Advanced Toy Controller

W55VA91

Design Guide

Ver.A4

Special Characteristics

- Powerful ARM926EJ-S core max 200MHz with fully 16/32 RISC architecture.
- High-performance 2D Graphic Engine performs high-speed 3D-like graphic function.
- GPU resolution support VGA and QVGA modes.
- Cost-effective on-chip JTAG interface debug solution.
- Built-in maximal 24KByte high-speed SRAM for graphics processing.
- Supports up to 64MB* 2 banks external SDRAM.
- 2-channel GDMA for general data transfer without CPU intervention.
- UART serial bus communication support 16-byte receive buffer.
- Built-in one independent 26-bit Timer, and 4 16-bit PWM Timer.
- GPIO support 3 ports 16-bit programmable IOs.
- Interrupt controller support programmable priority for all interrupt sources
- Vectored interrupt mode minimize interrupt latency.
- 4 Power modes: Normal, Slow, Idle, and Stop mode.
- GPU support special 2D effects, rotation & scaling, scrolling, fad-in, fad-out, a-blending.
- Support 4 Tiled or Bitmap BGs (BG0~BG3) and 1 Video_in BG4.
- Video_in support buffer mode allowable of play-on-TV interactive game.
- Sprites support 8*8~64*64 size; 16/256 color mode.
- Sprites support cell mode and bitmap mode.
- Video_out support NTSC or PAL mode.
- CRT light-pen interface for shooting game.
- Support 16-bit watch dog timer (WDT).
- 2-channel DAC, support stereo audio output.
- 2 rectangle window + sprite window function to aid graphic effect.
- Minimum 24 sprites per raster line.
- 1 TVDAC for TV encoder output.
- Support 8-channel 10-bit multiplexed ADC.
- Hardware JPEG codec engin.
- Hardware SD card and NAND-type Flash Controller.
- USB1.1 Host Controller with 2 hub ports.
- USB1.1 Device Controller supports for Full speed 12Mbps.

- Table of Contents -

1 General Description

The W55VA91 is an “edutainment” TV game controller based on the 16/32-bit ARM RISC-like, cost-effective, high-performance processor. It features a built-in TV encoder for convenient TV display, a high-quality stereo audio DAC output, a large set of peripherals, such as a USB Host/device, an 8-Channel ADC, SD/SDIO Card I/F, SPI I/F, and a complete set of system functions minimizing the number of external components. The highly integrated features can be combined with other auxiliary components to form a completed system platform that let TV game producers easily implement their cost-effective, competitive, and powerful TV game solutions.

The following integrated on-chip functions are described in details in this document.

- Embedded 20K bytes Boot ROM
- External Bus Interface Controller
- Graphic Processing Unit (GPU)
- 2-D Graphic Engine
- 1-Channel TV Encoder/DAC
- LCD controller supporting TFT/STN LCD modules
- MP3 class 16-bit 2-Channel Audio DAC
- 8-Channel 10-bit ADC
- GDMA
- SD/SDIO Card interface
- UART interface
- USB Host/ Device interface
- JPEG Codec
- Video In, CMOS interface
- SPI interface
- Timers and Watchdog Timer
- 24 Programmable I/O Ports
- Advanced Interrupt Controller
- 2-Channel PWM with capture function
- System Manager with PLL System Clock Synthesizer
- Real-time clock source
- Low-voltage Detector and Low-voltage Reset

2 Features

Architecture

- Fully 16/32-bit RISC architecture
- Little-endian mode supported
- Efficient and powerful ARM926EJ-S core
- Cost-effective JTAG-based debug solution

External Bus Interface

- 8/16/32-bit external bus support for ROM/SRAM, flash memory, SDRAM and external I/Os
- Support for SDRAM
- Programmable access cycle (0-7 wait cycle)
- Four-word depth write buffer for SDRAM write data
- Cost-effective memory-to-peripheral DMA interface

Graphic Processing Unit (GPU)

- Provide two Resolutions: QVGA (320x240) and VGA (640x480).
- Background/ Sprite provide both cell mode and bitmap mode.
- Programmable priority for sprites.
- Number of sprite per line: including rotation/scaling, worse case is 24 for 8*8 sprites.
- HV-Flip + Alpha blending + Fad-in + Fad-out + scrolling.
- Scrolling: HV scrolling, with programmable scrolling rate.
- Alpha blending: 16 levels resolution.
- Fad-in & Fad-out: 16 level resolution.
- Five Color Pallet RAMs supported: for sprites/ 4x backgrounds color pallets respectively.

2D Graphic Engine

- 6 sprites rotation and scaling per-line supported.
- 1 background rotation and scaling supported.
- Perspective projection for background.

Video-In Processor

- Direct connect to CCD or CMOS image sensor with CCIR601.
- Support the interface of TV-decoder.
- Input/ Output data format, YUV422, RGB555.

JPEG encoder/ decoder

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

Packet format:

- Support to encode interleaved YUYV format input image, output bitstream 4:2:2 and 4:2:0 format
- Support to decode interleaved YCbCr 4:4:4/4:2:2/4:2:0 format image

Support decoded output image RGB555 format (ORDER= 1).

USB Host / Device interface

- Provide 2 Host with both low speed and full speed
- Provide 1 Device with full speed

DMA Controller

- 2-channel General DMA for memory-to-memory data transfers without CPU intervention
- Initiated by software
- Increments or decrements a source or destination address in 8-bit, 16-bit or 32-bit data transfers
- 8-data burst mode

UART

- One UART (serial I/O) block with interrupt-based operation
- Support for 5-bit, 6-bit, 7-bit or 8-bit serial data transmit and receive
- Programmable baud rates
- 1, ½ or 2 stop bits
- Odd or even parity
- Break generation and detection
- Parity, overrun and framing error detection
- X16 clock mode

Timers / Watch-Dog Timer

- Four programmable 16-bit PWM timers with two 8-bit pre-scalars
- One programmable 26-bit Watchdog timer
- One-short mode or period mode operation

Programmable I/Os

- 32~ programmable I/O ports
- Pins individually configurable to input, output or I/O mode for dedicated signals
- I/O ports are configurable for Multiple functions

Advanced Interrupt Controller

- 31 interrupt sources, including 4 external interrupt sources
- Programmable normal or fast interrupt mode (IRQ, FIQ)
- Programmable as either edge-triggered or level-sensitive for 4 external interrupt sources
- Programmable as either low-active or high-active for 4 external interrupt sources
- Priority methodology is encoded to allow for interrupt daisy-chaining
- Automatically mask out the lower priority interrupt during interrupt nesting
- Automatically clear the interrupt flag when the interrupt source is programmed to be edge-triggered

Audio DAC

- Stereo 16-bit MP3 level output
- Headphone driver embedded

Display Interface

- Sync-Type TFT LCD Controller
- Sync-Type Color STN LCD Controller
- MPU-Type LCD Controller
- Embedded TV Encoder Support NTSC and PAL modes
- Support to merge YUV422 and RGB555 system.

Analog to Digital Converter and Touch Screen

- Maximum conversion rate: 250K samples per second
- 8 channel analog input
- Power supply voltage: 3.3V
- Analog input voltage range: 0 – 3.3 volts
- Touch screen semi-auto/auto conversion modes supported
- Support 4-wire and 5-wire Touch Screen
- Waiting for trigger mode supports
- Standby mode supports
- 8-level voltage detector

PLL

- Transfer the input oscillator clock (6~15MHz) into higher clock rate for system operation usage.

Real Time Clock (RTC) Source

- Independent power supply from chip to save power consumption.
- 32.768K Hz Crystal oscillation circuit
- Provide hardware alarm function

4-Channel PWM

- Four 16-bit timers
- Programmable duty control of output waveform (PWM)
- Auto reload mode or one-shot pulse mode
- Dead-zone generator
- Capture function.

Power management

- Programmable clock enables for individual peripheral
- IDLE mode to halt ARM Core and keep peripheral working
- Power-Down mode to stop all clocks including external crystal oscillator.
- Exit IDLE/Power-Down by interrupts

Operation Voltage Range

- 2.7 ~ 3.6 V for IO Buffer
- 1.62 ~ 1.98 V for Core Logic

Operation Temperature Range

- 0 ~ 70 Degree C

ESD 2KV

Operating Frequency

- CPU Core: Up to 200 MHz
- System: Up to 133 MHz

Package Type

- 216-pin LQFP

3 Block Diagram

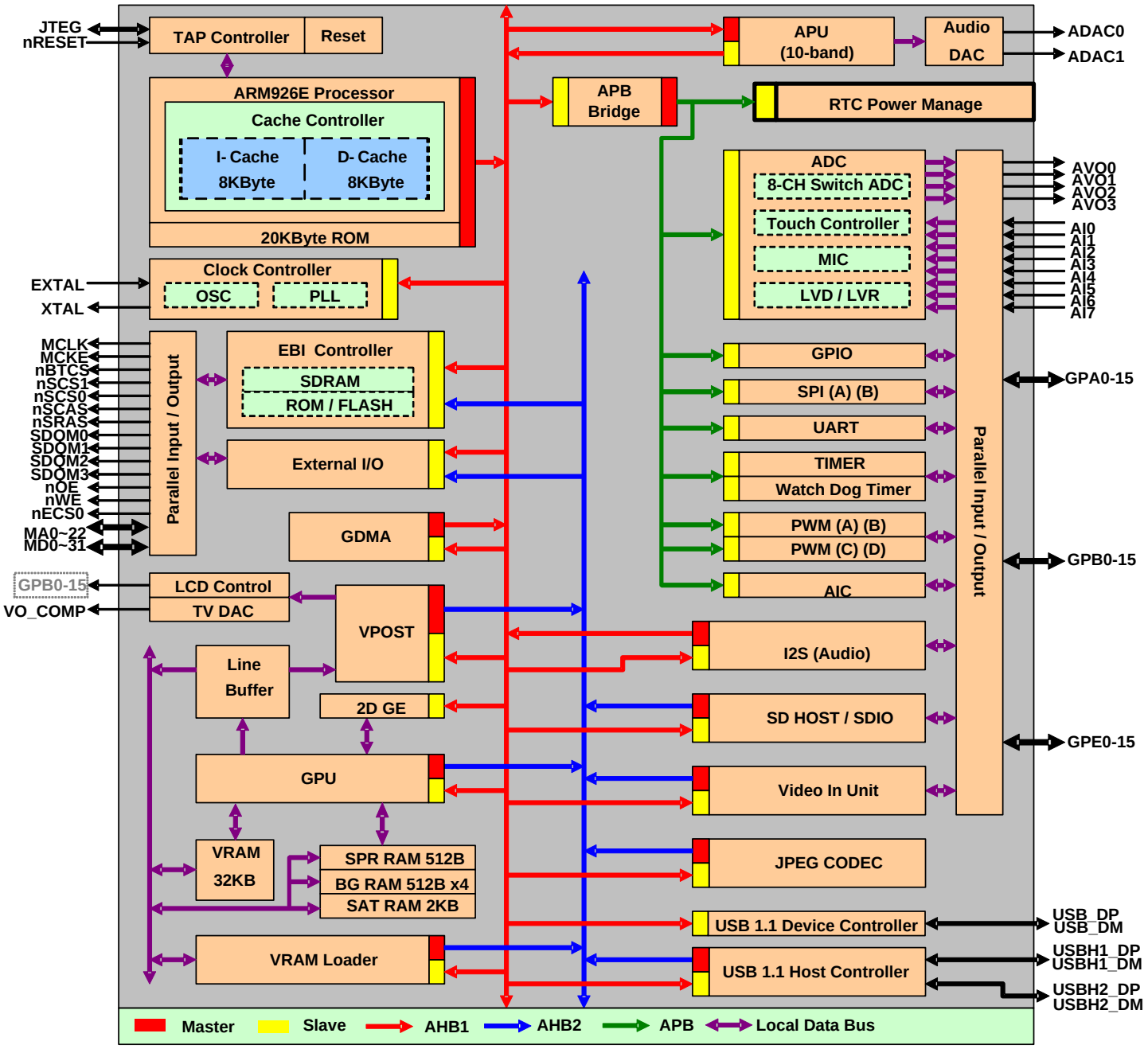


Figure 3—1 W55VA91 Block and Bus Architecture Diagram

4 Functional Description

4.1 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 trade off between high performance and high code density. The ARM926EJ-S processor has a Harvard cached architecture and provides a complete high-performance processor subsystem, including:

- An ARM926EJ-S integer core
- A Memory Management Unit (MMU)
- Separate instruction and data AMBA AHB bus interfaces

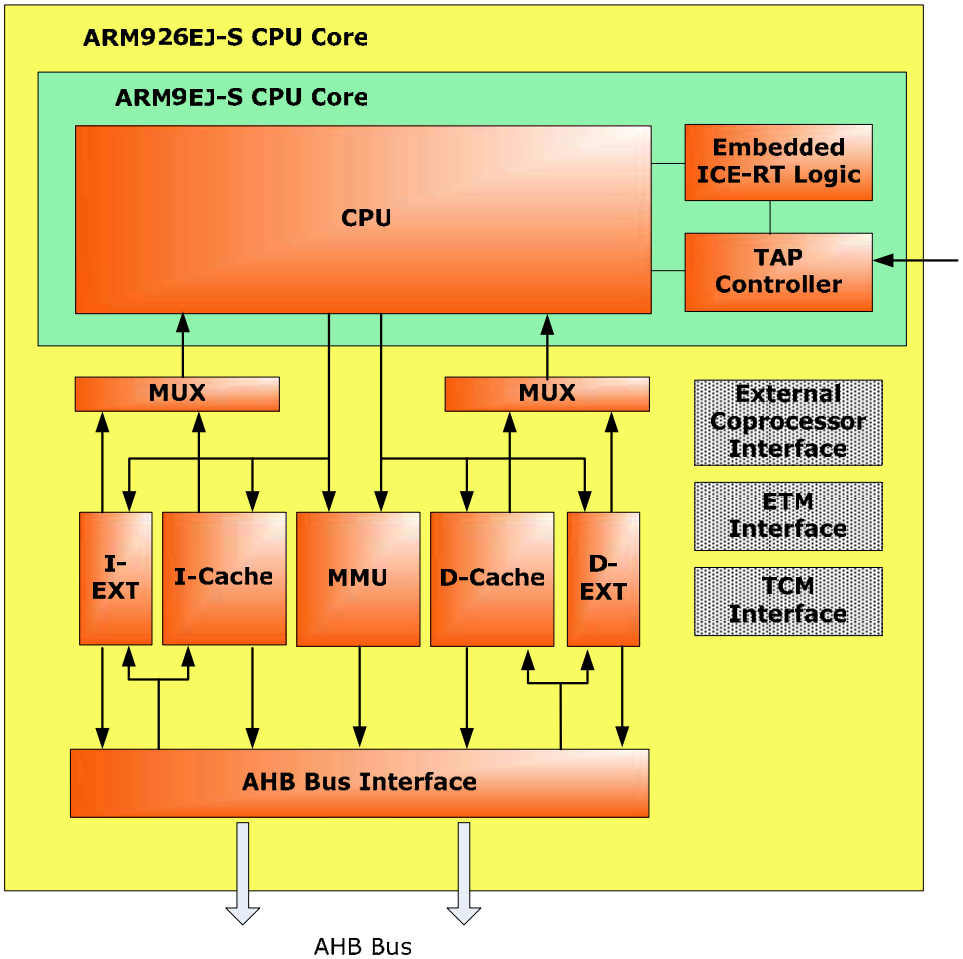


Figure 4.1-1 ARM926EJ-S CPU Core Block Diagrm

4.2 System Manager

4.2.1 Overview

The W55VA91 System Manager has the following functions.

- System memory map
- The width of external memory address
- Data bus connection with external memory
- Product identifier register
- Bus arbitration
- PLL module
- Clock divider setting for some special functions needing clocks other than system bus clock
- Clock select, enable and power saving control register
- Power-On setting reading and alternating

4.2.2 System Memory Map

W55VA91 provides 2G bytes cacheable address space and the other 2G bytes are non-cacheable. The On-Chip Peripherals bank is on 1M bytes top of the space (0xFFF0.0000 – 0xFFFF.FFFF) and the On-Chip RAM bank's start address is 0xFFE0.0000, the other banks can be located anywhere (cacheable space: 0x0~0x7FDF.FFFF if Cache ON; non-cacheable space: 0x8000.0000~0xFFDF.FFFF).

The size and location of each bank is determined by the register settings for "current bank base address pointer" and "current bank size". Please note that when setting the bank control registers, the address boundaries of consecutive banks must not overlap each other.

Except On-Chip Peripherals and On-Chip RAM, the start address of each memory bank is not fixed. You can use bank control registers to assign a specific bank start address by setting the bank's base pointer (13 bits). The address resolution is 256K bytes. The start address of a bank is defined as "base pointer << 18" and the size of this bank is "current bank size."

In the event of an access request to an address outside any programmed bank size, an abort signal is generated. The maximum accessible memory size of each external IO bank is 16M bytes, and 64M bytes on SDRAM banks.

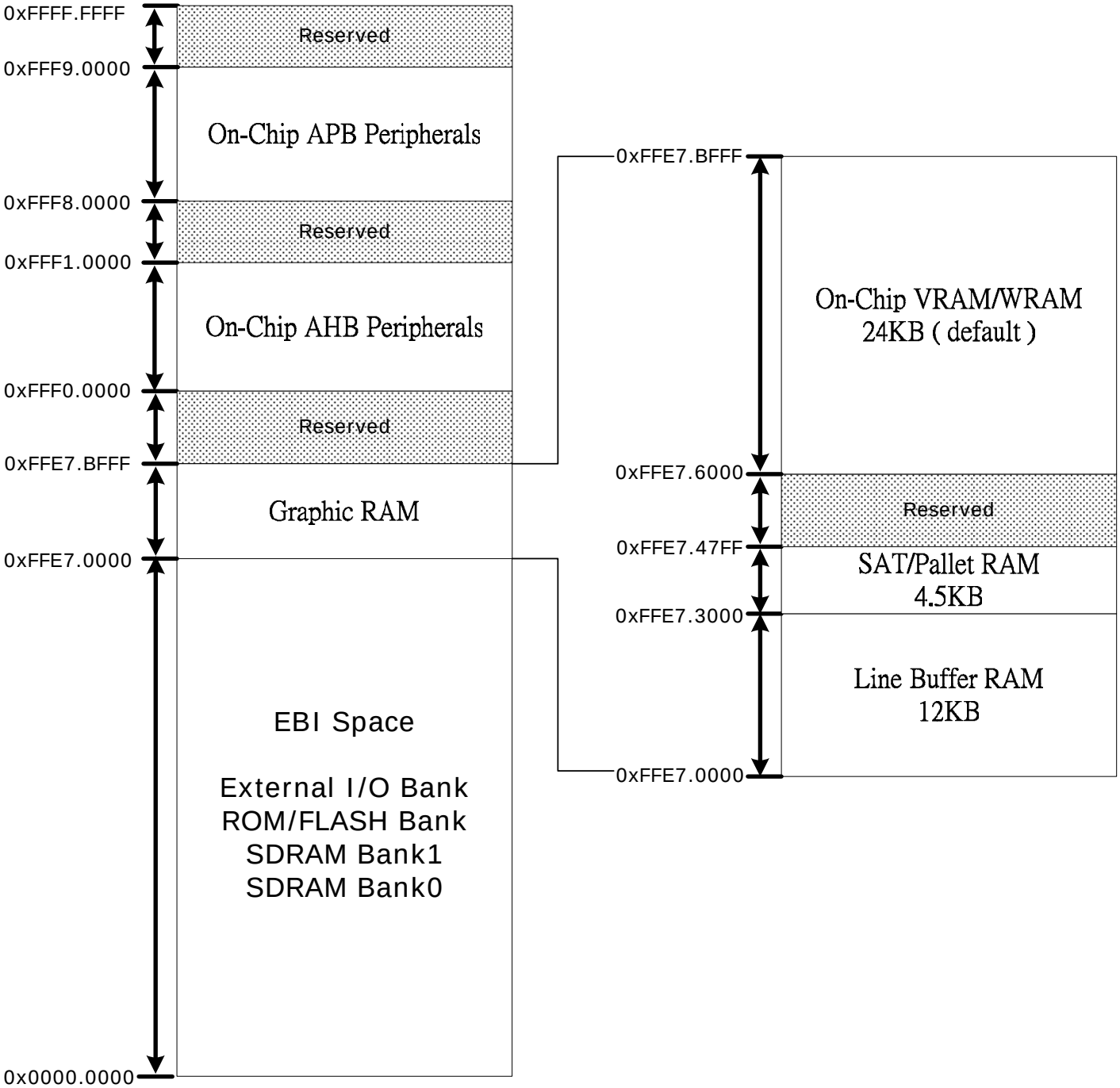


Figure 4.2-1 Figure System Memory Map

Base Address	Description
AHB Peripherals	
0xFFF0_0000	Product Identifier Register (PDID)
0xFFF0_0004	Power-On Configurations from EBI Address (PWRON)
0xFFF0_0008	Arbitration Control Register (ARBCON)
0xFFF0_000C	PLL Control Register (PLLCON)
0xFFF0_0010	Clock source Management for peripherals
0xFFF0_0014	Clock Source Select and Division setting (CLKSEL)
0xFFF0_0018	Reset Control Register (RSTCON)
0xFFF0_001C	Pins Multifunction Control Register (PINFUN)
0xFFF0_0020	Crystal on/off status Control Register (XTLCON)
0xFFF0_0024	IRQ Wakeup Control Register (IRQWAKEUPCON)
0xFFF0_0028	IRQ Wakeup Flag Regitser (IRQWAKEFLAG)
0xFFF0_002C	Power Manager Control Register (PMCON)
0xFFF0_0030	GPA/GPB I/O Cells Dual-Driving Select Control Register (GPABDS)
0xFFF0_0034	GPC/GPD I/O Cells Dual-Driving Select Control Register (GPCDDS)
0xFFF0_0038	EBI I/O Cells Triple-Driving Select Control Register(EBIDDS)
0xFFF0_003C	System clock divider (SYSDIV)
0xFFF0_0040	Internal Boot ROM Base (IBTBAS)
0xFFF0_0044	Working(or Video) RAM mapping to start address 0x0 (WPAM0)
0xFFF0_0048	Pins Multifunction 2 Control Register (PINFUN2)
0xFFF0_0050	CPU RAM BIST Control Register(RAMBIST)
0xFFF0_0054	CPU RAM BIST Control Register(RAMSTAT)
0xFFF0_1000	EBI Control Register (EBICON) Control Registers
0xFFF0_1004	ROM/FLASH Boot Memory (ROMCON) Control Registers
0xFFF0_1010	SDRAM bank 0 – 1 Control Registers
0xFFF0_1020	External I/O Control Registers
0xFFF0_3000	GDMA 0 – 1 Control Registers
0xFFF0_4000	GPU Control Registers
0xFFF0_5000	2-D Graphic Engine Control Registers
0xFFF0_6000	Display Interface (VPOST) Control Registers
0xFFF0_7000	Video-in Controller Control Registers
0xFFF0_8000	Audio IIS Controller (hidden)
0xFFF0_9000	SD Card Host Control Registers
0xFFF0_A000	Audio Processing Unit Control Registers
0xFFF0_B000	VRAM/PRAM Service System Registers
0xFFF0_C000	USB Device Registers
0xFFF0_D000	USB Host registers
0xFFF0_E000	JPEG control registers
APB Peripherals	
0xFFF8_0000	UART 0 (Tx, RX for console)
0xFFF8_1000	Timer 0, WDOG Timer
0xFFF8_2000	PWM/Timer 0 – 1
0xFFF8_3000	Advanced Interrupt Controller

0xFFF8_4000	GPIO
0xFFF8_5000	ADC Control Registers
0xFFF8_6000	SPI Interface Control Registers
0xFFF8_7000	RTC Interface Control Registers
0xFFF8_8000	SPI 1 Interface Control Registers
0xFFF8_9000	I2C Interface Control Registers

Table 4.2-1 On-Chip Peripherals Memory Map

The following table lists the base address of Working/Video RAM, Pallet RAM, SAT RAM, and Display Line Buffer RAM.

Base Address	Description
0xFFE7_0000	Display Line Buffer 0 Base (640 Words effective)
0xFFE7_1000	Display Line Buffer 1 Base (640 Words effective)
0xFFE7_2000	Display Line Buffer 2 Base (640 Words effective)
0xFFE7_3000	Background Color Pallet0 RAM Base (512 Bytes)
0xFFE7_3200	Background Color Pallet1 RAM Base (512 Bytes)
0xFFE7_3400	Background Color Pallet2 RAM Base (512 Bytes)
0xFFE7_3600	Background Color Pallet3 RAM Base (512 Bytes)
0xFFE7_3800	Sprite Color Pallet RAM Base (512 Bytes)
0xFFE7_4000	SAT RAM Base (2K Bytes)
0xFFE7_6000	Working/Video RAM Bank0 Base (8K Bytes)
0xFFE7_8000	Working/Video RAM Bank1 Base (8K Bytes)
0xFFE7_A000	Working/Video RAM Bank2 Base (8K Bytes)

Table 4.2-2 On-Chip RAM Base Address Table

4.2.3 Address Bus Generation

The W55VA91 address bus generation is depended on the required data bus width of each memory bank. The data bus width is determined by **DBWD** bits in each bank's control register.

The maximum accessible memory size of each external IO bank is 8M bytes.

Data Bus Width	External Address Pins			Maximum Accessible Memory Size
	A [19:0]	A20	A21	
8-bit	A19 – A0 (Internal)	A20 (Internal)	A21 (Internal)	4M bytes
16-bit	A20 – A1 (Internal)	A21 (Internal)	A22 (Internal)	8M bytes (half-words)
32-bit	A21 – A2 (Internal)	A22 (Internal)	A23 (Internal)	16M bytes (words)

Table 4.2-3 Address Bus Generation Guidelines

4.2.4 Data Bus Connection with External Memory

4.2.4.1 Memory Format – Little Endian

The memory format for the external memory data bus connection is little endian. For the little endian format, the lowest addressed byte in a word is considered the least significant byte of the word and the highest addressed bye is the most significant. So the byte at address 0 of the memory system connects to data lines 7 through 0.
For a word aligned address A, reference **Table 4.2-4**, shows how the word at address A, the half-word at addresses A and A+2, and the bytes at addresses A, A+1, A+2, and A+3 map on to each other.

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
Word at address A																															
Half-word at address A+2																**Half-word at address A**															
Byte at address A+3								**Byte at address A+2**								**Byte at address A+1**								**Byte at address A**							

Table 4.2-4 Little-endian addresses of bytes and half-words within words

4.2.4.2 Connection of External Memory with Various Data Width

The system diagram for W55VA91 connecting with the external memory is shown in

Base Address	**Description**
0xFFE7_0000	Display Line Buffer 0 Base (640 Words effective)
0xFFE7_1000	Display Line Buffer 1 Base (640 Words effective)
0xFFE7_2000	Display Line Buffer 2 Base (640 Words effective)
0xFFE7_3000	Background Color Pallet0 RAM Base (512 Bytes)
0xFFE7_3200	Background Color Pallet1 RAM Base (512 Bytes)
0xFFE7_3400	Background Color Pallet2 RAM Base (512 Bytes)
0xFFE7_3600	Background Color Pallet3 RAM Base (512 Bytes)
0xFFE7_3800	Sprite Color Pallet RAM Base (512 Bytes)
0xFFE7_4000	SAT RAM Base (2K Bytes)
0xFFE7_6000	Working/Video RAM Bank0 Base (8K Bytes)
0xFFE7_8000	Working/Video RAM Bank1 Base (8K Bytes)
0xFFE7_A000	Working/Video RAM Bank2 Base (8K Bytes)

Table 4.2-2 and **Figure 4.2-3**. Below tables (through **Table 4.2-5** and **Table 4.2-6**) show the program/data path between CPU register and the external memory using little endian and word/half-word/byte access.

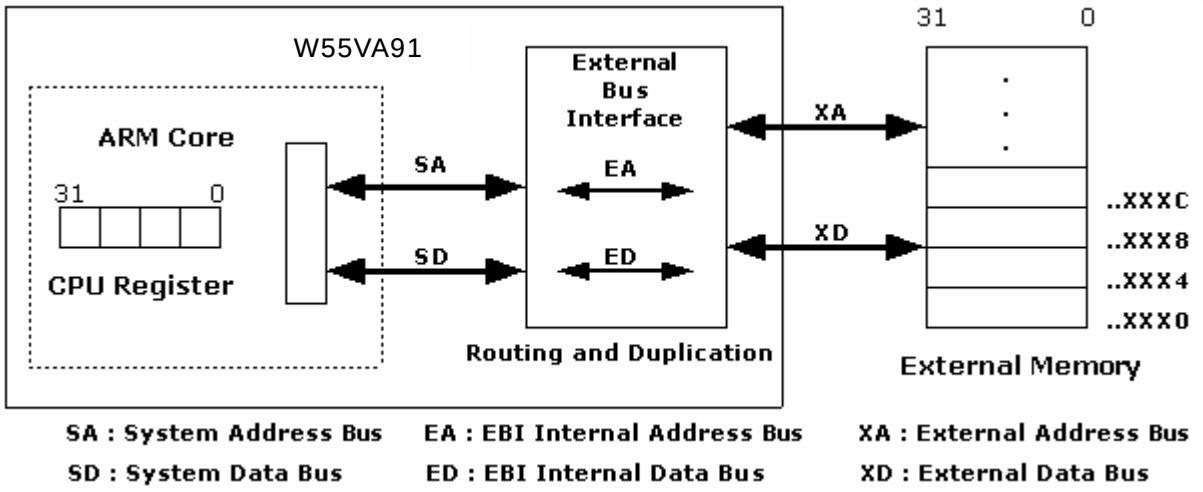


Figure 4.2-2 Address/Data bus connection with external memory

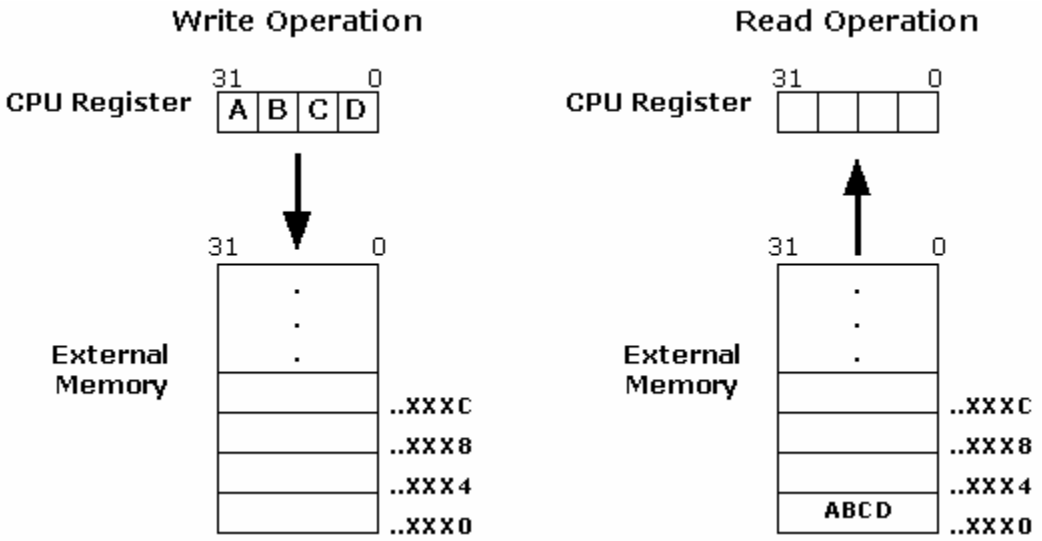


Figure 4.2-3 CPU register Read/Write with external memory

Using little-endian and word access, Program/Data path between register and external memory
WA = Address whose LSB is 0,4,8,C X = Don't care
nWBE [3-0] / SDQM [3-0] = A means active and U means inactive

Access Operation	**Write Operation (CPU Register → External Memory)**		
XD Width	**Word**	**Half Word**	**Byte**
Bit Number **CPU Reg Data**	31 0 ABCD	31 0 ABCD	31 0 ABCD
SA	WA	WA	WA
Bit Number **SD**	31 0 ABCD	31 0 AB CD	31 0 A B C D

Bit Number ED	31 0 ABCD	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
XA	WA	WA	WA+2	WA	WA+1	WA+2	WA+3
nWBE [3-0] / SDQM [3-0]	AAAA	XXAA	XXAA	XXXA	XXXA	XXXA	XXXA
Bit Number XD	31 0 ABCD	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Bit Number Ext. Mem Data	31 0 ABCD	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Timing Sequence		1st write	2nd write	1st write	2nd write	3rd write	4th write

Table 4.2-5 Word access write operation with Little Endian

Access Operation	Read Operation (CPU Register ← External Memory)						
XD Width	Word	Half Word		Byte			
Bit Number CPU Reg Data	31 0 ABCD	31 0 ABCD		31 0 ABCD			
SA	WA	WA		WA			
Bit Number SD	31 0 ABCD	31 0 AB CD		31 0 A B C D			
Bit Number ED	31 0 ABCD	31 0 XX CD	31 0 AB CD	31 0 X X X D	31 0 X X C D	31 0 X B C D	31 0 A B C D
XA	WA	WA	WA+2	WA	WA+1	WA+2	WA+3
SDQM [3-0]	AAAA	XXAA	XXAA	XXXA	XXXA	XXXA	XXXA
Bit Number XD	31 0 ABCD	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Bit Number Ext. Mem Data	31 0 ABCD	15 0 CD	15 0 AB	7 0 D	7 0 C	7 0 B	7 0 A
Timing Sequence		1st read	2nd read	1st read	2nd read	3rd read	4th read

Table 4.2-6 Word access read operation with Little Endian

Using little-endian and half-word access, Program/Data path between register and external memory.
HA = Address whose LSB is 0,2,4,6,8,A,C,E HAL = Address whose LSB is 0,4,8,C
HAU = Address whose LSB is 2,6,A,E X = Don't care
nWBE [3-0] / SDQM [3-0] = A means active and U means inactive

Access Operation	Write Operation (CPU Register → External Memory)			
XD Width	Word		Half Word	Byte
Bit Number CPU Reg Data	31 0 ABCD		31 0 ABCD	31 0 ABCD
SA	HAL	HAU	HA	HA

Bit Number SD	31 0 CD CD	31 0 CD CD	31 0 CD CD	31 0 CD CD	31 0 CD CD
Bit Number ED	31 0 CD CD	31 0 CD CD	31 0 CD CD	7 0 D	7 0 C
XA	HAL	HAL	HA	HA	HA+1
nWBE [3-0] / SDQM [3-0]	UUAA	AAUU	XXAA	XXXA	XXXA
Bit Number XD	31 0 CD CD	31 0 CD CD	15 0 CD	7 0 D	7 0 C
Bit Number Ext. Mem Data	15 0 CD	31 16 CD	15 0 CD	7 0 D	7 0 C
Timing Sequence				1st write	2nd write

Table 4.2-7 Half-word access write operation with Little Endian

Access Operation	Read Operation (CPU Register ← External Memory)				
XD Width	Word		Half Word	Byte	
Bit Number CPU Reg Data	15 0 CD	15 0 AB	15 0 CD	15 0 CD	
SA	HAL	HAU	HA	HA	
Bit Number SD	15 0 CD	15 0 AB	15 0 CD	15 0 CD	
Bit Number ED	15 0 CD	15 0 AB	15 0 CD	15 0 XD	15 0 CD
XA	HAL	HAL	HA	HA	HA+1
SDQM [3-0]	UUAA	AAUU	XXAA	XXXA	XXXA
Bit Number XD	31 0 AB CD	31 0 AB CD	15 0 CD	7 0 D	7 0 C
Bit Number Ext. Mem Data	31 0 ABCD		15 0 CD	7 0 D	7 0 C
Timing Sequence				1st read	2nd read

Table 4.2-8 Half-word access read operation with Little Endian

Using little-endian and byte access, Program/Data path between register and external memory.
BA = Address whose LSB is 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
BAL = Address whose LSB is 0,2,4,6,8,A,C,E BAU = Address whose LSB is 1,3,5,7,9,B,D,F
BA0 = Address whose LSB is 0,4,8,C BA1 = Address whose LSB is 1,5,9,D
BA2 = Address whose LSB is 2,6,A,E BA3 = Address whose LSB is 3,7,B,F

Access Operation	Write Operation (CPU Register → External Memory)						
XD Width	Word				Half Word		Byte
Bit Number CPU Reg Data	31 0 ABCD				31 0 ABCD		31 0 ABCD
SA	BA0	BA1	BA2	BA3	BAL	BAU	BA

Bit Number	31 0	31 0	31 0	31 0	31 0	31 0	31 0
SD	D D D D	D D D D	D D D D	D D D D	D D D D	D D D D	D D D D
Bit Number	7 0	15 8	23 16	31 24	7 0	15 8	7 0
ED	D	D	D	D	D	D	D
XA	BA0	BA0	BA0	BA0	BAL	BAL	BA
nWBE [3-0] / SDQM [3-0]	UUUA	UUAU	UAUU	AUUU	XXUA	XXAU	XXXA
Bit Number	31 0	31 0	31 0	31 0	15 0	15 0	7 0
XD	X X X D	X X D X	X D X X	D X X X	X D	D X	D
Bit Number	7 0	15 8	23 16	31 24	7 0	15 8	7 0
Ext. Mem Data	D	D	D	D	D	D	D
Timing Sequence							

Table 4.2-9 Byte access write operation with Little Endian

Access Operation	Read Operation (CPU Register ← External Memory)						
XD Width	Word				Half Word		Byte
Bit Number	7 0	7 0	7 0	7 0	7 0	7 0	7 0
CPU Reg Data	D	C	B	A	D	C	D
SA	BA0	BA1	BA2	BA3	BAL	BAU	BA
Bit Number	7 0	7 0	7 0	7 0	7 0	7 0	7 0
SD	D	C	B	A	D	C	D
Bit Number	7 0	7 0	7 0	7 0	7 0	7 0	7 0
ED	D	C	B	A	D	C	D
XA	BA0	BA0	BA0	BA0	BAL	BAL	BA
SDQM [3-0]	UUUA	UUAU	UAUU	AUUU	XXUA	XXAU	XXXA
Bit Number	31 0	31 0	31 0	31 0	15 0	15 0	7 0
XD	ABCD	ABCD	ABCD	ABCD	CD	CD	D
Bit Number	31 0				15 0		7 0
Ext. Mem Data	ABCD				CD		D
Timing Sequence							

Table 4.2-10 Byte access read operation with Little Endian

4.2.5 Bus Arbitration

The W55VA91's internal function blocks or external devices can request mastership of the system bus and then hold the system bus in order to perform data transfers. Because the design of W55VA91 bus allows only one bus master at a time, a bus controller is required to arbitrate when two or more internal units simultaneously request bus mastership. When bus mastership is granted to an internal function block, other pending requests are not acknowledged until the previous bus master has released the bus.
W55VA91 supports two priority modes, the **Fixed Priority Mode** and the **Rotate Priority Mode**, depends on the **PRTMOD** bit setting.

4.2.5.1 Fixed Priority Mode

In **Fixed Priority Mode** (**PRTMOD**=0, default value), to facilitate bus arbitration, priorities are assigned to each internal W55VA91 function block. The bus controller arbitration requests for the bus mastership according to these fixed priorities. In the event of contention, mastership is granted to the function block with the highest assigned priority. These priorities are listed in **Table 4.2-11**.

Bus Priority	Function Block
1 (Highest)	Video IN
2	VPOST
3	JPEG
4	GPU
5	2-D GE (debug)
6	VRAM Loader
7	USB Host
8	SD Card Host
9	Audio DAC DMA
10	General DMA0
11	General DMA1
12(Lowest)	ARM Core

Table 4.2-11 Bus Priorities for Arbitration in Fixed Priority Mode

4.2.5.2 Rotate Priority Mode

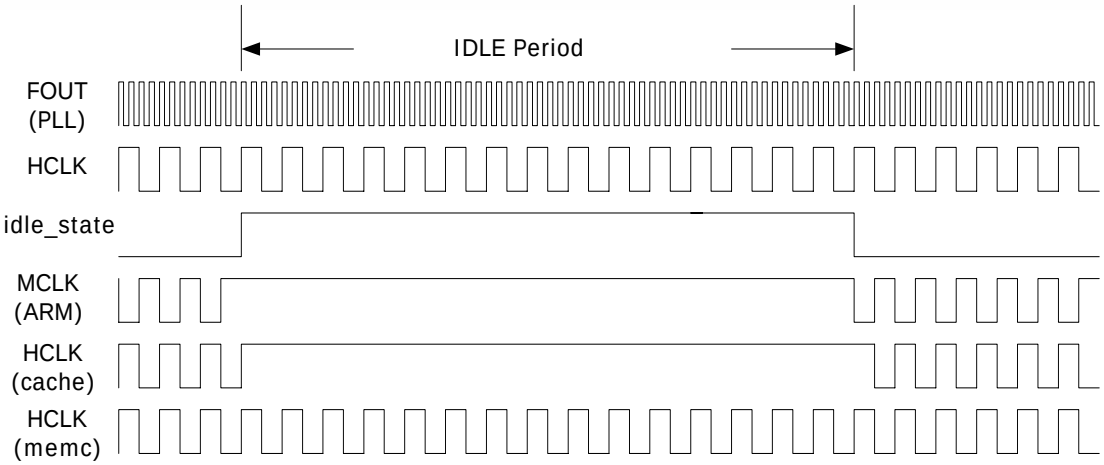
In **Rotate Priority Mode** (**PRTMOD**=1), W55VA91 uses a round robin arbitration scheme ensures that all bus masters have equal chance to gain the bus and that a retracted master does not lock up the bus.

4.2.6 Power Management

W55VA91 provide three power management scenarios to reduce power consumption. The peripherals' clocks can be enabled or disabled individually by controlling the corresponding bit in CLKSEL control register. Software can turn-off the unused modules' clocks to saving the unnecessary power consumption.

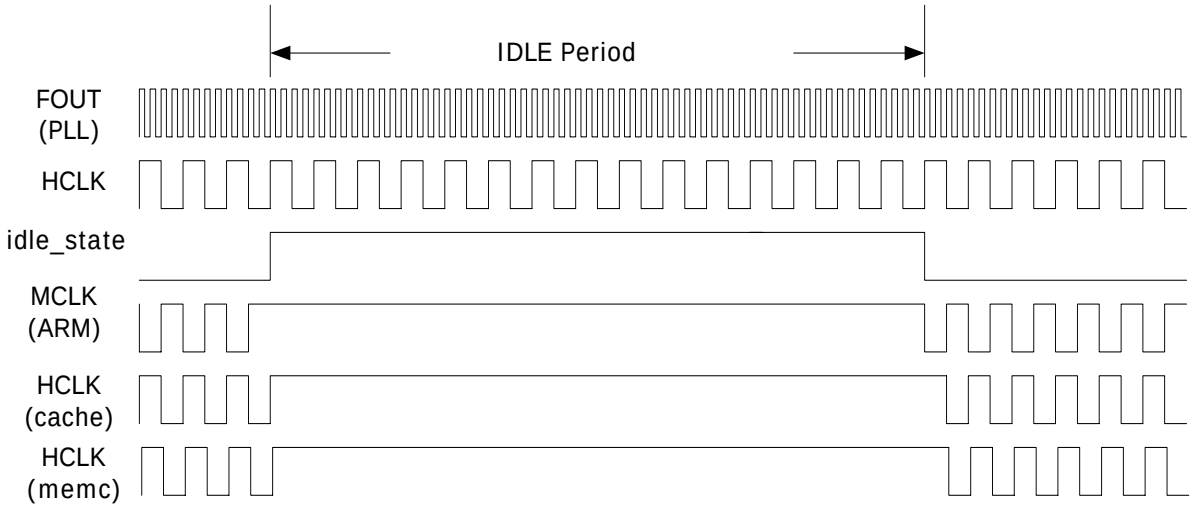
IDLE Mode

If the IDLE bit in Power Management Control Register (PMCON) is set, the ARM CORE clock source will be halted, the ARM CORE will not go forward. The AHB or APB clocks still active except the clock to ARM9 core are stopped. W55VA91 will exit idle state when nIRQ or nFIQ from any peripheral is received; like keypad, timer overflow interrupts and so on. The memory controller can also be forced to enter idle state if both MIDLE and IDLE bits are set. Software must switch SDRAM into self-refresh mode before force memory enter idle mode.



Case1. IDLE=1, PD=0, MIDLE=0

Figure 4.2-4 Power Management – Clock Control, Case 1



Case2. IDLE=1, PD=0, MIDLE=1

Figure 4.2-5 Power Management – Clock Control, Case 2

Power-Down Mode

The mode provides the minimum power consumption. When the W55VA91 system is not working or waiting an external event, software can write PD bit "1" to turn off all the clocks includes system crystal oscillator to let ARM CORE enter sleep mode. In this state, all peripherals are also in sleep mode since the clock source is stopped. W55VA91 will exit power down state when nIRQ/nFIQ is detected. W55VA91 provides external interrupt nIRQ[3:0], and Low-Voltage-Detector to wakeup the system clock. In addition, a Low-Voltage-Reset will also wakeup the system clock and activate a reset condition.

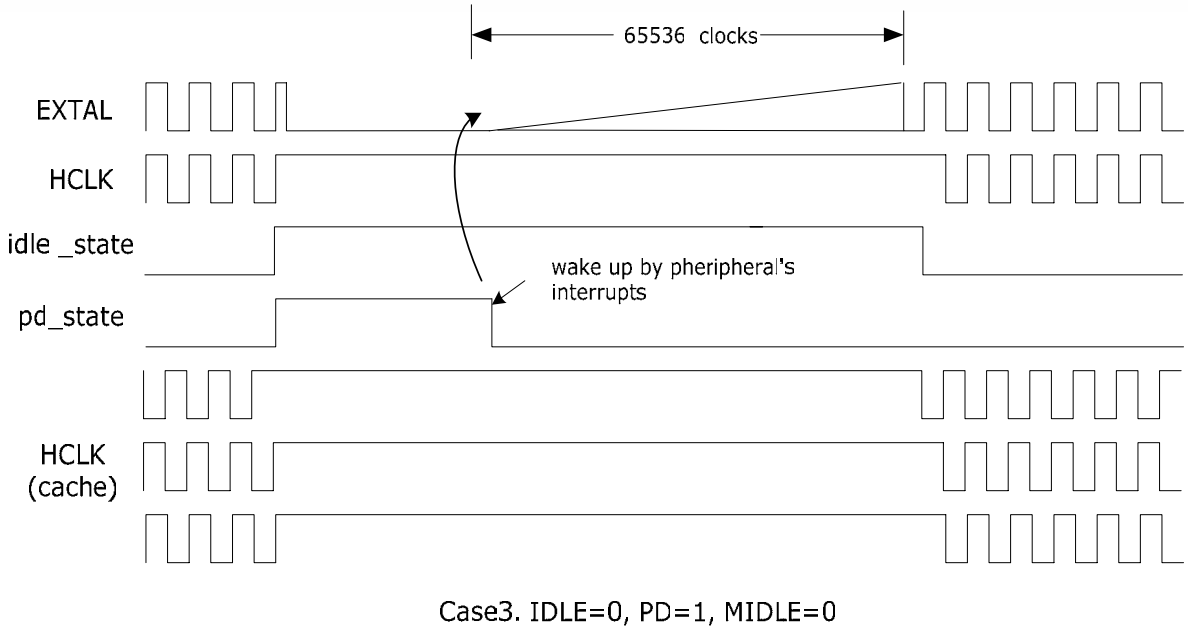


Figure 4.2-6 Power Management – Clock Control, Case 3

4.2.7 Power-On Setting

After power on reset, there are four Power-On setting pins to configure W55VA91 system configuration.

Power-On Setting	Pin
Internal PLL-generated System Clock Select	MA18
Internal Boot ROM Select	MA16
Boot ROM/FLASH Data Bus Width	MA [15:14]

Table 4.2-12 Power-On Setting and Description

MA18 pin：Internal PLL-generated System Clock Select
If pin MA18 is pull-down, the external clock from EXTAL pin is served as internal system clock.
If pin MA18 is pull-up, the PLL output clock is used as internal system clock.

MA16 pin： Internal Boot ROM Select
If pin MA16 is pull-down, the external ROM/Flash bank connected on EBI bus is served as Booting-up program source rather than the program code contained in the internal Boot ROM.
If pin MA16 is pull-up, the Booting-up program code contained in the internal Boot ROM is the one boots and setups the whole system.

MA [15:14] ： Boot ROM/FLASH Data Bus Width

MA[15:14]		Bus Width
Pull-down	Pull-down	8-bit
Pull-down	Pull-up	16-bit
Pull-up	Pull-down	32-bit
Pull-up	Pull-up	RESERVED

MA[13:8]: Power On Setting Value for Software Reference

These setting values are for software in boot ROM, especially for internal boot ROM, to detect as a reference for its own configuration usages.

NOTE: Power on setting values are transparent to W55VA91 through these address pins when pin "nRESET" is LOW. And they are latched and fixed after "nRESET" going high. After "nRESET" going high, the address pins will function as memory address pins. Following plot illustrates the power-on option latching process.

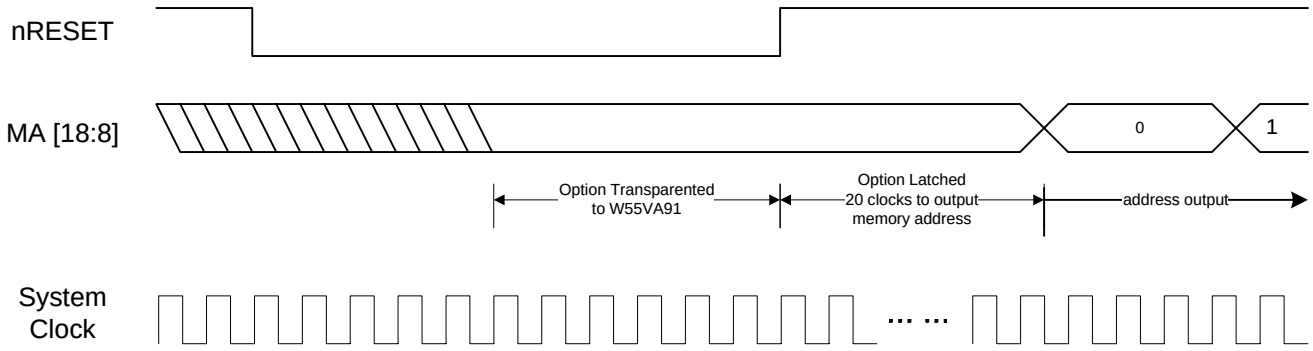


Figure 4.2-7 Power-on option latching

MA[21:19] : Test Mode Configuration at Power On

Power-on setting, from A[21:19] address pins of EBI(External Bus Interface), configures the W55VA91 pins to three test modes, **(1) CPU Core test (2) Internal SRAM test (3) DFT test mode.** The test mode enable pin "TEST" should also be always set HIGH.

TEST	MCKE	MA[21]	MA[20:19]	Configuration Mode	Description
1	1	x	xx	DFT test mode	Reserved for Design Testability
1	0	0	00	CPU Core Test	ARM9 test chip
1	0	0	01	Internal SRAM Test	Internal SRAM memory test
1	0	1	00	Analog Test Mode 1	Test Audio DAC and TV DAC
1	0	1	01	Analog Test Mode 2	Test AD Converter, Touch Panel and LVD / LVR
0	x	x	xx	Normal Run Mode	W55VA91 run in normal operating

Note:
1. All Nuvoton test mode is based on TEST = 1; TEST = 0 is normal run mode.

4.2.8 Internal Boot ROM Flow Chart

The Boot program combine different programs allowing download and/or upload into the internal memories of the chip. It implements the algorithm shown in Figure 4.2.8.

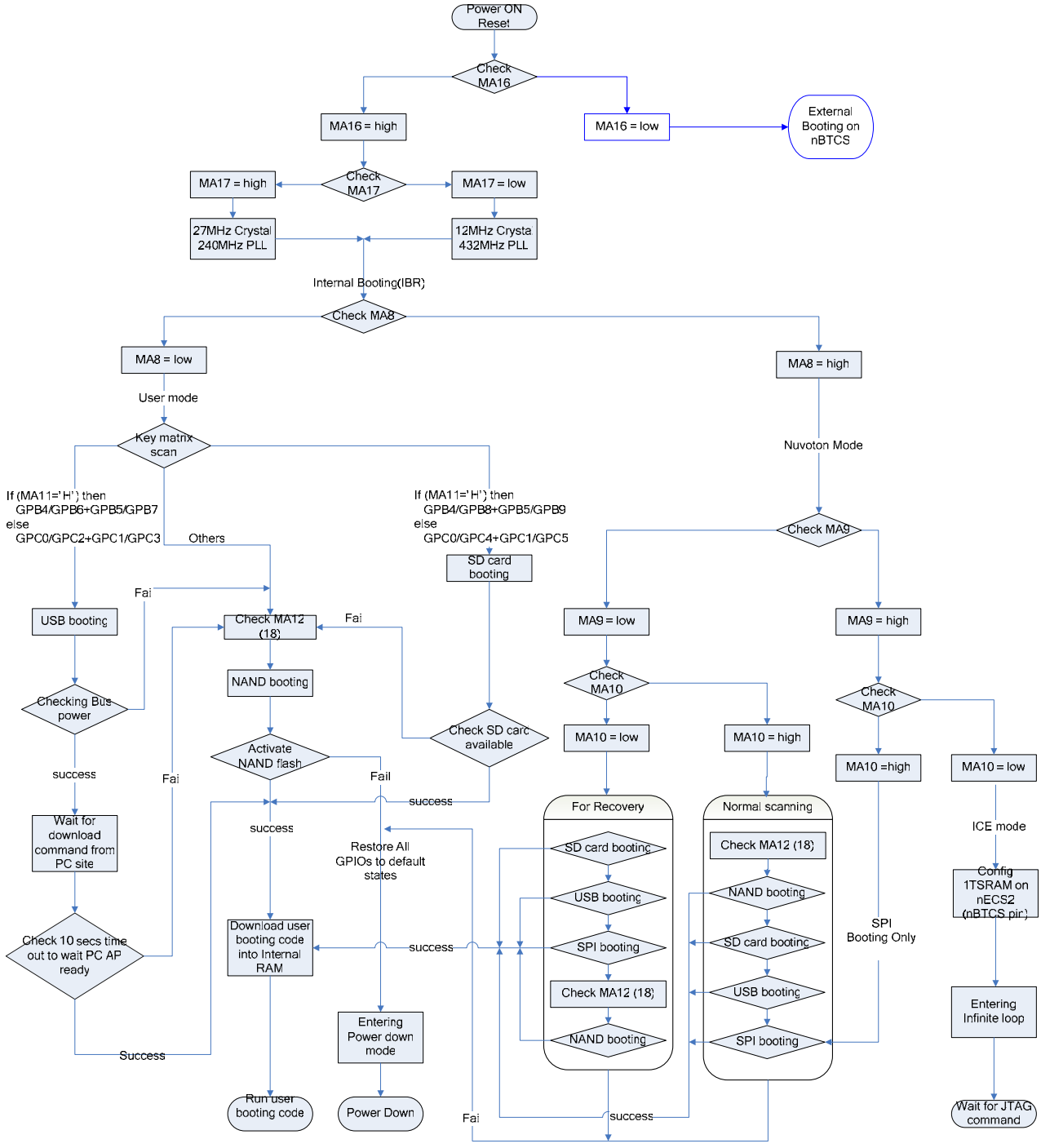


Figure 4.2-8 Internal Booting flow chart

There are some matters needing attentions:

1. If use SD or NAND booting, the system need a SDRAM to act as the bridge.
2. Can support SPI flash.
3. For Standard NAND Flash, load 8K(16 pages) on block 0 into internal RAM
4. For Advanced NAND Flash, load 16K(8 pages) on block 0 into internal RAM
5. User booting code on internal starting address is fixed to 0xFFE76000
6. Whenever booting fail on any devices, TV will show hint messages then boot next devices
7. SD card booting will occupy GPA1~GPA6 IO pins

8. When entering infinite loop, the ICE still can connect and download program
9. NAND booting also supports OTP (One-Time-Programmable) NAND type flash.
10. For SD card booting, IBR also load 16K bytes user booting code into internal RAM
11. For Nuvoton mode NAND flash recovery, MA9 = low and MA10 = low will entering recovery mode
12. Internal Booting ROM will be remapped to 0x60000000
13. If MA11=High ,It will be set GPB4 and GPB5 as input pin and GPB6, GPB7, GPB8, GPB9 as output pin (User Mode) ,If MA11=Low ,It will be set GPC0 and GPC1 as input pin and GPC2, GPC3, GPC4, GPC5 as output pin .
14. Once VA91 connected to PC, if AP on PC site is not activated or connected in 10secs with user mode and 20secs with Nuvoton mode. The time out occurs and switches booting path to NAND booting.
15. Before entering to power down, all the IPs and GPIO will be reset to their default states.
16. GPD0 is used for SDC power control. If booting success from SDC, it will be keep to GPIO function on address bus. But booting fail form SDC, it will be switch to address bus function.
17. 16 bit 1T-SRAM module must be connected to external IO bank 2 on nBTCS pin.
18. NAND flash must be connected with virtual NAND interface. The GPA11 is used for ready/busy status pin, GPA12 for Output enable, GPA13 for write enable, GPA10 for CLE pin and GPA9 for ALE pin.
19. If MA12 setting to “H”, It will be NAND booting from MD24 ~ MD31 data bus. But MA12 setting to “L”, It will be NAND booting from GPB3 ~ GPB10 data bus, If choose it function, User mode can not use GPB key matrix mode, only use GPC key matrix mode.

4.2.9 System Manager Control Registers

Register	Address	R/W	Description	Reset Value
SYS_BA = 0xFFF0_0000				
PDID	SYS_BA+0x000	R	Product Identifier Register	0xXX55_0091
PWRON	SYS_BA+0x004	R/W	Power-On Configurations from EBI Address	0x0000_0XXX
ARBCON	SYS_BA+0x008	R/W	Arbitration Control Register	0x0000_0000
PLLCON	SYS_BA+0x00C	R/W	PLL Control Register	0x0000_4F27
CLKMAN	SYS_BA+0x010	R/W	Clock source Management for peripherals	0x0016_EC48
CLKSEL	SYS_BA+0x014	R/W	Clock Source Select and Division setting Register	0x0011_45FF
RSTCON	SYS_BA+0x018	W	Reset Control Register	0x0000_0000
PINFUN	SYS_BA+0x01C	R/W	Pins Multifunction Control Register	0x0000_0000
XTLCON	SYS_BA+0x020	R/W	Crystal on/off status Control Register	0x0000_0009
IRQWAKECON	SYS_BA+0x024	R/W	IRQ Wakeup Control register	0x0000_0000
IRQWAKEFLAG	SYS_BA+0x028	R/W	IRQ wakeup Flag Register	0x0000_0000
PMCON	SYS_BA+0x02C	R/W	Power Manager Control Register	0x0000_0000
GPABDS	SYS_BA+0x030	R/W	GPA/B dual driving output selection	0x0000_0000
GPCEDS	SYS_BA+0x034	R/W	GPC/E dual driving output selection	0x0000_0000

EBIDS	SYS_BA+0x038	R/W	EBI triple driving output selection	0x0000_00AA
SYSDIV	SYS_BA+0x03C	R/W	System clock divider Register	0x0000_1111
IBTBAS	SYS_BA+0x040	R/W	Internal Boot ROM Base	0x0000_0000
WMAP0	SYS_BA+0x044	R/W	Working (or Video) RAM mapping to start address 0x0	0x0000_0000
PINFUN2	SYS_BA+0x048	R/W	Pins Multifunction 2 Control Register	0x0000_0000
RAMBIST	SYS_BA+0x050	R/W	CPU RAM BIST Control Register	0x0000_0000
RAMSTAT	SYS_BA+0x054	R	CPU RAM BIST Status Register	0x0000_0000

Product Identifier Register (PDID)

This register is for only read and enables software to recognize certain characteristics of the chip ID and the version number.

Register	Address	R/W	Description	Reset Value
PDID	SYS_BA+0x000	R	Product Identifier Register	0xXX55_0091

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

Bits	Descriptions		Default
[29:24]	**VERSION**	**Version of the chip.**	--
[23:0]	**CHPID**	**Chip identifier** The chip identifier of W55VA91 is 0x550091.	0x550091

Power-On Configuration from EBI Address (PWRON)

Register	Address	R/W	Description	Reset Value
PWRON	SYS_BA+0x004	R/W	Power-On Configuration and Alternate Setting Register	0x0000_0XXX

31	30	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					IPLLS	RESERVED	IBTS

7	6	5	4	3	2	1	0
EBTSIZE		PWR_SWREF					

Bits	Descriptions	
[10]	**IPLLS**	**Internal PLL-generated clock Selection** 0 = External Crystal oscillator clock is the main system clock source 1 = Internal PLL-generated clock is selected as the main system clock source **NOTE:** This bit has a initial setting at power-on from EBI MA[18] and can be alternated by software.
[8]	**IBTS**	**Internal Boot ROM Selected** 0 = External Boot ROM is selected as boot ROM device. 1 = Internal Boot ROM is selected as boot ROM device. **NOTE:** This bit has a initial setting at power-on from EBI MA[16] and can be alternated by software.
[7:6]	**EBTSIZE**	**External Boot ROM BIT width Size** 00 = External Boot ROM is 8-bit width 01 = External Boot ROM is 16-bit width 10 = External Boot ROM is 32-bit width 11 = Reserved **NOTE:** These bit have a initial setting at power-on from EBI MA[15:14] and can be alternated by software.
[5:0]	**PWR_SWREF**	**Power-On values for SoftWare Reference** The value of these bits are from EBI MA[13:8] at power-on. Any value in these bits would not affect any hardware behavior but software can access it to be a decision reference. All or some of these bits can be “don’t care” values if software doesn’t use them as references.

Arbitration Control Register (ARBCON)

Register	Address	R/W	Description	Reset Value
ARBCON	SYS_BA+0x008	R/W	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	PRTMOD1	PRTMOD0

Bits	Descriptions		Default
[3]	**IPACT**	**Interrupt priority active (Read only)** When IPEN=”1”, this bit is set when the ARM core has an unmasked interrupt request.	0x0

[2]	**IPEN**	**IPEN [2] ： Interrupt priority enable bit** 0 = the ARM core has the lowest priority. 1 = enable to raise the ARM core priority to second This bit is available only when the PRTMOD0=0.	0x0
[1]	**PRTMOD1**	**PRTMOD1： AHB2 Master Priority mode select** 0 = Fixed Priority Mode 1 = Rotate Priority Mode	0x0
[0]	**PRTMOD0**	**AHB1 Master Priority mode select** 0 = Fixed Priority Mode 1 = Rotate Priority Mode	0x0

PLL Control Register (PLLCON)

Register	Address	R/W	Description	Reset Value
PLLCON	SYS_BA+0x00C	R/W	PLL Control Register	0x0000_4F27

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

Bits	Descriptions		Default
[16]	**PWDEN**	**Power down mode enable** 0 = PLL is in normal mode 1 = PLL is in power down mode **NOTE:** If power-on setting of MA18 (IPLLS) is ‘0’. The default of this bit will be ‘1’, PLL is power-down. Otherwise, this bit is ‘0’.	0x0 (IPLLS = 1) 0x1 (IPLLS = 0)
[15:7]	**FBDV**	**PLL VCO output clock feedback divider** Feedback Divider divides the output clock from VCO of PLL.	0x9E
[6:5]	**OTDV**	**PLL output clock divider** OTDV [6:5] / Divided by 0 0 : 1 0 1 : 2 1 0 : 2 1 1 : 4	0x1
[4:0]	**INDV**	**PLL input clock divider** Input Divider divides the input reference clock into the PLL.	0x7

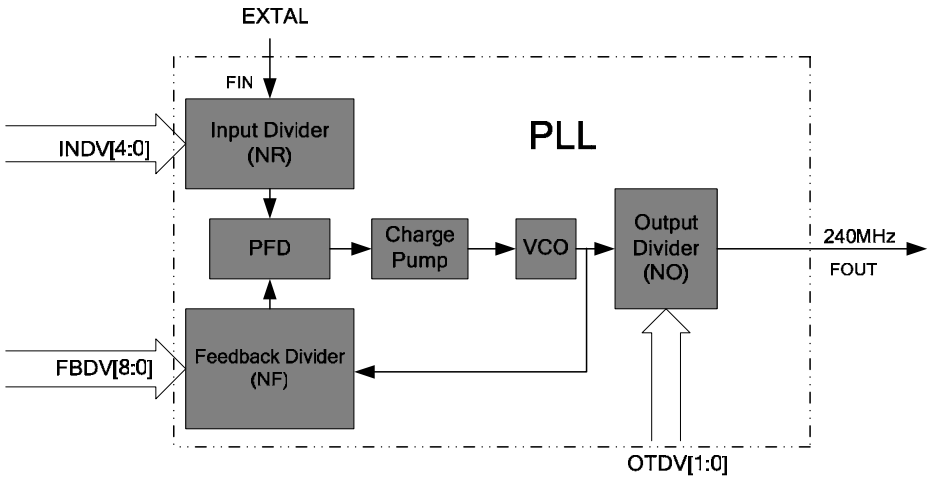


Figure 4.2-9 System PLL Clock Diagram

The formula of output clock of PLL is:

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

FOUT：Output clock of Output Divider
FIN：External clock into the Input Divider
NR：Input divider value (NR = INDV + 2)
NF：Feedback divider value (NF = FBDV + 2)
NO：Output divider value (NO = OTDV)

Clock Source Management for Peripherals (CLKMAN)

Register	Address	R/W	Description	Reset Value
CLKMAN	SYS_BA+0x010	R/W	Clock source enable/disable management for peripherals	0x0016_EC48

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED			**I2CCK_EN**	**UHC_EN**	**USB48_EN**	**USBCK_EN**	**RESERVED**
15	14	13	12	11	10	9	8
WDT_EN	**TMR_EN**	**PWM_EN**	**SPI_EN**	**UART_EN**	**ADC_EN**	**GDMA_EN**	**GPU_EN**
7	6	5	4	3	2	1	0
GE_EN	**VPST_EN**	**VID_EN**	**JPEG_EN**	**SDH_EN**	**AUD_EN**	**RESERVED**	

Bits	Descriptions		Default
[20]	**I2CCK_EN**	**I2C operation and driving clock enable bit** 0 = I2C operation is disabled. 1 = I2C operation is enabled.	0x1

[19]	**UHC_EN**	**USB Host Controller clock enable bit** 0 = USB Host Controller operation is disabled. 1 = USB Host Controller operation is enabled.	0x0
[18]	**USB48_EN**	**USB driving clock for 48MHz enable bit** 0 = USB 48MHz driving clock is disabled. 1 = USB 48MHz driving clock is enabled.	0x1
[17]	**USBCK_EN**	**USB operation and driving clock enable bit** 0 = USB operation is disabled. Driving clock to USB is also disabled. 1 = USB operation and driving clock are enabled.	0x1
[15]	**WDT_EN**	**WDT driving clock enable bit** 0 = WDT driving clock disabled. 1 = WDT driving clock enabled, setting to WDT and WDT operation is effective.	0x1
[14]	**TMR_EN**	**Timer operation and driving clock enable bit** 0 = Timer operation is disabled. Driving clock to timer is also disabled. 1 = Timer operation and driving clock are enabled.	0x1
[13]	**PWM_EN**	**PWM operation and driving clock enable bit** 0 = PWM operation is disabled. Driving clock to PWM is also disabled. 1 = PWM operation and driving clock are enabled.	0x1
[12]	**SPI_EN**	**SPI operation and driving clock enable bit** 0 = SPI operation is disabled. Driving clock to SPI is also disabled. 1 = SPI operation and driving clock are enabled.	0x0
[11]	**UART_EN**	**UART operation and driving clock enable bit** 0 = UART operation is disabled. Driving clock to UART is also disabled. 1 = UART operation and driving clock are enabled.	0x1
[10]	**ADC_EN**	**ADC operation and driving clock enable bit** 0 = ADC operation and Touch-Panel operation are disabled. Driving clock to ADC and Touch-Panel modules is also disabled. 1 = ADC and Touch-Panel operation and driving clock are enabled.	0x1
[9]	**GDMA_EN**	**GDMA operation and driving clock enable bit** 0 = GDMA operation is disabled. Driving clock to GDMA is also disabled. 1 = GDMA operation and driving clock are enabled.	0x0
[8]	**GPU_EN**	**Graphic Processing Unit operation and driving clock enable bit** 0 = Graphic Processing Unit operation is disabled. Driving clock to Graphic Processing Unit is also disabled. 1 = Graphic Processing Unit operation and driving clock are enabled.	0x0
[7]	**GE_EN**	**2D Graphic Acceleration operation and driving clock enable bit** 0 = 2D Graphic Acceleration operation is disabled. Driving clock to 2D Graphic Acceleration is also disabled. 1 = 2D Graphic Acceleration operation and driving clock are enabled.	0x0
[6]	**VPST_EN**	**V-Post (Display interface) operation and driving clock enable bit** 0 = V-Post operation is disabled. Driving clock to V-Post is also disabled. 1 = V-Post operation and driving clock are enabled. **NOTE:** V-Post (Display interface) including LCM module controller and TV-Encoder (TV-DAC included) interface	0x1

[5]	**VID_EN**	**Video-IN operation and driving enable clock bit** 0 = Video-in operation is disabled. Driving clock to Video-in is also disabled. 1 = Video-in operation and driving clock are enabled. **NOTE:** Video-in interface is a image sensor interface to receive image streaming from a image sensor.	0x0
[4]	**JPG_EN**	**JPEG operation and driving clock enable bit** 0 = JPEG operation is disabled. 1 = JPEG operation clock is enabled.	0x0
[3]	**SDH_EN**	**SD Card Host Controller/DMAC operation and driving clock enable bit** 0 = SD Card Host Controller/DMAC operation is disabled. Driving clock to SD Card Host Controller/DMAC is also disabled. 1 = SD Card Host Controller/DMAC operation and driving clock are enabled.	0x1
[2]	**AUD_EN**	**Audio Controller and Audio DAC operation and driving clock enable bit** 0 = Audio controller and audio DAC operation is disabled. Driving clock to Audio controller and audio DAC is also disabled. 1 = Audio controller and audio DAC operation and driving clock are enabled. **NOTE:** Directly turn on or turn off audio DAC by toggle this bit can generate unexpected noise. To avoid that issue, please also refer to the audio DAC control part to appropriately control the audio DAC power-down bits.	0x0

NOTE: By default, all setting values of the above described bits are "1". That means all function blocks and their driving clocks are enabled after reset. Users may individually set each block enabled or disabled by their system power saving requirements.

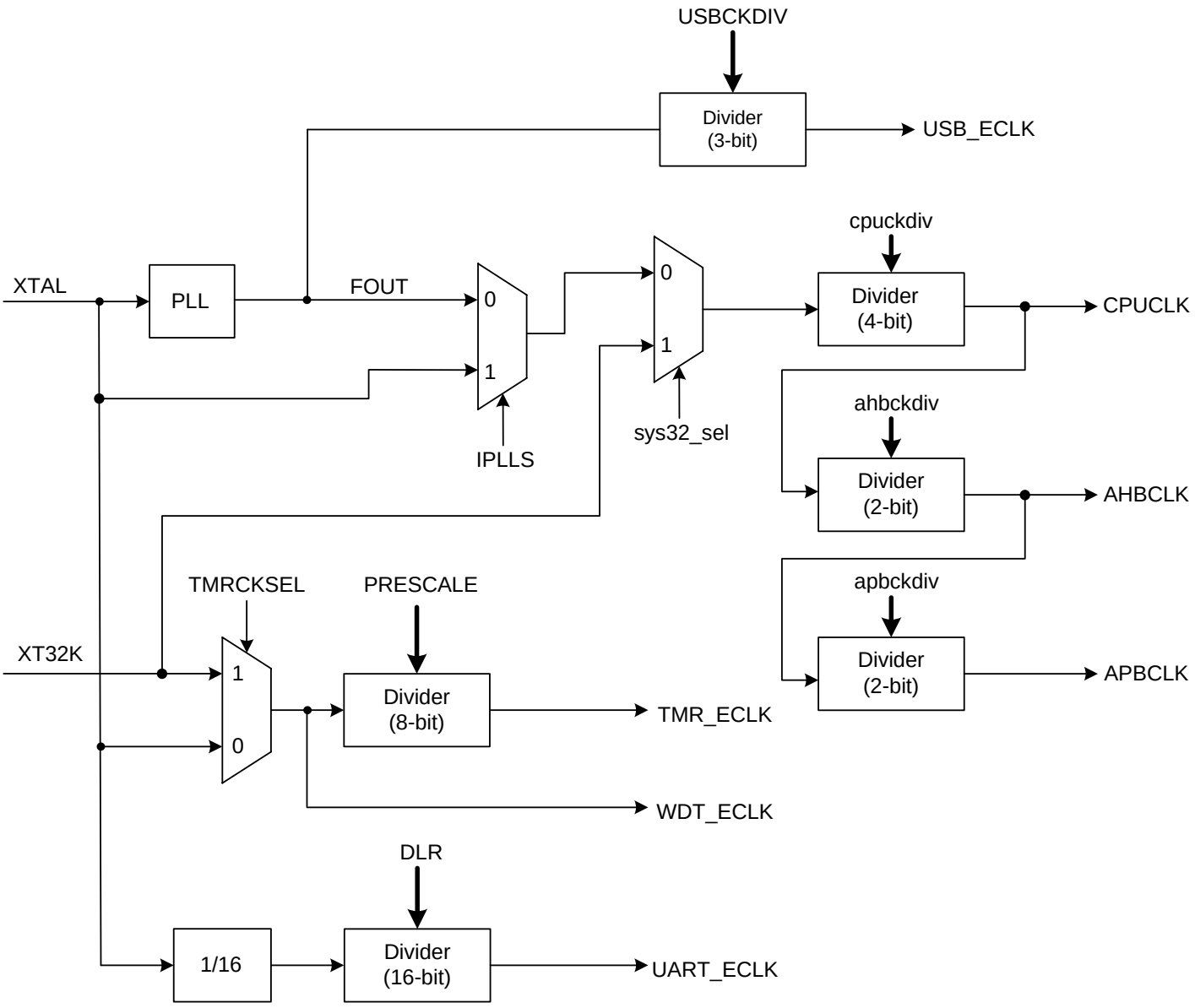
Clock Source Selection and Division Setting Register (CLKSEL)

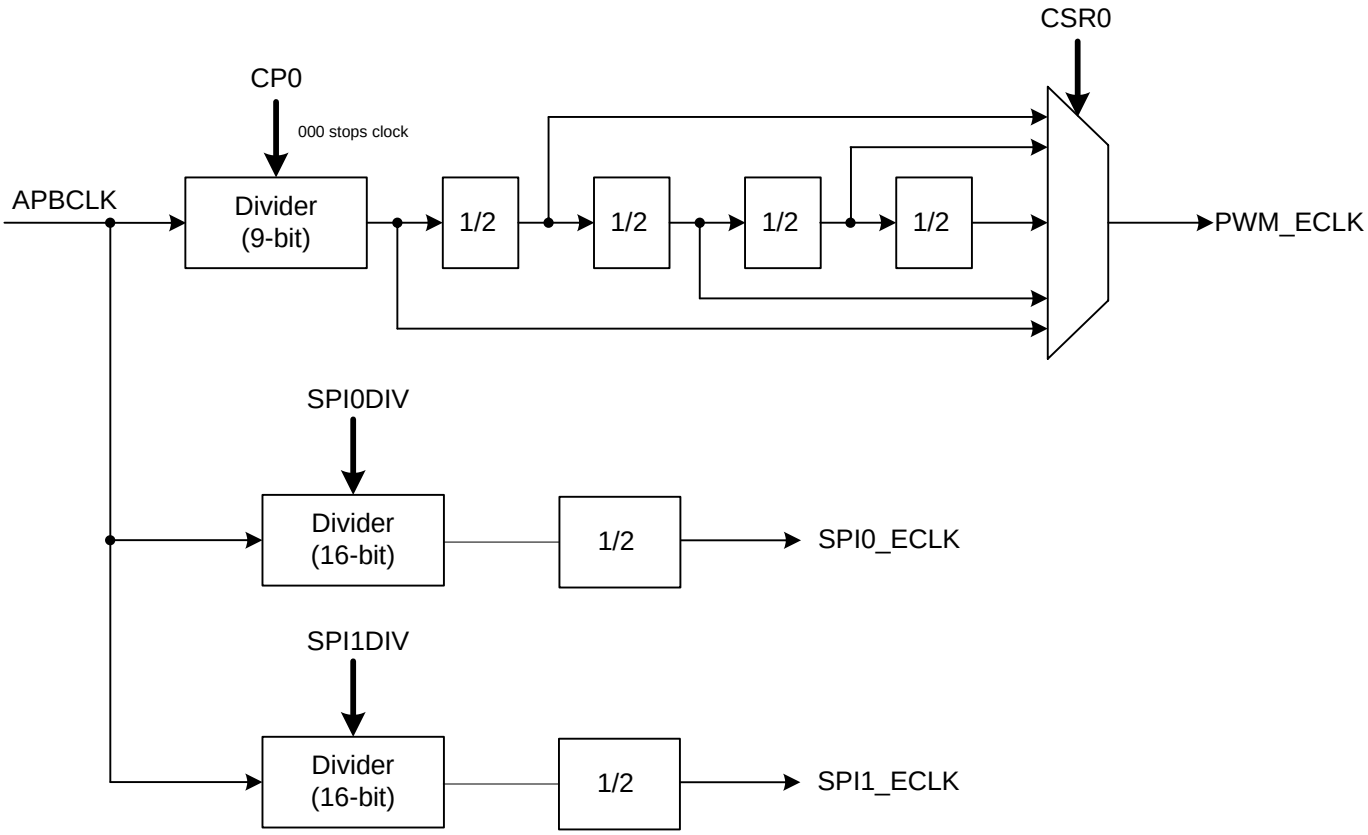
Register	Address	R/W	Description	Reset Value
CLKSEL	SYS_BA+0x014	R/W	Clock source enable/disable management for peripherals	0x0045_17FF

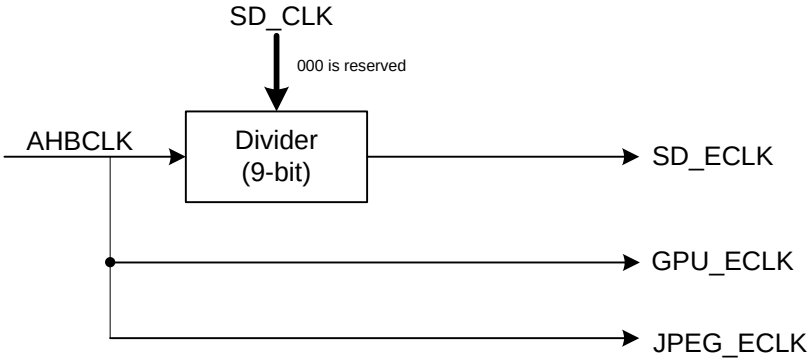
31	30	29	28	27	26	25	24
VLB_HLKEN	VPSTCKSEL	VIDCKSEL	LCDCKSEL	Reserved	TMRCKSEL	SYS32KSL	USBCKDIV
23	22	21	20	19	18	17	16
USBCKDIV			VPSTCKDIV				
15	14	13	12	11	10	9	8
VIDCKDIV						ADOCKDIV	
7	6	5	4	3	2	1	0
ADOCKDIV							

Bits	Descriptions		Default
[31]	**VLB_HCLKEN**	**VPOST Line Buffer system clock enable.** 0 = VPOST Line buffer clock source is not selected from system clock. 1 = VPOST Line buffer clock source is selected from system clock.	0x0
[30]	**VPSTCKSEL**	**VPOST driving clock selection** 0 = Main crystal oscillator clock is selected. 1 = Divided clock from PLL output clock ($FREQ_{DIVOUT}$) is selected.	0x0
[29]	**VIDCKSEL**	**VIDEO-IN (Sensor) driving clock selection** 0 = VPOST driving clock (27/13.5 MHz) is selected. 1 = Independent clock divider output is selected.	0x0
[28]	**LCDCKSEL**	**VPOST(LCD) driving clock selection** 0 = VPOST (LCD) driving clock is set to 13.5MHz clock source. 1 = VPOST (LCD) driving clock is set to 27MHz clock source.	0x0
[26]	**TMRCKSEL**	**Timer driving clock source selection** 0 = Main crystal oscillator clock is selected as clock source. 1 = 32768 crystal oscillator clock is selected as clock source.	0x0
[25]	**SYS32KSL**	**System clock selection from 32.768KHz crystal oscillator** 0 = System clock is from divided PLL generated output or Main crystal oscillator 1 = System clock is from 32.768KHz crystal oscillator	0x0
[24:21]	**USBCKDIV**	**USB 48M clock source divider** The clock divider generate a frequency-divided clock output from PLL-generated clock source for the engine clock(48MHz) of USB device and Host. If N = USBCKDIV[3:0], then $FREQ_{DIVOUT} = (FREQ_{PLL}) / (N+1)$.	0x2
[20:16]	**VPSTCKDIV**	**V-Post (Display Interface) engine clock divider** The clock divider generates a frequency-divided clock output from PLL-generated clock source for the engine clock of Display Interface. If N = VPSTCKDIV[4:0], then $FREQ_{DIVOUT} = (FREQ_{PLL}) / \{2(N+1)\}$.	0x5
[15:10]	**VIDCKDIV**	**Video sensor input controller engine clock divider** The clock divider generates a frequency-divided clock output from PLL-generated clock source for the engine clock of Video sensor input controller. If N = VIDCKDIV[5:0], then $FREQ_{DIVOUT} = (FREQ_{PLL}) / \{2(N+1)\}$.	0x5
[9:0]	**ADOCKDIV**	**Audio DAC clock source divider** The clock divider generates a frequency-divided clock output from PLL-generated clock source for the engine clock of Audio DAC. If N = ADOCKDIV[7:0], then $FREQ_{DIVOUT} = (FREQ_{PLL}) / \{2(N+1)\}$.	0x3FF

The following is a clock control diagram that simply illustrates how register bit values in “CLKMAN” and “CLKSEL” control all the system and peripheral clocks.







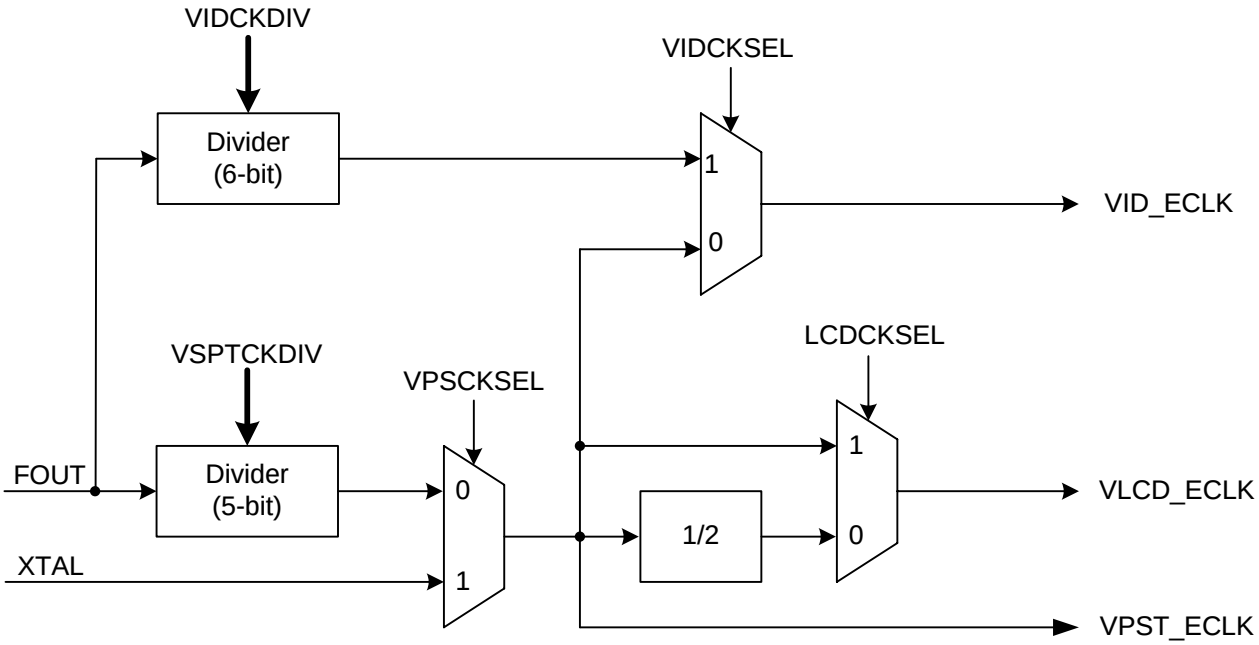


Figure 4.2-10 System and Peripheral Clocks Control Diagram

Reset Control Register (RSTCON)

Register	Address	R/W	Description	Reset Value
RSTCON	SYS_BA+0x018	W	Reset Control Register	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED				**I2CRST**	**UHCRST**	**RTCRST**	**USBRST**
15	14	13	12	11	10	9	8
AUDIFRST	**SPI1RST**	**TMRRST**	**PWMRST**	**SPI0RST**	**UARTRST**	**ADCRST**	**GDMARST**
7	6	5	4	3	2	1	0
GPURST	**GERST**	**VPSTRST**	**VIDRST**	**JPGRST**	**SDHRST**	**ADORST**	**LVRSTEN**

Bits	Descriptions	
[19]	**I2CRST**	**I2C Reset** Writing "1" to this bit will let the I2C block operation terminated. All operating status and all setting values in control registers of I2C will be reset to their default power-on values.
[18]	**UHCRST**	**USB Host Controller Reset** Writing "1" to this bit will let the USB Host Controller block operation terminated. All operating status and all setting values in control registers of USB Host Controller will be reset to their default power-on values.

[17]	**RTCRST**	**RTC Reset** Writing "1" to this bit will let the RTC block operation terminated. All operating status and all setting values in control registers of RTC will be reset to their default power-on values.
[16]	**USBRST**	**USB Controller Reset** Writing "1" to this bit will let the USB block operation terminated. All operating status and all setting values in control registers of USB controller will be reset to their default power-on values.
[15]	**AUDIFRST**	**I2S/AC97 Controller Reset** Writing "1" to this bit will let the I2S/AC97 block operation terminated. All operating status and all setting values in control registers of I2S/AC97 block will be reset to their default power-on values.
[14]	**SPI1RST**	**SPI Interface 1 Reset (Master Only)** Writing "1" to this bit will let SPI 1 block operation terminated. All operating status and all setting values in control registers of SPI controller will be reset to their default power-on values.x
[13]	**TMRRST**	**Timer Reset** Writing "1" to this bit will let Timer block operation terminated and all setting values in control registers of Timer will be reset to their default power-on values.
[12]	**PWMRST**	**PWM Reset** Writing "1" to this bit will let PWM0 and PWM1 blocks operation terminated and all setting values in control registers of PWM will be reset to their default power-on values.
[11]	**SPI0RST**	**SPI Interface 0 Reset** Writing "1" to this bit will let SPI 0 block operation terminated. All operating status and all setting values in control registers of SPI controller will be reset to their default power-on values.
[10]	**UARTRST**	**UART Reset** Writing "1" to this bit will let UART block operation terminated. All operating status and all setting values in control registers of UART controller will be reset to their default power-on values.
[9]	**ADCRST**	**ADC and Touch-Panel Reset** Writing "1" to this bit will let ADC and Touch-Panel blocks operation terminated. All operating status and all setting values in control registers of ADC controller will be reset to their default power-on values.
[8]	**GDMARST**	**GDMA Reset** Writing "1" to this bit will let GDMA block operation terminated. All operating status and all setting values in control registers of GDMA controller will be reset to their default power-on values.
[7]	**GPURST**	**Graphic Processing Unit Reset** Writing "1" to this bit will let Graphic Processing Unit operation terminated. All operating status and all setting values in control registers of Graphic Processing Unit will be reset to their default power-on values.
[6]	**GERST**	**Graphic Acceleration Engine Reset** Writing "1" to this bit will let this block operation terminated and all setting values in control registers will be reset to their default power-on values.

[5]	**VPSTRST**	**V-Post (Display Interface) Reset** Writing "1" to this bit will let the V-Post block operation terminated. All operating status and all setting values in control registers of V-Post controller will be reset to their default power-on values.
[4]	**VIDRST**	**Video Sensor Input Controller Reset** Writing "1" to this bit will let the Video Sensor Input block operation terminated. All operating status and all setting values in control registers of Video input controller will be reset to their default power-on values.
[3]	**JPGRST**	**JPEG Controller Reset** Writing "1" to this bit will let the JPEG block operation terminated. All operating status and all setting values in control registers of JPEG controller will be reset to their default power-on values.
[2]	**SDHRST**	**SD Card Host Controller Reset** Writing "1" to this bit will let the SD Card Host block operation terminated. All operating status and all setting values in control registers of this block will be reset to their default power-on values.
[1]	**ADORST**	**Audio Output Controller Reset** Writing "1" to this bit will let the Audio Controller and Audio DAC processor block operation terminated. All operating status and all setting values in control registers of this block will be reset to their default power-on values.
[0]	**LVRSTEN**	**Low Voltage Reset Enable** 1 = Low Voltage Reset is selected and enabled 0 = Low Voltage Reset is disabled NOTE1: Before setting this bit to '1', be sure PMCON[30]/LVR_WARM has been enabled for 5uS, otherwise, an unexpected reset may follow. NOTe2: Before setting this PMCON[30]/LVR_WARM to '0', be this bit has been disabled to '0', otherwise, an unexpected reset will follow.
NOTE1: All these bits described above don't have meaningful read back values. **NOTE2**: Writing "0" to above bits has no any effects.		

Pin Multifunction Control Register (PINFUN)

Register	Address	R/W	Description	Reset Value
PINFUN	SYS_BA+0x01C	R/W	Pins Multifunction Control Register	0x0000_0000

31	30	29	28	27	26	25	24
OVCR2_SEL				**OVCR1_SEL**			
23	22	21	20	19	18	17	16
PWM11S2	**PWM10S2**	**PWM11S1**	**PWM10S1**	**IISEN2**	**IISEN1**	**IISEN0**	**GPIOD_EN**
15	14	13	12	11	10	9	8
USB_FLO_EN	**VINPIN_EN**	**ECS1PIN_EN**	**SPI1PIN_EN**	**SDHPIN_EN**	**SPI0PIN_EN**	**SPI_SSOEN**	
7	6	5	4	3	2	1	0
UARTPIN_EN	**LCDM_EN2**	**LCDM_EN1**	**LCD16**	**PWM01S2**	**PWM00S2**	**PWM01S1**	**PWM00S1**

Bits	Descriptions	Default

[31:28]	OVCR2_SEL	**USB Host2 Over-Current Indicator Pin Selection** 0000 = USB Host2 over Current Indicator is disappeared. 0001 = USB Host2 over Current Indicator is appeared on GPA[0]. 0010 = USB Host2 over Current Indicator is appeared on GPA[15]. 0100-0111 = USB Host2 over Current Indicator is appeared on GPB[12:15]. 1000-1111 = USB Host2 over Current Indicator is appeared on GPE[0:7].	0x0
[27:24]	OVCR1_SEL	**USB Host1 Over-Current Indicator Pin Selection** 0000 = USB Host1 over Current Indicator is disappeared. 0001 = USB Host1 over Current Indicator is appeared on GPA[0]. 0010 = USB Host1 over Current Indicator is appeared on GPA[15]. 0100-0111 = USB Host1 over Current Indicator is appeared on GPB[12:15]. 1000-1111 = USB Host1 over Current Indicator is appeared on GPE[0:7].	0x0
[23]	PWM11S2	PWM1_1/Cap1_1 Pin Set2 Enable 0 = PWM1_1/Cap1_1 function output doesn't appear on GPE[0]. 1 = GPE[0] will be the PWM1_1 function output, or Cap1_1 input source if CAPENR.7 is set(CAPENR is described in Chapter 7.19 PWM).	0x0
[22]	PWM10S2	PWM1_0/Cap1_0 Pin Set2 Enable 0 = PWM1_0/Cap1_0 function output doesn't appear on GPB[15]. 1 = GPB[15] will be the PWM1_0 function output, or Cap1_0 input source if CAPENR.6 is set(CAPENR is described in Chapter 7.19 PWM).	0x0
[21]	PWM11S1	PWM1_1/Cap1_1 Pin Set1 Enable 0 = PWM1_1/Cap1_1 function output doesn't appear on GPB[12]. 1 = GPB[12] will be the PWM1_1 function output, or Cap1_1 input source if CAPENR.5 is set(CAPENR is described in Chapter 7.19 PWM).	0x0
[20]	PWM10S1	PWM1_0/Cap1_0 Pin Set1 Enable 0 = PWM1_0/Cap1_0 function output doesn't appear on GPB[11]. 1 = GPB[11] will be the PWM1_0 function output, or Cap1_0 input source if CAPENR.4 is set(CAPENR is described in Chapter 7.19 PWM).	0x0
[19]	IISEN2	IIS Pin Enable 2 0 = IIS Audio pins are disabled for IISM_DI 1 = IIS Audio pins are enabled for IISM_DI appearing on GPE[11]. This additional disable/enable bit is for some application that doesn't need a recorder function.	0x0
[18]	IISEN1	IIS Pin Enable 1 0 = IIS Audio pins are disabled for IISM_MCLK 1 = IIS Audio pins are enabled for IISM_MCLK appearing on GPE[7]. This additional disable/enable bit is for some IIS audio device that doesn't rely on MCLK provided by master.	0x0

[17]	IISEN0	IIS Pin Enable 0 0 = IIS Audio pins are disabled for IISM_LRCLK, IISM_BCLK, and IISM_DO. 1 = IIS Audio pins are enabled for IISM_BCLK, IISM_LRCLK, IISM_DO appearing on GPE[8], GPE[9], and GPE[10].	0x0
[16]	GPIOD_EN	GPD General I/O Port Enable 0 = GPD general I/O port disabled. 1 = GPD general I/O port enabled. MA[22:19] is GPD[3:0].	0x0
[15]	USB_FLO_EN	USB Floating detect Pin Enabled 0 = USB floating detect input function doesn't appear on any ports (as PINFUN2 [4:1] USBDET_SEL described). 1 = USB floating detect input function enabled, USB floating detect input is from corresponding ports (as PINFUN2 [4:1] USBDET_SEL described).	0x0
[14]	VINPIN_EN	Video Sensor Input Controller Function Pin Enabled 0 = Video sensor input controller function doesn't appear on I/O pins. 1 = Video sensor input controller function enabled on corresponding I/O pins. VID[7:0] (data inputs from sensor) will be represented on GPE[15:8]. VISCLK (clock output to sensor) will be represented on GPE[4]. VICLK (pixel clock input from sensor) source from GPE[5]. VIVync (vertical sync input from sensor) source from GPE[6]. VIHync (horizontal sync input from sensor) source from FPE[7].	0x0
[13]	ECS1PIN_EN	nECS1, External I/O Bank1 Output Enable 0 = nECS1 output doesn't appear on GPA[15]. 1 = nECS1 output appears on GPA[15].	0x0
[12]	SPI1PIN_EN	SPI Serial Interface 1 Function Pin Enabled 0 = SPI serial interface 1 functions don't appear on I/O pins. 1 = SPI serial interface 1 function enabled on corresponding I/O pins. GPA[1] will be SPI1_SO (serial out, output). GPA[5] will be SPI1_SI (serial in, input). GPA[6] will be SPI1_CLK (SPI clock to slave, output).	0x0
[11]	SDHPIN_EN	SD Card Host Function Pin Enabled 0 = SD Card Host Controller functions don't appear on I/O pins. 1 = SD Card Host Controller function enabled on corresponding I/O pins. GPA[4:1] will be the SD card data bus. GPA[5] will be SD_CMD (output). GPA[6] will be SD_CLK (output). SD Card detect (SD_CD) Pin Selection is described in PINFUN2 SDDET_SEL control register. .	0x0

[10]	SPIPIN_EN	SPI Function Pin Enabled 0 = SPI function doesn't appear on I/O pins. 1 = SPI function enabled on corresponding I/O pins. GPE[1] will be SPI_SS0 (slave select0 output), if SPI_SSOEN[0] is set. GPE[2] will be SPI_SS1 (slave select1 output), if SPI_SSOEN[1] is set. GPE[3] will be SPI_SO (master/slave serial out, output). GPE[4] will be SPI_SI (master/slave serial in, input). GPE[5] will be SPI_CLK (SPI master clock to slave or slave clock from master, output or input, respectively).	0x0
[9:8]	SPI_SSOEN	SPI Slave Selection Output Enabled SPI_SSOEN[1:0] = 0, SPI slave selection output on I/O pins GPE[2:1] is disabled. SPI_SSOEN[1:0] = 1, SPI slave selection output on I/O pins GPE[2:1] is enabled.	0x0
[7]	UARTPIN_EN	UART Function Pin Enabled 0 = UART function doesn't appear on I/O pins. 1 = UART function enabled on corresponding I/O pins. GPA[7] will be UART_RXD (input). GPA[8] will be UART_TXD (output).	0x0
[6]	LCDM_EN2	LCD Module Controller Function Pin (LCD_VDEN, LCD_VSYNC and LCD_HSYNC) Enabled 0 = LCD module controller function doesn't appear on I/O pins. 1 = LCD module controller enabled on corresponding I/O pins. GPB[1:0] will represent LCD control signals: LCD_HSYNC and LCD_VSYNC. GPA[0], will represent LCD_VDEN.	0x0
[5]	LCDM_EN1	LCD Module Controller Function Pin (LCD_DAT and LCD_PIXCLK) Enabled 0 = LCD module controller function doesn't appear on I/O pins. 1 = LCD module controller enabled on corresponding I/O pins. If LCD16=0, GPB[10:3] will represent LCD_DAT[7:0]. If LCD16=1, GPE[2:0] & GPB[15:3] will represent LCD_DAT[15:0] (data outputs to LCD module). GPB[2] will represent LCD control signals, LCD_PIXCLK.	0x0
[4]	LCD16	8/16 bits LCD data selection 0 = 8 bits LCD data [7:0] output from GPB[10:3], if LCDM_EN is Enabled. 1 = 16 bits LCD data [15:0] output from GPE[2:0] and GPB[15:3], if LCDM_EN is Enabled.	0x0
[3]	PWM01S2	PWM0_1/Cap0_1 Pin Set2 Enable 0 = PWM0_1/Cap0_1 function output doesn't appear on GPB[14]. 1 = GPB[14] will be the PWM0_1 function output, or Cap0_1 input source if CAPENR.3 is set(CAPENR is described in Chapter 7.19 PWM).	0x0

[2]	PWM00S2	PWM0_0/Cap0_0 Pin Set2 Enable 0 = PWM0_0/Cap0_0 function output doesn't appear on GPB[13]. 1 = GPB[13] will be the PWM0_0 function output, or Cap0_0 input source if CAPENR.2 is set(CAPENR is described in Chapter 7.19 PWM).	0x0
[1]	PWM01S1	PWM0_1/Cap0_1 Pin Set1 Enable 0 = PWM0_1/Cap0_1 function output doesn't appear on GPA[8]. 1 = GPA[8] will be the PWM0_1 function output, or Cap0_1 input source if CAPENR.1 is set(CAPENR is described in Chapter 7.19 PWM).	0x0
[0]	PWM00S1	PWM0_0/Cap0_0 Pin Set1 Enable 0 = PWM0_0/Cap0_0 function doesn't appear on GPA[7]. 1 = GPA[7] will be the PWM0_0 function output, or Cap0_0 input source if CAPENR.0 is set(CAPENR is described in Chapter 7.19 PWM).	0x0

Crystal on/off status Control Register (XTLCON)

Register	Address	R/W	Description	Reset Value
XTLCON	SYS_BA+0x020	R/W	Crystal on/off status Control Register	0x0000_0009

31	30	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				XT32KOFF	XTOFF	XT32KEN	XTEN

There are two sets of crystal oscillator circuits in this chip. One is Main Crystal Oscillator providing high frequency (6~15MHz) clock source for system normal operation and PLL module clock input. The other is 32K Sub-Crystal Oscillator providing real time clock (32768Hz) for Timer counting or system operation in SLOW mode. The following 4 registers showing the access ways to control and monitor the on/off status of these oscillator circuits.

Bits	Descriptions		Default
[3]	**XT32KOFF**	**32768Hz Sub-Crystal Oscillator On/Off State (Read Only)** 0 = Crystal oscillator is in operation. 1 = Crystal oscillator circuit is turn off (disabled).	0x1
[2]	**XTOFF**	**Main Crystal Oscillator On/Off State (Read Only)** 0 = Crystal oscillator is in operation. 1 = Crystal oscillator circuit is turn off (disabled).	0x0

[1]	**XT32KEN**	**32768Hz Sub-Crystal Oscillator Enable Control (R/W)** 0 = Disable the operation of the crystal oscillator. 1 = Enable the operation of the crystal oscillator. **NOTE:** After setting this bit from '1' to '0', immediate reading XT32KOFF will obtain a '1' (disabled). However, setting this bit from '0' to '1' following reading XT32KOFF won't get '0' immediately. Clock control circuits will count enough clock cycle to ensure the crystal oscillator output stable, then reset XT32KOFF to '0'.	0x0
[0]	**XTEN**	**Main Crystal Oscillator Enable Control (R/W)** 0 = Disable the operation of the crystal oscillator. 1 = Enable the operation of the crystal oscillator. **NOTE:** After setting this bit from '1' to '0', immediate reading XTOFF will obtain a '1' (disabled). However, setting this bit from '0' to '1' following reading XTOFF won't get '0' immediately. Clock control circuits will count enough clock cycle to ensure the crystal oscillator output stable, then reset XTOFF to '0'.	0x1

IRQ Wakeup Control Register (IRQWAKECON)

Register	Address	R/W	Description	Reset Value
IRQWAKECON	SYS_BA+0x024	R/W	IRQ Wakeup 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
IRQWAKEUPPOL				IRQWAKEUPEN			

Bits	Descriptions		Default
[7:4]	**IRQWAKEUPPOL**	**IRQ wakeup polarity** 1 = nIRQx is high level wakeup 0 = nIRQx is low level wakeup	0x0
[3:0]	**IRQWAKEUPEN**	**nIRQ[3:0] wakeup enable** 1 = nIRQx wakeup enable 0 = nIRQx wakeup disable	0x0

IRQ Wakeup Flag Register (IRQWAKEFLAG)

Register	Address	R/W	Description	Reset Value
IRQWAKEFLAG	SYS_BA+0x028	R/W	IRQ Wakeup Flag 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				IRQWAKEFLAG			

Bits	Descriptions		Default
[3:0]	**IRQWAKEFLAG**	**nIRQ[3:0] wakeup flag** This register is used to record the wakeup event, after clock is recovery, software should check these flags to identify which IRQ is used to wakeup the system. And clear the flags (write 0) in IRQ interrupt sevice routine. 1 = CPU is wakeup by nIRQx 0 = not wakeup	0x0

Power Management Control Register (PMCON)

Register	Address	R/W	Description	Reset Value
PMCON	SYS_BA+0x02C	R/W	Power Management Control Register	0x0000_0000

31	30	29	28	27	26	25	24
WARM_RST	LVR_WARM	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					MIDLE	PD	IDLE

Bits	Descriptions		Default
[31]	**WARM_RST**	**Warm system reset by software** 1 = write 1 to this bit cause whole chip reset immediately. 0 = no any effects occur.	0x0

[30]	**LVR_WARM**	**Low Voltage Reset Warm Up** 1 = Low Voltage Reset function is warmed up to operate, Low voltage reset signal is de-asserted by default. And asserts only when VDD 3.3V power is low to 2.4V. 0 = Low Voltage Reset function is not warmed up to operate, by this setting, Low voltage reset signal always asserts as a disabled output. **NOTE1:** When Warmed up is set, the LVR function block will always consumes a current of about several tens micro Amps. **NOTE2:** Do not disable this bit when RSTCON[0]/LVRSTEN is still set as '1'. Otherwise, an immediate reset follows the ill setting. And be sure to enable this bit before LVRSTEN enabled for at least 5 uS.	0x0
[2]	**MIDLE**	**Memory controller IDLE enable** Setting both MIDLE and IDLE bits HIGH will enable memory controller enter IDLE mode, the clock source of memory controller will be halted while ARM CORE enter IDLE mode. Software must let SDRAM enter self-refresh mode before enable this function. 1 = memory controller will enter IDLE mode when IDLE bit is set. 0 = memory controller still active when IDLE bit is set.	0x0
[1]	**PD**	**Power down enable** Setting this bit HIGH will let W55VA91 enter power saving mode. The clock source 12MHz/6MHz crystal oscillator and PLL are stopped to generate clock. User can use nIRQ[3:0], keypad and external RESET to wakeup W55VA91. 1 = power down mode 0 = not enable	0x0
[0]	**IDLE**	**IDLE mode** Setting this bit HIGH will let ARM Core enter power saving mode. The peripherals can still keep working if the clock enable bit in CONSEL is set. Any nIRQ or nFIQ to ARM Core will let ARM CORE to exit IDLE state. 1 = IDEL mode 0 = not enable	0x0

GPA/B I/O Cells Dual-Driving Select Control Register (GPABDS)

Register	Address	R/W	Description	Reset Value
GPABDS	SYS_BA+0x030	R/W	GPA/B dual driving output selection	0x0000_0000

31	30	29	28	27	26	25	24
GPB15_DS	GPB14_DS	GPB13_DS	GPB12_DS	GPB11_DS	GPB10_DS	GPB9_DS	GPB8_DS
23	22	21	20	19	18	17	16
GPB7_DS	GPB6_DS	GPB5_DS	GPB4_DS	GPB3_DS	GPB2_DS	GPB1_DS	GPB0_DS
15	14	13	12	11	10	9	8
GPA15_DS	GPA14_DS	GPA13_DS	GPA12_DS	GPA11_DS	GPA10_DS	GPA9_DS	GPA8_DS
7	6	5	4	3	2	1	0
GPA7_DS	GPA6_DS	GPA5_DS	GPA4_DS	GPA3_DS	GPA2_DS	GPA1_DS	GPA0_DS

Bits	Descriptions		Default
[31:16]	**GPBx_DS**	**GPB I/O Cells Dual-Driving Output selection** 0 = set GPBx I/O Cells as low drive output. 1 = set GPBx I/O Cells as high drive output.	0x0
[15:0]	**GPAx_DS**	**GPA I/O Cells Dual-Driving Output selection** 0 = set GPAx I/O Cells as low drive output. 1 = set GPAx I/O Cells as high drive output.	0x0

GPC/E I/O Cells Dual-Driving Select Control Register (GPCEDS)

Register	Address	R/W	Description	Reset Value
GPCEDS	SYS_BA+0x034	R/W	GPC/E dual driving output selection	0x0000_0000

31	30	29	28	27	26	25	24
GPE15_DS	GPE14_DS	GPE13_DS	GPE12_DS	GPE11_DS	GPE10_DS	GPE9_DS	GPE8_DS
23	22	21	20	19	18	17	16
GPE7_DS	GPE6_DS	GPE5_DS	GPE4_DS	GPE3_DS	GPE2_DS	GPE1_DS	GPE0_DS
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
GPC7_DS	GPC6_DS	GPC5_DS	GPC4_DS	GPC3_DS	GPC2_DS	GPC1_DS	GPC0_DS

Bits	Descriptions		Default
[31:16]	**GPEx_DS**	**GPE I/O Cells Dual-Driving Output selection** 0 = set GPEx I/O Cells as low drive output. 1 = set GPEx I/O Cells as high drive output.	0x0
[7:0]	**GPCx_DS**	**GPC I/O Cells Dual-Driving Output selection** 0 = set GPCx I/O Cells as low drive output. 1 = set GPCx I/O Cells as high drive output.	0x0

EBI I/O Cells Triple-Driving Select Control Register (EBIDS)

Register	Address	R/W	Description	Reset Value
EBIDS	SYS_BA+0x038	R/W	EBI triple driving output selection	0x0000_00AA

31	30	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						ND_DS	MCLK_DS
7	6	5	4	3	2	1	0
EBIC_DS		MD_DS		MAH_DS		MAL_DS	

Bits	Descriptions		Default
[9]	**ND_DS**	**NAND Flash Data ports [7:0] Dual-Driving Output selection** 0 = set ND[7:0] I/O Cell as low drive output. 1 = set ND[7:0] I/O Cell as high drive output.	0x00
[8]	**MCLK_DS**	**MCLK Control I/O Cells Dual-Driving Output selection** 0 = set MCLK I/O Cell as low drive output. 1 = set MCLK I/O Cell as high drive output.	0x00
[7:6]	**EBIC_DS**	**EBIC Control I/O Cells Triple-Driving Output selection** 00 = set EBIC Control I/O Cell as low drive output. 10 = set EBIC Control I/O Cell as middle drive output. 11 = set EBIC Control I/O Cell as high drive output. 01 = reserved. Note: This setting select EBI control pins (including nSCAS, nSRAS, nWE, MCKE, nSCS0, nSCS1, nBTCS, nECS0, nOE nWBE_SDQM0 ~ 3) triple-driving output capability.	0x10
[5:4]	**MD_DS**	**External memory data bus I/O Cells Triple-Driving Output selection** 00 = set MD[31:0] I/O Cell as low drive output. 10 = set MD[31:0] I/O Cell as middle drive output. 11 = set MD[31:0] I/O Cell as high drive output. 01 = reserved.	0x10
[3:2]	**MAH_DS**	**External memory address [22:15] I/O Cells Triple-Driving Output selection** 00 = set MA[22:15] I/O Cell as low drive output. 10 = set MA[22:15] I/O Cell as middle drive output. 11 = set MA[22:15] I/O Cell as high drive output. 01 = reserved.	0x10
[1:0]	**MAL_DS**	**External memory address[14:0] I/O Cells Triple-Driving Output selection** 00 = set MA[14:0] I/O Cell as low drive output. 10 = set MA[14:0] I/O Cell as middle drive output. 11 = set MA[14:0] I/O Cell as high drive output. 01 = reserved.	0x10

System clock Division Setting Register (SYSDIV)

Register	Address	R/W	Description	Reset Value
SYSDIV	SYS_BA+0x03C	R/W	System clock division management for CPU,AHB,APB clock	0x0000_1111

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED			SDCKDIV				
15	14	13	12	11	10	9	8

SDCKDIV					RESERVED		JPEGCKDIV
7	6	5	4	3	2	1	0
APBCKDIV		AHBCKDIV		CPUCKDIV			

Bits	Descriptions		Default
[20:11]	**SDCKDIV**	**SD clock source divider** The clock divider generates a frequency-divided clock output from AHB clock for the engine clock of SD. If N = SDCKDIV[3:0], then $FREQ_{DIVOUT} = (FREQ_{ahb}) / (N+1)$.	0x2
[8]	**JPEGCKDIV**	**JPEG clock source divider** The clock divider generates a frequency-divided clock output from AHB clock for the engine clock of JPEG. If JPEGCKDIV =0, $FREQ_{JPEG\_eclk} = FREQ_{ahb}$ If JPEGCKDIV =1, $FREQ_{JPEG\_eclk} = FREQ_{ahb}$ /2	0x1
[7:6]	**APBCKDIV**	**AMBA APB Clock Divider Control Register** APBCKDIV: 0 0 → $FREQ_{ahb}$; 0 1 → $FREQ_{ahb}$ /2; 1 0 → $FREQ_{ahb}$ /4; 1 1 → $FREQ_{ahb}$ /8 Note: APBCLK must <= 67MHz.	0x0
[5:4]	**AHBCKDIV**	**AMBA AHB Clock (AHBCLK) Divider Control Register** AHBCKDIV: 0 0 → $FREQ_{cpu}$; 0 1 → $FREQ_{cpu}$ /2; 1 0 → $FREQ_{cpu}$ /4; 1 1 → $FREQ_{cpu}$ /8 Note: AHBCLK must <= 133MHz.	0x1
[3:0]	**CPUCKDIV**	**CPU Clock Source Divider Control Register** $FREQ_{cpu} = (FREQ_{PLL}) / (N+1)$. Where CPUCKDIV is 0~7, and 8~15 is reserved Note: CPUCLK must <= 200MHz.	0x1

Internal Boot ROM Base (IBTBAS)

Register	Address	R/W	Description	Reset Value
IBTBAS	SYS_BA+0x040	R/W	Internal Boot ROM Base Address	0x0000_0000

31	30	29	28	27	26	25	24
IBTBAS							

23	22	21	20	19	18	17	16
IBTBAS							
15	14	13	12	11	10	9	8
IBTBAS		RESERVED					
7	6	5	4	3	2	1	0
RESERVED							

Bits	Descriptions		Default
[31:14]	**IBTBAS**	**Internal Boot ROM Address** **NOTE:** The boot ROM Address has 16K-Byte boundary limit. A wrong setting value can cause the Boot ROM address space overlap the existing EBI address.	0x0

Working RAM Mapping to 0x0 (WMAP0)

Register	Address	R/W	Description	Reset Value
WMAP0	SYS_BA+0x044	R/W	Working (or Video) RAM mapping to start address 0x0000_0000	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						WRAM2MAP0	WRAM1MAP0

Bits	Descriptions		Default
[1]	**WRAM2MAP0**	**WRAM block 2 map to 0x0** 1 = Working (Video) RAM (ranged from 0xFFE7_A000 to 0xFFE7_BFFF) can be accessed by address from 0x0000_0000 to 0x0000_1FFF. 0 = Nothing changed.	0x0
[0]	**WRAM1MAP0**	**WRAM block 1 map to 0x0** 1 = Working (Video) RAM (ranged from 0xFFE7_8000 to 0xFFE7_9FFF) can be accessed by address from 0x0000_0000 to 0x0000_1FFF. 0 = Nothing changed.	0x0
NOTE: Don't enable these two bits concurrently, otherwise the system behavior can't be predicted.			

Pin Multifunction 2 Control Register (PINFUN2)

Register	Address	R/W	Description	Reset Value
PINFUN2	SYS_BA+0x048	R/W	Pins Multifunction 2 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		GPIOC_EN	VIF_EN	ND_PUPEN	ND_EN		SDDET_SEL
7	6	5	4	3	2	1	0
SDDET_SEL			USBDET_SEL				I2CPIN_EN

Bits	Descriptions		Default
[13]	GPIOC_EN	**General Purpose I/O Port Enabled** 0 = General purpose I/O port isn't enabled on EBI interface MD[23:16]. 1 = General purpose I/O port is enabled on EBI interface MD[23:16].	0x0
[12]	VIF_EN	**VideoIn TV Field Input Pin Enable** 0 = VideoIn TV Field Input pin don't appear on GPE[3]. 1 = VideoIn TV Field Input pin appear on GPE[3]. Note: This pin (VIField) is for indicating Even or Odd field for the current field.	0x0
[11]	ND_PUPEN	**NAND Flash Data Ports Pull Up Enable** 0 = NAND Flash Data ports ND[7:0] pull up disable. 1 = NAND Flash Data ports ND[7:0] pull up enable.	0x0
[10:9]	ND_EN[1:0]	**NAND Flash Data Pins Enable** 00 = NAND Flash Data pins ND[7:0] don't appear on MD[31:24] or GPB[10:3]. 01 = NAND Flash Data pins ND[7:0] appear on MD[31:24]. 10 = NAND Flash Data pins ND[7:0] appear on GPB[10:3]. 11 = Reserved.	0x0
[8:5]	SDDET_SEL	**SD Card detect Pin Selection** 0000 = SD Card detect Pin is disappeared. 0001 = SD Card g detect Pin is appeared on GPA[0]. 0010 = SD Card detect Pin is appeared on GPA[15]. 0100-0111 = SD Card detect Pin is appeared on GPB[12:15]. 1000-1111 = SD Card detect Pin is appeared on GPE[0:7].	0x0
[4:1]	USBDET_SEL	**USB Floating detect Pin Selection** 0000 = USB Floating detect Pin is disappeared. 0001 = USB Floating detect Pin is appeared on GPA[0]. 0010 = USB Floating detect Pin is appeared on GPA[15]. 0100-0111 = USB Floating detect Pin is appeared on GPB[12:15]. 1000-1111 = USB Floating detect Pin is appeared on GPE[0:7].	0x0
[0]	I2CPIN_EN	**I2C Pin Enable** 0 = I2C pins don't appear on GPA[8:7]. 1 = I2C pins appear on GPA[8:7]. GPA[8] will be I2C_DATA GPA[7] will be I2C_CLK	0x0

CPU RAM BIST Control Register (RAMBIST)

Register	Address	R/W	Description	Reset Value
RAMBIST	SYS_BA+0x050	R/W	CPU RAM BIST 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						BISTEN	START

Bits	Descriptions	
[1]	BISTEN	**CPU RAM BIST Enable** 0= CPU RAM in normal mode 1= Set CPU RAM (such as I-Cache, D-Cache and MMU) in BIST test mode
[0]	START	**CPU RAM BIST Start Trigger** 0 = None 1 = when this bit is set with BISTEN bit active to trigger the CPU RAMs BIST testing; this bit is clear automatically.

CPU RAM BIST Status Register (RAMSTAT)

Register	Address	R/W	Description	Reset Value
RAMSTAT	SYS_BA+0x054	R	CPU RAM BIST 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
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	
[29]	**ICV_F**	**I-Cache Valid RAM BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[28]	**ICT_F**	**I-Cache TAG RAM BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[27]	**ICD3_F**	**I-Cache Data RAM 3 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[26]	**ICD2_F**	**I-Cache Data RAM 2 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[25]	**ICD1_F**	**I-Cache Data RAM 1 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[24]	**ICD0_F**	**I-Cache Data RAM 0 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[23]	**MMU_F**	**MMU RAM BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[22]	**DCDIR_F**	**D-Cache Dirty RAM BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[21]	**DCV_F**	**D-Cache Valid RAM BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[20]	**DCT_F**	**D-Cache TAG RAM BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[19]	**DCD3_F**	**D-Cache Data RAM 3 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[18]	**DCD2_F**	**D-Cache Data RAM 2 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[17]	**DCD1_F**	**D-Cache Data RAM 1 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[16]	**DCD0_F**	**D-Cache Data RAM 0 BIST Fail Flag** 1= RAM Fail; 0=RAM OK
[13]	**ICV_R**	**I-Cache Valid RAM BIST Running Flag** 1= RAM Running; 0=RAM OK
[12]	**ICT_R**	**I-Cache TAG RAM BIST Running Flag** 1= RAM Running; 0=RAM OK
[11]	**ICD3_R**	**I-Cache Data RAM 3 BIST Running Flag** 1= RAM Running; 0=RAM OK
[10]	**ICD2_R**	**I-Cache Data RAM 2 BIST Running Flag** 1= RAM Running; 0=RAM OK
[9]	**ICD1_R**	**I-Cache Data RAM 1 BIST Running Flag** 1= RAM Running; 0=RAM OK
[8]	**ICD0_R**	**I-Cache Data RAM 0 BIST Running Flag** 1= RAM Running; 0=RAM OK
[7]	**MMU_R**	**MMU RAM BIST Running Flag** 1= RAM Running; 0=RAM OK
[6]	**DCDIR_R**	**D-Cache Dirty RAM BIST Running Flag** 1= RAM Running; 0=RAM OK
[5]	**DCV_R**	**D-Cache Valid RAM BIST Running Flag** 1= RAM Running; 0=RAM OK
[4]	**DCT_R**	**D-Cache TAG RAM BIST Running Flag**

		1= RAM Running; 0=RAM OK
[3]	**DCD3_R**	**D-Cache Data RAM 3 BIST Running Flag** 1= RAM Running; 0=RAM OK
[2]	**DCD2_R**	**D-Cache Data RAM 2 BIST Running Flag** 1= RAM Running; 0=RAM OK
[1]	**DCD1_R**	**D-Cache Data RAM 1 BIST Running Flag** 1= RAM Running; 0=RAM OK
[0]	**DCD0_R**	**D-Cache Data RAM 0 BIST Running Flag** 1= RAM Running; 0=RAM OK

4.3 External Bus Interface

4.3.1 Overview

W55VA91 supports External Bus Interface (**EBI**), which controls the access to the external memory (ROM/FLASH, SDRAM) and External I/O devices. The **EBI** has four chip selects to select one ROM/FLASH bank, two SDRAM banks, and one External I/O bank. The chip selects for SDRAM banks can be optionally configured as chip selects for ROM/FLASH devices, hence the EBI can access up to 3 ROM/FLASH devices. The 22-bit address bus on the EBI can address at most 4 Mega data space for the connected devices. The 32-bit width data bus supports 8-bit, 16-bit, and 32-bit width external data transfer for each bank.
The EBI has the following functions：

- SDRAM controller
- EBI control register
- ROM/FLASH interface
- External I/O interface
- Customer code protection
- NAND-type flash 256-Byte Error Correction Code (ECC) Calculation

4.3.2 SDRAM Controller

The SDRAM controller module within W55VA91 contains configuration registers, timing control registers, common control registers and other logic to provide 8、16、32 bits SDRAM interface with a single 8, 16, or 32 bits SDRAM device, or two 8-bit devices wired to give a 16-bit data path or two 16-bit devices wired to give a 32-bit data path. The maximum size of each ROM, Flash or External IO bank is 4M beats. The meaning of beat is byte, half-word (16-bit) or word (32-bit), depending on the bit width that is user-programmable in the corresponding controller registers. For the two SDRAM banks, each one can connect to a SDRAM device with a size up to 32MB, so the total SDRAM memory space can be spanned up to 64MB by these two banks.

The SDRAM controller has the following features：

- Supports up to 2 external SDRAM devices
- Maximum size of each device is 32MB
- 8、16、32-bit data interface
- Programmable CAS Latency： 1、2 and 3
- Fixed Burst Length: 1
- Sequential burst type
- Write Burst Length mode is Burst
- Auto Refresh Mode and Self Refresh Mode
- Adjustable Refresh Rate
- Power up sequence

4.3.2.1 SDRAM Components Supported

Size	Type	Banks	Row Addressing	Column Addressing
16M bits	2Mx8	2	RA0~RA10	CA0~CA8
	1Mx16	2	RA0~RA10	CA0~CA7

64M bits	8Mx8	4	RA0~RA11	CA0~CA8
	4Mx16	4	RA0~RA11	CA0~CA7
	2Mx32	4	RA0~RA10	CA0~CA7
128M bits	16Mx8	4	RA0~RA11	CA0~CA9
	8Mx16	4	RA0~RA11	CA0~CA8
	4Mx32	4	RA0~RA11	CA0~CA7
256M bits	32Mx8	4	RA0~RA12	CA0~CA9
	16Mx16	4	RA0~RA12	CA0~CA8

Table 4.3-1 Supported SDRAM Components

4.3.2.2 AHB Bus Address Mapping to SDRAM Bus

Note: * indicates the signal is not used; ** indicates the signal is fixed at logic 0 and is not used;
The HADDR prefixes have been omitted on the following tables.
A14 ~ A0 are the Address pins of the W55VA91 EBI interface;
A14 and A13 are the Bank Selected Signal of SDRAM.

Total	Type	R x C	R/C	A14 (BS1)	A13 (BS0)	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16M	2Mx8	11x9	R	**	11	**	11*	22	21	20	19	18	17	16	15	14	13	12
			C	**	11	**	11*	AP	25*	10	9	8	7	6	5	4	3	2
16M	1Mx16	11x8	R	**	10	**	10*	11	21	20	19	18	17	16	15	14	13	12
			C	**	10	**	10*	AP	25*	10*	9	8	7	6	5	4	3	2
64M	8Mx8	12x9	R	11	12	11*	23	22	21	20	19	18	17	16	15	14	13	24
			C	11	12	11*	23*	AP	25*	10	9	8	7	6	5	4	3	2
64M	4Mx16	12x8	R	11	10	11*	23	22	21	20	19	18	17	16	15	14	13	12
			C	11	10	11*	23*	AP	25*	24*	9	8	7	6	5	4	3	2
64M	2Mx32	11x8	R	11	10	11*	23*	22	21	20	19	18	17	16	15	14	13	12
			C	11	10	11*	23*	AP	25*	24*	9	8	7	6	5	4	3	2
128M *	16Mx8	12x10	R	11	12	11*	23	22	21	20	19	18	17	16	15	14	13	24
			C	11	12	11*	23*	AP	25	10	9	8	7	6	5	4	3	2
128M	8Mx16	12x9	R	11	12	11*	23	22	21	20	19	18	17	16	15	14	13	24
			C	11	12	11*	23*	AP	25*	10	9	8	7	6	5	4	3	2
128M	4Mx32	12x8	R	11	10	11*	23	22	21	20	19	18	17	16	15	14	13	12
			C	11	10	11*	23*	AP	25*	10*	9	8	7	6	5	4	3	2
256M *	32Mx8	13x10	R	11	12	24	23	22	21	20	19	18	17	16	15	14	13	25
			C	11	12	24*	23*	AP	26*	10	9	8	7	6	5	4	3	2
256M *	16Mx16	13x9	R	11	12	24	23	22	21	20	19	18	17	16	15	14	13	25
			C	11	12	24*	23*	AP	26*	10*	9	8	7	6	5	4	3	2

Table 4.3-2 SDRAM Data Bus Width: 32-bit

Total	Type	R x C	R/C	A14 (BS1)	A13 (BS0)	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16M	2Mx8	11x9	R	**	10	**	10*	21	20	19	18	17	16	15	14	13	12	11
			C	**	**10**	**	10*	AP	24*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
16M	1Mx16	11x8	R	**	9	**	9*	**10**	20	19	18	17	16	15	14	13	12	11
			C	**	9	**	9*	AP	24*	9*	**8**	**7**	**6**	**5**	**4**	**3**	2	1
64M	8Mx8	12x9	R	10	11	10*	22	21	20	19	18	17	16	15	14	13	12	**23**
			C	10	11	10*	22*	AP	24*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
64M	4Mx16	12x8	R	10	9	10*	22	21	20	19	18	17	16	15	14	13	12	11
			C	10	9	10*	22*	AP	24*	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1
64M	2Mx32	11x8	R	10	9	10*	22*	21	20	19	18	17	16	15	14	13	12	11
			C	10	9	10*	22*	AP	24*	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1
128M	16Mx8	12x10	R	10	11	10*	22	21	20	19	18	17	16	15	14	13	12	**23**
			C	10	11	10*	22*	AP	**24**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
128M	8Mx16	12x9	R	10	11	10*	22	21	20	19	18	17	16	15	14	13	12	**23**
			C	10	11	10*	22*	AP	24*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
128M	4Mx32	12x8	R	10	9	10*	22	21	20	19	18	17	16	15	14	13	12	11
			C	10	9	10*	22*	AP	24*	9*	**8**	**7**	**6**	**5**	**4**	**3**	2	1
256M*	32Mx8	13x10	R	10	11	23	22	21	20	19	18	17	16	15	14	13	12	24
			C	10	11	23*	22*	AP	25*	**9**	**8**	**7**	**6**	**5**	**4**	**3**	2	1
256M	16Mx16	13x9	R	10	11	23	22	21	20	19	18	17	16	15	14	13	12	**24**
			C	10	11	23*	22*	AP	25*	9	**8**	**7**	**6**	**5**	**4**	**3**	2	1

Table 4.3-3 SDRAM Data Bus Width: 16-bit

Total	Type	R x C	R/C	A14 (BS1)	A13 (BS0)	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
16M	2Mx8	11x9	R	**	9	**	9*	20	19	18	17	16	15	14	13	12	11	10
			C	**	**9**	**	9*	AP	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
16M	1Mx16	11x8	R	**	8	**	8*	**9**	19	18	17	16	15	14	13	12	11	10
			C	**	8	**	8*	AP	23*	8*	**7**	**6**	**5**	**4**	**3**	2	1	0
64M	8Mx8	12x9	R	9	10	9*	21	20	19	18	17	16	15	14	13	12	11	**22**
			C	9	10	9*	21*	AP	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1	1
64M	4Mx16	12x8	R	9	8	9*	21	20	19	18	17	16	15	14	13	12	11	10
			C	9	8	9*	21*	AP	23*	22*	**7**	**6**	**5**	**4**	**3**	2	1	0
64M	2Mx32	11x8	R	9	8	9*	21*	20	19	18	17	16	15	14	13	12	11	10
			C	9	8	9*	21*	AP	23*	22*	**7**	**6**	**5**	**4**	**3**	2	1	0
128M	16Mx8	12x10	R	9	10	9*	21	20	19	18	17	16	15	14	13	12	11	**22**
			C	9	10	9*	21*	AP	23	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
128M	8Mx16	12x9	R	9	10	9*	21	20	19	18	17	16	15	14	13	12	11	**22**
			C	9	10	9*	21*	AP	23*	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0
128M	4Mx32	12x8	R	9	8	9*	21	20	19	18	17	16	15	14	13	12	11	10
			C	9	8	9*	21*	AP	23*	8*	**7**	**6**	**5**	**4**	**3**	2	1	0
256M	32Mx8	13x10	R	9	10	22	21	20	19	18	17	16	15	14	13	12	11	**23**
			C	9	10	22*	21*	AP	24	**8**	**7**	**6**	**5**	**4**	**3**	2	1	0

256M	16Mx16	13x9	R	9	10	22	21	20	19	18	17	16	15	14	13	12	11	23
			C	9	10	22*	21*	AP	24*	8	7	6	5	4	3	2	1	0

Table 4.3-4 SDRAM Data Bus Width: 8-bit

4.3.2.3 SDRAM Power Up Sequence

The default value of the mode register is not defined, therefore the mode register must be written after power up to operate the SDRAM. W55VA91 supports the function of Power Up Sequence, that is, after system power on the W55VA91 SDRAM Controller automatically executes the commands needed for Power Up Sequence and set the mode register of each bank to default value. The default value is :

- Burst Length = 1
- Burst Type = Sequential (fixed)
- CAS Latency = 2
- Write Burst Length = Burst (fixed)

The value of mode register can be changed after power up sequence by setting the value of corresponding bank's configuration register "**LATENCY**" bits and set the **MRSET** bit enable to execute the Mode Register Set command.

4.3.2.4 SDRAM Interface

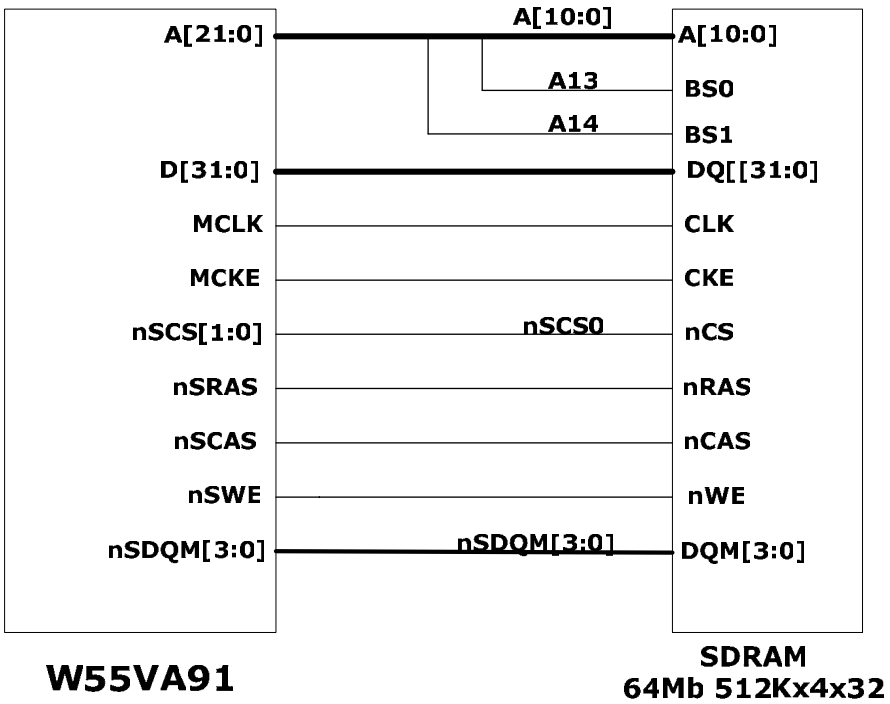


Figure 4.3-1 SDRAM Interface Diagram

4.3.3 EBI Control Register

Register	Address	R/W	Description	Reset Value
EBI_BA = 0xFFF0_1000				
EBICON	EBI_BA + 0x000	R/W	EBI control register	0x00X1_0001
ROMCON	EBI_BA + 0x004	R/W	ROM/FLASH control register	0x0000_0FFX
ROMCON1	EBI_BA + 0x008	R/W	ROM/FLASH control register	0xFFC0_0000
ROMCON2	EBI_BA + 0x00C	R/W	ROM/FLASH control register	0xFFE0_0000
SDCONF0	EBI_BA + 0x010	R/W	SDRAM bank 0 configuration register	0x0000_0800
SDCONF1	EBI_BA + 0x014	R/W	SDRAM bank 1 configuration register	0x0000_0800
SDTIME0	EBI_BA + 0x018	R/W	SDRAM bank 0 timing control register	0x0000_0000
SDTIME1	EBI_BA + 0x01C	R/W	SDRAM bank 1 timing control register	0x0000_0000
EXT0CON	EBI_BA + 0x020	R/W	External I/O 0 control register	0x0000_0000
EXT1CON	EBI_BA + 0x024	R/W	External I/O 1 control register	0x0000_0000
CKSKEW	EBI_BA + 0x028	R/W	Clock skew control register (for testing)	0xXXXX_0038
ICEMEM	EBI_BA + 0x02C	R/W	Memory Configuration especially for ICE	0x0000_0000
EXT2CON	EBI_BA + 0x030	R/W	External I/O 2 control register	0x0000_0000
EBIPDCON	EBI_BA + 0x080	R/W	EBI power down control register	0x0000_0000
EPIPDOUT0	EBI_BA + 0x084	R/W	EBI power down output register 0	0x0000_0000
EPIPDOUT1	EBI_BA + 0x088	R/W	EBI power down output register 1	0x0000_0000
EBIPDTRI0	EBI_BA + 0x08C	R/W	EBI power down tri-state control register 0	0x0000_0000
EBIPDTRI1	EBI_BA + 0x090	R/W	EBI power down tri-state control register 1	0x0000_0000

Table 4.3-5 EBI Control Register Address Map

EBI Control Register (EBICON)

Register	Address	R/W	Description	Reset Value
EBICON	EBI_BA + 0x000	R/W	EBI control register	0x00X1_0001

31	30	29	28	27	26	25	24
RESERVED			EXIOH16	RESERVED	EXBE2	EXBE1	EXBE0
23	22	21	20	19	18	17	16
RESERVED	ROME2	ROME1	ROME0	RD_APE	REFEN	REFMOD	CLKEN
15	14	13	12	11	10	9	8
REFRAT							
7	6	5	4	3	2	1	0
REFRAT					WAITVT		LITTLE

Bits	Descriptions		Default
[28]	**EXIOH16**	**External IO Data Bus at High 16 Bits** 1 = External IO data bus (or NAND Flash data bus) [15:0] bits are shifted to EBI data bus (MD) [31:16] bits. 0 = External IO data bus bits is shared to MD[31:0] **NOTE:** If enabled, external IO data bus is limited to 16 bits. The corresponding bit width setting in EXT0[/1/2]CON can't be 32 bits.	0x0
[26]	**EXBE2**	**External IO bank 2 byte enable** 1 = nWBE[3:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM 0 = nWBE[3:0] pin is byte write strobe signal. **NOTE:** External IO bank2 can't be mapped to a physical chip select. This bank is used when nBTCS, SCS0 or SCS1 is altered as external 1T-SRAM access chip select pin. And this feature should only be used when using 1T-SRAM to support development tool for ICE system.	0x0
[25]	**EXBE1**	**External IO bank 1 byte enable** 1 = nWBE[3:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM 0 = nWBE[3:0] pin is byte write strobe signal **NOTE:** External IO bank1 can't be mapped to a physical chip select by default. But when register PINFUN bit 13 is set, GPA15 is used as External IO bank1 chip selection.	0x0
[24]	**EXBE0**	**External IO bank 0 byte enable** 1 = nWBE[3:0] pin is byte enable signals, nWE will be used as write strobe signal to SRAM 0 = nWBE[3:0] pin is byte write strobe signal	0x0
[22]	**ROME2**	**ROM/FLASH bank 2 enable** 1 = ROM/FLASH bank 2 is enabled. The settings of base address and bank size are effective when this bit is set. And the SDRAM bank 1 chip select is alternated as ROM/FLASH bank 2 chip select. 0 = ROM/FLASH bank 2 is disabled. The corresponding settings and chip select pin won't be effective.	0x0
[21]	**ROME1**	**ROM/FLASH bank 1 enable** 1 = ROM/FLASH bank 1 is enabled. The settings of base address and bank size are effective when this bit is set. And the SDRAM bank 0 chip select is alternated as ROM/FLASH bank 1 chip select. 0 = ROM/FLASH bank 1 is disabled. The corresponding settings and chip select pin won't be effective.	0x0
[20]	**ROME0**	**ROME0 [20] : ROM/FLASH bank 0 enable** 1 = ROM/FLASH bank 0 is enabled. The settings of base address and bank size are effective when this bit is set. When the power-on setting pin MA16 is pull-down, meaning that Boot ROM code is from ROM/FLASH bank 0, this bit will be set to 1 after power-on. 0 = ROM/FLASH bank 0 is disabled. The corresponding settings and chip select pin won't be effective. When the power-on setting pin MA16 is pull-up, meaning that Boot ROM code is from internal Boot ROM, this bit will be set to 0 after power-on.	0xX

<table>
<tr><td>[19]</td><td>RD_APE</td><td>ROM/Flash Address pipeline mode enable in read access
RD_APE decides the issuing timing of an address for following sequential read access.
1 = The following read access address is issued at the latch clock for the previous read data.
0 = The following read access address is issued one clock after the latch clock for the previous read data.
NOTE: the clock referred is the main clock on the system bus.</td><td>0x0</td></tr>
<tr><td>[18]</td><td>REFEN</td><td>Enable SDRAM refresh cycle for SDRAM bank0 & bank1
Setting this bit will start the auto-refresh cycle to SDRAM. The refresh rate is according to REFRAT bits.</td><td>0x0</td></tr>
<tr><td>[17]</td><td>REFMOD</td><td>The refresh mode of SDRAM for SDRAM bank
Defines the refresh mode type of external SDRAM bank
Software can write this bit “1” to force SDRAM enter self-refresh mode. If any read/write to SDRAM occurs then this bit will be cleared to “0” by hardware automatically and SDRAM will enters auto-refresh mode.
0 = Auto refresh mode
1 = Self refresh mode</td><td>0x0</td></tr>
<tr><td>[16]</td><td>CLKEN</td><td>Clock enable for SDRAM
Enables the SDRAM clock enable (CKE) control signal
0 = Disable (power down mode)
1 = Enable</td><td>0x1</td></tr>
<tr><td>[15:3]</td><td>REFRAT</td><td>Refresh count value for SDRAM
The refresh period is calculated as $period = \frac{value}{fMCLK}$
The SDRAM Controller automatically provides an auto refresh cycle for every refresh period programmed into the REFRAT bits when the REFEN bit of each bank is set.</td><td>0x0</td></tr>
<tr><td>[2:1]</td><td>WAITVT</td><td>Valid time of nWAIT signal
W55VA91 recognizes the nEWAIT signal at the next “nth” MCLK rising edge after the nOE or nWBE active cycle. WAITVT bits determine the n.
<table>
<tr><th colspan="2">WAITVT [2:1]</th><th>nth MCLK</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table></td><td>0x0</td></tr>
<tr><td>[0]</td><td>LITTLE</td><td>Read only, Little Endian mode
0 = EBI memory format is Big Endian mode
1 = EBI memory format is Little Endian mode
This bit is left here only for compatible reason and should be fixed to 1.</td><td>0x1</td></tr>
</table>

ROM/Flash Control Register (ROMCON0)

Register	Address	R/W	Description	Reset Value

ROMCON0	EBI_BA + 0x004	R/W	ROM/FLASH control register of Bank 0	0x0000_0FFX

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

Bits	Descriptions		Default
[31:19]	**BASADDR**	**Base address pointer of ROM/Flash bank 0** The start address is calculated as ROM/Flash bank base pointer << 18. The base address pointer together with the "**SIZE**" bits constitutes the whole address range of this bank.	0x0
[18:16]	**SIZE**	**The size of ROM/FLASH memory of bank 0** SIZE [10:8] / Byte 0 0 0 — 256K 0 0 1 — 512K 0 1 0 — 1M 0 1 1 — 2M 1 0 0 — 4M 1 0 1 — 8M 1 1 0 — 16M 1 1 1 — Reserved	0x0
[15:12]	**tWACC**	**Write mode access cycle time** tWACC[11:8] / MCLK / tWACC[11:8] / MCLK 0 0 0 0 — 32 — 1 0 0 0 — 16 0 0 0 1 — 6 — 1 0 0 1 — 18 0 0 1 0 — 6 — 1 0 1 0 — 20 0 0 1 1 — 6 — 1 0 1 1 — 22 0 1 0 0 — 8 — 1 1 0 0 — 24 0 1 0 1 — 10 — 1 1 0 1 — 26 0 1 1 0 — 12 — 1 1 1 0 — 28 0 1 1 1 — 14 — 1 1 1 1 — 30	0x0

<table>
<tr>
<td>[11:8]</td>
<td>tPA</td>
<td>
Page mode access cycle time
<table>
<tr><th colspan="4">tPA[11:8]</th><th>MCLK</th><th colspan="4">tPA[11:8]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>11</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>3</td><td>1</td><td>0</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>4</td><td>1</td><td>0</td><td>1</td><td>0</td><td>15</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>5</td><td>1</td><td>0</td><td>1</td><td>1</td><td>17</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>6</td><td>1</td><td>1</td><td>0</td><td>0</td><td>19</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>7</td><td>1</td><td>1</td><td>0</td><td>1</td><td>21</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>8</td><td>1</td><td>1</td><td>1</td><td>0</td><td>23</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>9</td><td>1</td><td>1</td><td>1</td><td>1</td><td>25</td></tr>
</table>
NOTE: This page mode access speed-up function can only function normally when PGMODE setting is not zero and EBICON[19]/RD_APE is ‘1’.
</td>
<td>0xF</td>
</tr>
<tr>
<td>[7:4]</td>
<td>tACC</td>
<td>
Access cycle time
<table>
<tr><th colspan="4">tACC[7:4]</th><th>MCLK</th><th colspan="4">tACC[7:4]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>1</td><td>0</td><td>0</td><td>0</td><td>11</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>3</td><td>1</td><td>0</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>4</td><td>1</td><td>0</td><td>1</td><td>0</td><td>15</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>5</td><td>1</td><td>0</td><td>1</td><td>1</td><td>17</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>6</td><td>1</td><td>1</td><td>0</td><td>0</td><td>19</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>7</td><td>1</td><td>1</td><td>0</td><td>1</td><td>21</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>8</td><td>1</td><td>1</td><td>1</td><td>0</td><td>23</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>9</td><td>1</td><td>1</td><td>1</td><td>1</td><td>25</td></tr>
</table>
NOTE: The above numerical mapping is valid when EBICON[19]/RD_APE is ‘1’. When EBICON[19]/RD_APE is ‘0’, the mapping value in MCLK would be the value in the above table adding 1.
</td>
<td>0xF</td>
</tr>
<tr>
<td>[3:2]</td>
<td>BTSIZE</td>
<td>
boot ROM/FLASH data bus width
This ROM/Flash bank is designed for a boot ROM. BASADDR bits determine its start address. The external data bus width is determined by the data bus signals D [13:12] power-on setting.
<table>
<tr><th colspan="2">BTSIZE [3:2]</th><th>Bus Width</th></tr>
<tr><td>0</td><td>0</td><td>8-bit</td></tr>
<tr><td>0</td><td>1</td><td>16-bit</td></tr>
<tr><td>1</td><td>0</td><td>32-bit</td></tr>
<tr><td>1</td><td>1</td><td>RESERVED</td></tr>
</table>
<table>
<tr><th colspan="2">D [13:12]</th><th>Bus Width</th></tr>
<tr><td>Pull-down</td><td>Pull-down</td><td>8-bit</td></tr>
<tr><td>Pull-down</td><td>Pull-up</td><td>16-bit</td></tr>
<tr><td>Pull-up</td><td>Pull-down</td><td>32-bit</td></tr>
<tr><td>Pull-up</td><td>Pull-up</td><td>RESERVED</td></tr>
</table>
</td>
<td>0xX</td>
</tr>
<tr>
<td>[1:0]</td>
<td>PGMODE</td>
<td>
Page mode configuration
<table>
<tr><th colspan="2">PGMODE [1:0]</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>Normal ROM</td></tr>
<tr><td>0</td><td>1</td><td>4 word page</td></tr>
<tr><td>1</td><td>0</td><td>8 word page</td></tr>
<tr><td>1</td><td>1</td><td>16 word page</td></tr>
</table>
</td>
<td>0x0</td>
</tr>
</table>

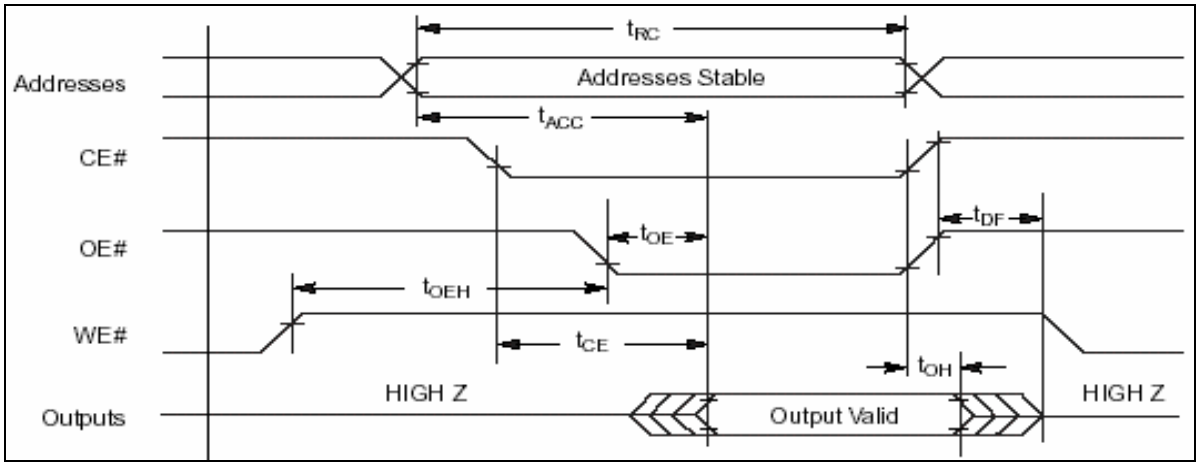


Figure 4.3-2 ROM/FLASH Read Operation Timing

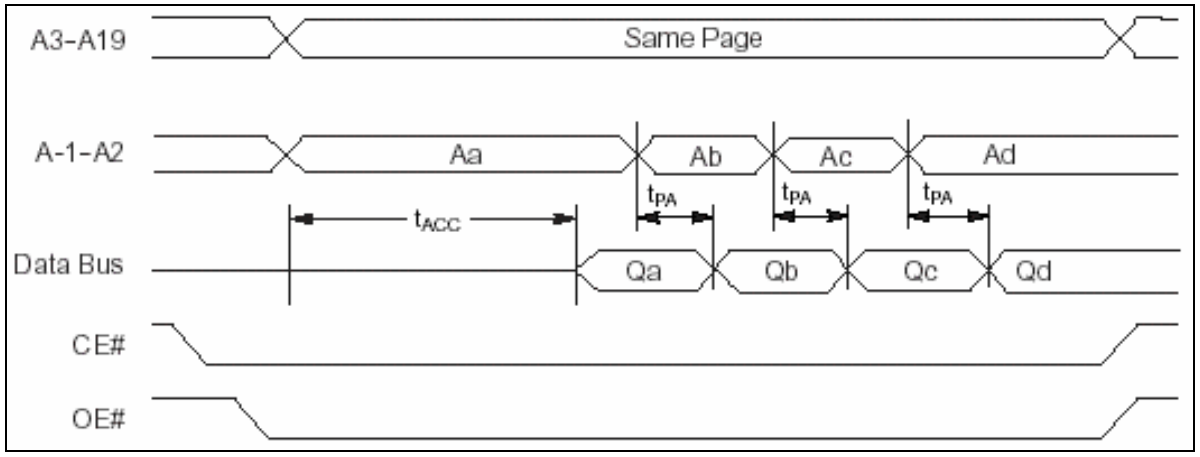


Figure 4.3-3 ROM/FLASH Page Read Operation Timing

ROM/Flash Control Register (ROMCON1/2)

Register	Address	R/W	Description	Reset Value
ROMCON1	EBI_BA + 0x008	R/W	ROM/FLASH control register of bank 1	0xFFC0_0000
ROMCON2	EBI_BA + 0x00C	R/W	ROM/FLASH control register of bank 2	0xFFE0_0000

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

Bits	Descriptions	Default

<table>
<tr><td>[31:19]</td><td>BASADDR</td><td>Base address pointer of ROM/Flash bank of bank 1/2
The start address is calculated as ROM/Flash bank base pointer << 18. The base address pointer together with the “SIZE” bits constitutes the whole address range of each bank.</td><td>0x0</td></tr>
<tr><td>[18:16]</td><td>SIZE</td><td>The size of ROM/FLASH memory of bank 1/2
<table>
<tr><th colspan="3">SIZE [10:8]</th><th>Byte</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>256K</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>512K</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1M</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2M</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4M</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>8M</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>16M</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
NOTE: The settings of the page mode access time, access time, data-port bus bandwidth and page mode enabling for bank 1 and bank 2 follow those settings configured in ROM/Flash bank 0. Bank 1 and bank 2 don’t have individual settings. A W55VA91 user should use the same type of ROM memory devices to connect to these ROM/Flash bank select pins.</td><td>0x0</td></tr>
</table>

Configuration Registers(SDCONF0/1)

The configuration registers enable software to set a number of operating parameters for the SDRAM controller. There are two configuration registers SDCONF0, SDCONF1 for SDRAM bank 0, bank 1 respectively. Each bank can have a different configuration.

Register	Address	R/W	Description	Reset Value
SDCONF0	EBI_BA + 0x010	R/W	SDRAM bank 0 configuration register	0x0000_0800
SDCONF1	EBI_BA + 0x014	R/W	SDRAM bank 1 configuration register	0x0000_0800

31	30	29	28	27	26	25	24
BASADDR							
23	22	21	20	19	18	17	16
BASADDR					RESERVED		
15	14	13	12	11	10	9	8
MRSET	RESERVED	AUTOPR	LATENCY		RESERVED		
7	6	5	4	3	2	1	0
COMPBK	DBWD		COLUMN		SIZE		

Bits	Descriptions	Default

<table>
<tr><td>[31:19]</td><td>BASADDR</td><td>Base address pointer of SDRAM bank 0 / 1
The start address is calculated as SDRAM bank 0/1 base pointer << 18. The SDRAM base address pointer together with the “SIZE” bits constitutes the whole address range of each SDRAM bank.</td><td>0x0</td></tr>
<tr><td>[15]</td><td>MRSET</td><td>SDRAM Mode register set command for SDRAM bank 0 / 1
Setting this bit will issue a mode register set command to SDRAM.</td><td>0x0</td></tr>
<tr><td>[13]</td><td>AUTOPR</td><td>Auto pre-charge mode of SDRAM for SDRAM bank 0 / 1
Enable the auto pre-charge function of external SDRAM bank 0/1
0 = No auto pre-charge, controller issue pre-charge command explicitly.
1 = Auto pre-charge, controller enable auto pre-charge in conjunction with a specific READ or WRITE command.</td><td>0x0</td></tr>
<tr><td>[12:11]</td><td>LATENCY</td><td>The CAS Latency of SDRAM bank 0 / 1
Defines the CAS latency of external SDRAM bank 0/1
<table><tr><th colspan="2">LATENCY [12:11]</th><th>MCLK</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>1</td><td>0</td><td>3</td></tr><tr><td>1</td><td>1</td><td>REVERSED</td></tr></table></td><td>0x1</td></tr>
<tr><td>[7]</td><td>COMPBK</td><td>Number of component bank in SDRAM bank 0 / 1
Indicates the number of component bank (2 or 4 banks) in external SDRAM bank 0/1.
0 = 2 banks
1 = 4 banks</td><td>0x0</td></tr>
<tr><td>[6:5]</td><td>DBWD</td><td>Data bus width for SDRAM bank 0 / 1
Indicates the external data bus width connect with SDRAM bank 0/1
If DBWD = 00, the assigned SDRAM access signal is not generated i.e. disable.
<table><tr><th colspan="2">DBWD [6:5]</th><th>Bits</th></tr><tr><td>0</td><td>0</td><td>Bank disable</td></tr><tr><td>0</td><td>1</td><td>8-bit (byte)</td></tr><tr><td>1</td><td>0</td><td>16-bit (half-word)</td></tr><tr><td>1</td><td>1</td><td>32-bit (word)</td></tr></table></td><td>0x0</td></tr>
<tr><td>[4:3]</td><td>COLUMN</td><td>Number of column address bits in SDRAM bank 0 / 1
Indicates the number of column address bits in external SDRAM bank 0/1.
<table><tr><th colspan="2">COLUMN [4:3]</th><th>Bits</th></tr><tr><td>0</td><td>0</td><td>8</td></tr><tr><td>0</td><td>1</td><td>9</td></tr><tr><td>1</td><td>0</td><td>10</td></tr><tr><td>1</td><td>1</td><td>REVERSED</td></tr></table></td><td>0x0</td></tr>
</table>

<table>
<tr>
<td>[2:0]</td>
<td>SIZE</td>
<td>Size of SDRAM bank 0 / 1
Indicates the memory size of external SDRAM bank 0/1
<table>
<tr><th colspan="3">SIZE [2:0]</th><th>Size of SDRAM (Byte)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Bank disable</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2M</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>4M</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>8M</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>16M</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>32M</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>64M</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>REVERSED</td></tr>
</table>
</td>
<td>0x0</td>
</tr>
</table>

Timing Control Registers (SDTIME0 / 1)

W55VA91 offers the flexible timing control registers to control the generation and processing of the control signals. This allows you to use different speed of SDRAM.

Register	Address	R / W	Description	Reset Value
SDTIME0	EBI_BA + 0x018	R/W	SDRAM bank 0 timing control register	0x0000_0000
SDTIME1	EBI_BA + 0x01C	R/W	SDRAM bank 1 timing 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					tRCD		
7	6	5	4	3	2	1	0
tRDL		tRP			tRAS		

<table>
<tr><th>Bits</th><th>Descriptions</th><th>Default</th></tr>
<tr>
<td>[10:8]</td>
<td>tRCD</td>
<td>SDRAM bank 0 / 1, /RAS to /CAS delay (see Figure 4.3-4)
<table>
<tr><th colspan="3">tRCD [10:8]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>8</td></tr>
</table>
</td>
<td>0x0</td>
</tr>
</table>

<table>
<tr>
<td>[7:6]</td>
<td>tRDL</td>
<td>SDRAM bank 0/1, last data in to pre-charge command (see Figure 4.3-5)
<table>
<tr><th colspan="2">tRDL [7:6]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table></td>
<td>0x0</td>
</tr>
<tr>
<td>[5:3]</td>
<td>tRP</td>
<td>SDRAM bank 0/1, Row pre-charge time (see Figure 4.3-4)
<table>
<tr><th colspan="3">tRP [5:3]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>8</td></tr>
</table></td>
<td>0x0</td>
</tr>
<tr>
<td>[2:0]</td>
<td>tRAS</td>
<td>SDRAM bank 0/1, Row active time (see Figure 4.3-4)
<table>
<tr><th colspan="3">tRAS [2:0]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>7</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>8</td></tr>
</table></td>
<td>0x0</td>
</tr>
</table>

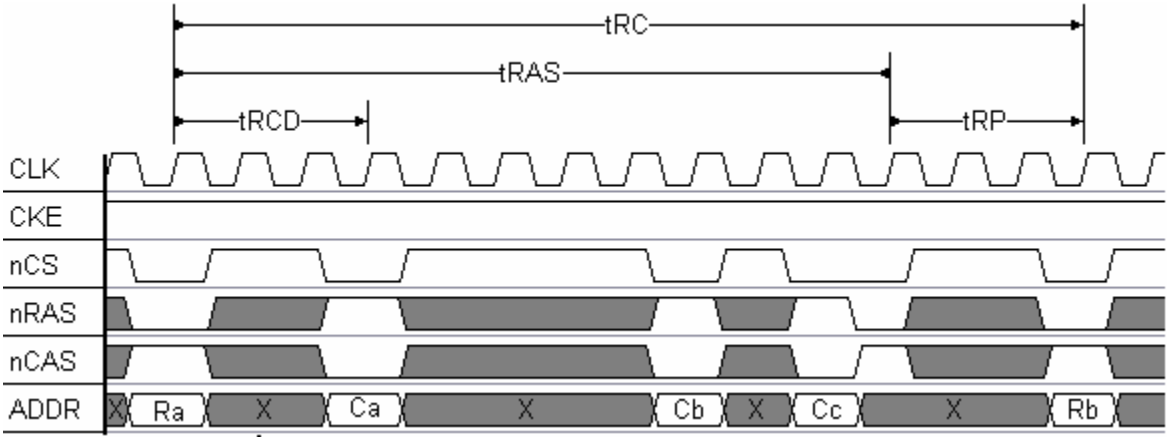


Figure 4.3-4 Access timing 1 of SDRAM

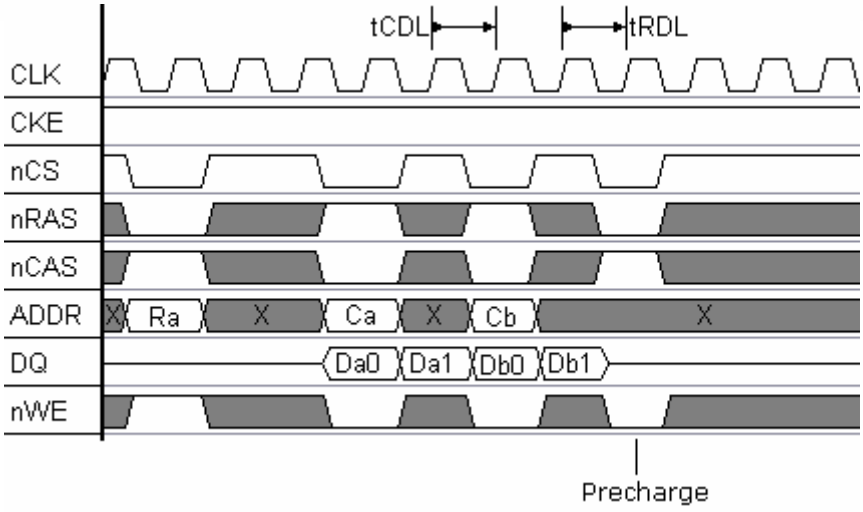


Figure 4.3-5 Access timing 2 of SDRAM

External I/O Control Registers (EXT0CON/EXT1CON/EXT2CON)

The W55VA91 supports an external device control without glue logic. It is very cost effective because address decoding and control signals timing logic are not needed. Using these control registers you can configure special external I/O devices for providing the low cost external devices control solution.

Register	Address	R/W	Description	Reset Value
EXT0CON	EBI_BA + 0x020	R/W	External I/O 0 control register	0x0000_0000
EXT1CON	EBI_BA + 0x024	R/W	External I/O 1 control register	0x0000_0000
EXT2CON	EBI_BA + 0x030	R/W	External I/O 2 control register	0x0000_0000

NOTE: EXT2CON is for an external bank that doesn't have a dedicated chip selection. This bank is multiplexed with nBTCS (ROM/FLASH), nSCS0 and nSCS1 (SDRAM) for read / write access if an asynchronous SRAM (or 1T-SRAM) is connected on these external banks (chip selections).

31	30	29	28	27	26	25	24
BASADDR							
23	22	21	20	19	18	17	16
BASADDR					SIZE		
15	14	13	12	11	10	9	8
ADRS	tACC				tCOH		
7	6	5	4	3	2	1	0
tACS			tCOS			DBWD	

Bits	Descriptions		Default
[31:19]	**BASADDR**	**Base address pointer of external I/O bank 0** The start address of each external I/O bank is calculated as "**BASADDR"** base pointer << 18. Each external I/O bank base address pointer together with the "**SIZE**" bits constitutes the whole address range of each external I/O bank.	0x0

<table>
<tr>
<td>[18:16]</td>
<td>SIZE</td>
<td>The size of the external I/O bank 0
<table>
<tr><th colspan="3">SIZE [18:16]</th><th>Byte</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>256K</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>512K</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1M</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2M</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4M</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>8M</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>16M</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Reserved</td></tr>
</table></td>
<td>0x0</td>
</tr>
<tr>
<td>[15]</td>
<td>ADRS</td>
<td>Address bus alignment for external I/O bank 0
When ADRS is set, EBI bus is alignment to byte address format, and ignores DBWD [1:0] setting.</td>
<td>0x0</td>
</tr>
<tr>
<td>[14:11]</td>
<td>tACC</td>
<td>Access cycles (nOE or nWBE active time)for external I/O bank 0
<table>
<tr><th colspan="4">tACC[14:11]</th><th>MCLK</th><th colspan="4">tACC[14:11]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Reserved</td><td>1</td><td>0</td><td>0</td><td>0</td><td>10</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td><td>1</td><td>0</td><td>0</td><td>1</td><td>12</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>3</td><td>1</td><td>0</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>4</td><td>1</td><td>0</td><td>1</td><td>1</td><td>16</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>5</td><td>1</td><td>1</td><td>0</td><td>0</td><td>18</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>6</td><td>1</td><td>1</td><td>0</td><td>1</td><td>20</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>7</td><td>1</td><td>1</td><td>1</td><td>0</td><td>22</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>8</td><td>1</td><td>1</td><td>1</td><td>1</td><td>24</td></tr>
</table></td>
<td>0x0</td>
</tr>
<tr>
<td>[10:8]</td>
<td>tCOH</td>
<td>Chip selection hold-on time on nOE or nWBE for external I/O bank 0
<table>
<tr><th colspan="3">tCOH [10:8]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table></td>
<td>0x0</td>
</tr>
<tr>
<td>[7:5]</td>
<td>tACS</td>
<td>Address set-up before nECS for external I/O bank 0
<table>
<tr><th colspan="3">tACS [7:5]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table></td>
<td>0x0</td>
</tr>
</table>

<table>
<tr>
<td>[4:2]</td>
<td>tCOS</td>
<td>Chip selection set-up time on nOE or nWBE for external I/O bank 0
When ROM/Flash memory bank is configured, the access to its bank stretches chip selection time before the nOE or new signal is activated.
<table>
<tr><th colspan="3">tCOS [4:2]</th><th>MCLK</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table>
</td>
<td>0x0</td>
</tr>
<tr>
<td>[1:0]</td>
<td>DBWD</td>
<td>Programmable data bus width for external I/O bank 0
<table>
<tr><th colspan="2">DBWD [1:0]</th><th>Width of Data Bus</th></tr>
<tr><td>0</td><td>0</td><td>Disable bus</td></tr>
<tr><td>0</td><td>1</td><td>8-bit</td></tr>
<tr><td>1</td><td>0</td><td>16-bit</td></tr>
<tr><td>1</td><td>1</td><td>32-bit</td></tr>
</table>
</td>
<td>0x0</td>
</tr>
</table>

tACS [7:5] ： tCOS [4:2]： DBWD [1:0] ：

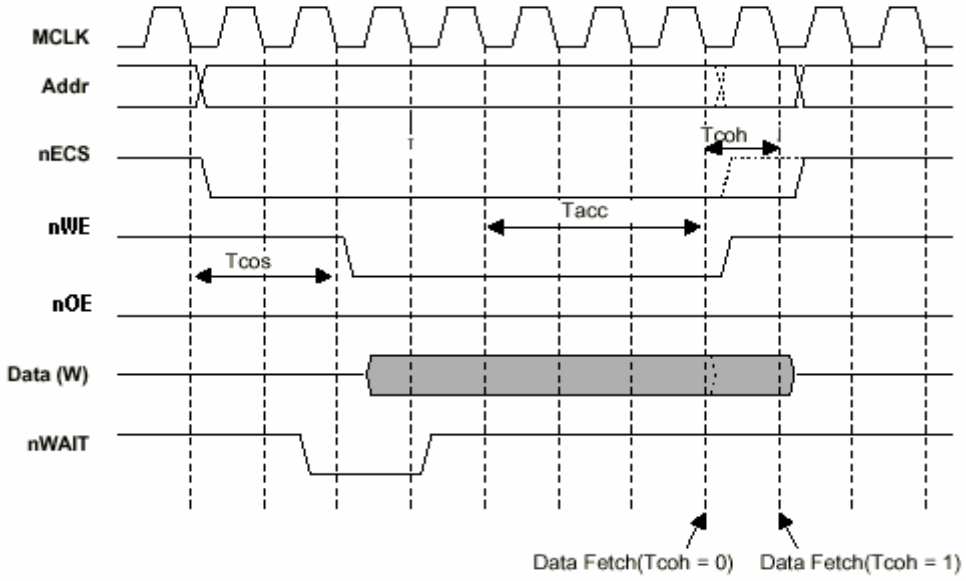


Figure 4.3-6 External I/O write operation timing

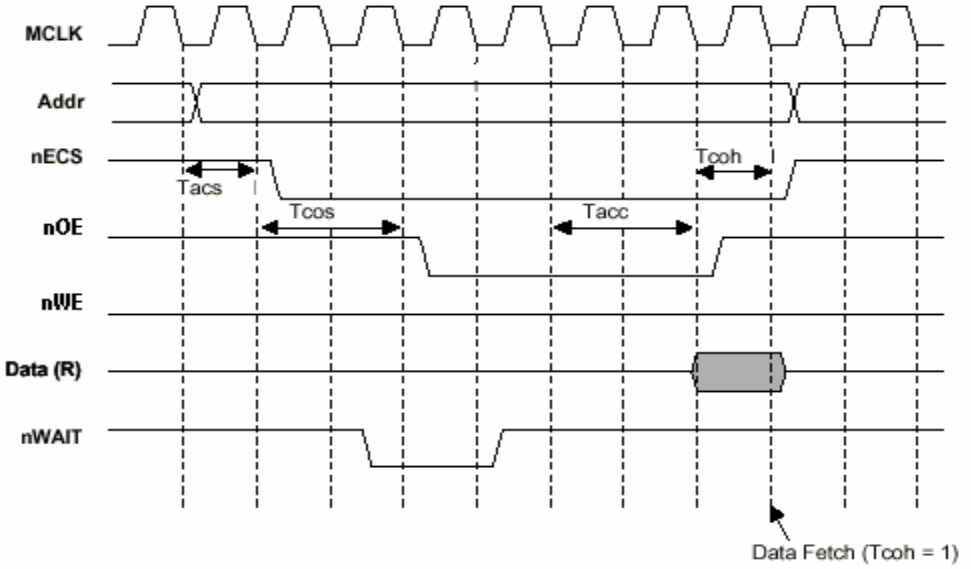


Figure 4.3-7 External I/O read operation timing

Clock Skew Control Register (CKSKEW)

Register	Address	R/W	Description	Reset Value
CKSKEW	EBI_BA + 0x28	R/W	Clock skew control register	0xXXXX_0038

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

Bits	Descriptions		Default
[31:16]	**DLH_CLK_REF**	**Latch DLH_CLK clock tree by HCLK positive edge. (Read Only)**	0xXXXX
[8]	**SWPON**	**SDRAM Initialization by Software trigger** Set this bit will issue a SDRAM power-on default setting command, this bit will be auto-clear by hardware.	0x0

<table>
<tr>
<td>[7:4]</td>
<td>DLH_CLK_SKEW</td>
<td>Data latch clock skew adjustment
<table>
<tr><th colspan="4">DLH_CLK_SKEW [7:4]</th><th>Gate Delay</th><th colspan="4">DLH_CLK_SKEW [7:4]</th><th>Gate Delay</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>P-0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>N-0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>P-1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>N-1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>P-2</td><td>1</td><td>0</td><td>1</td><td>0</td><td>N-2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>P-3</td><td>1</td><td>0</td><td>1</td><td>1</td><td>N-3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>P-4</td><td>1</td><td>1</td><td>0</td><td>0</td><td>N-4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>P-5</td><td>1</td><td>1</td><td>0</td><td>1</td><td>N-5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>P-6</td><td>1</td><td>1</td><td>1</td><td>0</td><td>N-6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>P-7</td><td>1</td><td>1</td><td>1</td><td>1</td><td>N-7</td></tr>
</table>
Note: P-x means Data latched Clock shift “X” gates delays by refer MCLKO positive edge.
Note: N-x means Data latched Clock shift “X” gates delays by refer MCLKO negative edge.</td>
<td>0x3</td>
</tr>
<tr>
<td>[3:0]</td>
<td>MCLK_O_D</td>
<td>MCLK output delay adjustment
<table>
<tr><th colspan="4">MCLK_O_D [3:0]</th><th>Gate Delay</th><th colspan="4">MCLK_O_D [3:0]</th><th>Gate Delay</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>P-0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>N-0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>P-1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>N-1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>P-2</td><td>1</td><td>0</td><td>1</td><td>0</td><td>N-2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>P-3</td><td>1</td><td>0</td><td>1</td><td>1</td><td>N-3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>P-4</td><td>1</td><td>1</td><td>0</td><td>0</td><td>N-4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>P-5</td><td>1</td><td>1</td><td>0</td><td>1</td><td>N-5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>P-6</td><td>1</td><td>1</td><td>1</td><td>0</td><td>N-6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>P-7</td><td>1</td><td>1</td><td>1</td><td>1</td><td>N-7</td></tr>
</table>
Note: P-x means MCLKO shift “X” gates delay by refer HCLK positive edge.
Note: N-x means MCLKO shift “X” gates delay by refer HCLK negative edge.
Note: MCLK is the output pin of MCLKO, which is an internal signal on chip.</td>
<td>0x8</td>
</tr>
</table>

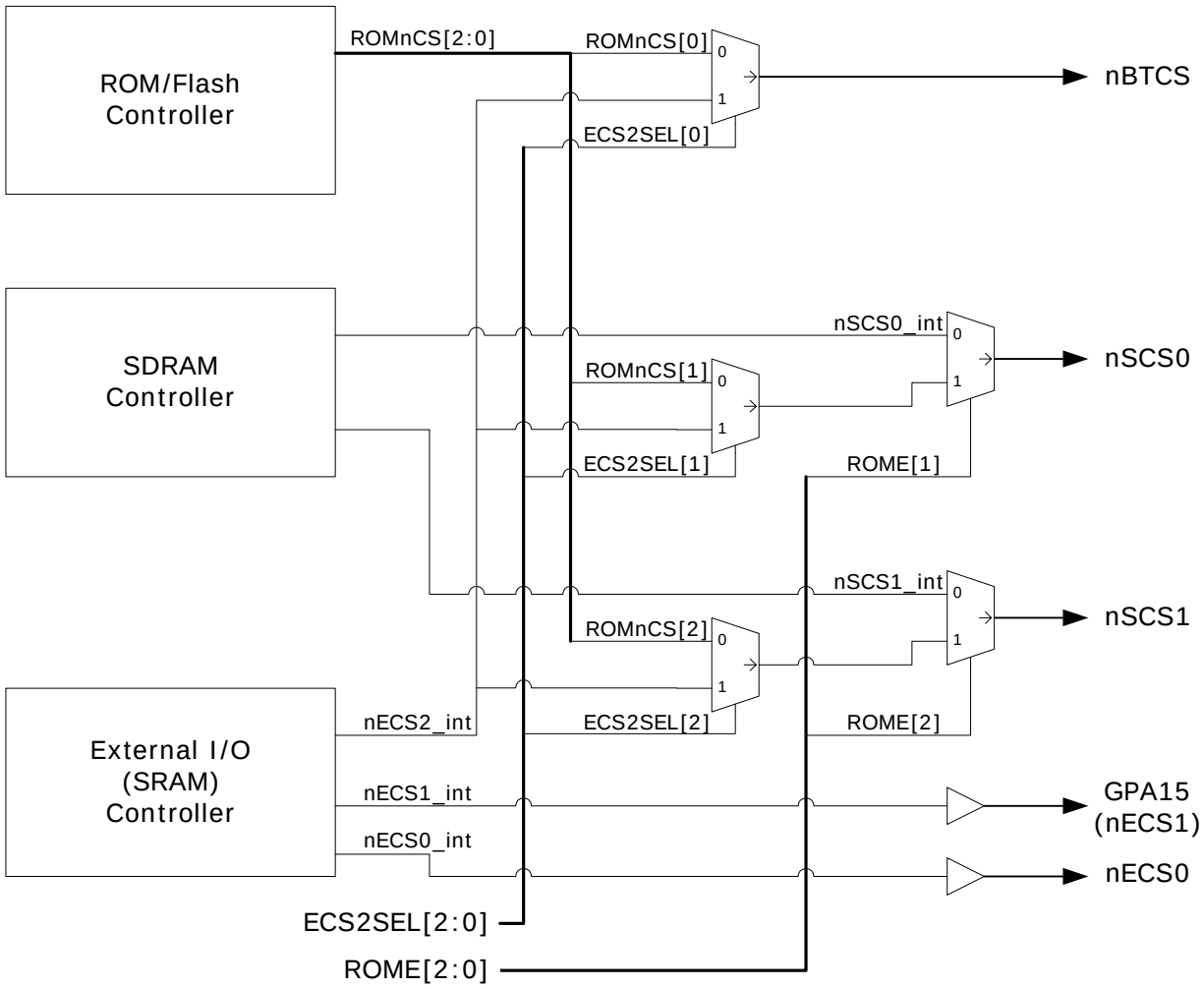
Memory Configuration Especially for ICE (ICEMEM)

Register	Address	R/W	Description	Reset Value
ICEMEM	EBI_BA + 0x02C	R/W	ROM/FLASH control register of Bank 0	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
RESVERED					ECS2SEL		

7	6	5	4	3	2	1	0
RESERVED						ROMBOE	FWRERREN

Bits	Descriptions		Default
[10:8]	**ECS2SEL**	**Enable to select External I/O generated chip selection as chip select output** 1 = Original chip select output is masked; nECS2 is the chip select output. 0 = Original chip select output is the chip select output. Bit 8 = 1, nECS2 appears at nBTCS Bit 9 = 1, nECS2 appears at nSCS0 Bit 10 = 1, nECS2 appears at nSCS1 **NOTE:** The chip select output relationships with control register settings are illustrated in Figure 4.3-8.	0x0
[1]	**ROMBOE**	**ROM/Flash Output Enable together with Byte Strobe Enable** 1 = Output enable asserts together with Byte Strobe (nWBE_SDQM) 0 = Output enable asserts without Byte Strobe asserting **NOTE1:** In normal application case, this bit doesn't need to be enabled. The setting is especially for the 1T-SRAM solution for ICE system. **NOTE2:** A waveform is illustrated to show the results when this bit is disabled or enabled (Figure 4.3-9).	0x0
[0]	**FWRERREN**	**Enable to Issue Bus Error If ROM/Flash Bank is Written** 1 = Issue Bus Error if a AMBA master writing ROM/Flash banks 0 = Accept write if a AMBA master writing ROM/Flash banks **NOTE:** Setting this bit to fit the emulated conditions when external Mask ROM is connected. And remind software developer not to write data to external ROM banks.	0x0



NOTE1: ROME[2:0] is the register control bits at EBICON, bit 22 ~ bit 20.
NOTE2: ROMnCS[2:0] are the internal signals, generated by ROM/Flash controller, for chip selections to external ROM or Flash devices.
NOTE3: nSCS0(1)_int are the internal signals, generated by SDRAM control, for chip selections to external SDRAM devices.
NOTE4: nECS0(/1/2)_int are the internal signals, generated by External I/O controller, for chip selections to external devices with asynchronous read/write IO ports.

Figure 4.3-8 External Memory Chip Selection Architecture and Related Register Settings

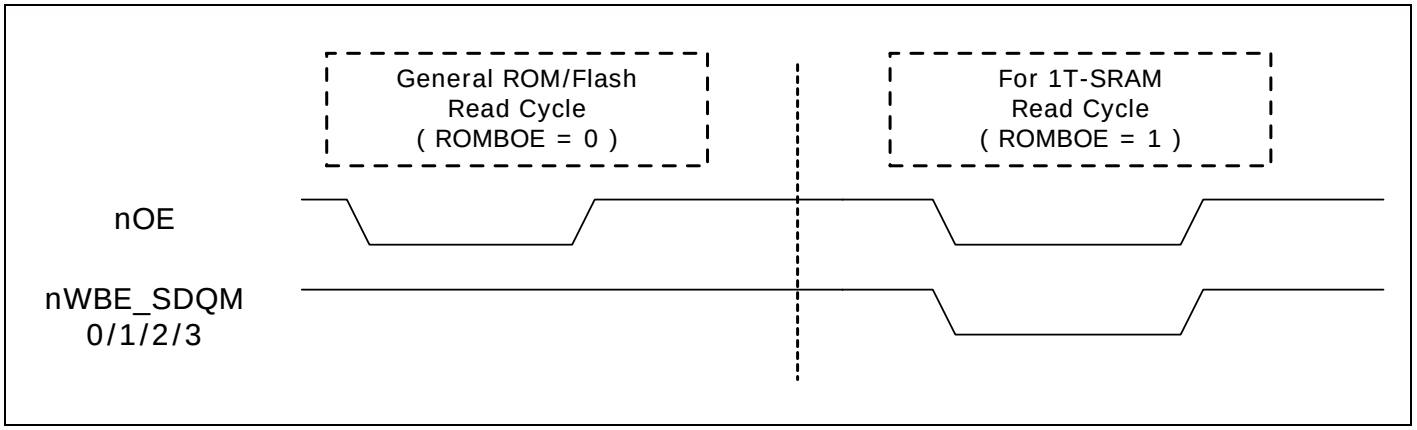


Figure 4.3-9 ROMBOE Setting Effects on nOE and nWBE_SDQM Signals

EBI Power Down Control Register (EBIPDCON)

Register	Address	R/W	Description	Reset Value
EBIPDCON	EBI_BA + 0x080	R/W	EBI power 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
RESERVED							
7	6	5	4	3	2	1	0
RESERVED						FEN	PDFEN

Bits	Descriptions		Default
[1]	**FEN**	**EBI Force Enable to specified value** 0 = Disable 1 = Enable	0x0
[0]	**PDFEN**	**EBI Force Enable to specified value when Power Down** 0 = Disable 1 = Enable	0x0

EBI Power Down Output Register 0 (EBIPDOUT0)

Register	Address	R/W	Description	Reset Value
EBIPDOUT0	EBI_BA + 0x084	R/W	EBI power down output register 0	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							MD_PDV

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

Bits	Descriptions		Default
[31:25]	RESERVED	RESERVED	0x0
[24]	**MD_PDV**	**EBI Data Bus Power Down Pad Output** 1 = All MD bits are High 0 = All MD bits are Low	0x0
[23]	RESERVED	RESERVED	0x0
[22:0]	**MA_PDV**	**EBI Address Bus Power Down Pad Output** MA_PDV[x] = 1, MA[x] is High MA_PDV[x] = 0, MA[x] is Low	0x0

EBI Power Down Output Register 1 (EBIPDOUT1)

Register	Address	R/W	Description	Reset Value
EBIPDOUT1	EBI_BA + 0x088	R/W	EBI power down output register 1	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						nSCS_PDV[1:0]	
15	14	13	12	11	10	9	8
nOE_PDV	RESERVED		nECS_PDV[1:0]		RESERVED		nBTCS_PDV
7	6	5	4	3	2	1	0
nWBE_SDQM_PDV[3:0]				MCKE_PDV	nWE_PDV	nSRAS_PDV	nSCAS_PDV

Bits	Descriptions		Default
[31:18]	RESERVED	RESERVED	0x0
[17:16]	**nSCS_PDV[1:0]**	**nSCS[1:0] Pad Output** 1 = High 0 = Low	0x0
[15]	**nOE_PDV**	**nOE Pad Output** 1 = High 0 = Low	0x0
[14:13]	RESERVED	RESERVED	0x0

[12:11]	**nECS_PDV[1:0]**	**nECS[1:0] Pad Output** 1 = High 0 = Low	0x0
[10:9]	RESERVED	RESERVED	0x0
[8]	**nBTCS_PDV**	**nBTCS Pad Output** 1 = High 0 = Low	0x0
[7:4]	**nWBE_SDQM_PDV[3:0]**	**nWBE_SDQM[3:0] Pad Output** 1 = High 0 = Low	0x0
[3]	**MCKE_PDV**	**MCKE Pad Output** 1 = High 0 = Low	0x0
[2]	**nWE_PDV**	**nWE Pad Output** 1 = High 0 = Low	0x0
[1]	**nSRAS_PDV**	**nSRAS Pad Output** 1 = High 0 = Low	0x0
[0]	**nSCAS_PDV**	**nSCAS Pad Output** 1 = High 0 = Low	0x0

EBI Power Down Tri-State Control Register 0 (EBIPDTRI0)

Register	Address	R/W	Description	Reset Value
EBIPDTRI0	EBI_BA + 0x08C	R/W	EBI power down tri-state control register 0	0x0000_0000

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

Bits	Descriptions		Default
[31:25]	RESERVED	RESERVED	0x0
[24]	**MDTE**	**EBI Data Bus Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0

[23]	RESERVED	RESERVED	0x0
[22:0]	**MATE**	**EBI Address Bus Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0

EBI Power Down Tri-state Control Register 1 (EBIPDTRI1)

Register	Address	R/W	Description	Reset Value
EBIPDTRI1	EBI_BA + 0x090	R/W	EBI power down tri-state control register 1	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED						nSCSTE[1:0]	
15	14	13	12	11	10	9	8
nOETE	RESERVED		nECSTE[1:0]		RESERVED		nBTCSTE
7	6	5	4	3	2	1	0
nWBE_SDQMTE[3:0]				MCKETE	nWETE	nSRASTE	nSCASTE

Bits	Descriptions		Default
[31:18]	RESERVED	RESERVED	0x0
[17:16]	**nSCSTE[1:0]**	**nSCS[1:0] Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0
[15]	**nOETE**	**nOE Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0
[14:13]	RESERVED	RESERVED	0x0
[12:11]	**nECSTE[1:0]**	**nECS[1:0] Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0
[10:9]	RESERVED	RESERVED	0x0
[8]	**nBTCSTE**	**nBTCS Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0
[7:4]	**nWBE_SDQMTE[3:0]**	**nWBE_SDQM[3:0] Tri-state Enable** 1 = High 0 = Low	0x0
[3]	**nMCKETE**	**nMCKE Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0

[2]	**nWETE**	**nWE Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0
[1]	**nSRASTE**	**nSRAS Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0
[0]	**nSCASTE**	**nSCAS Pad Tri-state Enable** 1 = Enable 0 = Disable	0x0

4.4 Video / Display Service SRAM System (VRAM)

4.4.1 Overview and Features

The Video (Display) service SRAM system contains several synchronous SRAM cells, control access circuits and test circuits to satisfy the following functions:

- Serve as display memory to store graphic-related data for graphics constructions
- Serve as VPOST (display circuits) display buffers (line buffers)
- Slave interface to support ARM9 core to store data or load data via these SRAMs
- Program temporary memory

The main features are:

- Total 40KB Display RAM (VRAM) memory in 5 blocks of 8KB SRAM cells, by program setting to define the usage as following
  - PRAM (or WRAM) mode: mainly for program data usage
  - VRAM mode: mainly for display graphics constructions by GPU
- 2KB SAT RAM to store Sprite Attribute Table, which is referenced by GPU sprite related function
- 512 Bytes Sprite Color Pallet RAM to store color pallet used as color table for GPU sprite related function
- 4*512 Bytes Background Color Pallet RAM to store color pallet used as color table for GPU 4 backgrounds related function.
- 640x20 bits two-port register file SRAM
  - Total 3 units
  - Used as line buffers to store GPU constructed graphic image, the stored image is then transferred by VPOST to external display interface (TV or LCD module)
- VRAM loader to frame buffer, including the transparency bit.

4.4.2 Video/Display Service SRAM System Block Diagram

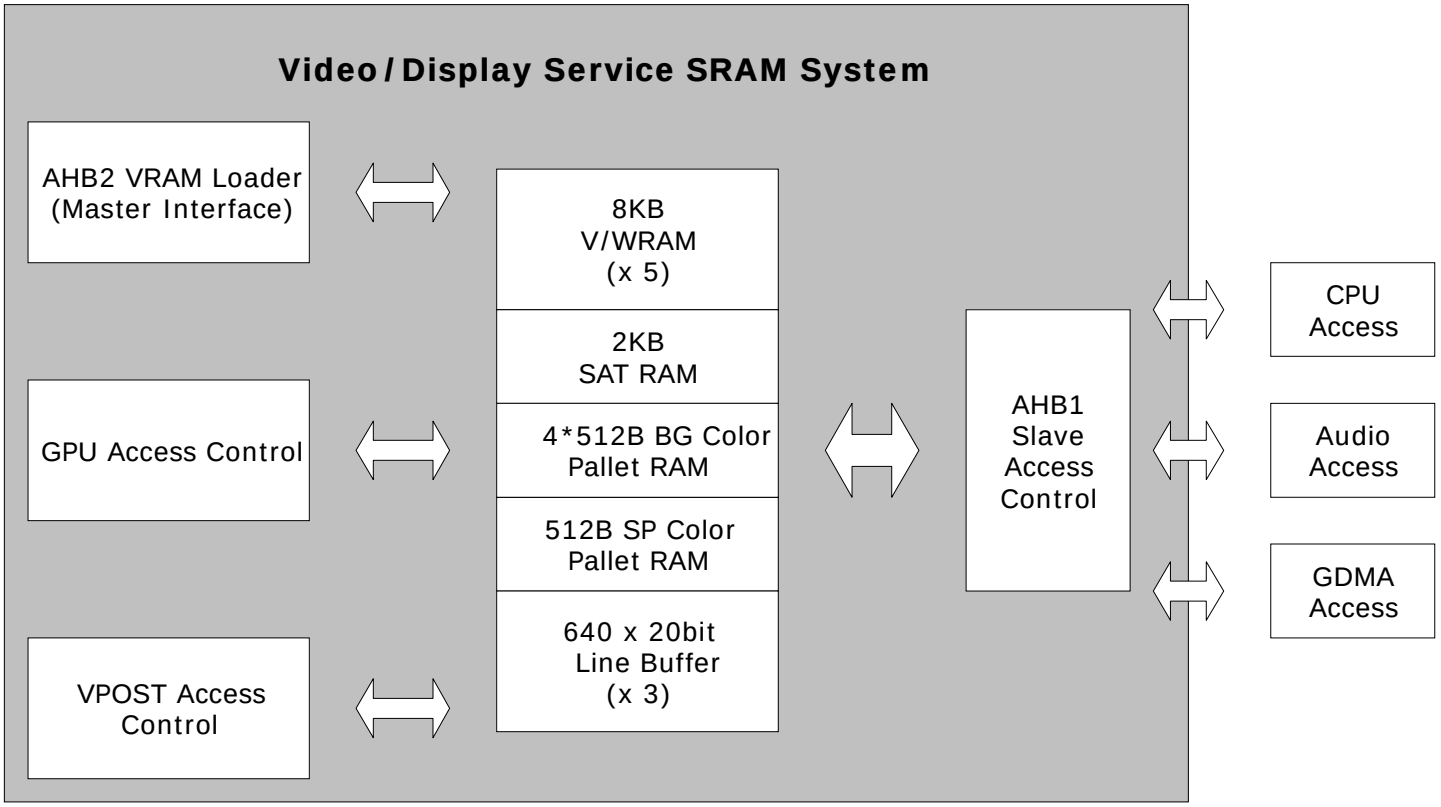


Figure 4.4-1 VRAM System Block Diagram

4.4.3 Control Register Map

Register	Address	R/W	Description	Reset Value
VLD_BA = 0xFFF0_B000				
VRAMCON	VLD_BA+0x000	R/W	Video/Working RAM Configuration	0x0000_0001
VLNBCON	VLD_BA+0x004	R/W	Line Buffer Configuration and Control	0x0001_0001
VSTLBCON	VLD_BA+0x010	R/W	Store Line Buffer Configuration	0x0000_0000
VSTLFMA0	VLD_BA+0x014	R/W	Stored Line Buffer Frame Start Address 0	0xF000_0000
VSTLFMA1	VLD_BA+0x018	R/W	Stored Line Buffer Frame Start Address 1	0xF100_0000
VSTLFOB0	VLD_BA+0x020	R/W	Stored Line Buffer Frame 0 0ffset Base	0x0000_0000
VSTLFOB1	VLD_BA+0x024	R/W	Stored Line Buffer Frame 1 0ffset Base	0x0000_0000
VSTLFPOA	VLD_BA+0x028	R	Stored Line Buffer Frame Progress Offset Address	0x0000_0000
VLDRCON	VLD_BA+0x040	R/W	VRAM Loader Control Register	0x0000_0000
VLDRIEC	VLD_BA+0x044	R/W	VRAM Loader Interrupt Enable Configuration	0x0000_0000

VLDRIST	VLD_BA+0x048	R/W	VRAM Loader Interrupt Raw Status	0x0000_0000
VLDSRBA	VLD_BA+0x050	R/W	VRAM Loader Source Base Address	0xF000_0000
VLDSRPTS	VLD_BA+0x054	R/W	VRAM Loader Source Per Transfer Setting	0x0000_0000
VLDSRTRC	VLD_BA+0x058	R/W	VRAM Loader Source Transfer Count	0x0000_0000
VLDDSBA	VLD_BA+0x060	R/W	VRAM Loader Destination Base Address	0xFFE7_0000
VLDDSPTS	VLD_BA+0x064	R/W	VRAM Loader Destination Per Transfer Setting	0x0000_0000
VLDDSTRC	VLD_BA+0x068	R/W	VRAM Loader Destination Transfer Count	0x0000_0000

4.4.4 Control Registers

Video / Working RAM Configure Control (VRAMCON)

Configure the five 8K-Byte SRAM blocks as working (program) SRAM or as Video/Display service SRAM individually.

Register	Address	R / W	Description	Reset Value
VRAMCON	VLD_BA + 0x000	R/W	Video/Working RAM Configuration	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					VRAMEN2	VRAMEN1	VRAMEN0

Bits	Descriptions		Default
[2]	**VRAMEN2**	**VRAMENx: Working RAM or VRAM** 1 = set as Video/Display service SRAM. GPU will grant the access for this SRAM. CPU access can be done while GPU access is inactive. But CPU access may be fail when GPU access is going. 0 = set as Working (program) SRAM. Only CPU, GDMA and Audio Processor can access it.	0x0
[1]	**VRAMEN1**		0x0
[0]	**VRAMEN0**		0x1
NOTE: By power-on default, 1 blocks of SRAM, totally 8K Bytes, are configured as Video/Display Service SRAM.			

The following table shows the memory space affected for each enable bit.

Base Address	Address Space	Description
VRAMEN0	0xFFE7_6000 ~ 0xFFE7_7FFF	Working/Video RAM Bank0 (8K Bytes)
VRAMEN1	0xFFE7_8000 ~ 0xFFE7_9FFF	Working/Video RAM Bank1 (8K Bytes)
VRAMEN2	0xFFE7_A000 ~ 0xFFE7_BFFF	Working/Video RAM Bank2 (8K Bytes)

Table 4.4-1 VRAM address space and enable bits VRAMENx

Line Buffer Configuration and Control (VLBNCON)

Configure the Display line buffer and provide the status for monitoring.

Register	Address	R/W	Description	Reset Value
VLBNCON	VLD_BA + 0x004	R/W	Line Buffer configuration and control	0x000X_0001

31	30	29	28	27	26	25	24
RESERVED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED					LBSW2	LBSW1	LBSW0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED							VP_ACEN

Bits	Descriptions		Default
[18]	**LBSW2**	**Line buffer switch status (Read Only)** Read out value explanations	X
[17]	**LBSW1**		X
[16]	**LBSW0**		X
[0]	**VP_ACEN**	**VPOST Access Enable** 1 = VPOST can access the plot-ready line buffer and display the content to external device. 0 = The plot-ready line buffer is not accessed by VPOST. The content may be transferred to a frame buffer for other processing.	0x1

LBSW2/1/0			Description
0	0	1	Line buffer 0 is ready for display or transfer to frame buffer. Line buffer 1 is accessed by GPU background processor. Line buffer 2 is accessed by GPU sprite processor.
0	1	0	Line buffer 1 is ready for display or transfer to frame buffer. Line buffer 2 is accessed by GPU background processor. Line buffer 0 is accessed by GPU sprite processor.
1	0	0	Line buffer 2 is ready for display or transfer to frame buffer. Line buffer 0 is accessed by GPU background processor. Line buffer 1 is accessed by GPU sprite processor.

Store Line Buffer Configuration (VSTLBCON)

Configure the function to store line buffer contents drawn by GPU to a frame buffer memory (usually an external memory device).

Register	Address	R/W	Description	Reset Value
VSTLBCON	VLD_BA + 0x010	R/W	Configure the Store Line Buffer 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
FIELD_ST[7:0]							
7	6	5	4	3	2	1	0
RESERVED				FRAME_EN	VGA_FEN	DBL_FEN	STLNBEN

Bits	Descriptions		Default
[15:8]	**FIELD_ST[7:0]**	**FIELD [x] is now empty, padding data in progress or ready to display** FIELD_ST[x] = 1, Field x is now ready to display. FIELD_ST[x] = 0, Field x is now empty or line storage is operating to pad data in this frame. Write '1' to the corresponding bit can clear the status to '0'.	0x0
[3]	**FRAME_EN**	**FRAME Mode Enable to Automatically Store Line Buffer** 1 = Detect line counter value change and automatically start line buffer storage operation if line counter value exceeds the start line value. 0 = Start line buffer storage operation only if the software (program) enable the operation and clear the line complete flag. **NOTE:** line counter value and start line value are both referenced from VPOST relative register settings.	0x0
[2]	**VGA_FEN**	**VGA Frame Storage Enable** 1 = Using 4 fields of memory locations to store line buffer contents. When one field is filled readily, the next field is continuously padded till the 3th field padded completely. When DBL_FEN is '1', the other frame (frame 0 or frame 1) is to be stored the line buffer contents. 0 = VGA Frame Storage mode is disabled, store line buffer to only one field of memory location. When one field is filled readily, this 320x240 pixels field is completed to be stored. And when DBL_FEN is '1', the other frame (frame 0 or frame 1) is to be stored the line buffer contents.	0x0
[1]	**DBL_FEN**	**Double Frame Buffer Enable** 1 = Using two-frame mechanism to store line buffer contents. Each frame has its own storage base address (start address). When one frame is stored completely, the other one is the new storage destination for the new coming line buffer data. Frame 0 and frame 1 refer to VSTLFMA0 and VSTLFMA1, respectively, as its base address. 0 = Using only frame 0 of memory location to store line buffer contents. When this frame is filled completely. The new coming data are still stored from the start address of this frame and old data is soon updated. Only VSTLFMA0 is used as the frame start address.	0x0
[0]	**STLNBEN**	**Store Line Buffer Enable** 1 = Enable to store line buffer contents to external frame buffer. 0 = Disable to store line buffer contents.	0x0

Stored Line Buffer Frame Start Address 0 (VSTLFMA0)

Set the base address to store the line buffer contents from this start address for frame 0. Then pad the following line buffer contents line by line till the whole frame graphic stored.

Register	Address	R/W	Description	Reset Value
VSTLFMA0	VLD_BA + 0x014	R/W	Base address to store frame 0 (1 field or 4 fields).	0xF000_0000

31	30	29	28	27	26	25	24
VSTLFMA0[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VSTLFMA0[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VSTLFMA0[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSTLFMA0[7:2]						RESERVED	

Bits	Descriptions		Default
[31:2]	**VSTLFMA0**	**Base address of the frame 0 to store line buffer contents** **NOTE:** This address is word aligned (4 Bytes) to system address because store line buffer to frame buffer transfer data on the basis of word type.	0xF000_0000

Stored Line Buffer Frame Start Address 1 (VSTLFMA1)

Set the base address to store the line buffer contents from this start address for frame 1. Then pad the following line buffer contents line by line till the whole frame graphic stored.

Register	Address	R/W	Description	Reset Value
VSTLFMA1	VLD_BA + 0x018	R/W	Base address to store frame 1 (1 field or 4 fields).	0xF100_0000

31	30	29	28	27	26	25	24
VSTLFMA1[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VSTLFMA1[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VSTLFMA1[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSTLFMA1[7:2]						RESERVED	

Bits	Descriptions	Default

[31:2]	**VSTLFMA1**	**Base address of the frame 1 to store line buffer contents** **NOTE:** This address is word aligned (4 Bytes) to system address because store line buffer to frame buffer transfer data on the basis of word type.	0xF100_0000

NOTE: !!! IMPORTANT!!! Any value changed or modified by software to VSTLFMA0/1 will make the line buffer storage operation to reset the storing address to the base start address of frame 0 (VSTLFMA0).

Stored Line Buffer Frame 0 Offset Base (VSTLFOB0)

Set the base address to store the line buffer contents offset from this start address of frame 0.

Register	Address	R/W	Description	Reset Value
VSTLFOB0	VLD_BA + 0x020	R/W	Offset Base to store frame 0 (1 field or 4 fields).	0x0000_0000

31	30	29	28	27	26	25	24
VSTLOB0[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VSTLFOB0[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VSTLFOB0[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSTLFOB0[7:2]						RESERVED	

Bits	Descriptions		Default
[31:2]	**VSTLFOB0**	**Offset Base of the frame 0 to store line buffer contents** **NOTE:** This address is word aligned (4 Bytes) to system address because store line buffer to frame buffer transfer data on the basis of word type.	0x0000_0000

Stored Line Buffer Frame 1 Offset Base (VSTLFOB1)

Set the base address to store the line buffer contents offset from this start address of frame 0.

Register	Address	R/W	Description	Reset Value
VSTLFOB1	VLD_BA + 0x024	R/W	Offset Base to store frame 1 (1 field or 4 fields).	0x0000_0000

31	30	29	28	27	26	25	24
VSTLOB1[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VSTLFOB1[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VSTLFOB1[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VSTLFOB1[7:2]						RESERVED	

Bits	Descriptions		Default
[31:2]	**VSTLFOB1**	**Offset Base of the frame 1 to store line buffer contents** **NOTE:** This address is word aligned (4 Bytes) to system address because store line buffer to frame buffer transfer data on the basis of word type.	0x0000_0000

Stored Line Buffer Frame Progress Offset Address (VSTLFPOA)

Read out the address to check the progressing address offset from this start address of frame 0 or 1.

Register	Address	R/W	Description	Reset Value
VSTLFPOA	VLD_BA + 0x028	R	Progress offset address of frame 0 or 1 (1 field or 4 fields).	0x0000_0000

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

Bits	Descriptions		Default
[23:0]	**VSTLFPOA**	**Progress Offset address of the frame 0 or 1** **NOTE:** This progress offset address is read only.	0x0000_0000

VRAM Loader Control Register (VLDRCON)

Register	Address	R/W	Description	Reset Value
VLDRCON	VLD_BA + 0x040	R/C	VRAM loader control and configuration.	0x0000_0000

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

Bits	Descriptions	Default

[16]	**VLDCOMP**	**VRAM Loader Transfer Completed** 1 = VRAM Loader transfer process is completed. This bit can only be set by hardware. 0 = Completion of VRAM Loader transfer doesn't occur. **NOTE:** Write 1 to this bit can clear it to 0.	0x0
[8]	**VLDHWT**	**VRAM Loader Half-Word Transfer** 1 = Half-word transfer. 0 = Word transfer.	0x0
[0]	**VLDRUN**	**Enable to Run VRAM Loader Transfer** 1 = Enable to run the transfer. If VLDCOMP=1 and not cleared, set this bit to 1 won't let the transfer be activated. 0 = Disable or halt the VRAM Loader transfer.	0x0

VRAM Loader Interrupt Enable Configuration (VLDRIEC)

Register	Address	R/W	Description	Reset Value
VLDRIEC	VLD_BA + 0x044	R/W	Configuration to enable several interrupt sources individually.	0x0000_0000

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

Bits	Descriptions		Default
[16]	**VLDCE**	**Interrupt Enable to VRAM Loader Transfer Complete** 1 = Enable interrupt when VRAM Loader transfer process is completed. 0 = Disable to assert interrupt when VRAM Loader transfer process is completed.	0x0
[8]	**VSTFCE**	**Interrupt Enable to One Frame Buffer Storage Completed** 1 = Enable interrupt when complete to store one frame or one line buffer with line buffer contents. 0 = Disable to assert interrupt if complete to store one frame or one line buffer. NOTE: FRAME_EN= 0, this bit is to enable the interrupt when one line storage completed event occur. But FRAME_EN= 1, this bit is to enable the interrupt when one frame storage completed event occur.	0x0

VRAM Loader Interrupt Raw Status (VLDRIST)

Register	Address	R/W	Description	Reset Value
VLDRIST	VLD_BA + 0x048	R/C	Configuration to enable several interrupt sources individually.	0x0000_0000

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

Bits	Descriptions		Default
[16]	**VLDC**	**Interrupt Flag to VRAM Loader Transfer Complete or Error** 1 = VRAM Loader transfer process is completed or halted by Error. 0 = VRAM Loader transfer process is not completed and no Error. **NOTE:** Write 1 to this bit can clear it to 0. To stop continuing assertion of interrupt, it's necessary to clear it by software after interrupt event occur.	0x0
[9]	**VSTLNC**	**Interrupt Flag to One Line Buffer Storage Completed** 1 = Complete to store one line buffer to external memory. 0 = Not complete to store one line buffer to external memory. **NOTE:** Write 1 to this bit can clear it to 0. To stop continuing assertion of interrupt, it's necessary to clear it by software after interrupt event occur. **NOTE:** When this bit is set by hardware, and line buffer storage is fully controlled by software (FRAME_EN, VSTLBCON.3=0), this bit should be cleared to start next line buffer storage operation.	0x0
[8]	**VSTFC**	**Interrupt Flag to One Field Buffer Storage Completed** 1 = Complete to store one field buffer to external memory with line buffer contents. 0 = Not complete to store one field buffer to external memory with line buffer contents. **NOTE:** Write 1 to this bit can clear it to 0. To stop continuing assertion of interrupt, it's necessary to clear it by software after interrupt event occur.	0x0

VRAM Loader Source Base Address (VLDSRBA)

Register	Address	R/W	Description	Reset Value
VLDSRBA	VLD_BA + 0x050	R/W	VRAM Loader Source Data Base address	0xF000_0000

31	30	29	28	27	26	25	24
VLDSRBA[30:23]							

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

Bits	Descriptions		Default
[31:1]	**VLDSRBA**	**VRAM Loader Data Source Base Address** VLDSRBA[30:1] is for word transfer VLDSRBA[30:0] is for half-word transfer **NOTE:** This address is word/half-word aligned (4/2 Bytes) to system address because VRAM loader transfer data on the basis of word or half-word type.	0xF000_0000

VRAM Loader Source Per Transfer Setting (VLDSRPTS)

Setting one transfer with continuous word address following with offset address to the next transfer.

Register	Address	R/W	Description	Reset Value
VLDSRPTS	VLD_BA + 0x054	R/W	Source data per transfer setting and offset setting.	0x0000_0000

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

Bits	Descriptions		Default
[31:17]	**VSROS**	**VRAM Loader Data Source Word/Half-Word Offset Between Each Continuous Transfer** VSROS[14:1] is for word transfer VSROS[14:0] is for half-word transfer **NOTE:** This Offset setting is word/half-word aligned (4/2 Bytes) to system address.	0x0
[15:1]	**VSRPTC**	**VRAM Loader Data Source Word/Half-Word Count for Each Continuous Transfer** VSRPTC[14:1] is for word transfer VSRPTC[14:0] is for half-word transfer **NOTE:** The continuous transfer count is set by byte unit counting. But the least significant 1 bit is reserved because the transfer is on a basis of word or half-word type transfer.	0x0

VRAM Loader Source Transfer Count (VLDSRTRC)

Register	Address	R/W	Description	Reset Value
VLDSRTRC	VLD_BA + 0x058	R/W	Set the source data total transfer count and showing transfer status.	0x0000_0000

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

Bits	Descriptions		Default
[17]	**VSRERR**	**VRAM Loader Data Source Error Flag** 1 = There is error occurred when taking data from source. 0 = No error occurred when taking data from source. **NOTE:** Write 1 to this bit can clear it to 0. When this bit is set by hardware, the transfer is halted.	0x0
[16]	**VSRST**	**VRAM Loader Data Source Transfer Status** 1 = VRAM Loader data source transfer is completed or halted by error. 0 = VRAM Loader data source transfer is not completed or halted. **NOTE:** Write 1 to this bit can clear it to 0. This bit must be cleared before next transfer started.	0x0
[13:0]	**VSRTC**	**VRAM Loader Data Source Transfer Count Number of Continuous Transfer**	0x0

VRAM Loader Destination Base Address (VLDDSBA)

Register	Address	R/W	Description	Reset Value
VLDDSBA	VLD_BA + 0x060	R/W	VRAM Loader destination data base address	0xFFE7_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**
VLDDSBA[14:7]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VLDDSBA[6:0]							RESERVED

Bits	Descriptions		Default
[15:1]	**VLDDSBA**	**VRAM Loader Data Destination Base Address** VLDDSBA[14:1] is for word transfer VLDDSBA[14:0] is for half-word transfer **NOTE1:** This address is word/half-word aligned (4/2 Bytes) to system address because VRAM loader transfers data on the basis of word or half-word type. **NOTE2:** The destination of VRAM Loader is only to the internal SRAM, stretching address from 0xFFE7_0000 to 0xFFE7_FFFF. Thus the MSB [31:16] are fixed to 0xFFE7.	0x0

VRAM Loader Destination Per Transfer Setting (VLDDSPTS)

Setting one transfer with continuous word address following with offset address to the next transfer.

Register	Address	R/W	Description	Reset Value
VLDDSPTS	VLD_BA + 0x064	R/W	Per Transfer Setting and Offset Setting.	0x0000_0000

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

Bits	Descriptions		Default
[31:17]	**VDSOS**	**VRAM Loader Data Destination Word Offset Between Each Continuous Transfer** VDSOS[14:1] is for word transfer VDSOS[14:0] is for half-word transfer **NOTE:** This Offset setting is word/half-word aligned (4/2 Bytes) to system address.	0x0
[15:1]	**VDSPTC**	**VRAM Loader Data Destination Word Count for Each Continuous Transfer** VDSPTC[14:1] is for word transfer VDSPTC[14:0] is for half-word transfer **NOTE:** The continuous transfer count is set by byte unit counting. But the least significant 1 bit is reserved because the transfer is on a basis of word or half-word type transfer.	0x0

VRAM Loader Destination Transfer Count (VLDDSTRC)

Register	Address	R/W	Description	Reset Value
VLDDSTRC	VLD_BA + 0x068	R/W	Set the total transfer count and showing transfer status.	0x0000_0000

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

Bits	Descriptions		Default
[17]	**VDSERR**	**VRAM Loader Data Destination Error Flag** 1 = There is error occurred when storing data to destination. 0 = No error occurred when storing data to destination. **NOTE:** Write 1 to this bit can clear it to 0. When this bit is set by hardware, the transfer is halted.	0x0
[16]	**VDSST**	**VRAM Loader Data Destination Transfer Status** 1 = VRAM Loader data destination transfer is completed or halted by error. 0 = VRAM Loader data destination transfer is not completed or halted. **NOTE:** Write 1 to this bit can clear it to 0. This bit must be cleared before next transfer started.	0x0
[13:0]	**VDSTC**	**VRAM Loader Data Destination Transfer Count Number of Continuous Transfer**	0x0

4.5 Advanced Interrupt Controller

An *interrupt* temporarily changes the sequence of program execution to react to a particular event such as power failure, watchdog timer timeout, transmit/receive request from Serial Interface (UART or SPI) Controller, and so on. The ARM926 processor provides two modes of interrupt, the **Fast Interrupt (FIQ)** mode for critical session and the **Interrupt (IRQ)** mode for general purpose. The IRQ exception is occurred when the nIRQ input is asserted. Similarly, the FIQ exception is occurred when the nFIQ input is asserted. The FIQ has privilege over the IRQ and can preempt an ongoing IRQ. It is possible to ignore the FIQ and the IRQ by setting the F and I bits in the **current program status register (CPSR)**.

The W55VA91 incorporates the **advanced interrupt controller (AIC)** that is capable of dealing with the interrupt requests from a total of 32 different sources. Currently, 31 interrupt sources are defined. 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 31 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. Interrupt sources within the priority levels other than 0 can petition for the IRQ. The IRQ can be preempted by the occurrence of the FIQ. Interrupt nesting is performed automatically by the AIC.

Though interrupt sources originated from the W55VA91 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 W55VA91 is put in the test mode, all interrupt sources must be configured as positive-edge triggered.

The advanced interrupt controller includes the following features:

- AMBA APB bus interface
- 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
- Individual mask for each interrupt source
- Proprietary 8-level interrupt scheme to ease the burden from the interrupt
- Priority methodology is adopted to allow for interrupt daisy-chaining
- 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

4.5.1 Interrupt Sources

Priority	VA91	VA91 Source
1 (Highest)	**WDT_INT**	Watch Dog Timer Interrupt
2	**nIRQ0**	External Interrupt 0
3	**nIRQ1**	External Interrupt 1
4	**nIRQ2**	External Interrupt 2
5	**nIRQ3**	External Interrupt 3
6	**USBH_INT**	USB Host Interface Controller Interrupt
7	**APU_INT**	Audio Processing Unit Interrupt
8	**VPOST_INT**	Display Interface Controller Interrupt
9	**ADC_INT**	AD Converter Interrupt
10	**UART_INT**	UART Interrupt
11	**T_INT**	Timer Interrupt
12	**GPU_SLM_INT**	Scanline Match Interrupt
13	**GPU_HS_INT**	Horizontal Sync Interrupt
14	**GPU_SHB_INT**	Sprite Horizontal Blank Interrupt
15	**GPU_BHB_INT**	Background Horizontal Blank Interrupt
16	**GPU_VB_INT**	Vertical Blank Interrupt
17	**VIN_INT**	Video-in Interface Interrupt
18	**USBD_INT**	USB Device Interrupt
19	**VLD_INT**	VRAM Loader Interrupt
20	**GDMA_INT0**	GDMA Channel 0 Interrupt
21	**GDMA_INT1**	GDMA Channel 1 Interrupt
22	**SDIO_INT**	SD Host Interrupt
23	**JPEG_INT**	JPEG Interrupt
24	**SPI_INT**	SPI (Master/Slave) Serial Interface Interrupt
25	**SPI1_INT**	SPI(Master) Serial Interface Interrupt
26	**RTC_INT**	RTC Interrupt
27	**PWM_CAP_INT0**	PWM/Capture 0 Interrupt
28	**PWM_CAP_INT1**	PWM/Capture 1 Interrupt
29	**PWM_CAP_INT2**	PWM/Capture 2 Interrupt
30	**PWM_CAP_INT3**	PWM/Capture 3 Interrupt

31 (Lowest)	**RESERVED**	RESERVED

Table 4.5-1 Interrupt Source and Natural Priority List

4.5.2 AIC Functional Description

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 line. Each interrupt source belongs to priority group between of 0 to 7. Group 0 has the highest priority and group 7 the lowest. 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 IRQ 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, including FIQ, 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 (including FIQ) 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 AIC de-asserts the NIRQ line 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 interrupt 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	

Table 4.5-2 Software Actions in Normal Mode and ICE/Debug Mode

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.

4.5.3 Example of AIC Usage

Here we take APU as an example:

- According to table 7.5-1, we know that APU interrupt is allocated in interrupt channel 7. Therefore, before activating APU module, we should set AIC_SCR07 with the correct sensitive type and priority.
- Set bit 7 of AIC_MECR to 1 in order to enable APU interrupt channel.
- Run APU. APU module will active the interrupt whenever the threshold of audio buffer is reached. The interrupt service routine can read AIC_ISNR register to check if the interrupt comes from APU.
- After service routine has updated audio buffer and all other necessary jobs, it can finish the interrupt sevices by setting any value to AIC_EOSCR.

4.5.4 AIC Register Map

Register	Address	R/W	Description	Reset Value
AIC_BA = 0xFFF8_3000				
AIC_SCR1	AIC_BA+0x004	R/W	Source Control Register 1	0x0000.0047
AIC_SCR2	AIC_BA+0x008	R/W	Source Control Register 2	0x0000.0047
AIC_SCR3	AIC_BA+0x00C	R/W	Source Control Register 3	0x0000.0047
AIC_SCR4	AIC_BA+0x010	R/W	Source Control Register 4	0x0000.0047
AIC_SCR5	AIC_BA+0x014	R/W	Source Control Register 5	0x0000.0047
AIC_SCR6	AIC_BA+0x018	R/W	Source Control Register 6	0x0000.0047
AIC_SCR7	AIC_BA+0x01C	R/W	Source Control Register 7	0x0000.0047
AIC_SCR8	AIC_BA+0x020	R/W	Source Control Register 8	0x0000.0047
AIC_SCR9	AIC_BA+0x024	R/W	Source Control Register 9	0x0000.0047
AIC_SCR10	AIC_BA+0x028	R/W	Source Control Register 10	0x0000.0047
AIC_SCR11	AIC_BA+0x02C	R/W	Source Control Register 11	0x0000.0047
AIC_SCR12	AIC_BA+0x030	R/W	Source Control Register 12	0x0000.0047
AIC_SCR13	AIC_BA+0x034	R/W	Source Control Register 13	0x0000.0047
AIC_SCR14	AIC_BA+0x038	R/W	Source Control Register 14	0x0000.0047

AIC_SCR15	AIC_BA+0x03C	R/W	Source Control Register 15	0x0000.0047
AIC_SCR16	AIC_BA+0x040	R/W	Source Control Register 16	0x0000.0047
AIC_SCR17	AIC_BA+0x044	R/W	Source Control Register 17	0x0000.0047
AIC_SCR18	AIC_BA+0x048	R/W	Source Control Register 18	0x0000.0047
AIC_SCR19	AIC_BA+0x04C	R/W	Source Control Register 19	0x0000.0047
AIC_SCR20	AIC_BA+0x050	R/W	Source Control Register 20	0x0000.0047
AIC_SCR21	AIC_BA+0x054	R/W	Source Control Register 21	0x0000.0047
AIC_SCR22	AIC_BA+0x058	R/W	Source Control Register 22	0x0000.0047
AIC_SCR23	AIC_BA+0x05C	R/W	Source Control Register 23	0x0000.0047
AIC_SCR24	AIC_BA+0x060	R/W	Source Control Register 24	0x0000.0047
AIC_SCR25	AIC_BA+0x064	R/W	Source Control Register 25	0x0000.0047
AIC_SCR26	AIC_BA+0x068	R/W	Source Control Register 26	0x0000.0047
AIC_SCR27	AIC_BA+0x06C	R/W	Source Control Register 27	0x0000.0047
AIC_SCR28	AIC_BA+0x070	R/W	Source Control Register 28	0x0000.0047
AIC_SCR29	AIC_BA+0x074	R/W	Source Control Register 29	0x0000.0047
AIC_SCR30	AIC_BA+0x078	R/W	Source Control Register 30	0x0000.0047
AIC_SCR31	AIC_BA+0x07C	R/W	Source Control Register 31	0x0000.0047
AIC_IRSR	AIC_BA+0x100	R	Interrupt Raw Status Register	0x0000.0000
AIC_IASR	AIC_BA+0x104	R	Interrupt Active Status Register	0x0000.0000
AIC_ISR	AIC_BA+0x108	R	Interrupt Status Register	0x0000.0000
AIC_IPER	AIC_BA+0x10C	R	Interrupt Priority Encoding Register	0x0000.0000
AIC_ISNR	AIC_BA+0x110	R	Interrupt Source Number Register	0x0000.0000
AIC_IMR	AIC_BA+0x114	R	Interrupt Mask Register	0x0000.0000
AIC_OISR	AIC_BA+0x118	R	Output Interrupt Status Register	0x0000.0000
AIC_MECR	AIC_BA+0x120	W	Mask Enable Command Register	Undefined
AIC_MDCR	AIC_BA+0x124	W	Mask Disable Command Register	Undefined
AIC_SSCR	AIC_BA+0x128	W	Source Set Command Register	Undefined
AIC_SCCR	AIC_BA+0x12C	W	Source Clear Command Register	Undefined
AIC_EOSCR	AIC_BA+0x130	W	End of Service Command Register	Undefined
AIC_TEST	AIC_BA+0x134	W	ICE/Debug mode Register	0x0000_0000

4.5.5 AIC Register Description

AIC Source Control Registers (AIC_SCR1 ~ AIC_SCR31)

Register	Address	R/W	Description	Reset Value
AIC_SCR1	AIC_BA+0x004	R/W	Source Control Register 1	0x0000_0047
AIC_SCR2	AIC_BA+0x008	R/W	Source Control Register 2	0x0000_0047
…	**…**	**…**	**…**	**…**

AIC_SCR28	AIC_BA+0x070	R/W	Source Control Register 28	0x0000_0047
AIC_SCR29	AIC_BA+0x074	R/W	Source Control Register 29	0x0000_0047
AIC_SCR30	AIC_BA+0x078	R/W	Source Control Register 30	0x0000_0047
AIC_SCR31	AIC_BA+0x07C	R/W	Source Control Register 31	0x0000_0047

31	30	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
SRCTYPE		RESERVED			PRIORITY		

SRCTYPE [7:6]: Interrupt Source Type

Whether an interrupt source is considered active or not by the AIC is subject to the settings of this field. Interrupt sources except nIRQ0, nIRQ1, nIRQ2, nIRQ3, should be configured as level sensitive during normal operation. nIRQ0, nIRQ1, nIRQ2, nIRQ3 should use edge trigger.

SRCTYPE [7:6]		Interrupt Source Type
0	0	Low-level Sensitive
0	1	High-level Sensitive
1	0	Negative-edge Triggered
1	1	Positive-edge Triggered

PRIORITY [2:0]: Priority Level

Every interrupt source must be assigned a priority level during initiation. Among them, priority level 0 has the highest priority and priority level 7 the lowest. Interrupt sources with priority level 0 are promoted to FIQ. Interrupt sources with priority level other than 0 belong to IRQ. For interrupt sources with the same priority level, one located in the lower channel number has higher priority.

AIC Interrupt Raw Status Register (AIC_IRSR)

Register	Address	R/W	Description	Reset Value
AIC_IRSR	AIC_BA+0x100	R	Interrupt Raw Status Register	0x0000_0000

31	30	29	28	27	26	25	24
ISR31	ISR30	ISR29	IRS28	IRS27	IRS26	IRS25	IRS24
23	22	21	20	19	18	17	16
IRS23	IRS22	IRS21	IRS20	IRS19	IRS18	IRS17	IRS16
15	14	13	12	11	10	9	8
IRS15	IRS14	IRS13	IRS12	IRS11	IRS10	IRS9	IRS8
7	6	5	4	3	2	1	0
IRS7	IRS6	IRS5	IRS4	IRS3	IRS2	IRS1	RESERVED

This register records the intrinsic state within each interrupt channel.

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+0x104	R	Interrupt Active Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IAS31	IAS30	IAS29	IAS28	IAS27	IAS26	IAS25	IAS24
23	22	21	20	19	18	17	16
IAS23	IAS22	IAS21	IAS20	IAS19	IAS18	IAS17	IAS16
15	14	13	12	11	10	9	8
IAS15	IAS14	IAS13	IAS12	IAS11	IAS10	IAS9	IAS8
7	6	5	4	3	2	1	0
IAS7	IAS6	IAS5	IAS4	IAS3	IAS2	IAS1	RESERVED

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.

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)

Reg ister	Address	R/W	Description	Reset Value
AIC_ISR	AIC_BA+0x108	R	Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24
23	22	21	20	19	18	17	16
IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
15	14	13	12	11	10	9	8
IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8
7	6	5	4	3	2	1	0
IS7	IS6	IS5	IS4	IS3	IS2	IS1	RESERVED

This register identifies those interrupt channels whose are both active and enabled.

IS*x*: Interrupt Status
Indicates the status of corresponding interrupt channel
0 = Two possibilities:
- The corresponding interrupt channel is inactive no matter whether it is enabled or disabled;

- 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+0x10C	R	Interrupt Priority Encoding Register	0x0000_0000

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

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; otherwise, it is IRQ. The value of **VECTOR** is copied to the register AIC_ISNR thereafter by the AIC. *This register was 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.

VECTOR [6:2]: 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+0x110	R	Interrupt Source Number Register	0x0000_0000

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

The purpose of this register is to record the interrupt channel number that is active, enabled, and has the highest priority.

IRQID [4:0]: 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+0x114	R	Interrupt Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24
23	22	21	20	19	18	17	16
IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
15	14	13	12	11	10	9	8
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8
7	6	5	4	3	2	1	0
IM7	IM6	IM5	IM4	IM3	IM2	IM1	RESERVED

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+0x118	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 and IRQ. This register indicates whether the asserted interrupt is FIQ or IRQ. If both IRQ and FIQ are equal to 0, it means there is no interrupt occurred.

IRQ [1]: Interrupt Request

0 = nIRQ line is inactive.
1 = nIRQ line is active.

FIQ [0]: 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+0x120	W	Mask Enable Command Register	Undefined

31	30	29	28	27	26	25	24
MEC31	MEC30	MEC29	MEC28	MEC27	MEC26	MEC25	MEC24
23	22	21	20	19	18	17	16
MEC23	MEC22	MEC21	MEC20	MEC19	MEC18	MEC17	MEC16
15	14	13	12	11	10	9	8
MEC15	MEC14	MEC13	MEC12	MEC11	MEC10	MEC9	MEC8
7	6	5	4	3	2	1	0
MEC7	MEC6	MEC5	MEC4	MEC3	MEC2	MEC1	RESERVED

MEC *x*: Mask Enable Command

0 = No effect

1 = Enable the corresponding interrupt channel

AIC Mask Disable Command Register (AIC_MDCR)

Register	Address	R/W	Description	Reset Value
AIC_MDCR	AIC_BA+0x124	W	Mask Disable Command Register	Undefined

31	30	29	28	27	26	25	24
MDC31	MDC30	MDC29	MDC28	MDC27	MDC26	MDC25	MDC24
23	22	21	20	19	18	17	16
MDC23	MDC22	MDC21	MDC20	MDC19	MDC18	MDC17	MDC16
15	14	13	12	11	10	9	8
MDC15	MDC14	MDC13	MDC12	MDC11	MDC10	MDC9	MDC8
7	6	5	4	3	2	1	0
MDC7	MDC6	MDC5	MDC4	MDC3	MDC2	MDC1	RESERVED

MDC *x*: Mask Enable Command

0 = No effect

1 = Disable the corresponding interrupt channel

AIC Source Set Command Register (AIC_SSCR)

Register	Address	R/W	Description	Reset Value
AIC_SSCR	AIC_BA+0x128	W	Source Set Command Register	Undefined

31	30	29	28	27	26	25	24
SSC31	SSC30	SSC29	SSC28	SSC27	SSC26	SSC25	SSC24

23	22	21	20	19	18	17	16
SSC23	SSC22	SSC21	SSC20	SSC19	SSC18	SSC17	SSC16
15	14	13	12	11	10	9	8
SSC15	SSC14	SSC13	SSC12	SSC11	SSC10	SSC9	SSC8
7	6	5	4	3	2	1	0
SSC7	SSC6	SSC5	SSC4	SSC3	SSC2	SSC1	RESERVED

When the W55VA91 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.

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+0x12C	W	Source Clear Command Register	Undefined

31	30	29	28	27	26	25	24
SCC31	SCC30	SCC29	SCC28	SCC27	SCC26	SCC25	SCC24
23	22	21	20	19	18	17	16
SCC23	SCC22	SCC21	SCC20	SCC19	SCC18	SCC17	SCC16
15	14	13	12	11	10	9	8
SCC15	SCC14	SCC13	SCC12	SCC11	SCC10	SCC9	SCC8
7	6	5	4	3	2	1	0
SCC7	SCC6	SCC5	SCC4	SCC3	SCC2	SCC1	RESERVED

When the W55VA91 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.

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+0x130	W	End of Service Command Register	Undefined

31	30	29	28	27	26	25	24
---	---	---	---	---	---	---	---
23	22	21	20	19	18	17	16
---	---	---	---	---	---	---	---
15	14	13	12	11	10	9	8
---	---	---	---	---	---	---	---

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

This register is used by the interrupt service routine to indicate that it is completely served. Thus, the interrupt handler can write any value to this register to indicate the end of its interrupt service.

AIC ICE/Debug Register (AIC_TEST)

Register	Address	R/W	Description	Reset Value
AIC_TEST	AIC_BA+0x134	W	**ICE/Debug 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							**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.

TEST: ICE/Debug mode
0 = normal mode.
1 = ICE/Debug mode.

4.6 Audio Processing Unit

The main purpose of Audio Processing Unit (APU) is used to playback the audio data (PCM format) which CPU decoded and stored in global RAM. The APU built in a stereophonic DAC with 16-bit resolution per channel which supports speakerphone output and stereophonic output for headphone. The APU is composed of an AHB Master and built in FIFO and timer.

4.6.1 Overview and Features

- Built in a stereophonic DAC with 16-bit resolution per channel
- AHB Master with DMA.
- Built in FIFO with length 16Bytes * 2.
- Read Audio PCM data from global RAM
- Built in timer to generate conversion trigger signal automatically.

4.6.2 APU Functional Description

The audio DAC clock is the clock input for DAC from clock control module. You can set CLKSEL[9:0] to generate a clock with needed frequency. The frequency of this clock must be equal to (input audio data sampling rate) x 128. For example, if the sampling rate is 48 KHz, then the clock should be 128 x 48 KHz = 6.144 MHz. The APU module internally handles a 7-bit counter that it will generate a STORBE signal to DAC whenever the counter reaches 128. This makes DAC to output the audio data with correct sampling rate.

The following is an example for APU run procedures.

1. Setup clock source :
   - If the PLL output frequency set by PLLCON is 240 MHz and the sample rate of input data is 48 KHz, the AUDIO_DAC clock should be set to 6.144 MHz. Therefore, CLKSEL[9:0] should be set a proper value to divide the clock source by about 39 (240/6.144)). We my set CLKSEL[9:0]=0x026.
2. Set base address and threshold addresses
   - The APU implement the ping-pong buffer mechanism and the buffers are consecutive. You can set the start address of first buffer in BSAD register, set the end address of fist buffer in THAD1 register, and set the end address of second buffer in THAD2 register. Remember to set APUINT register to enable threshold 1/2 interrupts. If the registers are set properly, every time the APU reach the end of each one buffer, it will issue an interrupt and then you can update the buffer.
3. Reset APU before start
   - Set bit 16 of APUCON register to 1 and then set it to 0 again. This action will reset internal buffers and registers. Remember to do this step before you start to run APU.
4. Start APU
   - Set bit 0 of APUCON register to 1. This makes APU start to transfer audio data from buffer to DAC.

4.6.3 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
APU_BA = 0xFFF0_A000				
APUCON	APU_BA + 0x00	R/W	APU Control Register	0x0000_0000
PARCON	APU_BA + 0x04	R/W	Parameter Control Register	0x0000_0030
VOLCON	APU_BA + 0x08	R/W	Volume Control Register	0x03ff_0000
APUINT	APU_BA + 0x0C	R/W	APU Interrupt Register	0x0000_0000
RAMBSAD	APU_BA + 0x10	R/W	RAM Base Address Register	0x0000_0000
THAD1	APU_BA + 0x14	R/W	Threshold 1 Address Register	0x0000_0000
THAD2	APU_BA + 0x18	R/W	Threshold 2 Address Register	0x0000_0000
CURAD	APU_BA + 0x1C	R	Current Access RAM Address Register	0x0000_0000
EQGAIN0	APU_BA + 0x20	R/W	Equalizer Band Gain Register 0	0x7777_7777
EQGAIN1	APU_BA + 0x24	R/W	Equalizer Band Gain Register 1	0x000d_0077
FMCON	APU_BA + 0x28	R/W	FM control register	0x0000_0000
APURAMBIST	APU_BA + 0x2C	R/W	APU ram BIST control register	0x0000_0000

4.6.4 Control Registers

APU Control Register

Register	Address	R/W	Description	Reset Value
APUCON	APU_BA+0x00	R/W	APU Control Register	0x0000_0000

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

Bits	Descriptions	
[31:17]	**Reserved**	**Reserved**
[16]	**APURST**	**APU Reset** 0 = No action 1 = Reset the whole ADC except register value.

		Notice: This Reset bit will not clear APURUN. If you want to reset the whole module, use bit1 of RSTCON (ADORST bit).
[15:1]	**Reserved**	**Reserved**
[0]	**APURUN**	**APU Run** 0 = Disable 1 = Enable If you want to start-up a new audio data transfer, remember to set this bit to 0 and then set it to 1 again.

Parameter Control Register

Register	Address	R/W	Description	Reset Value
PARCON	APU_BA+0x04	R/W	Parameter Control Register	0x0000_0030

31	30	29	28	27	26	25	24
Reserved						**ZERO_EN**	**EQU_EN**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**SWAP**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**Discharge_en**	**Discharge_con**		**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**POP_CON**		**Reserved**			**MONO**

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25]	**ZERO_EN**	**Zero cross detection enable** 0 = Disable 1 = Enable
[24]	**EQU_EN**	**Equalizer Enable** 0 : Disable 1 : Enable
[23:17]	**Reserved**	**Reserved**
[16]	**SWAP**	**PCM data format** 0 = MSB is sample data 2, LSB is sample data 1 1 = MSB is sample data 1, LSB is sample data 2
[15]	**Reserved**	**Reserved**
[14]	**Discharge_en**	Enable the discharging path for output coupling capacitor. "1"=enable, "0"=disable (This bit is write-only)
[13:12]	**Discharge_con**	Control the resistor on the discharging path. "00"=set R for 10uF C load, "01"=for 47uF, "10"=for 100uF, "11"=for 220uF. (The two bits are write-only)

[11:6]	**Reserved**	**Reserved**
[5:4]	**POP_CON**	**Pop noise control register**
[3:1]	**Reserved**	**Reserved**
[0]	**MONO**	**PCM data format is Mono or Stereo** 0 = stereo 1 = mono (extend the half-word sample data to both left and right channel)

MONO	SWAP	Global RAM data[31:16]	Global RAM data[15:0]
0	0	Sample data 1 right channel	Sample data 1 left channel
	1	Sample data 1 left channel	Sample data 1 right channel
1	0	Sample data 2	Sample data 1
	1	Sample data 1	Sample data 2

Volume Control Register

Register	Address	R/W	Description	Reset Value
VOLCON	APU_BA+0x08	R/W	Volume Control Register	0x03ff_0000

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

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25:16]	**ANA_PD**	**Audio DAC Power Down** ANA_PD[0] ：power down of ramp ANA_PD[1] ：power down of DAC_L ANA_PD[2] ：power down of DAC_R ANA_PD[3] ：power down of volume control L ANA_PD[4] ：power down of volume control R ANA_PD[5] ：power down of headphone amp L and headphone amp R ANA_PD[6] ：power down of reference voltage ANA_PD[7] ：power down of current bias ANA_PD[8] ： power-down control for left FM input;

		ANA_PD[9] : power-down control for right FM input; "1"=enable power down, "0"=disable power down
[15:13]	**Reserved**	**Reserved**
[12:8]	**LHPVL**	**Headphone Left Channel Volume**
[7:5]	**Reserved**	**Reserved**
[4:0]	**RHPVL**	**Headphone Right Channel Volume**

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

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

APU Interrupt Register

Register	Address	R/W	Description	Reset Value
APUINT	APU_BA+0x0C	R/W	APU Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**T2INTEN**	**T1INTEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**T2INTS**	**T1INTS**

Bits	Descriptions	
[31:18]	**Reserved**	**Reserved**
[17]	**T2INTEN**	**Threshold 2 Interrupt Enable** 0 = Disable 1 = Enable
[16]	**T1INTEN**	**Threshold 1 Interrupt Enable** 0 = Disable

		1 = Enable
[15:2]	**Reserved**	**Reserved**
[1]	**T2INTS**	**Threshold 2 Interrupt Status** APU fetch data from Threshold 2 complete. Write 0 to clear it.
[0]	**T1INTS**	**Threshold 1 Interrupt Status** APU fetch data from Threshold 1 complete. Write 0 to clear it.

RAM Base Address Register

Register	Address	R/W	Description	Reset Value
RAMBSAD	APU_BA+0x10	R/W	RAM Base Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**BSAD**	**Global RAM Base Address**

Threshold 1 Address Register

Register	Address	R/W	Description	Reset Value
THAD1	APU_BA+0x14	R/W	Threshold 1 Address Register	0x0000_0000

31	30	29	28	27	26	25	24
TH1[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TH1[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TH1[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

TH1[7:0]

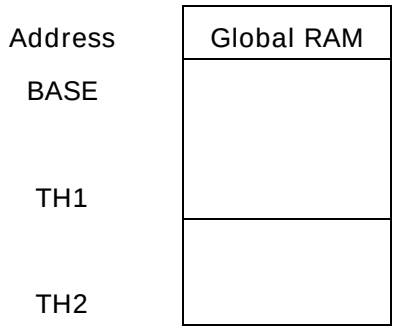
Bits	Descriptions	
[31:0]	TH1	Threshold 1 Address

Threshold 2 Address Register

Register	Address	R/W	Description	Reset Value
THAD2	APU_BA+0x18	R/W	Threshold 2 Address Register	0x0000_0000

31	30	29	28	27	26	25	24
TH2[31:24]							
23	22	21	20	19	18	17	16
TH2[23:16]							
15	14	13	12	11	10	9	8
TH2[15:8]							
7	6	5	4	3	2	1	0
TH2[7:0]							

Bits	Descriptions	
[31:0]	TH2	Threshold 2 Address



Current Address Register

Register	Address	R/W	Description	Reset Value
CURAD	APU_BA+0x1C	R	Current Access RAM Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURAD[31:24]							

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

Bits	Descriptions	
[31:0]	CURAD	Current APU Access RAM Address

Equalizer Band Gain Register 0

Register	Address	R/W	Description	Reset Value
EQGain0	APU_BA+0x20	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)
[27:24]	Gain07	Equalizer Band 07 Gain control (2000Hz)
[23:20]	Gain06	Equalizer Band 06 Gain control (1000Hz)
[19:16]	Gain05	Equalizer Band 05 Gain control (500Hz)
[15:12]	Gain04	Equalizer Band 04 Gain control (250Hz)
[11:8]	Gain03	Equalizer Band 03 Gain control (125Hz)
[7:4]	Gain02	Equalizer Band 02 Gain control (62Hz)
[3:0]	Gain01	Equalizer Band 01 Gain control (31Hz)

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	APU_BA+0x24	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)**
[3:0]	**Gain09**	**Equalizer Band 09 Gain control (8000Hz)**

FM module control register

Register	Address	R/W	Description	Reset Value
FMCON	APU_BA+0x28	R/W	FM module 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							EN_FM
7	6	5	4	3	2	1	0
Reserved		LFMVL		Reserved		RFMVL	

Bits	Descriptions	
[31:9]	**Reserved**	**Reserved**
[8]	**EN_FM**	**Input Signal Selection** 0 : DAC 1 : FM
[7:6]	**Reserved**	**Reserved**
[5:4]	**LFMVL**	**Left FM input volume control** 00 : 0dB 01 : 12dB 10 : 18dB 11 : 21.58dB
[3:2]	**Reserved**	**Reserved**
[1:0]	**RFMVL**	**Right FM input volume control** 00 : 0dB 01 : 12dB 10 : 18dB 11 : 21.58dB

APU RAM BIST Register (APURAMBIST)

Register	Address	R/W	Description	Reset Value
APURAMBIST	APU_BA+0x2C	R/W	APU ram BIST 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					BistFail	Finish	BISTEN

Bits	Descriptions

[2]	**BistFail**	**BIST Fail** The BistFail indicates if the BIST test fails or succeeds. If the BistFail is low at the end, the embedded SRAM pass the BIST test, otherwise, it is faulty. The BistFail will be high once the BIST detects the error and remains high during the BIST operation. This bit is a read only. Write 0 or 1 to this bit has no effect. User can write 0 to BISTEN to clear this bit.
[1]	**Finish**	**BIST Operation Finish** It indicates the end of the BIST operation. When BIST controller finishes all operations, this bit will be set high. This bit is a read only. Write 0 or 1 to this bit has no effect. User can write 0 to BISTEN to clear this bit.
[0]	**BISTEN**	**BIST Enable** The BISTEN is used to enable the BIST operation. If high enables the BIST controller to do embedded SRAM test. This bit is also used to do the reset for BIST circuit. It is necessary to reset the BIST circuit one clock cycle at least in order to initialize the BIST properly. The BISTEN can be disabled by write 0.

4.7 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 timing) or set by the LCD timing control register. The video/image data source may come from the image sensor, GPU line buffer, frame buffer which stored in system memory (SDRAM). The TV picture and LCD picture can display individual image source simultaneously when the timing is synchronized with TV timing. It also supports 2 pictures overlapping mode and record the overlapping picture to frame buffer. The 2 picture source can be frame buffer which has different data format or line buffer.

4.7.1 Overview and Features

- Supports 3 types LCD
  - 8bit Sync-Type TFT LCD
  - 16 bit Sync-Type TFT LCD
  - MPU-Type LCD
- Sync-Type TFT LCD
  - Supports CCIR601 4:2:2 YCbCr packet mode (NTSC/PAL)
  - Supports CCIR601 RGB Dummy mode (NTSC/PAL)
  - Supports CCIR656 Interface
  - Support RGB Through mode
- Support VDMA mode for 2 MPU type panel
- Support NTSC/PAL interlace & non-interlace system
- Convert GPU line buffer data RGB555 display data to CCIR601(YCrCb422)
- Support Dual Screen (TV and LCD display)
  - The TV picture and LCD picture can display individual image source simultaneously when the timing is synchronized with TV timing.
- LCD Timing Setting Method
  - Sync with TV(NTSC/PAL) Mode
  - Timing Control Register Setting
  - MPU type access timing can be programable
- Support 2 Picture Overlapping (different data format which the data source can come from Line Buffer or 2Frame Buffer)
- Support Overlapping data restore to Frame Buffer
- Support rainbow effect remove filter(Notch Filter)
- Support NTSC/PAL interlace mode flick free filter
- Manual Control GPU Line counter can support to VGA
- Manual update can be support on the VGA resolution

LCD device / Pin name	Sync-type High Color TFT LCD (**16 bit data bus**)	Sync-type TFT LCD (**8 bit data bus**)	MPU-type LCD	
			80 Mode	68 Mode
LCD_VSYNC(GPB0)	VSYNC	VSYNC	RD	EN
LCD_HSYNC(GPB1)	HSYNC	HSYNC	WR	RW

LCD_PIXCLK(GPB2)	PCLK	PCLK	CS	Cs
LCD_VDEN(GPA0)	DE	DE	RS	RS
LCD_VDATA[15:0]	DATA[15:0]	DATA[7:0]	DATA[15:0]	DATA[15:0]

Table 4.7-1 VPOST Controller Interface Diagram

LCD Data	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I/O Pin	GPE2	GPE1	GPE0	GPB15	GPB14	GPB13	GPB12	GPB11	GPB10	GPB9	GPB8	GPB7	GPB6	GPB5	GPB4	GPB3
RGB555	1'b0	R7	R6	R5	R4	R3	G7	G6	G5	G4	G3	B7	B6	B5	B4	B3
RGB565	R7	R6	R5	R4	R3	G7	G6	G5	G4	G3	G2	B7	B6	B5	B4	B3

Note: To configure the bit[6:4] of PINFUN(0xFFF0_001C) register to set the I/O pins as LCD related output pins.

Table 4.7-2 VPOST LCD Data Output Pin Definiton

4.7.2 VPOST Controller Block Diagram

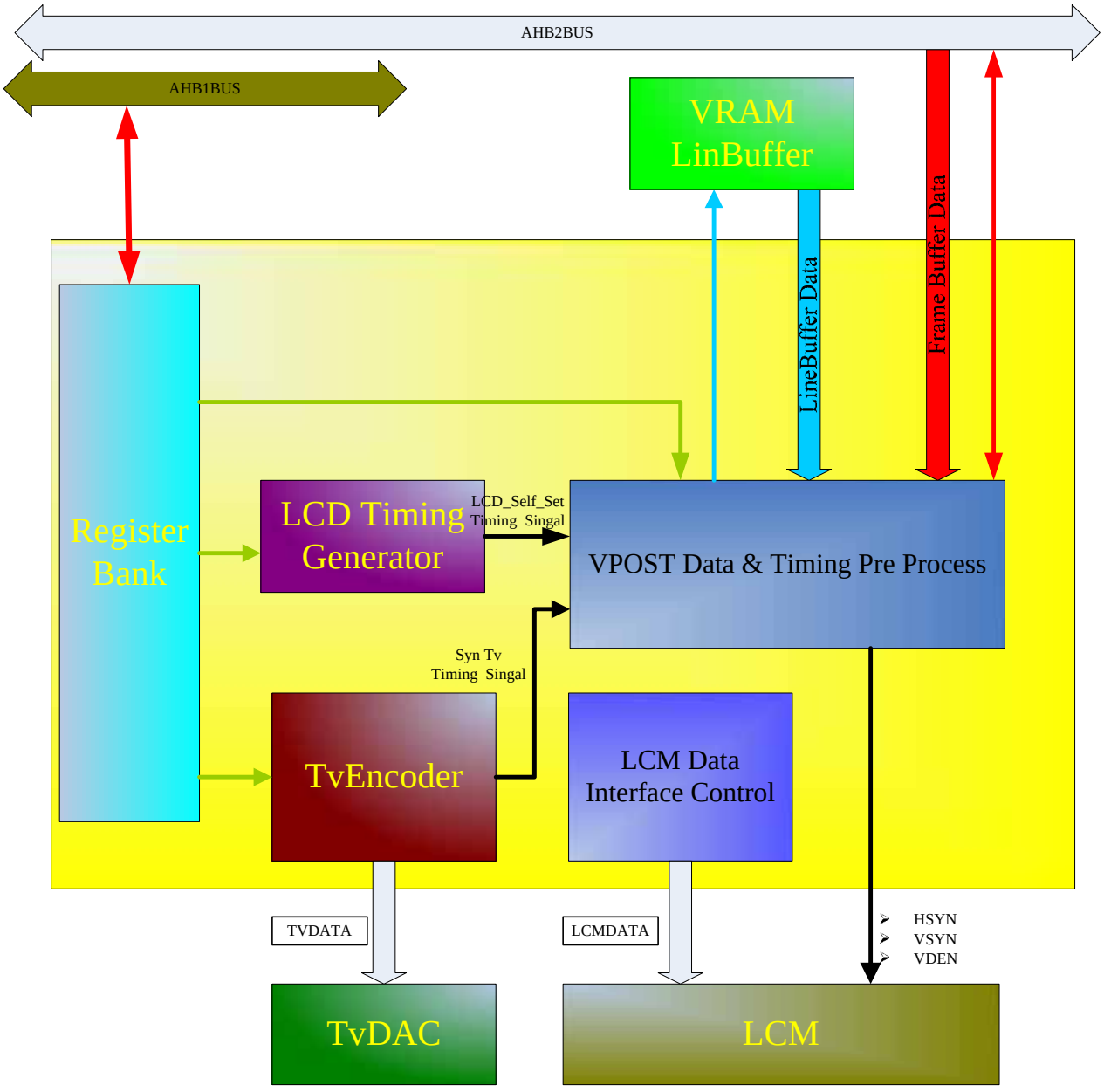


Figure 4.7-1 VPOST Controller Block Diagram

4.7.3 VPOST Controller Functional Description

TVEncoder

- Generate TV NTST/PAL data signal to current DAC.
- Generate timing control signal to LCD device.

LCD Timing Generator

- user-self control timing

Register Bank

- AHB Slave interface on AHB1.
- A bridge that CPU control and observe the state of LCD Controller.

VPOST Data & Timing Pre-Process

- To generate the data request signal
- To modify hsyn & 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.

4.7.4 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
SYS_BA = 0xFFF0_6000				
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
FEADDR	0xC	R/W	Frame Buffer End Address Register	0x0000_0000
TCON1	0x10	R/W	Timing Control Register 1	0x0000_0000
TCON2	0x14	R/W	Timing Control Register 2	0x0000_0000
TCON3	0x18	R/W	Timing Control Register 3	0x0000_0000
TCON4	0x1C	R/W	Timing Control Register 4	0x0140_0000
MPUCMD	0x20	R/W	MPU-type LCD Command Register	0x0000_0000
MPUTS	0x24	R/W	MPU Type timing control	0x0000_0000
FDFSel	0x28	R/W	2 Frame Buffer Data Format Selection	0x0008_0000
OlapCtl	0x2C	R/W	Overlapping Control Register	0x0000_0000
FSADDR1	0x30	R/W	Frame1 Buffer Base Address	0x0000_0000
FSADDR2	0x34	R/W	Frame2 Buffer Base Address	0x0000_0000
FSADDR3	0x38	R/W	Overlapping Data Record Base Address	0x0000_0000
CBAR	0x3C	R/W	Color Burst Active Region	0x0000_0000
TVCtl	0x40	R/W	TvControl Register	0x0000_0321
IIRA	0x44	R/W	IIR notch filter Denominator Coefficient	0x0058_0c00
IIRB	0x48	R/W	IIR notch filter Numberator Coefficient	0x042C_0d0b

COLORSET	0x4C	R/W	BackDraw Color Setting Register	0x0000_0000
FSADDR	0x50	R/W	Frame Buffer Start Address Register	0x0000_0000
TvDisCtl	0x54	R/W	TV Display Control Register	0x10F0_12A5
CBACtl1	0x58	R/W	Color Burst Amplitude Control Register1	0x0100_100b
CBACtl2	0x5C	R/W	Color Burst Amplitude Control Register2	0x0100_010b
MCDLR	0x60	R/W	Customer Display Control Register	0x0012_0000
TvContrast	0x64	R/W	Tv contrast adjust setting register	0x0080_8080
TvBright	0x68	R/W	Tv Bright adjust setting register	0x0000_0000
ESADDR	0x6c	R/W	EBI Start Address Register	0x2000_0000
MCDLR1	0x70	R/W	Customer Display Control Register	0x0000_EF00
RGBin	0x74	R/W	RGB 888 Data Input for RGB2YCbCr equation	0x000_0000
YCbCrout	0x78	R	YCbCr Data Output for RGB2YCbCr equation	0x0010_8080
YCbCrin	0x7c	R/W	YCbCr Data Input for YCbCr2RGB equation	0x0010_8080
RGBout	0x80	R	RGB Data Output for YCbCr2RGB equation	0x0010_1010

4.7.5 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
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							YUVBL
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				ATDOS		ATDBS	ATBURS
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ATDT	ATDS	ATDF	ATFDE	FBDS			LCDRUN

Bits	Descriptions	
[31:17]	**Reserved**	**Reserved**
[16]	**YUVBL**	The Y Cb(U) Cr(V) Big Endian & Little Endian selction 0 = BigEndian 1 = Little Endian **Note:** The bit was combined with FBDS for YCbCr packet data format selection

[15:12]	**Reserved**	**Reserved**
[11:10]	**ATDOS**	**Assemble & Transfer Data Output Selection** 00 = The data just output to EBI (SDRAM or External I/O 16 Data Pins) 01 = The data just output LCD interface(MPU Type timing) 10 = The data both output to EBI & LCD interface 11 = Reserved
[9]	**ATDBS**	**Assemble & Transfer Data Busy Status (only Read)** 0 = Assemble & Transfer is ready for next trigger 1 = Assemble & Transfer is under runnig
[8]	**ATBURS**	**Assemble & Transfer Burst Selection** 0 = 4 beat burst transfer (4 words for each transfer; More bandwidth are consumed.) 1 = Single Transfer (1 word for each transfer)
[7]	**ATDT**	**Assemble & Transfer Data Trigger** Only 0 →1 & ATDBS=0 : Trigger Enable 0 → 0, 1→0,1→1 : Trigger Disable
[6]	**ATDS**	**Assemble Data Half Word Sequence** 0 = Little Endian 1 = Big Endian
[5]	**ATDF**	**Assemble & Transfer Data Format** 0 = RGB555 1 = RGB565
[4]	**ATFDE**	**Enable a AHB bus request from** 0 = TV display 1 = VDMA (a H/W function inside VPOST) for moving frame buffer data
[3:1]	**FBDS**	**Frame Buffer Data Selection** 000 = RGB555 001 = RGB565 010 = Reserved 011 = Reserved 100 = Cb_0 Y_0 Cr_0 Y_1 101 = Y_0 Cb_0 Y_1 Cr_0 110 = Cr_0 Y_0 Cb_0 Y_1 111 = Y_0 Cr_0 Y_1 Cb_0 Note: YUVBL will impact the sequence, please refer to the below table for its effect. They are also used for overlapping display, please refer to the OlapDFSel.
[0]	**LCDRUN**	**LCD Controller Run** 0 = Disable 1 = Enable

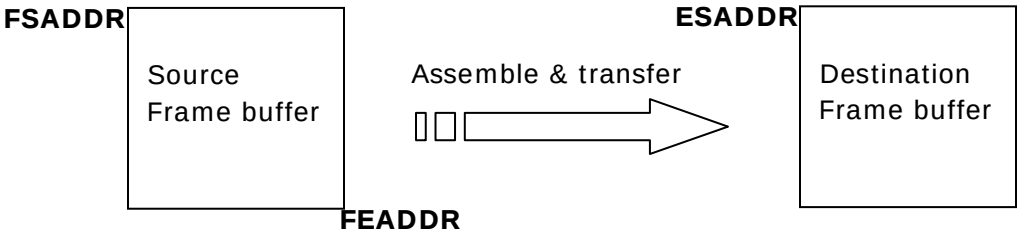
Note Assemble & transfer data function is for outputting a block of data to EBI and/or LCD interface. In addition, you need to configure at least these important registers below.

◎**FSADDR**: Frame buffer start address

◎**FEADDR**: Frame buffer end address

→ Source buffer information

◎**ESADDR**: EBI start address → Destination buffer information (Only the start address is specified!)



Related fields in register '**LCDCCtl**':

▲ **FBDS** (Frame Buffer Data Selection) specifies the format of source frame buffer data.

▲ **ATDF** (Assemble & Transfer Data Format) specifies the target converted data format which are about to output to destination buffer.

◎ LCD Controller Parameter Register

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

31	30	29	28	27	26	25	24
Even_Field_AL				Odd_Field_AL			

23	22	21	20	19	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
Reserved			LCDDataSel			LCDTYPE	

Bits	Descriptions	
[31:28]	**Odd_Field_AL**	CCIR656 Odd Field Dummy Active Line Odd Filed Total Active Line=240+ Odd_Field_AL
[27:24]	**Even_Field_AL**	CCIR656 Even Field Dummy Active Line Even Filed Total Active Line=240+ Even_Field_AL
[23:15]	**F1_EL**	CCIR656 Filed1(Odd Field) Ending Line
[14:9]	**F1_SL**	CCIR656 Filed1(Odd Field) Start Line
[8]	**LCDSynTv**	**LCD timing Synch with TV** 0 = disable 1 = enable
[4:2]	**LCDDataSel**	**LCD data interface Select** 000 = YUV422(CCIR601) 001 = dummy serial (R, G, B Dummy) (default) 010 = **CCIR656 Output** 100 = odd line data is RGB, even line is GBR 101 = odd line data is BGR, even line is RBG 110 = odd line data is GBR, even line is RGB 111 = odd line data is RBG, even line is BGR
[1:0]	**LCDTYPE**	**LCD device Type Select** 00 = High Resolution mode 01 = Sync-type TFT LCD

		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		**RecFrameINTEN**	**ATCINTEN**	**LCINTEN**	**FCINTEN**	**VINTEN**	**HINTEN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**RecFrameINT**	**ATCINT**	**LCINT**	**FCINT**	**VINT**	**HINT**

Bits	Descriptions	
[31:22]	**Reserved**	**Reserved**
[21]	**RecFrameINTEN**	**Record 1 Frame Overlapping Data Interrupt Enable** 0 = Disable 1 = Enable
[20]	**ATCINTEN**	**EBI Assemble &Transfer Complete Interrupt Enable** 0 = Disable 1 = Enable
[19]	**LCINTEN**	**Line Complete Interrupt Enable** 0 = Disable 1 = Enable
[18]	**FCINTEN**	**Field Complete Interrupt Enable** 0 = Disable 1 = Enable
[17]	**VINTEN**	**LCD VSYNC Interrupt Enable** 0 = Disable 1 = Enable
[16]	**HINTEN**	**LCD HSYNC Interrupt Enable** 0 = Disable 1 = Enable

[15:6]	**Reserved**	**Reserved**
[5]	**RecFrameINT**	**Record Overlapping Data 1 Frame Interrupt** When record 1 Frame Overlapping data, the bit will show 1 when CPU read it. Write 0 to clear it.
[4]	**ATCINT**	**EBI Assemble &Transfer Complete Interrupt** EBI assember & transfer complete; the bit will show 1 when cpu read it. Write 0 to clear it.
[3]	**LCINT**	**Line Transfer&Plot Complete Interrupt** For Manual Control Gpu Line Counter Mode, if the all lines in one field had done; the bit will show 1 when cpu read it. Write 0 to clear it.
[2]	**FCINT**	**Field Transfer&Plot Complete Interrupt** For Manual Control Gpu Line Counter Mode, if the all lines in one field had done; the bit will show 1 when cpu read it. Write 0 to clear it.
[1]	**VINT**	**LCD VSYNC/RD 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. For MPU-type LCD, if read this bit shows 1, LCDC doing a read cycle to LCD device complete. Write 0 to clear it.
[0]	**HINT**	**LCD HSYNC/WR 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. For MPU-type LCD, if read this bit shows 1, LCDC doing a write cycle to LCD device complete. Write 0 to clear it.

◎ **Frame Buffer End Address Register**

Register	Address	R/W	Description	Reset Value
FEADDR	0x0C	R/W	Frame Buffer End Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	**FEADDR**	Frame Buffer End Address

◎ **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 determines the HSYNC pulse's high level width by counting the number of the LCD Pixel Clock.
[15:8]	**HBPD**	Horizontal back porch is 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 is the number of LCD Pixel Clock periods between the end of active data and the rising edge of HSYNC.

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

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

◎ **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]							

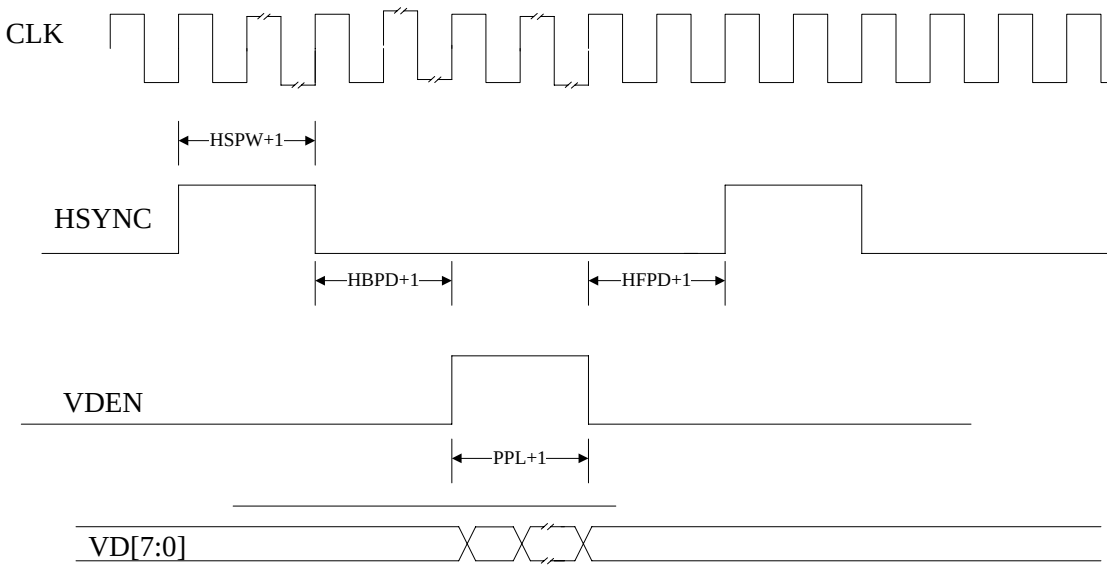
Bits	Descriptions	
[31:16]	**PPL**	Pixel Per-Line The PPL bit field specifies the number of pixels in each line or row of screen.
[15:0]	**LPP**	Lines Per-Panel The LPP bit field specifies the number of active lines per screen.

◎ **Timing Control Register 4**

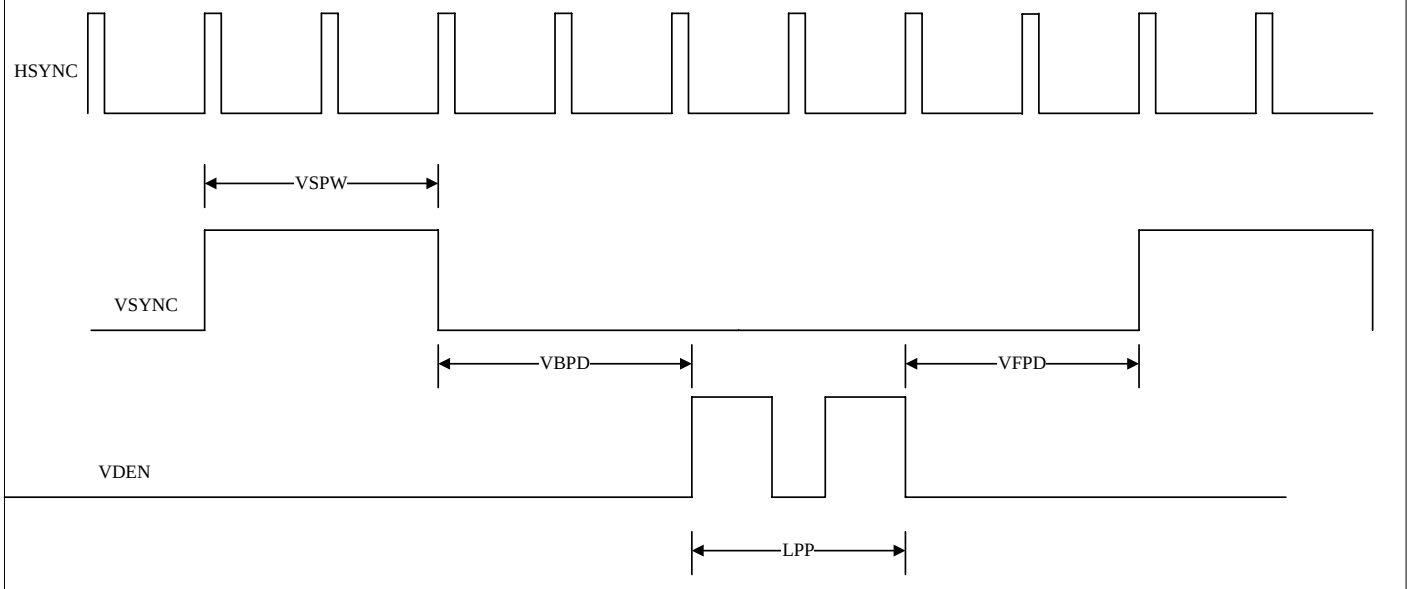
Register	Address	R/W	Description	Reset Value
TCON4	0x1c	R	Timing Control Register 4	0x0140_0000

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

Bits	Descriptions	
[31:26T]	**Reserved**	**Reserved**
[25:16]	**TAPN**	Total Active Pixel Number
[15:4]	**Reserved**	**Reserved**
[3]	**VSP**	**LCD VSYNC Polarity** 0 = Active Low 1 = Active High
[2]	**HSP**	**LCD HSYNC Polarity** 0 = Active Low 1 = Active High
[1]	**DEP**	**LCD VDEN Polarity** 0 = Active Low 1 = Active High
[0]	**PCLKP**	**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
Reserved		CMD_DISn	MPU_CS	CMD_ON	BUSY	WR_RS	READ
23	22	21	20	19	18	17	16
Reserved						MPUDisT	MPUFrame
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	
[30:29]	**CMD_DISn**	**Command or Display Mode Selction** 00 = Normal video display mode(Sync TV timing 320x240 source) 01 = Turn-on command mode for sending LCD command or parameter data; 10 = Video display come from Assembler & Transfer; 11 = Reserved
[28]	**MPU_CS**	**LCD interface CS control (for GPIOB2)** 0 = Output pin CS=0 1 = Output pin CS=1
[27]	**CMD_ON**	**Command Mode** Only 0 →1= Trigger Enable 0 → 0, 1→0,1→1 = Trigger Disable
[26]	**BUSY**	**Command interface is busy** 0 = Command interface is ready for next command 1 = Command interface is busy for writing/reading pending command
[25]	**WR_RS**	**Write/Read RS Setting** 0 = Output pin RS is equal to 0

		1 = Output pin RS is equal to 1
[24]	**MPU_RWn**	**Write/Read Status or data** 0 = Write command/parameter 1 = Read status/data from LCD
[23]	**MPU68**	**MPU interface Selection** 0 = 80-series MPU interface 1 = 68-series MPU interface
[17]	**MPUDisT**	MPU Display Trigger 0 = disable 1 = enable
[16]	**MPUFrame**	0 = Single Frame 1 = Multi Frame Mode
[15:0]	**MPU_CMD**	MPU-type LCD command/parameter data, read data

◎ **MPU-type LCD Timing Setting Register**

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

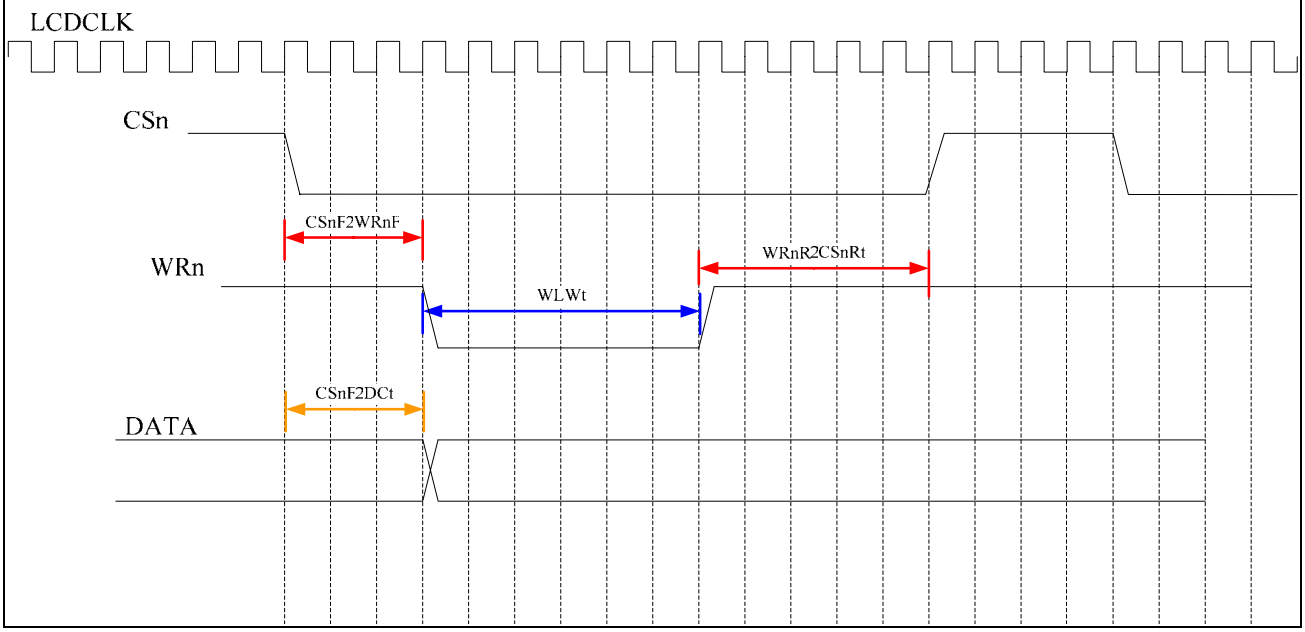
31	30	29	28	27	26	25	24
CSnF2DCt							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WRnR2CSnRt							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WRnLWt							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CSnF2WRnFt							

Bits	Descriptions	
[31:24]	**CSnF2DCt**	CSn fall edge to Data change clock counter
[23:16]	**WRnR2CSnRt**	WRn rising edge to CSn rising clock counter

[15:8]	**WRnLWt**	WR Low pulse clock counter
[7:0]	**CSnF2WRnFt**	Csn fall edge To WR falling edge clock counter

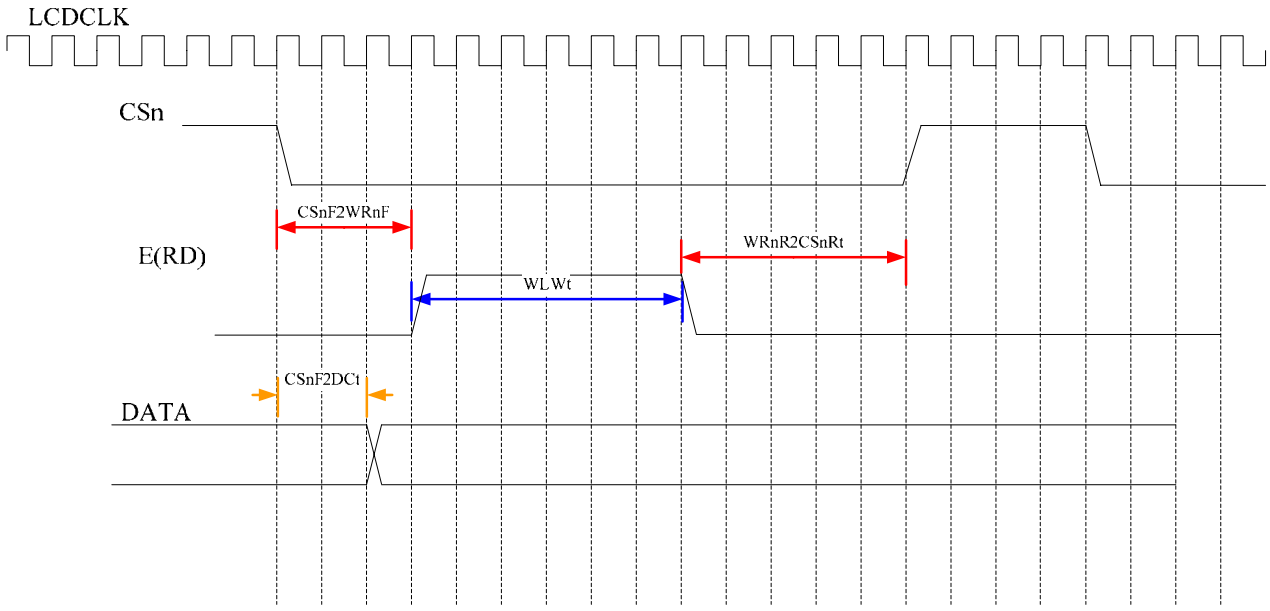
CF2WFt:Cs fall edge to WR falling edge clock count

80 Mode Write Data Timing WaveForm



CF2WFt:Cs fall edge to WR falling edge clock count

68 Mode Write Data Timing WaveForm

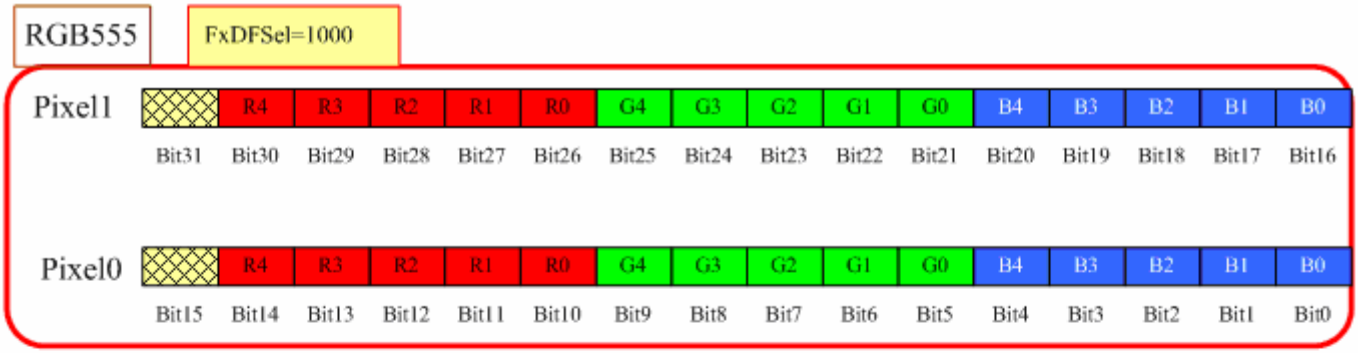


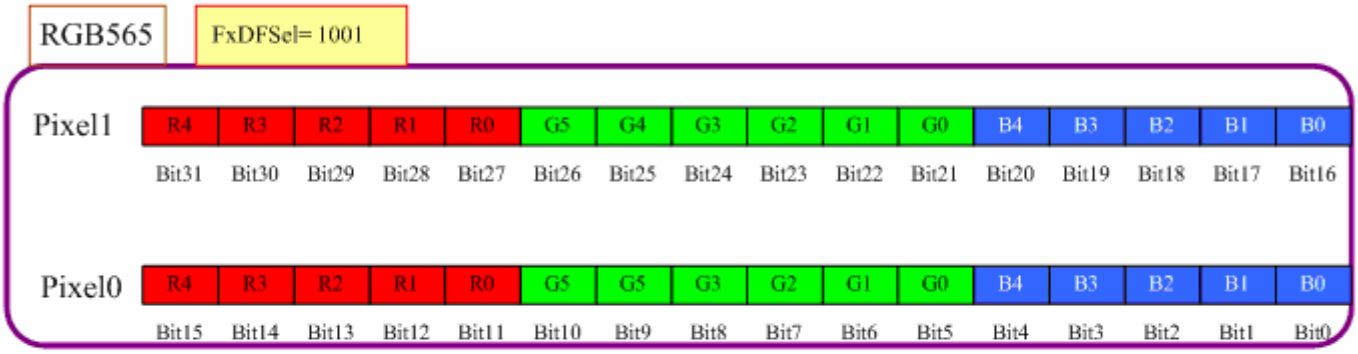
◎ Frame Buffer Source Data Format (For frame buffer overlapping)

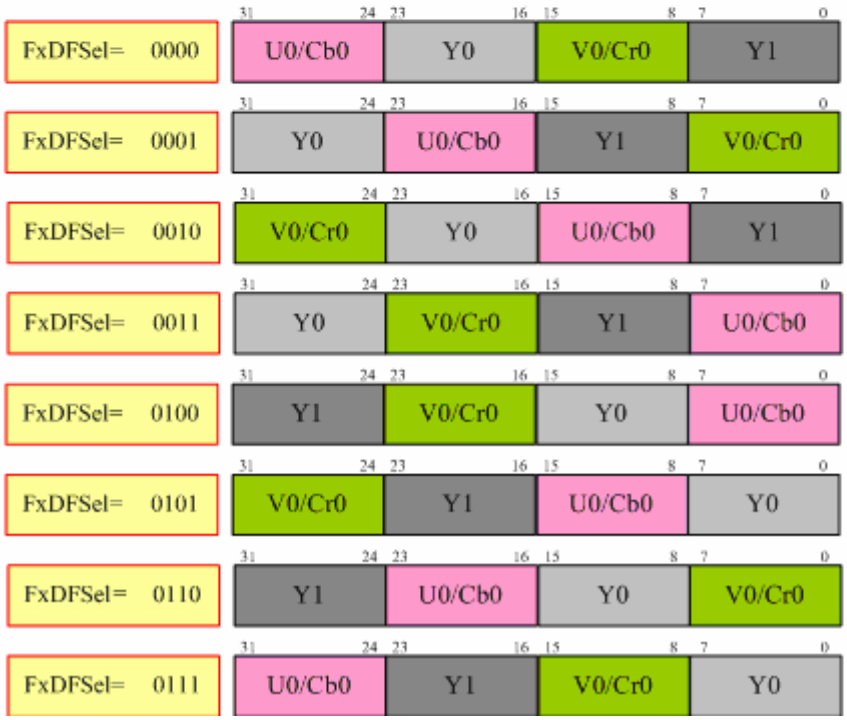
Register	Address	R/W	Description	Reset Value
FDFSel	0x28	R/W	Frame Buffer Source Data Format	0x0008_0000

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
				F1DFSel			

Bits	Descriptions	
[19:16]	**Reserved**	8 for RGB555 only
[3:0]	**F1DFSel**	**Frame Buffer 1 Data Format Selection** (Default: Cb0Y0Cr0Y1 mode) Please reference illustration below.







◎ **Overlapping Control Register**

Register	Address	R / W	Description	Reset Value
OlapCtl	0x2c	R/W	Overlapping Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			OlapDR	Reserved		OlapDRM	Pic2Sel
23	22	21	20	19	18	17	16
Reserved				Reserved			
15	14	13	12	11	10	9	8
Reserved				Reserved			
7	6	5	4	3	2	1	0
OlapDFSel				Reserved		OlapSel	

Bits	Descriptions	
[28]	**OlapDR**	**Overlapping Data Record** 0 = Record Disable 1 = Record Enable **Note:** ■ Need to set FSADDR1 same as FSADDR when OlapSel=10 ■ Need to set OlapDR from 0 to 1 if changing RecFrameINTEN & OlapDRM ■ Set OlapDR from 0 to 1 for each record if OlapDRM=0
[25]	**OlapDRM**	**Overlapping Data Record Method** 0 = Single Frame (For taking a picture) 1 = Multi Frame Mode (For video recording) **Ps.** The address of target frame buffer is specified in register '**FSADDR3**'.
[24]	**Pic2Sel**	**2 Picture Selection** 0 = VPOST only read 1 picture (For line buffer overlapping, OlapSel=10) 1 = VPOST read 2 picture (For frame buffer overlapping, OlapSel=01)
[7:4]	**OlapDFSel**	**Overlapping Data Output Data Format Selection** 000 = U Y0 V Y1 001 = Y0 U Y1 V 010 = V Y0 U Y1 011 = Y0 V Y1 U

		100 = Y1 V Y0 U 101 = V Y1 U Y0 110 = Y1 U Y0 V 111 = U Y1 V Y0 Ps. **U** is same as **Cb**; **V** is same as **Cr**. Note: The format should be same as FBDS & YUVBL of LCDCCtl
[1:0]	**OlapSel**	**Overlapping Selection:** 00 = None-Overlapping 01 = F1 vs F2 Overlapping (The format for F2 is fixed to RGB555 which include transparent bit; the 15th bit) 10 = F vs GPU_VRAM Line Buffer(refer the line buffer transparent bit) 11 = reserved **PS.** Only non-interlace mode allowed for QVGA, and interlace mode for VGA when overlappin used. Overlapping size is based on the TVCtl setting. Please refer to other notes below.

Note

There are two options for overlapping:

(1) Two frame buffers, F1 vs F2:

◎ The data format for F1 is optional; But it is fixed to RGB555 format for F2.

◎ The address for F1, F2 and the overlapped output buffer is specified in register '**FSADDR1**', '**FSADDR2**' and '**FSADDR3**', respectively.

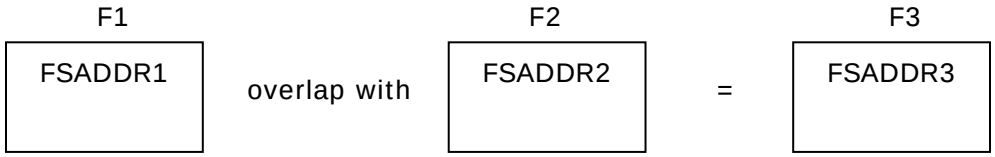
◎ The '**Pic2Sel**' field in '**OlapCtl**' register shouled be set to '**1**' for VPOST for reading 2 frames.

◎ The register '**FDFSel**' decides the data format for F1.

◎ The setting in '**OlapDFSel'** field in '**OlapCtl**' register decides the data format of the overlapped output buffer. Also should set '**FBDS'** & '**YUVBL'** field of '**LCDCCtl'** register with same format.

◎ For storing the overlapped output data into F3, the '**OlapDRM**' field must be set according to your needs (taking a picture or video recording).

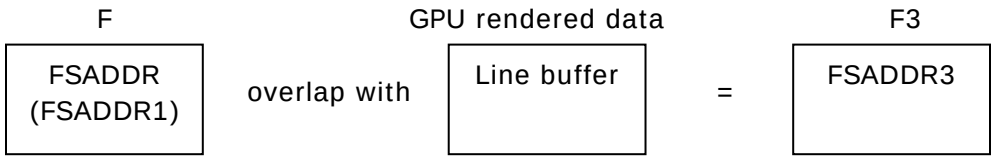
◎ Set '**OlapDR**' field to trigger recording of overlapped data. Then you can use data which stored in F3.



(2) F vs GPU_VRAM Line buffer:

◎ The data format for F1 is optional, and the other source is from GPU rendered data.

◎ The address for F and the overlapped output buffer is specified in register '**FSADDR**' and '**FSADDR3**', respectively.
◎ The '**Pic2Sel**' field in '**OlapCtl**' register shouled be set to '**0**' for VPOST for reading 1 frame.
◎ The register '**FDFSel**' decides the data format for F.
◎ The setting in '**OlapDFSel'** field in '**OlapCtl**' register decides the data format of the overlapped output buffer. Also should set '**FBDS'** & '**YUVBL'** field of '**LCDCCtl'** register with same format.
◎ For storing the overlapped output data into F3, the '**OlapDRM**' field must be set according to your needs (taking a picture or video recording).
◎ Set '**OlapDR**' field to trigger recording of overlapped data. Then you can use data which stored in F3.



NOTE The overlapped priority relationship:

(1) F2 > F1;

(2) Line Buffer > F.

◎ **Frame Buffer Base Address Register 1**

Register	Address	R/W	Description	Reset Value
FSADDR1	0x30	R/W	Frame Buffer Start Address 1	0x0000_0000

31	30	29	28	27	26	25	24
FSADDR1[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FSADDR1[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FSADDR1[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FSADDR1[7:0]							

Bits	Descriptions	
[31:0]	**FSADDR1**	Frame Buffer Start Address 1

◎ **Frame Buffer Base Address Register 2**

Register	Address	R/W	Description	Reset Value
FSADDR2	0x34	R/W	Frame Buffer Start Address 2	0x0000_0000

31	30	29	28	27	26	25	24
FSADDR2[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FSADDR2[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FSADDR2[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FSADDR2[7:0]							

Bits	Descriptions	
[31:0]	**FSADDR2**	Frame Buffer Start Address 2

◎ **Frame Buffer Base Address Register 3**

Register	Address	R/W	Description	Reset Value
FSADDR3	0x38	R/W	Frame Buffer Start Address 3	0x0000_0000

31	30	29	28	27	26	25	24
FSADDR3[31:24]							
23	22	21	20	19	18	17	16
FSADDR3[23:16]							
15	14	13	12	11	10	9	8
FSADDR3[15:8]							
7	6	5	4	3	2	1	0
FSADDR3[7:0]							

Bits	Descriptions	
[31:0]	**FSADDR3**	Frame Buffer Start Address 3

◎ **TvControl Register**

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

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

Bits	Descriptions	
[31]	**TvField**	**Tv Field Status (only read)** 0 = Even Field 1 = Odd Field
[27]	**GPU_SIZE**	**The GPU generation SIZE** **0=QVGA(320x240) 1=VGA(640x480)**
[26]	**TvFFFE**	**TV Flick Free Filter Enable (Interlace Mode)**

<table>
<tr><td></td><td></td><td>1 = enable 0 = disable

Ps. This function is for reducing the flicker phenomenon on TV screen under interlace mode. TVCtl [25:24] is the selection for Flick free filter.</td></tr>
<tr><td>[25:24]</td><td>TvFFFS</td><td>TV Flick Free Filter Selection
<table>
<tr><td>TvFFS</td><td colspan="3">00</td><td colspan="3">01</td><td colspan="3">10</td></tr>
<tr><td rowspan="3">Weight
Ration</td><td>0</td><td>1</td><td>0</td><td>1</td><td>2</td><td>1</td><td>1</td><td>3</td><td>1</td></tr>
<tr><td>0</td><td>2</td><td>0</td><td>2</td><td>4</td><td>2</td><td>1</td><td>4</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>2</td><td>1</td><td>1</td><td>3</td><td>1</td></tr>
</table></td></tr>
<tr><td>[16]</td><td>TvCMM</td><td>TV Color Modulation Method
1 = 27MHz 0 = 13.5MHz</td></tr>
<tr><td>[15:14]</td><td>FBSIZE</td><td>Frame Buffer Size
00 = 320x240(QVGA) 01= 640x240(HVGA)
10 = 640x480(VGA) 11= Reserved</td></tr>
<tr><td>[11:10]</td><td>LCDSrc</td><td>LCD Color Selection Color/Black
00 = Line Buffer 01 = Frame Buffer
10 = Register Setting Color 11 = Internal Color Bar</td></tr>
<tr><td>[9:8]</td><td>TvSrc</td><td>Tv Color Selection Color/Black
00 = Line Buffer 01 = Frame Buffer
10 = Register Setting Color 11 = Internal Color Bar</td></tr>
<tr><td>[6]</td><td>TvLBSA</td><td>Tv Line Buffer Scaling Alograthim (320->640)
1 = Interploation 0 = Duplicate</td></tr>
<tr><td>[5]</td><td>NotchE</td><td>Notch Filter Enable/Disable
1 = Enable 0 = Disable</td></tr>
<tr><td>[4]</td><td>Tvdac</td><td>Tv DAC Enable/Disable
0 = Enable(Normal Run) 1 = Disable(Entering Power Down Mode)</td></tr>
<tr><td>[3]</td><td>TvInter</td><td>Interlance or Non Interlace
0 = non-interlance 1 = Interlace</td></tr>
</table>

[2]	**TvSys**	**Tv System Selection** 0 = NTSC 1 = PAL
[1]	**TvColor**	**Tv Color Selection Color / Black** 0 = Color 1 = Black
[0]	**TvSleep**	**Tv Encoder Enable / Disable** 1 = enable TV Encoder 0 = disable TV Encoder

◎ **IIR Denominator Coefficient Register**

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

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

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

◎ **IIR Numerator Coefficient Register**

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

31	30	29	28	27	26	25	24
Reserved					**IIRB3[8:6]**		

23	22	21	20	19	18	17	16
IIRB3[5:0]						IIRB2[8:7]	
15	14	13	12	11	10	9	8
	IIRB2[6:0]						IIRB1[8]
7	6	5	4	3	2	1	0
IIRB1[7:0]							

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

◎ Color Setting Register

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

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

Bits	Descriptions	
[16]	CMOS_WE	CMOS write line buffer enable: 1 = Enabled ; Write RGB555 data from CMOS sensor to line buffer 0 = Disable the color set value from being written to the line buffer
[15]	ATRE	Alpha Target Blending 1 = Enabled

		0 = Disabled
[14:10]	**Color_R**	Color R Value
[9:5]	**Color_G**	Color G Value
[4:0]	**Color_B**	Color B Value

◎ **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_1280

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	
[31:25]	**FFRHS**	**Line First Request Horizontal Start Pixel**
[24:16]	**LCDHB**	LCD H blank setting for Syn TV Display
[15:8]	**TVDVS**	TV Display Start Line Register
[7:0]	**TVDHS**	TV Display Start Pixel Register

◎ **Sync Horizontal TV TFT Display Control Register**

Register	Address	R/W	Description	Reset Value
SHTFTCtl	0x58	R/W	Sync TV Horizontal TFT Display Start Control Register	0x0100_100bh

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

Bits	Descriptions	
[15:8]	**TFT_HWIDTH**	TFT Display Hsync Width Pixel Counter
[7:0]	**TFT_HDISP**	TFT Display Start Pixel Register from Hsync

◎ **Sync Vertical TV TFT Display Control Register**

Register	Address	R/W	Description	Reset Value
SVTFTCtl	0x5C	R/W	Sync Vertical TV TFT Display Start Control Register	0x0100_010bh

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

Bits	Descriptions	
[23:16]	**TFT_VPWIDTH**	**TFT Vsyn Width data / pixel Clock**
[10:8]	**TFT_VLWIDTH**	TFT Display Hsync Width Line Counter
[7:0]	**TFT_VDISP**	TFT Vsyn Singal offset from Hsyn Edge

◎ **Manual Control Display Line Counter Register**

Register	Address	R/W	Description	Reset Value
MCDLR	0x60	R/W	Manual Control Display Line Counter Register	0x1012_0000

31	30	29	28	27	26	25	24
SHSYOCNT							
23	22	21	20	19	18	17	16
SDVPTR							
15	14	13	12	11	10	9	8
FPT	LPUE	LCS	LBTCS	GPUCS	LCNTSEL	SFRAME	SVPTR[8]
7	6	5	4	3	2	1	0
SVPTR[7:0]							

Bits	Descriptions	
[31:24]	**SHSYOCNT**	**Software Mode Hsyn Pulse width setting**

[23:16]	**SDVPTR**	Display Start Line Register define by customer
[15]	**FPT**	**Field Plot Trigger** Only 0 →1= Trigger Enable 0 → 0, 1→0,1→1 = No Action
[14]	**LPUE**	Line Patial Update Enable 0 = disable 1 = enable
[13]	**LCS**	Line Complete Signal(GPU & VRAM Loader Complete Signal) 0 = underrun 1 = Transfer Complete
[12]	**LBTCS**	Line Buffer Transfer Complete Signal (only read) 0 = underrun 1 = Transfer Complete
[11]	**GPUCS**	GPU Complete Signal (only read) 0 =underun 1 = GPU complete
[10]	**LCNTSEL**	Line Couter selection control by VPOST or CPU 0 = CPU(default) 1 = VPOST
[9]	**SFRAME**	Frame Control define by customer 0 = Software Frame disable 1 = Software Frame Enable
[8:0]	**SVPTR**	Line Counter define by Customer

◎ **TV Contrast Adjust Control Register**

Register	Address	R/W	Description	Reset Value
TvContrast	0x64	R/W	Tv contrast adjust setting register	0x0080_8080h

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	
[23:16]	Cr_contrast	Cr compoment contrast adjust
[15:8]	Cb_contrast	Cb compoment contrast adjust
[7:0]	Y_contrast	Y compoment contrast adjust

◎ TV Bright Adjust Control Register

Register	Address	R/W	Description	Reset Value
TvBright	0x68	R/W	Tv Bright adjust setting register	0x0000_0000h

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

[23:16]	**Cr_gain**	Cr compoment bright adjust
[15:8]	**Cb_gain**	Cb compoment bright adjust
[7:0]	**Y_bright**	Y compoment bright adjust

Y Adjust Equation
Y_Adj = (Y-16)*Y_Contrast+16+Y_Bright
Y_Contrast is bit7 and is an interger, bit6~0 is a fix point
Y_Bright is a signed interger (-127~127)

Cb Adjust Equtaion
Cb_Adj = (Cb-128)*Cb_Contrast+128+Cb_Bright
Cb_Contast is bit7 and is an interger, bit6~0 is a fix point
Cb_Bright is a signed interger (-127~127)

Cr Adjust Equtaion
Cr_Adj = (Cr-128)*Cr_Contrast+128+Cr_Bright
Cr_Contast is bit7 and is an integer, bit6~0 is a fix point
Cr_Bright is a signed integer (-127~127)

◎ EBI Start Address Register

Register	Address	R/W	Description	Reset Value
ESADDR	0x6c	R/W	EBI Start Address	0x2000_0000

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

Bits	Descriptions	
[31:0]	**ESADDR**	Frame Buffer Start Address

◎ Manual Control Display Line Counter Register1

Register	Address	R/W	Description	Reset Value

MCDLR1	0x70	R/W	Manual Control Display Line Counter Register1	0x0100_EF00

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

Bits	Descriptions	
[31]	**Reserved**	**Reserved**
[30]	**LSE**	**Line Stop Enable** 1 = Hardware Line Counter will continue counting to next line 0 = Hardware Line Counter will stop when line complete(plot, sprite & data transfer all complete)
[29:20]	**UALC**	**Update Active Line Counter(Sprite Active Line)** **(only read)**
[18:10]	**UAEL**	Update Active End Line
[9:0]	**UASL**	Update Active Start Line

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

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

4.8 General DMA Controller

4.8.1 Overview and Features

The W55VA91 has a two-channel general DMA controller, called the GDMA. The two-channel GDMA performs the following data transfers without the CPU intervention:

- Memory-to-memory (memory to/from memory)
- Memory–to–IO
- IO- to -memory

The on-chip GDMA can be started by the software. Software can also be used to restart the GDMA operation after it has been stopped. The CPU can recognize the completion of a GDMA operation by software polling or when it receives an internal GDMA interrupt. The GDMA controller can increment source or destination address, decrement them as well, and conduct 8-bit (byte), 16-bit (half-word), or 32-bit (word) data transfers.

The GDMA includes the following features

- AMBA AHB compliant
- Supports 8-data burst mode to boost performance

4.8.2 GDMA Functional Description

The GDMA directly transfers data between source and destination. The GDMA starts to transfer data after it receives service requests from software. When the entire data have been transferred completely, the GDMA becomes idle. Nevertheless, if another transfer is needed, then the GDMA must be programmed again. There are two transfer modes:

Single Mode

Single mode requires a GDMA request for each data transfer. A GDMA software request causes one byte, one half-word, or one word to transfer if the 8-data burst mode is disabled, or four times of transfer width is the 8-data burst mode is enabled.

Since there is no external request source in W55VA91, once GDMA be enabled in software mode, the operation is continued until the transfer count register reaches zero. Therefore single mode is just operated as block mode.

Block Mode

The assertion of a single GDMA request causes all of the data to be transferred in a single operation. The GDMA transfer is completed when the current transfer count register reaches zero.

4.8.3 GDMA 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
GDMA_BA = 0xFFF0_3000				

Channel 0				
GDMA_CTL0	GDMA_BA+0x000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_SRCB0	GDMA_BA+0x004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_DSTB0	GDMA_BA+0x008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_TCNT0	GDMA_BA+0x00C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_CSRC0	GDMA_BA+0x010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CDST0	GDMA_BA+0x014	R	Channel 0 Current Destination Address Register	0x0000_0000
GDMA_CTCNT0	GDMA_BA+0x018	R	Channel 0 Current Transfer Count Register	0x0000_0000
Channel 1				
GDMA_CTL1	GDMA_BA+0x020	R/W	Channel 1 Control Register	0x0000_0000
GDMA_SRCB1	GDMA_BA+0x024	R/W	Channel 1 Source Base Address Register	0x0000_0000
GDMA_DSTB1	GDMA_BA+0x028	R/W	Channel 1 Destination Base Address Register	0x0000_0000
GDMA_TCNT1	GDMA_BA+0x02C	R/W	Channel 1 Transfer Count Register	0x0000_0000
GDMA_CSRC1	GDMA_BA+0x030	R	Channel 1 Current Source Address Register	0x0000_0000
GDMA_CDST1	GDMA_BA+0x034	R	Channel 1 Current Destination Address Register	0x0000_0000
GDMA_CTCNT1	GDMA_BA+0x038	R	Channel 1 Current Transfer Count Register	0x0000_0000
Internal Buffer & Interrupt				
GDMA_INTBUF0	GDMA_BA + 0x80	R	GDMA Internal Buffer Word 0	0x0000_0000
GDMA_INTBUF1	GDMA_BA + 0x84	R	GDMA Internal Buffer Word 1	0x0000_0000
GDMA_INTBUF2	GDMA_BA + 0x88	R	GDMA Internal Buffer Word 2	0x0000_0000
GDMA_INTBUF3	GDMA_BA + 0x8C	R	GDMA Internal Buffer Word 3	0x0000_0000
GDMA_INTBUF4	GDMA_BA + 0x90	R	GDMA Internal Buffer Word 4	0x0000_0000
GDMA_INTBUF5	GDMA_BA + 0x94	R	GDMA Internal Buffer Word 5	0x0000_0000
GDMA_INTBUF6	GDMA_BA + 0x98	R	GDMA Internal Buffer Word 6	0x0000_0000
GDMA_INTBUF7	GDMA_BA + 0x9C	R	GDMA Internal Buffer Word 7	0x0000_0000
GDMA_INTCS	GDMA_BA + 0xA0	R/W	Interrupt Control and Status Register (2 Chs)	0x0000_0000

4.8.4 GDMA Register Description

Channel 0 / 1 Control Register (GDMA_CTL0, GDMA_CTL1)

Register	Address	R / W	Description	Reset Value
GDMA_CTL0	GDMA_BA+0x000	R/W	Channel 0 Control Register	0x0000_0000
GDMA_CTL1	GDMA_BA+0x020	R/W	Channel 1 Control Register	0x0000_0000

RESERVED 31	SABNDERR 30	DABNDERR 29	RESERVED 28	AUTOIEN 27	RESERVED 26	BLOCK 25	SOFTREQ 24
15	14	13	12	11	10	9	8
RESERVED 23	RESERVED 22	21	TWS 20	SBMS 19	RESERVED 18	BME 17	RESERVED 16

7	6	5	4	3	2	1	0
SAFIX	DAFIX	SADIR	DADIR	GDMAMS		RESERVED	GDMAEN

Bits	Descriptions	
[22]	**SABNDERR**	**Source Address Boundary Alignment Error Flag** If TWS [13:12]=10, GDMA_SRCB [1:0] should be 00 If TWS [13:12]=01, GDMA_SRCB [0] should be 0 The address boundary alignment should be depended on TWS [13:12]. 0 = the GDMA_SRCB is on the boundary alignment. 1 = the GDMA_SRCB not on the boundary alignment The SABNDERR register bits just can be read only.
[21]	**DABNDERR**	**Destination Address Boundary Alignment Error Flag** If TWS [13:12]=10, GDMA_DSTB [1:0] should be 00 If TWS [13:12]=01, GDMA_DSTB [0] should be 0 The address boundary alignment should be depended on TWS [13:12]. 0 = the GDMA_DSTB is on the boundary alignment. 1 = the GDMA_DSTB not on the boundary alignment The DABNDERR register bits just can be read only.
[19]	**AUTOIEN**	**Auto initialization Enable** 0 = Disables auto initialization 1 = Enables auto initialization, the GDMA_CSRC0/1,GDMA_CDST0/1,and GDMA_CTCNT0/1 registers are updated by the GDMA_SRCB0/1, GDMA_DST0/1, and GDMA_TCNT0/1 registers automatically when transfer is complete.
[17]	**BLOCK**	**Bus Lock** 0 = Unlocks the bus during the period of transfer 1 = locks the bus during the period of transfer
[16]	**SOFTREQ**	**Software Triggered GDMA Request** Software can request the GDMA transfer service by setting this bit to 1. This bit is automatically cleared by hardware when the transfer is completed. This bit is available only while GDMAMS [3:2] register bits are set on software mode (memory to memory).
[13:12]	**TWS**	**Transfer Width Select** 00 = One byte (8 bits) is transferred for every GDMA operation 01 = One half-word (16 bits) is transferred for every GDMA operation 10 = One word (32 bits) is transferred for every GDMA operation 11 = Reserved The GDMA_SCRB and GDMA_DSTB should be alignment under the TWS selection
[11]	**SBMS**	**Single/Block Mode Select** 0 = Selects single mode. It requires an external GDMA request for every incurring GDMA operation. 1 = Selects block mode. It requires a single external GDMA request during the atomic GDMA operation. An atomic GDMA operation is defined as the sequence of GDMA operations until the transfer count register reaches zero. Note: For W55VA91, there is no external request source, so the selection between signal/block doesn't make any difference for GDMA operation.

[9]	**BME**	**Burst Mode Enable** 0 = Disables the 8-data burst mode 1 = Enables the 8-data burst mode If there are 8 words to be transferred, and BME [9]=1, the GDMA_TCNT should be 0x01; However, if BME [9]=0, the GDMA_TCNT should be 0x08.
[7]	**SAFIX**	**Source Address Fixed** 0 = Source address is changed during the GDMA operation 1 = Do not change the destination address during the GDMA operation. This feature can be used when data were transferred from a single source to multiple destinations.
[6]	**DAFIX**	**Destination Address Fixed** 0 = Destination address is changed during the GDMA operation 1 = Do not change the destination address during the GDMA operation. This feature can be used when data were transferred from multiple sources to a single destination.
[5]	**SADIR**	**Source Address Direction** 0 = Source address is incremented successively 1 = Source address is decremented successively
[4]	**DADIR**	**Destination Address Direction** 0 = Destination address is incremented successively 1 = Destination address is decremented successively
[3:2]	**GDMAMS**	**GDMA Mode Select** 00 = Software mode (memory-to-memory) only.
[0]	**GDMAEN**	**GDMA Enable** 0 = Disables the GDMA operation 1 = Enables the GDMA operation; this bit will be clear automatically when the transfer is complete on AUTOIEN [19] register bit is on Disable mode.

Channel 0 / 1 Source Base Address Register (GDMA_SRCB0, GDMA_SRCB1)

Register	Address	R/W	Description	Reset Value
GDMA_SRCB0	GDMA_BA+0x004	R/W	Channel 0 Source Base Address Register	0x0000_0000
GDMA_SRCB1	GDMA_BA+0x024	R/W	Channel 1 Source Base Address Register	0x0000_0000

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

Bits	Descriptions

[31:0]	**SRC_BASE_ADDR**	**32-bit Source Base Address** The GDMA channel starts reading its data from the source address as defined in this source base address register.

Channel 0 / 1 Destination Base Address Register (GDMA_DSTB0, GDMA_DSTB1)

Register	Address	R / W	Description	Reset Value
GDMA_DSTB0	GDMA_BA+0x008	R/W	Channel 0 Destination Base Address Register	0x0000_0000
GDMA_DSTB1	GDMA_BA+0x028	R/W	Channel 1 Destination Base Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**DST_BASE_ADDR**	**32-bit Destination Base Address** The GDMA channel starts writing its data to the destination address as defined in this destination base address register. During a block transfer, the GDMA determines successive destination addresses by adding to or subtracting from the destination base address.

Channel 0 / 1 Transfer Count Register (GDMA_TCNT0, GDMA_TCNT1)

Register	Address	R / W	Description	Reset Value
GDMA_TCNT0	GDMA_BA+0x00C	R/W	Channel 0 Transfer Count Register	0x0000_0000
GDMA_TCNT1	GDMA_BA+0x02C	R/W	Channel 1 Transfer Count Register	0x0000_0000

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

Bits	Descriptions	
[23:0]	**TFR_CNT**	**24-bit Transfer Count** The TFR_CNT represents the required number of GDMA transfers. The maximum transfer count is 16M –1.

Channel 0 / 1 Current Source Register (GDMA_CSRC0, GDMA_CSRC1)

Register	Address	R / W	Description	Reset Value
GDMA_CSRC0	GDMA_BA+0x010	R	Channel 0 Current Source Address Register	0x0000_0000
GDMA_CSRC1	GDMA_BA+0x030	R	Channel 1 Current Source Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CURRENT_SRC_ADDR**	**32-bit Current Source Address** The CURRENT_SRC_ADDR indicates the source address where the GDMA transfer is just occurring. During a block transfer, the GDMA determines the successive source addresses by adding to or subtracting from the source base address. Depending on the settings you make to the control register, the current source address will remain the same or will be incremented or decremented.

Channel 0 / 1 Current Destination Register (GDMA_CDST0, GDMA_CDST1)

Register	Address	R / W	Description	Reset Value
GDMA_CDST0	GDMA_BA+0x014	R	Channel 0 Current Destination Address Register	0x0000_0000
GDMA_CDST1	GDMA_BA+0x034	R	Channel 1 Current Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CURRENT_DST_ADDR [31:24]							
23	22	21	20	19	18	17	16
CURRENT_DST_ADDR [23:16]							
15	14	13	12	11	10	9	8

CURRENT_DST_ADDR [15:8]							
7	6	5	4	3	2	1	0
CURRENT_DST_ADDR [7:0]							

Bits	Descriptions	
[31:0]	**CURRENT_DST_ADDR**	**32-bit Current Destination Address** The CURRENT_DST_ADDR indicates the destination address where the GDMA transfer is just occurring. During a block transfer, the GDMA determines the successive destination addresses by adding to or subtracting from the destination base address. Depending on the settings you make to the control register, the current destination address will remain the same or will be incremented or decremented.

Channel 0 / 1 Current Transfer Count Register (GDMA_CTCNT0, GDMA_CTCNT1)

Register	Address	R / W	Description	Reset Value
GDMA_CTCNT0	GDMA_BA+0x018	R	Channel 0 Current Transfer Count Register	0x0000_0000
GDMA_CTCNT1	GDMA_BA+0x038	R	Channel 1 Current Transfer Count Register	0x0000_0000

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

Bits	Descriptions	
[23:0]	**CURRENT_TFR_CNT**	**Current Transfer Count** The Current transfer count register indicates the number of transfer being performed.

Channel 0 / 1 GDMA Internal Buffer Register (GDMA_INTBUF0 / 1)

Softeware can set the [16] bit of GDMA_INTCS to select channels and watch the value which has read from memory.

Register	Address	R / W	Description	Reset Value
GDMA_INTBUF0	GDMA_BA+0x80	R	GDMA Internal Buffer Word 0	0x0000_0000
GDMA_INTBUF1	GDMA_BA+0x84	R	GDMA Internal Buffer Word 1	0x0000_0000
GDMA_INTBUF2	GDMA_BA+0x88	R	GDMA Internal Buffer Word 2	0x0000_0000

GDMA_INTBUF3	GDMA_BA+0x8C	R	GDMA Internal Buffer Word 3	0x0000_0000
GDMA_INTBUF4	GDMA_BA+0x90	R	GDMA Internal Buffer Word 4	0x0000_0000
GDMA_INTBUF5	GDMA_BA+0x94	R	GDMA Internal Buffer Word 5	0x0000_0000
GDMA_INTBUF6	GDMA_BA+0x98	R	GDMA Internal Buffer Word 6	0x0000_0000
GDMA_INTBUF7	GDMA_BA+0x9C	R	GDMA Internal Buffer Word 7	0x0000_0000

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

Bits	Descriptions	
[31:0]	DATA_BUFFER	**Internal Buffer Register** **Each channel has its own internal buffer from Word 0 to Word 7. The [17-16] bit of GDMA_INTCS will determine the values of channels mapping to GDMA_INTBUF0~7.** **NOTE: The GDMA_INTBUF0~7 are available when burst mode used, otherwise, only the GDMA_INTBUF0 available.**

Channel 0/1 GDMA Interrupt Control and Status Register (GDMA_INTCS)

Register	Address	R/W	Description	Reset Value
GDMA_INTCS	GDMA_BA+0xA0	R/W	Interrupt Control and Status Register (2 Chs)	0x0000_0000

31	30	29	28	27	26	25	24
RESERVE							
23	22	21	20	19	18	17	16
RESERVE							BUF_RD_SEL
15	14	13	12	11	10	9	8
RESERVED				TERR1F	TC1F	TERR0F	TC0F
7	6	5	4	3	2	1	0
RESERVED				TERR1EN	TC1EN	TERR0EN	TC0EN

Bits	Descriptions	
[16]	**BUF_RD_SEL**	**Internal Buffer Read Select** 00 = Read Internal Buffer for Channel 0 01 = Read Internal Buffer for Channel 1 10 = RESERVED 11 = RESERVED
[11]	**TERR1F**	**Channel 1 Transfer Error** O = No error occurs 1 = Hardware sets this bit on a GDMA transfer failure This bit will be cleared when write logic 1. Transfer error will generate GDMA interrupt
[10]	**TC1F**	**Channel 1 Terminal Count** 0 = Channel does not expire 1 = Channel expires; this bit is set only by GDMA hardware, and clear by software to write logic 1. TC1 is the GDMA interrupt flag. TC1 or GDMATERR1 will generate interrupt
[9]	**TERR0F**	**Channel 0 Transfer Error** O = No error occurs 1 = Hardware sets this bit on a GDMA transfer failure This bit will be cleared when write logic 1. Transfer error will generate GDMA interrupt
[8]	**TC0F**	**Channel 0 Terminal Count** 0 = Channel does not expire 1 = Channel expires; this bit is set only by GDMA hardware, and clear by software to write logic 1. TC0 is the GDMA interrupt flag. TC0 or GDMATERR0 will generate interrupt
[3]	**TEER1EN**	**Channel 1 Interrupt Enable for Transfer Error** 0 = Disable Interrupt 1 = Enable Interrupt
[2]	**TC1EN**	**Channel 1 Interrupt Enable for Terminal Count** 0 = Disable Interrupt 1 = Enable Interrupt
[1]	**TEER0EN**	**Channel 0 Interrupt Enable for Transfer Error** 0 = Disable Interrupt 1 = Enable Interrupt
[0]	**TC0EN**	**Channel 0 Interrupt Enable for Terminal Count** 0 = Disable Interrupt 1 = Enable Interrupt

4.9 2-D Graphic Engine

A line buffer based 2D Graphics Engine is specially designed to improve the performance of graphics processing. It can accelerate the rotation, up-scaling, and down-scaling of background (BG) and sprite (SP) in a rastering scan line. Also, a perspective projection function is provided for the background, during background rotating and scaling simultaneously.

Basically, it supports the rectangular object scaling up/down and rotation with an arbitrary angle for the background and sprites. Especially, a perspective projection function is dedicated for the BG during 2D scaling and rotating with a virtual Z-depth programmed by users.

For scaling up/down, both the horizontal and vertical scaling factors are programmable to resize the objects, with up-scaling ratio to be 1X ~ 3X + (N/M), and down -scaling ratio to be 1/M ~ N/M.

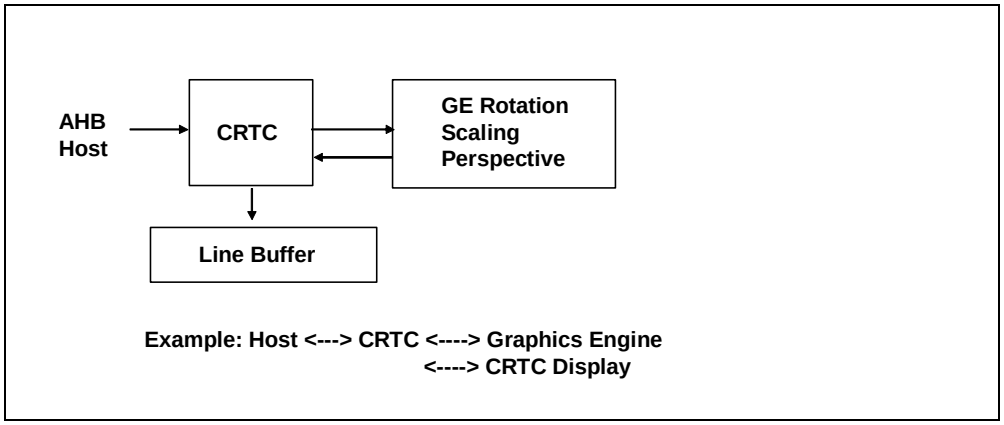


Figure 4.9-1 2-D Graphics Engine Architecture

4.9.1 Functional Description

4.9.1.1 Rotation and Scaling Up / Down

The main function is to support the 2D rotation and up/down scaling of any rectangular object during Graphics Engine operation. For the 2D rotation, it rotates with arbitrary degrees for the BG and sprites, and it also supports the flop (mirror) and the flip (up-side-down) orientations for the BG and sprites.

In background and sprite scaling up/down function, both programmable horizontal and vertical N/M scaling up/down factors are provided to resize the objects. For scaling-up, 1X ~ 3X + (N/M), or scaling-down, 1/M ~ N/M, the value of N can be equal or less than M.

This 2D graphics engine can rotate any rectangular object to the eight different octants, that is, 8 different sections by the 3-bit octant code as specified in the COMMAND register. The X-Y coordinates can be divided into 8 rotation sections, 45 degrees per section. These 8 rotation sections locate at the different octants as shown in Figure 4.9-2.

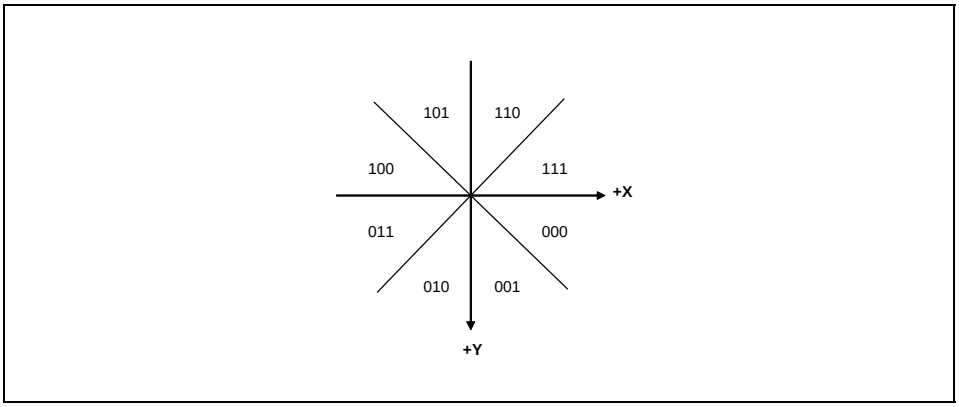


Figure 4.9-2 Rotation sections

4.9.1.2 Perspective Projection

We use the one-point perspective projection to implement the 3D-like effect. In principle, the perspective projections of any set of parallel lines that are not parallel to the projection plane converge to a vanishing point. In the real 3D world, those parallel lines meet or cross only at infinity. But for 2D graphics, we should choose a convenient position in the X-Y coordinate to simulate this behavior. So that, for the 2D graphics, the vanishing point can be thought of as a point that is located at some place or corner of the screen, even for a display screen with a resolution as small as 320x240.

4.9.2 2-D Graphic 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	Description	Reset Value
GE_BA = 0xFFF0_5000				
BG_RSCMD	GE_BA+0x000	R/W	BG Rotation/Scaling Command Control Register	0x0000_0000
BG_SSYX	GE_BA+0x004	R/W	BG Source Start Y/X Coordinates	0x0000_0000
BG_RRYX	GE_BA+0x008	R/W	BG Source Rotate Ref. Point Y/X Coordinates	0x0000_0000
BG_DRYX	GE_BA+0x00C	R/W	BG Destination Reference Point Y/X Coordinates	0x0000_0000
BG_VHSF	GE_BA+0x010	R/W	BG Fractional Part Vertical and Horizontal N/M	0x0000_0000
BG_HWYX	GE_BA+0x014	R/W	BG Height/Width Y/X Dimensions	0x0000_0000
GE_BGSTS	GE_BA+0x018	R	(DBG) Graphics Engine Background Status	0xXXXX_XXXX
GE_DEBUG	GE_BA+0x01C	R/W	(DBG) Graphics Engine Debug Mode Ctl Register	0x0000_0000
SP_RSCMD	GE_BA+0x020	R/W	(DBG) Sprite Rotation/Scaling Control Register	0x0000_0000
SP_SSXY	GE_BA+0x024	R/W	(DBG) Sprite Pattern Start X/Y Coordinates	0x0000_0000
SP_RRYX	GE_BA+0x028	R/W	(DBG) SP Pattern Rotate Ref. Point Y/X Coordinates	0x0000_0000
SP_DRYX	GE_BA+0x02C	R/W	(DBG) Screen Display Ref. Point Y/X Coordinates	0x0000_0000
SP_VHSF	GE_BA+0x030	R/W	(DBG) Sprite Fractional Vertical and Horizontal N/M	0x0000_0000
GE_SPSTS	GE_BA+0x034	R	(DBG) Graphics Engine Sprite Status	0xXXXX_XXXX
GE_DVID	GE_BA+0x03C	R	(DBG) Graphics Engine Date/Version ID Register	0x0915_E500

DBG_BGYXIN	GE_BA+0x040	R/W	(DBG) Background Y/X Input Coordinates	0xXXXX_XXXX
DBG_BGYXOUT	GE_BA+0x044	R	(DBG) Background Y/X Output Coordinates	0xXXXX_XXXX
DBG_SPYXIN	GE_BA+0x048	R/W	(DBG) Sprite Y/X Input Coordinates	0xXXXX_XXXX
DBG_SPYXOUT	GE_BA+0x04C	R	(DBG) Sprite Y/X Output Coordinates	0xXXXX_XXXX

4.9.3 2-D Graphics Engine Control Registers

BG Rotation and Scaling Command Control Register

Register	Address	R/W	Description	Reset Value
BG_RSCMD	GE_BA+0x000	R/W	BG Rotation/Scaling Command Control Register	0x0000_0000

31	30	29	28	27	26	25	24
DEGREE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Z DEPTH							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VY_TUNE	**HX_TUNE**	**VY_FLIP**	**HX_FLIP**	**XY OCTANT**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**A_CROSS**	**PERSPECT**	**INTEGER SCALE**			

Bits	Descriptions	
[31:24]	**DEGREE**	**Rotation Degree** This is the rotation angle specified for the rotated objects or sprites. It defines the 8-bit value from 0x00 to 0xFF by partitioning 45 degrees in a rotation section. That is, 0x00 is the lowest degree and 0xFF is the largest degree or 45 °. The X-Y coordinates can be divided into 8 rotation sections, with 45 degrees per section.
[23:16]	**Z DEPTH**	**Z Depth of Perspective Projection** It specifies the 8-bit Z depth when perspective projection is active.
[15]	**VY_TUNE**	**VY_TUNE is used for the mathematic common factor reduction.** 0= No effect 1= If BG VSF_N is equal to BG VSF_M, both VSF_N and VSF_M are reduced and normalized to 0x01.
[14]	**HX_TUNE**	**HX_TUNE is used for the mathematic common factor reduction.** 0= No effect 1= If BG HSF_N is equal to BG HSF_M, both HSF_N and HSF_M are reduced and normalized to 0x01.
[13]	**VY_FLIP**	**Vertical Direction Flip Enable** It enables the vertical direction flip or up-side-down of an object, with reference to the BG Source_Rotate_Reference_Point.
[12]	**HX_FLIP**	**Horizontal Direction Flip Enable** It enables the horizontal direction flip or mirror mapping of an object, with reference to the BG Source_Rotate_Reference_Point.

[10:8]	**YX Octant**	**YX Octant** It specifies the octant code when BG is rotating. 000 = Rotation Section 0 001 = Rotation Section 1 010 = Rotation Section 2 011 = Rotation Section 3 100 = Rotation Section 4 101 = Rotation Section 5 110 = Rotation Section 6 111 = Rotation Section 7
[5]	**A_CROSS**	**Anti_Crossbar Control** 0 = No effect 1 = Anti_Crossbar is active
[4]	**PERSPECT**	**Perspective Projection** 0 = BG perspective projection is not active 1 = BG perspective projection is active
[3:0]	**INTEGER SCALE**	**Integer Part of Scaling Factors** 00 = Scale down N/M (N <= M) 01 = Scale up 1X ~ (1X+N/M) 10 = Scale up 2X ~ (2X+N/M) 11 = Scale up 3X ~ (3X+N/M)

BG Source Start Y/X Coordinates Register

Register	Address	R/W	Description	Reset Value
BG_SSYX	GE_BA+0x004	R/W	BG Source Start Y/X Coordinates	0x0000_0000

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

Bits	Descriptions	
[26:16]	**BG Source Start Y**	**11-bit Source Start Y** This register specifies the source start Y by pixels.
[10:0]	**BG Source Start X**	**11-bit Source Start X** This register specifies the source start X by pixels.

BG Source Rotate Reference Point Y/X Coordinates Register

Register	Address	R/W	Description	Reset Value
BG_RRYX	GE_BA+0x008	R/W	BG Source Rotate Reference Point Y/X Coordinates	0x0000_0000

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

Bits	Descriptions	
[26:16]	BG Rotate Reference Y	**11-bit Source Rotate Reference Y** In BG rotation, it specifies the source reference point Y by pixels.
[10:0]	BG Rotate Reference X	**11-bit Source Rotate Reference X** In BG rotation, it specifies the source reference point X by pixels.

BG Destination Reference Point Y/X Coordinates Register

Register	Address	R/W	Description	Reset Value
BG_DRYX	GE_BA+0x00C	R/W	BG Destination Reference Point Y/X Coordinates.	0x0000_0000

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

Bits	Descriptions	
[26:16]	BG Destination Reference Y	**11-bit Destination Reference Y** This register specifies the destination reference Y by pixels.
[10:0]	BG Destination Reference X	**11-bit Destination Reference X** This register specifies the destination reference X by pixels.

BG Fractional Part Vertical/Horizontal Scaling Factors

Register	Address	R/W	Description	Reset Value
BG_VHSF	GE_BA+0x010	R/W	BG Fractional Part Vertical and Horizontal N/M	0x0000_0000

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

Bits	Descriptions	
[31:24]	**BG VSF_N**	**8-bit Vertical N Scaling Factor** An 8-bit value specifies the numerator part (N) of the vertical scaling factor in graphics engine. The output image height will be equal to the input image height x N / M. ***The value of N must be equal or less than M.***
[23:16]	**BG VSF_M**	**8-bit Vertical M Scaling Factor** An 8-bit value specifies the denominator part (M) of the vertical scaling factor in graphics engine. The output image height will be equal to the input image height x N / M. ***The value of N must be equal or less than M.***
[15:8]	**BG HSF_N**	**8-bit Horizontal N Scaling Factor** An 8-bit value specifies the numerator part (N) of the horizontal scaling factor in graphics engine. The output image width will be equal to the input image width x N / M. ***The value of N must be equal or less than M.***
[7:0]	**BG HSF_M**	**8-bit Horizontal M Scaling Factor** An 8-bit value specifies the denominator part (M) of the horizontal scaling factor in graphics engine. The output image width will be equal to the input image width x N / M. ***The value of N must be equal or less than M.***

BG Height/Width Y/X Dimensions Register

Register	Address	R/W	Description	Reset Value
BG_HWYX	GE_BA+0x014	R/W	BG Height/Width Y/X Dimensions	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					BG Height Y [10:8]		
23	22	21	20	19	18	17	16
Reserved			BG Height Y [7:0]		BG Width X [10:8]		
7	6	5	4	3	2	1	0

BG Width X [7:0]

Bits	Descriptions	
[26:16]	**Height Dimension Y**	**11-bit Height Dimension Y** This register specifies the height of rectangle object by pixels.
[10:0]	**Width Dimension X**	**11-bit Width Dimension X** This register specifies the width of rectangle object by pixels.

(Debug Port) Graphics Engine Status of sin/cos Table Register

Register	Address	R/W	Description	Reset Value
GE_BGSTS	GE_BA+0x018	R/W	Graphics Engine Status of sin/cos Tables	0x0000_00FF

31	30	29	28	27	26	25	24
Sign Bit Extension of sine(θ)							
23	22	21	20	19	18	17	16
Magnitude of sine(θ)							
15	14	13	12	11	10	9	8
Sign Bit Extension of cosine(θ)							
7	6	5	4	3	2	1	0
Magnitude of cosine(θ)							

Bits	Descriptions	
[31:24]	**Status**	**8-bit Sign Bit Extension of sine(θ)**
[23:16]	**Status**	**8-bit Absolute Value or Magnitude of sine(θ)**
[15:8]	**Status**	**8-bit Sign Bit Extension of cosine(θ)**
[7:0]	**Status**	**8-bit Absolute Value or Magnitude of cosine(θ)**

(Debug Port) Graphics Engine Debug Mode Control Register

Register	Address	R/W	Description	Reset Value
GE_DEBUG	GE_BA+0x01C	R/W	Graphics Engine Debug Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
ScanLine Counter							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		R/T Mode		Switch	Reserved		
7	6	5	4	3	2	1	0
Reserved							DEBUG

Bits	Descriptions	
[31:24]	**ScanLine Counter**	**8-bit ScanLine Counter** This is the quasi-perspective angle specified for the rotated object.
[13:12]	**R/T Mode**	**Round/Truncate Mode** 00 = Selectively Rounding/Truncating in GE 01 = Force Arithmetic Truncating in GE 10 = Force Arithmetic Rounding in GE 11 = Forec Arithmetic Rounding in GE
[11]	**Switch**	**Switch the 8-bit ScanLine Counter.** 1 = Programmable ScanLine Counter. 0 = Fixed ScanLine Counter. It is fixed at decimal value 240.
[0]	**DEBUG**	**Enter the Debug Mode.** 1 = Debug Mode. 0 = Normal Mode.

(Debug Port) Sprite Rotation and Scaling Command Control Register

Register	Address	R/W	Description	Reset Value
SP_RSCMD	GE_BA+0x020	R/W	Graphics Engine Sprite Rotation/Scaling Control Register	0x0000_0000

31	30	29	28	27	26	25	24
DEGREE							
23	22	21	20	19	18	17	16
SPRITE HEIGHT SIZE				SPRITE WIDTH SIZE			
15	14	13	12	11	10	9	8
	Reserved	VY_FLIP	HX_FLIP	XY OCTANT			
7	6	5	4	3	2	1	0
Reserved				INTEGER SCALE			

Bits	Descriptions	
[31:24]	**DEGREE**	**Rotation Degree** This is the rotation angle specified for the rotated objects or sprites. It defines the 8-bit value from 0x00 to 0xFF by partitioning 45 degrees in a rotation section. That is, 0x00 is the lowest degree and 0xFF is the largest degree or 45 °. The X-Y coordinates can be divided into 8 rotation sections, with 45 degrees per section.
[23:20]	**SPRITE HEIGHT SIZE**	**Source Sprite Height Size** 00 = Sprite height size is 8 pixels 01 = Sprite height size is 16 pixels 10 = Sprite height size is 32 pixels 11 = Sprite height size is 64 pixels
[18:16]	**SPRITE WIDTH SIZE**	**Source Sprite Width Size** 00 = Sprite width size is 8 pixels 01 = Sprite width size is 16 pixels 10 = Sprite width size is 32 pixels

		11 = Sprite width size is 64 pixels
[13]	**VY_FLIP**	**Vertical Direction Flip Enable** It enables the vertical direction flip or up-side-down of an object, with reference to the SP Source_Rotate_Reference_Point.
[12]	**HX_FLIP**	**Horizontal Direction Flip Enable** It enables the horizontal direction flip or mirror mapping of an object, with reference to the SP Source_Rotate_Reference_Point.
[11:8]	**YX Octant**	**YX Octant** It specifies the octant code when a sprite is rotating. 000 = Rotation Section 0 001 = Rotation Section 1 010 = Rotation Section 2 011 = Rotation Section 3 100 = Rotation Section 4 101 = Rotation Section 5 110 = Rotation Section 6 111 = Rotation Section 7
[3:0]	**INTEGER SCALE**	**Integer Part of Scaling Factor** 00 = Scale down N/M (N <= M) 01 = Scale up 1X ~ (1X+N/M) 10 = Scale up 2X ~ (2X+N/M) 11 = Scale up 3X ~ (3X+N/M)

(Debug Port) Sprite Pattern Start X/Y Coordinates Register

Register	Address	R/W	Description	Reset Value
SP_SSXY	GE_BA+0x024	R/W	Sprite Pattern Start X/Y Coordinates	0x0000_0000

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

Bits	Descriptions	
[26:16]	**SP Pattern Start Y**	**11-bit Sprite Pattern Start Y** This register specifies the source start Y by pixels.
[10:0]	**SP Pattern Start X**	**11-bit Sprite Pattern Start X** This register specifies the source start X by pixels.

(Debug Port) Sprite Pattern Rotate Reference Point Y/X Coordinates Register

Register	Address	R/W	Description	Reset Value
SP_RRYX	GE_BA+0x028	R/W	Sprite Pat Rotate Reference Point Y/X Coordinates	0x0000_0000

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

Bits	Descriptions	
[26:16]	**SP Rotate Reference Y**	**11-bit SP Pattern Rotate Reference Y** For SP's rotation, it specifies the reference point Y by pixels.
[10:0]	**SP Rotate Reference X**	**11-bit SP Pattern Rotate Reference X** For SP's rotation, it specifies the reference point X by pixels.

(Debug Port) Screen Display Reference Point Y/X Coordinates Register

Register	Address	R/W	Description	Reset Value
SP_DRYX	GE_BA+0x02C	R/W	Screen Display Reference Point Y/X Coordinates	0x0000_0000

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

Bits	Descriptions	
[26:16]	**Screen Display Reference Y**	**11-bit SP Destination Reference Y** In SP rotation, it specifies the destination reference Y by pixels.
[10:0]	**Screen Display Reference X**	**11-bit SP Destination Reference X** In SP rotation, it specifies the destination reference X by pixels.

(Debug Port) Sprite Fractional Part Vertical/Horizontal Scaling Factors

Register	Address	R/W	Description	Reset Value
SP_VHSF	GE_BA+0x030	R/W	Sprite Fractional Part Vertical and Horizontal N/M	0x0000_0000

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

Bits	Descriptions	
[31:24]	**SP VSF_N**	**8-bit Vertical N Scaling Factor** An 8-bit value specifies the numerator part (N) of the vertical scaling factor in graphics engine. The output image height will be equal to the input image height x N / M. ***The value of N must be equal or less than M.***
[23:16]	**SP VSF_M**	**8-bit Vertical M Scaling Factor** An 8-bit value specifies the denominator part (M) of the vertical scaling factor in graphics engine. The output image height will be equal to the input image height x N / M. ***The value of N must be equal or less than M.***
[15:8]	**SP HSF_N**	**8-bit Horizontal N Scaling Factor** An 8-bit value specifies the numerator part (N) of the horizontal scaling factor in graphics engine. The output image width will be equal to the input image width x N / M. ***The value of N must be equal or less than M.***
[7:0]	**SP HSF_M**	**8-bit Horizontal M Scaling Factor** An 8-bit value specifies the denominator part (M) of the horizontal scaling factor in graphics engine. The output image width will be equal to the input image width x N / M. ***The value of N must be equal or less than M.***

(Debug Port) Graphics Engine Sprite Status Register

Register	Address	R/W	Description	Reset Value
GE_SPSTS	GE_BA+0x034	R	Graphics Engine Sprite Status Register	0x0000_0000

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

Status[7:0]

Bits	Descriptions	
[31:0]	Status	GE Sprite Status

(Debug Port) Graphics Engine Date/Version ID Register

Register	Address	R/W	Description	Reset Value
GE_DVID	GE_BA+0x03C	R	Graphics Engine Date/Version ID Register	0x0915_E500

31	30	29	28	27	26	25	24
Date Month (09)							
23	22	21	20	19	18	17	16
Date Day (15)							
15	14	13	12	11	10	9	8
Version (E5)							
7	6	5	4	3	2	1	0
Reserved							

Bits	Descriptions	
[31:24]	Date	Month ID = 09
[23:16]	Date	Day ID = 15
[15:8]	Version	Version ID = E5

(Debug Port) Background Y/X Input Coordinates Register

Register	Address	R/W	Description	Reset Value
DBG_BGYXIN	GE_BA+0x040	R/W	Background Y/X Input Coordinates	0xXXXX_XXXX

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

Bits	Descriptions

[26:16]	**Background Y Input**	**11-bit Background Y Input Coordinate** For background debug, it specifies the BG Y input coordinate.
[10:0]	**Background X Input**	**11-bit Background X Input Coordinate** For background debug, it specifies the BG X input coordinate

(Debug Port) Background Y/X Output Coordinates Register

Register	Address	R/W	Description	Reset Value
DBG_BGYXOUT	GE_BA+0x044	R	Background Y/X Output Coordinates	0xXXXX_XXXX

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

Bits	Descriptions	
[26:16]	**Background Y Output**	**11-bit Background Y Output Coordinate** For background debug, it specifies the BG Y Output coordinate.
[10:0]	**Background X Output**	**11-bit Background X Output Coordinate** For background debug, it specifies the BG X Output coordinate

(Debug Port) Sprite Y/X Input Coordinates Register

Register	Address	R/W	Description	Reset Value
DBG_SPYXIN	GE_BA+0x048	R/W	Sprite Y/X Input Coordinates (Debug Port)	0xXXXX_XXXX

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

Bits	Descriptions	
[26:16]	**Sprite Y Input**	**11-bit Sprite Y Input Coordinate** For sprite debug, it specifies the sprite Y input coordinate.
[10:0]	**Sprite X Input**	**11-bit Sprite X Input Coordinate**

		For sprite debug, it specifies the sprite X input coordinate

(Debug Port) Sprite Y/X Output Coordinates Register

Register	Address	R/W	Description	Reset Value
DBG_SPYXOUT	GE_BA+0x04C	R	Sprite Y/X Output Coordinates (Debug Port)	0xXXXX_XXXX

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

Bits	Descriptions	
[26:16]	**Sprite Y Output**	**11-bit Sprite Y Output Coordinate** For sprite debug, it specifies the sprite Y output coordinate.
[10:0]	**Sprite Y Output**	**11-bit Sprite X Output Coordinate** For sprite debug, it specifies the sprite X output coordinate

4.10 Graphic Processing Unit

The Graphic Processing Unit (GPU) is designed to implement the TV-game, TV-toy, and pre-school education applications. It supports four layers of Background and 256 sprites in one frame display.

4.10.1 Overview and Features

- Resolution:
  - QVGA: 320 x 240 pixels.
  - VGA: 640x480 pixels.
- Colors: Up to 256 colors out of 32768 colors
  - Cell mode: Programmable 16/256 colors
    - One color palette in 256 color mode.
    - 16 sub-color palettes in 16 color mode.
  - Bitmap mode: 256/32768 colors for Background, 32768 colors for Sprite
  - Transparent:
    - The first color is defined as transparent by hardware in each sub-color palettes.
    - The other colors of each sub-color palettes can be defined as transparent separately by user.
- Backgrounds (BG):
  - Four layers of background. View window is
    - QVGA: 320 x 240 pixels.
    - VGA: 640 x 480 pixels.
  - Support cell mode or bitmap mode
    - Cell mode: The cell size is
      - QVGA: 8 x 8 or 16 x 16 pixels.
      - VGA: 16 x 16 or 32 x 32 pixels.
      - Cell patterns are out of 512/1024/2048/4096 BG cell patterns in 16 color mode or out of 8192 BG cell patterns in 256 color mode.
    - Bitmap mode: Supports 8 bits/pixel index bitmap mode or RGB555 15 bits/pixel bitmap mode.
  - Programmable color mode in cell mode: 16 colors x 16 sub-color palettes or 256 colors x 1 color palette.
  - Support four color palette RAM for four background layers.
  - Programmable BG source picture size, the view window is divided into BGxXBC x BGxYBC cells. BGxXBC and BGxYBC are defined by users, and the maximum virtual BG picture size is
    - QVGA: 512 x 512 pixels.
    - VGA: 1024 x 1024 pixels.
  - Programmable mode (bitmap or cell mode) for each BG layer. The priority of BG layers is fixed, BG0 > BG1 > BG2 > BG3 from the viewer.
  - Support one BG layer can be Rotation/Scaling and Perspective (RSP).
  - Support speed up mode and clipping window control for RSP BG.
  - Support alpha-blending, fading and invert color effects.
  - Support wallpaper mode, H/V flip and mosaic effect in cell mode.
  - Support mixed display of cell mode and bitmap mode.
- Sprites (SPR):
  - Support cell mode or bitmap mode
    - Cell mode: The cell size is
      - QVGA: Horizontal size: 8, 16, 32 or 64 pixels, Vertical size: 8, 16, 32 or 64 pixels.
      - VGA: Horizontal size: 16, 32, 64 or 128 pixels, Vertical size: 16, 32, 64 or 128 pixels.

- Bitmap mode: Horizontal size: 8, 16, 32, or 64 pixels, Vertical size: 8, 16, 32 or 64 pixels. (The hardware structure issue, sprite bitmap mode can't support 128 pixels size)

- Programmable color mode in cell mode: 16 colors x 16 sub-color palettes or 256 colors x 1 color palette.
- Cell patterns are out of
  - QVGA & VGA (based on 8x8 pixels cell size in 16 color mode):
    - 256 color mode : 131840 / 33536 sprite cell name
      - 768 cell names from internal RAM
      - 131072 / 32768 cell names from external memory
    - 16 color mode : 8960 sprite cell name
      - 768 cell names from internal RAM
      - 8192 cell names from external memory
    - Bitmap mode : 131840 sprite cell name
      - 768 cell names from internal RAM
      - 131072 cell names from external memory
- Maximum 256 Sprites (H size = 8 pixels) in one frame if no any sprite that turn on rotate/scale function.
- Support alpha-blending, fading and invert color effects.
- Support H/V flip and mosaic effect in cell mode.
- Programmable priority between BG layers and Sprite layer.
- Support 16 sets of rotate/scale parameters that each sprite can index one.
- Support at most 32 bitmap mode sprites.

- View processor:
  - 2 Windows control.
  - Light pen function.
  - 16-level fading effect coefficient of four BG layers and one SPR layer.
  - 16-level Alpha-blending effect coefficient of two BG layers or one BG layer with SPR layer.

4.10.2 GPU Block Diagram

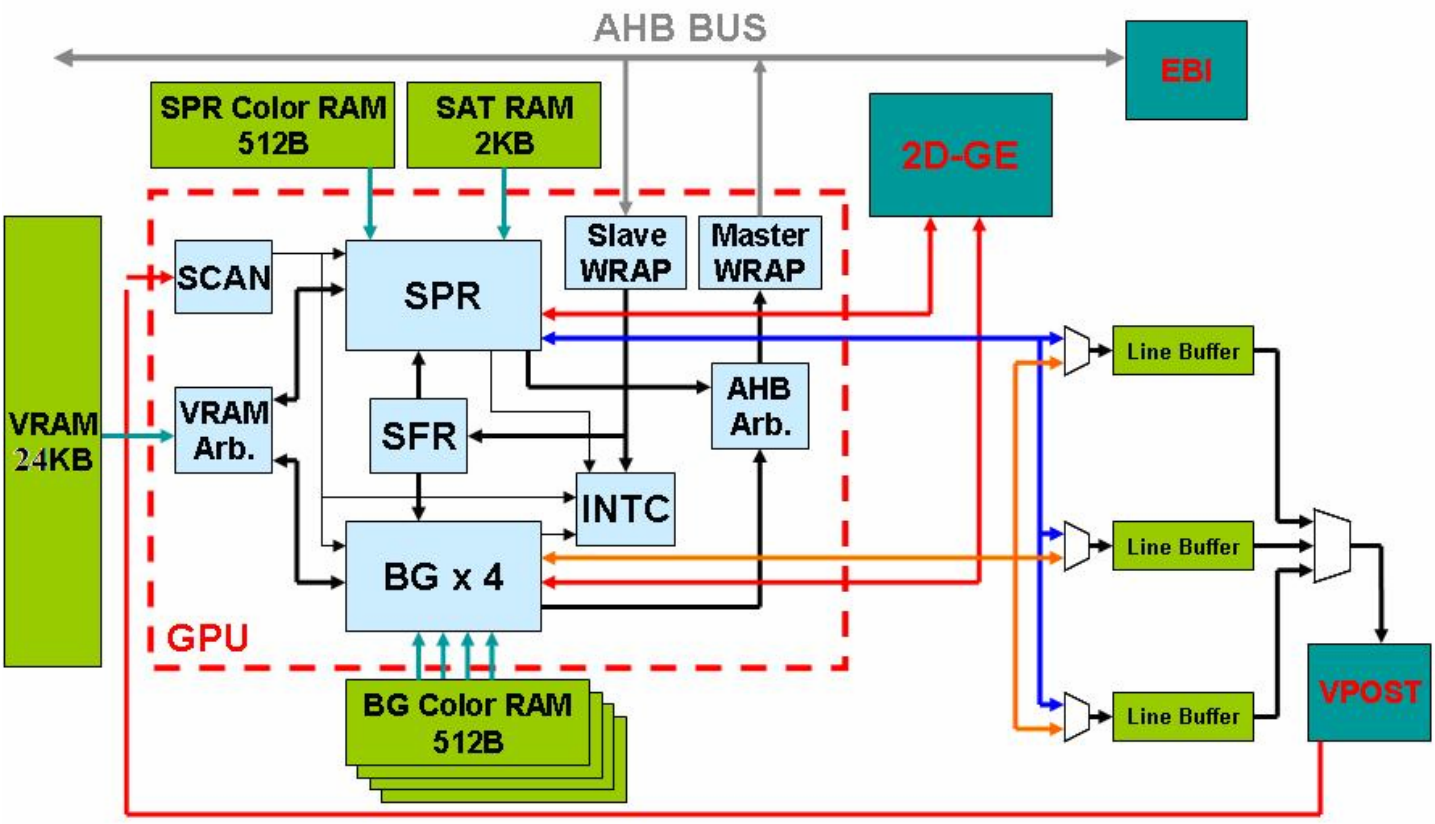


Figure 4.10-1 GPU Block Diagram

4.10.3 GPU Functional Description

4.10.3.1 Background

Support mixed display of cell mode and bitmap mode in one frame. These patterns of cell mode or bitmap mode can come from the internal VRAM or external memory by each BG layer setting. The active view window is 320 x 240 pixels (QVGA) or 640 x 480 pixels (VGA) and is divided into 40 x 30 cells (8 x 8 pixels for QVGA, 16 x 16 pixels for VGA) or 20 x 15 cells (16 x 16 pixels for QVGA, 32 x 32 pixels for VGA) in cell mode. These cells are selected out of 512/1024/2048/4096 BG cell patterns in 16 color mode or out of 8192 BG cell patterns in 256 color mode. Four BG layers, BG0, BG1, BG2 and BG3 are provided and all of them can be scrolled in horizontal or/and vertical direction. X/Y-axis mirror can be also provided in each single BG cell in cell mode. The mode of each BG layer can be programmable. The priority of these four BG layers is fixed in the order of BG0 > BG1 > BG2 > BG3. The highest priority BG0 layer draws to the line buffer first, then BG1, then BG2, and then BG3. Only one BG layer can be rotated/scaled by the hardware implementation. Support clipping window and speed up mode to implement the rotation/scaling function when the scan-line period is not enough.

Provide four background color palette RAM, these color palette RAM have fixed address,
0xFFE73000: background color palette RAM 0

0xFFE73200: background color palette RAM 1
0xFFE73400: background color palette RAM 2
0xFFE73600: background color palette RAM 3
Each background can use the same or different color palette RAM that is controlled by SFR BG0CTL[17:16], BG1CTL[17:16], BG2CTL[17:16], BG3CTL[17:16]. If all of them are [0,0], then the four backgrounds use the same color palette RAM like VA75. If user doesn't use all of the four color palette RAMs, they can be used as the internal RAM for programming.

4.10.3.1.1 BAT RAM

In cell mode, the Background Attribute Table (BAT) RAM is (BGxXBC[5:0] + 1) x (BGxYBC[5:0] + 1) x 2 bytes in size in 8 x 8 pixels QVGA mode, or (BGxXBC[5:0] + 1)/2 x (BGxYBC[5:0] + 1)/2 x 2 bytes in 16 x 16 pixels QVGA mode, or (BGxXBC[6:0] + 1)/2 x (BGxYBC[6:0] + 1)/2 x 2 bytes in 16 x 16 pixels VGA mode, or (BGxXBC[6:0] + 1)/4 x (BGxYBC[6:0]+1)/4 x 2 bytes in 32 x 32 pixels VGA mode for each BG layer and it records the sub-color palette index (16 color mode only), cell name index, H/V flip enable, and blend effect enable attributes for each single cell. All of BG0, BG1, BG2 and BG3 have their own BAT RAM starting address by programming the SFR BGxATB respectively. These BG cells [X,Y] are arranged in the following format mapping to the BG source picture, where [0,0] is in the most top-left corner and [BGxXBC,BGxYBC] is in the most bottom-right corner. The underflow or overflow area can be set as transparent or wraps around. Wraps around function must achieve on 2^n x 2^n background size. The maximum virtual BG picture size is 512 x 512 pixels that is divided into 64 x 64 cells (cell = 8 x 8 pixels) or 32 x 32 cells (cell = 16 x 16 pixels) in QVGA mode, or 1024 x 1024 pixels that is divided into 64 x 64 cells (cell = 16 x 16 pixels) or 32 x 32 cells (cell = 32 x 32 pixels) in VGA mode.

[0,0]	[1,0]							[BGxXBC,0]
[0,1]	[1,1]							[BGxXBC,1]
[0,BGxYBC-1]	[1,BGxYBC-1]							[BGxXBC,BGxYBC-1]
[0,BGxYBC]	[1,BGxYBC]							[BGxXBC,BGxYBC]

Table 4.10-1 BAT RAM Mapping Table of HC x VC Cells in Cell Mode

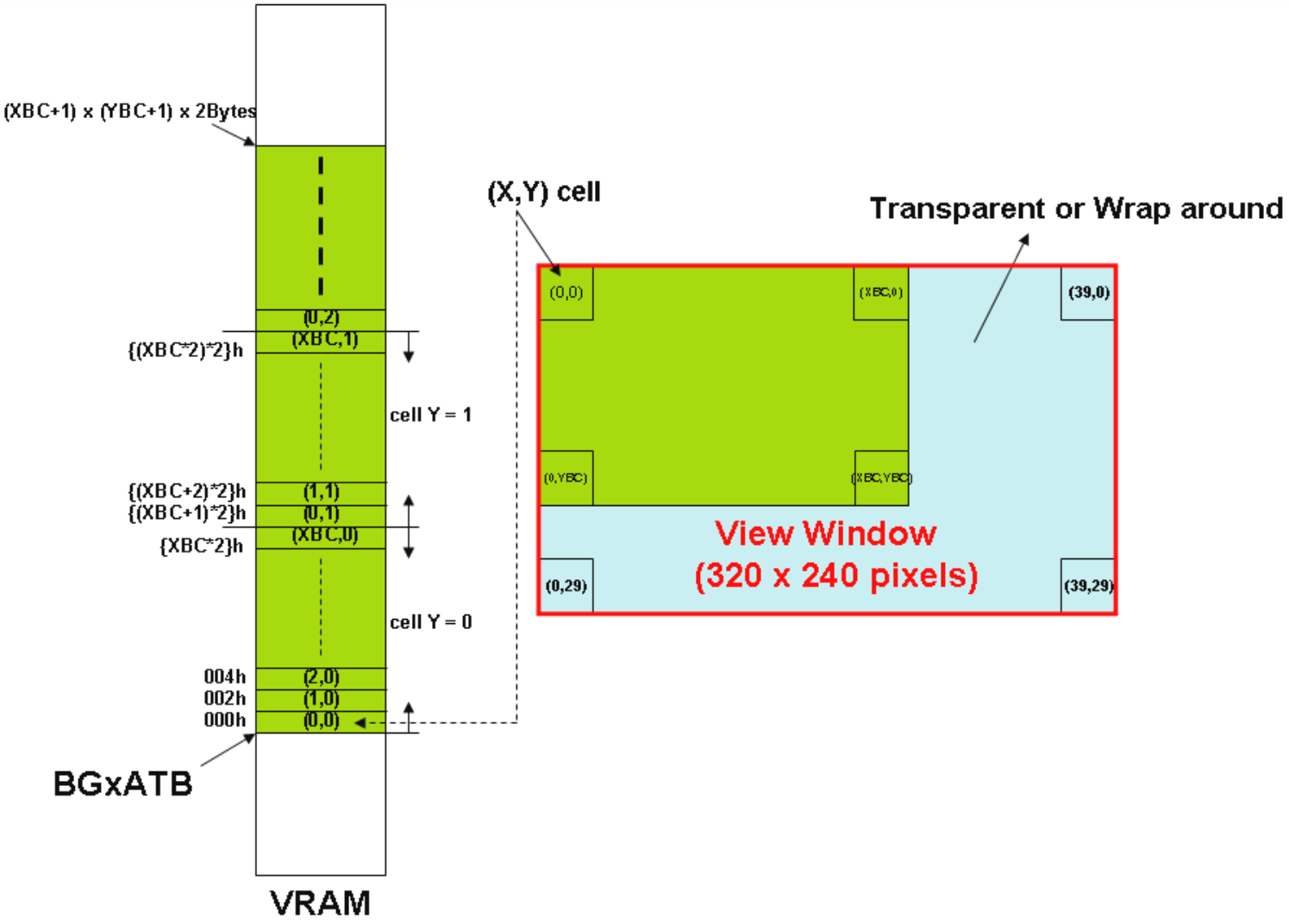


Figure 4.10-2 BAT Mapping in QVGA Cell Mode (Cell = 8x8 Pixels)

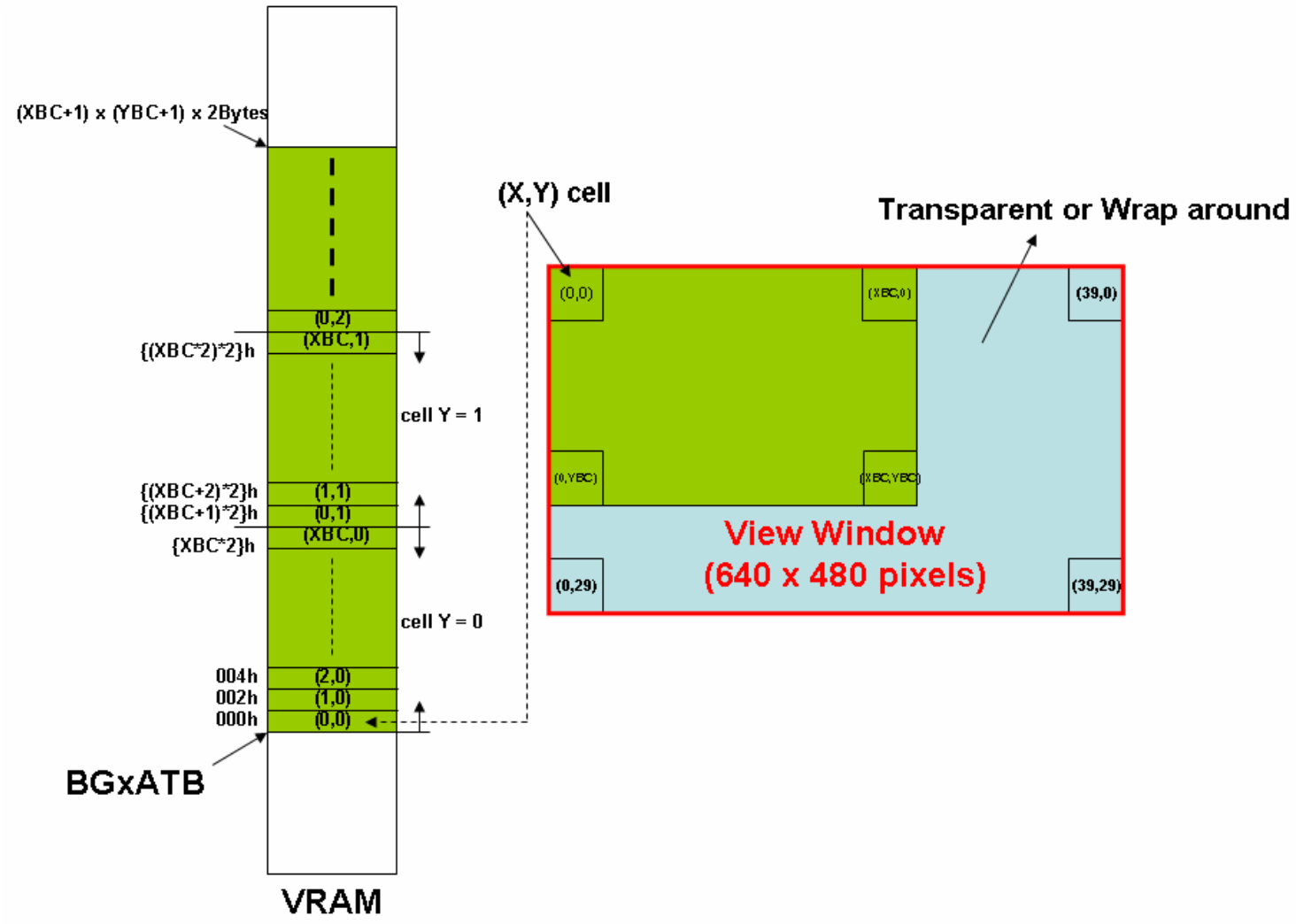


Figure 4.10-3 BAT Mapping in VGA Cell Mode (Cell = 16x16 Pixels)

4.10.3.1.2 BAT RAM Format

16 color mode

In 16 color mode, the CF, VF and HF attributes can be master controlled by the SFR BGxCTL.

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
SC3	SC2	SC1	SC0	CNx/ CF	CNy/ HF	CNz/ VF	CN8	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0

SC[3:0] : Sub-Color palette index.
CF : Color effect, 1=enable, 0=disable.*Note 1
HF : Horizontal Flip, 1=enable, 0=disable. *Note1
VF : Vertical Flip, 1=enable, 0=disable. *Note 1
CN[8:0]/CN[9:0]/CN[10:0]/CN[11:0] : Cell Name. BG cells out of 512/1024/2048/4096 BG cell patterns. *Note 2

*Note 1: CF is master controlled by CFC (BGxCTL[3]), HF is master controlled by HFC (BGxCTL[2]), and VF

is also master controlled by VFC (BGxCTL[6]) in 16 color mode only.

*Note 2: CN[11:9] depends on the CFC/HFC/VFC control bit enable or disable.

CFC/HFC/VFC	CN11	CN10	CN9	Cell Name Number
000	B11	B10	B9	4096
001	X	B11	B10	2048
010	X	B11	B9	2048
011	X	X	B11	1024
100	X	B10	B9	2048
101	X	X	B10	1024
110	X	X	B9	1024
111	X	X	X	512

256 color mode

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
CF	HF	VF	CN12	CN11	CN10	CN9	CN8	CN7	CN6	CN5	CN4	CN3	CN2	CN1	CN0

CF : Color effect, 1=enable, 0=disable.
HF : Horizontal Flip, 1=enable, 0=disable.
VF : Vertical Flip, 1=enable, 0=disable.
CN[12:0] : Cell Name. BG cells out of 8192 BG cell patterns.

4.10.3.1.3 Background Scrolling

BG layers can be scrolled at X and Y directions smoothly by users that write the proper values to these SFR BGxX[9:0]/BGxY[9:0] (QVGA mode) or BGxX[10:0]/BGxY[10:0] (VGA mode) that define the X/Y-axis coordinate of left-upper corner of BG layer. BGxX[9:0] (QVGA) or BGxX[10:0] (VGA) can be reloaded automatically in next head of scan line, it can make up the pixel shift-effect of next scan line in X direction. These registers are used to control the scrolling function in all four BG layers. The X/Y-axis coordinate system of BG layers are shown as below.

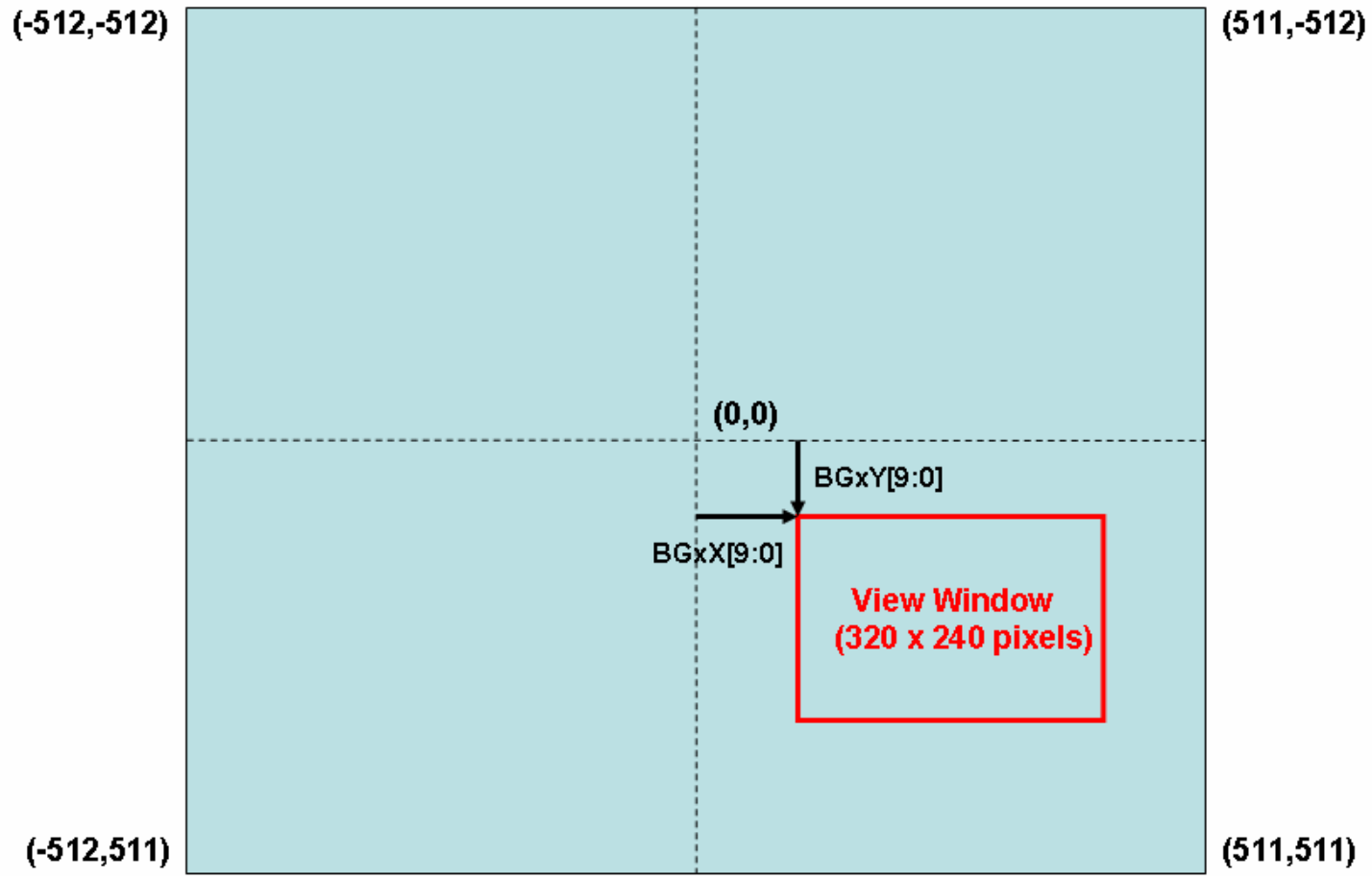


Figure 4.10-4 BG X / Y-axis Coordinate System in QVGA Mode

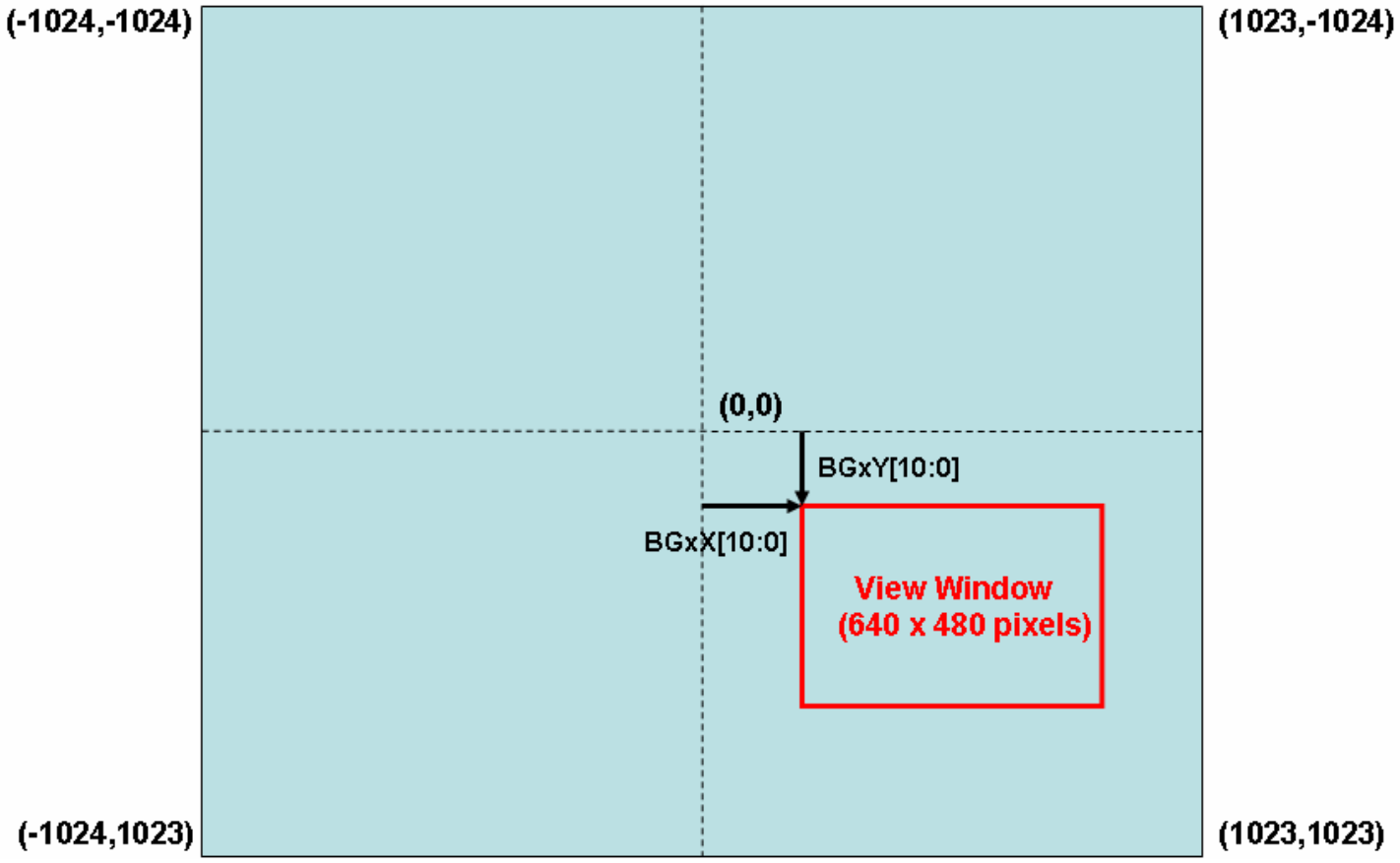


Figure 4.10-5 BG X/Y-axis Coordinate System in VGA Mode

4.10.3.1.4 Background Pattern Format

Two kinds of pattern format for each BG layer can be set, one is for cell mode and the other is for bitmap mode.

Cell Mode

One 8 x 8 pixels cell in QVGA mode, the cell pattern consist of 8 words in 16 color mode, but 16 words in 256 color mode. One 16 x 16 pixels cell in QVGA and VGA mode, the cell pattern consist of 32 words in 16 color mode, but 64 words in 256 color mode. One 32 x 32 pixels cell in VGA mode, the cell pattern consist of 128 words in 16 color mode, but 256 words in 256 color mode.

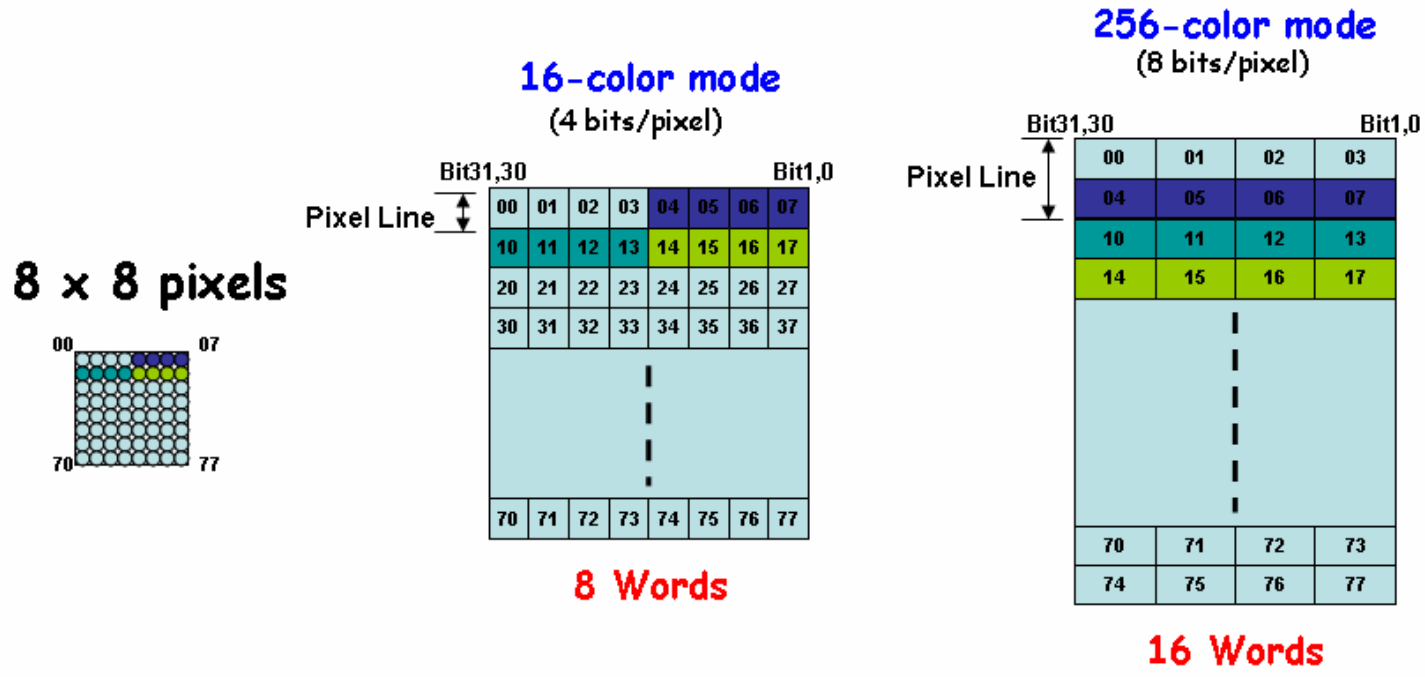


Figure 4.10-6 Pattern 8x8 Pixels Format in QVGA Mode

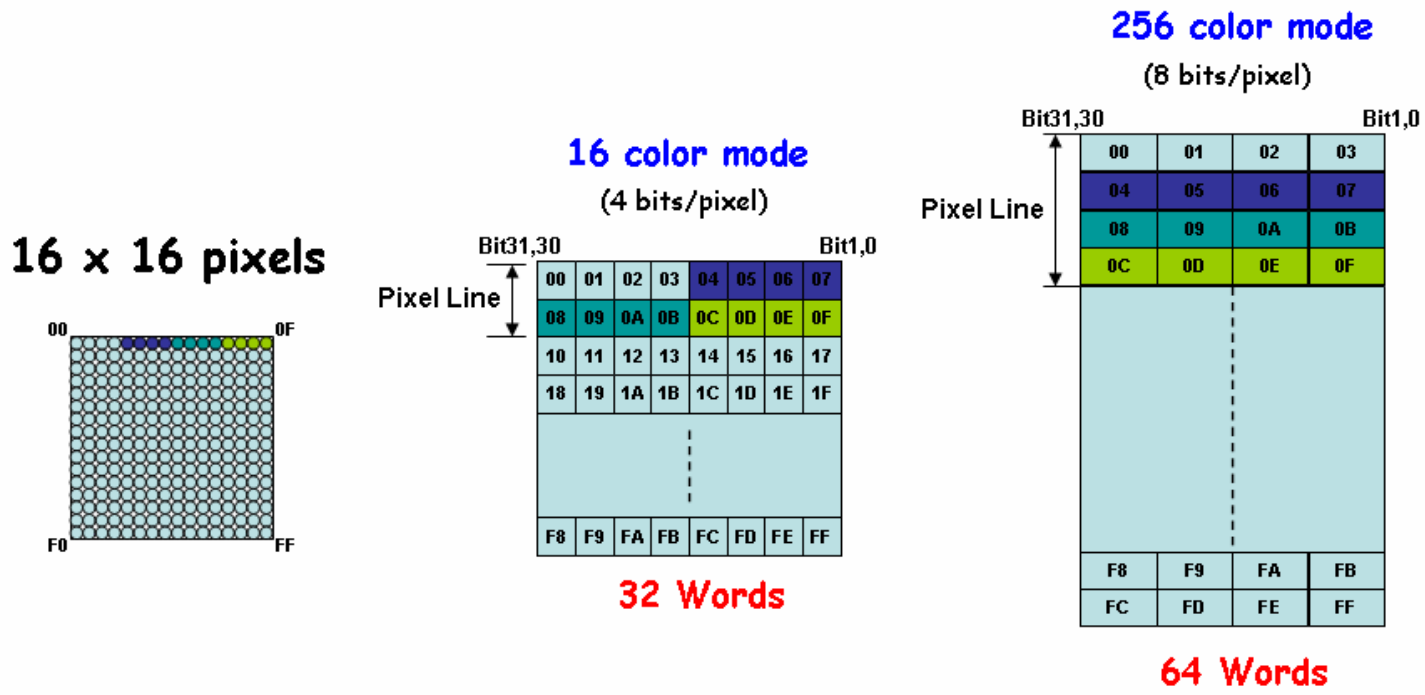


Figure 4.10-7 Pattern 16x16 Pixels Format in QVGA and VGA Mode

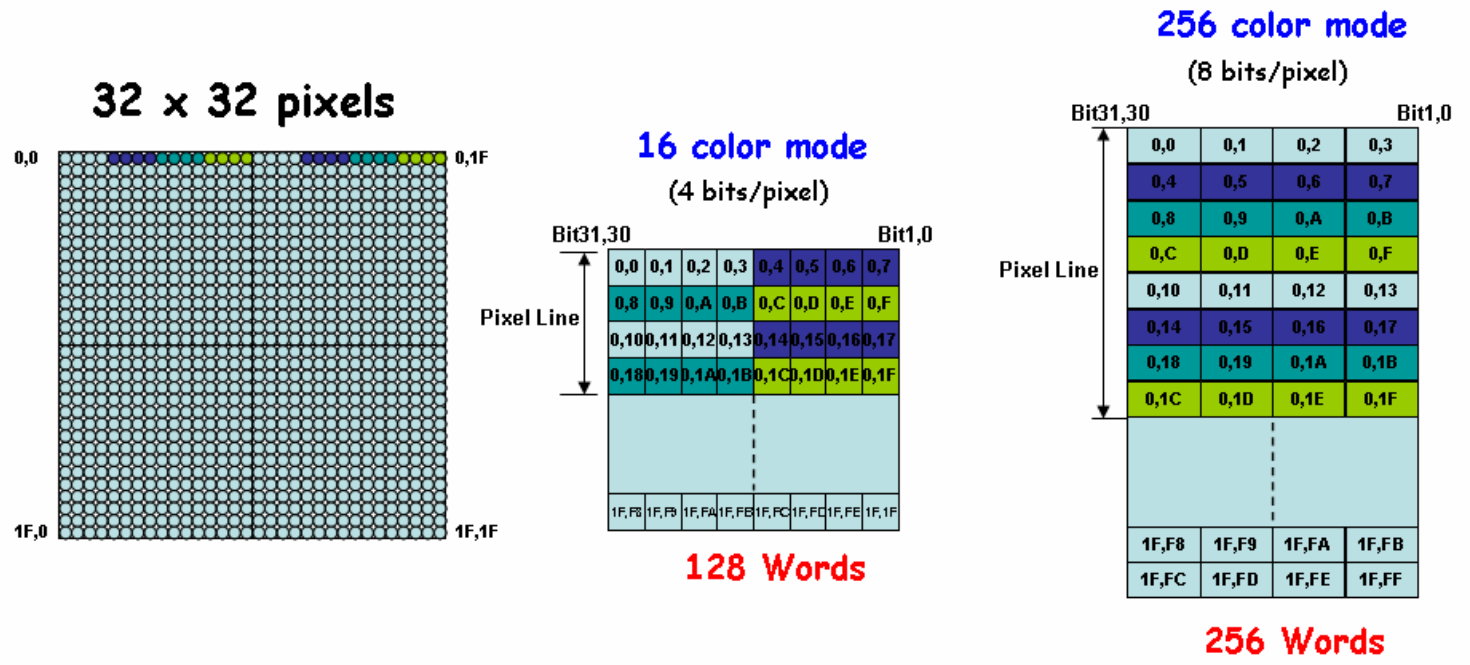


Figure 4.10-8 Pattern 32x32 Pixels Format in VGA Mode

Bitmap Mode

Support 8-bits (256 colors) index bitmap mode and 15-bits RGB555 bitmap mode. The pattern formats are shown as below

- 8-bits index bitmap mode (256 colors).
- 15-bits RGB555 bitmap mode (32768 colors).

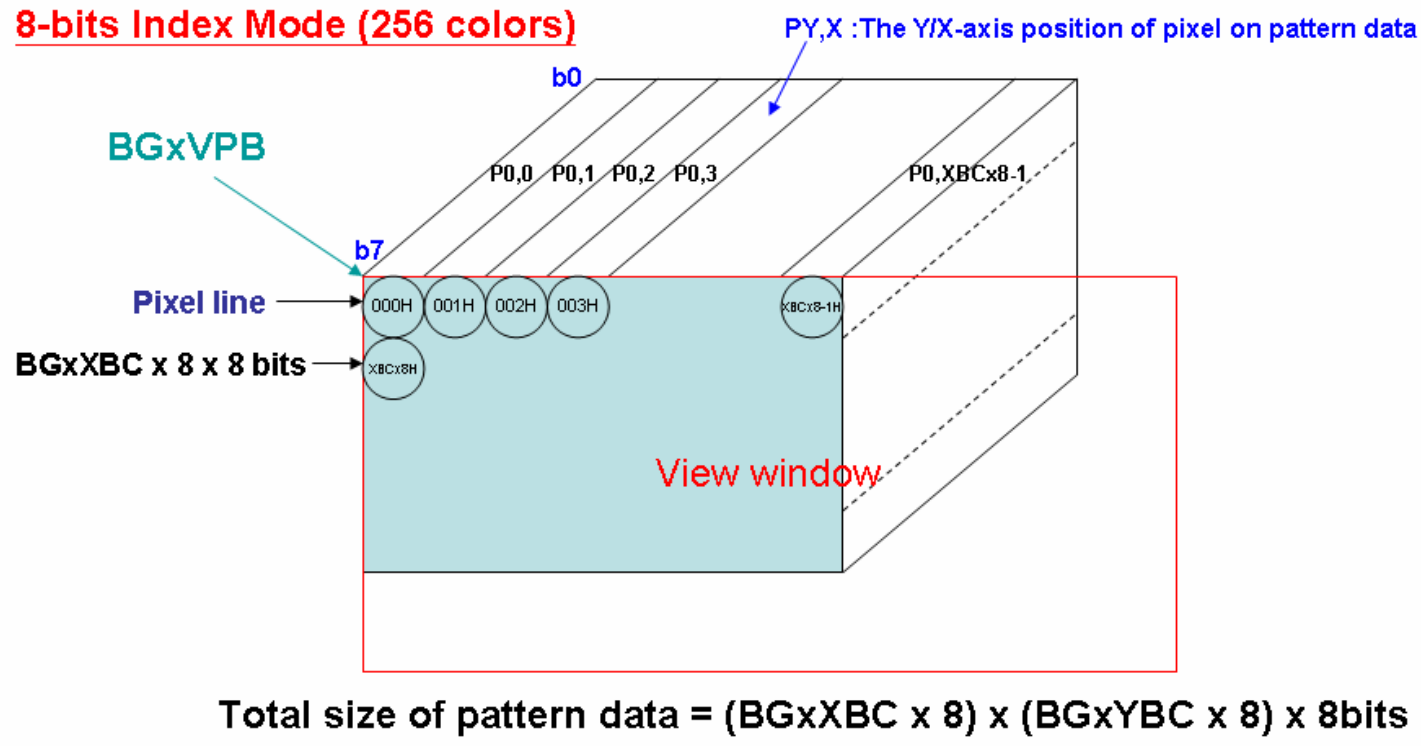


Figure 4.10-9 Pattern Format in 8-bits Index Bitmap Mode

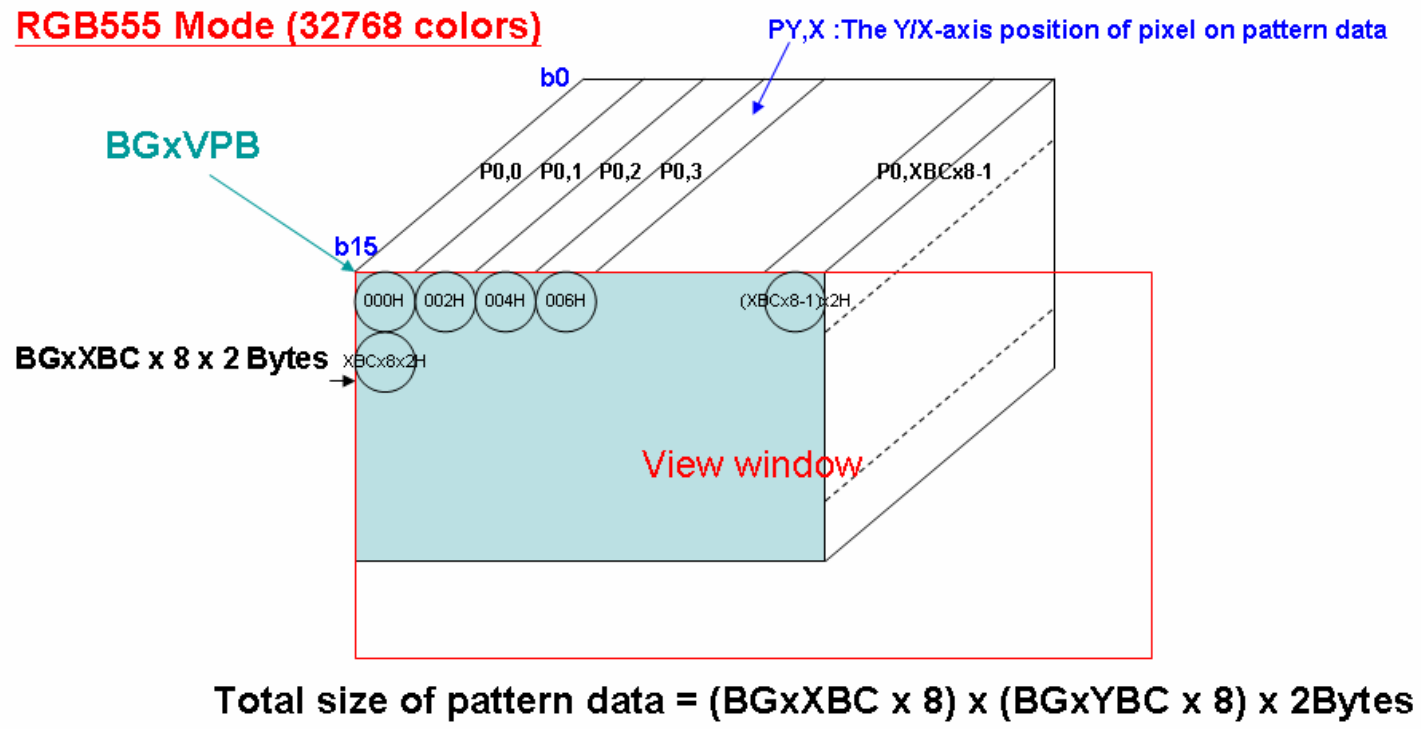


Figure 4.10-10 Pattern Format in RGB555 Bitmap Mode

4.10.3.1.5 Background Interface with 2D Graphic Engine

Signal Descriptions:

- C2G_Req: CRTC Request Signal
- C2G_XYin: CRTC Input X-Y Coordinates (HCNT, VCNT)
- G2C_ReqEn: GE Request Enable to CRTC
- G2C_XYRdy: GE Output X-Y Ready
- G2C_XYout: GE Output X-Y Coordinates
- G2C_XYHit: GE Output X-Y Hit (Inside Region of Source Object)
- G2C_ScanComplete: GE Reaches the Boundary of Source Object

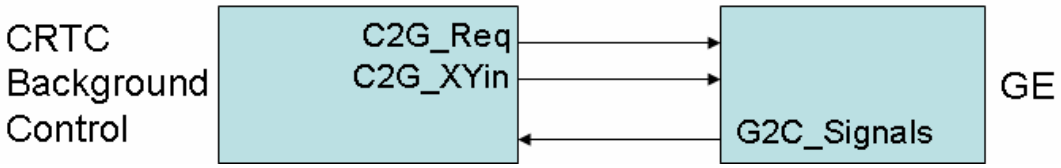


Figure 4.10-11 Signal Descriptions and Relationships with 2D GE

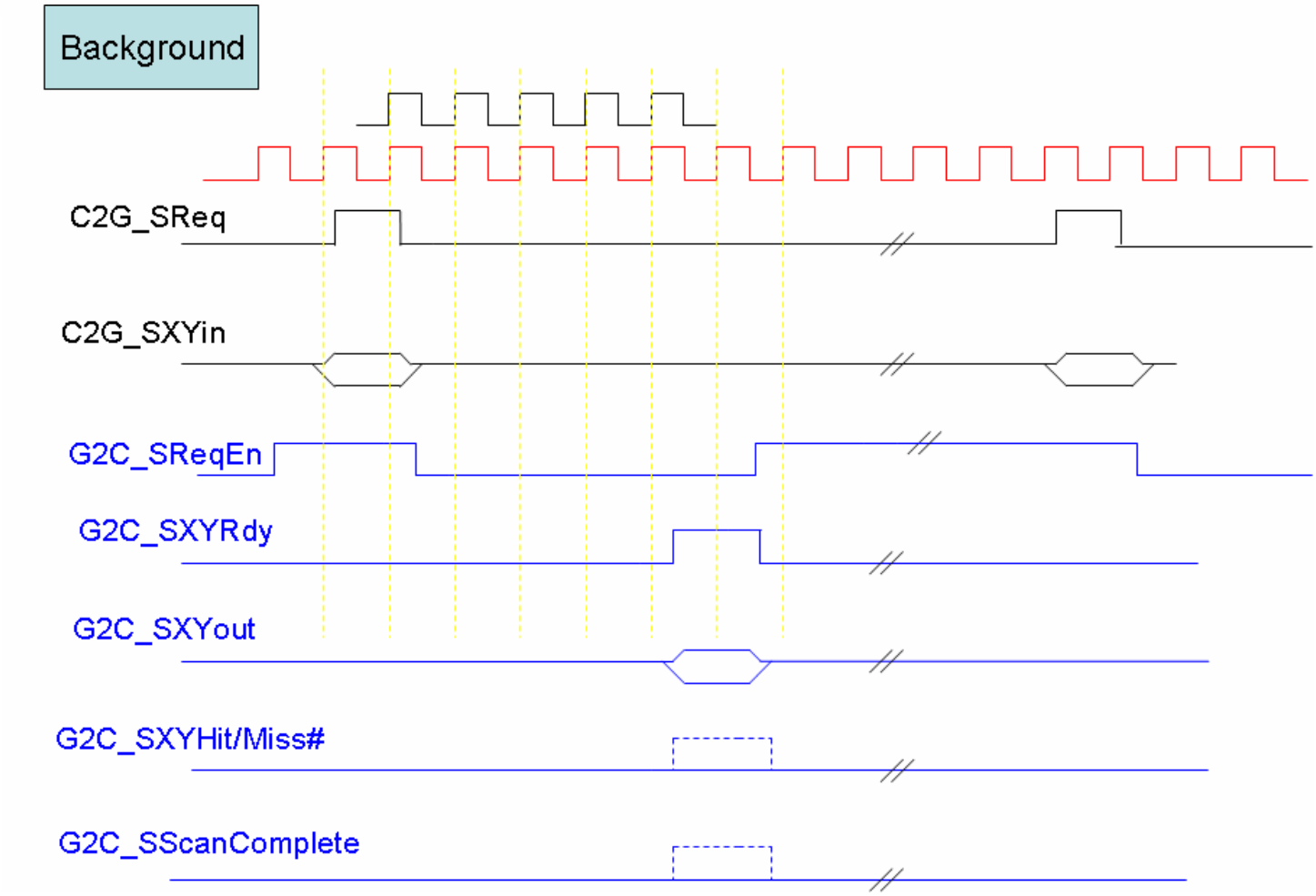


Figure 4.10-12 Signal Timing Diagram between BG and 2D GE

4.10.3.2 Sprite

One sprite size can be programmed as 16 kinds of size, i.e, 8x8, 8x16, 8x32, 8x64, 16x8, 16x16, 16x32, 16x64, 32x8, 32x16, 32x32, 32x64, 64x8, 64x16, 64x32, or 64x64 pixels in QVGA mode or 16x16, 16x32, 16x64, 16x128, 32x16, 32x32, 32x64, 32x128, 64x16, 64x32, 64x64, or 128x128 pixels in VGA mode. Sprite can be flipped (mirrored) horizontally and vertically. Each sprite also can be programmed as alpha blending effect target, fading in/out sprite, window sprite, or mosaic sprite. At most 256 sprites that defined in the SPFN[7:0] (rotate/scale functions are disable for all of 256 sprites) are available in one single frame. These sprite patterns are selected out of maximum 131840 (256 color VGA mode) or 33536 (256 color QVGA mode) or 8960 (16 color QVGA/VGA mode) sprite cell name (the cell name is defined as 8x8 pixels in 16 color mode, 768 cell name are from the internal Video RAM, the other 131072 (256 color VGA mode) or 32768 (256 color QVGA mode) or 8192 (16 color QVGA/VGA mode) cell name are from the external memory). To make sprite invisible from the view window, the user must turn it off (EN = 0 in QVGA mode or X[10:0]= 0x400 in VGA mode) or move the sprite Y-axis position to the outside of view window individually. (Not recommend to move the sprite X-axis position to the outside location of view window but Y-axis position is inside of view window to save some judging clocks). The sprite priority can be set between four BG layers and sprites. If sprites have the same priority, the lowest order number of sprite has the higher priority, i.e. SP0 > SP1 > … > SP255. There are two kinds of color modes for each sprite in cell mode. Sixteen sets of rotation/scaling parameters can be selected by any one sprite to implement the rotation or/and scaling functions. Sprite also support 15-bits RGB555 bitmap mode for the first 32 sprites, the bitmap mode size is 8x8, 8x16, 8x32, 8x64, 16x8, 16x16, 16x32, 16x64, 32x8, 32x16, 32x32, 32x64, 64x8,

64x16, 64x32, or 64x64 pixels in both QVGA and VGA mode. The sprite bitmap switches are defined in the SFR SPBMP[31:0].

4.10.3.2.1 SAT RAM

Each Sprite have eight attribute bytes that stored in the Sprite Attribute Table RAM (SAT RAM). The total sprite Attribute Table RAM size is equal to 256 x 8 bytes = 2048 bytes. It can provide maximum 256 sprites in a frame if no any sprite's rotation or scaling function is enable. The maximum TPM size is equal to 16 x 12 Bytes = 192 bytes that located on the last space of SAT RAM.

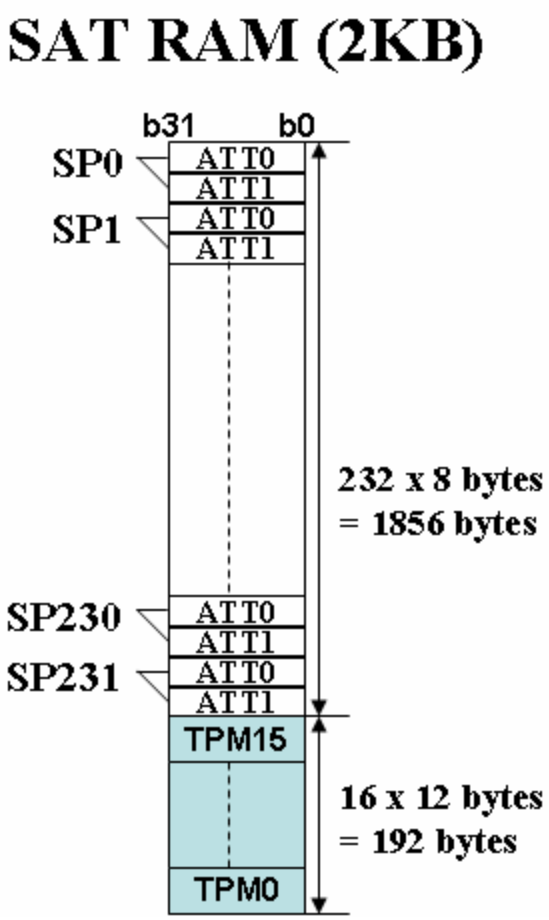


Figure 4.10-14.10-3 SAT RAM

4.10.3.2.2 SAT RAM Format

B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
BRTS[3:0]				**WN[1:0]**		**X[9:0] / X[10:1]**									
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
EN / X[0]	**RSEN**	**VF**	**HF**	**VS[1:0]**		**Y[9:0]**									

Table 4.10-2 SPx/ATT0 Format Table, x=0 – 255

B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
CM	**IC**	**MD**	**MS**	**EF[1:0]**		**PR[1:0]**		**HS[1:0]**		**BK[1:0]**		**TPM[3:0]**			
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
EXT	**SC[3:2]/ CN[14:13]**		**SC[1:0]/ CN[12:11]**		**CN[10:0]**										

Table 4.10-3 SPx/ATT1 Format Table, x=0 – 255

Bits	SPx/ATT0 Content Descriptions	
[31:28]	BRTS[3:0]	Boundary Rectangle Size 1111 = 384 x 384 pixels for QVGA or Bitmap Mode, 768 x 768 pixels for VGA Mode 1110 = 288 x 288 pixels for QVGA or Bitmap Mode, 576 x 576 pixels for VGA Mode 1101 = 192 x 192 pixels for QVGA or Bitmap Mode, 384 x 384 pixels for VGA Mode 1100 = 144 x 144 pixels for QVGA or Bitmap Mode, 288 x 288 pixels for VGA Mode 1011 = 96 x 96 pixels for QVGA or Bitmap Mode, 192 x 192 pixels for VGA Mode 1010 = 72 x 72 pixels for QVGA or Bitmap Mode, 144 x 144 pixels for VGA Mode 1001 = 48 x 48 pixels for QVGA or Bitmap Mode, 96 x 96 pixels for VGA Mode 1000 = 36 x 36 pixels for QVGA or Bitmap Mode, 72 x 72 pixels for VGA Mode 0111 = 24 x 24 pixels for QVGA or Bitmap Mode, 48 x 48 pixels for VGA Mode 0110 = 12 x 12 pixels for QVGA or Bitmap Mode, 24 x 24 pixels for VGA Mode 0101 = 256 x 256 pixels for QVGA or Bitmap Mode, 512 x 512 pixels for VGA Mode 0100 = 128 x 128 pixels for QVGA or Bitmap Mode, 256 x 256 pixels for VGA Mode 0011 = 64 x 64 pixels for QVGA or Bitmap Mode, 128 x 128 pixels for VGA Mode 0010 = 32 x 32 pixels for QVGA or Bitmap Mode, 64 x 64 pixels for VGA Mode 0001 = 16 x 16 pixels for QVGA or Bitmap Mode, 32 x 32 pixels for VGA Mode 0000 = 8 x 8 pixels for QVGA or Bitmap Mode, 16 x 16 pixels for VGA Mode
[27]	WN[1]	Window 1 Control 1 = Enable 0 = Disable
[26]	WN[0]	Window 0 Control 1 = Enable 0 = Disable
[25:16]	X[9:0]/ X[10:1]	Sprite X-axis location on screen. QVGA mode: X[9:0] that are ranged from -512 to 511. VGA mode: X[10:1] with En (=X[0]) that are ranged from -1024 to 1023.
[15]	EN/ X[0]	QVGA mode: Sprite Enable 1 = Enable 0 = Disable VGA mode: X[0], bit0 of Sprite X-axis location on screen *NOTE 1
[14]	RSEN	Rotate/Scale Enable 1 = Enable 0 = Disable
[13]	VF	Vertical Flip 1 = Enable

		0 = Disable
[12]	HF	Horizontal Flip 1 = Enable 0 = Disable
[11:10]	VS[1:0]	Vertical Size 11 = 64 pixels for QVGA or Bitmap Mode, 128 pixels for VGA Mode 10 = 32 pixels for QVGA or Bitmap Mode, 64 pixels for VGA Mode 01 = 16 pixels for QVGA or Bitmap Mode, 32 pixels for VGA Mode 00 = 8 pixels for QVGA or Bitmap Mode, 16 pixels for VGA Mode
[9:0]	Y[9:0]	Sprite Y-axis location on screen that are ranged from -512 to 511
Bits	**SPx/ATT1 Content Descriptions**	
[31]	CM	Color Mode 1 = 256 color mode 0 = 16 color mode
[30]	IC	Invert Color 1 = Enable 0 = Disable
[29]	MD	Sprite Mode 1 = Window sprite 0 = Normal sprite *NOTE 2 *NOTE 3
[28]	MS	Mosaic 1 = Enable 0 = Disable
[27:26]	EF[1:0]	Special Effect Control 11 = Fade-out (Y decrease) 10 = Fade-in (Y increase) 01 = Alpha Blending Target Enable (The sprite is always as the target layer and is always behind BG source layer (BG is higher priority) to implement the alpha blending effect) 00 = Disable
[25:24]	PR[1:0]	Priority 11 = the lowest priority 10 = 3rd 01 = 2nd 00 = the highest priority
[23:22]	HS[1:0]	Horizontal Size 11 = 64 pixels for QVGA or Bitmap Mode, 128 pixels for VGA Mode 10 = 32 pixels for QVGA or Bitmap Mode, 64 pixels for VGA Mode 01 = 16 pixels for QVGA or Bitmap Mode, 32 pixels for VGA Mode 00 = 8 pixels for QVGA or Bitmap Mode, 16 pixels for VGA Mode
[21:20]	BK[1:0]	Sprite Pattern Bank (only for external memory) 11 = bank 3 10 = bank 2 01 = bank 1 00 = bank 0

[19:16]	TPM[3:0]	Transformation Parameter Memory index for rotation and scaling.
[15]	EXT	External Memory Pattern Indicator 1 = Sprite pattern fetched from external memory 0 = Sprite pattern fetched from internal video RAM
[14:13]	SC[3:2]/ CN[14:13]	Sub-color Palette Index SC[3:2] in 16 Color Mode or Sprite Pattern Cell Name bit14~bit13 in 256 Color QVGA (SPCNO = 1) Mode or 256 Color VGA Mode or Bitmap Mode. *NOTE 4
[12:11]	SC[1:0]/ CN[12:11]	Sub-color Palette Index SC[1:0] in 16 Color Mode or Sprite Pattern Cell Name bit12~bit11 in 256 Color Mode *NOTE 4
[10:0]	CN[10:0]	Sprite Pattern Cell Name bit10~bit0

***NOTE 1**: Disable this bit (EN=0) if the sprite doesn't display any pixel in the view window in QVGA mode. But this bit is also defined as X[0] that is combined with X[10:1] to produce 11-bits sprite X-axis location, i.e, X[10:0] in VGA mode. When X[10:0] is equal to 0x400 that defined as "sprite disable" in VGA mode.

***NOTE 2**: The sprite window is also be controlled by window 0/1.

***NOTE 3**: The non-zero pattern index and the transparent color (transparent color: the bit15 of sprite color palette = 1) in sprite color palette RAM can be defined the object of window sprite.

***NOTE 4**: The sprite cell patterns address consists of

a. BK[1:0]+CN[10:0] in 16 color QVGA/VGA mode or

b. BK[1:0]+CN[12:0] in 256 color QVGA mode (SPCNO = 0) or

c. BK[1:0]+CN[14:0] in 256 color QVGA mode (SPCNO = 1) or 256 color VGA mode or Bitmap mode

4.10.3.2.3 Sprite Pattern Format

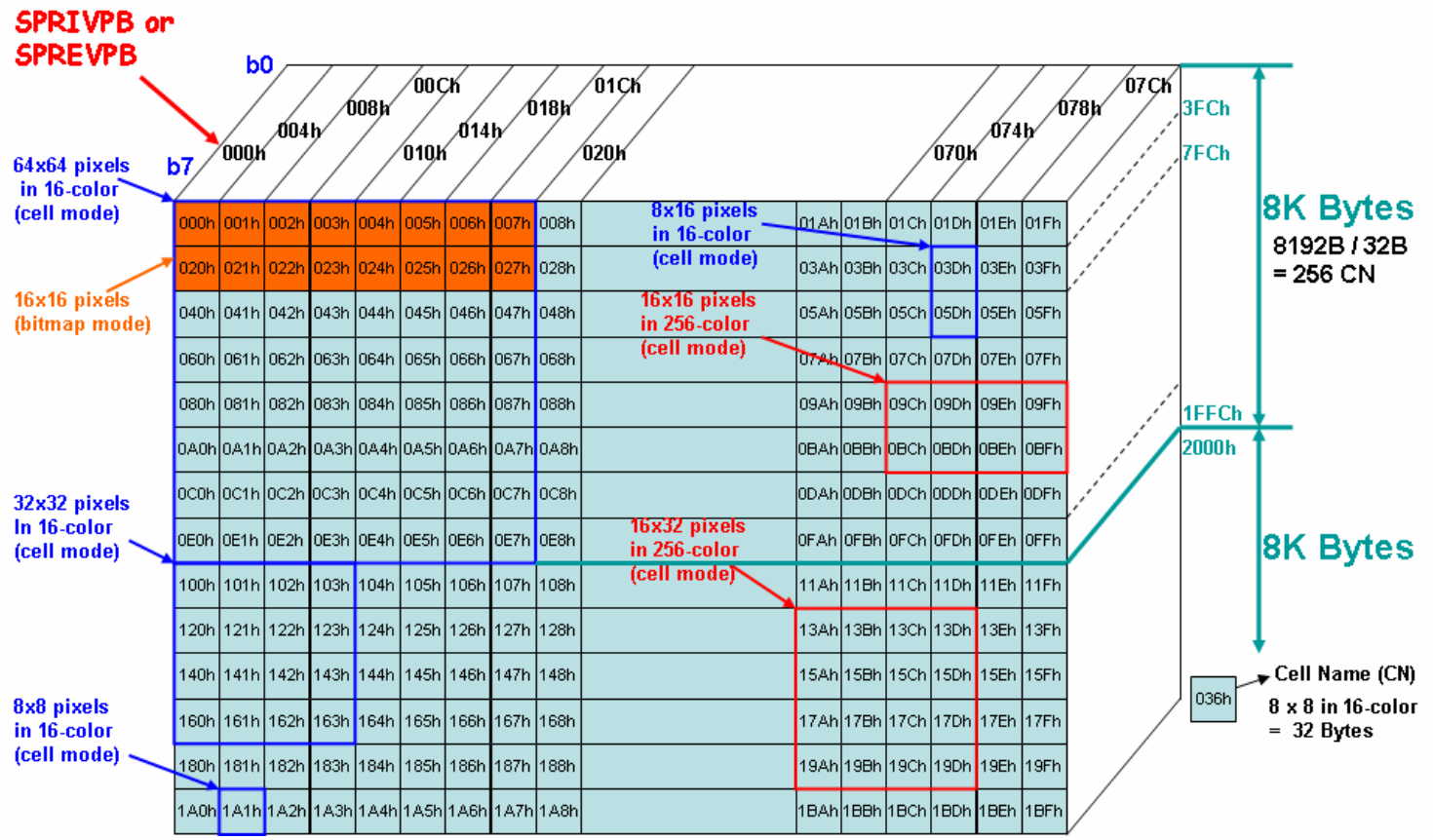


Figure 4.10-14.10-4 Sprite Pattern 2-Dimension Placement in Memory

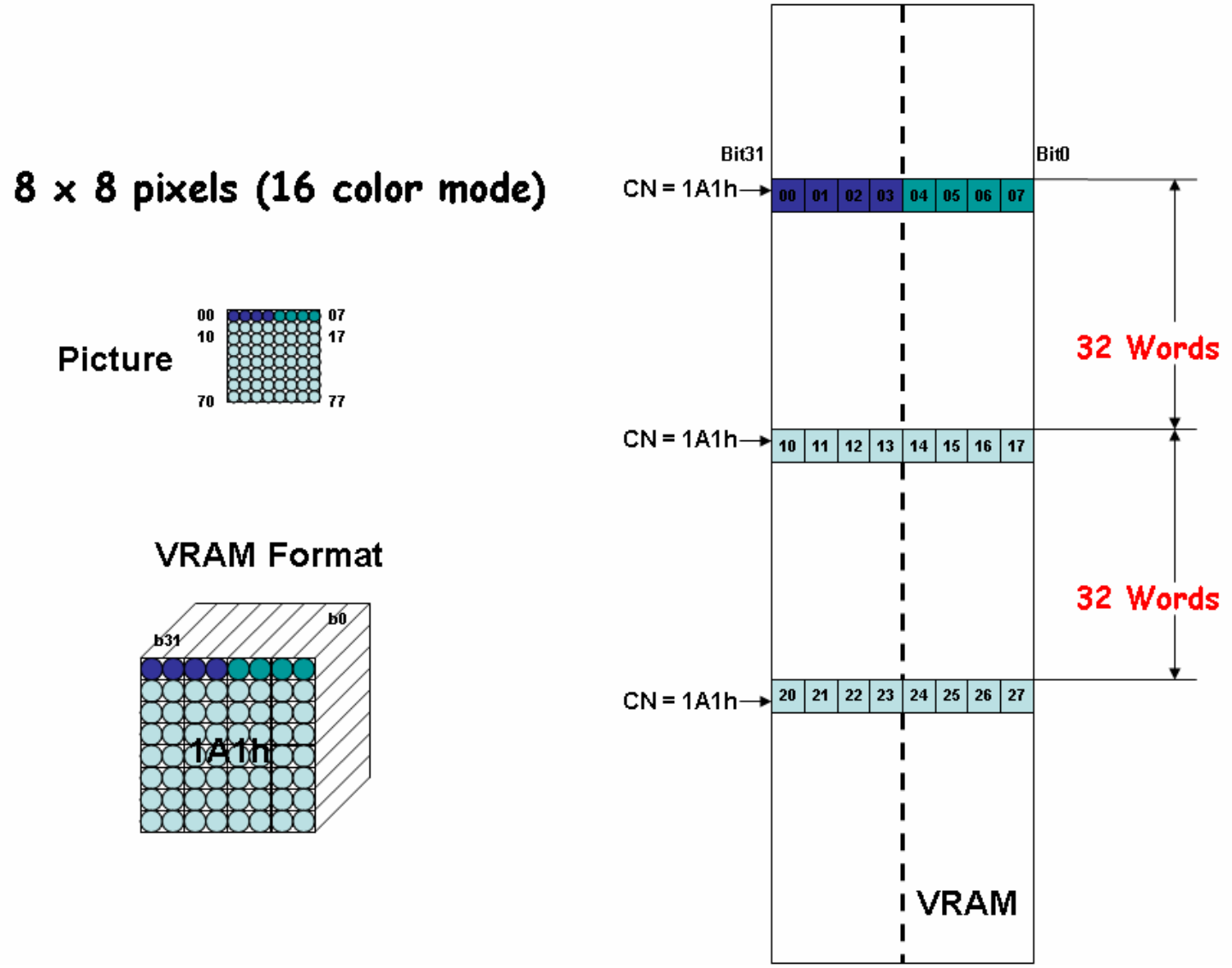


Figure 4.10-14.10-5 Sprite Pattern 8x8 Pixels Format in Cell Mode

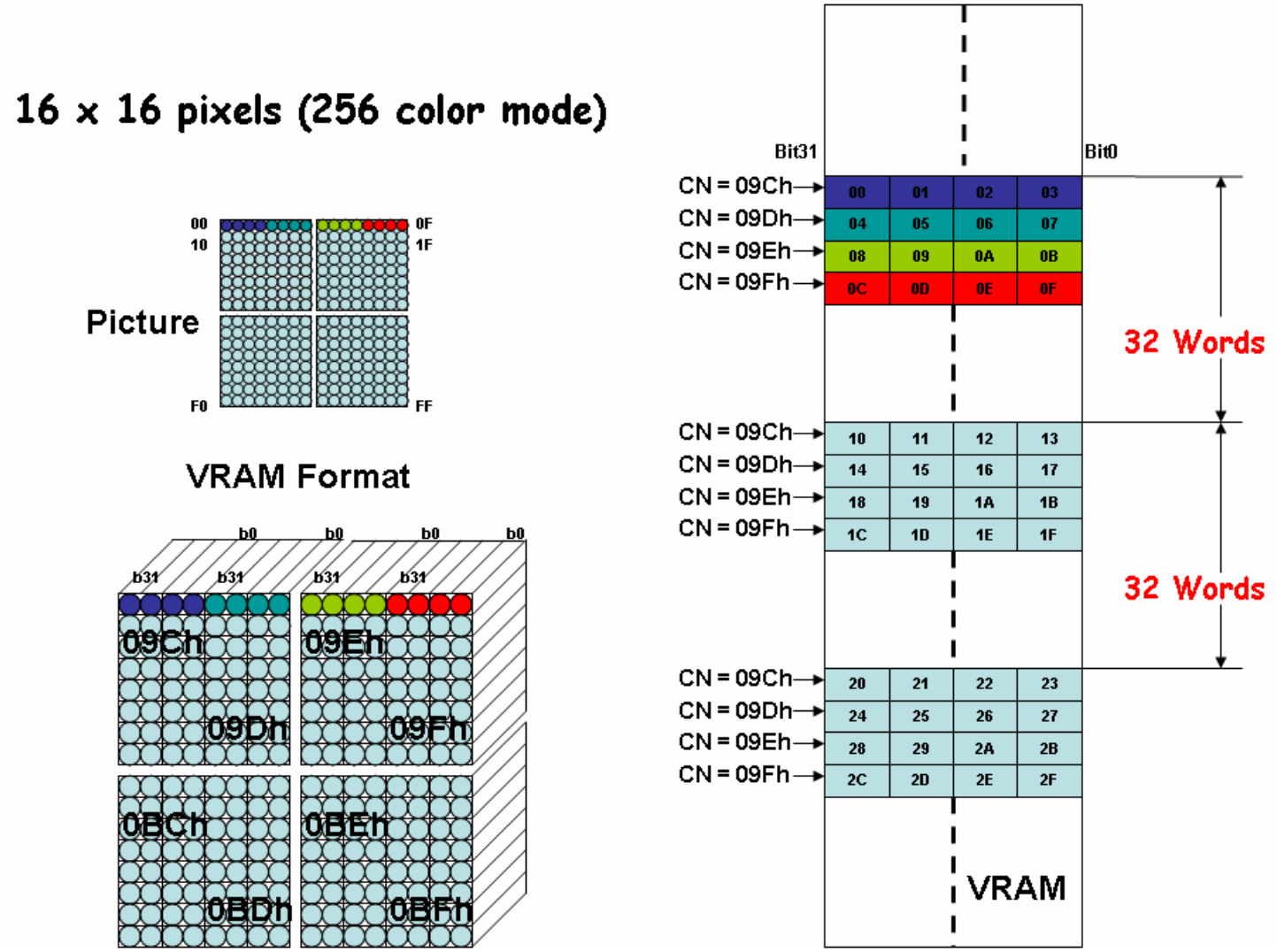


Figure 4.10-14.10-6 Sprite Pattern 16x16 Pixels Format in Cell Mode

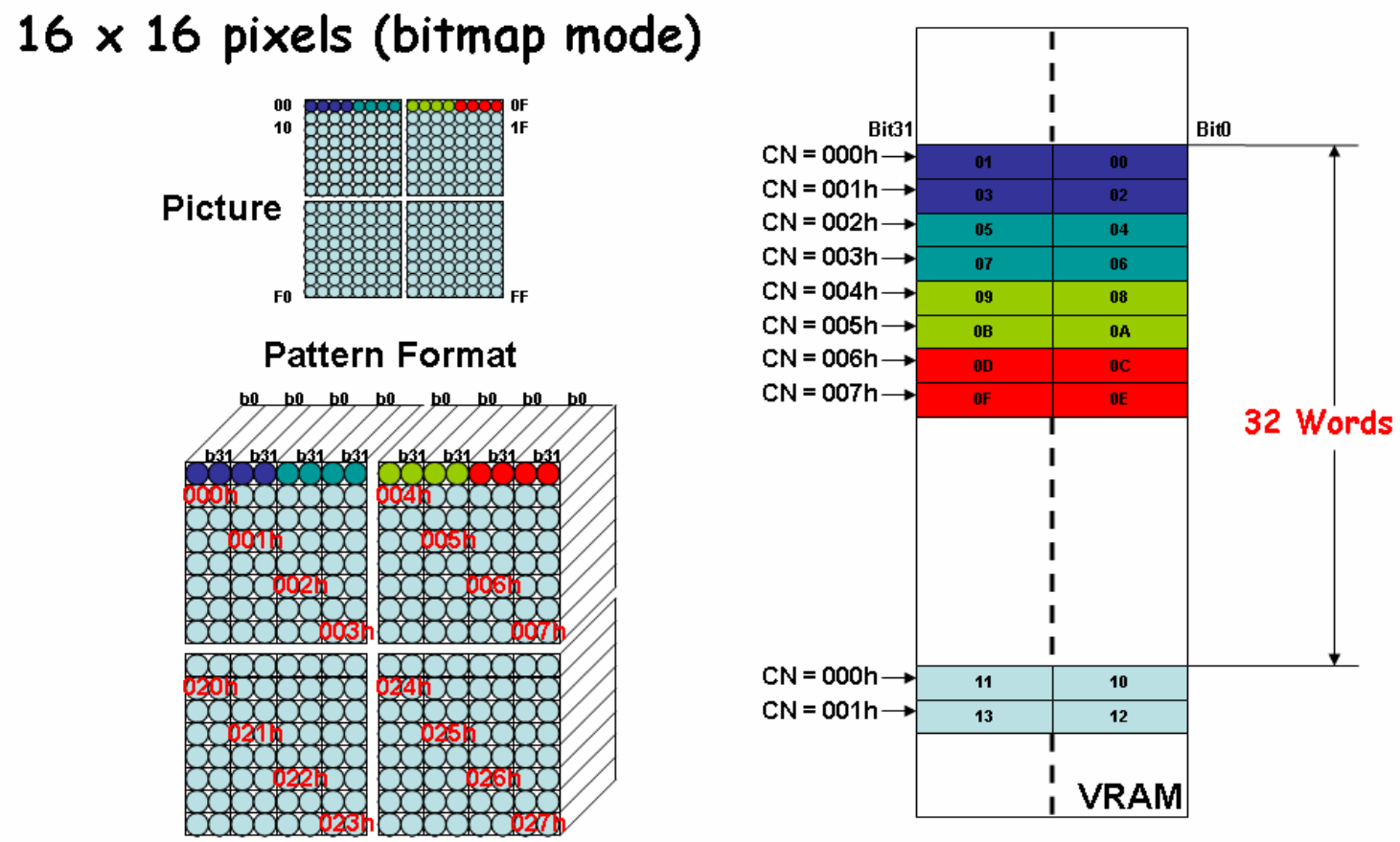


Figure 4.10-14.10-7 Sprite Pattern 16x16 Pixels Format in Bitmap Mode

4.10.3.2.4 Sprite Coordinate System

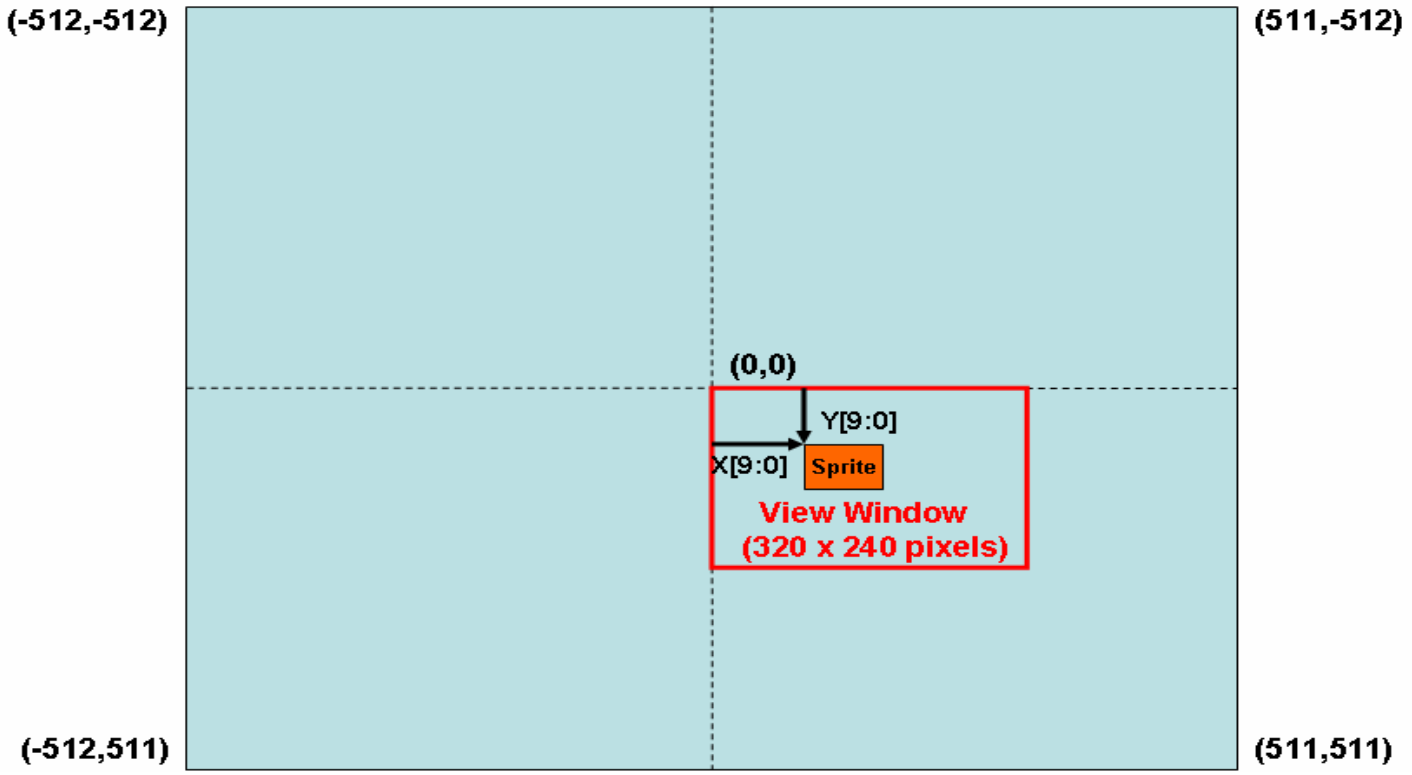


Figure 4.10-14.10-8 Sprite Coordinate System in QVGA Mode

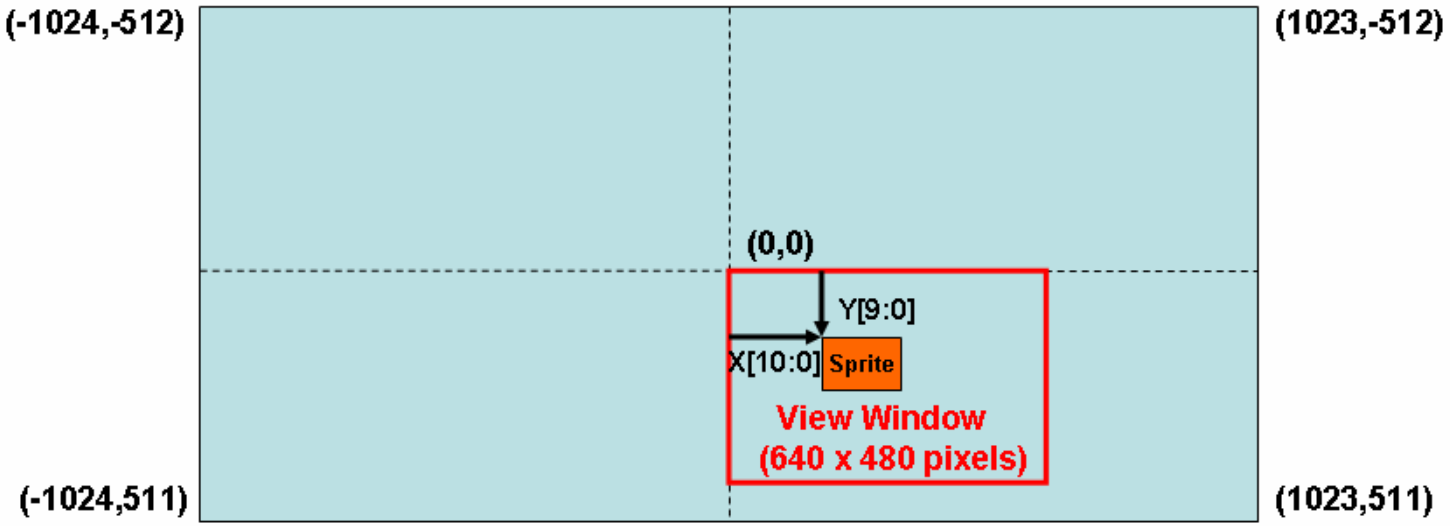


Figure 4.10-14.10-9 Sprite Coordinate System in VGA Mode

4.10.3.2.5 Sprite Transformation Parameter Memory (TPM) and Format

Support 16 sets of sprite Transformation Parameter Memory (TPM), each sprite TPM provides 12 bytes parameter for rotation/scaling implementation. The TPM are sized maximum 192 bytes and located on the last space of SAT RAM. The TPM formats are shown as below.

TPMx/TT0															
B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
Reserved			VSFM[4:0]					Reserved			VSFN[4:0]				
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Reserved			HSFM[4:0]					Reserved			HSFN[4:0]				
TPMx/TT1															
B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
TAFY[15:0]															
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
TAFX[15:0]															
TPMx/TT2															
B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
Reserved												TAFY[17:16]		TAFX[17:16]	
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Reserved		ACS	Reserved		OCT[2:0]			D[7:0]							

Bits	TPMx/TT0 Content Descriptions	
[31:29]	Reserved	Reserved
[28:24]	VSFM[4:0]	Vertical M scaling factor.
[23:21]	Reserved	Reserved
[20:16]	VSFN[4:0]	Vertical N scaling factor.
[15:13]	Reserved	Reserved
[12:8]	HSFM[4:0]	Horizontal M scaling factor.
[7:5]	Reserved	Reserved
[4:0]	HSFN[4:0]	Horizontal N scaling factor.
Bits	**TPMx/TT1 Content Descriptions**	
[31:16]	TAFY[15:0]	Bit 15 – 0 of Vertical Transformation Auxiliary Factor TAFY = $\frac{1}{INT*VSFM+VSFN}$ normalized to 40000h (18 bits precision)
[15:0]	TAFX[15:0]	Bit 15 – 0 of Horizontal Transformation Auxiliary Factor TAFX = $\frac{1}{INT*HSFM+HSFN}$ normalized to 40000h (18 bits precision)
Bits	**TPMx/TT2 Content Descriptions**	
[31:20]	Reserved	Reserved
[19:18]	TAFY[17:16]	Combined with TAFY[15:0] to form a 18-bit TAFY
[17:16]	TAFX[17:16]	Combined with TAFX[15:0] to form a 18-bit TAFX

[15:14]	Reserved	Reserved
[13]	ACS	R/S Across Control 1 = Enable 0 = Disable
[12:11]	Reserved	Reserved
[10:8]	OCT[2:0]	YX OCTant per 45 degree, clockwise enumeration.
[7:0]	D[7:0]	Rotation Degree, 45 degree in 8 bit pre-scaling.

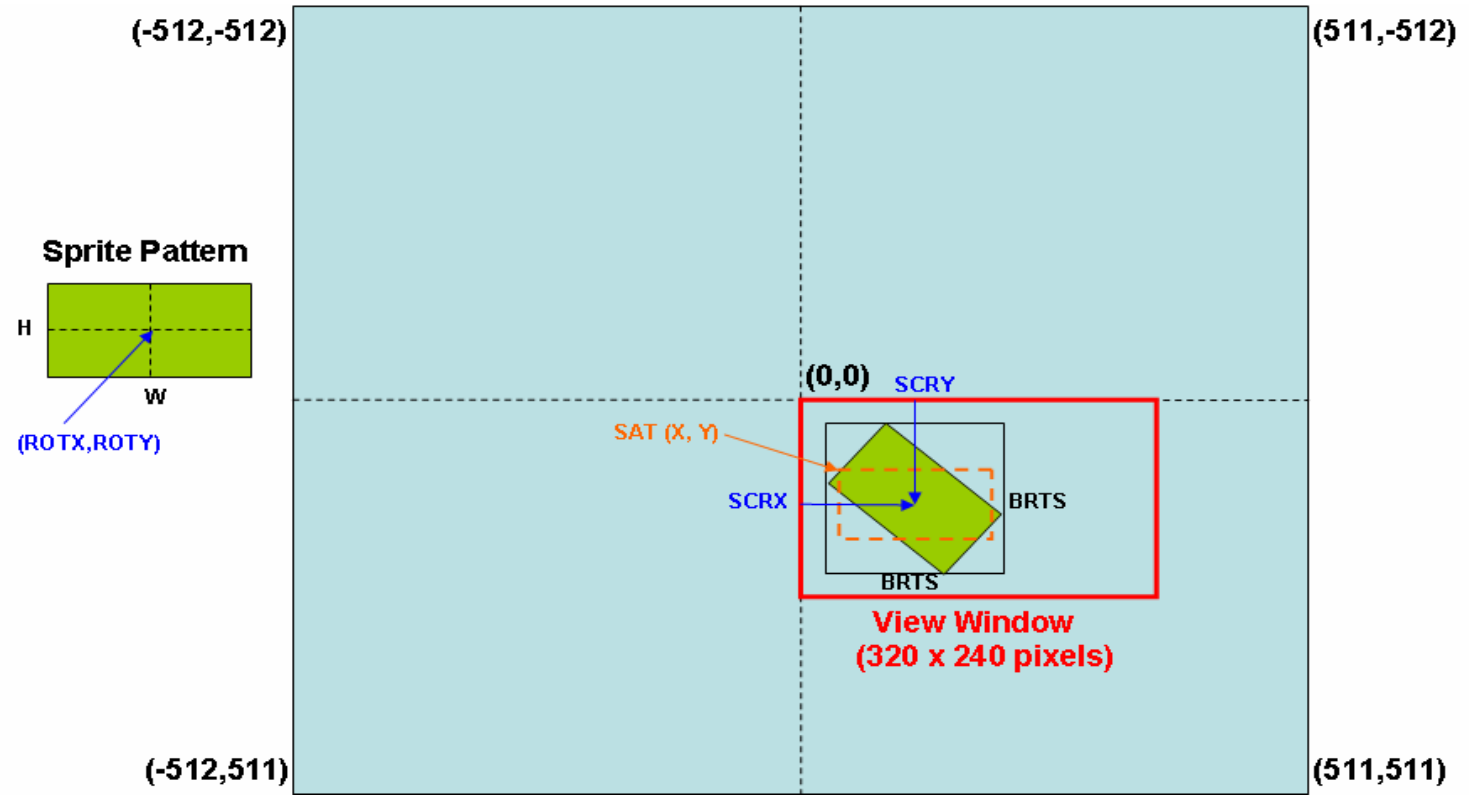


Figure 4.10-20 Sprite rotation / scaling in QVGA Mode

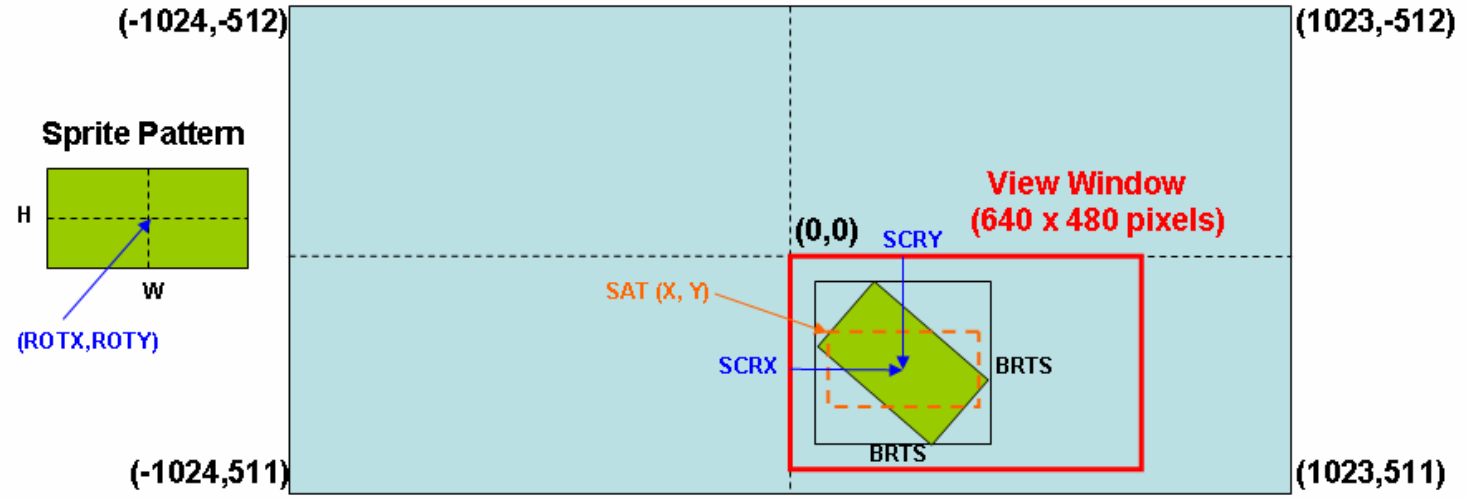


Figure 4.10-21 Sprite rotation / scaling in VGA Mode

4.10.3.2.6 Sprite Interface with 2D GE

Signal Descriptions:

- C2G_Req: CRTC Request Signal
- C2G_XYin: CRTC Input X-Y Coordinates (HCNT, VCNT)
- C2G_Flip[1:0]: CRTC V/H Flip
- C2G_WH[3:0]: CRTC Sprite Width/Height 8x8 ~ 64x64
- C2G_SCRN_XY: CRTC Sprite X-Y Coordinates on Screen
- C2G_TPM: CRTC Sprite Transform Parameter Memory Bus
- G2C_ReqEn: GE Request Enable to CRTC
- G2C_XYRdy: GE Output X-Y Ready
- G2C_XYout: GE Output X-Y Coordinates
- G2C_XYHit: GE Output X-Y Hit (Inside Region of Source Object)
- G2C_ScanComplete: GE Reaches the Boundary of Source Object

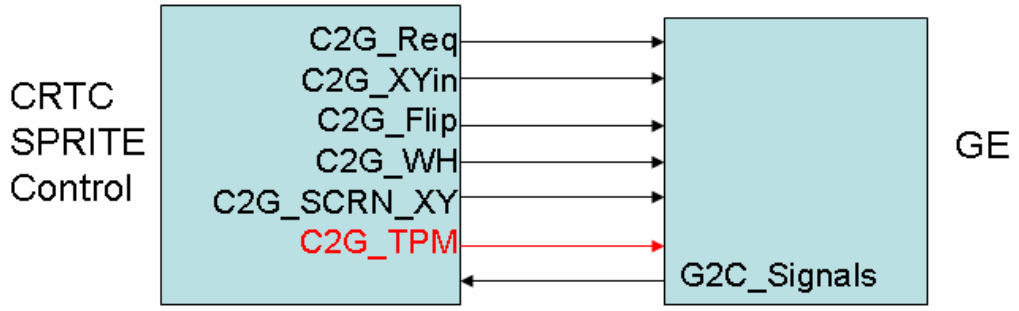


Figure 4.10-22 Sprite interface with 2D GE

C2G_TPM: (Transform Parameter Memory)
GPU to Graphics Engine Sprite Hardwired Signals

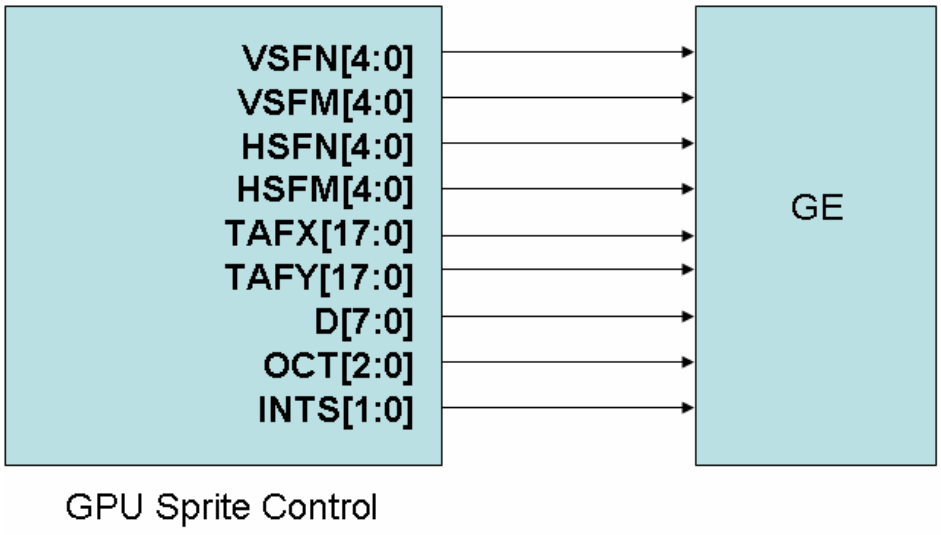


Figure 4.10-23 Sprite TPM Bus with 2D GE

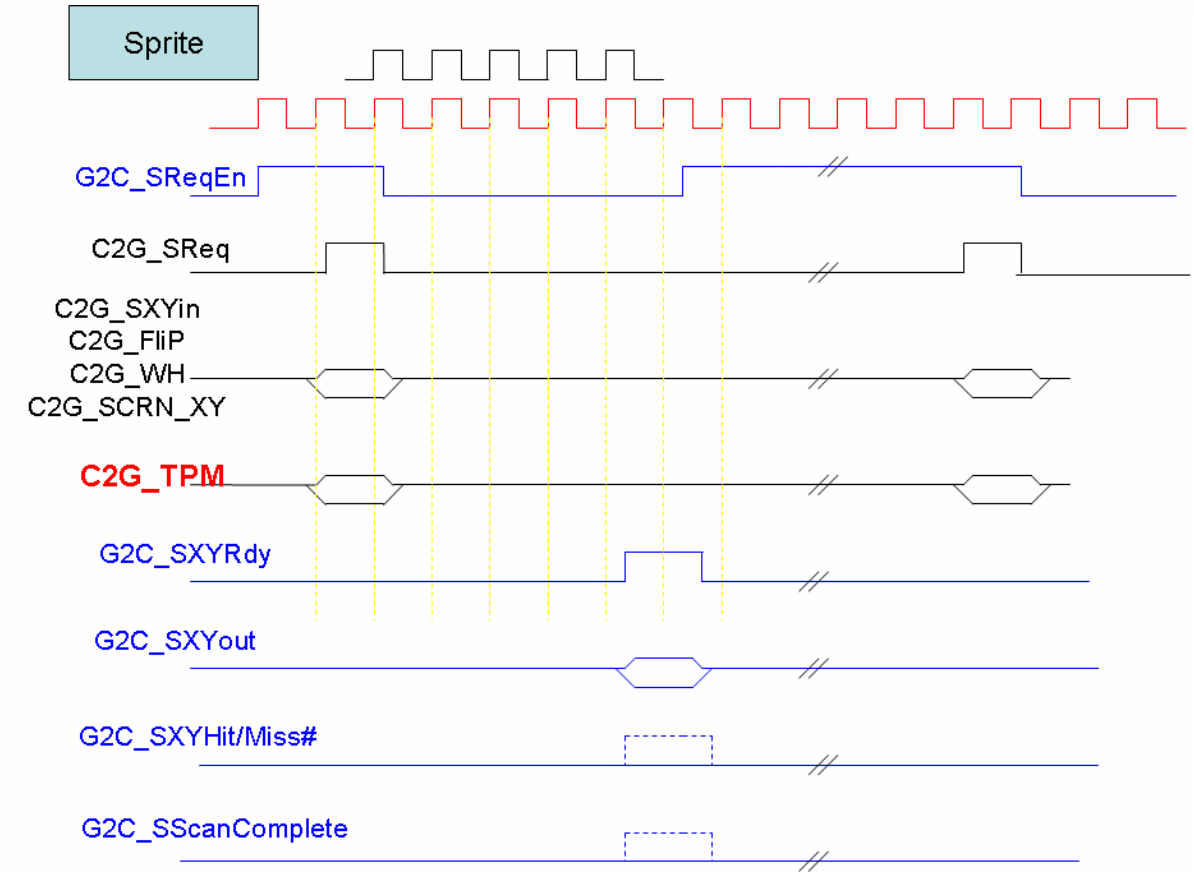


Figure 4.10-24 The timing of Sprite with 2D GE

4.10.3.2.7 Windows

Supports 2 windows, window0 and window1 that their areas are defined by SFR WIN0XY, WIN0WH, WIN1XY and WIN1WH. Also, provide two SFR WININ and WINOUT that control the BGs, Sprites and blending effect are inside or outside of these two windows. The priority is window 0 inside > window 1 inside > window 0 & 1 outside.

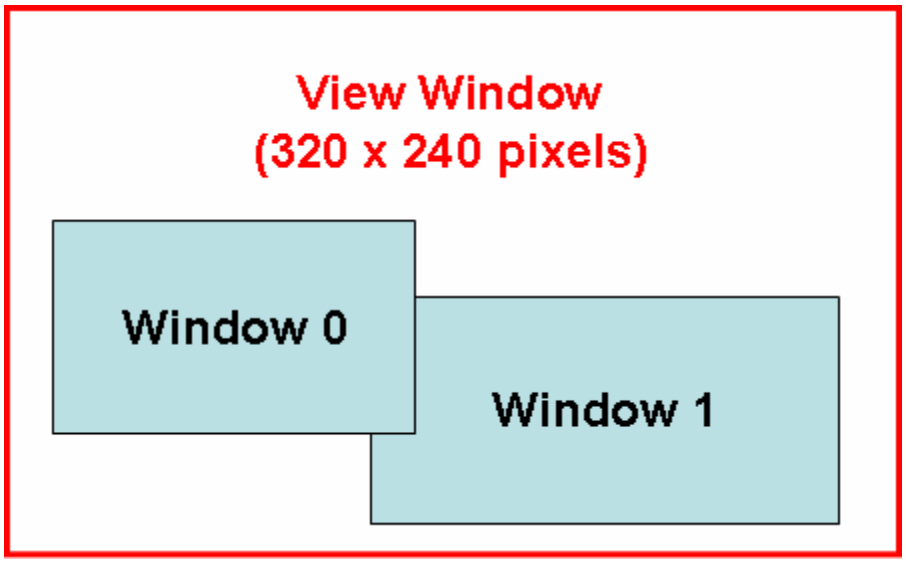


Figure 4.10-25 Windowed Control in QVGA Mode

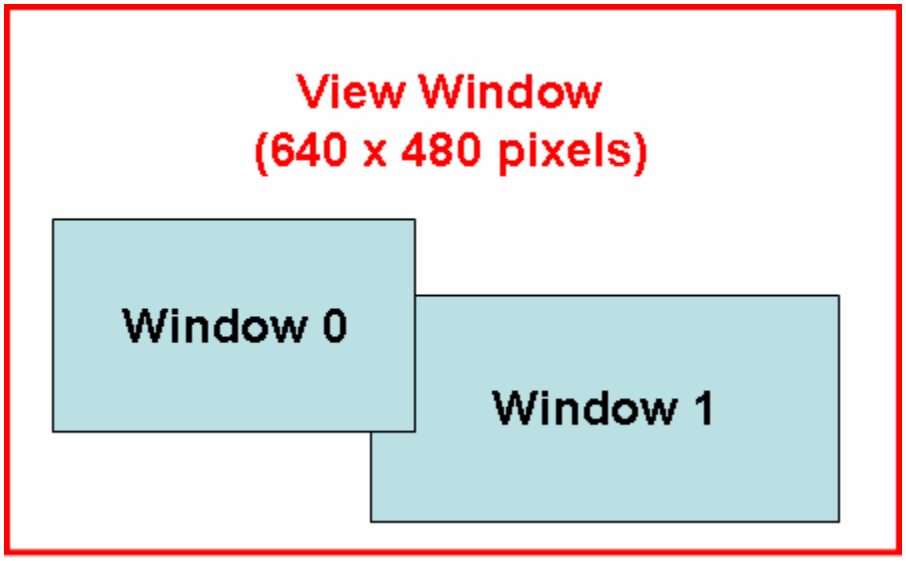


Figure 4.10-26 Windowed Control in VGA Mode

4.10.3.2.8 Compare Table of VGA and QVGA

	QVGA	VGA
Resolution	320*240	640*480
BAT		
Cell size of both Cell mode/ bitmap mode	8*8, 16*16	16*16, 32*32
The maximum virtual BG picture size	512*512	1024*1024
BG clipping window Y/X for rotation/ scaling	CWINY[8:0] CWINX[8:0]	CWINY[9:0] CWINX[9:0]
BG clipping window H/W for rotation/ scaling	CWINH[8:0] CWINW[8:0]	CWINH[9:0] CWINW[9:0]
BGx X-axis coordinate (BGxX), x = 0-3	BGxX[9:0] -512 ~ 511	BGxX[10:0] -1024 ~ 1023
BGx Y-axis coordinate (BGxY), x = 0-3	BGxY[9:0] -512 ~ 511	BGxY[10:0] -1024 ~ 1023
Sprite		
Vertical/ Horizontal size of Cell mode	8, 16, 32, 64	16, 32, 64, 128
Vertical/ Horizontal size of bitmap mode	8, 16, 32, 64	**8, 16, 32, 64**
Boundary rectangle size	384*384 288*288 192*192	768*768 576*576 384*384

	144*144 96*96 72*72 48*48 36*36 24*24 12*12 256*256 128*128 64*64 32*32 16*16 8*8	288*288 192*192 144*144 96*96 72*72 48*48 24*24 512*512 256*256 128*128 64*64 32*32 16*16
Sprite cell name, 16 color mode	CN[10:0]	CN[10:0]
Sprite cell name, 256 color mode and bitmap mode	CN[12:0]	CN[14:0]
X-axis location on screen	X[9:0]	X[9:0], EN
Window 0/1		
Window 0/1 Y/X Coordinate	WIN0Y[8:0] WIN0X[8:0]	WIN0Y[9:0] WIN0X[9:0]
Window 0/1 Height/ Width	WIN0H[8:0] WIN0W[8:0]	WIN0H[9:0] WIN0W[9:0]
Light pen		
Light pen scan line delay	0~15	2~30
Light pen X position	1 ~ 429 LPXY[26:18]	1 ~ 858 LPXY[27:18]
Mosaic		
Mosaic control	16, 8, 4, 2	32, 16, 8, 4

4.10.3.2.9 Color Effects

Alpha blending, Fad-in/Fade-out and invert color

ECS, ECT, ECY					Coeff.	ECS, ECT, ECY					Coeff.
0	0	0	0	0	0	0	1	0	0	0	8/16
0	0	0	0	1	1/16	0	1	0	0	1	9/16
0	0	0	1	0	2/16	0	1	0	1	0	10/16

0	0	0	1	1	3/16	0	1	0	1	1	11/16
0	0	1	0	0	4/16	0	1	1	0	0	12/16
0	0	1	0	1	5/16	0	1	1	0	1	13/16
0	0	1	1	0	6/16	0	1	1	1	0	14/16
0	0	1	1	1	7/16	0	1	1	1	1	15/16
						1	X	X	X	X	1

Alpha-blending
R' = Source pixel color R x ECS + Target pixel color R x ECT
G' = Source pixel color G x ECS + Target pixel color G x ECT
B' = Source pixel color B x ECS + Target pixel color B x ECT

Fade-in
R' = Source pixel color R + (31 – Source pixel color R) x ECY
G' = Source pixel color G + (31 – Source pixel color G) x ECY
B' = Source pixel color B + (31 – Source pixel color B) x ECY

Fade-out
R' = Source pixel color R – Source pixel color R x ECY
G' = Source pixel color G - Source pixel color G x ECY
B' = Source pixel color B - Source pixel color B x ECY

Invert color
R' = - Source pixel color R
G' = - Source pixel color G
B' = - Source pixel color B

4.10.3.2.10 Line Buffer Format

23	22	21	20	19	18	17	16
Reserved				MK	TG	PR[1:0]	
15	14	13	12	11	10	9	8
TRAN	R[4:0]					G[4:3]	
7	6	5	4	3	2	1	0
G[2:0]			B[4:0]				

Bits	Descriptions	
[23:20]	**Reserved**	**Reserved**
[19]	**MK**	**Mask** 1 = Sprite window mask 0 = Disable
[18:]	**TG**	**Blending Target**

		1 = Target 0 = Disable
[17:16]	**PR**	**Priority** 11 = the lowest priority 10 = 3rd 01 = 2nd 00 = the highest priority
[15]	**TRAN**	**Transparent** 1 = Transparent 0 = Color
[14:10]	**R**	**Red value**
[9:5]	**G**	**Green value**
[4:0]	**B**	**Blue value**

4.10.3.2.11 GPU Interrupts

There are five interrupt sources for GPU, scan line match (SLM) interrupt, horizontal synchronous (HS) interrupt, sprite horizontal blank (SHB) interrupt, background horizontal blank (BHB) interrupt, and vertical blank (VB) interrupt. The priority of these five interrupt sources is defined as the order SLM > HS > SHB > BHB > VB.

Scan line match (SLM) interrupt
When the SFR SLM[8:0] is matched with the scan line counter VCNT[8:0], the SLM event flag SLMEF will be set. And if the SLM interrupt enable SLMIE=1, the SLM interrupt will occur. SLMEF can be cleared only by the software.

Horizontal synchronous (HS) interrupt
When the horizontal synchronous signal occur (low pulse) in display active duration, the HS event flag HSEF will be set. And if the HS interrupt enable HSIE=1, the HS interrupt will occur. HSEF can be cleared by the software and hardware.

Sprite Horizontal blank (SHB) interrupt
When the horizontal blank of sprite layer occur in display active duration, the SHB event flag SHBEF will be set. And if the SHB interrupt enable SHBIE=1, the SHB interrupt will occur. SHBEF can be cleared by the software and hardware.

Background Horizontal blank (BHB) interrupt
When the horizontal blank of background layer occur in display active duration, the BHB event flag BHBEF will be set. And if the BHB interrupt enable HSIE=1, the BHB interrupt will occur. BHBEF can be cleared by the software and hardware.

Vertical blank (VB) interrupt
When the vertical blank occur in non-active display duration, the VB event flag VBEF will be set. And if the VB interrupt enable VBIE=1, the VB interrupt will occur. VBEF can be cleared only by the software.

4.10.3.2.12 Light Pen

If light pen enable, SPR LPE set, the falling edge of positive pulse from GPIO will trigger the light pen function to get the horizontal/vertical counter and store the current X/Y position to SFR LPX0, LPY0 and LPY1. Support 4-bits Light Pen Scan Line Delay (LPSLD) optional control register and 3-bits de-bounce (LPDB) optional control register to improve the stability of Light Pen input and get the suitable X-axis position.

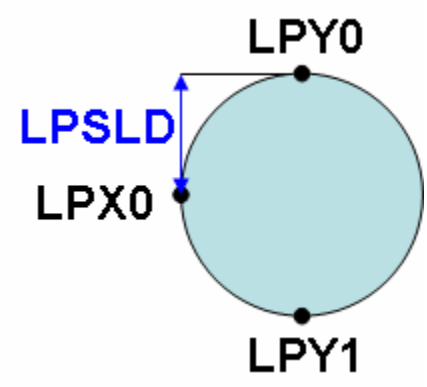


Figure 4.10-27 Light Pen

4.10.4 GPU 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
GPU_BA = 0xFFF0_4000				
GPUCR	GPU_BA+0x00	R/W	GPU Control Register	0x0000_0000
INTCR	GPU_BA+0x04	R/W	Interrupt Control Register	0x0000_0000
DISPSTAT	GPU_BA+0x08	R	Display Status Register	0x0000_1000
SLM	GPU_BA+0x0C	R/W	Scan Line Match Register	0x0000_0000
VCNT	GPU_BA+0x10	R	Vertical Counter Register	0x0000_0000
MSCTL	GPU_BA+0x14	R/W	Mosaic Control Register	0x0000_0000
LPCTL	GPU_BA+0x18	R/W	Light Pen Control Register	0x0000_0000
LPXY	GPU_BA+0x1C	R	Light Pen X/Y Register	0x0000_0000
CWINYX	GPU_BA+0x20	R/W	BG Clipping Window Y/X for Rotation/Scaling	0x0000_0000
CWINHW	GPU_BA+0x24	R/W	BG Clipping Window H/W for Rotation/Scaling	0x0000_0000
SPFN	GPU_BA+0x28	R/W	Sprite Frame Number Register	0x0000_0000
SPBMP	GPU_BA+0x2C	R/W	Sprite Bitmap Mode Switch Register	0x0000_0000
BG0CTL	GPU_BA+0x40	R/W	BG0 Control Register	0x0000_0000
BG0ATB	GPU_BA+0x44	R/W	BG0 Attribute Table Base	0x0000_0000
BG0VPB	GPU_BA+0x48	R/W	BG0 VROM Pattern Base	0x0000_0000
BG0XBC	GPU_BA+0x4C	R/W	BG0 X-axis Boundary Cell	0x0000_0000
BG0YBC	GPU_BA+0x50	R/W	BG0 Y-axis Boundary Cell	0x0000_0000
BG0X	GPU_BA+0x54	R/W	BG0 X-axis coordinate	0x0000_0000

BG0Y	GPU_BA+0x58	R/W	BG0 Y-axis coordinate	0x0000_0000
BG1CTL	GPU_BA+0x60	R/W	BG1 Control Register	0x0000_0000
BG1ATB	GPU_BA+0x64	R/W	BG1 Attribute Table Base	0x0000_0000
BG1VPB	GPU_BA+0x68	R/W	BG1 VROM Pattern Base	0x0000_0000
BG1XBC	GPU_BA+0x6C	R/W	BG1 X-axis Boundary Cell	0x0000_0000
BG1YBC	GPU_BA+0x70	R/W	BG1 Y-axis Boundary Cell	0x0000_0000
BG1X	GPU_BA+0x74	R/W	BG1 X-axis coordinate	0x0000_0000
BG1Y	GPU_BA+0x78	R/W	BG1 Y-axis coordinate	0x0000_0000
BG2CTL	GPU_BA+0x80	R/W	BG2 Control Register	0x0000_0000
BG2ATB	GPU_BA+0x84	R/W	BG2 Attribute Table Base	0x0000_0000
BG2VPB	GPU_BA+0x88	R/W	BG2 VROM Pattern Base	0x0000_0000
BG2XBC	GPU_BA+0x8C	R/W	BG2 X-axis Boundary Cell	0x0000_0000
BG2YBC	GPU_BA+0x90	R/W	BG2 Y-axis Boundary Cell	0x0000_0000
BG2X	GPU_BA+0x94	R/W	BG2 X-axis coordinate	0x0000_0000
BG2Y	GPU_BA+0x98	R/W	BG2 Y-axis coordinate	0x0000_0000
BG3CTL	GPU_BA+0xA0	R/W	BG3 Control Register	0x0000_0000
BG3ATB	GPU_BA+0xA4	R/W	BG3 Attribute Table Base	0x0000_0000
BG3VPB	GPU_BA+0xA8	R/W	BG3 VROM Pattern Base	0x0000_0000
BG3XBC	GPU_BA+0xAC	R/W	BG3 X-axis Boundary Cell	0x0000_0000
BG3YBC	GPU_BA+0xB0	R/W	BG3 Y-axis Boundary Cell	0x0000_0000
BG3X	GPU_BA+0xB4	R/W	BG3 X-axis coordinate	0x0000_0000
BG3Y	GPU_BA+0xB8	R/W	BG3 Y-axis coordinate	0x0000_0000
SPRVPB	GPU_BA+0xC0	R/W	Sprite VROM Pattern Base	0x0000_0000
WIN0YX	GPU_BA+0xC4	R/W	Window 0 Y/X Coordinate	0x0000_0000
WIN0HW	GPU_BA+0xC8	R/W	Window 0 Height/Width	0x0000_0000
WIN1YX	GPU_BA+0xCC	R/W	Window 1 Y/X Coordinate	0x0000_0000
WIN1HW	GPU_BA+0xD0	R/W	Window 1 Height/Width	0x0000_0000
WININ	GPU_BA+0xD4	R/W	Window Inside Control Register	0x0000_0000
WINOUT	GPU_BA+0xD8	R/W	Window Outside Control Register	0x0000_0000
ECST	GPU_BA+0xE0	R/W	Effect Coefficient S/T Register	0x0000_0000
ECY	GPU_BA+0xE4	R/W	Effect Coefficient Y Register	0x0000_0000

4.10.5 GPU Control Register Description

GPU Control Register (GPUCR)

Register	Address	R/W	Description	Reset Value
GPUCR	GPU_BA+0x00	R/W	GPU Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved					SPCNO	Reserved	Reserved
15	14	13	12	11	10	9	8
SPRWE	WIN1E	WIN0E	SPRE	BG3E	BG2E	BG1E	BG0E
7	6	5	4	3	2	1	0
BLKE	BLKCS		Reserved	CW	SP	SRSYN	SREN

Bits	Descriptions	
[31:19]	**Reserved**	**Reserved**
[18]	**SPCNO**	**Sprite Cell Name Option for 256 Color QVGA Mode** 1 = Extend the SC[3:2] as Cell Name bit14~13 0 = Not extend the SC[3:2] as Cell Name bit14~13
[17]	**Reserved**	**Reserved**
[16]	**Reserved**	**Reserved**
[15]	**SPRWE**	**Sprite Window Enable** 1 = Enable 0 = Disable
[14]	**WIN1E**	**Window 1 Enable** 1 = Enable 0 = Disable
[13]	**WIN0E**	**Window 0 Enable** 1 = Enable 0 = Disable
[12]	**SPRE**	**Sprite Layer Enable** 1 = Enable 0 = Disable
[11]	**BG3E**	**BG3 Layer Enable** 1 = Enable 0 = Disable
[10]	**BG2E**	**BG2 Layer Enable** 1 = Enable 0 = Disable
[9]	**BG1E**	**BG1 Layer Enable** 1 = Enable 0 = Disable
[8]	**BG0E**	**BG0 Layer Enable** 1 = Enable 0 = Disable
[7]	**BLKE**	**Bus Lock Enable for BG Rotation / Scaling BG** 1 = Enable 0 = Disable
[6:5]	**BLKCS**	**Bus Lock Clock Select for BG Rotation / Scaling** 11 = 256 system clocks locked by R/S BG layer 10 = 128 system clocks locked by R/S BG layer

		01 = 64 system clocks locked by R/S BG layer 00 = 32 system clocks locked by R/S BG layer
[4]	**Reserved**	**Reserved**
[3]	**CW**	**Clipping Window Control for BG rotation/scaling** 1 = Enable 0 = Disable
[2]	**SP**	**Speed Up Mode for BG rotation/scaling** 1 = Speed up BG operation, 1/2 0 = Normal BG operation
[1]	**SRSYN**	**Screen ON/OFF is synchronous to vertical sync.** 1 = Screen ON/OFF is synchronous to vertical sync. 0 = Screen ON/OFF is synchronous to horizontal sync.
[0]	**SREN**	**Screen ON/OFF** 1 = Enable 0 = Disable

Interrupt Control Register (INTCR)

Register	Address	R/W	Description	Reset Value
INTCR	0x04	R/W	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			**VBCF**	**BHBCF**	**SHBCF**	**HSCF**	**SLMCF**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**VBIE**	**BHBIE**	**SHBIE**	**HSIE**	**SLMIE**

Bits	Descriptions	
[31:13]	**Reserved**	**Reserved**
[12]	**VBCF**	**Clear Vertical Blank Event Flag** Write 1 to clear event flag
[11]	**BHBCF**	**Clear BG Horizontal Blank Event Flag** Write 1 to clear event flag
[10]	**SHBCF**	**Clear Sprite Horizontal Blank Event Flag** Write 1 to clear event flag
[9]	**HSCF**	**Clear Horizontal Sync. Event Flag** Write 1 to clear event flag
[8]	**SLMCF**	**Clear Scan Line Match Event Flag** Write 1 to clear event flag

[7:5]	**Reserved**	**Reserved**
[4]	**VBIE**	**Vertical Blank Interrupt Enable** 1 = Enable 0 = Disable
[3]	**BHBIE**	**BG Horizontal Blank Interrupt Enable** 1 = Enable 0 = Disable
[2]	**SHBIE**	**Sprite Horizontal Blank Interrupt Enable** 1 = Enable 0 = Disable
[1]	**HSIE**	**Horizontal Sync. Interrupt Enable** 1 = Enable 0 = Disable
[0]	**SLMIE**	**Scan Line Match Interrupt Enable** 1 = Enable 0 = Disable

Display Status Register (DISPSTAT)

Register	Address	R/W	Description	Reset Value
DISPSTAT	GPU_BA+0x08	R	Display Status Register	0x0000_1000

31	30	29	28	27	26	25	24
Reserved			**SPRCPL**	**BG3CPL**	**BG2CPL**	**BG1CPL**	**BG0CPL**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**SPRTO**	**BG3TO**	**BG2TO**	**BG1TO**	**BG0TO**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**VB**	**BHB**	**SHB**	**HS**	**VS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**VBEF**	**BHBEF**	**SHBEF**	**HSEF**	**SLMEF**

Bits	Descriptions	
[31:29]	**Reserved**	**Reserved**
[28]	**SPRCPL**	**Sprite Complete** *NOTE 1
[27]	**BG3CPL**	**BG3 Complete** *NOTE 1
[26]	**BG2CPL**	**BG2 Complete** *NOTE 1
[25]	**BG1CPL**	**BG1 Complete** *NOTE 1
[24]	**BG0CPL**	**BG0 Complete**

		*NOTE 1
[23:21]	**Reserved**	**Reserved**
[20]	**SPRTO**	**Sprite Time Out** *NOTE 1
[19]	**BG3TO**	**BG3 Time Out** *NOTE 1
[18]	**BG2TO**	**BG2 Time Out** *NOTE 1
[17]	**BG1TO**	**BG1 Time Out** *NOTE 1
[16]	**BG0TO**	**BG0 Time Out** *NOTE 1
[15:13]	**Reserved**	**Reserved**
[12]	**VB**	**Vertical Blank**
[11]	**BHB**	**BG Horizontal Blank**
[10]	**SHB**	**Sprite Horizontal Blank**
[9]	**HS**	**Horizontal Sync.**
[8]	**VS**	**Vertical Sync.**
[7:5]	**Reserved**	**Reserved**
[4]	**VBEF**	**Vertical Blank Event Flag** *NOTE 2
[3]	**BHBEF**	**BG Horizontal Blank Event Flag** *NOTE 1
[2]	**SHBEF**	**Sprite Horizontal Blank Event Flag** *NOTE 1
[1]	**HSEF**	**Horizontal Sync. Event Flag** *NOTE 1
[0]	**SLMEF**	**Scan Line Match Event Flag** *NOTE 2

*NOTE 1: Can be cleared by software and hardware.
*NOTE 2: Can be cleared only by software.

Scan Line Match Register (SLM)

Register	Address	R/W	Description	Reset Value
SLM	GPU_BA+0x0C	R/W	Scan Line Match 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							SLM[8]
7	6	5	4	3	2	1	0
SLM[7:0]							

Bits	Descriptions	
[31:9]	Reserved	Reserved
[8:0]	SLM	Scan Line Match (with VCNT)

Vertical Counter Register (VCNT)

Register	Address	R/W	Description	Reset Value
VCNT	GPU_BA+0x10	R	Vertical Counter 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							VCNT[8]
7	6	5	4	3	2	1	0
VCNT[7:0]							

Bits	Descriptions	
[31:9]	Reserved	Reserved
[8:0]	VCNT	Vertical Counter

Mosaic Control Register (MSCTL)

Register	Address	R/W	Description	Reset Value
MSCTL	GPU_BA+0x14	R/W	Mosaic Control Register	0x0000_0000

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

Bits	Descriptions	
[31:14]	**Reserved**	**Reserved**
[13:12]	**SPMV**	**Sprite Mosaic Vertical size,** 11 = 16 pixels for QVGA or Bitmap Mode, 32 pixels for VGA Mode 10 = 8 pixels for QVGA or Bitmap Mode, 16 pixels for VGA Mode 01 = 4 pixels for QVGA or Bitmap Mode, 8 pixels for VGA Mode 00 = 2 pixels for QVGA or Bitmap Mode, 4 pixels for VGA Mode
[11:10]	**Reserved**	**Reserved**
[9:8]	**SPMH**	**Sprite Mosaic Horizontal size** 11 = 16 pixels for QVGA or Bitmap Mode, 32 pixels for VGA Mode 10 = 8 pixels for QVGA or Bitmap Mode, 16 pixels for VGA Mode 01 = 4 pixels for QVGA or Bitmap Mode, 8 pixels for VGA Mode 00 = 2 pixels for QVGA or Bitmap Mode, 4 pixels for VGA Mode
[7:6]	**Reserved**	**Reserved**
[5:4]	**BGMV**	**BG Mosaic Vertical size** 11 = 16 pixels for QVGA Mode, 32 pixels for VGA Mode 10 = 8 pixels for QVGA Mode, 16 pixels for VGA Mode 01 = 4 pixels for QVGA Mode, 8 pixels for VGA Mode 00 = 2 pixels for QVGA Mode, 4 pixels for VGA Mode
[3:2]	**Reserved**	**Reserved**
[1:0]	**BGMH**	**BG Mosaic Horizontal size** 11 = 16 pixels for QVGA Mode, 32 pixels for VGA Mode 10 = 8 pixels for QVGA Mode, 16 pixels for VGA Mode 01 = 4 pixels for QVGA Mode, 8 pixels for VGA Mode 00 = 2 pixels for QVGA Mode, 4 pixels for VGA Mode

Light Pen Control Register (LPCTL)

Register	Address	R/W	Description	Reset Value
LPCTL	GPU_BA+0x18	R/W	Light Pen 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**
LPDBC[2:0]			**LPSLD[3:0]**				**LPE**

Bits	Descriptions	
[31:8]	**Reserved**	**Reserved**

[7:5]	**LPDBC**	**Light Pen De-bounce Control** 111 = 64 clocks de-bounce 110 = 32 clocks de-bounce 101 = 16 clocks de-bounce 100 = 8 clocks de-bounce 011 = 4 clocks de-bounce 010 = 2 clocks de-bounce 001 = 1 clock de-bounce 000 = no de-bounce
[4:1]	**LPSLD**	**Light Pen Scan Line Delay** 0 ~ 15 = Delay 0 ~ 15 scan lines to get LPX0[8:0]
[0]	**LPE**	**Light Pen Enable** 1 = Enable 0 = Disable

Light Pen X/Y Register (LPXY)

Register	Address	R/W	Description	Reset Value
LPXY	GPU_BA+0x1C	R	Light Pen X/Y Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**LPX0[9:6]**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
LPX0[5:0]						**LPY0[8:7]**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
LPY0[6:0]							**LPY1[8]**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LPY1[7:0]							

Bits	Descriptions	
[31:28]	**Reserved**	**Reserved**
[27:18]	**LPX0**	**Light pen X0 position** **QVGA: LPX0[8:0] (1 ~ 429)** **VGA: LPX0[9:0] (1 ~ 858)**
[17:9]	**LPY0**	**Light pen Y0 position (0 ~ 239)**
[8:0]	**LPY1**	**Light pen Y1 position (0 ~ 239)**

BG Clipping Window Y/X for Rotation/Scaling (CWINYX)

Register	Address	R/W	Description	Reset Value
CWINYX	GPU_BA+0x20	R/W	Clipping Window Y/X-axis coordinate	0x0000_0000

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

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25:16]	**CWINY**	**Clipping Window Y-axis coordinate** QVGA: CWINY[8:0] VGA: CWINY[9:0]
[15:10]	**Reserved**	**Reserved**
[9:0]	**CWINX**	**Clipping Window X-axis coordinate** QVGA: CWINX[8:0] VGA: CWINX[9:0]

BG Clipping Window H/W for Rotation/Scaling (CWINHW)

Register	Address	R/W	Description	Reset Value
CWIN0HW	GPU_BA+0x24	R/W	Clipping Window Height and Width	0x0000_0000

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

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25:16]	**CWINH**	**Clipping Window Height** QVGA: CWINH[8:0] VGA: CWINH[9:0]
[15:10]	**Reserved**	**Reserved**

[9:0]	**CWINW**	**Clipping Window Width** QVGA: CWINW[8:0] VGA: CWINW[9:0]

Sprite Frame Number Register (SPFN)

Register	Address	R/W	Description	Reset Value
SPFN	GPU_BA+0x28	R/W	Sprite 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**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPFN[7:0]							

Bits	Descriptions	
[31:8]	**Reserved**	**Reserved**
[7:0]	**SPFN**	**Sprite Frame Number (0 ~ 255), sprite number = SPFN + 1.**

Sprite bitmap switch register (SPBMP)

Register	Address	R/W	Description	Reset Value
SPBMP	GPU_BA+0x2C	R/W	Sprite Bitmap Mode Switch Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**SPBMP**	**Sprite Bitmap Mode Switch** **Mapping to the first 32 sprites, sprite0 ~ sprite31**

		1= Bitmap mode **0= Cell mode**

BGx Control Register (BGxCTL), x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0CTL	GPU_BA+0x40	R/W	BG0 Control Register	0x0000_0000
BG1CTL	GPU_BA+0x60	R/W	BG1 Control Register	0x0000_0000
BG2CTL	GPU_BA+0x80	R/W	BG2 Control Register	0x0000_0000
BG3CTL	GPU_BA+0xA0	R/W	BG3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**CPO**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TW	**MS**	**CF[1:0]**		**RS**	**MD**	**BM**	**WP**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**VFC**	**EXT**	**IC**	**CFC**	**HFC**	**CS**	**CM**

Bits	Descriptions	
[31:18]	**Reserved**	**Reserved**
[17:16]	**CPO**	**Color Palette RAM Option** 11 = this background use color palette RAM 3, 0xFFE73600 10 = this background use color palette RAM 2, 0xFFE73400 01 = this background use color palette RAM 1, 0xFFE73200 00 = this background use color palette RAM 0, 0xFFE73000
[15]	**TW**	**Transparent or Wraparound when underflow or overflow area** 1 = Wraparound 0 = Transparent
[14]	**MS**	**Mosaic** 1 = Enable 0 = Disable
[13:12]	**CF**	**Special Color Effect** 11 = Fade-out 10 = Fade-in 01 = Target 00 = Source
[11]	**RS**	**Rotate/Scale** 1 = Enable 0 = Disable

[10]	**MD**	**Mode** 1 = Bitmap mode 0 = Cell mode
[9]	**BM**	**Bitmap Mode** 1 = RGB555 mode 0 = 8-bit index mode
[8]	**WP**	**Wall Paper Mode (view window first cell)** 1 = Enable 0 = Disable
[7]	**Reserved**	**Reserved**
[6]	**VFC**	**Vertical Flip Control in 16 Color Mode** 1 = Enable 0 = Disable
[5]	**EXT**	**External Memory** 1 = BG patterns from internal VRAM 0 = BG patterns from external memory
[4]	**IC**	**Invert Color** 1 = Enable 0 = Disable
[3]	**CFC**	**Color Effect Control in 16 Color Mode** 1 = Enable 0 = Disable
[2]	**HFC**	**Horizontal Flip Control in 16 Color Mode** 1 = Enable 0 = Disable
[1]	**CS**	**Cell Size** 1 = 16 x 16 pixels in QVGA mode or 32 x 32 pixels in VGA mode. 0 = 8 x 8 pixels in QVGA mode or 16 x 16 pixels in VGA mode.
[0]	**CM**	**Color mode** 1 = 256 color mode 0 = 16 color mode

BGx Attribute Table Base (BGxATB) , x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0ATB	GPU_BA+0x44	R/W	BG0 Attribute Table Base	0x0000_0000
BG1ATB	GPU_BA+0x64	R/W	BG1 Attribute Table Base	0x0000_0000
BG2ATB	GPU_BA+0x84	R/W	BG2 Attribute Table Base	0x0000_0000
BGx3ATB	GPU_BA+0xA4	R/W	BG3 Attribute Table Base	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							BGxATB[8]
7	6	5	4	3	2	1	0
BGxATB[7:0]							

Bits	Descriptions	
[31:9]	Reserved	Reserved
[8:0]	BGxATB	BGx Attribute Table Base (0 ~ C0h blocks, each block = 128B)

BGx VROM Pattern Base (BGxVPB) , x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0VPB	GPU_BA+0x48	R/W	BG0 VROM Pattern Base	0x0000_0000
BG1VPB	GPU_BA+0x68	R/W	BG1 VROM Pattern Base	0x0000_0000
BG2VPB	GPU_BA+0x88	R/W	BG2 VROM Pattern Base	0x0000_0000
BG3VPB	GPU_BA+0xA8	R/W	BG3 VROM Pattern Base	0x0000_0000

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	BGxVPB	BGx VROM Pattern Base (0 ~ FFFFFFh blocks, each block = 256B) *NOTE 1 *NOTE 2

*NOTE 1: If BGxCTL[5] = 1 for BG to fetch pattern from the internal VRAM, the first block 0 (BGxVPB[23:0] == 0x000000) is mapping to the internal VRAM address 0xFFE7_6000.
*NOTE 2: BG VROM Pattern only can be located on the space that EBI can access it.

BGx X-axis Boundary Cell (BGxXBC) , x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0XBC	GPU_BA+0x4C	R/W	BG0 X-axis Boundary Cell	0x0000_0000
BG1XBC	GPU_BA+0x6C	R/W	BG1 X-axis Boundary Cell	0x0000_0000
BG2XBC	GPU_BA+0x8C	R/W	BG2 X-axis Boundary Cell	0x0000_0000
BG3XBC	GPU_BA+0xAC	R/W	BG3 X-axis Boundary Cell	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	**BGxXBC[6:0]**						

Bits	Descriptions	
[31:7]	**Reserved**	**Reserved**
[6:0]	**BGxXBC**	**BGx X-axis Boundary Cell (need 2^ - 1 in wrapper around)** BGxXBC[5:0] for QVGA mode BGxXBC[6:0] for VGA mode. NOTE: BGxXBC are defined in 8 pixels width no matter what cell size is.

BGx Y-axis Boundary Cell (BGxYBC) , x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0YBC	GPU_BA+0x50	R/W	BG0 Y-axis Boundary Cell	0x0000_0000
BG1YBC	GPU_BA+0x70	R/W	BG1 Y-axis Boundary Cell	0x0000_0000
BG2YBC	GPU_BA+0x90	R/W	BG2 Y-axis Boundary Cell	0x0000_0000
BG3YBC	GPU_BA+0xB0	R/W	BG3 Y-axis Boundary Cell	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	BGxYBC[6:0]						

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6:0]	BGxYBC	**BGx Y-axis Boundary Cell (need 2^ - 1 in wrapper around)** BGxYBC[5:0] for QVGA mode BGxYBC[6:0] for VGA mode. NOTE: BGxYBC are defined in 8 pixels Height no matter what cell size is.

BGx X-axis Coordinate (BGxX) , x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0X	GPU_BA+0x54	R/W	BG0 X-axis Coordinate	0x0000_0000
BG1X	GPU_BA+0x74	R/W	BG1 X-axis Coordinate	0x0000_0000
BG2X	GPU_BA+0x94	R/W	BG2 X-axis Coordinate	0x0000_0000
BG3X	GPU_BA+0xB4	R/W	BG3 X-axis Coordinate	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					BGxX[10:8]		
7	6	5	4	3	2	1	0
BGxX[7:0]							

Bits	Descriptions	
[31:11]	Reserved	Reserved
[10:0]	BGxX	**BGx X-axis coordinate** **QVGA: ranged from -512 to 511** **VGA: ranged from -1024 to 1023**

BGx Y-axis Coordinate (BGxY) , x = 0 – 3

Register	Address	R/W	Description	Reset Value
BG0Y	GPU_BA+0x58	R/W	BG0 Y-axis coordinate	0x0000_0000
BG1Y	GPU_BA+0x78	R/W	BG1 Y-axis coordinate	0x0000_0000

BG2Y	GPU_BA+0x98	R/W	BG2 Y-axis coordinate	0x0000_0000
BG3Y	GPU_BA+0xB8	R/W	BG3 Y-axis coordinate	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					**BGxY[10:8]**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BGxY[7:0]							

Bits	Descriptions	
[31:11]	**Reserved**	**Reserved**
[10:0]	**BGxY**	**BGx Y-axis coordinate** **QVGA: ranged from -512 to 511** **VGA: ranged from -1024 to 1023**

Sprite VROM Pattern Base (SPRVPB)

Register	Address	R/W	Description	Reset Value
SPRVPB	GPU_BA+0xC0	R/W	Sprite VROM Pattern Base	0x0000_0000

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

Bits	Descriptions	
[31:8]	**SPREVPB**	**Sprite External VROM Pattern Base (0 ~ FFFFFFh blocks, block = 256B)** *NOTE 1
[7:0]	**SPRIVPB**	**Sprite Internal VROM Pattern Base (0 ~ 5Fh blocks, block = 256B)**

*NOTE 1: Sprite External VROM Pattern only can be located on the space that EBI can access it.

Window 0 Y/X Coordinate (WIN0YX)

Register	Address	R/W	Description	Reset Value
WIN0YX	GPU_BA+0xC4	R/W	Window 0 Y/X-axis coordinate	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**WIN0Y[9:8]**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WIN0Y[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**WIN0X[9:8]**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WIN0X[7:0]							

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25:16]	**WIN0Y**	**Window 0 Y-axis coordinate** QVGA: WIN0Y[8:0] VGA: WIN0Y[9:0]
[15:10]	**Reserved**	**Reserved**
[9:0]	**WIN0X**	**Window 0 X-axis coordinate** QVGA: WIN0X[8:0] VGA: WIN0X[9:0]

Window 0 Height/Width (WIN0HW)

Register	Address	R/W	Description	Reset Value
WIN0HW	GPU_BA+0xC8	R/W	Window 0 Height and Width	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**WIN0H[9:8]**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WIN0H [7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**WIN0W[9:8]**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WIN0W[7:0]							

Bits	Descriptions

[31:26]	**Reserved**	**Reserved**
[25:16]	**WIN0H**	**Window 0 Height** QVGA: WIN0H[8:0] VGA: WIN0H[9:0]
[15:10]	**Reserved**	**Reserved**
[9:0]	**WIN0W**	**Window 0 Width** QVGA: WIN0W[8:0] VGA: WIN0W[9:0]

Window 1 Y/X Coordinate (WIN1YX)

Register	Address	R/W	Description	Reset Value
WIN1YX	GPU_BA+0xCC	R/W	Window 1 Y/X-axis coordinate	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**WIN1Y[9:8]**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WIN1Y[7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**WIN1X[9:8]**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WIN1X[7:0]							

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25:16]	**WIN1Y**	**Window 1 Y-axis coordinate** QVGA: WIN1Y[8:0] VGA: WIN1Y[9:0]
[15:10]	**Reserved**	**Reserved**
[9:0]	**WIN1X**	**Window 1 X-axis coordinate** QVGA: WIN1X[8:0] VGA: WIN1X[9:0]

Window 1 Height/Width (WIN1HW)

Register	Address	R/W	Description	Reset Value
WIN1HW	GPU_BA+0xD0	R/W	Window 1 Height and Width	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**WIN1H[9:8]**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**

WIN1H[7:0]							
15	14	13	12	11	10	9	8
Reserved						WIN1W[9:8]	
7	6	5	4	3	2	1	0
WIN1W[7:0]							

Bits	Descriptions	
[31:26]	**Reserved**	**Reserved**
[25:16]	**WIN1H**	**Window 1 Height** QVGA: WIN1H[8:0] VGA: WIN1H[9:0]
[15:10]	**Reserved**	**Reserved**
[9:0]	**WIN1W**	**Window 1 Width** QVGA: WIN1W[8:0] VGA: WIN1W[9:0]

Window Inside Control Register (WININ)

Register	Address	R/W	Description	Reset Value
WININ	GPU_BA+0xD4	R/W	Window Inside 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		CEF1	SPRD1	BG3D1	BG2D1	BG1D1	BG0D1
7	6	5	4	3	2	1	0
Reserved		CEF0	SPRD0	BG3D0	BG2D0	BG1D0	BG0D0

Bits	Descriptions	
[31:14]	**Reserved**	**Reserved**
[13]	**CEF1**	**Color Effect Flag of Window 1 Inside** 1 = Enable 0 = Disable
[12]	**SPRD1**	**Sprite Display Flag of Window 1 Inside** 1 = Display 0 = No display
[11]	**BG3D1**	**BG3 Display Flag of Window 1 Inside** 1 = Display

		0 = No display
[10]	**BG2D1**	**BG2 Display Flag of Window 1 Inside** 1 = Display 0 = No display
[9]	**BG1D1**	**BG1 Display Flag of Window 1 Inside** 1 = Display 0 = No display
[8]	**BG0D1**	**BG0 Display Flag of Window 1 Inside** 1 = Display 0 = No display
[7:6]	**Reserved**	**Reserved**
[5]	**CEF0**	**Color Effect Flag of Window 0 Inside** 1 = Enable 0 = Disable
[4]	**SPRD0**	**Sprite Display Flag of Window 0 Inside** 1 = Display 0 = No display
[3]	**BG3D0**	**BG3 Display Flag of Window 0 Inside** 1 = Display 0 = No display
[2]	**BG2D0**	**BG2 Display Flag of Window 0 Inside** 1 = Display 0 = No display
[1]	**BG1D0**	**BG1 Display Flag of Window 0 Inside** 1 = Display 0 = No display
[0]	**BG0D0**	**BG0 Display Flag of Window 0 Inside** 1 = Display 0 = No display

Window Outside Control Register (WINOUT)

Register	Address	R/W	Description	Reset Value
WINOUT	GPU_BA+0xD8	R/W	Window Outside 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**	**SPRDS**	**BG3DS**	**BG2DS**	**BG1DS**	**BG0DS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**CEFW**	**SPRDW**	**BG3DW**	**BG2DW**	**BG1DW**	**BG0DW**

Bits	Descriptions	
[31:13]	**Reserved**	**Reserved**
[12]	**SPRDS**	**Sprite Display Flag of Sprite Window Inside** 1 = Display 0 = No display
[11]	**BG3DS**	**BG3 Display Flag of Sprite Window Inside** 1 = Display 0 = No display
[10]	**BG2DS**	**BG2 Display Flag of Sprite Window Inside** 1 = Display 0 = No display
[9]	**BG1DS**	**BG1 Display Flag of Sprite Window Inside** 1 = Display 0 = No display
[8]	**BG0DS**	**BG0 Display Flag of Sprite Window Inside** 1 = Display 0 = No display
[7:6]	**Reserved**	**Reserved**
[5]	**CEFW**	**Color Effect Flag of Window 0, 1 Outside** 1 = Enable 0 = Disable
[4]	**SPRDW**	**Sprite Display Flag of Window 0, 1 Outside** 1 = Display 0 = No display
[3]	**BG3DW**	**BG3 Display Flag of Window 0, 1 Outside** 1 = Display 0 = No display
[2]	**BG2DW**	**BG2 Display Flag of Window 0, 1 Outside** 1 = Display 0 = No display
[1]	**BG1DW**	**BG1 Display Flag of Window 0, 1 Outside** 1 = Display 0 = No display
[0]	**BG0DW**	**BG0 Display Flag of Window 0, 1 Outside** 1 = Display 0 = No display

ECST Register (ECST)

Register	Address	R/W	Description	Reset Value
ECST	GPU_BA+0xE0	R/W	Effect Coefficient S/T 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			ECS[4:0]				
7	6	5	4	3	2	1	0
Reserved			ECT[4:0]				

Bits	Descriptions	
[31:5]	Reserved	Reserved
[12:8]	ECS	Effect Coefficient of Source
[7:5]	Reserved	Reserved
[4:0]	ECT	Effect Coefficient of Target

ECY Register (ECY)

Register	Address	R/W	Description	Reset Value
ECY	GPU_BA+0xE4	R/W	Effect Coefficient Y Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							SPRECY[4]
23	22	21	20	19	18	17	16
SPRECY[3:0]				BG3ECY[4:1]			
15	14	13	12	11	10	9	8
BG3ECY[0]	BG2ECY[4:0]					BG1ECY[4:3]	
7	6	5	4	3	2	1	0
BG1ECY[2:0]			BG0ECY[4:0]				

Bits	Descriptions	
[31:21]	Reserved	Reserved
[24:20]	SPRECY	Sprite Effect Coefficient Y
[19:15]	BG3ECY	BG3 Effect Coefficient Y
[14:10]	BG2ECY	BG2 Effect Coefficient Y
[9:5]	BG1ECY	BG1 Effect Coefficient Y
[4:0]	BG0ECY	BG0 Effect Coefficient Y

4.11 Flash Memory Interface Controller (FMI)

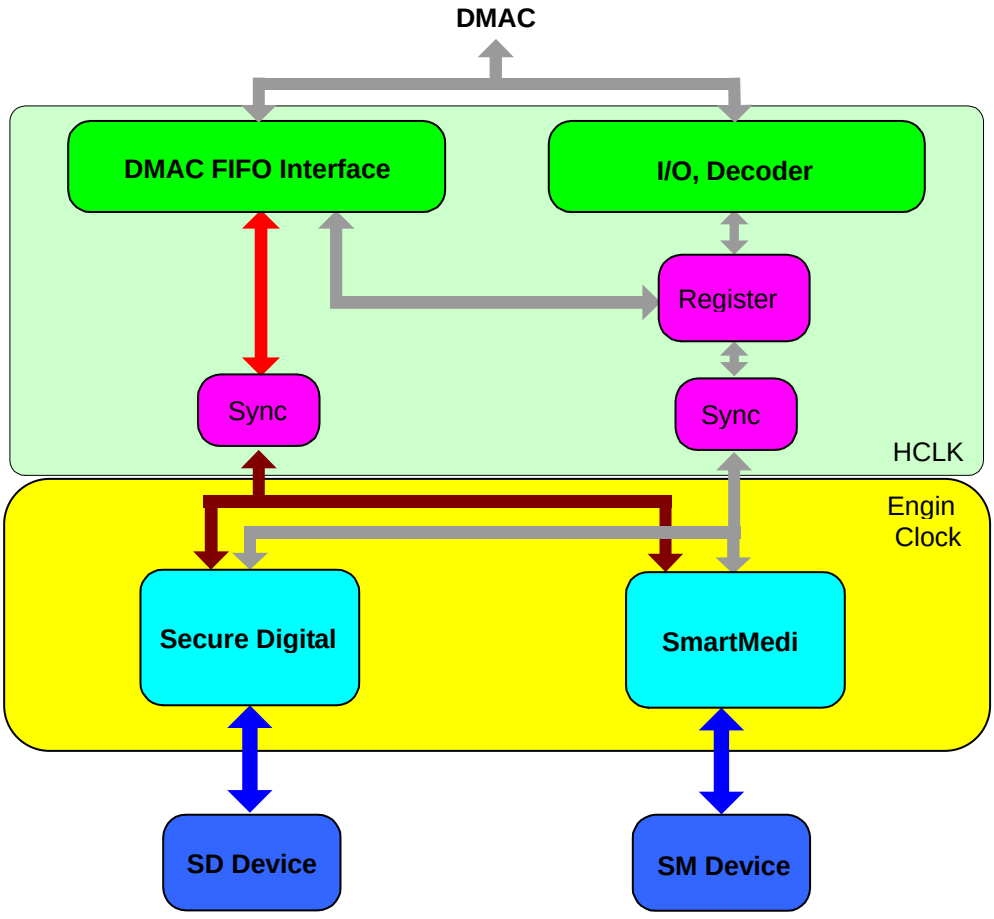
The Flash Memory Interface (FMI) of W55VA91 Chip supports SecureDigital (SD, SDIO & MMC) and NAND-type flash. FMI is cooperated with DMAC to provide a fast data transfer between system memory and cards. There is a single 64 bytes buffer embedded in DMAC for temporary data storage (separate into two 32 bytes ping-pong FIFO). Due to DMAC only has single channel, that means only one interface can be active at one time.

4.11.1 Features:

- Interface with DMAC for register read/write and data transfer
- 2 interfaces are provided: SecureDigital (2.0) and NAND-type (8-bit) flash
- Using single 64Bytes shared buffer for data exchange between system memory and cards. (separate into two 32 bytes ping-pong FIFO)
- Synchronous design for NAND-type flash interface with single clock domain, AHB bus clock (HCLK)
- Completely asynchronous design for SecureDigital with two clock domains, HCLK and Engine clock, note that frequency of HCLK should be higher than the frequency of engine clock.

4.11.2 Block Diagram

A simple block diagram of FMI Controller is shown as following.



FMI Controller Block Diagram

4.11.3 Function Description

SecureDigital (SD)

FMI provides an interface for SD card access. This SD controller provides 1 SD ports - The port have card detect function and SDIO interrupt. Each port can provide 1-bit/4-bit data bus mode for SD.

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 choice. 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/SmartMedia Controller.

FMI provides an interface for NAND-type Flash/SmartMedia access. It supports both 512+16bytes/page, 2048+64bytes/page and 4096+218bytes/page NAND. This NAND-type Flash controller provides all required signals for NAND flash, including R/–B –WP, –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 cycle to NAND, you should write 3 address 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.

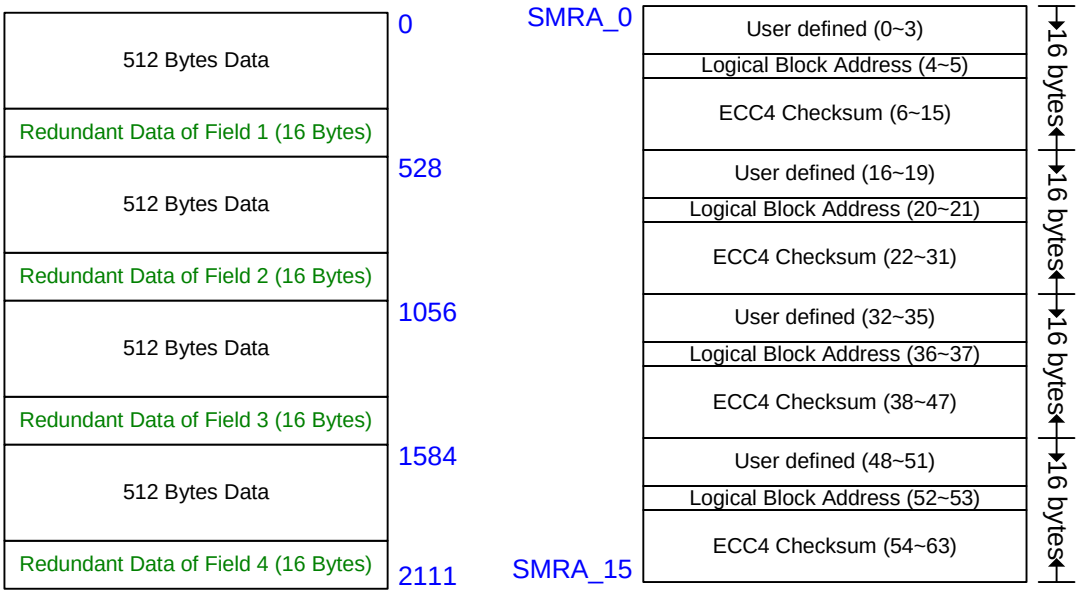
There are three page sizes for choice, 512+16bytes/page, 2048bytes/page and 4096+218bytes/page. Using SMCR[ECCX_EN] and SMCR[PSIZE] to select your NAND type. Software can use DMA function for

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.

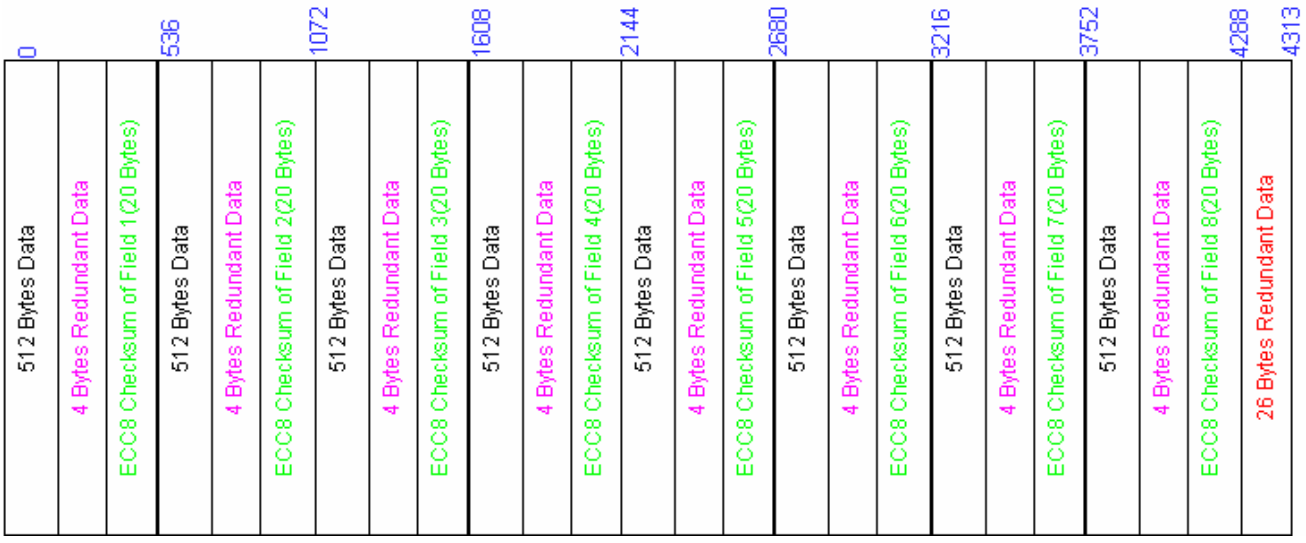
In error recovery part, there are two algorithms inside this NAND controller. One is the standard algorithm used in SmartMedia's specification which can correct 1 bit error and detect 2 bit errors for each field (256bytes or 512bytes). And the other is Reed-Solomon code which can correct and detect up to 4 errors or 8 errors for each field (512bytes).

For ECC1, data arrangement of redundant area will follow the rules defined in NAND/SmartMedia's specification. And software can determine the status via SMECCAD0, SMECCAD1. The correction information will be located at SMECCAD0, SMECCAD1.

For 2KB/Page NAND with ECC4, data arrangement of redundant area is shown below.



For 4KB/Page NAND with ECC8, data arrangement of redundant area is shown below.



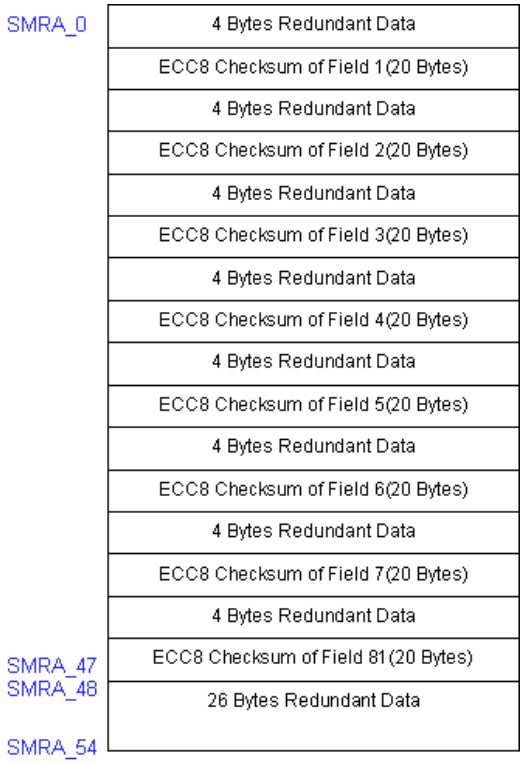


Fig : Distribution of the Redundant and ECC Checksum

2KB/Page NAND with ECC4 for old data arrangement of redundant area (*PAR_RW_EN* bit of **SMCRB** is set) is shown below.The ECC8 do the data arrangement as well.

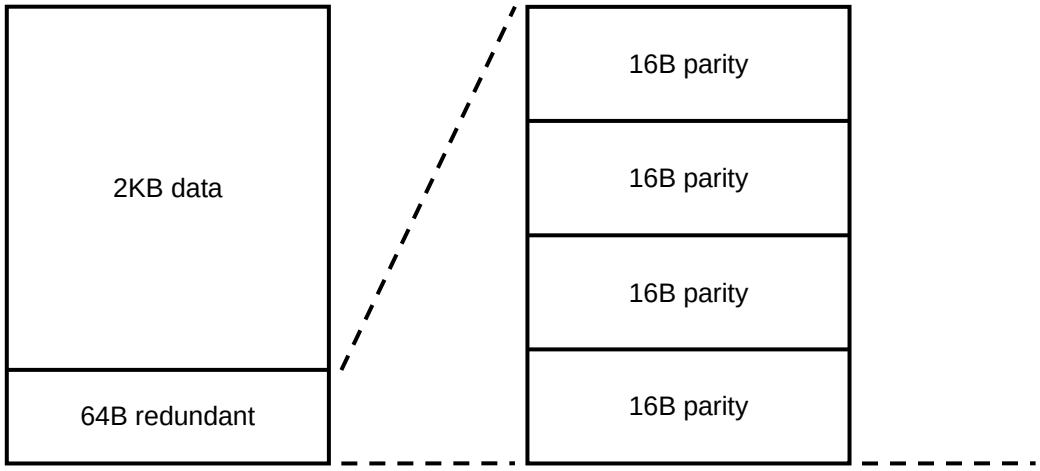


Fig. Old data arrangement

The figure above is for the old data arrangement for 2KB/page. In this mode, the readundant area of NAND must be programmed by the softare. The NAND controller will use DMAC to program the data area, but not redundant area.

4.11.4 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 = 0xFFF0_9800)				
FMICR	0x000	R/W	Global Control and Status Register	0x0000_0000
FMIIER	0x004	R/W	Global Interrupt Control Register	0x0000_0001
FMIISR	0x008	R/W	Global Interrupt Status Register	0x0000_0000
SecureDigital Registers (SD_BA = 0xFFF0_9820)				
SDCR	0x020	R/W	SD Control and Status Register	0x0101_0000
SDARG	0x024	R/W	SD Command Argument Register	0x0000_0000
SDIER	0x028	R/W	SD Interrupt Control Register	0x0000_0A00
SDISR	0x02C	R/W	SD Interrupt Status Register	0x000X_008C
SDRSP0	0x030	R	SD Receiving Response Token Register 0	0x0000_0000
SDRSP1	0x034	R	SD Receiving Response Token Register 1	0x0000_0000
SDBLEN	0x038	R/W	SD Block Length Register	0x0000_01FF
SDTMOUT	0x03C	R/W	SD Response/Data-in Time-out Register	0x0000_0000
SmartMedia/ NAND-type Flash Controller Registers (SM_BA = 0xFFF0_98A0)				
SMCR	0x0A0	R/W	SM/NAND Control and Status Register	0x0600_0080
SMTCR	0x0A4	R/W	SM/NAND Timing Control Register	0x0001_0105
SMIER	0x0A8	R/W	SM/NAND Interrupt Control Register	0x0000_0000
SMISR	0x0AC	R/W	SM/NAND Interrupt Status Register	0x000X_0000
SMCMD	0x0B0	W	SM/NAND Command Port Register	N/A
SMADDR	0x0B4	W	SM/NAND Address Port Register	N/A
SMDATA	0x0B8	R/W	SM/NAND Data Port Register	N/A
SM_ECC48_ST0	0x0D4	R	ECC4 Correction Status share with the four preceding statuses of the ECC8 Correction Status	0x0000_0000
SM_ECC48_ST1	0x0D8	R	The last four statuses of the ECC8 Correction Status	0x0000_0000
SM_ECC48_ADDR0	0x0DC	R	ECC4/8 error byte address for field 1-2	0x0000_0000
SM_ECC48_ADDR1	0x0E0	R	ECC4/8 error byte address for field 3-4	0x0000_0000
SM_ECC48_ADDR2	0x0E4	R	ECC8 error byte address for field 5-6	0x0000_0000
SM_ECC48_ADDR3	0x0E8	R	ECC8 error byte address for field 7-8	0x0000_0000
SM_ECC48_DATA0	0x0EC	R	ECC4/8 golden data byte for field 1-4	0x0000_0000
SM_ECC48_DATA1	0x0F0	R	ECC8 error byte address for field 5-8	0x0000_0000
SM_RA0 ... SM_RA54	0x100 ... 0x1D8	R/W	SM/NAND Redundant Area Register	N/A

4.11.5 FMI Controller Registers Map

FMI Global Control Register (FMICR)

Register	Offset	R/W	Description	Reset Value
FMICR	0x000	R/W	FMI Global 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			Reserved	SM_EN	Reserved	SD_EN	SW_RST

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	Reserved	Reserved
[3]	SM_EN	**SmartMedia/NAND-type Flash Functionality Enable** • 0 = Disable SM/NAND functionality of FMI. • 1 = Enable SM/NAND functionality of FMI.
[2]	Reserved	Reserved
[1]	SD_EN	**SecureDigital 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.

FMI Interrupt Control Register (FMIER)

Register	Offset	R/W	Description	Reset Value
FMIIER	0x004	R/W	FMI 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.

FMI Interrupt Status Register (FMISR)

Register	Offset	R/W	Description	Reset Value
FMIISR	0x008	R/W	FMI 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 Register (SDCR)

Register	Offset	R/W	Description	Reset Value
SDCR	0x020	R/W	SD Control Register	0x0101_0000

31	30	29	28	27	26	25	24
Reserved				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	
[31:28]	**Reserved**	**Reserved**
[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 Receieved** This field contains the block counts for data-in and data-out transfer. For **READ_MULTIPLE_BLOCK** and **WRITE_MULTIPLE_BLOCK** command, software can use this function to accelerate data transfer and improve performance. Note that only when SDBLEN=0x1FF, value 0x0 in this field means 256. Otherwise, don't fill 0x0 to this field. Note : For **READ_MULTIPLE_BLOCK** and **WRITE_MULTIPLE_BLOCK** command, the artual total lengh is BLK_CNT * BLKLEN[10:0].
[15]	**DBW**	**SD Data Bus Width (For 1-bit / 4-bit Selsction)** • 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. • 1 = Enable, SD host will output 8 clock cycles. **NOTE:** When operation is finished, this bit will be cleared automatically.
[5]	**CLK74_OE**	**Initial 74 Clock Cycles Output Enable** • 0 = No effect. • 1 = Enable, SD host will output 74 clock cycles to SD card. **NOTE:** When operation is finished, this bit will be cleared automatically.
[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 R2 response operation is finished, this bit will be cleared automatically.
[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 data output operation is finished, this bit will be cleared automatically.
[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 data input operation is finished, this bit will be cleared automatically.
[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 response input operation is finished, this bit will be cleared automatically.

[0]	**CO_EN**	**Command Output Enable** • 0 = No effect. • 1 = Enable, SD host will output a command to SD card. **NOTE:** When command output operation is finished, this bit will be cleared automatically.

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 Interrrupt Control Register	0x0000_0A00

31	30	29	28	27	26	25	24
Reserved	CD0SRC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	WKUP_EN	DITO_IE	RITO_IE	Reserved	SDIO0_IE	Reserved	CD0_IE
7	6	5	4	3	2	1	0
Reserved						CRC_IE	BLKD_IE

Bits	Descriptions	
[31]	**Reserved**	**Reserved**

[30]	**CD0SRC**	**SD0 Card Detect Source Selection** • 0 = From SD0 card's DAT3 pin. • 1 = From GPIO pin.
[29:15]	**Reserved**	**Reserved**
[14]	**WKUP_EN**	**Wake-Up Signal Generating Enable** Enable/Disable wake-up signal generating of SD host when SDIO card (current using) issues an interrupt (wke-up) via DAT[1] to host. • 0 = Disable. • 1 = Enable.
[13]	**DITO_IE**	**Data Input Time-out Interrupt Enable** Enable/Disable interrupt 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 interrupt 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 interrupt 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 interrupt 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				**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	
[31:20]	**Reserved**	**Reserved**
[19]	**Reserved**	**Reserved**
[18]	**SD0DAT1**	**DAT1 Pin Status of SD0 (Read Only)** This bit is the DAT1 pin status of SD0.
[17]	**Reserved**	**Reserved**
[16]	**CDPS0**	**Card Detect Status of SD0 (Read Only)** This bit is the card detect pin status of SD0, and it is using for card detection. When there is a card inserted in or removed from SD0, software should check this bit to confirm if there is really a card insertion or remove.
[15:14]	**Reserved**	**Reserved**

[13]	**DITO_IF**	**Data Input Time-out Interrupt Flag (Read Only)** This bit indicates that SD host counts to time-out value when receiving data (waiting start bit). • 0 = Not time-out. • 1 = Data input time-out. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[12]	**RITO_IF**	**Response Time-out Interrupt Flag (Read Only)** This bit indicates that SD host counts to time-out value when receiving response or R2 (waiting start bit). • 0 = Not time-out. • 1 = Response time-out. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[11]	**Reserved**	**Reserved**
[10]	**SDIO0_IF**	**SDIO 0 Interrupt Flag (Read Only)** This bit indicates that SDIO card 0 issues an interrupt to host. This interrupt is designed to level sensitive. Before clear it, turn off SDIER[SDIO0_IE] first. • 0 = No interrupt is issued by SDIO card 0. • 1 = An interrupt is issued by SDIO card 0. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[9]	**Reserved**	**Reserved**
[8]	**CD0_IF**	**SD0 Card Detection Interrupt Flag (Read Only)** This bit indicates that SD card 0 is inserted or removed. Only when SDIER[CD0_IE] is set to 1, this bit is active. • 0 = No card is inserted or removed. • 1 = There is a card inserted in or removed from SD0. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[7]	**SDDAT0**	**DAT0 Pin Status of Current Selected SD Port (Read Only)** This bit is the DAT0 pin status of current selected SD port.

[6:4]	**CRCSTAT**	**CRC Status Value of Data-out Transfer (Read Only)** SD host will record CRC status of data-out transfer. Software could use this value to identify what type of error is during data-out transfer. • 010 = Postive 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 ignare 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 finshed 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** A 11-bit value specifies the SD transfer byte count of a block. The actual byte count is equal to SDBLEN+1.

SD Response/Data-in Timeout 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.

SmartMedia/NAND-type Flash Control Register (SMCR)

Register	Address	R/W	Description	Reset Value
SMCR	FMI_BA+0x0A0	R/W	SM/ NAND Control Register	0x0601_0080

31	30	29	28	27	26	25	24
FAILED	FINISH	BIST_EN	Reserved		SM_CS		WP_
23	22	21	20	19	18	17	16
Reserved					SEC_CNT		
15	14	13	12	11	10	9	8
MECC4_8							
7	6	5	4	3	2	1	0
ECC4_8CHK	ECCX_EN		PAR_RW_EN	PSIZE	DWR_EN	DRD_EN	SW_RST

Bits	Descriptions	
[31]	**FAILED**	**BIST Failed** This bit indicates the BIST test of the embedded 216 Bytes buffer 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 216 Bytes buffers. 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.
[28:26]	**Reserved**	**Reserved**
[25]	**SM_CS**	**SmartMedia Card / NAND-type flash Enable** • 0 = card enable. • 1 = card disable.
[24:16]	**Reserved**	**Reserved**

[15:8]	**MECC4_8**	**Mask ECC4/8 During Write Page Data (ECC4/8 only)** These 4/8 bits indicate NAND controller to write out ECC4/8 parity or just 10/20 bytes 0xFF for each field. • 0 = Do not mask the ECC4/8 parity for chose field. • 1 = Mask ECC4/8 parity and write out 10/20 bytes 0xFF to NAND for chose field.
[7]	**ECC4_8CHK**	**None Used Field ECC4/8 Check After Read Page Data (ECC4/8 only)** • 0 = Disable. NAND controller will always check ECC4/8 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 ECC4/8 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 ECC4/8 check result. **Note:** when PAR_ALC_RWEN bit is set, it is recommended to disable this bit, regarding to the old data arrangement will not read/write the redundant data by HW.
[6:5]	**ECCX_EN**	**ECC Algorithm Selection** This field is used to select the ECC algorithm for data protecting. There are two ECC algorithms inside this NAND controller, one is the standard used in SmartMedia's specification and the other is Reed-Solomon code. For Reed-Solomon code, T can be 4 or 8 for choosing (correct 4 or 8 symbols). • 00 = reserved. • 01 = Using Reed-Solomon code encode/decode. (T=4) • 10 = Using Reed-Solomon code encode/decode. (T=8) • 11 = Cannot be this value, NAND controller will be abnormal.
[4]	**PAR_RW_EN**	**PARITY ALLOCATION READ-WRITE Enable** This field is used to read-write ECC parity data out from NAND Flash card or to NAND Flash by HW which is following each 512-Byte. • 0 = Enable H/W read/write parity data. In this mode, NAND controller will write ECC parity data both to SM_RA registers and NAND redundant area automatically. • 1 = Disable H/W read/write parity data. In this mode, NAND controller will write ECC parity data only SM_RA registers but not NAND redundant area. • **Note:** This bit is used to the old data arrangement only

[3]	**PSIZE**	**Page Size of NAND** This bit indicates the page size of NAND. Two types are supported. ECC4 (SMCR[ECCX_EN]=01), page size could be 512+16B or 2048+64B. For ECC8 (SMCR[ECCX_EN]=10), there is only single page size, 4096+218B per page. This bit will be ignored if SMCR[ECCX_EN]=10. • 0 = Page size is 512 Bytes. (512+16B) • 1 = Page size is 2048 Bytes. (2048+64B)
[2]	**DWR_EN**	**DMA Write Data Enable** This bit enables NAND controller to transfer data (1 page) from DMAC's embedded frame buffer into SmartMedia 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 SmartMedia 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.

SmartMedia//NAND-type Flash Timing Control Register (SMTCR)

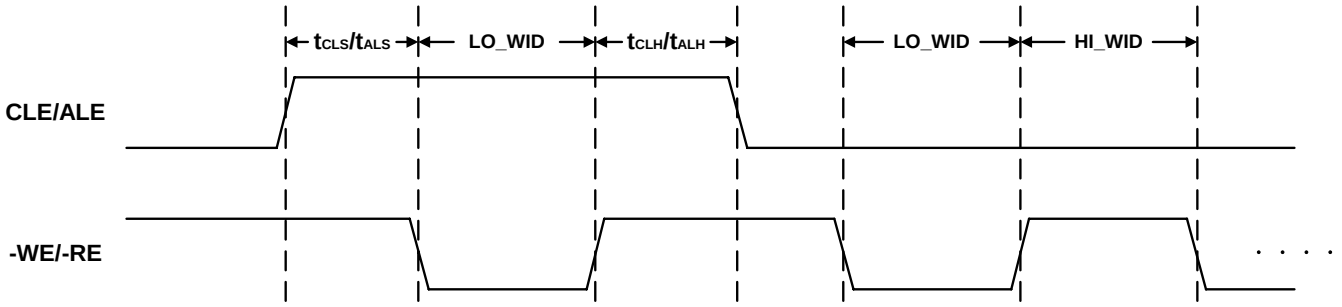
Register	Offset	R/W	Description	Reset Value
SMTCR	0x0A4	R/W	SM/NAND Timing Control Register	0x0001_0105

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)]) **NOTE:** Value of this field can not be 0x0.
[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)]) **NOTE:** Value of this field can not be 0x0.

NOTE1: The reset value is calculated base on 100MHz AHB Clock.



Timing Effect of Above 3 Registers

SmartMedia//NAND-type Flash Interrupt Control Register (SMIER)

Register	Address	R/W	Description	Reset Value
SMIER	FMI_BA+0x0A8	R/W	SM/NAND 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					RB_IE	Reserved	CD_IE
7	6	5	4	3	2	1	0
Reserved					ECC_FLD_IE	ECC_IE	DMA_IE

Bits	Descriptions	
[31:11]	Reserved	Reserved
[10]	RB_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]	Reserved	Reserved
[8]	CD_IE	**SM/NAND Card Detection Interrupt Enable** Enable/Disable interrupt generation of SM controller when card 0 is inserted or removed. • 0 = Disable. • 1 = Enable.
[7:3]	Reserved	Reserved
[2]	ECC_FLD_IE	**ECC Field Check Error Interrupt Enable** This bit can check the ECC error on each field (512bytes) of data transfer. Software can enable this bit to detect error and do error correction. • 0 = Disable. • 1 = Enable

[1]	**ECC_IE**	**ECC Check Error Interrupt Enable** When reading data from SM/NAND card, SM/NAND host will check the ECC code inside redundant area with the ECC code which calculated by itself. Enable this bit to generate interrupt when there is an ECC code mismatch. • 0 = Disable. • 1 = Enable.
[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.

SmartMedia//NAND-type Flash Interrupt Status Register (SMISR)

Register	Address	R/W	Description	Reset Value
SMISR	FMI_BA+0x0AC	R/W	SM/NAND Interrupt Status Register	0x000X_0000

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

Bits	Descriptions	
[31:19]	**Reserved**	**Reserved**
[18]	**RB_**	**Ready/-Busy Pin Status (Read Only)** This bit reflects the Ready/-Busy pin status of SmartMedia card.
[17]	**Reserved**	**Reserved**
[16]	**CD_**	**Card Detect Pin Status of SM (Read Only)** This bit is the card detect pin status of SM0, and it is using for card detection. When there is a card inserted in or removed from SM0, software should check this bit to confirm if there is really a card insertion or remove.
[15:11]	**Reserved**	**Reserved**

[10]	**RB_IF**	**Ready/-Busy Rising Edge Detect Interrupt Flag (Read Only)** • 0 = R/-B rising edge is not detected. • 1 = R/-B rising edge is detected. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[9]	**Reserved**	**Reserved**
[8]	**CD_IF**	**SM/NAND Card Detection Interrupt Flag (Read Only)** This bit indicates that SM/NAND card is inserted or removed. Only if SMIER[CD_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 SM. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[7:3]	**Reserved**	**Reserved**
[2]	**ECC_FLD_IF**	**ECC Field Check Error Interrupt Flag (Read Only)** This bit can check the ECC error on each field (512bytes) of data transfer. Software can read this bit to judge the error occurrence. • 0 = Disable. • 1 = Enable **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[1]	**ECC_IF**	**ECC Check Error Interrupt Flag (Read Only)** • 0 = No ECC mismatch occured. • 1 = ECC mismatch occurred. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.
[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.

SmartMedia/NAND-type Flash Command Port Register (SMCMD)

Register	Address	R/W	Description	Reset Value
SMCMD	FMI_BA+0x0B0	W	SM/NAND 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**	**SmartMedia/NAND Command Port** When CPU writes to this port, SM H/W circuit will send a command to SmartMedia card.

SmartMedia//NAND-type Flash Address Port Register (SMADDR)

Register	Address	R/W	Description	Reset Value
SMADDR	FMI_BA+0x0B4	W	SM/NAND 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/NAND host if this address is the last one or not. When software first writes to address port with this bit cleared, SM/NAND host will set ALE pin to active (HIGH). After the last address is written (with this bit set), SM/NAND 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**	**SmartMedia/NAND Address Port** When CPU writes to this port, SM/NAND H/W circuit will send an address to SmartMedia/NAND card.

SmartMedia//NAND-type Flash Data Port Register (SMDATA)

Register	Address	R/W	Description	Reset Value
SMDATA	FMI_BA+0x0B8	R/W	SM/NAND 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	**SmartMedia/NAND 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.

SmartMedia//NAND-type Flash ECC4/8 Error Status 0 (SM_ECC48_ST0)

Register	Address	R/W	Description	Reset Value
SM_ECC48_ST0	FMI_BA+0x0D4	R	SmartMedia/NAND ECC4/8 Error Status 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		F4_ECNT				F4_STAT	
23	22	21	20	19	18	17	16
Reserved		F3_ECNT				F3_STAT	
15	14	13	12	11	10	9	8
Reserved		F2_ECNT				F2_STAT	
7	6	5	4	3	2	1	0
Reserved		F1_ECNT				F1_STAT	

Bits	Descriptions	
[31:30]	Reserved	Reserved

[29:26]	**F4_ECNT**	**Error Count of ECC4/8-Field 4** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F4_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.
[25:24]	**F4_STAT**	**ECC4/8 Status of Field 4** This field contains the ECC4/8 correction status of ECC-field 4 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.
[23:22]	**Reserved**	**Reserved**
[21:18]	**F3_ECNT**	**Error Count of ECC4/8-Field 3** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F3_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.
[17:16]	**F3_STAT**	**ECC4/8 Status of Field 3** This field contains the ECC4/8 correction status of ECC-field 3 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.
[15:14]	**Reserved**	**Reserved**
[13:10]	**F2_ECNT**	**Error Count of ECC4/8-Field 2** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F2_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.

[9:8]	**F2_STAT**	**ECC4/8 Status of Field 2** This field contains the ECC4/8 correction status of ECC-field 2 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.
[7:6]	**Reserved**	**Reserved**
[5:2]	**F1_ECNT**	**Error Count of ECC4/8-Field 1** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F1_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.
[1:0]	**F1_STAT**	**ECC4/8 Status of Field 1** This field contains the ECC4/8 correction status of ECC-field 1 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.

SmartMedia//NAND-type Flash ECC4/8 Error Status 1 (SM_ECC48_ST1)

Register	Address	R/W	Description	Reset Value
SM_ECC48_ST1	FMI_BA+0x0D8	R	SmartMedia/NAND ECC4/8 Error Status 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		**F8_ECNT**				**F8_STAT**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**F7_ECNT**				**F7_STAT**	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**F6_ECNT**				**F6_STAT**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**F5_ECNT**				**F5_STAT**	

Bits	Descriptions	
[31:30]	**Reserved**	**Reserved**

[29:26]	**F8_ECNT**	**Error Count of ECC8-Field 8** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F8_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.
[25:24]	**F8_STAT**	**ECC8 Status of Field 8** This field contains the ECC4/8 correction status of ECC-field 8 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.
[23:22]	**Reserved**	**Reserved**
[21:18]	**F7_ECNT**	**Error Count of ECC8-Field 7** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F7_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.
[17:16]	**F7_STAT**	**ECC8 Status of Field 7** This field contains the ECC4/8 correction status of ECC-field 7 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.
[15:14]	**Reserved**	**Reserved**
[13:10]	**F6_ECNT**	**Error Count of ECC8-Field 6** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F6_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.

[9:8]	**F6_STAT**	**ECC8 Status of Field 6** This field contains the ECC4/8 correction status of ECC-field 6 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.
[7:6]	**Reserved**	**Reserved**
[5:2]	**F5_ECNT**	**Error Count of ECC8-Field 5** • This field conatins the error counts after ECC4/8 correct calculation. For this ECC4/8 core, it can correct up to 4/8 errors in a single field. Only when **F5_STAT** equals to 0x01, the value in this field is meaningful. The value could be 0x1 ~ 0x4/0x1 ~ 0x8; it means one error to four or eight errors depending on which ECC is used.
[1:0]	**F5_STAT**	**ECC8 Status of Field 5** This field contains the ECC4/8 correction status of ECC-field 5 (for page size 2048+64B/4096+218B). • 00 = No error. • 01 = Correctable error. • 10 = Uncorrectable error.

NOTE: SmartMedia/NAND host provide 1-bit error correction and 2-bit error detection. If there are more than two

NOTE: The SM_ECC48_ST0, 1 will be cleared when one of the H/W correction or local reset has done.

ECC4/8 Each Field Error Address 0 (SM_ECC48_ADDR0)

Register	Address	R/W	Description	Reset Value
SM_ECC48_ADDR0	FMI_BA+0x0DC	R	ECC4/8 Each Field Error Address 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							**F1_ADDR2**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
F1_ADDR2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**F1_ADDR1**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1_ADDR1							

Bits	Descriptions	
[31:25]	**Reserved**	**Reserved**
[24:16]	**F1_ADDR2**	**ECC4/8 Error Address 2 of Each-Field** This field contains a 9-bit ECC4/8 error address 2 of Each-field (for page size 2048+64B and 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA0[15:8] for correcting this error.
[15:9]	**Reserved**	**Reserved**
[8:0]	**F1_ADDR1**	**ECC4/8 Error Address 1 of Each-Field** This field contains a 9-bit ECC4/8 error address 1 of Each-field (for page size 2048+64B and 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA0[7:0] for correcting this error.

ECC48 Each Field Error Address 1 (SM_ECC48_ADDR1)

Register	Address	R/W	Description	Reset Value
SM_ECC48_ADDR1	FMI_BA+0x0E0	R	ECC4/8 Each Field Error Address 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							**F1_ADDR4**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
F1_ADDR4							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**F1_ADDR3**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1_ADDR3							

Bits	Descriptions	
[31:25]	**Reserved**	**Reserved**
[24:16]	**F1_ADDR4**	**ECC4/8 Error Address 4 of Each-Field** This field contains a 9-bit ECC4/8 error address 4 of Each-field (for page size 2048+64B and 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA0[31:24] for correcting this error.
[15:9]	**Reserved**	**Reserved**
[8:0]	**F1_ADDR3**	**ECC4/8 Error Address 3 of Each-Field** This field contains a 9-bit ECC4/8 error address 3 of Each-field (for page size 2048+64B and 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA0[23:16] for correcting this error.

ECC48 Each Field Error Address 2 (SM_ECC48_ADDR2)

Register	Address	R/W	Description	Reset Value
SM_ECC48_ADDR2	FMI_BA+0x0E4	R	ECC4/8 Each Field Error Address 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							F1_ADDR6
23	22	21	20	19	18	17	16
F1_ADDR6							
15	14	13	12	11	10	9	8
Reserved							F1_ADDR5
7	6	5	4	3	2	1	0
F1_ADDR5							

Bits	Descriptions	
[31:25]	Reserved	Reserved
[24:16]	F1_ADDR6	ECC8 Error Address 6 of Each-Field This field contains a 9-bit ECC8 error address 6 of Each-field (for page size 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA1[15:8] for correcting this error.
[15:9]	Reserved	Reserved
[8:0]	F1_ADDR5	ECC8 Error Address 5 of Each-Field This field contains a 9-bit ECC8 error address 5 of Each-field (for page size 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA1[7:0] for correcting this error.

ECC48 Last Field Error Address 3 (SM_ECC48_ADDR3)

Register	Address	R/W	Description	Reset Value
SM_ECC48_ADDR3	FMI_BA+0x0E8	R	ECC4/8 Each Field Error Address 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							F1_ADDR8
23	22	21	20	19	18	17	16
F1_ADDR8							
15	14	13	12	11	10	9	8
Reserved							F1_ADDR7
7	6	5	4	3	2	1	0
F1_ADDR7							

Bits	Descriptions	
[31:25]	**Reserved**	**Reserved**
[24:16]	**F1_ADDR8**	**ECC8 Error Address 8 of Each-Field** This field contains a 9-bit ECC8 error address 8 of Each-field (for page size 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA1[31:24] for correcting this error.
[15:9]	**Reserved**	**Reserved**
[8:0]	**F1_ADDR7**	**ECC8 Error Address 7 of Each-Field** This field contains a 9-bit ECC8 error address 7 of Each-field (for page size 4096+218B). If it is a correctable error, software can read the error data at SM_ECC48_DATA1[23:16] for correcting this error.

ECC48 Each Field Error Data (SM_ECC48_DATA0)

Register	Address	R/W	Description	Reset Value
SM_ECC48_DATA0	FMI_BA+0x0EC	R	ECC4/8 Each Field Error Data 0	0x0000_0000

31	30	29	28	27	26	25	24
F1_DATA4							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
F1_DATA3							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
F1_DATA2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1_DATA1							

Bits	Descriptions	
[31:24]	**F1_DATA4**	**ECC4/8 Error Data 4 of Each-Field** This field contains an 8-bit ECC4/8 error data 4 of Each-field (for page size 2048B+64B and 4096B+218B). 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 **F1_ADDR4**; the result will be the correct data.
[23:16]	**F1_DATA3**	**ECC4/8 Error Data 3 of Each-Field** This field contains an 8-bit ECC4/8 error data 3 of Each -field (for page size 2048B+64B and 4096B+218B). 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 **F1_ADDR3**; the result will be the correct data.

[15:8]	**F1_DATA2**	**ECC4/8 Error Data 2 of Each-Field** This field contains an 8-bit ECC4/8 error data 2 of Each -field (for page size 2048B+64B and 4096B+218B). 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 **F1_ADDR2**; the result will be the correct data.
[7:0]	**F1_DATA1**	**ECC4/8 Error Data 1 of Each-Field** This field contains an 8-bit ECC4/8 error data 1 of Each -field (for page size 2048B+64B and 4096B+218B). 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 **F1_ADDR1**; the result will be the correct data.

ECC48 Last Field Error Data (SM_ECC48_DATA1)

Register	Address	R/W	Description	Reset Value
SM_ECC48_DATA1	FMI_BA+0x0F0	R	ECC4/8 Each Field Error Data 1	0x0000_0000

31	30	29	28	27	26	25	24
F1_DATA8							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
F1_DATA7							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
F1_DATA6							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
F1_DATA5							

Bits	Descriptions	
[31:24]	**F1_DATA8**	**ECC8 Error Data 8 of Each-Field** This field contains an 8-bit ECC8 error data 8 of Each-field (for page size 4096B+218B). 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 **F1_ADDR8**; the result will be the correct data.
[23:16]	**F1_DATA7**	**ECC8 Error Data 7 of Each-Field** This field contains an 8-bit ECC8 error data 3 of Each-field (for page size 4096B+218B). 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 **F1_ADDR7**; the result will be the correct data.
[15:8]	**F1_DATA6**	**ECC8 Error Data 6 of Each-Field** This field contains an 8-bit ECC8 error data 2 of Each-field (for page size 4096B+218B). 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 **F1_ADDR6**; the result will be the correct data.

[7:0]	**F1_DATA5**	**ECC8 Error Data 5 of Each-Field** This field contains an 8-bit ECC8 error data 1 of Each-field (for page size 4096B+218B). 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 **F1_ADDR5**; the result will be the correct data.

SmartMedia/NAND-type Flash Redundant Area Register (SMRA)

Register	Address	R/W	Description	Reset Value
SM_RA0 **…** **SM_RA54**	FMI_BA+0x100 … FMI_BA+0x1D8	R/W	SM/NANDRedundant Area Register	N/A

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

Bits	Descriptions	
[31:0]	**RArea**	**Redundant Area** For ECC4 and ECC8: (new data arrangement) This field keeps the 64 bytes data of redundant area for flash memory which page size is 2KBytes. When **WRITE** to NAND: For 512+16Bytes/page models, only preceding 16 bytes redundant data are required. (i.e. Those 16 bytes redundant data must be programmed into SM_RA0, SM_RA1, SM_RA2 and SM_RA3.) For 2048+64Bytes/page models, the redundant data is programmed into SM_RA0 to SM_RA15. When **READ** from NAND: For 512+16 bytes/page models, the received redundant data will be placed at SM_RA0, SM_RA1, SM_RA2 and SM_RA3. For 2048+64 bytes/page models, the received redundant data will be placed at SM_RA0 to SM_RA15. For ECC8: (new data arrangement) The SM_RA0, SM_RA6, SM_RA12, SM_RA18, SM_RA24, SM_RA30, SM_RA36, SM_RA42 is a part from the 58 bytes Redundant data, and the 26 bytes of the SM_RA48 ~ SM_RA53 is the remaining of the redundant data are used by software arbitrarily. The others of the SM_RAxx redundant data (SM_RA1~SM_RA5, SM_RA7~SM_RA11, SM_RA13~SM_RA17, SM_RA19~SM_RA23, SM_RA25~SM_RA29, SM_RA31~SM_RA35, SM_RA37~SM_RA41, SM_RA43~SM_RA47) are ECC8 parity of each field. When NAND controller WRITEs data in NAND flash, it will also generate 20 bytes ECC8 parity for each field, and write these parity data both into NAND flash and relative redundant area. After all data and parity data have been transferred, the final 26 bytes redundant data are written. When NAND controller READ from NAND flash, those parity data and redundant data which stored in NAND and final 26 bytes redundant data will be put at these registers. For ECC4/ECC8: (old data arrangement) The SM_RA0~ SM_RA9(ECC4) are the ECC4 parity data, The SM_RA0~ SM_RA39(ECC8) are the ECC8 parity data.

4.12 FMI DMA Controller

The DMA Controller provides a DMA (Direct Memory Access) function for FMI to exchange data between system memory (ex. SDRAM) and shared buffer (64 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.

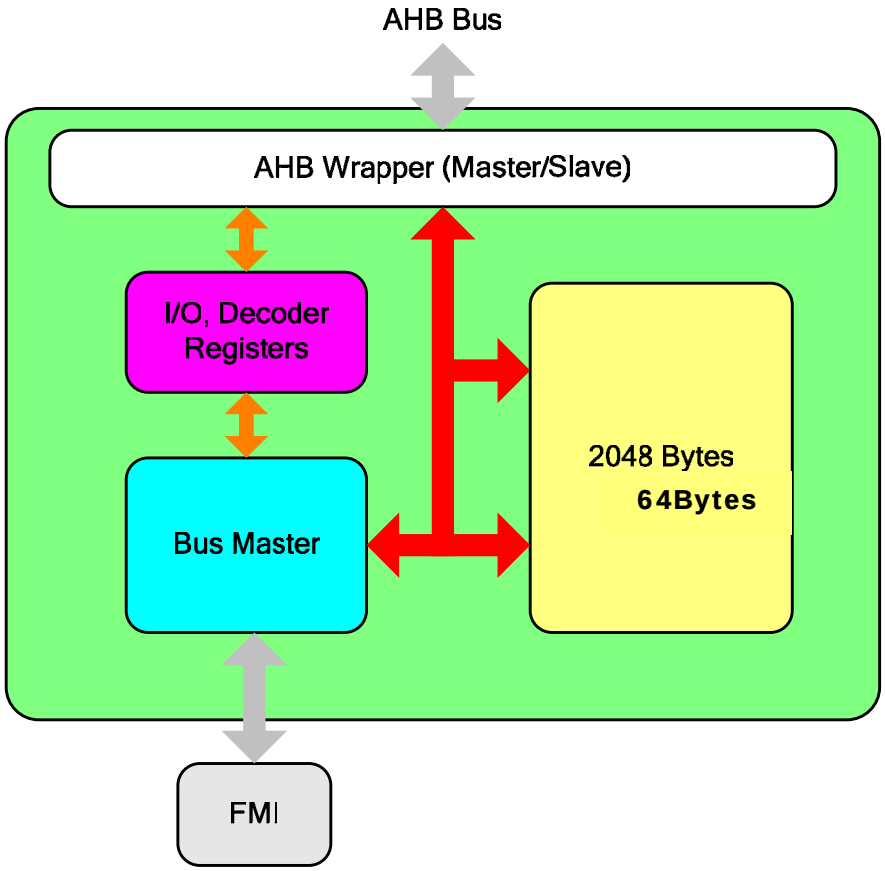
There is a 64 bytes shared buffer inside DMAC, separate into two 32 bytes ping-pong FIFO. It can provide multi-block transfers using ping-pong mechanism for FMI. Software can access these shared buffers directly when FMI is not in busy.

4.12.1 Features:

- AMBA AHB master/slave interface compatible, for data transfer and register read/write
- Support single DMA channel
- One 64 bytes shared buffer is embedded(separate into two 32 bytes ping-pong FIFO)
- Synchronous design with single clock domain, AHB bus clock (HCLK)

4.12.2 Block Diagram

The block diagram of DMA Controller is shown as following.



DMA Controller Block Diagram

4.12.3 Programming Flow

Here is a simple example programming flow with DMA enable.
1. Set DMACCSR[DMACEN] to enable DMAC.
2. Fill corresponding starting address in DMACSAR2 for FMI.
3. Enable IP to start DMA transfer.
4. Wait IP finished, software doesn't need to take care of DMAC.

4.12.4 DMA Controller Registers Map

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

Register	Offset	R/W	Description	Reset Value
Shared Buffer (DMAC_BA = 0xFFF0_9000)				
FB_0 **……** **FB_15**	DMAC_BA+0x000 …… DMAC_BA+0x03C	R/W	Shared Buffer (FIFO)	0x0000_0000
DMAC Registers (DMAC_BA = 0xFFF0_9400)				
DMACCSR	DMAC_BA+0x00	R/W	DMAC Control and Status Register	0x0000_0000
DMACSAR2	DMAC_BA+0x08	R/W	DMAC Transfer Starting Address Register 2	0x0000_0000
DMACBCR	DMAC_BA+0x0C	R	DMAC Transfer Byte Count Register	0x0000_0000
DMACIER	DMAC_BA+0x10	R/W	DMAC Interrupt Enable Register	0x0000_0001
DMACISR	DMAC_BA+0x14	R/W	DMAC Interrupt Status Register	0x0000_0000

4.12.5 DMAC Register Detail

DMAC Control and Status Register (DMACCSR)

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

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**FMI_BUSY**	**Reserved**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**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.
[8:2]	**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]	**DMACEN**	**DMAC Engine Enable** Setting this bit to 1 enables DMAC's operation. If this bit is cleared, DMAC will ignore all DMA request from ATAPI or FMI and force Bus Master into IDLE state. • 0 = Disable DMAC. • 1 = Enable DMAC. **NOTE:** If target abort is occurred, DMACEN will be cleared.

DMAC Transfer Starting Address Register (DMACSAR2)

Register	Offset	R/W	Description	Reset Value
DMACSAR2	0x08	R/W	DMAC Transfer Starting Address Register 2	0x0000_0000

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

Bits	Descriptions	
[31:0]	**DMACSA**	**DMA Transfer Starting Address for FMI** This field indicates a 32-bit starting address of system memory (SRAM/SDRAM) for DMAC to retrieve or fill in data (for FMI engine).

		If DMAC is not in normal mode, this field will be interpreted as a starting address of Physical Address Descriptor (PAD) table.

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

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							TABORT_IE

Bits	Descriptions	
[31:1]	Reserved	Reserved
[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							TABORT_IF

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	TABORT_IF	**DMA Read/Write Target Abort Interrupt Flag** • 0 = No bus ERROR response received. • 1 = Bus ERROR response received. **NOTE:** This bit is read only, but can be cleared by writing '1' to it.

NOTE: When DMAC's bus master received ERROR response, it means that target abort is happened. DMAC will stop transfer and respond this event to software, ATAPI and FMI; then go to IDLE state. When target abort occurred or WEOT_IF is set, software must reset DMAC and IP, and then transfer those data again.

4.13 Video-In Processor

The main purpose of Video-In Processor is processing the raw data that is output from sensor. Video-In Processor provides YCbCr or RGB or Y-only format to System Memory for Frame Buffer and RGB format to VPOST for Line Buffer. Output to Frame Buffer is for further processing. Output to Line Buffer is for real-time displaying.

4.13.1 Overview and Features

- Direct connect to CCD or CMOS image sensor with CCIR601
- Input Data format
  - YCbCr422
  - RGB565
- Output data format
  - YCbCr422
  - RGB555
  - Y-only
- Image Cropping
- Image Resize
  - Horizontal Down-Scaling
  - Vertical Down-Scaling
  - Frame Down-Scaling
- Built in 64bytes FIFO

4.13.2 Video-In Processor Functional Description

4.13.2.1 Timing Controller

- Generation Sensor module system clock source.
- Control Sensor module system clock to make the data received from Sensor is synchronous with VPOST Line Buffer timing.

4.13.2.2 Register Bank

- A bridge that CPU control and observe the state of Video-In Processor.

4.13.2.3 Detector

- Detect the video stream frame start / end.
- Detect the video stream row start / end.
- Swap these video stream order by user defined sequence.

4.13.2.4 Cropping

- Detect the valid column cropping.
- Detect the valid row cropping.

4.13.2.5 Scale Down

- Scale down the video pixel stream using normal distribution pixel scale-down method.
- Convert the YCbCr color space to RGB color space.

4.13.2.6 Count

- Prepare the video valid data and clock for FIFO.
- Total column/row/bytes of video valid data will been calculated.

4.13.2.7 FIFO

- Build in 64bytes FIFO to store video valid data before writing to Frame Buffer.

4.13.2.8 Master

- Write video stream data to Frame Buffer.

4.13.3 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_BA = 0xFFF0_7000				
VPRECtl	VIN_BA+0x00	R/W	Video-In Processor Control Register	0x0000_0000
VPREPrm	VIN_BA+0x04	R/W	Video-In Processor Parameter Register	0x0000_0000
VPREInt	VIN_BA+0x08	R/W	Video-In Processor Interrupt Register	0x0000_0000
MemBaseAd	VIN_BA+0x0c	R/W	System Memory Base Address Register	0x0000_0000
CurMemAd	VIN_BA+0x10	R	Current System Memory Address Register	0x0000_0000
VdCnt	VIN_BA+0x14	R	Video Data Row/Column Count Register	0x0000_0000
VdTalCnt	VIN_BA+0x18	R	Video Data Total Count Register	0x0000_0000
CpWStAd	VIN_BA+0x1c	R/W	Cropping Window Starting Address Register	0x0000_0000
CpWSz	VIN_BA+0x20	R/W	Cropping Window Size Register	0x0000_0000
SclVHFct	VIN_BA+0x24	R/W	Scaling Vertical/Horizontal Factor Register	0x0000_0000
SclFrFct	VIN_BA+0x28	R/W	Scaling Frame Rate Factor Register	0x0000_0000
CompMemAddr	VIN_BA+0x40	R/W	Compare Memory Base Address Register	0xffff_ffff
VOffMemAddr	VIN_BA+0x44	R/W	Video-In interlaced Offset Memory Address	0x0000_0000
VFrameCtl	VIN_BA+0x48	R/W	Video-In Frame tuning Control Register	0x0000_0000
VOverlayCnt	VIN_BA+0x4c	R	Video-In FIFO Overlay Count Register	0x0000_0000

4.13.4 Video-In Processor Control Register Description

Video-In Processor Control Register

Register	Address	R/W	Description	Reset Value

VPRECtl	VIN_BA+0x00	R/W	Video-In Processor Control Register	0x0000_0000

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

Bits	Descriptions	
[31:17]	**Reserved**	**Reserved**
[16]	**VPRST**	**Video-In Processor Reset** 0 = Disable, normal operation 1 = Reset the Video-In Processor except Registers
[15:2]	**Reserved**	**Reserved**
[1]	**AHBRUN**	**AHB Master Run** 0 = Disable 1 = Enable
[0]	**TCRUN**	**Timing Controller Run** 0 = Disable 1 = Enable

Video-In Processor Parameter Register

Register	Address	R/W	Description	Reset Value
VPREPrm	VIN_BA+0x04	R/W	Video-In Processor Parameter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	Reserved	VIIFP	VIIFE	Reserved	VIISS	VIIFD	VINSFMT
23	22	21	20	19	28	17	16
Reserved				CCWAIT	VSP	HSP	PCLKP
15	14	13	12	11	10	9	8
Reserved				PDORD		OUTFMT	
7	6	5	4	3	2	1	0
Reserved			INFMT				OPMODE

Bits	Descriptions	
[31:30]	**Reserved**	**Reserved**
[29]	**VIIFP**	**Video Input Interlaced Field Polarity** 0 = non- inversed 1 = inversed
[28]	**VIIFE**	**Video Input Interlaced Field Enable** 0 = disable 1 = enable
[27]	**Reserved**	**Reserved**
[26]	**VIISS**	**Video Input Interlace Scale Set** 0 = Progressive 1 = Set this bit for catching the scaling data in progressive with the interlace signal input(but set the bit [24]=0); ex: VPREPrm[26:24]=1X0
[25]	**VIIFD**	**Video Input Interlace Field Data** 0 = The first interlace field data is Odd 1 = The first interlace field data is Even
[24]	**VINSFMT**	**Video Input Signal Format** 0 = Progressive 1 = Interlaced
[23:20]	**Reserved**	**Reserved**
[19]	**CCWAIT**	**Waiting for VPOST Signals** 0 = Generate Sensor SCLK individual. 1 = Reference to VPOST Signal to generate Sensor SCLK
[18]	**VSP**	**Sensor Vsync Polarity** 0 = Active Low 1 = Active High
[17]	**HSP**	**Sensor Hsync Polarity** 0 = Active High 1 = Active Low
[16]	**PCLKP**	**Sensor Pixel Clock Polarity** 0 = Input video data and signals are latched by rising edge of Pixel Clock 1 = Input video data and signals are latched by falling edge of Pixel Clock
[15:12]	**Reserved**	**Reserved**
[11:10]	**PDORD**	**Sensor Input Data Order** 00 = U0 Y0 V0 Y1 01 = Y0 U0 Y1 V0 10 = V0 Y0 U0 Y1 11 = Y0 V0 Y1 U0
[9:8]	**OUTFMT**	**Image Data Format Output to System Memory** 00 = YCbCr422 format 01 = only output Y 10 = RGB555 format 11 = Reserved
[7:5]	**Reserved**	**Reserved**
[4]	**INFMT**	**Sensor Input Data Format** 0 = YCbCr422

		1 = RGB565
[3:1]	**Reserved**	**Reserved**
[0]	**OPMODE**	**Buffer Interface Operation Mode** 0 = Writing Only 1 frame image data to memory. 1 = Continuous writing image data to memory.

Video-In Processor Interrupt Register

Register	Address	R/W	Description	Reset Value
VPREInt	VIN_BA+0x08	R/W	Video-In Processor Interrupt Register	0x0000_0000

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

Bits	Descriptions	
[31:21]	**Reserved**	**Reserved**
[20]	**FieldDistinct**	**Represents the distinction of the interlaced field**
[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
[15:4]	**Reserved**	**Reserved**
[3]	**ADDRMINT**	**Memory Address Match Interrupt Flag.** If read this bit shows 1 Memory Address Match Interrupt occurs. Write 0 to clear it.
[2]	**Reserved**	**Reserved**
1	**MEINT**	**System Memory Error Interrupt** If read this bit shows 1, Memory Error occurs. Write 0 to clear it.
0	**VINT**	**Video Frame End Interrupt** If read this bit shows 1, received a frame complete. Write 0 to clear it.

System Memory Base Address Register

Register	Address	R/W	Description	Reset Value
MemBaseAd	VIN_BA+0x0c	R/W	System Memory Base Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**BASEAD**	**System Memory Base Address**

Current System Memory Address Register

Register	Address	R/W	Description	Reset Value
CurMemAd	VIN_BA+0x10	R	Current System Memory Address Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**CURAD**	**Current Accessing System Memory Address**

Video Data Row/Column Count Register

Register	Address	R/W	Description	Reset Value
VdCnt	VIN_BA+0x14	R	Video Data Row/Column Count Register	0x0000_0000

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

Bits	Descriptions	
[31:26]	Reserved	Reserved
[25:16]	ROWCNT	Processed Video Data Row Count
[15:11]	Reserved	Reserved
[10:0]	COLCNT	Processed Video Data Column Count

Video Data Total Count Register

Register	Address	R/W	Description	Reset Value
VdTalCnt	VIN_BA+0x18	R	Video Data Total Count Register	0x0000_0000

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	TALCNT	Processed Video Data Total Count

Cropping Window Starting Address Register

Register	Address	R/W	Description	Reset Value
CpWStAd	VIN_BA+0x1c	R/W	Cropping Window Starting Address Register	0x0000_0000

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

Bits	Descriptions	
[31:28]	CPWVSA offset	**Offset address for Cropping Window Vertical Starting Address of the different interlaced field** PS: CPWVSAoffset[3]: sign bit; 1:minus 0:plus for CPWVSAoffset[2:0]
[27:26]	Reserved	Reserved
[25:16]	CPWVSA	Cropping Window Vertical Starting Address
[15:11]	Reserved	Reserved
[10:0]	CPWHSA	Cropping Window Horizontal Starting Address

Cropping Window Size Register

Register	Address	R/W	Description	Reset Value
CpWSz	VIN_BA+0x20	R/W	Cropping Window Size Register	0x0000_0000

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

Bits	Descriptions	
[31:27]	**Reserved**	**Reserved**
[26:16]	**CPWHG**	**Cropping Image Window Height**
[15:12]	**Reserved**	**Reserved**
[11:0]	**CPWWD**	**Cropping Image Window Width**

Scaling Vertical/Horizontal Factor Register

Register	Address	R/W	Description	Reset Value
SclVHFct	VIN_BA+0x24	R/W	Scaling Vertical/Horizontal Factor Register	0x0000_0000

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

Bits	Descriptions	
[31:24]	**SCLVFN**	**Scaling Vertical Factor N** Specifies the denominator part (N) of the vertical scaling factor.
[23:16]	**SCLVFM**	**Scaling Vertical Factor M** Specifies the denominator part (M) of the vertical scaling 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.*** PS: for VIISS=1,VPREPrm[26:24]=1X0; *QVGA-320x240 SCLHFN/SCLHFM=1/2, SCLVFN/SCLVFM=1/1.* *QQVGA-160x120 SCLHFN/SCLHFM=1/4, SCLVFN/SCLVFM=1/2.*
[15:8]	**SCLHFN**	**Scaling Horizontal Factor N** Specifies the denominator part (N) of the horizontal scaling factor.
[7:0]	**SCLHFM**	**Scaling Horizontal Factor M** Specifies the denominator part (M) of the horizontal scaling 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.***

Scaling Frame Rate Factor Register

Register	Address	R/W	Description	Reset Value

SclFRFct	VIN_BA+0x28	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		**SCLFRFN**					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**SCLFRFM**					

Bits	Descriptions	
[31:14]	**Reserved**	**Reserved**
[13:8]	**SCLFRFN**	**Scaling Frame Rate Factor N** Specifies the denominator part (N) of the frame rate scaling factor.
[7:6]	**Reserved**	**Reserved**
[5:0]	**SCLFRFM**	**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.***

Compare Memory Address Register

Register	Address	R/W	Description	Reset Value
CompMemAddr	VIN_BA+0x40	R/W	Compare Memory Base Address Register	0xffff_ffff

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

Bits	Descriptions

[31:0]	**CompMemAddr**	**Compare Memory Base Address**

Video-In Interlaced Offset Memory Address Register

Register	Address	R/W	Description	Reset Value
VOffMemAddr	VIN_BA+0x44	R/W	Video-In interlaced Offset Memory Address	0x0000_0000

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

Bits	Descriptions	
[31:0]	**VOffMemAddr**	**Video-In interlaced Offset Memory Address**

Video-In Frame tuning Control Register

Register	Address	R/W	Description	Reset Value
VFrameCtl	VIN_BA+0x48	R/W	Video-In Frame tuning Control Register	0x0000_0000

31	30	29	28	27	26	25	24
VFrameCtl [31:27] Reserved					**Hsync DegEn**	**Vsync DegEn**	**PCLK DegEn**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**Data DeferEn**	**Data DeferCnt**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**HsyncEnd DeferEn**	**HsyncEnd DeferCnt**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**VOverCnt Clr**	**VOverCnt En**	**VAdAlign**	**INCRFlex**

Bits	Descriptions

[31:27]	**Reserved**	**Reserved**
[26]	**HsyncDegEn**	**Hsync PCLK De-glitch function enable**
[25]	**VsyncDegEn**	**Vsync PCLK De-glitch function enable**
[24]	**PCLKDegEn**	**PCLK De-glitch function enable**
[20]	**DataDeferEn**	**Video-In data defer enable**
[19:16]	**DataDeferCnt**	**Video-In data defer count**
[12]	**HsyncEnd DeferEn**	**Hsync. end signal defer enable**
[11:8]	**HsyncEnd DeferCnt**	**Hsync. end signal defer count**
[7:4]	**Reserved**	**Reserved**
[3]	**VOverCntClr**	**Video-In FIFO overlay count register clear**
[2]	**VOverCntEn**	**Video-In FIFO overlay count register enable**
[1]	**VAdAlign**	**Video line address align enable (ps: also need to set VOffMemAddr)**
[0]	**INCRFlex**	**Flexible transmission enable (INCR4)**

Video-In FIFO Overlay Count Register

Register	Address	R/W	Description	Reset Value
VOverlayCnt	VIN_BA+0x4c	R	Video-In FIFO Overlay Count Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**VOverlayCnt**	**Video-In FIFO Overlay Count Register**

4.14 JPEG Codec

4.14.1 Overview

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.

4.14.2 Features

- 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 Capture and JPEG hardware on-the-fly access mode for encode
- Support JPEG and Playback hardware on-the-fly access mode for decode
- Support software input/output on-the-fly access mode for both encode and decode
- Support arbitrary width and height image encode and decode
- Support three programmable quantization-tables
- Support standard default Huffman-table and programmable Huffman-table for decode
- Support arbitrarily 1X~8X image up-scaling function for encode mode
- Support down-scaling function for encode and decode modes
- Support specified window decode mode
- Support quantization-table adjustment for bit-rate and quality control in encode mode
- Support rotate function in encode mode

4.14.3 JPEG Encode

4.14.3.1 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. 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.

4.14.3.2 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. 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. 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.

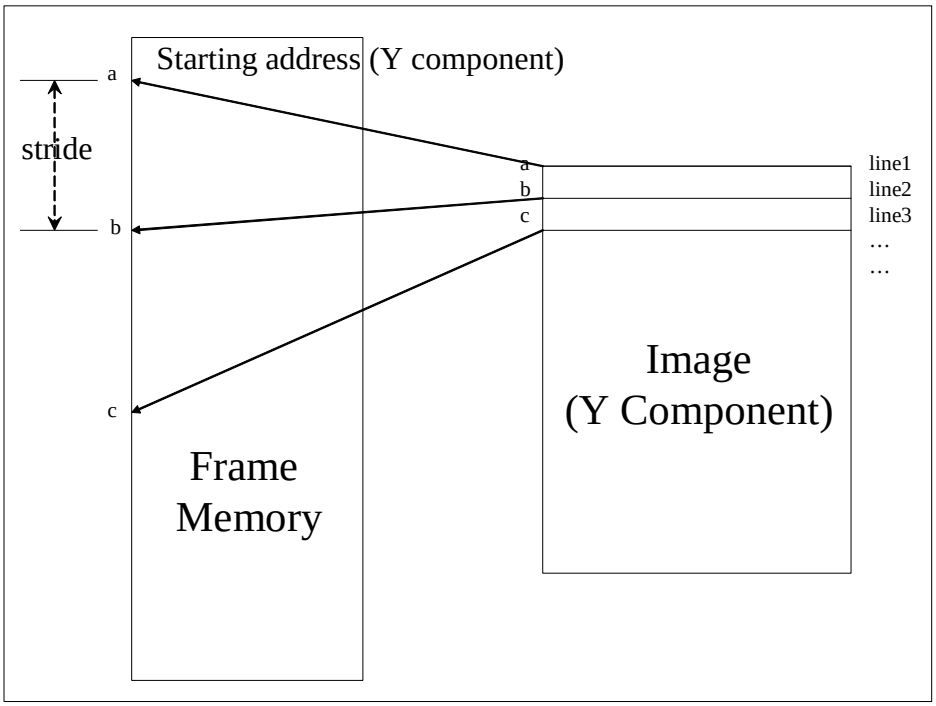


Figure 4.14.2 Image starting address and stride

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***.

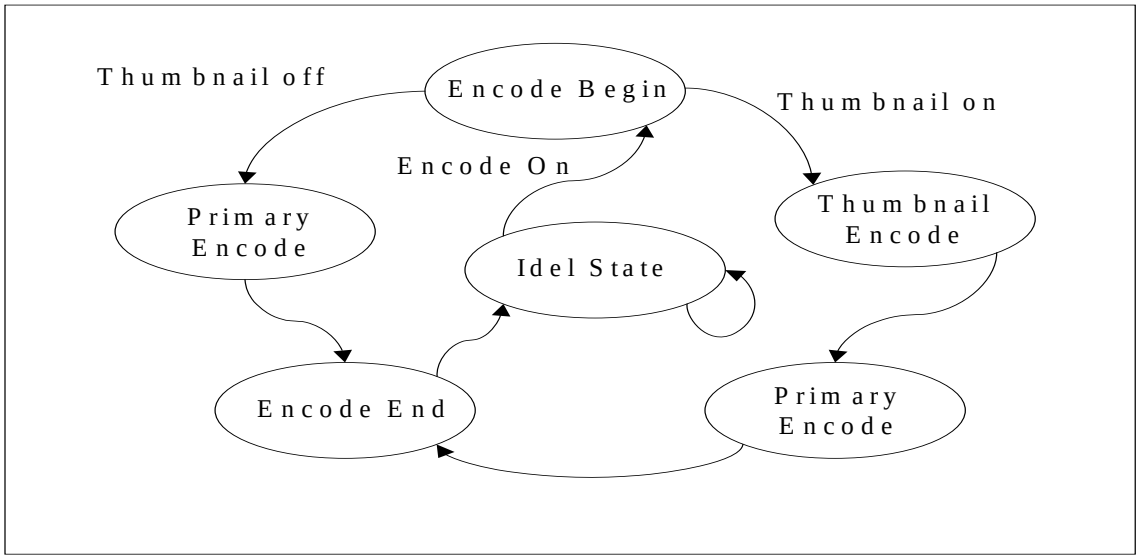


Figure 4.14.3 Primary and thumbnail encode

The JPEG Codec supports the 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 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 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 also supported. 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.

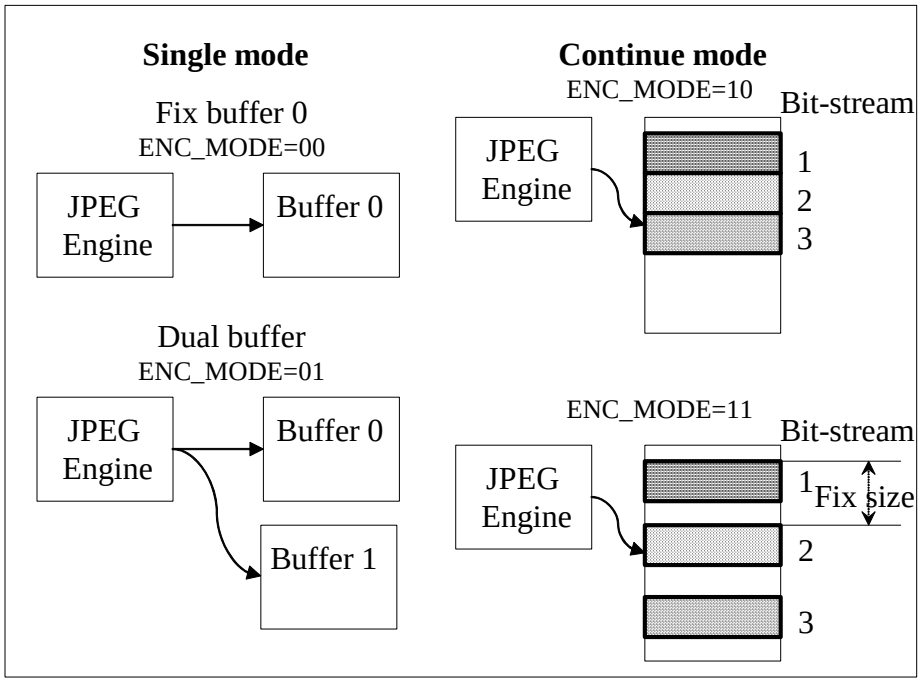


Figure 4.14.4 Single mode and continue mode

4.14.3.3 On-The-Fly Access Mode

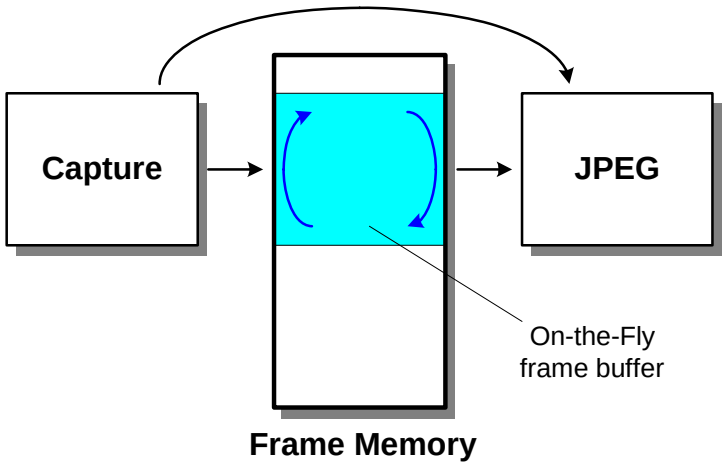


Figure 4.14.5 Hardware On-The-Fly encode mode

The JPEG codec supports hardware memory on-the-fly access in encode mode. This function can be enabled or disabled by setting the ***FLY_ON*** bit in register ***JMACR***. For normal memory access mode, JPEG engine can start the encode operation until the full frame picture has been stored into frame memory by Capture engine. For memory on-the-fly access mode, JPEG engine can start the encode operation while Capture engine is storing the current frame picture into frame memory. The buffer region in frame memory is accessed in a rotational manner. Therefore the smaller frame buffer is needed for this kind of operation. The on-the-fly frame buffer size is specified by register ***FLY_SEL***.

The JPEG codec also supports software memory on-the-fly access for both source picture and encoded bitstream. This input on-the-fly function can be enabled or disabled by setting the ***IP_SF_ON*** bit and the output on-the-fly function can be enabled or disabled by setting the ***OP_SF_ON*** bit. For normal memory access mode, the complete JPEG bitstream will be fully stored into frame memory during encode operation. For output on-the-fly access mode, a smaller buffer size can be specified for storing the encoded bitstream. The given buffer region in frame memory is accessed in a rotational and dual-buffer manner. After one of the half buffer regions is filled by JPEG, host needs to remove the bitstream data from this buffer. At the meanwhile, JPEG can store bitstream data into the other half buffer region. If the two buffers are full, JPEG will enter the pending state and wait for a RESUME by host to restart operation. The on-the-fly bitstream buffer size is specified by register ***BSF_SEL***.

For software input on-the-fly access mode, the operation is very like that of Capture and JPEG hardware on-the-fly mode except the YUV image data is filled by Host. In addition, the frame buffer is also accessed in a rotational and dual-buffer manner and the buffer size is fixed to 16-line for 4:2:2 and 32-line for 4:2:0 images. After one of the half buffer regions is filled by Host, a RESUME is needed to apply for JPEG to restart operation from a pending state if the two buffers are empty. The hardware on-the-fly and software output on-the-fly access mode can cooperation at the same time.

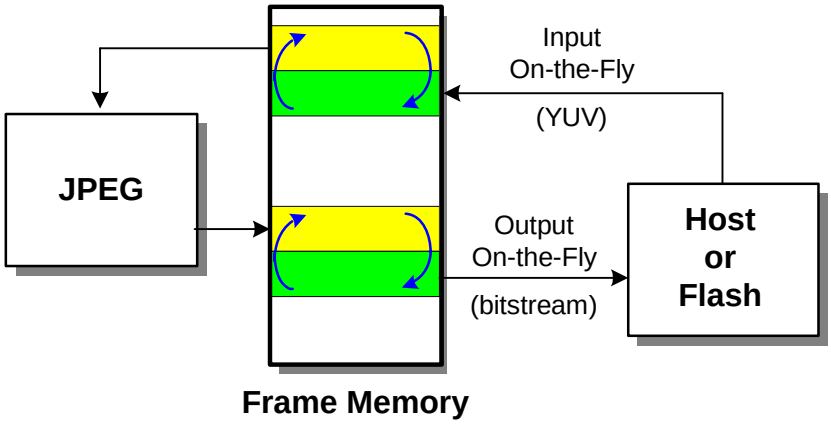


Figure 4.14.6 Software On-The-Fly encode mode (***FLY_TYPE***=2'b01)

In addition, two types of on-the-fly memory access mode are provided and can be selected by register ***FLY_TYPE***. The first type is dual-buffer manner on-the-fly (***FLY_TYPE***=2'b01) likes described above. The minimum frame buffer for encoding 4:2:2 images is 16-line for Y, Cb & Cr individually, and is 32-line for Y and 16-line for Cb & Cr individually for encoding 4:2:0 images. For 16 lines frame buffer size, after the first 8 lines have been stored into frame-memory by Capture engine, JPEG engine can start to fetch these 8 lines for encoding. At the meantime, the next 8 lines are continuing stored into frame-memory by Capture engine. After the second 8 lines have been stored, JPEG can start to fetch these 8 lines for encoding. At the meantime, the third 8 lines are atored into frame-memory from the initial starting address again. Obviously the processing speed of JPEG must faster than Capture in order to guarantee that the on-the-fly frame buffer won't be overwritten. Otherwise, the on-the-fly buffer size can be specified larger for preventing the overflow condition if the memory bandwidth is much critical.

The second one is single-buffer manner with fix addressing (***FLY_TYPE***=2'b10). In single-buffer mode, only half buffer is needed.

The single-buffer manner with fix addressing (***FLY_TYPE***=2'b10) mode can only be used with software input on-the-fly. The operation is almost the same as described above except that only a single

frame-buffer is used. The dual-buffer mode (***FLY_TYPE***=2'b01) can be used when software input on-the-fly or hardware input on-the-fly.

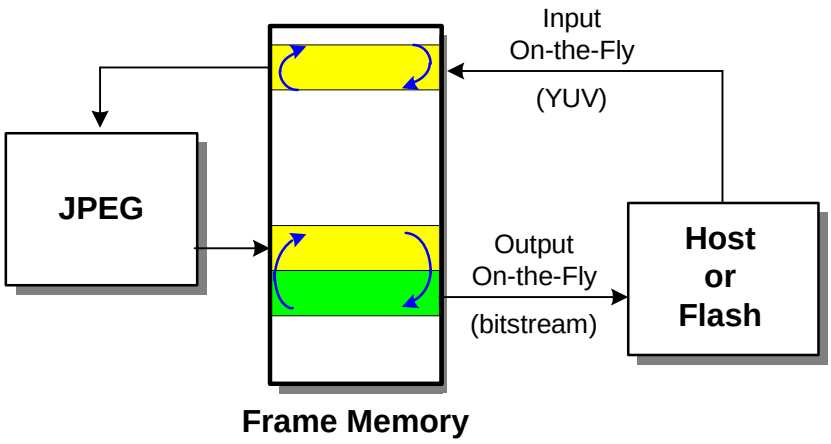


Figure 7.9.7 Software On-The-Fly encode mode (***FLY_TYPE***=2'b10)

4.14.3.4 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 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 Capture-JPEG hardware on-the-fly mode, the programming difference with normal mode is only need to additionally set ***FLY_ON*** and ***FLY_SEL*** and the same parameters specified in control registers of Capture engine. After JPEG on-the-fly encoding is triggered, JPEG engine will internally detect the next frame-start signal sent from Capture engine before starting the encode operation. However, the thumbnail encode function is not supported in on-the-fly encode mode.

For software output on-the-fly mode, the programming difference with normal mode is only need to additionally set ***OP_SF_ON*** and ***BSF_SEL*** and then follow the below steps:

(1) Trigger JPEG.

(2) Wait for the JPEG output-wait interrupt occurs and then start to remove the bitstream data.

(3) After one of the half bitstream buffers is removed, set ***RESUMEO*** bit to continue JPEG operation.

(4) Repeat step (2) & (3).

When JPEG is pending (bitstream buffer is full), the status bit ***JPG_WAITO*** will become high.

For software input on-the-fly mode, the programming difference with normal mode is only need to additionally set ***IP_SF_ON*** and then follow the below steps:

(1) Trigger JPEG after one of the YUV buffers (8 lines Y & Cb & Cr for 4:2:2; 16 lines Y, 8 lines Cb & Cr for 4:2:0) is filled.

(2) Fill next one of the YUV buffers and then set ***RESUMEI*** bit to continue JPEG operation if dual-buffer manner on-the-fly is applied.

(3) Wait for the JPEG input-wait interrupt occurs and then start to fill the next YUV buffer, set ***RESUMEI*** bit to continue JPEG operation.

(4) Repeat step (3).

When JPEG is pending (YUV buffer is empty), the status bit ***JPG_WAITI*** will become high.

4.14.4 JPEG Decode

4.14.4.1 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.

4.14.4.2 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 programmer can also get the quantization-table of the decoded JPEG bit-stream by reading the registers ***JQTAB0***~***JQTAB2***.

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 down-scaling in horizontal and vertical direction to adjust the decoded image size. The programmer can specify the register ***JPSCALD*** for image down-scaling function.

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***.

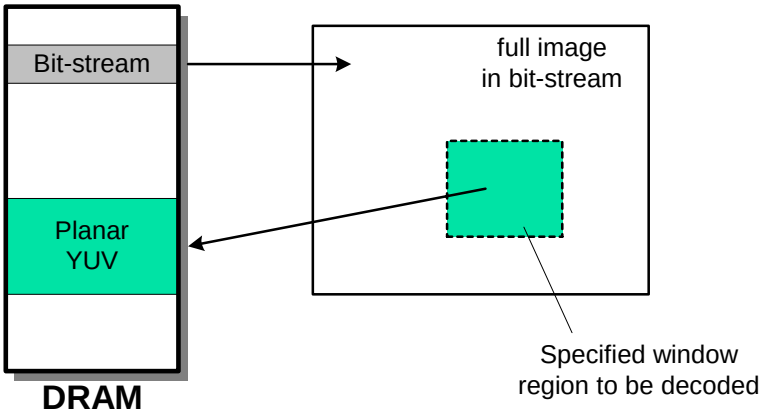


Figure 7.9.8 Specified window decode mode

4.14.4.3 On-The-Fly Access Mode

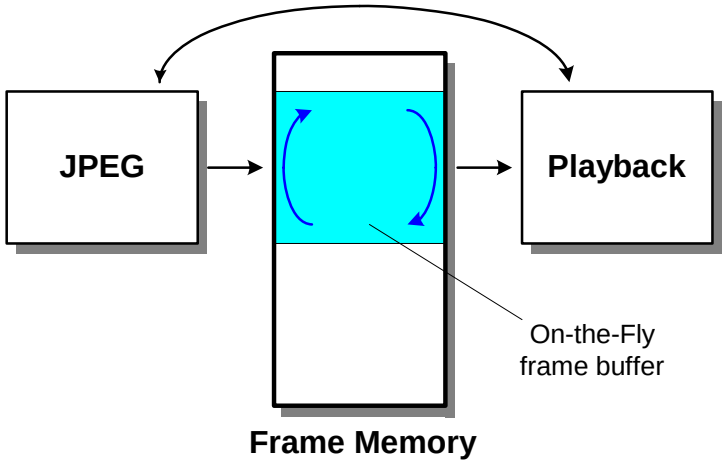


Figure 7.9.9 Hardware On-The-Fly decode mode

The JPEG codec supports hardware memory on-the-fly access in decode mode. This function can be enabled or disabled by setting the ***FLY_ON*** bit in register ***JMACR***. For normal memory access mode, JPEG engine will continue to decode and store the full frame picture into frame memory. And the Playback engine needs to wait for a whole frame picture is decoded before starting to convert the data format for display. For memory on-the-fly access mode, Playback engine can start the data format converting operation while JPEG engine is decoding and storing the current frame picture into frame memory. The buffer region in frame memory is accessed in a rotational manner. Therefore a smaller frame buffer is needed for this kind of operation. The on-the-fly frame buffer size is fixed to 16-line for Y, Cb & Cr of 4:4:4/4:2:2/4:1:1 images, and 32-line for Y and 16-line for Cb & Cr of 4:2:0 images in dual-buffer mode (***FLY_TYPE***=2'b01). And only half buffer size is needed in single-buffer mode (***FLY_TYPE***=2'b10).

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.

The JPEG codec also supports software memory on-the-fly access for both bitstream data and decoded images. The input on-the-fly function can be enabled or disabled by setting the ***IP_SF_ON*** bit and the output on-the-fly function can be enabled or disabled by setting the ***OP_SF_ON*** bit like that of encode mode. The operation is also very similar to that of encode mode except the input on-the-fly is used for the bitstream data to be decoded and the output on-the-fly is used for the decoded image data.

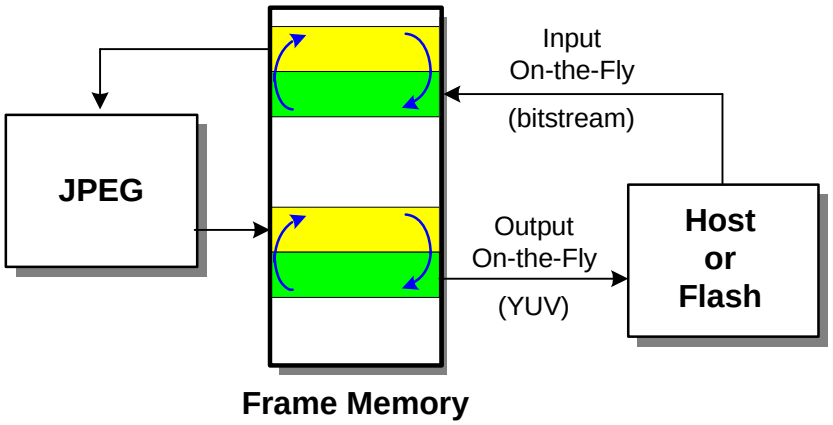


Figure 4.14.10 Software On-The-Fly decode mode (***FLY_TYPE***=2’b01)

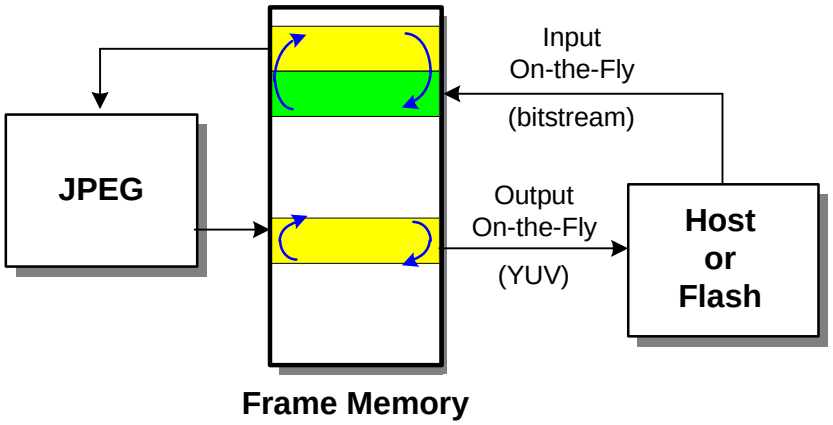


Figure 4.14.11 Software On-The-Fly decode mode (***FLY_TYPE***=2’b10)

4.14.5 JPEG Codec Interrupt

The JPEG codec supports encode complete, decode complete, encode error, decode error, input wait, output wait, 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 JPEG is operated with hardware on-the-fly encode mode and the condition of frame buffer overflow occurs, an encode error interrupt will be

issued to host. When JPEG is operated with software output on-the-fly mode and one of the YUV or bitstream buffers is filled, an output wait interrupt will be issued to host. When JPEG is operated with software input on-the-fly mode and one of the YUV or bitstream buffers is fetched, an input wait 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***.

4.14.6 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
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 Encode Primary Width/Height Register	0x0000_0000
JTHBWH	JPG_BA + 01C	R/W	JPEG Encode Thumbnail Width/Height Register	0x0000_0000
JPRST	JPG_BA + 020	R/W	JPEG Encode Primary Restart Interval Register	0x0000_0004
JTRST	JPG_BA + 024	R/W	JPEG Encode Thumbnail Restart Interval Register	0x0000_0004
JDECWH	JPG_BA + 028	R	JPEG Decode Image Width/Height Register	0x0000_0000
JINTCR	JPG_BA + 02C	R/W	JPEG Interrupt Control and Status Register	0x0020_0000
RESERVED	JPG_BA + 030 ~ JPG_BA + 03C	R/W	Reserved	0x0000_0000
JTEST	JPG_BA + 040	R/W	JPEG Test Control Register	0x0000_0000
JWINDEC0	JPG_BA + 044	R/W	JPEG Window Decode Mode Control Register 0	0x0000_0000
JWINDEC1	JPG_BA + 048	R/W	JPEG Window Decode Mode Control Register 1	0x0000_0000
JWINDEC2	JPG_BA + 04C	R/W	JPEG Window Decode Mode Control Register 2	0x0000_0000
JMACR	JPG_BA + 050	R/W	JPEG Memory Address Mode Control Register	0x0000_0000
JPSCALU	JPG_BA + 054	R/W	JPEG Primary Scaling-Up Control Register	0x0000_0000
JPSCALD	JPG_BA + 058	R/W	JPEG Primary Scaling-Down Control Register	0x0000_0000
JTSCALD	JPG_BA + 05C	R/W	JPEG Thumbnail Scaling-Down Control Register	0x0000_0000
JDBCR	JPG_BA + 060	R/W	JPEG Dual-Buffer Control Register	0x0000_0000
RESERVED	JPG_BA + 064 ~ JPG_BA + 06C	R/W	Reserved	0x0000_0000

JRESERVE	JPG_BA + 070	R/W	Primary Encode Bit-stream Reserved Size Register	0x0000_0000
JOFFSET	JPG_BA + 074	R/W	Address Offset Between Primary/Thumbnail Register	0x0000_0000
JFSTRIDE	JPG_BA + 078	R/W	JPEG Encode Bit-stream Frame Stride Register	0x0000_0000
JYADDR0	JPG_BA + 07C	R/W	Y Component 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 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 Encode Up-Scale Ratio	0x0000_0000
JBSFIFO	JPG_BA + 0B4	R/W	JPEG Bit-stream FIFO Control Register	0x0000_0032
JSRCH	JPG_BA + 0B8	R/W	JPEG Encode Source Image Height	0x0000_0FFF
RESERVED	JPG_BA + 0BC ~ JPG_BA + 0FC	R/W	Reserved	0x0000_0000
JQTAB0	JPG_BA + 100 ~ JPG_BA + 13F	R/W	JPEG Quantization-Table 0	0x0000_0000
JQTAB1	JPG_BA + 140 ~ JPG_BA + 17F	R/W	JPEG Quantization-Table 1	0x0000_0000
JQTAB2	JPG_BA + 180 ~ JPG_BA + 1BF	R/W	JPEG Quantization-Table 2	0x0000_0000
RESERVED	JPG_BA + 1C0 ~ JPG_BA + 1FC	R/W	Reserved	0x0000_0000

4.14.7 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							
15	14	13	12	11	10	9	8
PACKET_ON	**ORDER**	**Reserved**	**ROTATE**		**DYUV_MODE**		
7	6	5	4	3	2	1	0
EXIF	**EY_ONLY**	**DHEND**	**DTHB**	**E3QTAB**	**D3QTAB**	**ERR_DIS**	**PDHTAB**

Bits	Descriptions	
[31:16]	**Reserved**	**Reserved**
[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 = RGB555**
[13]	**Reserved**	
[12:11]	**ROTATE**	**Encode Image Rotate** • 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 Encode Primary Image Width/Height Register (JPRIWH)

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

31	30	29	28	27	26	25	24
Reserved				**P_HEIGHT[11:8]**			
23	22	21	20	19	18	17	16
P_HEIGHT[7:0]							

15	14	13	12	11	10	9	8
Reserved				**P_WIDTH[11:8]**			
7	6	5	4	3	2	1	0
P_WIDTH[7:0]							

Bits	Descriptions	
[31:28]	**Reserved**	**Reserved**
[27:16]	**P_HEIGHT**	**Primary Encode and Decode Image Height** A 12-bit value specifies the height of encoded 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 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 decode, the size is ceil((YSF/1024)*(height of image))
[15:12]	**Reserved**	**Reserved**
[11:0]	**P_WIDTH**	**Primary Encode and Decode Image Width** A 12-bit value specifies the width of encoded JPEG image. The value is equal to the size after scaling-up or scaling-down. When the down-scaling mode is enable in 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)

Register	Address	R/W	Description	Default Value
JTHBWH	JPG_BA + 01C	R/W	JPEG Encode Thumbnail Width/Height Register	0x0000_0000

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

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

JPEG Encode Primary Restart Interval Value Register (JPRST)

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

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

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

JPEG Encode Thumbnail Restart Interval Value Register (JTRST)

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

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

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

JPEG Decode Image Width/Height Register (JDECWH)

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

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

Bits	Descriptions	
[31:16]	**DEC_HEIGHT**	**Decode Image Height** A 16-bit value reports the height of decoded JPEG image.

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

JPEG Interrupt Control and Status Register (JINTCR)

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

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

Bits	Descriptions	
[31:24]	**Reserved**	**Reserved**
[23]	**JPG_WAITI**	**JPEG Input Wait Status (Read-Only)** • 0 = JPEG is operating or idle • 1 = JPEG is pending and is waiting for a input resume
[22]	**JPG_WAITO**	**JPEG Output Wait Status (Read-Only)** • 0 = JPEG is operating or idle • 1 = JPEG is pending and is waiting for a output resume
[21:17]	**Reserved**	**Reserved**

[16]	**BAbort**	**JPEG Memory Access Error Status (Read-Only)** • 0 = Normal operation • 1 = Wrong frame memory space is accessed
[15]	**CER_INTE**	**Un-complete Capture On-The-Fly Frame Occur Interrupt Enable** • 0 = Disable • 1 = 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**	**4.14.7.1 Decode Complete Interrupt Enable** • 0 = Disable • 1 = Enable
[9]	**DER_INTE**	**Decode Error Interrupt Enable** • 0 = Disable • 1 = Enable
[8]	**EER_INTE**	**Encode (On-The-Fly) Error Interrupt Enable** • 0 = Disable • 1 = Enable

[7]	**CER_INTS**	**Un-complete Capture On-The-Fly Frame Occur Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.
[6]	**DHE_INTS**	**JPEG Header Decode End Wait Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear and JPEG will resume operating from a pending state.
[5]	**IPW_INTS**	**Input Wait Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.
[4]	**OPW_INTS**	**Output Wait Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.
[3]	**ENC_INTS**	**Encode Complete Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.
[2]	**DEC_INTS**	**Decode Complete Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.
[1]	**DER_INTS**	**Decode Error Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.
[0]	**EER_INTS**	**Encode (On-The-Fly) Error Interrupt Status** • 0 = No Interrupt • 1 = Interrupt Generated Note: When write value “1” to this bit, the interrupt will be clear.

JPEG TEST Control Register (JTEST)

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

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

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

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

JPEG Window Decode Mode Control Register 0 (JWINDEC0)

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

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

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

[8:0]	**MCU_S_X**	**MCU Start Position X For Window Decode Mode** A 9-bit value specifies the MCU start position x of the window region within the whole image to be decoded when the window decode mode (**WIN_DEC**) is enabled. The position x is started from 0.

JPEG Window Decode Mode Control Register 1 (JWINDEC1)

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

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

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

JPEG Window Decode Mode Control Register 2 (JWINDEC2)

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

31	30	29	28	27	26	25	24
Reserved							

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

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

JPEG Memory Address Mode Control Register (JMACR)

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

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

Bits	Descriptions	
[31:30]	**Reserved**	**Reserved**

[29:24]	**FLY_SEL**	**Hardware Memory On-the-Fly Access Image Buffer-Size Selection for Encode** The numbers of lines used as frame-buffer = (FLY_SEL+1) * 16. Note: 1. This setting is only valid when **FLY_TYPE** = 2'b01. Otherwise the buffer-size is always fixed. 2. 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). 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	
[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.
[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	
[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
[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 encode mode, SCALX_F can be any value from 0 to 31. For decode mode, these register reserved in Project VA91.
[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 encode mode, SCALY_F can be any value from 0 to 63. For decode mode, these register reserved in Project VA91.

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	
[15]	**TSX_ON**	**Thumbnail Image Horizontal Down-Scaling For Encode** • 0 = Disable, no horizontal down-scale • 1 = Enable
[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.

[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	
[7]	**DBF_EN**	**Dual Buffering Control** • 0 = Disable dual buffering • 1 = Enable dual buffering
[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.
[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 Frame Buffer-0 Starting Address Register (JYADDR0)

Register	Address	R/W	Description	Default Value
JYADDR0	JPG_BA + 07C	R/W	JPEG Y Component 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 (*byte address of word aligned)*.

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 Frame Buffer-1 Starting Address Register (JYADDR1)

Register	Address	R/W	Description	Default Value
JYADDR1	JPG_BA + 088	R/W	JPEG Y Component 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 Frame Buffer-1 Starting Address** A 32-bit value specifies the starting address bits 31 to 0 of frame buffer-1 for Y component (*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)*.

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 Encode Up-Scale and Decode Down-Scale Ratio (JUPRAT)

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

31	30	29	28	27	26	25	24
Reserved		S_HEIGHT[11:8]					
23	22	21	20	19	18	17	16
S_HEIGHT[7:0]							
15	14	13	12	11	10	9	8
Reserved		S_WIDTH[11:8]					
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 Up-Scale or Down-Scale Ratio** • A 14-bit value specifies image height up-scale or down-scale ratio. The first 4 bits are integer part and the others are decimal part. The JPEG engine supports vertical arbitrarily up-scaling in encode mode and down-scaling in encode and 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(128/256).
[15:14]	**Reserved**	**Reserved**
[13:0]	**S_WIDTH**	• **JPEG Image Width Up-Scale or Down-Scale Ratio** • A 14-bit value specifies source image width up-scale or down-scale ratio. The first 4 bits are integer part and the others are decimal part. The JPEG engine supports horizontal arbitrarily up-scaling in encode mode and down-scaling in encode and 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(128/256).

JPEG Bit-stream FIFO Control Register (JBSFIFO)

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

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

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

JPEG Encode Source Image Height (JSRCH)

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

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8

Reserved				JSRCH[11:8]			
7	6	5	4	3	2	1	0
JSRCH[7:0]							

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

JPEG Quantization-Table 0 Register (JQTAB0)

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

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

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

[15:8]	**QTAB0_1**	**JPEG Quantization-Table 0 – 1** An 8-bit value specifies one element (1, 5, 9, …, 57, 61) of the Quantization-Table 0.
[7:0]	**QTAB0_0**	**JPEG Quantization-Table 0 – 0** An 8-bit value specifies one element (0, 4, 8, …, 56, 60) of the Quantization-Table 0.

JPEG Quantization-Table 1 Register (JQTAB1)

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

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

Bits	Descriptions	
[31:24]	**QTAB1_3**	**JPEG Quantization-Table 1 – 3** An 8-bit value specifies one element of the Quantization-Table 1. Note: When three-QTAB mode (**E3QTAB**) is enabled, JPEG encoder uses this table for coding Cb component. Otherwise JPEG encoder uses this table for coding both Cb and Cr component. *The other requirements are the same as QTAB0 described above.*
[23:16]	**QTAB1_2**	**4.14.7.2 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**	**4.14.7.3 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**	**4.14.7.4 JPEG Quantization-Table 2 – 0** An 8-bit value specifies one element of the Quantization-Table 2.

4.15 Analog to Digital Converter

The 10-bit analog to digital converter (ADC) in this chip is a successive approximation type ADC with 8-channel inputs, 2 inputs of them are dedicated for audio recorder. It needs 50 cycles to convert one sample, the maximum input clock to ADC is 25MHz, so the maximum conversion rate is 300K/sec, and the operating voltage range is 3.3V +/- 10%. The power down mode is supported in the ADC.
The touch screen interfaces are supported in this chp, it contains 4-wire, 5-wire and 8-wire analog resistive touch screen. The four switches to bias XP, XM, YP, YM are embedded in this chip. The CPU could access the ADC control register by APB bus, and the ADC output an interrupt signal to AIC to represent the completion of conversion.
Beside the 10-bit ADC, a 8 levels voltage detector is included in this chip. The detector result is independent with power supply, and it could give the system a warning signal when battery voltage is lower than an absolute reference voltage.

4.15.1 Features

- Maximum conversion rate: 300K sample per second
- Power supply voltage: 3.3V
- Analog input voltage range: 0 – 3.3 volts
- Touch screen semi-auto/auto conversion modes supported
- Waiting for trigger mode supports
- Standby mode supports
- 8-level voltage detector

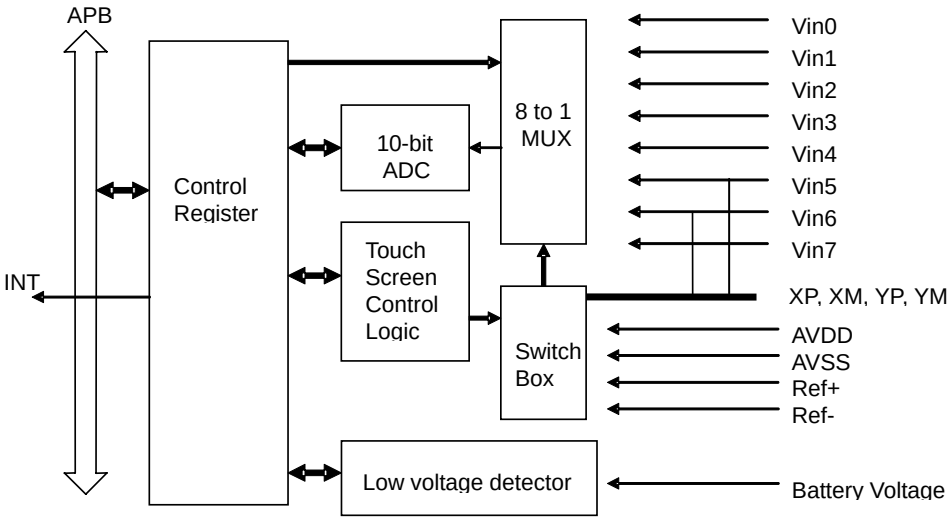


Figure 4.15-1 ADC Block Diagram

4.15.2 ADC Functional Description

4.15.2.1 Interface to Touch Screen

The touch screen control logic and the switch box could control the 4-wire, 5-wire and 8-wire type touch screen. The following figures show the interface for 4-wire, 5-wire and 8-wire touch screen respectively. Note that, the four switches to bias XP, XM, YP, YM have conduction resistance under 5 ohm. And the pull up PMOS have 200K ohm typically.

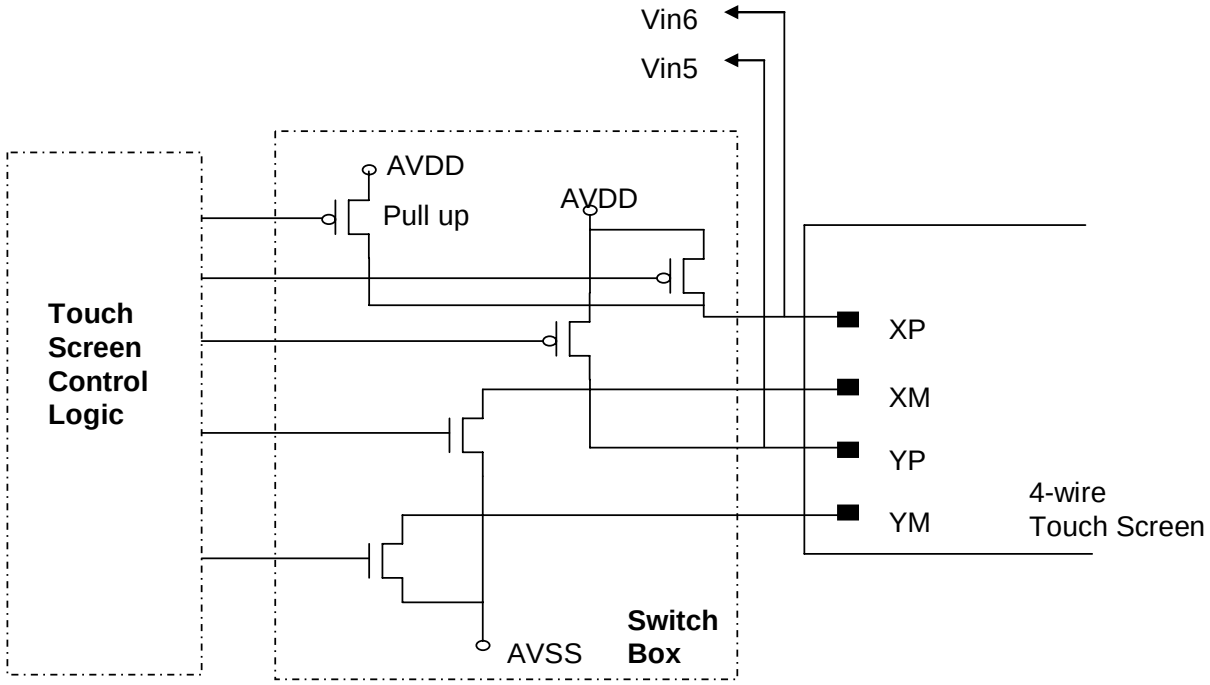


Figure 4.15-2 Interface for 4-wire Touch Screen

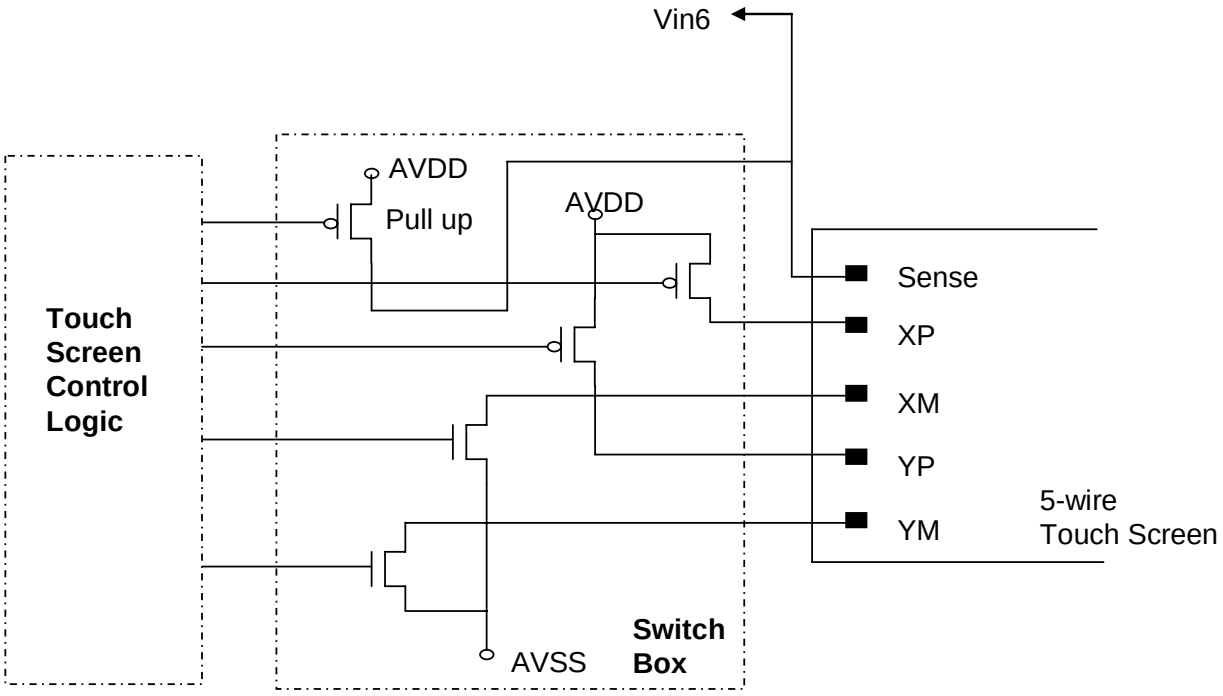


Figure 4.15-3 Interface for 5-wire Touch Screen

4.15.2.2 Waiting For Trigger Mode

This chip also supports the waiting for trigger mode in the touch screen control logic. In the 4-wire touch screen, when in waiting for trigger mode, the YM is connected to AVSS, XP is pulled high by a weak pull high PMOS, when the Stylus is down on the touch screen panel, Vin6 will receive a falling edge, then a active signal will be generated in INT output signal.
In the 5-wire touch screen, in the waiting for trigger mode, the YM is connected to AVSS, and the Sense will be pulled high. Vin6 also be in charge to detect the falling edge in this type of touch screen. The waiting for trigger mode for 8-wire touch screen is same as 4-wire.

4.15.2.3 Interface Mode

There are three control modes for ADC, normal conversion mode, semi-auto conversion mode, and auto conversion mode.
The normal conversion mode is for general purpose ADC, user could use the control register to control the 8 to 1 MUX to select analog input channel, start to conversion, wait interrupt or polling flag to confirm conversion finished, then to read the digital data. The conversion time is 50 ADC input clocks for each sample.
The semi-auto conversion mode and auto conversion mode are designed for touch screen. When the semi-auto conversion mode is proposed, the following procedures need to be followed.

a. Write the control register to start X-position detection. When starting detection, the proper switches will be enabled in switch box.

b. Waiting interrupt or polling status flag to confirm X-position is detected.
c. Write the control register to start Y-position detection. In the same way, the proper switches will be enabled in switch box.
d. Waiting interrupt or polling status flag to confirm Y-position id detected.
e. Read the X-position and Y-position in control registers.

In semi-auto conversion mode, for each position, the detection time is (2* 50 ADC clocks + program execution time between X-position and Y-position).

When the auto conversion mode is proposed, X-position and Y-position will be detected sequentially, the user could wait interrupt or polling status flag to confirm the detection finished, then to read the X-position and Y-position data in control register. In auto conversion mode, the detection time is 100 ADC clocks.

4.15.2.4 Standby mode

A standby mode is provided for ADC. When the ADC enable bit is cleared to 0, and the ADC clock is disable, the ADC enter standby mode. In the standby mode, the consumed current from AVDD will be less than 5uA. Note that the last conversion data is not cleared.

4.15.2.5 Voltage detector

The architecture of the voltage detector is shown as in the following figure.

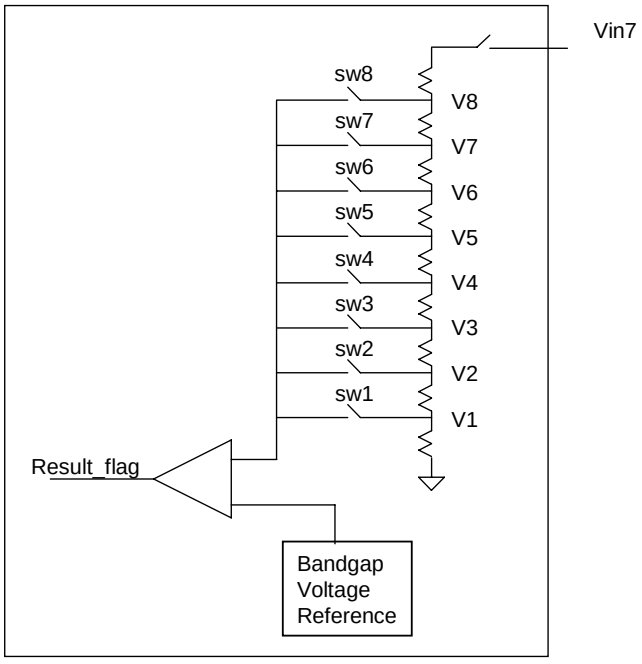


Figure 4.15-4 Architecture of Voltage Detector

By control the switch sw1, sw2, sw3, sw4, sw5, sw6, sw7 and sw8, to select the voltage V1, V2, V3, V4, V5, V6, V7 or V8 to be compared to reference voltage which will not be influenced by supply voltage or temperature.

4.15.2.6 Recording path

The audio recording path convert the audio analog to ditial data by means of the ADC hardware. When the

recording path is in usage, other data-conversion ADC function can't operate.

4.15.3 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 = 0xFFF8_5000				
ADC_CON	ADC_BA+0x000	R/W	ADC control register	0x0000_0001
ADC_TSC	ADC_BA+0x004	R/W	Touch screen control register	0x0000_0000
ADC_DLY	ADC_BA+0x008	R/W	ADC delay register	0x0000_0000
ADC_XDATA	ADC_BA+0x00C	R	ADC XDATA register	0x0000_0000
ADC_YDATA	ADC_BA+0x010	R	ADC YDATA register	0x0000_0000
LV_CON	ADC_BA+0x014	R/W	Low Voltage Detector Control register	0x0000_0000
LV_STS	ADC_BA+0x018	R/W	Low Voltage Detector Status register	0x0000_0000
AUDIO_CON	ADC_BA+0x01C	R/W	Audio control register	0x2000_0000

4.15.4 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_0001

31	30	29	28	27	26	25	24
RESERVED		**ADC_SYSCK_EN**	**ADC_SYS_DIV**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WT_INT_EN	**LVD_INT_EN**	**ADC_INT_EN**	**WT_INT**	**LVD_INT**	**ADC_INT**	**ADC_EN**	**ADC_RST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADC_TSC_MODE		**ADC_CONV**	**ADC_READ_CONV**	**ADC_MUX**			**ADC_DIV**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADC_DIV							**ADC_FINISH**

Bits	Descriptions	
[29]	**ADC_SYSCK_EN**	**Enable to System Clock Divider to ADC Input Clock** If ADC_SYSCK_EN = 0, the System Clock Divider doesn't work and ADC Input Clock doesn't toggle. ADC operation is halt. If ADC_SYSCK_EN = 1, the System clock Divider is enabled and ADC Input Clock toggles for ADC to provide operating clock.
[28:24]	**ADC_SYS_DIV**	**System Clock Divider to ADC Input Clock**

		ADC input clock is the system PLL clock divided by this 5 bit divider. The divider is enabled by the above bit 'ADC_SYSCK_EN'. The clock input frequency to this ADC control module is $Freq_{ADC\_INPUT\_CLK} = \frac{Freq_{PLL}}{(ADC\_SYS\_DIV + 1)}$ **NOTE1:** For the setting of PLL freuqency ($Freq_{PLL}$) please see the related description in system manager section. NOTE2: For the Recorder function, sampling rate are calculated as Record_Sampling_Rate = $\frac{Freq_{ADC\_INPUT\_CLK}}{1280 * 2}$
[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]	**LVD_INT_EN**	**Low voltage detector interrupt enable bit** If LVD_INT_EN=0, The LVD interrupt is disable If LVD_INT_EN=1, The LVD interrupt is enable The LV_INT_EN bit is read/write
[21]	**ADC_INT_EN**	**ADC interrupt enable bit** If ADC_INT_EN=0, The ADC interrupt is disable If ADC_INT_EN=1, The ADC interrupt is enable The ADC_INT bit is read/write
[20]	**WT_INT**	**Waiting for trigger interrupt status bit** If WT_INT=0, The waiting for trigger interrupt status is cleared If WT_INT=1, The waiting for trigger is in interrupt state The WT_INT bit is read/write and clear only, and set by hardware. Initial value after reset is unknow.
[19]	**LVD_INT**	**Low voltage detector (LVD) interrupt status bit** If LV_INT=0, The LVD interrupt status is cleared If LV_INT=1, The LVD is in interrupt state The LV_INT bit is read/write and clear only, and set by hardware. Initial value after reset is unknow.
[18]	**ADC_INT**	**ADC interrupt status bit** If ADC_INT=0, The ADC interrupt status is cleared If ADC_INT=1, The ADC is in interrupt state The ADC_INT bit is read/write and clear only, and set by hardware. Initial value after reset is unknow.
[17]	**ADC_EN**	**ADC block enable bit** If ADC_EN=0, The ADC block is disable If ADC_EN=1, The ADC block is enable The ADC_EN bit is read/write
[16]	**ADC_RST**	**ADC reset control bit** If ADC_RST=1, the ADC block is at reset mode If ADC_RST=0, the ADC block is at normal mode

		The ADC_RST bit is read/write
[15:14]	**ADC_TSC_MODE**	**The touch screen conversion mode control bits** If ADC_TSC_MODE=00, normal conversion mode is selected If ADC_TSC_MODE=01, semi-auto conversion mode is selected, and the ADC_CONV bit in ADC_CON register will be ignored If ADC_TSC_MODE=10, auto conversion mode is selected, and the ADC_CONV bit in ADC_CON register will be ignored If ADC_TSC_MODE=11, waiting for trigger mode is selected, and the ADC_CONV bit in ADC_CON register will be ignored **The ADC_TSC_MODE bits are read/write**
[13]	**ADC_CONV**	**ADC conversion control bit** If ADC_CONV=1, inform ADC to converse, when conversion finished, this bit will be auto clear. If ADC_CONV=0, the ADC no action, and this only could be cleared by hardware The ADC_CONV bit is read/write and could be set only
[12]	**ADC_READ_CONV**	This bit control if next conversion start after ADC_XDATA register is read in normal conversion mode. If ADC_READ_CONV=1, start next conversion after the ADC_XDATA is read, and ignore the ADC_CONV bit. If ADC_READ_CONV=0, after the ADC_XDATA is read, the ADC no action The ADC_READ_CONV bit is read/write
[11:9]	**ADC_MUX**	**These bits select ADC input from the 8 analog inputs in normal conversion mode.** ADC_MUX=000, not available in normal data conversion. 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 The ADC_MUX bits are read/write. **AIN0 and AIN1 channel are differential inputs for audio recorder only.** **When in Audio Recording operation (AUDIO_CON[1] / AUDIO_EN = 1), only AIN0 and AIN1 are dedicated inputs, and ADC_MUX value is not effective in this operation.**
[8:1]	**ADC_DIV**	The ADC input clock divider. The real ADC operating clock is the input clock divide ((round(ADC_DIV/2)+1)*2), except 1 and 0. When the ADC_DIV is set to 1 or 0, the ADC clock is equal to input clock. For example: When ADC_DIV = 0 or 1, ADC operating clock is ADC input clock. When ADC_DIV = 2 or 3, ADC operating clock is ADC input clock divided by 4. When ADC_DIV = 4 or 5, ADC operating clock is ADC input clock divided by 6;

		The real ADC operating clock, in term of $Freq_{PLL}$ can be calculated as below $Freq_{ADC\_REAL\_OPT} = \frac{Freq_{ADC\_INPUT\_CLK}}{\{[round(ADC\_DIV/2)+1]*2\}}$ $= \frac{Freq_{PLL}}{(ADC\_SYS\_DIV+1)\{[round(ADC\_DIV/2)+1)]*2\}}$ And the maximal ADC sample clock in automatic mode will be $Freq_{ADC\_Sample} = Freq_{ADC\_REAL\_OPT} / 50$
[0]	**ADC_FINISH**	This bit indicate the ADC is in conversion or not ADC_FINISH=0, the ADC is in conversion ADC_FINISH=1, the ADC is not in conversion The ADC_FINISH bit is read only.

Touch screen control register (ADC_TSC)

Register	Address	R/W	Description	Reset Value
ADC_TSC	ADC_BA+0x004	R/W	Touch screen control register	0x0000_0000

31	30	29	28	27	26	25	24
RESVERED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESVERED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESVERED							**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	
[8]	**ADC_TSC_XY**	**This bit control the X-position or Y-position detection when in semi-auto conversion mode** If ADC_TSC_XY = 0, X-position detection is select If ADC_TSC_XY = 1, Y-position detection is select The ADC_TSC_XY bit is read/write
[7]	**ADC_TSC_XP**	**This bit control the interface to XP of touch screen when in normal conversion mode** If ADC_TSC_XP = 0, XP is tri-state output If ADC_TSC_XP = 1, XP is connected to AVDD The ADC_TSC_XP bit is read/write
[6]	**ADC_TSC_XM**	**This bit control the interface to XM of touch screen when in normal conversion mode**

		If ADC_TSC_XM = 0, XM is tri-state output If ADC_TSC_XM = 1, XM is connected to AVSS The ADC_TSC_XM bit is read/write
[5]	**ADC_TSC_YP**	**This bit control the interface to YP of touch screen when in normal conversion mode** If ADC_TSC_YP = 0, YP is tri-state output If ADC_TSC_YP = 1, YP is connected to AVDD The ADC_TSC_YP bit is read/write
[4]	**ADC_TSC_YM**	**This bit control the interface to YM of touch screen when in normal conversion mode** If ADC_TSC_YM = 0, YM is tri-state output If ADC_TSC_YM = 1, YM is connected to AVSS The ADC_TSC_YM bit is read/write
[3]	**ADC_PU_EN**	**This bit control the internal pull up PMOS in switch box is enable or disable** If ADC_PU_EN = 0, the pull up PMOS is disable If ADC_PU_EN = 1, the pull up PMOS is enable The ADC_PU_EN bit is read/write
[2:1]	**ADC_TSC_TYPE [1:0]**	**The touch screen type selection bits** If ADC_TSC_TYPE[1:0]=00, 4-wire type is selected If ADC_TSC_TYPE[1:0]=01, 5-wire type is selected If ADC_TSC_TYPE[1:0]=10, 8-wire type is selected If ADC_TSC_TYPE[1:0]=11, unused **The ADC_TSC_TYPE[1:0] bits are read/write**
[0]	**ADC_UD**	**The up down state for stylus in waiting for trigger mode** If ADC_UD = 1, the stylus is in donw state If ADC_UD = 0, the stylus is in up state The ADC_UD bit is read only

ADC Delay Register (ADC_DLY)

Register	Address	R/W	Description	Reset Value
ADC_DLY	ADC_BA+0x008	R/W	ADC delay register	0x0000_0000

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

Bits	Descriptions

[17:0]	**ADC_DELAY**	**Delay for Conversion.** For normal conversion mode, the delay is between the ADC_CONV bit in ADC_CON register is set to the ADC begin conversion. For semi-auto conversion mode, the delay locates at each X-position and Y-position detection. For auto conversion mode, the delay locates at each position detection. The delay is defined as ADC_DELAY * ADC clock The ADC_DELAY[17:0] bits is read/write.

ADC X data buffer (ADC_XDATA)

Register	Address	R/W	Description	Reset Value
ADC_XDATA	ADC_BA+0x00C	R	ADC X data buffer	0x0000_0000

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

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**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

ADC_YDATA

Bits	Descriptions	
[9:0]	**ADC_YDATA**	**ADC Y Data Buffer.** When semi-auto conversion mode, the conversion data of Y-position detection is put at this register. When auto-conversion mode, the conversion data of Y-position detection is put at this register. The ADC_YDATA[9:0] bits is read only.

Low Voltage Detector Control Register (LV_CON)

Register	Address	R/W	Description	Reset Value
LV_CON	ADC_BA+0x014	R/W	Low voltage detector control register	0x0000_0000

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

Bits	Descriptions	
[3]	**LV_EN**	**Low voltage detector enable control pin** If LV_EN = 0, low voltage detector is disable If LV_EN = 1, low voltage detector is enable The LV_EN bit is read/write
[2:0]	**SW_CON**	**The low voltage detector voltage level switch control bits** If SW_CON = 000, SW1 is close, others are open. If SW_CON = 001, SW2 is close, others are open. If SW_CON = 010, SW3 is close, others are open. If SW_CON = 011, SW4 is close, others are open. If SW_CON = 100, SW5 is close, others are open. If SW_CON = 101, SW6 is close, others are open. If SW_CON = 110, SW7 is close, others are open. If SW_CON = 111, SW8 is close, others are open. The switches SW1 – SW8 are shown in Figure 4.15-4. The relationship of SW_CON setting vs. detected voltage level

SW_CON	Voltage Level (V)
000	2.0
001	2.1
010	2.2
011	2.3

		100	2.4
		101	2.5
		110	2.6
		111	2.7

NOTE: The voltage input of Low Voltage Detector is from ADC input AIN7, or named as GPA[7]. The switch control of SW_CON relations and the detector operation can be referred from Figure 4.15-4 Architecture of Voltage Detector.

Low Voltage Detector Status Register (LV_STS)

Register	Address	R/W	Description	Reset Value
LV_STS	ADC_BA+0x018	R/W	The status register of low voltage detector	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							**LV_status**

Bits	Descriptions	
[31:1]	**RESERVED**	Keep these bits as 0.
[0]	**LV_status**	**Low voltage detector status pin** • If LV_status = 0, the compared voltage is higher than reference voltage • If LV_status = 1, the compared voltage is lower than reference voltage The LV_status bit is read only

Audio control register (AUDIO_CON)

Register	Address	R/W	Description	Reset Value
AUDIO_CON	ADC_BA+0x01C	R/W	The status register of low voltage detector	0x2000_0000

31	30	29	28	27	26	25	24
RESERVED		**OP_OFFSET[3:0]**				**AUDIO_DATA[15:14]**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
AUDIO_DATA[13:6]							

15	14	13	12	11	10	9	8
AUDIO_DATA[5:0]						AUD_INT	VOL_EN
7	6	5	4	3	2	1	0
AUDIO_VOL[4:0]					AUDIO_HPEN	AUDIO_EN	AUDIO_RESET

Bits	Descriptions	
[29:26]	**OP_OFFSET[3:0]**	**Reserved for engineering used. Keep the setting value as default.**
[25:10]	**AUDIO_DATA[15:0]**	**Converted audio data** 16-bit digital audio data in 2's compliment format. These bits are read only and the initial values are unknown.
[9]	**AUD_INT**	**Audio interrupt flag bits** If AUD_INT = 0, the recording for one sample is not finished. If AUD_INT = 1, the recording for one sample is finished. The bit is readable and clear only. This flag can be set by above hardware event if AUDIO_EN = 1. And when it is set an interrupt signal is asserted to the interrupt controller through ADC interrupt source.
[8]	**VOL_EN**	**Volume control enable bit** If VOL_EN = 0, the hardware open the volume control path and open the recording path. If VOL_EN = 1, the hardware enable the volume control path and enable the recording path. The bit is R/W
[7:3]	**AUDIO_VOL[4:0]**	**Volume control bits** In FB5991, AUDIO_VOL[4:0] = [0, 1, 2, … , 31] indicate the volume from [-15dB, -14dB, …, 0dB, 1dB, … 16dB] In FC5991, AUDIO_VOL[4:0] = [0, 1, 2, … , 31] indicate the volume from [0dB, 1dB, …, 31dB] The bits are R/W
[2]	**AUDIO_HPEN**	**Record path high pass enable bit** If AUDIO_HPEN = 0, the digital high pass filter will be bypassed. If AUDIO_HPEN = 0, the digital high pass filter will be enable. The bit is R/W
[1]	**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
[0]	**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

4.16 General Purpose I/O

4.16.1 Overview and Features

56 pins of General Purpose I/O are shared with special feature functions. Among them, there are 8 pins of general purpose I/O especially shared with external memory data bus MD[24 ~ 31].
Supported Features of these I/O are: input or output facilities, pull-up resistors.
All these general purpose I/O functions are achieved by software programming setting and I/O cells selected from TSMC universal standard I/O Cell Library. And the following figures illustrate the control mechanism to achieve the GPIO functions.

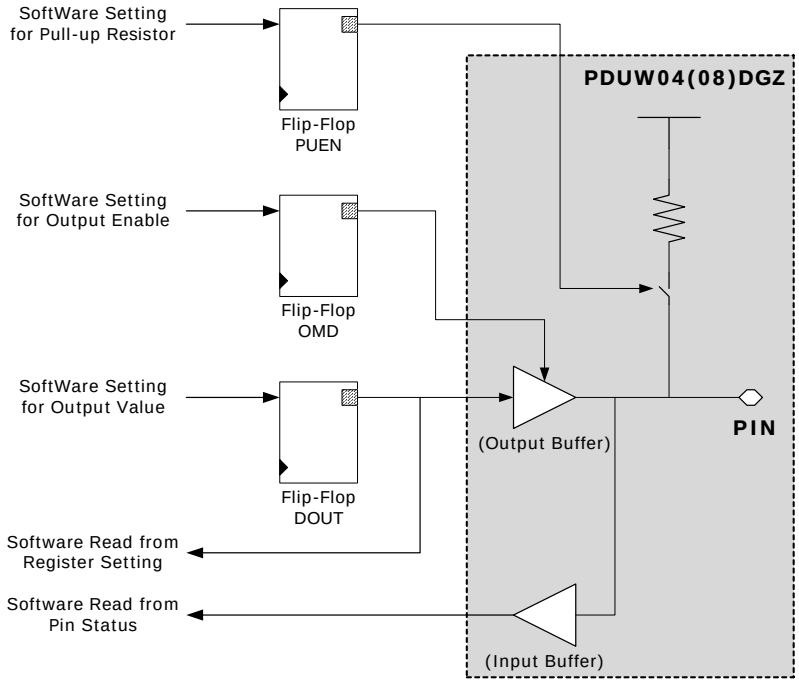


Figure 4.16-1 Type I GPIO: Input/Output Port with Program Controlled Weakly Pull-High

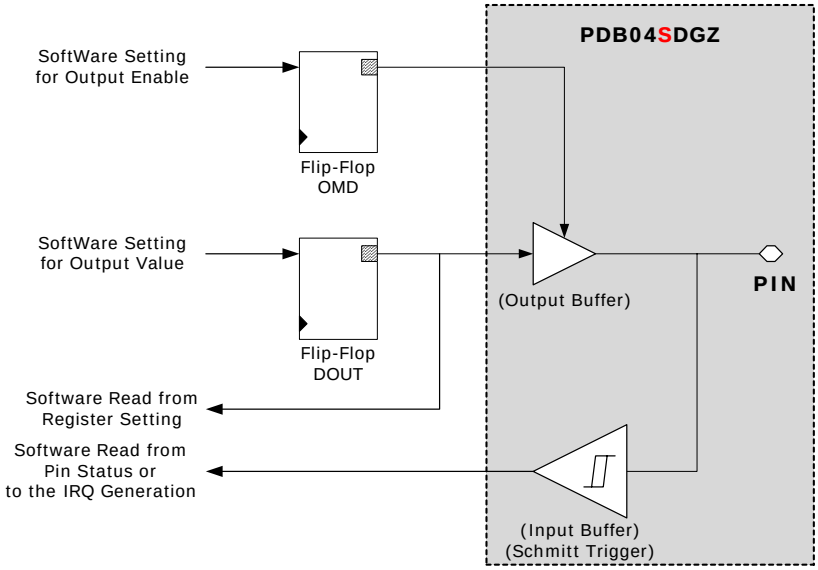


Figure 4.16-2 Type II GPIO: Input/Output Port with Schmitt-Trigger Input

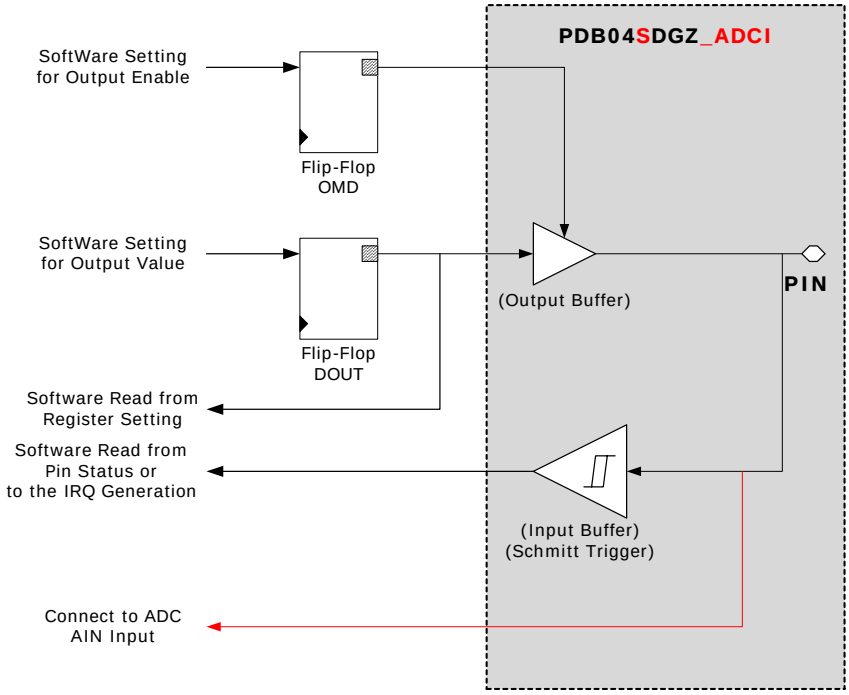


Figure 4.16-3 Type III GPIO: Input/Output Port with Schmitt-Trigger Input and Analog Input to Analog-to-Digital Converter

4.16.2 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 = 0xFFF8_4000				
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_0000
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_0000
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_0000
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_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_0000
GPIOE_DOUT	GP_BA+0x48	R/W	GPIO Port E Data Output Value	0x0000_0000
GPIOE_PIN	GP_BA+0x4C	R	GPIO Port E Pin Value	0xXXXX_XXXX
DBNCECON	GP_BA+0x70	R/W	External Interrupt De-bounce Control	0x0000_0000
IRQSRCGPA	GP_BA+0x80	R/W	GPIO Port A IRQ Source Grouping	0x0000_0000
IRQSRCGPB	GP_BA+0x84	R/W	GPIO Port B IRQ Source Grouping	0x5555_5555
IRQSRCGPC	GP_BA+0x88	R/W	GPIO Port C IRQ Source Grouping	0x0000_AAAA
IRQSRCGPE	GP_BA+0x8C	R/W	GPIO Port E IRQ Source Grouping	0xFFFF_FFFF
IRQENGPA	GP_BA+0x90	R/W	GPIO Port A Interrupt Enable	0x0000_0000
IRQENGPB	GP_BA+0x94	R/W	GPIO Port B Interrupt Enable	0x0000_0000
IRQENGPC	GP_BA+0x98	R/W	GPIO Port C Interrupt Enable	0x0000_0000
IRQENGPE	GP_BA+0x9C	R/W	GPIO Port E Interrupt Enable	0x0000_0000
IRQLHSEL	GP_BA+0xA0	R/W	Interrupt Latch Trigger Selection Register	0x0000_0000
IRQLHGPA	GP_BA+0xA4	R	GPIO Port A Interrupt Latch Value	0x0000_0000
IRQLHGPB	GP_BA+0xA8	R	GPIO Port B Interrupt Latch Value	0x0000_0000
IRQLHGPC	GP_BA+0xAC	R	GPIO Port C Interrupt Latch Value	0x0000_0000
IRQLHGPE	GP_BA+0xB0	R	GPIO Port E Interrupt Latch Value	0x0000_0000
IRQTGSRC0	GP_BA+0xB4	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port A and GPIO Port B	0x0000_0000
IRQTGSRC1	GP_BA+0xB8	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port C and GPIO Port E	0x0000_0000

4.16.3 GPIO Control Register Description

GPIO Port [A/B/E] Bit Output Mode Enable (GPIOA/B/E_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
GPIOE_OMD	GP_BA+0x40	R/W	GPIO Port E Bit Output Mode Enable	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
OMD15	OMD14	OMD13	OMD12	OMD11	OMD10	OMD9	OMD8

7	6	5	4	3	2	1	0
OMD7	OMD6	OMD5	OMD4	OMD3	OMD2	OMD1	OMD0

Bits	Descriptions		Default
[15:0]	**OMD15 ~ OMD0**	**Bit Output Mode Enable** 1 = GPIO port [A/B/E] bit [n] output mode is enabled, the bit value contained in the corresponding bit [n] of GPIO[A/B/E]_DOUT is driven on the pin. 0 = GPIO port [A/B/E] bit [n] output mode is disabled, the corresponding pin is in INPUT mode.	0x0000

GPIO Port [A/B/E] Bit Pull-up Resistor Enable (GPIOA/B/E_PUEN)

Register	Address	R/W	Description	Reset Value
GPIOA_PUEN	GP_BA+0x04	R/W	GPIO Port A Bit Pull-up Resistor Enable	0x0000_0000
GPIOB_PUEN	GP_BA+0x14	R/W	GPIO Port B Bit Pull-up Resistor Enable	0x0000_0000
GPIOE_PUEN	GP_BA+0x44	R/W	GPIO Port E Bit Pull-up Resistor 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**
PUEN15	PUEN14	PUEN13	PUEN12	PUEN11	PUEN10	PUEN9	PUEN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PUEN7	PUEN6	PUEN5	PUEN4	PUEN3	PUEN2	PUEN1	PUEN0

PUEN[n]: Bit Pull-up Resistor Enable
1 = GPIO port [A/B/E] bit [n] pull-up resistor is enabled.
0 = GPIO port [A/B/E] bit [n] pull-up resistor is disabled.

GPIO Port [A/B/E] Data Output Value (GPIOA/B/E_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
GPIOE_DOUT	GP_BA+0x48	R/W	GPIO Port E Data Output Value	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							

15	14	13	12	11	10	9	8
DOUT15	DOUT14	DOUT13	DOUT12	DOUT11	DOUT10	DOUT9	DOUT8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0

Bits	Descriptions		Default
[15:0]	**DOUT15 ~ DOUT0**	**Bit Output Value** 1 = GPIO port [A/B/E] bit [n] will drive High if the corresponding output mode enabling bit is set. 0 = GPIO port [A/B/E] bit [n] will drive Low if the corresponding output mode enabling bit is set.	0x0000

GPIO Port [A/B/E] Pin Value (GPIOA/B/E_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
GPIOE_PIN	GP_BA+0x4C	R	GPIO Port C 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/E] Pin Values**	0xXXXX

GPIO Port C Bit Output Mode Enable (GPIOC_OMD)

Register	Address	R/W	Description	Reset Value
GPIOC_OMD	GP_BA+0x20	R/W	GPIO Port C 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**
RESERVED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OMD7	OMD6	OMD5	OMD4	OMD3	OMD2	OMD1	OMD0

Bits	Descriptions		Default
[7:0]	**OMD7 ~ OMD0**	**Bit Output Mode Enable** 1 = GPIO port C bit [n] output mode is enabled, the bit value contained in the corresponding bit [n] of GPIOC_DOUT is driven on the pin. 0 = GPIO port C bit [n] output mode is disabled, the corresponding pin is in INPUT mode.	0x0000

GPIO Port C Bit Pull-up Resistor Enable (GPIOC_PUEN)

Register	Address	R/W	Description	Reset Value
GPIOC_PUEN	GP_BA+0x24	R/W	GPIO Port C Bit Pull-up Resistor 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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PUEN7	PUEN6	PUEN5	PUEN4	PUEN3	PUEN2	PUEN1	PUEN0

PUEN[n]: Bit Pull-up Resistor Enable
1 = GPIO port C bit [n] pull-up resistor is enabled.
0 = GPIO port C bit [n] pull-up resistor is disabled.

GPIO Port C Data Output Value (GPIOC_DOUT)

Register	Address	R/W	Description	Reset Value
GPIOC_DOUT	GP_BA+0x28	R/W	GPIO Port C 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**
RESERVED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0

Bits	Descriptions		Default

[7:0]	**DOUT7 ~ DOUT0**	**Bit Output Value** 1 = GPIO port C bit [n] will drive High if the corresponding output mode enabling bit is set. 0 = GPIO port C bit [n] will drive Low if the corresponding output mode enabling bit is set.	0x0000

GPIO Port C Pin Value (GPIOC_PIN)

Register	Address	R/W	Description	Reset Value
GPIOC_PIN	GP_BA+0x2C	R	GPIO Port C Pin Value	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**
PIN[7:0]							

Bits	Descriptions		Default
[7:0]	**PIN**	**Port C Pin Values**	0x000X

GPIO Port D Bit Output Mode Enable (GPIOD_OMD)

Register	Address	R/W	Description	Reset Value
GPIOD_OMD	GP_BA+0x30	R/W	GPIO Port D 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**
RESERVED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESERVED				OMD3	OMD2	OMD1	OMD0

Bits	Descriptions		Default
[3:0]	**OMD3 ~ OMD0**	**Bit Output Mode Enable** 1 = GPIO port D bit [n] output mode is enabled, the bit value contained in the corresponding bit [n] of GPIOD_DOUT is driven on the pin. 0 = GPIO port D bit [n] output mode is disabled, the corresponding pin is in INPUT mode.	0x0000

NOTE:

* GPIO Port D is a bit controllable 4-bit GPIO. This port is a multifunction with MA22, MA21, MA20, and MA19, corresponding to GPIOD[3], GPIOD[2], GPIOD[1] and GPIOD[0], respectively.
** After system power-on, the pins MA22, MA21, MA20 and MA19 are memory address outputs with "voltage low" by default. A programmer has to set "GPIOD_EN" register (PINFUN.bit16, address 0xFFF0_001C) to turn these four pins as GPIO at a time.
*** Once when "GPIOD_EN" is set to 1, all register setting values for GPIOD_OMD, GPIOD_DOUT, and GPIOD_PIN will be effective on MA22, MA21, MA20 and MA19 immediately. Otherwise, MA22, MA21, MA20 and MA19 will always operate as memory address outputs.

GPIO Port D Data Output Value (GPIOD_DOUT)

Register	Address	R/W	Description	Reset Value
GPIOD_DOUT	GP_BA+0x38	R/W	GPIO Port D 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
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				DOUT3	DOUT2	DOUT1	DOUT0

Bits	Descriptions		Default
[3:0]	**DOUT3 ~ DOUT0**	**Bit Output Value** 1 = GPIO port D bit [n] will drive High if the corresponding output mode enabling bit is set. 0 = GPIO port D bit [n] will drive Low if the corresponding output mode enabling bit is set.	0x0000

GPIO Port D Pin Value (GPIOD_PIN)

Register	Address	R/W	Description	Reset Value
GPIOD_PIN	GP_BA+0x3C	R	GPIO Port D Pin Value	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				PIN[3:0]			

Bits	Descriptions		Default
[3:0]	**PIN**	**Port D Pin Values**	0x000X

Interrupt Debounce 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**
Reserved	DBCLKSEL			DBEN			

Bits	Descriptions		Default
[6:4]	**DBCLKSEL**	**Debounce sampling cycle selection** **DBCLKSEL** / **Description** 0 – Always sample interrupt input 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	0x0
[3:0]	**DBEN**	**DBEN[x]: debounce sampling enable for each IRQx, x = 0 ~ 3** 1 = Interrupt input IRQx is filtered with de-bounce sampling 0 = Interrupt input IRQx is input directly without de-bounce sampling	0x0

IRQ Source Grouping (IRQSRCGPA)

Register	Address	R/W	Description	Reset Value
IRQSRCGPA	GP_BA+0x80	R/W	GPIO Port A IRQ Source Grouping	0x0000_0000

31	30	29	28	27	26	25	24
GPA15SEL		GPA14SEL		GPA13SEL		GPA12SEL	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPA11SEL		GPA10SEL		GPA9SEL		GPA8SEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**

GPA7SEL		GPA6SEL		GPA5SEL		GPA4SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPA3SEL		GPA2SEL		GPA1SEL		GPA0SEL	

Bits	Descriptions		Default
[2x+1:2x]	**GPAxSEL**	Selection for GPAx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x0

Where x=0~15.

GPAxSEL = 0, GPAx pin is grouped as one of interrupt sources to IRQ0.
1, GPAx pin is grouped as one of interrupt sources to IRQ1.
2, GPAx pin is grouped as one of interrupt sources to IRQ2.
3, GPAx pin is grouped as one of interrupt sources to IRQ3.

IRQ Source Grouping (IRQSRCGPB)

Register	Address	R/W	Description	Reset Value
IRQSRCGPB	GP_BA+0x84	R/W	GPIO Port B IRQ Source Grouping	0x5555_5555

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
GPB15SEL		GPB14SEL		GPB13SEL		GPB12SEL	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPB11SEL		GPB10SEL		GPB9SEL		GPB8SEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GPB7SEL		GPB6SEL		GPB5SEL		GPB4SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPB3SEL		GPB2SEL		GPB1SEL		GPB0SEL	

Bits	Descriptions		Default
[2x+1:2x]	**GPBxSEL**	Selection for GPBx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x1

Where x=0~15.

GPAxSEL = 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_AAAA

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
RESERVED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RESERVED							
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	0x2

Where x=0~7.

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 (IRQSRCGPE)

Register	Address	R/W	Description	Reset Value
IRQSRCGPE	GP_BA+0x8C	R/W	GPIO Port E IRQ Source Grouping	0xFFFF_FFFF

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
GPE15SEL		GPE14SEL		GPE13SEL		GPE12SEL	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPE11SEL		GPE10SEL		GPE9SEL		GPE8SEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
GPE7SEL		GPE6SEL		GPE5SEL		GPE4SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPE3SEL		GPE2SEL		GPE1SEL		GPE0SEL	

Bits	Descriptions		Default
[2x+1:2x]	**GPExSEL**	Selection for GPEx as one of input Pins to IRQ0, IRQ1, IRQ2, or IRQ3 interrupt source	0x3

Where x=0~15.

GPExSEL = 0, GPEx pin is grouped as one of interrupt sources to IRQ0.
1, GPEx pin is grouped as one of interrupt sources to IRQ1.
2, GPEx pin is grouped as one of interrupt sources to IRQ2.
3, GPEx pin is grouped as one of interrupt sources to IRQ3.

GPIO A Interrupt Enable (IRQENGPA)

Register	Address	R/W	Description	Reset Value
IRQENGPA	GP_BA+0x90	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.
NOTE3: When use a pin as power-down wake up source, if both edges are set, the high level will be set as wake up level.

GPIO B Interrupt Enable (IRQENGPB)

Register	Address	R/W	Description	Reset Value
IRQENGPB	GP_BA+0x94	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.
NOTE3: When use a pin as power-down wake up source, if both edges are set, the high level will be set as wake up level.

GPIO C Interrupt Enable (IRQENGPC)

Register	Address	R/W	Description	Reset Value
IRQENGPC	GP_BA+0x98	R/W	GPIO Port C Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PC7ENR	PC6ENR	PC5ENR	PC4ENR	PC3ENR	PC2ENR	PC1ENR	PC0ENR
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
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=0~7.
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.
NOTE3: When use a pin as power-down wake up source, if both edges are set, the high level will be set as wake up level.

GPIO E Interrupt Enable (IRQENGPE)

Register	Address	R/W	Description	Reset Value

IRQENGPE	GP_BA+0x9C	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.
NOTE3: When use a pin as power-down wake up source, if both edges are set, the high level will be set as wake up level.

Interrupt Latch Trigger Selection (IRQLHSEL)

Register	Address	R/W	Description	Reset Value
IRQLHSEL	GP_BA+0xA0	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**
RESERVED				IRQ3LHE	IRQ2LHE	IRQ1LHE	IRQ0LHE

Bits	Descriptions		Default
[x]	**IRQxLHE**	While IRQxLTH is "1", it enables active IRQx interrupt to latch the input values of GPAx/GPBx to IRQLHGPA/IRQLHGPB register simultaneously.	0x0

Where x=0~3.

GPIO A Interrupt Latch (IRQLHGPA)

Register	Address	R/W	Description	Reset Value
IRQLHGPA	GP_BA+0xA4	R	GPIO Port A 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**
PA15LHV	PA14LHV	PA13LHV	PA12LHV	PA11LHV	PA10LHV	PA9LHV	PA8LHV
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PA7LHV	PA6LHV	PA5LHV	PA4LHV	PA3LHV	PA2LHV	PA1LHV	PA0LHV

Bits	Descriptions		Default
[x]	**PAxLHV**	Latched value of GPAx while the IRQ (IRQ0~IRQ3) selected by IRQLHSEL is active.	0x0

Where x=0~15.

GPIO B Interrupt Latch (IRQLHGPB)

Register	Address	R/W	Description	Reset Value
IRQLHGPB	GP_BA+0xA8	R	GPIO Port B 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**
PB15LHV	PB14LHV	PB13LHV	PB12LHV	PB11LHV	PB10LHV	PB9LHV	PB8LHV
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PB7LHV	PB6LHV	PB5LHV	PB4LHV	PB3LHV	PB2LHV	PB1LHV	PB0LHV

Bits	Descriptions		Default
[x]	**PBxLHV**	Latched value of GPBx while the IRQ (IRQ0~IRQ3) selected by IRQLHSEL is active.	0x0

Where x=0~15.

GPIO C Interrupt Latch (IRQLHGPC)

Register	Address	R/W	Description	Reset Value
IRQLHGPC	GP_BA+0xA4	R	GPIO Port C 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
RESERVED							
7	6	5	4	3	2	1	0
PC7LHV	PC6LHV	PC5LHV	PC4LHV	PC3LHV	PC2LHV	PC1LHV	PC0LHV

Bits	Descriptions		Default
[x]	**PCxLHV**	Latched value of GPCx while the IRQ (IRQ0~IRQ3) selected by IRQLHSEL is active.	0x0

Where x=0~7.

GPIO E Interrupt Latch (IRQLHGPE)

Register	Address	R/W	Description	Reset Value
IRQLHGPE	GP_BA+0xB0	R	GPIO Port E Interrupt Latch Value	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED							
23	22	21	20	19	18	17	16
RESERVED							
15	14	13	12	11	10	9	8
PE15LHV	PE14LHV	PE13LHV	PE12LHV	PE11LHV	PE10LHV	PE9LHV	PE8LHV
7	6	5	4	3	2	1	0
PE7LHV	PE6LHV	PE5LHV	PE4LHV	PE3LHV	PE2LHV	PE1LHV	PE0LHV

Bits	Descriptions		Default
[x]	**PExLHV**	Latched value of GPEx while the IRQ (IRQ0~IRQ3) selected by IRQLHSEL is active.	0x0

Where 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+0xB4	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.	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.	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+0xB8	R/C	IRQ0~3 Interrupt Trigger Source Indicator from GPIO Port C and GPIO Port E	0x0000_0000

31	30	29	28	27	26	25	24
PE15TG	PE14TG	PE13TG	PE12TG	PE11TG	PE10TG	PE9TG	PE8TG
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PE7TG	PE6TG	PE5TG	PE4TG	PE3TG	PE2TG	PE1TG	PE0TG
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RESERVED							
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.	0x0
[y+16]	**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.	0x0

Where x=0~7, y=0~15.

NOTE: The trigger source will be latched when the corresponding rising or falling trigger enable is setup and the pin state toggle is recognised (through debounce or without debounce), no matter whether the source is an input or output pin.

Other NOTE for related setup

NOTE1: For the AIC's normal functionality to be triggered by external IO interrupt, the external IRQ source settings (for IRQ0/1/2/3) can only be set as positive edge, see AIC_SCRxx/bit7~6: SRCTYPE.

NOTE2: For power-down wake up setting, in order to keep normal wake up functionality, the wake up source polarity should be set as positive level, see IRQWAKECON/bit7~4: IRQWAKEUPPOL.

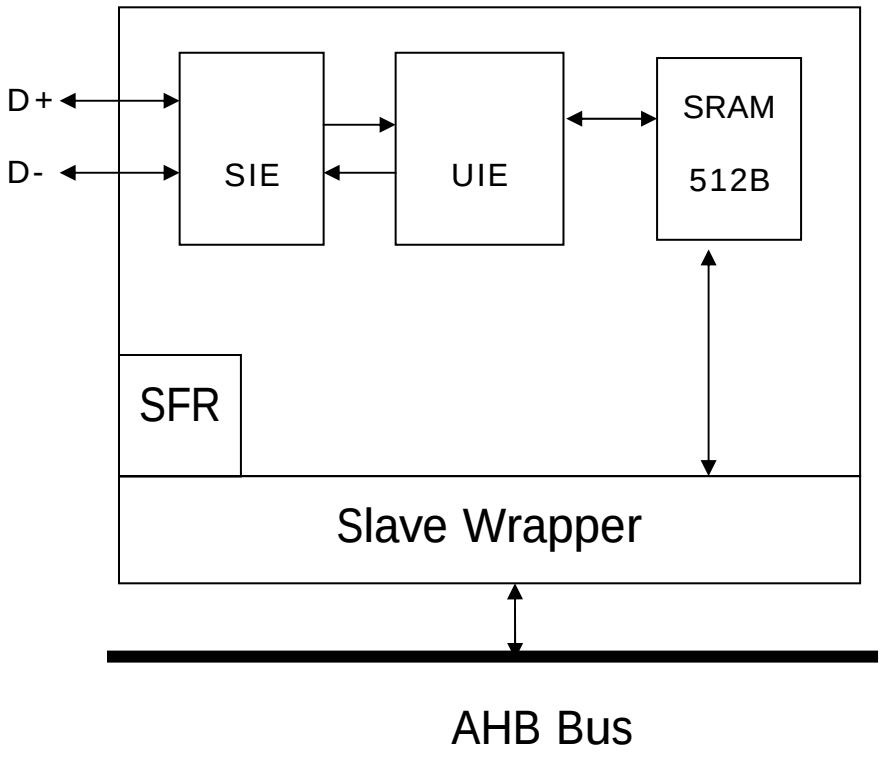
4.17 USB Device Controller

4.17.1 Feature

This Universal Serial Bus (USB) performs a serial interface with a single connector type for attaching all USB peripherals to the host system. Following is the feature list of this USB.

- Conforming to USB1.1 Device
  - Full Speed 12Mbps.
- Provide 1 interrupt source with 4 interrupt events.
- Support Control, Bulk In/Out, Interrupt and Isochronous transfers.
- Suspend when no bus signaling for 3 ms.
- Provide 6 endpoints.
- Include 512 Bytes internal SRAM as USB buffer.
- Provide remote wakeup capability.

4.17.2 Block diagram



4.17.3 Memory Mapping

Address	Size	Type	Description
FFF0_C000h ~ C0FFh	256 Bytes	Register	Special function register
FFF0_C100h ~ C2FFh	512 Bytes	SRAM	USB buffer

4.17.4 USB Control Registers

Register	Address	R/W	Description	Reset Value
USB_BA = 0xFFF0_C000				
IEF	USB_BA+0x000	R/W	Interrupt Enable Flag	0x0000_0000
EVF	USB_BA+0x004	R	Interrupt Event Flag	0x0000_0000
FADDR	USB_BA+0x008	R/W	Function Address	0x0000_0000
STS	USB_BA+0x00C	R, W	System state	0x0000_00x0
ATTR	USB_BA+0x010	W R/W R&C	Bus state & attribution	0x0000_0040
FLODET	USB_BA+0x014	R	Floating detect	0x0000_0000
BUFSEG	USB_BA+0x018	R/W	Buffer Segmentation	0x0000_0000
USBCFG	USB_BA+0x01C	R/W	USB configuration, internal test only	0x0000_0000
BUFSEG0	USB_BA+0x020	R/W	Buffer Segmentation of endpoint 0	0x0000_0000
MXPLD0	USB_BA+0x024	R/W	Maximal payload of endpoint 0	0x0000_0000
CFG0	USB_BA+0x028	R/W	Configuration of endpoint 0	0x0000_0000
CFGP0	USB_BA+0x02C	R/W W&C	stall control register and In/out ready clear flag of endpoint 0	0x0000_0000
BUFSEG1	USB_BA+0x030	R/W	Buffer Segmentation of endpoint 1	0x0000_0000
MXPLD1	USB_BA+0x034	R/W	Maximal payload of endpoint 1	0x0000_0000
CFG1	USB_BA+0x038	R/W	Configuration of endpoint 1	0x0000_0000
CFGP1	USB_BA+0x03C	R/W W&C	stall control register and In/out ready clear flag of endpoint 1	0x0000_0000
BUFSEG2	USB_BA+0x040	R/W	Buffer Segmentation of endpoint 2	0x0000_0000
MXPLD2	USB_BA+0x044	R/W	Maximal payload of endpoint 2	0x0000_0000
CFG2	USB_BA+0x048	R/W	Configuration of endpoint 2	0x0000_0000
CFGP2	USB_BA+0x04C	R/W W&C	stall control register and In/out ready clear flag of endpoint 2	0x0000_0000
BUFSEG3	USB_BA+0x050	R/W	Buffer Segmentation of endpoint 3	0x0000_0000
MXPLD3	USB_BA+0x054	R/W	Maximal payload of endpoint 3	0x0000_0000
CFG3	USB_BA+0x058	R/W	Configuration of endpoint 3	0x0000_0000

CFGP3	USB_BA+0x05C	R/W W&C	stall control register and In/out ready clear flag of endpoint 3	0x0000_0000
BUFSEG4	USB_BA+0x060	R/W	Buffer Segmentation of endpoint 4	0x0000_0000
MXPLD4	USB_BA+0x064	R/W	Maximal payload of endpoint 4	0x0000_0000
CFG4	USB_BA+0x068	R/W	Configuration of endpoint 4	0x0000_0000
CFGP4	USB_BA+0x06C	R/W W&C	stall control register and In/out ready clear flag of endpoint 4	0x0000_0000
BUFSEG5	USB_BA+0x070	R/W	Buffer Segmentation of endpoint 5	0x0000_0000
MXPLD5	USB_BA+0x074	R/W	Maximal payload of endpoint 5	0x0000_0000
CFG5	USB_BA+0x078	R/W	Configuration of endpoint 5	0x0000_0000
CFGP5	USB_BA+0x07C	R/W W&C	In ready clear flag of endpoint 5	0x0000_0000
DRVSE0	USB_BA+0x090	R/W	Force USB PHY to drive SE0 when set to 1	0x0000_0000
USBBIST	USB_BA+0x0A0	R/W	USB buffer test register	0x0000_0000

Interrupt Enable Register (IEF)

Register	Address	R/W	Description	Reset Value
IEF	USB_BA+0x000	R/W	Interrupt Enable Flag	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
INNAKen	RESERVED						wakeupEn
7	6	5	4	3	2	1	0
RESERVED				wakeup	fld	usb	bus

Bits	Descriptions		
[15]	**INNAKen**	W	Enable/disable IN NAK INT
[8]	**wakeupEN**	R/W	1/0 Enable/Disable USB wakeup function
[3]	**wakeup**	R/W	1/0 Enable/disable Wakeup Interrupt.
[2]	**fld**	R/W	1/0 Enable/disable Floating detect Interrupt
[1]	**usb**	R/W	1/0 Enable/disable USB event interrupt.
[0]	**bus**	R/W	1/0 Enable/disable BUS event interrupt.

Interrupt Event Flag Register (EVF)

This register is USB Interrupt Event Flag register, clear by read STS, ATTR or FLODET.

Register	Address	R/W	Description	Reset Value

EVF	USB_BA+0x004	R/W&C	Interrupt Event Flag	0x0000_0000

31	30	29	28	27	26	25	24
Setup	RESERVED						
23	22	21	20	19	18	17	16
RESERVED		EPTF5	EPTF4	EPTF3	EPTF2	EPTF1	EPTF0
15	14	13	12	11	10	9	8
RESERVED							
7	6	5	4	3	2	1	0
RESERVED				wakeup	fld	usb	bus

Bits	Descriptions		
[31]	**Setup**	R/W &C	1: Setup event occurred, cleared by read register “STS” or write 1to EVF[31].
[21]	**EPTF5**	R/W &C	1: USB event occurred, check STSX[17:15] to know which kind of USB event was occurred, cleared by read register “STS” or write 1 to EVF[21].
[20]	**EPTF4**	R/W &C	1: USB event occurred, check STSX[14:12] to know which kind of USB event was occurred, cleared by read register “STS” or write 1 to EVF[20].
[19]	**EPTF3**	R/W &C	1: USB event occurred, check STSX[11:9] to know which kind of USB event was occurred, cleared by read register “STS” or write 1 to EVF[19].
[18]	**EPTF2**	R/W &C	1: USB event occurred, check STSX[8:6] to know which kind of USB event was occurred, cleared by read register “STS” or write 1 to EVF[18].
[17]	**EPTF1**	R/W &C	1: USB event occurred, check STSX[5:3] to know which kind of USB event was occurred, cleared by read register “STS” or write 1 to EVF[17].
[16]	**EPTF0**	R/W &C	1: USB event occurred, check STSX[2:0] to know which kind of USB event was occurred, cleared by read register “STS” or write 1 to EVF[16].
[3]	**wakeup**	R/W &C	Wakeup event occurred, cleared by write 1 to EVF[3]
[2]	**fld**	R/W &C	Floating detect event occurred, cleared by write 1 to EVF[2].
[1]	**usb**	R/W &C	USB event occurred, check STS[6:4] or STS0~5[2:0] to know which kind of USB event was occurred, cleared by write 1 to EVF[1] or EPTF0~5 and Setup = 0.
[0]	**bus**	R/W &C	Bus event occurred, check ATTR[3:0] to know which kind of bus event was occurred, cleared by write 1 to EVF[0].

Function Address Register (FADDR)

A seven-bit value uses as the address of a device on the USB BUS.

Register	Address	R/W	Description	Reset Value
FADDR	USB_BA+0x008	R/W	Function Address	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	FADDR						

Bits	Descriptions		
[6:0]	**FADDR**	W/R	Function Address of this USB device.

System States Register (STS)

Register	Address	R/W	Description	Reset Value
STS	USB_BA+0x00C	R	System states	0x0000_00x0

31	30	29	28	27	26	25	24
RESERVED						STS5	
23	22	21	20	19	18	17	16
STS5	STS4			STS3			STS2
15	14	13	12	11	10	9	8
STS2		STS1			STS0		
7	6	5	4	3	2	1	0
Overrun	STS			EPT			

Bits	Descriptions		
[25:23]	**STS5**	R	System states of endpoint 5 000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end

[22:20]	**STS4**	R	System states of endpoint 4 000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end
[19:17]	**STS3**	R	System states of endpoint 3 000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end
[16:14]	**STS2**	R	System states of endpoint 2 000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end
[13:11]	**STS1**	R	System states of endpoint 1 000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end
[10:8]	**STS0**	R	System states of endpoint 0 000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end
[7]	**Overrun**	R	Out Data more than Max Payload or Setup Data more than 8 Bytes
[6:4]	**STS**	R	000: In ACK 001: In NAK 010: Out 0 ACK 110: Out 1 ACK 011: Setup ACK 111: Isochronous translation end
[3:0]	**EPT**	R	Endpoint number

Bus States & Attribution Register (ATTR)

Register	Address	R/W	Description	Reset Value
ATTR	USB_BA+0x010	W, R/W	Bus states & attribution	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							
7	6	5	4	3	2	1	0
enUSB	RESERVED	RWakeup	enPHY	Timeout	Resume	Suspend	usbRST

Bits	Descriptions		
[7]	**enUSB**	R/W	1/0: Enable/disable USB
[5]	**RWakeUp**	R/W	1: force USB bus to K state, used for remote wake-up.
[4]	**enPHY**	R/W	1/0: Enable/disable PHY
[3]	**Timeout**	R	No response more than 18 bits time
[2]	**Resume**	R	Resume from suspension
[1]	**Suspend**	R	Bus idle more than 3mS, either cable is plugged off or host is sleeping.
[0]	**usbRST**	R	Bus reset when SE0(single-ended 0) more than 2.5uS.

Floating detection Register (FLODET)

Register	Address	R/W	Description	Reset Value
FLODET	USB_BA+0x014	R	Floating detection	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							FLODET

Bits	Descriptions		
[0]	**FLODET**	R	1/0: connected/floating

Buffer Segmentation Register (BUFSEG)

For Setup token only.

Register	Address	R/W	Description	Reset Value
BUFSEG	USB_BA+0x018	R/W	Buffer segmentation	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							BUFSEG
7	6	5	4	3	2	1	0
BUFSEG					RESERVED		

Bits	Descriptions		
[8:3]	**BUFSEG**	R/W	For Setup token only. Effective starting address USB buffer = {BUFSEG[7:3], 3'b000}

USB Configuration Register (USBCFG)

For Setup token only.

Register	Address	R/W	Description	Reset Value
USBCFG	USB_BA+0x01C	R/W	USB configuration, internal test only	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							RCVmode

Bits	Descriptions		
[0]	**RCVmode**	R/W	1/0 : RCV signal come from RXD/RXDP

Buffer Segmentation Register (BUFSEGx) x = 0~5

Register	Address	R/W	Description	Reset Value
BUFSEG0	USB_BA+0x020	R/W	Buffer segmentation of endpoint 0	0x0000_0000
BUFSEG1	USB_BA+0x030	R/W	Buffer segmentation of endpoint 1	0x0000_0000
BUFSEG2	USB_BA+0x040	R/W	Buffer segmentation of endpoint 2	0x0000_0000

BUFSEG3	USB_BA+0x050	R/W	Buffer segmentation of endpoint 3	0x0000_0000
BUFSEG4	USB_BA+0x060	R/W	Buffer segmentation of endpoint 4	0x0000_0000
BUFSEG5	USB_BA+0x070	R/W	Buffer segmentation of endpoint 5	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							BUFSEG
7	6	5	4	3	2	1	0
BUFSEG					RESERVED		

Bits	Descriptions		
[8:3]	**BUFSEG**	R/W	Effective starting address USB buffer = {BUFSEG[7:3], 3'b000}

Maximal Payload Register (MXPLDx) x = 0~5

Register	Address	R/W	Description	Reset Value
MXPLD0	USB_BA+0x024	R/W	Maximal Payload of endpoint 0	0x0000_0000
MXPLD1	USB_BA+0x034	R/W	Maximal Payload of endpoint 1	0x0000_0000
MXPLD2	USB_BA+0x044	R/W	Maximal Payload of endpoint 2	0x0000_0000
MXPLD3	USB_BA+0x054	R/W	Maximal Payload of endpoint 3	0x0000_0000
MXPLD4	USB_BA+0x064	R/W	Maximal Payload of endpoint 4	0x0000_0000
MXPLD5	USB_BA+0x074	R/W	Maximal Payload of endpoint 5	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							MXPLD
7	6	5	4	3	2	1	0
MXPLD							

Bits	Descriptions

[8:0]	**MXPLD**	R/W	**Read:** IN: Length of Data transmitting to host. Out: Max Length of Data receiving from host. **Write:** IN: Length of Data to transmit to host, assert INrdy (IN buffer ready). OUT: Max Length of Data receiving from host, assert OUTrdy (OUT buffer ready). **Note:** once MXPLD is filled out, the data packets will be transmitted/received immediately after IN/OUT token arrived.

Configuration Register (CFGx) x = 0~5

Register	Address	R/W	Description	Reset Value
CFG0	USB_BA+0x028	R/W	Configuration of Endpoint 0	0x0000_0000
CFG1	USB_BA+0x038	R/W	Configuration of Endpoint 1	0x0000_0000
CFG2	USB_BA+0x048	R/W	Configuration of Endpoint 2	0x0000_0000
CFG3	USB_BA+0x058	R/W	Configuration of Endpoint 3	0x0000_0000
CFG4	USB_BA+0x068	R/W	Configuration of Endpoint 4	0x0000_0000
CFG5	USB_BA+0x078	R/W	Configuration of Endpoint 5	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						stall_ctl	RESERVED
7	6	5	4	3	2	1	0
DSQ	state		ISOCH	EPT			

Bits	Descriptions		
[9]	**stall_ctl**	R/W	0: disable auto clear stall 1: enable auto clear stall in setup stage
[7]	**DSQ**	R/W	Specify Data 0 or 1 after IN token toggle automatically after host ACK.
[6:5]	**state**	R/W	00: endpoint is disabled 01: Out endpoint 10: IN endpoint 11: undefined
[4]	**ISOCH**	R/W	Isochronous, no handshake.
[3:0]	**EPT**	R/W	Endpoint number.

Extra Configuration Register (CFGPx) x = 0~5

Register	Address	R/W	Description	Reset Value
CFGP0	USB_BA+0x02C	R/W W&C	stall control register and In/out ready clear flag of endpoint 0	0x0000_0000
CFGP 1	USB_BA+0x03C	R/W W&C	stall control register and In/out ready clear flag of endpoint 1	0x0000_0000
CFGP 2	USB_BA+0x04C	R/W W&C	stall control register and In/out ready clear flag of endpoint 2	0x0000_0000
CFGP 3	USB_BA+0x05C	R/W W&C	stall control register and In/out ready clear flag of endpoint 3	0x0000_0000
CFGP 4	USB_BA+0x06C	R/W W&C	stall control register and In/out ready clear flag of endpoint 4	0x0000_0000
CFGP 5	USB_BA+0x07C	R/W W&C	stall control register and In/out ready clear flag of endpoint 5	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						stall	CFGP

Bits	Descriptions		
[1]	**stall**	R/W	Force device to response STALL
[0]	**CFGP**	W&C	IN: Write '1' to clear in ready that was set by MXPLD. OUT: Write '1' to clear out ready that was set by MXPLD

Drive SE0 Register (DRVSE0)

Register	Address	R/W	Description	Reset Value
DRVSE0	USB_BA+0x090	R/W	Force USB PHY to drive SE0 while setting to 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
RESERVED							DRVSE0

Bits	Descriptions		
[0]	**DRVSE0**	R/W	1/0: drive se0/none

USB Buffer Test Register USBBIST

Register	Address	R/W	Description	Reset Value
USBBIST	USB_BA+0x0A0	R/W	USB buffer self test 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					BistFail	Finish	BISTEN

Bits	Descriptions		
[2]	**BISTFail**	R	**BIST Fail** The BistFail indicates if the BIST test fails or succeeds. If the BistFail is low at the end, the embedded SRAM pass the BIST test, otherwise, it is faulty. The BistFail will be high once the BIST detects the error and remains high during the BIST operation. The BistFail is a write clear field. Write 1 to this field clears the content and write 0 has no effect.
[1]	**Finish**	R	**BIST Operation Finish** It indicates the end of the BIST operation. When BIST controller finishes all operations, this bit will be set high. This bit is a write clear field. Write 1 to this field clears the content and write 0 has no effect.
[0]	**BISTEN**	R/W	**BIST mode enable** 1 = BIST is enabled; begin to perform BIST on selected memory group. 0 = BIST is disabled or completed (automatically cleared by BIST controller).

4.17.5 USB Function Description

This USB provides 6 endpoints for a full speed USB device. Most handshakes between Host and Device are handled by hardware. Any USB event will cause an interrupt, and a user can just check related event flags to acknowledge the events and store required data into buffer, which is then sent to host by hardware. A software-disable function is also available for this USB device, which simulates the

disconnection of this device from the host.

4.17.5.1 Interrupt

This USB provides 1 interrupt source with 4 interrupt events (wakeup, flo, USB, BUS). Wakeup interrupt is for stop wakeup only, flo interrupt is for USB plug-in or unplug, USB event notifies users of some USB requests, like IN ACK, OUT ACK etc., and BUS event notifies users of some bus events, like suspend, resume, etc. A user must enable both AIC and IEF of USB to enable USB interrupts.

Wakeup interrupt is only present after stop wakeup. After the IC enters power down mode, any change on D+, D- and floating detect pin can wake up VA91 (provided that USB wakeup function is enabled). If this change is not desired, for example, a noise on floating detect pin, no interrupt but wakeup interrupt will occur. After USB wakeup, this interrupt will occur when no other USB interrupt events are present for more than 20mS.

We provide hardware de-bounce for USB floating detect interrupt to avoid bounce problems on USB plug in and unplug. Floating detect interrupt appears about 10 ms later than USB plug-in and unplug. A user can acknowledge USB plug-in/unplug by reading SFR “FLODET”, and should understand that the flag in “FLODET” represents the current state on the bus without de-bounce. If the user poling this flag to check USB state, he/she must add software de-bounce if necessary.

USB interrupt is to notify users of any USB event on the bus, and a user can read SFR “STSX” and “EPTF” to acknowledge what kind of request is to which endpoint and take necessary responses.

Same as USB interrupt, BUS interrupt notifies users of some bus events, like USB reset, suspend, timeout, and resume. A user can read SFR “ATTR” to acknowledge bus events.

4.17.5.2 Power Saving

USB turns off PHY automatically to save power while VA91 enter power down mode. Furthermore, a user can write 0 into SFT ATTR[4] to turn off PHY under special circumstances like suspend to save power.

4.18 USB Host Controller

4.18.1 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.

4.18.2 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.

4.18.3 Operation

4.18.3.1 OHCI Controller

4.18.3.1.1 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.

4.18.3.1.2 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.

4.18.3.1.3 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.

4.18.4 Register Mapping

Register	Offset	R/W	Description	Reset Value
Capability Registers (USBH_BA = 0xFFF0_D000)				
HcRev	USBH_BA+0x000	R	Host Controller Revision Register	0x0000_0010
HcControl	USBH_BA+0x004	R/W	Host Controller Control Register	0x0000_0000
HcComSts	USBH_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000
HcIntSts	USBH_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000
HcIntEn	USBH_BA+0x010	R/W	Host Controller Interrupt Enable Register	0x0000_0000
HcIntDis	USBH_BA+0x014	R/W	Host Controller Interrupt Disable Register	0x0000_0000
HcHCCA	USBH_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000
HcPerCED	USBH_BA+0x01C	R/W	Host Controller Period Current ED Register	0x0000_0000
HcCtrHED	USBH_BA+0x020	R/W	Host Controller Control Head ED Register	0x0000_0000
HcCtrCED	USBH_BA+0x024	R/W	Host Controller Control Current ED Register	0x0000_0000
HcBlkHED	USBH_BA+0x028	R/W	Host Controller Bulk Head ED Register	0x0000_0000
HcBlkCED	USBH_BA+0x02C	R/W	Host Controller Bulk Current ED Register	0x0000_0000
HcDoneH	USBH_BA+0x030	R/W	Host Controller Done Head Register	0x0000_0000
HcFmIntv	USBH_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

HcFmRem	USBH_BA+0x038	R	Host Controller Frame Remaining Register	0x0000_0000
HcFNum	USBH_BA+0x03C	R	Host Controller Frame Number Register	0x0000_0000
HcPerSt	USBH_BA+0x040	R/W	Host Controller Periodic Start Register	0x0000_0000
HcLSTH	USBH_BA+0x044	R/W	Host Controller Low Speed Threshold Register	0x0000_0628
HcRhDeA	USBH_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0902
HcRhDeB	USBH_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000
HcRhSts	USBH_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000
HcRhPrt1	USBH_BA+0x054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPrt2	USBH_BA+0x058	R/W	Host Controller Root Hub Port Status [2]	0x0000_0000
	USBH_BA+0x05C ... USBH_BA+0x1FC		Reserved	Undefined
OHCI USB Configuration Register				
MiscCtrl	USBH_BA+0x200	R/W	USB Miscellaneous Control Register	0x0000_0000
OpModEn	USBH_BA+0x204	R/W	USB Operational Mode Enable Register	0X0000_0000

4.18.5 Register Details

Host Controller Revision Register (HcRev)

Register	Address	R/W	Description	Reset Value
HcRev	USBH_BA+0x000	R	Host Controller Revision Register	0x0000_0010

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

Bits	Descriptions	
[31:8]	**Reserved**	
[7:0]	**Rev**	**Revision** Indicates the Open HCI Specification revision number implemented by the Hardware. Host Controller supports 1.0 specification. (X.Y = XYh)

Host Controller Control Register (HcControl)

Register	Address	R/W	Description	Reset Value
HcControl	USBH_BA+0x004	R/W	Host Controller Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					RWakeEn	RWake	IntRoute
7	6	5	4	3	2	1	0
HcFunc		BlkEn	CtrlEn	ISOEn	PeriEn	CtrlBlkRatio	

Bits	Descriptions	
[31:11]	**Reserved**	
[10]	**RWakeEn**	**Remote Wakeup Connected Enable** If a remote wakeup signal is supported, this bit enables that operation. Since there is no remote wakeup signal supported, this bit is ignored.
[9]	**RWake**	**Remote Wakeup Connected** This bit indicated whether the HC supports a remote wakeup signal. This implementation does not support any such signal. The bit is hard-coded to '0.'
[8]	**IntRoute**	**Interrupt Routing** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
[7:6]	**HcFunc**	**Host Controller Functional State** This field sets the Host Controller state. The Controller may force a state change from USBSUSPEND to USBRESUME after detecting resume signaling from a downstream port. States are: 00: USBRESET 01: USBRESUME 10: USBOPERATIONAL 11: USBSUSPEND
[5]	**BlkEn**	**Bulk List Enable** When set this bit enables processing of the Bulk list.
[4]	**CtrlEn**	**Control List Enable** When set this bit enables processing of the Control list.
[3]	**ISOEn**	**Isochronous List Enable** When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous ED.

[2]	**PeriEn**	**Periodic List Enable** When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.
[1:0]	**CtrlBlkRatio**	**Control Bulk Service Ratio** Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e. '00' = 1 Control Endpoint; '11' = 3 Control Endpoints)

Host Controller Command Status Register (HcComSts)

Register	Address	R/W	Description	Reset Value
HcComSts	USBH_BA+0x008	R/W	Host Controller Command Status Register	0x0000_0000

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

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]	**OCReq**	**Ownership Chang Request** When set by software, this bit sets the **OwnershipChange** field in *HcInterruptStatus*. The bit is cleared by software.
[2]	**BlkFill**	**Bulk List Filled** Set to indicate there is an active ED on the Bulk List. The bit may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Bulk List.
[1]	**CtrlFill**	**Control List Filled** Set to indicate there is an active ED on the Control List. It may be set by either software or the Host Controller and cleared by the Host Controller each time it begins processing the head of the Control List.
[0]	**HCReset**	**Host Controller Reset** This bit is set to initiate the software reset. This bit is cleared by the Host Controller, upon completed of the reset operation.

Host Controller Interrupt Status Register (HcIntSts)

Register	Address	R/W	Description	Reset Value
HcIntSts	USBH_BA+0x00C	R/W	Host Controller Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	OC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RHSC	FNOF	UnRecErr	Resume	SOF	WBDnHD	SchOR

Bits	Descriptions	
[31]	**Reserved**	
[30]	**OC**	**Ownership Change** This bit is set when the **OwnershipChangeRequest** bit of *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]	**UnRecErr**	**Unrecoverable Error** This event is not implemented and is hard-coded to '0.' Writes are ignored.
[3]	**Resume**	**Resume Detected** Set when Host Controller detects resume signaling on a downstream port.
[2]	**SOF**	**Start Of Frame** Set when the Frame Management block signals a 'Start of Frame' event.
[1]	**WBDnHD**	**Write Back Done Head** Set after the Host Controller has written *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	USBH_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	URErrEn	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]	**URErrEn**	**Unrecoverable Error Enable** This event is not implemented. All writes to this bit are ignored.
[3]	**ResuEn**	**Resume Detected Enable** 0: Ignore 1: Enables interrupt generation due to Resume Detected.
[2]	**SOFEn**	**Start Of Frame Enable** 0: Ignore 1: Enables interrupt generation due to Start of Frame.
[1]	**WBDHEn**	**Write Back Done Head Enable** 0: Ignore 1: Enables interrupt generation due to Write-back Done Head.
[0]	**SchOREn**	**Scheduling Overrun Enable** 0: Ignore 1: Enables interrupt generation due to Scheduling Overrun.

Host Controller Interrupt Disable Register (HcIntDis)

Register	Address	R/W	Description	Reset Value
HcIntDis	USBH_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	URErrDis	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]	**URErrDis**	**Unrecoverable Error Disable** This event is not implemented. All writes to this bit are ignored.
[3]	**ResuDis**	**Resume Detected Disable** 0: Ignore 1: Disables interrupt generation due to Resume Detected.
[2]	**SOFDis**	**Start Of Frame Disable** 0: Ignore 1: Disables interrupt generation due to Start of Frame.
[1]	**WBDHDis**	**Write Back Done Head Disable** 0: Ignore 1: Disables interrupt generation due to Write-back Done Head.
[0]	**SchORDis**	**Scheduling Overrun Disable** 0: Ignore 1: Disables interrupt generation due to Scheduling Overrun.

Host Controller Communication Area Register (HcHCCA)

Register	Address	R/W	Description	Reset Value
HcHCCA	USBH_BA+0x018	R/W	Host Controller Communication Area Register	0x0000_0000

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

Bits	Descriptions	
[31:7]	**HCCA**	**Host Controller Communication Area** Pointer to HCCA base address.
[7:0]	**Reserved**	

Host Controller Period Current ED Register (HcPerCED)

Register	Address	R/W	Description	Reset Value
HcPerCED	USBH_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	USBH_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	USBH_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	USBH_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	USBH_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	USBH_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	USBH_BA+0x034	R/W	Host Controller Frame Interval Register	0x0000_2EDF

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

Bits	Descriptions	
[31]	**FmIntvT**	**Frame Interval Toggle** This bit is toggled by HCD when it loads a new value into **FrameInterval**.
[30: 16]	**FSDPktCnt**	**FS Largest Data Packet** This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
[15:14]	**Reserved**	
[13:0]	**FmInterval**	**Frame Interval** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

Host Controller Frame Remaining Register (HcFmRem)

Register	Address	R/W	Description	Reset Value
HcFmRem	USBH_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	USBH_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	USBH_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 Root Hub Descriptor A Register (HcRhDeA)

Register	Address	R/W	Description	Reset Value
HcRhDeA	USBH_BA+0x048	R/W	Host Controller Root Hub Descriptor A Register	0x0100_0902

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

Bits	Descriptions	
[31:24]	**PwrGDT**	Power On to Power Good Time This field value is represented as the number of 2 ms intervals, which ensuring that the power switching is effective within 2 ms. Only bits [25:24] are implemented as R/W. The remaining bits are read only as '0'. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should be written to support system implementation. This field should always be written to a non-zero value.
[23:13]	**Reserved**	
[12]	**NOCP**	**No Over Current Protection** Global over-current reporting implemented in HYDRA-2. This bit should be written to support the external system port over-current implementation. 0 = Over-current status is reported 1 = Over-current status is not reported

[11]	**OCPM**	**Over Current Protection Mode** Global over-current reporting implemented in HYDRA-2. This bit should be written 0 and is only valid when **NoOverCurrentProtection** is cleared. 0 = Global Over-Current 1 = Individual Over-Current
[10]	**DevType**	**Device Type** HYDRA-4is not a compound device.
[9]	**NPS**	**No Power Switching** Global power switching implemented in HYDRA-2. This bit should be written to support the external system port power switching implementation. 0 = Ports are power switched. 1 = Ports are always powered on.
[8]	**PSM**	**Power Switching Mode** Global power switching mode implemented in HYDRA-2. This bit is only valid when **NoPowerSwitching** is cleared. This bit should be written '0'. 0 = Global Switching 1 = Individual Switching
[7:0]	**DPortNum**	**Number Downstream Ports** HYDRA-4 supports two downstream ports.

Host Controller Root Hub Descriptor B Register (HcRhDeB)

Register	Address	R/W	Description	Reset Value
HcRhDeB	USBH_BA+0x04C	R/W	Host Controller Root Hub Descriptor B Register	0x0000_0000

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

Bits	Descriptions

[31:16]	**PPCM**	**Port Power Control Mask** Global-power switching. This field is only valid if **NoPowerSwitching** is cleared and **PowerSwitchingMode** is set (individual port switching). When set, the port only responds to individual port power switching commands (**Set/ClearPortPower**). When cleared, the port only responds to global power switching commands (**Set/ClearGlobalPower**). 0 = Device not removable 1 = Global-power mask Port Bit relationship - Unimplemented ports are reserved, read/write '0'. 0 : Reserved 1 : Port 1 2 : Port 2 ... 15 : Port 15
[15:0]	**DevRemove**	**Device Removable** HYDRA-4 ports default to removable devices. 0 = Device not removable 1 = Device removable Port Bit relationship 0 : Reserved 1 : Port 1 2 : Port 2 ... 15 : Port 15 Unimplemented ports are reserved, read/write '0'.

Host Controller Root Hub Status Register (HcRhSts)

Register	Address	R/W	Description	Reset Value
HcRhSts	USBH_BA+0x050	R/W	Host Controller Root Hub Status Register	0x0000_0000

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

Bits	Descriptions	
[31]	**RWEClr**	**Clear Remote Wakeup Enable** Writing a '1' to this bit clears **DeviceRemoteWakeupEnable**. Writing a '1' has no effect.
[30:18]	**Reserved**	

[17]	**OCIC**	**Over Current Indicator Change** This bit is set when **OverCurrentIndicator** changes. Writing a '1' clears this bit. Writing a '0' has no effect.
[16]	**LPSC**	(Read) **LocalPowerStatusChange** Not supported. Always read '0'. (Write) **SetGlobalPower** Write a '1' issues a **SetGlobalPower** command to the ports. Writing a '0' has no effect.
[15]	**DRWEn**	(Read) **DeviceRemoteWakeupEnable** This bit enables ports' **ConnectStatusChange** as a remote wakeup event. 0 = disabled 1 = enabled (Write) **SetRemoteWakeupEnable** Writing a '1' sets **DeviceRemoteWakeupEnable**. Writing a '0' has no effect.
[14:2]	**Reserved**	
[1]	**OC**	**Over Current Indicator** This bit reflects the state of the OVRCUR pin. This field is only valid if **NoOverCurrentProtection** and **OverCurrentProtectionMode** are cleared. 0 = No over-current condition 1 = Over-current condition
[0]	**LPS**	(Read) **LocalPowerStatus** Not Supported. Always read '0'. (Write) **ClearGlobalPower** Writing a '1' issues a **ClearGlobalPower** command to the ports. Writing a '0' has no effect.

Host Controller Root Hub Port Status (HcRhPrt [1: 2])

Register	Address	R/W	Description	Reset Value
HcRhPrt1	USBH_BA+0x054	R/W	Host Controller Root Hub Port Status [1]	0x0000_0000
HcRhPrt2	USBH_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	USBH_BA+0x200	R/W	USB Miscellaneous Control Register	0X0000_0000

31	30	29	28	27	26	25	24
Reserved				StbyEn			
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	USBH_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							
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	
[31:9]	**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.

4.19 PWM-Timer

4.19.1 Introduction

The W55VA91 have 4 channels pwm-timers. The 4 channels pwm-timers has 2 Prescaler, 2 clock divider, 4 clock selectors, 4 16-bit counters, 4 16-bit comparators, 2 Dead-Zone generator. They are all driven by system clock. Each channel can be used as a timer and issue interrupt independently.
Each two channels pwm-timers share the same prescaler(channel0-1 share prescalar0 and channel2-3 share prescalar1). Clock divider provides each channel with 5 clock sources (1, 1/2, 1/4, 1/8, 1/16). Each channel receives its own clock signal from clock divider which receives clock from 8-bit prescaler. The 16-bit counter in each channel 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-timer is blocked. Two output pin are all used as Dead-Zone generator output signal to control off-chip power device. Dead-Zone generator 0 are used to control outputs of channel0&1, and Dead-Zone generator 1 are used to control outputs of channel2&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.

The w55va91 have 4 channels pwm-timers and 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.

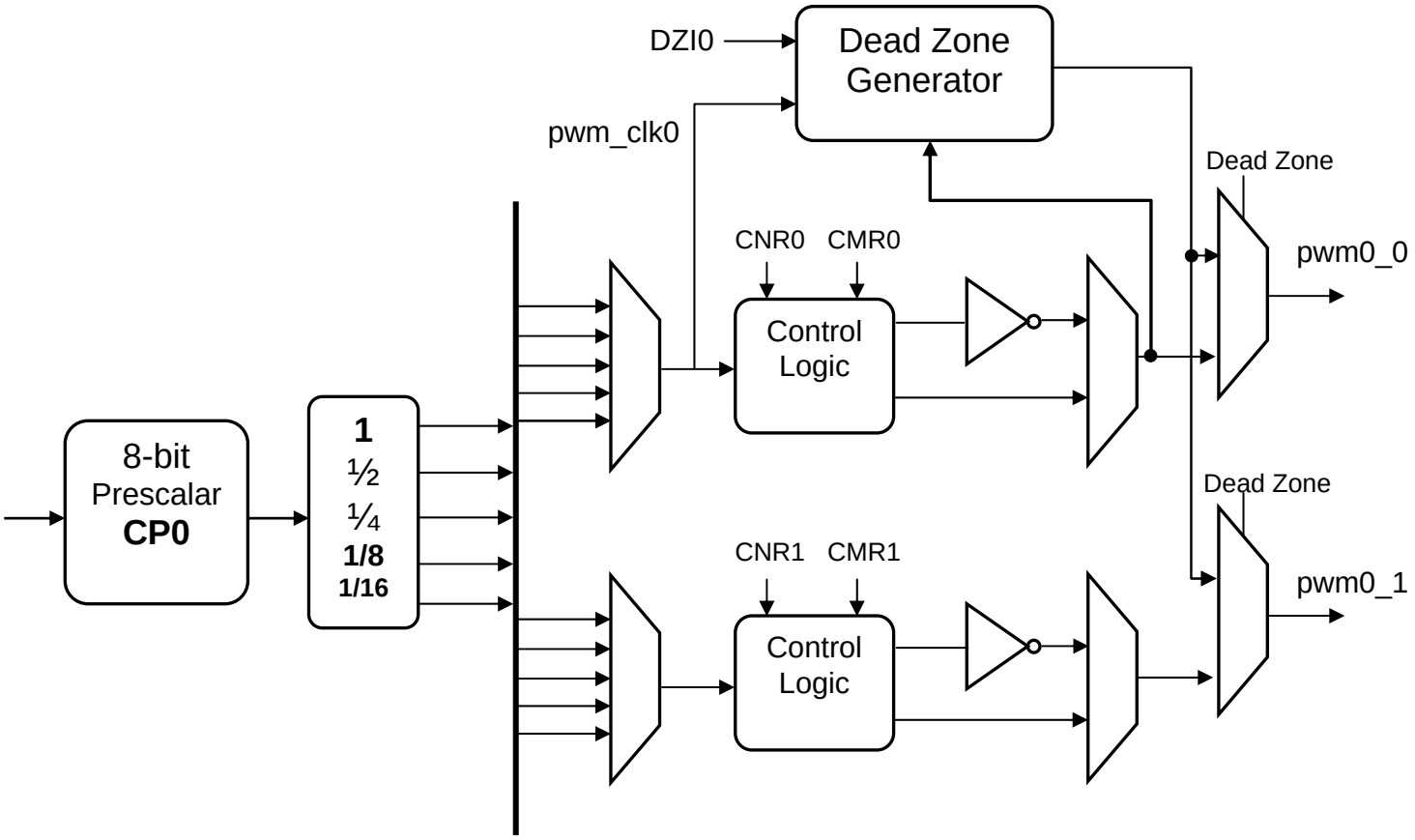
4.19.2 Features

- Two 8-bit prescalers and Two clock dividers

- Four clock selectors
- Four 16-bit counters and four 16-bit comparators
- Two Dead-Zone generator
- Capture function

4.19.3 PWM Architecture

The following figure describes the architecture of pwm in one group.(channel0&1 are in one group and channel2&3 are in another group)



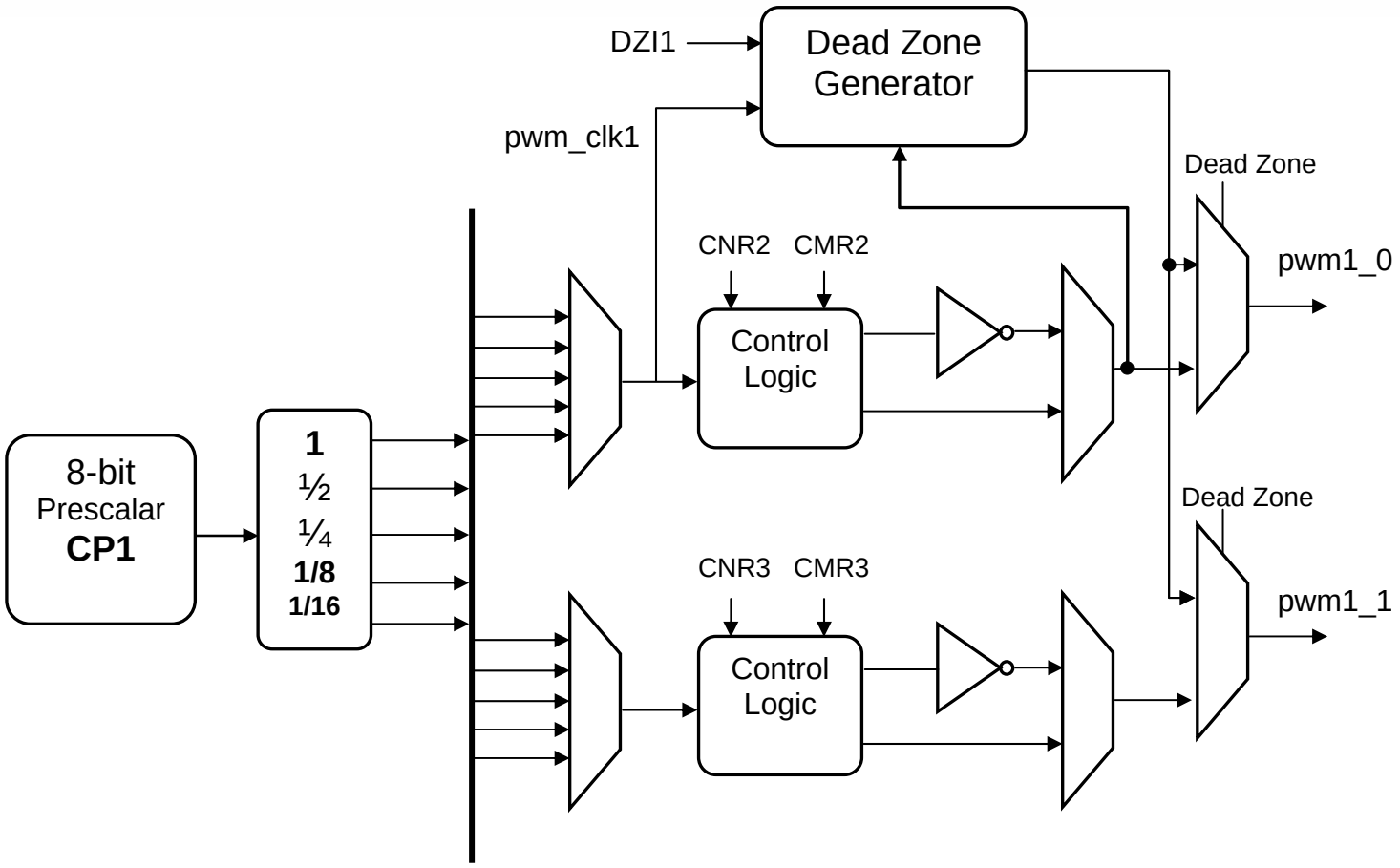


Figure 4.19-1 PWM Architecture Diagram

4.19.4 Basic Timer Operation

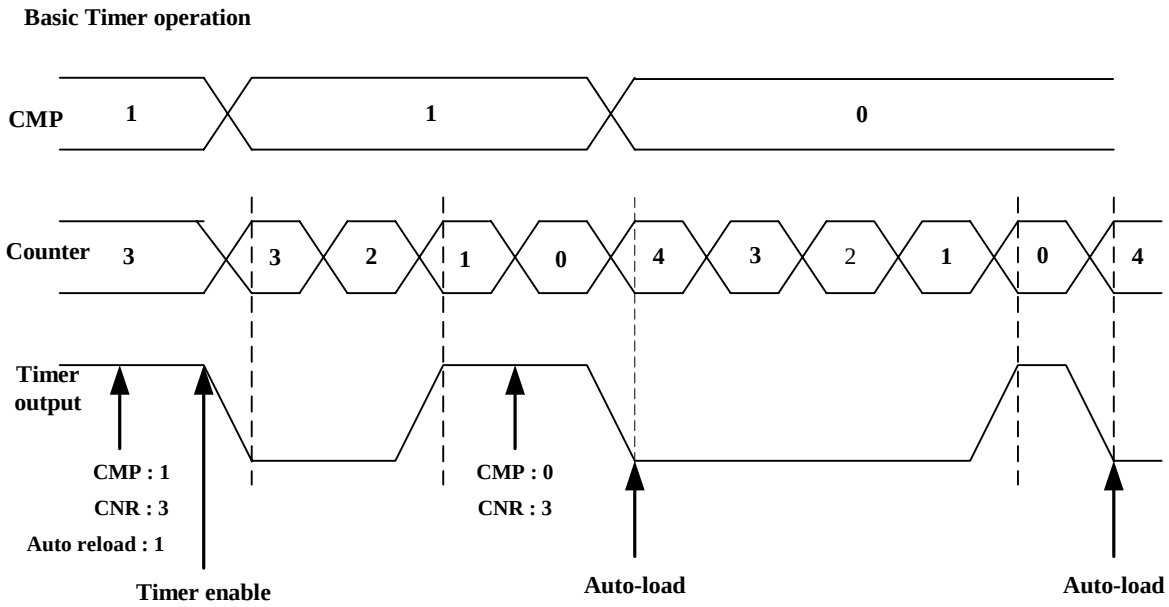


Figure 4.19-2 Basic Timer Operation Timing

4.19.5 PWM Double Buffering and Automatic Reload

W55VA91 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 copies 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

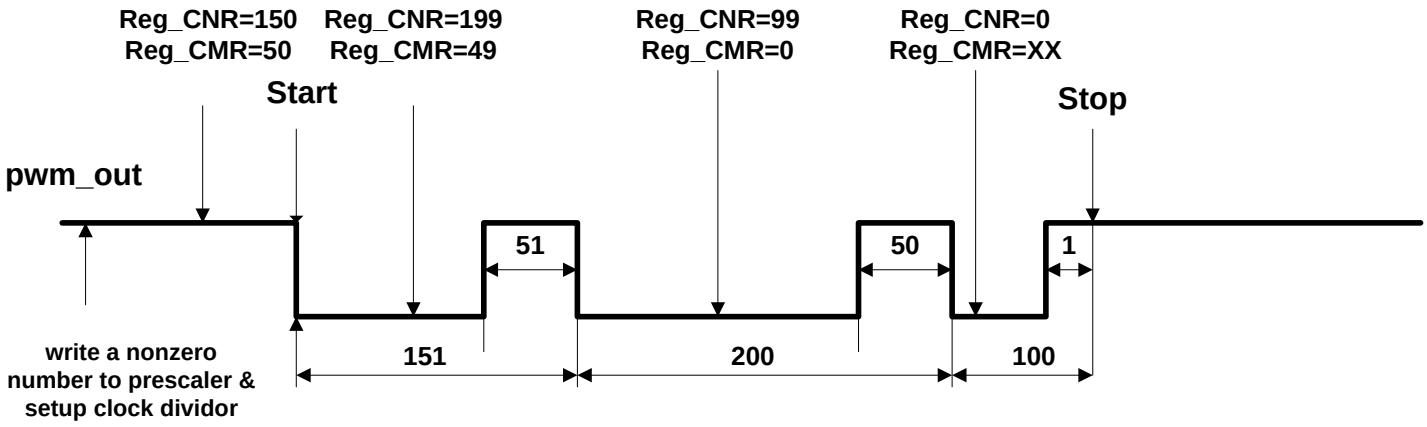


Figure 4.19-3 PWM Double Buffering Illustration

4.19.6 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)

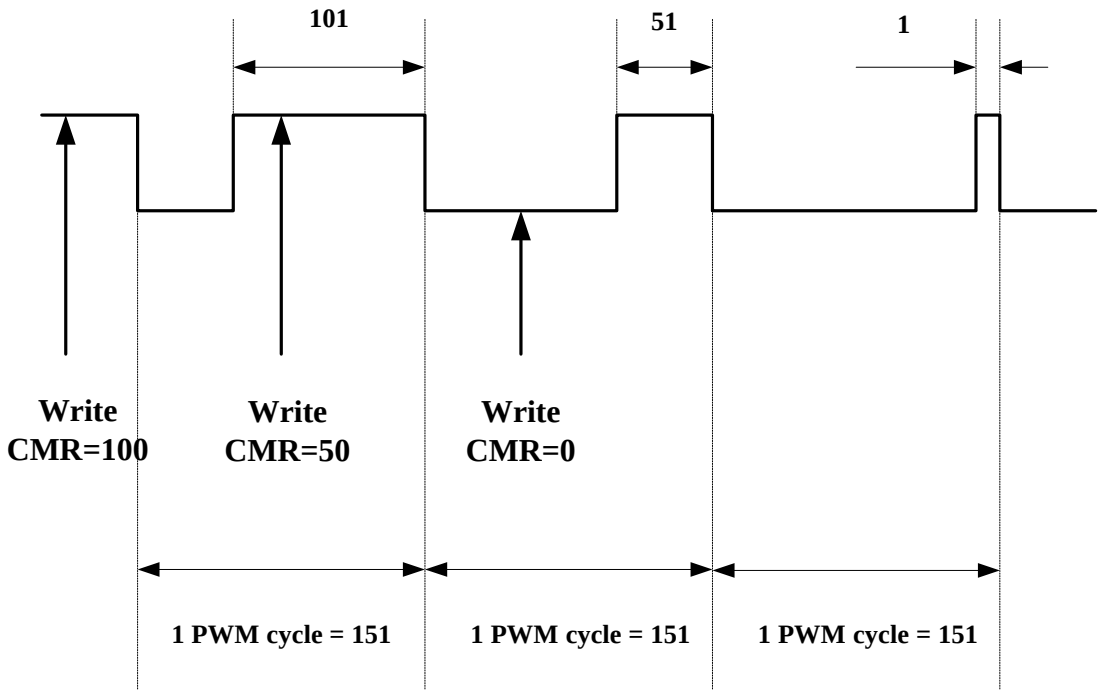


Figure 4.19-4 PWM Controller Output Duty Ratio

4.19.7 Dead-Zone Generator

W55VA91 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

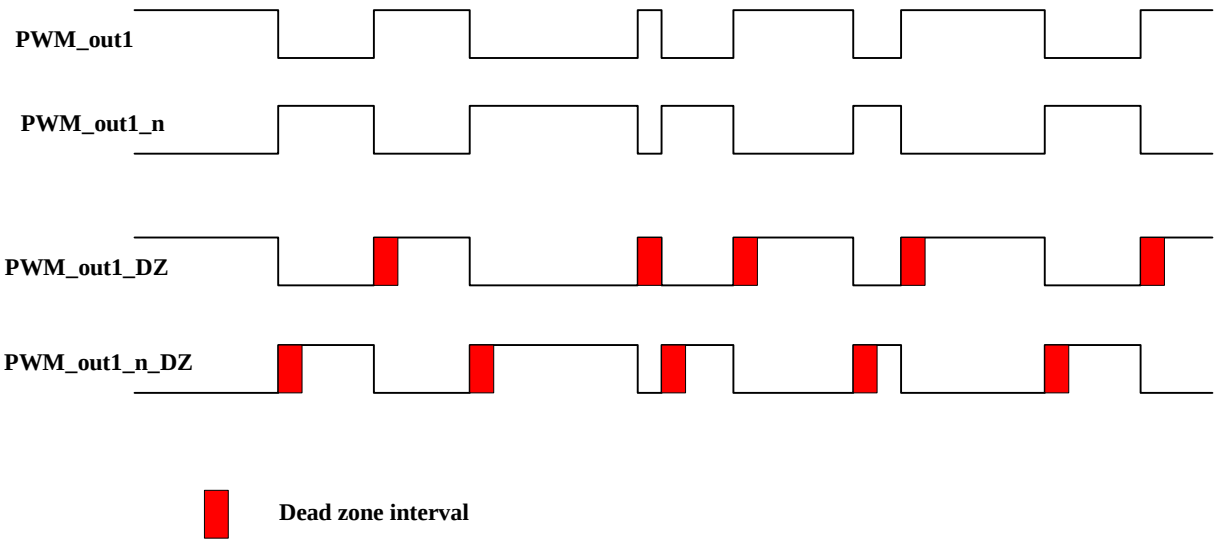


Figure 4.19-5 Dead Zone Generation Operation

4.19.8 PWM Timer Start Procedure

1. Setup clock selector (CSR)
2. Setup prescaler & 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. Enable pwm timer (PCR)

4.19.9 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)***

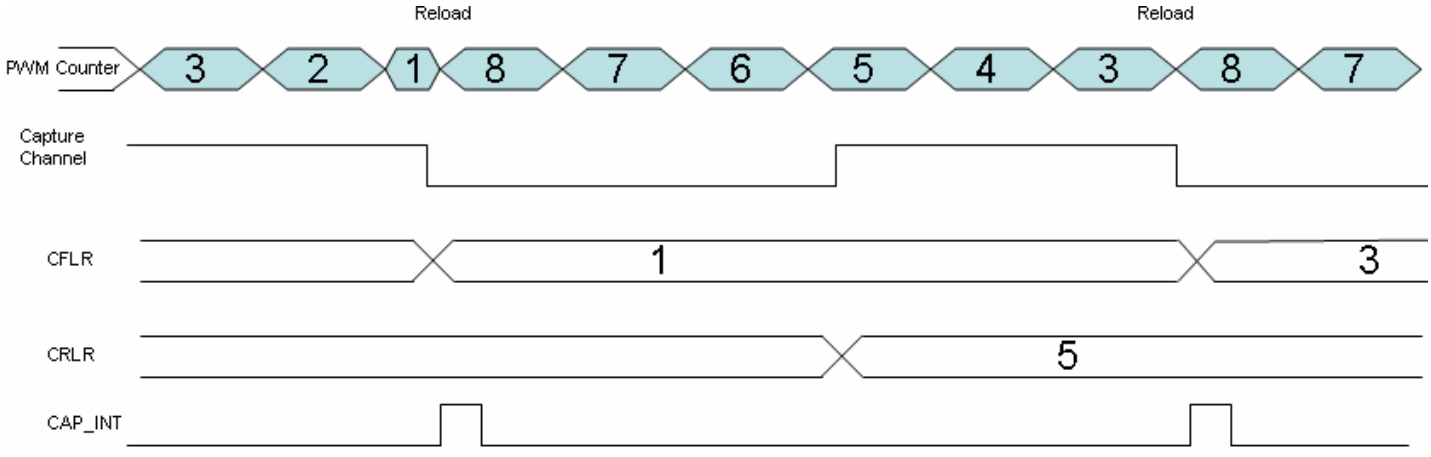
Method 2:
Set 16-bit down counter (CNR) as 0. When interrupt request happen, disable pwm timer (PCR). ***(Recommended)***

Method 3:
Disable pwm timer directly (PCR). ***(Not recommended)***

4.19.10 Capture Start Procedure

1. Setup clock selector (CSR)
2. Setup prescaler & 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 capture register (CCR)
7. Enable pwm timer (PCR)

4.19.11 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 (CNR – CRLR).
3. The channel high pulse width is (CRLR - CFLR).
4. The channel cycle time is (CNR – CFLR).

4.19.12 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 = 0xFFF8_2000				
PPR	PWM_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000
CSR	PWM_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000
PCR	PWM_BA+0x008	R/W	PWM Control Register	0x0000_0000
CNR0	PWM_BA+0x00C	R/W	PWM Counter Register 0	0x0000_0000
CMR0	PWM_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000
PDR0	PWM_BA+0x014	R	PWM Data Register 0	0x0000_0000
CNR1	PWM_BA+0x018	R/W	PWM Counter Register 1	0x0000_0000
CMR1	PWM_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000
PDR1	PWM_BA+0x020	R	PWM Data Register 1	0x0000_0000
CNR2	PWM_BA+0x024	R/W	PWM Counter Register 2	0x0000_0000
CMR2	PWM_BA+0x028	R/W	PWM Comparator Register 2	0x0000_0000
PDR2	PWM_BA+0x02C	R	PWM Data Register 2	0x0000_0000
CNR3	PWM_BA+0x030	R/W	PWM Counter Register 3	0x0000_0000
CMR3	PWM_BA+0x034	R/W	PWM Comparator Register 3	0x0000_0000
PDR3	PWM_BA+0x038	R	PWM Data Register 3	0x0000_0000
PIER	PWM_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000
PIIR	PWM_BA+0x044	R/C	PWM Interrupt Indication Register	0x0000_0000
CCR0	PWM_BA+0x050	R/W	Capture Control Register 0	0x0000_0000

CCR1	PWM_BA+0x054	R/W	Capture Control Register 1	0x0000_0000
CRLR0	PWM_BA+0x058	R/W	Capture Rising Latch Register (Channel 0)	0x0000_0000
CFLR0	PWM_BA+0x05C	R/W	Capture Falling Latch Register (Channel 0)	0x0000_0000
CRLR1	PWM_BA+0x060	R/W	Capture Rising Latch Register (Channel 1)	0x0000_0000
CFLR1	PWM_BA+0x064	R/W	Capture Falling Latch Register (Channel 1)	0x0000_0000
CRLR2	PWM_BA+0x068	R/W	Capture Rising Latch Register (Channel 2)	0x0000_0000
CFLR2	PWM_BA+0x06C	R/W	Capture Falling Latch Register (Channel 2)	0x0000_0000
CRLR3	PWM_BA+0x070	R/W	Capture Rising Latch Register (Channel 3)	0x0000_0000
CFLR3	PWM_BA+0x074	R/W	Capture Falling Latch Register (Channel 3)	0x0000_0000
CAPENR	PWM_BA+0x078	R/W	Capture Input Enable Register	0x0000_0000

4.19.13 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 prescaler 1 for PWM Timer channel 2 & 1 Clock input is divided by (CP0 + 1) before it is fed to the counter. 2 & 3 If CP1=0, then the prescaler 1 output clock will be stopped.
[7:0]	**CP0**	**Clock prescaler 0 for PWM Timer channel 0 & 1** Clock input is divided by (CP0 + 1) before it is fed to the counter. 0 & 1 If CP0=0, then the prescaler 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	
[14:12]	CSR3	Channel 3 Clock Source Selection Select clock input for channel 3. CSR3 [14:12] / Input clock divided by: 100 → 1; 011 → 16; 010 → 8; 001 → 4; 000 → 2
[10:8]	CSR2	Channel 2 Clock Source Selection Select clock input for channel 0. (Table is the same as CH3)
[6:4]	**CSR1**	**Channel 1 Clock Source Selection** Select clock input for channel 0. (Table is the same as CH3)
[2:0]	**CSR0**	**Channel 0 Clock Source Selection** Select clock input for channel 0. (Table is the same as CH3)

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	
[27]	CH3MOD	Channel 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	Channel 3 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[24]	CH3EN	Channel 3 Enable/Disable 1: Enable 0: Disable
[19]	CH2MOD	Channel 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	Channel 2 Inverter ON/OFF 1: Inverter ON 0: Inverter OFF
[16]	CH2EN	Channe l2 Enable/Disable 1: Enable 0: Disable
[11]	**CH1MOD**	**Channel 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**	**Channel 1 Inverter ON / OFF** 1: Inverter ON 0: Inverter OFF
[8]	**CH1EN**	**Channel 1 Enable / Disable** 1: Enable 0: Disable
[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**	**Channel 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**	**Channel 0 Inverter ON/OFF** 1: Inverter ON 0: Inverter OFF
[0]	**CH0EN**	**Channel 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	
[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	
[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 2	0x0000_0000
PDR3	PWM_BA+0x038	R	PWM Data 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
PDR [15:8]							

7	6	5	4	3	2	1	0
PDR [7:0]							

Bits	Descriptions	
[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	
[3]	PIER3	PWM Timer Channel 3 Interrupt Enable 1: Enable 0: Disable
[2]	PIER2	PWM Timer Channel 2 Interrupt Enable 1: Enable 0: Disable
[1]	**PIER1**	PWM Timer Channel 1 Interrupt Enable 1: Enable 0: Disable
[0]	**PIER0**	PWM Timer Channel 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	
[3]	PIIR3	PWM Timer Channel 3 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF
[2]	PIIR2	PWM Timer Channel 2 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF
[1]	**PIIR1**	PWM Timer Channel 1 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF
[0]	**PIIR0**	PWM Timer Channel 0 Interrupt Flag 1: Interrupt Flag ON 0: Interrupt Flag OFF

Note: User can clear each interrupt flag by writing a zero 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	
[23]	**CFLRD1**	**CFLR1 dirty bit** When input channel 1 has a rising transition, CFLR1 was updated and this bit was “1”.
[22]	**CRLRD1**	**CRLR1 dirty bit** When input channel 1 has a falling transition, CRLR1 was updated and this bit was “1”.
[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.
[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
[7]	**CFLRD0**	CFLR0 dirty bit When input channel 0 has a rising transition, CFLR0 was updated and this bit was “1”.
[6]	**CRLRD0**	CRLR0 dirty bit When input channel 0 has a falling transition, CRLR0 was updated and this bit was “1”.
[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.
[3]	**CAPCH0EN**	Capture Channel 0 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 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	
[23]	CFLRD3	CFLR3 dirty bit When input channel 3 has a rising transition, CFLR3 was updated and this bit was “1”.
[22]	CRLRD3	CRLR3 dirty bit When input channel 3 has a falling transition, CRLR3 was updated and this bit was “1”.
[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.
[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
[7]	CFLRD2	CFLR2 dirty bit When input channel 2 has a rising transition, CFLR2 was updated and this bit was “1”.
[6]	CRLRD2	CRLR2 dirty bit When input channel 2 has a falling transition, CRLR2 was updated and this bit was “1”.
[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.
[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	
[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	
[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**
CAPENR[7:0]							

Bits	Descriptions	
[7:0]	**CAPENR**	Capture Input Enable Register There are eight capture inputs from pad. Bit0~Bit7 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[7:0] 76543210 xxxxxxx1 ➔ Capture channel 0 is from GPA_DIN[7] xxxxx1x0 ➔ Capture channel 0 is from GPB_DIN[14] xxxxxx1x ➔ Capture channel 1 is from GPA_DIN[8] xxxx1x0x ➔ Capture channel 1 is from GPB_DIN[15] xxx1xxxx ➔ Capture channel 2 is from GPB_DIN[12] x1x0xxxx ➔ Capture channel 2 is from GPE_DIN[0] xx1xxxxx ➔ Capture channel 3 is from GPB_DIN[13] 1x0xxxxx ➔ Capture channel 3 is from GPE_DIN[1]

4.20 SPI0 Serial Interface Controller (Master/Slave)

4.20.1 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 successive.

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
- Fully static synchronous design with one clock domain
- Only Support the external master device that the frequency of its serial clock output is less 1/4 than the MICROWIRE/SPI Core clock input (pclk) and its slave select output is edge-active trigger.

4.20.2 SPI (MICROWIRE) Block Diagram (Master/Slave)

The block diagram of MICROWIRE/SPI Serial Interface controller is shown as following.

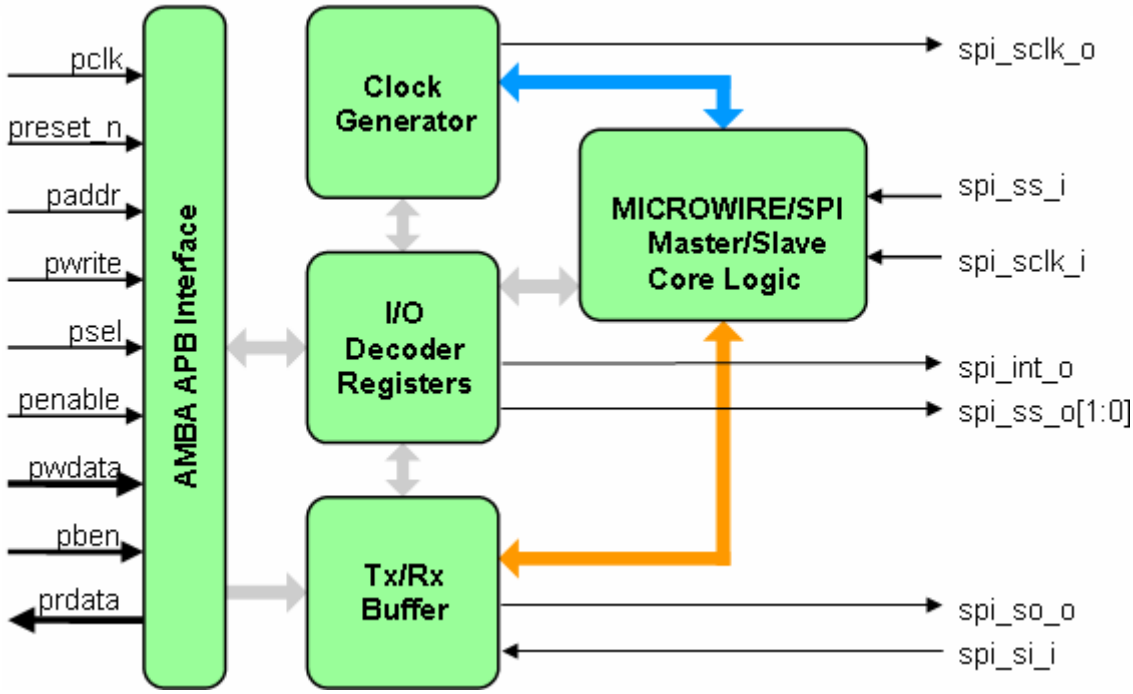


Figure 4.20-1 MICROWIRE/SPI Block Diagram (Master/Slave)

Pin descriptions:

spi_sclk_o: MICROWIRE/SPI master serial clock output pin.

spi_int_o: MICROWIRE/SPI interrupt signal output.

spi_ss_o[1:0]:MICROWIRE/SPI two slave/device select signals output.

spi_so_o: MICROWIRE/SPI serial data output pin (to slave device in master mode or to master device in slave mode).

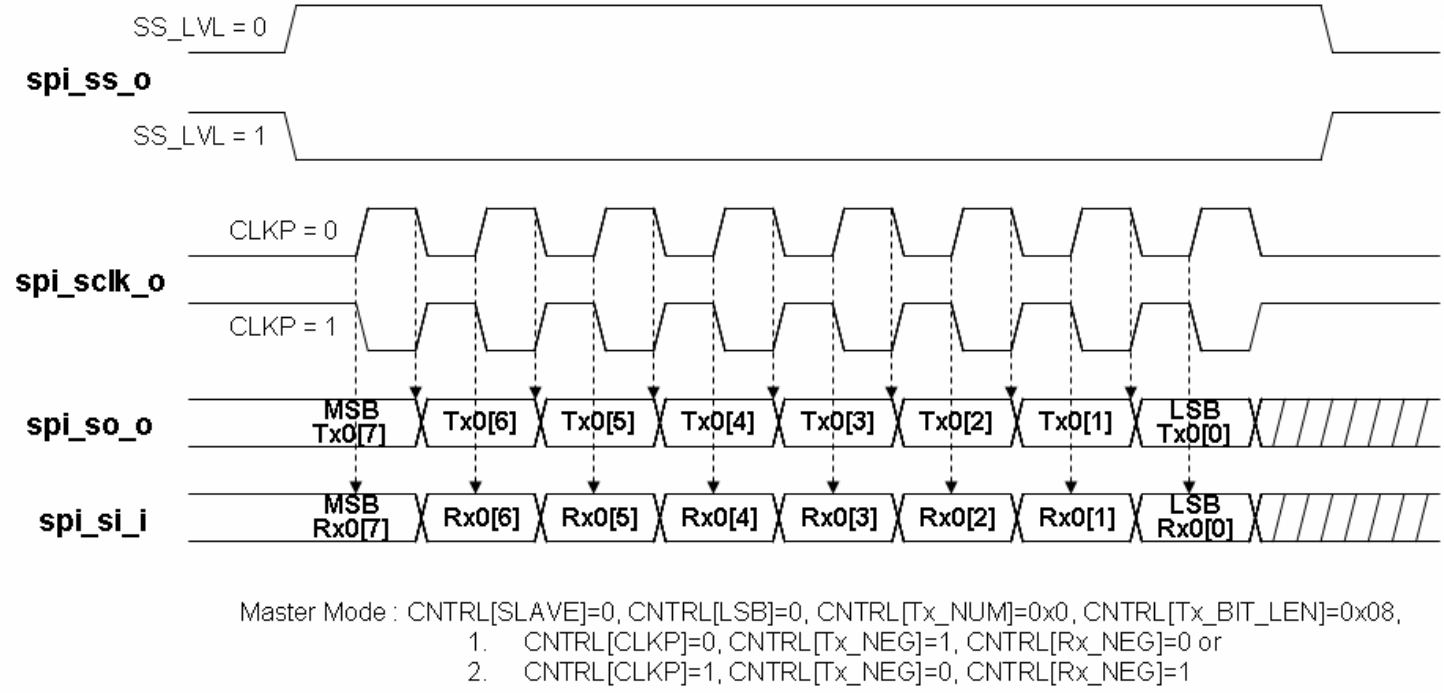
spi_si_i: MICROWIRE/SPI serial data input pin (from slave device in master mode or from master device in salve mode).

spi_sclk_i: MICROWIRE/SPI slave serial clock input pin.

spi_ss_i: MICROWIRE/SPI slave slave/device select signal input (edge-active trigger).

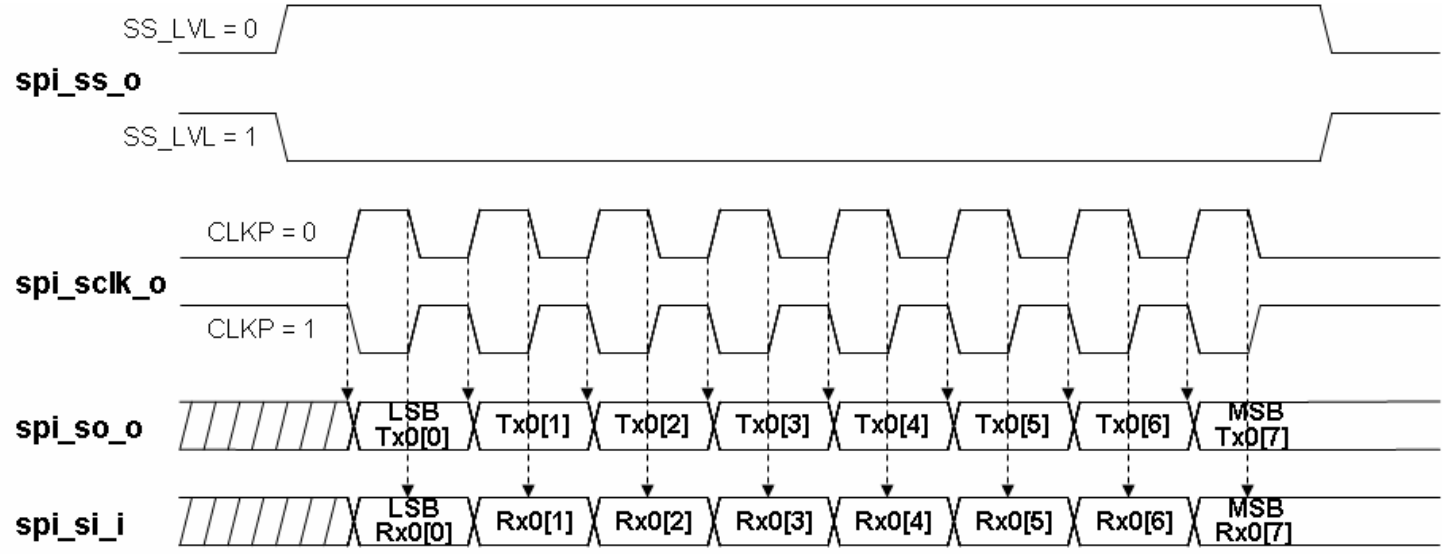
4.20.3 SPI (MICROWIRE) Timing Diagram (Master/Slave)

The timing diagrams of MICROWIRE/SPI Master/Slave are shown as following.



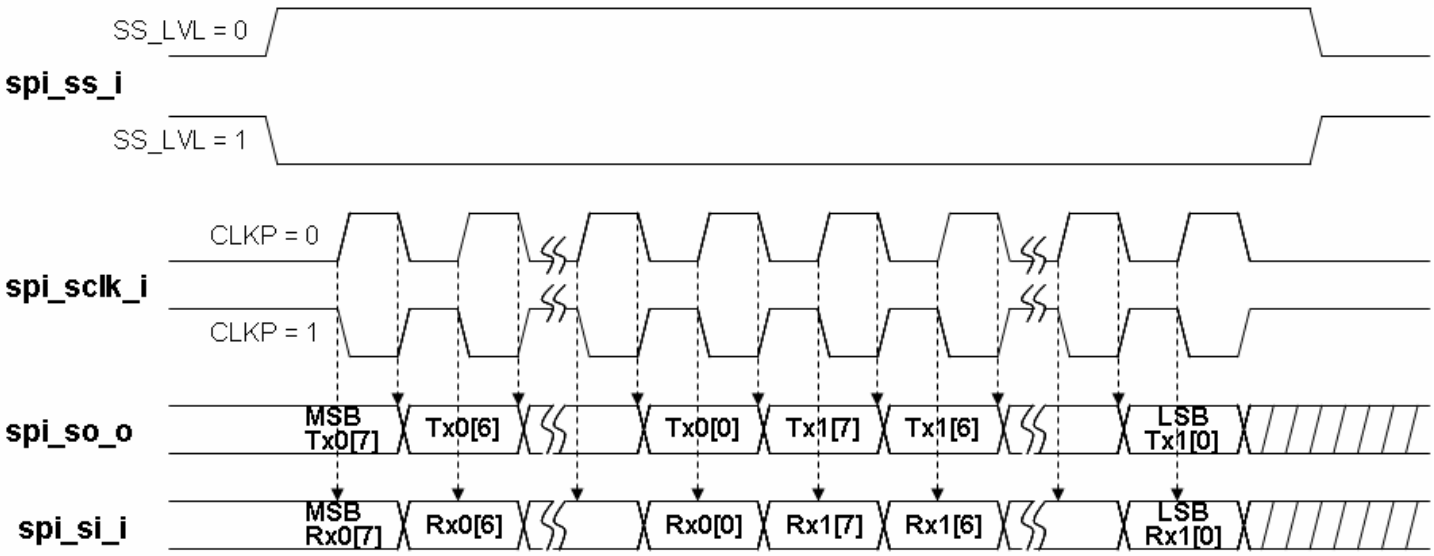
Master Mode : CNTRL[SLAVE]=0, CNTRL[LSB]=0, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0 or
2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1

Figure 4.20-2 MICROWIRE/SPI Timing (Master)



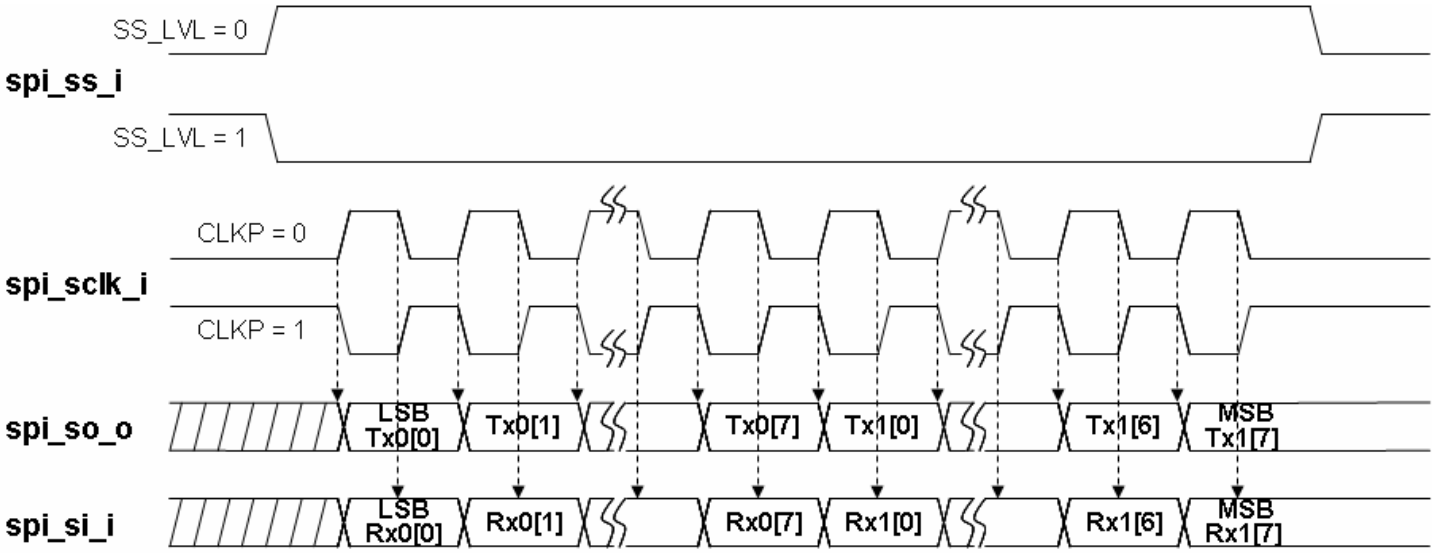
Master Mode : CNTRL[SLAVE]=0, CNTRL[LSB]=1, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
1. CNTRL[CLKP]=0, CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1 or
2. CNTRL[CLKP]=1, CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0

Figure 4.20-3 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 4.20-4 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 4.20-5 Alternate Phase SCLK Clock Timing (Slave)

4.20.4 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.

4.20.5 SPI Booting

If a SPI flash has been flashed through our software framework method, there'll be an 8-bytes specified ID in the very beginning and then followed by a User Boot Program inside SPI flash.

SPI Booting will fail if System Boot Code can't find this specified 8-bytes ID when booting.

4.20.6 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
SPI0_BA = 0xFFF8_6000				
SPI0_CNTRL	SPI0_BA + 0x00	R/W	Control and Status Register	0x0000_0004
SPI0_DIVIDER	SPI0_BA + 0x04	R/W	Clock Divider Register	0x0000_0000
SPI0_SSR	SPI0_BA + 0x08	R/W	Slave Select Register	0x0000_0000
Reserved	SPI0_BA + 0x0C	N/A	Reserved	N/A
SPI0_Rx0	SPI0_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI0_Rx1	SPI0_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI0_Rx2	SPI0_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI0_Rx3	SPI0_BA + 0x1C	R	Data Receive Register 3	0x0000_0000
SPI0_Tx0	SPI0_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI0_Tx1	SPI0_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI0_Tx2	SPI0_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI0_Tx3	SPI0_BA + 0x1C	W	Data Transmit Register 3	0x0000_0000

NOTE 1: When software programs CNTRL, the GO_BUSY bit should be written last.

4.20.7 SPI 0 (MICROWIRE) Control Register Description

Control and Status Register (CNTRL)

Register	Offset	R/W	Description	Reset Value
SPI0_CNTRL	SPI0_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					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	
[31:19]	**Reserved**	**Reserved**
[18]	**SLAVE**	**Interrupt Enable** 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 0x0. 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	SPI0_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 (master only)** 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.

Slave Select Register (SSR)

Register	Offset	R/W	Description	Reset Value
SPI0_SSR	SPI0_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				**ASS**	**SS_LVL**	**SSR[1:0]**	

Bits	Descriptions	
[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 SSR[1:0] 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	SPI0_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI0_Rx1	SPI0_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI0_Rx2	SPI0_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI0_Rx3	SPI0_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 FFs.

Data Transmit Register (TX)

Register	Offset	R/W	Description	Reset Value
SPI0_Tx0	SPI0_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI0_Tx1	SPI0_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI0_Tx2	SPI0_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI0_Tx3	SPI0_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 means 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.

4.21 SPI1 Serial Interface Controller (Master Only)

4.21.1 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. 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. This master 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 successive.

The MICROWIRE/SPI Master Core includes the following features:

- AMBA APB interface compatible
- Support MICROWIRE/SPI master 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
- Fully static synchronous design with one clock domain

4.21.2 SPI (MICROWIRE) Block Diagram (Master Only)

The block diagram of MICROWIRE/SPI Serial Interface controller is shown as following.

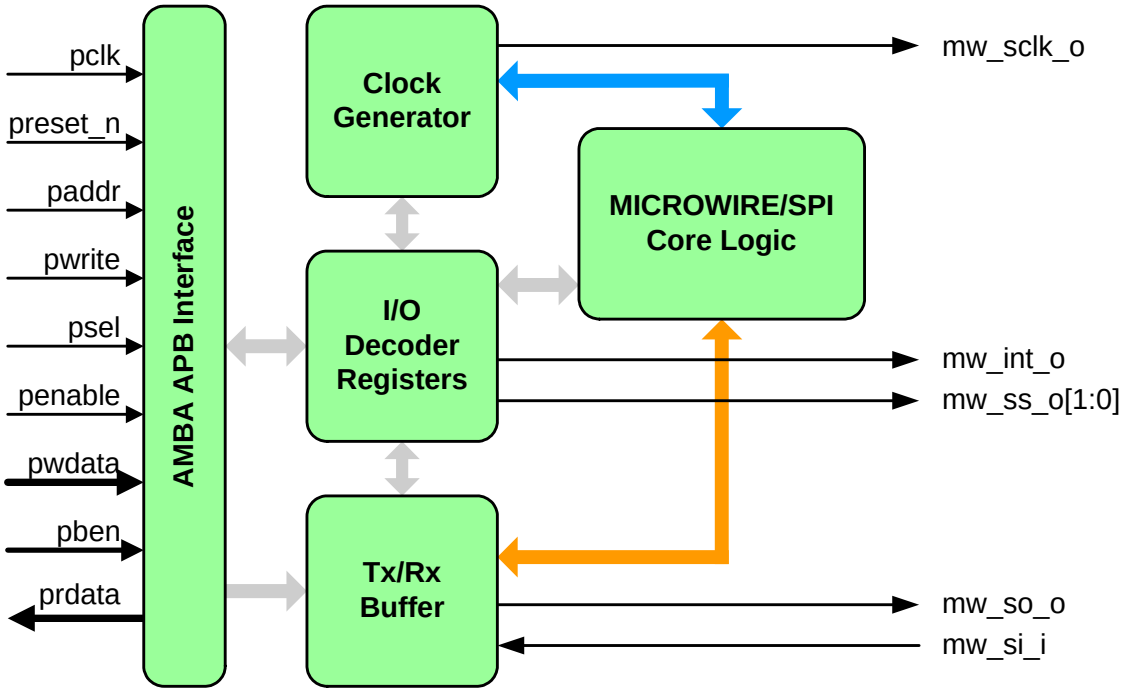
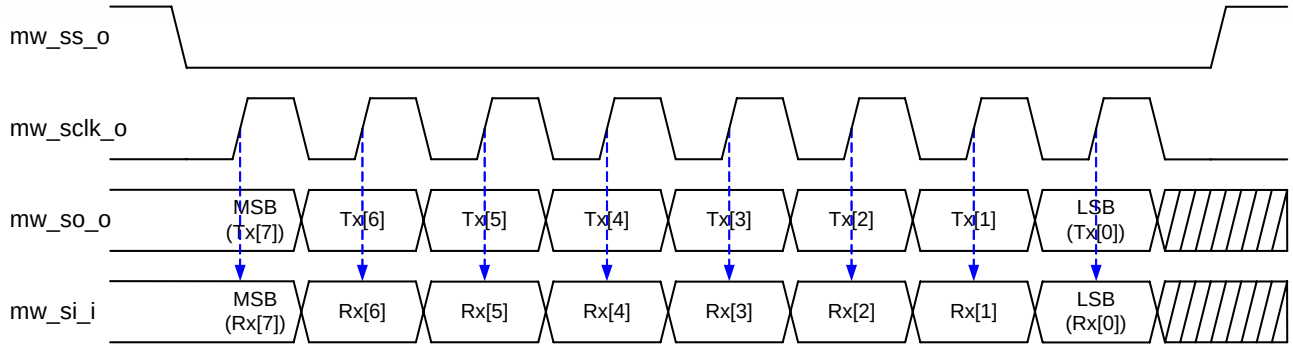


Figure 4.21-1 MICROWIRE/SPI Block Diagram (Master Only)

Pin descriptions:
mw_sclk_o: MICROWIRE/SPI serial clock output pin.
mw_int_o: MICROWIRE/SPI interrupt signal output.
mw_ss_o: MICROWIRE/SPI slave/device select signal output.
mw_so_o: MICROWIRE/SPI serial data output pin (to slave device).
mw_si_i: MICROWIRE/SPI serial data input pin (from slave device).

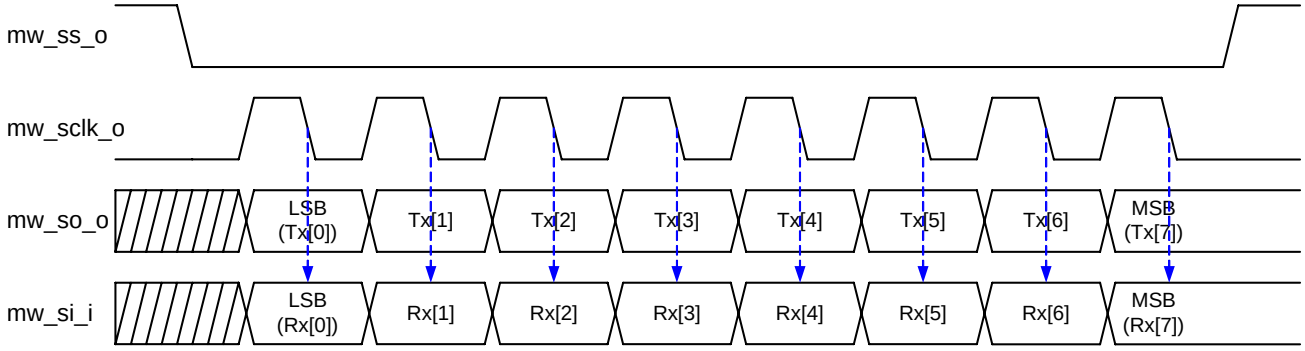
4.21.3 SPI (MICROWIRE) Timing Diagram

The timing diagram of MICROWIRE/SPI is shown as following.



CNTRL[LSB]=0, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
CNTRL[Tx_NEG]=1, CNTRL[Rx_NEG]=0, SSR[SS_LVL]=0

Figure 4.21-2 MICROWIRE/SPI Timing



CNTRL[LSB]=1, CNTRL[Tx_NUM]=0x0, CNTRL[Tx_BIT_LEN]=0x08,
CNTRL[Tx_NEG]=0, CNTRL[Rx_NEG]=1, SSR[SS_LVL]=0

Figure 4.21-3 Alternate Phase SCLK Clock Timing

4.21.4 SPI (MICROWIRE) Programming Example

When using this SPI controller (master only) to access a 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
- 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 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 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 data transfer or set SSR[0] or SSR[1] to 0 to inactivate the device.

4.21.5 SPI1 (MICROWIRE) Serial Interface Control Register Map

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

Register	Offset	R/W	Description	Reset Value
SPI1_BA = 0xFFF8_8000				
SPI1_CNTRL	SPI1_BA + 0x00	R/W	Control and Status Register	0x0000_0004
SPI1_DIVIDER	SPI1_BA + 0x04	R/W	Clock Divider Register	0x0000_0000
SPI1_SSR	SPI1_BA + 0x08	R/W	Slave Select Register	0x0000_0000
Reserved	SPI1_BA + 0x0C	N/A	Reserved	N/A
SPI1_Rx0	SPI1_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI1_Rx1	SPI1_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI1_Rx2	SPI1_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI1_Rx3	SPI1_BA + 0x1C	R	Data Receive Register 3	0x0000_0000
SPI1_Tx0	SPI1_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI1_Tx1	SPI1_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI1_Tx2	SPI1_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI1_Tx3	SPI1_BA + 0x1C	W	Data Transmit Register 3	0x0000_0000

NOTE 1: When software programs CNTRL, the GO_BUSY bit should be written last.

4.21.6 SPI1 (MICROWIRE) Control Register Description

Control and Status Register (CNTRL)

Register	Offset	R/W	Description	Reset Value
SPI1_CNTRL	SPI1_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						**IE**	**IF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SLEEP				**Reserved**	**LSB**	**Tx_NUM**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Tx_BIT_LEN					**Tx_NEG**	**Rx_NEG**	**GO_BUSY**

Bits	Descriptions	
[31:18]	**Reserved**	**Reserved**
[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** These four bits provide the configuration of suspend interval between two successive transmit/receive in a transfer. The default value is 0x0. 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]	**Reserved**	**Reserved**
[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 mw_so_o signal is changed on the rising edge of mw_sclk_o. 1 = The mw_so_o signal is changed on the falling edge of mw_sclk_o.
[1]	**Rx_NEG**	**Receive On Negative Edge**

		0 = The mw_si_i signal is latched on the rising edge of mw_sclk_o. 1 = The mw_si_i signal is latched on the falling edge of mw_sclk_o.
[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 core has no effect.

Divider Register (DIVIDER)

Register	Offset	R/W	Description	Reset Value
SPI1_DIVIDER	SPI1_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 mw_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.

Slave Select Register (SSR)

Register	Offset	R/W	Description	Reset Value
SPI1_SSR	SPI1_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				ASS	SS_LVL	SSR[1:0]	

Bits	Descriptions	
[3]	**ASS**	**Automatic Slave Select** 0 = If this bit is cleared, slave select signals are asserted and de-asserted by setting and clearing related bits in SSR register. 1 = If this bit is set, mw_ss_o signals are generated automatically. It means that device/slave select signal, which is set in 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 (mw_ss_o). 0 = The mw_ss_o slave select signal is active Low. 1 = The mw_ss_o slave select signal is active High.
[1:0]	**SSR**	**Slave Select Register** If SSR[ASS] bit is cleared, writing 1 to any bit location of this field sets the proper mw_ss_o 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 mw_ss_o 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 mw_ss_o 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.

Data Receive Register (RX)

Register	Offset	R/W	Description	Reset Value
SPI1_Rx0	SPI1_BA + 0x10	R	Data Receive Register 0	0x0000_0000
SPI1_Rx1	SPI1_BA + 0x14	R	Data Receive Register 1	0x0000_0000
SPI1_Rx2	SPI1_BA + 0x18	R	Data Receive Register 2	0x0000_0000
SPI1_Rx3	SPI1_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 FFs.

Data Transmit Register (TX)

Register	Offset	R/W	Description	Reset Value
SPI1_Tx0	SPI1_BA + 0x10	W	Data Transmit Register 0	0x0000_0000
SPI1_Tx1	SPI1_BA + 0x14	W	Data Transmit Register 1	0x0000_0000
SPI1_Tx2	SPI1_BA + 0x18	W	Data Transmit Register 2	0x0000_0000
SPI1_Tx3	SPI1_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 means 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.

4.22 Timer and Watchdog Timer

4.22.1 General Timer Controller

The timer allows user to easily implement a counting scheme for use. The timer can perform functions like frequency measurement, event counting, interval measurement, pulse generation, delay timing, and so on. The timer possesses features such as adjustable resolution, programmable counting period … See descriptions below for more 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

- Compliant with the AMBA APB
- One channel with a 26-bit down counter and an interrupt request.
- Maximum uninterrupted time = (1 / 15 MHz) * (256) * (2^26 - 1), if TCLK = 15 MHz

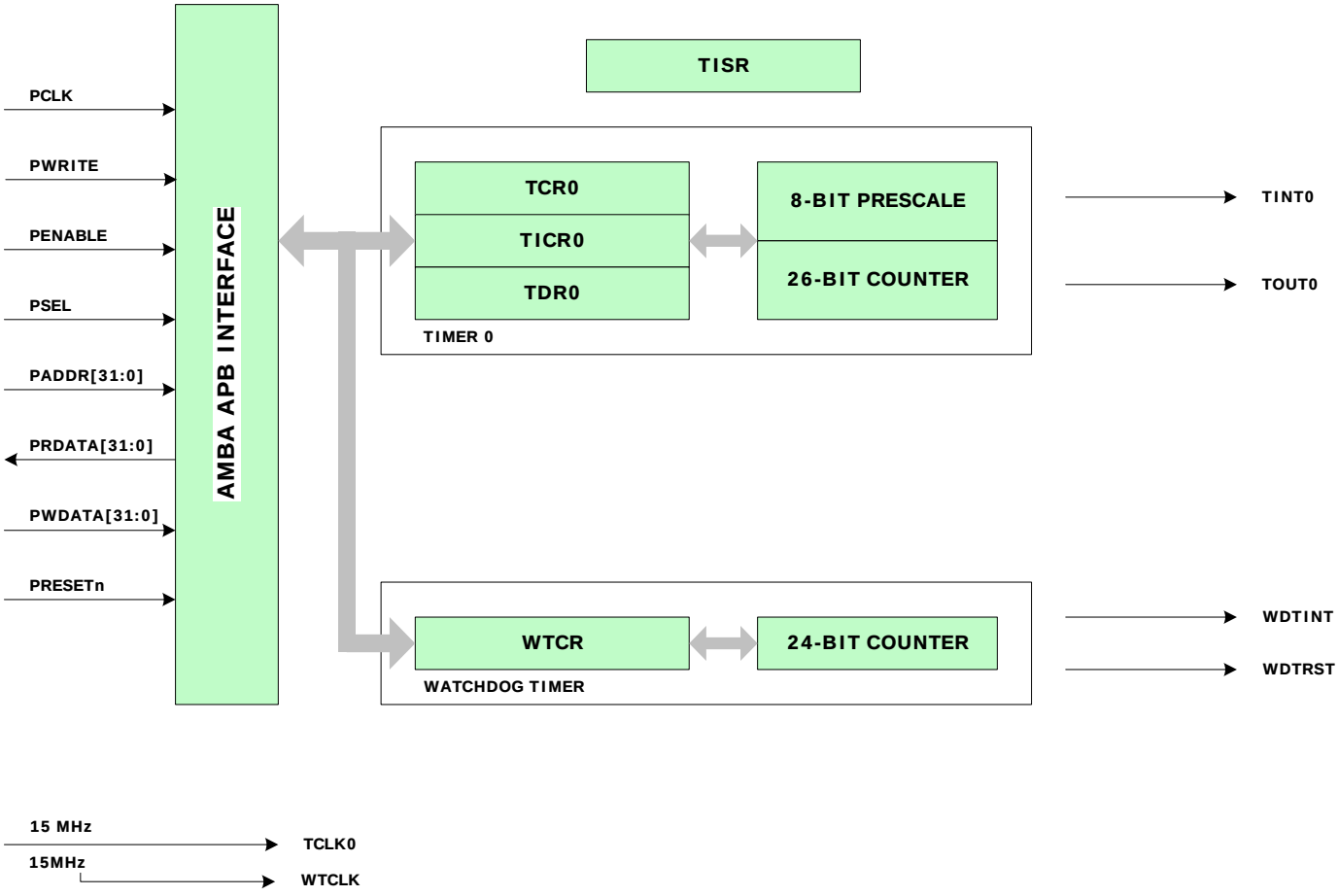


Figure 4.22-1 Timer Block Diagram

4.22.2 Watchdog Timer

The purpose of watchdog timer is to perform a system restart after the software running into a problem. This recovers system from crash for some reasons. It is a free running timer with programmable time-out intervals. When the specified time internal 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 reset is activated after 1024 WDT clocks. 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. The watchdog timer will start counting and time-out after a specified period of time. The time-out 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 15 WDT clock cycles long. When used as a simple timer, the reset function is disabled. 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.

4.22.3 Timer 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
TMR_BA = 0xFFF8_1000				
TCR0	TMR_BA+0x000	**R/W**	Timer Control Register 0	0x0000_0005
TICR0	TMR_BA+0x008	**R/W**	Timer Initial Control Register 0	0x0000_0000
TDR0	TMR_BA+0x010	**R**	Timer Data Register 0	0x0000_0000
TISR	TMR_BA+0x018	**R/C**	Timer Interrupt Status Register	0x0000_0000
WTCR	TMR_BA+0x01C	**R/W**	Watchdog Timer Control Register	0x0000_0400

4.22.4 Timer Control Register Description

Timer Control Register (TCR)

Register	Address	R/W/C	Description	Reset Value
TCR0	TMR_BA+0x000	R/W	Timer Control Register 0	0x0000_0005

31	30	29	28	27	26	25	24
nDBGACK_EN	**CE**	**IE**	**MODE**		**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							

nDBGACK_EN [31]: ICE debug mode acknowledge enable
0 = When DBGACK is high, the timer clock will be held
1 = No matter what DBGACK is high or not, the timer clock will not be held

CE [30]: Counter Enable
0 = Stops counting
1 = Starts counting

IE [29]: Interrupt Enable
0 = Disables timer interrupt
1 = Enables timer interrupt. If timer interrupt is enabled, the timer asserts its interrupt signal when the associated counter counts down to zero.

MODE [28:27]: Timer Operating Mode

MODE [28:27]	Timer Operating Mode
00	The timer is operating in the one-shot mode. The associated interrupt signal is generated once (if IE is enabled) and CE is automatically cleared then.
01	The timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if IE is enabled).
10	Reserved for further use
11	Reserved for further use

PRESCALE [7:0]
Clock input is divided by PRESCALE + 1 before it is fed to the counter (here PRESCALE is considered as a decimal number). If PRESCALE = 0, then there is no scaling.

Timer Initial Count Register (TICR)

Register	Address	R/W/C	Description	Reset Value
TICR0	TMR_BA+0x008	R/W	Timer Initial Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED						TIC [25:24]	
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]							

TIC [25:0]: Timer Initial Count
This is a 26-bit value representing the initial count. Timer will reload this value whenever the counter is

counted down to zero.

Timer Data Register (TDR)

Register	Address	R/W/C	Description	Reset Value
TDR0	TMR_BA+0x010	R	Timer Data Register 0	0x0000_0000

31	30	29	28	27	26	25	24
RESERVED						TDR [25:24]	
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]							

TDR [25:0]: Timer Data Register
The current count is recorded in this 26-bit register.

Timer Interrupt Status Register (TISR)

Register	Address	R/W/C	Description	Reset Value
TISR	TMR_BA+0x018	R/C	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							TIF0

TIF1 [0]: Timer Interrupt Flag 0
It indicates the interrupt status of the timer 0.
0 = It indicates that the timer 0 does not count down to zero yet. Software can reset this bit after the timer interrupt 0 has occurred.
1 = It indicates that the counter of timer 0 is decreased to zero, if the IE bit (TICR[29] bit) is set to “1”. Otherwise, if the IE bit (TICR[29] bit) is set to “0”, even the counter of timer is decreased to zero, the TIF[0] will not be set to “1”.
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/C	Description	Reset Value
WTCR	TMR_BA+0x01C	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

WTCLK [10] : 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).

nDBGACK_EN [9]: ICE debug mode acknowledge enable
0 = When DBGACK is high, the timer clock will be held
1 = No matter what DBGACK is high or not, the timer clock will not be held

WTTME [8]: Watchdog Timer Test Mode Enable
For reasons of efficiency, the 24-bit up counter within the watchdog timer is considered as two independent 12-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 13-bit counter overflows, a watchdog timer interrupt is generated.
0 = Put the watchdog time in the normal operating mode
1 = Put the watchdog timer in the test mode

WTE [7]: Watchdog Timer Enable
0 = Disable the watchdog timer
1 = Enable the watchdog timer

WTIE [6]: Watchdog Timer Interrupt Enable
0 = Disable the watchdog timer interrupt
1 = Enable the watchdog timer interrupt

WTIS [5:4]: Watchdog Timer Interval Select
These two bits select the interval for the watchdog timer. No matter which interval is chosen, the reset time-out is always occurred 1024 clocks later than the interrupt time-out.

WTIS	Interrupt Timeout	Reset 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.

WTIF [3]: Watchdog Timer Interrupt Flag
If the watchdog interrupt is enabled, then the hardware will set this bit to indicate that the watchdog interrupt has occurred. If the watchdog interrupt is not enabled, then this bit indicates that a time-out period has elapsed.

0 = Watchdog timer interrupt does not occur
1 = Watchdog timer interrupt occurs

WTRF [2]: 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

WTRE [1]: 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

WTR [0]: Watchdog Timer Reset
This bit brings the watchdog timer into a known state. It helps reset the watchdog timer before a time-out situation occurring. Failing to set **WTR** before time-out will initiates an interrupt if **WTIE** is set. If **WTRE** is set, a watchdog timer reset will be generated 1024 clocks after time-out. This bit is self-clearing.
0 = No operation
1 = Reset the contents of the watchdog timer

4.23 RTC

4.23.1 General RTC Controller

A built in RTC is designed to generate the periodic interrupt signal. The clock source comes from the 32768 Hz clock source. The periodic can be 0.25/0.5/1/2/4/8 sec.

1. RTC part has six pins, Vdd18, PWRSW, X32K, EX32K, WAKEUP, RGND.
2. RTC typical current, 14~15uA, maximum current, 20uA
3. “Power on” has “time out” function to avoid current always existence.

When battery low at wake up moment, system will hang up; H/W provides “time out” function to turn off system.
If system wake up, and work smoothly, “time out” function will be disabled by user.

4. When battery is inserted, the default of switch is on.
5. Provide “protection function” to disable switch.
   When match special SFR value, VA91 can turn off the switch.
6. Level transition provides two choices
   0.9V + 0.2V
   0.9V - 0.2V

Important PINs information:

- **PWRSW**: if it presents as low, all system turn on. (**PWRSW**, output pin, open drain, need pull up)
- **WAKEUP**: this pin is within RTC parts. It will make PWRSW pin low when WAKEUP pin is low. Should be pull up to RTC power.

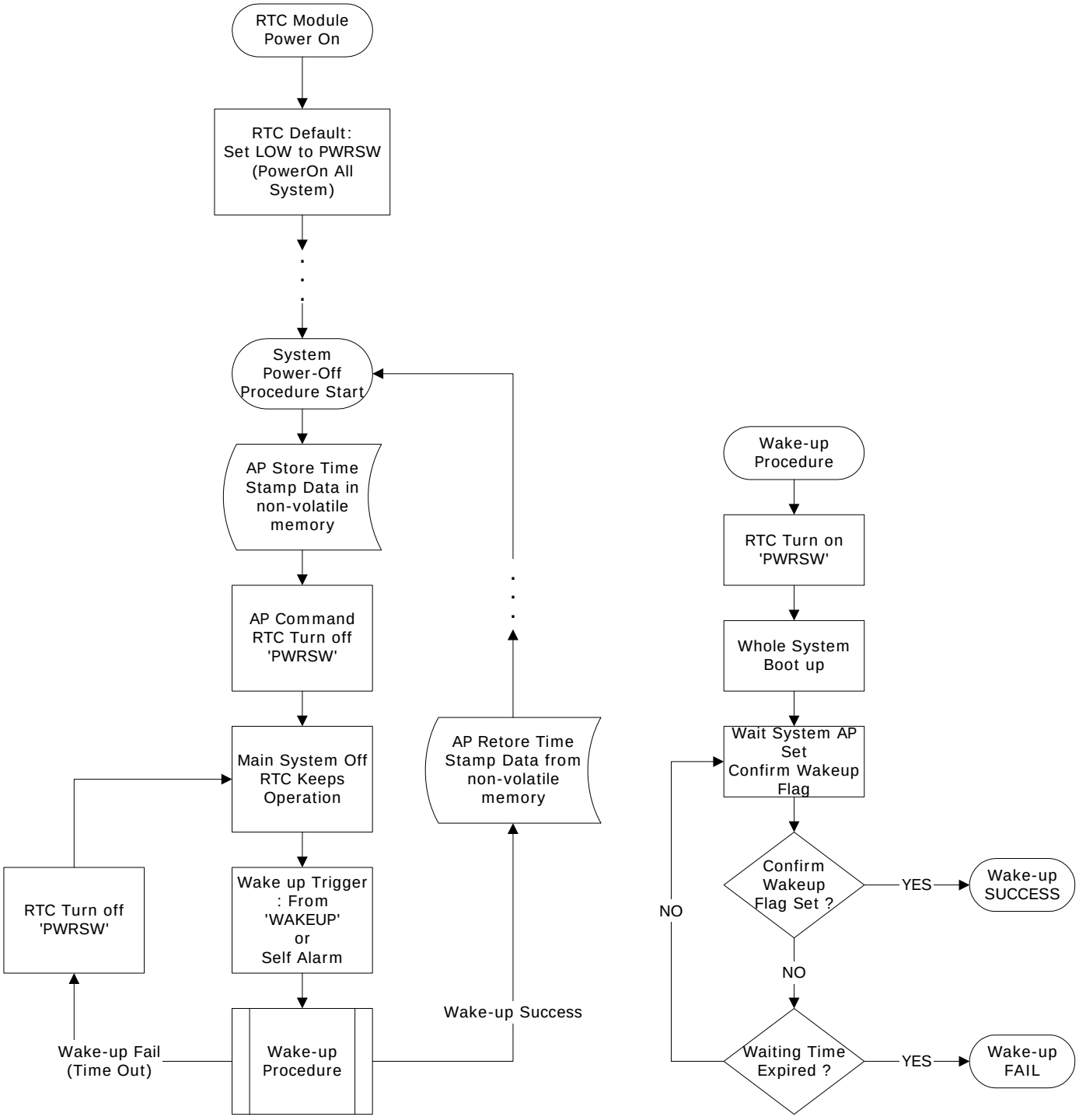
System operating procedure:

Step1: Stores the time, year/month/day/hour/minute/sec, into non-volatile memory, and programmer sets **PWR_OFF** bit (**RTCCTL**[16]) from low to high to make **PWRSW** pin high to power off system except RTC part.

Step2: RTC keep working, and increases the RTC counter under sleep mode.

Step3: When trigger occurred, **WAKEUP** low, **PWRSW** low (H/W set automatically), all system get back the power which controlled by PWRSW. System wakes up and reads out the time (year/month/day/hour/minute/sec), add the RTC counter into current time. System need to clear the **PWR_OFF** bit “low” to indicate the power off condition recovered.

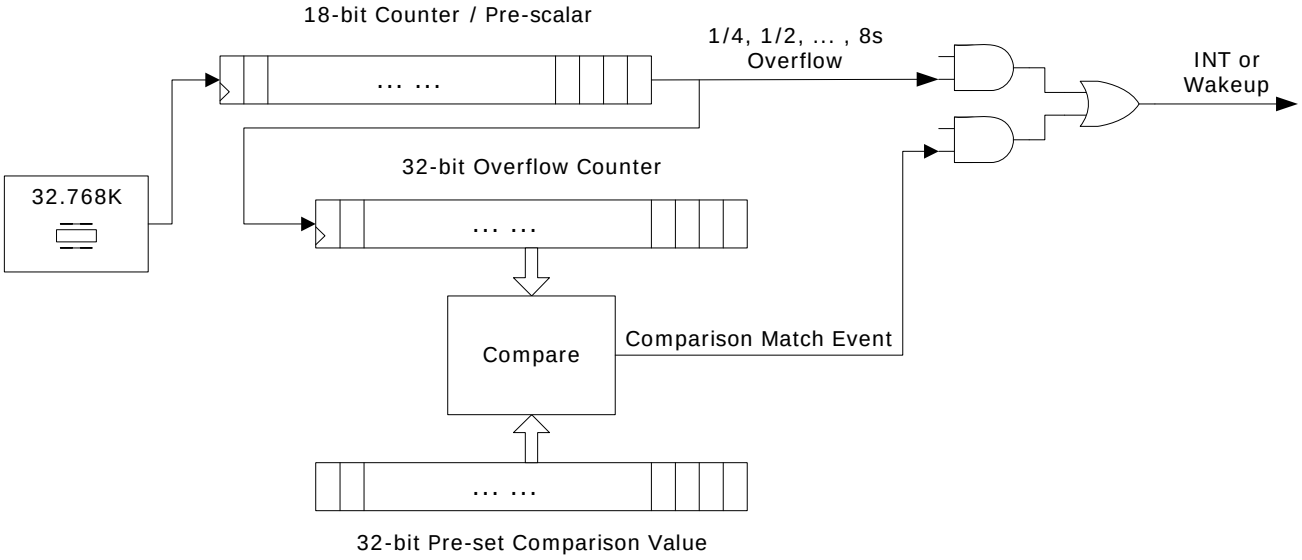
Following is the flow diagram for the RTC wakeup and wakeup time-out flow:



Note: The Time Out period is equal to the time of Self Alarm which is set by AP.

RTC provide “Alarm” function

1. An alarm time for waking up the system is set through AP
2. The AP converts the time to the RTC counter format and wrote to the alarm SFR in RTC
3. Alarm function enabled, the system continues to work or goes down afterward.
4. A comparator in RTC is used for comparing the alarm SFR and timer counter value. Once they are matched, an interrupt is then issued and it wakes up the system if the system is in sleep mode.
5. The system then processes the alarm event accordingly.



INT or Alarm Function

4.23.2 RTC 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
RTC_BA = 0xFFF8_7000				
RTCCTL	RTC_BA+0x000	**R/W**	RTC Control Register	0x0000_0000
RTCINTCTL	RTC_BA+0x004	**R/W**	RTC Interrupt Control Registers	0x0000_0000
RTCCNTS	RTC_BA+0x008	**R**	RTC Counter Status Register	0x0000_0000
RTCOFCNT	RTC_BA+0x00C	**R/W**	RTC Overflow Counter	0x0000_0000
RTCINIR	RTC_BA+0x010	**R/W**	RTC Initiation Register	0x0000_0000
RTCAER	RTC_BA+0x014	**R/W**	RTC Access Enable Register	0x0000_0000
RTCALARM	RTC_BA+0x018	**R/W**	RTC Alarm Control Register	0x0000_55aa

4.23.3 RTC Control Register Description

RTC Control Register (RTCCTL)

Register	Address	R/W	Description	Reset Value
RTCCTL	RTC_BA+0x000	R/W	RTC Setting Control Register	0x0000_0000

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

Bits	Descriptions		Default
[31:17]	**RESERVED**	RESERVED	0x0
[16]	**PWR_OFF**	**PWRSW control** PWRSW will change to high state when PWR_OFF value change from 0 to 1. Note 1: In RTC there is a register mapping to PWR_OFF. It may be different as PWR_OFF if VA91 ever power off. Note 2: Below conditions will make PWRSW low ■ Set PWR_OFF bit to 0 ■ WAKEUP pin low ■ RTCOFCNT matching with RTCALARM will inverts PWRSW state when RTC PWR_OFF bit is 1.	0x0
[15:3]	**RESERVED**	RESERVED	0x0
[2:0]	**RTCPSEL**	**RTC Period Selection divider** **RTCSET [2:0] (RTCPSEL)** / **Divided by** 0 0 0 — 0.25 sec 0 0 1 — 0.5 sec 0 1 0 — 1 sec 0 1 1 — 2 sec 1 0 0 — 4 sec 1 0 1 — 8 sec 1 1 0 — Reserved 1 1 1 — Reserved	0x0

RTC Interrupt Control Register (RTCINTCTL)

Register	Address	R/W	Description	Reset Value
RTCINTCTL	RTC_BA+0x004	R/W	RTC Interrupt Control Register	0x0000_0000

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

Bits	Descriptions		Default
[31:18]	**RESERVED**	RESERVED	0x0
[17]	**AINT_EN**	**Alarm Interrupt Enable** 0 = RTC alarm interrupt will not enable 1 = Enable RTC alarm interrupt	0x0
[16]	**OINT_EN**	**Timer Over Flow Interrupt Enable** 0 = RTC overflow flag will not enable 1 = Enable RTC overflow interrupt	0x0
[15:2]	**RESERVED**	RESERVED	0x0
[1]	**RIIR_AI**	**RTC Alarm Interrupt Flag** When the overflow counter meet the value of ALARM [offset = 0x18], the bit will set to 1. Write 1 to clear it.	0x55aa
[0]	**RIIR_OI**	**RTC Overflow Flag** When the up counter overflow and the bit will set to 1. Write 1 to clear it.	0x0

RTC Counter Status Register (RTCCNTS)

Register	Address	R/W	Description	Reset Value
RTCCNTS	RTC_BA+0x008	R	RTC Counter Status	0x0000_0000

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

RTCNTS[7:0]

Bits	Descriptions		Default
[31:18]	**RESERVED**	RESERVED	0x0
[17:0]	**RTCCNTS**	**RTC counter status** It shows that the internal RTC counter value.	0x0

RTC Overflow Counter Register (RTCOFCNT)

Register	Address	R/W	Description	Reset Value
RTCOFCNT	RTC_BA+0x00C	R/W	RTC Overflow Counter	0x0000_0000

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

Bits	Descriptions		Default
[31:0]	**RTCOFCNT**	**RTC Overflow Counter Register** The counter will add 1 when overflow signal happens The register will be cleared in these condition ➢ Reset ➢ To Write 32'h0000_0000 to clear the register ➢ RTC_EN=1'b0 Note: Can not read this register when WAKEUP pin is low.	0x0

RTC Initiation Register (RTCINIR)

Register	Address	R/W	Description	Reset Value
RTCINIR	RTC_BA+0x010	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
RTCINIR [31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**

RTCINIR [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RTCINIR [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTCINIR [7:4]				RTCINIR [3:1]			INIR/Active

Bits	Descriptions		Default
[31:0]	**RTCINIR**	**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	0x0
[0]	**Active**	**RTC Active Status** (Read only) 0: RTC is at reset state 1: RTC is at normal active state.	0x0

RTC Access Enable (RTCAER)

Register	Address	R/W	Description	Reset Value
RTCAER	RTC_BA+0x014	R/W	RTC Access Enable	0x0000_0000

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

Bits	Descriptions	
[16]	**ENF**	**RTC Register Access Enable Flag** (Read only) 1: RTC register read/write enable 0: RTC register read/write disable This bit will be set after AER[15:0] register is load a 0xA965, and be clear automatically in a long time or AER[15:0] is not 0xA965.
[15:0]	**AER**	**RTC Register Access Enable Password** (Write only) 0xA965: access enable

		Others: access disable

RTC Alarm Register (RTCALARM)

Register	Address	R/W	Description	Reset Value
RTCALARM	RTC_BA+0x018	R/W	RTC Alarm Register	0x0000_55aa

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

Bits	Descriptions		Default
[31:0]	**RTCALARM**	**RTC Alarm Register (32-bit Pre-set comparison value)** The alarm register is used to compare the overflow counter. If the overflow counter reach to value of alarm register, the RTC will wake up in power down mode or output interrupt when the interrupt enable bit RTCCTL[17] be set.	0x55aa

4.24 UART

4.24.1 Overview and Features

The **Universal Asynchronous Receiver/Transmitter (UART)** performs a serial-to-parallel conversion on data characters received from the peripheral such as MODEM, and a parallel-to-serial conversion on data characters received from the CPU. There are five types of interrupts, i.e., **line status interrupt, transmitter FIFO empty interrupt, receiver threshold level reaching interrupt, time out interrupt, and MODEM status interrupt.** One 16-byte transmitter FIFO **(TX_FIFO)** and one 16-byte (plus 3-bit of error data per byte) receiver FIFO **(RX_FIFO)** has been built in to reduce the number of interrupts presented to the CPU. The CPU can completely 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 any error conditions (parity, overrun, framing, or break interrupt) found. 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 needed. The equation is

BaudOut = crystal clock / 16 * [Divisor + 2].

- The UART includes the following features:
- Transmitter and receiver are buffered with a 16-byte FIFO each to reduce the number of interrupts presented to the CPU.
- Full set of MODEM control functions (CTS, RTS, DSR, DTR, RI and DCD)
- 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
- Line break generation and detection
- False start bit detection
- Full prioritized interrupt system controls
- Loop back mode for internal diagnostic testing

4.24.2 Functional Block Description

TX_FIFO
The transmitter is buffered with a 16-byte FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO
The receiver is buffered with a 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

BAUD RATE GENERATOR
Dividing the external clock by the divisor to get the desired internal clock

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 causes the MODEM status interrupt (CTS# or DSR# or RI# or DCD#).

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.

4.24.3 Finite State Machine

4.24.3.1 Transmitter

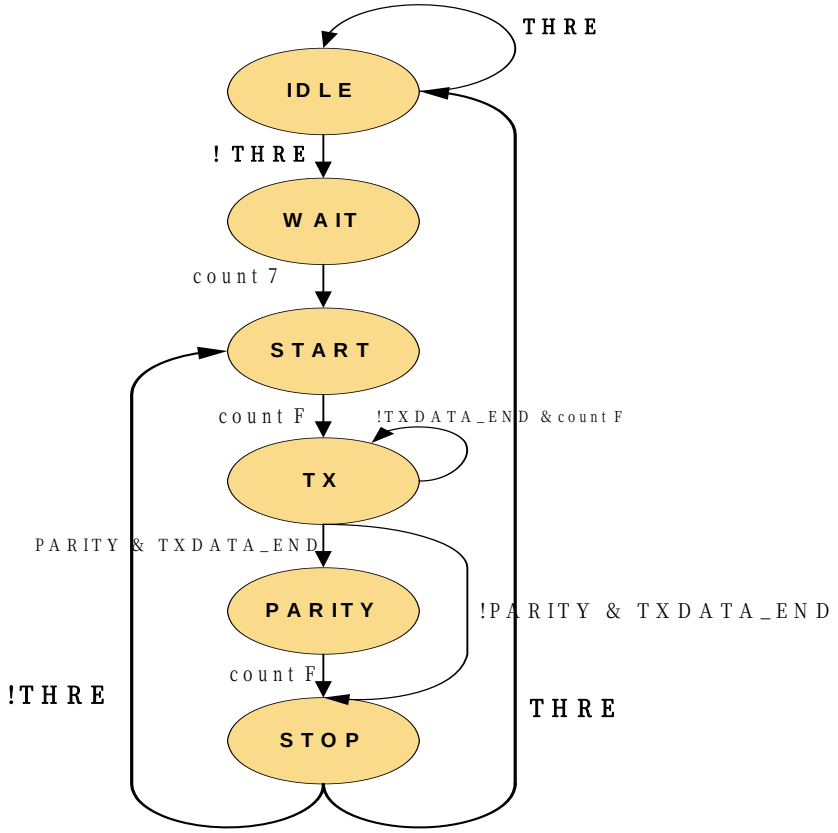


Figure 4.24-1 Transmitter finite state machine

STATE DEFINITION

IDLE
The transmitter has no data to transmit.
WAIT
When the transmitter's FIFO is not empty.
START
The transmitter transmits the start bit.
TX
The transmitter transmits the data.
PARITY
The transmitter transmits the parity bit.
STOP
The transmitter transmits the stop bit.

SIGNAL DESCRIPTION

THRE
The transmitter holding register is empty.
Count7
The counter of clock equals to 7.
CountF
The counter of clock equals to 15.
TXDATA_END
The data part transfer is finished.
PARITY
The transfer includes the parity bit.

NOTE:
The format of the transfer is as following:
One transfer = start bit + data + parity bit (if dedicated) + stop bit

4.24.3.2 Receiver

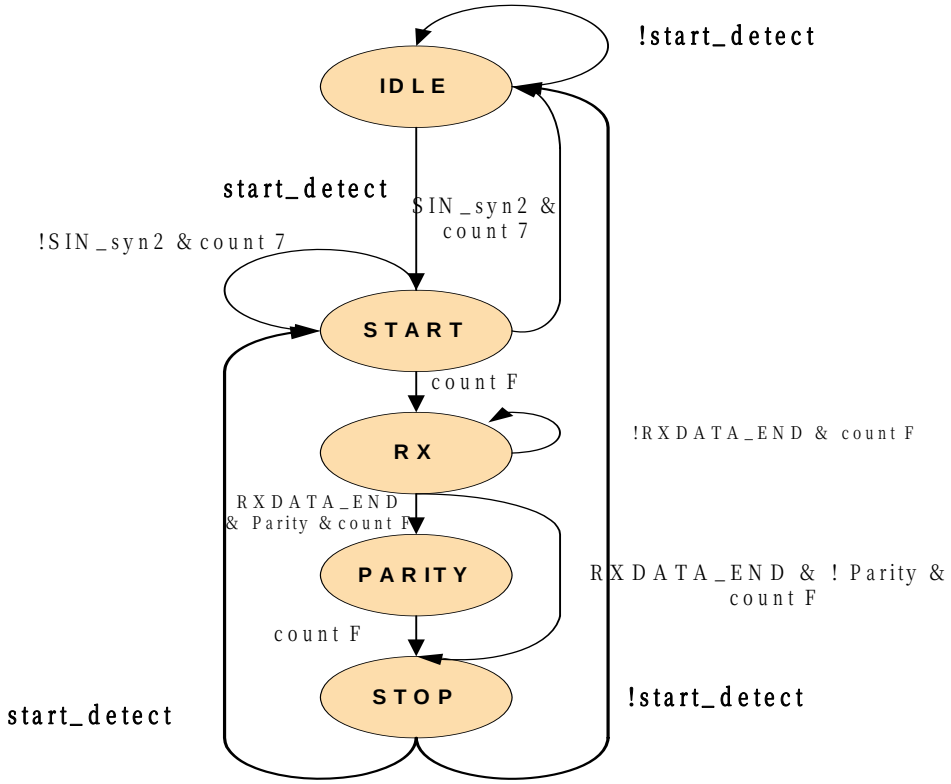


Figure 4.24-2 Receiver finite state machine

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

4.24.4 UART 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
UART_BA = 0xFFF8_0000				
RBR	UART_BA+0x000	R	Receive Buffer Register (DLAB = 0)	Undefined
THR	UART_BA+0x000	W	Transmit Holding Register (DLAB = 0)	Undefined
IER	UART_BA+0x004	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000
DLL	UART_BA+0x000	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000
DLM	UART_BA+0x004	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000
IIR	UART_BA+0x008	R	Interrupt Identification Register	0x8181_8181
FCR	UART_BA+0x008	W	FIFO Control Register	Undefined
LCR	UART_BA+0x00C	R/W	Line Control Register	0x0000_0000
MCR	UART_BA+0x010	R/W	Modem Control Register	0x0000_0000
LSR	UART_BA+0x014	R	Line Status Register	0x6060_6060
MSR	UART_BA+0x018	R	MODEM Status Register	0x0000_0000
TOR	UART_BA+0x01C	R/W	Time Out Register	0x0000_0000

4.24.5 UART Control Register Description

Receive Buffer Register (RBR)

Register	Address	R/W	Description	Reset Value
RBR	UART_BA+0x000	R	Receive Buffer Register (DLAB = 0)	Undefined

7	6	5	4	3	2	1	0
8-bit Received Data							

8-bit Received Data [7:0]
By reading this register, the UART will return an 8-bit data received from SIN pin (LSB first).

Transmit Holding Register (THR)

Register	Address	R/W	Description	Reset Value
THR	UART_BA+0x000	W	Transmit Holding Register (DLAB = 0)	Undefined

7	6	5	4	3	2	1	0
8-bit Transmitted Data							

8-bit Transmitted Data [7:0]
By writing to this register, the UART will send out an 8-bit data through the SOUT pin (LSB first).

Interrupt Enable Register (IER)

Register	Address	R/W	Description	Reset Value
IER	UART_BA+0x004	R/W	Interrupt Enable Register (DLAB = 0)	0x0000_0000

7	6	5	4	3	2	1	0
RESERVED			nDBGACK_EN	MSIE	RLSIE	THREIE	RDAIE

nDBGACK_EN [4]: 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

MSIE [3]: MODEM Status Interrupt (Irpt_MOS) Enable
0 = Mask off Irpt_MOS
1 = Enable Irpt_MOS

RLSIE [2]: Receive Line Status Interrupt (Irpt_RLS) Enable
0 = Mask off Irpt_RLS
1 = Enable Irpt_RLS

THREIE [1]: Transmit Holding Register Empty Interrupt (Irpt_THRE) Enable
0 = Mask off Irpt_THRE
1 = Enable Irpt_THRE

RDAIE [0]: Receive Data Available Interrupt (Irpt_RDA) Enable and
Time-out Interrupt **(Irpt_TOUT)** Enable
0 = Mask off Irpt_RDA and Irpt_TOUT

1 = Enable Irpt_RDA and Irpt_TOUT

Divider Latch (Low Byte) Register (DLL)

Register	Address	R/W	Description	Reset Value
DLL	UART_BA+0x000	R/W	Divisor Latch Register (LS) (DLAB = 1)	0x0000_0000

7	6	5	4	3	2	1	0
Baud Rate Divider (Low Byte)							

Baud Rate Divisor (Low Byte) [7:0]
The low byte of the baud rate divider

Divisor Latch (High Byte) Register (DLM)

Register	Address	R/W	Description	Reset Value
DLM	UART_BA+0x004	R/W	Divisor Latch Register (MS) (DLAB = 1)	0x0000_0000

7	6	5	4	3	2	1	0
Baud Rate Divider (High Byte)							

Baud Rate Divisor (High Byte) [7:0]
The high byte of the baud rate divider

This 16-bit divider {DLM, DLL} is used to determine the baud rate as follows

Baud Rate = Crystal Clock / {16 * [Divisor + 2]}

Note: This definition is different from 16550

Interrupt Identification Register (IIR)

Register	Address	R/W	Description	Reset Value
IIR	UART_BA+0x008	R	Interrupt Identification Register	0x8181_8181

7	6	5	4	3	2	1	0
FMES	RFTLS		DMS	IID			NIP

FMES [7]: FIFO Mode Enable Status
This bit indicates whether the FIFO mode is enabled or not. Since the FIFO mode is always enable, this bit always shows the logical 1 when CPU is reading this register.

RFTLS [6:5]: RX FIFO Threshold Level Status
These bits show the current setting of receiver FIFO threshold level (RTHO). The meaning of RTHO is defined in the following FCR description.

DMS [4]: DMA Mode Select
The DMA function is not implemented in this version. When reading IIR, the DMS is always returned 0.

IID [3:1]: Interrupt Identification
The IID together with NIP indicates the current interrupt request from UART.

NIP [0]: No Interrupt Pending
There is no pending interrupt.

IIR [3:0]	Priority	Interrupt Type	Interrupt Source	Interrupt Reset control
- - - 1	--	None	None	--
0110	Highest	Receiver Line Status (Irpt_RLS)	Overrun error, parity error, framing error, or break interrupt	Reading the LSR
0100	Second	Received Data Available (Irpt_RDA)	Receiver FIFO threshold level is reached	Receiver FIFO drops below the threshold level
1100	Second	Receiver FIFO Time-out (Irpt_TOUT)	Receiver FIFO is non-empty and no activities are occurred in the receiver FIFO during the TOR defined time duration	Reading the RBR
0010	Third	Transmitter Holing Register Empty (Irpt_THRE)	Transmitter holding register empty	Reading the IIR (if source of interrupt is Irpt_THRE) or writing into the THR
0000	Fourth	MODEM Status (Irpt_MOS)	The CTS, DSR, or DCD bits are changing state or the RI bit is changing from high to low.	Reading the MSR

Note: These definitions of bit 7, bit 6, bit 5, bit 4 are different from the 16550.

Table 4.24-1 Interrupt Control Functions

FIFO Control Register (FCR)

Register	Address	R/W	Description	Reset Value
FCR	UART_BA+0x008	W	FIFO Control Register	Undefined

7	6	5	4	3	2	1	0
RFITL		RESERVED		DMS	TFR	RFR	FME

RFITL [7:6]: RX FIFO Interrupt (Irpt_RDA) Trigger Level

RFITL [7:6]	Irpt_RDA Trigger Level (Bytes)
00	01
01	04
10	08
11	14

DMS [3]: DMA Mode Select
The DMA function is not implemented in this version.

TFR [2]: TX FIFO Reset
Setting this bit will generate an OSC cycle reset pulse to reset TX FIFO. The TX FIFO becomes empty (TX pointer is reset to 0) after such reset. This bit is returned to 0 automatically after the reset pulse is generated.

RFR [1]: RX FIFO Reset
Setting this bit will generate an OSC cycle reset pulse to reset RX FIFO. The RX FIFO becomes empty (RX pointer is reset to 0) after such reset. This bit is returned to 0 automatically after the reset pulse is generated.

FME [0]: FIFO Mode Enable
Because UART is always operating in the FIFO mode, writing this bit has no effect while reading always gets logical one. This bit must be 1 when other FCR bits are written to; otherwise, they will not be programmed.

Line Control Register (LCR)

Register	Address	R/W	Description	Reset Value
LCR	UART_BA+0x00C	R/W	Line Control Register	0x0000_0000

7	6	5	4	3	2	1	0
DLAB	BCB	SPE	EPE	PBE	NSB	WLS	

DLAB [7]: Divider Latch Access Bit
0 = It is used to access RBR, THR or IER.
1 = It is used to access Divisor Latch Registers {DLL, DLM}.

BCB [6]: 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.

SPE [5]: 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.

EPE [4]: 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.

PBE [3]: 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.

NSB [2]: 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.

WLS [1:0]: Word Length Select

WLS[1:0]	Character length
00	5 bits
01	6 bits
10	7 bits
11	8 bits

Modem Control Register (MCR)

Register	Address	R/W	Description	Reset Value
MCR	UART_BA+0x010	R/W	Modem Control Register	0x0000_0000

7	6	5	4	3	2	1	0
RESERVED			LBME	OUT2#	OUT1#	RTS#	DTR#

LBME [4]: Loop-back Mode Enable
0 = Disable
1 = When the loop-back mode is enabled, the following signals are connected internally:
SOUT connected to SIN and SOUT pin fixed at logic 1
DTR# connected to DSR# and DTR# pin fixed at logic 1
RTS# connected to CTS# and RTS# pin fixed at logic 1
OUT1# connected to RI# and OUT1# pin fixed at logic 1
OUT2# connected to DCD# and OUT2# pin fixed logic 1

OUT2#[3]: Complement version of OUT2# (user-designated output) signal

OUT1#[2]: Complement version of OUT1# (user-designated output) signal

RTS#[1]: Complement version of RTS# (Request-To-Send) signal

DTR#[0]: Complement version of DTR# (Data-Terminal-Ready) signal

Writing 0x00 to MCR, the DTR#, RTS#, nOUT1# and OUT2# bit are set to logic 1's;
Writing 0x0f to MCR, the DTR#, RTS#, nOUT1# and OUT2# bit are reset to logic 0's.

Line Status Control Register (LSR)

Register	Address	R/W	Description	Reset Value
LSR	UART_BA+0x014	R	Line Status Register	0x6060_6060

7	6	5	4	3	2	1	0
ERR_RX	TE	THRE	BII	FEI	PEI	OEI	RFDR

ERR_RX [7]: RX FIFO Error
0 = RX FIFO works normally
1 = There is at least one parity error (PE), framing error (FE), or break indication (BI) in the FIFO. ERR_RX is cleared when CPU reads the LSR and if there are no subsequent errors in the RX FIFO.

TE [6]: Transmitter Empty
0 = Either Transmitter Holding Register (**THR** - TX FIFO) or Transmitter Shift Register (**TSR**) are not empty.
1 = Both THR and TSR are empty.

THRE [5]: Transmitter Holding Register Empty
0 = THR is not empty.
1 = THR is empty.
THRE is set when the last data word of TX FIFO is transferred to Transmitter Shift Register (TSR). The CPU resets this bit when the THR (or TX FIFO) is loaded. This bit also causes the UART to issue an interrupt (Irpt_THRE) to the CPU when IER [1]=1.

BII [4]: 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 reads the contents of the LSR**.**

FEI [3]: 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 reads the contents of the LSR**.**

PEI [2]: 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 reads the contents of the LSR**.**

OEI [1]: Overrun Error Indicator
An overrun error will occur only after the RX FIFO is full and the next character has been completely received in the shift register. The character in the shift register is overwritten, but it is not transferred to the RX FIFO. OE is indicated to the CPU as soon as it happens and is reset whenever the CPU reads the contents of the LSR.

RFDR [0]: RX FIFO Data Ready
0 = RX FIFO is empty
1 = RX FIFO contains at least 1 received data word.

LSR [4:2] (BII, FEI, PEI) are revealed to the CPU when its associated character is at the top of the RX FIFO. These three error indicators are reset whenever the CPU reads the contents of the LSR.

LSR [4:1] (BII, FEI, PEI, OEI) are the error conditions that produce a "receiver line status interrupt" (Irpt_RLS) when IER [2]=1. Reading LSR clears Irpt_RLS. Writing LSR is a null operation (not suggested).

Modem Status Register (MSR)

Register	Address	R/W	Description	Reset Value
MSR	UART_BA+0x018	R	MODEM Status Register	0x0000_0000

7	6	5	4	3	2	1	0
DCD#	**RI#**	**DSR#**	**CTS#**	**DDCD**	**TERI**	**DDSR**	**DCTS**

DCD#[7]: Complement version of Data Carrier Detect (nDCD#) input

RI#[6]: Complement version of ring indicator (RI#) input

DSR#[5]: Complement version of data set ready (DSR#) input

CTS#[4]: Complement version of clear to send (CTS#) input

DDCD [3]: DCD# State Change
This bit is set whenever DCD# input has changed state, and it will be reset if the CPU reads the MSR.

TERI [2]: Tailing Edge of RI#
This bit is set whenever RI# input has changed from high to low, and it will be reset if the CPU reads the MSR.

DDSR [1]: DSR# State Change
This bit is set whenever DSR# input has changed state, and it will be reset if the CPU reads the MSR.

DCTS [0]: CTS# State Change
This bit is set whenever CTS# input has changed state, and it will be reset if the CPU reads the MSR.

Whenever any of MSR [3:0] is set to logic 1, a Modem Status Interrupt is generated if IER[3]=1. Writing MSR is a null operation (not suggested).

Time Out Register (TOR)

Register	Address	R/W	Description	Reset Value
TOR	UART_BA+0x01C	R/W	Time Out Register	0x0000_0000

7	6	5	4	3	2	1	0
TOIE	TOIC						

TOIE [7]: Time Out Interrupt Enable
The feature of receiver time out interrupt is enabled only when TOR [7] = IER[0] = 1.

TOIC [6:0]: 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 (Irpt_TOUT) is generated if TOR [7] = IER [0] = 1. A new incoming data word or RX FIFO empty clears Irpt_TOUT.

4.25 I2C interface

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).

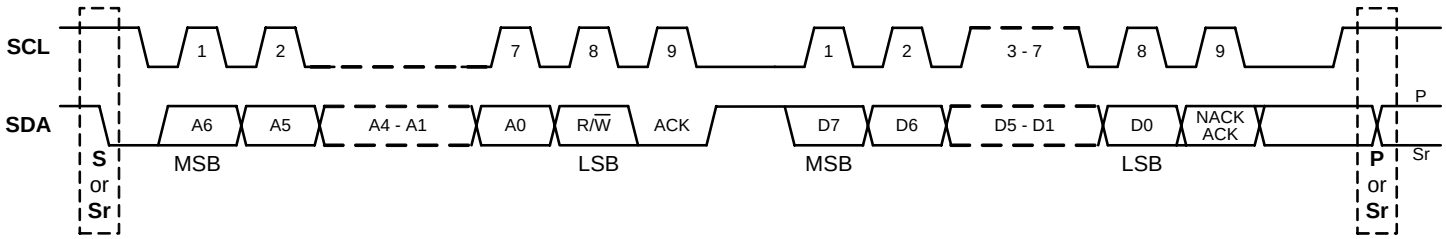
The I²C Master Core includes the following features:

- 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

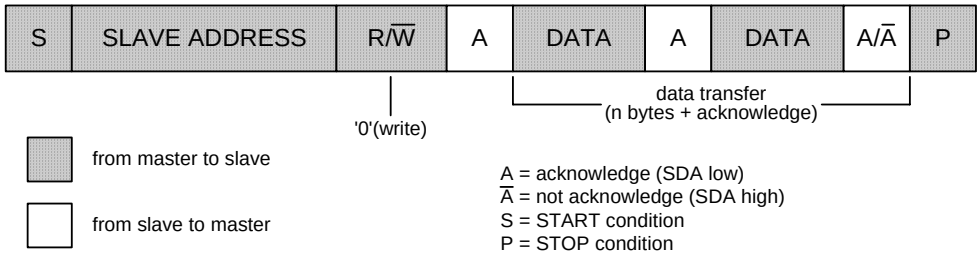
4.25.1 I²C Protocol

Normally, a standard communication consists of four parts:

1) START or Repeated START signal generation
2) Slave address transfer
3) Data transfer
4) STOP signal generation

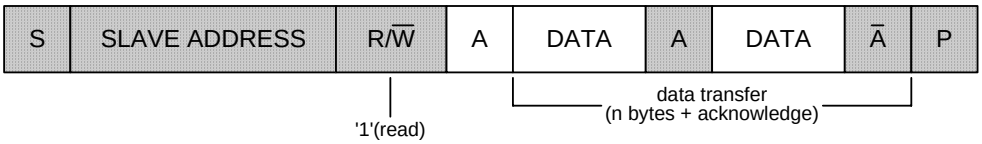


Data transfer on the I²C-bus



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

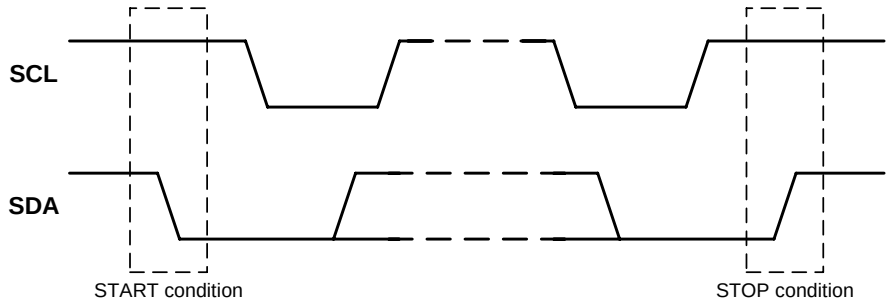
When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the **S-bit**, is defined as a **HIGH to LOW** transition on the SDA line while SCL is **HIGH**. 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.

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**.

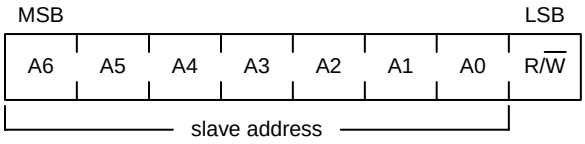


START and STOP conditions

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.



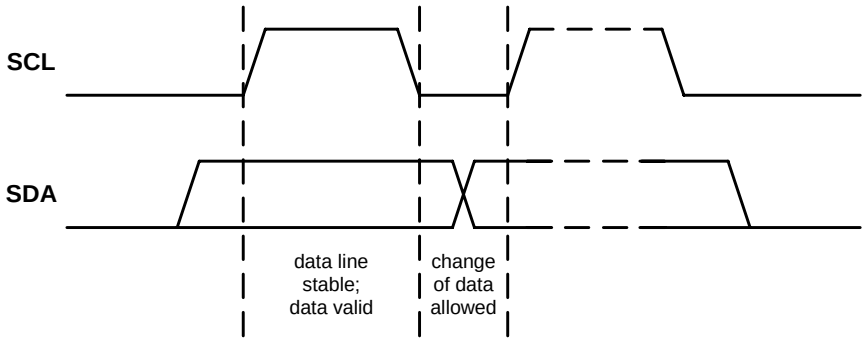
The first byte after the START procedure

Data Transfer

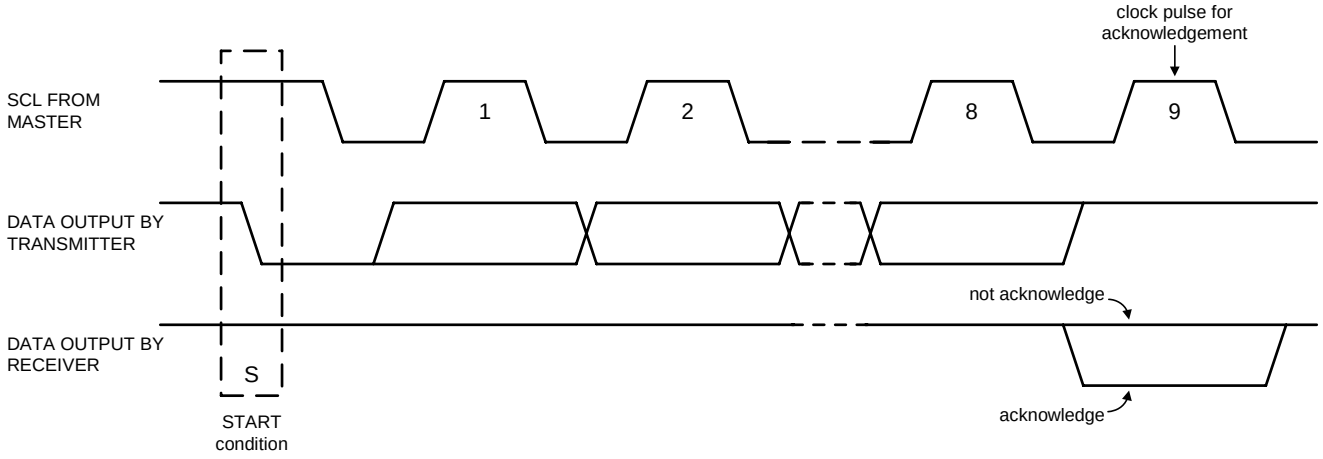
Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a **Not Acknowledge (NACK)**, 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 I²C-bus



Acknowledge on the I²C-bus

4.25.2 I²C Programming Examples

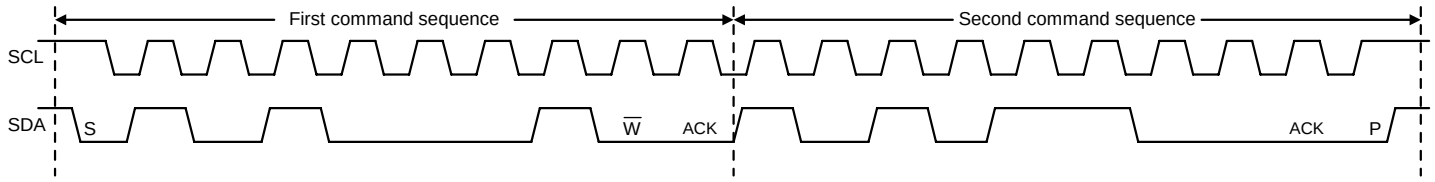
Example 1

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.
   –– Wait for interrupt or I2C_TIP flag to negate ––
5) Read I2C_RxACK bit from CSR Register, it should be '0'.
6) Set Tx_NUM = 0x0.
7) Set STOP bit.
   –– Wait for interrupt or I2C_TIP flag to negate ––



NOTE: Please note that the time for the Interrupt Service Routine is not shown here. It is assumed that the ISR is much faster then the I^2C cycle time, and therefore not visible.

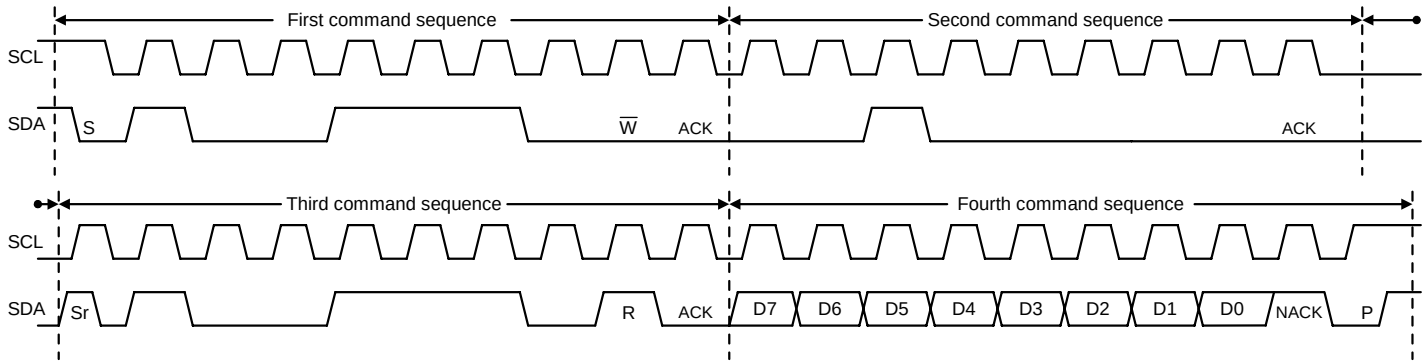
Example 2

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

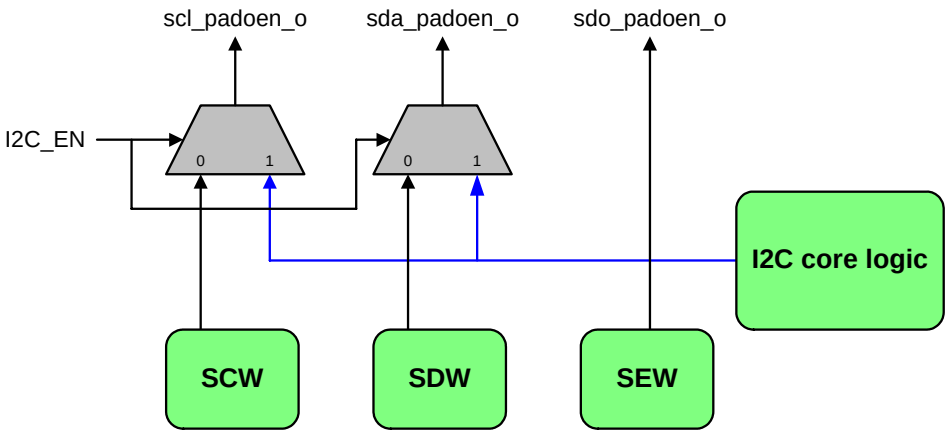
Commands:
1) Write a value into DIVIDER to determine the frequency of serial clock.
2) Set Tx_NUM = 0x0 and set I2C_EN = 1 to enable I^2C core.
3) Write 0x9C (address + write bit) to TxR[7:0], set START bit and WRITE bit.
   –– Wait for interrupt or I2C_TIP flag to negate ––
4) Read I2C_RxACK bit from 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 I^2C cycle time, and therefore not visible.

4.25.3 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.

4.25.4 I²C Serial Interface Control Registers Map

R: read only, **W**: write only, **R/W**: both read and write
Base Address: 0xFFF8_9000

Register	Offset	R/W/C	Description	Reset Value
I2C_BA = 0xFFF8_9000				
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		Reserved	IF	IE	I2C_EN

Bits	Descriptions	
[31:12]	Reserved	Reserved
[11]	I2C_RxACK	**Received Acknowledge From Slave (Read only)** This flag represents acknowledge from the addressed slave. • 0 = Acknowledge received (ACK). • 1 = Not acknowledge received (NACK).
[10]	I2C_BUSY	**I²C 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 I²C 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 I²C master core except SWR.
[7:6]	Reserved	Reserved
[5:4]	Tx_NUM	**Transmit Byte Counts**

		These two bits represent how many bytes are remained to transmit. When a byte has been transmitted, the Tx_NUM will decrease 1 until all bytes are transmitted (Tx_NUM = 0x0) or NACK received from slave. Then the interrupt signal will assert if IE was set. 0x0 = Only one byte is left for transmission. 0x1 = Two bytes are left to for transmission. 0x2 = Three bytes are left for transmission. 0x3 = Four bytes are left for transmission. **NOTE:** When NACK received, Tx_NUM will not decrease.
[3]	**Reserved**	**Reserved**
[2]	**IF**	**Interrupt Flag** The Interrupt Flag is set when: • Transfer has been completed. • Transfer has not been completed, but slave responded NACK (in multi-byte transmit mode). • Arbitration is lost. **NOTE:** This bit is read only, but can be cleared by writing 1 to this bit.
[1]	**IE**	**Interrupt Enable** • 0 = **Disable** I^2C Interrupt. • 1 = **Enable** I^2C Interrupt.
[0]	**I2C_EN**	**I^2C Core Enable** • 0 = **Disable** I^2C core, serial bus outputs are controlled by SDW/SCW. • 1 = **Enable** I^2C core, serial bus outputs are controlled by I^2C 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 I²C bus will put on this register. The I²C core only used 8-bit receive buffer.

Data Transmit Register (TxR)

Register	Offset	R/W/C	Description	Reset Value
TxR	0x14	R/W	Data Transmit Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**Tx**	**Data Transmit Register** The I²C core used 32-bit transmit buffer and provide multi-byte transmit function. Set CSR[Tx_NUM] to a value that you want to transmit. I²C core will always issue a transfer from the highest byte first. For example, if CSR[Tx_NUM] = 0x3, Tx[31:24] will be transmitted first, then Tx[23:16], and so on. In case of a data transfer, all bits will be treated as data. In case of a slave address transfer, the first 7 bits will be treated as 7-bit address and the LSB represent the R/W bit. In this case, LSB = 1, reading from slave LSB = 0, writing to slave

5 Electrical Specifications

5.1 Absolute Maximum Ratings

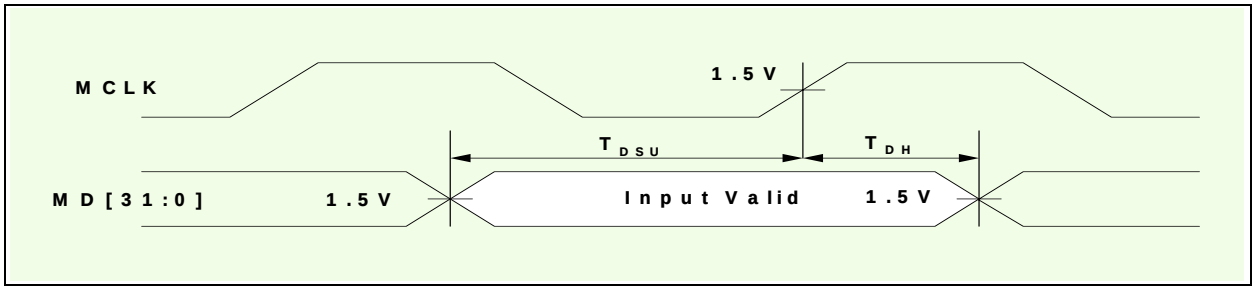
Ambient temperature ..... 0 °C~ 70 °C
Storage temperature ..... -40 °C ~ 125 °C
Voltage on any pin ..... -0.5V ~ 3.6V
Power supply voltage (Core logic) ..... -0.5V ~ 1.98V
Power supply voltage (IO Buffer) ..... -0.5V ~ 3.6V
Injection current (latch-up testing) 100mA
Crystal Frequency 27MHz

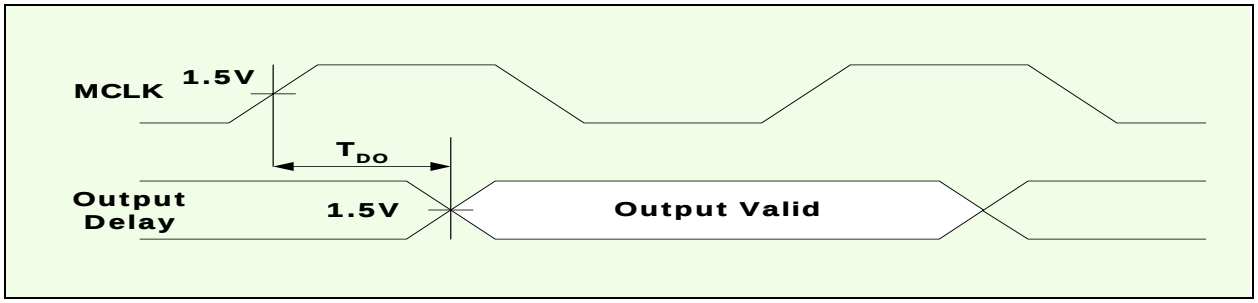
5.2 DC Specifications

Parameter			Condition	Min	Typ	Max	Unit
V_{DDIO33}	IO Post-Driver Voltage			3.00	3.30	3.60	V
V_{DD18}	Core Logic, Pre-Driver Voltage			1.62	1.80	1.98	V
V_{IL}	Input Low Voltage			-0.3		0.8	V
V_{IH}	Input High Voltage			2.0		5.5	V
V_T	Threshold Point			1.30	1.41	1.49	V
V_T^+	Shmitt trig. Low to High threshold point			1.49	1.54	1.58	V
V_T^-	Shmitt trig. High to Low threshold point			1.24	1.29	1.34	V
I_{CC}	Supply Current		FCPU = 81MHz			TBD	mA
I_I	Input Leackage Current		VI = 3.3V or 0V	-1		1	uA
I_{OZ}	Tri-State Output Leakage Current		VO = 3.3V or 0V	-1		1	uA
V_{OL}	Output Low Voltage		Depend on driving			0.4	V
V_{OH}	Output High Voltage		Depend on driving	2.4			V
I_{OL}	Low Level Output Current	4mA	V_{OL} = 0.4V	7.9	9.2	15.8	mA
		8mA	V_{OL} = 0.4V	11.8	15.6	23.8	mA
		12mA	V_{OL} = 0.4V	15.8	21.6	31.7	mA
		16mA	V_{OL} = 0.4V	19.7	30.1	39.6	mA
I_{OH}	High Level Output Current	4mA	V_{OH} = 2.4V	6.4	10.7	21.3	mA
		8mA	V_{OH} = 2.4V	12.7	18.9	42.6	mA
		12mA	V_{OH} = 2.4V	17.5	25.0	58.5	mA
		16mA	V_{OH} = 2.4V	23.8	48.2	79.8	mA
R_{PU}	Pull-up Resistor			38	54	84	KOhm

5.3 AC Specifications

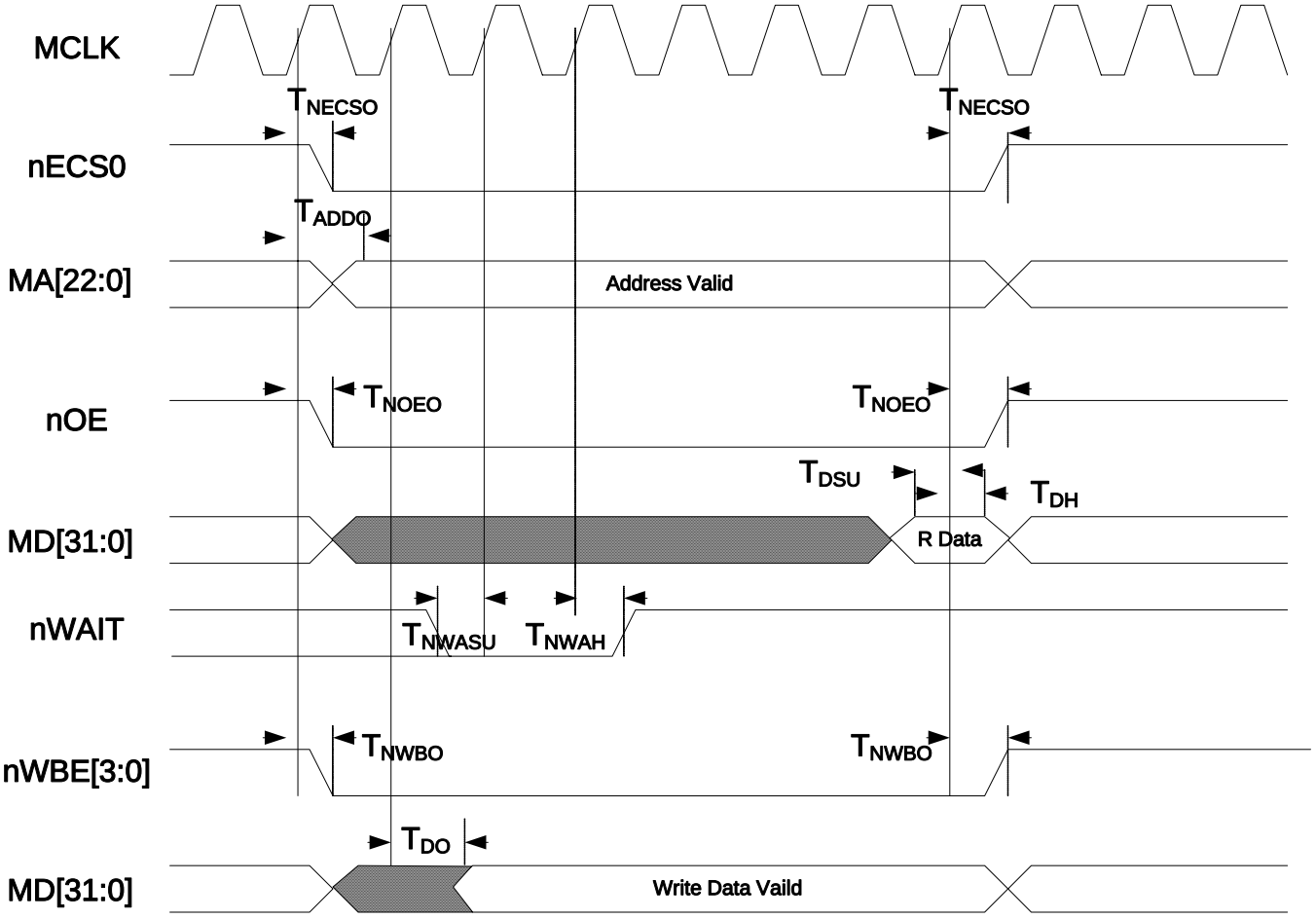
5.3.1 EBI/SDRAM Interface AC Characteristics





Parameter		Min	Max	Unit
T_{DSU}	D [31:0] Setup Time	0		ns
T_{DH}	D [31:0] Hold Time	7		ns
T_{DO}	D [31:0], A [24:0], nSCS [1:0], SDQM [3:0], CKE, nSWE, nSRAS, nSCAS	2	7	ns

5.3.2 EBI/(ROM/SRAM/External I/O) AC Characteristics



Parameter	SYN	MIN	MAX	UNIT
Address Output Delay Time	T_{ADDO}	2	7	ns
ROM/SRAM/Flash or External I/O Chip Select Delay Time	T_{NCSO}	2	7	ns
ROM/SRAM or External I/O Bank Output Enable Delay	T_{NOEO}	2	7	ns
ROM/SRAM or External I/O Bank Write Byte Enable Delay	T_{NWBO}	2	7	
Read Data Hold Time	T_{DH}	7		
Read Data Setup Time	T_{DSU}	0		
Write Data Output Delay Time (SRAM or External I/O)	T_{DO}	2	7	
External Wait Setup Time	T_{NWASU}	3		
External Wait Hold Time	T_{NWAH}	1		

5.4 Audio DAC Characteristic

Parameter	MIN	TYP	MAX	UNIT
Operating Voltage	3.0	3.3	3.6	V
Operating Voltage	1.62	1.8	1.98	V
Minimum Load Resistance		16		Ohm
Maximum Output Voltage Amplitude (R_L = 16Ohm)		1.8		V
Maximum Output Power		20		mW
THD + N (RL=16Ohm, f=1KHz, Output Power=20mW)		0.6		%
SNR		-80		dB

5.5 TV DAC Characteristic

5.5.1 DC Specifications

Parameter	SYM	MIN	TYP	MAX	UNIT
Operating Voltage	AVDD1 AVDD2	2.8	3.3	3.6	V
Operating Voltage	VDD	1.62	1.8	1.98	V
Full Scale Current	I_{OFS}		30.032		mA
Output Voltage	V_{OUT}		1.1262		V
Supply Current AVDD1 AVDD2			 0.270 37.433		mA
DAC Resolution	--			8	Bits
Integral Non-Linearity Error	INL	-1.0		+1.0	LSB
Differential Non-Linearity	DNL	-1.0		+1.0	LSB

5.5.2 AC Specifications

Parameter	SYM	MIN	TYP	MAX	UNIT

CK Period	T_{CK}	35			ns
CK to Valid Output	T_D		2.31	2.99	ns
Rise Time	T_r		2	2.32	ns
Fall Time	T_f		1.83	1.99	ns
Settling Time	T_{settle}		5.59	>37	ns

5.6 ADC Characteristic

5.6.1 DC/AC Specifications

Parameter		SYM	MIN	TYP	MAX	UNIT
Operation Voltage		VDD	3.0	3.3	3.6	V
Operation Current		IDD		330		
Reference Voltage		VREFP	2		VDD	V
Reference Current	VREF=3.3V	IREFP		250		
	VREF=2V			150		
Resolution				10		
Conversion time			3.33		10	
Sampling rate			100K		300K	
Integral Non-Linearity Error		INL	-0.5		0.5	
Differential Non-Linearity		DNL	-0.5		0.5	

5.7 Low voltage detect Characteristic

Parameter	SYN	MIN	TYP	MAX	UNIT
Operation voltage	VDD	3.0	3.3	3.6	V
Operation Current	IDD		270		uA
Temperature		-40	27	85	℃

LVD detect level

LVD_SW2	LVD_SW1	LVDSW0	Detect Voltage
0	0	0	2.0V
0	0	1	2.1V
0	1	0	2.2V
0	1	1	2.3V
1	0	0	2.4V
1	0	1	2.5V
1	1	0	2.6V
1	1	1	2.7V

5.8 Low voltage reset Characteristic

Parameter	SYN	MIN	TYP	MAX	UNIT
Operation voltage	VDD	3.0	3.3	3.6	V
Operation Current	IDD		70		uA
Temperature		-40	27	85	℃
LVR detect voltage		2.16	2.4	2.64	V

6 Pin Diagram

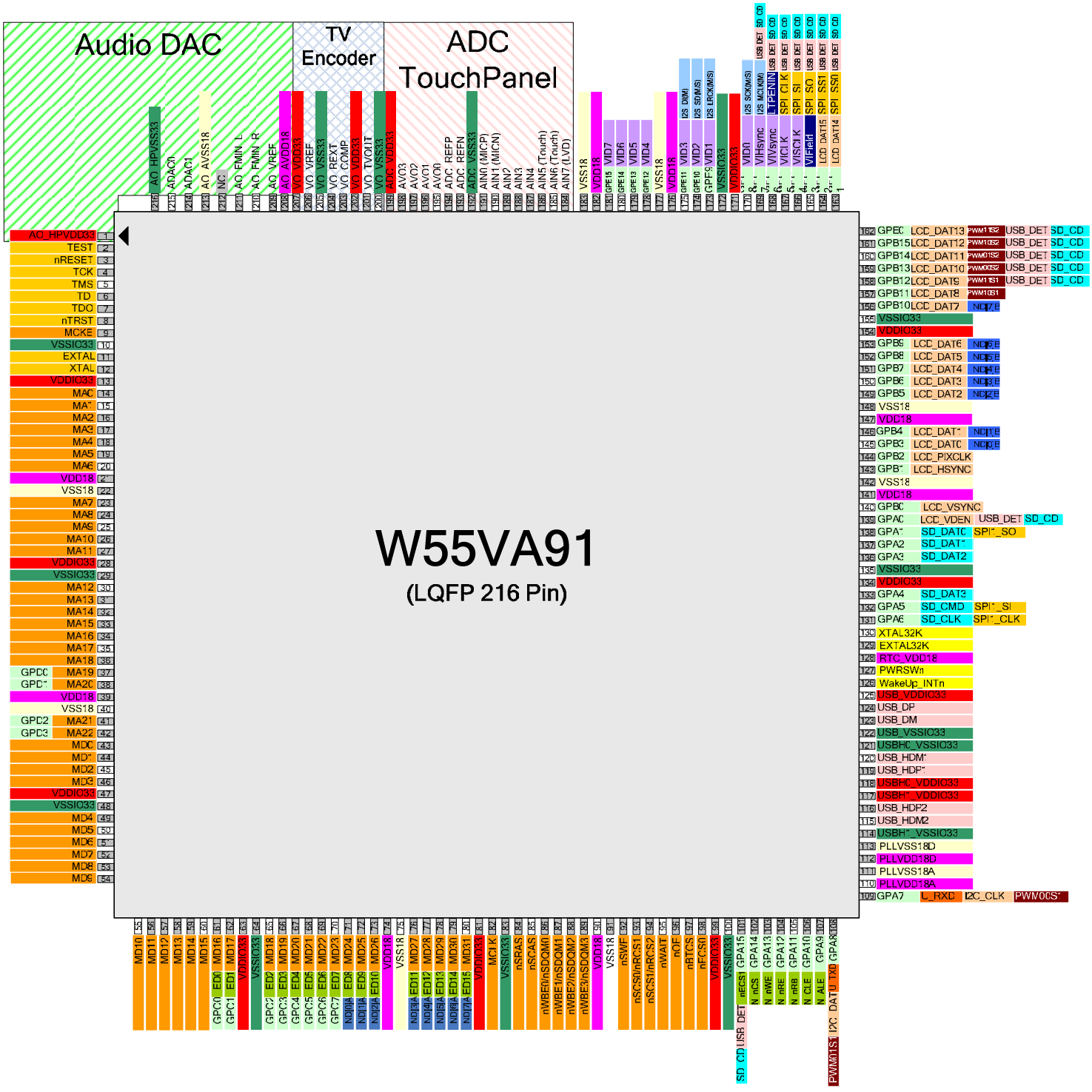


Figure 6—1 W55VA91 Pin Diagram

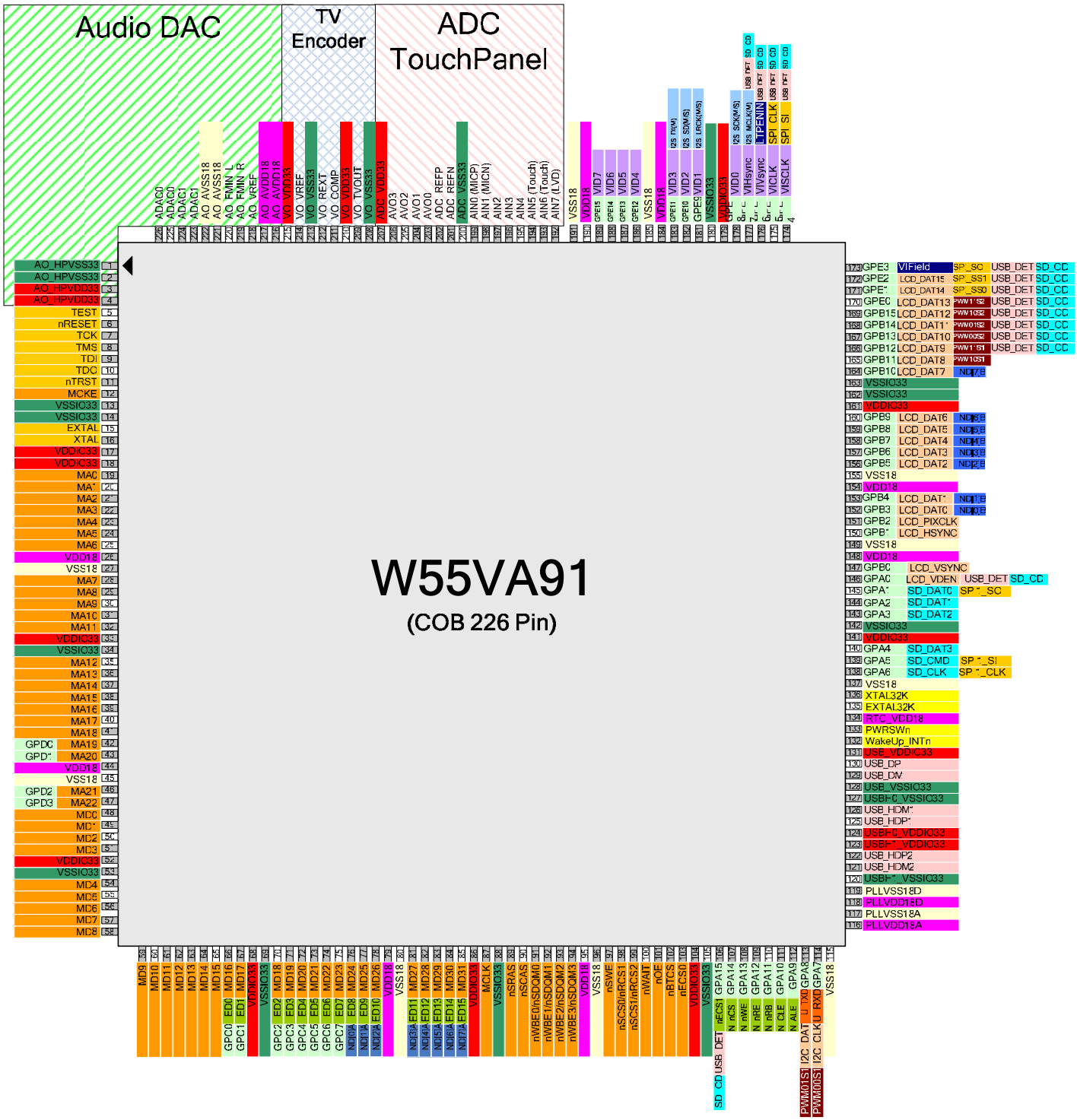


Figure 6—2 W55VA91 Pad Diagram for Chip-form on Board

7 Pin Description

I: input; O: output; IO: bi-direction; IU: input pull-up; ID: input pull-down; P: power; G: Ground

Pin Name	IO Type	Pad Type	Description
Clock & Reset			
EXTAL (27M)	I	PXOE1CDG	27MHz External Clock / Crystal Input
XTAL (27M)	O		27 MHz Crystal Output
nRESET	IU	PRS0408CDG	System Reset, active-low
TAP Interface			
TCK	ID	PRS0408CDG	JTAG Test Clock, internal pull-down with 49K ohm
TMS	IU	PRS0408CDG	JTAG Test Mode Select, internal pull-up with 54K ohm
TDI	IU	PRS0408CDG	JTAG Test Data in, internal pull-up with 54K ohm
TDO	O	PRS0408CDG	JTAG Test Data out
nTRST	IU	PRS0408CDG	JTAG Reset, active-low, internal pull-up with 54K ohm
External Bus Interface			
MA [22:19]	IO	PRS040812CDG	Address Bus of external memory and IO devices.
GPD [3:0]			General I/O port D bit 3 ~ 0. NOTE: In default, this function is disabled and used as memory address bit 22 ~ 19 until program setting to GPIO usage.
MA [18:8]	O	PRS040812CDG	Address Bus of external memory and IO devices. NOTE: MA[18:14] are also for hardware power-on setting in RESET status.
MA [7:0]	O	PRC040812CDG	Address Bus [7:0] of external memory and IO devices.
MD [15:0]	IO	PRC040812CDG	Data bus (LSB) of external memory and IO devices
MD [23:16]	IO	PRC040812CDG	Data bus (MSB) of external memory and IO devices.
ED [7:0]			EXIO data bus [15:8]
GPC[7:0]			General I/O port C bit 15 ~ 0.
MD [31:24]	IO	PRC040812CDG	Data bus [23:16] of external memory and IO devices
ED [15:8]			EXIO data bus [7:0]
ND [7:0]A			Nand Flash Data [7:0] portA
MCLK	O	PRS1216CDG	System Master Clock Out, SDRAM clock.
nSCAS	O	PRC040812CDG	Column Address Strobe for SDRAM, active-low.
nSRAS	O	PRC040812CDG	Row Address Strobe for SDRAM, active-low.
nWE	O	PRC040812CDG	SDRAM Write Enable, active-low
MCKE	O	PRC040812CDG	SDRAM Clock Enable, active-high
nWBE [3:0]	O	PRC040812CDG	Write Byte Enable for specific device (nECS0).
SDQM [3:0]	O		Data Bus Mask signal for SDRAM (nSCS [1:0]), active-low.
nSCS [1:0]	O	PRC040812CDG	SDRAM chip select for two external banks, active-low.
nROMCS [2:1]	O		Extended bank ROM/Flash chip select, active-low.
nBTCS	O	PRC040812CDG	ROM/Flash Chip Select, active-low.
nECS0	O	PRC040812CDG	External I/O Device Chip Select Bank 0, active-low.
nOE	O	PRC040812CDG	ROM/Flash, External Memory Output Enable, active-low.

Pin Name	IO Type	Pad Type	Description
nWAIT	IU	PRC040812CDG	External Wait, active-low. This pin indicates that the external devices need more active cycle during access operation.
GPIO Port A			
GPA [15]	IO	PRS0408CDG	General I/O Port A bit 15.
nECS1			External I/O Device Chip Select Bank1, active-low.
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPA [14]	IO	PRS0408CDG	General I/O Port A bit 14.
NAND_nCS			Nand Flash Chip Select, active-low
GPA [13]	IO	PRS0408CDG	General I/O Port A bit 13.
NAND_nWE			Nand Flash Write Enable, active-low.
GPA [12]	IO	PRS0408CDG	General I/O Port A bit 12.
NAND_nRE			Nand Flash Read Enable, active-low.
GPA [11]	IO	PRS0408CDG	General I/O Port A bit 11.
NAND_nRB			Nand Flash Ready/Busy input.
GPA [10]	IO	PRS0408CDG	General I/O Port A bit 10.
NAND_CLE			Nand Flash Command Latch Enable.
GPA [9]	IO	PRS0408CDG	General I/O Port A bit 9.
NAND_ALE			Nand Flash Address Latch Enable.
GPA [8]	IO	PRS0408CDG	General I/O Port A bit 8.
UART_TXD			Serial Transmit of UART interface, output.
PWM01S1			PWM0 channe1 output(set1)
I2C_DATA			I2C data input.
GPA [7]	IO	PRS0408CDG	General I/O Port A bit 7.
UART_RXD			Serial Reception of UART interface, input.
PWM00S1			PWM0 channe0 output(set1)
I2C_CLK			I2C clock input.
GPA [6]	IO	PRS0408CDG	General I/O Port A bit 6.
SD_CLK			SD/SDIO Mode: host clock to card, output.
SPI1_CLK			Master clock output for SPI 1 interface.
GPA [5]	IO	PRS0408CDG	General I/O Port A bit 5.
SD_CMD			SD/SDIO Mode: host command output to card and response input from card.
SPI1_SI			Serial input for SPI 1 interface.
GPA [4]	IO	PRS0408CDG	General I/O Port A bit 4.
SD_DAT [3]			SD/SDIO Mode: data line bit 3.
GPA [3]	IO	PRS0408CDG	General I/O Port A bit 3.
SD_DAT [2]			SD/SDIO Mode: data line bit 2.
GPA [2]	IO	PRS0408CDG	General I/O Port A bit 2.
SD_DAT [1]			SD/SDIO Mode: data line bit 1.
GPA [1]	IO	PRS0408CDG	General I/O Port A bit 1.
SD_DAT [0]			SD/SDIO Mode: data line bit 0.
SPI1_SO			Serial output for SPI 1 interface.
GPA [0]	IO	PRS0408CDG	General I/O Port A bit 0.
LCD_VDEN			Data enable bit to LCD module. (*2)

Pin Name	IO Type	Pad Type	Description
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPIO Port B			
GPB [15]	IO	PRS0408CDG	General I/O Port B bit 15.
LCD_DAT [12]			Bit 12 of Image data output to LCD module.
PWM10S2			PWM1 channe0 output(set2)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPB [14]	IO	PRS0408CDG	General I/O Port B bit 14.
LCD_DAT [11]			Bit 11 of Image data output to LCD module.
PWM01S2			PWM0 channe1 output(set2)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPB [13]	IO	PRS0408CDG	General I/O Port B bit 13.
LCD_DAT [10]			Bit 10 of Image data output to LCD module.
PWM00S2			PWM0 channe1 output(set2)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPB [12]	IO	PRS0408CDG	General I/O Port B bit 12.
LCD_DAT [9]			Bit 9 of Image data output to LCD module.
PWM11S1			PWM1 channe1 output(set1)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPB [11]	IO	PRS0408CDG	General I/O Port B bit 11.
LCD_DAT [8]			Bit 8 of Image data output to LCD module.
PWM10S1			PWM1 channe0 output(set1)
GPB [10]	IO	PRS0408CDG	General I/O Port B bit 10.
LCD_DAT [7]			Bit 7 of Image data output to LCD module.
ND[7]B			Nand Flash Data [7] portB.
GPB [9]	IO	PRS0408CDG	General I/O Port B bit 9.
LCD_DAT [6]			Bit 6 of Image data output to LCD module.
ND[6]B			Nand Flash Data [6] portB.
GPB [8]	IO	PRS0408CDG	General I/O Port B bit 8
LCD_DAT [5]			Bit 5 of Image data output to LCD module.
ND[5]B			Nand Flash Data [5] portB.
GPB [7]	IO	PRS0408CDG	General I/O Port B bit 7.
LCD_DAT [4]			Bit 4 of Image data output to LCD module.
ND[4]B			Nand Flash Data [4] portB.
GPB [6]	IO	PRS0408CDG	General I/O Port B bit 6.
LCD_DAT [3]			Bit 3 of Image data output to LCD module.
ND[3]B			Nand Flash Data [3] portB.
GPB [5]	IO	PRS0408CDG	General I/O Port B bit 5.
LCD_DAT [2]			Bit 2 of Image data output to LCD module.
ND[2]B			Nand Flash Data [2] portB.
GPB [4]	IO	PRS0408CDG	General I/O Port B bit 4.
LCD_DAT [1]			Bit 1 of Image data output to LCD module.
ND[1]B			Nand Flash Data [1] portB.

Pin Name	IO Type	Pad Type	Description
GPB [3]	IO	PRS0408CDG	General I/O Port B bit 3.
LCD_DAT [0]			Bit 0 of Image data output to LCD module.
ND[0]B			Nand Flash Data [0] portB.
GPB [2]	IO	PRS0408CDG	General I/O Port B bit 2.
LCD_PIXCLK			Pixel clock output to LCD module.
GPB [1]	IO	PRS0408CDG	General I/O Port B bit 1.
LCD_HSYNC			Horizontal sync or line sync output to LCD module. (*2)
GPB [0]	IO	PRS0408CDG	General I/O Port B bit 0.
LCD_VSYNC			Vertical sync or frame sync output to LCD module. (*2)
GPIO Port E			
GPE [15]	IO	PRS0408CDG	General I/O Port E bit 15.
VID [7]			Bit 7 of data bus of Video Image Input interface.
GPE [14]	IO	PRS0408CDG	General I/O Port E bit 14.
VID [6]			Bit 6 of data bus of Video Image Input interface.
GPE [13]	IO	PRS0408CDG	General I/O Port E bit 13.
VID [5]			Bit 5 of data bus of Video Image Input interface.
GPE [12]	IO	PRS0408CDG	General I/O Port E bit 12.
VID [4]			Bit 4 of data bus of Video Image Input interface.
GPE [11]	IO	PRS0408CDG	General I/O Port E bit 11.
VID [3]			Bit 3 of data bus of Video Image Input interface.
I2S_DI			IIS Master serial data input
GPE [10]	IO	PRS0408CDG	General I/O Port E bit 10.
VID [2]			Bit 2 of data bus of Video Image Input interface.
I2S_DO			IIS serial data output
GPE [9]	IO	PRS0408CDG	General I/O Port E bit 9.
VID [1]			Bit 1 of data bus of Video Image Input interface.
I2S_LRCK			IIS Receiver word select input.
GPE [8]	IO	PRS0408CDG	General I/O Port E bit 8.
VID [0]			Bit 0 of data bus of Video Image Input interface.
I2S_SCK			IIS Receiver serial clock input
GPE [7]	IO	PRS0408CDG	General I/O Port E bit 7.
VIHsync			Default H-sync input of Video Image Input interface. (*2)
I2S_MCLK			IIS Master Clock output
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [6]	IO	PRS0408CDG	General I/O Port E bit 6.
VIVsync			Default V-sync input of Video Image Input interface. (*2)
LIGHTPIN			Light gun signal input for GPU. (*3)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [5]	IO	PRS0408CDG	General I/O Port E bit 5.
VICLK			Default clock input of Video Image Input interface. (*2)
SPI0_CLK			Master clock output or Slave clock input for SPI 0 interface.
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [4]	IO	PRS0408CDG	General I/O Port E bit 4.
VISCLK			Clock output to sensor system clock in Video Image Input function.

Pin Name	IO Type	Pad Type	Description
SPI0_SI			Serial input for SPI0 Master/Slave interface.
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [3]	IO	PRS0408CDG	General I/O Port E bit 3.
VIField			Video Interlace Field Identification.
SPI0_SO			Serial output for SPI0 Master/Slave interface.
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [2]	IO	PRS0408CDG	General I/O Port E bit 2.
LCD_DAT [15]			Bit 15 of Image data output to LCD module.
SPI0_SS1			Slave select 1 Master output for SPI0 interface. (*2)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [1]	IO	PRS0408CDG	General I/O Port E bit 1.
LCD_DAT [14]			Bit 14 of Image data output to LCD module.
SPI0_SS0			Slave select 0 Master output or Slave input for SPI0 interface. (*2)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
GPE [0]	IO	PRS0408CDG	General I/O Port E bit 0.
LCD_DAT [13]			Bit 13 of Image data output to LCD module.
PWM11S2			PWM1 channe1 output(set2)
USBDET/USBOC/ SD_CD			USB floating detect/USB over current detect/SD card detect option
USB Device Port			
USB D+	IO		USB D+
USB D-	IO		USB D-
USB Host Port1			
USB HD1+	IO		USB Host Port1 D+
USB HD1-	IO		USB Host Port1 D -
USB Host Port2			
USB HD2+	IO		USB Host Port2 D+
USB HD2-	IO		USB Host Port2 D -
RTC			
WakeUP_INTn	I	PRS0408CDG	RTC Wake up Interrupt input. (DO NOT apply higher than 1.8V on this pin)
PWRSWn	O	PRS0408CDG	Power Switch output, open drain output. (DO NOT apply higher than 1.8V on this pin)
EXTAL32K (32768)	I	xosc32k	32768Hz External Clock / Crystal Input
XTAL 32K(32768)	O		32768Hz Crystal Output
Miscellaneous Logic			
TEST	ID	PRS0408CDG	Test mode enable pin, active high. In normal function mode, this pin should be shorted to ground or left unconnected. In opened data sheet, this pin should be tagged as “VSS33” or “NC”.

Pin Name	IO Type	Pad Type	Description
ADC / Touch Panel			
AI [7]	I	--	A/D Converter analog input channel 7.
LVDIN			Low voltage detection input.
AI [6]	I	--	A/D Converter analog input channel 6.
TouchX			Touch Screen panel X coordinate input
AI [5]	I	--	A/D Converter analog input channel 5.
TouchY			Touch Screen panel Y coordinate input
AI [4]	I	--	A/D Converter analog input channel 4.
AI [3]	I	--	A/D Converter analog input channel 3.
AI [2]	I	--	A/D Converter analog input channel 2.
AI [1]	I	--	A/D Converter analog input channel 1.
Mic-in_D-			Microphone negative input
AI [0]	I	--	A/D Converter analog input channel 0.
Mic-in_D+			Microphone positive input
ADC_REFN	O	--	A/D Converter reference -.
ADC_REFP	O	--	A/D Converter reference +.
TP_AVO [3]	IO	--	Touch panel I/O 3.
YM			Touch Panel YM
TP_AVO [2]	IO	--	Touch panel I/O 2.
YP			Touch Panel YP
TP_AVO [1]	IO	--	Touch panel I/O 1.
XM			Touch Panel XM
TP_AVO [0]	IO	--	Touch panel I/O 0.
XP			Touch Panel XP
TV Encoder			
VO_TVOUT	O	--	Composite/Chroma output
VO_COMP	O	--	External capacitor connection
VO_REXT	O	--	External resistor connection
VO_VREF	O	--	Reference voltage output
Audio DAC			
AO_VREF	O	--	Audio analog reference.
AO_FMIN_R	I	--	Audio DAC FM R input.
AO_FMIN_L	I	--	Audio DAC FM L input.
ADAC1	O	--	Audio DAC output channel 1.
ADAC0	O	--	Audio DAC output channel 0.
Power / Ground			
PLLVSS18D	G	PVSS18VAC	PLL digital ground (0V).
PLLVDD18D	P	PVDD18VAC	PLL digital power (1.8V).
PLLVSS18A	G	PVSS18VAC	PLL analog ground (0V).
PLLVDD18A	P	PVDD18VAC	PLL analog power (1.8V).
VDDIO33	P	PVDD2CDG / PVDD2POC	IO buffer power (3.3V).
VSSIO33	G	PVSS2DGZ	IO buffer ground (0V).
VDD18	P	PVDD1CDG	Core logic power (1.8V).
VSS18	G	PVSS1CDG	Core logic ground (0V).
USB_VDDIO33	P	--	USB Device IO buffer power (3.3V).

Pin Name	IO Type	Pad Type	Description
USB_VSSIO33	G	--	USB Device IO buffer ground (0V).
USBH0_VDDIO33	P	--	USB Host Port1 IO buffer power (3.3V).
USBH0_VSSIO33	G	--	USB Host Port1 IO buffer ground (0V).
USBH1_VDDIO33	P	--	USB Host Port2 IO buffer power (3.3V).
USBH1_VSSIO33	G	--	USB Host Port2 IO buffer ground (0V).
VD18	P	PVDD3CDG	RTC Core/EXTAL32K power
ADC_VDD33	P	--	A/D Converter power (3.3V).
ADC_VSS33	G	--	A/D Converter ground (0V).
VO_VDD33_1	P	--	TV Encoder analog power set 1 (3.3V).
VO_VSS33_1	G	--	TV Encoder analog ground set 1 (3.3V).
VO_VDD33_2	P	--	TV Encoder analog power set 2 (3.3V).
VO_VSS33_2	G	--	TV Encoder analog ground set 2 (3.3V).
AO_AVDD18	P	--	Audio DAC analog power (1.8V).
AO_AVSS	G	--	Audio DAC analog ground (0V).
AO_HPVSS33	G	--	Audio DAC headphone driver dedicated ground (0V).
AO_HPVSS33_2	G	--	Audio DAC headphone driver dedicated ground (0V).
AO_HPVDD33	P	--	Audio DAC headphone driver dedicated power (0V).
AO_HPVDD33_2	P	--	Audio DAC headphone driver dedicated power (0V).

Table 7—1 W55VA91 Pin Descriptions

NOTE:

***1 This definition is fully controlled by program and can be altered and assigned to another GPIO by program.**

***2 Active high or low can be configured by program.**

***3 When used as light gun input, this pin should be left in a default mode, e.g. GPIO input mode.**

Pad Type	I/O	Driving Capability	Description
PRC0408CDG	IO	4/8 mA	Dual-Driving I/O Cells with Slew Controlled Output.
PRC040812CDG	IO	4/8/12 mA	Triple-Driving I/O Cells with Slew Controlled Output.
PRS0408DGZ	IO	4/8 mA	Dual-Driving I/O Cells with Slew Controlled Output Schmitt Trigger input, 1.8V IO Buffer, 3.3V-Tolerant
PXOE1CDG	IO	--	Crystal Oscillator Input and Output with enable control
xosc32k_18u33v	IO	--	Crystal Oscillator Input and Output with enable control especially for 32768Hz (This is a local design IO cell).
Note: The Pad Type information is referred to the TSMC 0.18um 3.3V I/O Library.			

Table 7—2 W55VA91 Pad Type Descriptions

Default Function Name	Alternative Function 1	Alternative Function 2	Alternative Function 3	Alternative Function 4
EBI				
MA[0]				
MA[1]				
MA[2]				
MA[3]				

Default Function Name	Alternative Function 1	Alternative Function 2	Alternative Function 3	Alternative Function 4
MA[4]				
MA[5]				
MA[6]				
MA[7]				
MA[8]				
MA[9]				
MA[10]				
MA[11]				
MA[12]				
MA[13]				
MA[14]				
MA[15]				
MA[16]				
MA[17]				
MA[18]				
MA[19]	GPD[0]			
MA[20]	GPD[1]			
MA[21]	GPD[2]			
MA[22]	GPD[3]			
MD[0]				
MD[1]				
MD[2]				
MD[3]				
MD[4]				
MD[5]				
MD[6]				
MD[7]				
MD[8]				
MD[9]				
MD[10]				
MD[11]				
MD[12]				
MD[13]				
MD[14]				
MD[15]				
MD[16]	ED[0]	GPC[0]		
MD[17]	ED[1]	GPC[1]		
MD[18]	ED[2]	GPC[2]		
MD[19]	ED[3]	GPC[3]		
MD[20]	ED[4]	GPC[4]		
MD[21]	ED[5]	GPC[5]		
MD[22]	ED[6]	GPC[6]		
MD[23]	ED[7]	GPC[7]		
MD[24]	ED[8]	ND[0]A		
MD[25]	ED[9]	ND[1]A		
MD[26]	ED[10]	ND[2]A		
MD[27]	ED[11]	ND[3]A		
MD[28]	ED[12]	ND[4]A		
MD[29]	ED[13]	ND[5]A		
MD[30]	ED[14]	ND[6]A		
MD[31]	ED[15]	ND[7]A		
MCLK				

Default Function Name	Alternative Function 1	Alternative Function 2	Alternative Function 3	Alternative Function 4
nSCAS				
nSRAS				
nWE				
MCKE				
nSDQM[0]				
nSDQM[1]				
nSDQM[2]				
nSDQM[3]				
nSCS[0]				
nSCS[1]				
nBTCS				
nECS0				
nOE				
nWait				
GPIO				
GPA[0]	LCD_VDEN		USBDET/USBOC	SD_CD
GPA[1]	SD_DAT[0]	SPI1_SO		
GPA[2]	SD_DAT[1]			
GPA[3]	SD_DAT[2]			
GPA[4]	SD_DAT[3]			
GPA[5]	SD_CMD	SPI1_SI		
GPA[6]	SD_CLK	SPI1_CLK		
GPA[7]	UART_RXD	PWMO00	I2C_CLK	
GPA[8]	UART_TXD	PWMO01	I2C_DATA	
GPA[9]		NAND_ALE		
GPA[10]		NAND_CLE		
GPA[11]		NAND_nRB		
GPA[12]		NAND_nRE		
GPA[13]		NAND_nWE		
GPA[14]		NAND_nCS		
GPA[15]	nECS1		USBDET/USBOC	SD_CD
GPB[0]	LCD_VSYNC			
GPB[1]	LCD_HSYNC			
GPB[2]	LCD_PIXCLK			
GPB[3]	LCD_DAT[0]	ND[0]B		
GPB[4]	LCD_DAT[1]	ND[1]B		
GPB[5]	LCD_DAT[2]	ND[2]B		
GPB[6]	LCD_DAT[3]	ND[3]B		
GPB[7]	LCD_DAT[4]	ND[4]B		
GPB[8]	LCD_DAT[5]	ND[5]B		
GPB[9]	LCD_DAT[6]	ND[6]B		
GPB[10]	LCD_DAT[7]	ND[7]B		
GPB[11]	LCD_DAT[8]	PWMO10		
GPB[12]	LCD_DAT[9]	PWMO11	USBDET/USBOC	SD_CD
GPB[13]	LCD_DAT[10]	PWMO20	USBDET/USBOC	SD_CD
GPB[14]	LCD_DAT[11]	PWMO21	USBDET/USBOC	SD_CD
GPB[15]	LCD_DAT[12]	PWMO30	USBDET/USBOC	SD_CD
GPE[0]	LCD_DAT[13]	PWMO31	USBDET/USBOC	SD_CD
GPE[1]	LCD_DAT[14]	SPI0_SS0	USBDET/USBOC	SD_CD
GPE[2]	LCD_DAT[15]	SPI0_SS1	USBDET/USBOC	SD_CD
GPE[3]	VIField	SPI0_SO	USBDET/USBOC	SD_CD

Default Function Name	Alternative Function 1	Alternative Function 2	Alternative Function 3	Alternative Function 4
GPE[4]	VISCLK	SPI0_SI	USBDET/USBOC	SD_CD
GPE[5]	VICLK	SPI0_SCK	USBDET/USBOC	SD_CD
GPE[6]	VIVsnc	LIGHTPIN	USBDET/USBOC	SD_CD
GPE[7]	VIHsnc	I2S-MCLK(M)	USBDET/USBOC	SD_CD
GPE[8]	VID[0]	I2S-BCLK(M) SK(S)		
GPE[9]	VID[1]	I2S-LRCLK(M)(S)		
GPE[10]	VID[2]	I2S-DO(M)SD(S)		
GPE[11]	VID[3]	I2S-DI(M)		
GPE[12]	VID[4]			
GPE[13]	VID[5]			
GPE[14]	VID[6]			
GPE[15]	VID[7]			
TV Encoder				
VO_VDD33_1				
VO_VREF				
VO_VSS33_1				
VO_REXT				
VO_CEXT				
VO_VDD33_2				
VO_OUT				
VO_VSS33_2				
ADC				
ADC_VDD33				
TP_AVO3				
TP_AVO2				
TP_AVO1				
TP_AVO0				
ADC_REFP				
ADC_REFN				
ADC_VSS33				
ADC_AI0	MIC+			
ADC_AI1	MIC-			
ADC_AI2				
ADC_AI3				
ADC_AI4				
ADC_AI5	Touch X			
ADC_AI6	Touch Y			
ADC_AI7	LVD			
Audio DAC				
AO_HPVDD33(2)				
AO_HPVSS33(2)				
AO_OUT0(2)				
AO_OUT1(2)				
AO_VREF				
AO_AVSS(2)				
AO_AVDD18(2)				
USB1.1 Device				
USB_VDD33				
USB_DDP				
USB_DDM				
USB_VSS33				

Default Function Name	Alternative Function 1	Alternative Function 2	Alternative Function 3	Alternative Function 4
USB1.1 HOST				
USB_VDD33A				
USB_HDPA				
USB_HDMA				
USB_VSS33A				
USB_VDD33B				
USB_HDPB				
USB_HDMB				
USB_VSS33B				
RTC				
WAKEUP				
PWRSW				

Table 7—3 W55VA91 Multi-Function Pin Description

8 Document Revision History

Version	Date	Remarks
A0	06/20/2008	Formal release version
A1	11/12/2008	Transit from shuttle version to full version
A2	11/21/2008	● Add some notes for SPI booting ● Layout modification for register 'EBIDS'
	03/09/2009	● Modify the description of the "NG_LEVEL" field in the SFR of "LV_STS"
A3	03/13/2009	● Revise the default of AUDIO_CON ● Remove the description of AGC
	03/25/2009	● Add overlapping notes in OlapSel ● Modify the description of SRCTYPE of AIC_SRCx ● Revise pin voltage & crystal frequency of Absolute Maximum Ratings
	04/3/2009	● Revise the RTC PWR_OFF register description ● Revise WAKEUP pin description
A4	04/20/2009	● Add note for overlapping on FBDS & OlapFDSel description ● Revise line buffer overlapping with frame buffer FSADDR, and size setting.
	06/02/2009	● Section 4.1 ~ 4.7