



GPCM3 Series Programming Guide

Version 2.0 - Dec. 04, 2023

Generalplus Technology Inc. reserves the right to change this documentation without prior notice. Information provided by Generalplus Technology Inc. is believed to be accurate and reliable. However, Generalplus Technology Inc. makes no warranty for any errors which may appear in this document. Contact Generalplus Technology Inc. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by Generalplus Technology Inc. for any infringement of patent or other rights of third parties which may result from its use. In addition, Generalplus products are not authorized for use as critical components in life support devices/systems or aviation devices/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 Generalplus.

Table of Contents

	<u>PAGE</u>
Table of Contents	2
REVISION HISTORY	8
1. General Description.....	9
2. Features	9
2.1. GPCM3 Block Diagram	10
2.2. GPCM3 IO Special Functions.....	11
3. Memory Organization.....	13
4. ARM® Cortex®-M0 Core.....	14
4.1. Overview	14
4.2. System Control Block	15
4.3. Registers.....	15
4.4. System timer, SysTick.....	22
4.5. Register	22
5. Nested Vectored Interrupt Controller (NVIC).....	25
5.1. Overview	25
5.2. Features.....	25
5.3. Exception Types	25
6. SYSTEM Descriptions	27
6.1. Reset System	27
6.1.1. Introduction	27
6.1.2. Block Diagram	27
6.1.3. Function	27
6.1.4. Register Description.....	28
6.2. Clock Control Unit.....	30
6.2.1. Introduction	30
6.2.2. Block Diagram	31
6.2.3. Register Description.....	31
6.3. Low Power Mode & Shut Down Mode	39
6.3.1. Introduction	39
6.3.2. Function	39
6.4. System Management Unit.....	40
6.4.1. Introduction	40
6.4.2. Register Description.....	40
6.5. Interrupt Control Unit.....	43
6.5.1. Introduction	43
6.5.2. Function	44
6.5.3. Register Description.....	45
7. Analog Control Unit.....	53

7.1.	<i>Introduction</i>	53
7.2.	<i>Features</i>	53
7.3.	<i>Function</i>	53
7.4.	<i>Register Description</i>	54
8.	DMA Control Unit.....	61
8.1.	<i>Introduction</i>	61
8.2.	<i>Features</i>	61
8.3.	<i>Block Diagram</i>	61
8.4.	<i>Function Description</i>	62
8.4.1.	DMA Transaction.....	62
8.4.2.	DMA Request Arbiter.....	62
8.4.3.	DMA Channels	62
8.4.4.	Programmable Data Width and Data Alignment	63
8.4.5.	Error Management	63
8.4.6.	Interrupts	64
8.4.7.	Channel Selection	64
8.5.	<i>Register Description</i>	65
9.	CCP0/1 Control.....	73
9.1.	<i>Introduction</i>	73
9.2.	<i>Feature</i>	73
9.3.	<i>Block Diagrams</i>	73
9.4.	<i>Function</i>	74
9.5.	<i>Register Description</i>	76
10.	MAC Controller.....	91
10.1.	<i>Introduction</i>	91
10.2.	<i>Features</i>	91
10.3.	<i>Block Diagram</i>	92
10.4.	<i>Register Description</i>	92
10.4.1.	Registers Map	92
10.4.2.	Registers Function.....	93
11.	Timer	108
11.1.	<i>Introduction</i>	108
11.2.	<i>Feature</i>	108
11.3.	<i>Blcok Diagram</i>	108
11.4.	<i>Function</i>	108
11.5.	<i>Register Description</i>	108
12.	General-purpose I/O (GPIOs)	116
12.1.	<i>Introduction</i>	116
12.2.	<i>Feature</i>	116

12.3.	<i>Function</i>	116
12.4.	<i>Register Function</i>	118
13.	Line-IN ADC (SAR ADC) Controller	168
13.1.	<i>Introduction</i>	168
13.2.	<i>Features</i>	168
13.3.	<i>Block Diagram</i>	168
13.4.	<i>Function</i>	169
13.4.1.	SAR ADC On-Off Control	169
13.4.2.	Channel Selection	169
13.4.3.	Sampling Path	169
13.4.4.	Regulation Conversion Mode	170
13.4.5.	Injected Conversion Mode	171
13.4.6.	Combined Regular/Injected Simultaneous Mode	172
13.4.7.	Regular Conversion with DMA	174
13.4.8.	Data Alignment	174
13.4.9.	Analog Watchdog	175
13.4.10.	SAR ADC Program Example	176
13.5.	<i>Register Description</i>	176
14.	Audio Delta-Sigma ADC	190
14.1.	<i>Introduction</i>	190
14.2.	<i>Feature</i>	191
14.3.	<i>Block Diagram</i>	191
14.4.	<i>Register Description</i>	192
15.	DAC Up-Sampling/Audio PWM Control	201
15.1.	<i>Introduction</i>	201
15.2.	<i>Feature</i>	201
15.3.	<i>Block Diagram</i>	201
15.4.	<i>Function</i>	202
15.5.	<i>Register Description</i>	203
16.	Serial peripheral interface (SPI)	213
16.1.	<i>Introduction</i>	213
16.2.	<i>Features</i>	213
16.3.	<i>Serial Interface Control Pin Configuration</i>	213
16.3.1.	SPI0 IO Selection	213
16.3.2.	SPI1 IO Selection	213
16.4.	<i>Block Diagram of SPI</i>	214
16.5.	<i>Function</i>	215
16.5.1.	SPI Clock Phase and Clock Polarity	215
16.5.2.	Master Mode	216
16.5.3.	Slave Mode	219
16.6.	<i>Register Description</i>	222
16.6.1.	<i>Register map</i>	222

16.6.2.	Register Function	222
17.	SPI controller for FLASH device access (SPIFC)	227
17.1.	<i>Introduction</i>	227
17.2.	<i>Features</i>	227
17.3.	<i>Block Diagram</i>	227
17.4.	<i>Function</i>	228
17.4.1.	Pin Descriptions	228
17.4.2.	Timing Types	228
17.4.3.	Packet Types.....	229
17.5.	<i>Register Description</i>	232
17.5.1.	Register Map.....	232
17.5.2.	Register Function	232
18.	Universal Asynchronous Receiver Transmitter (UART)	242
18.1.	<i>Introduction</i>	242
18.2.	<i>Features</i>	242
18.3.	<i>Block Diagram</i>	242
18.4.	<i>Function</i>	243
18.4.1.	Frame Format	243
18.4.2.	Transmitter.....	244
18.4.3.	Signe Byte Transmission.....	245
18.4.4.	Transmit with DMA Function	246
18.4.5.	Receiver	246
18.4.6.	Baud Rate Generation.....	246
18.4.7.	Signe Byte Receive	247
18.4.8.	Receive with DMA Function.....	247
18.4.9.	Break Frame.....	248
18.4.10.	Idle Frame	248
18.4.11.	Overrun Error.....	248
18.4.12.	Parity Check	248
18.5.	<i>Register Description</i>	248
19.	Inter Integrated Circuit Interface (I2C)	256
19.1.	<i>Introduction</i>	256
19.2.	<i>Features</i>	256
19.3.	<i>Block Diagram</i>	256
19.4.	<i>Function</i>	257
19.4.1.	Master Mode	258
19.4.2.	Slave Mode	261
19.5.	<i>Register Description</i>	263
20.	I2S.....	269
20.1.	<i>Introduction</i>	269
20.2.	<i>Features</i>	269
20.3.	<i>Block Diagram</i>	269
20.4.	<i>Function</i>	270

20.4.1.	I2S Bus.....	270
20.4.2.	Normal Mode and I2S Mode.....	270
20.4.3.	I2S RX Interrupt.....	271
20.4.4.	I2S RX FIFO	271
20.5.	<i>Register Description</i>	271
21.	PWM I/O.....	279
21.1.	<i>Introduction</i>	279
21.2.	<i>Features</i>	279
21.3.	<i>Function</i>	279
21.4.	<i>Register Description</i>	279
22.	Quadrature Decoder.....	297
22.1.	<i>Introduction</i>	297
22.2.	<i>Features</i>	297
22.3.	<i>Function</i>	297
22.4.	<i>Register Description</i>	298
23.	Capacitive Touch Sensor (CTS).....	305
23.1.	<i>Introduction</i>	305
23.2.	<i>Feature</i>	305
23.3.	<i>Block Diagram</i>	305
23.4.	<i>Function</i>	305
23.5.	<i>Register Description</i>	306
24.	TIMEBASE Control	313
24.1.	<i>Introduction</i>	313
24.2.	<i>Feature</i>	313
24.3.	<i>Note</i>	313
24.4.	<i>Block Diagram</i>	313
24.5.	<i>Function</i>	314
24.6.	<i>Register Description</i>	314
25.	Watchdog Timer.....	318
25.1.	<i>Introduction</i>	318
25.2.	<i>Features</i>	318
25.3.	<i>Block Diagram</i>	318
25.4.	<i>Function</i>	318
25.5.	<i>Register Description</i>	319
26.	Sound Process Unit Control Register	322
26.1.	<i>General Description</i>	322
26.2.	<i>Feature</i>	322

26.3.	<i>Block Diagram</i>	322
26.4.	<i>Internal Memory Mapping</i>	323
26.5.	<i>Gain Control</i>	324
26.6.	<i>Fundamentals</i>	324
26.7.	<i>Musical Sound</i>	325
26.8.	<i>Control Register List</i>	327
26.9.	<i>Internal SRAM List</i>	330
26.10.	<i>SPU Control Register</i>	332
26.10.1.	SPU Control Flag	332
26.10.2.	Channel Enable Control	333
26.10.3.	Main Volume Control.....	334
26.10.4.	Beat Control	335
26.10.5.	Tone Control	336
26.10.6.	Channel IRQ Control	337
26.10.7.	PitchBend Enable Control	339
26.10.8.	Envelope Control	339
26.10.9.	Compressor Control.....	344
26.10.10.	Post Process Control	347
26.11.	<i>Internal SRAM</i>	350
26.11.1.	Tone-Color Register	353
26.11.2.	Volume Register.....	358
26.11.3.	Envelope Register	359
26.11.4.	Phase Register	365
26.11.5.	PitchBend Register.....	368
26.12.	<i>Programming Tips</i>	370
26.13.	<i>Programming Flow Example</i>	371
27.	Universal Serial Bus (USB).....	372
27.1.	<i>Introduction</i>	372
27.2.	<i>Features</i>	372
27.3.	<i>Serial Interface Control Pin Configuration</i>	372
27.4.	<i>USB Buffer Mapping</i>	372
27.5.	<i>Register Description</i>	372
27.5.1.	Register Map.....	372
27.5.2.	Register Function	373
28.	Emulation Board (EMU) and Piggyback	389
28.1.	<i>Introduction</i>	389
28.2.	<i>GPCM3 Piggyback</i>	389
28.3.	<i>GPCM3 EMU Board</i>	391

REVISION HISTORY

Revision	Date	By	Remark
V2.0	2023/12/04	Kim Huang	<ol style="list-style-type: none">1. Modify the description of PLL ACU_PLL_CTRL.DIVC.2. Modify the definition of ACU.REG33_CTRL Bit[5].3. Modify the supported voltage range of SAR ADC (from 2.4 V ~5.5 V to 2.7V~3.6V).4. Modify Channel Selection Table in Ch13.4.2.5. Modify the definition of ACU_APAD[7:0]Line-in pad.6. Modify the IO number Shut down mode supported (from 8*IO to 16*IO).7. Modify the description of ACU_REG33_CTRL, ACU_REG12_CTRL, ACU_PLL_CTRL, ACU_X32K_CTRL, AUDPWM_CTRL2, BEAT_COUNTER, CCP0_TMCMP_CTRL, CCP0_CAP_CTRL, CCP0_PWM_CTRL, CCP0_PWM_DTIME, COMPRESSOR, DAC_CHO_DMA_DATA0, DMA0_CTRL, DMA0_DTN, DAC_CTRL, DS_ADC_CTRL, DS_ADC_AGC_CTRL0&1, DS_ADC_MUTE_CTRL0, DS_ADC_STS, DS_ADC_DATA, ENV_DATA, I2S_RX_CTRL, I2S_TX_CTRL2, ICSR, ITU_SWIRQ, MAC_CTRL, MAC_LENGTH, MAC_ADDR_X&Y, MAC_STEP_XY, MAC_OUTPUT, MAC_OUTPUT_EXT8, MAC_MUL1_OUT, MAC_OUTPUT_EXT8_R, MAC_SATURATION_R, MAC_MUL2_OUT, PWMIO_CTRL0/1, PWMIO_PWMIO1_Duty, QD_FW_TH, QD_BW_TH, SPI0/1_CTRL, SPI0/1_STS, SPIFC_CTRL0/1/2, SPIFC_RX_BC, SPIFC_ADDRL, SPIFC_TIMING, SPU_CTRL_FLAG, TMO_CTRL, TMO_TMR_PLOAD, TM_INT_INTSTS, RCU_CTRL, SMU_SYSLOCK [7:0], UART0/1_CTRL, UART0/1_STS, UART0_FIFO, USB_DOWN, USB2_CTRL, USB2_INTEN, USB2_INTSTS, WAVE_ADPCM_CTRL Register.8. Modify the condition of AUDPWM_CTRL, ACU_APAD, CLOCK_SWTRIM, CTS_PADSEL, I2C_STS, I2S_RX_STS, MAC_INTSTS, MAC_START_CNT, QD_CTRL, SMU_SHUTDOWN_ON, SMU_SHUTDOWN_OFF, USB_DOWN, USB_EOT.9. Modify the introduction of the following chapters: Audio Delta-Sigma ADC, CCP0/1 Control, CTS, DAC&Audio PWM, DMA Control Unit, GPIO, MAC, Timer, UART, I2C, I2S, Interrupt Control Unit, PWMIO, QD, SAR-ADC, SPI, SPIFC, SPU, Timebase, USB, Voltage Regulator 3.3V (REG33), Watchdog Timer.
V1.0	2022/11/9	Kim Huang	Original

1. General Description

The GPCM3 series of microcontrollers are based on the ARM® Cortex®-M0 processor core and operated at a frequency of up to 120MHz. The GPCM3 series are applicable to the areas of digital sound process and voice recognition.

2. Features

■ CPU Subsystem

- CPU Core
- ARM® Cortex®-M0 32-bit CPU (120MHz max) with Code Fetch Accelerator
- Nested Vectored Interrupt Controller (NVIC) with 32 Interrupt Sources
- 24-bit SysTick Timer
- Single Cycle 32-bit Multiplier Instruction

■ Memory

- 32KB SRAM and 8KB Cache
- Running SPI External Code and Data in Auto Mode

■ Clock Management

- Internal 12MHz Oscillator and XTAL 12M Crystal
- Phase Lock Loop with Configurable Output Frequency: 120M Hz (max)
- Internal 32768Hz Oscillator and XTAL 32768Hz Crystal

■ Power Management

- Sleep Mode: Only the CPU is stopped, and system clock is ON.
- Deep Sleep Mode: All clocks are stopped.
- Halt Mode: In deep sleep mode; but 32KHz RTC Clock is ON.
- Shut Down Mode: Core power and all clocks are stopped.
- Regulator with configurable output SPI Voltage and Codec voltage.
- Low Voltage Detection

■ Reset Management

- Power On Reset
- Low Voltage Reset
- Watchdog Timeout Reset
- PAD (H/W Key) Reset
- Master (S/W) Reset
- System Reset

■ Analog Peripherals

- 16-bit Audio PWM
- Two 16-bit DACs have two 16-bit software channels with noise filter mixer and scalar to playback high quality sounds.
- 16-bit Sigma Delta Codec ADC
- 12-bit SAR ADC and 8 Line-IN PADs
- CTS (Capacitive Touch Sensor) supports 32 Touch IO PADs.

■ Timer

- Three General-Purpose 16-bit Timers/Counters
- Two 16-bit touch sensing timers for CTS.
- Two 16-bit CCP Timers (Capture/Compare/PWM) Units and 8 PWMIOs
- All of them can be used as general-purpose timers.

■ System Control

- Five-Channel DMA Controller
- System Management Unit (SMU) for system configuration and control.
- 16x16 MAC Function
- Two Sets of Quadrature Decoder

■ I/O Ports

- Up to 43 general I/Os (including SPI I/F).
- I/O port with pull-up resistor, pull-down resistor, output high, output low or floating input.
- Programmable driving current: 8mA/16mA for IOA, IOC and IOD; 2 ~ 16mA for IOB.

■ Communication Peripherals

- One I2C Hardware
- One I2S Input (Slave Mode) and Output (Master Mode) Hardware
- One SPIFC (SPI Controller for FLASH Device Access) and two SPI Serial interface I/Os
- Two UART Hardwares:
- One IR TX Hardware
- One USB1.1 Hardware

■ PDM

- Pulse-Density Modulation MEMS MIC

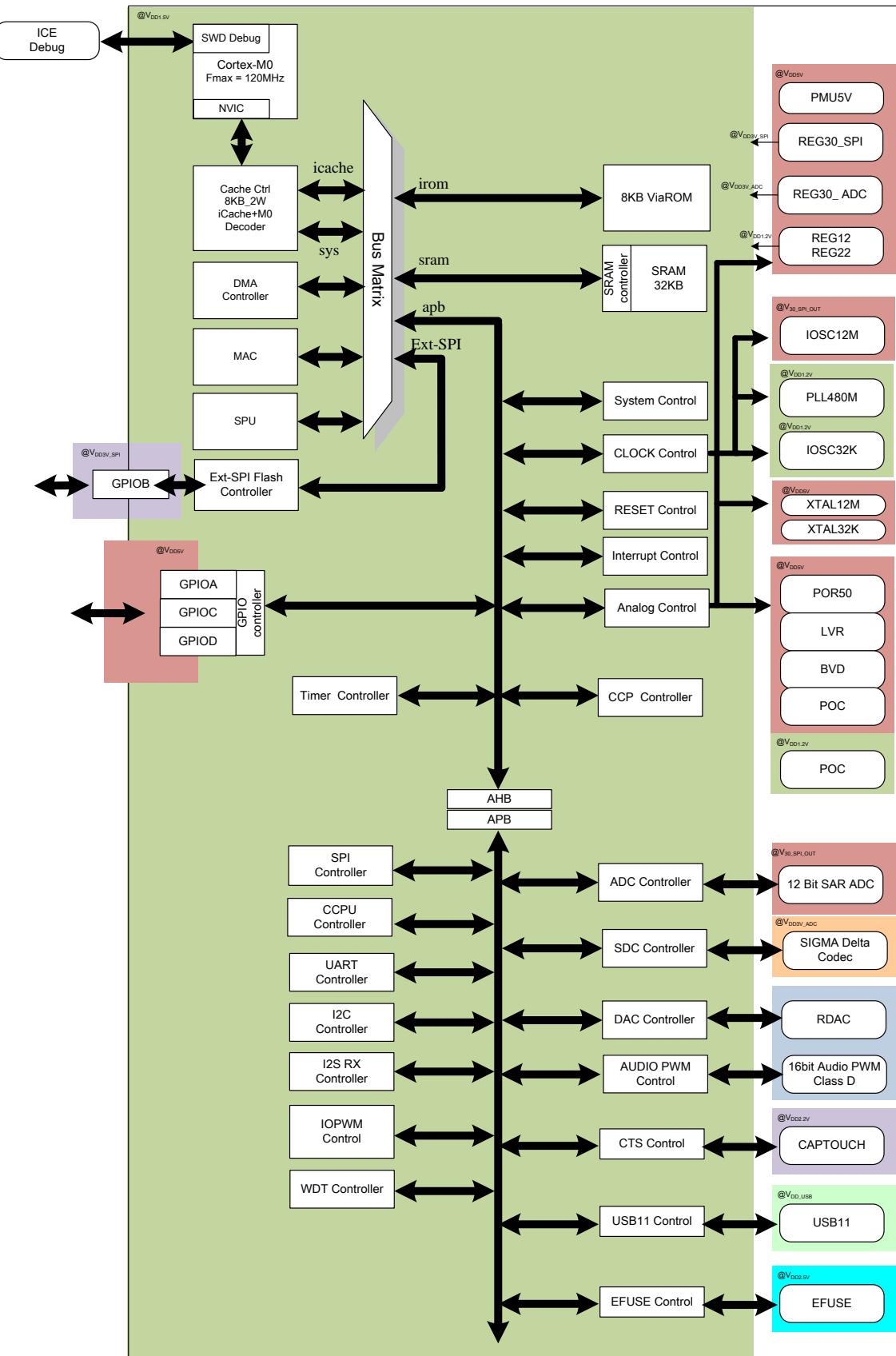
■ SPU

- 32-Channel Sound Process Unit

■ Debug System

- ARM Serial Wire Debug (SWD)
- Support up to 3 hardware breakpoints.

2.1.GPCM3 Block Diagram



2.2.GPCM3 IO Special Functions

IOA	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CCP0	-	-	-	CCP_A_0	CCP_A_1	CCP_A_2	CCP_A_3	-	-	-	-	-	-	CCP_B_0	CCP_B_1	CCP_B_2
CCP1	-	-	-	-	-	-	-	-	-	CCP_A_0	CCP_A_1	CCP_A_2	CCP_A_3	-	-	-
CTS	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
I2C	SCK_0	SDA_0	-	-	-	-	-	-	-	-	-	-	-	-	-	SCK_1
IOPWM	-	-	-	-	-	-	-	-	-	-	-	-	-	PWM0	PWM1	PWM2
I2S_In	-	-	-	MCLK	BCLK	LR	DATA	-	-	-	-	-	-	-	-	-
I2S_Out	-	-	-	-	-	-	-	-	-	MCLK	BCLK	LR	DATA	-	-	-
IR_Tx	-	-	-	-	-	-	-	TX_0	-	-	-	-	-	-	-	-
UART0	TX_0	RX_0	-	-	-	-	-	-	-	-	-	-	-	TX_1	RX_1	-
UART1	-	-	-	-	-	TX_0	RX_0	-	-	-	-	-	-	-	-	-
SPI0	-	-	-	SCK_0	MOSI_0	CS_0	MISO_0	-	-	-	-	-	-	-	-	-
SPI1	-	-	-	-	-	-	-	-	SCK_0	MOSI_0	CS_0	MISO_0	-	-	-	-
WakeUp	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WakeUp (SD)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	-
Feedback	-	-	-	IN_0	OUT_0	-	-	OUT_1	IN_1	-	-	-	-	-	-	-
EXT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PDM_In	-	-	-	-	-	DATA_0	CLK_0	-	-	-	-	-	-	-	-	-
TMCMP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
X32K	-	-	-	-	-	-	-	X32KO	X32KI	-	-	-	-	-	-	-

IOA	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
CCP0	CCP_B_3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CCP1	-	-	-	-	-	-	-	-	-	-	CCP_B_0	CCP_B_1	CCP_B_2	CCP_B_3	-	-
CTS	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
I2C	SDA_1	-	-	-	SCK_2	SDA_2-	-	-	-	-	-	-	-	-	-	-
IOPWM	PWM3	PWM4	PWM5	PWM6	PWM7	PWM8	PWM9	PWM10	PWM11	PWM12	PWM13	PWM14	PWM15	-	-	-
I2S_In	-	-	-	-	-	-	-	-	-	-	MCLK	BCLK	LR	DATA	-	-
I2S_Out	-	-	-	-	-	-	MCLK	BCLK	LR	DATA	-	-	-	-	-	-
IR_Tx	-	TX_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UART0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TX_2	RX_2
UART1	-	-	-	-	TX_1	RX_1	-	-	-	-	TX_2	RX_2-	-	-	-	-
SPI0	-	-	-	-	-	-	-	SCK_1	MOSI_1	CS_1	MISO_1	-	-	-	-	-
SPI1	-	-	-	-	-	-	SCK_1	MOSI_1	CS_1	MISO_1	-	-	-	-	-	-
WakeUp	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WakeUp (SD)	-	-	-	-	-	-	-	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes



GPCM3 SERIES PROGRAMMING GUIDE

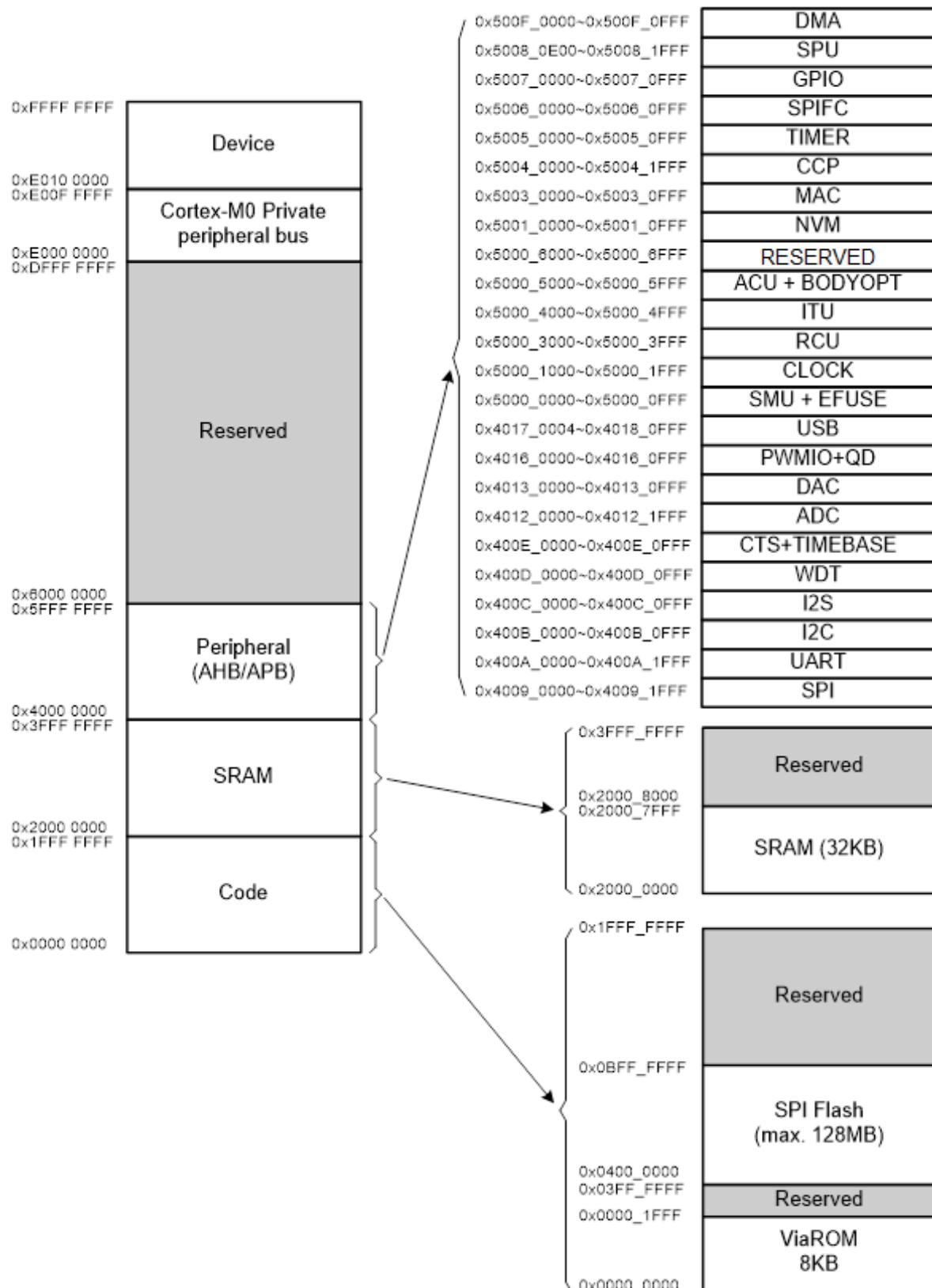
IOA	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
EXT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-
PDM_In	-	-	-	-	-	-	DATA_1	CLK_1	-	-	-	-	-	-	-	-
TMCMP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
LineIN	-	-	-	-	-	-	-	-	Yes							
QD	-	-	QD0_A	QD0_B	QD1_A	QD1_B	-	-	-	-	-	-	-	-	-	-

IOB	0	1	2	3	4	5	-	-	-	-	-	-	-	-	-	-
SPIFC	SIO3	SCLK	SIO0	CS	SIO1	SIO2	-	-	-	-	-	-	-	-	-	-
SPIO	-	SCLK_2	MOSI_2	CS_2	MISO_2	-	-	-	-	-	-	-	-	-	-	-
WakeUp	Yes	Yes	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	-	-	-
EXT	Yes	Yes	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	-	-	-
TMCAP	Yes	Yes	Yes	Yes	Yes	Yes	-	-	-	-	-	-	-	-	-	-

IOC	0	1	2	-	-	-	-	-	-	-	-	-	-	-	-	-
ICE I/F	-	-	SDA	-	-	-	-	-	-	-	-	-	-	-	-	-
X12M	X12MI	X12MO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
UARTO	-	-	TX_3	-	-	-	-	-	-	-	-	-	-	-	-	-

IOD	0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
USB	DP	DM	-	-	-	-	-	-	-	-	-	-	-	-	-	-

3. Memory Organization

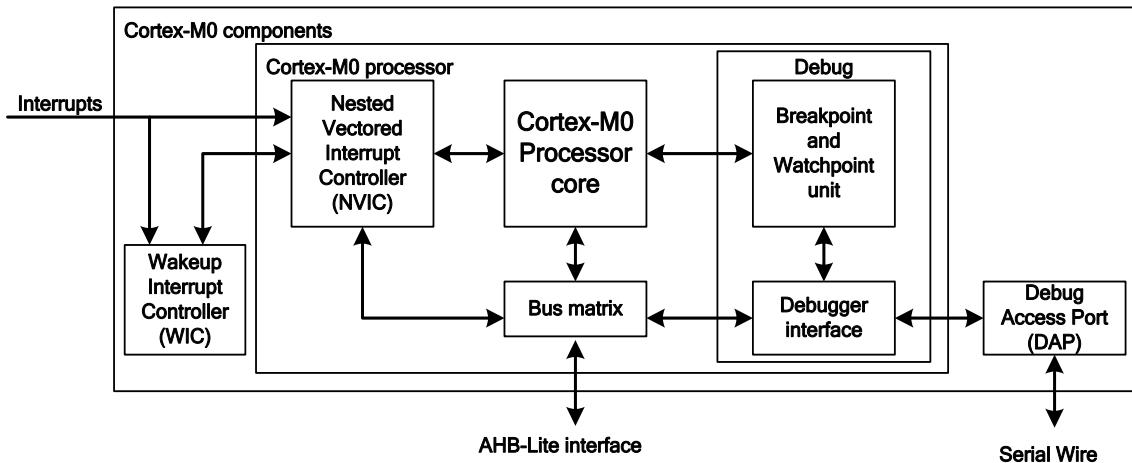


4. ARM® Cortex®-M0 Core

4.1. Overview

The Cortex®-M0 processor is a 32-bit entry-level ARM Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including:

- Simple and Easy-to-Use Programmer Models
- Highly Efficient Ultra-Low Power Operation
- Excellent Code Density
- Deterministic and High-Performance Interrupt Handling
- Upward compatibility with the rest of the Cortex-M processor family.



The Cortex-M0 processor is built on a 32-bit processor core, which is the optimization of area and low power consumption, with a 3-stage pipeline von Neumann architecture. The processor delivers exceptional energy efficiency through a small but powerful instruction set and extensively optimized design, and provides a high-end processing hardware with a single-cycle multiplier. The Cortex-M0 processor implements the ARMv6-M architecture, which is based on the 16-bit Thumb® instruction set and includes Thumb-2 technology. The Cortex-M0 instruction set provides exceptional performance expected of a modern 32-bit architecture, with a higher code density than other 8-bit and 16-bit microcontrollers. The Cortex-M0 processor closely integrates a configurable NVIC, to deliver industry leading interrupt performance. The NVIC provides 4 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of interrupt service routines (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to abandon and restart load-multiple and store-multiple operations. Interrupt handlers do not require any assembler wrapper code, which means it will remove any code overhead from the ISRs. Tail-chaining optimization also significantly reduces the overhead while switching from one ISR to another. To optimize low-power designs, the NVIC integrates with the sleep modes which include a deep sleep function that enables the entire device to be rapidly powered down.

Note: When GPCM3 EV chip is connected with G+Link pro, it cannot emulate the sleep mode and deep sleep mode. The G+Link pro must be disconnected with the EV chip first and powered on again to enter sleep mode and deep sleep mode.

Reference to ARM Documentation

- Cortex®-M0 Devices, Generic User Guide
- ARMv6-M Architecture Reference Manual
- Cortex Microcontroller Software Interface Standard (CMSIS)

4.2.System Control Block

The System Control Block (SCB) provides system implementation information and system control. This includes configuration, control, and reporting of the system exceptions.

4.3.Registers

Register Map

Name	Address	Description
CPU_ID	0xE000ED00	CPUID Base Register The CPUID register contains the processor part number, version, and implementation information.
ICSR	0xE000ED04	Interrupt Control and State Register The ICSR provides: <ul style="list-style-type: none">– Set-pending and clear-pending bits for the PendSV and SysTick exceptions.– The exception number of the exception being processed.– ICSR determines whether there are preempted active exceptions.– The exception number of the highest priority pending exception.– ICSR determines whether there's any pending interrupt.
AIRCR	0xE000ED0C	Application Interrupt and Reset Control Register The AIRCR register provides endian status for data accesses and reset control of the system. To write to this register, user must write 0x5FA to the VECTKEY field, otherwise the processor will ignore the write.
SCR	0xE000ED10	System Control Register The SCR controls features of entry to and exit from low power state.
CCR	0xE000ED14	Configuration and Control Register The CCR is a read-only register and it indicates some aspects of the behavior of the Cortex-M0 processor.
SHPR2	0xE000ED1C	System Handler Priority Register 2 The SHPR2 register sets the priority level for the SVCall handler.
SHPR3	0xE000ED20	System Handler Priority Register 3 The SHPR3 register sets the priority level for the SysTick and PendSV handlers.
SHCSR	0xE000ED24	System Handler Control and State Register The SHCSR register controls and provides the status of system handlers.

Registers

CPU_ID		0xE000ED00								CPUID Base Register	
Bit	31	30	29	28	27	26	25	24			
Function	IMPLEMENTER[7:0]										
Reset Value	0	0	0	0	1	0	0	1			

Bit	23	22	21	20	19	18	17	16
Function	VARIANT[3:0]					ARCHITECTURE[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	PARTNO[11:4]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PARTNO[3:0]					REVISION[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:24]	IMPLEMENTER[7:0]	R	Implementer code assigned by ARM. ARM == 0x41					-
[23:20]	VARIANT[3:0]	R	Reads as 0x0.					-
[19:16]	ARCHITECTURE[3:0]	R	Reads as 0xC for ARMv6-M parts.					-
[15:4]	PARTNO[11:0]	R	Reads as 0xC20.					-
[3:0]	REVISION[3:0]	R	Reads as 0x0.					-

ICSR									0xE000ED04		Interrupt Control and State Register		
Bit	31	30	29	28	27	26	25	24					
Function	NMIPENDSET	-	-	PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	-					
Reset Value	0	0	0	0	0	0	0	0					

Bit	23	22	21	20	19	18	17	16
Function	ISRPREEMPT	ISRPENDIN	-	VECTPENDING[8:4]				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	VECTPENDING[3:0]					-	VECTACTIVE[8]	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	VECTACTIVE[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31]	NMIPENDSET	R/W	NMI Set-Pending Bit Note: Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. After entering the handler, this bit will be cleared as 0.					Write 0: No effect. Write 1: Change NMI exception state to pending. Read 0: Not pending. Read 1: NMI exception pending.

Bit	Function	Type	Description	Condition
			This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.	
[30:29]	-	R	Reserved	-
[28]	PENDSVSET	R/W	PendSV Set-Pending Bit Note: Writing 1 to this bit is the only way to change the PendSV exception state to pending.	Write 0: No effect. Write 1: Change PendSV exception state to pending. Read 0: No pending. Read 1: PendSV exception is pending.
[27]	PENDSVCLR	W	PendSV Clear-Pending Bit Note: This bit is write-only. To clear the PENDSV bit, user must write 0 to PENDSVSET and write 1 to PENDSVCLR at the same time.	Write 0: No effect. Write 1: Remove the pending state from the PendSV exception.
[26]	PENDSTSET	R/W	SysTick Exception Set-Pending Bit	Write 0: No effect. Write 1: Change SysTick exception state to pending. Read 0: No pending. Read 1: SysTick exception is pending.
[25]	PENDSTCLR	W	SysTick Exception Clear-Pending Bit Note: This bit is write-only. When user wants to clear PENDST bit, please write 0 to PENDSTSET and write 1 to PENDSTCLR at the same time.	Write 0: No effect. Write 1: Remove the pending state from the SysTick exception.
[24]	-	R	Reserved	-
[23]	ISRPREEMPT	R	Interrupt Preemption Bit This bit is read only.	Read 0: No action. Read 1: A pending exception will be executed after the system exits debug halt state.
[22]	ISRPENDING	R	Interrupt Pending Flag This bit is read only. NMI and Faults are excluded.	0: No pending. 1: Interrupt Pending
[21]	-	R	Reserved	-
[20:12]	VECTPENDING[8:0]	R	Exception Number Of The Highest Priority Pending Enabled Exception VECTPENDING is read only.	0: No pending. >=1: Exception number of the highest priority pending enabled exception.
[11:9]	-	R	Reserved	-
[8:0]	VECTACTIVE[8:0]	R	VECTACTIVE contains the Active Exception Number, and it's read only.	0: Thread Mode >=1: Exception number of the currently active exception.

AIRCR **0xE000ED0C** Application Interrupt and Reset Control Register

Bit	31	30	29	28	27	26	25	24
Function	VECTORKEY[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	VECTORKEY[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SYSRESETREQ							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	VECTORKEY[15:0]	R/W	Register Access Key Write: When user writes to this register, the VECTORKEY field need to be set as 0x05FA, otherwise the write operation would be ignored. The VECTORKEY filed is used to prevent accidental write to this register which will reset the system or clear the exception status. Read: Read as 0xFA05.	-
[15:3]	-	R	Reserved.	-
[2]	SYSRESETREQ	W	System Reset Request Writing 1 to this bit will cause a reset signal to be asserted to the chip which indicates a reset requested. The bit is a write only bit, and self-clear is a part of the reset sequence.	-
[1]	VECTCLRACTIVE	R/W	Exception Active Status Clear Bit Reserved for debug use. User must write 0 to this bit while writing to the register, otherwise the behavior will be unpredictable.	-
[0]	-	R	Reserved	-

SCR **0xE000ED10** System Control Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-			SEVONPEN D	-	SLEEPDEEP	SLEEPONEXI T	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:5]	-	R	Reserved	-
[4]	SEVONPEND	R/W	Send Event On Pending Bit When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event will be registered and then affect the next WFE. The processor also wakes up on the execution of an SEV instruction or an external event.	0: Only enabled interrupts or events can wake-up the processor. Disabled interrupts are excluded. 1: Enabled events and all interrupts, including disabled interrupts, can wake-up the processor.
[3]	-	R	Reserved	-
[2]	SLEEPDEEP	R/W	Processor Deep Sleep and Sleep Mode Selection SLEEPDEEP determines the processor uses sleep or deep sleep as its low power mode.	0: Sleep Mode 1: Deep Sleep Mode
[1]	SLEEPONEXIT	R/W	Sleep-On-Exit Enable This bit indicates sleep-on-exit while returning from Handler mode to Thread mode: Setting this bit as 1 enables an interrupt driven application to avoid the system returns to an empty main application.	0: Do not sleep when returning to Thread mode. 1: Enter Sleep, or Deep Sleep, on return from Handler mode to Thread mode.
[0]	-	R	Reserved	-

0xE000ED14									Configuration and Control Register	
Bit	31	30	29	28	27	26	25	24		
Function	-									
Reset Value	0	0	0	0	0	0	0	0		
Bit	23	22	21	20	19	18	17	16		
Function	-									
Reset Value	0	0	0	0	0	0	0	0		



GPCM3 SERIES PROGRAMMING GUIDE

Bit	15	14	13	12	11	10	9	8
Function				-			STKALIGN	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function			-		UNALIGN_TRP		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:10]	-	R	Reserved					-
[9]	STKALIGN	R	Stack Alignment This bit always reads as 1, and indicates 8-byte stack alignment on exception entry. On exception entry, the processor uses bit[9] of the stacked PSR to indicate the stack alignment. On return from the exception, it uses this stacked bit to restore the correct stack alignment.					-
[8:4]	-	R	Reserved					-
[3]	UNALIGN_TRP	R	Unaligned Access Traps This bit always reads as 1, and indicates that all unaligned accesses generate a HardFault.					-
[2:0]	-	R	Reserved					-

SHPR2								
Bit	31	30	29	28	27	26	25	24
Function	PRI_11[1:0]				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:30]	PRI_11[1:0]	R/W	Priority Of System Handler 11 – SVCall "0" denotes the highest priority and "3" denotes the lowest priority.					-
[29:0]	-	R	Reserved					-

SHPR3
0xE000ED20
System Handler Priority Register 3

Bit	31	30	29	28	27	26	25	24
Function	PRI_15[1:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	PRI_14[1:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:30]	PRI_15[1:0]	R/W	Priority Of System Handler 15 – SysTick "0" denotes the highest priority and "3" denotes the lowest priority.						-
[29:24]	-	R	Reserved						-
[23:22]	PRI_14[1:0]	R/W	Priority Of System Handler 14 – PendSV "0" denotes the highest priority and "3" denotes the lowest priority.						-
[21:0]	-	R	Reserved						-

SHCSR
0xE000ED24
System Handler Control and State Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	SVCALLPEN	DED	-					
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-

Bit	Function	Type	Description	Condition
[15]	SVCALLPENDED	R/W	SVCall Pending bit This bit reflects the pending state on a read, and updates the pending state to the value written on a write.	0: SVCall is not pending. 1: SVCall is pending.
[14:0]	-	R	Reserved	-

4.4.System timer, SysTick

The processor has a 24-bit system timer, SysTick, which counts down from the reloaded value to zero, reloads the value in the SYST_RVR register on the next clock cycle, and then counts down on the subsequent clock cycles. The interrupt controller clock updates the SysTick count. When processor clock is selected and the clock signal is stopped for low power mode, the SysTick counter stops. When external clock is selected, the clock continues to run in low power mode and SysTick can be used as a wakeup source. Please ensure that the software uses aligned word accesses to access the SysTick registers. If the SysTick counter reloads and the current value is undefined when reset happens, the correct initialization sequence for the SysTick counter is:

1. Program reload value.
2. Clear the current value.

Note: When the processor is halted for debugging, the counter does not decrement.

4.5.Register

Register Map

Name	Address	Description
SYST_CSR	0xE000E010	SysTick Control and Status Register
SYST_RVR	0xE000E014	SysTick Reload Value Register
SYST_CVR	0xE000E018	SysTick Current Value Register

Registers

0xE000E010									SysTick Control and Status Register	
Bit	31	30	29	28	27	26	25	24		
Function									-	
Reset Value	0	0	0	0	0	0	0	0		
Bit	23	22	21	20	19	18	17	16		
Function									-	
Reset Value	0	0	0	0	0	0	0	0	COUNTFLAG	
Bit	15	14	13	12	11	10	9	8		
Function									-	
Reset Value	0	0	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0		
Function									CLKSOURCE	TICKINT
Reset Value	0	0	0	0	0	0	0	0	ENABLE	

Bit	Function	Type	Description	Condition
[31:17]	-	--	Reserved	-
[16]	COUNTFLAG	R	COUNTFLAG will return to 1 when the timer counts to 0. COUNTFLAG is set as 1 by a count transition. COUNTFLAG is cleared on read or by a write to the Current Value register.	-
[15:3]	-	--	Reserved	-
[2]	CLKSOURCE	R/W	Clock Source	0: Clock source is optional (refer to STCLK_S). 1: Core clock will be used as SysTick if no external clock is provided. This bit will then read 1 and ignore writes.
[1]	TICKINT	R/W	SysTick Exception	0: Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine the following actions of the system after a count to zero occurs. 1: Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick Current Value register by a register write in software will not cause SysTick to be pended.
[0]	ENABLE	R/W	SysTick Counter Enable	0: Counter is disabled. 1: Counter will operate in a multi-shot manner.

SYST_RVR								0xE000E014								SysTick Reload Value Register							
Bit	31	30	29	28	27	26	25	24															
Function					-																		
Reset Value	0	0	0	0	0	0	0	0															

Bit	23	22	21	20	19	18	17	16
Function					RELOAD[23:16]			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	RELOAD[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	RELOAD[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:24]	-	--	Reserved					-
[23:0]	RELOAD[23:0]	R/W	Value to load into the Current Value register when the counter reaches 0.					-

SYST_CVR								
0xE000E018								
Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	CURRENT[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	CURRENT[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CURRENT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:24]	-	--	Reserved					-
[23:0]	CURRENT[23:0]	R/W	Current Counter Value This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register as 0. CURRENT doesn't support bits RAZ (see SysTick Reload Value register).					-

5. Nested Vectored Interrupt Controller (NVIC)

5.1. Overview

GPCM3 series provides an interrupt controller as an integral part of the exception model. The NVIC and the processor core interface are closely coupled, which enables low latency interrupt processing and efficient processing of late arriving interrupts.

5.2. Features

The NVIC supports the following features:

- 32 Interrupt Nodes
- 4 programmable priority levels for each interrupt node.
- Support interrupt tail-chaining and late-arrival.
- Support an external Non-maskable Interrupt (NMI).
- Software Interrupt Generation

5.3. Exception Types

The exception types are described in the table below:

Exception Types	Descriptions
Reset	Reset is invoked by power up or hardware/software reset. The exception model treats reset as a special form of exception. When reset is asserted, the operation of the processor stops, potentially at any point in an instruction. When reset is de-asserted, execution restarts from the address provided by the reset entry in the vector table. Execution restarts in Thread mode.
HardFault	A HardFault is an exception that occurs because of an error during normal or exception processing. HardFaults have a fixed priority of -1, meaning they have higher priority than any exception with configurable priority.
SVCall	A supervisor call (SVC) is an exception that is triggered by the SVC instruction. In an OS environment, applications can use SVC instructions to access OS kernel functions and device drivers.
PendSV	PendSV is an interrupt-driven request for system-level service. In an OS environment, use PendSV for context switching when no other exception is active.
SysTick	A SysTick exception is an exception the system timer generates when it reaches zero. Software can also generate a SysTick exception. In an OS environment, the processor can use this exception as system tick.
Interrupt (IRQ)	An interrupt, or IRQ, is an exception signaled by a peripheral, or generated by a software request. All interrupts are asynchronous to instruction execution. In the system, peripherals use interrupts to communicate with the processor.

Properties of the different exception types:

Exception Number	IRQ Number	Exception Type	Priority	Vector Address	Activation
1	-	Reset	-3	0x0000 0004	Asynchronous
2	-	NMI	-2	0x0000 0008	Asynchronous
3	-13	Hard Fault	-1	0x0000 000C	Synchronous
4 ~ 10	-	Reserved	-	-	-
11	-5	SVCall	Configurable	0x0000 002C	Synchronous
12 ~ 13	-	Reserved	-	-	-
14	-2	PendSV	Configurable	0x0000 0038	Asynchronous
15	-1	SysTick	Configurable	0x0000 003C	Asynchronous
16 and above	0 and above	Interrupt (IRQ)	Configurable	0x0000 0040 and above	Asynchronous

System Interrupt Vector Map

Exception Number	IRQ Number	Interrupt Name	Description	Vector Address
16	0	USB_IRQHandler	USB Interrupt	0x0000 0040
17	1	VKEY_IRQHandler	Velocity Key Interrupt	0x0000 0044
18	2	EXTI_IRQHandler	EXT0,1,2,3 Interrupt	0x0000 0048
19	3	MAC_IRQHandler	MAC Interrupt	0x0000 004C
20	4	QD_IRQHandler	Quadrature Decoder Interrupt	0x0000 0050
21	5	SAR_ADC_IRQHandler	SAR ADC Interrupt	0x0000 0054
22	6	DS_ADC_IRQHandler	Delta-Sigma ADC Interrupt (MIC)	0x0000 0058
23	7	DAC_CH0_IRQHandler	DAC CH0 Interrupt	0x0000 005C
24	8	DAC_CH1_IRQHandler	DAC CH1 Interrupt	0x0000 0060
25	9	CCP0_IRQHandler	CCU40 Interrupt	0x0000 0064
26	10	CCP1_IRQHandler	CCU41 Interrupt	0x0000 0068
27	11	CTS_TMO_IRQHandler	CTS TM0 Interrupt	0x0000 006C
28	12	CTS_TM1_IRQHandler	CTS TM1 Interrupt	0x0000 0070
29	13	TIMEBASE_IRQHandler	TIMEBASE Interrupt	0x0000 0074
30	14	I2C_IRQHandler	I2C Interrupt	0x0000 0078
31	15	SPU_IRQHandler	SPU Interrupt	0x0000 007C
32	16	UART0_IRQHandler	UART0 Interrupt	0x0000 0080
33	17	UART1_IRQHandler	UART1 Interrupt	0x0000 0084
34	18	I2S_IRQHandler	I2S Interrupt	0x0000 0088
35	19	SPI0_IRQHandler	SPI0 Interrupt	0x0000 008C
36	20	SPI1_IRQHandler	SPI1 Interrupt	0x0000 0090
37	21	DMA0_IRQHandler	DMA0 Interrupt	0x0000 0094
38	22	DMA1_IRQHandler	DMA1 Interrupt	0x0000 0098
39	23	DMA2_IRQHandler	DMA2 Interrupt	0x0000 009C
40	24	DMA3_IRQHandler	DMA3 Interrupt	0x0000 00A0
41	25	DMA4_IRQHandler	DMA4 Interrupt	0x0000 00A4
42	26	TIMER0_IRQHandler	Timer0 Interrupt	0x0000 00A8
43	27	TIMER1_IRQHandler	Timer1 Interrupt	0x0000 00AC
44	28	TIMER2_IRQHandler	Timer2 Interrupt	0x0000 00B0
45	29	KEYCHG_IRQHandler	KEY_CHANGE Interrupt	0x0000 00B4
46	30	PDM_IRQHandler	PDM Interrupt	0x0000 00B8
47	31	SPUBeat_IRQHandler	SPU Beat Interrupt	0x0000 00BC

6. SYSTEM Descriptions

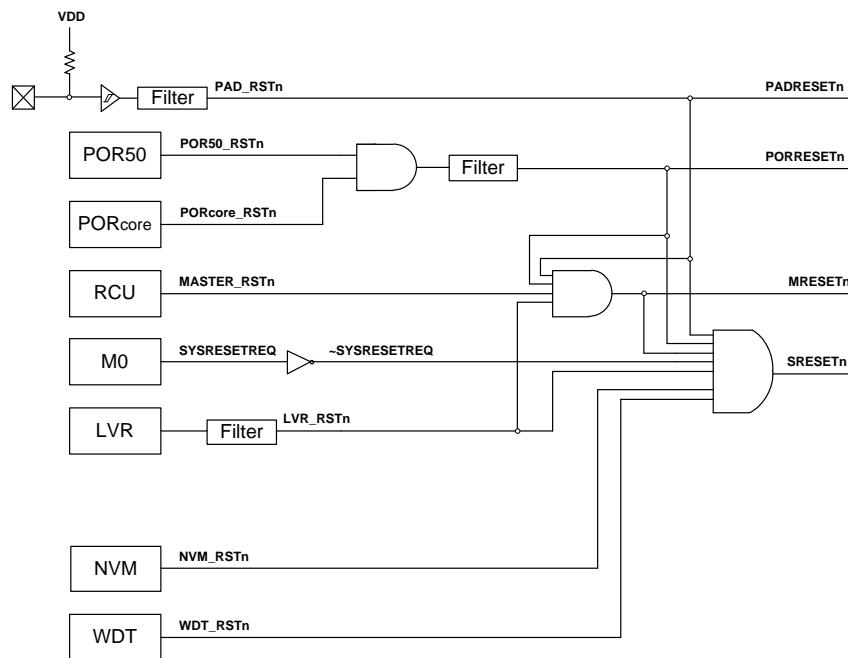
6.1. Reset System

6.1.1. Introduction

The GPCM3 Series has the following reset types for the system:

- PAD Reset (PADRESETn)
- Power on Reset (PORRESETn)
- Master Reset (MRESETn)
- System Reset (SRESETn)

6.1.2. Block Diagram



5.1 Reset Structure

6.1.3. Function

PADRESETn

PADRESETn is triggered by IO PAD Resetn and is active at Low State. In GPCM3, there's a built-in filter to avoid IO denounce. Generally, to make PADRESETn happen, user needs to keep IO PAD Resetn in zero status about 50 us.

PORRESETn

PORRESETn is triggered by GPCM3 power on. GPCM3 power on means to provide power supply to IO VDD5V, and it will trigger PORRESETn.

Master Reset

A complete reset of the whole chip is executed by a master reset. Master reset is composed of PADRESETn, PORRESETn, MASTER_RSTn and LVR_RSTn. Low voltage reset (LVR_RSTn), also known as brown-out reset, is asserted whenever the voltage falls below reset thresholds.

In addition, a **MASTER_RSTn** can be triggered by setting **RCU_CTRL.MRSTn** bit[0] as 0, and it will reset the whole chip.

System Reset

A system reset affects almost all logics but RCU registers and Debug system are exceptions if there's G+Link pro (debug link). The debug system is reset by System Reset in normal operation mode when debug link is not present. As long as there's G+Link pro (debug link), the Debug system will not be affected by System Reset.

6.1.4. Register Description

Register Map

Name	Address	Description
RCU_CTRL	0x50003000	Reset Control Register
RCU_STS	0x50003004	Reset Status Register

Register Function

RCU_CTRL									0x50003000	Reset Control Register
Bit	31	30	29	28	27	26	25	24		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	23	22	21	20	19	18	17	16		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0		
Function									MRST_EN_TRIGGER	
Reset Value	0	0	0	0	0	0	0	0		

Bit	Function	Type	Description						Condition
[31:1]	-	R	Reserved						-
[0]	MRST_EN_TRIGGER	R/W	Master Reset Trigger (Write Protected) Note: This bit is written-protected, please refer to SMU.SysUnLock to unlock. It will automatically return to 1 after programming 0 to MRSTn.						0: Trigger the Master Reset. 1: No effect.

RCU_STS									0x50003004	Reset Status Register
Bit	31	30	29	28	27	26	25	24		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	23	22	21	20	19	18	17	16		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0		
Function	-	SYS_WDT_FLAG	-	LVR_RST_FL AG	SYS_RST_FL AG	SOFT_RST_FL AG	KEY_RST_FL AG	POR_FLAG		
Reset Value	0	0	0	0	0	0	0	0		

Bit	Function	Type	Description	Condition
[31:7]	-	-	Reserved	-
[6]	SYS_WDT_FLAG	R/W	Watchdog Reset Flag Note: Write 1 to clear this bit.	0: No reset. 1: Watchdog reset occurs.
[5]	-	-	Reserved	-
[4]	LVR_RST_FLAG	R/W	Low Voltage Reset Flag Note: Write 1 to clear this bit.	0: No reset. 1: LVR reset occurs.
[3]	SYS_RST_FLAG	R/W	System Reset Flag Note1: Write 1 to clear this bit. Note2: While CPU is executing program, any attempt such as executing an unknown opcode, accessing an incorrect bus interface or memory location, trying to change ARM CPU's state, or other violations of programming logic may cause a hard fault and then trigger system reset.	0: No reset. 1: System reset occurs.
[2]	SOFT_RST_FLAG	R/W	Software Triggered Master Reset Flag Note: Write 1 to clear this bit.	0: No reset. 1: Software reset occurs.
[1]	KEY_RST_FLAG	R/W	Hardware Key Reset Flag Note: Write 1 to clear this bit.	0: No reset. 1: H/W Key reset occurs.
[0]	POR_FLAG	R/W	Power-On Reset Flag Note: Write 1 to clear this bit.	0: No reset. 1: POR occurs.

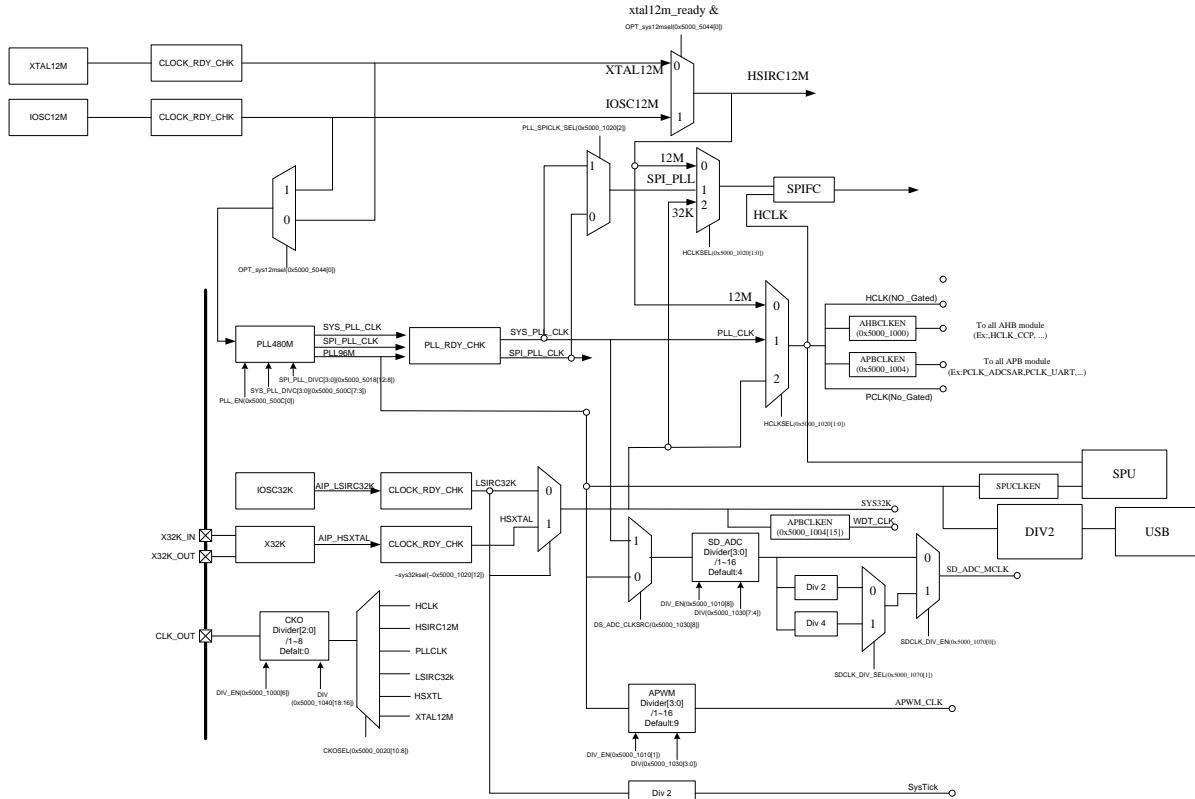
6.2.Clock Control Unit**6.2.1. Introduction**

The clock control unit has four clock sources:

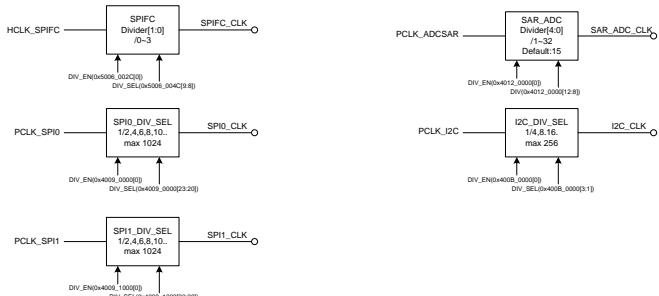
- Internal Low Speed RC Oscillator 32 KHz
- External Low Speed Crystal Oscillator 32KHz
- High Speed Internal RC Oscillator 12 MHz
- External High Speed Crystal Oscillator 12MHz
- PLL Up to 120 MHz

6.2.2. Block Diagram

Root Clock



Leaf Clock



Efuse Read Set IOOSC12M Trim	Read SPI & Set Option with IOOSC12M	Option Select XTAL12M Wait XTAL12M Ready	SYSCLK switch to XTAL12M	PLL Enable PLL 12MCKIN = XTAL12M PLL ready Counter by XTAL12M
3us	?	6us ready		84us ready

6.2.3. Register Description

Register Map

Name	Address	Description
CLOCK_AHBCKEN	0x50001000	AHB Peripherals Clock Enable Register
CLOCK_APBCKEN	0x50001010	APB Peripherals Clock Enable Register

Name	Address	Description
CLOCK_AHBCKSEL	0x50001020	AHB Peripherals Clock Selection Register
CLOCK_CLKDIV1	0x50001030	Analog Macro Clock Divider Register
CLOCK_CLKSTS	0x50001050	Clock Status Register
CLOCK_SWTRIM	0x50001060	Clock Software Trim
CLOCK_CLKDIV3	0x50001070	Clock Sigma-Delta Clock Divider Control

Register Function

CLOCK_AHBCKEN								
0x50001000								
Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	CPU_CLK_ENABLE	SYSTICK_CLK_ENABLE	ANALOG_CLK_ENABLE	INT_CLK_ENABLE	RAM_CLK_ENABLE	APB_CLK_ENABLE	AHBBUS_CLK_ENABLE	FLASH_CLK_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	GPIO_CLK_ENABLE	-	SPU_CLK_ENABLE	SPIFC_CLK_ENABLE	MAC_CLK_ENABLE	CCP1_CLK_ENABLE	CCPO_CLK_ENABLE	DMA_CLK_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15]	CPU_CLK_ENABLE	R/W	CPU Clock Enabling Bit	0: Disabled 1: Enabled
[14]	SYSTICK_CLK_ENABLE	R/W	SysTick External Clock Enabling Bit	0: Disabled 1: Enabled
[13]	ANALOG_CLK_ENABLE	R/W	Analog Control Clock Enabling Bit	0: Disabled 1: Enabled
[12]	INT_CLK_ENABLE	R/W	Interrupt Control Clock Enabling Bit	0: Disabled 1: Enabled
[11]	RAM_CLK_ENABLE	R/W	SRAM Control Clock Enabling Bit	0: Disabled 1: Enabled
[10]	APB_CLK_ENABLE	R/W	APB BUS System Clock Enabling Bit	0: Disabled 1: Enabled
[9]	AHBBUS_CLK_ENABLE	R/W	AHB BUS System Clock Enabling Bit	0: Disabled 1: Enabled
[8]	FLASH_CLK_ENABLE	R/W	EFLASH Control Clock Enabling Bit	0: Disabled 1: Enabled

Bit	Function	Type	Description				Condition
[7]	GPIO_CLK_ENABLE	R/W	GPIO Unit Enabling Bit				0: Disabled 1: Enabled
[6]	-	R	Reserved				-
[5]	SPU_CLK_ENABLE	R/W	SPU Clock Enabling Bit				0: Disabled 1: Enabled
[4]	SPIFC_CLK_ENABLE	R/W	SPIFC Clock Enabling Bit				0: Disabled 1: Enabled
[3]	MAC_CLK_ENABLE	R/W	MAC Clock Enabling Bit				0: Disabled 1: Enabled
[2]	CCP1_CLK_ENABLE	R/W	Capture and Compare Unit1 Clock Enabling Bit				0: Disabled 1: Enabled
[1]	CCPO_CLK_ENABLE	R/W	Capture and Compare Unit0 Clock Enabling Bit				0: Disabled 1: Enabled
[0]	DMA_CLK_ENABLE	R/W	DMA Clock Enabling Bit				0: Disabled 1: Enabled

CLOCK_APBCKEN **0x50001010** **APB Peripherals Clock Enable Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	WDG_CLK_ENABLE	TB_CLK_ENABLE	UART1_CLK_ENABLE	USB_CLK_ENABLE	AUDPWM_CLK_ENABLE	CTS_CLK_ENABLE	I2S_CLK_ENABLE	DSADC_CLK_ENABLE
Reset Value	0	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SARADC_CLK_ENABLE	SPI1_CLK_ENABLE	SPI0_CLK_ENABLE	UART0_CLK_ENABLE	I2C_CLK_ENABLE	TIMER_CLK_ENABLE	DAC_CLK_ENABLE	PWMIO_CLK_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	-	R	Reserved				-
[15]	WDG_CLK_ENABLE	R/W	Watchdog Clock Enabling Bit				0: Disabled 1: Enabled
[14]	TB_CLK_ENABLE	R/W	Timebase Clock Enabling Bit				0: Disabled 1: Enabled

Bit	Function	Type	Description	Condition
[13]	UART1_CLK_ENABLE	R/W	UART Control Clock Enabling Bit	0: Disabled 1: Enabled
[12]	USB_CLK_ENABLE	R/W	USB Clock Enabling Bit	0: Disabled 1: Enabled
[11]	AUDPWM_CLK_ENABLE	R/W	Audio PWM Control Clock Enabling Bit	0: Disabled 1: Enabled
[10]	CTS_CLK_ENABLE	R/W	CTS Control Clock Enabling Bit	0: Disabled 1: Enabled
[9]	I2S_CLK_ENABLE	R/W	I2S Control Clock Enabling Bit	0: Disabled 1: Enabled
[8]	DSADC_CLK_ENABLE	R/W	Delta Sigma ADC Clock Enabling Bit	0: Disabled 1: Enabled
[7]	SARADC_CLK_ENABLE	R/W	SAR ADC Control Clock Enabling Bit	0: Disabled 1: Enabled
[6]	SPI1_CLK_ENABLE	R/W	SPI1 Control Clock Enabling Bit	0: Disabled 1: Enabled
[5]	SPIO_CLK_ENABLE	R/W	SPI0 Control Clock Enabling Bit	0: Disabled 1: Enabled
[4]	UART0_CLK_ENABLE	R/W	UART0 Control Clock Enabling Bit	0: Disabled 1: Enabled
[3]	I2C_CLK_ENABLE	R/W	I2C Control Clock Enabling Bit	0: Disabled 1: Enabled
[2]	TIMER_CLK_ENABLE	R/W	Timer Control Clock Enabling Bit	0: Disabled 1: Enabled
[1]	DAC_CLK_ENABLE	R/W	DAC Control Clock Enabling Bit	0: Disabled 1: Enabled
[0]	PWMIO_CLK_ENABLE/ QD_CLK_ENABLE	R/W	Dual Functions: 1. PWMIO Control Clock Enabling Bit 2. QD Control Clock Enabling Bit	0: Disabled 1: Enabled

CLOCK_AHBCKSEL
0x50001020
AHB Peripherals Clock Selection Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	
Function	-				SYS32K_SEL	-			
Reset Value	0	0	0	1	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Function	-					PLL_SPICLK_SEL	HCLK_SEL[1:0]	
Reset Value	0	0	0	0	0	1	0	0

Bit	Function	Type	Description					Condition
[31:13]	-	R	Reserved					-
[12]	SYS32K_SEL	R/W	SYS32K Selection					0: XTAL32K 1: IOSC32K
[11:3]	-	R	Reserved					-
[2]	PLL_SPICLK_SEL	R/W	SPI Clock Source Selection					0: PLLSYS (Max: 120MHz) 1: PLLSPI (Max: 96MHz)
[1:0]	HCLK_SEL[1:0]	R/W	HCLK Source Selection Note: This bit is protected by SMU_SYSLOCK .					00: OSC 12M 01: PLL Clock 10: System 32KHz 11: Reserved

CLOCK_CLKDIV1									0x50001030	Analog Macro Clock Divider Register	
Bit	31	30	29	28	27	26	25	24			
Function									-		
Reset Value	0	0	0	0	0	0	0	0			

Bit	23	22	21	20	19	18	17	16
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0	
Function	DSADC_CLK_DIV[3:0]					AUDCLK_SEL[3:0]			
Reset Value	0	0	0	0	0	0	0	1	

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8]	DSADC_CLK_SRC	R/W	DSADC Clock Source					0: PLL 96MHz 1: PLL SYS
[7:4]	DSADC_CLK_DIV[3:0]	R/W	Codec ADC Output Clock Divider Output Clock = (MUX Output Clock) / (DSADC_CLK_DIV + 1)					0x0 = DSADC_CLK_DIV_1 0x1 = DSADC_CLK_DIV_2 0x2 = DSADC_CLK_DIV_3 0x3 = DSADC_CLK_DIV_4 0x4 = DSADC_CLK_DIV_5 0x5 = DSADC_CLK_DIV_6 0x6 = DSADC_CLK_DIV_7 0x7 = DSADC_CLK_DIV_8

Bit	Function	Type	Description	Condition
				0x8 = DSADC_CLK_DIV_9 0x9 = DSADC_CLK_DIV_10 0xA = DSADC_CLK_DIV_11 0xB = DSADC_CLK_DIV_12 0xC = DSADC_CLK_DIV_13 0xD = DSADC_CLK_DIV_14 0xE = DSADC_CLK_DIV_15 0xF = DSADC_CLK_DIV_16
[3:0]	AUDCLK_SEL[3:0]	R/W	Audio PWM Output Clock Divider Clock Output Clock = (MUX Output Clock) / (AUDCLK_SEL + 1)	0x0 = SYSCLK_DIV_1 0x1 = SYSCLK_DIV_2 0x2 = SYSCLK_DIV_3 0x3 = SYSCLK_DIV_4 0x4 = SYSCLK_DIV_5 0x5 = SYSCLK_DIV_6 0x6 = SYSCLK_DIV_7 0x7 = SYSCLK_DIV_8 0x8 = SYSCLK_DIV_9 0x9 = SYSCLK_DIV_10 0xA = SYSCLK_DIV_11 0xB = SYSCLK_DIV_12 0xC = SYSCLK_DIV_13 0xD = SYSCLK_DIV_14 0xE = SYSCLK_DIV_15 0xF = SYSCLK_DIV_16

CLOCK_CLKSTS 0x50001050 Clock Status Register								
Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	-			XTAL12M_R DY	XTAL32K_R DY	PLL_RDY	IOSC32K_R DY	IOSC12M_R DY
Reset Value	0	0	0	0	0	0	0	0
Bit	Function	Type	Description				Condition	
[31:5]	-	R	Reserved				-	

Bit	Function	Type	Description	Condition
[4]	XTAL12M_RDY	R	XTAL12MHz Ready Flag	0: XTAL12M is not ready. 1: XTAL12M is ready.
[3]	XTAL32K_RDY	R	External 32768Hz Crystal Oscillator Ready Flag	0: XTAL32K is not ready. 1: XTAL32K is ready.
[2]	PLL_RDY	R	PLL Ready Flag	0: PLL is not ready. 1: PLL is ready.
[1]	IOSC32K_RDY	R	Low Speed Internal 32768Hz RC Ready Flag	0: losc32K is not ready. 1: losc32K is ready.
[0]	IOSC12M_RDY	R	High Speed Internal 12MHz RC Ready Flag	0: losc12M is not ready. 1: losc12M is ready.

CLOCK_SWTRIM **0x50001060** **Software Trim for IOSC12M and IOSC32K**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	Counter[12:5]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	Counter[4:0]					RELOAD_EFUSE	COUNTER_RDY	SWTRIM_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:3]	Counter[12:0]	R	SW Trim Counter Value User can calculate Trim Value by reading Counter Value.	-
[2]	RELOAD_EFUSE	R/W	Writing 1 to this bit can reload IOSC12M Trim Value in Efuse to 0x50005008 bit[14:8].	Write 0: No effect. Write 1: Reload IOSC12M Trim Value.
[1]	COUNTER_RDY	R	Trim Counter Ready Flag	0: SW Trim is not Ready. 1: SW Trim is Ready.
[0]	SWTRIM_ENABLE	R/W	SW Trim Enable	0: SW Trim is disabled. 1: SW Trim is enabled.

CLOCK_CLKDIV3
0x50001070

Clock Sigma-Delta Clock Divider Control

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-			DSADC_CLK_DIV[1:0]	
Reset Value	0	0	0	0	0	0	1	1

Bit	Function	Type	Description	Condition
[31:2]	-	R	Reserved	-
[1:0]	DSADC_CLK_DIV [1:0]	R/W	Codec ADC Clock Out Divider Selection	00: Codec ADC Clock DIV1 01: Codec ADC Clock DIV2 10: Reserved 11: Codec ADC Clock DIV4

6.3.Low Power Mode & Shut Down Mode

6.3.1. Introduction

By default, the GPCM3 Serials is in normal mode after a system or a master reset. Several low power modes are available to save power when the CPU does not need to be kept running. For example, the case when it's waiting for an external event.

The GPCM3 Serials has three sleep modes and a shut down mode to reduce power consumption:

- Sleep Mode: Only the CPU stopped, and system clock is on.
- Deep Sleep Mode: All clocks are stopped but core power is on.
- Halt Mode: In deep sleep mode but the 32768Hz RTC clock is on.
- Shut Down Mode: Core power and all clocks are stopped.

6.3.2. Function

Entering Sleep Mode

The system can enter Sleep mode by executing the Wait For Interrupt (WFI) or Wait for Event (WFE) instructions.

Note: Before the system enters sleep mode, **do not** issue a Cache OFF command since the system might be unable to enter sleep mode.

Note: If all wakeup sources (ITU->EXTEN & ITU->EXTIEN) are disabled, the system will not wake up after it enters Sleep or Deep Sleep mode.

Wait for Interrupt

The wait for interrupt instruction, WFI, causes the immediate entry to sleep mode. When the processor executes a WFI instruction, it will stop executing instructions and enter sleep mode.

Wait for Event

Issue a WFE (Wait For Event) instruction to make the system enter Sleep mode.

Note: WFE must be issued twice to ensure the system enters sleep mode successfully.

```
DrvUart_SendString("EXT Event Sleep mode test.\r\n");
__WFE();           // Sleep Mode
__WFE();           // Sleep Mode
```

Note: If the corresponding Event Flag = 1 and it's not cleared before the system enters Sleep/Deep Sleep mode, the system will wake up immediately when WFE instruction is issued and won't enter Sleep/Deep Sleep mode successfully.

Sleep-on-exit

If the SLEEPONEXIT bit of the SCR is set as 1, when the processor completes the execution of an exception handler and returns to Thread mode, it will immediately enter sleep mode. This mechanism is used in applications that only require the processor to be working when an interrupt occurs.

Low Power Mode Summary

Mode	Entry	Wakeup	Clocks	Regulator Voltage	I/O State	
Sleep	WFI	Any Interrupt	CPU clock is OFF. No effect on other clocks or analog clock sources.	Keep ON and in normal mode voltage.	Keep the setting before entering low power modes.	
	WFE	Any Event				
Deep Sleep	WFI+SLEEPDEEP=1	External Input Interrupt	All clocks are OFF.	Keep ON but output voltage changes to Deep Sleep mode voltage.		
	WFE+SLEEPDEEP=1	External Input Event				

Entering Shut Down Mode

(a) **Shut Down Mode supports two wakeup sources:**

- (1) When SMU_SD_Ctrl[0] = 0, CPU wakeup can be implemented via key change wakeup source but only 16* IO(IOA[7:0] and [31:24]) can be the key-change wakeup source. **Note:** To wake up CPU successfully by the key-change wakeup source , user must latch the corresponding IO first before the system enters shut down mode.
- (2) When SMU_SD_Ctrl[0] = 1, CPU wakeup can be implemented via losc16KHz. After CPU enters shut down mode, the 16KHz Counter will start counting. When it reaches 8192 clocks, wakeup will be triggered. Note that the frequency of this clock is not very accurate, and it takes approximately 0.25 ~ 0.75 seconds to wake up CPU.

(b) **Shut down mode ON/OFF:**

- (1) Write 0x55 to SMU->SHUTDOWN_ON (0x50000058) to enter Shut down mode. **Note:** It must co-work with the Deep sleep command to make CPU enter Shut down mode.

```

READ_REG(GPIOA->IDATA);           // IOA status latched.
MODIFY_REG(SMU->SHUTDOWN_ON, 0xFF, 0x55); // Shut down mode enabled 0x50000058 = 0x55
SCB->SCR = (1UL << 2);          // Entry the Deep Sleep Mode
__WFI();

```

- (2) Shut Down Flag: After user writes 0x55 to SHUTDOWN_ON Register and issues a Deep sleep command (SCB->SCR = (1UL << 2)), the system will enter Shut down mode. At this point, Shut down flag = 1 (SMU_SD_Ctrl[31] = 1).
- (3) When CPU is woke up from Shut down mode, it will start executing the boot code. After SMU->SHUTDOWN_OFF(0x5000005C) is executed and user writes 0xAA command, the system will exit from Shut down mode. Here, Shut down flag = 0.

```
MODIFY_REG(SMU->SHUTDOWN_OFF, 0xFF, 0xAA);           // shut down mode released 0x5000005C = 0xAA
```

6.4.System Management Unit

6.4.1. Introduction

The system management unit includes user defined ID information, instruction fetch accelerator and some control setting.

6.4.2. Register Description

Register Map

Name	Address	Description
SMU_SYSLOCK	0x50000000	System Control Signal Lock Register
SMU_CACHE_CTRL	0x50000030	Cache Control Register
SMU_SHUTDOWN_ON	0x50000058	Shutdown Enable Control
SMU_SHUTDOWN_OFF	0x5000005C	Shutdown Disable Control
SMU_SHUTDOWN_CTRL	0x50000060	Shutdown Mode Control

Registers

SMU_SYSLOCK								0x50000000	System Control Signal Lock Register								
Bit	31	30	29	28	27	26	25	24									
Function	-																
Reset Value	0	0	0	0	0	0	0	0									

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							SYSLOCK_U NLOCK_STS
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SYSLOCK_UNLOCK_KEY[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:9]	-	R	Reserved	-
[8]	SYSLOCK_UNLOCK_STS	R	UnLOCK Status Flag Note: This bit status is based on UnLOCK_KEY operation.	0: Locked 1: UnLocked
[7:0]	SYSLOCK_UNLOCK_KEY [7:0]	R/W	Specified System Control UnLock Key Write 0xAB and 0x12 to this UNLOCK_KEY sequentially, and the specified registers will be unlocked. Note: If SysUnLOCK is in unlocked state and user writes values other than 0x12 to UnLOCK_KEY, the specified registers will be locked. Specified Lock Protect Register are listed as below: RCU_CTRL ANALOG Control Register	-

SMU_CACHE_CTRL **0x50000030** **Cache Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							CACHE_ENA BLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:1]	-	R	Reserved					-
[0]	CACHE_ENABLE	R/W	Cache Function Enable					0: Disabled 1: Enabled

SMU_SHUTDOWN_ON **0x50000058** **Shutdown Enable Control**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SHUTDOWN_ON[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R/W	Reserved					-
[7:0]	SHUTDOWN_ON	R/W	Shutdown Enable Control With Deep sleep instruction (SCB->SCR = (1UL << 2)), Shut down mode will be enabled when user writes 0x55 to SHUTDOWN_ON.					

SMU_SHUTDOWN_OFF **0x5000005C** **Shutdown Disable Control**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SHUTDOWN_OFF[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R/W	Reserved						-
[7:0]	SHUTDOWN_OFF	R/W	Shutdown Disable Control Shutdown mode will be disabled when user writes 0xAA to SHUTDOWN_OFF.						

SMU_SHUTDOWN_CTRL								
0x50000060								
Bit	31	30	29	28	27	26	25	24
Function	SD_FLAG				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							WK_SRC
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31]	SD_FLAG	R/W	System Shut Down Flag						0: Not shut down mode. 1: Shut Down Mode
[30:1]	-	R/W	Reserved						-
[0]	WK_SRC	R/W	Shutdown Wakeup Source Selection						0: Wakeup by key change. 1: Wakeup by internal 16KHz clock.

6.5. Interrupt Control Unit

6.5.1. Introduction

Interrupt control unit includes NMI (non-maskable interrupt) interrupt source configuration, software interrupt request setting and external interrupt/event control. The external interrupt/event controller consists of up to 4 edge detectors in connectivity line devices. Each input line can be independently configured to select the trigger event (rising, falling, both or level). When event control is enabled and an event happens, an event pulse will be generated.

6.5.2. Function

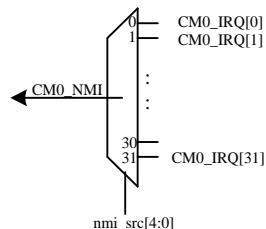
To trigger the interrupt, the interrupt line should be configured and enabled. This is done by programming the trigger registers with the desired edge detection or level detection. User can enable the interrupt request by writing a '1' to the corresponding bit in the interrupt enable register. When the selected detection condition occurs on the external interrupt line, an interrupt request is generated.

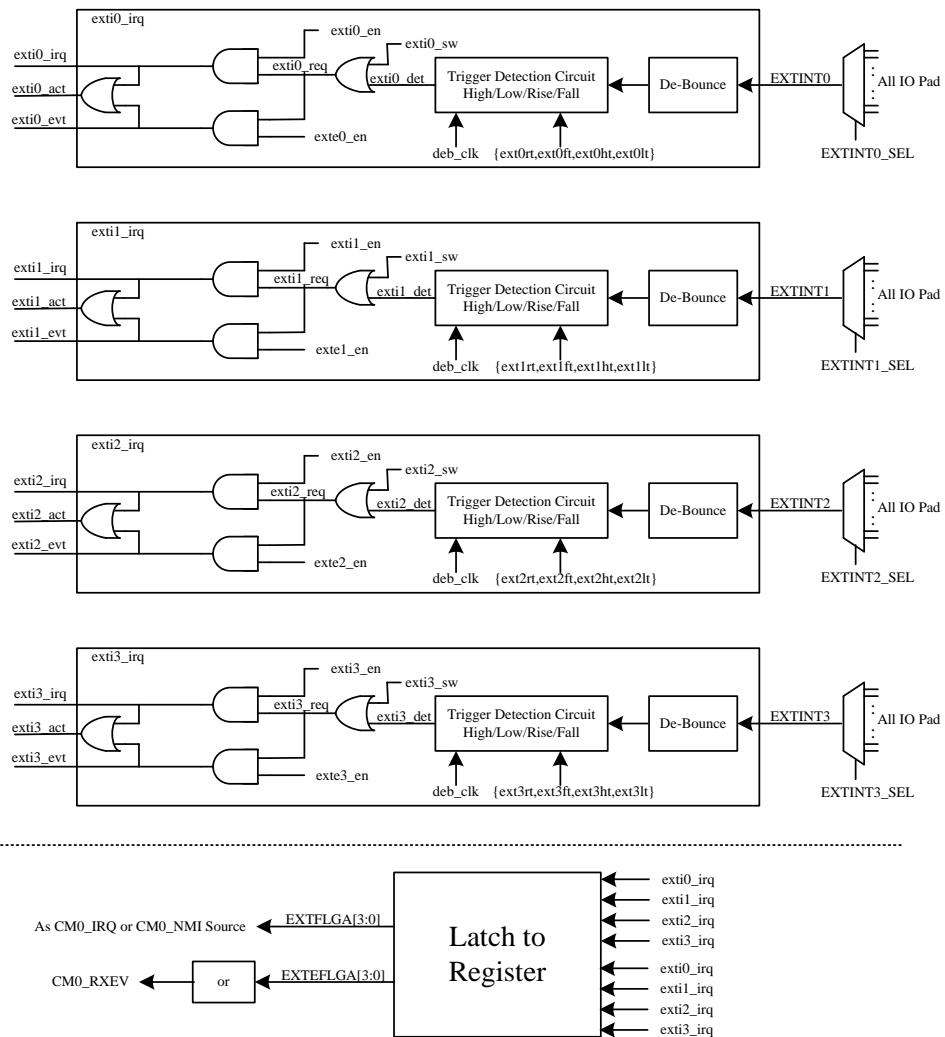
To trigger the event, the event line should be configured and enabled. This is done by programming the trigger registers with the desired edge detection or level detection. User can enable the event request by writing a '1' to the corresponding bit in the event enable register. When the selected detection condition occurs on the event line, an event pulse is generated.

An interrupt/event request can also be generated by the software writing a '1' in the software interrupt/event register.

NMI interrupts Source Selection

NMI interrupt source of GPCM3 SERIALS can be configured as one of NVIC IRQ Numbers. The NMISRC correspond to NVIC IRQ Numbers.



External Interrupt/Event Controller Block Diagram

Wakeup Event Management

The GPCM3Series is able to handle external or internal events in order to wake up the core (WFE). The wakeup event can be generated by user configures an external or internal EXTI line in event mode. When the CPU resumes from WFE, it is not necessary to clear the peripheral interrupt pending bit or the NVIC IRQ channel pending bit since the pending bit corresponding to the event line is not set as 1.

6.5.3. Register Description
Register Map

Name	Address	Description
ITU_NMICTRL	0x50004000	NMI Interrupt Control Register
ITU_SWIRQ	0x50004004	Software Interrupt Request Register
ITU_EXTIEN	0x50004010	External Interrupt Enable Register
ITU_EXTEEN	0x50004014	External Event Enable Register
ITU_EXTRHT	0x50004018	External Rising Edge/High Level Trigger Register

Name	Address	Description
ITU_EXTFLT	0x50004020	External Falling Edge/Low Level Trigger Register
ITU_EXTIFLG	0x50004028	External Interrupt Flag Register
ITU_EXTEFLG	0x5000402C	External Event Flag Register

Registers

ITU_NMICTRL								0x50004000	NMI Interrupt Control Register	
Bit	31	30	29	28	27	26	25	24		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	23	22	21	20	19	18	17	16		
Function					-				INT_ENABLE	
Reset Value	0	0	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8		
Function					-					
Reset Value	0	0	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0		
Function			-			SRC_SEL[4:0]				
Reset Value	0	0	0	0	0	0	0	0		

Bit	Function	Type	Description	Condition
[31:17]	-	R	Reserved	-
[16]	INT_ENABLE	R/W	NMI Interrupt Enabling Bit	0: Disabled 1: Enabled
[15:5]	-	R	Reserved	-
[4:0]	SRC_SEL[4:0]	R/W	NMI Source Selection For NMI Source numbers, please refer to IRQ number of System Interrupt Vector Map.	0x1F: BEATspu 0x1E: PDM 0x1D: KEYCHG 0x1C: TM2 0x1B: TM1 0x1A: TM0 0x19: DMA4 0x18: DMA3 0x17: DMA2 0x16: DMA1 0x15: DMA0 0x14: SPI1 0x13: SPI0 0x12: I2S 0x11: UART1 0x10: UART0 0x0F: SPU

Bit	Function	Type	Description	Condition
				0x0E: I2C 0x0D: TIMEBASE 0x0C: CTS_TMA1 0x0B: CTS_TMA0 0x0A: CCP1 0x09: CCP0 0x08: DAC_CH1 0x07: DAC_CH0 0x06: DS_ADC 0x05: SAR_ADC 0x04: QD 0x03: MAC 0x02: EXT 0X01: V-Key 0X00: USB

ITU_SWIRQ **0x50004004** **Software Interrupt Request Register**

Bit	31	30	29	28	27	26	25	24
Function	IRQ[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IRQ[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IRQ[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IRQ[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	IRQ[31:0]	R/W	Interrupt Request When the IRQ[n] is 0, setting IRQ[n] as 1 will generate an interrupt to Cortex™-M0 NVIC[n]. When the IRQ[n] is 1, which means an interrupt is asserted, setting IRQ[n] as 1 again will clear the interrupt while setting IRQ[n] as 0 has no effect. Note: Please refer to ICSR. VECTACTIVE to check the currently active exception.	-

ITU_EXTIEN **0x50004010** **External Interrupt Enable Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-				EXT3_INT_ENABLE	EXT2_INT_ENABLE	EXT1_INT_ENABLE	EXT0_INT_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:4]	-	R	Reserved					-
[3]	EXT3_INT_ENABLE	R/W	External Input 3 Interrupt Enable The bit number of EXTIEN means the corresponding port.					0: Disabled 1: Enabled
[2]	EXT2_INT_ENABLE	R/W	External Input 2 Interrupt Enable The bit number of EXTIEN means the corresponding port.					0: Disabled 1: Enabled
[1]	EXT1_INT_ENABLE	R/W	External Input 1 Interrupt Enable The bit number of EXTIEN means the corresponding port.					0: Disabled 1: Enabled
[0]	EXT0_INT_ENABLE	R/W	External Input 0 Interrupt Enable The bit number of EXTIEN means the corresponding port.					0: Disabled 1: Enabled

ITU_EXTEEN **0x50004014** **External Event Enable Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-			KEYCHG_EVT_ENABLE	EXT3_EVT_ENABLE	EXT2_EVT_ENABLE	EXT1_EVT_ENABLE	EXT0_EVT_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:5]	-	R	Reserved				-
[4]	KEYCHG_EVT_ENABLE	R/W	Key Change Event Enable				-
[3]	EXT3_EVT_ENABLE	R/W	External Input 3 Event Enable				0: Disabled 1: Enabled
[2]	EXT2_EVT_ENABLE	R/W	External Input 2 Event Enable				0: Disabled 1: Enabled
[1]	EXT1_EVT_ENABLE	R/W	External Input 1 Event Enable				0: Disabled 1: Enabled
[0]	EXT0_EVT_ENABLE	R/W	External Input 0 Event Enable				0: Disabled 1: Enabled

ITU_EXTRHT
0x50004018
External Input Rising Edge/High Level Trigger Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-				EXT3_HLVL_TRIG_ENABLE	EXT2_HLVL_TRIG_ENABLE	EXT1_HLVL_TRIG_ENABLE	EXT0_HLVL_TRIG_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-				EXT3_RISING_TRIG_ENABLE	EXT2_RISING_TRIG_ENABLE	EXT1_RISING_TRIG_ENABLE	EXT0_RISING_TRIG_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:20]	-	R	Reserved				-
[19]	EXT3_HLVL_TRIG_ENABLE	R/W	External Input 3 High Level Trigger Enable				0: Disabled 1: Enabled
[18]	EXT2_HLVL_TRIG_ENABLE	R/W	External Input 2 High Level Trigger Enable				0: Disabled 1: Enabled
[17]	EXT1_HLVL_TRIG_ENABLE	R/W	External Input 1 High Level Trigger Enable				0: Disabled 1: Enabled

Bit	Function	Type	Description		Condition
[16]	EXT0_HLVL_TRIG_ENABLE	R/W	External Input 0 High Level Trigger Enable		0: Disabled 1: Enabled
[15:4]	-	R	Reserved		-
[3]	EXT3_RISING_TRIG_ENABLE	R/W	External Input 3 Rising Trigger Enable		0: Disabled 1: Enabled
[2]	EXT2_RISING_TRIG_ENABLE	R/W	External Input 2 Rising Trigger Enable		0: Disabled 1: Enabled
[1]	EXT1_RISING_TRIG_ENABLE	R/W	External Input 1 Rising Trigger Enable		0: Disabled 1: Enabled
[0]	EXT0_RISING_TRIG_ENABLE	R/W	External Input 0 Rising Trigger Enable		0: Disabled 1: Enabled

ITU_EXTFLT **0x50004020** **External Falling Edge/Low Level Trigger Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-	EXT3_LLVL_TRIG_ENABLE	EXT2_LLVL_TRIG_ENABLE	EXT1_LLVL_TRIG_ENABLE	EXT0_LLVL_TRIG_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-	EXT3_FALLING_TRIG_ENABLE	EXT2_FALLING_TRIG_ENABLE	EXT1_FALLING_TRIG_ENABLE	EXT0_FALLING_TRIG_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description		Condition
[31:20]	-	R	Reserved		-
[19]	EXT3_LLVL_TRIG_ENABLE	R/W	External Input 3 Low Level Trigger Enable		0: Disabled 1: Enabled
[18]	EXT2_LLVL_TRIG_ENABLE	R/W	External Input 2 Low Level Trigger Enable		0: Disabled 1: Enabled
[17]	EXT1_LLVL_TRIG_ENABLE	R/W	External Input 1 Low Level Trigger Enable		0: Disabled 1: Enabled
[16]	EXT0_LLVL_TRIG_ENABLE	R/W	External Input 0 Low Level Trigger Enable		0: Disabled 1: Enabled
[15:4]	-	R	Reserved		-
[3]	EXT3_FALLING_TRIG_ENABLE	R/W	External Input 3 Falling Trigger Enable		0: Disabled 1: Enabled

Bit	Function	Type	Description				Condition	
[2]	EXT2_FALLING_TRIG_ENABLE	R/W	External Input 2 Falling Trigger Enable				0: Disabled 1: Enabled	
[1]	EXT1_FALLING_TRIG_ENABLE	R/W	External Input 1 Falling Trigger Enable				0: Disabled 1: Enabled	
[0]	EXT0_FALLING_TRIG_ENABLE	R/W	External Input 0 Falling Trigger Enable				0: Disabled 1: Enabled	

ITU_EXTIFLG									0x50004028									External Interrupt Flag Register								
Bit	31	30	29	28	27	26	25	24																		
Function					-																					
Reset Value	0	0	0	0	0	0	0	0																		
Bit	23	22	21	20	19	18	17	16																		
Function					-																					
Reset Value	0	0	0	0	0	0	0	0																		
Bit	15	14	13	12	11	10	9	8																		
Function					-																					
Reset Value	0	0	0	0	0	0	0	0																		
Bit	7	6	5	4	3	2	1	0																		
Function					EXT3_INT_F LAG	EXT2_INT_F LAG	EXT1_INT_F LAG	EXT0_INT_F LAG																		
Reset Value	0	0	0	0	0	0	0	0																		

Bit	Function	Type	Description				Condition			
[31:4]	-	R	Reserved				-			
[3]	EXT3_INT_FLAG	R/W	External 3 Interrupt Flag Note: Cleared by writing 1.				0: EXT3 no trigger request occurs. 1: EXT3 INT request occurs.			
[2]	EXT2_INT_FLAG	R/W	External 2 Interrupt Flag Note: Cleared by writing 1.				0: EXT2 no trigger request occurs. 1: EXT2 INT request occurs.			
[1]	EXT1_INT_FLAG	R/W	External 1 Interrupt Flag Note: Cleared by writing 1.				0: EXT1 no trigger request occurs. 1: EXT1 INT request occurs.			
[0]	EXT0_INT_FLAG	R/W	External 0 Interrupt Flag Note: Cleared by writing 1.				0: EXT0 no trigger request occurs. 1: EXT0 INT request occurs.			

ITU_EXTEFLG									0x5000402C									External Event Flag Register								
Bit	31	30	29	28	27	26	25	24																		
Function					-																					
Reset Value	0	0	0	0	0	0	0	0																		

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function			-	KEYCHG_EV T_FLAG	EXT3_EVT_F LAG	EXT2_EVT_F LAG	EXT1_EVT_F LAG	EXT0_EVT_F LAG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:5]	-	R	Reserved	-
[4]	KEYCHG_EVT_FLAG	R/W	Key Change Event Flag Note: Cleared by writing 1.	0: No trigger request occurs. 1: Key Change Event occurs.
[3]	EXT3_EVT_FLAG	R/W	External 3 Event Flag Note: Cleared by writing 1.	0: No trigger request occurs. 1: EXT3 request occurs.
[2]	EXT2_EVT_FLAG	R/W	External 2 Event Flag Note: Cleared by writing 1.	0: No trigger request occurs. 1: EXT2 request occurs.
[1]	EXT1_EVT_FLAG	R/W	External 1 Event Flag Note: Cleared by writing 1.	0: No trigger request occurs. 1: EXT1 request occurs.
[0]	EXT0_EVT_FLAG	R/W	External 0 Event Flag Note: Cleared by writing 1.	0: No trigger request occurs. 1: EXT0 request occurs.

7. Analog Control Unit

7.1. Introduction

Analog control unit is designed for GPCM3 Serials analog devices control

7.2. Features

- Voltage Regulator 3.3V (REG33)
- Voltage Regulator 1.2V (REG12), LVR and BVD
- Internal 12M Oscillator (IOSC12M)
- Internal 32768 Oscillator (IOSC32K)
- External 12M Crystal Oscillator (X12M)
- External 32K Crystal Oscillator (X32K)
- Phase Lock-Loop (PLL)
- Audio PWM (Aud-PWM)

7.3. Function

Voltage Regulator 3.3V (REG33)

Voltage regulator 3.3V can be the power supply of SPI IO (PAD IOB5:0]) and ADC. When **ACU_REG33_CTRL.DSADC_REG_EN** is enabled, **REG33** supplies the power of ADC IO. The SPI IO power is always supplied. When **ACU_REG33_CTRL.SLEEP_EN** is set as 1, **REG33** will provide 2.7V voltage as the power supply of SPI IO, and the regulator will enter Sleep mode.

Low Voltage Reset (LVR)

The on-chip Low Voltage Reset (LVR) circuitry forces the system to enter reset state when power supplying voltage falls below the specific LVR trigger voltage. This function prevents MCU from working at an invalid operating voltage range.

Battery Voltage Detector (BVD)

In order to satisfy some application usage, GPCM3 SERIALS provides battery voltage detector function. BVD function can be enabled or disabled by configuring **ACU_BVD_CTRL.EN**. The result of BVD detect voltage could be shown by **ACU_BVD_CTRL.LVL_SEL[3:0]**.

Internal 12M Oscillator (IOSC12M)

The HSIRC12M clock signal is generated from an internal 12MHz RC oscillator and can be directly used as a system clock or as one of PLL input. The HSIRCRDY flag in the **CLK_CLKSTS.HSIRCRDY** indicates whether the high-speed internal oscillator is stable or not.

External 12M X'tal Oscillator (X12M)

Either Iosc12MHz or X'TAL12MHz can be used as the system clock source.

Internal 32768 Oscillator (IOSC32K)

The LSIRC32K clock signal is generated from an internal 32K Hz RC Oscillator and can be directly used as a system clock. The LSIRCRDY flag in the **CLK_CLKSTS.LSIRCRDY** indicates whether the low-speed internal oscillator is stable or not.

External 32768 Crystal Oscillator (X32K)

The 32K external oscillator has the advantage of producing a very accurate rate on the main clock. The HSXTLRDY flag in the **CLK_CLKSTS.HSXTLRDY** indicates whether the external oscillator is stable or not.

Phase Lock-Loop (PLL)

With clock sources such as IOSC12M Clock, PLL can make the frequency reach the needed value for the frequency multiplier. The configuration of PLL (frequency multiplier value) is in **ACU_PLL_CTRL.DIVC[4:0]**. The configuration must be done before user enables the PLL. Each PLL should be enabled after its input clock becomes stable (ready flag). Once the CPU clock switches to PLL, these parameters cannot be changed. The PLLRDY flag in the **CLK_CLKSTS .PLLRDY** indicates whether the PLL is stable or not.

7.4. Register Description

Register Map

Name	Address	Description
ACU_REG33_CTRL	0x50005000	REG33 Control Register
ACU_REG12_CTRL	0x50005004	REG12 Control Register
ACU_SYS12M_CTRL	0x50005008	SYS12M Control Register
ACU_PLL_CTRL	0x5000500C	PLL Control Register
ACU_X32K_CTRL	0x50005010	X32K Control Register
ACU_I32K_CTRL	0x50005014	IOSC32K Control Register
ACU_PLL_SPI_DIV	0x50005018	PLL_SPI Clock Divide Register
ACU_BVD_CTRL	0x50005030	BVD Control Register
ACU_APAD	0x50005034	Analog Pad Register

Register Function

ACU_REG33_CTRL

0x50005000

REG33 Control Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	SPI_REG_AE	SLEEP_EN	DSADC_REG_EN			-	
Reset Value	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[31:7]	-	-	Reserved	-
[6]	SPI_REG_AE	R/W	LDO3.3V for SPIFC I/F Power Control Bit If this bit is set as 1, LDO (Low-DropOut) REG3.3V for SPIFC I/F is always enabled. Note: This bit is protected by SMU_SYSLOCK .	0: Disabled 1: Enabled

Bit	Function	Type	Description	Condition
[5]	SLEEP_EN	R/W	LDO3.3V SPI Sleep Mode Enable Note2: This bit is protected by SMU_SYSLOCK .	0: Keep in Normal mode (Tolerance +/-5%). 1: Keep in Sleep Mode (2.7V, Tolerance +/-10%)
[4]	DSADC_REG_EN	R/W	DS-ADC (Mic) Regulator Enable Note1: This bit determines whether to disable DS-ADC Power IN (no matter it's normal or sleep mode). Thus, remember to set it as disabled in sleep mode to save power, and set it back as enabled in normal mode. Note2: This bit is protected by SMU_SYSLOCK .	0: Disabled 1: Enabled
[3:0]	-	-	Reserved	-

ACU_REG12_CTRL **0x50005004** **REG12 Control Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	REG12_EN	-	-	SLEEP_EN	-	-	-
Reset Value	0	1	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:7]	-	-	Reserved	-
[6]	REG12_EN	R/W	LDO1.2V Enable Note: This bit is protected by SMU_SYSLOCK , and should always be set as 1, not 0.	0: Disabled 1: Enabled
[5:4]	-	-	Reserved	-
[3]	SLEEP_EN	R/W	LDO1.2V Sleep Mode Enable If this bit is set as 0 (Enabled), when CPU enters sleep mode, LDO1.2V will also enter sleep mode (the voltage will become 0.8V and the regulator will enter power saving mode).	0: Enabled 1: Disabled

Bit	Function	Type	Description					Condition
			If this bit is set as 1 (Disabled), when CPU enters sleep mode, LDO1.2V will be kept at the voltage level the same as it's in normal mode, which is 1.2V. Note: This bit is protected by SMU_SYSLOCK .					
[2:0]	-	-	Reserved					-

ACU_SYS12M_CTRL **0x50005008** **SYS12M Control Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function						STRONG_E_N	X12M_EN	IOSC12M_EN
Reset Value	0	0	0	1	0	0	1	0

Bit	Function	Type	Description					Condition
[31:15]	-	-	Reserved					-
[14:8]	IOSC12M_TRIM[6:0]	R/W	SYS12M Software Trim Bits Note: This bit is protected by SMU_SYSLOCK .					-
[7:3]	-	-	Reserved					-
[2]	STRONG_EN	R/W	12M X'tal String Mode Enable Note: This bit is protected by SMU_SYSLOCK .					0: Disabled 1: Enabled
[1]	X12M_EN	R/W	12M X'tal Enable (Low Active) Note: This bit is protected by SMU_SYSLOCK .					0: Enabled 1: Disabled
[0]	IOSC12M_EN	R/W	IOSC12M Enable (Low Activ) Note: This bit is protected by SMU_SYSLOCK .					0: Enabled 1: Disabled

ACU_PLL_CTRL **0x5000500C** **PLL Control Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Function			DIVC[4:0]			-		PLL_EN
Reset Value	0	0	1	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	-	Reserved					-
[7:3]	DIVC[4:0]	R/W	PLL Clock Selection Note: This bit is protected by SMU_SYSLOCK . Note: PLL Clock frequency is related to Clock source. If user selects IOSC as CLK source, the frequency multiplier value is 491.52MHz. If the external XTAL is 12MHz, the frequency multiplier value is 480MHz.					0x00~0x02 = 122.88M 0x03~0x1F = 491.52M/(DIVC + 1)
[2:1]	-	-	Reserved					-
[0]	PLL_EN	R/W	PLL Enable When HCLK selects PLL (selection is in 0x5000_1020[1:0]), this bit will be enabled automatically. Note: This bit is protected by SMU_SYSLOCK .					0: Disabled 1: Enabled

ACU_X32K_CTRL								0x50005010	X32K Control Register
Bit	31	30	29	28	27	26	25	24	
Function				-					
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MODE_SEL[1:0]		CTS_ADC_L DO33_SLP_EN	CTS_ADC_L DO12_SLP_EN	MACRO_SL_P_EN	32K_SLP_E_N	-	X32K_EN
Reset Value	1	1	0	0	0	1	0	1

Bit	Function	Type	Description	Condition
[31:8]	-	-	Reserved	-
[7:6]	MODE_SEL[1:0]	R/W	Strong/Enhance/Weak Mode Selection Note: This bit is protected by SMU_SYSLOCK .	00 = Weak Mode 01 = Enhance Mode 10 = Strong Mode 11 = Strong Mode
[5]	CTS_ADC_LDO33_SLP_EN	R/W	LDO33(ADC, CTS) Sleep Control Note: This bit is protected by SMU_SYSLOCK .	0: Disabled 1: Enabled
[4]	CTS_ADC_LDO12_SLP_EN	R/W	LDO12(ADC, CTS) Sleep Control Note: This bit is protected by SMU_SYSLOCK .	0: Disabled 1: Enabled
[3]	MACRO_SLP_EN	R/W	Macro Sleep control Note: This bit is protected by SMU_SYSLOCK .	0: Disabled 1: Enabled
[2]	32K_SLP_EN	-	IOSC32K or XTAL32K Sleep Control Note: This bit is protected by SMU_SYSLOCK .	0: Disabled 1: Enabled
[1]	-	-	Reserved	-
[0]	X32K_EN	-	X32K Enable (Low Active) Note: This bit is protected by SMU_SYSLOCK .	0: Enabled 1: Disabled

ACU_I32K_CTRL **0x50005014** **IOSC32K Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOSC32K_TRIM[6:0]							I32K_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	-	-	Reserved	-
[7:1]	IOSC32K_TRIM[6:0]	R/W	IOSC32K Trim Bits Note: This bit is protected by SMU_SYSLOCK	-
[0]	I32K_ENABLE	R/W	Enable I32K (Low Active1: I32K disabled) Note: This bit is protected by SMU_SYSLOCK	0: Enabled 1: Disabled

ACU_PLL_SPI_DIV
0x50005018
PLL_SPI Clock Divide Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-				DIVC[4:0]			
Reset Value	0	0	0	0	0	1	0	1

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:13]	-	-	Reserved						-
[12:8]	DIVC[4:0]	R/W	SPI Flash Controller's PLL Clock Note: This bit is protected by SMU_SYSLOCK .						0x00~0x02: 120M 0x03~0x1F: 480M/(DIVC + 1)
[7:0]	-	-	Reserved						-

ACU_BVD_CTRL
0x50005030
BVD Control Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-				EN			
Reset Value	0	0	0	0	1	0	0	0

Bit	Function	Type	Description						Condition
[31:5]	-	-	Reserved						-
[4]	EN	R/W	Battery Voltage Detect Enable						0: Disabled 1: Enabled

[3:0]	LVL_SEL[3:0]	R	Battery Voltage Detect Trigger Level (Read only)						0000: VDD≤2.0V 0001: 2.0V<VDD<2.2V 0010: 2.2V<VDD<2.4V 0011: 2.4V<VDD<2.6V
-------	--------------	---	--	--	--	--	--	--	---

Bit	Function	Type	Description					Condition
								0100: 2.6V<VDD<2.8V 0101: 2.8V<VDD<3.0V 0110: 3.0V<VDD<3.3V 0111: 3.3V<VDD<3.6V 1000: VDD>3.6V

ACU_APAD **Analog Pad Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function								X32K_PAD_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					LINE_PAD_EN[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	-	Reserved					-
[8]	X32K_PAD_EN	R/W	Crystal 32KHz Pad (IOA[8:7]) Enable					0: Disabled 1: Enabled
[7:0]	LINE_PAD_EN[7:0]	R/W	Line-in Pad Enable					[0]: IOA24 [1]: IOA25 [2]: IOA26 [3]: IOA27 [4]: IOA28 [5]: IOA29 [6]: IOA30 [7]: IOA31

8. DMA Control Unit

8.1. Introduction

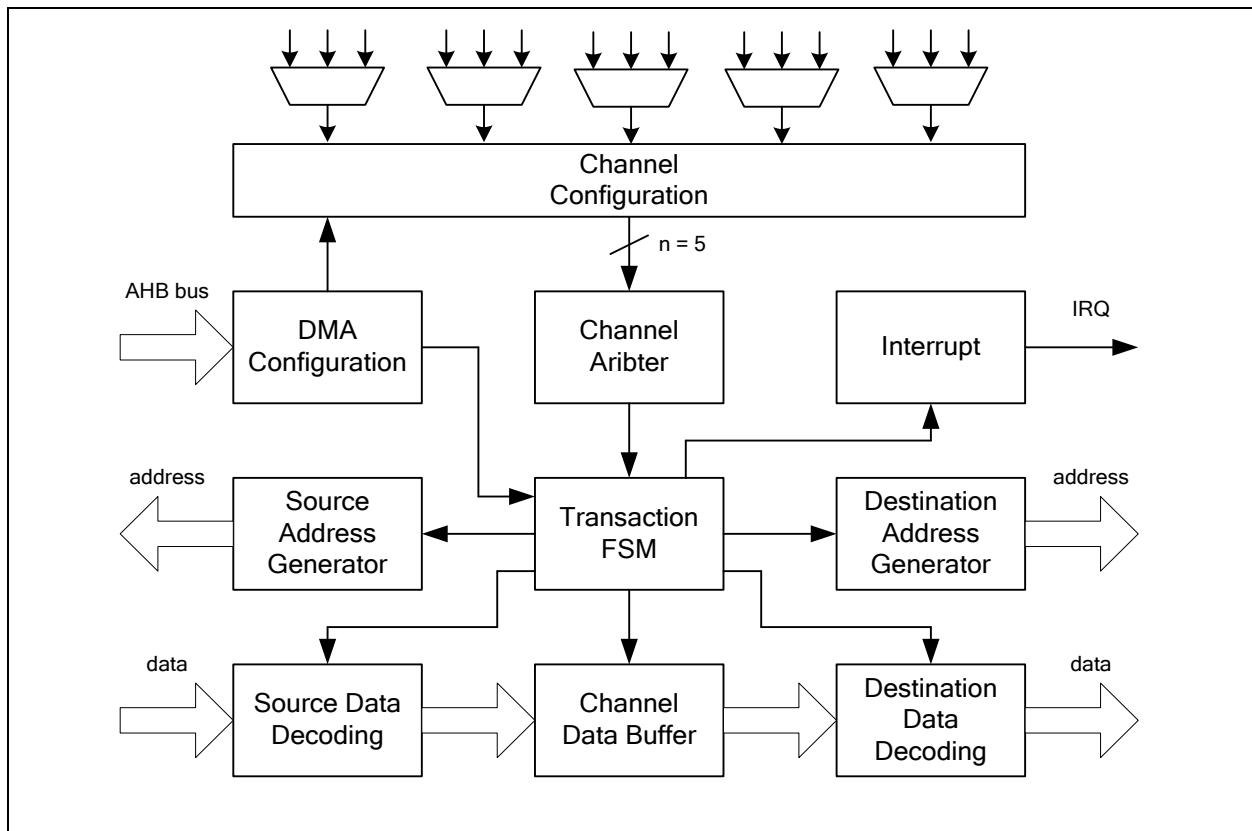
Direct Memory Access (DMA) is used to provide high-speed data transfer between peripherals and memory as well as memory and memory. Data can be moved quickly by DMA without any CPU actions. This keeps CPU resources free for other operations.

The DMA controllers have 5 channels, and each channel is dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority if there are multiple DMA requests.

8.2. Features

- Five Independently Configurable Channels
- Each channel can be connected to dedicated hardware DMA request. If there's no hardware trigger source, software trigger is also supported.
- Priorities between requests are based on channel number. For example, the request in channel 1 has higher priority than the request in channel 2.
- Independent Source and Destination Transfer Size (8-bit, 16-bit, 32-bit)
- Support burst operation.
- After DMA Transfer is completed, each channel will issue an interrupt request respectively.
- Support memory-to-memory, peripheral-to-memory, memory-to-peripheral and peripheral-to-peripheral transfer with the configurable DMA source and destination addresses.
- Programmable number of data to be transferred: up to 65536.

8.3. Block Diagram



8.4. Function Description

8.4.1. DMA Transaction

After an event, the peripheral sends a request signal to the DMA Controller. The DMA controller serves the request depending on the channel priorities. As soon as the DMA Controller accesses the peripheral, an Acknowledge is sent to the peripheral by the DMA Controller. The peripheral releases its request as soon as it gets the Acknowledge from the DMA Controller.

8.4.2. DMA Request Arbiter

The arbiter manages the channel requests based on their priority and launches the peripheral/memory access sequences. If 2 requests have the same software priority level, the channel with the lowest number will get priority versus the channel with the highest number. For example, channel 0 gets priority over channel 1.

8.4.3. DMA Channels

Each channel can handle DMA transfer between a source address and a destination address. The amount of data to be transferred (up to 65535) is programmable. The register which contains the amount of data items to be transferred is decremented after each transaction.

Programmable Data Sizes

Transfer data sizes of the peripheral and memory are fully programmable through the DMA_SRCSIZE and DMA_DSTSIZE bits in the DMA_CTRLx register.

Address Increment

Source and destination address can optionally be automatically post-incremented after each transaction depending on the DMA_SRCINC and DMA_DSTINC bits in the DMA_CTRLx register.

If incremented mode is enabled, the address of the next transfer will be the address of the previous one incremented by 1, 2 or 4 depending on the chosen data size. The first transfer address is the one programmed in the DMA_SRCADR / DMA_DSTADR registers. During transfer operations, these registers keep the initially programmed value. The current transfer addresses (in the current internal source/destination address register) are not accessible by software.

Burst Mode

In DMA burst mode, the first request must be sent from software or peripheral device. After first transfer completed, DMA controller will automatically execute next transfer and discrete the internal transfer counter, reloaded from DMA_DTN. When internal transfer counter is not equivalent to 0, DMA controller will execute transfer and discrete the internal transfer counter until total transfer number completed.

8.4.4. Programmable Data Width and Data Alignment

When **DMA_SRCSIZE** and **DMA_DSTSIZE** are not equal, the DMA will perform data alignments as described in the below table.

Source Width	Destination Width	Transfer Number	Source Content: Address/Data	Transfer Operation	Destination Content: Address/Data
8	8	4	@0x0 / A0 @0x1 / A1 @0x2 / A2 @0x3 / A3	1: Read 0xA0 @0x0 Write A0 @ 0x0 2: Read 0xA1 @0x1 Write A1 @ 0x1 3: Read 0xA2 @0x2 Write A2 @ 0x2 4: Read 0xA3 @0x3 Write A3 @ 0x3	@0x0 / A0 @0x1 / A1 @0x2 / A2 @0x3 / A3
8	16	4	@0x0 / A0 @0x1 / A1 @0x2 / A2 @0x3 / A3	1: Read 0xA0 @0x0 Write 0x00A0 @ 0x0 2: Read 0xA1 @0x1 Write 0x00A1 @ 0x2 3: Read 0xA2 @0x2 Write 0x00A2 @ 0x4 4: Read 0xA3 @0x3 Write 0x00A3 @ 0x6	@0x0 / 00A0 @0x2 / 00A1 @0x4 / 00A2 @0x6 / 00A3
8	32	4	@0x0 / A0 @0x1 / A1 @0x2 / A2 @0x3 / A3	1: Read 0xA0 @0x0 Write 0x000000A0 @ 0x0 2: Read 0xA1 @0x1 Write 0x000000A1 @ 0x4 3: Read 0xA2 @0x2 Write 0x000000A2 @ 0x8 4: Read 0xA3 @0x3 Write 0x000000A3 @ 0xC	@0x0 / 000000A0 @0x4 / 000000A1 @0x8 / 000000A2 @0xC / 000000A3
16	8	4	@0x0 / A1A0 @0x2 / A3A2 @0x4 / A5A4 @0x6 / A7A6	1: Read 0xA1A0 @0x0 Write 0xA0 @ 0x0 2: Read 0xA3A2 @0x2 Write 0xA2 @ 0x1 3: Read 0xA5A4 @0x4 Write 0xA4 @ 0x2 4: Read 0x7DA6 @0x6 Write 0xA6 @ 0x3	@0x0 / A0 @0x1 / A2 @0x2 / A4 @0x3 / A6
16	16	4	@0x0 / A1A0 @0x2 / A3A2 @0x4 / A5A4 @0x6 / A7A6	1: Read 0xA1A0 @0x0 Write 0xA1A0 @ 0x0 2: Read 0xA3A2 @0x2 Write 0xA3A2 @ 0x2 3: Read 0xA5A4 @0x4 Write 0xA5A4 @ 0x4 4: Read 0x7DA6 @0x6 Write 0xA7A6 @ 0x6	@0x0 / A1A0 @0x2 / A3A2 @0x4 / A5A4 @0x6 / A7A6
16	32	4	@0x0 / A1A0 @0x2 / A3A2 @0x4 / A5A4 @0x6 / A7A6	1: Read 0xA1A0 @0x0 Write 0x0000A1A0 @ 0x0 2: Read 0xA3A2 @0x2 Write 0x0000A3A2 @ 0x4 3: Read 0xA5A4 @0x4 Write 0x0000A5A4 @ 0x8 4: Read 0x7DA6 @0x6 Write 0x0000A7A6 @ 0xC	@0x0 / 0000A1A0 @0x4 / 0000A3A2 @0x8 / 0000A5A4 @0xC / 0000A7A6
32	8	4	@0x0 / A3A2A1A0 @0x4 / A7A6A5A4 @0x8 / ABAAA9A8 @0xC / AFAEADAC	1: Read 0xA3A2A1A0 @0x0 Write 0xA0 @ 0x0 2: Read 0xA7A6A5A4 @0x4 Write 0xA4 @ 0x1 3: Read 0xABAAA9A8 @0x8 Write 0xA8 @ 0x2 4: Read 0xAFAEADAC @0xC Write 0xAC @ 0x3	@0x0 / A0 @0x1 / A4 @0x2 / A8 @0x3 / AC
32	16	4	@0x0 / A3A2A1A0 @0x4 / A7A6A5A4 @0x8 / ABAAA9A8 @0xC / AFAEADAC	1: Read 0xA3A2A1A0 @0x0 Write 0xA1A0 @ 0x0 2: Read 0xA7A6A5A4 @0x4 Write 0xA5A4 @ 0x2 3: Read 0xABAAA9A8 @0x8 Write 0xA9A8 @ 0x4 4: Read 0xAFAEADAC @0xC Write 0xADAC @ 0x6	@0x0 / A1A0 @0x2 / A5A4 @0x4 / A9A8 @0x6 / ADAC
32	32	4	@0x0 / A3A2A1A0 @0x4 / A7A6A5A4 @0x8 / ABAAA9A8 @0xC / AFAEADAC	1: Read 0xA3A2A1A0 @0x0 Write 0xA3A2A1A0 @ 0x0 2: Read 0xA7A6A5A4 @0x4 Write 0xA7A6A5A4 @ 0x4 3: Read 0xABAAA9A8 @0x8 Write 0xABAAA9A8 @ 0x8 4: Read 0xAFAEADAC @0xC Write 0xAFAEADAC @ 0xC	@0x0 / A3A2A1A0 @0x4 / A7A6A5A4 @0x8 / ABAAA9A8 @0xC / AFAEADAC

8.4.5. Error Management

When DMA read or write access cycle is occupied by another master such CPU for over 256 system clocks, the transfer error flag (TEIF) will be set as 1.

8.4.6. Interrupts

Each DMA channel will issue an interrupt after the transfer is completed or a transfer error happens. Separate interrupt enable bits are available for flexibility.

The DMA controller interrupt which connected to NVIC is composed of Transfer complete logically ORed Transfer error interrupts.

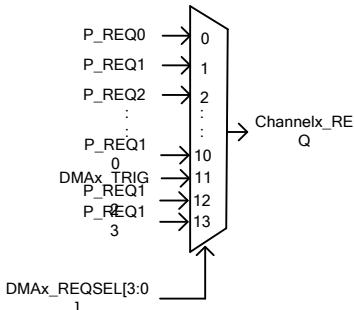
8.4.7. Channel Selection

Each DMA channel supports 8 different peripheral requests. The selection is controlled by the DMA_REQSEL[3:0] bits in the DMA_CTRLx register.

GPCM3 supports 5 sets of DMA, and there's no usage limits of DMA0~2. If user applies DMA3/4 in SPI transfer, SPI will not be able to receive the complete signal from DMA after data transfer is completed. Please apply DMA0~2 when there's a need of using DMA as SPI transfer.

In addition, if more than 4 or 5 sets of DMA are in use at the same time, please turn on Continuous mode function while applying DMA3/4. (User only has to turn on this function while using DMA3/4 but not DMA0~2. To turn on continuous mode function, please set DMA3/4 CTRL Bit[7] as 1.)

Note: Although there's usage limits for applying DMA3/4 in SPI transfer, the functions and INT of DMA3/4 can be used in other applications normally, such as playing A34Pro and MS02.

Channel Selection Diagram

DMA Request Mapping

Peripherals	Channel 0/1/2/3/4
ADC_REQ0	P-REQ0
DAC_REQ0	P-REQ1
DAC_REQ1	P-REQ2
SPI0	P-REQ3
UART	P-REQ4
I2S	P-REQ5
I2C	P-REQ6
ADC_REQ1	P-REQ7
SPI1	P-REQ8
SPUL	P-REQ9
SPUR	P-REQ10
DMA_TRIG	P-REQ11
CCP_REQ	P-REQ12
UART1	P-REQ13

8.5.Register Description
Register Map

Name	Address	Description
DMA_INTSTS	0x500F0000	DMA Interrupt Status Register
DMA0_CTRL	0x500F0010	DMA Channel 0 Control Register
DMA0_DTN	0x500F0014	DMA Channel 0 Data Transfer Number Register
DMA0_SRCADR	0x500F0018	DMA Channel 0 Source Address Register
DMA0_DSTADR	0x500F001C	DMA Channel 0 Destination Address Register
DMA1_CTRL	0x500F0020	DMA Channel 1 Control Register
DMA1_DTN	0x500F0024	DMA Channel 1 Data Transfer Number Register
DMA1_SRCADR	0x500F0028	DMA Channel 1 Source Address Register
DMA1_DSTADR	0x500F002C	DMA Channel 1 Destination Address Register
DMA2_CTRL	0x500F0030	DMA Channel 2 Control Register

Name	Address	Description
DMA2_DTN	0x500F0034	DMA Channel 2 Data Transfer Number Register
DMA2_SRCADR	0x500F0038	DMA Channel 2 Source Address Register
DMA2_DSTADR	0x500F003C	DMA Channel 2 Destination Address Register
DMA3_CTRL	0x500F0040	DMA Channel 3 Control Register
DMA3_DTN	0x500F0044	DMA Channel 3 Data Transfer Number Register
DMA3_SRCADR	0x500F0048	DMA Channel 3 Source Address Register
DMA3_DSTADR	0x500F004C	DMA Channel 3 Destination Address Register
DMA4_CTRL	0x500F0050	DMA Channel 4 Control Register
DMA4_DTN	0x500F0054	DMA Channel 4 Data Transfer Number Register
DMA4_SRCADR	0x500F0058	DMA Channel 4 Source Address Register
DMA4_DSTADR	0x500F005C	DMA Channel 4 Destination Address Register

Registers Function

DMA_INTERRUPT_STATUS_REGISTER							
Bit	31	30	29	28	27	26	25
Function	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17
Function	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9
Function	-	-	DMA4_TX_ERR_INT_FLAG	DMA3_TX_ERR_INT_FLAG	DMA2_TX_ERR_INT_FLAG	DMA1_TX_ERR_INT_FLAG	DMA0_TX_ERR_INT_FLAG
Reset Value	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1
Function	-	-	DMA4_DONE	DMA3_DONE	DMA2_DONE	DMA1_DONE	DMA0_DONE
Reset Value	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:13]	-	R	Reserved	-
[12]	DMA4_TX_ERR_INT_FLAG	R/W	Channel 4 Transfer Error Flag Note: It's cleared by the software writing 1.	0: There's no transfer error on channel 4. 1: There's transfer error on channel 4.
[11]	DMA3_TX_ERR_INT_FLAG	R/W	Channel 3 Transfer Error Flag Note: It's cleared by the software writing 1.	0: There's no transfer error on channel 3. 1: There's transfer error on channel 3.

Bit	Function	Type	Description	Condition
[10]	DMA2_TX_ERR_INT_FLAG	R/W	Channel 2 Transfer Error Flag Note: It's cleared by the software writing 1.	0: There's no transfer error on channel 2. 1: There's transfer error on channel 2.
[9]	DMA1_TX_ERR_INT_FLAG	R/W	Channel 1 Transfer Error Flag Note: It's cleared by the software writing 1.	0: There's no transfer error on channel 1. 1: There's transfer error on channel 1.
[8]	DMA0_TX_ERR_INT_FLAG	R/W	Channel 0 Transfer Error Flag Note: It's cleared by the software writing 1.	0: There's no transfer error on channel 0. 1: There's transfer error on channel 0.
[7:5]	-	R	Reserved	-
[4]	DMA4_DONE_INT_FLAG	R/W	Channel 4 Transaction Complete Flag Note: It's cleared by the software writing 1.	0: No transfer is completed on channel 4. 1: Transfer is completed on channel 4.
[3]	DMA3_DONE_INT_FLAG	R/W	Channel 3 Transaction Complete Flag Note: It's cleared by the software writing 1.	0: No transfer is completed on channel 3. 1: Transfer is completed on channel 3.
[2]	DMA2_DONE_INT_FLAG	R/W	Channel 2 Transaction Complete Flag Note: It's cleared by the software writing 1.	0: No transfer is completed on channel 2. 1: Transfer is completed on channel 2.
[1]	DMA1_DONE_INT_FLAG	R/W	Channel 1 Transaction Complete Flag Note: It's cleared by the software writing 1.	0: No transfer is completed on channel 1. 1: Transfer is completed on channel 1.
[0]	DMA0_DONE_INT_FLAG	R/W	Channel 0 Transaction Complete Flag Note: It's cleared by the software writing 1.	0: No transfer is completed on channel 0. 1: Transfer is completed on channel 0.

DMA0_CTRL	0x500F0010							DMA Channel 0 Control Register
DMA1_CTRL	0x500F0020							DMA Channel 1 Control Register
DMA2_CTRL	0x500F0030							DMA Channel 2 Control Register
DMA3_CTRL	0x500F0040							DMA Channel 3 Control Register
DMA4_CTRL	0x500F0050							DMA Channel 4 Control Register

Bit	31	30	29	28	27	26	25	24
Function	-						DMA_FULL_FLAG	DMA_BUSY_FLAG
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-				DMA_DUAL_EN	DMA_STPS	DMA_STOP	DMA_START
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DMA_REQSEL[3:0]				DMA_SRCSIZE[1:0]	DMA_DSTSIZE[1:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DMA_CONTINUOUS_ENABLE	DMA_BURST_ENABLE	DMA_SRCIN_C_ENABLE	DMA_DSTIN_C_ENABLE	DMA_CIRC_ENABLE	DMA_ERR_INT_ENABLE	DMA_DONE_INT_ENABLE	DMA_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:26]	-	R	Reserved	-
[25]	DMA_FULL_FLAG	R	Channel x Double Buffer Full Flag	0: Not full 1: Full
[24]	DMA_BUSY_FLAG	R	Channel x Operation Status This flag will keep high during data transfer.	0: Idle 1: Busy
[23:20]	-	R	Reserved	-
[19]	DMA_DUAL_EN	R/W	DMA Dual Trigger Mode Enable Note: DMA will move the data twice and thus trigger two INTs. After moving the data length set by the user for the first time, the first DMA INT will be triggered, and DMA will keep moving data at this time. The second DMA INT will be triggered after DMA finishes the second data moving.	0: Single Mode 1: Dual Trigger Mode
[18]	DMA_STPS	R/W	Channel x Software Stop Request Selection	0: Stop immediately. 1: Stop when data transfer is completed.
[17]	DMA_STOP	W	Channel x Software Stop Request Write 1 to enable this function. Write 0 and there will be no function. This bit always reads 0.	0: No function. 1: Stop.

Bit	Function	Type	Description	Condition
			Note: When this bit = 1, DMA_STPS (Bit[18]) will determine to stop DMA immediately or to stop DMA after the transmission is completed.	
[16]	DMA_START	W	Channel x Software Trigger Request Write 1 to enable this function. Write 0 and there will be no function. This bit always reads 0. Note: If no trigger source is in use, for example in the case of memory-to-memory, DMA data moving will be triggered by software trigger function.	0: No function. 1: Start.
[15:12]	DMA_REQSEL[3:0]	R/W	Channel x Peripheral REQ Selection Note: DMA_REQSEL only can be changed when DMA_BUSY = 0.	0000 = SAR-ADC_REQ 0001 = DAC0_REQ 0010 = DAC1_REQ 0011 = SPI0_REQ 0100 = UART0_REQ 0101 = I2S_REQ 0110 = I2C_REQ 0111 = DS-ADC_REQ 1000 = SPI1_REQ 1001 = SPUL_REQ 1010 = SPUR_REQ 1011 = MEM_REQ 1100 = CCP_REQ 1101 = UART1_REQ
[11:10]	DMA_SRCSIZE[1:0]	R/W	Channel x Source Data Size	00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = Reserved
[9:8]	DMA_DSTSIZE[1:0]	R/W	Channel x Destination Data Size	00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = Reserved
[7]	DMA_CONTINUOUS_ENABLE	R/W	DMA Continuous Mode	0: Disable 1: Enable
[6]	DMA_BURST_ENABLE	R/W	DMA Burst Transfer Mode Note: When DMA receives a request, it will automatically transfer all the data at once according to the transfer number set by user. Note: After Burst Mode is enabled, DMA will continue transferring the data, and other requests will be ignore during the time of transferring.	0: Disabled 1: Enabled
[5]	DMA_SRCINC_ENABLE	R/W	Channel x Source Address Increment Mode	0: Disabled 1: Enabled

Bit	Function	Type	Description	Condition
[4]	DMA_DSTINC_ENABLE	R/W	Channel x Destination Address Increment Mode Note: If Circular mode is enabled, DMA can perform data moving in infinite loops until user executes Stop command.	0: Disabled 1: Enabled
[3]	DMA_CIRC_ENABLE	R/W	Channel x Circular Mode	0: Disabled 1: Enabled
[2]	DMA_ERR_INT_ENABLE	R/W	Channel x Transfer Error Interrupt Enabling Bit	0: Disabled 1: Enabled
[1]	DMA_DONE_INT_ENABLE	R/W	Channel x Transfer Complete Interrupt Enabling Bit	0: Disabled 1: Enabled
[0]	DMA_ENABLE	R/W	DMA Channel x Enabling Bit Write 1 to enable this function. Write 0 and there will be no function. This bit always reads 0.	0: Disabled 1: Enabled

DMA0_DTN **0x500F0014** **DMA Channel 0 Data Transfer Number Register**

DMA1_DTN **0x500F0024** **DMA Channel 1 Data Transfer Number Register**

DMA2_DTN **0x500F0034** **DMA Channel 2 Data Transfer Number Register**

DMA3_DTN **0x500F0044** **DMA Channel 3 Data Transfer Number Register**

DMA4_DTN **0x500F0054** **DMA Channel 4 Data Transfer Number Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DMA_DTN[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DMA_DTN[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:0]	DMA_DTN[15:0]	R/W	Channel x Data Transfer Number Counter The number of data to be transferred. If this register is set as 0, there won't be any transfer whether the DMA channel (DMA_EN) is enabled or not.	-

Bit	Function	Type	Description	Condition
			<p>When the DMA channel is enabled from disable state, DMA_DTN will be reloaded to internal counter, and the counter will be decreased by one every time a transfer is completed.</p> <p>Even if DMA_DTN is changed by the software, the internal counter register will not be updated when DMA channel is enabled.</p>	

DMA0_SRCADR	0x500F0018	DMA Channel 0 Source Address Register
DMA1_SRCADR	0x500F0028	DMA Channel 1 Source Address Register
DMA2_SRCADR	0x500F0038	DMA Channel 2 Source Address Register
DMA3_SRCADR	0x500F0048	DMA Channel 3 Source Address Register
DMA4_SRCADR	0x500F0058	DMA Channel 4 Source Address Register

Bit	31	30	29	28	27	26	25	24
Function	DMA_SRCADR[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DMA_SRCADR[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DMA_SRCADR[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DMA_SRCADR[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	DMA_SRCADR[31:0]	R/W	<p>Channel x Source Data Transfer Address</p> <p>Base address of the source area from which the data will be read.</p> <p>When DMA_SRCSIZE is 01 (16-bit), the DMA_SRCADR[0] bit is ignored. Access is automatically aligned to a half-word address.</p> <p>When DMA_SRCSIZE is 10 (32-bit), DMA_SRCADR[1:0] are ignored. Access is automatically aligned to a word address.</p> <p>Note: Source address doesn't support from 0x00000000 ~ 0x1FFFFFFF.</p>	-

DMA0_DSTADR	0x500F001C	DMA Channel 0 Destination Address Register
DMA1_DSTADR	0x500F002C	DMA Channel 1 Destination Address Register
DMA2_DSTADR	0x500F003C	DMA Channel 2 Destination Address Register
DMA3_DSTADR	0x500F004C	DMA Channel 3 Destination Address Register
DMA4_DSTADR	0x500F005C	DMA Channel 4 Destination Address Register

Bit	31	30	29	28	27	26	25	24
Function	DMA_DSTADR[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DMA_DSTADR[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DMA_DSTADR[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DMA_DSTADR[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	DMA_DSTADR[31:0]	R/W	Channel x Destination Data Transfer Address Base address of the destination area to which the data will be written. When DMA_DSTSIZE is 01 (16-bit), the DMA_DSTADR[0] bit is ignored. Access is automatically aligned to a half-word address. When DMA_DSTSIZE is 10 (32-bit), DMA_DSTADR[1:0] are ignored. Access is automatically aligned to a word address. Note: Destination address doesn't support from 0x00000000 ~ 0xFFFFFFFF.	-

9. CCP0/1 Control

9.1. Introduction

GPCM3 has two sets of Comparison, Capture and PWM Function (CCP), and 8 channels of PWM outputs.

9.2. Feature

- Support 2xCCP Function CCP0 and CCP1.
- Each CCP supports 4 Chanel Capture mode, Comparison mode and PWMIO modes.
- Each CCP clock source has HCLK, HCLK_div2, ..., HCLK_div4096, TimerA, TimerB, and TimerC.
- CCP0/1 will trigger ADC sample through adc_trig signal.
- CCP1 PWMIO0 (IOA.26 or IOA.9) is able to update PWM duty via DMA.
- CCP PWM mode supports Complementary mode. This means the configuration of CCP0 Timer counter can be applied to Ch3 and Ch4 of CCP1. It can become 2+6 pins or remain 4+4 pins as the original setting, which makes applications more flexible.
- Each CCP all has independent Interrupt.

9.3. Block Diagrams

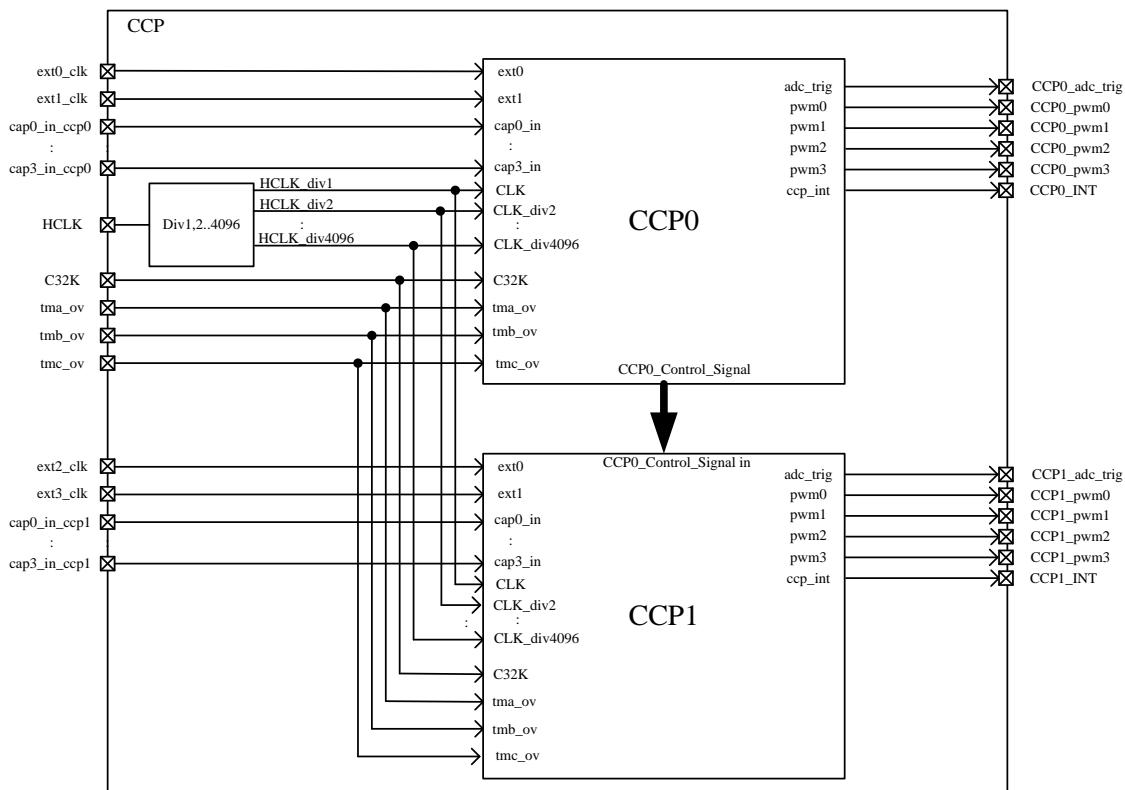


Fig.8.1 CCP Structure

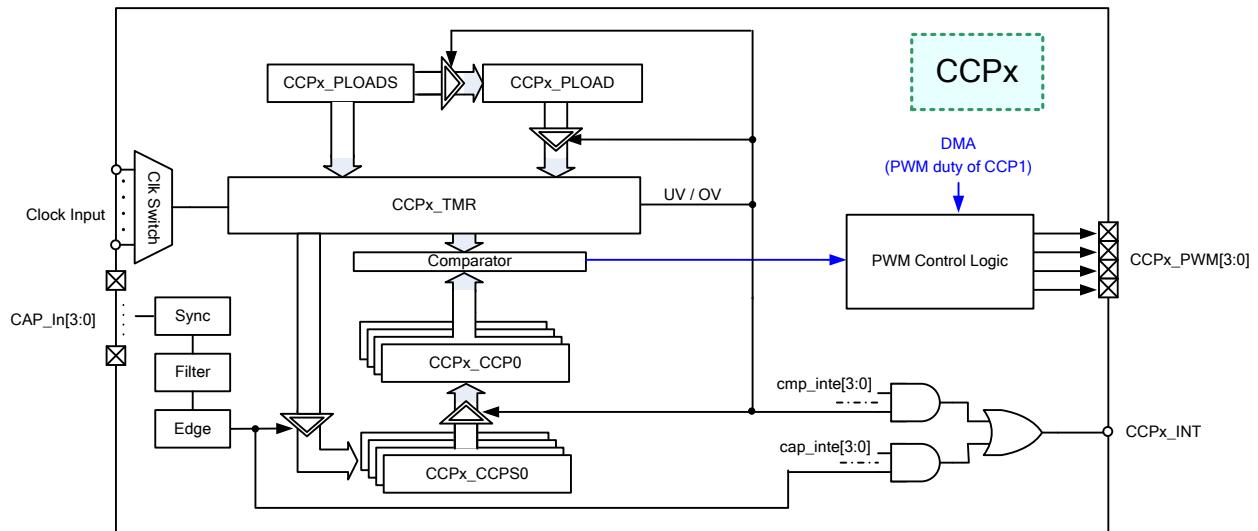


Fig.8.2 CCPx Structure

9.4.Function

All CCPX support three functions: Compare Mode, Capture Mode and PWM Mode. The mode options are selected via CCPx_TMR_CTRL.TMR_MODE.

The CCP0/1 IO map is shown as follow:

CCP0 in IO Selection	CCP0	
	GPIOFUNC_CTRL0[30] = 0	GPIOFUNC_CTRL0[30] = 1
CCP0 Pin0	IOA0[3]	IOA0 [13]
CCP0 Pin1	IOA0 [4]	IOA0 [14]
CCP0 Pin2	IOA0 [5]	IOA0 [15]
CCP0 Pin3	IOA0 [6]	IOA0 [16]
Capture in IO Selection	CCP1	
	GPIOFUNC_CTRL0[9] = 0	GPIOFUNC_CTRL0[9] = 1
CCP1 Pin0	IOA0[9]	IOA0 [26]
CCP1 Pin1	IOA0 [10]	IOA0 [27]
CCP1 Pin2	IOA0 [11]	IOA0 [28]
CCP1 Pin3	IOA0 [12]	IOA0 [29]

Table8.1 CCPx IO Map

Compare Mode:

When CCPx_TMR_CTRL.TMR_MODE is set as 'b00, CCPx will enter Compare Mode. In Compare Mode, the value in CCP0/1_CCPRx Register will be compared with the count value of Timer Counter (CCP0/1_PLOAD). When the Timer Counter value matches CCPRx value, a corresponding interrupt will be issued.

Counter_up:

If the CCPx_TMR_CTRL.TMR_UDC_SEL is set as 1, CCPx_TMR will execute counter up function. When CCPx_TMR_CTRL.TMR_STR be set as 1, CCPx_TMR will load zero value and start counting up. It will generate an overflow interrupt when CCPx_TMR value equals CCPx_TMR_PLOAD. When the overflow event happens, CCPx_TMR will load zero value and start counting up again.

Counter_down:

If the CCPx_TMR_CTRL.TMR_UDC_SEL is set as 0, CCPx_TMR will execute counter down function. When CCPx_TMR_CTRL.TMR_STR be set as 1, CCPx_TMR will load CCPx_TMR_PLOAD value and start counting down. It will generate an overflow interrupt when CCPx_TMR value equals zero. When the overflow event happens, CCPx_TMR will load CCPx_TMR_PLOAD value and start counting down again.

Capture Mode:

When CCPx_TMR_CTRL.TMR_MODE is set as 'b01, CCPx will enter Capture Mode. The CCPx Supports 4 Capture Source (CAPin0~3).

Every Capture pin supports IO debounce function to avoid IO bounce. For each pin, user is able to select when to trigger capture function, at Rising, Falling or Both Edge, through CCPx_CAP_CTRL.CAPx_EDGE. When CCPx IO inputs valid trigger signal, CCP will capture the current Timer Data to CCPR0~3, and the corresponding capture INT will be triggered.

Capture in IO Selection	CCPx_TMR Store in
Capin0	CCPR0
Capin1	CCPR1
Capin2	CCPR2
Capin3	CCPR3

Table8.2 CAPinx Map

The function of CCPx_TMR in Capture mode is the same it is in Counter/Compare Mode.

PWM Mode:

When CCPx_TMR_CTRL.TMR_MODE is set as 'b02, CCPx will enter PWM Mode. Each CCP module supports 4x PWMIO, and through CCPx_PWM_CTRL. PWMxpole, user can select the mode of each PWMIO, as Independent or Complementary mode. In Complementary mode, PWM1 Out = ~PWM0 Out and PWM3 Out = ~PWM2 Out. When PWM starts, CCPx_TMR function is the same as Counter/Compare counting down. It will load CCPx_TMR_PLOAD value and start counting down. When it reaches 0, it will reload CCPx_TMR_PLOAD value. When CCPx_TMR > CCPRx, PWMIOx will output 0, else PWMIO will output 1. The description of this function is shown in Fig 8.3. In complementary mode, PWM0/PWM2 will start dead time counter (DTIME) at PWM_reg rising edge but PWM1/PWM3 will start dead time counter (DTIME) at PWM_reg falling edge.

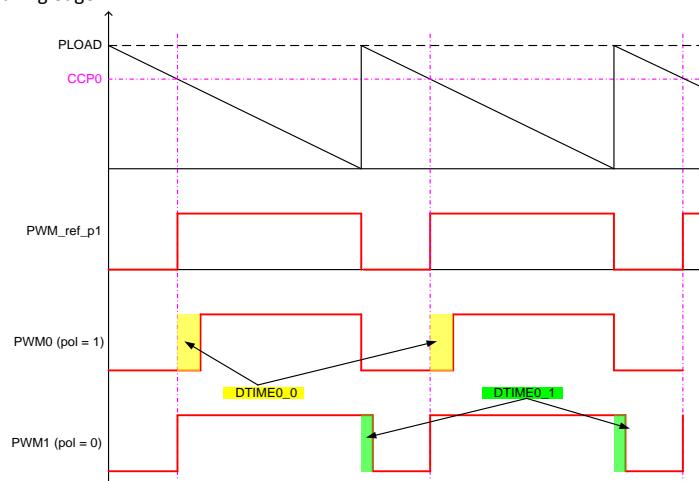
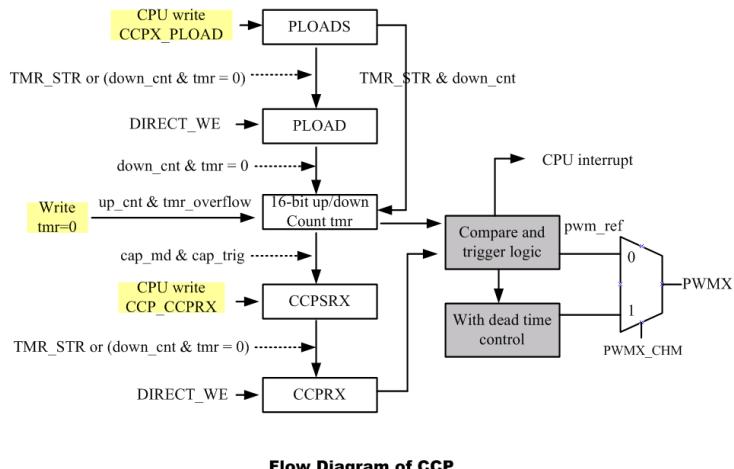


Fig 8.3 PWM Output

Control Flow:

Flow Diagram of CCP
9.5. Register Description
Registers Map

Name	Address	Description
CCPO_TMCMP_CTRL	0x50040000	CCPO Timer Control Register
CCPO_CAP_CTRL	0x50040004	CCPO Capture Mode Control Register
CCPO_PWM_CTRL	0x50040008	CCPO PWM Mode Control Register
CCPO_INTEN	0x50040010	CCPO Interrupt Enable Register
CCPO_INTSTS	0x50040014	CCPO Interrupt Status Register
CCPO_TMR_COUNT	0x50040018	CCPO Timer Actual Value Register
CCPO_PLOAD	0x5004001C	CCPO Pre-Load Data Register
CCPO_CCPR0	0x50040020	CCPO Compare/Capture/PWM Duty Register 0
CCPO_CCPR1	0x50040024	CCPO Compare/Capture/PWM Duty Register 1
CCPO_CCPR2	0x50040028	CCPO Compare/Capture/PWM Duty Register 2
CCPO_CCPR3	0x5004002C	CCPO Compare/Capture/PWM Duty Register 3
CCPO_PWM_DTIME	0x50040030	CCPO PWM Delay Time Register
CCP1_TMCMP_CTRL	0x50041000	CCP1 Timer Control Register
CCP1_CAP_CTRL	0x50041004	CCP1 Capture Mode Control Register
CCP1_PWM_CTRL	0x50041008	CCP1 PWM Mode Control Register
CCP1_INTEN	0x50041010	CCP1 Interrupt Enable Register
CCP1_INTSTS	0x50041014	CCP1 Interrupt Status Register
CCP1_TMR_COUNT	0x50041018	CCP1 Timer Actual Value Register
CCP1_PLOAD	0x5004101C	CCP1 Pre-Load Data Register
CCP1_CCPR0	0x50041020	CCP1 Compare/Capture/PWM Duty Register 0
CCP1_CCPR1	0x50041024	CCP1 Compare/Capture/PWM Duty Register 1
CCP1_CCPR2	0x50041028	CCP1 Compare/Capture/PWM Duty Register 2
CCP1_CCPR3	0x5004102C	CCP1 Compare/Capture/PWM Duty Register 3
CCP1_PWM_DTIME	0x50041030	CCP1 PWM Delay Time Register

Registers Function

CCP0_TMCMP_CTRL	0x50040000								CCPO Timer Control Register
CCP1_TMCMP_CTRL	0x50041000								CCP1 Timer Control Register
Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	
Bit	23	22	21	20	19	18	17	16	
Function	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	
Function	CMP3_ENA BLE	CMP2_ENA BLE	CMP1_ENA BLE	CMPO_ENA BLE	CCP1_PWM 3_TIMER_C TRL	CCP1_PWM 2_TIMER_C TRL	-	TMR_UDC_ SEL	
Reset Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
Function	TMR_CLK_SEL[3:0]				TMR_MODE_SEL[1:0]		DIRECT_WR _ENABLE	TMR_ENABLE	
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserve	-
[15]	CMP3_ENABLE	R/W	CCPx Compare Mode Channel 3 Enable Bits Note: The dead time function will be disabled automatically when timer is operating in compare mode.	0: CCPx channel 3 is disabled. 1: CCPx channel 3 is enabled.
[14]	CMP2_ENABLE	R/W	CCPx Compare Mode Channel 2 Enable Bits Note: The dead time function will be disabled automatically when timer is operating in compare mode.	0: CCPx channel 2 is disabled. 1: CCPx channel 2 is enabled.
[13]	CMP1_ENABLE	R/W	CCPx Compare Mode Channel 1 Enable Bits Note: The dead time function will be disabled automatically when timer is operating in compare mode.	0: CCPx channel 1 is disabled. 1: CCPx channel 1 is enabled.
[12]	CMPO_ENABLE	R/W	CCPx Compare Mode Channel 0 Enable Bits Note: The dead time function will be disabled automatically when timer is operating in compare mode.	0: CCPx channel 0 is disabled. 1: CCPx channel 0 is enabled.
[11]	CCP1_PWM3_TIMER_CTRL	R/W	CCP1 Channel 3 Control Selection (Only For CCP1) When this bit is set as 1, all CCP1 Channel3 control signal will refer to CCP0.	0: CCP1 channel 3 is controlled by CCP1. 1: CCP1 channel 3 is controlled by CCP0.
[10]	CCP1_PWM2_TIMER_CTRL	R/W	CCP1 Channel 2 Control Selection (Only For CCP1) When the bit set as 1, all CCP1 Channel2 control signal will refer to CCP0.	0: CCP1 channel 2 is controlled by CCP1. 1: CCP1 channel 2 is controlled by CCP0.

Bit	Function	Type	Description	Condition
[9]	-	R	Reserved	-
[8]	TMR_UDC_SEL	R/W	Timer Up/Down Count Selection Bit	0 : Down Count 1: Up Count
[7:4]	TMR_CLK_SEL[3:0]	R/W	Timer Input Clock Source Selection Bits	0000: System Clock 0001: System Clock/2 0010: System Clock/4 0011: System Clock/8 0100: System Clock/16 0101: System Clock/64 0110: System Clock/256 0111: System Clock/1024 1000: System Clock/2048 1001: System Clock/4096 1010: Iosc 32K 1011: EXTO 1100: EXT1 1101: TMA_OV 1110: TMB_OV 1111: TMC_OV
[3:2]	TMR_MODE_SEL[1:0]	R/W	Timer Operating Mode Selection Bits	00: Counter/Compare Mode 01: Capture Mode 10: PWM Mode 11: Reserved
[1]	DIRECT_WR_ENABLE	R/W	Data Direct Write Enable PLOAD and CCPRx Update Timing Control	0: The CCPx_PLOAD. PLOAD/CCPx_CCPRx.CCPRx will be updated when timer over/under flow occurs. 1: The CCPx_PLOAD. PLOAD/CCPx_CCPRx.CCPRx will be updated once CPU writes to Pload and CCPRx.
[0]	TMR_ENABLE	R/W	Timer Start Bit	0: Disabled 1: Enabled

CCP0_CAP_CTRL
0x50040004
CCP0 Capture Mode Control Register
CCP1_CAP_CTRL
0x50041004
CCP1 Capture Mode Control Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-		CAP_LPF_SEL[2:0]		CAP3_EDGE[1:0]		CAP2_EDGE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CAP1_EDGE[1:0]		CAP0_EDGE[1:0]			CAP_EN[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:15]	-	R	Reserved	-
[14:12]	CAP_LPF_SEL[2:0]	R/W	Low Pass Filter Selection Bits	000: LPF is disabled. 001: 4 Clock Cycles 010: 8 Clock Cycles 011: 16 Clock Cycles 100: 32 Clock Cycles 101: 40 Clock Cycles 110: 80 Clock Cycles 111: 128 Clock Cycles
[11:10]	CAP3_EDGE[1:0]	R/W	Capture Channel 3 Sample Edge Selection Bits	00: No Action 01: Rising Edge 10: Falling Edge 11: Both Edges
[9:8]	CAP2_EDGE[1:0]	R/W	Capture Channel 2 Sample Edge Selection Bits	00: No Action 01: Rising Edge 10: Falling Edge 11: Both Edges
[7:6]	CAP1_EDGE[1:0]	R/W	Capture Channel 1 Sample Edge Selection Bits	00: No Action 01: Rising Edge 10: Falling Edge 11: Both Edges
[5:4]	CAP0_EDGE[1:0]	R/W	Capture Channel 0 Sample Edge Selection Bits	00: No Action 01: Rising Edge 10: Falling Edge 11: Both Edges
[3:0]	CAP_EN[3:0]	R/W	Capture Mode Channel 3~0 Enable Bits 0: Disabled 1: Enabled CCP0_io_sel is determined by the GPIOFUNC=> CTRL0[30] (0x50070200) Bit[30] = 0, CAP0_IO[3:0] = IOA[6:3] Bit[30] = 1, CAP0_IO[3:0] = IOA[16:13] CCP1_io_sel is determined by the GPIOFUNC=> CTRL0[9] (0x50070200) Bit[9] = 0, CAP1_IO[3:0] = IOA[12:9] Bit[9] = 1, CAP1_IO[3:0] = IOA[29:26]	For CCP0 : Capture IO: 0001: CAPx_IO0 Enabled 0010: CAPx_IO1 Enabled 0100: CAPx_IO2 Enabled 1000: CAPx_IO3 Enabled

CCP0_PWM_CTRL								0x50040008	CCP0 PWM Mode Control Register			
CCP1_PWM_CTRL								0x50041008	CCP1 PWM Mode Control Register			
Bit	31	30	29	28	27	26	25	24				
Function	-	-	-	-	-	-	-	-				
Reset Value	0	0	0	0	0	0	0	0				
Bit	23	22	21	20	19	18	17	16				
Function	-	-	-	-	-	-	-	-				
Reset Value	0	0	0	0	0	0	0	0				
Bit	15	14	13	12	11	10	9	8				
Function	-	-	-	-	-	CCP1_DMA_EN	PWM23_CH_M	PWM01_CH_M				
Reset Value	0	0	0	0	0	0	0	0				
Bit	7	6	5	4	3	2	1	0				
Function	PWM3_POL	PWM2_POL	PWM1_POL	PWM0_POL	PWM3_ENA_BLE	PWM2_ENA_BLE	PWM1_ENA_BLE	PWM0_ENA_BLE				
Reset Value	0	0	0	0	0	0	0	0				

Bit	Function	Type	Description	Condition
[31:11]	-	R/W	Reserved	-
[10]	CCP1_DMA_EN	R/W	DMA trigger start bits for CCP1_PWM0 duty update. Note: CCP1 PWMIO can update duty via DMA but this function is only supported by CCP1's PWMIO 0 (IOA.26 or IOA.9).	0: Disabled 1: Enabled
[9]	PWM23_CHM	R/W	PWM Channel 2/3 Operating Mode Selection Bit	0 : Independent Mode 1 : Complementary Mode
[8]	PWM01_CHM	R/W	PWM Channel 0/1 Operating Mode Selection Bit	0 : Independent Mode 1 : Complementary Mode
[7]	PWM3_POL	R/W	PWM Channel 3 Polarity Control Bit	Independent Mode: 0: PWM3 outputs low when CCPX_TMR.TMR < CCPX_CCPR3.CCPR3. 1: PWM3 outputs high when CCPX_TMR.TMR < CCPX_CCPR3.CCPR3. Complementary Mode: 0: PWM3 outputs high when CCPX_TMR.TMR < CCPX_CCPR2.CCPR2. Else, it outputs low. 1: PWM3 outputs low when CCPX_TMR.TMR < CCPX_CCPR2.CCPR2. Else, it outputs high.

Bit	Function	Type	Description	Condition
[6]	PWM2_POL	R/W	PWM Channel 2 Polarity Control Bit	Independent Mode: 0: PWM2 outputs low when CCPX_TMR.TMR < CCPX_CCPR2.CCPR2. 1: PWM2 outputs high when CCPX_TMR.TMR < CCPX_CCPR2.CCPR2. Complementary Mode: 0: PWM2 outputs low when CCPX_TMR.TMR < CCPX_CCPR2.CCPR2. Else, it outputs high. 1: PWM2 outputs high when CCPX_TMR.TMR < CCPX_CCPR2.CCPR2. Else, it outputs low.
[5]	PWM1_POL	R/W	PWM Channel 1 Polarity Control Bit	Independent Mode: 0: PWM1 outputs low when CCPX_TMR.TMR < CCPX_CCPR1.CCPR1. 1: PWM1 outputs high when CCPX_TMR.TMR < CCPX_CCPR1.CCPR1. Complementary Mode: 0: PWM1 outputs high when CCPX_TMR.TMR < CCPX_CCPRO.CCPR0. Else, it outputs low. 1: PWM1 outputs low when CCPX_TMR.TMR < CCPX_CCPRO.CCPR0. Else, it outputs high.
[4]	PWMO_POL	R/W	PWM Channel 0 Polarity Control Bit	Independent Mode: 0: PWMO outputs low when CCPX_TMR.TMR < CCPX_CCPRO.CCPR0. 1: PWMO outputs high when CCPX_TMR.TMR < CCPX_CCPRO.CCPR0.

Bit	Function	Type	Description					Condition
								Complementary Mode: 0: PWM0 outputs low when CCPX_TMR.TMR < CCPX_CC PRO.CC PRO. Else, it outputs high. 1: PWM0 outputs high when CCPX_TMR.TMR < CCPX_CC PRO.CC PRO. Else, it outputs low.
[3]	PWM3_ENABLE	R/W	PWM Channel 3 Enable					0: Disabled 1: Enabled
[2]	PWM2_ENABLE	R/W	PWM Channel 2 Enable					0: Disabled 1: Enabled
[1]	PWM1_ENABLE	R/W	PWM Channel 1 Enable					0: Disabled 1: Enabled
[0]	PWM0_ENABLE	R/W	PWM Channel 0 Enable					0: Disabled 1: Enabled

CCP0_INTEN **0x50040010** **CCP0 Interrupt Enable Register**
CCP1_INTEN **0x50041010** **CCP1 Interrupt Enable Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	TMR_INT_E_NABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CAP3_INT_ENABLE	CAP2_INT_E_NABLE	CAP1_INT_E_NABLE	CAPO_INT_E_NABLE	CMP3_INT_ENABLE	CMP2_INT_ENABLE	CMP1_INT_ENABLE	CMPO_INT_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserve					-
[8]	TMR_INT_ENABLE	R/W	Timer Interrupts Enable Bit Note: This bit is used to gate interrupt signal that indicates the counter is overflow, underflow or greater than CCPx_PLOAD.PLOAD value.					0: Disabled 1: Enabled

Bit	Function	Type	Description		Condition
[7]	CAP3_INT_ENABLE	R/W	Capture Mode Channel 3 Interrupt Enable		0: Disabled 1: Enabled
[6]	CAP2_INT_ENABLE	R/W	Capture Mode Channel 2 Interrupt Enable		0: Disabled 1: Enabled
[5]	CAP1_INT_ENABLE	R/W	Capture Mode Channel 1 Interrupt Enable		0: Disabled 1: Enabled
[4]	CAPO_INT_ENABLE	R/W	Capture Mode Channel 0 Interrupt Enable		0: Disabled 1: Enabled
[3]	CMP3_INT_ENABLE	R/W	Compare Mode Channel 3 Interrupt Enable		0: Disabled 1: Enabled
[2]	CMP2_INT_ENABLE	R/W	Compare Mode Channel 2 Interrupt Enable		0: Disabled 1: Enabled
[1]	CMP1_INT_ENABLE	R/W	Compare Mode Channel 1 Interrupt Enable		0: Disabled 1: Enabled
[0]	CMPO_INT_ENABLE	R/W	Compare Mode Channel 0 Interrupt Enable		0: Disabled 1: Enabled

CCP0_INTSTS
0x50040014
CCP0 Interrupt Status Register
CCP1_INTSTS
0x50041014
CCP1 Interrupt Status Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				TMR_INT_F LAG
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CAP3_INT_F LAG	CAP2_INT_F LAG	CAP1_INT_F LAG	CAPO_INT_F LAG	CMP3_INT_F FLAG	CMP2_INT_F FLAG	CMP1_INT_F FLAG	CMPO_INT_F FLAG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description		Condition
[31:9]	-	R/W	Reserved		-
[8]	TMR_INT_FLAG	R/W	Timer Overflow/Underflow Interrupts Flag Note: This bit will be triggered when the counter is overflow, underflow or greater than CCPX_PLOAD.PLOAD value.		Read 0: Idle/Busy Read 1: Timer interrupt is triggered. Write 0: No effect. Write 1: Clear these bits.

Bit	Function	Type	Description	Condition
[7]	CAP3_INT_FLAG	R/W	Capture Mode Channel 3 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No capture event is triggered. Read 1: Capture event is triggered. Write 0: No effect. Write 1: Clear these bits.
[6]	CAP2_INT_FLAG	R/W	Capture Mode Channel 2 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No capture event is triggered. Read 1: Capture event is triggered. Write 0: No effect. Write 1: Clear these bits.
[5]	CAP1_INT_FLAG	R/W	Capture Mode Channel 1 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No capture event is triggered. Read 1: Capture event is triggered. Write 0: No effect. Write 1: Clear these bits.
[4]	CAP0_INT_FLAG	R/W	Capture Mode Channel 0 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No capture event is triggered. Read 1: Capture event is triggered. Write 0: No effect. Write 1: Clear these bits.
[3]	CMP3_INT_FLAG	R/W	Compare Mode Channel 3 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No counting is equivalent triggered. Read 1: Counting equivalent is triggered. Write 0: No effect. Write 1: Clear these bits.
[2]	CMP2_INT_FLAG	R/W	Compare Mode Channel 2 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No counting equivalent is triggered. Read 1: Counting equivalent is triggered. Write 0: No effect. Write 1: Clear these bits.
[1]	CMP1_INT_FLAG	R/W	Compare Mode Channel 1 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.	Read 0: No counting equivalent is triggered. Read 1: Counting equivalent is triggered. Write 0: No effect. Write 1: Clear these bits.

Bit	Function	Type	Description					Condition
[0]	CMPO_INT_FLAG	R/W	Compare Mode Channel 0 Interrupt Flag Note: This bit is available when the timer is not operated in concatenated mode.					Read 0: No counting equivalent is triggered. Read 1: Counting equivalent is triggered. Write 0: No effect. Write 1: Clear these bits.

CCP0_TMR_COUNT **0x50040018** **CCP0 Timer Actual Value Register**

CCP1_TMR_COUNT **0x50041018** **CCP1 Timer Actual Value Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	TMR[15:8]	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	TMR[7:0]	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserve					-
[15:0]	TMR[15:0]	R	Timer Actual Value					-

CCP0_PLOAD **0x5004001C** **CCP0 Pre-Load Data Register**

CCP1_PLOAD **0x5004101C** **CCP1 Pre-Load Data Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	TMR_PLOAD[15:8]	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	TMR_PLOAD[7:0]	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserve					-
[15:0]	TMR_PLOAD[15:0]	R/W	Timer Pre-Load Data or PWM Period Data Register Note: CCPX_PLOAD.PLOAD will be filled in CCPX_TMR.TMR when the timer is underflow and CCPX_TMR_CTRL.TMR_UDC_SEL is set as 1.					-

CCP0_CCPR0 **0x50040020** **CCP0 Compare/Capture/PWM Duty Register 0**
CCP1_CCPR0 **0x50041020** **CCP1 Compare/Capture/PWM Duty Register 0**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserve					-
[15:0]	CCP0[15:0]	R/W	Compare/Capture Value Register 0 In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch CCPX_TMR.TMR data.					-

CCP0_CCPR1 **0x50040024** **CCP0 Compare/Capture/PWM Duty Register 1**
CCP1_CCPR1 **0x50041024** **CCP1 Compare/Capture/PWM Duty Register 1**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CCPR1[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserve						-
[15:0]	CCPR1[15:0]	R/W	Compare/Capture Value Register 1 In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch CCPX_TMR.TMR data.						-

CCP0_CCPR2
0x50040028
CCP0 Compare/Capture/PWM Duty Register 2
CCP1_CCPR2
0x50041028
CCP1 Compare/Capture/PWM Duty Register 2

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	CCPR2[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CCPR2[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserve						-
[15:0]	CCPR2[15:0]	R/W	Compare/Capture Value Register 2 In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch CCPX_TMR.TMR data.						-

CCP0_CCPR3
0x5004002C
CCP0 Compare/Capture/PWM Duty Register 3
CCP1_CCPR3
0x5004102C
CCP1 Compare/Capture/PWM Duty Register 3

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	CCPR3[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CCPR3[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserve						-
[15:0]	CCPR3[15:0]	R/W	Compare/Capture Value Register 3 In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch CCPX_TMR.TMR data.						-

CCP0_PWM_DTIME								
0x50040030								
CCP0 PWM Delay Time Register								
Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	PWM_DTIME_1[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PWM_DTIME_0[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserve						-
[15:8]	PWM_DTIME_1[7:0]	R/W	PWM Delay Time Register for PWM 1 and PWM3 Channel These bits are available when PWM is operating in complementary mode. Complementary Mode: In the same CCP, the signals in PWM0&1 or PWM2&3 can be set as complementary to or independent of each other. CCP0's PWM0 and PWM2 are master pins, and PWM1 and PWM3 are slave pins.						-

Bit	Function	Type	Description	Condition
			<p>[15:8]: Settings of the delay time at slave pin's (PWM1 and 3) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, if PLOAD is 63 (64-1=0~63, total of 64-level), the time unit of DTIME is 1/64.</p>	
[7:0]	PWM_DTIME_0[7:0]	R/W	<p>PWM Delay Time Register for PWM0 and PWM2 Channel</p> <p>These bits are available when PWM is operating in complementary mode.</p> <p>[7:0]: Settings of the delay time at master pin's (PWM0 and 2) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, if PLOAD is 63 (64-1=0~63, total of 64-level), the time unit of DTIME is 1/64.</p>	-

CCP1_PWM_DTIME
0x50041030
CCP1 PWM Delay Time Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserve	-
[15:8]	PWM_DTIME_1[7:0]	R/W	<p>PWM Delay Time Register for PWM 1 and PWM3 Channel</p> <p>These bits are available when PWM is operating in complementary mode.</p> <p>Complementary Mode: In the same CCP, it allows the signals in PWM0&1 or PWM2&3 to be set as complementary to or independent of each other.</p>	-

Bit	Function	Type	Description	Condition
			<p>CCP1's PWM0 and PWM2 are master pins, and PWM1 and PWM3 are slave pins.</p> <p>[15:8]: Settings of the delay time at slave pin's (PWM1 and 3) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, if PLOAD is 63 (64-1=0~63, total of 64-level), the time unit of DTIME is 1/64.</p>	
[7:0]	PWM_DTIME_0[7:0]	R/W	<p>PWM Delay Time Register for PWM0 and PWM2 Channel</p> <p>These bits are available when PWM is operating in complementary mode.</p> <p>[7:0]: Settings of the delay time at master pin's (PWM0 and 2) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, if PLOAD is 63 (64-1=0~63, total of 64-level), the time unit of DTIME is 1/64.</p>	-

10. MAC Controller

10.1. Introduction

The MAC controller is designed to accelerate the audio MAC process of 16b system. In GPCM3, not only MAC is compatible with GPCM3's MAC, the MAC operation can also be extended after enhance mode (MAC_CTRL[31] = 1b) is enabled. The comparison is indicated in Table 10-1: Comparison between MAC Normal and Enhance Mode. In addition to the MAC function, it also has two sets of 32-bit sign data * 16-bit sign data hardware multiplier, a 32-bit unsigned data / 32bit unsigned data hardware divisor, and an EXP calculator to derive the effective exponent for preparing the normalization operation.

In the hardware multiplier, after user fills data into the MULx multiplier register (MAC_MUL12_C16), the result can be obtained from the register immediately. The hardware divider starts computing while the data is being written into the divisor register. After the DIV_BUSY flag of polling MAC_MUL_DIV_STS becomes 0, the result of quotient and remainder can be obtained from the register.

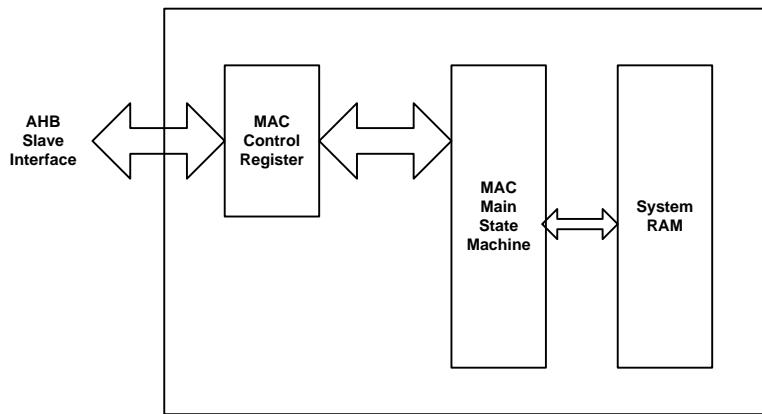
Table 10-1: Comparison between MAC Normal and Enhance Mode

MAC_CTRL[31] = 0b		MAC_CTRL[31] = 1b	
Name	Location	Name	Location
OUTPUT_SHIFT[4:0]	MAC_CTRL[12:8]	ENH_OUTPUT_SHIFT[5:0]	MAC_CTRL[13:8]
ADDRX_STEP[4:0]	MAC_STEP_XY[4:0]	ENH_ADDRX_STEP[9:0]	MAC_STEP_XY[9:0]
ADDY_STEP[4:0]	MAC_STEP_XY[12:8]	ENH_ADDY_STEP[9:0]	MAC_STEP_XY[25:16]
-	-	MAC_OUTPUT_R[31:0]	MAC_OUTPUT_R[31:0]
-	-	MAC_OUTPUT_EXT8_R[7:0]	MAC_OUTPUT_EXT8_R[7:0]
-	-	MAC_SATURATION_R[31:0]	MAC_SATURATION_R[31:0]
-	-	MAC_PRELOAD_DATA[31:0]	MAC_PRELOAD_DATA[31:0]
-	-	MAC_START_CNT[15:0]	MAC_START_CNT[15:0]

10.2. Features

- Support maximum 16x16 MAC function.
- Support programmable address increment or decrement per MAC operation.
- Support 8/16/32b saturation detection.
- Support 40b accumulator.
- Support programmable start address without align limitation.
- Support output auto shift function.
- Support two sets of 32b sign data * 16b sign data hardware multiplier.
- Support 32b unsigned data / 32b unsigned data hardware multiplier.
- Support Exponent calculator.

10.3. Block Diagram



10.4. Register Description

10.4.1. Registers Map

Name	Address	Description
MAC_CTRL	0x5003_0000	MAC Control Register
MAC_LENGTH	0x5003_0004	MAC Length Register
MAC_ADDR_X	0x5003_0008	MAC Start Address X Register
MAC_ADDR_Y	0x5003_000C	MAC Start Address Y Register
MAC_STEP_XY	0x5003_0010	MAC Step X/Y Register
MAC_OUTPUT	0x5003_0014	MAC Output Register
MAC_OUTPUT_EXT8	0x5003_0018	MAC Output Extent Register
MAC_SATURATION	0x5003_001C	MAC OUT Saturation Register
MAC_INSTS	0x5003_0020	MAC Interrupt Status Register
MAC_OUTPUT_R	0x5003_0024	MAC Output Register
MAC_OUTPUT_EXT8_R	0x5003_0028	MAC Output Extent Register
MAC_SATURATION_R	0x5003_002C	MAC OUT Saturation Register
MAC_PRELOAD_DATA	0x5003_0030	MAC Initialized Register
MAC_START_CNT	0x5003_0034	MAC Counter
MAC_MUL_DIV_STS	0x5003_0038	Multiplier/Divider Status Register
MAC_MUL1_X32	0x5003_003C	MUL1 Multiplicand Register
MAC_MUL2_X32	0x5003_0040	MUL2 Multiplicand Register
MAC_MUL12_C16	0x5003_0044	MUL1/MUL2 Multiplier Register
MAC_MUL1_OUT	0x5003_0048	MUL1 Product Register
MAC_MUL2_OUT	0x5003_004C	MUL2 Product Register
MAC_DIV_DIVIDEND	0x5003_0050	Dividend Register
MAC_DIV_DIVISOR	0x5003_0054	Divisor Register
MAC_DIV_Q	0x5003_0058	Quotient Register
MAC_DIV_R	0x5003_005C	Remainder Register
MAC_EXP_DATA	0x5003_0060	EXP Data Register
MAC_EXP_RESULT	0x5003_0064	EXP Result Register

10.4.2. Registers Function

MAC_CTRL									0x5003_0000									MAC Control Register									
Bit	31	30	29	28	27	26	25	24																			
Function	MODE_SEL_CTRL	-	ENH_DIV_I NT_EN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	OUTPUT_SHIFT[4:0]/ENH_OUTPUT_SHIFT[5:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	INT_EN	OPERATION_MODE_SEL	SATURATION_MODE_SEL[1:0]	-	-	-	RESET	START
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31]	MODE_SEL_CTRL	R/W	Normal Mode and Enhanced Mode Selection	0: Normal Mode 1: Enhance Mode
[30]	-	R	Reserved	-
[29]	ENH_DIV_INT_EN	R/W	Divider Interrupt Control	0: Disabled 1: Enabled
[28:14]	-	R	Reserved	-
[13:8]	OUTPUT_SHIFT[4:0] /ENH_OUTPUT_SHIF T[5:0]	R/W	MAC Output Shift Control	Refer to Table 10-2: MAC Output Shift Setting Table.
[7]	INT_EN	R/W	Interrupt Enable When this bit is set as 1, MAC will issue an interrupt to inform the CPU when it completes the calculation. Else the CPU has to check whether the MAC calculation is done or not by polling "START/BUSY".	0: Disabled 1: Enabled
[6]	OPERATION_MODE_SEL	R/W	MAC Calculate by Signed or Unsigned Mode	0: MAC+=signed(X) * signed(Y) 1: MAC+= signed(X) * unsigned(Y)
[5:4]	SATURATION_MODE_SEL[1:0]	R/W	Saturation Mode Control Register	0: 16-bit Saturation Mode 1: 32-bit Saturation Mode 2: 8-bit Saturation Mode 3: Reserved
[3:2]	-	R	Reserved	-
[1]	RESET	R/W	MAC RESET Control MAC Controller reset register. Writing 1 to this bit, and the MAC engine will be reset to default value immediately. This bit will be cleared immediately after the MAC controller is reset.	Read 0: Reset is done. Read 1: Busy Write 0: No effect. Write 1: Reset MAC.

Bit	Function	Type	Description	Condition
[0]	START	R/W	MAC START/BUSY Control When this bit is set as 1, MAC controller will start to perform MAC calculation. The length of the calculation is defined in P_MAC_LENGTH register.	Read 0: Idle Read 1: Busy Write 0: No effect. Write 1: MAC starts.

Table 10-2: MAC Output Shift Setting Table

	Normal Mode OUTPUT_SHIFT[4:0]	Enhance Mode ENH_OUTPUT_SHIFT[5:0]
0x0	MAC_OUT right shifts 0 step.	MAC_OUT right shifts 0 step.
0x1	MAC_OUT right shifts 1 step.	MAC_OUT right shifts 1 step.
0x2	MAC_OUT right shifts 2 steps.	MAC_OUT right shifts 2 steps.
0x3	MAC_OUT right shifts 3 steps.	MAC_OUT right shifts 3 steps.
0x4	MAC_OUT right shifts 4 steps.	MAC_OUT right shifts 4 steps.
0x5	MAC_OUT right shifts 5 steps.	MAC_OUT right shifts 5 steps.
0x6	MAC_OUT right shifts 6 steps.	MAC_OUT right shifts 6 steps.
0x7	MAC_OUT right shifts 7 steps.	MAC_OUT right shifts 7 steps.
0x8	MAC_OUT right shifts 8 steps.	MAC_OUT right shifts 8 steps.
0x9	MAC_OUT right shifts 9 steps.	MAC_OUT right shifts 9 steps.
0xA	MAC_OUT right shifts 10 steps.	MAC_OUT right shifts 10 steps.
0xB	MAC_OUT right shifts 11 steps.	MAC_OUT right shifts 11 steps.
0xC	MAC_OUT right shifts 12 steps.	MAC_OUT right shifts 12 steps.
0xD	MAC_OUT right shifts 13 steps.	MAC_OUT right shifts 13 steps.
0xE	MAC_OUT right shifts 14 steps.	MAC_OUT right shifts 14 steps.
0xF	MAC_OUT right shifts 15 steps.	MAC_OUT right shifts 15 steps.
0x10	MAC_OUT right shifts 16 steps.	MAC_OUT right shifts 16 steps.
0x11	No Action	MAC_OUT right shifts 17 steps.
0x12	No Action	MAC_OUT right shifts 18 steps.
0x13	No Action	MAC_OUT right shifts 19 steps.
0x14	No Action	MAC_OUT right shifts 20 steps.
0x15	No Action	MAC_OUT right shifts 21 steps.
0x16	No Action	MAC_OUT right shifts 22 steps.
0x17	No Action	MAC_OUT right shifts 23 steps.
0x18	No Action	MAC_OUT right shifts 24 steps.
0x19	No Action	MAC_OUT right shifts 25 steps.
0x1A	No Action	MAC_OUT right shifts 26 steps.
0x1B	No Action	MAC_OUT right shifts 27 steps.
0x1C	No Action	MAC_OUT right shifts 28 steps.
0x1D	No Action	MAC_OUT right shifts 29 steps.
0x1E	No Action	MAC_OUT right shifts 30 steps.
0x1F	MAC_OUT left shifts 1 step.	MAC_OUT right shifts 31 steps.
0x20	No Action	MAC_OUT right shifts 32 steps.
0x21~0x3B	No Action	No Action
0x3C	No Action	MAC_OUT left shifts 4 steps.
0x3D	No Action	MAC_OUT left shifts 3 steps.
0x3E	No Action	MAC_OUT left shifts 2 steps.
0x3F	No Action	MAC_OUT left shifts 1 step.

MAC_LENGTH
0x5003_0004
MAC Length Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	LENGTH[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	LENGTH[7:0]	R/W	MAC Length This register defines how many times does the register perform MAC operation.						0: 1 MAC operation 1: 2 MAC operations ... 255: 256 MAC operations

MAC_ADDR_X
0x5003_0008
MAC Start Address X Register

Bit	31	30	29	28	27	26	25	24
Function	MAC_ADDRx[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MAC_ADDRx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MAC_ADDRx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:24]	MAC_ADDRx[31:24]	R/W	The value in this register is the current address in SRAM, and the unit is byte. The value in register will be update automatically after user reads the result from the result port when MODE is set as 0.						-
[23:16]	-	R	Reserved						-

Bit	Function	Type	Description					Condition
[15:0]	MAC_ADDRx[15:0]	R/W	The value in this register is the current address in SRAM, and the unit is byte. Address[23:16] in MAC_ADDRx is reserved.					-

MAC_ADDR_Y 0x5003_000C MAC Start Address Y Register								
Bit	31	30	29	28	27	26	25	24
Function	MAC_ADDRy[31:24]							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	MAC_ADDRy[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	MAC_ADDRy[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:24]	MAC_ADDRy[31:24]	R/W	The value in this register is the current address in SRAM, and the unit is byte.					-
[23:16]	-	R	Reserved					-
[15:0]	MAC_ADDRy[15:0]	R/W	The value in this register is the current address in SRAM, and the unit is byte. Address[23:16] in MAC_ADDRy is reserved.					-

MAC_STEP_XY (Normal mode) 0x5003_0010 MAC Step X/Y Register								
Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	- ADDRy_STEP[4:0]							
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	- ADDRx_STEP[4:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:13]	-	R	Reserved	-
[12:8]	ADDRY_STEP[4:0]	R/W	This register determines the AddrY address will be increased or decreased after every MAC operation.	This is a 5-bits signed register. 0x10: -16 0x00: 0 0x0F: +15
[7:5]	-	R	Reserved	-
[4:0]	ADDRX_STEP[4:0]	R/W	This register determines the AddrX address will be increased or decreased after every MAC operation.	This is a 5-bits signed register. 0x10: -16 0x00: 0 0x0F: +15

MAC_STEP_XY (Enhance mode)
0x5003_0010
MAC Step X/Y Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	ENH_ADDRY_STEP[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENH_ADDRX_STEP[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:26]	-	R	Reserved	-
[25:16]	ENH_ADDRY_STEP [9:0]	R/W	This register determines the AddrY address will be increased or decreased after every MAC operation.	This is a 10-bits signed register. 0x200: -512 0x000: 0 0x1FF: +511
[15:10]	-	R	Reserved	-
[9:0]	ENH_ADDRX_STEP [9:0]	R/W	This register determines the AddrX address will be increased or decreased after every MAC operation.	This is a 10-bits signed register. 0x200: -512 0x000: 0 0x1FF: +511

MAC_OUTPUT
0x5003_0014
MAC Output Register

Bit	31	30	29	28	27	26	25	24
Function	MAC_OUT[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	MAC_OUT[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MAC_OUT[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MAC_OUT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	MAC_OUT[31:0]	R/W	MAC Output Result Port The value in this register is bit[31:0] of the MAC 40-bits result.					When user writes 1 to START bit and the MAC operation is done, this register will be updated.

MAC_OUTPUT_EXT8
0x5003_0018
MAC Output Extent Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MAC_OUT_EXT8[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	MAC_OUT_EXT8[7:0]	R/W	MAC Output Result Port The value in this register is bit[39:32] of the MAC 40-bits result.					When user writes 1 to START bit and the MAC operation is done, this register will be updated.

MAC_SATURATION
0x5003_001C
MAC OUT Saturation Register

Bit	31	30	29	28	27	26	25	24
Function	MAC_SAT_OUT[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	MAC_SAT_OUT[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MAC_SAT_OUT[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MAC_SAT_OUT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	MAC_SAT_OUT[31:0]	R/W	MAC Saturation Output Result Port This register is used to store the saturation result of MAC_OUT.					The actual result depends on SATMODE setting. This register will be updated when MAC_OUT is updated.

MAC_INTSTS
0x5003_0020
MAC Interrupt Status Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:1]	-	R	Reserved					-
[0]	MAC_INT	R/W	MAC Interrupt Status Flag					Read 0: Idle/Busy Read 1: Interrupt occurs. Write 0: No effect. Write 1: Clear this bit.

MAC_OUTPUT_R
0x5003_0024
MAC Output Register

Bit	31	30	29	28	27	26	25	24
Function	MAC_OUT_R[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	MAC_OUT_R[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MAC_OUT_R[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MAC_OUT_R[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:0]	MAC_OUT_R [31:0]	R	MAC Output Result Port The value in this register is bit[31:0] of the MAC 40-bits result.						-

MAC_OUTPUT_EXT8_R
0x5003_0028
MAC Output Extent Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MAC_OUT_EXT8_R[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	MAC_OUT_EXT8_R[7:0]	R	MAC Enhance Mode Output Result Port The value in this register is bit[39:32] of the MAC 40-bits result.						-

MAC_SATURATION_R
0x5003_002C
MAC OUT Saturation Register

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

Bit	Function	Type	Description						Condition
[31:0]	MAC_SAT_OUT_R[31:0]	R	MAC Enhance Mode Saturation Output Result Port This register is used to store the saturation result of MAC_OUT.						-

MAC_PRELOAD_DATA
0x5003_0030
MAC Initialized Register

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

Bit	Function	Type	Description						Condition
[31:0]	PRELOAD_DATA[31:0]	R/W	MAC's Initial Value						-

MAC_START_CNT
0x5003_0034
MAC Counter

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	START_CNT[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	START_CNT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	START_CNT[15:0]	R/W	Each time MAC performs multiply-accumulate, the counter will increment by 1.					Read: The current count of MAC. Write: Reset counter.

MAC_MUL_DIV_STS **0x5003_0038** **Multiplier/Divider Status Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0	
Function	-					DIV_BUSY	DIV0_FLAG	MUL2_BUSY	MUL1_BUSY
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[31:4]	-	R	Reserved					-
[3]	DIV_BUSY	R	Divider's Busy Flag					0: Idle 1: Busy
[2]	DIV0_FLAG	R	Flag of whether Divisor is 0. Note: If Divisor is 0, this flag = 1.					0: Divisor is not 0. 1: Divisor is 0.
[1]	MUL2_BUSY	R	MUL2's Busy Flag					0: Idle 1: Busy
[0]	MUL1_BUSY	R	MUL1's Busy Flag					0: Idle 1: Busy

MAC_MUL1_X32
0x5003_003C
MUL1 Multiplicand Register

Bit	31	30	29	28	27	26	25	24
Function	MUL1_X32[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	MUL1_X32[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MUL1_X32[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MUL1_X32[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	MUL1_X32[31:0]	R/W	MUL1 Multiplicand Register					The data in MUL1_X32 is 32-bits Signed Value.

MAC_MUL2_X32
0x5003_0040
MUL2 Multiplicand Register

Bit	31	30	29	28	27	26	25	24
Function	MUL2_X32[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	MUL2_X32[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MUL2_X32[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MUL2_X32[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	MUL2_X32[31:0]	R/W	MUL2 Multiplicand Register					The data in MUL2_X32 is 32-bits Signed Value.

MAC_MUL12_C16
0x5003_0044
MUL1/MUL2 Multiplier Register

Bit	31	30	29	28	27	26	25	24
Function	MUL2_C16[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	MUL2_C16[7:0]							
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	MUL1_C16[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	MUL1_C16[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	MUL2_C16[15:0]	R/W	MUL2 Multiplier Register After data is written in this register, the multiplier will be activated.					The data in MUL2_C16 is 16-bits Signed Value.
[15:0]	MUL1_C16[15:0]	R/W	MUL1 Multiplier Register After data is written in this register, the multiplier will be activated.					The data in MUL1_C16 is 16-bits Signed Value.

MAC_MUL1_OUT
0x5003_0048
MUL1 Product Register

Bit	31	30	29	28	27	26	25	24
Function	MUL1_OUT[31:24]							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	MUL1_OUT[23:16]							
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	MUL1_OUT[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	MUL1_OUT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	MUL1_OUT[31:0]	R	The place stores (32-bit sign data) * (16-bit sign data). The operation result is 32-bits signed data, and the data exceeds this range will be deleted automatically.					-

MAC_MUL2_OUT
0x5003_004C
MUL2 Product Register

Bit	31	30	29	28	27	26	25	24
Function	MUL2_OUT[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	MUL2_OUT[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MUL2_OUT[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	MUL2_OUT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:0]	MUL2_OUT[31:0]	R	The place stores (32-bit sign data) * (16-bit sign data). The operation result is 32-bits signed data, and the data exceeds this range will be deleted automatically.						-

MAC_DIV_DIVIDEND
0x5003_0050
Dividend Register

Bit	31	30	29	28	27	26	25	24
Function	DIV_DIVIDEND[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DIV_DIVIDEND[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DIV_DIVIDEND[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DIV_DIVIDEND[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:0]	DIV_DIVIDEND[31:0]	R/W	Divider's Dividend Register						The data in DIV_DIVIDEND is 32-bits Unsigned data.

MAC_DIV_DIVISOR
0x5003_0054
Divisor Register

Bit	31	30	29	28	27	26	25	24
Function	DIV_DIVISOR[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DIV_DIVISOR[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DIV_DIVISOR[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DIV_DIVISOR[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	DIV_DIVISOR[31:0]	R/W	Divider's Divisor Register					The data in DIV_DIVISOR is 32-bits Unsigned data.

MAC_DIV_Q								
0x5003_0058								
Bit	31	30	29	28	27	26	25	24
Function	DIV_Q[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DIV_Q[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DIV_Q[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DIV_Q[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	DIV_Q[31:0]	R	Divider's Quotient Register					The data in DIV_Q is 32-bits Unsigned data.

MAC_DIV_R								
0x5003_005C								
Bit	31	30	29	28	27	26	25	24
Function	DIV_R[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DIV_R[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DIV_R[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DIV_R[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	DIV_R[31:0]	R	Divider's Remainder Register					The data in DIV_R is 32-bits Unsigned data.

MAC_EXP_DATA								
Bit	31	30	29	28	27	26	25	24
Function	EXP_DATA[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	EXP_DATA[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	EXP_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	EXP_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	31:0	Function	Type	Description					Condition
[31:0]	EXP_DATA[31:0]	R/W	EXP_DATA can calculate how many bits the data has to be shifted and aligned. For example, when the data has to be left aligned, user is able to calculate how many useless data is on the left of the data, and the data has to be shifted to the left for how many bits.					The data in EXP_DATA is 32-bits Signed data.	

MAC_EXP_RESULT								
Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	31:5	Function	Type	Description					Condition
[31:5]	-	R	Reserved					-	
[4:0]	EXP_RESULT[4:0]	R	Result of the effective exponent.					-	

11. Timer

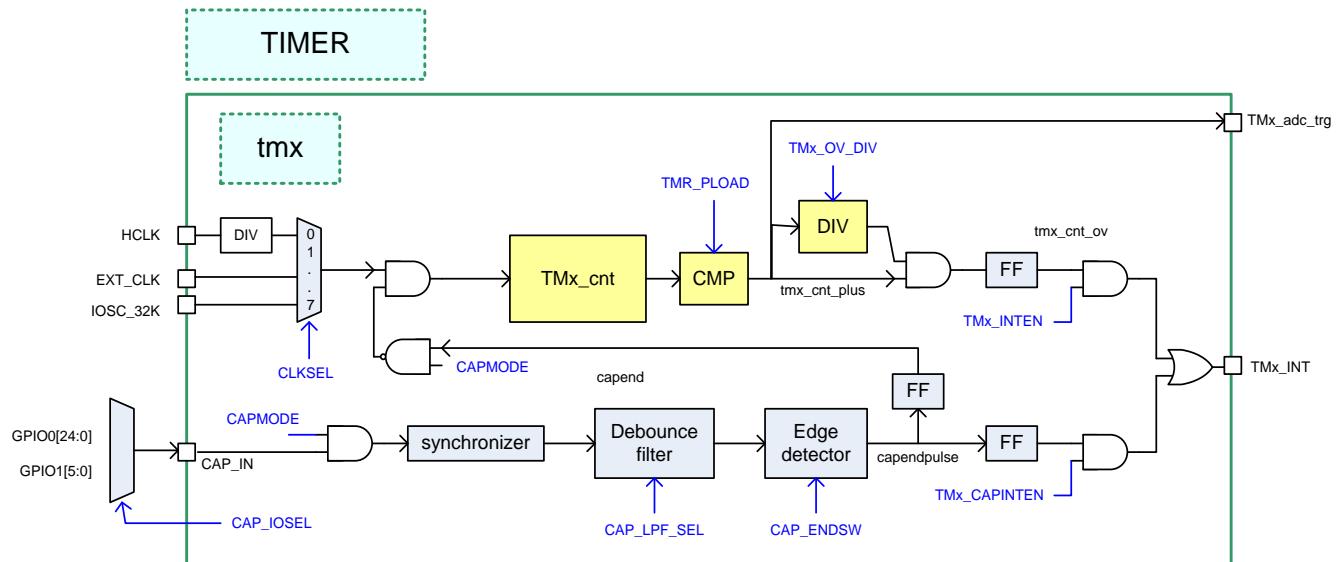
11.1. Introduction

The GPCM3 SERIALS provide 3 general 16-bit timers, counter mode and capture mode function.

11.2. Feature

- Support three independent 16-bit timers.
- Support timer trigger source to ADC/DAC.
- Support external clock source from GPIO.
- Support GPIO capture mode with De-bounce filter.

11.3. Block Diagram



11.4. Function

Timerx (x: 0~2) supports two modes: counter mode and capture mode.

Counter Mode:

When the TMR_EN is set as 1, the TMR will reload 0 and start up the counter. When the timer counter equals the TMR_PLOAD value, it will send out timer overflow event. The timer has an interrupt enable and an interrupt flag, and the TMR next count will return zero.

Capture Mode:

In capture mode, each TMx supports 1 capture channels in capture mode. Each capture channel has an enable signal to control the on and off of the function. In addition, it also has an interrupt enable and an interrupt flag.

11.5. Register Description

Registers Map

Name	Address	Description
TM0_TMR_PLOAD	0x50050000	Timer0 Pre-Load Data Register
TM0_TMR_COUNT	0x50050004	Timer0 Actual Value Register
TM0_CTRL	0x50050008	Timer0 Control Register

Name	Address	Description
TM0_CAPSRC	0x5005000C	Timer0 Capture Mode Source Selection Register
TM1_TMR_PLOAD	0x50050010	Timer1 Pre-Load Data Register
TM1_TMR_COUNT	0x50050014	Timer1 Actual Value Register
TM1_CTRL	0x50050018	Timer1 Control Register
TM1_CAPSRC	0x5005001C	Timer1 Capture Mode Source Selection Register
TM2_TMR_PLOAD	0x50050020	Timer2 Pre-Load Data Register
TM2_TMR_COUNT	0x50050024	Timer2 Actual Value Register
TM2_CTRL	0x50050028	Timer2 Control Register
TM2_CAPSRC	0x5005002C	Timer2 Capture Mode Source Selection Register
TM_INT_INTEN	0x50050030	Timer Interrupt Enable Register
TM_INT_INTSTS	0x50050034	Timer Interrupt Status Register

Registers Function
TM0_TMR_PLOAD **0x50050000** **Timer0 Pre-Load Data Register**
TM1_TMR_PLOAD **0x50050010** **Timer1 Pre-Load Data Register**
TM2_TMR_PLOAD **0x50050020** **Timer2 Pre-Load Data Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	PRELOAD_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Function	PRELOAD_DATA[7:0]							
Reset Value	1	1	1	0	1	0	0	0

Bit	Function	Type	Description				Condition
[31:16]	-	R	Reserved				-
[15:0]	PRELOAD_DATA[15:0]	R/W	Timer Pre-Load Data Register It will count from 0 to TMR_PLOAD value, and then trigger the timer overflow. TMR will thus return to zero. Note: Because Timer counter starts counting from 0, to count for N timers, user has to fill (N -1) in PLOAD.				-

TM0_TMR_COUNT	0x50050004	Timer0 Actual Value Register
TM1_TMR_COUNT	0x50050014	Timer1 Actual Value Register
TM2_TMR_COUNT	0x50050024	Timer2 Actual Value Register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	TMR[15:0]	R/W	Timer Value Register Count mode is used to play the current Timer count value. Capture mode is used to store Capture value. When CAP_START = 1, it will be cleared as 0.						-

TM0_CTRL	0x50050008	Timer0 Control Register
TM1_CTRL	0x50050018	Timer1 Control Register
TM2_CTRL	0x50050028	Timer2 Control Register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	CAP_ENDS_W	CAP_END	CAP_START	CAPMODE
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TM_RELOAD	TM_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:12]	TMINT_DIV[3:0]	R/W	<p>TIMER X INTERRUPT Divider</p> <p>With INT divider mechanism, TIMER X Interrupt is able to determine to assert the interrupt after the times of interrupt is divided by 1~16.</p>	0: Original Interrupt 1: Divided by 2 and the interrupt is asserted. 2: Divided by 3 and the interrupt is asserted ... 15: Divided by16 and the interrupt is asserted.
[11]	CAP_ENDSW	R/W	Capture End Edge Switch	0: Rise Edge 1: Fall Edge
[10]	CAP_END	R/W	<p>Capture End</p> <p>In Capture mode, when capture end edge happens, this bit will be tied as 1 and will stop TMR counter. It can be cleared by CPU.</p>	-
[9]	CAP_START	R/W	<p>Capture Start Control Bit</p> <p>This bit only works in capture mode. Writing 1 to start TMR capture function, and TMR count value will be cleared as 0. It will be cleared automatically when TMR clear finishes.</p>	Write 0: No effect. Write 1: Capture starts.
[8]	CAPMODE	R/W	Capture Mode Enable Bits	0: Disabled 1: Enabled
[7:5]	CLK_SEL[2:0]	R/W	Timer Input Clock Source Selection Bits	000 = HCLK 001 = HCLK/2 010 = HCLK/4 011 = HCLK/8 100 = HCLK/16 101 = HCLK/32 110 = External Clock 111 = IOSC32KHz
[4:2]	LPF_SEL[2:0]	R/W	Low Pass Filter Selection Bits	000 = LPF is disabled. 001 = 4 Clock Cycles 010 = 8 Clock Cycles 011 = 16 Clock Cycles 100 = 32 Clock Cycles 101 = 40 Clock Cycles 110 = 80 Clock Cycles 111 = 128 Clock Cycles
[1]	TM_RELOAD	R/W	<p>Timer Reload</p> <p>Writing 1 to this bit, and TMR will be cleared as 0. It will be cleared automatically when reload is done.</p>	-

Bit	Function	Type	Description	Condition
[0]	TM_ENABLE	R/W	Timer Enable Bit	0: Disabled 1: Enabled

TM0_CAPSRC 0x5005000C Timer0 Capture Mode Source Selection Register

TM1_CAPSRC 0x5005001C Timer1 Capture Mode Source Selection Register

TM2_CAPSRC 0x5005002C Timer2 Capture Mode Source Selection Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:5]	-	R	Reserved	-
[4:0]	IOSEL[4:0]	R/W	TIMER X Capture Mode Source GPIO Selection	0x0 = IOA0 0x1 = IOA1 0x2 = IOA2 0x3 = IOA3 0x4 = IOA4 0x5 = IOA5 0x6 = IOA6 0x7 = IOA7 0x8 = IOA8 0x9 = IOA9 0xA = IOA10 0xB = IOA11 0xC = IOA12 0xD = IOA13 0xE = IOA14 0xF = IOA15 0x10 = IOA16 0x11 = IOA17 0x12 = IOA18 0x13 = IOA19 0x14 = IOA20 0x15 = IOA21 0x16 = IOA22

Bit	Function	Type	Description					Condition
								0x17 = IOA23 0x18 = IOA24 0x19 = IOA25 0x1A = IOA26 0x1B = IOA27 0x1C = IOA28 0x1D = IOA29 0x1E = IOA30 0x1F = IOA31

TM_INT_INTEN **0x50050030** **Timer Interrupt Enable Register**

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-		TM2_CAPINT_ENABLE	TM2_INT_ENABLE	TM1_CAPINT_ENABLE	TM1_INT_ENABLE	TMO_CAPINT_ENABLE	TMO_INT_ENABLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:6]	-	R	Reserved					-
[5]	TM2_CAPINT_ENABLE	R/W	TM2 Capture Mode Interrupt Enable					0: Disabled 1: Enabled
[4]	TM2_INT_ENABLE	R/W	TM2 Interrupt Enable					0: Disabled 1: Enabled
[3]	TM1_CAPINT_ENABLE	R/W	TM1 Capture Mode Interrupt Enable					0: Disabled 1: Enabled
[2]	TM1_INT_ENABLE	R/W	TM1 Interrupt Enable					0: Disabled 1: Enabled
[1]	TMO_CAPINT_ENABLE	R/W	TMO Capture Mode Interrupt Enable					0: Disabled 1: Enabled
[0]	TMO_INT_ENABLE	R/W	TMO Interrupt Enable					0: Disabled 1: Enabled

TM_INT_INTSTS
0x50050034
Timer Interrupt Status Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function			TM2_CAPIN T_FLAG	TM2_INT_F LAG	TM1_CAPIN T_FLAG	TM1_INT_F LAG	TMO_CAPIN T_FLAG	TMO_INT_F LAG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:6]	-	R	Reserved	-
[5]	TM2_CAPINT_FLAG	WOC	TM2 Capture Mode Interrupt Flag It will be set as 1 when capture end edge is detected. Writing 1 to this bit and it will be cleared.	Read 0: Idle/Busy Read 1: Interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[4]	TM2_INT_FLAG	WOC	TM0 Interrupt Flag It will be set as 1 when TMR overflows. Writing 1 to this bit, and it will be cleared.	Read 0: Idle/Busy Read 1: Interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[3]	TM1_CAPINT_FLAG	WOC	TM1 Capture Mode Interrupt Flag It will be set as 1 when capture end edge is detected. Writing 1 to this bit, and it will be cleared.	Read 0: Idle/Busy Read 1: Interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[2]	TM1_INT_FLAG	WOC	TM0 Interrupt Flag It will be set as 1 when TMR overflows. Writing 1 to this bit, and it will be cleared.	Read 0: Idle/Busy Read 1: Interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[1]	TMO_CAPINT_FLAG	WOC	TMO Capture Mode Interrupt Flag It will be set as 1 when capture end edge is detected. Writing 1 to this bit, and it will be cleared.	Read 0: Idle/Busy Read 1: Interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.

Bit	Function	Type	Description	Condition
[0]	TM0_INT_FLAG	WOC	TM0 Interrupt Flag It will be set as 1 when TMR overflows. Writing 1 to this bit, and it will be cleared.	Read 0: Idle/Busy Read 1: Interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.

12. General-purpose I/O (GPIOs)

12.1. Introduction

The IC can connect with other devices through input port and output port. Four programmable I/O ports are available in GPCM3 series: PortA IOA[31:0], PortB IOB[5:0], PortC IOC[2:0] and PortD IOD[1:0]. All ports are ordinary I/O with programmable wakeup capability, and can do bit operation. In addition to regular I/O function, all ports also provide some special functions in certain pins.

12.2. Feature

- Support 32(IOA[31:0]) + 6(IOB[5:0]) + 3(IOC[2:0]) + 2(IOD[1:0]).
- All IO can be programing input, output, open-drain and pull high/low.
- SPIFC (Serial Peripheral Interface controller for FLASH device access) for IOB[5:0]
- Support Touch IO Function for IOA[31:0].
- Support External Interrupt Function for IOA[31:0].
- Support SAR ADC Line for IOA[31:24].
- Support X32K Crystal PAD for IOA[8:7].
- Support X12M Crystal PAD for IOC[1:0].
- Support IO Wakeup and Bit Operation.
- Support IO Feedback function.

12.3. Function

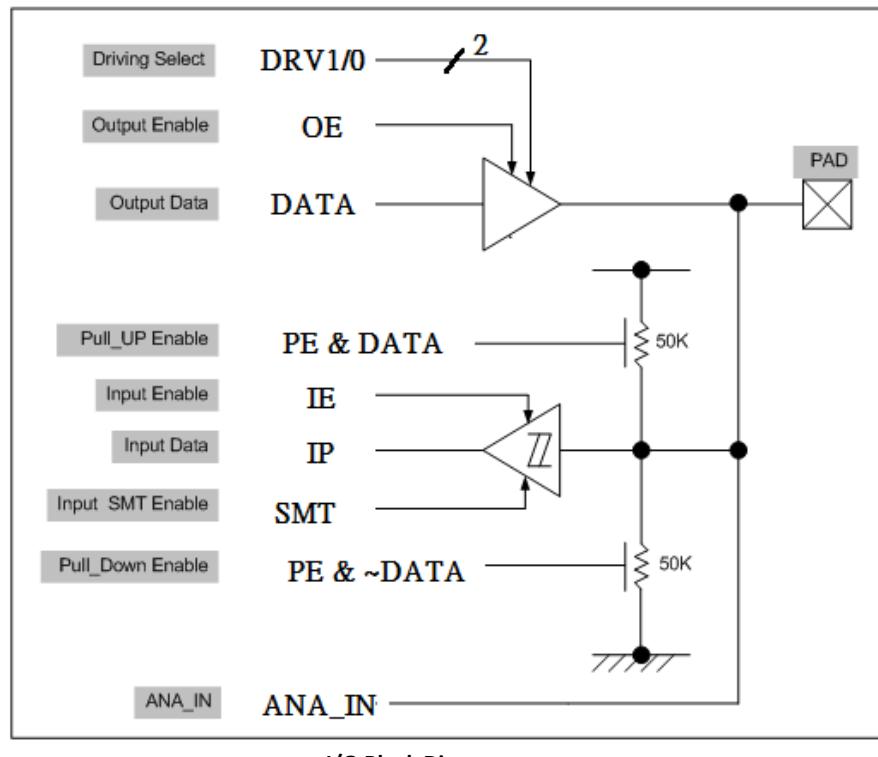
The GPCM3 series provides a bit-to-bit I/O configuration; every I/O bit can be defined individually. Each bit's configuration has four types: input, input floating, open-drain, and output DRV. Each IO pull-high and Schmitt trigger function can be enabled individually. Some IOs are multifunction, which means they can be used in both digital and analog function. The driving and sinking capability on each IO can be set up individually. User can configure IOs driving and sinking capability according to actual condition.

Please refer to the table below for the functional configuration of I/O pins.

CFG[1:0]		OBUF	Function	Wakeup	Description
0	0	0	Pull Low*	Yes**	Input with Pull Low
0	0	1	Pull High	Yes**	Input with Pull High
0	1	x	Float	Yes**	Input with Float
0	1	x	Float	Yes**	Input with Float
1	0	0	Output Open	No	Output Open-Drain Mode
1	0	1	Output Sink Mode	No	Output Open-Drain Mode
1	1	0	Output Low	No	Output Mode
1	1	1	Output High	No	Output Mode

*Default: Input with pull low.

All of IOA/B/C/D pins in the state of 000, 001,010 and 011 have wakeup capability. In Input state, user is able to turn off the wakeup function via the corresponding **GPIOFUNC_WAKEENx.


I/O Block Diagram

Bit Operation

User can set each IO output buffer data respectively. For example, to set GPIOA[6] output buffer data, user needs to read out **GPIOA_OBUF** and rewrite the original value except bit[6] to **GPIOA_OBUF**. In GPCM3 series, it supports IO bit operation. To adjust GPIOA[6] output buffer data setting, user can directly configures **GPIOA6_OBIT**.

KEY Change Detection

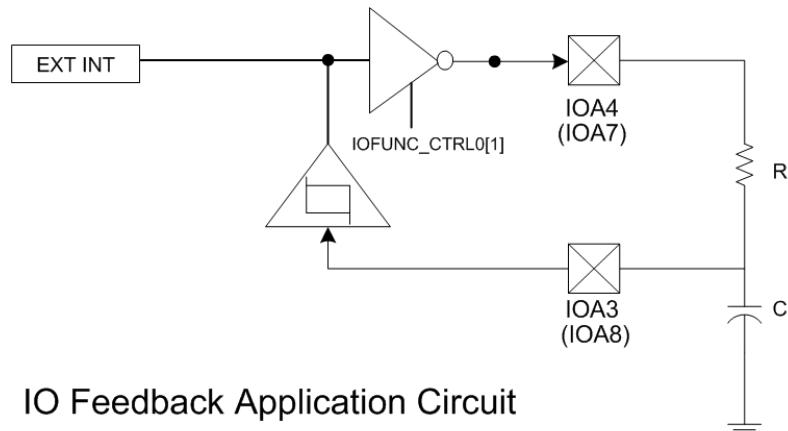
All pads support KEY-Change detection function. It can also be used as Wakeup System source. User can decide which pad enables KEY-Change detection by **GPIOFUNC_WAKEENx**. First, read **GPIOA_IDATA** to know the PAD status (including IOA, IOB, IOC and IOD), and all pad status will be latched. After reading **GPIOA_IDATA** and the data is latched, KEY-Change detection function will be enabled. If IO status changes, it will be shown in **GPIOFUNC_STS[31:0]**. Key change function also supports the trigger of interrupt control.

IR Function

GPIO Function supports IR Function; it supports 7 trigger sources by **IR_CLK_SEL**. 1 will be added the counter value when the IR counter detects a rising pulse of the trigger source. When counter==1 & trigger source = 1, the **IR_OUT** = 1. Else, **IR_OUT** = 0. IR function such as mask, pole and duty switch can be configured in **GPIOFUNC_CTRL1**.

IO Feedback Function

The following circuit shows the configuration of IOA3, IOA4, IOA7, and IOA8 with feedback application. With feedback function, the EXT can obtain an OSC frequency when user adds a RC circuit between IOA4 (IOA7) and IOA3 (IOA8). To make this feedback function work properly, the IOA4 (IOA7) must be configured as “output inverted” and IOA3 (IOA8) as float input mode.



IO Feedback Application Circuit

12.4. Register Function

Registers Map

Name	Address	Description
GPIOA_CFG0	0x50070000	IOA Mode Control Register 0
GPIOA_CFG1	0x50070004	IOA Mode Control Register 1
GPIOA_DRV	0x50070008	IOA Driving Control Register
GPIOA_IE	0x5007000C	IOA Input Enable Control Register
GPIOA_SMT	0x50070010	IOA Schmitt Trigger Control Register
GPIOA_OBUF	0x50070014	IOA Output Data Buffer Register
GPIOA_IDATA	0x50070018	IOA Input Data Status Register
GPIOA_FST	0x5007001C	IOA IO Function First Register
GPIOA0_OBIT	0x50070020	IOA0 Bit Operation Register
GPIOA1_OBIT	0x50070024	IOA1 Bit Operation Register
GPIOA2_OBIT	0x50070028	IOA2 Bit Operation Register
GPIOA3_OBIT	0x5007002C	IOA3 Bit Operation Register
GPIOA4_OBIT	0x50070030	IOA4 Bit Operation Register
GPIOA5_OBIT	0x50070034	IOA5 Bit Operation Register
GPIOA6_OBIT	0x50070038	IOA6 Bit Operation Register
GPIOA7_OBIT	0x5007003C	IOA7 Bit Operation Register
GPIOA8_OBIT	0x50070040	IOA8 Bit Operation Register
GPIOA9_OBIT	0x50070044	IOA9 Bit Operation Register
GPIOA10_OBIT	0x50070048	IOA10 Bit Operation Register
GPIOA11_OBIT	0x5007004C	IOA11 Bit Operation Register
GPIOA12_OBIT	0x50070050	IOA12 Bit Operation Register
GPIOA13_OBIT	0x50070054	IOA13 Bit Operation Register
GPIOA14_OBIT	0x50070058	IOA14 Bit Operation Register
GPIOA15_OBIT	0x5007005C	IOA15 Bit Operation Register
GPIOA16_OBIT	0x50070060	IOA16 Bit Operation Register
GPIOA17_OBIT	0x50070064	IOA17 Bit Operation Register

Name	Address	Description
GPIOA18_OBIT	0x50070068	IOA18 Bit Operation Register
GPIOA19_OBIT	0x5007006C	IOA19 Bit Operation Register
GPIOA20_OBIT	0x50070070	IOA20 Bit Operation Register
GPIOA21_OBIT	0x50070074	IOA21 Bit Operation Register
GPIOA22_OBIT	0x50070078	IOA22 Bit Operation Register
GPIOA23_OBIT	0x5007007C	IOA23 Bit Operation Register
GPIOA24_OBIT	0x50070080	IOA24 Bit Operation Register
GPIOA25_OBIT	0x50070084	IOA25 Bit Operation Register
GPIOA26_OBIT	0x50070088	IOA26 Bit Operation Register
GPIOA27_OBIT	0x5007008C	IOA27 Bit Operation Register
GPIOA28_OBIT	0x50070090	IOA28 Bit Operation Register
GPIOA29_OBIT	0x50070094	IOA29 Bit Operation Register
GPIOA30_OBIT	0x50070098	IOA30 Bit Operation Register
GPIOA31_OBIT	0x5007009C	IOA31 Bit Operation Register
GPIOB_CFG0	0x50070100	IOB Mode Control Register 0
GPIOB_DRV	0x50070108	IOB Driving Control Register
GPIOB_SMT	0x50070110	IOB Schmitt Trigger Control Register
GPIOB_OBUF	0x50070114	IOB Output Data Buffer Register
GPIOB_IDATA	0x50070118	IOB Input Data Status Register
GPIOB_FST	0x5007011C	IOB IO Function First Register
GPIOB0_OBIT	0x50070120	IOB0 Bit Operation Register
GPIOB1_OBIT	0x50070124	IOB1 Bit Operation Register
GPIOB2_OBIT	0x50070128	IOB2 Bit Operation Register
GPIOB3_OBIT	0x5007012C	IOB3 Bit Operation Register
GPIOB4_OBIT	0x50070130	IOB4 Bit Operation Register
GPIOB5_OBIT	0x50070134	IOB5 Bit Operation Register
GPIOC_CFG0	0x50070140	IOC Mode Control Register 0
GPIOC_DRV	0x50070148	IOC Driving Control Register
GPIOC_SMT	0x50070150	IOC Schmitt Trigger Control Register
GPIOC_OBUF	0x50070154	IOC Output Data Buffer Register
GPIOC_IDATA	0x50070158	IOC Input Data Status Register
GPIOC0_OBIT	0x50070160	IOC0 Bit Operation Register
GPIOC1_OBIT	0x50070164	IOC1 Bit Operation Register
GPIOC2_OBIT	0x50070168	IOC2 Bit Operation Register
GPIOD_CFG0	0x500701C0	IOD Mode Control Register 0
GPIOD_DRV	0x500701C8	IOD Driving Control Register
GPIOD_SMT	0x500701D0	IOD Schmitt Trigger Control Register
GPIOD_OBUF	0x500701D4	IOD Output Data Buffer Register
GPIOD_IDATA	0x500701D8	IOD Input Data Status Register
GPIOD0_OBIT	0x500701E0	IOD0 Bit Operation Register

Name	Address	Description
GPIOD1_OBIT	0x500701E4	IOD1 Bit Operation Register
GPIOFUNC_CTRL0	0x50070200	GPIO Function Selection Control Register
GPIOFUNC_CTRL1	0x50070204	GPIO IR Function Control Register
GPIOFUNC_CTRL2	0x50070208	GPIO EXT 0~3 Input Pins Select Control Register
GPIOFUNC_WAKEEN	0x50070210	GPIO Key-Change Wake Up Enable Register 1
GPIOFUNC_WAKEEN2	0x50070214	GPIO Key-Change Wake Up Enable Register 2
GPIOFUNC_STS	0x50070218	GPIO Key-Change Interrupt Status Register
GPIOFUNC_SPIFC	0x5007021C	SPIFC IO Select Register

Registers Function

GPIOA_CFG0								0x50070000	IOA Mode Control Register 0							
Bit	31	30	29	28	27	26	25	24								
Function	IOA15_MODE[1:0]		IOA14_MODE[1:0]		IOA13_MODE[1:0]		IOA12_MODE[1:0]									
Reset Value	0	0	0	0	0	0	0	0								

Bit	23	22	21	20	19	18	17	16
Function	IOA11_MODE[1:0]		IOA10_MODE[1:0]		IOA9_MODE[1:0]		IOA8_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IOA7_MODE[1:0]		IOA6_MODE[1:0]		IOA5_MODE[1:0]		IOA4_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOA3_MODE[1:0]		IOA2_MODE[1:0]		IOA1_MODE[1:0]		IOA0_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:30]	IOA15_MODE[1:0]	R/W	IOA15 Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[29:28]	IOA14_MODE[1:0]	R/W	IOA14 Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[27:26]	IOA13_MODE[1:0]	R/W	IOA13 Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[25:24]	IOA12_MODE[1:0]	R/W	IOA12 Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

Bit	Function	Type	Description	Condition
[23:22]	IOA11_MODE[1:0]	R/W	IOA11 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[21:20]	IOA10_MODE[1:0]	R/W	IOA10 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[19:18]	IOA9_MODE[1:0]	R/W	IOA9 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[17:16]	IOA8_MODE[1:0]	R/W	IOA8 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[15:14]	IOA7_MODE[1:0]	R/W	IOA7 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[13:12]	IOA6_MODE[1:0]	R/W	IOA6 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[11:10]	IOA5_MODE[1:0]	R/W	IOA5 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[9:8]	IOA4_MODE[1:0]	R/W	IOA4 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[7:6]	IOA3_MODE[1:0]	R/W	IOA3 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[5:4]	IOA2_MODE[1:0]	R/W	IOA2 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[3:2]	IOA1_MODE[1:0]	R/W	IOA1 Mode Control	00: Input Mode 01: Input Floating

Bit	Function	Type	Description	Condition
				10: Output Open-Drain 11: Output Mode
[1:0]	IOA0_MODE[1:0]	R/W	IOA0 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

GPIOA_CFG1								
0x50070004								
Bit	31	30	29	28	27	26	25	24
Function	IOA31_MODE[1:0]			IOA30_MODE[1:0]			IOA29_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	IOA27_MODE[1:0]		IOA26_MODE[1:0]		IOA25_MODE[1:0]		IOA24_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	IOA23_MODE[1:0]		IOA22_MODE[1:0]		IOA21_MODE[1:0]		IOA20_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	IOA19_MODE[1:0]		IOA18_MODE[1:0]		IOA17_MODE[1:0]		IOA16_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	IOA31_MODE[1:0]	R/W	IOA31 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[29:28]	IOA30_MODE[1:0]	R/W	IOA30 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[27:26]	IOA29_MODE[1:0]	R/W	IOA29 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[25:24]	IOA28_MODE[1:0]	R/W	IOA28 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[23:22]	IOA27_MODE[1:0]	R/W	IOA27 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

Bit	Function	Type	Description	Condition
[21:20]	IOA26_MODE[1:0]	R/W	IOA26 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[19:18]	IOA25_MODE[1:0]	R/W	IOA25 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[17:16]	IOA24_MODE[1:0]	R/W	IOA24 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[15:14]	IOA23_MODE[1:0]	R/W	IOA23 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[13:12]	IOA22_MODE[1:0]	R/W	IOA22 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[11:10]	IOA21_MODE[1:0]	R/W	IOA21 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[9:8]	IOA20_MODE[1:0]	R/W	IOA20 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[7:6]	IOA19_MODE[1:0]	R/W	IOA19 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[5:4]	IOA18_MODE[1:0]	R/W	IOA18 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[3:2]	IOA17_MODE[1:0]	R/W	IOA17 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[1:0]	IOA16_MODE[1:0]	R/W	IOA16 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

GPIOA_DRV **0x50070008** **IOA Driving Control Register**

Bit	31	30	29	28	27	26	25	24
Function	IOA_DRV[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IOA_DRV[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IOA_DRV[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOA_DRV[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:0]	IOA_DRV[31:0]	R/W	IOA Drive Current Select						0: 8mA 1: 16mA

GPIOA_IE **0x5007000C** **IOA Input Enable Control Register**

Bit	31	30	29	28	27	26	25	24
Function	IOA_IE[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IOA_IE[23:16]							
Reset Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Function	IOA_IE[15:8]							
Reset Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Function	IOA_IE[7:0]							
Reset Value	1	1	1	1	1	1	1	1

Bit	Function	Type	Description						Condition
[31:0]	IOA_IE[31:0]	R/W	IOA Input Enable Control						0: Disabled 1: Enabled

GPIOA_SMT **0x50070010** **IOA Schmitt Trigger Control Register**

Bit	31	30	29	28	27	26	25	24
Function	IOA_SMT[31:24]							
Reset Value	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Function	IOA_SMT[23:16]							
Reset Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Function	IOA_SMT[15:8]							
Reset Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Function	IOA_SMT[7:0]							
Reset Value	1	1	1	1	1	1	1	1

Bit	Function	Type	Description					Condition
[31:0]	IOA_SMT[31:0]	R/W	IOA Schmitt Trigger Enable					0: Disabled 1: Enabled

GPIOA_OBUF IOA Output Data Buffer Register

Bit	31	30	29	28	27	26	25	24
Function	IOA_OBUF[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IOA_OBUF[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IOA_OBUF[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOA_OBUF[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	IOA_OBUF[31:0]	R/W	IOA Output Data Buffer					-

Note: Writing data into **GPIOA_OBUF** will synchronously change the **GPIOA_OBUF** and **GPIOA_IDATA**.

GPIOA_IDATA IOA Input Data Status Register

Bit	31	30	29	28	27	26	25	24
Function	IOA_IDATA[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IOA_IDATA[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IOA_IDATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOA_IDATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	IOA_IDATA[31:0]	R	Read data from the IOA pad. IOA_IDATA is read only.					-

Bit	31	30	29	28	27	26	25	24
Function	IOA_FST[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IOA_FST[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IOA_FST[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOA_FST[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	IOA_FST[31:0]	R/W	IOA IO Function Priority First Enable					0: Disabled 1: Enabled

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:1]	-	-	Reserved						-
[0]	OBIT00	R/W	IOA0 Bit Operation S/W can write this register to change GPIOA_OBUF[0] value.						-

GPIOA1_OBIT
0x50070024
IOA1 Bit Operation Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-		OBIT01		
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-		OBIT02		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:3]	-	-	Reserved						-
[2]	OBIT02	R/W	IOA2 Bit Operation S/W can write this register to change GPIOA_OBUF[2] value.						-
[1:0]	-	-	Reserved						-

GPIOA3_OBIT **0x5007002C** **IOA3 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					OBIT03		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOA4_OBIT **0x50070030** **IOA4 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					OBIT04		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:5]	-	-	Reserved						-
[4]	OBIT04	R/W	IOA4 Bit Operation S/W can write this register to change GPIOA_OBUF[4] value.						-
[3:0]	-	-	Reserved						-

GPIOA5_OBIT **0x50070034** **IOA5 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function		-	OBIT05			-		
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	OBIT06			-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:7]	-	-	Reserved						-
[6]	OBIT06	R/W	IOA6 Bit Operation S/W can write this register to change GPIOA_OBUF[6] value.						-
[5:0]	-	-	Reserved						-

GPIOA7_OBIT **0x5007003C** **IOA7 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	OBIT07				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOA8_OBIT **0x50070040** **IOA8 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:9]	-	R	Reserved						-
[8]	OBIT08	R/W	IOA8 Bit Operation S/W can write this register to change GPIOA_OBUF[8] value.						-
[7:0]	-	R	Reserved						-

GPIOA9_OBIT **0x50070044** **IOA9 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function						OBIT09	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOA10_OBIT **0x50070048** **IOA10 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-	OBIT10	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:11]	-	R	Reserved						-
[10]	OBIT10	R/W	IOA10 Bit Operation S/W can write this register to change GPIOA_OBUF[10] value.						-
[9:0]	-	R	Reserved						-

GPIOA11_OBIT **0x5007004C** **IOA11 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					OBIT11		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOA12_OBIT **0x50070050** **IOA12 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				OBIT12			-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:13]	-	-	Reserved						-
[12]	OBIT12	R/W	IOA12 Bit Operation S/W can write this register to change GPIOA_OBUF[12] value.						-
[11:0]	-	-	Reserved						-

GPIOA13_OBIT **0x50070054** **IOA13 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-		OBIT13			-		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOA14_OBIT **0x50070058** **IOA14 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	OBIT14			-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:15]	-	-	Reserved						-
[14]	OBIT14	R/W	IOA14 Bit Operation S/W can write this register to change GPIOA_OBUF[14] value.						-
[13:0]	-	-	Reserved						-

GPIOA15_OBIT **0x5007005C** **IOA15 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	OBIT15				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOA16_OBIT **0x50070060** **IOA16 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			OBIT16
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:17]	-	-	Reserved						-
[16]	OBIT16	R/W	IOA16 Bit Operation S/W can write this register to change GPIOA_OBUF[16] value.						-
[15:0]	-	-	Reserved						-

GPIOA17_OBIT **0x50070064** **IOA17 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-			OBIT17	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

GPIOA18_OBIT **0x50070068** **IOA18 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-		OBIT18	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:19]	-	-	Reserved						-
[18]	OBIT18	R/W	IOA18 Bit Operation S/W can write this register to change GPIOA_OBUF[18] value.						-
[17:0]	-	-	Reserved						-

GPIOA19_OBIT **0x5007006C** **IOA19 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					OBIT19		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					OBIT20		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:21]	-	-	Reserved						-
[20]	OBIT20	R/W	IOA20 Bit Operation S/W can write this register to change GPIOA_OBUF[20] value.						-
[19:0]	-	-	Reserved						-

GPIOA21_OBIT **0x50070074** **IOA21 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-		OBIT21		-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

GPIOA22_OBIT **0x50070078** **IOA22 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function	-				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	OBIT22			-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:23]	-	-	Reserved						-
[22]	OBIT22	R/W	IOA22 Bit Operation S/W can write this register to change GPIOA_OBUF[22] value.						-
[21:0]	-	-	Reserved						-

GPIOA23_OBIT **0x5007007C** **IOA23 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	OBIT23				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

GPIOA24_OBIT **0x50070080** **IOA24 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function				-				OBIT24
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:25]	-	-	Reserved						-
[24]	OBIT24	R/W	IOA24 Bit Operation S/W can write this register to change GPIOA_OBUF[24] value.						-
[23:0]	-	-	Reserved						-

GPIOA25_OBIT **0x50070084** **IOA25 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function				-			OBIT25	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function				-			OBIT26	-
Reset Value	0	0	0	0	0	0	0	0

GPIOA26_OBIT **0x50070088** **IOA26 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function				-			OBIT26	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:27]	-	-	Reserved					-
[26]	OBIT26	R/W	IOA26 Bit Operation S/W can write this register to change GPIOA_OBUF[26] value.					-
[25:0]	-	-	Reserved					-

GPIOA27_OBIT **0x5007008C** **IOA27 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					OBIT27			-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function	-	-	Reserved					-
Reset Value	0	0	0	0	0	0	0	0

GPIOA28_OBIT **0x50070090** **IOA28 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function		-		OBIT28			-	
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:29]	-	-	Reserved						-
[28]	OBIT28	R/W	IOA28 Bit Operation S/W can write this register to change GPIOA_OBUF[28] value.						-
[27:0]	-	-	Reserved						-

GPIOA29_OBIT **0x50070094** **IOA29 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function	-		OBIT29		-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-		-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-		-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-		-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function	-	OBIT29		-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

GPIOA30_OBIT **0x50070098** **IOA30 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function	-	OBIT30		-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-		-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-		-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-		-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31]	-	-	Reserved						-
[30]	OBIT30	R/W	IOA30 Bit Operation S/W can write this register to change GPIOA_OBUF[30] value.						-
[29:0]	-	-	Reserved						-

GPIOA31_OBIT **0x5007009C** **IOA31 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function	OBIT31				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

GPIOB_CFG0 **0x50070100** **IOB Mode Control Register 0**

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-	IOB5_MODE[1:0]	IOB4_MODE[1:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOB3_MODE[1:0]		IOB2_MODE[1:0]		IOB1_MODE[1:0]		IOB0_MODE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:12]	-	-	Reserved						-

Bit	Function	Type	Description	Condition
[11:10]	IOB5_MODE[1:0]	R/W	IOB5 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[9:8]	IOB4_MODE[1:0]	R/W	IOB4 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[7:6]	IOB3_MODE[1:0]	R/W	IOB3 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[5:4]	IOB2_MODE[1:0]	R/W	IOB2 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[3:2]	IOB1_MODE[1:0]	R/W	IOB1 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[1:0]	IOB0_MODE[1:0]	R/W	IOB0 Mode Control	00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

GPIOB_DRV									0x50070108									IOB Driving Control Register								
Bit	31	30	29	28	27	26	25	24																		
Function	-								-								-									
Reset Value	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Bit	23	22	21	20	19	18	17	16
Function	-	IOB5_DRV[2:0]			-	IOB4_DRV[2:0]		
Reset Value	0	0	0	1	0	0	0	1

Bit	15	14	13	12	11	10	9	8
Function	-	IOB3_DRV[2:0]			-	IOB2_DRV[2:0]		
Reset Value	0	0	0	1	0	0	0	1

Bit	7	6	5	4	3	2	1	0
Function	-	IOB1_DRV[2:0]			-	IOB0_DRV[2:0]		
Reset Value	0	0	0	1	0	0	0	1

Bit	Function	Type	Description					Condition
[31:23]	-	-	Reserved					-

Bit	Function	Type	Description	Condition
[22:20]	IOB5_DRV[2:0]	R/W	IOB5 Drive Current Selection	000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA
[19]	-	-	Reserved	-
[18:16]	IOB4_DRV[2:0]	R/W	IOB4 Drive Current Selection	000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA
[15]	-	-	Reserved	-
[14:12]	IOB3_DRV[2:0]	R/W	IOB3 Drive Current Selection	000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA
[11]	-	-	Reserved	-
[10:8]	IOB2_DRV[2:0]	R/W	IOB2 Drive Current Selection	000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA
[7]	-	-	Reserved	-
[6:4]	IOB1_DRV[2:0]	R/W	IOB1 Drive Current Selection	000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA
[3]	-	-	Reserved	-

Bit	Function	Type	Description					Condition
[2:0]	IOB0_DRV[2:0]	R/W	IOB0 Drive current selection					000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA

GPIOB_SMT **0x50070110** **IOB Schmitt Trigger Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	1	1	1	1	1	1

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							IOB_OBUF[5:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:6]	-	-	Reserved						-
[5:0]	IOB_OBUF[5:0]	R/W	IOB Output Data Buffer Note: Writing data into GPIOB_OBUF will synchronously change the GPIOB_OBUF and GPIOB_IDATA .						-

GPIOB_IDATA								
0x50070118								
Bit	31	30	29	28	27	26	25	24
Function	-				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-			-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-			-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							IOB_IDATA[5:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:6]	-	-	Reserved						-
[5:0]	IOB_IDATA[5:0]	R	Read data from the IOB pad. IOB_IDATA is read only.						-

GPIOB_FST								
0x5007011C								
Bit	31	30	29	28	27	26	25	24
Function	-				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-			-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-			-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							IOB_FST[5:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:6]	-	-	Reserved	-
[5:0]	IOB_FST[5:0]	R/W	IOB IO Function Priority First Enable	0: Disabled 1: Enabled

GPIOB0_OBIT **0x50070120** **IOB0 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-			OBIT00
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function	-	-			-			
Reset Value	0	0	0	0	0	0	0	0

GPIOB1_OBIT **0x50070124** **IOB1 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-			OBIT01	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:2]	-	-	Reserved						-
[1]	OBIT01	R/W	IOB1 Bit Operation S/W writes this register to change GPIOB_OBUF[1] value.						-
[0]	-	-	Reserved						-

GPIOB2_OBIT **0x50070128** **IOB2 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-	OBIT02		-
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-	OBIT03		-
Reset Value	0	0	0	0	0	0	0	0

GPIOB3_OBIT **0x5007012C** **IOB3 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-	OBIT03		-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:4]	-	-	Reserved						-
[3]	OBIT03	R/W	IOB3 Bit Operation S/W writes this register to change GPIOB_OBUF[3] value.						-
[2:0]	-	-	Reserved						-

GPIOB4_OBIT **0x50070130** **IOB4 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					OBIT04			
Reset Value	0	0	0	0	0	0	0	0

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

GPIOB5_OBIT **0x50070134** **IOB5 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					OBIT05			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:6]	-	-	Reserved					-
[5]	OBIT05	R/W	IOB5 Bit Operation S/W writes this register to change GPIOB_OBUF[5] value.					-
[4:0]	-	-	Reserved					-

GPIOC_CFG0									0x50070140	IOC Mode Control Register 0		
Bit	31	30	29	28	27	26	25	24				
Function					-							
Reset Value	0	0	0	0	0	0	0	0				
Bit	23	22	21	20	19	18	17	16				
Function					-							
Reset Value	0	0	0	0	0	0	0	0				
Bit	15	14	13	12	11	10	9	8				
Function					-							
Reset Value	0	0	0	0	0	0	0	0				
Bit	7	6	5	4	3	2	1	0				
Function		-	IOC2_MODE[1:0]		IOC1_MODE[1:0]		IOCO_MODE[1:0]					
Reset Value	0	0	0	0	0	0	0	0				

Bit	Function	Type	Description					Condition
[31:6]	-	-	Reserved					-
[5:4]	IOC2_MODE[1:0]	R/W	IOC2 Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[3:2]	IOC1_MODE[1:0]	R/W	IOC1 Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[1:0]	IOCO_MODE[1:0]	R/W	IOCO Mode Control					00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

GPIOC_DRV									0x50070148	IOC Driving Control Register		
Bit	31	30	29	28	27	26	25	24				
Function					-							
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:3]	-	-	Reserved					-
[2:0]	IOC_DRV[2:0]	R/W	IOC Drive Current Selection					0: 8mA 1: 16mA

GPIOC_SMT **0x50070150** **IOC Schmitt Trigger Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	1	1	1

Bit	Function	Type	Description					Condition
[31:3]	-	-	Reserved					-
[2:0]	IOC_SMT[2:0]	R/W	IOC Schmitt Trigger Enable					0: Disabled 1: Enabled

GPIOC_OBUF **0x50070154** **IOC Output Data Buffer Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-			IOC_OBUF[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:3]	-	-	Reserved						-
[2:0]	IOC_OBUF[2:0]	R/W	IOC Output Data Buffer Note: Writing data into GPIOC_OBUF will synchronously change the GPIOC_OBUF and GPIOC_IDATA .						-

GPIOC_IDATA									0x50070158	IOC Input Data Status Register		
Bit	31	30	29	28	27	26	25	24				
Function				-								
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-			IOC_IDATA[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:3]	-	-	Reserved						-
[2:0]	IOC_IDATA[2:0]	R	Read data from the IOC pad. IOC_IDATA is read only.						-

GPIOC0_OBIT									0x50070160	IOC0 Bit Operation Register		
Bit	31	30	29	28	27	26	25	24				
Function				-								
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	OBIT00
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:1]	-	-	Reserved						-
[0]	OBIT00	R/W	IOCO Bit Operation S/W writes this register to change GPIOC_OBUF[0] value.						-

GPIOC1_OBIT **0x50070164** **IOC1 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	OBIT01
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:2]	-	-	Reserved						-
[1]	OBIT01	R/W	IOC1 Bit Operation S/W writes this register to change GPIOC_OBUF[1] value.						-
[0]	-	-	Reserved						-

GPIOC2_OBIT **0x50070168** **IOC2 Bit Operation Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-		OBIT02		-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:3]	-	-	Reserved						-
[2]	OBIT02	R/W	IOC2 Bit Operation S/W writes this register to change GPIOC_OBUF[2] value.						-
[1:0]	-	-	Reserved						-

GPIOD_CFG0 **0x500701C0** **IOD Mode Control Register 0**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-		IOD1_MODE[1:0]		IOD0_MODE[1:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:4]	-	-	Reserved						-
[3:2]	IOD1_MODE[1:0]	R/W	IOD1 Mode Control						00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode
[1:0]	IOD0_MODE[1:0]	R/W	IOD0 Mode Control						00: Input Mode 01: Input Floating 10: Output Open-Drain 11: Output Mode

GPIOD_DRV **0x500701C8** **IOD Driving Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOD1_DRV[2:0]					IOD0_DRV[2:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition	
[31:6]	-	-	Reserved				-	
[5:3]	IOD1_DRV[2:0]	R/W	IOD1 Drive Current Selection				000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA	
[2:0]	IOD0_DRV[2:0]	R/W	IOD0 Drive Current Selection				000: 2mA 001: 4mA 010: 6mA 011: 8mA 100: 10mA 101: 12mA 110: 14mA 111: 16mA	

GPIOD_SMT **0x500701D0** **IOD Schmitt Trigger Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	IOD_SMT[1:0]
Reset Value	0	0	0	0	0	0	1	1

Bit	Function	Type	Description					Condition
[31:2]	-	-	Reserved					-
[1:0]	IOD_SMT[1:0]	R/W	IOD Schmitt Trigger Enable					0: Disabled 1: Enabled

GPIOD_OBUF **0x500701D4** **IOD Output Data Buffer Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	IOD_OBUF[1:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:2]	-	-	Reserved					-
[1:0]	IOD_OBUF[1:0]	R/W	IOD Output Data Buffer Note: Writing data into GPIOD_OBUF will synchronously change the GPIOD_OBUF and GPIOD_IDATA .					-

GPIOD_IDATA **0x500701D8** **IOD Input Data Status Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	IOD_IDATA[1:0]	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:2]	-	-	Reserved						-
[1:0]	IOD_IDATA[1:0]	R	Read data from the IOD pad. IOD_IDATA is read only.						-

0x500701E0									IOD0 Bit Operation Register
Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	OBIT00
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:1]	-	-	Reserved						-
[0]	OBIT00	R/W	IOD0 Bit Operation S/W writes this register to change GPIOD_OBUF[0] value.						-

0x500701E4									IOD1 Bit Operation Register
Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-						OBIT01	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:2]	-	-	Reserved						-
[1]	OBIOT01	R/W	IOD1 Bit Operation S/W writes this register to change GPIOOD_OBUF[1] value.						-
[0]	-	-	Reserved						-

GPIOFUNC_CTRL0								
0x50070200								
Bit	31	30	29	28	27	26	25	24
Function	-	CCP0_IOSEL	PDM_IOSEL	FB_IOSEL	UART1_TXI_NV	UART0_TXI_NV	I2S_OUT_IOSEL	I2S_IN_IOSEL
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	SPI0_SWAP	SPI0_IOSEL[1:0]		UART1_SW_AP	UART1_IOSEL[1:0]		I2C_IOSEL[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	SPI1_SWAP	SPI1_IOSEL[1:0]		UART0_SW_AP	UART0_IOSEL[1:0]		CCP1_IOSEL	IR_IOSEL
Reset Value	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	SPI1_HWCS_EN	SPI0_HWCS_EN	-	-	FB_EN	ICE_EN
Reset Value	0	0	1	1	0	0	0	1

Bit	Function	Type	Description						Condition
[31]	-	-	Reserved						-
[30]	CCP0_IOSEL	R/W	CCP0 Capture Input Pins Selection						0: IOA[6:3] 1: IOA[16:13]
[29]	PDM_IOSEL	R/W	PDM Pins Selection						0: IOA[6:5] 1: IOA[23:22]
[28]	FB_IOSEL	R/W	Feedback Pins Selection						0: IOA[4:3] 1: IOA[8:7]
[27]	UART1_TXINV	R/W	UART1 TX Invert enable						0: Not inverted. 1: Inverted.
[26]	UART0_TXINV	R/W	UART0 TX Invert Enable						0: Not inverted. 1: Inverted.
[25]	I2S_OUT_IOSEL	R/W	I2S_OUT Pins Selection						0: IOA[12:9] 1:IOA[25:22]
[24]	I2S_IN_IOSEL	R/W	I2S_IN Pins Selection						0: IOA[6:3] 1: IOA[29:26]

Bit	Function	Type	Description	Condition
[23]	SPI0_SWAP	R/W	SPI0 MOSI/MISO Swap	0: Keep 1: Swap
[22:21]	SPI0_IOSEL[1:0]	R/W	SPI0 Pins Selection	00: IOA[6:3] 01: IOA[29:26] 10: IOB[4:1] 11: -
[20]	UART1_SWAP	R/W	UART1 TX/RX Swap	0: Keep 1: Swap
[19:18]	UART1_IOSEL[1:0]	R/W	UART1 Pins Selection	00: IOA[6:5] 01: IOA[21:20] 10: IOA[27:26] 11: -
[17:16]	I2C_IOSEL[1:0]	R/W	I2C Pins Selection	00: IOA[1:0] 01: IOA[16:15] 10: IOA[21:20] 11: -
[15]	SPI1_SWAP	R/W	SPI1 MOSI/MISO Swap	0: Keep 1: Swap
[14:13]	SPI1_IOSEL[1:0]	R/W	SPI1 Pins Selection If SPI1_IOSEL is set as 10 or 11, SPI1's 4 pins will all be turned into MOSI.	00: IOA[12:9] 01: IOA[25:22] 10: IOA[12:9] MOSI 11: IOA[25:22] MOSI
[12]	UART0_SWAP	R/W	UART0 TX/RX Swap Note: It's not able to swap when UART_IO_SEL = 11.	0: Keep 1: Swap
[11:10]	UART0_IOSEL[1:0]	R/W	UART0 Pins Selection	00:IOA[1:0] 01:IOA[14:13] 10:IOA[31:30] 11:IOC[2] & RST (ICE)
[9]	CCP1_IOSEL	R/W	CCP1 Capture Input Pin Selection	0: IOA[12:9] 1: IOA[29:26]
[8]	IR_IOSEL	R/W	IR Pins Selection	0:IOA7 1:IOA17
[7:6]	-	-	Reserved	-
[5]	SPI1_HWCS_EN	R/W	SPI1 CS H/W Control Enable Note: No matter CS pin is controlled by the hardware or the software, CS pin (IOA11 or IOA24) will be set as output mode; it cannot be configured as input mode.	0: By S/W 1: By H/W
[4]	SPI0_HWCS_EN	R/W	SPI0 CS H/W Control Enable Note: No matter CS pin is controlled by the hardware or the software, CS pin (IOA5 or IOA28 or IOB3) will be set as output mode; it cannot be configured as input mode.	0: By S/W 1: By H/W

Bit	Function	Type	Description				Condition	
[3:2]	-	-	Reserved				-	
[1]	FB_EN	R/W	IO Feedback Enable				0: Disabled 1: Enabled	
[0]	ICE_EN	R/W	ICE Enable				0: Disabled 1: Enabled	

GPIOFUNC_CTRL1
0x50070204
GPIO IR Function Control Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IR_POL	IR_MASK	IR_CLK_SEL[2:0]			IR_DUTY[1:0]		IR_TX_EN
Reset Value	0	1	0	0	0	0	0	0

Bit	Function	Type	Description				Condition	
[31:8]	-	-	Reserved				-	
[7]	IR_POL	R/W	IR TX Polarity Control				0: Positive 1: Negative	
[6]	IR_MASK	R/W	IR TX Mask				0: Disabled 1: Enabled	
[5:3]	IR_CLK_SEL[2:0]	R/W	IR TX Clock Source Selection				000: TM0 001: TM1 010: TM2 011: CCP0 100: CCP1 101: CTS0 110: CTS1 111: -	
[2:1]	IR_DUTY[1:0]	R/W	IR TX Duty Selection				00: 1/2 Duty 01: 1/3 Duty 10: 1/4 Duty 11: 1/5 Duty	
[0]	IR_TX_EN	R/W	IR TX Enable				0: Disabled 1: Enabled	

GPIOFUNC_CTRL2
0x50070208
GPIO EXT 0~3 Input Pins Select Control Register

Bit	31	30	29	28	27	26	25	24
Function	EXT3_PINSEL2[2:0]			EXT3_PINSEL1[4:0]				
Reset Value	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Function	EXT2_PINSEL2[2:0]			EXT2_PINSEL1[4:0]				
Reset Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Function	EXT1_PINSEL2[2:0]			EXT1_PINSEL1[4:0]				
Reset Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Function	EXT0_PINSEL2[2:0]			EXT0_PINSEL1[4:0]				
Reset Value	1	1	1	1	1	1	1	1

Bit	Function	Type	Description				Condition
[31:29]	EXT3_PINSEL2[2:0]	R/W	EXTINT3/CCP EXTCLK3 Key Selection				000: IOB0 001: IOB1 010: IOB2 011: IOB3 100: IOB4 101: IOB5 110: Disabled 111: Disabled
[28:24]	EXT3_PINSEL1[4:0]	R/W	EXTINT3/CCP EXTCLK3 Key Selection				00000: IOA0 00001: IOA1 00010: IOA2 00011: IOA3 00100: IOA4 00101: IOA5 00110: IOA6 00111: IOA7 01000: IOA8 01001: IOA9 01010: IOA10 01011: IOA11 01100: IOA12 01101: IOA13 01110: IOA14 01111: IOA15 10000: IOA16 10001: IOA17 10010: IOA18 10011: IOA19 10100: IOA20

Bit	Function	Type	Description	Condition
				10101: IOA21 10110: IOA22 10111: IOA23 11000: IOA24 11001: IOA25 11010: IOA26 11011: IOA27 11100: IOA28 11101: IOA29 11110: IOA30 11111: Disabled
[23:21]	EXT2_PINSEL2[2:0]	R/W	EXTINT2/Timer2 ExtCLK/CCP EXTCLK2 Key Selection	000: IOB0 001: IOB1 010: IOB2 011: IOB3 100: IOB4 101: IOB5 110: Disabled 111: Disabled
[20:16]	EXT2_PINSEL1[4:0]	R/W	EXTINT2/Timer2 ExtCLK/CCP EXTCLK2 Key Selection	00000: IOA0 00001: IOA1 00010: IOA2 00011: IOA3 00100: IOA4 00101: IOA5 00110: IOA6 00111: IOA7 01000: IOA8 01001: IOA9 01010: IOA10 01011: IOA11 01100: IOA12 01101: IOA13 01110: IOA14 01111: IOA15 10000: IOA16 10001: IOA17 10010: IOA18 10011: IOA19 10100: IOA20 10101: IOA21 10110: IOA22 10111: IOA23 11000: IOA24 11001: IOA25 11010: IOA26

Bit	Function	Type	Description	Condition
				11011: IOA27 11100: IOA28 11101: IOA29 11110: IOA30 11111: Disabled
[15:13]	EXT1_PINSEL2[2:0]	R/W	EXTINT1/Timer1 ExtCLK/CCP EXTCLK1 Key Selection	000: IOB0 001: IOB1 010: IOB2 011: IOB3 100: IOB4 101: IOB5 110: Disabled 111: Disabled
[12:8]	EXT1_PINSEL1[4:0]	R/W	EXTINT1/Timer1 ExtCLK/CCP EXTCLK1 Key Selection	00000: IOA0 00001: IOA1 00010: IOA2 00011: IOA3 00100: IOA4 00101: IOA5 00110: IOA6 00111: IOA7 01000: IOA8 01001: IOA9 01010: IOA10 01011: IOA11 01100: IOA12 01101: IOA13 01110: IOA14 01111: IOA15 10000: IOA16 10001: IOA17 10010: IOA18 10011: IOA19 10100: IOA20 10101: IOA21 10110: IOA22 10111: IOA23 11000: IOA24 11001: IOA25 11010: IOA26 11011: IOA27 11100: IOA28 11101: IOA29 11110: IOA30 11111: Disabled

Bit	Function	Type	Description	Condition
[7:5]	EXT0_PINSEL2[2:0]	R/W	EXTINT0/Timer0 ExtCLK/CCP0 EXTCLK Key Selection	000: IOB0 001: IOB1 010: IOB2 011: IOB3 100: IOB4 101: IOB5 110: Disabled 111: Disabled
[4:0]	EXT0_PINSEL1[4:0]	R/W	EXTINT0/Timer0 ExtCLK/CCP0 EXTCLK Key Selection	00000: IOA0 00001: IOA1 00010: IOA2 00011: IOA3 00100: IOA4 00101: IOA5 00110: IOA6 00111: IOA7 01000: IOA8 01001: IOA9 01010: IOA10 01011: IOA11 01100: IOA12 01101: IOA13 01110: IOA14 01111: IOA15 10000: IOA16 10001: IOA17 10010: IOA18 10011: IOA19 10100: IOA20 10101: IOA21 10110: IOA22 10111: IOA23 11000: IOA24 11001: IOA25 11010: IOA26 11011: IOA27 11100: IOA28 11101: IOA29 11110: IOA30 11111: Disabled

GPIOFUNC_WAKEEN								0x50070210	GPIO Key-Change Wake Up Enable Register 1							
Bit	31	30	29	28	27	26	25	24	IOA_EN[31:24]							
Function																
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	IOA_EN[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	IOA_EN[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOA_EN[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	IOA_EN[31:0]	R/W	IOA Key Change Wakeup Function Enable					0: Disabled 1: Enabled

GPIOFUNC_WAKEEN2 0x50070214 GPIO Key-Change Wake Up Enable Register 2

Bit	31	30	29	28	27	26	25	24	
Function	INT_EN	-							
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IOC_EN[1:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31]	INT_EN	R/W	Key Change Interrupt Enable					0: Disabled 1: Enabled
[30:11]	-	R	Reserved					-
[10:9]	IOD_EN[1:0]	R/W	IOD Key Change Wakeup Function Enable					0: Disabled 1: Enabled
[8:6]	IOC_EN[2:0]	R/W	IOC Key Change Wakeup Function Enable					0: Disabled 1: Enabled
[5:0]	IOB_EN[5:0]	R/W	IOB Key Change Wakeup Function Enable					0: Disabled 1: Enabled

GPIOFUNC_STS
0x50070218
GPIO Key-Change Interrupt Status Register

Bit	31	30	29	28	27	26	25	24
Function	INTF			-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31]	INTF	R/W	Key Change Interrupt Flag					Read 0: Not Occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[30:0]	-	-	Reserved					-

GPIOFUNC_SPIFC
0x5007021C
SPIFC IO Select Register

Bit	31	30	29	28	27	26	25	24
Function	-		SPIFC_IOB5_SEL[2:0]		-		SPIFC_IOB4_SEL[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-		SPIFC_IOB3_SEL[2:0]		-		SPIFC_IOB2_SEL[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-		SPIFC_IOB0_SEL[2:0]		-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-			-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31]	-	-	Reserved					-
[30:28]	SPIFC_IOB5_SEL[2:0]	R/W	SPIFC IOB5 Selection					000: WP 001: HOLD 010: MOSI 011: CS

Bit	Function	Type	Description	Condition
				100: MISO 101: WP 110: WP 111: WP
[27]	-	-	Reserved	-
[26:24]	SPIFC_IOB4_SEL[2:0]	R/W	SPIFC IOB4 Selection	000: MISO 001: HOLD 010: MOSI 011: CS 100: MISO 101: WP 110: WP 111: WP
[23]	-	-	Reserved	-
[22:20]	SPIFC_IOB3_SEL[2:0]	R/W	SPIFC IOB3 Selection	000: CS 001: HOLD 010: MOSI 011: CS 100: MISO 101: WP 110: WP 111: WP
[19]	-	-	Reserved	-
[18:16]	SPIFC_IOB2_SEL[2:0]	R/W	SPIFC IOB2 Selection	000: SI 001: HOLD 010: MOSI 011: CS 100: MISO 101: WP 110: WP 111: WP
[15]	-	-	Reserved	-
[14:12]	SPIFC_IOB0_SEL[2:0]	R/W	SPIFC IOB0 Selection	000: HOLD 001: HOLD 010: MOSI 011: CS 100: MISO 101: WP 110: WP 111: WP
[11:0]	-	-	Reserved	-

13. Line-IN ADC (SAR ADC) Controller

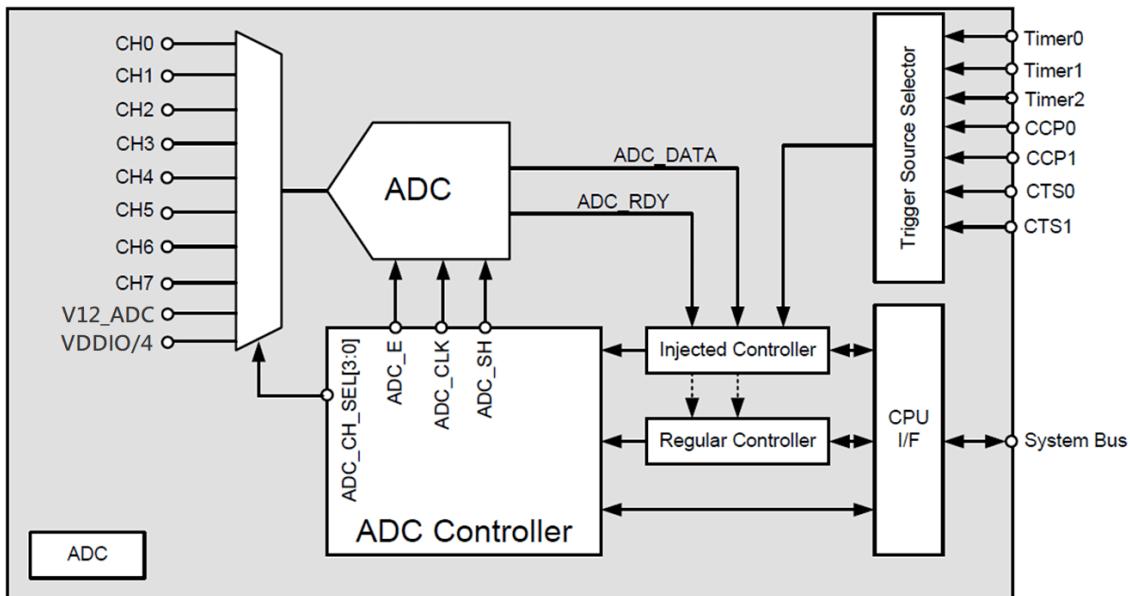
13.1. Introduction

The GPCM3 series has one 12-bit SAR ADC controller. It has up to 10 multiplexed channels allowing it to measure signals from eight external and two internal sources. A/D conversion can be performed in regular, regular with loop mode, and injected mode. The result of the conversion data is stored in a data register. User can set data alignment as left-aligned or right-aligned. Analog watchdog allows the application to detect input-voltage. It is used to detect whether the input voltage is out of the user-defined range. The input clock of SAR ADC is generated from the PCLK clock which is divided by a pre-scaler, and thus it's suggested the highest frequency must not exceed 3MHz. ADC resolution will be affected if the ADC CLK is too high.

13.2. Features

- 12-bit Resolution
- SAR ADC has up to 8 channels in regular mode and 4 channels in injected mode.
- Interrupt generation will happen in three conditions: at the end of a conversion, at end of an Injected conversion, or the occurrence of analog watchdog event.
- Single, Regular, and Regular Scan Conversion Modes
- Scan mode for automatic conversion of channel 0 to channel 'n'.
- Data alignment with left-aligned and right-aligned.
- Channel by Channel Programmable Sampling Time
- SAR ADC Supply Requirement: 2.0V to 3.6V (**Note:** The highest voltage is determined by the value of VDD30SPI_OUT, and the value of VDD30SPI_OUT can be 2.7V~3.6V.)
- SAR ADC Input Range: 0V ≤ VIN ≤ VREF (VDD30SPI_OUT)
- DMA request generation during regular channel conversion.

13.3. Block Diagram



13.4. Function

13.4.1. SAR ADC On-Off Control

The SAR ADC controller can be turned on by configuring the **SAR_ADC_GCTRL.ADC_EN** bit. After **SAR_ADC_GCTRL.ADC_EN** is set as 1, **SAR_ADC_STS.ADC_RDY** will become low, and this indicates SAR ADC is still initializing. Once **SAR_ADC_STS.ADC_RDY** becomes 0, the user can then start using the ADC. In regular mode, the conversion starts when **SAR_ADC_CTRL.SFT_STR** bit is set as 1. In injected mode, the conversion starts when one of the eight trigger sources is triggered. User can stop the conversion and put the SAR ADC in power down mode by disabling the **SAR_ADC_GCTRL.ADC_EN** bit.

13.4.2. Channel Selection

SAR ADC has up to 10 multiplexed channels allowing it to measure signals from eight external and two internal sources. It is possible to operate the conversions in two modes. A mode consists of a sequence of conversions which can be done on any channel and in any order.

- The regular mode is composed of up to 8 conversions. User has to select the regular channels and their order in the conversion sequence in the **SAR_ADC_REG_SEQ.SEQ0TH_SEL** registers. The total number of conversions in the regular mode must be written in the **SAR_ADC_CTRL.REG_CH_NUM**.
- The injected mode is composed of up to 4 conversions. User has to select the injected channels and their order in the conversion sequence in the **SAR_ADC_INJ_SEQ** register. The total number of conversions in the injected mode must be written in the **SAR_ADC_CTRL.INJy_EN** ($y = 0\sim 3$) bits.

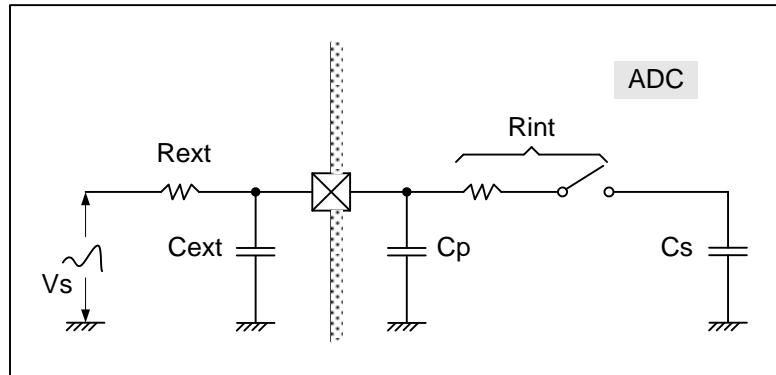
If the **SAR_ADC_REG_SEQ.SEQ0TH_SEL** or **SAR_ADC_CTRL.INJy_EN** ($y = 0\sim 3$) register is modified during a conversion, the current conversion data will be abnormal.

The following table shows the channel mapping of SAR ADC Channel Mapping Table.

ADC	
Channel #	Channel Mapping
0	CH0 (IOA.24)
1	CH1 (IOA.25)
2	CH2 (IOA.26)
3	CH3 (IOA.27)
4	CH4 (IOA.28)
5	CH5 (IOA.29)
6	CH6 (IOA.30)
7	CH7 (IOA.31)
8	VDD12
9	VDDUO/4

13.4.3. Sampling Path

The SAR ADC structure of the GPCM3 series uses a switched capacitor Cs to save the input signal. During signal sampling, the capacitor Cs is connected to the analog input via the multiplexer. The parasitic resistor of the internal switch is called Rint, and the analog voltage source is Vs. The value of Rext, Cext, Rint and Cs determines the required length of the sample cycles. The capacitance value of the switched capacitor Cs is approximately 10pf. The range of parasitic resistance Rint value is $200\Omega \sim 14K\Omega$ and is affected by different operating voltage and temperature. According to different applications, user should consider the equivalent value of resistor Rext and capacitor Cext. The figure below shows a single channel model of ADC.



A Single Channel Model of ADC

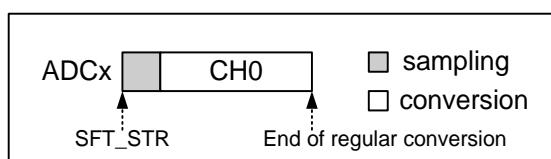
13.4.4. Regulation Conversion Mode

Regular Conversion with Loop Function Disabled

In regular conversion mode, the SAR ADC does one conversion. This mode can only be started when the **SAR_ADC_CTRL.SFT_STR** is set as 1.

Once the selected channel completes the conversion, there are two situations user has to consider.

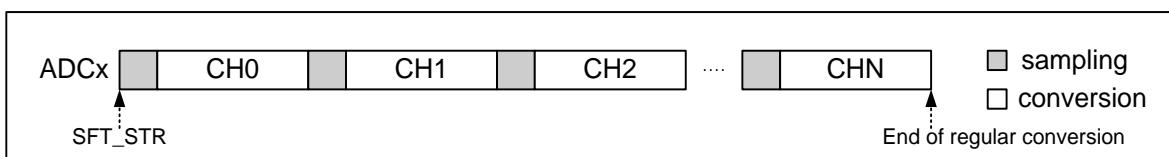
- If a regular channel is converted:
 1. The converted data is stored in the 16-bit **SAR_ADC_REG_DATA** register.
 2. The **SAR_ADC_CTRL.REG_INTF** flag is set as 1.
 3. An interrupt is generated if the **SAR_ADC_CTRL.REG_INTE** is set as 1.
 4. User clears **SAR_ADC_CTRL.REG_INTF** flag by writing 1 to **SAR_ADC_CTRL.REG_INTF**.



A Regular Conversion

- If N regular channels are converted:

1. The converted data is stored in the 16-bit **SAR_ADC_REG_DATA** register.
2. The **SAR_ADC_CTRL.REG_INTF** flag is set as 1.
3. An interrupt is generated if the **SAR_ADC_CTRL.REG_INTE** is set as 1.
4. User clears **SAR_ADC_CTRL.REG_INTF** flag by writing 1 to **SAR_ADC_CTRL.REG_INTF**.
5. When an ADC channel finishes converting, there will be an interval of the **SAR_ADC_CLK** cycles (set in **SAR_ADC_CTRL.REG_SEQ_GAP**). After the interval ends, the next conversion will be executed until all the regular channels finish.



N Regular Conversion

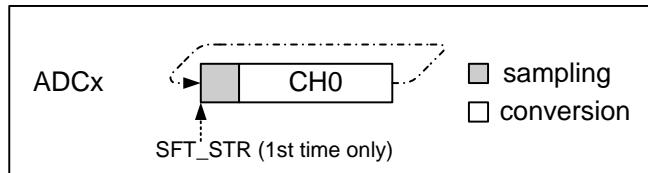
Regular Conversion with Loop Function Enabled

In regular conversion with loop enable mode, the SAR ADC repeats the channel conversion action. This mode is started by setting the **SAR_ADC_CTRL.SFT_STR** as 1. Before starting the SAR ADC conversion, user must set **SAR_ADC_CTRL.LOOP_EN** as 1. The data conversion will be triggered by the hardware automatically until the software disables SAR ADC or sets **SAR_ADC_CTRL.LOOP_EN** as 0.

Once the selected channel completes the conversion, there are two situations user has to consider.

- If only one regular channel is selected:

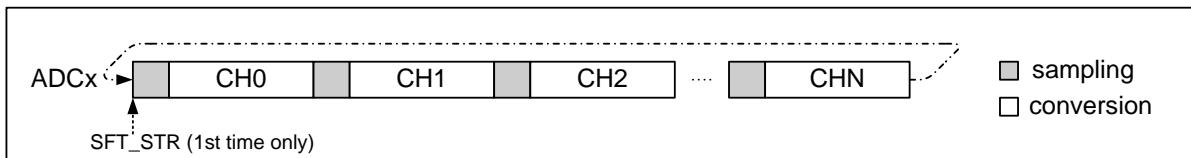
1. The converted data is stored in the 16-bit **SAR_ADC_REG_DATA** register.
2. The **SAR_ADC_CTRL.REG_INTF** flag is set as 1.
3. An interrupt is generated if the **SAR_ADC_CTRL.REG_INTE** is set as 1.
4. User clears **SAR_ADC_CTRL.REG_INTF** flag by writing 1 to **SAR_ADC_CTRL.REG_INTF**.
5. When an ADC channel finishes converting, there will be an interval of the **SAR_ADC_CLK** cycles (set in **SAR_ADC_CTRL.REG_SEQ_GAP**). After the interval ends, the next iteration will be executed until the software disables SAR ADC or sets **SAR_ADC_CTRL.LOOP_EN** as 0.



A Regular Conversion with Loop Function Enabled

- If N regular channels are converted:

1. The converted data is stored in the 16-bit **SAR_ADC_REG_DATA** register.
2. The **SAR_ADC_CTRL.REG_INTF** flag is set as 1.
3. An interrupt is generated if the **SAR_ADC_CTRL.REG_INTE** is set as 1.
4. User clears **SAR_ADC_CTRL.REG_INTF** flag by writing 1 to **SAR_ADC_CTRL.REG_INTF**.
5. When an ADC channel finishes converting, there will be an interval of the **SAR_ADC_CLK** cycles (set in **SAR_ADC_CTRL.REG_SEQ_GAP**). After the interval ends, the next conversion will be executed until all of the regular channel finish.
6. When an ADC channel finishes converting, there will be an interval of the **SAR_ADC_CLK** cycles (set in **SAR_ADC_CTRL.REG_SEQ_GAP**). After the interval ends, the next iteration will be executed until the software disables SAR ADC or sets **SAR_ADC_CTRL.LOOP_EN** as 0.

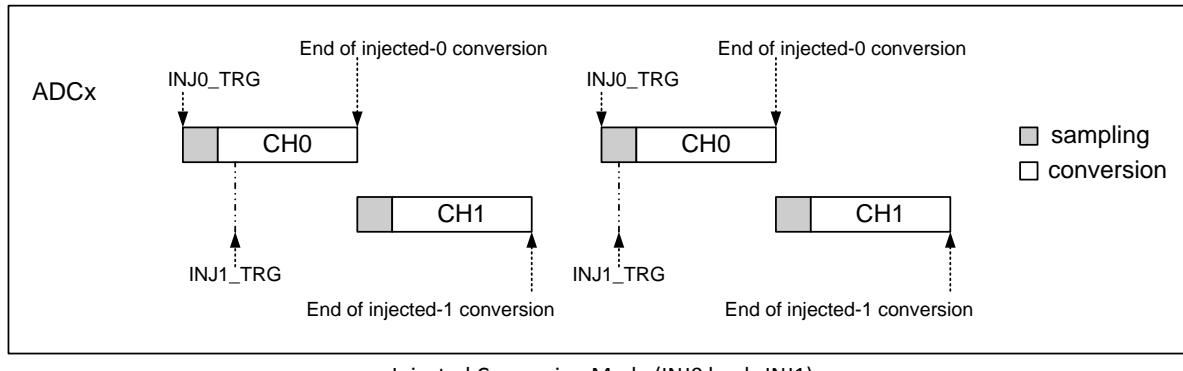


N Regular Conversion with Loop Function Enabled

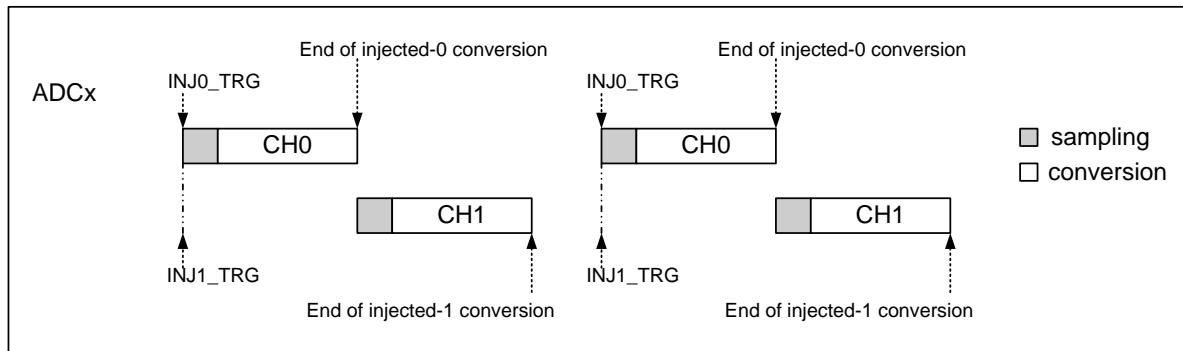
13.4.5. Injected Conversion Mode

To use triggered injection, one of **SAR_ADC_CTRL.INJy_EN** ($y = 0\sim 3$) must be set as 1. Each injected channel supports 8 trigger sources. User can select a trigger source by setting **SAR_ADC_CTRL.INJy_TRG_SEL** ($y = 0\sim 3$) as 1. The priority of the injected channels is INJ0 > INJ1 > INJ2 > INJ3. In addition, the priority of any injection channel is always higher than the regular channel. The figures below explain how injected mode works.

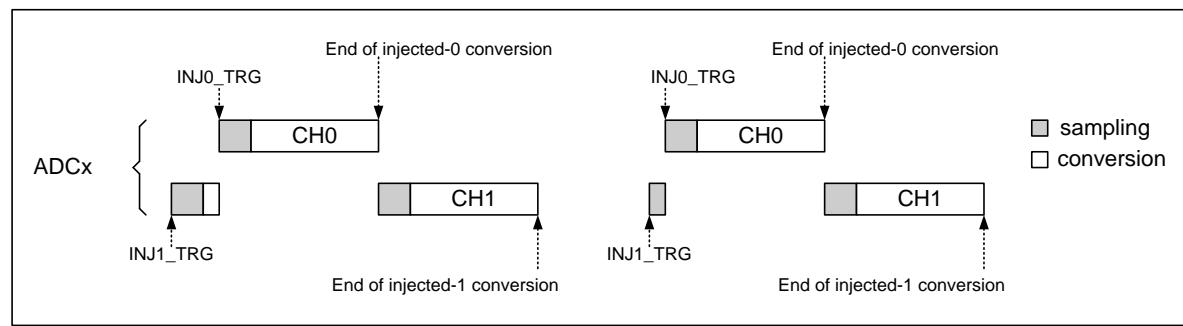
1. Start the conversion of an injected channel by a trigger source.
2. The converted data is stored in the 16-bit **SAR_ADC_INJy_DATA** register.
3. The **SAR_ADC_CTRL.INJy_INTF** flag is set as 1.
4. User clears **SAR_ADC_CTRL.INJy_INTF** flag by writing 1 to **SAR_ADC_CTRL.INJy_INTF**.



Injected Conversion Mode (INJ0 leads INJ1)



Injected Conversion Mode (INJ0 and INJ1 at the same time)



Injected Conversion Mode (INJ1 leads INJ0)

13.4.6. Combined Regular/Injected Simultaneous Mode

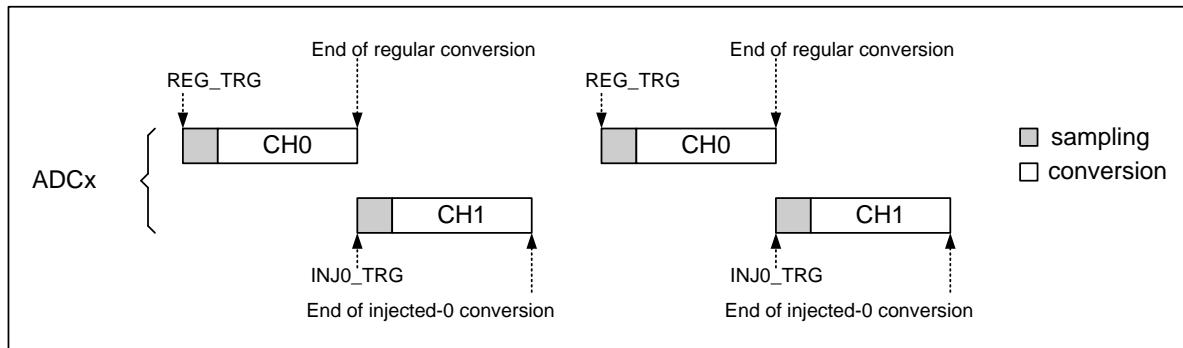
In combined regular and injected simultaneous mode, it is possible to interrupt regular channel simultaneous conversion and start injected trigger conversion of an injected channel. The figures below show the behaviors of an injected channel trigger interrupts a regular simultaneous conversion. The injected channel conversion will start immediately after the injected event triggers. If the regular conversion is already running, it will be suspended, and the regular channel will resume synchronously at the end of the injected conversion. The figures below show how regular and injected mode work.

- If the regular channel is converted first:

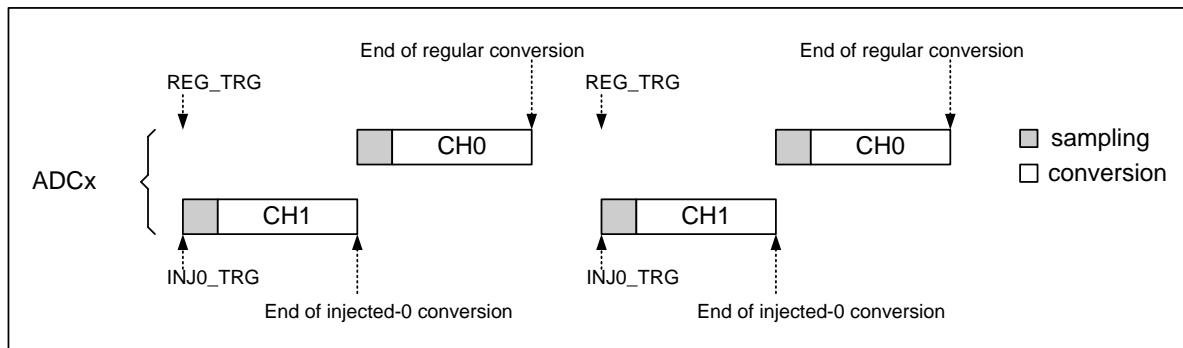
1. Regular channel is running.
2. If an injected trigger occurs during the regular channel conversion, the regular conversion will be suspended, and the injected channel will be converted.
3. The regular channel conversion will resume at the end of the injected conversion.

- If the injected channel is converted first:

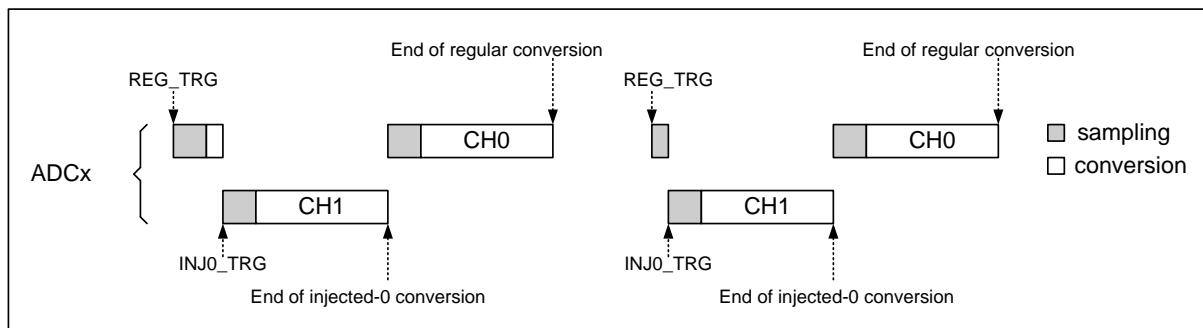
1. Injected channel is running.
2. If a regular trigger occurs during the injected channel conversion, the regular conversion will start at the end of the injected conversion.



Combined Regular/Injected Conversion Mode (regular leads INJ0)



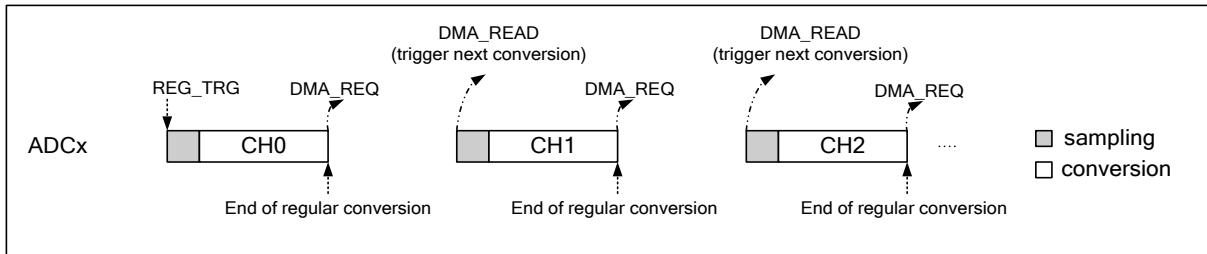
Combined Regular/Injected Conversion Mode (regular and INJ0 arrive at the same time)



Combined Regular/Injected Conversion Mode (INJ0 interrupts regular)

13.4.7. Regular Conversion with DMA

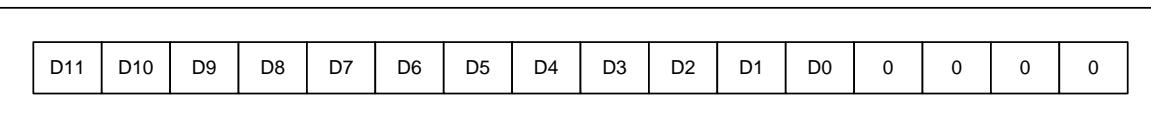
In regular mode, since the converted data are stored in **SAR_ADC_REG_DATA**, user can read data out by CPU access. In addition, DMA access is another access path. The DMA access is applied for conversion of more than one regular channel. This prevents the data stored in **SAR_ADC_REG_DATA** from being lost or overwritten. Please note that the DMA function does not support injected mode. The figure below shows an example of the regular conversion with DMA function.



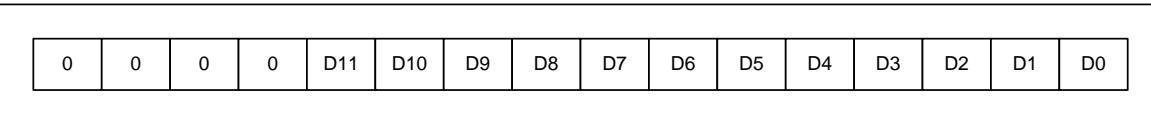
The Regular Conversion with DMA Function

13.4.8. Data Alignment

In GPCM3 series, SAR ADC provides two data alignment types, which means user can set data alignment as left-aligned or right-aligned. **SAR_ADC_CTRL.DAT_ALIGN** bit determines the alignment type of the data which will be stored after conversion. The following figures show some examples of the different data alignments.



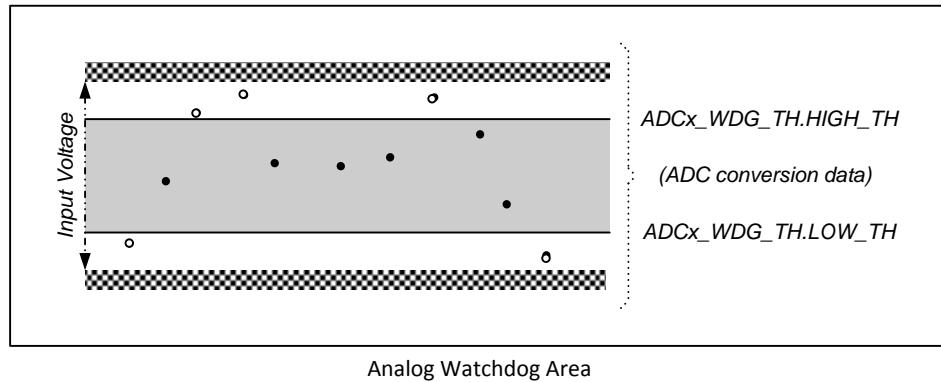
Left Alignment of SAR ADC Data (**SAR_ADC_CTRL.DAT_ALIGN = 0**)

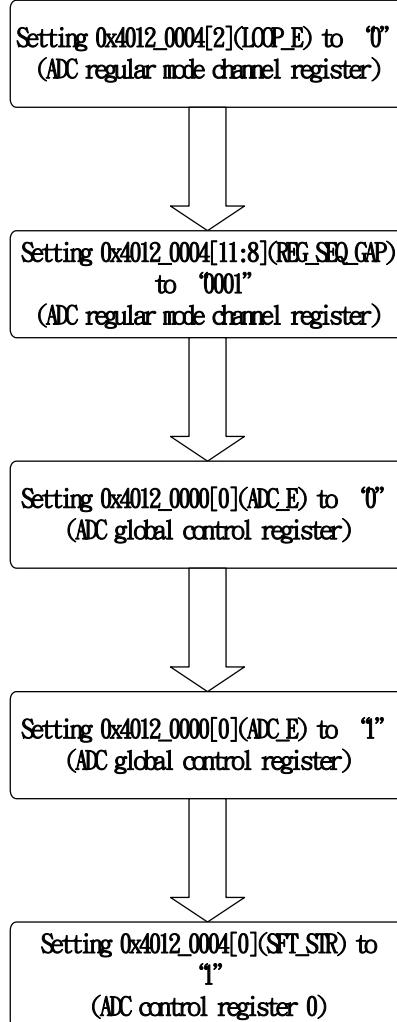


Right Alignment of SAR ADC Data (**SAR_ADC_CTRL.DAT_ALIGN = 1**)

13.4.9. Analog Watchdog

In GPCM3 series, SAR ADC provides a mechanism to measure voltage range of input signal. The SAR_ADC_CTRL.A_WDGR_INTF/SAR_ADC_CTRL.A_WDGJ_INTF will be set as 1 if the analog voltage converted by the SAR ADC is below SAR_ADC_WDG_TH.LOW_THRESHOLD or is above SAR_ADC_WDG_TH.HIGH_THRESHOLD. Two interrupts can be enabled by using the SAR_ADC_CTRL.A_WDGJ_INTE and SAR_ADC_CTRL.A_WDGR_INTE. In addition, the threshold setting is independent of the SAR_ADC_WDG_THRESHOLD setting. The comparison will be done before the data alignment.



13.4.10. SAR ADC Program Example
SAR ADC Change Mode From Loop Mode to Single Mode

13.5. Register Description
Register Map

Name	Address	Description
SAR_ADC_GCTRL	0x40120000	SAR ADC Global Control Register
SAR_ADC_CTRL	0x40120004	SAR ADC Control Register
SAR_ADC_STS	0x40120008	SAR ADC Status Register
SAR_ADC_SMP0	0x4012000C	SAR ADC Sample Time Control Register 0
SAR_ADC_SMP1	0x40120010	SAR ADC Sample Time Control Register 1
SAR_ADC_REG_COV	0x40120014	SAR ADC Regular Mode Conversion Channel Register
SAR_ADC_REG_SEQ	0x40120018	SAR ADC Regular Sequence Register
SAR_ADC_INJ_SEQ	0x4012001C	SAR ADC Injected Sequence Register
SAR_ADC_WDG_TH	0x40120020	SAR ADC Watch-Dog Threshold Register
SAR_ADC_REG_DATA	0x40120024	SAR ADC Regular Data Register
SAR_ADC_INJ0_DATA	0x40120028	SAR ADC Injected0 Data Register

Name	Address	Description
SAR_ADC_INJ1_DATA	0x4012002C	SAR ADC Injected1 Data Register
SAR_ADC_INJ2_DATA	0x40120030	SAR ADC Injected2 Data Register
SAR_ADC_INJ3_DATA	0x40120034	SAR ADC Injected3 Data Register

Registers Function
SAR_ADC_GCTRL **0x40120000** **SAR ADC Global Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-	ANALOG_W DGJ_INT_E N	ANALOG_W DGR_INT_E N	INJ3_INT_E N	INJ2_INT_E N	INJ1_INT_E N	INJ0_INT_E N	REG_MODE _INT_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	ADC_CLK_SEL[4:0]
Reset Value	0	0	0	0	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	DMA_EN	-	DAT_ALIGN	ADC_FMT_SEL	ADC_EN
Reset Value	0	0	0	0	0	0	1	0

Bit	Function	Type	Description	Condition
[31]	-	-	Reserved	-
[30]	ANALOG_WDGJ_INT_EN	R/W	Analog Watch-Dog Interrupt Enable on Injected Channel	0: Disabled 1: Enabled
[29]	ANALOG_WDGR_INT_EN	R/W	Analog Watch-Dog Interrupt Enable on Regular Channel	0: Disabled 1: Enabled
[28]	INJ3_INT_EN	R/W	Injected Conversion Interrupt Enable of Channel 3	0: Disabled 1: Enabled
[27]	INJ2_INT_EN	R/W	Injected Conversion Interrupt Enable of Channel 2	0: Disabled 1: Enabled
[26]	INJ1_INT_EN	R/W	Injected Conversion Interrupt Enable of Channel 1	0: Disabled 1: Enabled
[25]	INJ0_INT_EN	R/W	Injected Conversion Interrupt Enable of Channel 0	0: Disabled 1: Enabled
[24]	REG_MODE_INT_EN	R/W	Regular Mode Interrupt Enable	0: Disabled 1: Enabled
[23:13]	-	-	Reserved	-
[12:8]	ADC_CLK_SEL[4:0]	R/W	ADC Clock Selection Bits CLK = PCLK/((ADC_CLK_SEL + 1)*2)	-
[7:5]	-	-	Reserved	-

Bit	Function	Type	Description	Condition
[4]	DMA_EN	R/W	ADC DMA Request Enable	0: Disabled 1: Enabled
[3]	-	-	Reserved	-
[2]	DAT_ALIGN	R/W	SAR ADC Output Data Alignment	0: Left Aligned 1: Right Aligned
[1]	ADC_FMT_SEL	R/W	ADC Data Signed/Unsigned Selection	0: Unsigned 1: Signed
[0]	ADC_EN	R/W	ADC Analog Block Enable	0: Disabled 1: Enabled

SAR_ADC_CTRL **0x40120004** **SAR ADC Control Register**

Bit	31	30	29	28	27	26	25	24		
Function	INJ3_TRG_SEL[2:0]				INJ3_EN	INJ2_TRG_SEL[2:0]				INJ2_EN
Reset Value	0	0	0	0	0	0	0	0		

Bit	23	22	21	20	19	18	17	16		
Function	INJ1_TRG_SEL[2:0]				INJ1_EN	INJ0_TRG_SEL[2:0]				INJ0_EN
Reset Value	0	0	0	0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8	
Function	ANALOG_W	-				REG_SEQ_GAP_SEL[3:0]			
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Function	REG_CH_NUM[3:0]				ANALOG_W	LOOP_EN	REG_EN	SFT_STR
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:29]	INJ3_TRG_SEL[2:0]	R/W	Injection Channel 3 Trigger Source Selection Bits	000: TM0 001: TM1 010: TM2 011: CCP0 100: CCP1 101: CTS0 110: CTS1 111: -
[28]	INJ3_EN	R/W	Injection Channel 3 Enable Bit	0: Disabled 1: Enabled
[27:25]	INJ2_TRG_SEL[2:0]	R/W	Injection Channel 2 Trigger Source Selection Bits	000: TM0 001: TM1 010: TM2 011: CCP0

Bit	Function	Type	Description	Condition
				100: CCP1 101: CTS0 110: CTS1 111: -
[24]	INJ2_EN	R/W	Injection Channel 2 Enable Bit	0: Disabled 1: Enabled
[23:21]	INJ1_TRG_SEL[2:0]	R/W	Injection Channel 1 Trigger Source Selection Bits	000: TM0 001: TM1 010: TM2 011: CCP0 100: CCP1 101: CTS0 110: CTS1 111: -
[20]	INJ1_EN	R/W	Injection Channel 1 Enable Bit	0: Disabled 1: Enabled
[19:17]	INJ0_TRG_SEL[2:0]	R/W	Injection Channel 0 Trigger Source Selection Bits	000: TM0 001: TM1 010: TM2 011: CCP0 100: CCP1 101: CTS0 110: CTS1 111: -
[16]	INJ0_EN	R/W	Injection Channel 0 Enable Bit	0: Disabled 1: Enabled
[15]	ANALOG_WDGJ_EN	-	Analog Watch-Dog Enable on Injected Channel	0: Disabled 1: Enabled
[14:12]	-	-	Reserved	-
[11:8]	REG_SEQ_GAP_SEL[3:0]	R/W	Regular Sequence Gap Selection Bits	0000: 0 ADC Clock 0001: 1 ADC Clock 0010: 2 ADC Clock 0011: 3 ADC Clock 0100: 4 ADC Clock 0101: 5 ADC Clock 0110: 6 ADC Clock 0111: 7 ADC Clock 1000: TM0 1001: TM1 1010: TM2 1011: CCP0 1100: CCP1

Bit	Function	Type	Description	Condition
				1101: CTS0 1110: CTS1 1111: Manual
[7:4]	REG_CH_NUM[3:0]	R/W	Channel Number During Regular Sequence Channel No. = REG_CH_NUM + 1	-
[3]	ANALOG_WDGR_EN	R/W	Analog Watch-Dog Enable on Regular Channel	0: Disabled 1: Enabled
[2]	LOOP_EN	R/W	ADC Loop Scan Enable Note: This bit is available in regular mode only.	0: Single Conversion 1: Loop Conversion
[1]	REG_EN	R/W	Regular Mode Enable Bit	0: Disabled 1: Enabled
[0]	SFT_STR	R/W	Software Start Bit Note: This bit is available in regular mode only, and it will be cleared by the hardware in single conversion mode.	0: Disabled 1: Enabled

0x40120008								SAR ADC Status Register	
Bit	31	30	29	28	27	26	25	24	
Function	-	ANALOG_WDGJ_INTF	ANALOG_WDGR_INTF	INJ3_INTF	INJ2_INTF	INJ1_INTF	INJ0_INTF	REG_MODE_INTF	
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				ADC_RDY
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31]	-	-	Reserved	-
[30]	ANALOG_WDGJ_INTF	R/W	Analog watch-dog interrupt flag on injected channel.	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[29]	ANALOG_WDGR_INTF	R/W	Analog watch-dog interrupt flag on regular channel.	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.

Bit	Function	Type	Description	Condition
[28]	INJ3_INTF	R/W	Injected Conversion Interrupt Flag of Channel 3	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[27]	INJ2_INTF	R/W	Injected Conversion Interrupt Flag of Channel 2	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[26]	INJ1_INTF	R/W	Injected Conversion Interrupt Flag of Channel 1	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[25]	INJ0_INTF	R/W	Injected Conversion Interrupt Flag of Channel 0	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[24]	REG_MODE_INTF	R/W	Regular Conversion Interrupt Flag	Read 0: Not occurs. Read 1: Occurs. Write 0: No effect. Write 1: Clear the flag.
[23:1]	-	-	Reserved	-
[0]	ADC_RDY	R	ADC Ready Flag	0: Not completed. 1: Completed.

SAR_ADC_SMP0								0x4012000C SAR ADC Sample Time Control Register 0													
Bit	31	30	29	28	27	26	25	24	Bit	31	30	29	28	27	26	25					
Function	-	CH7_SMP_SEL[2:0]				-	CH6_SMP_SEL[2:0]				Function	-	CH5_SMP_SEL[2:0]				-	CH4_SMP_SEL[2:0]			
Reset Value	0	0	0	0	0	0	0	0	Reset Value	0	0	0	0	0	0	0					

Bit	23	22	21	20	19	18	17	16
Function	-	CH5_SMP_SEL[2:0]				-	CH4_SMP_SEL[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	CH3_SMP_SEL[2:0]				-	CH2_SMP_SEL[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	CH1_SMP_SEL[2:0]				-	CH0_SMP_SEL[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31]	-	-	Reserved					-
[30:28]	CH7_SMP_SEL[2:0]	R/W	Channel 7 Sample Time Selection					000: 1 ADC Clock 001: 2 ADC Clock

Bit	Function	Type	Description	Condition
				010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock
[27]	-	-	Reserved	-
[26:24]	CH6_SMP_SEL[2:0]	R/W	Channel 6 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock
[23]	-	-	Reserved	-
[22:20]	CH5_SMP_SEL[2:0]	R/W	Channel 5 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock
[19]	-	-	Reserved	-
[18:16]	CH4_SMP_SEL[2:0]	R/W	Channel 4 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock
[15]	-	-	Reserved	-
[14:12]	CH3_SMP_SEL[2:0]	R/W	Channel 3 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock



GPCM3 SERIES PROGRAMMING GUIDE

Bit	Function	Type	Description	Condition
[11]	-	-	Reserved	-
[10:8]	CH2_SMP_SEL[2:0]	R/W	Channel 2 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock
[7]	-	-	Reserved	-
[6:4]	CH1_SMP_SEL[2:0]	R/W	Channel 1 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock
[3]	-	-	Reserved	-
[2:0]	CH0_SMP_SEL[2:0]	R/W	Channel 0 Sample Time Selection	000: 1 ADC Clock 001: 2 ADC Clock 010: 4 ADC Clock 011: 8 ADC Clock 100: 16 ADC Clock 101: 32 ADC Clock 110: 48 ADC Clock 111: 64 ADC Clock

SAR_ADC_SMP1 **0x40120010** **SAR ADC Sample Time Control Register 1**

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:7]	-	-	Reserved	-
[6:4]	CH9_SMP_SEL[2:0]	R/W	Channel 9 sample time selection	000: 1 ADC clock 001: 2 ADC clock 010: 4 ADC clock 011: 8 ADC clock 100: 16 ADC clock 101: 32 ADC clock 110: 48 ADC clock 111: 64 ADC clock
[3]	-	-	Reserved	-
[2:0]	CH8_SMP_SEL[2:0]	R/W	Channel 8 sample time selection	000: 1 ADC clock 001: 2 ADC clock 010: 4 ADC clock 011: 8 ADC clock 100: 16 ADC clock 101: 32 ADC clock 110: 48 ADC clock 111: 64 ADC clock

SAR_ADC_REG_COV
0x40120014
SAR ADC Regular Mode Conversion Channel
Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-				REG_CHA[3:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:4]	-	-	Reserved	-
[3:0]	REG_CHA[3:0]	R	Regular Mode Conversion Channel It's the channel of currently conversion, and it will be updated after the conversion is done.	-

SAR_ADC_REG_SEQ
0x40120018
SAR ADC Regular Sequence Register

Bit	31	30	29	28	27	26	25	24
Function	SEQ7TH_SEL[3:0]					SEQ6TH_SEL[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	SEQ5TH_SEL[3:0]					SEQ4TH_SEL[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	SEQ3RD_SEL[3:0]					SEQ2ND_SEL[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SEQ1ST_SEL[3:0]					SEQ0TH_SEL[3:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:28]	SEQ7TH_SEL[3:0]	R/W	Regular Sequence 7 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[27:24]	SEQ6TH_SEL[3:0]	R/W	Regular Sequence 6 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[23:20]	SEQ5TH_SEL[3:0]	R/W	Regular Sequence 5 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[19:16]	SEQ4TH_SEL[3:0]	R/W	Regular Sequence 4 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[15:12]	SEQ3RD_SEL[3:0]	R/W	Regular Sequence 3 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[11:8]	SEQ2ND_SEL[3:0]	R/W	Regular Sequence 2 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[7:4]	SEQ1ST_SEL[3:0]	R/W	Regular Sequence 1 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-
[3:0]	SEQ0TH_SEL[3:0]	R/W	Regular Sequence 0 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.	-

SAR_ADC_INJ_SEQ
0x4012001C
SAR ADC Injected Sequence Register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	SEQ3RD_SEL[3:0]	-	-	-	SEQ2ND_SEL[3:0]	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SEQ1ST_SEL[3:0]	-	-	-	SEQ0TH_SEL[3:0]	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	-	Reserved						-
[15:12]	SEQ3RD_SEL[3:0]	R/W	Injected Sequence 3 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.						-
[11:8]	SEQ2ND_SEL[3:0]	R/W	Injected Sequence 2 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.						-
[7:4]	SEQ1ST_SEL[3:0]	R/W	Injected Sequence 1 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.						-
[3:0]	SEQ0TH_SEL[3:0]	R/W	Injected Sequence 0 Selection Note: For the channel mapping of ADC, please refer to SAR ADC channel mapping table.						-

SAR_ADC_WDG_TH
0x40120020
SAR ADC Watch-Dog Threshold Register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	HIGH_THRESHOLD[11:4]	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	LOW_THRESHOLD[3:0]				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:20]	HIGH_THRESHOLD [11:0]	R/W	Analog Watch-Dog High Threshold						-
[19:16]	-	-	Reserved						-
[15:4]	LOW_THRESHOLD [11:0]	R/W	Analog Watch-Dog Low Threshold						-
[3:0]	-	-	Reserved						-

SAR_ADC_REG_DATA
0x40120024
SAR ADC Regular Data Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	REG_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	REG_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	-	Reserved						-
[15:0]	REG_DATA[15:0]	R/W	SAR ADC Regular Data/Manual Trigger Source When ADC is in manual mode (REG_SEQ_GAP = 1111), the S/W must write 1 to trigger the next conversion.						-

SAR_ADC_INJ0_DATA
0x40120028
SAR ADC Injected0 Data Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	INJ0_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	INJ0_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	-	Reserved					-
[15:0]	INJ0_DATA[15:0]	R/W	Injected Channel 0 Data					-

SAR_ADC_INJ1_DATA	0x4012002C	SAR ADC Injected1 Data Register						
Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	INJ1_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	INJ1_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	-	Reserved					-
[15:0]	INJ1_DATA[15:0]	R/W	Injected Channel 1 Data					-

SAR_ADC_INJ2_DATA	0x40120030	SAR ADC Injected2 Data Register						
Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	INJ2_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	INJ2_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	-	Reserved					-
[15:0]	INJ2_DATA[15:0]	R/W	Injected Channel 2 Data					-

SAR_ADC_INJ3_DATA								
0x40120034								
Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					INJ3_DATA[15:8]			
Reset Value	0	0	0	0	0	0	0	0

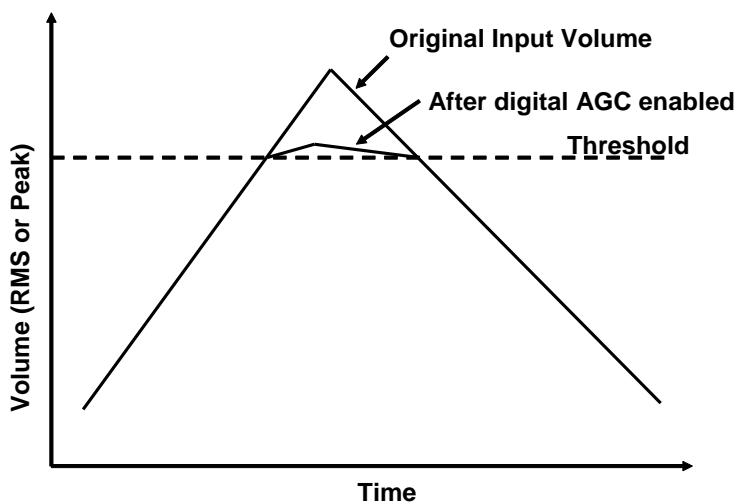
Bit	7	6	5	4	3	2	1	0
Function					INJ3_DATA[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	-	Reserved					-
[15:0]	INJ3_DATA[15:0]	R/W	Injected Channel 3 Data					-

14. Audio Delta-Sigma ADC

14.1. Introduction

A 16-bit resolution SDM ADC (Delta-Sigma Modulation ADC) has DAGC (Digital Audio Gain Control) and auto mute function which are embedded in GPCM3 series. When the peak or RMS of the record sound is larger than the value, {threshold, 0x80}, the digital AGC will be activated and then low the PGA gain. The definition of attack time is how fast the digital AGC responses to the sound data peak or to RMS exceeding the threshold. The fast attack time will result in fast response to the sound data, meaning the compression will start in a short time. It's mainly used to prevent saturation but the transient part in the wave may be lost. So, fast attack time is suitable for melody but not for drum. The slow attack time will result in slow response to the sound data, meaning the compression will start after a while. It can preserve the transient part in the wave but it's not suitable for preventing saturation. User can try both settings and find the best combinations for the application. If AGC_CTRL1[7:0] = 1 and Scale = 00, Attack time = 0.125ms. If AGC_CTRL1[7:0] = 0x100, Attack time = 128 * 0.125ms = 16ms. Tsample time is the recording internal of ADC. For example, when 8 kHz sample rate is applied, Tsample is 1/8K = 0.125ms. The definition of release time is how fast the compressor responses to the sound data peak or to RMS being lower than the threshold. The fast release time will result in the pump of breathing effect situation but long release time lowers the overall volume. User can try both settings and find the best one for the application.



Auto Mute:

- (a) Function: It is capable of reducing background noise after user performs AGC.
- (b) Threshold (0x4012100C Bit[31:16]): 16-bit. It determines the Threshold will enter Mute Mode or will return from mute mode to Normal Mode. The larger the Threshold is given, the easier it enters Mute Mode.
- (c) Normal and Mute Debounce (16-bit):
 - (1) When the current MIC volume reaches the Threshold user configures, the Debounce value will determine after how long the Threshold turns from normal mode to mute mode, or returns to normal mode from mute mode.
 - (2) Mute Debounce (0x40121010 Bit[15:0]): When it's Normal Mode and the current MIC volume is lower than the Threshold, Mute debounce counter value will +1. When the total count exceeds the setup value of Mute debounce, the Threshold enters Mute Mode.
 - (3) Normal Debounce (0x40121010 Bit[31:16]): When it's Mute Mode and the current MIC volume is higher than the Threshold, Normal debounce counter value will +1. When the total count exceeds the setup value of Normal debounce, the Threshold returns to Normal Mode.

(d) RAMP Counter:

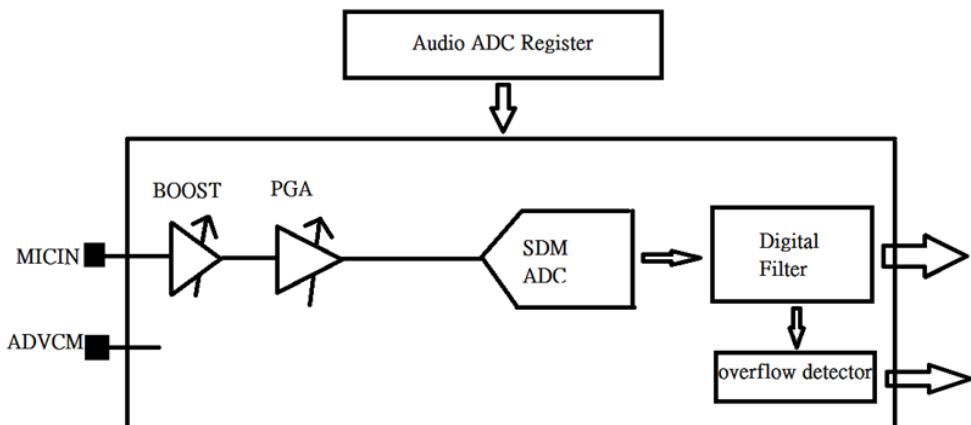
- (1) It is used to adjust the slope of entering Mute mode or returning to Normal mode, and there're eight adjustable levels in total.
- (2) When Normal mode is gradually entering Mute mode, every time Ramp Count -1, $1/8 \times \text{ADC}$ data will be decreased.
- (3) When Mute mode is gradually returning to Normal mode, every time Ramp Count +1, $1/8 \times \text{ADC}$ data will be increased.
- (4) RAMP Counter Step (0x4012100C Bit[3:2]): 2-bit. It is used to determine Ramp Counter will +1 or -1 after how many times of count. For example, if it is set as 03, the Counter will +1 or -1 after counting for more than three times.

14.2. Feature

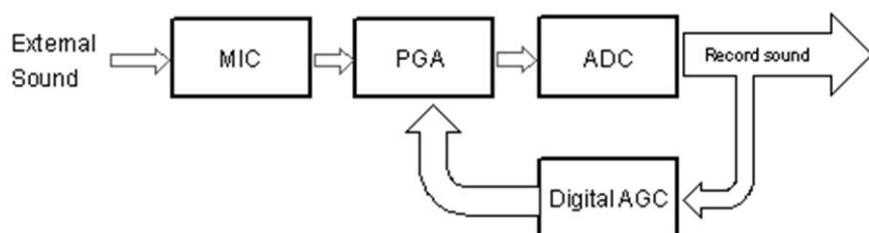
- 16-bit Delta-Sigma ADC
- ADC High Pass Filter
- Boost Gain Control (4-Level)
- PGA (Programmable Gain Amplifier) Control (32-Level)
- AGC (Auto Gain Control)
- Auto Mute Function
- Support PDM microphone.

14.3. Block Diagram

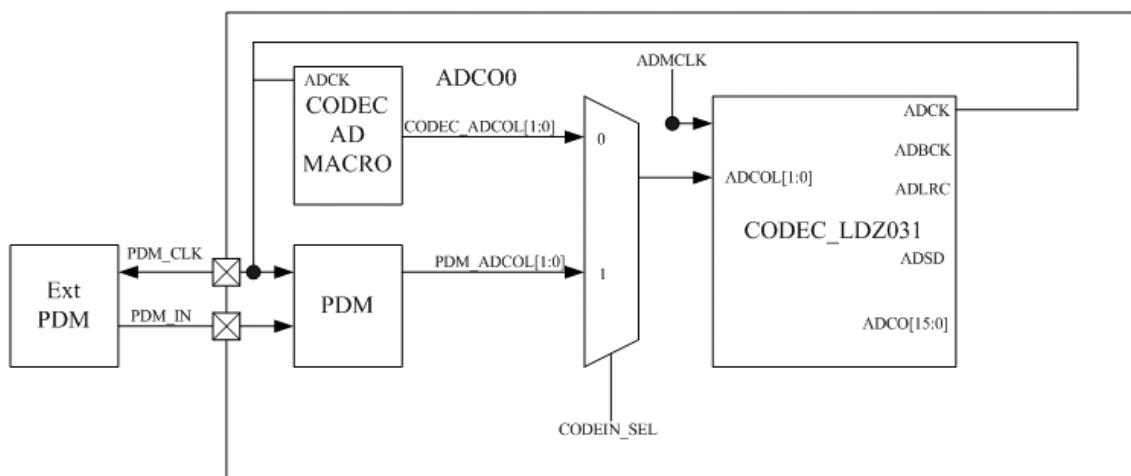
MIC_IN Analog Input signal will be amplified for the first time after passing through Booster gain, and then be reamplified after passing through PGA. After that, it will go through ADC and be converted from analog to digital signal before Digital Filter performs DAGC and Auto Mute processing.



The compressor detects the output level before the main volume multiplier, and it will control the main volume dynamically. Please refer to the following diagram for more details.



External PDM function is supported by serial digital data input and decoded by CODEC module.



PDM Microphone:

- (a) Select PDM IO pins via IOA[6:5] or IOA[23:22]. (IOFUNC->CTRL0 [29]: PDM_IOSEL setting)
 - (b) The audio source of Mic can be input via DS-ADC or PDM. When PDM is enabled, Mic-ADC is disabled and PDM Mic's Data will be stored in DS_ADC_DATA Register.
 - (c) Control registers relevant to PDM:
 - (1) DS_ADC_CTRL[28]: Right-Channel Data
 - (2) DS_ADC_CTRL[30:29]: Configure where the PDM_SAMP_Sel is resampled (Rising/Falling)
 - (3) DS_ADC_CTRL[31]: PDM Enable
- Note:** Support mono-channel input only, not dual-channel PDM.

14.4. Register Description

Register Map

Name	Address	Description
DS_ADC_CTRL	0x40121000	Delta-Sigma ADC Control Register
DS_ADC_AGC_CTRL0	0x40121004	Delta-Sigma ADC AGC Control Register 0
DS_ADC_AGC_CTRL1	0x40121008	Delta-Sigma ADC AGC Control Register 1
DS_ADC_MUTE_CTRL0	0x4012100C	Delta-Sigma ADC MIC Auto-Mute Control Register 0
DS_ADC_MUTE_CTRL1	0x40121010	Delta-Sigma ADC MIC Auto-Mute Control Register 1
DS_ADC_STS	0x40121018	Delta-Sigma ADC Status Register
DS_ADC_DATA	0x4012101C	Delta-Sigma ADC Data Register

Register Function

DS_ADC_CTRL								
Bit	31	30	29	28	27	26	25	24
Function	PDM_ENAB LE	PDM_RESAMPLE_SEL [1:0]	PDM_RCH_ DATA_SEL			-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	FIFO_LVL[2:0]			-	INT_ENABL E	FIFO_OVWR _ENABLE	FIFO_ENABL E
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	FMT_SEL	-		PGA_GAIN[4:0]				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	IN_LIMIT[1:0]		BOOST_GAIN[1:0]	DSADC_ENA BLE	DSADC_RES ET	HPF_ENABL E	MIC_ENABL E	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31]	PDM_ENABLE	R/W	External PDM enable	0: Disable 1: Enable
[30:29]	PDM_RESAMPLE_SEL[1:0]	R/W	PDM Resampling Selection	00: LCH resample on clock rising, RCH resample on clock falling. 01: LCH resample on clock falling, RCH resample on clock rising. 10: LCH resample on clock rising, RCH resample on clock rising. 11: LCH resample on clock falling, RCH resample on clock falling.
[28]	PDM_RCH_DATA_SEL	R/W	Right-Channel Data of PDM	0: R-CH Data = 0 1: R-CH Data = L-CH Data
[27:23]	-	R	Reserved	-
[22:20]	FIFO_LVL[2:0]	R/W	MIC FIFO_LEVEL When the data numbers in MIC FIFO is greater than "MIC_FIFO_LEVEL", FIFO_FULL_FLAG will be set as 1.	0x0: FIFO_LVL_0 0x1: FIFO_LVL_1 0x2: FIFO_LVL_2 0x3: FIFO_LVL_3 0x4: FIFO_LVL_4 0x5: FIFO_LVL_5 0x6: FIFO_LVL_6 0x7: FIFO_LVL_7

Bit	Function	Type	Description	Condition
[19]	-	R	Reserved	-
[18]	INT_ENABLE	R/W	MIC FIFO Full INT Enable If this bit is enabled, when MIC FIFO data numbers are greater than "MIC_FIFO_LEVEL" or the FIFO is full, MIC FIFO can issue an interrupt to CPU. If this bit is disabled, CPU has to confirm MIC FIFO status by polling "MIC_FIFO_FULL" flag.	0: MIC FIFO interrupt is disabled. 1: MIC FIFO interrupt is enabled.
[17]	FIFO_OVWR_ENABLE	R/W	MIC FIFO Overwrite When MIC FIFO is full, this register is used to determine whether the newest data will be overwritten by MIC FIFO or not.	0: The newest MIC data won't be overwritten by MIC FIFO when MIC FIFO is full. 1: The newest MIC data will be overwritten by MIC FIFO when MIC FIFO is full.
[16]	FIFO_ENABLE	R/W	MIC FIFO Enable This register determines whether to turn on MIC FIFO or not. If this bit is disabled, MIC data must be read by DMA engine real time.	0: Disable 1: Enable
[15]	FMT_SEL	R/W	Delta-Sigma ADC Data Sign/Unsigned Selection	0: Unsigned Data 1: Sign Data
[14:13]	-	R	Reserved	-
[12:8]	PGA_GAIN[4:0]	R/W	Delta-Sigma ADC PGA GAIN 0x1F (Min.) ~ 0x00 (Max.)	0x00 = LV31 (33 db) 0x01 = LV30 (31.5 db) 0x02 = LV29 (30 db) 0x03 = LV28 (28.5 db) 0x04 = LV27 (27 db) 0x05 = LV26 (25.5 db) 0x06 = LV25 (24 db) 0x07 = LV24 (22.5 db) 0x08 = LV23 (21 db) 0x09 = LV22 (19.5 db) 0x0A = LV21 (18 db) 0x0B = LV20 (16.5 db) 0x0C = LV19 (15 db) 0x0D = LV18 (13.5 db) 0x0E = LV17 (12 db) 0x0F = LV16 (10.5 db) 0x10 = LV15 (9 db) 0x11 = LV14 (7.5 db) 0x12 = LV13 (6 db)

Bit	Function	Type	Description	Condition
				0x13 = LV12 (4.5 db) 0x14 = LV11 (3 db) 0x15 = LV10 (1.5 db) 0x16 = LV9 (0 db) 0x17 = LV8 (-1.5 db) 0x18 = LV7 (-3 db) 0x19 = LV6 (-4.5 db) 0x1A = LV5 (-6 db) 0x1B = LV4 (-7.5 db) 0x1C = LV3 (-9 db) 0x1D = LV2 (-10.5 db) 0x1E = LV1 (12 db) 0x1F = LV0 (-∞ db)
[7:6]	IN_LIMIT[1:0]	R/W	ADC Input Limit Range Selection	00: 0.84*Full Range 01: 0.71*Full Range 10: 0.60*Full Range 11: 0.50*Full Range
[5:4]	BOOST_GAIN[1:0]	R/W	Boost Amplifier Gain Selection	00 : 0db 01 : 7db 10 : 14db 11 : 21db
[3]	DSADC_ENABLE	R/W	Delta-Sigma ADC Enable	0: Disabled 1: Enabled
[2]	DSADC_RESET	R/W	Delta-Sigma ADC Reset Note: This bit has to be set as 1 when ADC is enabled.	0: Reset 1: Normal Operation
[1]	HPF_ENABLE	R/W	ADC High Pass Filter Enable	0: Disabled 1: Enabled
[0]	MIC_ENABLE	R/W	Enable Microphone Amplifier	0: Disabled 1: Enabled

DS_ADC_AGC_CTRL0
0x40121004
Delta-Sigma ADC AGC Control Register 0

Bit	31	30	29	28	27	26	25	24
Function	-	TOGGLE_THRESHOLD[14:8]						
Reset Value	0	0	0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
Function	TOGGLE_THRESHOLD[7:0]							
Reset Value	1	1	1	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	THRESHOLD[6:0]						
Reset Value	0	0	0	0	0	1	0	0

Bit	7	6	5	4	3	2	1	0
Function	-					DAGC_ENA BLE	ZERO_CROS S_DISABLE	MODE_SEL
Reset Value	0	0	0	1	1	0	0	0

Bit	Function	Type	Description	Condition
[31]	-	R	Reserved	-
[30:16]	TOGGLE_THRESHOLD[14:0]	R/W	TOGGLE_THRESHOLD Register This register is used to determine the update threshold of digital AGC. When the difference in-between input volume and setting threshold is smaller this value, the PGA gain will be kept, otherwise, the PGA gain will be updated. This register can improve the stability of PGA gain but it will reduce the sensitivity of digital AGC.	-
[15]	-	R	Reserved	-
[14:8]	THRESHOLD[6:0]	R/W	Digital AGC Threshold Control The minimum value is 0x01, and the maximum value is 0x7F. Note: 0x00 is not allowed.	-
[7:3]	-	R	Reserved	-
[2]	DAGC_ENABLE	R/W	Digital AGC Enable Note: When programmer enables this bit, the original PGA_GAIN setting will be discarded, and digital AGC will take over the control of PGA_GAIN.	0: Disable DAGC. (Default) 1: Enable DAGC.
[1]	ZERO_CROSS_DISABLE	R	Disable Zero Cross Function of Compressor. As described before, the compressor will adjust the main volume automatically. This bit determines whether AGC adjusts the volume gain after zero cross or not.	0: Enable zero cross. 1: Disable zero cross.
[0]	MODE_SEL	R	Peak Mode Control Register	0: RMS Mode 1: Peak Mode

DS_ADC_AGC_CTRL1 **0x40121008** **Delta-Sigma ADC AGC Control Register 1**

Bit	31	30	29	28	27	26	25	24
Function	UPDATE_FREQ[11:4]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	UPDATE_FREQ[3:0]				RELEASE_SCALE[1:0]		RELEASE_TIME[7:6]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	RELEASE_TIME[5:0]						ATTACK_SCALE[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ATTACK_TIME[7:0]							
Reset Value	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[31:20]	UPDATE_FREQ[11:0]	R/W	Update Frequency Control (12-bit Resolution) Note: It is used to set up the AGC refresh rate. That is, the volume gain will be adjusted after how many samples.	-
[19:18]	RELEASE_SCALE[1:0]	R/W	Release Time Scale Control	00 = Release Time*1 01 = Release Time*4 10 = Release Time*16 11 = Release Time*64
[17:10]	RELEASE_TIME[7:0]	R/W	Release Time Control The definition of release time is how fast the compressor responses to the wave data peak or to RMS being lower than the threshold.	-
[9:8]	ATTACK_SCALE[1:0]	R/W	Attack Time Scale Control	00 = Attack Time*1 01 = Attack Time*4 10 = Attack Time*16 11 = Attack Time*64
[7:0]	ATTACK_TIME[7:0]	R/W	Attack Time Control The definition of attack time is how fast the compressor responses to the wave data peak or to RMS exceeding the threshold.	-

DS_ADC_MUTE_CTRL0
0x4012100C
Delta-Sigma ADC MIC Auto-Mute Control
Register 0

Bit	31	30	29	28	27	26	25	24
Function	SILENCE_TH[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	SILENCE_TH[7:0]							
Reset Value	1	1	1	1	0	1	0	0
Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-			ECHO_CAN CEL_ENABL E	RAMPCNT_STEP[1:0]		FMT_SEL	MIC_AUTO MUTE_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	SILENCE_TH[15:0]	R/W	Silence Threshold In normal mode, if the wave in data is smaller than silence threshold value, and it's kept for more than silence debounce count value, it will change to silence mode. Or it will keep in normal mode. In silence mode, if the wave in data is bigger than silence threshold value, and it's kept for more than normal debounce count value, it will return to normal mode. Or it will keep in silence mode.	-
[15:5]	-	R	Reserved	-
[4]	ECHO_CANCEL_ENABLE	R/W	Echo Canceling Control Bit This bit is determines whether bit[31:16] of DS_ADC_DATA register contain DAC conversion data or not.	0: DS_ADC_DATA[31:16] is "0x0000". 1: DS_ADC_DATA[31:16] contains DAC conversion data.
[3:2]	RAMPCNT_STEP[1:0]	R/W	RAMP Counter Step RAMPCNT_STEP determines after how many counts that Ramp Counter will -1 (Decreasing step, normal mode) or +1 (Increasing step, silence mode). This value means the sample counts in each step. For example, RAMPCNT_STEP = 0x3 means 3 sample data = 1 step.	00: 0 step 01: 1 step 10: 2 steps 11: 3 steps
[1]	FMT_SEL	R/W	MIC AGC Signed/Unsigned Data Selection This bit is used to determine the MSB bit of Delta-Sigma conversion data is signed or unsigned.	0: Unsigned 1: Signed
[0]	MIC_AUTOMUTE_EN	R/W	Auto Mute Function Control Bit	0: Disabled 1: Enabled

DS_ADC_MUTE_CTRL1
0x40121010
Delta-Sigma ADC MIC Auto-Mute Control
Register 1

Bit	31	30	29	28	27	26	25	24
Function	NORMAL_DEBOUNCE[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	NORMAL_DEBOUNCE[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	SILENCE_DEBOUNCE[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SILENCE_DEBOUNCE[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	NORMAL_DEBOUNCE[15:0]	R/W	Normal Debounce Value NORMAL_DEBOUNCE is the debounce count to enter the normal mode.					-
[15:0]	SILENCE_DEBOUNCE[15:0]	R/W	Mute Debounce Value SILENCE_DEBOUNCE is the debounce count to enter the mute mode.					-

0x40121018								Delta-Sigma ADC Status Register	
Bit	31	30	29	28	27	26	25	24	
Function						-			
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function			-		FIFO_FULL_FLAG	INT_FLAG	-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-					CURPGA_GAIN[4:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function			-					
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:20]	-	R	Reserved					-
[19]	FIFO_FULL_FLAG	R/W	MIC FIFO Full Flag Note: This bit will be cleared as 0 automatically when data is read from MIC FIFO.					0: Not full. 1: MIC FIFO is full or greater than MIC_FIFO_LEVEL.
[18]	INT_FLAG	R/W	Delta-Sigma ADC Interrupt Flag Write 1 to clear this flag.					0: No DS_ADC INT occurs. 1: DS_ADC INT occurs.

Bit	Function	Type	Description	Condition
[17:13]	-	R	Reserved	-
[12:8]	CURPGA_GAIN[4:0]	R	Current PGA Gain Value CURPGA_GAIN indicates the current PGA_GAIN gain to Delta-Sigma ADC.	-
[7:0]	-	R	Reserved	-

DS_ADC_DATA **0x4012101C** **Delta-Sigma Data Register**

Bit	31	30	29	28	27	26	25	24
Function	DAC_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	DAC_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DS_ADC_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DS_ADC_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	DAC_DATA[15:0]	R	DAC Data for Echo Application Whether to store DAC conversion data or not is determined by ECHO_CANCEL_EN bit. Note: With this function, it's able to obtain DMA ADC and DAC data synchronously for reducing the effect of audio feedback.	-
[15:0]	DS_ADC_DATA[15:0]	R	DS_ADC Data Register [15:0]: Delta-Sigma ADC Data Note: If MIC_MUTE_EN is enabled, the data is processed by Auto-Mute module. Else, the data will be the Delta-Sigma ADC conversion data.	-

15. DAC Up-Sampling/Audio PWM Control

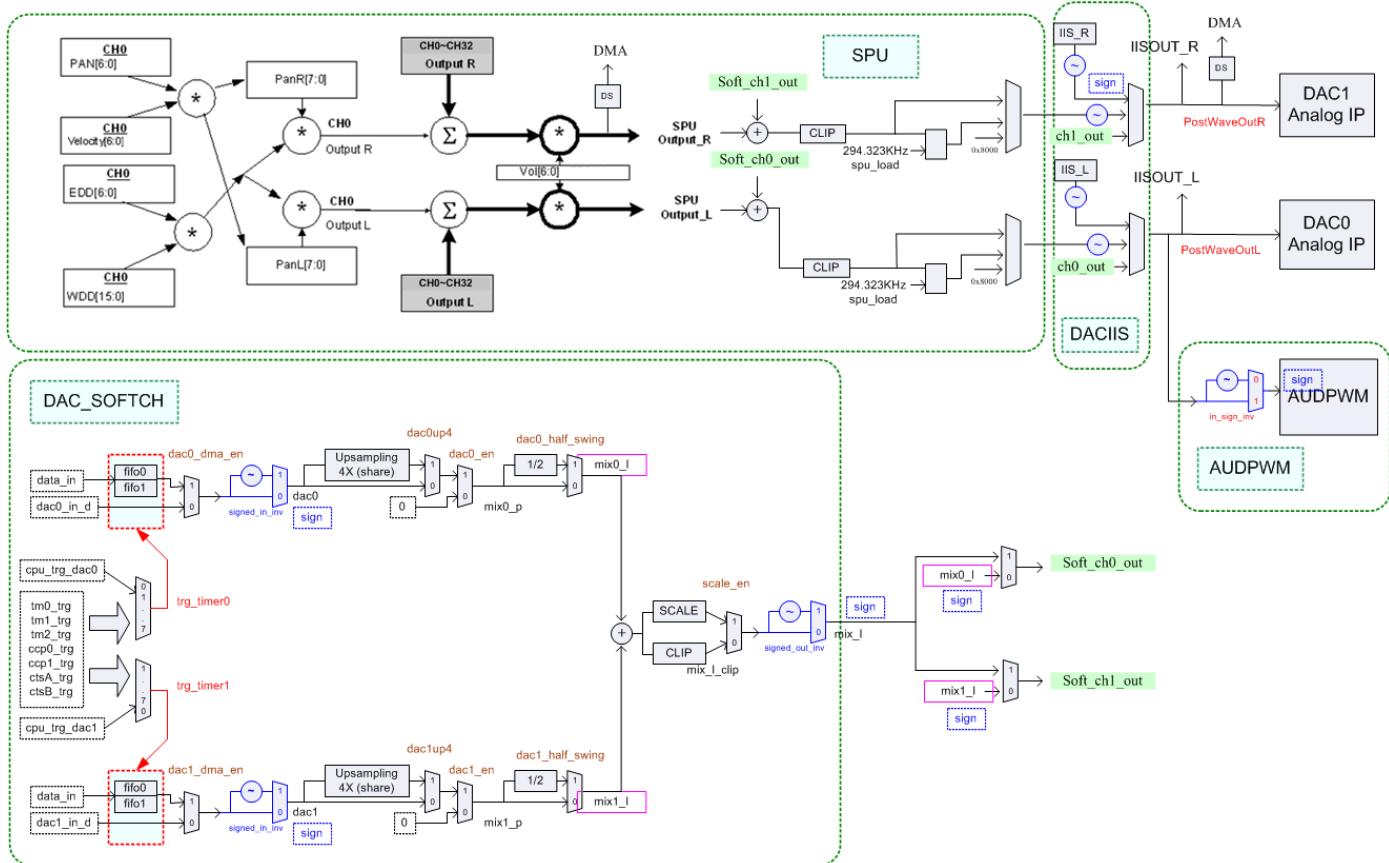
15.1. Introduction

The audio of GPCM3 SERIALS can be exported to the Voltage DAC or AUDPWM. The DAC data ranges from 0x0000 to 0xFFFF (16-bit DAC). To make DAC data become PCM data, user has to know the DC level for the DAC Data is 0x0000. The data of DAC should be delivered to DAC_CH0_DATA(W) (0x4013_0010) and DAC_CH1_DATA(W) (0x4013_0014). GPCM3 SERIALS also supports writing Ch0/1 data in through DMA. GPCM3 SERIALS supports two input channels data port and new 4x up-sampling function. User is allowed to send various data to different channels, and then, the IC will perform mix operation before DAC conversion.

15.2. Feature

- DAC mode supports stereo.
- Support 4x up-sampling.
- Support 2 input channels.
- Support DMA input mode.
- Support scale/truncation mode when data saturation happens.
- Support digital gain control (DAC mode and PWM mode).
- Audio PWM supports auto mute control.

15.3. Block Diagram



15.4. Function

DAC SOFTCH module can enable DAC CH0 or CH1 separately by configuring DAC_CHx_EN. The hardware samples CH0/CH1 input data when trigger pulse happens (by configuring CHx_TRG_SEL). User can enable/disable up-sampling mode by configuring DAC_CH0/1_UPSMP_MODE[1:0].

In addition to be used in audio output, AUDP/AUDN can be used in other purposes such as output high/output low/input pull high/input pull low. The following table depicts the available settings.

DACR_EN	DACL_EN	AUDPWM_IP_ENABLE	AUDPWM_EN	AUDP	AUDN
0	0	0	0	Pull High (AUDPI = 1) Pull Low (AUDPI = 0)	Pull High (AUDNI = 1) Pull Low (AUDNI = 0)
0	0	0	1	Pull High (AUDPI = 1) Pull Low (AUDPI = 0)	Pull High (AUDPI = 1) Pull Low (AUDPI = 0)
0	0	1	0	AUDP (AUDPI)	AUDN (AUDNI)
0	0	1	1	AUDP (PWM_AUDP)	AUDN (PWM_AUDN)
0	1	0	0	Pull High (AUDPI = 1) Pull Low (AUDPI = 0)	Output High
0	1	0	1	Pull High (AUDPI = 1) Pull Low (AUDPI = 0)	DAC-L
0	1	1	0	AUDP (AUDPI)	Output High
0	1	1	1	AUDP (AUDPI)	DAC-L
1	0	0	0	Output High	Pull High (AUDNI = 1) Pull Low (AUDNI = 0)
1	0	0	1	DAC-R	Pull High (AUDNI = 1) Pull Low (AUDNI = 0)
1	0	1	0	Output High	AUDN (AUDNI)
1	0	1	1	DAC-R	AUDN (AUDNI)
1	1	0	0	Output High	Output High
1	1	0	1	DAC-R	DAC-L
1	1	1	0	Output High	Output High
1	1	1	1	DAC-R	DAC-L

Output Mode:

Voltage DAC Mode:

After power on reset, the DAC analogy IP is in its default state, which means it's turned off. So if users wants to use the DAC, it can be turned on by setting VOLTAGE_DAC_ENABLE(0x40130000[23]) = 1.

AUDIO PWM Mode:

Set AUDPWMEN = 1 to enable AUDIO PWM analogy IP output, and set AUDPWM_EN = 1 to start Audio PWM generator. User can also configure AUDPWM_GAIN to re-scale audio data from 1/32x to 2x.

In Audio PWM mode, only the data of Postwave_CH0 can be output. Postwave_CH1 is only valid in DAC mode.

Input Mode:

Manual Mode:

User has to deliver the new updated DAC data to DAC_CH0_IN/DAC_CH1_IN.

DMA Mode:

User can also use DMA control unit to deliver DAC data. The hardware will fill DAC FIFO automatically when the FIFO is not full (by configuring DAC_CHx_DMA_ENABLE)

15.5. Register Description
Register Map

Name	Address	Description
DAC_CTRL	0x40130000	DAC Control Register
DAC_STS	0x40130004	DAC Status Register
AUDPWM_CTRL2	0x40130008	AUDIO PWM Control Register 2
AUDPWM_CTRL	0x4013000C	AUDIO PWM Control Register
DAC_CH0_DATA	0x40130010	DAC Channel0 Input Data
DAC_CH1_DATA	0x40130014	DAC Channel1 Input Data
MIX_DATA_OUT	0x40130018	DAC Mixed Output Data
DAC_CH0_DMA_DATA0	0x40130020	DAC Channel0 DMA FIFO Data0 Value
DAC_CH0_DMA_DATA1	0x40130024	DAC Channel0 DMA FIFO Data1 Value
DAC_CH1_DMA_DATA0	0x40130030	DAC Channel1 DMA FIFO Data0 Value
DAC_CH1_DMA_DATA1	0x40130034	DAC Channel1 DMA FIFO Data1 Value

Register Function

DAC_CTRL									0x40130000		DAC Control Register	
Bit	31	30	29	28	27	26	25	24				
Function	DAC_CH1_DMAERR_INT_ENABLE	DAC_CH0_DMAERR_INT_ENABLE	DAC_CH1_INTEGRITY_ENABLE	DAC_CH0_INTEGRITY_ENABLE	CH1_SRC	CHO_SRC	UPSMP_SW_RESET	DAC_SW_RESET				
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16				
Function	VOLTAGE_DA_C_ENABLE	CH1_POST_WAVE_OUT_SRC	DACOUT_FM_TSEL	DACIN_FMT_SEL	CHO_POST_WAVE_OUT_SRC	-	-	-	SCALE_ENA_BLE			
Reset Value	0	0	0	0	0	0	0	0	0			

Bit	15	14	13	12	11	10	9	8				
Function	CH1_HALF_ENABLE		DAC_CH1_TRG_SEL[2:0]		DAC_CH1_UPSMP_MODE[1:0]	DAC_CH1_DMA_ENABLE	DAC_CH1_ENABLE					
Reset Value	0	0	0	0	0	0	0	0				

Bit	7	6	5	4	3	2	1	0				
Function	DAC_CH0_HALF_ENABLE		DAC_CH0_TRG_SEL[2:0]		DAC_CH0_UPSMP_MODE[1:0]	DAC_CH0_DMA_ENABLE	DAC_CH0_ENABLE					
Reset Value	0	0	0	0	0	0	0	0				

Bit	Function	Type	Description	Condition
[31]	DAC_CH1_DMAERR_INT_ENABLE	R/W	DAC Channel1 DMA FIFO Error Interrupt Enable	0: Disabled 1: Enabled
[30]	DAC_CH0_DMAERR_INT_ENABLE	R/W	DAC Channel0 DMA FIFO Error Interrupt Enable	0: Disabled 1: Enabled

Bit	Function	Type	Description	Condition
[29]	DAC_CH1_INT_ENABLE	R/W	DAC Channel1 Interrupt Enable	0: Disabled 1: Enabled
[28]	DAC_CH0_INT_ENABLE	R/W	DAC Channel0 Interrupt Enable	0: Disabled 1: Enabled
[27]	CH1_SRC	R/W	DAC Channel1 Output Selection	0: DAC1 = CH0+CH1 1: DAC1 = CH1
[26]	CH0_SRC	R/W	DAC Channel0 Output Selection	0: DAC0 = CH0+CH1 1: DAC0 = CH0
[25]	UPSMP_SW_RESET	R/W	Up Sample Function Software Reset	0: No effect. 1: Reset is asserted, and it will de-assert automatically when reset is done.
[24]	DAC_SW_RESET	R/W	DAC Control Software Reset	0: No effect. 1: Reset is asserted, and it will de-assert automatically when reset is done.
[23]	VOLTAGE_DAC_ENABLE	R/W	Current DAC Enable	0: Disabled 1: Enabled
[22]	CH1_POSTWAVE_OUT_SRC	R/W	Post Wave Only Output Software Ch1 Data	0: SW CH + SPU 1: Only SW CH
[21]	DACOUT_FMT_SEL	R/W	Mixed Data Signed to Unsigned Enable (Default: Output unsigned data to DAC.)	0: Unsigned 1: Signed
[20]	DACIN_FMT_SEL	R/W	Input Data Signed/Unsigned Selection	0: Signed 1: Unsigned
[19]	CH0_POSTWAVE_OUT_SRC	R/W	Post Wave Only Output Software Ch0 Data	0: SW CH + SPU 1: Only SW CH
[18:17]	-	R	Reserved	-
[16]	SCALE_ENABLE	R/W	Saturation Mode Enable	0: Disabled (Truncation) 1: Enabled
[15]	CH1_HALF_ENABLE	R/W	DAC Channel1 Volume Control	0: Disabled 1: Enabled
[14:12]	DAC_CH1_TRG_SEL[2:0]	R/W	DAC Channel1 Trigger Source Selection	000: Manual 001: TM0 010: TM1 011: TM2 100: CCP0_TM 101: CCP1_TM 110: CTS_TM0 111: CTS_TM1

Bit	Function	Type	Description	Condition
[11:10]	DAC_CH1_UPSMP_MODE[1:0]	R/W	DAC Channel1 Up Sampling Selection	00: Disabled 01: 1x sampling (with 4x timer) 1x: 4x up sampling (with 4x timer)
[9]	DAC_CH1_DMA_ENABLE	R/W	DAC Channel1 DMA Enable Configure DMA control unit before user enables this function.	0: Disabled 1: Enabled
[8]	DAC_CH1_ENABLE	R/W	DAC Channel1 Enable	0: Disabled 1: Enabled
[7]	DAC_CH0_HALF_ENABLE	R/W	DAC Channel0 Volume Control	0: Disabled 1: Enabled
[6:4]	DAC_CH0_TRG_SEL[2:0]	R/W	DAC Channel0 Trigger Source Selection	000: Manual 001: TM0 010: TM1 011: TM2 100: CCP0_TM 101: CCP1_TM 110: CTS_TM0 111: CTS_TM
[3:2]	DAC_CH0_UPSMP_MODE[1:0]	R/W	DAC Channel0 Up Sampling Selection	00: Disable 01: 1x sampling (with 4x timer) 1x: 4x up sampling (with 4x timer)
[1]	DAC_CH0_DMA_ENABLE	R/W	DAC Channel0 DMA Enable Configure DMA control unit before user enables this function.	0: Disabled 1: Enabled
[0]	DAC_CH0_ENABLE	R/W	DAC Channel0 Enable	0: Disabled 1: Enabled

DAC_STS									0x40130004		DAC Status Register			
Bit	31	30	29	28	27	26	25	24						
Function	DAC_CH1_E RR_FLAG	DAC_CH0_E RR_FLAG	DAC_CH1_I NT_FLAG	DAC_CH0_I NT_FLAG										
Reset Value	0	0	0	0	0	0	0	0						

Bit	23	22	21	20	19	18	17	16
Function			-					
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31]	DAC_CH1_ERR_FLAG	R/W	DAC Channel1 DMA FIFO Error Flag When the flag is asserted, write 1 to clear this bit.					-
[30]	DAC_CH0_ERR_FLAG	R/W	DAC Channel0 DMA FIFO Error Flag When the flag is asserted, write 1 to clear this bit.					-
[29]	DAC_CH1_INT_FLAG	R/W	DAC Channel1 Interrupt Flag When the flag is asserted, write 1 to clear this bit.					-
[28]	DAC_CH0_INT_FLAG	R/W	DAC Channel0 Interrupt Flag When the flag is asserted, write 1 to clear this bit.					-
[27:0]	-	R	Reserved					-

AUDPWM_CTRL2								
Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				MUTE_OUT_SEL[6]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function							MUTE_OUT_CLEAR_TIM_E[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:2]	MUTE_OUT_SEL[6:0]	R/W	Exit Auto Mute Timing Setting For MuteCLK, please refer to the configuration of 0x4013000C[11:10].					7'h7F => MuteCLK * 1 7'h7E => MuteCLK * 2 7'h7D => MuteCLK * 3 7'h00 => MuteCLK * 128

Bit	Function	Type	Description		Condition
[1:0]	MUTE_OUT_CLEAR_TIME[1:0]	R/W	Recount MUTE_TIME_OUT Control Setting MUTE_OUT_CLEAR_TIME determines the recount timing of Auto Mute Timer Counter. If the data hasn't change in 64us~512us, Auto Mute Counter value will be cleared, and the counter will recount for 32us~32.768ms. If the data keep changing in 32us~32.768ms, the timer will exit mute mode. MuteOut counter will be cleared if the data hasn't changed for several time intervals. 00 means MuteOut counter will be cleared if the 1024 pieces of AUD_CLK data haven't changed.		00: AUD_CLK * 1024 01: AUD_CLK * 2048 10: AUD_CLK * 4096 11: AUD_CLK * 8192

AUDPWM_CTRL 0x4013000C AUDIO PWM Control Register							
Bit	31	30	29	28	27	26	25
Function		-			MUTE_IN_SEL[6:3]		
Reset Value	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17
Function	MUTE_IN_SEL[2:0]			AUDP_STAT_E	AUDN_STAT_E	DAC_CH1_I_P_ENABLE	AUDPWM_I_P_ENABLE
Reset Value	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9
Function	DS_SEL	MUTE_TYPE	-		MUTE_STATE_SEL[1:0]	AUDPWM_GAIN[5:4]	
Reset Value	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1
Function	AUDPWM_GAIN[3:0]				MUTE_CTRL[1:0]		AUDPWM_T_SEL
Reset Value	0	0	0	0	0	0	0

Bit	Function	Type	Description		Condition
[31:28]	-	R	Reserved		-
[27:21]	MUTE_IN_SEL[6:0]	R/W	Entry Auto Mute Timing Setting MUTE_IN_SEL determines how many MuteCLK the audio data is kept before it enters Mute for no changing. For MuteCLK, please refer to the configuration of 0x4013000C[11:10].		7'h7F => MuteCLK * 2 7'h7E => MuteCLK * 4 7'h7D => MuteCLK * 6 7'h00 => MuteCLK * 256
[20]	AUDP_STATE	R/W	AUDP Output Data Control		0: Low 1: High
[19]	AUDN_STATE	R/W	AUDN Output Data Control		0: Low 1: High
[18]	DAC_CH1_IP_ENABLE	R/W	DAC CH1 Enable Control		0: Disabled 1: Enabled

Bit	Function	Type	Description	Condition
[17]	DAC_CHO_IP_ENABLE	R/W	DAC CHO Enable Control	0: Disabled 1: Enabled
[16]	AUDPWM_IP_ENABLE	R/W	Analog PWM Enable Control	0: Disabled 1: Enabled
[15]	DS_SEL	R/W	Delta-Sigma Type Selection	0: DS3 1: DS6
[14]	MUTE_TYPE	R/W	AUDP/AUDN Output in Auto Mute Mode	0: Floating (PWMOFF) 1: 50% Duty
[13:12]	-	R	--	-
[11:10]	MUTE_STATE_SEL[1:0]	R/W	MuteCLK is the time unit of entering Mute.	00: AUD_CLK/4096 01: AUD_CLK/2048 10: AUD_CLK/1024 11: AUD_CLK/512
[9:4]	AUDPWM_GAIN[5:0]	R/W	Audio PWM Gain Register	6'h00: 1/32 6'h01: 2/32 6'h1f: 32/32 (1x gain) 6'h20: 33/32 6'h3E: 63/32 6'h3F: Forbidden
[3:2]	MUTE_CTRL[1:0]	R/W	Auto Mute Control Setting	00: Auto Mute Off 01: MUTE_TIME_IN determines the timing of entering mute mode. When data changes, it will exit mute mode. 10: MUTE_TIME_IN determines the timing of entering mute mode. The exiting timing is determined by AUDPWM_CTRL2[8:2] MUTE_TIME_OUT (0x40130008). 11: Auto Mute Off
[1]	DATAIN_FMT_SEL	R/W	Audio PWM Input Data Signed/Unsigned Selection	0: Signed 1: Unsigned
[0]	AUDPWM_ENABLE	R/W	Audio PWM Enable	0: Disabled 1: Enabled

DAC_CH0_DATA
0x40130010
DAC Channel0 Input Data

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	DAC_CH0_IN[15:0]	R/W	DAC Channel0 Input Data Value					-

DAC_CH1_DATA
0x40130014
DAC Channel1 Input Data

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	DAC_CH1_IN[15:0]	R/W	DAC Channel1 Input Data Value					-

MIX_DATA_OUT
0x40130018
DAC Mixed Output Data

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					MIX_DATA_OUT[15:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					MIX_DATA_OUT[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	MIX_DATA_OUT[15:0]	R	DAC Mixed Output Data Value					-

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					DAC0_DMA_DATA0[15:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					DAC0_DMA_DATA0[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	DAC0_DMA_DATA0[15:0]	R	DAC Channel0 DMA FIFO Data0 Value Note: FIFO Address Auto Change Mechanism: Ch0 has 2-level FIFO, located at 0x40130020 and 0x40130024 respectively. DMA destination Address is fixed as 0x40130020. The H/W will automatically switch between the two addresses. (For DMA only)					-

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				DAC0_DMA_DATA1[15:8]				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				DAC0_DMA_DATA1[7:0]				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	DAC0_DMA_DATA1[15:0]	R	DAC Channel0 DMA FIFO Data1 Value						-

DAC_CH1_DMA_DATA0									0x40130030	DAC Channel1 DMA FIFO Data0 Value		
Bit	31	30	29	28	27	26	25	24				
Function	-											
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				DAC1_DMA_DATA0[15:8]				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				DAC1_DMA_DATA0[7:0]				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	DAC1_DMA_DATA0[15:0]	R	DAC Channel1 DMA FIFO Data0 Value Note: FIFO Address Auto Change Mechanism: Ch1 has 2-level FIFO, located at 0x40130030 and 0x40130034 respectively. DMA destination Address is fixed as 0x40130030. The H/W will automatically switch between the two addresses. (For DMA only)						-

DAC_CH1_DMA_DATA1									0x40130034	DAC Channel1 DMA FIFO Data1 Value		
Bit	31	30	29	28	27	26	25	24				
Function	-											
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					DAC1_DMA_DATA1[15:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					DAC1_DMA_DATA1[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:0]	DAC1_DMA_DATA1[15:0]	R	DAC Channel1 DMA FIFO Data1 Value	-

16. Serial peripheral interface (SPI)

16.1. Introduction

The GPCM3 incorporates two Serial Peripheral Interfaces (SPI). SPI is a synchronous serial communication interface specification used for short distance communication. It allows half/full-duplex serial communication with external devices. The master device will issue the frame of reading or writing. For multiple slave applications, devices can select which slave device to connect by individual chip select (CS) lines.

16.2. Features

- Support half/full-duplex synchronous transfers.
- 8-bit Transfer Frame Format
- Master or Slave Operation
- Support Multi-master mode.
- Programmable Clock Polarity and Phase
- Transmission and Reception with DMA

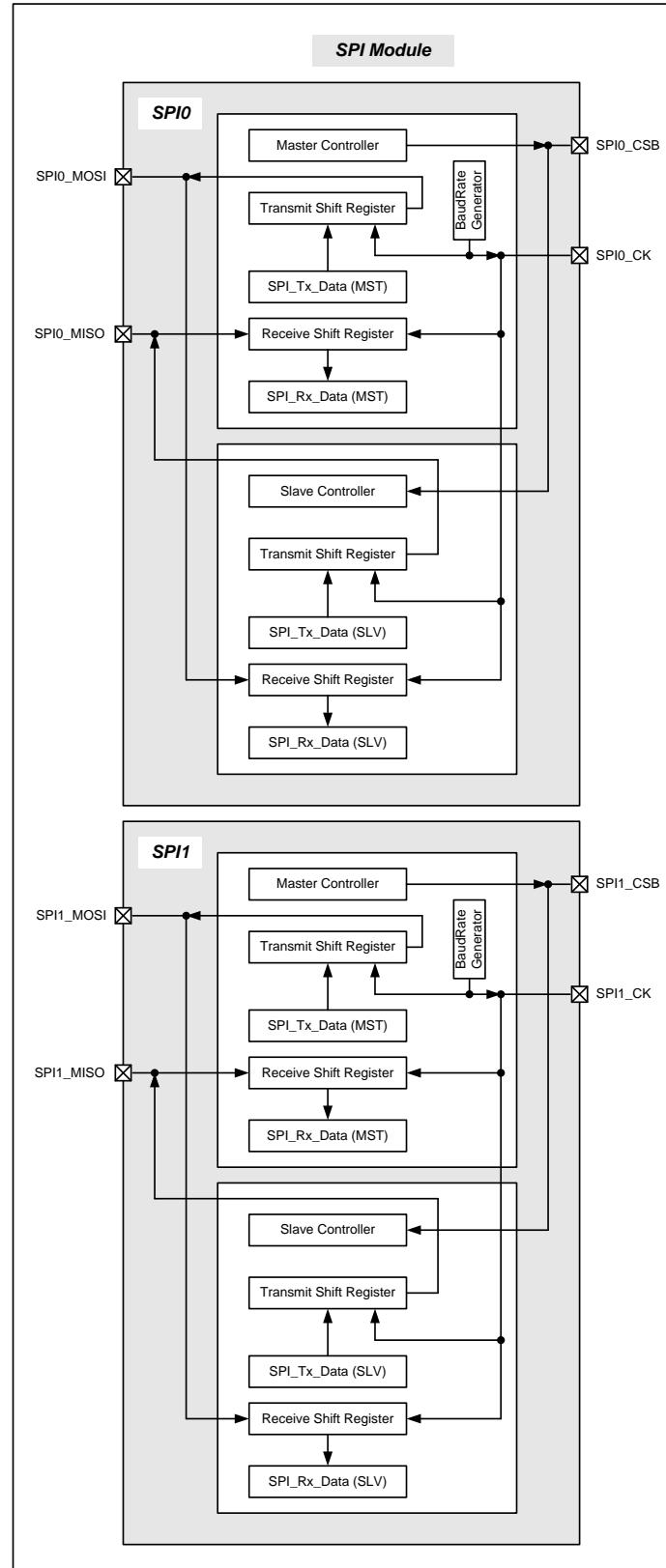
16.3. Serial Interface Control Pin Configuration

16.3.1. SPI0 IO Selection

GPIOFUNC->CTRL0[22:21]	0	0	1	1	2/3	2/3
GPIOFUNC->CTRL0[23]	0	1	0	1	0	1
SPI0_CK	IOA3	IOA3	IOA26	IOA26	IOB1	IOB1
SPI0_MOSI	IOA4	IOA6	IOA27	IOA29	IOB2	IOB4
SPI0_CS _B	IOA5	IOA5	IOA28	IOA28	IOB3	IOB3
SPI0_MISO	IOA6	IOA4	IOA29	IOA27	IOB4	IOB2

16.3.2. SPI1 IO Selection

GPIOFUNC->CTRL0[14:13]	0	0	1	1	2	3
GPIOFUNC->CTRL0[15]	0	1	0	1	x	x
SPI1_CK	IOA9	IOA9	IOA22	IOA22	IOA9	IOA22
SPI1_MOSI	IOA10	IOA12	IOA23	IOA25	IOA10	IOA23
SPI1_CS _B	IOA11	IOA11	IOA24	IOA24	IOA11	IOA24
SPI1_MISO	IOA12	IOA10	IOA25	IOA23	IOA12	IOA25

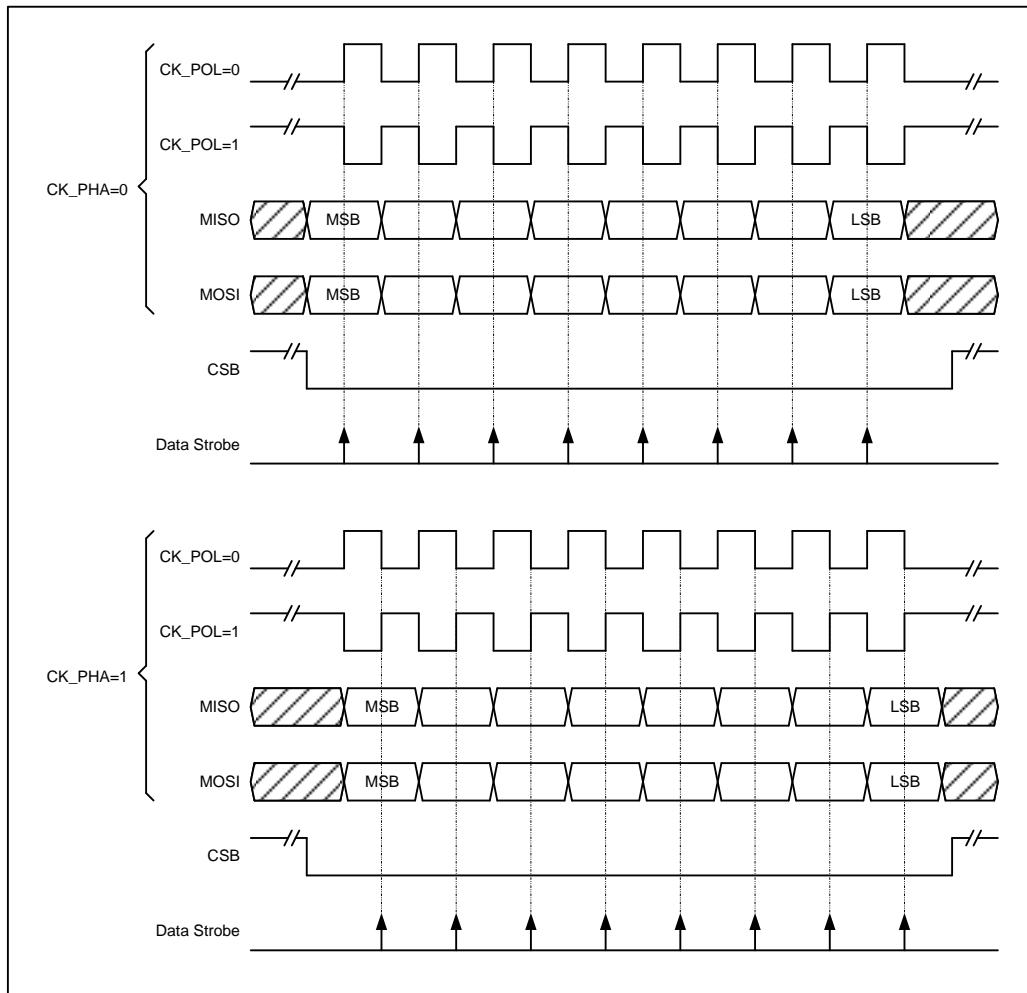
16.4. Block Diagram of SPI


16.5. Function

16.5.1. SPI Clock Phase and Clock Polarity

A standard SPI frame consists of four possible timing relationships which can be selected by configuring **SPIx_CTRL.CK_PHA** and **SPIx_CTRL.CK_POL**. When **SPIx_CTRL.CK_POL = 0**, the idle state of the clock is 0 and the active state is 1. Oppositely, the idle state of the clock is 1 and the active state is 0.

For clock phase choice, data is captured on the clock's rising edge and is changed at the falling edge when **SPIx_CTRL.CK_PHA = 0**. Oppositely, data is captured on the clock's falling edge and is changed at the rising edge when **SPIx_CTRL.CK_PHA = 1**. The following figure shows a SPI transfer with the four combinations of the **SPIx_CTRL.CK_POL** and **SPIx_CTRL.CK_PHA**.



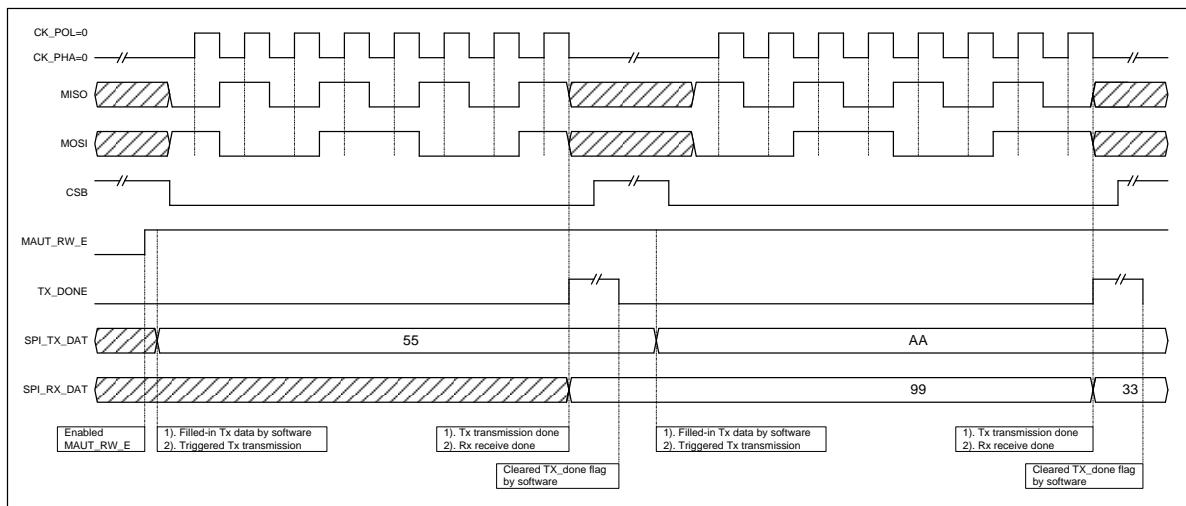
SPI Transfer with the Four Combinations of the SPIx_CTRL.CK_POL and SPIx_CTRL.CK_PHA

16.5.2. Master Mode

16.5.2.1. Single Byte Transmission with Manual Mode

In master mode, before using SPI transmitter function, user must set **SPIx_CTRL.MAUT_RW_EN** as 1. Then, SPI will send out data on MOSI pin immediately when CPU fills data in **SPIx_TX_DATA**. At the same time, CPU will receive the data sent from MISO pin and save the data in **SPIx_RX_DATA**.

During a SPI transmission, data is shifted out from most significant bit (MSB) on the MOSI pin, and then is shifted in from the MISO pin. In this mode, the **SPIx_TX_DATA** register is a buffer between the system bus and the transmit shift register, and **SPIx_RX_DATA** register is a buffer between the system bus and the receive shift register. Every frame consists of 8-bit data bit. The **SPIx_STS.TX_DONE** will be set as 1 when 8 bits data is shifted out by MOSI pin. An interrupt is generated if the **SPIx_CTRL.TX_INT_EN** bit is set as 1. The **SPIx_STS.TX_DONE** flag is set as 1 by the hardware and cleared by the software. After this bit is cleared, fill the next data in **SPIx_TX_DATA** to start the next transfer. The following figure shows a single byte communication with transmitting.

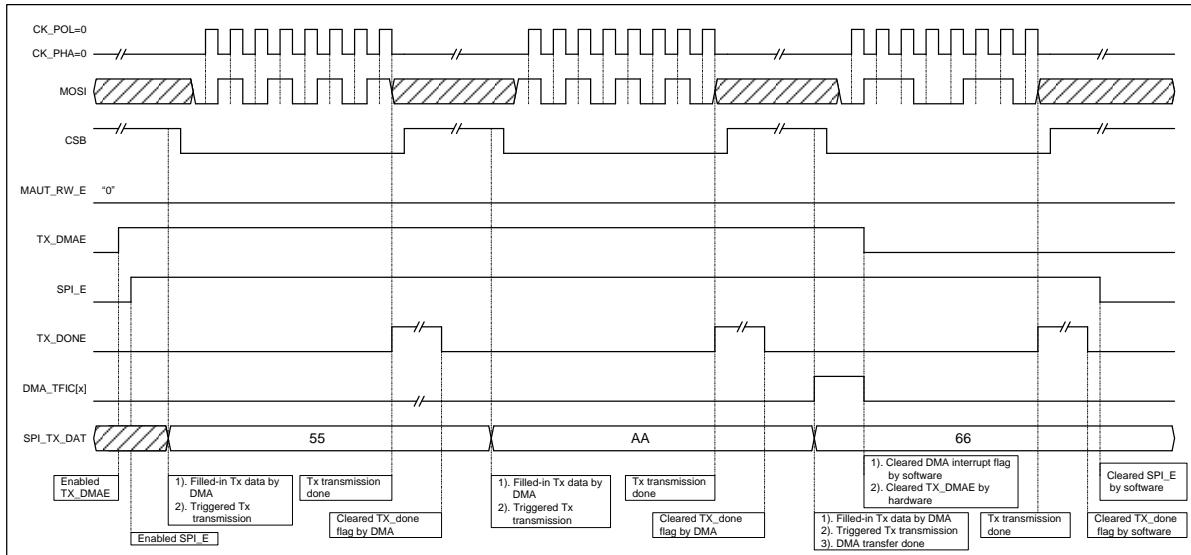


Single Byte Communication with Manual Mode (Master Mode)

16.5.2.2. Transmit with DMA Function

The transmitter of SPI master supports DMA mode which can be enabled by setting **SPIx_CTRL.TX_DMA_EN** as 1. Data is loaded from SRAM and configured by DMA peripheral.

When the number of data transfer is reached, the DMA controller will generate an interrupt. Once all of the transfer data has been written to **SPIx_TX_DATA** by DMA, user can monitor **SPIx_STS.TX_DONE** flag to know whether the SPI communication is completed or not. To avoid the incomplete transmission of the last data, user can only disable the SPI by the software until **SPIx_STS.TX_DONE = 1**.

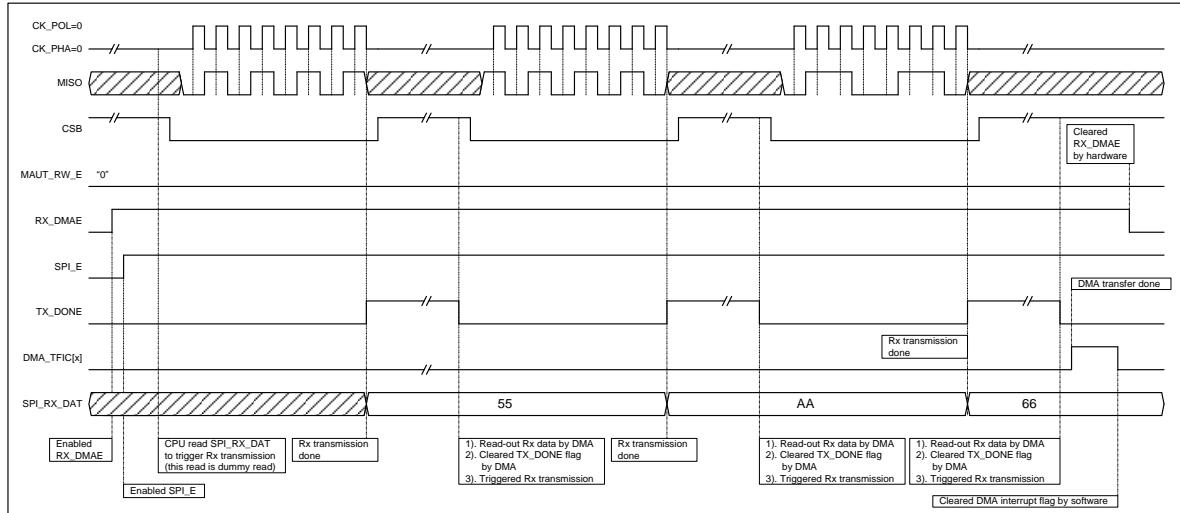


SPI Transmitting with DMA (Master Mode)

16.5.2.3. Receive with DMA Function

The receiver of SPI master supports DMA function which can be enabled by setting **SPIx_CTRL.RX_DMA_EN** as 1. Then, the controller triggers SPI interface to receive data by enabling **SPIx_CTRL.MASTER_RX_TRIG**. Data is stored in SRAM and configured by DMA peripheral.

When the number of data transfer is reached, the DMA controller will generate an interrupt. Then, user disables the SPI controller by setting **SPIx_CTRL.SPI_EN** as 0.



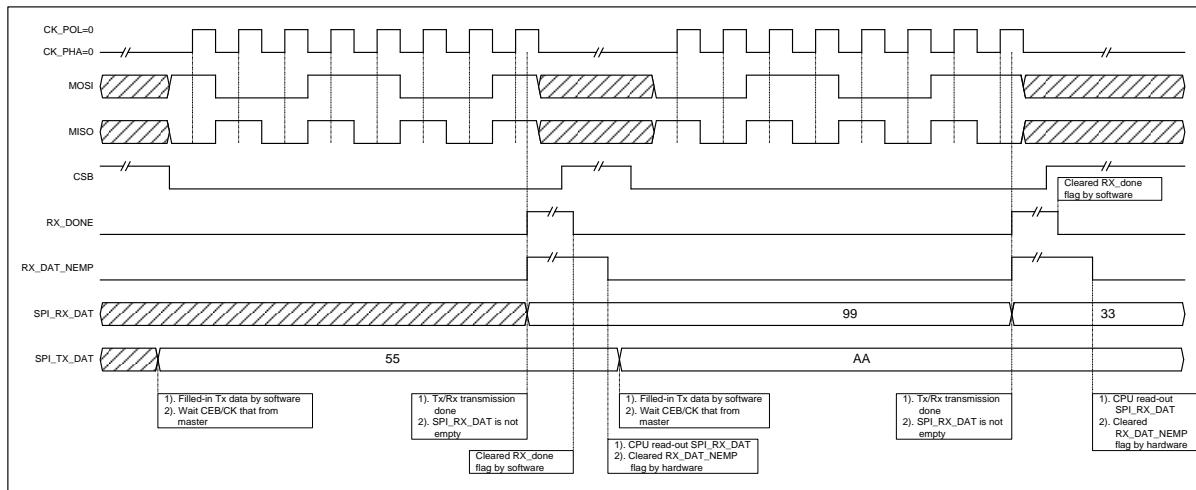
Receiving with DMA (Master Mode)

16.5.3. Slave Mode

16.5.3.1. Single Byte Transmission with Manual Mode

In slave mode, before using SPI function, user must set **SPIx_CTRL.MAUT_EN** as 1 and fill data in **SPIx_TX_DATA**. Then, SPI will send out data on MISO pin immediately when the controller receives SPI clock input that from the external master. In the same time, data received from MOSI pin will be saved in **SPIx_RX_DATA**.

During a SPI transmission, data is shifted out from most significant bit (MSB) on the MISO pin, and is shifted in from the MOSI pin. In this mode, the **SPIx_TX_DATA** register is a buffer between the system bus and the transmit shift register, and **SPIx_RX_DATA** register is a buffer between the system bus and the receive shift register. Every frame consists of 8-bit data bit. The **SPIx_STS.RX_DONE** will be set as 1 when 8 bits data is shifted in/out by MOSI/MISO pin. An interrupt is generated if the **SPIx_CTRL.RX_INT_EN** bit is set as 1. The **SPIx_STS.RX_DONE** flag is set as 1 by the hardware and cleared by the software. The following figure shows a single byte communication with transmitting

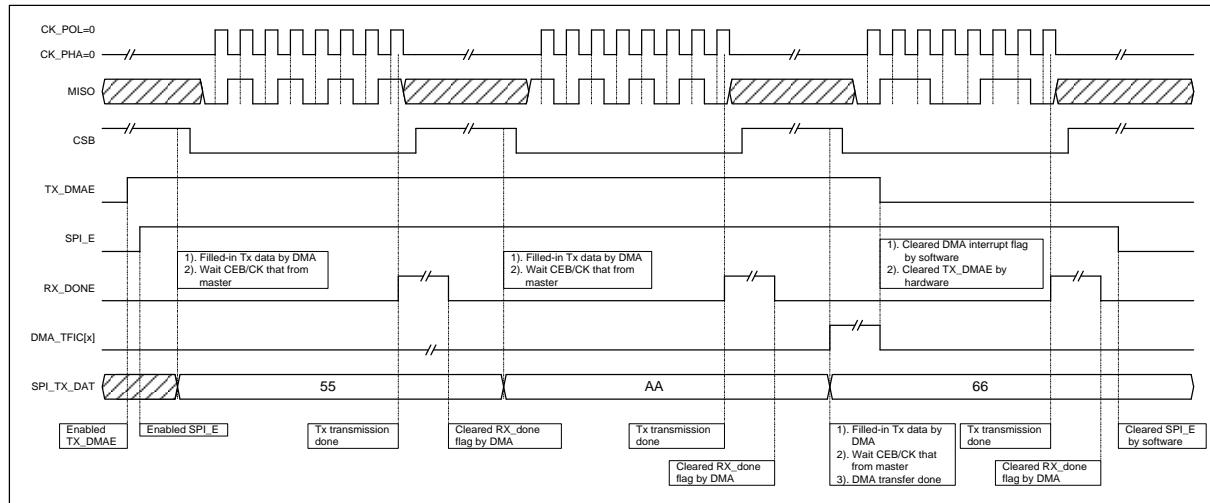


Single Byte Communication with Manual Mode (Slave Mode)

16.5.3.2. Transmit with DMA Function

The transmitter of SPI slave supports DMA mode which can be enabled by setting **SPIx_CTRL.TX_DMA_EN** as 1. Transmission data is loaded from SRAM and configured by DMA peripheral.

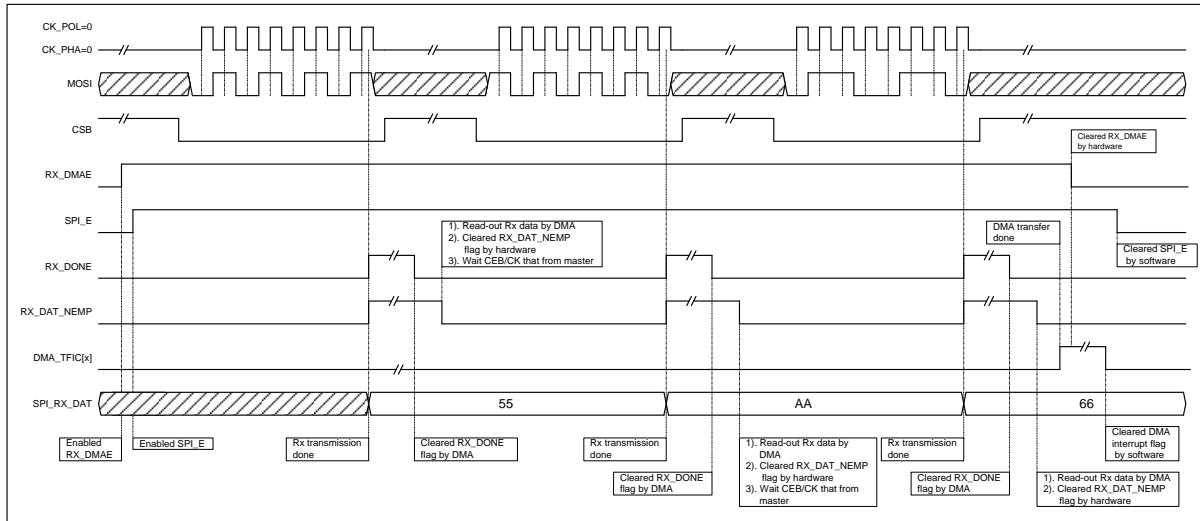
When the number of data transfer is reached, the DMA controller will generate an interrupt. Once all of the transfer data has been written to **SPIx_TX_DATA** by DMA, user can monitor **SPIx_STS.RX_DONE** flag to know whether the SPI communication is completed or not. To avoid receiving the last data incompletely, user can only disable the SPI by the software until **SPIx_STS.RX_DONE** = 1.



SPI Transmitting with DMA (Slave Mode)

16.5.3.3. Receive with DMA Function

The receiver of SPI slave supports DMA mode that can be enabled by setting **SPIx_CTRL.RX_DMA_EN** as 1. **SPIx_RX_DATA** starts to receive data from MOSI pin after **SPIx_CTRL.SPI_EN** = 1. The received data is stored in SRAM and configured by DMA peripheral. When the number of data transfer is reached, the DMA controller will generate an interrupt. Then, user disables the SPI controller by setting **SPIx_CTRL.SPI_EN** as 0.



SPI Receiving with DMA (Slave Mode)

16.5.3.4. Overrun Error

In slave mode, when a character is received and **SPIx_STS.RX_BUF_NEMPTY** has not been cleared, an overrun error will occur. The old data that is stored in **SPIx_RX_DATA** will be replaced by new data.

16.6. Register Description

16.6.1. Register map

Name	Address	Description
SPI0_CTRL	0x4009_0000	SPI 0 Control Register
SPI0_STS	0x4009_0004	SPI 0 Status Register
SPI0_TX_DATA	0x4009_0008	SPI 0 Transmit Data Register
SPI0_RX_DATA	0x4009_000C	SPI 0 Receive Data Register
SPI0_CLK_DIV	0x4009_0010	SPI 0 CLK DIV SEL
SPI1_CTRL	0x4009_1000	SPI 1 Control Register
SPI1_STS	0x4009_1004	SPI 1 Status Register
SPI1_TX_DATA	0x4009_1008	SPI 1 Transmit Data Register
SPI1_RX_DATA	0x4009_100C	SPI 1 Receive Data Register
SPI1_CLK_DIV	0x4009_1010	SPI 1 CLK DIV SEL

16.6.2. Register Function

SPI0_CTRL **0x4009_0000** **SPI 0 Control Register**

SPI1_CTRL **0x4009_1000** **SPI 1 Control Register**

Bit	31	30	29	28	27	26	25	24
Function	MASTER_RX_TRIG	TX_DISABLE			-			
Reset Value	0	0	1	1	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function		-			MOSI_DASE_L	MAUT_RW_EN	RX_DMA_EN	TX_DMA_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	CSB_SW_E_N		-		CSB_GPIO	ERR_INT_EN	RX_INT_EN	TX_INT_EN
Reset Value	0	0	0	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	SPICLK_CON TIUNE_EN	LOOPBACK_EN	CLK_POL	CLK_PHA	CSB_KEEP	MODE_SEL	SPI_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description			Condition
[31]	MASTER_RX_TRIGGER	R/W	SPI Master Rx Normal/DMA Trigger Note: In DMA mode, user has to write 1 to this bit so that DMA Rx can be enabled (that is, be triggered to move one data once) In Normal mode, user also has to write 1 to this bit in order to read one Rx data.			0: Disable 1: Enable (Auto Clear)
[30]	TX_DISABLE	R/W	Tx Mode Disable (For SPI Master Rx to disable Tx function.)			0: Tx Enable 1: Tx Disable

Bit	Function	Type	Description	Condition
[29:20]	-	R	Reserved	-
[19]	MOSI_DASEL	R/W	Select the source of the output data when DMA is used to receive data. Note: This bit is for master mode only.	0: Source from SPIx_RX_DATA. 1: Source from SPIx_TX_DATA.
[18]	MAUT_RW_EN	R/W	Manual Mode Enable Note: This bit is for master mode only.	0: Disabled 1: Enabled
[17]	RX_DMA_EN	R/W	Rx Buffer DMA Enable	0: Disabled 1: Enabled
[16]	TX_DMA_EN	R/W	Tx Buffer DMA Enable	0: Disabled 1: Enabled
[15]	CSB_SW_EN	R/W	SPI CSB Control by S/W (Bit[11] CSB_GPIO) Enable	0: Disabled 1: Enabled
[14:12]	-	R	Reserved	-
[11]	CSB_GPIO	R/W	SPI CSB Control Bit	0: Low 1: High
[10]	ERR_INT_EN	R/W	SPI Received Overrun Interrupt Enable	0: Disabled 1: Enabled
[9]	RX_INT_EN	R/W	SPI Received Complete Interrupt Enable	0: Disabled 1: Enabled
[8]	TX_INT_EN	R/W	SPI Transmitted Complete Interrupt Enable	0: Disabled 1: Enabled
[7]	-	R	Reserved	-
[6]	SPICLK_CONTIUNE_EN	R/W	Under the DMA's continuous transferring data mode, there is no delay between bytes.	0: Disabled 1: Enabled
[5]	LOOPBACK_EN	R/W	SPI Loop-Back Enable Note: The SPI Rx will connect to SPI Tx automatically when this bit is set as 1. This is a self-test function.	0: Disabled 1: Enabled
[4]	CLK_POL	R/W	SPI Clock Polarity	0: Data is captured on SPI clock's rising edge if CK_PHA = 0. 1: Data is captured on SPI clock's falling edge if CK_PHA = 0.
[3]	CLK_PHA	R/W	SPI Clock Phase	0: The 1st clock transition is the 1st data capture edge. 1: The 2nd clock transition is the 1st data capture edge.
[2]	CSB_KEEP_L	R/W	SPI CSB Keeping Low Control	0: SPI CSB is controlled by SPI controller. 1: SPI CSB will be kept in low state.

Bit	Function	Type	Description	Condition
[1]	MODE_SEL	R/W	SPI Operating Mode Selection	0: Master Mode 1: Slave Mode
[0]	SPI_EN	R/W	SPI Enabling Bit	0: Disable 1: Enable

SPI0_STS									SPI 0 Status Register								
SPI1_STS									SPI 1 Status Register								
Bit	31	30	29	28	27	26	25	24	Bit	23	22	21	20	19	18	17	16
Function					-				Function					-			
Reset Value	0	0	0	0	0	0	0	0	Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8									
Function					-				Function					-			
Reset Value	0	0	0	0	0	0	0	0									
Bit	7	6	5	4	3	2	1	0									
Function				RX_BUF_NE MPTY_FLAG	RX_OVER_R UN_FLAG	RX_DONE_F LAG	-	-	Function					TX_DONE_F LAG			
Reset Value	0	0	0	0	0	0	1	0									

Bit	Function	Type	Description	Condition
[31:5]	-	R	Reserved	-
[4]	RX_BUF_NEMPTY_FLAG	R	Receive Buffer Not Empty Flag	0: Rx buffer is empty. 1: Rx buffer is not empty.
[3]	RX_OVER_RUN_FLAG	R/W	SPI Received Overrun Flag Note: This flag is set as 1 when the Rx data is received but the previous Tx data has not yet been read from SPI_TXD.	Read 0: No overrun occurs. Read 1: Overrun occurs. Write 0: No effect. Write 1: Clear this bit.
[2]	RX_DONE_FLAG	R/W	SPI Receive Complete Flag (Slave Rx Mode Only) Note: When SPI0/1 is set as Mater Rx mode, checking RX_DONE Flag is invalid. User has to check TX_DONE Flag instead. Only when it's set as Slave Rx mode, internal Rx Counter will count, and checking RX_DONE will be valid.	Read 0: Receiving is not completed. Read 1: Receiving is completed. Write 0: No effect. Write 1: Clear this bit.
[1]	-	R	Reserved	-
[0]	TX_DONE_FLAG	R/W	SPI Transmission Complete Flag	Read 0: Transmission is not completed. Read 1: Transmission is completed. Write 0: No effect. Write 1: Clear this bit.

SPI0_TX_DATA **0x4009_0008** **SPI 0 Transmit Data Register**

SPI1_TX_DATA **0x4009_1008** **SPI 1 Transmit Data Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	SPI_TX_DATA[7:0]	R/W	The Transmit Data Register						-

SPI0_RX_DATA **0x4009_000C** **SPI 0 Receive Data Register**

SPI1_RX_DATA **0x4009_100C** **SPI 1 Receive Data Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	SPI_RX_DATA[7:0]	R	The Receive Data Register						-

SPI0_CLK_DIV								0x4009_0010	SPI 0 Clock Divider	
SPI1_CLK_DIV								0x4009_1010	SPI 1 Clock Divider	
Bit	31	30	29	28	27	26	25	24		
Function	-	-	-	-	-	-	-	-		
Reset Value	0	0	0	0	0	0	0	0		
Bit	23	22	21	20	19	18	17	16		
Function	-	-	-	-	-	-	-	-		
Reset Value	0	0	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8		
Function	-	-	-	-	-	-	PCLK_DIV[9:8]			
Reset Value	0	0	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0		
Function	PCLK_DIV[7:0]									
Reset Value	0	0	0	0	0	0	0	0		
Bit	Function	Type	Description					Condition		
[31:10]	-	R	Reserved					-		
[9:0]	PCLK_DIV[9:0]	R/W	SPI Clock Divide Selection					SPI Clock Frequency = System Clock/(PCLK_DIV[9:0] + 1)		

17. SPI controller for FLASH device access (SPIFC)

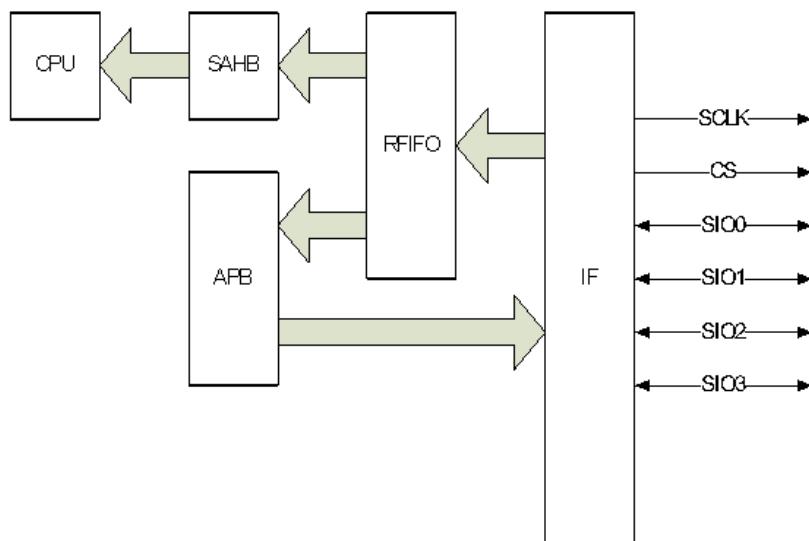
17.1. Introduction

The design provides an enhanced SPI (Serial Peripheral Interface) controller for FLASH device access. The interface includes a clock (SCLK), chip select (CS), and at most 4-bit parallel data pins. GPCM3 uses this communication Interface to read user's program and resources of the external memory. During the booting process, if any failure occurs, the boot loader will adjust the SPIFC IO configuration and try to run boot code process again; thus, after the boot code process is successfully executed, it will jump to the user's program. After that, the SPIFC IO configuration and Control register settings will not be in the default state but in auto mode. Since the user's program and resources are located in the external memory, SPIFC needs to be kept in auto mode so the CPU is allowed to execute external programs. When SPIFC manual mode is applied, the program must be moved to the SRAM. After jumping to the SRAM to execute the code, the SPIFC can be switched to the manual mode. Be sure to switch back to the auto mode before returning to the external program from the SRAM.

17.2. Features

- Support 1-bit, 2-bit and 4-bit data bus.
- Provide internal 8*FIFO (32-bit).
- Configurable Received Data Timing
- Configurable and Variable Packets Types
- Configurable Pin Definition

17.3. Block Diagram



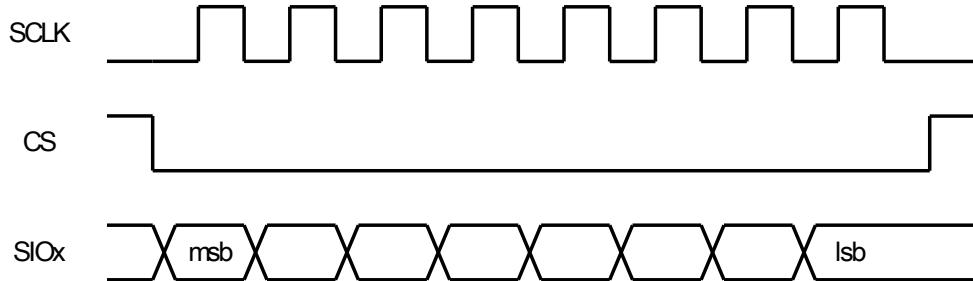
17.4. Function

17.4.1. Pin Descriptions

In GPCM3, except SCLK should be fixed as IOB1, other IOs' functions can be redefined through IOFUNC->SPIFC settings. For more details about these settings, please refer to the GPIO chapter. Without additional settings, the default pins are shown as follows:

Name	Default Pin	Description
SCLK	IOB1	Serial Clock
CS	IOB3	Chip Select for slave device.
SIO0(MOSI)	IOB4	Serial data output or serial data input/output for 2 x IO mode and 4 x IO mode.
SIO1(MISO)	IOB2	Serial data input or serial data input/output for 2 x IO mode and 4 x IO mode.
SIO2(WP)	IOB5	Serial data input/output for 4 x IO mode.
SIO3(HOLD)	IOB0	Serial data input/output for 4 x IO mode.

17.4.2. Timing Types



17.4.3. Packet Types

Several packet types are listed below, and these packet types could be composed by appropriate configuration.

COMMAND

COMMAND	ADDRESS
---------	---------

COMMAND	ADDRESS	TX	RX
---------	---------	----	----

COMMAND	ADDRESS	TX
---------	---------	----

COMMAND	ADDRESS	RX
---------	---------	----

COMMAND	ADDRESS	ENHAN	DUMMY	RX
---------	---------	-------	-------	----

ADDRESS	ENHAN	DUMMY	RX
---------	-------	-------	----

COMMAND	TX
---------	----

COMMAND	RX
---------	----

Examples of packet types are listed below.

- COMMAND:

SPIFC_CTRL0.OPERATION_MODE = 1b (Manual Mode)
SPIFC_CMD.SPICMD = User Defined
SPIFC_CMD.WITHOUTCMD_EN = 0b
SPIFC_CMD.ONLYCMD_EN = 1b
SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 0b
SPIFC_PARA.ENHAN_DIS = Don't Care
SPIFC_PARA.WITHOUT_ADDR_EN = Don't Care
SPIFC_PARA.ONLY_ADDR_EN = Don't Care
SPIFC_PARA.ENHANCE_BIT = Don't Care
SPIFC_PARA.DUMMY_CLK = Don't Care
SPIFC_RX_BC.RX_BC = Don't Care
SPIFC_TX_BC.TX_BC = Don't Care

Write SPIFC_CMD to initiate transaction.
Wait for SPIFC_CTRL0.PENDING_FLAG = 1.

- COMMAND + ADDRESS
 - SPIFC_CTRL0.OPERATION_MODE = 1b (Manual Mode)
 - SPIFC_CMD.SPICMD = User Defined
 - SPIFC_CMD.WITHOUTCMD_EN = 0b
 - SPIFC_CMD.ONLYCMD_EN = 0b
 - SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 0b
 - SPIFC_PARA.ENHAN_DIS = Don't Care
 - SPIFC_PARA.WITHOUT_ADDR_EN = 0b
 - SPIFC_PARA.ONLY_ADDR_EN = 1b
 - SPIFC_PARA.ENHANCE_BIT = Don't Care
 - SPIFC_PARA.DUMMY_CLK = User Defined
 - SPIFC_RX_BC.RX_BC = Don't Care
 - SPIFC_TX_BC.TX_BC = Don't Care

Write SPIFC_CMD to initiate transaction.
Wait for SPIFC_CTRL0.PENDING_FLAG = 1.

- COMMAND + ADDRESS + TX
 - SPIFC_CTRL0.OPERATION_MODE = 1b (Manual Mode)
 - SPIFC_CMD.SPICMD = User Defined
 - SPIFC_CMD.WITHOUTCMD_EN = 0b
 - SPIFC_CMD.ONLYCMD_EN = 0b
 - SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 0b
 - SPIFC_PARA.ENHAN_DIS = 1b
 - SPIFC_PARA.WITHOUT_ADDR_EN = 0b
 - SPIFC_PARA.ONLY_ADDR_EN = 0b
 - SPIFC_ADDR & SPIFC_ADDRH = Start address of transmitted data.
 - SPIFC_PARA.ENHANCE_BIT = Don't Care
 - SPIFC_PARA.DUMMY_CLK = User Defined
 - SPIFC_RX_BC.RX_BC = 0
 - SPIFC_TX_BC.TX_BC = Transmitted Byte Counts

Write SPIFC_CMD to initiate transaction.
Wait for SPIFC_CTRL0.PENDING_FLAG = 1.
Write SPIFC.TX_DATA and then wait for SPIFC_CTRL0.TX_DONE_FLAG. Repeat the above steps until Tx data transmission finishes.

- COMMAND + ADDRESS + RX
 - SPIFC_CTRL0.OPERATION_MODE = 1b (Manual Mode)
 - SPIFC_CMD.SPICMD = User Defined
 - SPIFC_CMD.WITHOUTCMD_EN = 0b
 - SPIFC_CMD.ONLYCMD_EN = 0b
 - SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 0b
 - SPIFC_PARA.ENHAN_DIS = 1b
 - SPIFC_PARA.WITHOUT_ADDR_EN = 0b
 - SPIFC_PARA.ONLY_ADDR_EN = 0b
 - SPIFC_ADDR & SPIFC_ADDRH = Start address of received data.
 - SPIFC_PARA.ENHANCE_BIT = Don't Care
 - SPIFC_PARA.DUMMY_CLK = User Defined
 - SPIFC_RX_BC.RX_BC = Received Byte Counts
 - SPIFC_TX_BC.TX_BC = 0

Write SPIFC_CMD to initiate transaction.
Wait SPIFC_CTRL0.PENDING_FLAG = 1.
Write SPIFC.RX_DATA and then wait for SPIFC_CTRL0.PENDING_FLAG. Read received data from SPIFC.RX_DATA and repeat the above steps until the expected byte counts are reached.

- COMMAND + ADDRESS + ENHAN + DUMMY + RX

- ADDRESS + ENHAN + DUMMY + RX
SPIFC_CTRL0.OPERATION_MODE = 0b (Auto Mode)
SPIFC_CMD.SPICMD = User Defined
SPIFC_CMD.WITHOUTCMD_EN = 0b
SPIFC_CMD.ONLYCMD_EN = 0b
SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 1b
SPIFC_PARA.ENHAN_DIS = 0b
SPIFC_PARA.WITHOUT_ADDR_EN = 0b
SPIFC_PARA.ONLY_ADDR_EN = 0b
SPIFC_PARA.ENHANCE_BIT = User Defined
SPIFC_PARA.DUMMY_CLK = User Defined
SPIFC_RX_BC.RX_BC = Received Byte Counts
SPIFC_TX_BC.TX_BC = 0

- COMMAND + TX
SPIFC_CTRL0.OPERATION_MODE = 1b (Manual Mode)
SPIFC_CMD.SPICMD = User Defined
SPIFC_CMD.WITHOUTCMD_EN = 0b
SPIFC_CMD.ONLYCMD_EN = 0b
SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 0b
SPIFC_PARA.ENHAN_DIS = 1b
SPIFC_PARA.WITHOUT_ADDR_EN = 1b
SPIFC_PARA.ONLY_ADDR_EN = 0b
SPIFC_PARA.ENHANCE_BIT = Don't Care
SPIFC_PARA.DUMMY_CLK = Don't Care
SPIFC_RX_BC.RX_BC = 0
SPIFC_TX_BC.TX_BC = Transmitted Byte Counts

Write SPIFC_CMD to initiate transaction.

Wait SPIFC_CTRL0.PENDING_FLAG = 1.

Write SPIFC.TX_DATA and then wait for SPIFC_CTRL0.TX_DONE_FLAG. Repeat the above steps until Tx data transmission finishes.

- COMMAND + RX
SPIFC_CTRL0.OPERATION_MODE = 1b (Manual Mode)
SPIFC_CMD.SPICMD = User Defined
SPIFC_CMD.WITHOUTCMD_EN = 0b
SPIFC_CMD.ONLYCMD_EN = 0b
SPIFC_CMD.1ST_TRANS_WITHCMD_EN = 0b
SPIFC_PARA.ENHAN_DIS = 1b
SPIFC_PARA.WITHOUT_ADDR_EN = 1b
SPIFC_PARA.ONLY_ADDR_EN = 0b
SPIFC_PARA.ENHANCE_BIT = Don't Care
SPIFC_PARA.DUMMY_CLK = Don't Care
SPIFC_RX_BC.RX_BC = Received Byte Counts
SPIFC_TX_BC.TX_BC = 0

Write SPIFC_CMD to initiate transaction.

Wait SPIFC_CTRL0.PENDING_FLAG = 1.

Write SPIFC.RX_DATA and then wait for SPIFC_CTRL0.PENDING_FLAG. Read received data from SPIFC.RX_DATA and repeat the above steps until the expected byte counts are reached.

17.5. Register Description

17.5.1. Register Map

Name	Address	Description
SPIFC_CTRL0	0x5006_0000	SPIFC Control Register0
SPIFC_CMD	0x5006_0004	SPIFC Command Register
SPIFC_PARA	0x5006_0008	SPIFC Parameter Register
SPIFC_ADDRL	0x5006_000C	SPIFC Address Low Register
SPIFC_ADDRH	0x5006_0010	SPIFC Address High Register
SPIFC_TX_DATA	0x5006_0014	SPIFC Write Data Register
SPIFC_RX_DATA	0x5006_0018	SPIFC Read Data Register
SPIFC_TX_BC	0x5006_001C	SPIFC Write Data Count Register
SPIFC_RX_BC	0x5006_0020	SPIFC Read Data Count Register
SPIFC_TIMING	0x5006_0024	SPIFC Timing Control Register
SPIFC_CTRL1	0x5006_002C	SPIFC Control Register 1
SPIFC_CTRL2	0x5006_0034	SPIFC Control Register 2
SPIFC_TX_DATA32	0x5006_0038	SPIFC 32-bit Tx Data Register
SPIFC_SCRAMBLE_CTRL	0x5006_0040	SPIFC Scramble Control Register

17.5.2. Register Function

0x5006_0000								SPIFC Control Register0	
Bit	31	30	29	28	27	26	25	24	
Function					-				
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16	
Function					-				
Reset Value	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	
Function	TX_DONE_FLAG	RX_FIFO_EMP_FLAG					IGNORE_LA_ST_CLK	OPERATION_MODE	
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
Function	CMIO[1:0]		AMIO[1:0]		MIO[1:0]		CLK_STATE	PENDING_FLAG	
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description				Condition
[31:16]	-	R	Reserved				-
[15]	TX_DONE_FLAG	R	This bit indicates whether the byte transmission is done.				0: Not yet. 1: Done
[14]	RX_FIFO_EMP_FLAG	R	This bit indicates whether the RX FIFO is empty.				0: FIFO is not empty. 1: FIFO is empty.

Bit	Function	Type	Description	Condition
[13:10]	-	R	Reserved	-
[9]	IGNORE_LAST_CLK	R/W	Ignore Last Clock The feature is only available when CLK_STATE = 0.	0: Not ignore clock. 1: Ignore clock.
[8]	OPERATION_MODE	R/W	SPIFC Operation Mode	0: Auto Mode 1: Manual Mode
[7:6]	CMIO[1:0]	R/W	IO bit width for command. Please refer to MIO for bit width definition.	00: 1-bit mode 01: 2-bit mode 10: 4-bit mode 11: Reserved
[5:4]	AMIO[1:0]	R/W	IO bit width definition for Address, enhancement and dummy data. Please refer to MIO for bit width definition.	00: 1-bit mode 01: 2-bit mode 10: 4-bit mode 11: Reserved
[3:2]	MIO[1:0]	R/W	IO bit width definition for TX and RX data.	00: 1-bit mode 01: 2-bit mode 10: 4-bit mode 11: Reserved
[1]	CLK_STATE	R/W	SPI Clock State Note: When SPI CLK pin is not in use, it has to be kept at low or high.	0: Low 1: High
[0]	PENDING_FLAG	R	This bit indicates whether SPI operation is done. Write 1 to clear this flag.	0: Not yet. 1: Done

0x5006_0004								SPIFC Command Register	
Bit	31	30	29	28	27	26	25	24	
Function									-
Reset Value									0
Bit	23	22	21	20	19	18	17	16	
Function									-
Reset Value									0
Bit	15	14	13	12	11	10	9	8	
Function			1ST_TRANS_WITHCMD_EN				ONLYCMD_EN	WITHOUTCMD_EN	
Reset Value			0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0	
Function	SPICMD[7:0]								
Reset Value									1
Bit	31	30	29	28	27	26	25	24	
Function									-
Reset Value									-

Bit	Function	Type	Description	Condition
[13]	1ST_TRANS_WITHCMD_EN	R/W	Only the first transaction has command bits.	0: Each transaction has command bits. 1: Only the first transaction has command bits.
[12:10]	-	R	Reserved	-
[9]	ONLYCMD_EN	R/W	A Transaction contains only command information.	0: Disabled 1: Only command information.
[8]	WITHOUTCMD_EN	R/W	Transaction without Command Bits Enable	0: Disabled 1: Without command bits.
[7:0]	SPICMD[7:0]	R/W	Command Bits in Each Transaction	-

0x5006_0008									SPIFC Parameter Register
Bit	31	30	29	28	27	26	25	24	
Function					-				
Reset Value	0	0	0	0	0	0	0	0	
Bit	23	22	21	20	19	18	17	16	
Function					-				
Reset Value	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	
Function	-	ENHAN_DIS	ONLY_ADDR_EN	WITHOUT_ADDR_EN			DUMMY_CLK[3:0]		
Reset Value	0	1	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
Function	ENHANCE_BIT[7:0]								
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:15]	-	R	Reserved	-
[14]	ENHAN_DIS	R/W	Transaction doesn't contain enhancement bits.	0: Transaction contains enhance bits. 1: Transaction doesn't contain enhance bits.
[13]	ONLY_ADDR_EN	R/W	Transaction only contains address bits.	0: Disabled 1: Transaction only contains address bits.
[12]	WITHOUT_ADDR_EN	R/W	Transaction doesn't contain address bits.	0: Disabled 1: Transaction doesn't contain address bits.

Bit	Function	Type	Description					Condition
[11:8]	DUMMY_CLK[3:0]	R/W	Dummy Clock Cycle					0x0 = 0 dummy cycle. 0x1= 1 dummy cycles. 0x2 = 2 dummy cycles. ... 0x9 = 9 dummy cycles. 0xA~F = Reserved
[7:0]	ENHANCE_BIT[7:0]	R/W	Enhancement bits in a transaction. Please refer to vendor's spec. for more descriptions.					-

SPIFC_ADDR								
0x5006_000C								
Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	SPIFC_ADDR_L[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	SPIFC_ADDR_L[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	SPIFC_ADDR_L[15:0]	R/W	The first 16 bits of SPI FLASH address. The function is valid when SPIFC is switched to manual mode. (Not for Auto mode.)					-

SPIFC_ADDRH								
0x5006_0010								
Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	SPIFC_ADDR_H[15:8]							
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	SPIFC_ADDR_H[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	SPIFC_ADDR_H[15:0]	R/W	The last 16 bits of 32 SPI FLASH address. The function is valid when SPIFC is switched to manual mode. (Not for Auto mode.)					-

SPIFC_TX_DATA **0x5006_0014** **SPIFC Write Data Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					TX_DATA[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	TX_DATA[7:0]	R/W	8-bit Write Data Make sure the controller is idle before writing the registers.					-

SPIFC_RX_DATA **0x5006_0018** **SPIFC Read Data Register**

Bit	31	30	29	28	27	26	25	24
Function					RX_DATA[31:24]			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					RX_DATA[23:16]			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					RX_DATA[15:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					RX_DATA[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	RX_DATA[31:0]	R	Obtain 4 bytes data from SPI FLASH.					-

SPIFC_TX_BC
0x5006_001C
SPIFC Write Data Count Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					TX_BC[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:9]	-	R	Reserved						-
[8:0]	TX_BC[8:0]	R/W	Transmitted Byte Counts of A Transaction						-

SPIFC_RX_BC
0x5006_0020
SPIFC Read Data Count Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					RX_BC[7:0]			
Reset Value	0	0	1	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:9]	-	R	Reserved						-
[8:0]	RX_BC[8:0]	R/W	Received Byte Counts of A Transaction If OPERATION_MODE is cleared, the bit[0] and [1] should be 0. Note: The unit of Auto mode is Word (1 word = 4-byte), so Bit[1:0] = 00.						-

SPIFC_TIMING
0x5006_0024
SPIFC Timing Control Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				SLCH[0]	-		CS_HI_CNT[2:0]	
Reset Value	0	0	0	1	0	1	1	1

Bit	7	6	5	4	3	2	1	0
Function			SMP_DELAY[3:0]		-		SMP_CLK_EDGE_SEL[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:13]	-	R	Reserved	-
[12]	SLCH[0]	R/W	Time setting from CS being pulled Low to the first CLK signal. Note: This function is valid only when SPIFC CTRL2[9:8]: SPIFC CLK Divide = 1 (0x50060034).	0: 0.5 SPIFC CLK 1: 1.5 SPIFC CLK
[11]	-	R	Reserved	-
[10:8]	CS_HI_CNT[2:0]	R/W	Note: It determines the time interval will be kept for how long. (CS pin is kept high for power saving.) The interval is between a piece of data is read (CS pin is pulled high and SPI Flash is disabled) and the next data to be read.	Duration of CS being kept high = (CS_HI_CNT +1)* SPIFC Clock
[7:4]	SMP_DELAY[3:0]	R/W	Sample Clock Delay Before Receiving Data	0: No delay. 1: Delay 1 delay_cell. ... N: Delay N delay_cell. Note: 1 Cell Time = 1ns
[3]	-	R	Reserved	-
[2:0]	SMP_CLK_EDGE_SEL[2:0]	R/W	Sample Timing for Received Data	0: Sample at the 1st clock edge. 1: Sample at the 2nd clock edge. ... N: Sample at the (N+1)th clock edge.

0x5006_002C								SPIFC Control Register 1	
Bit	31	30	29	28	27	26	25	24	
Function									-
Reset Value	0	0	0	0	0	0	0	0	
Bit	23	22	21	20	19	18	17	16	
Function									-
Reset Value	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	
Function	4BYTES_ADDR_EN[7:0]								
Reset Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
Function									SPIFC_EN
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:8]	4BYTES_ADDR_EN[7:0]	R/W	4 bytes Address Mode Enable Note: If SPI FLASH size exceeds 128 Mb, it is necessary to issue a 4-byte address mode to make bit[7:0] all "1". Once the control bits are not 0, the 32-bit addressing will be activated.	Bit[0]: Control addr[24] Bit[1]: Control addr[25] Bit[2]: Control addr[26] Bit[3]: Control addr[27] Bit[4]: Control addr[28] Bit[5]: Control addr[29] Bit[6]: Control addr[30] Bit[7]: Control addr[31] 0: Disable 1: Enable
[7:1]	-	R	Reserved	-
[0]	SPIFC_EN	R/W	SPIFC Controller Enabling Bit	0: Disable 1: Enable

0x5006_0034								SPIFC Control Register 2	
Bit	31	30	29	28	27	26	25	24	
Function									-
Reset Value	0	0	0	0	0	0	0	0	
Bit	23	22	21	20	19	18	17	16	
Function									-
Reset Value	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	
Function					CLK_ASYN				
Reset Value	0	0	0	1	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Function	CS_HOLD_CNT[7:0]							
Reset Value	1	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:13]	-	R	Reserved					-
[12]	CLK_ASYN	R/W	Asynchronous Clock Setting					0: Synchronous Clock (SPIFC clock equals the system clock.) 1: Asynchronous Clock (SPIFC clock is different from the system clock.)
[11:10]	-	R	Reserved					-
[9:8]	CLK_SEL[1:0]	R/W	SPI-FLASH Clock Divide The SPI_FLASH clock is divided by the controller SPI source clock.					00: Divided by 1. (SPI_FLASH Clock = Source Clock) 01: Divided by 2. 10: Divided by 3. 11: Divided by 4
[7:0]	CS_HOLD_CNT[7:0]	R/W	SPI-FLASH CS# Hold Cycle After the SPI_FLASH read transaction is done, the controller will keep the CS# in low state till the CS_HOLD_CNT count is done or the next non-continuous address access. If the CS_HOLD_CNT = 0xff, the CS# will keep low till the next non-continuous address access.					-

0x5006_0038								SPIFC 32-bit Tx Data Register
Bit	31	30	29	28	27	26	25	24
Function	TX_DATA[31:24]							
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function			TX_DATA[23:16]					
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function			TX_DATA[15:8]					
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function			TX_DATA[7:0]					
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:0]	TX_DATA[31:0]	R/W	32-bit Write Data Make sure the controller is idle before writing the registers.					-

SPIFC_SCRAMBLE_CTRL
0x5006_0040
SPIFC Scramble Control Register

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

Bit	Function	Type	Description	Condition
[31:2]	-	R	Reserved	-
[1]	ENC_EN	R/W	In manual mode, enable or disable encryption function when SPIFC is writing data.	0: Disabled 1: Enabled
[0]	DEC_EN	R/W	In manual mode, enable or disable decryption function when SPIFC is reading data.	0: Disabled 1: Enabled

18. Universal Asynchronous Receiver Transmitter (UART)

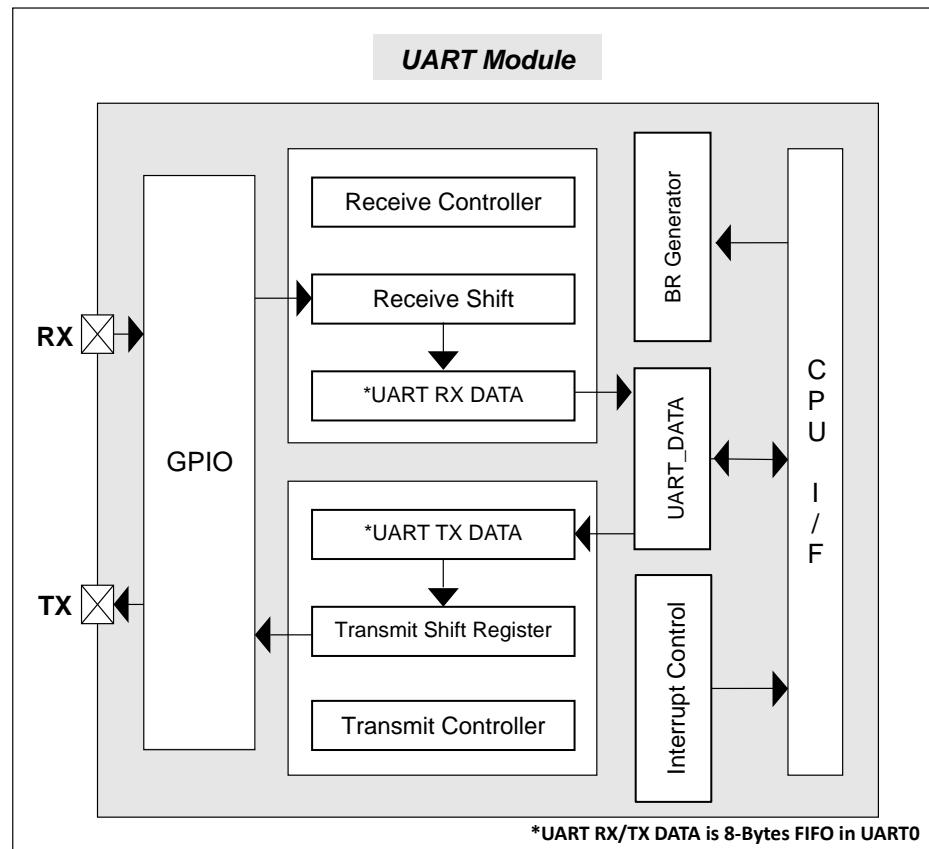
18.1. Introduction

The GPCM3 incorporates two universal asynchronous receiver transmitters (UART). The asynchronous serial channel includes the reception and the transmission frames. The receiver and transmitter are independent so frames can start at different points in time for transmission and reception. The UART provides a wide range of baud rates by baud rate generator. It also supports half-duplex single wire communication.

18.2. Features

- Half duplex, Tx and Rx Asynchronous Communications
- Single Wire Half Duplex Communication
- Wide Range Baud Rate Generator
- Programmable Data Word Length (8 or 9 bits)
- Configurable stop bits (1 or 2 stop bits).
- Separate enable bits for Transmitter and Receiver.
- DMA request generation during regular channel conversion (half duplex communication only).
- H/W Baud Rate Compensate Mechanism
- Parity Control
- 8-Bytes depth FIFO in both Tx and Rx direction (UART0 only).

18.3. Block Diagram

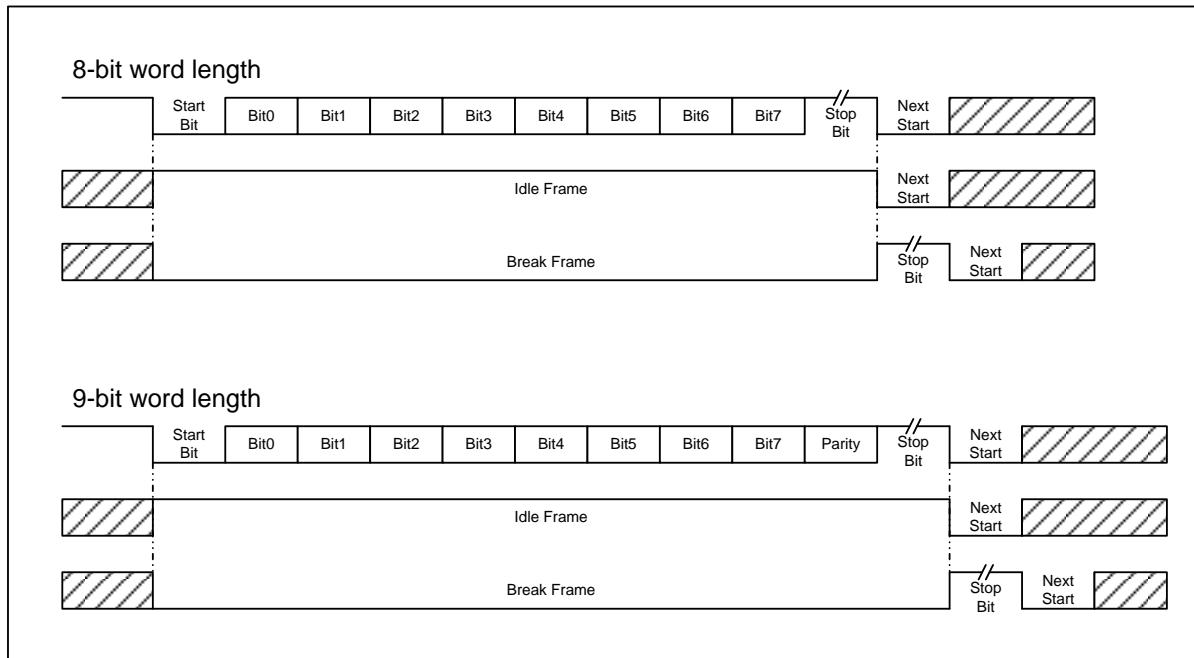


18.4. Function

18.4.1. Frame Format

A standard UART frame is shown in the following figure, and it consists of the features listed below.

- An idle character with the signal level 1.
- One start bit with the signal level 0.
- A data frame.
- A parity bit, which is programmable for either even or odd parity. In addition, it is possible to handle frames without parity bit.
- One or two stop bits with the signal level 1.



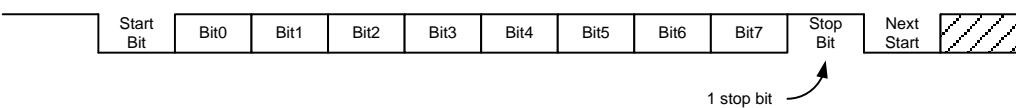
Frame Format of UART

18.4.2. Transmitter

The transmitter can send data words of either 8 or 9 bits according to the **UART_CTRL.PARITY_BIT_EN** bit status. Before using UART transmitter function, user must set **UART_CTRL.UART_EN** and **UART_CTRL.UART_TX_EN** as 1. Then fill the data in **UART_DATA**. When the baud-rate count overflows, the data in the transmit shift register will be output to the **UART_TX** pin.

During a UART transmission, the data will be shifted out from least significant bit (LSB) on the **UART_TX** pin first. In this mode, the **UART_DATA** register is a buffer between the system bus and the transmit shift register. Every character consists of a start bit which is at logic level low for one bit period. The character is terminated by a configurable number of stop bit. The number of stop bit can be configured in **UART_CTRL.STOP_SEL**, and the new start bit can be transferred directly after the last stop bit.

8-bit word length with 1 stop bit



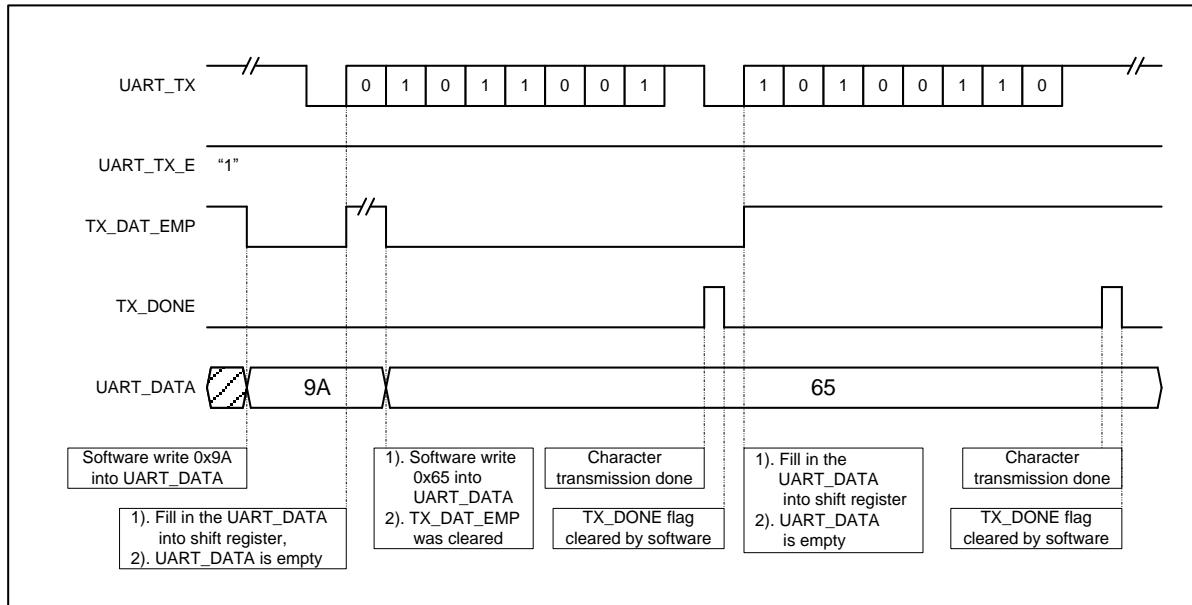
8-bit word length with 2 stop bits



Configurable Stop Bit

18.4.3. Single Byte Transmission

After **UART_CTRL.UART_EN** and **UART_CTRL.UART_TX_EN** are set as 1, UART will send out data immediately when the host (CPU or DMA) fills data in **UART_DATA**. The **UART_STS.TX_DAT_EMP** bit is always cleared by a write to the **UART_DATA** register. This bit is set as 1 by the hardware; it indicates the data has been moved from **UART_DATA** to the shift register and the data transmission has started. At this time, the **UART_DATA** register is empty. It allows user to write in the next data to **UART_DATA** register. In addition, **UART_STS.TX_DAT_EMP** can trigger an interrupt to CPU if **UART_CTRL.TX_INT_EN** is enabled.

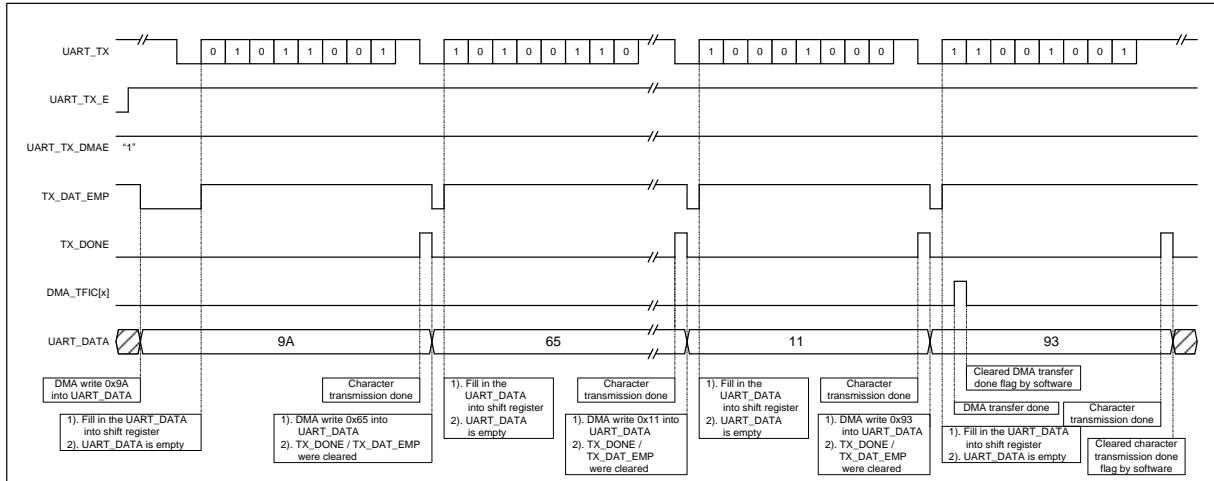


Single Byte Communication with Transmitting

18.4.4. Transmit with DMA Function

In GPCM3, UART supports DMA mode which can be enabled by setting **UART_CTRL.TX_DMAE** as 1. Data is loaded from SRAM and configured by DMA peripheral.

When the number of data transfers is reached, the DMA controller will generate an interrupt. Once all of the transfer data has been written to **UART_DATA** by DMA, user can monitor **UART_STS.TX_DONE** flag to know whether the UART communication is completed or not. User can only disable UART by the software until **UART_STS.TX_DONE** = 1. This avoids the incomplete transmission of the last data.



UART Transmitting with DMA

18.4.5. Receiver

The receiver can receive data words of either 8 or 9 bits according to the **UART_CTRL.PARITY_BIT_EN** bit status.

18.4.6. Baud Rate Generation

The UART provides a wide range of baud rates by baud rate generator. In order to avoid the transmission error caused by the error of clock, the hardware supports a mechanism of clock compensation.

For example:

PCLK = 16MHz, Baud Rate = 115200

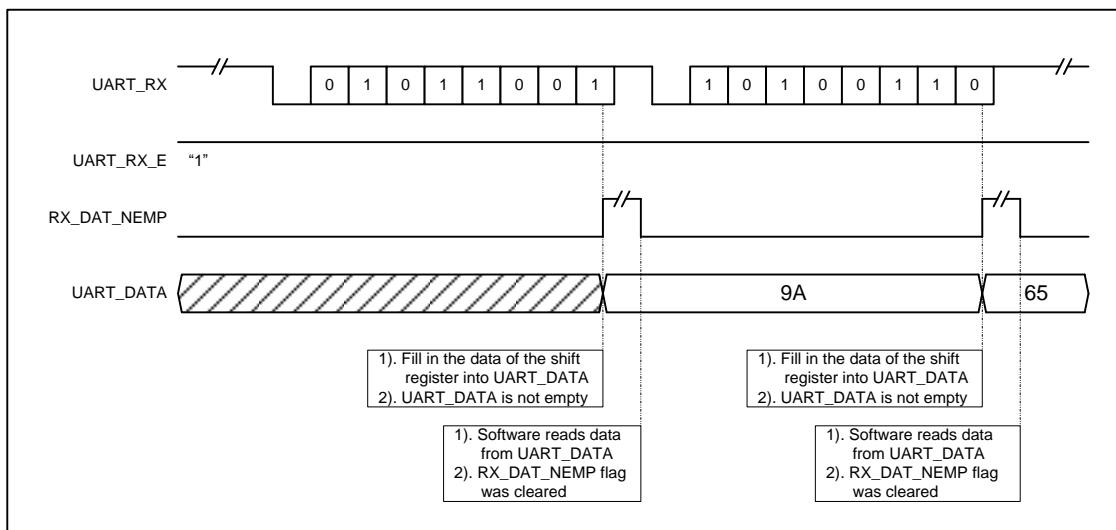
So, **UART_BAUD_RATE.BR_DIV** = $16M / (16 * 115200) = 8.6805555$

Choosing **BR_DIV** = 9 means the baud rate of the receiver is slower than the transmitter, and the difference between 8.6805555 and 9 is 0.3194445. 1 divide by 0.3194445 is 3.13. That means user must compensate baud rate generator every 3 baud rate count cycles. This mechanism reduces the cumulative errors.

18.4.7. Single Byte Receive

After **UART_CTRL.UART_EN** and **UART_CTRL.UART_RX_EN** are set as 1, data shifts in through the **UART_RX** pin. The **UART_STS.RX_DAT_NEMP** bit is always cleared by a read from the **UART_DATA** register. This bit is set as 1 by the hardware; it indicates the data has been moved from shift register to **UART_DATA** and the data transmission has finished. At this time, the **UART_DATA** register is not empty. It allows user to know it's able to read out data from **UART_DATA** register.

In addition, **UART_STS.RX_DAT_NEMP_FLAG** can trigger an interrupt to CPU if **UART_CTRL.RX_INT_EN** is enabled.

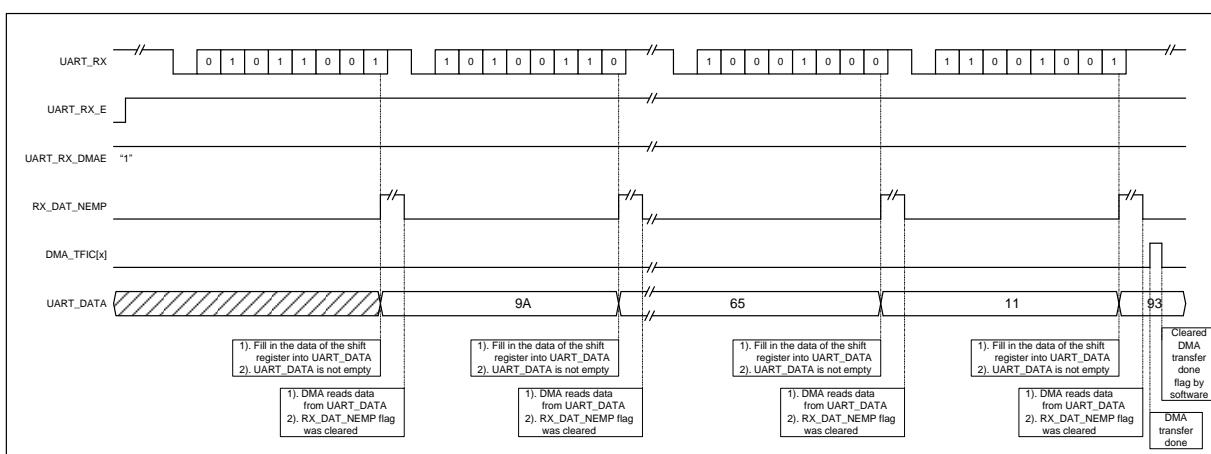


Single Byte Communication with Receiving

18.4.8. Receive with DMA Function

In GPCM3, UART supports DMA mode which can be enabled by setting **UART_CTRL.TX_DMAE** as 1. Data is stored in SRAM and configured by DMA peripheral.

When the number of data transfers is reached, the DMA controller generates an interrupt. Once all of the transfer data has written to SRAM by DMA, user can monitor **UART_STS.TX_DONE** flag to know whether the UART communication is complete or not.



UART Receiving with DMA

18.4.9. Break Frame

A break condition occurs when the receiver input is at the low level over a character frame. This is an unnecessary error. During data transmission, when signaling rates are mismatched, no meaningful characters can be sent. A long break signal can be used to notify a mismatched receiver. When a break frame is detected, the receiver will generate an interrupt if the **UART_CTRL.RX_BRK_INT_EN** bit is set as 1.

18.4.10. Idle Frame

An idle condition occurs when the receiver input is at the high level over a character frame. The procedure of idle frame is the same as normal data receive. When an idle frame is detected, the receiver will generate an interrupt if the **UART_CTRL.RX_IDLE_INT_EN** bit is set as 1.

18.4.11. Overrun Error

When a character is received and **UART_STS.RX_DAT_NEMP** has not been cleared, an overrun error will occur. The old data which is stored in **UART_DATA** will be replaced by the new data.

18.4.12. Parity Check

Parity check can be enabled by setting the **UART_CTRL.PARITY_CHK_EN** as 1. The even/odd parity type is selected by configuring **UART_CTRL.PARITY_SEL**. With even parity, the receiver will receive an even number of 1. With odd parity, the receiver will receive an odd number of 1. (The unit is frame.) User is able to know whether the received data is correct by calculation.

18.5. Register Description

Register Map

Name	Address	Description
UART0_CTRL	0x400A0000	UART0 Control Register
UART0_STS	0x400A0004	UART0 Status Register
UART0_BAUD_RATE	0x400A0008	UART0 Baud Rate Register
UART0_DATA	0x400A000C	UART0 Data Register
UART0_FIFO	0x400A0010	UART0 FIFO Control Register
UART1_CTRL	0x400A1000	UART1 Control Register
UART1_STS	0x400A1004	UART1 Status Register
UART1_BAUD_RATE	0x400A1008	UART1 Baud Rate Register
UART1_DATA	0x400A100C	UART1 Data Register

Register Function

UART0_CTRL **0x400A0000** **UART0 Control Register**

UART1_CTRL **0x400A1000** **UART1 Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-		RX_DMAE	TX_DMAE	LOOPBACK_EN	STOP_SEL	SEND_IDLE	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	RX_BRK_IN T_EN	RX_IDLE_IN T_EN	RX_INT_EN	TX_INT_EN	PARITY_SEL	PARITY_CHK _EN	PARITY_BIT _EN	UART_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	FIFO_EN			-			UART_RX_E N	UART_TX_E N
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:22]	-	-	Reserved	-
[21]	RX_DMAE	R/W	Rx Buffer DMA Enable	0: Disabled 1: Enabled
[20]	TX_DMAE	R/W	Tx Buffer DMA Enable	0: Disabled 1: Enabled
[19]	LOOPBACK_EN	R/W	UART Loop-Back Enable	0: Disabled 1: Enabled
[18]	STOP_SEL	R/W	STOP Bit Selection	0: 1 Stop bit 1: 2 Stop bits
[17]	SEND_IDLE	R/W	UART Send Idle Enable	0: Disabled 1: Enabled
[16]	-	-	Reserved	-
[15]	RX_BRK_INT_EN	R/W	UART Received Break Interrupt Enable	0: Disabled 1: Enabled
[14]	RX_IDLE_INT_EN	R/W	UART Detected Idle Interrupt Enable	0: Disabled 1: Enabled
[13]	RX_INT_EN	R/W	UART Received A Word Interrupt Enable	0: Disabled 1: Enabled
[12]	TX_INT_EN	R/W	UART Transmission Done Interrupt Enable	0: Disabled 1: Enabled
[11]	PARITY_SEL	R/W	Parity Selection	0: Even Parity 1: Odd Parity
[10]	PARITY_CHK_EN	R/W	Parity Check Enable	0: Disabled 1: Enabled
[9]	PARITY_BIT_EN	R/W	Parity Bit Enable Note: The parity bits of PARITY_BIT_EN and PARITY_CHK_EN must both be set as 1.	0: Disabled 1: Enabled
[8]	UART_EN	R/W	UART Enable	0: Disabled 1: Enabled
[7]	FIFO_EN	R/W	UART0 FIFO Enable (UART0 Only)	0: Disabled 1: Enabled
[6:2]	-	-	Reserved	-
[1]	UART_RX_EN	R/W	UART Receiver Enable	0: Disabled 1: Enabled
[0]	UART_TX_EN	R/W	UART Transmitter Enable	0: Disabled 1: Enabled

UART0_STS								0x400A0004								UART0 Status Register								
UART1_STS								0x400A1004								UART1 Status Register								
Bit	31	30	29	28	27	26	25	24	Bit	23	22	21	20	19	18	17	16							
Function	-	-	-	-	-	-	-	-	Function	RX_INT	TX_INT	-	-	TX_FIFO_E_MPTY	RX_FIFO_FU_LL	TX_FIFO_FU_LL	RX_FIFO_E_MPTY	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0	Reset Value	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8	Bit	7	6	5	4	3	2	1	0							
Function	RX_BRK	RX_IDLE	RX_DAT_NEMP	TX_DONE	-	-	-	RX_STP_ER_R	Function	RX_PARITY_ERR	RX_PRITY	RX_OV_RU_N	-	-	-	-	-	TX_DAT_EM_P	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0	Reset Value	0	0	0	0	0	0	1	0	-	-	-	-	-	-	-

Bit	Function	Type	Description								Condition
[31:24]	-	-	Reserved								-
[23]	RX_INT	R	UART RX Interrupt Flag (UART0 Only) (a) FIFO enabled (8-bytes depth): This bit is set as 1 by the hardware when the received FIFO level is more than or equal to RX FIFO level. (b) FIFO disabled (1-byte depth): Please refer to UART0_STS[13] to check whether there's a Rx INT flag.								FIFO Enabled: 0: No. in RX FIFO < RX FIFO Level Setting 1: No. in RX FIFO >= RX FIFO Level Setting FIFO Disabled: 0: RX data is not ready. 1: RX data is ready.
[22]	TX_INT	R	UART TX Interrupt Flag (UART0 Only) (a) FIFO enabled (8-byte depth): This bit is set as 1 by the hardware when the transmitting FIFO is less than or equal to TX FIFO level. (b) FIFO disabled (1-byte depth): The same as UART0_STS[12].								FIFO Enabled: 0: No. in TX FIFO > TX FIFO Level Setting 1: No. in TX FIFO <= TX FIFO Level setting FIFO Disabled: 0: TX buffer is not ready. 1: TX buffer is ready.
[21:20]	-	-	Reserved								-
[19]	TX_FIFO_EMPTY	R	UART TX FIFO Empty Flag (UART0 Only) (a) FIFO enabled (8-byte depth): This bit is set as 1 by the hardware when the transmitting FIFO is empty. (b) FIFO disabled (1-byte depth): This bit is set as 1 by the hardware when the transmitting hold register is empty.								FIFO Enabled: 0: No. in TX FIFO > 0 1: No. in TX FIFO = 0 FIFO Disabled: 0: TX buffer is not empty. 1: TX buffer is empty.

Bit	Function	Type	Description	Condition
[18]	RX_FIFO_FULL	R	UART RX FIFO Full Flag (UART0 Only) (a) FIFO enabled (8-byte depth): This bit is set as 1 by the hardware when the receive FIFO is full. (b) FIFO disabled (1-byte depth): This bit is set as 1 by the hardware when the receiving hold register is full.	FIFO Enabled: 0: No. in RX FIFO < 8 1: No. in RX FIFO = 8 FIFO Disabled: 0: RX buffer is not full. 1: RX buffer is full.
[17]	TX_FIFO_FULL	R	UART TX FIFO Full Flag (UART0 Only) (a) FIFO enabled (8-byte depth): This bit is set as 1 by the hardware when the transmitting FIFO is full. (b) FIFO disabled (1-byte depth): This bit is set as 1 by the hardware when the transmitting hold register is full.	FIFO Enabled: 0: No. in TX FIFO < 8 1: No. in TX FIFO = 8 FIFO Disabled: 0: TX buffer is not full. 1: TX buffer is full.
[16]	RX_FIFO_EMPTY	R	UART RX FIFO Empty Flag (UART0 Only) (a) FIFO enabled (8-byte depth): This bit is set as 1 by the hardware when the receiving FIFO is empty. (b) FIFO disabled (1-byte depth): This bit is set as 1 by the hardware when the receiving hold register is empty.	FIFO Enabled: 0: No. in RX FIFO > 0 1: No. in RX FIFO = 0 FIFO Disabled: 0: RX buffer is not empty. 1: RX buffer is empty.
[15]	RX_BRK	R/W	UART Receiver Break Detected Flag Note: This flag is used in receiver mode only.	Read 0: No break character is detected. Read 1: Break character is detected. Write 0: No effect. Write 1: Clear this bit.
[14]	RX_IDLE	R/W	UART Rx Idle Line Detected Flag Note: This flag is used in receiver mode only.	Read 0: No Idle Line is detected. Read 1: Idle Line is detected. Write 0: No effect. Write 1: Clear this bit.
[13]	RX_DAT_NEMP	R/W	UART Data Register Not Empty Flag Note: This bit will be cleared as 0 automatically when the CPU or DMA reads UART_DATA .	0: Not ready 1: Ready
[12]	TX_DONE	R/W	UART Transmission Complete Flag	Read 0: Transmission is not completed. Read 1: Transmission is completed. Write 0: No effect. Write 1: Clear this bit.
[11:9]	-	-	Reserved	-

Bit	Function	Type	Description	Condition
[8]	RX_STP_ERR	R/W	UART Receiver Stop Bit Error Flag UART Receiver Stop Bit Error Flag	Read 0: No stop bit error. Read 1: Stop Bit Error Write 0: No effect. Write 1: Clear this bit.
[7]	RX_PARITY_ERR	R/W	UART Receiver Parity Error Flag	Read 0: No parity error. Read 1: Parity Error Write 0: No effect. Write 1: Clear this bit.
[6]	RX_PRITY	R	UART Receiver Parity Bit Note: This bit is available when UART_CTRL.PARITY_BIT_EN and UART_CTRL.PARITY_CHK_EN are set as 1.	-
[5]	RX_OV_RUN	R/W	UART Over Run Error Flag Note: This flag is used in receiver mode only. The content of UART_DATA will be overwritten by the new data when this bit is set as 1.	Read 0: No overrun error. Read 1: Overrun error is detected. Write 0: No effect. Write 1: Clear this bit.
[4:2]	-	-	Reserved	-
[1]	TX_DAT_EMP	R/W	UART Transmit Data Register Empty Flag This bit is set as 1 when the data is transferred to the shift register (8-bit buffer). It acknowledges user to write the next Tx data into Transmitted data register. This bit is automatically cleared as 0 when the CPU or DMA writes UART_DATA .	0: TX buffer is not empty. 1: TX buffer is empty.
[0]	-	-	Reserved	-

UART0_BAUD_RATE
0x400A0008
UART0 Baud Rate Register
UART1_BAUD_RATE
0x400A1008
UART1 Baud Rate Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-			BR_CMP_SEL	BR_CMP_CYCLE[3:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-				BR_DIV[11:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	BR_DIV[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:21]	-	-	Reserved					-
[20]	BR_CMP_SEL	R/W	Baud Rate Compensate Selection					0: Add Counting Cycle 1: Sub Counting Cycle
[19:16]	BR_CMP_CYCLE[3:0]	R/W	Baud Rate Compensate Cycles Example: PCLK = 16MHz, and Baud Rate = 115200. So, BR_DIV = 16M/(16 * 115200) = 8.6805555. Choosing BR_DIV = 9 means the baud rate of the receiver will be slower than the transmitter. The difference between 8.6805555 and 9 is 0.3194445. 1 divide 0.3194445 is 3.13. That means user must compensate baud rate generator every 3 baud rate count cycle. This mechanism reduces the cumulative errors.					-
[15:12]	-	-	Reserved					-
[11:0]	BR_DIV[11:0]	R/W	Baud Rate Divider Tx/Rx Baud Rate = PCLK/(16 * BR_DIV[11:0]) BR_DIV[11:0] = PCLK/(16 * Baud Rate)					-

UART0_DATA **UART0 Data Register**
0x400A000C

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	-	Reserved					-
[7:0]	DATA[7:0]	R/W	The Received or Transmitted Data Register					-



GPCM3 SERIES PROGRAMMING GUIDE

Bit	Function	Type	Description	Condition
[31:15]	-	-	Reserved	-
[14:12]	TX_LEVEL[2:0]	R/W	<p>Transmit FIFO Interrupt Level Register</p> <p>This register indicates how many empty slots are required when an interrupt is issued. The smaller the value is configured, the lower interrupt penalty it will be since user can write more data in one interrupt.</p>	000: No. in TX FIFO < 1, 8 writes are allowed. 001: No. in TX FIFO < 2, 7 writes are allowed. 010: No. in TX FIFO < 3, 6 writes are allowed. 011: No. in TX FIFO < 4, 5 writes are allowed. 100: No. in TX FIFO < 5, 4 writes are allowed. 101: No. in TX FIFO < 6, 3 writes are allowed. 110: No. in TX FIFO < 7, 2 writes are allowed. 111: No. in TX FIFO < 8, 1 write is allowed.
[11]	-	-	Reserved	-
[10:8]	TX_FLAG[2:0]	R	<p>Transmit FIFO Data Level</p> <p>This register indicates how many data remains in transmit FIFO.</p>	000: 0 byte in FIFO. 001: 1 byte in FIFO. 010: 2 byte in FIFO. 011: 3 byte in FIFO. 100: 4 byte in FIFO. 101: 5 byte in FIFO. 110: 6 byte in FIFO. 111: 7 byte in FIFO.
[7]	-	-	Reserved	-

Bit	Function	Type	Description	Condition
[6:4]	RX_LEVEL[2:0]	R/W	<p>Receive FIFO Interrupt Level Register</p> <p>This register indicates how many byte is stored in receive FIFO when an interrupt is issued. The larger the value is configured, the lower interrupt penalty it will be since user can read more data in one interrupt.</p>	000: No. in RX FIFO \geq 1, 1 read is allowed. 001: No. in RX FIFO \geq 2, 2 reads are allowed. 010: No. in RX FIFO \geq 3, 3 reads are allowed. 011: No. in RX FIFO \geq 4, 4 reads are allowed. 100: No. in RX FIFO \geq 5, 5 reads are allowed. 101: No. in RX FIFO \geq 6, 6 reads are allowed. 110: No. in RX FIFO \geq 7, 7 reads are allowed. 111: No. in RX FIFO \geq 8, 8 reads are allowed.
[3]	-	-	Reserved	-
[2:0]	RX_FLAG[2:0]	R	<p>Receive FIFO Data Level</p> <p>This register indicates how many data has been received in receive FIFO.</p>	000: 0 byte in FIFO. 001: 1 byte in FIFO. 010: 2 bytes in FIFO. 011: 3 bytes in FIFO. 100: 4 bytes in FIFO. 101: 5 bytes in FIFO. 110: 6 bytes in FIFO. 111: 7 bytes in FIFO.

19. Inter Integrated Circuit Interface (I2C)

19.1. Introduction

The I²C is used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance.

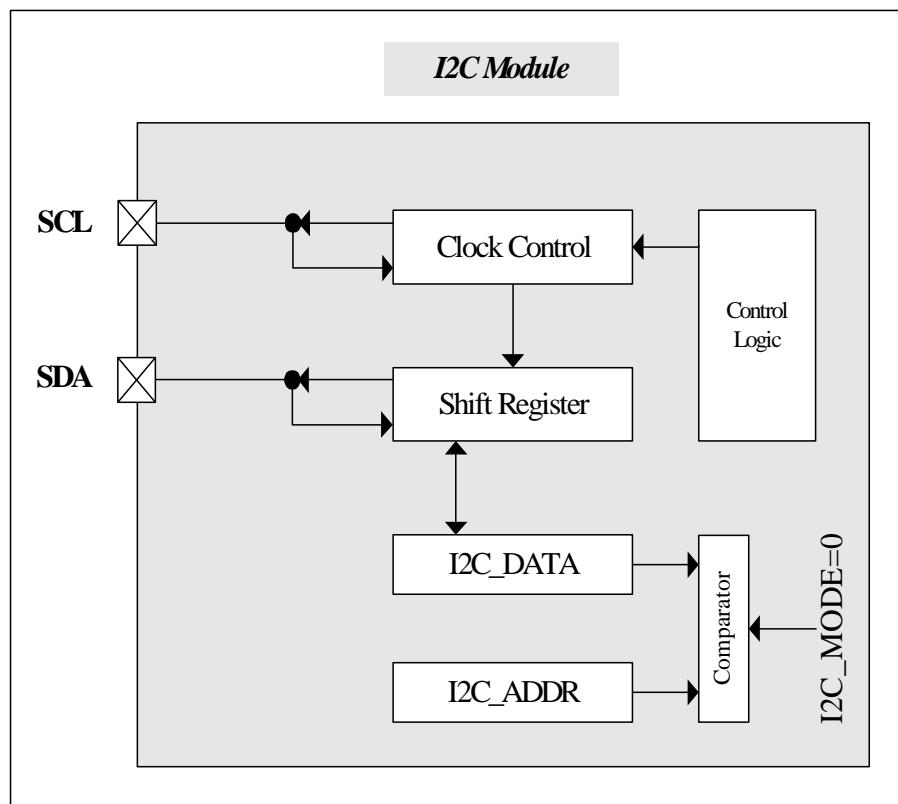
The I²C uses only two bidirectional open-drain lines: serial data line (SDA) and serial clock line (SCL). Logic 0 connects the bus to GND through I²C internal structure. Logic 1 makes the bus stay in floating state through I²C internal structure, and then pulls the bus to logic High by Pull-High resistance (both SDA and SCL are required).

19.2. Features

- Multi-Master Capability
- Support I²C master and slave mode.
- Generation and detection of 7-bit/10-bit addressing and General Call.
- Optional Clock
- Support DMA capability.

19.3. Block Diagram

The Figure is the block diagram of I²C.



19.4. Function

I²C supports 7-bit or 10-bit (depending on the device in use) address space. Only two wires (SCK and SDA) are needed to implement the protocol. In multi-master I²C-bus mode, multiple microprocessors can receive or transmit serial data to or from slave devices. If more than one master tries to control the line simultaneously, an arbitration mechanism is used to judge which one is the bus owner.

In I²C controller, four transfer modes are supported:

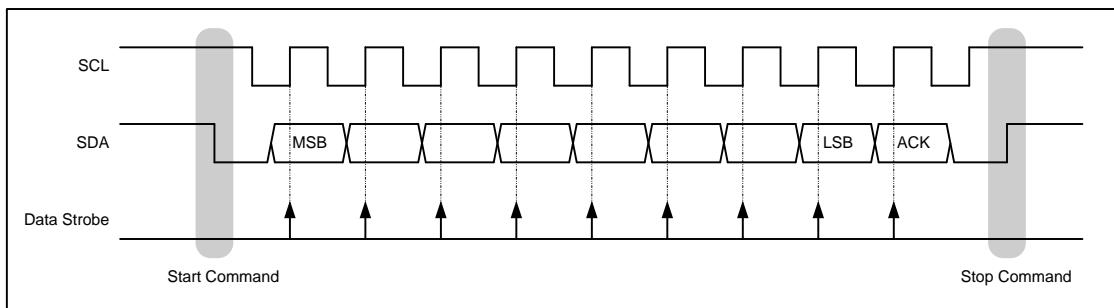
- Master Transmit – The master is sending data to a slave.
- Master Receive – The master is receiving data from a slave.
- Slave Transmit – The slave is sending data to the master.
- Slave Receive – The slave is receiving data from the master.

GPCM3 has three IO pin options for I²C0 I/F:

- IOA[0] (CLK) & IOA[1] (Data) pins.
- IOA[13] (CLK) & IOA[17] (Data) pins.
- IOA[15] (CLK) & IOA[16] (Data) pins.

Formats of I²C Frame

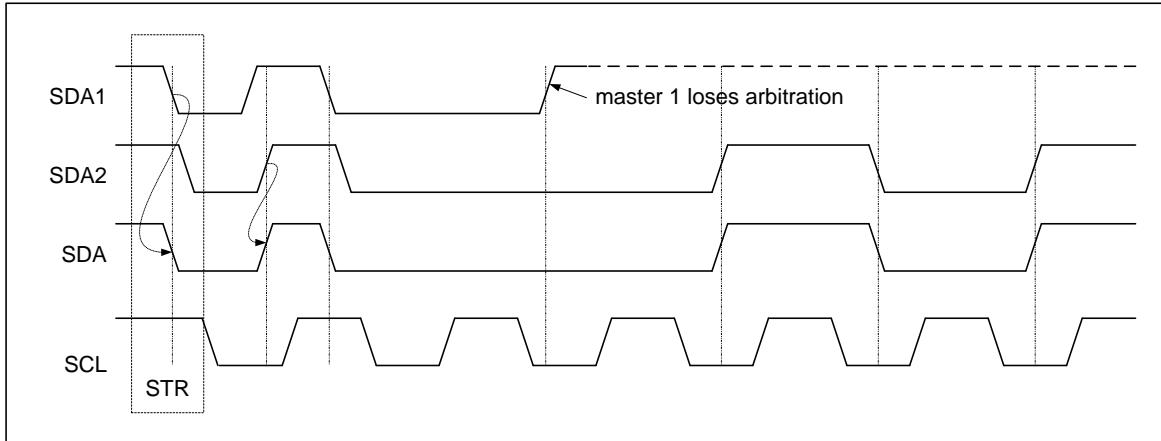
The following figure shows the frame format of I²C.



19.4.1. Master Mode

19.4.1.1. Arbitration Lost

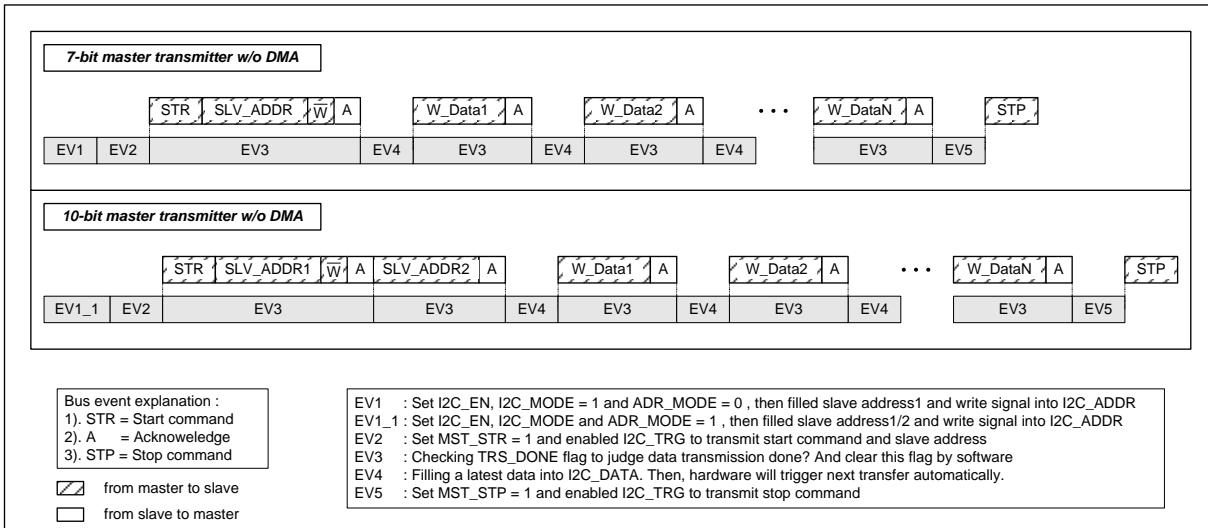
A master can only start a transfer when the bus is free. Arbitration takes place on the SDA line when the SCL line is at the HIGH level. If the master transmits a HIGH level while another master is transmitting a LOW level, the DATA output stage will be switched off because the level on the bus doesn't correspond to its own level. The **I2C_STS.ARB_LOST** is set as 1 by the hardware when the I2C interface detects an arbitration lost. Then, I2C controller switches from master to slave mode automatically. A master will still generate clock pulses until the end of the packet, even if it loses the arbitration. The following figure shows an arbitration mechanism of two masters.



Arbitration Mechanism of Two Masters

19.4.1.2. I2C Master Transmit with Manual Mode

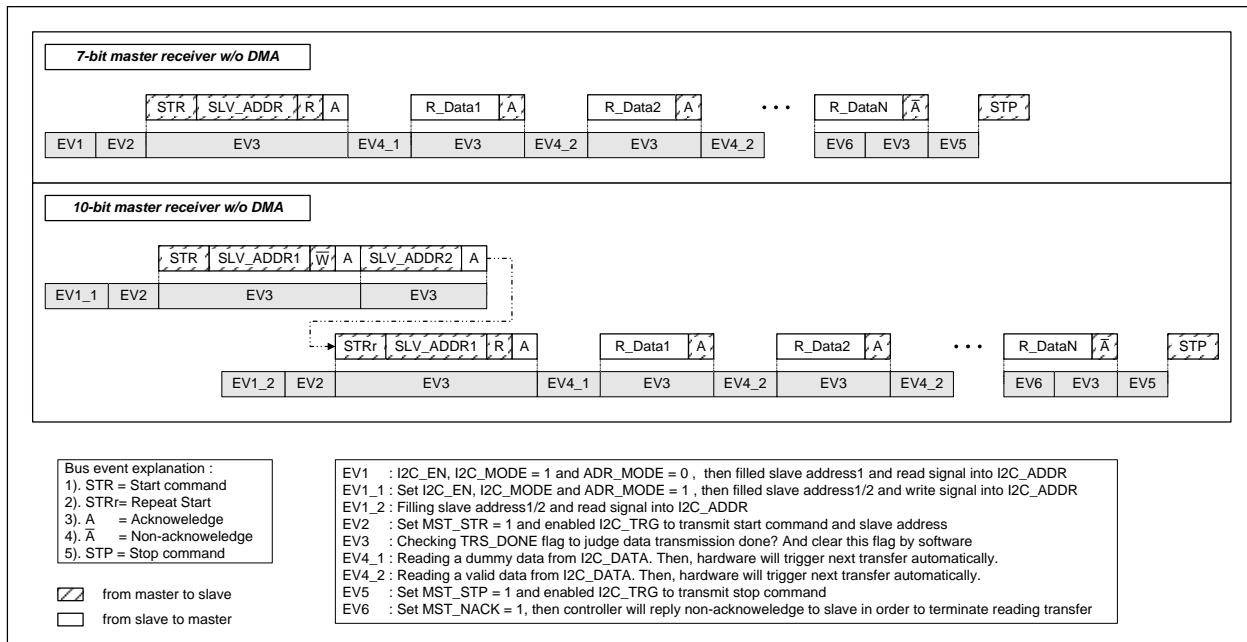
In master mode, before using I2C transmits function, user must fill in transfer data and set **I2C_CTRL.I2C_EN** as 1. In addition, master can inform slave that this transfer is a write transfer by setting **I2C_ADDR.RW_SEL** as 0. Then, I2C will send out data immediately when **I2C_CTRL.I2C_TRG = 1**. During an I2C transmission, the data on the SDA pin is shifted out from most significant bit first. The **I2C_STSTRS_DONE** flag will be set as 1 when acknowledge bit finishes. Before the next transmission, user must clear this bit by the software first, and then fill in the latest data in **I2C_DATA**. By doing so, the next transfer will be triggered. If user wants to terminate I2C transmission, set **I2C_CTRL.MST_STP** and **I2C_CTRL.I2C_TRG** as 1. The following figure shows a master transmit with manual mode.



I2C Master Transmit with Manual Mode

19.4.1.3. I2C Master Receive with Manual Mode

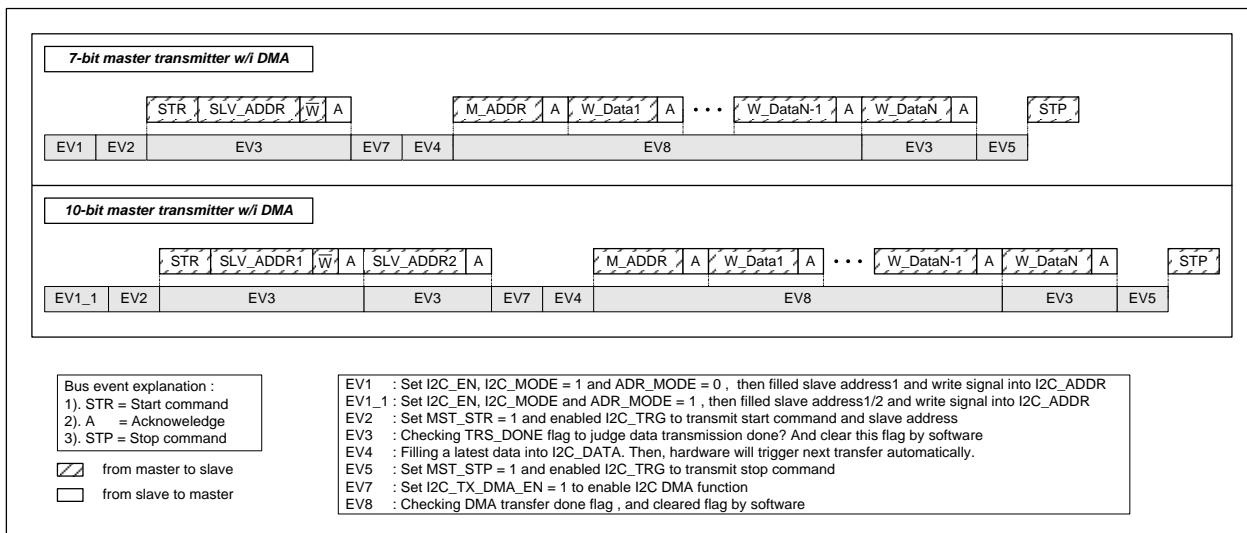
In master mode, before using I2C receives function, user must fill in transfer data and set **I2C_CTRL.I2C_EN** as 1. In addition, the master can inform the slave that this transfer is a read transfer by setting **I2C_ADDR.RW_SEL** as 1. Then, I2C will send out clock when **I2C_CTRL.I2C_TRG = 1**. During an I2C transmission, the data is shifted in from most significant bit of the SDA pin first. The **I2C_STS.TRS_DONE** will be set as 1 when acknowledge bit finishes. Before the next transmission, user must clear this bit by the software first, and read data from **I2C_DATA**. By doing so, the next transfer will be triggered. If user wants to terminate I2C transmission, set **I2C_CTRL.MST_NACK** as 1 during the final read transfer. After the final transfer is done, set **I2C_CTRL.MST_STP** and **I2C_CTRL.I2C_TRG** as 1. The following figure shows a master receive with manual mode.



I2C Master Receive with Manual Mode

19.4.1.4. I2C Master Transmit with DMA

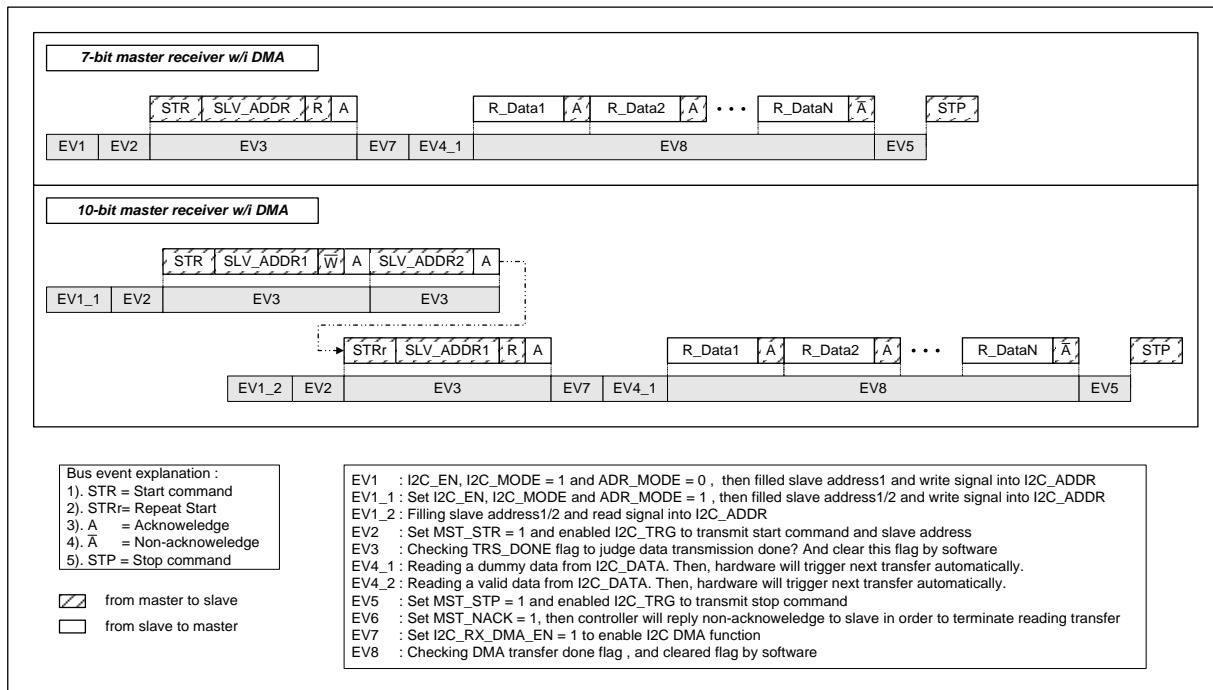
In master mode, before using I2C transmits function, user must fill in transfer data and set **I2C_CTRL.I2C_EN** as 1. In addition, the master can inform the slave that this transfer is a write transfer by setting **I2C_ADDR.RW_SEL** as 0. Then, I2C will send out data immediately when **I2C_CTRL.I2C_TRG = 1**. After slave-address transmission finishes, set **I2C_CTRL.TX_DMA_EN** as 1 and fill a piece of data in **I2Cx_DATA**, and the hardware will trigger the next transfer automatically. Next, check DMA transfer done flag to see whether the transmission is done or not. If DMA transfer done flag is set as 1, clear this flag by the software and wait for the final transfer done by checking **I2C_STS.TRS_DONE** flag. If user wants to terminate I2C transmission, set **I2C_CTRLMST_STP** and **I2C_CTRL.I2C_TRG** as 1. The following figure shows a master transmit with DMA mode.



I2C Master Transmitter with DMA

19.4.1.5. I2C Master Receive with DMA

In master mode, before using I2C receives function, user must fill in transfer data and set **I2C_CTRL.I2C_EN** as 1. In addition, the master can inform the slave that this transfer is a read transfer by setting **I2C_ADDR.RW_SEL** as 1. Then, I2C will send out clock when **I2C_CTRL.I2C_TRG = 1**. After slave-address transmission finishes, set **I2C_CTRL.RX_DMA_EN** as 1 and read a dummy data from **I2C_DATA**, and the hardware will trigger the next transfer automatically. Next, check DMA transfer done flag to see whether the transmission is done or not. If DMA transfer done flag is set as 1, clear this flag by the software. After DMA transfer is done, set **I2C_CTRLMST_STP** and **I2C_CTRL.I2C_TRG** as 1 and I2C transmission will be terminated. The following figure shows a master receive with DMA mode.



I2C Master Receiver with DMA

19.4.2. Slave Mode

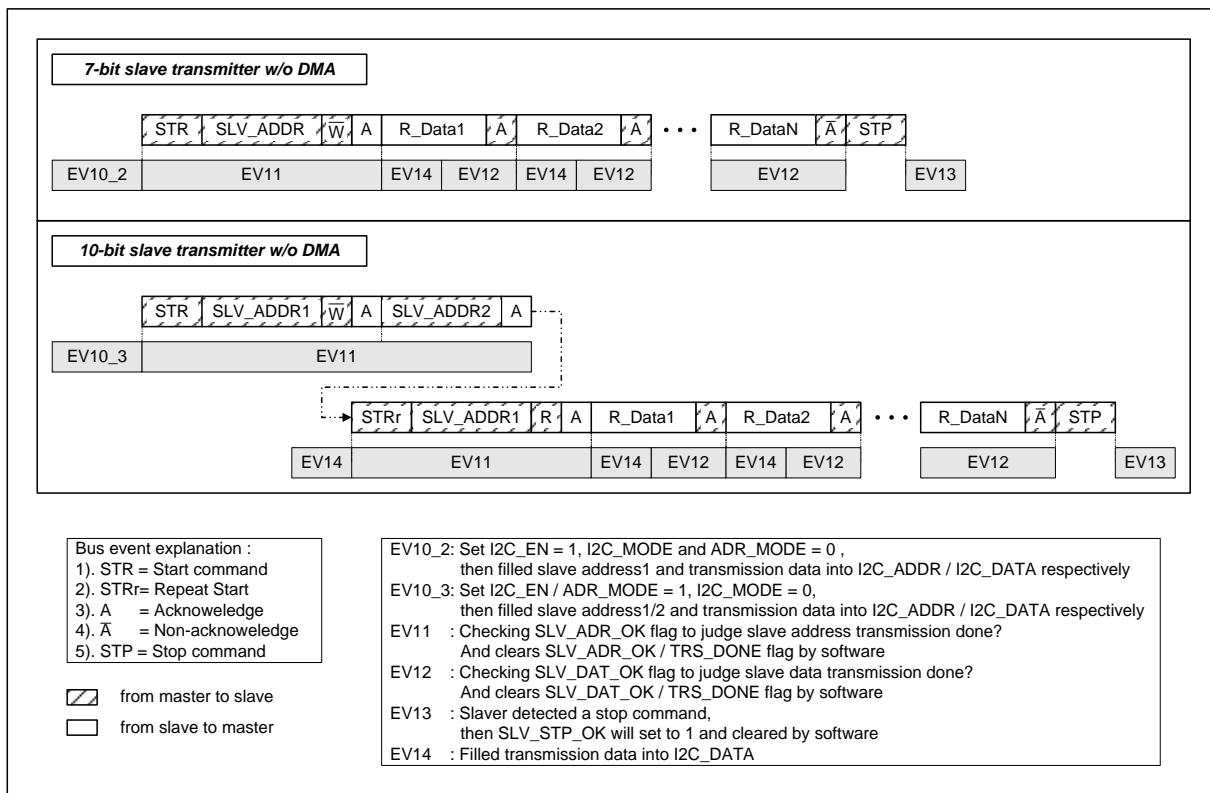
19.4.2.1. General Call

In I2C bus, a ‘general call’ address can address all devices. All devices should respond acknowledge when this address is used. However, devices can be made to ignore this address. The **I2C_STS.GEN_CALL** will be set as 1 when the slave receives a general call. CPU can receive an interrupt request if **I2C_CTRL.I2C_INT_EN** is set as 1.

19.4.2.2. I2C Slave Transmit with Manual Mode

In slave mode, before using I2C receives function, user must fill in slave-address, transfer data and set **I2C_CTRL.I2C_EN** as 1. Slave controller will set **I2C_STS.SLV_DATA_DONE** flag as 1 if the received data of slave address matches **I2Cx_ADDR**. Otherwise **I2C_STS.SLV_ADR_ERR** flag will be set as 1. After slave-address checking, fill data in **I2C_DATA** and waits **I2C_STS.DATA_DONE** flag to be set as 1. The transfer will continue until a stop command is received.

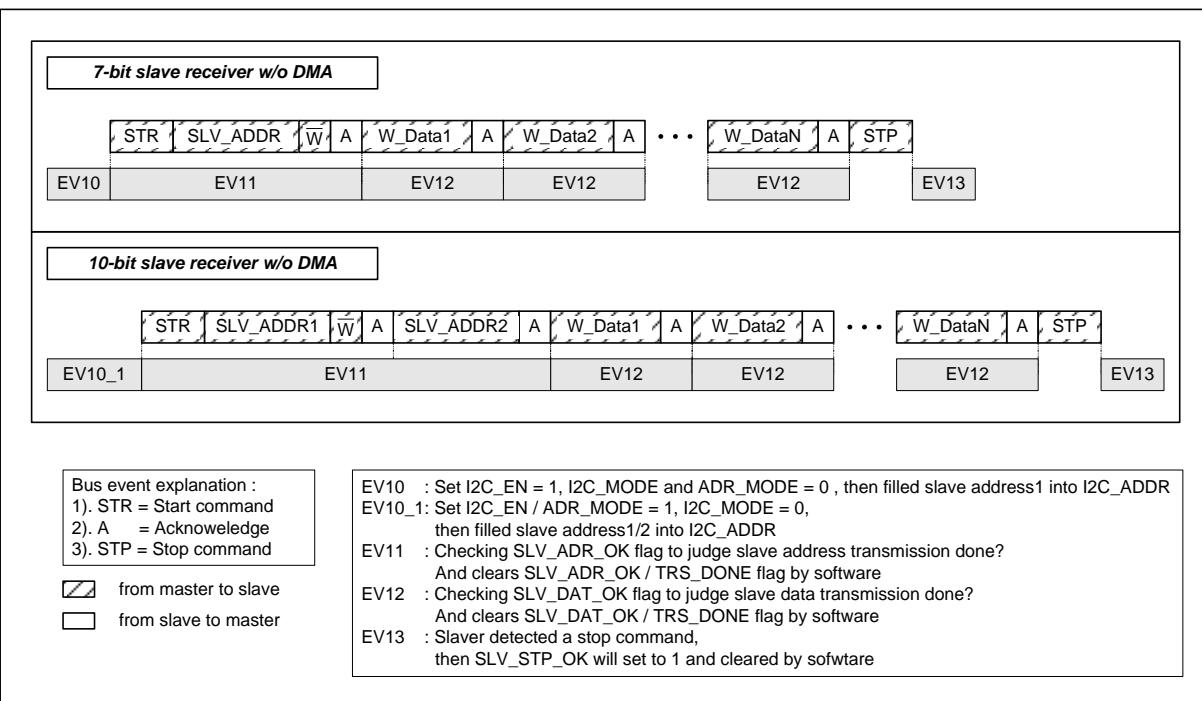
The following figure shows a slave receive with manual mode.



I2C Slave Transmit with Manual Mode

19.4.2.3. I2C Slave Receive with Manual Mode

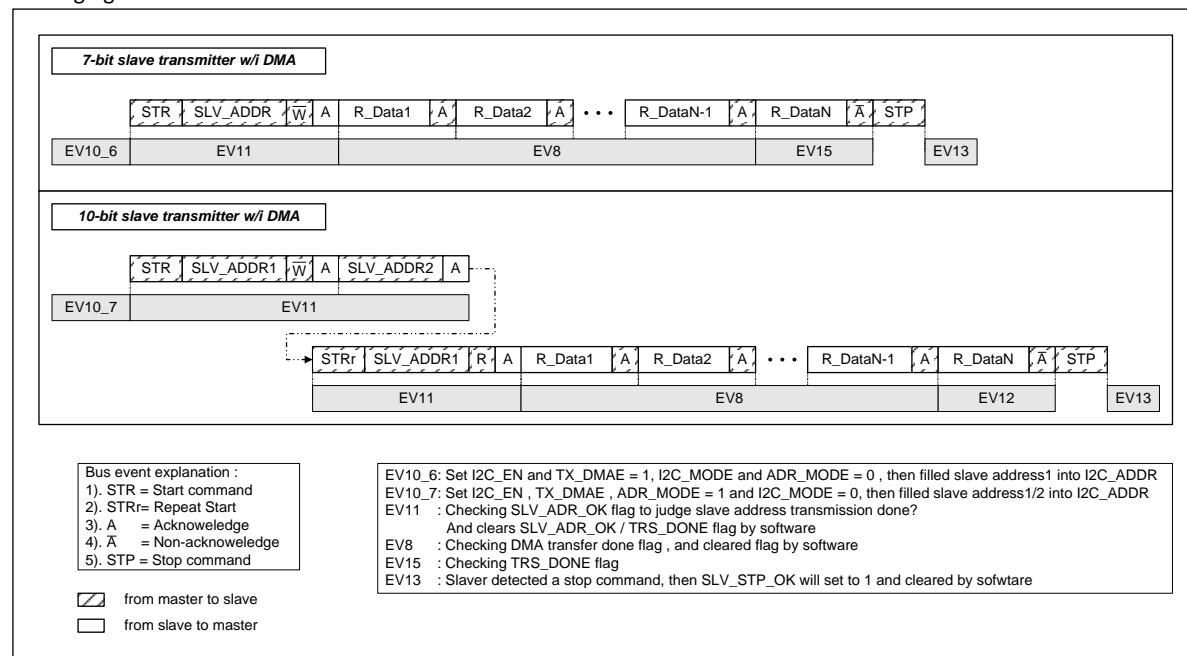
In slave mode, before receiving data from the master, user must fill in slave-address and set **I2C_CTRL.I2C_EN** as 1. Slave controller will set **I2C_STS.SLV_DATA_DONE** flag as 1 if the received data of slave address matches **I2Cx_ADDR**. Otherwise **I2C_STS.I2C_ADR_ERR** flag will be set as 1. After slave-address checking, slave controller starts to receive input data from the master. During an I2C transmission, the **I2C_STS.TRS_DONE** and **I2C_STS.DATA_DONE** flag will be set as 1. User must clear these flags by the software. Next, read the received data from **I2C_DATA**. The transfer will continue until a stop command is received. The following figure shows a slave transmit with manual mode.



I2C Slave Receive with Manual Mode

19.4.2.4. I2C Slave Transmit with DMA

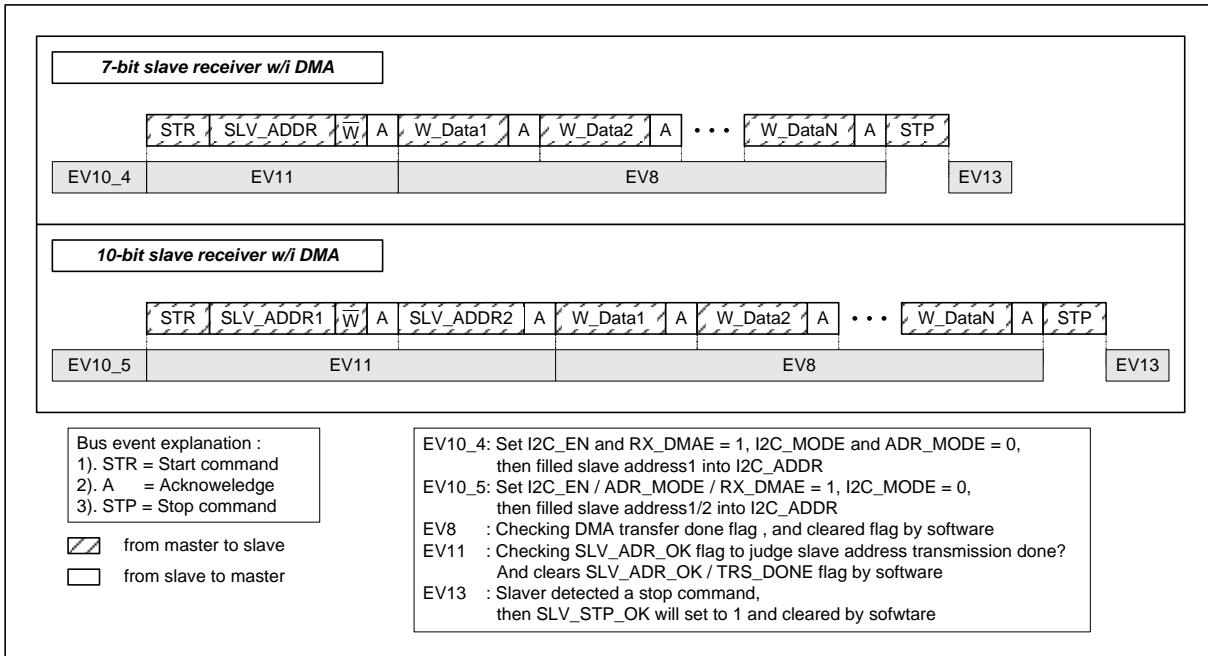
In slave mode, before using I2C transmits function, user must fill in slave-address and set **I2C_CTRL.TX_DMA_EN**, **I2C_CTRL.I2C_EN** as 1. After **I2C_STS.SLV_DATA_DONE** flag is set as 1, check DMA transfer done flag to see whether the transmission is done or not. If DMA finishes data transfer, the slave can tell whether the data transfer is done by checking **I2C_STS.TRS_DONE** flag. If the slave receives a stop command from the master, the transfer will be terminated. The following figure shows a slave transmit with DMA mode.



I2C Slave Transmit with DMA

19.4.2.5. I2C Slave Receive with DMA

In slave mode, before using I2C receives function, user must fill in slave-address and set **I2C_CTRL.RX_DMA_EN**, **I2C_CTRL.I2C_EN** as 1. After **I2C_STS.SLV_DATA_DONE** flag is set as 1, checks DMA transfer done flag to see whether the transmission is done or not. If DMA finishes data transfer, the slave can tell whether the data transfer is done by checking **I2C_STS.TRS_DONE** flag. These bits can be cleared by the software. If the slave receives a stop command from the master, the transfer will be terminated. The following figure shows a slave receive with DMA mode.



I2C Slave Receive with DMA

19.5. Register Description

Register map

Name	Address	Description
I2C_CTRL	0x400B0000	I2C Control Register
I2C_STS	0x400B0004	I2C Status Register
I2C_ADDR	0x400B0008	I2C Address Register
I2C_DATA	0x400B000C	I2C Data Register

Register Function

I2C_CTRL								0x400B0000			I2C Control Register		
Bit	31	30	29	28	27	26	25	24					
Function	-						RX_DMA_E	TX_DMA_E	MODE_SEL				
Reset Value	0	0	0	0	0	0	0	0					

Bit	23	22	21	20	19	18	17	16
Function	ERR_SADR_INT_EN	I2C_INT_EN	DB_TIME[5:0]					
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	MST_STR	MST_STP	MST_NACK	-	CLK_SEL[2:0]			I2C_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-				I2C_TRG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:27]	-	-	Reserved	-
[26]	RX_DMA_EN	R/W	I2C Rx Mode DMA Enable	0: Disabled 1: Enabled
[25]	TX_DMA_EN	R/W	I2C Tx Mode DMA Enable	0: Disabled 1: Enabled
[24]	MODE_SEL	R/W	I2C Controller Operating Mode Selection	0: Slave Mode 1: Master Mode
[23]	ERR_SADR_INT_EN	R/W	Slaver Address Error Interrupt Enable Note: This bit is available when I2C controller is operated in slaver mode.	0: Disabled 1: Enabled
[22]	I2C_INT_EN	R/W	I2C Interrupt Enable	0: Disabled 1: Enabled
[21:16]	DB_TIME[5:0]	R	SCL/SDA Input De-Bounce Time	-
[15]	MST_STR	R/W	I2C Controller Issued Start Command Enable Note: This bit will be cleared automatically when this transfer finishes.	0: Disabled 1: Enabled
[14]	MST_STP	R/W	I2C Controller Issued Stop Command Enable Note: This bit will be cleared automatically when this transfer finishes.	0: Disabled 1: Enabled
[13]	MST_NACK	R/W	I2C Controller Issued Non-Acknowledge Note: This bit will be cleared automatically when this transfer finishes.	0: Disabled 1: Enabled
[12]	-	-	Reserved	-
[11:9]	CLK_SEL[2:0]	R/W	I2C Controller Serial Clock Selection	000: System Clock/16 001: System Clock/32 010: System Clock/64 011: System Clock/128 100: System Clock/256 101: System Clock/768 110: System Clock/1024 111: -
[8]	I2C_EN	R/W	I2C Controller Enable	0: Disabled 1: Enabled
[7:1]	-	-	Reserved	-

Bit	Function	Type	Description					Condition
[0]	I2C_TRG	R/W	I2C Start Transmission Trigger This bit is for master mode only. The I2C master will begin to transmit or receive data when I2C_EN is set as 1. This bit will be cleared by the H/W automatically. Note: This bit is available when I2C_CTRL.I2C_EN is enabled.					0: Disabled 1: Enabled

I2C_STS								
0x400B0004								
Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-	SLV_ADR_E RR	BUSY	GEN_CALL	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	ARB_LOST	-	SLV_DATA_DONE	DATA_DONE	RX_STOP_C MD	RX_NO_ACK	TRS_DONE
Reset Value	0	0	0	0	0	0	1	0

Bit	Function	Type	Description					Condition
[31:12]	-	-	Reserved					-
[11]	SLV_ADR_ERR	R/W	I2C Slave Address Error Flag Note: This bit is only used in slave mode.					Read 0: Slave address is the correct address. Read 1: Slaver address is the wrong address. Write 0: No effect. Write 1: Clear this bit.
[10]	BUSY	R/W	I2C Controller Busy Flag					Read 0: No communication on the bus. Read 1: Communication is ongoing on the bus. Write 0: No effect. Write 1: Clear this bit.
[9]	GEN_CALL	R/W	I2C general call flag					Read 0: I2C master has not issued general call.

Bit	Function	Type	Description	Condition
				Read 1: I2C master has issued general call. Write 0: No effect. Write 1: Clear this bit.
[8:7]	-	-	Reserved	-
[6]	ARB_LOST	R/W	I2C Bus Arbitration Lost Flag Note: This error occurs when the I2C interface detects an arbitration lost condition.	Read 0: Not occur. Read 1: Occurs. Write 0: No effect. Write 1: Clear this bit.
[5]	-	-	Reserved	-
[4]	SLV_DATA_DONE	R	I2C Slave Address Received Flag Note: This bit is only used in slave mode.	Read 0: Slave address is not asserted. Read 1: Slave address is asserted and matches the setting. Write 0: No effect. Write 1: Clear this bit.
[3]	DATA_DONE	R/W	I2C Tx/Rx Data Done Flag Note: This bit is only used in slave mode.	Read 0: Data is transmitting or idle now. Read 1: Data transmission is completed. Write 0: No effect. Write 1: Clear this bit.
[2]	RX_STOP_CMD	R/W	I2C Stop Command Received Flag Note: This bit is only used in slave mode.	Read 0: Stop command has not been received. Read 1: Stop command has been received. Write 0: No effect. Write 1: Clear this bit.
[1]	RX_NO_ACK	R/W	I2C No Ack Received Flag	Read 0: Acknowledge. Read 1: No acknowledge. Write 0: Clear this bit. Write 1: No effect.
[0]	TRS_DONE	R/W	I2C Transmission Complete Flag	Read 0: I2C is idle or on going. Read 1: I2C finishes data transmission. Write 0: No effect. Write 1: Clear this bit.

I2C_ADDR									0x400B0008									I2C Address Register									
Bit	31	30	29	28	27	26	25	24																			
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							ADDR_MODE
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	ADDRESS_2[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ADDRESS_1[6:0]							RW_SEL
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:17]	-	-	Reserved	-
[16]	ADDR_MODE	R/W	I2C Address Mode Control Bit Note: This bit is available in slave mode only.	0: 7-bit Mode 1: 10-bit Mode
[15:8]	ADDRESS_2[7:0]	R/W	I2C Slave Address 2nd Byte	7-bit Address Mode: Don't Care 10-bit Address Mode: Slave Addr[7:0]
[7:1]	ADDRESS_1[6:0]	R/W	I2C Slave Address 1st 7Bits	7-bit Address Mode: Slave Addr[6:0] 10-bit Address Mode: 0x1E
[0]	RW_SEL	R/W	I2C Read/Write Control Bit Note: Master mode shares this bit with slave mode.	0: Write 1: Read

I2C_DATA									0x400B000C									I2C Data Register									
Bit	31	30	29	28	27	26	25	24																			
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	I2C_DATA[7:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	-	-	Reserved	-
[7:0]	I2C_DATA[7:0]	R/W	I2C Controller Read/Write Data Register Note: Master mode shares this bit with slave mode.	-

20. I2S

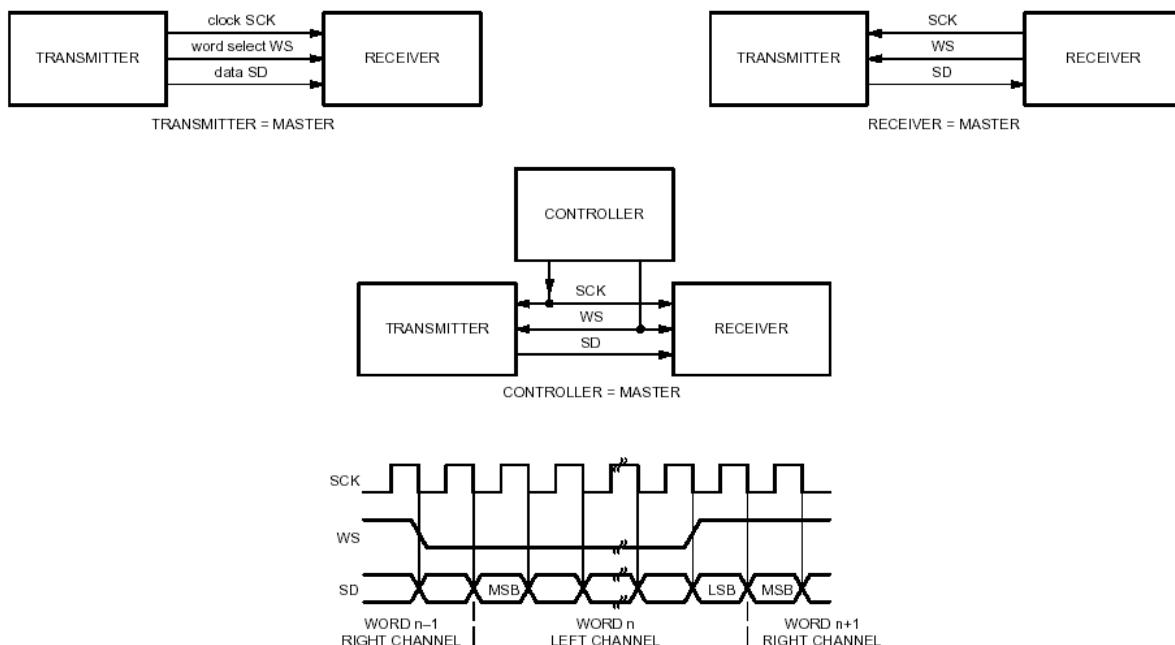
20.1. Introduction

The bus only has to handle audio data when other signals, such as sub-coding and controlling, are transferred separately. To minimize the number of pins and to keep wiring simple, a 3-line serial bus is used, consisting of a line for two time-multiplexed data channels: a word selection line and a clock line. Since the transmitter and receiver have the same clock signal for data transmission, the transmitter, as a master, has to generate the bit clock, word-select signal and data. In a complex system, however, there may be several transmitters and receivers, which make it difficult to define the master. In such system, there is usually a system master which controls the digital audio data-flow between ICs. The transmitters, acting as slaves, will then generate data under the control of an external clock. Note that the system master can be combined with a transmitter or receiver, and it may be enabled or disabled under software controlling or by pin programming.

20.2. Features

- One TX channel is supported (MCLK: IOA[9]/ BCLK: IOA[10]/ SLR: IOA[11]/ DATA: IOA[12] or MCLK: IOA[22]/ BCLK: IOA[23]/ SLR: IOA[24]/ DATA: IOA[25]).
- One RX channel is supported (MCLK: IOA[3]/ BCLK: IOA[4]/ SLR: IOA[5]/ DATA: IOA[6]) or MCLK: IOA[26]/ BCLK: IOA[27]/ SLR: IOA[28]/ DATA: IOA[29]).
- 2 Words RX FIFO
- Support configurable settings of each TX/RX channel for different frame sizes, word length, frame synchronization mode, data alignment, MSB/LSB first send mode, rising/falling sending edge mode, frame polarity, and the polarity of first transmitted frame.

20.3. Block Diagram



20.4. Function

20.4.1. I2S Bus

As shown in figure, the bus has three lines:

- Continuous Serial Clock (SCK), or called Bit Clock (BCLK)
- Word Select (WS) or Left Right Clock(LRCK)
- Serial Data (SD)

The device generating SCK and WS is the master.

Serial Data

Serial data is transmitted in two's complement with the MSB first. The MSB is transmitted first because the transmitter and receiver may have different word lengths. It is not necessary for the transmitter to know the number of bits the receiver can afford; neither the receiver needs to know the number of bits is being transmitted. When the system's word length is greater than the transmitter's word length, the word is truncated (least significant data bits are set as 0) for data transmission. If the receiver receives more bits than its word length, the bits after the LSB will be ignored. On the other hand, if the receiver receives fewer bits than its word length, the missing bits will be given zero instead. So, the MSB has a fixed position while the position of the LSB depends on the word length. The transmitter always sends the MSB of the next word one clock period after the WS changes.

Serial data sent by the transmitter may be synchronized with either the trailing (HIGH-to-LOW) or the leading (LOW-to-HIGH) edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions of transmitting data which is synchronized with the leading edge.

Word Select

The word select line indicates the channel being transmitted:

WS = 0; channel 1 (left)

WS = 1; channel 2 (right)

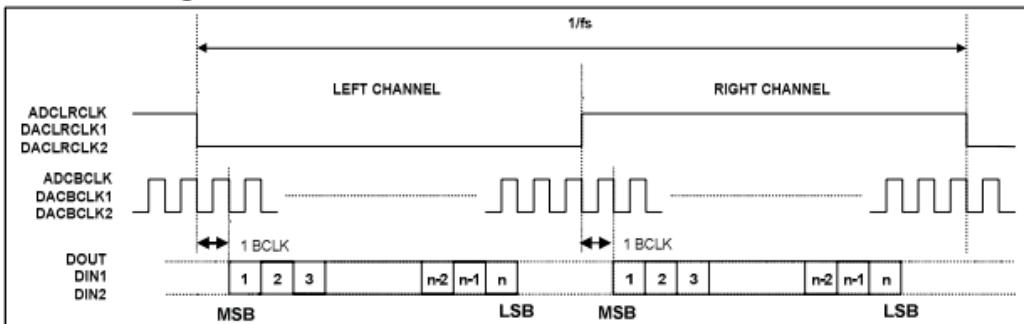
WS may change either on a trailing or leading edge of the serial clock but it doesn't need to be symmetrical. In slave, this signal is latched on the leading edge of the clock signal. The WS line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word.

20.4.2. Normal Mode and I2S Mode

I2S Mode

I2S mode is a subset of network mode and is used in audio application. One frame only consists of 2 slots: one slot is left audio channel, and the other is right audio channel. Generally, the I2S' frame sync asserts one bit before the first bit of the frame as shown the following figure.

I2S Mode Timing

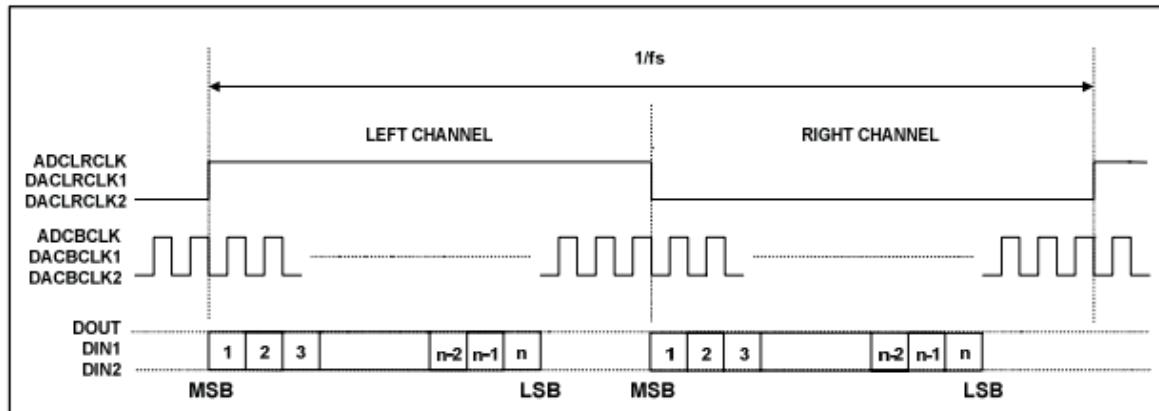


Normal Mode

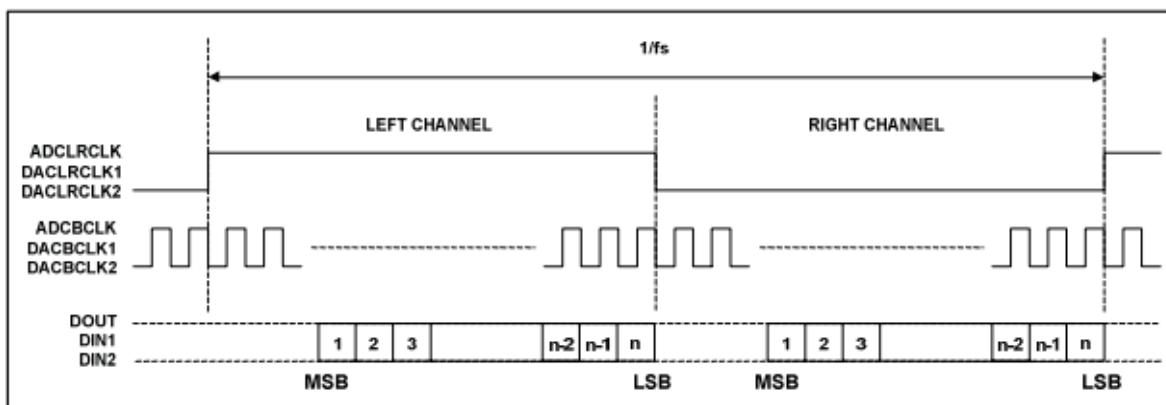
The major difference between Normal mode and I2S mode is the first bit of frame data. In Normal mode, the data will be aligned with Left/Right channel at rising/falling edge. But in I2S mode, the data will be shifted 1-bit backward.

As the examples shown below, the Normal mode has Left Justified and Right Justified mode.

Normal_LJ(Left Justified)



Normal_RJ(Right Justified)



20.4.3. I2S RX Interrupt

When the data stored in RX FIFO is more than half of its size, an interrupt will be activated. The function can be enabled or disabled by EN_HALF_FULL_IRT of I2S_CTRL.

20.4.4. I2S RX FIFO

RX FIFO is a 2-port FIFO, and it can read and write data at the same time. The I2S RX FIFO is 2 Words (2 * 32).

20.5. Register Description

Register Map

Name	Address	Description
I2S_RX_CTRL	0x400C0000	I2S RX Control Register
I2S_RX_DATA	0x400C0004	I2S RX Data Register
I2S_RX_STS	0x400C0008	I2S RX Status Register
I2S_RX_CTRL2	0x400C000C	I2S RX Control Register2



GPCM3 SERIES PROGRAMMING GUIDE

Name	Address	Description
I2S_TX_CTRL	0x400C0010	I2S TX Control Register
I2S_TX_CTRL2	0x400C0014	I2S TX Control Register2

Register Function

Bit	15	14	13	12	11	10	9	8
Function	-	INT_SEL	MODE_SEL	FRAMING_MODE[1:0]]		FRAME_SIZE[1:0]		DATA_LENGTH[2]
Reset Value	0	1	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
Function	DATA_LENGTH[1:0]		ALIGN	DATA_ENDIANNESS	DATA_LATC_H	FRAME_POL	1ST_FRAME_POL	RX_EN
Reset Value	0	0	1	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:23]	-	R	Reserved	-
[22]	RX_MCLK_EN	R/W	MCLK Output Enable/Disable	0 : Disable 1 : Enable
[21]	MONO_MODE_ENABLE	R/W	Mono Mode Enable Note: If MONO_MODE is 1, I2S RX will only receive the left or right channel data according to FRAME_POL and RX_FRAME_POL.	0: Stereo Mode 1: Mono Mode
[20]	RIGHT_LSB_ENABLE	R/W	Data Order in 16-bit Merge Mode Note: Whether the Right channel data is in LSB of RX_DATA, it is valid only when the word length is 16-bit and the MERGE function is on.	0: RX_DATA[15:0]: left channel data, RX_DATA[31:16]:right channel data. 1: RX_DATA[15:0]: right channel data, RX_DATA[31:16]: left channel data.

Bit	Function	Type	Description	Condition
[19]	MERGE_ENABLE	R/W	16-bit Merge Mode Note: For 24-bit or 32-bit frame length, this bit must be 0. Note: In Merge mode, it will merge two 16-bit data into one 32-bit data.	0 : Disable 1 : Enable
[18:17]	-	R	Reserved	-
[16]	HALF_FULL_INT_ENABLE	R/W	I2S FIFO Half Full Interrupt Enable	0 : Disable 1 : Enable
[15]	-	R	Reserved	-
[14]	INT_SEL	R/W	Interrupt Polarity Selection	0: Falling-Edge Triggered 1: Rising-Edge Triggered
[13]	MODE_SEL	R/W	Master/Slave Mode Selection When this bit is 0, the I2S RX controller acts as a slave, and receives LRCK and BLCK from the transmitter (Master).	0: Slave, receive LRCK_IN (fixed). 1: Master
[12:11]	FRAMING_MODE[1:0]	R/W	Framing Mode	00: I2S Mode 01: Normal Mode 1x: Reserved
[10:9]	FRAME_SIZE[1:0]	R/W	Frame Size If the frame size of the codec cannot be adjusted and equal to frame size, it's suggested to use 11 (the frame size is not predictable). By doing so, I2S controller will determine the size automatically.	00: 16-bit Length 01: 24-bit Length 10: 32-bit Length 11: Frame size is unpredictable, only used in slave mode.
[8:6]	DATA_LENGTH[2:0]	R/W	Valid Data Length When the valid data length is not 32-bit, extra MSB bits of the 32-bit word are "Don't Care". Only the specified data bit length is meaningful. For example, when data word length is 20-bit, bit[19:0] is audio data and bit[31:20] is invalid data. Note that the data word length must not be greater than the frame size.	000: 16-bit Length 001: 18-bit Length 010: 20-bit Length 011: 22-bit Length 100: 24-bit Length 101: 32-bit Length 110: 32-bit Length 111: 32-bit Length
[5]	ALIGN	R/W	Received Data Right/Left Alignment When the frame size is greater than the valid data length, left alignment informs I2S controller that the valid data bit will be received first and the dummy bits will be received later. Right alignment means the dummy bits will be received first and the valid data bits will be received later.	0: Right Alignment 1: Left Alignment (Default)

Bit	Function	Type	Description	Condition
[4]	DATA_ENDIANNESS	R/W	Bit Priority Receiving Mode	0: MSB First (Default) 1: LSB First
[3]	DATA_LATCH	R/W	Receiving Data Latch Mode Selection	0: Receive data bit on serial clock falling edge. (Default) 1: Receive data bit on serial clock rising edge.
[2]	FRAME_POL	R/W	Receiving Frame Polarity Selection User should assure the frame polarity configurations of RX and Transmitter are the same.	0: LRCK = 0 is the right frame. (Default) 1: LRCK = 0 is the left frame.
[1]	1ST_FRAME_POL	R/W	First Frame Polarity User should assure the first frame LR polarity configurations of RX and Transmitter are the same.	0: Left Frame (Default) 1: Right Frame
[0]	RX_EN	R/W	Enable I2S RX RX_EN is high active enable signal for receive data. User should finish all the required configurations before enabling I2S RX.	0: Disabled 1: Enabled

I2S_RX_DATA								0x400C0004								I2S RX Data Register								
Bit	31	30	29	28	27	26	25	24	RX_DATA[31:24]															
Function																								
Reset Value	0								0								0							
Bit	23	22	21	20	19	18	17	16	RX_DATA[23:16]															
Function																								
Reset Value	0								0								0							
Bit	15	14	13	12	11	10	9	8	RX_DATA[15:8]															
Function																								
Reset Value	0								0								0							
Bit	7	6	5	4	3	2	1	0	RX_DATA[7:0]															
Function																								
Reset Value	0								0								0							

Bit	Function	Type	Description					Condition
[31:0]	RX_DATA[31:0]	R	RX_DATA Port Reading data from this port by CPU or DMA will obtain data from RX FIFO.					-



GPCM3 SERIES PROGRAMMING GUIDE

Bit	Function	Type	Description	Condition
[31:19]	-	R	Reserved	-
[18]	CLEAR_FIFO_ENABLE	R/W	<p>Clear RX FIFO</p> <p>RX FIFO bit will be cleared as 0 automatically after the FIFO pointer is cleared.</p> <p>Write 1 to clear RX FIFO. After polling and this bit becomes 0, user will know that Clean RX FIFO action is done.</p>	<p>Read 0: Clear Rx FIFO action is done.</p> <p>Read 1: RX FIFO is not empty.</p> <p>Write 0: No effect.</p> <p>Write 1: Clear Rx FIFO.</p>
[17]	HALF_FULL_FLAG	R/W	I2S FIFO Half Full INT Status Flag	<p>0: No Interrupt</p> <p>1: Interrupt occurs. (Write 1 to clear.)</p>
[16]	-	R	Reserved	-
[15]	OVERFLOW_FLAG	R/W	<p>Overflow Flag</p> <p>If RX FIFO is full and it's still receiving data, OVERFLOW_FLAG will become 1.</p> <p>Note that when OVERFLOW_FLAG occurs, it means the receiving audio samples are lost and broken sounds occur. The software should avoid the occurrence of overflow.</p>	<p>Read 0: No overflow occurs.</p> <p>Read 1: Overflow occurs.</p> <p>Write 0: No effect.</p> <p>Write 1: Clear the flag.</p>
[14:4]	-	R	Reserved	-
[3:2]	FIFO_INT_LEVEL[1:0]	R	<p>RX FIFO Number</p> <p>The number of word stored in the RX FIFO.</p>	-
[1]	FIFO_FULL_FLAG	R	<p>RX FIFO Full Flag</p> <p>When FIFO becomes full, WORD_NO is 32; this flag is 1.</p>	<p>0: RX FIFO is not full.</p> <p>1: RX FIFO is full.</p>

Bit	Function	Type	Description					Condition
[0]	FIFO_EMPTY_FLAG	R	RX FIFO Empty Flag When FIFO does not have any data, WORD_NO is 0; this flag is 1.					0: RX FIFO is not empty. 1: RX FIFO is empty.

I2S_RX_CTRL2 **0x400C000C** **I2S RX Control Register2**

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

Bit	Function	Type	Description					Condition
[31:29]	BCLK_DIV_SEL[2:0]	R/W	I2S Master Mode BCLK					000 ~ 001: Reserved 010: I2S BCLK = MCLK/2 011: I2S BCLK = MCLK/3 100: I2S BCLK = MCLK/4 101: Reserved 110: I2S BCLK = MCLK/6 111: I2S BCLK = MCLK/8
[28:6]	-	R	Reserved					-
[5:0]	MCLK_DIV_SEL[5:0]	R/W	I2S Master Clock					0: MCLK = Pclk/2 1~63: MCLK = Pclk/(MCLK_DIV+1)

I2S_TX_CTRL **0x400C0010** **I2S TX Control Register**

Bit	31	30	29	28	27	26	25	24
Function	MCLK_DIV_SEL[5:0]					BCLK_DIV_SEL[2:1]		
Reset Value	0	0	0	0	1	0	1	1

Bit	23	22	21	20	19	18	17	16
Function	BCLK_DIV_SEL[0]	-			DATA_FMT	MCLK_EN	DATA_SRC[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	DATA_OUT[1:0]		-			FRAME_SIZE[1:0]		-
Reset Value	0	0	1	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Function	-		ALIGN	DATA_ENDIANNESS	DATA_EDGE_MODE	FRAME_MODE	-	TX_EN
Reset Value	0	0	1	0	0	1	0	0

Bit	Function	Type	Description	Condition
[31:26]	MCLK_DIV_SEL[5:0]	R/W	I2S Master Clock	0: MCLK = Pclk/2 1~63: MCLK = Pclk/(MCLK_DIV+1)
[25:23]	BCLK_DIV_SEL[2:0]	R/W	I2S Master Mode BCLK	000 ~ 001: Reserved 010: I2S BCLK = MCLK/2 011: I2S BCLK = MCLK/3 100: I2S BCLK = MCLK/4 101: Reserved 110: I2S BCLK = MCLK/6 111: I2S BCLK = MCLK/8
[22:20]	-	R	Reserved	-
[19]	DATA_FMT	R/W	Date Signed/Unsigned	0: Unsigned 1: Signed
[18]	MCLK_EN	R/W	MCLK Output Enable/Disable	0 : Disable 1 : Enable
[17:16]	DATA_SRC[1:0]	R/W	Audio Data Source Selection	00: Postwave (SPU+Soft CH) 01: SPU 10: Soft CH 11: Reserved
[15:14]	DATA_OUT[1:0]	R/W	Data Out Frame Pol and Mono/Stereo Selection	LRCK_L, LRCKH 00: Stereo (R, L) 01: Stereo (L, R) 10: Mono (L, L) 11: Mono (R, R)
[13:11]	-	R	Reserved	-
[10:9]	FRAME_SIZE[1:0]	R/W	Frame Size	00: 16-bit Length 01: 24-bit Length 10: 32-bit Length 11: 16-bit Length
[8:6]	-	R	Reserved	-
[5]	ALIGN	R/W	Data Right/Left Alignment	0: Right Alignment 1: Left Alignment (Default)
[4]	DATA_ENDIANNESS	R/W	Bit Priority Mode	0: MSB First (Default) 1: LSB First

Bit	Function	Type	Description	Condition
[3]	DATA_EDGE_MODE	R/W	Send Data Edge Mode Selection	0: Rising Edge 1: Falling Edge
[2]	FRAME_MODE	R/W	Frame Mode	0: I2S Mode 1: Normal Mode
[1]	-	R	Reserved	-
[0]	TX_EN	R/W	I2S TX Enable	0: Disabled 1: Enabled

I2S_TX_CTRL2 **0x400C0014** **I2S TX Control Register2**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					DUMMY_CYCLE[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					-		DELAY_EN	DELAY_TYPE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:24]	-	R	Reserved	-
[23:16]	DUMMY_CYCLE[7:0]	R/W	Dummy Cycle Counter	0x00: No Delay 0x01: Delay 1 MCLK ... 0xFF: Delay 256 MCLK
[15:2]	-	R	Reserved	-
[1]	DELAY_EN	R/W	Add delay to the last BCLK of TX data in order to adjust LR frequency.	0: Disabled 1: Enabled
[0]	DELAY_TYPE	R/W	The delay length can be determined by Dummy Cycle, or the delay will continue until TX data changes.	0: Dummy Cycle 1: Data Change

21. PWM I/O

21.1. Introduction

There are 16 PWM I/Os presented in GPCM3 series. PWM IO is usually used in the related applications of LCD and motor control. By configuring CLK source and Duty, the needed PWM output signal can be generated. There are seven different clock sources: $F_{CPU}/4096$, $F_{CPU}/1024$, $F_{CPU}/256$, $F_{CPU}/64$, $F_{CPU}/32$, $F_{CPU}/16$, $F_{CPU}/8$, and seven Timer/CCP overflow. It determines the PWM I/O frequency. Each I/O has individual port to control PWM duty. Note that the corresponding direction control bit must set as output; the buffer control bit of I/O is able to control output's polarity.

21.2. Features

- Support 16 PWM I/Os (IOA[28:13]).
- Support different clock sources (Timer0/1/2, CCP Timer0/1 and CPU CLK DIV 8 ~ 4096).
- Separate PWM duty control (8-bit Resolution, its valid bit will be effected by the given value of Period).
- Separate PWM Period control (32/64/128/256 period).

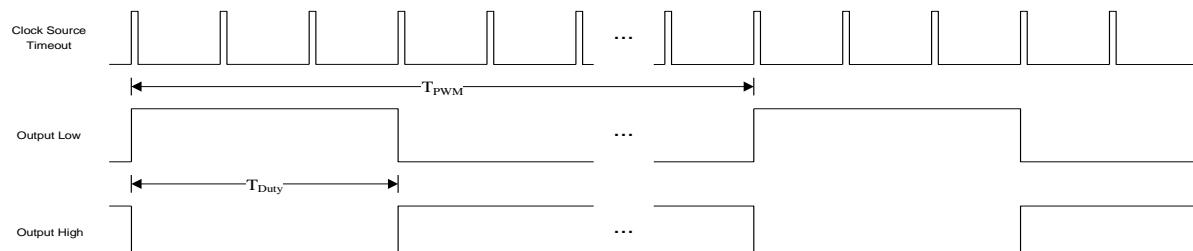
21.3. Function

PWM Output Diagram with I/O Configuration

Please refer to the following diagram.

- Set register CLK_SEL[3:0] for different clock source timeout.
- Set register Period_IOx[1:0] for released Tpwm period.
- Set register IOx_Duty[7:0] for released TDuty, or set TG_x = 1 to output PWM with 50% of duty cycle.
- Set register IOx_INV for inverse PWM out.

Note: PWM out mask function will refer to the configuration of PWMIO_CTRL1 Bit[31:16]. If the IOx_INV = 0, PWMIOx will be kept as 'Low'. If the IOx_INV = 1, PWMIOx will be kept as 'High'.



Note: $T_{PWM} = \text{Clock Source Timeout} * \text{Period}$

21.4. Register Description

Register Map

Name	Address	Description
PWMIO_CTRL0	0x40160000	PWMIO Control Register 0
PWMIO_CTRL1	0x40160004	PWMIO Control Register 1
PWMIO_TOGGLE_CTRL	0x40160008	PWMIO IO Toggle Control Register
PWMIO_PERIOD_CTRL	0x4016000C	PWMIO Period Control Register
PWMIO_PWMIO0_Duty	0x40160010	PWMIO0 Duty Register
PWMIO_PWMIO1_Duty	0x40160014	PWMIO1 Duty Register
PWMIO_PWMIO2_Duty	0x40160018	PWMIO2 Duty Register
PWMIO_PWMIO3_Duty	0x4016001C	PWMIO3 Duty Register

Name	Address	Description
PWMIO_PWMIO4_Duty	0x40160020	PWMIO4 Duty Register
PWMIO_PWMIO5_Duty	0x40160024	PWMIO5 Duty Register
PWMIO_PWMIO6_Duty	0x40160028	PWMIO6 Duty Register
PWMIO_PWMIO7_Duty	0x4016002C	PWMIO7 Duty Register
PWMIO_PWMIO8_Duty	0x40160030	PWMIO8 Duty Register
PWMIO_PWMIO9_Duty	0x40160034	PWMIO9 Duty Register
PWMIO_PWMIO10_Duty	0x40160038	PWMIO10 Duty Register
PWMIO_PWMIO11_Duty	0x4016003C	PWMIO11 Duty Register
PWMIO_PWMIO12_Duty	0x40160040	PWMIO12 Duty Register
PWMIO_PWMIO13_Duty	0x40160044	PWMIO13 Duty Register
PWMIO_PWMIO14_Duty	0x40160048	PWMIO14 Duty Register
PWMIO_PWMIO15_Duty	0x4016004C	PWMIO15 Duty Register
PWMIO_OFFSET	0x40160070	PWMIO Period Offset Register

Register Function
PWMIO_CTRL0
0x40160000
PWMIO Control Register 0

Bit	31	30	29	28	27	26	25	24
Function	PWM15_EN	PWM14_EN	PWM13_EN	PWM12_EN	PWM11_EN	PWM10_EN	PWM9_EN	PWM8_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	PWM7_EN	PWM6_EN	PWM5_EN	PWM4_EN	PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				SYNC_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function			CLK_SEL[3:0]				-	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31]	PWM15_EN	R/W	PWMIO 15 (IOA28) Enable				0: Disabled 1: Enabled
[30]	PWM14_EN	R/W	PWMIO 14 (IOA27)Enable				0: Disabled 1: Enabled
[29]	PWM13_EN	R/W	PWMIO 13 (IOA26)Enable				0: Disabled 1: Enabled
[28]	PWM12_EN	R/W	PWMIO 12 (IOA25) Enable				0: Disabled 1: Enabled
[27]	PWM11_EN	R/W	PWMIO 11 (IOA24) Enable				0: Disabled 1: Enabled
[26]	PWM10_EN	R/W	PWMIO 10 (IOA23) Enable				0: Disabled 1: Enabled

Bit	Function	Type	Description	Condition
[25]	PWM9_EN	R/W	PWMIO 9 (IOA22) Enable	0: Disabled 1: Enabled
[24]	PWM8_EN	R/W	PWMIO 8 (IOA21) Enable	0: Disabled 1: Enabled
[23]	PWM7_EN	R/W	PWMIO 7 (IOA20) Enable	0: Disabled 1: Enabled
[22]	PWM6_EN	R/W	PWMIO 6 (IOA19) Enable	0: Disabled 1: Enabled
[21]	PWM5_EN	R/W	PWMIO 5 (IOA18) Enable	0: Disabled 1: Enabled
[20]	PWM4_EN	R/W	PWMIO 4 (IOA17) Enable	0: Disabled 1: Enabled
[19]	PWM3_EN	R/W	PWMIO 3 (IOA16) Enable	0: Disabled 1: Enabled
[18]	PWM2_EN	R/W	PWMIO 2 (IOA15) Enable	0: Disabled 1: Enabled
[17]	PWM1_EN	R/W	PWMIO 1 (IOA14) Enable	0: Disabled 1: Enabled
[16]	PWM0_EN	R/W	PWMIO 0 (IOA13) Enable	0: Disabled 1: Enabled
[15:9]	-	-	Reserved	-
[8]	SYNC_EN	R/W	PWMIO Synchronization Function When this function is enabled, PWM IO output signal will be delayed for 256 cycles after the duty is changed.	0: Disabled 1: Enabled
[7:4]	CLK_SEL[3:0]	R/W	PWMIO Clock Source Selection	0000 = Not define yet. 0001 = Timer 0 0010 = Timer 1 0011 = Timer 2 0100 = CCP0 0101 = CCP1 0110 = Not define yet. 0111 = Not define yet. 1000 = FCPU/8 1001 = FCPU/16 1010 = FCPU/32 1011 = FCPU/64 1100 = FCPU/256 1101 = FCPU/1024 1110 = FCPU/4096 1111 = Not define yet.
[3:0]	-	-	Reserved	-

PWMIO_CTRL1
0x40160004
PWMIO Control Register 1

Bit	31	30	29	28	27	26	25	24
Function	PWM15_IN V	PWM14_IN V	PWM13_IN V	PWM12_IN V	PWM11_IN V	PWM10_IN V	PWM9_INV	PWM8_INV
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	PWM7_INV	PWM6_INV	PWM5_INV	PWM4_INV	PWM3_INV	PWM2_INV	PWM1_INV	PWM0_INV
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	PWM15_M SK	PWM14_M SK	PWM13_M SK	PWM12_M SK	PWM11_M SK	PWM10_M SK	PWM9_MS K	PWM8_MS K
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PWM7_MS K	PWM6_MS K	PWM5_MS K	PWM4_MS K	PWM3_MS K	PWM2_MS K	PWM1_MS K	PWM0_MS K
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description			Condition
[31]	PWM15_INV	R/W	Inverse PWMIO 15			0: Inverse OFF 1: Inverse ON
[30]	PWM14_INV	R/W	Inverse PWMIO 14			0: Inverse OFF 1: Inverse ON
[29]	PWM13_INV	R/W	Inverse PWMIO 13			0: Inverse OFF 1: Inverse ON
[28]	PWM12_INV	R/W	Inverse PWMIO 12			0: Inverse OFF 1: Inverse ON
[27]	PWM11_INV	R/W	Inverse PWMIO 11			0: Inverse OFF 1: Inverse ON
[26]	PWM10_INV	R/W	Inverse PWMIO 10			0: Inverse OFF 1: Inverse ON
[25]	PWM9_INV	R/W	Inverse PWMIO 9			0: Inverse OFF 1: Inverse ON
[24]	PWM8_INV	R/W	Inverse PWMIO 8			0: Inverse OFF 1: Inverse ON
[23]	PWM7_INV	R/W	Inverse PWMIO 7			0: Inverse OFF 1: Inverse ON
[22]	PWM6_INV	R/W	Inverse PWMIO 6			0: Inverse OFF 1: Inverse ON
[21]	PWM5_INV	R/W	Inverse PWMIO 5			0: Inverse OFF 1: Inverse ON
[20]	PWM4_INV	R/W	Inverse PWMIO 4			0: Inverse OFF 1: Inverse ON
[19]	PWM3_INV	R/W	Inverse PWMIO 3			0: Inverse OFF 1: Inverse ON

Bit	Function	Type	Description	Condition
[18]	PWM2_INV	R/W	Inverse PWMIO 2	0: Inverse OFF 1: Inverse ON
[17]	PWM1_INV	R/W	Inverse PWMIO 1	0: Inverse OFF 1: Inverse ON
[16]	PWM0_INV	R/W	Inverse PWMIO 0	0: Inverse OFF 1: Inverse ON
[15]	PWM15_MSK	R/W	MASK PWMIO 15	0: Mask OFF 1: Mask ON
[14]	PWM14_MSK	R/W	MASK PWMIO 14	0: Mask OFF 1: Mask ON
[13]	PWM13_MSK	R/W	MASK PWMIO 13	0: Mask OFF 1: Mask ON
[12]	PWM12_MSK	R/W	MASK PWMIO 12	0: Mask OFF 1: Mask ON
[11]	PWM11_MSK	R/W	MASK PWMIO 11	0: Mask OFF 1: Mask ON
[10]	PWM10_MSK	R/W	MASK PWMIO 10	0: Mask OFF 1: Mask ON
[9]	PWM9_MSK	R/W	MASK PWMIO 9	0: Mask OFF 1: Mask ON
[8]	PWM8_MSK	R/W	MASK PWMIO 8	0: Mask OFF 1: Mask ON
[7]	PWM7_MSK	R/W	MASK PWMIO 7	0: Mask OFF 1: Mask ON
[6]	PWM6_MSK	R/W	MASK PWMIO 6	0: Mask OFF 1: Mask ON
[5]	PWM5_MSK	R/W	MASK PWMIO 5	0: Mask OFF 1: Mask ON
[4]	PWM4_MSK	R/W	MASK PWMIO 4	0: Mask OFF 1: Mask ON
[3]	PWM3_MSK	R/W	MASK PWMIO 3	0: Mask OFF 1: Mask ON
[2]	PWM2_MSK	R/W	MASK PWMIO 2	0: Mask OFF 1: Mask ON
[1]	PWM1_MSK	R/W	MASK PWMIO 1	0: Mask OFF 1: Mask ON
[0]	PWM0_MSK	R/W	MASK PWMIO 0	0: Mask OFF 1: Mask ON

Note: When 0x0004[15:0] is enabled, PWM IO counter will be reset and re-count again. In addition, when PWM IO is in input mode, there won't be PWM output but the counter will keep counting.

PWMIO_TOGGLE_CTRL 0x40160008 PWMIO IO Toggle Control Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	TOGGLE15_EN	TOGGLE14_EN	TOGGLE13_EN	TOGGLE12_EN	TOGGLE11_EN	TOGGLE10_EN	TOGGLE9_E_N	TOGGLE8_E_N
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	TOGGLE7_E_N	TOGGLE6_E_N	TOGGLE5_E_N	TOGGLE4_E_N	TOGGLE3_E_N	TOGGLE2_E_N	TOGGLE1_E_N	TOGGLE0_E_N
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	-	Reserved	-
[15]	TOGGLE15_EN	R/W	PWMIO 15 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[14]	TOGGLE14_EN	R/W	PWMIO 14 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[13]	TOGGLE13_EN	R/W	PWMIO 13 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[12]	TOGGLE12_EN	R/W	PWMIO 12 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[11]	TOGGLE11_EN	R/W	PWMIO 11 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[10]	TOGGLE10_EN	R/W	PWMIO 10 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[9]	TOGGLE9_EN	R/W	PWMIO 9 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[8]	TOGGLE8_EN	R/W	PWMIO 8 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[7]	TOGGLE7_EN	R/W	PWMIO 7 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[6]	TOGGLE6_EN	R/W	PWMIO 6 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[5]	TOGGLE5_EN	R/W	PWMIO 5 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled
[4]	TOGGLE4_EN	R/W	PWMIO 4 with 50% of Duty Cycle Output Enable	0: Disabled 1: Enabled

Bit	Function	Type	Description				Condition
[3]	TOGGLE3_EN	R/W	PWMIO 3 with 50% of Duty Cycle Output Enable				0: Disabled 1: Enabled
[2]	TOGGLE2_EN	R/W	PWMIO 2 with 50% of Duty Cycle Output Enable				0: Disabled 1: Enabled
[1]	TOGGLE1_EN	R/W	PWMIO 1 with 50% of Duty Cycle Output Enable				0: Disabled 1: Enabled
[0]	TOGGLE0_EN	R/W	PWMIO 0 with 50% of Duty Cycle Output Enable				0: Disabled 1: Enabled

PWMIO_PERIOD_CTRL **0x4016000C** **PWMIO Period Control Register**

Bit	31	30	29	28	27	26	25	24
Function	PERIOD15[1:0]		PERIOD14[1:0]		PERIOD13[1:0]		PERIOD12[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	PERIOD11[1:0]		PERIOD10[1:0]		PERIOD9[1:0]		PERIOD8[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	PERIOD7[1:0]		PERIOD6[1:0]		PERIOD5[1:0]		PERIOD4[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PERIOD3[1:0]		PERIOD2[1:0]		PERIOD1[1:0]		PERIOD0[1:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:30]	PERIOD15[1:0]	R/W	PWMIO 15 Period Setting				00 = 256 01 = 128 10 = 64 11 = 32
[29:28]	PERIOD14[1:0]	R/W	PWMIO 14 Period Setting				00 = 256 01 = 128 10 = 64 11 = 32
[27:26]	PERIOD13[1:0]	R/W	PWMIO 13 Period Setting				00 = 256 01 = 128 10 = 64 11 = 32
[25:24]	PERIOD12[1:0]	R/W	PWMIO 12 Period Setting				00 = 256 01 = 128 10 = 64 11 = 32
[23:22]	PERIOD11[1:0]	R/W	PWMIO 11 Period Setting				00 = 256 01 = 128 10 = 64 11 = 32

Bit	Function	Type	Description	Condition
[21:20]	PERIOD10[1:0]	R/W	PWMIO 10 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[19:18]	PERIOD9[1:0]	R/W	PWMIO 9 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[17:16]	PERIOD8[1:0]	R/W	PWMIO 8 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[15:14]	PERIOD7[1:0]	R/W	PWMIO 7 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[13:12]	PERIOD6[1:0]	R/W	PWMIO 6 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[11:10]	PERIOD5[1:0]	R/W	PWMIO 5 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[9:8]	PERIOD4[1:0]	R/W	PWMIO 4 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[7:6]	PERIOD3[1:0]	R/W	PWMIO 3 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[5:4]	PERIOD2[1:0]	R/W	PWMIO 2 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[3:2]	PERIOD1[1:0]	R/W	PWMIO 1 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32
[1:0]	PERIODO[1:0]	R/W	PWMIO 0 Period Setting	00 = 256 01 = 128 10 = 64 11 = 32

PWMIO_PWMIO0_Duty
0x40160010
PWMIO0 Duty Register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM0_Duty[7:0]	R/W	Duty Control for PWMIO 0 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO1_Duty
0x40160014
PWMIO1 Duty Register

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM1_Duty[7:0]	R/W	Duty Control for PWMIO 1 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO2_Duty
0x40160018
PWMIO2 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM2_Duty[7:0]	R/W	Duty Control for PWMIO 2 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO3_Duty
0x4016001C
PWMIO3 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM3_Duty[7:0]	R/W	Duty Control for PWMIO 3 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO4_Duty
0x40160020
PWMIO4 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM4_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM4_Duty[7:0]	R/W	Duty Control for PWMIO 4 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO5_Duty
0x40160024
PWMIO5 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM5_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM5_Duty[7:0]	R/W	Duty Control for PWMIO 5 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO6_Duty
0x40160028
PWMIO6 Duty Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM6_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM6_Duty[7:0]	R/W	Duty Control for PWMIO 6 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO7_Duty
0x4016002C
PWMIO7 Duty Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM7_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM7_Duty[7:0]	R/W	Duty Control for PWMIO 7 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO8_Duty
0x40160030
PWMIO8 Duty Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM8_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM8_Duty[7:0]	R/W	Duty Control for PWMIO 8 Write = Duty Control Read = Represent Duty						-

PWMIO_PWMIO9_Duty
0x40160034
PWMIO9 Duty Register

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM9_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:8]	-	R	Reserved						-
[7:0]	PWM9_Duty[7:0]	R/W	Duty Control for PWMIO 9 Write = Duty Control Read = Represent Duty						-



GPCM3 SERIES PROGRAMMING GUIDE

PWMIO_PWMIO10_Duty		0x40160038					PWMIO10 Duty Register		
Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	
Bit	23	22	21	20	19	18	17	16	
Function	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	
Function	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
Function	PWM10_Duty[7:0]								
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[31:8]	-	R	Reserved	-
[7:0]	PWM10_Duty[7:0]	R/W	Duty Control for PWMIO 10 Write = Duty Control Read = Represent Duty	-

PWMIO_PWMIO11_Duty								0x4016003C	PWMIO11 Duty Register			
Bit	31	30	29	28	27	26	25	24				
Function									-			
Reset Value	0	0	0	0	0	0	0	0				
Bit	23	22	21	20	19	18	17	16				
Function									-			
Reset Value	0	0	0	0	0	0	0	0				
Bit	15	14	13	12	11	10	9	8				
Function									-			
Reset Value	0	0	0	0	0	0	0	0				
Bit	7	6	5	4	3	2	1	0				
Function									PWM11_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0				
Bit	Function		Type	Description					Condition			
[31:8]	-		R	Reserved					-			
[7:0]	PWM11_Duty[7:0]		R/W	Duty Control for PWMIO 11 Write = Duty cControl Read = Represent Duty					-			

PWMIO_PWMIO12_Duty
0x40160040
PWMIO12 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM12_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	PWM12_Duty[7:0]	R/W	Duty Control for PWMIO 12 Write = Duty Control Read = Represent Duty					-

PWMIO_PWMIO13_Duty
0x40160040
PWMIO13 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM13_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	PWM13_Duty[7:0]	R/W	Duty Control for PWMIO 13 Write = Duty Control Read = Represent Duty					-

PWMIO_PWMIO14_Duty
0x40160048
PWMIO14 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM14_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	PWM14_Duty[7:0]	R/W	Duty Control for PWMIO 14 Write = Duty Control Read = Represent Duty					-

PWMIO_PWMIO15_Duty
0x4016004C
PWMIO15 Duty Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PWM15_Duty[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	PWM15_Duty[7:0]	R/W	Duty Control for PWMIO 15 Write = Duty Control Read = Represent Duty					-

PWMIO_OFFSET
0x40160070
PWMIO Period Offset Register

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	PWM14_15_offset[1:0]	PWM12_13_offset[1:0]	PWM10_11_offset[1:0]	PWM8_9_offset[1:0]				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PWM6_7_offset[1:0]	PWM4_5_offset[1:0]	PWM2_3_offset[1:0]	PWM0_1_offset[1:0]				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	-	Reserved	-
[15:14]	PWM14_15_offset[1:0]	R/W	Offset Control for PWMIO 14~15	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.
[13:12]	PWM12_13_offset[1:0]	R/W	Offset Control for PWMIO 12~13	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.
[11:10]	PWM10_11_offset[1:0]	R/W	Offset Control for PWMIO 10~11	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.
[9:8]	PWM8_9_offset[1:0]	R/W	Offset Control for PWMIO 8~9	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.
[7:6]	PWM6_7_offset[1:0]	R/W	Offset Control for PWMIO 6~7	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.
[5:4]	PWM4_5_offset[1:0]	R/W	Offset Control for PWMIO 4~5	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.
[3:2]	PWM2_3_offset[1:0]	R/W	Offset Control for PWMIO 2~3	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.

Bit	Function	Type	Description	Condition
[1:0]	PWM0_1_offset[1:0]	R/W	Offset Control for PWMIO 0~1	0: No delay. 1: Delay 1/4 PWM period. 2: Delay 2/4 PWM period. 3: Delay 3/4 PWM period.

22. Quadrature Decoder

22.1. Introduction

The purpose of quadrature decoder (QD) is to detect the phase differences between two square waves, and it's normally used on rotating devices in order to detect the rotation velocity and position, etc. There are two sets of quadrature decoder in GPCM3 SERIALS series, each comprising two data inputs, control bits, counter and a clear port to reset the counter. User only needs to input two square waves into the quadrature decoder through two input pins and the QD will start running. If one of the waves changes, the quadrature decoder will determine the device is turing forward or backward based on the phase differences and will then change the value in the counter. When QD determines it's forward, the counter value will +1. If it's backward, the counter value will -1. In other words, the counter's range is between -16 and +15. For more information about the determination on forward and backward, please refer to the following section.

If the counter's value changes, the quadrature decoder will determine when to issue an interrupt according to user's settings in QD_CTRL Bit[0] and Bit[2] (QD0_INT_MODE/QD1_INT_MODE). There are two interrupt types provided by QD, all interrupt mode and overflow interrupt mode. When it's in all interrupt mode, an interrupt will be issued when the quadrature decoder detects any forward/backward signal. That is, an interrupt is issued whenever the counter changes. If it's in overflow interrupt mode, an interrupt will be issued in the following two conditions:

- In the case of turning forward, the counter value is greater than forward threshold (QD_FW_TH) value, and Positive Interrupt will be triggered.
- In the case of turning backward, the counter value is less than backward threshold (QD_FW_TH) value. The interrupt will be reserved.

Note:

1. The default value of Forward Threshold is 0x000F (+15), and the default value of Backward Threshold is 0xFFFF (-16, shown as 2's complement).
2. In overflow mode, there're two compare patterns of Counter and Threshold. All 16-bit will be compared in Default mode. But in Original mode, only Threshold Bit[15:2] will be compared with Counter[15:2].

Also, there is a clear port in each quadrature decoder to reset the counter. Writing any data into the clear port will reset the counter as zero. In addition, each quadrature decoder has two IRQ vectors, forward IRQ and backward IRQ. When forward or backward IRQ occurs, CPU will execute the corresponding IRQ subroutine.

22.2. Features

- Support two sets of quadrature decoder.
Quadrature Decoder 1 – INPUT1: IOA18, INPUT2: IOA19
Quadrature Decoder 2 – INPUT1: IOA20, INPUT2: IOA21
- Support 16-bit forward/backward counter.
- Support all interrupt and overflow interrupt mode.
- Support Default and Original Compare mode (Overflow mode only).
- Support input de-bounce filter.

22.3. Function

Quadrature Decoder Diagram

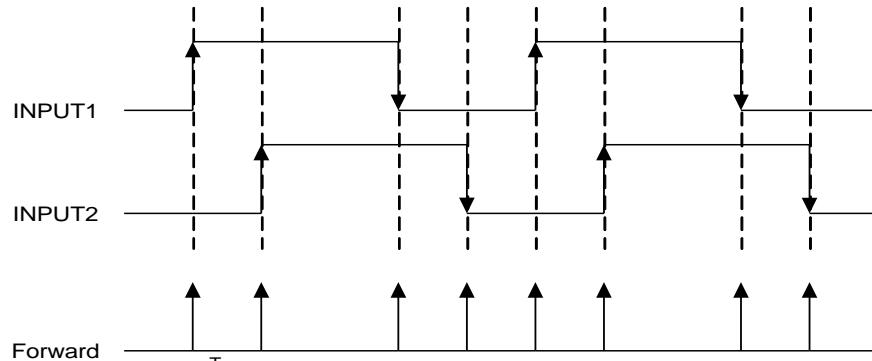
The following diagrams describe the forward/backward determining conditions of the quadrature decoder. Each quadrature decoder has two inputs, and they're summarized as follows:

Note that before starting the quadrature decoder, its corresponding INPUT I/Os must be configured as input mode.

Forward (Positive) Counter:

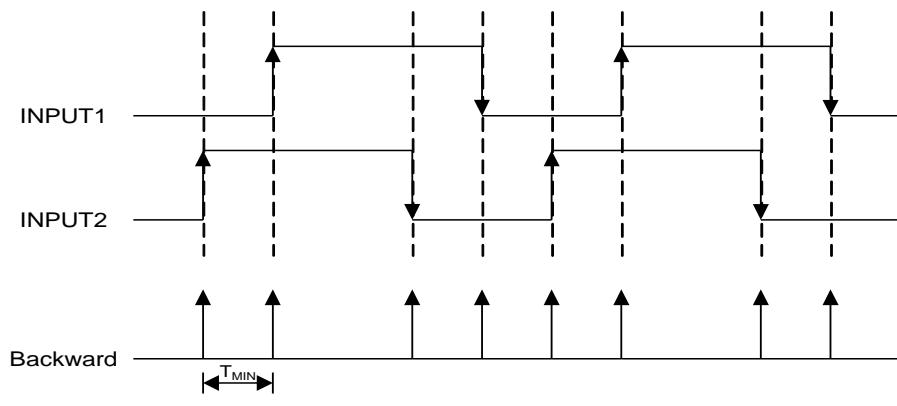
As shown in the following diagram, if INPUT1 signal arrives before INPUT2, the Counter +1.

The Counter is 16-bit resolution, and the range of the positive number is 0x0000 (0) ~ 0x7FFF (32767).


Backward (Reverse) Counter:

As shown in the following diagram, if INPUT2 signal arrives before INPUT1, the Counter -1.

The range of the Counter negative number is 0x8000 (-32768) ~ 0xFFFF (-1) (shown as 2's complement).



Note: T_{MIN} is the minimum interval between two events; it should be greater than $1/(F_{CPU}/2)$.

22.4. Register Description

Register Map

Name	Address	Description
QD_QD0_CNT	0x40160050	Counter of Quadrature Decoder 0
QD_QD1_CNT	0x40160054	Counter of Quadrature Decoder 1
QD_QD0_CLR	0x40160058	Clear Port of QD0 Counter
QD_QD1_CLR	0x4016005C	Clear Port of QD1 Counter
QD_CTRL	0x40160060	Quadrature Decoder Control Register
QD_STS	0x40160064	Quadrature Decoder Status Register
QD_FW_TH	0x40160068	Quadrature Decoder Forward Threshold Register
QD_BW_TH	0x4016006C	Quadrature Decoder Backward Threshold Register

Register Function
QD_QD0_CNT
0x40160050
Counter of Quadrature Decoder 0

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	QD0_CNT[15:0]	R	Counter of Quadrature Decoder 0						-

Note: Write is ineffective.

QD_QD1_CNT
0x40160054
Counter of Quadrature Decoder 1

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	QD1_CNT[15:0]	R	Counter of Quadrature Decoder 1						-

Note: Write is ineffective.

QD_QD0_CLR
0x40160058
Clear Port of QD0 Counter

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function						QD0_CLR[15:8]		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function						QD0_CLR[7:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	QD0_CLR[15:0]	W	Write any data to clear QD0 counter. Write = Clear QD0 counter.						-

QD_QD1_CLR
0x4016005C
Clear Port of QD1 Counter

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function						QD1_CLR[15:8]		
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function						QD1_CLR[7:0]		
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	QD1_CLR[15:0]	W	Write any data to clear QD1 counter. Write = Clear QD1 counter.						-

QD_CTRL									0x40160060	Quadrature Decoder Control Register			
Bit	31	30	29	28	27	26	25	24					
Function					-								
Reset Value	0	0	0	0	0	0	0	0					
Bit	23	22	21	20	19	18	17	16					
Function				-									
Reset Value	0	0	0	0	0	0	0	0					
Bit	15	14	13	12	11	10	9	8					
Function			-		QD1_REV_E N	QD1_POS_E N	QD0_REV_E N	QD0_POS_E N					
Reset Value	0	0	0	0	0	0	0	0					
Bit	7	6	5	4	3	2	1	0					
Function	OVERFLOW _MODE2_E ENABLE		DEBOUNCE_SEL[2:0]		QD1_EN	QD1_INT_ MODE	QD0_EN	QD0_INT_ MODE					
Reset Value	0	0	0	0	0	0	0	0					
Bit	Function	Type	Description					Condition					
[31:12]	-	R	Reserved					-					
[11]	QD1_REV_EN	R/W	QD1 Motor Reserve Interrupt Enable					0: Disabled 1: Enabled					
[10]	QD1_POS_EN	R/W	QD1 Motor Positive Interrupt Enable					0: Disabled 1: Enabled					
[9]	QD0_REV_EN	R/W	QD0 Motor Reserve Interrupt Enable					0: Disabled 1: Enabled					
[8]	QD0_POS_EN	R/W	QD0 Motor Positive Interrupt Enable					0: Disabled 1: Enabled					
[7]	OVERFLOW_MODE2_ENABLE	R/W	QD Overflow Mode Selection					0: Default Mode 1: Original Overflow Mode					
[6:4]	DEBOUNCE_SEL[2:0]	R/W	Quadrature Input Debounce Time Selection					000 = No debounce. 001 = 4T 010 = 8T 011 = 16T 100 = 32T 101 = 40T 110 = 80T 111 = 128T Note: 'T' is the Time unit of the system clock.					
[3]	QD1_EN	R/W	Quadrature Decoder 1 Control Bit					-					
[2]	QD1_INT_MODE	R/W	Quadrature Decoder 1 Interrupt Mode					0: All Interrupt Mode 1: Overflow Interrupt Mode					
[1]	QD0_EN	R/W	Quadrature Decoder 0 Control Bit					0: Disabled 1: Enabled					

Bit	Function	Type	Description				Condition
[0]	QD0_INT_MODE	R/W	Quadrature Decoder 0 Interrupt Mode				0: All Interrupt Mode 1: Overflow Interrupt Mode

QD_STS							
0x40160064							
Bit	31	30	29	28	27	26	25
Function	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-				QD1_REV_I NT_FLAG	QD1_POS_I NT_FLAG	QD0_REV_I NT_FLAG	QD0_POS_I NT_FLAG
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:12]	-	R	Reserved				-
[11]	QD1_REV_INT_FLAG	R/W	QD1 Motor Reverse Interrupt Flag				Read 0: QD1 motor is not reversed. Read 1: QD1 motor reverses. Write 0: No effect. Write 1: Clear this bit.
[10]	QD1_POS_INT_FLAG	R/W	QD1 Motor Positive Interrupt Flag				Read 0: QD1 motor is not positive. Read 1: QD1 motor is positive. Write 0: No effect. Write 1: Clear this bit.
[9]	QD0_REV_INT_FLAG	R/W	QD0 Motor Reverse Interrupt Flag				Read 0: QD0 motor is not reversed. Read 1: QD0 motor reverses. Write 0: No effect. Write 1: Clear this bit.
[8]	QD0_POS_INT_FLAG	R/W	QD0 Motor Positive Interrupt Flag				Read 0: QD0 motor is not positive. Read 1: QD0 motor is positive. Write 0: No effect. Write 1: Clear this bit.
[7:0]	-	R	Reserved				-

QD_FW_TH **0x40160068** **Quadrature Decoder Forward Threshold Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	QD_FW_TH[15:8]	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	QD_FW_TH[7:0]	-	-	-
Reset Value	0	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:0]	QD_FW_TH[15:0]	R/W	<p>Quadrature Counter Forward (Positive) Threshold</p> <p>Note: User can decide to use Original mode or Default mode by configuring QD_CTRL Bit.7 OVER_MODE.</p> <p>Default Mode: When counter[15:0] = QD_FW_TH [15:0], an interrupt will be asserted, and the QD counter will stop counting. The default value of Forward Threshold is 0x000F (+15), and the range of the positive number is 0x0000 (0) ~ 0x7FFF (32767).</p> <p>Original Mode: When counter[15:2] = QD_FW_TH[15:2], interrupt will be asserted, the QD counter will stop counting, and the lowest 2 bits will be ignored.</p>	-

QD_BW_TH **0x4016006C** **Quadrature Decoder Backward Threshold Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	QD_BW_TH[15:8]	-	-	-
Reset Value	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
Function	QD_BW_TH[7:0]							
Reset Value	1	1	1	1	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:0]	QD_BW_TH[15:0]	R/W	<p>Quadrature counter Backward (Reserve) Threshold</p> <p>Note: User can decide to use Original mode or Default mode by configuring QD_CTRL Bit.7 OVER_MODE.</p> <p>Default Mode:</p> <p>When counter[15:0] = QD_BW_TH [15:0], an interrupt will be asserted, and the QD counter will stop counting. The default value of Backward Threshold is 0xFFFF (-16), and the range of the negative number is 0x8000 (-32768) ~ 0xFFFF (-1).</p> <p>Original Mode:</p> <p>When counter[15:2] = QD_BW_TH[15:2], interrupt will be asserted, the QD counter will stop counting, and the lowest 2 bits will be ignored.</p> <p>[15]: Sign Bit</p>	-

23. Capacitive Touch Sensor (CTS)

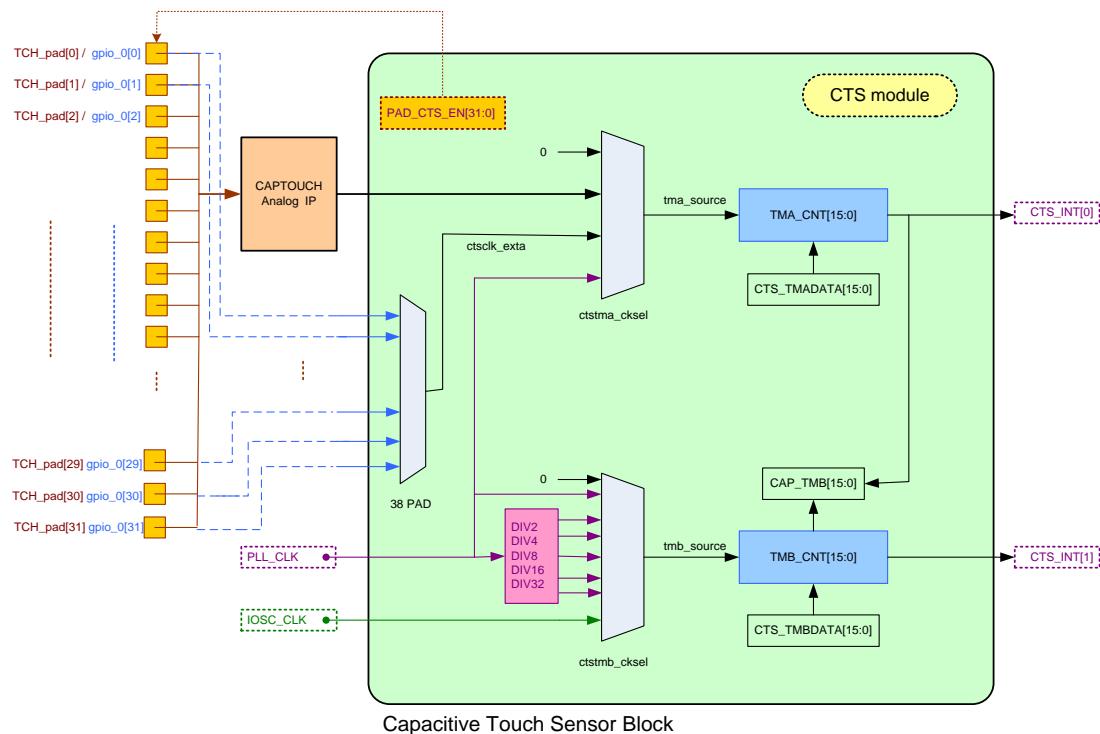
23.1. Introduction

GPCM3 SERIALS contains Capacitive Touch Sensor (CTS) module for related application. User can increase its precision by adjusting charge/discharge current.

23.2. Feature

- Self-Capacitance Touch pads support IOA[31:0], which means there are 32 IO pins in total for Touch pads.
- GPCM3 supports Group IO function.
- Two sets of 16- bit timers with preload function. When CTS function is not in use, CTS timer can be used as general timer.
 - (a) TMA CLK Source can be the CTS Module; or can be input through IO pin.
 - (b) TMB CLK Source can be losc_CLK (12MHz) or the frequency division from the system clock.
- Auto vs. Manual Stop Mode: The main difference of the two modes is whether TMA counter stops automatically when it counts to the Touch charge/discharge time value configured by user.
 - (a) Auto Stop Mode will stop CTS_TMA and TMB counter. For re-count, user has to write 0 first and then write 1 (generating a rising edge) to CTS_CTRL (0x400E_0000) Bit[1].
 - (b) Manual Stop Mode will not automatically stop and re-load TMB Counter Data.

23.3. Block Diagram



23.4. Function

Auto Stop Mode:

Write START_ENABLE = 1, and CTS_TMA/TMB will start counting. An interrupt will be asserted when CTS_TMA overflow happens. The hardware captures CTS_TMB data and save the data to CTS_TMBCAP (0x400E0018), and CTS_TMA/TMB counter will stop automatically. User has to write START_ENABLE = 1 (rising edge trigger) again to re-start counting.

Manual Stop Mode:

Write START_ENABLE = 1, and CTS_TMA/TMB will start counting. An interrupt will be asserted when CTS_TMA overflow happens. The hardware captures CTS_TMB data and save the data to CTS_TMBCAP (0x400E0018), and load CTS_TMBDATA. The Counter won't stop in this mode, so the hardware will have the same reaction when CTS_TMA overflows next time. User has to write 0 to CTS_Ctrl[1] to stop the counting. If CTS_TMA clock frequency doesn't change, CAP_TMB register data should be the same as the value captured last time.

23.5. Register Description
Register Map

Name	Address	Description
CTS_CTRL	0x400E0000	CTS Control Register
CTS_STS	0x400E0004	CTS Interrupt Status Register
CTS_TMAPLOAD	0x400E0008	CTS Preload Register For TimerA
CTS_TMACOUNT	0x400E000C	CTS Counter For TimerA
CTS_TMBUPLOAD	0x400E0010	CTS Preload Register For TimerB
CTS_TMBCOUNT	0x400E0014	CTS Counter For TimerB
CTS_TMBCAP	0x400E0018	CTS Capture Register For TimerB
CTS_PADSEL	0x400E001C	CTS TMA Clock Source Selection

Register Function

CTS_CTRL									0x400E0000		CTS Control Register	
Bit	31	30	29	28	27	26	25	24				
Function	-		TMB_INTEN_SEL	TMA_INTEN_SEL	-	-	-	-				
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function	TMB_CLK_SEL[2:0]				TMA_CLK_SEL[2:0]			TMA_MODE
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	LPF_SEL[1:0]		CHARGE_CURRENT[1:0]				-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-				AUTOSTOP_ENABLE	START_ENA	CTSEN_ENA	BLE
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:30]	-	R	Reserved	-
[29]	TMB_INTEN_SEL	R/W	Timer B Interrupt Enable	0: Disabled 1: Enabled
[28]	TMA_INTEN_SEL	R/W	Timer A Interrupt Enable	0: Disabled 1: Enabled
[27:24]	-	R	Reserved	-

Bit	Function	Type	Description	Condition
[23:21]	TMB_CLK_SEL[2:0]	R/W	TMB Clock Source Selection	000: SYS Clock Off 001: SYS Clock 010: SYS Clock/2 011: SYS Clock/4 100: SYS Clock/8 101: SYS Clock/16 110: SYS Clock/32 111: IOSC Clock 12Mhz
[20]	TMB_MODE	R/W	When this bit is 1, CTS TMB will be used as general timer.	0: Touch Sensor Timer 1: General Timer
[19:17]	TMA_CLK_SEL[2:0]	R/W	CTS TMA Clock Source Selection	000: Clock Off (Default) 001: From IO pad. 010: From touch sensor module. 011: SYS Clock 100: SYS Clock/2 101: SYS Clock/4 110: SYS Clock/8 111: SYS Clock/16
[16]	TMA_MODE	R/W	When this bit is 1, CTS TMA will be used as general timer.	0: Touch Sensor Timer 1: General Timer
[15:14]	LPF_SEL[1:0]	R/W	Low Pass Filter	00: No filter. 01: 1MHz 10/11: 500KHz
[13:12]	CHARGE_CURRENT[1:0]	R/W	Charge/Discharge Current Selection	00: 0uA 01: 25uA 10: 50uA 11: 100uA
[11:3]	-	R	Reserved	-
[2]	AUTOSTOP_ENABLE	R/W	CTS Auto Stop Control Bit In manual stop mode, user can stop CTS_TMA/TMB counter by writing 0x400E_0000 Bit[1] = 0. In auto stop mode, when CTS_TMA overflows, it will stop CTS_TMA/TMB counter automatically. Write 0 first and then write 1 to CTS_CTRL (0x400E_0000) Bit[1], and after a rising edge is generated, the timer will start counting again.	0: Manual Stop Mode 1: Auto Stop Mode
[1]	START_ENABLE	R/W	CTS Start Control Bit	0: Stop CTS_TMA/TMB counter in manual stop mode (0x400E0000[2] = 0).

Bit	Function	Type	Description				Condition		
							1: CTS_TMA/TMB counter starts counting, and it will stop automatically when CTS_TMA overflows.		
[0]	CTSEN_ENABLE	R/W	CTS Enable Control Bit				0: Disabled 1: Enabled		

CTS_STS								
0x400E0004								
CTS Interrupt Status Register								
Bit	31	30	29	28	27	26	25	24
Function	-		TMB_INTF	TMA_INTF		-		
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition		
[31:30]	-	R	Reserved				-		
[29]	TMB_INTF	R/W	CTS Timer B Overflow/Underflow Interrupts Flag				Read 0: Idle/Busy Read 1: Timer interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.		
[28]	TMA_INTF	R/W	CTS Timer A Overflow/Underflow Interrupts Flag				Read 0: Idle/Busy Read 1: Timer interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.		
[27:0]	-	R	Reserved				-		

CTS_TMAPLOAD								
0x400E0008								
CTS Preload Register For TimerA								
Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					TMAPLOAD[15:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					TMAPLOAD[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	TMAPLOAD[15:0]	R/W	Capacitive Touch Sensor TMA Data Register (TMB counter value when TMA = TMAPLOAD.)					-

CTS_TMACOUNT **0x400E000C** **CTS Counter For TimerA**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					TMACOUNT[15:8]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					TMACOUNT[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	-	R	Reserved					-
[15:0]	TMACOUNT[15:0]	R	Capacitive Touch Sensor TMA Count Register					-

CTS_TMBLOAD **0x400E0010** **CTS Preload Register For TimerB**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	TMBDATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	TMBDATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	TMBDATA[15:0]	R/W	Capacitive Touch Sensor TMB Data Register						-

CTS_TMBCOUNT	0x400E0014	CTS Counter For TimerB						
Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	TMBCOUNT[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	TMBCOUNT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:16]	-	R	Reserved						-
[15:0]	TMBCOUNT[15:0]	R	Capacitive Touch Sensor TMB Count Register						-

CTS_TMBCAP	0x400E0018	CTS Capture Register For TimerB						
Bit	31	30	29	28	27	26	25	24
Function	-							

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	TMBCAP[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	TMBCAP[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	-	R	Reserved	-
[15:0]	TMBCAP[15:0]	R	TMB counter's value will be captured and saved in TMBCAP when TMA = TMAPLOAD.	-

CTS_PADSEL **0x400E001C** **CTS TMA Clock Source Selection**

Bit	31	30	29	28	27	26	25	24
Function	CTS_PADSEL[31:24]							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	CTS_PADSEL[23:16]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	CTS_PADSEL[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CTS_PADSEL[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:0]	CTS_PADSEL [31:0]	R/W	<p>1. When CTS_CTRL[19:17] TMA_CLK_SEL = 010, CTS TMA clock source is from touch sensor module. [31:0]: PAD_IO bit determines how many IO will be used (which means it can be one or multiple IOs). Group IO Function: When multiple IOs are enabled, they will be merged for Touch scan.</p> <p>2. When CTS_CTRL[19:17] TMA_CLK_SEL = 001, CTS TMA clock source is from GPIO. PADSEL[5:0]: These bits are used to select the index of external GPIO.</p>	<p>1.TMA_CLK_SEL = 010, PADSEL[31:0] = Bit[0]: IOA0 (0: CTS Pad Disabled, 1: Enabled) Bit[1]: IOA1 (0: CTS Pad Disabled, 1: Enabled) Bit[2]: IOA2 (0: CTS Pad Disabled, 1: Enabled) Bit[3]: IOA3 (0: CTS Pad Disabled, 1: Enabled) ... Bit[30]: IOA30 (0: CTS Pad Disabled, 1: Enabled) Bit[31]: IOA31 (0: CTS Pad Disabled, 1: Enabled)</p>

Bit	Function	Type	Description	Condition
				2.TMA_CLK_SEL = 001, PADSEL[5:0] = 0x00: IOA0 (0: Disable, 1: CLK Src is from IOA.0) 0x01: IOA1 (0: Disable, 1: CLK Src is from IOA.1) 0x02: IOA2 (0: Disable, 1: CLK Src is from IOA.2) ... 0x1E: IOA30 (0: Disable, 1: CLK Src is from IOA.30) 0x1F: IOA31 (0: Disable, 1: CLK Src is from IOA.31)

24. TIMEBASE Control

24.1. Introduction

GPCM3 SERIALS contains TimeBase module for related application. The TimeBase provides a programmable periodic interrupt for sleep and deep sleep mode.

24.2. Feature

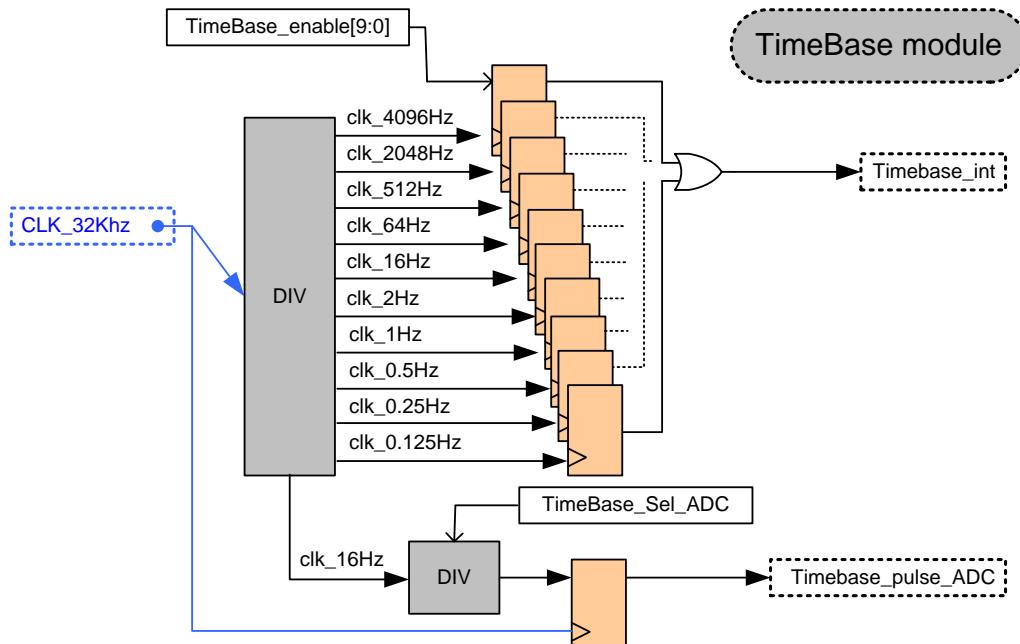
TimeBase module offers the following clock functions:

- Programmable periodic interrupts in 0.125Hz/0.25Hz/0.5Hz/1Hz/2Hz/16Hz/64Hz/512Hz/2048Hz/4096Hz.

24.3. Note

- 32768Hz CLK Source must be enabled for Timebase clock. The CLK source can be Internal 32KHz or External X'TAL 32768Hz.
- Must set CTS_CLK_ENABLE (0x50001010) Bit[10] = 1 (enabling the respective CLK Control Register) ; otherwise, the corresponding register of Timebase cannot be written any value.
- GPCM3 has two levels of interrupt control: M0's internal NVIC interrupt control and GPCM3's INT Control. Both NVIC and GPCM3 INT Control must be enabled in order to make the corresponding INT enabled.
`NVIC_EnableIRQ(TIMEBASE_IRQn);`
- Write 1 to TIMEBASE_CTRL (0x400E0060) Bit[9] (CLRCNT_ENABLE) to make Timebase counter Reset (re-count), and this can assure the interrupt frequency stays the same every time.
- When GPCM3 Timebase CLK source is switched from losc32K to XTAL32K, user can know the status of XTAL32K by checking CLK_STS(0x50001050)'s bit[3]. However, when X32K STS Flag = 1, user can only assure CLK Src has been switched from losc32K to XTAL32K but XTAL may not be stable at this time. To assure the system operates normally, it is recommended to wait for one or two additional 32KHz clocks before user turns off losc 32K. For more information about example codes, please refer to the settings in TimeBase_Initial at pack\Timebase example code.

24.4. Block Diagram





GPCM3 SERIES PROGRAMMING GUIDE

24.5. Function

Normal TimeBase Mode

Write TimeBase_enable = 1, then 14bit timebase counter will start counting. The counter uses the low-speed clock (LSIRC32K), so it can continue counting when the system enters sleep/deep sleep mode. An interrupt will be asserted when the counter reaches the related setting (TimeBase_INTEEn[5:0]). At this time, the counter won't stop but keep counting. The hardware will assert a programmable periodic interrupt when the timebase counter overflows the next time. User has to write TimeBase_enable = 0 to stop the counting.

24.6. Register Description

Register Map

Name	Address	Description
TIMEBASE_CTRL	0x400E0060	TimeBase Control Register
TIMEBASE_STS	0x400E0064	TimeBase Interrupt Status Register

Register

Bit	Function	Type	Description	Condition
[31:18]	-	R	Reserved	-
[17]	CLRCNT_ENABLE	WOC	TimeBase Clear Counter Bit	0: No effect. 1: Reset the counter as 0.
[16]	TB_ENABLE	R/W	TimeBase Enable	0: Disabled 1: Enabled
[15:10]	-	R	Reserved	-
[9]	4KHZ_ENABLE	R/W	Timebase 4096Hz Interrupt Enable	0: Disabled 1: Enabled
[8]	2KHZ_ENABLE	R/W	Timebase 2048Hz Interrupt Enable	0: Disabled 1: Enabled
[7]	512HZ_ENABLE	R/W	Timebase 512Hz Interrupt Enable	0: Disabled 1: Enabled

Bit	Function	Type	Description				Condition
[6]	64HZ_ENABLE	R/W	Timebase 64Hz Interrupt Enable				0: Disabled 1: Enabled
[5]	16HZ_ENABLE	R/W	Timebase 16Hz Interrupt Enable				0: Disabled 1: Enabled
[4]	2HZ_ENABLE	R/W	Timebase 2Hz Interrupt Enable				0: Disabled 1: Enabled
[3]	1HZ_ENABLE	R/W	Timebase 1Hz Interrupt Enable				0: Disabled 1: Enabled
[2]	0.5HZ_ENABLE	R/W	Timebase 0.5Hz Interrupt Enable				0: Disabled 1: Enabled
[1]	0.25HZ_ENABLE	R/W	Timebase 0.25Hz Interrupt Enable				0: Disabled 1: Enabled
[0]	0.125HZ_ENABLE	R/W	Timebase 0.125Hz Interrupt Enable				0: Disabled 1: Enabled

TIMEBASE_STS **0x400E0064** **TimeBase Interrupt Status Register**

Bit	31	30	29	28	27	26	25	24
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	4KHZ_INTF_FLAG	2KHZ_INTF_FLAG
Reset Value	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Function	512HZ_INTF_FLAG	64HZ_INTF_FLAG	16HZ_INTF_FLAG	2HZ_INTF_FLAG	1HZ_INTF_FLAG	0.5HZ_INTF_FLAG	0.25HZ_INTF_FLAG	0.125HZ_INTF_FLAG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:10]	-	R	Reserved				-
[9]	4KHZ_INTF_FLAG	R/W	Timebase 4096Hz Interrupt Flag				Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[8]	2KHZ_INTF_FLAG	R/W	Timebase 2048Hz Interrupt Flag				Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered.

Bit	Function	Type	Description	Condition
				Write 0: No effect. Write 1: Clear this bit.
[7]	512HZ_INTF_FLAG	R/W	Timebase 512Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[6]	64HZ_INTF_FLAG	R/W	Timebase 64Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[5]	16HZ_INTF_FLAG	R/W	Timebase 16Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[4]	2HZ_INTF_FLAG	R/W	Timebase 2Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[3]	1HZ_INTF_FLAG	R/W	Timebase 1Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[2]	0.5HZ_INTF_FLAG	R/W	Timebase 0.5Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.
[1]	0.25HZ_INTF_FLAG	R/W	Timebase 0.25Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered.

Bit	Function	Type	Description	Condition
				Write 0: No effect. Write 1: Clear this bit.
[0]	0.125HZ_INTF_FLAG	R/W	Timebase 0.125Hz Interrupt Flag	Read 0: No interrupt is triggered. Read 1: Timebase interrupt is triggered. Write 0: No effect. Write 1: Clear this bit.

25. Watchdog Timer

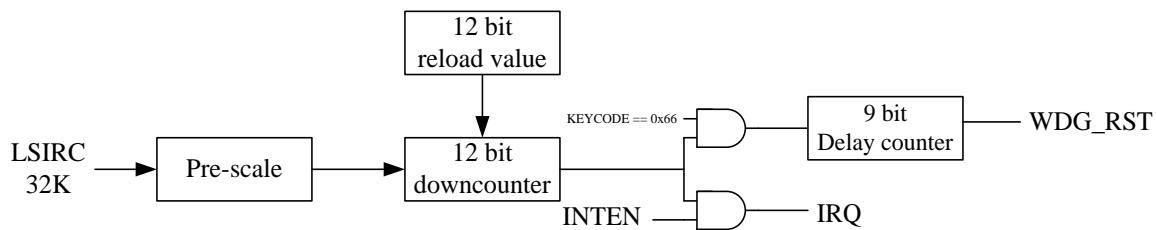
25.1. Introduction

The independent watchdog (WDG) uses the low-speed clock (LSIRC32K) and thus stays active even if the main clock fails. WDG is suitable for monitoring program which can prevent the main program from entering infinite loop.

25.2. Features

- Free-Running Down-Counter
- Use an independent low-speed clock (LSIRC32K) as the clock source.
- Reset (if watchdog is activated) when the down-counter value of 0x000 is reached.

25.3. Block Diagram



25.4. Function

When user write 0x99 or 0x66 to Key register (WDG_KEYCODE.KEYCODE), WDT will be activated, and start counting down from reset value, 0xFFFF. When it reaches the end of count value (0x000), a reset signal will be generated (WDG reset).

Whenever the key value 0xAA is written to the **KEYCODE** register, the **WDG_CTRL.RCNT** value will be reloaded to the counter preventing the trigger of watchdog reset.

Register Access Protection

Write access to the **WDG_CTRL** register is protected. To modify it, user must first write the code 0x55 in the **WDG_CTRL** register. A write access to this register with a different value will break the sequence, and register access will be protected again.

Example:

```
void WDT_SetClk(uint32_t ClkSel)
{
    WDG->KEYCODE = WDG_KEYCODE_ACCESS_ENABLE;
    MODIFY_REG(WDG->CTRL, WDG_CTRL_CLK_SEL_MSK, ClkSel);
    WDG->KEYCODE = WDG_KEYCODE_ACCESS_DISABLE;
}
```

The following figure implies the case of reload operation (writing 0xAA).

```
void WDT_Clear()
{
    WDG->KEYCODE = WDG_KEYCODE_WDG_CLEAR;
}
```



GPCM3 SERIES PROGRAMMING GUIDE

25.5. Register Description

Register Map

Name	Address	Description
WDG_KEYCODE	0x400D0000	Watchdog Key Code Register
WDG_CTRL	0x400D0004	Watchdog Control Register
WDG_STS	0x400D0008	Watchdog Status Register

Register Function

WDG_KEYCODE

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	-	R	Reserved	-
[7:0]	KEYCODE[7:0]	R/W	Watchdog Key Code	<p>00 = Disable Watchdog function.</p> <p>AA = Reload the watchdog down counter value from WDG_CTRL.RCNT.</p> <p>55 = Enable access to WDG_KEYCODE/WDG_CTRL.</p> <p>44 = Disable access to WDG_KEYCODE/WDG_CTRL.</p> <p>99 = Start the Watchdog counting without reset function.</p> <p>66 = Start the Watchdog counting with reset function.</p> <p>Note: If illegal command reset function (WDG_CTRL Bit.12 = 1) is enabled, when an illegal CMD is issued, Watchdog illegal CMD reset will be triggered immediately.</p>

WDG_CTRL **0x400D0004** **Watchdog Control Register**

Bit	31	30	29	28	27	26	25	24
Function	-				CNT[11:8]			
Reset Value	0	0	0	0	1	1	1	1

Bit	23	22	21	20	19	18	17	16
Function	-				CNT[7:0]			
Reset Value	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Function	-			ILL_RST_EN	-	CLK_SEL[2:0]		
Reset Value	0	0	0	1	0	0	1	0

Bit	7	6	5	4	3	2	1	0
Function	-					TIMEOUT_F	LAG	
Reset Value	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[31:28]	-	R	Reserved	-
[27:16]	CNT[11:0]	R/W	Watchdog Counter Reload Value	-
[15:13]	-	R	Reserved	-
[12]	ILL_RST_EN	R/W	Illegal Command Reset Function	0: Disabled 1: Enabled
[11]	-	R	Reserved	-
[10:8]	CLK_SEL[2:0]	R/W	Watchdog Clock Pre-Scaler	000 = LSIRC/1 001 = LSIRC/4 010 = LSIRC/8 011 = LSIRC/16 100 = LSIRC/32 101 = LSIRC/64 110 = LSIRC/128 111 = LSIRC/256
[7:2]	-	R	Reserved	-
[1]	TIMEOUT_FLAG	R/W	Watchdog Timer Timeout Flag Note: It's cleared by the software writing 1.	0: RCNT doesn't down count to 0. 1: RCNT down counts to 0.
[0]	-	R	Reserved	-

WDG_STS **0x400D0008** **Watchdog Status Register**

Bit	31	30	29	28	27	26	25	24
Function	-				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-				-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-			ILL_RESET_F LAG	-
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:2]	-	R	Reserved	-
[1]	ILL_RESET_FLAG	R/W	Watchdog Illegal Access Reset Flag Write 1 to clear this bit. If user writes illegal command to Key register (WDG_KEYCODE.KEYCODE) and turn on illegal command reset function (0x400D0004 Bit[12] = 1), Watchdog illegal CMD reset will be triggered. At this time, this flag = 1.	Read 0: Illegal reset doesn't happen. Read 1: Illegal reset happens. Write 0: No operation. Write 1: Clear this flag.
[0]	-	R	Reserved	-

26. Sound Process Unit Control Register

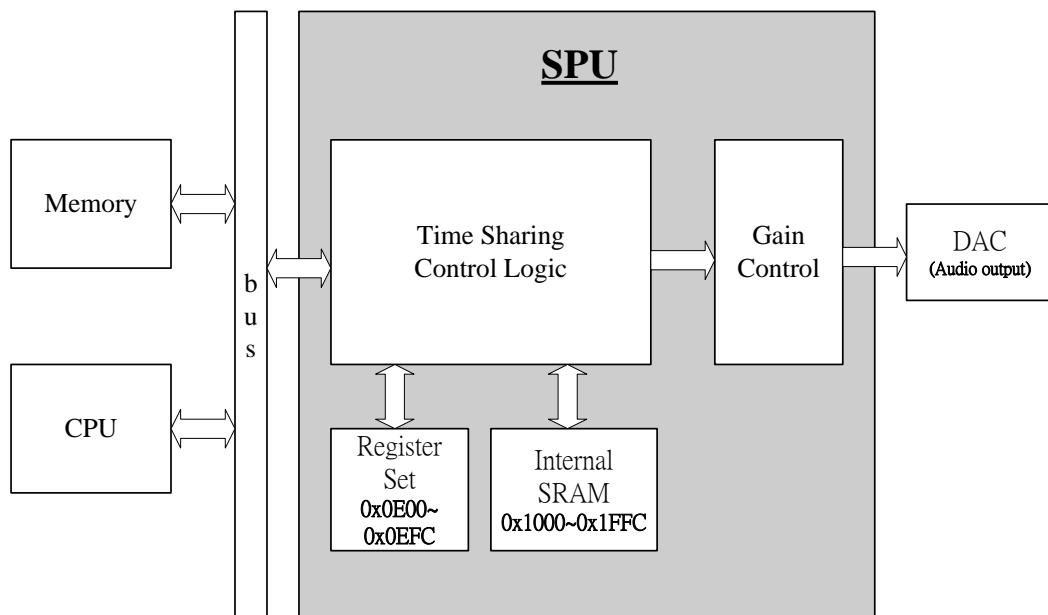
26.1. General Description

The SPU (Sound Process Unit) inside the GPCM3 series is designed to emulate various types of musical instruments and control the envelope slope of each channel. Besides playing Midi, each channel can also be configured as a speech channel to produce PCM-format sound effects, e.g. percussion, animal sounds, gun, explosions accompanied with the main music rhythm.

26.2. Feature

- 8-bit/16-bit (Software Mode) Stereo PCM
- 4-bit ADPCM/ADPCM36
- 32 Channels
- Up to 2048M x 16-bit addressing is supported.
- 7-bit Main Volume Control
- MIDI format gain control for each R/L channel of all 32 channels.
- Envelope control can be divided up to 256 piece-wise, and supports repeat function.
- 32-Channel IRQ Functions and Beat Event IRQ and Envelope IRQ are supported.
- Tone-Color and envelope can be controlled by the hardware or manual (software) mode.
- Auto Tone-Color Interpolation Function
- Up to 287.425kHz tone-color sample rate.
- Hardware Release Tone Color Function
- Individual Channel Release Function
- Hardware Pitch Bend Function
- Auto Volume Control (Compressor)

26.3. Block Diagram



26.4. Internal Memory Mapping

- SPU Base Address: 0x50080000
- Register Set: 0x0E00 ~ 0x0E7C (Channel 0~15), 0x0E80 ~ 0x0EFC (Channel 16~31)
- Internal SRAM: 32 Channels

Figure 26-2 Internal Memory Mapping

Channel	Phase Address Port (Hex)	Attribute Address Port (Hex)
0	3000~303C	1000~103C
1	3040~307C	1040~107C
2	3080~30BC	1080~10BC
3	30C0~30FC	10C0~10FC
4	3100~313C	1100~113C
5	3140~317C	1140~117C
6	3180~31BC	1180~11BC
7	31C0~31FC	11C0~11FC
8	3200~323C	1200~123C
9	3240~327C	1240~127C
10	3280~32BC	1280~12BC
11	32C0~32FC	12C0~12FC
12	3300~333C	1300~133C
13	3340~337C	1340~137C
14	3380~33BC	1380~13BC
15	33C0~33FC	13C0~13FC
16	3400~343C	1400~143C
17	3440~347C	1440~147C
18	3480~34BC	1480~14BC
19	34C0~34FC	14C0~14FC
20	3500~353C	1500~153C
21	3540~357C	1540~157C
22	3580~35BC	1580~15BC
23	35C0~35FC	15C0~15FC
24	3600~363C	1600~163C
25	3640~367C	1640~167C
26	3680~36BC	1680~16BC
27	36C0~36FC	16C0~16FC
28	3700~373C	1700~173C
29	3740~377C	1740~177C
30	3780~37BC	1780~17BC
31	37C0~37FC	17C0~17FC

26.5. Gain Control

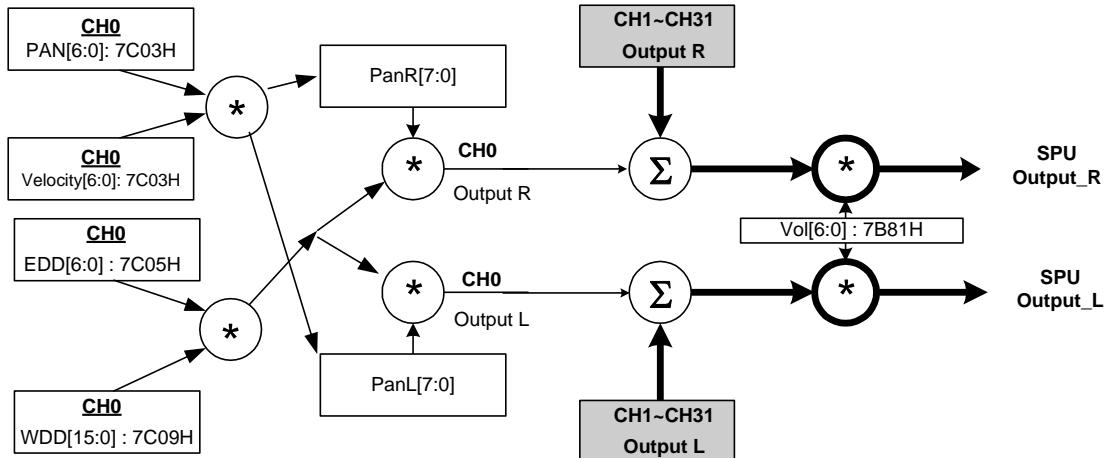


Figure 26-3 SPU Structure Diagram

The SPU features 32 channels. The above structure diagram only illustrates the Channel 0. For other channels, the structure is similar. In the diagram, the multiplication outcome of tone-color (timber) and envelope is delivered to each R/L channel to complete the panning effect. After data of all channels are added up, it will be output via the main volume control.

26.6. Fundamentals

Sound is composed of three essential elements: **pitch**, **tone-color (timbre)**, and **envelope/ADSR**. The tone color can be divided into musical sound and noise in digital music.

Musical Sound: With characteristics of cycles and vibration in certain pattern, e.g. piano, xylophone, violin, etc.

Noise: With the characteristic of vibration in random, e.g. animal sound.

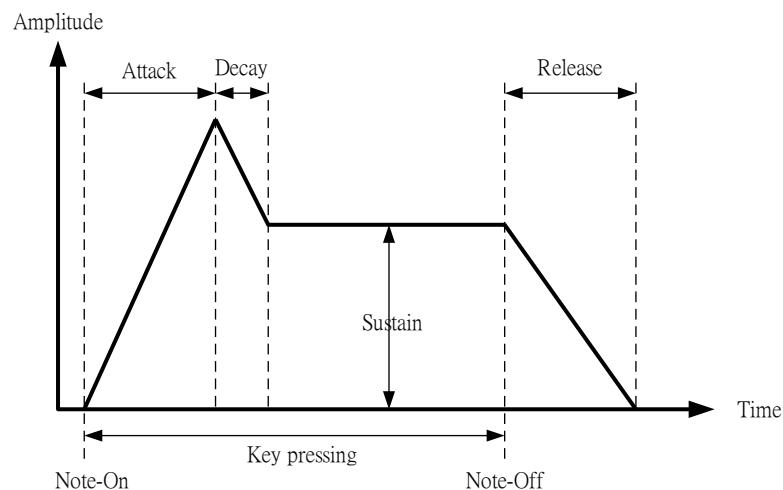


Figure 26-4 ADSR Envelope

26.7. Musical Sound

Because musical sound is a cycled wave, it is possible to be synthesized by repetition (must be supported by the hardware). The instrument can be expressed by envelope and tone-color. An envelope basically contains four major sections: Attack, Decay, Sustain, and Release, or **ADSR**, please refer to the waveform above. The following diagram indicates the modulation outcome from tone-color and envelope.

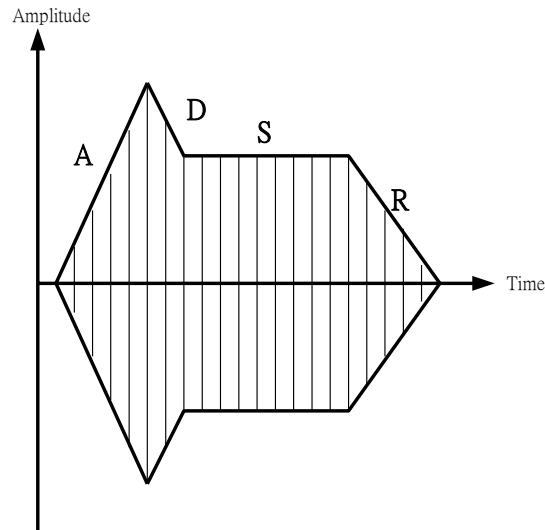
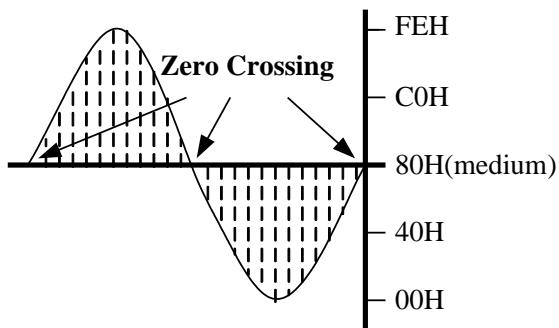


Figure 26-5 ADSR's Waveform

An envelope contains a series of repeating tone-color wave. The ADSR illustrated here is only one of all acoustical models. Not all instruments are produced by this mode.

- 8 bits tone-color code is expressed in the format of unsigned. The medium is 0x80; max is 0xFE, and min is 0x00. "0xFF" is defined as the end code of tone-color; therefore, data itself cannot be "0xFF".



Note: FFH defined as the end code of tone-Color

Figure 26-6 Sin Wave of Tone-Color



GPCM3 SERIES PROGRAMMING GUIDE

- The envelope data is expressed in the format of 7 bits unsigned. The maximum value is 0x7F, and the minimum is 0x00.
- Two envelop modes are supported: auto and manual mode.
 - a. **Auto Mode:** The CPU writes specific parameters to **register set** and **internal SRAM**, and the hardware will load tone-color and envelope data automatically based on the given parameters. The envelope slope of an instrument can be up to 128 types. In addition, it supports repeat function, which can be used to synthesize envelope LFO (Low Frequency Oscillation) easily.
 - b. **Envelope Manual Mode:** The envelope data is controlled by the software.

26.8. Control Register List

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description		
0xOE00																	ChEn[15:0]	Channel Enable	
0xOE04					--												Vol[6:0]	MAIN_VOLUME	
0xOE08																	ChFIQEn[15:0]	Channel FIQ Enable	
0xOE0C																	ChFIQSts[15:0]	Channel FIQ Status	
0xOE10			--														BeatBaseCnt[10:0]	Beat Base Counter	
0xOE14	BIE	BIS															BeatCnt[13:0]	Beat Counter	
0xOE18																	EnvClk[15:0], Ch 3~0	Envelope Interval Select	
0xOE1C																	EnvClk[31:16], Ch 7~4	Envelope Interval Select	
0xOE20									--										
0xOE24									--										
0xOE28																	EnvRampDown[15:0]	Envelope Fast Ramp Down	
0xOE2C																	ChStopSts[15:0]	Stop Channel Status	
0xOE30										--									
0xOE34	Saturate		--		CompEn	NoHigh	NoInt	--		VolSel	FOF	--	Init	--	--			Control Flags	
0xOE38	Peak									AttScale		RelScale		DisZC			Ratio		Compressor Control
0xOE3C																	ChSts[15:0]	Channel Status	
0xOE40										--									
0xOE44										--									
0xOE48											{ WaveOutR[15:0] : WaveOutL[15:0] }							R/L Mixer Output, 32bit Data	
0xOE4C											--						-		



GPCM3 SERIES PROGRAMMING GUIDE

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description	
0x0E50	ChRepeatEn[15:0]																Channel Repeat Enable Control	
0x0E54	ChEnvMode[15:0]																Channel Env Mode	
0x0E58	ChToneRelease[15:0]																Channel Tone Release Control	
0x0E5C	ChEnvIrqSts[15:0]																Channel Env Irq Status	
0x0E60	ChPitchBendEn[15:0]																Channel Pitch Bend Enable	
0x0E64	--																	
0x0E68	AttackTime								ReleaseTime									Attack/Release Time Control
0x0E6C	--																	
0x0E70	--																	
0x0E74	--																	
0x0E78	--																	
0x0E7C									BankAddr[8:0]									Wave Table's Bank Address

Note: All control registers are active-high.

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description
0x0E80	ChEn[31:16]																Channel Enable
0x0E84	-																-
0x0E88	ChFIQEn[31:16]																Channel FIQ Enable
0x0E8C	ChFIQSts[31:16]																Channel FIQ Status
0x0E90	--																--
0x0E94	PWEN	PWDS	--	PWCK	PWIE	PWIS	--	PWSE	PWSIL	--	--	--	--	PWF0V	Post Wave Counter and Control		



GPCM3 SERIES PROGRAMMING GUIDE

Address	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Description	
0x0E98	EnvClk[79:64], Ch 19~16																	Envelope Interval Select
0x0E9C	EnvClk[95:80], Ch 23~20																	Envelope Interval Select
0x0EA0	EnvClk[111:96], Ch 27~24																	Envelope Interval Select
0x0EA4	EnvClk[127:112], Ch 31~28																	Envelope Interval Select
0x0EA8	EnvRampDown[31:16]																	Envelope Fast Ramp Down
0x0EAC	ChStopSts[31:16]																	Stop Channel Status
0x0EB0	--																	
0x0EB4	--																	
0x0EB8	--																	
0x0EBC	ChSts[31:16]																	Channel Status
0x0EC0	--																	
0x0EC4	--																	
0x0EC8	--																	
0x0ECC	{ PostWaveOutR[15:0] : PostWaveOutR[15:0] }																	Post Wave Output
0x0EC0	ChRepeatEn[31:16]																	Channel Repeat Enable Control
0x0EC4	ChEnvMode[31:16]																	Channel Env Mode
0x0EC8	ChToneRelease[31:16]																	Channel Tone Release Control
0x0ECC	ChEnvIrqSts[31:16]																	Channel Env Irq Status
0x0EE0	ChPitchBendEn[31:16]																	Channel Pitch Bend Enable

26.9. Internal SRAM List

Internal Attribute SRAM Format, Channel 0 ~ Channel 32

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0																	
0x1000+0x40*X	Wave Address	Waddr[15:0]																																
0x1004+0x40*X	Mode	ADPCM	16M	ToneMode	LoopAddr[21:16]										Waddr[21:16]																			
0x1008+0x40*X	Loop Address	LoopAddr[15:0]																																
0x100C+0x40*X	Pan	-	Pan[6:0]						-	ChVolumn[6:0]																								
0x1010+0x40*X	Envelope0	Repeat Period	EnvTarget[6:0]						EnvSign	EnvInc[6:0]																								
0x1014+0x40*X	Envelope Data	EnvCnt[7:0]						-	EDD[6:0]																									
0x1018+0x40*X	Envelope1	RpCnt				Rpt	EnvLoad[7:0]																											
0x101C+0x40*X	Envelope Address	IrqFireAddress[8:0]								IrqEn	Eaddr[21:16]																							
0x1020+0x40*X	Envelope Address	Eaddr[15:0]																																
0x1024+0x40*X	Wave Data 0	WDD0[15:0]																																
0x1028+0x40*X	Envelope Loop Control	RampDownoffset[6:0]						Eaoffset[8:0]																										
0x102C+0x40*X	Wave Data	WDD[15:0]																																
0x1030+0x40*X	-																																	
0x1034+0x40*X	ADPCM Sel	ADPCM36	PointNumber																															
0x1038+0x40*X	Wave Loop Address HI		Waddr[30:28]			LoopAddr[27:22]						Waddr[27:22]																						
0x103C+0x40*X	Envelope Address HI	-						Eaddr[30:28]	Eaddr[27:22]																									



GPCM3 SERIES PROGRAMMING GUIDE

Internal Phase SRAM Channel Format, Channel 0 ~ Channel 32

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0x3000+0x40*X	Phase High																Phase[18:16]
0x3004+0x40*X	Phase Accumulator High																PhaseAcc[18:16]
0x3008+0x40*X	TargetPhase High																TargetPhase[18:16]
0x300C+0x40*X	RampDownClk																RampDownClk
0x3010+0x40*X	Phase	Phase[15:0]															
0x3014+0x40*X	Phase Accumulator	PhaseAcc[15:0]															
0x3018+0x40*X	Target Phase	Target Phase[15:0]															
0x301C+0x40*X	Phase Control	PhaseTimeStep	Sign	PhaseOffset													

26.10. SPU Control Register

26.10.1. SPU Control Flag

CTRL_FLAG		0x50080E34							SPU Control Flag
Bit	15	14	13	12	11	10	9	8	
Function	SATURATIO N	-	LOOP_INITI ALSOS1	-	COMPRESS OR_EN	INTERPOLATION_EN[1:0]	-	-	
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
Function	VOLUME_SEL[2:0]			OVERLOADI NG	PHASEADDE R_INITIA	-			
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15]	SATURATION	R/W	This bit indicates whether the output signal is saturated, meaning the output signal will be clipped if it exceeds the maximum range.	Read 0: Saturate doesn't happen. Read 1: Saturate happens. Write 0: No operation. Write 1: Clear Saturate flag.
[14]	-	R	Reserved	-
[13]	LOOP_INITIALIZOS1	R/W	When this bit is 1, the default value of WDD0/WDD1 will be set as 0x8000 in ADPCM36 mode.	0: No initial 1: Initial Enabled
[12]	-	R	Reserved	-
[11]	COMPRESSOR_EN	R/W	When this bit or COMPRESSOR_EN is set as 1, the internal compressor will be activated and dynamically control the output volume.	0: Compressor Off 1: Compressor On
[10:9]	INTERPOLATION_EN[1:0]	R/W	The interpolation may improve the overall performance, and the high quality interpolation logic will be used to compensate the error caused by the phase-jitter effect.	00: High Quality Interpolation On 10: Interpolation On 01/11: Interpolation Off
[8]	-	R	Reserved	-
[7:5]	VOLUME_SEL[2:0]	R/W	When VolSel is set as 0, the volume of a single channel will be 1/64 of the max volume. When VolSel is given a non-zero value, the volume of a single channel becomes larger.	000: 1/64 001: 1/32 010: 1/16 011: 1/8 100: 1/4 101: 1/2 110: 1 111: 2

Bit	Function	Type	Description	Condition
[4]	OVERLOADING	R/W	This flag indicates the SPU is unable to handle such high sample rate and it will cause data lost.	Read 0: Overloading doesn't happen. Read 1: Overloading happens. Write 0: No operation. Write 1: Clear the flag.
[3]	PHASEADDER_INITIA	W	Initialization of Channel's Accumulator The initialization ensures the system is able to operate correctly.	0: No operation. 1: Initialize accumulator.
[2:0]	-	R	Reserved	-

26.10.2. Channel Enable Control

Before user enables a channel (ChEn[x] = 1), it is a **must** to initialize the attribute and phase RAM. Please refer to **Programming Note** section for initialization. Writing 1 to ChEn[x] will enable the channel while writing 0 will disable it. Programmer can know whether SPU is playing or not via checking the ChSts flag.

SPU_EN_CH0_15 **0x50080E00** **Channel[15:0] Enable Control**
SPU_EN_CH16_31 **0x50080E80** **Channel[31:16] Enable Control**

Bit	15	14	13	12	11	10	9	8
Function	CHx_EN[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	CHx_EN[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	CHx_EN[15:0]	R/W	Channel Enable Control and Status	0: Disable 1: Enable

STOP_STATUS_CH0_15 **0x50080E2C** **Channel[15:0] Stop Status**
STOP_STATUS_CH16_31 **0x50080EAC** **Channel[31:16] Stop Status**

Bit	15	14	13	12	11	10	9	8
Function	STOP_FLAG_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	STOP_FLAG_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	STOP_FLAG_CHx[15:0]	R/W	Channel Stop Status					Read 0: Channel is ready. Read 1: Channel is stopped. Write 0: No operation. Write 1: Clear Stop status.

CH_STATUS_CH0_15 **0x50080E3C** **Channel[15:0] Status**

CH_STATUS_CH16_31 **0x50080EBC** **Channel[31:16] Status**

Bit	15	14	13	12	11	10	9	8
Function	BUSY_FLAG_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	BUSY_FLAG_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	BUSY_FLAG_CHx[15:0]	R	Channel Status					Read 0: Channel is valid. Read 1: Channel is busy.

26.10.3. Main Volume Control

The MAIN_VOLUME controls the volume to the entire system. It is to assure that output does not overflow when all channels are turned on with max WDD, EDD, velocity and global volume.

MAIN_VOLUME **0x50080E04** **Main volume control**

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

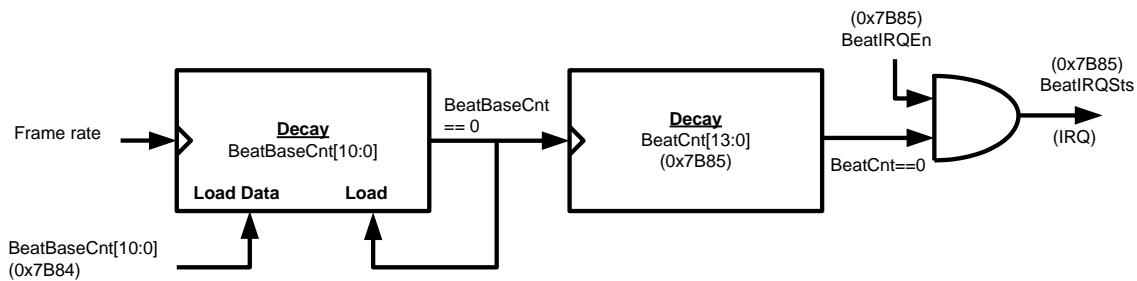
Bit	7	6	5	4	3	2	1	0	
Function	-	MAIN_VOLUME[6:0]							
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[15:7]	-	R	Reversed					-
[6:0]	MAIN_VOLUME[6:0]	R/W	Main Volume Control of The Entire System					0x00~0x7F

26.10.4. Beat Control

Beat trigger count will be subtracted by one every four frames, and beat base count will be loaded into beat trigger count when it reaches 0. Beat count will be subtracted by one every time when beat trigger count reaches 0.

$\text{BeatIRQ_Period} = (\text{BeatBaseCnt} * \text{BeatCnt}) * 4 * \text{Frame Rate}$ (Frame-Rate = 287.425kHz).



BEAT_BASE_COUNTER								0x50080E10	Beat Base Counter	
Bit	15	14	13	12	11	10	9	8		
Function	-								BeatBaseCnt[10:8]	
Reset Value	0	0	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
Function	BeatBaseCnt[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]	-	-	Reversed	-
[10:0]	BeatBaseCnt[10:0]	R/W	Beat base count will be subtracted by one every 4 frames.	0x000~0x7FF

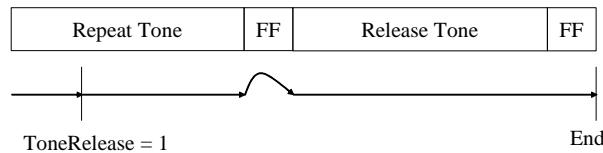
BEAT_COUNTER								0x50080E14	Beat Counter	
Bit	15	14	13	12	11	10	9	8		
Function	BEAT_COU NTER_INT_EN	BEAT_COUN TER_INT_FL	BEAT_COUNTER[13:8]							
Reset Value	0	0	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
Function	BEAT_COUNTER[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	BEAT_COUNTER_INT_EN	R/W	When Beat event (Beat decay = 0) is enabled, Beat IRQ will be activated. When Beat Counter IRQ is enabled and Beat decay = 0, Beat counter Interrupt will be triggered.	Read 0: BeatIRQ Disabled Read 1: BeatIRQ Enabled Write 0: Clear BeatIRQ. Write 1: No operation.
[14]	BEAT_COUNTER_INT_FLAG	R		0: BeatIRQ No Request 1: BeatIRQ Request
[13:0]	BEAT_COUNTER[13:0]	R/W	BeatIRQ_Period = BeatCnt * BeatBaseCnt * 4 * Frame	0x0000~0x3FFF

26.10.5. Tone Control

SPU output can be obtained from WaveOut and Waveln (mixed with SPU). For example, to achieve Echo effect, programmer should first acquire SPU's output from waveOut, and then mix Waveln with SPU after a certain period of prolongation. Release tone-color (timber) can be played via the ChToneRelease function. Channel will complete the current tone-color play and begin to play the completed release tone.



WaveOut 0x50080E48 Output Data of SPU							
Bit	31	30	29	28	27	26	25
Function	WAVE_OUT_R[15:8]						
Reset Value	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	WAVE_OUT_R[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	WAVE_OUT_L[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	WAVE_OUT_L[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:16]	WAVE_OUT_R[15:0]	R	Right Channel Data of SPU					-
[15:0]	WAVE_OUT_L[15:0]	R	Left Channel Data of SPU					-

TONE_RELEASE_CH0_15 **0x50080E58** **Channel[15:0] | Tone Release Register**

TONE_RELEASE_CH16_31 **0x50080ED8** **Channel[31:16] | Tone Release Register**

Bit	15	14	13	12	11	10	9	8
Function	TONE_RELEASE_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	TONE_RELEASE_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	TONE_RELEASE_CHx [15:0]	R/W	This register controls whether to perform tone release of each channel. This bit is cleared automatically after the channel is stopped					0: No operation. 1: Tone Release

26.10.6. Channel IRQ Control

Channel IRQ Control is another route to play acoustic by using software approach. When an end code of tone-color is reached, an IRQ interrupt will be established. For example, if user plays SACM library with A/B buffer, wave address is set in A and loop address in B at initial state before playing wave address. When A buffer is playing at the end code, it plays loop address (B buffer) and establishes an interrupt. Programmer, at this moment, should change the loop address to A buffer. When B buffer plays at the end code, it auto plays A buffer again and issues an interrupt. Programmer then changes the loop address to B buffer until the end of the acoustic.

INT_EN_CH0_15 **0x50080E08** **Channel[15:0] IRQ Enable**

INT_EN_CH16_31 **0x50080E88** **Channel[31:16] IRQ Enable**

Bit	15	14	13	12	11	10	9	8
Function	INT_CHx_EN[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	INT_CHx_EN[7:0]							
Reset Value	0	0	0	0	0	0	0	0



GPCM3 SERIES PROGRAMMING GUIDE

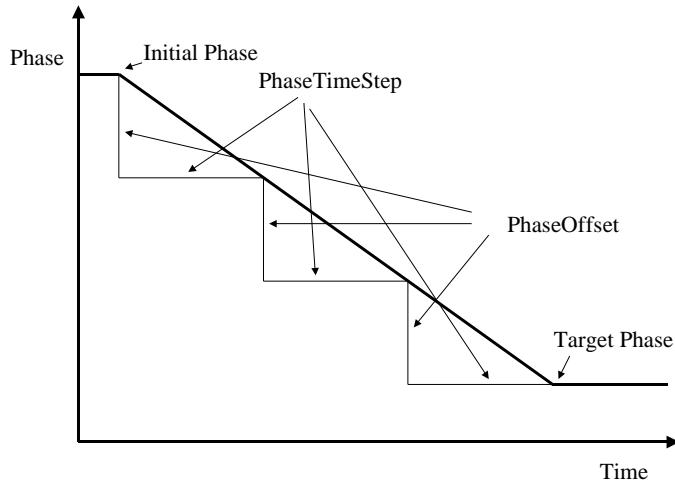
Bit	Function	Type	Description	Condition
[15:0]	INT_CHx_EN[15:0]	R/W	Control IRQ Enable Please refer to register INT_STATUS_CHx.	0: Disable 1: Enable

INT_STATUS_CH0_15	0x50080E0C	Channel[15:0] IRQ Status						
INT_STATUS_CH16_31	0x50080E8C	Channel[31:16] IRQ Status						
Bit	15	14	13	12	11	10	9	8
Function	INT_FLAG_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	INT_FLAG_CHx[15:0]	R/W	INT_STATUS_CHx occurs when END CODE is reached (0xFFFF for 16-bit mode and 0xFF for 8-bit mode). IRQ occurs when the corresponding INT_CHx_EN and INT_FLAG_CHx are activated.	Read 0: Channel IRQ is inactive. Read 1: Channel IRQ is active. Write 0: No operation. Write 1: Clear IRQ.

26.10.7. PitchBend Enable Control

The primary function of this register is to control the pitch bend of each channel. The TargetPhase[18:0], PhaseOffset[11:0], PhaseTimeStep[2:0], and PhaseSign will be used to increase/decrease phase. Please refer to **Internal SRAM** for more information.



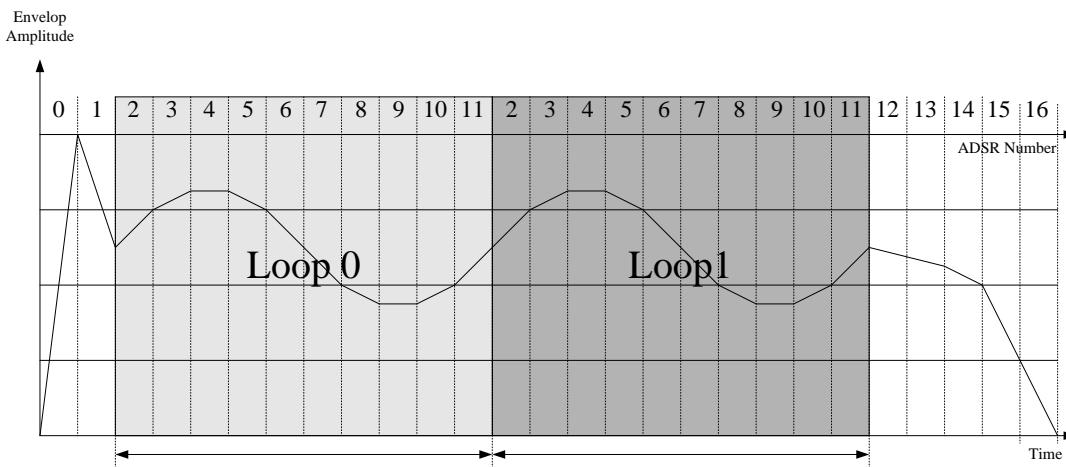
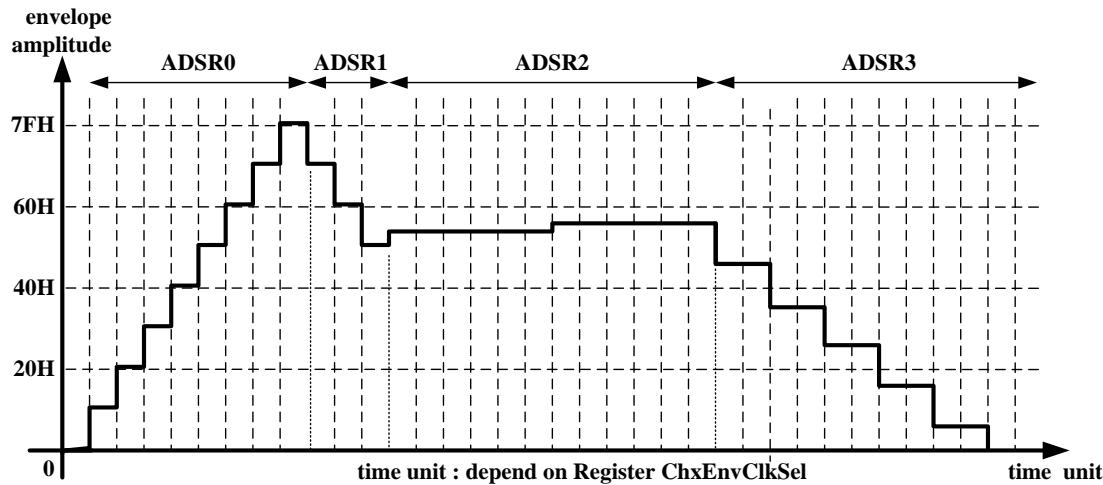
PITCHBEND_EN_CH0_15	0x50080E60	Channel[15:0] Pitch Bend Enable
PITCHBEND_EN_CH16_31	0x50080EE0	Channel[31:16] Pitch Bend Enable
Bit	15	14
Function	PITCHBEND_EN_CHx[15:8]	
Reset Value	0	0

Bit	7	6	5	4	3	2	1	0
Function	PITCHBEND_EN_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	PITCHBEND_EN_CHx[15:0]	R/W	This register controls the pitch bend feature of each channel.	0 : Disabled 1 : Enable

26.10.8. Envelope Control

Two envelop modes are supported, auto mode and manual mode. When manual mode is selected, programmer must update the envelop value in the envelop IRQ. When auto mode is selected, programmer should configure every channel's envelop clock to determine the velocity of updating envelop. After that, SPU will automatically update envelops; please refer to the diagram below. In addition, it has a repeat function to synthesize the Low Frequency OSC (LFO) envelop easily.



The envelop clock speed is shown as follows:

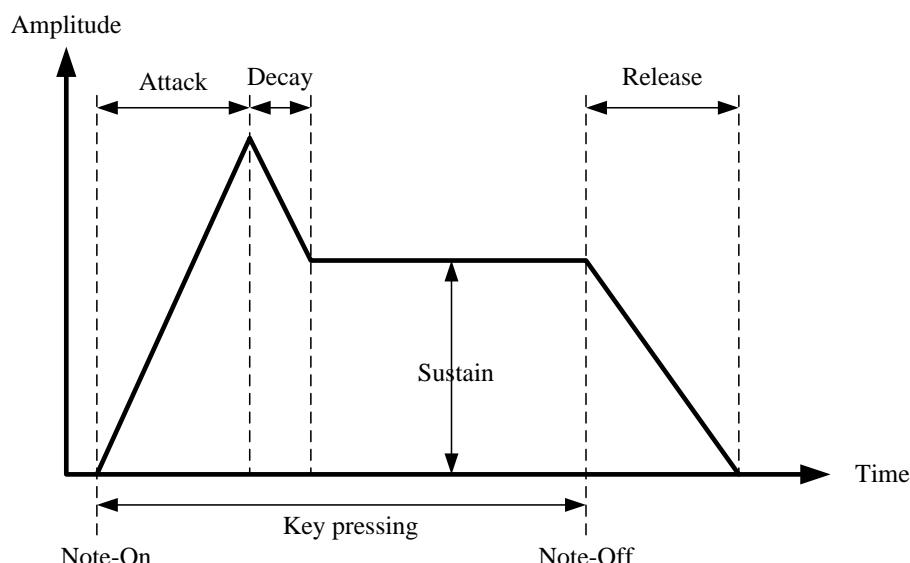
ChxEnvClk where, x is channel 0~31, Frame-Rate = 287.425KHz

Note: Presently, sunmidar2 supports the envelope clock of 2.25KHz.

Setting	Description	Unit
0000	EnvClk Count once = Frame-Rate / 4 * 4	17.96 KHz
0001	EnvClk Count once = Frame-Rate / 8 * 4	8.98 KHz
0010	EnvClk Count once = Frame-Rate / 16 * 4	4.49 KHz
0011	EnvClk Count once = Frame-Rate / 32 * 4	2.25 KHz
0100	EnvClk Count once = Frame-Rate / 64 * 4	1.12 KHz
0101	EnvClk Count once = Frame-Rate / 128 * 4	561 Hz
0110	EnvClk Count once = Frame-Rate / 256 * 4	280 Hz
0111	EnvClk Count once = Frame-Rate / 512 * 4	140 Hz
1000	EnvClk Count once = Frame-Rate / 1024 * 4	70 Hz
1001	EnvClk Count once = Frame-Rate / 2048 * 4	35 Hz
1010	EnvClk Count once = Frame-Rate / 4096 * 4	17.5 Hz

Setting	Description	Unit
1011	EnvClk Count once = Frame-Rate / 8192 * 4	8.8 Hz
1100	EnvClk Count once = Frame-Rate / 8192 * 4	8.8 Hz
1101	EnvClk Count once = Frame-Rate / 8192 * 4	8.8 Hz
1110	EnvClk Count once = Frame-Rate / 8192 * 4	8.8 Hz
1111	EnvClk Count once = Frame-Rate / 8192 * 4	8.8 Hz

As the diagram shown below, when Note-off is received, envelope will enter release stage (ramp down).



ENV_MODE_CH0_15 **0x50080E54** **Channel[15:0] Envelope Mode**

ENV_MODE_CH16_31 **0x50080ED4** **Channel[31:16] Envelope Mode**

Bit	15	14	13	12	11	10	9	8
Function	ENV_MODE_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENV_MODE_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ENV_MODE_CHx[15:0]	R/W	This register determines the envelope mode for each channel. Programmer can switch a channel from auto to manual mode anytime.	0: Auto Mode 1: Manual Mode

ENV_REPEAT_CH0_15	0x50080E50	Channel[15:0] Envelope Repeat Enable						
ENV_REPEAT_CH16_31	0x50080ED0	Channel[31:16] Envelope Repeat Enable						
Bit	15	14	13	12	11	10	9	8
Function	ENV_REPEAT_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENV_REPEAT_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ENV_REPEAT_CHx[15:0]	R/W	This register is used to control the repeat feature of the envelope data.					0: Disabled 1: Enabled

ENV_INT_STATUS_CH0_15	0x50080E5C	Channel[15:0] Envelope IRQ Status						
ENV_INT_STATUS_CH16_31	0x50080EDC	Channel[31:16] Envelope IRQ Status						
Bit	15	14	13	12	11	10	9	8
Function	ENV_INT_FLAG_CHx[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENV_INT_FLAG_CHx[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	ENV_INT_FLAG_CHx[15:0]	R/W	When ENV_INT_EN is 1 and the value of ENV_ADDR matches ENV_INT_ADDR, Envelope IRQ will be triggered, and at this time, Envelope INT Flag = 1.					0: No Envelope IRQ. 1: Envelope IRQ is triggered. Write 1 to clear this bit.

ENV_CLK_CH0_3	0x50080E18	Channel[3:0] Envelope Interval Selection						
ENV_CLK_CH4_7	0x50080E1C	Channel[7:4] Envelope Interval Selection						
ENV_CLK_CH8_11	0x50080E20	Channel[11:8] Envelope Interval Selection						
ENV_CLK_CH12_15	0x50080E24	Channel[15:12] Envelope Interval Selection						
Bit	15	14	13	12	11	10	9	8
Function	ENV_CLK_CHx[3:0]				ENV_CLK_CHx[3:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENV_CLK_CHx[3:0]				ENV_CLK_CHx[3:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	ENV_CLK_CHx [3:0]	R/W	CH3 Envelope Interval Selection (ENV_CLK_CH0_3) CH8 Envelope Interval Selection (ENV_CLK_CH4_7) CH11 Envelope Interval Selection (ENV_CLK_CH8_11) CH15 Envelope Interval Selection (ENV_CLK_CH12_15)				Please refer to Envelope clock table.
[11:8]	ENV_CLK_CHx [3:0]	R/W	CH2 Envelope Interval Selection (ENV_CLK_CH0_3) CH7 Envelope Interval Selection (ENV_CLK_CH4_7) CH10 Envelope Interval Selection (ENV_CLK_CH8_11) CH14 Envelope Interval Selection (ENV_CLK_CH12_15)				Please refer to Envelope clock table.
[7:4]	ENV_CLK_CHx [3:0]	R/W	CH1 Envelope Interval Selection (ENV_CLK_CH0_3) CH6 Envelope Interval Selection (ENV_CLK_CH4_7) CH9 Envelope Interval Selection (ENV_CLK_CH8_11) CH13 Envelope Interval Selection (ENV_CLK_CH12_15)				Please refer to Envelope clock table.
[3:0]	ENV_CLK_CHx [3:0]	R/W	CH0 Envelope Interval Selection (ENV_CLK_CH0_3) CH4 Envelope Interval Selection (ENV_CLK_CH4_7) CH8 Envelope Interval Selection (ENV_CLK_CH8_11) CH12 Envelope Interval Selection (ENV_CLK_CH12_15)				Please refer to Envelope clock table.

ENV_CLK_CH16_19	0x50080E98	Channel[19:16] Envelope Interval Selection
ENV_CLK_CH20_23	0x50080E9C	Channel[23:20] Envelope Interval Selection
ENV_CLK_CH24_27	0x50080EA0	Channel[27:24] Envelope Interval Selection
ENV_CLK_CH28_31	0x50080EA4	Channel[31:28] Envelope Interval Selection

Bit	15	14	13	12	11	10	9	8
Function	ENV_CLK_CHx[3:0]				ENV_CLK_CHx[3:0]			
Reset Value	0	0	0	0	0	0	0	0

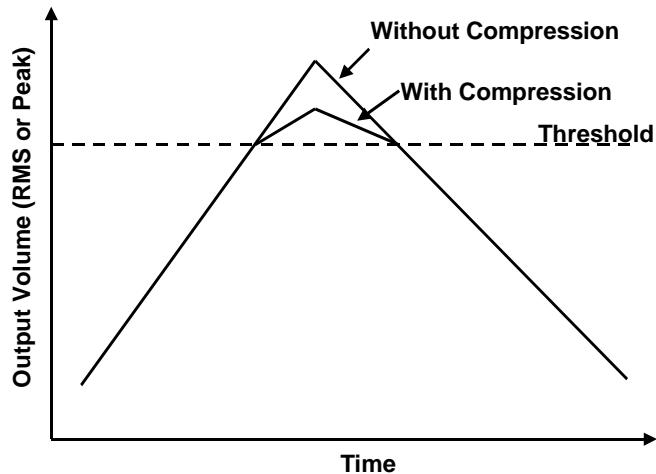
Bit	7	6	5	4	3	2	1	0
Function	ENV_CLK_CHx[3:0]				ENV_CLK_CHx[3:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:12]	ENV_CLK_CHx [3:0]	R/W	CH19 Envelope Interval Selection (ENV_CLK_CH16_19) CH23 Envelope Interval Selection (ENV_CLK_CH20_23) CH27 Envelope Interval Selection (ENV_CLK_CH24_27) CH31 Envelope Interval Selection (ENV_CLK_CH28_31)				Please refer to Envelope clock table.

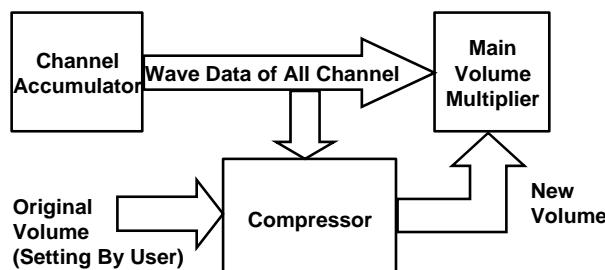
Bit	Function	Type	Description	Condition
[11:8]	ENV_CLK_CHx [3:0]	R/W	CH18 Envelope Interval Selection (ENV_CLK_CH16_19) CH22 Envelope Interval Selection (ENV_CLK_CH20_23) CH26 Envelope Interval Selection (ENV_CLK_CH24_27) CH30 Envelope Interval Selection (ENV_CLK_CH28_31)	Please refer to Envelope clock table.
[7:4]	ENV_CLK_CHx [3:0]	R/W	CH17 Envelope Interval Selection (ENV_CLK_CH16_19) CH21 Envelope Interval Selection (ENV_CLK_CH20_23) CH25 Envelope Interval Selection (ENV_CLK_CH24_27) CH29 Envelope Interval Selection (ENV_CLK_CH28_31)	Please refer to Envelope clock table.
[3:0]	ENV_CLK_CHx [3:0]	R/W	CH16 Envelope Interval Selection (ENV_CLK_CH16_19) CH20 Envelope Interval Selection (ENV_CLK_CH20_23) CH24 Envelope Interval Selection (ENV_CLK_CH24_27) CH38 Envelope Interval Selection (ENV_CLK_CH28_31)	Please refer to Envelope clock table.

26.10.9. Compressor Control

When CompEn is 1, the internal compressor will be activated and the output volume parameter of dynamically control will be stored in 0x0E38 and 0x0E64. When the VolSel is set as a non-zero value, it's possible that the output wave will saturate, and the compressor can raise the overall output volume without saturation.



The compressor detects the output level **before** the main volume multiplier, and it will control the main volume dynamically. Please refer to the following diagram for more details.

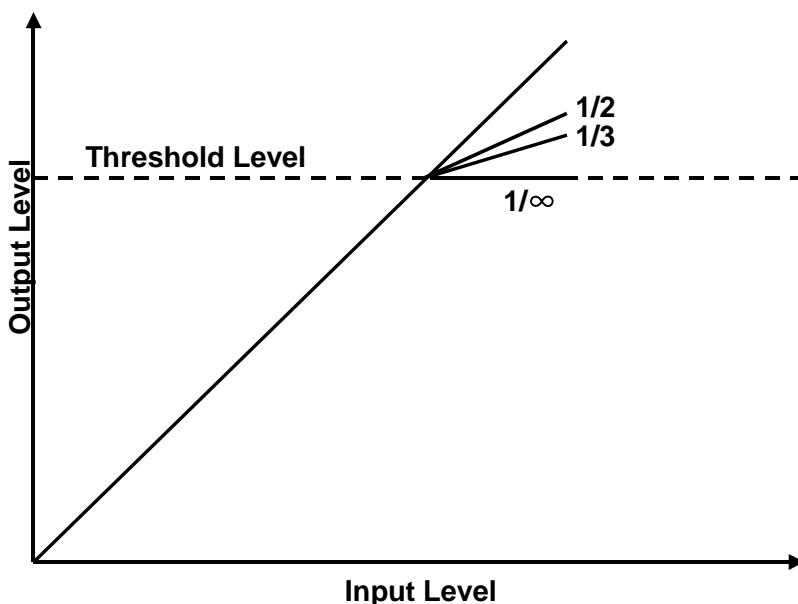


Suppose the maximum value of wave data at all channels is 1. When peak or RMS of wave data at all channels is larger than the threshold value, 0x80, the compressor will be activated to low the volume. This can prevent the case of saturation.

The definition of attack time is how fast the compressor responses to the situation of the wave data peak or RMS exceeding the threshold. The fast attack time will result in fast response to the wave data, meaning the compress will start in a short time. This is mainly to prevent saturation but the transient part in the wave may be lost. So, fast attack time is good for melody but not for drum. The slow attack time will result in slow response to the wave data, meaning the compress will start after a while. This is to preserve the transient part in the wave but it's not suitable for avoiding saturation. User can try both settings and find the best combinations for the application. Real attack time of "1" means the attack time is approx. 1.5ms. Real attack time of "100" means the attack time is approx. 150 ms..., etc.

The definition of release time is how fast the compressor responses to the situation of the wave data peak or RMS being lower than the threshold. Release time which is too fast will result in breathing effect situation but release time which is too slow will lower the overall volume. User can try both settings and find the best one for the application. Real release time of "1" means the release time is approx. 1.5 ms. Real release time of "100" means the release time is approx. 150 ms..., etc.

Compress Ratio Setting determines how much it compresses when output-wave level is higher than the threshold level.



Since the compress ratio highly depends on the setting of VolSel, the following table can be used for suitable ratio selection.

Recommended Threshold Level and Ratio Combination in Various VolSel Setting

VolSel	Minimum Channel Number to Cause Saturation			Effect	Suggested Threshold Level	Suggested Ratio Setting
	32-ch	16-ch	8ch			
00 (x1)	Not Possible			-	Turn-off Compressor	Turn-off Compressor
01 (x2)	16	8	4	Compress	0x20 (1/4)	0 (1/2)
				Limit	0x40 (1/2)	7 (1/ ∞)
10 (x4)	8	4	2	Compress	0x10 (1/8)	0 (1/2)
				Limit	0x20 (1/4)	7 (1/ ∞)
11 (x16)	2	1	0.5	Compress	0x04 (1/32)	0 (1/2)
				Limit	0x08 (1/16)	7 (1/ ∞)

To use compress ratio other than 1/2 and 1/ ∞ , user can use the threshold level between these two settings in order to get the best outcome. Larger compress ratio will lower the overall volume but also lower the possibility of saturation.

COMPRESSOR									0x50080E38									Compressor Control								
Bit	15	14	13	12	11	10	9	8	Bit	15	14	13	12	11	10	9	8	Bit	15	14	13	12	11	10	9	8
Function	COMPRESS OR_MODE								Function	COMPRESSOR_ATTACK[1:0]			COMPRESSOR_RELEASE[1:0]	COMPRESS OR_ZC_EN				Function	COMPRESSOR_THRESHOLD[6:0]							
Reset Value	0	0	0	0	0	0	0	0	Reset Value	0	0	0	0	0	0	0	0	Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	COMPRESSOR_ATTACK[1:0]		COMPRESSOR_RELEASE[1:0]	COMPRESS OR_ZC_EN				
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	COMPRESSOR_MODE	R/W	RMS or Peak Mode Selection	0: RMS Mode 1: Peak Mode
[14:8]	COMPRESSOR_THRESH OLD[6:0]	R/W	The threshold's maximum value is 0x7F, and the minimum is 0x01. 0x00 is not allowed.	0x01~0x7F
[7:6]	COMPRESSOR_ATTACK [1:0]	R/W	Attack Time Scale Control Register	00: AttackTime[7:0] * 1 01: AttackTime[7:0] * 4 10: AttackTime[7:0] * 16 11: AttackTime[7:0] * 64
[5:4]	COMPRESSOR_RELEASE E[1:0]	R/W	Release Time Scale Control Register	00: ReleaseTime [7:0] * 1 01: ReleaseTime [7:0] * 4 10: ReleaseTime [7:0] * 16 11: ReleaseTime [7:0] * 64

Bit	Function	Type	Description	Condition
[3]	COMPRESSOR_ZC_EN	R/W	Enable Zero Cross Function of Compressor. The compressor will adjust the main volume automatically, and this bit is used to determine whether the volume changes after the data crosses zero.	0: Enable zero cross. 1: Disable zero cross.
[2:0]	COMPRESSOR_RATIO [2:0]	R/W	The parameter determines the compression ratio of the volume when the output wave level is higher than the threshold level.	0: 1/2 1: 1/3 2: 1/4 3: 1/5 4: 1/6 5: 1/7 6: 1/8 7: 1/ ∞ (Limiter)

ATTACK_RELEASE_TIME									0x50080E68									Attack/Release Time Control								
Bit	15	14	13	12	11	10	9	8																		
Function	ATTACK_TIME[7:0]																									
Reset Value	0	0	0	0	0	0	0	0																		

Bit	7	6	5	4	3	2	1	0
Function	RELEASE_TIME[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	ATTACK_TIME[7:0]	R/W	The definition of attack time is how fast the compressor responses to the wave data peak or RMS over the threshold.	0x00~0xFF
[7:0]	RELEASE_TIME[7:0]	R/W	The definition of release time is how fast the compressor responses to the wave data peak or RMS lower than the threshold.	0x00~0xFF

26.10.10. Post Process Control

POSTWAVE_CTRL		0x50080E94									PostWave(PW) Processing Control									
Bit	15	14	13	12	11	10	9	8												
Function	DMA_EN	DIRECTOUT_EN	-							DMA_DOWNSAMPLE[1:0]										
Reset Value	0	0	0	0	0	0	0	0												

Bit	7	6	5	4	3	2	1	0
Function	FMT	SILENCE_EN			-			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	DMA_EN	R/W	Post Wave Output to DMA Enable	0: Disable 1: Enable
[14]	DIRECTOUT_EN	R/W	Direct Output Enable for Post Wave	0: Disable 1: Enable
[13:10]	-	R	Reserved	-
[9:8]	DMA_DOWNSAMPLE[1:0]	R/W	DMA down-sample for SPU waveout or SPU+SOFCH waveout.	11: No down sample. (DMA sample with 287.425KHz) 01: Down-sample by 6. (DMA sample with 47.9KHz) 10: Down-sample by 12. (DMA sample with 23.95KHz) 11: Down-sample by 31. (DMA sample with 9.27KHz)
[7]	FMT	R/W	Post Wave Output in signed or unsigned format.	0: Unsigned 1: Signed
[6]	SILENCE_EN	R/W	Post Wave Not Output to DAC	0: Not output to DAC. 1: Output to DAC.
[5:0]	-	R	Reserved	-

POSTWAVE_OUT									0x50080E94	Post Wave Output Data		
Bit	31	30	29	28	27	26	25	24				
Function	POST_WAVE_OUT_R[15:8]											
Reset Value	0	0	0	0	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Function	POST_WAVE_OUT_R[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	POST_WAVE_OUT_L[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	POST_WAVE_OUT_L[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:16]	POST_WAVE_OUT_R[15:0]	R	Right Channel Data of Post Wave	-
[15:0]	POST_WAVE_OUT_L[15:0]	R	Left Channel Data of Post Wave	-

26.11. Internal SRAM

Internal Attribute SRAM Format, Channel 0 ~ Channel 32

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0																		
0x1000+0x40*X	WAVE_ADDR	WAVE_ADDR[15:0]																																	
0x1004+0x40*X	WAVE_MODE	AVE_ALGORIT HM	WAVE_P CM	WAVE_PLAYMODE	WAVE_LOOP_ADDR[21:16]										WAVE_ADDR [21:16]																				
0x1008+0x40*X	WAVE_LOOPADDR	WAVE_LOOPADDR [15:0]																																	
0x100C+0x40*X	PAN_VELOCITY		VOLUME_PAN[6:0]							VELOCITY[6:0]																									
0x1010+0x40*X	ENV_CTRL0		ENV_TARGET[6:0]						ENV_SIG N	ENV_STEP[6:0]																									
0x1014+0x40*X	ENV_DATA	ENV_COUNTER[7:0]							ENV_DATA[6:0]																										
0x1018+0x40*X	ENV_CTRL1	ENV_REPEAT_CNT						ENV_RE PEAT_EN	ENV_LOAD_CNT[7:0]																										
0x101C+0x40*X	ENV_INT_CTRL	ENV_INT_ADDR[8:0]						ENV_INT _EN	ENV_ADDR[21:16]																										
0x1020+0x40*X	ENV_ADDR	ENV_ADDR[15:0]																																	
0x1024+0x40*X	WAVE_DATA0	WAVE_DATA0[15:0]																																	
0x1028+0x40*X	ENV_RAMPDOWN_STEP	ENV_RAMPDOWN_STEP[6:0]						ENV_REPEAT_OFFSET[8:0]																											
0x102C+0x40*X	WAVE_DATA	WAVE_DATA[15:0]																																	
0x1030+0x40*X	-																																		



GPCM3 SERIES PROGRAMMING GUIDE

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0		
0x1034+0x40*X	WAVE_ADPCM_CTRL	WAVE_ADPCM _SEL	WAVE_A DPCM_F INALFRA ME	WAVE_ADPCM_FRAMEPOINT															
0x1038+0x40*X	WAVE_ADDRH		WAVE_ADDR[30:28]		WAVE_LOOP_ADDR[27:22]						WAVE_ADDR [27:22]								
0x103C+0x40*X	ENV_ADDRH													ENV_ADDR[30:22]					

Note: 0x40*X, X = chnum. For example, the address of Mode ch0 = 0x1004/0x1008/0x100C/..., so the address should be SRAMBASE (= 0x50080000) + Attrib (0x103C) + ChannelNum << 6 (0x40, every channel takes 64 address).



GPCM3 SERIES PROGRAMMING GUIDE

Internal Phase SRAM Format, Channel 0 ~ Channel 32

Address	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0x3000+0x40*X	PHASE_VALUEH																PHASE_VALUE [18:16]
0x3004+0x40*X	PHASE_ACCH																PHASE_ACC[18:16]
0x3008+0x40*X	PITCHBEND_TARGETH																PITCHBEND_TARGET [18:16]
0x300C+0x40*X	ENV_RAMPDOWN_CLK																ENV_RAMPDOWN_CLK[2:0]
0x3010+0x40*X	PHASE_VALUE	PHASE_VALUE[15:0]															
0x3014+0x40*X	PHASE_ACC	PHASE_ACC[15:0]															
0x3018+0x40*X	PITCHBEND_TARGET	PITCHBEND_TARGET[15:0]															
0x301C+0x40*X	PITCHBEND_CTRL	PITCHBEND_CLK[15:13]		PITCHBE ND_SIGN	PITCHBEND_STEP[11:0]												

26.11.1. Tone-Color Register

Basically, there are 4 modes of wave data: 8-bitPCM, 16-bit PCM, ADPCM and ADPCM36. Only Auto-End and Auto-Repeat mode are available in ADPCM mode. When the end code (0xFFFF) is reached, this bit is cleared as 0 automatically, i.e. back to PCM mode. It means only the wave data in the first region will be treated as ADPCM data, and others are assumed as PCM data.

Four Ways to Play Tone-Color

00: Software Mode (S/W): Tone-Color S/W Mode.

In S/W mode, the tone color value is the value in WDD ($0x1024 + 0x40*X$), which is written by CPU.

01: Hardware (H/W) Auto-eEnd Mode: Tone-Color H/W output with auto end.

SPU auto loads tone-color and stops playing when tone-color hits the end-code ("0xFF" for 8-bit tone color data mode, "0xFFFF" for 16-bit tone color data mode, and "0x7FFF" for signed 16-bit tone color data mode).

10: Hardware (H/W) Auto-Repeat Mode: Tone-Color H/W output with auto repeat.

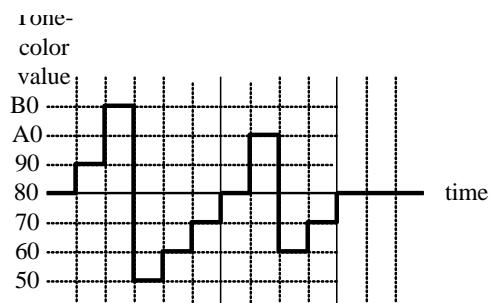
SPU reloads tone-color automatically when end-code is reached. The data between loop address and end-code is repeated till the channel is off. If ADPCM36 Mode is selected, it means the data in loop area is ADPCM36 format.

11: Hardware (H/W) Auto-Repeat Mode 1: Tone-Color H/W output with auto repeat.

SPU reloads tone-color automatically when end-code is reached. The data between loop address and end-code is repeated till the channel is off. When ADPCM36 Mode is used, it means the data in loop area is PCM format. The mode of PCM (8 or 16) is configured in WAVE_MODE Bit[14].

Example: H/W Auto-End Mode

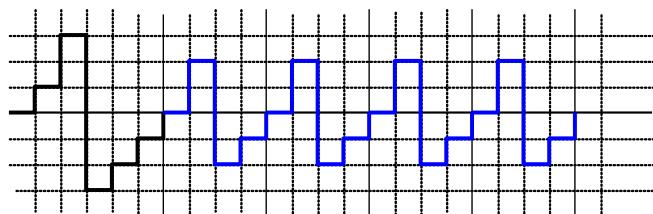
Address	Data
WAddr + 0	0x9080
WAddr + 1	0x50B0
WAddr + 2	0x7060
WAddr + 3	0xA080
WAddr + 4	0x7060
WAddr + 5	0xFFFF



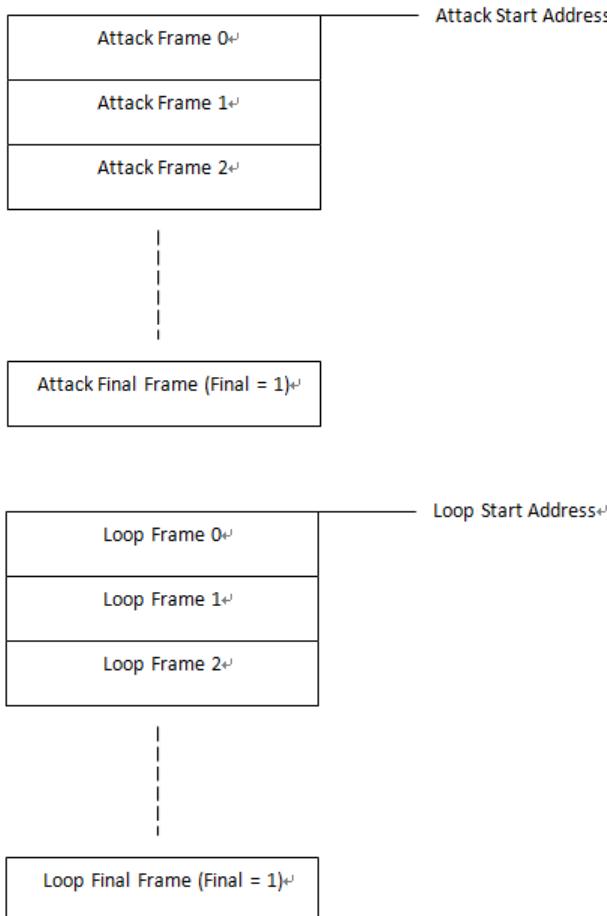
Example: H/W Auto-Repeat Mode

LoopAddr = WAddr + 3

Address	Data
WAddr + 0	0x9080
WAddr + 1	0x50B0
WAddr + 2	0x7060
WAddr + 3	0xA080
WAddr + 4	0x7060
WAddr + 5	0xFFFF

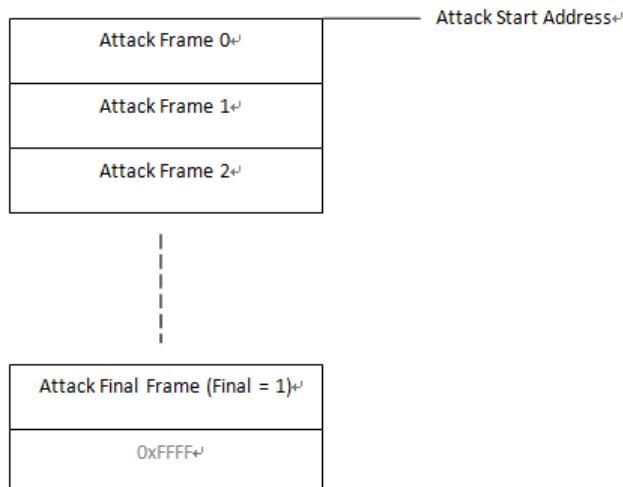


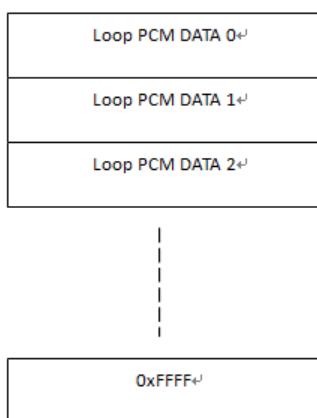
In ADPCM36 mode, when the ToneMode is set as 10b, the data format is shown in the following diagram:



ToneMode = 10b, ADPCM36 => ADPCM36 Mode

In ADPCM36 mode, an end code (0xFFFF) must be appended at the end of the attack data when the ToneMode is set as 11b. The data format is shown in the following diagram:





ToneMode = 11b, ADPCM36 => PCM Mode

WAVE_ADDR								0x50081000	Wave Data Start Address		
Bit	15	14	13	12	11	10	9	8	Function	WAVE_ADDR[15:8]	
Reset Value	0	0	0	0	0	0	0	0			

Bit	7	6	5	4	3	2	1	0	Function	WAVE_ADDR[7:0]	
Reset Value	0	0	0	0	0	0	0	0			

Bit	Function	Type	Description						Condition
[15:0]	WAVE_ADDR[15:0]	R/W	Wave data start address (WAVE_ADDR[30:0]) is composed of WAVE_MODE, WAVE_ADDR, and WAVE_ADDRH.						-

WAVE_MODE								0x50081004	Wave Mode Setting				
Bit	15	14	13	12	11	10	9	8	Function	WAVE_ALG ORITHM	WAVE_PCM	WAVE_PLAYMODE[1:0]	WAVE_LOOP_ADDR[21:18]
Reset Value	0	0	0	0	0	0	0	0					

Bit	7	6	5	4	3	2	1	0	Function	WAVE_LOOP_ADDR[17:16]	WAVE_ADDR[21:16]
Reset Value	0	0	0	0	0	0	0	0			

Bit	Function	Type	Description						Condition
[15]	WAVE_ALGORITHM	R/W	ADPCM Mode On/Off						0: PCM Data 1: ADPCM Data
[14]	WAVE_PCM	R/W	Tone Color Data Resolution Control In 8-bit tone color data mode, wave data is accessed byte by byte. In 16-bit tone color data mode, data is accessed by word.						0: 8-bit Data 1: 16-bit Data

Bit	Function	Type	Description	Condition
[13:12]	WAVE_PLAYMODE[1:0]	R/W	Tone-Color Mode Control 00: S/W Mode 01: H/W Auto-End Mode 10:H/W Auto-Repeat Mode 11: H/W Auto-Repeat Mode	
[11:6]	WAVE_LOOP_ADDR[21:16]	R/W	Wave Data Loop Address (Loop Addr[30:0])	-
[5:0]	WAVE_ADDR[21:16]	R/W	Wave Data Start Address (Wave Addr[30:0])	-

WAVE_LOOPADDR **0x50081008** **Wave Data Loop Address**

Bit	15	14	13	12	11	10	9	8
Function	WAVE_LOOPADDR[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	WAVE_LOOPADDR[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15:0]	WAVE_LOOPADDR[15:0]	R/W	Wave data loop address (WAVE_LOOPADDR[30:0]) is composed of WAVE_MODE, WAVE_LOOPADDR, and WAVE_LOOPADDRH. This address is used for the tone-color automatic repeat mode.						-

WAVE_ADDRH **0x50081038** **Wave/Loop Address**

Bit	15	14	13	12	11	10	9	8	
Function	-	WAVE_ADDR[30:28]/ WAVE_LOOPADDR[30:28]				WAVE_LOOPADDR[27:24]			
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Function	WAVE_LOOPADDR[23:22]				WAVE_ADDR[27:22]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15]	-	R	Reserved						-
[14:12]	WAVE_ADDR[30:28]/ WAVE_LOOPADDR[30:28]	R/W	Waddr and LoopAddr share the same control register for address[30:28].						-
[11:6]	WAVE_LOOPADDR [27:22]	R/W	Wave Data Loop Address (WAVE_LOOPADDR[30:0])						-
[5:0]	WAVE_ADDR[27:22]	R/W	Wave (Tone Color) Data Start Address (WAVE_ADDR[30:0])						-

WAVE_DATA0
0x50081024
Previous Wave Data

Bit	15	14	13	12	11	10	9	8
Function	WAVE_DATA0[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	WAVE_DATA0[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WAVE_DATA0[15:0]	R/W	User must write 0x8000 to this register in ADPCM36 mode. In other modes, this register can be ignored.					-

WAVE_DATA
0x5008102C
Wave Data

Bit	15	14	13	12	11	10	9	8
Function	WAVE_DATA[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	WAVE_DATA[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:0]	WAVE_DATA[15:0]	R/W	In tone color automatic mode (0x1004+0x40*X), WDD is obtained automatically by the hardware. In software mode, WDD has to be written by CPU. In 8-bit tone color mode (0x1004+0x40*X), the wave data should be located in WDD[15:8]. In 16-bit tone color mode, the tone color data should be located in WDD[15:0].					-

WAVE_ADPCM_CTRL
0x50081034
ADPCM Mode Selection

Bit	15	14	13	12	11	10	9	8
Function	ADPCM_SEL	ADPCM_FIN ALFRAME	ADPCM_FRAMEPOINT[4:0]					-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15]	ADPCM_SEL	R/W	When WAVE_MOVE Bit[15] = 1 (0x1004+0x40*X), user can set the system as ADPCM or ADPCM36 mode through this bit after ADPCM mode is activated.					0: ADPCM 1: ADPCM36

Bit	Function	Type	Description	Condition
[14]	ADPCM_FINALFRAME	R/W	This bit determines whether to update the frame data or not. Programmer should set this bit as 0.	0: Update 1: Not update
[13:9]	ADPCM_FRAMEPOINT[4:0]	R/W	Point Number in ADPCM36 Mode User must fill the correct value in this field before playing ADPCM36 tone colors. This field must be filled with 0 when the tone color is not in ADPCM36 mode.	0: 1 point 1: 2 points 2: 3 points 3: 4 points 31: 32 points
[8:0]	-	R	Reserved	-

26.11.2. Volume Register

When Pan < 64

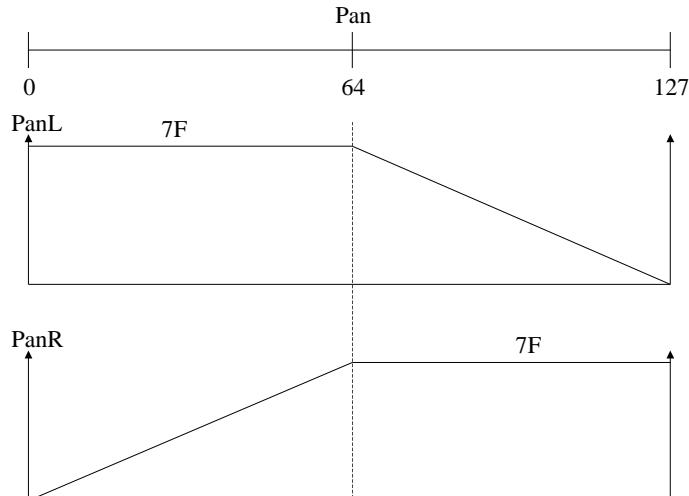
$$\text{PanL} = 0x7F * \text{ChVolume}$$

$$\text{PanR} = \text{Pan} * 2 * \text{ChVolume}$$

When Pan >= 64

$$\text{PanL} = (127-\text{Pan}) * 2 * \text{ChVolume}$$

$$\text{PanR} = 0x7F * \text{ChVolume}$$



PAN_VELOCITY

0x5008100C

Volume Setting for Right and Left Channel

Bit	15	14	13	12	11	10	9	8
Function								
Reset Value	0	0	0	0	0	0	0	0

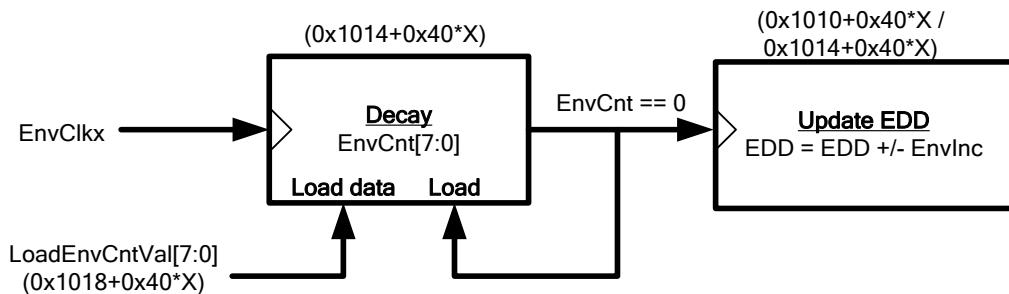
Bit	7	6	5	4	3	2	1	0
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description		Condition
[15]	-	R	Reserved		-
[14:8]	VOLUME_PAN[6:0]	R/W	Ratio Control for Right and Left Channel		00~7F
[7]	-	R	Reserved		-
[6:0]	VELOCITY[6:0]	R/W	Velocity Control for Right and Left Channel		00~7F

26.11.3. Envelope Register

0x1010+0x40*X, 0x1014+0x40*X and 0x1018+0x40*X are the three registers that control the envelope. In Envelope automatic mode (0x1004+0x40*X), these registers are automatically reloaded or updated by the hardware from memory. In Envelope manual mode, CPU has to update the envelope data (EDD in 0x1014+0x40*X). User can control the envelope in automatic mode by Envelope0 (0x1010+0x40*X) and Envelope1 (0x1018+0x40*X), which are stored in memory.

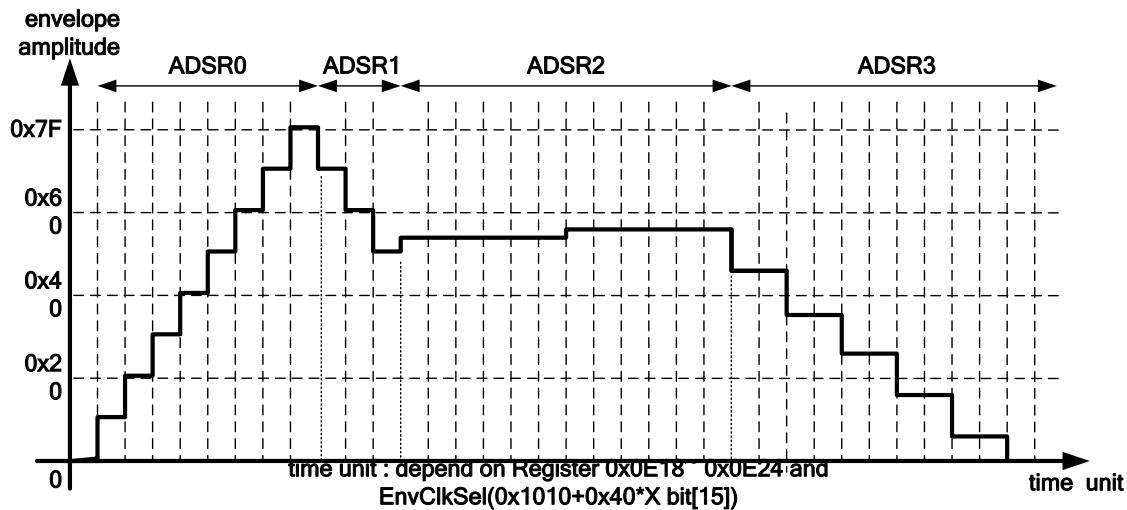
At the beginning, sound processor auto-loads EnvLoad[7:0] to internal register, **EnvCnt [7:0]**. The processor, according to the EnvClkx, decays EnvCnt. When EnvCnt becomes 0, the processor will calculate the next EDD (envelop) value, $EDD = EDD +/- EnvInc[6:0]$ based on **EnvSign(0/1)** bit. If EDD value reaches EnvTargetValue[6:0], the processor will automatically update the data in envelope 0/1 (Port 0x1010+0x40*X/0x1018+0x40*X).



Note : EnvClkx refer 0xE18~0xE24 and 0xE98~0xEA4 register description

- Example for Envelop Format without Envelop Repeat

N _{th} ADSR	Memory Offset[8:0] Address	Envelope0/1 (0x1010+0x40*X/0x1018+0x40*X)
0	0	Envelope 0
	1	Envelope 1
1	2	Envelope 0
	3	Envelope 1
2	4	Envelope 0
	5	Envelope 1
...



Suppose the waveform of envelop is shown as above. The start address of the envelope is as follows:

N _{th} ADSR	Memory Offset[8:0] Address	Envelop0/1 (0x1010+0x40*X/0x1018+0x40*X)
0	0	0x7F10
	1	0x0000
1	2	0x5090
	3	0x0000
2	4	0x5804
	5	0x0005
3	6	0x0090
	7	0x0001

1. In ADSR0, if user wants to increase the value from 0x00 to 0x7F in 8X units. The following values are obtained:
EnvTargetValue = 0x7F, EnvSign = 0, EnvInc = (0x7F-0)/8 = 0x10, EnvCnt = 0 (EDD counts each time unit).
2. In ADSR1, if user wants to decrease the value from 0x7F to 0x50 in 3X units. The following values are obtained:
EnvTargetValue = 0x50, EnvSign = 1, EnvInc = (0x7F-0x50)/3 = 0x10, EnvCnt = 0 (EDD counts each time unit).
3. In ADSR2, if user wants to increase the value from 0x50 to 0x58 in 12X units. The following values are obtained:
EnvTargetValue = 0x58, EnvSign = 0, EnvInc = (0x58-0x50)/2=0x04, EnvCnt = 5 (EDD counts each 6X units).
4. In ADSR3, if user wants to decrease the value from 0x58 to 0x00 in 12X units. The following values are obtained:
EnvTargetValue = 0x00, EnvSign = 1, EnvInc = (0x58-0x00)/6 = 0x10, EnvCnt = 1 (EDD counts each 2X units).

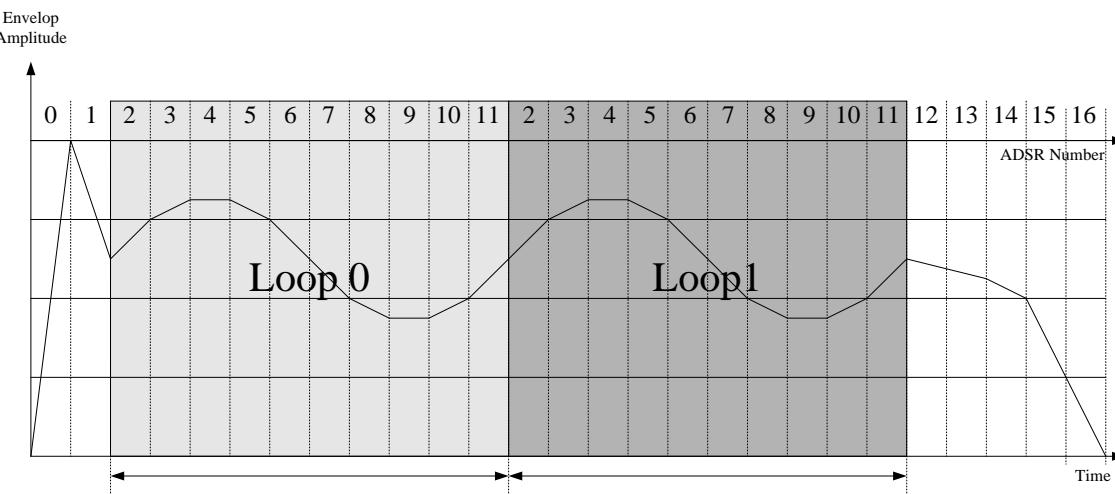
Note: For time unit definition, please refer to **Envelope Interval selection** (0x0E18 ~ 0x0E24 and 0x0E98 ~ 0x0EA4).

- Example for Envelop Format with Envelop Repeat

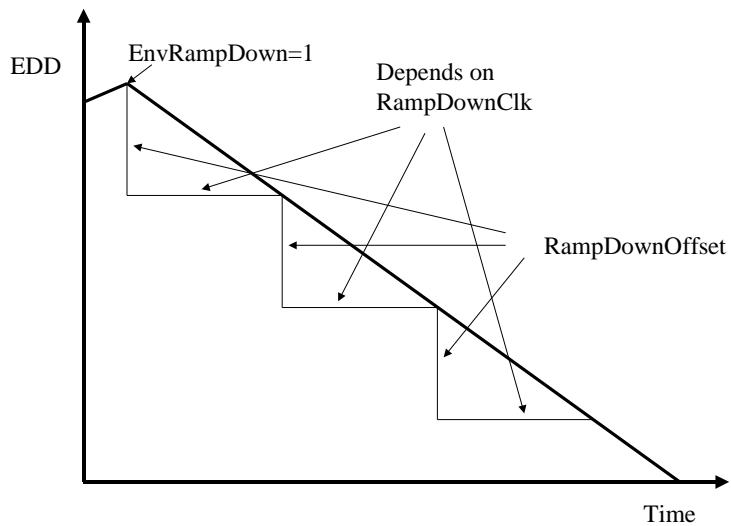
N _{th} ADSR	Memory Offset[8:0] Address	Envelop0/1 (0x1010+0x40*X/0x1018+0x40*X)
0	0	Envelope 0
	1	Envelope 1
1	2	Envelope 0
	3	Envelope 1
2	4	Envelope 0
	5	Envelope 1
3	6	Envelope 0
	7	Envelope 1
4	8	Envelope 0
	9	Envelope 1
5	A	Envelope 0
	B	Envelope 1
6	C	Envelope 0
	D	Envelope 1
7	E	Envelope 0
	F	Envelope 1
8	10	Envelope 0
	11	Envelope 1
9	12	Envelope 0
	13	Envelope 1
10	14	Envelope 0
	15	Envelope 1
11	16	Envelope 0
	17	Envelope 1 ^{*1}
	18	Envelop Loop Control ^{*2}
12	19	Envelope 0
	1A	Envelope 1
...

*1: Repeat = 1

*2: In this example, EAoffset[8:0] = 0x04, RpCnt[6:0] = 0x01.



When P_CtrEnvRampDown of a channel is set as 1, EnvRampDownClk will be read from P_ChRampDownClk, and Rampdownoffset will be read from P_ChEnvelopeLoop. After the reading, envelope will ramp down to 0 in 0 ~ 3 seconds, and the time period is determined by the values of EnvRampDownClk and RampDownOffset.



The Decreasing Value of EDD at Each Rramp Down Step

EnvRampDownClk	Envelope Ramp Down Step	Envelope Ramp Down Time
000	13 * 4*4 Frame	0.724ms
001	13 * 16*4 Frame	2.895ms
010	13 * 64*4 Frame	11.579ms
011	13 * 256*4 Frame	46.315ms
100	13 * 1024*4 Frame	185.259ms
101	13 * 4096*4 Frame	741.035ms
110	13 * 8192*4 Frame	1.482 s
111	13 * 8192*4 Frame	1.482 s

ENV_CTRL0									Envelope Control Register 0									
Bit	15	14	13	12	11	10	9	8										
Function	-	ENV_TARGET[6:0]																
Reset Value	0	0	0	0	0	0	0	0										
Bit	7	6	5	4	3	2	1	0										
Function	ENV_SIGN	ENV_STEP[6:0]																
Reset Value	0	0	0	0	0	0	0	0										

Bit	Function	Type	Description	Condition
[15]	-	R	Reserved	-
[14:8]	ENV_TARGET [6:0]	R/W	Envelope Target Value for Envelope Tracking	00~7F
[7]	ENV_SIGN	R/W	Envelope Sign Bit	0: Positive 1: Negative

Bit	Function	Type	Description					Condition
[6:0]	ENV_STEP[6:0]	R/W	Envelope Increment Value for Envelope Tracking EDD = EDD +/- ENV_STEP[6:0].					00~7F

ENV_CTRL1								
0x50081018								
Envelope Control Register 1								
Bit	15	14	13	12	11	10	9	8
Function	ENV_REPEAT_CN[6:0]							ENV_REPEAT_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENV_LOAD_CNT[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[15:9]	ENV_REPEAT_CN[6:0]	R/W	Envelope Repeat Counter					00~7F
[8]	ENV_REPEAT_EN	R/W	Envelope Repeat Control In repeat mode, the controller will refer to the value configured in ENV_REPEAT_OFFSET[7:0] of ENV_RAMPDOWN_STEP (0x1028+0x40*X), Eaoffset (Echo Offset), and RpCnt (Repeat Count).					0: Normal Mode 1: Repeat Mode
[7:0]	ENV_LOAD_CNT[7:0]	R/W	EnvLoad is the value loaded into ENV_COUNTER[7:0]. After ENV_COUNTER reaches 0, the ENV_LOAD_CNT is re-loaded automatically.					00~FF

ENV_DATA									
0x50081014									
Envelope Data									
Bit	15	14	13	12	11	10	9	8	
Function	ENV_COUNTER[7:0]								
Reset Value	0	0	0	0	0	0	0	0	
Bit	7	6	5	4	3	2	1	0	
Function	-	ENV_DATA[6:0]							
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[15:8]	ENV_COUNTER[7:0]	R/W	Envelop count is controlled by the hardware in Envelope auto mode. The clock source for envelop counter is from EnvClk.					00~FF
[7]	-	R	Reserved					-
[6:0]	ENV_DATA[6:0]	R/W	Envelope Data If envelope data reaches 0, the corresponding channel will be stopped automatically. (EDD)					00~7F

ENV_INT_CTRL
0x5008101C
Envelope Interrupt Control

Bit	15	14	13	12	11	10	9	8
Function	ENV_INT_ADDR[8:1]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0		
Function	ENV_INT_A DDR[0]	ENV_INT_E N	ENV_ADDR[21:16]							
Reset Value	0	0	0	0	0	0	0	0		

Bit	Function	Type	Description						Condition
[15:7]	ENV_INT_ADDR[8:0]	R/W	When REPEAT_OFFSET matches the ENV_INT_ADDR and ENV_INT_EN is set as 1, the envelope IRQ will be established and ENV_INT_STATUS_CHx will be set as 1.						-
[6]	ENV_INT_EN	R/W	Envelope IRQ Enable						0: Disable 1: Enable
[5:0]	ENV_ADDR[21:16]	R/W	Envelope Address						-

ENV_ADDR
0x50081020
Envelope Address

Bit	15	14	13	12	11	10	9	8
Function	ENV_ADDR[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	ENV_ADDR[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15:0]	ENV_ADDR[15:0]	R/W	Envelope Address						-

ENV_ADDRH
0x5008103C
Envelope Address

Bit	15	14	13	12	11	10	9	8	
Function	-								ENV_ADDR[30]
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0
Function	ENV_ADDR[29:22]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15:9]	-	R	Reserved						-
[8:0]	ENV_ADDR[30:22]	R/W	Envelope Address						-

ENV_RAMPDOWN_STEP								0x50081028	Envelope Rampdown Control
Bit	15	14	13	12	11	10	9	8	
Function	ENV_RAMPDOWN_STEP[6:0]								ENV_REPEAT_OFFSET[8:1]
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
Function	ENV_REPEAT_OFFSET[7:0]								
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[15:9]	ENV_RAMPDOWN_STEP[6:0]	R/W	Envelop Ramp Down Offset When the EnvRampDown of any channel is set as 1, EDD value will start to decrease by the configured value here.					0x00~0x7F
[8:0]	ENV_REPEAT_OFFSET[8:0]	R/W	Envelop Repeat Address Offset					0x00~0x1FF

ENV_RAMPDOWN_CLK								0x5008200C	Envelope Rampdown Clock
Bit	15	14	13	12	11	10	9	8	
Function	-								
Reset Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
Function	-					ENV_RAMPDOWN_CLK[2:0]			
Reset Value	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description					Condition
[15:3]	-	R	Reserved					-
[2:0]	ENV_RAMPDOWN_CLK[2:0]	R/W	Envelope Ramp Down Clock Selection					000: 0.724 ms 001: 2.895 ms 010: 11.579 ms 011: 46.315 ms 100: 185.259 ms 101: 741.035 ms 110: 1.482 s 111: 1.482 s

26.11.4. Phase Register

Use the following relation to get the phase of a sample rate.

$$\text{Phase} = \text{Sample-Rate} * 2^{19} / 287.425 \text{ kHz}$$

PHASE_VALUEH
0x50082000
Phase Adder Value

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15:3]	-	R	Reserved						-
[2:0]	PHASE_VALUE[18:16]	R/W	Phase = Sample-Rate * 2^{19} / 287.425 kHz Phase value is the 19-bit tone color phase which combines PHASE_VALUEH[2:0] and PHASE_VALUE[15:0].						-

PHASE_VALUE
0x50082010
Phase Adder Value

Bit	15	14	13	12	11	10	9	8
Function	PHASE_VALUE[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PHASE_VALUE[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15:0]	PHASE_VALUE[15:0]	R/W	Phase = Sample-Rate * 2^{19} / 287.425 kHz Phase value is the 19-bit tone color phase which combines PHASE_VALUEH[2:0] and PHASE_VALUE[15:0].						-

PHASE_ACCH
0x50082004
Tone-Color Pitch Accumulator

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[15:3]	-	R	Reserved						-
[2:0]	PHASE_ACC[18:16]	R/W	Combine P_PHASE_ACCH with P_PHASE_ACC to get the tone-color pitch accumulator PHASE_ACC [18:0].						-

PHASE_ACC 0x50082014 Tone-Color Pitch Accumulator

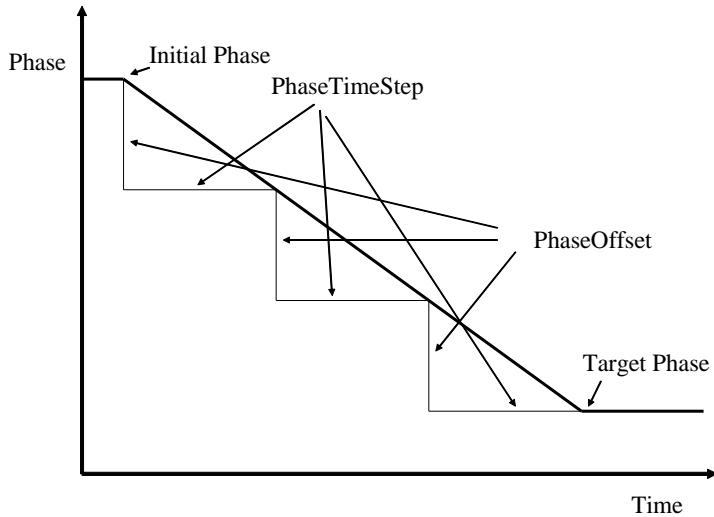
Bit	15	14	13	12	11	10	9	8
Function	PHASE_ACC[15:8]							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	PHASE_ACC[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[15:0]	PHASE_ACC[15:0]	R/W	Combine P_PHASE_ACCH with P_PHASE_ACC to get - the tone-color pitch accumulator PHASE_ACC [18:0].				-

26.11.5. PitchBend Register

If PitchBendEn[x] is 1, the channel will use PhaseOffset, PhaseSign, and PhaseTimeStep to increase/decrease Phase to the TargetPhase.



PITCHBEND_CTRL									0x5008201C									Pitch Bend Control									
Bit	15	14	13	12	11	10	9	8																			
Function	PITCHBEND_CLK[2:0]									PITCHBEND_SIGN	PITCHBEND_STEP[11:8]																
Reset Value	0	0	0	0	0	0	0	0																			

Bit	7	6	5	4	3	2	1	0
Function	PITCHBEND_STEP[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:13]	PITCHBEND_CLK[2:0]	R/W	This register is used to control the period of phase change.	000: 0.111ms 001: 0.223ms 010: 0.445ms 011: 0.890ms 100: 1.781ms 101: 3.563ms 110: 7.125ms 111: 14.250ms
[12]	PITCHBEND_SIGN	R/W	This is the increase or decrease selection of phase.	0: Increase phase. 1: Decrease phase.
[11:0]	PITCHBEND_STEP[11:0]	R/W	When pitchbend function is enabled, this bit value will be added or subtracted every time.	-

PITCHBEND_TARGETH
0x50082008
Pitch Bend Target Phase

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	PITCHBEND_TARGETH[18:16]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]	-	R	Reserved	-
[2:0]	PITCHBEND_TARGETH[18:16]	R/W	Pitch Bend Target Phase Combine P_PITCHBEND_TARGETH with P_PITCHBEND_TARGET to get the pitch bend target phase PITCHBEND_TARGET [18:0].	-

PITCHBEND_TARGET
0x50082018
Pitch Bend Target Phase

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	PITCHBEND_TARGET[15:8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	PITCHBEND_TARGET[7:0]
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	PITCHBEND_TARGET[15:0]	R/W	Pitch Bend Target Phase Combine P_PITCHBEND_TARGETH with P_PITCHBEND_TARGET to get the pitch bend target phase PITCHBEND_TARGET [18:0].	-

26.12. Programming Tips

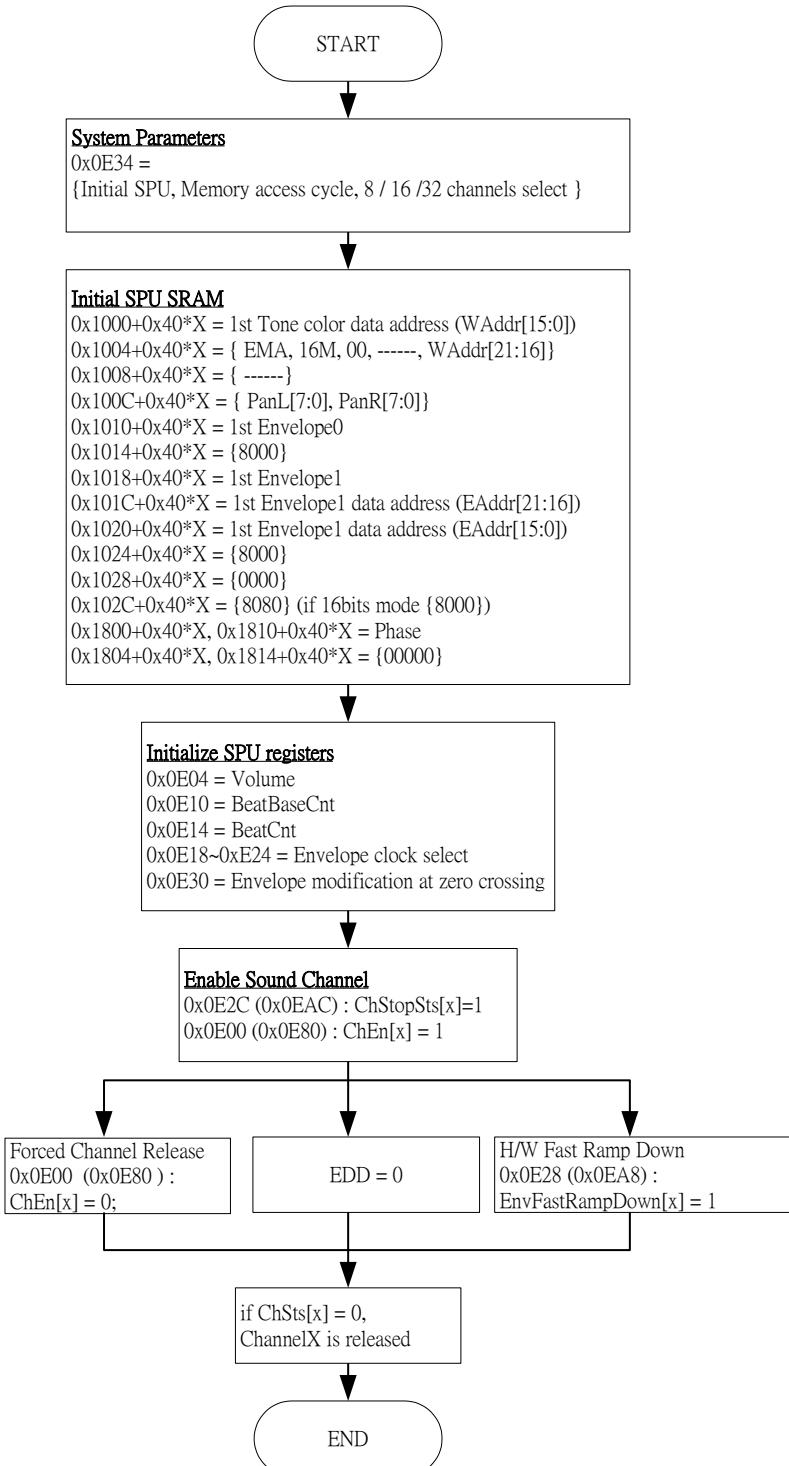
When user initializes SPU SRAM, the data must be written into Attribute SRAM and Phase SRAM. The Port[0x1014+0x40*X] and Port[0x1028+0x40*X] should be set as 0. For 8-bit tone color applications, Port[0x1024+0x40*X] = 0x8000 (middle level), and Port[0x102C+0x40*X] = 0x8080 or the first tone-color. For 16-bit tone color applications, Port[0x1024+0x40*X] equals to 0x8000 (middle level), and Port[0x102C+0x40*X] is 0x8000 or the first tone-color. Besides, "1" should be written into Port[0xE34] Bit[3] to set up the related SRAM initial value.

Note: During the time that beat event is updating data, make sure the corresponding bit is 0 in 0xE3C (ChSts). *Otherwise, the hardware control will be unpredictable.*

There are four ways to disable ChSts[x]:

- Disable ChEn[x] and the channel will be disabled in the next tone-color zero-cross.
- When the envelop data EDD(0x1014+0x40*X) is 0.
- The S/W writes EnvRampDown[x] = 1. The H/W will ramp down according to the speed defined in EnvRampDownClk[2:0] and the offset defined in EnvRampDownOffset[6:0] until the envelope value equals 0.
- When tone-color reaches the end code (0xFF) in the H/W auto-end mode.

26.13. Programming Flow Example



27. Universal Serial Bus (USB)

27.1. Introduction

GPCM3 series supports 6 buffers, Buffer 0 ~ Buffer 5, for USB function. These buffers can be utilized for different hardware interface, including CPU. Each buffer supports read/write operations in byte mode (8-bit), half word mode (16-bit), and word mode (32-bit) when it is linked to other devices. For buffer size, the size of Buffer 4 is 256 bytes which can support 48KHz sample rate, 16-bit resolution stereo Isochronous OUT transfer. The size of other buffers is 64 bytes. For more information about the memory mapping for each buffer, please refer to USB buffer mapping. Note that only one buffer is allowed to be linked with CPU for CPU reading and writing at the same time, and other buffers must be kept in the state of being linked to USB.

27.2. Features

- USB Specification 2.0 Compliant (USB 1.1 Version) ; Full-Speed (12 Mbps)
- USB supports Bulk IN, Bulk OUT, Interrupt IN, Isochronous IN, Isochronous OUT transfer mode, and Vendor command.

27.3. Serial Interface Control Pin Configuration

Name	I/O	Description
USB_DP	IOD0	USB D+ pin
USB_DM	IOD1	USB D- pin

27.4. USB Buffer Mapping

Buffer Index	Location	Size
Buffer 0	0x4018_0400 ~ 0x4018_043F	64 Bytes
Buffer 1	0x4018_0440 ~ 0x4018_047F	64 Bytes
Buffer 2	0x4018_0480 ~ 0x4018_04BF	64 Bytes
Buffer 3	0x4018_0500 ~ 0x4018_053F	64 Bytes
Buffer 4	0x4018_0600 ~ 0x4018_06FF	256 Bytes
Buffer 5	0x4018_0700 ~ 0x4018_073F	64 Bytes

27.5. Register Description

27.5.1. Register Map

Name	Address	Description
USB_UP	0x4017_0004	USB Up-Stream Control and Status Register
USB_DOWN	0x4017_0008	USB Down-Stream Control and Status Register
USB_EOT	0x4017_000C	End of Transaction Control Register
USB_BC2	0x4017_0010	Byte Count for Endpoint 2 Register
USB_BC1	0x4017_0014	Byte Count for Endpoint 1 Register
USB_BC0	0x4017_0018	Byte Count for Endpoint 0 Register
USB_STS	0x4017_001C	USB Status Register
USB_EPEN	0x4017_0020	USB Endpoint Buffer Linking Enable Register
USB_PHYCMD	0x4017_0028	USB Physical Layer Command Register
USB_BC4	0x4017_0030	Byte Count for Endpoint 4 Register
USB_BC5	0x4017_0034	Byte Count for Endpoint 5 Register
USB_BC3	0x4017_0038	Byte Count for Endpoint 3 Register
USB_INTERFACE	0x4017_003C	USB Interface Status

Name	Address	Description
USB2_BUFEN	0x4018_0000	Buffer Active Control Register
USB2_BUF10_LINK	0x4018_0004	Buffer 0 and Buffer 1 Link Control Register
USB2_BUF32_LINK	0x4018_0008	Buffer 2 and Buffer 3 Link Control Register
USB2_BUF54_LINK	0x4018_000C	Buffer 4 and Buffer 5 Link Control Register
USB2_PHYCTRL	0x4018_0010	USB PHY Control Register
USB2_CTRL	0x4018_0014	USB System Control Register
USB2_INTEN	0x4018_0018	USB Interrupt Enable Register
USB2_INTSTS	0x4018_001C	USB Interrupt Status Register

27.5.2. Register Function

USB_UP									0x4017_0004	USB Up-Stream Control and Status Register								
Bit	31	30	29	28	27	26	25	24										
Function									-									
Reset Value	0									0	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16										
Function									-									
Reset Value	0									0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8										
Function									-									
Reset Value	0									0	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0										
Function									EP3_UP_ST REAM									
Reset Value	0									0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description								Condition
[31:4]	-	R	Reserved								-
[3]	EP3_UP_STREAM	R/W	Endpoint 3 Upstream Buffer Control Bit This bit will be cleared automatically after the upstream transaction has been completed successfully. The upstream buffer is assigned to buffer5.								Read 0: The data upload is completed. Read 1: The data is still uploading. Write 0: No effect. Write 1: Start uploading data to the host.
[2]	EP2_UP_STREAM	R/W	Endpoint 2 Upstream Buffer Control Bit This bit will be cleared automatically after the upstream transaction has been completed successfully. The upstream buffer is assigned to buffer1 or 2.								Read 0: The data upload is completed. Read 1: The data is still uploading. Write 0: No effect. Write 1: Start uploading data to the host.

Bit	Function	Type	Description	Condition
[1]	EP1_UP_STREAM	R/W	Endpoint 1 Upstream Buffer Control Bit This bit will be cleared automatically after the upstream transaction has been completed successfully. The upstream buffer is assigned to buffer1 or 2.	Read 0: The data upload is completed. Read 1: The data is still uploading. Write 0: No effect. Write 1: Start uploading data to the host.
[0]	EPO_UP_STREAM	R/W	Endpoint 0 Upstream Buffer Control Bit This bit will be cleared automatically after the upstream transaction has been completed successfully. The upstream buffer is assigned to buffer0.	Read 0: The data upload is completed. Read 1: The data is still uploading. Write 0: No effect. Write 1: Start uploading data to the host.

Note:

1. Write USB_UP[x] = 1: CPU has written data in the buffer, and then waits for the H/W to transfer data to the Host.
2. Read USB_UP[x] = 0: Upstream transaction has been completed successfully.
3. Up flag is configured by the F/W (transfer start) and is cleared by the H/W (transfer ends and the Host replies ACK).
4. If the up flag hasn't been clear by the H/W for more than 1ms, user must clear the corresponding flag manually and retry this transaction.

USB_DOWN								0x4017_0008	USB Down-Stream Control and Status Register								
Bit	31	30	29	28	27	26	25	24									
Function									-								
Reset Value	0	0	0	0	0	0	0	0									

Bit	23	22	21	20	19	18	17	16
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function								
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					EP3_DOWN_STREAM	EP2_DOWN_STREAM	EP1_DOWN_STREAM	EPO_DOWN_STREAM
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:4]	-	R	Reserved				-

Bit	Function	Type	Description	Condition
[3]	EP3_DOWN_STREAM	R/W	Endpoint 3 Downstream Flag The downstream buffer is assigned to buffer4.	Read 0: The data is still downloading. Read 1: Download is completed. Write 0: Clear the flag. Write 1: No Effect.
[2]	EP2_DOWN_STREAM	R/W	Endpoint 2 Downstream Flag The downstream buffer is assigned to buffer1 or 2.	Read 0: The data is still downloading. Read 1: Download is completed. Write 0: Clear the flag. Write 1: No Effect.
[1]	EP1_DOWN_STREAM	R/W	Endpoint 1 Downstream Flag The downstream buffer is assigned to buffer1 or 2.	Read 0: The data is still downloading. Read 1: Download is completed. Write 0: Clear the flag. Write 1: No Effect.
[0]	EP0_DOWN_STREAM	R/W	Endpoint 0 Downstream Flag The downstream buffer is assigned to buffer0.	Read 0: The data is still downloading. Read 1: Download is completed. Write 0: Clear the flag. Write 1: No Effect.

Note:

1. Down flag is configured by the H/W and is cleared by the F/W.
2. If the host issues write command to the device, the H/W will receive data in downstream buffer and then set the corresponding Down flag as 1 automatically.
3. The F/W has to detect Rx (downstream) transaction by itself, and after the transaction is done, clear the Down flag. Otherwise if Down flag = 1, the host will keep receiving NAK command until Down flag = 0.

USB_EOT									0x4017_000C									End of Transaction Control Register								
Bit	31	30	29	28	27	26	25	24																		
Function									-																	
Reset Value	0	0	0	0	0	0	0	0																		

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-		EP2EOT_FL AG	EP1EOT_FL AG	EPOEOT_FL AG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:3]	-	R	Reserved	-
[2]	EP2EOT_FLAG	W	End of Endpoint 2 Transaction	Write 0: No effect. Write 1: Inform the system that the last data has been written in Up-stream buffer.
[1]	EP1EOT_FLAG	W	End of Endpoint 1 Transaction	Write 0: No effect. Write 1: Inform the system that the last data has been written in Up-stream buffer.
[0]	EPOEOT_FLAG	W	End of Endpoint 0 Transaction	Write 0: No effect. Write 1: Inform the system that the last data has been written in Up-stream buffer.

Note:

1. EOT flag is configured by the F/W and is cleared by the H/W.
2. Write `USB_EOT [x] = 1`: When the last data is written in Up-stream buffer, user has to write 1 to the corresponding `USB_EOT [x]`.
3. When transaction data length is less than 64 byte, user also has to write 1 to the corresponding `USB_EOT [x]`.

0x4017_0010								Byte Count for Endpoint 2 Register	
Bit	31	30	29	28	27	26	25	24	
Function				-					
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				BC2[8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	BC2[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:0]	BC2[8:0]	R/W	Byte Count of Endpoint 2					This value should be between 0 ~ 256.

USB_BC1								
0x4017_0014								
Bit	31	30	29	28	27	26	25	24
Function						-		
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			BC1[8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					BC1[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:0]	BC1[8:0]	R/W	Byte Count of Endpoint 1					This value should be between 0 ~ 256.

USB_BCO								
0x4017_0018								
Bit	31	30	29	28	27	26	25	24
Function						-		
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			BC0[8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	BC0[7:0]							
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:0]	BC0[8:0]	R/W	Byte count of Endpoint 0					This value should be between 0 ~ 256.

USB_STS								
0x4017_001C								
Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-	RST_FLAG	SUSPEND_FLAG	SETUP_FLAG	STALL_FLAG
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:4]	-	R	Reserved					-
[3]	RST_FLAG	R/W	This bit indicates whether the device receives Reset command from the host or not. Write 0 to clear the flag.					Read 0: Not receive Reset command from the host. Read 1: Receive Reset command from the host. Write 0: Clear the reset signal. Write 1: No effect.
[2]	SUSPEND_FLAG	R/W	This bit indicates whether the device receives Suspend command from the host or not. Write 0 to clear the flag.					Read 0: Not receive Suspend command from the host. Read 1: Receive Suspend command from the host. Write 0: Clear the suspend signal. Write 1: No effect.
[1]	SETUP_FLAG	R	This bit indicates whether the SIE (Device) has received a SETUP token or not.					Read 0: Not receive SETUP command from the host. Read 1: Receive SETUP command from the host.

Bit	Function	Type	Description					Condition
[0]	STALL_FLAG	R	If this flag =1, it means the STALL command from the Host has been received.					Read 0: Not receive STALL command from the host. Read 1: Receive STALL command from the host.

USB_EPEN **0x4017_0020** **USB Endpoint Buffer Linking Enable Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-		EP5_EN	EP4_EN	EP3_EN	EP2_EN	EP1_EN	EPO_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:6]	-	R	Reserved					-
[5]	EP5_EN	R/W	Read interrupt enable for ISO IN transfer (EP5).					0: Disabled 1: Enabled
[4]	EP4_EN	R/W	Write interrupt enable for ISO OUT transfer (EP4).					0: Disabled 1: Enabled
[3]	EP3_EN	R/W	EP3_UP_STREAM/EP3_DOWN_STREAM enable for interrupt (EP3).					0: Disabled 1: Enabled
[2]	EP2_EN	R/W	EP2_UP_STREAM/EP2_DOWN_STREAM/EP2EOT_FLAG enable for bulk out transfer (EP2 or EP1).					0: Disabled 1: Enabled
[1]	EP1_EN	R/W	EP1_UP_STREAM/EP1_DOWN_STREAM/EP1EOT_FLAG enable for bulk in transfer (EP2 or EP1).					0: Disabled 1: Enabled
[0]	EPO_EN	R/W	EPO_UP_STREAM/EPO_DOWN_STREAM/EPOEOT_FLAG enable for control transfer (EPO).					0: Disabled 1: Enabled

USB_PHYCMD **0x4017_0028** **USB Physical Layer Command Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-	BUR_RES	SIE_FIFO_RST	INTERFACE_RST	-	DEVICE_STALL	ADDR_CLR	SIE_RST
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:7]	-	R	Reserved	-
[6]	BUR_RES	W	Reset the burst mode counter.	Write 0: Useless Write 1: Reset the burst mode counter.
[5]	SIE_FIFO_RST	W	FIFO Reset This bit will reset the circuit of FIFO and counter between CPU and SIE.	Write 0: Useless Write 1: Reset Endpoint register setting.
[4]	INTERFACE_RST	W	Reset USB Interface When user writes 1 to this bit, all related registers will return to default value.	Write 0: Useless Write 1: Reset USB interface Register.
[3]	-	R	Reserved	-
[2]	DEVICE_STALL	R/W	Write 1 to send stall to the host. This bit will be cleared by the hardware when device stall is cleared by the host.	Read: Read D_STALL's value. Write 0: Useless Write 1: The device is stalled.
[1]	ADDR_CLR	R/W	Clear the address counter.	Read: Read CLR_ADDR's value. Write 0: Useless Write 1: Clear the address counter.
[0]	SIE_RST	R/W	Reset PLL in SIE This bit must set as 0 to initial CPU.	Read: Read RES_DPLL's value. Write 0: Release reset signal. Write 1: Reset PLL in SIE.

USB_BC4	0x4017_0030								Byte Count for Endpoint 4 Register
Bit	31	30	29	28	27	26	25	24	
Function	-	-	-	-	-	-	-	-	
Reset Value	0	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
Function	-	-	-	-	-	-	-	-
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				BC4[8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					BC4[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:0]	BC4[8:0]	R/W	Byte Count of Endpoint 4					This value should be between 0 ~ 256.

USB_BC5 **0x4017_0034** **Byte Count for Endpoint 5 Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				BC5[8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					BC5[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:0]	BC5[8:0]	R/W	Byte Count of Endpoint 5					This value should be between 0 ~ 256.

USB_BC3 **0x4017_0038** **Byte Count for Endpoint 3 Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				BC3[8]
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					BC3[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:9]	-	R	Reserved					-
[8:0]	BC3[8:0]	R/W	Byte Count of Endpoint 3					This value should be between 0 ~ 256.

USB_INTERFACE **0x4017_003C** **USB Interface Status**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					INT2ALT1	INT2ALTO	INT1ALT1	INT1ALTO
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:4]	-	R	Reserved					-
[3]	INT2ALT1	R/W	The host issues set interface 2 & AlternateSetting1.					Read 0: The host doesn't send "set interface" command. Read 1: The host sends "set interface" command. Write 0: Clear the flag. Write 1: No effect.
[2]	INT2ALTO	R/W	The host issues set interface 2 & AlternateSetting0.					Read 0: The host doesn't send "set interface" command. Read 1: The host sends "set interface" command. Write 0: Clear the flag. Write 1: No effect.

Bit	Function	Type	Description					Condition
[1]	INT1ALT1	R/W	The host issues set interface 1 & AlternateSetting1.					Read 0: The host doesn't send "set interface" command. Read 1: The host sends "set interface" command. Write 0: Clear the flag. Write 1: No effect.
[0]	INT1ALTO	R/W	The host issues set interface 1 & AlternateSetting0.					Read 0: The host doesn't send "set interface" command. Read 1: The host sends "set interface" command. Write 0: Clear the flag. Write 1: No effect.

USB2_BUFEN									0x4018_0000									Buffer Active Control Register									
Bit	31	30	29	28	27	26	25	24																			
Function	-		-		-		-		-		-		-		-		-		-		-		-		-		
Reset Value	0	0	0	0	0	0	0	0																			

Bit	23	22	21	20	19	18	17	16
Function	-		-		-		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-		-		-		-	
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-		BUF5_EN	BUF4_EN	BUF3_EN	BUF2_EN	BUF1_EN	BUFO_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:6]	-	R	Reserved					-
[5]	BUF5_EN	R/W	Buffer5 Active Control This bit is set as 1 to active buffer5 R/W function.					Read: Read buffer enable/disable status. Write 0: Buffer Disabled Write 1: Buffer Enabled
[4]	BUF4_EN	R/W	Buffer4 Active Control This bit is set as 1 to active buffer4 R/W function.					Read: Read buffer enable/disable status. Write 0: Buffer Disabled Write 1: Buffer Enabled
[3]	BUF3_EN	R/W	Buffer3 Active Control This bit is set as 1 to active buffer3 R/W function.					Read: Read buffer enable/disable status. Write 0: Buffer Disabled Write 1: Buffer Enabled

Bit	Function	Type	Description	Condition
[2]	BUF2_EN	R/W	Buffer2 Active Control This bit is set as 1 to active buffer2 R/W function.	Read: Read buffer enable/disable status. Write 0: Buffer Disabled Write 1: Buffer Enabled
[1]	BUF1_EN	R/W	Buffer1 Active Control This bit is set as 1 to active buffer1 R/W function.	Read: Read buffer enable/disable status. Write 0: Buffer Disabled Write 1: Buffer Enabled
[0]	BUFO_EN	R/W	Buffer0 Active Control This bit is set as 1 to active buffer0 R/W function.	Read: Read buffer enable/disable status. Write 0: Buffer Disabled Write 1: Buffer Enabled

Table 27-1: Buffer Linking Table

BUFx_LINK[2:0]	Linked Peripheral
000b	USB
011b	CPU

USB2_BUF10_LINK **0x4018_0004** **Buffer 0 and Buffer 1 Link Control Register**

Bit	31	30	29	28	27	26	25	24
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function					-			
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	-		BUF1_LINK[2:0]		-		BUFO_LINK[2:0]	
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:7]	-	R	Reserved				-
[6:4]	BUF1_LINK[2:0]	R/W	Buffer1 Link Selection				Refer to Table 27-1: Buffer Linking Table.
[3]	-	R	Reserved				-
[2:0]	BUFO_LINK[2:0]	R/W	Buffer0 Link Selection				Refer to Table 27-1: Buffer Linking Table.

USB2_BUF32_LINK
0x4018_0008
Buffer 2 and Buffer 3 Link Control Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	BUF3_LINK[2:0]				BUF2_LINK[2:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:7]	-	R	Reserved						-
[6:4]	BUF3_LINK[2:0]	R/W	Buffer3 Link Selection						Refer to Table 27-1: Buffer Linking Table.
[3]	-	R	Reserved						-
[2:0]	BUF2_LINK[2:0]	R/W	Buffer2 Link Selection						Refer to Table 27-1: Buffer Linking Table.

USB2_BUF54_LINK
0x4018_000C
Buffer 4 and Buffer 5 Link Control Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	BUF5_LINK[2:0]				BUF4_LINK[2:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description						Condition
[31:7]	-	R	Reserved						-
[6:4]	BUF5_LINK[2:0]	R/W	Buffer5 Link Selection						Refer to Table 27-1: Buffer Linking Table.
[3]	-	R	Reserved						-

Bit	Function	Type	Description					Condition
[2:0]	BUF4_LINK[2:0]	R/W	Buffer4 Link Selection					Refer to Table 27-1: Buffer Linking Table.

USB2_PHYCTRL **0x4018_0010** **USB PHY Control Register**

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function					PHY_CTRL[7:0]			
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:8]	-	R	Reserved					-
[7:0]	PHY_CTRL[7:0]	R/W	Tx Driver Rising/Falling Time Control					-

USB2_CTRL **0x4018_0014** **USB System Control Register**

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				ISO_EN	SUSPEND_E N	-	PULLH_EN	PYH_EN
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description					Condition
[31:5]	-	R	Reserved					-

Bit	Function	Type	Description	Condition
[4]	ISO_EN	R/W	ISO Mode Control Note: To activate ISO IN and ISO OUT, ISO_EN must be set as 1.	Write 0: USB ISO mode is disabled. Write 1: USB ISO mode is enabled.
[3]	SUSPEND_EN	R/W	USB Wake Up Control	Read: Read enable flag. Write 0: Disable UDC Suspend function. Write 1: Enable UDC Suspend function.
[2]	-	R	Reserved	-
[1]	PULLH_EN	R/W	D+ Pull-High Resister Enable	Read: Read enable flag. Write 0: Disable Write 1: Enable
[0]	PYH_EN	R/W	USB PHY Enable Control	Read: Read enable flag. Write 0: Disable Write 1: Enable

USB2_INTEN
0x4018_0018
USB Interrupt Enable Register

Bit	31	30	29	28	27	26	25	24
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function	-							
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function	SIE_EN	-				UP_STREA M_INT_EN	DN_STREA M_INT_EN	USB_INT_E N
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[31:8]	-	R	Reserved	-
[7]	SIE_EN	R/W	SIE (Serial Interface Engine) Enable	0: USB PHY is disabled. 1: USB PHY is enabled.
[6:3]	-	R	Reserved	-
[2]	UP_STREAM_INT_EN	R/W	USB Up Stream (Read) Interrupt Enable	0: Disabled 1: Enabled
[1]	DN_STREAM_INT_EN	R/W	USB Down Stream (Write) Interrupt Enable	0: Disabled 1: Enabled

Bit	Function	Type	Description				Condition
[0]	USB_INT_EN	R/W	USB Interrupt Enable				0: Disabled 1: Enabled

USB2_INTSTS **0x4018_001C** **USB Interrupt Status Register**

Bit	31	30	29	28	27	26	25	24
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Function				-				
Reset Value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Function				-		UP_STREA M_INTF	DN_STREA M_INTF	USB_INTF
Reset Value	0	0	0	0	0	0	0	0

Bit	Function	Type	Description				Condition
[31:3]	-	R	Reserved				-
[2]	UP_STREAM_INTF	R/W	USB Up Stream (Read) Interrupt Status Flag				Read: Read interrupt flag. Write 1: Clear the flag.
[1]	DN_STREAM_INTF	R/W	USB Down Stream (Write) Interrupt Status Flag				Read: Read interrupt flag. Write 1: Clear the flag.
[0]	USB_INTF	R/W	USB Interrupt Status Flag				Read: Read interrupt flag. Write 1: Clear the flag.

28. Emulation Board (EMU) and Piggyback

28.1. Introduction

Generalplus provides developer two hardware boards: EMU board and Piggyback. The EMU board provides more application circuits than Piggyback does, e.g. DAC application circuit. On the other hand, Piggyback is used for demonstration purpose, so the size of the board will be as small as possible. However, it still includes all the pin outs and functions. These two boards are able to emulate GPCM3 development environment. User can obtain these boards by contacting Generalplus.

28.2. GPCM3 Piggyback

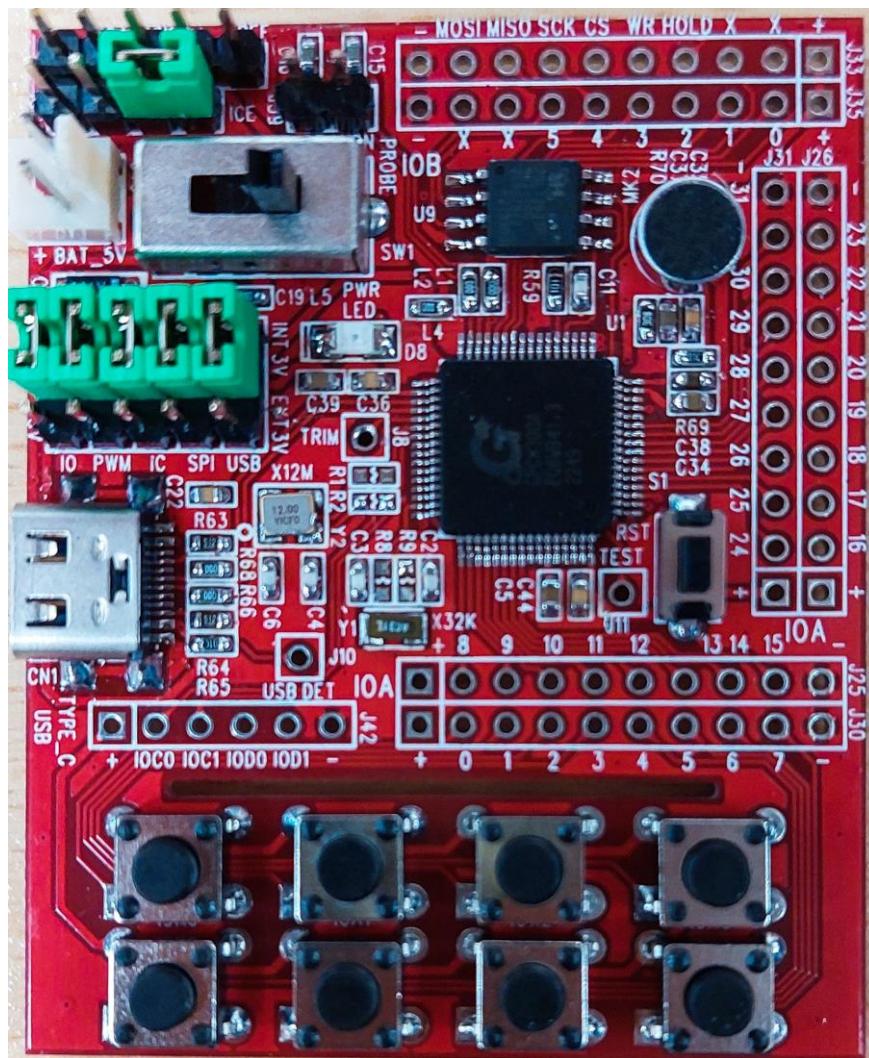


Figure 28-1 GPCM3 Piggyback

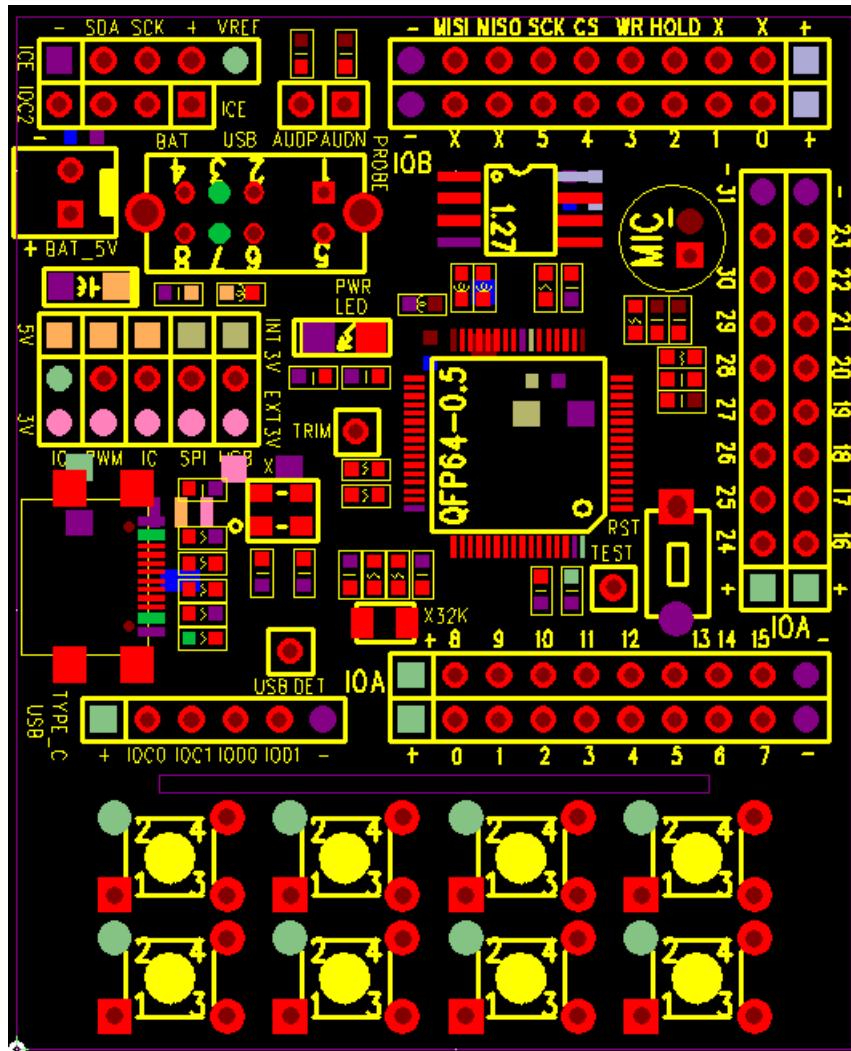
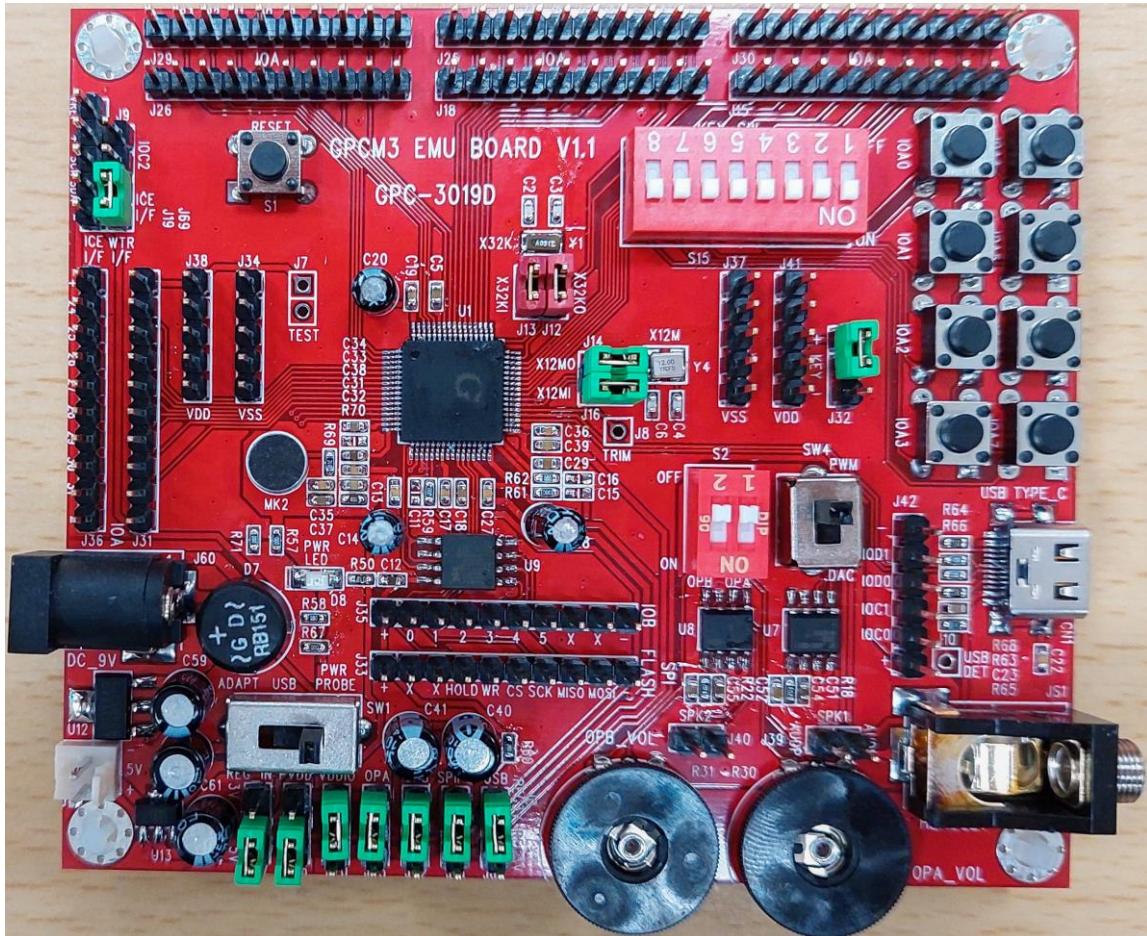


Figure 28-2 GPCM3 Piggyback Function Description

1. Jumper Function:
 - (a) J19: ICE I/F
 - (b) J25: IOA[15:8]
 - (c) J26: IOA[23:16]
 - (d) J30: IOA[7:0]
 - (e) J31: IOA[31:24]
 - (f) J33: SPIFC I/F
 - (g) J35: IOB[5:0]
 - (h) J39: Speaker Output
 - (i) J42: IOC[1:0] & IOD[1:0]
 - (j) J49: DC5V Power Input
 - (k) J66: PVDD/DAC Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V
 - (l) J67: VDD_REGIN Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V

- (m) J68: USB Power Selection Pins:
 - (1) 1-2: Internal Regulator 3V
 - (2) 2-3: External Regulator 3V
 - (n) J69: IOC2 Pin Function Selection:
 - (1) 1-2: ICE I/F
 - (2) 2-3: General IO (IOC2)
 - (o) J70: VDDIO Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V
 - (p) J71: SPIFC Power Selection Pins:
 - (1) 1-2: Internal Regulator 3V
 - (2) 2-3: External Regulator 3V
2. Switch Function:
- (a) S1: Reset Pin
 - (b) SW1: Power Source Selection
 - (1) 1-5 (Right): G+Link Pro Power
 - (2) 3-7 (Middle): USB Power
 - (3) 4-8 (Left): DC 5V Power

28.3. GPCM3 EMU Board



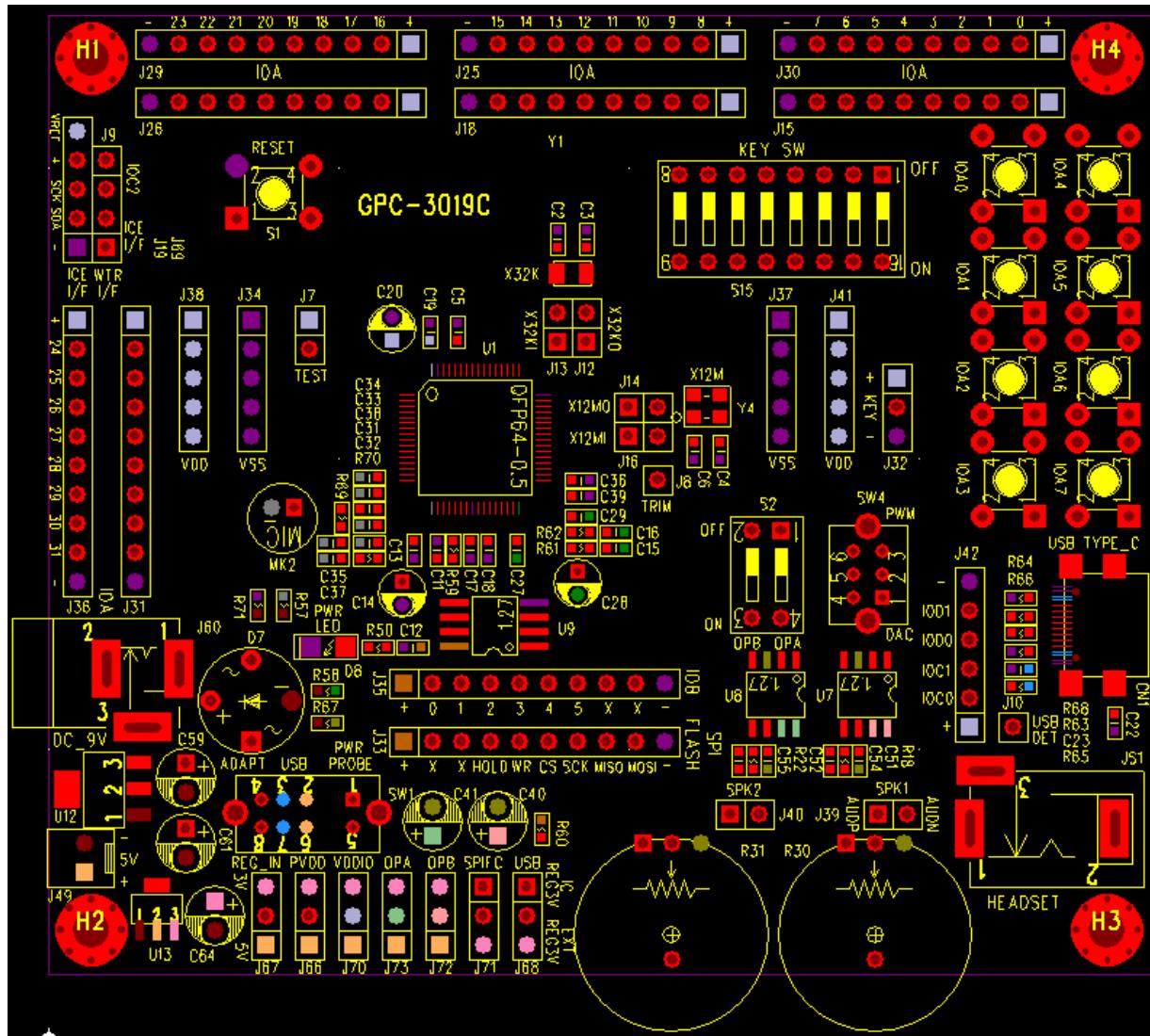


Figure 28-4 GPCM3 EMU Board Function Description

1. Jumper Function:

- J12 & J13: X'TAL 32KHz Input Selection Pins
- J14 & J16: X'TAL 12MHz Input Selection Pins
- J19: ICE I/F
- J33: SPIFC I/F
- J15 & J30: IOA[7:0]
- J18 & J25: IOA[15:8]
- J26 & J29: IOA[23:16]
- J31 & J36: IOA[31:24]
- J35: IOB[5:0]
- J42:IOC[1:0] & IOD[1:0]
- J39: Speaker Output 1 (PWM or OPA_DAC Out)
- J40: Speaker Output 2 (OPB_DAC Out)
- J49: DC 5V Power Input
- J60: DC 9V Jack
- J66: PVDD/DAC Power Selection Pins:
 - 1-2: DC 5V
 - 2-3: DC 3V

- (p) J67: VDD_REGIN Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V
 - (q) J68: USB Power Selection Pins:
 - (1) 1-2: Internal Regulator 3V
 - (2) 2-3: External Regulator 3V
 - (r) J69: IOC2 Pin Function Selection:
 - (1) 1-2: ICE I/F
 - (2) 2-3: General IO (IOC2)
 - (s) J70: VDDIO Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V
 - (t) J71: SPIFC Power Selection Pins:
 - (1) 1-2: Internal Regulator 3V
 - (2) 2-3: External Regulator 3V
 - (u) J72: GPY0030C OPA Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V
 - (v) J73: GPY0030C OPB Power Selection Pins:
 - (1) 1-2: DC 5V
 - (2) 2-3: DC 3V
2. Switch Function:
- (a) S1: Reset Pin
 - (b) S2: DAC OPA & OPB ON/OFF Selection
 - (1) 1: OPA ON/OFF
 - (2) 2: OPB ON/OFF
 - (c) S15: IOA[7:0] Key Switch Enabling/Disabling Selection
 - (d) SW1: Power Source Selection
 - (1) 1-5 (Right): G+Link Pro Power
 - (2) 3-7 (Middle): USB Power
 - (3) 4-8 (Left): Adapter DC 9V=> 5V Power
 - (e) SW4: Speaker Output 1 Source Selection:
 - (1) PWM Out
 - (2) DAC Out
3. CONN & Others:
- (a) CN1: USB TYPE-C Socket
 - (b) R30: OPA_DAC Volume Adjustment Resistor
 - (c) R31: OPB_DAC Volume Adjustment Resistor
 - (d) RS1: Headset Jack