FlashLite Platform with OpenVG and Video Accelerators

W55FA95 / N3291x

Design Guide

Ver.A0

Table of Contents

1 General Description

The W55FA95 is specially designed for accelerating FL (FlashLite) performance, when bitmap graphics is mainly used for constructing the arts used in content authoring. It is embedded with a 2D vector graphic (OpenVG) accelerator to boost FL performance while off-loading the CPU to save power consumption.

The W55FA95 is built on the ARM926EJ-S CPU core and integrated with video decoder (H.264, MPEG4, …), OpenVG (Hardware accelerated 2D vector graphic),JPEG codec, CMOS sensor interface, 32-channel SPU (Sound Processing Unit), ADC, DAC, & TV encoder, for meeting various kinds of application needs while saving the BOM cost. The combination of ARM926 @ 300MHz, DDR, 2D accelerator, & USB2.0 HS Device makes the W55FA95 the best choice for FlashLite-based ELA devices.

The W55FA95 is ported with FL3.1.8 and FL4.0 engines under Linux OS to leverage the driver availability of emerging functionalities, like Wi-Fi, browser, etc. On the other hand, the open source code environment also gives the product development more flexibility. Nuvoton's continuous optimization at FL kernel & flexible customization via EAS (Extended Action Script) provide customers with a cost-effective Flash Lite solution. Moreover, hybrid FL platform is introduced to best utilize the performance advantage at native C programming while enjoying the inherent benefit at FlashLite content development.

Maximum resolution for the W55FA95 is D1 (720x480) @ TV output & 1,024x768 @ TFT LCD panel. With increasing popularity of the Flash content, the FL player is the best fit for the application that requires fast turn-around time @ content development. The W55FA95 is well-positioned in terms of cost/performance for the ELA market where bitmap graphics is extensively used.

To reduce system complexity while cutting the BOM cost, the W55FA95 also comes with a 128-pin MCP (Multi-Chip Package) in LQFP. The 32Mbitx16 DDR is stacked inside the MCP to ensure higher performance and minimize the system design efforts, like EMI, noise coupling. Total BOM cost could be cut by employing 2-layer PCB along with the elimination of damping resistors, EMI prevention components, & with less board space.

Applications

- **e-Reader for kids.**
- **ELA (Educational Learning Aid)**
- **TV game**
- **HMI**
- **Home Appliance**
- **Advertisement**

2 Feature

CPU

- ARM926EJ-S 32-bit RISC CPU with 16KB I-Cache & 16KB D-Cache.
- Frequency up to 300MHz at worst case.
- Frequency up to 360MHz at 1.34V.
- J-TAG 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
    - SD-to-NAND flash bridge
  - NAND Interface
    - Raw NAND Flash
    - PBA-NAND flash
  - SPI Flash
  - USB

Memory Controller

- SDRAM Interface
- SDR, DDR, LPDDR & DDR2 type SDRAM supported.
- Frequency up to 150MHz.
- 16-bit data bus width supported.
- 2 external SDRAM banks (2 chip select pins) supported.
- Total memory size up to 256MB (128MB x 2).

EDMA (Enhanced DMA)

- Totally 6 DMA channels supported.
  - 4 peripheral DMA channels for transfer between memory and on-chip peripherals, such as ADC, UART and SPI.
  - 2 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.

Capture (CMOS Sensor I/F)

- CCIR601 & CCIR656 interfaces supported for connection to CMOS image sensor.
- Resolution up to 3M pixel.
- YUV422 and RGB565 color format supported for data-in from CMOS sensor.
- YUV422, RGB565, RGB555 and Y-only color format supported for data storing to system memory.
- Planar and packet data format supported for data storing to system memory.
- Image cropping supported with the cropping window up to 4096x2048.
- Image scaling-down supported.
  - Vertical and horizontal scaling-down for preview mode supported.
    - The scaling factor is N/M.
    - Two pairs of configurable 16-bit N and 16-bit M for vertical and horizontal scaling-down.
    - The value of N has to be equal to or less than M.
  - Frame rate control supported.
- Combines two interlace fields to a single frame supported for data in from TV-decoder.
- 3 kinds of color processing effect
  - Negative picture
  - Sepia picture
  - Posterization

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

- u **Support standard default Huffman-table and programmable Huffman-table for decode**
- u **Support arbitrarily 1X~8X image up-scaling function for encode mode**
- u **Support down-scaling function for encode and decode modes**
- u **Support specified window decode mode**
- u **Support quantization-table adjustment for bit-rate and quality control in encode mode**
- u **Support rotate function in encode mode**
- u **Support On-the-Fly interface with video data processor**

n **Packet Format**

- u **Support to encode interleaved YUYV format input image, output bitstream 4:2:2 and 4:2:0 format**
- u **Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0 format image**
- u **Support decoded output image RGB555, RGB565 and RGB888 formats.**
- u **The encoded JPEG bit-stream format is fully compatible with JFIF and EXIF standards**
- u **Support arbitrary width and height image encode and decode**
- u **Support three programmable quantization-tables**
- u **Support standard default Huffman-table and programmable Huffman-table for decode**
- u **Support arbitrarily 1X~8X image up-scaling function for encode mode**
- u **Support down-scaling function 1X~ 16X for Y422 and Y420, 1X~ 8X for Y444 for decode mode**
- u **Support specified window decode mode**
- u **Support quantization-table adjustment for bit-rate and quality control in encode mode**
- u **Support On-the-Fly interface with video data processor**
- u **Support Scatter-Gather mode for output frame buffer**

l Video Decoder

n Multi-standard video decoder

- u **MPEG-4 part-II simple profile decoding**
- u **H.264/AVC baseline profile decoding**
- u **VC-1(WMV9) main profile decoding**
- u **MPEG2 main profile decoding**

n Minimum decoding size is 64-pixel in both horizontal and vertical.

n Pre/post rotation/mirroring

- u 90xn (n=0,1,2,3) degree rotation /w mirroring for decoded image

n Up to 720x480 30fps decoding

l 2D Graphic and Video Data Processor

- **2D Graphic engine**
  - **Bit Block Transfer (BitBLT)**
  - **Alpha blending and color transformation supported**
  - **Bresenham Line Draw**
  - **Clipping**
  - **Rotation**
  - **Scaling up/down**
- **Video Data Processor**
  - **Image/video data format conversation**
    - **Source**
      - **Planar: YUV/YCbCr 444/422/420**
      - **Packet: YUV 422**
    - **Destination**
      - **Packet: YUV 422, RGB 555/565/888**
  - **Image/video 2-D rotation/coordinate transformation**
    - **Left/Right with 90°/180°, mirror, up-side-down and flip/flop**
  - **Arbitrary scaling up/down with Bilinear filter**
  - **MMU DMA**

- Hardware Accelerated 2D Vector Graphic (OpenVG)
  - **Maximum display color: RGB888**
  - **Vector functions**
    - **Path data processing**
      - **Path primitives: Close, Move, Line, Cubic Bezier, Quadratic Bezier**
      - **Elliptical Arc**
      - FLL rendering: RULE (Even, Odd, Non-Zero, Zero)
      - **STROKE rendering**
      - **Paint mode: Color, Radial Gradient, Linear Gradient, Pattern**
    - **Image Filter**
      - Color Matrix operation, Convolution filter, Gaussian Blur, Lookup Table operation
    - **Pixel Operation**
      - **Scissor, Color Transform, Blending, Anti-aliasing (2x2, 4x4), Multi-sample fill scene anti-aliasing (MSAA)**
  - **2D function (Sprite rendering)**
    - **Frame buffer type sprite rotation, stretch, deformation, BitBlt functions, super sample full anti-aliasing.**

- u **Linear rendering, point rendering**
- u **Aplha-blending**
- u **Point sampling, Bi-linearing filtering**
- u **Indexed color-texture, compressed texture image**

n **Other functions**

- u Copy arbitrary rectangular area to the buffer with color format conversion
- u **Display List Engine**

l VPOST (LCD/TV Interface)

- n **8/16/18/24-bit SYNC type and 8/9/16/18/24-bit MPU type TFT LCD supported.**
- n **Color format supported:**
  - u **YCbCr422, RGB565, RGB555, and RGB888 color formats supported for data in.**
  - u **YCbCr422, RGB565, RGB555, and RGB888 color formats supported for data out.**
- n **XGA (1024x768), SVGA (800x600), WVGA (800x480), D1 (720X480), VGA (640x480), WQVGA (480x272), QVGA (320x240) and HVGA (640x240) resolution supported.**
  - u **The maximum resolution is up to D1 (720X480) for TV output.**
  - u **The maximum resolution is up to 1024X768 for TFT LCD panel**
- n **Display scaler – to fit different size of LCD panels**
  - u **Horizontal: At most 4.0x scale.**
  - u **Vertical: At most 3.0x scale.**
- n **For SYNC type LCD:**
  - u **For 8-bit bus**
    - l **CCIR601 YCbCr422 packet mode (NTSC/PAL) supported.**
    - l **CCIR601 RGB Dummy mode (NTSC/PAL) supported.**
    - l **CCIR656 interface supported.**
    - l **RGB Through mode supported.**
  - u **For 16/18/24-bit bus**
    - l **Parallel pixel data output mode (1-pixel/1-clock)**
- n **NTSC/PAL interlace & non-interlace output supported.**
- n **Color format transform supported:**
  - u **Color format transform between YCbCr422 and RGB565.**
  - u **Color format transform from YCbCr422 to RGB888.**
- n **TV encoder supported.**
- n **Dual screen, outputs to TV and LCD simultaneously with same content, supported.**

- u LCD panel should be 320X240 MPU-type, or 8-bit SYNC-type LCD panel with TV timing.
- n Notch filter for NTSC supported to remove the rainbow color effect.
- n Support OSD function to overlap system information like battery life, brightness tuning, volume tuning or muting, etc.

SPU (Sound Processing Unit)

- n 32 stereo channels supported
- n PCM8/PCM16/4-bit MDPCM/TONE source format supported
- n 7-bit volume control supported for each of 32 channels
- n 5-bit pan control supported for each L/R of 32 channels
- n 10-band equalizer supported
- n Special code supported for loop playing and event detection
- n 13-bit DFA supported for source sampling rate control.

Audio DAC

- n 16-bit stereo DAC supported with headphone driver output.
- n H/W volume control supported.
- n Built-in PLL for supporting of various sampling rate

I2S Controller

- n I2S interface supported to connect external audio codec.
- n 16/18/20/24-bit data format supported.

Storage Interface Controller

- n Interface to NAND Flash:
  - u 8-bit data bus width supported.
  - u SLC and MLC type NAND Flash supported.
  - u 512B, 2KB, 4KB and 8KB page size NAND Flash supported.
  - u ECC4, ECC8, ECC12 and ECC15 algorithm supported for ECC generation, error detection and error correction.
  - u PBA-NAND flash supported
- n Interface to SD/MMC/SDIO/SDHC/micro-SD cards supported.
  - u SD-to-NAND flash bridge supported
- n DMA function supported to accelerate the data transfer between system memory and NAND Flash or SD/MMC/SDIO/SDHC/micro-SD.

USB Device Controller

- n USB2.0 HS (High-Speed) x 1 port.
- n 6 configurable endpoints supported.

- **Control, Bulk, Interrupt and Isochronous transfers supported.**
- **Suspend and remote wakeup supported.**

USB Host Controller

- **USB1.1 Host x 2 ports**
- **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 interfaces are supported**
  - **Both master and slave mode are supported in SPI interface 0.**
  - **Only master mode is supported in SPI interface 1.**
    - **Byte transfer with configurable stop interval supported.**

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.**
- **Dual clock sources are support, accurate 32.768 KHz crystal oscillation circuit and built-in coarse 32 KHz RC oscillator.**
- **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

- **Multi-purpose programmable general I/Os are 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 2MHz input clock supported.
  - Maximum 150K/s conversion rate supported.

Keypad Interface (KPI)

- Matrix keypad interface supported.
- Maximum 8X8 and minimum 3X3 keypad matrix supported.
- Configurable key de-bounce supported.
- Low power wakeup mode supported.
- Configurable three-key reset supported.

AES (Advance Encyption Standard) Engine

- Supports both encryption and decryption
- Supports only CBC (Cipher Block Chaining) mode
- All three kinds of key length, 128, 192, and 256 bits, are supported
- Built-in DMA function

Test Interface Controller

- SPI-Like slave interface supported for test purpose.
- 1-bit, 2-bit and 4-bit data width supported.
- Single and burst (4 32-bit data) data access supported.

Power Management

- Advanced power management including Power Down, Deep Standby, CPU Standby, and Normal Operating modes.
  - Normal Operating Mode
    - Core power is 1.2V and chip is in normal operation
  - CPU Standby Mode
    - Core power is 1.2V and only ARM CPU clock is turned OFF.
  - Deep Standby Mode
    - Core power is 1.2V and all IP clocks are turned OFF.
  - Power Down Mode
    - Only the RTC power is ON. Other 3.3V and 1.2V power are OFF.

Operating Voltage

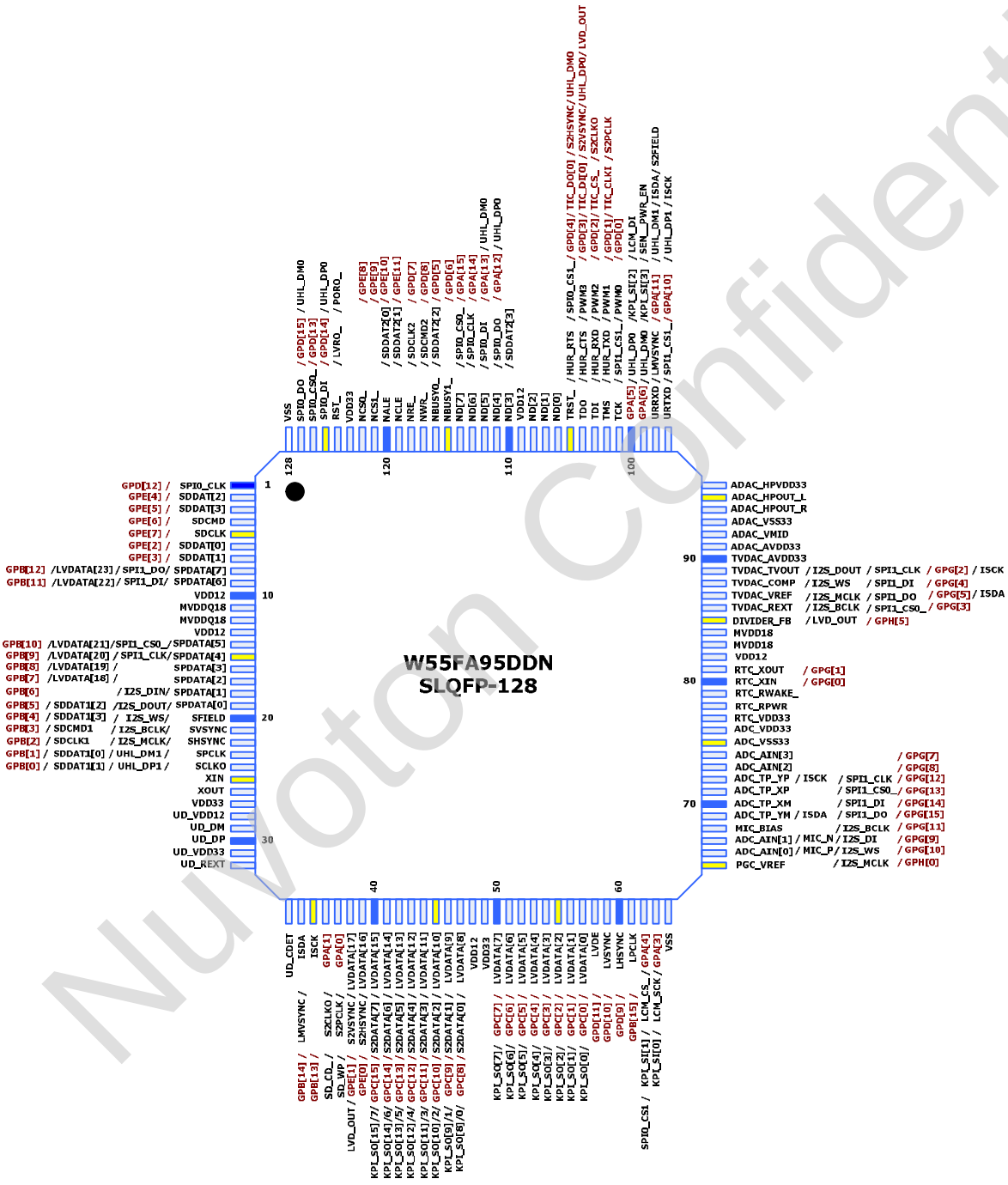
- I/O: 3.3V
- Core: 1.2V for 300MHz, 1.34V for 360MHz.

l Package

- **LQFP-128**

3 Pin Diagram

3.1 LQFP-128



4 Multi-Function Pin Configuration

4.1 Pad Multiplexing

Part Number	Pin Functions			
Package				
	Alternative 1	Alternative2	Alternative 3	Alternativ 4
Default Pin Name & Function				
Clock & Reset				
XIN				
XOUT				
RST_				
JTAG Interface				
TCK	SPI1_CS1_		PWM0	GPD[0]
TMS	HUR_TXD	TIC_CLKI/S2PCLK	PWM1	GPD[1]
TDI	HUR_RXD	TIC_CS_/S2CLKO	PWM2	GPD[2]
TDO	HUR_CTS	TIC_DI[0] / S2VSYNC / UHL_DP0/ LVD_OUT	PWM3	GPD[3]
TRST_	HUR_RTS	TIC_DO[0]/S2HSYNC / UHL_DM0	SPI0_CS1_	GPD[4]
Memory Interface				
MCLK				
MCLK_				
MCKE				
MCS0_				
MCS1_				
MRAS_				
MCAS_				
MWE_				
MA[12:0]				
MBA[0]				
MBA[1]				

MBA[2]				
MD[15:0]				
MDQM[0]				
MDQM[1]				
MDQS0				
MDQS1				
MVREF				
NCS0_				GPE[8]
NCS1_				GPE[9]
NALE	SDDAT2[0]			GPE[10]
NCLE	SDDAT2[1]			GPE[11]
NBUSY0_	SDDAT2[2]			GPD[5]
NBUSY1_				GPD[6]
ND0 / CHIPCFG0				
ND1 / CHIPCFG1				
ND2 / CHIPCFG2				
ND3 / CHIPCFG3	SDDAT2[3]			
ND4 / CHIPCFG4	SPI0_DO	UHL_DP0		GPA[12]
ND5 / CHIPCFG5	SPI0_DI	UHL_DM0		GPA[13]
ND6 / CHIPCFG6	SPI0_CLK			GPA[14]
ND7 / CHIPCFG7	SPI0_CS0			GPA[15]
NRE_	SDCLK2			GPD[7]
NWR_	SDCMD2			GPD[8]
Sensor Interface				
SCLKO	UHL_DP1		SDDAT1[1]	GPB[0]
SPCLK	UHL_DM1		SDDAT1[0]	GPB[1]
SHSYNC	I2S_MCLK		SDCLK1	GPB[2]
SVSYNC	I2S_BCLK		SDCMD1	GPB[3]
SFIELD	I2S_WS		SDDAT1[3]	GPB[4]
SPDATA[0]	I2S_DOUT		SDDAT1[2]	GPB[5]
SPDATA[1]	I2S_DIN			GPB[6]
SPDATA[2]			LVDATA[18]	GPB[7]

SPDATA[3]			LVDATA[19]	GPB[8]
SPDATA[4]	SPI1_CLK		LVDATA[20]	GPB[9]
SPDATA[5]	SPI1_CS0_		LVDATA[21]	GPB[10]
SPDATA[6]	SPI1_DI		LVDATA[22]	GPB[11]
SPDATA[7]	SPI1_DO		LVDATA[23]	GPB[12]
ISCK				GPB[13]
ISDA	LMVSYNC			GPB[14]
Display Interface				
LPCLK				GPB[15]
LHSYNC				GPD[9]
LVSYNC				GPD[10]
LVDE				GPD[11]
LVDATA[0]	SDRM_TBUSY		KPI_SO[0]	GPC[0]
LVDATA[1]	SDRM_CBUSY		KPI_SO[1]	PGC[1]
LVDATA[2]	SDRM_BBUSY		KPI_SO[2]	GPC[2]
LVDATA[3]	SDRM_TFAIL		KPI_SO[3]	GPC[3]
LVDATA[4] / CHIPCFG[8]	SDRM_CFAIL		KPI_SO[4]	GPC[4]
LVDATA[5] / CHIPCFG[9]	SDRM_BFAIL		KPI_SO[5]	GPC[5]
LVDATA[6] / CHIPCFG[10]			KPI_SO[6]	GPC[6]
LVDATA[7]			KPI_SO[7]	GPC[7]
LVDATA[8]	S2DATA[0]		KPI_SO[8]/0	GPC[8]
LVDATA[9]	S2DATA[1]		KPI_SO[9]/1	GPC[9]
LVDATA[10]	S2DATA[2]		KPI_SO[10]/2	GPC[10]
LVDATA[11]	S2DATA[3]		KPI_SO[11]/3	GPC[11]
LVDATA[12]	S2DATA[4]		KPI_SO[12]/4	GPC[12]
LVDATA[13]	S2DATA[5]		KPI_SO[13]/5	GPC[13]
LVDATA[14]	S2DATA[6]		KPI_SO[14]/6	GPC[14]
LVDATA[15]	S2DATA[7]		KPI_SO[15]/7	GPC[15]
LVDATA[16]	S2HSYNC			GPE[0]
LVDATA[17]	S2VSYNC	LVD_OUT		GPE[1]
UART Interface				
URTXD		UHL_DP1 / ISCK	SPI1_CS1_	GPA[10]
URRXD	S2FIELD	UHL_DM1 / ISDA	LMVSYNC	GPA[11]

SPI0 Interface				
SPI0_CLK				GPD[12]
SPI0_CS0_				GPD[13]
SPI0_DI			UHL_DP0	GPD[14]
SPI0_DO		LVD_OUT	UHL_DM0	GPD[15]
SD Card Interface				
SDCLK				GPE[7]
SDCMD				GPE[6]
SDDAT[0]				GPE[2]
SDDAT[1]				GPE[3]
SDDAT[2]				GPE[4]
SDDAT[3]				GPE[5]
GPIO				
GPA[0] /SD_WP			S2PCLK	
GPA[1]	SD_CD_		S2CLKO	
GPA[3] /LCM_SCK			KPI_SI[0]	
GPA[4] /LCM_CS_	SPI0_CS1_		KPI_SI[1]	
GPA[5] /LCM_DI	UHL_DP0		KPI_SI[2]	
GPA[6] /SEN_PWR_EN	UHL_DM0		KPI_SI[3]	
GPA[8] /DE_POP_CTL				
GPA[9] /TV_DET				
RTC				
RTC_XIN				GPG[0]
RTC_XOUT				GPG[1]
RTC_RWAKE_				
RTC_RPWR				
RTC_VDD				
RTC_VSS				
USB 2.0 I/F				
UD_CDET				
UD_DP				
UD_DM				
UD_REXT				

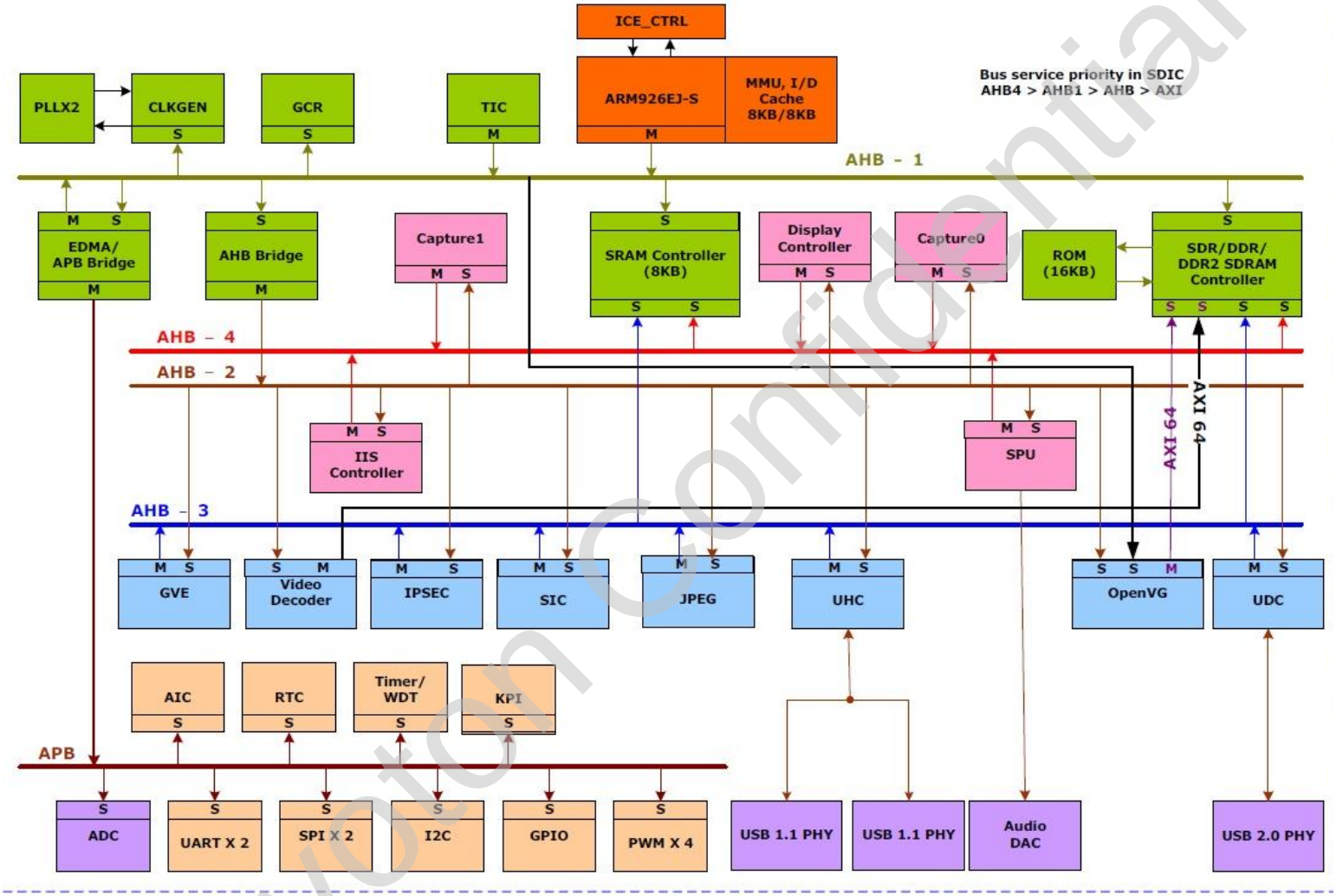
UD_VDD33				
UD_VSS33				
UD_VDD12				
UD_VSS12				
USB 1.1 I/F				
UH_DP1				
UH_DP1_RPD				
UH_DM1				
UH_DM1_RPD				
UH_VDD33				
UH_VSS33				
TV Out				
TVDAC_TVOUT	I2S_DOUT	ISCK	SPI1_CLK	GPG[2]
TVDAC_REXT	I2S_BCLK		SPI1_CS0_	GPG[3]
TVDAC_COMP	I2S_WS		SPI1_DI	GPG[4]
TVDAC_VREF	I2S_MCLK	ISDA	SPI1_DO	GPG[5]
TVDAC_VDD33				
TVDAC_VSS33				
ADC & Touch Panel				
ADC_AIN[4]				GPG[6]
ADC_AIN[3]				GPG[7]
ADC_AIN[2]				GPG(8]
ADC_AIN[1] /MIC_IN_M	I2S_DI			GPG[9]
ADC_AIN[0] /MIC_IN_P	I2S_WS			GPG[10]
ADC_VDD33				
MIC_BIAS	I2S_BCLK			GPG[11]
ADC_TP_YP	SPI1_CLK		ISCK	GPG[12]
ADC_TP_XP	SPI1_CS0_			GPG[13]
ADC_TP_XM	SPI1_DI			GPG[14]
ADC_TP_YM	SPI1_DO		ISDA	GPG[15]
PGC_VREF	I2S_MCLK			GPH[0]
ADC_VSS33				
Audio-DAC				

ADAC_HPOUT_R				
ADAC_HPOUT_L				
ADAC_VMID				
ADAC_VREFH				
ADAC_VREFL				
ADAC_HPVDD33				
ADAC_HPVSS33				
ADAC_AVDD33				
ADAC_AVSS33				
Power/Ground				
PLL_VDD18(12) (2 PLL)				
PLL_VSS18(12) (2 PLL)				
MVDD				
MVSS				
MVDDQ				
MVSSQ				
MVREF_GND_SHIELDING				
VDD33 (I/O)				
VDD12(Core)				
VSS12				
R_FB		LVD_OUT	GPH[5]	
EFUSE_VP				

4.2 LCD Interface Data Bus Configuration

Pin Name	RGB888	RGB666	RGB565	RGB555
LVDATA[0]	B0	B2	B3	B3
LVDATA[1]	B1	B3	B4	B4
LVDATA[2]	B2	B4	B5	B5
LVDATA[3]	B3	B5	B6	B6
LVDATA[4]	B4	B6	B7	B7
LVDATA[5]	B5	B7	G2	G3
LVDATA[6]	B6	G2	G3	G4
LVDATA[7]	B7	G3	G4	G5
LVDATA[8]	G0	G4	G5	G6
LVDATA[9]	G1	G5	G6	G7
LVDATA[10]	G2	G6	G7	R3
LVDATA[11]	G3	G7	R3	R4
LVDATA[12]	G4	R2	R4	R5
LVDATA[13]	G5	R3	R5	R6
LVDATA[14]	G6	R4	R6	R7
LVDATA[15]	G7	R5	R7	-
LVDATA[16]	R0	R6	-	-
LVDATA[17]	R1	R7	-	-
SPDATA[2]	R2	-	-	-
SPDATA[3]	R3	-	-	-
SPDATA[4]	R4	-	-	-
SPDATA[5]	R5	-	-	-
SPDATA[6]	R6	-	-	-
SPDATA[7]	R7	-	-	-

5 Functional Block Diagram

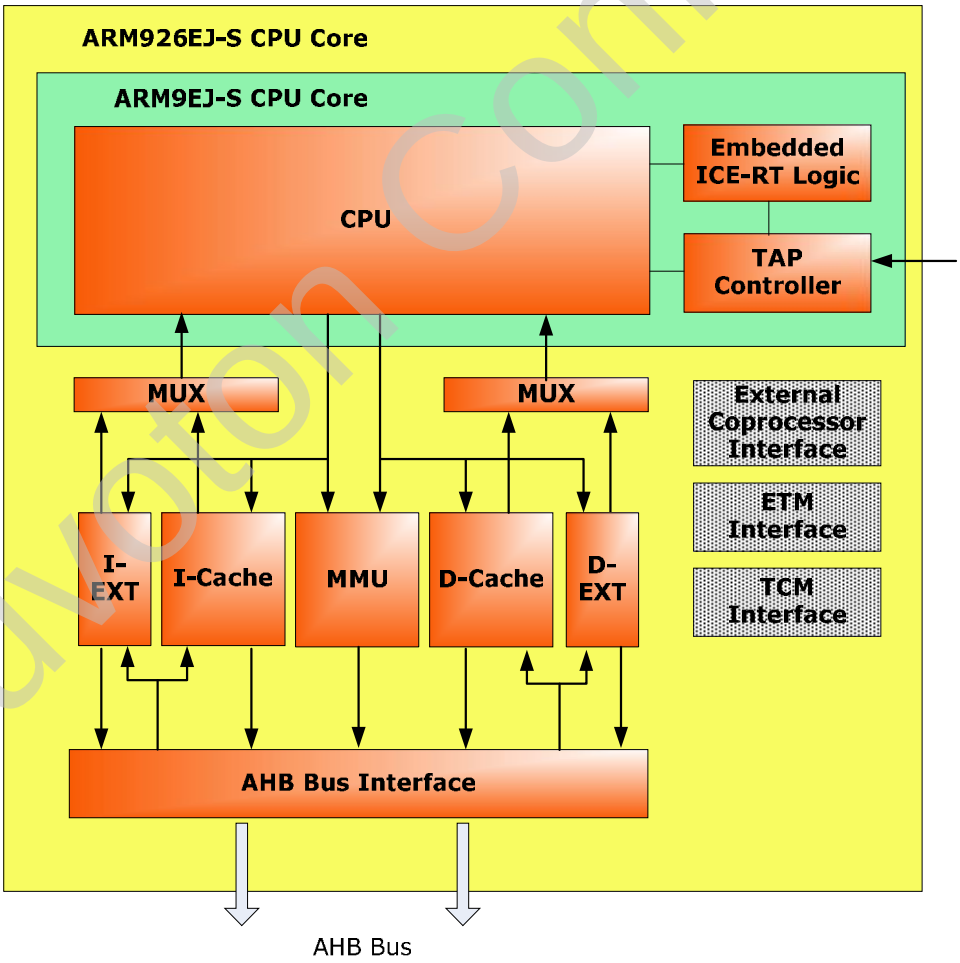


6 Functional Description

6.1 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 tradeoff 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



6.2 System Manager

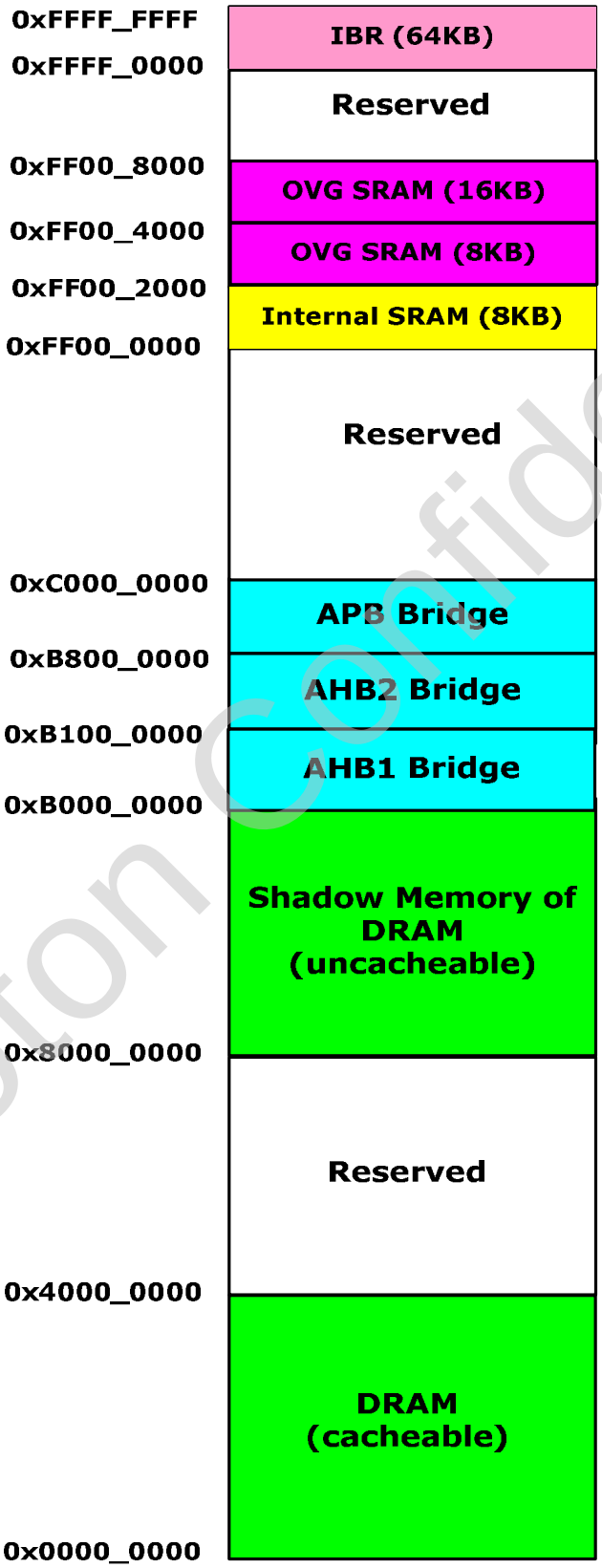
6.2.1 Overview

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.**

6.2.2 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
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
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	OVG	Open VG Control Register
0xB100_5000 – 0xB100_5FFF	0xB100_5000	Reserved	Reserved
0xB100_6000 – 0xB100_6FFF	0xB100_6000	SIC_BA	SIC Control Registers
0xB100_7000 – 0xB100_7FFF	0xB100_7000	VDE	Video Decoder 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	GVE	Graphic Video Control Registers
0xB100_D000 – 0xB100_DFFF	0xB100_D000	AES	AES 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_500 – 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

6.2.3 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.

<table>
<tr><th>Pin Name</th><th>Descriptions</th><th>Register Bit Mapping</th></tr>
<tr><td>LVDATA[6]</td><td>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></td><td>CHIPCFG[10]</td></tr>
<tr><td>LVDATA[5:4]</td><td>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 8KB</td></tr>
<tr><td>0x01</td><td>NAND page size is 4KB</td></tr>
<tr><td>0x10</td><td>NAND page size is 2KB</td></tr>
<tr><td>0x11</td><td>Ignore NAND power-on setting</td></tr>
</table></td><td>CHIPCFG[9:8]</td></tr>
<tr><td>ND[7]</td><td>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></td><td>CHIPCFG[7]</td></tr>
<tr><td>ND[6]</td><td>System Clock Source Selection
This 1-bit power-on setting value is to configure if the system clock source is from crystal input directly or from output of internal PLL.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0x0</td><td>Crystal input is served as system clock</td></tr>
</table></td><td>CHIPCFG[6]</td></tr>
</table>

		CHIPCFG[5:4]
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.**	

Value	Description
	source.
0x1	PLL output is used as system clock source.

Value	Description
0x00	SDR SDRAM (normal SDRAM) is used.
0x01	Low Power DDR (mDDR) SDRAM is used.
0x10	DDR SDRAM is used.
0x11	DDR2 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 or test mode.** **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 input crystal is 12MHz or 27MHz.** **In test mode, this field will configure the chip to enter a specific test mode for internal circuit test. Such as SRAM and internal ROM BIST test, USB 2.0 transceiver test, Audio DAC & TV-DAC test and ADC test.** **This chip also implements a test control interface TIC for mass production test. In this test mode, CPU will be kept in reset state and all control registers could be accessed through TIC interface.**	CHIPCFG[3:0]

Value	Description
0x0000	Configure CHIP to enter ADC, Audio-DAC and TV-DAC test mode.
0x0001	Configure CHIP to enter Audio DAC and USB 1.1 transceiver test mode.
0x0010 0x0011 0x0011	EFUSE Program Mode
0x0100	Configure CHIP to enter SRAM & ROM BIST test mode
0x0101	Configure CHIP to enter USB 2.0 PHY test mode.
0x0110	Reserved
0x0111	Configure CHIP to enter TIC mode.
0x1000	Boot from IBR Recover Mode with crystal input is 27MHz & DRAM Speed=66MHz.
0x1001	Boot from IBR Normal Mode with crystal input is 27MHz & DRAM Speed=66MHz.
0x1010	Boot from IBR Recover Mode with crystal input is 27MHz & DRAM Speed=132MHz.
0x1011	Boot From IBR Normal Mode with crystal input is 27MHz & DRAM Speed=132MHz.
0x1100	Boot from IBR Recover Mode with crystal input is 12MHz & DRAM Speed=66MHz.
0x1101	Boot from IBR Normal Mode with crystal input is 12MHz & DRAM Speed=66MHz.
0x1110	Boot from IBR Recover Mode with crystal input is 12MHz & DRAM Speed=132MHz.
0x1111	Boot From IBR Normal Mode with crystal input is 12MHz & DRAM Speed=132MHz.

The following is the truth table for different chip operation mode described above:

Please note if any of the reserved setting is used, CHIP's behavior is undefined.

6.2.4 Power-On Setting Waveform

6.2.5 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, PRTMOD1 and PRTMOD2 of the *Arbitration Control Register*. These three fields control the different priority group respectively. Each priority includes several functional circuits. The mapping between these three fields and priority groups is as following table.

PRTMOD0, PRTMOD1 and PRTMOD2 are for AHB1, AHB3 and AHB4 bus arbitration control respectively.

Priority Modes	Priority Group
PRTMOD0	ARM926EJ-S Master I/F for Instruction (ARM926-I), ARM926EJ-S Master I/F for Data (ARM926-D) and PDMA.
PRTMOD1	BitBLT Accelerator, JPEG, SIC, UHC and UDC.
PRTMOD2	SPU, I2S Controller, Capture, and Display Controller.

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.

The bus arbiter also provides a mechanism to allow S/W to program the maximum burst length for each AHB bus transfer. When the current AHB data transfer count is equal to the maximum burst length, the access of current AHB bus owner will be broken. S/W can use this mechanism to adjust the AHB bus transfer for each engine.

In fixed priority mode of AHB-4 Bus, a programmable time-out is attached to any host except the Audio to ensure that it is not blocked indefinitely by higher priority hosts. When a low-priority master requests the arbitration (i.e., Display, Video Capture, or FMI), the associated time-out counter is loaded with the programmed value and starts decrementing. If a counter reaches 0, the associated master gets the highest priority for its next request (that may be already pending). If several requestors are in this situation, the priority order is: Display, Video Capture, and FMI.

6.2.5.1 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. W55FA95 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.

6.2.5.2 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.

6.2.5.3 Priority algorithm for AHB-4 DMA bus

Because there are many real time interfaces attached on AHB4-DMA bus, an algorithm, fixed priority with timeout control, is used for best bus utilization and bandwidth control. A programmable time-out is attached to any host except the Audio to ensure that it is not blocked indefinitely by higher priority hosts. When the master is requesting the ownership of the bus, the counter is decreased by 1 every 4 AHB4-HCLK clocks. While a counter reaches 0, the associated master gets the highest priority. If several requestors are in this situation, the priority order is: Display, Video Capture, Rotation Engine and SIC. If the counter value is set as 0, then the corresponding time-out scheme for that master is disabled.

6.2.5.4 Rotate rule Example:

In the default sequence of AHB-3 DMA Master bus, the priority is Audio > LCM > CAP > FMI_DMAC > USB> JPEG > Video Codec > IPA.

If the Video Codec has been granted, the next priority order will be .IPA > Audio > LCM > CAP > FMI_DMAC >

6.2.6 Power Management Mode

W55FA95 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. W55FA95 also implements power management bit to disabled all system clock simultaneously. Besides, by embedding a voltage-scalable LDO, the core power could be scaled down from 1.8V to 1.2V to save the power consumption. And, 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 four 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 interrupt, USB host connect/disconnect/remote-wakeup event, USB device connect/disconnect event and RTC alarm.

Power-Down Mode

In W55FA95, 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 register of RTC properly , the power of core power domain will be turned off and W55FA95 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 the register of RTC, the power

The comparison of four power consumption level is listed as following.

6.2.7 IBR (Internal Boot ROM) Sequence

6.2.8 System Management Control Registers

Register	Address	R/W	Description	Reset Value
GCR_BA = 0xB000_0000				
CHIPID	GCR_BA+0x00	R	Chip Identification Register	0x00FAD006
CHIPCFG	GCR_BA+0x04	R/W	Chip Power-On Configuration Register	0x0003_0XXX
OVCKCFG	GCR_BA+0x08	R/W	Overclk configuration register	0x0000_0000
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_0300
SDRBIST	GCR_BA+0x24	R/W	SDRAM BIST Test Status 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	0x00FF_00XX
POR_LVRD	GCR_BA+0x74	R/W	POR and LVRD Control Register	0x0000_00XX
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_00001
GPFFUN	GCR_BA+0x98	R/W	GPIO F Multi-function Control Register	0x0000_0000
MISCPCR	GCR_BA+0xA0	R/W	Miscellaneous Pins Control Register	0x0000_0000
MISC_SL_GPA	GCR_BA+0xA4	R/W	GPIO A Slew Rate control	0x0000_0000
MISC_SL_GPB	GCR_BA+0xA8	R/W	GPIO B Slew Rate control	0x0000_0000
MISC_SL_GPC	GCR_BA+0xAC	R/W	GPIO C Slew Rate control	0x0000_0000

MISC_SL_GPD	GCR_BA+0xB0	R/W	GPIO D Slew Rate control	0x0000_0000
MISC_SL_GPE	GCR_BA+0xB4	R/W	GPIO E Slew Rate control	0x0000_0000
MISC_SL_ND	GCR_BA+0xB8	R/W	ND PAD Slew Rate control	0x0000_0000
MISC_DS_GPA	GCR_BA+0xBC	R/W	GPIO A Driver Strength control	0x0000_0000
MISC_DS_GPB	GCR_BA+0xC0	R/W	GPIO A Driver Strength control	0x0000_0000
MISC_DS_GPC	GCR_BA+0xC4	R/W	GPIO A Driver Strength control	0x0000_0000
MISC_DS_GPD	GCR_BA+0xC8	R/W	GPIO A Driver Strength control	0x0000_0000
MISC_DS_GPE	GCR_BA+0xCC	R/W	GPIO A Driver Strength control	0x0000_0000
MISC_DS_ND	GCR_BA+0xD0	R/W	ND PAD Driver Strength control	0x0000_0000
MISC_SSEL	GCR_BA+0xD4	R/W	SSTL2 and LVTTL Driver Strength control	0x0000_0300
GPGFUN	GCR_BA+0xD8	R/W	GPIO G Multi-function Control Register	0x0000_0000
GPHFUN	GCR_BA+0xDC	R/W	GPIO H Multi-function Control Register	0x0000_0000
ShrPin_TVDAC	GCR_BA+0xF0	R/W	Share Pins with TVDAC	0x8XXX_XXXX
ShrPin_AUDIO	GCR_BA+0xF4	R/W	Share Pins with Audio ADC	0xFXXX_XXXX
ShrPin_TOUCH	GCR_BA+0xF8	R/W	Share Pins with Touch Panel ADC	0xEXXX_XXXX
ShrPin_R_FB	GCR_BA+0xFC	R/W	Share Pins with R_Divider	0x0000_010E

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_D006

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_D006" to indicate the chip product ID is "FAD006".

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	CLK_SRC	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. Value: 0x0 — Description: EF (Error Free) NAND flash memory Value: 0x1 — Description: Raw NAND flash memory
[9:8]	NPAGE	NAND Page Size

<table>
<tr><td></td><td></td><td>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></td></tr>
<tr><td>[7]</td><td>NADDR</td><td>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></td></tr>
<tr><td>[6]</td><td>CLK_SRC</td><td>System Clock Source Selection
This bit reflects the power-on setting value about the system clock source is from crystal input directly or from output of internal PLL.
1′b0: Crystal input is served as system clock source.
1′b1: PLL output is used as system clock source.</td></tr>
<tr><td>[5:4]</td><td>SDRAMSEL</td><td>SDRAM Type Selection
This field reflects the power-on setting value about which type of SDRAM is used for system memory.
Ÿ 2′b00: SDR SDRAM (normal SDRAM) is used.
Ÿ 2′b01: Low Power DDR (mDDR) SDRAM is used.
Ÿ 2′b10: DDR SDRAM is used.
Ÿ 2′b11: DDR2 SDRAM is used.</td></tr>
</table>

<table>
<tr>
<td>[3:0]</td>
<td>COPMODE</td>
<td>
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>Configure CHIP to enter ADC, Audio-DAC and TV-DAC test mode.</td></tr>
<tr><td>0x0001</td><td>Configure CHIP to enter Audio DAC and USB 1.1 transceiver test mode.</td></tr>
<tr><td>0x0010
0x0011</td><td>EFUSE Program Mode</td></tr>
<tr><td>0x0011</td><td></td></tr>
<tr><td>0x0100</td><td>Configure CHIP to enter SRAM & ROM BIST test mode</td></tr>
<tr><td>0x0101</td><td>Configure CHIP to enter USB 2.0 PHY test mode.</td></tr>
<tr><td>0x0110</td><td></td></tr>
<tr><td>0x0111</td><td>Configure CHIP to enter TIC mode.</td></tr>
<tr><td>0x1000</td><td>Boot from IBR Recover Mode with crystal input is 27MHz & DRAM Speed=66MHz.</td></tr>
<tr><td>0x1001</td><td>Boot from IBR Normal Mode with crystal input is 27MHz & DRAM Speed=66MHz.</td></tr>
<tr><td>0x1010</td><td>Boot from IBR Recover Mode with crystal input is 27MHz & DRAM Speed=132MHz.</td></tr>
<tr><td>0x1011</td><td>Boot From IBR Normal Mode with crystal input is 27MHz & DRAM Speed=132MHz.</td></tr>
<tr><td>0x1100</td><td>Boot from IBR Recover Mode with crystal input is 12MHz & DRAM Speed=66MHz.</td></tr>
<tr><td>0x1101</td><td>Boot from IBR Normal Mode with crystal input is 12MHz & DRAM Speed=66MHz.</td></tr>
<tr><td>0x1110</td><td>Boot from IBR Recover Mode with crystal input is 12MHz & DRAM Speed=132MHz.</td></tr>
<tr><td>0x1111</td><td>Boot From IBR Normal Mode with crystal input is 12MHz & DRAM Speed=132MHz.</td></tr>
</table>
In SRAM & ROM BIST test mode, the chip will do BIST test for internal 8KB SRAM, for embedded SRAM of each function, for cache RAM of ARM926EJ-S and for internal 16KB boot ROM.
</td>
</tr>
</table>

There is a 16-bit EFUSE built in FA95, following is the proposed bit definition of each bit.

Bit number	Description
Bit[15:8]	Software definition
Bit[7:4]	AES Key Selection
Bit[3]	
Bit[2]	
Bit[1]	Turn on OVG
Bit[0]	Turn on CNM

Note:

Default bit status is "1" if it is not programmed, the programming process will make the bit status to be "0".

Ÿ1: On

Ÿ0:Off

Overclk Configuration Register (OVCKCFG)

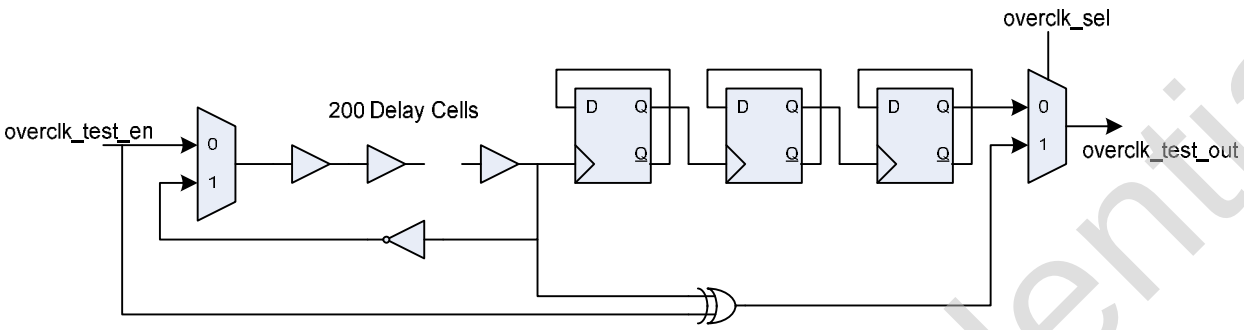
This register provides information for software to overclk setting. Bits [7:0] are the counter of the ring oscillator pulse width in GSPLL clock domain. These configuration bits could be modified by software programming.

Register	Address	R/W	Description	Reset Value
OVCKCFG	GCR_BA+0x08	R/W	Overclk Configuration Register	0x0000_0000

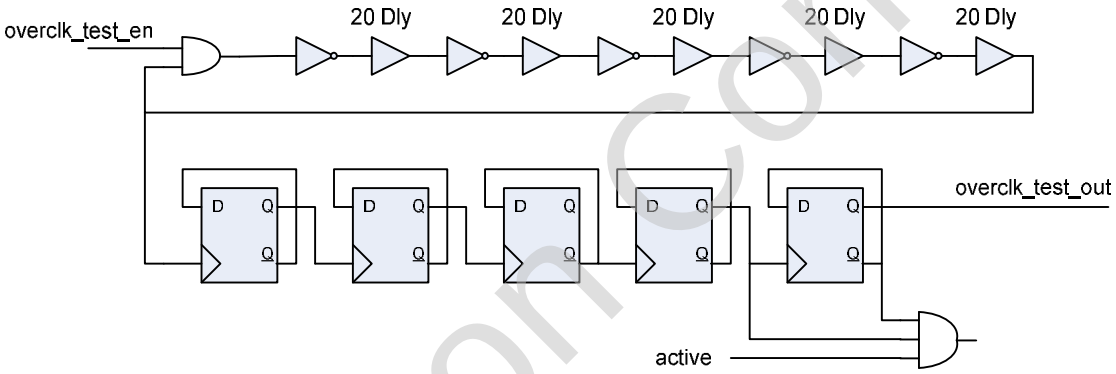
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					OVCK_SET[2:0]		
7	6	5	4	3	2	1	0
OVCK_CNT[7:0]							

Bits	Descriptions	
[31:11]	Reserved	Reserved
[10]	OVCK_SET	OVCK_CNT output select 0: output from overclk_dff circuit 1: output from overclk_div circuit
[9]	OVCK_SET	Overclk_dff output select 0: output from xor output 1: output from divider output
[8]	OVCK_SET	Overclk_circuit enable 0: disable (default) 1: enable
[7:0]	OVCK_CNT	Overclk circuit counting result (Read Only)

[Overclk_dff]



[Overclk_div]



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
[4]	IPEN	CPU Priority Raising Enable during Interrupt Period
[3]	Reserved	Reserved
[2]	Reserved	Reserved
[1]	PRTMOD1	Priority Mode Control 1
[0]	PRTMOD0	Priority Mode Control 0

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			GVESW_RST	VPERST	VIN1RST	JPGRST	Reserved
15	14	13	12	11	10	9	8
VDERST	IPSECRST	GERST	OVGRST	VIN0RST	VPOSRTST	I2SRST	SPURST
7	6	5	4	3	2	1	0
UHCRST	UDCRST	SICRST	TICRST	EDMARST	SRAMRST	Reserved	SDICRST

Bits	Descriptions	
[31:21]	Reserved	Reserved
[20]	GVESW_RST	Graphics Video Switch Engine Reset 1′b0: Graphics Video Switch Engine reset is no active 1′b1: Graphics Video Switch Engine reset is active
[19]	VPERST	Video Engine Reset 1′b0: Video Engine reset is no active 1′b1: Video Engine reset is active
[18]	VIN1RST	Video In1 Reset 1′b0: Video In1 reset is no active. 1′b1: Video In1 reset is active.
[17]	JPGRST	JPEG Reset 1′b0: JPEG reset is no active. 1′b1: JPEG reset is active.
[16]	Reserved	Reserved

[15]	VDERST	Video Decoder Reset 1'b0: Video Decoder reset is no active 1'b1: Video Decoder reset is active
[14]	IPSECRST	AES Engine Reset 1'b0: AES reset is no active 1'b1: AES reset is active
[13]	GERST	Graphics Engine Reset 1'b0: Graphics Engine reset is no active 1'b1: Graphics Engine reset is active
[12]	OVGRST	Open VG Reset 1'b0: Open VG reset is no active 1'b1: Open VG reset is active
[11]	VIN0RST	Video In0 Reset 1'b0: Video In0 reset is no active. 1'b1: Video In0 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]	UHCRST	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
Reserved	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
[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]	GPIORST	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_0300

31	30	29	28	27	26	25	24
Reserved							WDTRSTEN
23	22	21	20	19	18	17	16
UTMISnoop	Reserved			SEL_HSCUR		SEL_PHASE	
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.
[23]	UTMISnoop	UTMI Monitor Mode Enable This bit is to enable the UTMI monitor mode to snoop the UTMI I/F signals between USB 2.0 device controller and USB 2.0 PHY during normal operation. Please note some other functional pins will be dis-functioned if this bit is enabled. 1′b0: UTMI monitor mode disabled. (Default) 1′b1: UTMI monitor mode enabled.
[22:20]	Reserved	Reserved

[19:18]	SEL_HSCUR	USB 2.0 PHY Control SEL_HSCUR
[17:16]	SEL_PHASE	USB 2.0 PHY Control SEL_PHASE
[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 W55FA95 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 W55FA95 would never be affected by low voltage reset function. While this bit is set high (1′b1), W55FA95 would affected by low voltage reset function. 1′b0: Low Voltage Reset function is gated and would never affect the W55FA95. 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 *LVR_RDY* in low. Or the W55FA95 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 *LVR_RDY* is still set as '1′. Otherwise, an immediate reset follows the ill setting. And be sure to enable *LVR_RDY* bit after *LVD_EN* 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.

SDRAM BIST Test Status Register (SDRBIST)

Register	Address	R/W	Description	Reset Value
SDRBIST	GCR_BA+0x24	R	SDRAM BIST Test Status Register	0x0000_0000

31	30	29	28	27	26	25	24
TEST_BUSY	CON_BUYS	BIST_BUSY	TEST_FAIL	CON_FAIL	BIST_FAIL	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]	TEST_BUSY	Test BUSY This bit indicates the SDRAM test is on going or finished. The test includes connection test and SDRAM BIST test. 0 = Test finished. 1 = Test is on going.
[30]	CON_BUSY	Connection Test Busy This bit indicates the connection test is on going or finished. 0 = Connection test finished. 1 = Connection test is on going.
[29]	BIST_BUSY	BIST Test Busy This bit indicates the SDRAM BIST test is on going or finished. 0 = SDRAM BIST test finished. 1 = SDRAM BIST test is on going.

[28]	TEST_FAIL	**Test Failed** This bit indicates if the SDRAM test failed or succeeded. The test includes connection test and SDRAM BIST test. User checks this bit after the TEST_BUSY is 0. 0: Test is OK. 1: Test failed. The fail may be connection test fail or SDRAM BIST test fail. The test stopped while the first error occurred.
[27]	CON_FAIL	**Connection Test Failed** This bit indicates if the connection (CKE, CS_ and DQM) test failed or succeeded. User checks this bit after the CON_BUSY is 0. 0: Connection test is OK. 1: Connection test failed.
[26]	BIST_FAIL	**BIST Test Failed** This bit indicates if the SDRAM BIST test failed or succeeded. User checks this bit after the BIST_BUSY is 0. The first checked error is record on registers TFADDR and TFDATA. 0: SDRAM BIST test is OK. 1: SDRAM BIST test failed.
[25:0]	Reserved	Reserved

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 *ICV_R* 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 *ICT_R* 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 *ICD3_R* 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 *ICD2_R* 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 *ICD1_R* 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 *ICD0_R* 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 *MMU_R* 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 *DCDIR_R* 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 *DCV_R* 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 *DCT_R* 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 *DCD3_R* 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 *DCD2_R* 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 *DCD1_R* 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 *DCD0_R* 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 *BISTEN* 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 ***BISTEN*** 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 filed defined EDMA channel 1 is to service which on-chip peripherals. If this filed 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	0x00FF_00XX

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
		POR12_RST	CPU_RST	WDT_RST	KPI_RST	LVR_RST	EXT_RST

Bits	Descriptions	
[31]	KPI_WS	KPI Wake-Up Status
[30]	ADC_WS	ADC Wake-Up Status
[29]	UHC_WS	UHC Wake-Up Status
[28]	UDC_WS	UDC Wake-Up Status
[27]	UART_WS	UART Wake-Up Status
[26]	SDH_WS	SD Controller Wake-Up Status
[25]	RTC_WS	RTC Wake-Up Status
[24]	GPIO_WS	GPIO Wake-Up Status
[23]	KPI_WE	KPI Wake-Up Enable
[22]	ADC_WE	ADC Wake-Up Enable
[21]	UHC_WE	UHC Wake-Up Enable
[20]	UDC_WE	UDC Wake-Up Enable
[19]	UART_WE	UART Wake-Up Enable
[18]	SDH_WE	SD Controller Wake-Up Enable

[17]	RTC_WE	RTC Wake-Up Enable
[16]	GPIO_WE	GPIO Wake-Up Enable
[15:6]	Reserved	Reserved
[5]	POR12_RST	POR12 Reset Active Status
[4]	CPU_RST	CPU Reset Active Status
[3]	WDT_RST	WDT Reset Active Status
[2]	KPI_RST	KPI Reset Active Status
[1]	LVR_RST	LVR Reset Active Status
[0]	EXT_RST	External Reset Pin Active Status

POR and LVRD Control Register (POR_LVRD)

This register defines the control function description for POR and LVRD

Register	Address	R/W	Description	Reset Value
POR_LVRD	GCR_BA+0x74	R/W	POR and LVRD Control 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							
7	6	5	4	3	2	1	0
Reserved	UPLL_LKDT	APLL_LKDT	POR_ENB	EN_LVR	EN_LVD	LVD_SEL	LVD_OUTB

Bits	Descriptions	
[31:5]	Reserved	Reserved This field is reserved and keep all these bits in zero is necessary.
[6]	UPLL_LKDT	UPLL lock status 0: UPLL isn't locked 1: UPLL is locked
[5]	APLL_LKDT	APLL lock status 0: APLL isn't locked 1: APLL is locked
[4]	POR_ENB	Power on circuit enable bar: active low (default low)
[3]	EN_LVR	LVR enable: active high (default high)
[2]	EN_LVD	LVD enable: active high (default low)
[1]	LVD_SEL	LVD select 0: 2.6V detect @3.0V mode, 1: 2.8V detect @3.3V mode,

[0]	LVD_OUTB	LVD flag It is read only.

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
MF_GPA15		MF_GPA14		MF_GPA13		MF_GPA12	
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	
[31:30]	MF_GPA15	GPIOA[15] Multi-function

Pin Name	MF_GPA15			
ND[7]	2′b00	2′b01	2′b10	2′b11
	Reserved	SPI0_CS0	GPIOA[15]	Reserved

[29:28]	MF_GPA14	GPIOA[14] Multi-function

Pin Name	MF_GPA14			
ND[6]	2′b00	2′b01	2′b10	2′b11
	Reserved	SPI0_CLK	GPIOA[14]	Reserved

[27:26]	MF_GPA13	GPIOA[13] Multi-function MF_GPF[27:26] must be 2′b00

Pin Name	MF_GPA13			
ND[5]	2′b00	2′b01	2′b10	2′b11
	Reserved	SPI0_DI	GPIOA[13]	Reserved

<table>
<tr><td>[25:24]</td><td>MF_GPA12</td><td>GPIOA[12] Multi-function
MF_GPF[25:24] must be 2′b00
<table><tr><td>Pin Name</td><td colspan="4">MF_GPA12</td></tr><tr><td rowspan="2">ND[4]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>Reserved</td><td>SPI0_DO</td><td>GPIOA[12]</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[23:22]</td><td>MF_GPA11</td><td>GPIOA[11] Multi-function
MF_GPF[31:30] must be 2′b00
<table><tr><td>Pin Name</td><td colspan="4">MF_GPA11</td></tr><tr><td rowspan="2">URRXD</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[11]</td><td>LMVSYNC</td><td>TIC_DI[1]</td><td>URRXD</td></tr></table></td></tr>
<tr><td>[21:20]</td><td>MF_GPA10</td><td>GPIOA[10] Multi-function
MF_GPF[29:28] must be 2′b00
<table><tr><td>Pin Name</td><td colspan="4">MF_GPA10</td></tr><tr><td rowspan="2">URTXD</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[10]</td><td>SPI1_CS1_</td><td>TIC_DO[1]</td><td>URTXD</td></tr></table></td></tr>
<tr><td>[19:18]</td><td>MF_GPA9</td><td>GPIOA[9] Multi-function
<table><tr><td>Pin Name</td><td colspan="4">MF_GPA9</td></tr><tr><td rowspan="2">GPA[9]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[9]</td><td>Reserved</td><td>ISDA(GPA[11])</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[17:16]</td><td>MF_GPA8</td><td>GPIOA[8] Multi-function
<table><tr><td>Pin Name</td><td colspan="4">MF_GPA8</td></tr><tr><td rowspan="2">GPA[8]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[8]</td><td>Reserved</td><td>ISCK(GPA[10])</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[15:14]</td><td>MF_GPA7</td><td>GPIOA[7] Multi-function
<table><tr><td>Pin Name</td><td colspan="4">MF_GPA7</td></tr><tr><td rowspan="2">GPA[7]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[7]</td><td>Reserved</td><td>WDT_RST_</td><td>WDT_RST_</td></tr></table></td></tr>
</table>

<table>
<tr><td>[13:12]</td><td>MF_GPA6</td><td>GPIOA[6] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA6</td></tr><tr><td rowspan="2">GPA[6]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[6]</td><td>KPI_SI[3]</td><td>Reserved</td><td>UHL_DM0</td></tr></table></td></tr>
<tr><td>[11:10]</td><td>MF_GPA5</td><td>GPIOA[5] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA5</td></tr><tr><td rowspan="2">GPA[5]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[5]</td><td>KPI_SI[2]</td><td>Reserved</td><td>UHL_DP0</td></tr></table></td></tr>
<tr><td>[9:8]</td><td>MF_GPA4</td><td>GPIOA[4] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA4</td></tr><tr><td rowspan="2">GPA[4]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[4]</td><td>KPI_SI[1]</td><td>SPI0_CS1_</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[7:6]</td><td>MF_GPA3</td><td>GPIOA[3] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA3</td></tr><tr><td rowspan="2">GPA[3]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[3]</td><td>KPI_SI[0]</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[5:4]</td><td>MF_GPA2</td><td>GPIOA[2] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA2</td></tr><tr><td rowspan="2">GPA[2]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[2]</td><td>Reserved</td><td>LMVSYNC</td><td>LMVSYNC</td></tr></table></td></tr>
<tr><td>[3:2]</td><td>MF_GPA1</td><td>GPIOA[1] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA1</td></tr><tr><td rowspan="2">GPA[1]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[1]</td><td>SCLKO_2</td><td>SD_CD_</td><td>SD_CD_</td></tr></table></td></tr>
<tr><td>[1:0]</td><td>MF_GPA0</td><td>GPIOA[0] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPA0</td></tr><tr><td rowspan="2">GPA[0]</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOA[0]</td><td>SPCLK_2</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
</table>

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	2'b11
	GPIOB[15]	Reserved	LPCLK	LPCLK

Bits	Descriptions	
[29:28]	MF_GPB14	GPIOB[14] Multi-function

Pin Name	MF_GPB14			
ISDA	2'b00	2'b01	2'b10	2'b11
	GPIOB[14]	LMVSYNC	ISDA	ISDA

Bits	Descriptions	
[27:26]	MF_GPB13	GPIOB[13] Multi-function

Pin Name	MF_GPB13			
ISCK	2'b00	2'b01	2'b10	2'b11
	GPIOB[13]	Reserved	ISCK	ISCK

[25:24]	MF_GPB12	GPIOB[12] Multi-function

Pin Name	MF_GPB12			
SPDATA[7]	2'b00	2'b01	2'b10	2'b11
	GPIOB[12]	SPI1_DO	LVDATA[23]	SPDATA[7]

[23:22]	MF_GPB11	GPIOB[11] Multi-function

Pin Name	MF_GPB11			
SPDATA[6]	2'b00	2'b01	2'b10	2'b11
	GPIOB[11]	SPI1_DI	LVDATA[22]	SPDATA[6]

[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
[9:8]	MF_GPB4	GPIOB[4] Multi-function
[7:6]	MF_GPB3	GPIOB[3] Multi-function
[5:4]	MF_GPB2	GPIOB[2] Multi-function
[3:2]	MF_GPB1	GPIOB[1] Multi-function
[1:0]	MF_GPB0	GPIOB[0] Multi-function

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]

GPIOB[4] Multi-function

Pin Name	MF_GPB4			
SFIELD	2'b00	2'b01	2'b10	2'b11
	GPIOB[4]	I2S_WS	SDDAT1[3]	SFIELD

GPIOB[3] Multi-function

Pin Name	MF_GPB3			
SVSYNC	2'b00	2'b01	2'b10	2'b11
	GPIOB[3]	I2S_BCLK	SDCMD1	SVSYNC

GPIOB[2] Multi-function

Pin Name	MF_GPB2			
SHSYNC	2'b00	2'b01	2'b10	2'b11
	GPIOB[2]	I2S_MCLK	SDCLK1	SHSYNC

GPIOB[1] Multi-function

Pin Name	MF_GPB1			
SPCLK	2'b00	2'b01	2'b10	2'b11
	GPIOB[1]	UHL_DM1	SDDAT1[0]	SPCLK

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	2'b11
	GPIOC[15]	SPDATA[7]	LVDATA[15]	KPI_SO[15]

Bits	Descriptions	
[29:28]	MF_GPC14	GPIOC[14] Multi-function

Pin Name	MF_GPC14			
LVDATA[14]	2'b00	2'b01	2'b10	2'b11
	GPIOC[14]	SPDATA[6]	LVDATA[14]	KPI_SO[14]

Bits	Descriptions	
[27:26]	MF_GPC13	GPIOC[13] Multi-function

Pin Name	MF_GPC13			
LVDATA[13]	2'b00	2'b01	2'b10	2'b11
	GPIOC[13]	SPDATA[5]	LVDATA[13]	KPI_SO[13]

[25:24]	MF_GPC12	GPIOC[12] Multi-function

Pin Name	MF_GPC12			
LVDATA[12]	2'b00	2'b01	2'b10	2'b11
	GPIOC[12]	SPDATA[4]	LVDATA[12]	KPI_SO[12]

[23:22]	MF_GPC11	GPIOC[11] Multi-function

Pin Name	MF_GPC11			
LVDATA[11]	2'b00	2'b01	2'b10	2'b11
	GPIOC[11]	SPDATA[3]	LVDATA[11]	KPI_SO[11]

[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]	LVDATA[10]	KPI_SO[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]	LVDATA[9]	KPI_SO[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]	LVDATA[8]	KPI_SO[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]	KPI_SO[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]	KPI_SO[6]

[11:10]	MF_GPC5	GPIOC[5] Multi-function
[9:8]	MF_GPC4	GPIOC[4] Multi-function
[7:6]	MF_GPC3	GPIOC[3] Multi-function
[5:4]	MF_GPC2	GPIOC[2] Multi-function
[3:2]	MF_GPC1	GPIOC[1] Multi-function
[1:0]	MF_GPC0	GPIOC[0] Multi-function

Pin Name	MF_GPC5			
LVDATA[5]	2'b00	2'b01	2'b10	2'b11
	GPIOC[5]	SDRM_BFAIL	LVDATA[5]	KPI_SO[5]

Pin Name	MF_GPC4			
LVDATA[4]	2'b00	2'b01	2'b10	2'b11
	GPIOC[4]	SDRM_CFAIL	LVDATA[4]	KPI_SO[4]

Pin Name	MF_GPC3			
LVDATA[3]	2'b00	2'b01	2'b10	2'b11
	GPIOC[3]	SDRM_TFAIL	LVDATA[3]	KPI_SO[3]

Pin Name	MF_GPC2			
LVDATA[2]	2'b00	2'b01	2'b10	2'b11
	GPIOC[2]	SDRM_BBUSY	LVDATA[2]	KPI_SO[2]

Pin Name	MF_GPC1			
LVDATA[1]	2'b00	2'b01	2'b10	2'b11
	GPIOC[1]	SDRM_CBUSY	LVDATA[1]	KPI_SO[1]

Pin Name	MF_GPC0			
LVDATA[0]	2'b00	2'b01	2'b10	2'b11
	GPIOC[0]	SDRM_TBUSY	LVDATA[0]	KPI_SO[0]

GPIO D Multi-function Control Register (GPDFUN)

Note: The pull-up enable of the pins shared with GPIOD is controlled by register GPIOD_PUEN directly. Users have to set the GPIOD_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	GPIOD[15] Multi-function

Pin Name	MF_GPD15			
SPI0_DO	2'b00	2'b01	2'b10	2'b11
	GPIOD[15]	TIC_DO[0]	SPI0_DO	UHL_DM0

Bits	Descriptions	
[29:28]	MF_GPD14	GPIOD[14] Multi-function

Pin Name	MF_GPD14			
SPI0_DI	2'b00	2'b01	2'b10	2'b11
	GPIOD[14]	TIC_DI[0]	SPI0_DI	UHL_DP0

Bits	Descriptions	
[27:26]	MF_GPD13	GPIOD[13] Multi-function

Pin Name	MF_GPD13			
SPI0_CS0_	2'b00	2'b01	2'b10	2'b11
	GPIOD[13]	TIC_CS_	SPI0_CS0_	SPI0_CS0_

[25:24]	MF_GPD12	GPIOD[12] Multi-function

Pin Name	MF_GPD12			
SPI0_CLK	2'b00	2'b01	2'b10	2'b11
	GPIOD[12]	TIC_CLKI	SPI0_CLK	SPI0_CLK

[23:22]	MF_GPD11	GPIOD[11] Multi-function

Pin Name	MF_GPD11			
LVDE	2'b00	2'b01	2'b10	2'b11
	GPIOD[11]	WDT_RST_	LVDE	LVDE

[21:20]	MF_GPD10	GPIOD[10] Multi-function

Pin Name	MF_GPD10			
LVSYNC	2'b00	2'b01	2'b10	2'b11
	GPIOD[10]	Reserved	LVSYNC	LVSYNC

[19:18]	MF_GPD9	GPIOD[9] Multi-function

Pin Name	MF_GPD9			
LHSYNC	2'b00	2'b01	2'b10	2'b11
	GPIOD[9]	Reserved	LHSYNC	LHSYNC

[17:16]	MF_GPD8	GPIOD[8] Multi-function

Pin Name	MF_GPD8			
NWR_	2'b00	2'b01	2'b10	2'b11
	GPIOD[8]	SDCMD2	NWR_	NWR_

[15:14]	MF_GPD7	GPIOD[7] Multi-function

Pin Name	MF_GPD7			
NRE_	2'b00	2'b01	2'b10	2'b11
	GPIOD[7]	SDCLK2	NRE_	NRE_

[13:12]	MF_GPD6	GPIOD[6] Multi-function

Pin Name	MF_GPD6			
NBUSY1_	2'b00	2'b01	2'b10	2'b11
	GPIOD[6]		NBUSY1_	NBUSY1_

<table>
<tr><td>[11:10]</td><td>MF_GPD5</td><td>GPIOD[5] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPD5</td></tr><tr><td rowspan="2">NBUSY0_</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOD[5]</td><td>SDDAT2[2]</td><td>NBUSY0_</td><td>NBUSY0_</td></tr></table></td></tr>
<tr><td>[9:8]</td><td>MF_GPD4</td><td>GPIOD[4] Multi-function
MF_GPF[23:22] must be 2′b00<table><tr><td>Pin Name</td><td colspan="4">MF_GPD4</td></tr><tr><td rowspan="2">TRST_</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOD[4]</td><td>HUR_RTS</td><td>SPI0_CS1_</td><td>TRST_</td></tr></table></td></tr>
<tr><td>[7:6]</td><td>MF_GPD3</td><td>GPIOD[3] Multi-function
MF_GPF[21:20] must be 2′b00<table><tr><td>Pin Name</td><td colspan="4">MF_GPD3</td></tr><tr><td rowspan="2">TDO</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOD[3]</td><td>HUR_CTS</td><td>PWM3</td><td>TDO</td></tr></table></td></tr>
<tr><td>[5:4]</td><td>MF_GPD2</td><td>GPIOD[2] Multi-function
MF_GPF[19:18] must be 2′b00<table><tr><td>Pin Name</td><td colspan="4">MF_GPD2</td></tr><tr><td rowspan="2">TDI</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOD[2]</td><td>HUR_RXD</td><td>PWM2</td><td>TDI</td></tr></table></td></tr>
<tr><td>[3:2]</td><td>MF_GPD1</td><td>GPIOD[1] Multi-function
MF_GPF[17:16] must be 2′b00<table><tr><td>Pin Name</td><td colspan="4">MF_GPD1</td></tr><tr><td rowspan="2">TMS</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOD[1]</td><td>HUR_TXD</td><td>PWM1</td><td>TMS</td></tr></table></td></tr>
<tr><td>[1:0]</td><td>MF_GPD0</td><td>GPIOD[0] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPD0</td></tr><tr><td rowspan="2">TCK</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOD[0]</td><td>SPI1_CS1_</td><td>PWM0</td><td>TCK</td></tr></table></td></tr>
</table>

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		MF_GPE13		MF_GPE12	
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	
[31:26]	Reserved	Reserved This field is reserved and keep all these bits in zero is necessary.
[27:26]	MF_GPE13	GPIOE[13] Multi-function

Pin Name	MF_GPE11			
Reserved	2'b00	2'b01	2'b10	2'b11
	Reserved	SFIELD(GPA[11])	Reserved	Reserved

[25:24]	MF_GPE12	GPIOE[12] Multi-function

Pin Name	MF_GPE11			
Reserved	2'b00	2'b01	2'b10	2'b11
	Reserved	SDDAT2[3](ND[3])	Reserved	Reserved

[23:22]	MF_GPE11	GPIOE[11] Multi-function

Pin Name	MF_GPE11			
NCLE	2'b00	2'b01	2'b10	2'b11
	GPIOE[11]	SDDAT2[1]	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]	SDDAT2[0]	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]	Reserved	NCS1_	NCS1_

[17:16]	MF_GPE8	GPIOE[8] Multi-function

Pin Name	MF_GPE8			
NCS0_	2'b00	2'b01	2'b10	2'b11
	GPIOE[8]	Reserved	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]	TIC_DO[3]	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]	TIC_DO[2]	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]	TIC_DI[3]	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]	TIC_DI[2]	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]	LVD_OUT

[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

GPIO F Multi-function Control Register (GPFFUN)

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

Register	Address	R/W	Description	Reset Value
GPFFUN	GCR_BA+0x98	R/W	GPIO F Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MF_GPF15		MF_GPF14		MF_GPF13		MF_GPF12	
23	22	21	20	19	18	17	16
MF_GPF11		MF_GPF10		MF_GPF9		MF_GPF8	
15	14	13	12	11	10	9	8
MF_GPF7		MF_GPF6		MF_GPF5		MF_GPF4	
7	6	5	4	3	2	1	0
MF_GPF3		MF_GPF2		MF_GPF1		MF_GPF0	

Bits	Descriptions	
[31:30]	MF_GPF15	GPIOF[15] Multi-function

Pin Name	MF_GPF15			
	2'b00	2'b01	2'b10	2'b11
	Reserved	UHL_DM1(GPA[11])	Reserved	Reserved

Bits	Descriptions	
[29:28]	MF_GPF14	GPIOF[14] Multi-function

Pin Name	MF_GPF14			
	2'b00	2'b01	2'b10	2'b11
	Reserved	UHL_DP1(GPA[10])	Reserved	Reserved

Bits	Descriptions	
[27:26]	MF_GPF13	GPIOF[13] Multi-function

Pin Name	MF_GPF13			
	2'b00	2'b01	2'b10	2'b11
	Reserved	UHL_DM0(ND[5])	Reserved	Reserved

<table>
<tr><td>[25:24]</td><td>MF_GPF12</td><td>GPIOF[12] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF12</td></tr><tr><td rowspan="2">TRST_</td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>UHL_DP0(ND[4])</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[23:22]</td><td>MF_GPF11</td><td>GPIOF[11] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF11</td></tr><tr><td rowspan="2">TDO</td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>TIC_DO</td><td>SHSYNC_2</td><td>UHL_DM0(GPD[4])</td></tr></table></td></tr>
<tr><td>[21:20]</td><td>MF_GPF10</td><td>GPIOF[10] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF10</td></tr><tr><td rowspan="2">TDI</td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>TIC_DI</td><td>SVSYNC_2</td><td>UHL_DP0(GPD[3])</td></tr></table></td></tr>
<tr><td>[19:18]</td><td>MF_GPF9</td><td>GPIOF[9] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF9</td></tr><tr><td rowspan="2">TMS</td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>TIC_CS</td><td>SCLKO_2</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[17:16]</td><td>MF_GPF8</td><td>GPIOF[8] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF8</td></tr><tr><td rowspan="2"></td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>TIC_CLK</td><td>SPCLK_2</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[15:14]</td><td>MF_GPF7</td><td>GPIOF[7] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF7</td></tr><tr><td rowspan="2"></td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[13:12]</td><td>MF_GPF6</td><td>GPIOF[6] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPF6</td></tr><tr><td rowspan="2"></td><td>2'b00</td><td>2'b01</td><td>2'b10</td><td>2'b11</td></tr><tr><td>Reserved</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
</table>

[11:10]	MF_GPF5	GPIOF[5] Multi-function Pin Name: MF_GPF5 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[9:8]	MF_GPF4	GPIOF[4] Multi-function Pin Name: MF_GPF4 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[7:6]	MF_GPF3	GPIOF[3] Multi-function Pin Name: MF_GPF3 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[5:4]	MF_GPF2	GPIOF[2] Multi-function Pin Name: MF_GPF2 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[3:2]	MF_GPF1	GPIOF[1] Multi-function Pin Name: MF_GPF1 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[1:0]	MF_GPF0	GPIOF[0] Multi-function Pin Name: MF_GPF0 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved

Miscellaneous Pins Control Register (MISCPCR)

Register	Address	R/W	Description	Reset Value
MISCPCR	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).

GPIO A Slew Rate Control (MISC_SL_GPA)

Register	Address	R/W	Description	Reset Value
MISC_SL_GPA	GCR_BA+0xA4	R/W	GPIO A Slew Rate 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				SL_GPA[11:0]			
7	6	5	4	3	2	1	0
SL_GPA[11:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	SL_GPA[11]	GPA[11] Pin Slew Rate Control This bit control the output slew rate of GPA[11] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[10]	SL_GPA[10]	GPA[10] Pin Slew Rate Control This bit control the output slew rate of GPA[10] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[9]	SL_GPA[9]	GPA[9] Pin Slew Rate Control This bit control the output slew rate of GPA[9] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[8]	SL_GPA[8]	GPA[8] Pin Slew Rate Control This bit control the output slew rate of GPA[8] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[7]	SL_GPA[7]	GPA[7] Pin Slew Rate Control This bit control the output slew rate of GPA[7] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[6]	SL_GPA[6]	GPA[6] Pin Slew Rate Control This bit control the output slew rate of GPA[6] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[5]	SL_GPA[5]	GPA[5] Pin Slew Rate Control This bit control the output slew rate of GPA[5] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[4]	SL_GPA[4]	GPA[4] Pin Slew Rate Control This bit control the output slew rate of GPA[4] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[3]	SL_GPA[3]	GPA[3] Pin Slew Rate Control This bit control the output slew rate of GPA[3] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[2]	SL_GPA[2]	GPA[2] Pin Slew Rate Control This bit control the output slew rate of GPA[2] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[1]	SL_GPA[1]	GPA[1] Pin Slew Rate Control This bit control the output slew rate of GPA[1] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[0]	SL_GPA[0]	GPA[0] Pin Slew Rate Control This bit control the output slew rate of GPA[0] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

GPIO B Slew Rate Control (MISC_SL_GPB)

Register	Address	R/W	Description	Reset Value
MISC_SL_GPB	GCR_BA+0xA8	R/W	GPIO B Slew Rate 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
SL_GPB[15:0]							
7	6	5	4	3	2	1	0
SL_GPB[15:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	SL_GPB[15]	GPB[15] Pin Slew Rate Control This bit control the output slew rate of GPB[15] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[14]	SL_GPB[14]	GPB[14] Pin Slew Rate Control This bit control the output slew rate of GPB[14] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[13]	SL_GPB[13]	GPB[13] Pin Slew Rate Control This bit control the output slew rate of GPB[13] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[12]	SL_GPB[12]	GPB[12] Pin Slew Rate Control This bit control the output slew rate of GPB[12] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[11]	SL_GPB[11]	GPB[11] Pin Slew Rate Control This bit control the output slew rate of GPB[11] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[10]	SL_GPB[10]	GPB[10] Pin Slew Rate Control This bit control the output slew rate of GPB[10] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[9]	SL_GPB[9]	GPB[9] Pin Slew Rate Control This bit control the output slew rate of GPB[9] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[8]	SL_GPB[8]	GPB[8] Pin Slew Rate Control This bit control the output slew rate of GPB[8] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[7]	SL_GPB[7]	GPB[7] Pin Slew Rate Control This bit control the output slew rate of GPB[7] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[6]	SL_GPB[6]	GPB[6] Pin Slew Rate Control This bit control the output slew rate of GPB[6] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[5]	SL_GPB[5]	GPB[5] Pin Slew Rate Control This bit control the output slew rate of GPB[5] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[4]	SL_GPB[4]	GPB[4] Pin Slew Rate Control This bit control the output slew rate of GPB[4] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[3]	SL_GPB[3]	GPB[3] Pin Slew Rate Control This bit control the output slew rate of GPB[3] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[2]	SL_GPB[2]	GPB[2] Pin Slew Rate Control This bit control the output slew rate of GPB[2] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[1]	SL_GPB[1]	GPB[1] Pin Slew Rate Control This bit control the output slew rate of GPB[1] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[0]	SL_GPB[0]	GPB[0] Pin Slew Rate Control This bit control the output slew rate of GPB[0] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

GPIO C Slew Rate Control (MISC_SL_GPC)

Register	Address	R/W	Description	Reset Value
MISC_SL_GPC	GCR_BA+0xAC	R/W	GPIO C Slew Rate 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
SL_GPC[15:0]							
7	6	5	4	3	2	1	0
SL_GPC[15:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	SL_GPC[15]	GPC[15] Pin Slew Rate Control This bit control the output slew rate of GPC[15] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[14]	SL_GPC[14]	GPC[14] Pin Slew Rate Control This bit control the output slew rate of GPC[14] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[13]	SL_GPC[13]	GPC[13] Pin Slew Rate Control This bit control the output slew rate of GPC[13] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[12]	SL_GPC[12]	GPC[12] Pin Slew Rate Control This bit control the output slew rate of GPC[12] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[11]	SL_GPC[11]	GPC[11] Pin Slew Rate Control This bit control the output slew rate of GPC[11] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[10]	SL_GPC[10]	GPC[10] Pin Slew Rate Control This bit control the output slew rate of GPC[10] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[9]	SL_GPC[9]	GPC[9] Pin Slew Rate Control This bit control the output slew rate of GPC[9] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[8]	SL_GPC[8]	GPC[8] Pin Slew Rate Control This bit control the output slew rate of GPC[8] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[7]	SL_GPC[7]	GPC[7] Pin Slew Rate Control This bit control the output slew rate of GPC[7] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[6]	SL_GPC[6]	GPC[6] Pin Slew Rate Control This bit control the output slew rate of GPC[6] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[5]	SL_GPC[5]	GPC[5] Pin Slew Rate Control This bit control the output slew rate of GPC[5] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[4]	SL_GPC[4]	GPC[4] Pin Slew Rate Control This bit control the output slew rate of GPC[4] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

[3]	SL_GPC[3]	GPC[3] Pin Slew Rate Control This bit control the output slew rate of GPC[3] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[2]	SL_GPC[2]	GPC[2] Pin Slew Rate Control This bit control the output slew rate of GPC[2] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[1]	SL_GPC[1]	GPC[1] Pin Slew Rate Control This bit control the output slew rate of GPC[1] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[0]	SL_GPC[0]	GPC[0] Pin Slew Rate Control This bit control the output slew rate of GPC[0] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

GPIO D Slew Rate Control (MISC_SL_GPD)

Register	Address	R/W	Description	Reset Value
MISC_SL_GPD	GCR_BA+0XB0	R/W	GPIO D Slew Rate 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
SL_GPD[15:0]							
7	6	5	4	3	2	1	0
SL_GPD[15:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	SL_GPD[15]	GPD[15] Pin Slew Rate Control This bit control the output slew rate of GPD[15] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[14]	SL_GPD[14]	GPD[14] Pin Slew Rate Control This bit control the output slew rate of GPD[14] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[13]	SL_GPD[13]	GPD[13] Pin Slew Rate Control This bit control the output slew rate of GPD[13] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[12]	SL_GPD[12]	GPD[12] Pin Slew Rate Control This bit control the output slew rate of GPD[12] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[11]	SL_GPD[11]	GPD[11] Pin Slew Rate Control This bit control the output slew rate of GPD[11] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[10]	SL_GPD[10]	GPD[10] Pin Slew Rate Control This bit control the output slew rate of GPD[10] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[9]	SL_GPD[9]	GPD[9] Pin Slew Rate Control This bit control the output slew rate of GPD[9] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[8]	SL_GPD[8]	GPD[8] Pin Slew Rate Control This bit control the output slew rate of GPD[8] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[7]	SL_GPD[7]	GPD[7] Pin Slew Rate Control This bit control the output slew rate of GPD[7] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[6]	SL_GPD[6]	GPD[6] Pin Slew Rate Control This bit control the output slew rate of GPD[6] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[5]	SL_GPD[5]	GPD[5] Pin Slew Rate Control This bit control the output slew rate of GPD[5] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[4]	SL_GPD[4]	GPD[4] Pin Slew Rate Control This bit control the output slew rate of GPD[4] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[3]	SL_GPD[3]	GPD[3] Pin Slew Rate Control This bit control the output slew rate of GPD[3] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[2]	SL_GPD[2]	GPD[2] Pin Slew Rate Control This bit control the output slew rate of GPD[2] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[1]	SL_GPD[1]	GPD[1] Pin Slew Rate Control This bit control the output slew rate of GPD[1] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[0]	SL_GPD[0]	GPD[0] Pin Slew Rate Control This bit control the output slew rate of GPD[0] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

GPIO E Slew Rate Control (MISC_SL_GPE)

Register	Address	R/W	Description	Reset Value
MISC_SL_GPE	GCR_BA+0XB4	R/W	GPIO E Slew Rate 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				SL_GPE[11:0]			
7	6	5	4	3	2	1	0
SL_GPE[11:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	SL_GPE[11]	GPE[11] Pin Slew Rate Control This bit control the output slew rate of GPE[11] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[10]	SL_GPE[10]	GPE[10] Pin Slew Rate Control This bit control the output slew rate of GPE[10] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[9]	SL_GPE[9]	GPE[9] Pin Slew Rate Control This bit control the output slew rate of GPE[9] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[8]	SL_GPE[8]	GPE[8] Pin Slew Rate Control This bit control the output slew rate of GPE[8] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[7]	SL_GPE[7]	GPE[7] Pin Slew Rate Control This bit control the output slew rate of GPE[7] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[6]	SL_GPE[6]	GPE[6] Pin Slew Rate Control This bit control the output slew rate of GPE[6] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[5]	SL_GPE[5]	GPE[5] Pin Slew Rate Control This bit control the output slew rate of GPE[5] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[4]	SL_GPE[4]	GPE[4] Pin Slew Rate Control This bit control the output slew rate of GPE[4] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[3]	SL_GPE[3]	GPE[3] Pin Slew Rate Control This bit control the output slew rate of GPE[3] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[2]	SL_GPE[2]	GPE[2] Pin Slew Rate Control This bit control the output slew rate of GPE[2] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[1]	SL_GPE[1]	GPE[1] Pin Slew Rate Control This bit control the output slew rate of GPE[1] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[0]	SL_GPE[0]	GPE[0] Pin Slew Rate Control This bit control the output slew rate of GPE[0] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

ND PAD Slew Rate Control (MISC_SL_ND)

Register	Address	R/W	Description	Reset Value
MISC_SL_ND	GCR_BA+0XB8	R/W	ND PAD Slew Rate 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_ND[7:0]							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7]	SL_ND[7]	ND[7] Pin Slew Rate Control This bit control the output slew rate of ND[7] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[6]	SL_ND[6]	ND[6] Pin Slew Rate Control This bit control the output slew rate of ND[6] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[5]	SL_ND[5]	ND[5] Pin Slew Rate Control This bit control the output slew rate of ND[5] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.
[4]	SL_ND[4]	ND[4] Pin Slew Rate Control This bit control the output slew rate of ND[4] pin. 1'b0: Output slew rate is Fast. 1'b1: Output slew rate is Slow.

[3]	SL_ND[3]	ND[3] Pin Slew Rate Control This bit control the output slew rate of ND[3] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[2]	SL_ND[2]	ND[2] Pin Slew Rate Control This bit control the output slew rate of ND[2] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[1]	SL_ND[1]	ND[1] Pin Slew Rate Control This bit control the output slew rate of ND[1] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.
[0]	SL_ND[0]	ND[0] Pin Slew Rate Control This bit control the output slew rate of ND[0] pin. 1′b0: Output slew rate is Fast. 1′b1: Output slew rate is Slow.

GPIO A Driver Strength Control (MISC_DS_GPA)

Register	Address	R/W	Description	Reset Value
MISC_DS_GPA	GCR_BA+0xBC	R/W	GPIO A Driver Strength 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				DS_GPA[11:0]			
7	6	5	4	3	2	1	0
DS_GPA[11:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	DS_GPA[11:0]	GPA[11:0] Pin Driver Strength Control This bit control the output Driver Strength of GPA[11:0] pin. 1′b0: 4mA 1′b1: 8mA

GPIO B Driver Strength Control (MISC_DS_GPB)

Register	Address	R/W	Description	Reset Value
MISC_DS_GPB	GCR_BA+0xC0	R/W	GPIO B Driver Strength 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
DS_GPB[15:0]							
7	6	5	4	3	2	1	0
DS_GPB[15:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	DS_GPB[15:0]	GPB[15:0] Pin Driver Strength Control This bit control the output Driver Strength of GPB[15:0] pin. 1′b0: 4mA 1′b1: 8mA

GPIO C Driver Strength Control (MISC_DS_GPC)

Register	Address	R/W	Description	Reset Value
MISC_DS_GPC	GCR_BA+0XC4	R/W	GPIO C Driver Strength 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
DS_GPC[15:0]							
7	6	5	4	3	2	1	0
DS_GPC[15:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	DS_GPC[15:0]	GPC[15:0] Pin Driver Strength Control This bit control the output Driver Strength of GPC[15:0] pin. 1′b0: 4mA 1′b1: 8mA

GPIO D Driver Strength Control (MISC_DS_GPD)

Register	Address	R/W	Description	Reset Value
MISC_DS_GPD	GCR_BA+0XC8	R/W	GPIO D Driver Strength 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
DS_GPD[15:0]							
7	6	5	4	3	2	1	0
DS_GPD[15:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	DS_GPD[15:0]	GPD[15:0] Pin Driver Strength Control This bit control the output Driver Strength of GPD[15:0] pin. 1′b0: 4mA 1′b1: 8mA

GPIO E Driver Strength Control (MISC_DS_GPE)

Register	Address	R/W	Description	Reset Value
MISC_DS_GPE	GCR_BA+0xCC	R/W	GPIO E Driver Strength 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				DS_GPE[11:0]			
7	6	5	4	3	2	1	0
DS_GPE[11:0]							

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11:0]	DS_GPE[11:0]	GPE[11:0] Pin Driver Strength Control This bit control the output Driver Strength of GPE[11:0] pin. 1′b0: 4mA 1′b1: 8mA

ND PAD Driver Strength Control (MISC_DS_ND)

Register	Address	R/W	Description	Reset Value
MISC_DS_ND	GCR_BA+0XD0	R/W	ND PAD Driver Strength 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
DS_ND[7:0]							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	DS_ND[7:0]	ND[7:0] Pin Driver Strength Control This bit control the output driver strength of ND[7:0] pin. 1′b0: 4mA 1′b1: 8mA

SSTL2 and LVTTL Driver Strength Control (MISC_SSEL)

Register	Address	R/W	Description	Reset Value
MISC_SSEL	GCR_BA+0XD4	R/W	SSTL2 and LVTTL Driver Strength control	0x0000_0300

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						MCLK	
7	6	5	4	3	2	1	0
MA		MBA		MCTL		MD	

Bits	Descriptions	
[31:10]	Reserved	Reserved
[9:8]	MCLK	MCLK Pin Driving Strength Control and Mode

Mode	Power Supply	MCLK [1:0]			
		Reduced Strength		Full Strength	
SSTL18	1.8V	01		11	
SSTL2	2.5V	00		10	
LPDDR	1.8V	00		11	
LVTTL	3.3V	00(12mA)	01(16mA)	10(24mA)	11(30mA)

[7:6]	MA	MA Pin Driving Strength Control and Mode

Mode	Power Supply	MA [1:0]			
		Reduced Strength		Full Strength	
SSTL18	1.8V	01		11	
SSTL2	2.5V	00		10	
LPDDR	1.8V	00		11	
LVTTL	3.3V	00(12mA)	01(16mA)	10(24mA)	11(30mA)

[5:4]	MBA	MBA Pin Driving Strength Control and Mode

Mode	Power Supply	MBA [1:0]			
		Reduced Strength		Full Strength	
SSTL18	1.8V	01		11	
SSTL2	2.5V	00		10	
LPDDR	1.8V	00		11	
LVTTL	3.3V	00(12mA)	01(16mA)	10(24mA)	11(30mA)

[3:2]	MCTL	MCTL Pin Driving Strength Control and Mode

Mode	Power Supply	MCTL [1:0]			
		Reduced Strength		Full Strength	
SSTL18	1.8V	01		11	
SSTL2	2.5V	00		10	
LPDDR	1.8V	00		11	
LVTTL	3.3V	00(12mA)	01(16mA)	10(24mA)	11(30mA)

[1:0]	MD	MD Pin Driving Strength Control and Mode

Mode	Power Supply	MD [1:0]			
		Reduced Strength		Full Strength	
SSTL18	1.8V	01		11	
SSTL2	2.5V	00		10	
LPDDR	1.8V	00		11	
LVTTL	3.3V	00(12mA)	01(16mA)	10(24mA)	11(30mA)

GPIO G Multi-function Control Register (GPGFUN)

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

Register	Address	R/W	Description	Reset Value
GPGFUN	GCR_BA+0xD8	R/W	GPIO G Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MF_GPG15		MF_GPG14		MF_GPG13		MF_GPG12	
23	22	21	20	19	18	17	16
MF_GPG11		MF_GPG10		MF_GPG9		MF_GPG8	
15	14	13	12	11	10	9	8
MF_GPG7		MF_GPG6		MF_GPG5		MF_GPG4	
7	6	5	4	3	2	1	0
MF_GPG3		MF_GPG2		MF_GPG1		MF_GPG0	

Bits	Descriptions	
[31:30]	MF_GPG15	GPIOG[15] Multi-function

Pin Name	MF_GPG15			
ADC_TP_YM	2′b00	2′b01	2′b10	2′b11
	GPIOG[15]	Reserved	SPI1_DO	ISDA

Bits	Descriptions	
[29:28]	MF_GPG14	GPIOG[14] Multi-function

Pin Name	MF_GPG14			
ADC_TP_XM	2′b00	2′b01	2′b10	2′b11
	GPIOG[14]	Reserved	SPI1_DI	Reserved

Bits	Descriptions	
[27:26]	MF_GPG13	GPIOG[13] Multi-function

Pin Name	MF_GPG13			
ADC_TP_XP	2′b00	2′b01	2′b10	2′b11
	GPIOG[13]	Reserved	SPI1_CS0	Reserved

[25:24]	MF_GPG12	GPIOG[12] Multi-function

Pin Name	MF_GPG12			
ADC_TP_YP	2'b00	2'b01	2'b10	2'b11
	GPIOG[12]	Reserved	SPI1_CLK	ISCK

[23:22]	MF_GPG11	GPIOG[11] Multi-function

Pin Name	MF_GPG11			
MIC_BIAS	2'b00	2'b01	2'b10	2'b11
	GPIOG[11]	Reserved	I2S_BCLK	Reserved

[21:20]	MF_GPG10	GPIOG[10] Multi-function

Pin Name	MF_GPG10			
ADC_AIN[0]	2'b00	2'b01	2'b10	2'b11
	GPIOG[10]	Reserved	I2S_WS	Reserved

[19:18]	MF_GPG9	GPIOG[9] Multi-function

Pin Name	MF_GPG9			
ADC_AIN[1]	2'b00	2'b01	2'b10	2'b11
	GPIOG[9]	Reserved	I2S_DI	Reserved

[17:16]	MF_GPG8	GPIOG[8] Multi-function

Pin Name	MF_GPG8			
ADC_AIN[2]	2'b00	2'b01	2'b10	2'b11
	GPIOG[8]	Reserved	Reserved	Reserved

[15:14]	MF_GPG7	GPIOG[7] Multi-function

Pin Name	MF_GPG7			
ADC_AIN3	2'b00	2'b01	2'b10	2'b11
	GPIOG[7]	Reserved	Reserved	Reserved

[13:12]	MF_GPG6	GPIOG[6] Multi-function

Pin Name	MF_GPG6			
ADC_AIN[4]	2'b00	2'b01	2'b10	2'b11
	GPIOG[6]	Reserved	Reserved	Reserved

[11:10]	MF_GPG5	GPIOG[5] Multi-function
[9:8]	MF_GPG4	GPIOG[4] Multi-function
[7:6]	MF_GPG3	GPIOG[3] Multi-function
[5:4]	MF_GPG2	GPIOG[2] Multi-function
[3:2]	MF_GPG1	GPIOG[1] Multi-function
[1:0]	MF_GPG0	GPIOG[0] Multi-function

GPIOG[5] Multi-function

Pin Name	MF_GPG5			
TVDAC_VREF	2'b00	2'b01	2'b10	2'b11
	GPIOG[5]	Reserved	SPI_DO	I2S_MCLK

GPIOG[4] Multi-function

Pin Name	MF_GPG4			
TVDAC_COMP	2'b00	2'b01	2'b10	2'b11
	GPIOG[4]	Reserved	SPI1_DI	I2S_WS

GPIOG[3] Multi-function

Pin Name	MF_GPG3			
TVDAC_REXT	2'b00	2'b01	2'b10	2'b11
	GPIOG[3]	Reserved	SPI1_CS0	I2S_BCLK

GPIOG[2] Multi-function

Pin Name	MF_GPG2			
TVDAC_TVOUT	2'b00	2'b01	2'b10	2'b11
	GPIOG[2]	Reserved	SPI1_CLK	I2S_DOUT

GPIOG[1] Multi-function

Pin Name	MF_GPG1			
RTC_XOUT	2'b00	2'b01	2'b10	2'b11
	GPIOG[1]	Reserved	Reserved	Reserved

GPIOG[0] Multi-function

Pin Name	MF_GPG0			
RTC_XIN	2'b00	2'b01	2'b10	2'b11
	GPIOG[0]	Reserved	Reserved	Reserved

GPIO H Multi-function Control Register (GPHFUN)

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

Register	Address	R/W	Description	Reset Value
GPHFUN	GCR_BA+0xDC	R/W	GPIO H Multi-function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
MF_GPH15		MF_GPH14		MF_GPH13		MF_GPH12	
23	22	21	20	19	18	17	16
MF_GPH11		MF_GPH10		MF_GPH9		MF_GPH8	
15	14	13	12	11	10	9	8
MF_GPH7		MF_GPH6		MF_GPH5		MF_GPH4	
7	6	5	4	3	2	1	0
MF_GPH3		MF_GPH2		MF_GPH1		MF_GPH0	

Bits	Descriptions	
[31:30]	MF_GPH15	GPIOH[15] Multi-function Pin Name: MF_GPH15 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[29:28]	MF_GPH14	GPIOH[14] Multi-function Pin Name: MF_GPH14 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[27:26]	MF_GPH13	GPIOH[13] Multi-function Pin Name: MF_GPH13 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved

[25:24]	MF_GPH12	GPIOH[12] Multi-function Pin Name: MF_GPH12 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[23:22]	MF_GPH11	GPIOH[11] Multi-function Pin Name: MF_GPH11 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[21:20]	MF_GPH10	GPIOH[10] Multi-function Pin Name: MF_GPH10 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[19:18]	MF_GPH9	GPIOH[9] Multi-function Pin Name: MF_GPH9 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[17:16]	MF_GPH8	GPIOG[8] Multi-function Pin Name: MF_GPH8 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[15:14]	MF_GPH7	GPIOG[7] Multi-function Pin Name: MF_GPH7 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved
[13:12]	MF_GPH6	GPIOH[6] Multi-function Pin Name: MF_GPH6 2'b00: Reserved; 2'b01: Reserved; 2'b10: Reserved; 2'b11: Reserved

<table>
<tr><td>[11:10]</td><td>MF_GPH5</td><td>GPIOH[5] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPH5</td></tr><tr><td rowspan="2">R_FB</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOH[5]</td><td>LVD_OUT</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[9:8]</td><td>MF_GPH4</td><td>GPIOH[4] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPH4</td></tr><tr><td rowspan="2"></td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>Reserved</td><td>LVD_OUT(GPD[15])</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[7:6]</td><td>MF_GPH3</td><td>GPIOH[3] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPH3</td></tr><tr><td rowspan="2"></td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>Reserved</td><td>LVD_OUT(GPD[3])</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[5:4]</td><td>MF_GPH2</td><td>GPIOH[2] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPH2</td></tr><tr><td rowspan="2"></td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>Reserved</td><td>ISDA(VREF)</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[3:2]</td><td>MF_GPH1</td><td>GPIOH[1] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPH1</td></tr><tr><td rowspan="2"></td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>Reserved</td><td>ISCK(TVOUT)</td><td>Reserved</td><td>Reserved</td></tr></table></td></tr>
<tr><td>[1:0]</td><td>MF_GPH0</td><td>GPIOH[0] Multi-function<table><tr><td>Pin Name</td><td colspan="4">MF_GPH0</td></tr><tr><td rowspan="2">PGC_VREF</td><td>2′b00</td><td>2′b01</td><td>2′b10</td><td>2′b11</td></tr><tr><td>GPIOH[0]</td><td>Reserved</td><td>I2S_MCLK</td><td>Reserved</td></tr></table></td></tr>
</table>

Share Pin with TVDAC Control (ShrPin_TVDAC)

Register	Address	R/W	Description	Reset Value
ShrPin_TVDAC	GCR_BA+0XF0	R/W	Share Pins with TVDAC	0x8XXX_XXXX

31	30	29	28	27	26	25	24
SMTVDAC_AEN	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]	SMTVDAC_AEN	Analog and Digital Share I/O Pad Control Bit for TVDAC VREF, REXT, COMP, IOUT Pins. 1'b1: Analog Pin 1'b0: Digital Pin
[30:0]	Reserved	

Share Pin with AUDIO control (ShrPin_AUDIO)

Register	Address	R/W	Description	Reset Value
ShrPin_AUDIO	GCR_BA+0XF4	R/W	Share Pins with AUDIO ADC	0xFXXX_XXXX

31	30	29	28	27	26	25	24
MIC_AEN	AIN2_AEN	AIN3_AEN	AIN4_AEN	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]	MIC_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC MICP, MICN, and MIC_BIAS Pins. 1'b1: Analog Pin 1'b0: Digital Pin
[30]	AIN2_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC AIN2 Pins. 1'b1: Analog Pin 1'b0: Digital Pin
[29]	AIN3_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC AIN3 Pins. 1'b1: Analog Pin 1'b0: Digital Pin
[28]	AIN4_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC AIN4 Pins. 1'b1: Analog Pin 1'b0: Digital Pin
[27:0]	Reserved	

Share Pin with TOUCH Control (ShrPin_TOUCH)

Register	Address	R/W	Description	Reset Value
ShrPin_TOUCH	GCR_BA+0XF8	R/W	Share Pins with TOUCH ADC	0xEXXX_XXXX

31	30	29	28	27	26	25	24
PGC_VREF_AEN	TP_AEN	MIC_BIAS_AEN	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]	PGC_VREF_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC PGC_VREF Pins. 1′b1: Analog Pin 1′b0: Digital Pin
[30]	TP_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC XP, XM, YP, YM Pins. 1′b1: Analog Pin 1′b0: Digital Pin
[29]	MIC_BIAS_AEN	Analog and Digital Share I/O Pad Control Bit for SAR-ADC MIC_BIAS Pin. 1′b1: Analog Pin 1′b0: Digital Pin
[28:0]	Reserved	

Share Pin with R_FB control (ShrPin_R_FB)

Register	Address	R/W	Description	Reset Value
ShrPin_R_FB	GCR_BA+0XFC	R/W	Share Pins with R_Divider	0x0000_010E

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				R_DIV_ENB	Reserved	R_SEL	
7	6	5	4	3	2	1	0
Reserved	R_TUNE_SEL			R_FB_AEN	Reserved		

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	R_DIV_ENB	R-Divider enable bar: active low(default=0) 1′b0: enable 1′b1: disable
[10]	Reserved	Reserved
[9:8]	R_SEL	FB resistor selection: 2′00: 1.00V output 2′b01: 1.2V output è default value 2′b10: 1.32V output 2′b11: 1.40V output
[7]	Reserved	Reserved
[6:4]	R_TUNE _SEL	Fine Tune FB output voltage control 0~7 : 0~7% of output voltage fine tune Default = 3′b000

[3]	R_FB_AEN	Analog enable: default = 1 1′b0: digital I/O mode 1′b1: analog mode
[2:0]	Reserved	Reserved

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

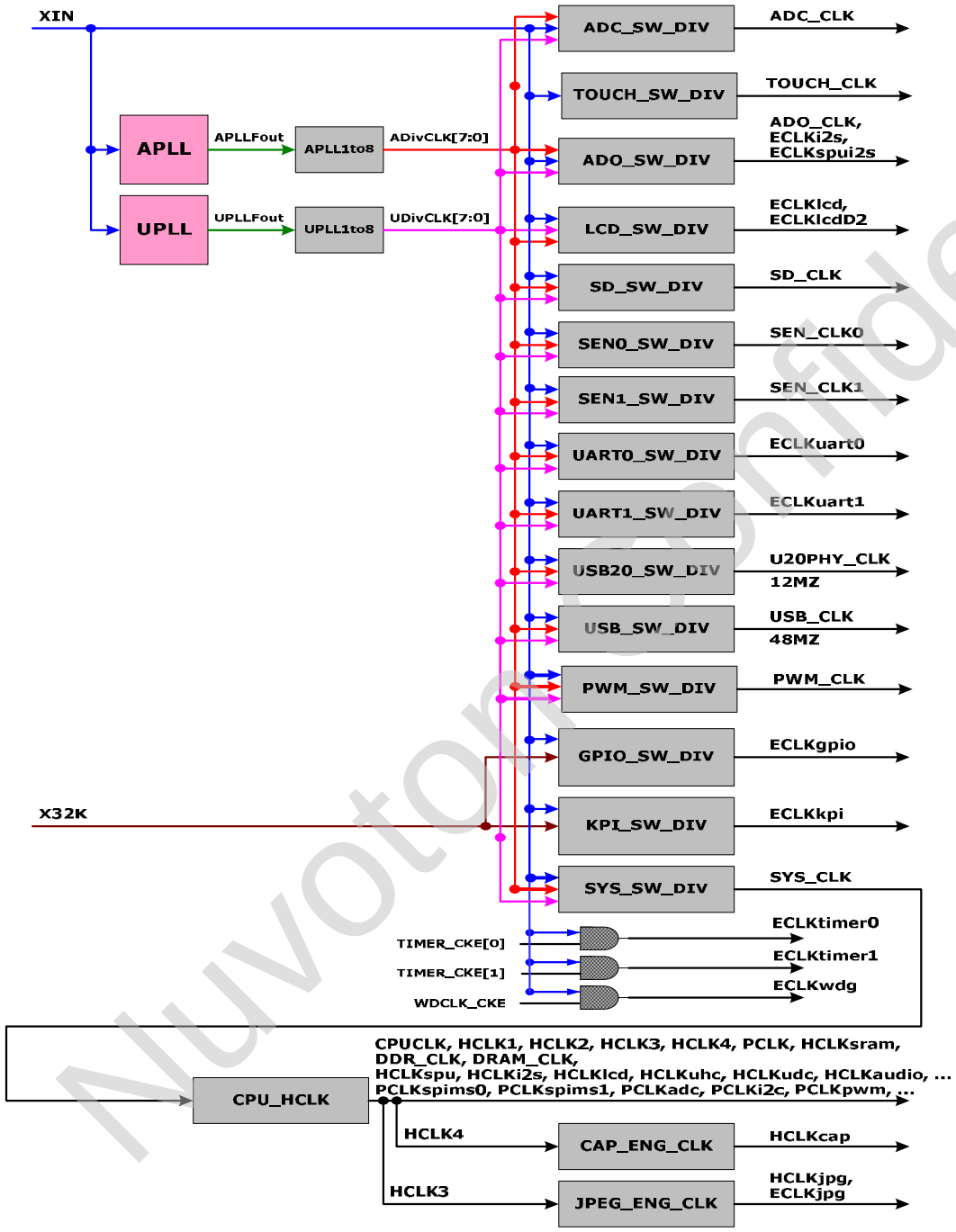
6.3 Clock Controller

6.3.1 Clock controller overview

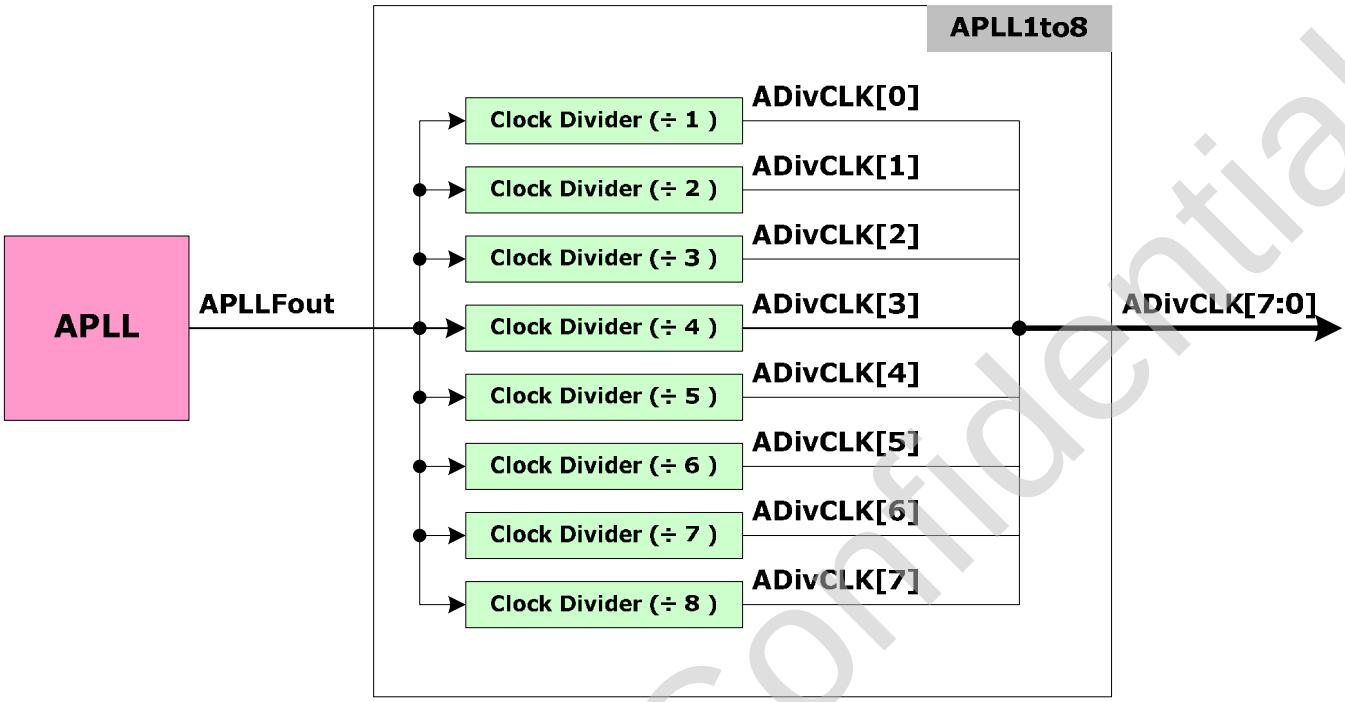
The clock controller generates the clocks for the whole chip, it include all of IPs on AHB, APB and engine clock like USB, UART and so on. There are two PLLs in this chip, and the PLL clock source is from the external crystal input. It also implements the power control function, include the individually 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.

6.3.2 Block diagram

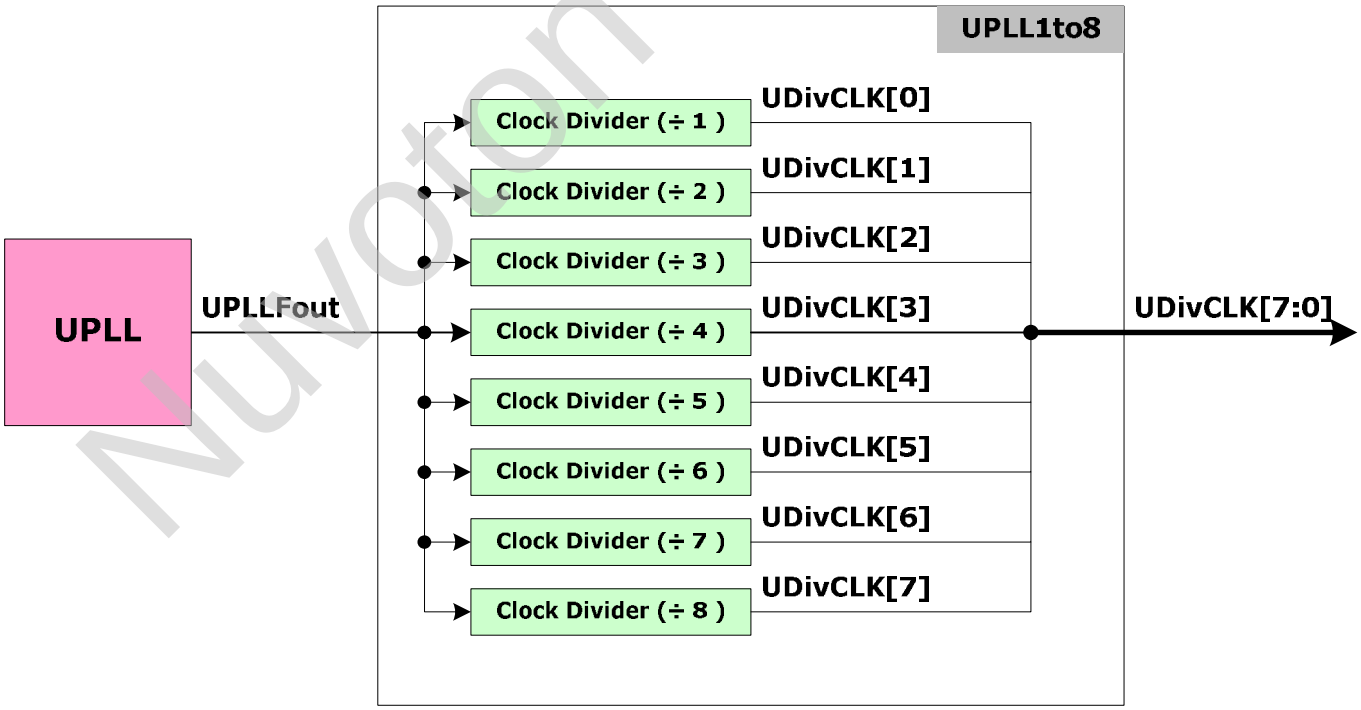
6.3.2.1 Clock Controller Overview



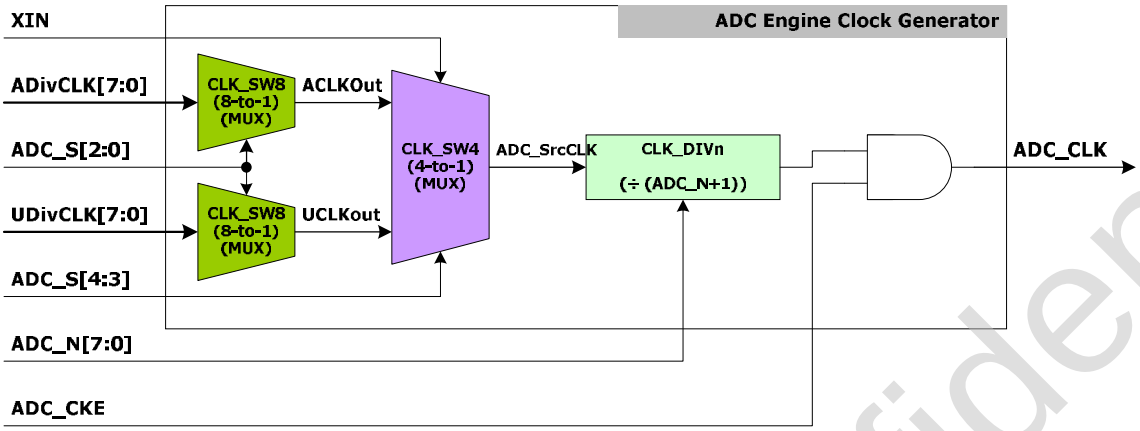
6.3.2.2 APLL1to8



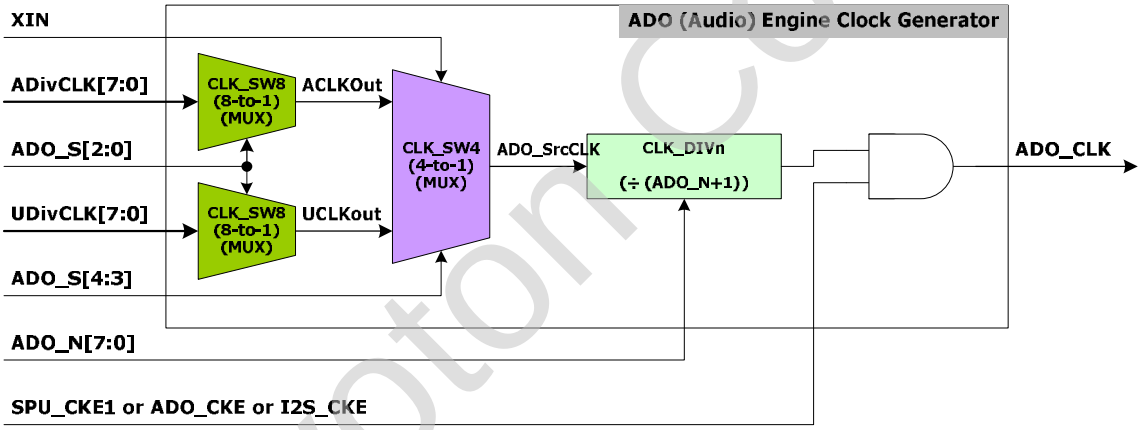
6.3.2.3 UPLL1to8



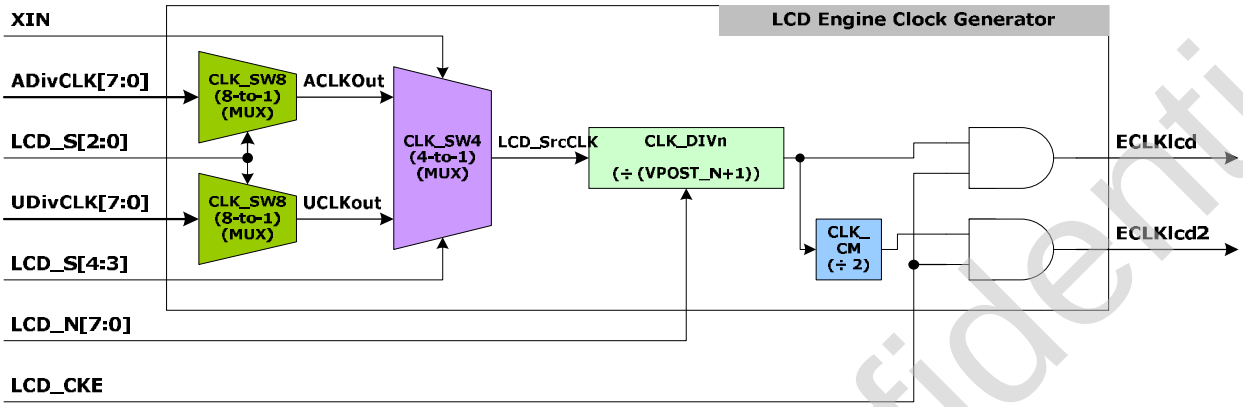
6.3.2.4 ADC_SW_DIV



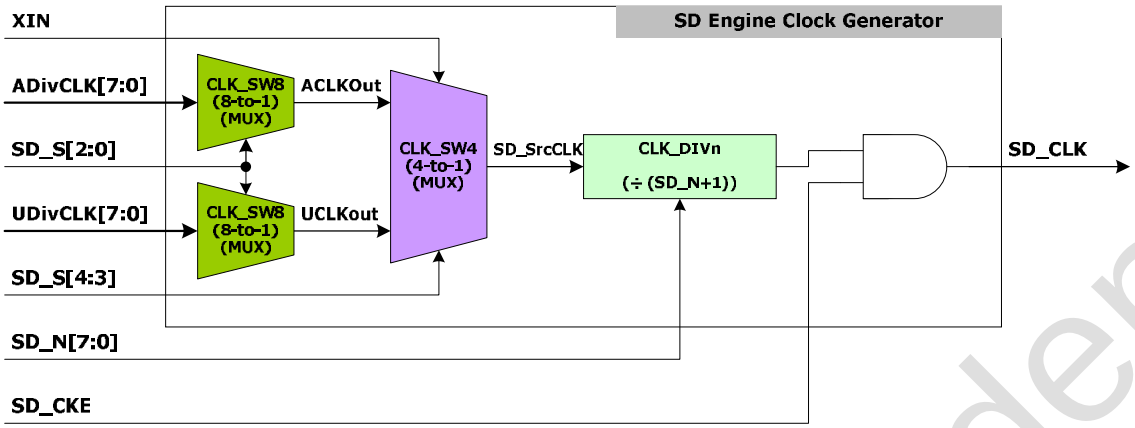
6.3.2.5 ADO_SW_DIV



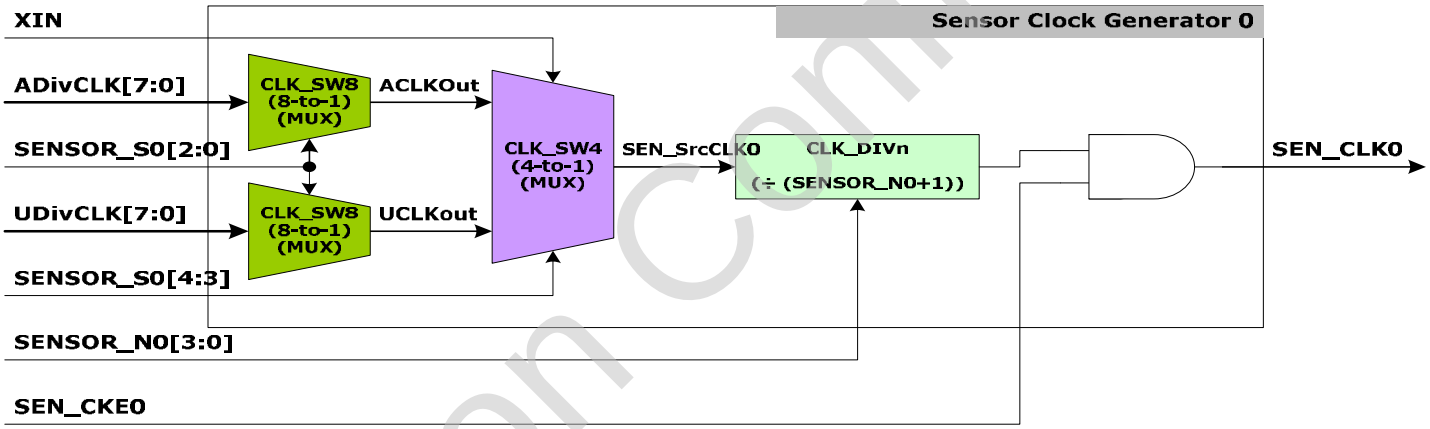
6.3.2.6 LCD_SW_DIV



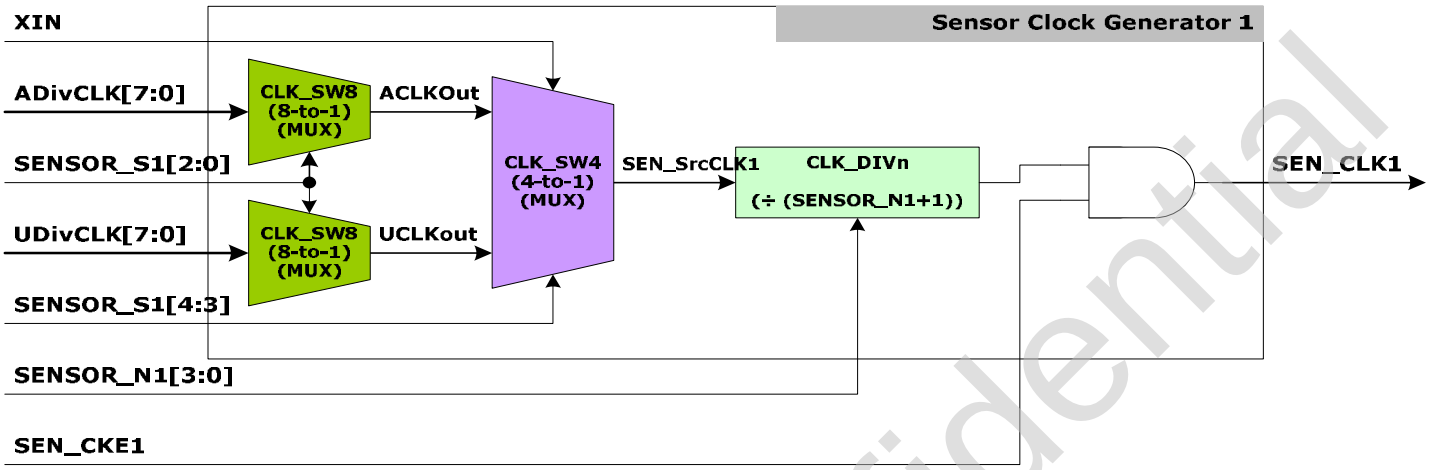
6.3.2.7 SD_SW_DIV



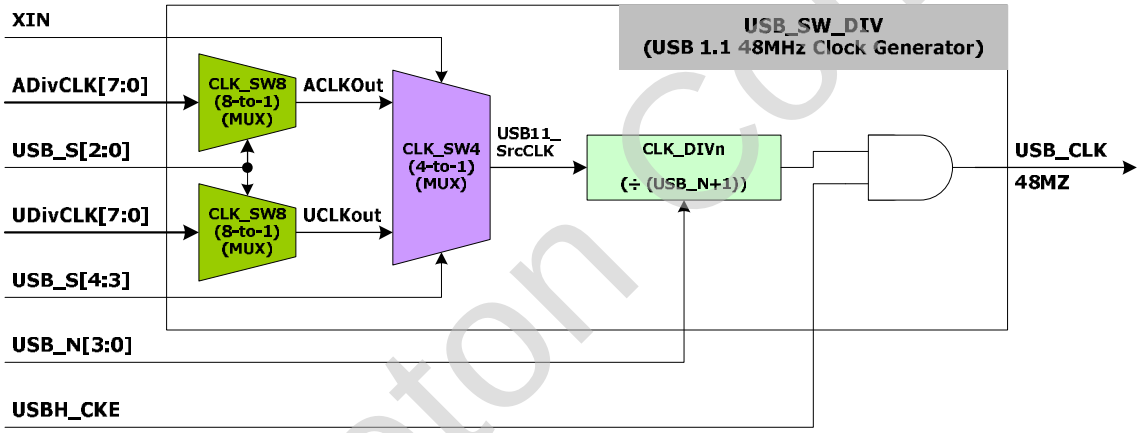
6.3.2.8 SEN_SW_DIV0



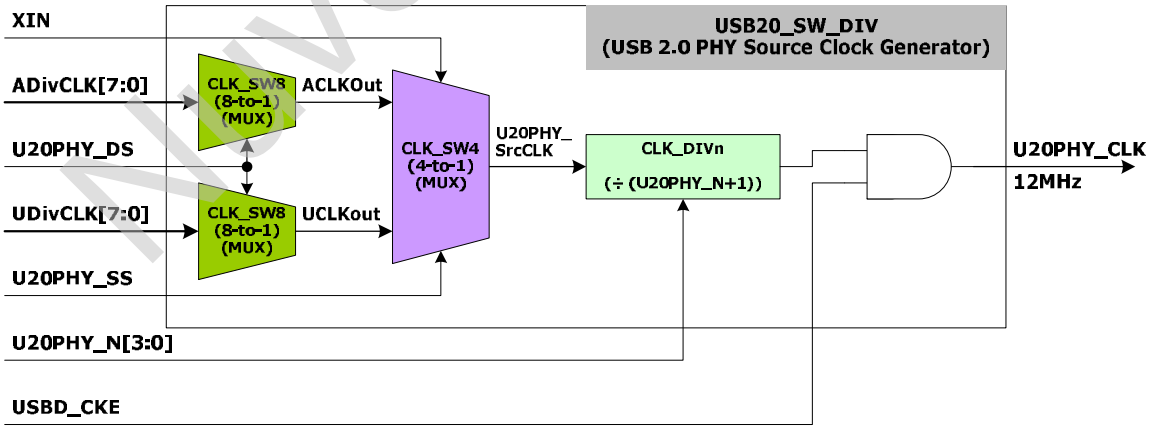
6.3.2.9 SEN_SW_DIV1



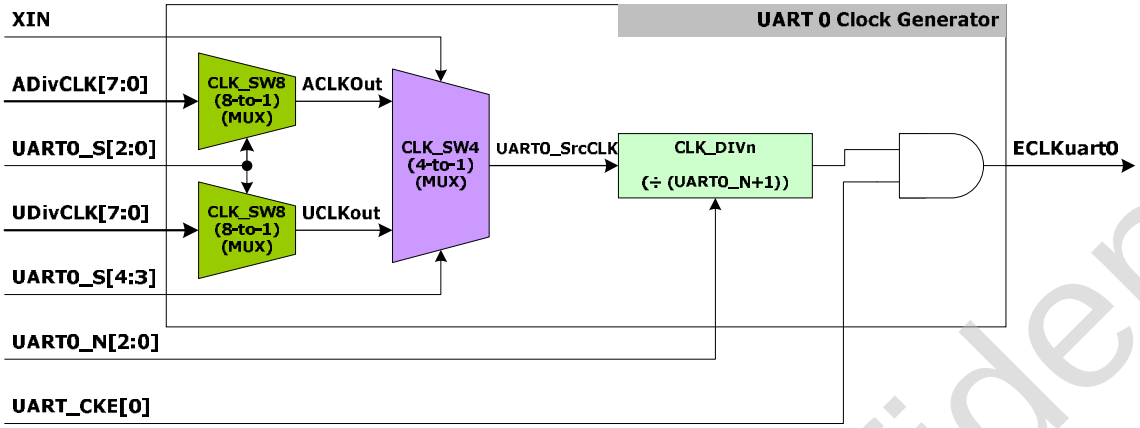
6.3.2.10 USB_SW_DIV



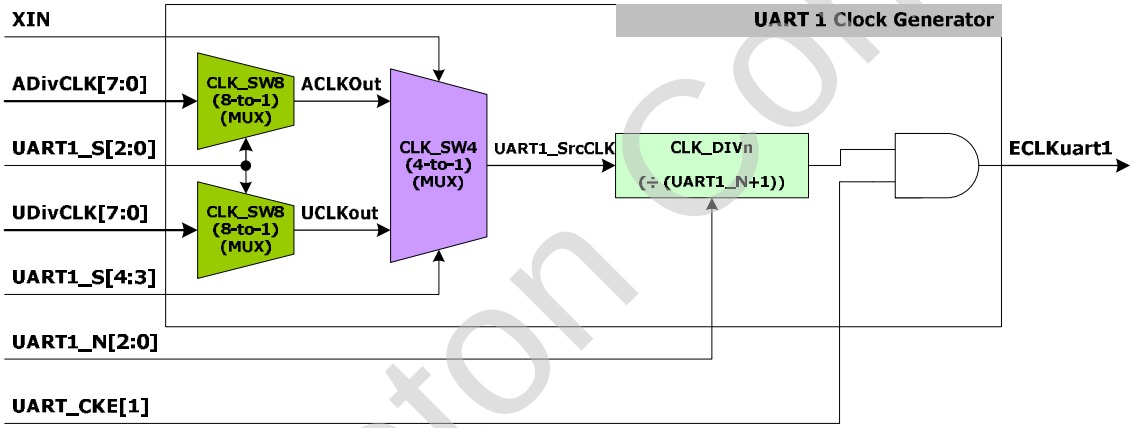
6.3.2.11 USB20_SW_DIV



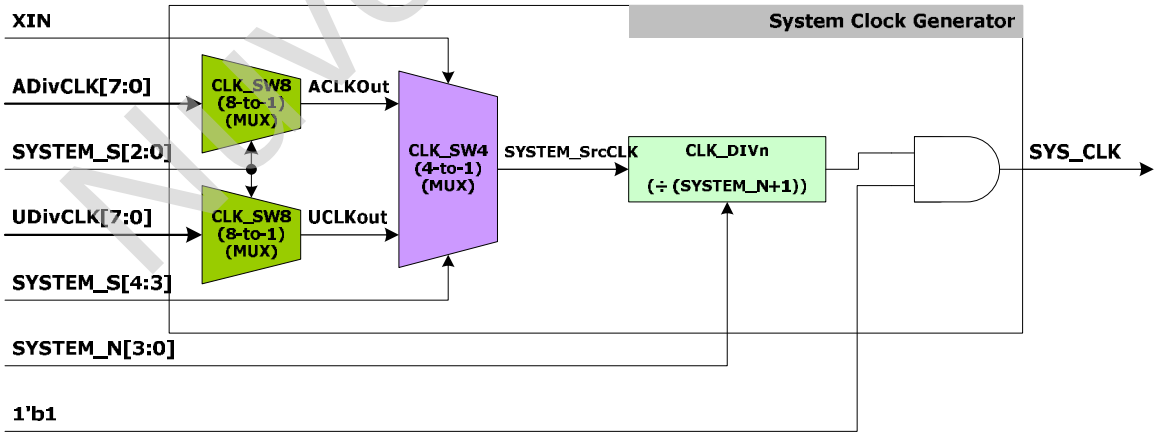
6.3.2.12 UART0_SW_DIV



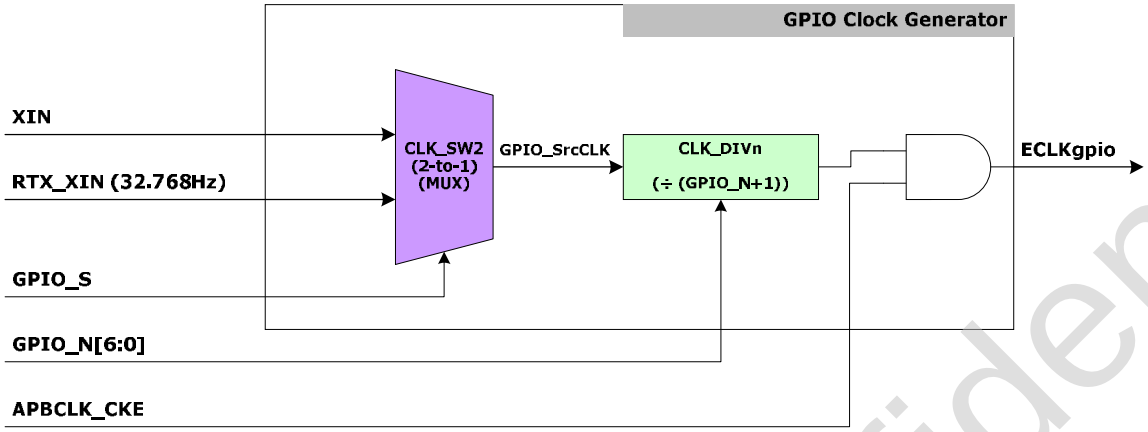
6.3.2.13 UART1_SW_DIV



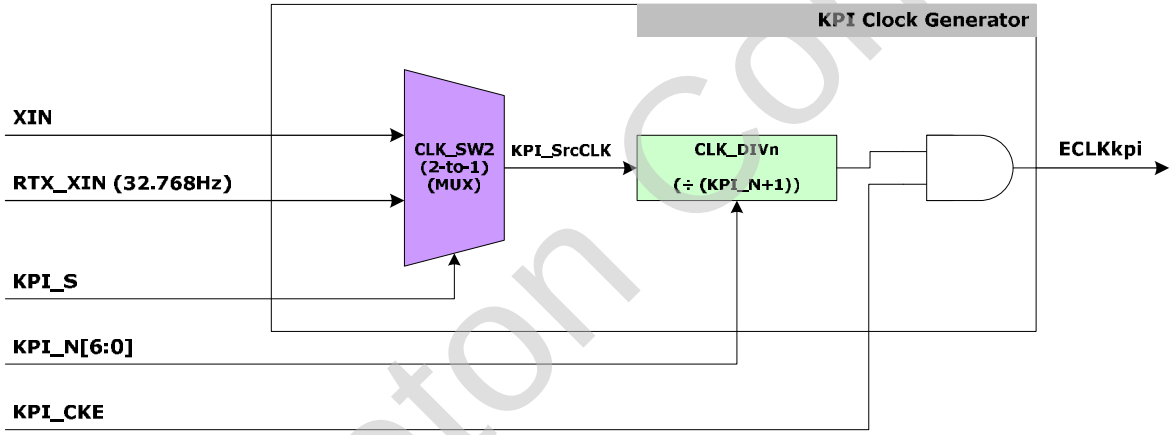
6.3.2.14 SYS_SW_DIV



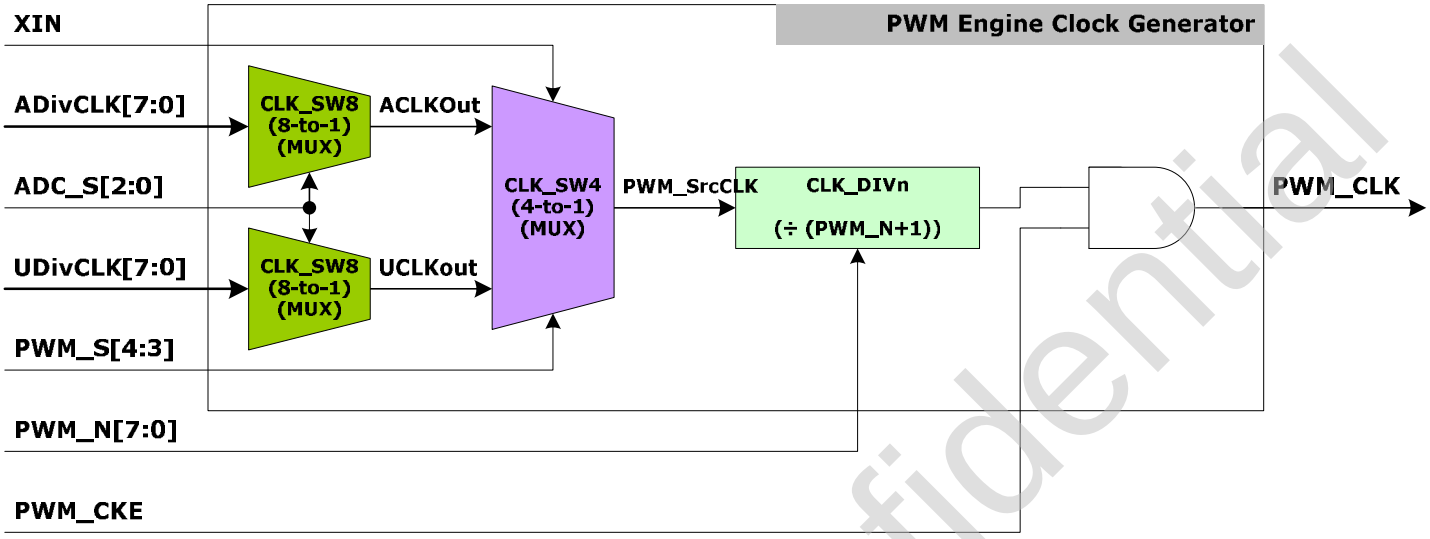
6.3.2.15 GPIO_SW_DIV



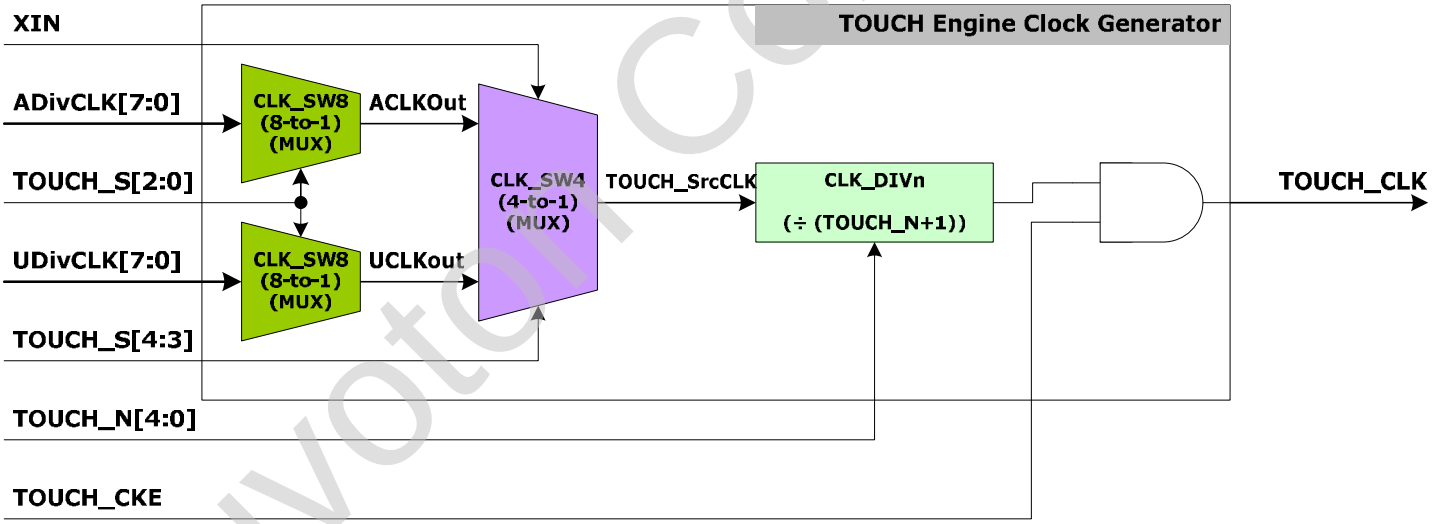
6.3.2.16 KPI_SW_DIV



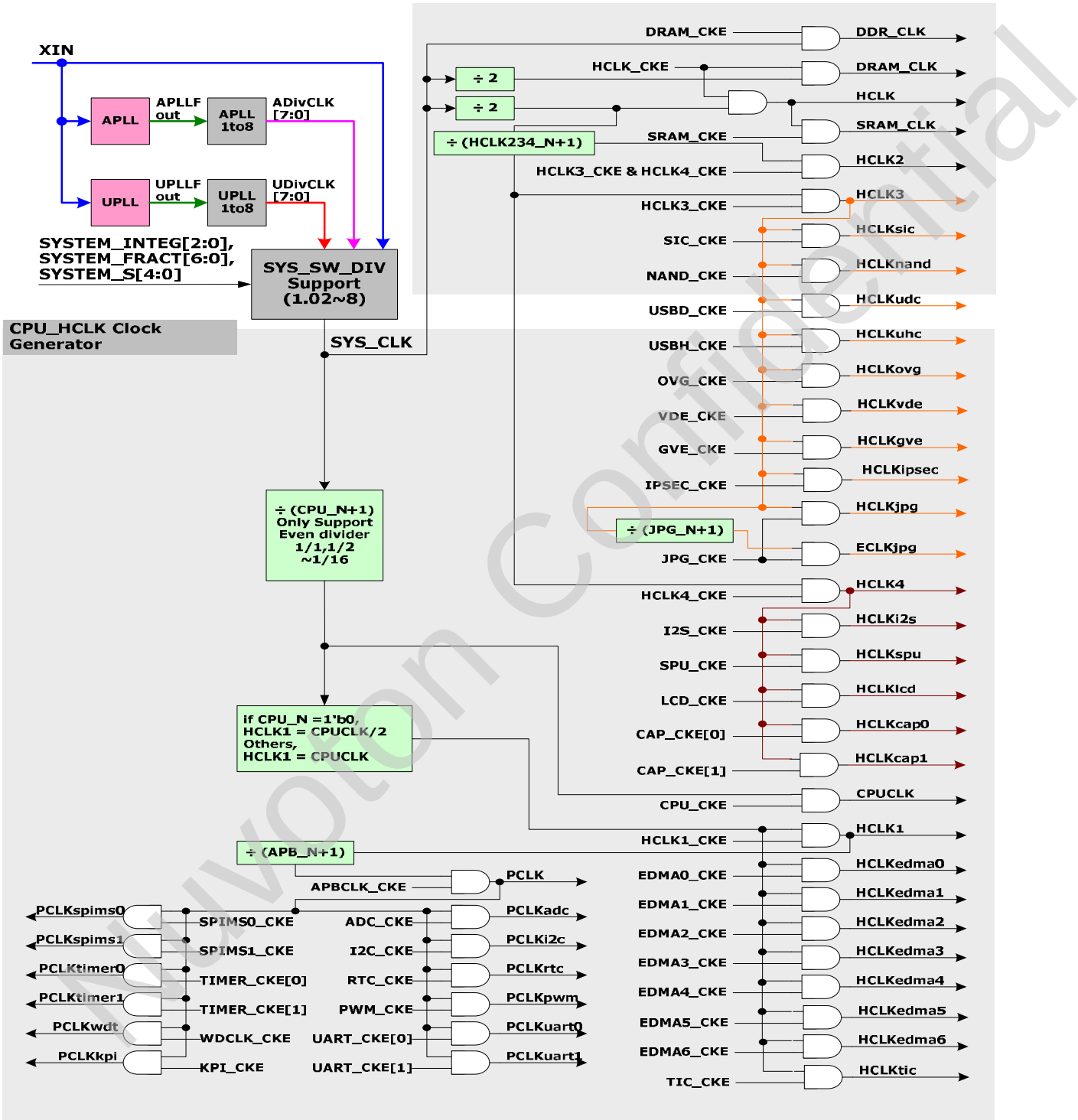
6.3.2.17 PWM_SW_DIV



6.3.2.18 TOUCH_SW_DIV



6.3.2.18.1 CPU_HCLK



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 0x0, the HCLK1 will be the half of CPU clock. If CPU_N is not 0x0, the HCLK1 will be the same CPU clock.

The CPU_N only supported 0x0,0x1,0x3,0x5,0x7,0x9,0xB,0xD,0xF.

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.

6.3.3 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	0x3800_0100
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	0x0000_5118
UPLLCON	CLK_BA + 24	R/W	UPLL Control Register	0x0000_5118
CLK_TREG	CLK_BA + 30	R/W	TEST Clock Control Register	0x0000_0000
AHBCLK2	CLK_BA + 34	R/W	Clock Enable Control Register	0x0000_0000
CLKDIV5	CLK_BA + 38	R/W	Clock Divider Number Register 5	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		SEN1_OFF_ST	SEN0_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).
[7:6]	Reserved	Reserved
[5]	SEN1_OFF_ST	Sensor1 clock level on clock off state 0 = sensor clock keep on low level 1 = sensor clock keep on high level
[4]	SEN0_OFF_ST	Sensor0 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	SEN0_CKE	VIN0_CKE	VPOST_CKE	I2S_CKE	SPU_CKE	HCLK4_CKE
23	22	21	20	19	18	17	16
SD_CKE	NAND_CKE	SIC_CKE	OVG_CKE	GVE_CKE	USBD_CKE	USBH_CKE	HCLK3_CKE
15	14	13	12	11	10	9	8
VDE_CKE	EDMA4_CKE	EDMA3_CKE	EDMA2_CKE	EDMA1_CKE	EDMA0_CKE	IPSEC_CKE	HCLK1_CKE
7	6	5	4	3	2	1	0
JPG_CKE	VPE_CKE	GE_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]	SEN0_CKE	Sensor0 Interface Clock Enable Control 0 = Disable 1 = Enable
[28]	VIN0_CKE	Video_0 In Clock Enable Control (Also is Video In 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]	OVG_CKE	OPEN VG Processing Unit Clock Enable Control 0 = Disable 1 = Enable
[19]	GVE_CKE	Graphic Video Switch Engine Clock Enable Control 0 = Disable 1 = Enable It is always 1′b1
[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]	VDE_CKE	Video Decoder Clock Enable Control 0 = Disable 1 = Enable
[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]	IPSEC_CKE	AES Clock Enable Control 0 = Disable 1 = Enable
[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]	VPE_CKE	Video Engine Clock Enable Control 0 = Disable 1 = Enable
[5]	GE_CKE	Graphic Engine Clock Enable Control 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				TOUCH_CKE	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.
[14:11]	Reserved	Reserved

[10]	TOUCH_CKE	TOUCH Clock Enable Control 0 = Disable 1 = Enable
[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	0x3800_0100

31	30	29	28	27	26	25	24
SYSTEM_INTEG			SYSTEM_FRACT		SENSOR_N0		
23	22	21	20	19	18	17	16
	SENSOR_S0					KPI_N[6:5]	
15	14	13	12	11	10	9	8
KPI_N[4:0]					SYSTEM_FRACT		
7	6	5	4	3	2	1	0
SYSTEM_FRACT		KPI_S	SYSTEM_S				

Bits	Descriptions	
[31:29]	SYSTEM_INTEG	System Clock Divide Integer Part This field defines the clock divide integer number for clock divider to generate the system clock SYS_CLK. The SYS_CLK = SYSTEM_SrcCLK /(SYSTEM_INTEG+SYSTEM_FRACT*0.01) For example: IF SYSTEM_INTEG[2:0]=5 and SYSTEM_FRACT[6:0]=79 (both are decimal) The SYS_CLK = SYSTEM_SrcCLK/5.79 1<= Clock Divide Integer Part[3:0] <=7
[28:27]	SYSTEM_FRACT	System Clock Divide Fractional Part[6:5] This field defines the clock divide fractional number for clock divider to generate the system clock SYS_CLK. The SYS_CLK = SYSTEM_SrcCLK /(SYSTEM_INTEG+SYSTEM_FRACT*0.01) For example: IF SYSTEM_INTEG[2:0]=5 and SYSTEM_FRACT[6:0]=79 (both are decimal) The SYS_CLK = SYSTEM_SrcCLK/5.79 2<= Clock Divide Fractional Part[6:0] <=100

[26:23]	SENSOR_N0	**Sensor0 Clock Divide** This field defines the clock divide number for clock divider to generate the sensor clock. The actual clock divide number is (SENSOR_N0 + 1). So, SEN_CLK0 = SEN_SrcCLK0 / (SENSOR_N0 + 1)
[22:21]	SENSOR_S0	**Sensor0 Clock Source Selection** This field selects which clock is used to be the source of sensor clock. 00: SEN_SrcCLK0 = XIN 01: SEN_SrcCLK0 = Reserved 10: SEN_SrcCLK0 = ACLKOut 11: SEN_SrcCLK0 = UCLKOut
[20:18]		**Sensor0 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 SENSOR_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
[17:11]	KPI_N[6:0]	**KPI Engine Clock Divider Bits [6:0]** This field defines the bits [3:0] of clock divide number for clock divider to generate the engine clock for KPI. So, ECLKkpi = KPI_SrcCLK / (KPI_N [6:0] + 1)

[10:6]	SYSTEM_FRACT	**System Clock Divide Fractional Part[4:0]** This field defines the clock divide fractional number for clock divider to generate the system clock SYS_CLK. The SYS_CLK = SYSTEM_SrcCLK /(SYSTEM_INTEG+SYSTEM_FRACT*0.01) For example: IF SYSTEM_INTEG[2:0]=5 and SYSTEM_FRACT[6:0]=79 (both are decimal) The SYS_CLK = SYSTEM_SrcCLK/5.79 2<= Clock Divide Fractional Part[6:0] <=100
[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 SYSTEM_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 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. ***And, this field only takes effect while the VPOST_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 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 SD_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
[15:12]	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 U20PHY_SS 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. *And, this field only takes effect while the ADC_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
[15:13]	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. *And, this field only takes effect while the UART1_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
[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. ***And, this field only takes effect while the UART0_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 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	VIN1_N				JPG_N		
23	22	21	20	19	18	17	16
GPIO_N							GPIO_S
15	14	13	12	11	10	9	8
HCLK1_N	VIN0_N			APB_N			
7	6	5	4	3	2	1	0
HCLK234_N				CPU_N			

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29:27]	VIN1_N	Capture_1 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 (VIN1_N + 1). So, ECLKcap1 = HCLK4 / (VIN1_N + 1)
[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, ECLKjpg = 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, ECLKgpio = 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]	HCLK1_N	AHB1 Clock Divider This field defines the clock divide number for clock divider to generate the HCLK1 for AHB1 bus. The actual clock divide number is (HCLK1_N + 1). So, HCLK1 = CPUCLK / (HCLK1_N + 1) CPUCLK:HCLK1 only is 1:1 or 2:1
[14:12]	VIN0_N	Capture_0 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 (VIN0_N + 1). So, ECLKcap0 = HCLK4 / (VIN0_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, HCLK2 = HCLK / (HCLK234_N + 1) HCLK3 = HCLK / (HCLK234_N + 1) HCLK4 = HCLK / (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, CPUCLK = SYS_CLK / (CPU_N + 1)

APLL Control Register（APLLCON）

Register	Address	R/W	Description	Reset Value
APLLCON	CLK_BA + 20	R/W	APLL Control Register	0x0000_5118

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

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:17]	PLLWait_CNT	PLL waiting cycle numbers
[16]	PLLWaitFEn	PLL Wait Enable 0 = PLL wait enable 1 = PLL wait disable
[15]	BP	PLL ByPass Control 0 = PLL at Normal mode (Default) 1 = By pass Fin (i.e. Fout = XIN)
[14]	PD	Power Down Mode 0 = PLL at Normal mode (Default) 1 = PLL at power-down mode
[13]	Reserved	Reserved
[12:11]	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: $OutputDivider = 2^{OUT\_DV[0]+2*OUT\_DV[1]}$ This field connected to pin OD of PLL directly.

[10:7]	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: $InputDivider = IN\_DV$ This field connected to pin N of PLL directly.
[6:0]	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: $FeedbackDivider = FB\_DV$ 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. $1MHz < \frac{FIN}{NR} < 50\,MHz$
2. $500\,MHz \le FOUT \times NO \le 1500\,MHz$
3. $M \ge 4; N \ge 2$
4. $M = FB\_DV[6]*128 + FB\_DV[5]*64 + FB\_DV[4]*32 + FB\_DV[3]*16 + FB\_DV[2]*8 + FB\_DV[1]*4 + FB\_DV[0]*2$

   $N = IN\_DV[3]*8 + IN\_DV[2]*4 + IN\_DV[1]*2 + IN\_DV[0]*1$

FOUT	Output Clock Frequency
FIN	Input (Reference) Clock Frequency
NR	Input Divider (IN_DV)
NF	Feedback Divider (FB_DV)
NO	OUT_DV = "00" : NO = 1 OUT_DV = "01" : NO = 2 OUT_DV = "10" : NO = 4 OUT_DV = "11" : NO = 8

Default Setting

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

The default value: 0x0_5118

FIN = 27 MHz

NR = 4′b0010 => 2

NF = 7′b0011000 => 48

NO = 2′b10 => 4

***FOUT* = 27 x 48/2 x 1/4 = 162 MHz**

UPLL Control Register（UPLLCON）

Register	Address	R/W	Description	Reset Value
UPLLCON	CLK_BA + 24	R/W	UPLL Control Register	0x0000_5118

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
BP	PD	Reserved	OUT_DV		IN_DV		
7	6	5	4	3	2	1	0
IN_DV	FB_DV						

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	BP	PLL ByPass Control 0 = PLL at Normal mode (Default) 1 = By pass Fin (i.e. Fout = XIN)
[14]	PD	Power Down Mode 0 = PLL at Normal mode (Default) 1 = PLL at power-down mode
[13]	Reserved	Reserved
[12:11]	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: $OutputDivider = 2^{OUT\_DV[0]+2*OUT\_DV[1]}$ This field connected to pin OD of PLL directly.

[10:7]	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: $InputDivider = IN\_DV$ This field connected to pin N of PLL directly.
[6:0]	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: $FeedbackDivider = DB\_DV$ 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. $1MHz < \frac{FIN}{NR} < 50\,MHz$
2. $500\,MHz \le FOUT \times NO \le 1500\,MHz$
3. $M \ge 4; N \ge 2$
4. $M = FB\_DV[6]*128 + FB\_DV[5]*64 + FB\_DV[4]*32 + FB\_DV[3]*16 + FB\_DV[2]*8 + FB\_DV[1]*4 + FB\_DV[0]*2$

   $N = IN\_DV[3]*8 + IN\_DV[2]*4 + IN\_DV[1]*2 + IN\_DV[0]*1$

FOUT	Output Clock Frequency
FIN	Input (Reference) Clock Frequency
NR	Input Divider (IN_DV)
NF	Feedback Divider (FB_DV)
NO	OUT_DV = "00" : NO = 1 OUT_DV = "01" : NO = 2 OUT_DV = "10" : NO = 4 OUT_DV = "11" : NO = 8

Default Setting

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

The default value: 0x1_1118

FIN = 27 MHz

NR = 4′b0010 => 2

NF = 7′b0011000 => 48

NO = 2′b10 => 4

***FOUT* = 27 x 48/2 x 1/4 = 162 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 = HCLKvin 01_0001 = HCLKge 01_0010 = HCLKsic 01_0011 = HCLKnand 01_0100 = HCLKusbd 01_0101 = HCLKi2s 01_0110 = HCLKspu 01_0111 = HCLKvpost 01_1000 = HCLKtic 01_1001 = Reserved 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 = ECLKvin 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

AHB Devices Clock Enable Control Register2 （AHBCLK2）

Register	Address	R/W	Description	Reset Value
AHBCLK2	CLK_BA + 34	R/W	AHB IPs Clock Enable Control Register2	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				SEN1_CKE	VIN1_CKE	EDMA6_CKE	EDMA5_CKE

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	SEN1_CKE	Sensor1 Interface Clock Enable Control 0 = Disable 1 = Enable
[2]	VIN1_CKE	Video_1 In Clock Enable Control (Also is Video In engine clock enable control) 0 = Disable 1 = Enable
[1]	EDMA6_CKE	EDMA Controller Channel 6 Clock Enable Control 0 = Disable 1 = Enable
[0]	EDMA5_CKE	EDMA Controller Channel 5 Clock Enable Control 0 = Disable 1 = Enable

Clock Divider Register 5 (CLKDIV5)

Register	Address	R/W	Description	Reset Value
CLKDIV5	CLK_BA+ 38	R/W	Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
TOUCH_N					TOUCH_S		
23	22	21	20	19	18	17	16
TOUCH_S		SENSOR_N1			SENSOR_S1		
15	14	13	12	11	10	9	8
SENSOR_S1			PWM_N				
7	6	5	4	3	2	1	0
PWM_N			PWM_S				

Bits	Descriptions	
[31:27]	TOUCH_N	TOUCH Engine Clock Divider Bits [4:0] This field defines the clock divide number for clock divider to generate the engine clock for TOUCH ADC controller. So, TOUCH_CLK = TOUCH_SrcCLK / (TOUCH_N [4:0] + 1)
[26:22]	TOUCH_S	TOUCH Engine Clock Source Selection This field selects which clock is used to be the source of engine clock for TOUCH ADC controller. 00: TOUCH_SrcCLK = XIN 01: TOUCH_SrcCLK = Reserved 10: TOUCH_SrcCLK = ACLKOut 11: TOUCH_SrcCLK = UCLKOut
[21:18]	SENSOR_N1	Sensor1 Clock Divide This field defines the clock divide number for clock divider to generate the sensor clock. The actual clock divide number is (SENSOR_N1 + 1). So, SEN_CLK1 = SEN_SrcCLK1 / (SENSOR_N1 + 1)

[17:16]	SENSOR_S1	**Sensor1 Clock Source Selection** This field selects which clock is used to be the source of sensor clock. 00: SEN_SrcCLK1 = XIN 01: SEN_SrcCLK1 = Reserved 10: SEN_SrcCLK1 = ACLKOut 11: SEN_SrcCLK1 = UCLKOut
[15:13]		**Sensor1 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 SENSOR_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
[12:5]	PWM_N	**PWM Engine Clock Divider Bits [7:0]** This field defines the clock divide number for clock divider to generate the engine clock for PWM. So, PWM_CLK = PWM_SrcCLK / (PWM_N [7:0] + 1)
[4:3]	PWM_S	**PWM Engine Clock Source Selection** This field selects which clock is used to be the source of engine clock for PWM controller. 00: PWM_SrcCLK = XIN 01: PWM_SrcCLK = Reserved 10: PWM_SrcCLK = ACLKOut 11: PWM_SrcCLK = UCLKOut

[2:0]		**PWM 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 UART0_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

6.4 SDRAM Controller 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.

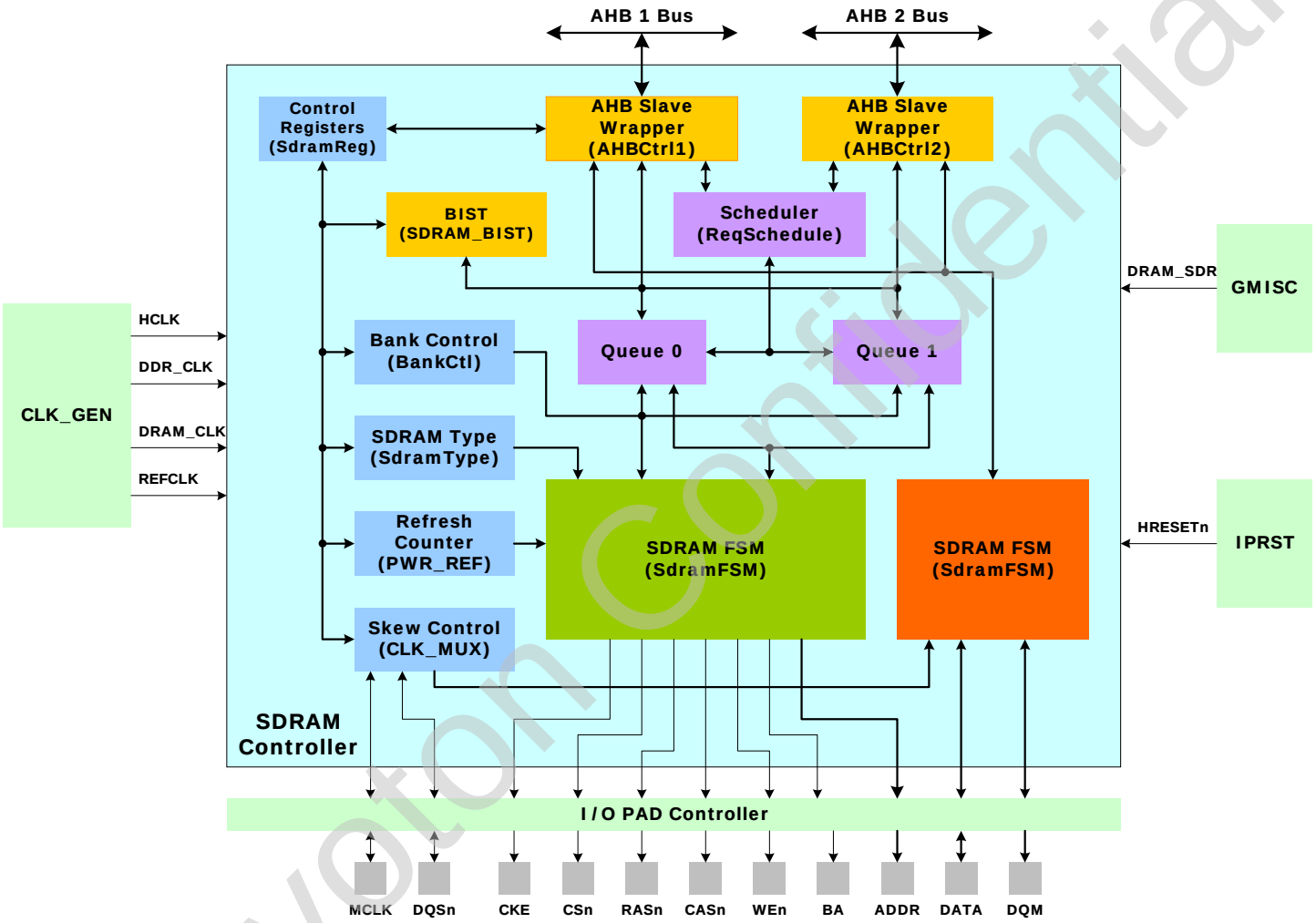
For performance and function issue, the SDRAM controller also supports the proprietary Enhanced-AHB. The EAHB add the down-count address mode, byte-enable signal and explicit burst access number. The explicit access number function is reached by modify the HBURST signal to EHBURST and it represent the 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.

6.4.1 Block Diagram



6.4.2 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.

6.4.3 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 SDRM 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.

6.4.4 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	

6.4.5 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

6.4.6 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.

6.4.6.1 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

6.4.6.2 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	14x10	R	12	11	26	25	23	24	13	22	21	20	19	18	17	16	15	14

64Mx16																		
		C						AP	10	9	8	7	6	5	4	3	2	1

6.4.6.3 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.

6.4.7 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
TESTCR	SDRAM_BA + 40	R/W	SDRAM test control register	0x0000_0000
TSTATUS	SDRAM_BA + 44	R	SDRAM test status register	0x0000_0000
TFDATA	SDRAM_BA + 48	R	SDRAM test fail data	0x0000_0000
TGDATA	SDRAM_BA + 4C	R	SDRAM test gold data	0x0000_0000
TFADDR	SDRAM_BA + 50	R	SDRAM test fail address	0x0000_0000
DBGREG1	SDRAM_BA + 70	R	SDRAM Debug Register 1	0x0000_0001

6.4.8 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
AutoAlign_PHASE[3:0]				Allow_DDR1Gb	AutoAlign_en	Reserved	
23	22	21	20	19	18	17	16
Reserved			AutoAlign_RST	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
[27]	Allow_DDR1Gb	Allow master to access DDR1Gb(64M*16bit) This bit can be used to avoid DRAM state machine error when BA[2] does not connect to DRAM and access address on BA[2] =1 range. 0: DRAM controller access max. address = 64MB. 1: DRAM controller access max. address = 128MB(64M*16bit)
[26]	AutoAlign_en	Auto-Align enable for 130MHz DDRII DLL lock only This bit is used for DRAM data input latch scheme. While DRAM CLK is greater then 130MHz, the DRAM data input latch cycle may be delayed to next cycle at worst case. In such case, this bit cab be used to calculate data latch timing automatically. 0: no Auto-Align scheme 1: enable Aotu-Align scheme
[25:21]	Reserved	
[20]	AutoAlign_RST	Auto-Align RESET signal This bit is used to reset AutoAlign_PHASE[3:0] state machine. 0 = no RESET autoAlign_PHASE[3:0]. 1 = RESET autoAlign_PHASE[3:0] and autoAlign_PHASE will be re-calculated at next cycle.

[19]	OEDelay	**Output Enable Delay Half MCLK** This bit controls the data output enable signal. If setting high, the data output enable will be turned off earlier by half MCLK. 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 sample read data based on the following timing: If CL is 2, the read data output latency will be $2*t_{CK}+t_{AC}$. 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 W55FA95 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. [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. 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 W55FA95, 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 W55FA95, 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.

<table>
<tr>
<td>[6:4]</td>
<td>LATENCY</td>
<td>
CAS Latency
This field defines the CAS latency parameter of external SDRAM device.
In W55FA95, 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>
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.
</td>
</tr>
<tr>
<td>[3]</td>
<td>BrstType</td>
<td>
Burst Type
This bit indicates the burst type of SDRAM device is sequential or interleaved.
In W55FA95, the SDRAM controller only support sequential burst type and this bit will be fixed at 1′b0.
0 = Sequential burst type. (Default)
1 = Reserved.
</td>
</tr>
</table>

[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	DLLEN

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]	DLLEN	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. $^{t}WTR = t_{HCLK} * (tWTR+1)$ 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. $^{t}RRD = t_{HCLK} * (tRRD+1)$ 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. $^{t}RC = t_{HCLK} * (tRC+1)$ 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

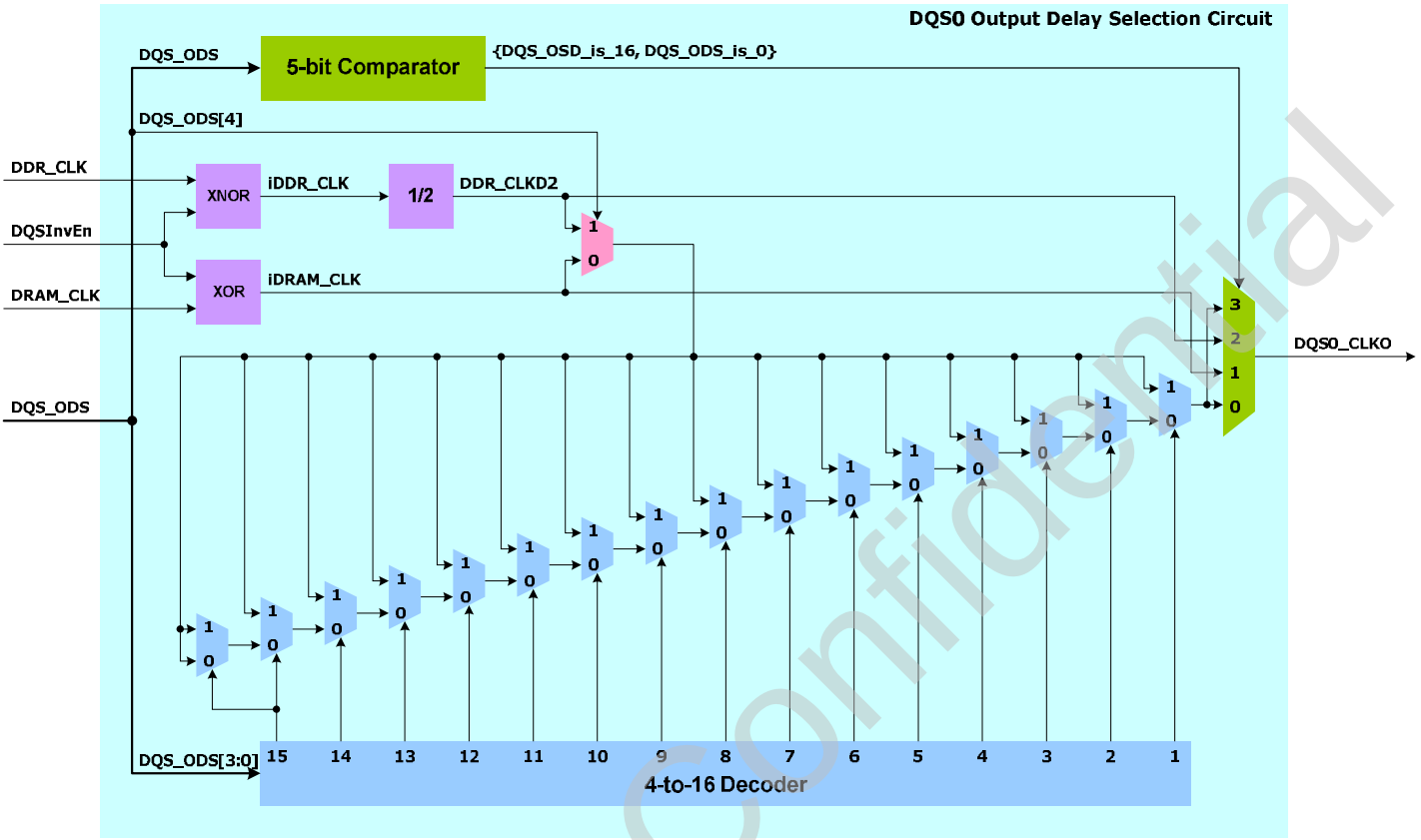
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		DQSInvEn	DQS1_ODS				
7	6	5	4	3	2	1	0
Reserved			DQS0_ODS				

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	0x0088_8800

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
DQS_PHASE_SKEW				Reserved			
15	14	13	12	11	10	9	8
DQS1_SKEW				DQS0_SKEW			
7	6	5	4	3	2	1	0
DCLK_DS				DCLKSrcSel	MCLK_ODS		

Bits	Descriptions	
[31:24]	**Reserved**	**Reserved**
[23:20]	DQS_PHASE_SKEW	**DQS PHASE counter delay selection** **These bits control DQS_PHASE counter delay. It may only use default value for DQS_PHASE count.**
[19:16]	Reserved	Reserved
[15:12]	**DQS1_SKEW**	**DQS1 input latch Delay Selection** **This field controls the DQS1 input delay selection circuit to generate a clock signal DQS1_CLKIn. DQS01_CLKIn is used to sample the data bits [15:8] 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:** $DQS1\_CLKIn\ delay = DQS1\_SKEW * Delay_{CLKMUX}$ **$Delay_{CLKMUX}$: It's the gate delay of a CLKMUX gate.**
[11:8]	**DQS0_SKEW**	**DQS0 Input latch Delay Selection** **This field controls the DQS0 input delay selection circuit to generate a clock signal DQS0_CLKIn. DQS0_CLKIn is used to sample the data bits [7:0] 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:** $DQS0\_CLKIn\ delay = DQS0\_SKEW * Delay_{CLKMUX}$ **$Delay_{CLKMUX}$: 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.

SDRAM Test Control Register (TESTCR)

The SDRAM test control registers is used to do auto test for SDRAM device.

Register	Address	R/W	Description	Reset Value
TESTCR	SDRAM_BA + 40	R/W	SDRAM Test Control Register	0x0000_0000

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

Bits	Descriptions	
[31]	STATUS_CLR	**Test Status Clear** This bit is used to clear the previous test result. This bit will auto return to 0 after the previous test result is cleared. 0 = Keep previous test result. (Default) 1 = Clear previous test result.
[30]	TEST_EN	**Connection Test Enable** This bit is to enable the connection test to check if CS_, CKE and DQM connections between SDRAM controller and SDRAM device are O.K. Write 1′b1 to this bit enables the connection test. And after finishing it, the SDRAM controller will enable BIST test automatically for SDRAM device. Write 1′b0 to this bit disables the connection test. If BIST_EN is also 0, the BIST test will be disabled too. 0 = Disable connection test. (Default) 1 = Enable connection test.

[29]	BIST_EN	SDRAM BIST Enable This bit is to enable the SDRAM controller to do BIST test for SDRAM device. The SDRAM BIST test also could be enabled by set TEST_EN high. 0 = Disable SDRAM BIST. The SDRAM BIST is disabled by clear TEST_EN and BIST_EN both to 0. (Default) 1 = Enable SDRAM BIST. The SDRAM BIST can be enabled by setting this bit or TEST_EN bit high.
[28]	MARCH_C	MARCH_C Algorithm Used This bit indicates which algorithm is used in SDRAM BIST test. The SDRAM controller implements two different BIST algorithms, March-C and MAST+. The test sequence for MAST+ is {↓(w0), ↑(r0,w1), ↓(r1, w0)} while March-C is {↓(w0), ↑(r0,w1), ↑(r1,w0), ↓(r0, w1), ↓(r1, w0), ↑(r0))} 0 = MAST+ algorithm is used for SDRAM BIST. (Default) 1 = March-C algorithm is used for SDRAM BIST.
[27]	Reserved	Reserved
[26:0]	MAX_ADDR	Maximum Test Address This filed indicate the maximum test address. The BIST is access on 32-bits mode and test range is from address 0 to MAX_ADDR.

SDRAM Test Status Register（TSTATUS）

Register	Address	R/W	Description	Reset Value
TSTATUS	SDRAM_BA + 44	R/W	SDRAM Test Status Register	0x0000_0000

31	30	29	28	27	26	25	24
TEST_BUSY	CON_BUYS	BIST_BUSY	TEST_FAIL	CON_FAIL	BIST_FAIL	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]	TEST_BUSY	Test BUSY This bit indicates the SDRAM test is on going or finished. The test includes connection test and SDRAM BIST test. 0 = Test finished. 1 = Test is on going.
[30]	CON_BUSY	Connection Test Busy This bit indicates the connection test is on going or finished. 0 = Connection test finished. 1 = Connection test is on going.
[29]	BIST_BUSY	BIST Test Busy This bit indicates the SDRAM BIST test is on going or finished. 0 = SDRAM BIST test finished. 1 = SDRAM BIST test is on going.

[28]	TEST_FAIL	**Test Failed** This bit indicates if the SDRAM test failed or succeeded. The test includes connection test and SDRAM BIST test. User checks this bit after the TEST_BUSY is 0. 0: Test is OK. 1: Test failed. The fail may be connection test fail or SDRAM BIST test fail. The test stopped while the first error occurred.
[27]	CON_FAIL	**Connection Test Failed** This bit indicates if the connection (CKE, CS_ and DQM) test failed or succeeded. User checks this bit after the CON_BUSY is 0. 0: Connection test is OK. 1: Connection test failed.
[26]	BIST_FAIL	**BIST Test Failed** This bit indicates if the SDRAM BIST test failed or succeeded. User checks this bit after the BIST_BUSY is 0. The first checked error is record on registers TFADDR and TFDATA. 0: SDRAM BIST test is OK. 1: SDRAM BIST test failed.
[25:0]	Reserved	Reserved

SDRAM Test Fail Data Registers（TFDATA）

Register	Address	R/W	Description	Reset Value
TFDATA	SDRAM_BA + 48	R/W	SDRAM Test Fail Data Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	TFDATA	Test Failed Data This field keeps the SDRAM BIST first checked fail data. This failed data is only valid when BIST_FAIL flag is high.

SDRAM Test Gold Data Registers（TGDATA）

Register	Address	R/W	Description	Reset Value
TGDATA	SDRAM_BA + 4C	R/W	SDRAM Test Gold Data Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	TGDATA	Test Gold Data This field keeps the SDRAM BIST first checked gold data.

SDRAM Test Fail Address Register（TFADDR）

Register	Address	R/W	Description	Reset Value
TFADDR	SDRAM_BA + 50	R/W	SDRAM Test Fail Address Register	0x0000_0000

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

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:0]	TFADDR	Test Failed Address This field keeps the SDRAM BIST first checked fail address. This failed address is only valid when BIST_FAIL flag is high.

SDRAM Debug Register 1 (DBGREG1)

Register	Address	R/W	Description	Reset Value
DBGREG	SDRAM_BA + 70	R	SDRAM Debug Register 1	0x0000_0001

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

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27:0]	FSM_CSTATE	FSM Current State This debug register indicates the current state of SDRAM controller main FSM.

6.5 OpenVG 2D Vector Graphics Engine

6.5.1 Introduction

OpenVG is a high performance graphics processor IP core, and is easily implemented to user SoC and software environment. It is carefully designed to meet embedded graphics requirements such as rich GUI or gaming, and suitable for mobile phone, digital AV equipments or gaming devices.

OpenVG graphics core delivers various functions required for 2D vector graphics and conforms to OpenVG 1.1 standard. OpenVG has passed Conformance Test conducted by the Khronos Group, with no error.

6.5.2 Outline of Features

6.5.2.1 Key Features

Maximum display color	RGB = 888
Color Buffer size	2048 x 2048 (can be allocated in 8K x 8K size Color Buffer)
2D function (Sprite rendering)	Frame Buffer type sprite Rotation, stretch, deformation BitBlt functions, Super sample full Anti-aliasing (FSAA) Line rendering, point rendering Alpha-blending Point sampling, Bi-linear filtering Indexed color Texture, Compressed texture image
Vector functions	Path data processing (Internal operation precision: IEEE standard single precision float) Path primitives: Close, Move, Line, Cubic Bezier, Quadratic Bezier, Elliptical Arc Rendering mode: FILL, STROKE, (or both) FLL rendering: RULE (Even, Odd, Non-Zero, Zero) STROKE rendering: CAP(Butt, Round, Square), JOIN(Bevel, Round, Miter), Dash Paint mode: Color, Radial Gradient, Linear Gradient, Pattern Image Filter Color Matrix operation, Convolution filter, Gaussian Blur, Lookup Table operation Pixel Operation Scissor, Color Transform, Blending, Anti-aliasing (2x2, 4x4), Multi-sample fill scene anti-aliasing (MSAA)
Other functions	Function to copy arbitrary rectangular area to the buffer with color format conversion Display List Engine

6.5.3 Rasterizer and Pixel Operation Features

6.5.3.1 Rasterizer Feature

Scissor

Function	Description
Scissor Rectangle	Up to 32 scissoring rectangles are supported.

6.5.3.2 Pixel Operation Features

Color Buffer

Function	Description
Buffer size	Horizontal size: 8 to 2048 pixels in the unit of 8 pixels Vertical size: 1 to 2048 pixels in the unit of 1 pixel
Color format	32-bit(8888): RGBA, ARGB, BGRA, ABGR, RGBX, XRGB, BGRX, XBGR 16-bit(5551): RGBA, BGRA 16-bit(1555): ARGB, ABGR 16-bit(4444): RGBA, ARGB, BGRA, ABGR 16-bit(565): RGB, BGR 8-bit: L8, A8 4-bit: A4 1-bit: A1, BW1

Alpha Mask Buffer

Function	Description
Buffer size	Horizontal size: 8 to 2048 pixels in the unit of 8 pixels Vertical size: 1 to 2048 pixels in the unit of 1 pixel （The Color Buffer size is the same as what is specified to back buffer）
Alpha format	8-bit: A8 4-bit: A4 1-bit: A1

Chroma key

Function	Description
Transparent color for texture image	One transparent color each for two texture units can be assigned. The chroma keying operation is applied to source pixels prior to blending operation.
Transparent color for paint color	As for the transparent color, R, G, B or A component can be specified independently.

Alpha Test

Function	Description
Transparent color for texture image	One transparent color each for two texture units can be assigned. The chroma keying operation is applied to source pixels prior to blending operation.

Color space conversion

Function	Description
Conversion	Gamma and Inverse-Gamma conversions PreMultiply and UnPreMultiply conversions Color Transform (Scale, Bias) LookUpTable conversion RGBA to RGBA (channel independent) Index to RGBA

Coverage

Function	Description
Coverage operation	Enables/Disables Coverage Test function. Supports constant value coverage.

Image and Patten (Texture functions)

Function	Description
Texture Wrap	Supports Repeat/Reflect/Pad.
Texture size	Can be specified from 1 to 8192 by the unit of 1
Texture format	32-bit(8888): RGBA, ARGB, BGRA, ABGR, RGBX, XRGB, BGRX, XBGR 16-bit(5551): RGBA, BGRA 16-bit(1555): ARGB, ABGR 16-bit(4444): RGBA, ARGB, BGRA, ABGR 16-bit(565): RGB, BGR 16-bit(YUV): YUV422 8-bit: L8, A8, Paletted 4-bit: A4, LVQ, Paletted 2-bit: Paletted 1-bit: A1, BW1, Paletted
Texture cache size	Cache line dimension : Square (8 x 8 texel tile unit) 16-bit or less /texel single texture mode 128 tiles 16-bit or less /texel dual texture mode 64 tiles per texture unit 32-bit/texel single texture mode 64 tiles 32-bit/texel dual texture mode 32 tiles per texture unit Cache line dimension : Horizontal (up to 16-bit/texel: cache line = 64 x 1 texel) (32-bit/texel: cache line = 32 x 1 texel) Single texture mode 128 lines Dual texture mode 64 lines per texture unit
Number of texture units	Two units: Texture unit #0 for Paint and #1 for Image
Texture filter	Bi-linear filter
Color conversion	Can be converted by LUT for each RGBA channel independently Channel A value can be replaced by R value. RGB values can be clamped by A value PreMultiply conversion is supported

Blending

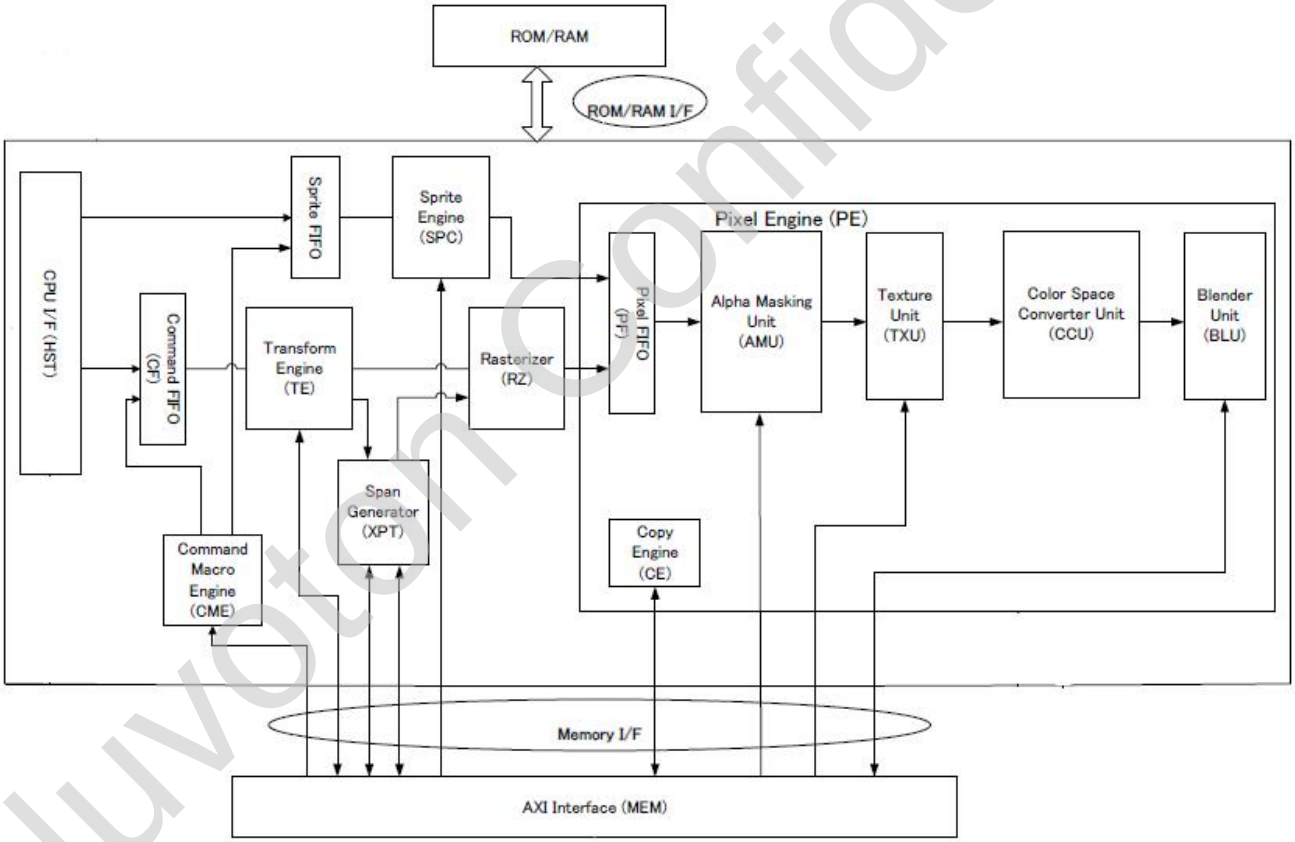
Function	Description
Blending	Supports all functions of both OpenVG1.1 and OpenGL ES 1.1 & 1.1 Extension.

Anti-aliasing

Function	Description
Anti-aliasing	Supports 2x2 and 4x4 sampling mode.

6.5.4 Top level block diagram

The following figure illustrates a top-level block diagram and a typical system connection.



6.6 GVE 2D Graphics Engine

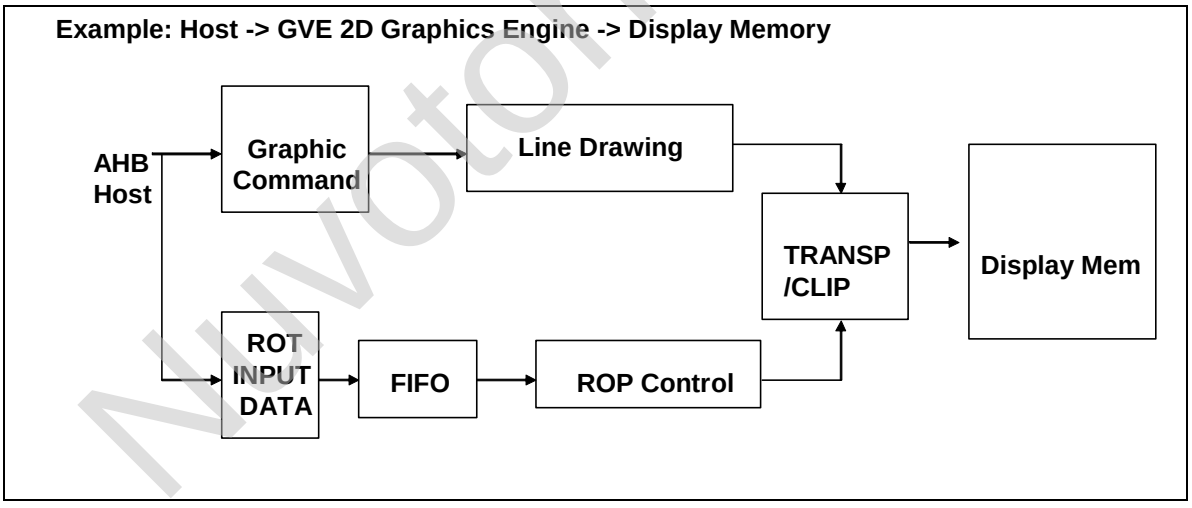
GVE contains two acceleration engines. The first one is 2D computer graphics processing (2D GE) and the second is for video data processing (VPE). It is specially designed to improve the performance of computer GUI functions, such as BitBLTs and Bresenham Line Draw. Meanwhile, GVE can handle the 2D image rotation, bilinear up/down-scaling, and data format conversion. Also a 3X3 DDA filter is used for better image quality. 2D supports the Command-Queue DMA and VPE supports the MMU memory mapping, both with their MMU tables or DMA link lists on system memories.

6.6.1 2D Graphics Engine

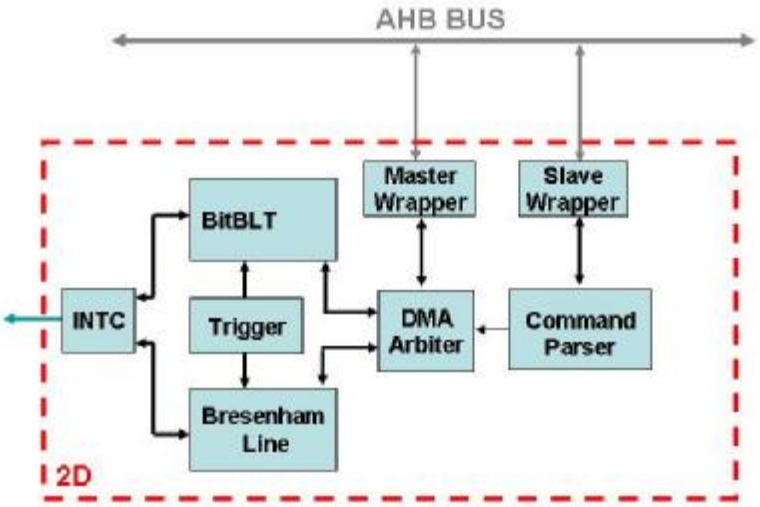
A 32-bit 2D Graphics Engine (2D GE) is specially designed to improve the performance of graphics processing. It can accelerate the operation of individual GUI functions such as BitBLTs and Bresenham Line Draw to operate at all pixel depths including 4/8/15/16/32 bit-per-pixel. A pixel is the smallest addressable screen element as defined in Microsoft Windows, and lines and pictures are made up by a variety of pixels. 2D GE is used to speed up graphics performance in pixel data moving and line drawing, as well as accelerating almost all computer graphics Boolean operations to eliminate the CPU overhead. Meanwhile, the functions of rotation and up/down-scaling are implemented in GVE for some special applications. For the image scaling up/down functions, both the programmable horizontal and vertical N/M scaling factors are provided for resizing the image.

For the 2D GE rotation, it can rotate left or right 45, 90 or 180 degrees, and it also supports the flip/flop, mirror or up-side-down pictures. An H/W filter is dedicated for smoothing object during 2D scaling and 45 degree rotation. 2D rotation needs a rotation reference point but VPE does not.

For the VPE rotation, it can rotate left or right by 90 or 180 degrees, and it also supports the flip/flop, mirror or up-side-down pictures.



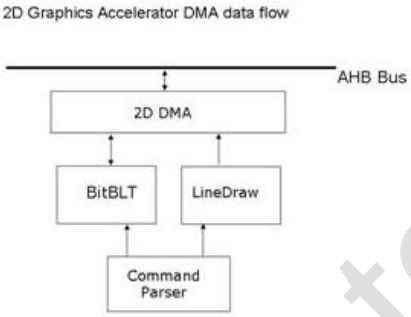
GVE Hardware Architecture

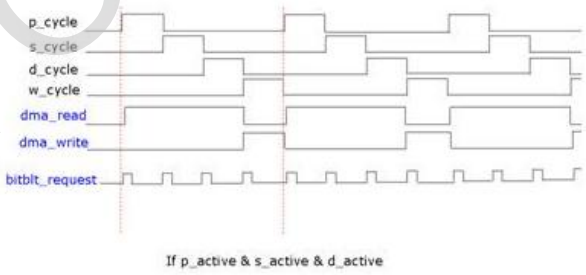


Basically GE has three different operands, i.e., pattern, source, and destination.

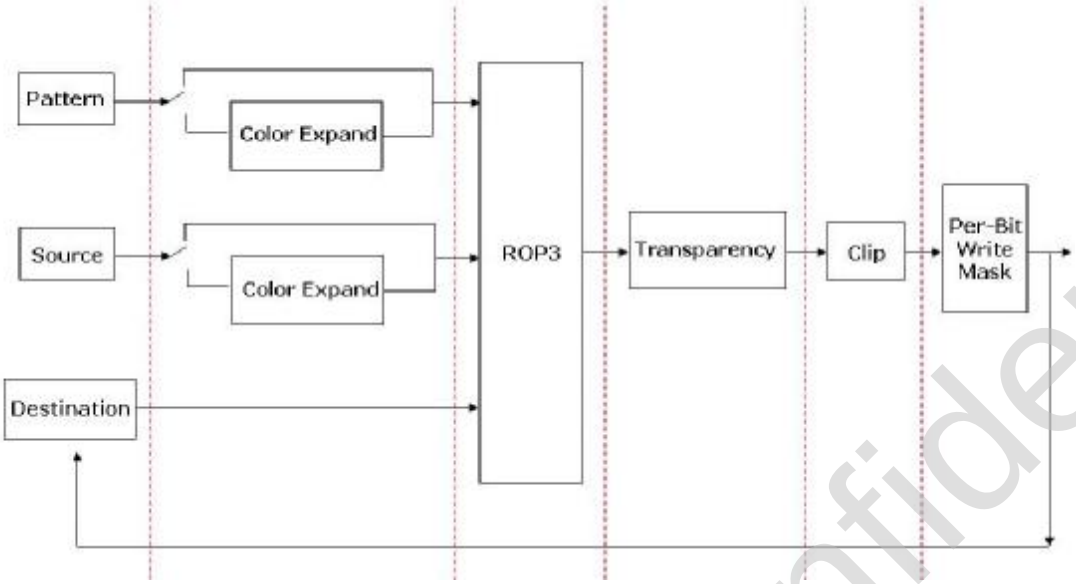
In the following diagrams, the signal p_cycle means it is the time of pattern cycle.

The signal s_cycle means it is the time of source cycle, and d_cycle for destination.

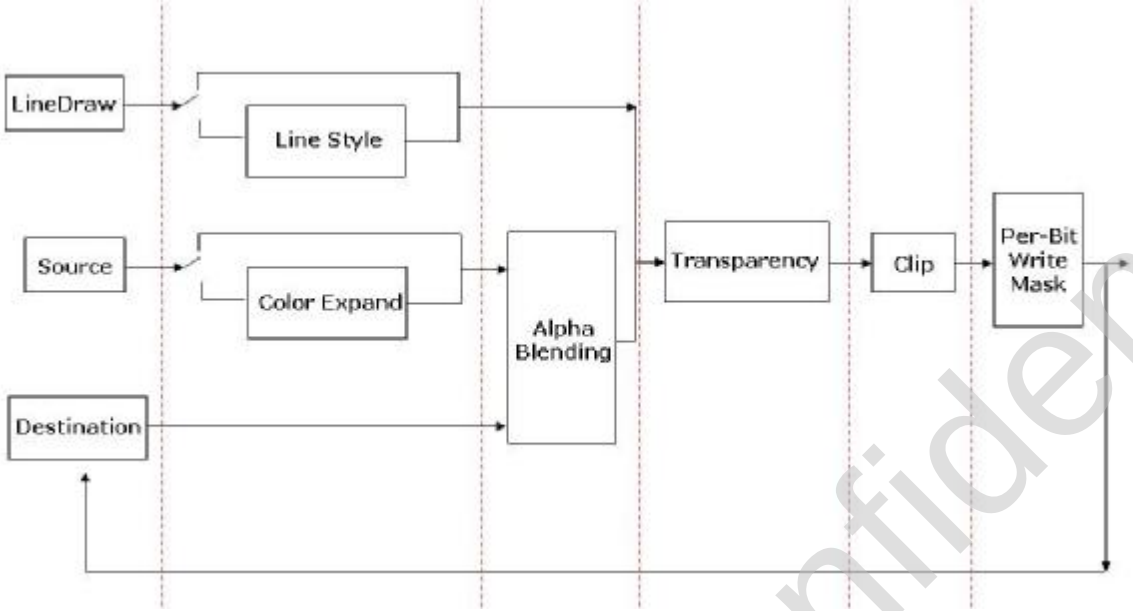




The normal GE dataflow is shown as below.



Another dataflow is shown as below, when the Alpha-Blending is turned on.



6.6.2 Function Descriptions

6.6.2.1 Bit Block Transfer (BitBLT)

BitBLT is the operation that accelerates transfers of data between regions of display memory, or between system memory and display memory. The BitBLT engine operates on three pixel-mapped operands: source, pattern, and destination, with all 256 possible raster operations (ROPs). The Graphics Engine can support several kinds of BitBLT including HostBLT, Pattern BLT, Color/Font Expanding BLT, Transparent BLT, Color/Font Expansion, and Rectangle Fill, etc.

The source map data may reside in display memory or system memory, both can be color or monochrome. The 8×8 pattern map data may reside in display memory when it is color data, or may come from internal pattern register when it is monochrome data. The destination map data must reside in display memory. The resultant destination data generated by the engine is normally written back to the display memory, or it may be read back by the CPU.

6.6.2.2 HostBLT (Between System Memory and Display Memory)

The HostBLT is used to transfer data quickly between system memory and display memory. To transfer data quickly from system memory to display memory, the host may set up a host write BLT to select system memory as source data. Data may be color or monochrome. In monochrome data case for color/font expansion, all ones in the source are expanded to a pixel of foreground color and all zeros are either expanded to a pixel of background color or transparent. Note that color/font expansion can be performed using X/Y addressing only.

To transfer data quickly from display memory to system memory, the host may set up a host read BLT to let the resultant destination data go to the system memory. Note that only source copy operation (SRCCOPY) is available in a host read BLT.

HostBLT is executed through eight 32-bit MMIO data ports for bit block data transfer. The host must perform 32-bit word-aligned accesses when writing data to, or reading data from the Graphics Engine.

6.6.2.3 Pattern BLT

The Pattern BLT is used to accelerate filling of an area of arbitrary size with a repeating pattern in color or mono, and any raster operation may be involved if necessary. The pattern size is 8×8 pixels, chosen for compatibility with Microsoft Windows. Note that all patterns are always aligned to the top-left corner of display memory, but not to the destination area, and the pattern source must be aligned on a boundary, which is equal to the size of the destination.

The 8×8 pattern may be color or monochrome. In the color case, the host first writes the desired pattern to off-screen memory of the frame buffer in a linear fashion. When BitBLT involving pattern is executed, the Graphics Engine will fetch the corresponding row pattern from the off-screen memory and then repetitively copies it to the destination area.

In the monochrome case, the host first writes the desired mono pattern to the internal 8-byte pattern register. The Graphics Engine then repetitively copies it to the destination area, with all ones in the pattern being expanded to a pixel of foreground color and all zeros being either expanded to a pixel of background color or transparent. Note that only X/Y addressing can be used for mono Pattern BLT.

6.6.2.4 Color/Font Expanding BLT

To further accelerate the color/font expanding operation, the host may first write the monochrome bitmap to off-screen memory of the frame buffer, then sets up a Color/Font Expanding BLT. In this kind of BitBLT, the

Graphics Engine fetches the monochrome source data from off-screen memory of the frame buffer, expands it to pixels, and then writes them to the desired destination area. All ones in the source are expanded to a pixel of foreground color and all zeros are either expanded to a pixel of background color or transparent. Note that X/Y addressing fashion is always used in Color/Font Expanding BLT.

6.6.2.5 Transparent BLT

During a BitBLT operation, a certain area of destination may be transparent with the rest being opaque. Monochrome and color transparency are all supported in GVE. In monochrome transparency, either source or pattern data may control the transparency of BitBLT, with all ones being expanded to a pixel of foreground color and all zeros being transparent.

In color transparency, source and destination transparency are both supported, with only one being in effect at a time. Transparent source colors (a color, or color space) do not overwrite the background destination, while only transparent destination colors may be overwritten by the source.

6.6.2.6 Color/Font Expansion

Color/Font Expansion is used to expand a monochrome data to a full depth color pixel (including 8/15/16/32 bit-per-pixel color) with transparency, which greatly accelerates the rendering of text, icons, and other monochrome source objects.

Color/Font Expansion may be implemented using either host write BLT or Color/Font Expanding BLT described above. Note that color/font expansion and mono pattern use the same background and foreground color.

6.6.2.7 Rectangle Fill

A fixed-color rectangle may be filled with either foreground color or background color by using Rectangle Fill. No source area is required. All BitBLT operations are available normally.

6.6.2.8 Raster Operation (ROP)

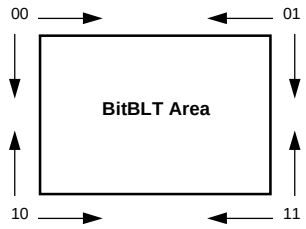
The GE supports all Microsoft 256 ternary raster-operation codes. These ternary raster-operation codes define how the BitBLT combines the bits in a source map with the bits in a brush or pattern map, and the bits in the destination map.

Raster operation is always active during BitBLT and must be loaded with appropriate value.

6.6.2.9 BitBLT Direction

The BitBLT direction indicates the direction in which the X, Y address is stepped across the rectangle. It also defines the starting corner of the transfer. This is significant if the destination rectangle overlaps the source rectangle. One must be certain that the operation progresses so that the source area is not overwritten prior to being used.

BitBLT direction is controlled as shown in the following diagram.



BitBLT Direction

6.6.2.10 Linear and X/Y Addressing

All BitBLT operations may be performed using linear or X/Y addressing. In linear addressing all source start address, destination start address, dimension X, source pitch, and destination pitch are expressed in bytes.

In X/Y addressing, all source start X, destination start X, and dimension X are expressed in pixels. Source pitch and destination pitch are usually the same in X/Y addressing which is defined with free choices, such as 128,160, 320, or above 640.

6.6.2.11 Auto Destination Update

When the Auto Update bit is set to 1, the destination start X, Y (address) registers are automatically updated at the end of each BitBLT operation immediately to the right of the top-right corner of the previous destination area. This is especially usefully at improving Font Expanding BLT where, at most of the time, only source start X, Y (address) registers are needed to update for each BitBLT.

6.6.2.12 Alpha Blending

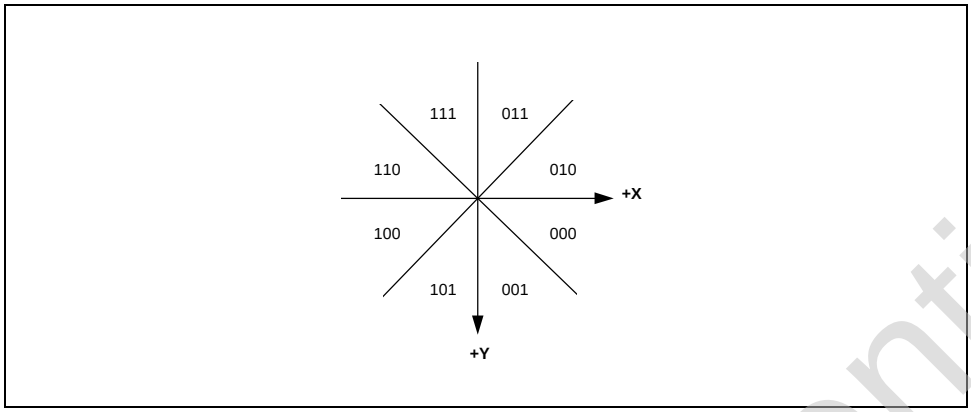
Alpha blending is supported for graphic overlaying.

Two 8-bit alpha-depth blending factors are supported for overlaying by properly programming the alpha blending factors Ks and Kd in Miscellaneous Control register. Ks specify the 8-bit alpha value of source stream, and Kd specifies the 8-bit alpha value of destination stream respectively. Note that Ks + Kd would be <= 256. The blending equation is: [Ps $\times$ Ks + Pd $\times$ Kd]/256, where Ps means the source stream pixels and Pd means the destination stream pixels.

6.6.2.13 Bresenham Line Draw

The Bresenham line drawing algorithm is used to draw a pixel wide solid or textured line from screen coordinates x_1, y_1 to x_2, y_2. To draw a solid line, the foreground color is used to specify color of the line. To draw a textured line, a 16-bit line style is used to specify the pattern of line, with all ones in the style being expanded to a pixel of foreground color and all zeros being either expanded to a pixel of background color or transparent.

The Bresenham line drawing algorithm operates with all parameters normalized to the first octant (octant 0). A 3-bit octant code is specified as shown in the figure as below.



Bresenham Line Draw Octant Encoding

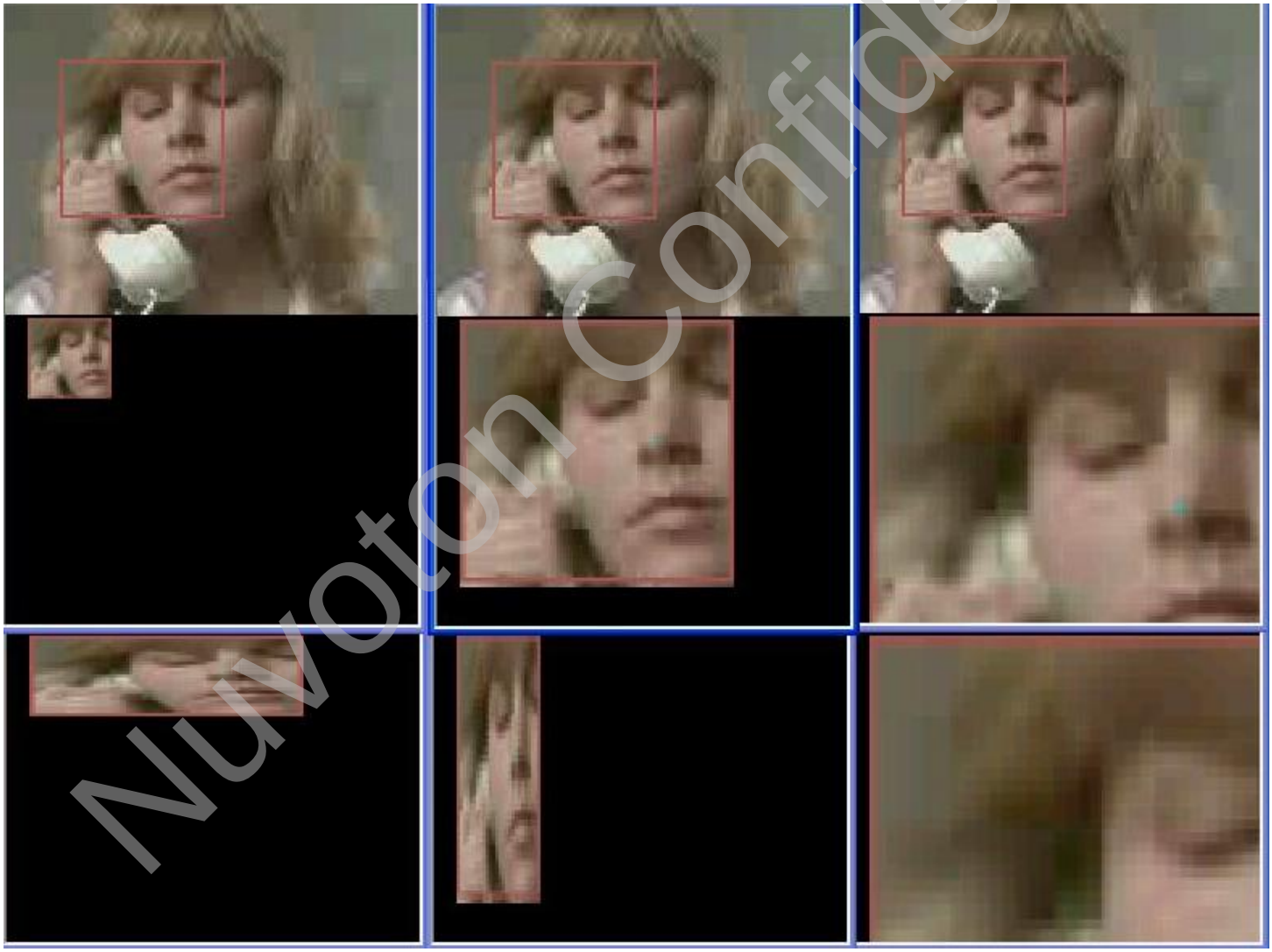
In order to avoid drawing the endpoints of poly-lines twice, the GVE provides a function that inhibits the drawing of the last pixel of the line, and this function is provided for the Bresenham Line Draw. The Bresenham Line Draw operation may be either a draw operation, or mere a move operation. On completion of the Bresenham Line Draw operation, destination start X, Y normally points at the last pixel of the line by setting Auto Destination Update to be 1. It may also points at the original position of the line when Auto Destination Update is 0. Note that the Bresenham Line Draw operation is available only in X/Y addressing mode.

6.6.2.14 Clipping

The clipping function supports clipped drawing writes inside or outside of any rectangular region in display memory during Graphics Engine operation. The GVE supports both rectangle clipping for BitBLTs and line clipping for Bresenham Line. When enabled, the clipping function simply masks writes within or outside of the clipping window. Note that the clipping function is available only in X/Y addressing mode.

6.6.2.15 Rotation and Scaling Up/Down

This function is to support the rotation or scaling up/down in any rectangular region in display memory during Graphics Engine operation. For the 2D GE rotation, it can rotate left or right 45, 90 or 180 degrees, and it also supports the flip/flop, mirror or up-side-down pictures. Just as the rectangle clipping for BitBLTs and line clipping for Bresenham Line, when a clip flag is enabled, the clipping function simply masks writes inside the clipping window. Users can trun on the rectangle clipping functions for their different and special view effect. In image scale up/down function, both programmable horizontal and vertical N/M scaling up/down factors are provided for resizing the image. In order to scale up (N/M, N≥M) or scale down (N/M, N≤M), the value of N can be equal to, greater than, or less than M. Although scaling-down is unlimited, scaling-up supports a range of 1.0 ~ 8.0.



6.6.2.16 Command Queues

Command Queues are groups of 24-word command sets and they are linked by pointers from the start to the

next. The start link list address resides on 2D_SLLACQ (GE_BA + 00C), while the next link list pointer is NLLPCQ (CQ _BA + 00C) as the following table per Command Queue on system memory.

6.6.3 2D Graphics Engine 24-Word Command Set Items Per Command Queue:

Command Set Item	Command Queue Address	Command Set Item Description
TRIG	CQ_BA + 000	Trigger Control Register
XYSOA	CQ _BA + 004	Source Origin Address
TILEXY or VHSF	CQ _BA + 008	TileBLT Width/Height or V/H Scale Factor N/M
NLLPCQ	CQ _BA + 00C	Next Link List Pointer of Command Queues
COUNTER	CQ _BA + 010	Command Queue Read Counter Status
PLS	CQ _BA + 014	Pattern Location Starting Address
BER	CQ _BA + 018	Bresenham Error Term Stepping Constant
BIR	CQ _BA + 01C	Bresenham Initial Error, Pixel Count Major M
2DCMD	CQ _BA + 020	2D Command Control
BCOLOR	CQ _BA + 024	Background Color
FCOLOR	CQ _BA + 028	Foreground Color
TCOLOR	CQ _BA + 02C	Transparency Color
TCMSK	CQ _BA + 030	Transparency Color Mask
XYDOA	CQ _BA + 034	Destination/Display Origin Address
SDPTCH	CQ _BA + 038	Source/Destination Pitch
SSXYL	CQ _BA + 03C	Source Start XY/Linear Address
DSXYL	CQ _BA + 040	Destination Start XY/Linear
DIXYL	CQ _BA + 044	Dimension XY/Linear
CBTL	CQ _BA + 048	Clipping Window Top/Left Boundary
CBBR	CQ _BA + 04C	Clipping Window Bottom/Right Boundary
PTNA	CQ _BA + 050	Pattern A Parameters
PTNB	CQ _BA + 054	Pattern B Parameters
WPMSK	CQ _BA + 058	Write Plane Mask Register
2DMISC	CQ _BA + 05C	2D Miscellaneous Control

CQ_BA means the start address of the current Command Queue on DRAM system memory.

6.6.4 2D Graphics Engine Control Registers Map

6.6.5 GE_BA = 0xB100_C000

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
2D_GETRIG	GE_BA + 000	R/W	Graphics Engine Trigger Control Register	0x0000_0000
2D_GEXYSOA	GE_BA + 004	R/W	Graphics Engine Source Origin Address	0x0000_0000
2D_TILEXY 2D_VHSF	GE_BA + 008	R/W	Graphics Engine TileBLT Width/Height	0x0000_0000
2D_GERRXY 2D_GEXYD2A 2D_SLLACQ	GE_BA + 00C	R/W	Graphics Engine Rotate Reference Point or The Secondary Destination Origin Address Registers or The Start Link List Address of Command Queue	0x0000_0000
2D_GEINTS	GE_BA + 010	R/W	Graphics Engine Interrupt Status Register	0x0000_0000
2D_GEPLS	GE_BA + 014	R/W	Graphics Engine Pattern Location Starting Address	0x0000_0000
2D_GEBER	GE_BA + 018	R/W	GE Bresenham Error Term Stepping Constant Register Graphics Engine Vertical Scale Factor N/M Register	0x0000_0000
2D_GEBIR	GE_BA + 01C	R/W	GE Bresenham Initial Error, Pixel Count Major M Register Graphics Engine Horizontal Scale Factor N/M Register	0x0000_0000
2D_GECMD	GE_BA + 020	R/W	Graphics Engine Command Control Register	0x0000_0000
2D_GEBC	GE_BA + 024	R/W	Graphics Engine Background Color Register	0x0000_0000
2D_GEFC	GE_BA + 028	R/W	Graphics Engine Foreground Color Register	0x0000_0000
2D_GETC	GE_BA + 02C	R/W	Graphics Engine Transparency Color Register	0x0000_0000
2D_GETCM	GE_BA + 030	R/W	Graphics Engine Transparency Color Mask Register	0x0000_0000
2D_GEXYDOA	GE_BA + 034	R/W	Graphics Engine Destination/Display Origin Address	0x0000_0000
2D_GESDP	GE_BA + 038	R/W	Graphics Engine Source/Destination Pitch Register	0x0000_0000
2D_GESSXYL	GE_BA + 03C	R/W	Graphics Engine Source Start XY/Linear Address Register	0x0000_0000
2D_GEDSXYL	GE_BA + 040	R/W	Graphics Engine Destination Start XY/Linear Register	0x0000_0000
2D_GEDIXYL	GE_BA + 044	R/W	Graphics Engine Dimension XY/Linear Register	0x0000_0000
2D_GECBTL	GE_BA + 048	R/W	Graphics Engine Clipping Window Top/Left Boundary	0x0000_0000
2D_GECBBR	GE_BA + 04C	R/W	Graphics Engine Clipping Window Bottom/Right Boundary	0x0000_0000
2D_GEPTNA	GE_BA + 050	R/W	Graphics Engine Pattern A Register	0x0000_0000

2D_GEPTNB	GE_BA + 054	R/W	Graphics Engine Pattern B Register	0x0000_0000
2D_GEWPM	GE_BA + 058	R/W	Graphics Engine Write Plane Mask Register	0x0000_0000
2D_GEMISC	GE_BA + 05C	R/W	Graphics Engine Miscellaneous Control Register	0x0000_0800
2D_GEHBDW0	GE_BA + 060	R/W	Graphics Engine HostBLT Word Data Port0 Register	0x0000_0000
2D_GEHBDW1	GE_BA + 064	R/W	Graphics Engine HostBLT Word Data Port1 Register	0x0000_0000
2D_GEHBDW2	GE_BA + 068	R/W	Graphics Engine HostBLT Word Data Port2 Register	0x0000_0000
2D_GEHBDW3	GE_BA + 06C	R/W	Graphics Engine HostBLT Word Data Port3 Register	0x0000_0000
2D_GEHBDW4	GE_BA + 070	R/W	Graphics Engine HostBLT Word Data Port4 Register	0x0000_0000
2D_GEHBDW5	GE_BA + 074	R/W	Graphics Engine HostBLT Word Data Port5 Register	0x0000_0000
2D_GEHBDW6	GE_BA + 078	R/W	Graphics Engine HostBLT Word Data Port6 Register	0x0000_0000
2D_GEHBDW7	GE_BA + 07C	R/W	Graphics Engine HostBLT Word Data Port7 Register	0x0000_0000

The following registers are used for monitoring control registers of the current Command Queue.

Register	Address	R/W	Description	Reset Value
CQ_GETRIG	GE_BA + 100	R	Graphics Engine Trigger Control Register	0x0000_0000
CQ_GEXYSOA	GE_BA + 104	R	Graphics Engine Source Origin Address	0x0000_0000
CQ _TILEXY CQ _VHSF	GE_BA + 108	R	Graphics Engine TileBLT Width/Height or Graphics Engine V/H Scale Factor N/M Registers	0x0000_0000
CQ _NLLPCQ	GE_BA + 10C	R	Next Link List Pointer Address of Command Queues	0x0000_0000
CQ _COUNTER	GE_BA + 110	R	Graphics Engine Command Queue Read Counter Status	0x0000_0000
CQ _GEPLS	GE_BA + 114	R	Graphics Engine Pattern Location Starting Address	0x0000_0000
CQ _GEBER	GE_BA + 118	R	GE Bresenham Error Term Stepping Constant Register	0x0000_0000
CQ _GEBIR	GE_BA + 11C	R	GE Bresenham Initial Error, Pixel Count Major M Register	0x0000_0000
CQ _GECMD	GE_BA + 120	R	Graphics Engine Command Control Register	0x0000_0000
CQ _GEBC	GE_BA + 124	R	Graphics Engine Background Color Register	0x0000_0000
CQ _GEFC	GE_BA + 128	R	Graphics Engine Foreground Color Register	0x0000_0000
CQ _GETC	GE_BA + 12C	R	Graphics Engine Transparency Color Register	0x0000_0000
CQ _GETCM	GE_BA + 130	R	Graphics Engine Transparency Color Mask Register	0x0000_0000
CQ _GEXYDOA	GE_BA + 134	R	Graphics Engine Destination/Display Origin Address	0x0000_0000
CQ _GESDP	GE_BA + 138	R	Graphics Engine Source/Destination Pitch Register	0x0000_0000
CQ _GESSXYL	GE_BA + 13C	R	Graphics Engine Source Start XY/Linear Address Register	0x0000_0000
CQ _GEDSXYL	GE_BA + 140	R	Graphics Engine Destination Start XY/Linear Register	0x0000_0000
CQ _GEDIXYL	GE_BA + 144	R	Graphics Engine Dimension XY/Linear Register	0x0000_0000
CQ _GECBTL	GE_BA + 148	R	Graphics Engine Clipping Window Top/Left Boundary	0x0000_0000
CQ _GECBBR	GE_BA + 14C	R	Graphics Engine Clipping Window Bottom/Right Boundary	0x0000_0000
CQ _GEPTNA	GE_BA + 150	R	Graphics Engine Pattern A Register	0x0000_0000
CQ _GEPTNB	GE_BA + 154	R	Graphics Engine Pattern B Register	0x0000_0000
CQ _GEWPM	GE_BA + 158	R	Graphics Engine Write Plane Mask Register	0x0000_0000
CQ _GEMISC	GE_BA + 15C	R	Graphics Engine Miscellaneous Control Register	0x0000_0000

6.6.6 2D Graphics Engine Control Registers

Graphics Engine Trigger Control Register

Register	Address	R/W	Description	Reset Value
2D_GETRIG	GE_BA + 000	R/W	2D Graphics Engine Trigger 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						QUEUE	GO

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1]	QUEUE	**2D Graphics Engine Command Queue DMA** 1 = Enable 2D command queue DMA. 0 = Disable 2D command queue DMA.
[0]	GO	**Trigger 2D Graphics Engine Acceleration** 1 = Start GE acceleration, automatically cleared when operation completed. 0 = No acceleration or the acceleration is finished.

Graphics Engine XY Mode Source Memory Origin Address Register

Register	Address	R/W	Description	Reset Value
2D_GEXYSOA	GE_BA + 004	R/W	2D Graphics Engine Source Origin Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Source Origin Starting Address	**Source Origin Starting Address (byte unit)** This 32-bit byte address specifies the starting address of an object or a picture in the display memory when addressed by X/Y mode.

Graphics Engine TileBLT Width / Height Numbers

Register	Address	R / W	Description	Reset Value
2D_TileXY	GE_BA + 008	R/W	2D TileBLT Width X and Tile Height Y 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
Tile Height Y [7:0]							
7	6	5	4	3	2	1	0
Tile Width X [7:0]							

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:8]	Tile Height Y	**8-bit tile height Y value** This divider provides the tile height Y value.
[7:0]	Tile Width X	**8-bit tile width X value** This divider provides the tile width X value.

Graphics Engine Rotate Reference Point or Secondary Destination Origin Address Register or Start Link List Address of Command Queue

Register	Address	R/W	Description	Reset Value
2D_GERRXY	GE_BA + 00C	R/W	Graphics Engine Rotate Reference Point	0x0000_0000
2D_GEXYD2A	GE_BA + 00C	R/W	Secondary Destination Origin Address	0x0000_0000
2D_SLLACQ	GE_BA + 00C	R/W	Start Link List Address of Command Queue	0x0000_0000

(A)

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

Bits	Descriptions	
[26:16]	Rotate Reference Y	**11-bit Rotate Reference Y** For Rotation in X/Y addressing, this register specifies the reference point Y in pixels.
[10:0]	Rotate Reference X	**11-bit Rotate Reference X** For Rotation in X/Y addressing, this register specifies the reference point X in pixels.

(B)

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

Bits	Descriptions	
[31:0]	Secondary Destination Origin Address	**Secondary Destination Origin Starting Address (byte unit)** This 32-bit byte address specifies the secondary destination origin address of an object or a picture in the display memory when addressed by X/Y mode. This value is specified by bytes.

(C)

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

Bits	Descriptions	
[31:0]	Command Queue Start Link List Address	**Command Queue Start Link List Address (byte unit)** This 32-bit byte address specifies the start link list address of an Command Queue in the system memory. This value is specified by bytes.

Graphics Engine Interrupt Status Register

Register	Address	R/W	Description	Reset Value
2D_GEINTS	GE_BA + 010	R/W	Graphics Engine 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						CQ_INTS	INTS

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1]	CQ_INTS	**Command Set complete interrupt status** (in GE Command Queue) 0 = No Command Set interrupt occurs 1 = Command Set interrupt occurs, host writes one to clear INTS.
[0]	INTS	**GE interrupt status** 0 = No interrupt occurs 1 = Interrupt occurs, host writes one to clear INTS.

Graphics Engine Pattern Location Starting Address Register

Register	Address	R/W	Description	Reset Value
2D_GEPLS	GE_BA + 014	R/W	Pattern Location Starting Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Pattern Location	**32-bit Pattern Location (byte unit)** This 32-bit byte address specifies the beginning location of an 8×8 pixel pattern stored in the off-screen memory when in BitBLT operation. This value must be programmed on an M-byte boundary. M=8*8*BPP/8 bytes, where BPP=8/16/32.

Graphics Engine Bresenham Error Term Stepping Constant Register

Register	Address	R/W	Description	Reset Value
2D_GEBER	GE_BA + 018	R/W	Bresenham Error Term Stepping Constant	0x0000_0000
2D_ VSF	GE_BA + 018	R/W	DDA Vertical Scaling Up/Down Factor	0x0000_0000

(A)

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

Bits	Descriptions	
[29:16]	Diagonal Error Increment	**14-bit Diagonal Error Increment** For Bresenham line draw, this register specifies the constant to be added to the Error Term for diagonal stepping (Error > 0). The initial value is (2 ∗ (delta Y - delta X)) after normalization to first octant.
[13:0]	Axial Error Increment	**14-bit Axial Error Increment** For Bresenham line draw, this register specifies the constant to be added to the Error Term for axial stepping (Error < 0). The initial value is (2 ∗ delta Y) after normalization to first octant.

(B)

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

Bits	Descriptions	
[26:16]	VSF_N	11-bit Vertical N Scaling Factor An 11-bit value specifies the numerator part (N) of the vertical scaling factor in graphic engine. The output image height will be equal to the input image height * (N / M).
[10:0]	VSF_M	11-bit Vertical M Scaling Factor An 11-bit value specifies the denominator part (M) of the vertical scaling factor in graphic engine. The output image height will be equal to the input image height * (N / M).

Graphics Engine Bresenham Initial Error, Pixel Count Major –1 Register

Register	Address	R/W	Description	Reset Value
2D_GEBIR	GE_BA + 01C	R/W	Bresenham Initial Error, Pixel Count Major –1	0x0000_0000
2D_ HSF	GE_BA + 01C	R/W	DDA Horizontal Scaling Up/Down Factor	0x0000_0000

(A)

31	30	29	28	27	26	25	24
Initial Error Term [13:8]							
23	22	21	20	19	18	17	16
Initial Error Term [7:0]							
15	14	13	12	11	10	9	8
Line Pixel Count Major -1 [10:8]							
7	6	5	4	3	2	1	0
Line Pixel Count Major -1 [7:0]							

Bits	Descriptions	
[29:16]	Initial Error Term	**14-bit Initial Error Term** For Bresenham line draw, this register specifies the initial Error Term. The initial value is (2 * (delta Y) - delta X) after normalization to first octant.
[10:0]	Line Pixel Count Major -1	**11-bit Line Pixel Count Major -1** For Bresenham line draw, this register specifies the pixel count of major axis

(B)

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

Bits	Descriptions	
[26:16]	HSF_N	11-bit Horizontal N Scaling Factor An 11-bit value specifies the numerator part (N) of the horizontal scaling factor in graphic engine. The output image width will be equal to the input image width * (N / M).
[10:0]	HSF_M	11-bit Horizontal M Scaling Factor An 11-bit value specifies the denominator part (M) of the horizontal scaling factor in graphic engine. The output image width will be equal to the input image width * (N / M).

Graphics Engine Command Control Register

Register	Address	R/W	Description	Reset Value
2D_GECMD	GE_BA + 020	R/W	Graphics Engine Command Control Register	0x0000_0000

31	30	29	28	27	26	25	24
ROP (Raster Operation Code)							
23	22	21	20	19	18	17	16
COMMAND			LINE_STYLE /ZOOM-IN	DRAW /BYP_DDA	CQINT_EN	INT_EN	ADDR_MD
15	14	13	12	11	10	9	8
TRANSPARENCY		MTS	CTS	CTP	AU	CLIP_EN	CLPC
7	6	5	4	3	2	1	0
SDT	SRCS		PDT	XY Octant			DDTO

Bits	Descriptions	
[31:24]	ROP	**ROP Code** It supports all Microsoft 256 Raster Operation Codes. Each raster operation code is an 8-bit value that represents the result of the Boolean operation on pre-defined pattern, source, and destination.
[23:21]	COMMAND	**Graphics Engine Commands** Ÿ 000 = Scaling or Rotation Command Ÿ 001 = Alpha Blending Command Ÿ 010 = BitBLT Acceleration Command Ÿ 011 = BitBLT (write data to the secondary destination address) Ÿ 100 = Bresenham Line Draw, with the last pixel of this line be drawn Ÿ 101 = Bresenham Line Draw, with the last pixel of this line will not be drawn Ÿ 110 = Rectangle Border Drawing Command. Ÿ 111 = Rectangle Border Drawing Command. Rotate with 45 degree
[20]	LINE_STYLE /ZOOM-IN	**(1) Line Style Control (COMMAND: 100, 101, 110, or 111)** Ÿ 0 = Disable line style. It means to draw a solid line.

		Ÿ 1 = Enable line style. It means to draw a texture line. **(2) Zoom-in Effect when the object rotating with 45 degrees** Ÿ 0 = No zoom-in effect or keeping 1:1 ratio, when rotating with 45 degrees. Ÿ 1 = Zoom-in by a factor of 1.414 up-scaling for both width and height.
[19]	**DRAW / BYP_DDA**	**(1) Border/ Line-Drawing Control (COMMAND: 100, 101, 110, or 111)** Ÿ 0 = Disable the border/line-drawing Ÿ 1 = Enable the border/line-drawing or force the missing line to show up. **(2) Scaling Down Filter On/Off (COMMAND=000 and XY_Octant=000)** Ÿ 0 = Turn on H/W DDA filter when scaling down Ÿ 1 = Bypass DDA filter or turn off DDA filter when scaling down
[18]	**CQINT_EN**	**Command Queue Interrupt Enable** Ÿ 0 = Interrupt occurs when all Command Queue command sets are finished Ÿ 1 = Interrupt occurs per each Command Queue command set is finished
[17]	**INT_EN**	**Interrupt Enable of BitBLT/Bresenham Line Draw accelerations.** Ÿ 0 = Disable GE completeness interrupt Ÿ 1 = Interrupt occurs when GE is completed or finished.
[16]	**ADDR_MD**	**Graphics Engine Addressing Mode** Ÿ 0 = Linear addressing mode Ÿ 1 = X/Y addressing mode
[15:14]	**TRANSPX**	**GE Transparency** Ÿ 00 = Disabled Ÿ 01 = Mono transparency Ÿ 10 = Color transparency Ÿ 11 = Reserved
[13]	**MTS**	**Mono Transparency Select** Ÿ 0 = Source Ÿ 1 = Pattern
[12]	**CTS**	**Color Transparency Select** Ÿ 0 = Source pixels control transparency

		Ÿ 1 = Destination pixels control transparency
[11]	CTP	**Color Transparency Polarity** Ÿ 0 = Matching pixels are transparent Ÿ 1 = Matching pixels are opaque
[10]	AU	**Auto Update** Ÿ 0 = Disable Ÿ 1 = Enable. Destination X, Y register is automatically updated at the end of each BitBLT operation.
[9]	CLIP_EN	**Clipping Enable** Ÿ 0 = Disabled Ÿ 1 = Enabled
[8]	CLPC	**Clipping Control** Ÿ 0 = Only pixels inside the clipping rectangle are drawn Ÿ 1 = Only pixels outside the clipping rectangle are drawn
[7]	SDT	**Source Data Type** Ÿ 0 = Color Ÿ 1 = Mono Ÿ Note: Source and pattern data are not allowed to be both in mono format.
[6:5]	SRCS	**Source Data Select** Ÿ 00 = Display memory Ÿ 01 = System memory Ÿ 10 = GE background color Ÿ 11 = GE foreground color
[4]	PDT	**Pattern Data Type** Ÿ 0 = Color (from display memory) Ÿ 1 = Mono (from internal pattern registers)
[3:1]	XY Octant	**XY Octant** Ÿ Determine the directions for BitBLT, Bresenham line, and Rotate. Ÿ 000=Right-down(BitBLT); +X,+Y, |DX| ≥ |DY| (Line); Scaling Up/Down

		Ÿ 001=Right-down(BitBLT); +X,+Y, |DX| < |DY| (Line); Rotate right 45° Ÿ 010=Left-down (BitBLT); +X, –Y, |DX| ≥ |DY| (Line); Rotate left 45° Ÿ 011=Left-down (BitBLT); +X, –Y, |DX| < |DY| (Line); Rotate left 90° Ÿ 100=Right-up (BitBLT); –X,+Y, |DX| ≥ |DY| (Line); Up-Side-Down Ÿ 101=Right-up (BitBLT); –X,+Y, |DX| < |DY| (Line); Rotate right 90° Ÿ 110=Left-up (BitBLT); –X,–Y, |DX| ≥ |DY| (Line); Rotate 180° Ÿ 111=Left-up (BitBLT); –X,–Y, |DX| < |DY| (Line); Mirror or Flop
[0]	**DDTO**	**Destination Data Direction, new destination data to** Ÿ 0 = Display memory Ÿ 1 = System memory

Note:

For 2D BLT, the total rendering W/H area is from (startX, startY) to (startX+W-1, startY+H-1); while for the rectangle/rhomboid border, drawing a border from (startX, startY) to (startX+W, startY+H).

Graphics Engine Background Color Register

Register	Address	R/W	Description	Reset Value
2D_GEBC	GE_BA + 024	R/W	Graphics Engine Background Color	0x0000_0000

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

Bits	Descriptions	
[23:0]	Background Color	**24-bit Background Color** These bits specify the background color for Graphics Engine. Only the corresponding number of bits-per-pixel in the display mode is required in the register. In RGB 8:8:8 color mode, bits 23-16 have the red value, bits 15-8 have the green value, and bits 7-0 have the blue value.

Graphics Engine Foreground Color Register

Register	Address	R/W	Description	Reset Value
2D_GEFC	GE_BA + 028	R/W	Graphics Engine Foreground Color	0x0000_0000

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

Bits	Descriptions	
[23:0]	Foreground Color	**Graphics Engine Foreground Color** These bits specify the foreground color for Graphics Engine. Only the corresponding number of bits-per-pixel in the display mode is required in the register. In RGB 8:8:8 color mode, bits 23-16 have the red value, bits 15-8 have the green value, and bits 7-0 have the blue value.

Graphics Engine Transparency Color Register

Register	Address	R/W	Description	Reset Value
2D_GETC	GE_BA + 02C	R/W	Graphics Engine Transparency Color	0x0000_0000

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

Bits	Descriptions	
[23:0]	Transparency Color	**24-bit Transparency Color** These bits specify the transparency color for Graphics Engine. Only the corresponding number of bits-per-pixel in the display mode is required in the register. In RGB 8:8:8 color mode, bits 23-16 have the red value, bits 15-8 have the green value, and bits 7-0 have the blue value.

Graphics Engine Transparency Color Mask Register

Register	Address	R/W	Description	Reset Value
2D_GETCM	GE_BA + 030	R/W	Graphics Engine Transparency Color Mask	0x0000_0000

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

Bits	Descriptions	
[23:0]	Transparency Color Mask	**24-bit Transparency Color Mask** These bits specify a mask for use in comparison against the transparency color. Only the corresponding number of bits-per-pixel in the display mode is required in the register.

Graphics Engine Destination / Display Memory Origin Address Register

Register	Address	R / W	Description	Reset Value
2D_GEXYDOA	GE_BA + 034	R / W	Destination Display Origin Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Destination Origin Starting Address	**32-bit X / Y Mode Origin Starting Address (byte unit)** This 32-bit byte address specifies the starting address of an object or a picture in the display memory when addressed by X/Y mode. This value is specified by bytes.

Graphics Engine Source / Destination Pitch Register

Register	Address	R / W	Description	Reset Value
2D_GESDP	GE_BA + 038	R/W	Graphics Engine Source/Destination Pitch	0x0000_0000

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

Bits	Descriptions	
[28:16]	Destination Pitch	**Bits 28-16 Destination Pitch** This 13-bit register specifies the destination pitch in bytes in linear addressing mode, and in X/Y addressing by pixels.
[12:0]	Source Pitch	**Bits 12-0 Source Pitch** This 13-bit register specifies the source pitch in bytes in linear addressing mode, and in X/Y addressing by pixels.

Graphics Engine Source Start XY/Linear Addressing Register

Register	Address	R/W	Description	Reset Value
2D_GESSXY	GE_BA + 03C	R/W	Graphics Engine Source Start in X/Y addressing (by pixel unit)	0x0000_0000
2D_GESSL	GE_BA + 03C	R/W	Graphics Engine Source Start in linear addressing (by byte unit)	0x0000_0000

(A)

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

Bits	Descriptions	
[26:16]	Source Start Y	**11-bit Source Start Y** For BitBLTs in X/Y addressing, this register specifies the source start Y in pixels.
[10:0]	Source Start X	**11-bit Source Start X** For BitBLTs in X/Y addressing, this register specifies the source start X in pixels.

(B)

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

Bits	Descriptions	
[31:0]	**Source Linear Starting Address**	**32-bit Source Start Address** For BitBLTs in linear addressing, this 32-bit byte address specifies the beginning location of the source.

Graphics Engine Destination Start XY/Linear Register

Register	Address	R/W	Description	Reset Value
2D_GEDSXY	GE_BA + 040	R/W	Graphics Engine Destination Start X/Y addressing (by pixel unit)	0x0000_0000
2D_GEDSL	GE_BA + 040	R/W	Graphics Engine Destination Start Linear address (by byte unit)	0x0000_0000

(A)

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

Bits	Descriptions	
[26:16]	Destination Start Y	**11-bit Destination Start Y** For BitBLTs and Bresenham line draw in X/Y addressing, this register specifies the destination start Y in pixels.
[10:0]	Destination Start X	**11-bit Destination Start X** For BitBLTs and Bresenham line draw in X/Y addressing, this register specifies the destination start X in pixels.

(B)

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

Bits	Descriptions	
[31:0]	**Destination Linear Starting Address**	**32-bit Destination Linear Starting Address** For BitBLTs in linear addressing mode, this register specifies the destination linear starting address in bytes.

Graphics Engine Height / Width Dimension for XY / Linear Modes Register

Register	Address	R / W	Description	Reset Value
2D_GEDIXYL	GE_BA + 044	R/W	Graphics Engine Dimension in XY (by pixel) or linear mode (by byte)	0x0000_0000

Bits	Descriptions	
[26:16]	Dimension Y	**11-bit Dimension Y** For BitBLTs, this register specifies the height of rectangle in X/Y addressing (by pixel) or in linear addressing (by byte).
[10:0]	Dimension X	**11-bit Dimension X** For BitBLTs, this register specifies the width of rectangle in X/Y addressing (by pixel) or in linear addressing (by byte).

Note:

For 2D BLT, the total rendering W/H area is from (startX, startY) to (startX+W-1, startY+H-1); while for the rectangle/rhomboid border, drawing a border from (startX, startY) to (startX+W, startY+H).

Graphics Engine Clipping Boundary Top / Left Register

Register	Address	R / W	Description	Reset Value
2D_GECBTL	GE_BA + 048	R/W	Graphics Engine Clipping Boundary Top/Left (by X/Y pixel)	0x0000_0000

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

Bits	Descriptions	
[26:16]	Clipping Boundary Top	**11-bit Clipping Boundary Top** This register specifies the top of the clipping rectangle.
[10:0]	Clipping Boundary Left	**11-bit Clipping Boundary Left** This register specifies the left limit of the clipping rectangle.

Graphics Engine Clipping Boundary Bottom / Right Register

Register	Address	R / W	Description	Reset Value
2D_GECBBR	GE_BA + 04C	R/W	Graphics Engine Clipping Boundary Bottom/Right (pixel)	0x0000_0000

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

Bits	Descriptions	
[26:16]	Clipping Boundary Bottom	**11-bit Clipping Boundary Bottom** This register specifies the bottom of the clipping rectangle.
[10:0]	Clipping Boundary Right	**11-bit Clipping Boundary Right** This register specifies the right limit of the clipping rectangle.

Graphics Engine Pattern Group A Register

Register	Address	R/W	Description	Reset Value
2D_GEPTNA	GE_BA + 050	R/W	Graphics Engine Pattern Group A	0x0000_0000

31	30	29	28	27	26	25	24
Pattern3							
23	22	21	20	19	18	17	16
Pattern2							
15	14	13	12	11	10	9	8
Pattern1							
7	6	5	4	3	2	1	0
Pattern0							

Bits	Descriptions	
[31:0]	Pattern Group A	**Bits 31-24 Pattern 3 Register** When pattern is monochrome, this is the 4th line of the 8×8 pattern. **Bits 23-16 Pattern 2 Register** When pattern is monochrome, this is the 3rd line of the 8×8 pattern. **Bits 15-8 Pattern 1 Register** When pattern is monochrome, this is the 2nd line of the 8×8 pattern. **Bits 7-0 Pattern 0 Register** When pattern is monochrome, this is the 1st line of the 8×8 pattern.

Graphics Engine Pattern Group B Register

Register	Address	R/W	Description	Reset Value
2D_GEPTNB	GE_BA + 054	R/W	Graphics Engine Pattern Group B	0x0000_0000

31	30	29	28	27	26	25	24
Pattern7							
23	22	21	20	19	18	17	16
Pattern6							
15	14	13	12	11	10	9	8
Pattern5							
7	6	5	4	3	2	1	0
Pattern4							

Bits	Descriptions	
[31:0]	Pattern Group B	**Bits 31-24 Pattern 7 Register** This is the 8th line of the 8×8 pattern. **Bits 23-16 Pattern 6 Register** This is the 7th line of the 8×8 pattern. **Bits 15-8 Pattern 5 Register** This is the 6th line of the 8×8 pattern. **Bits 7-0 Pattern 4 Register** This is the 5th line of the 8×8 pattern.

Graphics Engine Write Plane Mask Register

Register	Address	R/W	Description	Reset Value
2D_GEWPM	GE_BA + 058	R/W	Graphics Engine Write Plane Mask	0x0000_0000

31	30	29	28	27	26	25	24
RGB555 ID							
23	22	21	20	19	18	17	16
Write Plane Mask [23:16]							
15	14	13	12	11	10	9	8
Write Plane Mask [15:8]							
7	6	5	4	3	2	1	0
Write Plane Mask [7:0]							

Bits	Descriptions	
[31:24]	RGB555 ID	**8-bit ID for RGB555 mode (15 bpp)** **If both this ID is equal to 8'h00 and bit-15 is 0 in "Write Plane Mask Register", then the pixel depth would be 15 bpp when BPP = 2'b01.**
[23:0]	Write Plane Mask	**24-bit Write Plane Mask** These bits specify which bits within each pixel are subject to update by the Graphics Engine. A one enable writing to the corresponding bit plane and a 0 inhibits writing to the corresponding bit plane. Only the corresponding number of bits-per-pixel in the display mode is required in the register.

Graphics Engine Miscellaneous Control Register

Register	Address	R/W	Description	Reset Value
2D_GEMISC	GE_BA + 05C	R/W	Graphics Engine Miscellaneous Control	0x0000_0800

31	30	29	28	27	26	25	24
LINE STYLE PATTERN 1 / Alpha Blending Source Ks							
23	22	21	20	19	18	17	16
LINE STYLE PATTERN 0 / Alpha Blending Destination Kd							
15	14	13	12	11	10	9	8
FIFO STATUS				EMPTY	FULL	BitBltSTS	BUSY
7	6	5	4	3	2	1	0
RST_GE2D	RGB555_EN	BPP		BLT_MD	BLT_TYPE		

Bits	Descriptions	
[31:16]	Line Style Pattern1 Line Style Pattern0	Bits 31-16 Ÿ 16-bit line style pattern for Bresenham line drawing.
[31:16]	Alpha Blending Ks and Kd	Bits 31-24 Bits 23-16 Ÿ 8-bit alpha blending factor Ks for source data and Ÿ 8-bit alpha blending factor Kd for destination data.
[15:12]	FIFO Status	GE FIFO counter status Ÿ 0000 ~ 0111 = FIFO current level Ÿ 0000 = empty and 1000 = full
[11]	EMPTY	FIFO empty status Ÿ 0 = Not empty Ÿ 1 = Empty
[10]	FULL	FIFO full status Ÿ 0 = Not full Ÿ 1 = Full

[9]	**BitBLT_STS**	GE BitBLT operation complete status Ÿ 0 = No complete status occur Ÿ 1 = BitBLT operation complete status occur
[8]	**BUSY**	**GE Operation status** Ÿ 0 = Ready, No GE operation Ÿ 1 = Busy, GE operation is still under working
[7]	**RST_GE2D**	Bit 7 Software Reset GE2D Ÿ 0 = No software reset GE2D. Ÿ 1 = Software reset GE2D.
[6]	**RGB555_EN**	Bit 6 Enable RGB555 Format Ÿ 0 = Enable RGB555 format. Ÿ 1 = Disable RGB555 format.
[5:4]	**Bit Per Pixel**	**Bits 5-4** Graphics Engine Pixel Depth Ÿ 00 = 8-bit byte data format (for linear address mode) Ÿ 01 = 16-bit RGB555/RGB565 format Ÿ 10 = 32-bit RGB888 format Ÿ 11 = 32-bit ARGB8888 format
[3]	**BLT_MODE**	Ÿ 0 = BitBLT type is according to GEC control bits Ÿ 1 = BitBLT type follows BLT_TYPE[2:0] setting as below
[2:0]	**BLT_TYPE**	**Bits 2-0** BitBLT Type Setting Ÿ 000 = HostBLT (write mode) Ÿ 001 = HostBLT (read mode) Ÿ 010 = SolidFillBLT Ÿ 011 = PatternBLT Ÿ 100 = BlockMoveBLT Ÿ 101 = Color/Font Expansion BLT Ÿ 110 = Monochrome Transparent BLT Ÿ 111 = Color Transparent BLT

Graphics Engine HostBLT Data Port Register

Register	Address	R/W	Description	Reset Value
2D_GEHBDW0	GE_BA + 060	R/W	Graphics Engine Host bit block transfer data port0	0x0000_0000
2D_GEHBDW1	GE_BA + 064	R/W	Graphics Engine Host bit block transfer data port1	0x0000_0000
2D_GEHBDW2	GE_BA + 068	R/W	Graphics Engine Host bit block transfer data port2	0x0000_0000
2D_GEHBDW3	GE_BA + 06C	R/W	Graphics Engine Host bit block transfer data port3	0x0000_0000
2D_GEHBDW4	GE_BA + 070	R/W	Graphics Engine Host bit block transfer data port4	0x0000_0000
2D_GEHBDW5	GE_BA + 074	R/W	Graphics Engine Host bit block transfer data port5	0x0000_0000
2D_GEHBDW6	GE_BA + 078	R/W	Graphics Engine Host bit block transfer data port6	0x0000_0000
2D_GEHBDW7	GE_BA + 07C	R/W	Graphics Engine Host bit block transfer data port7	0x0000_0000

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

Bits	Descriptions	
[31:0]	HostBLT Word Data	**32-bit HostBLT High/Low Word Data** These registers specify the high/low words of HostBLT ports.

6.6.7 GVE VPE Video Data Processing Engine

GVE contains two acceleration engines. The first one is 2D computer graphics processing (2D GE) and the second is for video data processing (VPE). It is specially designed to improve the performance of computer GUI functions, such as BitBLTs and Bresenham Line Draw. Meanwhile, GVE can handle the 2D image rotation, bilinear up/down-scaling, and data format conversion. Also a 3X3 DDA filter is used for better image quality. 2D supports the Command-Queue DMA and VPE supports the MMU memory mapping, both with their MMU tables or DMA link lists on system memories.

6.6.7.1 Overview

Video Data Processing Engine (VPE) contains hardware acceleration engines for the still images and the video movies.

The first function is for the image/video data format conversion and the second function is for the image/video 2D rotation or the coordinate transformation.

Basically, it converts the source planar or packet YUV/YCbCr data to the destination packet YUV/RGB data. VPE reads the planar YUV/YCbCr data from the frame buffer and converts the video pictures into packet YUV/RGB data formats.

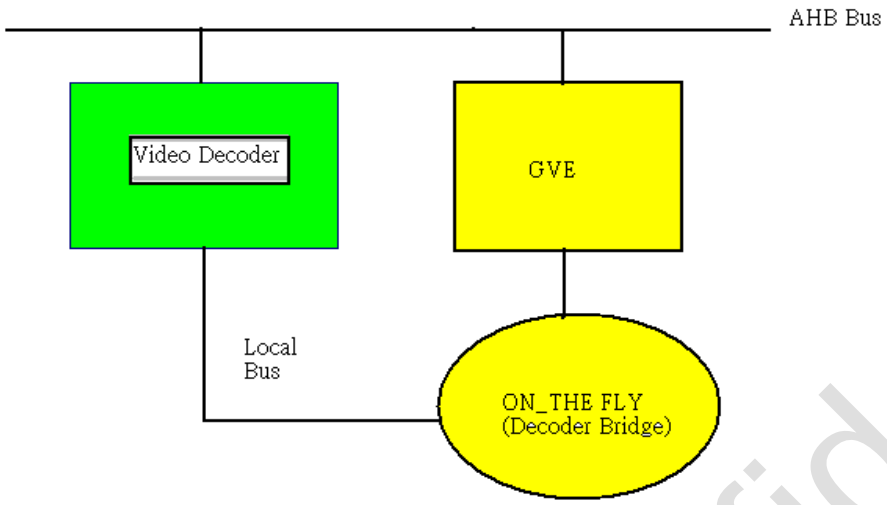
VPE is specially designed to improve the performance of bandwidth hungered functions such as the continuous video or still image rotation, bilinear up/downscaling, and data format conversion.

During the VPE data format conversion, users can specify the bilinear up/downscaling and rotation directions with flip/flop, mirror, left, or right 90/180 degrees at the same time. Both the horizontal and vertical bilinear up/downscaling factors are programmable.

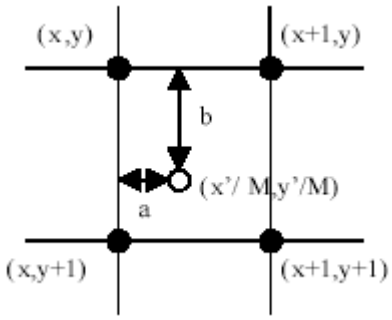
In addition to the really physical address mapping, the standard MMU or virtual address translation mechanism is also implemented in VPE. With the help of VPE MMU mapping, a large resolution picture can be easily processed by using the randomly scattered 4K-byte page sizes.

Besides some specific hosts, the source data can come from frame memory via AHB bus. Basically, VPE can be tightly coupled with the specific hosts such as C&M video decoder and JPEG decoder. These local buses are hardwired and with a simple Request/Grant handshake protocol. In addition to the standard bilinear filter for smoothly up/down-scaling is implemented inside, also a 3X3 average filter with DDA is supported for a better image quality

The following figure shows the ON-THE-FLY local bus, a bridge between a video decoder and VPE. They are hardwired and almost all control signals and data are driven by hosts, except that a VPE Grant signal is sent to hosts for the simple Request/Grant handshake protocol.



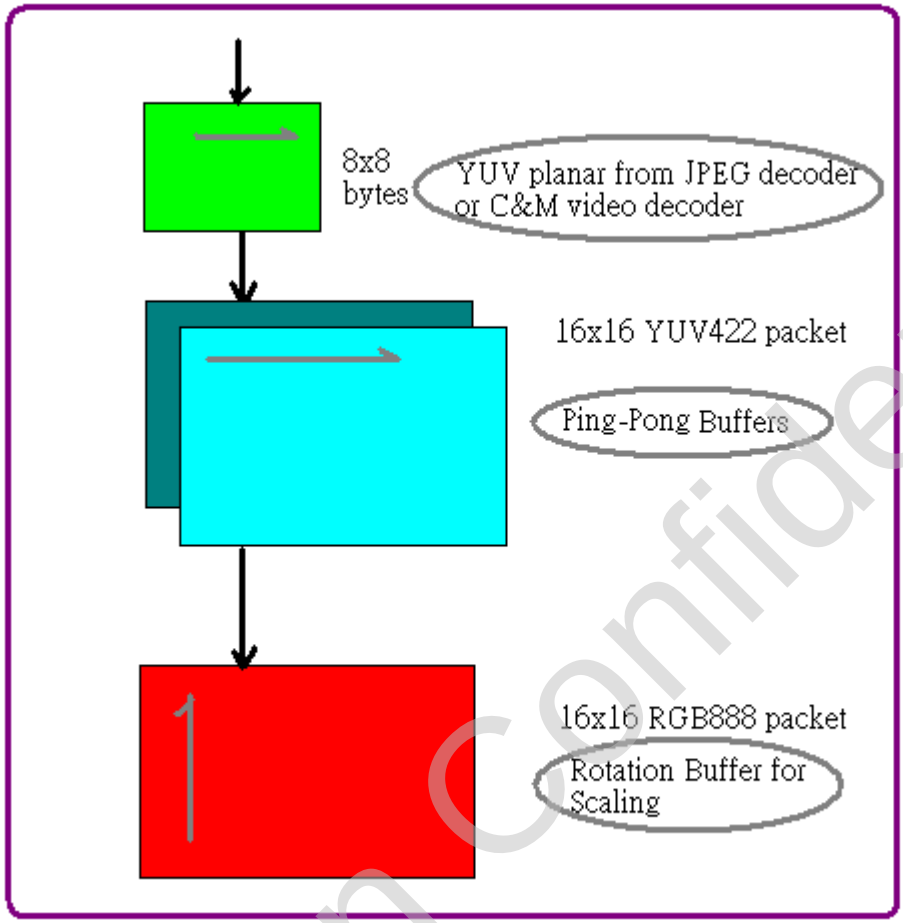
VPE Bilinear Filter Formula:



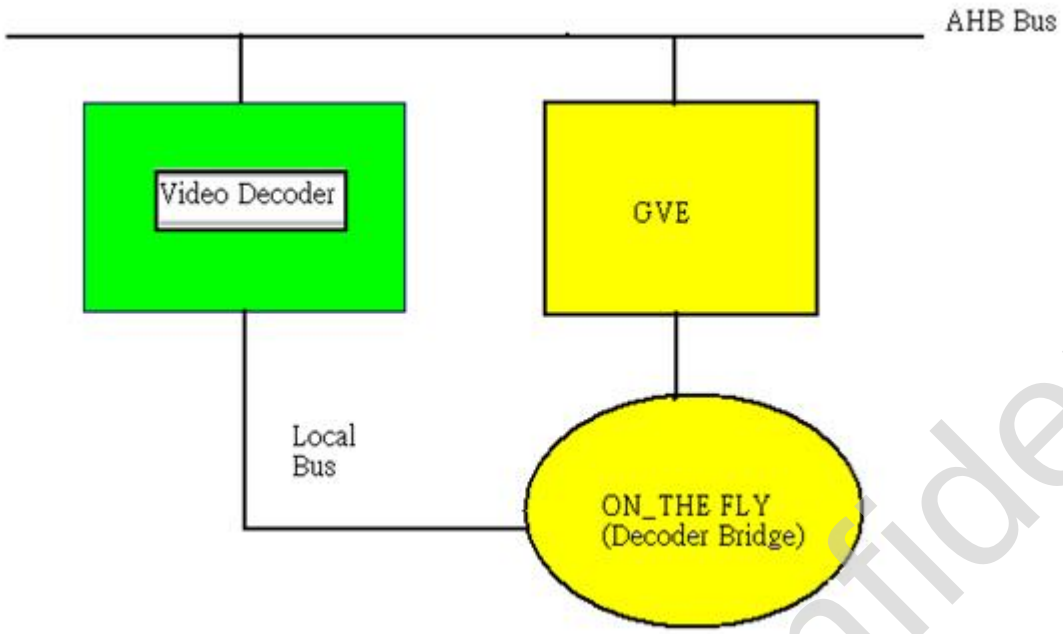
$$F[x',y]=(1-a)\times I[x,y]+a\times I[x+1,y], \qquad x=(\text{int})\frac{x'}{M}, \qquad a=\frac{x'}{M}-x$$

$$O[x',y']=(1-b)\times F[x',y]+b\times F[x',y+1]$$

In order to receive the planar YUV data from C&M video decoder or JPEG engine, VPE needs some 8x8 block-based input buffers to hold the burst data. A set of ping-pong buffers for packet YUV 422 and two RGB buffers are used for rotations. Internal de-blocking filter is implemented to remove the aliasing or artifact between macro blocks.

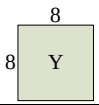
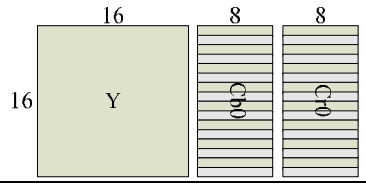
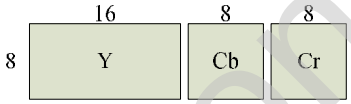
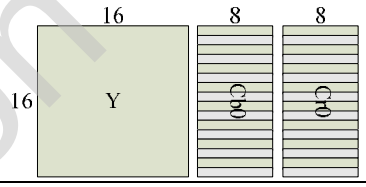
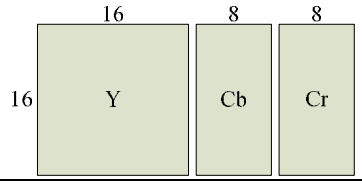
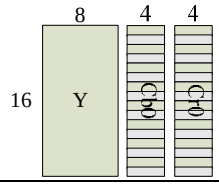


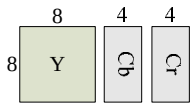
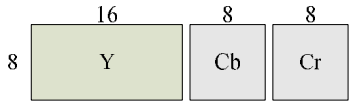
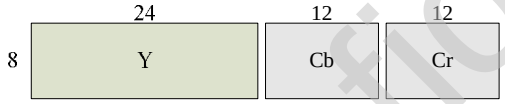
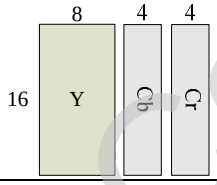
VPE ON_THE_FLY BUFFERS

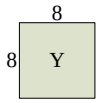


VPE supports all types of block-based data from C&M and JPEG decoders.

Video Decoder block types and block sequences are listed as follows:

Data Format	Y/Cb/Cr MCU H x V	Block Type	Output Pattern Block Sequence
PL400	BLK 1x1	0	
PL420	Y 2x2 Cb 1x1 Cr 1x1	1	
	Y 4x1 Cb 1x1 Cr 1x1	2	
PL422	Y 2x1 Cb 1x1 Cr 1x1	3	
	Y 4x1 Cb 2x1 Cr 2x1	4	
	Y 2x2 Cb 2x1 Cr 2x1	5	
	Y 2x2 Cb 1x2 Cr 1x2	6	
	Y 1x2 Cb 1x1 Cr 1x1	7	

Data Format	Y/Cb/Cr MCU H x V	Block Type	Output Pattern Block Sequence
PL444	Y 1x1 Cb 1x1 Cr 1x1	8	
	Y 2x1 Cb 2x1 Cr 2x1	9	
	Y 3x1 Cb 3x1 Cr 3x1	10	
	Y 1x2 Cb 1x2 Cr 1x2	11	

Data Format	PK422/RGB MCU H x V	Block Type	Output Pattern Block Sequence
PK422	BLK 1x1	12	
RGB555		13	
RGB565		14	
RGB888		15	

6.6.8 Video Playback Mode

VPE reads the source video data, with a format as planar YUV/YCbCr 444/422/420, packet YUV 422 or RGB. It converts the video data to packet YUV 422 or RGB 555/565/888, and stores the results into the destination memory. Users can define the arbitrary up/downscaling factors during data conversion process. Before starting this function, programmers must define the source/destination video sizes and formats. The source and destination video formats are defined in the VPE command control register.

VPE provides a CCIR601 control bit during data format conversion, for source Y/U/V components being a range of CCIR601 or full range (0~255). H/W 1-D Y and 2-D RGB filters for smoothing the up/downscaled image is also provided.

Rotation and Bilinear Filter for Interpolation/Decimation

This function is to support the rotation and the bilinear up/downscaling for any rectangular object in the display memory. In the rotation, it can rotate left or right with 90 or 180 degrees, and it supports the flip/flop, mirror or up-side-down pictures. In the bilinear up/downscaling function, both programmable horizontal and vertical N/M up/downscaling factors are provided for resizing the image. Of course, for downscaling, the scale factor (N/M), the numerator value of N must be equal or less than the denominator value M. For up-scaling, the scaling factor (N/M), the numerator value of N must be equal or larger than the denominator value M. The width/height of source picture are exactly the same as the denominator M's, and they are stored in the VPE horizontal/vertical scaling factor registers.

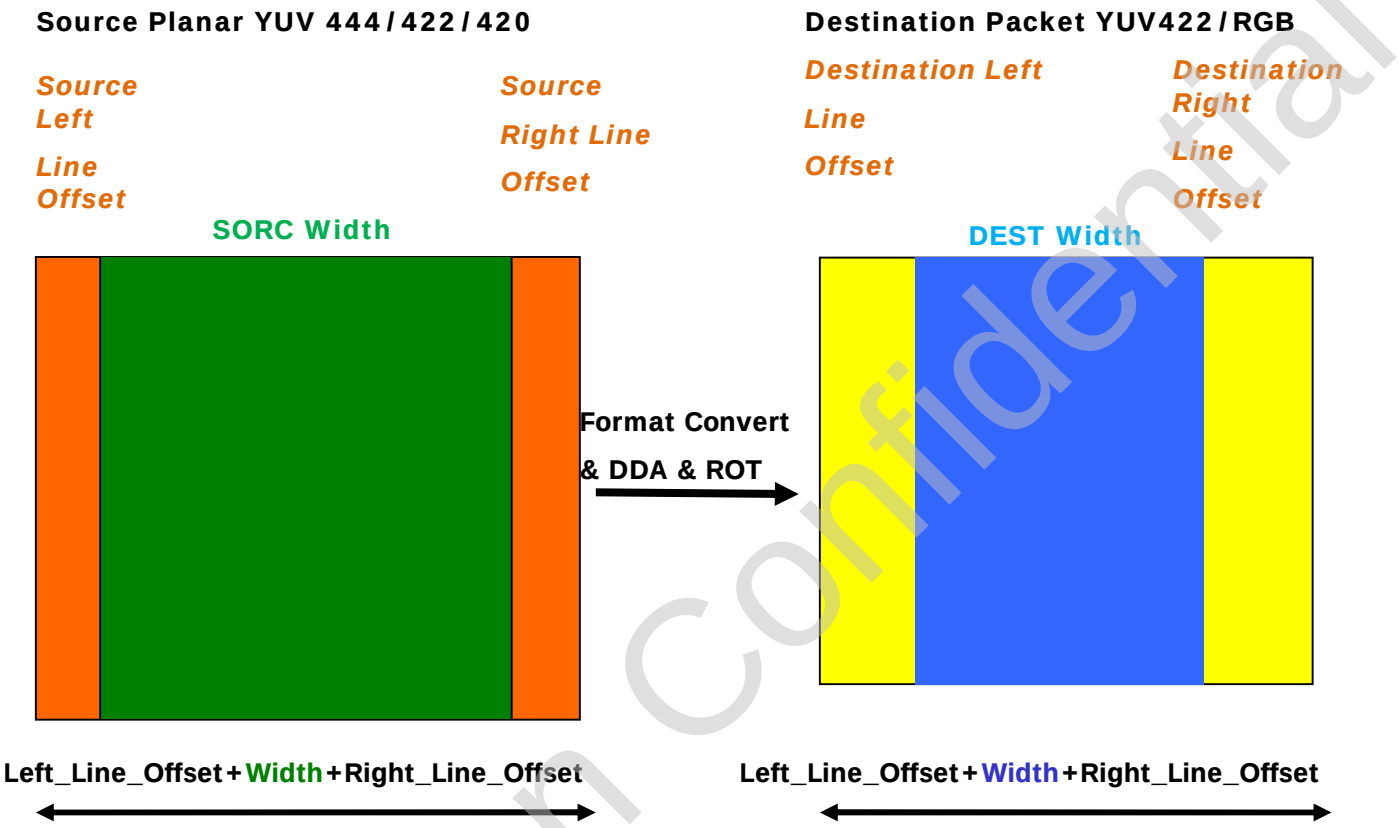
Bilinear Filter for Interpolation/Decimation and Up/Downscaling

Bilinear interpolation/decimation filters for smoothly up/downscaling are implemented inside VPE. 4 points are precisely chosen from the source pictures, and the position of target pixels are calculated with the following bilinear filter formula to render the pixels.

$$F[x',y]=(1-a)\times I[x,y]+a\times I[x+1,y],\qquad x=(\text{int})\frac{x'}{M},\qquad a=\frac{x'}{M}-x$$

$$O[x',y']=(1-b)\times F[x',y]+b\times F[x',y+1]$$

FA95 VPE Picture Format



6.6.9 VPE 3X3 Filter with DDA Down-Sampling

According to the sampling theory, a pre-filter had better be inserted before the down-sampling or decimation process, whereas a post-filter had better be added after the up-sampling or interpolation.

For this reason, in addition to the VPE bilinear decimation/interpolation filters, VPE also implements a dedicated 3X3 filter with DDA down-sampling to improve better image qualities both for the down-scaling and the up-scaling pictures.

Two 13-bits value specifies the numerator and denominator (N/M) of the vertical down-sampling factor in this 3X3 DDA low pass filter. The output image height will be equal to the input image height * (N / M), where N must be equal to or less than M for this 3X3 DDA low pass filter.

Two 13-bits value specifies the numerator and denominator (N/M) of the horizontal down-sampling factor in this 3X3 DDA low pass filter. The output image width will be equal to the input image width * (N / M), where N must be equal to or less than M for this 3X3 DDA low pass filter.

The 3X3 DDA filter is the standard low pass filter, based on 9 pixels with 9 tap coefficients, to generate a mixed output pixel. It can only work for the source image with the packet formats such as YUV422, RGB555, RGB565,

or RGB888. The 9 tap coefficients are 1 central dominant coefficient and the other 8 surrounding tap coefficients from VPE_FCOEFF0 C0~C3 and VPE_FCOEFF1 C4~C7.

C0 C1 C2 pixel_0 pixel_1 pixel_2

C3 CD C4 * pixel_3 pixel_central pixel_4

C5 C6 C7 pixel_5 pixel_6 pixel_7

The default values of the 3X3 DDA filter tap coefficients in H/W are:

CD: the central dominant pixel tap coefficient is 160 (8'hA0)

C0: the surrounding pixel_0 tap coefficient is 8 (8'h08)

C1: the surrounding pixel_1 tap coefficient is 16 (8'h10)

C2: the surrounding pixel_2 tap coefficient is 8 (8'h08)

C3: the surrounding pixel_3 tap coefficient is 16 (8'h10)

C4: the surrounding pixel_4 tap coefficient is 16 (8'h10)

C5: the surrounding pixel_5 tap coefficient is 8 (8'h08)

C6: the surrounding pixel_6 tap coefficient is 16 (8'h10)

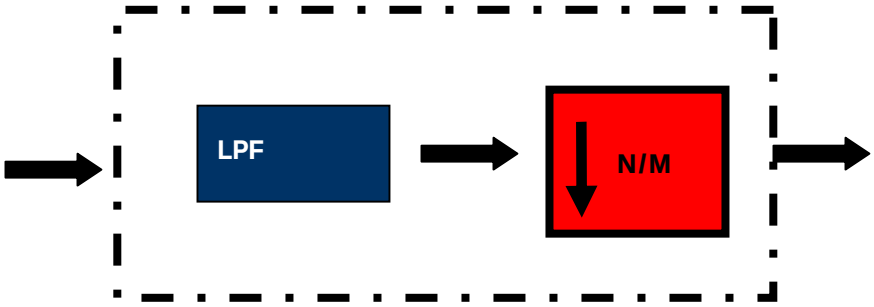
C7: the surrounding pixel_7 tap coefficient is 8 (8'h08)

The total sum of CD+C0~C0 should to be equal to or less than 256.

The setting value of CD ranges from 1 to 256, with the "CD equal to 0" representing for 256.

Mixed output pixel by 3X3 DDA filter = { (pixel_central *CD)+ ∑(Cn*pixel_n)} / 256

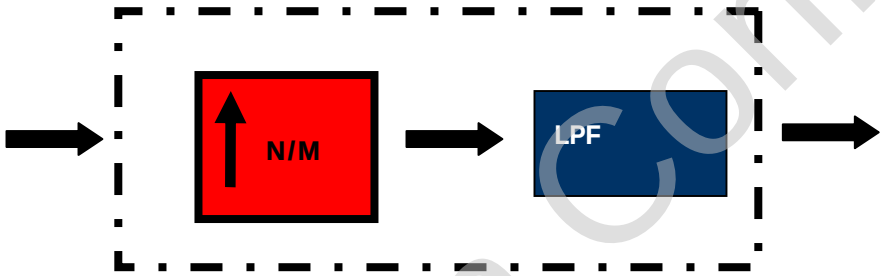
(1) Down-sampling flow:



For example, a downscaling factor N/M = (n1*n2)/(m1*m2) , where N=n1*n2, M=m1*m2, N<= M.

A better image quality can be achieved if a downscaling factor (n1/m1) is processed by the 3X3 DDA filter at first, and then the result is followed by the bilinear decimation filter with a downscaling factor (n2/m2) latter on. Such as for a downscaling factor 1/6 = (1/3) *(1/2) is processed by two steps with 1/3 down-sampled in 3X3 DDA filter and with 1/2 in the bilinear decimation filter laterly.

(2) Up-sampling flow:



For example, an up-scaling factor N/M = (n1*n2)/(m1*m2) , where N=n1*n2, M=m1*m2, N>= M.

A better image quality can be achieved if an up-scaling factor (n1/m1) is processed by the bilinear interpolation filter at first, and then the result is followed by the 3X3 DDA filter with a downscaling factor (n2/m2) latter on. Such as for an up-scaling factor 5/2 = (5/1) *(1/2) is processed by two steps with 5/1 up-sampled in the bilinear interpolation filter and with 1/2 in the following 3X3 DDA filter.

Another example of up-scaling factor N/M = (N*1)/(M*1) , where N>= M.

A better image quality can be achieved if an up-scaling factor (N/M) is processed by the bilinear interpolation filter at first, and then the result is followed by the 3X3 DDA filter with a downscaling factor 1:1 latter on. Such as for an up-scaling factor 5/2 = (5/2) *(1/1) is processed by two steps with 5/2 up-sampled in the bilinear interpolation filter and with 1/1 in the following 3X3 DDA filter.

Following figures are rendered with different cropping windows and left/right offsets during the on-the-fly.

In order to show the original picture's quality, the VPE bilinear filter is turned off in this case.

Source planar YUV420: 176x144

Destination packet YUV422: 192x132

Destination packet YUV422: 208x144

Cropping Window + Left Offset + Right Offset

Destination packet YUV422: 176x144

6.6.10 Video Processing Engine Control Registers Map

VPE_BA = 0xB100_C800

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
VPE_TG	VPE_BA + 000	R/W	Video Process Engine (VPE) Trigger Control Register	0x0000_0000
VPE_PLYA_PK	VPE_BA + 004	R/W	VPE Source Planar Y or Packet YUV Start Address	0x0000_0000
VPE_PLUA	VPE_BA + 008	R/W	VPE Source Planar U Start Address	0x0000_0000
VPE_PLVA	VPE_BA + 00C	R/W	VPE Source Planar V Start Address	0x0000_0000
VPE_INTS	VPE_BA + 010	R/W	VPE Interrupt Status Register	0x0000_0000
VPE_SLORO	VPE_BA + 014	R/W	Source Packet / Planar Y Left Line Offset and Right Line Offset	0x0000_0000
VPE_VYDSF	VPE_BA + 018	R/W	Vertical Divider for DDA Scaling Up/Down	0x0000_0000
VPE_HXDSF	VPE_BA + 01C	R/W	Horizontal Divider for DDA Scaling Up/Down	0x0000_0000
VPE_CMD	VPE_BA + 020	R/W	VPE Command Control Register	0x0000_0000
VPE_DEST_PK	VPE_BA + 024	R/W	Data Format Conversion Packet Destination Start Address	0x0000_0000
VPE_DLORO	VPE_BA + 028	R/W	Destination Packet Left Line Offset and Right Line Offset	0x0000_0000
VPE_FCOEF0	VPE_BA + 02C	R/W	3X3 Filter0, 4 Coefficients Around the Central Pixel	0x0000_0000
VPE_CROPS	VPE_BA + 02C	R/W	ONF Cropping Window Start Point X/Y Coordinate	0x0000_0000
VPE_FCOEF1	VPE_BA + 030	R/W	3X3 Filter1, 4 Coefficients Around the Central Pixel	0x0000_0000
VPE_CROPE	VPE_BA + 030	R/W	ONF Cropping Window End Point X/Y Coordinate	0x0000_0000
VPE_RESET	VPE_BA + 034	R/W	VPE Reset Control Register	0x0000_0000
VPE_MCU	VPE_BA + 038	R/W	VPE Y/X Minimum Coded Unit Number Register	0x0000_0000

6.6.11 VPE MMU Control Registers Map

VPE_BA = 0xB100_C800

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
VMMU_CR	VPE_BA + 080	R/W	VPE MMU Control Register	0x0000_0000
VMMU_TTB	VPE_BA + 084	R/W	VPE MMU Translation Table Base Register	0x0000_0000
VMMU_PFTVA	VPE_BA + 088	R	VPE MMU Page Fault Virtual Address Register	0x0000_0000
VMMU_CMD	VPE_BA + 08C	R/W	VPE MMU Resume and Invalidate Command	0x0000_0000
VMMU_L1PT0	VPE_BA + 090	R/W	VPE MMU Level-One Page Table Entry 0 Descriptor	0x0000_0000
VMMU_L1PT1	VPE_BA + 094	R/W	VPE MMU Level-One Page Table Entry 1 Descriptor	0x0000_0000
VMMU_L1PT2	VPE_BA + 098	R/W	VPE MMU Level-One Page Table Entry 2 Descriptor	0x0000_0000
VMMU_L1PT3	VPE_BA + 09C	R/W	VPE MMU Level-One Page Table Entry 3 Descriptor	0x0000_0000
VMMU_L1PT4	VPE_BA + 0A0	R/W	VPE MMU Level-One Page Table Entry 4 Descriptor	0x0000_0000
VMMU_L1PT5	VPE_BA + 0A4	R/W	VPE MMU Level-One Page Table Entry 5 Descriptor	0x0000_0000
VMMU_L1PT6	VPE_BA + 0A8	R/W	VPE MMU Level-One Page Table Entry 6 Descriptor	0x0000_0000
VMMU_L1PT7	VPE_BA + 0AC	R/W	VPE MMU Level-One Page Table Entry 7 Descriptor	0x0000_0000
VMMU_CVA	VPE_BA + 0B0	R	VPE MMU Current Virtual Address Register	0x0000_0000
VMMU_CVPN	VPE_BA + 0B4	R	VPE MMU Current Virtual Page Number Register	0x0000_0000
VMMU_CPA	VPE_BA + 0B8	R	VPE MMU Current Physical Address Register	0x0000_0000
VMMU_CPPN	VPE_BA + 0BC	R	VPE MMU Current Physical Page Number Register	0x0000_0000

Video Processing Engine Control Registers

Register	Address	R/W	Description	Reset Value
VPE_TG	VPE_BA + 000	R/W	Video Process Engine Trigger Control Reg.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
VPE 3X3 Filter Current Central Pixel TAP Coefficient [7:0]							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							GO

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Central Pixel TAP Coefficient	8-bit Central Pixel Coefficient For VPE 3x3 filter, this tap coefficient specifies the weighting factor to be multiplied to the current central pixel. The default value 0 stands for 256.
[15:1]	Reserved	Reserved
[0]	GO	Trigger Video Process Engine Operation (Software trigger mode) 1 = Start VPE operation, automatically cleared when VPE is completed. 0 = No VPE operation or the VPE operation is finished.

Video Processing Engine Source Planar Y or Packet YUV 422 Start Address Register

Register	Address	R/W	Description	Reset Value
VPE_PLYA_PK	VPE_BA + 004	R/W	Video Process Engine Source Planar Y or Packet YUV 422 Start Address	0x0000_0000

31	30	29	28	27	26	25	24
Planar Y/Packet YUV 422 Address [31:24]							
23	22	21	20	19	18	17	16
Planar Y/Packet YUV 422 Address [23:16]							
15	14	13	12	11	10	9	8
Planar Y/Packet YUV 422 Address [15:8]							
7	6	5	4	3	2	1	0
Planar Y/Packet YUV 422 Address [7:0]							

Bits	Descriptions	
[31:0]	Planar Y /Packet YUV 422 Address	32-bit Source Planar Y or Packet YUV 422 Start Address (byte unit) This 32-bit byte address specifies the source planar Y or packet YUV 422 starting address of an object or a picture in the display memory.

Video Processing Engine U Space Planar Type Start Address Register

Register	Address	R/W	Description	Reset Value
VPE_PLUA	VPE_BA + 008	R/W	Video Process Engine U Space Planar Type Start Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Planar U Address	**32-bit Planar U Space Start Address (byte unit)** This 32-bit byte address specifies the U space planar type starting address of an object or a picture in the display memory.

Video Processing Engine V Space Planar Type Start Address Register

Register	Address	R/W	Description	Reset Value
VPE_PLVA	VPE_BA + 00C	R/W	Video Process Engine V Space Planar Type Start Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Planar V Address	**32-bit Planar V Space Start Address (byte unit)** This 32-bit byte address specifies the V space planar type starting address of an object or a picture in the display memory.

Video Processing Engine Interrupt Status Register

Register	Address	R/W	Description	Reset Value
VPE_INTS	VPE_BA + 010	R/W	Video Process Engine Interrupt Status Reg.	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		TA_INTS	DE_INTS	MB_INTS	PG_MISS	PF_INTS	VP_INTS

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5]	TA_INTS	**VPE DMA Target Abort or Data Abort Interrupt Status** 0 = No DMA target abort interrupt occur. No AHB Bus Error or not any AHB Abort response occurs. 1 = DMA target abort interrupt occurs, host writes one to clear TA_INTS.
[4]	DE_INTS	**Decoder Block Sequence Error Interrupt Status** **0 = No video decoder block sequence out of order interrupt occur.** **1 = Video decoder block sequence error interrupt occurs, host writes one to clear DE_INTS.**
[3]	MB_INTS	**Decoder MCU/Macro Block Detection Interrupt Status** **0 = No video decoder MCU/Macro Block detection occur.** **1 = Video decoder MCU/Macro Block detection interrupt occurs, host writes one to clear this MB_INTS.**
[2]	PG_MISS	**VPE MMU Page Miss Interrupt Status** 0 = No VPE MMU Page Miss interrupt. (The unmatched level-one descriptors) 1 = VPE MMU mets a Page Miss interrupt. Write one to clear this interrupt.

[1]	PF_INTS	**VPE MMU Page Fault Interrupt Status** 0 = No VPE MMU page fault interrupt occurs. 1 = VPE MMU page fault interrupt occurs.
[0]	VP_INTS	**VPE Completion Interrupt Status** 0 = No interrupt occurs. 1 = VPE is completed and its interrupt occur, host writes one to clear this VP_INTS.

Video Processing Engine Source Packet/Planar Y Left/Right Line Offset Register (pixel unit)

Register	Address	R/W	Description	Reset Value
VPE_SLORO	VPE_BA + 014	R/W	Source Packet/Planar Y Left Line and Right Line Offset	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			Source Packet/Planar Y Left Line Offset [12:8]				
23	22	21	20	19	18	17	16
Source Packet/Planar Y Left Line Offset [7:0]							
15	14	13	12	11	10	9	8
Reserved			Source Packet/Planar Y Right Line Offset [12:8]				
7	6	5	4	3	2	1	0
Source Packet/Planar Y Right Line Offset [7:0]							

Bits	Descriptions	
[28:16]	Source Packet/Planar Y Left Line Offset	13-bit Source Packet/Planar Y Left Line Offset This register specifies the Source Packet/Planar Y Left Line Offset (by pixel).
[12:0]	Source Packet/Planar Y Right Line Offset	13-bit Source Planar Y Right Line Offset This register specifies the Source Packet/Planar Y Right Line Offset (by pixel).

Video Processing Engine Vertical DDA N/M Divider (Divider for DDA Scaling Up/Down)

Register	Address	R/W	Description	Reset Value
VPE_VYDSF	VPE_BA + 018	R/W	Vertical Divider for DDA Scaling Up/Down	0x0000_0000

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

Bits	Descriptions	
[28:16]	VSF_N	**13-bit Vertical N Scaling Factor** A 13-bits value specifies the numerator part (N) of the vertical scaling factor in playback mode. The output image height will be equal to the input image height * (N / M). The maximum is 4096.
[12:0]	VSF_M	**13-bit Vertical M Scaling Factor** A 13-bits value specifies the denominator part (M) of the vertical scaling factor in playback mode. The output image height will be equal to the input image height * (N / M). The maximum is 4096.

Video Processing Engine Horizontal DDA N/M Divider (Divider for DDA Scaling Up/Down)

Register	Address	R/W	Description	Reset Value
VPE_HXDSF	VPE_BA + 01C	R/W	Horizontal Divider for DDA Scaling Up/Down	0x0000_0000

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

Bits	Descriptions	
[28:16]	HSF_N	**13-bit Horizontal N Scaling Factor** A 13-bits value specifies the numerator part (N) of the horizontal scaling factor in playback mode. The output image width will be equal to the input image width * (N / M). The maximum is 4096.
[12:0]	HSF_M	**13-bit Horizontal M Scaling Factor** A 13-bits value specifies the denominator part (M) of the horizontal scaling factor in playback mode. The output image width will be equal to the input image width * (N / M). The maximum is 4096.

Video Processing Engine Command Control Register

Register	Address	R/W	Description	Reset Value
VPE_CMD	VPE_BA + 020	R/W	Video Process Engine Command Control Reg.	0x0000_0000

31	30	29	28	27	26	25	24
CCIR601	SORC (Read Only)			LEVEL	TRACE	DEST	
23	22	21	20	19	18	17	16
SORC BLOCK SEQUENCE				OPERATE COMMAND			
15	14	13	12	11	10	9	8
uTLB-SET		Y_Round	X_Round	SINGLE	CROP	BYPASS	TAP
7	6	5	4	3	2	1	0
BILINEAR	BURST	HOST SELECT		MB_EN	PMS_EN	PFT_EN	INT_EN

Bits	Descriptions	
[31]	CCIR601	**Source YUV Level Range** **0 = YUV level is full range. (YUV are 0~255)** **1 = YUV level is CCIR601 range.** **(Y is 16~235, and U/V are 16~240)**
[30:28]	SORC	**Source YUV/RGB Formats (Read Only)** 000 = Planar YUV 420 Format 001 = Planar YUV 422 Format 010 = Planar YUV 444 Format 011 = Packet YUV 422 Format 100 = Planar YUV 400 Format 101 = Packet RGB 555 Format 110 = Packet RGB 565 Format 111 = Packet RGB 888 Format
[27]	LEVEL	This LEVEL bit is used for DEST packet YUV level adjustment to the CCIR601 range when SORC YUV data are full range (0~255). Users can control the DEST packet YUV level to be different from the SORC YUV level by this LEVEL bit. 0 = No DEST packet YUV level adjustment.

		1 = Turn on DEST packet YUV level adjustment. (Full Range à CCIR601)
[26]	**TRACE**	**Trace the Internal Hardware States (For debugging purpose)** 1 = Trace the Internal VPE Finite State Machines. 0 = No Trace.
[25:24]	**DEST**	**Destination YUV/RGB Formats** 00 = Packet YUV 422 Format 01 = Packet RGB555 Format 10 = Packet RGB565 Format 11 = Packet RGB888 Format
[23:20]	**SORC** **BLOCK SEQUENCE**	**The Different Types of Block Sequences of Source Data** Block sequences are defined as follows: 0000 = Video decoder block type is Y Luminance only. (PL YUV 400 Format) 0001 = Video decoder block type is PL YCbCr420. (Planar YUV 420 Format) 0010 = Video decoder block type is PL YCbCr420. (Planar YUV 420 Format) 0011 = Video decoder block type is PL YCbCr422. (Planar YUV 422 Format) 0100 = Video decoder block type is PL YCbCr422. (Planar YUV 422 Format) 0101 = Video decoder block type is PL YCbCr422. (Planar YUV 422 Format) 0110 = Video decoder block type is PL YCbCr422. (Planar YUV 422 Format) 0111 = Video decoder block type is PL YCbCr422. (Planar YUV 422 Format) 1000 = Video decoder block type is PL YCbCr422. (Planar YUV 444 Format) 1001 = Video decoder block type is PL YCbCr422. (Planar YUV 444 Format) 1010 = Video decoder block type is PL YCbCr422. (Planar YUV 444 Format) 1011 = Video decoder block type is PL YCbCr422. (Planar YUV 444 Format) 1100 = Video decoder block type is PK YCbCr422. (Packet YUV 422 Format) 1101 = Video decoder block type is PK RGB 555. (Packet RGB 555 Format) 1110 = Video decoder block type is PK RGB 565. (Packet RGB 565 Format) 1111 = Video decoder block type is PK RGB 888. (Packet RGB 888 Format)
[19:16]	**OPERATE COMMAND**	**Operate Command of Rotate Direction or 3X3 Average Filter** **0000 = Normal. (No Rotation)** **0001 = Rotate right by 90 degrees.**

		0010 = Rotate left by 90 degrees. 0011 = Rotate by 180 degrees. 0100 = Up-Side-Down (Flip Picture) 0101 = Mirror. (Flop Picture) 1XXX = DDA Down-Scaling Operation with 3X3 Average Filter (Scale Down) Others = Reserved Note: The 3X3 average filter is turned on/off by VPE_CMD bit-9 BYPASS. The 3X3 average filter can be used under two situations: OPERATE COMMAND=0XXX This 3X3 filter is cascaded in series with the bilinear filter by macroblock. OPERATE COMMAND=1XXX This 3X3 filter is used stand-alone for DDA down-scaling by frame based.
[15:14]	uTLB_SET	Micro TLB (uTLB) dump to VPE MMU CR's 80~FF. (For debugging purpose) uTLB Set Select Numbers for VPE MMU Micro TLB Dump 00 = Dumping the VPE MMU Micro TLB Set 0 01 = Dumping the VPE MMU Micro TLB Set 1 10 = Dumping the VPE MMU Micro TLB Set 2 11 = Dumping the VPE MMU Micro TLB Set 3
[13]	Y_Round	Y-axis Round Control. (For debugging purpose) Y_Round can force the negative Y-axis value to become the positive zero. 0 = No rounding the Y-axis value. 1 = Rounding the Y-axis value
[12]	X_Round	X-axis Round Control. (For debugging purpose) X_Round can force the negative X-axis value to become the positive zero. 0 = No rounding the X-axis value. 1 = Rounding the X-axis value
[11]	SINGLE	VPE Write Buffer Mode 0 = Dual Write Buffer Mode 1 = Single Write Buffer Mode
[10]	CROP	ONF Cropping Window 0 = Disable the on-the-fly cropping window

		1 = Enable the on-the-fly cropping window
[9]	BYPASS	3X3 Filter On/Off (When Operate Command is DDA Down-Scaling) 0 = 3X3 Filter is Turned On during DDA Down-Scaling Operation. 1 = 3X3 Filter is Turned Off (Bypassed) during DDA Down-Scaling.
[8]	TAP	3X3 Average Filter Tap Coefficients 0 = Software Mode. Configure the 3X3 Filter Tap Coefficients by S/W. 1 = Hardware Mode. Turn on the 3X3 Filter Tap Coefficients in H/W.
[7]	BILINEAR	VPE Bilinear Up/Down-Scaling Filter Function On/Off 0 = Turn off bilinear filter. 1 = Turn on bilinear filter.
[6]	BUSRT	AHB Write Burst Length for Bilinear Filter to Fetch More Pixels 0 = AHB Write Short Burst Length (Single One Word) 1 = AHB Write Long Burst Length
[5:4]	HOST SELECT	Host Select (C&M Video Decoder, JPEG Decoder) 00~11 represent different hosts 00 = AHB Master(Source is accessed by Macroblocks from Frame Buffers) 01 = AHB Master(Source is accessed by Lines from Frame Buffers) 10 = JPEG Decoder (On-The-Fly) 11 = C&M Video Decoder (On-The-Fly) Note: HOST = 01, only 1-D filter takes effect, no 3X3 or bilinear filter active
[3]	MB_EN	0 = Disable the Macroblock/MCU Detection for Video Decoder. 1 = Enable the Macroblock/MCU Detection for Video Decoder.
[2]	PMS_EN	Page Miss Interrupt Enable for VPE MMU 0 = Disable the VPE MMU Page Miss Interrupt. 1 = Enable the VPE MMU Page Miss Interrupt.
[1]	PFT_EN	Page Fault Interrupt Enable for VPE MMU 0 = Disable the VPE MMU Page Fault Interrupt. 1 = Enable the VPE MMU Page Fault Interrupt.
[0]	INT_EN	Interrupt Enable for VPE operations. 0 = Disable the VPE Operation Interrupt. 1 = Enable the VPE Operation Interrupt.

Video Processing Engine Data Format Conversion Packet Format Destination Start Address

Register	Address	R/W	Description	Reset Value
VPE_DEST_PK	VPE_BA + 024	R/W	Data Format Packet Destination Start Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Data Format Packet Destination	32-bit Data Format Packet Destination Starting Address (byte unit) This 32-bit byte address specifies the packet type starting address for data format conversion of an object or a picture in the display destination memory.

VPE Destination Packet Format Data Left/Right Line Offset Register (by pixel unit)

Register	Address	R/W	Description	Reset Value
VPE_DLORO	VPE_BA + 028	R/W	Destination Packet Data Left Line Offset and Right Line Offset	0x0000_0000

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

Bits	Descriptions	
[28:16]	Destination Packet Left Line Offset	13-bit Destination Packet Format Data Left Line Offset This register specifies the Destination Packet Data Left Line Offset (by pixel).
[12:0]	Destination Packet Right Line Offset	13-bit Destination Packet Format Data Right Line Offset This register specifies the Destination Packet Data Right Line Offset (by pixel).

Video Processing Engine 3X3 Filter0 Coefficient / ONF Cropping Window Start Point X / Y Register

Register	Address	R / W	Description	Reset Value
VPE_FCOEF0	VPE_BA + 02C	R/W	3X3 Filter0 4 Coefficients Around Central Pixel	0x0000_0000
VPE_CROPS	VPE_BA + 02C	R/W	ONF Cropping Window Start Point X/Y Coordinate	0x0000_0000

(A)

31	30	29	28	27	26	25	24
Coefficient0[7:0]							
23	22	21	20	19	18	17	16
Coefficient1[7:0]							
15	14	13	12	11	10	9	8
Coefficient2[7:0]							
7	6	5	4	3	2	1	0
Coefficient3[7:0]							

Bits	Descriptions	
[31:24]	Coefficient0	8-bit Unsigned Coefficient0 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-0 around the central one.
[23:16]	Coefficient1	8-bit Unsigned Coefficient1 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-1 around the central one.
[15:8]	Coefficient2	8-bit Unsigned Coefficient2 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-2 around the central one.
[7:0]	Coefficient3	8-bit Unsigned Coefficient3 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-3 around the central one.

(B)

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

Bits	Descriptions	
[28:16]	OTF Cropping Window Start Y	13-bit Source OTF Cropping Window Start Point Y This register specifies the Source Picture OTF Cropping Window Start Point Y Coordinate (by pixel).
[12:0]	OTF Cropping Window Start X	13-bit Source OTF Cropping Window Start Point X This register specifies the Source Picture OTF Cropping Window Start Point X Coordinate (by pixel).

Video Processing Engine 3X3 Filter1 Coefficient/OTF Cropping Window End Point X/Y Register

Register	Address	R/W	Description	Reset Value
VPE_FCOEF1	VPE_BA + 030	R/W	3X3 Filter1 4 Coefficients Around Central Pixel	0x0000_0000
VPE_CROPE	VPE_BA + 030	R/W	OTF Cropping Window End Point X/Y Coordinate	0x0000_0000

(A)

31	30	29	28	27	26	25	24
Coefficient4[7:0]							
23	22	21	20	19	18	17	16
Coefficient5[7:0]							
15	14	13	12	11	10	9	8
Coefficient6[7:0]							
7	6	5	4	3	2	1	0
Coefficient7[7:0]							

Bits	Descriptions	
[31:24]	Coefficient4	8-bit Unsigned Coefficient4 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-4 around the central one.
[23:16]	Coefficient5	8-bit Unsigned Coefficient5 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-5 around the central one.
[15:8]	Coefficient6	8-bit Unsigned Coefficient6 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-6 around the central one.
[7:0]	Coefficient7	8-bit Unsigned Coefficient7 Around Central Pixel Filter Coefficient around the central pixel. It's the weighting factor to be multiplied to the pixel-7 around the central one.

(B)

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

Bits	Descriptions	
[28:16]	OTF Cropping Window End Y	13-bit Source OTF Cropping Window End Point Y This register specifies the Source Picture OTF Cropping Window End Point Y Coordinate (by pixel).
[12:0]	OTF Cropping Window End X	13-bit Source OTF Cropping Window End Point X This register specifies the Source Picture OTF Cropping Window End Point X Coordinate (by pixel).

Video Processing Engine Reset Control Register

Register	Address	R/W	Description	Reset Value
VPE_RESET	VPE_BA + 034	R/W	Video Process Engine 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
Reserved							
7	6	5	4	3	2	1	0
Reserved						RST_FIFO	RESET

Bits	Descriptions	
[1]	RST_FIFO	RESET VPE FIFO Control 0 = Not reset VPE FIFO Control 1 = Reset VPE FIFO Control
[0]	RESET	RESET VPE Operation 0 = Not reset VPE operation 1 = Reset VPE operation

Video Processing Engine X/Y Minimum Coded Unit (MCU) Number Register (by pixel unit)

Register	Address	R/W	Description	Reset Value
VPE_MCU	VPE_BA + 038	R/W	Video Process Engine MCU Dimension (by pixel)	0x0000_0000

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

Bits	Descriptions	
[28:16]	Y-Dimension MCU	13-bit Y-Dimension MCU Number VPE_CMD MB_EN bit enables the Macro Block or MCU Detection. This MCU number specifies the height of rectangle in the X/Y addressing. It's for decoder MB_INTS or MCU interrupt detection.
[12:0]	X-Dimension MCU	13-bit X-Dimension MCU Number VPE_CMD MB_EN bit enables the Macro Block or MCU Detection. This MCU number specifies the width of rectangle in the X/Y addressing. It's for decoder MB_INTS or MCU interrupt detection.

6.6.12 Video Processing Engine Memory Management Unit (VPE MMU)

VPE MMU converts the address mapping from virtual addresses to physical addresses during VPE data accesses or processing. Basically, a virtual memory up to 4G bytes can be translated and mapped onto the limited physical memory dependent on different specific applications. VPE MMU has the built-in page table cache and Translation Lookaside Buffer (TLB) to provide the very fast and much more efficient memory accesses.

The memory space of a virtual memory system, logically up to 4G bytes, can be translated onto the real memory space by VPE MMU with the help of TLB entries and page tables.

When abnormal address translation exceptions occur, VPE MMU Exception Registers are loaded with some relevant information about the address that caused the exception. Users can debug the root cause at the time upon the occurrence of VPE MMU Page Fault Interrupt.

The hardware translation process is initiated when the TLB does not contain a valid translation for the requested virtual addresses. A single set of two-level page tables stored in the main memory is used to control the address translation. Level-one page table is directed by the Translation Table Base (TTB) register, as ARM CP15 register C2, pointing to the base address of a table in physical memory that contains section or page descriptors. The 14 lower-order bits [13:0] of the TTB register are not used, and the table must reside on a 16KB boundary. Level-one translation table has up to 4096 entries, 32 bits per entry, each describing 1MB of virtual memory. This kind of memory mapping can enable up to 4G bytes of virtual memory to be addressed.

Each entry of level-one page table represents a coarse page table descriptor that provides the base address of a page table containing level-two descriptors for small page accesses. Coarse page tables have 256 entries, splitting the 1MB that the page table descriptors into 4KB blocks.

VPE MMU contains 8-entry level-one page table to record the currently used 4MB among all 4GB.

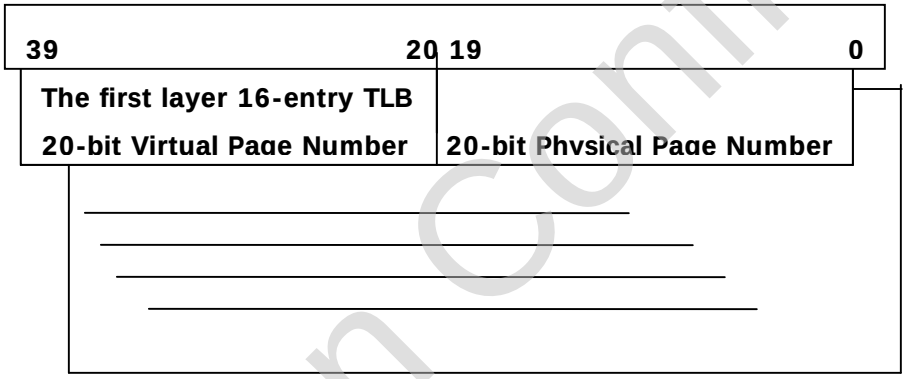
VPE MMU also uses the on-chip TLB to accelerate the mapping process. A 2-layer TLB structures are implemented with 4x16-entry per layer-1-TLB and 4x256-entry per layer-2-TLB in order to increase the hit rate. If both layer-1-TLB and layer-2-TLB are judged to be missed, a physical address is generated by referencing the level-one and level-two page tables sequentially. If the level-one page table entries are judged to be missed, users can detect out a Page Fault Exception.

6.6.13 VPE TLB Structure and VPE MMU Operation Flow

VPE TLB

VPE TLB adopts the 2-layer TLB structures to improve the hit rates of virtual address translation. A 2-layer TLB structures are implemented with 4 sets of 16-entry per layer-1-TLB and 4 sets of 256-entry per layer-2-TLB. The first layer TLB is by using 16-entry flip/flop based cache and the second layer is by using 256-entry SRAM for every TLB set.

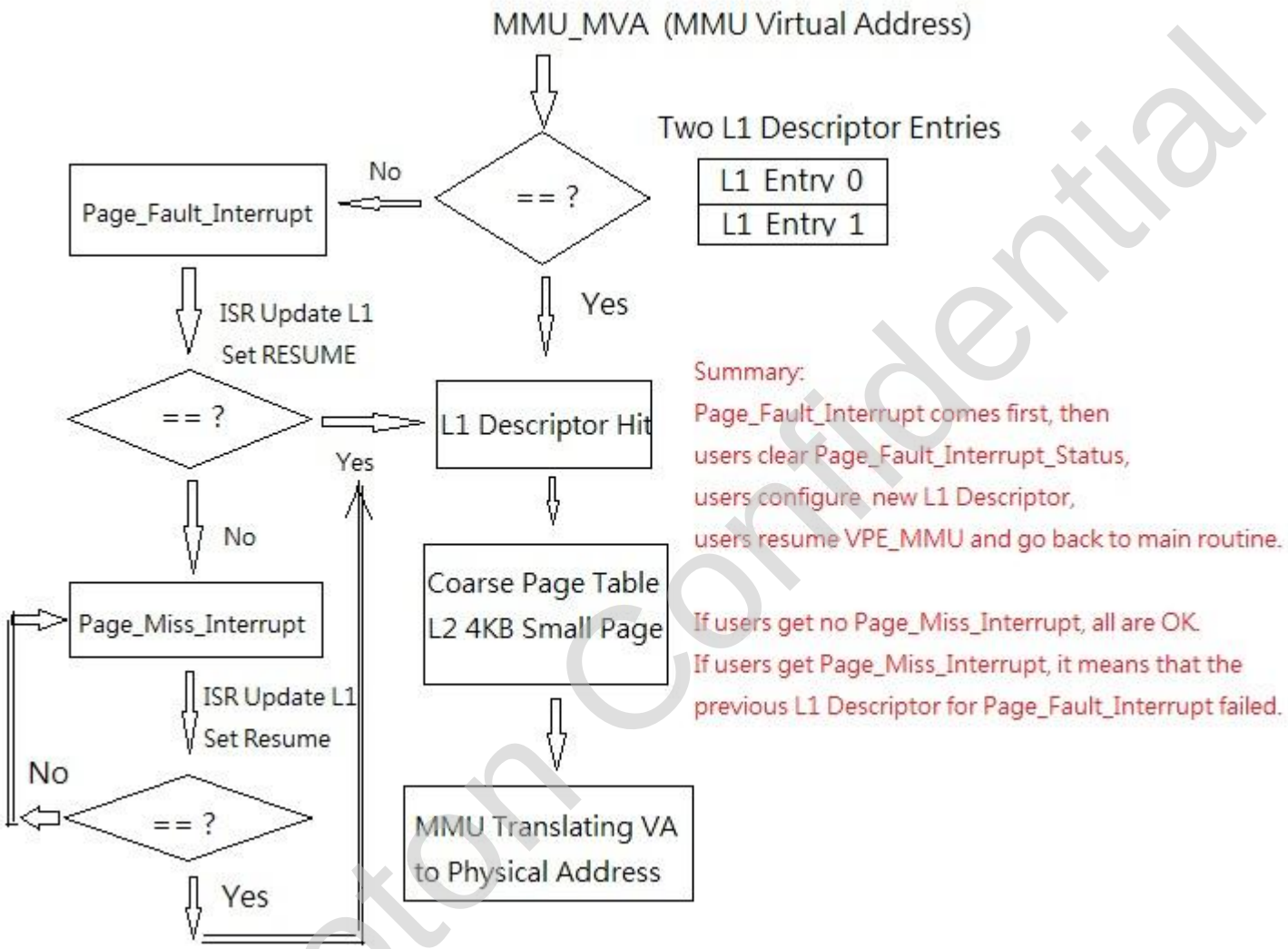
The following block diagrams show the first layer 16-entry TLB and the second layer 256-entry TLB per set.



8-bit VPN	12-bit VPN	20-bit Physical Page Number

The second layer 256-entry TLB SRAM

VPE MMU Operation Flow



VPE MMU Address Translation

Basically, MMU translates virtual addresses into physical addresses for the external memory access, and also performs access permission checking. However, W55FA95 VPE MMU only translates the virtual address mapping without the access permission checking.

A mechanism of MMU table-walking hardware is used to add entries to the TLB. The translation information that comprises the address translation data that resides in a translation table located in physical memory. VPE MMU provides the logic for automatically traversing this translation table and loading entries into the internal on-chip TLB, simultaneously to the first layer and the second layer due to 2-layer TLB structures.

The number of stages in the hardware table walking and permission checking process is one or two depending on whether the address is marked as a section-mapped access or a page-mapped access.

Normally, there are three sizes of page-mapped accesses and one size of section-mapped access.

Page-mapped accesses are for:

- large pages (64KB per page)
- small pages (4KB per page)
- tiny pages. (1KB per page)

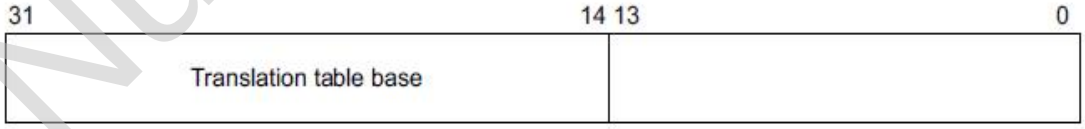
Section-mapped access is dedicated for 1MB memory space.

But in W55FA95 VPE MMU, only the page-mapped accesses with 4KB page size are supported.

The translation process always begins in the same way, with a level one fetch. A section-mapped access requires only a level one fetch, but a page-mapped access requires an additional level two fetch.

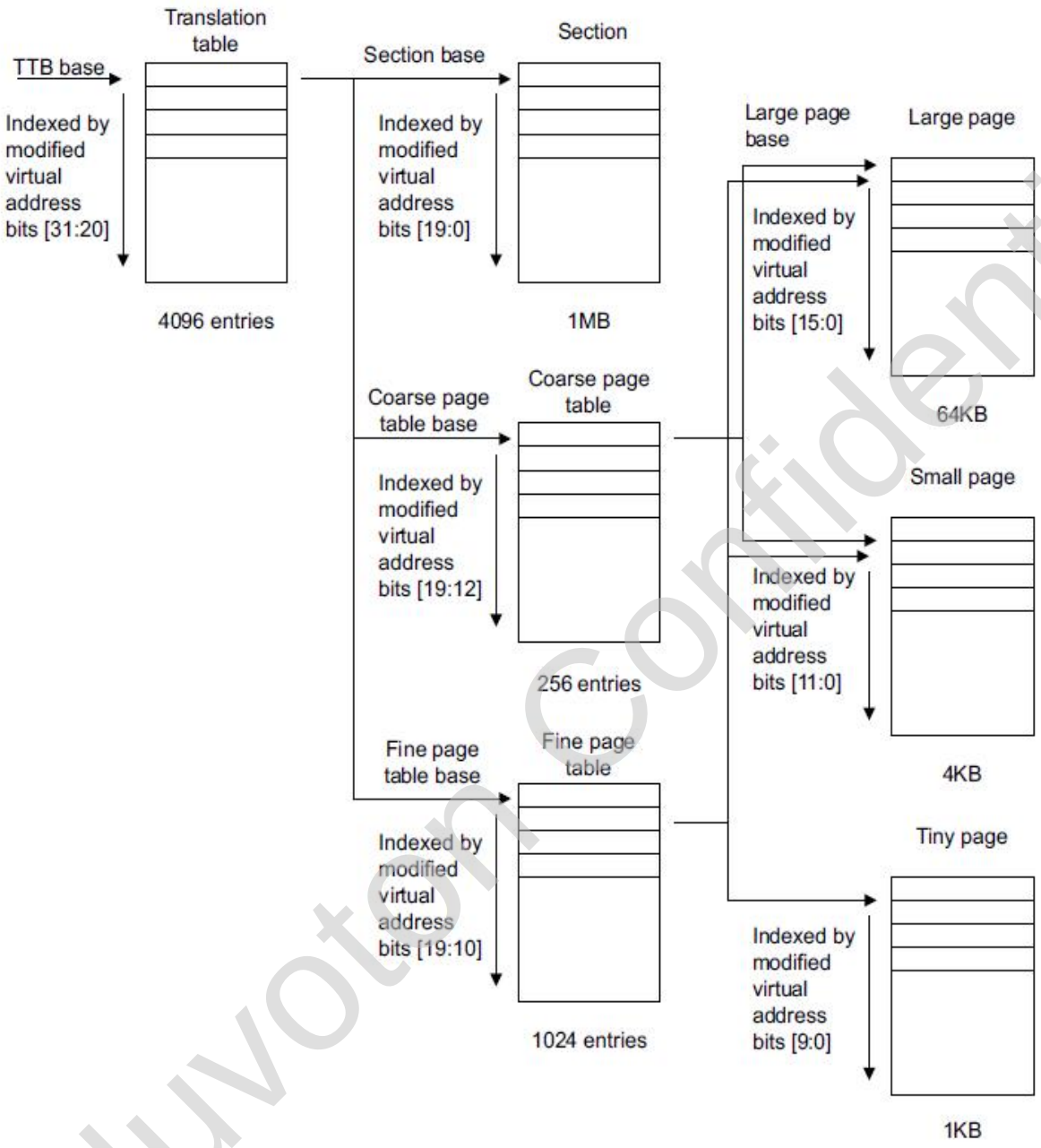
The hardware translation process is initiated when the TLB does not contain a translation for the requested virtual address. The Translation Table Base Register (TTB) pointing to the base address of a table in physical memory that contains section or page descriptors, or both.

TTB is the physical base address of the descriptors for the level-one page table or 1MB section. The 14 low-order bits [13:0] of the TTBR are unpredictable on a read, and the table must reside on a 16KB boundary.



Different virtual address mapping are required dependent on the different engine architectures.

A standard virtual address mapping mechanism of translation page tables are shown as below:



6.6.14 VPE MMU Control Registers Map

VPE_BA = 0xB100_C800

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
VMMU_CR	VPE_BA + 080	R/W	VPE MMU Control Register	0x0000_0000
VMMU_TTB	VPE_BA + 084	R/W	VPE MMU Translation Table Base Register	0x0000_0000
VMMU_PFTVA	VPE_BA + 088	R/W	VPE MMU Page Fault Virtual Address Register	0x0000_0000
VMMU_CMD	VPE_BA + 08C	R/W	VPE MMU Resume and Invalidate Command	0x0000_0000
VMMU_L1PT0	VPE_BA + 090	R/W	VPE MMU Level-One Page Table Entry 0 Descriptor	0x0000_0000
VMMU_L1PT1	VPE_BA + 094	R/W	VPE MMU Level-One Page Table Entry 1 Descriptor	0x0000_0000
VMMU_L1PT2	VPE_BA + 098	R/W	VPE MMU Level-One Page Table Entry 2 Descriptor	0x0000_0000
VMMU_L1PT3	VPE_BA + 09C	R/W	VPE MMU Level-One Page Table Entry 3 Descriptor	0x0000_0000
VMMU_L1PT4	VPE_BA + 0A0	R/W	VPE MMU Level-One Page Table Entry 4 Descriptor	0x0000_0000
VMMU_L1PT5	VPE_BA + 0A4	R/W	VPE MMU Level-One Page Table Entry 5 Descriptor	0x0000_0000
VMMU_L1PT6	VPE_BA + 0A8	R/W	VPE MMU Level-One Page Table Entry 6 Descriptor	0x0000_0000
VMMU_L1PT7	VPE_BA + 0AC	R/W	VPE MMU Level-One Page Table Entry 7 Descriptor	0x0000_0000
VMMU_CVA	VPE_BA + 0B0	R	VPE MMU Current Virtual Address Register	0x0000_0000
VMMU_CVPN	VPE_BA + 0B4	R	VPE MMU Current Virtual Page Number Register	0x0000_0000
VMMU_CPA	VPE_BA + 0B8	R	VPE MMU Current Physical Address Register	0x0000_0000
VMMU_CPPN	VPE_BA + 0BC	R	VPE MMU Current Physical Page Number Register	0x0000_0000

VPE MMU Control Registers

Register	Address	R/W	Description	Reset Value
VMMU_CR	VPE_BA + 080	R/W	VPE MMU 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			MAIN_TLB	Reserved		MAIN_EN	MMU_EN

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	MAIN_TLB	VPE MMU Main TLB Service Channels 1 = VPE MMU Main TLB service for the source and destination DMA channels. 0 = VPE MMU Main TLB service for the destination DMA channel only.
[3:2]	Reserved	Reserved
[1]	MAIN_EN	Turn On/Off VPE MMU Main TLB (SRAM Buffers) 1 = Turn on VPE MMU Main TLB. 0 = Turn off VPE MMU Main TLB.
[0]	MMU_EN	Enable or Disable VPE MMU Virtual Address Translation 1 = Enable VPE MMU virtual address translation 0 = Disable VPE MMU virtual address translation

VPE MMU Translation Table Base Address Register

Register	Address	R/W	Description	Reset Value
VMMU_TTB	VPE_BA + 084	R/W	VPE MMU Translation Table Base Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	TTB Address	**32-bit Translation Table Base Address** This 32-bit physical address specifies the base address of a table in physical memory that contains level-one page table descriptors. It is 16KB boundary.

VPE MMU Page Fault Virtual Address Register

Register	Address	R/W	Description	Reset Value
VMMU_PFTVA	VPE_BA + 088	R/W	VPE MMU Page Fault Virtual Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	Page Fault Virtual Address	**32-bit Page Fault Virtual Address** This 32-bit byte address specifies the page fault virtual address when Page Fault Interrupt occurs.

VPE MMU Resume and Invalidate Command Register

Register	Address	R/W	Description	Reset Value
VMMU_CMD	VPE_BA + 08C	R/W	VPE MMU Resume and Invalidate Command	0x0000_0000

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

Bits	Descriptions	
[31:18]	Reserved	Reserved
[17]	MTLB_FAIL	**Flush VPE MMU Main TLB Entries Pass/Fail Status** 0 = Pass 1 = Fail
[16]	MTLB_FINISH	**Flush VPE MMU Main TLB Entries Finish/Busy Status** 0 = Busy 1 = Finish
[15:3]	Reserved	Reserved
[2]	FLUSH	**Flush VPE MMU Main TLB Entries of Main TLB SRAM** 0 = No flush. 1 = Write “1” to flush Main TLB entries. User should keep it at “0” normally.
[1]	INVALID	**Invalidate VPE MMU Micro TLB Entries** 0 = No invalidation. 1 = Write “1” to invalidate Micro TLB entries.

		User should keep it at “0” normally.
[0]	**RESUME**	**Resume VPE MMU Transaction** 0 = No Resume. 1 = Resume MMU transaction when Page Fault Interrupt occurs. It is auto-cleared after VPE MMU resumes the suspended transaction.

VPE MMU Level-One Page Table Entry 0 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT0	VPE_BA + 090	R/W	VPE MMU Level-One Page Table Entry 0 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 0	**32-bit Descriptor Information for Level-One Page Table Entry 0** It is the 32-bit descriptor information for the level-one page table entry 0. It is dedicated for the source Y channel.

VPE MMU Level-One Page Table Entry 1 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT1	VPE_BA + 094	R/W	VPE MMU Level-One Page Table Entry 1 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 1	32-bit Descriptor Information for Level-One Page Table Entry 1 It is the 32-bit descriptor information for the level-one page table entry 1. It is dedicated for the source U channel.

VPE MMU Level-One Page Table Entry 2 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT2	VPE_BA + 098	R/W	VPE MMU Level-One Page Table Entry 2 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 2	32-bit Descriptor Information for Level-One Page Table Entry 2 It is the 32-bit descriptor information for the level-one page table entry 2. It is dedicated for the source V channel.

VPE MMU Level-One Page Table Entry 3 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT3	VPE_BA + 09C	R/W	VPE MMU Level-One Page Table Entry 3 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 3	32-bit Descriptor Information for Level-One Page Table Entry 3 It is the 32-bit descriptor information for the level-one page table entry 3. It is dedicated for the destination channel with packet formats.

VPE MMU Level-One Page Table Entry 4 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT4	VPE_BA + 0A0	R/W	VPE MMU Level-One Page Table Entry 4 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 4	32-bit Descriptor Information for Level-One Page Table Entry 4 It is the 32-bit descriptor information for the level-one page table entry 4. It is dedicated for the source Y channel.

VPE MMU Level-One Page Table Entry 5 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT5	VPE_BA + 0A4	R/W	VPE MMU Level-One Page Table Entry 5 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 5	32-bit Descriptor Information for Level-One Page Table Entry 5 It is the 32-bit descriptor information for the level-one page table entry 5. It is dedicated for the source U channel.

VPE MMU Level-One Page Table Entry 6 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT6	VPE_BA + 0A8	R/W	VPE MMU Level-One Page Table Entry 6 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 6	32-bit Descriptor Information for Level-One Page Table Entry 6 It is the 32-bit descriptor information for the level-one page table entry 6. It is dedicated for the source V channel.

VPE MMU Level-One Page Table Entry 7 Register

Register	Address	R/W	Description	Reset Value
VMMU_L1PT7	VPE_BA + 0AC	R/W	VPE MMU Level-One Page Table Entry 7 Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Level-One Page Table Entry 7	32-bit Descriptor Information for Level-One Page Table Entry 7 It is the 32-bit descriptor information for the level-one page table entry 7. It is dedicated for the destination channel with packet formats.

VPE MMU Current Virtual Address Register

Register	Address	R/W	Description	Reset Value
VMMU_CVA	VPE_BA + 0B0	R	VPE MMU Current Virtual Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Current Virtual Address	32-bit Current Virtual Address It records the information of the current virtual address during the MMU translation process for the present.

VPE MMU Current Virtual Page Number Register

Register	Address	R/W	Description	Reset Value
VMMU_CVPN	VPE_BA + 0B4	R	VPE MMU Current Virtual Page Number Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Current Virtual Page Number	32-bit Current Virtual Page Number It records the information of the current virtual page number during the MMU translation process for the present.

VPE MMU Current Physical Address Register

Register	Address	R/W	Description	Reset Value
VMMU_CPA	VPE_BA + 0B8	R	VPE MMU Current Physical Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Current Physical Address	32-bit Current Physical Address It records the information of the current physical address during the MMU translation process for the present.

VPE MMU Current Physical Page Number Register

Register	Address	R/W	Description	Reset Value
VMMU_CPPN	VPE_BA + 0BC	R	VPE MMU Current Physical Page Number Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	Current Physical Page Number	32-bit Current Physical Page Number It records the information of the current physical page number during the MMU translation process for the present.

6.7 JPEG Codec

6.7.1 Overview

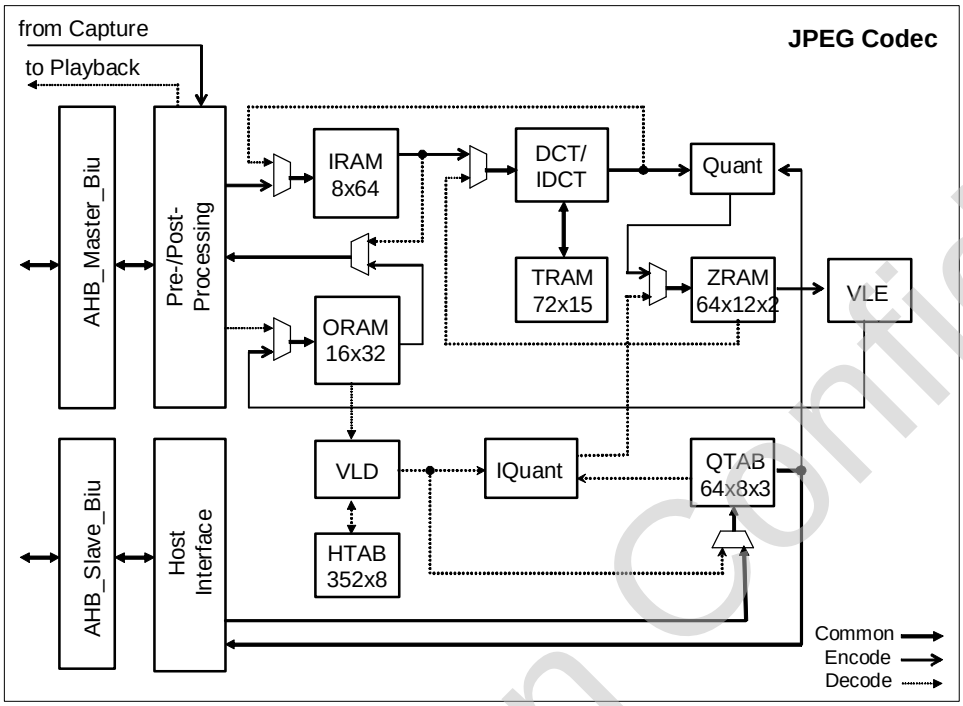


Figure 6.7-1 JPEG Codec block diagram

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.

6.7.2 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 (up to 8192x8192)
- 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
- On-the-Fly with video processing engine is supported

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
- Support Scatter-Gather mode for output frame buffer
- On-the-Fly with video processing engine is supported

6.7.3 JPEG Encode

6.7.3.1 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.

6.7.3.2 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.

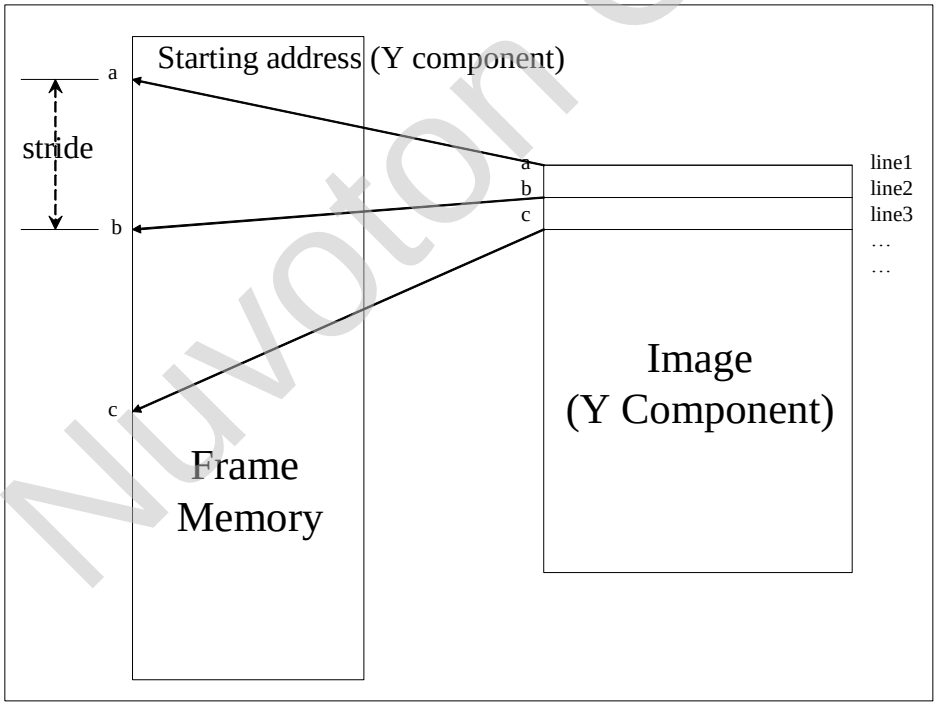


Figure 6.7-2 Image starting address and stride

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*.

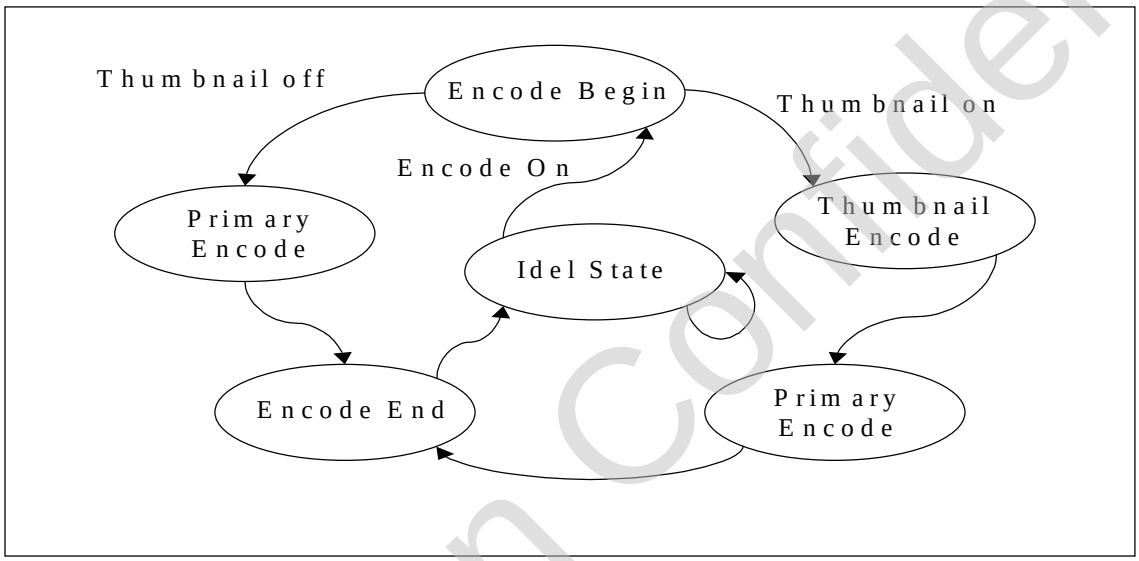


Figure 6.7-3 Primary and thumbnail encode

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.

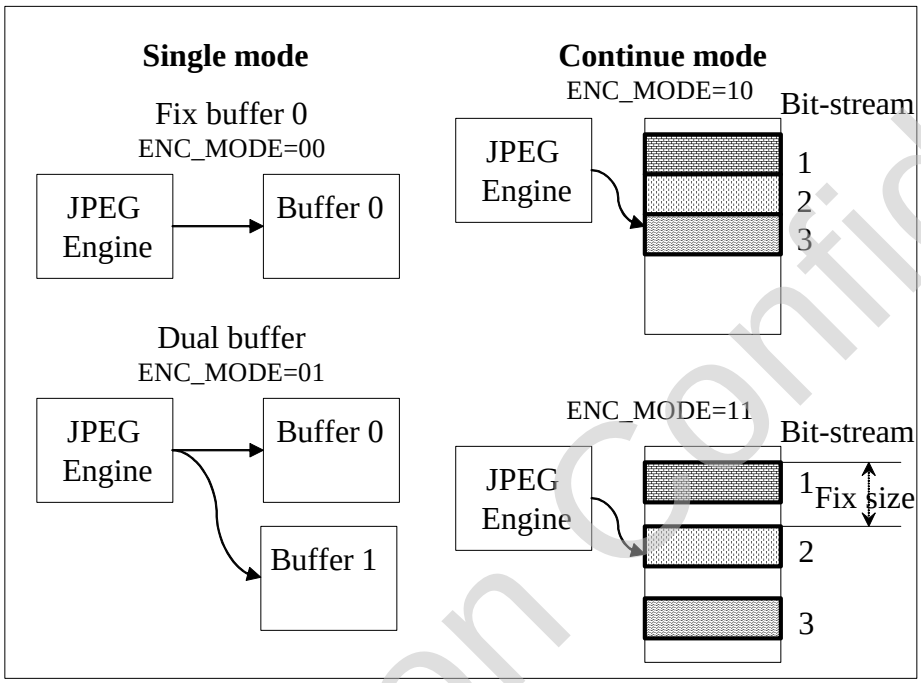


Figure 6.7-4 Single mode and continue mode

6.7.3.3 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.

6.7.4 JPEG Decode

6.7.4.1 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.

6.7.4.2 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*.**

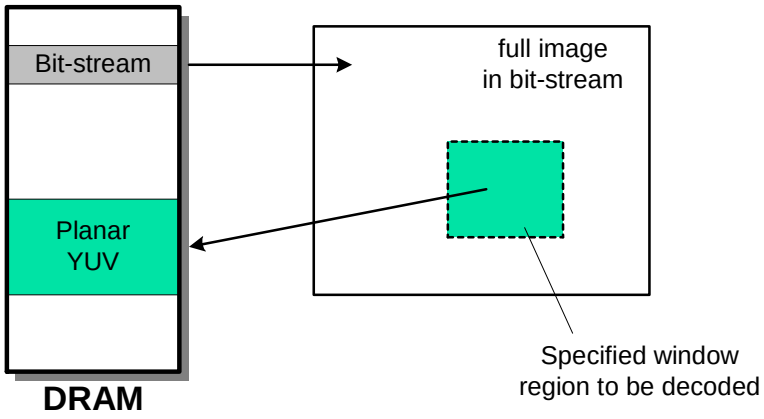


Figure 6.7-5 Specified window decode mode

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.

6.7.5 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*.

6.7.6 JPEG Engine 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/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 + 034~ JPG_BA + 038	R/W	Reserved	0x0000_0000
JDOWFBS	JPG_BA + 03C	R/W	Decoding Output Wait Frame Buffer Size	0xFFFF_FFFF
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
JPKTFBS	JPG_BA + 1C0	R/W	Decoding Output Wait Frame Buffer Size	0xFFFF_FFFF
JPKTFBSADR	JPG_BA + 1C4	R/W	Packet Buffer Start Address in Decoder Output W	
RESERVED	JPG_BA + 1C8 ~ JPG_BA + 1FC	R/W	Reserved	0x0000_0000

6.7.7 JPEG Engine Control Register

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	RESUMEO
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]	RESUMEO	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					Dec_Scatter_Gather	Dec_on_the_Fly	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
18	Dec_Output_Wait_Go	While the frame size and start address are setting, writing "1" will trigger the decoding output wait feature. The decoding processing will be halted and an interrupt will issued while the current frame buffer is full. The decoding process will be resumed and new frame buffer size and start address will be loaded while another "1" is written. Check 0x3C about the packet Frame Buffer Size.
17	Dec_On_the_Fly	1: Enable On_the_Fly path between JPEG and VPE 0: Disable
[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

JPEG Encode Primary Quantization-Table Control Register (JPRIQC)

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[12:8]				
23	22	21	20	19	18	17	16
P_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved			P_WIDTH[12:8]				
7	6	5	4	3	2	1	0
P_WIDTH[7:0]							

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28:16]	P_HEIGHT	Primary Encode and Packet Format Decode Image Height A 13-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 ceil((YSF/1024)*(height of image))
[15:13]	Reserved	Reserved

[12:0]	P_WIDTH	Primary Encode and Packet Format Decode Image Width A 13-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 ceil((XSF/1024)*(width of image)/16) * (16 + Block1), while Block1 = 1 when mod((XSF/1024)*(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[12:8]				
23	22	21	20	19	18	17	16
T_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved			T_WIDTH[12:8]				
7	6	5	4	3	2	1	0
T_WIDTH[7:0]							

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28:16]	T_HEIGHT	Thumbnail Encode Image Height (For Planar Format Only) A 13-bit value specifies the height of encoded JPEG thumbnail image. The value is equal to the size after scaling-down.
[15:13]	Reserved	Reserved
[12:0]	T_WIDTH	Thumbnail Encode Image Width (For Planar Format Only) A 13-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					BAbort
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	
[31:29]	Reserved	Reserved
[28]	DOW_INTE	Decoding Output Wait Interrupt Enable
[27:25]	Reserved	Reserved
[24]	DOW_INTS	Status of Decoding Output Wait. 0: No Interrupt 1: Decoding packet buffer is full Writing "1" will clear the status
[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]	BAbort	JPEG Memory Access Error Status (Read-Only) 0 = Normal operation 1 = Wrong frame memory space is accessed
[15]	DOW_INTE	Decoding Output Wait Interrupt Enable
[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	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]	DOW_INTS	Status of Decoding Output Wait. 0: No Interrupt 1: Decoding packet buffer is full Writing "1" will clear the status
[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 Decoding Output Wait Frame Buffer Size

Register	Address	R/W	Description	Default Value
JDOWFBS	JPG_BA + 03C	R/W	JPEG Decoding Output Wait Frame Buffer Size	0xFFFF_FFFF

Bits	Descriptions	
[31:0]	JDOWFBS	JPEG Decoding Output Wait Frame Buffer Size The JPEG output decoding process will be paused while this buffer is full and decoding process will be resumed while new buffer size/address is set and "1" is written to bit[18] of register 0x08. Note: The buffer size must be multiples of MCU-line.

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[9:8]	
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[9:8]	
7	6	5	4	3	2	1	0
MCU_S_X[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	MCU_S_Y	MCU(Minimum Coded Unit) Start Position Y For Window Decode Mode A 10-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.
[15:10]	Reserved	Reserved
[9:0]	MCU_S_X	MCU Start Position X For Window Decode Mode A 10-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[9:8]	
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[9:8]	
7	6	5	4	3	2	1	0
MCU_E_X[7:0]							

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	MCU_E_Y	MCU End Position Y For Window Decode Mode A 10-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.
[15:10]	Reserved	Reserved
[9:0]	MCU_E_X	MCU End Position X For Window Decode Mode A 10-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[12:8]				
7	6	5	4	3	2	1	0
WD_WIDTH[7:0]							

Bits	Descriptions	
[31:13]	Reserved	Reserved
[12:0]	WD_WIDTH	Image Width (Y-Stride) For Window Decode Mode A 13-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 *JDBCR[0]* 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 Y420, 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	Reserved
[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	Reserved

[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 *(byte address)* 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 *(byte address)* 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 (*byte address of word aligned*).

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 (*byte address of word aligned)*. If a "1" is written to Decoding_Output_Wait_Go (Bit[18] of 0x08), the content of this register will be reloaded as start address of next packet frame buffer and the content of 0x1C0 will be loaded as buffer size in next packet ytransferring.

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 (*byte address of word aligned)*

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 *byte* offset of memory address of vertical adjacent line for Y component (*byte address of word aligned)*. 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 *byte* offset of memory address of vertical adjacent line for U component (*byte address of word aligned)*.

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 *byte* offset of memory address of vertical adjacent line for V component (*byte address of word aligned)*.

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 (*byte address of word aligned)*.

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 (*byte address of word aligned)*.

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 *byte* 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 *byte* 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[13:8]					
23	22	21	20	19	18	17	16
S_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved		S_WIDTH[13:8]					
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 ceil(4096/8192). The height is 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 ceil(4096/8192). The width is 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 Encode Source Image Height 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. *The other requirements are the same as QTAB0 described above.*
[23:16]	QTAB1_2	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. *The other requirements are the same as QTAB0 described above.*
[23:16]	QTAB2_2	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	JPEG Quantization-Table 2 – 0 An 8-bit value specifies one element of the Quantization-Table 2.

6.8 Video decoder

6.8.1 Features

The video decoder is a multi-standard decoder that can handle H.263P3, MPEG-4 SP, H.264 BP, VC-1 MP and MPEG2 MP@ML. This decoder includes following features

- **Multi-standard video decoder**
  - **MPEG-4 part-II simple profile decoding**
  - **H.264/AVC baseline profile decoding**
  - **VC-1(WMV9) main profile decoding**
  - **MPEG2 main profile decoding**
- **Decoding tools**
  - **H.264 BP**
    - **Supports all H.264 baseline profile features**
    - **Variable block size (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4)**
    - **Error detection, concealment and error resilience tools such as FMO, ASO etc.**
  - MPEG-4 SP
    - Supports all MPEG-4 simple profile features
    - 4MV with unrestricted motion vector
    - Error resilience tools: re-sync marker, data partitioning with RVLC
  - H.263 P3
    - Baseline profile plus Annex I,J,K (RS=0 and ASO=0) and T are supported
  - VC1-MP
    - Supports all VC-1 main profiles features
    - Bidirectional frame
    - Variable-sized transform and loop filter
    - Overlapped transform
    - 1 or 4 motion vector with UMV (quarter-pixel motion compensation)
    - Extended motion vector
    - Dynamic resolution change
    - Intensity compensation
    - Range adjustment
  - MPEG2 MP@ML
    - ISO/IEC 13818-2 fully compatible
    - I,P and B frame support
    - Field/Frame picture suppprts
  - MPEG-1 is supported
  - Minimum decoding size is 16-pixel in both horizontal and vertical.
- Pre/post rotation/mirroring

- l 90xn (n=0,1,2,3) degree rotation /w mirroring for decoded image
- n Up to 720x480 30fps decoding

6.8.2 Block Diagram

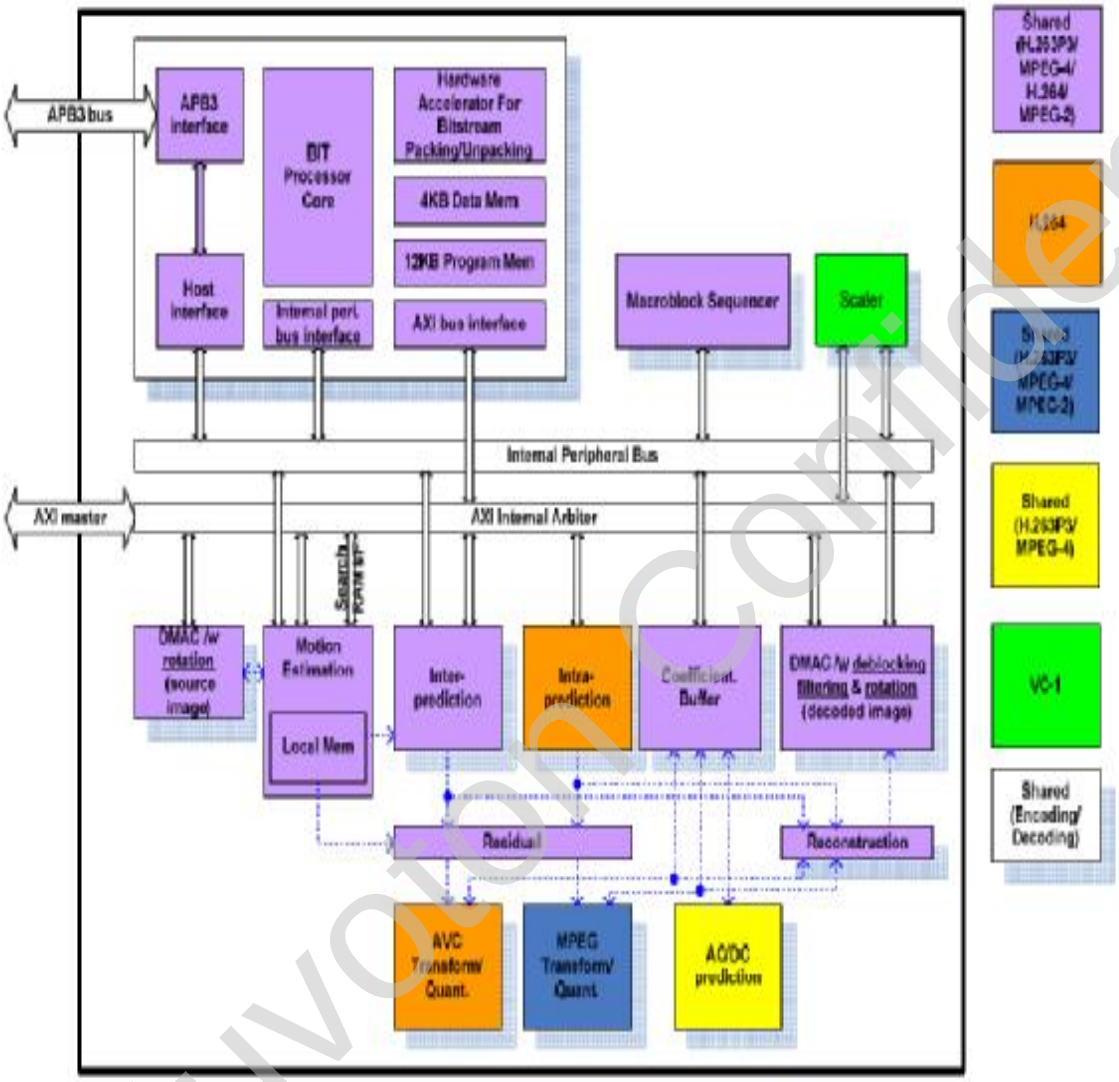


Figure 1-1. Block Diagram

The video decoder is a high performance multi-format video decoder that supports H.263P3, MPEG4 SP, H.264 BP and VC-1 MP, MPEG2 MP. This decoder consists of a embedded processor, called BIT processor, and video decoder core module. The BIT processor parses the input bitstream and controls the video decoder. To speed up the bitstream processing, some hardware accelerators are included in the BIT processor. The program and data for the BIT processor is downloaded through AMBA APB bus and AMBA AXI bus.

The video decoder is optimized to reduce logic gate count with sharing large part of sub-modules for multi-standard. The motion compensation module uses a search RAM to reduce the bandwidth on external SDRAM. Generally, motion compensation reads reference pixel data several times. The motion compensation module loads reference pixel data from external SDRAM and store them to search RAM. The search RAM is accessed through the AMBA AHB.

The macroblock sequencer module schedules the processing flow of the functional blocks of video decoder to reduce loads on BIT processor and firmware complexity. This decoder includes a rotation/mirroring module. In case of rotation and/or mirroring, the decoded image with any rotation and mirroring and rotated and/or mirrored version are written to external memory.

The internal AXI arbiter module arbitrates requests from the internal DMA controller for ease of integration to SoC.

Figure 1.1 shows the functional block diagram of this decoder and Figure 1.2 shows the role of BIT processor and video decoder core and how to interface with application software. Basically, at the frame level, the host processor communicates with this decoder through provided API's. To given the video decoder more flexibility and debugging capability, all process related to bitstream are assigned to the BIT processor.

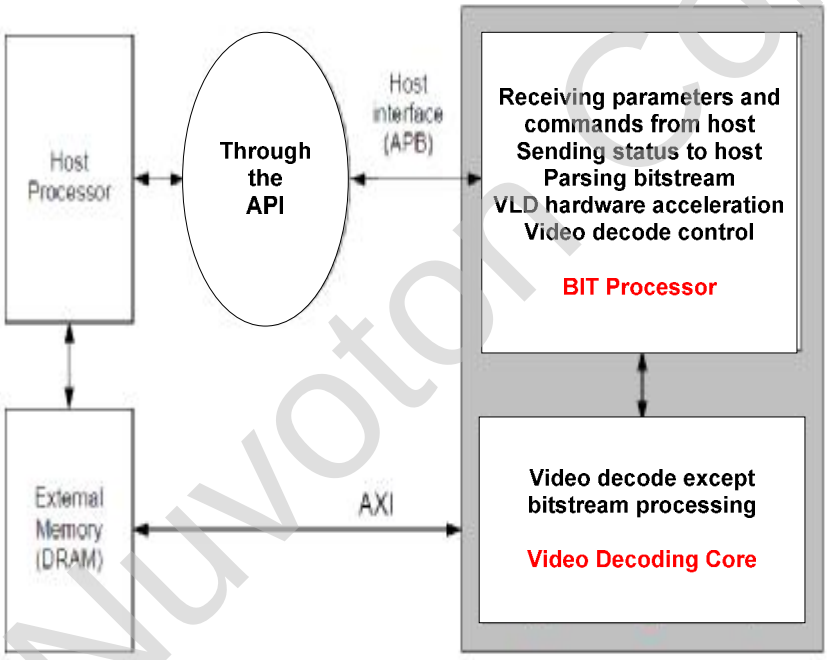


Figure 1-2. Role of the BIT processor and video decoding module

6.9 CAPTURE Engine

6.9.1 Overview

CAPTURE engine is designed to capture image data from sensor or TV decoder. After capturing or fetching image data, capture engine processes the image data, and then FIFO output them into frame buffer.

6.9.2 Capture Functional Block Diagram

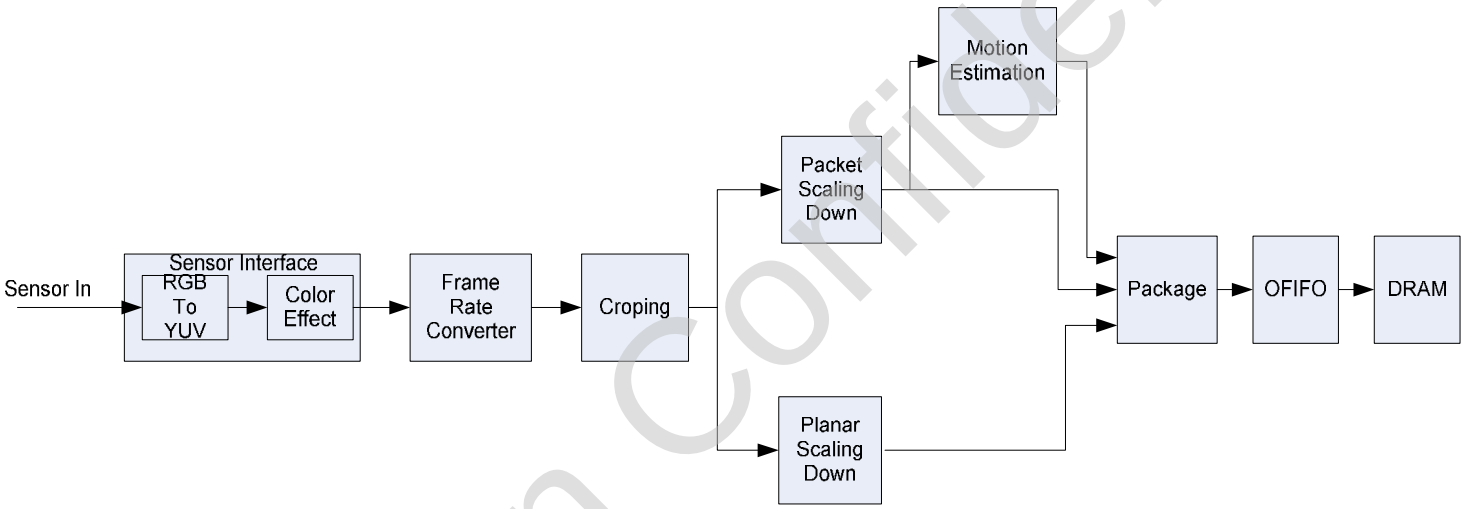


Figure 6.9-1 Capture Functional Block Diagram

6.9.3 Feature

- **Support 8-bits RGB565 sensor.**
- **Support 8-bits YUV422 sensor.**
- **Support TV decoder interface (compliant with CCIR601 and CCIR656).**
- **Support CCIR601 YCbCr color range scale to full YUV color range.**
- **Support 4 packaging format for packet data output: YUYV, Y only, RGB565, RGB555.**
- **Support YUV422 planar data output.**
- **Support CROP function to crop input image to the required size for digital application.**
- **Support down scaling function to scale input image to the required size for digital application.**
- **Support frame rate control**

- Support field detection and even/odd field skip mechanism
- Support packet output dual buffer control through hardware buffer controller.
- Support negative/sepia/posterization color effect
- Support two independent video in interfaces

Pin Mux of Two Video In Interface

	Video capture 0		Video capture 1	
	Muxed Pin	Register Setting	Muxed Pin	Register Setting
SPDADA[7:0]	SPDATA[7:0]	GPBFUN[25:10]=0xFFFF	LVDATA[15:8]	GPCFUN[31:16]=0x5555
SFIELD	SFIELD	GPBFUN[9:8]=2'b11	GPA[1]	GPAFUN[3:2]=2'b01
SVSYNC	SVSYNC	GPBFUN[7:6]=2'b11	LVDATA[16]	GPEFUN[3:2]=2'b01
			TDO	GPFFUN[21:20]=2'b10
SHSYNC	SHSYNC	GPBFUN[5:4]=2'b11	LVDATA[17]	GPEFUN[1:0]=2'b01
			TRSR_	GPFFUN[23:22]=2'b10
SPCLK	SPCLK	GPBFUN[3:2]=2'b11	GPA[0]	GPAFUN[1:0]=2'b01
			TMS	GPFFUN[17:16]=2'b10
SCLK0	SCLKO	GPBFUN[1:0]=2'b11	GPA[1]	GPAFUN[3:2]=2'b01
			TDI	GPFFUN19:18]=2'b10

6.9.4 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
VIN_BA0 = 0xB100_3000 VIN_BA1 = 0xB100_3800				
VINCTL	VIN_BA+0x00	R/W	Capture Control Register	0x0000_0040
VINPAR	VIN_BA+0x04	R/W	Capture Parameter Register	0x0000_0000
VININT	VIN_BA+0x08	R/W	Capture Interrupt Register	0x0000_0000
VPOSTERIZE	VIN_BA+0x0c	R/W	YUV Component Posterizing Factor Register	0x0000_0000
VINMD	VIN_BA+0x10	R/W	Motion Detection Register	0x0000_0000
MDADDR	VIN_BA+0x14	R/W	Motion Detection Output Address Register	0x0000_0000
MDYADDR	VIN_BA+0x18	R/W	Motion Detection Temp YOutput Address Register	0x0000_0000
VSEPIA	VIN_BA+0x1c	R/W	Sepia Effect Control Register	0x0000_0000
VINCWSP	VIN_BA+0x20	R/W	Cropping Window Starting Address Register	0x0000_0000
VINCWS	VIN_BA+0x24	R/W	Cropping Window Size Register	0x0000_0000
VINPKDSL	VIN_BA+0x28	R/W	Packet Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000
VINPNDSL	VIN_BA+0x2c	R/W	Planar Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000
VINFRC	VIN_BA+0x30	R/W	Scaling Frame Rate Factor Register	0x0000_0000
VSTRIDE	VIN_BA+0x34	R/W	Frame Output Pixel Stride Register	0x0000_0000
VFIFO	VIN_BA+0x3c	R/W	FIFO threshold Register	0x0000_0000
CMPADDR	VIN_BA+0x40	R/W	Compare Packet Memory Base Address Register	0xFFFF_FFFC
Reserve	VIN_BA+0x44		Reserve	
VINPKDSM	VIN_BA+0x48	R/W	Packet Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000
VINPNDSM	VIN_BA+0x4c	R/W	Planar Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000
CURADDRP	VIN_BA+0x50	R	Current Packet System Memory Address Register	0x0000_0000
CURADDRY	VIN_BA+0x54	R	Current Planar Y System Memory Address	0x0000_0000

			Register	
CURADDRU	VIN_BA+0x58	R	Current Planar U System Memory Address Register	0x0000_0000
CURADDRV	VIN_BA+0x5c	R	Current Planar V System Memory Address Register	0x0000_0000
PACBA0	VIN_BA+0x60	R/W	System Memory Packet Base Address Register	0x0000_0000
PACBA1	VIN_BA+0x64	R/W	System Memory Packet Base Address Register	0x0000_0000
Reserve	VIN_BA+0x68~6c	R/W	Reserve	0x0000_0000
YBA0	VIN_BA+0x80	R/W	System Memory Planar Y Base Address Register	0x0000_0000
UBA0	VIN_BA+0x84	R/W	System Memory Planar U Base Address Register	0x0000_0000
VBA0	VIN_BA+0x88	R/W	System Memory Planar V Base Address Register	0x0000_0000

6.9.5 Capture Control Register Description

Capture Control Register

Register	Address	R/W	Description	Reset Value
VINCTRL	VIN_BA+0x00	R/W	Capture Control Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							VPRST
23	22	21	20	19	18	17	16
Reserved			UPDATE	Reserved			SHUTTER
15	14	13	12	11	10	9	8
Reserved							VINBIST
7	6	5	4	3	2	1	0
Reserved	PKEN	PNEN	Reserved	ADDRSW	FBMODE	Reserved	VINEN

Bits	Descriptions	
[31:25]	Reserved	Reserved
[24]	VPRST	Capture Reset 0 = Disable, normal operation 1 = Reset the Capture except Registers
[23:21]	Reserved	Reserved
[20]	UPDATE	Update Register at New Frame 0 = update when capture engine is not enable. 1 = update when capture a new frame. Auto clear to 0 when register updated.
[19:16]	Reserved	Reserved
[16]	SHUTTER	Capture One Frame 0 = Disable, normal operation. 1 = Capture One Frame. Enable the capture engine automatically. After a frame had been captured, disable the capture engine automatically.
[15:9]	Reserved	Reserved
[8]	VINBIST	Software BIST Enable

		0 = Disable 1 = Enable
[7]	Reserved	Reserved
[6]	PKEN	Packet Output Enable 0 = Disable 1 = Enable
[5]	PNEN	Planar Output Enable 0 = Disable 1 = Enable
[4]	Reserved	Reserved
[3]	ADDRSW	Packet Buffer Address Switch 0 = Select 1 = Enable
[2]	FBMODE	Frame Buffer Switch Mode 0 = Packet Buffer Address select by ADDR_SW 1 = Hardware Buffer Address from Buffer Switch Controller
[1]	Reserved	Reserved
[0]	VINEN	Capture Engine Enable 0 = Disable. The Engine CLK will stop. 1 = Enable

Capture Parameter Register

Register	Address	R/W	Description	Reset Value
VINPAR	VIN_BA+0x04	R/W	Capture Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			VINBFIN	BFAIL			
23	22	21	20	19	18	17	16
Reserved			FLDID	Reserved		FLD1EN	FLD0EN
15	14	13	12	11	10	9	8
FLDDETP	FLDDETM	FLDSWAP	Color Effect		VSP	HSP	PCLKP
7	6	5	4	3	2	1	0
PNFMT	RANGE	OUTFMT		PDORD		SNRTYPE	INFMT

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28]	VINBFIN	BIST Finish [Read Only] 0 = even filed 1 = odd field
[27:24]	BFAIL	BIST Fail Flag [Read Only] [24] = 1 : Packet OFIFO BIST Fail [25] = 1 : Planar Y OFIFO BIST Fail [26] = 1 : Planar U OFIFO BIST Fail [27] = 1 : Planar V OFIFO BIST Fail
[23:21]	Reserved	Reserved
[20]	FLDID	Field ID [Read Only] 0 = even filed 1 = odd field
[19:18]	Reserved	Reserved
[17]	FLD1EN	Field 1 Input Enable 0 = Disable. 1 = Enable.
[16]	FLD0EN	Field 0 Input Enable

		0 = Disable. 1 = Enable.
[15]	FLDDETP	Field Detect Position 0 = Vsync start 1 = Vsync end
[14]	FLDDETM	Field Detect Mode in CCIR601 0 = Detect field by Vsync & Hsync 1 = Detect field by input FIELD PIN
[13]	FLDSWAP	Swap Input Field 0 = input field:1 for odd fields, field:0 for even fields (default) 1 = input field:0 for odd fields, field:1 for even fields
[12:11]	Color_EFFect	Special Color Effect Processing 00: Normal Color 01: Sepia effect, corresponding U,V component value is set at register VSEPIA 10: Negative picture 11: Posterize image, the Y, U, V components posterizing factor are set at register VPOSTERIZE
[10]	VSP	Sensor Vsync Polarity 0 = sync Low 1 = sync High
[9]	HSP	Sensor Hsync Polarity 0 = sync Low 1 = sync High
[8]	PCLKP	Sensor Pixel Clock Polarity 0 = Input video data and signals are latched by falling edge of Pixel Clock 1 = Input video data and signals are latched by rising edge of Pixel Clock
[7]	PNFMT	Planar Output YUV Format 0 = YUV422 1 = YUV420
[6]	RANGE	Scale Input YUV CCIR601 color range to full range 0 = default 1 = scale to full range
[5:4]	OUTFMT	Image Data Format Output to System Memory

<table>
<tr><td></td><td></td><td>00 = YCbCr422
01 = only output Y
10 = RGB555
11 = RGB565</td></tr>
<tr><td>[3:2]</td><td>PDORD</td><td>Sensor Input Data Order
If INFMT = 0 (YCbCr),
Byte 0 1 2 3
00 = Y0 U0 Y1 V0
01 = Y0 V0 Y1 U0
10 = U0 Y0 V0 Y1
11 = V0 Y0 U0 Y1
If INFMT = 1 (RGB565),
00
<table><tr><th>Byte</th><th colspan="2">Bit [7:0]</th></tr><tr><td>0</td><td>R[4:0]</td><td>G[5:3]</td></tr><tr><td>1</td><td>G[2:0]</td><td>B[4:0]</td></tr></table>01
<table><tr><th>Byte</th><th colspan="2">Bit [7:0]</th></tr><tr><td>0</td><td>B[4:0]</td><td>G[5:3]</td></tr><tr><td>1</td><td>G[2:0]</td><td>R[4:0]</td></tr></table>10
<table><tr><th>Byte</th><th colspan="2">Bit [7:0]</th></tr><tr><td>0</td><td>G[2:0]</td><td>B[4:0]</td></tr><tr><td>1</td><td>R[4:0]</td><td>G[5:3]</td></tr></table>11
<table><tr><th>Byte</th><th colspan="2">Bit [7:0]</th></tr><tr><td>1</td><td>G[2:0]</td><td>R[4:0]</td></tr><tr><td>0</td><td>B[4:0]</td><td>G[5:3]</td></tr></table></td></tr>
<tr><td>[1]</td><td>SNRTYPE</td><td>Sensor Input Type
0 = CCIR601
1 = CCIR656, VSync & Hsync embedded in data signal</td></tr>
<tr><td>[0]</td><td>INFMT</td><td>Sensor Input Data Format</td></tr>
</table>

		0 = YCbCr422 1 = RGB565

Capture Interrupt Register

Register	Address	R/W	Description	Reset Value
VININT	VIN_BA+0x08	R/W	Capture Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			MDINTEN	ADDRMEN	Reserved	MEINTEN	VINTEN
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			MDINT	ADDRMINT	Reserved	MEINT	VINT

Bits	Descriptions	
[31:21]	Reserved	Reserved
[20]	MDINTEN	Motion Detection Output Finish Interrupt Enable
[19]	ADDRMEN	Address Match Interrupt Enable
[18]	Reserved	Reserved
[17]	MEINTEN	System Memory Error Interrupt Enable 0 = Disable 1 = Enable
[16]	VINTEN	Video Frame End Interrupt Enable 0 = Disable 1 = Enable
[15:5]	Reserved	Reserved
[4]	MDINT	Motion Detection Output Finish Interrupt. If read this bit shows 1 Motion Detection Output Finish Interrupt occurs. Write 1 to clear it.
[3]	ADDRMINT	Memory Address Match Interrupt. If read this bit shows 1 Memory Address Match Interrupt occurs.

		Write 1 to clear it.
[2]	Reserved	Reserved
[1]	MEINT	Bus Master Transfer Error Interrupt If read this bit shows 1, Transfer Error occurs. Write 1 to clear it.
[0]	VINT	Frame End Interrupt If read this bit shows 1, received a frame complete. Write 1 to clear it.

YUV Component Posterizing Factor Register

Register	Address	R/W	Description	Reset Value
VPOSTERIZE	VIN_BA+0x0C	R/W	YUV Component Posterizing Factor Register	0x0000_0000

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Y Component Posterizing Factor	Y Component Posterizing Factor Final_Y_Out = Original_Y[7:0] & Y_Posterizing_Factor
[15:8]	U Component Posterizing Factor	U Component Posterizing Factor Final_U_Out = Original_U[7:0] & U_Posterizing_Factor
[7:0]	V Component Posterizing Factor	V Component Posterizing Factor Final_V_Out = Original_V[7:0] & V_Posterizing_Factor

Motion Detection Register

Register	Address	R/W	Description	Reset Value
VINMD	VIN_BA+0x10	R/W	Motion Detection Register	0x0000_0000

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

Bits	Descriptions	
[31:21]	Reserved	Reserved
[20:16]	MDTHR	Motion Detection Differential Threshold
[15:12]	Reserved	Reserved
[11:10]	MDDF	Motion Detection Detect Frequency 00 = each frame 01 = every 2 frame 10 = every 3 frame 11 = every 4 frame
[9]	MDSM	Motion Detection Save Mode 0 = 1 bit DIFF + 7 bit Y Differential 1 = 1 bit DIFF only
[8]	MDBS	Motion Detection Block Size 0 = 16x16 1 = 8x8
[7:1]	Reserved	Reserved

[0]	MDEN	Motion Detection Enable 0 = Disable 1 = Enable

Motion Detection Output Address Register

Register	Address	R/W	Description	Reset Value
MDADDR	VIN_BA+0x14	R/W	Motion Detection Output Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	MDADDR	Motion Detection Output Address Register (word alignment)

Motion Detection Temp Y Output Address Register

Register	Address	R/W	Description	Reset Value
MDYADDR	VIN_BA+0x18	R/W	Motion Detection Temp Y Output Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	MDYADDR	Motion Detection Temp Y Output Address Register (word alignment)

Sepia Effect Control Register

Register	Address	R/W	Description	Reset Value
VSEPIA	VIN_BA+0x1C	R/W	Sepia Effect Control Register	0x0000_8080

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
Sepia U Component							
7	6	5	4	3	2	1	0
Sepia V Component							

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Reserved	Reserved
[15:8]	Sepia U Component	Define the constant U component while "Sepia" color effect is turned on.
[7:0]	Sepia V Component	Define the constant V component while "Sepia" color effect is turned on.

Cropping Window Starting Address Register

Register	Address	R/W	Description	Reset Value
VINCWSP	VIN_BA+0x20	R/W	Cropping Window Starting Address Register	0x0000_0000

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

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:16]	CWSPV	Cropping Window Vertical Starting Address
[15:12]	Reserved	Reserved
[11:0]	CWSPH	Cropping Window Horizontal Starting Address

Cropping Window Size Register

Register	Address	R/W	Description	Reset Value
VINCWS	VIN_BA+0x24	R/W	Cropping Window Size Register	0x0000_0000

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

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:16]	CWSH	Cropping Image Window Height
[15:12]	Reserved	Reserved
[11:0]	CWSW	Cropping Image Window Width

Packet Scaling Vertical/Horizontal Factor Register (LSB)

Register	Address	R/W	Description	Reset Value
VINPKDSL	VIN_BA+0x28	R/W	Packet Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000

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

Bits	Descriptions	
[31:24]	PKDSVNL	Packet Scaling Vertical Factor N (Lower 8-bit) Specifies the lower 8-bit of numerator part (N) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSVNH) to form a 16-bit numerator of vertical factor.
[23:16]	PKDSVML	Packet Scaling Vertical Factor M (Lower 8-bit) Specifies the lower 8-bit of denominator part (M) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSVMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image height * N/M. *The value of N must be equal or less than M.*
[15:8]	PKDSHNL	Packet Scaling Horizontal Factor N (Lower 8-bit) Specifies the lower 8-bit of numerator part (N) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSHNH) to form a 16-bit numerator of horizontal factor.
[7:0]	PKDSHML	Packet Scaling Horizontal Factor M (Lower 8-bit) Specifies the lower 8-bit of denominator part (M) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PKDSHMH) to form a 16-bit

		denominator (M) of vertical factor. **The output image width will be equal to the image width * N/M.** ***The value of N must be equal or less than M.***

Planar Scaling Vertical / Horizontal Factor Register (LSB)

Register	Address	R / W	Description	Reset Value
VINPNDSL	VIN_BA+0x2c	R/W	Planar Scaling Vertical/Horizontal Factor Register (LSB)	0x0000_0000

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

Bits	Descriptions	
[31:24]	PNDSVNL	Planar Scaling Vertical Factor N (Lower 8-bit) Specifies the lower 8-bit of numerator part (N) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSVNH) to form a 16-bit numerator of vertical factor.
[23:16]	PNDSVML	Planar Scaling Vertical Factor M (Lower 8-bit) Specifies the lower 8-bit of denominator part (M) of the vertical scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSVMH) to form a 16-bit denominator (M) of vertical factor. The output image width will be equal to the image height * N/M. *The value of N must be equal or less than M.*
[15:8]	PNDSHNL	Planar Scaling Horizontal Factor N (Lower 8-bit) Specifies the lower 8-bit of numerator part (N) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSHNH) to form a 16-bit numerator of horizontal factor.
[7:0]	PNDSHML	Planar Scaling Horizontal Factor M (Lower 8-bit) Specifies the lower 8-bit of denominator part (M) of the horizontal scaling factor. The lower 8-bit will be cascaded with higher 8-bit (PNDSHMH) to form a 16-bit

		denominator (M) of vertical factor. The output image width will be equal to the image width * N/M. ***The value of N must be equal or less than M.***

Packet Scaling Vertical/Horizontal Factor Register (MSB)

Register	Address	R/W	Description	Reset Value
VINPKDSH	VIN_BA+0x48	R/W	Packet Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000

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

Bits	Descriptions	
[31:24]	PKDSVNH	Packet Scaling Vertical Factor N (Higher 8-bit) Specifies the higher 8-bit of numerator part (N) of the vertical scaling factor. Please refer to register "VINPKDSL" to check the cooperation between these two registers.
[23:16]	PKDSVMH	Packet Scaling Vertical Factor M (Higher 8-bit) Specifies the lower 8-bit of denominator part (M) of the vertical scaling factor. Please refer to register "VINPKDSL" to check the cooperation between these two registers.
[15:8]	PKDSHNH	Packet Scaling Horizontal Factor N (Higher 8-bit) Specifies the lower 8-bit of numerator part (N) of the horizontal scaling factor. Please refer to register "VINPKDSL" to get the detail operation.
[7:0]	PKDSHMH	Packet Scaling Horizontal Factor M (Higher 8-bit) Specifies the lower 8-bit of denominator part (M) of the horizontal scaling factor. Please refer to register "VINPKDSL" to get the detail operation.

Planar Scaling Vertical / Horizontal Factor Register (MSB)

Register	Address	R / W	Description	Reset Value
VINPNDSH	VIN_BA+0x4c	R/W	Planar Scaling Vertical/Horizontal Factor Register (MSB)	0x0000_0000

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

Bits	Descriptions	
[31:24]	PNDSVNH	Planar Scaling Vertical Factor N (Higher 8-bit) Specifies the higher 8-bit of numerator part (N) of the vertical scaling factor. For detail programming, please refer to register "VINPNDSL"
[23:16]	PNDSVMH	Planar Scaling Vertical Factor M (Higher 8-bit) Specifies the lower 8-bit of denominator part (M) of the vertical scaling factor. For detail programming, please refer to register "VINPNDSL"
[15:8]	PNDSHNH	Planar Scaling Horizontal Factor N (Higher 8-bit) Specifies the higher 8-bit of numerator part (N) of the horizontal scaling factor. For detail programming, please refer to register "VINPNDSL"
[7:0]	PNDSHMH	Planar Scaling Horizontal Factor M (Higher 8-bit) Specifies the higher 8-bit of denominator part (M) of the horizontal scaling factor For detail programming, please refer to register "VINPNDSL"

Scaling Frame Rate Factor Register

Register	Address	R/W	Description	Reset Value
VINFRC	VIN_BA+0x30	R/W	Scaling Frame Rate Factor Register	0x0000_0000

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

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13:8]	FRCN	Scaling Frame Rate Factor N Specifies the denominator part (N) of the frame rate scaling factor.
[7:6]	Reserved	Reserved
[5:0]	FRCM	Scaling Frame Rate Factor M Specifies the denominator part (M) of the frame rate scaling factor. The output image frame rate will be equal to input image frame rate * (N/M). *The value of N must be equal or less than M.*

Output Frame Pixel Stride Width

Register	Address	R/W	Description	Reset Value
VSTRIDE	VIN_BA+0x34	R/W	Frame Output Pixel Stride Width Register	0x0000_0000

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

Bits	Descriptions	
[31:28]	Reserved	Reserved
[29:16]	PNSTRIDE	Planar Frame Output Pixel Stride Width The planar pipe output pixel stride size.
[15:12]	Reserved	Reserved
[13:0]	PKSTRIDE	Packet Frame Output Pixel Stride Width The packet pipe output pixel stride size.

FIFO Threshold Register

Register	Address	R / W	Description	Reset Value
VFIFO	VIN_BA+0x3c	R/W	FIFO Threshold Register	0x0000_0000

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

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27:24]	FTHP	Packet FIFO Threshold
[23:19]	Reserved	Reserved
[19:16]	FTHY	Planar Y FIFO Threshold
[15:11]	Reserved	Reserved
[10:8]	FTHU	Planar U FIFO Threshold
[7:3]	Reserved	Reserved
[2:0]	FTHV	Planar V FIFO Threshold

Compare Memory Address Register

Register	Address	R/W	Description	Reset Value
CMPADDR	VIN_BA+0x40	R/W	Compare Memory Base Address Register	0xffff_fffc

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

Bits	Descriptions	
[31:0]	CMPADDR	Compare Memory Base Address Word align address, ignore the bits [1:0]

Current Packet Output Memory Size Register

Register	Address	R/W	Description	Reset Value
CURADDRP	VIN_BA+0x50	R	Current Packet Output Memory Size Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	CURADDRP	Current Packet Output Memory Size (Bytes)

Current Planar Y Output Memory Size Register

Register	Address	R / W	Description	Reset Value
CURADDRY	VIN_BA+0x54	R	Current Planar Y Output Memory Size Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	CURADDRY	Current Planar Y Output Memory Size (Bytes)

Current Planar U Output Memory Size Register

Register	Address	R / W	Description	Reset Value
CURADDRU	VIN_BA+0x58	R	Current Planar U Output Memory Size Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	CURADDRU	Current Planar U Output Memory Size (Bytes)

Current Planar V Output Memory Size Register

Register	Address	R / W	Description	Reset Value
CURADDRV	VIN_BA+0x5c	R	Current Planar V Output Memory Size Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	CURADDRV	Current Planar V Output Memory Size (Bytes)

System Memory Packet Base Address 0 Register

Register	Address	R/W	Description	Reset Value
PACBA0	VIN_BA+0x60	R/W	System Memory Packet Base Address 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
PACBA0[31:24]							
23	22	21	20	19	18	17	16
PACBA0[23:16]							
15	14	13	12	11	10	9	8
PACBA0[15:8]							
7	6	5	4	3	2	1	0
PACBA0[7:0]							

Bits	Descriptions	
[31:0]	PACBA0	System Memory Packet Base Address 0 Word align address, ignore the bits [1:0]

System Memory Packet Base Address 1 Register

Register	Address	R/W	Description	Reset Value
PACBA1	VIN_BA+0x64	R/W	System Memory Packet Base Address 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
PACBA1[31:24]							
23	22	21	20	19	18	17	16
PACBA1[23:16]							
15	14	13	12	11	10	9	8
PACBA1 [15:8]							
7	6	5	4	3	2	1	0
PACBA1[7:0]							

Bits	Descriptions	
[31:0]	PACBA1	System Memory Packet Base Address Word align address, ignore the bits [1:0]

System Memory Planar Y Base Address Register

Register	Address	R / W	Description	Reset Value
YBA0	VIN_BA+0x80	R/W	System Memory Planar Y Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
YBA0[31:24]							
23	22	21	20	19	18	17	16
YBA0[23:16]							
15	14	13	12	11	10	9	8
YBA0[15:8]							
7	6	5	4	3	2	1	0
YBA0[7:0]							

Bits	Descriptions	
[31:0]	YBA0	System Memory Planar Y Base Address Word align address, ignore the bits [1:0]

System Memory Planar U Base Address Register

Register	Address	R/W	Description	Reset Value
UBA0	VIN_BA+0x84	R/W	System Memory Planar U Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
UBA0[31:24]							
23	22	21	20	19	18	17	16
UBA0[23:16]							
15	14	13	12	11	10	9	8
UBA0[15:8]							
7	6	5	4	3	2	1	0
UBA0[7:0]							

Bits	Descriptions	
[31:0]	UBA0	System Memory Planar U Base Address Word align address, ignore the bits [1:0]

System Memory Planar V Base Address Register

Register	Address	R/W	Description	Reset Value
VBA0	VIN_BA+0x88	R/W	System Memory Planar V Base Address Register	0x0000_0000

31	30	29	28	27	26	25	24
VBA0[31:24]							
23	22	21	20	19	18	17	16
VBA0[23:16]							
15	14	13	12	11	10	9	8
VBA0[15:8]							
7	6	5	4	3	2	1	0
VBA0[7:0]							

Bits	Descriptions	
[31:0]	VBA0	System Memory Planar V Base Address Word align address, ignore the bits [1:0]

6.10 Display Interface Controller (VPOST)

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).

6.10.1 Overview and 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/stripe mode
- Support NTSC/PAL interlace & non-interlace system
- LCD Timing Setting Method
  - Resolution can up to 1024x1024
  - 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

6.10.2 VPOST Controller Interface

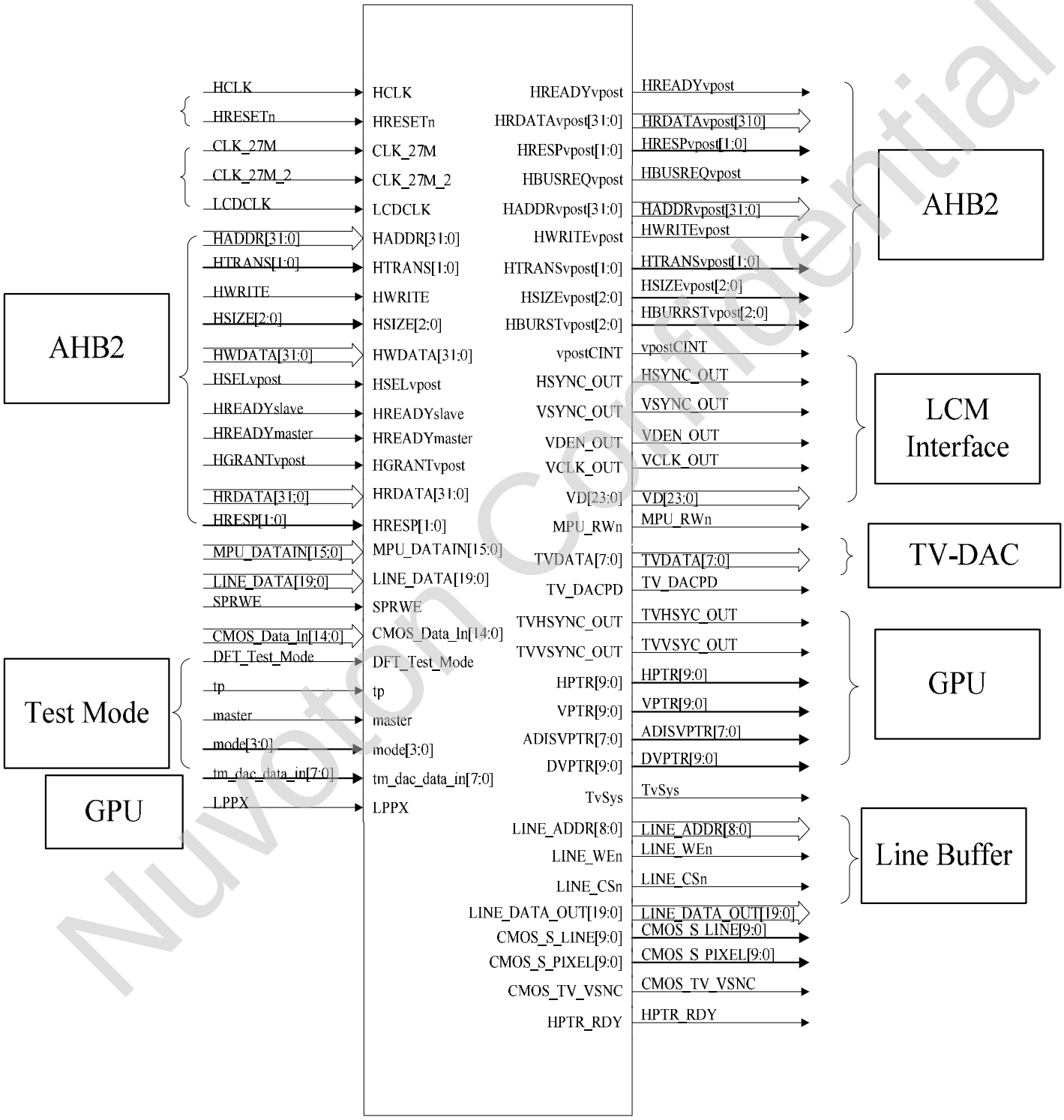


Table NO,:1110-0001-08-A

LCD device / Pin name	Sync-type High Color TFT LCD (24 bit data bus)	Sync-type TFT LCD (8 bit data bus)	MPU-type LCD	
			80 Mode	68 Mode
LVSYNC(GPD10)	VSYNC	VSYNC	RD	EN
LHSYNC(GPD9)	HSYNC	HSYNC	WR	RW
LPCLK(GPB15)	PCLK	PCLK	CS	CS
LVDE(GPD11)	DE	DE	RS	RS
LVDATA[23:0]	DATA[23:0]	DATA[7:0]	DATA[23:0]	DATA[23:0]

Figure 6.10-1 VPOST Controller Interface Diagram

LCD DATA	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	GPB 12	GPB 11	GPB 10	GPB 9	GPB 8	GPB 7	GPE 1	GPE 0	GPC 15	GPC 14	GPC 13	GPC 12	GPC 11	GPC 10	GPC 9	GPC 8	GPC 7	GPC 6	GPC 5	GPC 4	GPC 3	GPC 2	GPC 1	GPC 0
8-bit mode																	D7	D6	D5	D4	D3	D2	D1	D0
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: How to set the I/O pin as LCD related pin out, please define these control registers.

- **GPBPINFUN(0xB000_0084)**
- **GPCPINFUN(0xB000_0088)**
- **GPDPINFUN(0xB000_008c)**
- **GPEPINFUN(0xB000_0090)**

Figure 6.10-2 VPOST LCD Data Output Pin Definition

6.10.3 VPOST Controller Block Diagram

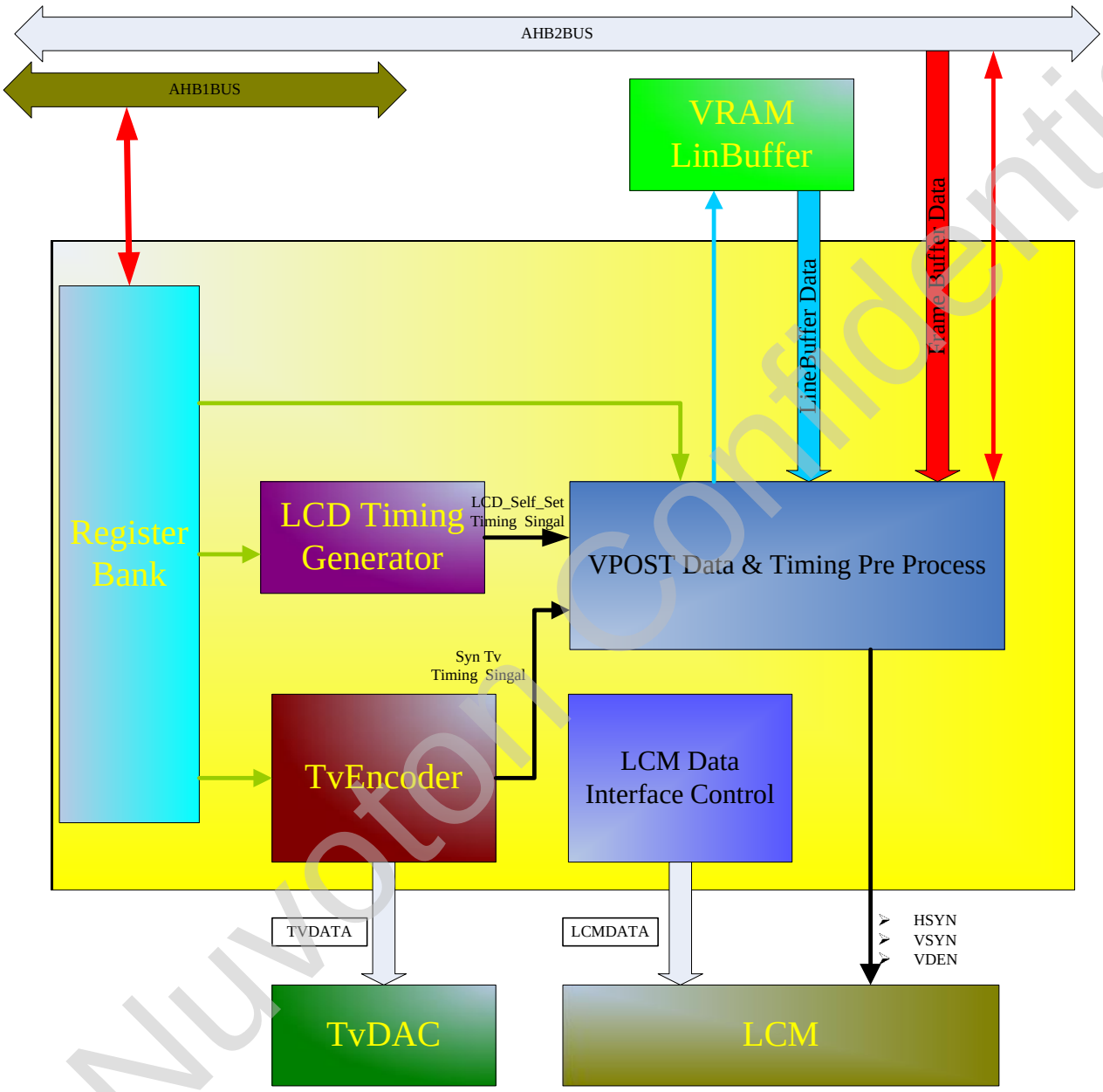


Figure 6.10-3 VPOST Controller Block Diagram

6.10.4 VPOST Controller Functional Description

- TVEncoder
  - Generate TV NTSC/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 & vsyn signal for TFT LCM requirement
  - To generate MPU control signal
- LCM Data Interface Control
  - Transfer the CRTC data and Frame Buffer Data to fit different types of LCD device.

6.10.5 VPOST Controller Control 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_0100
MPUCMD	0x20	R/W	MPU-type LCD Command Register	0x0000_0000
MPUTS	0x24	R/W	MPU Type timing control	0x0101_0101
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	0x0000_0000
OSD_1BEP	0x34	R/W	OSD Bar End Position	0x0001_0001
OSD_BO	0x38	R/W	OSD Bar Offset	0x0001_0001
CBAR	0x3C	R/W	Color Burst Active Region	0x006E_0050
TVCtl	0x40	R/W	TvControl Register	0x0000_0310
TVOUT_FLT	0x44	R/W	TV Output Filter Select Register	0x0000_001A
TVOUT_ADJ	0x48	R/W	TV Output Active Adjust Register	0x0000_0000
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_1299
CBACtl	0x58	R/W	Color Burst Amplitude Control Register	0x251A_1004
OSD_ADDR	0x5C	R/W	OSD Frame Buffer Start Address	0x0000_0000
TV_FFFSET1	0x60	R/W	TV Flick Free Filter Setting 1	0x4200_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 2	0x0001_1000

LINE_STRIPE	0x70	R/W	Line Stripe Offset	0x0000_0000
RGBin	0x74	R/W	RGB 888 Data Input for RGB2YCbCr equation	0x0000_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
Reserved	0x84	R/W	Reserved	0x0000_0000
LBTestCtl	0x88	R	Line Buffer Test Control	0x0000_0000
Reserved	0x8C	R/W	Reserved	0x0101_0000
OSD_TC_MASK	0x90	R/W	OSD Transparent Mask Control	0x0000_0000
OSD_CONT_ALPHA	0x94	R/W	OSD Constant Alpha Setting	0x0000_0000
VA_TEST	0x98	R/W	Frame Buffer Check Sum	0x8000_0000
KPI_HS_DLY	0x9C	R/W	KPI Hsync Time Setting	0x000A_001E

6.10.6 VPOST Controller Control Registers

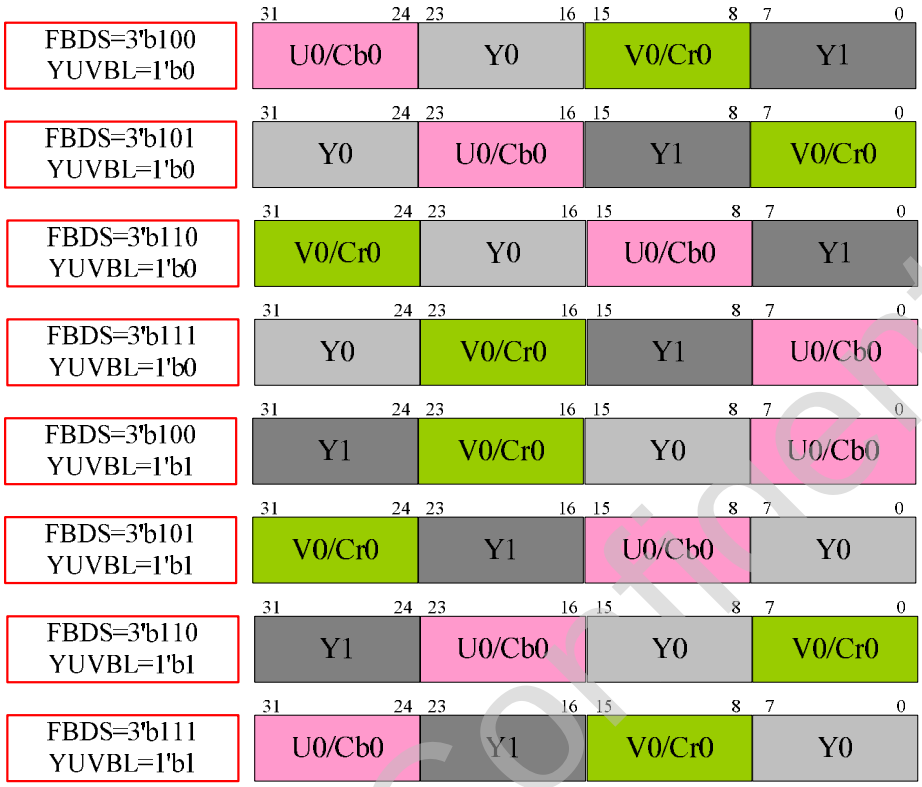
LCD Controller Control Register

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	Reserved					
23	22	21	20	19	18	17	16
Reserved		PRDB_SEL		Reserved		YUV_CLIP_EN	YUVBL
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			FBSWMODE	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
[29:22]	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
[19:18]	Reserved	Reserved
[17]	YUV_CLIP_EN	Adjust YCbCr range from (0~255) to (Y:0~235, CbCr:0~240)
[16]	YUVBL	The Y Cb(U) Cr(V) Big Endian & Little Endian selection

	(1′b0)	0:Big Endian 1:Little Endian Note: The bit was combined with FBDS for YCbCr packet data format selection
[15:5]	Reserved	Reserved
[4]	FBSWMODE	Frame Buffer Switch Mode 0: Original mode. 1: Frame buffer base address update in VSYNC region.
[3:1]	FBDS (3′b000)	Frame Buffer Data Selection 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
Odd_Field_AL				Even_Field_AL			
23	22	21	20	19	28	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 Field Total Active Line=240+ Odd_Field_AL
[27:24]	Even_Field_AL (4'h3)	CCIR656 Even Field Dummy Active Line Even Field Total Active Line=240+ Even_Field_AL
[23:15]	F1_EL (9'h109)	CCIR656 Field1(Odd Field) Ending Line
[14:9]	F1_SL (6'h4)	CCIR656 Field1(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)	Serial RGB Odd Line Selection 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
Reserved			MPUCPLEN		TVFIELD_INTEN	VINTEN	HINTEN
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			MPUCPL		TVFIELD_INT	VINT	HINT

Bits	Descriptions	
[31:21]	Reserved	Reserved
[20]	MPUCPLEN	MPU Frame Complete Enable
[19]	Reserved	Reserved
[18]	TVFIELD_INTEN	TVFIELD Interrupt Enable 0 = Disable 1 = Enable
[17]	VINTEN (1'b0)	LCD VSYNC Interrupt Enable 0 = Disable 1 = Enable
[16]	HINTEN (1'b0)	LCD HSYNC Interrupt Enable 0 = Disable 1 = Enable
[15:5]	Reserved	Reserved
[4]	MPUCPL	MPU Frame Complete

[3]	Reserved	Reserved
[2]	TVFIELD_INT	TV Field Interrupt (LCDSynTv must be 1)
[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 (1′b0)	LCD HSYNC End Interrupt 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.)

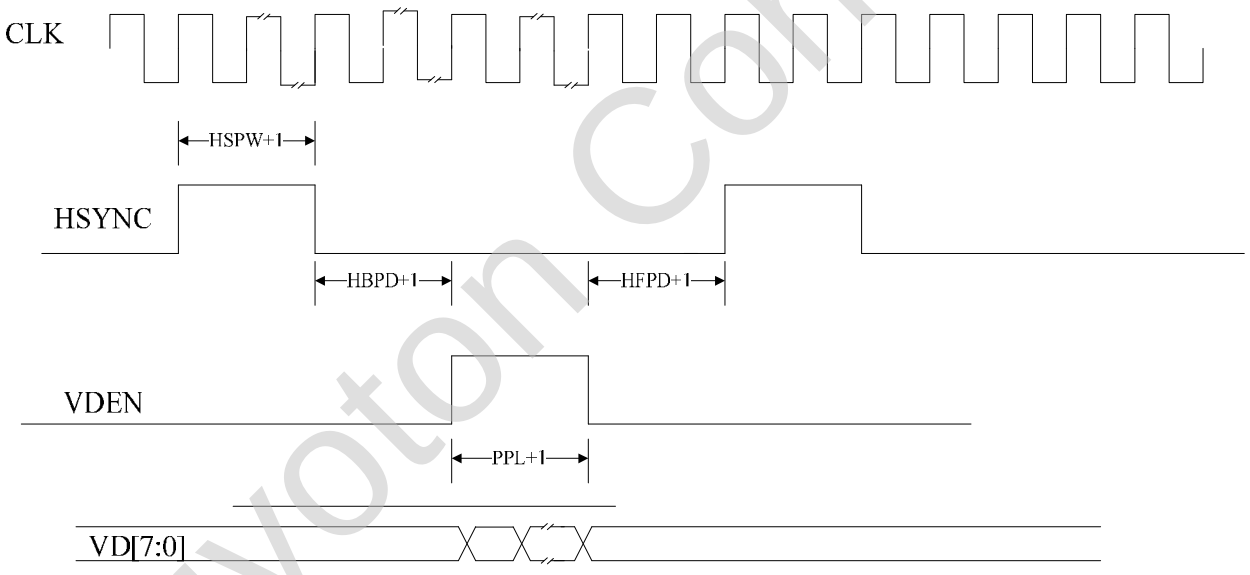
Timing Control Register 4

Register	Address	R/W	Description	Reset Value
TCON4	0x1c	R	Timing Control Register 4	0x0140_0100

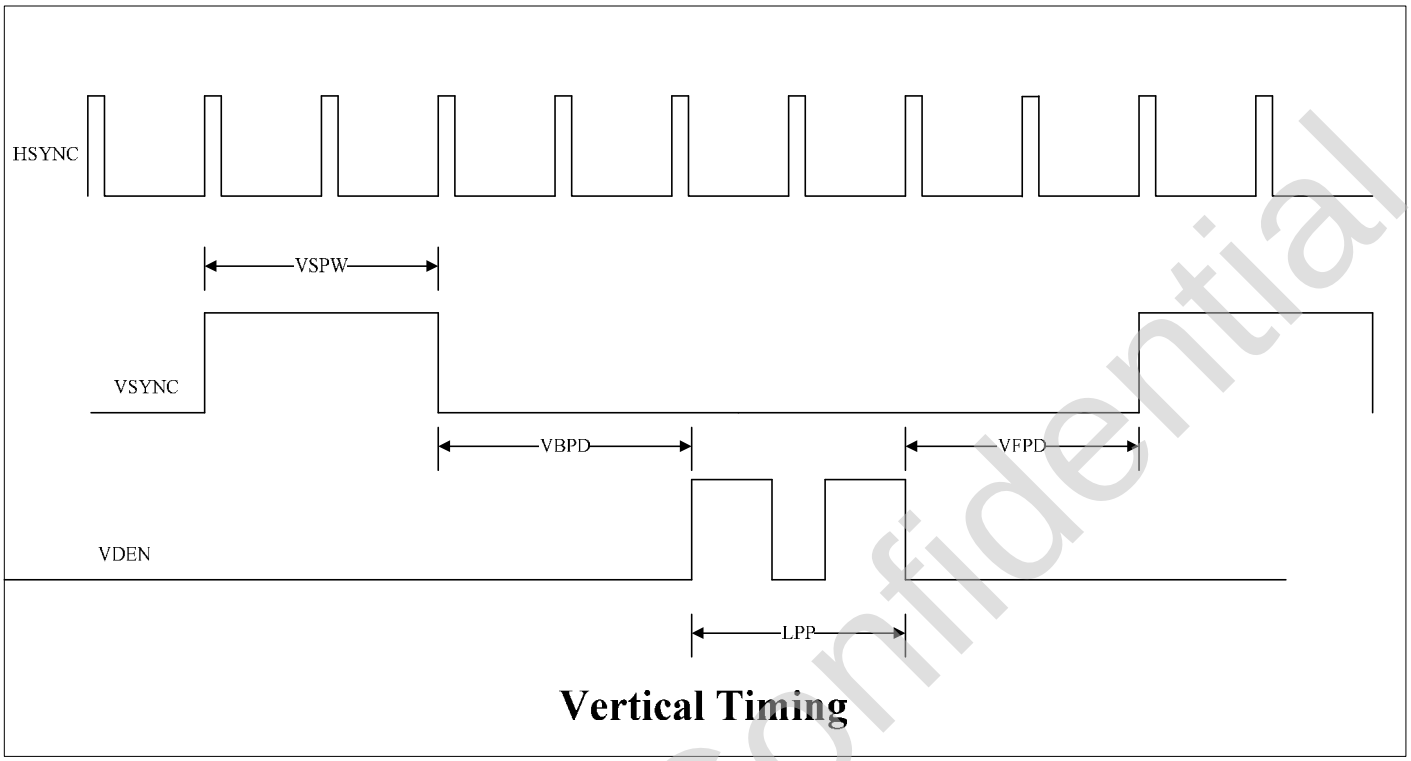
31	30	29	28	27	26	25	24
Reserved						TAPN[9:8]	
23	22	21	20	19	18	17	16
TAPN[7:0]							
15	14	13	12	11	10	9	8
MVPW							
7	6	5	4	3	2	1	0
Reserved		MPU_FMARKP	MPU_VSYNCP	VSP	HSP	DEP	PCLKP

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	TAPN	Total Active Pixel Number (for LCDSynTv = 0)
[15:4]	Reserved	Reserved
[15:8]	MVPW	MPU Vsync Pulse Width
[7:6]	Reserved	Reserved
[5]	MPU_FMARKP	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_VSYNCP	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



Vertical 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	MPU_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 VSYN/FMARK Pin Selection 1′b0: MPU VSYN/FMARK as Output Pin 1′b1: MPU VSYN/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 (LCDSynTv must = 1) 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	1st transfer	8 bit	
				2nd Transfer	8 bit	
MPU8_Mode1	4'h1	8bit	18bit	1st transfer	2 bit	
				2nd Transfer	8 bit	
				3rd Transfer	8 bit	
MPU8_Mode2	4'h2	8bit	18bit	1st transfer	6 bit	
				2nd Transfer	6 bit	
				3rd Transfer	6 bit	
MPU8_Mode3	4'h3	8bit	24bit	1st transfer	8 bit	
				2nd Transfer	8 bit	
				3rd Transfer	8 bit	
MPU9_Mode0	4'h4	9bit	18bit	1st transfer	9 bit	
				2nd Transfer	9 bit	

Mode	MPU_SI_SEL	Bus Width	Pixel Color Depth	Transfer Method		Note
MPU16_Mode0	4'h5	16bit	16bit	1st transfer	16 bit	
MPU16_Mode1	4'h6	16bit	18bit	1st transfer	16 bit	
				2nd Transfer	2 bit	
MPU16_Mode2	4'h7	16bit	18bit	1st transfer	2 bit	
				2nd Transfer	16 bit	
MPU16_Mode3	4'h8	16bit	24bit	1st transfer	16 bit	
				2nd Transfer	8 bit	
MPU18_Mode0	4'h9	18bit	18bit	1st transfer	18 bit	
MPU18_Mode1	4'ha	18bit	24bit	1st transfer	18 bit	
				2nd Transfer	6 bit	
MPU24_Mode0	4'hb	24bit	24bit	1st 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	0x0101_0101

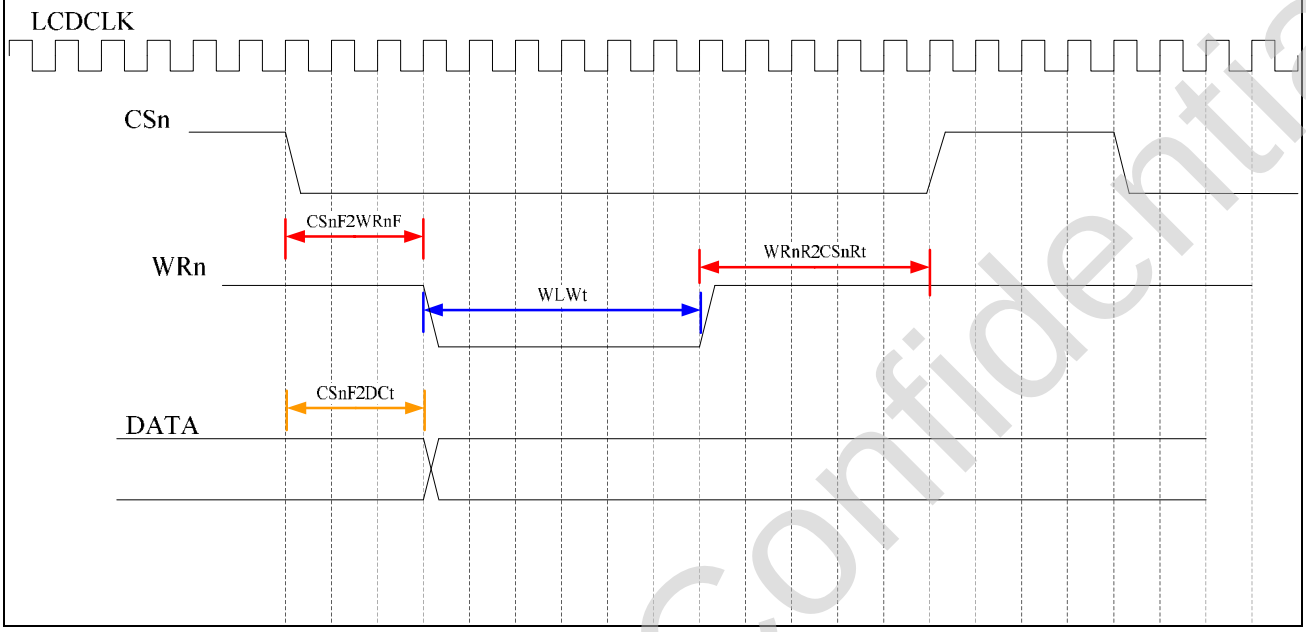
31	30	29	28	27	26	25	24
CSnF2DCt							
23	22	21	20	19	18	17	16
WRnR2CSnRt							
15	14	13	12	11	10	9	8
WRnLWt							
7	6	5	4	3	2	1	0
CSnF2WRnFt							

Bits	Descriptions	
[31:24]	CSnF2DCt	CSn fall edge to Data change clock counter (Ref Value: 1)
[23:16]	WRnR2CSnRt	WRn rising edge to CSn rising clock counter (Ref Value: 1)
[15:8]	WRnLWt	WR Low pulse clock counter (Ref Value: 1)
[7:0]	CSnF2WRnFt	Csn fall edge To WR falling edge clock counter (Ref Value: 1)

(Note: CSnF2WRnFt must be larger than or equal to CSnF2DCt)

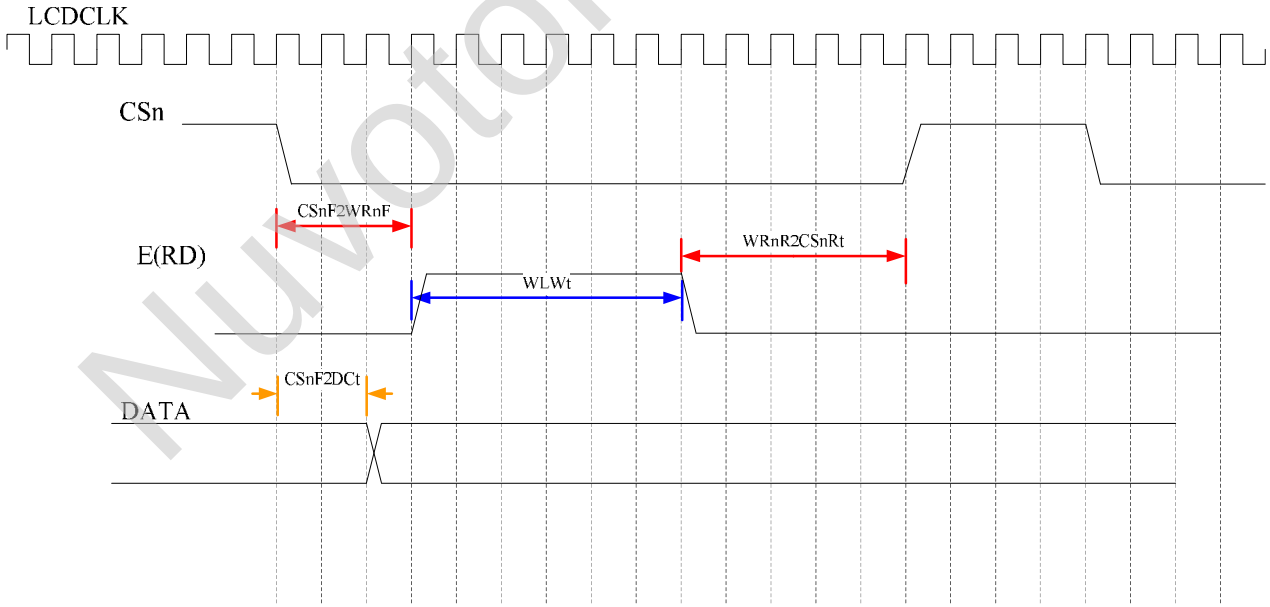
CF2WFt:Cs fall edge to WR falling edge clock count

80 Mode Write Data Timing WaveForm



CF2WFt:Cs fall edge to WR falling edge clock count

68 Mode Write Data Timing WaveForm



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 RGB555 & RGB565 & YUV422 format RGB555: R5=OSD_TC[14:10] G5=OSD_TC[9:5] B5=OSD_TC[4:0] 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]

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
Reserved						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	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						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 1st Bar End Position on the background picture	0x0001_0001

31	30	29	28	27	26	25	24
Reserved						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]							

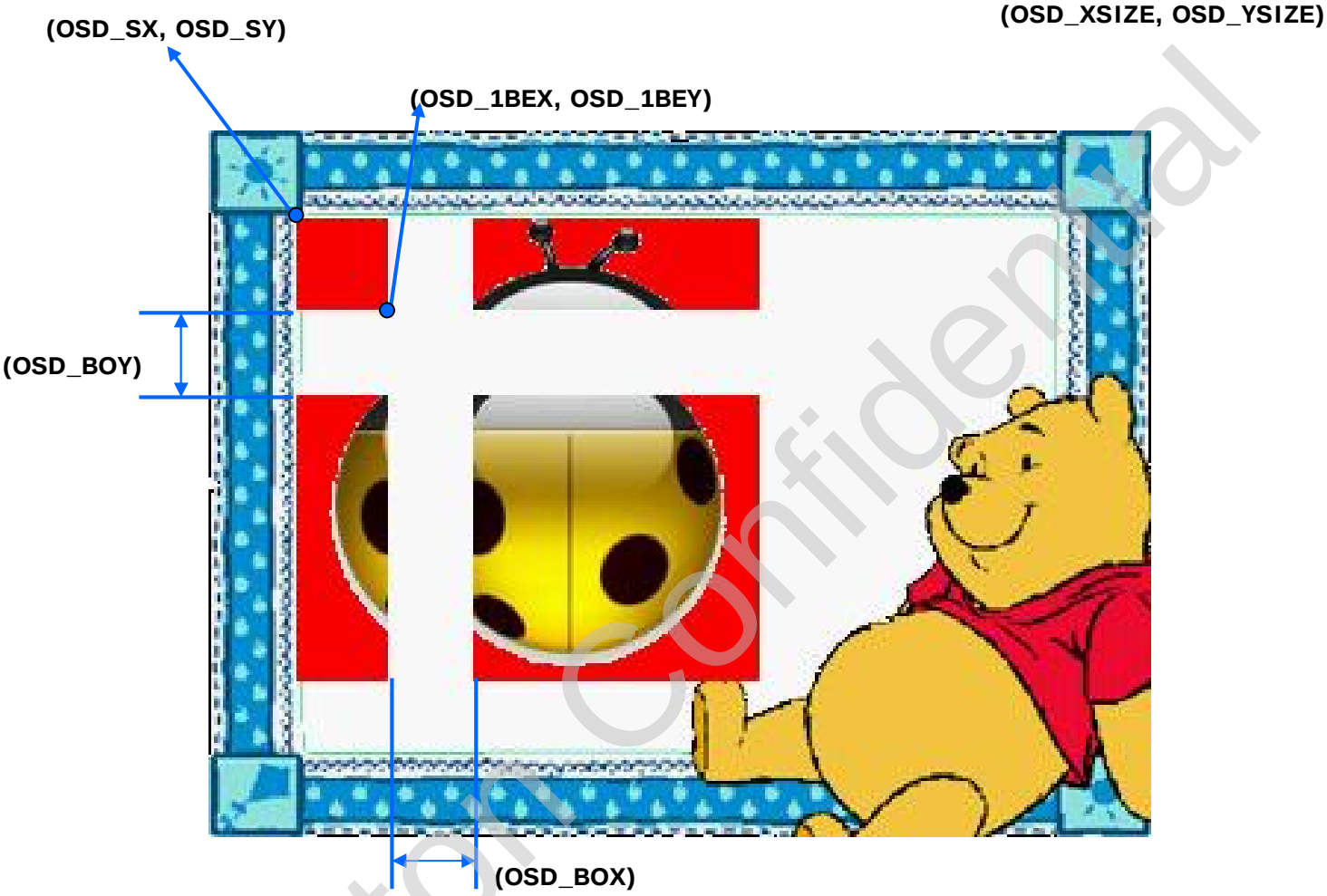
Bits	Descriptions	
[31:26]	Reserved	Reserved
[23:16]	OSD_1BEY	OSD Vertical 1st Bar End Position (Line) on the background picture (OSD_SY <= OSD_1BEY<= OSD_SY+OSD_VSIZE)
[15:10]	Reserved	Reserved
[9:0]	OSD_1BEX	OSD Horizontal 1st Bar End Position (Pixel) on the background picture (OSD_SX <= OSD_1BEX<=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

OSD Bar Offset Setting Register

Register	Address	R / W	Description	Reset Value
OSD_BO	0x38	R/W	OSD Bar Offset	0x0001_0001

31	30	29	28	27	26	25	24
Reserved						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) (Must >= 1)
[15:10]	Reserved	Reserved
[9:0]	OSD_BOX	OSD Horizontal Bar Offset (Pixel) (Must >= 1)



Color Burst Active Region Control Register

Register	Address	R/W	Description	Reset Value
CBAR_CTL	0x3c	R/W	Color Burst Active Region Control	0x006E_0050

31	30	29	28	27	26	25	24
VLCBAR		Reserved	EQ6SEL	Reserved		HCBEPC[9:8]	
23	22	21	20	19	18	17	16
HCBEPC[7:0]							
15	14	13	12	11	10	9	8
Reserved						HCBBPC[9:8]	
7	6	5	4	3	2	1	0
HCBBPC[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]	HCBEPC	Horizontal Color Burst End Pixel Counter for non-interlace PAL mode (Ref Value 10′h6E)
[15:10]	Reserved	Reserved

[9:0]	HCBBPC	Horizontal Color Burst Begin Pixel Counter for non-interlace PAL mode (Ref Value 10′h50)

TvControl Register

Register	Address	R/W	Description	Reset Value
TVCtl	0x40	R/W	TvControl Register	0x0000_0310

31	30	29	28	27	26	25	24
TvField	Reserved				TvFFFE	Reserved	
23	22	21	20	19	18	17	16
NTSC_TYPE		PAL_TYPE		Reserved		PAL288	TvMSel
15	14	13	12	11	10	9	8
FBSIZE		Reserved		LCDSrc		TvSrc	
7	6	5	4	3	2	1	0
Reserved	TvLBSA	TVCLKINV	TvDac	TvInter	TvSys	TvColor	TvSleep

Bits	Descriptions	
[31]	TvField (1'b0)	Tv Field Status (only read) 1'b0 = Even Field 1'b1 = Odd Field
[30:27]	Reserved	Reserved
[26]	TvFFFE	TV Flicker Free Filter Enable 1'b0: Disable 1'b1: Enable
[25:24]	Reserved	Reserved
[23:22]	NTSC_TYPE (2'h0)	NTSC Type Selection 00: NTSC
[21:20]	PAL_TYPE (2'h0)	PAL Type Selection 00: PAL
[19:18]	Reserved	Reserved
[17]	PAL288	Support 288-line mode (source pic size must be 640x576 or 720x576)
[16]	TvMSel (1'h1)	TV Color Modulation Method 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]	Reserved	Reserved
[11:10]	LCDSrc (2'h0)	LCD Source Selection 00 = Reserved 01 = Frame Buffer 10 = Register Setting Color 11 = Internal Color Bar
[9:8]	TvSrc (2'h3)	Tv Source Selection 00 = Reserved 01 = Frame Buffer 10 = Register Setting Color 11= Internal Color Bar
[7]	Noninter_Type (1'b0)	Non-interlace type Selection 0 = 263 lines 1 = 262 lines
[6]	TvLBSA (1'h0)	Tv Line Buffer Scaling Algorithm(320->640) 0 = Duplicate 1 = Interpolation
[5]	TVCLKINV	TVDAC Input Clock Inverse
[4]	TvDac (1'b1)	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	TV System Selection

	(1′b0)	0 = NTSC 1 = PAL
[1]	TvColor (1′b0)	TV Color Selection Color / Black 0 = Color 1 = Black
[0]	TvSleep (1′b0)	Digital TV Encoder Enable / Disable 1 =Digital TV Timing Enable 0 =Digital TV Timing Disable Note: The bit just controls the Digital TV Timing

TV Output Filter Register

Register	Address	R/W	Description	Reset Value
TVOUT_FLT	0x44	R/W	TV Output Filter Register	0x0000_001A

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	YLPF_SEL	Reserved	UVLPF_SEL	YUP_SEL		UVUP_SEL	

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6]	YLPF_SEL	Luma Low-pass Filter Selection (ref. 1'b0) 0 = Disable 1 = 9-tap
[5]	Reserved	Reserved
[4]	UVLPF_SEL	Chroma Low-pass Filter Selection (ref. 1'b1) 0 = Disable 1 = 9-tap
[3:2]	YUP_SEL	Luma Upsample Filter Selection (ref. 2'b10) 00 = Disable 01 = 2-tap 10 = 3-tap 11 = 7-tap
[2:1]	UVUP_SEL	Luma Upsample Filter Selection (ref. 2'b10) 00 = Disable 01 = 2-tap

		10 = 3-tap 11 = 7-tap

TV Output Adjust Register

Register	Address	R / W	Description	Reset Value
TVOUT_ADJ	0x48	R/W	TV Output Active Adjust Register	0x0000_0000

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

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28:24]	VER_ACTADJ	TV Vertical Output Active Position Adjust
[23:22]	Reserved	Reserved
[21:16]	HOR_ACTADJ	TV Horizontal Output Active Position Adjust
[15:0]	Reserved	Reserved

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_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	
[31:25]	Reserved	Reserved
[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_EN = 1 PRDB_SEL = 1	User function with 18-bits LCM	VD_SWAP_EN = 1 PRDB_SEL = 0	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

TV Display Control Register

Register	Address	R/W	Description	Reset Value
TvDisCtl	0x54	R/W	TV Display Start Control Register	0x10F0_1299

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

Bits	Descriptions	
[31:24]	FFRHS (8'h10)	Line First Request Horizontal Start Pixel
[23:16]	LCDHB (8'hf0)	LCD H blank setting for Sync TV Display
[15:8]	TVDVS (8'h12)	TV Display Start Line Register
[7:0]	TVDHS (8'h99)	TV Display Start Pixel Register

Color Burst Amplitude Control Register

Register	Address	R/W	Description	Reset Value
CBACtl	0x58	R/W	Color Burst Amplitude Control	0x251A_1004

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-intelace PAL mode (Ref Value 6′h25)
[23:22]	Reserved	Reserved
[21:16]	CBA_CB3	Color Burst Amplitude Adjust factor3 in non-interlace PAL mode (Ref Value 6′h1A)
[15:14]	Reserved	Reserved
[13:8]	CBA_CB2	Color Burst Amplitude Adjust factor 2 in non-interlace PAL mode (Ref Value 6′h10)
[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

TV Flick Free Filter Set1 Register

Register	Address	R/W	Description	Reset Value
TV_FFFSET1	0x60	R/W	TV Flick Free Filter Setting 1	0x4200_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
W22							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	W22	W22 parameter setting

Weighting Coefficient Table Reference: These registers are located on the TV_FFFSET1 & TV_FFFSET2 register

W11	W12	W13
W21	W22	W23
W31	W32	W33

TV Contrast Adjust Control Register

Register	Address	R / W	Description	Reset Value
TvContrast	0x64	R/W	Tv contrast adjust setting register	0x0080_8080

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Cr_contrast	Cr component contrast adjust
[15:8]	Cb_contrast	Cb component contrast adjust
[7:0]	Y_contrast	Y component contrast adjust

TV Bright Adjust Control Register

Register	Address	R/W	Description	Reset Value
TvBright	0x68	R/W	Tv Bright adjust setting register	0x0000_0000

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	Cr_gain	Cr component bright adjust
[15:8]	Cb_gain	Cb component bright adjust
[7:0]	Y_bright	Y component bright adjust

Y Adjust Equation

Y_Adj=(Y-16)*Y_Contrast+16+Y_Bright

Y_Contrast is bit7 is integer, bit6~0 fix point

Y_Bright is signed integer (-127~127)

Cb Adjust Equation

Cb_Adj=(Cb-128)*Cb_Contrast+128+Cb_Bright

Cb_Contast is bit7 is integer, bit6~0 fix point

Cb_Bright is signed integer (-127~127)

Cr Adjust Equation

Cr_Adj=(Cr-128)*Cr_Contrast+128+Cr_Bright

Cr_Contast is bit7 is integer, bit6~0 fix point

Cr_Bright is signed integer (-127~127)

TV Flick Free Filter Set2 Register

Register	Address	R/W	Description	Reset Value
TV_FFFSET2	0x6C	R/W	TV Flick Free Filter Setting 2	0x0001_1000

31	30	29	28	27	26	25	24
W33				W23			
23	22	21	20	19	18	17	16
W13				W32			
15	14	13	12	11	10	9	8
W12				W31			
7	6	5	4	3	2	1	0
W21				W11			

Bits	Descriptions	
[31:28]	W33	W33 weighting setting
[27:24]	W23	W23 weighting setting
[23:20]	W13	W13 weighting setting
[19:16]	W32	W32 weighting setting
[15:12]	W12	W12 weighting setting
[11:8]	W31	W31 weighting setting
[7:4]	W21	W21 weighting setting
[3:0]	W11	W11 weighting setting

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_LSL[15:8]							
23	22	21	20	19	18	17	16
OSD_LSL[7:0]							
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	
[31:16]	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	
[15:11]	Reserved	Reserved
[10:8]	BistFailR	
[7:3]	Reserved	Reserved
[2:0]	BistModeR	

OSD Transparent mask control

Register	Address	R/W	Description	Reset Value
OSD_TC_MASK	0x90	R/W	OSD Transparent mask control	0x0000_0000

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	OSD_MASK	OSD Transparent Mask Setting for RGB555 & RGB565 & YUV422 format (OSD_CONALPHA_EN must be 0) RGB555: R5=OSD_MASK[14:10] G5=OSD_MASK[9:5] B5=OSD_MASK[4:0] RGB565: R5=OSD_MASK[15:11] G6=OSD_MASK[10:5] B5=OSD_MASK[4:0] YUV: Y=OSD_MASK[23:16]; Cb=OSD_MASK[15:8] Cr=OSD_MASK[7:0] RGB888:

		R=OSD_MASK[23:16]; G=OSD_MASK[15:8] B=OSD_MASK[7:0]

OSD Constant Alpha Setting

Register	Address	R/W	Description	Reset Value
OSD_CONT_ALPHA	0x94	R/W	OSD Constant Alpha Setting	0x0000_0000

31	30	29	28	27	26	25	24
OSD_CONALPHA_EN	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
OSD_CONALPHA							

Bits	Descriptions	
[31]	OSD_CONALPHA_EN	OSD Constant Enable (when OSD_CONALPHA_EN=1, the OSD_TPEN will be disable)
[30:8]	Reserved	Reserved
[7:0]	OSD_CONALPHA	OSD Constant Value Setting FF: opaque 00: transparent

VA_TEST

Register	Address	R/W	Description	Reset Value
VA_TEST	0x98	R/W	Frame Buffer Check Sum	0x8000_0000

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

Bits	Descriptions	
[31]	CHECK_START	Check Sum Start Control (default 1) 1′b1: HW will enable checksum adder from next Vsync signal. 1′b0: HW will disable checksum adder from next Vsync signal.
[30:16]	Reserved	Reserved
[15:0]	CHECK_SUM	Frame Buffer Check Sum (Read only): HW will add frame buffer data by every frame. CHECK_SUM[15:0] = DATA[31:16] + DATA[15:0] + CHECK_SUM[15:0] The overflow of 16-bit CHECK_SUM will be clipped.

KPI_HS_DLY

Register	Address	R/W	Description	Reset Value
KPI_HS_DLY	0x9C	R/W	LCD share Hsync Bus to KPI Time Setting	0x000A_001E

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

Bits	Descriptions	
[31]	KPI_REF_SYNC	Set to 1 will use larger timing range to share LCD bus: 0: Switch to KPI only in HSPW, 1: Switch to KPI in HFPD+HSPW+HBPD Note: Set to 1 may cause some LCDs display incorrect.
[30:27]	Reserved	Reserved
[26:16]	KPI_HFD	Hsync Front Delay time for KPI: Count by LCDCLK cycle
[15:11]	Reserved	Reserved
[10:0]	KPI_HBD	Hsync Back Delay time for KPI: Count by LCDCLK cycle

6.11 Sound Processing Unit (SPU)

6.11.1 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.

6.11.2 Features

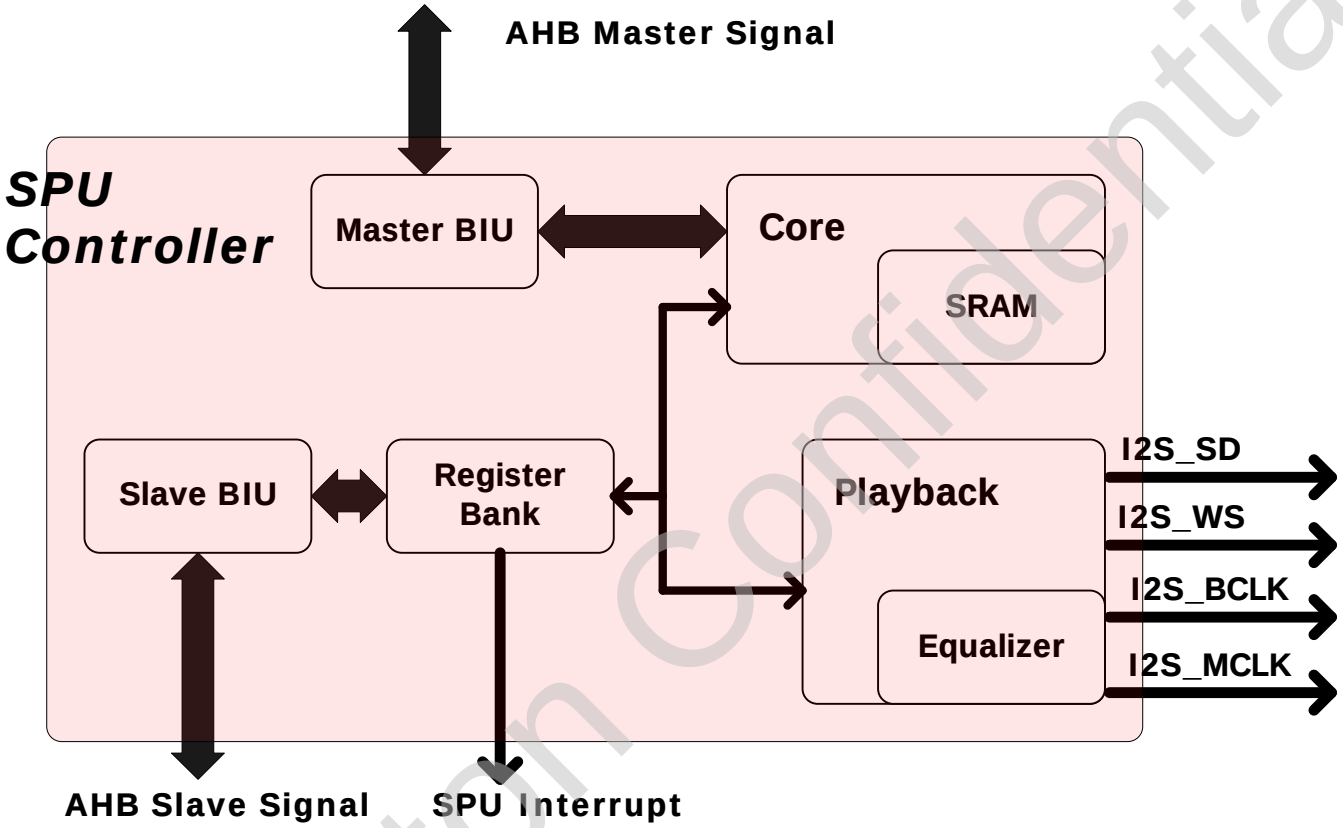
- Left / Right 16-bit stereo output to DAC with 6-bit master Volume Control
- I2S output interface
- Support 32 channels sources
- Support E-MDPCM (4bit) / PCM16 / LP8 data type with event
- Special code in source bitstream for loop playing, silence, end and user event.
- Support raw PCM16 mono/stereo 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)

6.11.3 SPU Block Diagram

The block diagram of SPU is shown as following. The SPU controller uses AHB Slave interface to control the register, which can set up the parameters with each channel. However, the AHB master interface is used to connect the system memory with the SRAM in the CORE block. The information crossing this interface is the data for playback and is the source for the core black to generate the samples. By the way, the SRAM is not only saved the playback source, but saved each channel settings. In the detailed description, we use AHB slave interface to set the channel parameters, and these parameter will be saved to the SRAM. Then, if the SPU has been started, the state machines in the core block will auto generate the samples with the channel parameters and use the playback block to play the audio. And the output frequency is controlled by the DAC_CLK. The equation is that the "output frequency = DAC_CLK / 256". The ratio 256 is depended on the delta-sigma DAC IP.

Because the SPU has 32 channels, we must share the registers for each channel. Therefore, if we want to know one channel situation (states), we must use the "Load" command. First, we set the Channel Number with which we want to know. Second, set the Load Command. Then, these channel parameters will be saved to register bank from the SRAM. Therefore, we can get the parameters from the register bank.

One thing is importance. The SRAM is used to saving the channel parameters. And the initial value about the SRAM is unknown. Therefore, if we don't set all the parameters with each channel in the first, the channel parameters will be random. It is because the SRAM initial value is random.



Pin descriptions:

I2S_SD : I2S Data out

I2S_WS : I2S Word Select

I2S_BCLK : I2S Bit Clock

I2S_MCLK : I2S Master Clock Output

SPU_INT : SPU interrupt signal

6.11.4 SPU Control 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/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
Clk_PAR	SPU_BA + 0x08	R/W	Clock Parameter Register	0x0000_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	0x0000_0000
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
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
DAC_CTRL	SPU_BA + 0x50	R/W	DAC Control Interface Command	0x0000_0000

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	RampUpMod			RESERVED	I2S_JUSTIFIED	RESERVED	I2S_EN
7	6	5	4	3	2	1	0
RESERVED			SRAM_CLK_GATED	RESERVED	SPU_END_FLAG	SPU_END_CTRL	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.
[14:12]	RampUpMod	Ramp Up Mode Select 3'd0 : 8 audio output samples 3'd1 : 32 audio output samples 3'd2 : 128 audio output samples 3'd3 : 512 audio output samples 3'd4 : 2048 audio output samples 3'd5 : 8192 audio output samples

		3′d6,7 : Disable If we play out the audio or open the SPU to playback the audio suddenly, there may be some pop sound. Therefore, it must have a ramp up time to decrease the influence about the pop sound. The "RampUpMod " is to select the ramp up time. This function is only useful when playing the audio from all channel pause situation or all channel disable situation.
[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
[2]	SPU_END_CTRL	SPU End Function Control If wanting to close the SPU function and keeping the register information, you can set this bit to high. And the SPU will stop all block when it finishes the current sample generated. Then, if the bit is set to low, the SPU state machine will start again. 1′b1 : STOP the SPU, when Finish the current sample generated. 1′b0 : Re-start the SPU process
[1]	SPU_END_FLAG	SPU End Function Flag If setting the SPU_END_CTRL high and the State Machine is stop, this flag will be set to high. It means the SPU has already been stop. 1′b1 : Has been STOP (SPU_END_CTRL process finish)

		1′b0 : SPU process is going
[0]	SPU_EN	SPU Core Enable • 0 = Disable • 1 = Enable NOTE: All xxxx.

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	Reserved	Reserved		Reserved			
7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved		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
[23:0]	Reserved	Reserved

Clock Parameter Register

Register	Address	R/W	Description	Reset Value
Clk_PAR	SPU_BA + 0x08	R/W	Clock Parameter Register	0x2000_0000

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

Bits	Descriptions	
[31]	ClkRst	Clock Divider Module Reset 1 : Reset 0 : Normal
[30]	PllSelMod	Select the SPU PLL Clock comes from. 1 : PLL Clock from DAC PLL OUT 0 : PLL Clock from Internal PLL. If setting the clock from DAC, we should output a 12MHz / 13.5Mhz MCLK to the DAC. And it will generate the PLL OUT for us, which frequency is controlled by the DAC register. Then, we will generate the BCLK or WS depended on the PLL OUT clock. On the other hand, if setting the clock from Internal PLL, the BCLK, MCLK, WS are controlled by the Clock Parameter Register.
[29]	DAC_RST	DAC Reset Signal 1'b1 : Disable Reset 1'b0 : Reset
[28]	Reserved	Reserved
[23:12]	IntegerNum	Clock Divider Integer Number Part
[11:0]	FractionNum	Clock Divider Fraction Number Part

This register parameter is to control the radio between the MCLK and FPLL, which FPLL is the SPU audio input clock. And the operation method is that if you want to set the "MCLK = FPLL / 27.311(decimal number)", you can set this register as IntegerNum = 0x027 and FractionNum = 0x311. It means set the register 0x00270311. So if you set the register as 0x00099863, it means "MCLK = FPLL / 99.863". However, if the IntegerNum = 0x00, it is equal to IntegerNum = 0x01 (MCLK = FPLL / 1).

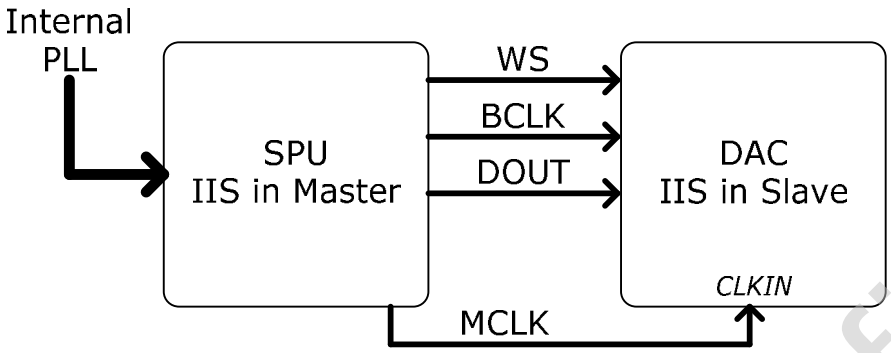


Figure 6.11-1 Setting the PllSelMod = 0

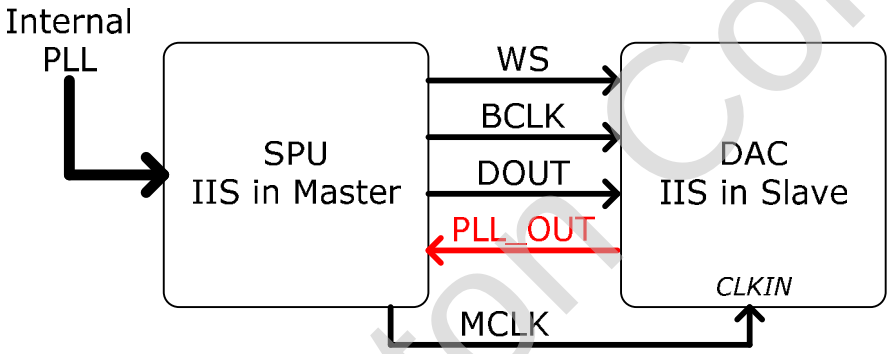


Figure 6.11-2 Setting the PllSelMod = 1

Figure 6.11-3 and Figure 6.11-4 shows the different about the PllSelMod. Because the DAC has a PLL, so this function is to choose the clock source for playing the audio come from internal PLL or DAC PLL. However, the DAC PLL needs a master clock to synchronize. Therefore, if set the PllSelMod = 1, the clock MCLK is used for synchronization by the DAC and output a PllOutClock for us.

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						VIR_I2C_IRQ_EN	VIR_I2C_IRQ_FG
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_ADDR	RESERVED	CH_FN[1:0]	

Bits	Descriptions	
[28:24]	CH_NO	Select Channel Number (5bits) Set target channel.
[23:18]	Reserved	Reserved
[17]	VIR_I2C_IRQ_EN	Virtual I2C Interrupt Enable 1'b0: Disable 1'b1: Enable
[16]	VIR_I2C_IRQ_FG	Virtual I2C Done Interrupt Flag (Write 1 Clear)
[15:13]	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, word boundary The channel source starts address.

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, word boundary When the channel current address cross this threshold address, the threshold interrupt flag will up.

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, word boundary The channel source end address. 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

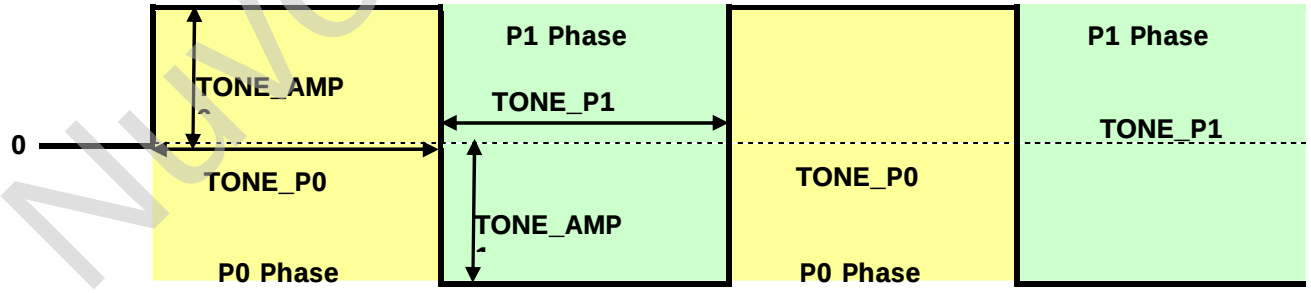
Tone Amplitude – share register with End Address

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

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]). The equation is that: Output sampling rate = input sampling rate * (x + y / 1024), which x is the integral part of DFA (DFA[12:10]) and the y is the fractional part of DFA (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. Ÿ 0 = Disable Ÿ 1 = Enable, default If disable the bit, the event flag should only be cleared by write 1. And if enable this bit, the event flag will be clear after read the event register (which is the process about load the channel parameters command).

[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 It shows the information that the playback process has played on which location (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 meets this pause address, the current channel will be auto paused. Note: This function is useful when the PA_INT_EN is enable.

DAC Control Command

Register	Address	R/W/C	Description	Reset Value
DAC_CTRL	SPU_BA + 0x50	R/W	DAC Control Interface Command	0x0000_0000

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

Bits	Descriptions	
[31]	BUSY	If this register has been written, the HW would change the command to the I2C format. It is because the DAC interface is only I2C. However, the speed of the I2C is slow. So, this bit is used to indicate the end of the I2C command. 1′b1: I2C command is not finish 1′b0: I2C command is finish
[30:24]	SCK_DIV	Control the SCK Timing Parameter The SCK frequency is (SPU_CLK frequency / (SCK_DIV * 4)). Note: Can′t be zero.
[23:17]	DEVICE_ID	The ID information which is use to read from the DAC or write to the DAC (Default DAC ID is 40H)
[16]	RW	Control this command is to read data from the DAC or write data to the DAC 1′b1 : Write to the DAC

		1′b0 : Read from the DAC
[15:8]	ADDR	The address information which is use to read from the DAC or write to the DAC
[7:0]	DATA	The data information which is use to read from the DAC or write to the DAC

DAC Control Register address:

Left Channel Analog Volume Control Register (Address: 00H; Default: 00H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:5]	Reserved	000	
[4:0]	AVOLL	0_0000	DAC Left Channel Gain Control. The value can be programmed from 00h to 1FH, stepped by -2dB 00H: max volume 1FH: analog mute

Right Channel Analog Volume Control Register (Address: 01H; Default: 00H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:5]	Reserved	000	
[4:0]	AVOLR	0_0000	DAC Right Channel Gain Control. The value can be programmed from 00h to 1FH, stepped by -2dB 00H: max volume 1FH: analog mute

PLL Output Frequency Control Register

LABEL	DESCRIPTION				
	Fin(MHz)	DIVM[7:0]	DIVN[7:0]	DP[7:0]	Fout(Hz)
DP[7:0] Address: 02H Default: 00H DIVM[7:0] Address: 0BH Default: 00H DVIN[7:0] Address: 0CH Default: 00H	12	0000_0000	0000_0000	0000_0110	8K
		0000_0010	0010_1010	0000_0010	11.025K
		0000_0000	0000_0000	0000_0010	12K
		0000_0000	0000_0000	0000_0101	16K
		0000_0010	0010_1010	0000_0001	22.05K
		0000_0000	0000_0000	0000_0001	24K
		0000_0000	0000_0000	0000_0100	32K
		0000_0010	0010_1010	0000_0000	44.1K
		0000_0000	0000_0000	0000_0000	48K
		0000_0000	0000_0000	0000_1000	96K
	12.288	0000_0000	0000_0110	0000_0110	8K

		0000_0000	0001_1001	0000_0010	11.025K
		0000_0000	0000_0110	0000_0010	12K
		0000_0000	0000_0110	0000_0101	16K
		0000_0000	0001_1001	0000_0001	22.05K
		0000_0000	0000_0110	0000_0001	24K
		0000_0000	0000_0110	0000_0100	32K
		0000_0000	0001_1001	0000_0000	44.1K
		0000_0000	0000_0110	0000_0000	48K
		0000_0000	0000_0110	0000_1000	96K

Mute and Zero Crossing Detector Register (Address: 03H; Default: 05H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7]	DMUTER	0	Left Channel Digital Part mute: 0: not mute 1: mute
[6]	DMUTER	0	Right Channel Digital Part mute: 0: not mute 1: mute
[5]	AMUTER	0	Reserved
[4]	AMUTER	0	Reserved
[3:2]	MUXSELL	01	Left Channel MUX select: 00: no signal input is selected 01: lineout signal is selected 10: analog signal is selected 11: Mixer
[1:0]	MUXSELR	01	Right Channel MUX select: 00: no signal input is selected 01: lineout signal is selected 10: analog signal is selected 11: Mixer

DAC Power Consumption Control Register (Address: 04H; Default: 01H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:3]	Reserved	0_0000	
[2:0]	RESADJ	001	Current Biasing Resistor Select: 000: Smallest Biasing Resistor 100: Medium Small Biasing Resistor 010: Medium Big Biasing Resistor 001: Biggest Biasing Resistor Other: Reserved

DAC power Down Control Register (Address: 05H; Default: FFH; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7]	BYPASSPLL	1	Bypass mode control input: 0: normal mode for PLL 1: bypass mode, PLL input goes to the PLL output frequency divider, and then to the output
[6]	PDPLL	1	PLL power down control mode: 1: Power down 0: Power on
[5]	PDBIAS	1	Reference power down control: 1: Power down 0: Power on
[4]	PDCHG	1	Pre-charge Vref Cap.: 1: Power down 0: Power on
[3]	PWDNL	1	DAC L Channel Power Down Control (analog): 1: Disables DAC L channel 0: Enable DAC L channel
[2]	PWDNR	1	DAC R Channel Power Down Control (analog): 1: Disables DAC R channel 0: Enable DAC R channel
[1]	PDPAL	1	L Headphone Power Down Control (analog): 1: Power down

			0: Power on
[0]	PDPAR	1	R Headphone Power Down Control (analog): 1: Power down 0: Power on

Analog Test Register (Address: 06H; Default: 00H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:0]	ANA_TEST	0000_0000	Analog Test Register

DAC Control Register (Address: 07H; Default: 00H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:3]	Reserved	0_0000	
[2:1]	DAC_MODE	00	DAC Stereo/Mono Control Signal Or Digital Filter Soft Reset Control: 00: Stereo mode 01: L Mono mode 10: R Mono mode 11: (L+R)/2 Mono mode
[0]	DACEN	0	0: Digital DAC Disabled 1: Digital DAC Enabled

I2S Interface Register (Address: 08H; Default: 00H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:1]	Reserved	000_0000	
[0]	SLAVE	0	I2S Interface Mode Select: 0: Master Mode 1: Slave Mode

Left Channel Digital Volume Control Register (Address: 09H; Default: 80H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:0]	DVOLL	1000_0000	DAC Left Channel Gain Control. The value can be programmed from 00h to 80H, stepped by 1/128*Vfs

			Default: 80h (Max volume)

Right Channel Digital Volume Control Register (Address: 0AH; Default: 80H; Access: R/W)

BITS	LABEL	DEFAULT	DESCRIPION
[7:0]	DVOLR	1000_0000	DAC Right Channel Gain Control. The value can be programmed from 00h to 80H, stepped by 1/128*Vfs Default: 80h (Max volume)

6.12 I2S Controller

6.12.1 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.

The following are the property of the DMA.

- **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.

6.12.2 Block diagram

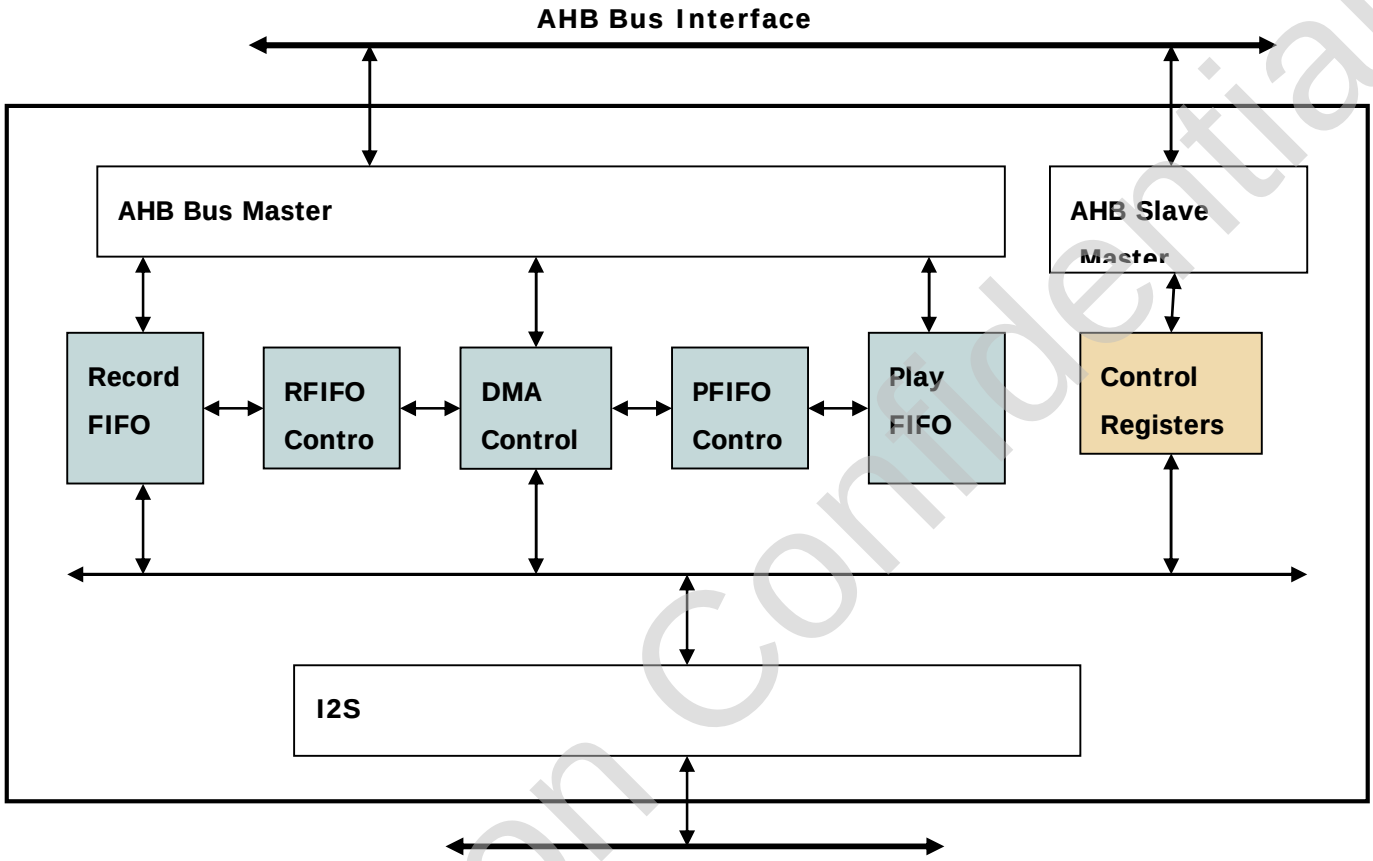


Figure 6.12-1 Block diagram of Audio Controller

6.12.3 I2S Interface

The I2S interface signals are shown as the following figure

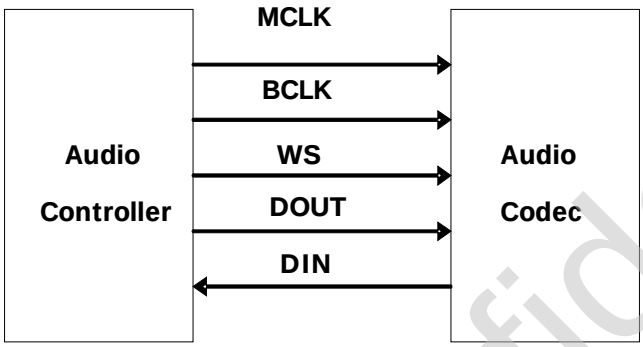


Figure 6.12-2 The interface signal of I2S

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

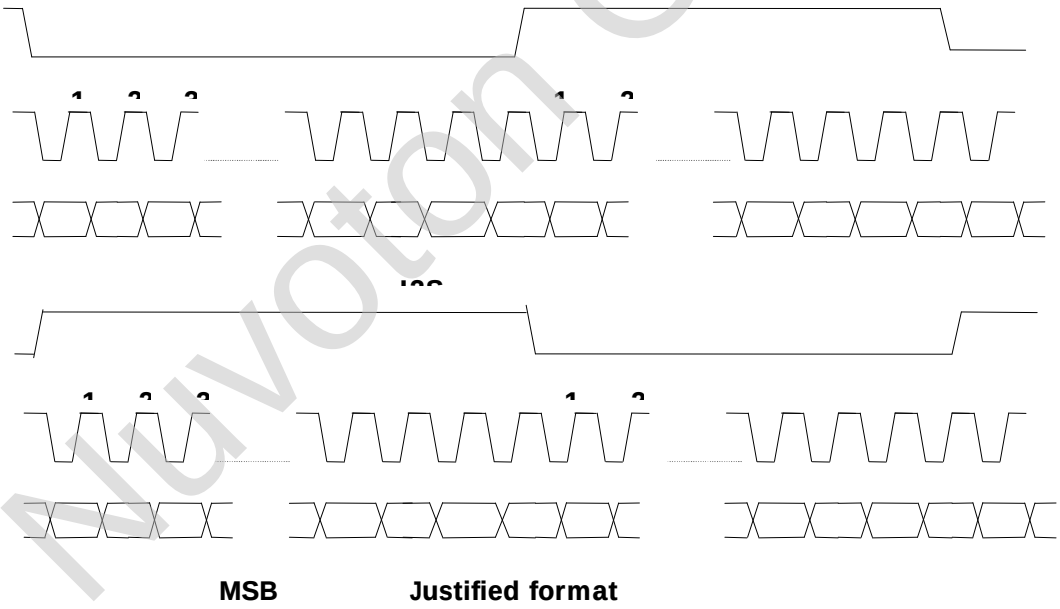


Figure 6.12-3 The format of I2S

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

6.12.4 Audio Controller 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
ACTL_PauseLength	I2S_BA + 0x30	R/W	Pause Function Length Location	0xFFFF_FFFF
ACTL_CLKDIV	I2S_BA + 0x34	R/W	Clock Divider Parameter	0x0000_0000

6.12.5 Register Details

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_PAUSE_IRQ_EN	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	
[31:23]	RESERVED	RESERVED
[22]	P_PAUSE_IRQ_EN	Playback Pause Function Interrupt Request Enable Bit • 0: not allowed to generation R_DMA_IRQ • 1: allowed to generation R_DMA_IRQ The P_PAUSE_IRQ_EN bit is read/write
[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
[8]	P_PAUSE_IRQ	Playback Pause Function IRQ Ÿ 0: not reach pause address Ÿ 1: reach pause address The Pause IRQ bit is readable , and only can be clear by write "1" to clear this bit
[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 Ÿ 0: not found DMA_COUNTER to zero Ÿ 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	Ÿ DMA_DATA zero IRQ Ÿ 0: not found DMA DATA is zero or sign change(two channel) Ÿ 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 Ÿ 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. Ÿ 0: the P_FIFO full error is not happened. Ÿ 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 Ÿ 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. Ÿ 0: the P_FIFO empty error is not happened. Ÿ 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 Ÿ 0: Playback DMA address does not reach the specific address by P_DMA_IRQ_SEL bits. Ÿ 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) Ÿ 0000: MCLK=FPLL/1 Ÿ 0001: MCLK=FPLL/2 Ÿ 0010: MCLK=FPLL/3 Ÿ 0011: MCLK=FPLL/4 Ÿ 0100: MCLK=FPLL/5 Ÿ 0101: MCLK=FPLL/6 Ÿ 0110: MCLK=FPLL/7 Ÿ 0111: MCLK=FPLL/8 Ÿ 1000: RESERVED Ÿ 1001: MCLK=FPLL/10 Ÿ 1010: RESERVED

		Ÿ 1011: MCLK=FPLL/12 Ÿ 1100: RESERVED Ÿ 1101: MCLK=FPLL/14 Ÿ 1110: RESERVED Ÿ 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 Ÿ 00: The frequency of bit clock (BCLK) is MCLK/8. Ÿ 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. Ÿ 1x: RESERVED The BCLK_SEL[1:0] bits are read/write
[5]	WS_SEL	I2S Sampling Word Selection Bit Ÿ If BCLK_SEL[1:0]=00, and WS_SEL=0, 32ws is selected, the word selection (WS) = MCLK/(8*32) = MCLK/(256) Ÿ If BCLK_SEL[1:0]=00, and WS_SEL=1, 48ws is selected, the word selection (WS) = MCLK/(8*48) = MCLK/(384) Ÿ If BCLK_SEL[1:0]=01, this bit is ignored, 32ws is selected, the word selection (WS) = MCLK/(12*32) = MCLK/(384) Ÿ (WS is sampling rate) The WS_SEL bit is read/write Example:

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 Ÿ 0: I2S MCLK output will follow the PRS [3:0] setting. Ÿ 1: I2S MCLK output will be the same with FPLL. The MCLK_SEL bit is read/write
[3]	FORMAT	I2S Format Selection bit Ÿ 0: I2S compatible format is selected. Ÿ 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.

Pause Function Length Location (ACTL_PauseLength)

Register	Address	R/W	Description	Reset Value
ACTL_PauseLength	I2S_BA + 0x30	R/W	Pause Function Length Location	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_PauseLength[31:0]	**Pause Function Length Location Information** When use the Pause Function to set which location should be stop, we can set the length in this register. And the stop address is "ACTL_PDST + ACTL_PauseLength". If the pause situation happens, the Pause_IRQ will be set to hight (Pause_IRQ_En is open). Then, clear the Pause_IRQ, the playback process will go again.

Clock Divider Parameter (ACTL_CLKDIV)

Register	Address	R/W	Description	Reset Value
ACTL_CLKDIV	I2S_BA + 0x34	R/W	Clock Divider Parameter	0x0000_0000

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

Bits	Descriptions	
[31]	ClkSelMod	MCLK Source Mode Select, 1′b0: MCLK frequency is control by register ACTL_I2SCON parameter 1′b1: MCLK frequency is control by register ClkDiv. And the parameter PRS can′t be zero and MCLK_SEL should be one which are in the register ACTL_I2SCON.
[30:24]	Reserved	Reserved
[23:12]	IntegerNum	Clock Divider Integer Number Part
[11:0]	FractionNum	Clock Divider Fraction Number Part

If MCLK source is selected from ACTL_CLKDIV, the Frequency MCLK = FPLL / (IntegerNum + FractionNum / 1000). For example, FPLL = 40MHz, IntegerNum = 0x008, and FractionNum = 0x125. The MCLK frequency is equal to 40MHz / (8.125) = 4.923 MHz.

6.13 Storage Interface Controller

The Storage Interface Controller (SIC) of W55FA95 Chip has SIC_DMA unit and SIC_FMI unit. The SIC_DMAC 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.

6.13.1 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/24bit 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.

6.13.2 Block Diagram and Card Pad Assignment

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

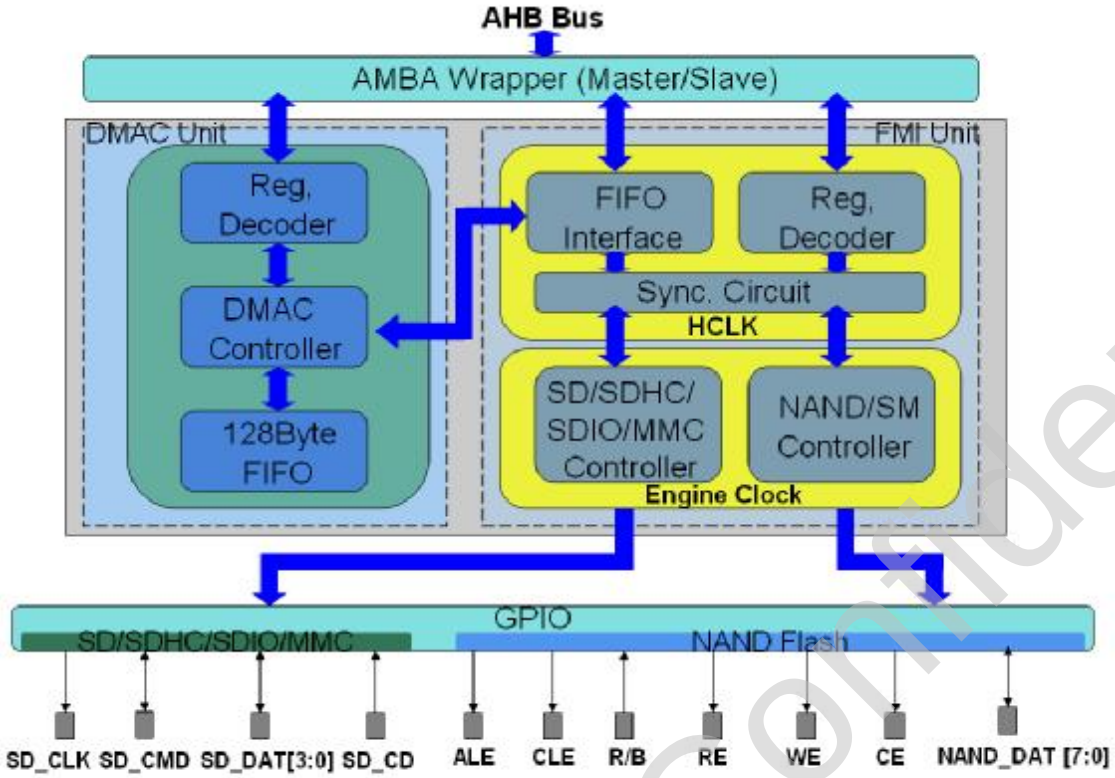


Figure 6.13-1 SIC Controller Block Diagram

	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 6.13-1 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.

Table 6.13-2 NAND/SM Card Pad Assignment

6.13.3 SIC 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	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
FMIIER	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_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
SM_PROT_ADDR0	FMI_BA + 0xE0	R/W	Smart-Media Protect region end address 0	0x0000_0000
SM_PROT_ADDR1	FMI_BA + 0xE4	R/W	Smart-Media Protect region end address 1	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
BCH_ECC_ADDR8	ECC_BA + 0x20	R	BCH error byte address 8	0x0000_0000
BCH_ECC_ADDR9	ECC_BA + 0x24	R	BCH error byte address 9	0x0000_0000
BCH_ECC_ADDR10	ECC_BA + 0x28	R	BCH error byte address 10	0x0000_0000
BCH_ECC_ADDR	ECC_BA + 0x2C	R	BCH error byte address 11	0x0000_0000

11				
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 1	0x8080_8080
BCH_ECC_DATA 2	ECC_BA + 0x68	R	BCH ECC Error Data 2	0x8080_8080
BCH_ECC_DATA 3	ECC_BA + 0x6C	R	BCH ECC Error Data 3	0x8080_8080
BCH_ECC_DATA 4	ECC_BA + 0x70	R	BCH ECC Error Data 4	0x8080_8080
BCH_ECC_DATA 5	ECC_BA + 0x74	R	BCH ECC Error Data 5	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

6.13.4 SIC DMA Controller

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.

6.13.4.1 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).

6.13.4.2 Symbol Diagram

The symbol diagram of DMA Controller is shown as following.

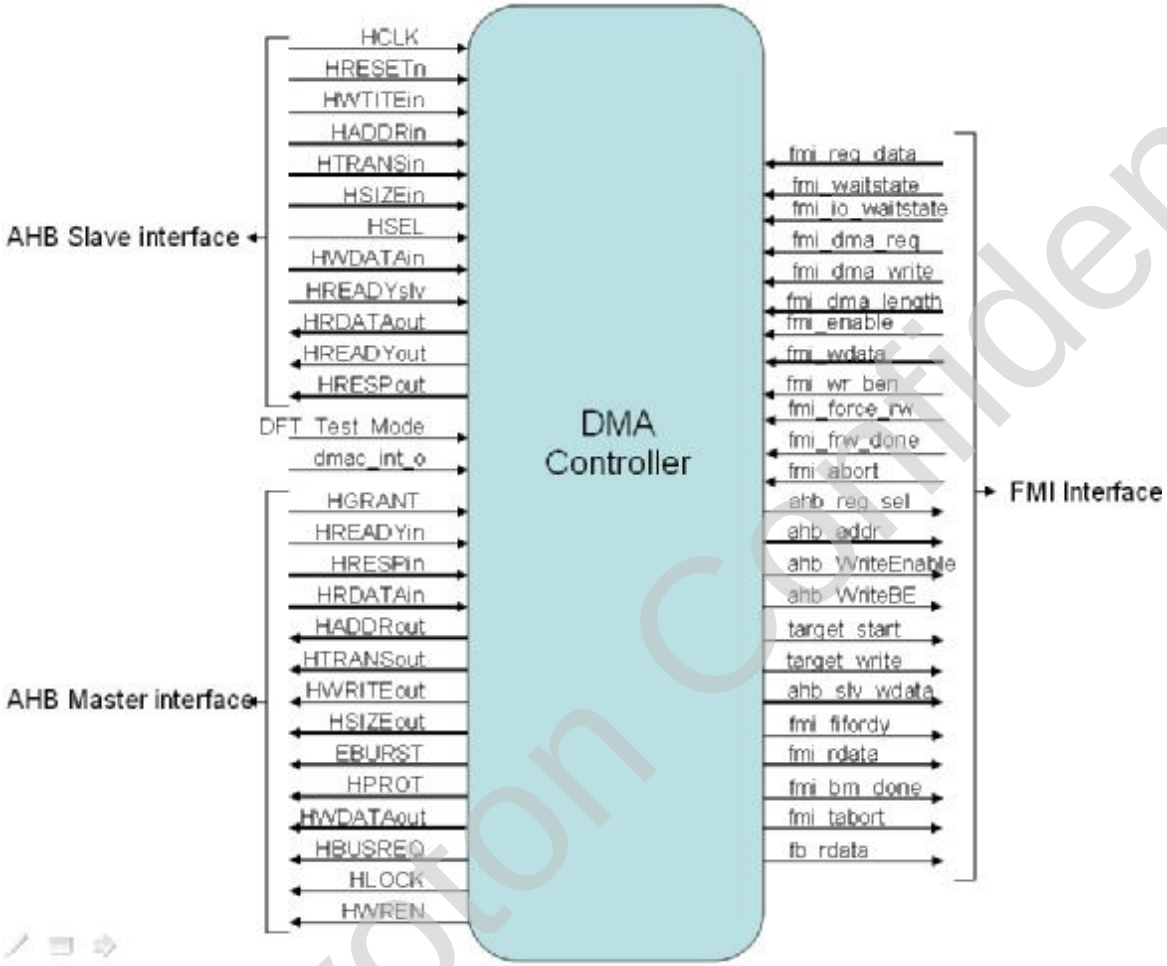


Figure 6.13-2 DMA Controller Symbol Diagram

6.13.4.3 Block Diagram

The block diagram of DMA Controller is shown as following.

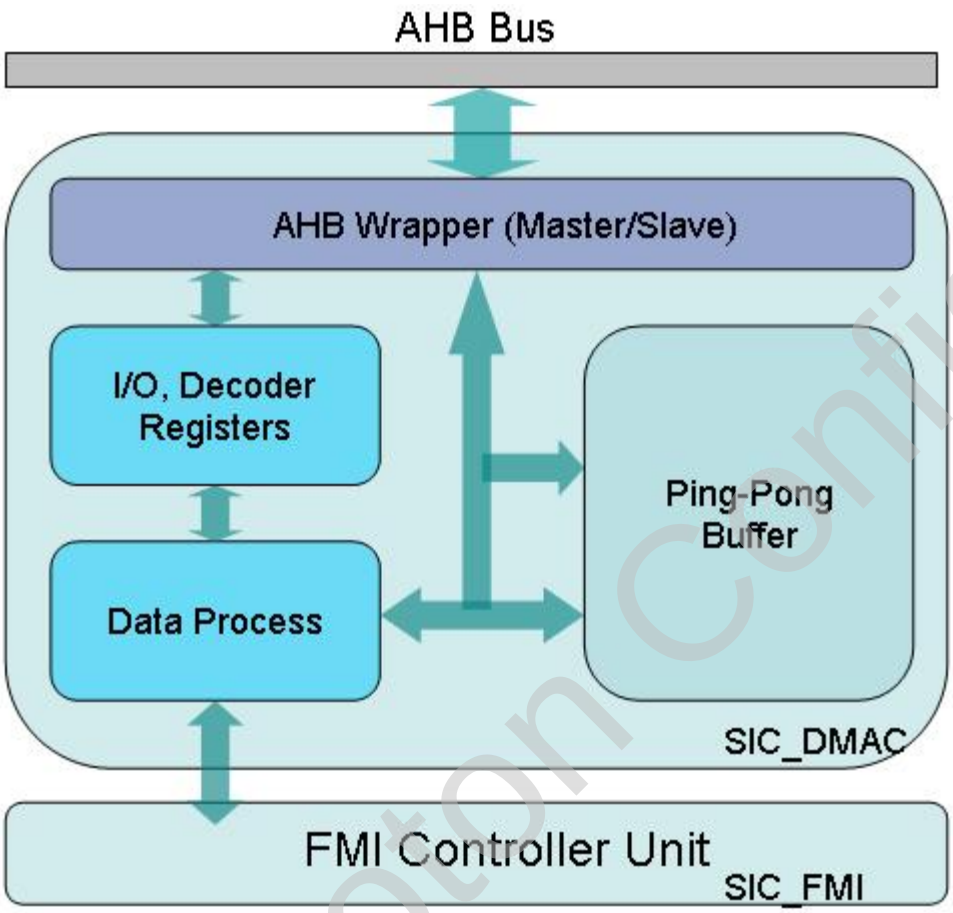


Figure 6.13-3 DMA Controller Block Diagram

6.13.4.4 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.**

6.13.4.5 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	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

6.13.4.6 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	Reserve	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 DMACSAR as starting pointer of a single block memory. • 1 = Enable scatter-gather operation. DMAC will treat the starting address in DMACSAR 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 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 (DMACSAR)

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

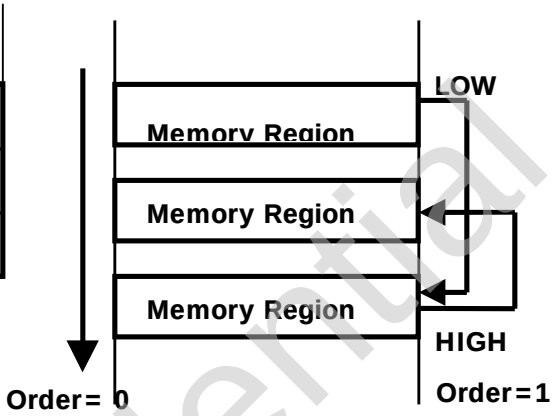
31	30	29	28	27	26	25	24
DMACSA[31:24]							
23	22	21	20	19	18	17	16
DMACSA[23:16]							
15	14	13	12	11	10	9	8
DMACSA[15:8]							
7	6	5	4	3	2	1	0
DMACSA[7:0]							

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 must be word aligned, for example, 0x0000_0000, 0x0000_0004...

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 Address			
EOT	Reserved	Byte Count	



Physical Base Address: 32-bit

Byte Count: must be multiples of 4 bytes, Max:65532 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 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 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, 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.

6.13.5 Flash Memory Interface Controller (FMI)

The Flash Memory Interface of W55FA95 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.

6.13.5.1 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/24bit 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.**

6.13.5.2 Symbol Diagram

The symbol diagram of FMI Controller is shown as following.

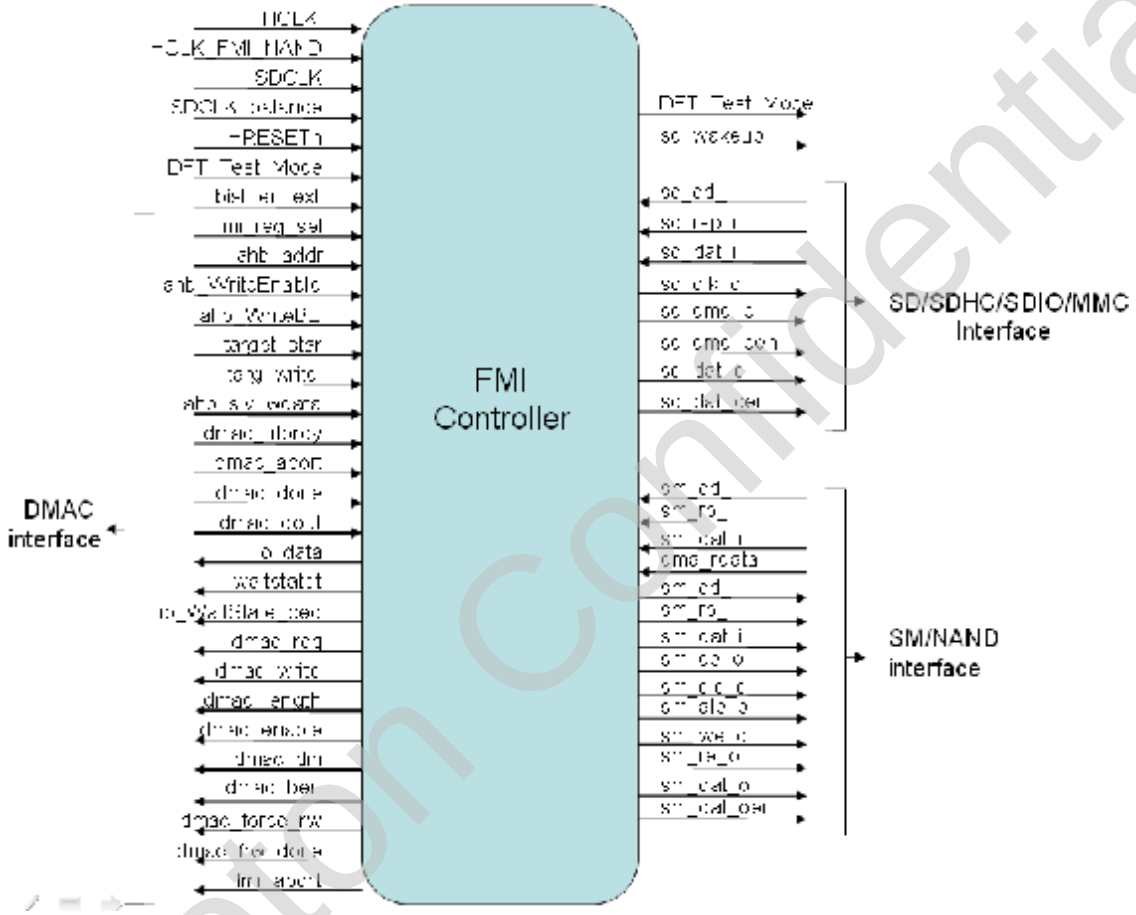


Figure 6.13-4 FMI Controller Symbol Diagram

6.13.5.3 Block Diagram

A simple block diagram of FMI Controller is shown as following.

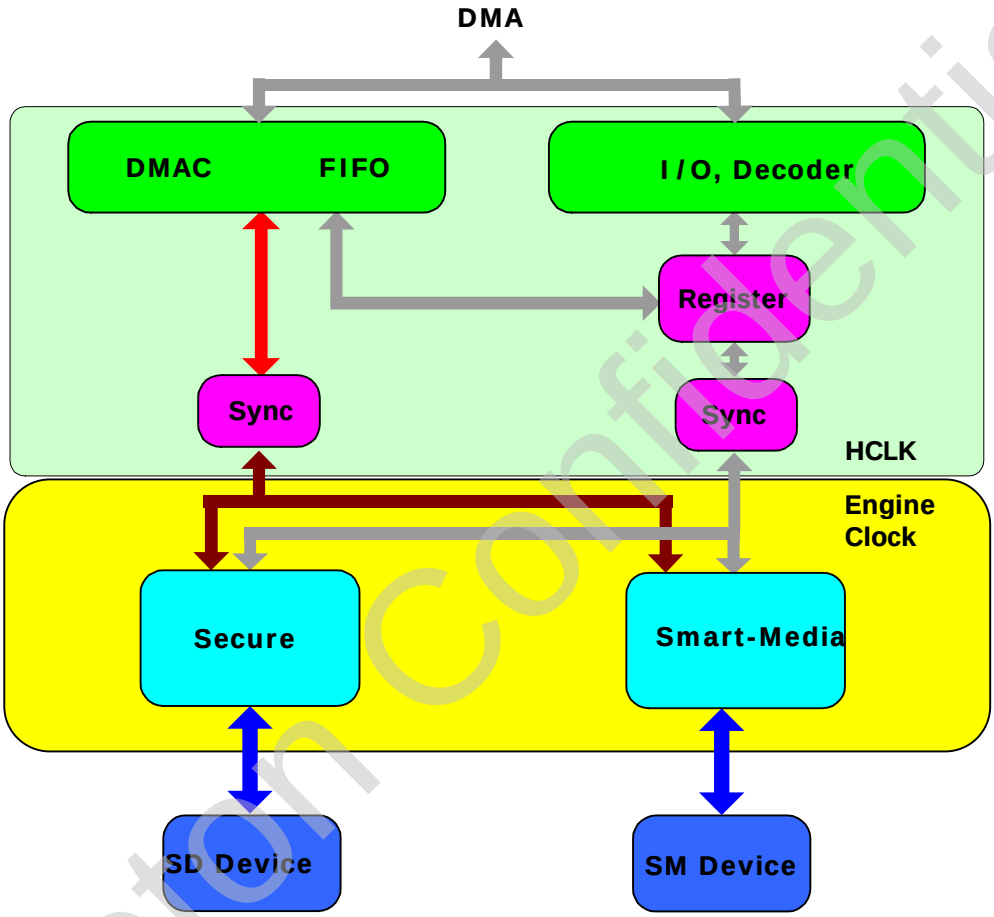


Figure 6.13-5 FMI Controller Block Diagram

6.13.5.4 Function Description

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.

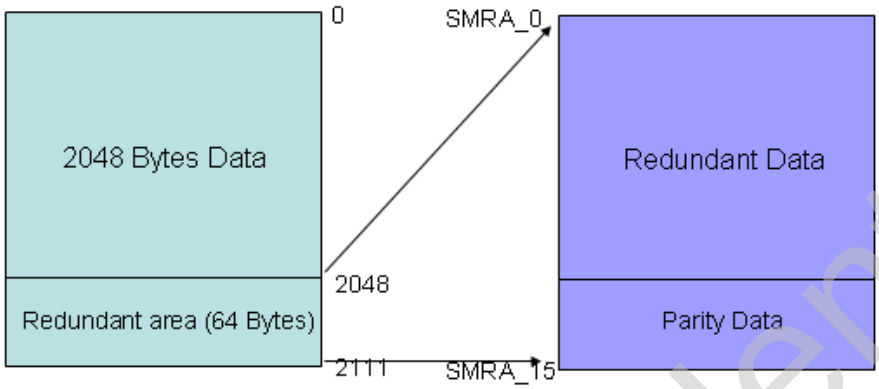
There are four page sizes for choose, 512bytes/page, 2048bytes/page, 4096bytes/page and 8192bytes/page. Using SMCR [PSIZE] to select your NAND type. Software can use DMA function for data transfer to increase your performance. For different model of NAND, software should adjust the timing parameter at 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 or 24 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.

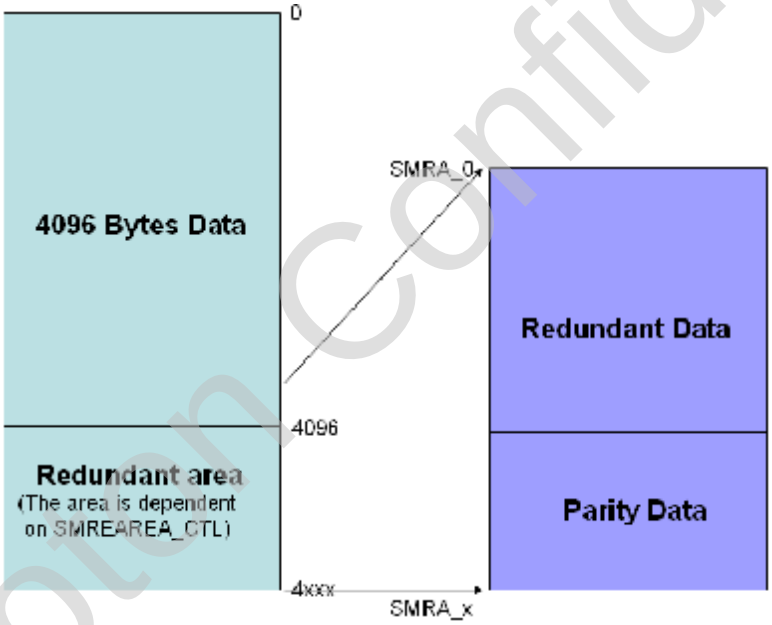
For 512/2K/4K/8K Page size NAND flash with BCH algorithm, T can be t4 or t8 or t12 or t15 or t24 which parity number and redundant number is shown in table 1.3.4-a, and the data arrangement of redundant area is shown in fig 1.3.4-a ~ 1.3.4-c.

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	32	64	128
BCH T8	15	60	120	240
BCH T12	23	92	184	368
BCH T15	29	116	232	464
BCH T24	No support	90	180	360

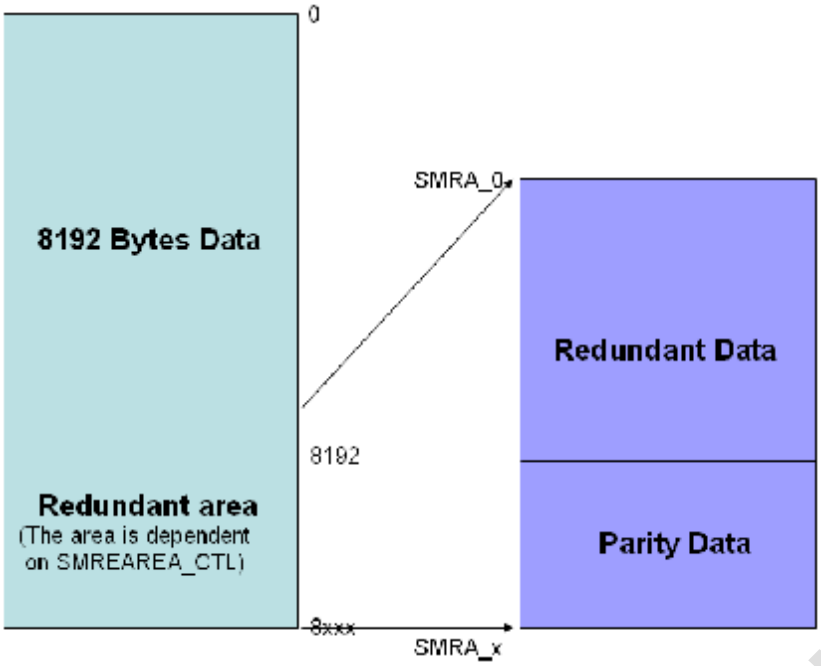
1.3.4-a Parity/Redundant number of BCH algorithm



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

6.13.5.5 FMI Controller Registers 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
FMIIER	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
SM_PROT_ADDR0	FMI_BA + 0xE0	R/W	Smart-Media Protect region end address 0	0x0000_0000
SM_PROT_ADDR1	FMI_BA + 0xE4	R/W	Smart-Media Protect region end address 1	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
BCH_ECC_ADDR8	ECC_BA + 0x20	R	BCH error byte address 8	0x0000_0000
BCH_ECC_ADDR9	ECC_BA + 0x24	R	BCH error byte address 9	0x0000_0000
BCH_ECC_ADDR10	ECC_BA + 0x28	R	BCH error byte address 10	0x0000_0000
BCH_ECC_ADDR11	ECC_BA + 0x2C	R	BCH error byte address 11	0x0000_0000
Smart-Media BCH Error Data Register				
BCH_ECC_DATA0	ECC_BA + 0x60	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DATA1	ECC_BA + 0x64	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DATA2	ECC_BA + 0x68	R	BCH ECC Error Data 0	0x8080_8080
BCH_ECC_DATA3	ECC_BA + 0x6C	R	BCH ECC Error Data 0	0x8080_8080

BCH_ECC_DATA4	ECC_BA + 0x70	R	BCH ECC Error Data 4	0x8080_8080
BCH_ECC_DATA5	ECC_BA + 0x74	R	BCH ECC Error Data 5	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

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 (FMIIER)

Register	Offset	R/W	Description	Reset Value
FMIIER	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. Don't fill 0x0 to this field. • Note: For READ_MULTIPLE_BLOCK and WRITE_MULTIPLE_BLOCK command, the actual total length is BLK_CNT * (SDBLEN +1).
[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_KEEP0	**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. Host need clock to got data on pin DAT3. Please make sure SDCR[CLK_KEEP0] is 1 in order to generate free running clock for 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. • If SDIER[CD0SRC] = 0 to select DAT3 for card detect 0 = card removed 1 = card inserted • If SDIER[CD0SRC] = 1 to select GPIO for card detect 0 = card inserted 1 = card removed

[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					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	PROT_ REGION_EN	REDUN_ AUTO_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 or 24 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.

[22:18]	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 or 24 for choosing (correct 4 or 8 or 12 or 15 or 24 bits). • 10000 = Using BCH T15 to encode/decode (T15). • 01000 = Using BCH T12 to encode/decode (T12). • 00100 = Using BCH T8 to encode/decode (T8). • 00010 = Using BCH T4 to encode/decode (T4). • 00001 = Using BCH T24 to encode/decode (T24).(1024 Bytes per block)
[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. Before setting PSIZE register, user must set BCH_TSEL register at first. • 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 Softeware 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]	Reserved	Reserved
[5]	PROT_REGION_EN	Protect Region Enable This field is used to protect NAND Flash region from address 0 to address {SM_PROT_ADDR1,, SM_PROT_ADDR0} not be written. • 0 = Disable • 1 = Enable
[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 = Disable auto write redundant data out to NAND flash. • 1 = Enable 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)

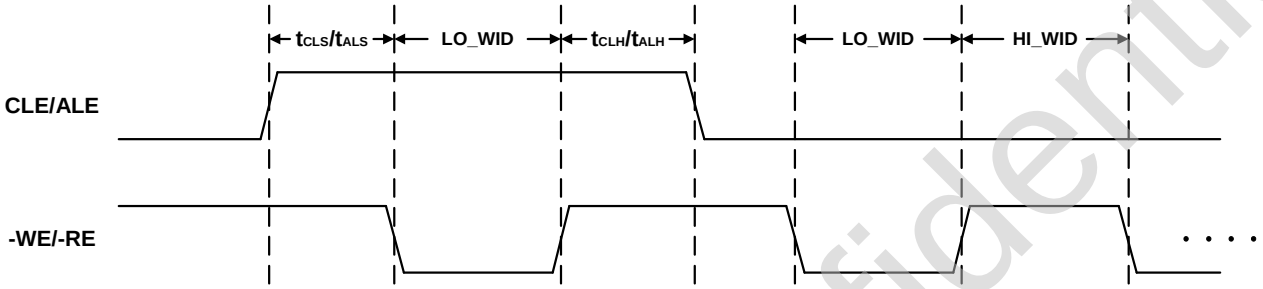
Register	Offset	R/W	Description	Reset Value
SMTCR	0x0A4	R/W	Smart-Media Timing Control Register	0x0001_0105

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 −WE. The setup/hold time can be calculated using following equation: t_{CLS} = (CALE_SH+1)*T_{AHB} t_{CLH} = ((CALE_SH*2)+2)*T_{AHB} t_{ALS} = (CALE_SH+1)*T_{AHB} t_{ALH} = ((CALE_SH*2)+2)*T_{AHB}
[15:8]	HI_WID	Read/Write Enable Signal High Pulse Width This field controls the high pulse width of signals −RE and −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 [clock period*(HI_WID+1)])
[7:0]	LO_WID	Read/Write Enable Signal Low Pulse Width

		This field controls the low pulse width of signals −RE and −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 [clock period*(LO_WID+1)])

NOTE1: The reset value is calculated base on 100MHz AHB Clock.



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				PROT_REGION_WR_IE	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.
[9:4]	Reserved	Reserved
[3]	PROT_REGION_WR_IE	Protect Region Write Detect Interrupt Enable • 0=Disable interrupt generation for detect writing to NAND Flash's protect region • 1=Enable interrupt generation for detect writing to NAND Flash's protect region

[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				PROT_REGION_WR_IF	ECC_FLD_IF	Reserved	DMA_IF

Bits	Descriptions	
[31:19]	Reserved	Reserved
[19]	RB1_Status	Ready/-Busy 1 Pin Status (Read Only) This bit reflects the Ready/-Busy pin status of Smart-Media card.
[18]	RB0_Status	Ready/-Busy 0 Pin Status (Read Only) This bit reflects the Ready/-Busy pin status of Smart-Media card.
[17:16]	Reserved	Reserved
[15:12]	Reserved	Reserved
[11]	RB1_IF	Ready/-Busy 1 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.

[10]	RB0_IF	Ready/-Busy 0 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.
[9:4]	Reserved	Reserved
[3]	PROT_REGION_WR_IF	Protect Region Write Detect Interrupt Flag (Read Only) • 0 = Writing to NAND Flash's protect region is not detected. • 1 = Writing to NAND Flash's protect region is detected. NOTE: This bit is read only, but can be cleared by writing '1' to it.
[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 = Error not occurrence. • 1 = Error occurrence 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]	Reserved	Reserved
[30:26]	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]	Reserved	Reserved

[22:18]	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]	Reserved	Reserved
[14:10]	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]	Reserved	Reserved
[6:2]	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]	Reserved	Reserved
[30:26]	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]	Reserved	Reserved

[22:18]	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]	Reserved	Reserved
[14:10]	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]	Reserved	Reserved
[6:2]	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]	Reserved	Reserved
[30:26]	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]	Reserved	Reserved

[22:18]	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]	Reserved	Reserved
[14:10]	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. F4_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]	Reserved	Reserved
[6:2]	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]	Reserved	Reserved
[30:26]	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]	Reserved	Reserved

[22:18]	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]	Reserved	Reserved
[14:10]	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]	Reserved	Reserved
[6:2]	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.

Smart-Media Protect End Address Register 0 (SM_PROT_ADDR0)

Register	Address	R/W	Description	Reset Value
SM_PROT_ADDR0	FMI_BA+0x0E0	R/W	Smart-Media Protect End Address Register 0	0x0000_0000

31	30	29	28	27	26	25	24
SM_PROT_ADDR0							
23	22	21	20	19	18	17	16
SM_PROT_ADDR0							
15	14	13	12	11	10	9	8
SM_PROT_ADDR0							
7	6	5	4	3	2	1	0
SM_PROT_ADDR0							

Bits	Descriptions	
[31:0]	SM_PROT_ADDR0	Smart-Media Protect End Address Register 0 By setting register SM_PROT_ADDR0, SM_PROT_ADDR1 and enable SMCR[PROT_REGION_EN], the NAND Flash from address 0 to address {SM_PROT_ADDR1, SM_PROT_ADDR0} region will be write protect.

Smart-Media Protect End Address Register 1 (SM_PROT_ADDR1)

Register	Address	R/W	Description	Reset Value
SM_PROT_ADDR1	FMI_BA+0x0E4	R/W	Smart-Media Protect End Address 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
SM_PROT_ADDR1							

Bits	Descriptions	
[7:0]	SM_PROT_ADDR1	Smart-Media Protect End Address Register 1 By setting register SM_PROT_ADDR0, SM_PROT_ADDR1 and enable SMCR[PROT_REGION_EN], the NAND Flash from address 0 to address {SM_PROT_ADDR1, SM_PROT_ADDR0} region will be write protect.

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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_1	ECC Error Address First Field of Error 1 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_0	ECC Error Address First Field of Error 0 This field contains a 11-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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_3	ECC Error Address First Field of Error 3 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_2	ECC Error Address First Field of Error 2 This field contains a 11-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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_5	ECC Error Address First Field of Error 5 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_4	ECC Error Address First Field of Error 4 This field contains a 11-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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_7	ECC Error Address First Field of Error 7 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_6	ECC Error Address First Field of Error 6 This field contains a 11-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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_9	ECC Error Address First Field of Error 9 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_8	ECC Error Address First Field of Error 8 This field contains a 11-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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_11	ECC Error Address First Field of Error 11 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_10	ECC Error Address First Field of Error 10 This field contains a 11-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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_13	ECC Error Address First Field of Error 13 This field contains a 11-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.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_12	ECC Error Address First Field of Error 12 This field contains a 11-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					E_ADDR_FF_15		
23	22	21	20	19	18	17	16
E_ADDR_FF_15							
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	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_15	ECC Error Address First Field of Error 15 This field contains a 11-bit ECC error address 15 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF15 for correcting this error.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_14	ECC Error Address First Field of Error 14 This field contains a 11-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 Error Address 8 (BCH_ERR_ADDR8)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR8	FMI_BA+0x120	R	BCH Error Byte Address8.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					E_ADDR_FF_17		
23	22	21	20	19	18	17	16
E_ADDR_FF_17							
15	14	13	12	11	10	9	8
Reserved					E_ADDR_FF_16		
7	6	5	4	3	2	1	0
E_ADDR_FF_16							

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_17	ECC Error Address First Field of Error 17 This field contains a 11-bit ECC error address 17 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF17 for correcting this error.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_16	ECC Error Address First Field of Error 16 This field contains a 11-bit ECC error address 16 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF16 for correcting this error.

BCH Error Address 9 (BCH_ERR_ADDR9)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR9	FMI_BA+0x124	R	BCH Error Byte Address9.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					E_ADDR_FF_19		
23	22	21	20	19	18	17	16
E_ADDR_FF_19							
15	14	13	12	11	10	9	8
Reserved					E_ADDR_FF_18		
7	6	5	4	3	2	1	0
E_ADDR_FF_18							

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_19	ECC Error Address First Field of Error 19 This field contains a 11-bit ECC error address 19 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF19 for correcting this error.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_18	ECC Error Address First Field of Error 18 This field contains a 11-bit ECC error address 18 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF18 for correcting this error.

BCH Error Address 10 (BCH_ERR_ADDR10)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR10	FMI_BA+0x128	R	BCH Error Byte Address10.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					E_ADDR_FF_21		
23	22	21	20	19	18	17	16
E_ADDR_FF_21							
15	14	13	12	11	10	9	8
Reserved					E_ADDR_FF_20		
7	6	5	4	3	2	1	0
E_ADDR_FF_20							

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_21	ECC Error Address First Field of Error 21 This field contains a 11-bit ECC error address 21 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF21 for correcting this error.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_20	ECC Error Address First Field of Error 20 This field contains a 11-bit ECC error address 20 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF20 for correcting this error.

BCH Error Address 11 (BCH_ERR_ADDR11)

Register	Address	R/W	Description	Reset Value
BCH_ERR_ADDR11	FMI_BA+0x12C	R	BCH Error Byte Address11.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					E_ADDR_FF_23		
23	22	21	20	19	18	17	16
E_ADDR_FF_23							
15	14	13	12	11	10	9	8
Reserved					E_ADDR_FF_22		
7	6	5	4	3	2	1	0
E_ADDR_FF_22							

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:16]	E_ADDR_FF_23	ECC Error Address First Field of Error 23 This field contains a 11-bit ECC error address 23 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF23 for correcting this error.
[15:11]	Reserved	Reserved
[10:0]	E_ADDR_FF_22	ECC Error Address First Field of Error 22 This field contains a 11-bit ECC error address 22 of first field. If it is a correctable error, software can read the error data at BCH_ECC_DATA_FF22 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	0x8080_8080

31	30	29	28	27	26	25	24
E_DATA_FF_15							
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]	E_DATA_FF_15	ECC Error Data Of First Field 15 This field contains an 8-bit BCH ECC error data 15 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_15; the result will be the correct data.
[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.

BCH ECC Error Data (BCH_ECC_DATA4)

Register	Address	R/W	Description	Reset Value
BCH_ECC_DATA4	FMI_BA+0x170	R	BCH ECC Error Data 4	0x8080_8080

31	30	29	28	27	26	25	24
E_DATA_FF_19							
23	22	21	20	19	18	17	16
E_DATA_FF_18							
15	14	13	12	11	10	9	8
E_DATA_FF_17							
7	6	5	4	3	2	1	0
E_DATA_FF_16							

Bits	Descriptions	
[31:24]	E_DATA_FF_19	ECC Error Data Of First Field 19 This field contains an 8-bit BCH ECC error data 19 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_19; the result will be the correct data.
[23:16]	E_DATA_FF_18	ECC Error Data Of First Field 18 This field contains an 8-bit BCH ECC error data 18 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_18; the result will be the correct data.
[15:8]	E_DATA_FF_17	ECC Error Data Of First Field 17 This field contains an 8-bit BCH ECC error data 17 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_17; the result will be the correct data.

[7:0]	E_DATA_FF_16	**ECC Error Data Of First Field 16** This field contains an 8-bit BCH ECC error data 16 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_16; the result will be the correct data.

BCH ECC Error Data (BCH_ECC_DATA5)

Register	Address	R/W	Description	Reset Value
BCH_ECC_DATA5	FMI_BA+0x174	R	BCH ECC Error Data 5	0x8080_8080

31	30	29	28	27	26	25	24
E_DATA_FF_23							
23	22	21	20	19	18	17	16
E_DATA_FF_22							
15	14	13	12	11	10	9	8
E_DATA_FF_21							
7	6	5	4	3	2	1	0
E_DATA_FF_20							

Bits	Descriptions	
[31:24]	E_DATA_FF_23	ECC Error Data Of First Field 23 This field contains an 8-bit BCH ECC error data 23 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_23; the result will be the correct data.
[23:16]	E_DATA_FF_22	ECC Error Data Of First Field 22 This field contains an 8-bit BCH ECC error data 22 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_22; the result will be the correct data.
[15:8]	E_DATA_FF_21	ECC Error Data Of First Field 21 This field contains an 8-bit BCH ECC error data 21 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_21; the result will be the correct data.

[7:0]	E_DATA_FF_20	**ECC Error Data Of First Field 20** This field contains an 8-bit BCH ECC error data 20 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_20; the result will be the correct data.

Smart-Media Redundant Area Register (SMRA)

Register	Address	R/W	Description	Reset Value
SM_RA0 ... SM_RA117	FMI_BA+0x200 ... FMI_BA+0x3D4	R/W	Smart-Media Redundant Area Register	N/A

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.

6.14 USB 2.0 Device Controller

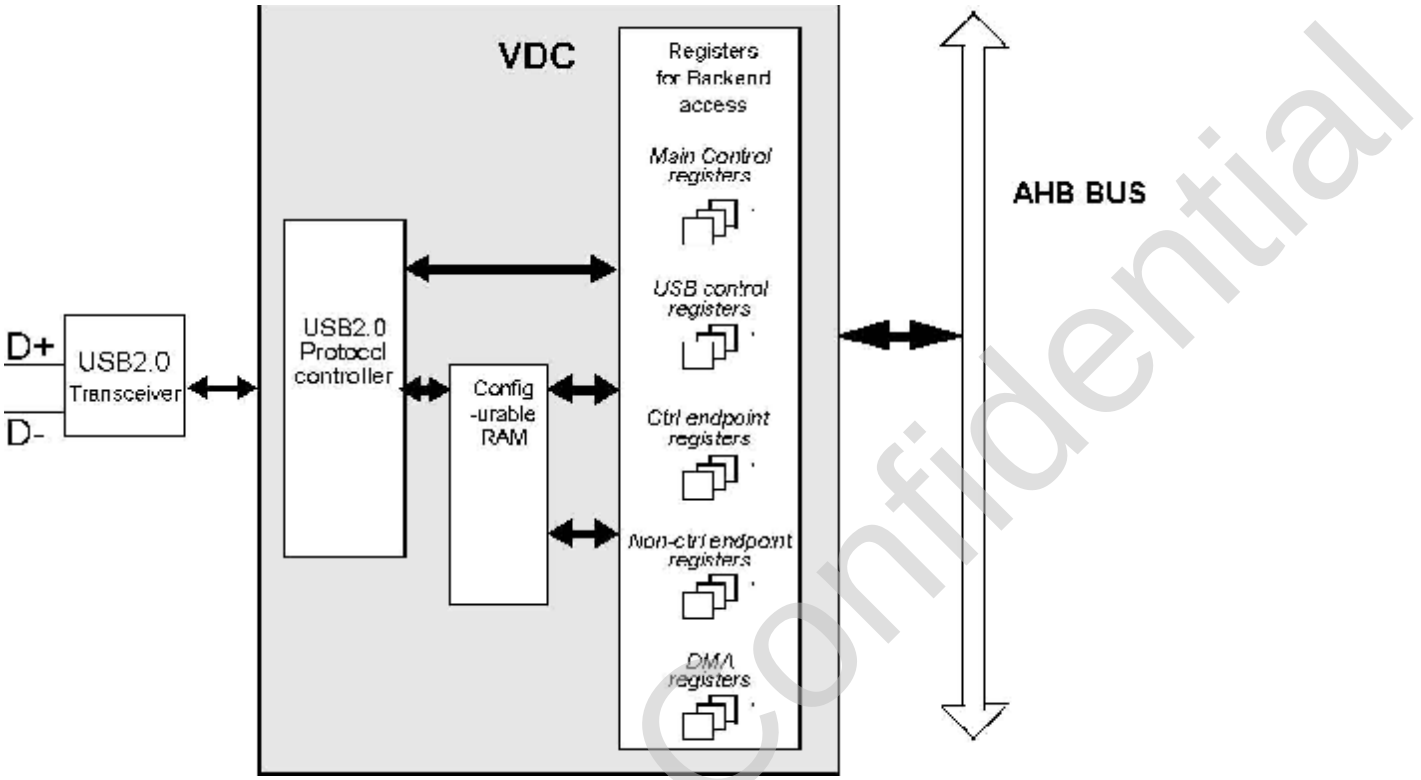
6.14.1 Overview

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.

6.14.2 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.**

6.14.3 Internal Block Diagram



6.14.4 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

6.14.5 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_0000
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_0000
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_0000
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_0000
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_0000
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

6.14.6 USB Device Control Registers

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	
[31:6]	Reserved	
[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	
[31:6]	Reserved	
[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
Reserved	TCLKOK_IS	DMACOM_IS	HISPD_IS	SUS_IS	RUM_IS	RST_IS	SOF_IS

Bits	Descriptions	
[31:9]	Reserved	
[8]	VBUS_IS	VBUS Interrupt This bit is set when the Vbus is attached/de-attached. Writing `1″ clears this bit.
[7]	Reserved	
[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 weither 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
Reserved	TCLKOK_IE	DMACOM_IE	HISPD_IE	SUS_IE	RUM_IE	RST_IE	SOF_IE

Bits	Descriptions	
[31:9]	Reserved	
[8]	VBUS_IE	VBUS Detected Interrupt Enable. This bit enables the Vbus detected interrupt.
[7]	Reserved	
[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	
[31:3]	Reserved	
[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	
[31:14]	Reserved	
[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	
[31:7]	Reserved	
[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	
[31:3]	Reserved	
[2:0]	TESTMODE	Test Mode Select. Ÿ Value Test Ÿ Normal Operation Ÿ Test_J Ÿ 010 Test_K Ÿ 011 Test_SE0_NAK Ÿ 100 Test_Packet Ÿ 101 Reserved Ÿ 110 Reserved Ÿ 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	USBD_BA+0x028	R	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 from ram buffer).

Control-ep Control and Status (CEP_CTRL_STAT)

Register	Address	R/W	Description	Default Value
CEP_CTRL_STAT	USBD_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	
[31:4]	Reserved	
[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		Reversed	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_IE	SETUP_TK_IE

Bits	Descriptions	
[31:14]	Reserved	
[13]	Reserved	
[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_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		Reversed	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	
[31:14]	Reversed	
[13]	Reversed	
[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 controlendpt receives a ping token from the host. Write "1" clear.
[3]	IN_TK_IS	**In Token Interrupt.** This bit is set when the controlendpt 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	
[31:8]	Reserved	
[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
Reserved							
7	6	5	4	3	2	1	0
OUT_TRF_CNT							

Bits	Descriptions	
[31:8]	Reserved	
[7:0]	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	
[31:16]	Reserved	
[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	
[31:16]	Reserved	

[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.

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

<table>
<tr>
<td>[7:0]</td>
<td>SETUP0</td>
<td>
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.
<table>
<tr><th>Bits</th><th colspan="2">Descriptions</th></tr>
<tr><td>[7]</td><td>Direction</td><td>0 = host to device;
1 = device to host</td></tr>
<tr><td>[6:5]</td><td>Type</td><td>0 = Standard,
1 = Class,
2 = Vendor,
3 = Reserved</td></tr>
<tr><td>[4:0]</td><td>Recipient</td><td>0 = Device,
1 = Interface,
2 = Endpoint,
3 = Other,
4-31 Reserved</td></tr>
</table>
</td>
</tr>
</table>

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	
[31:16]	Reserved	
[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	
[31:16]	Reserved	
[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	
[31:16]	Reserved	
[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	
[31:12]	Reserved	
[11:0]	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	
[31:12]	Reserved	
[11:0]	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	
[31:20]		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	Endpoint A Data Register	0x0000_0000
EPB_DATA_BUF	USBD_BA+0x08C	R	Endpoint B Data Register	0x0000_0000
EPC_DATA_BUF	USBD_BA+0x0B4	R	Endpoint C Data Register	0x0000_0000
EPD_DATA_BUF	USBD_BA+0x0DC	R	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 only).

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_0000
EPB_IRQ_STAT	USBD_BA+0x090	R/W	Endpoint B Interrupt Status Register	0x0000_0000
EPC_IRQ_STAT	USBD_BA+0x0B8	R/W	Endpoint C Interrupt Status Register	0x0000_0000
EPD_IRQ_STAT	USBD_BA+0x0E0	R/W	Endpoint D 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			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	
[31:13]	Reserved	
[12]	O_SHORT_PKT_IS	Bulk out short packet received. Received bulkout 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	
[31:13]	Reserved	
[12]	O_SHORT_PKT_IE	Bulk out short packet interrupt enable When set, this bit enables a local interrupt to be set whenever bulkout 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	
[31]	Reserved	
[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	
[31:8]	Reserved	
[7]	DIS_BUF	Disable buffer This bit is used to disable buffer (set buffer size to 1) when received a bulkout short packet.
[6]	PK_END	Packet End. This bit is applicable only in case of Auto-Validate Mtheod. 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) 1. If the endpoint is selected to be operating in auto-validation mode, the endpointresponds only with data payloads to be equal to EP_MPS register. The endpoint controller waits until the amount of data is equal to EP_MPS value and then validates the data. 2. If the endpoint is selected to be operating in manual-validation mode, the endpoint responds only when the data in the buffer is validated by the local-CPU every time. 3. The fly-mode of operation is the simplest mode of operation, where in there is no validation procedure. The buffer is being filled by the local-CPU.
[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	
[31:11]	Reserved	
[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	
[31:11]	Reserved	
[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	
[31:10]	Reserved	
[9:8]	EP_MULT	MULT Field. This field indicates number of transactions to be carried out in one single micro frame. [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. [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	
[31:12]	Reserved	
[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	
[31:12]	Reserved	
[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							Reserved
7	6	5	4	3	2	1	0
Reserved				FAIL_A	FINISH_A	ERR_A	MODE_A

Bits	Descriptions	
[31:4]	Rversed	
[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

Ps:support 2K bytes internal SRAM

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	
[31:4]	Reserved	
[3:0]	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_0420

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

Bits	Descriptions	
[31]	Vbus_status	Vbus status 1: Vbus on 0: Vbus off It is read only
[30:24]	Reserved	
[23:16]	Reserved	
[15:12]	Reserved	
[11:10]	Reserved	
[9]	Phy_suspend	Set this bit low will cause USB PHY suspend.
[8]	vbus_detect	Set PHY vbus_detect
[7]	Reserved	
[6]	Reserved	
[5:4]	Reserved	
[3]	Reserved	
[2]	Reserved	
[1]	Reversed	

[0]	bisten	**Built-In Self-Test (BIST) Enable** **This bit controls the input signal of bist_enb of USB PHY 0.** **This signal activates the BIST algorithm to check internal analog and digital functionality. The bist_error signal is asserted if a problem is encountered with the internal USB PHY circuitry.**

6.15 USB Host Controller (UHC)

6.15.1 Overview

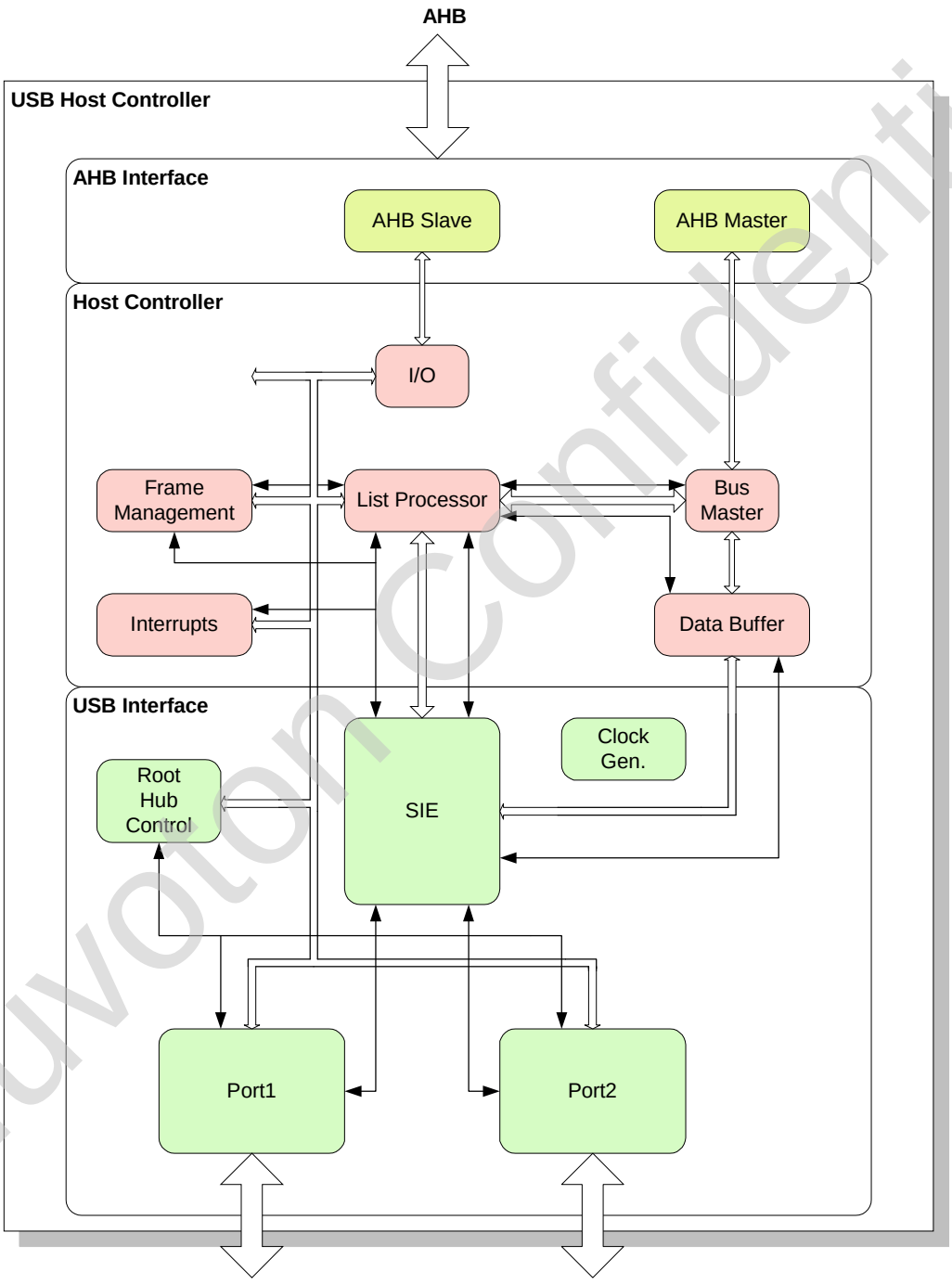
The Universal Serial Bus (USB) is a low-cost, low-to mid-speed peripheral interface standard intended for modem, scanners, PDAs, keyboards, mice, and other devices that do not require a high-bandwidth parallel interface. The USB is a 4-wire serial cable bus that supports serial data exchange between 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.

6.15.2 Features

- Fully compliant with USB Revision 1.1 specification.
- Open Host Controller Interface (OHCI) Revision 1.0 compatible.
- Supports both full-speed (12Mbps) and low-speed (1.5Mbps) USB devices.
- Supports Control, Bulk, Interrupt and Isochronous transfers.
- Built-in DMA for real-time data transfer.
- Multiple low power modes for efficient power management.

6.15.3 Block Diagram



6.15.4 Operation

6.15.4.1 OHCI Controller

6.15.4.1.1 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.

6.15.4.1.2 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.

6.15.4.1.3 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.

6.15.5 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	**0x0000_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**

6.15.6 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							Rev
7	6	5	4	3	2	1	0
Reserved							

Bits	Descriptions	
[31:9]	Reserved	
[8:0]	Rev	Revision Indicates the Open HCI Specification revision number implemented by the Hardware. Host Controller supports 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					Reserved	Reserved	IntRoute
7	6	5	4	3	2	1	0
HcFunc		BlkEn	CtrlEn	ISOEn	PeriEn	CtrlBlkRatio	

Bits	Descriptions	
[31:11]	Reserved	
[10]	Reserved	
[9]	Reserved	
[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				Reserved	BlkFill	CtrlFill	Reserved

Bits	Descriptions	
[31:18]	Reserved	
[17:16]	SchOverRun	Schedule Overrun Count This field is increment every time the SchedulingOverrun bit in *HcInterruptStatus* is set. The count wraps from '11' to '00.'
[15:4]	Reserved	
[3]	Reserved	
[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]	Reserved	

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	Reserved	Resume	SOF	WBDnHD	SchOR

Bits	Descriptions	
[31]	Reserved	
[30]	OC	Ownership Change This bit is set when the OwnershipChangeRequest bit of *HcCommandStatus* is set.
[29:7]	Reserved	
[6]	RHSC	Root Hub Status Change This bit is set when the content of *HcRhStatus* or the content of any *HcRhPortStatus* register has changed.
[5]	FNOF	Frame Number Overflow Set when bit 15 of FrameNumber changes value.
[4]	Reserved	
[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 *HcDoneHead* to *HccaDoneHead*.

[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	Reserved	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]	Reserved	
[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	Reserved	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]	Reserved	
[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	
[31:8]	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
CtrlCED							
23	22	21	20	19	18	17	16
CtrlCED							
15	14	13	12	11	10	9	8
CtrlCED							
7	6	5	4	3	2	1	0
CtrlCED				Reserved			

Bits	Descriptions	
[31:4]	CtrlCED	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
BlkHED							
23	22	21	20	19	18	17	16
BlkHED							
15	14	13	12	11	10	9	8
BlkHED							
7	6	5	4	3	2	1	0
BlkHED				Reserved			

Bits	Descriptions	
[31:4]	BlkHED	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	Reserved	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]	Reserved	
[29: 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_0628

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	
[11:0]	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	0x1000_0002

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

Bits	Descriptions	
[31:26]	Reserved	
[25:24]	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]	Reserved	
[9]	Reserved	
[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
[7:4]	Reserved	
[3:0]	DPortNum	Number Downstream Ports HYDRA-4 supports two downstream ports. It is read only.

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
Reserved							
23	22	21	20	19	18	17	16
Reserved					PPCM		Reserved
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					DevRemove		Reserved

Bits	Descriptions	
[31:19]	Reserved	
[18:17]	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
[16:3]	Reserved	

[2:1]	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 Unimplemented ports are reserved, read/write '0'.
[0]	Reserved	

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
Reserved							
23	22	21	20	19	18	17	16
Reserved						OCIC	Reserved
15	14	13	12	11	10	9	8
DRWEn	Reserved						
7	6	5	4	3	2	1	0
Reserved						OC	Reserved

Bits	Descriptions	
[31:18]	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]	Reserved	
[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 It is read only.
[0]	Reserved	

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]	OCALow	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.

6.16 Enhanced DMA Controller

The W55FA95 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 6-channel DMA that include 2 channel VDMA (Video-DMA, Memory-to-Memory) and 4 channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral). For channel0/5~~/6~~ 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 W55FA95 also support hardware scatter-gather function, software can set CSRx [SG_EN] to enable scatter-gather function.

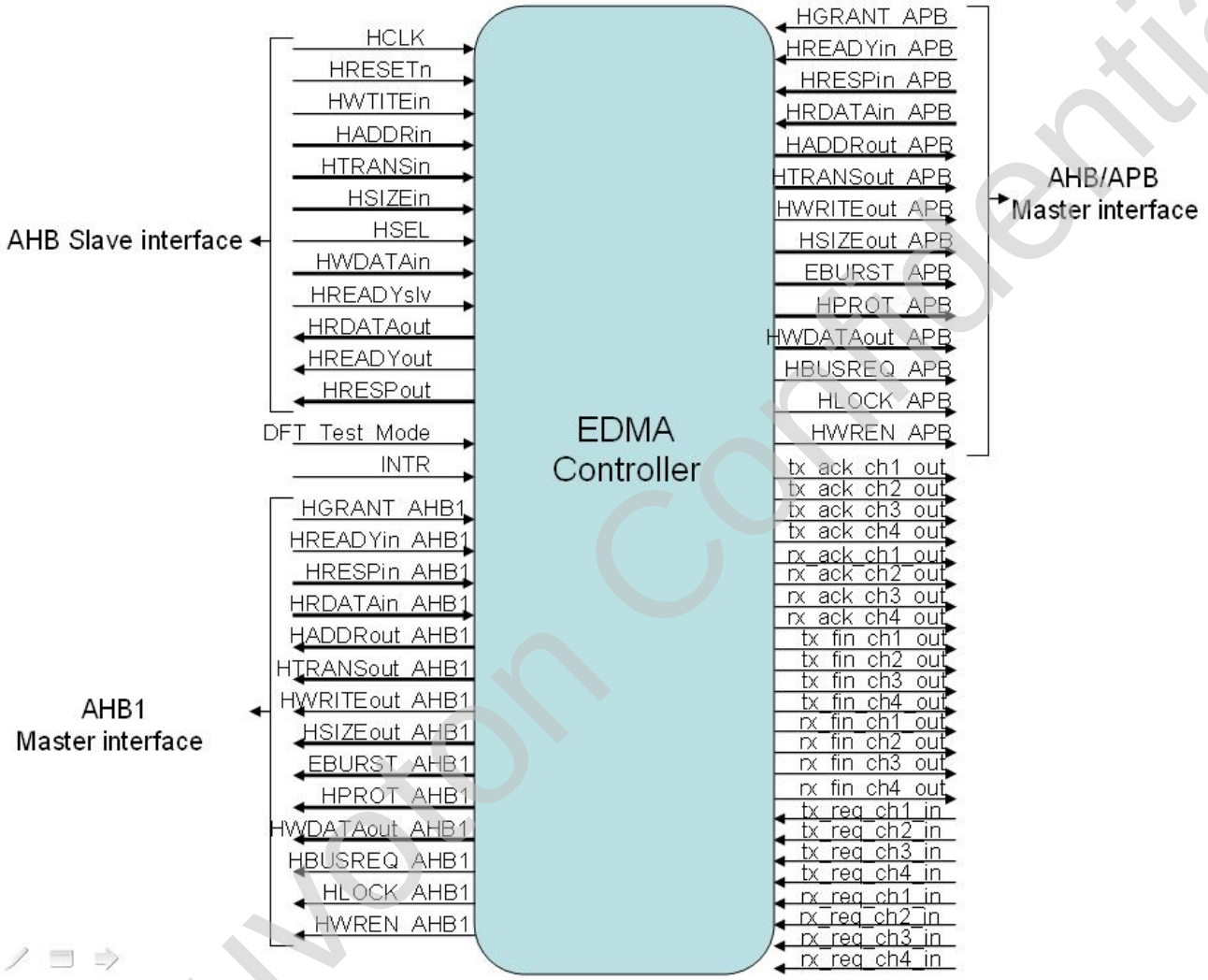
Software can stop the EDMA operation by disable DMA [DMACEN]. The CPU can recognize the completion of an EDMA operation by software polling or when it receives an internal EDMA interrupt. The W55FA95 VDMA controller can increment source or destination address, decrement or fixed them as well, and the PDMA can increment source or destination, fixed or wrap around address.

6.16.1 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.

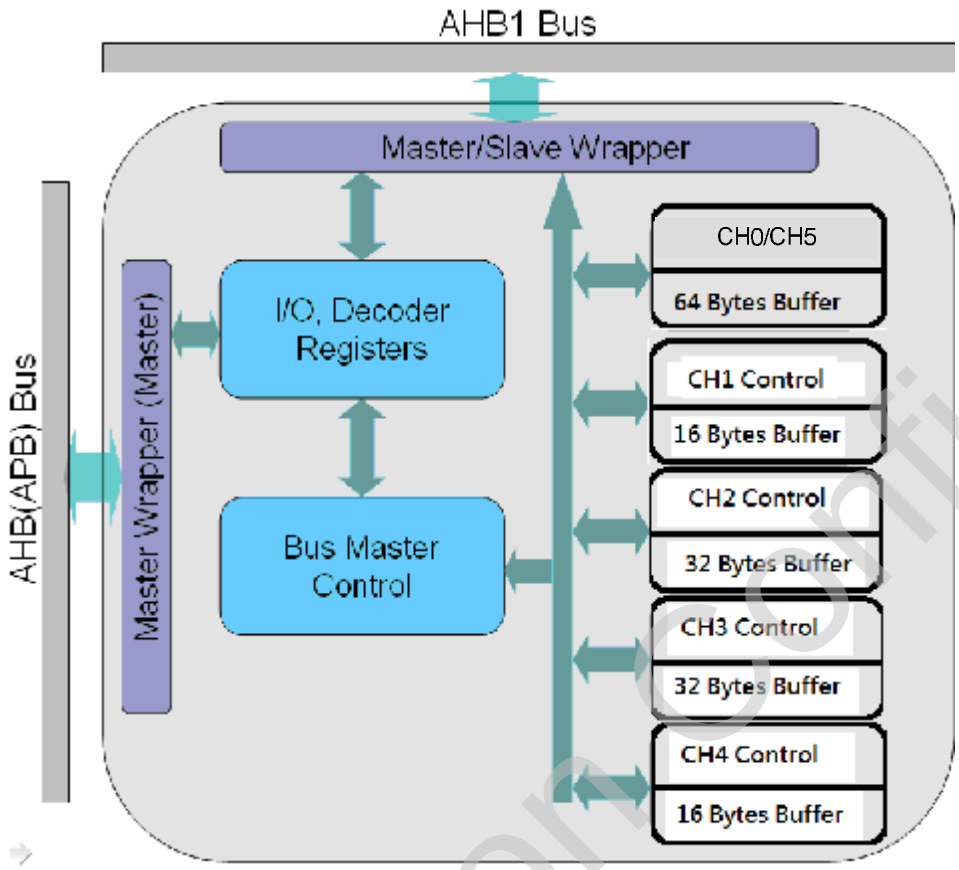
6.16.2 Symbol Diagram

6.16.3 The symbol diagram of EDMA controller is shown as fallowing.



EDMA Controller Symbol Diagram

6.16.4 Block Diagram



EDMA Controller Block Diagram

6.16.5 Function Description

The W55FA95 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 6-channel DMA that include 2 channels VDMA (Video-DMA, Memory-to-Memory) and 4 channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral). For channel 0/5~~/6~~ 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 W55FA95 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 W55FA95 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.

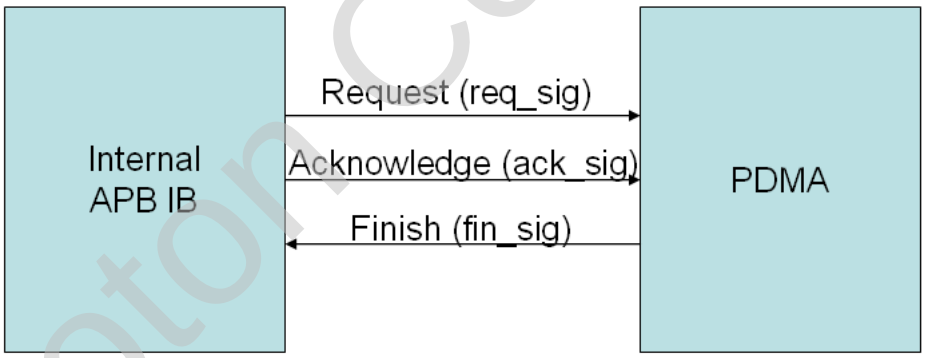


Figure 6.16-1 Peripheral interface signal

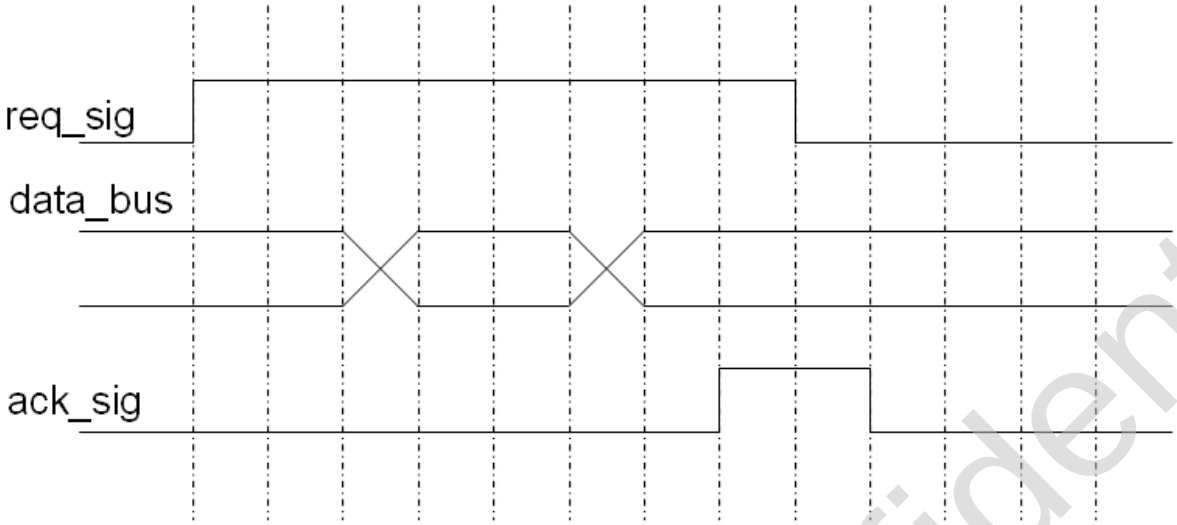


Figure 6.16-2.handsharking signal

The W55FA95 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.

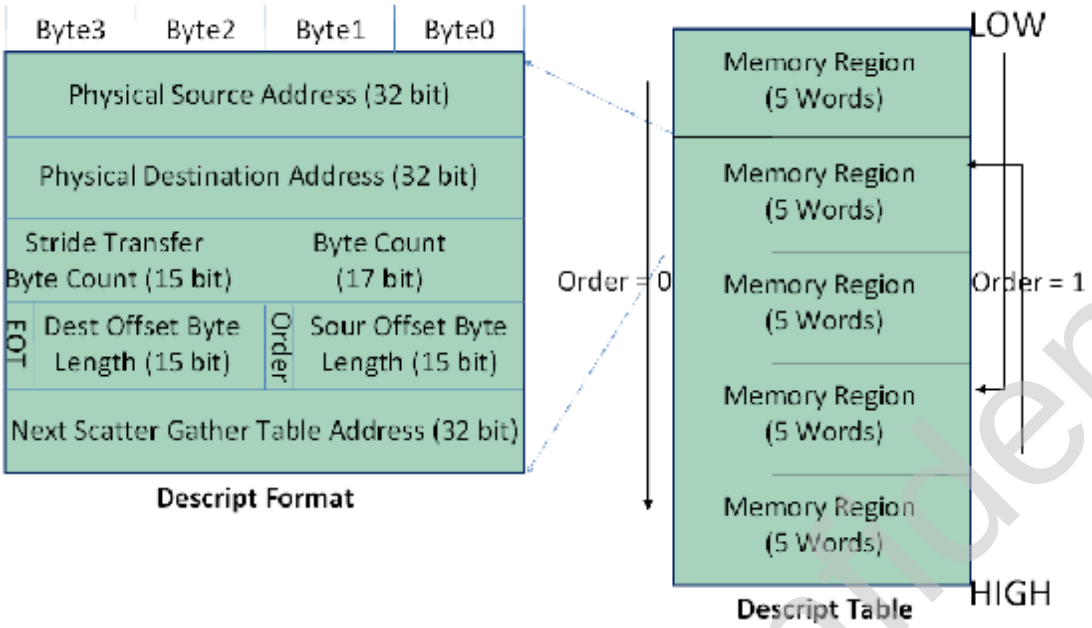
1. Set DMA_CSR [DMACEN] and DMA_CSR [SG_EN] to enable DMA and Scatter-Gather function.

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 6.16-3. Descriptor Table Format

6.16.6 EDMA Controller Registers Map

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

VDMA_BA = 0xB000_8000 (Channel 0 Base Address) VDMA_BA = 0xB000_8500 (Channel 5 Base Address) ~~VDMA_BA = 0xB000_8600 (Channel 6 Base Address)~~				
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_SBUF 0 ~ VDMA_SBUF15	VDMA_BA + 0x80 ~ VDMA_BA + 0xBC	R/W	VDMA Shared Buffer FIFO 0 ~ VDMA Shared Buffer FIFO 15	0x0000_0000
PDMA_BA = 0xB000_8100 (Channel 1) PDMA_BA = 0xB000_8200 (Channel 2) PDMA_BA = 0xB000_8300 (Channel 3) PDMA_BA = 0xB000_8400 (Channel 4)				
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	0xXXXX_0000
PDMA_SBUF 0 ~ PDMA_SBUF 3 (CH1/4)	PDMA_BA + 0x80~ PDMA_BA + 0x8C	R/W	PDMA Shared Buffer FIFO 0 ~ PDMA Shared Buffer FIFO 3	0x0000_0000
PDMA_SBUF 0 ~ PDMA_SBUF 7 (CH2/3)	PDMA_BA + 0x80~ PDMA_BA + 0x9C	R/W	PDMA Shared Buffer FIFO 0 ~ PDMA Shared Buffer FIFO 7	0x0000_0000

6.16.7 EDMA Control Register

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_SCAR)

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	INTR6	INTR5	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]	INTR6	Interrupt Pin Status of Channel 6(Read Only) This bit is the Interrupt pin status of EDMA channel4. Note: This bit is read only
[29]	INTR5	Interrupt Pin Status of Channel 5(Read Only) This bit is the Interrupt pin status of EDMA channel4. Note: This bit is read only
[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_EN	Reserved					Col_tra_EN	Stride_EN

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]							

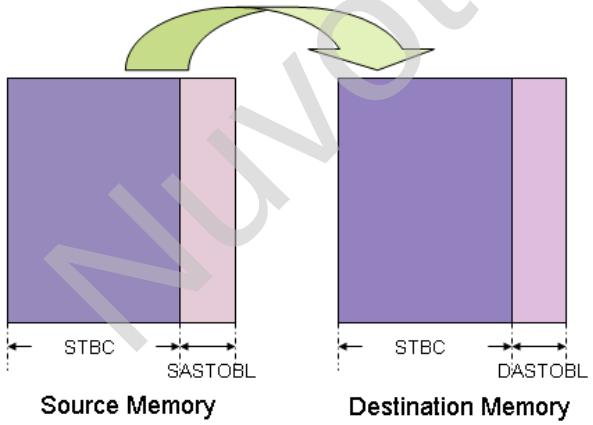
Bits	Descriptions	
[31:16]	STBC	VDMA Stride Transfer Byte Count The 16 bits register define the stride transfer byte count of each row. When in scatter-gather mode, this field will be updated automatically by descriptor table, so when in scatter-gather mode this field is not meaningful.
[15:0]	SASTOBL	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 ~ 15 (VDMA_SBUF0 ~ VDMA_SBUF15)

Register	Offset	R/W	Description	Reset Value
VDMA_SBUF0 ~ VDMA_SBUF15	0x080~ 0xBC	R/W	VDMA Shared Buffer FIFO (0 ~ 15) Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	VDMA_SBUF0~15	VDMA Shared Buffer FIFO 0~15 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	0xXXXX_0000
PDMA_POINT2	0x23C	R	PDMA Internal Buffer Pointer Register CH2	0xXXXX_0000
PDMA_POINT3	0x33C	R	PDMA Internal Buffer Pointer Register CH3	0xXXXX_0000
PDMA_POINT4	0x43C	R	PDMA Internal Buffer Pointer Register CH4	0xXXXX_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 ~ 3 (PDMA_SBUF0_ch1 and PDMA_SBUF0_ch4)

Register	Offset	R/W	Description	Reset Value
PDMA_SBUF0_c1	0x180	R/W	PDMA Shared Buffer FIFO 0 Register CH1	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 ~ 3 [31:24]							
23	22	21	20	19	18	17	16
PDMA_SBUF0 ~3 [23:16]							
15	14	13	12	11	10	9	8
PDMA_SBUF0~3 [15:8]							
7	6	5	4	3	2	1	0
PDMA_SBUF0~3 [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SBUF0 ~ 3	PDMA Shared Buffer FIFO 0 ~ FIFO 3 Each channel has its own 4 words internal buffer.

PDMA Shared Buffer FIFO 0 ~ 7 (PDMA_SBUF0_ch2 and PDMA_SBUF0_ch3)

Register	Offset	R/W	Description	Reset Value
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

31	30	29	28	27	26	25	24
PDMA_SBUF0 ~ 7 [31:24]							
23	22	21	20	19	18	17	16
PDMA_SBUF0 ~7 [23:16]							
15	14	13	12	11	10	9	8
PDMA_SBUF0~7 [15:8]							
7	6	5	4	3	2	1	0
PDMA_SBUF0~7 [7:0]							

Bits	Descriptions	
[31:0]	PDMA_SBUF0 ~ 7	PDMA Shared Buffer FIFO 0 ~ FIFO 7 Each channel has its own 4 words internal buffer.

6.17 Test Interface Controller

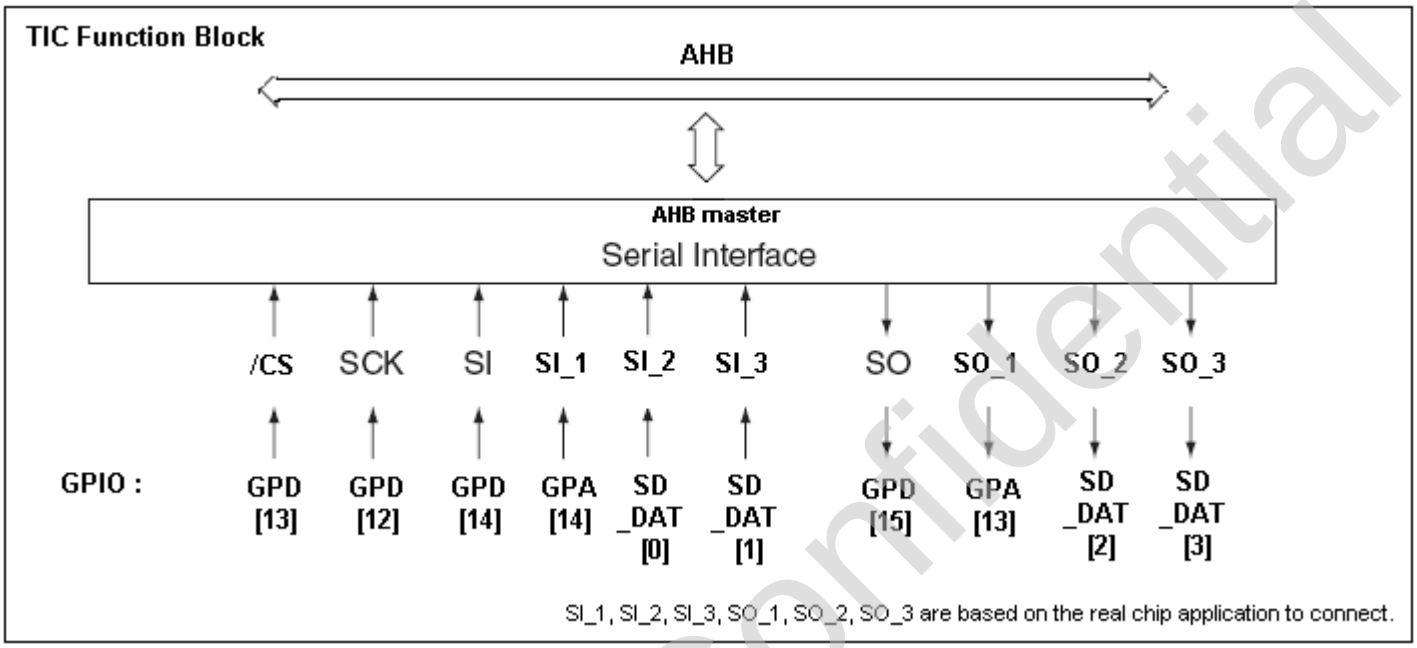


Figure 6.17-1 Port Assignments

Table 6.17-1: Serial Port Description

Symbol	Port Name	Functions
SCK	Serial Clock	To provide the timing of the serial interface. Commands, addresses, or input data are latched on the rising edge of the clock input, while output data is shifted out on the falling edge of the clock input.
SI	Serial Data Input	To transfer commands, addresses, or data serially input. Inputs are latched on the rising edge of the serial clock.
SO	Serial Data Output	To transfer data serially output. Data is shifted out on the falling edge of the serial clock.
CS	Chip Select	It is enabled by a high to low transition on CS. CS must remain low for the duration of any command sequence.
SI_1	Serial Data input	To transfer data serially input. Inputs are latched on the Port mode = 2′b01 or 2′b10.
SO_1	Serial Data Output	To transfer data serially output.

		Data is shifted out on the Port mode = 2′b01 or 2′b10.
SI_2 SI_3	Serial Data input	To transfer data serially input. Inputs are only latched on the Port mode = 2′b10.
SO_2 SO_3	Serial Data Output	To transfer data serially output. Data is only shifted out on the Port mode = 2′b10.

6.17.1 TIC Operation

The serial data input (SI) is sampled at the rising edge of the SCK clock signal and the serial data output (SO) is driven after the falling edge of the SCK clock signal.

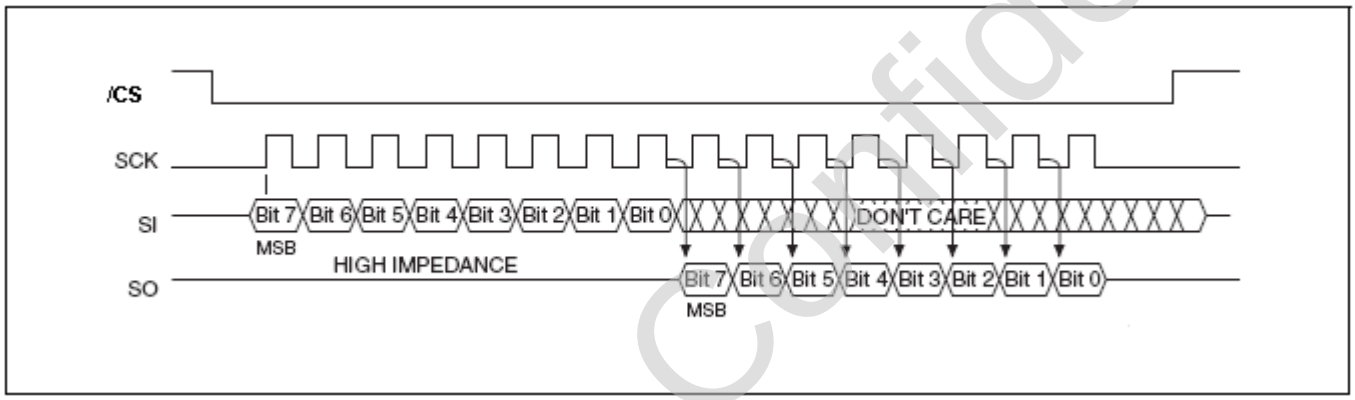


Figure 6.17-2: Serial Interface Protocol

6.17.2 Status Register

The status register provides status on whether the TIC is available for any read or write operation and data input and output port selection.

Table 6.17-2: Status Register Description

Bit	Name	Function	Default at Power-up	Read / Write
7	Busy	1= Internal Read/Write operation is in progress 0= No internal Read/Write operation is in progress	0	R
6:5	Port mode	00 = 1 input and 1 output port support 01 = 2 input and 2 output ports support 10 = 4 input and 4 output ports support	00	R/W

4	Instruction Error	1 = Un-correct Instruction found (write "1" clear) 0 = Correct Instruction	0	R/W
3	Fail	1 = Last sequence not complete (write "1" clear) 0 = Last sequence complete	0	R/W
2:0	Program mode	001 = byte write 010 = one word data write (Auto Address Increment) 011 = one word data read (Auto Address Increment) 100 = 4 words data write (Auto Address Increment) 110 = 4 words data read (Auto Address Increment)	001	R/W

Busy

The Busy bit determines whether there is an internal program operation in progress. A "1" for the Busy bit indicates the TIC is busy with an operation in progress. A "0" indicates the TIC is ready for the next valid operation.

Port mode

The default is one input port (SI) and one output port (SO). To change mode by instruction 01h write port mode status to status register. A 2-bit "01" indicates two input ports (SI, SI_1) and two output ports (SO, SO_1) for serial data input or output. A 2-bit "10" indicates 4 input ports (SI, SI_1, SI_2, SI_3) and 4 output ports (SO, SO_1, SO_2, SO_3) for serial data input or output.

Instruction Error

The instruction error bit is going to "1" once any instruction did not include in Table 3. The error did not interrupt the next instruction sequence. Use instruction 01h to write "1" to clear the error bit.

Fail

A 1-bit "1" indicates CS going to high while the last instruction sequence did not complete. A 1-bit "0" indicates the last instruction sequence is complete. Use instruction 01h to write "1" to clear the fail bit.

Program mode

The first action for TIC operation is consensus with the program mode. The default is at byte write mode. To change mode by instruction 01h write program mode status to status register. A 3-bit "010" indicates write one word data to AHB for each instruction D2h. A 3-bit "011" indicates read one word data from AHB for each instruction D3h. A 3-bit "100" indicates write 4 words data to AHB for each instruction 92h. A 3-bit "110" indicates read 4 words data from AHB for each instruction 93h.

Instruction Protocol

Programmer should check status if there is internal busy or instruction error before any instruction send. Define port mode and program mode before READ/WRITE instruction send. Write address instruction before READ/WRITE instruction if the program mode is Auto Address Increment. Follow the Figure 6.17-3 ~ 11 to each instruction sequence and data input/output for detail. The dual input and dual output bits are data dependent on Auto Address Increment program mode. The 4 input and 4 output bits are same as the dual bits. Others are all in one bit port mode.

Byte write protocol: 05h -> 01h -> 02h

Write word protocol: 05h -> 01h -> ADh -> 05h -> D2h -> 05h -> D2h ---

Read word protocol: 05h -> 01h -> ADh -> 05h -> D3h -> 05h -> D3h ---

Write 4 words protocol: 05h -> 01h -> ADh -> 05h -> 92h -> 05h -> 92h ---

Read 4 words protocol: 05h -> 01h -> ADh -> 05h -> 93h -> 05h -> 93h ---

Read-ID : 05h -> 91h

Table 6.17-3 Operation Instructions (One input and one output port)

INSTRUCTION NAME	BYTE 1 CODE	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 17
Read Status Register	05h	(S7-S0)					
Write Status Register	01h	(D7-D0)					
Read Word Data	D3h	(D7-D0)	(D15-D8)	(D23-D16)	(D31-D24)		
Write Word Data	D2h	(D7-D0)	(D15-D8)	(D23-D16)	(D31-D24)		
Read 4 Words Data	93h	(D7-D0)	(D15-D8)	(D23-D16)	(D31-D24)	continuous	(D127-D120)
Write 4 Words Data	92h	(D7-D0)	(D15-D8)	(D23-D16)	(D31-D24)	continuous	(D127-D120)
Write Byte Data	02h	A31-A24	A23-A16	A15-A8	A7-A0	(D7-D0)	
Write Address	ADh	A31-A24	A23-A16	A15-A8	A7-A0		
Manufacturer/ Device ID	91h	00h	00h	(M7-M0) 55h	(ID7-ID0) 92h		

Note: CS Operation Characteristics

	67 MHz		100 MHz		133 MHz	
Parameter	Min	Max	Min	Max	Min	Max
CS High Time	3 tHCLK		3 tHCLK		3 tHCLK	

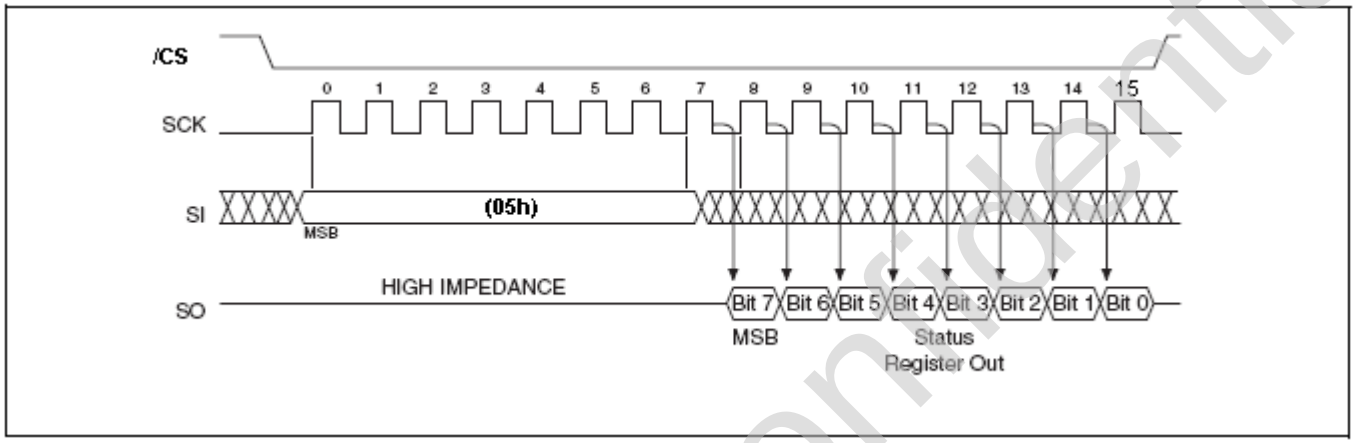


Figure 6.17-3 Read Status Register Sequence (05h)

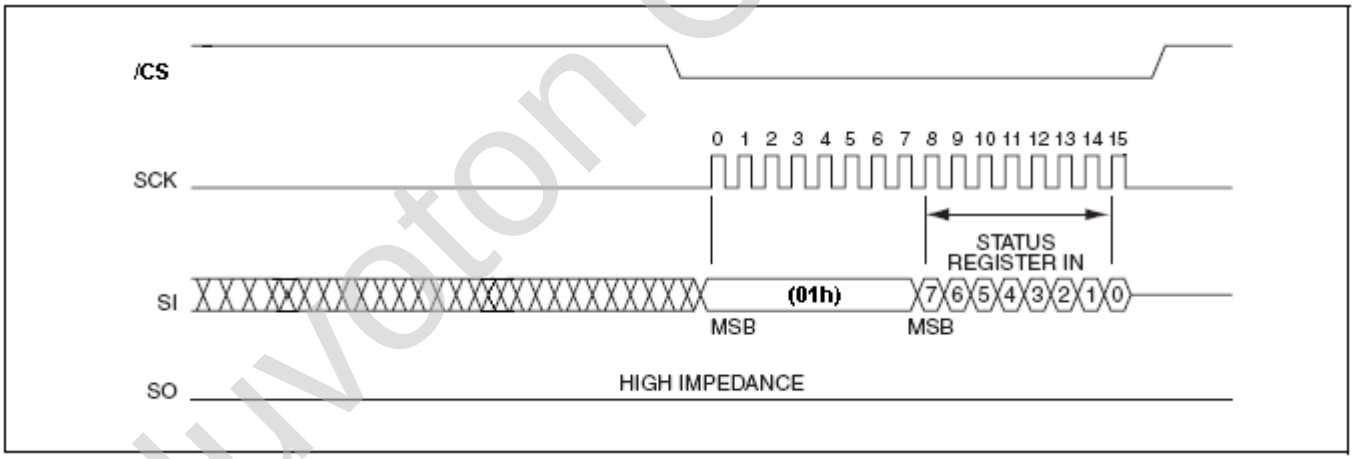


Figure 6.17-4 Write Status Register Sequence (01h)

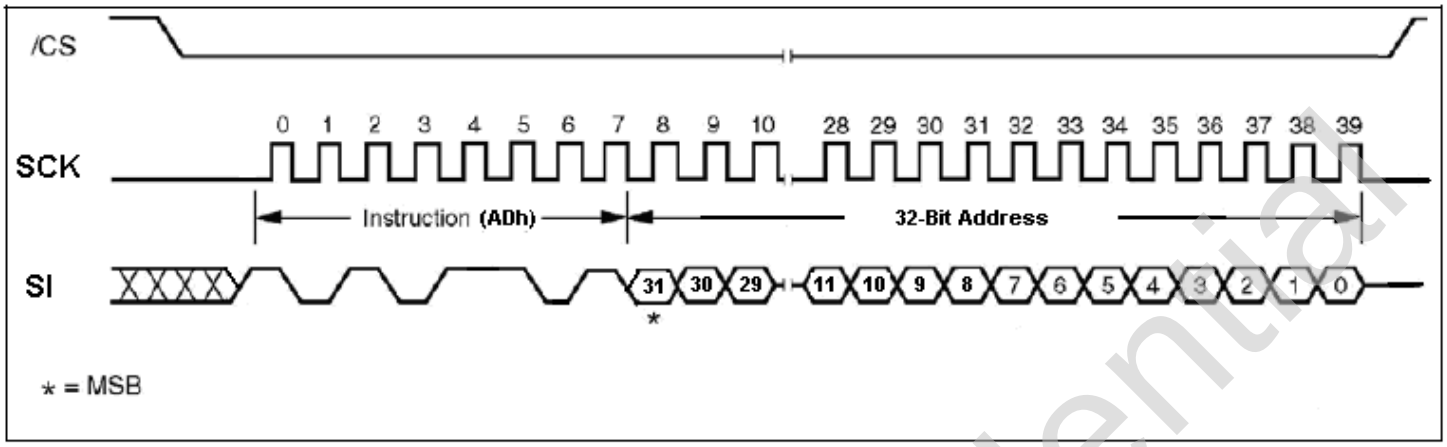


Figure 6.17-5 Auto Address Increment Start Address Sequence (ADh)

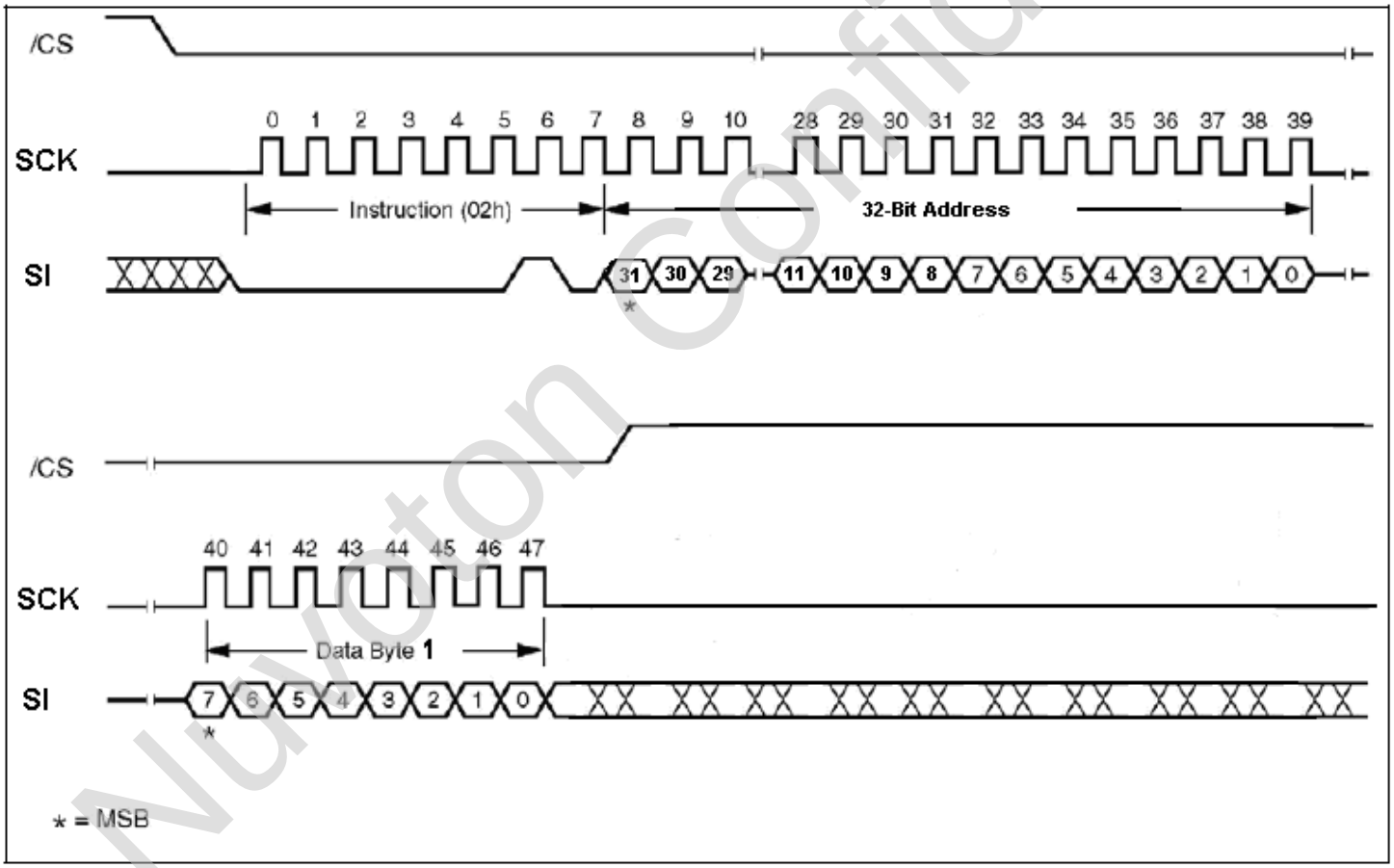


Figure 6.17-6 Byte Write Sequence (02h)

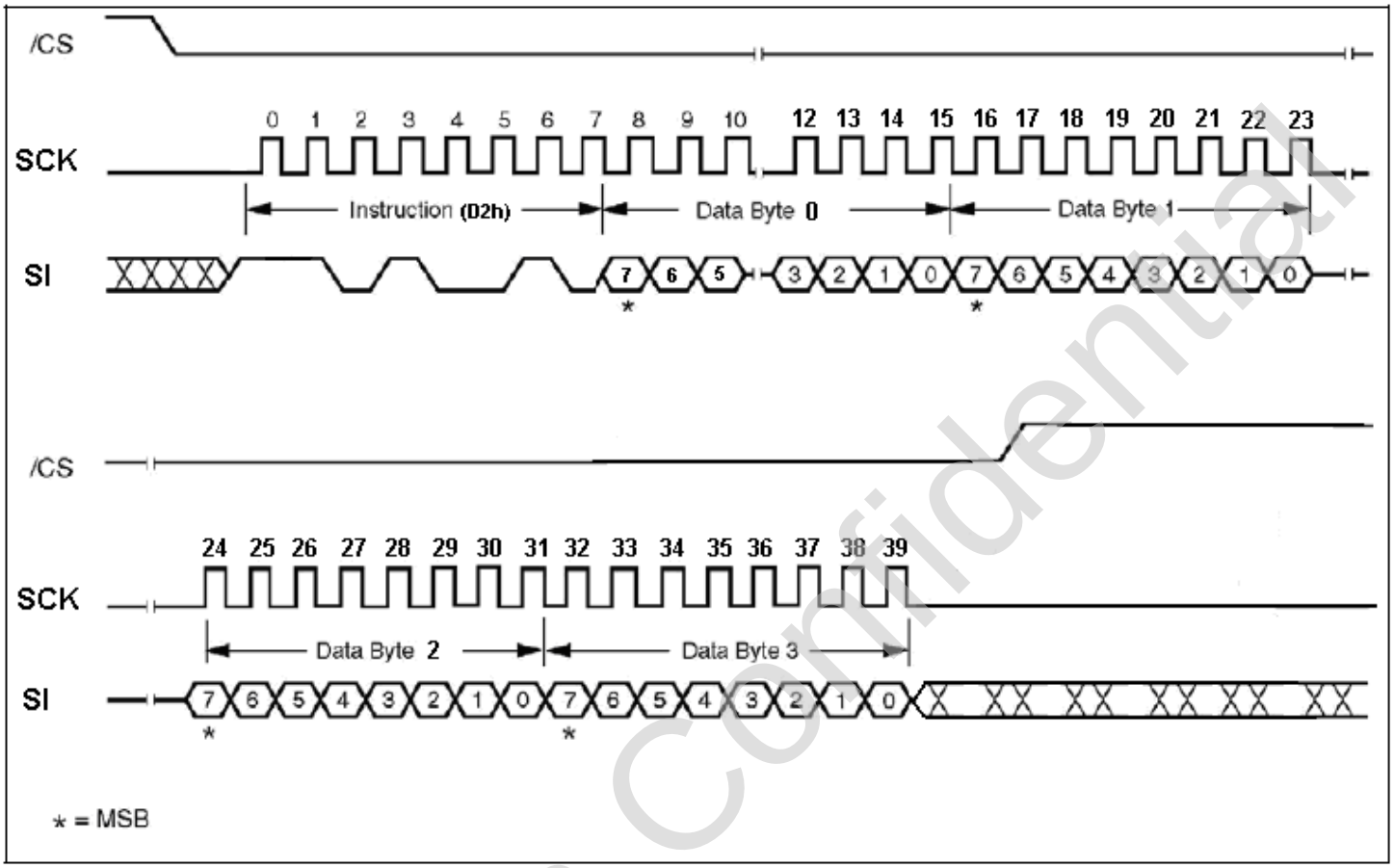


Figure 6.17-7 One Word Data Write Sequence (D2h) (Default Port Mode)

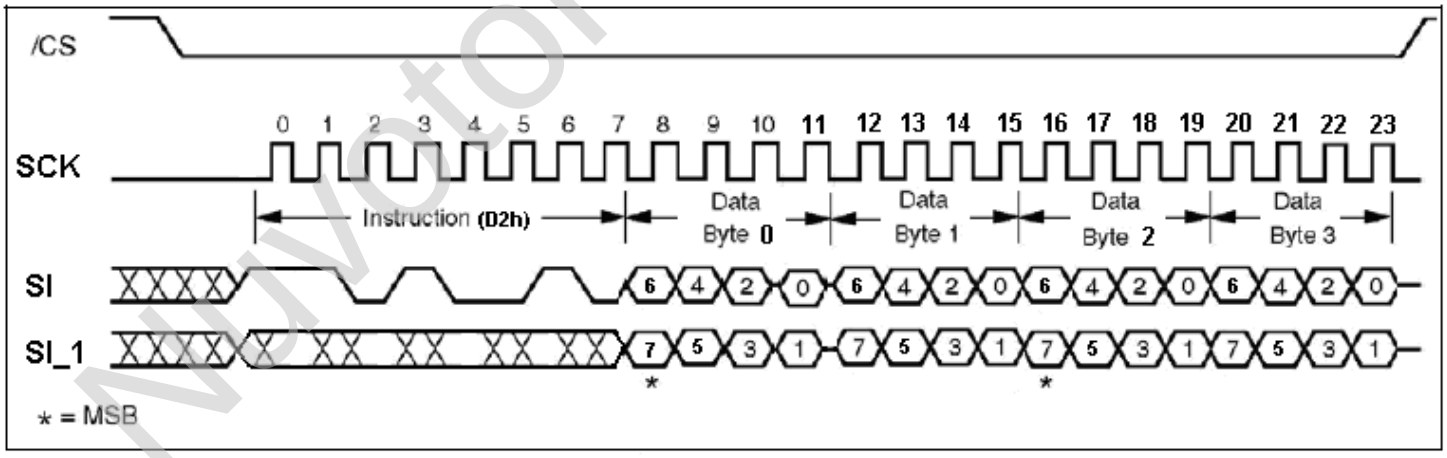


Figure 6.17-8 One Word Data Write Sequence (D2h) (Port Mode "01")

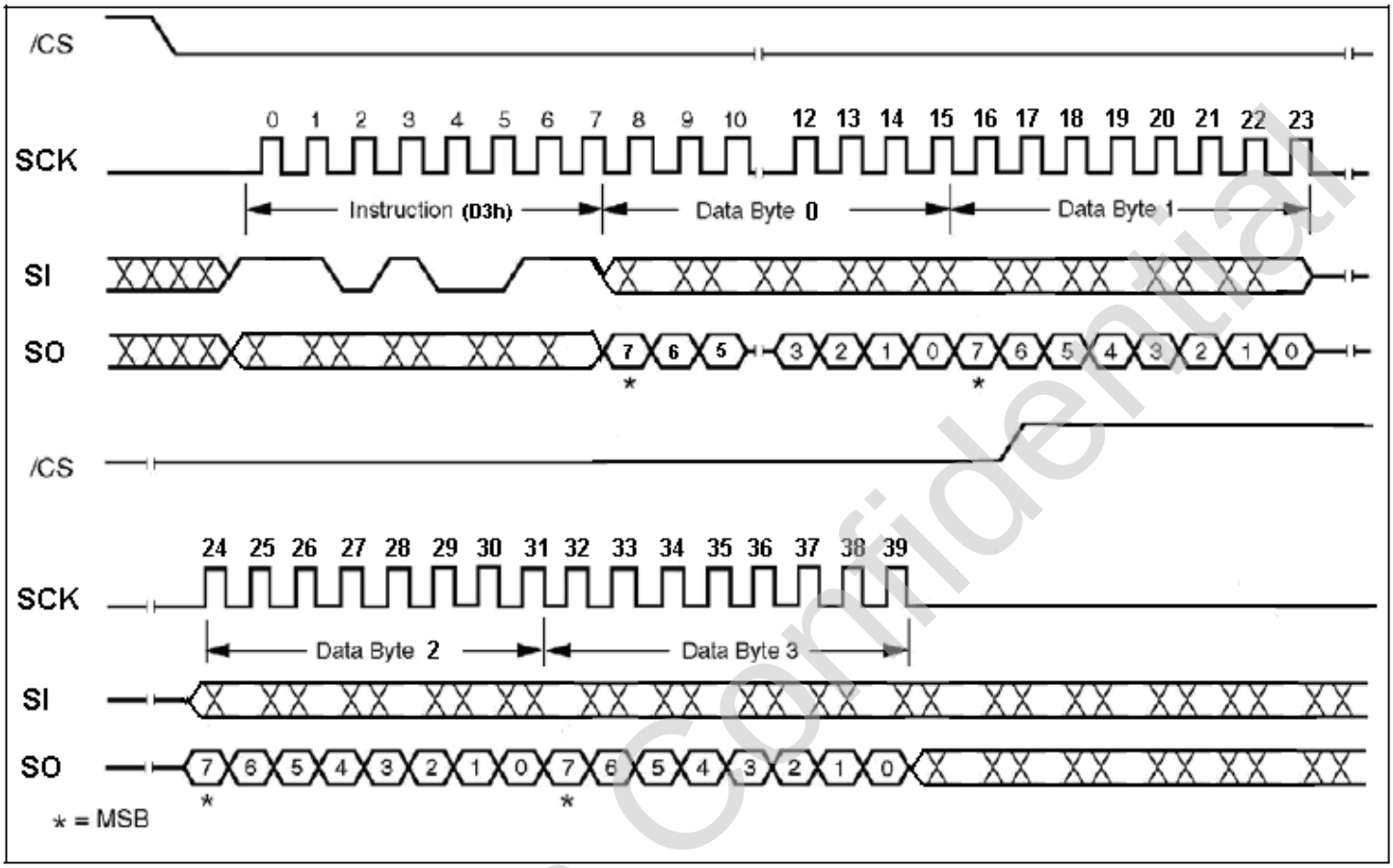


Figure 6.17-9 One Word Data Read Sequence (D3h)

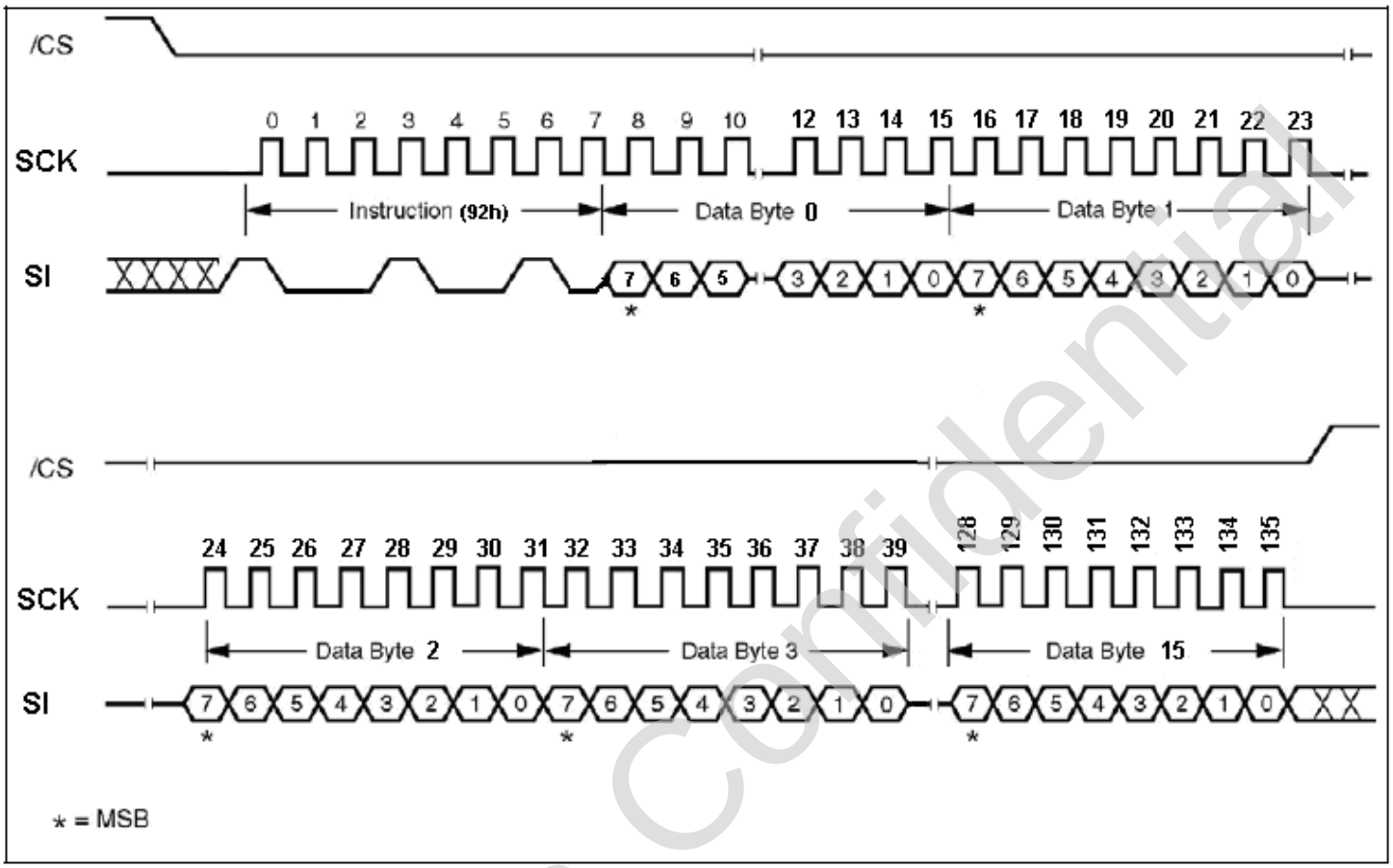


Figure 6.17-10 Words Data Write Sequence (92h)

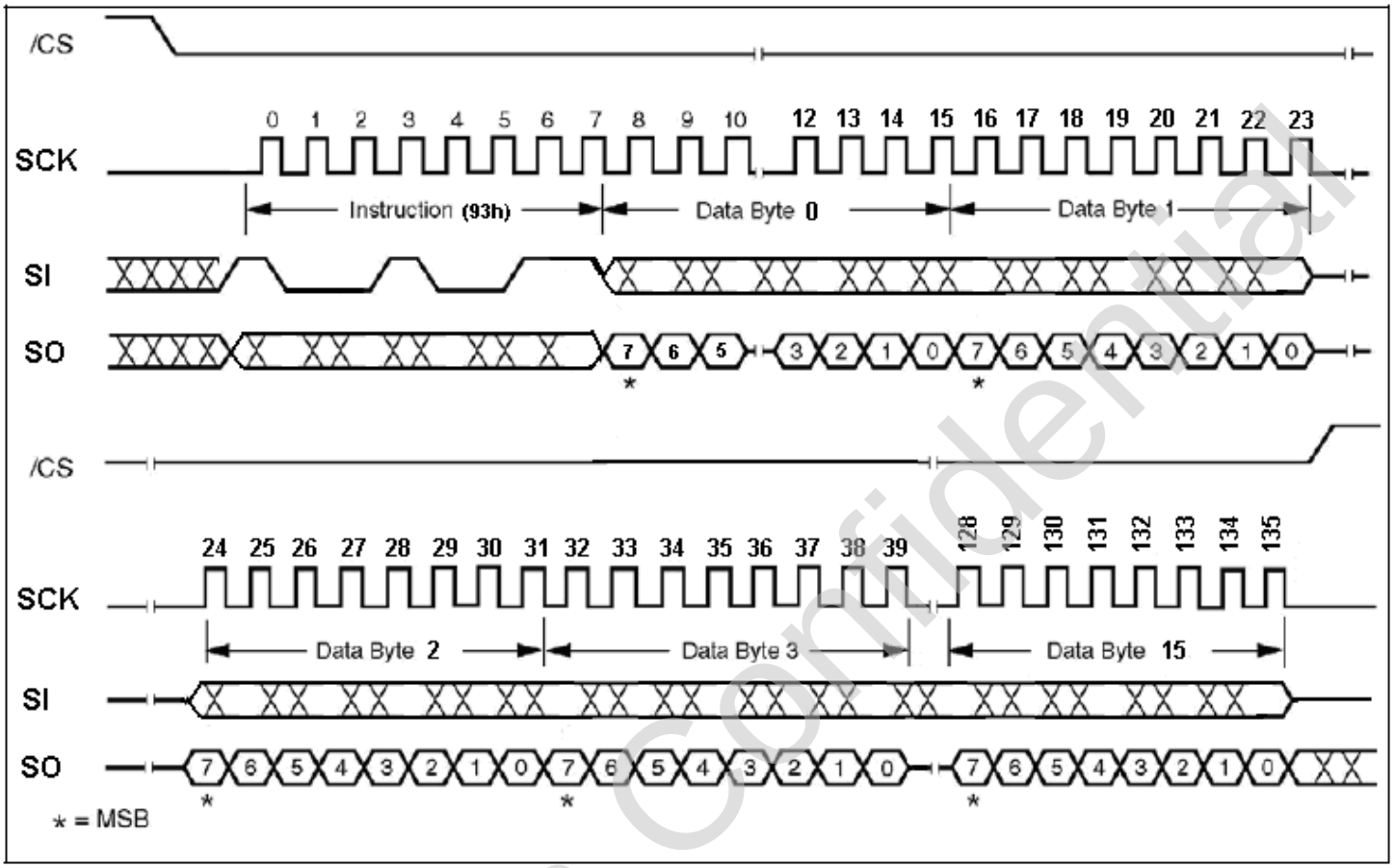


Figure 6.17-11 Words Data Read Sequence (93h)

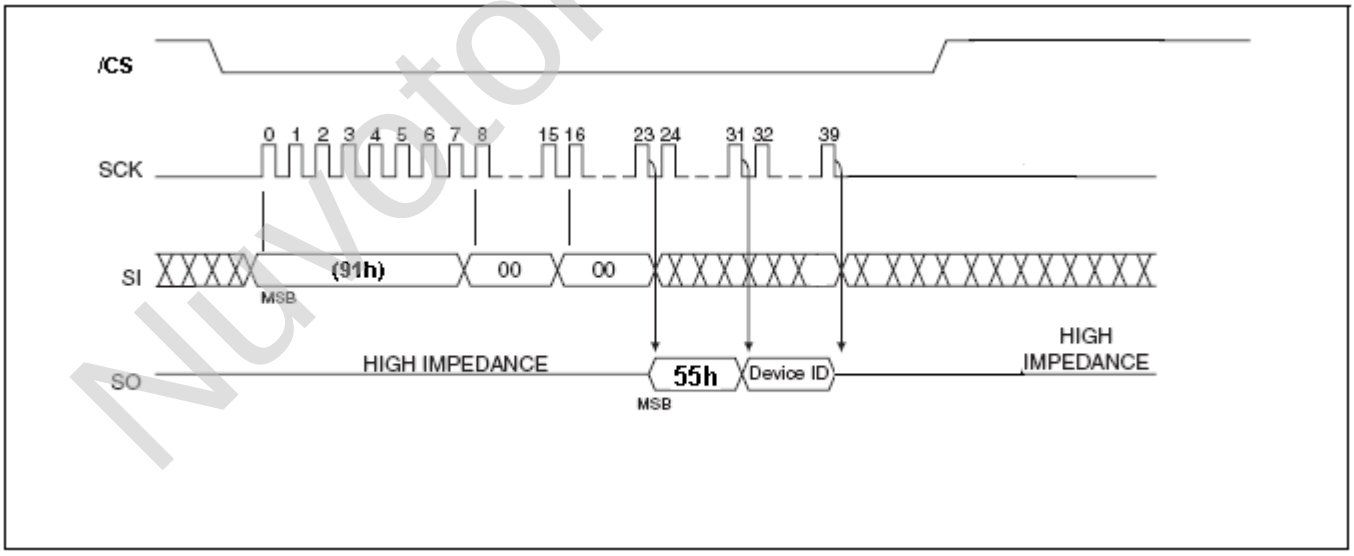


Figure 6.17-12 Read-ID Sequence (91h)

Table 6.17-4 Operation Instructions (Two input and two output ports)

INSTRUCTION NAME	BYTE 1 CODE	BYTE 2 (bit 0,1,2,3)	BYTE 2 (bit 4,5,6,7)	BYTE 3 (bit 0,1,2,3)	BYTE 3 (bit 4,5,6,7)	BYTE 4 (bit 0,1,2,3)	BYTE 9 (bit 4,5,6,7)
Read Word Data	D3h	(D6,4,2,0) (D7,5,3,1)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous		
Write Word Data	D2hz	(D6,4,2,0) (D7,5,3,1)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous		
Read 4 Words Data	93h	(D6,4,2,0) (D7,5,3,1)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous	continuous	(D127-D120)
Write 4 Words Data	92h	(D6,4,2,0) (D7,5,3,1)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous	continuous	(D127-D120)

Table 6.17-5 Operation Instructions (4 input and 4 output ports)

INSTRUCTION NAME	BYTE 1 CODE	BYTE 2 (bit 0,1)	BYTE 2 (bit 2,3)	BYTE 2 (bit 4,5)	BYTE 2 (bit 6,7)	BYTE 3 (bit 0,1)	BYTE 5 (bit 6,7)
Read one Word Data	D3h	(D4,D0) (D5,D1) (D6,D2) (D7,D3)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous		
Write one Word Data	D2h	(D4,D0) (D5,D1) (D6,D2) (D7,D3)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous		
Read 4 Words Data	93h	(D4,D0) (D5,D1) (D6,D2) (D7,D3)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous	continuous	(D127-D120)
Write 4 Words Data	92h	(D4,D0) (D5,D1) (D6,D2) (D7,D3)	(D15-D8) as previous	(D23-D16) as previous	(D31-D24) as previous	continuous	(D127-D120)

6.18 Advanced Interrupt Controller

6.18.1 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 W55FA95 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 1. 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 W55FA95 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 W55FA95 is put in the test mode, all interrupt sources must be configured as positive-edge triggered.

6.18.2 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

6.18.3 Interrupt Sources

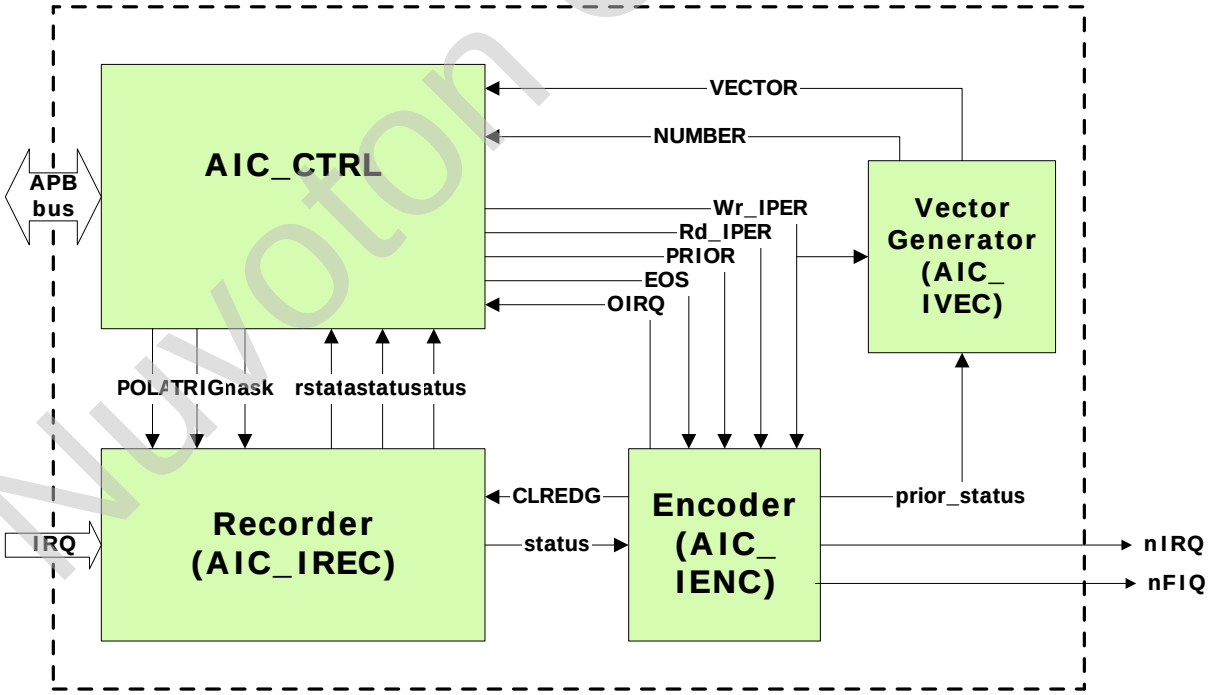
The following table lists all interrupts from various peripheral interface modules or external devices.

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/GPIO_INT3	SCR2[7:0]	GPIO Interrupt 2 and GPIO Interrupt 3	Low
5	IPSEC_INT	SCR2[15:8]	AES Interrupt	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	VIN_INT	SCR3[15:8]	Video In Interrupt	Low
10	OVG_INT	SCR3[23:16]	Open VG Interrupt	Low
11	GVE_GE_INT	SCR3[31:24]	Graphics Video Engine Interrupt	Low
12	GVE_VPE_INT	SCR4[7:0]	Graphics Video Engine 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	SPIMS_INT	SCR6[7:0]	SPI Master/Slave Controller Interrupt	Low
21	SPIM_INT	SCR6[15:8]	SPI Master Controller 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	VDE_INT	SCR7[31:24]	Video Decoder Interrupt	Low
28	KPI_INT	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	Reserved	KPI	VDE	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	SPIM	SPIMS	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	VPE	GE	OVG	VIN	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	IPSEC	GPIO2_3	GPIO1	GPIO0	WDT	---

6.18.4 AIC Block Diagram



AIC Functional Block Diagram

6.18.5 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.

6.18.6 AIC Registers Map

Register	Address	R/W	Description	Reset Value
AIC_BA = 0x				
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

6.18.7 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]	ISR*x*:	**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 W55FA95 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]	SSC*x*	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 W55FA95 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]	SCC*x*	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.

Bits	Descriptions	
[31:0]	---	---

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.

6.19 General Purpose I/O

6.19.1 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 SMIC universal standard I/O Cell Library. And the following figures illustrate the control mechanism to achieve the GPIO functions.

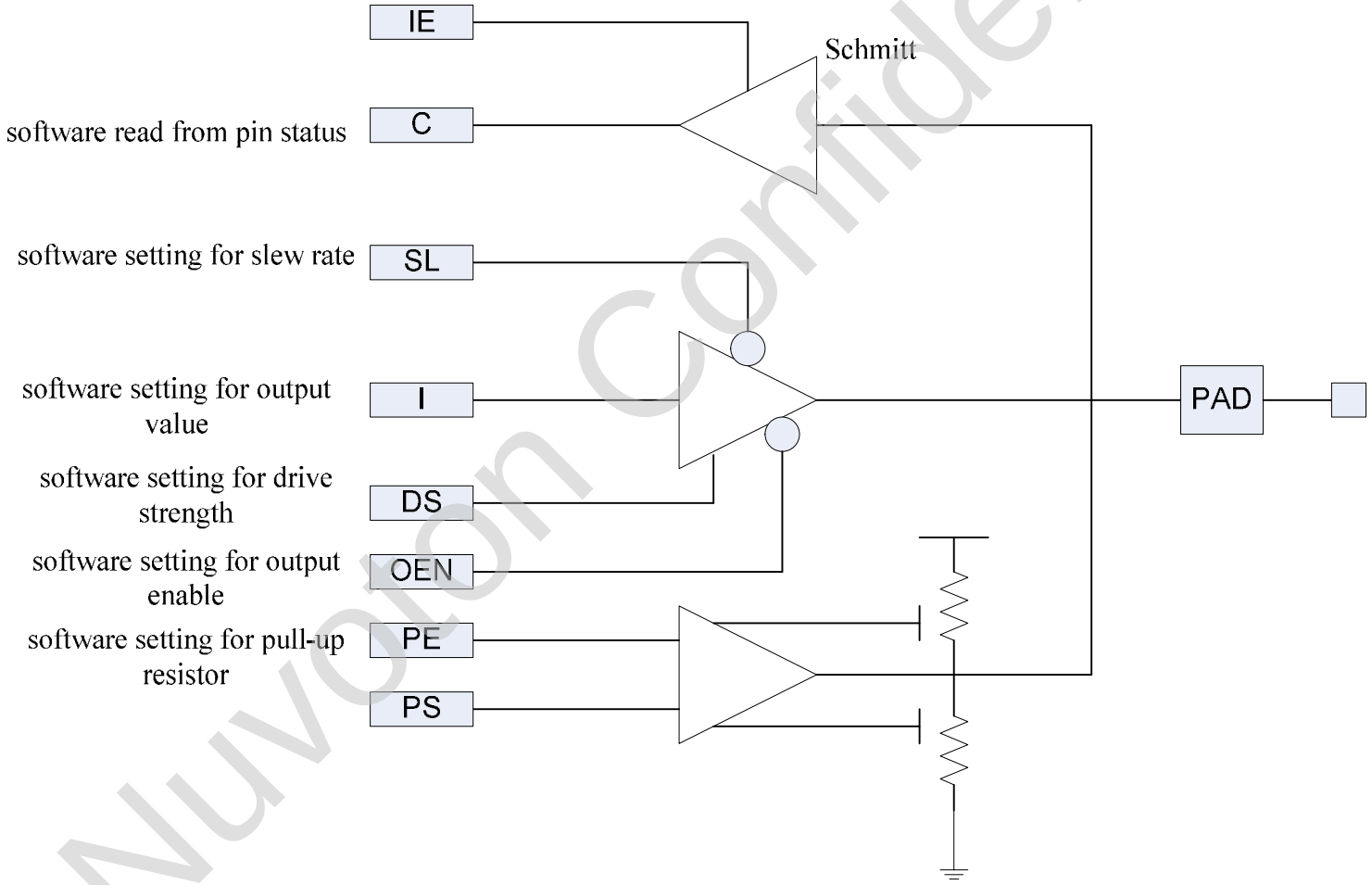


Figure 6.19-1 GPIO: Input/Output Port with Program Controlled Weakly Pull-High, Schmitt-Trigger Input, Drive Strength, Slew Rate(PBSCUDL0408R)

6.19.2 GPIO Control 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
GPIOG_OMD	GP_BA+0x50	R/W	GPIO Port G Bit Output Mode Enable	0x0000_0000
GPIOG_PUEN	GP_BA+0x54	R/W	GPIO Port G Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOG_DOUT	GP_BA+0x58	R/W	GPIO Port G Data Output Value	0x0000_0000
GPIOG_PIN	GP_BA+0x5C	R	GPIO Port G Pin Value	0xXXXX_XXXX
GPIOH_OMD	GP_BA+0x60	R/W	GPIO Port H Bit Output Mode Enable	0x0000_0000

GPIOH_PUEN	GP_BA+0x64	R/W	GPIO Port H Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOH_DOUT	GP_BA+0x68	R/W	GPIO Port H Data Output Value	0x0000_0000
GPIOH_PIN	GP_BA+0x6C	R	GPIO Port H 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
IRQSRCGPG	GP_BA+0x94	R/W	GPIO Port G IRQ Source Grouping	0xFFFF_FFFF
IRQSRCGPH	GP_BA+0x98	R/W	GPIO Port H 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
IRQENGPG	GP_BA+0xB4	R/W	GPIO Port G Interrupt Enable	0x0000_0000
IRQENGPH	GP_BA+0xB8	R/W	GPIO Port H 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
IRQLHGPG	GP_BA+0xE4	R	GPIO Port G Interrupt Latch Value	0x0000_0000
IRQLHGPH	GP_BA+0xE8	R	GPIO Port H 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 and GPIO Port D	0x0000_0000
IRQTGSRC2	GP_BA+0xF8	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port E and GPIO Port G	0x0000_0000

IRQTGSRC3	GP_BA+0xFC	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port H	0x0000_0000

6.19.3 GPIO Control 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
GPIOG_OMD	GP_BA+0x50	R/W	GPIO Port G Bit Output Mode Enable	0x0000_0000
GPIOH_OMD	GP_BA+0x60	R/W	GPIO Port H 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 Ÿ 1 = GPIO port [A/B/C/D/E/G/H] bit [n] output mode is enabled, the bit value contained in the corresponding bit [n] of GPIO[A/B/C/D/E/G/H]_DOUT is driven on the pin. Ÿ 0 = GPIO port [A/B/C/D/E/G/H] bit [n] output mode is disabled, the corresponding pin is in INPUT mode. Ps: 1.GPG[0] and GPG[1] are always input mode! 2.GPIO port H is only GPH[0] can work !	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
GPIOG_PUEN	GP_BA+0x54	R/W	GPIO Port G Bit Pull-up Resistor Enable	0x0000_FFFF
GPIOH_PUEN	GP_BA+0x64	R/W	GPIO Port H 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 Ÿ 1 = GPIO port [A/B/C/D/E/G/H] bit [n] pull-up resistor is enabled. Ÿ 0 = GPIO port [A/B/C/D/E/G/H] bit [n] pull-up resistor is disabled. Ps: 1.GPG[0] and GPG[1] are always input mode! 2.GPIO port H is only GPH[0] can work !	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
GPIOG_DOUT	GP_BA+0x58	R/W	GPIO Port G Data Output Value	0x0000_0000
GPIOH_DOUT	GP_BA+0x68	R/W	GPIO Port H 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 Ÿ 1 = GPIO port [A/B/C/D/E/G/H] bit [n] will drive High if the corresponding output mode enabling bit is set. Ÿ 0 = GPIO port [A/B/C/D/E/G/H] bit [n] will drive Low if the corresponding output mode enabling bit is set. Ps: 1.GPG[0] and GPG[1] are always input mode! 2.GPIO port H is only GPH[0] can work !	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
GPIOG_PIN	GP_BA+0x5C	R	GPIO Port G Pin Value	0x0000_XXXX
GPIOH_PIN	GP_BA+0x6C	R	GPIO Port H 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/G/H] Pin Values Each bit of this register reflects the value of each GPIO pin. Ps: 1.GPG[0] and GPG[1] are always input mode! 2.GPIO port H is only GPH[0] can work !	0xXXXX

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

<table>
<tr>
<td>[7:4]</td>
<td>DBCLKSEL</td>
<td>Debounce sampling cycle selection
<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>
</td>
<td>0x0</td>
</tr>
<tr>
<td>[3:0]</td>
<td>DBEN</td>
<td>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</td>
<td>0x0</td>
</tr>
</table>

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	0x0000_0000

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	0x0000_0000

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	0x0000_0000

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 GPDx 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	0x0000_0000

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 GPEx 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.

IRQ Source Grouping (IRQSRCGPG)

Register	Address	R/W	Description	Reset Value
IRQSRCGPG	GP_BA+0x94	R/W	GPIO Port G IRQ Source Grouping	0x0000_0000

31	30	29	28	27	26	25	24
GPG15SEL		GPG14SEL		GPG13SEL		GPG12SEL	
23	22	21	20	19	18	17	16
GPG11SEL		GPG10SEL		GPG9SEL		GPG8SEL	
15	14	13	12	11	10	9	8
GPG7SEL		GPG6SEL		GPG5SEL		GPG4SEL	
7	6	5	4	3	2	1	0
GPG3SEL		GPG2SEL		GPG1SEL		GPG0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPGxSEL	Selection for GPGx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source Ps: 1.GPG[0] and GPG[1] are always input mode! 2.GPIO port H is only GPH[0] can work !	0x0

Where x=0~15.

GPGxSEL = 0, GPGx pin is grouped as one of interrupt sources to IRQ0.

1, GPGx pin is grouped as one of interrupt sources to IRQ1.

2, GPGx pin is grouped as one of interrupt sources to IRQ2.

3, GPGx pin is grouped as one of interrupt sources to IRQ3.

IRQ Source Grouping (IRQSRCGPG)

Register	Address	R/W	Description	Reset Value
IRQSRCGPH	GP_BA+0x98	R/W	GPIO Port H IRQ Source Grouping	0x0000_0000

31	30	29	28	27	26	25	24
GPH15SEL		GPH14SEL		GPH13SEL		GPH12SEL	
23	22	21	20	19	18	17	16
GPH11SEL		GPH10SEL		GPH9SEL		GPH8SEL	
15	14	13	12	11	10	9	8
GPH7SEL		GPH6SEL		GPH5SEL		GPH4SEL	
7	6	5	4	3	2	1	0
GPH3SEL		GPH2SEL		GPH1SEL		GPH0SEL	

Bits	Descriptions		Default
[2x+1:2x]	GPHxSEL	Selection for GPHx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source Ps: 1.GPG[0] and GPG[1] are always input mode! 2.GPIO port H is only GPH[0] can work !	0x0

Where x=0~15.

GPHxSEL = 0, GPHx pin is grouped as one of interrupt sources to IRQ0.

1, GPHx pin is grouped as one of interrupt sources to IRQ1.

2, GPHx pin is grouped as one of interrupt sources to IRQ2.

3, GPHx 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.

GPIO G Interrupt Enable (IRQENGPE)

Register	Address	R/W	Description	Reset Value
IRQENGPG	GP_BA+0xB4	R/W	GPIO Port G Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PG15ENR	PG14ENR	PG13ENR	PG12ENR	PG11ENR	PG10ENR	PG9ENR	PG8ENR
23	22	21	20	19	18	17	16
PG7ENR	PG6ENR	PG5ENR	PG4ENR	PG3ENR	PG2ENR	PG1ENR	PG0ENR
15	14	13	12	11	10	9	8
PG15ENF	PG14ENF	PG13ENF	PG12ENF	PG11ENF	PG10ENF	PG9ENF	PG8ENF
7	6	5	4	3	2	1	0
PG7ENF	PG6ENF	PG5ENF	PG4ENF	PG3ENF	PG2ENF	PG1ENF	PG0ENF

Bits	Descriptions		Default
[x]	PGxENF	Enable GPGx input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPG register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PGxENR	Enable GPGx input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPG register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PGxENF and PGxENR can be set "1" at the same time.

NOTE1: In normal operation mode, for each pin, PGxENF and PGxENR 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.

NOTE3:

1.GPG[0] and GPG[1] are always input mode!

2.GPIO port H is only GPH[0] can work !

GPIO H Interrupt Enable (IRQENGPE)

Register	Address	R/W	Description	Reset Value
IRQENGPH	GP_BA+0xB4	R/W	GPIO Port H Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
PH15ENR	PH14ENR	PH13ENR	PH12ENR	PH11ENR	PH10ENR	PH9ENR	PH8ENR
23	22	21	20	19	18	17	16
PH7ENR	PH6ENR	PH5ENR	PH4ENR	PH3ENR	PH2ENR	PH1ENR	PH0ENR
15	14	13	12	11	10	9	8
PH15ENF	PH14ENF	PH13ENF	PH12ENF	PH11ENF	PH10ENF	PH9ENF	PH8ENF
7	6	5	4	3	2	1	0
PH7ENF	PH6ENF	PH5ENF	PH4ENF	PH3ENF	PH2ENF	PH1ENF	PH0ENF

Bits	Descriptions		Default
[x]	PHxENF	Enable GPHx input falling edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPH register determines which IRQn (n=0~3) is the destination.	0x0
[x+16]	PHxENR	Enable GPHx input rising edge to trigger one of interrupt sources IRQ0~IRQ3. IRQSRCGPH register determines which IRQn (n=0~3) is the destination.	0x0

Where x=0~15.

PHxENF and PHxENR can be set "1" at the same time.

NOTE1: In normal operation mode, for each pin, PHxENF and PHxENR 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.

NOTE3:

1.GPG[0] and GPG[1] are always input mode!

2.GPIO port H is only GPH[0] can work !

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/GPG/GPH to IRQLHGPA/IRQLHGPB/IRQLHGPC/IRQLHGPD/IRQLHGPE/ IRQLHGPG/ IRQLHGPH 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
IRQLHGPG	GP_BA+0XE4	R	GPIO Port G Interrupt Latch Value	0x0000_0000
IRQLHGPH	GP_BA+0xE8	R	GPIO Port H 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 or G or H, x=0~15

NOTE: When a latched pin value is `0′, there will be 2 meanings: either the pin′s input is recogized 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 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 and GPIO Port G	0x0000_0000

31	30	29	28	27	26	25	24
PG15TG	PG14TG	PG13TG	PG12TG	PG11TG	PG10TG	PG9TG	PG8TG
23	22	21	20	19	18	17	16
PG7TG	PG6TG	PG5TG	PG4TG	PG3TG	PG2TG	PG1TG	PG0TG
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]	PExTG	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
[x+16]	PGxTG	When this bit is read as "1", it indicates GPGx 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 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).

IRQ Interrupt Trigger Source 3 (IRQTGSRC3)

Register	Address	R/W	Description	Reset Value
IRQTGSRC3	GP_BA+0xFC	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port H	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
PH15TG	PH14TG	PH13TG	PH12TG	PH11TG	PH10TG	PH9TG	PH8TG
7	6	5	4	3	2	1	0
PH7TG	PH6TG	PH5TG	PH4TG	PH3TG	PH2TG	PH1TG	PH0TG

Bits	Descriptions		Default
[x]	PHxTG	When this bit is read as "1", it indicates GPHx 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 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).

6.20 TIMER Controller

6.20.1 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.

The general TIMER Controller includes the following 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 MHz) * (2^8) * (2^32), if TCLK = 25 MHz**

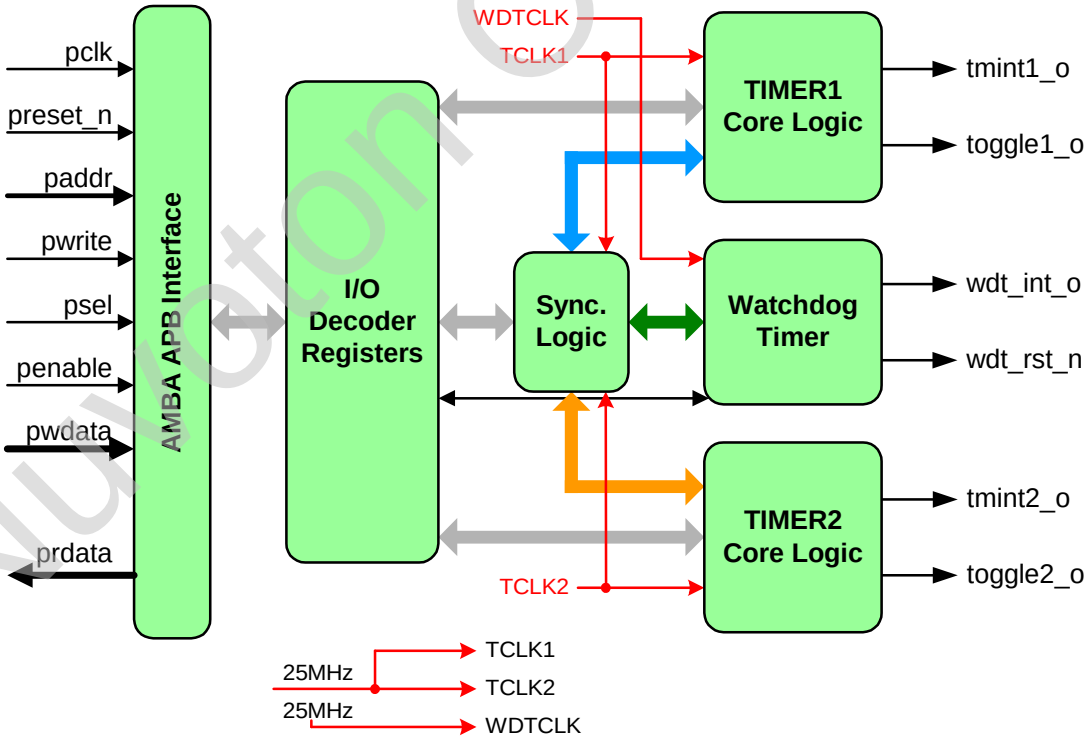


Figure 6.20-1 TIMER Block Diagram

6.20.2 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.

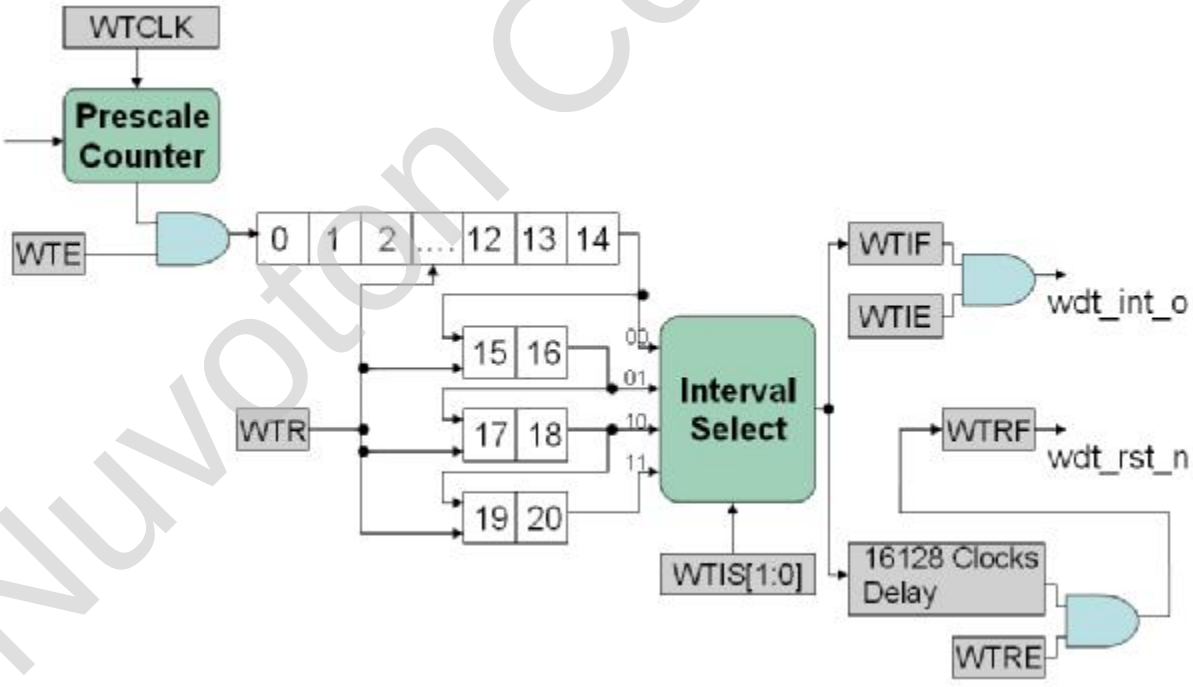
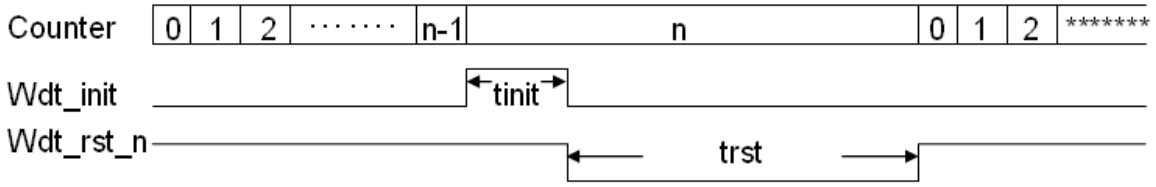


Figure 6.20-2 Watchdog Timer Block Diagram



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 6.20-3 Watchdog Timer Timing Diagram

6.20.3 Timer Control Registers 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 = 0xB8002000				
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

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: 00 — Timer Operating Mode: The timer is operating in the one-shot mode. The associated

		<table><tr><td></td><td>interrupt signal is generated once (if IE is enabled) and CEN is automatically cleared then.</td></tr><tr><td>01</td><td>The timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if IE is enabled).</td></tr><tr><td>10</td><td>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.</td></tr><tr><td>11</td><td>The timer is operating in the uninterrupted mode. The associated interrupt signal is generated when TDR = TICR (if IE is enabled) .</td></tr></table>
[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

6.21 Real Time Clock (RTC)

6.21.1 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 or internal oscillator . 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.

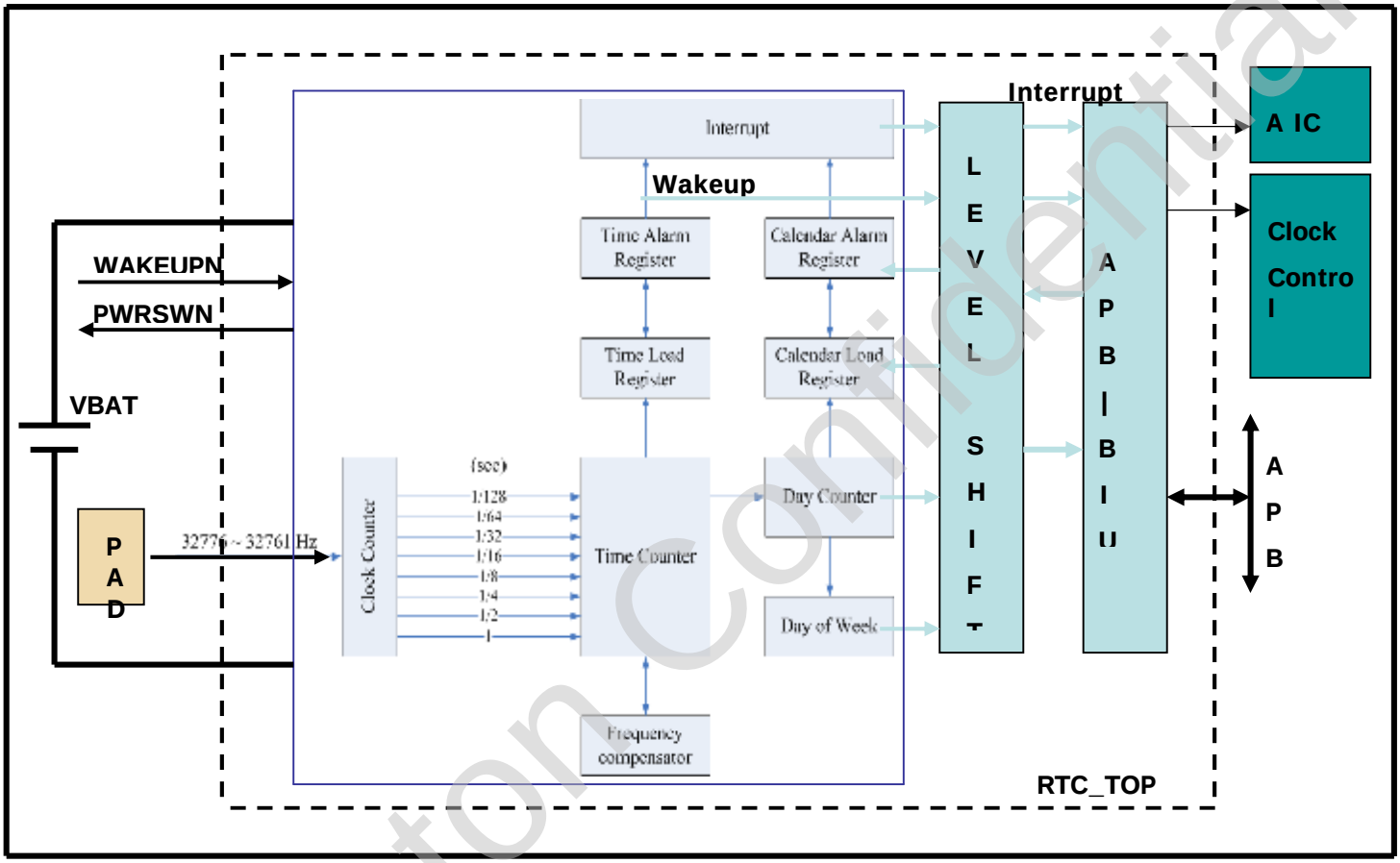
The wakeup signal is used to wake the system up from sleep mode.

6.21.2 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.

6.21.3 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.

6.21.4 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,**

For example, Load CLR as 201a (year), 13 (month), 00 (day), or CLR does not match with DWR, etc.

3. **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

4. **FCR Calibration :**

Example 1(use external crystal 32768Hz):

Frequency counter measurement : 32773.65Hz (> 32768 Hz)

Integer part: 32773 => 0x8005

FCR_int = 0x8005

Fraction part: 0.65 x 60 = 39 => 0x27

FCR_fra = 0x27

Example 2(use internal oscillator):

When PCLK = 50MHz, the internal oscillator =32768Hz,

PCLK period = 1/50MHz = 20ns,

Internal oscillator period = 1/32768Hz = 30500ns

So, OSC_32K_CNT = 30500ns/20ns = 1525

Then, software read OSC_32K_CNT and calculate 1/(1525*20ns) to get

FCR value 32768 = 0x8000

5. 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.

6.21.5 System Power Control Flow

6.21.5.1 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 whenX2=1
PWCE is open drain output X1, internal pull-up X2, it is R/W able				

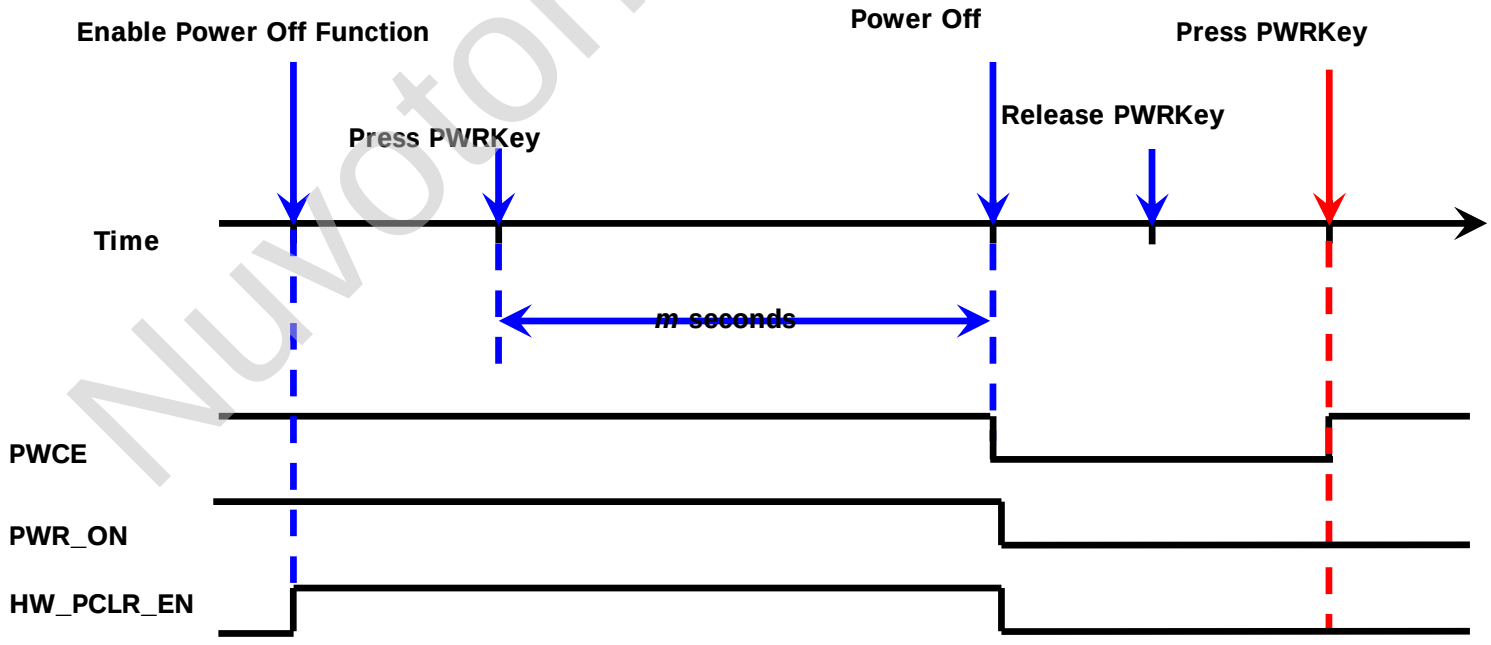
There is Interrupt from key be pressed

6.21.5.2 Force system Power Off Control Flow

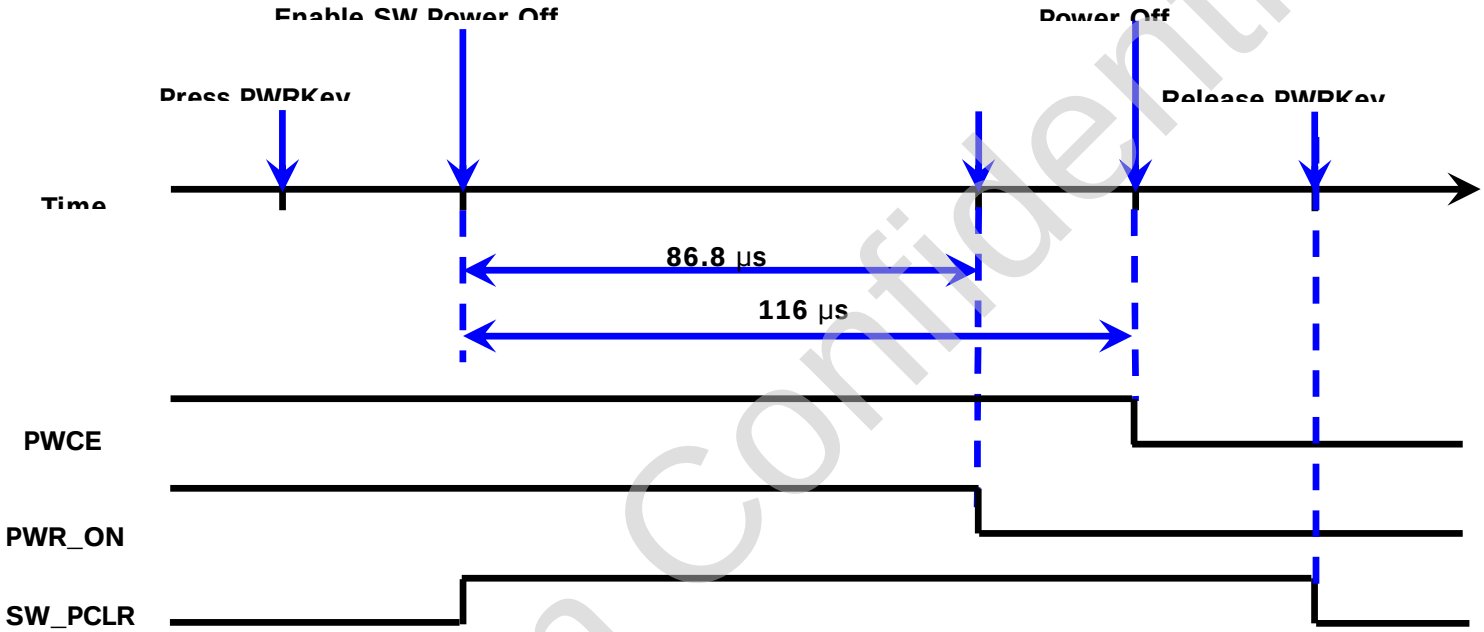
The RTC supports a hardware automatic power off function and a software power off function like Notebook. For hardware power off function, it can be enable and disable in HW_PCLR_EN bit and the user presses the power button for a few seconds to power off system. The time to press power the button to 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.



6.21.6 RTC Register Mapping

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	0x007F_FF00
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_0001
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
RTC_SET	RTC_BA+0x038	R/W	RTC Setting Register	0x0000_0000
OSC_32K	RTC_BA+0x03C	R/W	RC oscillator setting Register	0x0000_0000
OSC_CAL	RTC_BA+0x040	R	RC oscillator calibration Register	0x0000_0000
REG_FLAG	RTC_BA+0x044	R	RTC Register write complete	0x0000_0000

6.21.7 Register Descriptions

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 in 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	0x007F_FF00

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

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:24]	RPWR_DELAY	Delay RPWR output 3'b000: no change 3'b001: delay 62ms~124ms 3'b010: delay 124ms~248ms 3'b011: delay 186ms~372ms 3'b100: delay 248ms~496ms 3'b101: delay 310ms~620ms 3'b110: delay 372ms~744ms 3'b111: delay 434ms~868ms
[23:8]	INTEGER	Integer Part Real Oscillator Frequency = (Integer part of detected value-1)Ex: 32769Hz => 16'h8000 32768Hz => 16'h7FFF 32767Hz => 16'h7FFE
[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_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							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	12-hour time scale
00	12(AM12)	12	32(PM12)
01	01(AM01)	13	21(PM01)
02	02(AM02)	14	22(PM02)
03	03(AM03)	15	23(PM03)
04	04(AM04)	16	24(PM04)
05	05(AM05)	17	25(PM05)
06	06(AM06)	18	26(PM06)
07	07(AM07)	19	27(PM07)
08	08(AM08)	20	28(PM08)
09	09(AM09)	21	29(PM09)
10	10(AM10)	22	30(PM10)
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				RAIER	PSWIER	TIER	AIER

Bits	Descriptions	
[3]	RAIER	Relative Alarm Interrupt Enable 1 => RTC Relative Alarm Interrupt enable 0 => RTC Relative Alarm Interrupt disable
[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				RAI	PSWI	TI	AI

Bits	Descriptions	
[3]	RAI	RTC Relative Alarm Interrupt Indication 1: It indicates that Relative 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 Relative alarm interrupt never occurred. Software can also clear this bit after RTC interrupt has occurred.
[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. 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
RELATIVE_TIME[11:0]							
23	22	21	20	19	18	17	16
RELATIVE_TIME[11:0]				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		REL_ALARM_EN	ALARM_EN	HW_PCLR_EN	SW_PCLR	PWR_ON

Bits	Descriptions	
[31:20]	RELATIVE_TIME	Relative Time alarm period (second unit) The PCLR_TIME indicate the period of the relative time alarm, its maximum values is 12'd1800. When REL_ALARM_EN = 1'b0 , it will be cleared 0
[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.
[6:5]	RESERVED	RESERVED
[4]	REL_ALARM_EN	REL_ALARM_EN 1: If this bit is set to 1, enable relative time alarm control 0: If this bit is set to 0, disable relative time alarm control

[3]	ALARM_EN	ALARM_EN 1: If this bit is set to 1, enable alarm_int_control 0: If this bit is set to 0, disable alarm_int_control
[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.

RTC Setting Register (RTC_SET)

Register	Address	R/W/C	Description	Reset Value
RTC_SET	RTC_BA+0x038	R/W	RTC Setting 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			XOUT_XC	XIN_XC	RTC_AEN	RTC_EN	RTC_WEAK

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	XOUT_XC	XIN IO PAD input data(It is read only)
[3]	XIN_XC	XIN IO PAD input data (It is read only)
[2]	RTC_AEN	RTC_AEN XIN/XOUT PAD Digital I/O mode enable 0: digital I/O mode 1: analog mode (default value)
[1]	RTC_EN	RTC_EN 1: If this bit is set to 1, Crystal buffer Enable 0: If this bit is set to 0, Crystal buffer Disable
[0]	RTC_WEAK	RTC_WEAK 1: If this bit is set to 1, RTC macro is in weak mode(2uA) 0: If this bit is set to 0, RTC macro is in strong mode(8uA)

RC Oscillator Setting Register (OSC_32K)

Register	Address	R/W/C	Description	Reset Value
OSC_32K	RTC_BA+0x03C	R/W	RC oscillator setting 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							OSC_32K_EN

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	OSC_32K_EN	OSC_32K_EN 1: If this bit is set to 1, enable internal RC oscillator 0: If this bit is set to 0, disable internal RC oscillator

RTC Setting Register (OSC_32K)

Register	Address	R/W/C	Description	Reset Value
OSC_32K_CNT	RTC_BA+0x040	R	RC oscillator calibration counter register	0x0000_0000

31	30	29	28	27	26	25	24
OSC_32K_CNT[31:24]							
23	22	21	20	19	18	17	16
OSC_32K_CNT[23:16]							
15	14	13	12	11	10	9	8
OSC_32K_CNT[15:8]							
7	6	5	4	3	2	1	0
OSC_32K_CNT[7:0]							

Bits	Descriptions	
[31:0]	OSC_32K_CNT	To use PCLK to count OSC_32K width, then use this value to calculate real OSC_32K FCR values!

RTC Register Complete Register (REG_FLAG)

Register	Address	R/W/C	Description	Reset Value
REG_FLAG	RTC_BA+0x044	R	RTC Register write complete	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						REG_FLAG	

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	REG_FLAG	Polling the flag to detect RTC register write complete 0: can not write 1: write complete

6.22 I2C Synchronous Serial Interface Controller

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.

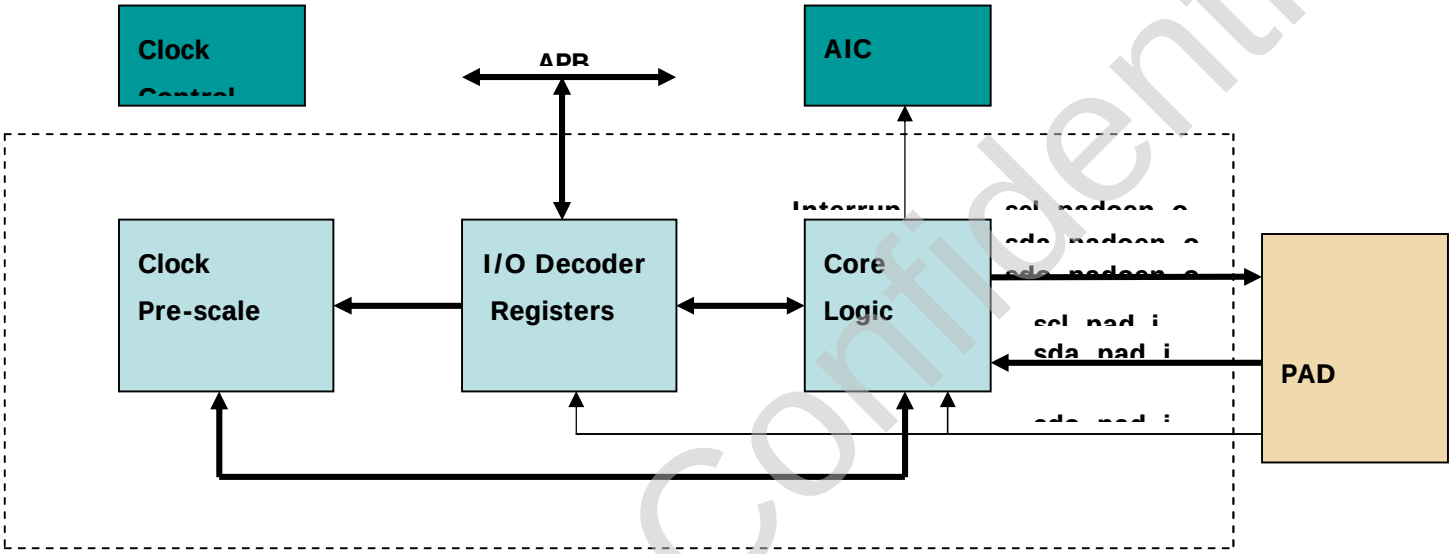
Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during 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).

The I²C Master Core includes the following 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**

6.22.1 I2C Serial Interface Block Diagram

The block diagram of I²C Serial Interface controller is shown as following.



I²C Block Diagram

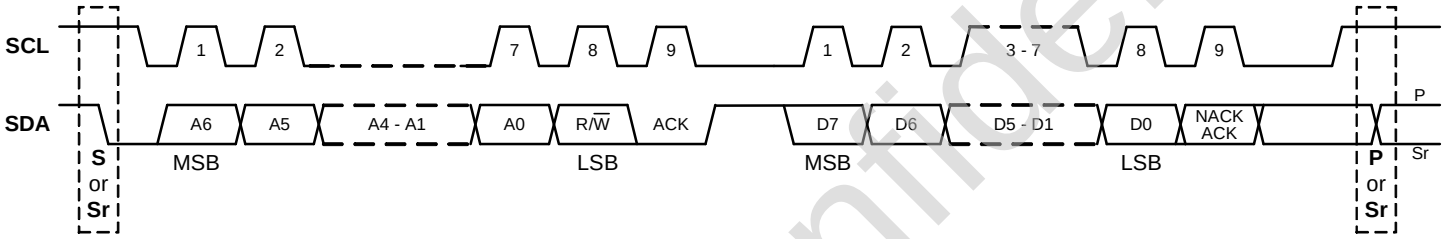
NOTE1: scl_pad_o, sda_pad_o and sdo_pad_o are always tied to 1'b0.

NOTE2: scl_padoen_o, sda_padoen_o and sdo_padoen_o are active low signals.

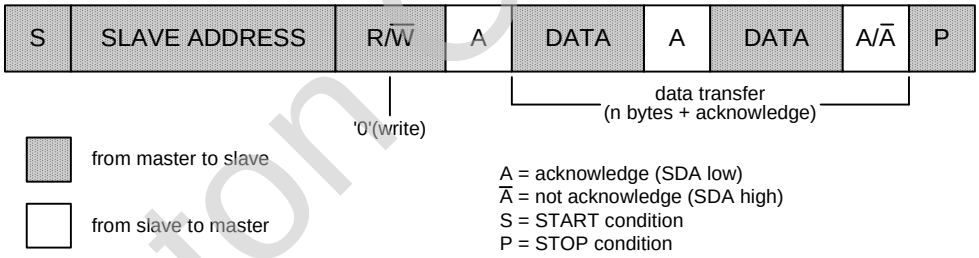
6.22.2 I2C Protocol

Normally, a standard communication consists of four parts:

- **START or Repeated START signal generation**
- **Slave address transfer**
- **Data transfer**
- **STOP signal generation**

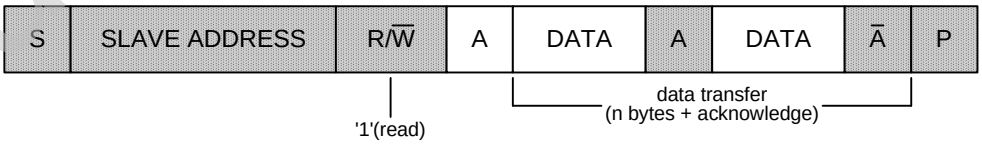


Data transfer on the I²C-bus



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)

START or Repeated START signal

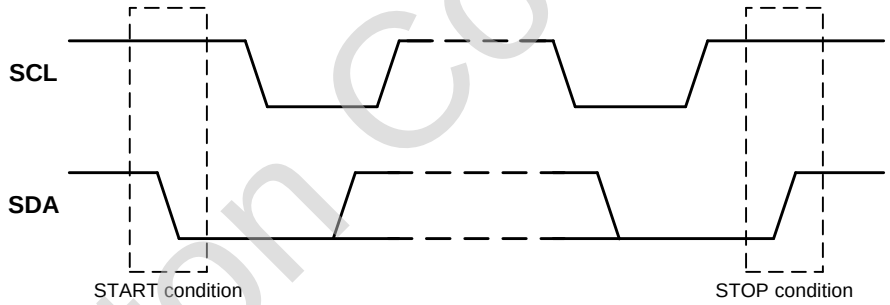
When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START 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 I^2C 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.

6.22.3 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.

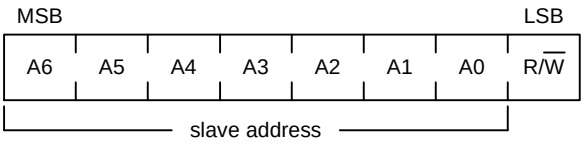


START and STOP conditions

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.



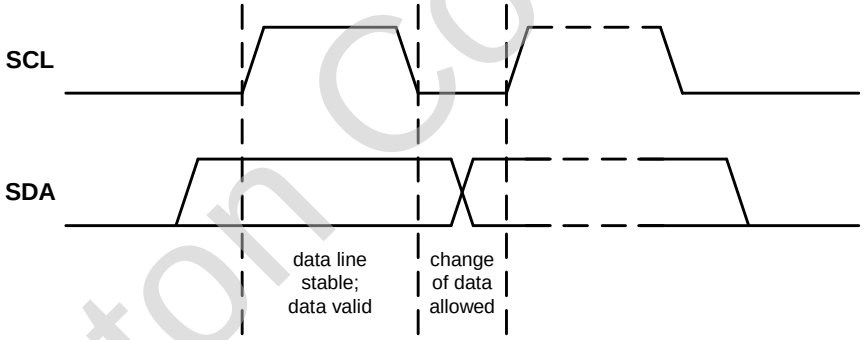
The first byte after the START procedure

Data Transfer

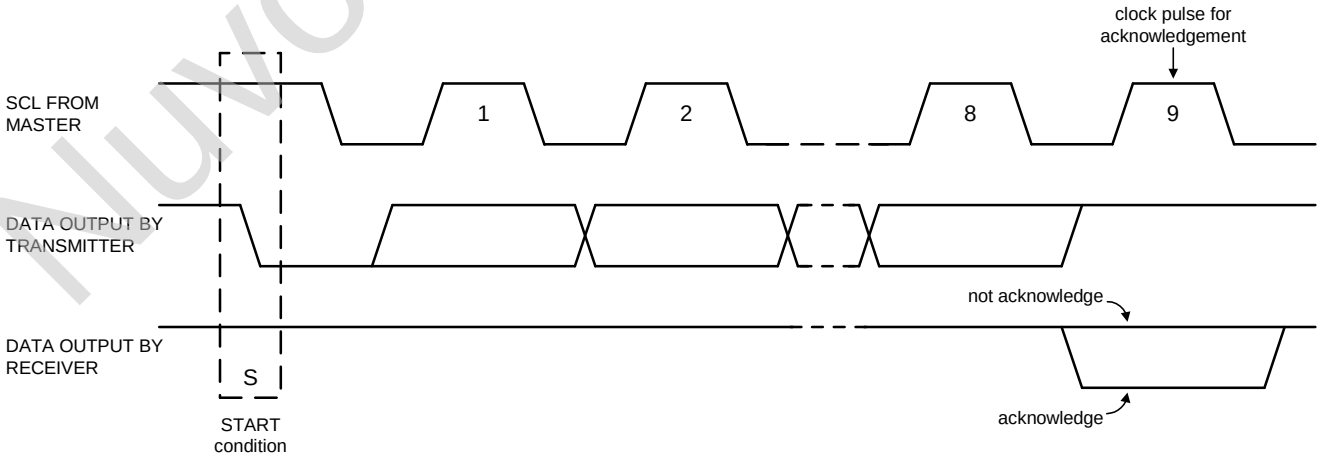
Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the 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.



Bit transfer on the I^2C-bus



6.22.4 I2C Programming Examples

Example 1

Write 1 byte of data to a slave (using multi-byte transmit mode).

Slave address = 0x51 (7b′1010001)

Data to write = 0xAC

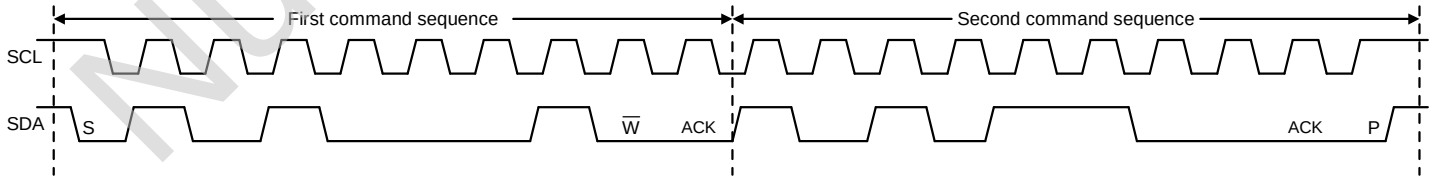
I²C Sequence:

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.**
5) **−− Wait for interrupt or I2C_TIP flag to negate −−**
6) **Read I2C_RxACK bit from CSR Register, it should be `0′.**
7) **Set Tx_NUM = 0x0.**
8) **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 then the I²C cycle time, and therefore not visible.

Example 2

Read a byte of data from an I2C memory device (using single byte transfer mode).

Slave address = 0x4E (7′b1001110)

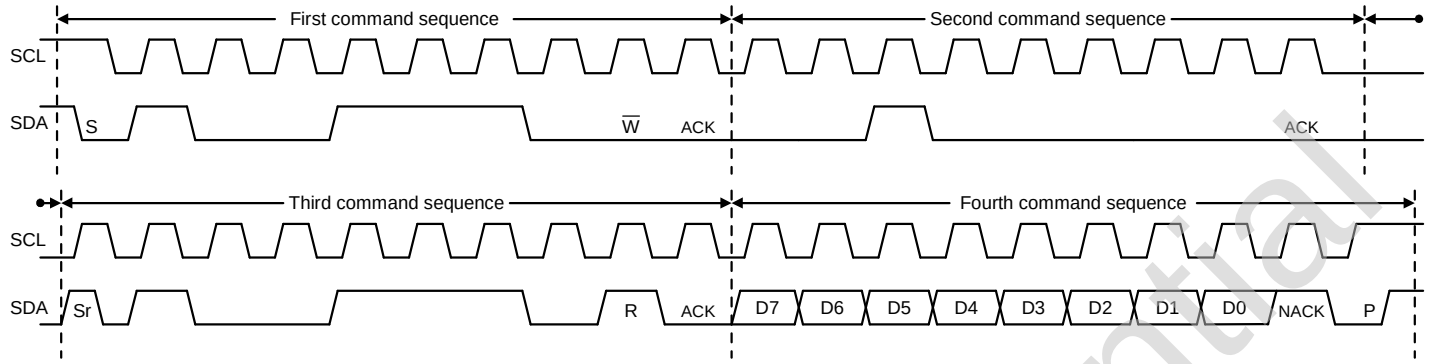
Memory location to read from = 0x20

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:

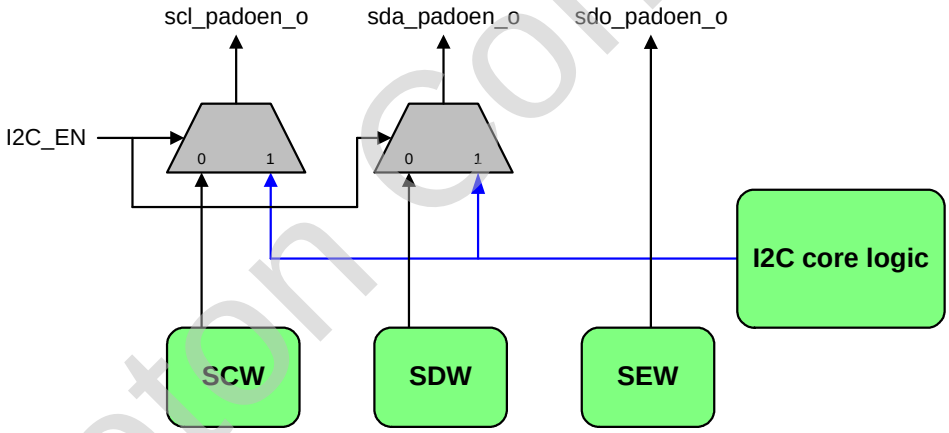
1) **Write a value into DIVIDER to determine the frequency of serial clock.**
2) **Set Tx_NUM = 0x0 and set I2C_EN = 1 to enable I^2C core.**
3) **Write 0x9C (address + write bit) to TxR[7:0], set START bit and WRITE bit.**
   -- Wait for interrupt or I2C_TIP flag to negate --
4) **Read I2C_RxACK bit 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 then the I²C cycle time, and therefore not visible.

6.22.5 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.



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.

6.22.6 I2C Serial Interface Control 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.

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		SGMST_EN	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^2C 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^2C 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^2C 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]	SGMST_EN	**Single master mode enable** • 0 = Multiple master mode. (using Sync. and Async. logic to detect START and STOP) • 1 = Single master mode. (only using Sync. logic to detect START and STOP)
[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 I^2C 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 $prescale = \frac{32\ MHz}{5 * 100\ KHz} - 1 = 63\ (dec) = 3F\ (hex)$

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^2C bus.
[3]	STOP	Generate Stop Condition Generate stop condition on I^2C 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^2C 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 I^2C. 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^2C bus will put on this register. The I^2C 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 I^2C core used 32-bit transmit buffer and provide multi-byte transmit function. Set CSR[Tx_NUM] to a value that you want to transmit. I^2C 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

6.23 PWM-Timer

6.23.1 Introduction

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 Crystal or 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.

6.23.2 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

6.23.2.1 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)**

6.23.3 PWM Architecture

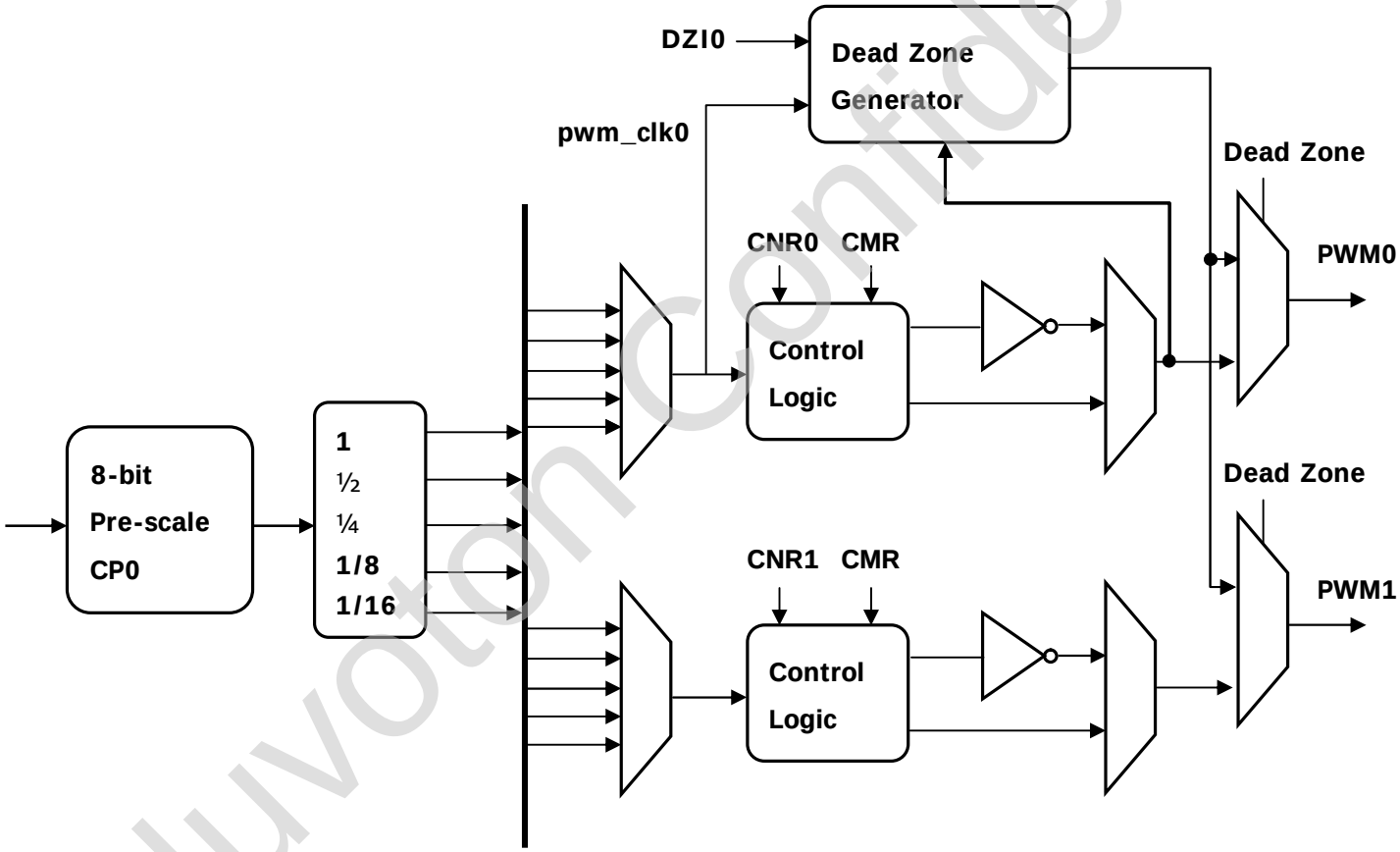
PWM_OE[0] enable ----à timer PWM0 output ----à GPB[1]

PWM_OE[1] enable ----à timer PWM1 output ----à GPB[2]

PWM_OE[2] enable ----à timer PWM2 output ----à GPB[3]

PWM_OE[3] enable ----à timer PWM3 output ----à 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)



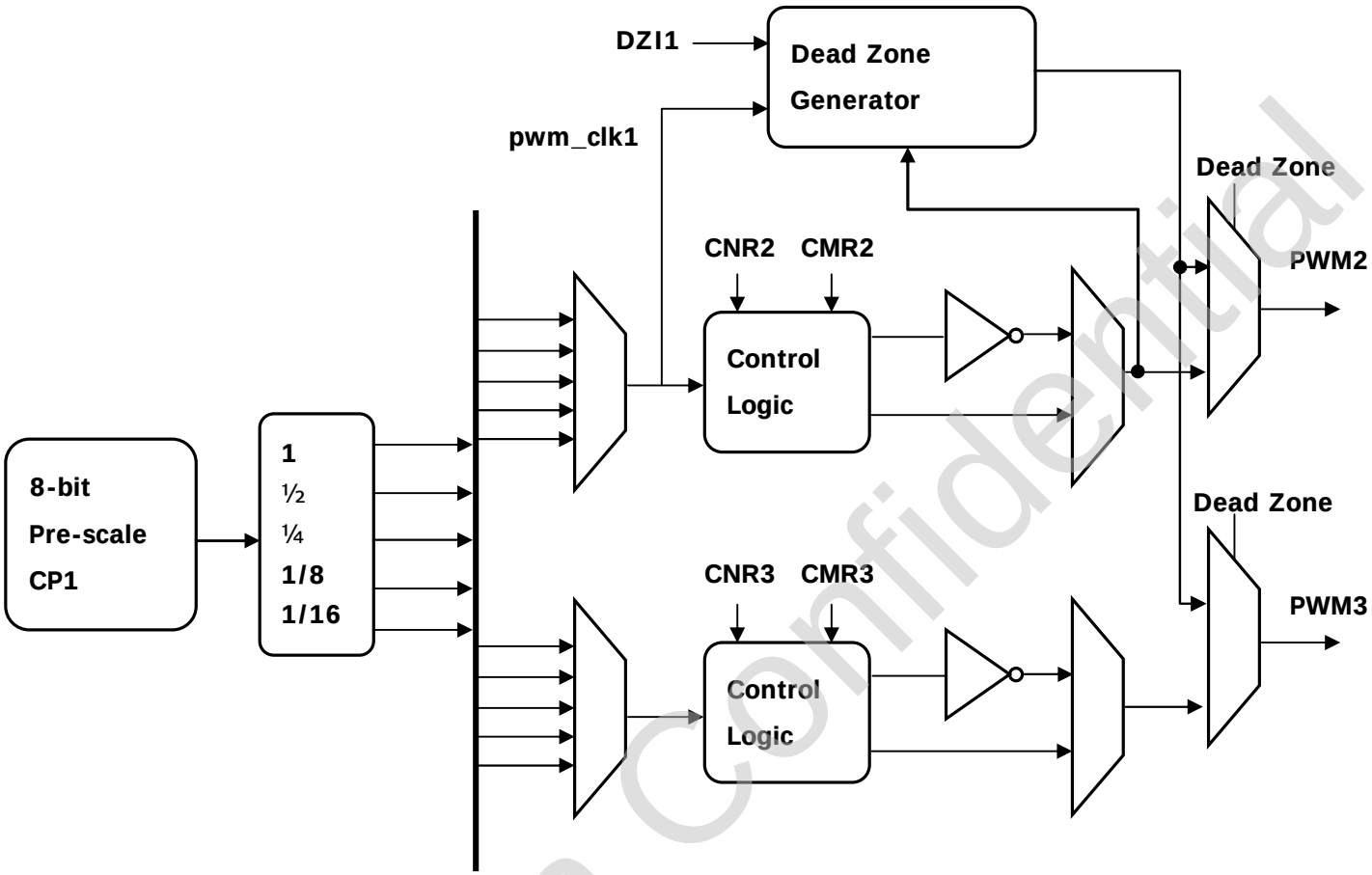


Figure 6.23-1 PWM Architecture Diagram

6.23.4 Basic Timer Operation

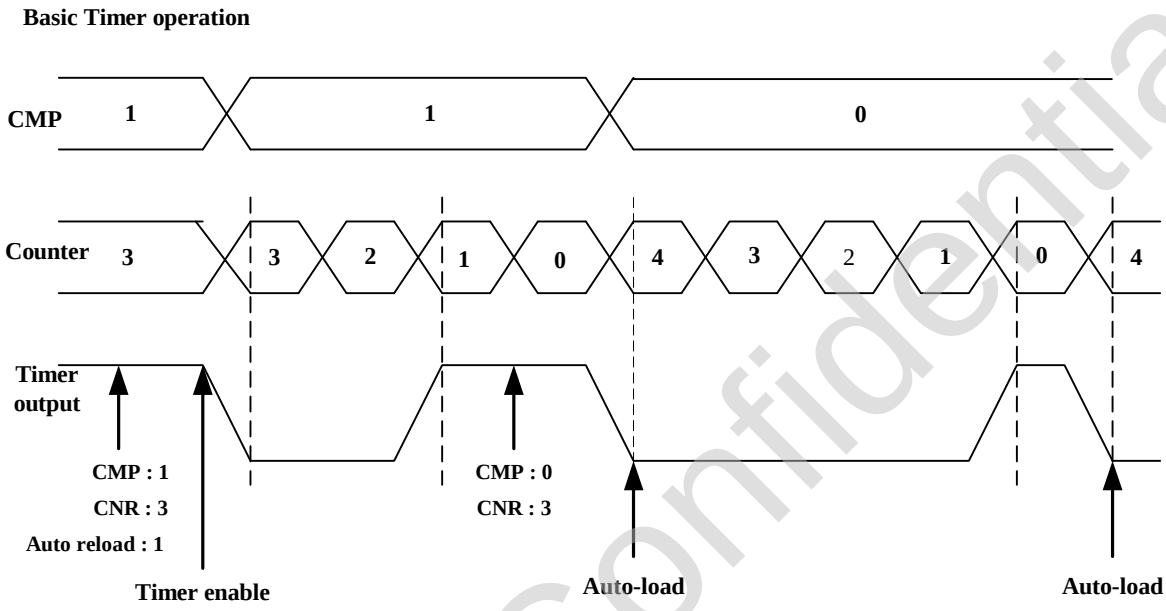


Figure 6.23-2 Basic Timer Operation Timing

6.23.5 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

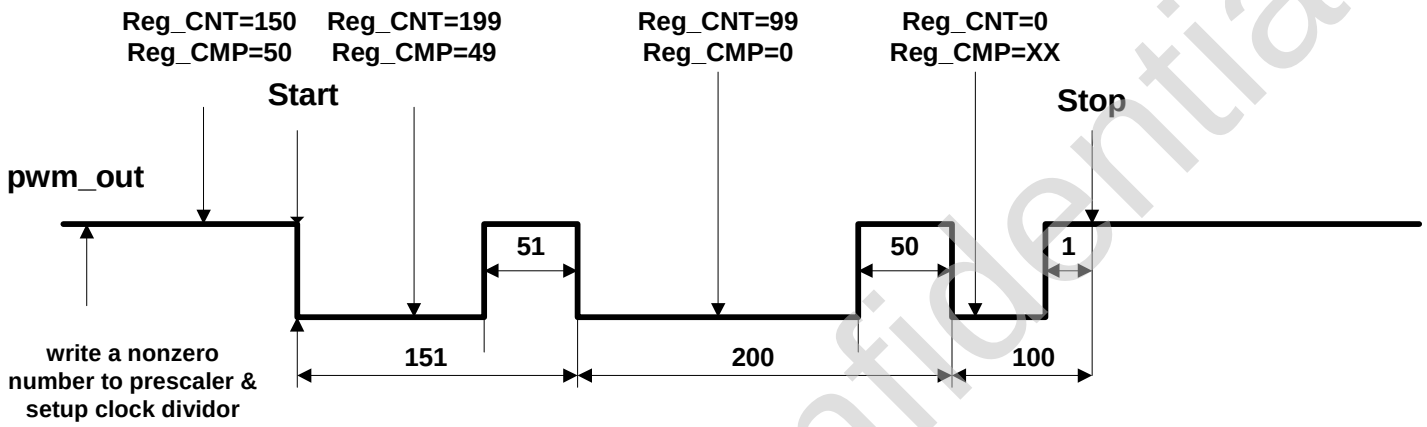


Figure 6.23-3 PWM Double Buffering Illustration

6.23.6 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 ouput duty ratio(CNR = 150)

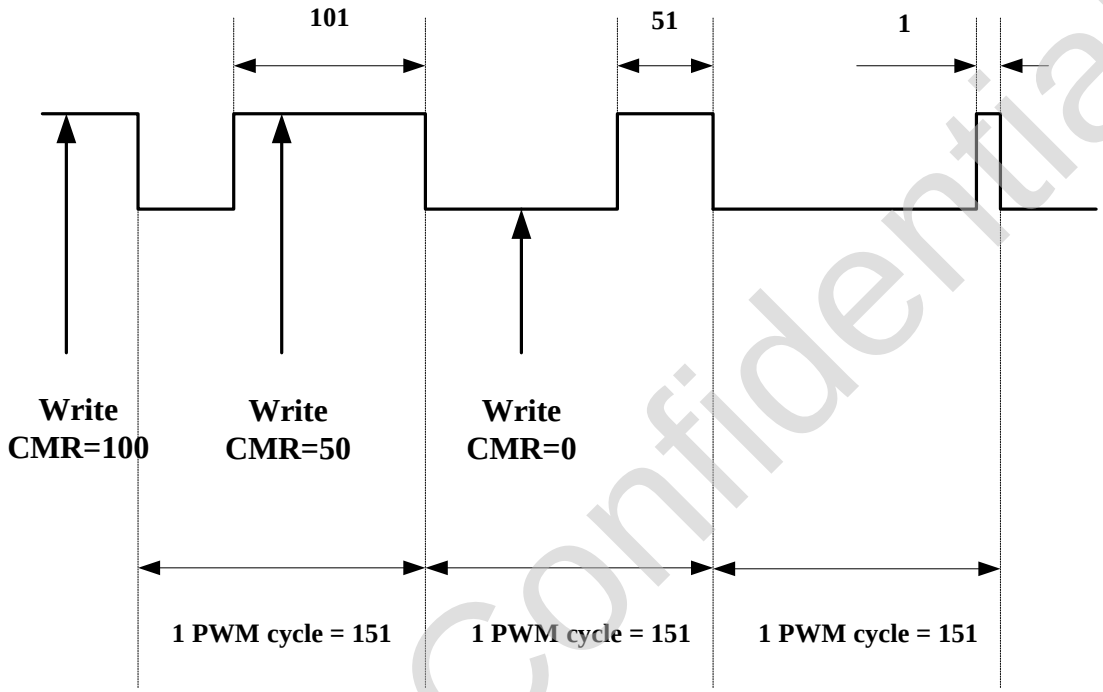


Figure 6.23-4 PWM Controller Output Duty Ratio

6.23.7 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

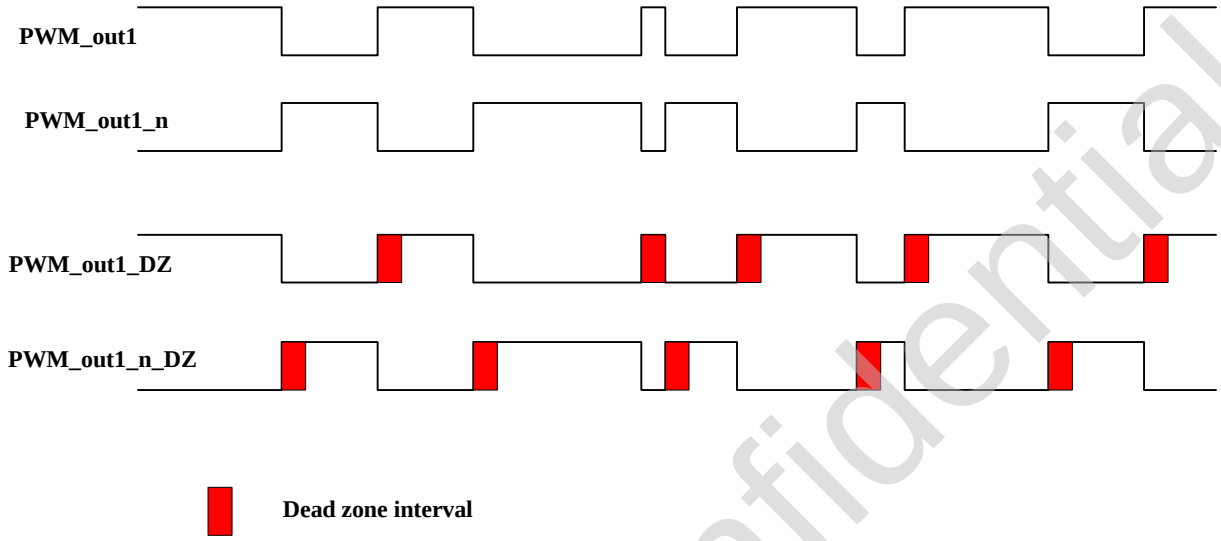


Figure 6.23-5 Dead Zone Generation Operation

6.23.8 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)

6.23.9 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)

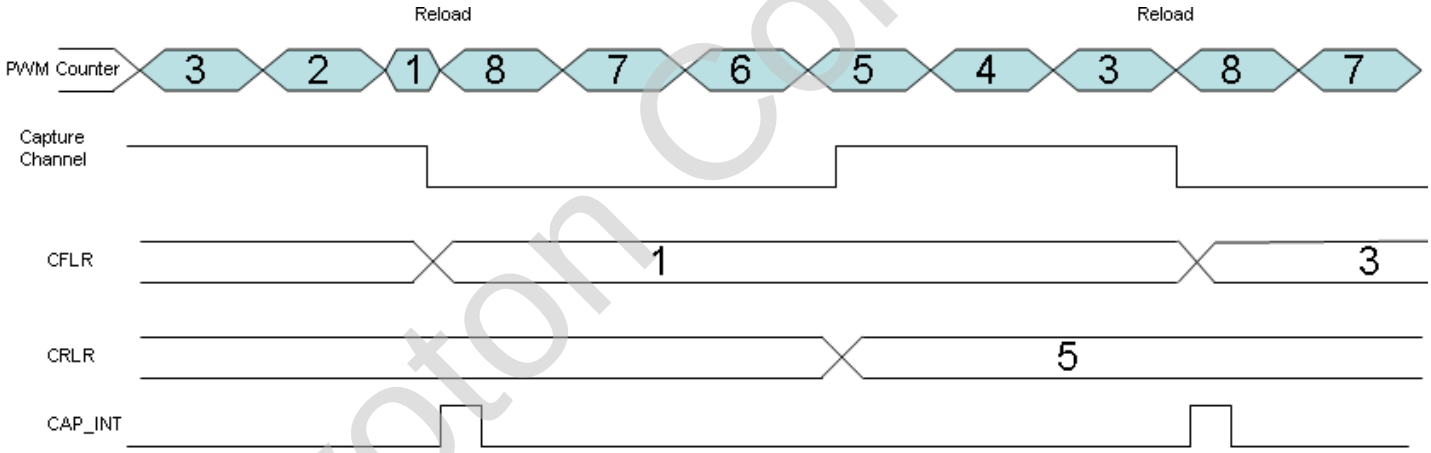
Method 3:

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).

6.23.10 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

Table NO,:1110-0001-08-A

CAPENR	PWM_BA+0x078	R/W	Capture Input Enable Register	0x0000_0000
POE	PWM_BA+0x07C	R/W	PWM Output Enable	0x0000_0000

6.23.11 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]	DZI11	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.
[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)

CSR3 [14:12]	Input clock divided by
100	1
011	16
010	8
001	4
000	2

[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
[15:12]	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] 3210 xxx1 è Capture channel 0 is from GPD[0] xx1x è Capture channel 1 is from GPD[1] x1xx è Capture channel 2 is from GPD[2] 1xxx è Capture channel 3 is from 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

6.24 UART Interface Controller

6.24.1 Overview

The W55FA95 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 Baud Rate = UART_CLK / M * [BRD + 2], where M and BRD are defined in Baud Rate Divider Register (UA_BAUD). 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	UART_CLK / [16 * (A+2)]
Enable	0	B	A	UART_CLK / [(B+1) * (A+2)] , B must >= 8
Enable	1	B	A	UART_CLK / (A+2), A must >=3

6.24.2 Features:

64 byte/16 byte entry FIFOs for received and transmitted data payloads.

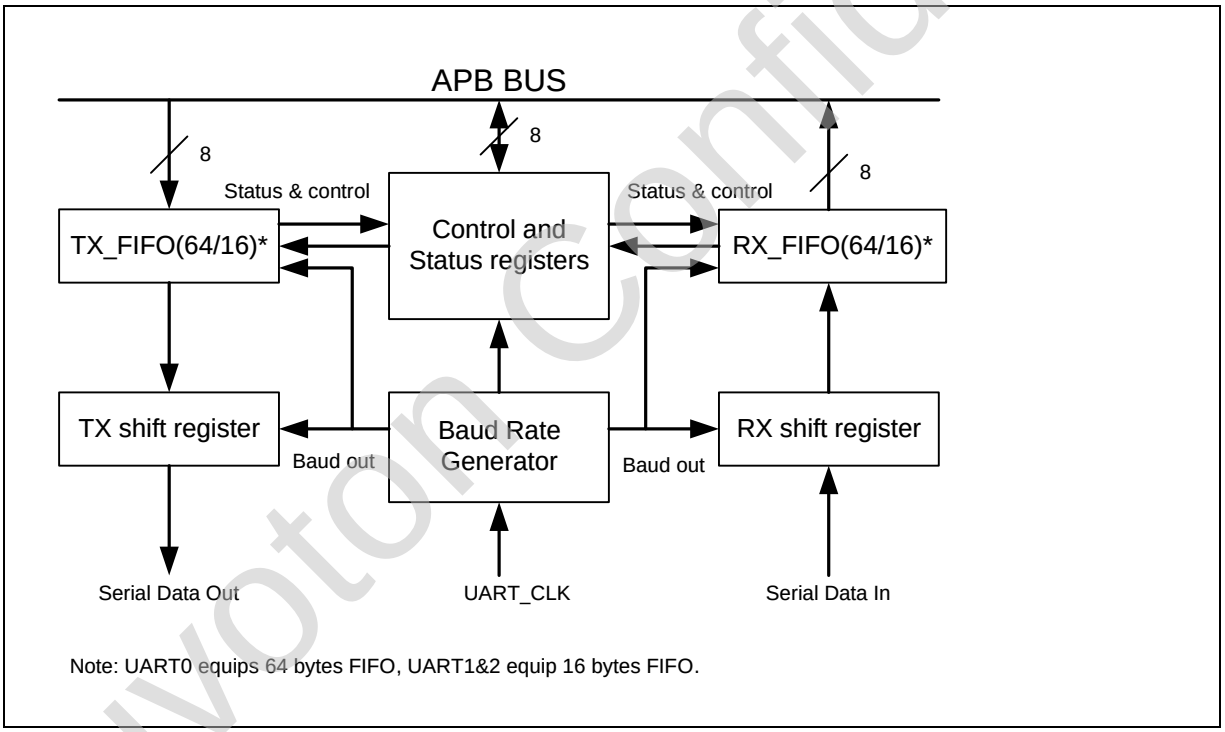
Auto flow control/flow control function (CTS, RTS) are supported (Normal speed UART not support).

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**

6.24.3 Block Diagram



UART Block Diagram

6.24.4 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.

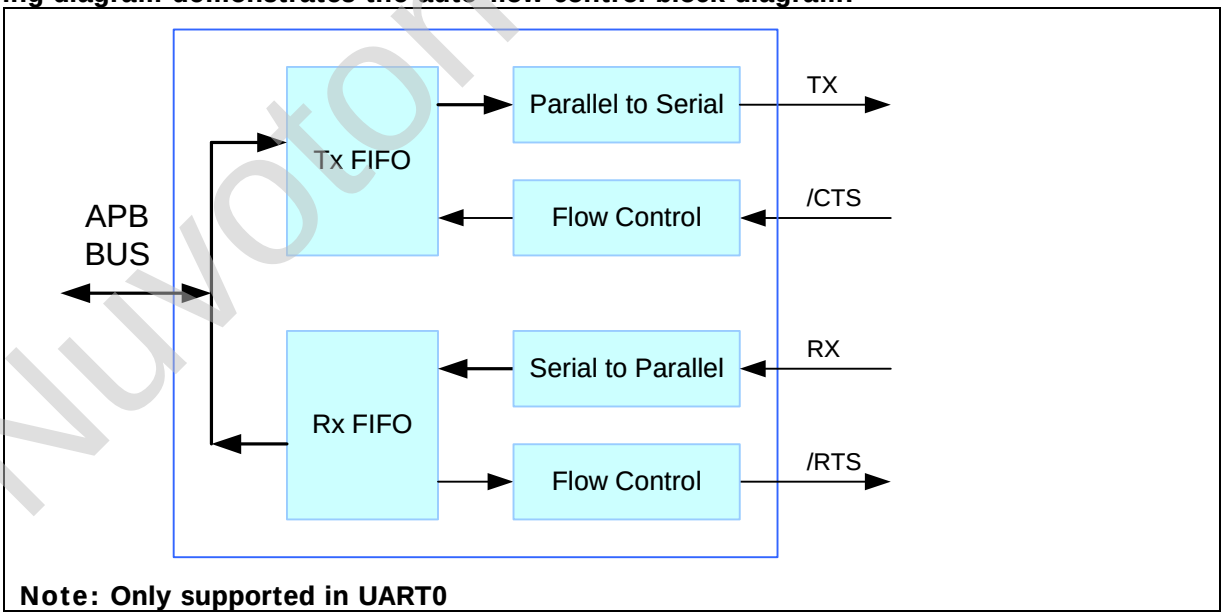
Baud Rate Generator

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.

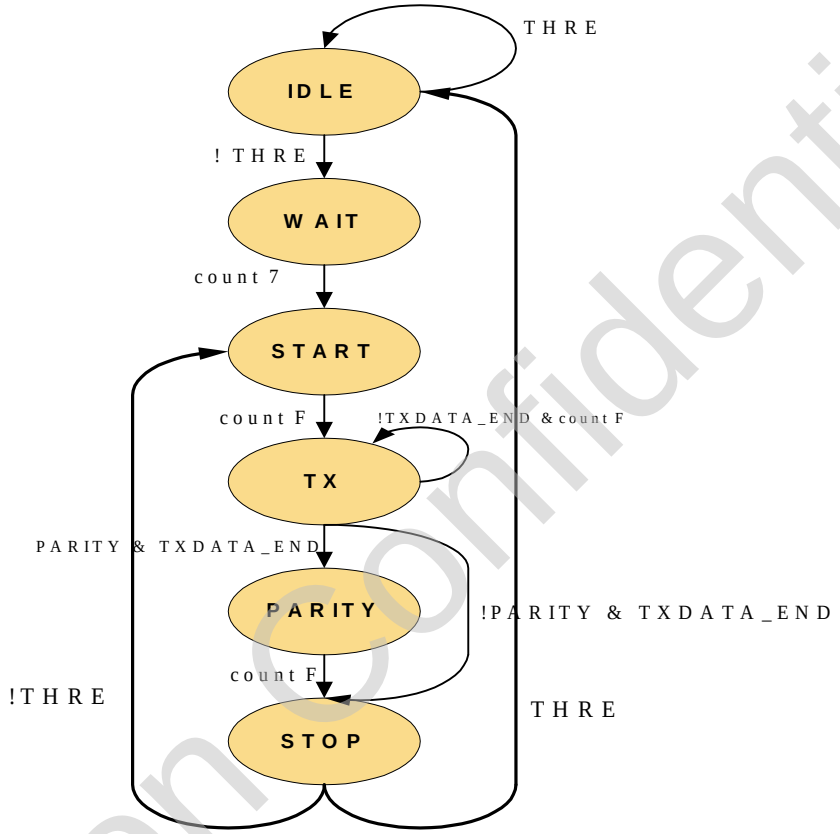


Note: Only supported in UART0

Auto Flow Control Block Diagram

6.24.5 Finite State Machine

6.24.5.1 Transmitter



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

Te 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

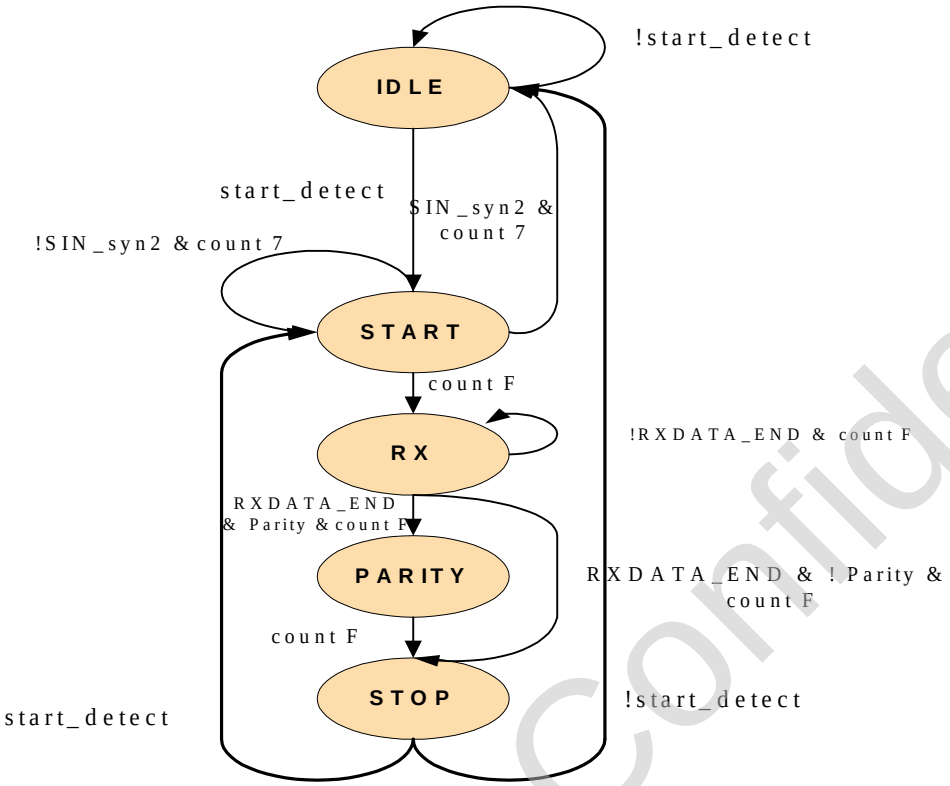
The transfer includes the parity bit.

NOTE:

The format of the transfer is as following:

One transfer = Start + Data + Parity bit (if dedicated) + Stop bit

6.24.5.2 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

6.24.6 UART Interface Control 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_8100

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

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_EN	DMA_Tx_EN	Auto_CTS_EN	Auto_RTS_EN	Time_out_EN	Reserved		
7	6	5	4	3	2	1	0
Reserved	Wake_IEN	BUF_ERR_IEN	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 dessert 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
RTS_ctrl_n	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	
[31:24]	Reserved	Reserved
[23]	RTS_ctrl_n	RTS Control FIFO Enable (Active-low) 0 : RxFIFO is controlled by RTS, it can not be written when RTS is active. 1 : RxFIFO can be written until RxFIFO is full.
[22:20]	Reserved	Reserved

<table>
<tr><td>[19:16]</td><td>RTS_tri_lev</td><td>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>
Note: This bit is use for auto RTS flow control.</td></tr>
<tr><td>[7:4]</td><td>RFITL</td><td>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).
<table>
<tr><th>RFITL</th><th>INTR_RDA 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></td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2]</td><td>Tx_RST</td><td>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.</td></tr>
</table>

[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 _Flag	HW_ Wake_INT	HW_Buf_ Err_INT	HW_ Tout_INT	HW_ Modem_INT	HW_ RLS_INT	Reserved	
23	22	21	20	19	18	17	16
DMA_Tx _Flag	HW_ Wake_IF	HW_Buf_ Err_IF	HW_ Tout_IF	HW_ Modem_IF	HW_ RLS_IF	Reserved	
15	14	13	12	11	10	9	8
Soft_Rx _Flag	Wake_INT	Buf_Err _INT	Tout _INT	Modem _INT	RLS _INT	THRE _INT	RDA _INT
7	6	5	4	3	2	1	0
Soft_Tx _Flag	Wake_IF	Buf_Err _IF	Tout _IF	Modem _IF	RLS _IF	THRE _IF	RDA _IF

Bits	Descriptions	
[31]	DMA_Rx _Flag	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 occurres 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(DCTSF=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 DCTSF is cleared by a write 1 on DCTSF.
[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 occurres 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(DCTSF=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 DCTSF is cleared by a write 1 on DCTSF.
[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 = (DCTSF)	Write '1' to DCTSF
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
TOIC							

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:0]	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 Baud Rate = Clock / [M * (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 Baud Rate = Clock / M * [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	Clock / 16*(A+2)
1	0	B	A	Clock / B*(A+2), B must > 8
1	1	B	A	Clock / (A+2), A must > 3

6.25 SPI0 Serial Interface Controller (Master/Slave)

6.25.1 SPI (MICROWIRE) Synchronous Serial Interface Controller

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]

The MICROWIRE/SPI Master/Slave Core includes the following features:

- **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**

6.25.2 SPI (MICROWIRE) Block Diagram (Master/Slave)

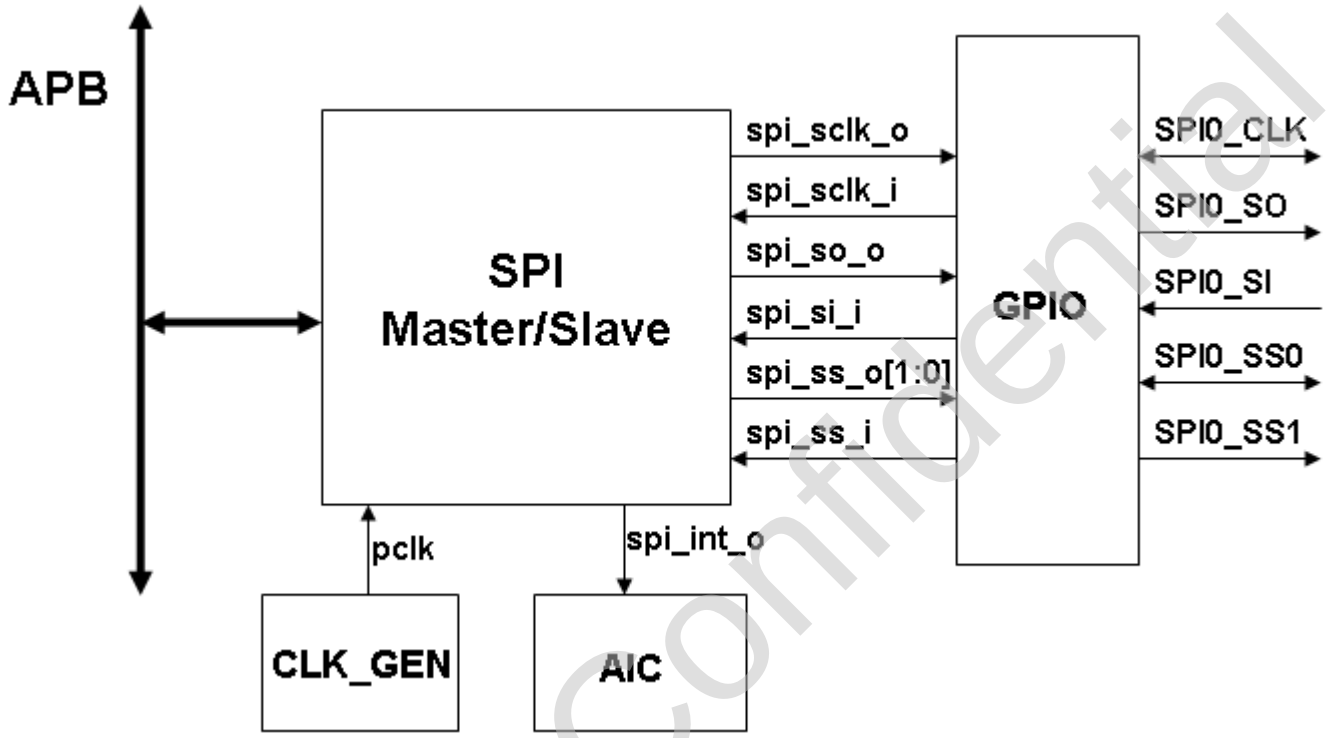


Figure 6.25-1 MICROWIRE/SPI Block Diagram (Master/Slave)

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.

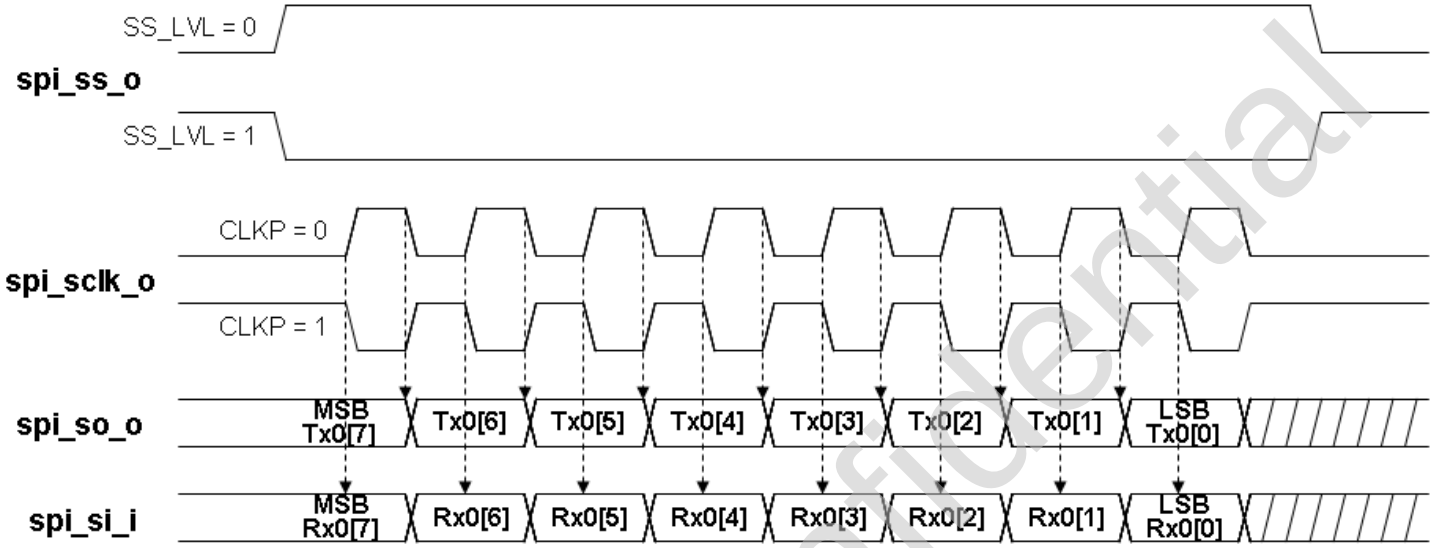
spi_sclk_i: MICROWIRE/SPI slave serial clock input.

spi_ss_i: MICROWIRE/SPI slave select signal input (edge-active trigger).

External pin descriptions:

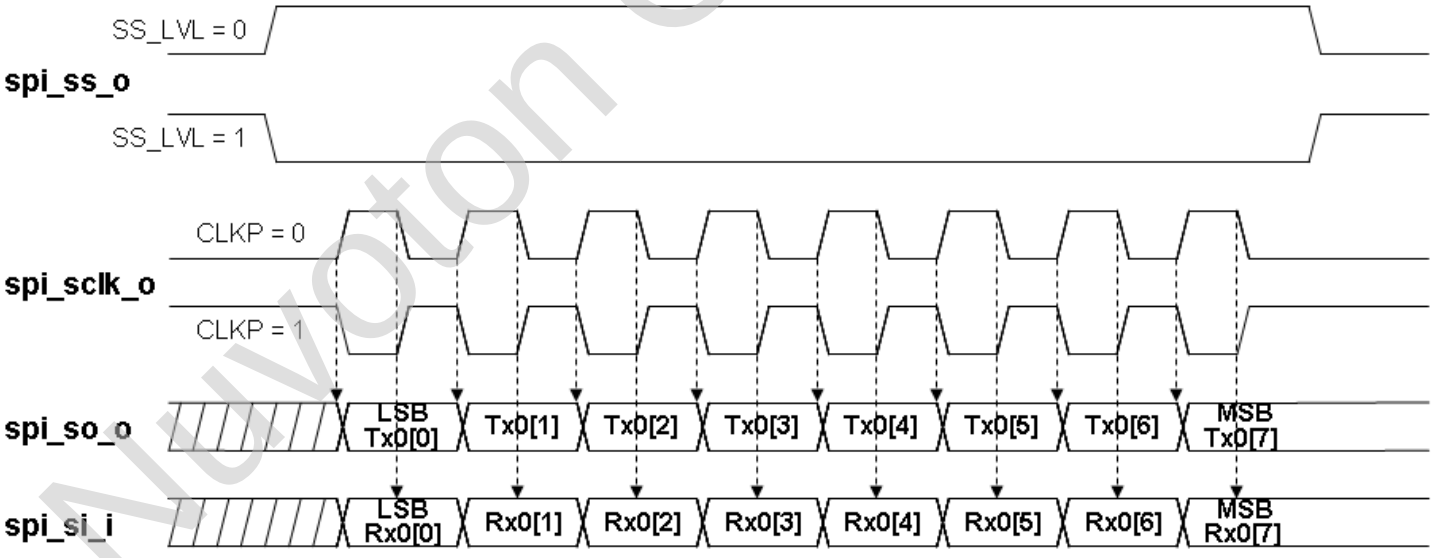
Pin Name	Pin Description	Type		GPIO
		Master	Slave	
SPI0_CLK	SPI 0 Clock	Output	Input	GPD[12]
SPI0_SO	SPI 0 Data Output	Output	Output	GPD[15]
SPI0_SI	SPI 0 Data Input	Input	Input	GPD[14]
SPI0_SS0	SPI 0 Slave Select 0	Output	Input	GPD[13]
SPI0_SS1	SPI 0 Slave Select 1	Output	Unused	GPA[5]/GPD[4]

6.25.3 SPI (MICROWIRE) Timing Diagram (Master/Slave)



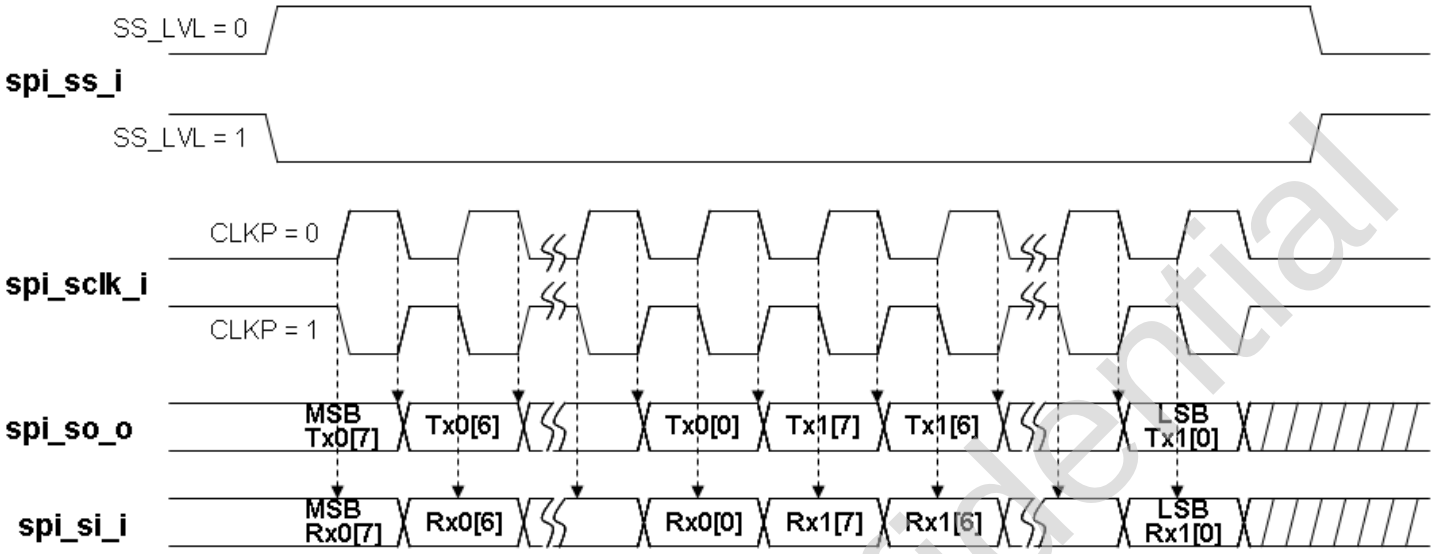
Master Mode : CNTRL[SLAVE]=0, CNTRL[LSB]=0, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0 or
2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1

Figure 6.25-2 MICROWIRE/SPI Timing (Master)



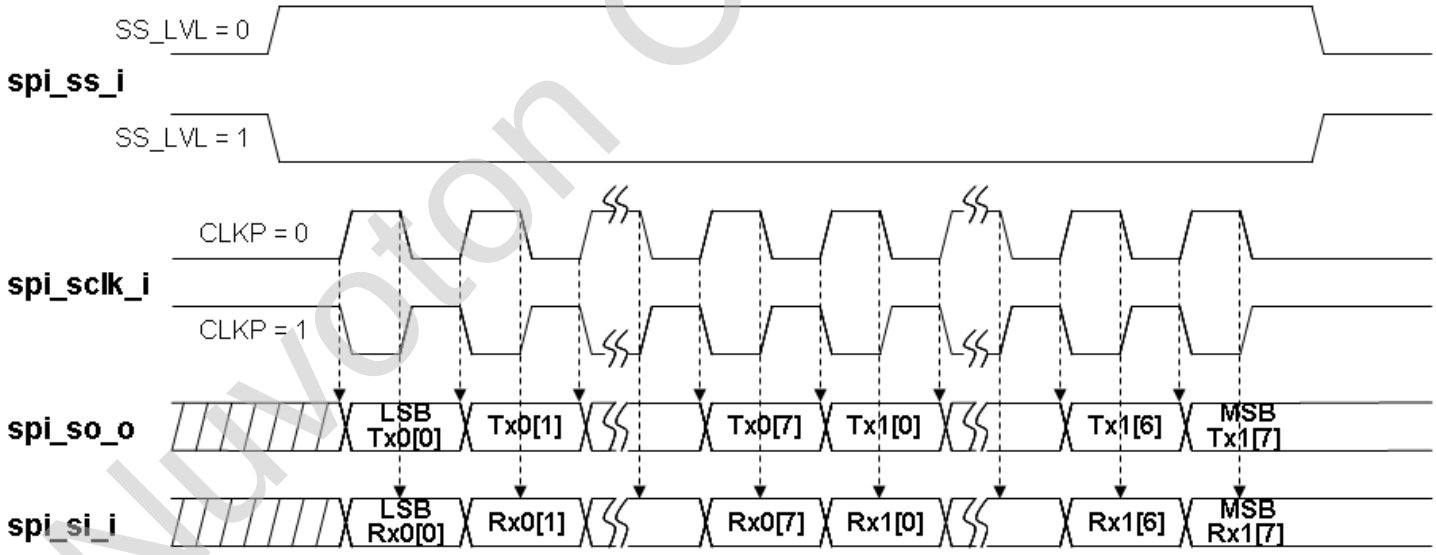
Master Mode : CNTRL[SLAVE]=0, CNTRL[LSB]=1, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1 or
2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0

Figure 6.25-3 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 6.25-4 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 6.25-5 Alternate Phase SCLK Clock Timing (Slave)

6.25.4 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.**

6.25.5 Wireless Joystick SPI Programming Example

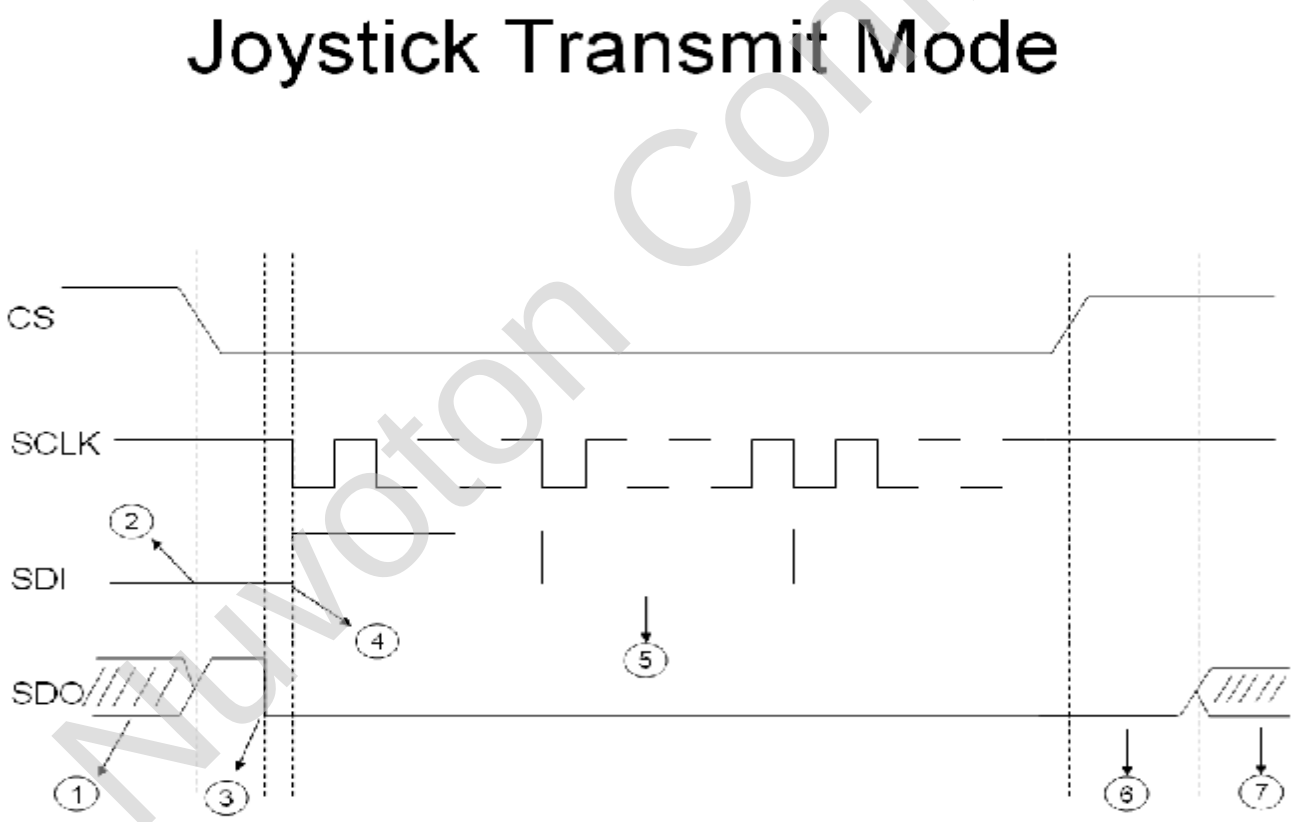
While connecting to wireless joystick, SPI should be set to slave mode and SPI0_JS[0] should be set to 1.

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:



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.

Drive SDO to 0 by setting SPI_JS[8] if slave is ready.

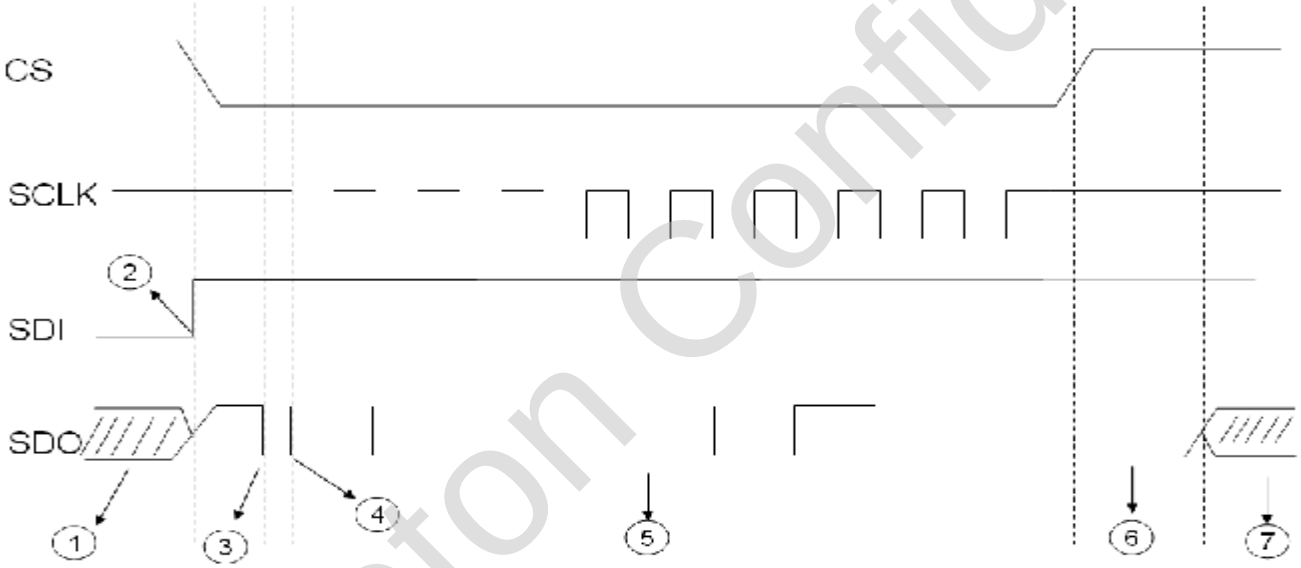
(4) When master detect that slave is ready, it starts to transmit data.

(5) Master will transmit 27-byte data.

(6) 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.

(7) 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.

Drive SDO to 0 by setting SPI_JS[8] if slave is ready.

(4) When master detect that slave is ready, it starts to receive data.

(5) Master will receive 7-byte data

(6) 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.

(7) 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.

6.25.6 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_TURBO	SPI0_BA+0x24	R/W	SPI0 turbo mode	0x0000_0000
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.

6.25.7 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)*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]	DIVIDER	**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] è CS is activated. JS_INT_FLAG[6] è 8-byte data is available in the buffer. JS_INT_FLAG[7] è 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.

Turbo Mode Control Register (Turbo)

Register	Offset	R/W	Description	Reset Value
SPI0_TURBO	SPI0_BA + 0x24	R/W	SPI turbo mode	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							TURBO

Bits	Descriptions	
[0]	TURBO	SPI turbo mode Originally, SPI0 only provide spi_sclk_o with frequency <= PCLK/2. If you set this bit to enter TURBO mode, the frequency of spi_sclk_o can be equal to PCLK. Remember to set this bit before setting SPI_CONTRL register.

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.

6.26 Analog to Digital Converter Control

6.26.1 ADC Control Block Diagram

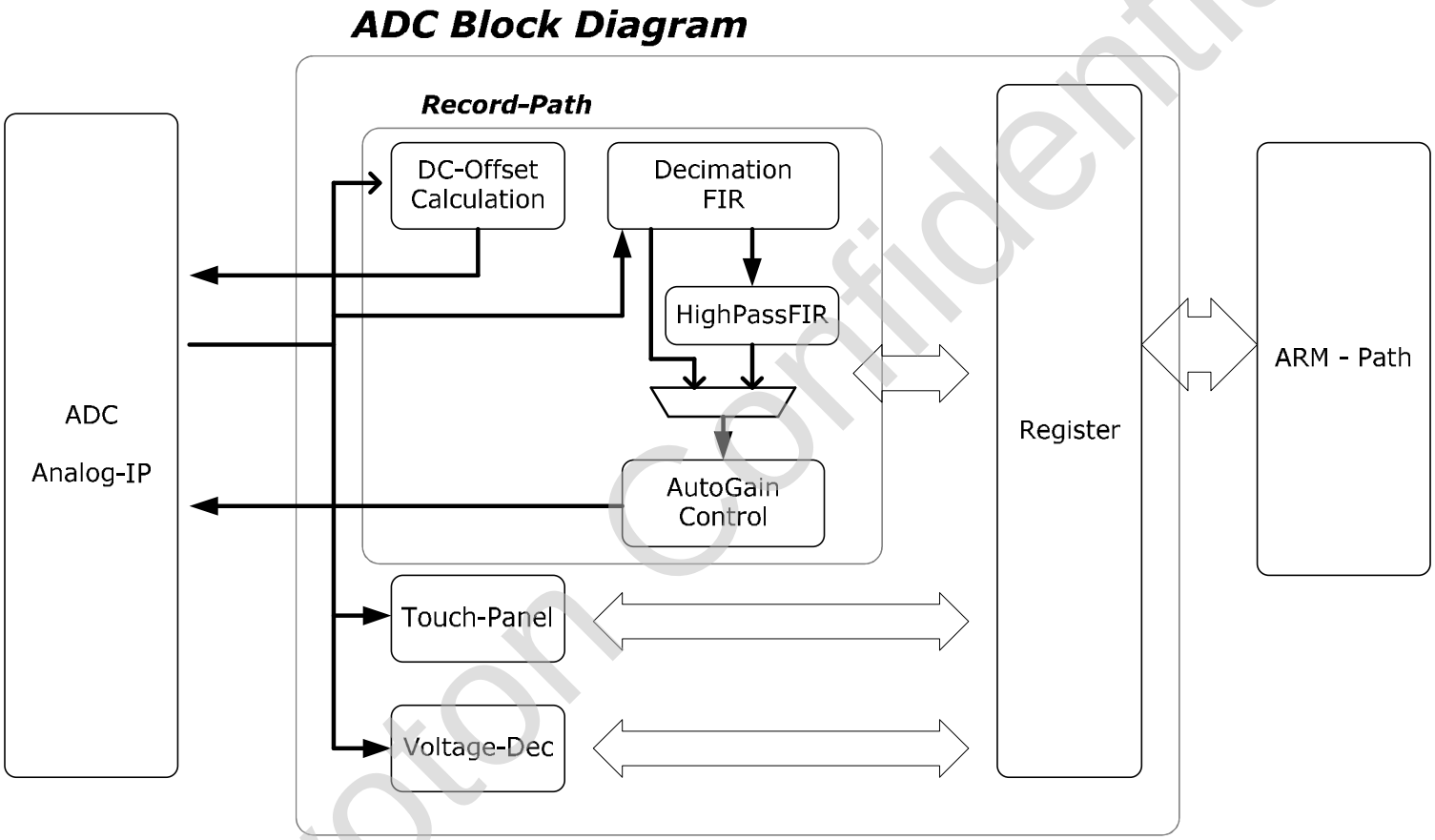


Figure 6.26-1 The ADC Block Diagram

The ADC control block has three functions. One is to record the voice. Which sampling rate is 8 kHz. And Second function is for touch panel use. The other function is for detecting the battery voltage. Therefore, we use the APB bus to control the ADC control register to order the command. Then, the ADC control block will control the analog ADC IP to get the conversion data. This data can be voice, location information, or voltage information. It depends on which function selecting.

6.26.2 Audio Record Block

This function is to record the sound. And, the sampling rate with audio data output is controlled by the input ADC

Clock. The equation is that the output sampling rate is the "ADC clock frequency / (AudioClkDiv+1) / 16". Which AudioClkDiv is the bits 7~1 in ADC_CON register. And the "16" is means Sampling-Rate-Change Rate. See the Fig.2. The decimation filter processes the 16 times down sampling rate convertor. Then, the high pass filter block is to cancel the DC portion. If there is the DC part in the input signal, it will cause the decrease with the SNR. And, we can control HPF_EN to close the HighPassFilter. The division is controlled by the bits 7~1 in ADC_CON register. It is used to auto generate the ADC_Start signal, if we open the audio recorder function.

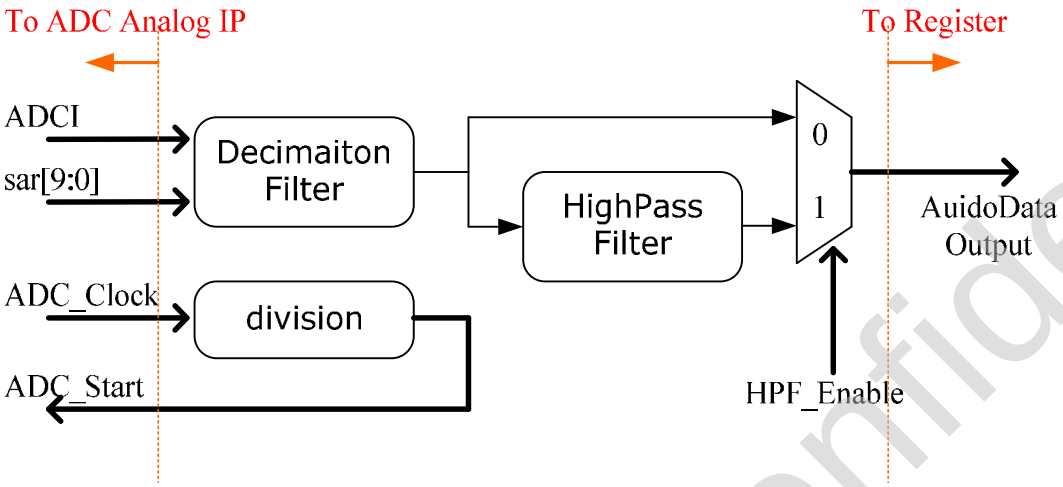


Figure 6.26-2 The Audio Record Block Diagram

6.26.3 Auto Gain Control Block (AGC)

Fig.3 shows the AGC block diagram. First, the audio data crosses the Moving Average Block to calculate the data power. Then, it will be passed to two block, target level compare and noise level compare, to generate the information about how to change the gain. The information from target level compare is used to change the gain into the target level. And, the noise level compare block is used to check this signal is noise or not. However, we avoid detecting Noise function too sensitive. We use the Enter NG Strength block and the Leave NG Strength Block to calculate the strength about noise or signal.

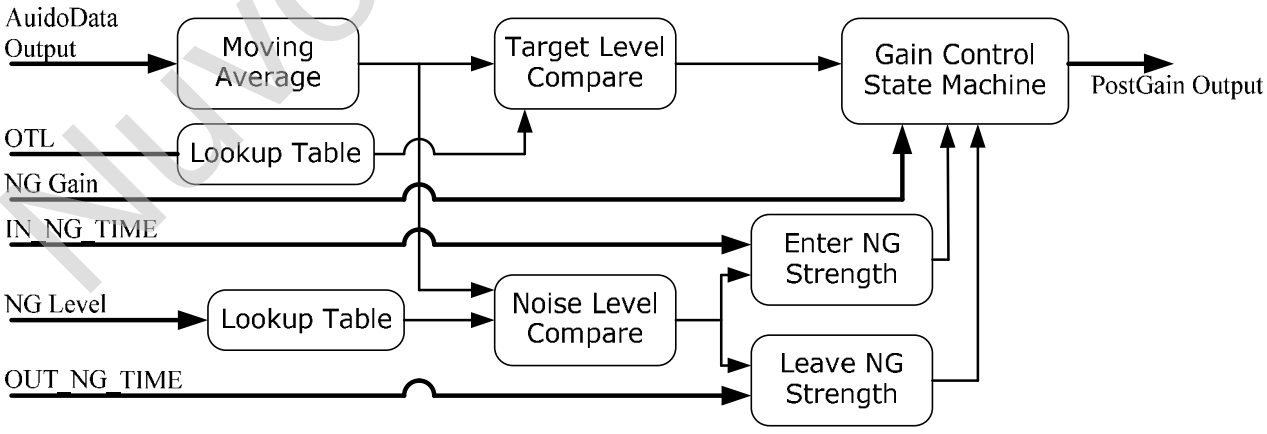


Figure 6.26-3 Auto Gain Control Block Diagram.

The algorithm is to count the time which the input data power is lower than the noise threshold, when in the Enter NG Strength Block. And if the data power is higher than the threshold, the timer would be reset. The algorithm is the as the leaver NG Strength Block. But the compare condition is opposite. By the way, the input data power can be see the register 0xB800E048

6.26.4 OP Offset Block

If in audio record mode and detecting the rising edge of the OPOC_EN, the OP Offset Block will be started. This block input data is SAR_ADC 10 bits output data. Then, the first "OPOC_DSC" data will be drop. Then, following 128 / 256 / 512 / 1024 samples (which depended on "OPOC_TCSN") will be summed together. Finally, uses the look-up table to find the corresponding bias compensation value. By the way, in the period with the OP Offset Calculation, the output PreGain will be set 0 dB, the PostGain will be set 12 dB, and the bias value will be set 0 mV.

The Summed Data output can be seen at the register 0xB800E050.

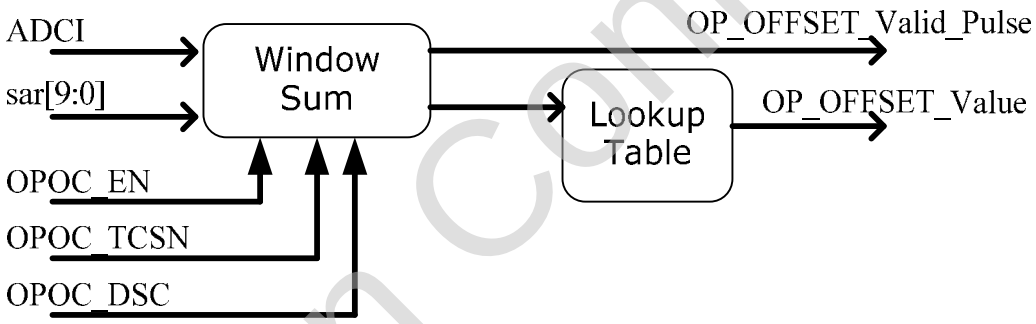


Figure 6.26-4 OP Offset Calculation Block Diagram

6.26.5 Touch Panel Block

Seeing Figure 6.26-5, it is the touch panel block diagram. There are four touch panel modes in this function. One is auto mode, semi mode, normal mode, and wait-for-trigger mode. In wait-for-trigger mode, if the WT_wakeup input signal is low, the interrupt will be happen. It means the touch panel has been touched. This mode is only use to check touch or not. Then, there are two cases in the auto mode. It depends on the MAV_EN. If MAV_EN is set to one, it means using average data as output data. Therefore, we will get four X / Y location to average one location. And we will check each data is on touch or not. If anyone is not on touch, this four data are all invalid and dropped. However, without MAV_EN, we will only get one X / Y location data, and check the same. Then, in the semi mode, we only get X or Y information at one times. If you want to get the other information, you should generate another start and change the register setting (ADC_TSC_XY). In Normal Node, the ADC_sar 10 bits data will be bypass, and the all register setting should be controlled by our self.

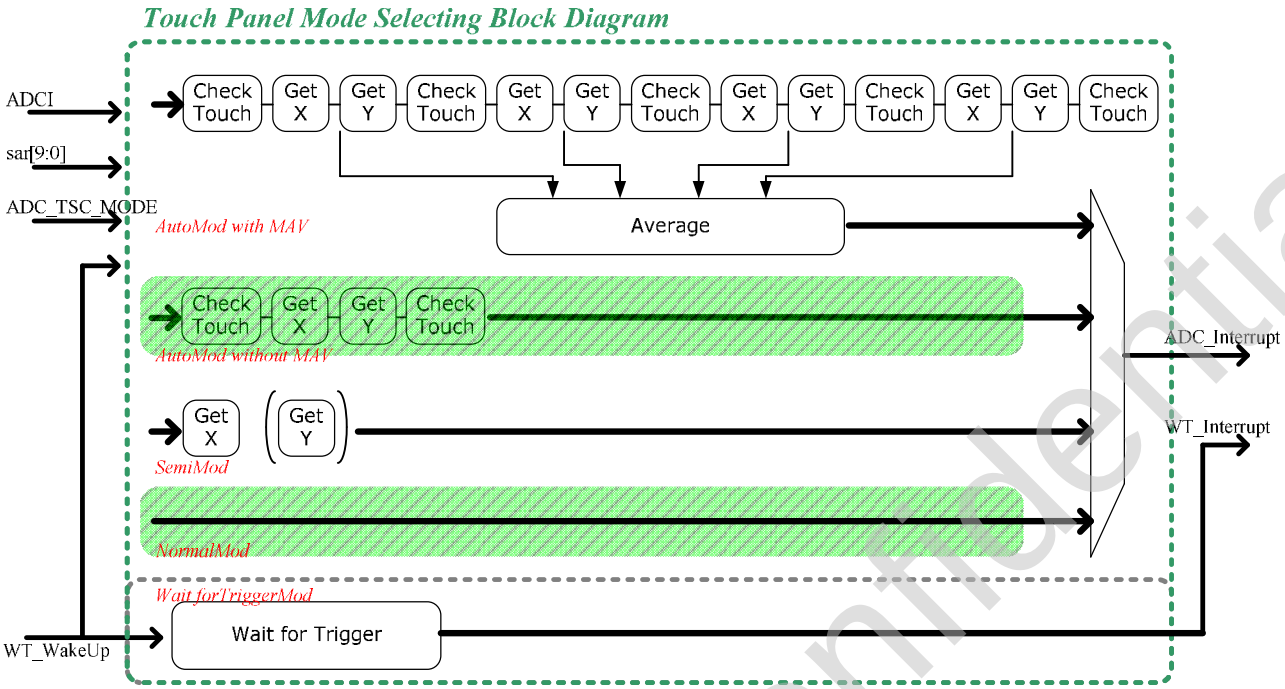


Figure 6.26-6 Touch Panel Block Diagram

6.26.6 ADC Control Register

6.26.6.1.1 ADC 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_001D
ADC_TSC	ADC_BA+0x004	R/W	Touch screen control register	0x0011_0000
ADC_DLY	ADC_BA+0x008	R/W	ADC delay register	0x0080_0010
ADC_XDATA	ADC_BA+0x00C	R	ADC XDATA register	0x0000_0000
ADC_YDATA	ADC_BA+0x010	R	ADC YDATA register	0x0000_0000
AudADC_CTRL	ADC_BA+0x014	R/W	AudADC Parameter Control register	0x0003_0300
AUDIO_CON	ADC_BA+0x01C	R/W	Audio control register	0x0050_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_B325
AGC_CON	ADC_BA+0x034	R/W	AGC Control register	0x00C0_0333
OPOC	ADC_BA+0x038	R/W	OP Offset Calculation Register	0x0002_00A0
DelPeakCtrl	ADC_BA+0x03C	R/W	Touch Panel and FPGA Peak Delete	0x0020_0040
NG_Ctrl	ADC_BA+0x040	R/W	Noise Gate Control	0x0033_201D
StateFlag	ADC_BA+0x044	R	State Flag and Situation	0x0000_2500
AGC_MAVSum	ADC_BA+0x048	R	AGC Moving Average Sum	0x0000_0000
OvFw_Ctrl	ADC_BA+0x04C	R/W	OverFlow Threshold control register	0x3FFF_0000
OPOC_CalSum	ADC_BA+0x050	R	DC Offset Calculation Sum	0x0000_0000
NoiseCalibra	ADC_BA+0x054	R/W	Noise Level Calibration	0x0000_0000
TSC_SORT4	ADC_BA+0x070	R	4th Touch Screen MAV	0x0000_0000
TSC_SORT3	ADC_BA+0x074	R	3rd Touch Screen MAV	0x0000_0000
TSC_SORT2	ADC_BA+0x078	R	2nd Touch Screen MAV	0x0000_0000
TSC_SORT1	ADC_BA+0x07C	R	1st Touch Screen MAV	0x0000_0000
TSC_MAV_X	ADC_BA+0x080	R	Touch Screen MAV X-Data	0x0000_0000
TSC_MAV_Y	ADC_BA+0x084	R	Touch Screen MAV Y-Data	0x0000_0000

6.26.7 ADC Control 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_01D

31	30	29	28	27	26	25	24
RESERVED				AudADC_EN	RESERVED		
23	22	21	20	19	18	17	16
WT_INT_EN	RESERVED	ADC_INT_EN	WT_INT	RESERVED	ADC_INT	ADC_EN	ADC_RST
15	14	13	12	11	10	9	8
ADC_TSC_MODE		ADC_CONV	ADC_READ_CONV	ADC_MUX			Reserved
7	6	5	4	3	2	1	0
AudioClkDiv							ADC_FINISH

Bits	Descriptions	
[31:24]	RESERVED	Reserved
[27]	AudADC_EN	Audio ADC Enable Bit 1′b1: Enable Audio ADC 1′b0: Disable Audio ADC
[26:24]	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.
[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.
[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 2'b00 : Normal conversion mode is selected 2'b01 : Semi-auto conversion mode is selected 2'b10 : Auto conversion mode is selected 2'b11 : Waiting for trigger mode is selected
[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. And the bit only use as Touch Panel. If audio record function is enable, this bit will be ignored.
[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. ADC_MUX=001, not available in normal data conversion. ADC_MUX=010, select AIN2 ADC_MUX=011, select AIN3 ADC_MUX=100, select AIN4 ADC_MUX=101, select AIN5 ADC_MUX=110, select AIN6 ADC_MUX=111, select AIN7 AIN0 and AIN1 channel are differential inputs for audio recorder only.
[8]	Reserved	
[7:1]	AudioClkDiv	This number can control the PLL_Audio_Clock frequency, because the PLL_Audio_Clock must be higher than the output frequency (fs). And the equation is Fs (Pll_Audio_Clock) = 16 * fs * (AudioClkDiv + 1) Note: Default 0x31
[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	0x0011_0000

31	30	29	28	27	26	25	24
CheckTime[14:7]							
23	22	21	20	19	18	17	16
CheckTime[6:0]							ADC_TSC_MAV_EN
15	14	13	12	11	10	9	8
Reserved							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	
[31:20]	CheckTime	In the AutoMode, we want to check the data which is valid or not by checking the pen is on the touch panel or not. And if the pen presses the panel, it must need enough time for changing the Voltage. Therefore, the needed change time can be set by these bits. (The unit is the ADC clock sample.)
[16]	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 done 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	0x0080_0010

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

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

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
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						ADC_XDATA[9:8]	
7	6	5	4	3	2	1	0
ADC_XDATA[7:0]							

Bits	Descriptions	
[9:0]	ADC_XDATA	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[9:8]	
7	6	5	4	3	2	1	0
ADC_YDATA[7:0]							

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.

AudADC Parameter Control (AudADC_CTRL)

Register	Address	R/W	Description	Reset Value
AudADC_CTRL	ADC_BA+0x014	R/W	AudADC Parameter control register	0x0003_0300

31	30	29	28	27	26	25	24
RESVERED							
23	22	21	20	19	18	17	16
AudADC_StrTime[7:0]							
15	14	13	12	11	10	9	8
TPADC_StrTime[7:0]							
7	6	5	4	3	2	1	0
RESVERED		AudADC_CUR	AudADC_VDD12	MIC_BIAS_EN	RESVERED	AudADC_ChSel	

Bits	Descriptions	
[23:16]	AudADC_StrTime	Audio ADC Start Period Time for Sample and Hold stable When detecting the falling edge about the Start Signal, the ADC will begin to convert the data. So the data which is stable or not is controlled by the sample and hold time. Therefore, this parameter is to control the sample and hold period time. The unit is ADC_CLK.
[15:8]	TPADC_StrTime	Touch Panel ADC Start Period Time for Sample and Hold stable When detecting the falling edge about the Start Signal, the ADC will begin to convert the data. So the data which is stable or not is controlled by the sample and hold time. Therefore, this parameter is to control the sample and hold period time. The unit is ADC_CLK.
[7:6]	RESERVED	
[5]	AudADC_CUR	Audio ADC Current Source Select Analog test use for controlling the current source.
[4]	AudADC_VDD12	Audio ADC Voltage Source Select Analog test use for controlling the voltage source
[3]	MIC_BIAS_EN	MIC Bias Enable 1′b0 : Disable the MIC Bias

		1′b1 : Enable the MIC Bias
[2]	RESERVED	
[1:0]	AudADC_ChSel	Audio ADC Channel Select Control 3′d0 : Channel Selected on MIC Audio In 3′d1 : Channel Selected on Ground 3′d2 : Channel Selected on AIN[2] 3′d3 : Channel Selected on 3.3V

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	0x0050_0000

31	30	29	28	27	26	25	24
AUD_INT_MODE[1:0]		AUD_INT	AUD_INTEN	VOL_EN	AUDIO_HPF_EN	AUDIO_EN	AUDIO_RESET
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. Note: If this bit be set, the ADC_MUX will be not effect.
[26]	AUDIO_HPF_EN	Audio High Pass Filter Enable 1′b1: Open Function 1′b0: Close Function
[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) The first AUD_CYCLE data will be ignored, because the decimation filter must has a period time to initialize the memory.
[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	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	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	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	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 regiser (AGCP1)

Register	Address	R/W	Description	Reset Value
AGCP1	ADC_BA+0x030	R/W	AGC Parameter Register Setting	0x0070_B325

31	30	29	28	27	26	25	24
EDMA_MODE	Reserved						DownSampSel
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/Disable 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
[31:25]	Reserved	Reserved
[24]	DownSampSel	Down Sampling Rate Selection 1′b1: Audio Record Down Sampling Rate 8 1′b0: Audio Record Down Sampling Rate 16 If the output sampling rate wants 16kHz, the down sampling rate must be selected to 8. It is limited by the max ADC clock is 2 MHz.
[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 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 -25.5dB~-3dB @ 1.5dB step size 4′hF : -3 dB 4′hE : -4.5 dB 4′hD : -6 dB : 4′h1 : -24 dB 4′h0 : -25.5 dB
[11]	UPBAND	1′b0:OTL+0.5dB 1′b1:OTL+0.75dB
[10]	DOWNBAND	1′b0:OTL-0.5dB 1′b1:OTL-0.75dB
[9:8]	PRAGA	Pre - Amplifier Gain Control 2′b00:-6 dB 2′b01:0 dB 2′b10:8 dB 2′b11:14 dB(default)
[7:6]	Reserved	Reserved
[5:0]	AUDIO_VOL	Audio Volume Control

		-12dB~35.25dB @ 0.75dB step size Note: If AGC is opened, the bits will have been ignored.

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

AGC Control Register (AGC_CON)

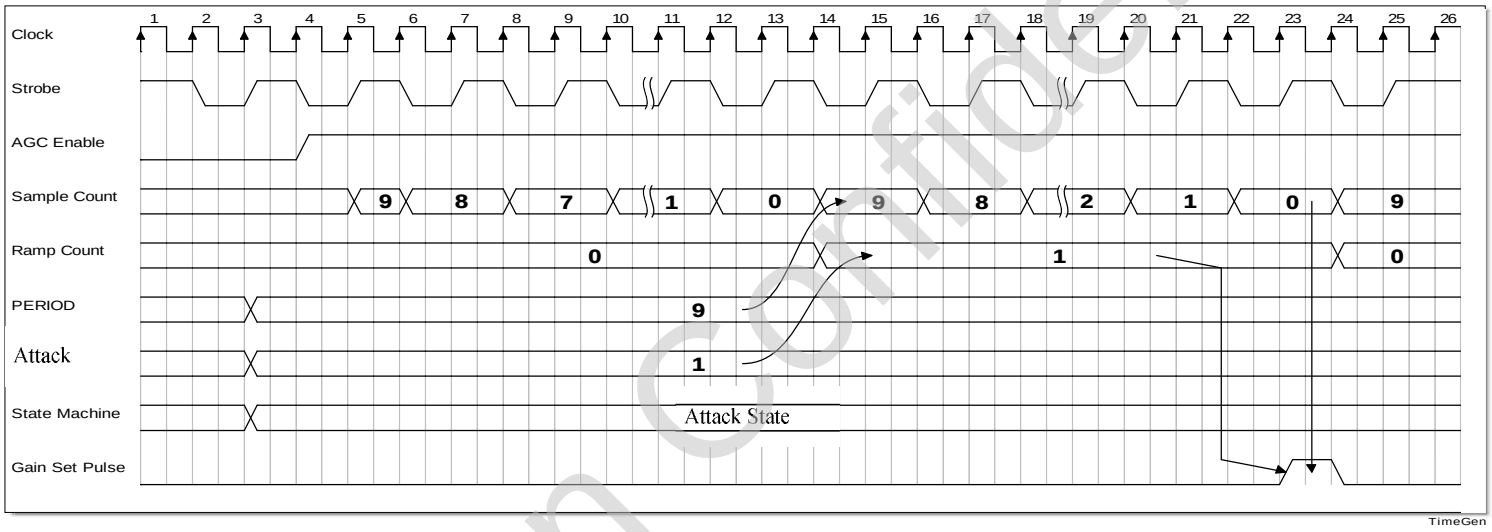
Register	Address	R/W	Description	Reset Value
AGC_CON	ADC_BA+0x034	R/W	AGC Control Register	0x00C0_0333

31	30	29	28	27	26	25	24
NG_EN	AGC_EN	Reserved				PERIOD[9:8]	
23	22	21	20	19	18	17	16
PERIOD[7:0]							
15	14	13	12	11	10	9	8
Reserved				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 definition of the register NG_LEVEL and IN_NG_TIME times continuously. Then the gain will be forced to the NG_Gain to prevent AGC enlarging the volume and enlarging the noise.
[30]	AGC_EN	Auto gain control enable bit If AGC_EN = 0, the hardware AGC function is disable. If AGC_EN = 1, the hardware AGC function is enable. When the AGC function is disabled, the input volume is controlled by register AUDIO_VOL. Otherwise, it is controlled by AGC function.
[29:26]	Reserved	Reserved
[25:16]	PERIOD[9:0]	Maximum Peak Detector Calculation Period: PERIOD[9:0] * 0.125ms (0.125ms is because the sampling rate is 8KHz)
[15:12]	Reserved	Reserved
[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)
[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)

AGC Timing Diagram :



See the Figure. If detecting the rising edge about the AGC Enable, the Sample Count will load the period number at the same time and count down with the samples (by strobe signal). Then, if the sample count is equal to zero, it will reload the period value. And the Ramp Count will increase one. The two steps are repeated again and again until the sampling count is equal to zero and the Ramp Count is equal to ATTACH (which depends on the state machine). Then, the Ramp Count will refresh to zero and the GainSet Pulse will generate one pulse to change the gain. Therefore, the speed of the gain changing can be controlled by the PERIOD and the ATTACH (RECOVERY) values.

OP Offset Calculation regiser (OPOC)

Register	Address	R/W	Description	Reset Value
OPOC	ADC_BA+0x038	R/W	OPOC Parameter Register Setting	0x0002_00A0

31	30	29	28	27	26	25	24
MUTE_SW	OOC	OPOCM	OPOC_SW				
23	22	21	20	19	18	17	16
	Reserved					OPOC_TCSN	
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 be controlled by SW.
[30]	OOC	DC Offset Calibration Enable 1′b0: Hardware OP Offset Calculation Disable 1′b1: Hardware OP Offset Calculation Enable Using the rising edge of the signal to operate ones DC Calibration.
[29]	OPOCM	OP Offset Cancellation Method This controls the offset cancellation parameter about ADC comes from SW or HW. 1′b0: Software setting mode 1′b1: Hardware setting mode (set calibration out result)
[28:24]	OPOC_SW	OP Offset Cancellation Software Setting (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: 256 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

Delete Peak Ctrl

Register	Address	R/W	Description	Reset Value
DelPeakCtrl	ADC_BA+0x03C	R/W	Delete Peak for Touch Panel and FPGA	0x0020_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		PeakDel_EN		Reserved		PeakDelThreshold[9:8]	
7	6	5	4	3	2	1	0
PeakDelThreshold[7:0]							

Bits	Descriptions	
[31:14]	RESERVED	
[13:12]	PeakDel_EN	If the audio input data has the peak noise. This function can delete this peak noise which is bigger than Peak Threshold. 2'd3 : Enable this function 2'd0 : Disable this function
[11:10]	Reserved	
[9:0]	PeakDelThreshold	If enable the Peak Delete function, then we will find out the Peak point which difference between the neighborhoods are both bigger than the PeakDelThreshold and replace it by the average between the neighborhoods.

Noise Gate Control (NG_Ctrl)

Register	Address	R/W	Description	Reset Value
NG_Ctrl	ADC_BA+0x040	R/W	Noise Gate Control	0x0033_201D

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

Bits	Descriptions	
[31:26]	Reserved	
[25:24]	GainChgMod	Select the change gain step. 2′d0: 0.75 dB for per step. 2′d1: 1.5 dB for per step. 2′d2: 2.25 dB for per step. 2′d3: 3 dB for per step.
[23:20]	IN_NG_TIME	If detecting the power smaller than Noise Gate Threshold "IN_NG_TIME" times continuously, then the state will become Noise Gate State. And the gain will be decreased to NG_Gain.
[19:16]	OUT_NG_TIME	If in the NG state and detecting the power bigger than Noise Gate Threshold "OUT_NG_TIME" times continuously, then the state will leave Noise Gate State. And the gain will be increased to meet the target level.
[15:14]	Reserved	
[13:8]	NG_Gain	If Noise Gate situation happens, the Gain will be decreased to this level.
[7]	Reserved	
[6:0]	NG_Level	The Noise Gate Threshold Setting. 7′h7F：-0.00 dB 7′h7E：-0.75 dB

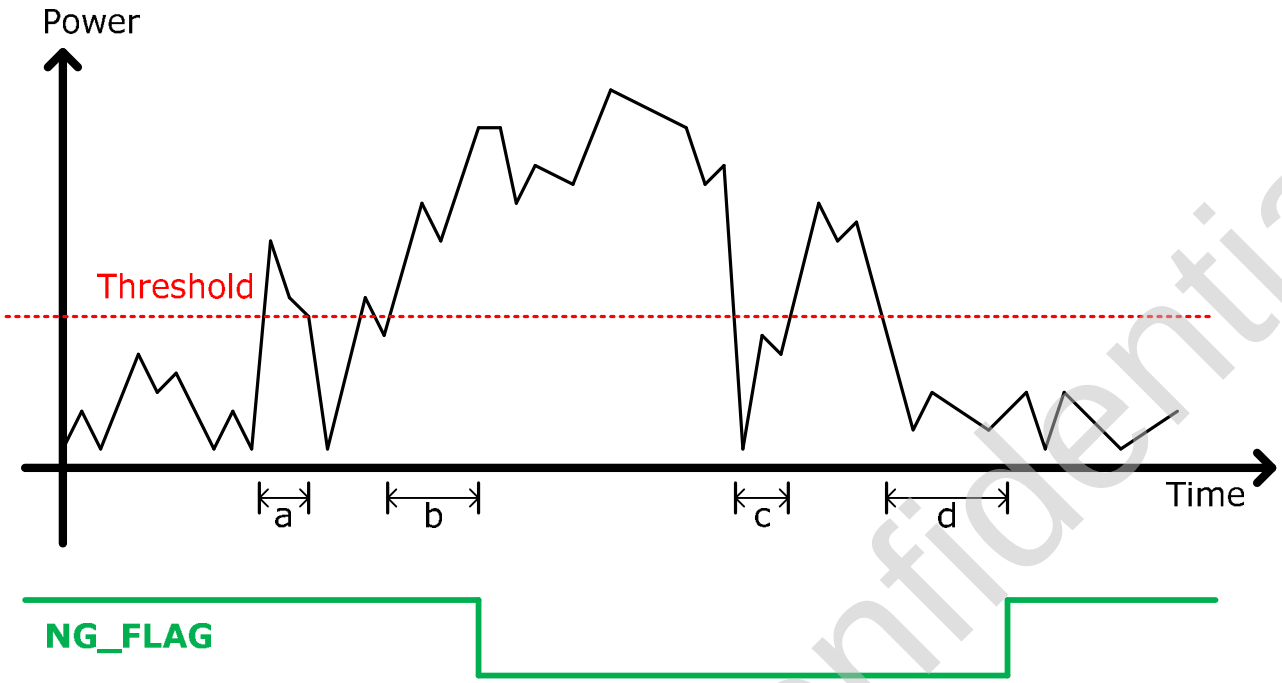
		7'h7D：-1.50 dB ….

NG_Gain :

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

Noise Gate Detection Block

Seeing the picture below, the red line 'Threshold' is controlled by the NG_Level. If the power is smaller than the threshold, it means the time is noise period. So, if the power is bigger than the threshold, it is the signal period. Therefore, in the picture, the stare time is noise, and the NG_FLAG should be set to high. Then, the power with the period 'a' is over the threshold, but the period 'a' is smaller than the register OUT_NG_TIME. So, it is still in the NG state. Following, the period 'b' is larger than the OUT_NG_TIME. Therefore, it will leave the NG state, and the NG_FLAG should be set to low. Equally, the period 'c' is smaller than IN_NG_TIME, so it will not go into the NG state, until the period 'd'. Therefore, we can control the threshold, OUT_NG_TIME and IN_NG_TIME to change the Noise Gate Detection sensitivity.



State Flag

Register	Address	R/W	Description	Reset Value
StateFlag	ADC_BA+0x044	R	Some State Situation and Flag	0x0000_2500

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

Bits	Descriptions	
[31]	OP_MUTE_FLAG	OPOC Calibration Period Flag. 1′b1: The Calibration process is going. 1′b0: the Calibration process is finish.
[30]	NG_FLAG	Noise Gate State FLAG. 1′b1: this time should be detected as noise 1′b0: this time should be detected as voice
[29:21]	Reserved	
[21:16]	OPOC_CAL_NUM	OPOC Calibration Result. It shows the DC Bias selecting for ADC Analog IP.
[15:14]	Reserved	
[13:8]	AGC_GAIN_NUM	AGC Function to adjust the gain result. It shows the gain selection for ADC Analog IP.
[7:0]	Reserved	

AGC_Moving Average Sum

Register	Address	R/W	Description	Reset Value
AGC_MAVSum	ADC_BA+0x048	R	AGC Moving Average Sum	0x0000_0000

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

Bits	Descriptions	
[31:0]	AGC_MovAvgSum	AGC Moving Average Sum. It means the input data power information, and is for turning the gain (Auto Gain Control Use).

OverFlow_Detection Control

Register	Address	R/W	Description	Reset Value
OvFw_Ctrl	ADC_BA+0x04C	R/W	OverFlow Control	0x3FFF_0000

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

Bits	Descriptions	
[31:16]	OvFw_ThresHold	The input audio power overflow threshold.
[15:1]	Reserved	
[0]	OvFw_En	OverFlow Detection Function Enable: 1′b1: Enable 1′b0: Disable When the function is enabled, if finding the input audio signal power is greater than OvFw_Threshold, the AGC gain will be decreased 9dB with each changing gain step. It can avoid having large time which the input audio signal is overflow. Then, if the input power is not greater than threshold, the gain changing step should be the same as original. Note: This function is useful when AGC is opened.

DC Offset Calculation Sum

Register	Address	R/W	Description	Reset Value
OPOC_CalSum	ADC_BA+0x050	R	DC Offset Calculation Sum	0x0000_0000

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

Bits	Descriptions	
[31]	BIAS_MOD	Touch Panel Input Data Bias Modify Mod If 1′b1 : Input Data will be decreased by the BIAS_NUM If 1;b0: Input Data will be increased by the BIAS_NUM
[30:24]	BIAS_NUM	Touch Panel Input Data Bias Modify Number
[23:10]	Reserved	
[9:0]	OPOC_CalSum	DC Offset Calculation Sum, When processing the OPOC calibration, the finial calculation sum value would be set this. SW can read the information for SW DC Bias adjusting. The OPOC_CalSum is read only.

This is used to compensate the Touch Panel ADC bias. So, first step, you select the ADC source channel to VDD, then do one times calibration (Address 0x54). Then you read the value from OPOC_CalSum (Address 0x50), and save it. Second step, you select the ADC source channel to VSS, and do the calibration again.

Now, you have two values which are the bias for the VDD and VSS. Therefore, you can use them to adjust the register values BIAS_MOD and BIAS_NUM.

Noise Calibration

Register	Address	R/W	Description	Reset Value
NoiseCalibra	ADC_BA+0x054	R/W	Noise Calibration Function	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						RESERVED	
7	6	5	4	3	2	1	0
NoiseCalibra_En	NoiseCalibraOut						

Bits	Descriptions	
[31:8]	RESERVED	
[7]	NoiseCalibra_En	Noise Calibration Enable 1′b1:Enable 1′b0:Disable This bit will be auto cleared, if the calibration process is finished.
[6:0]	NoiseCalibraOut	Noise Calibration Output Result If enable Noise Calibration Process, HW will sum the 256 samples data as the average power about this period. Then, the result will be re-mapped to the "Noise Level" parameter. The NoiseCalibraOut is read only.

4th Touch Screen MAV Data

Register	Address	R/W	Description	Reset Value
TSC_SORT4	ADC_BA+0x070	R	4th Touch Screen Data	0x0000_0000

31	30	29	28	27	26	25	24
TOUCH_EN_4	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]	TOUCH_EN_4	This 4th data is valid or not 1′b1: Valid 1′b0: Not Valid This bit should write "1" to clear.
[30:26]	RESERVED	RESERVED
[25:16]	Y_MAV4	4th MAV Y-Data
[15:10]	RESERVED	RESERVED
[9:0]	X_MAV4	4th MAV X-Data

3rd Touch Screen MAV Data

Register	Address	R/W	Description	Reset Value
TSC_SORT3	ADC_BA+0x074	R	3rd Touch Screen Data	0x0000_0000

31	30	29	28	27	26	25	24
TOUCH_EN_3	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]	TOUCH_EN_3	This 3rd data is valid or not 1′b1: Valid 1′b0: Not Valid This bit should write "1" to clear.
[30:26]	RESERVED	RESERVED
[25:16]	Y_MAV3	3rd MAV Y-Data
[15:10]	RESERVED	RESERVED
[9:0]	X_MAV3	3rd MAV X-Data

2nd Touch Screen MAV Data

Register	Address	R/W	Description	Reset Value
TSC_SORT2	ADC_BA+0x078	R	2nd Touch Screen Data	0x0000_0000

31	30	29	28	27	26	25	24
TOUCH_EN_2	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]	TOUCH_EN_2	This 2nd data is valid or not 1′b1: Valid 1′b0: Not Valid This bit should write "1" to clear.
[30:26]	RESERVED	RESERVED
[25:16]	Y_MAV2	2rd MAV Y-Data
[15:10]	RESERVED	RESERVED
[9:0]	X_MAV2	2rd MAV X-Data

1st Touch Screen MAV Data

Register	Address	R/W	Description	Reset Value
TSC_SORT1	ADC_BA+0x07C	R	1st Touch Screen Data	0x0000_0000

31	30	29	28	27	26	25	24
TOUCH_EN_1	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]	TOUCH_EN_1	This 1st data is valid or not 1′b1: Valid 1′b0: Not Valid This bit should write "1" to clear.
[30:26]	RESERVED	RESERVED
[25:16]	Y_MAV1	1st MAV Y-Data
[15:10]	RESERVED	RESERVED
[9:0]	X_MAV1	1st MAV X-Data

Touch Screen MAV X-Data Output

Register	Address	R/W	Description	Reset Value
TSC_MAV_X	ADC_BA+0x080	R	Touch Screen Average X-Data Output	0x0000_0000

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

Bits	Descriptions	
[31]	Touch Enable Flag	If it is '1', it means this average data is valid. Otherwise, if it is '0', it means this average data can't be use. And this bit is cleared by writing '1'.
[30:10]	Reserved	RESERVED
[9:0]	X_MAV_Data	Average X-Data Output

Touch Screen MAV Y-Data Output

Register	Address	R/W	Description	Reset Value
TSC_MAV_Y	ADC_BA+0x084	R	Touch Screen Average Y-Data Output	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_MAV_Data[9:8]	
7	6	5	4	3	2	1	0
Y_MAV_Data[7:0]							

Bits	Descriptions	
[31:10]	RESERVED	RESERVED
[9:0]	Y_MAV_Data	Average Y-Data Output

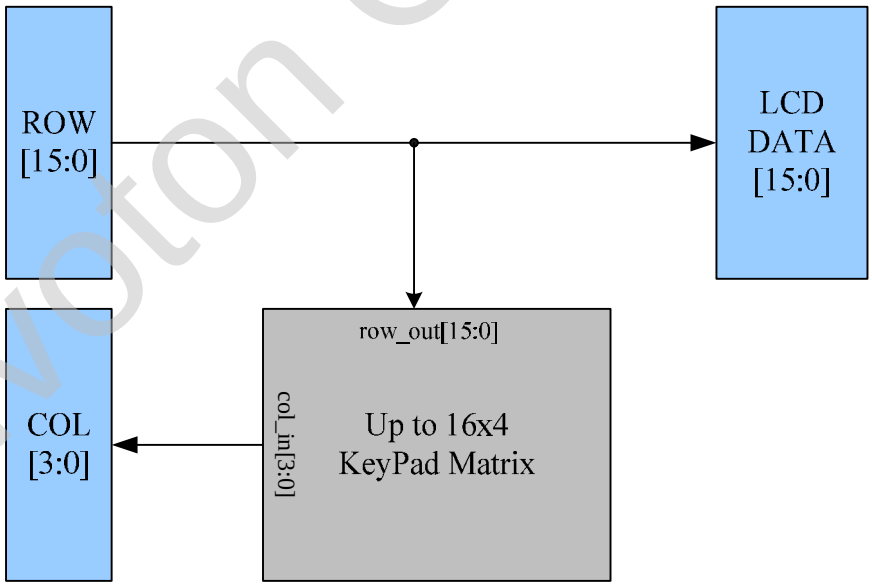
6.27 Keypad Interface (KPI)

The Keypad Interface (KPI) is an APB slave with configurable minimum 2-row up to 16-row scan output and minimum 1-column up to 4-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

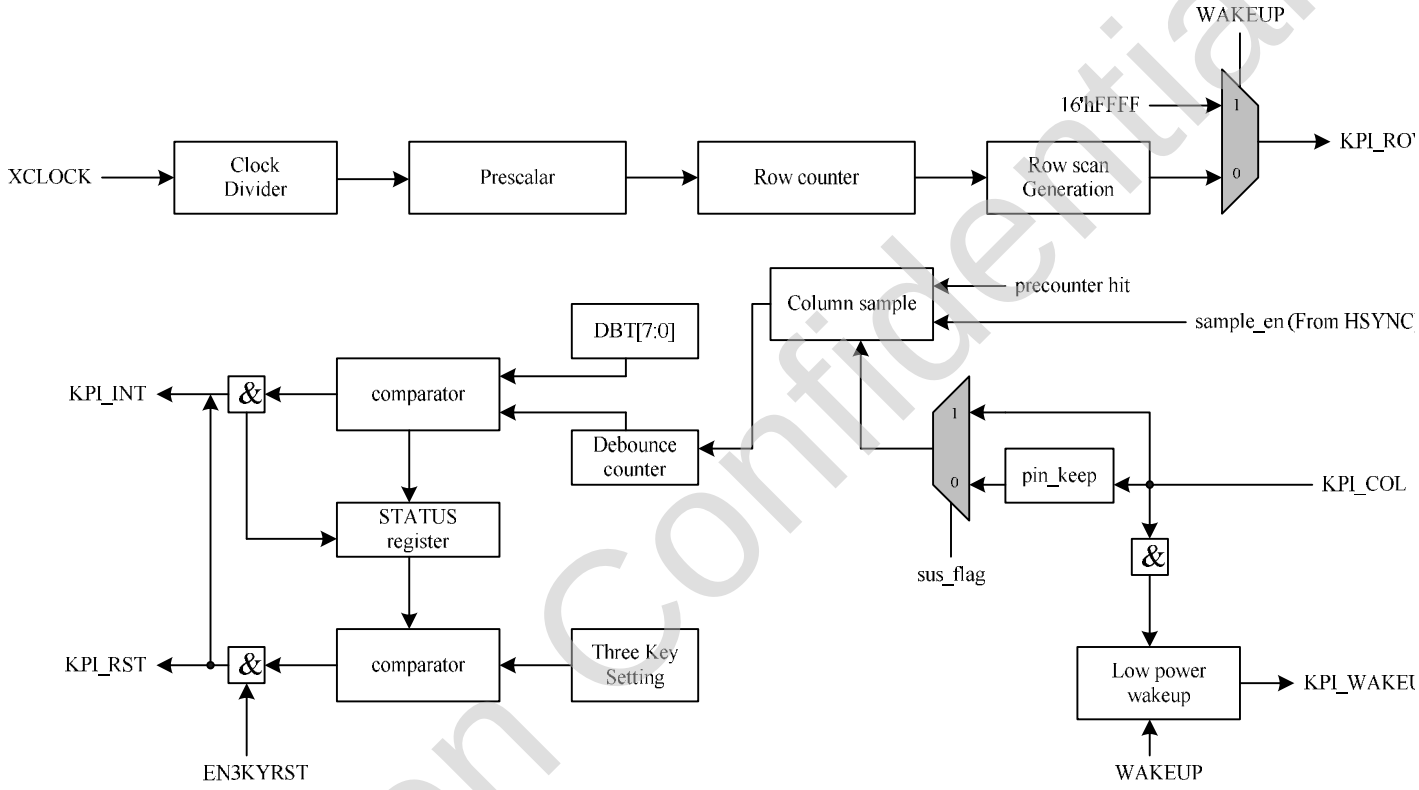
The keypad interface has the following features:

- **matrix keypad interface (maximum 16x4 array, and minimum 2x1array)**
- **programmable de-bounce time**
- **low-power wakeup mode**
- **programmable three-key reset**
- **Generate interrupt and update all the keys(maximum 64 keys, minimum 2 keys) information(press/release) every time the user pressing or releasing**
- **Support sync-type LCD 16-bit bus share to KPI.**



Keypad Interface

6.27.1 KPI Block Diagram



Keypad Controller Block Diagram

6.27.2 Keypad Interface Register Map

Register	Address	R/W	Description	Reset Value
KPI_BA = 0xB800_5000				
KPICONF	KPI_BA+0x000	R/W	Keypad configuration Register	0x0000_0000
KPI3KCONF	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	0x0000_0000
KPIKEST1	KPI_BA+0x014	R/O	Keypad state register 1	0x0000_0000
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
KPILCM	KPI_BA+0x02C	R/W	Keypad and LCM Bus Share Setting	0x0000_0100
KPISUS	KPI_BA+0x030	R/W	Keypad Suspend Mode Setting	0x000F_FFFF

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
KROW[3:0]				Reserved		KCOL[1:0]	
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

<table>
<tr><td>[31:28]</td><td>KROW</td><td>Keypad Matrix ROW number
The keypad matrix is set by ROW x COL. The ROW number can be set 2 to 16
<table>
<tr><th>KROW[31:28]</th><th>Keypad maxtrix ROW number</th></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>11</td></tr>
<tr><td>1011</td><td>12</td></tr>
<tr><td>1100</td><td>13</td></tr>
<tr><td>1101</td><td>14</td></tr>
<tr><td>1110</td><td>15</td></tr>
<tr><td>1111</td><td>16</td></tr>
</table></td></tr>
<tr><td>[27:26]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[25:24]</td><td>KCOL</td><td>Keypad Matrix COL Number
The keypad matrix is set by ROW x COL. The COL number can be set 1 to 4
<table>
<tr><th>KCOL[25:24]</th><th>Keypad maxtrix COLUMN number</th></tr>
<tr><td>00</td><td>1</td></tr>
<tr><td>01</td><td>2</td></tr>
<tr><td>10</td><td>3</td></tr>
<tr><td>11</td><td>4</td></tr>
</table></td></tr>
<tr><td>[21]</td><td>DB_EN</td><td>Scan In Signal De-bounce Enable
0 = The de-bounce function is disabled
1 = The de-bounce function is enabled</td></tr>
</table>

[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 divided number 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
[7:6]	Reserved	Reserved
[5]	WAKEUP	Lower Power Wakeup Enable Setting this bit enables low power wakeup 1 = Wakeup enable 0 = Not enable Note: Set the bit will force all KPI scan out to high.
[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 Note: the bit will be reset when KPI reset occurred.

[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 Note: the bit will be reset when KPI reset occurred.
[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 Note: the bit will be reset when KPI reset occurred.
[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 Note: the bit will be reset when KPI reset occurred.
[23:22]	Reserved	Reserved
[21:18]	K32R	The #2 Key Row Address The #2 means the row address and the column address is the highest of the specified 3-keys
[17:16]	K32C	The #2 Key Column Address
[15:14]	Reserved	Reserved
[13:10]	K31R	The #1 Key Row Address The #1 means the row address and the column address is the 2nd of the specified 3-keys
[9:8]	K31C	The #1 Key Column Address
[7:6]	Reserved	Reserved

[5:2]	K30R	The #0 Key Row Address The #0 means the row address and the column address is the lowest of the specified 3-keys
[1:0]	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
RESERVED							
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED			PKEY_INT	RKEY_INT	KEY_INT	RST_3KEY	PDWAKE

Bits	Descriptions	
[31:5]	Reserved	Reserved
[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: DWORD RKE0, RKE1; PKE0 = reg_read(KPIKRE0); PKE1 = reg_read(KPIKRE1); Reg_write(KPIKRE0,RKE0); Reg_write(KPIKRE1,RKE1);
[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	0x0000_0000

31	30	29	28	27	26	25	24
KEST73	KEST72	KEST71	KEST70	KEST63	KEST62	KEST61	KEST60
23	22	21	20	19	18	17	16
KEST53	KEST52	KEST51	KEST50	KEST43	KEST42	KEST41	KEST40
15	14	13	12	11	10	9	8
KEST33	KEST32	KEST31	KEST30	KEST23	KEST22	KEST21	KEST20
7	6	5	4	3	2	1	0
KEST13	KEST12	KEST11	KEST10	KEST03	KEST02	KEST01	KEST00

Bits	Descriptions	
[31:0]	Key state	Key State KESTm,n: m is row number, n is column number 1:Keym,n is pressing 0:keym,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	0x0000_0000

31	30	29	28	27	26	25	24
KESTF3	KESTF2	KESTF1	KESTF0	KESTE3	KESTE2	KESTE1	KEST3E0
23	22	21	20	19	18	17	16
KESTD3	KESTD2	KESTD1	KESTD0	KESTC3	KESTC2	KESTC1	KESTC0
15	14	13	12	11	10	9	8
KESTB3	KESTB2	KESTB1	KESTB0	KESTA3	KESTA2	KESTA1	KESTA0
7	6	5	4	3	2	1	0
KEST93	KEST92	KEST91	KEST90	KEST83	KEST82	KEST81	KEST80

Bits	Descriptions	
[31:0]	Key state	KESTm,n: m is row number, n is column number 1:Keym,n is pressing 0: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
KPE73	KPE72	KPE71	KPE70	KPE63	KPE62	KPE61	KPE60
23	22	21	20	19	18	17	16
KPE53	KPE52	KPE51	KPE50	KPE43	KPE42	KPE41	KPE40
15	14	13	12	11	10	9	8
KPE33	KPE32	KPE31	KPE30	KPE23	KPE22	KPE21	KPE20
7	6	5	4	3	2	1	0
KPE13	KPE12	KPE11	KPE10	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
KPEF3	KPEF2	KPEF1	KPEF0	KPEE3	KPEE2	KPEE1	KPEE0
23	22	21	20	19	18	17	16
KPED3	KPED2	KPED1	KPED0	KPEC3	KPEC2	KPEC1	KPEC0
15	14	13	12	11	10	9	8
KPEB3	KPEB2	KPEB1	KPEB0	KPEA3	KPEA2	KPEA1	KPEA0
7	6	5	4	3	2	1	0
KPE93	KPE92	KPE91	KPE90	KPE83	KPE82	KPE81	KPE80

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
KRE73	KRE72	KRE71	KRE70	KRE63	KRE62	KRE61	KRE60
23	22	21	20	19	18	17	16
KRE53	KRE52	KRE51	KRE50	KRE43	KRE42	KRE41	KRE40
15	14	13	12	11	10	9	8
KRE33	KRE32	KRE31	KRE30	KRE23	KRE22	KRE21	KRE20
7	6	5	4	3	2	1	0
KRE13	KRE12	KRE11	KRE10	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
KREF3	KREF2	KREF1	KREF0	KREE3	KREE2	KREE1	KREE0
23	22	21	20	19	18	17	16
KRED3	KRED2	KRED1	KRED0	KREC3	KREC2	KREC1	KREC0
15	14	13	12	11	10	9	8
KREB3	KREB2	KREB1	KREB0	KREA3	KREA2	KREA1	KREA0
7	6	5	4	3	2	1	0
KRE93	KRE92	KRE91	KRE90	KRE83	KRE82	KRE81	KRE80

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

PrescaleDivider (PrescaleDivider)

Register	Address	R/W	Description	Reset Value
PrescaleDivider	KPI_BA+0x028	R/W	Prescale divider	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
Prescale 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 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

KPILCM

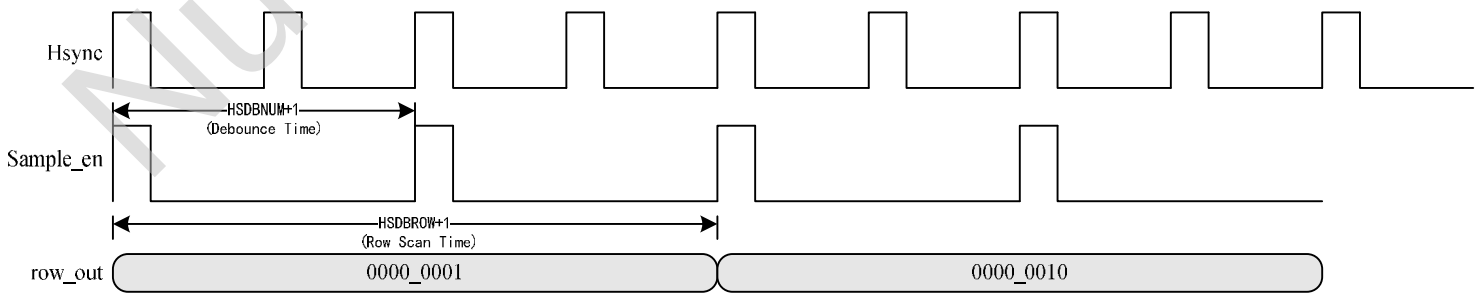
Register	Address	R/W	Description	Reset Value
KPILCM	KPI_BA+0x02C	R/W	Keypad and LCM Bus Share Setting	0x0000_0100

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

Bits	Descriptions	
[31]	LCMMODE	LCM Mode Enable 0: LVDATA[15:0] are always LCD output. 1: KPI Scan-out output from LVDATA[15:0], share with LCD bus.
[30]	KPI_8BIT	KPI 8Bit mode
[29:12]	Reserved	Reserved
[11:8]	HSDBROW	Row scan time (Number of sample_en, must >=1)
[7:0]	HSDBNUM	Debounce time (Number of Hsync pulse)

Figure 6.27-1 Debounce setting in LCD Hsync mode.

For example: HSDBNUM=1, HSDBROW=1



KPISUS

Register	Address	R/W	Description	Reset Value
KPISUS	KPI_BA+0x030	R/W	Keypad Suspend Mode Setting	0x000F_FFFF

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

Bits	Descriptions	
[31]	SUSFORCE	0: Normal detection mode (decide by SUSCNUM) 1: Force KPI into suspend mode.
[30:20]	Reserved	Reserved
[19:0]	SUSCNUM	suspend mode detection (Number of cycles) If Hsync cannot be detected for SUSCNUM cycles, KPI will switch LCM mode to suspend mode until Hsync is detected again. Detect Time = KPI_CLOCK_PERIOD x SUSCNUM In LCM mode: Refer to all setting in 0x2C to 0x30. In Suspend mode: (Self-Run) Refer to all setting in 0x0 to 0x28.

6.28 AES Accelerator

6.28.1 Overview

The AES accelerator in W55FA95 is a fully compliant implementation of the AES (Advance Encryption Standard) algorithm. Such accelerator supports both encryption and decryption. The AES accelerator can be used in different data security applications, such as secure communications, that need to provide cryptographic protection.

The AES accelerator supports the DMA function to reduce the CPU's intervention. For DMA function, two burst lengths, the 8-word and 4-word, are supported.

6.28.2 Features

- **Supports both encryption and decryption.**
- **Supports only CBC (Cipher Block Chaining) mode.**
- **All three kinds of key length, 128, 192, and 256 bits, are supported.**
- **Built-in DMA function.**
- **Two 48 bytes internal FIFO.**

6.28.3 AES Engine

6.28.3.1 Introduction

AES standard specifies the **Rijndael** algorithm, s symmetric block cipher that can process data block of 128 bits, using cipher keys with length of 128, 192, and 256 bits.

The algorithm may be used with three different key lengths indicated above, and therefore these different flavors may be referred as “AES-128”, “AES-192” and “AES-256”.

6.28.3.2 AES Operation

For the AES algorithm, the length of the **Cipher Key,** K, is 128, 192, or 256 bits. The key length is represented by Nk=4, 6, or 8, whicj reflects the number of 32-bit words (number of columns) in the Cipher Key.

During the execution of algorithm, the number of rounds to be performed is dependent on the key size. The number of rounds is represented by Nr. The relationship between key size, block size and number of rounds are shown in following table,

	Key Length (Nk Words)	Block Size (Nb Words)	Number of Rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Where Word=32 bits

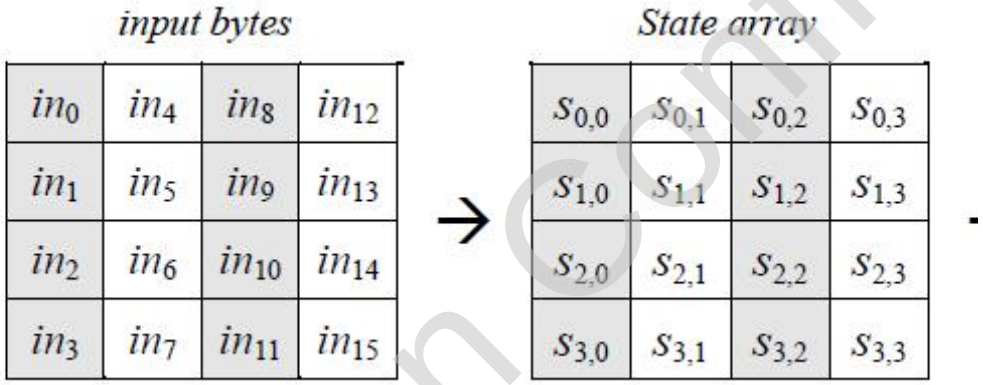
For both Cipher and Inverse Cipher, the AES algorithm use a round function that is composed of four different byte-oriented transformations:

- Byte substitution using a substitution table called S-box.
- Shifting row of State array by different offsets
- Mixing data within each column of State array
- Adding a Round Key to the state.

These transformations will be described in the following section.

6.28.3.3 Cipher

At the start of Cipher, the input is copied to the state array as following



Where

in_x: x-th byte of 128-bit input,

$S_{(x,y)}$: (x-th row, y-th column) of the state array

$s[r, c] = in[r + 4c]$ for $0 \le r < 4$ and $0 \le c < Nb$,

After an initial Round Key addition, the state array is transformed by implementing a round function 10, 12 or 14 times (depending on the Key Length) with final round differing slightly from the first (Nr-1) rounds. The final State is then copied to the output as following

State array

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

→

output bytes

out_0	out_4	out_8	out_{12}
out_1	out_5	out_9	out_{13}
out_2	out_6	out_{10}	out_{14}
out_3	out_7	out_{11}	out_{15}

out_x: x-th byte of 128-bit output,

$S_{(x,y)}$: (x-th row, y-th column) of the state array

$out[r + 4c] = s[r, c]$ for $0 \le r < 4$ and $0 \le c < Nb$.

Whole Cipher procedure is described in flowing pseudo code and individual transformations – SubBytes(), ShiftRows(), MixColumns(), and AddRoundKey()- process the State and are described in the following subsections.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[0, Nb-1])

   for round = 1 step 1 to Nr-1
      SubBytes(state)
      ShiftRows(state)
      MixColumns(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for

   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   out = state
end
```

6.28.3.3.1 SubByes Transformation

The SubBytes() transformation is a non-linear byte substitution that operates independently on each byte of the State using following substitution table (S-box). The S-box which is invertible, used in the SubBytes() transformation is presented in Hex form. For example, if $S_{(1,1)}=\{5,3\}$, then the substitution value would be determined by the intersection of 5th row and 3th column. This would result in $S'_{(1,1)}=\{e,d\}$.

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

. S-box: substitution values for the byte xy (in hexadecimal format).

6.28.3.3.2 ShiftRows Transformation

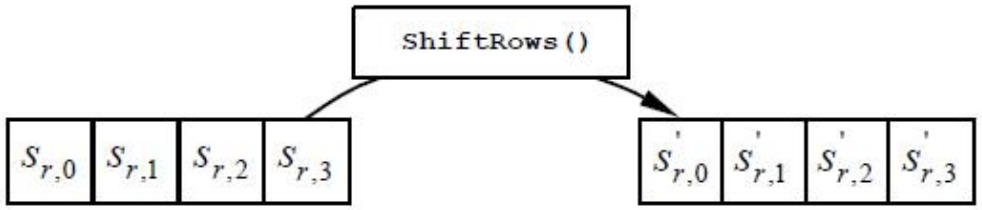
In the ShiftRows transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets). The first row, r=0, is not shifted. Specially, the ShiftRows transformation proceeds as follows,

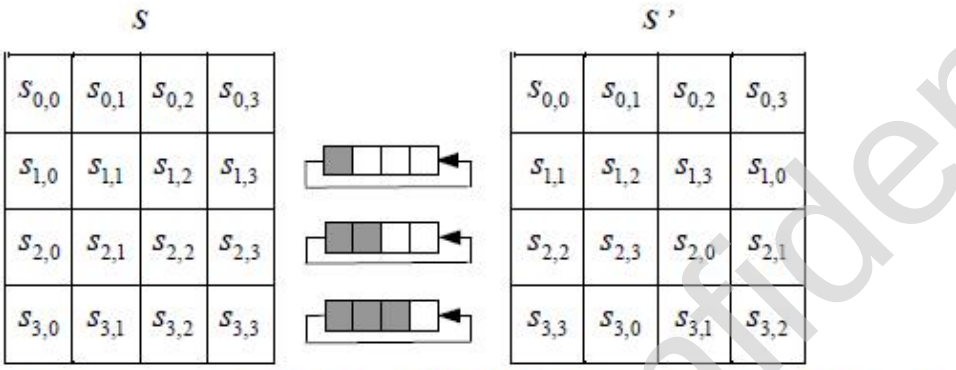
$$s'_{r,c} = s_{r,(c+shift(r,Nb)) \bmod Nb} \quad \text{for } 0 < r < 4 \quad \text{and} \quad 0 \le c < Nb,$$

where the shift value $shift(r,Nb)$ depends on the row number, r, as follows (recall that $Nb = 4$):

$$shift(1,4) = 1; \quad shift(2,4) = 2; \quad shift(3,4) = 3.$$

Following figures illustrates the Shuftrows transformation.





ShiftRows() cyclically shifts the last three rows in the State.

6.28.3.3.3 MixColumns Transformation

The MixColumn transformation operates on the State column-by-column, this can be written as a matrix multiplication. Let

$$s'(x) = a(x) \otimes s(x):$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb.$$

As a result of this multiplication, the four bytes in a column are replaced by the following:

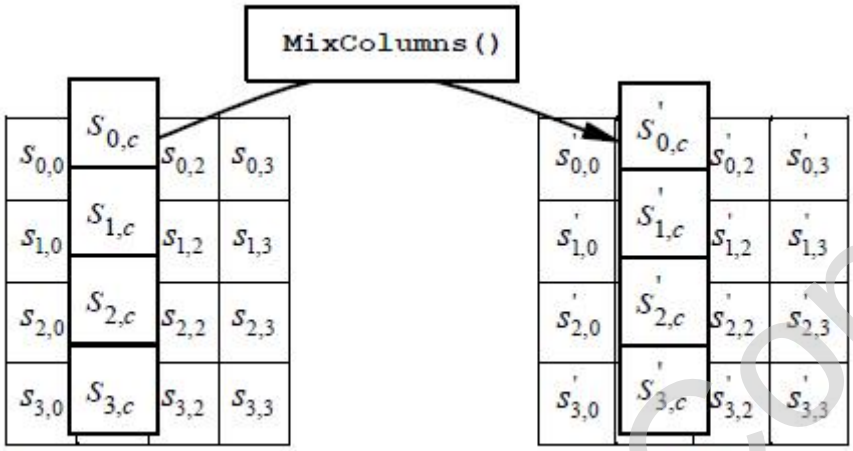
$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

Next figure shows the MixColumn transformation



9. MixColumns() operates on the State column-by-column.

6.28.3.3.4 AddRoundKey Transformation

In the AddRoundKey transformation, a Round Key is added to the State by a simple bitwise XOR operation. Each Round Key consists of Nb words from the key schedule (described in 1.1.4). The Nb words are each added into the column of the state, such that

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{round*Nb+c}] \quad \text{for } 0 \le c < Nb,$$

Where [W_i] are the key schedule words and round is a value in the range $0 \le round \le Nr$

In the Cipher, the initial Round Key addition occurs when round=0, prior to the first application of round function.. The application of AddRoundKey transformation to the Nr round of Cipher occurs when $1 \le round \le Nr$ **.**

6.28.3.4 Key Expansion

The AES algorithm takes the Cipher Key, K, and performs a Key Expansion routine to generate a key schedule. The Key Expansion generates a total of Nb(Nr+1) words: the algorithm requires an initial set of Nb words, and each of the Nr rounds require Nb words of key data. The resulting key schedule consists of a linear array of 4-byte words, denoted [w_i], with i in the range $0 \le round \le Nb*(Nr+1)$

The expansion of the input key into the key schedule proceeds according to the pseudo code in following,

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word  temp

   i = 0

   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while

   i = Nk

   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
         temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end

Note that Nk=4, 6, and 8 do not all have to be implemented;
they are all included in the conditional statement above for
conciseness.   Specific implementation requirements for the
Cipher Key are presented in Sec. 6.1.
```

Pseudo Code for Key Expansion.[2]

- **SubWord is a function that takes a four-byte input word and applied the S-box (described above) to each of the four bytes to produce an output word.**
- **RotWord takes a word [a0, a1, a2, a3] as input perform a cyclic permutation and returns the word[a1,a3,a3,a0].**
- **The round constant word array, Rcon[i], contains the valus given by [x^{i-1}, {00},{00},{00}], with x^{i-1} being the powers of x (x is denoted as {0 2}).**

From above pseudo code, it can be seen that the first Nk words of the expanded key are filled with the Cipher Key.Every following word, w[i], is equal to the XOR of previous word, w[i-1], and the word NK position earlier, w[i-Nk]. For words in position that are multiple of Nk, a transformation is applied to w[i-1] prior to the XOR, followed by an XOR with a round constant, Rcon[i]. This transformation consists of a cyclic shift of the byte in a

word (RotWord), followed by the application of a table lookup to all four bytes of the word (SubWord).

It is important to note that the Key Expansion routine for 256-bit Cipher Keys (Nk=8) is slightly different than 128- and 192-bit Cipher Keys. If Nk=8 and i-4 is a multiple of Nk, the SubWord is applied to w[i-1] prior to the XOR.

6.28.3.5 Decipher

The above Cipher transformation can be inverted and then implemented in inverse order to produce a straightforward Decipher for the AES algorithm. The individual transformations used in the Decipher – InvShiftRows(), InvSubBytes(), InvMixColumn(), and AddRoundKey() – process the State and are described in following subsections.

Following pseudo code describes the Decipher. In this pseudo code, the array w[] contains the key schedule that was described in above section.

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   for round = Nr-1 step -1 downto 1
      InvShiftRows(state)
      InvSubBytes(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
      InvMixColumns(state)
   end for

   InvShiftRows(state)
   InvSubBytes(state)
   AddRoundKey(state, w[0, Nb-1])

   out = state
end
```

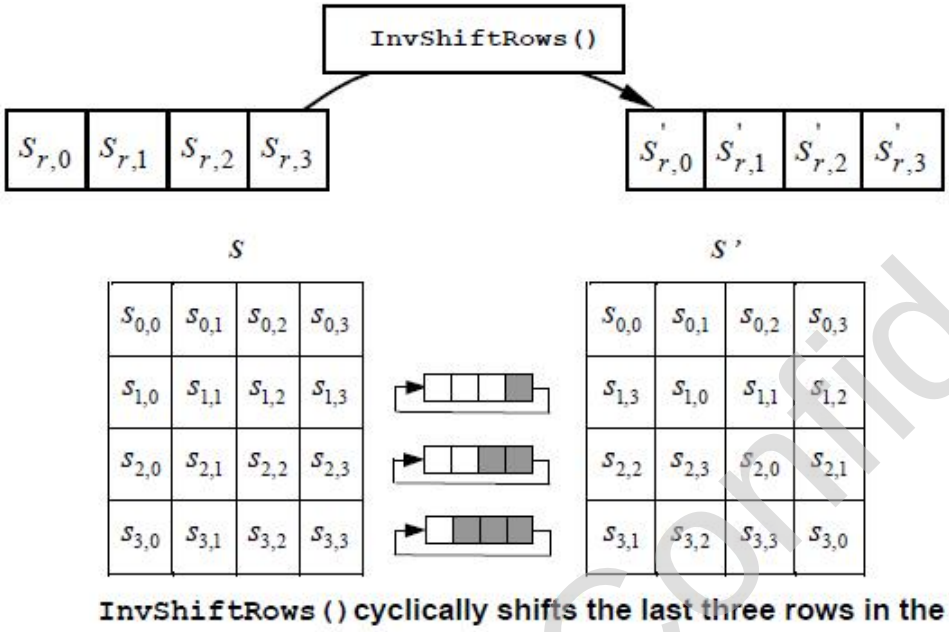
Pseudo code for Decipher

6.28.3.5.1 InvShiftRows Transformation

InvShiftRows is the inverse of the ShiftRows transformation. The bytes in the last three rows of the State are

cyclically shifted over different numbers of bytes. The first row, r=0, is not shifted. The bottom three rows are cyclically shifted by (Nb-shift(r,Nb)) bytes, where the shift value shuft(r,Nb) depends on the row number and is given in Sec.1.1.3.3.2.

Following figures shows the InvShiftRows transformation.



InvShiftRows() cyclically shifts the last three rows in the State.

6.28.3.5.2 InvSubBytes Transformation

InvSubBytes is the inverse of the byte substitution transformation, in which the inverse S-box is applied to each byte of the State. The inverse S-box used in the InvSubBytes transformation is presented in following table.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Inverse S-box: substitution values for the byte xy (in hexadecimal format).

6.28.3.5.3 InvMixColumns Transformation

InvMixColumns is the inverse of the MixColumns transformation. InvMixColumns operates on the State column-by-column. Same as describe in Sec. 1.1.3.3.3, the InvMixColumn can be also written as a matrix multiplication,

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb.$$

As a result of this multiplication, the four bytes in a column are replaced by the following:

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$

$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

6.28.3.5.4 Inverse of AddRoundKey Transformation

The AddRoundKey transformation that is described in Sec. 1.1.3.3.4 is its own inverse.

6.28.4 AES Engine Control Register Map

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

AES_BASE=0xB100_D000

Register	Address	R/W	Description	Reset Value
AESKW0R	AES_BASE+0x000	R/W	AES Key Word 0 Register	0x0000_0000
AESKW1R	AES_BASE+0x004	R/W	AES Key Word 1 Register	0x0000_0000
AESKW2R	AES_BASE+0x008	R/W	AES Key Word 2 Register	0x0000_0000
AESKW3R	AES_BASE+0x00C	R/W	AES Key Word 3 Register	0x0000_0000
AESKW4R	AES_BASE+0x010	R/W	AES Key Word 4 Register	0x0000_0000
AESKW5R	AES_BASE+0x014	R/W	AES Key Word 5 Register	0x0000_0000
AESKW6R	AES_BASE+0x018	R/W	AES Key Word 6 Register	0x0000_0000
AESKW7R	AES_BASE+0x01C	R/W	AES Key Word 7 Register	0x0000_0000
AESIV0R	AES_BASE+0x020	R/W	AES Initial Vector Word 0 Register	0x0000_0000
AESIV1R	AES_BASE+0x024	R/W	AES Initial Vector Word 1 Register	0x0000_0000
AESIV2R	AES_BASE+0x028	R/W	AES Initial Vector Word 2 Register	0x0000_0000
AESIV3R	AES_BASE+0x02C	R/W	AES Initial Vector Word 3 Register	0x0000_0000
AESCR	AES_BASE+0x030	R/W	AES Control Register	0x0000_0000
AESSAR	AES_BASE+0x034	R/W	AES Source Address Register	0x0000_0000
AESDAR	AES_BASE+0x038	R/W	AES Destination Address Register	0x0000_0000
AESBCR	AES_BASE+0x03C	R/W	AES Byte Count Register	0x0000_0000
AESISR	AES_BASE+0x040	R/W	AES Interrupt Status Register	0x0000_0000
AESIER	AES_BASE+0x044	R/W	AES Interrupt Enable Register	0x0000_0000
ACSAR	AES_BASE+0x048	R	AES Current Source Address Register	0x0000_0000
ACDAR	AES_BASE+0x04C	R	AES Current Destination Address Register	0x0000_0000
ACBCR	AES_BASE+0x050	R	AES Current Byte Count Register	0x0000_0000

6.28.5 AES Engine Control Register

AES Key Word 0 Register (AESKW0R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 31~0 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW0R	AES_BASE+0x000	R/W	AES Key Word 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW0							
23	22	21	20	19	18	17	16
AESKW0							
15	14	13	12	11	10	9	8
AESKW0							
7	6	5	4	3	2	1	0
AESKW0							

Bits	Descriptions	
[31:0]	AESKW0	AES Key Word 0 The AESKW0 keeps the bit 31~0 of security key for AES operation.

AES Key Word 1 Register (AESKW1R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 63~32 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW1R	AES_BASE+0x004	R/W	AES Key Word 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW1							
23	22	21	20	19	18	17	16
AESKW1							
15	14	13	12	11	10	9	8
AESKW1							
7	6	5	4	3	2	1	0
AESKW1							

Bits	Descriptions	
[31:0]	AESKW1	AES Key Word 1 The AESKW1 keeps the bit 63~32 of security key for AES operation.

AES Key Word 2 Register (AESKW2R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 95~64 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW2R	AES_BASE+0x008	R/W	AES Key Word 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW2							
23	22	21	20	19	18	17	16
AESKW2							
15	14	13	12	11	10	9	8
AESKW2							
7	6	5	4	3	2	1	0
AESKW2							

Bits	Descriptions	
[31:0]	AESKW2	AES Key Word 2 The AESKW2 keeps the bit 95~64 of security key for AES operation.

AES Key Word 3 Register (AESKW3R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 127~96 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW3R	AES_BASE+0x00C	R/W	AES Key Word 3 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW3							
23	22	21	20	19	18	17	16
AESKW3							
15	14	13	12	11	10	9	8
AESKW3							
7	6	5	4	3	2	1	0
AESKW3							

Bits	Descriptions	
[31:0]	AESKW3	AES Key Word 3 The AESKW3 keeps the bit 127~96 of security key for AES operation.

AES Key Word 4 Register (AESKW4R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 159~128 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW4R	AES_BASE+0x010	R/W	AES Key Word 4 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW4							
23	22	21	20	19	18	17	16
AESKW4							
15	14	13	12	11	10	9	8
AESKW4							
7	6	5	4	3	2	1	0
AESKW4							

Bits	Descriptions	
[31:0]	AESKW4	AES Key Word 4 The AESKW4 keeps the bit 159~128 of security key for AES operation.

AES Key Word 5 Register (AESKW5R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 191~160 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW5R	AES_BASE+0x014	R/W	AES Key Word 5 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW5							
23	22	21	20	19	18	17	16
AESKW5							
15	14	13	12	11	10	9	8
AESKW5							
7	6	5	4	3	2	1	0
AESKW5							

Bits	Descriptions	
[31:0]	AESKW5	AES Key Word 5 The AESKW5 keeps the bit 191~160 of security key for AES operation.

AES Key Word 6 Register (AESKW6R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 223~192 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW6R	AES_BASE+0x018	R/W	AES Key Word 6 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW6							
23	22	21	20	19	18	17	16
AESKW6							
15	14	13	12	11	10	9	8
AESKW6							
7	6	5	4	3	2	1	0
AESKW6							

Bits	Descriptions	
[31:0]	AESKW6	AES Key Word 6 The AESKW6 keeps the bit 223~192 of security key for AES operation.

AES Key Word 7 Register (AESKW7R)

The security key for AES accelerator is 128, 192 or 256 bits and then four, six or eight 32-bits registers are needed to store each security key. This register is used to keep the bit 255~224 of security key for AES operation.

{AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 128 bits security key for AES operation.

{AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 192 bits security key for AES operation.

{AESKW7R, AESKW6R, AESKW5R, AESKW4R, AESKW3R, AESKW2R, AESKW1R, AESKW0R} indicates the 256 bits security key for AES operation.

Register	Address	R/W	Description	Reset Value
AESKW7R	AES_BASE+0x01C	R/W	AES Key Word 7 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESKW7							
23	22	21	20	19	18	17	16
AESKW7							
15	14	13	12	11	10	9	8
AESKW7							
7	6	5	4	3	2	1	0
AESKW7							

Bits	Descriptions	
[31:0]	AESKW7	AES Key Word 7 The AESKW7 keeps the bit 255~224 of security key for AES operation.

AES Initial Vector Word 0 Register (AESIV0R)

While AES is in CBC mode, following four initial vectors (AESIV0R, AESIV1R, AESIV2R, AESIV3R) will be needed. These four initial vectors will be latched when turn on AES core.

Register	Address	R/W	Description	Reset Value
AESIV0R	AES_BASE+0x020	R/W	AES Initial Vector Word 0 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESIV0							
23	22	21	20	19	18	17	16
AESKIV0							
15	14	13	12	11	10	9	8
AESIV0							
7	6	5	4	3	2	1	0
AESIV0							

Bits	Descriptions	
[31:0]	AESIV0	AES Initial Vector Word 0

AES Initial Vector Word 1 Register (AESIV1R)

Register	Address	R/W	Description	Reset Value
AESIV1R	AES_BASE+0x024	R/W	AES Initial Vector Word 1 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESIV1							
23	22	21	20	19	18	17	16
AESKIV1							
15	14	13	12	11	10	9	8
AESIV1							
7	6	5	4	3	2	1	0
AESIV1							

Bits	Descriptions	
[31:0]	AESIV1	AES Initial Vector Word 1

AES Initial Vector Word 2 Register (AESIV2R)

Register	Address	R/W	Description	Reset Value
AESIV2R	AES_BASE+0x028	R/W	AES Initial Vector Word 2 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESIV2							
23	22	21	20	19	18	17	16
AESKIV2							
15	14	13	12	11	10	9	8
AESIV2							
7	6	5	4	3	2	1	0
AESIV2							

Bits	Descriptions	
[31:0]	AESIV2	AES Initial Vector Word 2

AES Initial Vector Word 3 Register (AESIV3R)

Register	Address	R/W	Description	Reset Value
AESIV1R	AES_BASE+0x02C	R/W	AES Initial Vector Word 3 Register	0x0000_0000

31	30	29	28	27	26	25	24
AESIV3							
23	22	21	20	19	18	17	16
AESKIV3							
15	14	13	12	11	10	9	8
AESIV3							
7	6	5	4	3	2	1	0
AESIV3							

Bits	Descriptions	
[31:0]	AESIV3	AES Initial Vector Word 3

AES Control Register (AESCR)

The AESCR implements different configurations and control bits for AES operation. The key size selection, encryption/decryption selection and operation enable are all defined in this register.

Register	Address	R/W	Description	Reset Value
AESCR	AES_BASE+0x030	R/W	AES Control 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
		TRANS	Reserved	KSIZE		ENCRPT	AESON

Bits	Descriptions	
[31:6]	Reserved	
5	TRANS	data format transform Setting this bit high will swap {byte3, byte2, byte1, byte0} to {byte0, byte1, byte2, byte3}
4	Reserved	
[3:2]	KSIZE	Key Size The KSIZE defines three different key sizes for AES operation. 2'b00: 128 bits key 2'b01: 192 bits key 2'b10: 256 bits key 2'b11: Reserved The KSIZE can be read and written. But write to KSIZE during the AES accelerator is doing AES operation, the AESON is enabled, has no effect and the value of KSIZE will not be updated.

[1]	ENCRPT	**Encryption** The ENCRPT defines the encryption or decryption for AES operation. Set ENCRPT to 1 enables the AES accelerator to do AES encryption operation while 0 enables the AES accelerator to do AES decryption operation. The ENCRPT can be read and written. But write to ENCRPT during the AES accelerator is doing AES operation, the AESON is enabled, has no effect and the value of ENCRPT will not be updated. 1′b1: The AES operation is encryption. 1′b0: The AES operation is decryption.
[0]	AESON	**AES On** The AESON controls the enable of AES operation. Set AESON to 1 enables the AES accelerator to do AES operation. Clear AESON to 0 terminates the AES operation. Clear and then set AESON again makes the AES accelerator reload the starting address, destination address, data byte count, key size selection and encryption/decryption selection, and then restart the AES operation. If the AESON is cleared during the AES accelerator is doing AES operation, the operation will be terminated after the current 128-bit AES operation has finished.

AES Source Address Register (AESSAR)

The AES accelerator supports DMA function to transfer the plain text between system memory and embedded FIFO. The AESSAR keeps source address of the data buffer where the plain text is stored. Based on the source address, the AES accelerator can read the plain text from system memory and do AES operation. The value of AESSAR and AESDAR can be the same.

Register	Address	R/W	Description	Reset Value
AESSAR	AES_BASE+0x034	R/W	AES Source Address Register	

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

Bits	Descriptions	
[31:0]	AESSA	**AES Source Address** The AESSA keeps the source address of the data buffer where the plain text is stored. The start of source address is limited to locate at word boundary. In other words, bits 1~0 of AESSA are ignored. AESSA can be read and written. Write to AESSA during the AES accelerator is doing AES operation, the AESON is enabled, doesn't affect the current AES operation but the value of AESSA will be updated. Consequently, software can prepare the source address needed by next AES operation.

AES Destination Address Register (AESDAR)

The AES accelerator supports DMA function to transfer the cipher text between system memory and embedded FIFO. The AESDAR keeps destination address of the data buffer where the cipher text will be stored. Based on the destination address, the AES accelerator can write the cipher text back to system memory after the finish of AES operation. The value of AESDAR and AESSAR can be the same.

Register	Address	R/W	Description	Reset Value
AESDAR	AES_BASE+0x038	R/W	AES Destination Address Register	

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

Bits	Descriptions	
[31:0]	AESDA	**AES Destination Address** The AESDA keeps the destination address of the data buffer where the cipher text will be stored. The start of destination address is limited to locate at word boundary. In other words, bits 1~0 of AESDA are ignored. AESDA can be read and written. Write to AESDA during the AES accelerator is doing AES operation, the AESON is enabled, doesn't affect the current AES operation but the value of AESDA will be updated. Consequently, software can prepare the destination address needed by next AES operation.

AES Byte Count Register (AESBCR)

The AES accelerator supports the DMA function to reduce the CPU's intervention in AES operation. The AESBCR keeps the byte count of plain text that needed to do AES operation.

Register	Address	R/W	Description	Reset Value
AESBCR	AES_BASE+0x03C	R/W	AES Byte 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
BCNT							
7	6	5	4	3	2	1	0
BCNT							

Bits	Descriptions	
[31:16]	Reserved	
[15:0]	BCNT	Byte Count The BCNT keeps the byte count of plain text that needed to do AES operation. The BCNT is 16 bits and the maximum byte count is 65535 bytes. BCNT can be read and written. Write to BCNT during the AES accelerator is doing AES operation, the AESON is enabled, doesn't affect the current AES operation but the value of BCNT will be updated. Consequently, software can prepare the data byte count needed by next AES operation. Because the AES operation needs 128-bits data, the 16 bytes, it is recommended that BCNT should be divisible by 16.If BCNT is not a multiple of 16-byte, the accelerator will align the BCNT to a number of 16-byte multiple that is closest to BCNT and less than BCNT.

AES Interrupt Status Register (AESISR)

The AESISR register implements AES statuses that will trigger interrupt to CPU. It includes AES operation statuses and DMA operation statuses. AESISR is a write clear register. Write 1 to corresponding bit clears the status and also clears the interrupt.

Register	Address	R/W	Description	Reset Value
AESISR	AES_BASE+0x040	R/W	AES 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	0
Reserved					AESOK	BERR	AESINT

Bits	Descriptions	
[31:2]	Reserved	
[1]	AESOK	AES Operation OK Interrupt Status The AESOK high represents the AES operation has finished. If the AESOK is high and ENAESOK of AESIER register is enabled, the AESINT will be high. Write 1 to this bit clears the AESOK status. 1'b0: AES operation has not finished yet. 1'b1: AES operation has finished.
[0]	BERR	Bus Error Interrupt Status The BERR high represents the memory controller replies ERROR response while AES accelerator access memory through DMA during AES operation. If the BERR is high and ENBERR of AESIER register is enabled, the AESINT will be high. Write 1 to this bit clears the BERR status. 1'b0: No ERROR response is received. 1'b1: ERROR response is received.

AES Interrupt Enable Register (AESIER)

The AESIER register controls the AES interrupt generation. If any status bit of AESISR register is set and its corresponding enable bit of AESIER are enabled, the AES accelerator generates the AES interrupt to CPU. Otherwise, no AES interrupt is generated.

Register	Address	R/W	Description	Reset Value
AESIER	AES_BASE+0x044	R/W	AES 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						ENAESOK	ENBERR

Bits	Descriptions	
[31:2]	Reserved	
[1]	ENAESOK	Enable AES Operation OK Interrupt The ENAESOK controls the AESOK interrupt generation. If AESOK of AESISR register is set, and ENAESOK are enabled, the AES accelerator generates the AES interrupt to CPU. If ENAESOK is disabled, no AES interrupt is generated to CPU even the AESOK of AESISR register is set. 1′b0: AESOK of AESISR register is masked from AES interrupt generation. 1′b1: AESOK of AESISR register can participate in AES interrupt generation.
[0]	ENBERR	Enable Bus Error Interrupt The ENBERR controls the BERR interrupt generation. If BERR of AESISR register is set, and ENBERR are enabled, the AES accelerator generates the AES interrupt to CPU. If ENBERR is disabled, no AES interrupt is generated to CPU even the BERR of AESISR register is set. 1′b0: BERR of AESISR register is masked from AES interrupt generation. 1′b1: BERR of AESISR register can participate in AES interrupt generation.

AES Current Source Address Register (ACSAR)

The ACSAR indicates the source address used by AES accelerator currently to read plain text from system memory. The ACSAR is a read only register and write to it has no effect.

Register	Address	R/W	Description	Reset Value
ACSAR	AES_BASE+0x048	R	AES Current Source Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	CAESA	Current AES Source Address The CAESA keeps the source address used by AES accelerator currently for AES operation.

AES Current Destination Address Register (ACDAR)

The ACDAR indicates the destination address used by AES accelerator currently to write cipher text back to system memory. The ACDAR is a read only register and write to it has no effect.

Register	Address	R/W	Description	Reset Value
ACDAR	AES_BASE+0x04C	R	AES Current Destination Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	CAESDA	Current AES Destination Address The CAESDA keeps the destination address used by AES accelerator currently for AES operation.

AES Current Byte Count Register (ACBCR)

The ACBCR indicates the byte count of plain text that AES accelerator hasn't read yet. The ACBCR is a down counter. The ACBCR is a read only register and write to it has no effect.

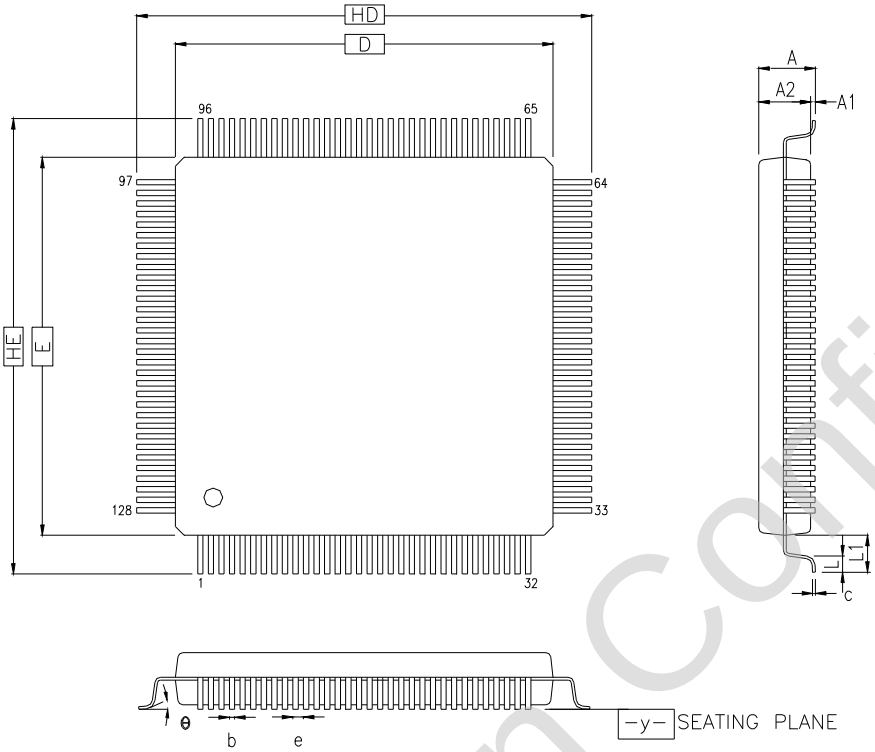
Register	Address	R/W	Description	Reset Value
ACBCR	AES_BASE+0x050	R	AES Current Byte Count Register	0x0000_0000

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
CBCNT							
7	6	5	4	3	2	1	0
CBCNT							

Bits	Descriptions	
[31:16]	Reserved	
[15:0]	CBCNT	Current Byte Count The CBCNT keeps the byte count that AES accelerator hasn't read yet for AES operation.

7 Package Specifications

7.1 128L S2LQFP (14X14X1.4mm body, 0.4mm pitch)



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
HD	16.00 BSC.			0.630 BSC.		
D	14.00 BSC.			0.551 BSC.		
HE	16.00 BSC.			0.630 BSC.		
E	14.00 BSC.			0.551 BSC.		
b	0.13	0.16	0.23	0.005	0.006	0.009
e	0.40 BSC.			0.016 BSC.		
θ	0°	3.5°	7°	0°	3.5°	7°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
y	–	–	0.1	–	–	0.004

8 Document Revision History

Version	Date	Remarks
A0	11/28/2011	Preliminary release version

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of the product could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.