	0x0000_0000
USB_PHY_CTL	USBD_BA+0x704	R/W	USB PHY control register	0x0000_0420

5.12.6 Registers Description

Interrupt Register (IRQ)

Register	Address	R/W	Description	Default Value
IRQ	USBD_BA+0x000	R	Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		EPD_INT	EPC_INT	EPB_INT	EPA_INT	CEP_INT	USB_INT

Bits	Descriptions	
[5]	EPD_INT	This bit conveys the interrupt for Endpoints D. When set, the corresponding Endpoint D' s interrupt status register should be read to determine the cause of the interrupt.
[4]	EPC_INT	This bit conveys the interrupt for Endpoints C. When set, the corresponding Endpoint C's interrupt status register should be read to determine the cause of the interrupt.
[3]	EPB_INT	This bit conveys the interrupt for Endpoints B. When set, the corresponding Endpoint B's interrupt status register should be read to determine the cause of the interrupt.
[2]	EPA_INT	This bit conveys the interrupt for Endpoints A. When set, the corresponding Endpoint A's interrupt status register should be read to determine the cause of the interrupt.
[1]	CEP_INT	Control Endpoint Interrupt. This bit conveys the interrupt status for control endpoint. When set, Control-ep' s interrupt status register should be read to determine the cause of the interrupt.
[0]	USB_INT	USB Interrupt. This bit conveys the interrupt status for USB specific events endpoint. When set, USB interrupt status register should be read to determine the cause of the interrupt.

Interrupt Enable Low Register (IRQ_ENB_L)

Register	Address	R/W	Description	Default Value
IRQ_ENB_L	USBD_BA+0x008	R/W	Interrupt Enable Low Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		EPD_IE	EPC_IE	EPB_IE	EPA_IE	CEP_IE	USB_IE

Bits	Descriptions	
[5]	EPD_IE	Interrupt Enable for Endpoint D. When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint D
[4]	EPC_IE	Interrupt Enable for Endpoint C. When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint C
[3]	EPB_IE	Interrupt Enable for Endpoint B. When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint B
[2]	EPA_IE	Interrupt Enable for Endpoint A. When set, this bit enables a local interrupt to be generated when an interrupt is pending for the endpoint A.
[1]	CEP_IE	Control Endpoint Interrupt Enable. When set, this bit enables a local interrupt to be generated when an interrupt is pending for the control endpoint.
[0]	USB_IE	USB Interrupt Enable. When set, this bit enables a local interrupt to be generated when a USB event occurs on the bus.

USB Interrupt Status Register (USB_IRQ_STAT)

Register	Address	R/W	Description	Default Value
USB_IRQ_STAT	USBD_BA+0x010	R/W	USB Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							VBUS_IS
7	6	5	4	3	2	1	0
RUM_NOCLK_IS	TCLKOK_IS	DMACOM_IS	HISPD_IS	SUS_IS	RUM_IS	RST_IS	SOF_IS

Bits	Descriptions	
[8]	VBUS_IS	VBUS Interrupt This bit is set when the Vbus is attached/de-attached. Writing '1' clears this bit.
[7]	RUM_NOCLK	RESUME from no clock This bit is used to detect the DP and DM transition when USB clock is off
[6]	TCLKOK_IS	Usable Clock Interrupt. This bit is set when usable clock is available from the transceiver. Writing '1' clears this bit.
[5]	DMACOM_IS	DMA Completion Interrupt. This bit is set when the DMA transfer is over. Writing '1' clears this bit.
[4]	HISPD_IS	High Speed Settle. This bit is set when the valid high-speed reset protocol is over and the device has settled in high-speed. Writing '1' clears this bit.
[3]	SUS_IS	Suspend Request. This bit is set as default and it has to be cleared by writing '1' before the USB reset. This bit is also set when a USB Suspend request is detected from the host. Writing '1' clears this bit.
[2]	RUM_IS	Resume. When set, this bit indicates that a device resume has occurred. Writing a '1' clears this bit.
[1]	RST_IS	Reset Status. When set, this bit indicates that whether the USB root port reset is end. Writing a '1' clears this bit.
[0]	SOF_IS	SOF. This bit indicates when a start-of-frame packet has been received. Writing a '1' clears this bit.

USB Interrupt Enable Register (USB_IRQ_ENB)

Register	Address	R/W	Description	Default Value
USB_IRQ_ENB	USBD_BA+0x014	R/W	USB Interrupt Enable Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							VBUS_IE
7	6	5	4	3	2	1	0
RUM_NOCLK_IE	TCLKOK_IE	DMACOM_IE	HISPD_IE	SUS_IE	RUM_IE	RST_IE	SOF_IE

Bits	Descriptions	
[8]	VBUS_IE	VBUS Detected Interrupt Enable. This bit enables the Vbus detected interrupt.
[7]	RUM_NOCLK_IE	Usable Clock Interrupt Enable. This bit enables the DP and DM transition interrupt.
[6]	TCLKOK_IE	Usable Clock Interrupt Enable. This bit enables the usable clock interrupt.
[5]	DMACOM_IE	DMA Completion Interrupt Enable. This bit enables the DMA completion interrupt.
[4]	HISPD_IE	High Speed Settle Interrupt Enable. This bit enables the high-speed settle interrupt.
[3]	SUS_IE	Suspend Request Interrupt Enable. This bit enables the Suspend interrupt.
[2]	RUM_IE	Resume Interrupt Enable. This bit enables the Resume interrupt.
[1]	RST_IE	Reset Status Interrupt Enable. This bit enables the USB-Reset interrupt.
[0]	SOF_IE	SOF Interrupt Enable. This bit enables the SOF interrupt.

USB Operational Register (USB_OPER)

Register	Address	R/W	Description	Default Value
USB_OPER	USBD_BA+0x018	R/W	Usb Operational Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CUR_SPD	SET_HISPD	GEN_RUM

Bits	Descriptions	
[2]	CUR_SPD	USB Current Speed. When set, this bit indicates that the DEVICE CONTROLLER has settled in High Speed and a zero indicates that the device has settled in Full Speed.(READ ONLY)
[1]	SET_HISPD	USB High Speed. When set to one, this bit indicates the DEVICE CONTROLLER to initiate a chirp-sequence during reset protocol, if it set to zero, it indicates the DEVICE CONTROLLER to suppress the chirp-sequence during reset protocol, thereby allowing the DEVICE CONTROLLER to settle in full-speed, even though it is connected to a USB2.0 Host.
[0]	GEN_RUM	Generate Resume. Writing a 1 to this bit causes a Resume sequence to be initiated to the host if device remote wakeup is enabled. This bit is self-clearing.

USB Frame Count Register (USB_FRAME_CNT)

Register	Address	R/W	Description	Default Value
USB_FRAME_CNT	USBD_BA+0x01C	R	USB Frame Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FRAME_CNT					
7	6	5	4	3	2	1	0
FRAME_CNT					MFRAME_CNT		

Bits	Descriptions	
[13:3]	FRAME_CNT	FRAME COUNTER. This field contains the frame count from the most recent start-of-frame packet.
[2:0]	MFRAME_CNT	MICRO FRAME COUNTER. This field contains the micro-frame number for the frame number in the frame counter field.

USB Address Register (USB_ADDR)

Register	Address	R/W	Description	Default Value
USB_ADDR	USBD_BA+0x020	R/W	USB Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	ADDR						

Bits	Descriptions	
[6:0]	ADDR	This field contains the current USB address of the device. This field is cleared when a root port reset is detected.

USB Test Mode Register (USB_TEST)

Register	Address	R/W	Description	Default Value
USB_TEST	USBD_BA+0x024	R/W	USB Test Mode Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					TESTMODE		

Bits	Descriptions	
[2:0]	TESTMODE	Test Mode Select.
		Value Test
		000 Normal Operation
		001 Test_J
		010 Test_K
		011 Test_SE0_NAK
		100 Test_Packet
		101 Test_Force_Enable
		110 Reserved
		111 Reserved
		This field is cleared when root port reset is detected.

Control-ep Data Buffer (CEP_DATA_BUF)

Register	Address	R/W	Description	Default Value
CEP_DATA_BUF	USB_D_BA+0x028	RW	Control-ep Data Buffer	0x0000_0000

31	30	29	28	27	26	25	24
DATA_BUF							
23	22	21	20	19	18	17	16
DATA_BUF							
15	14	13	12	11	10	9	8
DATA_BUF							
7	6	5	4	3	2	1	0
DATA_BUF							

Bits	Descriptions	
[31:0]	DATA_BUF	Control-ep Data Buffer. the data port for the buffer transaction (read or write).

Control-ep Control and Status (CEP_CTRL_STAT)

Register	Address	R/W	Description	Default Value
CEP_CTRL_STAT	USB_D_BA+0x02C	RW	Control-ep Control and Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				FLUSH	ZEROLEN	STLALL	NAK_CLEAR

Bits	Descriptions	
[3]	FLUSH	CEP-FLUSH Bit. Writing 1 to this bit cause the packet buffer and its corresponding CEP_AVL_CNT register to be cleared. This bit is self-cleaning.
[2]	ZEROLEN	ZEROLEN Bit. This bit is valid for auto validation mode only. When this bit is set, DEVICE CONTROLLER can send a zero length packet to the host during Data stage to an IN token. This bit gets cleared once the zero length data packet is sent. So, the local CPU need not write again to clear this bit.
[1]	STLALL	STALL. This bit is a read/write bit. When this stall bit is set, the control endpoint sends a stall handshake in response to any in or out token thereafter. This is typically used for response to invalid/unsupported requests. When this bit is being set the NAK clear bit has to be cleared at the same time since the NAK clear bit has highest priority than STALL. It is automatically cleared on receipt of a next setup-token. So, the local CPU need not write again to clear this bit. NOTE: ONLY when cpu write data[1:0] is 2' b10 or 2' b00, this bit can be updated.
[0]	NAK_CLEAR	NAK_CLEAR. This is a read/write bit. This bit plays a crucial role in any control transfer. This bit is set to one by the DEVICE CONTROLLER, whenever a setup token is received. The local CPU can take its own time to finish off any house-keeping work based on the request and then clear this bit. Unless the bit is being cleared by the local CPU by writing zero, the DEVICE CONTROLLER will be responding with NAKs for the subsequent status phase. This mechanism holds the host from moving to the next request, until the local CPU is also ready to process the next request. NOTE: ONLY when cpu write data[1:0] is 2'b10 or 2'b00, this bit can be updated.

Control Endpoint Interrupt Enable (CEP_IRQ_ENABLE)

Register	Address	R/W	Description	Default Value
CEP_IRQ_ENABLE	USBD_BA+0x030	R/W	Control Endpoint Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		O_SHORT_IE	EMPTY_IE	FULL_IE	STACOM_IE	ERR_IE	STALL_IE
7	6	5	4	3	2	1	0
NAK_IE	DATA_RxED_IE	DATA_TxED_IE	PING_IE	IN_TK_IE	OUT_TK_IE	SETUP_PK_I E	SETUP_TK_IE

Bits	Descriptions	
[13]	O_SHORT_IE	OUT SHORT packet Interrupt. This bit enables the out short packet interrupt.
[12]	EMPTY_IE	Buffer Empty Interrupt. This bit enables the buffer empty interrupt.
[11]	FULL_IE	Buffer Full Interrupt. This bit enables the buffer full interrupt.
[10]	STACOM_IE	Status Completion Interrupt. This bit enables the Status Completion interrupt.
[9]	ERR_IE	USB Error Interrupt. This bit enables the USB Error interrupt.
[8]	STALL_IE	STALL Sent Interrupt. This bit enables the STALL sent interrupt
[7]	NAK_IE	NAK Sent Interrupt. This bit enables the NAK sent interrupt.
[6]	DATA_RxED_IE	Data Packet Received Interrupt. This bit enables the data received interrupt.
[5]	DATA_TxED_IE	Data Packet Transmitted Interrupt. This bit enables the data packet transmitted interrupt.
[4]	PING_IE	Ping Token Interrupt. This bit enables the ping token interrupt.
[3]	IN_TK_IE	In Token Interrupt. This bit enables the in token interrupt

[2]	OUT_TK_IE	Out Token Interrupt. This bit enables the out token interrupt.
[1]	SETUP_PK_IE	Setup Packet Interrupt. This bit enables the setup packet interrupt.
[0]	SETUP_TK_IE	Setup Token Interrupt Enable. This bit enables the setup token interrupt.

Control-ep Interrupt Status (CEP_IRQ_STAT)

Register	Address	R/W	Description	Default Value
CEP_IRQ_STAT	USBD_BA+0x034	R/W	Control-ep Interrupt Status	0x0000_1800

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		O_SHORT_IS	EMPTY_IS	FULL_IS	STACOM_IS	ERR_IS	STALL_IS
7	6	5	4	3	2	1	0
NAK_IS	DATA_RxED_IS	DATA_TxED_IS	PING_IS	IN_TK_IS	OUT_TK_IS	SETUP_PK_IS	SETUP_TK_IS

Bits	Descriptions	
[13]	O_SHORT_IS	OUT SHORT packet Interrupt. This bit is set when the out short packet is received.(READ ONLY)
[12]	EMPTY_IS	Buffer Empty Interrupt. This bit is set when the control-ednpt buffer is empty.(READ ONLY)
[11]	FULL_IS	Buffer Full Interrupt. This bit is set when the controlendpt buffer is full.(READ ONLY)
[10]	STACOM_IS	Status Completion Interrupt. This bit is set when the status stage of a USB transaction has completed successfully. Write "1" clear.
[9]	ERR_IS	USB Error Interrupt. This bit is set when an error had occurred during the transaction. Write "1" clear.
[8]	STALL_IS	STALL Sent Interrupt. This bit is set when a stall-token is sent in response to an in/out token. Write "1" clear.
[7]	NAK_IS	NAK Sent Interrupt. This bit is set when a nak-token is sent in response to an in/out token. Write "1" clear.
[6]	DATA_RxED_IS	Data Packet Received Interrupt. This bit is set when a data packet is successfully received from the host for an out token and an ack is sent to the host. Write "1" clear.
[5]	DATA_TxED_IS	Data Packet Transmitted Interrupt. This bit is set when a data packet is successfully transmitted to the host in response to an in-token and an ack-token is received for the same. Write "1" clear.

[4]	PING_IS	Ping Token Interrupt. This bit is set when the control endpoint receives a ping token from the host. Write “1” clear.
[3]	IN_TK_IS	In Token Interrupt. This bit is set when the control endpoint receives an in token from the host. Write “1” clear.
[2]	OUT_TK_IS	Out Token Interrupt. This bit is set when the control endpoint receives a out token from the host. Write “1” clear.
[1]	SETUP_PK_IS	Setup Packet Interrupt. This bit is set when a setup packet has been received from the host. This bit must be cleared (by writing a 1) before the next setup packet can be received. If the bit is not cleared, then the successive setup packets will be overwritten in the setup packet buffer. Write “1” clear.
[0]	SETUP_TK_IS	Setup Token Interrupt. This bit indicates when a setup token is received. Writing a 1 clears this status bit. Write “1” clear.

In-transfer data count (IN_TRF_CNT)

Register	Address	R/W	Description	Default Value
IN_TRF_CNT	USBD_BA+0x038	R/W	In-transfer data count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
IN_TRF_CNT							

Bits	Descriptions	
[7:0]	IN_TRF_CNT	In-transfer data count. There is no mode selection for the control endpoint (but it operates like manual mode). The local-CPU has to fill the control-endpoint buffer with the data to be sent for an in-token and to write the count of bytes in this register. When zero is written into this field, a zero length packet is sent to the host. When the count written in the register is more than the MPS, the data sent will be of only MPS.

Out-transfer data count (OUT_TRF_CNT)

Register	Address	R/W	Description	Default Value
OUT_TRF_CNT	USBD_BA+0x03C	R	Out-transfer data count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
OUT_TRF_CNT							
7	6	5	4	3	2	1	0
OUT_TRF_CNT							

Bits	Descriptions	
[15:0]	OUT_TRF_CNT	Out-Transfer Data Count. The DEVICE CONTROLLER maintains the count of the data received in case of an out transfer, during the control transfer.

Control-ep data count (CEP_CNT)

Register	Address	R/W	Description	Default Value
CEP_CNT	USBD_BA+0x040	R	Control-ep data count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CEP_CNT							
7	6	5	4	3	2	1	0
CEP_CNT							

Bits	Descriptions	
[15:0]	CEP_CNT	Control-ep Data Count. The DEVICE CONTROLLER maintains the count of the data of control-ep.

Setup1 & Setup0 bytes (SETUP1_0)

Register	Address	R/W	Description	Default Value
SETUP1_0	USBD_BA+0x044	R	Setup1 & Setup0 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP1							
7	6	5	4	3	2	1	0
SETUP0							

Bits	Descriptions																													
[15:8]	SETUP1	Setup Byte 1[15:8]. This register provides byte 1 of the last setup packet received. For a Standard Device Request, the following bRequest Code information is returned.																												
		<table><tr><th>Code</th><th>Descriptions</th></tr><tr><td>0x00</td><td>Get Status</td></tr><tr><td>0x01</td><td>Clear Feature</td></tr><tr><td>0x02</td><td>Reserved</td></tr><tr><td>0x03</td><td>Set Feature</td></tr><tr><td>0x04</td><td>Reserved</td></tr><tr><td>0x05</td><td>Set Address</td></tr><tr><td>0x06</td><td>Get Descriptor</td></tr><tr><td>0x07</td><td>Set Descriptor</td></tr><tr><td>0x08</td><td>Get Configuration</td></tr><tr><td>0x09</td><td>Set Configuration</td></tr><tr><td>0x0A</td><td>Get Interface</td></tr><tr><td>0x0B</td><td>Set Interface</td></tr><tr><td>0x0C</td><td>Synch Frame</td></tr></table>	Code	Descriptions	0x00	Get Status	0x01	Clear Feature	0x02	Reserved	0x03	Set Feature	0x04	Reserved	0x05	Set Address	0x06	Get Descriptor	0x07	Set Descriptor	0x08	Get Configuration	0x09	Set Configuration	0x0A	Get Interface	0x0B	Set Interface	0x0C	Synch Frame
		Code	Descriptions																											
		0x00	Get Status																											
		0x01	Clear Feature																											
		0x02	Reserved																											
		0x03	Set Feature																											
		0x04	Reserved																											
		0x05	Set Address																											
		0x06	Get Descriptor																											
		0x07	Set Descriptor																											
		0x08	Get Configuration																											
		0x09	Set Configuration																											
		0x0A	Get Interface																											
		0x0B	Set Interface																											
0x0C	Synch Frame																													

[7:0]	SETUP0	Setup Byte 0[7:0]. This register provides byte 0 of the last setup packet received. For a Standard Device Request, the following bmRequestType information is returned.	
		Bits	Descriptions
		[7]	Direction 0 = host to device; 1 = device to host
		[6:5]	Type 0 = Standard, 1 = Class, 2 = Vendor, 3 = Reserved
		[4:0]	Recipient 0 = Device, 1 = Interface, 2 = Endpoint, 3 = Other, 4-31 Reserved

Setup3 & Setup2 bytes (SETUP3_2)

Register	Address	R/W	Description	Default Value
SETUP3_2	USBD_BA+0x048	R	Setup3 & Setup2 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP3							
7	6	5	4	3	2	1	0
SETUP2							

Bits	Descriptions	
[15:8]	SETUP3	Setup Byte 3 [15:8]. This register provides byte 3 of the last setup packet received. For a Standard Device Request, the most significant byte of the wValue field is returned.
[7:0]	SETUP2	Setup Byte 2 [7:0]. This register provides byte 2 of the last setup packet received. For a Standard Device Request, the least significant byte of the wValue field is returned.

Setup5 & Setup4 bytes (SETUP5_4)

Register	Address	R/W	Description	Default Value
SETUP5_4	USBD_BA+0x04C	R	Setup5 & Setup4 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP5							
7	6	5	4	3	2	1	0
SETUP4							

Bits	Descriptions	
[15:8]	SETUP5	Setup Byte 5[15:8]. This register provides byte 5 of the last setup packet received. For a Standard Device Request, the most significant byte of the wIndex field is returned.
[7:0]	SETUP4	Setup Byte 4[7:0]. This register provides byte 4 of the last setup packet received. For a Standard Device Request, the least significant byte of the wIndex is returned.

Setup7 & Setup6 bytes (SETUP7_6)

Register	Address	R/W	Description	Default Value
SETUP7_6	USBD_BA+0x050	R	Setup7 & Setup6 bytes	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
SETUP7							
7	6	5	4	3	2	1	0
SETUP6							

Bits	Descriptions	
[15:8]	SETUP7	Setup Byte 7[15:8]. This register provides byte 7 of the last setup packet received. For a Standard Device Request, the most significant byte of the wLength field is returned.
[7:0]	SETUP6	Setup Byte 6[7:0]. This register provides byte 6 of the last setup packet received. For a Standard Device Request, the least significant byte of the wLength field is returned.

Control Ep RAM Start Addr Register (CEP_START_ADDR)

Register	Address	R/W	Description	Default Value
CEP_START_ADDR	USBD_BA+0x054	R/W	Control Ep RAM Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					CEP_START_ADDR		
7	6	5	4	3	2	1	0
CEP_START_ADDR							

Bits	Descriptions	
[10:0]	CEP_START_ADDR	This is the start-address of the RAM space allocated for the control-endpoint

Control Ep RAM End Addr Register (CEP_END_ADDR)

Register	Address	R/W	Description	Default Value
CEP_END_ADDR	USBD_BA+0x058	R/W	Control Ep RAM End Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					CEP_END_ADDR		
7	6	5	4	3	2	1	0
CEP_END_ADDR							

Bits	Descriptions	
[10:0]	CEP_END_ADDR	This is the end-address of the RAM space allocated for the control-endpoint

DMA Control Status Register (DMA_CTRL_STS)

Register	Address	R/W	Description	Default Value
DMA_CTRL_STS	USBD_BA+0x05C	R/W	DMA Control Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
RST_DMA	SCAT_GA_EN	DMA_EN	DMA_RD	DMA_ADDR			

Bits	Descriptions	
[31:8]		Reserved
[7]	RST_DMA	Reset DMA state machine.
[6]	SCAT_GA_EN	Scatter gather function enable
[5]	DMA_EN	DMA Enable Bit
[4]	DMA_RD	DMA Operation Bit. If '1', the operation is a DMA read and if '0' the operation is a DMA write.
[3:0]	DMA_ADDR	DMA ep_addr Bits

When enable scatter gather DMA function, SCAT_GA_EN needs to be set high and DMA_CNT set to 8 bytes. Then DMA will enable to fetch the descriptor which describes the real memory address and length. The descriptor will be a 8-byte format, like the following:

[31]	[30]	[29:0]	
MEM_ADDR[31:0]			
EOT	RD	reserved	count[19:0]

MEM_ADDR: It specifies the memory address(AHB address).

EOT: endo of transfer. When this bit sets to high, it means this is the last descriptor.

RD: "1" means read from memory into buffer. "0" means read from buffer into memory.

DMA Count Register (DMA_CNT)

Register	Address	R/W	Description	Default Value
DMA_CNT	USBD_BA+0x60	R/W	DMA Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				DMA_CNT			
15	14	13	12	11	10	9	8
DMA_CNT							
7	6	5	4	3	2	1	0
DMA_CNT							

Bits	Descriptions	
[31:19]		Reserved
[19:0]	DMA_CNT	The transfer count of the DMA operation to be performed is written to this register.

Endpoint A~D Data Register (EPA_DATA_BUF~ EPC_DATA_BUF)

Register	Address	R/W	Description	Default Value
EPA_DATA_BUF	USBD_BA+0x064	R/W	Endpoint A Data Register	0x0000_0000
EPB_DATA_BUF	USBD_BA+0x08C	R/W	Endpoint B Data Register	0x0000_0000
EPC_DATA_BUF	USBD_BA+0x0B4	R/W	Endpoint C Data Register	0x0000_0000
EPD_DATA_BUF	USBD_BA+0x0DC	R/W	Endpoint D Data Register	0x0000_0000

31	30	29	28	27	26	25	24
EP_DATA_BUF							
23	22	21	20	19	18	17	16
EP_DATA_BUF							
15	14	13	12	11	10	9	8
EP_DATA_BUF							
7	6	5	4	3	2	1	0
EP_DATA_BUF							

Bits	Descriptions	
[31:0]	EP_DATA_BUF	Endpoint A~D Data Register. The data port for the buffer transaction (read or write).

Endpoint A~D Interrupt Status Register (EPA_IRQ_STAT~ EPC_IRQ_STAT)

Register	Address	R/W	Description	Default Value
EPA_IRQ_STAT	USBD_BA+0x068	R/W	Endpoint A Interrupt Status Register	0x0000_0003
EPB_IRQ_STAT	USBD_BA+0x090	R/W	Endpoint B Interrupt Status Register	0x0000_0003
EPC_IRQ_STAT	USBD_BA+0x0B8	R/W	Endpoint C Interrupt Status Register	0x0000_0003
EPD_IRQ_STAT	USBD_BA+0x0E0	R/W	Endpoint D Interrupt Status Register	0x0000_0003

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			O_SHORT_PKT_IS	ERR_IS	NYET_IS	STALL_IS	NAK_IS
7	6	5	4	3	2	1	0
PING_IS	IN_TK_IS	OUT_TK_IS	DATA_RxED_IS	DATA_TxED_IS	SHORT_PKT_IS	EMPTY_IS	FULL_IS

Bits	Descriptions	
[12]	O_SHORT_PKT_IS	Bulk out short packet received. Received bulk out short packet (including zero length packet). Writing a '1' clears this bit.
[11]	ERR_IS	ERR Sent. This bit is set when there occurs any error in the transaction. Writing a '1' clears this bit.
[10]	NYET_IS	NYET Sent. This bit is set when the space available in the RAM is not sufficient to accommodate the next on coming data packet. Writing a '1' clears this bit.
[9]	STALL_IS	USB STALL Sent. The last USB packet could not be accepted or provided because the endpoint was stalled, and was acknowledged with a STALL. Writing a '1' clears this bit.
[8]	NAK_IS	USB NAK sent. The last USB IN packet could not be provided, and was acknowledged with a NAK. Writing a '1' clears this bit.
[7]	PING_IS	PING Token Interrupt. This bit is set when a PING token has been received from the host. Writing a '1' clears this bit.
[6]	IN_TK_IS	Data IN Token Interrupt. This bit is set when a Data IN token has been received from the host. Writing a '1' clears this bit.

[5]	OUT_TK_IS	Data OUT Token Interrupt. This bit is set when a Data OUT token has been received from the host. This bit also set by PING tokens (in high-speed only). Writing a '1' clears this bit.
[4]	DATA_RxED_IS	Data Packet Received Interrupt. This bit is set when a data packet is received from the host by the endpoint. Writing a '1' clears this bit.
[3]	DATA_TxED_IS	Data Packet Transmitted Interrupt. This bit is set when a data packet is transmitted from the endpoint to the host. Writing a '1' clears this bit.
[2]	SHORT_PKT_IS	Short Packet Transferred Interrupt. This bit is set when the length of the last packet was less than the Maximum Packet Size(EP_MPS). Writing a '1' clears this bit.
[1]	EMPTY_IS	Buffer Empty. For an IN endpoint, a buffer is available to the local side for writing up to FIFO full of bytes. This bit is set when the endpoint buffer is empty. For an OUT endpoint, the currently selected buffer has a count of 0, or no buffer is available on the local side (nothing to read).(READ ONLY)
[0]	FULL_IS	Buffer Full. This bit is set when the endpoint packet buffer is full. For an IN endpoint, the currently selected buffer is full, or no buffer is available to the local side for writing (no space to write). For an OUT endpoint, there is a buffer available on the local side, and there are FIFO full of bytes available to be read (entire packet is available for reading).(READ ONLY)

Endpoint A~D Interrupt Enable Register (EPA_IRQ_ENB~ EPC_IRQ_ENB)

Register	Address	R/W	Description	Default Value
EPA_IRQ_ENB	USBD_BA+0x06C	R/W	Endpoint A Interrupt Enable Register	0x0000_0000
EPB_IRQ_ENB	USBD_BA+0x094	R/W	Endpoint B Interrupt Enable Register	0x0000_0000
EPC_IRQ_ENB	USBD_BA+0x0BC	R/W	Endpoint C Interrupt Enable Register	0x0000_0000
EPD_IRQ_ENB	USBD_BA+0x0E4	R/W	Endpoint D Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			O_SHORT_PKT_IE	ERR_IE	NYET_IE	STALL_IE	NAK_IE
7	6	5	4	3	2	1	0
PING_IE	IN_TK_IE	OUT_TK_IE	DATA_RxED_IE	DATA_TxED_IE	SHORT_PKT_IE	EMPTY_IE	FULL_IE

Bits	Descriptions	
[12]	O_SHORT_PKT_IE	Bulk out short packet interrupt enable When set, this bit enables a local interrupt to be set whenever bulk out short packet occurs on the bus for this endpoint.
[11]	ERR_IE	ERR interrupt Enable. When set, this bit enables a local interrupt to be set whenever ERR condition occurs on the bus for this endpoint.
[10]	NYET_IE	NYET Interrupt Enable. When set, this bit enables a local interrupt to be set whenever NYET condition occurs on the bus for this endpoint.
[9]	STALL_IE	USB STALL Sent Interrupt Enable. When set, this bit enables a local interrupt to be set when a stall token is sent to the host.
[8]	NAK_IE	USB NAK Sent Interrupt Enable. When set, this bit enables a local interrupt to be set when a nak token is sent to the host.
[7]	PING_IE	PING Token Interrupt Enable. When set, this bit enables a local interrupt to be set when a ping token has been received from the host.
[6]	IN_TK_IE	Data IN Token Interrupt Enable. When set, this bit enables a local interrupt to be set when a Data IN token has been received from the host.
[5]	OUT_TK_IE	Data OUT Token Interrupt Enable. When set, this bit enables a local interrupt to be set when a Data OUT token has been received from the host.

[4]	DATA_RxED_IE	Data Packet Received Interrupt Enable. When set, this bit enables a local interrupt to be set when a data packet has been transmitted to the host.
[3]	DATA_TxED_IE	Data Packet Transmitted Interrupt Enable. When set, this bit enables a local interrupt to be set when a data packet has been received from the host.
[2]	SHORT_PKT_IE	Short Packet Transferred Interrupt Enable. When set, this bit enables a local interrupt to be set when a short data packet has been transferred to/from the host.
[1]	EMPTY_IE	Buffer Empty Interrupt. When set, this bit enables a local interrupt to be set when a buffer empty condition is detected on the bus.
[0]	FULL_IE	Buffer Full Interrupt. When set, this bit enables a local interrupt to be set when a buffer full condition is detected on the bus.

Endpoint A~D Data Available count register (EPA_DATA_CNT~ EPC_DATA_CNT)

Register	Address	R/W	Description	Default Value
EPA_DATA_CNT	USBD_BA+0x070	R	Endpoint A Data Available count register	0x0000_0000
EPB_DATA_CNT	USBD_BA+0x098	R	Endpoint B Data Available count register	0x0000_0000
EPC_DATA_CNT	USBD_BA+0x0C0	R	Endpoint C Data Available count register	0x0000_0000
EPD_DATA_CNT	USBD_BA+0x0E8	R	Endpoint D Data Available count register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		DMA_LOOP					
23	22	21	20	19	18	17	16
DMA_LOOP							
15	14	13	12	11	10	9	8
DATA_CNT							
7	6	5	4	3	2	1	0
DATA_CNT							

Bits	Descriptions	
[30:16]	DMA_LOOP	This register is the remaining DMA loop to complete. Each loop means 32-byte transfer.
[15:0]	DATA_CNT	For an OUT / IN endpoint, this register returns the number of valid bytes in the endpoint packet buffer.

Endpoint A~D Response Set/Clear Register (EPA_RSP_SC~ EPC_RSP_SC)

Register	Address	R/W	Description	Default Value
EPA_RSP_SC	USBD_BA+0x074	R/W	Endpoint A Response Set/Clear Register	0x0000_0000
EPB_RSP_SC	USBD_BA+0x09C	R/W	Endpoint B Response Set/Clear Register	0x0000_0000
EPC_RSP_SC	USBD_BA+0x0C4	R/W	Endpoint C Response Set/Clear Register	0x0000_0000
EPD_RSP_SC	USBD_BA+0x0EC	R/W	Endpoint D Response Set/Clear Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DIS_BUF	PK_END	ZEROLEN	HALT	TOGGLE	MODE		BUF_FLUSH

Bits	Descriptions	
[7]	DIS_BUF	Disable buffer This bit is used to disable buffer (set buffer size to 1) when received a bulk out short packet.
[6]	PK_END	Packet End. This bit is applicable only in case of Auto-Validate Method. This bit is set to validate any remaining data in the buffer which is not equal to the MPS of the endpoint, and happens to be the last transfer.
[5]	ZEROLEN	Zerolen In. This bit is used to send a zero-length packet in response to an in-token. When this bit is set, a zero packet is sent to the host on reception of an in-token.
[4]	HALT	Endpoint Halt. This bit is used to send a stall handshake as response to the token from the host. When an Endpoint Set Feature (ep_halt) is detected by the local CPU, it must write a '1' to this bit.
[3]	TOGGLE	Endpoint Toggle. This bit is used to clear the endpoint data toggle bit. Reading this bit returns the current state of the endpoint data toggle bit. The local CPU may use this bit, to initialize the end-point's toggle incase of reception of a Set Interface request or a Clear Feature (ep_halt) request from the host. Only when toggle bit is "1", this bit can be written into the inversed write data bit[3].

[2:1]	MODE	Mode. These two bits decide the mode of operation of the in-endpoint.	
		MODE[2:1]	Mode Description
		2' b00	Auto-Validate Mode
		2' b01	Manual-Validate Mode
		2' b10	Fly Mode
		2' b11	Reserved.
		These bits are not valid for an out-endpoint. The auto validate mode will be activated when the reserved mode is selected. (These modes are explained detailed in later sections)	
[0]	BUF_FLUSH	Buffer Flush. Writing a 1 to this bit causes the packet buffer to be flushed and the corresponding EP_AVAIL register to be cleared. This bit is self-clearing. This bit should always be written after a configuration event.	

Endpoint A~D Maximum Packet Size Register (EPA_MPS~ EPC_MPS)

Register	Address	R/W	Description	Default Value
EPA_MPS	USBD_BA+0x078	R/W	Endpoint A Maximum Packet Size Register	0x0000_0000
EPB_MPS	USBD_BA+0x0A0	R/W	Endpoint B Maximum Packet Size Register	0x0000_0000
EPC_MPS	USBD_BA+0x0C8	R/W	Endpoint C Maximum Packet Size Register	0x0000_0000
EPD_MPS	USBD_BA+0x0F0	R/W	Endpoint D Maximum Packet Size Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					EP_MPS		
7	6	5	4	3	2	1	0
EP_MPS							

Bits	Descriptions	
[10:0]	EP_MPS	Endpoint Maximum Packet Size. This field determines the Endpoint Maximum Packet Size.

Endpoint A~D Transfer Count Register (EPA_TRF_CNT~ EPC_TRF_CNT)

Register	Address	R/W	Description	Default Value
EPA_TRF_CNT	USBD_BA+0x07C	R/W	Endpoint A Transfer Count Register	0x0000_0000
EPB_TRF_CNT	USBD_BA+0x0A4	R/W	Endpoint B Transfer Count Register	0x0000_0000
EPC_TRF_CNT	USBD_BA+0x0CC	R/W	Endpoint C Transfer Count Register	0x0000_0000
EPD_TRF_CNT	USBD_BA+0x0F4	R/W	Endpoint D Transfer Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					EP_TRF_CNT		
7	6	5	4	3	2	1	0
EP_TRF_CNT							

Bits	Descriptions	
[10:0]	EP_TRF_CNT	For IN endpoints, this field determines the total number of bytes to be sent to the host in case of manual validation method. For OUT endpoints, this field has no effect

Endpoint A~D Configuration Register (EPA_CFG~ EPC_CFG)

Register	Address	R/W	Description	Default Value
EPA_CFG	USBD_BA+0x080	R/W	Endpoint A Configuration Register	0x0000_0012
EPB_CFG	USBD_BA+0x0A8	R/W	Endpoint B Configuration Register	0x0000_0022
EPC_CFG	USBD_BA+0x0D0	R/W	Endpoint C Configuration Register	0x0000_0032
EPD_CFG	USBD_BA+0x0F8	R/W	Endpoint D Configuration Register	0x0000_0042

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						EP_MULT	
7	6	5	4	3	2	1	0
EP_NUM				EP_DIR	EP_TYPE		EP_VALID

Bits	Descriptions											
[9:8]	EP_MULT	MULT Field. This field indicates number of transactions to be carried out in one single micro frame.										
		<table><tr><th>[9:8]</th><th>Description</th></tr><tr><td>0x00</td><td>One transaction</td></tr><tr><td>0x01</td><td>Reserved</td></tr><tr><td>0x10</td><td>Reserved</td></tr><tr><td>0x11</td><td>Invalid</td></tr></table>	[9:8]	Description	0x00	One transaction	0x01	Reserved	0x10	Reserved	0x11	Invalid
		[9:8]	Description									
		0x00	One transaction									
		0x01	Reserved									
		0x10	Reserved									
0x11	Invalid											
[7:4]	EP_NUM	Endpoint Number. This field selects the number of the endpoint. Valid numbers 1 to 15.										
[3]	EP_DIR	Endpoint Direction. EP_DIR = 0 - OUT EP (Host OUT to Device) EP_DIR = 1- IN EP (Host IN to Device) Note that a maximum of one OUT and IN endpoint is allowed for each endpoint number.										
[2:1]	EP_TYPE	Endpoint Type. This field selects the type of this endpoint. Endpoint 0 is forced to a Control type.										
		<table><tr><th>[2:1]</th><th>Description</th></tr><tr><td>0x00</td><td>Reserved</td></tr><tr><td>0x01</td><td>Bulk</td></tr><tr><td>0x10</td><td>Interrupt</td></tr><tr><td>0x11</td><td>Isochronous</td></tr></table>	[2:1]	Description	0x00	Reserved	0x01	Bulk	0x10	Interrupt	0x11	Isochronous
		[2:1]	Description									
		0x00	Reserved									
		0x01	Bulk									
		0x10	Interrupt									
0x11	Isochronous											

[0]	EP_VALID	Endpoint Valid. When set, this bit enables this endpoint. This bit has no effect on Endpoint 0, which is always enabled.

Endpoint A~D RAM Start Address Register (EPA_START_ADDR~ EPC_START_ADDR)

Register	Address	R/W	Description	Default Value
EPA_START_ADDR	USBD_BA+0x084	R/W	Endpoint A RAM Start Address Register	0x0000_0000
EPB_START_ADDR	USBD_BA+0x0AC	R/W	Endpoint B RAM Start Address Register	0x0000_0000
EPC_START_ADDR	USBD_BA+0x0D4	R/W	Endpoint C RAM Start Address Register	0x0000_0000
EPD_START_ADDR	USBD_BA+0x0FC	R/W	Endpoint D RAM Start Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				EP_START_ADDR			
7	6	5	4	3	2	1	0
EP_START_ADDR							

Bits	Descriptions	
[11:0]	EP_START_ADDR	This is the start-address of the RAM space allocated for the endpoint A~F. The start address must be word boundary.

Endpoint A~D RAM End Address Register (EPA_END_ADDR~ EPC_END_ADDR)

Register	Address	R/W	Description	Default Value
EPA_END_ADDR	USBD_BA+0x088	R/W	Endpoint A RAM End Address Register	0x0000_0000
EPB_END_ADDR	USBD_BA+0x0B0	R/W	Endpoint B RAM End Address Register	0x0000_0000
EPC_END_ADDR	USBD_BA+0x0D8	R/W	Endpoint C RAM End Address Register	0x0000_0000
EPD_END_ADDR	USBD_BA+0x100	R/W	Endpoint D RAM End Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				EP_END_ADDR			
7	6	5	4	3	2	1	0
EP_END_ADDR							

Bits	Descriptions	
[11:0]	EP_END_ADDR	This is the end-address of the RAM space allocated for the endpoint A~F. The allocated buffer size for each endpoint must be word boundary.

USB Memory test (USB_MEM_TEST)

Register	Address	R/W	Description	Default Value
USB_MEM_TEST	USBD_BA+0x154	R/W	USB memory test register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							OUT_O
7	6	5	4	3	2	1	0
FAIL_O	FINISH_O	SHIFT_O	MODE_O	FAIL_A	FINISH_A	ERR_A	MODE_A

Bits	Descriptions	
[31:9]		Reserved
[8]	OUT_O	BIST signature out
[7]	FAIL_O	BIST result of internal ROM
[6]	FINISH_O	BIST finish for internal ROM
[5]	SHIFT_O	BIST shift end for internal ROM
[4]	MODE_O	BIST enable for internal ROM
[3]	FAIL_A	BIST result of internal RAM
[2]	FINISH_A	BIST finish for internal RAM
[1]	ERR_A	BIST error map for internal RAM
[0]	MODE_A	BIST enable for internal RAM

USB Head word0(USB_HEAD0)

Register	Address	R/W	Description	Default Value
USB_head word0	USBD_BA+0x158	R/W	USB head word0	0x0000_0000

31	30	29	28	27	26	25	24
HEAD_WORD0							
23	22	21	20	19	18	17	16
HEAD_WORD0							
15	14	13	12	11	10	9	8
HEAD_WORD0							
7	6	5	4	3	2	1	0
HEAD_WORD0							

Bits	Descriptions	
[31:0]	HRAD WORD0	The first head data(byte 0 was sent first)

USB Head word1(USB_HEAD1)

Register	Address	R/W	Description	Default Value
USB_head word1	USBD_BA+0x15C	R/W	USB head word1	0x0000_0000

31	30	29	28	27	26	25	24
HEAD_WORD1							
23	22	21	20	19	18	17	16
HEAD_WORD1							
15	14	13	12	11	10	9	8
HEAD_WORD1							
7	6	5	4	3	2	1	0
HEAD_WORD1							

Bits	Descriptions	
[31:0]	HRAD WORD1	The second head data(byte 0 was sent first)

USB Head word2(USB_HEAD2)

Register	Address	R/W	Description	Default Value
USB_head word2	USBD_BA+0x160	R/W	USB head word2	0x0000_0000

31	30	29	28	27	26	25	24
HEAD_WORD2							
23	22	21	20	19	18	17	16
HEAD_WORD2							
15	14	13	12	11	10	9	8
HEAD_WORD2							
7	6	5	4	3	2	1	0
HEAD_WORD2							

Bits	Descriptions	
[31:0]	HRAD WORD2	The third head data(byte 0 was sent first)

Endpoint A~D RAM End Address Register (EPA_END_ADDR~EPC_END_ADDR)

Register	Address	R/W	Description	Default Value
EPA_HEAD_CNT	USBD_BA+0x164	R/W	Endpoint A header count	0x0000_0000
EPB_HEAD_CNT	USBD_BA+0x168	R/W	Endpoint B header count	0x0000_0000
EPC_HEAD_CNT	USBD_BA+0x16C	R/W	Endpoint C header count	0x0000_0000
EPD_HEAD_CNT	USBD_BA+0x170	R/W	Endpoint D header count	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				EP_HEAD_CNT			

Bits	Descriptions	
[11:0]	EP_HEAD_CNT	This is the header count for the endpoint A~F. The header count must be EVEN.

AHB Address Register (AHB_DMA_ADDR)

Register	Address	R/W	Description	Default Value
AHB_DMA_ADDR	USBD_BA+0x700	R/W	AHB address register	0x0000_0000

31	30	29	28	27	26	25	24
AHB_DMA_ADDR							
23	22	21	20	19	18	17	16
AHB_DMA_ADDR							
15	14	13	12	11	10	9	8
AHB_DMA_ADDR							
7	6	5	4	3	2	1	0
AHB_DMA_ADDR							

Bits	Descriptions	
[31:0]	AHB_DMA_ADDR	It specifies the address from which the DMA has to read / write. The address must WORD (32-bits) aligned.

USB PHY Control (USB_PHY_CTL)

Register	Address	R/W	Description	Default Value
USB_PHY_CTL	USBD_BA+0x704	R/W	USB PHY control register	0x0000_0020

31	30	29	28	27	26	25	24
Vbus_status	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				Reserved		Phy_suspend	vbus_detect
7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved		Reserved	Reserved	Reserved	Reserved

Bits	Descriptions	
[31]	Vbus_status	Vbus status 1: Vbus on 0: Vbus off
[30:24]		Reserved
[9]	Phy_suspend	Set this bit low will cause USB PHY to be suspended.
[8]	vbus_detect	Set PHY vbus_detect

5.13 USB Host Controller (UHC)

5.13.1 Overview

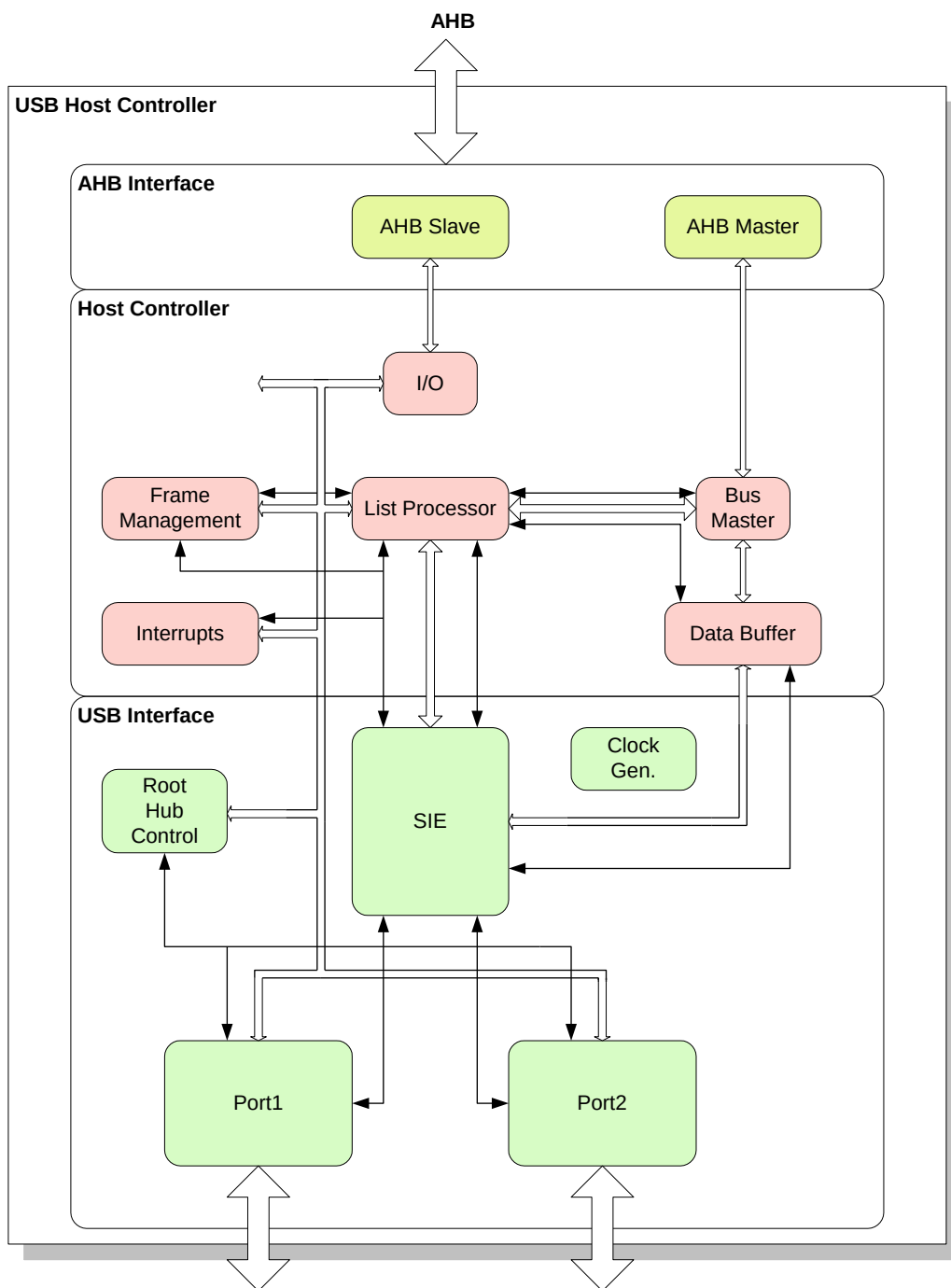
The **Universal Serial Bus (USB)** is a low-cost, low-to mid-speed peripheral interface standard intended for modem, scanners, PDAs, keyboards, mice, and other devices that do not require a high-bandwidth parallel interface. The USB is a 4-wire serial cable bus that supports serial data exchange between a Host Controller and a network of peripheral devices. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. Peripherals may be attached, configured, used, and detached, while the host and other peripherals continue operation (i.e. hot plug and unplug is supported).

A major design goal of the USB standard was to allow flexible, plug-and-play networks of USB devices. In any USB network, there will be only one host, but there can be many devices and hubs.

5.13.2 Features

- Compliant with USB Revision 1.1 specification.
- Open Host Controller Interface (OHCI) Revision 1.0 compatible.
- Support both full-speed (12Mbps) and low-speed (1.5Mbps) USB devices.
- Support Control, Bulk, Interrupt and Isochronous transfers.
- Support Built-in DMA for real-time data transfer.
- Support multiple low power modes for efficient power management.

5.13.3 Block Diagram



5.13.4 Operation

5.13.4.1 OHCI Controller

AHB Interface

The OpenHCI Host Controller is connected to the system by the AHB bus. The design requires both master and slave bus operations. As a master, the Host Controller is responsible for running cycles on the AHB bus to access EDs and TDs as well as transferring data between memory and the local data buffer. As a slave, the Host Controller monitors the cycles on the AHB bus and determines when to respond to these cycles. Configuration and non-real-time control access to the Host Controller operational registers are through the AHB bus slave interface.

AHB Master

The master issues the address and data onto the bus when granted.

AHB Slave

The configuration of the Host Controller is through the slave interface.

Host Controller

List Processing

The List Processor manages the data structures from the Host Controller Driver and coordinates all activity within the Host Controller.

Frame Management

Frame Management is responsible for managing the frame specific tasks required by the USB specification and the OpenHCI specification. These tasks are:

- 1) Management of the OpenHCI frame specific Operational Registers
- 2) Operation of the Largest Data Packet Counter.
- 3) Performing frame qualifications on USB Transaction requests to the SIE.
- 4) Generate SOF token requests to the SIE.

Interrupt Processing

Interrupts are the communication method for HC-initiated communication with the Host Controller Driver. There are several events that may trigger an interrupt from the Host Controller. Each specific event sets a specific bit in the *HcInterruptStatus* register.

Host Controller Bus Master

The Host Controller Bus Master is the central block in the data path. The Host Controller Bus Master coordinates all access to the AHB Interface. There are two sources of bus mastering within Host Controller: the List Processor and the Data Buffer Engine.

Data Buffer

The Data Buffer serves as the data interface between the Bus Master and the SIE. It is a combination of a 64-byte latched based bi-directional asynchronous FIFO and a single Dword AHB Holding Register.

USB Interface

The USB interface includes the integrated Root Hub with two external ports, Port 1 and Port 2 as well as the Serial Interface Engine (SIE) and USB clock generator. The interface combines responsibility for executing bus transactions requested by the HC as well as the hub and port management specified by USB.

SIE

The SIE is responsible for managing all transactions to the USB. It controls the bus protocol, packet generation/extraction, data parallel-to-serial conversion, CRC coding, bit stuffing, and NRZI encoding. All transactions on the USB are requested from the List

Processor and Frame Manager.

Root Hub

The Root Hub is a collection of ports that are individually controlled and a hub that maintains control/status over functions common to all ports.

5.13.5 Register Mapping

Register	Offset	R/W	Description	Reset Value
Capability Registers (UHC_BA = 0xB100_9000)				
HcRev	UHC_BA+0x000	R	Host Controller Revision Register	0x0000_0110
HcControl	UHC_BA+0x004	R/W	Host Controller Control Register	0x0000_0000
HcComSts	UHC_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000
HcIntSts	UHC_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcIntEn	UHC_BA+0x010	R/W	Host Controller Interrupt Enable Register	0x0000_0000
HcIntDis	UHC_BA+0x014	R/W	Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	UHC_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPerCED	UHC_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcCtrHED	UHC_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcCtrCED	UHC_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBlkHED	UHC_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBlkCED	UHC_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000
HcDoneH	UHC_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmIntv	UHC_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF
HcFmRem	UHC_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000
HcFNum	UHC_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000
HcPerSt	UHC_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSTH	UHC_BA+0x044	R/W	Host Controller Low Speed Threshold Register	0x0000_0628
HcRhDeA	UHC_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0002
HcRhDeB	UHC_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhSts	UHC_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPrt1	UHC_BA+0x054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPrt2	UHC_BA+0x058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000
	UHC_BA+0x05C ... UHC_BA+0x1FC		Reserved	Undefined
OHCI USB Configuration Register				
MiscCtrl	UHC_BA+0x200	R/W	USB Miscellaneous Control Register	0x0000_0000
OpModEn	UHC_BA+0x204	R/W	USB Operational Mode Enable Register	0x0000_0000

5.13.6 Register Details

Host Controller Revision Register (HcRev)

Register	Address	R/W	Description	Reset Value
HcRev	UHC_BA+0x000	R	Host Controller Revision Register	0x0000_0110

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Rev							

Bits	Descriptions	
[31:8]	Reserved	
[7:0]	Rev	Revision Indicates the Open HCI Specification revision number implemented by the Hardware. Host Controller supports USB 1.1 specification. (X.Y = XYh)

Host Controller Control Register (HcControl)

Register	Address	R/W	Description	Reset Value
HcControl	UHC_BA+0x004	R/W	Host Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					RWakeEn	RWake	IntRoute
7	6	5	4	3	2	1	0
HcFunc		BlkEn	CtrlEn	ISOEn	PeriEn	CtrlBlkRatio	

Bits	Descriptions	
[31:11]	Reserved	
[10]	RWakeEn	Remote Wakeup Connected Enable If a remote wakeup signal is supported, this bit enables that operation. Since there is no remote wakeup signal supported, this bit is ignored.
[9]	RWake	Remote Wakeup Connected This bit indicated whether the HC supports a remote wakeup signal. This implementation does not support any such signal. The bit is hard-coded to '0.'
[8]	IntRoute	Interrupt Routing This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
[7:6]	HcFunc	Host Controller Functional State This field sets the Host Controller state. The Controller may force a state change from USBsuspend to USBRESUME after detecting resume signaling from a downstream port. States are: 00: USBRESET 01: USBRESUME 10: USBOPERATIONAL 11: USBSUSPEND
[5]	BlkEn	Bulk List Enable When set this bit enables processing of the Bulk list.
[4]	CtrlEn	Control List Enable When set this bit enables processing of the Control list.

[3]	ISOEn	Isochronous List Enable When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous ED.
[2]	PeriEn	Periodic List Enable When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.
[1:0]	CtrlBlkRatio	Control Bulk Service Ratio Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e. '00' = 1 Control Endpoint; '11' = 3 Control Endpoints)

Host Controller Command Status Register (HcComSts)

Register	Address	R/W	Description	Reset Value
HcComSts	UHC_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						SchOverRun	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				OCReq	BlkFill	CtrlFill	HCRreset

Bits	Descriptions	
[31:18]	Reserved	
[17:16]	SchOverRun	Schedule Overrun Count This field is increment every time the SchedulingOverrun bit in <i>HcInterruptStatus</i> is set. The count wraps from '11' to '00.'
[15:4]	Reserved	
[3]	OCReq	Ownership Chang Request When set by software, this bit sets the OwnershipChange field in <i>HcInterruptStatus</i> . The bit is cleared by software.
[2]	BlkFill	Bulk List Filled Set to indicate there is an active ED on the Bulk List. The bit may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Bulk List.
[1]	CtrlFill	Control List Filled Set to indicate there is an active ED on the Control List. It may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Control List.
[0]	HCRreset	Host Controller Reset This bit is set to initiate the software reset. This bit is cleared by the Host Controller, upon completed of the reset operation.

Host Controller Interrupt Status Register (HcIntSts)

Register	Address	R/W	Description	Reset Value
HcIntSts	UHC_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	OC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSC	FNOF	UnRecErr	Resume	SOF	WBDnHD	SchOR

Bits	Descriptions	
[31]	Reserved	
[30]	OC	Ownership Change This bit is set when the OwnershipChangeRequest bit of <i>HcCommandStatus</i> is set.
[29:7]	Reserved	
[6]	RHSC	Root Hub Status Change This bit is set when the content of <i>HcRhStatus</i> or the content of any <i>HcRhPortStatus</i> register has changed.
[5]	FNOF	Frame Number Overflow Set when bit 15 of <i>FrameNumber</i> changes value.
[4]	UnRecErr	Unrecoverable Error This event is not implemented and is hard-coded to '0.' Writes are ignored.
[3]	Resume	Resume Detected Set when Host Controller detects resume signaling on a downstream port.
[2]	SOF	Start Of Frame Set when the Frame Management block signals a 'Start of Frame' event.
[1]	WBDnHD	Write Back Done Head Set after the Host Controller has written <i>HcDoneHead</i> to <i>HccaDoneHead</i> .
[0]	SchOR	Scheduling Overrun Set when the List Processor determines a Schedule Overrun has occurred.

Host Controller Interrupt Enable Register (HcIntEn)

Register	Address	R/W	Description	Reset Value
HcIntEn	UHC_BA+0x010	R/W	Host Controller Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
IntEn	OCEn	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSCEn	FNOFEn	UREREn	ResuEn	SOFEn	WBDHEn	SchOREn

Bits	Descriptions	
[31]	IntEn	Master Interrupt Enable This bit is a global interrupt enable. A write of '1' allows interrupts to be enabled via the specific enable bits listed above.
[30]	OCEn	Ownership Change Enable 0: Ignore 1: Enables interrupt generation due to Ownership Change.
[29:7]	Reserved	
[6]	RHSCEn	Root Hub Status Change Enable 0: Ignore 1: Enables interrupt generation due to Root Hub Status Change.
[5]	FNOFEn	Frame Number Overflow Enable 0: Ignore 1: Enables interrupt generation due to Frame Number Overflow.
[4]	UREREn	Unrecoverable Error Enable This event is not implemented. All writes to this bit are ignored.
[3]	ResuEn	Resume Detected Enable 0: Ignore 1: Enables interrupt generation due to Resume Detected.
[2]	SOFEn	Start Of Frame Enable 0: Ignore 1: Enables interrupt generation due to Start of Frame.
[1]	WBDHEn	Write Back Done Head Enable 0: Ignore 1: Enables interrupt generation due to Write-back Done Head.

[0]	SchOREn	Scheduling Overrun Enable 0: Ignore 1: Enables interrupt generation due to Scheduling Overrun.

Host Controller Interrupt Disable Register (HcIntDis)

Register	Address	R/W	Description	Reset Value
HcIntDis	UHC_BA+0x014	R/W	Host Controller Interrupt Disable Register	0x0000_0000

31	30	29	28	27	26	25	24
IntDis	OCDIs	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSCDis	FNOFDis	URERDis	ResuDis	SOFDis	WBDHDis	SchORDis

Bits	Descriptions	
[31]	IntDis	Master Interrupt Disable Global interrupt disable. A write of '1' disables all interrupts.
[30]	OCDIs	Ownership Change Disable 0: Ignore 1: Disables interrupt generation due to Ownership Change.
[29:7]	Reserved	
[6]	RHSCDis	Root Hub Status Change Disable 0: Ignore 1: Disables interrupt generation due to Root Hub Status Change.
[5]	FNOFDis	Frame Number Overflow Disable 0: Ignore 1: Disables interrupt generation due to Frame Number Overflow.
[4]	URERDis	Unrecoverable Error Disable This event is not implemented. All writes to this bit are ignored.
[3]	ResuDis	Resume Detected Disable 0: Ignore 1: Disables interrupt generation due to Resume Detected.
[2]	SOFDis	Start Of Frame Disable 0: Ignore 1: Disables interrupt generation due to Start of Frame.
[1]	WBDHDis	Write Back Done Head Disable 0: Ignore 1: Disables interrupt generation due to Write-back Done Head.

[0]	SchORDis	Scheduling Overrun Disable 0: Ignore 1: Disables interrupt generation due to Scheduling Overrun.

Host Controller Communication Area Register (HcHCCA)

Register	Address	R/W	Description	Reset Value
HcHCCA	UHC_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000

31	30	29	28	27	26	25	24
HCCA							
23	22	21	20	19	18	17	16
HCCA							
15	14	13	12	11	10	9	8
HCCA							
7	6	5	4	3	2	1	0
Reserved							

Bits	Descriptions	
[31:7]	HCCA	Host Controller Communication Area Pointer to HCCA base address.
[7:0]	Reserved	

Host Controller Period Current ED Register (HcPerCED)

Register	Address	R/W	Description	Reset Value
HcPerCED	UHC_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
PeriCED							
23	22	21	20	19	18	17	16
PeriCED							
15	14	13	12	11	10	9	8
PeriCED							
7	6	5	4	3	2	1	0
PeriCED				Reserved			

Bits	Descriptions	
[31:4]	PeriCED	Periodic Current ED Pointer to the current Periodic List ED.
[3:0]	Reserved	

Host Controller Control Head ED Register (HcCtrHED)

Register	Address	R/W	Description	Reset Value
HcCtrHED	UHC_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000

31	30	29	28	27	26	25	24
CtrlHED							
23	22	21	20	19	18	17	16
CtrlHED							
15	14	13	12	11	10	9	8
CtrlHED							
7	6	5	4	3	2	1	0
CtrlHED				Reserved			

Bits	Descriptions	
[31:4]	CtrlHED	Control Head ED Pointer to the Control List Head ED.
[3:0]	Reserved	

Host Controller Control Current ED Register (HcCtrCED)

Register	Address	R/W	Description	Reset Value
HcCtrCED	UHC_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
CtrICED							
23	22	21	20	19	18	17	16
CtrICED							
15	14	13	12	11	10	9	8
CtrICED							
7	6	5	4	3	2	1	0
CtrICED				Reserved			

Bits	Descriptions	
[31:4]	CtrICED	Control Current Head ED Pointer to the current Control List Head ED.
[3:0]	Reserved	

Host Controller Bulk Head ED Register (HcBlkHED)

Register	Address	R/W	Description	Reset Value
HcBlkHED	UHC_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000

31	30	29	28	27	26	25	24
BIKHED							
23	22	21	20	19	18	17	16
BIKHED							
15	14	13	12	11	10	9	8
BIKHED							
7	6	5	4	3	2	1	0
BIKHED				Reserved			

Bits	Descriptions	
[31:4]	BIKHED	Bulk Head ED Pointer to the Bulk List Head ED.
[3:0]	Reserved	

Host Controller Bulk Current Head ED Register (HcBlkCED)

Register	Address	R/W	Description	Reset Value
HcBlkCED	UHC_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000

31	30	29	28	27	26	25	24
BlkCED							
23	22	21	20	19	18	17	16
BlkCED							
15	14	13	12	11	10	9	8
BlkCED							
7	6	5	4	3	2	1	0
BlkCED				Reserved			

Bits	Descriptions	
[31:4]	BlkCED	Bulk Current Head ED Pointer to the current Bulk List Head ED.
[3:0]	Reserved	

Host Controller Done Head Register (HcDoneH)

Register	Address	R/W	Description	Reset Value
HcDoneH	UHC_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000

31	30	29	28	27	26	25	24
DoneH							
23	22	21	20	19	18	17	16
DoneH							
15	14	13	12	11	10	9	8
DoneH							
7	6	5	4	3	2	1	0
DoneH				Reserved			

Bits	Descriptions	
[31:4]	DoneH	Done Head Pointer to the current Done List Head ED.
[3:0]	Reserved	

Host Controller Frame Interval Register (HcFmIntv)

Register	Address	R/W	Description	Reset Value
HcFmIntv	UHC_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

31	30	29	28	27	26	25	24
FmIntvT	FSDPktCnt						
23	22	21	20	19	18	17	16
FSDPktCnt							
15	14	13	12	11	10	9	8
Reserved		FmInterval					
7	6	5	4	3	2	1	0
FmInterval							

Bits	Descriptions	
[31]	FmIntvT	Frame Interval Toggle This bit is toggled by HCD when it loads a new value into FrameInterval.
[30: 16]	FSDPktCnt	FS Largest Data Packet This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
[15:14]	Reserved	
[13:0]	FmInterval	Frame Interval This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

Host Controller Frame Remaining Register (HcFmRem)

Register	Address	R/W	Description	Reset Value
HcFmRem	UHC_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000

31	30	29	28	27	26	25	24
FmRemT	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		FmRemain					
7	6	5	4	3	2	1	0
FmRemain							

Bits	Descriptions	
[31]	FmRemT	Frame Remaining Toggle Loaded with FrameIntervalToggle when FrameRemaining is loaded.
[30:14]	Reserved	
[13:0]	FmRemain	Frame Remaining When the Host Controller is in the USBOPERATIONAL state, this 14-bit field decrements each 12 MHz clock period. When the count reaches 0, (end of frame) the counter reloads with FrameInterval. In addition, the counter loads when the Host Controller transitions into USBOPERATIONAL.

Host Controller Frame Number Register (HcFNum)

Register	Address	R/W	Description	Reset Value
HcFNum	UHC_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
FmNum							
7	6	5	4	3	2	1	0
FmNum							

Bits	Descriptions	
[31:16]	Reserved	
[15:0]	FmNum	Frame Number This 16-bit incrementing counter field is incremented coincident with the loading of FrameRemaining. The count rolls over from 'FFFFh' to '0h.'

Host Controller Periodic Start Register (HcPerSt)

Register	Address	R/W	Description	Reset Value
HcPerSt	UHC_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		PeriStart					
7	6	5	4	3	2	1	0
PeriStart							

Bits	Descriptions	
[31:14]	Reserved	
[13:0]	PeriStart	Periodic Start This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

Host Controller Low Speed Threshold Register (HcLSTH)

Register	Address	R/W	Description	Reset Value
HcLSTH	UHC_BA+0x044	R/W	Host Controller Low Speed Threshold Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				LST			
7	6	5	4	3	2	1	0
LST							

Bits	Descriptions	
[31:12]	Reserved	
[12:0]	LST	Low Speed Threshold This field contains a value which is compared to the FrameRemaining field prior to initiating a Low Speed transaction. The transaction is started only if FrameRemaining >= this field. The value is calculated by HCD with the consideration of transmission and setup overhead.

Host Controller Root Hub Descriptor A Register (HcRhDeA)

Register	Address	R/W	Description	Reset Value
HcRhDeA	UHC_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0002

31	30	29	28	27	26	25	24
PwrGDT							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved			NOCP	OCPM	DevType	NPS	PSM
7	6	5	4	3	2	1	0
DPortNum							

Bits	Descriptions	
[31:24]	PwrGDT	Power On to Power Good Time This field value is represented as the number of 2 ms intervals, which ensuring that the power switching is effective within 2 ms. Only bits [25:24] are implemented as R/W. The remaining bits are read only as '0'. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should be written to support system implementation. This field should always be written to a non-zero value.
[23:13]	Reserved	
[12]	NOCP	No Over Current Protection Global over-current reporting implemented in HYDRA-2. This bit should be written to support the external system port over-current implementation. 0 = Over-current status is reported 1 = Over-current status is not reported
[11]	OCPM	Over Current Protection Mode Global over-current reporting implemented in HYDRA-2. This bit should be written 0 and is only valid when NoOverCurrentProtection is cleared. 0 = Global Over-Current 1 = Individual Over-Current
[10]	DevType	Device Type HYDRA-4is not a compound device.
[9]	NPS	No Power Switching Global power switching implemented in HYDRA-2. This bit should be written to support the external system port power switching implementation. 0 = Ports are power switched. 1 = Ports are always powered on.

[8]	PSM	Power Switching Mode Global power switching mode implemented in HYDRA-2. This bit is only valid when NoPowerSwitching is cleared. This bit should be written '0'. 0 = Global Switching 1 = Individual Switching
[7:0]	DPortNum	Number Downstream Ports HYDRA-4 supports two downstream ports.

Host Controller Root Hub Descriptor B Register (HcRhDeB)

Register	Address	R/W	Description	Reset Value
HcRhDeB	UHC_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000

31	30	29	28	27	26	25	24
PPCM							
23	22	21	20	19	18	17	16
PPCM							
15	14	13	12	11	10	9	8
DevRemove							
7	6	5	4	3	2	1	0
DevRemove							

Bits	Descriptions	
[31:16]	PPCM	Port Power Control Mask Global-power switching. This field is only valid if NoPowerSwitching is cleared and PowerSwitchingMode is set (individual port switching). When set, the port only responds to individual port power switching commands (Set/ClearPortPower). When cleared, the port only responds to global power switching commands (Set/ClearGlobalPower). 0 = Device not removable 1 = Global-power mask Port Bit relationship - Unimplemented ports are reserved, read/write '0'. 0 : Reserved 1 : Port 1 2 : Port 2 ... 15 : Port 15
[15:0]	DevRemove	Device Removable HYDRA-4 ports default to removable devices. 0 = Device not removable 1 = Device removable Port Bit relationship 0 : Reserved 1 : Port 1 2 : Port 2 ... 15 : Port 15 Unimplemented ports are reserved, read/write '0'.

Host Controller Root Hub Status Register (HcRhSts)

Register	Address	R/W	Description	Reset Value
HcRhSts	UHC_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000

31	30	29	28	27	26	25	24
RWEClr	Reserved						
23	22	21	20	19	18	17	16
Reserved						OCIC	LPSC
15	14	13	12	11	10	9	8
DRWEn	Reserved						
7	6	5	4	3	2	1	0
Reserved						OC	LPS

Bits	Descriptions	
[31]	RWEClr	Clear Remote Wakeup Enable Writing a '1' to this bit clears DeviceRemoteWakeupEnable. Writing a '1' has no effect.
[30:18]	Reserved	
[17]	OCIC	Over Current Indicator Change This bit is set when OverCurrentIndicator changes. Writing a '1' clears this bit. Writing a '0' has no effect.
[16]	LPSC	(Read) LocalPowerStatusChange Not supported. Always read '0'. (Write) SetGlobalPower Write a '1' issues a SetGlobalPower command to the ports. Writing a '0' has no effect.
[15]	DRWEn	(Read) DeviceRemoteWakeupEnable This bit enables ports' ConnectStatusChange as a remote wakeup event. 0 = disabled 1 = enabled (Write) SetRemoteWakeupEnable Writing a '1' sets DeviceRemoteWakeupEnable. Writing a '0' has no effect.
[14:2]	Reserved	
[1]	OC	Over Current Indicator This bit reflects the state of the OVRCUR pin. This field is only valid if NoOverCurrentProtection and OverCurrentProtectionMode are cleared. 0 = No over-current condition 1 = Over-current condition

[0]	LPS	(Read) LocalPowerStatus Not Supported. Always read '0'. (Write) ClearGlobalPower Writing a '1' issues a ClearGlobalPower command to the ports. Writing a '0' has no effect.

Host Controller Root Hub Port Status (HcRhPrt [1: 2])

Register	Address	R/W	Description	Reset Value
HcRhPrt1	UHC_BA+0x054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPrt2	UHC_BA+0x058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			PRSC	POCIC	PSSC	PESC	CSC
15	14	13	12	11	10	9	8
Reserved						LSDev	PPS
7	6	5	4	3	2	1	0
Reserved			PR	POC	PS	PE	CC

Bits	Descriptions	
[31:21]	Reserved	
[20]	PRSC	Port Reset Status Change This bit indicates that the port reset signal has completed. 0 = Port reset is not complete. 1 = Port reset is complete.
[19]	POCIC	Port Over Current Indicator Change This bit is set when OverCurrentIndicator changes. Writing a '1' clears this bit. Writing a '0' has no effect.
[18]	PSSC	Port Suspend Status Change This bit indicates the completion of the selective resume sequence for the port. 0 = Port is not resumed. 1 = Port resume is complete.
[17]	PESC	Port Enable Status Change This bit indicates that the port has been disabled due to a hardware event (cleared PortEnableStatus). 0 = Port has not been disabled. 1 = PortEnableStatus has been cleared.
[16]	CSC	Connect Status Change This bit indicates connect or disconnect event has been detected. Writing a '1' clears this bit. Writing a '0' has no effect. 0 = No connect/disconnect event. 1 = Hardware detection of connect/disconnect event. Note: If DeviceRemoveable is set, this bit resets to '1'.
[15:10]	Reserved	

[9]	LSDev	(Read) LowSpeedDeviceAttached This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0 = Full Speed device 1 = Low Speed device (Write) ClearPortPower Writing a '1' clears PortPowerStatus. Writing a '0' has no effect
[8]	PPS	(Read) PortPowerStatus This bit reflects the power state of the port regardless of the power switching mode. 0 = Port power is off. 1 = Port power is on. Note: If NoPowerSwitching is set, this bit is always read as '1'. (Write) SetPortPower Writing a '1' sets PortPowerStatus. Writing a '0' has no effect.
[7:5]	Reserved	
[4]	PR	(Read) PortResetStatus 0 = Port reset signal is not active. 1 = Port reset signal is active. (Write) SetPortReset Writing a '1' sets PortResetStatus. Writing a '0' has no effect.
[3]	POC	(Read) PortOverCurrentIndicator HYDRA-2 supports global over-current reporting. This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0 = No over-current condition 1 = Over-current condition (Write) ClearPortSuspend Writing a '1' initiates the selective resume sequence for the port. Writing a '0' has no effect.
[2]	PS	(Read) PortSuspendStatus 0 = Port is not suspended 1 = Port is selectively suspended (Write) SetPortSuspend Writing a '1' sets PortSuspendStatus. Writing a '0' has no effect.
[1]	PE	(Read) PortEnableStatus 0 = Port disabled. 1 = Port enabled. (Write) SetPortEnable Writing a '1' sets PortEnableStatus. Writing a '0' has no effect.
[0]	CC	(Read) CurrentConnectStatus 0 = No device connected. 1 = Device connected. NOTE: If DeviceRemoveable is set (not removable) this bit is always '1'. (Write) ClearPortEnable Writing '1' a clears PortEnableStatus. Writing a '0' has no effect.

USB Miscellaneous Control Register (MiscCtrl)

Register	Address	R/W	Description	Reset Value
MiscCtrl	UHC_BA+0x200	R/W	USB Miscellaneous Control Register	0X0000_0000

31	30	29	28	27	26	25	24
Reserved				StbyEn	Reserved		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Descriptions	
[31:28]	Reserved	
[27]	StbyEn	USB Tranceiver Standby Enable This bit controls if USB 1.1 transceiver could enter the standby mode to reduce power consumption. If this bit is low, the USB 1.1 transceiver would never enter the standby mode. If this bit is high, the USB 1.1 transceiver will enter standby mode while port is in power off state (port power is inactive).
[26:0]	Reserved	

USB Operational Mode Enable Register (OpModEn)

Register	Address	R/W	Description	Reset Value
OpModEn	UHC_BA+0x204	R/W	USB Operational Mode Enable Register	0X0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						DisPrt2	DisPrt1
15	14	13	12	11	10	9	8
Reserved							SIEPDis
7	6	5	4	3	2	1	0
Reserved			PPCLow	OCALow	Reserved	ABORT	DBR16

Bits	Descriptions	
[31:18]	Reserved	
[17]	DisPrt2	Disable Port 2 This bit controls if the connection between USB host controller and transceiver of port 2 is disabled. If the connection is disabled, the USB host controller will not recognize any event of USB bus. Set this bit high, the transceiver of port 2 will also be forced into the standby mode no matter what USB host controller operation is. 1'b0: The connection between USB host controller and transceiver of port 2 is enabled. 1'b1: The connection between USB host controller and transceiver of port 2 is disabled and the transceiver of port 2 will also be forced into the standby mode.
[16]	DisPrt1	Disable Port 1 This bit controls if the connection between USB host controller and transceiver of port 1 is disabled. If the connection is disabled, the USB host controller will not recognize any event of USB bus. Set this bit high, the transceiver of port 1 will also be forced into the standby mode no matter what USB host controller operation is. 1'b0: The connection between USB host controller and transceiver of port 1 is enabled. 1'b1: The connection between USB host controller and transceiver of port 1 is disabled and the transceiver of port 1 will also be forced into the standby mode
[15:9]	Reserved	Reserved
[8]	SIEPDis	SIE Pipeline Disable When set, waits for all USB bus activity to complete prior to returning completion status to the List Processor. This is a failsafe mechanism to avoid potential problems with the clk_dr transition between 1.5 MHz and 12 MHz.
[7:5]	Reserved	

[4]	PPCALow	Port Power Control Active Low This bit controls the polarity of port power control to external power IC. 0: Port power control is high active 1: Port power control is low active
[3]	OCCALow	Over Current Active Low This bit controls the polarity of over current flag from external power IC. 0: Over current flag is high active 1: Over current flag is low active
[2]	Reserved	
[1]	ABORT	AHB Bus ERROR Response This bit indicates there is an ERROR response received in AHB bus. 0: No ERROR response received 1: ERROR response received
[0]	DBR16	Data Buffer Region 16 When set, the size of the data buffer region is 16 bytes. Otherwise, the size is 32 bytes.

5.14 Enhanced DMA Controller

The N9H20 contains an enhanced direct memory access (EDMA) controller that transfers data to and from memory or transfer data to and from APB. The EDMA controller has five-channel DMA that include a one channel VDMA (Video-DMA, Memory-to-Memory) and four channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral). For channel0 VDMA mode, it also support color format transform and stripe mode transfer. For PDMA channel (EDMA CH1~CH4), it can transfer data between the Peripherals APB IP (ex: UART, SPI, ADC....) and Memory. The N9H20 also support hardware scatter-gather function, software can set CSRx [SG_EN] to enable scatter-gather function.

Software can stop the EDMA operation by disable DMA [DMACEN]. The CPU can recognize the completion of an EDMA operation by software polling or when it receives an internal EDMA interrupt. The N9H20 VDMA controller can increment source or destination address, decrement or fixed them as well, and the PDMA can increment source or destination, fixed or wrap around address.

5.14.1 Features

- AMBA AHB master/slave interface compatible, for data transfer and register read/write.
- Support packaging format color space transforms (RGB565, RGB555, RGB888 and YUV422) for VDMA.
- Support stride mode transfer mode for VDMA.
- VDMA support 32-bit source and destination addressing range, address increment, decrement and fixed.
- PDMA support 32-bit source and destination addressing range address increment, fixed and wrap around.
- Support hardware Scatter-Gather function.

5.14.2 Symbol Diagram

The symbol diagram of EDMA controller is shown as following.

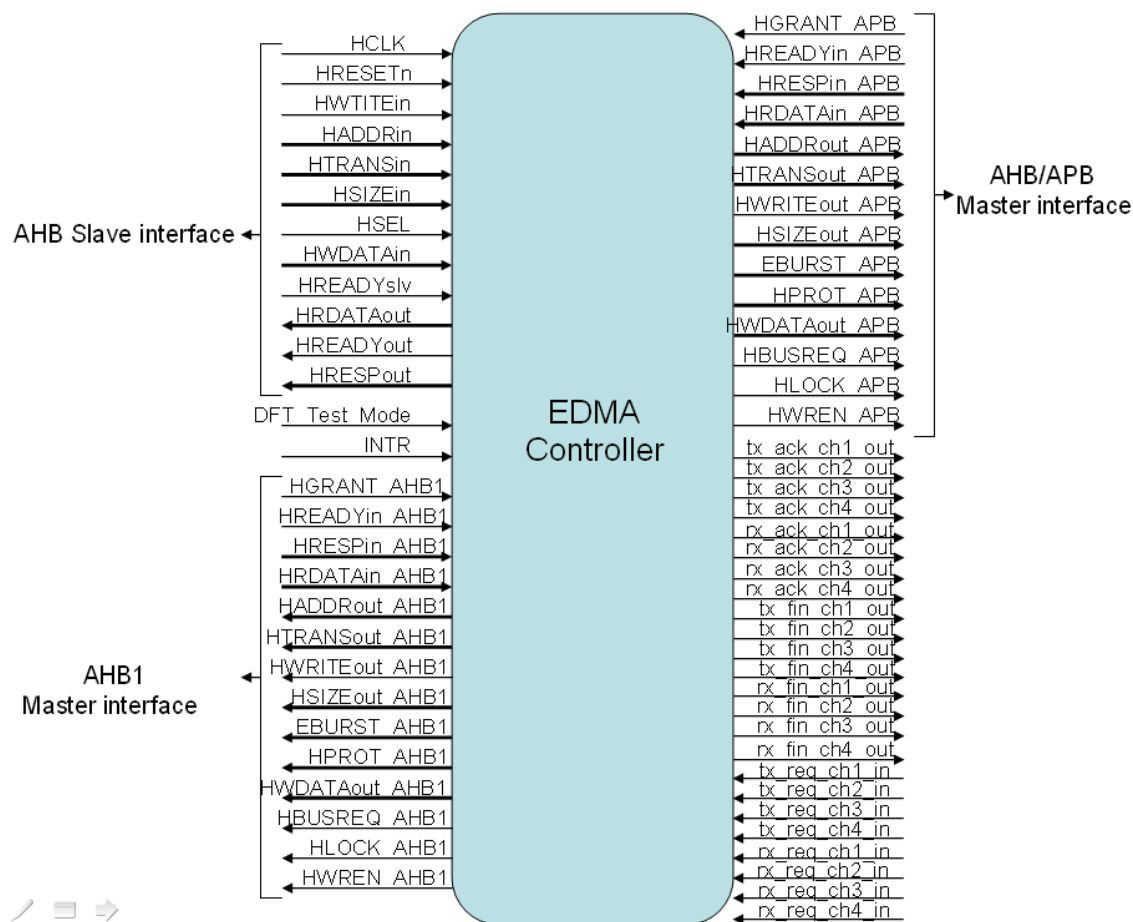


Figure 5.14-1 EDMA Controller Symbol Diagram

5.14.3 Block Diagram

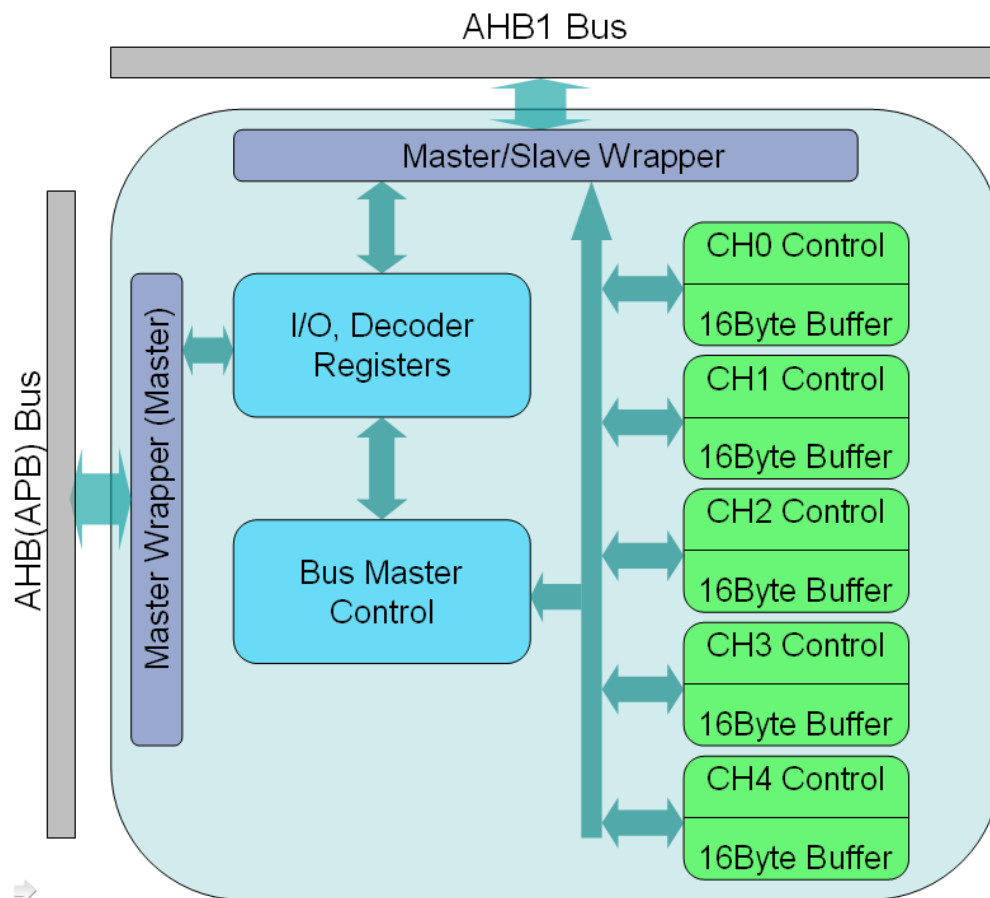


Figure 5.14-2 EDMA Controller Block Diagram

5.14.4 Function Description

The N9H20 contains an enhanced direct memory access (EDMA) controller that transfers data to and from memory or transfer data to and from APB. The EDMA controller has five-channel DMA that include a one channel VDMA (Video-DMA, Memory-to-Memory) and four channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral). For channel0 VDMA mode, it also support color format transform and stripe mode transfer. For PDMA channel (EDMA CH1~CH4), it can transfer data between the Peripherals APB IP (ex: UART, SPI, ADC...) and Memory. The N9H20 also support hardware scatter-gather function, software can set CSRx [SG_EN] to enable scatter-gather function.

Software can stop the EDMA operation by disable DMA [DMACEN]. The CPU can recognize the completion of an EDMA operation by software polling or when it receives an internal EDMA interrupt. The N9H20 VDMA controller can increment source or destination address, decrement or fixed them as well, and the PDMA can increment source or destination, fixed or wrap around address.

Software must enable DMA channel DMA [DMACEN] and then write a valid source address to the DMA_SARx register, a destination address to the DMA_DSABx register, and a transfer count to the DMA_BCRx register. Next, trigger the DMA_CSRx [Trig_EN]. If the source address and destination is not in wrap around mode, the transfer will start transfer until DMA_CBCRx reaches zero (in wrap around mode, when DMA_CBCRx equal zero, the DMA will reload DMA_CBCRx and work around until software disable DMA_CSRx [DMACEN]), If an error occurs during the EDMA operation, the channel stops unless software clears the error condition and sets the DMA_CSRx [SW_RST] to reset the EDMA channel and set EDMA_CSRx [EDMACEN] and [Trig_EN] bits field to start again.

In PDMA (Peripheral-to-Memory or Memory-to-Peripheral) mode, DMA can transfer data between the Peripherals APB IP (ex: UART, SPI, ADC....) and Memory. Each internal peripheral IP that want to do transfer with DMA must have some handshaking signals (see in Figure1).

The PDMA handshaking flow is as flowing

1. Software must set up APB IP and set up each PDMA register (The source and destination address must be word alignment), and then trigger PDMA.
2. APB IP must to check acknowledge signal (ack_sig) to be low and then assert request signal (req_sig) to PDMA to request RX or TX service.
3. When PDMA finish one RX or TX service, PDMA will sent a acknowledge signal (ack_sig) to APB IP.
4. APB IP releases request signal (req_sig) until receive acknowledge signal (ack_sig), when the acknowledge signal (ack_sig) to be low, APB can arrest request signal (req_sig) to request next transfer service.
5. PDMA will send finish signal (fin_sig) to APB when the last data transfer commences.

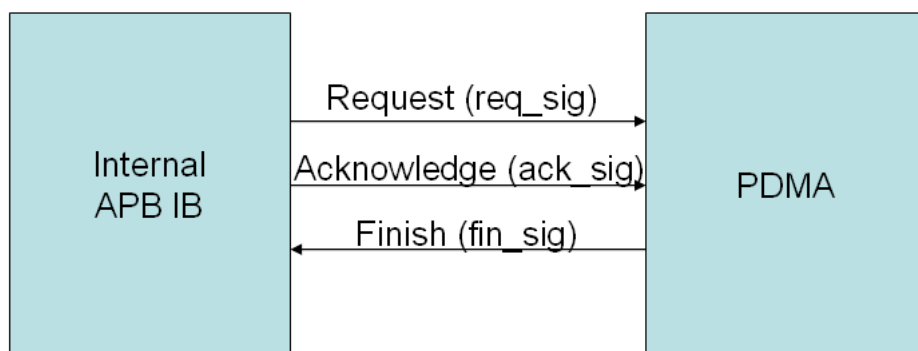


Figure 5.14-3 Peripheral interface signal

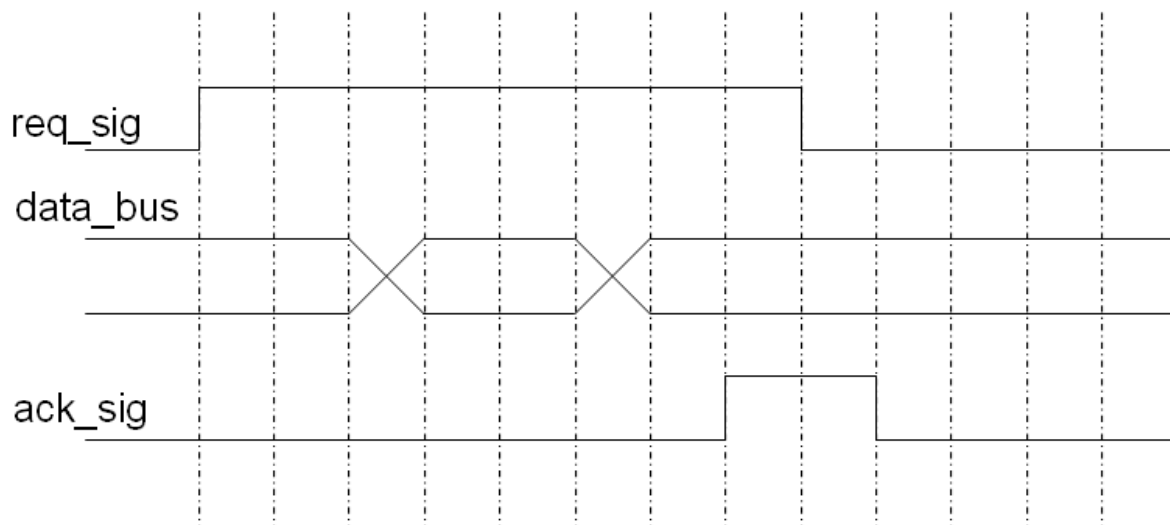
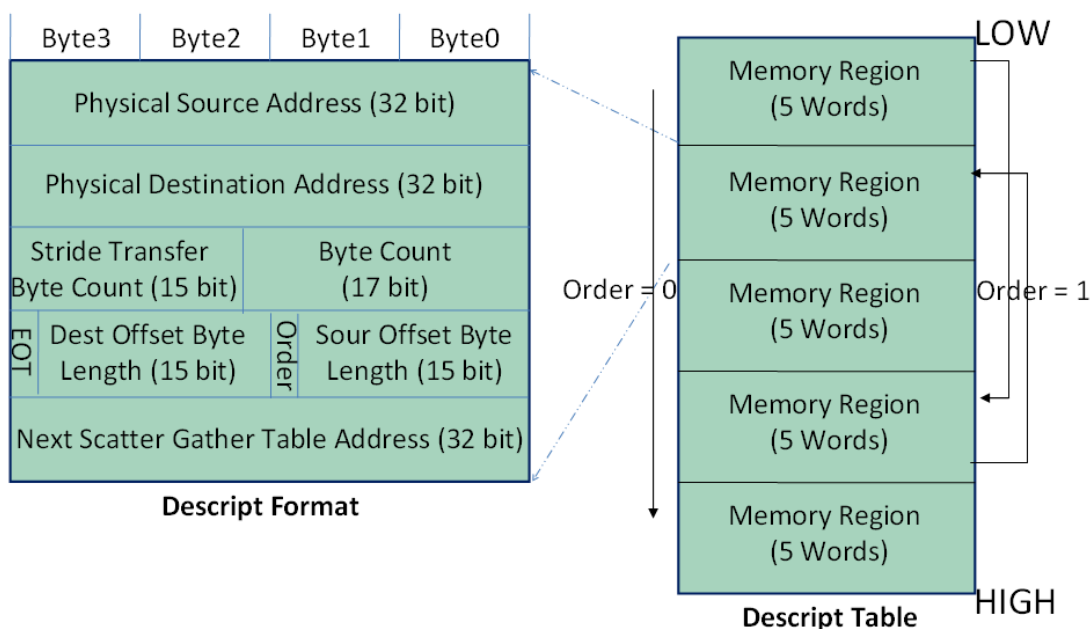


Figure 5.14-2.Handshaking signal

The N9H20 also support hardware scatter-gather function, software can set DMA_CSRx [SG_EN] to enable scatter-gather function. When in scatter-gather function mode, some register will automatically updated by descriptor table. The descriptor table format and program flow is show as following:

Here is a simple example programming flow with DMA Scatter-Gather enable.

1. Set DMA_CSR [DMACEN] and DMA_CSR [SG_EN] to enable DMA and Scatter-Gather function.
2. Setting operation function (ex: memory to memory or memory to APB or color transfer).
3. Fill corresponding starting address of Scatter-Gather descriptor table in DMA_SGAR.
4. Setting [Trig_EN] bits to start transfer.
5. When DMA transfer completed, [Trig_EN] bit will be cleared automatically, and BLKD_IF will be set to 1.



- Physical Source Address (32 bit)
- Physical Destination Address (32 bit)
- Byte Count : Trans byte count (17 bit)
- Stride Transfer Byte Count (15 bit)
- EOT : End of Table (1 bit)
- Source Offset Byte Length (15 bit)
- Order : Scatter Gather table in link list mode or not (1 bit)
- Destination Offset Byte Length (15 bit)
- Next Scatter Gather Table Address (32 bit)

Note : Only when in stride transfer mode (CTCSR [Stride_EN] = 1), Stride Transfer Byte Count, Sour Offset Byte Length and Destination Offset Byte Length is meaningful

Figure 5.14-3. Descriptor Table Format

5.14.5 Register Map

R: read only, W: write only, R/W: both read and write

VDMA_BA = 0xB000_8000				
Register	Offset	R/W	Description	Reset Value
VDMA_CSR	VDMA_BA + 0x00	R/W	VDMA Control Register	0x0000_0000
VDMA_SAR	VDMA_BA + 0x04	R/W	VDMA Source Address Register	0x0000_0000
VDMA_DAR	VDMA_BA + 0x08	R/W	VDMA Destination Address Register	0x0000_0000
VDMA_BCR	VDMA_BA + 0x0C	R/W	VDMA Transfer Byte Count Register	0x0000_0000
VDMA_SGAR	VDMA_BA + 0x10	R/W	VDMA Scatter-Gather Start Address	0x0000_0000
VDMA_CSAR	VDMA_BA + 0x14	R	VDMA Current Source Address Register	0x0000_0000
VDMA_CDAR	VDMA_BA + 0x18	R	VDMA Current Destination Address Register	0x0000_0000
VDMA_CBCR	VDMA_BA + 0x1C	R	VDMA Current Transfer Byte Count Register	0x0000_0000
VDMA_IER	VDMA_BA + 0x20	R/W	VDMA Interrupt Enable Register	0x0000_0001
VDMA_ISR	VDMA_BA + 0x24	R/W	VDMA Interrupt Status Register	0x0000_0000
VDMA_CTCSR	VDMA_BA + 0x28	R/W	VDMA Color Transform Control Register	0x0000_0000
VDMA_SASOCR	VDMA_BA + 0x2C	R/W	VDMA Source Address Stride Offset Control Register	0x0000_0000
VDMA_DASOCR	VDMA_BA + 0x30	R/W	VDMA Destination Address Stride Offset Control Register	0x0000_0000
VDMA_SBUF0	VDMA_BA + 0x80	R/W	VDMA Shared Buffer FIFO 0	0x0000_0000
VDMA_SBUF1	VDMA_BA + 0x84	R/W	VDMA Shared Buffer FIFO 1	0x0000_0000
VDMA_SBUF2	VDMA_BA + 0x88	R/W	VDMA Shared Buffer FIFO 2	0x0000_0000
VDMA_SBUF3	VDMA_BA + 0x8C	R/W	VDMA Shared Buffer FIFO 3	0x0000_0000
PDMA_BA = 0xB000_8100				
PDMA_CSR	PDMA_BA + 0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_SAR	PDMA_BA + 0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_DAR	PDMA_BA + 0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_BCR	PDMA_BA + 0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_SGAR	PDMA_BA + 0x10	R/W	PDMA Scatter-Gather Start Address	0x0000_0000
PDMA_CSAR	PDMA_BA + 0x14	R	PDMA Current Source Address Register	0x0000_0000

PDMA_CDAR	PDMA_BA + 0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CBCR	PDMA_BA + 0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_IER	PDMA_BA + 0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_ISR	PDMA_BA + 0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_POINT	PDMA_BA + 0x3C	R	PDMA Internal buffer pointer	0xFFFF_0000
PDMA_SBUF0	PDMA_BA + 0x80	R/W	PDMA Shared Buffer FIFO 0	0x0000_0000
PDMA_SBUF1	PDMA_BA + 0x84	R/W	PDMA Shared Buffer FIFO 1	0x0000_0000
PDMA_SBUF2	PDMA_BA + 0x88	R/W	PDMA Shared Buffer FIFO 2	0x0000_0000
PDMA_SBUF3	PDMA_BA + 0x8C	R/W	PDMA Shared Buffer FIFO 3	0x0000_0000

5.14.6 Register Description

VDMA Control and Status Register (VDMA_CSR)

Register	Offset	R/W	Description	Reset Value
VDMA_CSR	0x000	R/W	VMAC Control and Status Register (EDMA CH0)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Trig_EN	Reserved						
15	14	13	12	11	10	9	8
Reserved						SG_EN	VDMA_RST
7	6	5	4	3	2	1	0
DAD_SEL		SAD_SEL		Reserved		SW_RST	VDMACEN

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23]	Trig_EN	Trig_EN 0 = No effect. 1 = Enable EDMA data read or write transfer. Note: When EDMA transfer completed, this bit will be cleared automatically. Note: If the bus error occurs, all EDMA transfer will be stopped. Software must reset all EDMA channel (EDMA_RST), and then trig again.
[22:10]	Reserved	Reserved
[9]	SG_EN	EDMA Scatter-Gather Function Enable Enable EDMA scatter-gather function or not. 0 = Normal operation. 1 = Enable scatter-gather operation. Note: When all scatter-gather table transfer completed, this bit will be cleared automatically.
[8]	EDMA_RST	EDMA Software Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the all channels internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after few clock cycles. Note: This bit can reset all channels.
[7:6]	DAD_SEL	Transfer Destination Address Direction Select 00 = Transfer Destination address is incremented successively. 01 = Transfer Destination address is decremented successively.
[5:4]	SAD_SEL	Transfer Source Address Direction Select 00 = Transfer Source address is incremented successively. 01 = Transfer Source address is decremented successively.

[3:2]	Reserved	Reserved
[1]	SW_RST	Software Engine Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after few clock cycles.
[0]	VDMACEN	VDMA Channel Enable Setting this bit to 1 enables VDMA's operation. If this bit is cleared, VDMA will ignore all VDMA request and force Bus Master into IDLE state. Note: SW_RST will clear this bit.

VDMA Transfer Source Address Register (VDMA_SAR)

Register	Offset	R/W	Description	Reset Value
VDMA_SAR	0x004	R/W	VDMA Transfer Source Address Register.	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SAR [31:24]							
23	22	21	20	19	18	17	16
VDMA_SAR [23:16]							
15	14	13	12	11	10	9	8
VDMA_SAR [15:8]							
7	6	5	4	3	2	1	0
VDMA_SAR [7:0]							

Bits	Descriptions	
[31:0]	VDMA_SAR	VDMA Transfer Source Address Register This field indicates a 32-bit source address of VDMA. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful. Note: When in color transfer mode, the source address must be word aligned.

VDMA Transfer Destination Address Register (VDMA_DAR)

Register	Offset	R/W	Description	Reset Value
VDMA_DAR4	0x008	R/W	VDMA Transfer Destination Address Register.	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_DAR [31:24]							
23	22	21	20	19	18	17	16
VDMA_DAR [23:16]							
15	14	13	12	11	10	9	8
VDMA_DAR [15:8]							
7	6	5	4	3	2	1	0
VDMA_DAR [7:0]							

Bits	Descriptions	
[31:0]	VDMA_DAR	VDMA Transfer Destination Address Register This field indicates a 32-bit destination address of VDMA. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful. Note: When in color transfer mode, the destination address must be word aligned.

VDMA Transfer Byte Count Register (VDMA_BCR)

Register	Offset	R/W	Description	Reset Value
VDMA_BCR	0x00C	R/W	VDMA Transfer Byte Count Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
VDMA_BCR [23:16]							
15	14	13	12	11	10	9	8
VDMA_BCR [15:8]							
7	6	5	4	3	2	1	0
VDMA_BCR [7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	VDMA_BCR	VDMA Transfer Byte Count Register This field indicates a 24-bit transfer byte count of VDMA. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful. Note: When in color transfer mode, the transfer byte count must be word aligned.

VDMA Scatter Gather Table Start Address Register (VDMA_SGAR)

Register	Offset	R/W	Description	Reset Value
VDMA_SGAR	0x010	R/W	VDMA Scatter Gather Table Start Address Register.	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SGAR [31:24]							
23	22	21	20	19	18	17	16
VDMA_SGAR [23:16]							
15	14	13	12	11	10	9	8
VDMA_SGAR [15:8]							
7	6	5	4	3	2	1	0
VDMA_SGAR [7:0]							

Bits	Descriptions	
[31:0]	VDMA_SGAR	VDMA Scatter Gather Table Start Address Register This field indicates a 32-bit Scatter Gather Table Start address of VDMA. When in scatter gather mode (DMAx_CSR [SG_EN]), this field is meaningful. Note: The address must be word aligned.

VDMA Current Source Address Register (VDMA_CSAR)

Register	Offset	R/W	Description	Reset Value
VDMA_CSAR4	0x014	R	VDMA Current Source Address Register.	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_CSAR [31:24]							
23	22	21	20	19	18	17	16
VDMA_CSAR [23:16]							
15	14	13	12	11	10	9	8
VDMA_CSAR [15:8]							
7	6	5	4	3	2	1	0
VDMA_CSAR [7:0]							

Bits	Descriptions	
[31:0]	VDMA_CSAR	VDMA Current Source Address Register (Read Only) This field indicates the source address where the VDMA transfer is just occurring.

EDMA Current Destination Address Register (VDMA_CDAR)

Register	Offset	R/W	Description	Reset Value
VDMA_CDAR	0x018	R	VDMA Current Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_CDAR [31:24]							
23	22	21	20	19	18	17	16
VDMA_CDAR [23:16]							
15	14	13	12	11	10	9	8
VDMA_CDAR [15:8]							
7	6	5	4	3	2	1	0
VDMA_CDAR [7:0]							

Bits	Descriptions	
[31:0]	VDMA_CDAR	VDMA Current Destination Address Register (Read Only) This field indicates the destination address where the VDMA transfer is just occurring.

VDMA Current Byte Count Register (VDMA_CBCR)

Register	Offset	R/W	Description	Reset Value
VDMA_CBCR	0x01C	R	VDMA Current Byte Count Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
VDMA_CBCR [23:16]							
15	14	13	12	11	10	9	8
VDMA_CBCR [15:8]							
7	6	5	4	3	2	1	0
VDMA_CBCR [7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	VDMA_CBCR	VDMA Current Byte Count Register (Read Only) This field indicates the current remained byte count of VDMA.

VDMA Interrupt Enable Control Register (VDMA_IER)

Register	Offset	R/W	Description	Reset Value
VDMA_IER	0x020	R/W	VDMA Interrupt Enable Control Register.	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				SG_IEN	Reserved	BLKD_IE	TABORT_IE

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	SG_IEN	VDMA Scatter-Gather Interrupt Enable 0 = Disable scatter-gather interrupt generator. 1 = Enable interrupt generator during every scatter-gather descriptor table transfer done.
[2]	Reserved	Reserved
[1]	BLKD_IE	VDMA Block Transfer Done Interrupt Enable 0 = Disable interrupt generator during VDMA transfer done. 1 = Enable interrupt generator during VDMA transfer done.
[0]	TABORT_IE	VDMA Read/Write Target Abort Interrupt Enable 0 = Disable target abort interrupt generation during VDMA transfer. 1 = Enable target abort interrupt generation during VDMA transfer.

VDMA Interrupt Status Register (VDMA_ISR)

Register	Offset	R/W	Description	Reset Value
VDMA_ISR	0x024	R/W	VDMA Interrupt Status Register.	0x0x0x_0000

31	30	29	28	27	26	25	24
INTR	Reserved		INTR4	INTR3	INTR2	INTR1	INTR0
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Busy	Reserved						
7	6	5	4	3	2	1	0
Reserved				SG_IF	Reserved	BLKD_IF	TABORT_IF

Bits	Descriptions	
[31]	INTR	Interrupt Pin Status (Read Only) This bit is the Interrupt pin status of EDMA controller. Note: This bit is read only
[30:29]	Reserved	Reserved
[28]	INTR4	Interrupt Pin Status of Channel 4 (Read Only) This bit is the Interrupt pin status of EDMA channel4. Note: This bit is read only
[27]	INTR3	Interrupt Pin Status of Channel 3 (Read Only) This bit is the Interrupt pin status of EDMA channel3. Note: This bit is read only
[26]	INTR2	Interrupt Pin Status of Channel 2 (Read Only) This bit is the Interrupt pin status of EDMA channel2. Note: This bit is read only
[25]	INTR1	Interrupt Pin Status of Channel 1 (Read Only) This bit is the Interrupt pin status of EDMA channel1. Note: This bit is read only
[24]	INTR0	Interrupt Pin Status of Channel 0 (Read Only) This bit is the Interrupt pin status of EDMA channel0. Note: This bit is read only
[23:16]	Reserved	Reserved
[15]	Busy	EDMA Transfer is in Progress (Read Only) 0 = EDMA transfer is not in progress. 1 = EDMA transfer is in progress.
[14:4]	Reserved	Reserved

[3]	SG_IF	VDMA Scatter-Gather Interrupt Flag 0 = Scatter-gather descriptor table not finished transfer. 1 = A scatter-gather descriptor table have been transfer done. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[2]	Reserved	Reserved
[1]	BLKD_IF	Block Transfer Done Interrupt Flag This bit indicates that VDMA has finished all data transfer. 0 = Not finished yet. 1 = Done. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[0]	TABORT_IF	EDMA Read/Write Target Abort Interrupt Flag 0 = No bus ERROR response received. 1 = Bus ERROR response received. NOTE: This bit is read only, but can be cleared by writing '1' to it.

NOTE: The VDMA_ISR [TABORT_IF] indicate bus master received ERROR response or not, if bus master received occur it means that target abort is happened. EDMAC will stop transfer and respond this event to software then go to IDLE state. When target abort occurred, software must reset EDMAC, and then transfer those data again.

VDMA Color Transform Control Register (VDMA_CTCSR)

Register	Offset	R/W	Description	Reset Value
VDMA_CTCSR	0x028	R/W	VDMA Color Transform Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				Sour Format			
23	22	21	20	19	18	17	16
Reserved				Dest Format			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Clamping_	Reserved					Col_tra_EN	Stride_

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27:24]	Sour_Format	Source Address Color Format Choose 1000 = The Source format is YCbCr 422 (VYUY) packaging format 0100 = The Source format is RGB565 packaging format 0010 = The Source format is RGB555 packaging format 0001 = The Source format is RGB888 packaging format Note: Software must enable Col_tra_EN and choose source and destination color format to do color transfer, and when in color transfer mode, the VDMA_CAR, VDMA_DAR, VDMA_BCR must be word aligned (if the source format is RGB888, the VDMA_BCR must be 2 word aligned).
[31:28]	Reserved	Reserved
[27:24]	Dest_Format	Destination Address Color Format Choose 1000 = The Destination is YCbCr 422 (VYUY) packaging format 0100 = The Destination is RGB565 packaging format 0010 = The Destination is RGB555 packaging format 0001 = The Destination is RGB888 packaging format Note: Software must enable Col_tra_EN and choose source and destination color format to do color transfer, and when in color transfer mode, the VDMA_CAR, VDMA_DAR, VDMA_BCR must be word aligned.
[15:8]	Reserved	Reserved
[7]	Clamping_EN	Clamping Enable 0 = Disable YCbCr Color Clamping. 1 = Enable YCbCr Color Clamping.
[6:2]	Reserved	Reserved

[1]	Col_tra_EN	Color Transfer Mode Enable 0 = Disable Color transfer enable mode. 1 = Enable Color transfer mode.
[0]	Stride_EN	Stride Mode Enable 0 = Disable stride transfer mode. 1 = Enable stride transfer mode.

VDMA Source Address Stride Offset Control Register (VDMA SASOCR)

Register	Offset	R/W	Description	Reset Value
VDMA_SASOCR	0x02C	R/W	VDMA Source Address Stride Offset Control Register	0x0000_0000

31	30	29	28	27	26	25	24
STBC[15:8]							
23	22	21	20	19	18	17	16
STBC[7:0]							
15	14	13	12	11	10	9	8
SASTOBL[15:8]							
7	6	5	4	3	2	1	0
SASTOBL[7:0]							

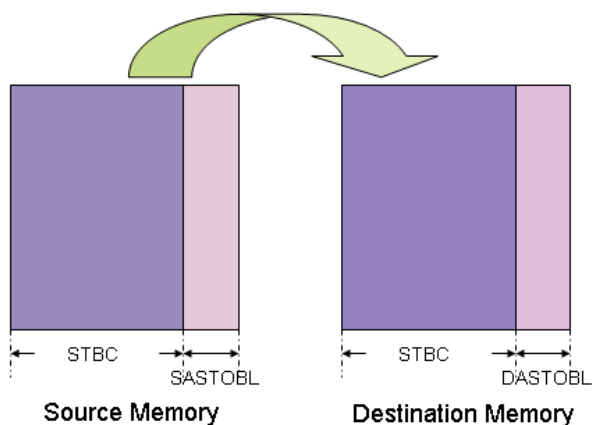
Bits	Descriptions	
[31:16]	STBC	VDMA Stride Transfer Byte Count The 16 bits register define the stride transfer byte count of each row. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful.
[15:0]	SASTOBL	VDMA Source Address Stride Offset Byte Length The 16 bits register define the source address stride transfer offset count of each row. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful.

VDMA Stride Destination Address Offset Control Register (VDMA_DASOCR)

Register	Offset	R/W	Description	Reset Value
VDMA_DASOCR	0x030	R/W	VDMA Destination Address Stride Offset Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DASTOBL[15:8]							
7	6	5	4	3	2	1	0
DASTOBL[7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	DASTOBL	VDMA Destination Address Stride Offset Byte Length The 16 bits register define the destination address stride transfer offset count or each row. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful.



VDMA Shared Buffer FIFO 0 (VDMA_SBUF0)

Register	Offset	R/W	Description	Reset Value
VDMA_SBUF0	0x080	R/W	VDMA Shared Buffer FIFO 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SBUF0 [31:24]							
23	22	21	20	19	18	17	16
VDMA_SBUF0 [23:16]							
15	14	13	12	11	10	9	8
VDMA_SBUF0 [15:8]							
7	6	5	4	3	2	1	0
VDMA_SBUF0 [7:0]							

Bits	Descriptions	
[31:0]	VDMA_SBUF0	VDMA Shared Buffer FIFO 0 Each channel has its own 4 words internal buffer.

VDMA Shared Buffer FIFO 1 (VDMA_SBUF1)

Register	Offset	R/W	Description	Reset Value
VDMA_SBUF1	0x084	R/W	VDMA Shared Buffer FIFO 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SBUF1 [31:24]							
23	22	21	20	19	18	17	16
VDMA_SBUF1 [23:16]							
15	14	13	12	11	10	9	8
VDMA_SBUF1 [15:8]							
7	6	5	4	3	2	1	0
VDMA_SBUF1 [7:0]							

Bits	Descriptions	
[31:0]	VDMA_SBUF1	VDMA Shared Buffer FIFO 1 Each channel has its own 4 words internal buffer.

VDMA Shared Buffer FIFO 2 (VDMA_SBUF2)

Register	Offset	R/W	Description	Reset Value
VDMA_SBUF2	0x088	R/W	VDMA Shared Buffer FIFO 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SBUF2 [31:24]							
23	22	21	20	19	18	17	16
VDMA_SBUF2 [23:16]							
15	14	13	12	11	10	9	8
VDMA_SBUF2 [15:8]							
7	6	5	4	3	2	1	0
VDMA_SBUF2 [7:0]							

Bits	Descriptions	
[31:0]	VDMA_SBUF2	VDMA Shared Buffer FIFO 2 Each channel has its own 4 words internal buffer.

VDMA Shared Buffer FIFO 3 (VDMA_SBUF3)

Register	Offset	R/W	Description	Reset Value
VDMA_SBUF3	0x08C	R/W	VDMA Shared Buffer FIFO 3 Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SBUF3 [31:24]							
23	22	21	20	19	18	17	16
VDMA_SBUF3 [23:16]							
15	14	13	12	11	10	9	8
VDMA_SBUF3 [15:8]							
7	6	5	4	3	2	1	0
VDMA_SBUF3 [7:0]							

Bits	Descriptions	
[31:0]	VDMA_SBUF3	VDMA Shared Buffer FIFO 3 Each channel has its own 4 words internal buffer.

PDMA Control and Status Register (PDMA_CSR)

Register	Offset	R/W	Description	Reset Value
PDMA_CSR1	0x100	R/W	PMAC Control and Status Register CH1	0x0000_0000
PDMA_CSR2	0x200	R/W	PMAC Control and Status Register CH2	0x0000_0000
PDMA_CSR3	0x300	R/W	PMAC Control and Status Register CH3	0x0000_0000
PDMA_CSR4	0x400	R/W	PMAC Control and Status Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Trig_EN	Reserved		APB_TWS		Reserved		
15	14	13	12	11	10	9	8
WAR_BCR_SEL				Reserved		SG_EN	EDMA_RST
7	6	5	4	3	2	1	0
DAD_SEL		SAD_SEL		MODE_SEL		SW_RST	PDMACEN

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23]	Trig_EN	Trig_EN 0 = No effect. 1 = Enable PDMA data read or write transfer. Note: When PDMA transfer completed, this bit will be cleared automatically. Note: If the bus error occurs, all PDMA transfer will be stopped. Software must reset all PDMA channel, and then trig again.
[22:21]	Reserved	Reserved
[20:19]	APB_TWS	Peripheral transfer Width Select 00 = One word (32 bits) is transferred for every PDMA operation. 01 = One byte (8 bits) is transferred for every PDMA operation. 10 = One half-word (16 bits) is transferred for every PDMA operation. 11 = Reserved.
[18:16]	Reserved	Reserved

[15:12]	WAR_BCR_SEL	Wrap around transfer byte count interrupt Select 0001 = Interrupt flag is occurred when the PDMA_CBCR equal 0. 0010 = Interrupt flag is occurred when the PDMA_CBCR equal 3/4 PDMA_BCR. 0100 = Interrupt flag is occurred when the PDMA_CBCR equal 1/2 PDMA_BCR. 1000 = Interrupt flag is occurred when the PDMA_CBCR equal 1/4 PDMA_BCR. 0000 = No Interrupt. Note: Only when the PDMA_CSRx [SAD_SEL] equals 2'b11 or PDMA_CSRx [DAD_SEL] equals to 2'b11, the value in this field is meaningful.
[11:10]	Reserved	Reserved
[9]	SG_EN	PDMA Scatter-Gather Function Enable Enable PDMA scatter-gather function or not. 0 = Normal operation. 1 = Enable scatter-gather operation.
[8]	EDMA_RST	EDMA Software Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the all channels internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after few clock cycles. Note: This bit can reset all channels.
[7:6]	DAD_SEL	Transfer Destination Address Direction Select 00 = Transfer Destination address is incremented successively. 01 = Reserved. 10 = Transfer Destination address is fixed (This feature can be used when data where transferred from multiple sources to a single destination). 11 = Transfer Destination address is wrap around (When the PDMA_CBCR equal zero, the PDMA_CDAR and PDMA_CBCR register will updated by PDMA_DAR and PDMA_BCR automatically. PDMA will start another transfer which without software trigger until PDMA_EN disable. When the PDMA_EN disable, the PDMA will complete the active transfer but the remained data which in the PDMA_BUF will not transfer to destination address).
[5:4]	SAD_SEL	Transfer Source Address Direction Select 00 = Transfer Source address is incremented successively. 01 = Reserved. 10 = Transfer Source address is fixed (This feature can be used when data where transferred from a single source to multiple destinations). 11 = Transfer Source address is wrap around (When the PDMA_CBCR equal zero, the PDMA_CSAR and PDMA_CBCR register will updated by PDMA_SAR and PDMA_BCR automatically. PDMA will start another transfer which without software trigger until PDMA_EN disable. When the PDMA_EN disable, the PDMA will complete the active transfer but the remained data which in the PDMA_BUF will not transfer to destination address).
[3:2]	MODE_SEL	PDMA Mode Select 01 = IP to Memory mode (APB-to-Memory). 10 = Memory to IP mode (Memory-to-APB).
[1]	SW_RST	Software Engine Reset 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after few clock cycles.

[0]	PDMACEN	PDMA Channel Enable Setting this bit to 1 enables PDMA's operation. If this bit is cleared, PDMA will ignore all PDMA request and force Bus Master into IDLE state. Note: SW_RST will clear this bit.

PDMA Transfer Source Address Register (PDMA_SAR)

Register	Offset	R/W	Description	Reset Value
PDMA_SAR1	0x104	R/W	PDMA Transfer Source Address Register CH1	0x0000_0000
PDMA_SAR2	0x204	R/W	PDMA Transfer Source Address Register CH2	0x0000_0000
PDMA_SAR3	0x304	R/W	PDMA Transfer Source Address Register CH3	0x0000_0000
PDMA_SAR4	0x404	R/W	PDMA Transfer Source Address Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SAR [31:24]							
23	22	21	20	19	18	17	16
PDMA_SAR [23:16]							
15	14	13	12	11	10	9	8
PDMA_SAR [15:8]							
7	6	5	4	3	2	1	0
PDMA_SAR [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SAR	PDMA Transfer Source Address Register This field indicates a 32-bit source address of PDMA. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful. Note: When in PDMA mode, the source address must be word aligned.

PDMA Transfer Destination Address Register (PDMA_DAR)

Register	Offset	R/W	Description	Reset Value
PDMA_DAR1	0x108	R/W	PDMA Transfer Destination Address Register CH1	0x0000_0000
PDMA_DAR2	0x208	R/W	PDMA Transfer Destination Address Register CH2	0x0000_0000
PDMA_DAR3	0x308	R/W	PDMA Transfer Destination Address Register CH3	0x0000_0000
PDMA_DAR4	0x408	R/W	PDMA Transfer Destination Address Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_DAR [31:24]							
23	22	21	20	19	18	17	16
PDMA_DAR [23:16]							
15	14	13	12	11	10	9	8
PDMA_DAR [15:8]							
7	6	5	4	3	2	1	0
PDMA_DAR [7:0]							

Bits	Descriptions	
[31:0]	PDMA_DAR	PDMA Transfer Destination Address Register This field indicates a 32-bit destination address of PDMA. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful. Note: When in PDMA mode, the destination address must be word aligned.

PDMA Transfer Byte Count Register (PDMA_BCR)

Register	Offset	R/W	Description	Reset Value
PDMA_BCR1	0x10C	R/W	PDMA Transfer Byte Count Register CH1	0x0000_0000
PDMA_BCR2	0x20C	R/W	PDMA Transfer Byte Count Register CH2	0x0000_0000
PDMA_BCR3	0x30C	R/W	PDMA Transfer Byte Count Register CH3	0x0000_0000
PDMA_BCR4	0x40C	R/W	PDMA Transfer Byte Count Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PDMA_BCR [23:16]							
15	14	13	12	11	10	9	8
PDMA_BCR [15:8]							
7	6	5	4	3	2	1	0
PDMA_BCR [7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	PDMA_BCR	PDMA Transfer Byte Count Register This field indicates a 24-bit transfer byte count of PDMA. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful.

PDMA Scatter Gather Table Start Address Register (PDMA_SGAR)

Register	Offset	R/W	Description	Reset Value
PDMA_SGAR1	0x110	R/W	PDMA Scatter Gather Table Start Address Register CH1.	0x0000_0000
PDMA_SGAR2	0x210	R/W	PDMA Scatter Gather Table Start Address Register CH2.	0x0000_0000
PDMA_SGAR3	0x310	R/W	PDMA Scatter Gather Table Start Address Register CH3.	0x0000_0000
PDMA_SGAR4	0x410	R/W	PDMA Scatter Gather Table Start Address Register CH4.	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SGAR [31:24]							
23	22	21	20	19	18	17	16
PDMA_SGAR [23:16]							
15	14	13	12	11	10	9	8
PDMA_SGAR [15:8]							
7	6	5	4	3	2	1	0
PDMA_SGAR [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SGAR	PDMA Scatter Gather Table Start Address Register This field indicates a 32-bit Scatter Gather Table Start address of PDMA. When in scatter gather mode (DMAx_CSR [SG_EN]), this field is meaningful. Note: The address must be word aligned.

PDMA Current Source Address Register (PDMA_CSAR)

Register	Offset	R/W	Description	Reset Value
PDMA_CSAR1	0x114	R	PDMA Current Source Address Register CH1	0x0000_0000
PDMA_CSAR2	0x214	R	PDMA Current Source Address Register CH2	0x0000_0000
PDMA_CSAR3	0x314	R	PDMA Current Source Address Register CH3	0x0000_0000
PDMA_CSAR4	0x414	R	PDMA Current Source Address Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_CSAR [31:24]							
23	22	21	20	19	18	17	16
PDMA_CSAR [23:16]							
15	14	13	12	11	10	9	8
PDMA_CSAR [15:8]							
7	6	5	4	3	2	1	0
PDMA_CSAR [7:0]							

Bits	Descriptions	
[31:0]	PDMA_CSAR	PDMA Current Source Address Register (Read Only) This field indicates the source address where the PDMA transfer is just occurring.

EDMA Current Destination Address Register (PDMA_CDAR)

Register	Offset	R/W	Description	Reset Value
PDMA_CDAR1	0x118	R	PDMA Current Destination Address Register CH1	0x0000_0000
PDMA_CDAR2	0x218	R	PDMA Current Destination Address Register CH2	0x0000_0000
PDMA_CDAR3	0x318	R	PDMA Current Destination Address Register CH3	0x0000_0000
PDMA_CDAR4	0x418	R	PDMA Current Destination Address Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_CDAR [31:24]							
23	22	21	20	19	18	17	16
PDMA_CDAR [23:16]							
15	14	13	12	11	10	9	8
PDMA_CDAR [15:8]							
7	6	5	4	3	2	1	0
PDMA_CDAR [7:0]							

Bits	Descriptions	
[31:0]	PDMA_CDAR	PDMA Current Destination Address Register (Read Only) This field indicates the destination address where the PDMA transfer is just occurring.

PDMA Current Byte Count Register (PDMA_CBCR)

Register	Offset	R/W	Description	Reset Value
PDMA_CBCR1	0x11C	R	PDMA Current Byte Count Register CH1	0x0000_0000
PDMA_CBCR2	0x21C	R	PDMA Current Byte Count Register CH2	0x0000_0000
PDMA_CBCR3	0x31C	R	PDMA Current Byte Count Register CH3	0x0000_0000
PDMA_CBCR4	0x41C	R	PDMA Current Byte Count Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PDMA_CBCR [23:16]							
15	14	13	12	11	10	9	8
PDMA_CBCR [15:8]							
7	6	5	4	3	2	1	0
PDMA_CBCR [7:0]							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	PDMA_CBCR	PDMA Current Byte Count Register (Read Only) This field indicates the current remained byte count of PDMA.

PDMA Interrupt Enable Control Register (PDMA_IER)

Register	Offset	R/W	Description	Reset Value
PDMA_IER1	0x120	R/W	PDMA Interrupt Enable Control Register CH1	0x0000_0001
PDMA_IER2	0x220	R/W	PDMA Interrupt Enable Control Register CH2	0x0000_0001
PDMA_IER3	0x320	R/W	PDMA Interrupt Enable Control Register CH3	0x0000_0001
PDMA_IER4	0x420	R/W	PDMA Interrupt Enable Control Register CH4	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				SG_IEN	WAR_IE	BLKD_IE	TABORT_IE

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	SG_IEN	PDMA Scatter-Gather Interrupt Enable 0 = Disable scatter-gather interrupt generator. 1 = Enable interrupt generator during every scatter-gather descriptor table transfer done.
[2]	WAR_IE	Wrap Around Interrupt Enable 0 = Disable Wrap around PDMA interrupt generator. 1 = Enable Wrap Around interrupt generator during PDMA transfer done.
[1]	BLKD_IE	PDMA Block Transfer Done Interrupt Enable 0 = Disable interrupt generator during PDMA transfer done. 1 = Enable interrupt generator during PDMA transfer done.
[0]	TABORT_IE	PDMA Read/Write Target Abort Interrupt Enable 0 = Disable target abort interrupt generation during PDMA transfer. 1 = Enable target abort interrupt generation during PDMA transfer.

PDMA Interrupt Status Register (PDMA_ISR)

Register	Offset	R/W	Description	Reset Value
PDMA_ISR1	0x124	R/W	PDMA Interrupt Status Register CH1	0x0x0x_0000
PDMA_ISR2	0x224	R/W	PDMA Interrupt Status Register CH2	0x0x0x_0000
PDMA_ISR3	0x324	R/W	PDMA Interrupt Status Register CH3	0x0x0x_0000
PDMA_ISR4	0x424	R/W	PDMA Interrupt Status Register CH4	0x0x0x_0000

31	30	29	28	27	26	25	24
INTR	Reserved		INTR4	INTR3	INTR2	INTR1	INTR0
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Busy	Reserved			WAR_BCR_IF			
7	6	5	4	3	2	1	0
Reserved				SG_IF	Reserved	BLKD_IF	TABORT_IF

Bits	Descriptions	
[31]	INTR	Interrupt Pin Status (Read Only) This bit is the Interrupt pin status of EDMA controller. NOTE: The bit is read only.
[30:29]	Reserved	Reserved
[28]	INTR4	Interrupt Pin Status of Channel 4 (Read Only) This bit is the Interrupt pin status of EDMA channel4. NOTE: The bit is read only.
[27]	INTR3	Interrupt Pin Status of Channel 3 (Read Only) This bit is the Interrupt pin status of EDMA channel3. NOTE: The bit is read only.
[26]	INTR2	Interrupt Pin Status of Channel 2 (Read Only) This bit is the Interrupt pin status of EDMA channel2. NOTE: The bit is read only.
[25]	INTR1	Interrupt Pin Status of Channel 1 (Read Only) This bit is the Interrupt pin status of EDMA channel1. NOTE: The bit is read only.
[24]	INTR0	Interrupt Pin Status of Channel 0 (Read Only) This bit is the Interrupt pin status of EDMA channel0. NOTE: The bit is read only.
[23:16]	Reserved	Reserved

[15]	Busy	PDMA Transfer is in Progress (Read Only) 0 = PDMA transfer is not in progress. 1 = PDMA transfer is in progress.
[14:12]	Reserved	Reserved
[11:8]	WAR_BCR_IF	Wrap around transfer byte count interrupt flag (Read Only) 0001 = PDMA_CBCR equal 0 flag (Read Only). 0010 = PDMA_CBCR equal 3/4 PDMA_BCR (Read Only). 0100 = PDMA_CBCR equal 1/2 PDMA_BCR (Read Only). 1000 = PDMA_CBCR equal 1/4 PDMA_BCR (Read Only).
[7:4]	Reserved	Reserved
[3]	SG_IF	PDMA Scatter-Gather Interrupt Flag 0 = Scatter-gather descriptor table not finished transfer. 1 = A scatter-gather descriptor table have been transfer done. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[2]	Reserved	Reserved
[1]	BLKD_IF	Block Transfer Done Interrupt Flag This bit indicates that SD host has finished all data-in or data-out block transfer. If there is a CRC-16 error or incorrect CRC status during multiple block data transfer, the transfer will be broken and this bit will also be set. 0 = Not finished yet. 1 = Done. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[0]	TABORT_IF	PDMA Read/Write Target Abort Interrupt Flag 0 = No bus ERROR response received. 1 = Bus ERROR response received. NOTE: This bit is read only, but can be cleared by writing '1' to it.

NOTE: The PDMA_ISR [TABORT_IF] indicate bus master received ERROR response or not, if bus master received occur it means that target abort is happened. PDMAC will stop transfer and respond this event to software then go to IDLE state. When target abort occurred, software must reset PDMAC, and then transfer those data again.

PDMA Internal Buffer Pointer Register (PDMA_POINT)

Register	Offset	R/W	Description	Reset Value
PDMA_POINT1	0x13C	R	PDMA Internal Buffer Pointer Register CH1	0xFFFF_0000
PDMA_POINT2	0x23C	R	PDMA Internal Buffer Pointer Register CH2	0xFFFF_0000
PDMA_POINT3	0x33C	R	PDMA Internal Buffer Pointer Register CH3	0xFFFF_0000
PDMA_POINT4	0x43C	R	PDMA Internal Buffer Pointer Register CH4	0xFFFF_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				PDMA_POINT			

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4:0]	PDMA_POINT	PDMA Internal Buffer Pointer Register (Read Only) This field indicates the internal buffer pointer.

PDMA Shared Buffer FIFO 0 (PDMA_SBUF0_cx)

Register	Offset	R/W	Description	Reset Value
PDMA_SBUF0_c1	0x180	R/W	PDMA Shared Buffer FIFO 0 Register CH1	0x0000_0000
PDMA_SBUF0_c2	0x280	R/W	PDMA Shared Buffer FIFO 0 Register CH2	0x0000_0000
PDMA_SBUF0_c3	0x380	R/W	PDMA Shared Buffer FIFO 0 Register CH3	0x0000_0000
PDMA_SBUF0_c4	0x480	R/W	PDMA Shared Buffer FIFO 0 Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SBUF0 [31:24]							
23	22	21	20	19	18	17	16
PDMA_SBUF0 [23:16]							
15	14	13	12	11	10	9	8
PDMA_SBUF0 [15:8]							
7	6	5	4	3	2	1	0
PDMA_SBUF0 [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SBUF0	PDMA Shared Buffer FIFO 0 Each channel has its own 4 words internal buffer.

PDMA Shared Buffer FIFO 1 (PDMA_SBUF1_cx)

Register	Offset	R/W	Description	Reset Value
PDMA_SBUF1_c1	0x184	R/W	PDMA Shared Buffer FIFO 1 Register CH1	0x0000_0000
PDMA_SBUF1_c2	0x284	R/W	PDMA Shared Buffer FIFO 1 Register CH2	0x0000_0000
PDMA_SBUF1_c3	0x384	R/W	PDMA Shared Buffer FIFO 1 Register CH3	0x0000_0000
PDMA_SBUF1_c4	0x484	R/W	PDMA Shared Buffer FIFO 1 Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SBUF1 [31:24]							
23	22	21	20	19	18	17	16
PDMA_SBUF1 [23:16]							
15	14	13	12	11	10	9	8
PDMA_SBUF1 [15:8]							
7	6	5	4	3	2	1	0
PDMA_SBUF1 [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SBUF1	PDMA Shared Buffer FIFO 1 Each channel has its own 4 words internal buffer.

PDMA Shared Buffer FIFO 2 (PDMA_SBUF2_cx)

Register	Offset	R/W	Description	Reset Value
PDMA_SBUF2_c1	0x188	R/W	PDMA Shared Buffer FIFO 2 Register CH1	0x0000_0000
PDMA_SBUF2_c2	0x288	R/W	PDMA Shared Buffer FIFO 2 Register CH2	0x0000_0000
PDMA_SBUF2_c3	0x388	R/W	PDMA Shared Buffer FIFO 2 Register CH3	0x0000_0000
PDMA_SBUF2_c4	0x488	R/W	PDMA Shared Buffer FIFO 2 Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SBUF2 [31:24]							
23	22	21	20	19	18	17	16
PDMA_SBUF2 [23:16]							
15	14	13	12	11	10	9	8
PDMA_SBUF2 [15:8]							
7	6	5	4	3	2	1	0
PDMA_SBUF2 [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SBUF2	PDMA Shared Buffer FIFO 2 Each channel has its own 4 words internal buffer.

PDMA Shared Buffer FIFO 3 (PDMA_SBUF3_cx)

Register	Offset	R/W	Description	Reset Value
PDMA_SBUF3_c1	0x18C	R/W	PDMA Shared Buffer FIFO 3 Register CH1	0x0000_0000
PDMA_SBUF3_c2	0x28C	R/W	PDMA Shared Buffer FIFO 3 Register CH2	0x0000_0000
PDMA_SBUF3_c3	0x38C	R/W	PDMA Shared Buffer FIFO 3 Register CH3	0x0000_0000
PDMA_SBUF3_c4	0x48C	R/W	PDMA Shared Buffer FIFO 3 Register CH4	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SBUF3 [31:24]							
23	22	21	20	19	18	17	16
PDMA_SBUF3 [23:16]							
15	14	13	12	11	10	9	8
PDMA_SBUF3 [15:8]							
7	6	5	4	3	2	1	0
PDMA_SBUF3 [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SBUF3	PDMA Shared Buffer FIFO 3 Each channel has its own 4 words internal buffer.

5.15 Advanced Interrupt Controller

5.15.1 Overview

An interrupt temporarily changes the execution sequence of a program to react to a particular event such as power failure, watchdog timer timeout, and engine complete, system events, external event trigger and so on. The ARM processor provides two modes of interrupts, the **Fast Interrupt (FIQ)** mode for critical session and the **Interrupt (IRQ)** mode for general purpose. The IRQ exception mode is occurred when the NIRQ input is asserted. Similarly, the FIQ exception mode is occurred when the NFIQ input is asserted. The FIQ mode has privilege over the IRQ mode and can preempt an ongoing IRQ mode. It is possible to ignore the NFIQ and the NIRQ by setting the F-bit and I-bit in the **current program status register (CPSR)**.

The N9H20 incorporates the **advanced interrupt controller (AIC)** that is capable of dealing with the interrupt requests from different sources. Each interrupt source is uniquely assigned to an interrupt channel. For example, the watchdog timer interrupt is assigned to channel 2. The AIC implements a proprietary eight-level priority scheme that differentiates the available interrupt sources into eight priority levels. Interrupt sources within the priority level 0 have the highest priority and the priority level 7 has the lowest. To work this scheme properly, you must specify a certain priority level to each interrupt source during power-on initialization; otherwise, the system shall behave unexpectedly. Within each priority level, interrupt source that is positioned in a lower channel has a higher priority. Interrupt source that is active, enabled, and positioned in the lowest channel within the priority level 0 is promoted to the FIQ mode. Interrupt sources within the priority levels other than 0 can petition for the IRQ mode. The IRQ mode can be preempted by the occurrence of the FIQ mode. Interrupt nesting is performed automatically by the AIC. A higher priority interrupt source will cause the NIRQ to CPU be asserted again when CPU is servicing a lower priority interrupt if the I-bit in CPSR is enabled.

Though interrupt sources originated from the N9H20 itself are intrinsically high-level sensitive, the AIC can be configured as either low-level sensitive, high-level sensitive, negative-edge triggered, or positive-edge triggered to each interrupt source. When the N9H20 is put in the test mode, all interrupt sources must be configured as positive-edge triggered.

5.15.2 Features

- AMBA APB bus interface and Individual mask for each interrupt source
- External interrupts can be programmed as either edge-triggered or level-sensitive
- External interrupts can be programmed as either low-active or high-active
- Has flags to reflect the status of each interrupt source
- Proprietary 8-level interrupt scheme to ease the burden from the interrupt
- Daisy-chain priority mechanism is applied to interrupts set as the same priority level.
- Automatically masking out the lower priority interrupt during interrupt nesting
- Automatically clearing the interrupt flag when the external interrupt source is programmed to be edge-triggered

5.15.3 Interrupt Sources

The following table lists all interrupts from various peripheral interface modules or external devices.

Channel	Name	SCR	Source	Reset (Default) Level
1	WDT_INT	SCR1[15:8]	Watch Dog Timer Interrupt	Low
2	GPIO_INT0	SCR1[23:16]	GPIO Interrupt 0	Low
3	GPIO_INT1	SCR1[31:24]	GPIO Interrupt 1	Low
4	GPIO_INT2	SCR2[7:0]	GPIO Interrupt 2	Low
5	GPIO_INT3	SCR2[15:8]	GPIO Interrupt 3	Low
6	SPU_INT	SCR2[23:16]	SPU Interrupt	Low
7	I2S_INT	SCR2[31:24]	I2S Controller Interrupt	Low
8	VPOST_INT	SCR3[7:0]	VPOST Interrupt	Low
9	CAP_INT	SCR3[15:8]	Capture Interrupt	Low
10	Reserved	Reserved	Reserved	Reserved
11	BLT_INT	SCR3[31:24]	BitBLT Accelerator Interrupt	Low
12	FSC_INT	SCR4[7:0]	Frame Switch Controller Interrupt	Low
13	HUART_INT	SCR4[15:8]	High Speed UART Interrupt	Low
14	TMR0_INT	SCR4[23:16]	Timer 0 Interrupt	Low
15	TMR1_INT	SCR4[31:24]	Timer 1 Interrupt	Low
16	UDC_INT	SCR5[7:0]	USB Device Controller Interrupt	Low
17	SIC_INT	SCR5[15:8]	Storage Interface Controller Interrupt	Low
18	UHC_INT	SCR5[23:16]	USB Host Controller Interrupt	Low
19	EDMA_INT	SCR5[31:24]	Enhanced DMA Interrupt	Low
20	SPIMS0_INT	SCR6[7:0]	SPI Master/Slave Controller 0 Interrupt	Low
21	SPIMS1_INT	SCR6[15:8]	SPI Master/Slave Controller 1 Interrupt	Low
22	ADC_INT	SCR6[23:16]	ADC Interrupt	Low
23	RTC_INT	SCR6[31:24]	RTC Interrupt	Low
24	UART_INT	SCR7[7:0]	UART Interrupt	Low
25	PWM_INT	SCR7[15:8]	PWM Interrupt	Low
26	JPG_INT	SCR7[23:16]	JPEG Codec Interrupt	Low
27	Reserved	Reserved	Reserved	Reserved
28	KPI	SCR8[7:0]	Keypad Interrupt	Low
29	Reserved	SCR8[15:8]	Reserved	Reserved
30	I2C_INT	SCR8[23:16]	I2C Interrupt	Low
31	PWR_INT	SCR8[31:24]	System Wake-Up Interrupt	Low

31	30	29	28	27	26	25	24
SCR8[31:24]	SCR8[23:16]	SCR8[15:8]	SCR8[7:0]	SCR7[31:24]	SCR7[23:16]	SCR7[15:8]	SCR7[7:0]
PWR	I2C	---	KPI	---	JPEG	PWM	UART
23	22	21	20	19	18	17	16
SCR6[31:24]	SCR6[23:16]	SCR6[15:8]	SCR6[7:0]	SCR5[31:24]	SCR5[23:16]	SCR5[15:8]	SCR5[7:0]
RTC	ADC	SPIMS1	SPIMS0	EDMA	UHC	SIC	UDC
15	14	13	12	11	10	9	8
SCR4[31:24]	SCR4[23:16]	SCR4[15:8]	SCR4[7:0]	SCR3[31:24]	SCR3[23:16]	SCR3[15:8]	SCR3[7:0]
TMR1	TMR0	HUART	FSC	BLT	---	CAP	VPOST
7	6	5	4	3	2	1	0
SCR2[31:24]	SCR2[23:16]	SCR2[15:8]	SCR2[7:0]	SCR1[31:24]	SCR1[23:16]	SCR1[15:8]	SCR1[7:0]
I2S	SPU	GPIO3	GPIO2	GPIO1	GPIO0	WDT	---

5.15.4 AIC Block Diagram

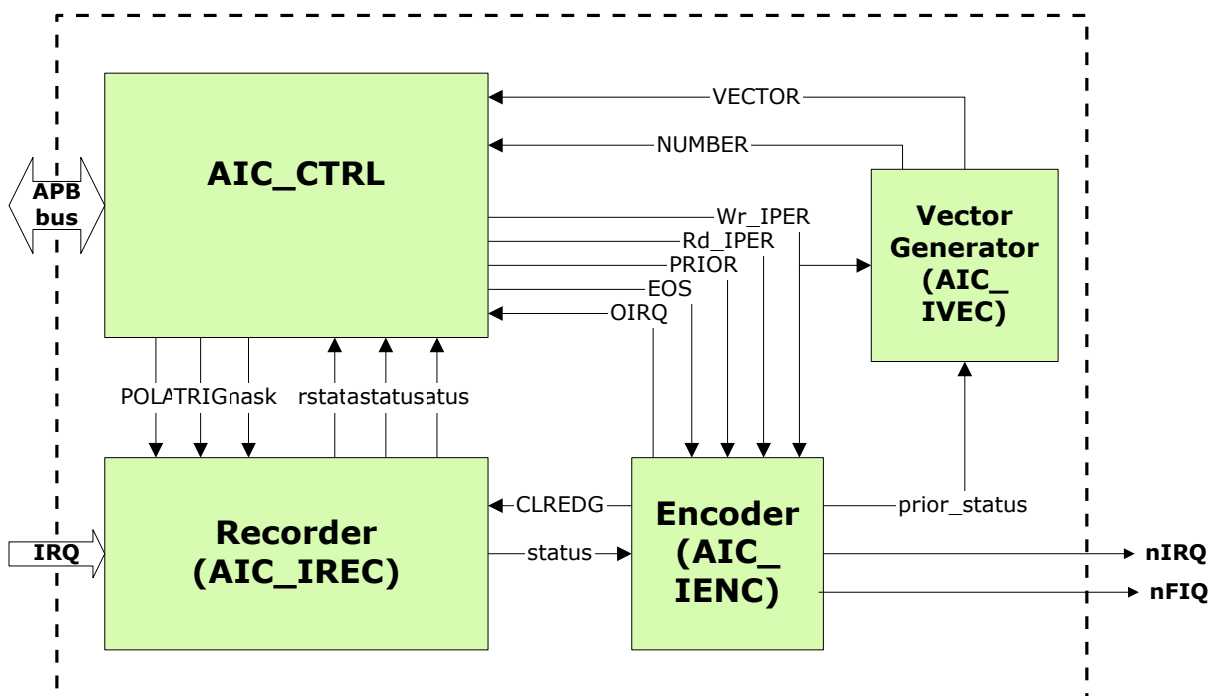


Figure 5.15-1 AIC Functional Block Diagram

5.15.5 AIC Functional Descriptions

Hardware Interrupt Vectoring

The hardware interrupt vectoring can be used to shorten the interrupt latency. If not used, priority determination must be carried out by software. When the Interrupt Priority Encoding Register (AIC_IPER) is read, it will return an integer representing the channel that is active and having the highest priority. This integer is equivalent to multiplied by 4 (shifted left two bits to word-align it) such that it may be used directly to index into a branch table to select the appropriate interrupt service routine vector.

Priority Controller

An 8-level priority encoder controls the NIRQ and NFIQ line. Each interrupt source belongs to priority group between of 0 to 7. Group 0 has the highest priority and group 7 the lowest. Group 0 means FIQ mode group. When more than one unmasked interrupt channels are active at a time, the interrupt with the highest priority is serviced first. If all active interrupts have equal priority, the interrupt with the lowest interrupt source number is serviced first.

The current priority level is defined as the priority level of the interrupt with the highest priority at the time the register AIC_IPER is read. In the case when a higher priority unmasked interrupt occurs while an interrupt already exits, there are two possible outcomes depending on whether the AIC_IPER has been read.

If the processor has already read the AIC_IPER and caused the NIRQ line to be de-asserted, then the NIRQ line is reasserted. When the processor has enabled nested interrupts and reads the AIC_IPER again, it reads the new, higher priority interrupt vector. At the same time, the current priority level is updated to the higher priority.

If the AIC_IPER has not been read after the NIRQ line has been asserted, then the processor will read

the new higher priority interrupt vector in the AIC_IPER register and the current priority level is updated.

When the End of Service Command Register (AIC_EOSCR) is written, the current interrupt level is updated with the last stored interrupt level from the stack (if any). Therefore, at the end of a higher priority interrupt, the AIC returns to the previous state corresponding to the preceding lower priority interrupt which had been interrupted.

Interrupt Handling

When the NIRQ line is asserted, the interrupt handler must read the AIC_IPER as soon as possible. This can de-assert the NIRQ request to the processor and clears the interrupt if it is programmed to be edge triggered. This allows the AIC to assert the NIRQ line again when a higher priority unmasked interrupt occurs.

The AIC_EOSCR (End of Service Command Register) must be written at the end of the interrupt service routine. This permits pending interrupts to be serviced.

Interrupt Masking

Each interrupt source can be enabled or disabled individually by using the command registers AIC_MECR and AIC_MDCR. The status of interrupt mask can be read in the read only register AIC_IMR. A disabled interrupt doesn't affect the servicing of other interrupts.

Interrupt Clearing and Setting

All interrupt sources can be individually set or clear by respectively writing to the registers AIC_SSCR and AIC_SCCR when they are programmed to be edge triggered. This feature of the AIC is useful in auto-testing or software debugging.

Fake Interrupt

When the AIC asserts the NIRQ line, the processor enters interrupt mode and the interrupt handler reads the AIC_IPER, it may happen that interrupt sources de-assert NIRQ lines after the processor has taken into account the NIRQ assertion and before the read of the AIC_IPER.

This behavior is called a fake interrupt.

The AIC is able to detect these fake interrupts and returns all zero when AIC_IPER is read. The same mechanism of fake interrupt occurs if the processor reads the AIC_IPER (application software or ICE) when there is no interrupting pending. The current priority level is not updated in this situation. Hence, the AIC_EOSCR shouldn't be written.

ICE/Debug Mode

This mode allows reading of the AIC_IPER without performing the associated automatic operations. This is necessary when working with a debug system. When an ICE or debug monitor reads the AIC user interface, the AIC_IPER can be read. This has the following consequences in normal mode:

If there is no enabled pending interrupt, the fake vector will be returned.

If an enabled interrupt with a higher priority than the current one is pending, it will be stacked.

In the second case, an End-of-Service command would be necessary to restore the state of the AIC. This operation is generally not performed by the debug system. Therefore, the debug system would become strongly intrusive, and could cause the application to enter an undesired state.

This can be avoided by using ICE/Debug Mode. When this mode is enabled, the AIC performs interrupt stacking only when a write access is performed on the AIC_IPER. Hence, the interrupt service routine must write to the AIC_IPER (any value) just after reading it. When AIC_IPER is written, the new status of AIC, including the value of interrupt source number register (AIC_ISNR), is updated with the value that is kept at previous reading of AIC_IPER, the debug system must not write to the AIC_IPER as this would cause undesirable effects.

The following table shows the main steps of an interrupt and the order in which they are performed according to the mode:

Action	Normal Mode	ICE/Debug Mode
Calculate active interrupt	Read AIC_IPER	Read AIC_IPER
Determine and return the vector of the active interrupt	Read AIC_IPER	Read AIC_IPER
Push on internal stack the current priority level	Read AIC_IPER	Write AIC_IPER
Acknowledge the interrupt (Note 1)	Read AIC_IPER	Write AIC_IPER
No effect (Note 2)	Read AIC_IPER	

Notes:

NIRQ de-assertion and automatic interrupt clearing if the source is programmed as level sensitive.

Note that software which has been written and debugged using this mode will run correctly in normal mode without modification. However, in normal mode writing to AIC_IPER has no effect and can be removed to optimize the code.

5.15.6 AIC Registers Map

Register	Address	R/W	Description	Reset Value
AIC_BA = 0xB800_0000				
AIC_SCR1	AIC_BA+000	R/W	Source Control Register 1	0x4747_4747
AIC_SCR2	AIC_BA+004	R/W	Source Control Register 2	0x4747_4747
AIC_SCR3	AIC_BA+008	R/W	Source Control Register 3	0x4747_4747
AIC_SCR4	AIC_BA+00C	R/W	Source Control Register 4	0x4747_4747
AIC_SCR5	AIC_BA+010	R/W	Source Control Register 5	0x4747_4747
AIC_SCR6	AIC_BA+014	R/W	Source Control Register 6	0x4747_4747
AIC_SCR7	AIC_BA+018	R/W	Source Control Register 7	0x4747_4747
AIC_SCR8	AIC_BA+01C	R/W	Source Control Register 8	0x4747_4747
Reserved	Reserved		Reserved	Undefined
AIC_IRSR	AIC_BA+100	R	Interrupt Raw Status Register	0x0000_0000
AIC_IASR	AIC_BA+104	R	Interrupt Active Status Register	0x0000_0000
AIC_ISR	AIC_BA+108	R	Interrupt Status Register	0x0000_0000
AIC_IPER	AIC_BA+10C	R	Interrupt Priority Encoding Register	0x0000_0000
AIC_ISNR	AIC_BA+110	R	Interrupt Source Number Register	0x0000_0000
AIC_IMR	AIC_BA+114	R	Interrupt Mask Register	0x0000_0000
AIC_OISR	AIC_BA+118	R	Output Interrupt Status Register	0x0000_0000
Reserved	Reserved		Reserved	Undefined
AIC_MECR	AIC_BA+120	W	Mask Enable Command Register	Undefined
AIC_MDCR	AIC_BA+124	W	Mask Disable Command Register	Undefined
AIC_SSCR	AIC_BA+128	W	Source Set Command Register	Undefined
AIC_SCCR	AIC_BA+12C	W	Source Clear Command Register	Undefined
AIC_EOSCR	AIC_BA+130	W	End of Service Command Register	Undefined
AIC_TEST	AIC_BA+134	W/R	ICE/Debug mode Register	0x0000_0000
Reserved	Reserved		Reserved	Undefined

5.15.7 AIC Control Registers

AIC Source Control Registers (AIC_SCR1 ~ AIC_SCR8)

Register	Address	R/W/C	Description	Reset Value
AIC_SCR1 ~ AIC_SCR8	AIC_BA+000 ~ AIC_BA+01C	R/W	Source Control Register 1 ~ Source Control Register 8	0x4747_4747

31	30	29	28	27	26	25	24
TYPE (Channel 3)		Reserved			PRIORITY (Channel 3)		
23	22	21	20	19	18	17	16
TYPE (Channel 2)		Reserved			PRIORITY (Channel 2)		
15	14	13	12	11	10	9	8
TYPE (Channel 1)		Reserved			PRIORITY (Channel 1)		
7	6	5	4	3	2	1	0
TYPE (channel 0)		Reserved			PRIORITY (Channel 0)		

Bits	Descriptions	
[7:6]	TYPE	Interrupt Type TYPE [7] indicates the level (0)/edge (1) triggered. TYPE [6] indicates the low (0)/high (1) level. 00: low-active level triggered 01: high-active level triggered 10: low-active edge triggered 11: high-active edge triggered Interrupts other than INT_EXT can be configured as level triggered during normal operation unless in the test mode.
[2:0]	PRIORITY	Priority Level (0 – 7) The level 0 indicates the highest priority and the level 7 indicates the lowest priority. An interrupt is treated as a FIQ mode for the priority level 0, and is treated as an IRQ mode for other levels. If two or more interrupts have the identical priority level, the interrupts located in the upper rows of the interrupt source table, have higher priorities.

AIC Interrupt Raw Status Register (AIC_IRSR)

Register	Address	R/W	Description	Reset Value
AIC_IRSR	AIC_BA+100	R	Interrupt Raw Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IRS[31:24]							
23	22	21	20	19	18	17	16
IRS[23:16]							
15	14	13	12	11	10	9	8
IRS[15:8]							
7	6	5	4	3	2	1	0
IRS[7:0]							

This register records the intrinsic state within each interrupt channel.

Bits	Descriptions	
[31:0]	IRS x	Interrupt Status Indicate the intrinsic status of the corresponding interrupt source 0 = Interrupt channel is in the voltage level 0 1 = Interrupt channel is in the voltage level 1

AIC Interrupt Active Status Register (AIC_IASR)

Register	Address	R/W	Description	Reset Value
AIC_IASR	AIC_BA+104	R	Interrupt Active Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IAS[31:24]							
23	22	21	20	19	18	17	16
IAS[23:16]							
15	14	13	12	11	10	9	8
IAS[15:8]							
7	6	5	4	3	2	1	0
IAS[7:0]							

This register indicates the status of each interrupt channel in consideration of the interrupt source type as defined in the corresponding Source Control Register, but regardless of its mask setting.

Bits	Descriptions	
[31:0]	IAS x	Interrupt Active Status Indicate the status of the corresponding interrupt source 0 = Corresponding interrupt channel is inactive 1 = Corresponding interrupt channel is active

AIC Interrupt Status Register (AIC_ISR)

Register	Address	R/W	Description	Reset Value
AIC_ISR	AIC_BA+108	R	Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
ISR[31:24]							
23	22	21	20	19	18	17	16
ISR[23:16]							
15	14	13	12	11	10	9	8
ISR[15:8]							
7	6	5	4	3	2	1	0
ISR[7:0]							

This register identifies those interrupt channels whose are both active and enabled.

Bits	Descriptions	
[31:0]	ISRx:	Interrupt Status Register Indicates the status of corresponding interrupt channel 0 = Two possibilities: (a)The corresponding interrupt channel is inactive no matter whether it is enabled or disabled (b)It is active but not enabled 1 = Corresponding interrupt channel is both active and enabled (can assert an interrupt)

AIC IRQ Priority Encoding Register (AIC_IPER)

Register	Address	R/W	Description	Reset Value
AIC_IPER	AIC_BA+10C	R	Interrupt Priority Encoding Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	VECTOR					Reserved	

When the AIC generates the interrupt, **VECTOR** represents the interrupt channel number that is active, enabled, and has the highest priority. If the representing interrupt channel possesses a priority level 0, then the interrupt asserted is FIQ mode; otherwise, it is IRQ mode. The value of **VECTOR** is copied to the register AIC_ISNR thereafter by the AIC. This register is restored a value 0 after it was read by the interrupt handler. This register can help indexing into a branch table to quickly jump to the corresponding interrupt service routine. The reserved bits are set to zero.

Bits	Descriptions	
[6:2]	VECTOR	Interrupt Vector 0 = no interrupt occurs 1 ~ 31 = representing the interrupt channel that is active, enabled, and having the highest priority

AIC Interrupt Source Number Register (AIC_ISNR)

Register	Address	R/W	Description	Reset Value
AIC_ISNR	AIC_BA+110	R	Interrupt Source Number Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				IRQID			

The purpose of this register is to record the interrupt channel number that is active, enabled, and has the highest priority. The reserved bits are set to zero.

Bits	Descriptions	
[4:0]	IRQID	IRQ Identification Stands for the interrupt channel number

AIC Interrupt Mask Register (AIC_IMR)

Register	Address	R/W	Description	Reset Value
AIC_IMR	AIC_BA+114	R	Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
IM[31:24]							
23	22	21	20	19	18	17	16
IM[23:16]							
15	14	13	12	11	10	9	8
IM[15:8]							
7	6	5	4	3	2	1	0
IM [7:0]							

Bits	Descriptions	
[31:0]	IM x	Interrupt Mask This bit determines whether the corresponding interrupt channel is enabled or disabled. Every interrupt channel can be active no matter whether it is enabled or disabled. If an interrupt channel is enabled, it does not definitely mean it is active. Every interrupt channel can be authorized by the AIC only when it is both active and enabled. 0 = Corresponding interrupt channel is disabled 1 = Corresponding interrupt channel is enabled

AIC Output Interrupt Status Register (AIC_OISR)

Register	Address	R/W	Description	Reset Value
AIC_OISR	AIC_BA+118	R	Output Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						IRQ	FIQ

The AIC classifies the interrupt into FIQ mode and IRQ mode. This register indicates whether the asserted interrupt is NFIQ or NIRQ. If both NIRQ and NFIQ are equal to 0, it means there is no interrupt occurred.

Bits	Descriptions	
[1]	IRQ	Interrupt Request 0 = NIRQ line is inactive. 1 = NIRQ line is active.
[0]	FIQ	Fast Interrupt Request 0 = NFIQ line is inactive. 1 = NFIQ line is active

AIC Mask Enable Command Register (AIC_MECR)

Register	Address	R/W	Description	Reset Value
AIC_MECR	AIC_BA+120	W	Mask Enable Command Register	Undefined

31	30	29	28	27	26	25	24
MEC[31:24]							
23	22	21	20	19	18	17	16
MEC[23:16]							
15	14	13	12	11	10	9	8
MEC[15:8]							
7	6	5	4	3	2	1	0
MEC[7:0]							

Bits	Descriptions	
[31:0]	MEC x	Mask Enable Command 0 = No effect 1 = Enables the corresponding interrupt channel

AIC Mask Disable Command Register (AIC_MDCR)

Register	Address	R/W	Description	Reset Value
AIC_MDCR	AIC_BA+124	W	Mask Disable Command Register	Undefined

31	30	29	28	27	26	25	24
MDC[31:24]							
23	22	21	20	19	18	17	16
MDC[23:16]							
15	14	13	12	11	10	9	8
MDC[15:8]							
7	6	5	4	3	2	1	0
MDC[7:0]							

Bits	Descriptions	
[31:0]	MDC x	Mask Disable Command 0 = No effect 1 = Disables the corresponding interrupt channel

AIC Source Set Command Register (AIC_SSCR)

Register	Address	R/W	Description	Reset Value
AIC_SSCR	AIC_BA+128	W	Source Set Command Register	Undefined

31	30	29	28	27	26	25	24
SSC[31:24]							
23	22	21	20	19	18	17	16
SSC[23:16]							
15	14	13	12	11	10	9	8
SSC[15:8]							
7	6	5	4	3	2	1	0
SSC[7:0]							

When the N9H20 is under debugging or verification, software can activate any interrupt channel by setting the corresponding bit in this register. This feature is useful in hardware verification or software debugging.

Bits	Descriptions	
[31:0]	SSCx	Source Set Command 0 = No effect. 1 = Activates the corresponding interrupt channel

AIC Source Clear Command Register (AIC_SCCR)

Register	Address	R/W	Description	Reset Value
AIC_SCCR	AIC_BA+12C	W	Source Clear Command Register	Undefined

31	30	29	28	27	26	25	24
SCC[31:24]							
23	22	21	20	19	18	17	16
SCC[23:16]							
15	14	13	12	11	10	9	8
SCC[15:8]							
7	6	5	4	3	2	1	0
SCC[7:0]							

When the N9H20 is under debugging or verification, software can deactivate any interrupt channel by setting the corresponding bit in this register. This feature is useful in hardware verification or software debugging.

Bits	Descriptions	
[31:0]	SCCx	Source Clear Command 0 = No effect. 1 = Deactivates the corresponding interrupt channels

AIC End of Service Command Register (AIC_EOSCR)

Register	Address	R/W	Description	Reset Value
AIC_EOSCR	AIC_BA+130	W	End of Service Command Register	Undefined

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	---
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---
7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This register is used by the interrupt service routine to indicate that it is completely served. Thus, the interrupt handler can write any value to this register to indicate the end of its interrupt service.

AIC ICE/Debug Register (AIC_TEST)

Register	Address	R/W	Description	Reset Value
AIC_TEST	AIC_BA+134	W	ICE/Debug mode Register	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							Test

This register indicates whether AIC_IPER will be cleared or not after been read. If bit0 of AIC_TEST has been set, ICE or debug monitor can read AIC_IPER for verification and the AIC_IPER will not be cleared automatically. Write access to the AIC_IPER will perform the interrupt stacking in this mode.

Bits	Descriptions	
[0]	TEST	ICE/Debug mode 0 = normal mode. 1 = ICE/Debug mode.

5.16 General Purpose I/O

5.16.1 Overview and Features

80 pins of General Purpose I/O are shared with special feature functions.

Supported Features of these I/O are: input or output facilities, pull-up resistors.

All these general purpose I/O functions are achieved by software programming setting and I/O cells selected from TSMC universal standard I/O Cell Library. And the following figures illustrate the control mechanism to achieve the GPIO functions.

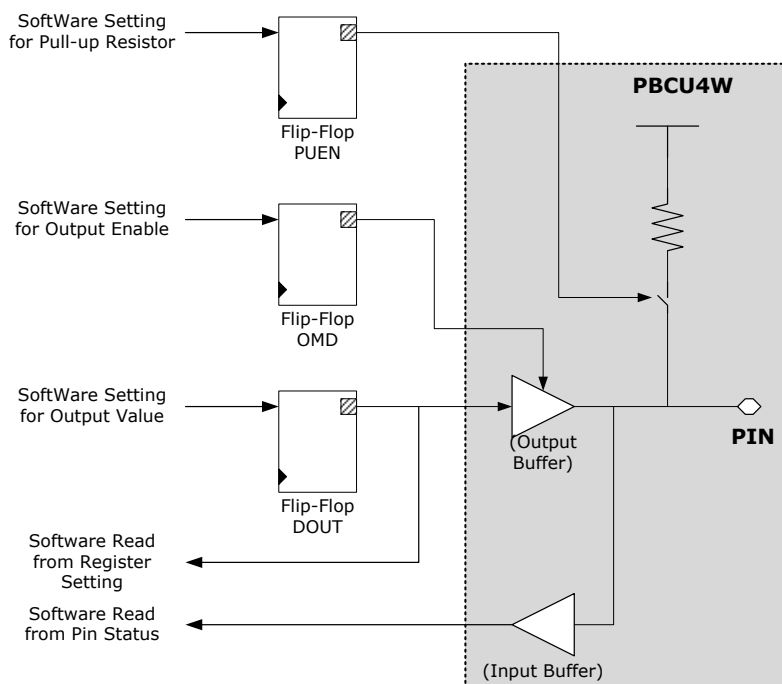


Figure 5.16-1 Type I GPIO: Input/Output Port with Program Controlled Weakly Pull-High

5.16.2 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Address	R/W	Description	Reset Value
GP_BA = 0xB8001000				
GPIOA_OMD	GP_BA+0x00	R/W	GPIO Port A Bit Output Mode Enable	0x0000_0000
GPIOA_PUEN	GP_BA+0x04	R/W	GPIO Port A Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOA_DOUT	GP_BA+0x08	R/W	GPIO Port A Data Output Value	0x0000_0000
GPIOA_PIN	GP_BA+0x0C	R	GPIO Port A Pin Value	0XXXXX_XXXX
GPIOB_OMD	GP_BA+0x10	R/W	GPIO Port B Bit Output Mode Enable	0x0000_0000
GPIOB_PUEN	GP_BA+0x14	R/W	GPIO Port B Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOB_DOUT	GP_BA+0x18	R/W	GPIO Port B Data Output Value	0x0000_0000
GPIOB_PIN	GP_BA+0x1C	R	GPIO Port B Pin Value	0XXXXX_XXXX
GPIOC_OMD	GP_BA+0x20	R/W	GPIO Port C Bit Output Mode Enable	0x0000_0000
GPIOC_PUEN	GP_BA+0x24	R/W	GPIO Port C Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOC_DOUT	GP_BA+0x28	R/W	GPIO Port C Data Output Value	0x0000_0000
GPIOC_PIN	GP_BA+0x2C	R	GPIO Port C Pin Value	0XXXXX_XXXX
GPIOD_OMD	GP_BA+0x30	R/W	GPIO Port D Bit Output Mode Enable	0x0000_0000
GPIOD_PUEN	GP_BA+0x34	R/W	GPIO Port D Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOD_DOUT	GP_BA+0x38	R/W	GPIO Port D Data Output Value	0x0000_0000
GPIOD_PIN	GP_BA+0x3C	R	GPIO Port D Pin Value	0XXXXX_XXXX
GPIOE_OMD	GP_BA+0x40	R/W	GPIO Port E Bit Output Mode Enable	0x0000_0000
GPIOE_PUEN	GP_BA+0x44	R/W	GPIO Port E Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOE_DOUT	GP_BA+0x48	R/W	GPIO Port E Data Output Value	0x0000_0000
GPIOE_PIN	GP_BA+0x4C	R	GPIO Port E Pin Value	0XXXXX_XXXX
DBNCECON	GP_BA+0x70	R/W	External Interrupt De-bounce Control	0x0000_0000
IRQSRCGPA	GP_BA+0x80	R/W	GPIO Port A IRQ Source Grouping	0x0000_0000
IRQSRCGPB	GP_BA+0x84	R/W	GPIO Port B IRQ Source Grouping	0x5555_5555
IRQSRCGPC	GP_BA+0x88	R/W	GPIO Port C IRQ Source Grouping	0xAAAA_AAAA
IRQSRCGPD	GP_BA+0x8C	R/W	GPIO Port D IRQ Source Grouping	0xFFFF_FFFF
IRQSRCGPE	GP_BA+0x90	R/W	GPIO Port E IRQ Source Grouping	0xFFFF_FFFF
IRQENGPA	GP_BA+0xA0	R/W	GPIO Port A Interrupt Enable	0x0000_0000
IRQENGPB	GP_BA+0xA4	R/W	GPIO Port B Interrupt Enable	0x0000_0000
IRQENGPC	GP_BA+0xA8	R/W	GPIO Port C Interrupt Enable	0x0000_0000
IRQENGPD	GP_BA+0xAC	R/W	GPIO Port D Interrupt Enable	0x0000_0000
IRQENGPE	GP_BA+0xB0	R/W	GPIO Port E Interrupt Enable	0x0000_0000

IRQLHSEL	GP_BA+0xC0	R/W	Interrupt Latch Trigger Selection Register	0x0000_0000
IRQLHGPA	GP_BA+0xD0	R	GPIO Port A Interrupt Latch Value	0x0000_0000
IRQLHGPB	GP_BA+0xD4	R	GPIO Port B Interrupt Latch Value	0x0000_0000
IRQLHGPC	GP_BA+0xD8	R	GPIO Port C Interrupt Latch Value	0x0000_0000
IRQLHGPD	GP_BA+0xDC	R	GPIO Port D Interrupt Latch Value	0x0000_0000
IRQLHGPE	GP_BA+0xE0	R	GPIO Port E Interrupt Latch Value	0x0000_0000
IRQTGSRC0	GP_BA+0xF0	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port A and GPIO Port B	0x0000_0000
IRQTGSRC1	GP_BA+0xF4	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port C	0x0000_0000
IRQTGSRC2	GP_BA+0xF8	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port E	0x0000_0000

5.16.3 Register Description

GPIO Port [X] Bit Output Mode Enable (GPIOX_OMD)

Register	Address	R/W	Description	Reset Value
GPIOA_OMD	GP_BA+0x00	R/W	GPIO Port A Bit Output Mode Enable	0x0000_0000
GPIOB_OMD	GP_BA+0x10	R/W	GPIO Port B Bit Output Mode Enable	0x0000_0000
GPIOC_OMD	GP_BA+0x20	R/W	GPIO Port C Bit Output Mode Enable	0x0000_0000
GPIOD_OMD	GP_BA+0x30	R/W	GPIO Port D Bit Output Mode Enable	0x0000_0000
GPIOE_OMD	GP_BA+0x40	R/W	GPIO Port E Bit Output Mode Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
OMD15	OMD14	OMD13	OMD12	OMD11	OMD10	OMD9	OMD8
7	6	5	4	3	2	1	0
OMD7	OMD6	OMD5	OMD4	OMD3	OMD2	OMD1	OMD0

Bits	Descriptions		Default
[n]	OMDn	Bit Output Mode Enable 1 = GPIO port [A/B/C/D/E] bit [n] output mode is enabled, the bit value contained in the corresponding bit [n] of GPIO[A/B/C/D/E]_DOUT is driven on the pin. 0 = GPIO port [A/B/C/D/E] bit [n] output mode is disabled, the corresponding pin is in INPUT mode.	0x0000

GPIO Port [X] Bit Pull-up Resistor Enable (GPIOX_PUEN)

Register	Address	R/W	Description	Reset Value
GPIOA_PUEN	GP_BA+0x04	R/W	GPIO Port A Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOB_PUEN	GP_BA+0x14	R/W	GPIO Port B Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOC_PUEN	GP_BA+0x24	R/W	GPIO Port C Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOD_PUEN	GP_BA+0x34	R/W	GPIO Port D Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOE_PUEN	GP_BA+0x44	R/W	GPIO Port E Bit Pull-up Resistor Enable	0x0000_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PUEN15	PUEN14	PUEN13	PUEN12	PUEN11	PUEN10	PUEN9	PUEN8
7	6	5	4	3	2	1	0
PUEN7	PUEN6	PUEN5	PUEN4	PUEN3	PUEN2	PUEN1	PUEN0

Bits	Descriptions		Default
[n]	PUENn	PUEN[n]: Bit Pull-up Resistor Enable 1 = GPIO port [A/B/C/D/E] bit [n] pull-up resistor is enabled. 0 = GPIO port [A/B/C/D/E] bit [n] pull-up resistor is disabled.	0xFFFF

GPIO Port [X] Data Output Value (GPIOX_DOUT)

Register	Address	R/W	Description	Reset Value
GPIOA_DOUT	GP_BA+0x08	R/W	GPIO Port A Data Output Value	0x0000_0000
GPIOB_DOUT	GP_BA+0x18	R/W	GPIO Port B Data Output Value	0x0000_0000
GPIOC_DOUT	GP_BA+0x28	R/W	GPIO Port C Data Output Value	0x0000_0000
GPIOD_DOUT	GP_BA+0x38	R/W	GPIO Port D Data Output Value	0x0000_0000
GPIOE_DOUT	GP_BA+0x48	R/W	GPIO Port E Data Output Value	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DOUT15	DOUT14	DOUT13	DOUT12	DOUT11	DOUT10	DOUT9	DOUT8
7	6	5	4	3	2	1	0
DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0

Bits	Descriptions		Default
[n]	DOUTn	Bit Output Value 1 = GPIO port [A/B/C/D/E] bit [n] will drive High if the corresponding output mode enabling bit is set. 0 = GPIO port [A/B/C/D/E] bit [n] will drive Low if the corresponding output mode enabling bit is set.	0x0000

GPIO Port [X] Pin Value (GPIOX_PIN)

Register	Address	R/W	Description	Reset Value
GPIOA_PIN	GP_BA+0x0C	R	GPIO Port A Pin Value	0x0000_XXXX
GPIOB_PIN	GP_BA+0x1C	R	GPIO Port B Pin Value	0x0000_XXXX
GPIOC_PIN	GP_BA+0x2C	R	GPIO Port C Pin Value	0x0000_XXXX
GPIOD_PIN	GP_BA+0x3C	R	GPIO Port D Pin Value	0x0000_XXXX
GPIOE_PIN	GP_BA+0x4C	R	GPIO Port E Pin Value	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PIN[15:8]							
7	6	5	4	3	2	1	0
PIN[7:0]							

Bits	Descriptions		Default
[15:0]	PIN	Port [A/B/C/D/E] Pin Values Each bit of this register reflects the value of each GPIO pin.	0xFFFF

Interrupt De-bounce Control (DBNCECON)

Register	Address	R/W	Description	Reset Value
DBNCECON	GP_BA+0x70	R/W	External Interrupt De-bounce Control	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
DBCLKSEL				DBEN			

Bits	Descriptions		Default																																		
[7:4]	DBCLKSEL	Debounce sampling cycle selection	0x0																																		
		<table><tr><th>DBCLKSEL</th><th>Description</th></tr><tr><td>0</td><td>Sample interrupt input once per 1 clocks</td></tr><tr><td>1</td><td>Sample interrupt input once per 2 clocks</td></tr><tr><td>2</td><td>Sample interrupt input once per 4 clocks</td></tr><tr><td>3</td><td>Sample interrupt input once per 8 clocks</td></tr><tr><td>4</td><td>Sample interrupt input once per 16 clocks</td></tr><tr><td>5</td><td>Sample interrupt input once per 32 clocks</td></tr><tr><td>6</td><td>Sample interrupt input once per 64 clocks</td></tr><tr><td>7</td><td>Sample interrupt input once per 128 clocks</td></tr><tr><td>8</td><td>Sample interrupt input once per 256 clocks</td></tr><tr><td>9</td><td>Sample interrupt input once per 2*256 clocks</td></tr><tr><td>10</td><td>Sample interrupt input once per 4*256clocks</td></tr><tr><td>11</td><td>Sample interrupt input once per 8*256 clocks</td></tr><tr><td>12</td><td>Sample interrupt input once per 16*256 clocks</td></tr><tr><td>13</td><td>Sample interrupt input once per 32*256 clocks</td></tr><tr><td>14</td><td>Sample interrupt input once per 64*256 clocks</td></tr><tr><td>15</td><td>Sample interrupt input once per 128*256 clocks</td></tr></table>		DBCLKSEL	Description	0	Sample interrupt input once per 1 clocks	1	Sample interrupt input once per 2 clocks	2	Sample interrupt input once per 4 clocks	3	Sample interrupt input once per 8 clocks	4	Sample interrupt input once per 16 clocks	5	Sample interrupt input once per 32 clocks	6	Sample interrupt input once per 64 clocks	7	Sample interrupt input once per 128 clocks	8	Sample interrupt input once per 256 clocks	9	Sample interrupt input once per 2*256 clocks	10	Sample interrupt input once per 4*256clocks	11	Sample interrupt input once per 8*256 clocks	12	Sample interrupt input once per 16*256 clocks	13	Sample interrupt input once per 32*256 clocks	14	Sample interrupt input once per 64*256 clocks	15	Sample interrupt input once per 128*256 clocks
		DBCLKSEL		Description																																	
		0		Sample interrupt input once per 1 clocks																																	
		1		Sample interrupt input once per 2 clocks																																	
		2		Sample interrupt input once per 4 clocks																																	
		3		Sample interrupt input once per 8 clocks																																	
		4		Sample interrupt input once per 16 clocks																																	
		5		Sample interrupt input once per 32 clocks																																	
		6		Sample interrupt input once per 64 clocks																																	
		7		Sample interrupt input once per 128 clocks																																	
		8		Sample interrupt input once per 256 clocks																																	
		9		Sample interrupt input once per 2*256 clocks																																	
		10		Sample interrupt input once per 4*256clocks																																	
		11		Sample interrupt input once per 8*256 clocks																																	
		12		Sample interrupt input once per 16*256 clocks																																	
		13		Sample interrupt input once per 32*256 clocks																																	
14	Sample interrupt input once per 64*256 clocks																																				
15	Sample interrupt input once per 128*256 clocks																																				
[3:0]	DBEN	DBEN[x]: de-bounce sampling enable for each IRQx, x = 0 ~ 3 1 = Interrupt input IRQx is filtered with de-bounce sampling 0 = Interrupt input IRQx is input directly without de-bounce sampling	0x0																																		

IRQ Source Grouping (IRQSRCGPA)

Register	Address	R/W	Description	Reset Value
IRQSRCGPA	GP_BA+0x80	R/W	GPIO Port A IRQ Source Grouping	0x0000_0000

31	30	29	28	27	26	25	24
GPA15SEL		GPA14SEL		GPA13SEL		GPA12SEL	
23	22	21	20	19	18	17	16
GPA11SEL		GPA10SEL		GPA9SEL		GPA8SEL	
15	14	13	12	11	10	9	8
GPA7SEL		GPA6SEL		GPA5SEL		GPA4SEL	
7	6	5	4	3	2	1	0
GPA3SEL		GPA2SEL		GPA1SEL		GPA0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPAxSEL	Selection for GPAx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x0

Where x=0~15.

- GPAxSEL =
- 0, GPAx pin is grouped as one of interrupt sources to IRQ0.
 - 1, GPAx pin is grouped as one of interrupt sources to IRQ1.
 - 2, GPAx pin is grouped as one of interrupt sources to IRQ2.
 - 3, GPAx pin is grouped as one of interrupt sources to IRQ3.

IRQ Source Grouping (IRQSRCGPB)

Register	Address	R/W	Description	Reset Value
IRQSRCGPB	GP_BA+0x84	R/W	GPIO Port B IRQ Source Grouping	0x5555_5555

31	30	29	28	27	26	25	24
GPB15SEL		GPB14SEL		GPB13SEL		GPB12SEL	
23	22	21	20	19	18	17	16
GPB11SEL		GPB10SEL		GPB9SEL		GPB8SEL	
15	14	13	12	11	10	9	8
GPB7SEL		GPB6SEL		GPB5SEL		GPB4SEL	
7	6	5	4	3	2	1	0
GPB3SEL		GPB2SEL		GPB1SEL		GPB0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPBxSEL	Selection for GPBx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x0

Where x=0~15.

- GPBxSEL =
- 0, GPBx pin is grouped as one of interrupt sources to IRQ0.
 - 1, GPBx pin is grouped as one of interrupt sources to IRQ1.
 - 2, GPBx pin is grouped as one of interrupt sources to IRQ2.
 - 3, GPBx pin is grouped as one of interrupt sources to IRQ3.

IRQ Source Grouping (IRQSRCGPC)

Register	Address	R/W	Description	Reset Value
IRQSRCGPC	GP_BA+0x88	R/W	GPIO Port C IRQ Source Grouping	0xAAAA_AAAA

31	30	29	28	27	26	25	24
GPC15SEL		GPC14SEL		GPC13SEL		GPC12SEL	
23	22	21	20	19	18	17	16
GPC11SEL		GPC10SEL		GPC9SEL		GPC8SEL	
15	14	13	12	11	10	9	8
GPC7SEL		GPC6SEL		GPC5SEL		GPC4SEL	
7	6	5	4	3	2	1	0
GPC3SEL		GPC2SEL		GPC1SEL		GPC0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPCxSEL	Selection for GPCx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x0

Where x=0~15.

- GPCxSEL =
- 0, GPCx pin is grouped as one of interrupt sources to IRQ0.
 - 1, GPCx pin is grouped as one of interrupt sources to IRQ1.
 - 2, GPCx pin is grouped as one of interrupt sources to IRQ2.
 - 3, GPCx pin is grouped as one of interrupt sources to IRQ3.

IRQ Source Grouping (IRQSRCGPD)

Register	Address	R/W	Description	Reset Value
IRQSRCGPD	GP_BA+0x8C	R/W	GPIO Port D IRQ Source Grouping	0xFFFF_FFFF

31	30	29	28	27	26	25	24
GPD15SEL		GPD14SEL		GPD13SEL		GPD12SEL	
23	22	21	20	19	18	17	16
GPD11SEL		GPD10SEL		GPD9SEL		GPD8SEL	
15	14	13	12	11	10	9	8
GPD7SEL		GPD6SEL		GPD5SEL		GPD4SEL	
7	6	5	4	3	2	1	0
GPD3SEL		GPD2SEL		GPD1SEL		GPD0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPDxSEL	Selection for GPAX as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x0

Where x=0~15.

- GPDxSEL =
- 0, GPDx pin is grouped as one of interrupt sources to IRQ0.
 - 1, GPDx pin is grouped as one of interrupt sources to IRQ1.
 - 2, GPDx pin is grouped as one of interrupt sources to IRQ2.
 - 3, GPDx pin is grouped as one of interrupt sources to IRQ3.

IRQ Source Grouping (IRQSRCGPE)

Register	Address	R/W	Description	Reset Value
IRQSRCGPE	GP_BA+0x90	R/W	GPIO Port E IRQ Source Grouping	0xFFFF_FFFF

31	30	29	28	27	26	25	24
GPE15SEL		GPE14SEL		GPE13SEL		GPE12SEL	
23	22	21	20	19	18	17	16
GPE11SEL		GPE10SEL		GPE9SEL		GPE8SEL	
15	14	13	12	11	10	9	8
GPE7SEL		GPE6SEL		GPE5SEL		GPE4SEL	
7	6	5	4	3	2	1	0
GPE3SEL		GPE2SEL		GPE1SEL		GPE0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPExSEL	Selection for GPAX as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x0

Where x=0~15.

- GPExSEL =
- 0, GPEx pin is grouped as one of interrupt sources to IRQ0.
 - 1, GPEx pin is grouped as one of interrupt sources to IRQ1.
 - 2, GPEx pin is grouped as one of interrupt sources to IRQ2.
 - 3, GPEx pin is grouped as one of interrupt sources to IRQ3.

GPIO A Interrupt Enable (IRQENGPA)

Register	Address	R/W	Description	Reset Value
IRQENGPA	GP_BA+0xA0	R/W	GPIO Port A Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PA15ENR	PA14ENR	PA13ENR	PA12ENR	PA11ENR	PA10ENR	PA9ENR	PA8ENR
23	22	21	20	19	18	17	16
PA7ENR	PA6ENR	PA5ENR	PA4ENR	PA3ENR	PA2ENR	PA1ENR	PA0ENR
15	14	13	12	11	10	9	8
PA15ENF	PA14ENF	PA13ENF	PA12ENF	PA11ENF	PA10ENF	PA9ENF	PA8ENF
7	6	5	4	3	2	1	0
PA7ENF	PA6ENF	PA5ENF	PA4ENF	PA3ENF	PA2ENF	PA1ENF	PA0ENF

Bits	Descriptions		Default
[x]	PAXENF	Enable GPAX input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPA register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PAXENR	Enable GPAX input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPA register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PAXENF and PAXENR can be set “1” at the same time.

NOTE1: In normal operation mode, for each pin, PAXENF and PAXENR can be set both to detect both rising and falling edge.

NOTE2: When use a pin as powerdown wake up source, the setting of edges must be explained as level trigger. For example, if set one pin for rising, user must keep this pin low while start to enter power down, a high level will make power-down entrance be ignored. After entering power down, a high level at this pin will make chip leave power-down.

GPIO B Interrupt Enable (IRQENGPB)

Register	Address	R/W	Description	Reset Value
IRQENGPB	GP_BA+0xA4	R/W	GPIO Port B Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PB15ENR	PB14ENR	PB13ENR	PB12ENR	PB11ENR	PB10ENR	PB9ENR	PB8ENR
23	22	21	20	19	18	17	16
PB7ENR	PB6ENR	PB5ENR	PB4ENR	PB3ENR	PB2ENR	PB1ENR	PB0ENR
15	14	13	12	11	10	9	8
PB15ENF	PB14ENF	PB13ENF	PB12ENF	PB11ENF	PB10ENF	PB9ENF	PB8ENF
7	6	5	4	3	2	1	0
PB7ENF	PB6ENF	PB5ENF	PB4ENF	PB3ENF	PB2ENF	PB1ENF	PB0ENF

Bits	Descriptions		Default
[x]	PBxENF	Enable GPBx input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPB register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PBxENR	Enable GPBx input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPB register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PBxENF and PBxENR can be set "1" at the same time.

NOTE1: In normal operation mode, for each pin, PBxENF and PBxENR can be set both to detect both rising and falling edge.

NOTE2: When use a pin as powerdown wake up source, the setting of edges must be explained as level trigger. For example, if set one pin for rising, user must keep this pin low while start to enter power down, a high level will make power-down entrance be ignored. After entering power down, a high level at this pin will make chip leave power-down.

GPIO C Interrupt Enable (IRQENGPC)

Register	Address	R/W	Description	Reset Value
IRQENGPC	GP_BA+0xA8	R/W	GPIO Port C Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PC15ENR	PC14ENR	PC13ENR	PC12ENR	PC11ENR	PC10ENR	PC9ENR	PC8ENR
23	22	21	20	19	18	17	16
PC7ENR	PC6ENR	PC5ENR	PC4ENR	PC3ENR	PC2ENR	PC1ENR	PC0ENR
15	14	13	12	11	10	9	8
PC15ENF	PC14ENF	PC13ENF	PC12ENF	PC11ENF	PC10ENF	PC9ENF	PC8ENF
7	6	5	4	3	2	1	0
PC7ENF	PC6ENF	PC5ENF	PC4ENF	PC3ENF	PC2ENF	PC1ENF	PC0ENF

Bits	Descriptions		Default
[x]	PCxENF	Enable GPCx input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPC register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PCxENR	Enable GPCx input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPC register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PCxENF and PCxENR can be set "1" at the same time.

NOTE1: In normal operation mode, for each pin, PCxENF and PCxENR can be set both to detect both rising and falling edge.

NOTE2: When use a pin as powerdown wake up source, the setting of edges must be explained as level trigger. For example, if set one pin for rising, user must keep this pin low while start to enter power down, a high level will make power-down entrance be ignored. After entering power down, a high level at this pin will make chip leave power-down.

GPIO D Interrupt Enable (IRQENGPD)

Register	Address	R/W	Description	Reset Value
IRQENGPD	GP_BA+0xAC	R/W	GPIO Port D Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PD15ENR	PD14ENR	PD13ENR	PD12ENR	PD11ENR	PD10ENR	PD9ENR	PD8ENR
23	22	21	20	19	18	17	16
PD7ENR	PD6ENR	PD5ENR	PD4ENR	PD3ENR	PD2ENR	PD1ENR	PD0ENR
15	14	13	12	11	10	9	8
PD15ENF	PD14ENF	PD13ENF	PD12ENF	PD11ENF	PD10ENF	PD9ENF	PD8ENF
7	6	5	4	3	2	1	0
PD7ENF	PD6ENF	PD5ENF	PD4ENF	PD3ENF	PD2ENF	PD1ENF	PD0ENF

Bits	Descriptions		Default
[x]	PDxENF	Enable GPDx input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPD register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PDxENR	Enable GPDx input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPD register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PDxENF and PDxENR can be set "1" at the same time.

NOTE1: In normal operation mode, for each pin, PDxENF and PDxENR can be set both to detect both rising and falling edge.

NOTE2: When use a pin as powerdown wake up source, the setting of edges must be explained as level trigger. For example, if set one pin for rising, user must keep this pin low while start to enter power down, a high level will make power-down entrance be ignored. After entering power down, a high level at this pin will make chip leave power-down.

GPIO E Interrupt Enable (IRQENGPE)

Register	Address	R/W	Description	Reset Value
IRQENGPE	GP_BA+0xB0	R/W	GPIO Port E Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PE15ENR	PE14ENR	PE13ENR	PE12ENR	PE11ENR	PE10ENR	PE9ENR	PE8ENR
23	22	21	20	19	18	17	16
PE7ENR	PE6ENR	PE5ENR	PE4ENR	PE3ENR	PE2ENR	PE1ENR	PE0ENR
15	14	13	12	11	10	9	8
PE15ENF	PE14ENF	PE13ENF	PE12ENF	PE11ENF	PE10ENF	PE9ENF	PE8ENF
7	6	5	4	3	2	1	0
PE7ENF	PE6ENF	PE5ENF	PE4ENF	PE3ENF	PE2ENF	PE1ENF	PE0ENF

Bits	Descriptions		Default
[x]	PExENF	Enable GPEX input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPE register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PExENR	Enable GPEX input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPE register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PExENF and PExENR can be set "1" at the same time.

NOTE1: In normal operation mode, for each pin, PExENF and PExENR can be set both to detect both rising and falling edge.

NOTE2: When use a pin as powerdown wake up source, the setting of edges must be explained as level trigger. For example, if set one pin for rising, user must keep this pin low while start to enter power down, a high level will make power-down entrance be ignored. After entering power down, a high level at this pin will make chip leave power-down.

Interrupt Latch Trigger Selection (IRQLHSEL)

Register	Address	R/W	Description	Reset Value
IRQLHSEL	GP_BA+0xC0	R/W	Interrupt Latch Trigger Selection Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
IRQ3Wake	IRQ2Wake	IRQ1Wake	IRQ0Wake	IRQ3LHE	IRQ2LHE	IRQ1LHE	IRQ0LHE

Bits	Descriptions		Default
[7:4]	IRQxWake	GPIO interrupt wake up system enable While IRQxWake is "1", enable the GPIO IRQx wake up the chip from power down mode.	
[3:0]	IRQxLHE	Interrupt Latch Enable While IRQxLTH is "1", it enables active IRQx interrupt to latch the input values of GPA/GPB/GPC/GPD/GPE to IRQLHGPA/IRQLHGPB/IRQLHGPC/IRQLHGPD/IRQLHGPE register simultaneously.	0x0

Where x=0~3.

GPIO X Interrupt Latch (IRQLHGPX)

Register	Address	R/W	Description	Reset Value
IRQLHGPA	GP_BA+0xD0	R	GPIO Port A Interrupt Latch Value	0x0000_0000
IRQLHGPB	GP_BA+0xD4	R	GPIO Port B Interrupt Latch Value	0x0000_0000
IRQLHGPC	GP_BA+0xD8	R	GPIO Port C Interrupt Latch Value	0x0000_0000
IRQLHGPD	GP_BA+0xDC	R	GPIO Port D Interrupt Latch Value	0x0000_0000
IRQLHGPE	GP_BA+0xE0	R	GPIO Port E Interrupt Latch Value	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PX15LHV	PX14LHV	PX13LHV	PX12LHV	PX11LHV	PX10LHV	PX9LHV	PX8LHV
7	6	5	4	3	2	1	0
PX7LHV	PX6LHV	PX5LHV	PX4LHV	PX3LHV	PX2LHV	PX1LHV	PX0LHV

Bits	Descriptions		Default
[x]	PXxLHV	Latched value of GPXx while the IRQ (IRQ0~IRQ3) selected by IRQLHSEL is active.	0x0

Where X=A or B or C or D, x=0~15

NOTE: When a latched pin value is '0', there will be 2 meanings: either the pin's input is recognized as LOW or the pin is setup as output mode, so the input value is masked as '0'.

IRQ Interrupt Trigger Source 0 (IRQTGSRC0)

Register	Address	R/W	Description	Reset Value
IRQTGSRC0	GP_BA+0xF0	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port A and GPIO Port B	0x0000_0000

31	30	29	28	27	26	25	24
PB15TG	PB14TG	PB13TG	PB12TG	PB11TG	PB10TG	PB9TG	PB8TG
23	22	21	20	19	18	17	16
PB7TG	PB6TG	PB5TG	PB4TG	PB3TG	PB2TG	PB1TG	PB0TG
15	14	13	12	11	10	9	8
PA15TG	PA14TG	PA13TG	PA12TG	PA11TG	PA10TG	PA9TG	PA8TG
7	6	5	4	3	2	1	0
PA7TG	PA6TG	PA5TG	PA4TG	PA3TG	PA2TG	PA1TG	PA0TG

Bits	Descriptions		Default
[x]	PAxTG	When this bit is read as “1”, it indicates GPAX is the trigger source to generate interrupt to the IRQ (IRQ0~IRQ3) selected by IRQLHSEL. Write 1 to the bit[x] will clear the correspond interrupt source	0x0
[x+16]	PBxTG	When this bit is read as “1”, it indicates GPBx is the trigger source to generate interrupt to the IRQ (IRQ0~IRQ3) selected by IRQLHSEL. Write 1 to the bit[x] will clear the correspond interrupt source	0x0

Where x=0~15.

NOTE: The trigger source will be latched when the corresponding rising or falling trigger enable is setup and the pin state toggle is recognized (through debounce or without debounce), no matter whether the source is an input or output pin.

IRQ Interrupt Trigger Source 1 (IRQTGSRC1)

Register	Address	R/W	Description	Reset Value
IRQTGSRC1	GP_BA+0xF4	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port C and GPIO Port D	0x0000_0000

31	30	29	28	27	26	25	24
PD15TG	PD14TG	PD13TG	PD12TG	PD11TG	PD10TG	PD9TG	PD8TG
23	22	21	20	19	18	17	16
PD7TG	PD6TG	PD5TG	PD4TG	PD3TG	PD2TG	PD1TG	PD0TG
15	14	13	12	11	10	9	8
PC15TG	PC14TG	PC13TG	PC12TG	PC11TG	PC10TG	PC9TG	PC8TG
7	6	5	4	3	2	1	0
PC7TG	PC6TG	PC5TG	PC4TG	PC3TG	PC2TG	PC1TG	PC0TG

Bits	Descriptions		Default
[x]	PCxTG	When this bit is read as “1”, it indicates GPCx is the trigger source to generate interrupt to the IRQ (IRQ0~IRQ3) selected by IRQLHSEL. Write 1 to the bit[x] will clear the correspond interrupt source	0x0
[x+16]	PDxTG	When this bit is read as “1”, it indicates GPCx is the trigger source to generate interrupt to the IRQ (IRQ0~IRQ3) selected by IRQLHSEL. Write 1 to the bit[x] will clear the correspond interrupt source	0x0

Where x=0~15.

NOTE: The trigger source will be latched when the corresponding rising or falling trigger enable is setup and the pin state toggle is recognised (through debounce or without debounce), no matter whether the source is an input or output pin.

Other NOTE for related setup

NOTE1: For the AIC's normal functionality to be triggered by external IO interrupt, the external IRQ source settings (for IRQ0/1/2/3) can only be set as positive edge, see AIC_SCRxx/bit7~6: SRCTYPE.

NOTE2: For power-down wake up setting, in order to keep normal wake up functionality, the wake up source polarity should be set as positive level, see IRQLHSEL[7~4]:IRQENGPX(X=C or D in this case).

IRQ Interrupt Trigger Source 2 (IRQTGSRC2)

Register	Address	R/W	Description	Reset Value
IRQTGSRC2	GP_BA+0xF8	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port E	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PE15TG	PE14TG	PE13TG	PE12TG	PE11TG	PE10TG	PE9TG	PE8TG
7	6	5	4	3	2	1	0
PE7TG	PE6TG	PE5TG	PE4TG	PE3TG	PE2TG	PE1TG	PE0TG

Bits	Descriptions		Default
[x]	PE _x TG	When this bit is read as "1", it indicates GPEx is the trigger source to generate interrupt to the IRQ (IRQ0~IRQ3) selected by IRQLHSEL. Write 1 to the bit[x] will clear the correspond interrupt source	0x0

Where x=0~15.

NOTE: The trigger source will be latched when the corresponding rising or falling trigger enable is setup and the pin state toggle is recognised (through debounce or without debounce), no matter whether the source is an input or output pin.

Other NOTE for related setup

NOTE1: For the AIC's normal functionality to be triggered by external IO interrupt, the external IRQ source settings (for IRQ0/1/2/3) can only be set as positive edge, see AIC_SCRxx/bit7~6: SRCTYPE.

NOTE2: For power-down wake up setting, in order to keep normal wake up functionality, the wake up source polarity should be set as positive level, see IRQLHSEL[7~4]:IRQENGPX(X=E in this case).

5.17 TIMER Controller

5.17.1 General Timer Controller

The timer module includes two channels, TIMER0~TIMER1, which allow you to easily implement a counting scheme for use. The timer can perform functions like frequency measurement, event counting, interval measurement, clock generation, delay timing, and so on. The timer possesses features such as adjustable resolution, programmable counting period, and detailed information. The timer can generate an interrupt signal upon timeout, or provide the current value of count during operation.

5.17.2 Features

- AMBA APB interface compatible
- Each channel with 8-bit pre-scale counter/32-bit counter and an interrupt request signal.
- Independent clock source for each channel(TCLK0,TCLK1)
- Maximum uninterrupted time = $(1 / 25 \text{ MHz}) * (2^8) * (2^{32})$, if TCLK = 25 MHz

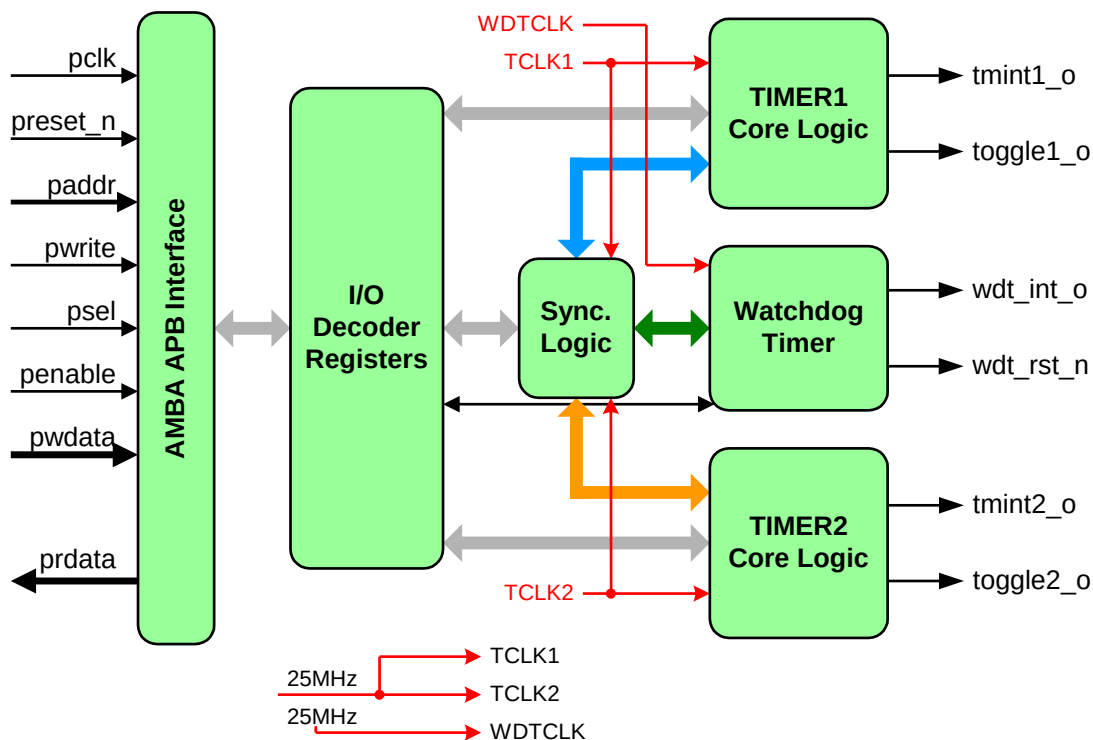


Figure 5.17-1 TIMER Block Diagram

5.17.3 Watchdog Timer

The purpose of Watchdog Timer is to perform a system restart after the software running into a

problem. This prevents system from hanging for an indefinite period of time. It is a free running timer with programmable timeout intervals. When the specified time interval expires, a system reset can be generated. If the Watchdog Timer reset function is enabled and the Watchdog Timer is not being reset before timing out, then the Watchdog Timer reset is activated after **1024 WDT clock cycles** (Interrupt timeout). Setting **WTE** in the register **WTCR** enables the Watchdog Timer.

The **WTR** should be set before making use of Watchdog Timer. This ensures that the Watchdog Timer restarts from a known state. Watchdog Timer will start counting and timeout after a specified period of time. The timeout interval is selected by two bits, **WTIS[1:0]**. The **WTR** is self-clearing, i.e., after setting it; the hardware will automatically reset it.

When timeout occurs, Watchdog Timer interrupt flag is set. Watchdog Timer waits for an additional **1024 WDT clock cycles** before issuing a reset signal, if the **WTRE** is set. The **WTRF** will be set and the reset signal will last for **16128 WDT clock cycles** long. Watchdog Timer will set the **WTIF** each time a timeout occurs. The **WTIF** can be polled to check the status, and software can restart the timer by setting the **WTR**. The Watchdog Timer can be put in the test mode by setting **WTTME** in the register **WTCR**.

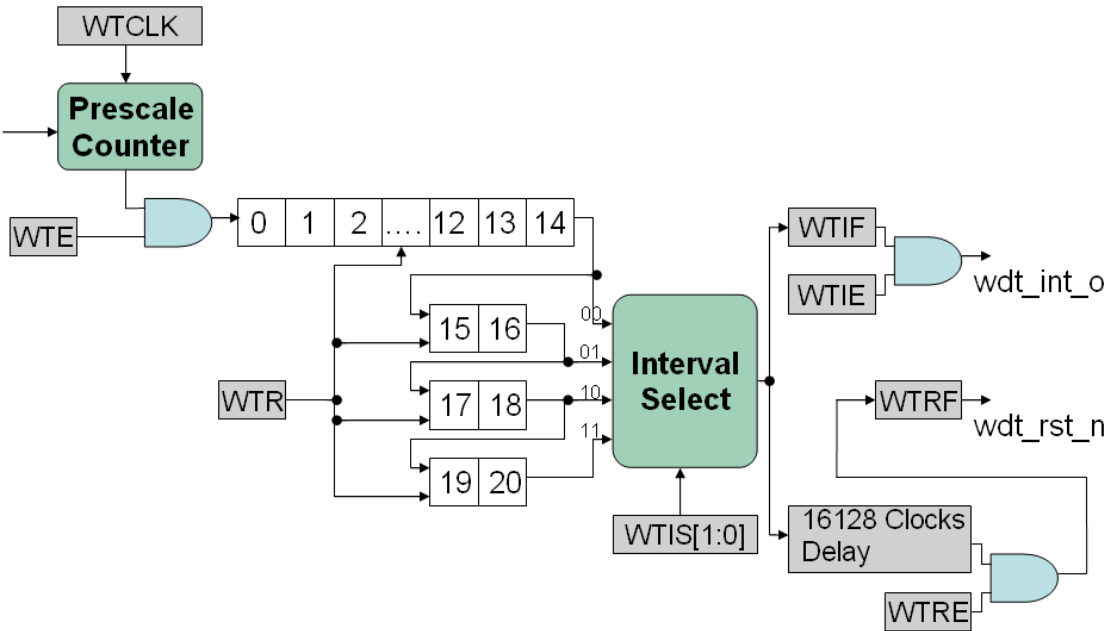
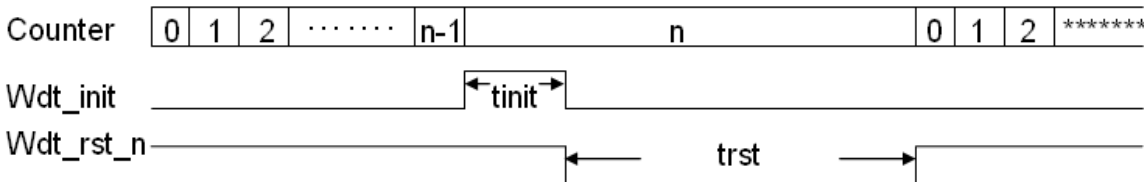


Figure 5.17-2 Watchdog Timer Block Diagram



Parameter	Min	Max	Unit
n	2 ¹⁴	2 ²⁰	wdt clock cycle
t_init		1024	wdt clock cycle
t_rst		16128	wdt clock cycle

Figure 5.17-3 Watchdog Timer Timing Diagram

5.17.4 Register Map

R: read only, W: write only, R/W: both read and write

TMR_BA = 0xB800_2000

Register	Address	R/W/C	Description	Reset Value
TMR_BA = 0xB800_2000				
TCSR0	TMR_BA+00	R/W	Timer Control and Status Register 0	0x0000_0005
TCSR1	TMR_BA+04	R/W	Timer Control and Status Register 1	0x0000_0005
TICR0	TMR_BA+08	R/W	Timer Initial Control Register 0	0x0000_0000
TICR1	TMR_BA+0C	R/W	Timer Initial Control Register 1	0x0000_0000
TDR0	TMR_BA+10	R	Timer Data Register 0	0x0000_0000
TDR1	TMR_BA+14	R	Timer Data Register 1	0x0000_0000
TISR	TMR_BA+18	R/W	Timer Interrupt Status Register	0x0000_0000
WTCR	TMR_BA+1C	R/W	Watchdog Timer Control Register	0x0000_0400

5.17.5 Register Description

Timer Control Register 0~1 (TCSR0~TCSR1)

Register	Address	R/W	Description	Reset Value
TCSR0	TMR_BA+000	R/W	Timer Control and Status Register 0	0x0000_0005
TCSR1	TMR_BA+004	R/W	Timer Control and Status Register 1	0x0000_0005

31	30	29	28	27	26	25	24
nDBGACK_EN	CEN	IE	MODE[1:0]		CRST	CACT	Reserved
23	22	21	20	19	18	17	16
Reserved							TDR_EN
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PRESCALE[7:0]							

Bits	Descriptions		
[31]	nDBGACK_EN	ICE debug mode acknowledge enable 0 = When DBGACK is high, the TIMER counter will be held 1 = No matter DBGACK is high or not, the TIMER counter will not be held	
[30]	CEN	Counter Enable 0 = Stops/Suspends counting 1 = Starts counting	
[29]	IE	Interrupt Enable 0 = Disable TIMER Interrupt. 1 = Enable TIMER Interrupt. If timer interrupt is enabled, the timer asserts its interrupt signal when the associated counter is equal to TICR.	
[28:27]	MODE	Timer Operating Mode	
		MODE	Timer Operating Mode
		00	The timer is operating in the one-shot mode. The associated interrupt signal is generated once (if IE is enabled) and CEN is automatically cleared then.
		01	The timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if IE is enabled).
		10	The timer is operating in the toggle mode. The interrupt signal is generated periodically (if IE is enabled). And the associated signal (tout) is changing back and forth with 50% duty cycle.
		11	The timer is operating in the uninterrupted mode. The associated interrupt signal is generated when TDR = TICR (if IE is enabled).
[26]	CRST	Counter Reset	

		Set this bit will reset the TIMER counter, and also force CEN to 0. 0 = No effect. 1 = Reset Timer's prescale counter, internal 32-bit counter and CEN.
[25]	CACT	Timer is in Active This bit indicates the counter status of timer. 0 = Timer is not active. 1 = Timer is in active.
[24:17]	Reserved	Reserved
[16]	TDR_EN	1 = Timer Data Register update enable. 0 = Timer Data Register update disable.
[15:8]	Reserved	Reserved
[7:0]	PRESCALE	Prescale Clock input is divided by Prescale+1 before it is fed to the counter. If Prescale=0, then there is no scaling.

Timer Initial Count Register 0~1 (TICR0~TICR1)

Register	Address	R/W	Description	Reset Value
TICR0	TMR_BA+008	R/W	Timer Initial Control Register 0	0x0000_0000
TICR1	TMR_BA+00C	R/W	Timer Initial Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
TIC[31:24]							
23	22	21	20	19	18	17	16
TIC [23:16]							
15	14	13	12	11	10	9	8
TIC [15:8]							
7	6	5	4	3	2	1	0
TIC [7:0]							

Bits	Descriptions	
[31:0]	TIC	Timer Initial Count This is a 32-bit value representing the initial count. Timer will reload this value whenever the counter is equal to the definition value (software define). NOTE1: Never write 0x0 or 0x1 in TIC, or the core will run into unknown state. NOTE2: No matter CEN is 0 or 1, whenever software write a new value into this register, TIMER will restart counting using this new value and abort previous count.

Timer Data Register 0~1 (TDR0~TDR1)

Register	Address	R/W	Description	Reset Value
TDR0	TMR_BA+10	R	Timer Data Register 0	0x0000_0000
TDR1	TMR_BA+14	R	Timer Data Register 1	0x0000_0000

31	30	29	28	27	26	25	24
TDR[31:24]							
23	22	21	20	19	18	17	16
TDR [23:16]							
15	14	13	12	11	10	9	8
TDR [15:8]							
7	6	5	4	3	2	1	0
TDR [7:0]							

Bits	Descriptions	
[31:0]	TDR	Timer Data Register The current count is registered in this 32-bit value. NOTE: Software can read a correct current value on this register only when CEN = 0, or the value represents here could not be a correct one.

Timer Interrupt Status Register (TISR)

Register	Address	R/W	Description	Reset Value
TISR	TMR_BA+18	R/W	Timer Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TIF1	TIF0

Bits	Descriptions	
[1]	TIF1	Timer Interrupt Flag 1 This bit indicates the interrupt status of Timer channel 1. 0 = It indicates that the Timer 1 dose not equal the definition value (software define) yet. 1 = It indicates that the counter of Timer 1 has equal to the definition value. The interrupt flag is set if it was enable. NOTE: This bit is read only, but can be cleared by writing 1 to this bit.
[0]	TIF0	Timer Interrupt Flag 0 This bit indicates the interrupt status of Timer channel 0. 0 = It indicates that the Timer 0 dose not equal the definition value (software define) yet. 1 = It indicates that the counter of Timer 0 has has equal to the definition value. The interrupt flag is set if it was enable. NOTE: This bit is read only, but can be cleared by writing 1 to this bit.

Watchdog Timer Control Register (WTCR)

Register	Address	R/W	Description	Reset Value
WTCR	TMR_BA+01C	R/W	Watchdog Timer Control Register	0x0000_0400

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					WTCLK	nDBGACK_EN	WTTME
7	6	5	4	3	2	1	0
WTE	WTIE	WTIS		WTIF	WTRF	WTRE	WTR

Bits	Descriptions	
[31:11]	Reserved	Reserved
[10]	WTCLK	Watchdog Timer Clock This bit is used for deciding whether the Watchdog timer clock input is divided by 256 or not. Clock source of Watchdog timer is Crystal input. 0 = Using original clock input 1 = The clock input will be divided by 256 NOTE: When WTTME = 1, set this bit has no effect on WDT clock (using original clock input).
[9]	nDBGACK_EN	ICE debug mode acknowledge enable 0 = When DBGACK is high, the Watchdog timer counter will be held 1 = No matter DBGACK is high or not, the Watchdog timer counter will not be held
[8]	WTTME	Watchdog Timer Test Mode Enable For reasons of efficiency, the 20-bit counter within the Watchdog timer is considered as two independent 10-bit counters in the test mode. They are operated concurrently and separately during the test. This approach can save a lot of time spent in the test. When the 10-bit counter overflows, a Watchdog timer interrupt is generated. 0 = Put the Watchdog timer in normal operating mode 1 = Put the Watchdog timer in test mode
[7]	WTE	Watchdog Timer Enable 0 = Disable the Watchdog timer (This action will reset the internal counter) 1 = Enable the Watchdog timer
[6]	WTIE	Watchdog Timer Interrupt Enable 0 = Disable the Watchdog timer interrupt 1 = Enable the Watchdog timer interrupt
[5:4]	WTIS	Watchdog Timer Interval Select These two bits select the interval for the Watchdog timer. No matter which interval is chosen, the reset timeout is always occurred 16128 WDT clock cycles later than the interrupt timeout.

		WTIS	Timeout	Interrupt Timeout	Real Time Interval (CLK=15MHz/256)
		00	2 ¹⁴ clocks	2 ¹⁴ + 1024 clocks	0.28 sec.
		01	2 ¹⁶ clocks	2 ¹⁶ + 1024 clocks	1.12 sec.
		10	2 ¹⁸ clocks	2 ¹⁸ + 1024 clocks	4.47 sec.
		11	2 ²⁰ clocks	2 ²⁰ + 1024 clocks	17.9 sec.
		Note : Reference the Figure3 Watchdog Timer Timing Diagram			
[3]	WTIF	Watchdog Timer Interrupt Flag If the Watchdog timer interrupt is enabled, then the hardware will set this bit to indicate that the Watchdog timer interrupt has occurred. If the Watchdog timer interrupt is not enabled, then this bit indicates that a timeout period has elapsed. 0 = Watchdog timer interrupt does not occur 1 = Watchdog timer interrupt occurs NOTE: This bit is read only, but can be cleared by writing 1 to this bit.			
[2]	WTRF	Watchdog Timer Reset Flag When the Watchdog timer initiates a reset, the hardware will set this bit. This flag can be read by software to determine the source of reset. Software is responsible to clear it up manually. If WTRE is disabled, then the Watchdog timer has no effect on this bit. 0 = Watchdog timer reset does not occur 1 = Watchdog timer reset occurs NOTE: This bit is read only, but can be cleared by writing 1 to this bit.			
[1]	WTRE	Watchdog Timer Reset Enable Setting this bit will enable the Watchdog timer reset function. 0 = Disable Watchdog timer reset function 1 = Enable Watchdog timer reset function			
[0]	WTR	Watchdog Timer Reset This bit brings the Watchdog timer into a known state. It helps reset the Watchdog timer before a timeout situation occurring. Failing to set WTR before timeout will initiates an interrupt if WTIE is set. If the WTRE bit is set, Watchdog timer reset will be occurred 16128 WDT clock cycles after interrupt timeout. This bit is self-clearing. 0 = Writing 0 to this bit has no effect 1 = Reset the contents of the Watchdog timer NOTE: This bit will auto clear after few clock cycle			

5.18 Real Time Clock (RTC)

5.18.1 Overview

Real Time Clock (RTC) block can be operated by independent power supply while the system power is off. The RTC uses a 32.768 KHz external crystal. It can transmit data to CPU with BCD values. The data includes the time by (second, minute and hour), the day by (day, month and year). In addition, to achieve better frequency accuracy, the RTC counter can be adjusted by software.

The built in RTC is designed to generate the alarm interrupt and periodic interrupt signals. The period interrupt can be 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second. The alarm interrupt indicates that time counter and calendar counter have counted to a specified time recorded in TAR and CAR.

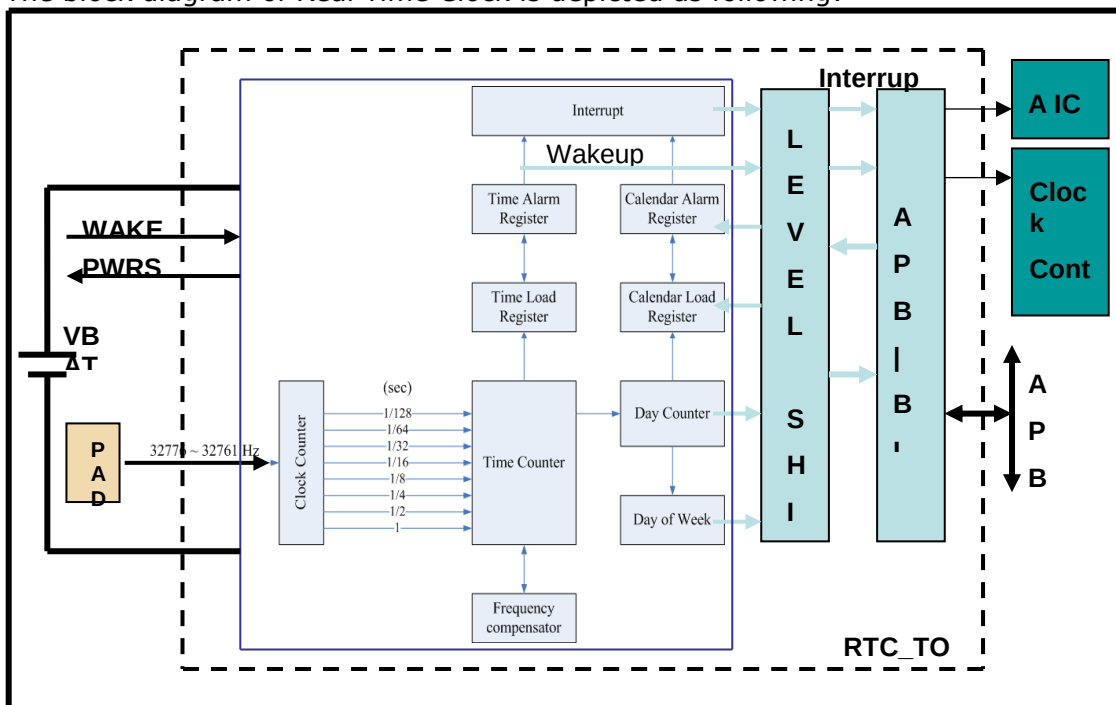
The wake up signal is used to wake the system up from sleep mode.

5.18.2 RTC Features

- There is a time counter (second, minute, hour) and calendar counter (day, month, year) for user to check the time.
- Alarm register (second, minute, hour, day, month, year).
- 12-hour or 24-hour mode is selectable.
- Recognize leap year automatically.
- The day of week counter.
- Frequency compensate register (FCR).
- Beside FCR, all clock and alarm data expressed in BCD code.
- Support time tick interrupt.
- Support wake up function.

5.18.3 RTC Block Diagram

The block diagram of Real Time Clock is depicted as following:



Note: PWRSWN = 1, all system turn on.

Note: WAKEUPN pin, this pin is within RTC parts.

5.18.4 RTC Function Description

RTC Initiation

When RTC block is power on, programmer has to write a number (0xa5eb1357) to INIR to reset all logic. INIR act as hardware reset circuit. Once INIR has been set as 0xa5eb1357, there is no action for RTC if any value be programmed into INIR register.

RTC Read/Write Enable

Register AER bit 15~0 is served as RTC read/write password. It is used to avoid signal interference from system during system power off. AER bit 15~0 has to be set as 0xa965 after system power on. Once it is set, it will take effect 128 RTC clocks later (about 4ms). Programmer can read AER bit 16 to find out whether RTC register can be accessed.

Frequency Compensation

The RTC FCR allows software control digital compensation of a 32.768 KHz crystal oscillator. User can utilize a frequency counter to measure RTC clock in one of GPIO pin during manufacture, and store the value in Flash memory for retrieval when the product is first power on.

Time and Calendar counter

TLR and CLR are used to load the time and calendar. TAR and CAR are used for alarm. They are all represented by BCD.

12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on TSSR bit 0.

Day of the week counter

Count from Sunday to Saturday.

Time tick interrupt

RTC block use a counter to calibrate the time tick count value. When the value in counter reaches zero, RTC will issue an interrupt.

RTC register property

When system power is off but RTC power is on, data stored in RTC registers will not lost except RIER and RIIR. Because of clock difference between RTC clock and system clock, when user write new data to any one of the registers, the register will not be updated until 2 RTC clocks later (60us). Hence programmer should consider about access sequence between TSSR, TAR and TLR.

In addition, user must be aware that RTC block does not check whether loaded data is out of bounds or not. RTC does not check rationality between DWR and CLR either.

Note:

1. TAR, CAR, TLR and CLR registers are all BCD counter.
2. Programmer has to make sure that the loaded values are reasonable,
3. For example, Load CLR as 201a (year), 13 (month), 00 (day), or CLR does not match with DWR, etc.
4. Reset state :

Register	Reset State
AER	0(RTC read/write disable)
CLR	05, 1, 1 (2005-1-1)
TLR	00 hr: 00 min: 00 sec
CAR	00/00/00
TAR	00:00:00
TSSR	1 (24 hr mode)
DWR	6 (Saturday)
RIER	0
RIIR	0
LIR	0
TTR	0
PWRON	50000

1. FCR Calibration : Example 1 , Frequency counter measurement : 32773.65Hz (> 32768 Hz)

Integer part: 32773 => 0x8005
 $FCR\_int = 0x05 - 0x01 + 0x08 =$

0x0c

Fraction part: $0.65 \times 60 = 39 \Rightarrow 0x27$
 $FCR\_fra = 0x27$

Example 2 , Frequency counter measurement : 32765.27Hz

Publication Release Date: April 19, 2018

($\leq$ 32768 Hz)

Integer part: 32765 => 0x7ffd
FCR_int = 0x0d - 0x01 - 0x08 =

0x04

Fraction part: 0.27 x 60 = 16.2=> 0x10
FCR_fra = 0x10

- In TLR and TAR, only 2 BCD digits are used to express "year". We assume 2 BCD digits of xY denote 20xY, but not 19xY or 21xY.

5.18.5 System Power Control Flow

5.18.5.1 Normal system Power Control Flow

Step1:

User press the power key, RPWR, to makes the power control signal, PWRCE pin, to high. If the PWR_ON bit, PTOUT[0], be set, the power key can be released and the PWRCE will keep on. If the PWR_ON bit, PTOUT[0], doesn't be set as 1, the PWRCE will back to low when the power key is released.

Step 2:

If there is another pulse on power key when the PWR_ON bit is set, the system will get an interrupt signal. User can decide to clear the PWR_ON or not. If this bit is clear, the PWRCE will go to low to turn off the core power. If the PWRON bit is also kept high, the PWRCE pin will keep in high level.

If there is not any pulse on the power key and the PWR_ON bit is clear by user, the PWRCE pin is also be set to low at this time.

The follow table is the system power control flow true table.

Input			Output	Note
X1	X2	X3	Y	
PWRKey	PWR_ON	RST_	PWCE	
1	0	0	0	RTC powered only (Default state)
0	0	X	1	Press key, Power On
0	1	1	1	keep key & S/W Set X2, Power On
1	1	1	1	Left key, Power keep On
0	1	1	1	Press key, get INT, intend to power Off
1	0	1	0	Left key & S/W clean X2, power Off Or S/W clean X2 , don't need press key, power off
X	1	0	1	RST_ active, still keep power when X2=1

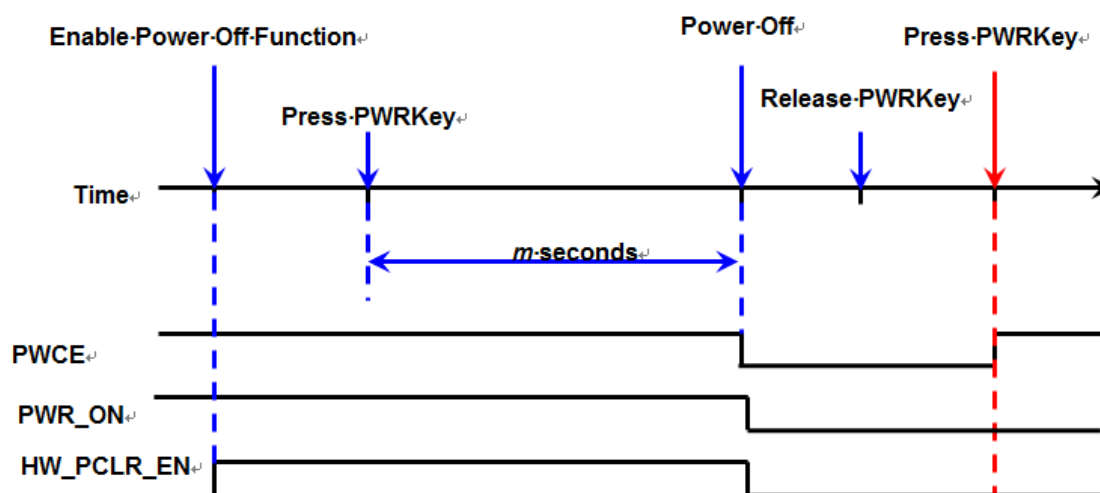
PWCE is open drain output
X1, internal pull-up
X2, it is R/W able
There is Interrupt from key be pressed

5.18.6 Force system Power Off Control Flow

The RTC supports a hardware automatic power off function and a software power off function like Notebook. For hardware power off function, it can be enable and disable in HW_PCLR_EN bit and the user presses the power button for a few seconds to power off system. The time to press power the button to power off is configured in PCLR_TIME.

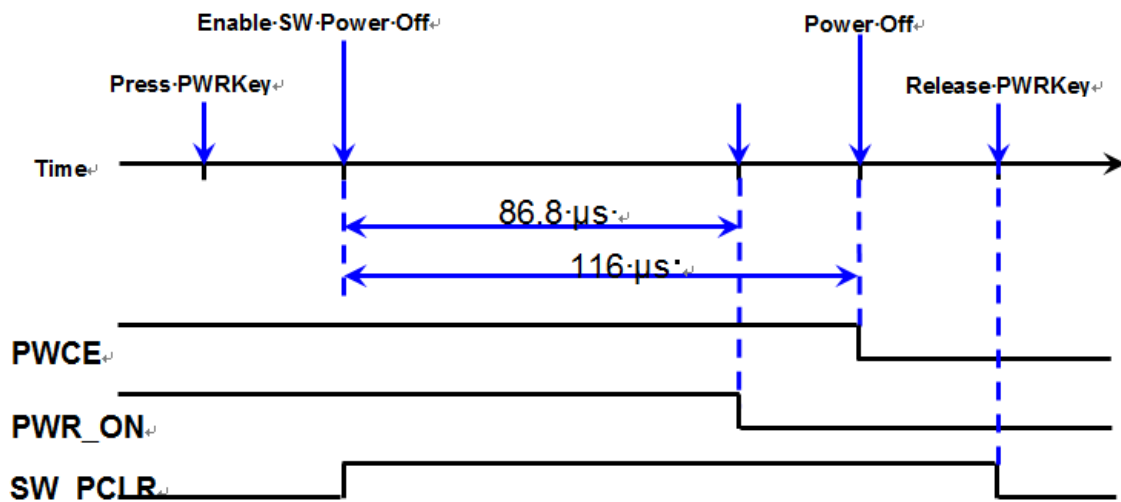
PCLR_TIME Setting	Pressed Time To Power Off	PCLR_TIME Setting	Pressed Time To Power Off
0	Power off right away	8	7~8 seconds
1	0~1 second	9	8~9 seconds
2	1~2 seconds	10	9~10 seconds
3	2~3 seconds	11	10~11 seconds
4	3~4 seconds	12	11~12 seconds
5	4~5 seconds	13	12~13 seconds
6	5~6 seconds	14	13~14 seconds
7	6~7 seconds	15	14~15 seconds

The timing of the hardware power off function is following



The RTC also supports a software power off function to provide the power off flow like Notebook. The user presses the power button for a few seconds to power off the system. The time to press power key to power off is counted by user. When the PWR_ON bit is cleared by user, the PWRCE outputs low after 116us and the SW_PCLR bit is cleared when the power key is released. See the timing Figure as

following.



5.18.7 Register Map

Register	Address	R/W	Description	Reset Value
RTC_BA = 0xB800_3000				
INIR	RTC_BA+0x000	R/W	RTC Initiation Register	0x0000_0000
AER	RTC_BA+0x004	R/W	RTC Access Enable Register	0x0000_0000
FCR	RTC_BA+0x008	R/W	RTC Frequency Compensation Register	0x0000_0700
TLR	RTC_BA+0x00C	R/W	Time Loading Register	0x0000_0000
CLR	RTC_BA+0x010	R/W	Calendar Loading Register	0x0005_0101
TSSR	RTC_BA+0x014	R/W	Time Scale Selection Register	0x0000_0000
DWR	RTC_BA+0x018	R/W	Day of the Week Register	0x0000_0006
TAR	RTC_BA+0x01C	R/W	Time Alarm Register	0x0000_0000
CAR	RTC_BA+0x020	R/W	Calendar Alarm Register	0x0000_0000
LIR	RTC_BA+0x024	R	Leap year Indicator Register	0x0000_0000
RIER	RTC_BA+0x028	R/W	RTC Interrupt Enable Register	0x0000_0000
RIIR	RTC_BA+0x02C	R/C	RTC Interrupt Indicator Register	0x0000_0000
TTR	RTC_BA+0x030	R/W	RTC Time Tick Register	0x0000_0000
PWRON	RTC_BA+0x034	R/W	RTC Power Time Out Register	0x0005_0000

5.18.8 Register Description

RTC Initiation Register (INIR)

Register	Address	R/W/C	Description	Reset Value
INIR	RTC_BA+0x000	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
INIR							
23	22	21	20	19	18	17	16
INIR							
15	14	13	12	11	10	9	8
INIR							
7	6	5	4	3	2	1	0
INIR							INIR/Active

Bits	Descriptions	
[0]	Active	RTC Active Status (Read only), 0: RTC is at reset state 1: RTC is at normal active state.
[31:0]	INIR	RTC Initiation When RTC block is power on, RTC is at reset state; programmer has to write a number (0x a5eb1357) to INIR to release all of logic and counters. INIR act as hardware reset circuit.

RTC Access Enable Register (AER)

Register	Address	R/W/C	Description	Reset Value
AER	RTC_BA+0x004	R/W	RTC Access Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							ENF
15	14	13	12	11	10	9	8
AER							
7	6	5	4	3	2	1	0
AER							

Bits	Descriptions	
[16]	ENF	RTC Register Access Enable Flag (Read only) 1: RTC register read/write enable 0: RTC register read/write disable This bit will be set after AER[15:0] register is load a 0xA965, and be clear automatically in a long time or AER[15:0] is not 0xA965.
[15:0]	AER	RTC Register Access Enable Password (Write only) 0xA965: access enable Others: access disable

RTC Frequency Compensation Register (FCR)

Register	Address	R/W/C	Description	Reset Value
FCR	RTC_BA+0x008	R/W	Frequency Compensation Register	0x0000_0700

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				INTEGER			
7	6	5	4	3	2	1	0
Reserved		FRACTION					

Bits	Descriptions			
[11:8]	INTEGER	Integer Part		
		Integer part of detected value	FCR[11:8]	Integer part of detected value
		32776	1111	32768
		32775	1110	32767
		32774	1101	32766
		32773	1100	32765
		32772	1011	32764
		32771	1010	32763
		32770	1001	32762
		32769	1000	32761
[5:0]	FRACTION	Fraction Part Formula = (fraction part of detected value) x 60 Note: Digit in FCR must be expressed as hexadecimal number.		

Note: This register can be read back after the RTC access enable (AER) is active.

RTC Time Loading Register (TLR)

Register	Address	R/W/C	Description	Reset Value
TLR	RTC_BA+0x00C	R/W	Time Loading Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		10HR		1HR			
15	14	13	12	11	10	9	8
Reserved	10MIN			1MIN			
7	6	5	4	3	2	1	0
Reserved	10SEC			1SEC			

Bits	Descriptions	
[21:20]	10HR	10 Hour Time Digit
[19:16]	1HR	1 Hour Time Digit
[14:12]	10MIN	10 Min Time Digit
[11:8]	1MIN	1 Min Time Digit
[6:4]	10SEC	10 Sec Time Digit
[3:0]	1SEC	1 Sec Time Digit

Notes: TLR is a BCD digit counter and RTC will not check loaded data.

RTC Calendar Loading Register (CLR)

Register	Address	R/W/C	Description	Reset Value
CLR	RTC_BA+0x010	R/W	Calendar Loading Register	0x0005_0101

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
10YEAR				1YEAR			
15	14	13	12	11	10	9	8
Reserved			10MON	1MON			
7	6	5	4	3	2	1	0
Reserved		10DAY		1DAY			

Bits	Descriptions	
[23:20]	10YEAR	10-Year Calendar Digit
[19:16]	1YEAR	1-Year Calendar Digit
[12]	10MON	10-Month Calendar Digit
[11:8]	1MON	1-Month Calendar Digit
[5:4]	10DAY	10-Day Calendar Digit
[3:0]	1DAY	1-Day Calendar Digit

Notes: CLR is a BCD digit counter and RTC will not check loaded data.

RTC Time Scale Selection Register (TSSR)

Register	Address	R/W/C	Description	Reset Value
TSSR	RTC_BA+0x014	R/W	Time Scale Selection Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							24hr/12hr

Bits	Descriptions			
[0]	24hr/12hr	24-Hour / 12-Hour Mode Selection		
		It indicate that TLR and TAR are in 24-hour mode or 12-hour mode		
		1: select 24-hour time scale		
		0: select 12-hour time scale with AM and PM indication		
		24-hour time scale	12-hour time scale	24-hour time scale
		00	12(AM12)	12
		01	01(AM01)	13
		02	02(AM02)	14
		03	03(AM03)	15
		04	04(AM04)	16
		05	05(AM05)	17
		06	06(AM06)	18
		07	07(AM07)	19
		08	08(AM08)	20
		09	09(AM09)	21
		10	10(AM10)	22
		11	11(AM11)	23
				31(PM11)

Note: This register can be read back after the RTC access enable (AER) is active and this register shall be programmed once time after the INIR reset done.

RTC Day of the Week Register (DWR)

Register	Address	R/W/C	Description	Reset Value
DWR	RTC_BA+0x018	R/W	Day of the Week Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					DWR		

Bits	Descriptions	
[2:0]	DWR	Day of the Week Register
		0 Sunday
		1 Monday
		2 Tuesday
		3 Wednesday
		4 Thursday
		5 Friday
		6 Saturday

RTC Time Alarm Register (TAR)

Register	Address	R/W/C	Description	Reset Value
TAR	RTC_BA+0x01C	R/W	Time Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		10HR		1HR			
15	14	13	12	11	10	9	8
Reserved	10MIN			1MIN			
7	6	5	4	3	2	1	0
Reserved	10SEC			1SEC			

Bits	Descriptions	
[21:20]	10HR	10 Hour Time Digit
[19:16]	1HR	1 Hour Time Digit
[14:12]	10MIN	10 Min Time Digit
[11:8]	1MIN	1 Min Time Digit
[6:4]	10SEC	10 Sec Time Digit
[3:0]	1SEC	1 Sec Time Digit

Notes: 1. TAR is a BCD digit counter and RTC will not check loaded data.
 2. This register can be read back after the RTC access enable (AER) is active.

RTC Calendar Alarm Register (CAR)

Register	Address	R/W/C	Description	Reset Value
CAR	RTC_BA+0x020	R/W	Calendar Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
10YEAR				1YEAR			
15	14	13	12	11	10	9	8
Reserved			10MON	1MON			
7	6	5	4	3	2	1	0
Reserved		10DAY		1DAY			

Bits	Descriptions	
[23:20]	10YEAR	10-Year Calendar Digit
[19:16]	1YEAR	1-Year Calendar Digit
[12]	10MON	10-Month Calendar Digit
[11:8]	1MON	1-Month Calendar Digit
[5:4]	10DAY	10-Day Calendar Digit
[3:0]	1DAY	1-Day Calendar Digit

Notes: 1. CAR is a BCD digit counter and RTC will not check loaded data.

2. This register can be read back after the RTC access enable (AER) is active.

RTC Leap Year Indication Register (LIR)

Register	Address	R/W/C	Description	Reset Value
LIR	RTC_BA+0x024	R	RTC Leap year Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							LIR

Bits	Descriptions	
[0]	LIR	Leap Year Indication REGISTER (Real only). 1 : It indicate that this year is leap year 0 : It indicate that this year is not a leap year

RTC Interrupt Enable Register (RIER)

Register	Address	R/W/C	Description	Reset Value
RIER	RTC_BA+0x028	R/W	RTC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					PSWIER	TIER	AIER

Bits	Descriptions	
[2]	PSWIER	Power Switch Interrupt Enable 1 => Power Switch Be Pressed Interrupt Enable 0 => Power Switch Be Pressed Interrupt disable
[1]	TIER	Time Tick Interrupt Enable 1 => RTC Time Tick Interrupt and counter enable 0 => RTC Time Tick Interrupt and counter disable
[0]	AIER	Alarm Interrupt Enable 1 => RTC Alarm Interrupt enable 0 => RTC Alarm Interrupt disable

RTC Interrupt Indication Register (RIIR)

Register	Address	R/W/C	Description	Reset Value
RIIR	RTC_BA+0x02C	R/C	RTC Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					PSWI	TI	AI

Bits	Descriptions	
[2]	PSWI	Power Switch Interrupt Indication 1: It indicates that there is power switch has been activated. 0: It indicates that the power switch interrupt never occurred. Software can also clear this bit after RTC interrupt has occur.
[1]	TI	RTC Time Tick Interrupt Indication 1: It indicates that time tick interrupt has been activated. 0: It indicates that time tick interrupt never occurred. Software can also clear this bit after RTC interrupt has occur.
[0]	AI	RTC Alarm Interrupt Indication 1: It indicates that time counter and calendar counter have counted to a specified time recorded in TAR and CAR. RTC alarm interrupt has been activated. 0: It indicates that alarm interrupt never occurred. Software can also clear this bit after RTC interrupt has occurred.

RTC Time Tick Register (TTR)

Register	Address	R/W/C	Description	Reset Value
TTR	RTC_BA+0x030	R/C	RTC Time Tick Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					TTR[2:0]		

Bits	Descriptions																			
[2:0]	TTR	Time Tick Register The RTC time tick is used for interrupt request.																		
		<table><tr><th>TTR[2:0]</th><th>Time tick (second)</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>1/2</td></tr><tr><td>2</td><td>1/4</td></tr><tr><td>3</td><td>1/8</td></tr><tr><td>4</td><td>1/16</td></tr><tr><td>5</td><td>1/32</td></tr><tr><td>6</td><td>1/64</td></tr><tr><td>7</td><td>1/128</td></tr></table>	TTR[2:0]	Time tick (second)	0	1	1	1/2	2	1/4	3	1/8	4	1/16	5	1/32	6	1/64	7	1/128
		TTR[2:0]	Time tick (second)																	
		0	1																	
		1	1/2																	
		2	1/4																	
		3	1/8																	
		4	1/16																	
		5	1/32																	
		6	1/64																	
7	1/128																			
Note: This register can be read back after the RTC enable is active.																				

Note: This register can be read back after the RTC access enable (AER) is active.

RTC Power Time Out Register (PWRON)

Register	Address	R/W/C	Description	Reset Value
PWRON	RTC_BA+0x034	R/W	RTC Power Time On Register	0x0005_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PCLR_TIME[3:0]			
15	14	13	12	11	10	9	8
SW_STATUS[7:0]							
7	6	5	4	3	2	1	0
PWR_KEY	Reserved				HW_PCLR_EN	SW_PCLR	PWR_ON

Bits	Descriptions	
[31:20]	RESERVED	RESERVED
[19:16]	PCLR_TIME	Power Clear Period The PCLR_TIME indicate the period of the power core will be cleared after the power key is pressed. Its time scalar is one second so that the default is 5 second.
[15:8]	SW_STATUS	SW_STATUS These bits are used to storage the software information.
[7]	PWR_KEY	Power Key Status 1: Indicated the power key status is high 0: Indicated the power key is pressed to low.
[6:3]	RESERVED	RESERVED
[2]	HW_PCLR_EN	Hardware Power Clear Enable 1: If this bit is set to 1, the RPWR pin will clear to low when the power key is pressed over the PCLR_TIME second and this bit will be cleared automatically at this time. 0: If this bit is set to 0, the RPWR pin won't be influenced by the pressed time of power key.
[1]	SW_PCLR	Software Core Power Disable If the power key is pressed, the RPWR pin can be clear by setting this bit and this can be cleared to 0 when the pressed power key, RPWR is released. If the power doesn't be pressed, it is not use to set this bit. 1: Force the RPWR to low.
[0]	PWR_ON	Power ON PWRCE will change to high state when PWR_ON value change from 0 to 1. Note 1: In RTC there is a register mapping to PWR_ON. Note 2: Below conditions will make PWRCE low - Set PWR_ON bit to 0 - HW_PCLR_EN is set to 1 and the power key is pressed over the period of PCLR_TIME. Note: This register can be read back after the RTC enable is active. PWRCE is the output signal in section 1.1.5 true table

Note: This register can be read back after the RTC access enable (AER) is active.

5.19 I2C Synchronous Serial Interface Controller

5.19.1 Overview

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.

Serial, 8-bit oriented bi-directional data transfers can be made up to 100 kbit/s in Standard-mode, up to 400 kbit/s in the Fast-mode, or up to 3.4 Mbit/s in the High-speed mode. Only 100kbps and 400kbps modes are supported directly. For High-speed mode special IOs are needed. If these IOs are available and used, then High-speed mode is also supported.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a **byte-by-byte** basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the **MSB being transmitted first**. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP).

5.19.2 Features

- AMBA APB interface compatible
- Compatible with Philips I²C standard, support master mode
- Multi Master Operation
- Clock stretching and wait state generation
- Provide multi-byte transmit operation, up to 4 bytes can be transmitted in a single transfer
- Software programmable acknowledge bit
- Arbitration lost interrupt, with automatic transfer cancellation
- Start/Stop/Repeated Start/Acknowledge generation
- Start/Stop/Repeated Start detection
- Bus busy detection
- Supports 7 bit addressing mode
- Fully static synchronous design with one clock domain
- Software mode I²C

5.19.3 I2C Serial Interface Block Diagram

The block diagram of I²C Serial Interface controller is shown as following.

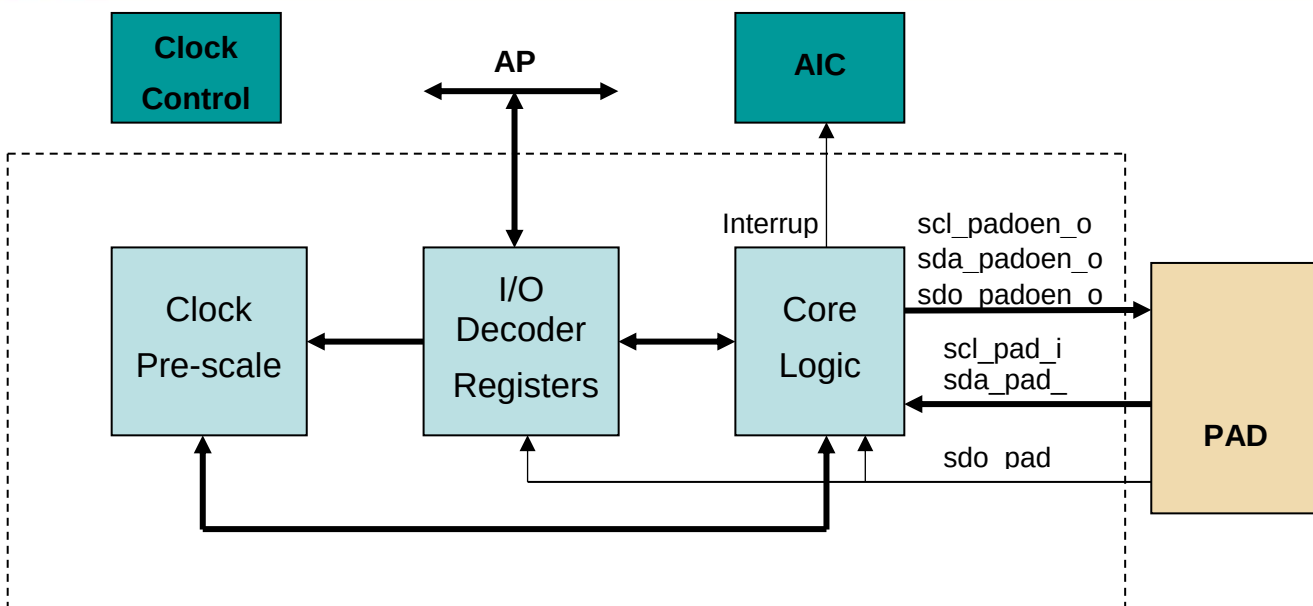


Figure 5.19-1 I²C Block Diagram

NOTE1: scl_pad_o, sda_pad_o and sdo_pad_o are always tied to 1'b0.

NOTE2: scl_padoen_o, sda_padoen_o and sdo_padoen_o are active low signals.

5.19.4 I²C Protocol

Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- Slave address transfer
- Data transfer
- STOP signal generation

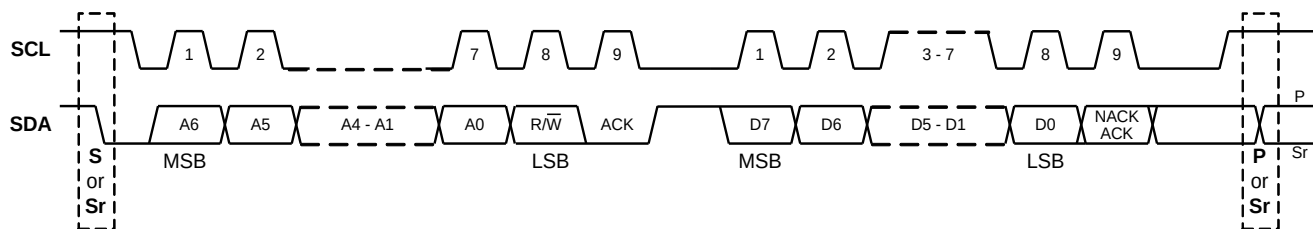
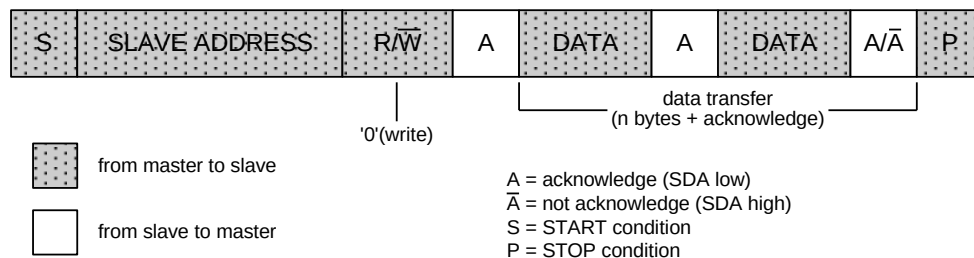
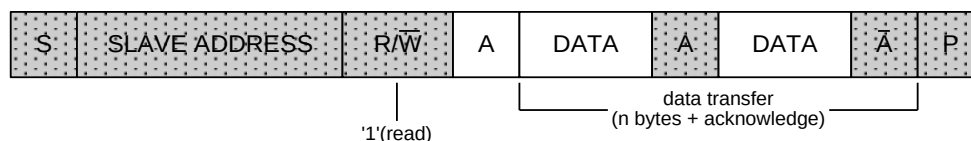


Figure 5.19-2 Data transfer on the I²C-bus



A master-transmitter addressing a slave receiver with a 7-bit address

The transfer direction is not changed



A master reads a slave immediately after the first byte (address)

5.19.5 START or Repeated START signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the **S-bit**, is defined as a **HIGH to LOW** transition on the SDA line while SCL is **HIGH**. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

The I²C core generates a START signal when the START bit in the Command Register (CMDR) is set and the READ or WRITE bits are also set. Depending on the current status of the SCL line, a START or Repeated START is generated.

5.19.5.1 STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the **P-bit**, is defined as a **LOW to HIGH** transition on the SDA line while SCL is **HIGH**.

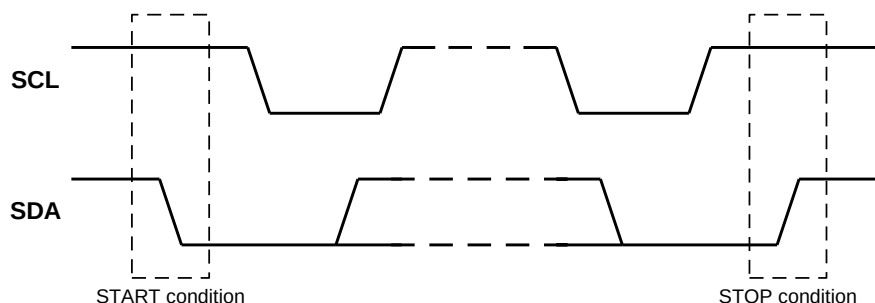


Figure 5.19-3 START and STOP conditions

5.19.5.2 Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bits calling address followed by a RW bit. The RW bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

The core treats a Slave Address Transfer as any other write action. Store the slave device's address in the Transmit Register (TxR) and set the WRITE bit. The core will then transfer the slave address on the bus.

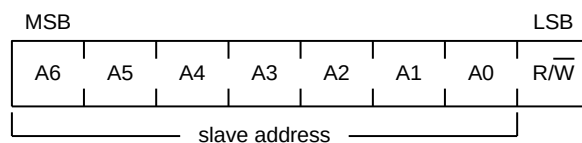


Figure 5.19-4 The first byte after the START procedure

5.19.5.3 Data Transfer

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a **Not Acknowledge (NACK)**, the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does **Not Acknowledge (NACK)** the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

To write data to a slave, store the data to be transmitted in the Transmit Register (TxR) and set the WRITE bit. To read data from a slave, set the READ bit. During a transfer the core set the I2C_TIP flag, indicating that a **Transfer is In Progress**. When the transfer is done the I2C_TIP flag is cleared, the IF flag set if enabled, then an interrupt generated. The Receive Register (RxR) contains valid data after the IF flag has been set. The software may issue a new write or read command when the I2C_TIP flag is cleared.

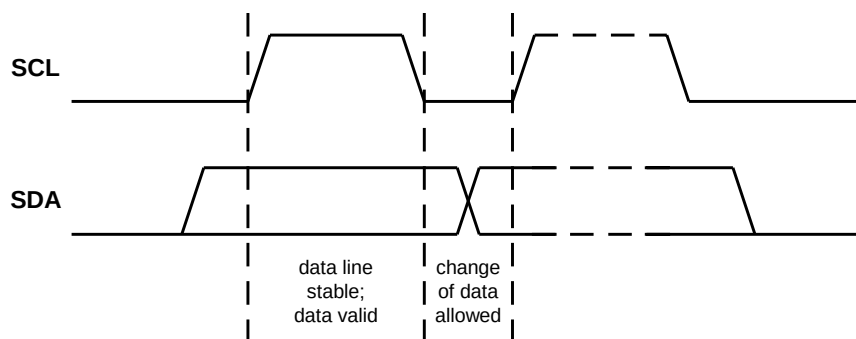


Figure 5.19-5 Bit transfer on the I²C-bus

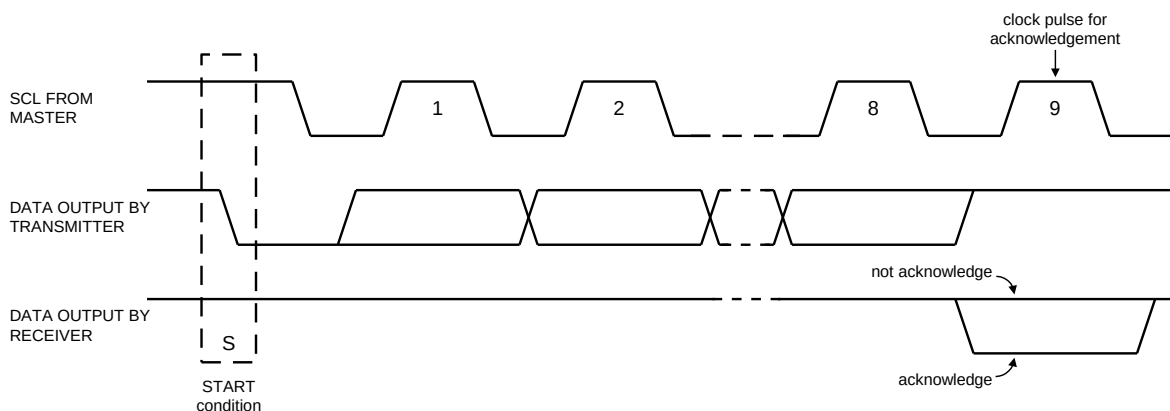


Figure 5.19-6 Acknowledge on the I²C-bus

5.19.6 I2C Programming Examples

5.19.6.1 Example 1

Write 1 byte of data to a slave (using multi-byte transmit mode).

Slave address = 0x51 (7b'1010001)

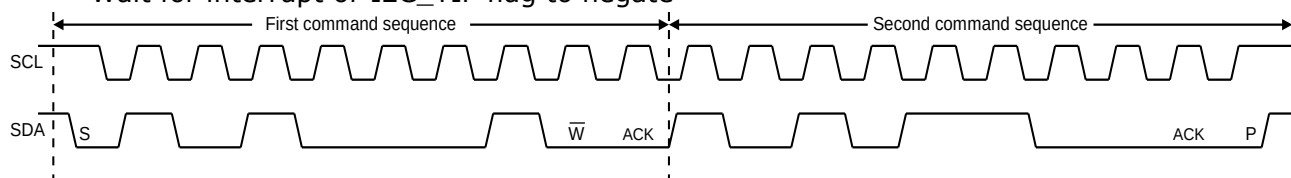
Data to write = 0xAC

I²C Sequence:

- 1) generate start command
- 2) write slave address + write bit
- 3) receive acknowledge from slave
- 4) write data
- 5) receive acknowledge from slave
- 6) generate stop command

Commands:

- 1) Write a value into DIVIDER to determine the frequency of serial clock.
- 2) Set Tx_NUM = 0x1 and set I2C_EN = 1 to enable I²C core.
- 3) Write 0xA2 (address + write bit) to Transmit Register (TxR[15:8]) and 0xAC to TxR[7:0].
- 4) Set START bit and WRITE bit.
- Wait for interrupt or I2C_TIP flag to negate --
- 5) Read I2C_RxACK bit from CSR Register, it should be '0'.
- 6) Set Tx_NUM = 0x0.
- 7) Set STOP bit.
- Wait for interrupt or I2C_TIP flag to negate --



NOTE: Please note that the time for the Interrupt Service Routine is not shown here. It is assumed that the ISR is much faster than the I²C cycle time, and therefore not visible.

5.19.6.2 Example 2

Read a byte of data from an I2C memory device (using single byte transfer mode).

Slave address = 0x4E (7b'1001110)

Memory location to read from = 0x20

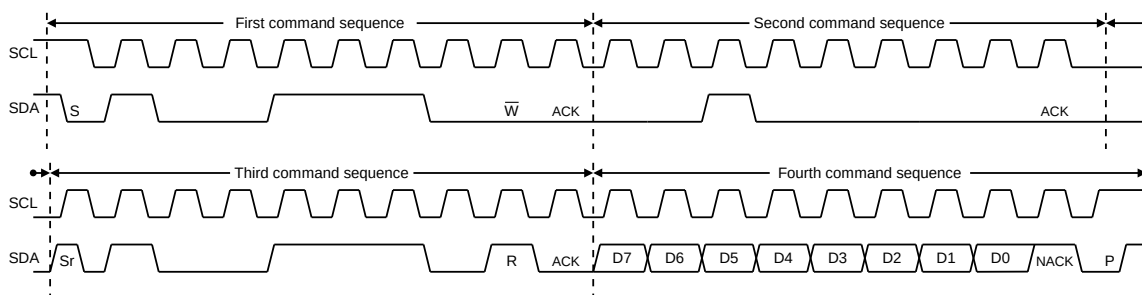
I2C sequence:

1. generate start signal
2. write slave address + write bit, then receive acknowledge from slave
3. write memory location, then receive acknowledge from slave
4. generate repeated start signal

5. write slave address + read bit, then receive acknowledge from slave
6. read byte from slave
7. write not acknowledge (NACK) to slave, indicating end of transfer
8. generate stop signal

Commands:

1. Write a value into DIVIDER to determine the frequency of serial clock.
2. Set Tx_NUM = 0x0 and set I2C_EN = 1 to enable I²C core.
3. Write 0x9C (address + write bit) to TxR[7:0], set START bit and WRITE bit.
— Wait for interrupt or I2C_TIP flag to negate —
4. Read I2C_RxACK bit from CSR Register, it should be '0'.
5. Write 0x20 to TxR[7:0], set WRITE bit.
— Wait for interrupt or I2C_TIP flag to negate —
6. Read I2C_RxACK bit from CSR Register, it should be '0'.
7. Write 0x9D (address + read bit) to TxR[7:0], set START bit, set WRITE bit.
— Wait for interrupt or I2C_TIP flag to negate —
8. Read I2C_RxACK bit from CSR Register, it should be '0'.
9. Set READ bit, set ACK to '1' (NACK), set STOP bit.
10. Read out received data from RxR, it will put on RxR[7:0].



NOTE: Please note that the time for the Interrupt Service Routine is not shown here. It is assumed that the ISR is much faster than the I²C cycle time, and therefore not visible.

5.19.7 1.5 Software I2C Operation

The software I²C function contains 3 registers for software to control the output enable of pad actually. The implementation of software I²C is shown bellow.

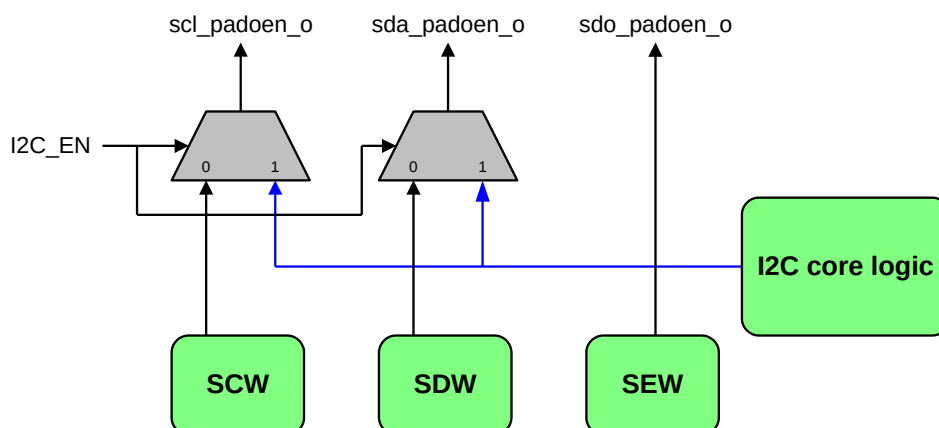


Figure 5.19-7 Implementation of Software I²C

The other three registers – SCR, SDR and SER just represent the status of input port - scl_pad_i, sda_pad_i and sdo_pad_i.

Software can read/write this register at any time, but the output enable – scl_padoen_o and sda_padoen_o are controlled by software only when I2C_EN = 0.

5.19.8 Registers Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W/C	Description	Reset Value
I2C_BA = 0xB800_4000				
CSR	I2C_BA+0x00	R/W	Control And Status Register	0x0000_0000
DIVIDER	I2C_BA+0x04	R/W	Clock Prescale Register	0x0000_0000
CMDR	I2C_BA+0x08	R/W	Command Register	0x0000_0000
SWR	I2C_BA+0x0C	R/W	Software Mode Control Register	0x0000_003F
RXR	I2C_BA+0x10	R	Data Receive Register	0x0000_0000
TXR	I2C_BA+0x14	R/W	Data Transmit Register	0x0000_0000

NOTE: The reset value of SWR is 0x3F only when SCR, SDR and SER are connected to pull high resistor.

5.19.9 Register Description

Control and Status Register (CSR)

Register	Offset	R/W/C	Description	Reset Value
CSR	0x00	R/W	Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				I2C_RxACK	I2C_BUSY	I2C_AL	I2C_TIP
7	6	5	4	3	2	1	0
Reserved		TX_NUM		Reserved	IF	IE	I2C_EN

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	I2C_RxACK	Received Acknowledge From Slave (Read only) This flag represents acknowledge from the addressed slave. 0 = Acknowledge received (ACK). 1 = Not acknowledge received (NACK).
[10]	I2C_BUSY	I ² C Bus Busy (Read only) 0 = After STOP signal detected. 1 = After START signal detected.
[9]	I2C_AL	Arbitration Lost (Read only) This bit is set when the I ² C core lost arbitration. Arbitration is lost when: A STOP signal is detected, but no requested. The master drives SDA high, but SDA is low.
[8]	I2C_TIP	Transfer In Progress (Read only) 0 = Transfer complete. 1 = Transferring data. NOTE: When a transfer is in progress, you will not allow writing to any register of the I ² C master core except SWR.
[7:6]	Reserved	Reserved

[5:4]	TX_NUM	Transmit Byte Counts These two bits represent how many bytes are remained to transmit. When a byte has been transmitted, the Tx_NUM will decrease 1 until all bytes are transmitted (Tx_NUM = 0x0) or NACK received from slave. Then the interrupt signal will assert if IE was set. 0x0 = Only one byte is left for transmission. 0x1 = Two bytes are left to for transmission. 0x2 = Three bytes are left for transmission. 0x3 = Four bytes are left for transmission. NOTE: When NACK received, TX_NUM will not decrease.
[3]	Reserved	Reserved
[2]	IF	Interrupt Flag The Interrupt Flag is set when: Transfer has been completed. Transfer has not been completed, but slave responded NACK (in multi-byte transmit mode). Arbitration is lost. NOTE: This bit is read only, but can be cleared by writing 1 to this bit.
[1]	IE	Interrupt Enable 0 = Disable I²C Interrupt. 1 = Enable I²C Interrupt.
[0]	I2C_EN	I²C Core Enable 0 = Disable I²C core, serial bus outputs are controlled by SDW/SCW. 1 = Enable I²C core, serial bus outputs are controlled by I²C core.

Prescale Register (DIVIDER)

Register	Offset	R/W/C	Description	Reset Value
DIVIDER	0x04	R/W	Clock Prescale Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DIVIDER[15:8]							
7	6	5	4	3	2	1	0
DIVIDER[7:0]							

Bits	Descriptions	
[15:0]	DIVIDER	Clock Prescale Register It is used to prescale the SCL clock line. Due to the structure of the I²C interface, the core uses a 5*SCL clock internally. The prescale register must be programmed to this 5*SCL frequency (minus 1). Change the value of the prescale register only when the "I2C_EN" bit is cleared. Example: pclk = 32MHz, desired SCL = 100KHz $prescale = \frac{32MHz}{5 * 100KHz} - 1 = 63(dec) = 3F(hex)$

Command Register (CMDR)

Register	Offset	R/W/C	Description	Reset Value
CMDR	0x08	R/W	Command Register	0x0000_000x

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	START	STOP	READ	WRITE	ACK

NOTE: Software can write this register only when I2C_EN = 1.

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	START	Generate Start Condition Generate (repeated) start condition on I ² C bus.
[3]	STOP	Generate Stop Condition Generate stop condition on I ² C bus.
[2]	READ	Read Data From Slave Retrieve data from slave.
[1]	WRITE	Write Data To Slave Transmit data to slave.
[0]	ACK	Send Acknowledge To Slave When I ² C behaves as a receiver, sent ACK (ACK = '0') or NACK (ACK = '1') to slave.

NOTE: The START, STOP, READ and WRITE bits are cleared automatically while transfer finished. READ and WRITE cannot be set concurrently.

Software Mode Register (SWR)

Register	Offset	R/W/C	Description	Reset Value
SWR	0x0C	R/W	Software Mode Control Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		SER	SDR	SCR	SEW	SDW	SCW

NOTE: This register is used as software mode of I²C. Software can read/write this register no matter I2C_EN is 0 or 1. But SCL and SDA are controlled by software only when I2C_EN = 0.

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5]	SER	Serial Interface SDO Status (Read only) 0 = SDO is Low. 1 = SDO is High.
[4]	SDR	Serial Interface SDA Status (Read only) 0 = SDA is Low. 1 = SDA is High.
[3]	SCR	Serial Interface SCK Status (Read only) 0 = SCL is Low. 1 = SCL is High.
[2]	SEW	Serial Interface SDO Output Control 0 = SDO pin is driven Low. 1 = SDO pin is tri-state.
[1]	SDW	Serial Interface SDA Output Control 0 = SDA pin is driven Low. 1 = SDA pin is tri-state.
[0]	SCW	Serial Interface SCK Output Control 0 = SCL pin is driven Low. 1 = SCL pin is tri-state.

Data Receive Register (RXR)

Register	Offset	R/W/C	Description	Reset Value
RxR	0x10	R	Data Receive Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
RX [7:0]							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	RX	Data Receive Register The last byte received via I ² C bus will put on this register. The I ² C core only used 8-bit receive buffer.

Data Transmit Register (TXR)

Register	Offset	R/W/C	Description	Reset Value
TxR	0x14	R/W	Data Transmit Register	0x0000_0000

31	30	29	28	27	26	25	24
TX [31:24]							
23	22	21	20	19	18	17	16
TX [23:16]							
15	14	13	12	11	10	9	8
TX [15:8]							
7	6	5	4	3	2	1	0
TX [7:0]							

Bits	Descriptions	
[31:0]	TX	Data Transmit Register The I²C core used 32-bit transmit buffer and provide multi-byte transmit function. Set CSR[Tx_NUM] to a value that you want to transmit. I²C core will always issue a transfer from the highest byte first. For example, if CSR[Tx_NUM] = 0x3, Tx[31:24] will be transmitted first, then Tx[23:16], and so on. In case of a data transfer, all bits will be treated as data. In case of a slave address transfer, the first 7 bits will be treated as 7-bit address and the LSB represent the R/W bit. In this case, LSB = 1, reading from slave LSB = 0, writing to slave

5.20 PWM-Timer

5.20.1 Overview

There are 4 PWM-Timers. The 4 PWM-Timers has 2 Pre-scale, 2 clock divider, 4 clock selectors, 4 16-bit counters, 4 16-bit comparators, 2 Dead-Zone generators. They are all driven by system clock. Each can be used as a timer and issues interrupt independently.

Each two PWM-Timers share the same pre-scale (0-1 share prescale0 and 2-3 share prescale1). Clock divider provides each timer with 5 clock sources (1, 1/2, 1/4, 1/8, 1/16). Each timer receives its own clock signal from clock divider which receives clock from 8-bit pre-scale. The 16-bit counter in each timer receive clock signal from clock selector and can be used to handle one PWM period. The 16-bit comparator compares number in counter with threshold number in register loaded previously to generate PWM duty cycle.

The clock signal from clock divider is called PWM clock. Dead-Zone generator utilize PWM clock as clock source. Once Dead-Zone generator is enabled, output of two PWM-Timers are blocked. Two output pin are all used as Dead-Zone generator output signal to control off-chip power device. Dead-Zone generator 0 is used to control outputs of timer 0&1, and Dead-Zone generator 1 is used to control outputs of timer 2&3.

To prevent PWM driving output pin with unsteady waveform, 16-bit counter and 16-bit comparator are implemented with double buffering feature. User can feel free to write data to counter buffer register and comparator buffer register without generating glitch.

When 16-bit down counter reaches zero, the interrupt request is generated to inform CPU that time is up. When counter reaches zero, if counter is set as toggle mode, it is reloaded automatically and start to generate next cycle. User can set counter as one-shot mode instead of toggle mode. If counter is set as one-shot mode, counter will stop and generate one interrupt request when it reaches zero.

The value of comparator is used for pulse width modulation. The counter control logic changes the output level when down-counter value matches the value of compare register.

Each PWM-Timer includes a capture channel. The Capture 0 and PWM 0 share a timer that included in PWM 0; and the Capture 1 and PWM 1 share another timer, and etc. Therefore user must setup the PWM-Timer before turn on Capture feature. Please reference the section of PWM-Timer for more detail description of setup PWM-Timer. After enabling capture feature, the capture always latched PWM-counter to CRLR when input channel has a rising transition and latched PWM-counter to CFLR when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting CCR0[1] (Rising latch Interrupt enable) and CCR0[2] (Falling latch Interrupt enable) to decide the condition of interrupt occur. Capture channel 1 has the same feature by setting CCR0[17] and CCR0[18]. And capture channel 2 & 3 has the same feature by setting CCR1[1], CCR1[2] and CCR1[17], CCR1[18] respectively. Whenever Capture issues Interrupt 0/1/2/3, the PWM counter 0/1/2/3 will be reload at this moment.

There are only four interrupts from PWM to advanced interrupt controller (AIC). PWM 0 and Capture 0 share the same interrupt; PWM1 and Capture 1 share the same interrupt and so on. Therefore, PWM function and Capture function in the same channel cannot be used at the same time.

5.20.2 Features

- Two 8-bit pre-scales and Two clock dividers
- Four clock selectors
- Four 16-bit counters and four 16-bit comparators
- Two Dead-Zone generator
- Capture function

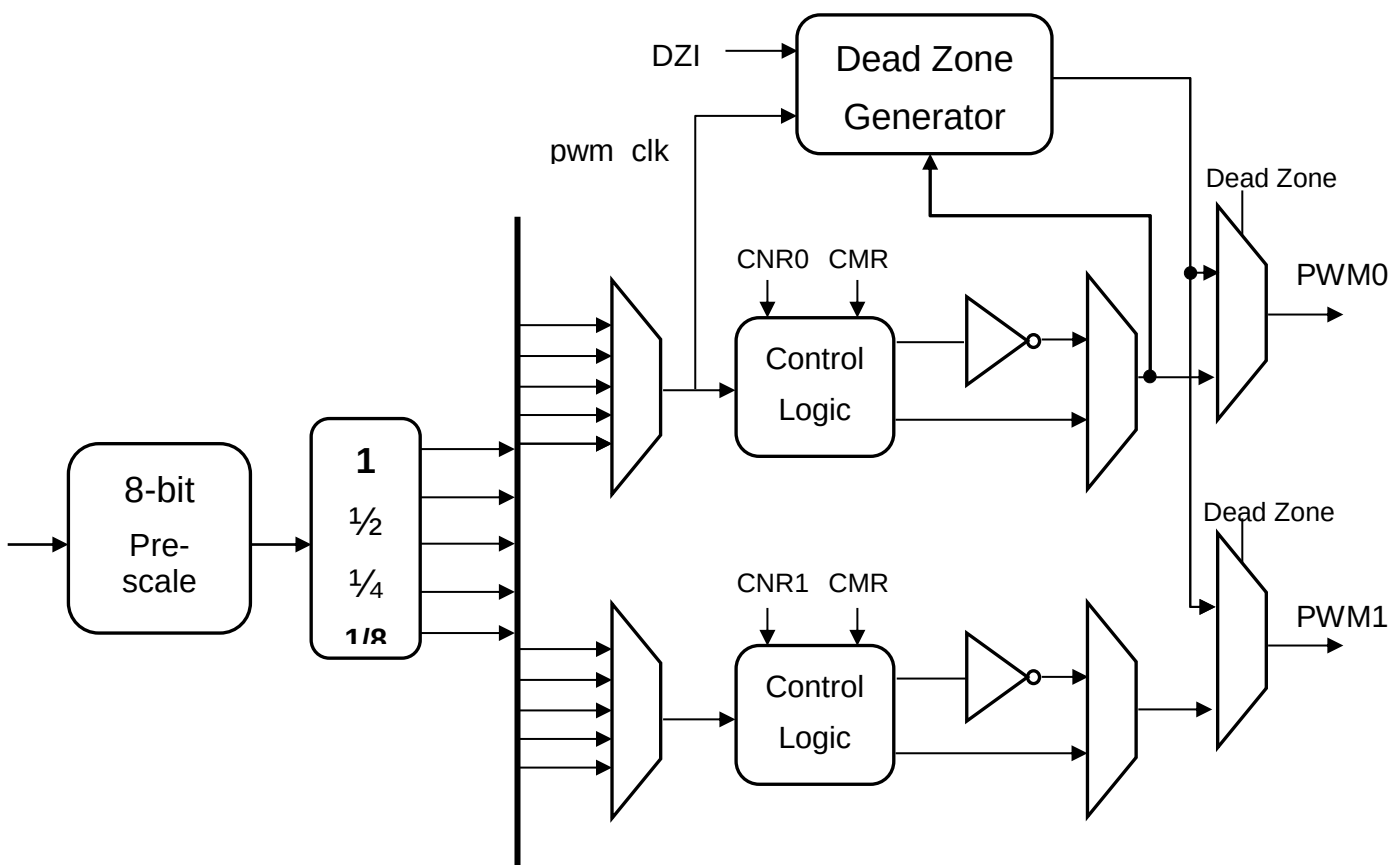
5.20.3 PWM Timer Start Procedure

1. Setup clock selector (CSR)
2. Setup pre-scale & dead zone interval (PPR)
3. Setup inverter on/off, dead zone generator on/off, toggle mode /one-shot mode, and PWM timer off. (PCR)
4. Setup comparator register (CMR)
5. Setup counter register (CNR)
6. Setup interrupt enable register (PIER)
7. Setup PWM output enable (POE)
8. Enable PWM timer (PCR)

5.20.4 PWM Architecture

PWM_OE[0] enable ----> timer PWM0 output ----> GPB[1]
 PWM_OE[1] enable ----> timer PWM1 output ----> GPB[2]
 PWM_OE[2] enable ----> timer PWM2 output ----> GPB[3]
 PWM_OE[3] enable ----> timer PWM3 output ----> GPB[4]

The following figure describes the architecture of PWM in one group. (Timer 0&1 are in one group and timer 2&3 are in another group)



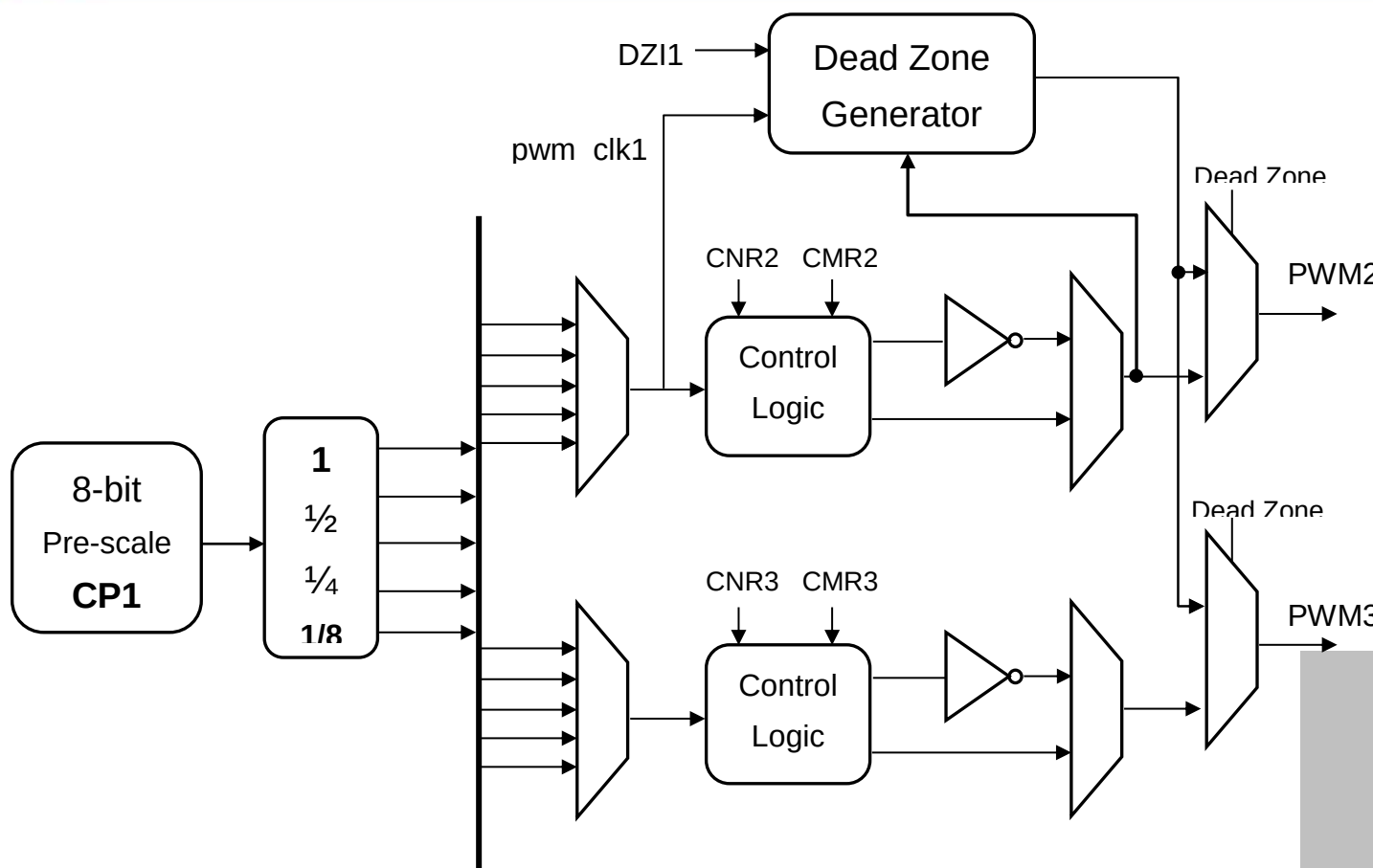


Figure 5.20-1 PWM Architecture Diagram

5.20.5 Basic Timer Operation

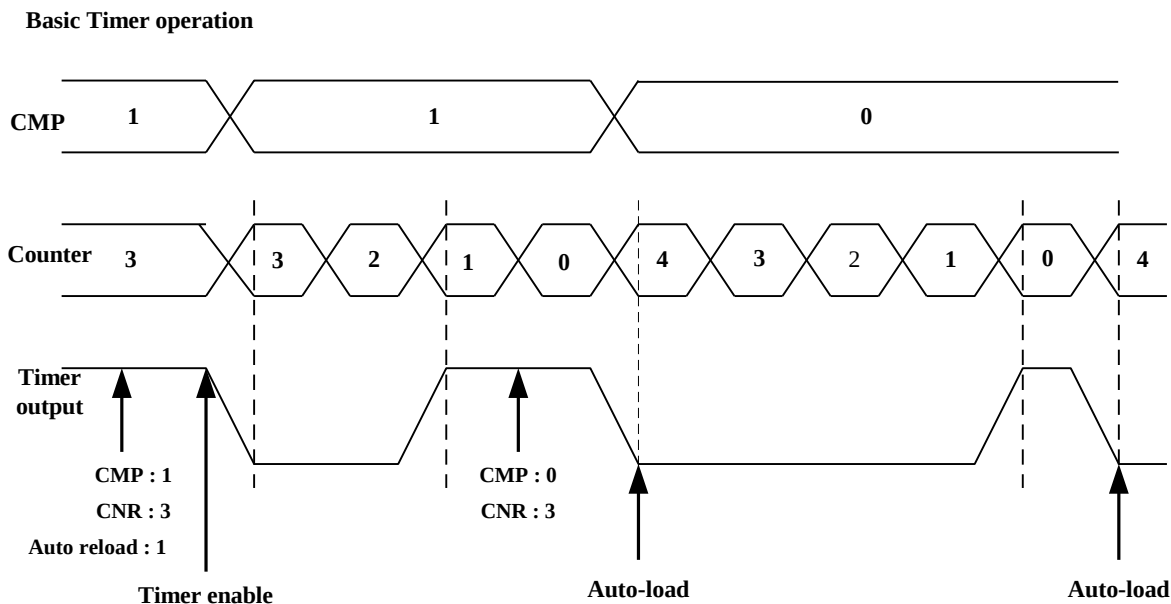


Figure 5.20-2 Basic Timer Operation Timing

5.20.6 PWM Double Buffering and Automatic Reload

PWM-Timers have a double buffering function, enabling the reload value changed for next timer operation without stopping current timer operation. Although new timer value is set, current timer operation, still operate successfully.

The counter value can be written into CNR0~3 and current counter value can be read from PDR0~3.

The auto-reload operation will copy from CNR0~3 to down-counter when down-counter reaches zero. If CNR0~3 are set as zero, counter will be halt when counter count to zero. If auto-reload bit is set as zero, counter will be stopped immediately.

PWM double buffering

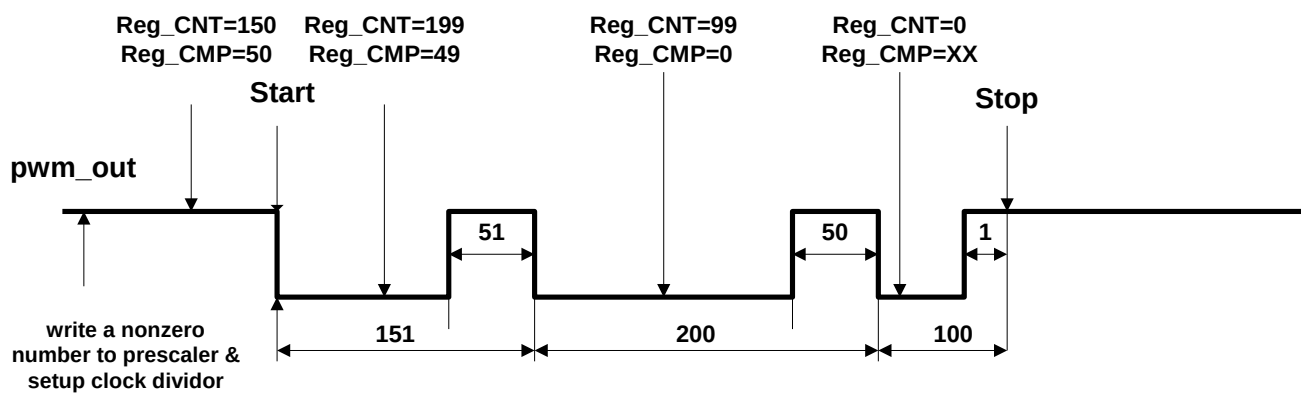


Figure 5.20-3 PWM Double Buffering Illustration

5.20.7 Modulate Duty Ratio

The double buffering function allows CMR written at any point in current cycle. The loaded value will take effect from next cycle.

Modulate PWM controller output duty ratio(CNR = 150)

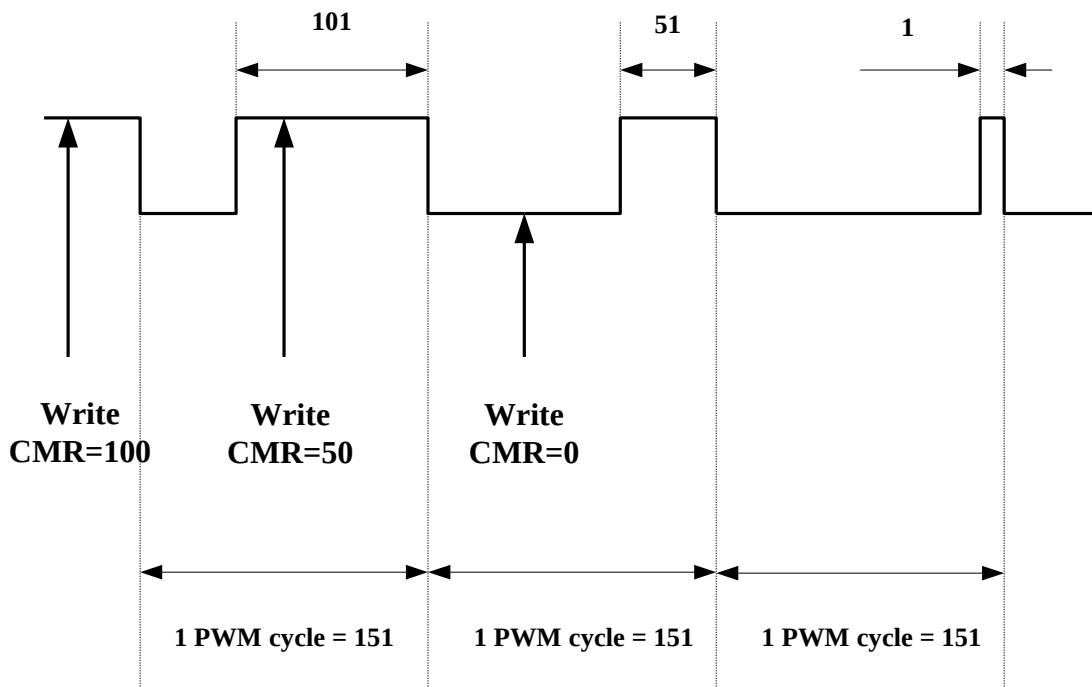


Figure 7.11-5.20-4 PWM Controller Output Duty Ratio

5.20.8 Dead-Zone Generator

PWM is implemented with Dead Zone generator. They are built for power device protection. This function enables generation of a programmable time gap at the rising of PWM output waveform. User can program PPR [31:24] and PPR [23:16] to determine the two Dead Zone interval respectively.

Dead zone generator operation

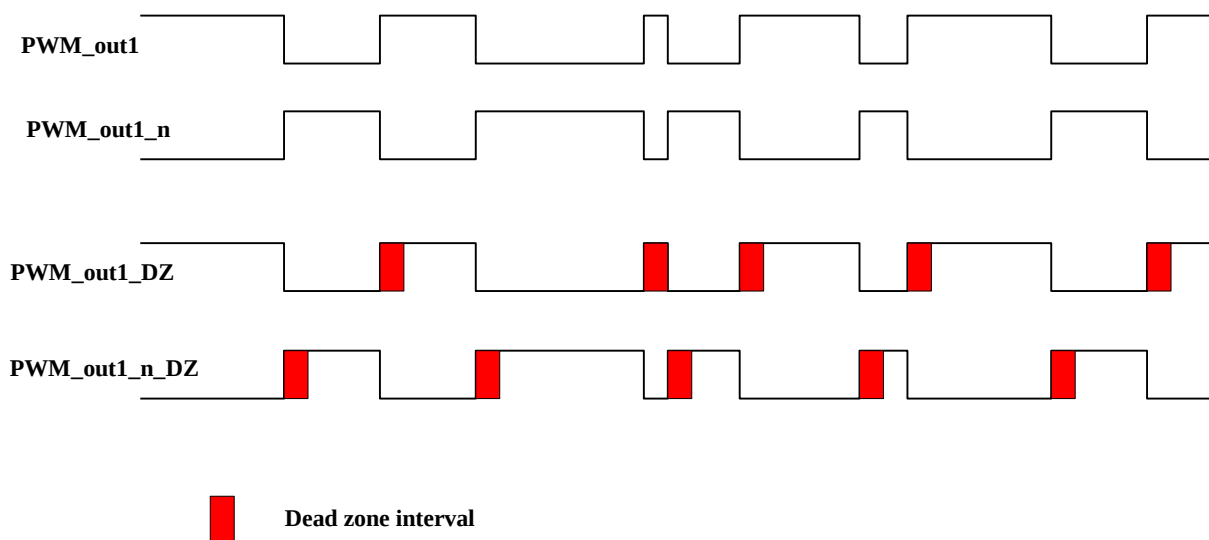


Figure 5.20-5 Dead Zone Generation Operation

5.20.9 PWM Timer Start Procedure

1. Setup clock selector (CSR)
2. Setup pre-scale & dead zone interval (PPR)
3. Setup inverter on/off, dead zone generator on/off, toggle mode /one-shot mode, and PWM timer off. (PCR)
4. Setup the comparator register (CMR)
5. Setup the counter register (CNR)
6. Setup the interrupt enable register (PIER)
7. Setup PWM output enables (POE)
8. Enable PWM timer (PCR)

5.20.10 PWM Timer Stop Procedure

Method 1:

Set 16-bit down counter (CNR) as 0, and monitor PDR. When PDR reaches to 0, disable PWM timer (PCR). **(Recommended)**

Method 2:

Set 16-bit down counter (CNR) as 0. When interrupt request happen, disable PWM timer (PCR). **(Recommended)**

Method 3:

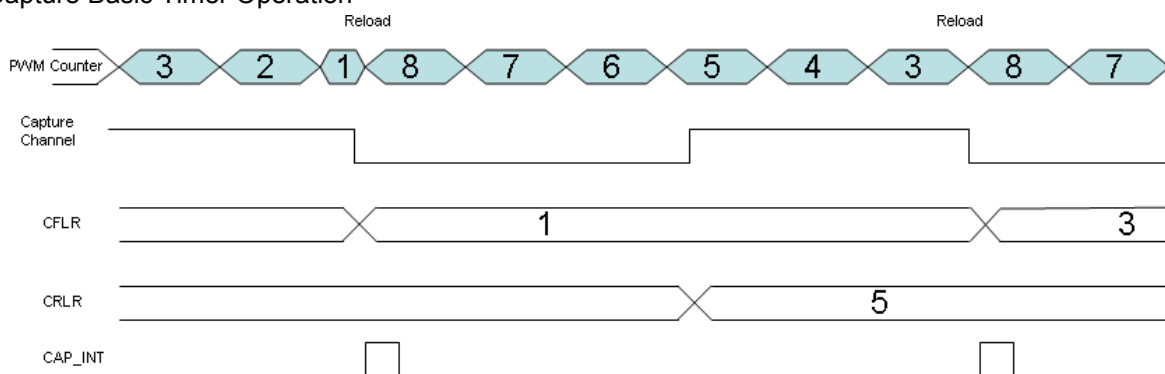
Disable PWM timer directly (PCR). (Not recommended)

Capture Start Procedure

1. Setup clock selector (CSR)

2. Setup pre-scale & dead zone interval (PPR)
3. Setup inverter on/off, dead zone generator on/off, toggle mode /one-shot mode, and PWM timer off. (PCR)
4. Setup the comparator register (CMR)
5. Setup the counter register (CNR)
6. Setup the capture register (CCR)
7. Setup PWM output enables (POE)
8. Enable PWM timer (PCR)

Capture Basic Timer Operation



At this case, the CNR is 8:

1. When set falling interrupt enable, the PWM counter will be reload at time of interrupt occur.
2. The channel low pulse width is $(CNT - CRLR)$.
3. The channel high pulse width is $(CRLR - CFLR)$.
4. The channel cycle time is $(CNR - CFLR)$.

5.20.11 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W	Description	Reset Value
PWM_BA = 0xB800_7000				
PPR	PWM_BA+0x000	R/W	PWM Pre-scale Register	0x0000_0000
CSR	PWM_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000
PCR	PWM_BA+0x008	R/W	PWM Control Register	0x0000_0000
CNR0	PWM_BA+0x00C	R/W	PWM Counter Register 0	0x0000_0000
CMR0	PWM_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000
PDR0	PWM_BA+0x014	R	PWM Data Register 0	0x0000_0000
CNR1	PWM_BA+0x018	R/W	PWM Counter Register 1	0x0000_0000
CMR1	PWM_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000
PDR1	PWM_BA+0x020	R	PWM Data Register 1	0x0000_0000
CNR2	PWM_BA+0x024	R/W	PWM Counter Register 2	0x0000_0000
CMR2	PWM_BA+0x028	R/W	PWM Comparator Register 2	0x0000_0000
PDR2	PWM_BA+0x02C	R	PWM Data Register 2	0x0000_0000
CNR3	PWM_BA+0x030	R/W	PWM Counter Register 3	0x0000_0000
CMR3	PWM_BA+0x034	R/W	PWM Comparator Register 3	0x0000_0000
PDR3	PWM_BA+0x038	R	PWM Data Register 3	0x0000_0000
PIER	PWM_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000
PIIR	PWM_BA+0x044	R/C	PWM Interrupt Indication Register	0x0000_0000
CCR0	PWM_BA+0x050	R/W	Capture Control Register 0	0x0000_0000
CCR1	PWM_BA+0x054	R/W	Capture Control Register 1	0x0000_0000
CRLR0	PWM_BA+0x058	R/W	Capture Rising Latch Register (Channel 0)	0x0000_0000
CFLR0	PWM_BA+0x05C	R/W	Capture Falling Latch Register (Channel 0)	0x0000_0000
CRLR1	PWM_BA+0x060	R/W	Capture Rising Latch Register (Channel 1)	0x0000_0000
CFLR1	PWM_BA+0x064	R/W	Capture Falling Latch Register (Channel 1)	0x0000_0000
CRLR2	PWM_BA+0x068	R/W	Capture Rising Latch Register (Channel 2)	0x0000_0000
CFLR2	PWM_BA+0x06C	R/W	Capture Falling Latch Register (Channel 2)	0x0000_0000
CRLR3	PWM_BA+0x070	R/W	Capture Rising Latch Register (Channel 3)	0x0000_0000
CFLR3	PWM_BA+0x074	R/W	Capture Falling Latch Register (Channel 3)	0x0000_0000
CAPENR	PWM_BA+0x078	R/W	Capture Input Enable Register	0x0000_0000
POE	PWM_BA+0x07C	R/W	PWM Output Enable	0x0000_0000

5.20.12 Register Description

PWM Pre-scale Register (PPR)

Register	Offset	R/W	Description	Reset Value
PPR	PWM_BA+0x000	R/W	PWM Pre-scale Register	0x0000_0000

31	30	29	28	27	26	25	24
DZI1							
23	22	21	20	19	18	17	16
DZI0							
15	14	13	12	11	10	9	8
CP1							
7	6	5	4	3	2	1	0
CP0							

Bits	Descriptions	
[31:24]	DZI1	Dead zone interval register 1 These 8-bit determine dead zone length. The 1 unit time of dead zone length is received from clock selector 1.
[23:16]	DZI0	Dead zone interval register 0 These 8-bit determine dead zone length. The 1 unit time of dead zone length is received from clock selector 0.
[15:8]	CP1	Clock pre-scale 1 for PWM Timer 2 & 3 Clock input is divided by (CP1 + 1) before it is fed to the counter. 2 & 3 If CP1=0, then the pre-scale 1 output clock will be stopped.
[7:0]	CP0	Clock pre-scale 0 for PWM Timer 0 & 1 Clock input is divided by (CP0 + 1) before it is fed to the counter. 0 & 1 If CP0=0, then the pre-scale 0 output clock will be stopped.

PWM Clock Selector Register (CSR)

Register	Offset	R/W	Description	Reset Value
CSR	PWM_BA+0x004	R/W	PWM Clock Selector Register (CSR)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	CSR3			Reserved	CSR2		
7	6	5	4	3	2	1	0
Reserved	CSR1			Reserved	CSR0		

Bits	Descriptions													
[31:15]	Reserved	Reserved												
[14:12]	CSR3	Timer 3 Clock Source Selection Select clock input for timer 3.												
		<table><tr><th>CSR3 [14:12]</th><th>Input clock divided by</th></tr><tr><td>100</td><td>1</td></tr><tr><td>011</td><td>16</td></tr><tr><td>010</td><td>8</td></tr><tr><td>001</td><td>4</td></tr><tr><td>000</td><td>2</td></tr></table>	CSR3 [14:12]	Input clock divided by	100	1	011	16	010	8	001	4	000	2
		CSR3 [14:12]	Input clock divided by											
		100	1											
		011	16											
		010	8											
		001	4											
000	2													
[11]	Reserved	Reserved												
[10:8]	CSR2	Timer 2 Clock Source Selection Select clock input for timer 0. (Table is the same as CSR3)												
[7]	Reserved	Reserved												
[6:4]	CSR1	Timer 1 Clock Source Selection Select clock input for timer 0. (Table is the same as CSR3)												
[3]	Reserved	Reserved												
[2:0]	CSR0	Timer 0 Clock Source Selection Select clock input for timer 0. (Table is the same as CSR3)												

PWM Control Register (PCR)

Register	Offset	R/W	Description	Reset Value
PCR	PWM_BA+0x008	R/W	PWM Control Register (PCR)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CH3MOD	CH3INV	Reserved	CH3EN
23	22	21	20	19	18	17	16
Reserved				CH2MOD	CH2INV	Reserved	CH2EN
15	14	13	12	11	10	9	8
Reserved				CH1MOD	CH1INV	Reserved	CH1EN
7	6	5	4	3	2	1	0
Reserved		DZEN1	DZEN0	CH0MOD	CH0INV	Reserved	CH0EN

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27]	CH3MOD	Timer 3 Toggle/One-Shot Mode 1: Toggle Mode 0: One-Shot Mode NOTE: If there is a rising transition at this bit, it will cause CNR3 and CMR3 be clear.
[26]	CH3INV	Timer 3 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[25]	Reserved	Reserved
[24]	CH3EN	Timer 3 Enable/Disable 1: Enable 0: Disable
[23:20]	Reserved	Reserved
[19]	CH2MOD	Timer 2 Toggle/One-Shot Mode 1: Toggle Mode 0: One-Shot Mode NOTE: If there is a rising transition at this bit, it will cause CNR2 and CMR2 be clear.
[18]	CH2INV	Timer 2 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[17]	Reserved	Reserved
[16]	CH2EN	Timer2 Enable/Disable 1: Enable 0: Disable

[15:10]	Reserved	Reserved
[11]	CH1MOD	Timer 1 Toggle/One-Shot Mode 1: Toggle Mode 0: One-Shot Mode NOTE: If there is a rising transition at this bit, it will cause CNR1 and CMR1 be clear.
[10]	CH1INV	Timer 1 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[9]	Reserved	Reserved
[8]	CH1EN	Timer 1 Enable/Disable 1: Enable 0: Disable
[7:6]	Reserved	Reserved
[5]	DZEN1	Dead-Zone 1 Generator Enable/Disable 1: Enable 0: Disable
[4]	DZEN0	Dead-Zone 0 Generator Enable/Disable 1: Enable 0: Disable
[3]	CH0MOD	Timer 0 Toggle/One-Shot Mode 1: Toggle Mode 0: One-Shot Mode NOTE: If there is a rising transition at this bit, it will cause CNR0 and CMR0 be clear.
[2]	CH0INV	Timer 0 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[1]	Reserved	Reserved
[0]	CH0EN	Timer 0 Enable/Disable 1: Enable 0: Disable

PWM Counter Register 3-0 (CNR3-0)

Register	Offset	R/W	Description	Reset Value
CNR0	PWM_BA+0x00C	R/W	PWM Counter Register 0	0x0000_0000
CNR1	PWM_BA+0x018	R/W	PWM Counter Register 1	0x0000_0000
CNR2	PWM_BA+0x024	R/W	PWM Counter Register 2	0x0000_0000
CNR3	PWM_BA+0x030	R/W	PWM Counter Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CNR [15:8]							
7	6	5	4	3	2	1	0
CNR [7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	CNR	PWM Counter/Timer Loaded Value Inserted data range : 65535~0 (Unit : 1 PWM clock cycle)
		Note 1: One PWM cycle width = CNR + 1. If CNR equal zero, PWM counter/timer will be stopped.
		Note 2: Programmer can feel free to write a data to CNR at any time, and it will take effect in next cycle.

PWM Comparator Register 3-0 (CMR3-0)

Register	Offset	R/W	Description	Reset Value
CMR0	PWM_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000
CMR1	PWM_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000
CMR2	PWM_BA+0x028	R/W	PWM Comparator Register 2	0x0000_0000
CMR3	PWM_BA+0x034	R/W	PWM Comparator Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CMR [15:8]							
7	6	5	4	3	2	1	0
CMR [7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	CMR	PWM Comparator Register Inserted data range : 65535~0 (Unit : 1 PWM clock cycle) CMR are used to determine PWM output duty ratio. Assumption : PWM output initial : high CMR >= CNR : PWM output is always high CMR < CNR : PWM output high => (CMR + 1) unit CMR = 0 : PWM output high => 1 unit
		Note 1: PWM duty = CMR + 1. If CMR equal zero, PWM duty = 1
		Note 2: Programmer can feel free to write a data to CMR at any time, and it will take effect in next cycle.

PWM Data Register 3-0 (PDR 3-0)

Register	Offset	R/W	Description	Reset Value
PDR0	PWM_BA+0x014	R	PWM Data Register 0	0x0000_0000
PDR1	PWM_BA+0x020	R	PWM Data Register 1	0x0000_0000
PDR2	PWM_BA+0x02C	R	PWM Data Register 1	0x0000_0000
PDR3	PWM_BA+0x038	R	PWM Data Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PDR [15:8]							
7	6	5	4	3	2	1	0
PDR [7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	PDR	PWM Data Register User can monitor PDR to know current value in 16-bit down counter.

PWM Interrupt Enable Register (PIER)

Register	Offset	R/W	Description	Reset Value
PIER	PWM_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				PIER3	PIER2	PIER1	PIER0

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	PIER3	PWM Timer 3 Interrupt Enable 1: Enable 0: Disable
[2]	PIER2	PWM Timer 2 Interrupt Enable 1: Enable 0: Disable
[1]	PIER1	PWM Timer 1 Interrupt Enable 1: Enable 0: Disable
[0]	PIER0	PWM Timer 0 Interrupt Enable 1: Enable 0: Disable

PWM Interrupt Indication Register (PIIR)

Register	Offset	R/W	Description	Reset Value
PIIR	PWM_BA+0x044	R/W	PWM Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				PIIR3	PIIR2	PIIR1	PIIR0

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	PIIR3	PWM Timer 3 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF
[2]	PIIR2	PWM Timer 2 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF
[1]	PIIR1	PWM Timer 1 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF
[0]	PIIR0	PWM Timer 0 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF

Note: User can clear each interrupt flag by writing a one to corresponding bit in PIIR.

Capture Control Register (CCR0)

Register	Offset	R/W	Description	Reset Value
CCR0	PWM_BA+0x050	R/W	Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CFLRD1	CRLRD1	Reserved	CIIR1	CAPCH1EN	FL&IE1	RL&IE1	INV1
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CFLRD0	CRLRD0	Reserved	CIIR0	CAPCH0EN	FL&IE0	RL&IE0	INV0

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23]	CFLRD1	CFLR1 dirty bit When input channel 1 has a rising transition, CFLR1 was updated and this bit was "1". Write "1" clear.
[22]	CRLRD1	CRLR1 dirty bit When input channel 1 has a falling transition, CRLR1 was updated and this bit was "1". Write "1" clear.
[21]	Reserved	Reserved
[20]	CIIR1	Capture Interrupt Indication 1 Enable/Disable 1: Interrupt Flag ON 0: Interrupt Flag OFF Note: If this bit is "1", PWM-counter 1 will not reload when next capture interrupt occur. Write "1" clear.
[19]	CAPCH1EN	Capture Channel 1 transition Enable/Disable 1: Enable 0: Disable When Enable, Capture latched the PMW-counter and saved to CRLR (Rising latch) and CFLR (Falling latch). When Disable, Capture does not update CRLR and CFLR, and disable Channel 1 Interrupt.

[18]	FL&IE1	Channel1 Falling Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 1 has falling transition, Capture issues an Interrupt.
[17]	RL&IE1	Channel 1 Rising Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 1 has rising transition, Capture issues an Interrupt.
[16]	INV1	Channel 1 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[15:8]	Reserved	Reserved
[7]	CFLRD0	CFLR0 dirty bit When input channel 0 has a falling transition, CFLR0 was updated and this bit was "1". Write "1" clear.
[6]	CRLRD0	CRLR0 dirty bit When input channel 0 has a falling transition, CRLR0 was updated and this bit was "1". Write "1" clear.
[5]	Reserved	Reserved
[4]	CIIR0	Capture Interrupt Indication 0 Enable/Disable 1: Interrupt Flag ON 0: Interrupt Flag OFF Note: If this bit is "1", PWM-counter 0 will not reload when next capture interrupt occur. Write "1" clear.
[3]	CAPCH0EN	Capture Channel 0 transition Enable/Disable 1: Enable 0: Disable When Enable, Capture latched the PWM-counter value and saved to CRLR (Rising latch) and CFLR (Falling latch). When Disable, Capture does not update CRLR and CFLR, and disable Channel 0 Interrupt.
[2]	FL&IE0	Channel 0 Falling Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 0 has falling transition, Capture issues an Interrupt.
[1]	RL&IE0	Channel 0 Rising Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 0 has rising transition, Capture issues an Interrupt.

[0]	INV0	Channel 0 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF

Capture Control Register (CCR1)

Register	Offset	R/W	Description	Reset Value
CCR1	PWM_BA+0x054	R/W	Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CFLRD3	CRLRD3	Reserved	CIIR3	CAPCH3EN	FL&IE3	RL&IE3	INV3
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CFLRD2	CRLRD2	Reserved	CIIR2	CAPCH2EN	FL&IE2	RL&IE2	INV2

Bits	Descriptions	
[31:23]	Reserved	Reserved
[23]	CFLRD3	CFLR3 dirty bit When input channel 1 has a falling transition, CFLR3 was updated and this bit was "1". Write "1" clear.
[22]	CRLRD3	CRLR3 dirty bit When input channel 1 has a falling transition, CRLR3 was updated and this bit was "1". Write "1" clear.
[21]	Reserved	Reserved
[20]	CIIR3	Capture Interrupt Indication 3 Enable/Disable 1: Interrupt Flag ON 0: Interrupt Flag OFF Note: If this bit is "1", PWM-counter 3 will not reload when next capture interrupt occur. Write "1" clear.
[19]	CAPCH3EN	Capture Channel 3 transition Enable/Disable 1: Enable 0: Disable When Enable, Capture latched the PMW-counter and saved to CRLR (Rising latch) and CFLR (Falling latch). When Disable, Capture does not update CRLR and CFLR, and disable Channel 3 Interrupt.

[18]	FL&IE3	Channel 3 Falling Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 3 has falling transition, Capture issues an Interrupt.
[17]	RL&IE3	Channel 3 Rising Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 3 has rising transition, Capture issues an Interrupt.
[16]	INV3	Channel 3 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[15:8]	Reserved	Reserved
[7]	CFLRD2	CFLR2 dirty bit When input channel 2 has a rising transition, CFLR2 was updated and this bit was "1". Write "1" clear.
[6]	CRLRD2	CRLR2 dirty bit When input channel 2 has a falling transition, CRLR2 was updated and this bit was "1". Write "1" clear.
[5]	Reserved	Reserved
[4]	CIIR2	Capture Interrupt Indication 2 Enable/Disable 1: Interrupt Flag ON 0: Interrupt Flag OFF Note: If this bit is "1", PWM-counter 2 will not reload when next capture interrupt occur. Write "1" clear.
[3]	CAPCH2EN	Capture Channel 2 transition Enable/Disable 1: Enable 0: Disable When Enable, Capture latched the PMW-counter value and saved to CRLR (Rising latch) and CFLR (Falling latch). When Disable, Capture does not update CRLR and CFLR, and disable Channel 2 Interrupt.
[2]	FL&IE2	Channel 2 Falling Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 2 has falling transition, Capture issues an Interrupt.
[1]	RL&IE2	Channel 2 Rising Interrupt Enable ON/OFF 1: Enable 0: Disable When Enable, if Capture detects Channel 2 has rising transition, Capture issues an Interrupt.

[0]	INV20	Channel 2 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF

Capture Rising Latch Register3-0 (CRLR3-0)

Register	Offset	R/W	Description	Reset Value
CRLR0	PWM_BA+0x058	R/W	Capture Rising Latch Register (channel 0)	0x0000_0000
CRLR1	PWM_BA+0x060	R/W	Capture Rising Latch Register (channel 1)	0x0000_0000
CRLR2	PWM_BA+0x068	R/W	Capture Rising Latch Register (channel 2)	0x0000_0000
CRLR3	PWM_BA+0x070	R/W	Capture Rising Latch Register (channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CRLR0 [15:8]							
7	6	5	4	3	2	1	0
CRLR0 [7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	CRLR0	Capture Rising Latch Register0 Latch the PWM counter when Channel 0 has rising transition.

Capture Falling Latch Register3-0 (CFLR3-0)

Register	Offset	R/W	Description	Reset Value
CFLR0	PWM_BA+0x05C	R/W	Capture Falling Latch Register (channel 0)	0x0000_0000
CFLR1	PWM_BA+0x064	R/W	Capture Falling Latch Register (channel 1)	0x0000_0000
CFLR2	PWM_BA+0x06C	R/W	Capture Falling Latch Register (channel 2)	0x0000_0000
CFLR3	PWM_BA+0x074	R/W	Capture Falling Latch Register (channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
CFLR0[15:8]							
7	6	5	4	3	2	1	0
CFLR0[7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	CFLR0	Capture Falling Latch Register0 Latch the PWM counter when Channel 0 has Falling transition.

Capture Input Enable Register (CAPENR)

Register	Offset	R/W	Description	Reset Value
CAPENR	PWM_BA+0x078	R/W	Capture Input Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				CAPENR[3:0]			

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3:0]	CAPENR	Capture Input Enable Register There are eight capture inputs from pad. Bit0~Bit3 are used to control each inputs ON or OFF. (At most 4 inputs can be used at the same time) 0 : OFF / 1 : ON CAPENR[3:0] <u>3210</u> xxx1 → Capture channel 0 is from GPB[1] or GPC[12] or GPD[0] xx1x → Capture channel 1 is from GPB[2] or GPC[13] or GPD[1] x1xx → Capture channel 2 is from GPB[3] or GPC[14] or GPD[2] 1xxx → Capture channel 3 is from GPB[4] or GPC[15] or GPD[3]

PWM Output Enable Register (PWM)

Register	Offset	R/W	Description	Reset Value
POE	PWM_BA+0x07C	R/W	PWM Output Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				PWM3	PWM2	PWM1	PWM0

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	PWM3	PWM timer 3 Output Enable Setup. 1 : Enable 0 : Disable
[2]	PWM2	PWM timer 2 Output Enable Setup. 1 : Enable 0 : Disable
[1]	PWM1	PWM timer 1 Output Enable Setup. 1 : Enable 0 : Disable
[0]	PWM0	PWM timer 0 Output Enable Setup. 1 : Enable 0 : Disable

5.21 UART Interface Controller

5.21.1 Overview

The N9H20 provides two channels of Universal Asynchronous Receiver/Transmitters (UART). UART0 supports High Speed UART and UART1 perform Normal Speed UART, besides, only UART0 support flow control function.

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data characters received from the peripheral, and a parallel-to-serial conversion on data characters received from the CPU. Each UART channel supports six types of interrupts including transmitter FIFO empty interrupt(Int_THRE), receiver threshold level reaching interrupt (Int_RDA), line status interrupt (overrun error or parity error or framing error or break interrupt) (Int_RLS) , time out interrupt (Int_Tout), MODEM status interrupt (Int_Modem) and Wake up status interrupt (Int_WakeUp).

The UART0 are built-in with a 64-byte transmitter FIFO (TX_FIFO) and a 64-byte that reduces the number of interrupts presented to the CPU and the UART1 are equipped 16-byte transmitter FIFO (TX_FIFO) and 16-byte receiver FIFO (RX_FIFO). The CPU can read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as 4 error conditions (parity error, overrun error, framing error and break interrupt) probably occur while receiving data. The UART includes a programmable baud rate generator that is capable of dividing crystal clock input by divisors to produce the clock that transmitter and receiver need. The baud rate equation is $\text{Baud Rate} = \text{UART\_CLK} / M * [\text{BRD} + 2]$, where M and BRD are defined in **Error! Reference source not found..** Following table lists the equations in the various conditions.

The UART0 controller support auto-flow control function that uses two low-level signals, /CTS (clear-to-send) and /RTS (request-to-send), to control the flow of data transfer between the UART and external devices (ex: Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts /RTS to external device. When the number of bytes in the Rx FIFO equals the value of UA_FCR.RTS_Tri_Lev [19:16], the /RTS is de-asserted. The UART sends data out when UART controller detects /CTS is asserted from external device. If a validly asserted /CTS is not detected the UART controller will not send data out.

DIV_X_EN	DIV_X_ONE	Divider X	BRD	Baud Rate Equation
Disable	0	B	A	$\text{UART\_CLK} / [16 * (A+2)]$
Enable	0	B	A	$\text{UART\_CLK} / [(B+1) * (A+2)]$, B must ≥ 8
Enable	1	B	A	$\text{UART\_CLK} / (A+2)$, A must ≥ 3

5.21.2 Features

64 byte/16 byte entry FIFOs for received and transmitted data payloads.

Auto flow control/flow control function (CTS, RTS) are supported (Normal speed UART not support).

Programmable baud-rate generator.

Fully programmable serial-interface characteristics:

- 5-, 6-, 7-, or 8-bit character
- Even, odd, or no-parity bit generation and detection
- 1-, 1&1/2, or 2-stop bit generation
- Baud rate generation

- False start bit detection.
- Loop back mode for internal diagnostic testing

5.21.3 Block Diagram

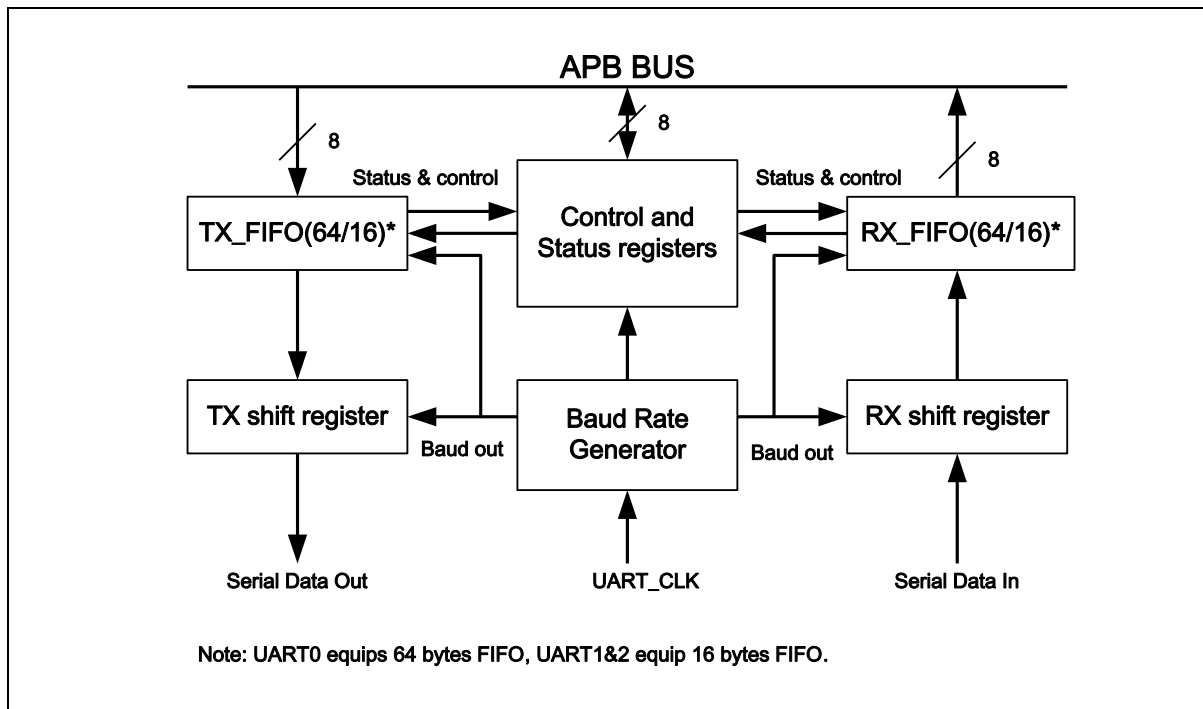


Figure 5.21-1 UART Block Diagram

5.21.4 Functional Blocks Descriptions

TX_FIFO

The transmitter is buffered with a 64/16 byte FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO

The receiver is buffered with a 64/16 byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU.

TX shift Register

Shifting the transmitting data out serially

RX shift Register

Shifting the receiving data in serially

Modem Control Register

This register controls the interface to the MODEM or data set (or a peripheral device emulating a MODEM).

Modem Status Register

This register provides the current status of the control lines from the MODEM and cause the MODEM

status interrupt (CTS# or DSR# or RI# or DCD#)

Note: Only CTS#/RTS# can be used in this version, and normal speed not support.

Baud Rate Generator

Dividing the external clock by the divider to get the desired internal clock

Control and Status Register

This is a register set, including the FIFO control registers (FCR), FIFO status registers (FSR), and line control register (LCR) for transmitter and receiver. The line status register (LSR) provides information to the CPU concerning the data transfer. The time out control register (TOR) identifies the condition of time out interrupt. This register set also includes the interrupt enable register (IER) and interrupt identification register (IIR) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are four types of interrupts: line status interrupt (overrun error or parity error or framing error or break interrupt), transmitter holding register empty interrupt, receiver threshold level reaching, and time out interrupt.

The following diagram demonstrates the auto-flow control block diagram.

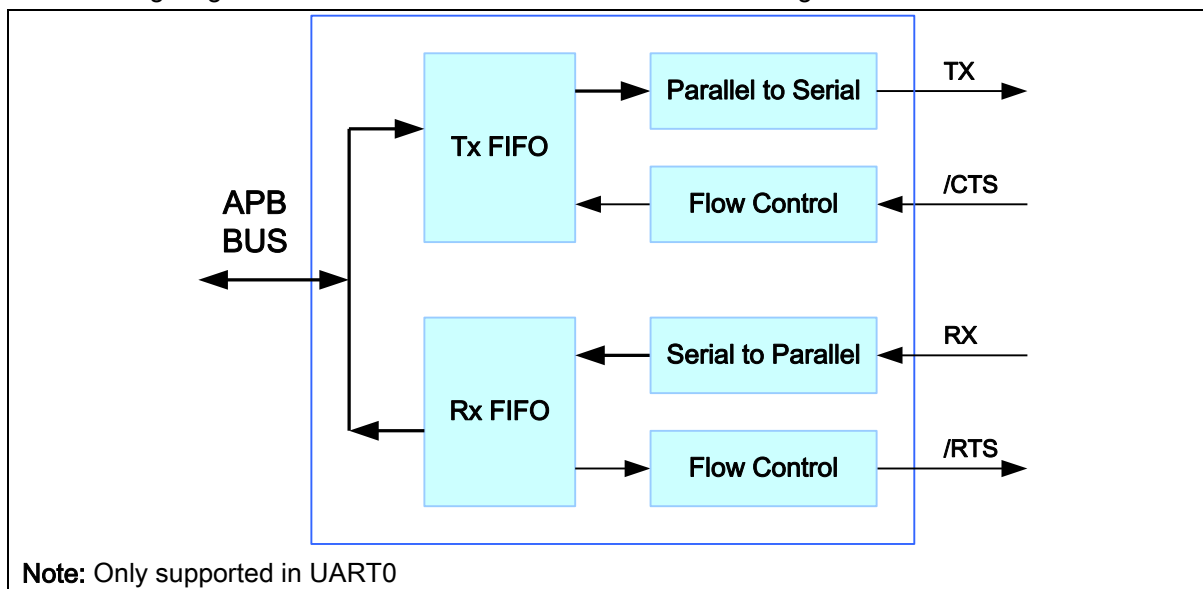
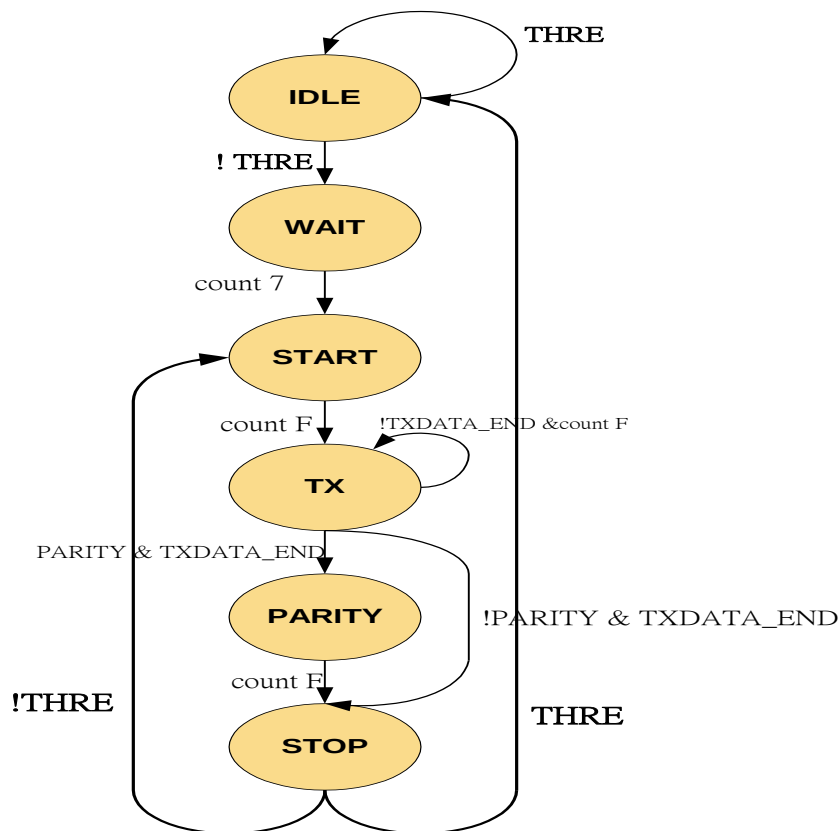


Figure 5.21-2 Auto Flow Control Block Diagram

5.21.5 Finite State Machine

5.21.5.1 Transmitter



5.21.5.2 State Definition

- IDLE
The transmitter has no data to transmit.
- WAIT
The transmitter's FIFO is not empty.
- START
The transmitter transmits the start bit.
- TX
The transmitter transmits the data.
- PARITY
The transmitter transmits the parity bit.
- STOP
The transmitter transmits the stop bit.

Signal Description

-
- THRE

The transmitter holding register is empty.

- Count7

The counter of clock equals to 7.

- CountF

The counter of clock equals to 15.

- TXDATA_END

The data part transfer is finished.

- PARITY

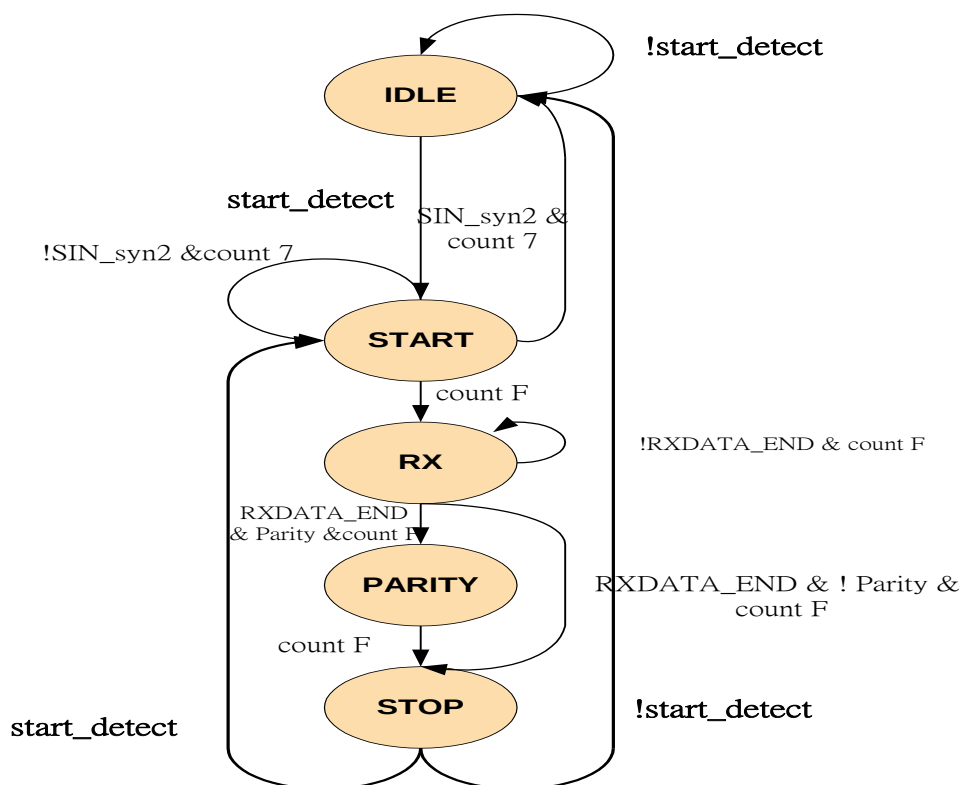
The transfer includes the parity bit.

NOTE:

The format of the transfer is as following:

One transfer = Start + Data + Parity bit (if dedicated) + Stop bit

5.21.5.3 Receiver



State Definition

- IDLE

The receiver has no data to receive.

- **START**

The receiver receives the start bit.

- **RX**

The receiver receives the desired data.

- **PARITY**

The receiver receives the parity bit.

- **STOP**

The receiver receives the parity bit.

Signal Description

- **Start_detect**

To detect the start of the transfer

- **SIN_syn2**

The synchronized input data

- **Count7**

The counter of clock equals to 7.

- **CountF**

The counter of clock equals to F.

- **RXDATA_END**

The data received finished

PARITY

Receiving the parity bit if needed

5.21.6 Registers Map

R: read only, W: write only, R/W: both read and write, C: Only value 0 can be written

First set of the UART Interface register Map

Channel0: UART_Base0 (High Speed) = 0xB800_8000

Channel1: UART_Base1 (Normal Speed) = 0xB800_8000

Register	Address	R/W	Description	Reset Value
UART Base address Channel0 : UA_BA (High Speed) = 0xB800_8000 Channel1 : UA_BA (Normal Speed) = 0xB800_8100				
UA_RBR	UA_BA + 0x00	R	Receive Buffer Register.	Undefined
UA_THR	UA_BA + 0x00	W	Transmit Holding Register.	Undefined
UA_IER	UA_BA + 0x04	R/W	Interrupt Enable Register.	0x0000_0000
UA_FCR	UA_BA + 0x08	R/W	FIFO Control Register.	0x0000_0001
UA_LCR	UA_BA + 0x0C	R/W	Line Control Register.	0x0000_0000
UA_MCR	UA_BA + 0x10	R/W	Modem Control Register.	0x0000_2000
UA_MSR	UA_BA + 0x14	R/W	Modem Status Register.	0x0000_00XX
UA_FSR	UA_BA + 0x18	R/W	FIFO Status Register.	0x1040_4000
UA_ISR	UA_BA + 0x1C	R/W	Interrupt Status Register.	0x0000_80XX
UA_TOR	UA_BA + 0x20	R/W	Time Out Register	0x0000_0000
UA_BAUD	UA_BA + 0x24	R/W	Baud Rate Divider Register	0x0F00_0000

5.21.7 Register Description

Receive Buffer Register (UA_RBR)

Register	Address	R/W	Description	Reset Value
UA_RBR	UA_BA + 0x00	R	Receive Buffer Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
8-bit Received Data							

Bits	Descriptions	
[7:0]	8-bit Received Data	Receive Buffer Register By reading this register, the UART will return an 8-bit data received from SIN pin (LSB first).

Transmit Holding Register (UA_THR)

Register	Address	R/W	Description	Reset Value
UA_THR	UA_BA + 0x00	W	Transmit Holding Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
8-bit Transmitted Data							

Bits	Descriptions	
[7:0]	8-bit Transmitted Data	Transmit Holding Register By writing to this register, the UART will send out an 8-bit data through the SOUT pin (LSB first).

Interrupt Enable Register (UA_IER)

Register	Address	R/W	Description	Reset Value
UA_IER	UA_BA + 0x04	R/W	Interrupt Enable Register.	0x0000_0000

31	30	29	28	27	26	25	24
nDBGACK_EN	Reserved						
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DMA_Rx	DMA_Tx	Auto_CTS	Auto_RTS	Time_out	Reserved		
7	6	5	4	3	2	1	0
Reserved	Wake_IEN	BUF_ERR	RTO_IEN	MS_IEN	RLS_IEN	THRE_IEN	RDA_IEN

Bits	Descriptions	
[31]	nDBGACK_EN	ICE debug mode acknowledge enable 0 = When DBGACK is high, the UART receiver time-out clock will be held 1 = No matter what DBGACK is high or not, the UART receiver timer-out clock will not be held.
[30:16]	Reserved	Reserved
[15]	DMA_Rx_EN	Rx DMA Enable 0 = Enable Rx DMA. 1 = Disable Rx DMA.
[14]	DMA_Tx_EN	Tx DMA Enable 0 = Enable Tx DMA. 1 = Disable Tx DMA.
[13]	Auto_CTS_EN	CTS Auto Flow Control Enable (not available in UART1 channel) 1 = Enable CTS auto flow control. 0 = Disable CTS auto flow control. When CTS auto-flow is enabled, the UART will send data to external device when CTS input assert (UART will not send data to device until CTS is asserted).
[12]	Auto_RTS_EN	RTS Auto Flow Control Enable (not available in UART1 channel) 1 = Enable RTS auto flow control. 0 = Disable RTS auto flow control. When RTS auto-flow is enabled, if the number of bytes in the Rx FIFO equals the UA_FCR [RTS_Tri_Lev], the UART will assert RTS signal.
[11]	Time_out_EN	Time Output Counter Enable 1 = Enable Time output counter. 0 = Disable Time output counter.
[10:7]	Reserved	Reserved

[6]	Wake_IEN	Wake up interrupt enable for INTR[wakeup] 0 = Mask off INTR_Wakeup 1 = Enable INTR_Wakeup function, when the system is in deep sleep mode, an external /CTS change will wake up CPU from deep sleep mode.
[5]	BUF_ERR_IEN	Buffer Error interrupt enable 0 = Mask off INTR_Buf_err 1 = Enable INTR_Buf_err
[4]	RTO_IEN	Rx Time out Interrupt Enable 0 = Mask off INTR_tout 1 = Enable INTR_tout
[3]	MS_IEN	MODEM Status Interrupt (INTR_MOS) Enable 0 = Mask off INTR_MOS 1 = Enable INTR_MOS
[2]	RLS_IEN	Receive Line Status Interrupt (INTR_RLS) Enable 0 = Mask off INTR_RLS 1 = Enable INTR_RLS
[1]	THRE_IEN	Transmit Holding Register Empty Interrupt (INTR_THRE) Enable 0 = Mask off INTR_THRE 1 = Enable INTR_THRE
[0]	RDA_IEN	Receive Data Available Interrupt (INTR_RDA) Enable. 0 = Mask off INTR_RDA 1 = Enable INTR_RDA

FIFO Control Register (UA_FCR)

Register	Address	R/W	Description	Reset Value
UA_FCR	UA_BA + 0x08	R/W	FIFO Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				RTS_Tri_lev			
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
RFITL				Reserved	TFR	RFR	Reserved

Bits	Descriptions																			
[31:20]	Reserved	Reserved																		
[19:16]	RTS_tri_lev	RTS Trigger Level (not available in UART1 channel)																		
		<table><tr><th>RTS_tri_lev</th><th>Trigger Level (Bytes)</th></tr><tr><td>0000</td><td>01</td></tr><tr><td>0001</td><td>04</td></tr><tr><td>0010</td><td>08</td></tr><tr><td>0011</td><td>14</td></tr><tr><td>0100</td><td>30/14 (High Speed/Normal Speed)</td></tr><tr><td>0101</td><td>46/14 (High Speed/Normal Speed)</td></tr><tr><td>0110</td><td>62/14 (High Speed/Normal Speed)</td></tr><tr><td>others</td><td>62/14 (High Speed/Normal Speed)</td></tr></table>	RTS_tri_lev	Trigger Level (Bytes)	0000	01	0001	04	0010	08	0011	14	0100	30/14 (High Speed/Normal Speed)	0101	46/14 (High Speed/Normal Speed)	0110	62/14 (High Speed/Normal Speed)	others	62/14 (High Speed/Normal Speed)
		RTS_tri_lev	Trigger Level (Bytes)																	
		0000	01																	
		0001	04																	
		0010	08																	
		0011	14																	
		0100	30/14 (High Speed/Normal Speed)																	
		0101	46/14 (High Speed/Normal Speed)																	
		0110	62/14 (High Speed/Normal Speed)																	
		others	62/14 (High Speed/Normal Speed)																	
Note: This bit is use for auto RTS flow control.																				

[7:4]	RFITL	RX FIFO Interrupt (INTR_RDA) Trigger Level When the number of bytes in the receive FIFO equals the RFITL then the RDA_IF will be set (if IER [RDA_IEN] is enable, an interrupt will generated).	
		RFITL	INTR_RDA Trigger Level (Bytes)
		0000	01
		0001	04
		0010	08
		0011	14
		0100	30/14 (High Speed/Normal Speed)
		0101	46/14 (High Speed/Normal Speed)
		0110	62/14 (High Speed/Normal Speed)
		others	62/14 (High Speed/Normal Speed)
[3]	Reserved	Reserved	
[2]	Tx_RST	Tx Software Reset When Tx_RST is set, all the bytes in the transmit FIFO and Tx internal state machine are cleared. 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the Tx internal state machine and pointers. Note: This bit will auto clear after few clock cycles.	
[1]	Rx_RST	Rx Software Reset When Rx_RST is set, all the bytes in the transmit FIFO and Rx internal state machine are cleared. 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the Rx internal state machine and pointers. Note: This bit will auto clear after few clock cycles.	
[0]	Reserved	Reserved	

Line Control Register (UA_LCR)

Register	Address	R/W	Description	Reset Value
UA_LCR	UA_BA + 0x0C	R/W	Line Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	BCB	SPE	EPE	PBE	NSB	WLS	

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6]	BCB	Break Control Bit When this bit is set to logic 1, the serial data output (SOUT) is forced to the Spacing State (logic 0). This bit acts only on SOUT and has no effect on the transmitter logic.
[5]	SPE	Stick Parity Enable 0 = Disable stick parity 1 = Parity bit is transmitted and checked as a logic 1 if bit 4 is 0 (odd parity), or as a logic 0 if bit 4 is 1 (even parity). This bit has effect only when bit 3 (parity bit enable) is set.
[4]	EPE	Even Parity Enable 0 = Odd number of logic 1's are transmitted or checked in the data word and parity bits. 1 = Even number of logic 1's are transmitted or checked in the data word and parity bits. This bit has effect only when bit 3 (parity bit enable) is set.
[3]	PBE	Parity Bit Enable 0 = Parity bit is not generated (transmit data) or checked (receive data) during transfer. 1 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data.
[2]	NSB	Number of "STOP bit" 0= One " STOP bit" is generated in the transmitted data 1= One and a half " STOP bit" is generated in the transmitted data when 5-bit word length is selected; Two "STOP bit" is generated when 6-, 7- and 8-bit word length is selected.

[1:0]	WLS	Word Length Select	
		WLS[1:0]	Character length
		00	5 bits
		01	6 bits
		10	7 bits
		11	8 bits

MODEM Control Register (UA_MCR)

Register	Address	R/W	Description	Reset Value
UA_MCR	UA_BA + 0x10	R/W	MODEM Control Register	0x0000_2000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		RTS_st	Reserved			Lev_RTS	Reserved
7	6	5	4	3	2	1	0
Reserved			LBME	Reserved		RTS#	Reserved

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13]	RTS_st	RTS Pin State (not available in UART1 channel) This bit is the pin status of RTS.
[12:10]	Reserved	Reserved
[9]	Lev_RTS	RTS Trigger Level (not available in UART1 channel) 0: low level triggered 1: high level triggered
[8:5]	Reserved	Reserved
[4]	LBME	Loop-back Mode Enable 0 = Disable 1 = When the loop-back mode is enable, the following signals are connected internally: SOUT connected to SIN and SOUT pin fixed at logic 1 RTS# connected to CTS# and RTS# pin fixed at logic 1
[3:2]	Reserved	Reserved
[1]	RTS#	RTS (Request-To-Send) signal (not available in UART1 channel) 0: Drive RTS pin to logic 1 (If the Lev_RTS set to low level triggered). 1: Drive RTS pin to logic 0 (If the Lev_RTS set to low level triggered). 0: Drive RTS pin to logic 0 (If the Lev_RTS set to high level triggered). 1: Drive RTS pin to logic 1 (If the Lev_RTS set to high level triggered).

Modem Status Register (UA MSR)

Register	Address	R/W	Description	Reset Value
UA_MSR	UA_BA + 0x14	R/W	Modem Status Register	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							Lev_CTS
7	6	5	4	3	2	1	0
Reserved			CTS_st	Reserved			DCTS

Bits	Descriptions	
[31:9]	Reserved	Reserved
[8]	Lev_CTS	CTS Trigger Level (not available in UART1 channel) This bit can change the CTS trigger level. 0: low level triggered 1: high level triggered
[7:5]	Reserved	Reserved
[4]	CTS_st	CTS Pin Status (not available in UART1 channel) This bit is the pin status of CTS.
[3:1]	Reserved	Reserved
[0]	DCTS	Detect CTS State Change Flag (not available in UART1 channel) This bit is set whenever CTS input has change state, and it will generate interrupt to cup (Irpt_Modem). NOTE: This bit is read only, but can be cleared by writing '1' to it.

FIFO Status Register (UA_FSR)

Register	Address	R/W	Description	Reset Value
UA_FSR	UA_BA + 0x18	R	FIFO Status Register	0x1040_4000

31	30	29	28	27	26	25	24
Reserved			TE_Flag	Reserved			Tx_Over_IF
23	22	21	20	19	18	17	16
Tx_Full	Tx_Empty	Tx_Pointer					
15	14	13	12	11	10	9	8
Rx_Full	Rx_Empty	Rx_Pointer					
7	6	5	4	3	2	1	0
Reserved	BII	FEI	PEI	Reserved			Rx_Over_IF

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28]	TE_Flag	Transmitter Empty Flag (Read Only) 0 = Bit is cleared automatically when Tx FIFO is not empty or the last byte transmission has not completed. 1 = Bit is set by hardware when Tx FIFO(UA_THR) is empty and the STOP bit of the last byte has been transmitted. NOTE: This bit is read only.
[27:25]	Reserved	Reserved
[24]	Tx_Over_IF	Tx Overflow Error IF (Read Only) If Tx FIFO(UA_THR) is full, an additional write to UA_THR will cause this bit to logic 1. 1 = None. 0 = an overflow error will occur. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[23]	Tx_Full	Transmitter FIFO Full (Read Only) This bit indicates Tx FIFO full or not. This bit is set when Tx_Point is equal to 64/16(UART0/UART1), otherwise is cleared by hardware. 0 = None. 1 = Tx FIFO are full
[22]	Tx_Empty	Transmitter FIFO Empty (Read Only) This bit indicates Tx FIFO empty or not. When the last byte of Tx FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into THR (Tx FIFO not empty). 0 = None. 1 = Tx FIFO are empty.

[21:16]	Tx_Pointer	TX FIFO Pointer (Read Only) This field indicates the Tx FIFO Buffer Pointer. When CPU writes one byte into UA_THR, Tx_Pointer increases one. When one byte of Tx FIFO is transferred to Transmitter Shift Register, Tx_Pointer decreases one.
[15]	Rx_Full	Receiver FIFO Full (Read Only) This bit initiates Rx FIFO full or not. This bit is set when Rx_Point is equal to 64/16(UART0/UART1), otherwise is cleared by hardware. 0 = None. 1 = Rx FIFO full
[14]	Rx_Empty	Receiver FIFO Empty (Read Only) This bit initiate Rx FIFO empty or not. When the last byte of Rx FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data. 0 = None. 1 = Receiver FIFO are empty.
[13:8]	Rx_Pointer	Rx FIFO pointer (Read Only) This field indicates the Rx FIFO Buffer Pointer. When UART receives one byte from external device, Rx_Pointer increases one. When one byte of Rx FIFO is read by CPU, Rx_Pointer decreases one.
[7]	Reserved	Reserved
[6]	BII	Break Interrupt Indicator This bit is set to a logic 1 whenever the received data input is held in the "spacing state" (logic 0) for longer than a full word transmission time (that is, the total time of "start bit" + data bits + parity + stop bits) and is reset whenever the CPU write 1 to the contents of the UA_FSR[BII] or UA_FSR[FEI] or UA_FSR[PEI].
[5]	FEI	Framing Error Indicator This bit is set to logic 1 whenever the received character does not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as a logic 0), and is reset whenever the CPU write 1 to the contents of the UA_FSR[BII] or UA_FSR[FEI] or UA_FSR[PEI].
[4]	PEI	Parity Error Indicator This bit is set to logic 1 whenever the received character does not have a valid "parity bit", and is reset whenever the CPU write 1 to the contents of the UA_FSR[BII] or UA_FSR[FEI] or UA_FSR[PEI].
[3:1]	Reserved	Reserved
[0]	Rx_over_IF	Rx overflow Error IF (Read Only) This bit is set when Rx FIFO overflow. If the number of bytes of received data is greater than Rx FIFO(UA_RBR) size, 64/16 bytes of UART0/UART1, this bit will be set. 1 = None. 0 = an overflow error will occur. NOTE: This bit is read only, but can be cleared by writing '1' to it.

Interrupt Status Control Register (UA_ISR)

Register	Address	R/W	Description	Reset Value
UA_ISR	UA_BA + 0x1C	R/W	Interrupt Status Register.	0x0000_80XX

31	30	29	28	27	26	25	24
DMA_Rx	HW_	HW_Buf_	HW_	HW_	HW_	Reserved	
23	22	21	20	19	18	17	16
DMA_Tx	HW_	HW_Buf_	HW_	HW_	HW_	Reserved	
15	14	13	12	11	10	9	8
Soft_Rx	Wake_INT	Buf_Err	Tout	Modem	RLS	THRE	RDA
7	6	5	4	3	2	1	0
Soft_Tx	Wake_IF	Buf_Err	Tout	Modem	RLS	THRE	RDA

Bits	Descriptions	
[31]	DMA_Rx_Flg	DMA RX Mode Flag (Read Only) 0 = The UART is not work in DMA Rx mode 1 = The UART is work in DMA Rx mode
[30]	HW_Wake_INT	Hardware Wake Up Interrupt Pin Status 0 = None. 1 = Wake up interrupt occur when in DMA mode.
[29]	HW_Buf_Err_INT	Hardware Buffer Error Interrupt Pin Status (INTR_Buf_Err) An AND output with inputs of BUF_ERR_IEN and Buf_Err_IF 0 = None. 1 = Buffer Error interrupt occur when in DMA mode.
[28]	HW_Tout_INT	Hardware Time Out Interrupt Pin Status (INTR_Tout) An AND output with inputs of RTO_IEN and Tout_IF 0 = None. 1 = Time out interrupt occur when in DMA mode.
[27]	HW_Modem_INT	Hardware MODEM Status Interrupt Pin Status (INTR_MOS) (not available in UART1 channel) An AND output with inputs of Modem_IEN and Modem_IF. 0 = None. 1 = Modem status interrupt occur when in DMA mode.
[26]	HW_RLS_INT	Hardware Receive Line Status Interrupt Pin Status (INTR_RLS). An AND output with inputs of RLS_IEN and RLS_IF. 0 = None. 1 = Receive line status interrupt occur when in DMA mode.
[25:24]	Reserved	Reserved

[23]	DMA_Tx _Flag	Hardware DMA Tx Mode Flag (Read Only) 0 = The UART is not work in DMA TX mode 1 = The UART is work in DMA TX mode
[22]	HW_ Wake_IF	Hardware Wake Up Flag (Read Only) 0 = None. 1 = Wake up flag occur when in DMA mode.
[21]	HW_Buf_ Err_IF	Hardware Buffer Error Flag (Read Only) This bit is set when the Tx or Rx FIFO overflows (Tx_Over_IF or Rx_Over_IF is set). When Buf_Err_IF is set, the transfer maybe is not correct. If IER[Buf_Err_IEN] is enabled, the buffer error interrupt will be generated. 0 = None. 1 = Buffer Error flag occur when in DMA mode. NOTE: This bit is cleared when both Tx_Over_IF and Rx_Over_IF are cleared.
[20]	HW_ Tout_INT	Hardware Time out Flag (Read Only) This bit is set when the Rx FIFO is not empty and no activities occurs in the Rx FIFO and the time out counter equal to TOIC. If IER[Tout_IEN] is enabled, the Tout interrupt will be generated. 0 = None. 1 = Time out status flag occur when in DMA mode. NOTE: This bit is read only and user can read UA_RBR (Rx is in active) to clear it.
[19]	HW_ Modem_IF	Hardware MODEM Status Flag (Read Only) (not available in UART1 channel) This bit is set when the CTS pin has state change(DCTSIF=1). if IER[Modem_IEN] is enabled, the Modem interrupt will be generated. 0 = None. 1 = Modem status flag occur when in DMA mode. NOTE: This bit is read only and reset to 0 when bit DCTSIF is cleared by a write 1 on DCTSIF.
[18]	HW_ RLS_IF	Hardware Receive Line Status Flag. (Read Only) This bit is set when the Rx receive data have parity error, framing error or break error (at least one of 3 bits, BIF, FEF and PEF, is set). If IER[RLS_IEN] is enabled, the RLS interrupt will be generated. 0 = None. 1 = Receive line status flag occur when in DMA mode. NOTE: This bit is read only and reset to 0 when all bits of BIF, FEF and PEF are cleared.
[17:16]	Reserved	Reserved
[15]	Soft_Rx _Flag	Soft RX Mode Flag (Read Only) 0 = The UART is not work in Software Rx mode 1 = The UART is work in Software Rx mode
[14]	Wake_INT	Wake Up Interrupt Pin Status 0 = None. 1 = Wake up interrupt occur when in Software mode.
[13]	Buf_Err _INT	Buffer Error Interrupt Pin Status (INTR_Buf_Err) An AND output with inputs of BUF_ERR_IEN and Buf_Err_IF 0 = None. 1 = Buffer Error interrupt occur when in Software mode.

[12]	Tout _INT	Time Out Interrupt Pin Status (INTR_Tout) An AND output with inputs of RTO_IEN and Tout_IF 0 = None. 1 = Time out interrupt occur when in Software mode.
[11]	Modem _INT	MODEM Status Interrupt Pin Status (INTR_MOS). (not available in UART1 channel) An AND output with inputs of Modem_IEN and Modem_IF 0 = None. 1 = Modem status interrupt occur when in Software mode.
[10]	RLS _INT	Receive Line Status Interrupt Pin Status (INTR_RLS). An AND output with inputs of RLS_IEN and RLS_IF 0 = None. 1 = Receive line status interrupt occur when in Software mode.
[9]	THRE _INT	Transmit Holding Register Empty Interrupt Pin Status (INTR_THRE). An AND output with inputs of THRE_IEN and THRE_IF 0 = None. 1 = Transmit holding register empty interrupt occur when in Software mode.
[8]	RDA _INT	Receive Data Available Interrupt Pin Status (INTR_RDA). An AND output with inputs of RDA_IEN and RDA_IF 0 = None. 1 = receive data available interrupt occur when in Software mode.
[7]	Soft_Tx _Flag	Software Tx Mode Flag (Read Only) 0 = The UART is not work in Software Tx mode. 1 = The UART is work in Software Tx mode.
[6]	Wake_IF	Wake Up Flag (Read Only) 0 = None. 1 = Wake up flag occur when in Software mode.
[5]	Buf_Err _IF	Buffer Error Flag (Read Only) This bit is set when the Tx or Rx FIFO overflows (Tx_Over_IF or Rx_Over_IF is set). When Buf_Err_IF is set, the transfer maybe is not correct. If IER[Buf_Err_IEN] is enabled, the buffer error interrupt will be generated. 0 = None. 1 = Buffer Error flag occur when in Software mode. NOTE: This bit is cleared when both Tx_Over_IF and Rx_Over_IF are cleared.
[4]	Tout _IF	Time Out Interrupt Flag (Read Only) This bit is set when the Rx FIFO is not empty and no activities occurs in the Rx FIFO and the time out counter equal to TOIC. If IER[Tout_IEN] is enabled, the Tout interrupt will be generated. 0 = None. 1 = Time out flag occur when in Software mode. NOTE: This bit is read only and user can read UA_RBR (Rx is in active) to clear it.

[3]	Modem _IF	MODEM Status Flag (Read Only) (not available in UART1 channel) This bit is set when the CTS pin has state change(DCTSIF=1). If IER[Modem_IEN] is enabled, the Modem interrupt will be generated. 0 = None. 1 = Modem status flag occur when in Software mode. NOTE: This bit is read only and reset to 0 when bit DCTSIF is cleared by a write 1 on DCTSIF.
[2]	RLS _IF	Receive Line Status Flag. (Read Only) This bit is set when the Rx receive data have parity error, framing error or break error (at least one of 3 bits, BIF, FEF and PEF, is set). If IER[RLS_IEN] is enabled, the RLS interrupt will be generated. 0 = None. 1 = Receive line status flag occur when in Software mode. NOTE: This bit is read only and reset to 0 when all bits of BIF, FEF and PEF are cleared.
[1]	THRE _IF	Transmit Holding Register Empty Flag. (Read Only) This bit is set when the last data of Tx FIFO is transferred to Transmitter Shift Register. If IER[THRE_IEN] is enabled, the THRE interrupt will be generated. 0 = None. 1 = Transmit holding register empty flag occur when in Software mode. NOTE: This bit is read only and it will be cleared when writing data into THR (Tx FIFO not empty).
[0]	RDA _IF	Receive Data Available Flag. (Read Only) When the number of bytes in the Rx FIFO equals the RFITL then the RDA_IF will be set. If IER[RDA_IEN] is enabled, the RDA interrupt will be generated. 0 = None. 1 = receive data available flag occur when in Software mode. NOTE: This bit is read only and it will be cleared when the number of unread bytes of Rx FIFO drops below the threshold level (RFITL).

UART Interrupt Sources and Flags Table In DMA Mode

UART Interrupt Source	Interrupt Enable Bit	Interrupt Indicator To Interrupt Controller	Interrupt Flag	Flag Clear By
LIN RX Break Field Detected interrupt	LIN_RX_BRK_IEN	HW_LIN_Rx_Break_INT	HW_LIN_Rx_Break_IF	Write '1' to LIN_Rx_Break_IF
Buffer Error Interrupt INT_Buf_Err	BUF_ERR_IEN	HW_Buf_Err_INT	HW_Buf_Err_IF = (Tx_Over_IF or Rx_Over_IF)	Write '1' to Tx_Over_IF/ Rx_Over_IF
Rx Timeout Interrupt INT_Tout	RTO_IEN	HW_Tout_INT	HW_Tout_IF	Read UA_RBR
Modem Status Interrupt INT_Modem	Modem_IEN	HW_Modem_INT	HW_Modem_IF = (DCTSIF)	Write '1' to DCTSIF
Receive Line Status Interrupt INT_RLS	RLS_IEN	HW_RLS_INT	HW_RLS_IF = (BIF or FEF or PEF)	Write '1' to BIF/FEF/PEF
Transmit Holding Register Empty Interrupt INT_THRE	THRE_IEN	HW_THRE_INT	HW_THRE_IF	Write UA_THR

Receive Data Available Interrupt INT_RDA	RDA_IEN	HW_RDA_INT	HW_RDA_IF	Read UA_RBR

Time Out Register (UA TOR)

Register	Address	R/W	Description	Reset Value
UA_TOR	UA_BA + 0x20	R/W	Time Out Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	TOIC						

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6:0]	TOIC	Time Out Interrupt Comparator The time out counter resets and starts counting (the counting clock = baud rate) whenever the RX FIFO receives a new data word. Once the content of time out counter (TOUT_CNT) is equal to that of time out interrupt comparator (TOIC), a receiver time out interrupt (INTR_TOUT) is generated if UA_IER[RTO_IEN]. A new incoming data word or RX FIFO empty clears INTR_TOUT.

Baud Rate Divider Register

Register	Address	R/W	Description	Reset Value
UA_BAUD	UA_BA + 0x24	R/W	Baud Rate Divider Register	0x0F00_0000

31	30	29	28	27	26	25	24
Reserved		DIV_X_EN	DIV_X_ONE	Divider X			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Baud Rate Divider DLM (High Byte)							
7	6	5	4	3	2	1	0
Baud Rate Divider DLL (Low Byte)							

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29]	DIV_X_EN	Divider X Enable The BRD = Baud Rate Divider, and the baud rate equation is $\text{Baud Rate} = \text{Clock} / [M * (\text{BRD} + 2)]$; The default value of M is 16. 0 = Disable divider X (the equation of M = 16) 1 = Enable divider X (the equation of M = X+1, but Divider_X[27:24 must > 8).
[28]	DIV_X_ONE	Divider X equal 1 0 = Divider M = X (the equation of M = X+1, but Divider_X[27:24] must > 8) 1 = Divider M = 1(the equation of M = 1, but BRD[15:0] must > 3).
[27:24]	Divider X	Divider X The baud rate divider M = X+1.
[23:16]	Reserved	Reserved
[15:8]	Baud Rate Divider (High Byte)	Baud Rate Divider DLM The high byte of the baud rate divider
[7:0]	Baud Rate Divider (Low Byte)	Baud Rate Divider DLL The low byte of the baud rate divider

Note : The Divider = DLL + DLM, and the baud rate equation is $\text{Baud Rate} = \text{Clock} / M * [\text{Divider} + 2]$
 (The default value of M is 16).

Example:

DIV_X_EN	DIV_X_ONE	Divider X	DLL + DLM	Baud rate equation
0	0	B	A	$\text{Clock} / 16 \cdot (A+2)$
1	0	B	A	$\text{Clock} / B \cdot (A+2)$, B must > 8
1	1	B	A	$\text{Clock} / (A+2)$, A must > 3

5.22 SPI0/1 Serial Interface Controller (Master/Slave)

5.22.1 Overview

The MICROWIRE/SPI Synchronous Serial Interface performs a serial-to-parallel conversion on data characters received from the peripheral, and a parallel-to-serial conversion on data characters received from CPU. This interface can drive up to 2 external peripherals and is seen as the master or can be driven as the slave. It can generate an interrupt signal when data transfer is finished and can be cleared by writing 1 to the interrupt flag. The active level of device/slave select signal can be chosen to low active or high active, which depends on the peripheral it's connected. Writing a divisor into DIVIDER register can program the frequency of serial clock output when it is as the master. This master/slave core contains four 32-bit transmit/receive buffers, and can provide burst mode operation. The maximum bits can be transmitted/received is 32 bits, and can transmit/receive data up to four times successively.

There is EDMA mode for transmit or received data access by enable the EDMA bit in SPI_EDMA[0]

5.22.2 Feature

- AMBA APB interface compatible
- Support MICROWIRE/SPI master/slave mode
- Full duplex synchronous serial data transfer
- Variable length of transfer word up to 32 bits
- Provide burst mode operation, transmit/receive can be executed up to four times in one transfer
- MSB or LSB first data transfer
- Rx and Tx on both rising or falling edge of serial clock independently
- 2 slave/device select lines when it is as the master, and 1 slave/device select line when it is as the slave
- BYTE SLEEP
- Only Support the external master device that the frequency of its serial clock output is less 1/5 than the MICROWIRE/SPI Core clock input (pclk) and its slave select output is edge-active trigger.
- EDMA mode

5.22.3 SPI (MICROWIRE) Block Diagram (Master/Slave)

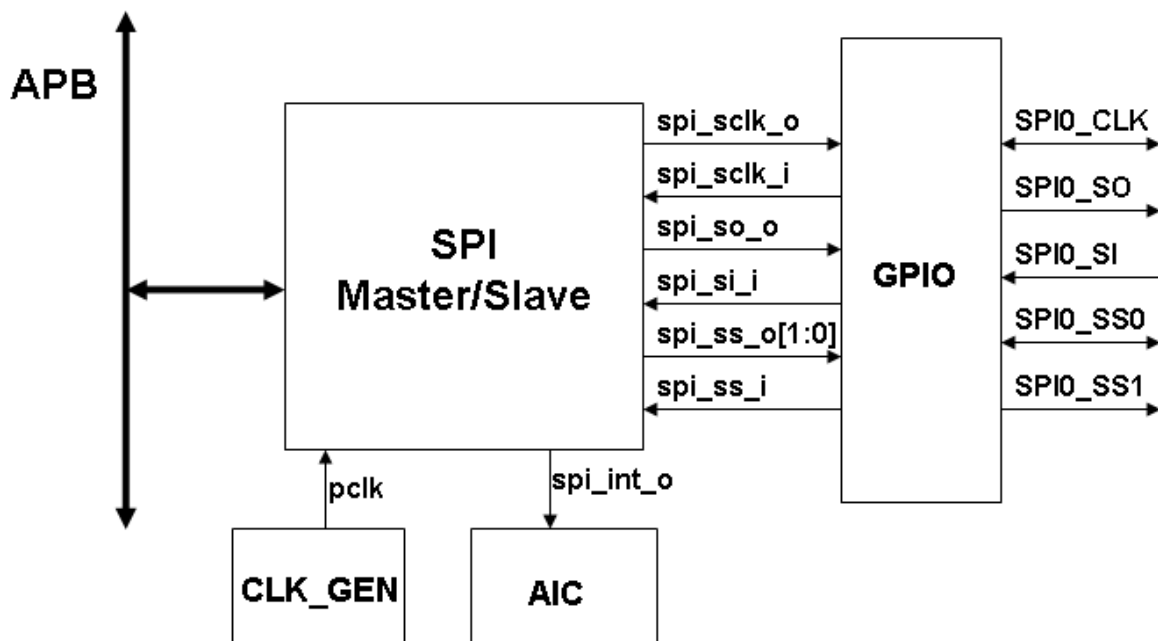


Figure 5.22-1 MICROWIRE/SPI Block Diagram (Master/Slave)

Note. SPI1 is same as SPI0

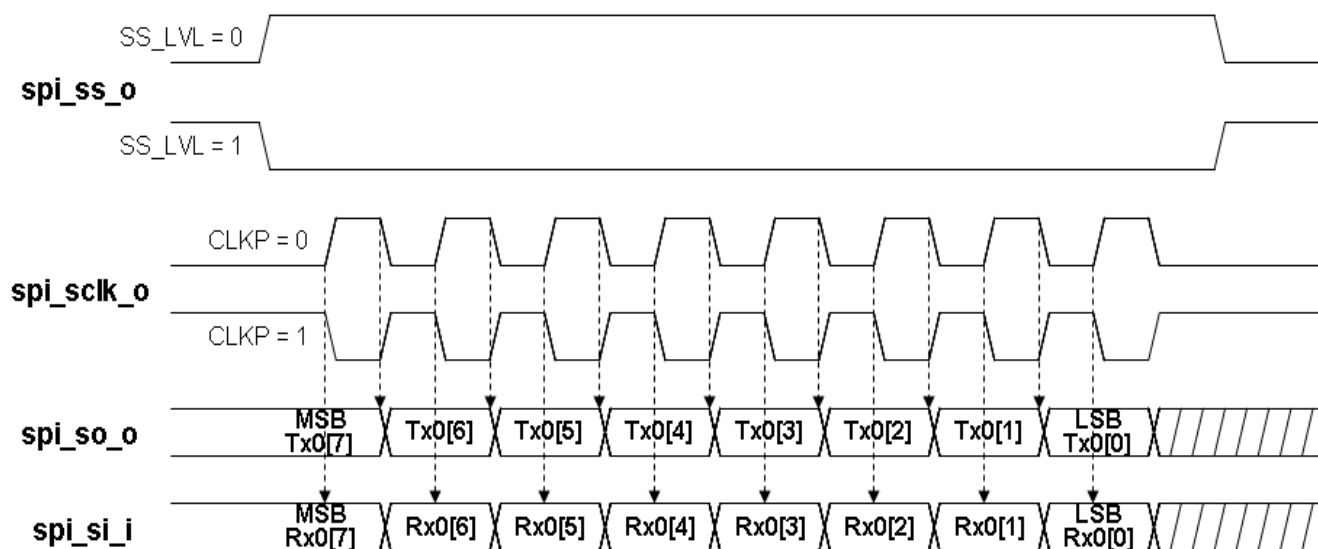
Internal pin descriptions:

- spi_sclk_o: MICROWIRE/SPI master serial clock output.
- spi_int_o: MICROWIRE/SPI interrupts signal output.
- spi_ss_o[1:0]: MICROWIRE/SPI two slave/device select signals output.
- spi_so_o: MICROWIRE/SPI serial data output to slave device in master mode or to master device in slave mode.
- spi_si_i: MICROWIRE/SPI serial data input from slave device in master mode or from master device in slave mode.
- spi_sclk_i: MICROWIRE/SPI slave serial clock input.
- spi_ss_i: MICROWIRE/SPI slave select signal input (edge-active trigger).

External pin descriptions:

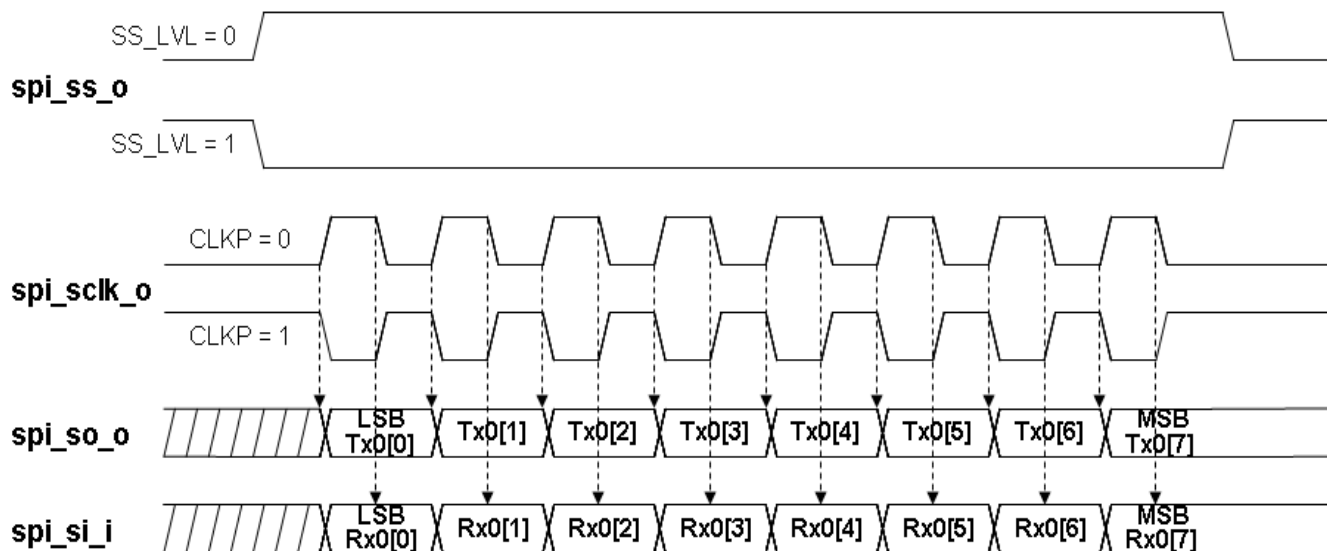
Pin Name	Pin Description	Type		GPIO
		Master	Slave	
SPI0_CLK	SPI 0 Clock	Output	Input	GPD[12]
SPI0_SO	SPI 0 Data Output	Output	Output	GPD[15]
SPI0_SI	SPI 0 Data Input	Input	Input	GPD[14]
SPI0_SS0	SPI 0 Slave Select 0	Output	Input	GPD[13]
SPI0_SS1	SPI 0 Slave Select 1	Output	Input	GPA[5]/GPD[4]
SPI1_CLK	SPI 1 Clock	Output	Input	GPB[9]
SPI1_SO	SPI 1 Data Output	Output	Output	GPB[12]
SPI1_SI	SPI 1 Data Input	Input	Input	GPB[11]
SPI1_SS0	SPI 1 Slave Select 0	Output	Input	GPB[10]
SPI1_SS1	SPI 1 Slave Select 1	Output	Input	GPA[6]/GPA[10]/GPD[0]

5.22.4 SPI (MICROWIRE) Timing Diagram (Master/Slave)



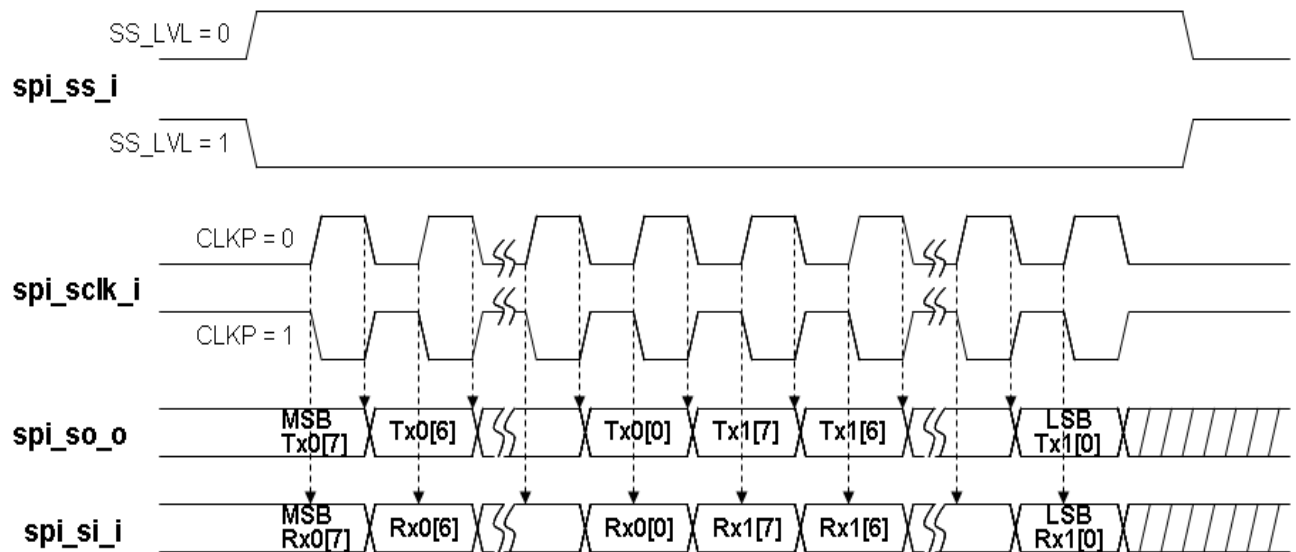
Master Mode : CNTRL[SLAVE]=0, CNTRL[LSB]=0, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
 1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0 or
 2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1

Figure 5.22-2 MICROWIRE/SPI Timing (Master)



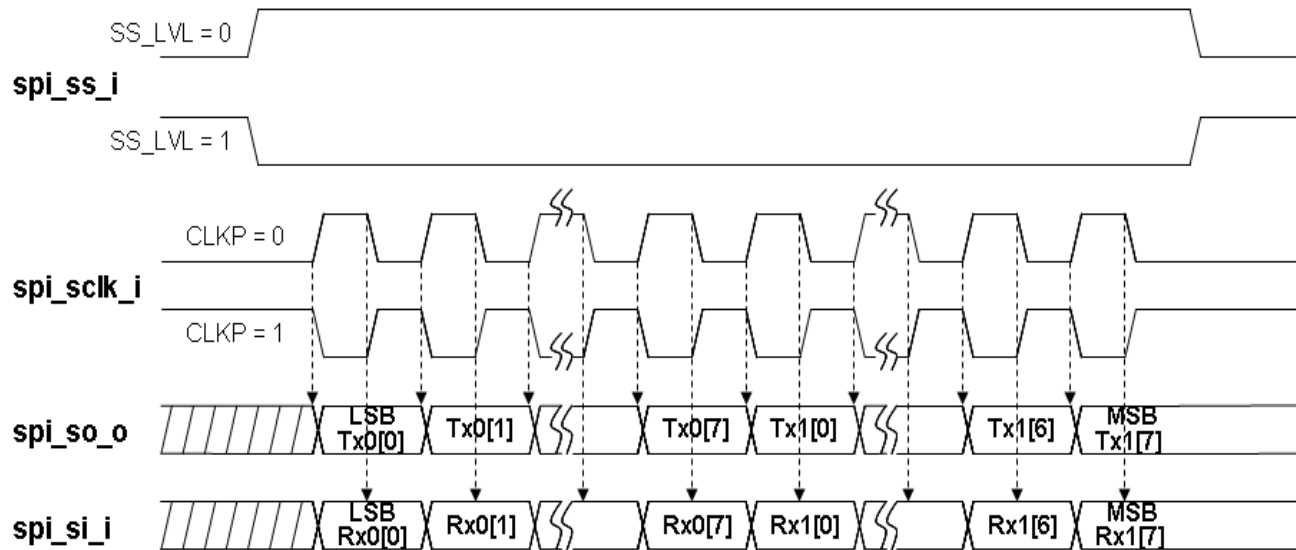
Master Mode : CNTRL[SLAVE]=0, CNTRL[LSB]=1, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
 1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1 or
 2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0

Figure 5.22-3 Alternate Phase SCLK Clock Timing (Master)



Slave Mode : CNTRL[SLAVE]=1, CNTRL[LSB]=0, CNTRL[Tx_NUM]=0x01, CNTRL[Tx_BIT_LEN]=0x08,
 1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0 or
 2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1

Figure 5.22-4 MICROWIRE/SPI Timing (Slave)



Slave Mode : CNTRL[SLAVE]=1, CNTRL[LSB]=1, CNTRL[Tx_NUM]=0x01, CNTRL[Tx_BIT_LEN]=0x08,
 1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1 or
 2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0

Figure 5.22-5 Alternate Phase SCLK Clock Timing (Slave)

5.22.5 SPI (MICROWIRE) Programming Example

When using this SPI controller as a master to access a slave device (as slave device) with following specifications:

- Data bit latches on positive edge of serial clock
- Data bit drives on negative edge of serial clock
- Data is transferred with the MSB first
- SCLK idle low.
- Only one byte transmits/receives in a transfer
- Chip select signal is active low

Basically, the following actions should be done (also, the specification of the connected slave device should be referred to when consider the following steps in detail):

1. Write a divisor into DIVIDER to determine the frequency of serial clock.
2. Write in SSR, set ASS = 0, SS_LVL = 0 and SSR[0] or SSR[1] to 1 to activate the device to be accessed.

When transmit (write) data to device:

3. Write the data to be transmitted into Tx0[7:0].

When receive (read) data from device:

4. Write 0xFFFFFFFF into Tx0.
5. Write in CNTRL, set SLAVE = 0, CLKP = 0, Rx_NEG = 0, Tx_NEG = 1, Tx_BIT_LEN = 0x08, Tx_NUM = 0x0, LSB = 0, SLEEP = 0x0 and GO_BUSY = 1 to start the transfer.
-- Wait for interrupt (if IE = 1) or polling the GO_BUSY bit until it turns to 0 --
6. Read out the received data from Rx0.
7. Go to 3) to continue another data transfer or set SSR[0] or SSR[1] to 0 to inactivate the device.

When using this SPI controller as a slave device and connected to a master device, suppose the external master device accesses the on chip SPI interface with the following specifications:

- Data bit latches on positive edge of serial clock
- Data bit drives on negative edge of serial clock
- Data is transferred with the LSB first
- SCLK idle high.
- Only one byte transmits/receives in a transfer
- Chip select signal is active high trigger.

Basically, the following actions should be done (also, the specification of the connected master device should be referred to when consider the following steps in detail):

1. Write in SSR, set SS_LVL = 1.

When transmit (write) data to device:

2. Write the data to be transmitted into Tx0[7:0].

When receive (read) data from device:

3. Write 0xFFFFFFFF into Tx0.

4. Write in CNTRL, set SLAVE = 1, CLKP = 1, Rx_NEG = 0, Tx_NEG = 1, Tx_BIT_LEN = 0x08, Tx_NUM = 0x0, LSB = 1, and GO_BUSY = 1 to start the transfer and waiting for the slave select input and serial clock input signals from the external master device.
 -- Wait for interrupt (if IE = 1) or polling the GO_BUSY bit until it turns to 0 --
5. Read out the received data from Rx0.
6. Go to 2) to continue another data transfer.

5.22.6 Wireless Joystick SPI Programming Example

While connecting to wireless joystick, SPI should be set to slave mode and SPI0_JS[0] should be set to 1.

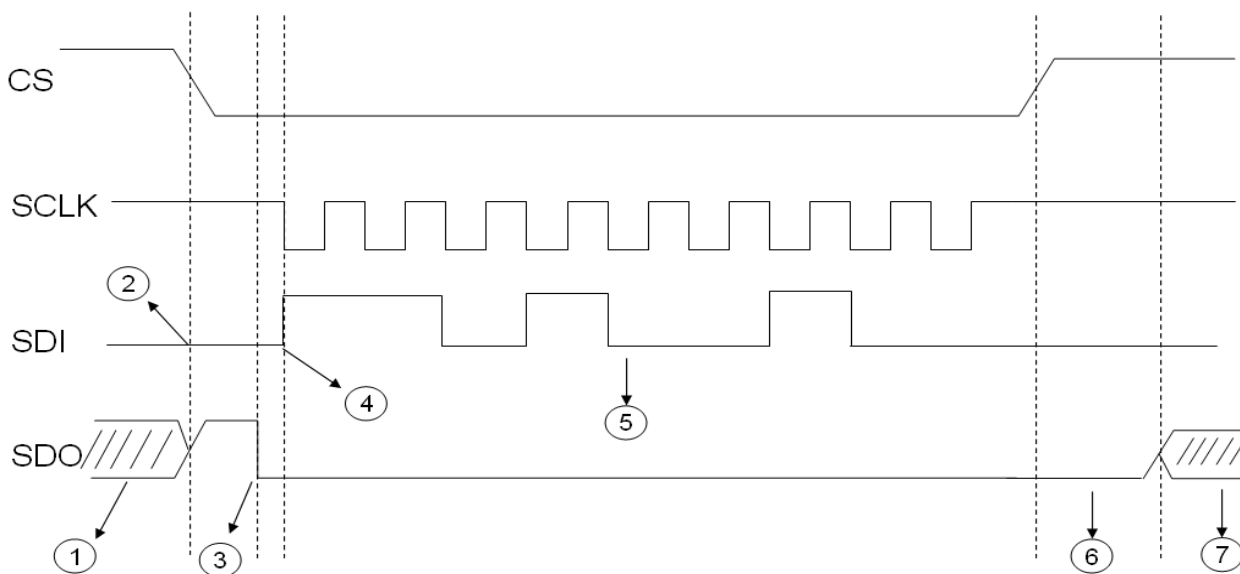
Others setting required for this mode:

1. It use rising edge to transmit data and falling edge to receive data. SPI_CNTRL[2]=1 and SPI_CNTRL[1]=0
2. Most significant bit transmit/receive first. SPI_CNTRL[10]=0
3. Clock polarity is idle high. SPI_CNTRL[11]=1
4. If master want to write data to SPI, it transfer 27bytes at one time.

If master want to read data from SPI, it read 7 bytes at one time.

Transmit/Receive Timing Diagram:

Joystick Transmit Mode

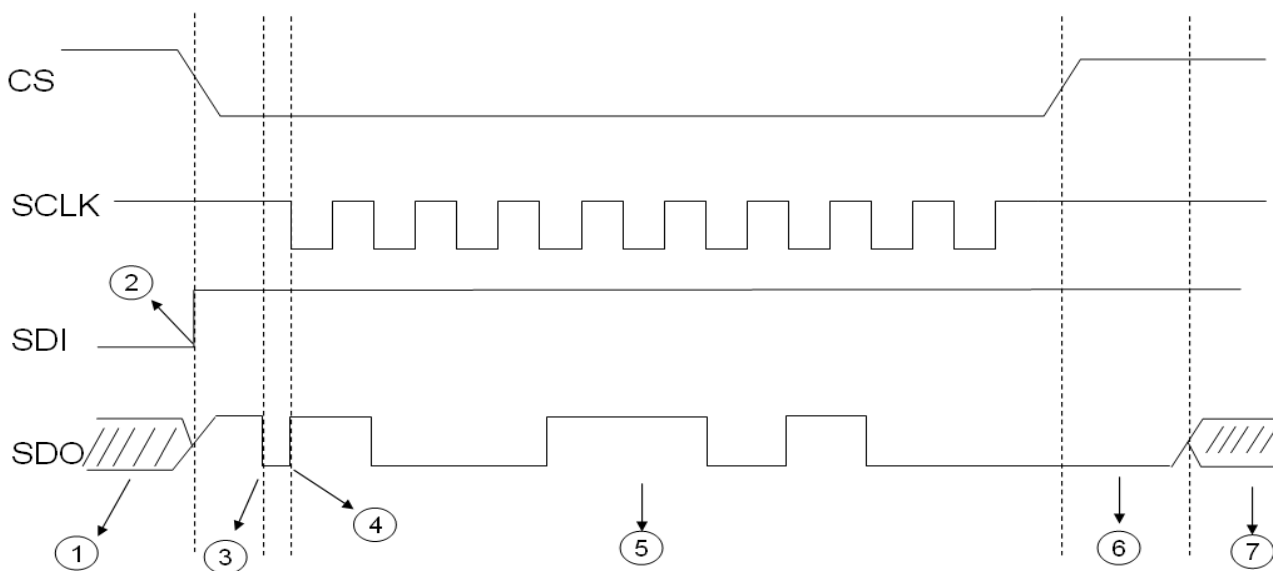


Description:

1. When CS is high, slave will keep SDO as input mode.
2. In the moment CS is setting low, master drives SDO (slave's SDI) to 0 if master want to transmit data.
3. Slave must keep SDO high if it's not ready to transmit data.
4. Drive SDO to 0 by setting SPI_JS[8] if slave is ready.

5. When master detect that slave is ready, it starts to transmit data.
6. Master will transmit 27-byte data.
7. After CS is pulled high, the transmission is finished. Slave should drive SDO low for a specified period. User can set SPI_DIVIDER to control the period.
8. After the counter in (6) reaches 0, slave will set SDO as input mode again.

Joystick Receive Mode



Description:

1. When CS is high, slave will keep SDO as input mode.
2. In the moment CS is setting low, master drives SDO to 1 if it wants to receive data.
3. Slave must keep SDO high if it's not ready to receive data.
4. Drive SDO to 0 by setting SPI_JS[8] if slave is ready.
5. When master detect that slave is ready, it starts to receive data.
6. Master will receive 7-byte data
7. After CS is pulled high, the transmission is finished. Slave should drive SDO low for a specified period. User can set SPI_DIVIDER to control the period.
8. After the counter in (6) reaches 0, slave will set SDO as input mode again.

SPI module will issue an interrupt to CPU when CS is activated and de-activated. And it will also issue an interrupt whenever it transmit or receive 8-byte data. JS_INT_FLAG[2:0] are flags to show what kind of interrupt is happened now. Software can update the buffer if it's an 8-byte data interrupt. Notice that the transmit/receive size are not limited to 7/27 bytes.

5.22.7 SPI0 (MICROWIRE) Serial Interface Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SPIMS0_BA = 0xB800_C000				
SPI0_CNTRL	SPIMS0_BA + 0x00	R/W	Control and Status Register	0x0000_0004
SPI0_DIVIDER	SPIMS0_BA + 0x04	R/W	Clock Divider Register	0x0000_0000
SPI0_SSR	SPIMS0_BA + 0x08	R/W	Slave Select Register	0x0000_0000
Reserved	SPIMS0_BA + 0x0C	N/A	Reserved	N/A
SPI0_Rx0	SPIMS0_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI0_Rx1	SPIMS0_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI0_Rx2	SPIMS0_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI0_Rx3	SPIMS0_BA + 0x1C	R	Data Receive Register 3	0x0000_0000
SPI0_Tx0	SPIMS0_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI0_Tx1	SPIMS0_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI0_Tx2	SPIMS0_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI0_Tx3	SPIMS0_BA + 0x1C	W	Data Transmit Register 3	0x0000_0000
SPI0_JS	SPIMS0_BA + 0x20	R/W	Dongle joystick control register	0x0000_0100
SPI0_EDMA	SPIMS0_BA + 0x28	R/W	EDMA control register	0x0000_0000

NOTE 1: When software programs CNTRL, the GO_BUSY bit should be written last.

5.22.8 SPI 0 (MICROWIRE) Control Register Description

Control and Status Register (CNTRL)

Register	Offset	R/W	Description	Reset Value
SPI0_CNTRL	SPIMS0_BA + 0x00	R/W	Control and Status Register	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			BYTE_ENDIN	BYTE_SLEEP	SLAVE	IE	IF
15	14	13	12	11	10	9	8
SLEEP				CLKP	LSB	Tx_NUM	
7	6	5	4	3	2	1	0
Tx_BIT_LEN					Tx_NEG	Rx_NEG	GO_BUSY

Bits	Descriptions	
[31:21]	Reserved	Reserved
[20]	BYTE_ENDIN	BYTE ENDIN 0 = Disable the BYTE ENDIN. 1 = Active the BYTE ENDIN. Only the 16, 24, and 32 bits which define in TX_BIT_LEN are support. When the transmission is set as MSB first and the BYTE ENDIN bit is set high, the data store in the TX buffer will be arranged in order as [BYTE0, BYTE1, BYTE2, BYTE3] in TX_BIT_LEN = 32 bit mode, and the sequence of transmitted data will be BYTE0, BYTE1, BYTE2, and BYTE3. If the TX_BIT_LEN is set as 24-bit mode, the data in TX buffer will be arranged as [BYTE3, BYTE0, BYTE1, BYTE2] and the BYTE0, BYTE1, and BYTE2 will be transmitted step by step in MSB first. The rule of 16-bit mode is the same above.
[19]	BYTE_SLEEP	BYTE SLEEP 0 = There is not SLEEP interval among each byte in word mode transaction. 1 = There is SLEEP interval (defined in [15:12]) among each byte in word transaction mode.
[18]	SLAVE	SLAVE Mode Indication 0 = Master mode. 1 = Slave mode.
[17]	IE	Interrupt Enable 0 = Disable MICROWIRE/SPI Interrupt. 1 = Enable MICROWIRE/SPI Interrupt.
[16]	IF	Interrupt Flag 0 = It indicates that the transfer dose not finish yet. 1 = It indicates that the transfer is done. The interrupt flag is set if it was enable. NOTE: This bit is read only, but can be cleared by writing 1 to this bit.

[15:12]	SLEEP	Suspend Interval (master only) These four bits provide the configuration of suspend interval between two successive transmit/receive in a transfer. The default value is 0. When CNTRL[Tx_NUM] = 00, setting this field has no effect on transfer. The desired interval is obtained according to the following equation (from the last falling edge of current sclk to the first rising edge of next sclk): $(CNTRL[SLEEP] + 2) * \text{period of SCLK}$ SLEEP = 0x0 ... 2 SCLK clock cycle SLEEP = 0x1 ... 3 SCLK clock cycle SLEEP = 0xe ... 16 SCLK clock cycle SLEEP = 0xf ... 17 SCLK clock cycle
[11]	CLKP	Clock Polarity 0 = SCLK idle low. 1 = SCLK idle high.
[10]	LSB	Send LSB First 0 = The MSB is transmitted/received first (which bit in TxX/RxX register that is depends on the Tx_BIT_LEN field in the CNTRL register). 1 = The LSB is sent first on the line (bit TxX[0]), and the first bit received from the line will be put in the LSB position in the Rx register (bit RxX[0]).
[9:8]	Tx_NUM	Transmit/Receive Numbers This field specifies how many transmit/receive numbers should be executed in one transfer. 00 = Only one transmit/receive will be executed in one transfer. 01 = Two successive transmit/receive will be executed in one transfer. 10 = Three successive transmit/receive will be executed in one transfer. 11 = Four successive transmit/receive will be executed in one transfer.
[7:3]	Tx_BIT_LEN	Transmit Bit Length This field specifies how many bits are transmitted in one transmit/receive. Up to 32 bits can be transmitted. Tx_BIT_LEN = 0x01 ... 1 bit Tx_BIT_LEN = 0x02 ... 2 bits Tx_BIT_LEN = 0x1f ... 31 bits Tx_BIT_LEN = 0x00 ... 32 bits
[2]	Tx_NEG	Transmit On Negative Edge 0 = The spi_so_o signal is changed on the rising edge of spi_sclk_o in master mode or spi_sclk_i in slave mode. 1 = The spi_so_o signal is changed on the falling edge of spi_sclk_o in master mode or spi_sclk_i in slave mode..
[1]	Rx_NEG	Receive On Negative Edge 0 = The spi_si_i signal is latched on the rising edge of spi_sclk_o in master mode or spi_sclk_i in slave mode.. 1 = The spi_si_i signal is latched on the falling edge of spi_sclk_o in master mode or spi_sclk_i in slave mode..

[0]	GO_BUSY	Go and Busy Status 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit starts the transfer. This bit remains set during the transfer and is automatically cleared after transfer finished. NOTE: All registers should be set before writing 1 to the GO_BUSY bit in the CNTRL register. When a transfer is in progress, writing to any register of the MICROWIRE/SPI master/slave core has no effect.

Divider Register (DIVIDER)

Register	Offset	R/W	Description	Reset Value
SPI0_DIVIDER	SPIMS0_BA + 0x04	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DIVIDER[15:8]							
7	6	5	4	3	2	1	0
DIVIDER[7:0]							

Bits	Descriptions
[15:0]	Clock Divider Register The value in this field is the frequency divider of the system clock pclk to generate the serial clock on the output spi_sclk_o. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{pclk}}{(DIVIDER + 1) * 2}$ NOTE: Suggest DIVIDER should be at least 1 in master mode. In slave mode, the period of sclk input shall be equal or over 6 times pclk at least.

Slave Select Register (SSR)

Register	Offset	R/W	Description	Reset Value
SPI0_SSR	SPIMS0_BA + 0x08	R/W	Slave Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		LTRIG_FLAG	SS_LTRIG	ASS	SS_LVL	SSR[1:0]	

Bits	Descriptions	
[5]	LTRIG_FLAG	Level Trigger Flag When the SS_LTRIG bit is set in slave mode, this bit can be read to indicate the received bit number is met the requirement or not. 1: The received number and received bits met the requirement which defines in TX_NUM and TX_BIT_LEN among one transaction. 0: One of the received number and the received bit length doesn't meet the requirement in one transaction.
[4]	SS_LTRIG	Slave Select Level Trigger 0: The input slave select signal is edge-trigger. This is default value. 1: The slave select signal will be level-trigger. It depends on SS_LVL to decide the signal is active low or active high.
[3]	ASS	Automatic Slave Select (master only) 0 = If this bit is cleared, slave select signals are asserted and de-asserted by setting and clearing related bits in SSR[1:0] register. 1 = If this bit is set, spi_ss_o[1:0] signals are generated automatically. It means that device/slave select signal, which is set in the SSR register is asserted by the MICROWIRE/SPI controller when transmit/receive is started by setting CNTRL[GO_BUSY], and is de-asserted after every transmit/receive is finished.
[2]	SS_LVL	Slave Select Active Level It defines the active level of device/slave select signal (spi_ss_o[1:0]). 0 = The spi_ss_o slave select signal is active Low. 1 = The spi_ss_o slave select signal is active High.

[1:0]	SSR	Slave Select Register (master only) If SSR[ASS] bit is cleared, writing 1 to any bit location of this field sets the proper spi_ss_o[1:0] line to an active state and writing 0 sets the line back to inactive state. If SSR[ASS] bit is set, writing 1 to any bit location of this field will select appropriate spi_ss_o[1:0] line to be automatically driven to active state for the duration of the transmit/receive, and will be driven to inactive state for the rest of the time. (The active level of spi_ss_o[1:0] is specified in SSR[SS_LVL]). NOTE: This interface can only drive one device/slave at a given time. Therefore, the slave select of the selected device must be set to its active level before starting any read or write transfer. NOTE: spi_ss_o[0] is also defined as device/slave select input spi_ss_i signal in slave mode. And that the slave select input spi_ss_i must be driven by edge active trigger which level depend on the SS_LVL setting, otherwise the SPI slave core will go into dead path until the edge active trigger again or reset the SPI core by software.

Data Receive Register (RX)

Register	Offset	R/W	Description	Reset Value
SPI0_Rx0	SPIMS0_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI0_Rx1	SPIMS0_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI0_Rx2	SPIMS0_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI0_Rx3	SPIMS0_BA + 0x1C	R	Data Receive Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Rx [31:24]							
23	22	21	20	19	18	17	16
Rx [23:16]							
15	14	13	12	11	10	9	8
Rx [15:8]							
7	6	5	4	3	2	1	0
Rx [7:0]							

Bits	Descriptions	
[31:0]	Rx	Data Receive Register The Data Receive Registers hold the value of received data of the last executed transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if CNTRL[Tx_BIT_LEN] is set to 0x08 and CNTRL[Tx_NUM] is set to 0x0, bit Rx0[7:0] holds the received data. NOTE: The Data Receive Registers are read only registers. A Write to these registers will actually modify the Data Transmit Registers because those registers share the same flip-flops.

Data Transmit Register (TX)

Register	Offset	R/W	Description	Reset Value
SPI0_Tx0	SPIMS0_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI0_Tx1	SPIMS0_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI0_Tx2	SPIMS0_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI0_Tx3	SPIMS0_BA + 0x1C	W	Data Transmit Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Tx [31:24]							
23	22	21	20	19	18	17	16
Tx [23:16]							
15	14	13	12	11	10	9	8
Tx [15:8]							
7	6	5	4	3	2	1	0
Tx [7:0]							

Bits	Descriptions	
[31:0]	Tx	Data Transmit Register The Data Transmit Registers hold the data to be transmitted in the next transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if CNTRL[Tx_BIT_LEN] is set to 0x08 and the CNTRL[Tx_NUM] is set to 0x0, the bit Tx0[7:0] will be transmitted in next transfer. If CNTRL[Tx_BIT_LEN] is set to 0x00 and CNTRL[Tx_NUM] is set to 0x3, the core will perform four 32-bit transmit/receive successive using the same setting (the order is Tx0[31:0], Tx1[31:0], Tx2[31:0], Tx3[31:0]). NOTE: The RxX and TxX registers share the same flip-flops, which mean that what is received from the input data line in one transfer will be transmitted on the output data line in the next transfer if no write access to the TxX register is executed between the transfers.

Dongle Joystick Control Register (JS)

Register	Offset	R/W	Description	Reset Value
SPI0_JS	SPIMS0_BA + 0x20	R/W	Dongle joystick control register	0x0000_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							READY
7	6	5	4	3	2	1	0
JS_INT_FLAG			JS_RW	Reserved			JS

Bits	Descriptions	
[8]	READYB	Slave is ready to transmit/receive data In Dongle Joystick mode, SDO will be set to input mode when CS (chip select) is high. When the outside master is pull CS low, slave should set SDO as output mode. This READY bit can control SDO output at this situation. READYB=1 means slave is not ready to transfer or receive data. READYB=0 means slave is ready to transmit/receive data.
[7:5]	JS_INT_FLAG	Joystick Mode Interrupt Flag JS_INT_FLAG[5] → CS is activated. JS_INT_FLAG[6] → 8-byte data is available in the buffer. JS_INT_FLAG[7] → CS is de-activated. These bits are read-only.
[4]	JS_RW	Read/Write Mode At the moment that CS is set low, SDI=1 means master wants to read data from slave; SDI=0 means master wants to write data to slave. RW will record SDI value at that moment. This bit is read-only.
[0]	JS	Dongle Joystick mode (slave only) Set this bit to 1 if SPI0 is connecting to Dongle Joystick SPI interface.

EDMA Control Register (EDMACTL)

Register	Offset	R/W	Description	Reset Value
SPI0_EDMA	SPIMS0_BA + 0x28	R/W	EDMA mode control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						EDMA_RW	EDMA_GO

Bits	Descriptions	
[1]	EDMA_RW	EDMA Read or EDMA Write 0: EDMA write to SPI module 1: EDMA read from SPI module
[0]	EDMA_GO	EDMA start Set this bit to 1 will start the EDMA process. SPI module will issue edma_request to EDMA module automatically and it will be clear after the EDAM transaction done. If using EDMA mode to transmit data, remember not to set GO_BUSY bit of SPI_CNTRL register. The EDMA controller inside SPI module will set it automatically whenever necessary.

5.22.9 SPI1 (MICROWIRE) Serial Interface Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SPIMS1_BA = 0xB800_C400				
SPI1_CNTRL	SPIMS1_BA + 0x00	R/W	Control and Status Register	0x0000_0004
SPI1_DIVIDER	SPIMS1_BA + 0x04	R/W	Clock Divider Register	0x0000_0000
SPI1_SSR	SPIMS1_BA + 0x08	R/W	Slave Select Register	0x0000_0000
Reserved	SPIMS1_BA + 0x0C	N/A	Reserved	N/A
SPI1_Rx0	SPIMS1_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI1_Rx1	SPIMS1_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI1_Rx2	SPIMS1_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI1_Rx3	SPIMS1_BA + 0x1C	R	Data Receive Register 3	0x0000_0000
SPI1_Tx0	SPIMS1_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI1_Tx1	SPIMS1_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI1_Tx2	SPIMS1_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI1_Tx3	SPIMS1_BA + 0x1C	W	Data Transmit Register 3	0x0000_0000
SPI1_JS	SPIMS1_BA + 0x20	R/W	Dongle joystick control register	0x0000_0100
SPI1_EDMA	SPIMS1_BA + 0x28	R/W	EDMA control register	0x0000_0000

NOTE 1: When software programs CNTRL, the GO_BUSY bit should be written last.

5.22.10 SPI 1 (MICROWIRE) Control Register Description

Control and Status Register (CNTRL)

Register	Offset	R/W	Description	Reset Value
SPI1_CNTRL	SPIMS1_BA + 0x00	R/W	Control and Status Register	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			BYTE_ENDIN	BYTE_SLEEP	SLAVE	IE	IF
15	14	13	12	11	10	9	8
SLEEP				CLKP	LSB	Tx_NUM	
7	6	5	4	3	2	1	0
Tx_BIT_LEN					Tx_NEG	Rx_NEG	GO_BUSY

Bits	Descriptions	
[31:21]	Reserved	Reserved
[20]	BYTE_ENDIN	BYTE ENDIN 0 = Disable the BYTE ENDIN. 1 = Active the BYTE ENDIN. Only the 16, 24, and 32 bits which define in TX_BIT_LEN are support. When the transmission is set as MSB first and the BYTE ENDIN bit is set high, the data store in the TX buffer will be arranged in order as [BYTE0, BYTE1, BYTE2, BYTE3] in TX_BIT_LEN = 32 bit mode, and the sequence of transmitted data will be BYTE0, BYTE1, BYTE2, and BYTE3. If the TX_BIT_LEN is set as 24-bit mode, the data in TX buffer will be arranged as [BYTE3, BYTE0, BYTE1, BYTE2] and the BYTE0, BYTE1, and BYTE2 will be transmitted step by step in MSB first. The rule of 16-bit mode is the same above.
[19]	BYTE_SLEEP	BYTE SLEEP 0 = There is not SLEEP interval among each byte in word mode transaction. 1 = There is SLEEP interval (defined in [15:12]) among each byte in word transaction mode.
[18]	SLAVE	SLAVE Mode Indication 0 = Master mode. 1 = Slave mode.
[17]	IE	Interrupt Enable 0 = Disable MICROWIRE/SPI Interrupt. 1 = Enable MICROWIRE/SPI Interrupt.
[16]	IF	Interrupt Flag 0 = It indicates that the transfer dose not finish yet. 1 = It indicates that the transfer is done. The interrupt flag is set if it was enable. NOTE: This bit is read only, but can be cleared by writing 1 to this bit.

[15:12]	SLEEP	Suspend Interval (master only) These four bits provide the configuration of suspend interval between two successive transmit/receive in a transfer. The default value is 0. When CNTRL[Tx_NUM] = 00, setting this field has no effect on transfer. The desired interval is obtained according to the following equation (from the last falling edge of current sclk to the first rising edge of next sclk): $(CNTRL[SLEEP] + 2) * \text{period of SCLK}$ SLEEP = 0x0 ... 2 SCLK clock cycle SLEEP = 0x1 ... 3 SCLK clock cycle SLEEP = 0xe ... 16 SCLK clock cycle SLEEP = 0xf ... 17 SCLK clock cycle
[11]	CLKP	Clock Polarity 0 = SCLK idle low. 1 = SCLK idle high.
[10]	LSB	Send LSB First 0 = The MSB is transmitted/received first (which bit in TxX/RxX register that is depends on the Tx_BIT_LEN field in the CNTRL register). 1 = The LSB is sent first on the line (bit TxX[0]), and the first bit received from the line will be put in the LSB position in the Rx register (bit RxX[0]).
[9:8]	Tx_NUM	Transmit/Receive Numbers This field specifies how many transmit/receive numbers should be executed in one transfer. 00 = Only one transmit/receive will be executed in one transfer. 01 = Two successive transmit/receive will be executed in one transfer. 10 = Three successive transmit/receive will be executed in one transfer. 11 = Four successive transmit/receive will be executed in one transfer.
[7:3]	Tx_BIT_LEN	Transmit Bit Length This field specifies how many bits are transmitted in one transmit/receive. Up to 32 bits can be transmitted. Tx_BIT_LEN = 0x01 ... 1 bit Tx_BIT_LEN = 0x02 ... 2 bits Tx_BIT_LEN = 0x1f ... 31 bits Tx_BIT_LEN = 0x00 ... 32 bits
[2]	Tx_NEG	Transmit On Negative Edge 0 = The spi_so_o signal is changed on the rising edge of spi_sclk_o in master mode or spi_sclk_i in slave mode. 1 = The spi_so_o signal is changed on the falling edge of spi_sclk_o in master mode or spi_sclk_i in slave mode..
[1]	Rx_NEG	Receive On Negative Edge 0 = The spi_si_i signal is latched on the rising edge of spi_sclk_o in master mode or spi_sclk_i in slave mode.. 1 = The spi_si_i signal is latched on the falling edge of spi_sclk_o in master mode or spi_sclk_i in slave mode..

[0]	GO_BUSY	Go and Busy Status 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit starts the transfer. This bit remains set during the transfer and is automatically cleared after transfer finished. NOTE: All registers should be set before writing 1 to the GO_BUSY bit in the CNTRL register. When a transfer is in progress, writing to any register of the MICROWIRE/SPI master/slave core has no effect.

Divider Register (DIVIDER)

Register	Offset	R/W	Description	Reset Value
SPI1_DIVIDER	SPIMS1_BA + 0x04	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
DIVIDER[15:8]							
7	6	5	4	3	2	1	0
DIVIDER[7:0]							

Bits	Descriptions
[15:0]	Clock Divider Register The value in this field is the frequency divider of the system clock pclk to generate the serial clock on the output spi_sclk_o. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{pclk}}{(DIVIDER + 1) * 2}$ NOTE: Suggest DIVIDER should be at least 1 in master mode. In slave mode, the period of sclk input shall be equal or over 6 times pclk at least.

Slave Select Register (SSR)

Register	Offset	R/W	Description	Reset Value
SPI1_SSR	SPIMS1_BA + 0x08	R/W	Slave Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		LTRIG_FLAG	SS_LTRIG	ASS	SS_LVL	SSR[1:0]	

Bits	Descriptions	
[5]	LTRIG_FLAG	Level Trigger Flag When the SS_LTRIG bit is set in slave mode, this bit can be read to indicate the received bit number is met the requirement or not. 1: The received number and received bits met the requirement which defines in TX_NUM and TX_BIT_LEN among one transaction. 0: One of the received number and the received bit length doesn't meet the requirement in one transaction.
[4]	SS_LTRIG	Slave Select Level Trigger 0: The input slave select signal is edge-trigger. This is default value. 1: The slave select signal will be level-trigger. It depends on SS_LVL to decide the signal is active low or active high.
[3]	ASS	Automatic Slave Select (master only) 0 = If this bit is cleared, slave select signals are asserted and de-asserted by setting and clearing related bits in SSR[1:0] register. 1 = If this bit is set, spi_ss_o[1:0] signals are generated automatically. It means that device/slave select signal, which is set in the SSR register is asserted by the MICROWIRE/SPI controller when transmit/receive is started by setting CNTRL[GO_BUSY], and is de-asserted after every transmit/receive is finished.
[2]	SS_LVL	Slave Select Active Level It defines the active level of device/slave select signal (spi_ss_o[1:0]). 0 = The spi_ss_o slave select signal is active Low. 1 = The spi_ss_o slave select signal is active High.

[1:0]	SSR	Slave Select Register (master only) If SSR[ASS] bit is cleared, writing 1 to any bit location of this field sets the proper spi_ss_o[1:0] line to an active state and writing 0 sets the line back to inactive state. If SSR[ASS] bit is set, writing 1 to any bit location of this field will select appropriate spi_ss_o[1:0] line to be automatically driven to active state for the duration of the transmit/receive, and will be driven to inactive state for the rest of the time. (The active level of spi_ss_o[1:0] is specified in SSR[SS_LVL]). NOTE: This interface can only drive one device/slave at a given time. Therefore, the slave select of the selected device must be set to its active level before starting any read or write transfer. NOTE: spi_ss_o[0] is also defined as device/slave select input spi_ss_i signal in slave mode. And that the slave select input spi_ss_i must be driven by edge active trigger which level depend on the SS_LVL setting, otherwise the SPI slave core will go into dead path until the edge active trigger again or reset the SPI core by software.

Data Receive Register (RX)

Register	Offset	R/W	Description	Reset Value
SPI1_Rx0	SPIMS1_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI1_Rx1	SPIMS1_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI1_Rx2	SPIMS1_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI1_Rx3	SPIMS1_BA + 0x1C	R	Data Receive Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Rx [31:24]							
23	22	21	20	19	18	17	16
Rx [23:16]							
15	14	13	12	11	10	9	8
Rx [15:8]							
7	6	5	4	3	2	1	0
Rx [7:0]							

Bits	Descriptions	
[31:0]	Rx	Data Receive Register The Data Receive Registers hold the value of received data of the last executed transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if CNTRL[Tx_BIT_LEN] is set to 0x08 and CNTRL[Tx_NUM] is set to 0x0, bit Rx0[7:0] holds the received data. NOTE: The Data Receive Registers are read only registers. A Write to these registers will actually modify the Data Transmit Registers because those registers share the same flip-flops.

Data Transmit Register (TX)

Register	Offset	R/W	Description	Reset Value
SPI1_Tx0	SPIMS1_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI1_Tx1	SPIMS1_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI1_Tx2	SPIMS1_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI1_Tx3	SPIMS1_BA + 0x1C	W	Data Transmit Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Tx [31:24]							
23	22	21	20	19	18	17	16
Tx [23:16]							
15	14	13	12	11	10	9	8
Tx [15:8]							
7	6	5	4	3	2	1	0
Tx [7:0]							

Bits	Descriptions	
[31:0]	Tx	Data Transmit Register The Data Transmit Registers hold the data to be transmitted in the next transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if CNTRL[Tx_BIT_LEN] is set to 0x08 and the CNTRL[Tx_NUM] is set to 0x0, the bit Tx0[7:0] will be transmitted in next transfer. If CNTRL[Tx_BIT_LEN] is set to 0x00 and CNTRL[Tx_NUM] is set to 0x3, the core will perform four 32-bit transmit/receive successive using the same setting (the order is Tx0[31:0], Tx1[31:0], Tx2[31:0], Tx3[31:0]). NOTE: The RxX and TxX registers share the same flip-flops, which mean that what is received from the input data line in one transfer will be transmitted on the output data line in the next transfer if no write access to the TxX register is executed between the transfers.

Dongle Joystick Control Register (JS)

Register	Offset	R/W	Description	Reset Value
SPI1_JS	SPIMS1_BA + 0x20	R/W	Dongle joystick control register	0x0000_0100

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							READY
7	6	5	4	3	2	1	0
JS_INT_FLAG			JS_RW	Reserved			JS

Bits	Descriptions	
[8]	READYB	Slave is ready to transmit/receive data In Dongle Joystick mode, SDO will be set to input mode when CS (chip select) is high. When the outside master is pull CS low, slave should set SDO as output mode. This READY bit can control SDO output at this situation. READYB=1 means slave is not ready to transfer or receive data. READYB=0 means slave is ready to transmit/receive data.
[7:5]	JS_INT_FLAG	Joystick Mode Interrupt Flag JS_INT_FLAG[5] → CS is activated. JS_INT_FLAG[6] → 8-byte data is available in the buffer. JS_INT_FLAG[7] → CS is de-activated. These bits are read-only.
[4]	JS_RW	Read/Write Mode At the moment that CS is set low, SDI=1 means master wants to read data from slave; SDI=0 means master wants to write data to slave. RW will record SDI value at that moment. This bit is read-only.
[0]	JS	Dongle Joystick mode (slave only) Set this bit to 1 if SPI0 is connecting to Dongle Joystick SPI interface.

EDMA Control Register (EDMACTL)

Register	Offset	R/W	Description	Reset Value
SPI1_EDMA	SPIMS1_BA + 0x28	R/W	EDMA mode control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						EDMA_RW	EDMA_GO

Bits	Descriptions	
[1]	EDMA_RW	EDMA Read or EDMA Write 0: EDMA write to SPI module 1: EDMA read from SPI module
[0]	EDMA_GO	EDMA start Set this bit to 1 will start the EDMA process. SPI module will issue edma_request to EDMA module automatically and it will be clear after the EDAM transaction done. If using EDMA mode to transmit data, remember not to set GO_BUSY bit of SPI_CNTRL register. The EDMA controller inside SPI module will set it automatically whenever necessary.

5.23 Analog to Digital Controller (ADC)

5.23.1 Overview

The 10-bit analog to digital converter (ADC) in this chip is a successive approximation type ADC with 6-channel inputs, 2 inputs of them are dedicated for audio recorder. It needs 34 cycles to convert one sample, the maximum conversion rate is 500K/sec, so the maximum input clock to ADC is 17MHz, so and the operating voltage range is 3.3V +/- 10%. The power down mode is supported in the ADC.

The touch screen interfaces are supported in this chip, it contains 4-wire, 5-wire and 8-wire analog resistive touch screen. The four switches to bias XP, XM, YP, YM are embedded in this chip. The CPU could access the ADC control register by APB bus, and the ADC output an interrupt signal to AIC to represent the completion of conversion.

Beside the 10-bit ADC, a 8 levels voltage detector is included in this chip. The detector result is independent with power supply, and it could give the system a warning signal when battery voltage is lower than an absolute reference voltage.

5.23.2 Features

- Maximum conversion rate: 500K sample per second
- Power supply voltage: 3.3V
- Analog input voltage range: 0 – 3.3 volts
- Touch screen semi-auto/auto auto with moving average filter(MAV) conversion modes supported
- Waiting for trigger mode supports
- Standby mode supports
- 8-level voltage detector

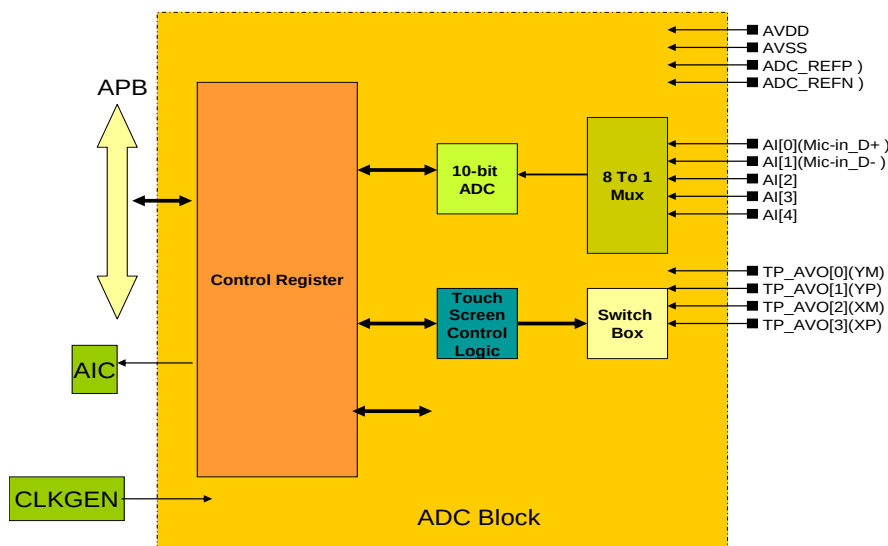


Figure 5.23-1 ADC Block Diagram

5.23.3 ADC Functional Description

5.23.3.1 Interface to Touch Screen

The touch screen control logic and the switch box could control the 4-wire, 5-wire and 8-wire type touch screen. The following figures show the interface for 4-wire, 5-wire and 8-wire touch screen respectively.

Note that, the four switches to bias XP, XM, YP, YM has conduction resistance under 5 ohm. And the pull up PMOS have 200K ohm typically.

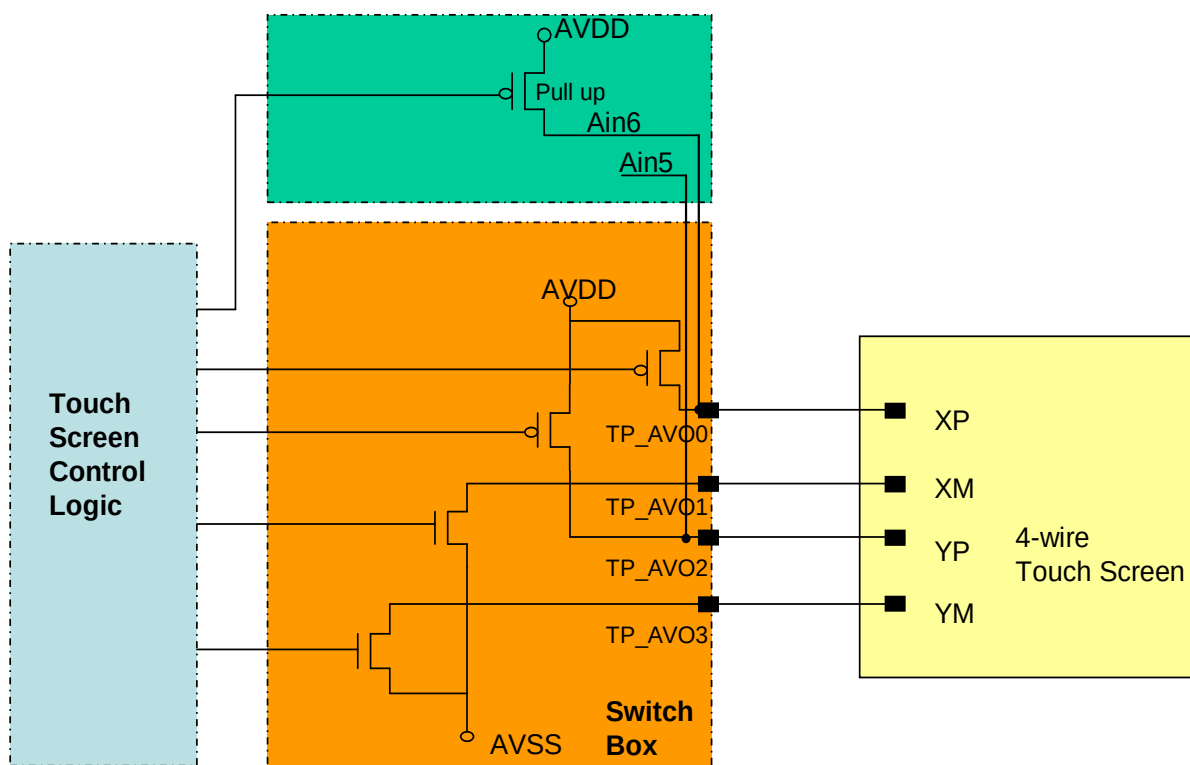


Figure 5.23-2 Interface for 4-wire Touch Screen

5.23.4 Waiting for Trigger Mode

This chip also supports the waiting for trigger mode in the touch screen control logic. In the 4-wire touch screen, when in waiting for trigger mode, the YM is connected to AVSS, XP is pulled high by a weak pull high PMOS, when the Stylus is down on the touch screen panel, Vin6 will receive a falling edge, then a active signal will be generated in INT output signal.

In the 5-wire touch screen, in the waiting for trigger mode, the YM is connected to AVSS, and the Sense will be pulled high. Vin6 also be in charge to detect the falling edge in this type of touch screen. The waiting for trigger mode for 8-wire touch screen is same as 4-wire.

5.23.5 Interface Mode

There are three control modes for ADC, normal conversion mode, semi-auto conversion mode, and auto conversion mode.

The normal conversion mode is for general purpose ADC, user could use the control register to control the 8 to 1 MUX to select analog input channel, start to conversion, wait interrupt or polling flag to confirm conversion finished, then to read the digital data. The conversion time is 34 ADC input clocks for each sample.

The semi-auto conversion mode and auto conversion mode are designed for touch screen. When the semi-auto conversion mode is proposed, the following procedures need to be followed.

1. Write the control register to start X-position detection. When starting detection, the proper switches will be enabled in switch box.
2. Waiting interrupt or polling status flag to confirm X-position is detected.
3. Write the control register to start Y-position detection. In the same way, the proper switches will be enabled in switch box.
4. Waiting interrupt or polling status flag to confirm Y-position id detected.
5. Read the X-position and Y-position in control registers.

In semi-auto conversion mode, for each position, the detection time is (2* 50 ADC clocks + program execution time between X-position and Y-position).

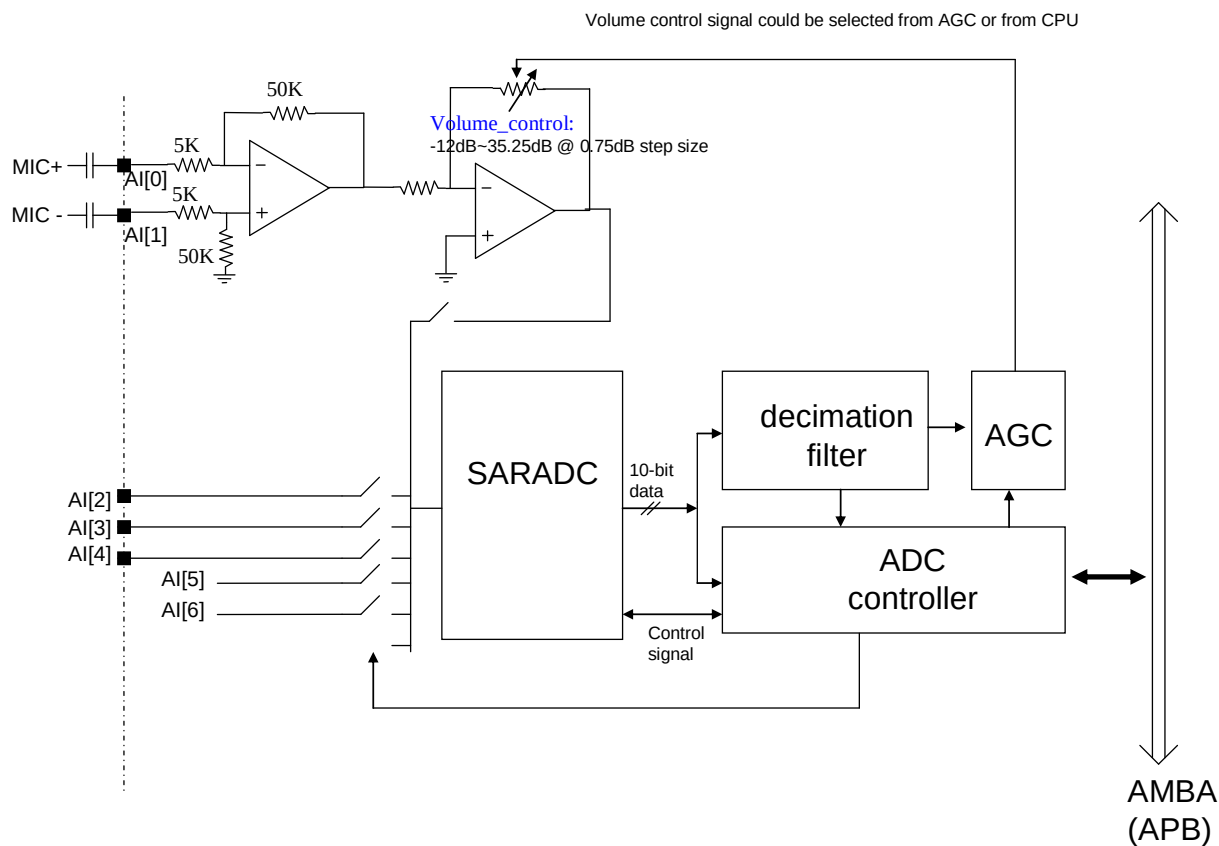
When the auto conversion mode is proposed, X-position and Y-position will be detected sequentially, the user could wait interrupt or polling status flag to confirm the detection finished, then to read the X-position and Y-position data in control register. In auto conversion mode, the detection time is 100 ADC clocks.

5.23.6 Standby mode

A standby mode is provided for ADC. When the ADC enable bit is cleared to 0, and the ADC clock is disable, the ADC enter standby mode. In the standby mode, the consumed current from AVDD will be less than 5uA. Note that the last conversion data is not cleared.

5.23.7 Recording path and AGC

The audio recording path converts the audio analog to ditial data by means of the ADC hardware. When the recording path is in usage, other data-conversion ADC function can' t operate.



Normal ADC path: (Ain2 – Ain7) → SARADC → ADC controller → APB
 Record path: Ain0 → OPAMP → SARADC → decimation filter → ADC controller → APB

5.23.8 Analog Macro Control Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Address	R/W	Description	Reset Value
ADC_BA = 0xB800_E000				
ADC_CON	ADC_BA+0x000	R/W	ADC control register	0x0000_0000
ADC_TSC	ADC_BA+0x004	R/W	Touch screen control register	0x0000_0000
ADC_DLY	ADC_BA+0x008	R/W	ADC delay register	0x0000_0000
ADC_XDATA	ADC_BA+0x00C	R	ADC XDATA register	0x0000_0000
ADC_YDATA	ADC_BA+0x010	R	ADC YDATA register	0x0000_0000
Reserved	ADC_BA+0x014	R	Reserved	0x0000_0000
Reserved	ADC_BA+0x018	R	Reserved	0x0000_0000
AUDIO_CON	ADC_BA+0x01C	R/W	Audio control register	0x0000_0000
AUDIO_BUF0	ADC_BA+0x020	R/W	Audio data register	0x0000_0000
AUDIO_BUF1	ADC_BA+0x024	R/W	Audio data register	0x0000_0000
AUDIO_BUF2	ADC_BA+0x028	R/W	Audio data register	0x0000_0000
AUDIO_BUF3	ADC_BA+0x02C	R/W	Audio data register	0x0000_0000
AGCP1	ADC_BA+0x030	R/W	AGC Parameter1 register	0x0070_B310
AGC_CON	ADC_BA+0x034	R/W	AGC Control register	0x0000_0000
OPOC	ADC_BA+0x038	R/W	OP Offset Calculation Register	0x0000_0000
OPOC_S1	ADC_BA+0x040	R/W	OP Offset Calculation Status	0x0000_0000
TSC_SORT10	ADC_BA+0x070	R	10 th Touch Screen MAV (the minimum value)	0x0000_0000
TSC_SORT9	ADC_BA+0x074	R	9 th Touch Screen MAV	0x0000_0000
TSC_SORT8	ADC_BA+0x078	R	8 th Touch Screen MAV	0x0000_0000
TSC_SORT7	ADC_BA+0x07C	R	7 th Touch Screen MAV	0x0000_0000
TSC_SORT6	ADC_BA+0x080	R	6 th Touch Screen MAV	0x0000_0000
TSC_SORT5	ADC_BA+0x084	R	5 th Touch Screen MAV	0x0000_0000
TSC_SORT4	ADC_BA+0x088	R	4 th Touch Screen MAV	0x0000_0000
TSC_SORT3	ADC_BA+0x08C	R	3 th Touch Screen MAV	0x0000_0000
TSC_SORT2	ADC_BA+0x090	R	2 th Touch Screen MAV	0x0000_0000
TSC_SORT1	ADC_BA+0x094	R	1 th Touch Screen MAV(the max value)	0x0000_0000
TSC_MAV_X	ADC_BA+0x098	R	Touch Screen MAV X-Data	0x0000_0000
TSC_MAV_Y	ADC_BA+0x09c	R	Touch Screen MAV Y-Data	0x0000_0000

5.23.9 Register Description

ADC Control Register (ADC_CON)

Register	Address	R/W	Description	Reset Value
ADC_CON	ADC_BA+0x000	R/W	ADC control register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved			Reserved				
23	22	21	20	19	18	17	16
WT_INT_EN	Reserved	ADC_INT_EN	WT_INT	LVD_INT	ADC_INT	ADC_EN	ADC_RST
15	14	13	12	11	10	9	8
ADC_TSC_MODE		ADC_CONV	ADC_READ_CONV	ADC_MUX			AUDIODATA0 SRC
7	6	5	4	3	2	1	0
Reserved							ADC_FINISH

Bits	Descriptions	
[31:29]	RESERVED	Reserved
[28:24]	RESERVED	Reserved
[23]	WT_INT_EN	Waiting for trigger interrupt enable bit If WT_INT_EN=0, The waiting for trigger interrupt is disable If WT_INT_EN=1, The waiting for trigger interrupt is enable The WT_INT_EN bit is read/write
[22]	RESERVED	Reserved
[21]	ADC_INT_EN	ADC interrupt enable bit If ADC_INT_EN=0, The ADC interrupt is disable If ADC_INT_EN=1, The ADC interrupt is enable The ADC_INT bit is read/write
[20]	WT_INT	Waiting for trigger interrupt status bit If WT_INT=0, The waiting for trigger interrupt status is cleared If WT_INT=1, The waiting for trigger is in interrupt state The WT_INT bit is read/write and clear only, and set by hardware. Initial value after reset is unknown.
[19]	RESERVED	Reserved
[18]	ADC_INT	ADC interrupt status bit If ADC_INT=0, The ADC interrupt status is cleared If ADC_INT=1, The ADC is in interrupt state The ADC_INT bit is read/write and clear only, and set by hardware. Initial value after reset is unknown.
[17]	ADC_EN	ADC block enable bit

		If ADC_EN=0, The Analog SAR ADC block is disable If ADC_EN=1, The Analog SAR ADC block is enable The ADC_EN bit is read/write
[16]	ADC_RST	ADC reset control bit If ADC_RST=1, the ADC block is at reset mode If ADC_RST=0, the ADC block is at normal mode The ADC_RST bit is read/write
[15:14]	ADC_TSC_MODE	The touch screen conversion mode control bits If ADC_TSC_MODE=00, normal conversion mode is selected If ADC_TSC_MODE=01, semi-auto conversion mode is selected, and the ADC_CONV bit in ADC_CON register will be ignored If ADC_TSC_MODE=10, auto conversion mode is selected, and the ADC_CONV bit in ADC_CON register will be ignored If ADC_TSC_MODE=11, waiting for trigger mode is selected, and the ADC_CONV bit in ADC_CON register will be ignored The ADC_TSC_MODE bits are read/write
[13]	ADC_CONV	ADC conversion control bit If ADC_CONV=1, inform ADC to converse, when conversion finished, this bit will be auto clear. If ADC_CONV=0, the ADC no action, and this only could be cleared by hardware The ADC_CONV bit is read/write and could be set only
[12]	ADC_READ_CONV	This bit control if next conversion start after ADC_XDATA register is read in normal conversion mode. If ADC_READ_CONV=1, start next conversion after the ADC_XDATA is read, and ignore the ADC_CONV bit. If ADC_READ_CONV=0, after the ADC_XDATA is read, the ADC no action The ADC_READ_CONV bit is read/write
[11:9]	ADC_MUX	These bits select ADC input from the 8 analog inputs in normal conversion mode. ADC_MUX=000 not available in normal data conversion. It is for audio recording. ADC_MUX=001, not available in normal data conversion. It is for audio recording. ADC_MUX=010, select AIN2 ADC_MUX=011, select AIN3 ADC_MUX=100, select AIN4 ADC_MUX=101, not available in normal data conversion. It is for touch panel using ADC_MUX=110, not available in normal data conversion. It is for touch panel using ADC_MUX=111, not available in normal data conversion. The ADC_MUX bits are read/write. AIN0 and AIN1 channel are differential inputs for audio recorder only. When in Audio Recording operation (AUDIO_CON[1] / AUDIO_EN = 1), only AIN0 and AIN1 are dedicated inputs, and ADC_MUX value is not effective in this operation.
[8]	AUDIODATA0_SRC	Control AUDIO_DATA0 source from the decimation filter output or directly from ADC.

		AUDIODATA0_SRC = 0, AUDIO_DATA0 from decimation filter output AUDIODATA0_SRC = 1, AUDIO_DATA0 from ADC output
[7:1]	Reserved	Reserved
[0]	ADC_FINISH	This bit indicate the ADC is in conversion or not ADC_FINISH=0, the ADC is in conversion ADC_FINISH=1, the ADC is not in conversion The ADC_FINISH bit is read only.

Touch Screen Control Register (ADC_TSC)

Register	Address	R/W	Description	Reset Value
ADC_TSC	ADC_BA+0x004	R/W	Touch screen control register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
RESERVED						ADC_TSC_MAV_EN	ADC_TSC_XY
7	6	5	4	3	2	1	0
ADC_TSC_XP	ADC_TSC_XM	ADC_TSC_YP	ADC_TSC_YM	ADC_PU_EN	ADC_TSC_TYPE		ADC_UD

Bits	Descriptions	
[9]	ADC_TSC_MAV_EN	MAV Filter Enable/Disable for the Touch Screen AutoMode If ADC_TSC_XY = 0, MAV Filter is disable If ADC_TSC_XY = 1, MAV Filter is enable The ADC_TSC_MAV_EN bit is read/write
[8]	ADC_TSC_XY	This bit control the X-position or Y-position detection when in semi-auto conversion mode If ADC_TSC_XY = 0, X-position detection is select If ADC_TSC_XY = 1, Y-position detection is select The ADC_TSC_XY bit is read/write
[7]	ADC_TSC_XP	This bit control the interface to XP of touch screen when in normal conversion mode If ADC_TSC_XP = 0, XP is tri-state output If ADC_TSC_XP = 1, XP is connected to AVDD The ADC_TSC_XP bit is read/write
[6]	ADC_TSC_XM	This bit control the interface to XM of touch screen when in normal conversion mode If ADC_TSC_XM = 0, XM is tri-state output If ADC_TSC_XM = 1, XM is connected to AVSS The ADC_TSC_XM bit is read/write
[5]	ADC_TSC_YP	This bit control the interface to YP of touch screen when in normal conversion mode If ADC_TSC_YP = 0, YP is tri-state output If ADC_TSC_YP = 1, YP is connected to AVDD The ADC_TSC_YP bit is read/write
[4]	ADC_TSC_YM	This bit control the interface to YM of touch screen when in normal conversion mode If ADC_TSC_YM = 0, YM is tri-state output

		If ADC_TSC_YM = 1, YM is connected to AVSS The ADC_TSC_YM bit is read/write
[3]	ADC_PU_EN	This bit control the internal pull up PMOS in switch box is enable or disable If ADC_PU_EN = 0, the pull up PMOS is disable If ADC_PU_EN = 1, the pull up PMOS is enable The ADC_PU_EN bit is read/write
[2:1]	ADC_TSC_TYPE [1:0]	The touch screen type selection bits If ADC_TSC_TYPE[1:0]=00, 4-wire type is selected If ADC_TSC_TYPE[1:0]=01, 5-wire type is selected If ADC_TSC_TYPE[1:0]=10, 8-wire type is selected If ADC_TSC_TYPE[1:0]=11, unused The ADC_TSC_TYPE[1:0] bits are read/write
[0]	ADC_UD	The up down state for stylus in waiting for trigger mode If ADC_UD = 1, the stylus is in donw state If ADC_UD = 0, the stylus is in up state The ADC_UD bit is read only

ADC Delay Register (ADC_DLY)

Register	Address	R/W	Description	Reset Value
ADC_DLY	ADC_BA+0x008	R/W	ADC delay register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						ADC_DELAY	
15	14	13	12	11	10	9	8
ADC_DELAY							
7	6	5	4	3	2	1	0
ADC_DELAY							

Bits	Descriptions	
[17:0]	ADC_DELAY	Delay for Conversion. For normal conversion mode, the delay is between the ADC_CONV bit in ADC_CON register is set to the ADC begin conversion. For semi-auto conversion mode, the delay locates at each X-position and Y-position detection. For auto conversion mode, the delay locates at each position detection. The delay is defined as $ADC\_DELAY * ADC\ clock$ The ADC_DELAY[17:0] bits is read/write.

ADC X Data Buffer (ADC_XDATA)

Register	Address	R/W	Description	Reset Value
ADC_XDATA	ADC_BA+0x00C	R	ADC X data buffer	0x0000_0000

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
ADC_XDATA							

Bits	Descriptions
[9:0]	ADC Data Buffer When normal conversion mode, the conversion data is always put at this register. When semi-auto conversion mode, the conversion data of X-position detection is put at this register. When auto-conversion mode, the conversion data of X-position detection is put at this register. The ADC_XDATA[9:0] bits is read only.

ADC Y Data Buffer (ADC_YDATA)

Register	Address	R/W	Description	Reset Value
ADC_YDATA	ADC_BA+0x010	R	ADC Y data buffer	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						ADC_YDATA	
7	6	5	4	3	2	1	0
ADC_YDATA							

Bits	Descriptions	
[9:0]	ADC_YDATA	ADC Y Data Buffer. When semi-auto conversion mode, the conversion data of Y-position detection is put at this register. When auto-conversion mode, the conversion data of Y-position detection is put at this register. The ADC_YDATA[9:0] bits is read only.

Audio Control Register (AUDIO_CON)

Register	Address	R/W	Description	Reset Value
AUDIO_CON	ADC_BA+0x01C	R/W	Audio control, status and data register	0x0000_0000

31	30	29	28	27	26	25	24
AUD_INT_MODE[1:0]		AUD_INT	AUD_INTEN	VOL_EN	AUDIO_HPEN	AUDIO_EN	AUDIO_RESE
23	22	21	20	19	18	17	16
AUD_CCYCLE							
15	14	13	12	11	10	9	8
AUDIO_DATA[15:8]							
7	6	5	4	3	2	1	0
AUDIO_DATA[7:0]							

Bits	Descriptions	
[31:30]	AUD_INT_MODE[1:0]	Audio interrupt mode selection 2'b00: If AUD_INT=1, the recording for one sample is finished. 2'b01: If AUD_INT=1, the recording for two samples are finished. 2'b10: If AUD_INT=1, the recording for four samples are finished. 2'b11: If AUD_INT=1, the recording for eight samples are finished.
[29]	AUD_INT	Audio interrupt flag bits If AUD_INT=0, the recording for 1/2/4/8 samples are not finished. If AUD_INT=1, the recording for 1/2/4/8 samples are finished. The bit is readable and clear only. This flag can be set by above hardware event if AUD_INT_EN = 1. And when it is set an interrupt signal is asserted to the interrupt controller through ADC interrupt source.
[28]	AUD_INT_EN	Audio interrupt Enable 1'b0: Audio Interrupt Disable 1'b1: Audio Interrupt Enable
[27]	VOL_EN	Volume control enable bit If VOL_EN = 0, the hardware open the volume control path and open the recording path. If VOL_EN = 1, the hardware enable the volume control path and enable the recording path. The bit is R/W
[26]	AUDIO_HPEN	Record path high pass enable bit If AUDIO_HPEN = 0, the digital high pass filter will be bypassed. If AUDIO_HPEN = 1, the digital high pass filter will be enable. The bit is R/W

[25]	AUDIO_EN	Record operation enable bit If AUDIO_EN = 0, the hardware digital decimation filter will be disabled. If AUDIO_EN = 1, the hardware digital decimation filter will be enabled. The bit is R/W
[24]	AUDIO_RESET	Digital filter reset bit If AUDIO_RESET = 0, the digital filter is not reset. If AUDIO_RESET = 1, the digital filter is on the reset state. The bit is R/W
[23:16]	AUD_CYCLE[7:0] (8'h50)	Audio Conversion Cycle(Minimum Value 34) 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are unknown.
[15:0]	AUDIO_DATA[15:0] (15h'00)	Last converted audio data before AUD_INT 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are unknown.

Audio Buffer Register (AUDIO_BUF0)

Register	Address	R/W	Description	Reset Value
AUDIO_BUF0	ADC_BA+0x020	R/W	Audio data register	0x0000_0000

31	30	29	28	27	26	25	24
AUDIO_DATA1[15:8]							
23	22	21	20	19	18	17	16
AUDIO_DATA1[7:0]							
15	14	13	12	11	10	9	8
AUDIO_DATA0[15:8]							
7	6	5	4	3	2	1	0
AUDIO_DATA0[7:0]							

Bits	Descriptions	
[31:16]	AUDIO_DATA1[15:0]	Converted audio data1 at buffer0 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.
[15:0]	AUDIO_DATA0[15:0]	Converted audio data0 at buffer0 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.

Audio Control Register (AUDIO_BUF1)

Register	Address	R/W	Description	Reset Value
AUDIO_BUF1	ADC_BA+0x024	R/W	Audio data register	0x0000_0000

31	30	29	28	27	26	25	24
AUDIO_DATA3[15:8]							
23	22	21	20	19	18	17	16
AUDIO_DATA3[7:0]							
15	14	13	12	11	10	9	8
AUDIO_DATA2[15:8]							
7	6	5	4	3	2	1	0
AUDIO_DATA2[7:0]							

Bits	Descriptions	
[31:16]	AUDIO_DATA3[15:0]	Converted audio data3 at buffer1 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.
[15:0]	AUDIO_DATA2[15:0]	Converted audio data2 at buffer1 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.

Audio Control Register (AUDIO_BUF2)

Register	Address	R/W	Description	Reset Value
AUDIO_BUF2	ADC_BA+0x028	R/W	Audio data register	0x0000_0000

31	30	29	28	27	26	25	24
AUDIO_DATA5[15:8]							
23	22	21	20	19	18	17	16
AUDIO_DATA5[7:0]							
15	14	13	12	11	10	9	8
AUDIO_DATA4[15:8]							
7	6	5	4	3	2	1	0
AUDIO_DATA4[7:0]							

Bits	Descriptions	
[31:16]	AUDIO_DATA5[15:0]	Converted audio data5 at buffer2 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.
[15:0]	AUDIO_DATA4[15:0]	Converted audio data4 at buffer2 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.

Audio Control Register (AUDIO_BUF3)

Register	Address	R/W	Description	Reset Value
AUDIO_BUF3	ADC_BA+0x02C	R/W	Audio data register	0x0000_0000

31	30	29	28	27	26	25	24
AUDIO_DATA7[15:8]							
23	22	21	20	19	18	17	16
AUDIO_DATA7[7:0]							
15	14	13	12	11	10	9	8
AUDIO_DATA6[15:8]							
7	6	5	4	3	2	1	0
AUDIO_DATA6[7:0]							

Bits	Descriptions	
[31:16]	AUDIO_DATA7[15:0]	Converted audio data7 at buffer3 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.
[15:0]	AUDIO_DATA6[15:0]	Converted audio data6 at buffer3 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are 16'h0.

AGC Parameter1 Register (AGCP1)

Register	Address	R/W	Description	Reset Value
AGCP1	ADC_BA+0x030	R/W	AGC Parameter Register Setting	0x0070_B310

31	30	29	28	27	26	25	24
EDMA_MODE	Reserved						
23	22	21	20	19	18	17	16
Reserved	MAXGAIN			Reserved	MINGAIN		
15	14	13	12	11	10	9	8
OTL				UPBAND	DOWNBAND	PRAGA	
7	6	5	4	3	2	1	0
Reserved		AUDIO_VOL					

Bits	Descriptions	
[31]	EDMA_MODE	EDMA mode Enable/Dislabe 1'b0: EDMA Mode disable 1'b1: EDMA Mode Enable, when the bit is set, hardware will clear AUD_INT automatically when EDMA ACK is high
[31:23]	Reserved	Reserved
[22:20]	MAXGAIN	AGC MAXGAIN Control Register -6.75dB~35.25dB @ 6dB step size 3'b000:-6.75dB 3'b001:-0.75dB 3'b010:5.25dB 3'b011:11.25dB 3'b100:17.25dB 3'b101:23.25dB 3'b110:29.25dB 3'b111:35.25dB(default)
[19]	Reserved	Reserved
[18:16]	MINGAIN	AGC MINGAIN Control Register -12dB~30dB @ 6dB step size 3'b000:-12dB(deault) 3'b001:-6dB

		3'b010:0 dB 3'b011:6dB 3'b100:12dB 3'b101:18dB 3'b110:24dB 3'b111:30dB
[15:12]	OTL	Output Target Level (default:-12dB(4'hB)) -28.5dB~-6dB @ 1.5dB step size
[11]	UPBAND	1'b0:OTL+0.5dB (default) 1'b1:OTL+0.75dB
[10]	DOWNBAND	1'b0:OTL-0.5dB (default) 1'b1:OTL-0.75dB
[9:8]	PRAGA	Pre- Amplifer Gain Control(default:20dB(2'b11)) 2'b00:0 dB 2'b01:8 dB 2'b10:14 dB 2'b11:20 dB(default)
[7:6]	Reserved	Reserved
[5:0]	AUDIO_VOL	Audio Volume Control(default:0dB(6'h10)) -12dB~35.25dB @ 0.75dB step size

AUDIO_VOL

Digital Value	DB	Digital Value	DB	Digital Value	DB	Digital Value	DB
6'h00	-12dB	6'h10	0dB	6'h20	12dB	6'h30	24dB
6'h01	-11.25dB	6'h11	0.75dB	6'h21	12.75dB	6'h31	24.75 dB
6'h02	-10.5dB	6'h12	1.5dB	6'h22	13.5dB	6'h32	25.5 dB
6'h03	-9.75dB	6'h13	2.25dB	6'h23	14.25dB	6'h33	26.25 dB
6'h04	-9dB	6'h14	3dB	6'h24	15dB	6'h34	27 dB
6'h05	-8.25dB	6'h15	3.75dB	6'h25	15.75dB	6'h35	27.75 dB
6'h06	-7.5dB	6'h16	4.5dB	6'h26	16.5 dB	6'h36	28.5 dB
6'h07	-6.75dB	6'h17	5.25dB	6'h27	17.25 dB	6'h37	29.25 dB
6'h08	-6dB	6'h18	6dB	6'h28	18 dB	6'h38	30 dB
6'h09	-5.25dB	6'h19	6.75dB	6'h29	18.75dB	6'h39	30.75 dB
6'h0A	-4.5dB	6'h1A	7.5dB	6'h2A	19.5 dB	6'h3A	31.5 dB
6'h0B	-3.75dB	6'h1B	8.25dB	6'h2B	20.25dB	6'h3B	32.25 dB
6'h0C	-3dB	6'h1C	9dB	6'h2C	21 dB	6'h3C	33 dB
6'h0D	-2.25dB	6'h1D	9.75dB	6'h2D	21.75 dB	6'h3D	33.75 dB
6'h0E	-1.5dB	6'h1E	10.5dB	6'h2E	22.5 dB	6'h3E	34.5dB
6'h0F	-0.75dB	6'h1F	11.25dB	6'h2F	23.25dB	6'h3F	35.25dB

OTL (Output Target Leve)

4'h0	-28.5dB	15'h04CF
4'h1	-27dB	15'h05B7
4'h2	25.5dB	15'h06CB
4'h3	-24 dB	15'h0813
4'h4	-22.5 dB	15'h0999
4'h5	-21 dB	15'h0B68
4'h6	-19.5 dB	15'h0D8E
4'h7	-18 dB	15'h101D
4'h8	-16.5 dB	15'h1326
4'h9	-15 dB	15'h16C3
4'hA	-13.5 dB	15'h1B0D
4'hB	-12 dB	15'h2026
4'hC	-10.5 dB	15'h2636
4'hD	-9 dB	15'h2D6A
4'hE	-7.5 dB	15'h35FA
4'hF	-6 dB	15'h4026

Up Band Threshold+0.5dB

4'h0	-28.5dB+0.5dB	15'h0518
4'h1	-27dB+0.5dB	15'h060E
4'h2	25.5dB+0.5dB	15'h0732
4'h3	-24 dB+0.5dB	15'h088E
4'h4	-22.5 dB+0.5dB	15'h0A2A
4'h5	-21 dB+0.5dB	15'h0C15
4'h6	-19.5 dB+0.5dB	15'h0E5C
4'h7	-18 dB+0.5dB	15'h1111
4'h8	-16.5 dB+0.5dB	15'h1449
4'h9	-15 dB+0.5dB	15'h181C
4'hA	-13.5 dB+0.5dB	15'h1CA7
4'hB	-12 dB+0.5dB	15'h220E
4'hC	-10.5 dB+0.5dB	15'h287A
4'hD	-9 dB+0.5dB	15'h301B
4'hE	-7.5 dB+0.5dB	15'h392C
4'hF	-6 dB+0.5dB	15'h43F4

DownBand=Threshold-0.5Db

4'h0	-28.5dB-0.5dB	15'h048A
4'h1	-27dB-0.5dB	15'h0565
4'h2	25.5dB-0.5dB	15'h066A
4'h3	-24 dB-0.5dB	15'h079F
4'h4	-22.5 dB-0.5dB	15'h090F
4'h5	-21 dB-0.5dB	15'h0AC5
4'h6	-19.5 dB-0.5dB	15'h0CCC
4'h7	-18 dB-0.5dB	15'h0F36
4'h8	-16.5 dB-0.5dB	15'h1214
4'h9	-15 dB-0.5dB	15'h157D
4'hA	-13.5 dB-0.5dB	15'h198A
4'hB	-12 dB-0.5dB	15'h1E5A
4'hC	-10.5 dB-0.5dB	15'h2413
4'hD	-9 dB-0.5dB	15'h2AE0
4'hE	-7.5 dB-0.5dB	15'h32F5
4'hF	-6 dB-0.5dB	15'h3C90

Up_Band=Threshold+0.75dB

4'h0	-28.5dB+0.75dB	15'h053E
4'h1	-27dB+0.75dB	15'h063B
4'h2	25.5dB+0.75dB	15'h0768
4'h3	-24 dB+0.75dB	15'h08CD
4'h4	-22.5 dB+0.75dB	15'h0A76
4'h5	-21 dB+0.75dB	15'h0C6F
4'h6	-19.5 dB+0.75dB	15'h0EC7
4'h7	-18 dB+0.75dB	15'h1191
4'h8	-16.5 dB+0.75dB	15'h14E1
4'h9	-15 dB+0.75dB	15'h18D0
4'hA	-13.5 dB+0.75dB	15'h1D7E
4'hB	-12 dB+0.75dB	15'h230D
4'hC	-10.5 dB+0.75dB	15'h29A8
4'hD	-9 dB+0.75dB	15'h3183
4'hE	-7.5 dB+0.75dB	15'h3AD8
4'hF	-6 dB+0.75dB	15'h45EF

DownBand=Threshold-0.75Db

4'h0	-28.5dB	15'h0469
4'h1	-27dB	15'h053E
4'h2	25.5dB	15'h063B
4'h3	-24 dB	15'h0768
4'h4	-22.5 dB	15'h08CD
4'h5	-21 dB	15'h0A76
4'h6	-19.5 dB	15'h0C6F
4'h7	-18 dB	15'h0EC7
4'h8	-16.5 dB	15'h1191
4'h9	-15 dB	15'h14E1
4'hA	-13.5 dB	15'h18D0
4'hB	-12 dB	15'h1D7E
4'hC	-10.5 dB	15'h230D
4'hD	-9 dB	15'h29A8
4'hE	-7.5 dB	15'h3183
4'hF	-6 dB	15'h3AD8

AGC Control Register (AGC_CON)

Register	Address	R/W	Description	Reset Value
AGC_CON	ADC_BA+0x034	R/W	AGC Control Register	0x0000_0000

31	30	29	28	27	26	25	24
NG_EN	AGC_EN	PAVG_MODE		Reserved		PERIOD[9:8]	
23	22	21	20	19	18	17	16
PERIOD[7:0]							
15	14	13	12	11	10	9	8
AGAIN_STEP	Reserved	NG_LEVEL[2:0]		ATTACK			
7	6	5	4	3	2	1	0
RECOVERY				HOLD			

Bits	Descriptions	
[31]	NG_EN	Noise gate enable bit - If NG_EN = 0, the hardware noise gate function is disable. - If NG_EN = 1, the hardware noise gate function is enable. When the noise gate function is enabled, the input amplitude less than the define in the register NG_LEVEL[1:0] will force the gain to 0dB to prevent AGC enlarging the volume and enlarging the noise. The NG_EN bit is R/W
[30]	AGC_EN	Auto gain control enable bit - If AGC_EN = 0, the hardware AGC function is disable. When the AGC function is disabled, the input volume is controlled by register AUDIO_VOL. - If AGC_EN = 1, the hardware AGC function is enable. When the AGC function is enabled, the input volume is controlled by AGC not by register AUDIO_VOL. The AGC_EN bit is R/W
[29:28]	PAVG_MODE	Peak Average Mode 2'b00: To use the Max sample value as max-peak during detection period 2'b01: To use the (1st Max + 2nd Max)/2 as max-peak during detection period 2'b10: To use the (4 Max Value)/4 as max-peak during detection period 2'b11: To use the (8 Max Value)/8 as max-peak during detection period
[27:26]	Reserved	Reserved
[25:16]	PERIOD[9:0]	Maximum Peak Detector Calculation Period: PERIOD[6:0] * 0.125ms (0.125ms ~ 16 ms, when sampling rate is 8KHz)
[15]	AGAIN_STEP	Attack Gain Step 1'b0: 0.75dB/per step 1'b1: 1.5dB/per step

[14]	Reserved	Reserved	
[13:12]	NG_LEVEL	Noise Gate Threshold:	
		NG_LEVEL[1:0]	Noise Gate Threshold
		2'b00	-30 dB
		2'b01	-36 dB
		2'b10	-42 dB
		2'b11	-48 dB
[11:8]	ATTACK	AGC attack (gain ramp-down) time: (Base on the condition: PERIOD = 10 ms) 10ms ~ 160ms / Per step (from value 0 to 15)	
[7:4]	RECOVERY	AGC recovery (gain ramp-up) time: (Base on the condition: PERIOD = 10 ms) 10ms ~ 160ms / Per step (from value 0 to 15) 60ms ~ 960ms / Per 6dB (from value 0 to 15)	
[3:0]	HOLD	AGC hold time before gain is increased: (Base on the condition: PERIOD = 10 ms) 0 ms ~ 150ms (from value 0 to 15)	

OP Offset Calculation Register (OPOC)

Register	Address	R/W	Description	Reset Value
OPOC	ADC_BA+0x038	R/W	AGC Parameter Register Setting	0x0000_0000

31	30	29	28	27	26	25	24
MUTE_SW	OOC	OPOCM	OPOC_SW				
23	22	21	20	19	18	17	16
Reserved						OPOCTCSN	
15	14	13	12	11	10	9	8
OPOC_DSC[15:0]							
7	6	5	4	3	2	1	0
OPOC_DSC[7:0]							

Bits	Descriptions	
[31]	MUTE_SW	Mute control under software mode 1'b0: mute disable 1'b1: mute enable When mute enable the AUDIO_VOL will force to 6'h10(0dB)
[30]	Reserved	Reserved
[29]	OPOCM	OP Offset Cancellation Method 1'b0: Software setting mode 1'b1: Hardware auto calculation
[28:24]	OPOC_SW	OP Offset Cancellation 32mV~-32mV @ -2mV step size 5'h00:32mV 5'h01:30mV . 5'h10:0mV. . 5'h1e:-28mV 5'h1f:-30mV Note: The register is used on the Software setting mode
[23:18]	Reserved	Reserved
[17:16]	OPOC_TCSN	OP Offset Total calculation sample number 2'b00: 128 samples

		2'b01: 255 samples 2'b10: 512 samples 2'b11: 1024 samples Note: The control register is used under the hardware auto calculation mode
[15:0]	OPOC_DSC	OP Offset Calculation Delay Sample Count The delay sample count after mute action, and then to begin calculate. Note: The control register is used under the hardware auto calculation mode

OP Offset Calculation Status Register 1 (OPOCS1)

Register	Address	R/W	Description	Reset Value
OPOCS1	ADC_BA+0x040	R/W	OP Offset Calculation Status	0x0000_0000

31	30	29	28	27	26	25	24
		OP_OFFSET_CAL[9:4]					
23	22	21	20	19	18	17	16
	OP_OFFSET_CAL[3:0]			ADC_DATA_SUM[19:16]			
15	14	13	12	11	10	9	8
ADC_DATA_SUM[15:8]							
7	6	5	4	3	2	1	0
ADC_DATA_SUM[7:0]							

Bits	Descriptions	
[29:20]	OP_OFFSET_CAL	The OP Offset value after hardware calculation(read only)
[19:0]	ADC_DATA_SUM	ADC Data Sum(read only) The ADC data summation after calculation.

10th Touch Screen MAV (TSC_SORT10)

Register	Address	R/W	Description	Reset Value
TSC_SORT10	ADC_BA+0x070	R	10 th Touch Screen MAV Sorting Value	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV10[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV10[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV10[9:8]	
7	6	5	4	3	2	1	0
X_MAV10[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV10	10 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV10	10 th MAV X-Data

9th Touch Screen MAV SORT (TSC_SORT9)

Register	Address	R/W	Description	Reset Value
TSC_SORT9	ADC_BA+0x074	R	9 th Touch Screen MAV Sort Value	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV9[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV9[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV9[9:8]	
7	6	5	4	3	2	1	0
X_MAV9[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV9	9 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV9	9 th MAV X-Data

8th Touch Screen MAV SORT (TSC_SORT8)

Register	Address	R/W	Description	Reset Value
TSC_SORT10	ADC_BA+0x078	R	8 th Touch Screen MAV Sort Value	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV8[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV8[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV8[9:8]	
7	6	5	4	3	2	1	0
X_MAV8[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV8	8 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV8	8 th MAV X-Data

7th Touch Screen MAV (TSC_Sort7)

Register	Address	R/W	Description	Reset Value
TSC_SORT7	ADC_BA+0x07C	R	7 th Touch Screen MAV Sort Value	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV7[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV7[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV7[9:8]	
7	6	5	4	3	2	1	0
X_MAV7[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV7	7 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV7	7 th MAV X-Data

6th Touch Screen MAV (TSC_SORT6)

Register	Address	R/W	Description	Reset Value
TSC_Sort6	ADC_BA+0x080	R	6 th Touch Screen MAV	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						Y_MAV6[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV6[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV6[9:8]	
7	6	5	4	3	2	1	0
X_MAV6[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV6	6 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV6	6 th MAV X-Data

5th Touch Screen MAV (TSC_Sort5)

Register	Address	R/W	Description	Reset Value
TSC_SORT5	ADC_BA+0x084	R	5 th Touch Screen MAV	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV5[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV5[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV5[9:8]	
7	6	5	4	3	2	1	0
X_MAV5[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV5	5 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV5	5 th MAV X-Data

4th Touch Screen MAV (TSC_SORT4)

Register	Address	R/W	Description	Reset Value
TSC_SORT4	ADC_BA+0x088	R	4 th Touch Screen MAV	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV4[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV4[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV4[9:8]	
7	6	5	4	3	2	1	0
X_MAV4[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV4	4 th MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV4	4 th MAV X-Data

3rd Touch Screen MAV (TSC_Sort3)

Register	Address	R/W	Description	Reset Value
TSC_Sort3	ADC_BA+0x08C	R	3 rd Touch Screen MAV	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV3[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV3[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV3[9:8]	
7	6	5	4	3	2	1	0
X_MAV3[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV3	3 rd MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV3	3 rd MAV X-Data

2nd Touch Screen MAV (TSC_Sort2)

Register	Address	R/W	Description	Reset Value
TSC_Sort2	ADC_BA+0x090	R	2 nd Touch Screen MAV	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV2[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV2[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV2[9:8]	
7	6	5	4	3	2	1	0
X_MAV2[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV2	2 nd MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV2	2 nd MAV X-Data

1st Touch Screen MAV (TSC_Sort1)

Register	Address	R/W	Description	Reset Value
TSC_Sort1	ADC_BA+0x094	R	2 nd Touch Screen MAV	0x0000_0000

31	30	29	28	27	26	25	24
		Reserved				Y_MAV1[9:8]	
23	22	21	20	19	18	17	16
	Y_MAV1[7:0]						
15	14	13	12	11	10	9	8
Reserved						X_MAV1[9:8]	
7	6	5	4	3	2	1	0
X_MAV1[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	Y_MAV1	1 st MAV Y-Data
[15:10]	Reserved	Reserved
[9:0]	X_MAV1	1 st MAV X-Data

1st Touch Screen MAV (TSC_Sort1)

Register	Address	R/W	Description	Reset Value
TSC_MAV_X	ADC_BA+0x098	R	Touch Screen MAV X-Data	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						X_MAV1[9:8]	
7	6	5	4	3	2	1	0
X_MAV1[7:0]							

Bits	Descriptions	
[31:10]	Reserved	Reserved
[9:0]	X_MAV1	1 st MAV X-Data

5.24 Keypad Interface (KPI)

5.24.1 Overview

The Keypad Interface (**KPI**) is an APB slave with configurable minimum 2-row up to 8-row scan output and minimum 2-column up to 8-column scan input. Any keys in the array pressed or released are de-bounced and generate an interrupt.

The KPI supports release multiple keys, press multiple keys scan interrupt and specified INT_3KEYs interrupt for chip reset. If the 3 pressed keys matches with the 3 keys defined in **KPI3KCONF**, it will generate an interrupt and chip reset (**ENRST** must setting) depend on the **ENRST** setting. The interrupt is generated whenever it detects any key in the keypad pressing or releasing or waking up from IDLE or three-key reset. User can know the interrupt source by querying KPISTATUS register

5.24.2 Features:

- Matrix keypad interface (maximum 8x8 array, and minimum 2x2array)
- Programmable de-bounce time
- Low-power wakeup mode
- Programmable three-key reset
- Generate interrupt and update all the keys(maximum 64 keys, minimum 4 keys) information(press/release) every time the user pressing or releasing

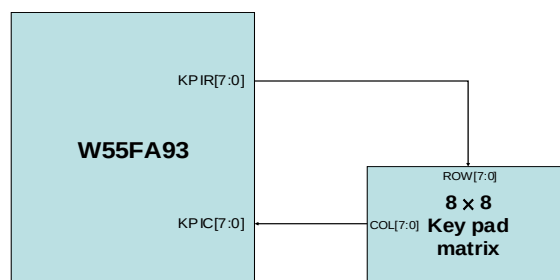


Figure 5.24-1 Keypad Interface

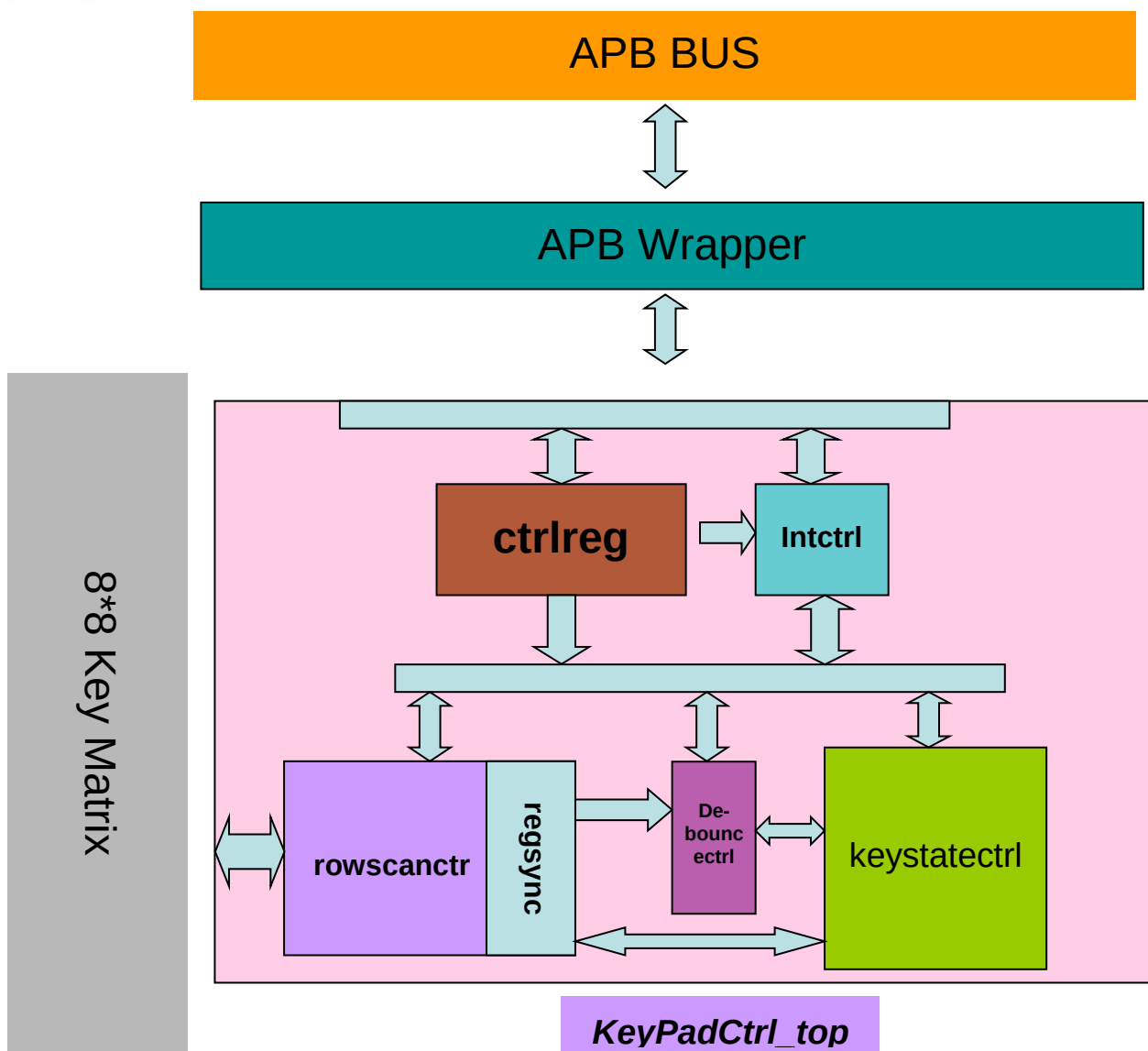


Figure 5.24-2 Keypad architecture

5.24.3 KPI Block Diagram

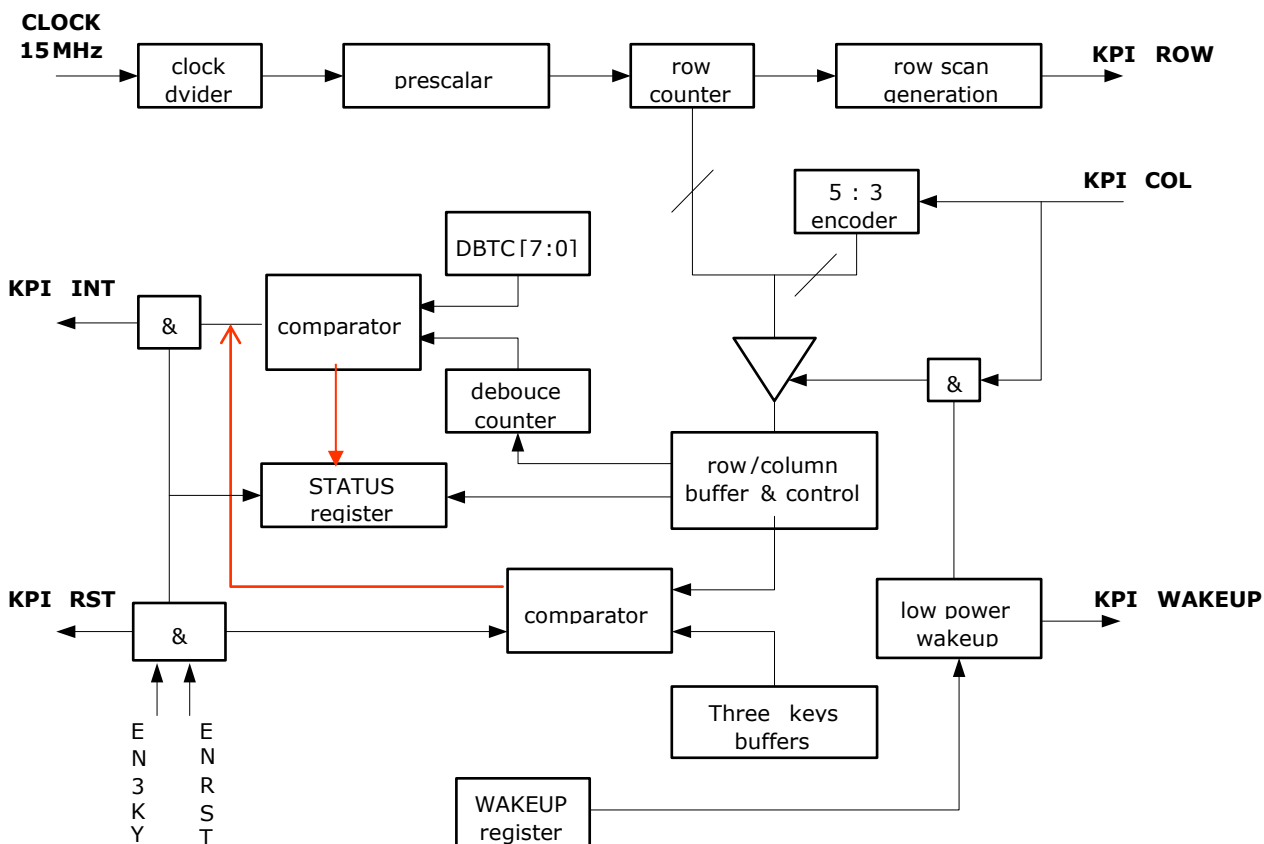


Figure 5.24-3 Keypad Controller Block Diagram

5.24.4 Keypad Interface Register Map

Register	Address	R/W	Description	Reset Value
KPI_BA = 0xB800_5000				
KPICNF	KPI_BA+0x000	R/W	Keypad configuration Register	0x0000_0000
KPI3KCNF	KPI_BA+0x004	R/W	Keypad 3-keys configuration register	0x0000_0000
KPISTATUS	KPI_BA+0x008	R/O	Keypad status register	0x0000_0000
KPIRSTC	KPI_BA+0x00C	R/O	Keypad reset period controller register	0x0000_0000
KPIKEST0	KPI_BA+0x010	R/O	Keypad state register 0	0xFFFF_FFFF
KPIKEST1	KPI_BA+0x014	R/O	Keypad state register 1	0xFFFF_FFFF
KPIKPE0	KPI_BA+0x018	R/O	Lower 32 Press Key event indicator	0x0000_0000
KPIKPE1	KPI_BA+0x01C	R/O	Higher 32 Press Key event indicator	0x0000_0000
KPIKRE0	KPI_BA+0x020	R/O	Lower 32 Release Key event indicator	0x0000_0000
KPIKRE1	KPI_BA+0x024	R/O	Higher 32 Release Key event indicator	0x0000_0000
KPIPRESCALDIV	KPI_BA+0x028	R/W	Pre-scale divider	0x0000_001F

Keypad Controller Configuration Register (KPI_CONF)

Register	Address	R/W	Description	Reset Value
KPICONF	KPI_BA+0x000	R/W	Keypad configuration register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	KROW[2:0]			Reserved	KCOL[2:0]		
23	22	21	20	19	18	17	16
		DB_EN	Reserved	DBCLKSEL			
15	14	13	12	11	10	9	8
PRESCALE[7:0]							
7	6	5	4	3	2	1	0
Reserved	INPU	WAKEUP	ODEN	INTEN	RKINTEN	PKINTEN	ENKP

Bits	Descriptions																	
[31]	Reserved	Reserved																
[30:28]	KROW	Keypad Matrix ROW number The keypad matrix is set by ROW x COL. The ROW number can be set 2, 3, 4, 5, 6, 7, or 8																
		<table><tr><th>KROW[30:28]</th><th>Keypad matrix ROW number</th></tr><tr><td>001</td><td>2</td></tr><tr><td>010</td><td>3</td></tr><tr><td>011</td><td>4</td></tr><tr><td>100</td><td>5</td></tr><tr><td>101</td><td>6</td></tr><tr><td>110</td><td>7</td></tr><tr><td>000</td><td>8 (Default)</td></tr></table>	KROW[30:28]	Keypad matrix ROW number	001	2	010	3	011	4	100	5	101	6	110	7	000	8 (Default)
		KROW[30:28]	Keypad matrix ROW number															
		001	2															
		010	3															
		011	4															
		100	5															
		101	6															
		110	7															
000	8 (Default)																	
[27]	Reserved	Reserved																

[26:24]	KCOL	Keypad Matrix COL Number The keypad matrix is set by ROW x COL. The COL number can be set 2, 3, 4, 5, 6, 7, or 8	
		KCOL[26:24]	Keypad matrix COLUMN number
		001	2
		010	3
		011	4
		100	5
		101	6
		110	7
		000	8 (Default)
[21]	DB_EN	Scan In Signal De-bounce Enable 0 = The de-bounce function is disabled 1 = The de-bounce function is enabled	
[20]	Reserved	Reserved	

[19:16]	DBCLKSEL	Scan In De-bounce sampling cycle selection	
		DBCLKSEL	Description
		0	Sample interrupt input once per 1 clocks
		1	Sample interrupt input once per 2 clocks
		2	Sample interrupt input once per 4 clocks
		3	Sample interrupt input once per 8 clocks
		4	Sample interrupt input once per 16 clocks
		5	Sample interrupt input once per 32 clocks
		6	Sample interrupt input once per 64 clocks
		7	Sample interrupt input once per 128 clocks
		8	Sample interrupt input once per 256 clocks
		9	Sample interrupt input once per 2*256 clocks
		10	Sample interrupt input once per 4*256clocks
		11	Sample interrupt input once per 8*256 clocks
		12	Sample interrupt input once per 16*256 clocks
		13	Sample interrupt input once per 32*256 clocks
		14	Reserved
		15	Reserved
suggestion: row scan time >= 2 * debounce sampling cycle row scan time = prescale * 32 (xclock) xclock = 1MHz ~32KHz bouncing time last for 1ms, for example,if xclock = 1MHz debounce sampling cycle should choose 1024 xclock row scan time should chose 2048 xclock,suppose PrescaleDivider = 0x1F,then prescale =2048/32 = 64			
[15:8]	PRESCALE	Row Scan Cycle Pre-scale Value This value is used to pre-scale row scan cycle. The pre-scale counter is clocked by the divided crystal clock, xCLOCK. The PRESCALE value is from 1 to 256. Eg.If the crystal clock is 1Mhz then the xCLOCK period is 1us. If the keypad matric is 3x3 then Each row scan time = xCLOCK x PRESCALE x PrescaleDivider Key array scan time = Each row scan time x ROWS Example scan time for PRESCALE = 0x41,and PrescaleDivider = 0x1F Each row scan time = 1us x 65 x 32 = 2.08ms Scan time = 2.08 x 3 = 6.24ms Notes: When PRESCALE is determined, De-bounce sampling cycle should not exceed the half of (PRESCALE x PrescaleDivider),in the above example The maximum DBCLKSEL should be 4*256 xCLOCK ,bouncing time is 1ms	
[7]	Reserved	Reserved	

[6]	INPU	key Scan In Pull-UP Enable Register 1 = A pull-up resistor is connected to the respective keypad pin 0 = The respective keypad pin pull-up resistor is disconnected
[5]	WAKEUP	Lower Power Wakeup Enable Setting this bit enables low power wakeup 1 = Wakeup enable 0 = Not enable
[4]	ODEN	Open Drain Enable If there are more than one key are pressed in the same column, then "short-circuit" will appear between active and inactive scan row. Software can set this bit HIGH to enable scan output KPI_ROW[4:0] pins work as "open-drain" to avoid the "short-circuit" 0 = Push-Pull drive 1 = Open drain
[3]	INTEN	Key Interrupt Enable Control 0 = disable the keypad interrupt 1 = enable the keypad interrupt
[2]	RKINTEN	Release Key Interrupt Enable Control The keypad controller will generate an interrupt when the controller detects keypad status changes from press to release. 0 = disable the keypad release interrupt 1 = enable the keypad release interrupt
[1]	PKINTEN	Press Key Interrupt Enable Control The keypad controller will generate an interrupt when the controller detects any effective key press 0 = disable the keypad press interrupt 1 = enable the keypad press interrupt
[0]	ENKP	Keypad Scan Enable Setting this bit high enable the key scan function. 1 = Enable keypad scan 0 = Disable keypad scan

Keypad Controller 3-keys Configuration Register (KPI3KCONF)

Register	Address	R/W	Description	Reset Value
KPI3KCONF	KPI_BA+0x004	W/R	Keypad 3-keys configuration register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							EN3KYRST
23	22	21	20	19	18	17	16
Reserved		K32R			K32C		
15	14	13	12	11	10	9	8
Reserved		K31R			K31C		
7	6	5	4	3	2	1	0
Reserved		K30R			K30C		

Bits	Descriptions	
[31:25]	Reserved	Reserved
[24]	EN3KYRST	Enable Three-key Reset Setting this bit enable hardware reset when three-key is detected 1: Three-key function is enable 0: Three-key function is disable
[23:22]	Reserved	Reserved
[21:19]	K32R	The #2 Key Row Address The #2 means the row address and the column address is the highest of the specified 3-keys
[18:16]	K32C	The #2 Key Column Address
[15:14]	Reserved	Reserved
[13:11]	K31R	The #1 Key Row Address The #1 means the row address and the column address is the 2nd of the specified 3-keys
[10:8]	K31C	The #1 Key Column Address
[7:6]	Reserved	Reserved
[5:3]	K30R	The #0 Key Row Address The #0 means the row address and the column address is the lowest of the specified 3-keys
[2:0]	K30C	The #0 Key Column Address

Key Pad Interface Status Register (KPISTATUS)

Register	Address	R/W	Description	Reset Value
KPISTATUS	KPI_BA+0x008	R/O	Keypad status register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
RROW7	RROW6	RROW5	RROW4	RROW3	RROW2	RROW1	RROW0
15	14	13	12	11	10	9	8
PROW7	PROW6	PROW5	PROW4	PROW3	PROW2	PROW1	PROW0
7	6	5	4	3	2	1	0
Reserved		PKEY_INT	RKEY_INT	KEY_INT	RST_3KEY		PDWAKE

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	RELEASE ROW MAP [n]	Release key row coordinate Show the row of the release key, for example if key (2,x) was released ,x = 0~7,then RROW will show 0000_0100
[15:8]	PRESS ROW MAP [n]	Press key row coordinate Show the row of the press key, for example if key (2,x) was pressed ,x = 0~7,then PROW will show 0000_0100
[4]	PKEY_INT	Press key interrupt This bit indicates that some keys (one or multiple key) were pressed Read 1 : At least one key press 0 = no key press Notes: In order to clear PKEY_INT, software must clear each pressing event that are shown on "KPIKPE1, KPIKPE0". C code example: DWORD PKE0, PKE1; PKE0 = reg_read(KPIKPE0); PKE1 = reg_read(KPIKPE1); Reg_write(KPIKPE0,PKE0); Reg_write(KPIKPE1,PKE1);

[3]	RKEY_INT	Release key interrupt This bit indicates that some keys (one or multiple key) were released Read 1 = At least one key release 0 = no key release Notes: In order to clear RKEY_INT, software must clear each releasing event that are shown on "key releasing event". C code example: <pre>DWORD RKE0, RKE1; PKE0 = reg_read(KPIKRE0); PKE1 = reg_read(KPIKRE1); Reg_write(KPIKRE0,RKE0); Reg_write(KPIKRE1,RKE1);</pre>
[2]	KEY_INT	Key Interrupt This bit indicates the key scan interrupt is active when any key press or release or three key reset or wakeup Read 1 = key press/release/3-key reset/wakeup interrupt occur 0 = Not reset
[1]	RST_3KEY	3-Keys Reset Flag This bit will be set after 3-keys reset occur. Read 1 = 3 keys reset interrupt occur 0 = Not reset Write 1 = clear interrupt flag 0 = no operation
[0]	PDWAKE	Power Down Wakeup Flag This flag indicates the chip is wakeup from power down by keypad Read 1 = Wakeup up by keypad 0 = Not wakeup Write 1 = clear interrupt flag 0 = no operation

Keypad Reset Period Controller Register (KPIRSTC)

Register	Address	R/W	Description	Reset Value
KPIRSTC	KPI_BA+0x00C	R/O	Keypad Reset Period Control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
RSTC							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	RSTC	3-key Reset Period Count The keypad controller generates a reset signal when it detects 3-key match condition, if the ENRST is set. The RSTC is used to control the reset period. Reset period = 64 * RSTC XCLOCK

Keypad KEY STATE 0 (KPIKEST0)

Register	Address	R/W	Description	Reset Value
KPIKEST0	KPI_BA+0x010	R/O	Key state register 0	0xFFFF_FFFF

31	30	29	28	27	26	25	24
KEST37	KEST36	KEST35	KEST34	KEST33	KEST32	KEST31	KEST30
23	22	21	20	19	18	17	16
KEST27	KEST26	KEST25	KEST24	KEST23	KEST22	KEST21	KEST20
15	14	13	12	11	10	9	8
KEST17	KEST16	KEST15	KEST14	KEST13	KEST12	KEST11	KEST10
7	6	5	4	3	2	1	0
KEST07	KEST06	KEST05	KEST04	KEST03	KEST02	KEST01	KEST00

Bits	Descriptions	
[31:0]	Key state	Key State KEST m,n : m is row number, n is column number 0:Key m,n is pressing 1:key m,n is releasing

Keypad KEY STATE 1 (KPIKEST1)

Register	Address	R/W	Description	Reset Value
KPIKEST1	KPI_BA+0x014	R/O	Key state register 1	0xFFFF_FFFF

31	30	29	28	27	26	25	24
KEST77	KEST76	KEST75	KEST74	KEST73	KEST72	KEST71	KEST70
23	22	21	20	19	18	17	16
KEST67	KEST66	KEST65	KEST64	KEST63	KEST62	KEST61	KEST60
15	14	13	12	11	10	9	8
KEST57	KEST56	KEST55	KEST54	KEST53	KEST52	KEST51	KEST50
7	6	5	4	3	2	1	0
KEST47	KEST46	KEST45	KEST44	KEST43	KEST42	KEST41	KEST40

Bits	Descriptions	
[31:0]	Key state	KESTm,n: m is row number, n is column number 0:Keym,n is pressing 1:keym,n is releasing

KPIKPE0 (KPIKPE0)

Register	Address	R/W	Description	Reset Value
KPIKPE0	KPI_BA+0x018	R/W	Lower 32 Key press event indicator	0x0000_0000

31	30	29	28	27	26	25	24
KPE37	KPE36	KPE35	KPE34	KPE33	KPE32	KPE31	KPE30
23	22	21	20	19	18	17	16
KPE27	KPE26	KPE25	KPE24	KPE23	KPE22	KPE21	KPE20
15	14	13	12	11	10	9	8
KPE17	KPE16	KPE15	KPE14	KPE13	KPE12	KPE11	KPE10
7	6	5	4	3	2	1	0
KPE07	KPE06	KPE05	KPE04	KPE03	KPE02	KPE01	KPE00

Bits	Descriptions	
[X]	KPEmn[X]	Lower 32 key Press event change indicator ,m=row ,n=column KPE mn[X] = 1: corresponding key have a high to low event change Note: Hardware will set this bit, software should clear this bit by writing 1 Notes: software can clear PKEY_INT (KPISTATUS[4]) by writing 1 bit by bit to this register

KPIKPE1 (KPIKPE1)

Register	Address	R/W	Description	Reset Value
KPIKPE1	KPI_BA+0x01C	R/O	Upper 32 Key press event indicator	0x0000_0000

31	30	29	28	27	26	25	24
KPE77	KPE76	KPE75	KPE74	KPE73	KPE72	KPE71	KPE70
23	22	21	20	19	18	17	16
KPE67	KPE66	KPE65	KPE64	KPE63	KPE62	KPE61	KPE60
15	14	13	12	11	10	9	8
KPE57	KPE56	KPE55	KPE54	KPE53	KPE52	KPE51	KPE50
7	6	5	4	3	2	1	0
KPE47	KPE46	KPE45	KPE44	KPE43	KPE42	KPE41	KPE40

Bits	Descriptions	
[X]	KEYmn[X]	Upper 32 key Press key indicator, m=row, n=column KPEmn [X] = 1: corresponding key have a high to low event change Note: Hardware will set this bit, software should clear this bit by writing 1 Notes: software can clear PKEY_INT (KPISTATUS[4]) by writing 1 bit by bit to this register

KPIKRE0 (KPIKRE0)

Register	Address	R/W	Description	Reset Value
KPIKRE0	KPI_BA+0x020	R/O	Lower 32 Key release event indicator	0x0000_0000

31	30	29	28	27	26	25	24
KRE37	KRE36	KRE35	KRE34	KRE33	KRE32	KRE31	KRE30
23	22	21	20	19	18	17	16
KRE27	KRE26	KRE25	KRE24	KRE23	KRE22	KRE21	KRE20
15	14	13	12	11	10	9	8
KRE17	KRE16	KRE15	KRE14	KRE13	KRE12	KRE11	KRE10
7	6	5	4	3	2	1	0
KRE07	KRE06	KRE05	KRE04	KRE03	KRE02	KRE01	KRE00

Bits	Descriptions	
[X]	KREmn[X]	Lower 32 key release event indicator, m=row, n=column KREmn[X] = 1: corresponding key has a low to high event change Note: Hardware will set this bit, software should clear this bit by writing 1 Notes: software can clear RKEY_INT (KPISTATUS[3]) by writing 1 bit by bit to this register

KPIKRE1 (KPIKRE1)

Register	Address	R/W	Description	Reset Value
KPIKRE1	KPI_BA+0x024	R/O	Upper 32 Key release indicator	0x0000_0000

31	30	29	28	27	26	25	24
KRE77	KRE76	KRE75	KRE74	KRE73	KRE72	KRE71	KRE70
23	22	21	20	19	18	17	16
KRE67	KRE66	KRE65	KRE64	KRE63	KRE62	KRE61	KRE60
15	14	13	12	11	10	9	8
KRE57	KRE56	KRE55	KRE54	KRE53	KRE52	KRE51	KRE50
7	6	5	4	3	2	1	0
KRE47	KRE46	KRE45	KRE44	KRE43	KRE42	KRE41	KRE40

Bits	Descriptions	
[X]	KREmn[X]	Upper 32 key releasing key indicator, m=row, n=column KREmn[X] = 1: corresponding key has a low to high event change Note: Notes: software can clear RKEY_INT (KPISTATUS[3]) by writing 1 bit by bit to this register

Pre-scaleDivider (KPIPRESCALDIV)

Register	Address	R/W	Description	Reset Value
KPIPRESCALDIV	KPI_BA+0x028	R/W	Pre-scale divider	0x0000_001F

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Pre-scale divider[7:0]							

Bits	Descriptions	
[7:0]	Prescale divider	Divide Prescaler This value is used to divide RESCALE that is set in KPI_CONF[15:8]. The Prescale divider counter is clocked by the divided crystal clock, xCLOCK. The pre-scale divider number is from 1 to 256. Eg.If the crystal clock is 1Mhz then the xCLOCK period is 1us. If the keypad matrix is 3x3 then Each row scan time = xCLOCK x PRESCALE x PrescaleDivider Key array scan time = Each row scan time x ROWS Example scan time for PRESCALE = 0x41,and PrescaleDivider = 0x1F Each row scan time = 1us x 65 x 32 = 2.08ms Scan time = 2.08 x 3 = 6.24ms Notes: When PRESCALE is determined, De-bounce sampling cycle should not exceed the half of (PRESCALE x PrescaleDivider),in the above example The maximum DBCLKSEL should be 4*256 xclock ,bouncing time is 1ms

5.24.5 Timing Diagram

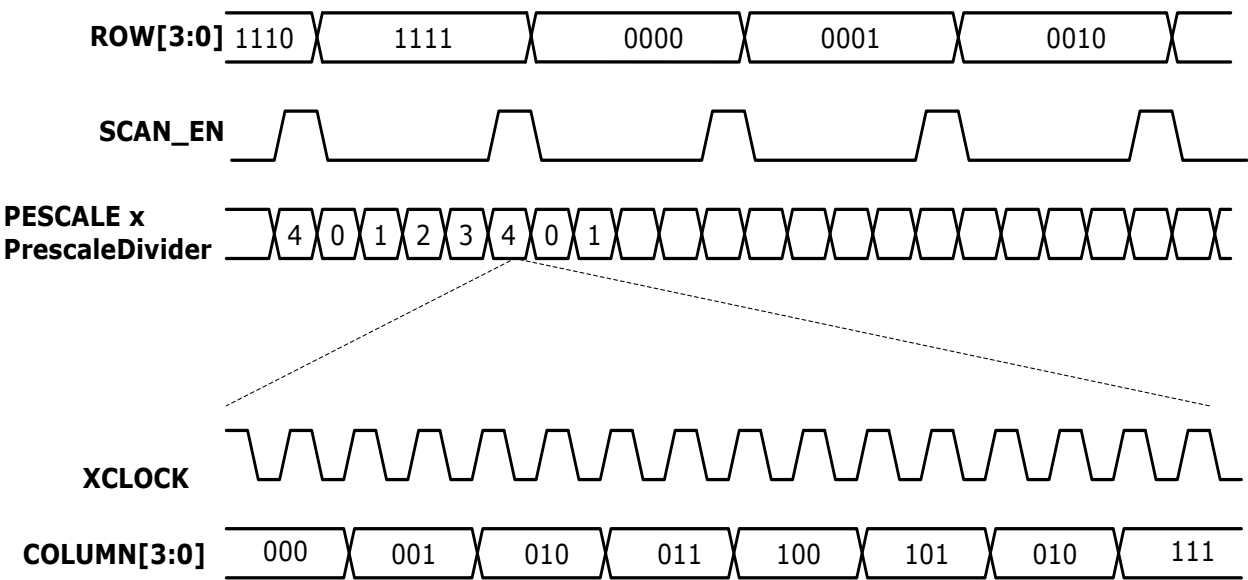


Figure 5.24-4 16x8 Keypad Scan Timing Diagram

6 ELECTRICAL CHARACTERISTICS

6.1 Absolute Maximum Ratings

Parameters	Values
Ambient Temperature	-20 °C ~ 85 °C
Storage Temperature	-40 °C ~ 125 °C
Voltage On Any Pin	-0.3V ~ 3.6V
Power Supply Voltage (Core Logic)	-0.5V ~ 1.5V
Power Supply Voltage (I/O Buffer)	-0.5V ~ 4.6V
Injection Current (Latch-Up Testing)	100mA
Crystal Frequency	1MHz ~ 27MHz

Note: Exposure to conditions beyond those listed under absolute maximum ratings may adversely affects the life and reliability of the device.

6.2 DC Electrical Characteristics

6.2.1 N9H20 Series DC Electrical Characteristics

(VDD-VSS=3.3 V, TA = 25°C, FOSC = 12 MHz unless otherwise specified.)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD33	I/O Buffer Post-Driver Voltage		3.0	3.3	3.6	V
MVDD33	SDRAM Operation Voltage (for N9H2011N)	100MHz	3.0	3.3	3.6	V
MVDD18	DDR Operation Voltage (for N9H2031N or N9H2051N)	100MHz	1.7	1.8	1.9	
VDD18	Core Logic and I/O Buffer Pre-Driver Voltage	200MHz	1.7	1.8	1.9	V
RTC_VDD	RTC Power Supply		1.2	-	1.8	V
I _{RTC_VDD}	RTC Supply Current	RTC_VDD<VDD18	-	4	-	uA
V _{IH}	Input High Voltage		2.0	-	5.5	V
V _{IL}	Input Low Voltage		-0.3	-	0.8	V
V _T	Threshold Point		1.45	1.58	1.74	V
V _{T+}	Schmitt Trigger Low to High Threshold Point		1.44	1.50	1.56	V
V _{T-}	Schmitt Trigger High to Low Threshold Point		0.89	0.94	0.99	V
I _{CC}	Core Power Supply Current	F _{CPU} = 200MHz, MCLK = 100MHz, VDD18 = 1.8V	-	160	-	mA
I _L	Input Leakage Current		-10	-	10	uA
I _{OZ}	Tri-State Output Leakage Current		-10	-	10	uA
R _{PU}	Pull-Up Resistor		39	65	116	kΩ
R _{PD}	Pull-Down Resistor		40	56	108	kΩ
V _{OL}	Output Low Voltage		-	-	0.4	V
V _{OH}	Output High Voltage		2.4	-	-	V
I _{OL}	Low Level Output Current	4mA I/O V _{OL} = 0.4V	-	4.0	-	mA
I _{OH}	High Level Output Current	4mA I/O V _{OH} = 2.4V	-	5.9	-	mA

6.2.2 ADC Characteristics

Parameter	Min.	Typ.	Max.	Unit
SAR ADC Input Voltage Range	3.0	-	3.6	V

Parameter	Min.	Typ.	Max.	Unit
Resolution of ADC	-	-	10	bit
Signal-to-Noise Plus Distortion of ADC from Line In	-	TBD	-	dB
Integral Non-Linearity of ADC	-	±2.0	-	LSB
Differential Non-Linearity of ADC	-	±0.8	-	LSB
No Missing Code	-	10	-	bit
AD Conversion Rate=ADCCLK/16	-	-	400	KHz

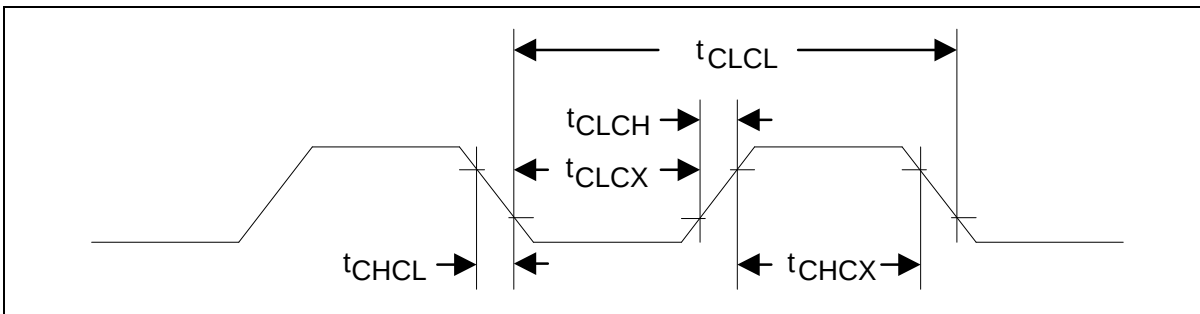
6.2.3 Audio DAC Characteristics

Parameter	Min	Typ	Max	Unit
Operating Voltage	3.0	3.3	3.6	V
Reference Voltage	-	DAC_VDD/2	-	V
Reference Capacitor	-	0.1	-	uF
Full Scale output voltage	-	0.74	-	Vrms
Full Scale output voltage	-	2.08	-	Vpp
Maximum Output Power	-	-	52	mW
Maximum Output Power @ 32ohm load	-	-	46	mW
Maximum Output Power @ 16ohm load	-	-	41	mW
L-Channel SNR	-	86	-	dBV
R-Channel SNR	-	85	-	dBV
L-Channel THD+N	-	-64	-	dB
R-Channel THD+N	-	-64	-	dB
L-Channel THD+N @ 32ohm load	-	-63	-	dB
R-Channel THD+N @ 32ohm load	-	-63	-	dB
L-Channel THD+N @ 16ohm load	-	-62	-	dB
R-Channel THD+N @ 16ohm load	-	-62	-	dB

Test conditions: RL = 10K / 50pF, BW = 20Hz ~ 20KHz, Freq.= 1KHz, Sample Rate = 48KHz.

6.3 AC Electrical Characteristics

6.3.1 External 12 MHz Crystal



Note: Duty cycle is 50%.

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	CONDITION
Clock High Time	t_{CHCX}	20	-	125	nS	
Clock Low Time	t_{CLCX}	20	-	125	nS	
Clock Rise Time	t_{CLCH}	-	-	10	nS	
Clock Fall Time	t_{CHCL}	-	-	10	nS	

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{XIN}	Clock Input Frequency	-	12	-	MHz
XIN_{DUTY}	Clock Input Duty Cycle	45	50	55	%

6.3.2 External 32.768 kHz Low Speed Crystal

Symbol	Parameter	Min.	Typ.	Max	Unit	Test Conditions
V_{LXT}	Operation Voltage	1.2		1.8	V	-
T_A	Temperature	-20	-	85	°C	-
f_{LXT}	Clock Frequency	-	32.768	-	kHz	-

6.3.2.1 Typical Crystal Application Circuits

Crystal	C1	C2
32.768 kHz	12 pf	12 pf

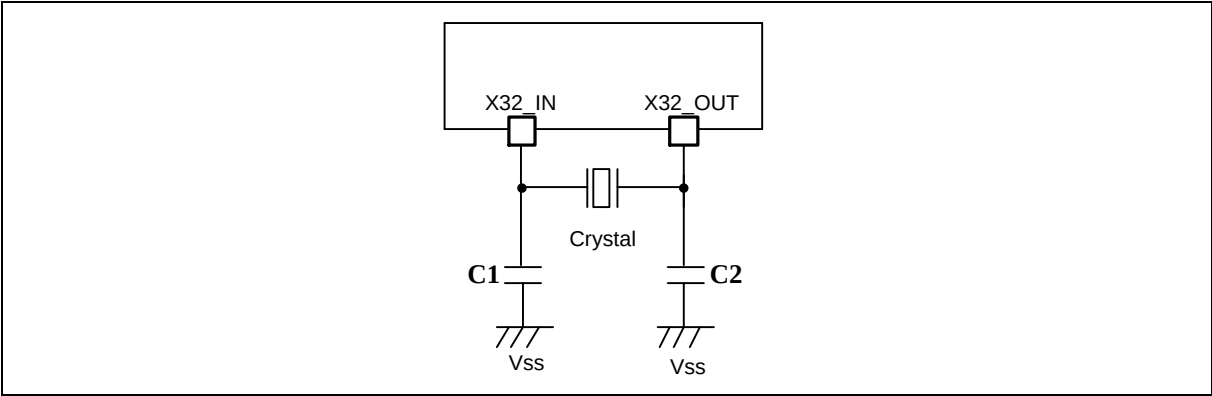
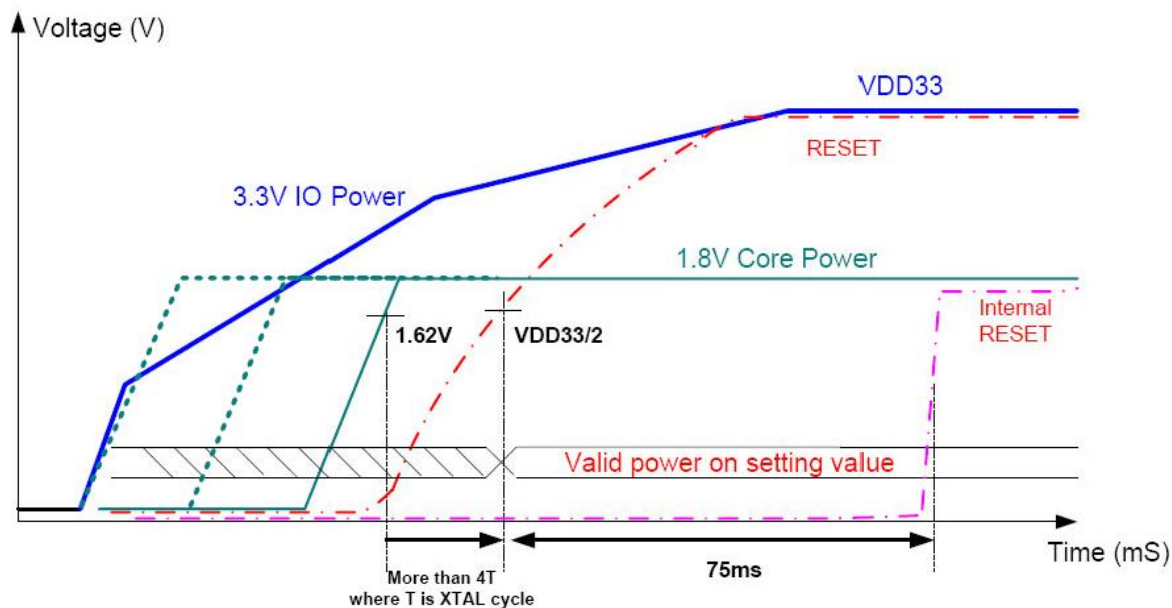


Figure 6.3-1 Typical LXT Crystal Application Circuit

6.3.3 Power-on Sequence & RESET



6.3.3.1 Power up sequence

- Higher Voltage IO (3.3V) First
- Sequence: $T_{33} \geq T_{18}$, (The time of delay gap between $< 500\mu S$ is prefer)

6.3.3.2 Power down Sequence

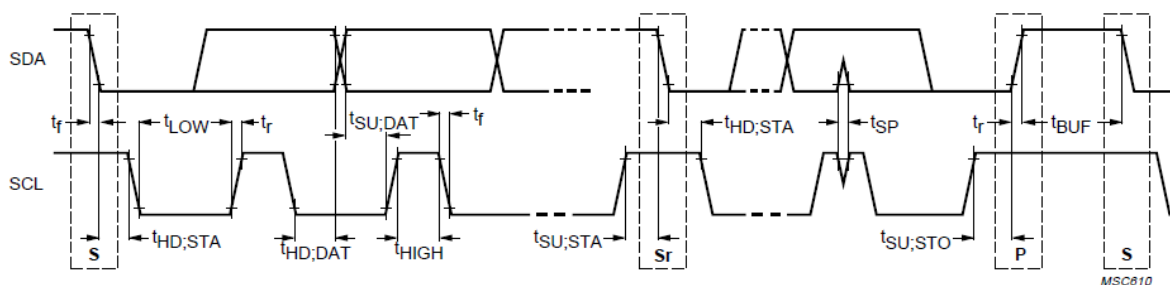
- The lower voltage (1.8V) should be powered down first
- Sequence: $T_{18} \geq T_{33}$

Note.

T18 represents 1.8V powered time for Core power

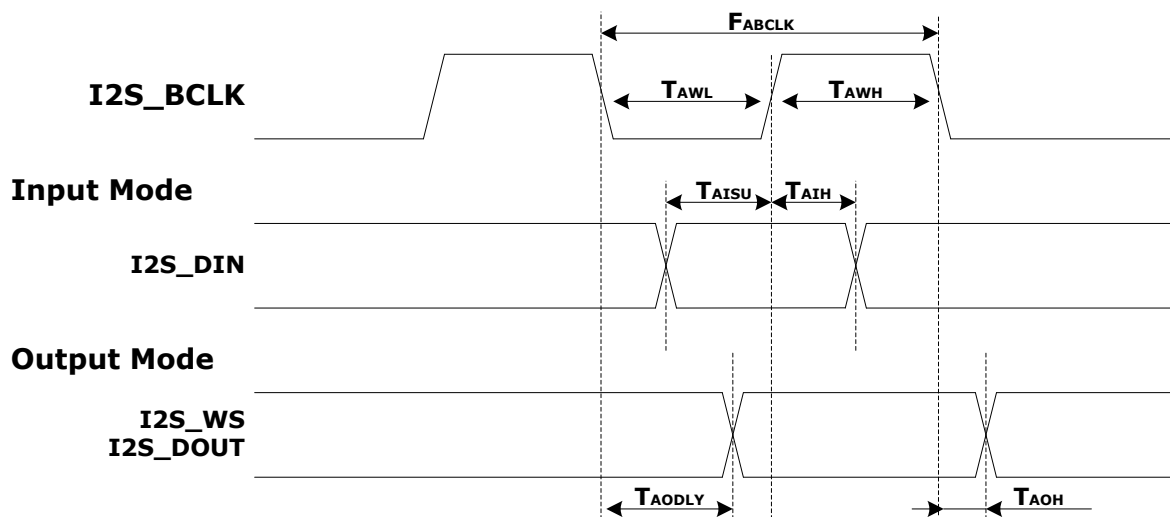
T33 represents 3.3V powered time for I/O power

6.3.4 I2C Interface



PARAMETER	SYMBOL	STANDARD-MODE		FAST-MODE		UNIT
		MIN.	MAX.	MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD,STA}$	4.0	—	0.6	—	μs
LOW period of the SCL clock	t_{LOW}	4.7	—	1.3	—	μs
HIGH period of the SCL clock	t_{HIGH}	4.0	—	0.6	—	μs
Set-up time for a repeated START condition	$t_{SU,STA}$	4.7	—	0.6	—	μs
Data hold time: for CBUS compatible masters (see NOTE, Section 10.1.3) for I ² C-bus devices	$t_{HD,DAT}$	5.0	—	—	—	μs
		0 ⁽²⁾	3.45 ⁽³⁾	0 ⁽²⁾	0.9 ⁽³⁾	μs
Data set-up time	$t_{SU,DAT}$	250	—	100 ⁽⁴⁾	—	ns
Rise time of both SDA and SCL signals	t_r	—	1000	$20 + 0.1C_b^{(5)}$	300	ns
Fall time of both SDA and SCL signals	t_f	—	300	$20 + 0.1C_b^{(5)}$	300	ns
Set-up time for STOP condition	$t_{SU,STO}$	4.0	—	0.6	—	μs
Bus free time between a STOP and START condition	t_{BUF}	4.7	—	1.3	—	μs
Capacitive load for each bus line	C_b	—	400	—	400	pF
Noise margin at the LOW level for each connected device (including hysteresis)	V_{nL}	$0.1V_{DD}$	—	$0.1V_{DD}$	—	V
Noise margin at the HIGH level for each connected device (including hysteresis)	V_{nH}	$0.2V_{DD}$	—	$0.2V_{DD}$	—	V

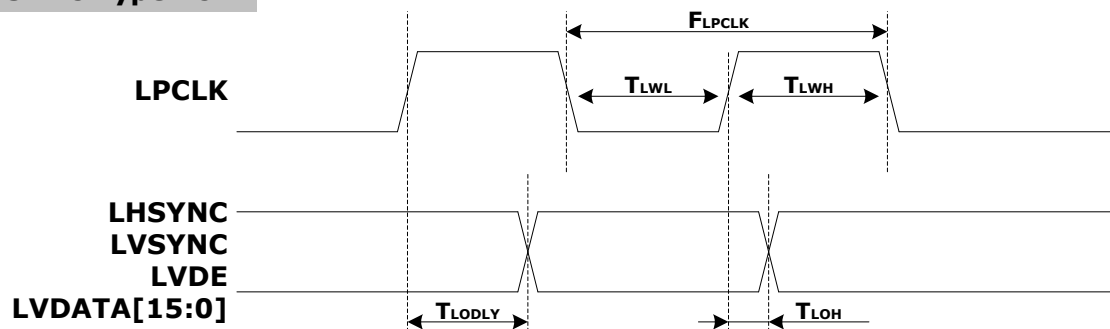
6.3.5 I2S Interface



Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{ABCLK}	I2S_BCLK Clock Frequency	-	-	16	MHz
T_{AWL}	I2S_BCLK Clock Low Time	31.25	-	-	ns
T_{AWH}	I2S_BCLK Clock High Time	31.25	-	-	ns
T_{AISU}	I2S_DIN Setup Time	10	-	-	ns
T_{AIH}	I2S_DIN Hold Time	10	-	-	ns
T_{AODLY}	I2S_DOUT Output Delay Time	-	-	0.5	ns
T_{AOH}	I2S_DOUT Output Hold Time	0.1	-	-	ns

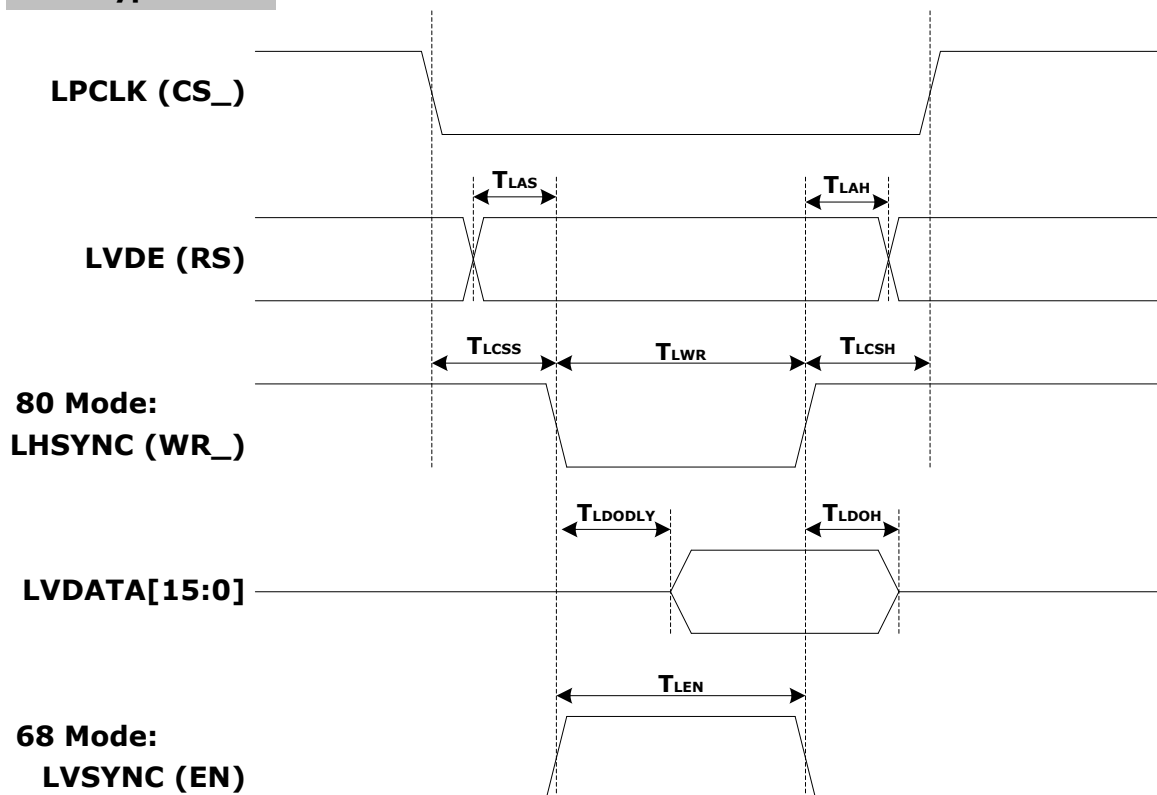
6.3.6 LCD/Display Interface

SYNC Type LCD



Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{LPCLK}	LPCLK Clock Frequency	-	-	27	MHz
T_{LWL}	LPCLK Clock Low Time	18.5	-	-	ns
T_{LWH}	LPCLK Clock High Time	18.5	-	-	ns
T_{LODLY}	LHSYNC, LVSYNC, LVDE and LVDATA Output Delay Time	-	-	1.3	ns
T_{LOH}	LHSYNC, LVSYNC, LVDE and LVDATA Output Hold Time	0.67	-	-	ns

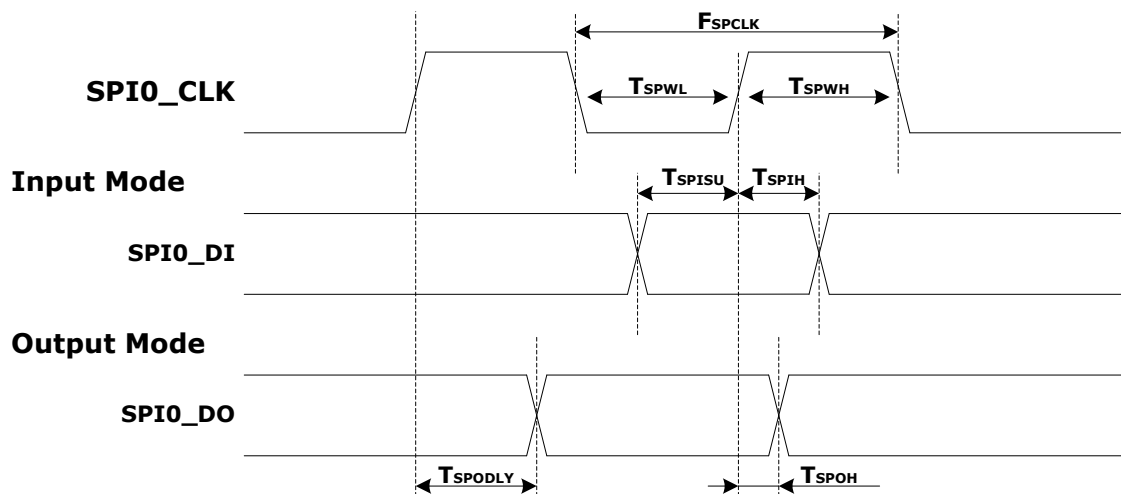
MPU Type LCD



Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
T_{LCSS}	CS_ to WR_ Setup Time		2	-	-	PCLK
T_{LCSH}	CS_ to WR_ Hold Time		1	-	-	PCLK
T_{LAS}	RS to WR_ Setup Time		1	-	-	PCLK
T_{LAH}	RS to WR_ Hold Time		1	-	-	PCLK
T_{LDODLY}	LVDATA Output Delay Time		-	-	1	PCLK
T_{LDOH}	LVDATA Output Hold Time		1	-	-	PCLK
T_{LWR}	WR_ Pulse Width	80 Mode	1	-	-	PCLK
T_{LEN}	EN Pulse Width	68 Mode	1	-	-	PCLK

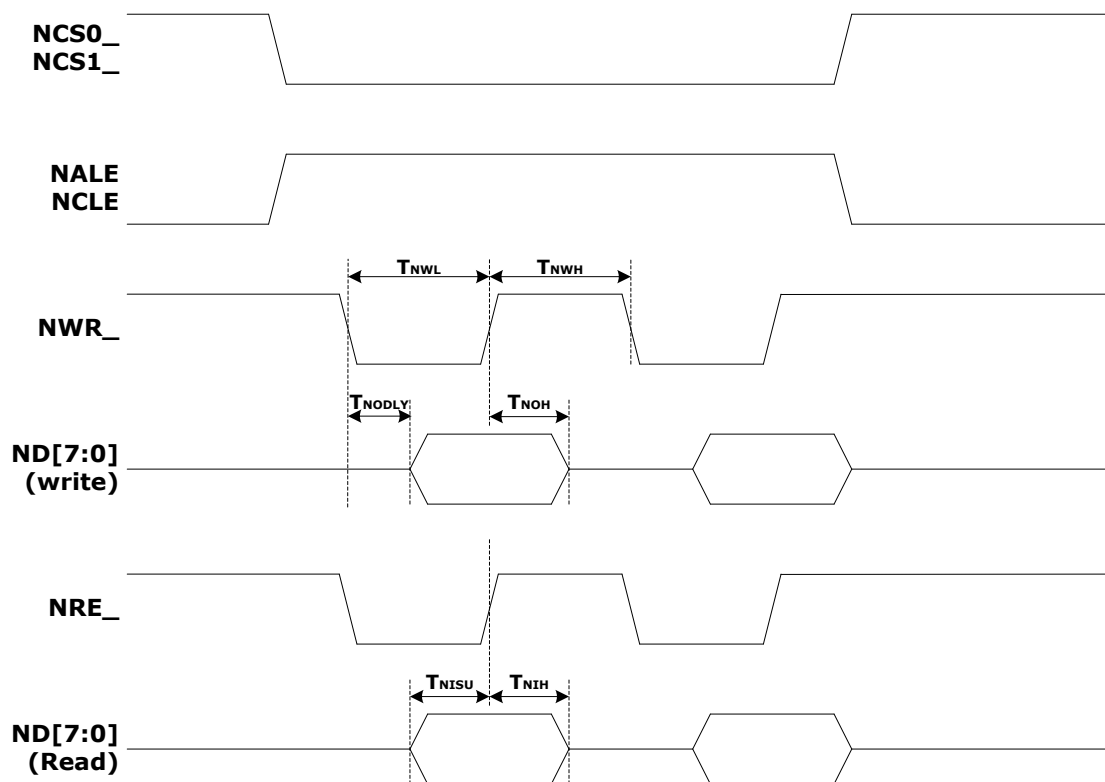
Note: Where PCLK is APB bus clock.

6.3.7 SPI Interface



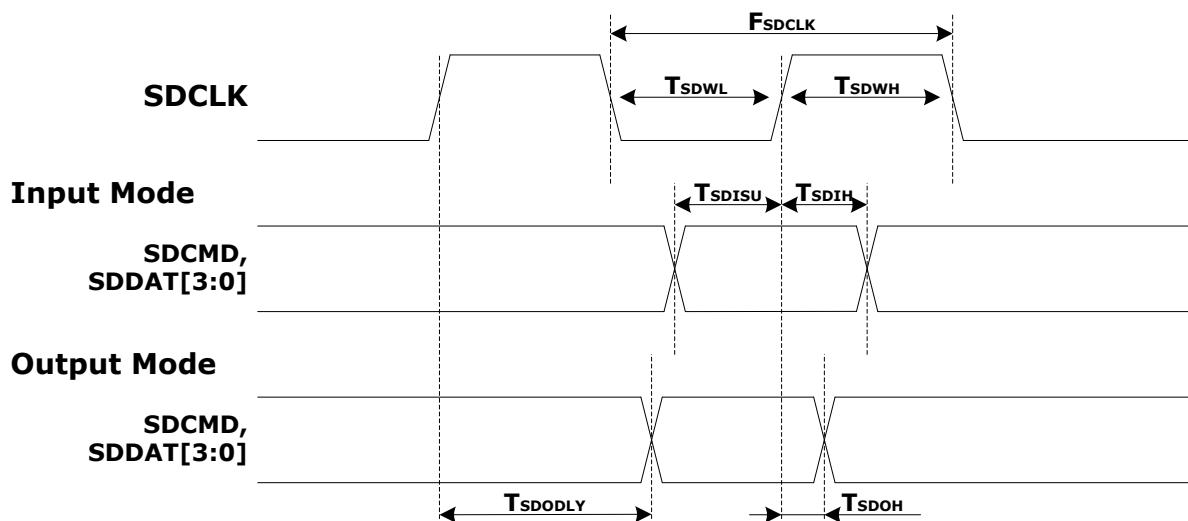
Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{SPCLK}	SPI0_CLK Clock Frequency	-	-	25	MHz
T_{SPWL}	SPI0_CLK Clock Low Time	20	-	-	ns
T_{SPWH}	SPI0_CLK Clock High Time	20	-	-	ns
T_{SPISU}	SPI0_DI Setup Time	10	-	-	ns
T_{SPIH}	SPI0_DI Hold Time	10	-	-	ns
T_{SPODLY}	SPI0_DO Output Delay Time	-	-	1	ns
T_{SPOH}	SPI0_DO Output Hold Time	0.2	-	-	ns

6.3.8 NAND Interface



Symbol	Parameter	Min.	Typ.	Max.	Unit
T_{NWL}	Write Pulse Low Width	10	-	-	ns
T_{NWH}	NWR_ High Hold Time	10	-	-	ns
T_{NODLY}	ND[7:0] Output Delay Time	-	-	2.5	ns
T_{NOH}	ND[7:0] Output Hold Time	10	-	-	ns
T_{NISU}	ND[7:0] Data in Setup Time	3.2	-	-	ns
T_{NIH}	ND[7:0] Data in hold time	1	-	-	ns

6.3.9 SD Card Interface



Symbol	Parameter	Min.	Typ.	Max.	Unit
Clock SDCLK					
F_{SDCLK}	Clock Frequency in Data Transfer Mode	-	-	24	MHz
F_{SDCLK}	Clock Frequency in Identification Mode	100	-	400	KHz
T_{SDWL}	Clock Low Time	10	-	-	ns
T_{SDWH}	Clock High Time	10	-	-	ns
Input SDCMD, SDDAT[3:0] (referenced to SDCLK)					
T_{SDISU}	Input Setup Time	6	-	-	ns
T_{SDIH}	Input Hold Time	2	-	-	ns
Output SDCMD, SDDAT[3:0] (referenced to SDCLK)					
T_{SDODLY}	Output Delay Time	-	-	14	ns
T_{SDOH}	Output Hold Time	2.5	-	-	ns

6.3.10 USB PHY Specifications

6.3.10.1 USB DC Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ	Max.	Unit
V _{IH}	Input high (driven)		2.0	-	-	V
V _{IL}	Input low		-	-	0.8	V
V _{DI}	Differential input sensitivity	PADP-PADM	0.2	-	-	V
V _{CM}	Differential common-mode range	Includes VDI range	0.8	-	2.5	V
V _{SE}	Single-ended receiver threshold		0.8	-	2.0	V
	Receiver hysteresis		-	400	-	mV
V _{OL}	Output low (driven)		0	-	0.3	V
V _{OH}	Output high (driven)		2.8	-	3.6	V
V _{CRS}	Output signal cross voltage		1.3	-	2.0	V
R _{PU}	Pull-up resistor		1.425	-	1.575	kΩ
V _{TRM}	Pull-down resistor		14.25	-	15.75	kΩ
Z _{DRV}	Termination Voltage for upstream port pull up (RPU)		3.0	-	3.6	V
C _{IN}	Driver output resistance	Steady state drive*	28	-	49.5	Ω
V _{IH}	Transceiver capacitance	Pin to V _{SS}	-	-	20	pF

Note: Driver output resistance does not include series resistor resistance.

6.3.10.2 USB Full-Speed Driver Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ	Max.	Unit
T _{FR}	Rising time	CL = 50p	4	-	20	ns
T _{FF}	Falling time	CL = 50p	4	-	20	ns

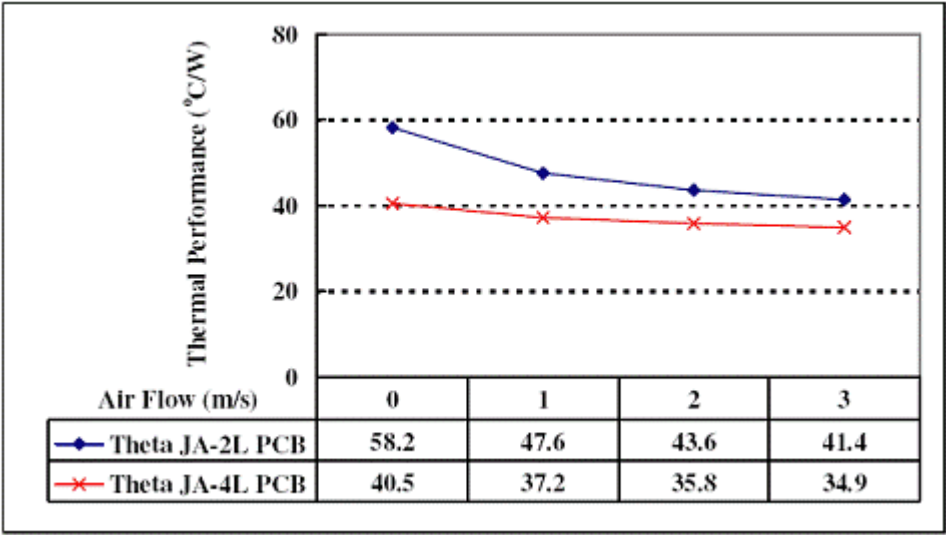
6.3.10.3 USB High-Speed Driver Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ	Max.	Unit
T _{FR}	Rising time	CL = 5p	500			ps
T _{FF}	Falling time	CL = 5p	500			ps

6.3.10.4 Specification of Low Voltage Reset

Parameter	Conditions	Min.	Typ	Max.	Unit
Operation voltage	-20°C ~ +85°C	2	3.3	3.6	V
LVR Detect Levels	VDD rises	2.16	2.4	2.64	V
	VDD falls	2.115	2.35	2.585	V

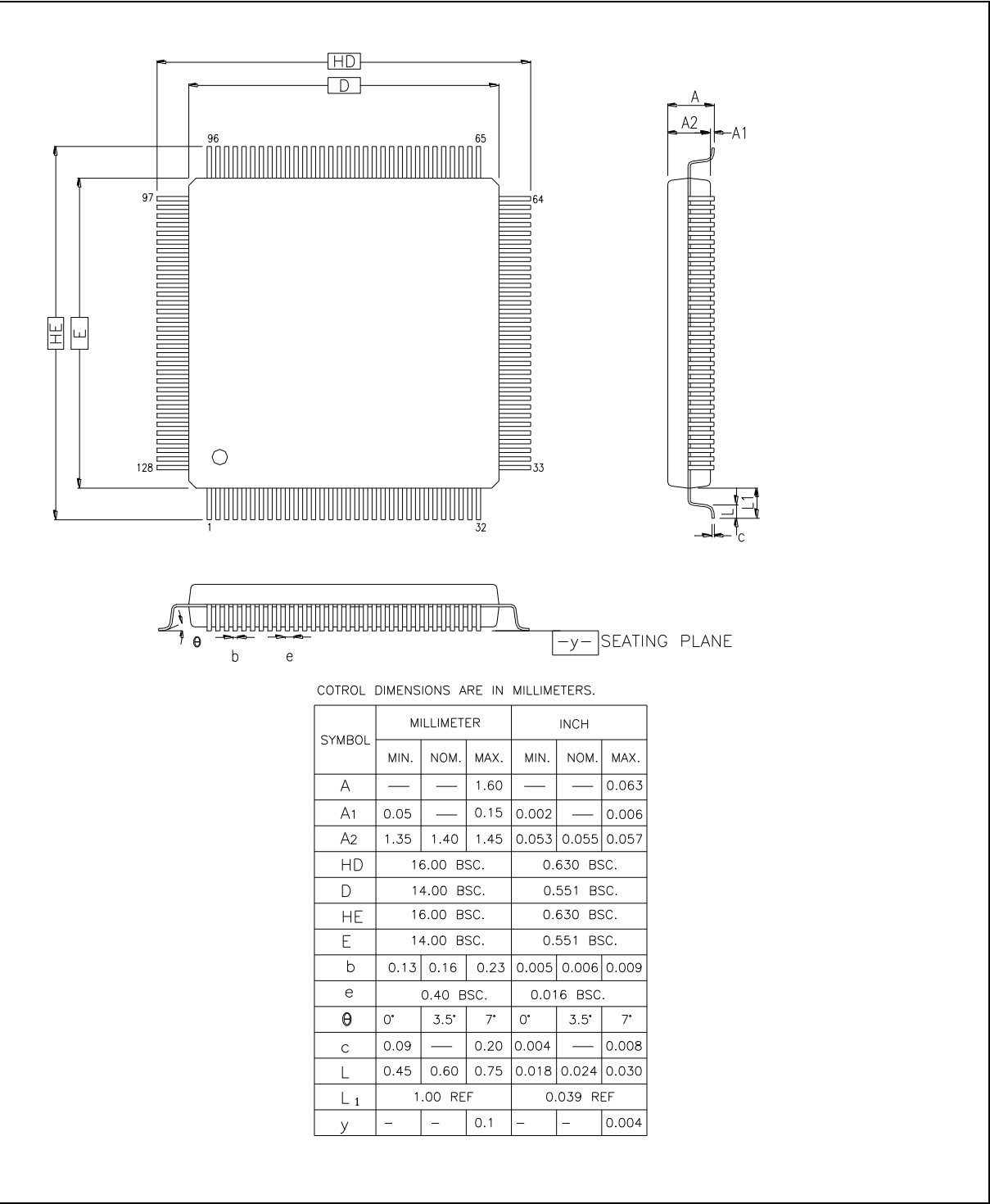
6.3.11 Thermal characteristics of N9H20 sweries LQFP-128 Package



Thermal Performance of LQFP-128 under Forced Convection

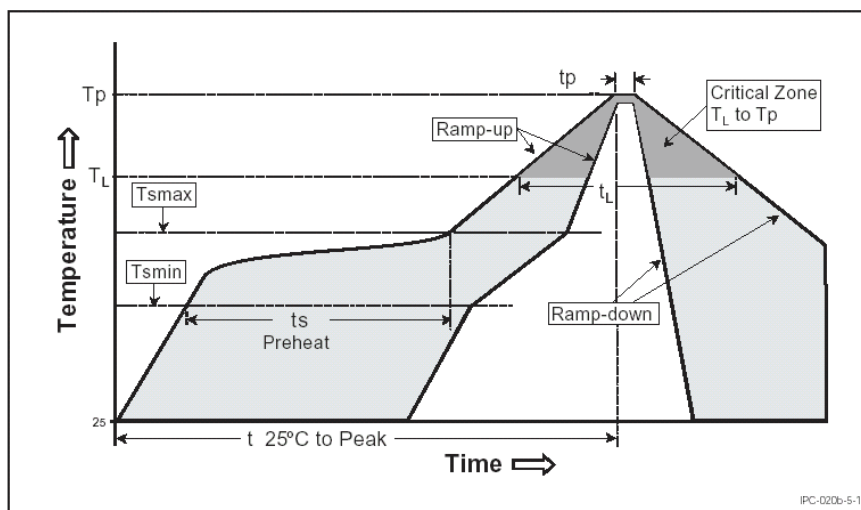
7 PACKAGE DIMENSIONS

7.1 LQFP 128L (14x14x1.4mm footprint 2.0mm)



7.2 PCB Reflow Profile Suggestion

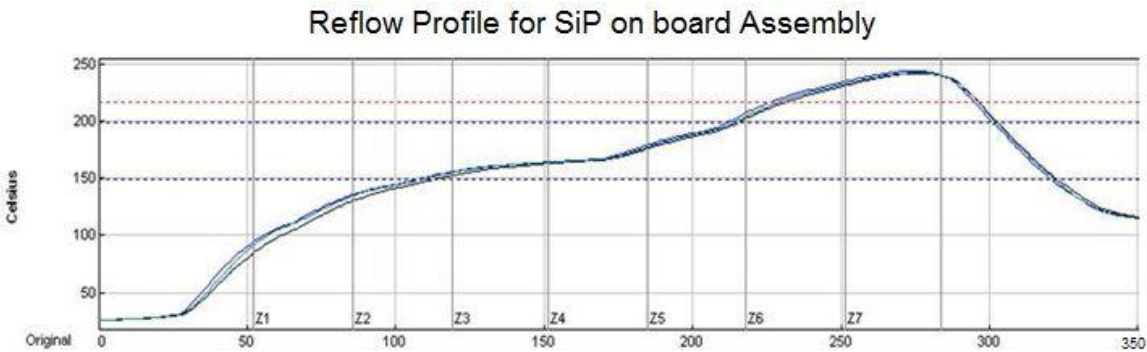
7.2.1.1 Profile Setting Consideration



Profile Feature	Sn-Pb Eutectic Assembly		Pb-Free Assembly	
	Large Body	Small Body	Large Body	Small Body
Average ramp-up rate (T_L to T_p)	< 3°C/second		< 3°C/second	
Preheat				
-Temperature Min (T_{smin})	100°C		150°C	
-Temperature Max (T_{smax})	150°C		200°C	
-Time (min to max) (t_s)	60-120 seconds		60-180 seconds	
Time maintained above:				
-Temperature (T_L)	183°C		217°C	
-Time (t_L)	60-150 seconds		60-150 seconds	
Peak Temperature (T_p)	225+0/-5°C		245+5/-5°C	
Time within 5°C of actual Peak Temperature (t_p)	10-20 seconds		10-30 seconds	
Ramp-down Rate	3°C/second max.		3°C/second max.	
Time 25°C to Peak Temperature	6 minutes max.		8 minutes max.	

Note: 1. All temperatures refer to topside of the package, measured on the package body surface.
2. Depends on other parts on board density and follower solder paste manufacturers's guideline

7.2.1.2 Profile Suggestion for N9H20 series



Preheat time	150℃—200℃: 105+/-15sec
Dwell time	Over 220℃: 70+5/-10 sec
Peak Temp	240 +10/-5℃
Ramp Up/Down Rate	Up: 3 +0/-2 ℃/ sec Down: 2 +0/-1℃/ sec

8 REVISION HISTORY

Date	Revision	Description
2018.04.19	1.00	Preliminary version.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, “Insecure Usage”.

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer’s risk, and in the event that third parties lay claims to Nuvoton as a result of customer’s Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*