SPCE3200 Programming Guide

V1.1 - Apr 11, 2006

English Version

19, Innovation First Road • Science Park • Hsin-Chu • Taiwan 300 • R.O.C.

Tel: 886-3-578-6005 Fax: 886-3-578-4418 Web: www.sunplus.com

Important Notice

SUNPLUS TECHNOLOGY CO. reserves the right to change this documentation without prior notice. Information provided by SUNPLUS TECHNOLOGY CO. is believed to be accurate and reliable. However, SUNPLUS TECHNOLOGY CO. makes no warranty for any errors which may appear in this document. Contact SUNPLUS TECHNOLOGY CO. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by SUNPLUS TECHNOLOGY CO. for any infringement of patent or other rights of third parties which may result from its use. In addition, SUNPLUS products are not authorized for use as critical components in life support systems or aviation systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Sunplus.

Revision History

Revision	Date	By	Remark	Page Number(s)
1.1	04/11/2006	Edward Su	• Modify valid bits of register 0x8804_0008 (HOR_ACT) to be D11-D0, not D10-D0. • Add description of register 0x8804_00a0. • Add description of register 0x8804_10a0. • Modify chapter “CD SERVO“. • Modify the clock calculation table in Section 9.2(CKG). • Remove DDR related description. • Remove the memory space 0x0802_0000 ~ 0x0802_FFFF of JPG in CH10 (MIU) since the original JPEG CODEC had been replaced by MPEG4. • Modify CH24 (ADC) • Modify CH25 (I2S) • Remove TEXT extension function in CH12 (PPU). • Modify “2 bits 4 color mode” to be 128 total Palette Banks in the table in CH12.3 (Palette Memory). • Modify to be =>”100h means no compression; 80h represents double compression.” in CH12.9.2 (Horizontal compression). • Add note at P_BLNDMA_SRCA_ADDR P_BLNDMA_SRCB_ADDR P_BLNDMA_DEST_ADDR P_BLNDMA_ABASE_ADDR P_BLNDMA_BBASE_ADDR P_BLNDMA_DBASE_ADDR in CH16 (BLNDMA). • Modify bits “SW_LCD” and “SW_CSI” of P_GPIO_DEVICE_SET in CH19 (Control Registers). • Modify bits “SW_PERI” of P_GPIO_FUNC_SET in CH19. • Modify bits “SW_KETC” of P_GPIO_DRAM_SET in CH19. • Modify bit “SW_SERVO_34” of P_GPIO_OUTPUT in CH19. • Modify bits “TFT_GPIO_PU” of P_GPIO_TFT_PU in CH19. • Modify bits “CSI_GPIO_PU” of P_GPIO_CSI_PULL in CH19.	

			• Modify bits "NFLASH_GPIO_PU" of P_GPIO_NFLASH_PULL in CH19. • Modify bits "JTAG_GPIO_PU" of P_GPIO_JTAG_PORT in CH19. • Modify bits "XGPIO_PU" of P_GPIO_GLOBAL_PORT in CH19. • Modify bits "XUSB_PU" of P_GPIO_USB_PORT in CH19. • Modify bits "XUART_PU" of P_GPIO_UART_PORT in CH19. • Modify bits "XI2C_PU" of P_GPIO_I2C_PORT in CH19. • Modify bits "XADC_PU" of P_GPIO_ADC_PORT in CH19. • Modify bits "CDS_PU" of P_GPIO_CDSERVO_PORT in CH19. • Modify bit "XDRAM_PU" of P_GPIO_DRAM_PORT in CH19.	
1.0	12/15/2005	Jackie Chang	Add CD-Servo/MPEG4/DMA function/SFTCFG/BUFCFG…..	
0.2	08/25/2005	Jackie Chang, Joe Chen	Second edition	
0.1	05/30/2005	Wang shu	First edition	

Table of Content

1 Processor Brief

The SPCE3200 is equipped with *S+core® 7*, pronounced as *score-seven*. The S+core® 7 is a 32-bit RISC with Sunplus-owned Instruction Set Architecture (ISA). The ISA has 32/16-bit hybrid instruction mode and parallel conditional execution for high code density, high performance and versatile applications. The micro-architecture includes AMBA bus for SoC integration, coprocessor and custom engine interface for function flexibility, and SJTAG for efficient debugging and In-Circuit Emulation (ICE).

The S+core®7 is a single issue, 7-pipeline stage, high performance and high speed 32-bit RISC. For SPCE3200, S+core®7 can run up to 162 MHz. It supports 4-KB two-way set associative I/D-cache and 4KB LIM/LDM (Local Instruction/Data Memory). The MMU (Memory Management Unit) is also supported for RTOS. We also define two custom engines and three coprocessor interfaces for user-defined function extension. The custom engine supports 32-bit signed/un-signed multiplication and division. The bus interface of the processor is compliant with the AHB Specification (Rev2.0) for easy integration into SoC implementation.

Up to sixty-three prioritized vector interrupts are supported for fast interrupt service. The high performance Sunplus-patented unaligned load/store and pre/post-increment addressing instructions are provided for string copy or memory moving. The powerful bit operation instructions and branch instructions using the counter register as loop iteration counter are provided for control application, and sleep instruction is provided for low-power application.

1.1 Endianess

All data access in SPCE3200 32-bit system are **Little Endian.** The most significant byte in a WORD, 32-bit data, is at the last location of a 4-byte string.

For data read operations, a simple example is shown below for word (32 bits), half-word (16 bits), and byte (8 bits) access:

Memory Address	0x00	0x01	0x02	0x03
Data	0x12	0x34	0x56	0x78

Read word	0x78563412
Read half word	0x3412 | 0x7856
Read byte	0x12 | 0x34 | 0x56 | 0x78

1.2 Key Features

The features of S+core® 7 are listed as follows:

- 32/16-bit Hybrid Execution Mode
- Parallel Conditional Execution
- Capability for Further Software Security Design
- A Harvard Architecture (I-Cache/D-Cache) solution
- MMU (Fixed mode and Full-function mode)
- Compliant with the AMBA Specification (Rev 2.0) for easy integration into SOC implementation
- Vectored Interrupt
- SJTAG (Sunplus JTAG)

2 Debugging in SPCE3200

2.1 Overview

The debug solution in processor includes the following features:

- **Off-chip debug memory access**

  SJTAG allows processor in the debug mode to access the instructions or data that are not physically on the chip. Memory access to the special segment will be transferred to JTAG port and then sent to the debug probe that is controlled by debug host PC. Debug probe will handle the memory access by redirect the access to its local memory. After that, the accessed data will be fed back to processor via JTAG again.

- **Hardware breakpoint**

  Two types of hardware breakpoints are included which can be configured to cause debug exception on:

  - Instruction fetch of a specified address (Breakpoint)
  - Data fetch of a specified address and the access data value (Watchpoint)

- **Software breakpoint instruction**

  Two additional instructions are added to achieve system debug: Software Debug Breakpoint (SDBBP) and Debug Exception Return (DRET).

- **Single step execution**

  A dedicated single step exception in processor is provided for single step debugging.

- **Debug interrupt**

  A debug interrupt is provided to force processor to enter debug mode at any time.

- **DMA Access**

  A DMA channel that is controlled by SJTAG directly accesses system bus through BIU. This feature is quite useful when users need to download code to system memory.

2.2 SJTAG Module

The features of SJTAG module include:

- The hardware breakpoint unit will assert the breakpoint exception request to core.
- A memory controller handles the memory request from core if the address is located in debug segment. If the access is in debug register segment, it will directly access the debug registers such as breakpoint unit register and debug control register. If the

access is in debug memory segment, on the other hand, it will pass the access to debug probe through JTAG interface.

- A JTAG controller handles the communication between probe and SJTAG module.

2.3 RS232

The SPCE3200 still provides a RS232 for debugging purpose. Users should be acquainted with the I/O function printf(). In addition, during the development, a simple command monitor is implemented to allow users to issue commands via the RS232. For example, users can issue a dump command to dump memory content.

2.4 Reference

- S+Core 7 Processor Core Technical Reference Manual (for Software Use) V2.1 – Feb. 01, 2005, by Sunplus Technology Co., Ltd.
- S+Core IDE User Manual, by Sunplus Technology Co., Ltd.

3 S+Core IDE

S+Core IDE, a 32-bit powerful Integrated Development Environment (IDE) for developing application in C or assembler for S+Core series CPU, which owns fully integrated Windows development tool.

3.1 Installation of S+Core IDE

The S+Core IDE can be run on Windows98®, Windows2000®, and WindowsXP®. Follow the on-screen instruction, and S+Core IDE will be installed on your computer. The following two points must be considered:

- You must have administrator privilege when you install S+Core IDE on Win2000/XP.
- The printer-port mode of “SPP” must be set in BIOS setting, and hardware address must be set to 378H as well.

3.2 External Devices for Simulator

In Simulator mode, the S+Core IDE provides two external devices for easy debugging your program.

- **LCD**

  LCD display can be shown during running after LCD is enabled from [Project]->[Setting]->[Simulator]->[Peripheral equipments].

  NOTES:

  - LCD display can be used in simulator mode only.
  - Default LCD is off.

- **UART**

  UART printf can be enabled during running after UART is enabled from [Project]->[Setting]->[Simulator]-[Peripheral equipments]. For more information about UART printf use, please refer to example "UART printf" provided in installation kit.

  NOTES:

  - UART printf can be used in simulator mode only.
  - Default UART is off.

4 System

4.1 Introduction

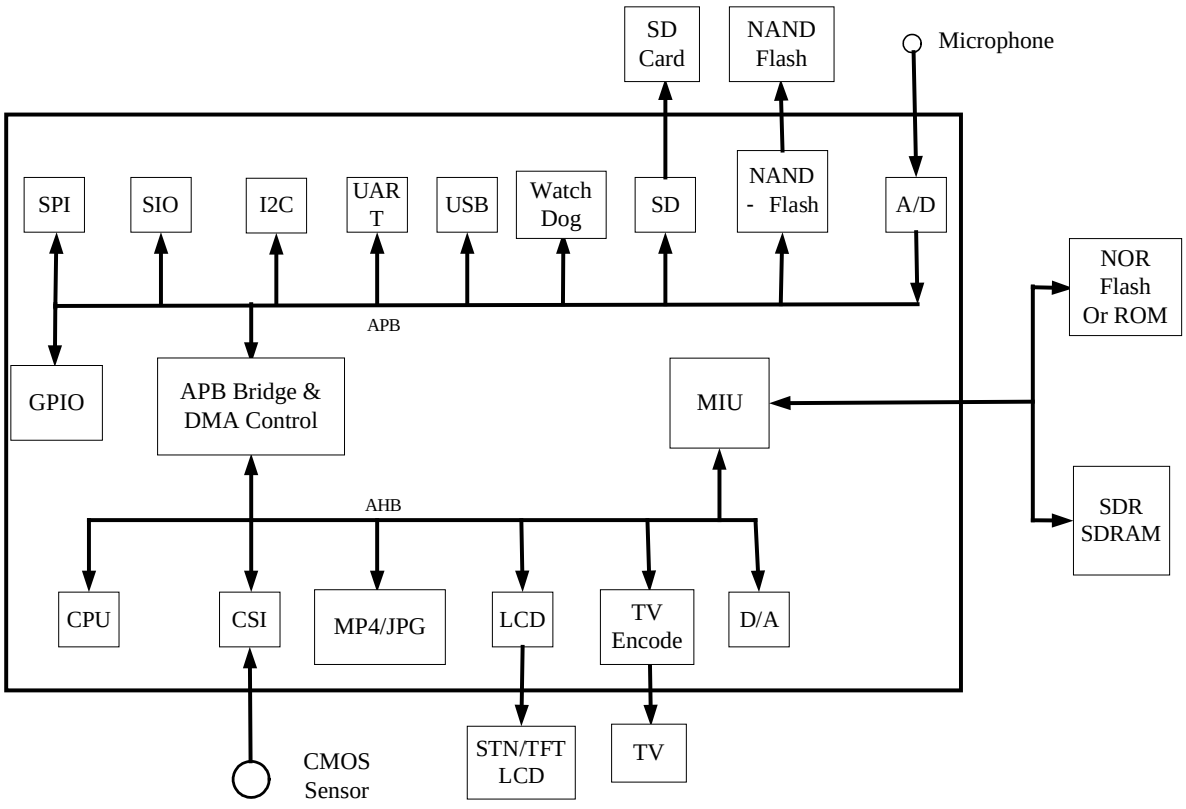
SPCE3200, a new highly integrated System-on-Chip developed by Sunplus, is a designated design for the target fields of audio and video processing and handheld products. Composed of the newest 32-bit Sunplus CPU -- S+core®, SPCE3200 is able to generate graphics and sound for the television system (NTSC or PAL) and LCD. SD card and NAND-type flash are also supported for mass data storage. The operating voltage supply is 3.0V~3.6V and CPU speed is 27~162 MHz. Along with six 16-bit timers, 32768Hz Real Time Clock, Low Voltage Detect, Low Voltage Reset, 12-bit ADC, UART interface, SPI interface, SIO interface, I2C master interface and many other features that facilitate connecting with varieties of I/O devices such as TFT LCD, color STN LCD, CMOS image sensor, TV decoder, Light Pen, and Touch panel, the SPCE3200 provides not only the latest video game technology, but also the full service and support of SUNPLUS.

4.2 Features

- Operating Voltage: I/O VDD is 3.0V ~ 3.6V, core VDD is 1.62V ~ 1.98V
- CPU speed: 27 ~ 162 MHz
- Supports SDR DRAM and DDR DRAM with capacity up to 16M bytes
- Supports 32- and 16-bit SDRAM data buses
- Maximum 128M x32 memory space (512M bytes)
- Supports interlaced/non-interlaced NTSC/PAL composite video output
- Graphic resolution: VGA (640 pixels x 480 pixels) and CIF (320 pixels x 240 pixels)
- Supports 65536 color mode (RGB565 format)
- Programmable 4/16/64/256/32768/65536 color mode
- Supports hardware MPEG-4/JPEG codec
- MPEG-4 frame rate: up to 30 frame/s at CIF resolution
- Provides 4-channel APB DMA: APB peripherals to DRAM or DRAM to APB peripherals
- Provides DRAM DMA: DRAM-to-DRAM data transfer by hardware
- Provides 24 PCM/ADPCM sound channels
- Built-in 4-band digital equalizer
- Two 16-bit high speed DACs for stereo sound quality
- Fully ADSR envelope control
- 3 programmable built-in PLLs to provide system clocks
- 27MHz crystal for NTSC/PAL system
- Real Time Clock (RTC): 32768Hz
- Six 16-bit timers/counters (Programmable and Auto-Reloadable)

- 16 interrupt sources: Timer, Time-base, External Input, Key wakeup
- Key wakeup function
- 8-channel 12-bit AD converter with 9-bit accuracy
- USB 1.1 host or USB 1.1 device
- UART: Universal Asynchronous Receiver and Transmitter
- SPI: Serial Peripheral Interface (master and slave)
- SIO: Sunplus Serial Input/Output Interface
- Built-in Watchdog function
- Light Pen interface
- TFT LCD interface
- CSTN LCD interface
- Supports Sunplus CMOS image sensor interface to connect with Sunplus CMOS image sensor device
- Supports CCIR-601/656 CMOS image sensor / TV decoder interface
- Supports SD card and NAND-type flash for data storage

4.3 Block Diagram



5 PLLS and CKG

5.1 PLLS

There are three PLL sources embedded in SPCE3200, called PLLV, PLLA and PLLU respectively.

- The PLLV generates system clock that is used by the whole system. The output frequency of PLLV can be adjusted through input pins. It can generate a clock with frequency to be the multiple of 27MHz. The maximum frequency in SPCE3200 is 162MHz.
- PLLA generates a 67.7376MHz clock for I2S.
- The PLLU generates clock used specifically by USB Module. PLLU generates a 48MHz clock to USB Module.

5.2 CKG

There is no clock generation circuit inside each module. All clock generation and distribution circuits are inside CKG. The maximum frequency of CPU clock is 162MHz, and the maximum frequency of AHB clock is 108MHz. The clock frequencies of each module are as follows (N=1, 2, 3, 4, 5):

Module	Main clock	AHB master clock	AHB slave clock	APB clock
CPU	PLLV / N	CPU main clock / M		
MIU to CPU			CPU main clock / M	
MIU to other modules	CPU main clock / M		CPU main clock / M	
MP4	MIU main clock / P	CPU main clock / M	CPU main clock / M	
CSI	27 MHz	CPU main clock / M	CPU main clock / M	
TVE	27 MHz	CPU main clock / M	CPU main clock / M	
LCD	27 MHz	CPU main clock / M	CPU main clock / M	
UART	27~67.5 MHz			27MHz
TIMER	32768 Hz			27MHz
WDOG	32768 Hz			27MHz

The table above indicates:

- All AHB slaves (connected to CPU) and AHB masters (connected to MIU) of each module except CPU are running at the same clock frequency with maximum frequency to be 108MHz.
- AHB master clock frequency of Master JPG module is the multiple of core clock frequency. Suppose that the AHB master clock frequency is higher than core clock and the maximum frequency of Core is 54MHz.
- LDMDMA's AHB master is using MIU_CLK, and the LDMDMA's LDM Interface is using CPU_CLK. These two have multiple relations, and also have asynchronous circuit implementation. Suppose that the CPU_CLK's clock frequency is not lower than MIU_CLK.
- Asynchronous circuit implementation is a must among AHB master of CSI, TVE, and LCD. We can suppose that AHB master's clock is not lower than main clock, but these two have no multiple relations. As for connection between AHB slave and core, as AHB slave sets control register's value only, only the enable register class needs to take care of synchronicity.

5.3 Control Registers

5.3.1 P_SLEEP_MODE_CTRL (0x88210000): CPU Clock Configuration

NAME	D6	D5	D4	D3	D2	D1	D0
P_SLEEP_MODE_CTRL	CPUALLRST	CPUWRST	CPUPRST	CPUPLLIRQ	CPUCKMIU	CPUCKIRQ	CPUCK

CPUCK: Temporarily Stop CPUCK for Changing Frequency

0: Enable (Stop)

1: Disable

CPUCKIRQ: Permanently Stop CPUCK until IRQ or Key Wakeup Happens

0: Enable (Stop)

1: Disable

CPUCKMIU: Permanently Stop CPUCK and MIU Clock until IRQ or Key Wakeup Happens

0: Enable (Stop)

1: Disable

CPUPLLIRQ: Permanently Stop CPU and MIU Clock at 0 and then Disable PLL until IRQ or Key Wakeup Happens

0: Disable

1: Enable (Stop)

CPUPRST: Temporarily Reset CPU (to Poweron_n port)

0: Disable

1: Enable (Reset)

CPUWRST: Temporarily Reset CPU (to warmrst_n port)

0: Disable

1: Enable (Reset)

CPUALLRST: Temporarily Reset CPU (to warmrst_n port) and All Other Modules

0: Disable

1: Enable (Reset)

5.3.2 P_CLK_CPU_SEL (0x88210004): CPU Clock Frequency Selection

NAME	D3-D0
P_CLK_CPU_SEL	CPU_CLOCK

CPU_CLOCK: Select CPU clock frequency

0000: Video PLL output divided by 1

0001: Video PLL output divided by 2

0010: Video PLL output divided by 3

0011: Video PLL output divided by 4

0100: Video PLL output divided by 6

0101: Video PLL output divided by 8

0110: Audio PLL output divided by 1

0111: Audio PLL output divided by 2

1000: Audio PLL output divided by 3

1001: Audio PLL output divided by 4

1010: Audio PLL output divided by 6

1011: Audio PLL output divided by 8

5.3.3 P_CLK_AHB_CONF (0x88210008): CPU AHB Clock Config

NAME	D0
P_CLK_AHB_CONF	CPUAHB_CLK_Stop

CPUAHB_CLK_Stop: Temporarily Stop CPUAHB_CLK for Changing Frequency

0: Enable (Stop)

1: Disable

5.3.4 P_CLK_AHB_SEL (0x8821000C): AHB Bus Frequency Selection

NAME	D1-D0
P_CLK_AHB_SEL	CPUAHB_CLK

CPUAHB_CLK: Select AHB Bus Frequency

00: CPU clock divided by 1

01: CPU clock divided by 2

10: CPU clock divided by 3

11: CPU clock divided by 4

5.3.5 P_MIU_CLK_CONF (0x88210010): MIU Clock Config

NAME	D2	D1	D0
P_MIU_CLK_CONF	MIU_RST	MIU_DRAM	MIU_STOP

MIU_STOP: Stop the MIU Clock but Don't Force SDRAM to Enter the Self-Refresh Mode

0: Enable (Stop)

1: Disable

MIU_DRAM: Force DRAM to Enter the Self-Refresh Mode and Stop MIU Clock Until Wakeup Happens

0: Enable (Stop)

1: Disable

MIU_RST: Reset MIU Setting

0: Enable (Reset)

1: Disable

5.3.6 P_CSI_CLK_CONF (0x88210018): CSI Clock Config

NAME	D1	D0
P_CSI_CLK_CONF	CSI_RST	CSI_STOP

CSI_STOP: Stop CSI Clock

0: Enable (Stop)

1: Disable

CSI_RST: Reset CSI Module

0: Enable (Reset)

1: Disable

5.3.7 P_CSI_CLK_SEL (0x8821001C): CSI Clock Selection

NAME	D3	D2	D1-D0
P_CSI_CLK_SEL	CLOCK_FORM	CSI_SOURCE	CSI_CLOCK

CSI_CLOCK: Select the CSI Clock

00: CSI source clock divided by 1

01: CSI source clock divided by 2

10: CSI source clock divided by 4

11: CSI source clock divided by 8

CSI_SOURCE: Select the CSI Source Clock

0: 27MHz clock

1: 24MHz clock

CSI_FORM: Select the Clock Form

0: Select the inverted clock from CMOS sensor as the pixel clock

1: Select the clock from CMOS sensor as the pixel clock

5.3.8 P_JPEG_CLK_CONF (0x88210028): JPG Clock Config

NAME	D1	D0
P_JPEG_CLK_CONF	JPG _RST	JPG _STOP

JPG_STOP: Stop JPG Clock

0: Enable (Stop)

1: Disable

JPG _RST: Reset JPG Module

0: Enable (Reset)

1: Disable

5.3.9 P_JPEG_CLK_SEL (0x8821002C): JPG Clock Frequency Selection

NAME	D3-D0
P_JPEG_CLK_SEL	JPG_CLOCK

JPG_CLOCK: Select JPG clock frequency

0000: Video PLL output divided by 1

0001: Video PLL output divided by 2

0010: Video PLL output divided by 3

0011: Video PLL output divided by 4

0100: Video PLL output divided by 6

0101: Video PLL output divided by 8

0110: Audio PLL output divided by 1

0111: Audio PLL output divided by 2

1000: Audio PLL output divided by 3

1001: Audio PLL output divided by 4

1010: Audio PLL output divided by 6

1011: Audio PLL output divided by 8

5.3.10 P_TV_CLK_CONF (0x88210030): TVE Clock Config

NAME	D1	D0
P_TV_CLK_CONF	TVE_RST	TVE_STOP

TVE_STOP: Stop TVE Clock

0: Enable (Stop)

1: Disable

TVE_RST: Reset TVE Module

0: Enable (Reset)

1: Disable

5.3.11 P_LCD_CLK_CONF (0x88210034): LCD Clock Config

NAME	D1	D0
P_LCD_CLK_CONF	LCD_RST	LCD_STOP

LCD_STOP: Stop LCD Clock

0: Enable (Stop)

1: Disable

LCD_RST: Reset LCD Module

0: Enable (Reset)

1: Disable

5.3.12 P_LCD_CLK_SEL (0x88210038): LCD Clock Frequency Selection

NAME	D5-D3	D2-D0
P_LCD_CLK_SEL	TFT_CLK	STN_CLK

STN_CLK: Select CSTN clock frequency

000: Video PLL output divided by 1

001: Video PLL output divided by 2

010: Video PLL output divided by 3

011: Video PLL output divided by 4

100: Video PLL output divided by 6

101: Video PLL output divided by 8

TFT_CLK: Select TFT clock frequency

000: 27MHz

001: USB PLL output divided by 2

010: USB PLL output divided by 3

011: USB PLL output divided by 4

100: USB PLL output divided by 6

101: SB PLL output divided by 8

5.3.13 P_BLNDMA_CLK_CONF (0x88210050): BLNDMA Clock Config

NAME	D1	D0
P_BLNDMA_CLK_CONF	BLNDMA_RST	BLNDMA_STOP

BLNDMA_STOP: Stop BLNDMA Clock

0: Enable (Stop)

1: Disable

BLNDMA_RST: Reset BLNDMA Module

0: Enable (Reset)

1: Disable

5.3.14 P_DMA_CLK_CONF (0x88210058): APBDMA Clock Config

NAME	D1	D0
P_DMA_CLK_CONF	APBDMA_RST	APBDMA_STOP

APBDMA_STOP: Stop APBDMA Clock

0: Enable (Stop)

1: Disable

APBDMA_RST: Reset APBDMA Module

0: Enable (Reset)

1: Disable

5.3.15 P_UART_CLK_CONF (0x8821005C): UART Clock Config

NAME	D1	D0
P_UART_CLK_CONF	UART_RST	UART_STOP

UART_STOP: Stop UART Clock

0: Enable (Stop)

1: Disable

UART_RST: Reset UART Module

0: Enable (Reset)

1: Disable

5.3.16 P_USB_CLK_CONF (0x88210064): USB Clock Config

NAME	D2	D1	D0
P_USB_CLK_CONF	USB_RST	USBH_STOP	USBD_STOP

USBD_STOP: Stop USB Device Clock

0: Enable (Stop)

1: Disable

USBH_STOP: Stop USB Host Clock

0: Enable (Stop)

1: Disable

USB_RST: Reset USB Module

0: Enable (Reset)

1: Disable

5.3.17 P_TIMER0_CLK_CONF (0x8821006C): Timer0 Clock Config

NAME	D1	D0
P_TIMER0_CLK_CONF	TIMER0_RST	TIMER0_STOP

TIMER0_STOP: Stop Timer0 Clock

0: Enable (Stop)

1: Disable

TIMER0_RST: Reset Timer0 Module

0: Enable (Reset)

1: Disable

5.3.18 P_TIMER1_CLK_CONF (0x88210070): Timer1 Clock Config

NAME	D1	D0
P_TIMER1_CLK_CONF	TIMER1_RST	TIMER1_STOP

TIMER1_STOP: Stop Timer1 Clock

0: Enable (Stop)

1: Disable

TIMER1_RST: Reset Timer1 Module

0: Enable (Reset)

1: Disable

5.3.19 P_TIMER2_CLK_CONF (0x88210074): Timer2 Clock Config

NAME	D1	D0
P_TIMER2_CLK_CONF	TIMER2_RST	TIMER2_STOP

TIMER2_STOP: Stop Timer2 Clock

0: Enable (Stop)

1: Disable

TIMER2_RST: Reset Timer2 Module

0: Enable (Reset)

1: Disable

5.3.20 P_TIMER3_CLK_CONF (0x88210078): Timer3 Clock Config

NAME	D1	D0
P_TIMER3_CLK_CONF	TIMER3_RST	TIMER3_STOP

TIMER3_STOP: Stop Timer3 Clock

0: Enable (Stop)

1: Disable

TIMER3_RST: Reset Timer3 Module

0: Enable (Reset)

1: Disable

5.3.21 P_TIMER4_CLK_CONF (0x8821007C): Timer4 Clock Config

NAME	D1	D0
P_TIMER4_CLK_CONF	TIMER4_RST	TIMER4_STOP

TIMER4_STOP: Stop Timer4 Clock

0: Enable (Stop)

1: Disable

TIMER4_RST: Reset Timer4 Module

0: Enable (Reset)

1: Disable

5.3.22 P_TIMER5_CLK_CONF (0x88210080): Timer5 Clock Config

NAME	D1	D0
P_TIMER5_CLK_CONF	TIMER5_RST	TIMER5_STOP

TIMER5_STOP: Stop Timer5 Clock

0: Enable (Stop)

1: Disable

TIMER5_RST: Reset Timer5 Module

0: Enable (Reset)

1: Disable

5.3.23 P_WDOG_CLK_CONF (0x88210084): Watch Dog Clock Config

NAME	D1	D0
P_WDOG_CLK_CONF	WDOG_RST	WDOG_STOP

WDOG_STOP: Stop Watch Dog Clock

0: Enable (Stop)

1: Disable

WDOG_RST: Reset Watch Dog Module

0: Enable (Reset)

1: Disable

5.3.24 P_RTC_CLK_CONF (0x88210088): RTC Clock Config

NAME	D1	D0
P_RTC_CLK_CONF	RTC_RST	RTC_STOP

RTC_STOP: Stop RTC Clock

0: Enable (Stop)

1: Disable

RTC_RST: Reset RTC Module

0: Enable (Reset)

1: Disable

5.3.25 P_I2S_CLK_CONF (0x8821008C): I2S Clock Config

NAME	D3	D2	D1	D0
P_I2S_CLK_CONF	I2S_RST	I2SM_STOP	I2Sx256_STOP	I2SAPB_STOP

I2SAPB_STOP: Stop I2S APB Clock

0: Enable (Stop)

1: Disable

I2Sx256_STOP: Stop I2S x 256 Clock

0: Enable (Stop)

1: Disable

I2SM_STOP: Stop I2S Master Clock

0: Enable (Stop)

1: Disable

I2S_RST: Reset I2S Module

0: Enable (Reset)

1: Disable

5.3.26 P_I2S_CLK_SEL (0x88210090): I2S Master Clock Divided Setting

NAME	D5-D1	D0
P_I2S_CLK_SEL	DividedNum	I2SM_CLK

I2SM_CLK: Select I2S Master Clock Divided Number

0: Select audio PLL output divided by 2 as the I2S master clock

1: Select audio PLL output divided by 4 as the I2S master clock

DividedNum: Audio PLL output divided by (DividedNum +1), DividedNum is not allowed to be 1

5.3.27 P_I2C_CLK_CONF (0x88210094): I2C Clock Config

NAME	D1	D0
P_I2C_CLK_CONF	I2C_RST	I2C_STOP

I2C_STOP: Stop I2C Clock

0: Enable (Stop)

1: Disable

I2C_RST: Reset I2C Module

0: Enable (Reset)

1: Disable

5.3.28 P_SPI_CLK_CONF (0x88210098): SPI Clock Config

NAME	D1	D0
P_SPI_CLK_CONF	SPI_RST	SPI_STOP

SPI_STOP: Stop SPI Clock

0: Enable (Stop)

1: Disable

SPI_RST: Reset SPI Module

0: Enable (Reset)

1: Disable

5.3.29 P_SIO_CLK_CONF (0x882100A0): SIO Clock Config

NAME	D1	D0
P_SIO_CLK_CONF	SIO_RST	SIO_STOP

SIO_STOP: Stop SIO Clock

0: Enable (Stop)

1: Disable

SIO_RST: Reset SIO Module

0: Enable (Reset)

1: Disable

5.3.30 P_SD_CLK_CONF (0x882100A4): SDCard Clock Config

NAME	D1	D0
P_SD_CLK_CONF	SD_RST	SD_STOP

SD_STOP: Stop SD Clock

0: Enable (Stop)

1: Disable

SD_RST: Reset SD Module

0: Enable (Reset)

1: Disable

5.3.31 P_NAND_CLK_CONF (0x882100A8): NAND-Type Flash Clock Config

NAME	D1	D0
P_NAND_CLK_CONF	FLASH_RST	FLASH_STOP

FLASH_STOP: Stop NAND-type FLASH Clock

0: Enable (Stop)

1: Disable

FLASH_RST: Reset NAND-type FLASH Module

0: Enable (Reset)

1: Disable

5.3.32 P_ADC_CLK_CONF (0x882100AC): ADC Clock Config

NAME	D1	D0
P_ADC_CLK_CONF	ADC_RST	ADC_STOP

ADC_STOP: Stop ADC Clock

0: Enable (Stop)

1: Disable

ADC_RST: Reset ADC Module

0: Enable (Reset)

1: Disable

5.3.33 P_ADC_CLK_SEL (0x882100B0): ADC Source Clock Selection

NAME	D8	D7-D0
P_ADC_CLK_SEL	ADC_Source	DividedNUM

ADC_Source: Select ADC Source Clock

0: Select audio PLL output as the ADC source clock

1: Select USB PLL output as the ADC source clock

DividedNUM: ADC clock is ADC source clock divided by (DividedNUM +1),

DividedNUM is not allowed to be 1

5.3.34 P_CLK_PLLV_CONF (0x882100B4): Video PLL Enable Setting

NAME	D0
P_CLK_PLLV_CONF	PLLV_EN

PLLV_EN: Video PLL Enable

0: Disable

1: Enable

5.3.35 P_CLK_PLLV_SEL (0x882100B8): Video PLL output Clock Frequency Selection

NAME	D3-D0
P_CLK_PLLV_SEL	FREQUENCY

FREQUENCY: Select video PLL output clock frequency

0000: Reserved

0001: Reserved

0011: 81MHz

0100: 87.75MHz

0101: 97.5MHz

0110: 101.25MHz

0111: 108MHz

1000: 114.75MHz

1001: 121.5MHz

1010: 128.25MHz

1011: 135MHz

1100: 141.75MHz

1101: 148.5MHz

1110: 155.25MHz

1111: 162MHz

5.3.36 P_CLK_PLLAU_CONF (0x882100BC): Audio and USB PLL Config

NAME	D2	D1	D0
P_ADC_CLK_CONF	USBPLL_EN	PLLA_CLK	PLLA_EN

PLLA_EN: Audio PLL Enable

0: Disable

1: Enable

PLLA_CLK: Audio PLL Output Clock Selection

0: 73.728MHz

1: 67.6766MHz

USBPLL_EN: USB PLL Enable

0: Disable

1: Enable

5.3.37 P_WAKEUP_KEYC_CLR (0x882100C0): KEYCHG_WAKEUP Clear

NAME	D0
P_WAKEUP_KEYC_CLR	KEYCHG_CLR

KEYCHG_CLR: Clear KEYCHG_WAKEUP register

0: Clear

1: Not clear

5.3.38 P_LVR_RESET_CTRL (0x882100C4): Low Voltage Reset Enable

NAME	D0
P_LVR_RESET_CTRL	LVR_EN

LVR_EN: LVR Enable

0: Disable

1: Enable

5.3.39 P_BUFCTRL_CLK_CONF (0x882100C8): Buffer Control Clock Config

NAME	D1	D0
P_BUFCTRL_CLK_CONF	BUFCTL_RST	BUFCTL_STOP

BUFCTL_STOP: Stop BUFCTL Clock

0: Enable (Stop)

1: Disable

BUFCTL_RST: Reset BUFCTL Module

0: Enable (Reset)

1: Disable

5.3.40 P_LDM_CLK_CONF (0x882100CC): LDMDMA Clock Config

NAME	D1	D0
P_LDM_CLK_CONF	LDMDMA_RST	LDMDMA_STOP

LDMDMA_STOP: Stop LDMDMA Clock

0: Enable (Stop)

1: Disable

LDMDMA_RST: Reset LDMDMA Module

0: Enable (Reset)

1: Disable

5.3.41 P_INT_CLK_CONF (0x882100D0): IRQ Clock Config

NAME	D0
P_INT_CLK_CONF	IRQ_RST

IRQ_RST: Reset IRQCTL Module

0: Enable (Reset)

1: Disable

5.3.42 P_SLEEP_CLK_SEL (0x882100DC): Sleep Mode Clock Selection

NAME	D0
P_SLEEP_CLK_SEL	SLEEP_CLK

SLEEP_CLK: Sleep Mode Clock Selection

0: Select video PLL output divided by 8 as the clock in sleep mode

1: Select 32768Hz clock as the clock in sleep mode

5.3.43 P_TMB_CLK_CONF (0x882100E0): Time Base Clock Config

NAME	D2	D1	D0
P_TMB_CLK_CONF	TBASE_RST	--	TBASE_STOP

TBASE_STOP: Stop TBASE Clock

0: Enable (Stop)

1: Disable

TBASE_RST: Reset TBASE Module

0: Enable (Reset)

1: Disable

5.3.44 P_TIMER_CLK_SEL (0x882100E4): Each Timer Source Clock Selection

NAME	D13	D12	D11	D10	D9	D8	D7-D0
P_TIMER_CLK_SEL	Timer5C	Timer4C	Timer3C	Timer2C	Timer1C	Timer0C	DividedNum

DivideNum: TIMER source clock is 27MHz clock divided by “DivideNum +1”

Timer0C: TIMER0 Source Clock Selection

0: Select TIMER source clock as the clock for TIMER0

1: Select 32768Hz clock as the clock for TIMER0

Timer1C: TIMER1 Source Clock Selection

0: Select TIMER source clock as the clock for TIMER1

1: Select 32768Hz clock as the clock for TIMER1

Timer2C: TIMER2 Source Clock Selection

0: Select TIMER source clock as the clock for TIMER 2

1: Select 32768Hz clock as the clock for TIMER 2

Timer3C: TIMER3 Source Clock Selection

0: Select TIMER source clock as the clock for TIMER 3

1: Select 32768Hz clock as the clock for TIMER 3

Timer4C: TIMER4 Source Clock Selection

0: Select TIMER source clock as the clock for TIMER 4

1: Select 32768Hz clock as the clock for TIMER 4

Timer5C: TIMER5 Source Clock Selection

0: Select TIMER source clock as the clock for TIMER 5

1: Select 32768Hz clock as the clock for TIMER 5

5.3.45 P_WDOG_RESET_STATUS (0x882100E8): Watch Dog Interrupt Status

NAME	D1	D0
P_WDOG_RESET_STATUS	WDOG_INT	WDOG_ERRINT

WDOG_ERRINT: Read 1: Reset caused by Watch Dog error occurs

Write 1: Clear this flag

WDOG_INT: Read 1: Reset caused by Watch Dog occurs

5.3.46 P_MPEG4_CLK_CONF (0x882100EC): MP4 Clock Config

NAME	D2	D1	D0
P_MPEG4_CLK_CONF	MP4_RST	MP4M_STOP	MP4E_STOP

MP4E_STOP: Stop MP4 Extra Clock

0: Enable (Stop)

1: Disable

MP4M_STOP: Stop MP4 Main Clock

0: Enable (Stop)

1: Disable

MP4_RST: Reset MP4 Module

0: Enable (Reset)

1: Disable

5.3.47 P_MPEG4_CLK_SEL (0x882100F0): MP4 Clock Selection

NAME	D1-D0
P_MPEG4_CLK_SEL	MP4_CLK

MP4_CLK: 00: MP4 clock is MIU clock divided by 1

01: MP4 clock is MIU clock divided by 2

10: MP4 clock is MIU clock divided by 3

11: MP4 clock is MIU clock divided by 4

5.3.48 P_USB_DETECT_CTRL (0x882100F4): USB Detection Setting

NAME	D1	D0
P_USB_DETECT_CTRL	USB_CLR	USB_EN

USB_EN: USB Detection Enable Setting

0: Disable

1: Enable

USB_CLR: USB Detection Clear Setting

0: Disable

1: Enable (Clear)

5.3.49 P_GPIO_CLK_CONF (0x882100FC): Software Config Clock Config

NAME	D1	D0
P_GPIO_CLK_CONF	SFTCFG_RST	SFTCFG_STOP

SFTCFG_STOP: Stop SFTCFG Clock

0: Enable (Stop)

1: Disable

SFTCFG_RST: Reset SFTCFG Module

0: Enable (Reset)

1: Disable

5.3.50 P_UART_WAKEUP_STATUS / P_USB_WAKEUP_STATUS (0x88210110): Wakeup Status for USB/ UART

NAME	D1	D0
P_UART_WAKEUP_STATUS P_USB_WAKEUP_STATUS	UART_Wakeup	USB_Wakeup

USB_Wakeup: USB Wakeup Status

0: Doesn't occur

1: Occurs

UART_Wakeup: UART Wakeup Status

0: Doesn't occur

1: Occurs

5.3.51 P_CLK_32K_CONF (0x88210114): Crystal Pad Enable Setting

NAME	D0
P_CLK_32K_CONF	Crystal_En

Crystal_En: 32768Hz Crystal Pad Enable

0: Disable

1: Enable

6 Memory Interface Unit - MIU

MIU supports several different types of external memories, including SDRAM, Parallel ROM, and NAND Flash, to help users manage and design their application systems with more flexible selections of memory chips. In addition, MIU also manages two internal embedded memory blocks, 64K-bit internal SRAM as LDM (**Local Data Memory)** and 256K-bit ROM as embedded **BOOT ROM**.

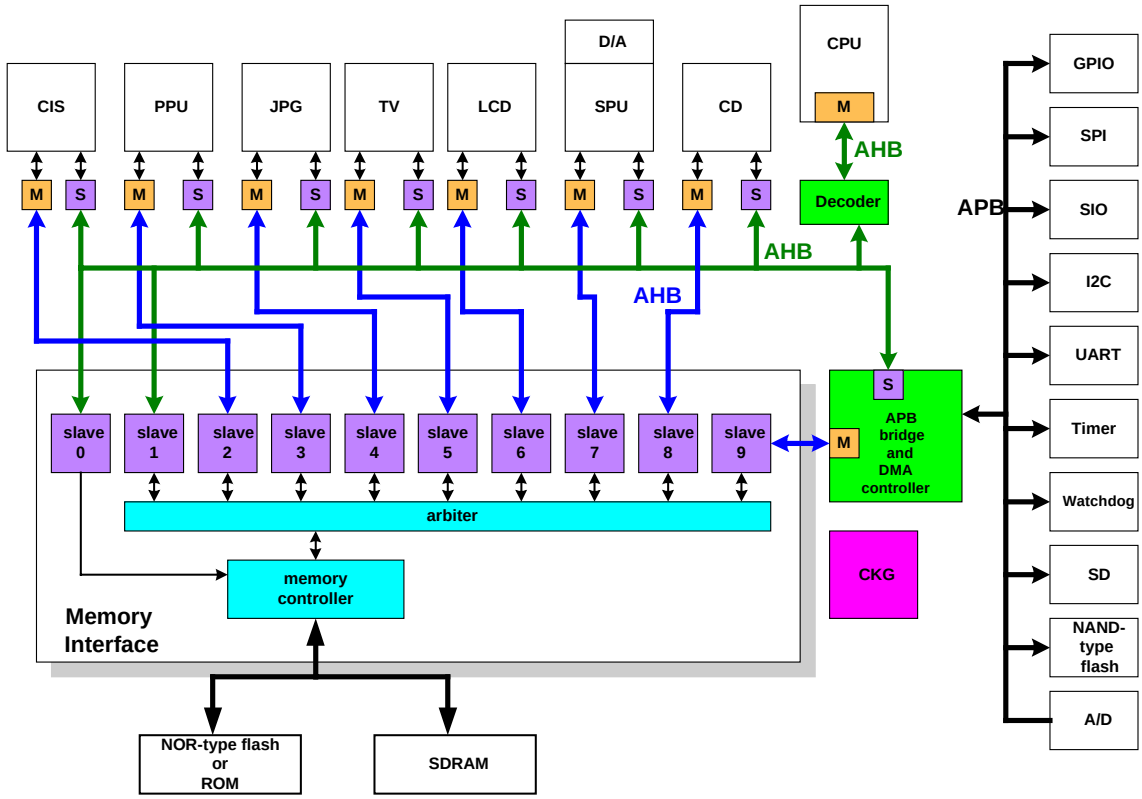


Fig 6-1 MIU Structure

6.2 Memory Mapping

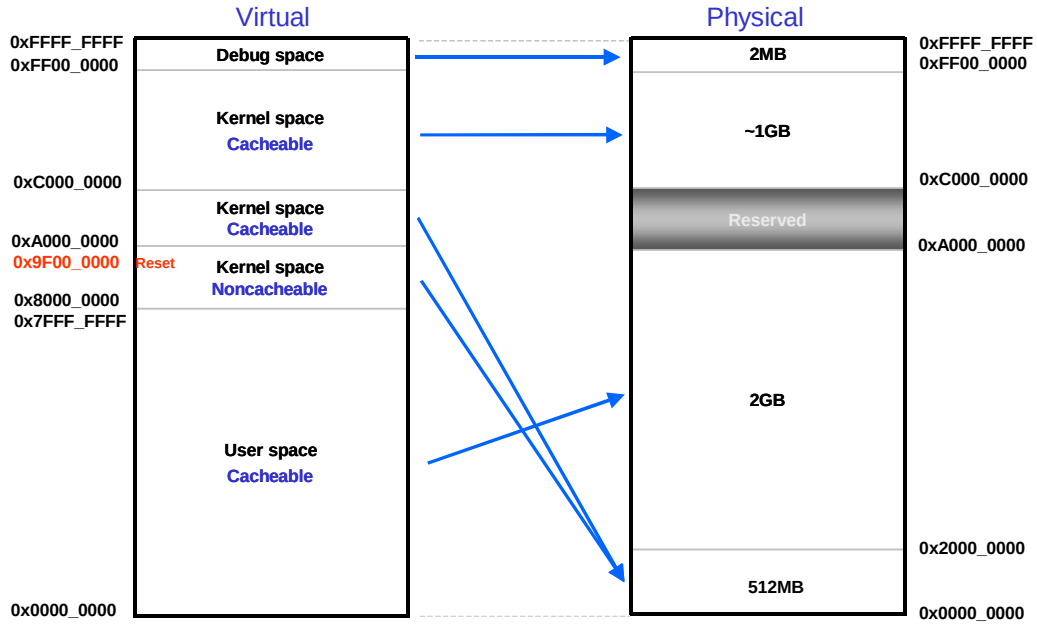


Fig 6-2 Memory Map

The memory map of AHB decoder is as follows:

- Virtual space 0x8000_0000 ~ 0x9FFF_FFFF maps to physical space 0x0000_0000 ~ 0x1FFF_FFFF. When CPU accesses this virtual space, corresponding data are directly read from or written to SRAM without the interference of cache controller.
- Virtual space 0xA000_0000 ~ 0xBFFF_FFFF also maps to physical space 0x0000_0000 ~ 0x1FFF_FFFF. When CPU accesses this virtual space, cache controller controls whether to read (write) corresponding data from (to) either SRAM or other memories.

In addition, S+core processor embeds some memories that are mapped to the physical space 0xA000_0000 ~ 0xC000_0000. See the following table for details.

Name	Address Range	Size
Internal boot ROM	0xB0000000 ~ 0xBFFFFFFF	32k bytes
Internal SRAM	0xA0000000 ~ 0xAFFFFFFF	32k bytes
LDM	Configurable	8k bytes

6.3 System Registers

Each module uses 32K bytes.

Range	Description
0x0800_0000 ~ 0x0800_FFFF	CSI
0x0803_0000 ~ 0x0803_FFFF	TVE
0x0804_0000 ~ 0x0804_FFFF	LCD
0x0807_0000 ~ 0x0807_FFFF	MIU (system register)
0x0808_0000 ~ 0x0808_FFFF	APBDMA (system register)
0x0809_0000 ~ 0x0809_FFFF	BUFCTL
0x080A_0000 ~ 0x080A_FFFF	IRQCTL
0x080C_0000 ~ 0x080C_FFFF	LDMDMA
0x080D_0000 ~ 0x080D_FFFF	BLNDMA
0x080F_0000 ~ 0x080F_FFFF	AHBDEC
0x0810_0000 ~ 0x0810_FFFF	GPIO
0x0811_0000 ~ 0x0811_FFFF	SPI
0x0812_0000 ~ 0x0812_FFFF	SIO
0x0813_0000 ~ 0x0813_FFFF	I2C
0x0814_0000 ~ 0x0814_FFFF	I2S
0x0815_0000 ~ 0x0815_FFFF	UART
0x0816_0000 ~ 0x0816_0FFF	TIMER1
0x0816_1000 ~ 0x0816_1FFF	TIMER2
0x0816_2000 ~ 0x0816_2FFF	TIMER3
0x0816_3000 ~ 0x0816_3FFF	TIMER4
0x0816_4000 ~ 0x0816_4FFF	TIMER5
0x0816_5000 ~ 0x0816_5FFF	TIMER6
0x0816_6000 ~ 0x0816_6FFF	RTC
0x0817_0000 ~ 0x0817_FFFF	WDOG
0x0818_0000 ~ 0x0818_FFFF	SD

Range	Description
0x0819_0000 ~ 0x0819_FFFF	FLASH
0x081A_0000 ~ 0x081A_FFFF	ADC
0x081B_0000 ~ 0x081B_FFFF	USB device
0x081C_0000 ~ 0x081C_FFFF	USB host
0x0820_0000 ~ 0x0820_FFFF	SFTCFG
0x0821_0000 ~ 0x0821_FFFF	CKG
0x0822_0000 ~ 0x0822_FFFF	MP4
0x0823_0000 ~ 0x0823_FFFF	MIU2 (system register)
0x0824_0000 ~ 0x0824_FFFF	ECC

6.4 Control Registers

MIU has some control registers, which is read/written as AHB slave 0 by CPU.

6.4.1 P_TV_BUFFER_SAx: TVE Frame Buffer Start Address

Name	ADDR	D23-D0
P_TV_BUFFER_SA1	0x88070000	TV_START_ADR1
P_TV_BUFFER_SA2	0x88070004	TV_START_ADR2
P_TV_BUFFER_SA3	0x88070008	TV_START_ADR3

TV_START_ADRx: The start address of TVE frame buffer x (x=1, 2, 3)

The D0-D7 must be zero.

6.4.2 P_MIU_SDRAM_POWER (0x8807005C): SDRAM Power Setting

Name	D3	D2	D1	D0
P_MIU_SDRAM_POWER	DEEP	REDO_MRS	POWER_DW	SELF_REF

SELF_REF: Force SDRAM to enter self-refresh mode

0: Disable

1: Enable

POWER_DW: Force SDRAM to enter power-down mode

0: Disable

1: Enable

REDO_MRS: Re-Do MRS

0: Disable

1: Enable

DEEP: Force SDRAM to enter deep power-down mode

0: Disable

1: Enable

6.4.3 P_MIU_SDRAM_SETUP1 (0x88070060): MIU1 SDRAM Setting

Name	D14-D11	D10-D3	D2	D1	D0
P_MIU_SDRAM_SETUP1	RFR_CYCLE_NUM	ATRFR_PERIOD	CAS_LA_CYCLE	tRCD_CYCLE	tRP_CYCLE
Name	**D31**	**D30-D27**	**D26-D25**	**D24-D23**	**D22-D15**
P_MIU_SDRAM_SETUP1	MIU_EN	SDCLK_SEL	SDRAM_RN	SDRAM_CW	SFRFR_PERIOD

tRP_CYCLE: tRP cycles setup

0: tRP is 1 cycle

1: tRP is 2 cycles (Default)

tRCD_CYCLE: tRCD cycles setup

0: tRCD is 1 cycle

1: tRCD is 2 cycles (Default)

CAS_LA_CYCLE: Cas latency cycles setup

0: Cas Latency is 2 cycles (Default)

1: Cas Latency is 3 cycles

ATRFR_PERIOD: An auto-refresh is issued every (ATRFR_PERIOD x 16) cycles

RFR_CYCLE_NUM: Cycles needed for an auto-refresh

SFRFR_PERIOD: If ADG_REQ doesn't occur for (SFRFR_PERIOD x 256 x ATRFR_PERIOD x 16) cycles, a hardware-triggered self-refresh is issued

SDRAM_CW: SDRAM Col Width

00: 8-bit word SDRAM is used

01: 16-bit word SDRAM is used

10: 32-bit word SDRAM is used

11: Reserved

SDRAM_RN: SDRAM Row Number

00: 1024 rows per bank

01: 2048 rows per bank

10: 4096 rows per bank

11: 8192 rows per bank

SDCLK_SEL: SDRAM clock selection

Select the phase delay of SDRAM_CLK

MIU_EN: MIU enable setup

0: Disable

1: Enable

6.4.4 P_MIU_SDRAM_STATUS (0x8807006C): MIU1 SDRAM Status

Name	D0
P_MIU_SDRAM_STATUS	MIU_STATUS

MIU_STATUS: Status of MIU1 SDRAM

0: No status occurs

1: SDRAM is in the self-refresh mode or power-down mode

6.4.5 P_MPEG4_RAWBUFFER_SAx: MP4 RAW Image Frame Buffer Start Address

Name	ADDR	D23-D0
P_MPEG4_RAWBUFFER_SA1	0x88070070	MP4RAW_START_ADR1
P_MPEG4_RAWBUFFER_SA2	0x88070074	MP4RAW_START_ADR2
P_MPEG4_RAWBUFFER_SA3	0x88070078	MP4RAW_START_ADR3

MP4RAW_START_ADRx: Start address of MP4 raw image frame buffer x (x=1, 2, 3)

The D0-D7 must be zero.

6.4.6 P_MPEG4_WRITEBUFFER_SAx: MP4 Write Frame Buffer Start Address

Name	ADDR	D23-D0
P_MPEG4_WRITEBUFFER_SA1	0x8807007C	MP4W_START_ADR1
P_MPEG4_WRITEBUFFER_SA2	0x88070080	MP4W_START_ADR2
P_MPEG4_WRITEBUFFER_SA3	0x88070084	MP4W_START_ADR3

MP4W_START_ADRx: Start address of MP4 write frame buffer x

The D0-D7 must be zero.

6.4.7 P_MPEG4_VLCBUFFER_SAx: MP4 VLC Buffer Start Address

Name	ADDR	D23- D0
P_MPEG4_VLCBUFFER_SA1	0x88070088	MP4V_START_ADR1
P_MPEG4_VLCBUFFER_SA2	0x8807008C	MP4V_START_ADR2

MP4V_START_ADRx: Start address of MP4 VLC buffer x

The D0-D9 must be zero.

6.4.8 P_MPEG4_FRAMEBUFFER_HSIZE (0x88070090): MP4 Frame Buffer H-Size Setting

Name	D9-D2	D1-D0
P_MPEG4_FRAMEBUFFER_HSIZE	MP4_FB_HSIZE	

MP4_FB_HSIZE: MP4 Frame Buffer H-Size

The number of pixels on a scan line.

6.4.9 P_MIU_SDRAM_SETUP2 (0x88070094): tRP &tRCD Cycle Setup for SDRAM

Name	D1	D0
P_MIU_SDRAM_SETUP2	tRCD_CYCLE	tRP_CYCLE

tRP_CYCLE: tRP cycles setup

0: See MIU1_SDRAM_SETTING[0]

1: tRP is 3 cycles

tRCD_CYCLE: tRCD cycles setup

0: See MIU1_SDRAM_SETTING[1]

1: tRCD is 3 cycles

6.5 Arbiter

Memory request priority is TVE > LCD > CSI, and priorities of other modules are determined by round-robin.

7 Buffer Control - BUFCTL

Buffer control unit is a comprehensive hardware logic to coordinate all double or triple buffers in CSI unit, TV encode unit and LCD unit. This will considerably simplify the program flow and reduce the possible timing or interrupt function overhead problems in the system.

7.1 Control Registers

7.1.1 P_CSI_BUFFER_SEL (R/W)(0x88090008): Software Buffer Pointer for CSI

Name	D1-D0
P_CSI_BUFFER_SEL	CSI_PTR

CSI_PTR: Write this register to set the active frame buffer from 0 to 2 of CSI

Read this register to get the active frame buffer from 0 to 2 of CSI

0: CSI frame buffer 0

1: CSI frame buffer 1

2: CSI frame buffer 2

7.1.2 P_TV_BUFFER_SEL (R/W)(0x88090020): Software Buffer Pointer for TVE

Name	D1-D0
P_TV_BUFFER_SEL	TVE_PTR

TVE_PTR: Write this register to set the active frame buffer from 0 to 2 of TVE

Read this register to get the active frame buffer from 0 to 2 of TVE

0: TVE frame buffer0

1: TVE frame buffer1

2: TVE frame buffer2

7.1.3 P_LCD_BUFFER_SEL (R/W)(0x88090024): Software Buffer Pointer for LCD

Name	D1-D0
P_LCD_BUFFER_SEL	LCD_PTR

LCD_PTR: Write this register to set the active frame buffer from 0 to 2 of LCD

Read this register to get the active frame buffer from 0 to 2 of LCD

0: LCD frame buffer0

1: LCD frame buffer1

2: LCD frame buffer2

7.1.4 P_BUFCTRL_FRAMELOSS_INT (R/W)(0x88090028): Buffer Control Status

Name	D0
P_BUFCTRL_FRAMELOSS_INT	CSI_Floss

CSI_Floss: CSI frame loss interrupt flag

Write 0: Clear the interrupt flag

7.1.5 P_TV_FRAME_COUNT (R/W)(0x88090034): TVE Frame Counter

Name	D7-D0
P_TV_FRAME_COUNT	FrameCounter

FrameCounter: TVE Frame Counter

Write any value to reset the counter to 0.

7.1.6 P_MPEG4_RAWBUFFER_SEL (R/W)(0x88090044): MP4 RAW Image Buffer Pointer

Name	D1-D0
P_MPEG4_RAWBUFFER_SEL	MP4RAW_BUF

MP4RAW_BUF: Software buffer pointer for MP4 RAW image buffer

7.1.7 P_MPEG4_WRITEBUFFER_SEL (R/W)(0x88090048): MP4 Re-Constructed Buffer Pointer

Name	D1-D0
P_MPEG4_WRITEBUFFER_SEL	MP4W_BUF

MP4W_BUF: Software buffer pointer for MP4 re-constructed buffer

7.1.8 P_BUFCTRL_INT_CTRL (R/W)(0x8809004C): Buffer Controller Interrupt Mask

Name	D0
P_BUFCTRL_INT_CTRL	BUFCTL_SET

BUFCTL_SET: Mask Buffer Control Interrupt to CPU

0: Disable

1: Enable (Mask)

7.1.9 P_MPEG4_REFBUFFER_SEL (R/W)(0x88090050): MP4 Reference Buffer Pointer

Name	D1-D0
P_MPEG4_REFBUFFER_SEL	MP4R_BUF

MP4R_BUF: Software buffer pointer for MP4 reference buffer

7.1.10 P_MPEG4_VLCBUFFER_SEL (R/W)(0x88090054): MP4 VLC Buffer Pointer

Name	D0
P_MPEG4_VLCBUFFER_SEL	MP4V_BUF

MP4V_BUF: Software buffer pointer for MP4 VLC buffer

7.1.11 P_MPEG4_ENCODE_FRAMENUM (R/W)(0x8809005C): MP4 Encode Base on TVE Frame Number

Name	D1-D0
P_MPEG4_ENCODE_FRAMENUM	FRAMENUM

FRAMENUM: Use (N+1) times of TVE frames to encode 1 MP4 frame

8 Display Unit – TV Encoder

8.1 Introduction

The display unit, TV encoder unit in SPCE3200 is a multi-TV system and a multi-screen mode TV encoder with 9-bit video DAC to generate composite video signal to TV screen at VGA resolution.

8.2 TV System and Screen Mode

The display unit can support several TV systems with different screen modes as the list below shows.

NTSC		PAL	
Interlaced	**Non-Interlaced**	**Interlaced**	**Non-Interlaced**
640 x 480	640 x 240	640 x 480	640 x 240
320 x 240	320 x 240	320 x 240	320 x 240
630x240	640x240	640x240	640x240

PAL system includes PAL-(B, D, G, H, I, N) modes.

8.3 Display

SPCE3200 can display arbitrary rectangle area within the whole external Dynamic RAM area by setting the left-top pointer of the frame buffer as the frame buffer start address.

8.4 TVE Registers

SPCE3200 TV encoder unit (TVE) provides two digital luminance filters, a low pass filter, an edge enhancement filter, and an all pass filter to let users adjust the best TV signal output quality depending on their content's characteristic. For example, when displaying true color digital photos, user can enable the low pass filter to let the image on TV screen look much smoother. On the contrary, the edge enhancement filter will be more suitable for the computer graphics. The all pass filter will bypass all digital filters, so users can place their own analog filter circuit on system PCB to adjust their ideal TV output signal quality. In addition, the TV encoder unit can be disabled by programming to save the system power consumption whenever the TV composite output is not required, especially for handheld applications.

8.4.1 P_TV_MODE_CTRL (0x88030000)

Name	D12	D11	D10	D9	D8	D7-D6	D5-D4	D3-D2	D1	D0
P_TV_MODE_CTRL	TVEN	YCBCRMODE	RGBMODE	ENDIAN	MODE	NTSCTYPE		Resolution	PAL	Intrlce

Intrlce: Interlaced Mode Selection

0: Non-Interlaced mode

1: Interlaced mode

PAL: NTSC/PAL Mode Selection

0: NTSC Mode

1: PAL Mode

Resolution: TV Resolution Selection

0: QVGA (320x240)

1: VGA (640x480)

2: HVGA (640x240)

3: QVGA (320x240)

NTSCTYPE: NTSC Type Selection

0: NTSC

1: NTSCJ

2: NTSC443

3: NTSC

MODE: Frame Buffer Format Selection

0: YCbCr (YCbCr422 or 4Y4U4Y4V)

1: RGB (RGB565 or ARGB1555)

ENDIAN: Endian Mode Selection

0: Little Endian

1: Big Endian

RGBMODE: Frame Buffer RGB Format Selection

0: RGB565

1: ARGB1555

YCBCRMODE: Frame Buffer YCbCr Format Selection

0: YCbCr422

1: 4Y4U4Y4V for MPEG4

TVEN: TV Encoder Enable

0: Disable

1: Enable

8.4.2 P_TV_SATURATION_SETUP(0x88030004) & P_TV_HUE_SETUP (0x88030008)

Saturation & Hue level of TV encoder can be adjusted by programming the following two registers. Generally, these two registers are initialized automatically when SPCE3200 is booted. Be sure to leave them unchanged when running normal applications.

Name	D7-D0
P_TV_SATURATION_SETUP	Saturation adjustment
P_TV_HUE_SETUP	Hue adjustment

8.4.3 P_TV_FADE_SETUP (0x8803000C): Fade Effect Control Register

Fade effect provides the fade-in and fade-out control. It's controlled by the register at 0x8803000C. The Lumince_Adjust value defines the fade-out level. The following diagrams are the example of the fade effect. The right side diagram has the smaller Lumince_Adjust value than those in the left side.

Name	D7-D0
P_TV_FADE_SETUP	Lumince_Adjust

8.4.4 P_TV_FILTER_SEL (0x88030010): Low Pass Filter Select Control Register

Name	D7-D6	D5-D4	D3-D2	D1-D0
P_TV_FILTER_SEL	UPS_TYPE	UV_TYPE	YUPS_TYPE	Y_TYPE

UPS_TYPE: U/V UPS Filter Type

UV_TYPE: U/V Filter Type

YUPS_TYPE: Y UPS Filter Type

Y_TYPE: Y Filter Type

8.4.5 Light Gun Interface

Light Gun Interface is mainly used for latching the luster position. The position of the luster can be read from 0x88030030 and 0x88030034 (player one) and 0x88030038 and 0x8803003C (player two). These two ports are for read only; write operation is invalid.

P_TV_LIGHTGUN_CTRL (0x8803002C)

P_TV_LIGHTGUN0_X (0x88030030)

P_TV_LIGHTGUN0_Y (0x88030034)

P_TV_LIGHTGUN1_X (0x88030038)

P_TV_LIGHTGUN1_Y (0x8803003C)

Name	D0
P_TV_LIGHTGUN_CTRL	Latch1st

Latch1st: Latch the First Pulse

0: Latch each pulse in a frame

1: Latch the first pulse only in a frame

Name	D9-D0
P_TV_LIGHTGUN0_CTRL	LightGun0HPosition
P_TV_LIGHTGUN0_CTRL	LightGun0VPosition
P_TV_LIGHTGUN1_CTRL	LightGun1HPosition
P_TV_LIGHTGUN1_CTRL	LightGun1VPosition

9 APBDMA

9.1 Architecture

APB Bridge and DMA Controller architecture is shown below.

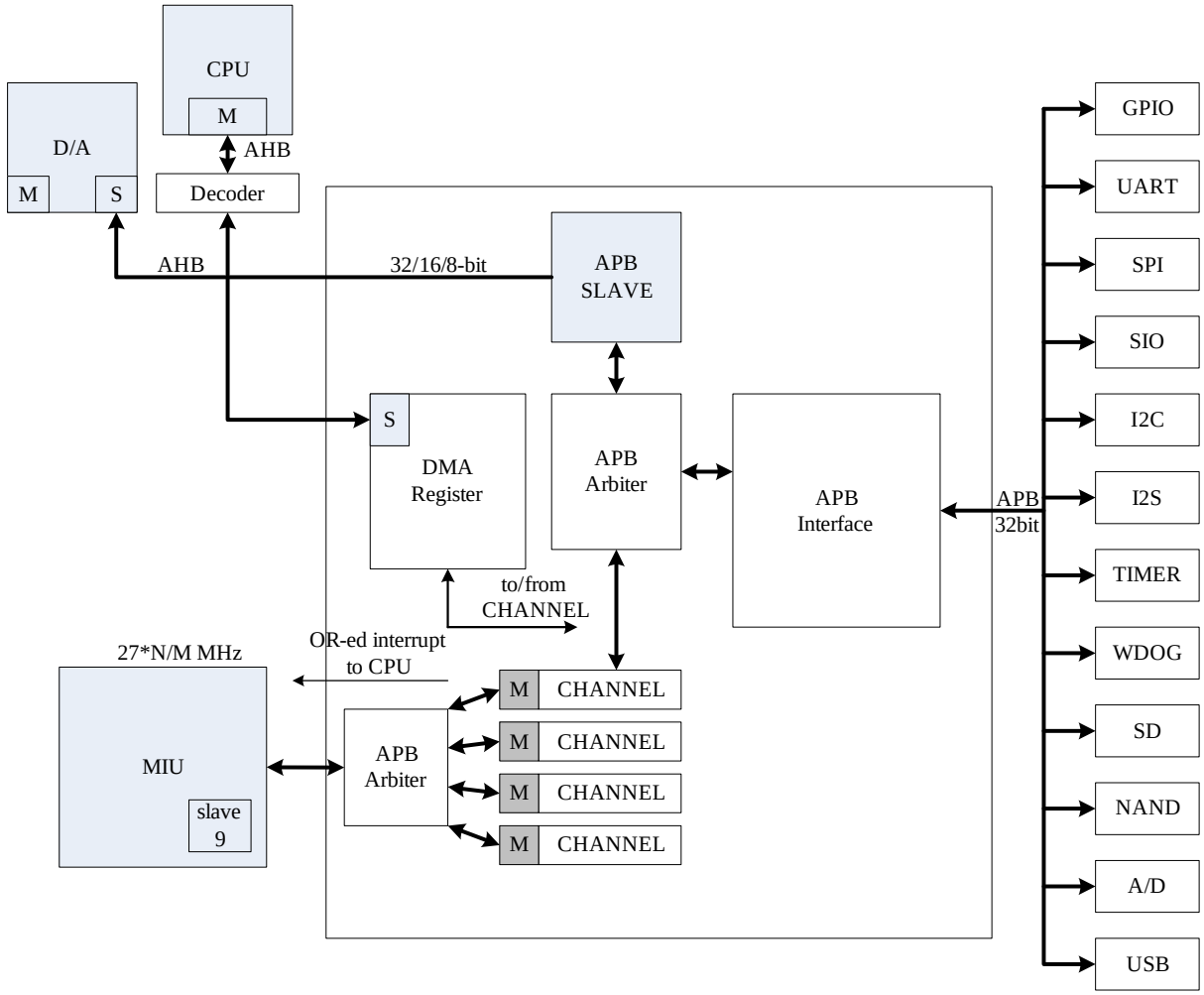


Fig 9-1 APBDMA Architecture

9.2 APB Bridge

CPU can read and write each APB peripheral module through APB Bridge.

9.3 DMA Controller

DMA controller can offer four channels to R/W MIU memory for APB peripheral modules at the same time; each channel can be established into the following four transmission

modes:

- 8-bit single transfer
- 16-bit single transfer
- 32-bit single transfer
- 32-bit burst transfer

DMA has two starting modes:

(1) When channel enable bit is set to 1, DMA controller begins to read or write the data successively. It is terminated when the operation done.

(2) When channel enable bit is set to 1 and meanwhile APB peripheral modules send a REQ signal, DMA controller carries out Read or Write operation once. When all operations corresponding to these REQ signals are done, it is terminated.

The space planning of Memory has two ways when DMA R/W MIU:

(1) Single buffering. By specifying the start and end DMA address, DMA controller only reads or writes this memory space, and it is terminated by sending an IRQ signal when read or write operation completes.

(2) Double buffering. By specifying the start and end address of BUFFER A and BUFFER B at the same time, DMA will read or write BUFFERA and BUFFERB alternately. It will generate an IRQ signal when either transfer completes. Setting channel enable bit back to '0' can generate a DMA operation cease request, and DMA controller will terminate its operation when the current block transfer completes.

For the DMA-to-APB read/write operation, there are two addressing modes to go to related PORT:

(1) Regular address

(2) Continuous address

9.4 Control Registers

9.4.1 P_DMA_BUSY_STATUS (0x88080000): Channel Busy Status

NAME	D3	D2	D1	D0
P_DMA_BUSY_STATUS	CH3_BUSY	CH2_BUSY	CH1_BUSY	CH0_BUSY

CH0_BUSY: Channel 0 Busy Status

CH1_BUSY: Channel 1 Busy Status

CH2_BUSY: Channel 2 Busy Status

CH3_BUSY: Channel 3 Busy Status

9.4.2 P_DMA_INT_STATUS (0x88080004): Channel Interrupt Status

NAME	D3	D2	D1	D0
P_DMA_INT_STATUS	CH3_IRQ	CH2_IRQ	CH1_IRQ	CH0_IRQ

CH0_IRQ: Channel 0 IRQ Status

Read 0: DMA IRQ does not occur
Read 1: DMA IRQ occurs
Write 0: No operation
Write 1: Clear DMA IRQ

CH1_IRQ: Channel 1 IRQ Status

Read 0: DMA IRQ does not occur
Read 1: DMA IRQ occurs
Write 0: No operation
Write 1: Clear DMA IRQ

CH2_IRQ: Channel 2 IRQ Status

Read 0: DMA IRQ does not occur
Read 1: DMA IRQ occurs
Write 0: No operation
Write 1: Clear DMA IRQ

CH3_IRQ: Channel 3 IRQ Status

Read 0: DMA IRQ does not occur
Read 1: DMA IRQ occurs
Write 0: No operation
Write 1: Clear DMA IRQ

9.4.3 P_DMA_AHB_SAxBA: Buffer A Start Address

NAME	ADDRESS	D31-D0
P_DMA_AHB_SA0BA	88080008H	BUFFER A Start Address for Channel1
P_DMA_AHB_SA1BA	8808000CH	BUFFER A Start Address for Channel2
P_DMA_AHB__SA2BA	88080010H	BUFFER A Start Address for Channel3
P_DMA_AHB_SA3BA	88080014H	BUFFER A Start Address for Channel4

9.4.4 P_DMA_AHB_EAxBA: Buffer A End Address

NAME	ADDRESS	D31-D0
P_DMA_AHB_EA0BA	88080018H	BUFFER A End Address for Channel1
P_DMA_AHB_EA1BA	8808001CH	BUFFER A End Address for Channel2
P_DMA_AHB_EA2BA	88080020H	BUFFER A End Address for Channel3
P_DMA_AHB_EA3BA	88080024H	BUFFER A End Address for Channel4

9.4.5 P_DMA_APB_SAx: APB Module Access Port Start Address

NAME	ADDRESS	D31-D0
P_DMA_APB_SA0	88080028H	APB Module Access Port Start Address for Channel1
P_DMA_APB_SA1	8808002CH	APB Module Access Port Start Address for Channel2
P_DMA_APB_SA2	88080030H	APB Module Access Port Start Address for Channel3
P_DMA_APB_SA3	88080034H	APB Module Access Port Start Address for Channel4

9.4.6 P_DMA_AHB_SAxBB: Buffer B Start Address

NAME	ADDRESS	D31-D0
P_DMA_AHB_SA0BB	8808004CH	BUFFER B Start Address for Channel1
P_DMA_AHB_SA1BB	88080050H	BUFFER B Start Address for Channel2
P_DMA_AHB_SA2BB	88080054H	BUFFER B Start Address for Channel3
P_DMA_AHB_SA3BB	88080058H	BUFFER B Start Address for Channel4

9.4.7 P_DMA_AHB_EAxBB: Buffer B End Address

NAME	ADDRESS	D31-D0
P_DMA_AHB_EA0BB	8808005CH	BUFFER B End Address for Channel1
P_DMA_AHB_EA1BB	88080060H	BUFFER B End Address for Channel2
P_DMA_AHB_EA2BB	88080064H	BUFFER B End Address for Channel3
P_DMA_AHB_EA3BB	88080068H	BUFFER B End Address for Channel4

9.4.8 P_DMA_CHANNEL0_CTRL (0x8808006C): Channel One Setting

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_DMA_CHANNEL0_CTRL	CH1_EN	CH1_IRQ	CH1_TRANS		CH1_MEM	CH1_MODE	CH1_DMA	CH1_DIR

CH1_DIR: Channel1 direction for R/W

0: MIU to APB

1: APB to MIU

CH1_DMA: Channel1 DMA Mode selection

0: Auto mode

1: Polling mode

CH1_MODE: Channel1 DMA's APB peripheral Modules location mode

0: Continuous mode

1: Regular mode

CH1_MEM: Channel1 MIU Memory mode

0: Single Buffer

1: Double Buffer

CH1_TRANS: Channel1 transfer mode

00: 8 bits single transfer

01: 16 bits single transfer

10: 32 bits single transfer

11: 32 bits burst transfer

CH1_IRQ: Channel1 IRQ Mask

0: Disable DMA IRQ

1: Enable DMA IRQ

CH1_EN: Channel1 Enable

0: Disable DMA

1: Enable DMA

9.4.9 P_DMA_CHANNEL1_CTRL (0x88080070): Channel Two Setting

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_DMA_CHANNEL1_CTRL	CH2_EN	CH2_IRQ	CH2_TRANS		CH2_MEM	CH2_MODE	CH2_DMA	CH2_DIR

CH2_DIR: Channel2 direction for R/W

0: MIU to APB

1: APB to MIU

CH2_DMA: Channel2 DMA Mode Selection

0: Auto mode

1: Polling mode

CH2_MODE: Channel2 DMA's APB peripheral Modules location mode

0: Continuous mode

1: Regular mode

CH2_MEM: Channel2 MIU Memory mode

0: Single Buffer

1: Double Buffer

CH2_TRANS: Channel2 transfer mode

00: 8 bits single transfer

01: 16 bits single transfer

10: 32 bits single transfer

11: 32 bits burst transfer

CH2_IRQ: Channel2 IRQ Mask

0: Disable DMA IRQ

1: Enable DMA IRQ

CH2_EN: Channel2 Enable

0: Disable DMA

1: Enable DMA

9.4.10 P_DMA_CHANNEL2_CTRL (0x88080074): Channel Three Setting

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_DMA_CHANNEL 2_CTRL	CH3_EN	CH3_IRQ	CH3_TRANS		CH3_MEM	CH3_MODE	CH3_DMA	CH3_DIR

CH3_DIR: Channel3 direction for R/W

0: MIU to APB

1: APB to MIU

CH3_DMA: Channel3 DMA Mode Selection

0: Auto mode

1: Polling mode

CH3_MODE: Channel3 DMA's APB peripheral Modules location mode

0: Continuous mode

1: Regular mode

CH3_MEM: Channel3 MIU Memory mode

0: Single Buffer

1: Double Buffer

CH3_TRANS: Channel3 transfer mode

00: 8 bits single transfer

01: 16 bits single transfer

10: 32 bits single transfer

11: 32 bits burst transfer

CH3_IRQ: Channel3 IRQ Mask

0: Disable DMA IRQ

1: Enable DMA IRQ

CH3_EN: Channel3 Enable

0: Disable DMA

1: Enable DMA

9.4.11 P_DMA_CHANNEL3_CTRL (0x88080078): Channel Four Setting

Name	D7	D6	D5	D4	D3	D2	D1	D0
P_DMA_CHANNEL3_CTRL	CH4_EN	CH4_IRQ	CH4_TRANS		CH4_MEM	CH4_MODE	CH4_DMA	CH4_DIR

CH4_DIR: Channel4 direction for R/W

0: MIU to APB

1: APB to MIU

CH4_DMA: Channel4 DMA Mode Selection

0: Auto mode

1: Polling mode

CH4_MODE: Channel4 DMA's APB peripheral Modules location mode

0: Continuous mode

1: Regular mode

CH4_MEM: Channel4 MIU Memory mode

0: Single Buffer

1: Double Buffer

CH4_TRANS: Channel4 transfer mode

00: 8 bits single transfer

01: 16 bits single transfer

10: 32 bits single transfer

11: 32 bits burst transfer

CH4_IRQ: Channel4 IRQ Mask

0: Disable DMA IRQ

1: Enable DMA IRQ

CH4_EN: Channel4 Enable

0: Disable DMA

1: Enable DMA

9.4.12 P_DMA_CHANNEL_RESET (0x8808007C): Channel Software Reset

NAME	D3	D2	D1	D0
P_DMA_CHANNEL_RESET	CH3_REST	CH2_REST	CH1_REST	CH0_REST

CH0_REST: Channel 0 Software Reset

0: Disable

1: Enable

CH1_REST: Channel 1 Software Reset

0: Disable

1: Enable

CH2_REST: Channel2 Software Reset

0: Disable

1: Enable

CH3_REST: Channel 3 Software Reset

0: Disable

1: Enable

9.5 DMA END

When DMA finishes, users need to write ‘1’ to the corresponding IRQ STATUS bit to clear DMA interrupt and write ‘0’ to the corresponding CHANNEL ENABLE to terminate the operation.

10 BITBLT DMA with Alpha Blending Operation - BLNDMA

10.1 Features

"BLNDMA" is the abbreviation of the "Blending and DMA controller". Functions of the BLNDMA controller are depicted below.

- DRAM-to-DRAM DMA (BitBlt) function
- DMA Pattern Fill function
- Transparent filter (Color Key) function
- Blending function
- Clipping function
- YUV2RGB conversion function

10.1.1 DRAM-to-DRAM DMA (BitBlt) Function

The BLNDMA Controller supports a DRAM-to-DRAM DMA function, which performs data transfer from source address to destination address in the SDRAM. Source address, destination address, and the number of word to be transferred must be programmed before transfer. The transfer is word-aligned. One word includes two pixel data.

The BLNDMA Controller supports both linear address mode and block address mode. If it selects the block address mode to perform DMA function, it is just BitBlt (BIT Block Transfer) function. The BitBlt function may be combined with color key, blending, and clipping function. The BitBlt background width size could be 256/320/512/640/1024/2048 pixels. The BitBlt background height size could be 240/256/480/512/1024/2048 pixels. The background width and height must be programmed before performing the BitBlt function. If it performs the DRAM-to-DRAM DMA function, however, the number of word to be transferred is determined by the "DMA_TRANS_WIDTH" and "DMA_TRANS_HEIGHT" for the register address 0x880D000C. Either field has 12 bits; that is, the maximum transfer size is 4096x4096 words. The "DMA_TRANS_WIDTH" must be the multiple of 8 words.

10.1.2 DMA Pattern Fill Function

Pattern Fill function is to fill a rectangle with constant color value. If it performs the DMA Pattern Fill function, Pattern is decided by the 32-bit register "DMA_FILL_PAT" for the register address 0x880D0010.

10.1.3 Transparent Filter (Color Key) Function

If it performs the DRAM-to-DRAM DMA with the transparent filter function (transparent BitBlt), the transparent filter pattern is decided by the 16-bit register "DMA_FILTER_PAT" for the register address 0x880D0020. If the transferred data is matched with the transparent filter pattern, it won't be written into the destination address in the SDRAM.

10.1.4 Blending Function

The Blending function is based on the two constant alpha values (SrcA_Factor and SrcB_Factor). The alpha value is decided by the 6-bit register "SrcA_Factor and SrcB_Factor" for the register address 0x880D001C. There are two ways to perform the blending function and make the overlapping effect.

(1) Blend two source bitmaps (SrcA and SrcB) and then write into the destination bitmap.

Dst.R = round((SrcA.R * SrcA_Factor ± SrcB.R * SrcB_Factor) / 64)

Dst.G = round((SrcA.G * SrcA_Factor ± SrcB.G * SrcB_Factor) / 64)

Dst.B = round((SrcA.B * SrcA_Factor ± SrcB.B * SrcB_Factor) / 64)

(2) Blend one source bitmap (SrcA) and the destination bitmap (Dst) and then write back to the destination bitmap.

Dst.R = round((SrcA.R * SrcA_Factor ± Dst. R * SrcB_Factor) / 64)

Dst.G = round((SrcA.G * SrcA_Factor ± Dst.G * SrcB_Factor) / 64)

Dst.B = round((SrcA.B * SrcA_Factor ± Dst.B * SrcB_Factor) / 64)

10.1.5 Clipping Function

The clipping function clips the copied source bitmap according to the BitBlt background width and height. The clipping function can be performed only in the block address mode; that is, there is no clipping function in the linear address mode.

10.1.6 YUV2RGB Conversion Function

The YUV2RGB function is to convert the image from YUV422 format into the RGB555 or ARGB1555 format. The BLNDMA output color format is 16-bit RGB565 or ARGB1555 and the source of BitBlt and blending may be 16-bit RGB565 or ARGB1555.

When the BLNDMA controller performs the above functions, there are two ways to let CPU know the BLNDMA status. One is the Polling Mode, in which CPU can read the bit 8 "DMA_STATUS" for the register address 0x880D0018 and monitor whether the BLNDMA Controller finishes the data transfer. The other is the Interrupt Mode, in which the BLNDMA Controller will send an interrupt request to the CPU when it finishes the data transfer.

10.2 Architecture

The architecture of the BLNDMA Controller is shown below.

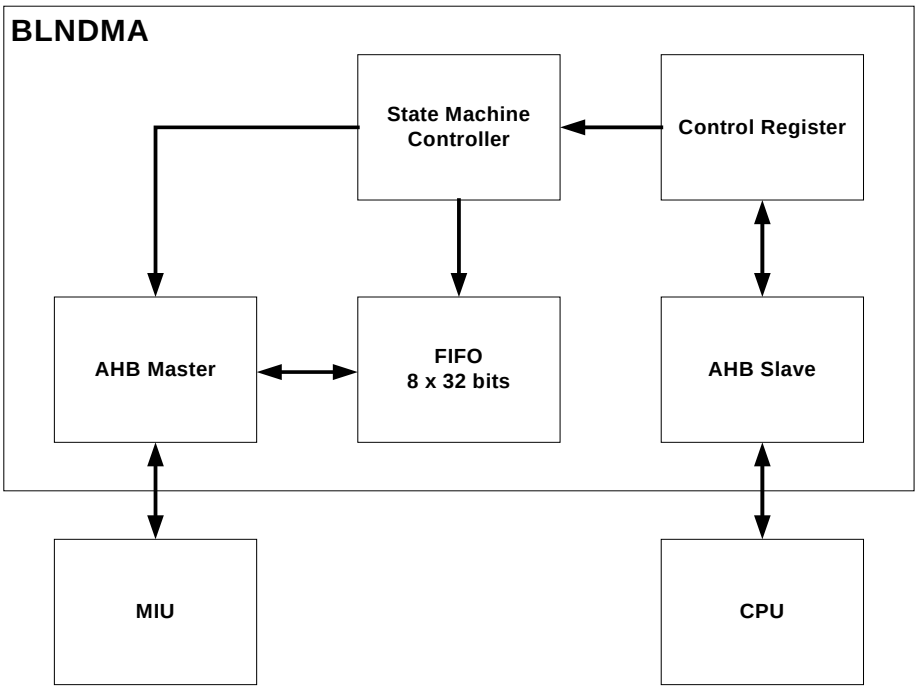


Fig 10-1 BLNDMA Controller Architecture

The BLNDMA Controller is mainly consisted of AHB Slave, State Machine Controller, control register, and AHB Master.

CPU can not only access the internal control register of the BLNDMA controller by the AHB Slave of the BLNDMA controller but also activate the state machine controller by the control register setting. After that, the state machine controller will control the AHB Master of the BLNDMA controller and access the data in the SDRAM through the MIU controller. Each time the AHB Master reads 8 words from MIU into the internal FIFO of the BLNDMA controller, and then performs the function of the BLNDMA controller, ultimately writes the data into the SDRAM.

10.3 Control Registers

10.3.1 P_BLNDMA_SOURCEA_SA (R/W) (0x880D0000): DMA Source A Starting Address

NAME	D27-D0
P_BLNDMA_SOURCEA_SA	DMA_SCRA_ADDR

Note: Obey 0xXXXXX0, 0xXXXXX4, 0xXXXXX8, 0xXXXXXC starting address rule.

10.3.2 P_BLNDMA_SOURCEB_SA (R/W) (0x880D0004): DMA Source B Starting Address

NAME	D27-D0
P_BLNDMA_SOURCEB_SA	DMA_SCRB_ADDR

Note: Obey 0xXXXXX0, 0xXXXXX4, 0xXXXXX8, 0xXXXXXC starting address rule.

10.3.3 P_BLNDMA_DESTINATION_SA (R/W) (0x880D0008): DMA Destination Starting Address

NAME	D27-D0
P_BLNDMA_DESTINATION_SA	DMA_DEST_ADDR

Note: Obey 0xXXXXX0, 0xXXXXX4, 0xXXXXX8, 0xXXXXXC starting address rule.

10.3.4 P_BLNDMA_WIDTH_HEIGHT (R/W) (0x880D000C): DMA Transfer Width

NAME	D26-D16	D15-D11	D10-D0
P_BLNDMA_WIDTH_HEIGHT	DMA_TRANS_HEIGH	-	DMA_TRANS_WIDTH

Note: 1. Unit is pixel, and one word includes two pixel data.

2. Must be the multiple of 8 words.

10.3.5 P_BLNDMA_FILL_PATTERN (R/W) (0x880D0010): DMA Fill Pattern

NAME	D31-D0
P_BLNDMA_FILL_PATTERN	DMA_FILL_PAT

10.3.6 P_BLNDMA_MODE_CTRL1 (R/W) (0x880D0014): DMA Operation Mode

NAME	D24	D23-D17	D16	D15-D9	D8	D7-D3	D2-D0
P_BLNDMA_MODE_CTRL1	DMA_START		FILTER_MODE		BLEND_MODE		OP_MODE

OP_MODE: DMA operation mode

0: Idle mode

1: Copy A and then Paste to Destination

2: Blend copied A and B and then Paste to Destination

3: Fill Pattern to Destination

4: Convert the format of copied A (YUV2RGB) and then paste to Destination

BLEND_MODE: DMA Blending mode

0: (A x A_FACTOR + B x B_FACTOR) >> 6

1: (A x A_FACTOR - B x B_FACTOR) >> 6

FILTER_MODE: DMA transparent filter mode

0: Disable

1: Enable

DMA_START: DMA operating function start

0: Disable

1: Enable (Start)

10.3.7 P_BLNDMA_INT_STATUS (R/W) (0x880D0018): DMA Interrupt Control

NAME	D24	D23-D17	D16	D15-D9	D8	D7-D1	D0
P_BLNDMA_INT_STATUS	DMA_INT_CLEAR		DMA_INT_EN		DMA_BUSY		DMA_IRQ

DMA_IRQ: DMA Interrupt Status

0: DMA Interrupt does not occur

1: DMA Interrupt occurs

DMA_BUSY: DMA Operation Busy

0: DMA Operation is Idle

1: DMA Operation is Busy

DMA_INT_EN: DMA Interrupt Control

0: Disable

1: Enable

DMA_INT_CLEAR: DMA Interrupt Clear

0: none

1: Clear DMA Interrupt

10.3.8 P_BLNDMA_ALPHA_VALUE (R/W) (0x880D001C): DMA Blending Factor

NAME	D13-D8	D7-D6	D5-D0
P_BLNDMA_ALPHA_VALUE	DMA_B_FACTOR		DMA_A_FACTOR

DMA_A_FACTOR: Blend source A Alpha value

DMA_B_FACTOR: Blend source B Alpha value

10.3.9 P_BLNDMA_FILTER_PATTERN (R/W) (0x880D0020): DMA Transparent Filter Pattern

NAME	D15-D0
P_BLNDMA_FILTER_PATTERN	DMA_FILTER_PAT

DMA_FILTER_PAT: DMA Transparent Filter Pattern

10.3.10 P_BLNDMA_ADDR_MODE(R/W)(0x880D0024): DMA Address Mode Control

NAME	D16	D15-D9	D8	D7-D1	D0
P_BLNDMA_ADDR_MODE	DEST_ADDR		B_ADDR		A_ADDR

A_ADDR: DMA Source A Addressing Mode

0: Linear Mode

1: Block Mode

B_ADDR: DMA Source B Addressing Mode

0: Linear Mode

1: Block Mode

DEST _ADDR: DMA Destination Addressing Mode

0: Linear Mode

1: Block Mode

10.3.11 P_BLNDMA_MODE_CTRL2 (R/W) (0x880D0028): DMA Control 2

NAME	D18-D16	D15-D9	D8	D7-D1	D0
P_BLNDMA_MODE_CTRL2	DMA_STATE		DMA_COLOR_MODE		DMA_ALPHA

DMA_ALPHA: Alpha bit for the ARGB1555 mode

0: Disable

1: Enable

DMA_COLOR_MODE: RGB type selection

0: RGB565

1: ARGB1555

DMA_STATE: DMA State Machine Status

0: DMA is Idle

1: DMA is Read A

2: DMA is Read B

3: DMA is Blend State or Convert YUV to RGB

4: DMA is Write

10.3.12 P_BLNDMA_SOURCEA_BA (R/W) (0x880D0030): DMA Source A Base Address

NAME	D27-D0
P_BLNDMA_SOURCEA_BA	DMA_ABASE_ADDR

Note: Obey 0xXXXX00 base address rule.

10.3.13 P_BLNDMA_SOURCEA_OA (R/W) (0x880D0034): DMA Source A Offset Address XY

NAME	D26-D16	D15-D11	D10-D0
P_BLNDMA_SOURCEA_OA	OFFSETA_Y		OFFSETA_X

Note: 1. Unit is pixel, and one word contains two pixel data.

2. Must be the multiple of 1 word.

10.3.14 P_BLNDMA_SOURCEA_BACKGROUND (R/W) (0x880D0038): DMA Source A Background Width/Height

NAME	D10-D8	D7-D3	D2-D0
P_BLNDMA_SOURCEA_BACKGROUND	BG_HEIGH		BG_WIDTH

BG_WIDTH: DMA Source A Background Width (Pixel)

0: 256

1: 320

2: 512

3: 640

4: 1024

5: 2048

BG_HEIGH: DMA Source A Background Height (Pixel)

0: 240

1: 256

2: 480

3: 512

4: 1024

5: 2048

10.3.15 P_BLNDMA_SOURCEB_BA (R/W) (0x880D0040): DMA Source B Base Address

NAME	D27-D0
P_BLNDMA_SOURCEB_BA	DMA_BBASE_ADDR

Note: Obey 0xXXXX00 base address rule.

10.3.16 P_BLNDMA_SOURCEB_OA (R/W) (0x880D0044): DMA Source B Offset Address XY

NAME	D26-D16	D15-D11	D10-D0
P_BLNDMA_SOURCEB_OA	OFFSETB_Y		OFFSETB_X

Note: 1. Unit is pixel, and one word includes two pixel data.

2. Must be the multiple of 1 word.

10.3.17 P_BLNDMA_SOURCEB_BACKGROUND (R/W) (0x880D0048): DMA Source B Background Width/Height

NAME	D10-D8	D7-D3	D2-D0
P_BLNDMA_SOURCEB_BACKGROUND	BG_HEIGH		BG_WIDTH

BG_WIDTH: DMA Source B Background Width (Pixel)

0: 256
1: 320
2: 512
3: 640
4: 1024
5: 2048

BG_HEIGH: DMA Source B Background Height (Pixel)

0: 240
1: 256
2: 480
3: 512
4: 1024
5: 2048

10.3.18 P_BLNDMA_DESTINATION_BA (R/W) (0x880D0050): DMA Destination Base Address

NAME	D27-D0
P_BLNDMA_DESTINATION_BA	DMA_DBASE_ADDR

Note: Obey 0xXXXX00 base address rule

10.3.19 P_BLNDMA_DESTINATION_OA (R/W) (0x880D0054): DMA Destination Offset Address XY

NAME	D26-D16	D15-D11	D10-D0
P_BLNDMA_DESTINATION_OA	OFFSETD_Y		OFFSETD_X

Note: 1. Unit is pixel, and one word contains two pixel data.

2. Must be the multiple of 1 word.

10.3.20 P_BLNDMA_DESTINATION_BACKGROUND (R/W)(0x880D0058): DMA Destination Background Width/Height

NAME	D10-D8	D7-D3	D2-D0
P_BLNDMA_DESTINATION_BACKGROUND	BG_HEIGH		BG_WIDTH

BG_WIDTH: DMA Destination Background Width (Pixel)

0: 256
1: 320
2: 512
3: 640
4: 1024
5: 2048

BG_HEIGH: DMA Destination Background Height (Pixel)

0: 240
1: 256
2: 480
3: 512
4: 1024
5: 2048

11 Local Data Memory - LDM

11.1 Architecture

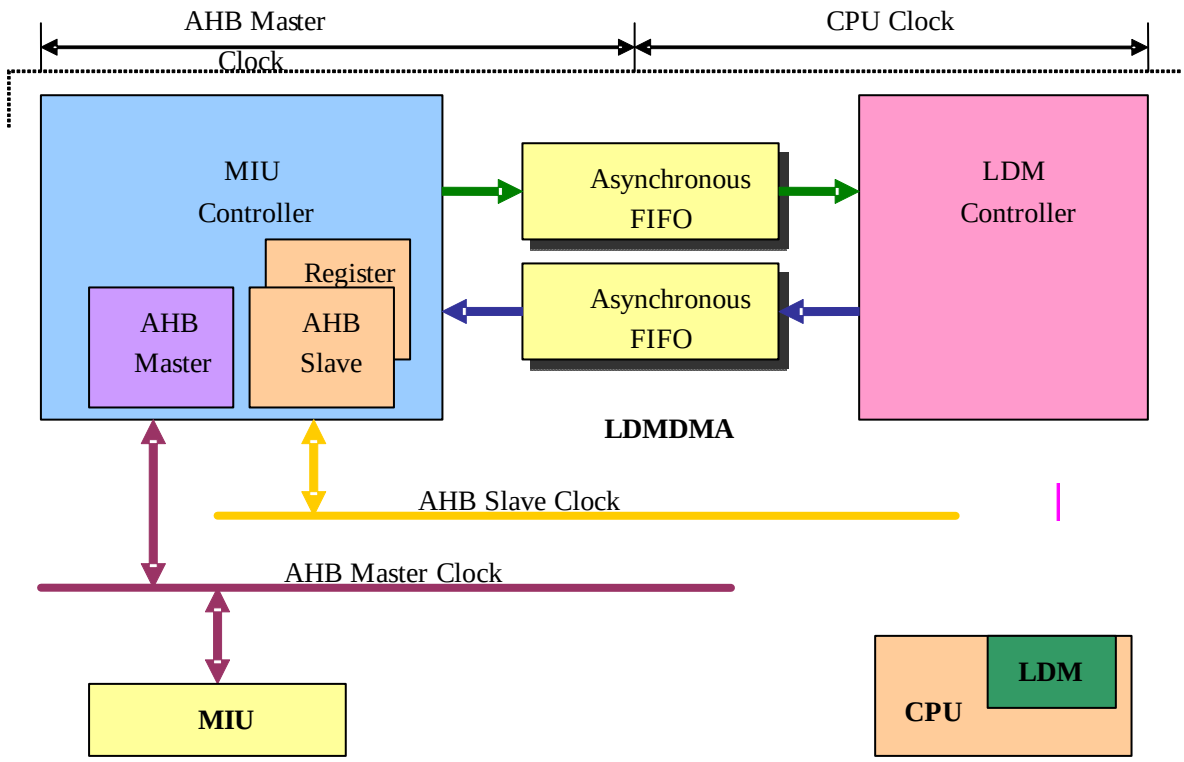


Fig 11-1 LDMDMA Architecture

MIU controller, acting as AHB Master, is used to access MIU, and LDM controller comes to act as master to access Local Data Memory (LDM) within CPU. Besides, CPU could set registers within LDMDMA by AHB slave.

11.2 LDMDMA Controller

This DMA controller adopts an half-duplex way, and only permits a kind of route (MIU to LDM/ LDM to MIU) to exist at the same time. When LDMDMA finishes, it will send an interrupt to notify CPU. LDMDMA supports four kinds of transmission ways:

(1) Byte mode

(2) Half-word mode

(3) Word mode

(4) 4-beat burst word mode

Since the whole system employs a 32-bit bus architecture, word addressing mode is used here, say, 4-beat burst word mode or word mode, for better transfer performance. But if you want to use byte or half-word addressing mode in system, SPCE3200 will facilitate you, too. So SPCE3200 provides these four modes for users to make a better choice.

11.3 Control Registers

Control registers in LDMDMA is accessed (R/W) by CPU through AHB slave.

11.3.1 P_LDM_MODE_CTRL (0x880C0000): DMA Control

NAME	D31	D30	D29	D28	D27-D26	D25-D0
P_LDM_MODE_CTRL	MODULE_EN	IRQ_EN	DIRECT	-	TRANS_MODE	-

TRANS_MODE: Transfer Mode

00: Byte mode (8 bits)

01: Half-word mode (16 bits)

10: Word mode (32 bits)

11: 4-beat burst word mode (32 bits of burst)

DIRECT: Access direction

0: MIU to LDM

1: LDM to MIU

IRQ_EN: IRQ Enable

0: Disable

1: Enable

MODULE_EN: Module Enable

0: Disable

1: Enable

11.3.2 P_LDM_MODE_STATUS (0x880C0004): DMA Status

NAME	D31	D30-D0
P_LDM_MODE_STATUS	INT_REQ	-

INT_REQ: Interrupt request (IRQ) (MIU to LDM/ LDM to MIU finish)

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No operation

Write 1: Clear Status

11.3.3 P_LDM_MIU_SA (0x880C0008): MIU Start Address

NAME	D27-D0
P_LDM_MIU_SA	MIU_START_ADDR (Byte address mode)

11.3.4 P_LDM_MIU_EA (0x880C000C): MIU End Address

NAME	D27-D0
P_LDM_MIU_EA	MIU_END_ADDR (Byte address mode)

11.3.5 P_LDM_START_ADDR (0x880C0010): LDM Start Address

NAME	D17-D0
P_LDM_START_ADDR	LDM_START_ADDR (Byte address mode)

11.3.6 P_LDM_END_ADDR (0x880C0014): LDM End Address

NAME	D17-D0
P_LDM_END_ADDR	LDM_END_ADDR (Byte address mode)

12 Peripheral Interrupt Controller

12.1 Introduction

The interrupt controller provides masking capability for all 40 interrupt sources and combines them into their final state for IRQ processor interrupt. It plays a role to generate the IRQ interrupt request to the S+Core processor after making arbitration process when there are multiple interrupt requests from internal peripherals and external interrupt request pins.

Although S+Core processor receives the IRQ interrupt events from the peripheral devices, it does not have a priority mechanism. Therefore the software like interrupt handler must deal with this kind of issues with hardware in the interrupt controller. For instance, assume that all interrupt sources have been set as IRQs, and ten of them request interrupt to processor at then same time, the software can determine the interrupt service priority by reading interrupt pending register, which denotes what kind of interrupt request happens.

This kind of interrupt process requires long interrupt latency to jump to the corresponding service routine. To resolve such inefficiency, S+Core processor provides another interrupt processing mechanism called vectored interrupt mode, a general feature of the CISC type Microprocessor. When multiple interrupt sources request interrupt simultaneously, the hardware priority logic will determine which interrupt should be serviced first. Meanwhile, this hardware logic applies the offset value from the vector table base address, which has a jump instruction to the corresponding service routine. With hardware assistance, it only needs several instructions to fetch a vector table entry and reduce the interrupt latency dramatically.

12.2 Features

- Compliant with the AMBA specification (Rev 2.0) for easy integration into System-on-Chip (SoC) implementation
- 40 interrupt sources
- Each interrupt is high active, and level sensitive IRQ.
- Supports vectored interrupt
- Programmable Priority for each interrupt source**Control Registers**

The following table shows the registers associated with the interrupt controller and physical address used to access them.

12.3.1 P_INT_REQ_STATUS1 (R) (0x880A0000): Interrupt Request Status Register.

Each of the 40 bits in the interrupt pending register, INTPND, corresponds to an interrupt

source. When an interrupt request is generated from the external device, the corresponding bit in the register is set to "1". Although several interrupt sources generate requests simultaneously, the INTPND indicates what interrupt sources generate the interrupt requests.

Name	D31-D0
P_INT_REQ_STATUS1	INTPND

INTPND: Indicate the interrupt request status

0: The interrupt has not been requested

1: The interrupt source has asserted the interrupt request

12.3.2 P_INT_REQ_STATUS2 (R) (0x880A0004): Interrupt Request Status High Byte

Name	D7-D0
P_INT_REQ_STATUS2	INTPND_H

INTPND_H: Indicate the interrupt request status

0: The interrupt has not been requested

1: The interrupt source has asserted the interrupt request

12.3.3 P_INT_GROUP_PRI (R/W) (0x880A0008): Interrupt Priority of IRQ Service Slave

This register determines the priorities among 4 IRQ service slaves. Each IRQ service slave can receive 8 IRQ sources.

Name	D7-D0
P_INT_GROUP_PRI	X_PMST

X_PMST: These 8 bits determine priorities among 4 slaves.

[1:0]: Priority of slave group 0

[3:2]: Priority of slave group 1

[5:4]: Priority of slave group 2

[7:6]: Priority of slave group 3

00 = 1st, 01 = 2nd, 02 = 3rd, 03 = 4th

12.3.4 P_INT_GROUPx_PRI (0x880A0010~0x880A001C): IRQ Priority Register

This register determines the priority of each IRQ service slave. Each IRQ service slave can receive 8 IRQ sources.

NAME	ADDR	D23-D21	D20-D18	D17-D15	D14-D12	D11-D9	D8-D6	D5-D3	D2-D0

P_INT_GROUP0_PRI	0x880A0010	PSLV0_7	PSLV0_6	PSLV0_5	PSLV0_4	PSLV0_3	PSLV0_2	PSLV0_1	PSLV0_0
P_INT_GROUP1_PRI	0x880A0014	PSLV1_7	PSLV1_6	PSLV1_5	PSLV1_4	PSLV1_3	PSLV1_2	PSLV1_1	PSLV1_0
P_INT_GROUP2_PRI	0x880A0018	PSLV2_7	PSLV2_6	PSLV2_5	PSLV2_4	PSLV2_3	PSLV2_2	PSLV2_1	PSLV2_0
P_INT_GROUP3_PRI	0x880A001C	PSLV3_7	PSLV3_6	PSLV3_5	PSLV3_4	PSLV3_3	PSLV3_2	PSLV3_1	PSLV3_0

PSLVX_0: Priority of source 0

PSLVX_1: Priority of source 1

PSLVX_2: Priority of source 2

PSLVX_3: Priority of source 3

PSLVX_4: Priority of source 4

PSLVX_5: Priority of source 5

PSLVX_6: Priority of source 6

PSLVX_7: Priority of source 7

NOTE: X = Slave X

000 = 1st, 001 = 2nd, 010 = 3rd, 011 = 4th

100 = 5th, 101 = 6th, 110 = 7th, 111 = 8th

12.4 Vector Address

The interrupt handler has the vector address to return to the processor respective to IRQ interrupt. The interrupt handler uses this vector value without calculating the offset address in the interrupt vector table to find the interrupt service routine. It allows the processor to skip the search process required in the non-vectored mode to get the offset address whenever it receives an interrupt. This mechanism makes a system improve the overall performance dramatically. The value in this register will not change until another interrupt is generated.

12.5 Interrupt Sources

Slave Group	Source Number	Source	Vector Address

Slave Group	Source Number	Source	Vector Address
0	—	—	—
	—	—	—
	—	—	—
	—	—	—
	4	ADC gain overflow / ADC recorder FIFO overflow	59
	5	General-purpose ADC	58
	6	Timer base	57
	7	Timer	56
1	—	—	—
	9	LCD vblanking start	54
	—	—	—
	11	Light Gun	52
	12	Sensor frame end	51
	13	Sensor coordinate hit	50
	14	Sensor motion frame end	49
	15	Sensor capture done + sensor debug IRQ	48
2	16	TV coordinate hit	47
	—	—	—
	18	USB host+device	45
	19	SIO	44
	20	SPI	43
	21	UART (IrDA)	42
	22	NAND	41
	23	SD	40
3	24	I^2C master	39
	25	I^2S slave	38
	26	APBDMA CH1	37
	27	APBDMA CH2	36

Slave Group	Source Number	Source	Vector Address
	28	LDM_DMA	35
	29	BLN_DMA	34
	30	APBDMA CH3	33
	31	APBDMA CH4	32
4	32	Alarm + HMS	31
	33	MP4	30
	34	C3 (ECC module)	29
	35	GPIO	28
	36	Bufctl (for debugging)	27
	37	RESERVED1	26
	38	RESERVED2	25
	39	RESERVED3	24

13 Software Config - SFTCFG

13.1 Introduction

The Software Configuration (SFTCFG) module can generate five groups of signals. The first is the hardware configuration in the normal mode and the generated signals are JTAG enable, warm configuration reset, boot mode, and bypass PLL signals. The second is the hardware configuration in the test mode and the generated signals are connected to the pin multiplex (PINMUX) module as the mode selection signals. The third is the software configuration generated when the CPU programs the related registers and the generated signals are also connected to the pin multiplex (PINMUX) module as the mode selection signals. The fourth is the GPIO configuration generated when the CPU programs the corresponding registers and the generated signals including output port data, output enable, pull-up and pull-down signals are also connected to the input/output pads. The fifth is the external interrupt sources (GPIO), which can be programmed to be rising or falling trigger. When the external interrupt source is triggered, the SFTCFG module will signal an interrupt request to the CPU.

13.2 Architecture

The architecture of the SFTCFG module is shown below.

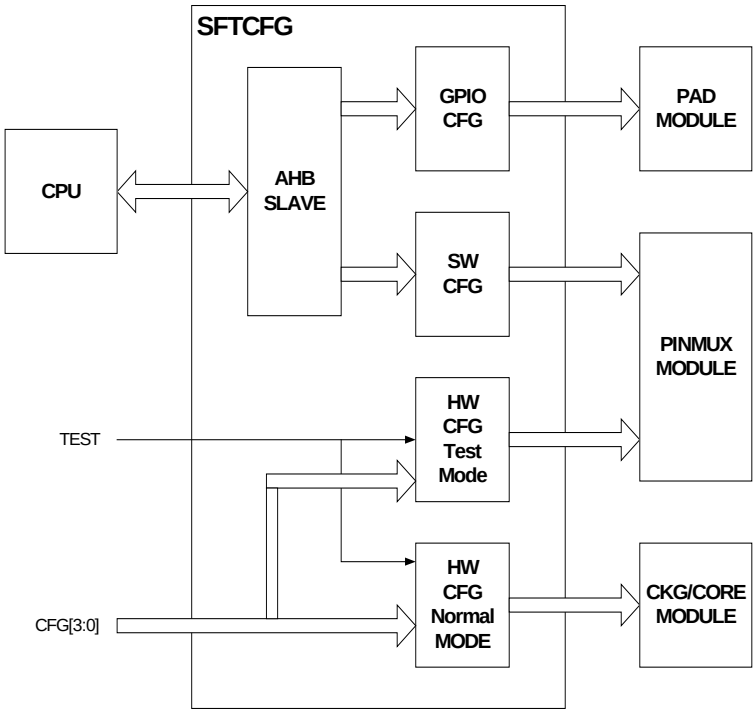


Fig 13-1 SFTCFG Architecture

13.3 Control Registers

13.3.1 P_UART_INTERFACE_SEL / P_LCD_INTERFACE_SEL / P_CSI_INTERFACE_SEL / P_TV_INTERFACE_SEL (0x88200000): Software Configuration for Device Selection

NAME	D24	D23-D17	D16	D15-D10	D9-D8	D7-D2	D1-D0
P_UART_INTERFACE_SEL **P_LCD_INTERFACE_SEL** **P_CSI_INTERFACE_SEL** **P_TV_INTERFACE_SEL**	SW_UART		SW_TVE		SW_CSI		SW_LCD

SW_LCD: Software Configuration for LCD selection

00: Not as LCD Interface

01: TFT_AUO

10: TFT_TOPPOLY

11: STN

SW_CSI: Software Configuration for Sensor Interface Format selection

00: Not as Sensor Interface

01: CCIR601

10: SUNPLUS format

11: CCIR656

SW_TVE: Software Configuration for TV Encoder Hsync/Vsync Probe Selection

0: Disable

1: Enable

SW_UART: Software Configuration for UART selection

0: As GPIO

1: As UART

13.3.2 P_SIO_INTERFACE_SEL / P_I2C_INTERFACE_SEL / P_CSI_PROBE_SEL / P_I2S_INTERFACE_SEL / P_ROMCSN_INTERFACE_SEL (0x88200004): Software Configuration for Function Selection

NAME	D24	D23-D17	D16	D15-D11	D10-D8	D7-D1	D0
P_SIO_INTERFACE_SEL **P_I2C_INTERFACE_SEL** **P_CSI_PROBE_SEL** **P_I2S_INTERFACE_SEL**	SW_EROM		SW_CSI_PROBE		SW_PERI		SW_I2C

NAME	D24	D23-D17	D16	D15-D11	D10-D8	D7-D1	D0
P_ROMCSN_INTERFACE_SEL							

SW_I2C: Software Configuration for I2C Selection

0: As GPIO

1: As I2C

SW_PERI: Software Configuration for Peripheral Selection

000: JTAG

001: SIO

010: I2S Master Mode

011: I2S Slave Mode

Others: Not as Peripheral Interface

SW_CSI_PROBE: Software Configuration for Sensor Hsync/Vsync/Field Probe Selection

0: Disable

1: Enable

SW_EROM: Software Configuration for External ROM Selection

0: As GPIO

1: As External ROM

13.3.3 P_WAKEUP_KEYC_SEL / P_DRAM_INTERFACE_SEL (0x88200008): Software Configuration for DRAM Selection

NAME	D24	D23-D17	D16	D15-D9	D8	D7-D3	D2-D0
P_WAKEUP_KEYC_SEL P_DRAM_INTERFACE_SEL	SW_DRAM_BA1		SW_DRAM_A12		SW_DRAM_A11		SW_KEYC

SW_KEYC: Software Configuration for Key Change Interface Selection

1: TFT Interface

2: CSI Interface

3: ADC1 Configuration

4: ADC2 Configuration

5: NFLASH Interface

Others: No Key Change Interface

SW_DRAM_A11: Software Configuration for DRAM A11 Selection

0: As GPIO

1: As DRAM A11

SW_DRAM_A12: Software Configuration for DRAM A12 Selection

0: As GPIO

1: As DRAM A12

SW_DRAM_BA1: Software Configuration for DRAM BA1 Selection

0: As GPIO

1: As DRAM BA1

13.3.4 P_USB_INTERFACE_SEL (0x8820000C): Software Configuration for Transceiver Output Selection

NAME	D24	D23-D17	D16	D15-D9	D8	D7-D1	D0
P_USB_INTERFACE_SEL					SW_VDAC_OUT		SW_USBTRX_OUT

SW_USBTRX_OUT: Software Configuration for USB transceiver Output

0: Disable

1: Enable

SW_VDAC_OUT: Software Configuration for VIDEO DAC Output

0: Disable

1: Enable

13.3.5 P_TIMER_INTERFACE_SEL (0x88200010): Software Configuration for LightPen Selection

NAME	D24	D23-D22	D21-D16	D15-D10	D9	D8	D7-D1	D0
P_TIMER_INTERFACE_SEL	SW_AFE		SW_TCCP		SW_LP1	SW_LP0		SW_SPEED

SW_SPEED: Software Configuration for Speed Test

0: Disable

1: Enable

SW_LP0: Software Configuration for LightPen 0

0: Disable

1: Enable

SW_LP1: Software Configuration for LightPen 1

0: Disable

1: Enable

SW_TCCP: Software Configuration for Timer CCP

0: Disable

1: Enable

SW_AFE: Software Configuration for AFE test

0: Disable

1: Enable

13.3.6 P_TFT_GPIO_DATA (0x88200014): TFT GPIO Port Data

NAME	D19-D0
P_TFT_GPIO_DATA	TFT_GPIO_O

TFT_GPIO_O: TFT GPIO Port Data

13.3.7 P_TFT_GPIO_OUTPUTEN (0x88200018): TFT GPIO Port Output Setting

NAME	D19-D0
P_TFT_GPIO_OUTPUTEN	TFT_GPIO_OE

TFT_GPIO_OE: TFT GPIO Port Output Enable

0: Disable

1: Enable

13.3.8 P_TFT_GPIO_PULLUP (0x8820001C): TFT GPIO Port Pull Up Setting

NAME	D19-D0
P_TFT_GPIO_PULLUP	TFT_GPIO_PU

TFT_GPIO_PU: TFT GPIO Port Pull Up Enable

1: Disable

0: Enable

13.3.9 P_TFT_GPIO_PULLDOWN (0x88200020): TFT GPIO Port Pull Down Setting

NAME	D19-D0
P_TFT_GPIO_PULLDOWN	TFT_GPIO_PD

TFT_GPIO_PD: TFT GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.10 P_CSI_GPIO_SETUP (0x88200024): CSI GPIO Port Control

NAME	D28-D16	D15-D13	D12-D0
P_CSI_GPIO_SETUP	CSI_GPIO_OE		CSI_GPIO_O

CSI_GPIO_O: CSI GPIO Port Data

CSI_GPIO_OE: CSI GPIO Port Output Enable

0: Disable

1: Enable

13.3.11 P_CSI_GPIO_PULL (0x88200028): CSI GPIO Port Pull Status

NAME	D28-D16	D15-D13	D12-D0
P_CSI_GPIO_PULL	CSI_GPIO_PD		CSI_GPIO_PU

CSI_GPIO_PU: CSI GPIO Port Pull Up Enable

1: Disable

0: Enable

CSI_GPIO_PD: CSI GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.12 P_NAND_GPIO_SETUP (0x8820002C): NFLASH GPIO Port Control

NAME	D31-D16	D15-D0
P_NAND_GPIO_SETUP	NFLASH_GPIO_OE	NFLASH_GPIO_O

NFLASH_GPIO_O: NFLASH GPIO Port Data

NFLASH_GPIO_OE: NFLASH GPIO Port Output Enable

0: Disable

1: Enable

13.3.13 P_NAND_GPIO_PULL (0x88200030): NFLASH GPIO Port Pull Status

NAME	D31-D16	D15-D0
P_NAND_GPIO_PULL	NFLASH_GPIO_PD	NFLASH_GPIO_PU

NFLASH_GPIO_PU: NFLASH GPIO Port Pull Up Enable

1: Disable

0: Enable

NFLASH_GPIO_PD: NFLASH GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.14 P_JTAG_GPIO_SETUP (0x88200034): JTAG GPIO Port Control

NAME	D28-D24	D23-D21	D20-D16	D15-D13	D12-D8	D7-D5	D4-D0
P_JTAG_GPIO_SETUP	JTAG_GPIO_PD		JTAG_GPIO_PU		JTAG_GPIO_OE		JTAG_GPIO_O

JTAG_GPIO_O: JTAG GPIO Port Data

JTAG_GPIO_OE: JTAG GPIO Port Output Enable

0: Disable

1: Enable

JTAG_GPIO_PU: JTAG GPIO Port Pull Up Enable

1: Disable

0: Enable

JTAG_GPIO_PD: JTAG GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.15 P_IOA_GPIO_SETUP (0x88200038): GLOBAL GPIO Port Control

NAME	D25-D24	D23-D18	D17-D16	D15-D10	D9-D8	D7-D2	D1-D0
P_IOA_GPIO_SETUP	XGPIO_PD		XGPIO_PU		GLOBAL_OE		GLOBAL_O

GLOBAL _O: GLOBAL GPIO Port Data

GLOBAL _OE: GLOBAL GPIO Port Output Enable

0: Disable

1: Enable

XGPIO_PU: GLOBAL GPIO Port Pull Up Enable

1: Disable

0: Enable

XGPIO_PD: GLOBAL GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.16 P_USB_GPIO_SETUP (0x8820003C): USB GPIO Port Control

NAME	D24	D23-D17	D16	D15-D9	D8	D7-D1	D0
P_USB_GPIO_SETUP	XUSB_PD		XUSB_PU		USBDET_OE		USBDET_O

USBDET _O: USBDET GPIO Port Data

USBDET _OE: USBDET GPIO Port Output Enable

0: Disable

1: Enable

XUSB_PU: USBDET GPIO Port Pull Up Enable

1: Disable

0: Enable

XUSB_PD: USBDET GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.17 P_UART_GPIO_SETUP (0x88200040): UART GPIO Port Control

NAME	D25-D24	D23-D18	D17-D16	D15-D10	D9-D8	D7-D2	D1-D0
P_UART_GPIO_SETUP	XUART_PD		XUART_PU		UART_OE		UART_O

UART_O: UART GPIO Port Data

UART _OE: UART GPIO Port Output Enable

0: Disable

1: Enable

XUART_PU: UART GPIO Port Pull Up Enable

1: Disable

0: Enable

XUART_PD: UART GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.18 P_I2C_GPIO_SETUP (0x88200044): I^2C GPIO Port Control

NAME	D25-D24	D23-D18	D17-D16	D15-D10	D9-D8	D7-D2	D1-D0
P_I2C_GPIO_SETUP	X I2C_PD		X I2C_PU		I2C_OE		I2C_O

I2C_O: I^2C GPIO Port Data

I2C_OE: I^2C GPIO Port Output Enable

0: Disable

1: Enable

XI2C_PU: I^2C GPIO Port Pull Up Enable

1: Disable

0: Enable

XI2C_PD: I^2C GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.19 P_ADC_GPIO_SETUP (0x88200048): ADC GPIO Port Control

NAME	D31-D24	D23-D16	D15-D8	D7-D0
P_ADC_GPIO_SETUP	XADC_PD	XAD_PU	ADC_OE	ADC_O

ADC_O: ADC GPIO Port Data

ADC_OE: ADC GPIO Port Output Enable

0: Disable

1: Enable

XADC_PU: ADC GPIO Port Pull Up Enable

1: Disable

0: Enable

XADC_PD: ADC GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.20 P_DRAM_GPIO_SETUP (0x88200050): DRAM GPIO Port Control

NAME	D27-D24	D23-D17	D16	D15-D12	D11-D8	D7-D4	D3-D0
P_DRAM_GPIO_SETUP	XDRAM_PD		XDRAM_PU		DRAM_OE		DRAM_O

DRAM_O: DRAM GPIO Port Data

D0: ROMCSN

D1: DRAM_A11

D2: DRAM_A12

D3: DRAM_BA1

DRAM_OE: DRAM GPIO Port Output Enable

Bit 8:

0: Disable

1: Enable

Bit 9~11:

0: Enable

1: Disable

XDRAM_PU: DRAM GPIO Port Pull Up Enable

1: Disable

0: Enable

XDRAM_PD: DRAM GPIO Port Pull Down Enable

0: Disable

1: Enable

13.3.21 P_ADC_AINPUT_CTRL (0x88200054): ADC GPIO Port Analog Input Control

NAME	D16	D15-D9	D8	D7-D0
P_ADC_AINPUT_CTRL	SWEFU_ENB		SWEFU_READ	ADC_GPIO_AEN

ADC_GPIO_AEN: ADC GPIO Port Analog Input Enable

0: Disable

1: Enable

SWEFU_READ: EFUSE Software Read Signal, Active High

SWEFU_ENB: EFUSE Software Enable Signal, Active Low

13.3.22 P_TFT_GPIO_INPUT (0x88200064): TFT GPIO Input Control

NAME	D19-D0
P_TFT_GPIO_INPUT	TFT_INPUT

TFT_INPUT: TFT GPIO Input Value

13.3.23 P_CSI_GPIO_INPUT (0x88200068): CSI GPIO Input Control

NAME	D12-D0
P_CSI_GPIO_INPUT	CSI_INPUT

CSI_INPUT: CSI GPIO Input Value

13.3.24 P_NAND_GPIO_INPUT (0x8820006C): NFLASH GPIO Input Control

NAME	D15-D0
P_NAND_GPIO_INPUT	NFLASH_INPUT

NFLASH_INPUT: NFLASH GPIO Input Value

13.3.25 P_JTAG_GPIO_INPUT / P_IOB_GPIO_INPUT / P_DRAM_GPIO_INPUT (0x88200070): JTAG/CDServo/DRAM GPIO Input Control

NAME	D19-D16	D15-D14	D13-D8	D7-D5	D4-D0
P_JTAG_GPIO_INPUT P_IOB_GPIO_INPUT P_DRAM_GPIO_INPUT	DRAM_INPUT		—		JTAG_INPUT

JTAG_INPUT: JTAG GPIO Input Value

DRAM_INPUT: DRAM GPIO Input Value

13.3.26 P_UART_GPIO_INPUT / P_I2C_GPIO_INPUT / P_IOA_GPIO_INPUT / P_USB_GPIO_INPUT (0x88200074): GUUI GPIO Input Control

NAME	D25-D24	D23-D18	D17-D16	D15-D9	D8	D7-D2	D1-D0
P_UART_GPIO_INPUT P_I2C_GPIO_INPUT P_IOA_GPIO_INPUT P_USB_GPIO_INPUT	I2C_IN		UART_IN		USBDET_IN		GLOBAL_IN

GLOBAL_IN: Global GPIO Input Value

USBDET_IN: USBDET JTAG GPIO Input Value

UART_IN: UART GPIO Input Value

I2C_IN: I2C GPIO Input Value

13.3.27 P_ADC_GPIO_INPUT (0x88200078): ADC GPIO Input Control

NAME	D7-D0
P_ADC_GPIO_INPUT	ADC_INPUT

ADC_INPUT: ADC GPIO Input Value

13.3.28 P_TFT_GPIO_INT (0x88200080): TFT GPIO Interrupt Control

NAME	D27-D24	D23-D20	D19-D16	D15-D12	D11-D8	D7-D4	D3-D0
P_TFT_GPIO_INT	TFT_FI		TFT_RI		TFT_FIEN		TFT_RIEN

TFT_RIEN: TFT GPIO Rising Interrupt Enable

0: Disable

1: Enable

TFT_FIEN: TFT GPIO Falling Interrupt Enable

0: Disable

1: Enable

TFT_RI: TFT GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

TFT_FI: TFT GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.29 P_CSI_GPIO_INT (0x88200084): CSI GPIO Interrupt Control

NAME	D27-D24	D23-D20	D19-D16	D15-D12	D11-D8	D7-D4	D3-D0
P_CSI_GPIO_INT	CSI_FI		CSI_RI		CSI _FIEN		CSI _RIEN

CSI _RIEN: CSI GPIO Rising Interrupt Enable

0: Disable

1: Enable

CSI _FIEN: CSI GPIO Falling Interrupt Enable

0: Disable

1: Enable

CSI _RI: CSI GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

CSI _FI: CSI GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.30 P_NAND_GPIO_INT (0x88200088): NFLASH GPIO Interrupt Control

NAME	D27-D24	D23-D20	D19-D16	D15-D12	D11-D8	D7-D4	D3-D0
P_NAND_GPIO_INT	NFLASH_FI		NFLASH _RI		NFLASH _FIEN		NFLASH _RIEN

NFLASH _RIEN: NFLASH GPIO Rising Interrupt Enable

0: Disable

1: Enable

NFLASH _FIEN: NFLASH GPIO Falling Interrupt Enable

0: Disable

1: Enable

NFLASH _RI: NFLASH GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

NFLASH _FI: NFLASH GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.31 P_JTAG_GPIO_INT (0x8820008C): JTAG GPIO Interrupt Control

NAME	D28-D24	D23-D21	D20-D16	D15-D13	D12-D8	D7-D5	D4-D0
P_JTAG_GPIO_INT	JTAG_FI		JTAG _RI		JTAG _FIEN		JTAG _RIEN

JTAG_RIEN: JTAG GPIO Rising Interrupt Enable

0: Disable

1: Enable

JTAG_FIEN: JTAG GPIO Falling Interrupt Enable

0: Disable

1: Enable

JTAG_RI: JTAG GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

JTAG_FI: JTAG GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.32 P_IOA_GPIO_INT (0x88200090): GLOBAL GPIO Interrupt Control

NAME	D25-D24	D23-D18	D17-D16	D15-D10	D9-D8	D7-D2	D1-D0
P_IOA_GPIO_INT	GLOBAL_FI		GLOBAL _RI		GLOBAL _FIEN		GLOBAL _RIEN

GLOBAL _RIEN: GLOBAL GPIO Rising Interrupt Enable

0: Disable

1: Enable

GLOBAL _FIEN: GLOBAL GPIO Falling Interrupt Enable

0: Disable

1: Enable

GLOBAL _RI: GLOBAL GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

GLOBAL _FI: GLOBAL GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.33 P_UART_GPIO_INT (0x88200094): UART GPIO Interrupt Control

NAME	D25-D24	D23-D18	D17-D16	D15-D10	D9-D8	D7-D2	D1-D0
P_UART_GPIO_INT	UART_FI		UART _RI		UART _FIEN		UART _RIEN

UART _RIEN: UART GPIO Rising Interrupt Enable

0: Disable

1: Enable

UART _FIEN: UART GPIO Falling Interrupt Enable

0: Disable

1: Enable

UART _RI: UART GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

UART _FI: UART GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.34 P_I2C_GPIO_INT (0x88200098): I^{2}C GPIO Interrupt Control

NAME	D25-D24	D23-D18	D17-D16	D15-D10	D9-D8	D7-D2	D1-D0
P_I2C_GPIO_INT	I2C_FI		I2C _RI		I2C _FIEN		I2C _RIEN

I2C _RIEN: I^{2}C GPIO Rising Interrupt Enable

0: Disable

1: Enable

I2C _FIEN: I^{2}C GPIO Falling Interrupt Enable

0: Disable

1: Enable

I2C _RI: I^{2}C GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

I2C _FI: I^2C GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.35 P_ADC_GPIO_INT (0x8820009C): ADC GPIO Interrupt Control

NAME	D27-D24	D23-D20	D19-D16	D15-D12	D11-D8	D7-D4	D3-D0
P_ADC_GPIO_INT	ADC_FI		ADC _RI		ADC _FIEN		ADC _RIEN

ADC _RIEN: ADC GPIO Rising Interrupt Enable

0: Disable

1: Enable

ADC _FIEN: ADC GPIO Falling Interrupt Enable

0: Disable

1: Enable

ADC _RI: ADC GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

ADC _FI: ADC GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.36 P_USB_GPIO_INT (0x882000A0): USBDET GPIO Interrupt Control

NAME	D24	D23-D17	D16	D15-D9	D8	D7-D1	D0
P_USB_GPIO_INT	USBDET_FI		USBDET _RI		USBDET _FIEN		USBDET _RIEN

USBDET _RIEN: USBDET GPIO Rising Interrupt Enable

0: Disable

1: Enable

USBDET _FIEN: USBDET GPIO Falling Interrupt Enable

0: Disable

1: Enable

USBDET _RI: USBDET GPIO Rising Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

USBDET _FI: USBDET GPIO Falling Interrupt Event

Read 1: Interrupt Event Occurs

Read 0: No Interrupt Event

Write 1: Clear Interrupt Event

Write 0: No Action

13.3.37 P_SD_INTERFACE_SEL / P_NAND_INTERFACE_SEL / P_SPI_INTERFACE_SEL (0x882000A4): NFSDSPI Control

NAME	D16	D15-D9	D8	D7-D1	D0
P_SD_INTERFACE_SEL **P_NAND_INTERFACE_SEL** **P_SPI_INTERFACE_SEL**	SD_EN		SPI_EN		NFLASH_EN

NFLASH_EN: NFLASH Enable

0: Disable

1: Enable

SPI_EN: SPI Enable

0: Disable

1: Enable

SD_EN: SDCard Enable

0: Disable

1: Enable

14 Timer

14.1 Introduction

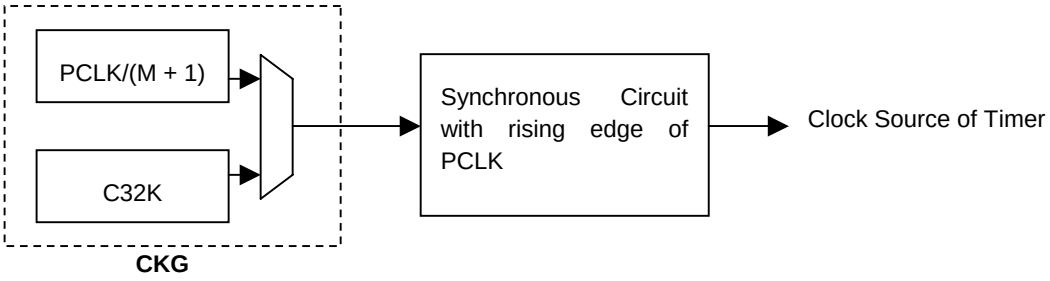
SPCE3200 contains six 16-bit timers/counters. All timers support Capture/Comparison/PWM (CCP) functions when cooperating with their own two 16-bit registers (preload register and CCP register). The clock source of these six timers can be programmed to internal clock (PCLK/2(**max 13.5MHz**) or C32K.

Features of these six timers/counters:

(1) PCLK/2 base and C32K base clock source selection for each timer source

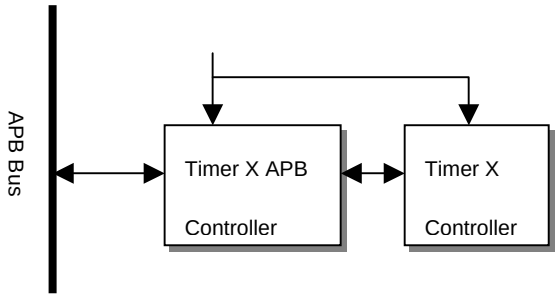
(2) Each Programmable clock must be synchronous with PCLK for each timer

(3) Each timer supports Capture / Comparison / PWM function



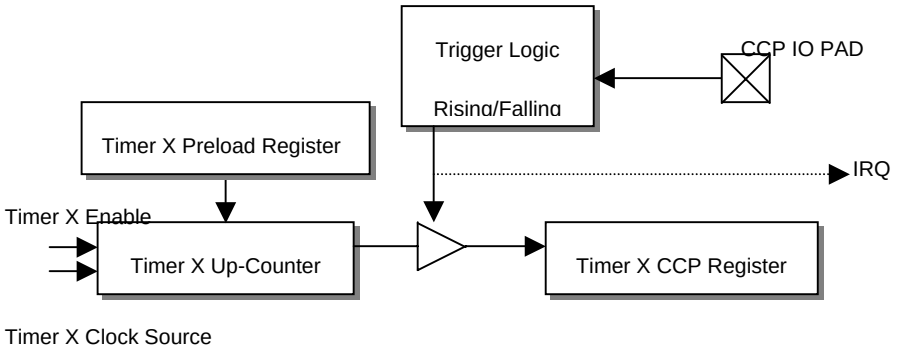
M = 1 ~ 255, and M = 0 is illegal

14.2 Architecture

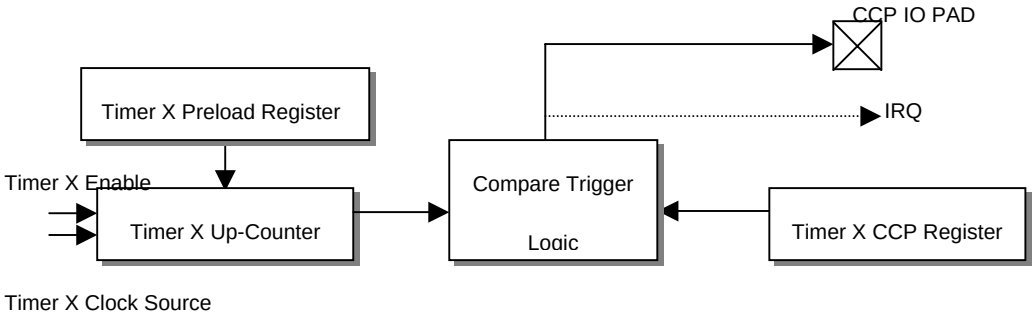


X = 0 ~ 5 (Timer 0 ~ Timer 5)

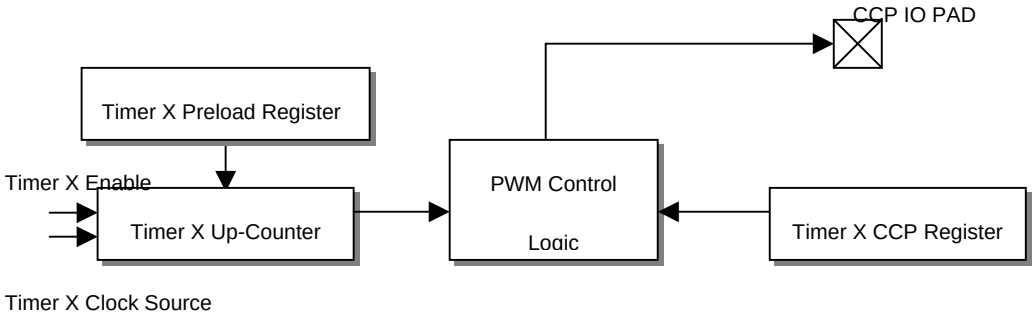
- **X Capture Mode Function Diagram**

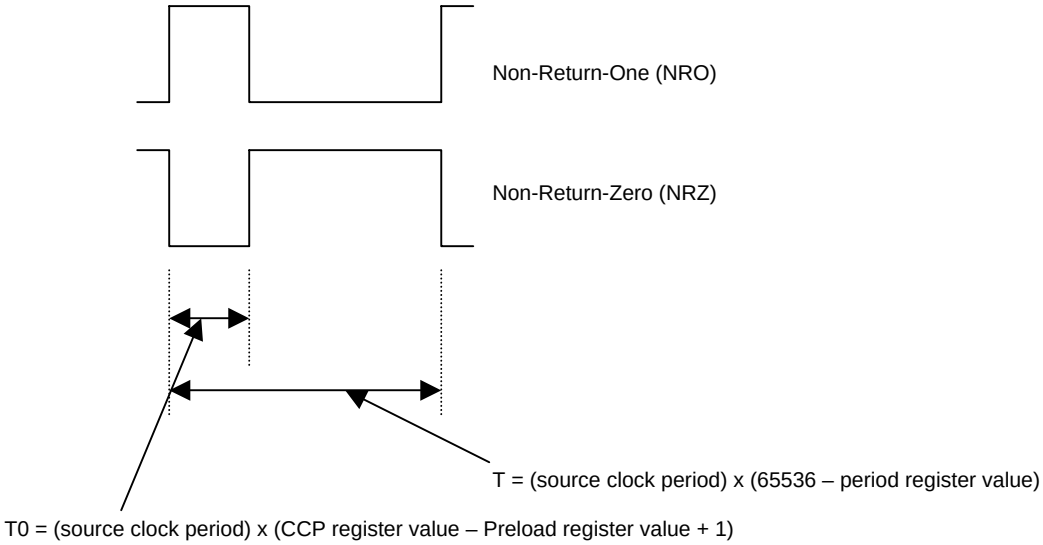


■ X Comparison Mode Function Diagram



■ X PWM Mode Function Diagram





14.3 Control Registers

14.3.1 P_TIMERx_MODE_CTRL (0x8816X000): Timer X Control

NAME	D31	D30-D28	D27	D26	D25-D0
P_TIMERx_MODE_CTRL	TIMER_X_EN		TIMER_X_IRQ_EN	TIMER_X_IRQ_FLAG	

TIMER_X_EN: Timer X Enable

0: Disable

1: Enable

TIMER_X_IRQ_EN: Timer X Interrupt Enable

0: Disable

1: Enable

TIMER_X_IRQ_FLAG: Timer X Interrupt Flag

Read 0: IRQ doesn't happen

Read 1: IRQ occurs

Write 1: Clear the flag

14.3.2 P_TIMERx_CCP_CTRL (0x8816X004): CCP Control

NAME	D31-D30	D29-D28	D27	D26	D25	D24-D0
P_TIMERx_CCP_CTRL	CCP_MODE		CAP_MODE	COM_MODE	PWM_MODE	

PWM_MODE: PWM Mode Selection

0: NRO output

1: NRZ output

COM_MODE: Comparison Mode Selection

0: High pulse

1: Low pulse

CAP_MODE: Capture Mode Selection

0: Every falling

1: Every rising

CCP_MODE: CCP Mode Selection

00: Timer Mode

01: Capture Mode

10: Comparison Mode

11: PWM Mode

14.3.3 P_TIMERx_PRELOAD_DATA (0x8816X008): Timer X Preload

NAME	D31-D0
P_TIMERx_PRELOAD_DATA	TIMER_X_PRELOAD_REGS

TIMER_X_PRELOAD_REGS: Timer X Preload number

14.3.4 P_TIMERx_CCP_DATA (0x8816X00C): Timer X CCP

NAME	D31-D0
P_TIMERx_CCP_DATA	TIMER_X_CCP_REGS

TIMER_X_CCP_REGS: Timer X CCP data

14.3.5 P_TIMERx_COUNT_DATA (0x8816X010): Timer X Up Counter

NAME	D31 -D0
P_TIMERx_COUNT_DATA	TIMER_X_UPCOUNT

TIMER_X_UPCOUNT: Timer X counts in Up Counter

15 Real Time Clock - RTC

15.1 Introduction

The Real Time Clock (RTC) unit is operating in 32768Hz clock domain. The data includes second, minute, hour, date, and alarm. The system also supports a time-base function which is based on 32768Hz.

Features of the RTC unit:

- Supports half-second, one second, one minute, and one hour interrupt request
- Supports alarm function interrupt request
- Supports 3 Time-Base Functions

15.2 Architecture

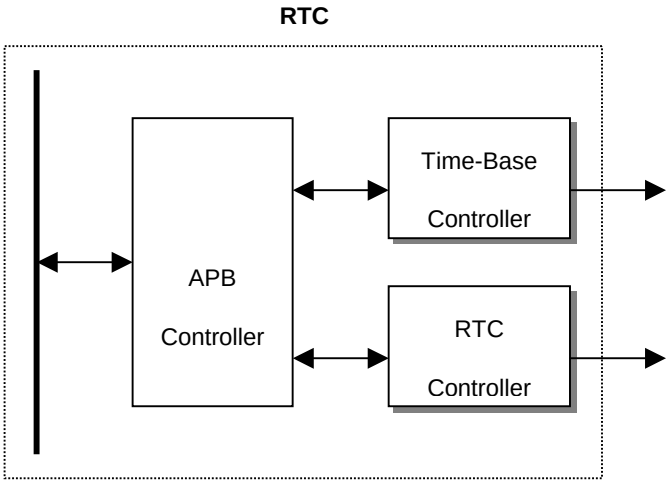


Fig 15-1 RTC Architecture

15.3 Control Registers

15.3.1 P_RTC_TIME_SEC (0x88166000): RTC Second Setup

NAME	D5 -D0
P_RTC_TIME_SEC	SECOND_SET

SECOND_SET: Parameter for second setup

15.3.2 P_RTC_TIME_MIN (0x88166004): RTC Minute Setup

NAME	D5 -D0
P_RTC_TIME_MIN	MINUTE_SET

MINUTE_SET: Parameter for minute setup

15.3.3 P_RTC_TIME_HOUR (0x88166008): RTC Hour Setup

NAME	D4 -D0
P_RTC_TIME_HOUR	HOUR_SET

HOUR_SET: Parameter for hour setup

15.3.4 P_RTC_ALM_SEC (0x8816600C): Alarm Second Setup

NAME	D5 -D0
P_RTC_ALM_SEC	SECOND_SET

SECOND _SET: Parameter for alarm second setup

15.3.5 P_RTC_ALM_MIN (0x88166010): Alarm Minute Setup

NAME	D5 -D0
P_RTC_ALM_MIN	MINUTE_SET

MINUTE_SET: Parameter for alarm minute setup

15.3.6 P_RTC_ALM_HOUR (0x88166014): Alarm Hour Setup

NAME	D4 -D0
P_RTC_ALM_HOUR	HOUR_SET

HOUR_SET: Parameter for alarm hour setup

15.3.7 P_RTC_MODE_CTRL (0x88166018): RTC Enable

NAME	D31
P_RTC_MODE_CTRL	RTC_EN

RTC_EN: RTC Enable

0: Disable

1: Enable

15.3.8 P_RTC_INT_STATUS (0x8816601C): RTC Status

NAME	D22	D21-D20	D19	D18	D17-D16	D15	D14
P_RTC_INT_STATUS	MIN_INTEN		SEC_INT	SEC_INTEN		HSEC_INT	HSEC_INTEN
NAME	**D31**	**D30**	**D29-D28**	**D27**	**D26**	**D25-D24**	**D23**
P_RTC_INT_STATUS	ALM_INT	ALM_INTEN		HOUR_INT	HOUR_INTEN		MIN_INT

HSEC_INTEN: Half-Second Interrupt Enable

0: Disable

1: Enable

HSEC_INT: Half-Second Interrupt Flag

Write 1: Clear the flag

SEC_INTEN: Second Interrupt Enable

0: Disable

1: Enable

SEC_INT: Second Interrupt Flag

Write 1: Clear the flag

MIN_INTEN: Minute Interrupt Enable

0: Disable

1: Enable

MIN_INT: Minute Interrupt Flag

Write 1: Clear the flag

HOUR_INTEN: Hour Interrupt Enable

0: Disable

1: Enable

HOUR_INT: Hour Interrupt Flag

Write 1: Clear the flag

ALM_INTEN: Alarm Interrupt Enable

0: Disable

1: Enable

ALM_INT: Alarm Interrupt Flag

Write 1: Clear the flag

15.3.9 P_TMB_MODE_CTRL (0x88166020): Time Base Control

NAME	D31	D30-D28	D27-D24	D23	D22-D20	D19-D16	D15	D14-D12	D11-D8
P_TMB_MODE_CTRL	TMB0_EN		TMB0_SEL	TMB1_EN		TMB1_SEL	TMB2_EN		TMB2_SEL

TMB2_SEL: Time-Base 2 Period Selection

0000: 1Hz
0001: 2Hz
0010: 4Hz
0011: 8Hz
0100: 16Hz
0101: 32Hz
0110: 64Hz
0111: 128Hz
1000: 256Hz
1001: 512Hz
1010: 1024Hz
1011: 2048Hz

TMB2_EN: Time-Base 2 Enable

0: Disable
1: Enable

TMB1_SEL: Time-Base 1 Period Selection

0000: 1Hz
0001: 2Hz
0010: 4Hz
0011: 8Hz
0100: 16Hz
0101: 32Hz
0110: 64Hz
0111: 128Hz
1000: 256Hz
1001: 512Hz
1010: 1024Hz
1011: 2048Hz

TMB1_EN: Time-Base 1 Enable

0: Disable

1: Enable

TMB0_SEL: Time-Base 0 Period Selection

0000: 1Hz

0001: 2Hz

0010: 4Hz

0011: 8Hz

0100: 16Hz

0101: 32Hz

0110: 64Hz

0111: 128Hz

1000: 256Hz

1001: 512Hz

1010: 1024Hz

1011: 2048Hz

TMB0_EN: Time-Base 0 Enable

0: Disable

1: Enable

15.3.10 P_TMB_INT_STATUS (0x88166024): Time Base Status

NAME	D31	D30	D29-D28	D27	D26	D25-D24	D23	D22	D21-D0
P_TMB_INT_STATUS	TMB0_INT	TMB0_INTEN		TMB1_INT	TMB1_INTEN		TMB2_INT	TMB2_INTEN	

TMB2_INTEN: Time-Base 2 Interrupt Enable

0: Disable

1: Enable

TMB2_INT: Time-Base 2 Interrupt Flag

Write 1: Clear the flag

TMB1_INTEN: Time-Base 1 Interrupt Enable

0: Disable

1: Enable

TMB1_INT: Time-Base 1 Interrupt Flag

Write 1: Clear the flag

TMB0_INTEN: Time-Base 0 Interrupt Enable

0: Disable

1: Enable

TMB0_INT: Time-Base 0 Interrupt Flag

Write 1: Clear the flag

15.3.11 P_TMB_RESET_COMMAND (0x88166028): Time Base Reset

NAME	D31-D0
P_TMB_RESET_COMMAND	RESET_COM

RESET_COM: Time-Base Reset Command

Write 0x5xxxxx5 to reset Time-Base Counter.

16 Watchdog - WDOG

16.1 Introduction

In order to avoid system crash, the watchdog (WDOG) controller is always necessary. When watchdog system is enabled, we need to clear its counter per period (programmable by control registers). If the watchdog counter is not cleared within a specific period, the system will be reset.

16.2 Architecture

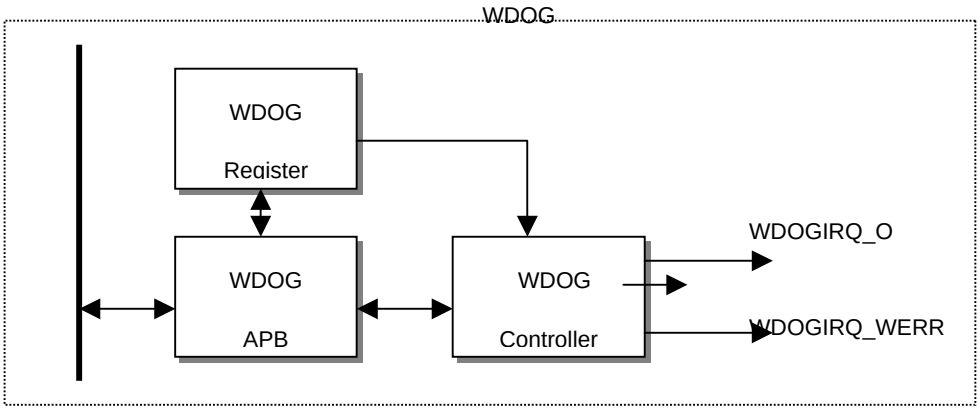


Fig 16-1 WDOG Architecture

16.3 Control Registers

16.3.1 P_WDOG_MODE_CTRL (0x88170000): Watchdog Control

NAME	D31
P_WDOG_MODE_CTRL	WDOG_EN

WDOG_EN: Watchdog Enable

0: Disable

1: Enable

16.3.2 P_WDOG_CYCLE_SETUP (0x88170004): Reset Counter Number

NAME	D31-D0
P_WDOG_CYCLE_SETUP	RESET_COM

RESET_COM: Reset Counter Number

16.3.3 P_WDOG_CLR_COMMAND (0x88170008): Clear Counter Number

NAME	D31-D0
P_WDOG_CLR_COMMAND	CLEAR_COM

CLEAR_COM: Clear Command

Write 0xaxxxxxx5 to clear, and write other values to reset system

17 Sleep/Wakeup

17.1 Sleep

Upon Power-On Reset, CPU starts working. As soon as a sleep command is received, IC will turn off system clock (PLL) and enter sleep mode. After that, program counter will stop at the next address and will not restore until wakeup is activated.

According to the ON/OFF state of PLL, two sleep modes are classified below:

- Wait Mode: The PLL is NOT turned off in this mode.
- Half Mode: The PLL is turned off in this mode.

The control register for entering sleep modes is 0X88210000, refer to Subsection 5.3.1 about the PLLS/CKG.

17.2 Wakeup

Waking up from sleep mode requires a wakeup signal to turn on the system clock (PLL). At the same time, a wakeup IRQ is also generated. The IRQ signal leads CPU to completing the wakeup process as well as initialization. After wakeup interrupt is completed, program counter will continue to execute the next command.

In each sleep mode, the wakeup source can be either key change signals or interrupts from any modules, the clock of which is not turned off. All wakeup sources are listed in the following table.

Wakeup Sources	Description
—	—
—	—
—	—
—	—
MIC INT	ADC Microphone
ADC INT	ADC General-Purpose Interrupt
TB INT	Time Base Interrupt
Timer INT	Timer1 or Timer2 or … Timer 6 event trigger
LCD INT	LCD vertical blanking start
—	—

Wakeup Sources	Description
TVE INT3	TVE light gun hit
CSI INT1	Sensor frame end
CSI INT3	Sensor coordinate hit
CSI INT2	Sensor motion frame end
CSI INT0	Capture done
TVE INT4	TVE coordinate hit
—	—
USB INT	USB device or host Interrupt Please refer to USB for details
SIO INT	SIO Interrupt
SPI INT	SPI Interrupt
UART INT	UART Interrupt
Flash INT	Nand-type Flash Interrupt
SD INT	SD card Interrupt
I^2C INT	I^2C master Interrupt
I^2S INT	I^2S master or I^2S slave Interrupt
APBDMA INT1	APBDMA channel 1 done
APBDMA INT2	APBDMA channel 2 done
LDMDMA INT	LDM DMA done (LDM to DRAM or DRAM to LDM)
BLN Int	Blending DMA done (DRAM to DRAM)
APBDMA INT3	APBDMA channel 3 done
APBDMA INT4	APBDMA channel 4 done
RTC INT	Alarm or HMS (per hour or per minute or per second event)
MPG INT	MPEG4 encoding or decoding a frame has done
ECC INT	The event that ECC has found errors
SFTCFG INT	For the rising or falling event of assigning PIN Please refer to SFTCFG for more information
DEBUG INT	TVE vertical blanking start

Wakeup Sources	Description
KEYCHG INT	Key change Interrupt Refer to PINMUX to know which pins can be used to generate key change signals.
LVD INT	Low voltage detect Interrupt

18 Analog to Digital Converter - ADC

18.1 Introduction

The analog-to-digital converter provides voice record or general-purpose analog to digital conversion functions. There are totally nine channels with one channel dedicated to microphone mode and other channels dedicated to 12-bit general-purpose ADC. The ADC for microphone mode doesn't support Auto Gain Control (AGC) function; therefore extra software is needed. Different from other general ADC channels, microphone channel data can only be transferred by using DMA mode instead of interrupt mode. The general-purpose mode supports auto data sampled at programmable sampling rate and manual data sampled at sampling command. All general-purpose ADCs can read data with interrupt mode only.

Specifications of ADC:

(1) General-purpose application: 2.0mA

(2) Audio application without MICBIAS: 4.0mA / with MICBIAS: 7.5mA

(3) Power-down mode: 1μA

(4) 9-channel 12-bit resolution for audio and general-purpose ADC application

(5) On-chip microphone booster amplifier and programmable gain amplifier (33, 31.5, …, -12, -∞ db)

(6) On-chip anti-aliasing filter and bias-voltage output (MICBIAS)

(7) Reference top voltage and bias current/voltage generators

18.2 Architecture

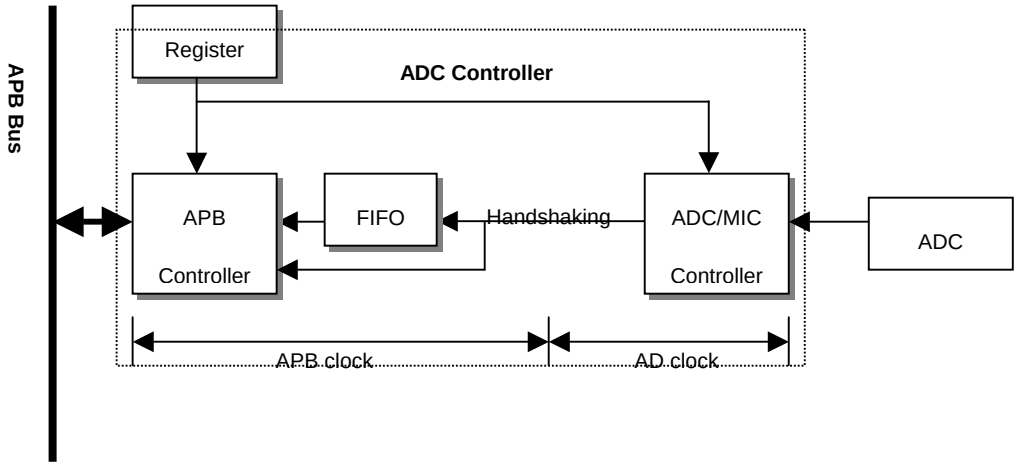


Fig 18-1 ADC Architecture

18.3 Control Registers

18.3.1 P_ADC_MIC_CTRL1 (0x881A0000): ADC Setup

NAME	D31	D30	D29	D28	D27	D26	D25	D24-D4	D3-D0
P_ADC_MIC_CTRL1	ADC_SEL	ADC_EN	MIC_EN		MIC_BOOST	MIC_BIAS	TOP_VOL		CH_SEL

CH_SEL: Channel Selection

0000: GPIO 0
0001: GPIO 1
0010: GPIO 2
0011: GPIO 3
0100: GPIO 4
0101: GPIO 5
0110: GPIO 6
0111: GPIO 7
1xxx: MIC

TOP_VOL: Top reference voltage internal or external

0: Internal
1: External

MIC_BIAS: MIC bias enable (only for MIC mode)

0: Disable
1: Enable

MIC_BOOST: MIC boost enable (only for MIC mode)

0: Disable
1: Enable

MIC_EN: MIC enable

0: Disable
1: Enable

ADC_EN: ADC enable

0: Disable
1: Enable

ADC_SEL: ADC Selection

0: Disable
1: Enable

18.3.2 P_ADC_MIC_GAIN (0x881A0004): MIC Gain

NAME	D4-D0
P_ADC_MIC_GAIN	MIC_GAIN

MIC_GAIN: MIC Gain value

00000: 33.0 db
00001: 31.5 db
00010: 30.0 db
…..
10101: 1.5 db
10110: 0.0 db
10111: -1.5 db
….
11110: -12.0 db
11111: -∞

18.3.3 P_ADC_SAMPLE_CLK (0x881A0008): ADC Clock Setup

NAME	D15-D0
P_ADC_SAMPLE_CLK	ADC_CLOCK_CYCLE

ADC_CLOCK_CYCLE: ADC clock cycle is ACC ([15:0]) + 1

18.3.4 P_ADC_SAMPLE_HOLD (0x881A000C): Sample Hold Setup

NAME	D31-D28	D27-D16	D15-D0
P_ADC_SAMPLE_HOLD	HOLD_WIDTH		HOLD_CYCLE

HOLD_CYCLE: Sample Hold Cycle: [15:0] + 1 (>= 15)

HOLD_WIDTH: Sample Hold Width: [3:0] width (>= 2)

NOTE: Let's illustrate the method about setting up ADC clock cycle and sample-hold width/cycle. If we need 44.1K sample rate for ADC and ADC controller clock is 33.8688MHz, sample cycle = 33.8688M / 44.1K = 768. If we need 16 Sample Hold cycles, ADC clock cycle = 768 / 16 = 48. Therefore, ACC = 47, SampleHoldCycle = 15, and SampleHoldWidth = 2.

18.3.5 P_ADC_MIC_CTRL2 (0x881A0010): ADC Controller

NAME	D23	D22	D21-D7	D6-D4	D3	D2-D0
P_ADC_MIC_CTRL2	AUTO_CLR			FIFO_INT		FIFO_RD
NAME	D31	D30-D28	D27	D26	D25	D24
P_ADC_MIC_CTRL2	ADC_EN		AUTO_SAMPLE	MIC_MODE	MUTE_SEL	DMA_SIZE

ADC_EN: ADC Controller enable

0: Disable

1: Enable

AUTO_SAMPLE: Auto sampling for general-purpose mode

0: Manual

1: Auto

MIC_MODE: Data unsigned/signed selection for MIC mode

0: Signed

1: Unsigned

MUTE_SEL: Mute selection for MIC mode

0: Normal

1: Mute

DMA_SIZE: DMA data size for MIC mode

0: 32 bits

1: 16 bits

AUTO_CLR: Auto clear irq flag for general-purpose mode

0: Disable

1: Enable

FIFO_INT: FIFO receive depth for interrupt level

0: Receiving one will irq

1: Receiving two will irq

2: Receiving three will irq

3: Receiving four will irq

4: Receiving five will irq

5: Receiving six will irq

6: Receiving seven will irq

7: Receiving eight will irq

FIFO_RD: FIFO receive depth read

18.3.6 P_ADC_INT_STATUS (0x881A0014): ADC Controller 2

NAME	D25-D24	D23	D22	D21	D20-D16	D15
P_ADC_INT_STATUS		FIFO_OFE	FIFO_FULL	FIFO_EMPTY		CONVERT_ADC
NAME	D31	D30	D29	D28	D27	D26
P_ADC_INT_STATUS	ADC_INT	ADC_INTEN	ADC_AINT	ADC_AINTEN	MIC_OFINT	MIC_OFINTEN

ADC_INT: ADC Manual Interrupt Flag

Read 0: Occurs

Write 1: Clear the flag

ADC_INTEN: ADC Manual Interrupt Enable

0: Disable

1: Enable

ADC_AINT: ADC Auto Interrupt Flag

Read 0: Occurs

Write 1: Clear the flag

ADC_AINTEN: ADC Auto Interrupt Enable

0: Disable

1: Enable

MIC_OFINT: MIC Overflow Interrupt Flag

Read 0: Occurs

Write 1: Clear the flag

MIC_OFINTEN: MIC Overflow Interrupt Enable

0: Disable

1: Enable

FIFO_OFE: FIFO receive overflow Error

Read 0: Occurs

Write 1: Clear the flag

FIFO_FULL: FIFO Full Flag

0: Not Full

1: Full

FIFO_EMPTY: FIFO Empty Flag

0: Not Empty

1: Empty

CONVERT_ADC: Start Conversion for ADC manual mode

Write 1: Start Conversion

18.3.7 P_ADC_MANUAL_DATA (0x881A0018): ADC Data Read

NAME	D31-D20
P_ADC_MANUAL_DATA	ADC_Self_Data

18.3.8 P_ADC_AUTO_DATA (0x881A001C): ASP Data Read

NAME	D31-D20
P_ADC_AUTO_DATA	ADC_Auto_Data

18.3.9 P_ADC_MIC_DATA (0x881A0020): MIC Data Read

NAME	D31-D0
P_ADC_MIC_DATA	MIC_Data

19 Interactive Information Services - I2S

19.1 Introduction

I^2S is a protocol for digital stereo audio. The I^2S controller embedded in SPCE3200 supports the receive function in the master/slave mode and the maximum receive data size is 32-bit for left/right channel.

19.2 Architecture

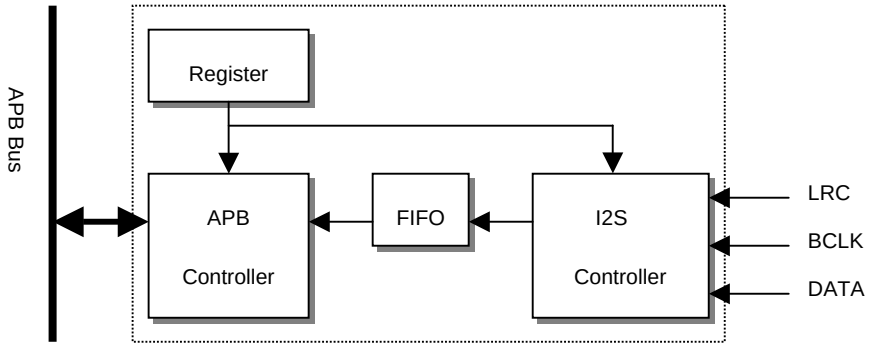


Fig 19-1 I^2S Architecture

19.3 Control Registers

19.3.1 P_I2S_MODE_CTRL (R/W)(0x88140000): I^2S Control

NAME	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22-D0
P_I2S_MODE_CTRL	RX_EN	MASTER	LJM	LRF	LCE	RCE	DMA	DMASIZE	AUTO	

AUTO: Auto Interrupt flag clear for IRQ mode

0: Disable

1: Enable

DMASIZE: DMA data size

0: 32-bit

1: 16-bit

DMA: DMA /IRQ mode selection

0: IRQ mode

1: DMA mode

RCE: Right channel enable

0: Disable

1: Enable

LCE: Left channel enable

0: Disable

1: Enable

LRF: Left / Right first receive selection

0: Left

1: Right

LJM: I^2S / Left justified mode select

0: I^2S mode

1: Left mode

MASTER: I^2S master mode / slave mode

0: Slave mode

1: Master mode

RX_EN: I^2S Receiver Enable

0: Disable

1: Enable

19.3.2 P_I2S_MASTER_SETUP (R/W)(0x88140004): I^2S IRQ Status

NAME	D31-D30	D29-D8	D7-D0
P_I2S_MASTER_SETUP	LRF_SEL	-	BCLK_SET

BCLK_SET: (value + 1)

LRF_SEL: L/R frame data size selection

00: 16-bit per L/R frame

01: 24-bit per L/R frame

10: 32-bit per L/R frame

11: Reserved

19.3.3 P_I2S_INT_STATUS (R/W)(0x88140008): I^2S IRQ Status

NAME	D31	D30	D29	D28-D0
P_I2S_INT_STATUS	INT_FLAG	INT_EN	FIFO_OVEN	

INT_EN: I^2S interrupt enable

0: Disable

1: Enable

INT_FLAG: I^2S interrupt flag

Read 1: IRQ occurred

Write 1: Clear IRQ

FIFO_OVEN: I^2S FIFO overflow enable

0: Disable

1: Enable

19.3.4 P_I2S_FIFO_STATUS (R/W)(0x8814000C): I^2S FIFO Config

NAME	D31	D30	D29	D28-D5	D6-D4	D3	D2-D0
P_I2S_FIFO_STATUS	OV_ERR	FULL_FLAG	EMPTY_FLAG		RX_INT_LV		RX_LV

RX_LV: FIFO receiver level

RX_INT_LV: FIFO receive depth for interrupt level

000: Receiving one will irq

001: Receiving two will irq

010: Receiving three will irq

011: Receiving four will irq

100: Receiving five will irq

101: Receiving six will irq

110: Receiving seven will irq

111: Receiving eight will irq

EMPTY_FLAG: FIFO empty flag

0: Not empty

1: Empty

FULL_FLAG: FIFO full flag

0: Not full

1: Full

OV_ERR: FIFO receive overflow error

Read 1: Error occurs

Write 1: Clear

19.3.5 P_I2S_RX_DATA (R/W)(0x88140010): I^2S Receive Data

NAME	D31-D0
P_I2S_RX_DATA	I2S_RDATA

For IRQ mode, 16-bit data is legal; for DMA mode, 16/32-bit data is legal.

20 Inter-Integrated Circuit - I2C

20.1 Introduction

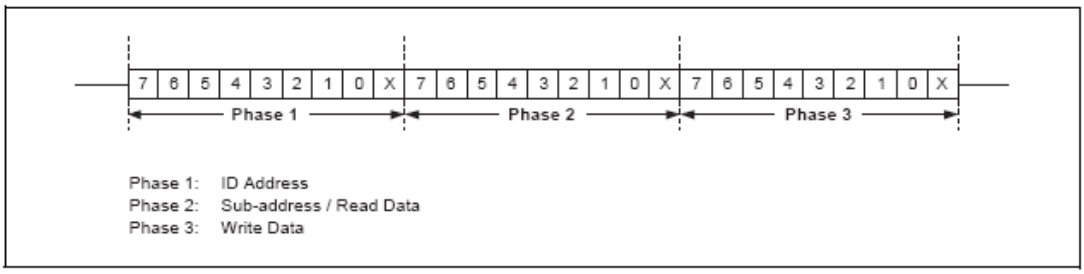
I^2C is a two-wire, bidirectional serial bus that provides a simple, efficient method for data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance among many devices. The flexible I^2C allows additional devices to be connected to the bus for expansion and system development.

20.2 Initialization

Before starting I^2C, initialization must be done. There are 2 major jobs:

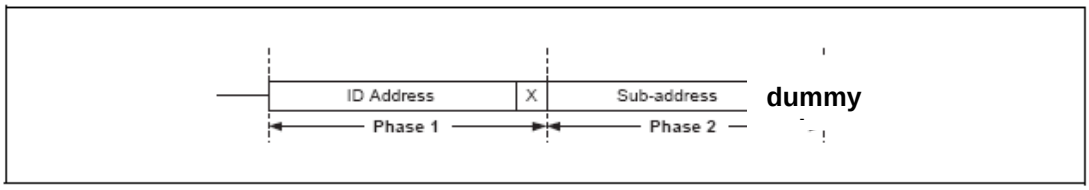
- Decide the I^2C clock rate by configuring P_I2C_RATE_SETUP
- Decide the I^2C slave ID Address by configuring P_I2C_SLAVE_ADDR

The SPCE3200 supports I^2C master only and 3 types of transactions: 8 bits type, 16 bits type, and 8 bits * n type. For 8 bits type, writing P_I2C_MODE_CTRL [0] can start the transaction and reading P_I2C_MODE_CTRL [3] can get the result (if INT_EN is 1, I^2C interrupt will happen after a full transaction). Here is an example for I^2C 8 bits writing:

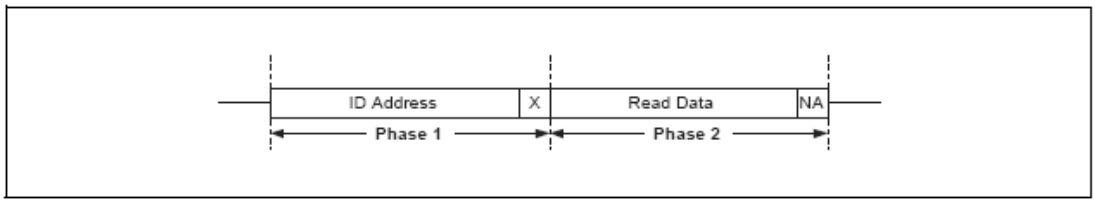


0^{th} bit in ID Address must be 0.

Besides, here is an example for I^2C 8 bits reading:

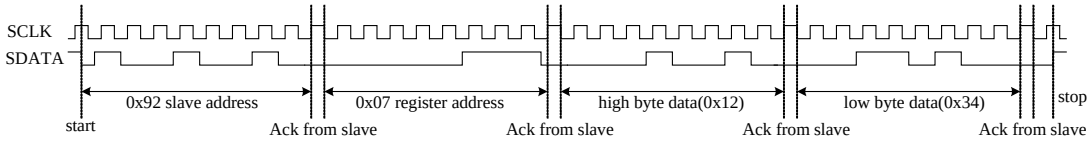


0^{th} bit in ID Address must be 0.

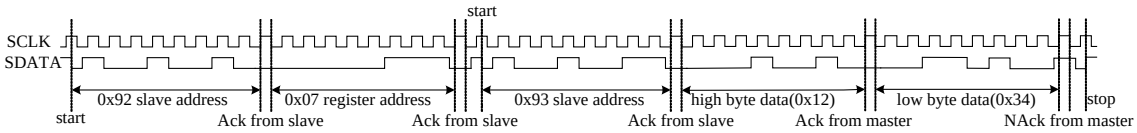


0th bit in ID Address must be 1.

For 16 bits type, writing P_I2C_MODE_CTRL [1] can start the transaction and reading P_I2C_MODE_CTRL [4] can get the result (if INT_EN is 1, I2C interrupt will happen after a full transaction). Here is an example for I2C 16 bits writing:



Here is an example for I2C 16 bits reading:



For 8 bits * n type transaction, each transaction can be considered as each phase transaction. The definition of “phase transaction” is presented as above. Writing P_I2C_MODE_CTRL [2] can start each phase transaction and reading P_I2C_MODE_CTRL [5] can get the result (if INT_EN is 1, I2C interrupt will happen after each phase transaction). In addition, writing P_I2C_MODE_CTRL [8] can stop the full transaction.

20.3 Control Registers

20.3.1 P_I2C_MODE_CTRL (R/W)(0x88130020): I2C Config

NAME	D8	D7	D6	D5	D4	D3	D2	D1	D0
P_I2C_MODE_CTRL	STOPN	GAK	TRANS_MODE	ACKN	ACK16	ACK8	STARTN	START16	START8

START8: Start 8-bit transfer

START16: Start 16-bit transfer

STARTN: Start general transaction

ACK8: ACK from 8-bit transfer

ACK16: ACK from 16-bit transfer

ACKN: ACK from general transaction

TRANS_MODE: Read/ Write Mode selection

0: Write Mode

1: Read Mode

GAK: ACK/NAK mode selection

0: Generate NAK at last transaction

1: Generate ACK at last transaction

STOPN: Stop general transaction

20.3.2 P_I2C_INT_STATUS (R/W)(0x88130024): I²C Interrupt

NAME	D1	D0
P_I2C_INT_STATUS	INT_EN	INT_FLAG

INT_FLAG: Interrupt flag

INT_EN: Interrupt enable bit

20.3.3 P_I2C_RATE_SETUP (R/W)(0x88130028): I²C Clock Setting

NAME	D9-D0
P_I2C_RATE_SETUP	CLK_SET

20.3.4 P_I2C_SLAVE_ADDR (R/W)(0x8813002C): I²C ID

NAME	D7-D1
P_I2C_SLAVE_ADDR	ID_REG

20.3.5 P_I2C_DATA_ADDR (R/W)(0x88130030): I²C Port Address

NAME	D7-D0
P_I2C_DATA_ADDR	ADDR

20.3.6 P_I2C_TX_DATA (R/W)(0x88130034): I²C Write Data

NAME	D15-D0
P_I2C_TX_DATA	I2C_WDATA

20.3.7 P_I2C_RX_DATA (R/W)(0x88130038): I²C Read Data

NAME	D15-D0
P_I2C_RX_DATA	I2C_RDATA

21 Serial Peripheral Interface - SPI

21.1 Introduction

A Serial Peripheral Interface (SPI) controller is built in SPCE3200 to facilitate communicating with other devices and components.

The SPI performs parallel-to-serial conversion on data written to an internal 8-bit transmit buffer. It also performs serial-to-parallel conversion on the serial input data, and buffers it in an 8-bit receive buffer. The SPI asserts interrupts to request service to both transmit buffer and receive buffer, and to indicate an overrun situation in the receive buffer.

The SPI includes a programmable bit rate clock divider and pre-scalar to generate the serial output clock SCLK from the internal clock. The operating mode, frame format and size are programmable by setting control register P_SPI_Ctrl. The SSB output operates as an active LOW slave for SPI and Micro wire.

21.2 Features

There are four control signals on SPI, SPICSN, SPICLK, SPIRX, and SPITX. Main features of SPI include:

- Supports Master / Slave mode for single byte and consecutive bytes transfer.
- Supports overrun error indication
- Supports transmitting / receiving interrupt request
- Programmable phase and polarity of master clock
- Selectable data sampling time
- Programmable master SCK clock frequency (System clock /2, /4, /8, /16, /32, /64, /128)
- Built-in 8-depth 8-bit FIFO in both transmit and receive direction (the interrupt level of these two FIFOs is programmable)

21.3 Architecture

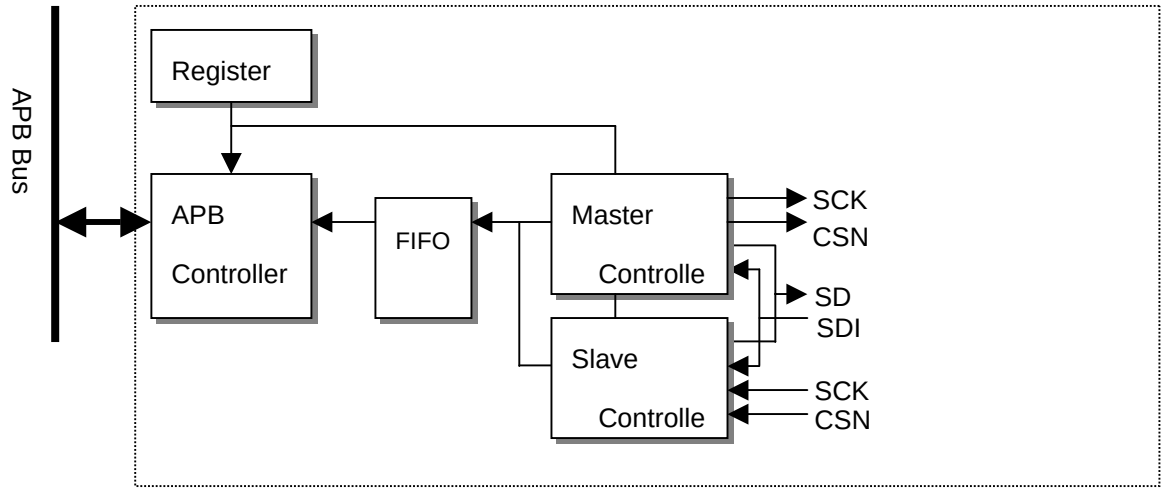


Fig 21-1 SPI Architecture

21.4 Control Registers

21.4.1 P_SPI_MODE_CTRL (0x88110000): SPI Control

NAME	D31	D30-D28	D27	D26	D25	D24-D8	D7	D6	D5-D3	D2-D0
P_SPI_MODE_CTRL	SPI_EN		SPI_LB	SOFT_REST	SPI_MODE		SPI_CK_PH	SPI_CK_PO		SPI_CLOCK

SPI_CLOCK: Master Clock Selection

000: PCLK/2
001: PCLK/4
010: PCLK/8
011: PCLK/16
100: PCLK/32
101: PCLK/64
110: PCLK/128

SPI_CK_PO: Refer to timing diagram

SPI_CK_PH: Refer to timing diagram

SPI_MODE: SPI Master/Slave Mode Selection

0: Master
1: Slave

SOFT_REST: SPI Software Reset

0: No effect
1: Reset

SOFT_LB: SPI Loop Back Mode Selection (Master)

0: Normal

1: SPIRX= SPITX

SOFT_EN: SPI Enable

0: Disable

1: Enable

21.4.2 P_SPI_TX_STATUS (0x88110004): SPI Transmit Status

NAME	D31	D30	D29-D28	D27	D26-D7	D6-D4	D3	D2-D0
P_SPI_TX_STATUS	TX_FLAG	TX_EN		TX_EMPTY_FLAG		TX_FIFO_INT_LEVEL		FIFO_DATA_LEVEL

FIFO_DATA_LEVEL: SPI Tx FIFO Data Level

000: data number < 1, interrupt will happen

001: data number < 2, interrupt will happen

010: data number < 3, interrupt will happen

011: data number < 4, interrupt will happen

100: data number < 5, interrupt will happen

101: data number < 6, interrupt will happen

110: data number < 7, interrupt will happen

111: data number < 8, interrupt will happen

TX_EMPTY_FLAG: SPI Tx FIFO Empty Flag

0: Not Empty

1: Empty

TX_EN: SPI Tx Interrupt Enable

0: Disable

1: Enable

TX_FLAG: SPI Tx Interrupt Flag

Read 0: Doesn't happen

Read 1: Interrupt happens

Write 1: Clear Flag

21.4.3 P_SPI_TX_DATA (0x88110008): SPI Transmit Data

NAME	D7-D0
P_SPI_TX_DATA	SPI_TX_DATA

21.4.4 P_SPI_RX_STATUS (0x8811000C): SPI Receive Status

NAME	D31	D30	D29-D28	D27	D26	D25-D5	D6-D4	D3	D2-D0
P_SPI_RX_STATUS	R_FLAG	RX_EN		R_EMPTY_FLAG	RX_OR_ERR		R_FIFO_INT_LEVEL		FIFO_DATA_LEVEL

FIFO_DATA_LEVEL: SPI Rx FIFO Data Level

000: one data in FIFO, interrupt will happen
001: two data in FIFO, interrupt will happen
010: three data in FIFO, interrupt will happen
011: four data in FIFO, interrupt will happen
100: five data in FIFO, interrupt will happen
101: six data in FIFO, interrupt will happen
110 seven data in FIFO, interrupt will happen
111: eight data in FIFO, interrupt will happen

RX_OR_ERR: SPI Rx FIFO overrun error

Read 0: Doesn't happen
Read 1: Happens
Write 1: Clear flag

RX_FULL_FLAG: SPI Rx FIFO Full Flag

0: Not Full
1: Full

RX_EN: SPI Rx Interrupt Enable

0: Disable
1: Enable

Rx_FLAG: SPI Rx Interrupt Flag

Read 0: Doesn't happen
Read 1: Interrupt happens
Write 1: Clear Flag

21.4.5 P_SPI_RX_DATA (0x88110010): SPI Receive Data

NAME	D7-D0
P_SPI_RX_DATA	SPI_RX_DATA

21.4.6 P_SPI_TXRX_STATUS (0x88110014): SPI Transmit/Receive Status

NAME	D31	D30	D29-D5	D4	D3	D2	D1	D0
P_SPI_TXRX_STATUS	SPI_OW_MODE	SPI_SMART		SPI_BUSY	RX_FULL	RX_EMPTY	TX_FULL	TX_EMPTY

TX_EMPTY: SPI Transmit FIFO empty flag

0: Not empty

1: Empty

TX_FULL: SPI Transmit FIFO Full flag

0: Full

1: Not full

RX_EMPTY: SPI receiver FIFO Empty flag

0: Empty

1: Not empty

RX_FULL: SPI receiver FIFO full flag

0: Not full

1: Full

SPI_BUSY: SPI busy status

0: Idle

1: Busy

SPI_SMART: SPI smart mode

0: Manual Clear IRQ

1: Auto Clear IRQ

SPI_OW_MODE: SPI Overwrite mode

0: Skip data when OV

1: Overwrite data when OV

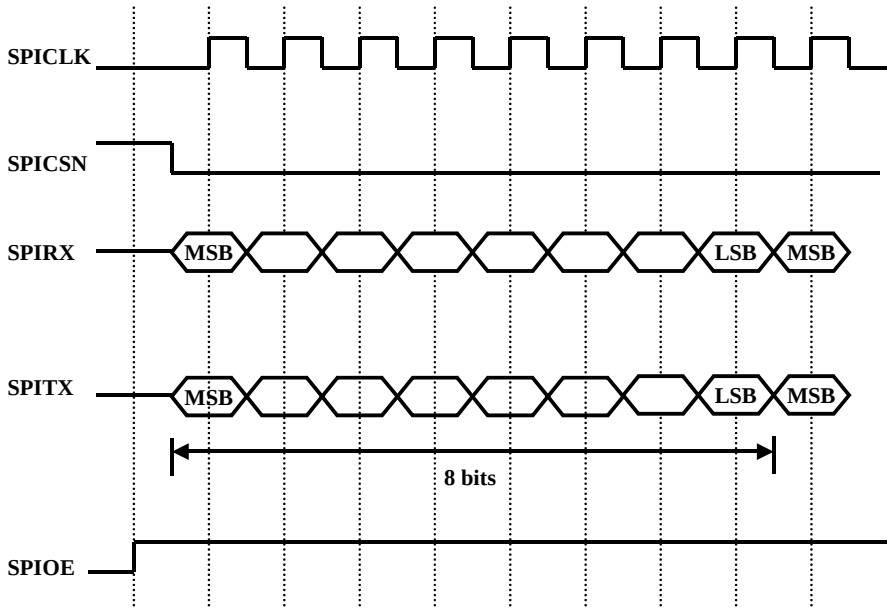


Fig 21-2 SPI Data Receiving/Transmitting Timing (Master Mode, SPO=0, SPH=0)

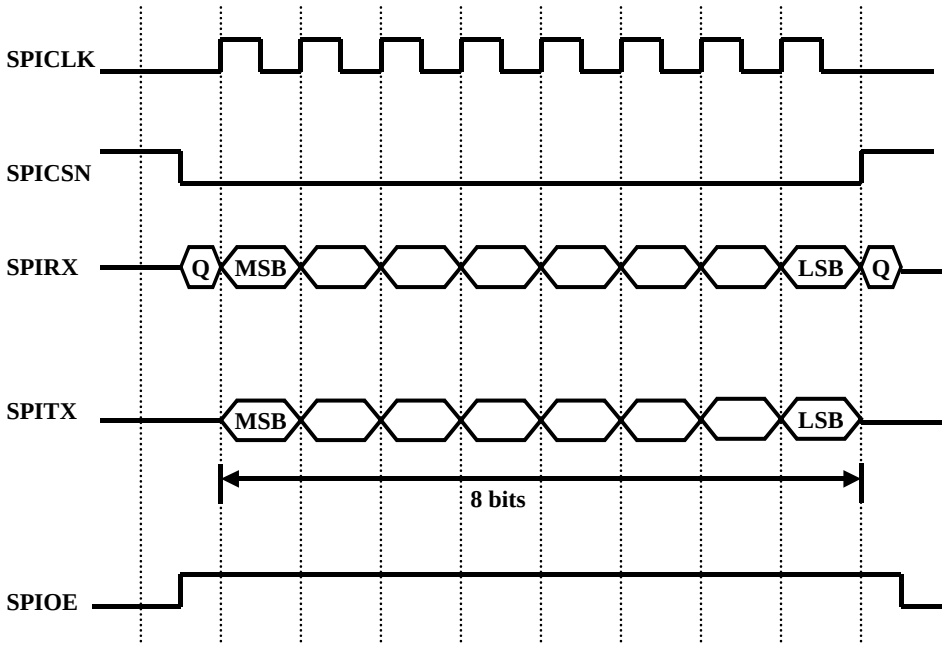


Fig 21-3 SPI Data Receiving/Transmitting Timing (Master Mode, SPO=0, SPH=1)

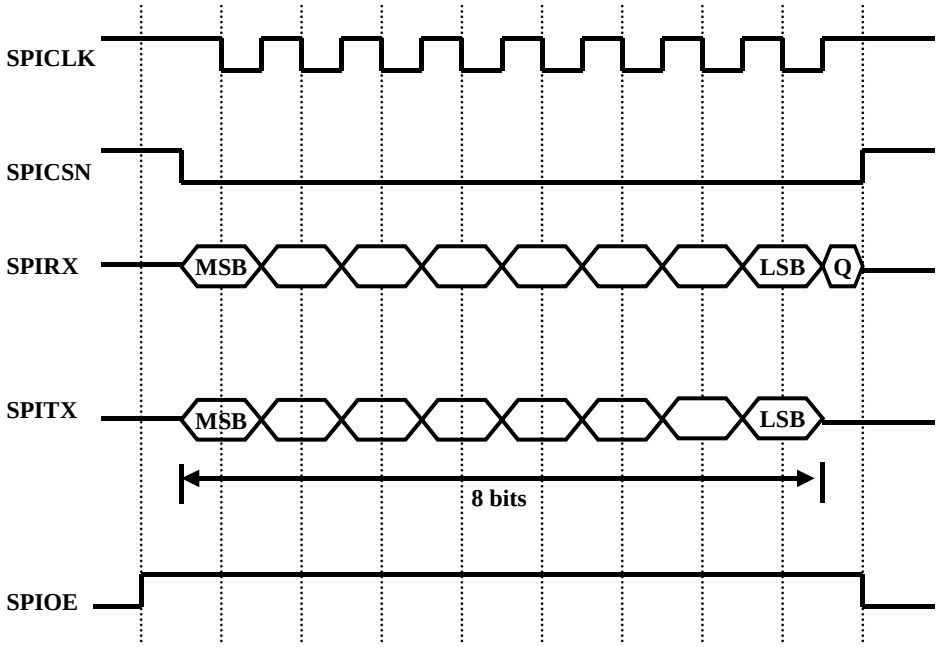


Fig 21-4 SPI Data Receiving/Transmitting Timing (Master Mode, SPO=1, SPH=0)

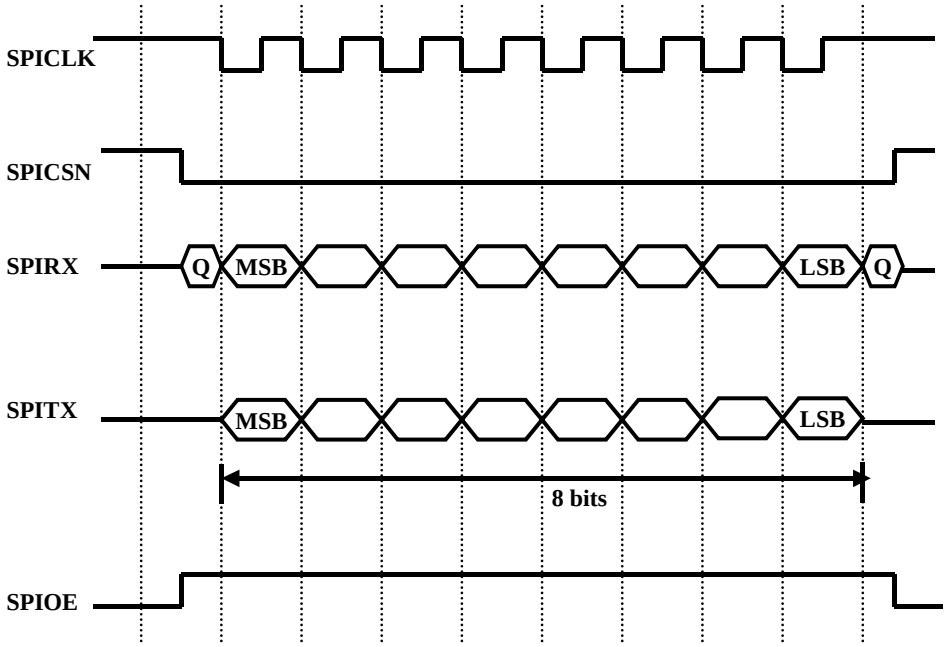


Fig 21-5 SPI Data Receiving/Transmitting Timing (Master Mode, SPO=1, SPH=1)

22 Serial Interface I/O - SIO

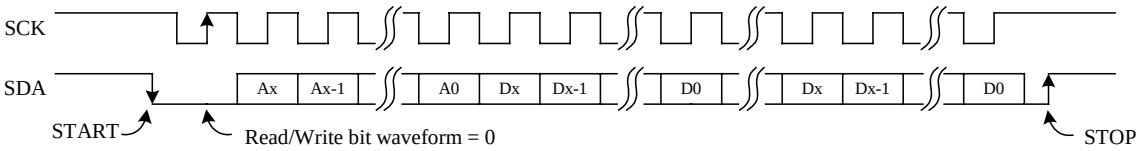
22.1 Introduction

SIO (Serial Interface I/O) is Sunplus's proprietary serial interface, which can be used to communicate with other devices. This serial interface is capable of transmitting or receiving data via 2 pins, SCK and SDA.

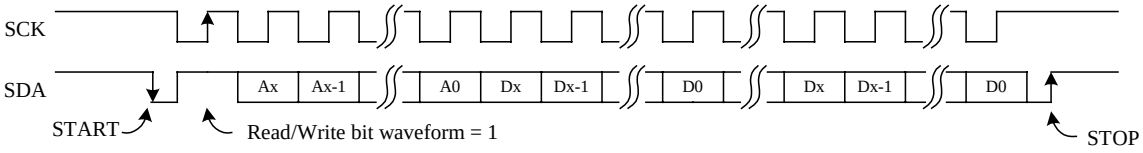
22.2 SIO Protocol

The SIO protocol is composed of start bit, read/write control bit, address word, data word, and stop bit. The figures below show the SIO protocol for the read and write mode respectively. If the SDA makes a transition from 1 to 0 when the SCK is high, it stands for the "start bit" of the SIO transfer. If the SDA makes a transition from 0 to 1 when the SCK is high, it represents the "stop bit". In the SIO waveform other than the "start & stop bit", the SDA can only change its logic level when the SCK is low. Keep in mind that the address word & data word are transmitted with MSB first. And if the data word is 16 bits in width, the **low** byte will be transmitted/received first.

SIO Write Mode :



SIO Read Mode :



22.3 Features

The SIO interface provides the following features:

- Supports "master read" and "master write" function via SCK and SDA
- Supports 4 address modes (No address, 8-bit address, 16-bit address, 24-bit address)
- Supports burst reading or writing function
- Supports both 8- and 16-bit data width
- Supports interrupt and polling function
- Supports 4 SIO baud rates

- Provides automatic bit-stream transmitting mode when communicating with SPDS301 IC

22.4 Control Registers

22.4.1 P_SIO_MODE_CTRL (0x88120000): SIO Control

NAME	D7	D6-D3		D2	D1	D0
P_SIO_MODE_CTRL	SIO_IRQ_CLR			SIO_TRANSFER	SIO_RW_CTL	SIO_START
NAME	D15	D14	D13	D12	D11-D10	D9-D8
P_SIO_MODE_CTRL	SIO_DATA_WIDTH	APBDMA_EN	IRQ_EN	PROTOCOL	BAUDRATE	ADDR_MODE
NAME	D23-D16					
P_SIO_MODE_CTRL						
NAME	D31	D30	D29-D24			
P_SIO_MODE_CTRL	SIO_REQUEST	SIO_IRQ_STS				

SIO_START: SIO Start/Stop control bit

0: Stop the SIO transmitting/receiving

1: Start the SIO transmitting/receiving

SIO_RW_CTL: SIO Read/Write control bit

0: SIO read mode

1: SIO write mode

SIO_TRANSFER: Transfer data low byte first

0: SIO transfer Data Register High Portion (0x8812000D) before Data Register Low Portion (0x8812000C)

1: SIO transfer Data Register Low Portion (0x8812000C) before Data Register High Portion (0x8812000D)

SIO_IRQ_CLR: SIO Interrupt clear bit

ADDR_Mode: SIO addressing mode selection bits

00 = 16-bit address

01 = No address

10 = 8-bit address

11 = 24-bit address

BAUDRATE: SIO baud rate selection bits

00 = 27/16 MHz

01 = 27/4 MHz

10 = 27/8 MHz

11 = 27/32 MHz

PROTOCOL: SIO protocol read/write bit waveform definition. The definition of the SIO protocol read/write bit is shown in the "SIO Protocol" section.

0 = In write mode, the read/write bit waveform will be low.
In read mode, the read/write bit waveform will be high.

1 = In both write and read modes, the read/write bit waveform will be low.

IRQ_EN: SIO interrupt enable control bit

0 = SIO polling method

1 = SIO interrupt enable. The SIO interrupt uses IRQ44.

APBDMA_EN: APBDMA enable

0 = SIO transfers data by CPU.

1 = SIO transfers data by APBDMA (for details, refer to APBDMA).

SIO_DATA_WIDTH: SIO data width control bit

0 = 8 bits

1 = 16 bits

SIO_IRQ_STS: SIO interrupt status bit (read only)

1: SIO interrupt occurs

0: SIO interrupt doesn't occur

SIO_REQUEST: SIO data transfer request bit (read only)

This bit is useful when you use the polling method instead of the interrupt method. If this bit is read as 1 in the write mode, it means the SIO interface circuit is waiting for users to write an 8-/16-bit data into SIO_DATA register. If this bit is read as 1 in the read mode, however, it means the SIO interface circuit has received an 8-/16-bit data and stores it in the SIO_DATA register. Users should read the SIO_DATA register to get the received data.

Note: If "**automatic bit-stream transmitting mode**" is turned on, this bit will always be 0.

22.4.2 P_SIO_AUTO_CTRL (0x88120004): Automatic Transmit Word Number

NAME	D7-D4	D3-D2	D1	D0
P_SIO_AUTO_CTRL	SIO_WORD_NUM		DS301_READY	AUTOMATIC_EN

AUTOMATIC_EN: Automatic bit-stream transmitting mode for the SPDS301

0: Disable the automatic bit-stream transmitting mode

1: Enable the automatic bit-stream transmitting mode

The SIO interface circuit will keep polling the status register of SPDS301 until the "decode work bit" and "request ready bit" are both 1. Upon the completion of the polling, an SIO interrupt will occur, and users should write a bit-stream data word to the SIO_DATA register to clear this interrupt. In summary, each time the SIO interrupt occurs during the automatic bit-stream transmitting mode, just write bit-stream data into the SIO_DATA register. For the detailed information of the SPDS301, please refer to the *SPDS301 Programming Guide*.

DS301_READY: Use DS301_ready pin

0 = Not used

1 = Used

The DS301_ready pin may locate at IOB[0] or IOC[10] according to the SIO port selection setting. This bit is effective only when "automatic bit-stream transmitting mode for the SPDS301" is turned on. For proper use of this control bit, refer to the "SIO Usage Procedure" section.

SIO_WORD_NUM: Automatic Transmitting Word Number

Number of data word that will be transmitted to SPDS301 in one package. Default value is 15. This register is effective only when "automatic bit-stream transmitting mode for the SPDS301" is turned on. The value of this register should **not** be set as 0. Otherwise, it will cause un-predictable result.

22.4.3 P_SIO_DATA_ADDR (0x88120008): SIO Start Address

NAME	D23-D16	D15-D8	D7-D0
P_SIO_DATA_ADDR	SIO_ADDRH	SIO_ADDRM	SIO_ADDRL

SIO_ADDRL: SIO Start Address Register Low Portion.

It is used when addressing mode is not equal to 01.

SIO_ADDRM: SIO Start Address Register Middle Portion.

It is used when addressing mode is equal to 00 or 11.

SIO_ADDRH: SIO Start Address Register High Portion

It is used when addressing mode is equal to 11.

22.4.4 P_SIO_TXRX_DATA (0x8812000C): SIO Data

NAME	D15-D8	D7-D0

P_SIO_TXRX_DATA	SIO_DATAH	SIO_DATAL

SIO_DATAL: SIO Data Register Low Portion

SIO_DATAH: SIO Data Register High Portion

It is used when data width is equal to 1.

22.5 SIO Interrupt Processing

The SIO interrupt signal is controlled by the 0x8812_0000[13]. If 0x8812_0000[13] is 1, the SIO interrupt is enabled. Note that setting 0x8812_0000[7] could clear the SIO interrupt signal. Following tables depict what the proper actions should be.

Not using "Automatic Bit-Stream Transmitting Mode"	
SIO Status when the Interrupt Occurs	**Proper Action**
If SIO interrupt is generated during the "Read Mode", it means the SIO interface has received a data byte/word.	• Read the SIO_DATA register and then set the interrupt clear register to clear interrupt so that the SIO interface will continue receiving data. • Even clear the start/stop control bit of the SIO, the interrupt should still be cleared by setting the interrupt clear register.
If SIO interrupt is generated during the "Write Mode", it mans the SIO interface is waiting for data to be transmitted.	• Writing data to the SIO_DATA register will generate an interrupt after the data is transfered. Setting the interrupt clear register can clear interrupt. • Even clear the start/stop control bit of the SIO, the interrupt should still be cleared by setting the interrupt clear register.

Using "Automatic Bit-Stream Transmitting Mode"	
SIO Status when the Interrupt Occurs	**Proper Action**
If SIO interrupt is generated, it means the SIO interface is waiting for data to be transmitted.	Writing data to the SIO_DATA register will generate an interrupt after the data is transfered. Set the interrupt clear register to clear interrupt

22.6 SIO Usage Procedure

■ **Not** using “automatic bit-stream transmitting mode”

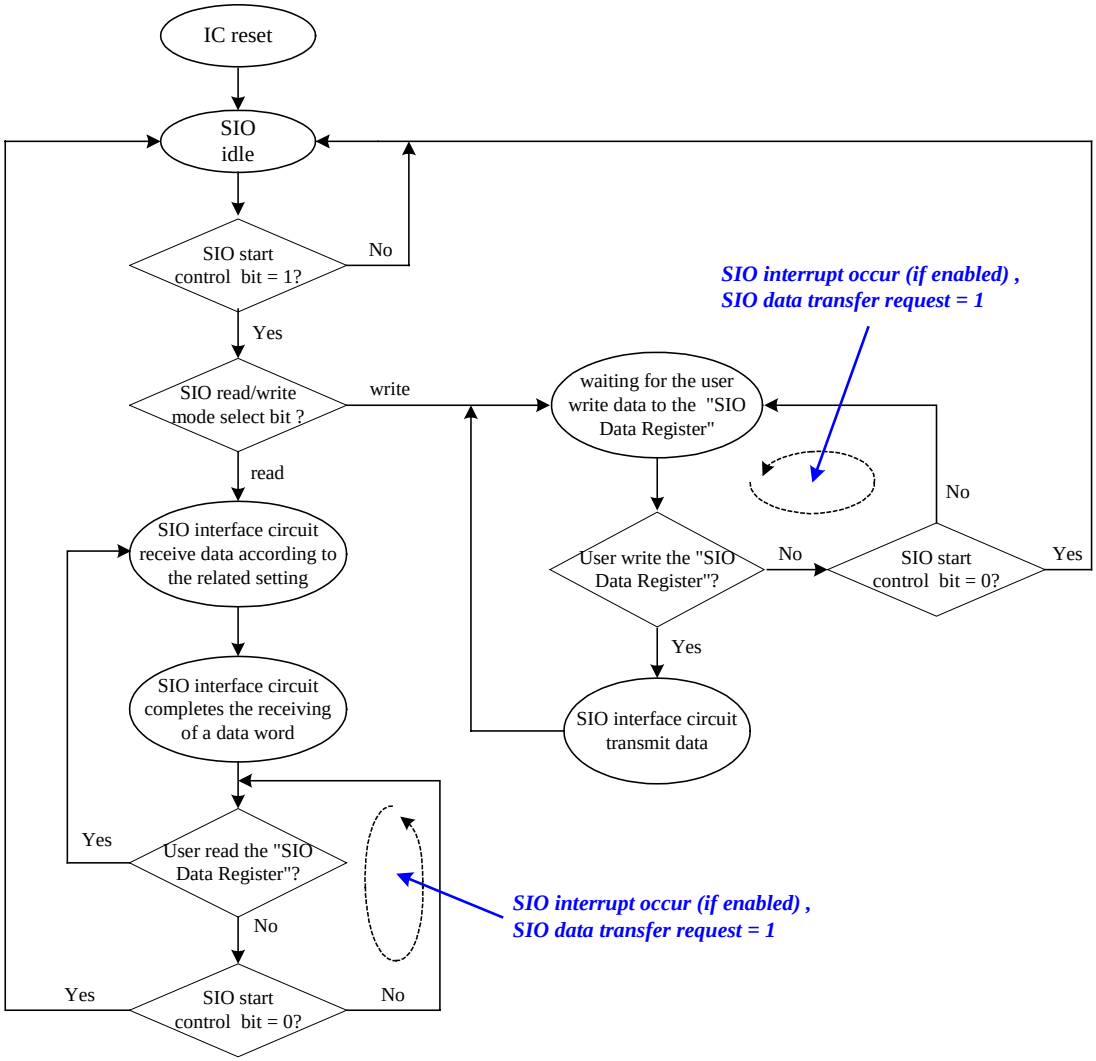


Fig 22-1 SIO Application 1

- Using “automatic bit-stream transmitting mode” & **not** using DS301_ready pin

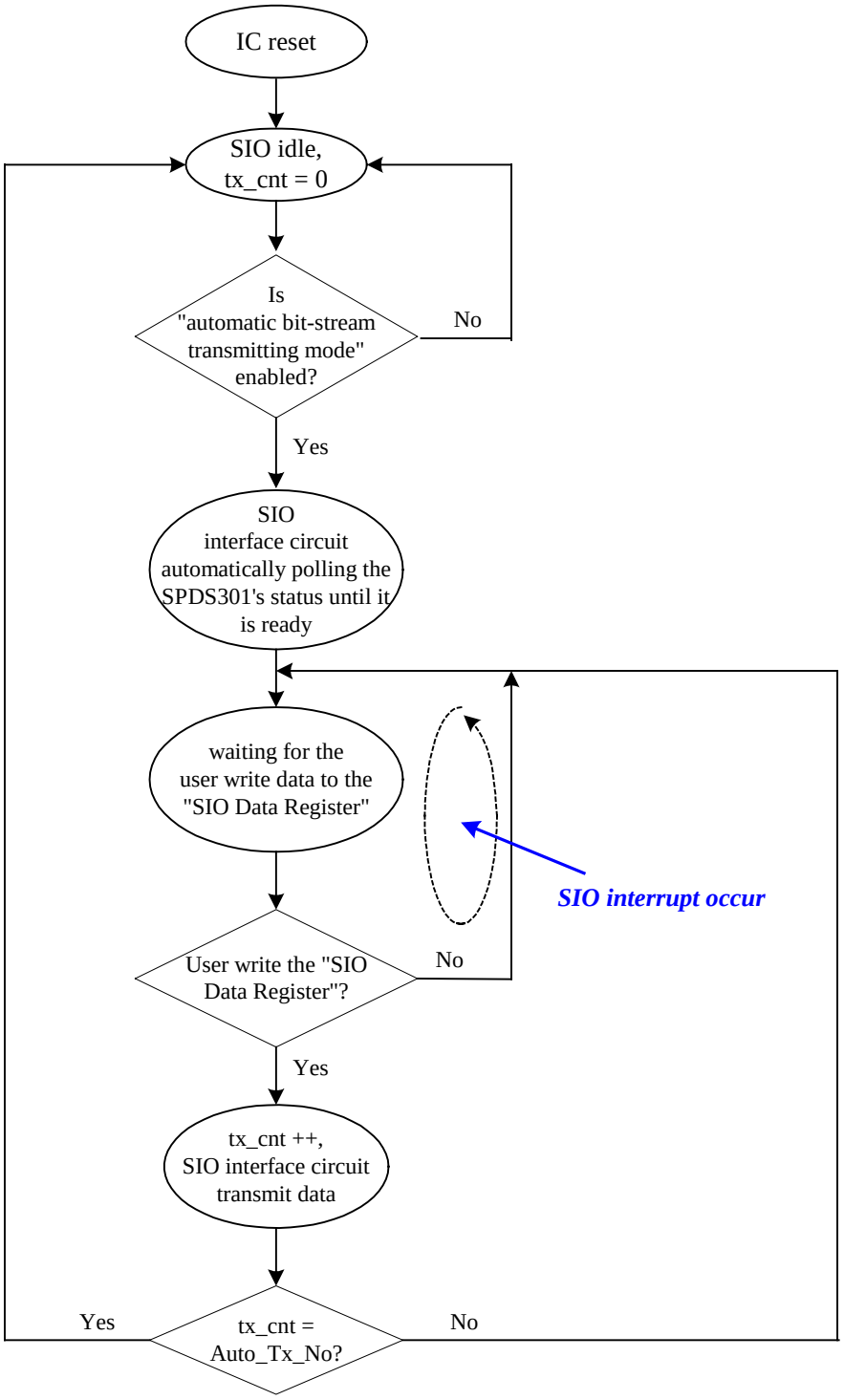


Fig 22-2 SIO Application 2

- Using both “automatic bit-stream transmitting mode” & DS301_ready pin

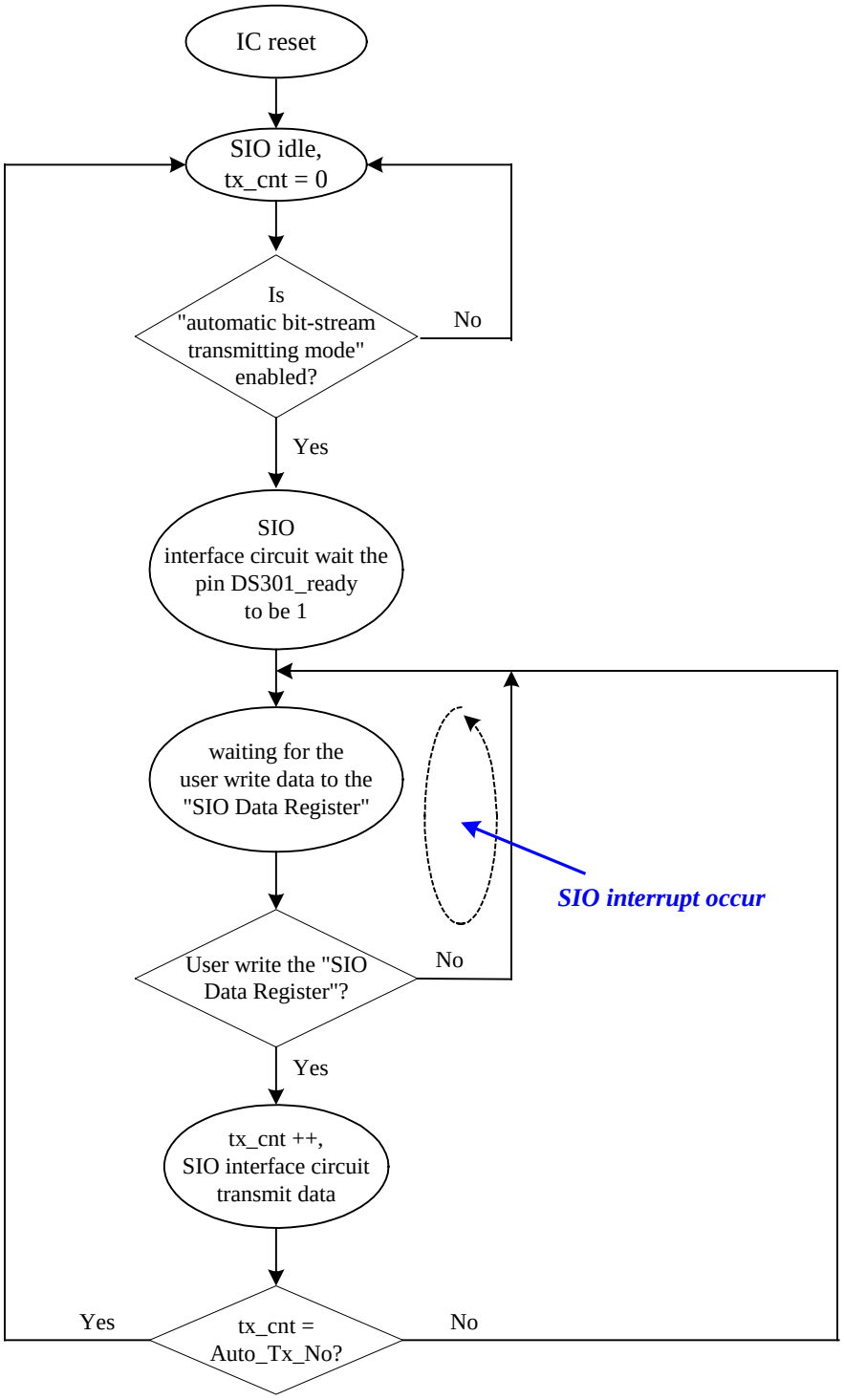


Fig 22-3 SIO Application 3

23 Universal Asynchronous Receiver/Transmitter - UART

23.1 Features

The UART of SPCE3200 is an enhanced serial communication unit for more powerful and flexible data exchange between MCUs. Major features of UART include:

- Maximum baudrate: 460.8Kbps
- Programmable transmitting wait time and receiving latency
- Embedded 2-byte transmit FIFO and 8-byte receive FIFO
- Programmable receive FIFO size: 0~8 bytes
- Independent masking of transmit FIFO, receive FIFO, and receiver timeout interrupts
- False start bit detection
- Link break generation and detection
- Standard MODEM control signals: CTS, RTS, DSR, DTR, and DCD
- Fully-programmable serial interface:
  - Data can be 5, 6, 7 or 8 bits
  - Even, odd or no-parity bit generation and detection
  - 1 or 2 stop bits generation

23.2 Block Diagram

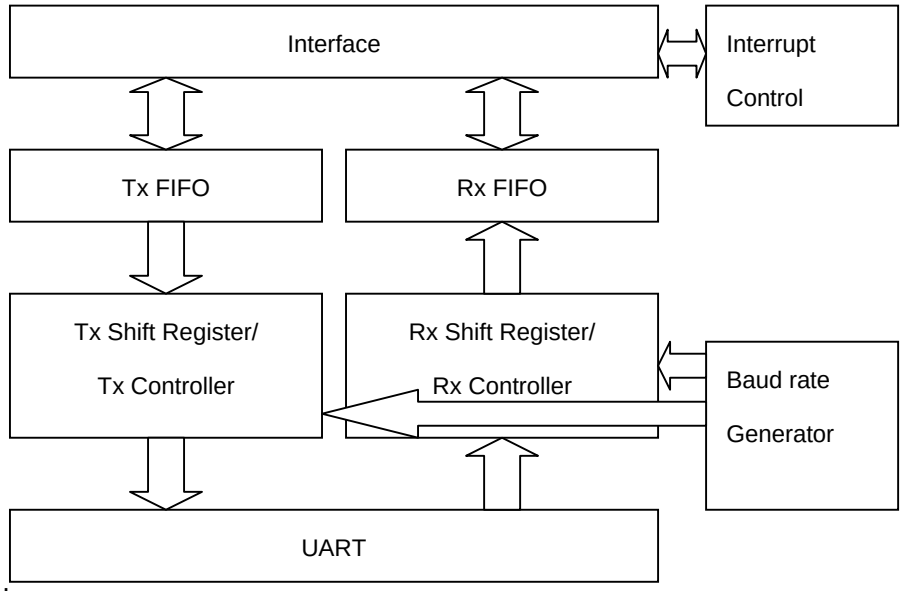


Fig 23-1 UART Block Diagram

23.3 Control Registers

23.3.1 P_UART_TXRX_DATA (R/W)(0x88150000): UART Data Tx/Rx

NAME	B7	B6	B5	B4	B3	B2	B1	B0
P_UART_TXRX_DATA	UART_Data							

UART_Data: UART data Read/Write Register

23.3.2 P_UART_ERR_STATUS (R/W)(0x88150004): UART Tx & Rx Error Flag

Name	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_UART_ERR_STATUS	RxD												OE	BE	PE	FE

RxD: UART RxD Signal

OE: Overrun Error

This bit is set to 1 if data is received and the FIFO is already full.

Read 0: Doesn't occur

Read 1: Happens

Write 1: Clear this error flag

BE: Break Error

This bit is set to 1 if a break condition is detected, indicating the received data is held LOW for a period longer than a full-word transmission time (defined as start, data, parity, and stop bits).

Read 0: Doesn't occur

Read 1: Happens

Write 1: Clear this error flag

PE: Parity Error

This bit is set to 1 if the parity of the received data character does not match with the parity selected in PSEL control bit. This bit is refreshed in every read. So, it is necessary to check this bit after DATA register is read.

Read 0: Doesn't occur

Read 1: Happens

Write 1: Clear the error flag

FE: Frame Error

This bit is set to 1 if the received character does not have a valid stop bit (a valid stop bit is 1). This bit is refreshed in every read. So, it is necessary to check this bit after DATA register is read.

Read 0: Doesn't occur

Read 1: Happens

Write 1: Clear this error flag

23.3.3 P_UART_MODE_CTRL (R/W)(0x88150008): UART Control

Name	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_UART_MODE_CTRL	RIEN	TIEN	RT	UEN						WLSEL		FEN	SBSEL	PSEL	PEN	SB

RIEN: Receive Interrupt Enable

0: Disable

1: Enable

TIEN: Transmit Interrupt Enable

0: Disable

1: Enable

RT: Receive Timeout Interrupt Enable

0: Disable

1: Enable

UEN: UART Enable

0: Disable

1: Enable

WLSEL: Word Length Definition

Indicate number of data bits transmitted or received in a frame

00: 5 bits

01: 6 bits

10; 7 bits

11: 8 bits

FEN: FIFO Buffer Enable/Disable

Enable FIFO buffer during transmission and receiving. When this bit is cleared to “0”, the FIFO becomes 1-byte hold register.

0: Disable

1: Enable

SBSEL: Stop Bit Size Selection

When this bit is set to “1”, two stop bits are transmitted at the end of the frame; but the receive logic won’t check these two stop bits.

0: 1 stop bit

1: 2 stop bits

PSEL: Parity Selection

If this bit is set to “1”, even parity generation and checking is performed during

transmission and receiving, which checks an even number of 1's in data and parity bits. When this bit is cleared to "0", the odd parity is performed, which checks an odd number of 1s. This bit has no effect when PEN Control bit is cleared to "0".

0: Odd Parity (if PEN= 1)

1: Even Parity (if PEN= 1)

PEN: Parity Enable

If this bit is set to "1", parity checking and generation is enabled; or else parity is disabled and no parity bit is added to the data frame.

0: Disable

1: Enable

SB: Send Break

If this bit is set to "1", a low level is continually output on the TX output pin after the current character is transmitted. This bit must be asserted within at least one complete frame transmission time in order to generate a break condition. The contents in transmit FIFO remain unaffected during a break condition. For normal use, this bit must be cleared to "0".

0: Normal Operation

1: Send Break Signal

23.3.4 P_UART_BAUDRATE_SETUP (R/W)(0x8815000C): UART Baud Rate Setup

Name	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_UART_BAUDRATE_SETUP	Baud rate															

To get a desired Baud Rate, write the corresponding value to the Port $8815000CH according to the following Baud Rate equation:

Baud Rate = Fosc/Baud Rate register -1

23.3.5 P_UART_TXRX_STATUS (R/W)(0x88150010): UART Status

Name	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_UART_TXRX_STATUS	RI	TI	RT	MIT		URI	RTS	DTR	TE	RF	TF	RE	BY	DCD	DSR	CTS

RI: Receive Interrupt Flag

Read 0: Doesn't occur

Read 1: Happens

TI: Transmit Interrupt Flag

Read 0: Doesn't occur

Read 1: Happens

RT: Receive Timeout Interrupt Flag

Read 0: Doesn't occur

Read 1: Happens

MIT: Modem Status Interrupt Flag

Read 0: Doesn't occur

Read 1: Happens

URI: Complement of the uUARTRI modem status input

0: uUARTRI is 1

1: uUARTRI is 0

RTS: Modem output

DTR: Modem output'direction

TE: Transmit FIFO Empty Flag

The meaning of this bit depends on the state of the FEN control bit.

If the FIFO is disabled, this bit is set to "1" when the transmit hold register is empty. If the FIFO is enabled, this bit is set to "1" when the transmit FIFO is empty.

0: Not empty

1: Empty

RF: Receive FIFO Full Flag

The meaning of this bit depends on the state of the FEN control bit.

If the FIFO is disabled, this bit is set to "1" when the receive hold register is full. If the FIFO is enabled, this bit is set to "1" when the receive FIFO is full.

0: Not full

1: Full

TF: Transmit FIFO Full Flag

The meaning of this bit depends on the state of the FEN control bit.

If the FIFO is disabled, this bit is set to "1" when the transmit hold register is full. If the FIFO is enabled, this bit is set to "1" when the transmit FIFO is full.

0: Not full

1: Full

RE: Receive FIFO Empty Flag

The meaning of this bit depends on the state of the FEN control bit.

If the FIFO is disabled, this bit is set to "1" when the receive hold register is empty. If the FIFO is enabled, this bit is set to "1" when the receive FIFO is empty.

0: Not empty

1: Empty

BY: BUSY

If this bit is set to “1”, the UART module is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, is sent from the shift register.

0: Not busy

1: Busy

DCD: Complement of the uUARTDCD modem status input

0: nUARTDCD is 1

1: nUARTDCD is 0

DSR: Complement of the uUARTDSR modem status input

0: nUARTDSR is 1

1: nUARTDSR is 0

CTS: Complement of the nUARTCTS modem status input

0: nUARTCTS is 1

1: nUARTCTS is 0

24 Universal Serial Bus – USB 1.1

24.1 Introduction

The USB controller built in SPCE3200 has two functions: to be a USB device or to be a USB mini-host. To serve as a USB device, it has 4 endpoints: each of which can be set into control pipe, bulk in, bulk out, and interrupt in. To serve as a USB mini host, on the other hand, basic transaction and function are supported. There is only one FIFO implemented by 128x8-bit single port SRAM.

To speed up the transaction when bulk in/bulk out is enabled, DMA function is supported and the FIFO is regarded as dual FIFO (each one is 64 bytes). In addition, it's permitted that bulk in can bulk out can happen simultaneously when DMA is disabled.

24.2 Features

- USB 1.1 Version
- Supports USB device
- Supports USB mini-host
- Built-in USB transceiver
- Supports 4 endpoints when USB device is enabled
  - Control pipe for standard command
  - Bulk IN for a large number of data transfer
  - Bulk OUT for a large number of data transfer
  - Interrupt in for data transfer (seldom happen)
- An 8-byte DFF FIFO for control pipe (for USB device only)
- For USB device, a 128x8-bit single port SRAM is provided for Bulk IN and Bulk OUT; for USB host, all transmission are using this single port SRAM
- An 8-byte DFF FIFO for Interrupt IN (for USB device only)
- Functions supported when USB mini-host is enabled:
  - Setup command or data transaction
  - IN/OUT transaction
  - Programmable packet delay time or timeout latency
  - SOF timer/frame number generator
  - Reset signal
- Interrupt mode or polling mode for driver

24.3 USB Device

When SPCE3200 is used as a USB Device, it supports 4 endpoints, control pipe, bulk in,

bulk out, and interrupt in. For control pipe, as soon as SPCE3200 receives a standard command, it will automatically reply to it except Get/Set Descriptor; that is, when it receives GET_STATUS, CLEAR_FEATURE, SET_FEATURE, SET_ADDRESS, GET_CONFIGURATION, SET_CONFIGURATION, GET_INTERFACE, and SET_INTERFACE, the SPCE3200 will automatically reply to these standard commands. For bulk in and bulk out, the maximum packet size is 64 bytes. SPCE3200 supports both non-DMA and DMA transmission/receiving. For non-DMA mode, it is 8 bits for MCU access; for DMA mode, it is 16 bits.

24.4 USB Mini Host

When SPCE3200 is used as a USB mini host, it supports commands, IN, and OUT transfer, all with the maximum packet size to be 64 bytes. It is 8 bits for MCU access or 16 bits for DMA access. When it uses DMA mode, the data to be transferred must be multi-pipe of 64 bytes.

24.5 USB Device Control Registers

24.5.1 P_USBD_MODE_CTRL1 (0x881B00C0): USB Config

NAME	D11-D8	D7-D6	D5	D4	D3	D2-D1	D0
P_USBD_MODE_CTRL1	TNSPH	TNSPL	RWIPEN	SPWR	USBEN	-	BYPASS

TNSPH: USB Transceiver Pull High

TNSPL: USB Transceiver Pull Low

RWIPEN: Remote Wakeup Enable

0: Disable

1: Enable

SPWR: Self Power of Device

0: USB device is bus-powered

1: USB device is self-powered

USBEN: USB Transceiver Enable

0: Disable

1: Enable

BYPASS: USB Bypass Mode

0: Bypass is disabled

1: Bypass is enabled, but the inner USB transceiver is disabled.

24.5.2 P_USBD_MODE_CTRL3 (0x881B015C): USB Device

NAME	D15-D14	D13-D12	D11-D10	D9-D8	D7	D6	D5	D4	D3-D1	D0
P_USBD_MODE_CTRL3	EP4_Type	EP3_Type	EP2_Type	EP1_Type	EP4_IO	EP3_IO	EP2_IO	EP1_IO	-	MODE

EP4_Type: Endpoint4 Type

00: Reserved

01: Reserved

10: Bulk

11: Interrupt

EP3_Type: Endpoint3 Type

00: Reserved

01: Reserved

10: Bulk

11: Interrupt

EP2_Type: Endpoint2 Type

00: Reserved

01: Reserved

10: Bulk

11: Interrupt

EP1_Type: Endpoint1 Type

00: Reserved

01: Reserved

10: Bulk

11: Interrupt

EP4_IO: Endpoint4 IN/OUT

0: OUT

1: In

EP3_IO: Endpoint3 IN/OUT

0: OUT

1: In

EP2_IO: Endpoint2 IN/OUT

0: OUT

1: In

EP1_IO: Endpoint1 IN/OUT

0: OUT

1: In

MODE: Mode Selection

0: Normal mode

1: Debug Mode

24.5.3 P_USBD_MODE_CTRL2 (0x881B00C4): USB Function

NAME	D11	D10	D9	D8-D7	D6-D0
P_USBD_MODE_CTRL2	SRST	DMA_BOEN	DMA_BIEN	ConfigVal	FNC_Addr

SRST: Software Reset

DMA_BOEN: DMA Bulk Out Enable

0: Disable DMA function with bulk out

1: Enable DMA function with bulk out (Bulk In must be disabled)

DMA_BIEN: DMA Bulk In Enable

0: Disable DMA function with bulk in

1: Enable DMA function with bulk in (Bulk out must be disabled)

ConfigVal: Configure Value

The USB configuration value of the device can be read from these two bits when receiving set configuration command.

FNC_Addr: Function Address

When the device gets the “Set Address” command from the host, the address is stored in these bits.

24.5.4 P_USBD_DMAINT_STATUS (0x881B0164): USB DMA Interrupt

NAME	D1	D0
P_USBD_DMAINT_STATUS	DMAINTEN	DMAINT

DMAINTEN: DMA Interrupt Enable

0: Disable, this interrupt will be masked.

1: Enable, hardware will issue an IRQ to CPU.

DMAINT: Bulk Out Interrupt

Read 0: Doesn’t occur

Read 1: Occurs

24.5.5 P_USBD_POWER_CTRL (0x881B00C8): USB Power Management

NAME	D4	D3	D2	D1	D0
P_USBD_POWER_CTRL	RESWKE	RE_WA	RE_WAFEA	RST	SUS_MOD

RESWKE: Resume Wakeup Enable

0: Disable

1: Enable

RE_WA: Remote Wakeup

0: Disable

1: Enable

RE_WAFEA: Remote Wakeup Feature

0: Disable

1: Enable

RST: Reset, assigned to USBBUS reset

0: USB reset is low

1: USB reset is high

SUS_MOD: Suspend Mode

This bit is set by hardware when it enters suspend mode.

This bit is cleared by hardware when it returns from suspend mode or the USB reset signal is generated.

24.5.6 P_USBD_EP0_DATA (0x881B00CC): USB Endpoint0 Data

NAME	D7-D0
P_USBD_EP0_DATA	EP0DATA

EP0DATA: Endpoint0_Data

24.5.7 P_USBD_BULKIN_DATA (0x881B00D0): USB Bulk IN Data

NAME	D7-D0
P_USBD_BULKIN_DATA	BIDATA

BIDATA: Bulk In Data

24.5.8 P_USBD_BULKOUT_DATA(0x881B00D4): USB Bulk Out Data

NAME	D7-D0
P_USBD_BULKOUT_DATA	BODATA

BODATA: Bulk Out Data

24.5.9 P_USBD_INTIN_DATA (0x881B00D8): USB Interrupt In Data

NAME	D7-D0
P_USBD_INTIN_DATA	INTINDATA

INTINDATA: Interrupt IN Data

24.5.10 P_USBD_NULL_PACKET (0x881B0160): USB Null Packet

NAME	D0
P_USBD_NULL_PACKET	NULLPKT

NULLPKT: Send Null packet

0: Disable the function

1: Enable the function

24.5.11 P_USBD_EP_EVENT (0x881B00DC): USB Endpoint Event

NAME	D15	D14	D13	D12	D11	D10	D9	D8
P_USBD_EP_EVENT	IINNA	IINPR	BONA	BOPR	BOPE	BINA	BIPC	BIPR
NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_USBD_EP_EVENT	E0SNA	E0SEN	E0INNA	E0INPR	E0ONA	E0OPR	E0OPE	E0SFPR

IINNA: Interrupt In NACK

Read 1: Occurs

Read 0: Doesn't occur

Write 0: No Effect

Write 1: Clear the flag

IINPR: Interrupt In Packet Ready

Write 1: IN Packet is ready in the INTERRUPT_IN FIFO.

BONA: Bulk Out NACK

Read 1: An OUT packet happens, but the device sends a NAK signal

Read 0: Doesn't occur
Write 0: No Effect
Write 1: Clear the flag

BOPR: Bulk Out Packet Ready

Read 1: Ready
Read 0: Not Ready
Write 1: Clear the flag

BOPE: Bulk Out Packen Enable

Write 1: Enable

BINA: Bulk IN NACK

Read 1: An IN request happens, but the device sends a NAK signal
Read 0: Doesn't occur
Write 1: Clear the flag

BIPC: Bulk IN Packet Clear

Read 1: An IN packet is read from the host
Read 0: Doesn't occur
Write 1: Clear the flag

BIPR: Bulk IN Packet Ready

Write 1: Packet is ready

E0SNA: EP0 Status NACK

Read 1: The request of status transaction happens, but the device send a NAK signal.
Read 0: Doesn't occur
Write 1: Clear the flag

E0SEN: EP0 Status Enable

0: Disable
1: Enable

E0INNA: EP0 IN NACK

Read 1: An IN packet happens, but the device sends a NAK signal
Read 0: Doesn't occur
Write 1: Clear the flag

E0INPR: EP0 IN Packet Ready

Write 1 to indicate IN packet is ready in Endpoint0 FIFO

E0ONA: EP0 Out NACK

Read 1: An OUT packet happens, but the device sends a NAK signal

Read 0: Doesn't occur

Write 1: Clear the flag

E0OPR: EP0 Out Packet Ready

Read 1: Ready

Write 1: Clear the flag

E0OPE: EP0 Out Packet Enable

Write 1: Enable the incoming packet for OUT data.

E0SPR: EP0 Setup Packet Ready

Read 1: A non-standard setup command or get/set descriptor command is loaded into the endpoint0 FIFO.

Write 1: Clear the flag

24.5.12 P_USBD_INT_STATUS1 (0x881B00E0): USB Global Interrupt

NAME	D6	D5	D4	D3	D2	D1	D0
P_USBD_INT_STATUS1	DMA	STANDARD	POWER	INT	BO	BI	EP0

DMA: DMA Interrupt

Read 1: Occurs

Write 1: Clear the flag

STANDARD: Standard Command Interrupt

Read 1: Occurs

Write 1: Clear the flag

Power: Power Management Interrupt

Read 1: Occurs

Write 1: Clear the flag

INT: Interrupt In Interrupt

Read 1: One of INTERRUPT_IN interrupt happens

Write 1: Clear the flag

BO: Bulk Out Interrupt

Read 1: Occurs

Write 1: Clear the flag

BI: Bulk In Interrupt

Read 1: Occurs

Write 1: Clear the flag

EP0: Endpoint0 Interrupt

Read 1: Occurs

Write 1: Clear the flag

24.5.13 P_USBD_INT_CTRL (0x881B00E4): USB Interrupt Enable

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_USBD_INT_CTRL	BIPC	E0SNA	E0SC	E0INNA	E0INPC	E0ONA	E0OPS	E0SPS
NAME	D15	D14	D13	D12	D11	D10	D9	D8
P_USBD_INT_CTRL	RST	RME	SUS	IINNA	IINPC	BONA	BOPS	BINA

RST: Reset Interrupt Enable

0: Disable

1: Enable

RME: Resume Interrupt Enable

0: Disable

1: Enable

SUS: Suspend Interrupt Enable

0: Disable

1: Enable

IINNA: Interrupt In NACK Interrupt Enable

0: Disable

1: Enable

IINPC: Interrupt In Packet Clear Interrupt Enable

0: Disable

1: Enable

BONA: Bulk Out NACK Interrupt Enable

0: Disable

1: Enable

BOPS: Bulk Out Packet Set Interrupt Enable

0: Disable

1: Enable

BINA: Bulk In NACK Interrupt Enable

0: Disable

1: Enable

BIPC: Bulk In Packet Clear Interrupt Enable

0: Disable

1: Enable

E0SNA: EP0 Status NACK Interrupt Enable

0: Disable

1: Enable

E0SC: EP0 Status Clear Interrupt Enable

0: Disable

1: Enable

E0INNA: EP0 In NACK Interrupt Enable

0: Disable

1: Enable

E0INPC: EP0 In Packet Clear Interrupt Enable

0: Disable

1: Enable

E0ONA: EP0 Out NACK Interrupt Enable

0: Disable

1: Enable

E0OPS: EP0 Out Packet Set Interrupt Enable

0: Disable

1: Enable

E0SPS: EP0 Setup Packet Set Interrupt Enable

0: Disable

1: Enable

24.5.14 P_USBD_INT_STATUS2 (0x881B00E8): USB Interrupt

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_USBD_INT_STATUS2	BIPC	E0SNA	E0SC	E0INNA	E0INPC	E0ONA	E0OPS	E0SPS
NAME	D15	D14	D13	D12	D11	D10	D9	D80
P_USBD_INT_STATUS2	RST	RME	SUS	IINNA	IINPC	BONA	BOPS	BINA

RST: Reset Interrupt Flag

Read 0: Doesn't occur

Read 1: Occurs

Write 1: Clear the flag

RME: Resume Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

SUS: Suspend Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

IINNA: Interrupt In NACK Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

IINPC: Interrupt In Packet Clear Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

BONA: Bulk Out NACK Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

BOPS: Bulk Out Packet Set Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

BINA: Bulk In NACK Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

BIPC: Bulk In Packet Clear Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0SNA: EP0 Status NACK Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0SC: EP0 Status Clear Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0INNA: EP0 In NACK Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0INPC: EP0 In Packet Clear Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0ONA: EP0 Out NACK Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0OPS: EP0 Out Packet Set Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

E0SPS: EP0 Setup Packet Set Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

24.5.15 P_USBD_COMMANDINT_CTRL (0x881B00EC): USB Standard Command Interrupt Enable

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_USBD_COMMANDINT_CTRL	GSTS	CFEA	SFEA	SADD	GCON	SCON	GINT	SINT

GSTS: Get Status Interrupt Enable

0: Disable
1: Enable

CFEA: Clear Feature Interrupt Enable

0: Disable
1: Enable

SFEA: Set Feature Interrupt Enable

0: Disable

1: Enable

SADD: Set Address Interrupt Enable

0: Disable

1: Enable

GCON: Get Configuration Interrupt Enable

0: Disable

1: Enable

SCON: Set Configuration Interrupt Enable

0: Disable

1: Enable

GINT: Get Interface Interrupt Enable

0: Disable

1: Enable

SINT: Set Interface Interrupt Enable

0: Disable

1: Enable

24.5.16 P_USBD_COMMANDINT_STATUS (0x881B00F0): USB Standard Command Interrupt

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_USBD_COMMANDINT_STATUS	GSTS	CFEA	SFEA	SADD	GCON	SCON	GINT	SINT

GSTS: Get Status Interrupt Flag

Read 0: Doesn't occur

Read 1: Occurs

Write 1: Clear the flag

CFEA: Clear Feature Interrupt Flag

Read 0: Doesn't occur

Read 1: Occur

Write 1: Clear the flag

SFEA: Set Feature Interrupt Flag

Read 0: Doesn't occur

Read 1: Occurs

Write 1: Clear the flag

SADD: Set Address Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

GCON: Get Configuration Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

SCON: Set Configuration Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

GINT: Get Interface Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

SINT: Set Interface Interrupt Flag

Read 0: Doesn't occur
Read 1: Occurs
Write 1: Clear the flag

24.5.17 P_USBD_EP_AUTOSET (0x881B00F4): USB Endpoint Auto Set

NAME	D5	D4	D3	D2	D1	D0
P_USBD_EP_AUTOSET	EP0ASE	IAINPR	BAOPE	BAIPR	E0AIPR	E0AOPE

EP0ASE: EP0 Auto Status Enable Set

Please refer to P_USBD_ EP_EVENT [6]

IAINPR: Interrupt Auto In Packet Ready Set

Please refer to P_USBD_ EP_EVENT [14]

BAOPE: Bulk Auto Out Packet Enable Set

Please refer to P_USBD_ EP_EVENT [11] or [12]

BAIPR: Bulk Auto In Packet Ready Set

Please refer to P_USBD_ EP_EVENT [8]

E0AIPR: EP0 Auto In Packet Ready Set

Please refer to P_USBD_ EP_EVENT [4]

E0AOPE: EP0 Auto Out Packet Enable Set

Please refer to P_USBD_ EP_EVENT [1] or [2]

24.5.18 P_USBD_EP_SETSTALL (0x881B00F8): USB Endpoint Set Stall

NAME	D3	D2	D1	D0
P_USBD_EP_SETSTALL	IINS	BOS	BIS	EP0S

IINS: Interrupt In Set Stall

1: Enable STALL response

0: Disable STALL response

BOS: Bulk Out Set Stall

1: Enable STALL response

0: Disable STALL response

BIS: Bulk In Set Stall

1: Enable STALL response

0: Disable STALL response

EP0S: EP0 Set Stall

1: Enable STALL response

0: Disable STALL response

24.5.19 P_USBD_EP_BUFFERCLR (0x881B00FC): USB Endpoint Buffer Clear

NAME	D3	D2	D1	D0
P_USBD_EP_BUFFERCLR	IINBC	BOBC	BIBC	EP0BC

IINBC: Interrupt In Buffer Clear

Write “1” to clear Interrupt In buffer

BOBC: Bulk Out Buffer Clear

Write “1” to clear Bulk Out buffer

BIBC: Bulk In Buffer Clear

Write “1” to clear Bulk In buffer

EP0BC: EP0 Buffer Clear

Write “1” to clear EP0 buffer

24.5.20 P_USBD_EP_EVENTCLR (0x881B0100): USB Endpoint Event Clear

NAME	D5	D4	D3	D2	D1	D0
P_USBD_EP_EVENTCLR	IINPC	BOEC	BIPC	EP0SC	EP0IPC	EP0OEC

IINPC: Interrupt In Packet Clear

Write “1” to this bit to clear P_USBD_EP_EVENT [14]

BOEC: Bulk Out Enable Clear

Write “1” to this bit to clear P_USBD_EP_EVENT [11]

BIPC: Bulk In Packet Clear

Write “1” to this bit to clear P_USBD_EP_EVENT [9]

EP0SC: EP0 Status Clear

Write “1” to this bit to clear P_USBD_EP_EVENT [6]

EP0IPC: EP0 In Packet Clear

Write “1” to this bit to clear P_USBD_EP_EVENT [4]

EP0OEC: EP0 Out Enable Clear

Write “1” to this bit to clear P_USBD_EP_EVENT [1]

24.5.21 P_USBD_EP0_WRITECOUNT (0x881B0104): USB Endpoint0 Write Count

NAME	D3-D0
P_USBD_EP0_WRITECOUNT	EP0WC

EP0WC: EP0 Write Count

Read these bits to indicate the number of data in the EP0 FIFO.

24.5.22 P_USBD_BULKOUT_WRITECOUNT (0x881B0108): USB Bulk Out Write Count

NAME	D6-D0
P_USBD_BULKOUT_WRITECOUNT	BOWC

BOWC: Bulk Out Write Count

Read these bits to indicate the number of data in the Bulk Out FIFO.

24.5.23 P_USBD_EP0_BUFFERSEL (0x881B010C): USB Endpoint0 Buffer Pointer

NAME	D5-D3	D2-D0
P_USBD_EP0_BUFFERSEL	EP0WBP	EP0RBP

EP0WBP: EP0 Write Buffer Pointer

The write pointer of EP0 FIFO

EP0RBP: EP0 Read Buffer Pointer

The read pointer of EP0 FIFO

24.5.24 P_USBD_BULKIN_BUFFERSEL (0x881B0110): USB Bulk In Buffer Pointer

NAME	D15-D8	D7-D0
P_USBD_BULKIN_BUFFERSEL	BIBRP	BIBWP

BIBRP: Bulk IN Write Buffer Pointer

The write pointer of Bulk IN FIFO

BIBWP: Bulk IN Read Buffer Pointer

The read pointer of Bulk IN FIFO

24.5.25 P_USBD_BULKOUT_BUFFERSEL (0x881B0114): USB Bulk Out Buffer Pointer

NAME	D15-D8	D7-D0
P_USBD_BULKOUT_BUFFERSEL	BOBRP	BOBWP

BOBRP: Bulk OUT Buffer Write Pointer

The write pointer of Bulk OUT FIFO

BOBWP: Bulk OUT Buffer Read Pointer

The read pointer of Bulk OUT FIFO

24.5.26 P_USBD_EP0_BMREQUESTTYPE (0x881B0118): USB Endpoint0 bmRequestType

NAME	D7-D0
P_USBD_EP0_BMREQUESTTYPE	EP0RTR

EP0RTR: EP0 bmRequestType

The bmRequestType of setup command

24.5.27 P_USBD_EP0_BREQUEST (0x881B011C): USB Endpoint0 bRequest

NAME	D7-D0
P_USBD_EP0_BREQUEST	EP0RR

EP0RR: EP0 bRequest

The bRequest of setup command

24.5.28 P_USBD_EP0_WVALUE (0x881B0120): USB Endpoint0 wValue

NAME	D15-D0
P_USBD_EP0_WVALUE	EP0VR

EP0VR: EP0 wValue

The wValue of setup command

24.5.29 P_USBD_EP0_WINDEX (0x881B0124): USB Endpoint0 wIndex

NAME	D15-D0
P_USBD_EP0_WINDEX	EP0IR

EP0IR: EP0 wIndex

The wIndex of setup command

24.5.30 P_USBD_EP0_WLENGTH (0x881B0128): USB Endpoint0 wLength

NAME	D15-D0
P_USBD_EP0_WLENGTH	EP0LR

EP0LR: EP0 wLength

The wLength of setup command

24.5.31 P_USBD_DMA_WRITECOUNT (0x881B0140): USB DMA Byte Count Register

NAME	D23-D0
P_USBD_DMA_WRITECOUNT	DMAWC

DMAWC: DMA Write Count

The register is used in DMA mode. Set this register to indicate how many word data are transferred.

24.5.32 P_USBD_DMA_ACKCOUNT (0x881B0144): USB DMAACK Count

NAME	D18-D0
P_USBD_DMA_ACKCOUNT	DMAACK

DMAACK: DMA ACK Count

Write this port to reset DMAWC and DMAACK

24.5.33 P_USBD_EP_STALL (0x881B0150): USB Endpoint Stall Bit

NAME	D11	D10	D9	D8	D7-D4	D3	D2	D1	D0
P_USBD_EP_STALL	IISS	BOSS	BISS	EP0SS		IISB	BOSB	BISB	EP0SB

IISS: Interrupt IN

It indicates USB device has replied to the request by sending STALL

Write 1: Clear the flag

BOSS: Bulk OUT

It indicates USB device has replied to the request by sending STALL

Write 1: Clear the flag

BISS: Bulk IN

It indicates USB device has replied to the request by sending STALL

Write 1: Clear the flag

EP0SS: Endpoint0

It indicates USB device has replied to the request by sending STALL

Write 1: Clear the flag

IISB: Interrupt IN

If STALL happens, the bit is set. The bit is for debugging only.

BOSB: Bulk OUT

If STALL happens, the bit is set. The bit is for debugging only.

BISB: Bulk IN

If STALL happened, the bit is set. The bit is for debugging only.

EP0SB: Endpoint0

If STALL happens, the bit is set. The bit is for debugging only.

24.6 USB Host Control Registers

24.6.1 P_USBH_MODE_CTRL (0x881C0000): USB Host Configuration

NAME	D5	D4	D3	D2	D1	D0
P_USBH_MODE_CTRL	SUS		ASOF	SOFTR		HOSTEN

SUS: Host Suspend

0: HOST SUSPEND is disabled.

1: HOST SUSPEND is enabled. USB Transceiver is in SUSPEND mode. (The SUSPEND mode is controlled by software when HOST is

enabled.)

ASOF: Auto Generate SOF

0: Generate SOF by software

1: Generate SOF by hardware

SOFTR: SOF Timer

0: Disable SOF timer

1: Enable SOF timer

HOSTEN: Host Enable

0: Host is disabled

1: Host is enabled (Device is disabled)

This bit must be set to 1 when accessing the USB Host functions.

24.6.2 P_USBH_TIMING_SETUP (0x881C0004): USB Host Timing Config

NAME	D5	D4	D3-D2	D1-D0
P_USBH_TIMING_SETUP	SAU	PAC	TC	IPD

SAU: Storage 1/2 Auto Mode

This bit should always be 1 in order not to set up P_USBH_StorageRST after each transaction in non-DMA mode.

PAC: Pointer Auto Clear

This bit should always be 1, since READ/WRITE pointer will be automatically reset to 0 after any transaction. However, this mode is automatically disabled in DMA mode.

The DMA mode is entered by configuring P_USBH_AUTO_TRANSFER [0] or P_USBH_AutoTrans[1].

TC: TimeOut Criteria

00: 16T

01: 18T

10: 20T

11: 22T

IPD: Inter Packet Delay

00: 2T

01: 4T

10: 6T

11: 8T

24.6.3 P_USBH_TXRX_DATA (0x881C0008): USB Host Data

NAME	D7-D0
P_USBH_TXRX_DATA	HDATA

HDATA: Host Data

24.6.4 P_USBH_TRANSFER_MODE (0x881C000C): USB Host Transfer

NAME	D6	D5	D4	D3	D2	D1	D0
P_USBH_TRANSFER_MODE	RST	OD1	OD0	ID1	ID0	Setup	SOF

RST: Reset Signal

1: Generate reset signal

0: Stop reset signal

OD1: Out Data1 Transfer

Write 1 to generate OUT DATA1 transfer. This bit is cleared automatically if the transfer is completed.

OD0: Out Data0 Transfer

Write 1 to generate OUT DATA0 transfer. This bit is cleared automatically if the transfer is completed.

ID1: In Data1 Transfer

Write 1 to generate IN DATA1 transfer. This bit is cleared automatically if the transfer is completed.

ID0: In Data0 Transfer

Write 1 to generate IN DATA0 transfer. This bit is cleared automatically if the transfer is completed.

Setup: Setup Transfer

Write 1 to generate SETUP transfer. This bit is cleared automatically if the transfer is completed.

SOF: SOF Transfer

Write 1 to generate SOF transfer (SOF is generated by software). This bit is cleared automatically if the transfer is completed.

24.6.5 P_USBH_DEVICE_ADDR (0x881C0010): USB Device Address

NAME	D6-D0
P_USBH_DEVICE_ADDR	DAddr

DAddr: Device Address

Write the device address to these bits that host want to access.

24.6.6 P_USBH_DEVICE_EP (0x881C0014): USB Device Endpoint

NAME	D3-D0
P_USBH_DEVICE_EP	DEP

DEP: Device Endpoint

Write the device endpoint to these bits that host want to access.

24.6.7 P_USBH_TX_COUNT (0x881C0018): USB Host Transmit Count

NAME	D6-D0
P_USBH_TX_COUNT	TXC

TXC: Host Transmitting Count

Write the number of data to these bits that host wants to transmit.

24.6.8 P_USBH_RX_COUNT (0x881C001C): USB Receive Count

NAME	D6-D0
P_USBH_RX_COUNT	RXC

RXC: Host Receiving Count

The number of data received by host is stored in these bits.

24.6.9 P_USBH_FIFOIN_POINTER (0x881C0020): USB Host FIFO Input Pointer

NAME	D7-D0
P_USBH_FIFOIN_POINTER	HFIP

HFIP: Host FIFO Input Pointer

24.6.10 P_USBH_FIFOOUT_POINTER (0x881C0024): USB Host FIFO Output Pointer

NAME	D7-D0
P_USBH_FIFOOUT_POINTER	HFOP

HFOP: Host FIFO Output Pointer

24.6.11 P_USBH_AUTOIN_COUNT (0x881C0028): USB Host Automatic In Transaction Byte Count

NAME	D15-D0
P_USBH_AUTOIN_COUNT	HAIBC

HAIBC: Host Automatic IN Transaction Byte Count

Write the number of IN transaction that has to be generated in these bits. For example, if the host is going to receive 512 bytes from the device, it has to generate 512/64=8 IN transaction, and write 8 to this register. This register is used only in DMA mode.

24.6.12 P_USBH_AUTOOUT_COUNT (0x881C002C): USB Host Automatic Out Transaction Byte Count

NAME	D15-D0
P_USBH_AUTOOUT_COUNT	HAOBC

HAOBC: Host Automatic OUT Transaction Byte Count

Write the number of OUT transaction in these bits that has to be generated. For instance, if the host is going to transmit 512 bytes to the device, it has to generate 512/64=8 OUT transaction, and write 8 to this register. This register is used only in DMA mode.

24.6.13 P_USBH_AUTO_TRANSFER (0x881C0030): USB Host Auto Transfer

NAME	D3	D2	D1	D0
P_USBH_AUTO_TRANSFER	CAO	CAI	AOX	AIX

CAO: Clear Auto Out Transfer

Write “1” to this bit to clear P_USBH_AutoTrans[1]

CAI: Clear Auto IN Transfer

Write “1” to this bit to clear P_USBH_AutoTrans[0]

AOX: Auto Out Transfer

This bit is set to 1 by entering DMA mode for Bulk out. When transaction is surely finished, this bit must be cleared.

Note: After DMA is finished (DMA counter reaches 0), the transaction may not be finished yet. Enabling this bit, programming the UHAOBCR, and then triggering the OUT transfer will start the OUT transaction.

This bit is cleared by software.

AIX: Auto In Transfer

This bit is set to 1 by entering DMA mode for Bulk in. When transaction is surely finished, this bit must be cleared. Enabling this bit, programming the UHAIBCR, and then triggering the IN transfer will start the IN transaction. This bit is cleared by software.

24.6.14 P_USBH_MODE_STATUS (0x881C0034): USB Host Status

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_USBH_MODE_STATUS	TO	CRC	DE	BS	UP	SH	NH	AH

TO: TimeOut, No Response

This flag is set when timeout or no response from the device

CRC: CRC16 Error Packet Received

This flag is set when In data packet is received with CRC16 error.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

DE: Data Sequence Error Packet Received

This flag is set when In data packet is received with data sequence error.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

BS: Bit Stuffing Error Packet Received

This flag is set when In data contains bit stuffing error.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

UP: Unknown PID Packet Received

This flag is set when receiving a packet with a unknown PID

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

SH: Stall Handshake Received

This flag is set when receiving a stall handshake.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

NH: NACK Handshake Received

This flag is set when receiving a NACK handshake.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

AH: ACK Handshake Received

This flag is set when receiving an ACK handshake.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

24.6.15 P_USBH_INT_STATUS (0x881C0038): USB Host Interrupt

NAME	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
P_USBH_INT_STATUS	DPO	TRST	TSOFI	ITOK	TXO	VSC	AOX	AIX	RX	TX	SOF	DSC

DPO: Device Plug Out Interrupt

This interrupt is asserted whenever host detects device is plug out. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

TRST: Transmit USB Reset Interrupt

This interrupt is asserted whenever host sends a USB RESET signal. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

TSOFI: Transmit SOF Interrupt

This interrupt is asserted whenever host sends a SOF. Write 1 to clear

the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

ITOK: IN Token Transmit Interrupt

This interrupt is asserted whenever host sends an IN Token. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

TXO: Transmit Data Interrupt

This interrupt is asserted whenever host sends a DATA packet. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

VSC: VBUS Status Change Interrupt

This interrupt is asserted whenever VBUS status is changed.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

AOX: Automatic Out Transfer Interrupt

This interrupt is asserted when the host transmits the data to the device and then receives an ACK from the device.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

AIX: Automatic In Transfer Interrupt

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

RX: Receive Interrupt

This interrupt is asserted whenever the host receives a packet from the device. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

TX: Transmit Interrupt

This interrupt is asserted whenever the TX task is completed. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

SOF: SOF Interrupt

This interrupt is periodically generated every 1ms frame time. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

DSC: DP DM Status Change Interrupt

D+/D- status change interrupt. This interrupt is used to detect the device connection when the host controller is in the idle state. Once the device is plug-in, this interrupt must be disabled. Write 1 to clear the interrupt.

Read 0: Doesn't occur

Read 1: Occurs

Write 0: No effect

Write 1: Clear the flag

24.6.16 P_USBH_INT_CTRL (0x881C003C): USB Host Interrupt Enable

NAME	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
P_USBH_INT_CTRL	DPO	TRST	TSOFI	ITOK	TXO	VSC	AOX	AIX	RX	TX	SOF	DSC

DPO: Device Plug Out Interrupt Enable

0: Disable

1: Enable

TRST: Transmit USB Reset Interrupt Enable

0: Disable

1: Enable

TSOFI: Transmit SOF Interrupt Enable

0: Disable

1: Enable

ITOK: IN Token Transmit Interrupt Enable

0: Disable

1: Enable

TXO: Transmit Data Interrupt Enable

0: Disable

1: Enable

VSC: VBUS Status Change Interrupt Enable

0: Disable

1: Enable

AOX: Automatic Out Transfer Interrupt Enable

0: Disable

1: Enable

AIX: Automatic In Transfer Interrupt Enable

0: Disable

1: Enable

RX: Receive Interrupt Enable

0: Disable

1: Enable

TX: Transmit Interrupt Enable

0: Disable

1: Enable

SOF: SOF Interrupt Enable

0: Disable

1: Enable

DSC: DP DM Status Change Interrupt Enable

0: Disable

1: Enable

24.6.17 P_USBH_SOFT_RESET (0x881C0044): USB Software Reset Register/Device Plug Out

NAME	D8	D7-D1	D0
P_USBH_SOFT_RESET	DPOE	DPOTV	SRST

DPOE: Device Plug Out Timer Enable

Write “1” to this bit to enable the timer

DPOTV: Device Plug Out Timer Value

If DPOE is 1, the inside timer4 is enabled.

For each clock cycle, if D+ and D- are all 0, the timer is added by 1. Besides, if one of D+/D- is not 0, the timer is reset to 0. When it counts to DEVICE_PLUG_OUT_ TIMER_VALUE, an interrupt is generated and DEVICE_PLUG_ TIMER_OUT_ENABLE is reset.

SRST: Software Reset

24.6.18 P_USBH_INACK_COUNT (0x881C005C): USB IN ACK Count

NAME	D15-D0
P_USBH_INACK_COUNT	INACK

INACK: IN ACK Count

24.6.19 P_USBH_OUTACK_COUNT (0x881C0060): USB Out ACK Count

NAME	D15-D0
P_USBH_OUTACK_COUNT	OUTACK

OUTACK: OUT ACK Count

24.6.20 P_USBH_RESETACK_COUNT (0x881C0064): USB Reset ACK Count

NAME	D1	D0
P_USBH_RESETACK_COUNT	IARST	OARST

IARST: IN ACK Reset

Write “1” to reset P_USBH_ INACK_COUNT

OARST: OUT ACK Reset

Write “1” to reset P_USBH_ OUTACK_COUNT

24.6.21 P_USBH_D_READBACK (0x881C006C): USB D+/ D- Readback

NAME	D1	D0
P_USBH_D_READBACK	DM	DP

DM: VPIN input signal

DP: VMIN input signal

25 Secure Digital Card – SDCard Controller

25.1 Introduction

Secure Digital (SD) Memory card is a Flash-Based memory card that is specifically designed to meet the security, capacity, performance and environment requirements inherent in newly emerging audio and video consumer electronic device. The SD Memory Card communication is based on an advanced 9-pin interface (Clock, Command, 4xData and 3xPower lines) designed to operate in a low voltage range.

SD IO card is based on and compatible with the SD memory card. The intent of the SD IO card is to provide high-speed data I/O with low power consumption for mobile electronic devices.

The SD card controller built in SPCE3200 is designed to provide high performance transfer rate using DMA access which can achieve the best performance/cost ratio.

25.2 Features

- Fully compatible with SD Memory card specification
- Receives SD command directly, improving the compatibility
- Programmable clock speed on the SD bus
- SD bus clock control while buffer is full
- Interrupt generation
- DMA R/W operation
- Supports both 1- and 4-bit SD mode
- SD IO card interrupt detection

25.3 Block Diagram

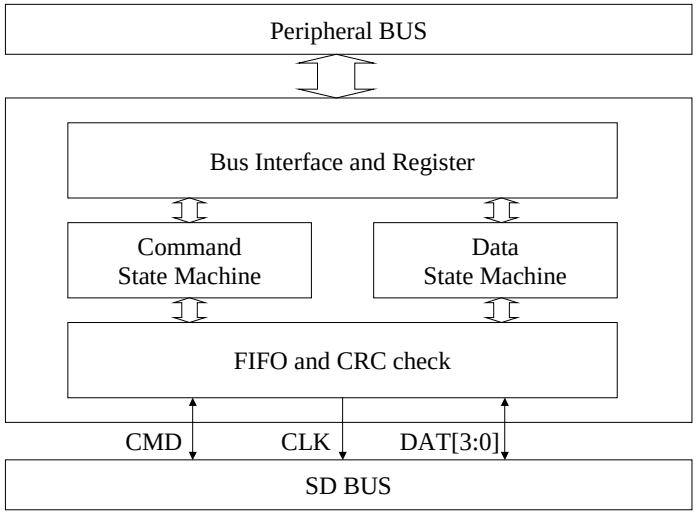


Fig 25-1 Functional Block Diagram of SD Memory Controller

25.4 Control Registers

Host uses two registers to transfer data to and from SD Card. Host needs to poll the DatBufFull/DatBuffEmpty bits in the status register to determine if the controller is able to receive/transfer data.

25.4.1 P_SD_TX_DATA (R/W)(0x88180000): SD Card Data Transmit

Host writes 32-bit data to this register and the controller will transmit it to SD card. When the data stored in the buffer is transmitted, DATBUFEMPTY bit in status register will be set or the DMA request will be issued. Note that data could be written to this register only when **DATBUFEMPTY** is 1.

Name	B31-B0
P_SD_TX_DATA	DataTx

DataTx: Write Data to SDCard. Data can be written to this register only when DATBUFEMPTY is 1 (0 by default).

25.4.2 P_SD_RX_DATA (R) (0x88180004): SD Card Data Receive

This register is used to store the data read from the SD card. When 32-bit data is received, DATBUFFULL bit in status register will be set or the DMA request will be issued. Note that data could be read from this register only when **DATBUFFULL** is 1.

Name	B31-B0
P_SD_RX_DATA	DataRx

DataRx: Read data from SD card. Reading data from this register will only valid when the DATBUFFULL is set; otherwise, it will return zeros.

25.4.3 P_SD_COMMAND_SETUP (R/W)(0x88180008): SD Card Command

Host uses this register to control the behavior of controller. Controller will use the information in this register to determine what to do.

Name	B7	B6	B5	B4	B3	B2	B1	B0
P_SD_COMMAND_SETUP	RUNCMD	STPCMD	CMDCODE					
Name	B15	B14	B13	B12	B11	B10	B9	B8
P_SD_COMMAND_SETUP		RESPTYPE			INICARD	MULBLK	TxData	CMDWD

CMDCODE: The Command code host wishes to transfer

STPCMD: Writing 1 to this bit will force the controller back to IDLE state. This bit will be cleared to 0 after the controller is back to IDLE state.

RUNCMD: Writing 1 to this bit will initiate the SD command on the SD bus according to current configuration of the controller. This bit will be cleared to 0 after the transaction starts. A new transaction can be only be started when BUSY bit is 0.

CMDWD: Indicate if this command is with data

0: Command without data

1: Command with data

TRANDATA: Indicate whether this command is to transfer or receive data

0: Receive data (Read)

1: Transfer data (Write)

MUL BLK: Indicate whether it is multi-block transfer or not

0: Single block transfer

1: Multiple block transfer

INICARD: Writing 1 to this bit will start 74 clock cycles on the clock line.

RESPTYPE: Indicate the response type of this command. Currently, only the response type R2 has a 128-bit response length, and all other response will have 32-bit length. Response type R1b will keep the controller to wait for busy signal on the SD bus.

000: No response

001: Response type R1

010: Response type R2

011: Response type R3

110: Response type R6

111: Response type R1

25.4.4 P_SD_ARGUMENT_DATA (R/W)(0x8818000C): SD Card Argument

Host writes the argument it wants to transfer to SD card in this register. Controller will use data in this register as the command argument to transfer to the card. The SD Command needs a 32-bit length.

Name	B31-B0
P_SD_ARGUMENT_DATA	Argument

Argument: Argument[15:0] transfer to SD Card

25.4.5 P_SD_INT_STATUS (R)(0x88180014): SD Card Controller Status

All the controller status and transaction status will be stored in this register. Host can get information of the controller to determine what to do.

Name	B7	B6	B5	B4	B3	B2	B1	B0
P_SD_INT_STATUS	DATBUFFULL	CMDBUFFULL	RSPCRCERR	RSPIDXERR	DATCOM	CMDCOM	CARDBUSY	BUSY
Name	**B15**	**B14**	**B13**	**B12**	**B11**	**B10**	**B9**	**B8**
P_SD_INT_STATUS			CARDINT	CARDPRE	CARDWP	DATCRCERR	TIMEOUT	DATBUFEMPTY

BUSY: Indicate whether the controller is busy.

0: Controller is idle.

1: Controller is busy.

CARDBUSY: Indicate whether the SD card is busy (drive the DAT0 low). Host needs to poll this bit after a write command is issued.

0: Card is not busy.

1: Card is busy.

CMDCOM: Indicate whether the corresponding response is received or generated by timeout after sending a command.

0: Response is received.

1: Response is generated by timeout.

DATCOM: Indicate whether data transmission/receiving is completed.

0: Not completed

1: Completed

RSPIDXERR: Indicate whether the command index in the response is failed.

0: Not failed.

1: Failed

RSPCRCERR: Indicate the CRC bit in the response is failed. This bit will be set if the CRC received is not 6'b111111 in the case of response R3.

CMDBUFFULL: Indicate the RESP register is full. Reading RESP register, starting a new transaction, or setting STPCMD in command register will clear this bit.

DATBUFFULL: This bit will be set when data stored in FIFO are higher than the trigger level. This bit will be cleared after appropriate number of data are read from the DATARx register or 1 is written to STPCMD bit in command register.

DATBUFEMPTY: This bit will be set when data stored in the FIFO are lower than the trigger level. This bit will be cleared after appropriate number of data are written to the DATATx register or 1 is written to STPCMD bit in command register.

TIMEOUT: Indicate whether it is command response timeout or read data response timeout.

DATCRCERR: Indicate whether it is read data with CRC error or write data with CRC error response.

CARDWP: Indicate whether the card is write-protect. This bit only detects the write-protect pin on the interface. Controller's behavior will not be affected by this bit. Host needs to protect the card.

CARDPRE: Indicate whether the card is present. This bit only detects the DAT3 on the SD interface when the controller is idle. Controller's behavior will not be affected by this bit. Host can initiate a transaction no matter what this bit is. Writing 1 to this bit will clear the pending interrupt of card presence.

CARDINT: Indicate a SD IO card interrupt is pending. This bit will be set only when IOEN in control register is 1. Host needs to clear the interrupt using device specific command. Writing 1 to this bit will have no effect.

25.4.6 P_SD_RESPONSE_DATA (R) (0x88180010): SD Card Response

The response of the SD card will be stored in this register. Note the controller will not

interpret the information in this register. Host driver needs to take care of this.

Commands with response R1, R1b, R3, R6 have 6-bit command index and 32-bit response length. The response will be stored in this register. Command with response R2 has 128-bit response length. Host needs to poll the CMDBUFFULL bit in the status register to determine when to read this register. The data in this register is valid only when CMDBUFFULL bit is 1.

Name	B31-B0
P_SD_RESPONSE_DATA	Response

Response: Response data from SD card. Read data from this register will be valid only when the **CMDBUFFULL** is set.

25.4.7 P_SD_MODE_CTRL(R/W)(0x88180018): SD Card Control

The register controls the clock speed on the SD bus. The block length is also controlled by this register during data transmission or receiving. This register is changeable only when BUSY bit in status register is 0.

Name	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P_SD_MODE_CTRL					EN_SD	IOEN	DMAMODE	BUSWIDTH	CLKDIV							
Name	B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
P_SD_MODE_CTRL					BLKLEN											

CLKDIV: The Clock Speed on the SD bus is calculated from this register.

$F_{SDCLK} = F_{SYSCLK}/2(CLKDIV+1)$, and default value is 0x54.

BUSWIDTH: Indicate the data bus width during transfer

0: 1 bit

1: 4 bits

DMAMODE: Indicate whether it is DMA mode or not

IOEN: SD IO card interrupt detection Enable

0: Disable

1: Enable

EN_SD: Indicate if the IO is used by the SD controller. Programmer must write 1 to this bit before using the SD controller.

0: SDCLK, SDCMD, SDDAT0 are used as GPIOs.

1: SDCLK, SDCMD, SDDAT0 are used as SD control signals.

If BUSWIDTH is set to 1, the SDDAT1, SDDAT2, SDDAT3 will become SD control signals when this bit is set to 1.

BLKLEN: Data block length to be transferred (unit: byte). The values in this register should be equal to the block length of the SD/MMC card.

25.4.8 P_SD_INT_CTRL(R/W) (0x8818001C): SD Card Interrupt

This register is used to enable/disable the interrupt of the SD memory card controller.

Name	B15-B7	B6	B5	B4	B3	B2	B1	B0
P_SD_INT_CTRL		INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0

INTEN0: 0: Disable CMDCOM interrupt

1: Enable CMDCOM interrupt, and fire interrupt when **CMDCOM** is set.

INTEN1: 0: Disable **DATCOM** interrupt.

1: Enable **DATCOM** interrupt, and fire interrupt when **DATCOM** is set.

INTEN2: 0: Disable **CMDBUFFULL** interrupt.

1: Enable CMDBUFFULL interrupt, and fire interrupt when **CMDBUFFULL** is set.

INTEN3: 0: Disable **DATBUFFULL** interrupt.

1: Enable **DATBUFFULL** interrupt, and fire interrupt when **DATBUFFULL** is set.

INTEN4: 0: Disable **DATBUFEMPTY** interrupt.

1: Enable **DATBUFEMPTY** interrupt, and fire interrupt when **DATBUFEMPTY** is set.

INTEN5: 0: Disable card insertion/removement interrupt.

1: Enable card insertion/removement interrupt.

INTEN6: 0: Disable SD IO card interrupt.

1: Enable SD IO card interrupt.

26 Universal File System – UFAT Library

26.1 Introduction

UFAT, Universal FAT, is a file system that can be used in most embedded applications with file-like storage. The mainly UFAT API functions of SPCE3200 UFAT LIB are listed as follows.

API Name	Description
Open	Open the specified file with the specified operation mode
Read	Read bytes from the file specified by a file node index
Write	Write bytes into the file specified by a file node index
Close	Close the file specified by the file node index

26.2 Structure

UFAT is designed as the following structure:

Interface Layer	APIs for the application developers
API Layer	Operations for files and directories
File System Layer	Operation for FAT and clusters
Logic Block Layer	Sector based access and caching
Driver Layer	Low level device access

Interface Layer: A thin layer translates the API layer function calls and error numbers into POSIX function calls and error numbers.

API Layer: Operates files and directories.

File System Layer: Operates the FAT and clusters by following the FAT storage rules.

Logic Block Layer: A sector base access interface is defined in this layer to read sectors or write sectors with cache.

Driver Layer: Provides a hardware independent access interface for the upper layer. SPCE3200 UFAT library supports both **FAT16** and **FAT32** storage formats.

26.3 API Functions

26.3.1 open

Open a specified file.

int open (const char *filename, int flags)

Parameters:

Filename: Pointer to a path name string.

Flags:

Flags	Description
O_OPEN	Open an existing file
O_TRUNC	Open the file and truncate the file to zero length
O_CREAT	If set, the file will be created in case it doesn't exist yet
O_RDONLY	Open the file for reading
O_WRONLY	Open the file for writing
O_RDWR	Open the file for both reading and writing
O_EXCL	If set, the open operation will fail when trying to create an existing file

Return value: When successful, the return value of open() is a non-negative integer file handle. Return -1 when error occurs.

26.3.2 close

Close a file handle.

int close (int filehd)

Parameters:

filehd: Integer file handle.

Return value: When successful, return 0; -1 when error occurs.

26.3.3 read

Read bytes from the specified file handle.

Int read (int filehd, void *buffer, unsigned int size)

Parameters:

filehd: Integer file handle returned from open function.

buffer: Pointer to the buffer for storing the data.

size: Bytes to read.

Return values: When successful, it will return the actual number of bytes read from the specified file handle. Return -1 when error occurs.

26.3.4 write

Write bytes into a specified file handle.

Int write (int filehd, void *buffer, unsigned int size)

Parameters:

filehd:	Integer file handle returned from open function.
buffer:	Pointer to the buffer for writing.
size:	Bytes to write.
Return value:	When successful, it will return the actual number of bytes written to the specified file handle. Return -1 when error occurs.

27 NAND-Type Flash

27.1 Introduction

A NAND-type flash controller is embedded in SPCE3200 for mass storage application. The controller supports standard 8-bit NAND flash equipped with ECC calculation circuit. The smart media card is also supported.

27.2 Features

- Fully programmable CLE and ALE sequences which can support various kinds of NAND flash
- Supports polling/interrupt/DMA access method
- Supports page size from 528 bytes to 2112 bytes
- Hardware ECC calculation circuit
- Programmable read/write cycles
- Programmable ALE cycles

27.3 Block Diagram

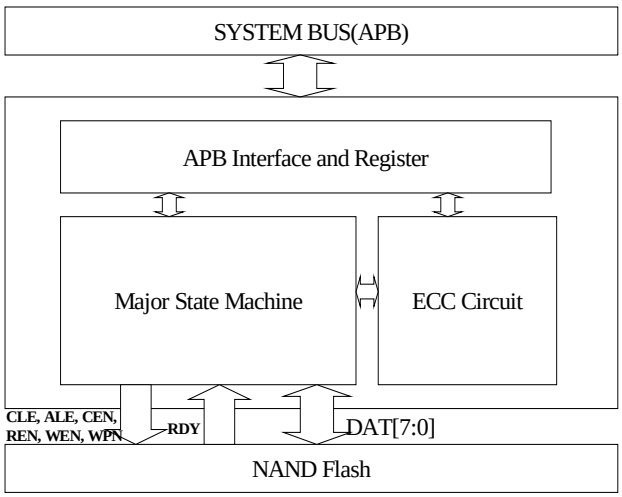


Fig 27-1 Functional Block Diagram of Flash Controller

27.4 Interface Signals

27.4.1 APB Bus Signals

This part is the same as define in AMBA specification.

27.4.2 Flash Bus Signals

CEN	O	Chip select of NAND flash, low active
CLE	O	Command latch enable, high active
ALE	O	Address latch enable, high active
WEN	O	Write enable, low active
REN	O	Read enable, low active
WPN	O	Write protect enable, low active
RDY	I	NAND flash ready input, high active
DATA[7:0]	I/O	Data bus

27.4.3 Other Signals

INT	O	Interrupt pin, high active
DMARQ	O	DMA request signal, high active

27.5 Control Registers

27.5.1 P_NAND_MODE_CTRL(R/W)(0x88190000): Flash Control

Flash controller control register. This register is used for host to control the behavior of controller, say, the timing setting of page size setting. Programmer must program this register based on the specification of NAND flash.

NAME	D7-D6	D5-D4	D3	D2	D1	D0
P_NAND_MODE_CTRL	ALECYC	RDTYPE	WRCMD	FLWPN	FLCEN	FLEN
NAME	**D31-D28**	**D27-D16**	**D15-D14**	**D13-D12**	**D11-D10**	**D9-D8**
P_NAND_MODE_CTRL		TBYTES	RDHIGH	RDLOW	WRHIGH	WRLOW

FLEN: Flash interface enable register.

0: Disable

1: Enable

FLCEN: CEN control register.

0: CEN output 0

1: CEN output 1

FLWPN: WPN control register.

0: WPN output 0

1: WPN output 1

WRCMD: Read/Write command control

0: Read Command

1: Write Command

RDTYPE: Indicate when to start the read process

00: Read after RDY rise

01: Read after CLE fall

10: Read after ALE fall

11: Reserved

ALECYC: Indicate how many write cycles will be issued when P_NAND_ALE_ADDR is written

00: One write cycle

01: Two write cycles

10: Three write cycles

11: Four write cycles

WRLOW: Indicate the clock cycles of WEN low pulse

00: One Clock cycle

01: Two Clock cycle

10: Three Clock cycles

11: Four Clock cycles

WRHIGH: Indicate the clock cycles of WEN high pulse.

00: One clock cycle

01: Two clock cycles

10: Three clock cycles

11: Four clock cycles

RDLOW: Indicate the clock cycles of REN low pulse.

00: One clock cycle

01: Two clock cycles

10: Three clock cycles

11: Four clock cycles

RDHIGH: Indicate the clock cycles of REN high pulse.

00: One clock cycle

01: Two clock cycles

10: Three clock cycles

11: Four clock cycles

TBYTES: Total bytes -1 of this transfer

27.5.2 P_NAND_CLE_COMMAND(R/W)(0x88190004): Command Latch Enable

This register is used to control the CLE of NAND flash bus. If programmer writes data to this register when controller is not busy, a CLE pulse will be issued with command data written in this register.

NAME	D7-D0
P_NAND_CLE_COMMAND	FL_CLE

FL_CLE: If programmer writes data to this register when controller is not busy, a CLE pulse will be issued with command data written in this register.

27.5.3 P_NAND_ALE_ADDR(R/W)(0x88190008): Address Latch Enable

This register is used to control the ALE of NAND flash bus. If programmer writes data to this register when controller is not busy, an ALE pulse will be issued with address data written in this register. The cycles of ALE command are defined in FL_CR register.

NAME	D31-D0
P_NAND_ALE_ADDR	FL_ALE

FL_ALE: If programmer writes data to this register when controller is not busy, an ALE pulse will be issued with command data written in this register.

27.5.4 P_NAND_TX_DATA(R/W)(0x8819000C): Write Data

This register is used to write data to NAND flash bus. Programmer must write data to this register when WREMPTY is 1. Otherwise, the WRLOSS bit will be set and the written data will lose.

NAME	D31-D0
P_NAND_TX_DATA	FL_WD

FL_WD: This register is used to write data to NAND flash bus. Programmer must write data to this register when WREMPTY is 1.

27.5.5 P_NAND_RX_DATA(R/W)(0x88190010): Read Data

This register is used to read data from NAND flash bus. Programmer must read data

from this register when RDFULL is 1. Otherwise, the RDMISS bit will be set and the read data will be invalid.

NAME	D31-D0
P_NAND_RX_DATA	FL_RD

FL_RD: This register is used to read data from NAND flash bus. Programmer must read data from this register when RDFULL is 1.

27.5.6 P_NAND_INT_CTRL(R/W)(0x88190014): Interrupt Enable

Host uses this register to control the interrupt output.

NAME	D5	D4	D3	D2	D1	D0
P_NAND_INT_CTRL	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0

INTEN0: Write Empty Interrupt enable

0: Disable

1: Enable

INTEN1: Read Full Interrupt enable

0: Disable

1: Enable

INTEN2: Ready Rise Interrupt enable

0: Disable

1: Enable

INTEN3: Write Loss Interrupt enable

0: Disable

1: Enable

INTEN4: Read Miss Interrupt enable

0: Disable

1: Enable

INTEN5: Command Loss Interrupt enable

0: Disable

1: Enable

27.5.7 P_NAND_INT_STATUS(R/C)(0x88190018): Interrupt Status

The register is used to read the status of flash controller.

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_NAND_INT_STATUS	RDY	BUSY	CMDLOSS	RDMISS	WRLOSS	RDYRISE	RDFULL	WREMPTY

WREMPTY: Write Empty Interrupt status

Read 0: Write-buffer is full. If data is further written to P_NAND_TX_DATA, Write Loss will become 1 and the write data will lose.

Read 1: Write-buffer is empty, and further writing is allowed.

NOTE: This bit will be cleared automatically when the write-buffer is written.

RD_FULL: Read Full Interrupt status

Read 0: Read-buffer is empty. If data is further read from P_NAND_RX_DATA, Read Miss will become 1 and the read data will be invalid.

Read 1: Read buffer is full, and reading from FL_RD is allowed.

NOTE: This bit will be cleared automatically when the read buffer is read.

RDYRISE: Ready Rise interrupt status

Read 0: RDY rise does not happen.

Read 1: RDY rise happens.

Write 0: No effect.

Write 1: Clear this bit.

WRLOSS: Write Loss Interrupt status

Read 0: Write loss event does not happen.

Read 1: Write loss event happens.

Write 0: No effect.

Write 1: Clear this bit.

RDMISS: Read Miss Interrupt status

Read 0: Read miss event does not happen.

Read 1: Read miss event happens.

Write 0: No effect.

Write 1: Clear this bit.

CMDLOSS: Command Loss Interrupt status

Read 0: Command loss event does not happen.

Read 1: Command loss miss event happens.

Write 0: No effect.

Write 1: Clear this bit.

BUSY: Controllers busy flag

0: Controller is not busy, and ALE/CLE command is allowed.

1: Controller is busy, and ALE/CLE command is not allowed. If an ALE/CLE command is received in this state, CMDLOSS will be set to 1 and the command will lose.

RDY: NAND flash busy flag. This bit will show the status of RDY on the NAND flash bus.

0: NAND flash is busy.

1: NAND flash is not busy.

27.5.8 P_NAND_ECC_TRUELP(R/W)(0x8819001C): True Line Parity

Programmer can use this register to write the true LP result for ECC circuit to calculate the error bits in a page.

NAME	D31-D16	D15-D0
P_NAND_ECC_TRUELP	TRUELP1	TRUELP0

TRUELP0: The true line parity of bytes 0~255 in a page. This register is written by programmer and used in error line/bit calculation.

TRUELP1: The true line parity of bytes 256~511 in a page. This register is written by programmer and used in error line/bit calculation.

27.5.9 P_NAND_ECC_TRUECP(R/W)(0x88190020): True Column Parity

Programmer can use this register to write the true CP result for ECC circuit to calculate the error bits in a page.

NAME	D11-D6	D5-D0
P_NAND_ECC_TRUECP	TRUECP1	TRUECP0

TRUECP0: The true column parity of bytes 0~255 in a page. This register is written by programmer and used in error column/bit calculation.

TRUECP1: The true column parity of bytes 256~511 in a page. This register is written by programmer and used in error column/bit calculation.

27.5.10 P_NAND_ECC_CMPLP(R)(0x88190024): Calculated Line Parity

Programmer can read the line parity of ECC calculation result from this register.

NAME	D31-D16	D15-D0
P_NAND_ECC_CMPLP	CALLP1	CALLP0

CALLP0: The line parity of bytes 0~255 in a page calculated by ECC circuit. This register is written by ECC circuit and used in error line/bit calculation.

CALLP1: The line parity of bytes 256~511 in a page calculated by ECC circuit. This register is written by ECC circuit and used in error line/bit calculation.

27.5.11 P_NAND_ECC_CMPCP(R)(0x88190028): Calculated Column Parity

Programmer can read the column parity of ECC calculation result from this register.

NAME	D11-D6	D5-D0
P_NAND_ECC_CMPCP	CALCP1	CALCP0

CALCP0: The column parity of bytes 0~255 in a page calculated by ECC circuit. This register is written by ECC circuit and used in error column/bit calculation.

CALCP1: The column parity of bytes 256~511 in a page calculated by ECC circuit. This register is written by ECC circuit and used in error column/bit calculation.

27.5.12 P_NAND_ECC_STATUS(R)(0x8819002C): ECC Status

The error line and error bit will be shown on this register.

NAME	D31-D30	D29-D28	D26-D24	D23-D16	D15-D14	D13-D11	D10-D8	D7-D0
P_NAND_ECC_STATUS	ERR1		ERRBIT1	ERRBYTE1	ERR0		ERRBIT0	ERRBYTE0

ERRBYTE0: The error byte of bytes 0~255 detected by ECC circuit

ERRBIT0: The error bit of ERRBYTE0 detected by ECC circuit

ERR0: Error detection of bytes 0~255 in a page.

00: No error is detected

01: One error is detected, correctable.

10: Reserved

11: Two or more errors are detected, non-correctable.

ERRBYTE1: The error byte of bytes 256~511 detected by ECC circuit

ERRBIT1: The error bit of ERRBYTE1 detected by ECC circuit

ERR1: Error detection of bytes 256~511 in a page.

00: No error is detected

01: One error is detected, correctable.

10: Reserved

11: Two or more errors are detected, non-correctable.

27.5.13 P_NAND_ECC_CTRL(W)(0x88190030): ECC Control

This register is used to control the ECC circuit.

NAME	D2	D1	D0
P_NAND_ECC_CTRL	CLRECC1	CLRECC0	CALECC

CALECC: Write 0: No effect.

Write 1: The ECC circuit will use P_NAND_ECC_TRUELP, P_NAND_ECC_TRUECP, P_NAND_ECC_CMPLP, and P_NAND_ECC_CMPCP to find the error byte and error bit.

This bit will be cleared after the calculation is done.

CLRECC0: Write 0: No effect.

Write 1: The ECC part 0 will be reset.

This bit will be cleared after the calculation is done. This bit is used when the page size is larger than 528 bytes.

CLRECC1: Write 0: No effect.

Write 1: The ECC part 1 will be reset.

This bit will be cleared after the calculation is done. This bit is used when the page size is larger than 528 bytes.

28 TFT LCD Control

28.1 Introduction

The built-in TFT LCD interface supports multiple TFT LCD panel input formats, such as DataEnable (DE) mode, Hsync/Vsync mode, 15-bit parallel RGB mode, 8-bit delta RGB mode, and CCIR601/656 mode with resolution 320(H) X 240(V), and NTSC/PAL formats. The configurable position and width of synchronous signal can fit various TFT LCD panel specifications.

28.2 Output Interface

Following is the IO pins related to TFT controller:

- Clk: TFT LCD clock
- Data_en: data enable
- Vsync: vertical synchronous signal
- Hsync: horizontal synchronous signal
- Data[15:0]: data bus

28.3 Data Format Supported

The LCD TFT controller supports the following data formats:

- Parallel RGB
- Serial RGB
- Serial RGBDm
- CCIR-601/CCIR-656

28.4 Clock Frequency

- System Clock (27MHz)
- System Clock/2 (13.5MHz)
- System Clock/4 (6.75MHz)
- System Clock/8 (3.375MHz)

28.5 Control Registers

28.5.1 P_TFT_MODE_CTRL(0x88040000): TFT LCD Control

NAME	D24	D23-D12	D11-D10	D9-D8	D7-D2	D1-D0
P_TFT_MODE_CTRL	TFT_EN		VER_SCALING	HOR_SCALING		TFT_CLK_SEL

TFT_EN: TFT LCD Enable

VER_SCALING: Image Scaling In Vertical

0: No scaling

1: The display image in vertical is scaling up to double size

2: The display image size in vertical is scaling down

HOR_SCALING: Image scaling in horizontal

0: No scaling

1: The display image in horizontal is scaling up to double size

2: The display image is scaling down to half size

3: No scaling

TFT_CLK_SEL: TFT LCD output clock selection

0: System clock (27MHz)

1: System clock /2 (13.5MHz)

2: System clock /4 (6.75MHz)

3: System clock/8 (3.875MHz)

28.5.2 P_TFT_DATA_FMT(0x88040004): Data Format

NAME	D8	D7-D3	D2-D0
P_TFT_DATA_FMT	CCIR656_M		TFT_FMT

CCIR656_M: This function is for standard CCIR656.

1: The controller will output 720 valid pixels, but it contains 80 pixels black in two sides of picture.

0: The controller will output 640 valid pixels.

TFT_FMT: TFT LCD output data format

0: Parallel RGB

1: Serial RGB

2: Serial RGBDm

3: CCIR 601

4: CCIR 656

28.5.3 P_TFT_HOR_ACT(0x88040008)

NAME	D11-D0
P_TFT_HOR_ACT	HOR_ACT

HOR_ACT: The horizontal active area.

Unit: system clock

28.5.4 P_TFT_HOR_FRONT(0x8804000C)

NAME	D9-D0
P_TFT_HOR_FRONT	HOR_FBLK

HOR_FBLK: The front horizontal blanking area

Unit: system clock

28.5.5 P_TFT_HOR_BACK(0x88040010)

NAME	D9-D0
P_TFT_HOR_BACK	HOR_BBLK

HOR_BBLK: The back horizontal blanking area

Unit: system clock

28.5.6 P_TFT_HOR_SYNC(0x88040014)

NAME	D24	D23-D8	D7-D0
P_TFT_HOR_SYNC	HS_P		HOR_SYNCW

HS_P: The polarity of horizontal synchronous pulse.

0: Negative

1: Positive

HOR_SYNCW: The width of horizontal synchronous pulse

Unit: system clock

28.5.7 P_TFT_VER_ACT(0x88040018)

NAME	D9-D0
P_TFT_VER_ACT	VER_ACT

VER_ACT: The vertical active area.

Unit: line

28.5.8 P_TFT_VER_FRONT(0x8804001C)

NAME	D7-D0
P_TFT_VER_FRONT	VER_FBLK

VER_FBLK: The front vertical blanking area.

Unit: line

28.5.9 P_TFT_VER_BACK(0x88040020)

NAME	D7-D0
P_TFT_VER_BACK	VER_BBLK

VER_BBLK: The back vertical blanking area.

Unit: line

28.5.10 P_TFT_VER_SYNC(0x88040024)

NAME	D24	D23-D5	D4-D0
P_TFT_VER_SYNC	VS_P		VER_SYNCW

VS_P: The polarity of vertical synchronous pulse.

0: Negative

1: Positive

VER_SYNCW: The width of vertical synchronous pulse

Setting range: 1 ~ 31

28.5.11 P_TFT_FRAME_FMT1(0x88040028)

NAME	D1-D0
P_TFT_FRAME_FMT1	FB_FMT

FB_FMT: Frame buffer data format.

0: RGB565 (little endian)

1: RGB1555 (big endian)

2: YUYV (little endian)

3: 4Y4U4Y4V (little endian)

28.5.12 P_TFT_ROW_START(0x8804002C)

NAME	D9-D0
P_TFT_ROW_START	STR_LNO

STR_LNO: The number of start line in frame buffer.

Unit: line

28.5.13 P_TFT_COL_START(0x88040030)

NAME	D9-D0
P_TFT_COL_START	STR_PNO

STR_PNO: The number of start pixel in frame buffer.

It must be the multiple of 16.

Unit: pixel

28.5.14 P_TFT_COL_WIDTH(0x88040034)

NAME	D9-D0
P_TFT_COL_WIDTH	PIX_NUM

PIX_NUM: The number of image pixel of one line in frame buffer.

It must be the multiple of 16.

Unit: pixel (16 bits)

28.5.15 P_TFT_DUMMY_WIDTH (0x88040038)

NAME	D9-D0
P_TFT_DUMMY_WIDTH	DUMP_PIX

DUMP_PIX: The number of dummy pixel in frame buffer.

It must be the multiple of 16.

Unit: pixel (16 bits)

28.5.16 P_TFT_BUFFER_STATUS (0x8804003C)

NAME	D24	D23-D17	D16	D15-D0
P_TFT_BUFFER_STATUS	BUF_ERR		TFT_FINISH	

BUF_ERR: TFT LCD buffer underflow.

When reading is faster than writing in LCD buffer, it will be set “1” by TFT LCD controller.

Read 0: Doesn’t occur

Read 1: Happens

Write 1: Clear the flag

TFT_FINISH: When this bit is set “1”, it means the TFT controller completes internal operation and users can turn off the TFT clock.

28.5.17 P_TFT_DATA_SEQ (0x88040040)

NAME	D24	D23-D19	D18-D16	D15-D11	D10-D8	D7-D3	D2-D0
P_TFT_DATA_SEQ	YUV_FMT		YUV_SEQ		RGB_OLSEQ		RGB_ELSEQ

YUV_FMT: Output data is YUV/YCbCr.

0: YCbCr

1: YUV

YUV_SEQ: YUV data output sequence for CCIR601 and CCIR656

000: Y0U0Y1V0 / Y0Cb0Y1Cr0

001: Y0V0Y1U0 / Y0Cr0Y1Cb0

010: U0Y0V0Y1 / Cb0Y0Cr0Y1

011: V0Y0U0Y1 / Cr0Y0Cb0Y1

100: Y1U0Y0V0 / Y1Cb0Y0Cr0 (reserved for test)

101: Y1V0Y0U0 / Y1Cr0Y0Cb0 (reserved for test)

110: U0Y1V0Y0 / Cb0Y1Cr0Y0 (reserved for test)

111: V0Y1U0Y0 / Cr0Y1Cb0Y0 (reserved for test)

RGB_OLSEQ: RGB data output sequence in odd line for serial RGB and serial RGBDm

000: RGB (RGBDm)

001: GBR (GBRDm)

010: BRG (BRGDm)

011: RBG (RBGDm)

100: BGR (BGRDm)

101: GRB (GRBDm)

RGB_ELSEQ: RGB data output sequence in even line for serial RGB and serial RGBDm

000: RGB (RGBDm)
001: GBR (GBRDm)
010: BRG (BRGDm)
011: RBG (RBGDm)
100: BGR (BGRDm)
101: GRB (GRBDm)

28.5.18 P_TFT_INT_STATUS (0x88040050)

NAME	D24	D23-D17	D16	D15-D0
P_TFT_INT_STATUS	INT_EN		VBLK_INT	

INT_EN: The blanking interrupt of vertical enable

VBLK_INT: The blanking interrupt of vertical

Read 0: Doesn't occur
Read 1: Happens
Write 1: Clear the flag

28.5.19 P_TFT_FRAME_FMT2 (0x880400A0)

NAME	D31-D1	D0 (R/W)
P_TFT_FRAME_FMT2	Reserved	FB_YCbCr

FB_YCbCr: Frame buffer data format is YCbCr/YUV for YUYV and 4Y4U4Y4V mode.

0: YUV
1: YCbCr

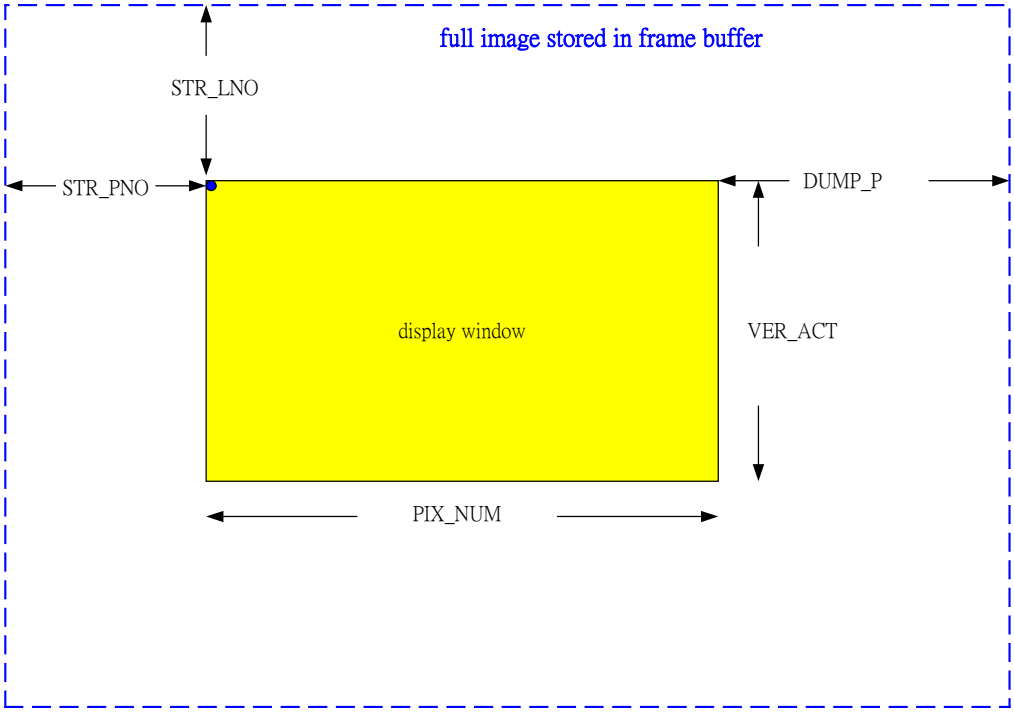


Fig 28-1 TFT LCD Screen Structure

29 STN LCD Controller

29.1 Introduction

The SPCE3200 contains a powerful LCD controller, which can support resolution up to 320(H) X 320(V). The STN LCD controller supports 4096 colors for color STN. It also contains a built-in hardware scroll function to minimize software overhead. Moreover, the interface supports flexible 4- or 8-bit data interface to connect with various LCD panels.

29.2 Features

- Supports resolution up to 320(H) X 320(V) LCD panel.
- Supports 4096 color STN

29.3 Interface Signals

Name	I/O	Description
FP	O	LCD interface, frame signal
LP	O	LCD interface, line signal
LACD	O	LCD interface, alternating signal
LD[7:0]	O	LCD interface, data bus
LCLK	O	LCD interface, dot clock

29.4 Control Registers

29.4.1 P_STN_MODE_CTRL (0x88041000)

NAME	D28	D27-D25	D24	D22-D17	D16	D15-D9	D8	D7-D2	D1-D0
P_STN_MODE_CTRL	FPSIF		FPIEN		LCDEN		SELREF		BUSW

FPSIF: STN LCD FP Signal Interrupt Flag.

FP interrupt flag is asserted at FP rising edge and cleared by writing “1” to this bit.

Read 0: Doesn’t occur

Read 1: Happens

Write 1: Clear the flag

FPIEN: STN LCD FP Signal Interrupt Enable

If this bit is set to “1” and FP interrupt happens, hardware will issue an IRQ to CPU. If this bit is cleared to “0”, this interrupt will be masked

LCDEN: STN LCD Enable

SELREF: Self-Refresh Mode Enable

0: Don’t enter self-refresh mode

1: Enter self-refresh mode

When the LCD driver contains the built-in memory and supports the self-refresh mode, the LCDC module can be configured into self-refresh mode. The LCD driver shows the last display data, and the LCDC outputs LACD, FP, and LP signals only. The LCDCLK and LD signals will keep at LOW state until SelfRef is 0.

BUSW: LCD hardware data bus width configuration. 4 or 8 bits can be supported.

00: Reserved

01: 4 bits (LCDD[3:0] Valid)

10: 8 bits (LCDD[7:0] Valid)

11: Reserved

The data is re-packaged into 4- or 8-bit data bus with the control of BUSW bit and it is transferred to LCD driver through LD [3:0], or LD [7:0] respectively.

Color, Bus width=8

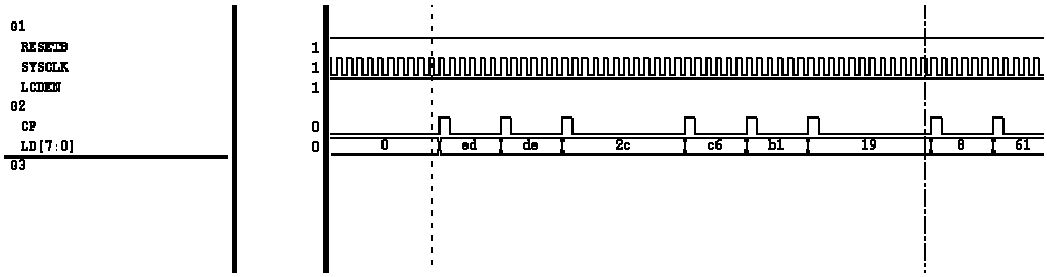


Fig 29-1 LCD Data Transmission on 8-Bit Data Bus

Color, Bus width=4

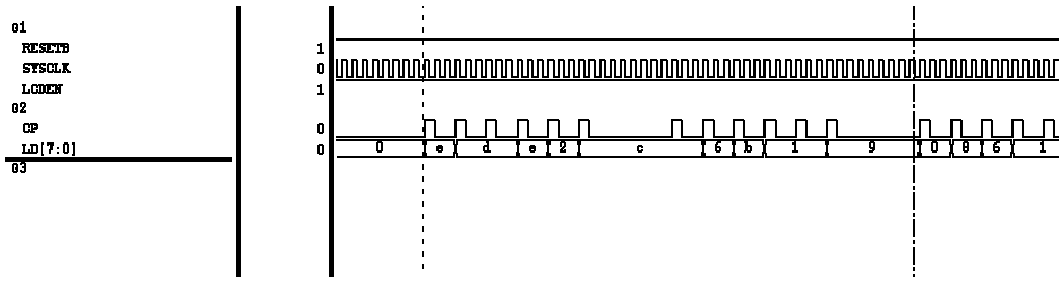


Fig 29-2 LCD Data Transmission on 4-Bit Data Bus

29.4.2 P_STN_PIXEL_CLK (0x88041004): Pixel Clock Divider

NAME	D9-D0
P_STN_PIXEL_CLK	LCDCLK

LCDCLK: Pixel Clock Divider

The SYSCLK signal is divided by N (= PCD [9:0] +2) to generate LCD clock.

Where N = SYSCLK / (COM * SEG * Frame Rate)

For example: SYSCLK = 48MHz

COM * SEG * Frame Rate	N
16 * 16 * 183	1025
80 * 80 * 178.6	42
160 * 160 * 187.5	10

COM: Number of common

SEG: Number of segment

Frame rate: Frequency of alternating signal (LACD).

29.4.3 P_STN_SEGMENT_NUM (0x88041008)

NAME	D8-D0
P_STN_SEGMENT_NUM	LCDSEG

LCDSEG: LCD Panel Segment Number Register

It must be the multiple of 16.

The horizontal size (in pixel) of the LCD panel, LCDSEG [8:0] = 16 ~ 320.

29.4.4 P_STN_COMMON_NUM (0x8804100C)

NAME	D8-D0
P_STN_COMMON_NUM	LCDCOM

LCDCOM: LCD Panel Common Number Register

LINEVAL [8:0]: The vertical size of the LCD panel where LINEVAL [8:0]= 16 ~ 320.

29.4.5 P_STN_ROW_START (0x88041010)

NAME	D9-D0
P_STN_ROW_START	STN_STR_LNO

STN_STR_LNO: The Start line number in frame buffer

Unit: line

29.4.6 P_STN_COL_START (0x88041014)

NAME	D9-D0
P_STN_COL_START	STN_STR_PNO

STN_STR_PNO: The Start pixel number in frame buffer

It must be the multiple of 16.

Unit: pixel (16 bits)

29.4.7 P_STN_DUMMY_WIDTH(0x88041018)

NAME	D9-D0
P_STN_DUMMY_WIDTH	STN_DMY_PIX

STN_DMY_PIX: The dummy pixel number in frame buffer

It must be the multiple of 16.

Unit: pixel (16 bits)

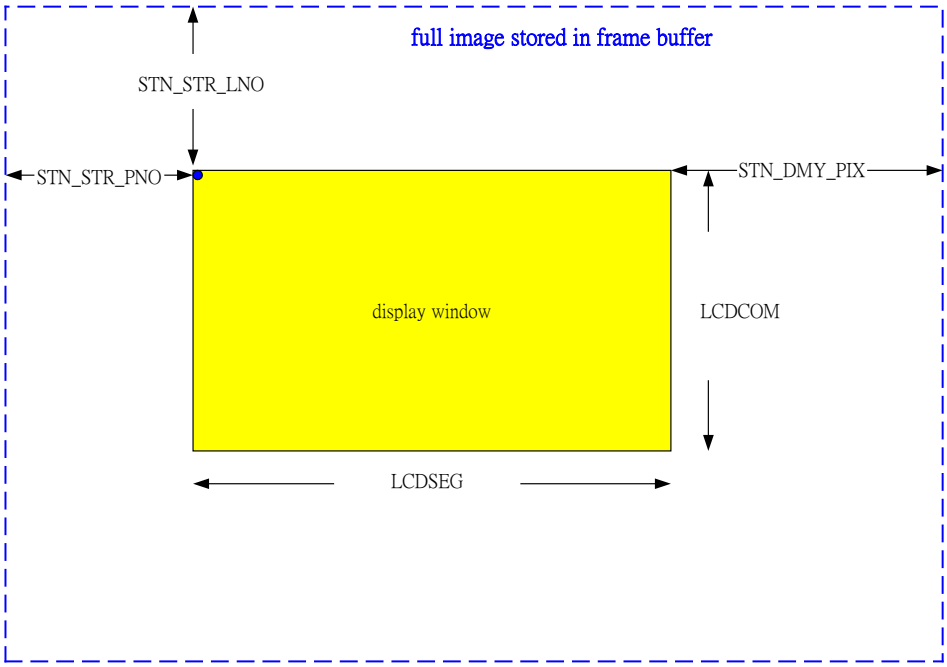


Fig 29-3 STN LCD Screen Structure

29.4.8 P_STN_LINE_COUNT(0x8804101C)

NAME	D19-D16	D15-D12	D11-D8	D7-D4	D3-D0
P_STN_LINE_COUNT	LBVL		LPW		LPCPD

LBVL: Line Blank Width

T= (LBVL+1) x CLCPCLK

LPW: LP Pulse Width

T= (LPW+1) x CLCPCLK

LPCPD: LP to CP Delay

T= (LPCPD+1) x CLCPCLK

LPW [3:0]: LP pulse width

LPCPD [3:0]: LP to CP delay

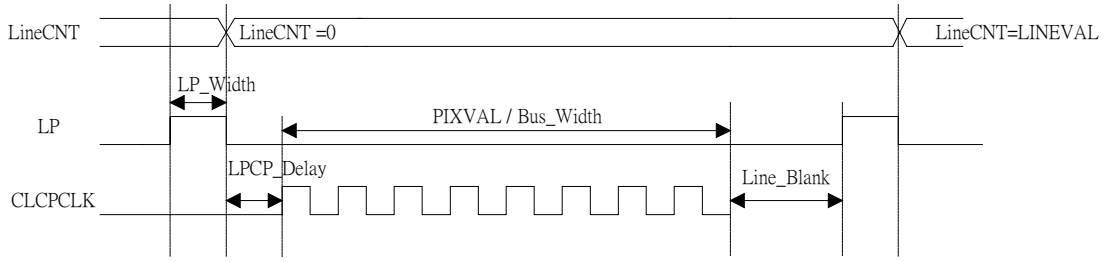


Fig 29-4 STN LCD Timing Diagram

- LBVL = 0H ~ FH

  Actual Line Blank width= (LBVL + 1) * CLCPCLK cycle time

- LPW = 0H ~FH

  Actual LP Pulse Width= (LPW + 1) * CLCPCLK cycle time

- LPCPD = 0H ~FH

  Actual LP to CP Delay= (LPCPD + 1) * CLCPCLK cycle time

29.4.9 P_STN_DATA_FMT(0x88041020)

NAME	D24	D23-D8	D7-D0
P_STN_DATA_FMT	BCMOD		MVAL

BCMOD: LCD Frame Modulation Type Selection

0: B type

1: C type

MVAL: Define the frequency of Frame Modulation when C type is active

When BCMOD=1, the frequency of LACD depends on MVAL +1.

That is, LACD changes its state every MVAL + 1 line.

- When BCMOD=0 (B type), LACD changes its state every FP signal.

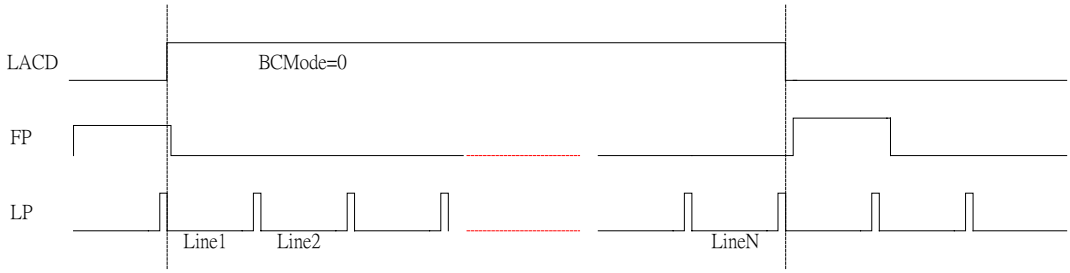


Fig 29-5 LACD State Change Timing Diagram (BCMOD=0)

- When BCMOD=1 and MVAL=2, LACD changes its state every three (=MVAL+1) LP signals.

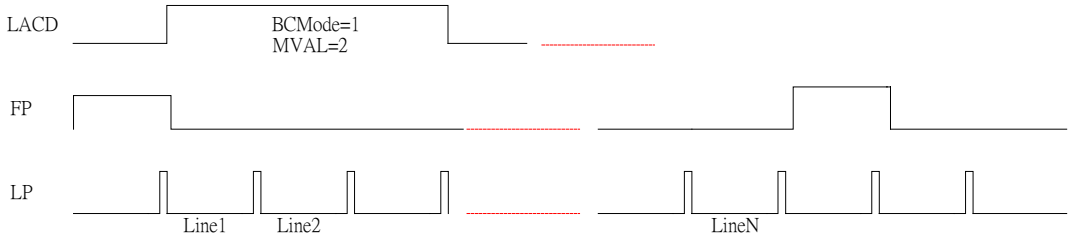


Fig 29-6 LACD State Change Timing Diagram (BCMOD=1)

29.4.10 P_STN_BUFFER_STATUS(0x88041024)

NAME	D24	D23-D17	D16	D15-D0
P_STN_BUFFER_STATUS	BUF_ERR		STN_FINISH	

BUF_ERR: STN LCD buffer underflow.

When reading is faster than writing in LCD buffer, it will be set to "1" by STN LCD controller.

Read 0: Doesn't occur

Read 1: Happens

Write 1: Clear the flag

STN_FINISH: When this bit is set to "1", it means the STN controller completes internal operation and users can turn off the STN clock.

29.4.11 P_STN_FRAME_FMT1 (0x88041028)

NAME	D12	D11-D9	D8	D7-D2	D1-D0
P_STN_FRAME_FMT1	VER_SCLDN		HOR_SCLDN		FB_FMT

VER_SCLDN: 0: No scaling.

1: The display image is scaling down to half size in vertical.

HOR_SCLDN: 0: No scaling.

1: The display image is scaling down to half size in horizontal.

FB_FMT: Frame buffer data format

0: RGB565 (little endian)

1: RGB1555 (big endian)

2: YUYV (little endian)

3: 4Y4U4Y4V (little endian)

29.4.12 P_STN_DATA_SEQ(0x8804102C)

NAME	D24	D23-D11	D10-D8	D7-D3	D2-D0
P_STN_DATA_SEQ	MOSAIC_EN		RGB_OLSEQ		RGB_ELSEQ

MOSAIC_EN: Mosaic Color STN LCD Enable

RGB_OLSEQ: RGB data output sequence in odd line for mosaic color STN LCD.

000: RGB

001: GBR

010: BRG

011: RBG
100: BGR
101: GRB

RGB_ELSEQ: RGB data output sequence in even line for mosaic color STN LCD.
000: RGB
001: GBR
010: BRG
011: RBG
100: BGR
101: GRB

29.4.13 P_STN_FRAME_FMT2 (0x880410A0)

Bit	Function	Type	Description	Default value
[31:1]			Reserved	
0	FB_YCbCr	R/W	Frame buffer data format is YCbCr/YUV for YUYV and 4Y4U4Y4V mode. 0: YUV 1: YCbCr	0

30 CMOS Sensor Interface – CSI Application

30.1 Frame Buffer Control

When CMOS Image Sensor captures an image, it will transmit it into DRAM using CMOS Sensor Interface (CSI), then all display-related modules including TV, LCD or JPG will be able access this image data, and directly or mix it with its own data to generate output. Users do not need to wait for the CMOS Sensor data transfer to be completed before using these data because CSI has an independent Frame Buffer already. This mechanism makes it possible for each of the TV, LCD and JPG module to work independently. BUFCTL is responsible for all the buffer control of each module, and the following Output Ports are for CSI module setting:

Name	I/O	Function Description
CSI_FRAME_END	output	A pulse to indicate CSI has completed the write-transfer of a frame. Every frame must have a FRAME_END pulse, even when the frame is disabled by the DISABLE_ CSI_FRAME input.

CSI_FRAME_END is driven by AHB master clock.

BUFCTL module has the following I/O ports:

Name	I/O	Function Description
CSI_FRAME_END	Input	A pulse to indicate CSI has completed the write-transfer of a frame. Every frame must have a FRAME_END pulse, even when the frame is disabled by the DISABLE_ CSI_FRAME input.
CSI_BUFFER_PTR[1:0]	Output	2'b00: write CSI frame buffer A 2'b01: write CSI frame buffer B 2'b10: write CSI frame buffer C
DISABLE_CSI_FRAME	Input	Disable the write-transfer for a whole frame, CSI holds HTRANS at IDLE. This input changes after the CSI_FRAME_END pulse.

CSI_BUFFER_PTR connects to MIU, DISABLE_CSI_FRAME connects to CSI.

30.2 Control Registers

CPU can access the internal control registers of the CSI module by the AHB slave. CSI has a disable register; it will wait for the completion of AHB transfer managed by MIU and

stop the data transfer. Using this status register bit, CSI will notify CPU that AHB data transfer managed by the MIU is finished when the AHB transfer is completed.

30.3 Time Generator

In order to interface with specific sensor, the parameters of TG are necessary to be set up, including TG_CR, TG_LSTART, TG_START, and TG_END.

30.3.1 P_CSI_TG_CTRL (0x88000000): Time Generator Control

NAME	D7	D6	D5	D4	D3	D2	D1	D0
P_CSI_TG_CTRL	BSEN	YUVOUT	YUVIN	CLKINV	RGB565	VGAEN	MASEN	CSIEN
NAME	D15	D14	D13	D12	D11	D10	D9	D8
P_CSI_TG_CTRL	INTERLACE	FIELDSEL	YUV_TYPE	VRST_TYPE	VADD_TYPE	HRST_TYPE	FGET_TYPE	CCIR656
NAME	D23	D22	D21	D20	D19	D18	D17	D16
P_CSI_TG_CTRL		RGB1555	QVGA27	CSICLKOINV	RESIZE	D_TYPE		
NAME	D23	D22	D29	D28	D27	D26	D25	D24
P_CSI_TG_CTRL			CUT_EN	HALF_VGA	BS_MODE	INV_UV	MP4OUT	BIGED

CSIEN: Sensor interface enable register

0: Disable

1: Enable

MASEN: SPCE Master/Slave mode selection

0: SPCE Slave mode (if connected with OV7648/OV7649)

1: SPCE master mode (if connected with S202/S201)

VGAEN: When SPCE is slave mode (MASEN is 0)

0: QVGA (each line contains 320 pixels)

1: VGA (each line contains 640 pixels)

When SPCE is master mode (MASEN is 1)

0: S202 is connected

1: S201 is connected

RGB565: The OV7648 has a RGB565 mode output. This bit is used to control it is RGB888 or RGB565 mode.

0: RGB888 mode

1: RGB565 mode

This bit is valid only when MASEN is 0 and YUVIN is 0

CLKINV: Sensor Clock inverse selection

0: Not inverse input sensor clock

1: Inverse input sensor clock

YUVIN: YUV input mode

0: Input data is RGB

1: Input data is YUV

This bit is valid only when MASEN is 0 (i.e. OV7648/OV7649 is connected)

YUVOUT: YUV output mode

0: Output data is RGBRGB

1: Output data is YUV422

BSEN: Blue Screen effect selection

0: Disable Blue Screen Effect

1: Enable Blue Screen Effect

CCIR656: CCIR601/CCIR656 selection

0: CCIR601 Interface

1: CCIR656 Interface

This bit is valid only when MASEN is 0.

FGET_TYPE: Field get type

0: Refer to field at falling edge of VSYHC

1: Refer to field at rising edge of VSYHC.

It's set to 1 if connected with OV7648

HRST_TYPE: Horizontal counter

0: Reset at the falling edge of HSYNC

1: Reset at the rising edge of HSYNC

VADD_TYPE: Vertical counter

0: Add at the falling edge of HSYNC

1: Add at the rising edge of HSYNC

VRST_TYPE: Vertical counter

0: Reset at the falling edge of VSYNC

1: Reset at the rising edge of VSYNC

YUV_TYPE: Sensor data sequence

0: UYVY (BGRB)

1: YUYU (GBGR)

FIELDSEL: The polarity of input FODD

0: Reversing is not necessary

1: Reverse the FODD signal

INTERLACE: Non-Interlaced/Interlaced mode selection

0: Non-Interlaced mode (regardless of field)

1: Interlaced mode

D_TYPE: The Parameter of data sequence control

RESIZE: Re-Size frame buffer control

0: Not re-size

1: VGA => QVGA re-size will be applied when VGA sensor is connected

CSICLKOINV: Invert sensor output clock

QVGA27: QVGA Input 27 MHz Mode (for OV7660's QVGA Mode)

RGB1555: RGB1555 mode, the bit 15 has two different functions.

BIGED: Big endian mode selection.

0: Little endian output

1: Big endian output

MP4OUT: MP4 Output format selection

0: YUYV output

1: 4Y4U4Y4V output

INV_UV: YUV or YcbCr selection

0: YUV output

1: YCbCr output

BS_MODE: Blue Screen Mode

0: For SPCE3200, blue screen is determined by bit 15 in RGB1555 mode.

Blue screen function does not work in other mode.

HALF_VGA: Half VGA mode (640x240) mode

0: Normal VGA mode (640x480)

1: Output frame buffer will have only 640x240:

In interlaced mode, the field input will be neglected. So the frame rate will be 60 when CSI clock is 27 MHz.

In non-interlaced mode, the even line will be neglected. So the frame rate will be 30 when CSI clock is 27 MHz.

CUT_EN: Cut screen function enable bit

0: Normal operation, the output frame buffer size is 320x240 or 640x480

1: Cut screen mode, only the selected output region will be written to the frame buffer.

30.3.2 P_CSI_LINE_START(0x88000004)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_LINE_START	TG_VL1START	TG_VL0START	TG_HLSTART

TG_HLSTART: The parameter of horizontal TG.

To get the actual starting pixel data at a line from sensor.

TG_VL0START: The parameter of vertical TG.

To get the actual starting line from sensor in field 0 (interlaced mode) or in each frame (non-interlaced mode)

TG_VL1START: The parameter of vertical TG.

To get the actual starting line from sensor in field 1 (interlaced mode).

30.3.3 P_CSI_WINDOW_START(0x88000008)

NAME	D24-D16	D15-D10	D9-D0
P_CSI_WINDOW_START	TG_VSTART		TG_HSTART

TG_HSTART: The horizontal start of a sensor window.

If it's 0, the left side contains 0 black column. If it's 20, the left side contains 20 black columns where the black color is defined in P_CSI_BACKGROUND_COLOR.

TG_VSTART: The vertical start of a sensor window.

If it's 0, the top side contains 0 black row. If it's 10, the top side contains 10 black rows where the black color is defined in P_CSI_BACKGROUND_COLOR.

30.3.4 P_CSI_WINDOW_END(0x8800000C):

NAME	D24-D16	D15-D10	D9-D0
P_CSI_WINDOW_END	TG_VEND		TG_HEND

TG_HEND: The horizontal end of a sensor window.

For example, in VGA mode, if it's 640, the right side contains 0 black column; if it's 620, the right side contains 20 black columns where the black color is defined in P_CSI_BACKGROUND_COLOR.

TG_VEND: The vertical end of a sensor window.

If it's 0, the bottom side contains 0 black row. If it's 10, the bottom side contains 10 black rows where the black color is defined in

P_CSI_BACKGROUND_COLOR.

30.3.5 P_CSI_BACKGROUND_COLOR(0x88000010)

NAME	D23-D0
P_CSI_BACKGROUND_COLOR	TG_BLACK

TG_BLACK: The Black color definition.

When YUVOUT is 0, this is RGB888.

When YUVOUT is 1, this is YUV888.

The following diagram shows the definition of these registers.

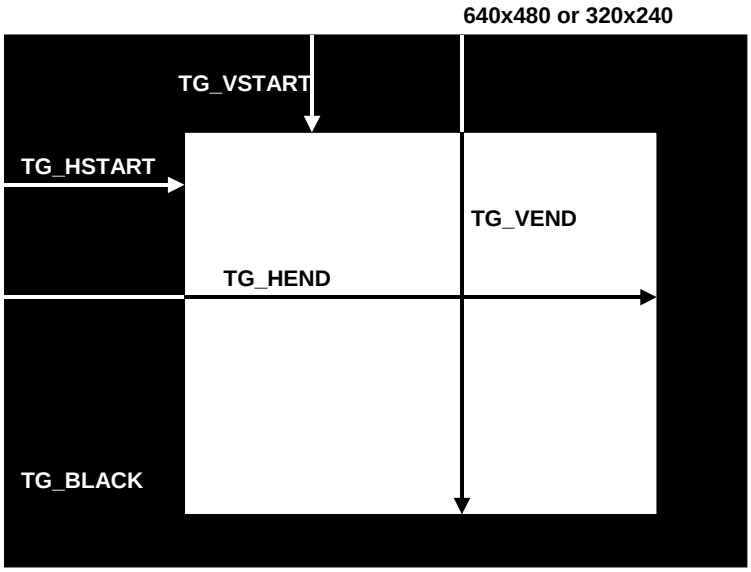


Fig 30-1 CSI Parameters

30.3.6 P_CSI_TOP_DETECT(0x88000014)

NAME	D23-D16	D15-D8	D7-D0
P_CSI_TOP_DETECT	TG_BSUPPER_B	TG_BSUPPER_G	TG_BSUPPER_R

TG_BSUPPER_R: The upper boundary of Red Color for Blue screen detection.

TG_BSUPPER_G: The upper boundary of Green Color for Blue screen detection.

TG_BSUPPER_B: The upper boundary of Blue Color for Blue screen detection.

30.3.7 P_CSI_BOTTOM_DETECT(0x08000018)

NAME	D23-D16	D15-D8	D7-D0
P_CSI_BOTTOM_DETECT	TG_BSLOWER_B	TG_BSLOWER_G	TG_BSLOWER_R

TG_BSLOWER_R: The lower boundary of Red Color for Blue screen detection.

TG_BSLOWER_G: The lower boundary of Green Color for Blue screen detection.

TG_BSLOWER_B: The lower boundary of Blue Color for Blue screen detection.

The blue screen region is defined in the following equations.

(TG_BSLOWER_R < R/(R+G+B) < TG_BSUPPER_R) AND

(TG_BSLOWER_G < G/(R+G+B) < TG_BSUPPER_G) AND

(TG_BSLOWER_B < B/(R+G+B) < TG_BSUPPER_B) AND **BSEN**

30.3.8 P_CSI_TRANSPARENT_COLOR(0x0800001C)

NAME	D23-D0
P_CSI_TRANSPARENT_COLOR	TG_TRANSP

TG_TRANSP: The transparent color definition.

When YUVOUT is 0, this is RGB888.

When YUVOUT is 1, this is YUV888.

This register will be used as the output data of sensor when a pixel is defined as the blue screen region.

30.3.9 P_CSI_BUFFER_SA1(0x88000020): Frame Buffer 1's start address

NAME	D31-D0
P_CSI_BUFFER_SA1	TG_FBSADDR1

30.3.10 P_CSI_BUFFER_SA2(0x88000024): Frame Buffer 2's start address

NAME	D31-D0
P_CSI_BUFFER_SA2	TG_FBSADDR2

30.3.11 P_CSI_BUFFER_SA3(0x88000028): Frame Buffer 3's start address

NAME	D31-D0
P_CSI_BUFFER_SA3	TG_FBSADDR3

PS: The frame buffer control is outside of sensor interface, and the external hardware will determine which frame buffer is going to be written.

30.3.12 P_CSI_CAPTURE_CTRL(0x8800002C): TG Capture Control

NAME	D0
P_CSI_CAPTURE_CTRL	TG_CAP

TG_CAP: Capture control bit.

0: The sensor will return to normal function.

1: The sensor will stop updating frame buffer after the current frame is updated completely. An interrupt will be asserted after the sensor stops.

The following figure shows the capture flow.

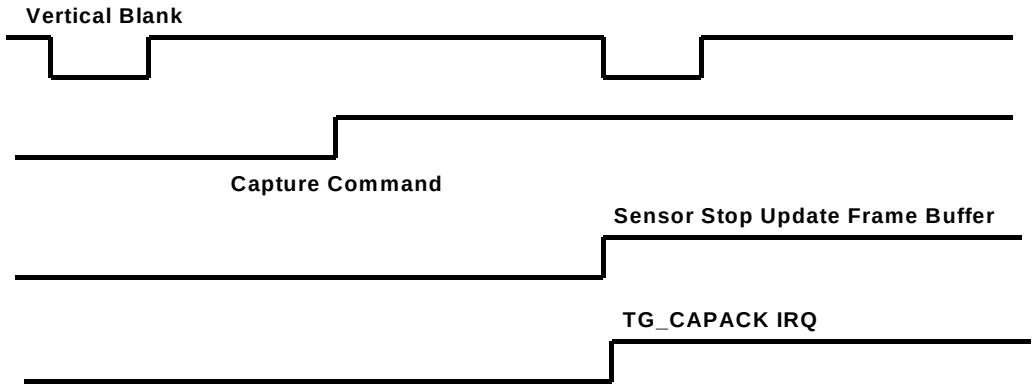


Fig 30-2 Capture Flow

30.3.13 P_CSI_CUT_SETUP(0x08000060)

NAME	D29-D24	D23-D21	D20-D16	D15-D14	D13-D8	D7-D5	D4-D0
P_CSI_CUT_SETUP	TG_HCUTSTART		TG_VCUTSTART		TG_HSIZE		TG_VSIZE

TG_VSIZE: The Vertical Cut region size in unit of 16 points (only 1~30 are allowed)

TG_HSIZE: The Horizontal Cut region size in unit of 16 points (only 1~40 are allowed)

TG_VCUTSTART: The vertical cut region start address in unit of 16 points (only 1~30

are allowed)

TG_HCUTSTART: The Horizontal cut region start address in unit of 16 points (only 1~40 are allowed)

The following diagram shows how the cut mechanism works.

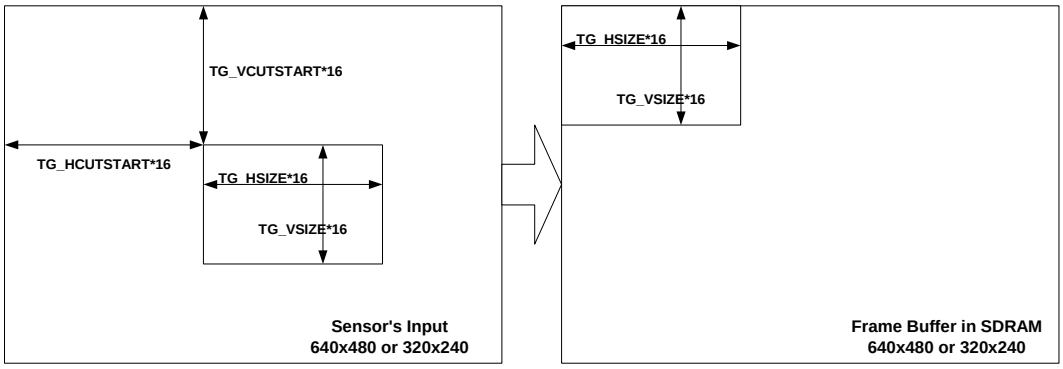


Fig 30-3 Cut Operation

You can change the position of the cut region in SDRAM by changing the TG_FBSADDR0~TG_FBSADDR2.

30.4 Motion Detect Control

30.4.1 P_CSI_MD_CTRL(0x88000030): Motion Detect Control

NAME	D15-D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
P_CSI_MD_CTRL	DIFFY_THRES	SAMP_TYPE	CC_WHITE		CC_BLACK		FRAME_TYPE		MD_TYPE	

MD_TYPE: Motion detect type selection bits.

00: Motion detect function disable

01: Reserved

10: SPCE3200's Y difference detect function

11: SPCE3200's Y difference detect function and color classification function

FRAME_TYPE: Control the sampling frequency of motion detect function

00: Sample in each frame

01: Sample in every two frames

10: Sample in every four frames

11: Sample in every eight frames

CC_BLACK: Black color definition of color classification circuit

CC_WHITE: White color definition of color classification circuit

SAMP_TYPE: Motion Detect size selection

0: 16x16

1: 8x8, when MD_TYPE is 0x11 and VGA mode is selected, this mode is not allowed.

DIFFY_THRES: The recognition function is:

DIFF = |Ynew – Yold|/2 >= DIFFY_THRES

Notes: DIFF bit is stored in address defined in MD_SADDR, for each word in memory: {Y3[7:1], Y2[7:1], Y1[7:1], Y0[7:1], DIFF3, DIFF2,DIFF1, DIFF0}.

Sample 2 is at the right side of sample 1.

30.4.2 P_CSI_RECOGNITION_SA(0x88000034)

NAME	D31-D0
P_CSI_RECOGNITION_SA	MD_SADDR

MD_SADDR: Recognition start address

In QVGA mode:

If recognition sample type 16x16 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 299

If recognition sample type 8x8 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 1199

In VGA mode:

If recognition sample type 16x16 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 1199

If recognition sample type 8x8 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 4799

30.4.3 P_CSI_POSITION_SETUP(0x88000038)

NAME	D24-D16	D15-D10	D9-D0
P_CSI_POSITION_SETUP	MD_VPOS		MD_HPOS

MD_HPOS: Horizontal IRQ position setting register

MD_VPOS: Vertical IRQ position setting register

NOTE: When the sensor controller detects the current position is equal to {MD_HPOS, MD_VPOS}, an interrupt (IRQ50) flag will be asserted and the color of this position will be stored in the P_CSI_CAPTURE_COLORVALUE register.

30.4.4 P_CSI_CLASSIFICATION_SA(0x8800003C)

NAME	D31-D0
P_CSI_CLASSIFICATION_SA	MD_SADDR1

MD_SADDR1: Color Classification start address.

In QVGA mode:

If recognition sample type 16x16 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 75

If recognition sample type 8x8 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 300

In VGA mode:

If recognition sample type 16x16 is used, the actual address mapping is:

MD_SADDR ~ MD_SADDR + 300

30.4.5 P_CSI_COLOR_TABLE1(0x88000040)

NAME	D31-D0
P_CSI_COLOR_TABLE1	MD_CTABLE0

MD_CTABLE0: Color Classification table 0, color of index 0 ~ 15

30.4.6 P_CSI_COLOR_TABLE2(0x88000044)

NAME	D31-D0
P_CSI_COLOR_TABLE2	MD_CTABLE1

MD_CTABLE1: Color Classification table 1, color of index 16 ~ 31

30.4.7 P_CSI_COLOR_TABLE3(0x88000048)

NAME	D31-D0
P_CSI_COLOR_TABLE3	MD_CTABLE2

MD_CTABLE2: Color Classification table 2, color of index 32 ~ 47

30.4.8 P_CSI_COLOR_TABLE4(0x8800004C)

NAME	D31-D0
P_CSI_COLOR_TABLE4	MD_CTABLE3

MD_CTABLE3: Color Classification table 3, color of index 48 ~ 53

30.4.9 P_CSI_DETECT_REGION1(0x88000050)

NAME	D31-D24	D23-D16	D15-D8	D7-D0
P_CSI_DETECT_REGION1	L1	R1	T1	B1

B1: Bottom boundary of region 1, -128~128.

T1: Top boundary of region 1, -128~128.

R1: Right boundary of region 1, -128~128.

L1: Left boundary of region 1, -128~128.

30.4.10 P_CSI_DETECT_REGION2(0x88000054)

NAME	D31-D24	D23-D16	D15-D8	D7-D0
P_CSI_DETECT_REGION2	L2	R2	T2	B2

B2: Bottom boundary of region 2, -128~128.

T2: Top boundary of region 2, -128~128.

R2: Right boundary of region 2, -128~128.

L2: Left boundary of region 2, -128~128.

30.4.11 P_CSI_DETECT_REGION3(0x88000058)

NAME	D31-D24	D23-D16	D15-D8	D7-D0
P_CSI_DETECT_REGION3	L3	R3	T3	B3

B3: Bottom boundary of region 3, -128~128.

T3: Top boundary of region 3, -128~128.

R3: Right boundary of region 3, -128~128.

L3: Left boundary of region 3, -128~128.

30.4.12 P_CSI_THRESHOLD_SETUP(0x8800005C)

NAME	D31-D24	D23-D16	D15-D8	D7-D0
P_CSI_THRESHOLD_SETUP	TH_B	TH_W	TH1	TH2

TH2: Threshold of region 2, 0~191.

TH1: Threshold of region 1, 0~191.

TH_W: Threshold of white color, 0~191.

TH_B: Threshold of black color, 0~191.

After each sample (8x8 or 16x16) is calculated by color classification algorithm, one color index (00, 01, 10 or 11) will be given and kept in sequence from the address set in P_CSI_CLASSIFICATION_SA. Following introduces how the index is decided.

R, G, B are the values calculated after color correction of CSI hardware.

X = B – G

Y = R – G

RGB = (R + G + B) / 4

((L1*2 < X < R1*2) & (T1*2 < Y < B1*2)) & (0 ≦ RGB < TH1)) (1)

((L2*2 < X < R2*2) & (T2*2 < Y < B2*2)) & (TH1 ≦ RGB < TH2)) (2)

((L3*2 < X < R3*2) & (T3*2 < Y < B3*2)) & (TH2 ≦ RGB)) (3)

If one or more of the 3 equations above is TRUE,

RGB < TH_B ➔ Color Index = CC_BLACK(P_CSI_MD_CTRL[[5:4])

RGB > TH_W ➔ Color Index = CC_WHITE(P_CSI_MD_CTRL[[7:6])

If the 3 equations above are all FALSE, color index will be set as corresponding value filled in P_CSI_COLOR_TABLE0~3 by the following look-up table.

Z = 16*(R+G+B)

	Z > 102*G	Z > 68*G	Z > 51*G	Z > 41*G	Z > 34*G	Z > 30*G	Z > 26*G	Z < 26*G
Z > 102*R	0	1	2	3	4	5	6	7
Z > 68*R	8	9	10	11	12	13	14	15
Z > 51*R	16	17	18	19	20	21	22	23
Z > 41*R	24	25	26	27	28	29	30	31
Z > 34*R	32	33	34	35	36	37	38	39
Z > 30*R	39	40	41	42	43	44	45	46
Z > 26*R	45	46	47	48	49	50	51	52
Z < 26*R	50	51	52	53	54	55	56	57

If table index is equal to 54, 55, 56 or 57, color index will be filled with ‘00’.

30.4.13 P_CSI_CAPTURE_COLORVALUE(0x88000074)

NAME	D23-D0
P_CSI_CAPTURE_COLORVALUE	MD_RGB

MD_RGB: The capture color value.

When YUVOUT is 1, the value in this register is YUV888.

When YUVOUT is 0, the value in this register is RGB888.

30.5 Interrupt

There are two registers to control the CSI related interrupts. One is P_CSI_INT_CTRL, which is a mask register for the CSI interrupt. When this register is set to '1', the interrupt signal will be masked; that is, the signal will remain inside of CSI module and the CPU can not receive this signal, therefore it will not enter interrupt service routine.

The other is an interrupt status register, P_CSI_INT_STATUS. When an interrupt event is received, the CPU will query this status register to determine which interrupt event happens, and then enter into the corresponding interrupt service routine. The CPU will clear the interrupt event flag automatically when it reads the status register.

30.5.1 P_CSI_INT_CTRL(0x88000078)

NAME	D6	D5	D4	D3	D2	D1	D0
P_CSI_INT_CTRL	MD_UF_IRQ	FRAME_DIS_IRQ	POS_HIT_IRQ	MD_FRAME_IRQ	FRAME_END_IRQ	TG_CAPACK_IRQ	TG_OF_IRQ

TG_OF_IRQ: TG_OF IRQ enable register.

0: Disable TG_OF IRQ.

1: Enable TG_OF IRQ.

TG_CAPACK_IRQ: TG_CAPACK IRQ enable register.

0: Disable TG_CAPACK IRQ.

1: Enable TG_CAPACK IRQ.

FRAME_END_IRQ: FRAME_END IRQ enable register.

0: Disable FRAME_END IRQ.

1: Enable FRAME_END IRQ.

MD_FRAME_IRQ: MD_FRAME IRQ enable register.

0: Disable MD_FRAME IRQ.

1: Enable MD_FRAME IRQ.

POS_HIT_IRQ: POS_HIT IRQ enable register.

0: Disable POS_HIT IRQ.

1: Enable POS_HIT IRQ.

FRAME_DIS_IRQ: FRAMD_DIS IRQ enable register.

0: Disable FRAMD_DIS IRQ.

1: Enable FRAMD_DIS IRQ.

MD_UF_IRQ: MD_UF IRQ enable register.

0: Disable MD_UF IRQ.

1: Enable MD_UF IRQ.

30.5.2 P_CSI_INT_STATUS(0x8800007C)

NAME	D6	D5	D4	D3	D2	D1	D0
P_CSI_INT_STATUS	MD_UF_STS	FRAME_DIS_STS	POS_HIT_STS	MD_FRAME_STS	FRAME_END_STS	TG_CAPACK_STS	TG_OF_STS

TG_OF_STS: TG_OF IRQ status register. (IRQ48)

Read 0: TG_OF IRQ does not happen.

Read 1: The sensor output data overflow situation happens.

Write 0: No effect.

Write 1: Clear this bit.

TG_CAPACK_STS: TG_CAPACK IRQ status register. (IRQ48)

Read 0: TG_CAPACK IRQ does not happen.

Read 1: The capture procedure is done.

Write 0: No effect.

Write 1: Clear this bit.

FRAME_END_STS: FRAME_END IRQ status register. (IRQ51)

Read 0: FRAME_END IRQ does not happen.

Read 1: A frame is end.

Write 0: No effect.

Write 1: Clear this bit.

MD_FRAME_STS: MD_FRAME IRQ status register. (IRQ49)

Read 0: MD_FRAME IRQ does not happen.

Read 1: A motion detect frame is end.

Write 0: No effect.

Write 1: Clear this bit.

POS_HIT_STS: POS_HIT IRQ status register. (IRQ50)

Read 0: POS_HIT IRQ does not happen.

Read 1: The luster hits the position defined in {MD_HPOS, MD_VPOS}

Write 0: No effect.

Write 1: Clear this bit.

FRAME_DIS_STS: Frame disable interrupt. (IRQ48)

Read 0: FRAME_DIS IRQ does not happen.

Read 1: A frame disable flag is received by CSI module.

Write 0: No effect.

Write 1: Clear this bit.

MD_UF_STS: Motion Detect under Flow interrupt. (IRQ48)

Read 0: MD_UF IRQ does not happen.

Read 1: A motion detect buffer under-run situation happens.

Write 0: No effect.

Write 1: Clear this bit.

30.6 Color Mode

30.6.1 P_CSI_Y2R_FACTOR1(0x880000E8)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_Y2R_FACTOR1	Y2R_A11	Y2R_A12	Y2R_A13

Y2R_A13: A13 parameter of YUV➔RGB transfer

Y2R_A12: A12 parameter of YUV➔RGB transfer

Y2R_A11: A11 parameter of YUV➔RGB transfer

30.6.2 P_CSI_Y2R_FACTOR2(0x880000EC)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_Y2R_FACTOR2	Y2R_A21	Y2R_A22	Y2R_A23

Y2R_A23: A23 parameter of YUV➔RGB transfer

Y2R_A22: A22 parameter of YUV➔RGB transfer

Y2R_A21: A21 parameter of YUV➔RGB transfer

30.6.3 P_CSI_Y2R_FACTOR3(0x880000F0)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_Y2R_FACTOR3	Y2R_A31	Y2R_A32	Y2R_A33

Y2R_A33: A33 parameter of YUV➔RGB transfer

Y2R_A32: A32 parameter of YUV➔RGB transfer

Y2R_A31: A31 parameter of YUV➔RGB transfer

NOTE: The parameters of YUV=> RGB transfer are all 10-bit signed integer, which will be divided by 256 in the hardware. In other words, the values of these parameters are in the following range:

-512/256 <= Y2R_Axy <= 511/256

30.6.4 P_CSI_R2Y_FACTOR1(0x880000E8)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_R2Y_FACTOR1	R2Y_A11	R2Y_A12	R2Y_A13

R2Y_A13: A13 parameter of RGB➔YUV transfer

R2Y_A12: A12 parameter of RGB➔ YUV transfer

R2Y_A11: A11 parameter of RGB➔ YUV transfer

30.6.5 P_CSI_R2Y_FACTOR2(0x880000EC)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_R2Y_FACTOR2	R2Y_A21	R2Y_A22	R2Y_A23

R2Y_A23: A23 parameter of RGB➔YUV transfer

R2Y_A22: A22 parameter of RGB➔ YUV transfer

R2Y_A21: A21 parameter of RGB➔ YUV transfer

30.6.6 P_CSI_R2Y_FACTOR3(0x880000F0)

NAME	D29-D20	D19-D10	D9-D0
P_CSI_R2Y_FACTOR3	R2Y_A31	R2Y_A32	R2Y_A33

R2Y_A33: A33 parameter of RGB➔YUV transfer

R2Y_A32: A32 parameter of RGB➔ YUV transfer

R2Y_A31: A31 parameter of RGB➔ YUV transfer

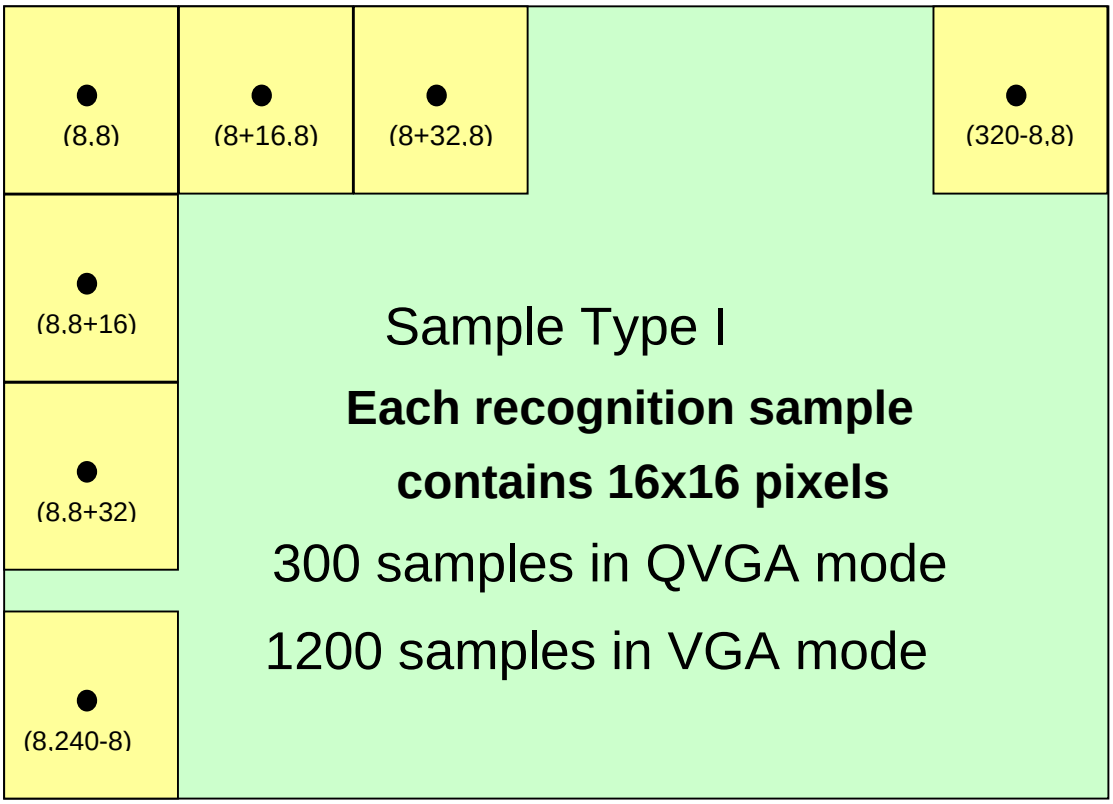
NOTE: The parameters of RGB=> YUV transfer are all 9-bit signed integer, which will be divided by 256 in the hardware. In other words, the values of these parameters are in the following range:

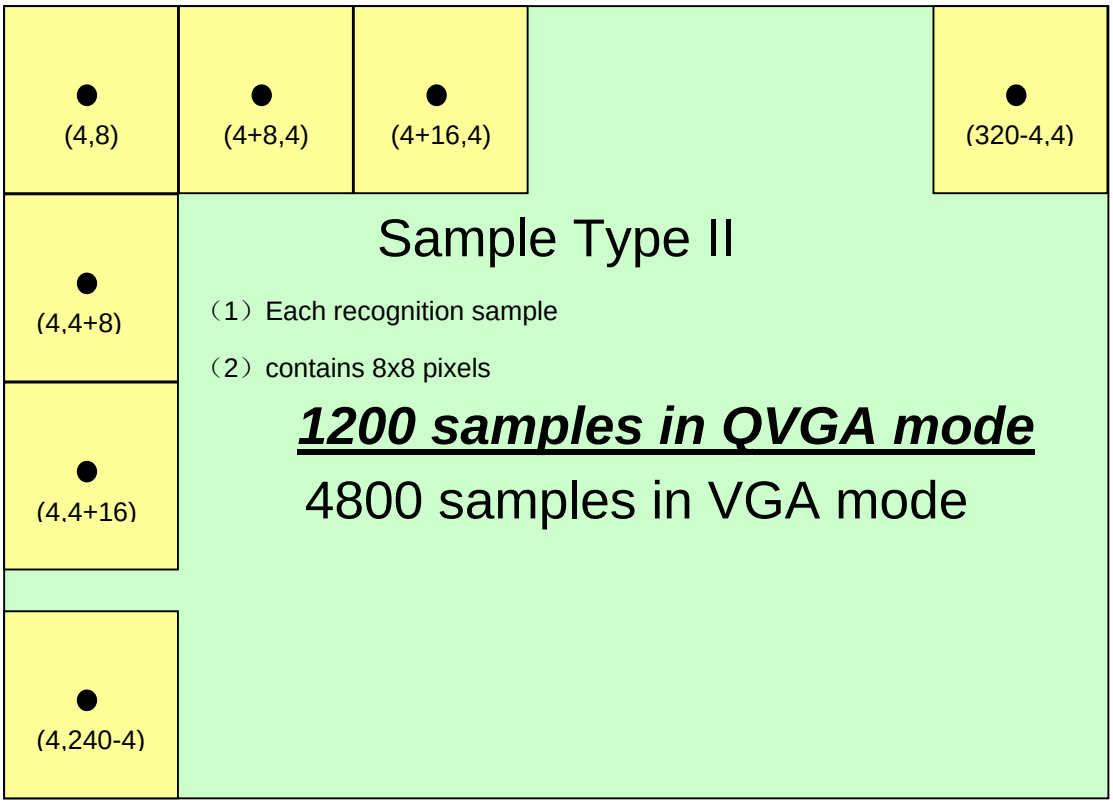
-256/256 <= R2Y_Axy <= 255/256

CSI_Y2R and **CSI_R2Y share the same register**. When YUVIN mode is selected, these registers must be filled by YUV=>RGB parameter. When RGBIN mode is selected, these registers must be filled by RGB=>YUV parameter.

30.7 Motion Detection

30.7.1 Recognition Sample Type





30.7.2 Recognition Reference Filed / Frame

If CPU does not have much time to handle motion detect algorithm, recognition data can not be always calculated at each frame. The following table shows the reference field or frame decided by setting up parameters. For examples, if INTERLACE = 1, RC_FIELD = 0, and FRAME_TYPE = 2, each reference frame is frame2 field0 instead of frame2 field1.

INTERLACE	RC_FIELD	FRAME_TYPE	Field0	Field1	Frame0	Frame1	Frame2	Frame3
0	0/1	0			HIT	HIT	HIT	HIT
0	0/1	1				HIT		HIT
0	0/1	2					HIT	
0	0/1	3						HIT
1	0	0	REF		HIT	HIT	HIT	HIT
1	0	1	REF			HIT		HIT
1	0	2	REF				HIT	
1	0	3	REF					HIT
1	1	0		REF	HIT	HIT	HIT	HIT

INTERLACE	RC_FIELD	FRAME_TYPE	Field0	Field1	Frame0	Frame1	Frame2	Frame3
1	1	1		REF		HIT		HIT
1	1	2		REF			HIT	
1	1	3		REF				HIT

Note: REF means which field is referred.

FRAME_TYPE	FRAME_COUNTER
0	Always 0
1	Always from 0 to 1
2	Always from 0 to 2
3	Always from 0 to 3

31 MPEG-4/JPEG Codec

31.1 Introduction

The hardware-implemented MPEG-4 (Moving Picture Experts Group 4) codec supports the simple profile for decoding and encoding. The image resolution can be QVGA or VGA. When the codec is running at 54MHz, it can achieve 30f/s frame rate for encoding or decoding with QVGA resolution. The quantization step size is programmable to get the high compression ratio or to obtain the good image quality.

The same codec also provides JPEG (Joint Photographic Experts Group) function. It supports both 4:2:2 and 4:2:0 data formats for decoding and encoding. The image resolution can be QVGA or VGA.

31.2 Functional Description

31.2.1 MPEG Encoding

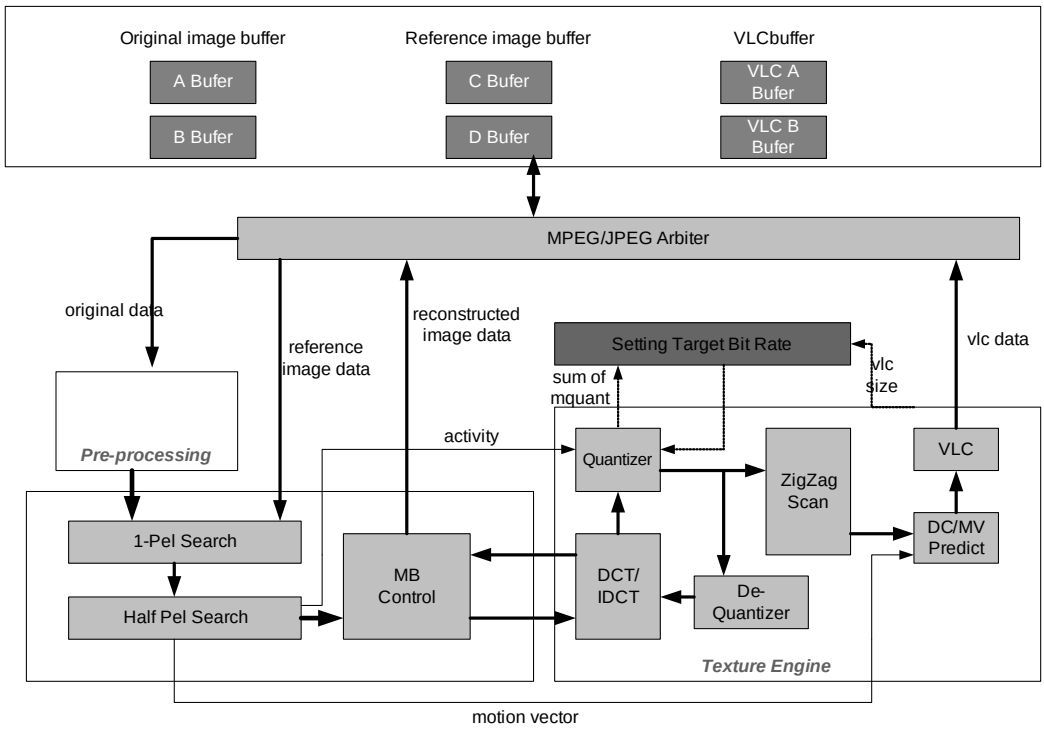


Fig 31-1 MPEG Encoding Principle

When you encode a series of pictures to a video, you must set and control the original image buffer, reference image buffer, and VLC buffer memory start address. A series of video image start addresses must be set in the registers P_MPEG4_RAWBUFFER_SAx

(refer to Chapter 6 MIU), and you can switch these image buffers by setting the register P_MPEG4_RAWBUFFER_SEL (refer to the BUFCTL chapter). It's better to make the image buffer start address 1K aligned. You can adjust the start point of video image by MPEG engine registers "**Mjhoffset**" and "**Mjvoffset**" if necessary.

The MPEG engine needs reference image buffers when it operates MPEG encoding. Two image buffer start addresses can be set in the registers P_MPEG4_WRITEBUFFER_SA1 and P_MPEG4_WRITEBUFFER_SA2 (refer to Chapter 6 MIU). Note that the buffer size and the video image are the same. Of course, the start addresses of reference image buffers are also 1K aligned. The control registers for switching these start addresses are P_MPEG4_WRITEBUFFER_SEL and P_MPEG4_REFBUFFER_SEL (refer to Chapter 7 BUFCTL) and their settings can't be the same. After completing a frame encoding, you must exchange their values.

You can assign where the encoded video code is stored by writing the buffer start address in register P_MPEG4_VLCBUFFER_SAx (refer to Chapter 6 MIU) and using the register P_MPEG4_VLCBUFFER_SEL (refer to Chapter 7 BUFCTL) to switch these buffer start addresses.

You can also set the MPEG engine register P_MPEG4_VLCOFFSET_SAHIGH to help MPEG engine bypass the MPEG file header you create which stores some information about this MPEG4 video.

The MPEG engine will output the MPEG4 video object plane part and you need to add the MPEG4 visual object and video object layer before the codes which MPEG engine generates to complete a MPEG4 video file. After that, you can make it up to a 3gp or avi file and save them to other media if you know the formats of these files.

The next step is to set the video image size. Besides the MPEG engine registers P_MPEG4_WIDTH_HIGH and P_MPEG4_HEIGHT_HIGH, you need set the P_MPEG4_FRAMEBUFFER_HSIZE (refer to Chapter 6 MIU), too.

You must know this MPEG engine doesn't support the B-frame. For the limitation of the MPEG engine, refer to Section 31.4.

The MPEG engine register P_MPEG4_PFRAME_NUMBER facilitate MPEG engine to encode one I-frame after the static number P-frames automatically. If you want to control when to encode I-frame or P-frame, you can use the register P_MPEG4_FRAME_MODE to decide the frame type. You can switch the two choices by setting the register P_MPEG4_MODE_CTRL3. No matter what mode you choose, you must set non-zero value to the MPEG engine registers P_MPEG4_PFRAME_NUMBER. When you generate I-frame or P-frame by the register P_MPEG4_FRAME_MODE, you had better set 2 to P_MPEG4_PFRAME_NUMBER.

The recommend value to the register P_MPEG4_VOPTIMEINC_LENGTH is 14.

Because the MPEG specific doesn't use quantizing table, you had better enable the "**QscaleEn**" bit in P_MPEG4_QTABLE_CTRL register.

By setting these registers P_MPEG4_IFRAME_QSCALE and P_MPEG4_PFRAME_QSCALE, the MPEG engine can generate Quantizer for video frame automatically. The greater of the Qscale, the less of the compressed code and the poorer of the video quality.

Other bits like UNRESTRICT (in P_MPEG4_MODE_CTRL3), HPELEN (in P_MPEG4_MODE_CTRL3) and SKIPTYPE (in P_MPEG4_QTABLE_CTRL) will affect the MPEG engine performance. You can get more information about MPEG4 and H.263 from ISO/IEC 14496-2.

After setting the MPEG related registers, you also need to set the MPEG engine register **SRAM CS_N** bit in P_MP4MPEG4_SRAM_CS_EN to 1 to turn on the internal static ram of MPEG texture engine.

Apart from the setting of switching MPEG engines buffer automatically, you also need to inform the MPEG engine to apply to this. Setting "**SYS_MPG_CAMMODE**" to 3 (see Chapter 13 SFTCFG), "**doubleEn**" to 1, and "**SOF mode**" to 0 makes the MPEG engine start encoding the next frame after the former frame is encoded automatically.

When the MPEG engine encodes MPEG4 video slower than the picture source making, we must force the hardware to bypass some frames to be encoded when encoding frame is automatic. To achieve this, we can set the register P_MPEG4_ENCODE_FRAMENUM (refer to Chapter 7 BUFCTL).

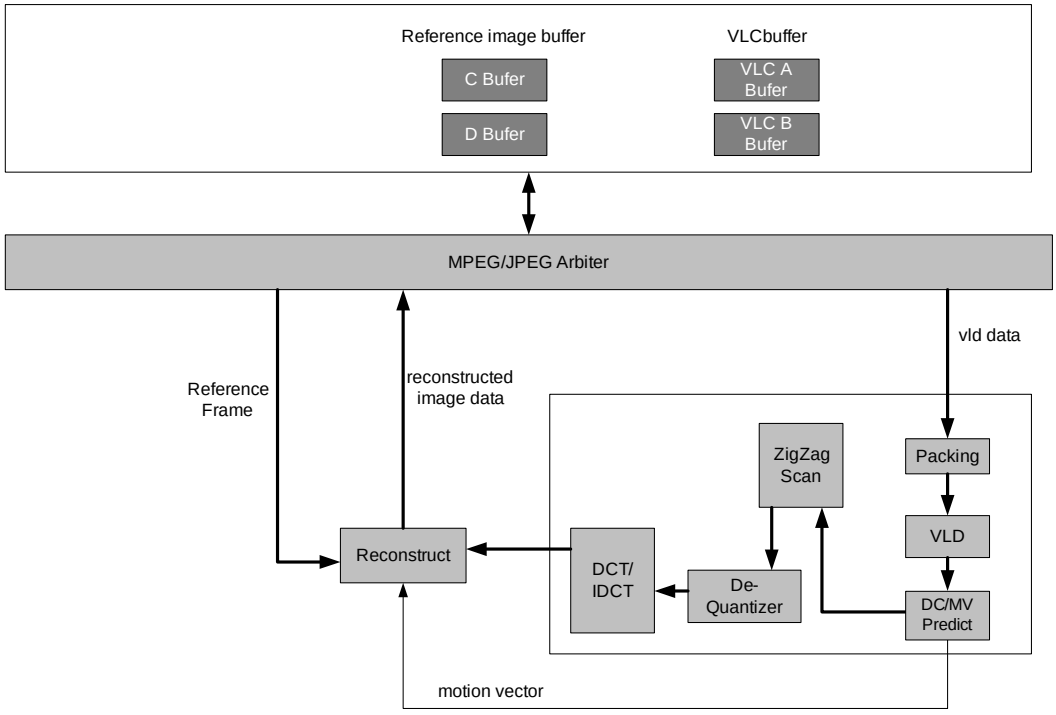
When you are sure that you have filled all registers for MPEG encoding function, the software can start the encoding process by setting "MJPGENC" to 1 in the register P_MPEG4_COMPRESS_CTRL.

You can be acknowledged by polling the bit "EOFSTATUS" in P_MP4_SOFCTRL register or interrupt after the MPEG engine finishes its MPEG encoding work.

The MPEG engine interrupt is 33 (see Chapter 12 Peripheral Interrupt Controller), and its interrupt mask is "EOFEN" bit in P_MPEG4_INT_CTRL register. If the MPEG interrupt occurs, you must clear it by writing the bit EOFINTCLR in P_MPEG4_INT_STATUS register.

The last step is to read data "**VLCSize**" in P_MPEG4_VLCSIZE_x register for getting the MPEG code length.

31.2.2 MPEG Decoding



When you decode a series of pictures from a MPEG video file, you just need to set and control the reference image buffer and VLC buffer memory. The video file format must meet the limitation of the MPEG engine. You can check the items listed as fellow content (See 31.4).

The MPEG engine needs reference image buffers when it operates MPEG decoding. Two image buffer start addresses can be set in the registers P_MPEG4_WRITEBUFFER_SA1 and P_MPEG4_WRITEBUFFER_SA2 (refer to Chapter 6 MIU). Note that the buffer sizes are the same those of the video images that you want to get. No doubt, the start addresses of reference image buffers are 1K aligned. The control registers for switching these start addresses are P_MPEG4_WRITEBUFFER_SEL and P_MPEG4_REFBUFFER_SEL (refer to Chapter 11 BUFCTL) and their settings can't be the same. After completing a frame image decoding, you can get the video image from the start address (P_MPEG4_WRITEBUFFER_SA1 or P_MPEG4_WRITEBUFFER_SA2) to which P_MPEG4_WRITEBUFFER_SEL points and you must put the value of P_MPEG4_WRITEBUFFER_SEL into the register P_MPEG4_REFBUFFER_SEL. After that, the video image newly decoded will be referenced by MPEG engine as it decodes the next P-frame encoded image.

You can assign where the video file is stored by writing the buffer start address in register P_MPEG4_VLCBUFFER_SAx (refer to Chapter 6 MIU) and using the register P_MPEG4_VLCBUFFER_SEL (refer to Chapter 7 BUFCTL) to switch these buffer start addresses.

You can also set the MPEG engine register P_MPEG4_DECVLCOFFSET_SAx to help

MPEG engine bypass the video file header. Note that the value set into register P_MPEG4_DECVLCOFFSET_SAx should not be too big.

The MPEG engine can just decode MPEG4 video object plane part, so you must drop the code before MPEG4 video object plane in a MPEG4 video file no matter the file format is 3gp or avi.

The next step is to set the video image size that can be extracted from the video file header. Besides the MPEG engine registers P_MPEG4_WIDTH_HIGH and P_MPEG4_HEIGHT_HIGH, you also need set the P_MPEG4_FRAMEBUFFER_HSIZE (refer to Chapter 6 MIU), too.

You must enable "UV420TO422" bit and "UV420TO422" bit in the P_MPEG4_YUV_MODE register, because SPCE3200 display interfaces (TVE or LCD) just support YUV422; otherwise the MPEG engine will output YUV420 image.

To inform the MPEG engine that we want to decode a video file, you need set the MPEG4 decode mode using **"Opt_mode"** in the MPEG engine register P_MPEG4_MODE_CTRL1.

If you are decoding MPEG4 visual object plane, you had better set the visual object plane start code 0x000001b6 to the P_MPEG4_MATCH_CODEx(x=0,1,2,3) register.

If you are decoding H.263 video file, you should take care of these bits "**H263en**" , "**H263format**" in the P_MPEG4_H263_CTRL register and "**H263GOBen**" in the P_MPEG4_H263_STRUCTURE register.

For more information about MPEG4 and H.263, refer to ISO/IEC 14496-2 Specification.

Because the MPEG specification doesn't use quantizing table, you had better enable bit "QSCALEEN" in the P_MPEG4_QTABLE_CTRL register.

The "VOPTimeIncLen" value to the register P_MPEG4_VOPTIMEINC_LENGTH, "FOURMVEn", "ACPREDEn" to the register P_MPEG4_H263_CTRL need to be filled by users or the MPEG engine will get the information from MPEG4 visual object plane.

After setting the MPEG related registers, you also need to set "**SRAM_CS_N**" to 1 in the MPEG engine register P_MPEG4_SRAM_EN to turn on the internal static ram of MPEG texture engine.

The software gets the control of video decoding by setting "SOFMODE" to 1 in the register P_MPEG4_MODE_CTRL3.

WHEN YOU NEED TO START THE MPEG ENGINE DECODING, YOU JUST SET "MJPGDSOF" IN THE REGISTER P_MPEG4_COMPRESS_CTRL.

You can be acknowledged by polling MJPGDEC bit in the P_MP4_SOFCTRL register or

interrupt after the MPEG engine finishes its MPEG encoding work.

The MPEG engine interrupt is 33 (see Chapter 12 Peripheral Interrupt Controller), and its interrupt mask is “EOFEN” in the P_MPEG4_INT_CTRL register. If the MPEG interrupt occurs, you must clear it by writing “EOFINTCLR” bit in the register P_MPEG4_INT_STATUS.

31.2.3 JPEG Encoding

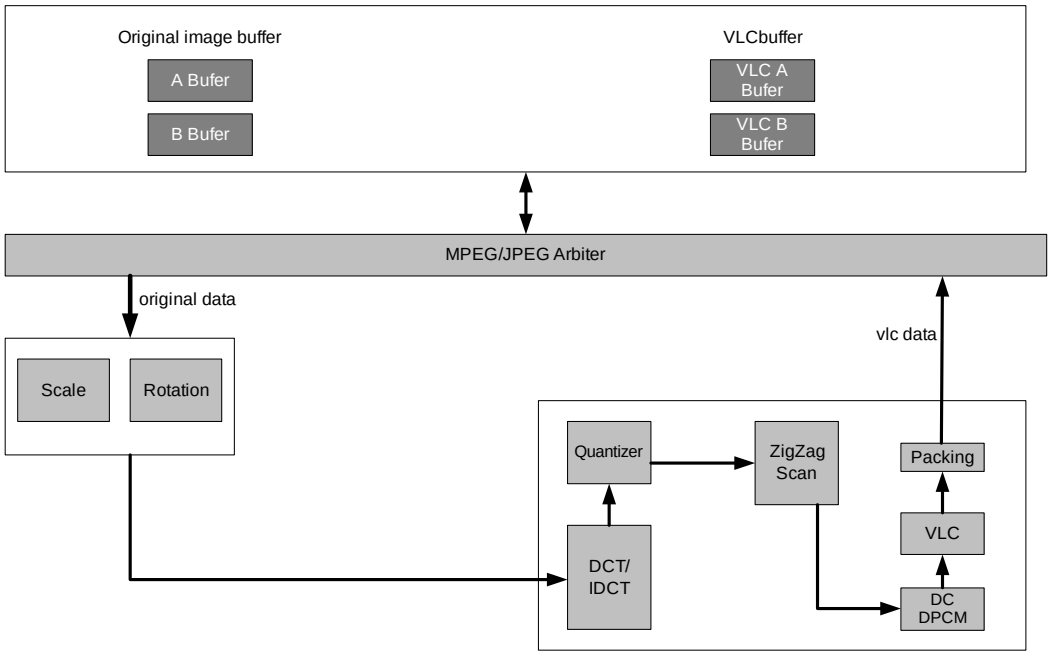


Fig 31-2

When you want to encode JPEG file, set the P_MPEG4_RAWBUFFER_SAx (reference Chapter 6 MIU) firstly which tells the MPEG engine where the original picture stored. MPEG engine registers MP4_MJHOFFSET and MP4_MJVOFFSET let you adjust the start point of original picture if the image memory start address is not just at where P_MPEG4_RAWBUFFER_SAx specified.

Then set P_MPEG4_VLCBUFFER_SAx (reference Chapter 6 MIU) which tell the MPEG engine where you want to store the compressed JPEG file. The start address should be word align. Be careful that MPEG engine will not fill the JPEG file header, you must generate it by yourself. You can set the MPEG engine registers P_MPEG4_VLCOFFSET_SAHIGH which lets MPEG engine bypass the JPEG file header you create.

The next step is setting the original picture size. Beside the MPEG engine registers MP4_MJWIDTH and MP4_MJHEIGHT, you need set the P_MPEG4_FRAMEBUFFER_HSIZE (reference Chapter 6 MIU), too.

To tell the MPEG engine that we want to compress a picture, you need set JPEG encode mode (OPT_MODE=000) in the MPEG engine register P_MPEG4_MODE_CTRL1. And tell the MPEG engine what sample type which your original picture is (YUV422, YUV420, gray-level) in the P_MPEG4_YUV_SEL register and P_MPEG4_MODE_CTRL1 register.

To enable "**JFIF**" bit in the MPEG engine register P_MPEG4_JFIF_COMPATIBLE for ensuring your compressed jpeg file can meet the JPEG File Interchange Format.

No matter what mode you operation, you must set nonzero value to "GOPVA" in the MPEG engine register P_MPEG4_PFRAME_NUMBER. If the MPEG engine is not in MPEG encode mode, you had better setting 2 to "GOPVAL".

I don't suggest that enable the MPEG engine by hardware, unless you very understand the MIU and BUFCTL operation. So I recommend setting "SOFMODE" =1, "DoubleEn" =0 in the MPEG engine registers P_MPEG4_MODE_CTRL3 and P_MPEG4_BUFFER_CTRL respectively.

Before you fill the Quantization Table by which you want to compress the original picture to JPEG format, you should turn on the SRAM which the MPEG engine save the Quantization Table by setting "**SRAM_CS_N**" =01 in the register P_MPEG4_SRAM_EN and "QSRAMEN"=1 in the register P_MPEG4_QTABLE_SRAM. The choice of JPEG file's Quantization Table will affect the JPEG's compression ratio and the quality. You must be careful about the order when filling Quantization Table. The order of Quantization Table saved in JPEG file header is zigzag, and the MPEG engine registers P_MPEG4_QTABLE1 (luminance quantization table) and P_MPEG4_QTABLE2 (chrominance quantization table) are followed by Quantization Table matrix order.

For saving power consumption, I suggest turn off the MPEG move estimation sram of the MPEG engine power by setting **Mesramdis** to 1 in the P_MPEG4_THUMB_BURSTLENGTH register when encoding or decoding JPEG file.

The JPEG encoding of the MPEG engine is using the static Huffman Table, so you don't need fill the Huffman Table.

When you sure that you have filled all registers for JPEG encoding function, you can start the encoding process by setting MJPGENC to 1 in the register P_MPEG4_COMPRESS_CTRL.

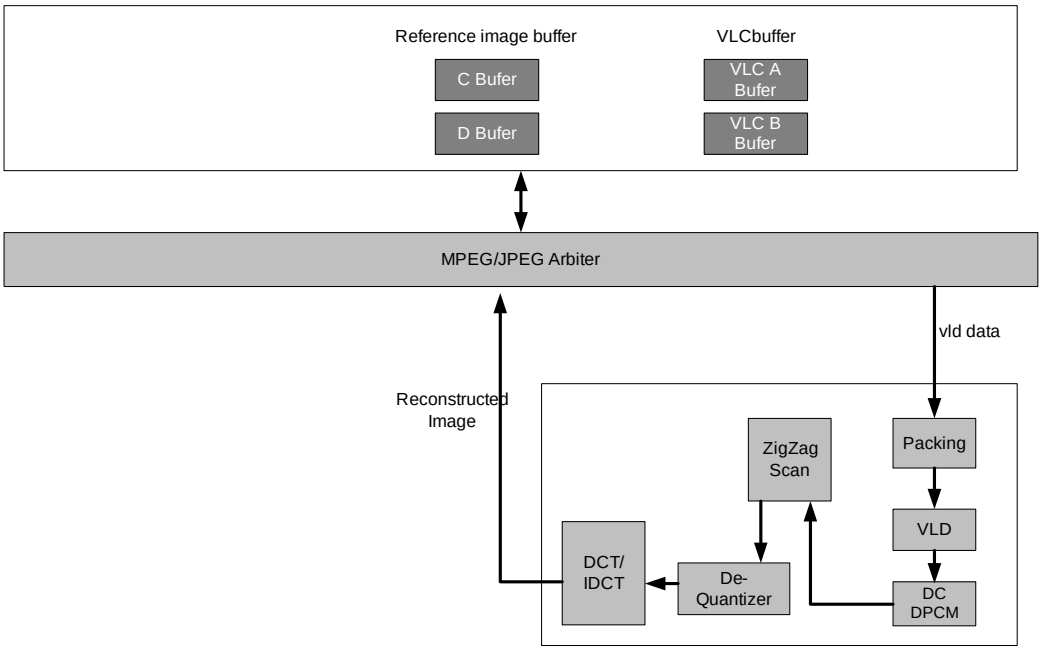
You can be acknowledged by polling EOFSTATUS bit in the register P_MP4_SOFCTRL or interrupt after the MPEG engine finishes its JPEG encoding work.

The MPEG engine interrupt is 33(see Chapter 12 Peripheral Interrupt Controller), and its interrupt mask is "EOFEN" bit in the register P_MPEG4_INT_CTRL. If the MPEG

interrupt occurs, you must clear it by writing EOFINTCLR bit in the register P_MPEG4_INT_STATUS.

The last step is reading "**VLCSize**" in the register MP4_VLCSize for getting the JPEG file length.

31.2.4 JPEG Decoding



If you want to decode JPEG file, you should decode JPEG file header for getting some useful information filled the MPEG engine registers by yourself firstly. You can getting JPEG file's Quantization Table, Huffman Table, image size, YUV sample type … etc from JPEG file header. Before you fill the Quantization Table by which gotten from JPEG file header, you should turn on the SRAM which the MPEG engine save the Quantization Table by setting "**SRAM_CS_N**" =01 and "QSRAMEN"=1 in the MPEG engine registers P_MPEG4_SRAM_EN and P_MPEG4_QTABLE_SRAM respectively.

You must be careful about the order when filling Quantization Table. The order of Quantization Table saved in JPEG file header is zigzag, and the MPEG engine register P_MPEG4_QTABLE 1 (luminance quantization table) and P_MPEG4_QTABLE 2 (chrominance quantization table) are followed by Quantization Table matrix order. After you filled Quantization Table, you should turn off "QSRAMEN" bit in the register P_MPEG4_QTABLE_SRAM.

For saving power consumption, I suggest turn off the MPEG move estimation static ram of the MPEG engine power by setting "**Mesramdis**" =1 in the register P_MPEG4_THUMB_BURSTLENGTH when decoding or decoding JPEG file.

The Huffman Tables stored in JPEG file header are not filled into the MPEG engine

registers directly. You must find the code values, code sizes and code words of Huffman tables to fill these registers P_MPEG4_LUMHUFFMANTABLE, P_MPEG4_LUMOFFSET and P_MPEG4_LUMDCCODE (Luminance DC Huffman Table), P_MPEG4_CHROMHUFFMANTABLE, P_MPEG4_CHROMHUFFMANTABLE and P_MPEG4_CHROMDCCOD (Chrominance DC Huffman Table), P_MPEG4_LUMACHUFFMANOFFSET and P_MPEG4_LUMACHUFFMANTABLE (Luminance AC Huffman Table), P_MPEG4_CHROMACHUFFMANOFFSET and P_MPEG4_CHROMACHUFFMANTABLE (Chrominance AC Huffman Table)and Huffman Table static ram. You can reference the CCITT Recommendation T.81 Annex K to understand how to do this.

Set the P_MPEG4_VLCBUFFER_SAx (reference Chapter 6 MIU) firstly which tells the MPEG engine where the JPEG file stored. MPEG engine register MP4_DVLCOFFADDR let you adjust the start point of JPEG file if you need to bypass the JPEG file header.

Then set P_MPEG4_RAWBUFFER_SAx (reference Chapter 6 MIU) which tell the MPEG engine where you want to store decompressed image.

The next step is setting the original picture size that is the part of JPEG file header. Beside the MPEG engine registers P_MPEG4_WIDTH_HIGH and P_MPEG4_HEIGHT_HIGH , you need set the "P_MPEG4_FRAMEBUFFER_HSIZE" (reference Chapter 6 MIU), too.

To tell the MPEG engine that we want to decompress a JPEG file, you need set JPEG decode mode (OPT_MODE=001) in the MPEG engine register P_MPEG4_MODE_CTRL1. And tell the MPEG engine what sample type which your original picture is (YUV422, YUV420, gray-level). Please remember set " UV420TO422" and "UV420TO422EN" in the register P_MPEG4_YUV_MODE.

To enable "**JFIF**" bit in the register P_MPEG4_JFIF_COMPATIBLE if the JPEG file meets the JPEG File Interchange Format.

No matter what mode you operation, you must set nonzero value to GOPVAL" in the registers P_MPEG4_PFRAME_NUMBER. If the MPEG engine is not in MPEG encode mode, you had better setting 2 to "GOPVAL" .

I don' t suggest that start the MPEG engine operation by hardware, unless you very understand the operation between MPEG engine, MIU and BUFCTL. So I recommend setting "**SOF mode**" =1, "**doubleEn**" =0 in the MPEG engine registers P_MPEG4_MODE_CTRL3 and P_MPEG4_BUFFER_CTRL severally.

When you sure that you have filled all registers for JPEG decoding function, you can start the encoding process by setting "MJPGDEC" =1 in the MPEG engine register P_MPEG4_COMPRESS_CTRL.

You can be acknowledged by polling the register "MJPGDEC" in the register P_MP4_SOFCTRL or interrupt after the MPEG engine finishes its JPEG decoding work.

The MPEG engine interrupt is 33(see Chapter 12 Peripheral Interrupt Controller), and its interrupt mask setting is "EOFEN" bit in register P_MPEG4_INT_CTRL. If the MPEG interrupt occurs, you must clear it by writing "EOFINTCLR" bit in the register P_MPEG4_INT_STATUS.

You can decode a 640x480 JPEG file to 320x240 picture by setting PBRSCAL [2:0]=4 in the register P_MPEG4_MODE_CTRL2.

I have not tried the rotate function of this MPEG engine and its bits "ROTCHG" , "ROTXDN" and "ROTYDN" in the register P_MPEG4_MODE_CTRL2.

I have not tried the Region of Interest function of this MPEG engine and its "**ROIEn**" in the register P_MPEG4_GATE_CLK.

I have not tried the Vertical scale factor register P_MPEG4_VEL_SCALE and Horizontal scale factor register P_MPEG4_HOR_SCALE.

I have not tried the JPEG Thumbnail function of this MPEG engine and its set of registers P_MPEG4_THUMB_GENERATION, P_MPEG4_THUMB_BURSTLENGTH and P_MPEG4_THUMB_SAHIGH.

31.2.5 Coding Note

If AHB bus frequency is equal to MPEG Engine frequency, set P2M_SETTING (0x8820_00c0) =6, please.

If AHB bus frequency is double MPEG Engine frequency, set P2M_SETTING (0x8820_00c0) =10, please (see Chapter 13 SFTCFG).

When set the start address of image or VLC buffer memory start address, please keep the start address 1K boundaries.

MPEG engine only deal with the 4Y4U4Y4V pixel format.

If you want to generate the RGB565 pixel format from MPEG 4Y4U4Y4V, you can reference the Chapter 15 BLNDMA YUV2RGB conversion.

31.3 MPEG Engine Control Register Definition

31.3.1 P_MPEG4_WIDTH_LOW(0x88220000): Image Width Low Byte Control

NAME	D7-D0
P_MPEG4_WIDTH_LOW	MJWIDTHL

MJWIDTHL: Image Width, low byte

It must be the multiple of 16. If rotation or scaling occurs, this value will be image width after rotation and scaling.

31.3.2 P_MPEG4_WIDTH_HIGH(0x88220004): Image Width High Byte Control

NAME	D11-D8
P_MPEG4_WIDTH_HIGH	MJWIDTH

MJWIDTH: Image Width, high byte.

31.3.3 P_MPEG4_HEIGHT_LOW(0x88220008): Image Height Low byte Control

NAME	D7-D0
P_MPEG4_HEIGHT_LOW	MJHEIGHTL

MJHEIGHTL: Image Height, low byte

It must be the multiple of 16. If rotation or scaling occurs, this value will be image width after rotation and scaling.

31.3.4 P_MPEG4_HEIGHT_HIGH(0x8822000C): Image Height High byte Control

NAME	D11-D8
P_MPEG4_HEIGHT_HIGH	MJHEIGHT

MJHEIGHT: Image Height, high byte.

31.3.5 P_MPEG4_HOROFFSET_LOW(0x88220010): Image Horizontal Offset Low Byte Control

NAME	D7-D0
P_MPEG4_HOROFFSET_LOW	MJHOFFSETL

MJHOFFSETL: Image Horizontal offset, low byte, byte boundary.

31.3.6 P_MPEG4_HOROFFSET_HIGH(0x88220014): Image Horizontal Offset Control

NAME	D11-D8
P_MPEG4_HOROFFSET_HIGH	MJHOFFSET

MJHOFFSET: Image Horizontal offset, high byte.

31.3.7 P_MPEG4_VEROFFSET_LOW(0x88220018): Image Vertical Offset Low Byte Control

NAME	D7-D0
P_MPEG4_VEROFFSET_LOW	MJVOFFSETL

MJVOFFSETL: Image Vertical offset, low byte, byte boundary.

31.3.8 P_MPEG4_VEROFFSET_HIGH(0x8822001C): Image Vertical Offset Control

NAME	D11-D8
P_MPEG4_VEROFFSET_HIGH	MJVOFFSET

MJVOFFSET: Image Vertical offset, high byte.

31.3.9 P_MPEG4_VLCOFFSET_SALOW(0x88220020): VLC Bit Stream Starting Offset Address Low Byte

NAME	D7-D0
P_MPEG4_VLCOFFSET_SALOW	VLCOFFADDRL

VLCOFFADDRL: VLC Bit Stream Starting Offset Address, low byte, byte boundary.

31.3.10 P_MPEG4_VLCOFFSET_SAMID(0x88220024):VLC Bit Stream Starting Offset Address Middle Byte

NAME	D15-D8
P_MPEG4_VLCOFFSET_SAMID	VLCOFFADDRM

VLCOFFADDRM: VLC Bit Stream Starting Offset Address, middle byte.

31.3.11 P_MPEG4_VLCOFFSET_SAHIGH(0x88220028):VLC Bit Stream Starting Offset Address High Byte

NAME	D23-D16
P_MPEG4_VLCOFFSET_SAHIGH	VLCOFFADDR

VLCOFFADDR: VLC Bit Stream Starting Offset Address, high byte.

31.3.12 P_MPEG4_THUMB_SALOW(0x88220040): Thumb Nail Starting Address Low Byte

NAME	D7-D0
P_MPEG4_THUMB_SALOW	TMBOFFADDRL

TMBOFFADDRL: Thumb nail starting address, low byte, word boundary, bit 0 is ignored.

31.3.13 P_MPEG4_THUMB_SAMID(0x88220044): Thumb Nail Starting Address Middle Byte

NAME	D15-D8
P_MPEG4_THUMB_SAMID	TMBOFFADDRM

TMBOFFADDRM: Thumb nail starting address, middle byte.

31.3.14 P_MPEG4_THUMB_SAHIGH(0x88220048): Thumb Nail Starting Address High Byte

NAME	D23-D16
P_MPEG4_THUMB_SAHIGH	TMBOFFADDR

TMBOFFADDR: Thumb nail starting address, high byte.

31.3.15 P_MPEG4_YUV_SEL(0x8822004C): YUV Group Selection

NAME	D0
P_MPEG4_YUV_SEL	YUV_TYPE

YUV_TYPE: YUV Group Selection

0: YUV420/YUV422

1: YUV444/YUV411

31.3.16 P_MPEG4_MODE_CTRL1(0x88220050): MJPG Control

NAME	D7	D6	D5	D4	D3	D2-D0
P_MPEG4_MODE_CTRL1	FULLRANGEUV	FULLRANGEY	JPG_BW	BW_MODE	YUV_SEL	OPT_MODE

OPT_MODE: MJPG Operation Mode Selection

000: JPEG encode mode

001: JPEG decode mode

010: MPEG 1 encode mode

011: MPEG 1 decode mode

110: MPEG 4 encode mode

111: MPEG 4 decode mode

YUV_SEL: YUV Mode Select Base Mode Setting depends on YUV Group Selection (0x8822004C)

If P_MPEG4_YUV_SEL is 0

0: YUV420 mode

1: YUV422 mode

If P_MPEG4_YUV_SEL is 1

0: YUV411 mode

1: YUV444 mode

BW_MODE: Black/White mode for JPG and MPEG

0: Color Mode

1: Black/White mode

Note: For MPEG, UV is filled with zero.

For JPEG, B/W mode depends on 0x88220050 [5].

JPG_BW: JPEG Black/White mode selection

0: JPEG Black/White standard mode

1: UV filled with zero

FULLRANGEY: YUV full range selection for Y

0: Full range

Y'= 0~255

1: Not Full Range

Y' = 16~235

FULLRANGEUV: YUV Full Range Selection for UV

0: Full Range

U'=U

V'=V

1: Not Full Range

U'=(U*224)/256

V'=(V*224)/256

31.3.17 P_MPEG4_MODE_CTRL2(0x88220054): MJPG Setting

NAME	D7	D6	D5	D4	D3	D2-D0
P_MPEG4_MODE_CTRL2	RANGESAT	ROTYDN	ROTXDN	ROTCHG		PBRSCAL

PBRSCAL: Play Back Rescale Factor (for JPEG decompression only)

000: 8/8

001: 1/8

010: 2/8

011: 3/8

100: 4/8

101: 5/8

110: 6/8

111: 7/8

ROTCHG: X/Y change, it is not valid for YUV422 image.

If this bit is 1, image width and height (defined at 0x88220000~0x8822000C) are the width and height after rotating.

ROTXDN: X mirror

ROTYDN: Y mirror

NOTE: Rotchg, rotxdn, rotydn control direction of original image. They are only used at compression mode.

	ROTCHG	ROTXDN	ROTYDN
normal	0	0	0
Rotate 90'	1	0	1
Rotate 180'	0	1	1
Rotate 270'	1	1	0
Horizontal Mirror	0	1	0
Vertical Mirror	0	0	1
HOR. / VER. Mirror	0	1	1

RANGESAT: Saturate input YUV data >= 250 to 250

31.3.18 P_MPEG4_HOR_SCALE(0x88220058): Horizontal Scale Factor

NAME	D7-D0
P_MPEG4_HOR_SCALE	HSF

HSF: Horizontal scale factor for read original image. It is only used at compression mode for image scaling up. Scaling method is linear interpolation.

0: No scale

others: Hsf[7:0] = (original width –1)/ (output width –1) * 256

31.3.19 P_MPEG4_VEL_SCALE(0x8822005C): Vertical Scale Factor

NAME	D7-D0
P_MPEG4_VEL_SCALE	VSF

VSF: Vertical scale factor for read original image. It is only used at compression mode for image scaling up. Scaling method is linear interpolation.

0: No scale

others: Vsf[7:0] = (original height –1)/ (output height –1) * 256

31.3.20 P_MPEG4_PFRAME_NUMBER(0x88220060): P Frame number for MPEG Compress Mode Selection

NAME	D7-D0
P_MPEG4_PFRAME_NUMBER	GOPVAL

GOPVAL: P frame number for MPEG compression mode

0: I only

1: IPIP….

2 IPPIPP…

…

31.3.21 P_MPEG4_MODE_CTRL3(0x88220064): Setting Enable

NAME	D6	D5	D4	D3	D2	D1	D0
P_MPEG4_MODE_CTRL3	GOPSETTING	UNRESTRICT	SOFMODE	RCINISET	BR_CTRLEN	HPELEN	GOPRSTt

GOPRST: Reset GOP Counter

0: No Reset, GOP counter will auto back to 0 if it counts to GOP value (REG 0x88220060)

1: H/W GOP counter is reset to zero.

HPELEN: Half Pel Enable

0: Disable

1: Enable

BR_CTRLEN: H/W Bit Rate Control Enable

0: Disable

1: Enable

RCINISET: Rate Control Initial Setting

0: Not Set

1: Set

SOFMODE: SOF Mode

0: Triggered by HW at video clip and pc cam mode

1: Always triggered by CPU

UNRESTRICT: Unrestricted MC Enable. It is only used at MPEG4 Encoding, and this function will prolong some search time.

0: Disable

1: Enable

GOPSETTING: I/P Frame Decision Depending

0: I/P frame decision depends on H/W GOP counter

1: I/P frame decision depends on bit 0 of register P_MPEG4_FRAME_MODE (0x88220088)

31.3.22 P_MPEG4_VARTHRESHOLE_LOW(0x88220068): Variance Threshold LSB Byte

NAME	D7-D0
P_MPEG4_VARTHRESHOLE_LOW	VARDTHRL

VARDTHRL: Variance threshold for inter/intra type decision, LSB byte

31.3.23 P_MPEG4_VARTHRESHOLE_HIHG(0x8822006C): Variance Threshold MSB Byte

NAME	D15-D8
P_MPEG4_VARTHRESHOLE_HIHG	VARDTHRM

VARDTHRM: Variance threshold for inter/intra type decision, MSB byte

31.3.24 P_MPEG4_AAV_INIT (0x88220070): Initial Average Activity Value

NAME	D7-D0
P_MPEG4_AAV_INIT	INIAVGACT

INIAVGACT: Initial average activity value

Integer part is 2 bits.

31.3.25 P_MPEG4_AAV_WEIGHTLOW(0x88220078): Normalize Average Activity Value Weighting LSB Byte

NAME	D7-D0
P_MPEG4_AAV_WEIGHTLOW	AVGACTWEIL

AVGACTWEIL: Normalize average activity value weighting, LSB byte.

The center value is (IMAGE_WIDTH/16)* (IMAGE_HEIGHT/16).

If we decrease this value, the weight of normalize average will increase

31.3.26 P_MPEG4_AAV_WEIGHTHIGH(0x8822007C):Normalize Average Activity Value Weight MSB Byte

NAME	D15-D8
P_MPEG4_AAV_WEIGHTHIGH	AVGACTWEIM

AVGACTWEIM: Normalize average activity value weighting, MSB byte.

The center value is (IMAGE_WIDTH/16)* (IMAGE_HEIGHT/16).

If we decrease this value, the weight of normalize average will increase.

31.3.27 P_MPEG4_SOFT_RESET(0x88220080):Software Reset for MJPG

NAME	D0
P_MPEG4_SOFT_RESET	MJPGRST

MJPGRST: Soft ware reset for MJPG. This SW reset will clear all counters and state machines at MJPG, but register setting will not reset.

0: Disable

1: Enable

31.3.28 P_MPEG4_BUFFER_CTRL(0x88220084): MJPG Buffer Control

NAME	D7	D6-D5	D4	D3-D2	D1	D0
P_MPEG4_BUFFER_CTRL	VCMODE		ZEROMV		DoubleMode	DoubleEN

DoubleEN: Double Buffer Control

0: Disable

1: Enable

DoubleMode: Double Buffer Update Mode

0: From SOF

1: From Static DRAM control

ZEROMV: Fore Zero Motion Vector

0: Disable

1: Enable

VCMODE: Video Conference mode

31.3.29 P_MPEG4_FRAME_MODE(0x88220088): Frame Mode

NAME	D0
P_MPEG4_FRAME_MODE	IFRAME

IFRAME: Frame Mode Selection

0: P Frame

1: I Frame

31.3.30 P_MPEG4_THUMB_BURSTLENGTH(0x8822008C): Thumb Nail Burst Length

NAME	D4	D3	D2-D0
P_MPEG4_THUMB_BURSTLENGTH	Thumb422Mode	Mesramdis	ThumbLength

ThumbLength: Thumb nail burst length

Mesramdis: ME SRAM Disable

Thumb422Mode: For YUV422 image, Thumb nail mode selection

0: No mapping

1: Mapped to YYYYYYYYUUVV data

31.3.31 P_MPEG4_SRAM_CTRL(0x88220090): SRAM Setting

NAME	D6	D5-D4	D3	D2-D0
P_MPEG4_SRAM_CTRL	MCSRAMTEST	MCSRAMSEL	MESRAMTEST	MESRAMSel

MESRAMSel: CPU direct write me static ram selection. MC static ram is mapped to 0x88220000~0x882200ff

MESRAMTEST: CPU Direct Write ME SRAM

0: Disable

1: Enable

MCSRAMSEL: CPU direct write MC Static RAM selection. ME static ram is mapped to 0x88220000~0x882200ff

MCSRAMTEST: CPU Direct Write MC Static RAM

0: Disable

1: Enable

31.3.32 P_MPEG4_PROBE_MODE(0x88220094): H/W Probe Mode

NAME	D7-D0
P_MPEG4_PROBE_MODE	PROBEMODE

31.3.33 P_MPEG4_MEMORY_CTRL (0x88220098): Memory Control

NAME	D3	D2	D1-D0
P_MPEG4_MEMORY_CTRL	MemGroupSel	BurstMode	

BurstMode: Burst Mode

0: Disable

1: Enable

MemGroupSel: Burst Group Selector

0: Disable

1: Enable

31.3.34 P_MPEG4_COMPRESS_CTRL(0x882200A0): Start of Compressing Control

NAME	D4	D3	D2	D1	D0
P_MPEG4_COMPRESS_CTRL	MJPGDEC	SOFINFO	SKIPFRAME	EOFSTATUS	MJPGENC

MJPGENC: MJPG Encode Control

Read: When you write this bit, this bit will be 1. It will be set to 0 when this image is done.

Write: At video clip mode, Starting of Compressing one image is trigged by CDSP. At other mode, Starting of Compressing/Decompressing one image is triggered by writing 1 to this bit.

EOFSTATUS: Writing 1 to bit 0 will reset it to zero. It will be set to 1 by Hw when EOF occurs.

SKIPFRAME: Skip frame status

0: No skip

1: Skip frame

SOFINFO: CPU Starting of Compressing information

MJPGDEC: MJPG Decode Control

Read: It will be set to 0 when this image is done

Write: At Video clip mode, Starting of Compressing one image is trigged by CDSP. At other mode, Starting of Compressing / Decompressing one image is triggered by writing 1 to this bit.

31.3.35 P_MPEG4_PFRAME_MONITER(0x882200A4): GOP Counter Monitor

NAME	D3-D0
P_MPEG4_PFRAME_MONITER	IFRAME

IFRAME: GOP Counter Monitor

0: I Frame

Others: P Frame

31.3.36 P_MPEG4_AAV_PERFRAME(0x882200A8): Normalize Average Activity Value for Every Frame

NAME	D7-D0
P_MPEG4_AAV_PERFRAME	AVGACT

AVGACT: Normalize average activity value for every frame. It is similar to TM5 model. The difference is that we use the sum of absolute difference, not sum of square difference. Integer part is 2 bits.

31.3.37 P_MPEG4_ADV_VALUE(0x882200AC): Average Difference Variance

NAME	D7-D0
P_MPEG4_ADV_VALUE	AVGVARD

31.3.38 P_MPEG4_HORINIT_SCALE(0x882200C0): Horizontal Initial Zoom Ratio

NAME	D7-D0
P_MPEG4_HORINIT_SCALE	HSFINI

31.3.39 P_MPEG4_VERINIT_SCALE(0x882200C4): Vertical Initial Zoom Ratio

NAME	D7-D0
P_MPEG4_VERINIT_SCALE	VSFINI

31.3.40 P_MPEG4_YUV_MODE(0x882200C8): MC Write UV Transfer Mode

NAME	D5	D4	D3	D2-D0
P_MPEG4_YUV_MODE	UV420TO422EN	UV420T422	MCWRMODE	

MCWRMODE: MC Write Mode

UV420TO422: 1: UV420 to 422 conversion mode

0: Compression mode

UV420TO422EN: Enable for UV420 to 422

31.3.41 MPEG4_INT_CTRL(0x882200D0): Interrupt Mask

NAME	D5	D4	D3	D2	D1	D0
MPEG4_INT_CTRL	VLDLAST	DECERR	SOFEN	EOFEN	JPGEN	MCINTEN

MCINTEN: Interrupt Mask for MC

JPGEN: Interrupt Mask for JPEG

EOFEN: Interrupt Mask for EOF

SOFEN: Interrupt Mask for SOF

DECERR: Interrupt Mask for Decoder Error

VLDLAST: Interrupt Mask for VLD Stream Decompress

31.3.42 P_MPEG4_INT_STATUS(0x882200D4): Interrupt Clear

NAME	D5	D4	D3	D2	D1	D0
P_MPEG4_INT_STATUS	VLDRESTART	DECERRCLR	SOFINTCLR	EOFINTCLR		MCINTCLR

MCINTCLR: MC Interrupt Status

Read 0: MC Interrupt Status

Write 1: Clear the Flag

EOFNTCLR: EOF Interrupt Status

Read 0: EOF Interrupt Status

Write 1: Clear the Flag

SOFINTCLR: SOF Interrupt Status

Read 0: SOF Interrupt Status

Write 1: Clear the Flag

DECERRCLR: Decode Error Interrupt Status

Read 0: Decode Error Interrupt Status

Write 1: Clear the Flag

VLDRESTART: VLD Restart Interrupt Status

Read 0: VLD Restart Interrupt Status

Write 1: Clear the Flag

31.3.43 P_MPEG4_ARBITERGATE_CLK(0x882200D8): Disable for Gated Clock

NAME	D1	D0
P_MPEG4_ARBITERGATE_CLK	ENMDRST	DISGATED

DISGATED: Disable for Gated Clock

0: Disable

1: Enable

ENMDRST: Enable for Mode Reset

0: Disable

1: Enable

31.3.44 P_MPEG4_BUFFERx_x: MPEG4/JPEG VLC Buffer Capability

NAME	Address	D7-D0
P_MPEG4_BUFFERA_LOW	0x882200DC	VLCBUFAL
NAME	Address	D15-D8
P_MPEG4_BUFFERA_MID	0x882200E0	VLCBUFAM
NAME	Address	D23-D16
P_MPEG4_BUFFERA_HIGH	0x882200E4	VLCBUFA

NAME	Address	D7-D0
P_MPEG4_BUFFERB_LOW	0x882200E8	VLCBUFBL
NAME	Address	D15-D8
P_MPEG4_BUFFERB_MID	0x882200EC	VLCBUFBM
NAME	Address	D23-D16
P_MPEG4_BUFFERB_HIGH	0x882200F0	VLCBUFB

VLCBUFXL: VLC A/B Buffer Size low byte

VLCBUFXM: VLC A/B Buffer Size Middle Byte

VLCBUFX: VLC A/B Buffer Size High Byte

31.3.45 P_MPEG4_DECWIDTH_LOW(0x88220100): Decode Image Width Low Byte Control

NAME	D7-D0
P_MPEG4_DECWIDTH_LOW	DECWIDTHL

DECWIDTHL: Decode Image width, low byte

It must be the multiple of 16. If rotation or scaling occurs, this value will be image width after rotation and scaling.

31.3.46 P_MPEG4_DECWIDTH_HIGH(0x88220104): Decode Image Width High Byte Control

NAME	D11-D8
P_MPEG4_DECWIDTH_HIGH	DECWIDTH

DECWIDTH: Decode Image Width, high byte

31.3.47 P_MPEG4_DECHEIGHT_LOW(0x88220108): Decode Image Height Low byte Control

NAME	D7-D0
P_MPEG4_DECHEIGHT_LOW	DECHEIGHT

DECHEIGHT: Decode Image height, low byte

It must be multiple of 16. If rotation or scaling occurs, this value will be image height after rotation and scaling.

31.3.48 P_MPEG4_DECHEIGHT_HIGH(0x8822010C): Decode Image Height High byte Control

NAME	D11-D8
P_MPEG4_DECHEIGHT_HIGH	DECHEIGHT

DECHEIGHT: Decode Image Height, high byte.

31.3.49 P_MPEG4_DECVLCOFFSET_Sax: Decode VLC Bitstream Start Offset Address

NAME	Address	D7-D0
P_MPEG4_DECVLCOFFSET_SALOW	0x88220110	DVLCOFFADDRL
NAME	Address	D15-D8
P_MPEG4_DECVLCOFFSET_SAMID	0x88220114	DVLCOFFADDRM
NAME	Address	D23-D16
P_MPEG4_DECVLCOFFSET_SAHIGH	0x88220118	DVLCOFFADDR

Note: Decode VLC Bit Stream Starting Offset, Low/Middle/High Byte, byte boundary.

31.3.50 P_MPEG4_LASTADDR_x: Last Memory Address

NAME	Address	D7-D0
P_MPEG4_LASTADDR_LOW	0x8822011C	LASTMEMADDRL
NAME	Address	D15-D8
P_MPEG4_LASTADDR_MID	0x88220120	LASTMEMADDRM
NAME	Address	D23-D16
P_MPEG4_LASTADDR_HIGH	0x88220124	LASTMEMADDR

Note: Last Memory Address (word alignment),

LastMemAddr = ((Size + VLCOFFADDR) – (LastLength + 1))

31.3.51 P_MPEG4_LASTMEMORY_LENGTH(0x88220128): Last Memory Length

NAME	D3-D0
P_MPEG4_LASTMEMORY_LENGTH	LastLength

LastLength: Last Memory Length

LastLength = (Size + VLCOFFADDR – 1) %16

31.4 Texture Engine Control Register Definition

31.4.1 Quantization Table 1

NAME	Address
P_MPEG4_QTABLE1_START	0x88220400
P_MPEG4_QTABLE1_END	0x882204FC

JPEG: Quantization Table for Y-Component

MPGE4: Invalid

The Values of Q are filled in the raster-scan order.

31.4.2 Quantization Table 2

NAME	Address
P_MPEG4_QTABLE2_START	0x88220500
P_MPEG4_QTABLE2_END	0x882205FC

JPEG: Quantization Table for C-Component

MPGE4: Invalid

The Values of Q are filled in the raster-scan order.

31.4.3 P_MPEG4_QTABLE_INDEX(0x88220600): Q-Table Currently Used

NAME	D7-D0
P_MPEG4_QTABLE_INDEX	NOQTBL

NOQTBL: Index of the Q-table currently used, and this register is for internal check only.

31.4.4 P_MPEG4_QTABLE_CTRL(0x88220604): Q-Table Setting

NAME	D3	D2	D1	D0
P_MPEG4_QTABLE_CTRL	EXHEADEN	SKIPTYPE	QSCALEEN	QVALUE

QVALUE: 0: Original

1: Quantizer Coefficients are set to one

QSCALEEN: Enable qscale function.

0: Use QTable1 & QTable2 as the quantize step size.

1: Hardwire quantize step size.

SKIPTYPE: Used for MPEG skip MCU algorithm selection

0: Skip MCU if the luminance blocks are zero.

1: Skip MCU if all the blocks are zero.

EXHEADEN: Extra Information is appended before MPEG4 bit stream

0: Disable

1: Enable

The 0th byte is the number of skip frames, and the 1st to 3rd bytes are the frame time information.

31.4.5 P_MPEG4_QTABLE_SRAM(0x88220608): Q-Table SRAM Control

NAME	D0
P_MPEG4_QTABLE_SRAM	QSRAMEN

QSRAMEN: Q-Table SRAM Control

0: CPU cannot access Q-Table SRAM

1: Enable CPU to access Q-Table SRAM

31.4.6 P_MPEG4_THUMB_GENERATION (0x8822060C): Thumb Nail Image Generation Control

NAME	D1	D0
P_MPEG4_THUMB_GENERATION	STOPEN	ENTHUMB

ENTHUMB: 0: Disable Thumb Nail Image Generation

1: Enable Thumb Nail Image Generation

STOPEN: 0: Normal operation.

1: Set this bit to enable stop mode. The decompression engine stops for every block until the flag “blockend” is cleared.

31.4.7 P_MPEG4_JFIF_COMPATIBLE (0x88220610): JFIF Compatible Control

NAME	D0
P_MPEG4_JFIF_COMPATIBLE	JFIF

JFIF: JFIF Compatible Control bit.

0: Original

1: Compatible with JFIF bit stream

In a standard JFIF bit stream, 0XFF is used as a marker to indicate the start point of a special control sequence. To represent VLC data 0XFF, 0X00 must be appended to distinguish from the marker.

31.4.8 P_MPEG4_TRUNCATE_CTRL(0x88220614): Handling Selection after Quantization

NAME	D1	D0
P_MPEG4_TRUNCATE_CTRL	DCTresolution	Truncated

Truncated: 0: Rounded after quantization

1: Truncated after quantization

DCTresolution: 0: 9-bit decimal for intra macroblock, and 8-bit decimal for non-intra macroblock.

1: 2-bit decimal for intra macroblock, and 1-bit decimal for non-intra macroblock.

31.4.9 P_MPEG4_VLC_BIT (0x88220618): VLC Bitstream Data Stuffing Bit Information

NAME	D2-D0
P_MPEG4_VLC_BIT	VLCBIT

VLCBIT: Indicates the number of stuffing bits in the last byte of the VLC stream.

0: Non-stuffing 1's

1: 1 stuffing 1's

……

7: 7 stuffing 1's

This information is passed to the PC in the Video mode for the software decoding to double-check the JPEG VLC stream data integrity.

31.4.10 P_MPEG4_JFIF_END (0x8822061C): JFIF-Compliant Data Decompression Finish

NAME	D0
P_MPEG4_JFIF_END	JFIFEND

JFIFEND: After JFIF-compliant data is decompressed, check this bit to identify if data stream is decompressed completely or not.

0: Error happens.

1: Decompressed completely.

31.4.11 P_MPEG4_RESETMCU_X: Restart MCU

NAME	Address	D7-D0
P_MPEG4_RESETMCU_LOW	0x88220620	RESTARTMCUL
NAME	**Address**	**D15-D8**
P_MPEG4_RESETMCU_HIGH	0x88220624	RESTARTMCU

RESTARTMCUX: If the vlc stream includes MCU restart code, fill the number of restart MCU code to these two registers.

Otherwise, set the number of restart MCU code as 0 to disable restart code decompression.

31.4.12 P_MPEG4_IFRAME_QSCALE(0x88220640): Quantizer Scale Value for I-Frame

NAME	D4-D0
P_MPEG4_IFRAME_QSCALE	IFRAMEQSCALE

31.4.13 P_MPEG4_PFRAME_QSCALE(0x88220644): Quantizer Scale Value for P-Frame

NAME	D4-D0
P_MPEG4_PFRAME_QSCALE	PFRAMEQSCALE

31.4.14 P_MPEG4_MATCHCODE_CTRL (0x88220648): Match Code Function Control

NAME	D1-D0
P_MPEG4_MATCHCODE_CTRL	MATCHCNT

MATCHCNT: Match Code Function Control

0: Disable match code function

1: Enable match 2 bytes

2: Enable match 4 bytes

31.4.15 P_MPEG4_MATCH_CODEX: Match Code

NAME	Address	D7-D0
P_MPEG4_MATCH_CODE0	0x8822064C	MATCHCODE0
NAME	**Address**	**D15-D8**
P_MPEG4_MATCH_CODE1	0x88220650	MATCHCODE1
NAME	**Address**	**D23-D16**
P_MPEG4_MATCH_CODE2	0x88220654	MATCHCODE2
NAME	**Address**	**D31-D24**
P_MPEG4_MATCH_CODE3	0x88220658	MATCHCODE3

31.4.16 P_MPEG4_MATCHCODE_OFFSET(0x8822065C): Offset Value

NAME	D7-D0
P_MPEG4_MATCHCODE_OFFSET	OFFSET

31.4.17 P_MPEG4_VOPTIMEINC_SEL (0x88220660): VOP Time Increment Mode Selection

NAME	D7-D0
P_MPEG4_VOPTIMEINC_SEL	VOPTIMEINC

VOPTIMEINC: VOP time increment mode selection

0: VOP time increment is based on P_MPEG4_MS_COUNT (0x8822_0664) and P_MPEG4_MS_EXTRAEN (0x8822_0668)

1: VOP time increment is based on P_MPEG4_VOPTIMEINC_SEL (0x8822_0688) and P_MPEG4_VOPTIMEINC_SEL (0x8822_068C)

31.4.18 P_MPEG4_MS_COUNT (0x88220664): Mini Second Counter

NAME	D5-D0
P_MPEG4_MS_COUNT	MSCNT

31.4.19 P_MPEG4_MS_EXTRAEN (0x88220668): Enable One Extra Mini Second Added Counter

NAME	D7
P_MPEG4_MS_EXTRAEN	MSPLUSCNTEN

31.4.20 P_MPEG4_MS_EXTRACTRL(0x8822066C): Extra Mini Second Added Control

NAME	D7	D6-D5	D4-D0
P_MPEG4_MS_EXTRACTRL	MSPLUSCNTEN		MSPLUSCNT

MSPLUSCNT: One Extra mini Second Added Counter

MSPLUSCNTEN: Enable One Extra mini Second Added Counter

31.4.21 P_MPEG4_VOPTIMEINC_RESx: VOP Time Increment Resolution

NAME	Address	D7-D0
P_MPEG4_VOPTIMEINC_RESLOW	0x88220680	VOPTimeIncResL
NAME	Address	D15-D8
P_MPEG4_VOPTIMEINC_RESHIGH	0x88220684	VOPTimeIncRes

31.4.22 P_MPEG4_VOPTIMEINC_UNITx: VOP Time Increment Unit Setting

NAME	Address	D7-D0
P_MPEG4_VOPTIMEINC_UNITLOW	0x88220688	VOPTimeIncL
NAME	Address	D15-D8
P_MPEG4_VOPTIMEINC_UNITHIGH	0x8822068C	VOPTimeInc

31.4.23 P_MPEG4_VOPTIMEINC_LENGTH(0x88220690): Length of VOP Time Increment

NAME	D3-D0
P_MPEG4_VOPTIMEINC_LENGTH	VOPTimeIncLen

VOPTimeIncLen= Length of VOP time Increment Resolution – 1

31.4.24 P_MPEG4_VLCSIZE_x: Decompressed Data Capability

NAME	Address	D7-D0
P_MPEG4_VLCSIZE_LOW	0x882206C8	VLCSizeL
NAME	**Address**	**D15-D8**
P_MPEG4_VLCSIZE_MID	0x882206CC	VLCSizeM

31.4.25 MPEG4_SRAM_EN(0x882206F8): SRAM On/Off Control

NAME	D0
P_MPEG4_SRAM_EN	SRAM_CS_N

SRAM_CS_N: 0: Turn off static ram (default)

1: Turn on static ram

31.4.26 P_MPEG4_AUTO_RESET(0x882206FC): Decompress Auto Reset Function On/Off Control

NAME	D0
P_MPEG4_AUTO_RESET	AUTO_RESET

AUTO_RESET: 0: Turn on auto reset during decompression (default)

1: Turn off auto reset during decompression

31.4.27 P_MPEG4_SRAM_TEST (0x88220780): Internal SRAM Test Selection

NAME	D5-D4	D3-D2	D1-D0
P_MPEG4_SRAM_TEST	SRAMPAGE		SRAMSEL

SRAMSEL: Internal SRAM test selection

SRAMPAGE: Internal SRAM test page selection

31.4.28 P_MPEG4_JPEGPROBE_SEL (0x88220784): JPEG Probe Bus Selection

NAME	D7-D0
P_MPEG4_JPEGPROBE_SEL	JPG_SEL

31.4.29 MPEG4_BLOCK_END (0x88220790): Block End Flag

NAME	D0
MPEG4_BLOCK_END	BlockEnd

In Stop Mode, the status will be active high at the end of block. Write 0 to clear this flag.

31.4.30 P_MPEG4_DEHUFFMAN_CTRL (0x88220798): Dehuffman Mode Control

NAME	D0
P_MPEG4_DEHUFFMAN_CTRL	Dehuffman

Dehuffman: Dehuffman mode selection

0: Disable hardware de-huffman function

1: Enable hardware de-huffman function

31.4.31 P_MPEG4_DEHUFFMAN_READY (0x8822079C): Firmware Dehuffman Flow Receiving Ready

NAME	D0
P_MPEG4_DEHUFFMAN_READY	Dehuffmanrdy

DeHuffmanrdy: Firmware dehuffman flow: Check if inverse quantization engine is ready to receive VLD data extracted from firmware.

31.4.32 P_MPEG4_DEHUFFMAN_DATAx: Firmware Dehuffman Flow Port

NAME	Address	D7-D0
P_MPEG4_DEHUFFMAN_DATALOW	0x882207A0	VLDDataL
NAME	Address	D10-D8

P_MPEG4_DEHUFFMAN_DATAHIGH	0x882207A4	VLDData

Firmware dehuffman flow: VLD data is fed to the 0x882207A0 and 0x882207A4 data ports.

31.4.33 P_MPEG4_VIDEO_RESET (0x882207A8): Video Reset

NAME	D0
P_MPEG4_VIDEO_RESET	VRstMode

VRstMode: 0: Video Reset until the first frame occurs

1: Video Reset until the first non-skip frame occurs

31.4.34 P_MPEG4_H263_CTRL(0x882207C8): H263 Format Control

NAME	D7	D6	D5-D4	D3-D1	D0
MPEG4_H263_CTRL	ACPREDEn	FOURMVEn		H263Format	H263En

H263En: Enable H.263 Encoding/Decoding

0: Disable

1: Enable

H263Format: Specify H.263 Format

001: Sub-QCIF (128x96)

010: QCIF (176x144)

011: CIF (352x288)

100: 4CIF (704x576)

FOURMVEn: Enable 4MV Decoding mode

0: Disable

1: Enable

ACPREDEn: Enable AC Prediction decoding mode

0: Disable

1: Enable

31.4.35 P_MPEG4_H263_STRUCTURE(0x882207CC): H263 Structure

NAME	D7-D4	D3-D1	D0

P_MPEG4_H263_STRUCTURE	MBNumLen		H263GOBEn

H263GOBEn: Enable H.263 GOB structure for Encoding/Decoding

MBNumLen: Specify number of bits needed for certain resolution

31.4.36 P_MPEG4_HMVSUM_X: Absolute Horizontal Movement Vector Sum (HalfPel Unit)

NAME	Address	D7-D0
P_MPEG4_HMVSUM_LOW	0x882207D0	HMVSumL
NAME	**Address**	**D14-D8**
P_MPEG4_HMVSUM_HIGH	0x882207D4	HMVSum

31.4.37 P_MPEG4_VMVSUM_x: Absolute Vertical Movement Vector Sum (HalfPel Unit)

NAME	Address	D7-D0
P_MPEG4_VMVSUM_LOW	0x882207D8	VMVSumL
NAME	**Address**	**D14-D8**
P_MPEG4_VMVSUM_HIGH	0x882207DC	VMVSum

31.4.38 MPEG4_GATE_CLK(0x882207E4): Region of Interest Mode

NAME	D2	D1	D0
P_MPEG4_GATE_CLK	ROIEn		DISGatedClock

DISGatedClock: Gated Clock Disable

0: Enable Gated Clock

1: Disable Gated Clock

ROIEn: Enable Region of Interest mode

31.4.39 P_MPEG4_HUFFMAN_START / END: Huffman Table

NAME	D7-D0
P_MPEG4_HUFFMAN_START	0x88220800
P_MPEG4_HUFFMAN_END	0x882208FC

31.4.40 P_MPEG4_LUMDCCODE_START / END: Luminance DC Codeword

NAME	D7-D0
P_MPEG4_LUMDCCODE_START	0x88220800
P_MPEG4_LUMDCCODE_END	0x8822085C

31.4.41 P_MPEG4_LUMOFFSET_SA / END: Luminance Address Offset for DC First Codeword

NAME	Address
P_MPEG4_LUMOFFSET_SA	0x88220860
P_MPEG4_LUMOFFSET_END	0x8822087C

Luminance address offset for DC first codeword (TsramMode = 2), 8-bit register.

31.4.42 P_MPEG4_LUMHUFFMANTABLE_START / END: Luminance DC Huffman Table Content

NAME	Address
P_MPEG4_LUMHUFFMANTABLE_START	0x88220880
P_MPEG4_LUMHUFFMANTABLE_END	0x8822089C

Luminance DC Huffman table content (TsramMode = 2), 8-bit register.

31.4.43 P_MPEG4_CHROMDCCODE_START / END: Chrominance DC Codeword

NAME	Address
P_MPEG4_CHROMDCCODE_START	0x882208C0
P_MPEG4_CHROMDCCODE_END	0x8822091C

Chrominance DC codeword (TsramMode = 2), 8-bit register.

31.4.44 P_MPEG4_CHROMOFFSET_SA / EA: Chrominance DC Codeword Address Offset

NAME	Address
P_MPEG4_CHROMOFFSET_SA	0x88220920
P_MPEG4_CHROMOFFSET_EA	0x8822093C

Chrominance DC codeword (TsramMode = 2), 8-bit register.

31.4.45 P_MPEG4_CHROMHUFFMANTABLE_START / END: Chrominance DC Huffman Table Content

NAME	Address
P_MPEG4_CHROMHUFFMANTABLE_START	0x88220920
P_MPEG4_CHROMHUFFMANTABLE_END	0x8822093C

Chrominance DC Huffman table content (TsramMode = 2), 8-bit register.

31.4.46 P_MPEG4_LUMACHUFFMANTABLE_START / END: Luminance AC Huffman Table Codeword

NAME	Address
P_MPEG4_LUMACHUFFMANTABLE_START	0x88220980
P_MPEG4_LUMACHUFFMANTABLE_END	0x882209DC

Luminance AC Huffman table codeword (TsramMode = 2), 8-bit register.

31.4.47 P_MPEG4_LUMACHUFFMANOFFSET_SA / EA: Luminance AC Huffman Table Address Offset for AC First 1-Bit Codeword

NAME	Address
P_MPEG4_LUMACHUFFMANOFFSET_SA	0x882209E0
P_MPEG4_LUMACHUFFMANOFFSET_EA	0x88220A34

Luminance AC Huffman table address offset for AC first 1-bit codeword (TsramMode = 2), 8-bit register.

31.4.48 P_MPEG4_CHROMACHUFFMANTABLE_START / END: Chrominance AC Huffman Table Codeword

NAME	Address
P_MPEG4_CHROMACHUFFMANTABLE_START	0x88220A40
P_MPEG4_CHROMACHUFFMANTABLE_END	0x88220A90

Chrominance AC Huffman table codeword (TsramMode = 2), 8-bit register.

31.4.49 P_MPEG4_CHROMACHUFFMANOFFSET_SA / EA: Chrominance AC Huffman Table Address Offset for Codeword

NAME	Address
P_MPEG4_CHROMACHUFFMANOFFSET_SA	0x88220AA0
P_MPEG4_CHROMACHUFFMANOFFSET_EA	0x88220AF4

Chrominance AC Huffman table address offset for codeword (TsramMode = 2), 8-bit register.

31.5 MPEG/JPEG Engine Limitation

31.5.1 JPEG Engine Limitation

Engine	Mode	Description
JPEG engine	Capture	• YUV422: Image width: Multiple of 16 (<4080) Image height: Multiple of 8 (<4080) • YUV420: Image width: Multiple of 16 (<4080) Image height: Multiple of 16 (<4080) Cropping image: Yes Rotation: Yes Flip: Yes Scaling down: No Scaling up: Yes (8-bit factor)
	Playback	• YUV422: Image width: Multiple of 16 (<4080) Image height: Multiple of 8 (<4080) • YUV420: Image width: Multiple of 16 (<4080) Image height: Multiple of 16 (<4080) • YUV411: Image width: Multiple of 16 (<4080) Image height: Multiple of 8 (<4080 • YUV444:

Engine	Mode	Description
		Image width: Multiple of 16 (<4080) Image height: Multiple of 16 (<4080) Cropping image: Yes Rotation: No Flip: No Scaling down: only 1/8 ~ 7/8 Scaling up: No

31.5.2 MPEG Engine Limitation

Engine	Mode	Description
MPEG engine	Video clip,	Image width: Multiple of 16 (<=640) Image height: Multiple of 16 (<= 480) Cropping image: Yes Rotation: Yes Flip: Yes Scaling down: No Scaling up : Yes (8-bit factor)
MPEG engine	Video Playback	Image width: Multiple of 16 (<=640) Image height: Multiple of 16 (<= 480) Cropping image: No Rotation: No Flip: No Scaling down: No Scaling up : No