



GPCM1F Series Programming Guide

ARM® Cortex®-M0 32-bit

Version 1.5 – Nov. 30, 2022

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
<i>REVISION HISTORY</i>	7
1. General Description.....	8
2. Features	8
2.1. <i>GPCM1F Block Diagram</i>	9
2.2. <i>GPCM1F IO Special Functions.....</i>	10
3. Memory Organization.....	11
4. ARM® Cortex®-M0 Core.....	12
4.1. <i>Overview</i>	12
4.2. <i>System Control Block</i>	13
4.3. <i>Registers.....</i>	13
4.4. <i>System timer, SysTick.....</i>	18
4.5. <i>Register</i>	19
5. Nested Vectored Interrupt Controller (NVIC).....	21
5.1. <i>Overview</i>	21
5.2. <i>Features.....</i>	21
5.3. <i>Exception Types</i>	21
6. SYSTEM Descriptions	23
6.1. <i>Reset System</i>	23
6.1.1. <i>Introduction</i>	23
6.1.2. <i>Block Diagram</i>	23
6.1.3. <i>Function.....</i>	23
6.1.4. <i>Register Description.....</i>	24
6.2. <i>Clock Control Unit.....</i>	26
6.2.1. <i>Introduction</i>	26
6.2.2. <i>Block Diagram</i>	26
6.2.3. <i>Register Description.....</i>	27
6.3. <i>Low Power Mode.....</i>	31
6.3.1. <i>Introduction</i>	31
6.3.2. <i>Function</i>	31
6.4. <i>System Management Unit.....</i>	32
6.4.1. <i>Introduction</i>	32
6.4.2. <i>Register Description.....</i>	32
6.5. <i>Interrupt Control Unit.....</i>	36
6.5.1. <i>Introduction</i>	36
6.5.2. <i>Function</i>	36
6.5.3. <i>Register Description.....</i>	37
7. Analog Control Unit.....	45

7.1.	<i>Introduction</i>	45
7.2.	<i>Features</i>	45
7.3.	<i>Function</i>	45
7.4.	<i>Register Description</i>	46
8.	DMA Control Unit.....	50
8.1.	<i>Introduction</i>	50
8.2.	<i>Features</i>	50
8.3.	<i>Block Diagram</i>	50
8.4.	<i>Function Description</i>	51
8.4.1.	DMA Transaction.....	51
8.4.2.	DMA Request Arbiter.....	51
8.4.3.	DMA Channels	51
8.4.4.	Programmable data width and data alignment	52
8.4.5.	Error management.....	53
8.4.6.	Interrupts.....	53
8.4.7.	Channel selection.....	54
8.5.	<i>Register Description</i>	55
9.	CCP0/1 Control.....	60
9.1.	<i>Introduction</i>	60
9.2.	<i>Feature</i>	60
9.3.	<i>Block Diagrams</i>	60
9.4.	<i>Function</i>	61
9.5.	<i>Register Description</i>	63
10.	Embedded Memory FLASH Controller	76
10.1.	<i>Introduction</i>	76
10.2.	<i>Feature</i>	76
10.3.	<i>Block Diagram</i>	76
10.4.	<i>Function</i>	76
10.4.1.	Device	76
10.4.2.	Memory Organization	76
10.4.3.	E-FLASH Erase and Program.....	77
10.5.	<i>Register Description</i>	80
11.	MAC Controller.....	84
11.1.	<i>Introduction</i>	84
11.2.	<i>Features</i>	84
11.3.	<i>Block Diagram</i>	84
11.4.	<i>Register Description</i>	84
12.	Timer	90
12.1.	<i>Introduction</i>	90
12.2.	<i>Feature</i>	90

12.3.	<i>Block Diagram</i>	90
12.4.	<i>Function</i>	90
12.5.	<i>Register Description</i>	90
13.	General-purpose I/O (GPIOs)	96
13.1.	<i>Introduction</i>	96
13.2.	<i>Feature</i>	96
13.3.	<i>Function</i>	96
13.4.	<i>Register Function</i>	97
14.	Line-IN ADC (SAR ADC) Controller	126
14.1.	<i>Introduction</i>	126
14.2.	<i>Features</i>	126
14.3.	<i>Block Diagram</i>	126
14.4.	<i>Function</i>	126
14.4.1.	SAR ADC on-off control	126
14.4.2.	Channel Selection	127
14.4.3.	Sampling path	127
14.4.4.	Regulation conversion mode	128
14.4.5.	Injected conversion mode.....	129
14.4.6.	Combined regular/injected simultaneous mode	130
14.4.7.	Regular conversion with DMA	131
14.4.8.	Data alignment.....	132
14.4.9.	Analog watchdog	132
14.4.10.	SAR ADC Program Example.....	133
14.5.	<i>Register Description</i>	133
15.	Audio Delta-Sigma ADC	147
15.1.	<i>Introduction</i>	147
15.2.	<i>Feature</i>	148
15.3.	<i>Block Diagram</i>	148
15.4.	<i>Register Description</i>	148
16.	DAC Up-sampling / Audio PWM control	157
16.1.	<i>Introduction</i>	157
16.2.	<i>Feature</i>	157
16.3.	<i>Block Diagram</i>	157
16.4.	<i>Function</i>	157
16.5.	<i>Register Description</i>	158
17.	Serial peripheral interface (SPI)	164
17.1.	<i>Introduction</i>	164
17.2.	<i>Features</i>	164
17.3.	<i>Block Diagram</i>	165
17.4.	<i>Function</i>	166

17.4.1.	SPI clock phase and clock polarity	166
17.4.2.	Master mode	166
17.4.3.	Slave mode.....	168
17.5.	<i>Register Description</i>	171
18.	SPI controller for FLASH device access (SPIFC)	175
18.1.	<i>Introduction</i>	175
18.2.	<i>Features</i>	175
18.3.	<i>Block Diagram</i>	175
18.4.	<i>Function</i>	175
18.5.	<i>Register Description</i>	180
19.	Universal asynchronous receiver transmitter (UART)	189
19.1.	<i>Introduction</i>	189
19.2.	<i>Features</i>	189
19.3.	<i>Block Diagram</i>	189
19.4.	<i>Function</i>	190
19.4.1.	Frame Format	190
19.4.2.	Transmitter.....	190
19.4.3.	Signe byte transmission	191
19.4.4.	Transmit with DMA function.....	191
19.4.5.	Receiver	192
19.4.6.	Baud Rate Generation.....	192
19.4.7.	Start Bit Detection	192
19.4.8.	Signe byte receive	193
19.4.9.	Receive with DMA function	193
19.4.10.	Noise error reduction	194
19.4.11.	Break frame	194
19.4.12.	Idle frame	194
19.4.13.	Overrun error.....	194
19.4.14.	Parity check.....	194
19.5.	<i>Register Description</i>	194
20.	Inter integrated circuit interface (I2C)	199
20.1.	<i>Introduction</i>	199
20.2.	<i>Features</i>	199
20.3.	<i>Block Diagram</i>	199
20.4.	<i>Function</i>	200
20.4.1.	Master mode	200
20.4.2.	Slave mode.....	204
20.5.	<i>Register Description</i>	207
21.	I2S	212
21.1.	<i>Introduction</i>	212
21.2.	<i>Features</i>	212
21.3.	<i>Block Diagram</i>	212

21.4.	<i>Function</i>	212
21.4.1.	I2S Bus.....	212
21.4.2.	Normal mode & I2S mode	213
21.4.3.	I2S RX Interrupt.....	214
21.4.4.	I2S RX FIFO	214
21.5.	<i>Register Description</i>	214
22.	PWM I/O.....	219
22.1.	<i>Introduction</i>	219
22.2.	<i>Features</i>	219
22.3.	<i>Function</i>	219
22.4.	<i>Register Description</i>	219
23.	Quadrature Decoder.....	227
23.1.	<i>Introduction</i>	227
23.2.	<i>Features</i>	227
23.3.	<i>Function</i>	227
23.4.	<i>Register Description</i>	228
24.	Capacitive Touch Sensor (CTS).....	234
24.1.	<i>Introduction</i>	234
24.2.	<i>Feature</i>	234
24.3.	<i>Notes</i>	234
24.4.	<i>Block Diagram</i>	235
24.5.	<i>Function</i>	235
24.6.	<i>Register Description</i>	235
25.	TIMEBASE Control	242
25.1.	<i>Introduction</i>	242
25.2.	<i>Feature</i>	242
25.3.	<i>Note</i>	242
25.4.	<i>Block Diagram</i>	243
25.5.	<i>Function</i>	243
25.6.	<i>Register Description</i>	243
26.	Watchdog Timer.....	246
26.1.	<i>Introduction</i>	246
26.2.	<i>Features</i>	246
26.3.	<i>Block Diagram</i>	246
26.4.	<i>Function</i>	246
26.5.	<i>Register Description</i>	246

**REVISION HISTORY**

Revision	Date	By	Remark
V1.5	2022/11/30	Kim Huang	<ul style="list-style-type: none">1. Corrected the description of CCP0/1 in System Interrupt Vector Map2. Revise the description of SAR-ADC ON/OFF control content3. Revise the SPI feature description4. Fixed the QD Pins description
V1.4	2021/09/01	Kim Huang	<ul style="list-style-type: none">1. Add the Introduction of DMA's dual mode.
V1.3	2021/02/18	Kim Huang	<ul style="list-style-type: none">1. Revise the page size of the internal flash(0x5001_000C bit[31:9])2. Add the notices for using CTS (Ch.24)
V1.2	2020/10/22	Kim Huang	<ul style="list-style-type: none">1. Add the SPI0 & SPI1 CS H/W Control function description (Ch.13)
V1.1	2020/08/28	Kim Huang	<ul style="list-style-type: none">1. Naming consistency for registers and GPCM1Fx.h header2. Naming consistency for IRQ Handler and startup_GPCM1Fx.s3. Making correction and providing more detail description for registers.
V1.0	2020/01/22	Kim Huang	<ul style="list-style-type: none">1. Update general description (Ch.1)2. Update GPCM1F Block Diagram (Ch2.1)3. Add IO special functions table (Ch2.2)4. Update the clock control unit (Ch.6)5. Add IO mapping table of the Line-IN ADC(Ch.14)6. Update SPIFC pin description (Ch.18)
V0.9	2019/09/25	Kim Huang	Preliminary version

1. General Description

The GPCM1F series of microcontrollers based on the ARM® Cortex®-M0 processor core and operating at a frequency of up to 81.92MHz. The GPCM1F 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 (81.92MHz 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

- 13KB SRAM and 2KB Cache
- Up to 64KB Program Memory

■ Clock Management

- Internal oscillator: 8.192MHz
- Phase Lock Loop with configurable output frequency: 81.92 M Hz (Max)
- Internal 32KHz oscillator and external 32KHz crystal supported

■ Power Management

- Sleep mode: Only the CPU stopped, and system clock ON.
- Deep Sleep mode: All clocks are stopped.
- Halt mode: In deep sleep mode; but 32KHz RTC Clock ON
- 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
- 12-bit DAC with two 16-bit software channel with noise filter mixer and scalar to playback high quality sound
- 16-bit Sigma Delta Codec ADC
- 12-bit SAR ADC and 8 line PADs
- CTS(Capacitor sensing touch) support 10 Touch IO PADs

■ Timer

- Three general-purpose 16-bit timers/ counters
- Two touch sensing timers
- Two 16-bit CCP timers (All of them can be used as general-purpose timers).

■ System Control

- Three-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 31 multifunction bi-directional I/Os
 - Each incorporate with pull-up resistor, pull-down resistor, output high, output low or floating input, depending on programmer's settings on the corresponding registers
- I/O ports with 4mA/ 8mA/ 12mA/ 16mA source/ sink current

■ Industrial Control Peripherals

- Two 16-bit CCP (Capture/ Compare/ PWM) Unit
- 8 PWMIOs

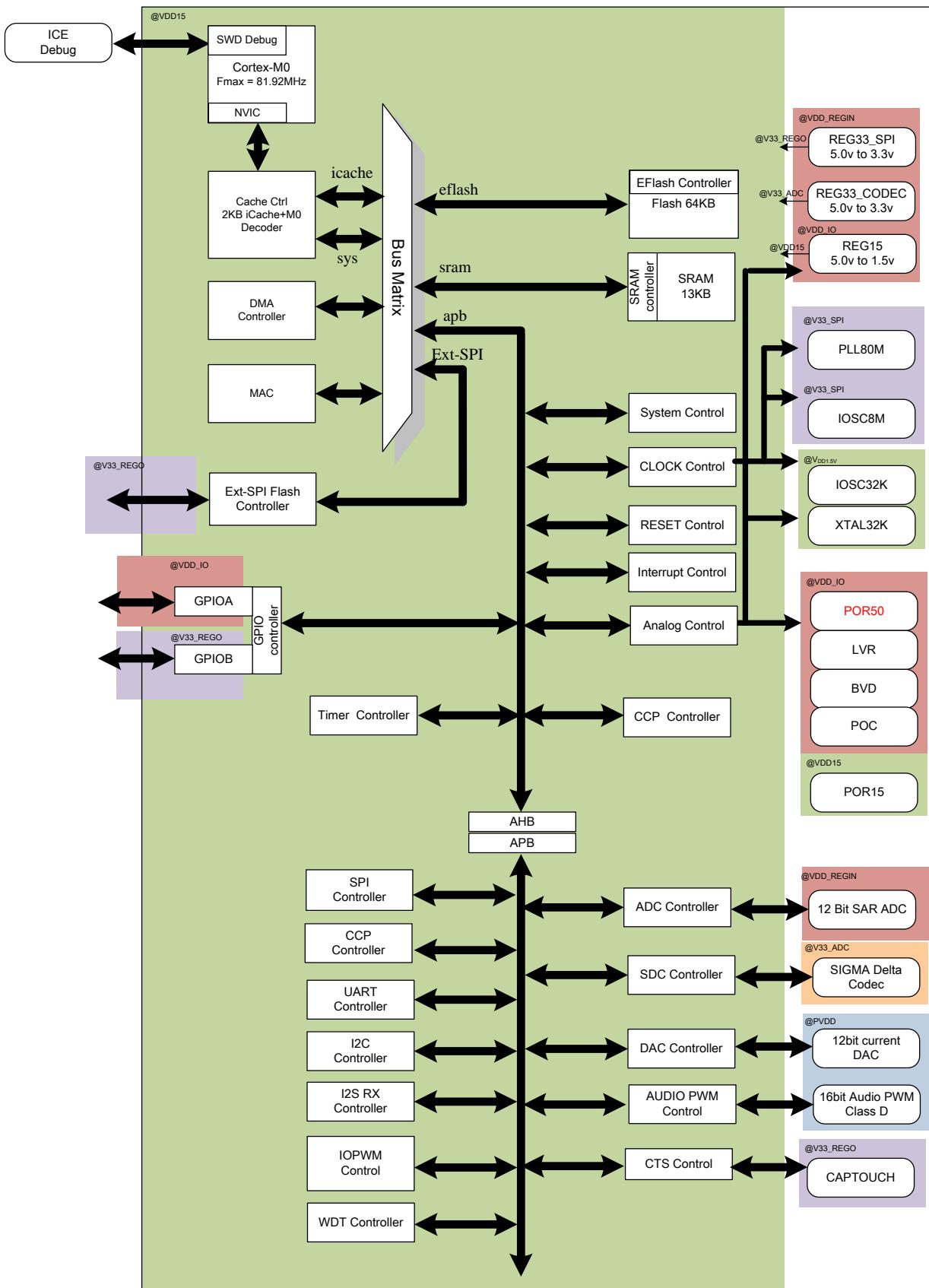
■ Communication peripherals

- 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
- One UART hardware
- One IR TX hardware

■ Debug System

- ARM serial wire debug (SWD)
- Supporting up to 3 hardware breakpoints

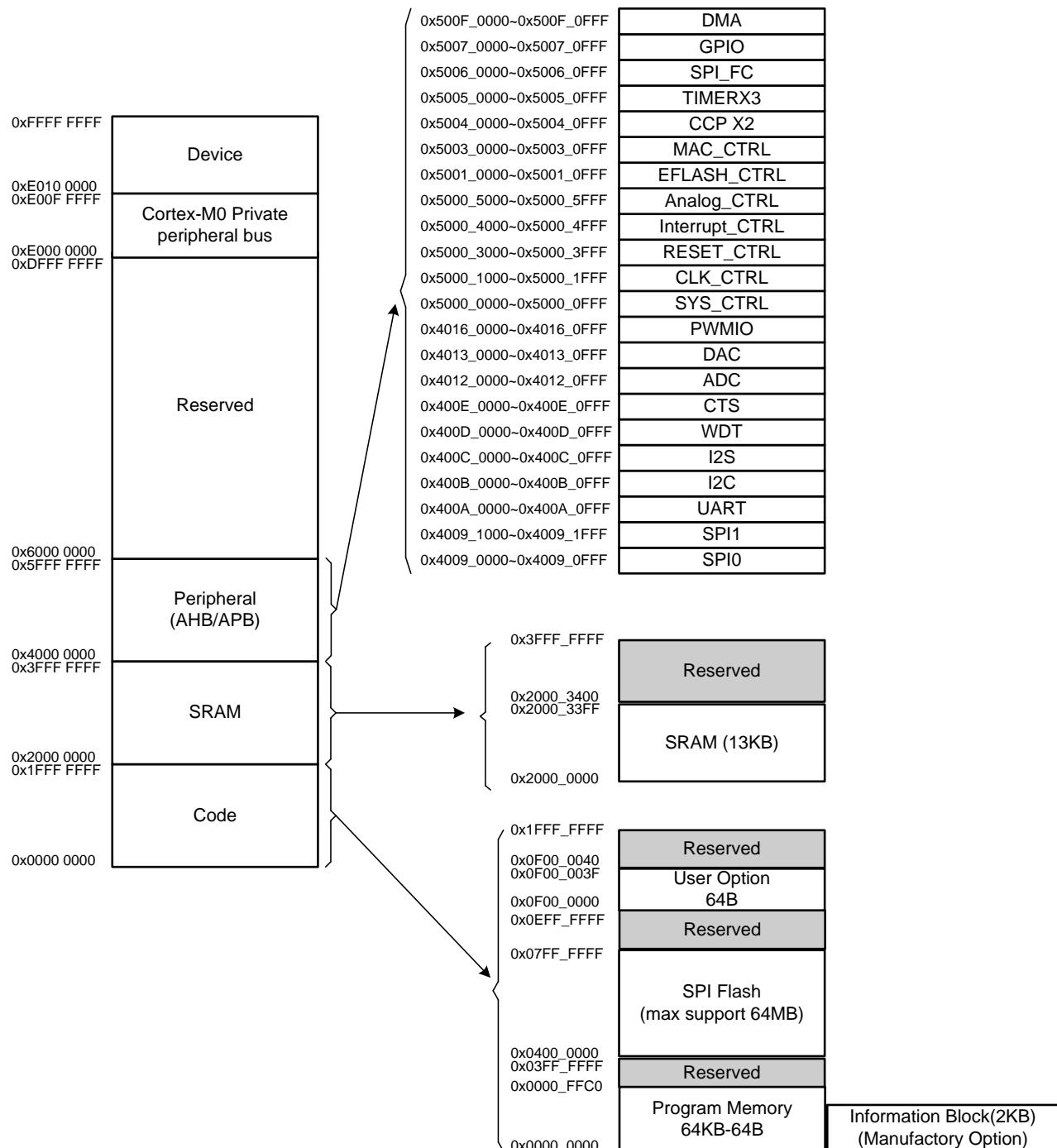
2.1. GPCM1F Block Diagram



2.2. GPCM1F IO Special Functions

Name	SPIFC	SPI0	SPI1	IR_Tx	CCP	IO PWM	Line-in	UART	I2C	I2S	Quadrature Decoder	CTS	Key Change Wakeup	X32K	ICE I/F
IOA[0]			SCLK_1		CCP0_1			Tx_3	SCK_0			v	v		
IOA[1]			SI_1		CCP0_1				SDA_0			v	v		
IOA[2]			CS_1		CCP0_1							v	v		
IOA[3]			SO_1		CCP0_1							v	v		
IOA[4]				TX_1					Tx_0			v	v		
IOA[5]						v			Rx_0			v	v		
IOA[6]						v						v	v		
IOA[7]						v						v	v		
IOA[8]						v						v	v		
IOA[9]		SCLK_1				v			Rx_3			v	v		
IOA[10]		SI_1				v							v		
IOA[11]		CS_1				v							v		
IOA[12]		SO_1				v							v		
IOA[13]					CCP0_0			Tx_1	SCK_2				v	X32KI	
IOA[14]					CCP0_0				Rx_1				v	X32KO	
IOA[15]					CCP0_0					SCK_1			v		
IOA[16]					CCP0_0					SDA_1			v		
IOA[17]				TX_0			v			SDA_2			v		
IOA[18]		SCLK_0			CCP1		v			MCLK	QD0_A		v		
IOA[19]		SI_0			CCP1		v			BCLK	QD0_B		v		
IOA[20]		CS_0			CCP1		v			LR	QD1_A		v		
IOA[21]		SO_0			CCP1		v			Data	QD1_B		v		
IOA[22]							v	Tx_2					v		
IOA[23]							v	Rx_2					v		
IOA[24]							v						v		SDA
IOB[0]	SIO3												v		
IOB[1]	SCLK	SCLK_0											v		
IOB[2]	SIO0	SI_0											v		
IOB[3]	CS	CS_0											v		
IOB[4]	SIO1	SO_0											v		
IOB[5]	SIO2												v		

3. Memory Organization

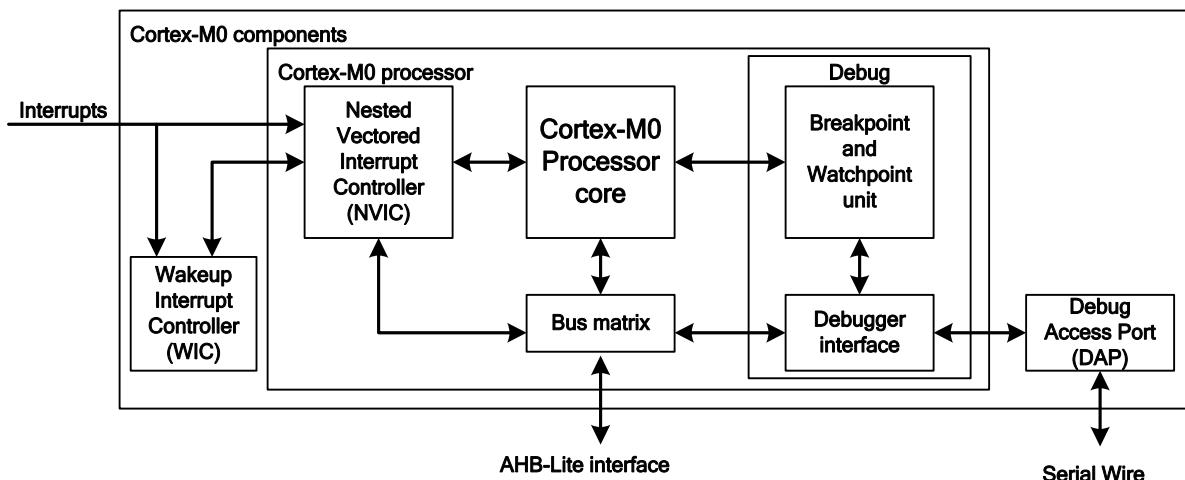


4. ARM® Cortex®-M0 Core

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, easy-to-use programmers model
- highly efficient ultra-low power operation
- excellent code density
- deterministic, high-performance interrupt handling
- upward compatibility with the rest of the Cortex-M processor family



The Cortex-M0 processor is built on a highly area and power optimized 32-bit processor core, 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, providing high-end processing hardware including 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, removing any code overhead from the ISRs. Tail-chaining optimization also significantly reduces the overhead when switching from one ISR to another. To optimize low-power designs, the NVIC integrates with the sleep modes that include a deep sleep function that enables the entire device to be rapidly powered down.

Note: When GPCM1F 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 power it 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

Base Address : 0xE000_E000				
Name	Description	Address	Type	Reset value
CPU_ID	CPUID Base Register. The CPUID register contains the processor part number, version, and implementation information.	0xE0000_ED00	R	0x410C_C200
ICSR	Interrupt Control and State Register. The ICSR provides: – set-pending and clear-pending bits for the PendSV and SysTick exceptions – the exception number of the exception being processed – whether there are preempted active exceptions – the exception number of the highest priority pending exception – Whether any interrupts are pending.	0xE0000_ED04	R/W	0x0000_0000
AIRCR	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, you must write 0x5FA to the VECTKEY field, otherwise the processor ignores the write.	0xE0000_ED0C	R/W	0xFA05_0000
SCR	System Control Register. The SCR controls features of entry to and exit from low power state.	0xE0000_ED10	R	0x0000_0000
CCR	Configuration and Control Register. The CCR is a read-only register and it indicates some aspects of the behavior of the Cortex-M0 processor	0xE0000_ED14	R/W	0x0000_0208
SHPR2	System Handler Priority Register 2. The SHPR2 register sets the priority level for the SVCAll handler.	0xE0000_ED1C	R/W	0x0000_0000
SHPR3	System Handler Priority Register 3. The SHPR3 register sets the priority level for the SysTick and PendSV handlers.	0xE0000_ED20	R/W	0x0000_0000
SHCSR	System Handler Control and State Register. The SHCSR register controls and provides the status of system handlers	0xE0000_ED24	R/W	0x0000_0000

Registers

Bit	Name	Description	Access	Reset value
[31:24]	IMPLEMENTER	Implementer code assigned by ARM. ARM == 0x41	R	0x41
[23:20]	VARIANT	Reads as 0x0	R	0x0
[19:16]	ARCHITECTURE	Reads as 0xC for ARMv6-M parts	R	0xC
[15:4]	PARTNO	Reads as 0xC20.	R	0xC20

Bit	Name	Description	Access	Reset value
[3:0]	REVISION	Reads as 0x0	R	0x0

ICSR Interrupt Control and State Register								Address : 0xE000 ED04
31	30	29	28	27	26	25	24	
NMIPENDSET	--		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	--	
23	22	21	20	19	18	17	16	
ISRPREEMPT	ISR PENDING	--			VECTPENDING			
15	14	13	12	11	10	9	8	
		VECTPENDING		--			VECTACTIVE	
7	6	5	4	3	2	1	0	
			VECTACTIVE					

Bit	Name	Description	Access	Reset value
[31]	NMIPENDSET	<p>NMI Set-Pending Bit.</p> <p>Write: 0 = No effect. 1 = Changes NMI exception state to pending.</p> <p>Read: 0 = NMI exception not pending. 1 = NMI exception pending.</p> <p>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. Entering the handler then clears this bit to 0. 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.</p>	R/W	0x0
[30:29]	--	Reserved	R	0x0
[28]	PENDSVSET	<p>PendSV Set-Pending Bit</p> <p>Write: 0 = No effect. 1 = Changes PendSV exception state to pending.</p> <p>Read: 0 = PendSV exception is not pending. 1 = PendSV exception is pending.</p> <p>Note: Writing 1 to this bit is the only way to set the PendSV exception state to pending.</p>	R/W	0x0
[27]	PENDSVCLR	<p>PendSV Clear-Pending Bit</p> <p>Write: 0 = No effect. 1 = Removes the pending state from the PendSV exception.</p> <p>Note: This bit is write-only. To clear the PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVCLR” at the same time.</p>	W	0x0
[26]	PENDSTSET	<p>SysTick Exception Set-Pending Bit</p> <p>Write: 0 = No effect. 1 = Changes SysTick exception state to pending.</p> <p>Read: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.</p>	R/W	0x0
[25]	PENDSTCLR	<p>SysTick Exception Clear-Pending Bit</p> <p>Write: 0 = No effect. 1 = Removes the pending state from the SysTick exception.</p> <p>Note: This bit is write-only. When you want to clear PENDST bit, you must “write 0 to PENDSTSET and write 1 to PENDSTCLR” at the same time.</p>	W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
[24]	--	Reserved	R	0x0
[23]	ISRPREEMPT	Interrupt Preemption Bit If set, a pending exception will be serviced on exit from the debug halt state. This bit is read only.	R	0x0
[22]	ISR PENDING	Interrupt Pending Flag, Excluding NMI And Faults 0 = Interrupt not pending. 1 = Interrupt pending. This bit is read only.	R	0x0
[21]	--	Reserved	R	0x0
[20:12]	VECTPENDING	Exception Number Of The Highest Priority Pending Enabled Exception 0 = No pending exceptions. Non-zero = Exception number of the highest priority pending enabled exception. This bit is read only.	R	0x0
[11:9]	--	Reserved	R	0x0
[8:0]	VECTACTIVE	Contains The Active Exception Number 0 = Thread mode. Non-zero = Exception number of the currently active exception. This bit is read only.	R	0x0

AIRCR Application Interrupt and Reset Control Register Address : 0xE000 ED0C

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

Bit	Name	Description	Access	Reset value
[31:16]	VECTORKEY	Register Access Key Write: When writing to this register, the VECTORKEY field need to be set to 0x05FA, otherwise the write operation would be ignored. The VECTORKEY filed is used to prevent accidental write to this register from resetting the system or clearing of the exception status. Read: Read as 0xFA05.	R	0xFA05
[15:3]	--	Reserved.	R	0x0

Bit	Name	Description	Access	Reset value
[2]	SYSRESETREQ	<p>System Reset Request</p> <p>Writing this bit 1 will cause a reset signal to be asserted to the chip to indicate a reset is requested.</p> <p>The bit is a write only bit and self-clears as part of the reset sequence.</p>	R	0x0
[1]	VECTCLRACTIVE	<p>Exception Active Status Clear Bit</p> <p>Reserved for debug use. When writing to the register, user must write 0 to this bit, otherwise behavior is unpredictable.</p>	R	0x0
[0]	--	Reserved	R	0x0

System Control Register								Address : 0xE000 ED10
31	30	29	28	27	26	25	24	
				--				
23	22	21	20	19	18	17	16	
				--				
15	14	13	12	11	10	9	8	
				--				
7	6	5	4	3	2	1	0	
	--		SEVONPEND	--	SLEEPDEEP	SLEEPONEXIT	--	

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0x0
[4]	SEVONPEND	<p>Send Event On Pending Bit</p> <p>0 = Only enabled interrupts or events can wake-up the processor, disabled interrupts are excluded.</p> <p>1 = Enabled events and all interrupts, including disabled interrupts, can wake-up the processor.</p> <p>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 is registered and affects next WFE.</p> <p>The processor also wakes up on execution of an SEV instruction or an external event.</p>	R/W	0x0
[3]	--	Reserved	R	0x0
[2]	SLEEPDEEP	<p>Processor Deep Sleep And Sleep Mode Selection</p> <p>Controls whether the processor uses sleep or deep sleep as its low power mode:</p> <p>0 = Sleep mode. 1 = Deep Sleep mode.</p>	R/W	0x0
[1]	SLEEPONEXIT	<p>Sleep-On-Exit Enable</p> <p>This bit indicates sleep-on-exit when returning from Handler mode to Thread mode:</p> <p>0 = Do not sleep when returning to Thread mode.</p> <p>1 = Enter Sleep, or Deep Sleep, on return from ISR to Thread mode.</p> <p>Setting this bit to 1 enables an interrupt driven application to avoid</p>	R/W	0x0

Bit	Name	Description	Access	Reset value
	--	returning to an empty main application.		
[0]	--	Reserved	R	0x0

CCR Configuration and Control Register								Address : 0xE000 ED14
31	30	29	28	27	26	25	24	
				--				
23	22	21	20	19	18	17	16	
				--				
15	14	13	12	11	10	9	8	
			--			STKALIGN		
7	6	5	4	3	2	1	0	
		--		UNALIGN_TRP		--		

Bit	Name	Description	Access	Reset value
[31:10]	--	Reserved	R	0x0
[9]	STKALIGN	Stack Alignment This bit always reads as 1, 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.	R	0x1
[8:4]	--	Reserved	R	0x0
[3]	UNALIGN_TRP	Unaligned Access Traps This bit always reads as 1, indicates that all unaligned accesses generate a HardFault.	R	0x1
[2:0]	--	Reserved	R	0x0

SHPR2 System Handler Priority Register 2								Address : 0xE000 ED1C
31	30	29	28	27	26	25	24	
PRI_11				--				
23	22	21	20	19	18	17	16	
				--				
15	14	13	12	11	10	9	8	
			--					
7	6	5	4	3	2	1	0	
		--						

Bit	Name	Description	Access	Reset value
[31:30]	PRI_11	Priority Of System Handler 11 – SVCall “0” denotes the highest priority and “3” denotes the lowest priority.	R/W	0x0
[29:0]	--	Reserved	R	0x0

System Handler Priority Register 3								Address : 0xE000 ED20
31	30	29	28	27	26	25	24	
PRI_15				--				
23	22	21	20	19	18	17	16	
PRI_14				--				
15	14	13	12	11	10	9	8	
			--					
7	6	5	4	3	2	1	0	
			--					

Bit	Name	Description	Access	Reset value
[31:30]	PRI_15	Priority Of System Handler 15 – SysTick "0" denotes the highest priority and "3" denotes the lowest priority.	R/W	0x0
[29:24]	--	Reserved	R	0x0
[23:22]	PRI_14	Priority Of System Handler 14 – PendSV "0" denotes the highest priority and "3" denotes the lowest priority.	R/W	0x0
[21:0]	--	Reserved	R	0x0

System Handler Control and State Register								Address : 0xE000 ED24
31	30	29	28	27	26	25	24	
			--					
23	22	21	20	19	18	17	16	
			--					
15	14	13	12	11	10	9	8	
SVCALLPEND ED			--					
7	6	5	4	3	2	1	0	
			--					

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15]	SVCALLPENDED	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 pending1).	R/W	0x0
[14:0]	--	Reserved	R	0x0

4.4. System timer, SysTick

The processor has a 24-bit system timer, SysTick, that counts down from the reloaded value to zero, and reloads the value in the SYST_RVR register on the next clock cycle, and then counts down on 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



GPCM1F SERIES PROGRAMMING GUIDE

wakeup source. Ensure software uses aligned word accesses to access the SysTick registers. If the SysTick counter reload and current value are undefined at reset, the correct initialization sequence for the SysTick counter is:

- ## 1. Program reload value

- ## 2. Clear current value

Program Control and Status register

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

4.5. Register

Register Map

Base Address : 0xE000_E000				
Name	Description	Offset Address	Type	Reset value
SYST_CSR	SysTick Control and Status Register	0xE000_E010	R/W	0x0000 0000
SYST_RVR	SysTick Reload value Register	0xE000_E014	R/W	--
SYST_CVR	SysTick Current value Register	0xE000_E018	R/W	--

Registers

SYST_CSR SysTick Control and Status Register

Address : 0xE000 E010

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

Bit	Name	Description	Access	Reset value
[31:17]	--	Reserved	--	-
[16]	COUNTFLAG	Returns 1 if timer counted to 0 since this register was read last time. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.	R	--
[15:3]	--	Reserved	--	-
[2]	CLKSOURCE	0 = Clock source is optional (refer to STCLK_S). 1 = Core clock used for SysTick if no external clock provided; this bit will read as 1 and ignore writes.	R/W	0
[1]	TICKINT	0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 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.	R/W	0
[0]	ENABLE	0 = Counter Disabled 1 = Counter will operate in a multi-shot manner.	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

SysTick Reload Value Register								Address : 0xE000 E014	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
RELOAD									
15	14	13	12	11	10	9	8		
RELOAD									
7	6	5	4	3	2	1	0		
RELOAD									

Bit	Name	Description	Access	Reset value
[31:24]	--	Reserved	--	-
[23:0]	RELOAD	Value to load into the Current Value register when the counter reaches 0.	R/W	--

SysTick Current Value Register								Address : 0xE000 E018	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
CURRENT									
15	14	13	12	11	10	9	8		
CURRENT									
7	6	5	4	3	2	1	0		
CURRENT									

Bit	Name	Description	Access	Reset value
[31:24]	--	Reserved	--	-
[23:0]	CURRENT	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 to 0. Unsupported bits RAZ (see SysTick Reload Value register).	R/W	--

5. Nested Vectored Interrupt Controller (NVIC)

5.1. Overview

GPCM1F 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
- Supports for interrupt tail-chaining and late-arrival
- Supports an external Non-maskable Interrupt (NMI)
- Software interrupt generation

5.3. Exception Types

The exception types are described as below table:

Exception Types	Descriptions
Reset	Reset is invoked on power up or a 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	Reserved	NC	0x0000 0040
17	1	Reserved	NC	0x0000 0044
18	2	DMA_IRQHandler	DMA 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	CCP0 Interrupt	0x0000 0064
26	10	CCP1_IRQHandler	CCP1 Interrupt	0x0000 0068
27	11	CTS_TM0_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	Reserved	NC	0x0000 007C
32	16	UART_IRQHandler	UART Interrupt	0x0000 0080
33	17	Reserved	NC	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	EXTI0_IRQHandler	External Input 0 Interrupt	0x0000 0094
38	22	EXTI1_IRQHandler	External Input 1 Interrupt	0x0000 0098
39	23	EXTI2_IRQHandler	External Input 2 Interrupt	0x0000 009C
40	24	EXTI3_IRQHandler	External Input 3 Interrupt	0x0000 00A0
41	25	Reserved	NC	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	Reserved	NC	0x0000 00B8
47	31	Reserved	NC	0x0000 00BC

6. SYSTEM Descriptions

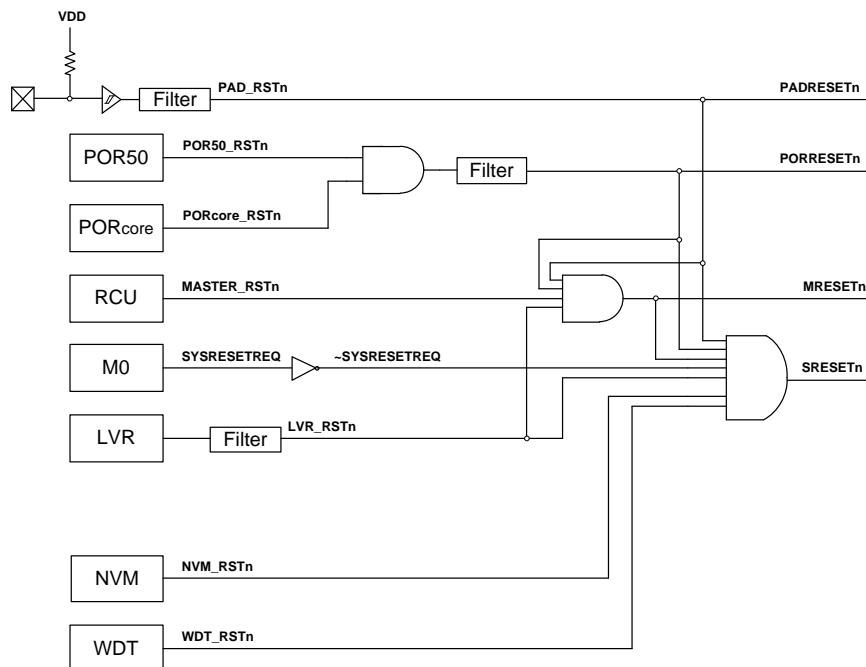
6.1. Reset System

6.1.1. Introduction

The GPCM1F 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 from IO PAD Resetn and active at Low State. In GPCM1F internal have built Filter to avoid IO denounce. In general speak, to make PADRESETn happened, user needs to let IO PAD Resetn keep zero status about 50 us.

PORRESETn

PADRESETn is triggered From GPCM1F power on. When supplying IO VDD5V power, it will trigger PORRESETn

Master Reset

A complete reset to the whole chip is executed by a master reset. Master reset 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 additional, a MASTER_RSTn can be triggered by setting bit **RCU_CTRL.MRSTn** to reset the whole chip..

System Reset

A system reset affects almost all logics, but exceptions are RCU registers and Debug system if debug link is present.

The debug system is reset by System Reset in normal operation mode when debug link is not present. As debug link present,

System Reset would not affect the Debug system.

6.1.4. Register Description

Register Map

Base Address : 0x5000_3000							
Name	Description				Address	Type	Reset value
RCU_CTRL	Reset Control Register. Enabling of reset triggered by critical events				0x5000_3000	R/W	0x0000_0001
RCU_STS	Reset Status Register. Reset source triggered flag				0x5000_3004	R/W	0x0000_0001

Register Function

RCU_CTRL Reset Control Register								Address : 0x5000_3000
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
--								MRSTn

Bit	Name	Description						Access	Reset value
[31:1]	--	Reserved						R	0
[0]	MRST_EN_TRIGGER (MRSTn)	Master Reset Trigger (Write Protected) 0 = Trigger the Master Reset 1 = No effect Note: This bit is written-protected, refer to SMU.SysUnLock to unlock. It will auto recover to 1 after programming 0 to MRSTn.						R/W	1

RCU_STS Reset Status Register								Address : 0x5000_3004
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
--	WDGRF	--	LVRRF	SYSRF	MRF	PADRF	PORRF	

Bit	Name	Description						Access	Reset value
[31:7]	--	Reserved						--	0
[6]	SYS_WDT_FLAG (WDGRF)	Watchdog Reset Flag 0 = No reset from watchdog reset. 1 = Watchdog reset occurs. Note : Write 1 to clear this bit						R/W	0
[5]	--	Reserved						--	0
[4]	LVR_RST_FLAG (LVRRF)	Low Voltage Reset Flag 0 = No reset from LVR reset. 1 = LVR reset occurs. Note : Write 1 to clear this bit						R/W	0
[3]	SYS_RST_FLAG (SYSRF)	System Reset Flag 0 = No reset from SYSRESETREQ reset. 1 = SYSRESETREQ						R/W	0

Bit	Name	Description	Access	Reset value
		reset occurs. Note.1 : Write 1 to clear this bit Note.2 : 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 further to trigger a system reset.		
[2]	SOFT_RST_FLAG (MRF)	Software Triggered Master Reset Flag 0 = No reset from software triggered master reset. 1 = Software triggered reset occurs. Note: Write 1 to clear this bit.	R/W	0
[1]	KEY_RST_FLAG (PADRF)	External PAD Reset Flag 0 = No reset from external reset pad. 1 = External PAD reset occurs. Note: Write 1 to clear this bit.	R/W	0
[0]	POR_FLAG (PORRF)	POR Reset Flag 0 = No reset from Power-On Reset. 1 = Power-On Reset occurs. Note: Write 1 to clear this bit.	R/W	1

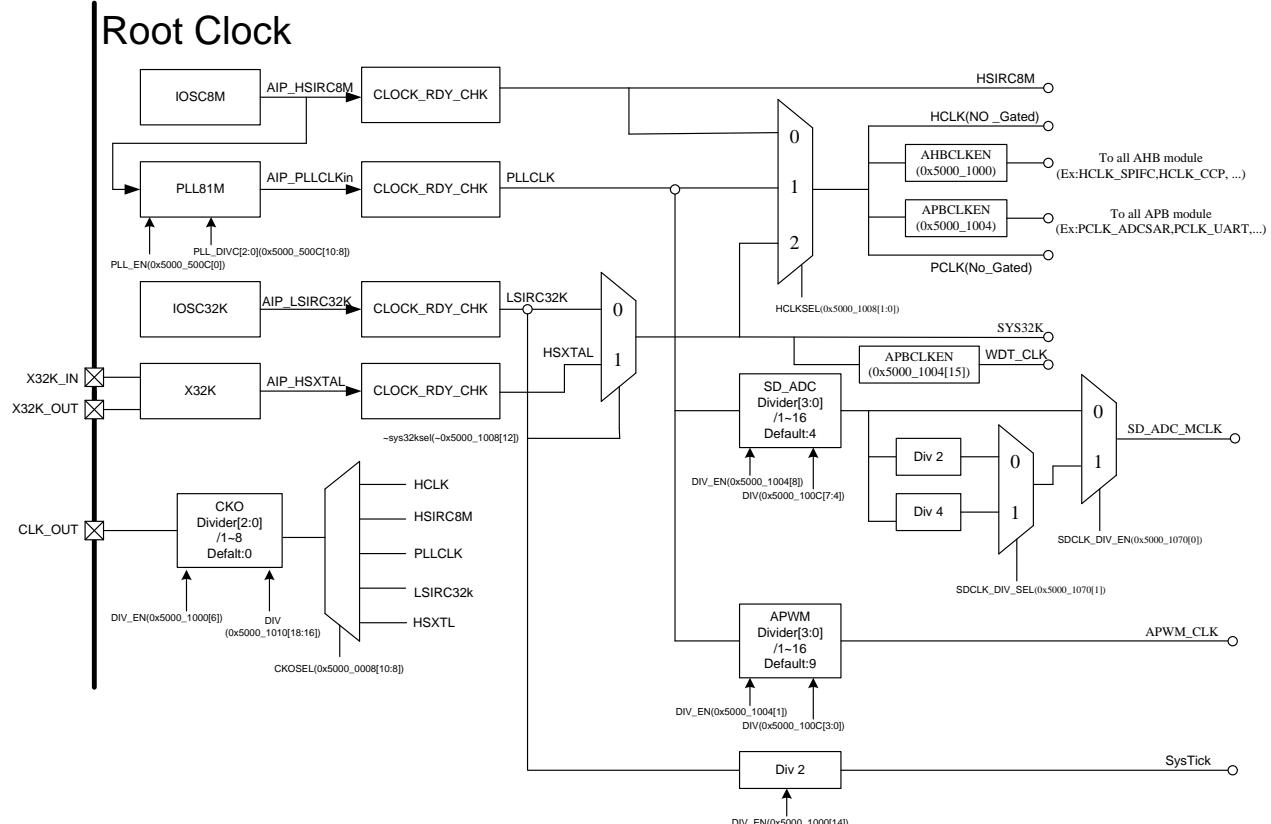
6.2. Clock Control Unit

6.2.1. Introduction

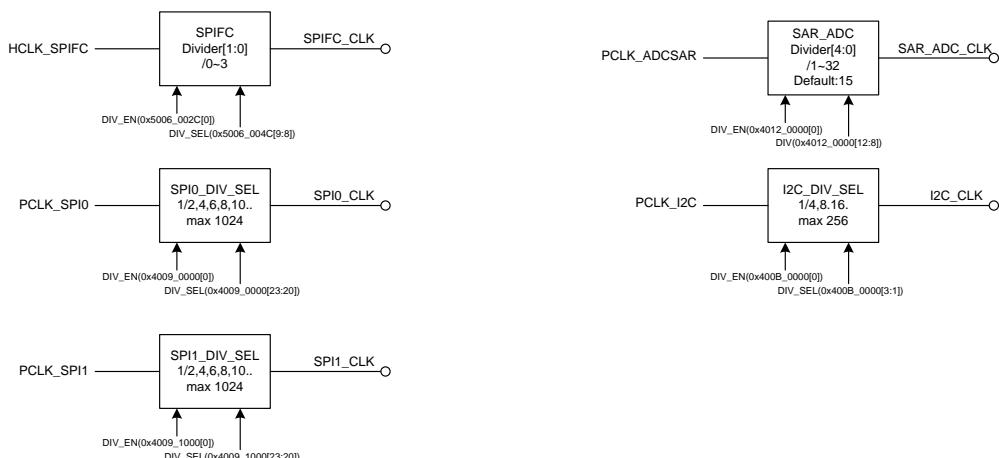
The clock control unit has four clock sources:

- Low Speed Internal RC Oscillator 32 KHz
- High Speed Internal RC Oscillator 8.192 MHz
- External Low Speed Crystal Oscillator 32KHz
- PLL up to 81.92 MHz

6.2.2. Block Diagram



Leaf Clock



6.2.3. Register Description

Register Map

Base Address : 0x5000_1000					
Name	Description		Address	Type	Reset value
CLOCK_AHBCKEN	AHB(Advanced High-performance Bus) Peripherals Clock Enable Register		0x5000_1000	R/W	0x0000_FF80
CLOCK_APBCKEN	APB(Advanced Peripheral Bus) Peripherals Clock Enable Register		0x5000_1010	R/W	0x0000_F000
CLOCK_AHBCKSEL	AHB Peripherals Clock Selection Register		0x5000_1020	R/W	0x0000_0000
CLOCK_CLKDIV1	Analog Macro Clock Divider Register		0x5000_1030	R/W	0x0000_0049
CLOCK_CLKSTS	Clock Status Register		0x5000_1050	R	0x0000_0000
CLOCK_CLKDIV3	Clock Sigma-Delta Clock Divider Control		0x5000_1070	R/W	0x0000_1F1F

Register Function

CLOCK_AHBCKEN AHB Peripherals Clock Enable Register								Address : 0x5000_1000
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
CPUCKEN	SYSTICKCKEN	ANACKEN	INTCKEN	SRAMCKEN	APB_SYSCKEN	AHB_SYSCKEN	EFLASHCKEN	
7	6	5	4	3	2	1	0	
GPIOCKEN	--	SPIFCCKEN	MACCKEN	CCP1CKEN	CCP0CKEN	DMACKEN		

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15]	CPU_CLK_ENABLE (CPUCKEN)	CPU Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[14]	SYSTICK_CLK_ENABLE (SYSTICKCKEN)	SysTick External Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[13]	ANALOG_CLK_ENABLE (ANACKEN)	Analog Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[12]	INT_CLK_ENABLE (INTCKEN)	Interrupt Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[11]	RAM_CLK_ENABLE (SRAMCKEN)	SRAM Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[10]	APB_CLK_ENABLE (APB_SYSCKEN)	APB BUS System Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[9]	AHBBUS_CLK_ENABLE (AHB_SYSCKEN)	AHB BUS System Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[8]	FLASH_CLK_ENABLE (EFLAHCKEN)	EFLASH Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[7]	GPIO_CLK_ENABLE (GPIOCKEN)	GPIO Unit Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1

Bit	Name	Description	Access	Reset value
[6:5]	--	Reserved	R	0
[4]	SPIFC_CLK_ENABLE (SPIFCKEN)	SPIFC Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[3]	MAC_CLK_ENABLE (MACCKEN)	MAC Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[2]	CCP1_CLK_ENABLE (CCP1CKEN)	Capture and Compare Unit1 Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[1]	CCP0_CLK_ENABLE (CCP0CKEN)	Capture and Compare Unit0 Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[0]	DMA_CLK_ENABLE (DMACKEN)	DMA Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0

CLOCK_APBCKEN APB Peripherals Clock Enable Register
Address : 0x5000_1010

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
WDTCKEN	--			APWMCKEN	CTS_TBCKEN	I2SCKEN	CODECADCCKEN (MIC)
7	6	5	4	3	2	1	0
SARADCCKEN	SPI1CKEN	SPI0CKEN	UARTCKEN	I2CCKEN	TMRCKEN	DACCKEN	PWMIOCKEN

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15]	WDG_CLK_ENABLE (WDTCKEN)	Watchdog Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	1
[14:12]	--	Reserved	R	0x7
[11]	AUDPWM_CLK_ENABLE (APWMCKEN)	Audio PWM Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[10]	CTS_TB_CLK_ENABLE (CTS_TBCKEN)	CTS and TimeBase Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[9]	I2S_CLK_ENABLE (I2SCKEN)	I2S Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[8]	DSADC_CLK_ENABLE (CODECADCCKEN)	Delta sigma ADC clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[7]	SARADC_CLK_ENABLE (SARADCCKEN)	SAR ADC Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[6]	SPI1_CLK_ENABLE (SPI1CKEN)	SPI1 Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[5]	SPI0_CLK_ENABLE (SPI0CKEN)	SPI0 Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[4]	UART_CLK_ENABLE (UARTCKEN)	UART Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[3]	I2C_CLK_ENABLE (I2CCKEN)	I2C Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[2]	TIMER_CLK_ENABLE	Timer Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
	(TMRCKEN)			
[1]	DAC_CLK_ENABLE (DACCKEN)	DAC Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[0]	PWMIO_CLK_ENABLE QD_CLK_ENABLE (PWMIOCKEN)	Dual functions: PWMIO Control Clock Enabling bit. 0 = Disabled, 1 = Enabled QD Control Clock Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0

CLOCK_AHBCKSEL AHB Peripherals Clock Selection 0 Register

Address : 0x5000_1020

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--				SYS32KSEL			
7	6	5	4	3	2	1	0
--				HCLKSEL			

Bit	Name	Description	Access	Reset value
[31:13]	--	Reserved	R	0
[12]	SYS32K_SEL_IOSC 32K (SYS32KSEL)	SYS32K Selection.1: IOSC32K, 0: XTAL32K	R/W	1
[11:2]	--	Reserved	R	0
[1:0]	HCLK_SEL (HCLKSEL)	HCLK Source Selection 00 = HCLK_SEL_IOSC8M, IOSC 8M 01 = HCLK_SEL_PLL, PLL Clock 10 = HCLK_SEL_SYS32K, System 32KHz 11 = Reserved	R/W	0

CLOCK_CLKDIV1 Analog macro Clock Divider Register

Address : 0x5000_1030

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

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7:4]	DSADC_CLK_DIV (SD_ADC_CLKDIV)	Codec ADC Output Clock Divider Output clock = (MUX Output Clock) / (SD_ADC_CLKDIV + 1)	R/W	'h4
[3:0]	AUDCLK_SEL_SYSCL K_DIV (APWM_CLKDIV)	Audio PWM Output Clock Divider Clock output clock = (MUX Output Clock) / (APWM_CLKDIV + 1)	R/W	'h9

CLOCK_CLKSTS Clock Status Register								Address : 0x5000_1050	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--								HSXTLRDY	PLLRDY
--								LSIRCRDY	HSIRCRDY

Bit	Name	Description	Access	Reset value
[31:4]	--	Reserved	R	0
[3]	XTAL32K_RDY (HSXTLRDY)	External 32768Hz Crystal Oscillator Ready Flag 0 = HSXTL is not ready. 1 = HSXTL is ready.	R	0
[2]	PLL_RDY (PLLRDY)	PLL Ready Flag 0 = PLL is not ready. 1 = PLL is ready.	R	0
[1]	IOSC32K_RDY (LSIRCRDY)	Low Speed Internal RC Ready Flag 0 = LSIRC is not ready. 1 = LSIRC is ready.	R	0
[0]	IOSC8M_RDY (HSIRCRDY)	High Speed Internal RC Ready Flag 0 = HSIRC is not ready. 1 = HSIRC is ready.	R	0

CLOCK_CLKDIV3 Codec Clock Divider Control Register								Address : 0x5000_1070	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0	SD_CLK_DIV_SEL	

Bit	Name	Description	Access	Reset value
[31:2]	--	Reserved	R	0
[1:0]	DSADC_CLK_DIV (SD_CLK_DIV_SEL)	Codec ADC Clock out Divider Selection 00: Codec ADC Clock DIV1 01: Codec ADC Clock DIV2 10: Reserved 11: Codec ADC Clock DIV4	R/W	'h11

6.3. Low Power Mode

6.3.1. Introduction

By default, the GPCM1F 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 when waiting for an external event.

The GPCM1F Serials has two sleep modes to reduce power consumption:

- Sleep (Halt) mode (only 32K)
- Deep Sleep mode (All Clock Off)

6.3.2. Function

Entering Sleep Mode

The Sleep mode is entered by executing the Wait For Interrupt (WFI) or Wait for Event (WFE) instructions.

Note: Before system enters sleep mode, do not issue a Cache OFF command because it may cause system unable to enter sleep mode.

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

Wait for interrupt

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

Wait for event

The wait for event instruction, WFE, causes entry to sleep mode depending on the value of a one-bit event register.

Note: When using Event Wakeup, WFE() must be issued twice before entering sleep mode; otherwise, it cannot enter Sleep mode.

Note: To use Event Wakeup function, we must clear Event Flag before entering Sleep/Deep Sleep mode. In case that Event Flag is not "0", system is unable to enter Sleep/Deep Sleep mode, instead, it will wake up immediately.

Sleep-on-exit

If the SLEEPONEXIT bit of the SCR is set to 1, when the processor completes the execution of an exception handler and returns to Thread mode, it immediately enters sleep mode. This mechanism is used in applications that only require the processor to run 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 in On and in normal mode voltage.	Keep in the setting before entering low power modes.
	WFE	Any event			
Deep Sleep	WFI+SLEEPDEEP=1	External input interrupt	All clocks are OFF	Keep in On but output voltage changes to Deep Sleep mode voltage.	Keep in the setting before entering low power modes.
	WFE+SLEEPDEEP=1	External input event			

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.

Note: If program is placed in external SPI FLASH memory, cache memory only supports the first 256KB.

6.4.2. Register Description

Register Map

Base Address : 0x5000_0000					
Name	Description		Address	Access	Reset value
SMU_SYSLOCK	System Control Signal Lock Register		0x5000_0000	R/W	0x0000_0000
SMU_PID	Produce ID Register		0x5000_0004	R	0xFFFF_0000
SMU_UID0	User Defined ID 0 Mirror Register		0x5000_0008	R	0xFFFF_FFFF
SMU_UID1	User Defined ID 1 Mirror Register		0x5000_000C	R	0xFFFF_FFFF
SMU_UID2	User Defined ID 2 Mirror Register		0x5000_0010	R	0xFFFF_FFFF
SMU_UID3	User Defined ID 3 Mirror Register		0x5000_0014	R	0xFFFF_FFFF
SMU_CACHE_CTRL	Cache Control Register		0x5000_0030	R/W	0x0000_0001
SMU_EXP_SADDR	EFLASH Execute only protect Range Start address Register		0x5000_0034	R/W	0x0000_0000
SMU_EXP_ENDADDR	EFLASH Execute only protect Range End address Register		0x5000_0038	R/W	0x0000_0000

Registers

SMU_SYSLOCK System Control Signal Lock Register								Address : 0x5000_0000
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								SysUnLOCK
7	6	5	4	3	2	1	0	
UnLOCK_KEY								

Bit	Name	Description	Access	Reset value
[31:9]	--	Reserved	R	0
[8]	UNLOCK_STS_FLAG (SysUnLOCK)	UnLOCK Status Flag 0 = Locked 1 = UnLocked Note: This bit status was based on UnLOCK_KEY operation.	R	0
[7:0]	UNLOCK_KEY1 UNLOCK_KEY2 (UnLOCK_KEY)	Specified System Control UnLock Key Write 0xAB and 0x12 to this UnLOCK_KEY consequently, the specified registers will be unlock. Note: If SysUnLOCK in unlocked state, write other words except 0x12 to UnLOCK_KEY, the specified registers will be locked. Specified Lock Protect Register as below RCU_CTRL ANALOG Control REGISTER	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

Product ID Register								Address : 0x5000_0004
31	30	29	28	27	26	25	24	
PID[15:8]								
23	22	21	20	19	18	17	16	
PID[7:0]								
15	14	13	12	11	10	9	8	
BID[15:8]								
7	6	5	4	3	2	1	0	
BID[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	PID	Product ID of Chip	R	0xFFFF
[15:0]	BID	Body ID of Chip	R	0x0000

User Defined ID 0 Register								Address : 0x5000_0008
31	30	29	28	27	26	25	24	
UID0[31:24]								
23	22	21	20	19	18	17	16	
UID0[23:16]								
15	14	13	12	11	10	9	8	
UID0[15:8]								
7	6	5	4	3	2	1	0	
UID0[7:0]								

Bit	Name	Description	Access	Reset value
[31:0]	UID0	User Defined ID 0	R	0xFFFF_FFFF

User Defined ID 1 Register								Address : 0x5000_000C
31	30	29	28	27	26	25	24	
UID1[31:24]								
23	22	21	20	19	18	17	16	
UID1[23:16]								
15	14	13	12	11	10	9	8	
UID1[15:8]								
7	6	5	4	3	2	1	0	
UID1[7:0]								

Bit	Name	Description	Access	Reset value
[31:0]	UID1	User Defined ID 1	R	0xFFFF_FFFF

SMU_UID2 User Defined ID 2 Register Address : 0x5000_0010

31	30	29	28	27	26	25	24
UID2[31:24]							
23	22	21	20	19	18	17	16
UID2[23:16]							
15	14	13	12	11	10	9	8
UID2[15:8]							
7	6	5	4	3	2	1	0
UID2[7:0]							

Bit	Name	Description	Access	Reset value
[31:0]	UID2	User Defined ID 2	R	0xFFFF_FFFF

SMU_UID3 User Defined ID 3 Register Address : 0x5000_0014

31	30	29	28	27	26	25	24
UID3[31:24]							
23	22	21	20	19	18	17	16
UID3[23:16]							
15	14	13	12	11	10	9	8
UID3[15:8]							
7	6	5	4	3	2	1	0
UID3[7:0]							

Bit	Name	Description	Access	Reset value
[31:0]	UID3	User Defined ID 3	R	0xFFFF_FFFF

SMU_CACHE_CTRL M0 Cache Control Register Address : 0x5000_0030

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

Cache_EN

Bit	Name	Description	Access	Reset value
[31:1]	--	Reserved	R	0
[0]	CACHE_ENABLE (Cache_EN)	Cache Function Enable 0 = Disabled, 1 = Cache Function enabled	R/W	1

SMU_EXP_SADDR EFLASH Execute only Protection Start address Register Address : 0x5000_0034

31	30	29	28	27	26	25	24
--							



GPCM1F SERIES PROGRAMMING GUIDE

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

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	EXP_Saddr	eFLASH Execute only protect Range Start address	R/W	0

SMU_EXP_ENDADDR EFLASH Execute only Protection End address Register								Address : 0x5000_0038
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
			EXP_Endaddr[15:8]					
7	6	5	4	3	2	1	0	
			EXP_Endaddr[7:0]					

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	EXP_Endaddr	eFLASH Execute only protect Range End address	R/W	0

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 or falling or both or level) and generated event pulse when event control enabled.

6.5.2. Function

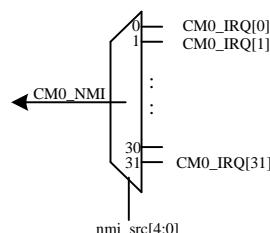
To generate the interrupt, the interrupt line should be configured and enabled. This is done by programming the trigger registers with the desired edge detection or level detection and by enabling the interrupt request by writing a '1' to the corresponding bit in the interrupt enable register. When the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

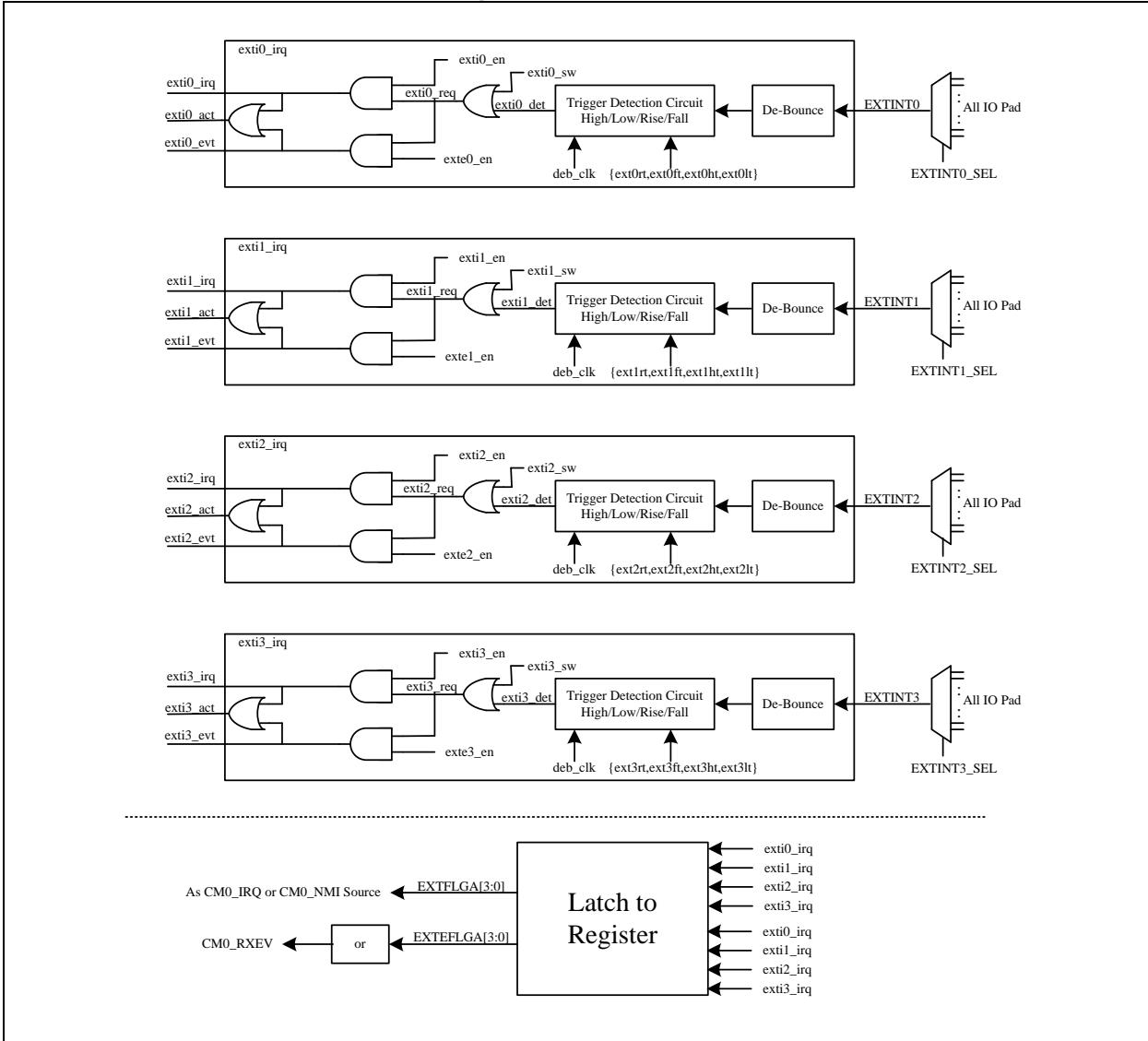
To generate the event, the event line should be configured and enabled. This is done by programming the trigger registers with the desired edge detection or level detection and by enabling the event request by writing a '1' to the corresponding bit in the event enable register. When the selected detection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

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

NMI interrupts source selection

NMI interrupt source of GPCM1F SERIALS can be configured as one of NVIC IRQ Numbers. The NMISRC are corresponded to NVIC IRQ Numbers.



External interrupt/event controller block diagram

Wakeup event management

The GPCM1FSeries is able to handle external or internal events in order to wake up the core (WFE). The wakeup event can be generated either by: Configure 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 as the pending bit corresponding to the event line is not set.

6.5.3. Register Description
Register Map

Base Address : 0x5000_4000				
Name	Description	Address	Access	Reset value
ITU_NMICTRL	NMI Interrupt Control Register	0x5000_4000	R/W	0x0000_0000
ITU_SWIRQ	Software Interrupt Request Register	0x5000_4004	R/W	0x0000_0000
ITU_EXTIEN	External Interrupt Enable Register	0x5000_4010	R/W	0x0000_0000
ITU_EXTEEN	External Event Enable Register	0x5000_4014	R/W	0x0000_0000
ITU_EXTRHT	External Rising Edge / High Level Trigger Register	0x5000_4018	R/W	0x0000_0000

Base Address : 0x5000_4000						
Name	Description			Address	Access	Reset value
ITU_EXTFLT	External Falling Edge / Low Level Trigger Register			0x5000_4020	R/W	0x0000_0000
ITU_EXTIFLG	External Interrupt Flag Register			0x5000_4028	R/W	0x0000_0000
ITU_EXTEFLG	External Event Flag Register			0x5000_402C	R/W	0x0000_0000

Registers

ITU_NMICTRL NMI Interrupt Control Register								Address : 0x5000_4000
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								NMIEN
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
NMISRC								

Bit	Name	Description	Access	Reset value
[31:17]	--	Reserved	R	0
[16]	INT_ENABLE (NMIEN)	NMI Interrupt Enabling bit. 0 = Disabled, 1 = Enabled	R/W	0
[15:5]	--	Reserved	R	0
[4:0]	SRC_SEL (NMISRC)	NMI Source Selection NMI Source numbers are referred to IRQ number of System Interrupt Vector Map. 0x1D: SRC_SEL_KEYCHG 0x1C: SRC_SEL_TM2 0x1B: SRC_SEL_TM1 0x1A: SRC_SEL_TM0 0x18: SRC_SEL_EXTI3 0x17: SRC_SEL_EXTI2 0x16: SRC_SEL_EXTI1 0x15: SRC_SEL_EXTI0 0x14: SRC_SEL_SPI1 0x13: SRC_SEL_SPI0 0x12: SRC_SEL_I2S 0x10: SRC_SEL_UART 0x0E: SRC_SEL_I2C 0x0D: SRC_SEL_TIMEBASE 0x0C: SRC_SEL_CTS_TMA1 0x0B: SRC_SEL_CTS_TMA0 0x0A: SRC_SEL_CCP1 0x09: SRC_SEL_CCP0 0x08: SRC_SEL_DAC_CH1 0x07: SRC_SEL_DAC_CH0 0x06: SRC_SEL_DS_ADC	R/W	0

Bit	Name	Description	Access	Reset value
		0x05: SRC_SEL_SAR_ADC 0x04: SRC_SEL_QD 0x03: SRC_SEL_MAC 0x02: SRC_SEL_DMA		

ITU_SW_IRQ Interrupt Request Register								Address : 0x5000_4004	
31	30	29	28	27	26	25	24	IRQ	
23	22	21	20	19	18	17	16	IRQ	
15	14	13	12	11	10	9	8	IRQ	
7	6	5	4	3	2	1	0	IRQ	

Bit	Name	Description	Access	Reset value
[31:0]	IRQ	Interrupt Request When the IRQ[n] is 0, setting IRQ[n] to 1 will generate an interrupt to Cortex™-M0 NVIC[n] When the IRQ[n] is 1 (mean an interrupt is assert), setting 1 to the MCU_bit[n] will clear the interrupt and setting IRQ[n] 0 has no effect Note: Refer to ICSR. VECTACTIVE to check the currently active exception.	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

ITU_EXTIEN External Interrupt Enable Register								Address : 0x5000_4010	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	EXT3IEN	EXT2IEN
								EXT1IEN	EXT0IEN

Bit	Name	Description	Access	Reset value
[31:4]	--	Reserved	R	0
[3]	EXT3_INT_ENABLE (EXT3IEN)	External Input 3 Interrupt Enable Bit number of EXTIEN means corresponding port. 0 = EXT3 INT request is disabled. 1 = EXT3 INT request is enabled.	R/W	0
[2]	EXT2_INT_ENABLE (EXT2IEN)	External Input 2 Interrupt Enable Bit number of EXTIEN means corresponding port. 0 = EXT2 INT request is disabled. 1 = EXT2 INT request is enabled.	R/W	0
[1]	EXT1_INT_ENABLE (EXT1IEN)	External Input 1 Interrupt Enable Bit number of EXTIEN means corresponding port. 0 = EXT1 INT request is disabled. 1 = EXT1 INT request is enabled.	R/W	0
[0]	EXT0_INT_ENABLE (EXT0IEN)	External Input 0 Interrupt Enable Bit number of EXTIEN means corresponding port. 0 = EXT0 INT request is disabled. 1 = EXT0 INT request is enabled.	R/W	0

ITU_EXTEEN External Input Event Enable Register								Address : 0x5000_4014	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	KCEEN	EXT3EEN
								EXT2EEN	EXT1EEN
								EXT0EEN	

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0
[4]	KEYCHG_EVT_ENABLE (KCEEN)	Key Change Event Enable 0: Key Change Event is disabled. 1: Key Change Event is enabled.	R/W	0
[3]	EXT3_EVT_ENABLE	External Input 3 Event Enable	R/W	0

Bit	Name	Description	Access	Reset value
	(EXT3EEN)	0 = EXT3 event request is disabled. 1 = EXT3 event request is enabled.		
[2]	EXT2_EVT_ENABLE (EXT2EEN)	External Input 2 Event Enable 0 = EXT2 event request is disabled. 1 = EXT2 event request is enabled.	R/W	0
[1]	EXT1_EVT_ENABLE (EXT1EEN)	External Input 1 Event Enable 0 = EXT1 event request is disabled. 1 = EXT1 event request is enabled.	R/W	0
[0]	EXT0_EVT_ENABLE (EXT0EEN)	External Input 0 Event Enable 0 = EXT0 event request is disabled. 1 = EXT0 event request is enabled.	R/W	0

ITU_EXTRHT External Input Rising Edge / High Level Trigger Register								Address : 0x5000_4018
31	30	29	28	27	26	25	24	
--	--	--	--	--	--	--	--	
23	22	21	20	19	18	17	16	
--	--	--	--	EXT3HT	EXT2HT	EXT1HT	EXT0HT	
15	14	13	12	11	10	9	8	
--	--	--	--	--	--	--	--	
7	6	5	4	3	2	1	0	
--	--	--	--	EXT3RT	EXT2RT	EXT1RT	EXT0RT	

Bit	Name	Description	Access	Reset value
[31:20]	--	Reserved	R	0
[19]	EXT3_HLVL_TRIG_ENA BLE (EXT3HT)	External Input 3 High Level Trigger Enable 0 = EXT3 high level trigger is disabled. 1 = EXT3 high level trigger is enabled.	R/W	0
[18]	EXT2_HLVL_TRIG_ENA BLE (EXT2HT)	External Input 2 High Level Trigger Enable 0 = EXT2 high level trigger is disabled. 1 = EXT2 high level trigger is enabled.	R/W	0
[17]	EXT1_HLVL_TRIG_ENA BLE (EXT1HT)	External Input 1 High Level Trigger Enable 0 = EXT1 high level trigger is disabled. 1 = EXT1 high level trigger is enabled.	R/W	0
[16]	EXT0HLVL_TRIG_ENAB LE (EXT0HT)	External Input 0 High Level Trigger Enable 0 = EXT0 high level trigger is disabled. 1 = EXT0 high level trigger is enabled.	R/W	0
[15:4]	--	Reserved	R	0
[3]	EXT3_RISING_TRIG_E NABLE (EXT3RT)	External Input 3 Rising Trigger Enable 0 = EXT3 rising trigger is disabled. 1 = EXT3 rising trigger is enabled.	R/W	0
[2]	EXT2_RISING_TRIG_E NABLE (EXT2RT)	External Input 2 Rising Trigger Enable 0 = EXT2 rising trigger is disabled. 1 = EXT2 rising trigger is enabled.	R/W	0
[1]	EXT1_RISING_TRIG_E NABLE	External Input 1 Rising Trigger Enable 0 = EXT1 rising trigger is disabled.	R/W	0

Bit	Name	Description	Access	Reset value
	(EXT1RT)	1 = EXT1 rising trigger is enabled.		
[0]	EXT0_RISING_TRIG_E NABLE (EXT0RT)	External Input 0 Rising Trigger Enable 0 = EXT0 rising trigger is disabled. 1 = EXT0 rising trigger is enabled.	R/W	0

ITU_EXTFLT External Input Falling Edge / Low Level Trigger Register								Address : 0x5000_4020
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--				EXT3LT	EXT2LT	EXT1LT	EXT0LT	
15	14	13	12	11	10	9	8	
--				EXT3FT	EXT2FT	EXT1FT	EXT0FT	
7	6	5	4	3	2	1	0	
--				EXT3FT	EXT2FT	EXT1FT	EXT0FT	

Bit	Name	Description	Access	Reset value
[31:20]	--	Reserved	R	0
[19]	EXT3_LLVL_TRIG_ENA BLE (EXT3LT)	External Input 3 Low Level Trigger Enable 0 = EXT3 low level trigger is disabled. 1 = EXT3 low level trigger is enabled.	R/W	0
[18]	EXT2_LLVL_TRIG_ENA BLE (EXT2LT)	External Input 2 Low Level Trigger Enable 0 = EXT2 low level trigger is disabled. 1 = EXT2 low level trigger is enabled.	R/W	0
[17]	EXT1_LLVL_TRIG_ENA BLE (EXT1LT)	External Input 1 Low Level Trigger Enable 0 = EXT1 low level trigger is disabled. 1 = EXT1 low level trigger is enabled.	R/W	0
[16]	EXT0_LLVL_TRIG_ENA BLE (EXT0LT)	External Input 0 Low Level Trigger Enable 0 = EXT0 low level trigger is disabled. 1 = EXT0 low level trigger is enabled.	R/W	0
[15:4]	--	Reserved	R	0
[3]	EXT3_FALLING_TRIG_ENABLE (EXT3FT)	External Input 3 Falling Trigger Enable 0 = EXT3 falling trigger is disabled. 1 = EXT3 falling trigger is enabled.	R/W	0
[2]	EXT2_FALLING_TRIG_ENABLE (EXT2FT)	External Input 2 Falling Trigger Enable 0 = EXT2 falling trigger is disabled. 1 = EXT2 falling trigger is enabled.	R/W	0
[1]	EXT1_FALLING_TRIG_ENABLE (EXT1FT)	External Input 1 Falling Trigger Enable 0 = EXT1 falling trigger is disabled. 1 = EXT1 falling trigger is enabled.	R/W	0
[0]	EXT0_FALLING_TRIG_ENABLE (EXT0FT)	External Input 0 Falling Trigger Enable 0 = EXT0 falling trigger is disabled. 1 = EXT0 falling trigger is enabled.	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

ITU_EXTIFLG External Input Interrupt Flag Register								Address : 0x5000_4028	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	--	
				EXT3IFLG	EXT2IFLG	EXT1IFLG	EXT0IFLG		

Bit	Name	Description	Access	Reset value
[31:4]	--	Reserved	R	0
[3]	EXT3_INT_FLAG (EXT3IFLG)	External 3 Interrupt Flag 0 = EXT3 no trigger request occurred. 1 = EXT3 INT request occurred. Note: Cleared by writing 1.	R/W	0
[2]	EXT2_INT_FLAG (EXT2IFLG)	External 2 Interrupt Flag 0 = EXT2 no trigger request occurred. 1 = EXT2 INT request occurred. Note: Cleared by writing 1.	R/W	0
[1]	EXT1_INT_FLAG (EXT1IFLG)	External 1 Interrupt Flag 0 = EXT1 no trigger request occurred. 1 = EXT1 INT request occurred. Note: Cleared by writing 1.	R/W	0
[0]	EXT0_INT_FLAG (EXT0IFLG)	External 0 Interrupt Flag 0 = EXT0 no trigger request occurred. 1 = EXT0 INT request occurred. Note: Cleared by writing 1.	R/W	0

ITU_EXTEFLG External Event Flag Register								Address : 0x5000_402C	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	--	
				KCGEFLG	EXT3EFLG	EXT2EFLG	EXT1EFLG	EXT0EFLG	

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0
[4]	KEYCHG_EVT_FLAG (KCGEFLG)	Key Change Event Flag 0: No trigger request occurred. 1: Key Change Event occurred.	R/W	0
[3]	EXT3_EVT_FLAG (EXT3EFLG)	External 3 Event Flag 0 = No trigger request occurred. 1 = EXT3 request occurred.	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
		Note: Cleared by writing 1.		
[2]	EXT2_EVT_FLAG (EXT2EFLG)	External 2 Event Flag 0 = No trigger request occurred. 1 = EXT2 request occurred. Note: Cleared by writing 1.	R/W	0
[1]	EXT1_EVT_FLAG (EXT1EFLG)	External 1 Event Flag 0 = No trigger request occurred. 1 = EXT1 request occurred. Note: Cleared by writing 1.	R/W	0
[0]	EXT0_EVT_FLAG (EXT0EFLG)	External 0 Event Flag 0 = No trigger request occurred. 1 = EXT0 request occurred. Note: Cleared by writing 1.	R/W	0

7. Analog Control Unit

7.1. Introduction

Analog control unit is designed for GPCM1F Serials analog devices Control

7.2. Features

- Voltage Regulator 3.3V(REG33)
- Voltage Regulator 1.5V(REG15) ,LVR and BVD
- Internal 8.192M Oscillator(IOSC8M)
- Internal 32768 Oscillator(IOSC32K)
- External Crystal Oscillator(X32K)
- Phase Lock-Loop(PLL)
- Audio PWM (Aud-PWM)

7.3. Function

Voltage Regulator 3.3V (REG33)

Voltage regulator 3.3V supplies the 2 type power of SPI IO (PAD IOB5:0]and ADC. When **ACU_REG33_CTRL.REG33_ADC_EN** is enabled, **REG33** supplies the ADC IO power. The SPI IO power is always supplied. When **ACU_REG33_CTRL.REG33_SPI_SLEEP** is setting, **REG33** will supply the 2.7V Voltage to the power of SPI IO and go to the sleep mode.

Low Voltage Reset (LVR)

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

Battery Voltage Detector (BVD)

In order to satisfy some application usage,GPCM1F SERIALS provides battery voltage detector. BVD function can be enable or disable by setting **ACU_BVD_CTRL.BVDEN**. the result of BVD detect voltage could be show by **ACU_BVD_CTRL.BVD[2:0]**.

Internal 8.192M Oscillator(IOSC8M)

The HSIRC8M clock signal is generated from an internal 8.192 MHz RC oscillator and can be used directly as a system clock or to be used as one of PLL input. The HSIRCRDY flag in the **CLK_CLKSTS.HSIRCRDY** indicates if the high-speed internal oscillator is stable or not.

Internal 32768 Oscillator(IOSC32K)

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

External 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 if the external oscillator is stable or not.

Phase Lock-Loop(PLL)

The PLL provides a frequency multiplier starting from one of the following clock sources: IOSC8M clock

The configuration of PLL(multiplication value) by set **ACU_PLL_CTRL.DIVC[2:0]**. The configuration behave must be done before enabling the PLL. Each PLL should be enabled after its input clock becomes stable (ready flag). Once the CPU clock switched to

PLL, these parameters cannot be changed. The PLLRDY flag in the **CLK_CLKSTS.PLLRDY** indicates if the PLL is stable or not.

7.4. Register Description

Register Map

Base Address : 0x5000_5000						
Name	Description			Address	Type	Reset value
ACU_REG33_CTRL	Macro REG33 Control			0x5000_5000	R/W	0x0000_001A
ACU_PLL_CTRL	Macro PLL Control			0x5000_500C	R/W	0x0000_0548
ACU_X32K_CTRL	Macro X32K Control			0x5000_5010	R/W	0x0000_0007
ACU_I32K_CTRL	Macro IOSC32K Control			0x5000_5014	R/W	0x0000_0100
ACU_BVD_CTRL	Macro BVD Control			0x5000_5030	R/W	0x0000_000F
ACU_APAD_CTRL	Macro Analog Pad Enable			0x5000_5034	R/W	0x0000_0000

Register Function

ACU_REG33_CTRL REG33 Ctrl Register								Address : 0x5000_5000
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
--				REG33_ADC_EN	REG33_SPI_VSEL[1:0]			--

Bit	Name	Description	Access	Reset value										
[31:5]	--	Reserved	R	0										
[4]	DSADC_REG_EN_ENA BLE (REG33_ADC_EN)	DS-ADC (Mic) Regulator Enable: 1: DS-ADC power enabled 0: DS-ADC power disabled Note.1: DS-ADC REG and REG33 SPI REG share a power in pin. This bit determines whether to disable DS-ADC Power IN (regardless of normal or sleep mode). Thus, remember to set it to disabled in sleep mode to save power and set it back to enabled in normal mode; otherwise, DS-ADC will have no power. Note.2: This bit is protected by UnLOCK_KEY .	R/W	1										
[3:2]	OUTPUT_VOL_SEL (REG33_SPI_VSEL)	3.3V SPI power select bits for normal mode <table border="1" data-bbox="500 1695 1056 1875"> <tr> <td>V33_SPI_SEL[1:0]</td> <td>V15</td> </tr> <tr> <td>00</td> <td>2.7V +/- 3%</td> </tr> <tr> <td>01</td> <td>3.0V +/- 3%</td> </tr> <tr> <td>10</td> <td>3.3V +/- 3%</td> </tr> <tr> <td>11</td> <td>3.6V +/- 3%</td> </tr> </table> Note: This bit is protected by UnLOCK_KEY Note: When system powers on, the REG33_SPI Regulator output voltage will be reloaded as the setting of manufactory option (body option). These option bits allow users to re-assign REG33 SPI Regulator output voltage	V33_SPI_SEL[1:0]	V15	00	2.7V +/- 3%	01	3.0V +/- 3%	10	3.3V +/- 3%	11	3.6V +/- 3%	R/W	0x2
V33_SPI_SEL[1:0]	V15													
00	2.7V +/- 3%													
01	3.0V +/- 3%													
10	3.3V +/- 3%													
11	3.6V +/- 3%													



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
		after executing a program.		
[1:0]	--	Reserved	R/W	0x2

ACU_PLL_CTRL PLL Ctrl Register								Address : 0x5000_500C
31	30	29	28	27	26	25	24	
			--					
23	22	21	20	19	18	17	16	
			--					
15	14	13	12	11	10	9	8	
R			--			DIVC[2:0]		
7	6	5	4	3	2	1	0	
			--					PLL80M_EN

Bit	Name	Description	Access	Reset value
[31:11]	--	Reserved	R	0
[10:8]	CLK_SEL (DIVC)	PLL_CKO selection:	R/W	0x5
		DIVC[2:0] P8M_CKO		
		000 24.576M		
		001 32.768M		
		010 40.960M		
		011 49.192M		
		100 57.344M		
		101 65.536M		
		110 73.728M		
		111 81.920M		
[7:1]	--	Reserved	R	0x24
[0]	PLL_ENABLE (PLL80M_EN)	PLLS enable signal 1=enabled. When HCLK selection PLL (by 0x5000_1020[2:0] setting), it will auto enable(=1) Note: This bit is protected by UnLOCK_KEY	R/W	0

ACU_X32K_CTRL X32K Ctrl Register								Address : 0x5000_5010
31	30	29	28	27	26	25	24	
			--					
23	22	21	20	19	18	17	16	
			--					
15	14	13	12	11	10	9	8	
			--					
7	6	5	4	3	2	1	0	
			--		CLK32KSLP_E N	X32K_STRON G	X32K_ENB	

Bit	Name	Description	Access	Reset value
[31:3]	--	Reserved	R	0
[2]	32K_SLP_ENABLE	For System go halt mode, needs to set this bit to provide 32KHz Clock	R/W	1

Bit	Name	Description	Access	Reset value
	(CLK32KSLP_EN)	1: 32K control by X32K_ENB or I32K_ENB 0: will disable X32K and I32K in DeepSleep mode Note: This bit is protected by UnLOCK_KEY .		
[1]	MODE_STRONG (STRONG)	STRONG mode option 1: Strong mode 0: Weak mode Note: This bit is protected by UnLOCK_KEY .	R/W	1
[0]	X32K_ENABLE (X32K_ENB)	Crystal enable, low active 1: X32K disabled 0: X32K enabled Note: This bit is protected by UnLOCK_KEY .	R/W	1

ACU_I32K_CTRL I32K Ctrl Register								Address: 0x5000_5014
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	I32K_ENB
--								

Bit	Name	Description	Access	Reset value
[31:1]	--	Reserved	R	-
[0]	I32K_ENABLE (I32K_ENB)	Enable IOSC CKT, low active 1: I32K disabled 0: I32K enabled Note: This bit is protected by UnLOCK_KEY .	R/W	0

ACU_BVD_CTRL Lower Voltage Detection Ctrl Register								Address : 0x5000_5030
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	BVDEN
--								BVD[2:0]

Bit	Name	Description	Access	Reset value
[31:4]	--	Reserved	R	0
[3]	BVD_CTRL_ENABLE (BVDEN)	Battery voltage detect enable (High active) 0: low voltage detect function disabled 1: low voltage detect function enabled	R/W	1
[2:0]	BVD_LVL_SEL	Battery voltage detect trigger level, (Read only)	R	111

	(BVD)	BVD_SEL[2:0]	BVD Voltage				
		000	$\leq 1.94V +/- 7.5\%$				
		001	$2.134V +/- 7.5\%$				
		010	$2.328V +/- 7.5\%$				
		011	$2.522V +/- 7.5\%$				
		100	$2.716V +/- 7.5\%$				
		101	$2.91V +/- 7.5\%$				
		110	$3.2V +/- 7.5\%$				
		111	$>3.2V +/- 7.5\%$				

ACU_APAD_CTRL Analog Pad Control Register								Address : 0x5000_5034
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								X32K_PAD_E
								N
7	6	5	4	3	2	1	0	
LINEINPAD_EN[7:0]								

Bit	Name	Description	Access	Reset value																				
[31:9]	--	Reserved	R	0																				
[8]	X32K_PAD_ENABLE (X32K_PAD_EN)	Crystal 32KHz Pad (IOA[14:13]) Enable 1: Enabled 0: Disabled	R/W	0																				
[7:0]	LINE_PAD_EN (LINEINPAD_EN[7:0])	1: line In PAD Enable <table border="1"> <tr> <th>LINEINPAD_EN</th> <th>IO</th> <th>LINEINPAD_EN</th> <th>IO</th> </tr> <tr> <td>LINEINPAD_EN[0]</td> <td>IOA24</td> <td>LINEINPAD_EN[1]</td> <td>IOA23</td> </tr> <tr> <td>LINEINPAD_EN[2]</td> <td>IOA22</td> <td>LINEINPAD_EN[3]</td> <td>IOA21</td> </tr> <tr> <td>LINEINPAD_EN[4]</td> <td>IOA20</td> <td>LINEINPAD_EN[5]</td> <td>IOA19</td> </tr> <tr> <td>LINEINPAD_EN[5]</td> <td>IOA18</td> <td>LINEINPAD_EN[7]</td> <td>IOA17</td> </tr> </table>	LINEINPAD_EN	IO	LINEINPAD_EN	IO	LINEINPAD_EN[0]	IOA24	LINEINPAD_EN[1]	IOA23	LINEINPAD_EN[2]	IOA22	LINEINPAD_EN[3]	IOA21	LINEINPAD_EN[4]	IOA20	LINEINPAD_EN[5]	IOA19	LINEINPAD_EN[5]	IOA18	LINEINPAD_EN[7]	IOA17	R/W	0
LINEINPAD_EN	IO	LINEINPAD_EN	IO																					
LINEINPAD_EN[0]	IOA24	LINEINPAD_EN[1]	IOA23																					
LINEINPAD_EN[2]	IOA22	LINEINPAD_EN[3]	IOA21																					
LINEINPAD_EN[4]	IOA20	LINEINPAD_EN[5]	IOA19																					
LINEINPAD_EN[5]	IOA18	LINEINPAD_EN[7]	IOA17																					

8. DMA Control Unit

8.1. Introduction

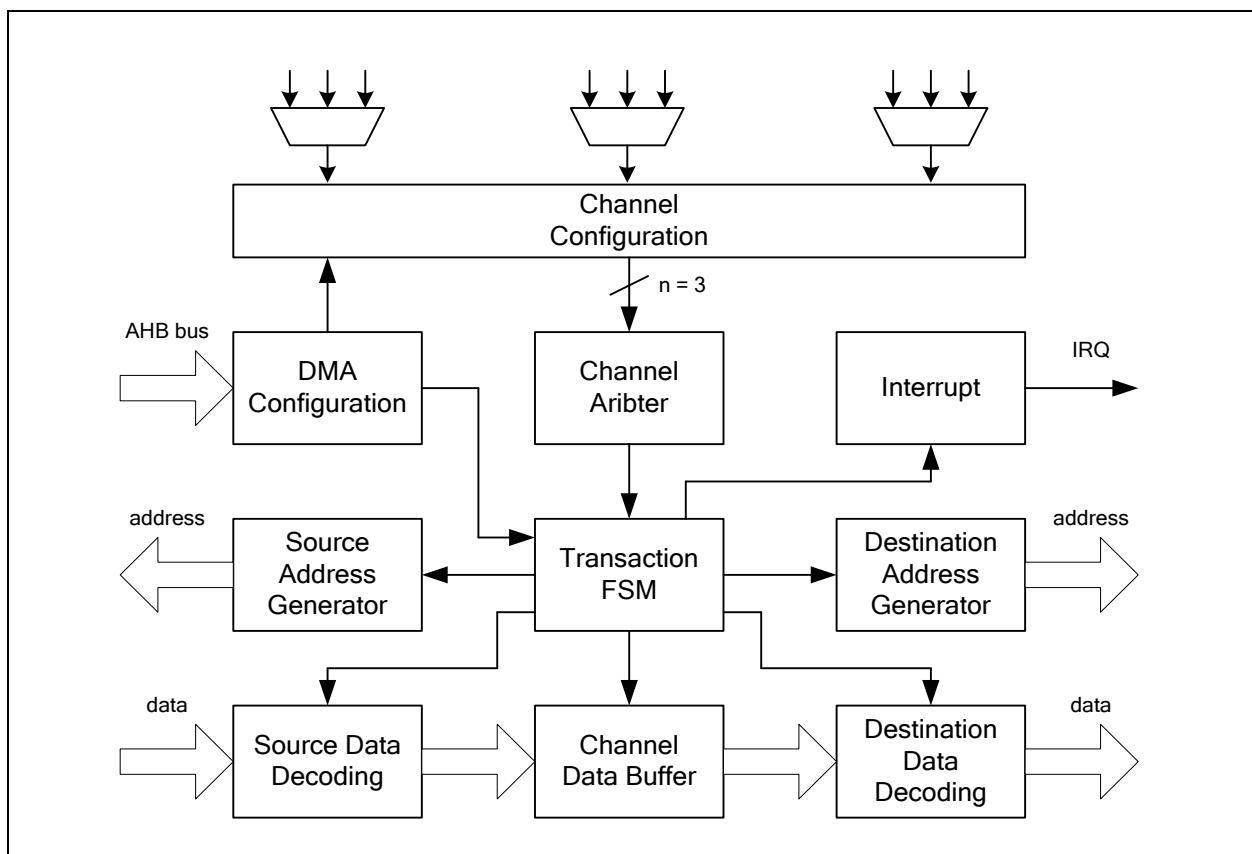
Direct memory access (DMA) is used in order to provide high-speed data transfer between peripherals and memory as well as memory to memory. Data can be quickly moved by DMA without any CPU actions. This keeps CPU resources free for other operations.

The DMA controllers have 3 channels, each dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority between DMA requests.

8.2. Features

- Three independently configurable channels (requests)
- Each of the channels is connected to dedicated hardware DMA request, software trigger is also supported on each channel.
- Priorities between requests were based on channel number, such as request in channel 1 has higher priority than request channel 2.
- Independent source and destination transfer size (8-bit, 16-bit, 32-bit).
- Support for burst operation.
- DMA Transfer complete send a interrupt request for each channel
- Support Memory-to-memory transfer, peripheral-to-memory, and memory-to-peripheral transfers with the programmable 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.

Note: DMA cannot be disabled during execution, or it may cause DMA to be pending. To terminate DMA execution, do the followings:

- (1) Before turning off DMA, turn the DMA triggering source off first , e.g. DSADC or Timer.
- (2) DMA stopped.
- (3) Initialize DMA and re-trigger DMA.
- (4) Enable DMA triggering source again.

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. Suppose we need to move the DMA data into two buffers, Buffer A and Buffer B, with data length of buffer_size.

Dual Mode

DMA allows users to issue two consecutive DMA triggering instructions to keep DMA work continuously. If only one DMA triggering instruction is issued, DMA will stop moving data after one set of data is moved completely, and it will automatically jump to the DMA ISR (Interrupt Service Routine) if DMA INT is enabled. Suppose a dual mode is activated, in addition to the DMA finishes the first set of data moving to Buffer A, it will continue to move the second set of data to Buffer B while CPU is executing the DMA ISR to notify users the first data moving is finished.

The way to activate the dual mode and the issues may arise:

- a. To activate the dual mode, simply issues two DMA triggering instructions such as the box in red below:

```

/*
 * DMA0 init
 */
DMA_Init(DMA0, DMA_REQSEL_MEM, DMA_SRC_DATA_16B, DMA_SRC_ADDR_INC, DMA_DST_DATA_16B, DMA_DST_ADDR_INC);
DMA_InstallIsrService(DMA0, DmaIsr_Dma0Trigger);
DMA_EnableInt(DMA0);
NVIC_EnableIRQ(DMA IRQn);

DMA_Trigger(DMA0, (uint32_t)&DMASrcBuf, (uint32_t)&DMADstBuf, 20);           // DMA0 first trigger
DMA_Trigger(DMA0, (uint32_t)&DMASrcBuf[10], (uint32_t)&DMADstBuf[10], 20); // DMA0 second trigger

```

- b. When DMA has moved the data completely, it will enter DMA ISR to notify users (if DMA INT is enabled) the data movement has been completed. Since the dual mode is enabled, it will triggered DMA INTs.

```

void DmaIsr_Dma0Trigger()
{
    DMAINTFlag = 1;
    DMA_INT->INTSTS = DMA_INTSTS_DMA0_DONE_INT_FLAG;           // Clear DMA0 INT Flag
}

```

- c. Suppose we'd like to keep dual mode working continuously, do not place the DMA triggering instruction inside the DMAISR(). That is because when the second DMA_Trigger is triggered before the first DMA_Trigger's ISR execution leaving out from DMA ISR(), a incoming third DMA_Trigger instruction will replace the second DMA_Trigger instruction. Thus, the second DMA_Trigger will never being executed in this case.

- d. To avoid the issue depicted in (c), we recommend users add instruction such as DMAINTFlag= 1 in DMA ISR routine, and check it in the main loop for judgement measure. When DMAINTFlag is checked "1", execute the DMA_Trigger instruction. In this case, each dual mode trigger will be executed normally and correctly.

Note: Under the debug mode of G+IDE for ARM or Keil tool, if Dual mode is enabled and a breakpoint is set, DMA will not be stopped after its first execution is completed. Instead, it will keep executing until the second DMA is completed. Thus, the status of the first execution cannot be seen.

```

while(1)
{
    if(DMAINTFlag == 1)
    {
        DMA_Trigger(DMA0, (uint32_t)&DMASrcBuf, (uint32_t)&DMADstBuf[20], 20);
        DMAINTFlag = 0;
    }
    WDT_Clear();
}

```

8.4.4. Programmable data width and data alignment

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

Source width	Destination width	Transfer Number	Source content: Address / Data	Transfer operation	Destination content: Address / Data
8	8	4	@0x0 / A0	1: Read 0xA0 @0x0 Write A0 @ 0x0	@0x0 / A0
			@0x1 / A1	2: Read 0xA1 @0x1 Write A1 @ 0x1	@0x1 / A1
			@0x2 / A2	3: Read 0xA2 @0x2 Write A2 @ 0x2	@0x2 / A2
			@0x3 / A3	4: Read 0xA3 @0x3 Write A3 @ 0x3	@0x3 / A3
8	16	4	@0x0 / A0	1: Read 0xA0 @0x0 Write 0x00A0 @ 0x0	@0x0 / 00A0

Source width	Destination width	Transfer Number	Source content: Address / Data	Transfer operation	Destination content: Address / Data
			@0x1 / A1 @0x2 / A2 @0x3 / A3	2: Read 0xA1 @ 0x1 Write 0x00A1 @ 0x2 3: Read 0xA2 @ 0x2 Write 0x00A2 @ 0x4 4: Read 0xA3 @ 0x3 Write 0x00A3 @ 0x6	@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 be occupied by another master such CPU over 256 system clocks, the transfer error flag (TEIF) will be asserted.

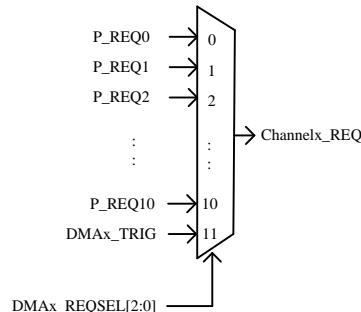
8.4.6. Interrupts

An interrupt can be produced on a Transfer complete or Transfer error for each DMA channel. 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 channel is associated with a DMA request that can be selected out of 8 possible peripheral requests. The selection is controlled by the DMA_REQSEL[3:0] bits in the DMA_CTRLx register.

Channel selection diagram

DMA request mapping

Peripherals	Channel 0/1/2
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
NC	P-REQ9
NC	P-REQ10
DMA_TRIG	P-REQ11

8.5. Register Description
Register Map

Base Address : 0x500F_0000				
Name	Description	Address	Access	Reset value
DMA_INTSTS	DMA Interrupt Status Register	0x500F_0000	R/W	0x0000_0000
DMA0_CTRL	DMA Channel 0 Control Register	0x500F_0010	R/W	0x0000_0000
DMA0_DTN	DMA Channel 0 Data Transfer Number Register	0x500F_0014	R/W	0x0000_0000
DMA0_SRCADR	DMA Channel 0 Source Address Register	0x500F_0018	R/W	0x0000_0000
DMA0_DSTADR	DMA Channel 0 Destination Address Register	0x500F_001C	R/W	0x0000_0000
DMA1_CTRL	DMA Channel 1 Control Register	0x500F_0020	R/W	0x0000_0000
DMA1_DTN	DMA Channel 1 Data Transfer Number Register	0x500F_0024	R/W	0x0000_0000
DMA1_SRCADR	DMA Channel 1 Source Address Register	0x500F_0028	R/W	0x0000_0000
DMA1_DSTADR	DMA Channel 1 Destination Address Register	0x500F_002C	R/W	0x0000_0000
DMA2_CTRL	DMA Channel 2 Control Register	0x500F_0030	R/W	0x0000_0000
DMA2_DTN	DMA Channel 2 Data Transfer Number Register	0x500F_0034	R/W	0x0000_0000
DMA2_SRCADR	DMA Channel 2 Source Address Register	0x500F_0038	R/W	0x0000_0000
DMA2_DSTADR	DMA Channel 2 Destination Address Register	0x500F_003C	R/W	0x0000_0000

Registers Function

DMA_INTSTS DMA Interrupt Status Register								Address : 0x500F_0000	
31	30	29	28	27	26	25	24		
--	--	--	--	--	--	--	--		
23	22	21	20	19	18	17	16		
--	--	--	--	--	--	--	--		
15	14	13	12	11	10	9	8	DMA2_TEIF	DMA1_TEIF
--	--	--	--	--	--	--	--	DMA0_TEIF	
7	6	5	4	3	2	1	0	DMA2_TCIF	DMA1_TCIF
--	--	--	--	--	--	--	--	DMA0_TCIF	

Bit	Name	Description	Access	Reset value
[31:11]	--	Reserved	R	0
[10]	DMA2_TX_ERR_INT_FL AG (DMA2_TEIF)	Channel 2 transfer error flag 0 = No transfer error on channel 2. 1 = Transfer error on channel 2. Note: It's cleared by software writing 1.	R/W	0
[9]	DMA1_TX_ERR_INT_FL AG (DMA1_TEIF)	Channel 1 transfer error flag 0 = No transfer error on channel 1. 1 = Transfer error on channel 1. Note: It's cleared by software writing 1.	R/W	0
[8]	DMA0_TX_ERR_INT_FL AG (DMA0_TEIF)	Channel 0 transfer error flag 0 = No transfer error on channel 0. 1 = Transfer error on channel 0. Note: It's cleared by software writing 1.	R/W	0
[7:3]	--	Reserved	R	0
[2]	DMA2_DONE_INT_FLA G (DMA2_TCIF)	Channel 2 transaction complete flag 0 = No transfer complete on channel 2. 1 = Transfer completed on channel 2. Note: It's cleared by software writing 1.	R/W	0
[1]	DMA1_DONE_INT_FLA G (DMA1_TCIF)	Channel 1 transaction complete flag 0 = No transfer complete on channel 1. 1 = Transfer completed on channel 1. Note: It's cleared by software writing 1.	R/W	0
[0]	DMA0_DONE_INT_FLA G (DMA0_TCIF)	Channel 0 transaction complete flag 0 = No transfer complete on channel 0. 1 = Transfer completed on channel 0. Note: It's cleared by software writing 1.	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

DMA0_CTRL	DMA Channel 0 Control Register							Address : 0x500F_0010
DMA1_CTRL	DMA Channel 1 Control Register							Address : 0x500F_0020
DMA2_CTRL	DMA Channel 2 Control Register							Address : 0x500F_0030
31	30	29	28	27	26	25	24	
--						DMA_DBF	DMA_BUSY	
23	22	21	20	19	18	17	16	
--						DMA_TRG		
15	14	13	12	11	10	9	8	
DMA_REQSEL				DMA_SRCSIZE	DMA_DSTSIZE			
7	6	5	4	3	2	1	0	
--	DMA_BURST	DMA_SRCINC	DMA_DSTINC	DMA_CIRC	DMA_TEIE	DMA_TCIE	DMA_EN	

Bit	Name	Description	Access	Reset value
[31:26]	--	Reserved	R	0
[25]	DMA_FULL_FLAG (DMA_DBF)	Channel x Double Buffer Full Flag 0= not full 1= full	R	0
[24]	DMA_BUSY_FLAG (DMA_BUSY)	Channel x operation status 0 = Idle 1 = Busy When the DMA is activated and writes the DMAx_DTN again, this bit will be set to 1. The value of DMAx_SRCADR and DMAx_DSTADR can be updated before write operation on DMAx_DTN. When the current DMA action is completed, CPU will automatically reload the values in these three registers and perform the next DMA transfer.	R	0
[23:17]	--	Reserved	R	0
[16]	DMA_START (DMA_TRG)	Channel x software trigger request 0 = Disabled. 1 = Enabled. Note: When DMA accepted the DMA_TRG, it will be cleared by H.W.	R/W	0
[15:12]	DMA_REQSEL	Channel x Peripheral REQ Selection 0000 = P-REQ0.(ADC_REQ0) 0001 = P-REQ1.(DAC_REQ0) 0010 = P-REQ2.(DAC_REQ1) 0011 = P-REQ3.(SPI0) 0100 = P-REQ4.(UART) 0101 = P-REQ5.(I2S) 0110 = P-REQ6.(I2C) 0111 = P-REQ7.(ADC_REQ1) 1000 = P-REQ8.(SPI1) 1011 = P-REQ11.(DMA_TRIGGER) Note: DMA_REQSEL only can be changed when DMA_EN = 0.	R/W	0
[11:10]	DMA_SRCSIZE	Channel x source data size 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = Reserved	R/W	0
[9:8]	DMA_DSTSIZE	Channel x destination data size	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
		00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = Reserved		
[7]	--	Reserved	R	0
[6]	DMA_BURST_ENABLE (DMA_BURST)	DMA Burst Transfer Mode 0 = Disabled 1 = Enabled Note: When DMA received a first request, it will complete this transaction automatically, based on transfer number. Note: DMA will ignore any request during transferring until transaction completed.	R/W	0
[5]	DMA_SRCINC_ENABLE (DMA_SRCINC)	Channel x source address increment mode 0 = Disabled, 1 = Enabled	R/W	0
[4]	DMA_DSTINC_ENABLE (DMA_DSTINC)	Channel x destination address increment mode 0 = Disabled, 1 = Enabled	R/W	0
[3]	DMA_CIRC_ENABLE (DMA_CIRC)	Channel x circular mode 0 = Disabled, 1 = Enabled	R	0
[2]	DMA_ERR_INT_ENABL E (DMA_TEIE)	Channel x transfer error interrupt enabling bit 0 = Disabled, 1 = Enabled	R/W	0
[1]	DMA_DONE_INT_ENAB LE (DMA_TCIE)	Channel x transfer complete interrupt enabling bit 0 = Disabled, 1 = Enabled.	R/W	0
[0]	DMA_ENABLE (DMA_EN)	DMA Channel x Enabling bit 0 = Disabled 1 = Enabled Note: DMA_EN can be cleared by SW or HW. Note: When one transaction completed, DMA_EN will be cleared by H.W. Note: DMA_EN can't be set to Enable when destination address from 0x0000_ 0000 ~ 0x1FFF_FFFF or source address from 0x00FF_FFFF to destination address 0x500F_0100 Note: DMA_EN can't be set to Enable when DMA_DTNx = 0.	R/W	0

DMA0_DTN DMA Channel 0 Data Transfer Number Register Address : 0x500F_0014

DMA1_DTN DMA Channel 1 Data Transfer Number Register Address : 0x500F_0024

DMA2_DTN DMA Channel 2 Data Transfer Number Register Address : 0x500F_0034

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

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DMA_DTN	Channel x data transfer number counter Number of data to be transferred. If this register is zero, no transaction can be served whether the DMA channel (DMA_EN) is enabled or not. When the DMA channel is enabled from disable state, DMA_DTN will be reloaded to internal counter, and counter will be decreased by one every transfer completed. During DMA channel enabled, the internal counter register will not be updated even DMA_DTN changed by software.	R/W	0

DMA0_SRCADR DMA Channel 0 Source Address Register **Address : 0x500F_0018**

DMA1_SRCADR DMA Channel 1 Source Address Register **Address : 0x500F_0028**

DMA2_SRCADR DMA Channel 2 Source Address Register **Address : 0x500F_0038**

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

Bit	Name	Description	Access	Reset value
[31:0]	DMA_SRCADR	Channel x source data transfer address Base address of the source area from which the data will be read. When DMA_SRCSIZE is 01 (16-bit), the DMA_SRCADR[0] bit is ignored. Access is automatically aligned to a half-word address. When DMA_SRCSIZE is 10 (32-bit), DMA_SRCADR[1:0] are ignored. Access is automatically aligned to a word address. Note: Source address doesn't support from 0x00000000 ~ 0xFFFFFFFF.	R/W	0

DMA0_DSTADR DMA Channel 0 Destination Address Register **Address : 0x500F_001C**

DMA1_DSTADR DMA Channel 1 Destination Address Register **Address : 0x500F_002C**

DMA2_DSTADR DMA Channel 2 Destination Address Register **Address : 0x500F_003C**

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

Bit	Name	Description	Access	Reset value
[31:0]	DMA_DSTADR	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 ~ 0x1FFFFFFF.	R/W	0

9. CCP0/1 Control

9.1. Introduction

GPCM1F has two sets of Comparison, Capture and PWM Function(CCP) and 8 channels of PWM outputs.

9.2. Feature

- Supports 2xCCP Function CCP0 and CCP1.
- Each CCP all supports 4 Chanel Capture mode, Comparison mode and PWMIO modes.
- Each CCP clock source has HCLK, HCLK_div2, ...HCLK_div4096, TimerA, TimerB, and TimerC.
- Each CCP all has output adc_trig to Trigger ADC Sample.
- CCP1 Control come from CCP0 to support 8x PWM output at the same time
- Each CCP all has independent Interrupt.

9.3. Block Diagrams

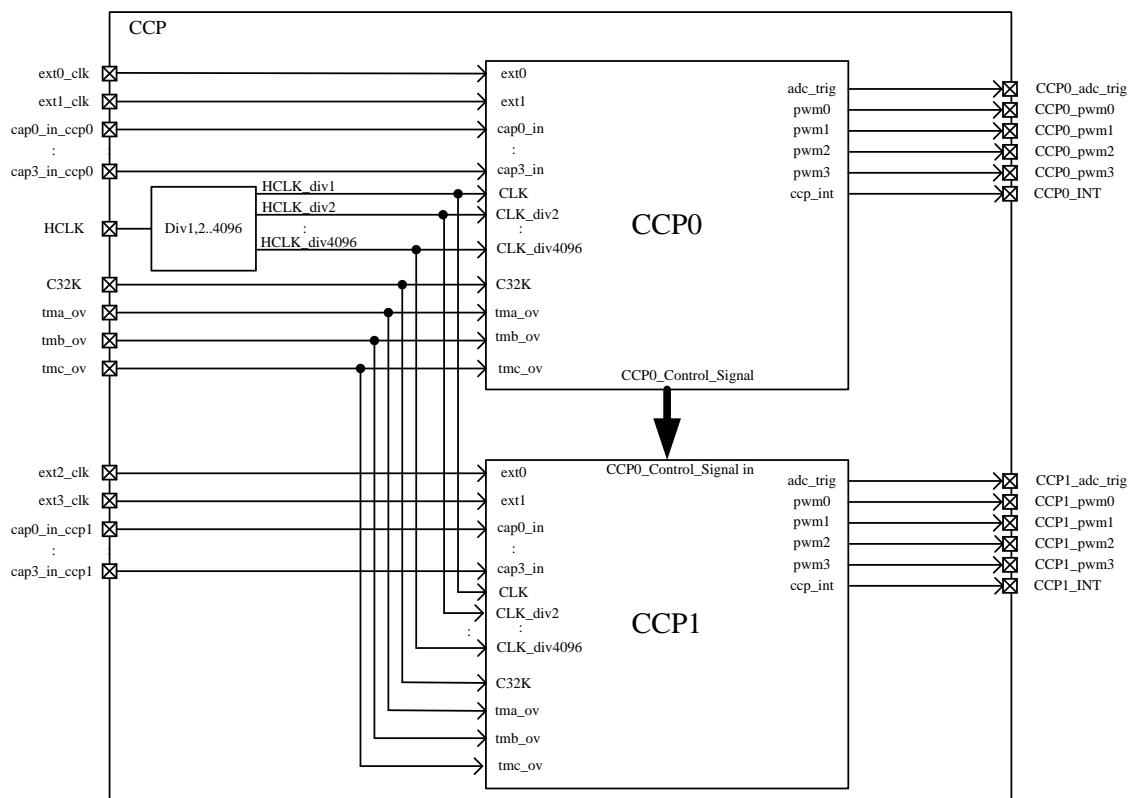


Fig.8.1 CCP Structure

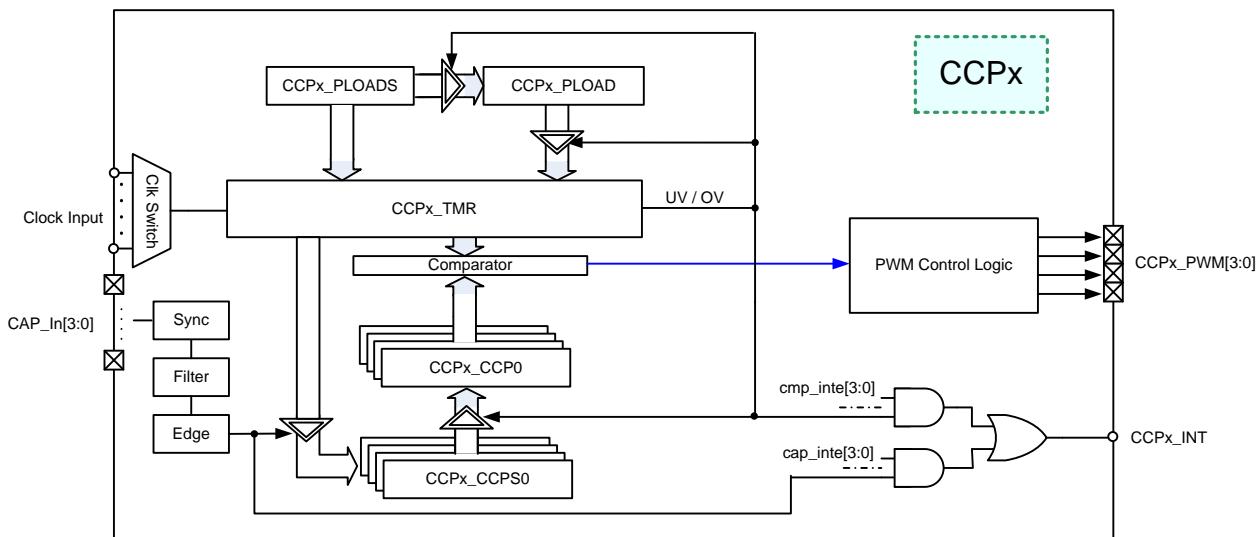


Fig.8.2 CCPx Structure

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.

Compare Mode:

When CCPx_TMR_CTRL.TMR_MODE set 'b00, CCPx will go Compare Mode. CCPx_TMR Compare which CCPx_CCP determined by CCPx_TMR_CTRL.CMP_E.

Counter_up:

If the CCPx_TMR_CTRL.TMR_UDC_SEL set 1, CCPx_TMR will do counter up function. When CCPx_TMR_CTRL.TMR_STR be set, CCPx_TMR will load zero value and start counter up. It will generate a overflow interrupt when CCPx_TMR value equal CCPx_TMR_PLOAD. When the overflow event happens, CCPx_TMR will load zero value again and restart counter up.

Counter_down:

If the CCPx_TMR_CTRL.TMR_UDC_SEL set 0, CCPx_TMR will do counter down function. When CCPx_TMR_CTRL.TMR_STR be set, CCPx_TMR will load CCPx_TMR_PLOAD value and start counter down. It will generate an overflow interrupt when CCPx_TMR value equal zero. When the overflow event happens, CCPx_TMR will load CCPx_TMR_PLOAD value again and restart counter down.

Capture Mode:

When CCPx_TMR_CTRL.TMR_MODE set 'b01, CCPx will go Capture Mode. The CCPx Support 4 Capture Source(CAPin0~3). The CAPin IO map as follows:

Capture in IO Selection	CCP0		CCP1
	P_GPIO_FUN_CTR.ccp0_io_sel=0	P_GPIO_FUN_CTR.ccp0_io_sel=1	
CAPin0	IOA0[0]	IOA0 [13]	IOA0[18]
CAPin1	IOA0 [1]	IOA0 [14]	IOA0[19]
CAPin2	IOA0 [2]	IOA0 [15]	IOA0[20]
CAPin3	IOA0 [3]	IOA0 [16]	IOA0[21]

Table8.1 CAPinx IO Map

Each CAPin IO all support debounce Function to avoid IO bounce. We can choose CAPin IO rise edge, fall edge or both edge happen to trigle capture function by CCPx_CAP_CTRL. CAPx_EDGE. For differ Capin IO trigle capture function, the CCPx_TMR value will be stored to different CCPx_CCPRx as table 8.2 and will generate a capture interrupt.

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

Table8.1 CAPinx map

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

PWM Mode:

When CCPx_TMR_CTRL.TMR_MODE set 'b02, CCPx will go PWM Mode. Each CCP moduld all support 4x PWMIO. Each PWMIO can selection Independent or Complementary mode by CCPx_PWM_CTRL. PWMypole. In Complementary mode, PWM1 Out = ~PWM0 Out and PWM3 Out = ~PWM2 Out. Whwn PWM Start, CCPx_TMR Function is the same as Counter/Compare Mode counter down. It will load CCPx_TMR_PLOAD value and start counter down and when CCPx_TMR at zero, it will reload CCPx_TMR_PLOAD value. When CCPx_TMR > CCPRx , The PWMx Output 0 else The PWMx Output 1. The function is description Fig 8.3. In complemantary mode, PWM0/PWM2 will start counter dead time(DTIME) at PWM_reg rising edge but PWM0/PWM2 will start counter dead time(DTIME) at PWM_reg falling edge

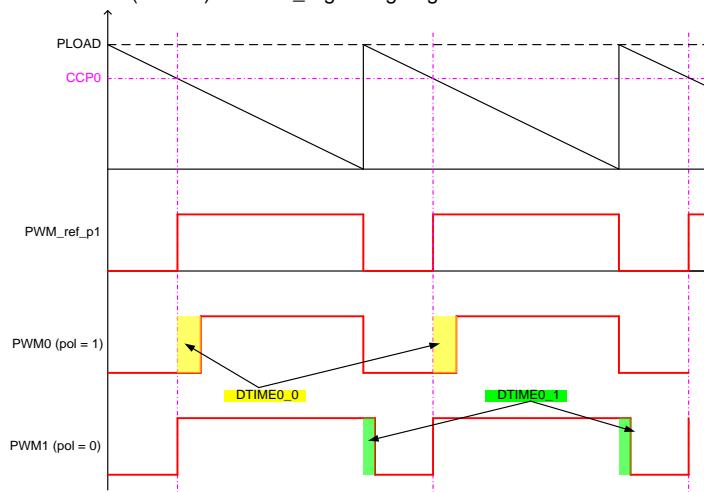
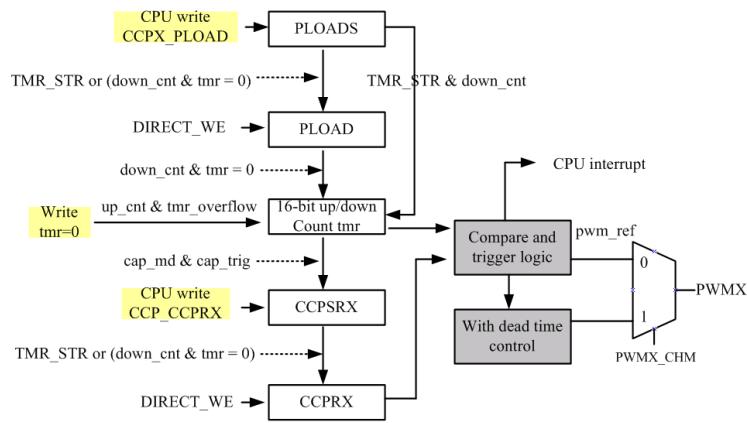


Fig 8.3 PWM Output

Control Flow:


Flow Diagram of CCP

9.5. Register Description

Registers Map

Base Address : 0x5004_0000				
Name	Description	Address	Access	Reset value
CCP0_TMCMP_CTRL	CCP0 Timer Control Register	0x5004_0000	R/W	0x0000_0000
CCP0_CAP_CTRL	CCP0 Capture Mode Control Register	0x5004_0004	R/W	0x0000_0000
CCP0_PWM_CTRL	CCP0 PWM Mode Control Register	0x5004_0008	R/W	0x0000_0000
CCP0_INTEN	CCP0 Capture Mode Interrupt Enable Register	0x5004_0010	R/W	0x0000_0000
CCP0_INSTS	CCP0 Status Register	0x5004_0014	R/W	0x0000_0000
CCP0_TMR_COUNT	CCP0 Timer Actual Value Register	0x5004_0018	R/W	0x0000_0000
CCP0_PLOAD	CCP0 Pre-Load Data Register	0x5004_001C	R/W	0x0000_0000
CCP0_CCPR0	CCP0 Compare/Capture/PWM duty Register 0	0x5004_0020	R/W	0x0000_0000
CCP0_CCPR1	CCP0 Compare/Capture/PWM duty Register 1	0x5004_0024	R/W	0x0000_0000
CCP0_CCPR2	CCP0 Compare/Capture/PWM duty Register 2	0x5004_0028	R/W	0x0000_0000
CCP0_CCPR3	CCP0 Compare/Capture/PWM duty Register 3	0x5004_002C	R/W	0x0000_0000
CCP0_PWM_DTIME	CCP0 PWM Delay Time Register	0x5004_0030	R/W	0x0000_0000
CCP1_TMCMP_CTRL	CCP1 Timer Control Register	0x5004_1000	R/W	0x0000_0000
CCP1_CAP_CTRL	CCP1 Capture Mode Control Register	0x5004_1004	R/W	0x0000_0000
CCP1_PWM_CTRL	CCP1 PWM Mode Control Register	0x5004_1008	R/W	0x0000_0000
CCP1_INTEN	CCP1 Capture Mode Interrupt Enable Register	0x5004_1010	R/W	0x0000_0000
CCP1_INSTS	CCP1 Status Register	0x5004_1014	R/W	0x0000_0000
CCP1_TMR_COUNT	CCP1 Timer Actual Value Register	0x5004_1018	R/W	0x0000_0000
CCP1_PLOAD	CCP1 Pre-Load Data Register	0x5004_101C	R/W	0x0000_0000
CCP1_CCPR0	CCP1 Compare/Capture/PWM duty Register 0	0x5004_1020	R/W	0x0000_0000
CCP1_CCPR1	CCP1 Compare/Capture/PWM duty Register 1	0x5004_1024	R/W	0x0000_0000
CCP1_CCPR2	CCP1 Compare/Capture/PWM duty Register 2	0x5004_1028	R/W	0x0000_0000
CCP1_CCPR3	CCP1 Compare/Capture/PWM duty Register 3	0x5004_102C	R/W	0x0000_0000
CCP1_PWM_DTIME	CCP1 PWM Delay Time Register	0x5004_1030	R/W	0x0000_0000

Registers Function
CCP0_TMCMP_CTRL CCP0 Timer Control Register
Address : 0x5004_0000
CCP1_TMCMP_CTRL CCP1 Timer Control Register
Address : 0x5004_1000

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CMP3_EN	CMP2_EN	CMP1_EN	CMP0_EN	CH3_CTL_SEL	CH2_CTL_SEL	--	TMR_UDC_SE
7	6	5	4	3	2	1	0
TMR_CLK_SEL				TMR_MODE		DIRECT_WE	TMR_STR

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0
[15]	CMP3_ENABLE (CMP3_EN)	CCPx compare mode channel 3 enable bits 0: CCPx channel 3 is disabled 1: CCPx channel 3 is enabled Note: The dead time function will be disabled automatically when timer operating in compare mode.	R/W	0x0
[14]	CMP2_ENABLE (CMP2_EN)	CCPx compare mode channel 2 enable bits 0: CCPx channel 2 is disabled 1: CCPx channel 2 is enabled Note: The dead time function will be disabled automatically when timer operating in compare mode.	R/W	0x0
[13]	CMP1_ENABLE (CMP1_EN)	CCPx compare mode channel 1 enable bits 0: CCPx channel 1 is disabled 1: CCPx channel 1 is enabled Note: The dead time function will be disabled automatically when timer operating in compare mode.	R/W	0x0
[12]	CMP0_ENABLE (CMP0_EN)	CCPx compare mode channel 0 enable bits 0: CCPx channel 0 is disabled 1: CCPx channel 0 is enabled Note: The dead time function will be disabled automatically when timer operating in compare mode.	R/W	0x0
[11]	CCP1_PWM3_TIMER_C TRL (CH3_CTL_SEL)	CCP1 channel 3 control selection (Only For CCP1) 0: CCP1 channel 3 is controlled by CCP1 1: CCP1 channel 3 is controlled by CCP0 When the bit set all CCP1 Channel3 control signal will referenc CCP0.	R/W	0x0
[10]	CCP1_PWM2_TIMER_C TRL (CH2_CTL_SEL)	CCP1 channel 2 control selection(Only For CCP1) 0: CCP1 channel 2 is controlled by CCP1 1: CCP1 channel 2 is controlled by CCP0 When the bit set all CCP1 Channel2 control signal will referenc CCP0.	R/W	0x0
[9]	--	Reserved	R	0
[8]	TMR_UDC_SEL	Timer up / down count selection bit 1: up count, 0 : down count	R/W	0x0
[7:4]	TMR_CLK_SEL	Timer input clock source selection bits	R/W	0x0

Bit	Name	Description	Access	Reset value																																		
		<table border="1"> <tr><td>TMR_CLK_SEL[3:0]</td><td>Clock Source</td></tr> <tr><td>0000</td><td>System clock</td></tr> <tr><td>0001</td><td>System clock / 2</td></tr> <tr><td>0010</td><td>System clock / 4</td></tr> <tr><td>0011</td><td>System clock / 8</td></tr> <tr><td>0100</td><td>System clock / 16</td></tr> <tr><td>0101</td><td>System clock / 64</td></tr> <tr><td>0110</td><td>System clock / 256</td></tr> <tr><td>0111</td><td>System clock / 1024</td></tr> <tr><td>1000</td><td>System clock / 2048</td></tr> <tr><td>1001</td><td>System clock / 4096</td></tr> <tr><td>1010</td><td>C32K</td></tr> <tr><td>1011</td><td>EXT1</td></tr> <tr><td>1100</td><td>EXT2</td></tr> <tr><td>1101</td><td>TMA_OV</td></tr> <tr><td>1110</td><td>TMB_OV</td></tr> <tr><td>1111</td><td>TMC_OV</td></tr> </table>	TMR_CLK_SEL[3:0]	Clock Source	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	C32K	1011	EXT1	1100	EXT2	1101	TMA_OV	1110	TMB_OV	1111	TMC_OV		
TMR_CLK_SEL[3:0]	Clock Source																																					
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	C32K																																					
1011	EXT1																																					
1100	EXT2																																					
1101	TMA_OV																																					
1110	TMB_OV																																					
1111	TMC_OV																																					
[3:2]	TMR_MODE_SEL (TMR_MODE)	<p>Timer operating mode selection bits</p> <table border="1"> <tr><td>TMR_MODE[1:0]</td><td>Timer Mode</td></tr> <tr><td>00</td><td>Counter / Compare Mode</td></tr> <tr><td>01</td><td>Capture Mode</td></tr> <tr><td>10</td><td>PWM Mode</td></tr> <tr><td>11</td><td>Reserved</td></tr> </table>	TMR_MODE[1:0]	Timer Mode	00	Counter / Compare Mode	01	Capture Mode	10	PWM Mode	11	Reserved	R/W	0x0																								
TMR_MODE[1:0]	Timer Mode																																					
00	Counter / Compare Mode																																					
01	Capture Mode																																					
10	PWM Mode																																					
11	Reserved																																					
[1]	DIRECT_WR_ENABLE (DIRECT_WE)	<p>Data direct write enable</p> <p>0: The CCPx_PLOAD.PLOAD / CCPx_CCPRx.CCPRx will be updated while timer is over/under flow occurred.</p> <p>1: The CCPx_PLOAD.PLOAD / CCPx_CCPRx.CCPRx will be updated immediately while CPU write data into CCPx_PLOAD.PLOAD / CCPx_CCPRx.CCPRx</p>	R/W	0x0																																		
[0]	TMR_ENABLE (TMR_STR)	<p>Timer start bit</p> <p>0 = disabled</p> <p>1 = enabled</p>	R/W	0x0																																		

CCP0_CAP_CTRL CCP0 Capture Mode Control Register
Address : 0x5004_0004
CCP1_CAP_CTRL CCP1 Capture Mode Control Register
0x5004_1004

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--	CAP_LPF_SEL			CAP3_EDGE		CAP2_EDGE	
7	6	5	4	3	2	1	0
CAP1_EDGE		CAP0_EDGE		CAP_E[3:0]			

Bit	Name	Description	Access	Reset value
[31:15]	--	Reserved	R	0
[14:12]	CAP_LPF_SEL	Low pass filter selection bits	R/W	0x0
		TMR_LPF_SEL[1:0] Filter Cycles		
		000 LPF is disable		
		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	Capture channel 3 sample edge selection bits	R/W	0x0
		CAP3_EDGE Capture Edge		
		00 NO Action		
		01 Rising Edge		
		10 Falling Edge		
		11 Both Edge		
[9:8]	CAP2_EDGE	Capture channel 2 sample edge selection bits	R/W	0x0
		CAP2_EDGE Capture Edge		
		00 NO Action		
		01 Rising Edge		
		10 Falling Edge		
		11 Both Edge		
[7:6]	CAP1_EDGE	Capture channel 1 sample edge selection bits	R/W	0x0
		CAP1_EDGE Capture Edge		
		00 NO Action		
		01 Rising Edge		
		10 Falling Edge		
		11 Both Edge		
[5:4]	CAP0_EDGE	Capture channel 0 sample edge selection bits	R/W	0x0
		CAP0_EDGE Capture Edge		
		00 NO Action		
		01 Rising Edge		
		10 Falling Edge		
		11 Both Edge		
[3:0]	CAP_E	Capture mode channel 3~0 enable bits. 0 = disabled 1 = enabled	R/W	0x00

Bit	Name	Description			Access	Reset value																												
		For CCP0 : Capture IO: Ccp0_io_sel = gpio_fun_ctrl[5](0x5102_0200[5]) <table border="1" style="margin-left: 20px;"> <tr> <td>CAP_E</td> <td colspan="2">I/O Mapping</td> </tr> <tr> <td></td> <td>ccp0_io_sel=1</td> <td>ccp0_io_sel=0</td> </tr> <tr> <td>0001 (CAP0_ENABLE)</td> <td>IOA0 [13]</td> <td>IOA0 [0]</td> </tr> <tr> <td>0010 (CAP1_ENABLE)</td> <td>IOA0 [14]</td> <td>IOA0 [1]</td> </tr> <tr> <td>0100 (CAP2_ENABLE)</td> <td>IOA0 [15]</td> <td>IOA0 [2]</td> </tr> <tr> <td>1000 (CAP3_ENABLE)</td> <td>IOA0 [16]</td> <td>IOA0 [3]</td> </tr> </table> For CCP1 : Capture IO MAP <table border="1" style="margin-left: 20px;"> <tr> <td>CAP_E</td> <td>I/O Mapping</td> </tr> <tr> <td>0001 (CAP0_ENABLE)</td> <td>IOA0[18]</td> </tr> <tr> <td>0010 (CAP1_ENABLE)</td> <td>IOA0 [19]</td> </tr> <tr> <td>0100 (CAP2_ENABLE)</td> <td>IOA0 [20]</td> </tr> <tr> <td>1000 (CAP3_ENABLE)</td> <td>IOA0 [21]</td> </tr> </table> Note: All of the trigger sources are from external I/O. Above table is the trigger source mapping table.			CAP_E	I/O Mapping			ccp0_io_sel=1	ccp0_io_sel=0	0001 (CAP0_ENABLE)	IOA0 [13]	IOA0 [0]	0010 (CAP1_ENABLE)	IOA0 [14]	IOA0 [1]	0100 (CAP2_ENABLE)	IOA0 [15]	IOA0 [2]	1000 (CAP3_ENABLE)	IOA0 [16]	IOA0 [3]	CAP_E	I/O Mapping	0001 (CAP0_ENABLE)	IOA0[18]	0010 (CAP1_ENABLE)	IOA0 [19]	0100 (CAP2_ENABLE)	IOA0 [20]	1000 (CAP3_ENABLE)	IOA0 [21]		
CAP_E	I/O Mapping																																	
	ccp0_io_sel=1	ccp0_io_sel=0																																
0001 (CAP0_ENABLE)	IOA0 [13]	IOA0 [0]																																
0010 (CAP1_ENABLE)	IOA0 [14]	IOA0 [1]																																
0100 (CAP2_ENABLE)	IOA0 [15]	IOA0 [2]																																
1000 (CAP3_ENABLE)	IOA0 [16]	IOA0 [3]																																
CAP_E	I/O Mapping																																	
0001 (CAP0_ENABLE)	IOA0[18]																																	
0010 (CAP1_ENABLE)	IOA0 [19]																																	
0100 (CAP2_ENABLE)	IOA0 [20]																																	
1000 (CAP3_ENABLE)	IOA0 [21]																																	

CCP0_PWM_CTRL CCP0 PWM Control Register
Address : 0x5004_0008
CCP1_PWM_CTRL CCP1 PWM Control Register
0x5004_1008

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--						PWM23_CHM	PWM01_CHM
7	6	5	4	3	2	1	0
PWM3_POL	PWM2_POL	PWM1_POL	PWM0_POL	PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN

Bit	Name	Description	Access	Reset value
[31:10]	--	Reserved	R/W	0x00
[9]	PWM23_CHM	PWM channel 2/3 operationg mode selection bit 0 : independent mode 1 : complementary mode	R/W	0x0
[8]	PWM01_CHM	PWM channel 0/1 operationg mode selection bit 0 : independent mode 1 : complementary mode	R/W	0x0
[7]	PWM3_POL	PWM channel 3 polarity control bit Independent mode : 0: PWM3 output low while CCPX_TMR.TMR < CCPX_CCPR3.CCPR3 1: PWM3 output high while CCPX_TMR.TMR < CCPX_CCPR3.CCPR3 Complementary mode : 0: PWM3 output high while CCPX_TMR.TMR < CCPX_CCPR2.CCPR2, else output low 1: PWM3 output low while CCPX_TMR.TMR < CCPX_CCPR2.CCPR2,	R/W	0x0

Bit	Name	Description	Access	Reset value
		else output high		
[6]	PWM2_POL	<p>PWM channel 2 polarity control bit</p> <p>Independent mode :</p> <p>0: PWM2 output low while CCPX_TMR.TMR < CCPX_CCPR2.CCPR2</p> <p>1: PWM2 output high while CCPX_TMR.TMR < CCPX_CCPR2.CCPR2</p> <p>Complementary mode :</p> <p>0: PWM2 output low while CCPX_TMR.TMR < CCPX_CCPR2.CCPR2, else output high</p> <p>1: PWM2 output high while CCPX_TMR.TMR < CCPX_CCPR2.CCPR2, else output low</p>	R/W	0x0
[5]	PWM1_POL	<p>PWM channel 1 polarity control bit</p> <p>Independent mode :</p> <p>0: PWM1 output low while CCPX_TMR.TMR < CCPX_CCPR1.CCPR1</p> <p>1: PWM1 output high while CCPX_TMR.TMR < CCPX_CCPR1.CCPR1</p> <p>Complementary mode :</p> <p>0: PWM1 output high while CCPX_TMR.TMR < CCPX_CCPR0.CCPR0, else output low</p> <p>1: PWM1 output low while CCPX_TMR.TMR < CCPX_CCPR0.CCPR0, else output high</p>	R/W	0x0
[4]	PWM0_POL	<p>PWM channel 0 polarity control bit</p> <p>Independent mode :</p> <p>0: PWM0 output low while CCPX_TMR.TMR < CCPX_CCPR0.CCPR0</p> <p>1: PWM0 output high while CCPX_TMR.TMR < CCPX_CCPR0.CCPR0</p> <p>Complementary mode :</p> <p>0: PWM0 output low while CCPX_TMR.TMR < CCPX_CCPR0.CCPR0, else output high</p> <p>1: PWM0 output high while CCPX_TMR.TMR < CCPX_CCPR0.CCPR0, else output low</p>	R/W	0x0
[3]	PWM3_ENABLE (PWM3_EN)	PWM channel 3 enable 0 = Disabled 1 = Enabled	R/W	0x0
[2]	PWM2_ENABLE (PWM2_EN)	PWM channel 2 enable 0 = Disabled 1 = Enabled	R/W	0x0
[1]	PWM1_ENABLE (PWM1_EN)	PWM channel 1 enable 0 = Disabled 1 = Enabled	R/W	0x0
[0]	PWM0_ENABLE (PWM0_EN)	PWM channel 0 enable 0 = Disabled 1 = Enabled	R/W	0x0

CCP0_CMP_INTEN	CCP0 Interrupt Enable Register							Address : 0x5004_0010
CCP1_CMP_INTEN	CCP1 Interrupt Enable Register							0x5004_1010
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	TMR_INTEN
CAP3_INTE	CAP2_INTE	CAP1_INTE	CAP0_INTE	CMP3_INTE	CMP2_INTE	CMP1_INTE	CMP0_INTE	

Bit	Name	Description	Access	Reset value
[31:9]	--	Reserve	R	0
[8]	TMR_INT_ENABLE (TMR_INTE)	Timer interrupts enable bit 0 = disabled 1 = enabled Note: This bit is used to gate interrupt signal that indicates counter is overflow, underflow or greater than CCPx_PLOAD.PLOAD value.	R/W	0x0
[7]	CAP3_INT_ENABLE (CAP3_INTE)	Capture mode channel 3 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[6]	CAP2_INT_ENABLE (CAP2_INTE)	Capture mode channel 2 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[5]	CAP1_INT_ENABLE (CAP1_INTE)	Capture mode channel 1 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[4]	CAP0_INT_ENABLE (CAP0_INTE)	Capture mode channel 0 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[3]	CMP3_INT_ENABLE (CMP3_INTE)	Compare mode channel 3 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[2]	CMP2_INT_ENABLE (CMP2_INTE)	Compare mode channel 2 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[1]	CMP1_INT_ENABLE (CMP1_INTE)	Compare mode channel 1 interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[0]	CMP0_INT_ENABLE (CMP0_INTE)	Compare mode channel 0 interrupt enable 0 = disabled 1 = enabled	R/W	0x0

CCP0_INTSTS	CCP0 Interrupt Status Register							Address : 0x5004_0014
CCP1_INTSTS	CCP1 Interrupt Status Register							0x5004_1014
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	TMR_INTF
CAP3_INTF	CAP2_INTF	CAP1_INTF	CAP0_INTF	CMP3_INTF	CMP2_INTF	CMP1_INTF	CMP0_INTF	

Bit	Name	Description	Access	Reset value
[31:9]	--	Reserved	R/W	0x00
[8]	TMR_INT_FLAG (TMR_INTF)	<p>Timer overflow/underflow interrupts flag</p> <p>read :</p> <p>0 : idle / busy</p> <p>1 : timer interrupt triggered</p> <p>write :</p> <p>0 : no effect</p> <p>1 : clear these bits</p> <p>Note: This bit will be triggered when counter is overflow, underflow or greater than CCPX_PLOAD.PLOAD value.</p>	R/W	0x0
[7]	CAP3_INT_FLAG (CAP3_INTF)	<p>Capture mode channel 3 interrupt flag.</p> <p>read :</p> <p>0 : no capture event triggered</p> <p>1 : capture event triggered</p> <p>write :</p> <p>0 : no effect</p> <p>1 : clear these bits</p> <p>Note: This bit is available when timer does not operate in concatenated mode.</p>	R/W	0x0
[6]	CAP2_INT_FLAG (CAP2_INTF)	<p>Capture mode channel 2 interrupt flag.</p> <p>read :</p> <p>0 : no capture event triggered</p> <p>1 : capture event triggered</p> <p>write :</p> <p>0 : no effect</p> <p>1 : clear these bits</p> <p>Note: This bit is available when timer does not operate in concatenated mode.</p>	R/W	0x0
[5]	CAP1_INT_FLAG (CAP1_INTF)	<p>Capture mode channel 1 interrupt flag.</p> <p>read :</p> <p>0 : no capture event triggered</p> <p>1 : capture event triggered</p> <p>write :</p> <p>0 : no effect</p>	R/W	0x0

Bit	Name	Description	Access	Reset value
		1 : clear these bits Note: This bit is available when timer does not operate in concatenated mode.		
[4]	CAP0_INT_FLAG (CAP0_INTF)	Capture mode channel 0 interrupt flag. read : 0 : no capture event triggered 1 : capture event triggered write : 0 : no effect 1 : clear these bits Note: This bit is available when timer does not operate in concatenated mode.	R/W	0x0
[3]	CMP3_INT_FLAG (CMP3_INTF)	Compare mode channel 3 interrupt flag. read : 0 : no counting equivalent triggered 1 : counting equivalent triggered write : 0 : no effect 1 : clear these bits Note: This bit is available when timer does not operate in concatenated mode.	R/W	0x0
[2]	CMP2_INT_FLAG (CMP2_INTF)	Compare mode channel 2 interrupt flag. read : 0 : no counting equivalent triggered 1 : counting equivalent triggered write : 0 : no effect 1 : clear these bits Note: This bit is available when timer does not operate in concatenated mode.	R/W	0x0
[1]	CMP1_INT_FLAG (CMP1_INTF)	Compare mode channel 1 interrupt flag. read : 0 : no counting equivalent triggered 1 : counting equivalent triggered write : 0 : no effect 1 : clear these bits Note: This bit is available when timer does not operate in concatenated mode.	R/W	0x0
[0]	CMP0_INT_FLAG (CMP0_INTF)	Compare mode channel 0 interrupt flag. read : 0 : no counting equivalent triggered 1 : counting equivalent triggered write : 0 : no effect 1 : clear these bits	R/W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description						Access	Reset value
		Note: This bit is available when timer does not operate in concatenated mode.							

CCP0_TMR_COUNT CCP0 Timer Actual Value Register Address : 0x5004_0018
CCP1_TMR_COUNT CCP1 Timer Actual Value Register 0x5004_1018

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

Bit	Name	Description						Access	Reset value
[31:16]	--	Reserve						R	0x0000
[15:0]	TMR	Timer actual value.						R	0x0000

CCP0_PLOAD CCP0 Pre-Load Data Register Address : 0x5004_001C
CCP1_PLOAD CCP1 Pre-Load Data Register 0x5004_101C

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

Bit	Name	Description						Access	Reset value
[31:16]	--	Reserve						R	0
[15:0]	TMR_PLOAD	Timer pre-load data or PWM period data register Note: CCPX_PLOAD.PLOAD will be filled in CCPX_TMR.TMR when timer is underflow and CCPX_TMR_CTRL.TMR_UDC_SEL is set to 1.						R/W	0x0000

CCP0_CCPR0 CCP0 Compare/Capture/PWM Duty Register 0 Address : 0x5004_0020
CCP1_CCPR0 CCP1 Compare/Capture/PWM Duty Register 0 0x5004_1020

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CCPR0[15:8]							

7	6	5	4	3	2	1	0
CCPR0[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0x0000
[15:0]	CCPR0	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.	R/W	0x0000

CCP0_CCPR1 CCP0 Compare/Capture/PWM Duty Register 1							
CCP1_CCPR1 CCP1 Compare/Capture/PWM Duty Register 1							
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CCPR1[15:8]							
7	6	5	4	3	2	1	0
CCPR1[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0x0000
[15:0]	CCPR1	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.	R/W	0x0000

CCP0_CCPR2 CCP0 Compare/Capture/PWM Duty Register 2							
CCP1_CCPR2 CCP1 Compare/Capture/PWM Duty Register 2							
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CCPR2[15:8]							
7	6	5	4	3	2	1	0
CCPR2[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0x0000
[15:0]	CCPR2	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	R/W	0x0000

Bit	Name	Description						Access	Reset value
		are used to latch CCPX_TMR.TMR data.							

CCP0_CCPR3	CCP0 Compare/Capture/PWM Duty Register 3	Address : 0x5004_002C							
CCP1_CCPR3	CCP1 Compare/Capture/PWM Duty Register 3	0x5004_102C							
31	30	29	28	27	26	25	24		
--	--	--	--	--	--	--	--		
23	22	21	20	19	18	17	16		
--	--	--	--	--	--	--	--		
15	14	13	12	11	10	9	8		
CCPR3[15:8]									
7	6	5	4	3	2	1	0		
CCPR3[7:0]									

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0x0000
[15:0]	CCPR3	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.	R/W	0x0000

CCP0_PWM_DTIME	CCP0 PWM Delay Time Register	Address : 0x5004_0030							
31	30	29	28	27	26	25	24		
--	--	--	--	--	--	--	--		
23	22	21	20	19	18	17	16		
--	--	--	--	--	--	--	--		
15	14	13	12	11	10	9	8		
PWM_DTIME_1									
7	6	5	4	3	2	1	0		
PWM_DTIME_0									

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0x0000
[15:8]	PWM_DTIME_1	PWM delay time register for PWM 1 & PWM3 channel These bits are available when PWM operating in complementary mode. Complementary Mode: In same CCP, it allows setting the signals in PWM0 & PWM1 or PWM2 & PWM3 to be complementary or independent from each other. CCP0's PWM0 & PWM2 are master pins, and PWM1 & PWM3 are slave pins. [15:8]: settings of the delay time at slave pin's (PWM1 & 3) lower edge. Note: The time unit of DTIME is relevant to Pload's value. For example, suppose PLOAD is 63(64-1=0~63, total of 64-level). Thus, the time unit of DTIME is 1/64.	R/W	0x00

Bit	Name	Description	Access	Reset value
[7:0]	PWM_DTIME_0	<p>PWM delay time register for PWM0 & PWM2 channel These bits are available when PWM operating in complementary mode. [7:0]: settings of the delay time at master pin's (PWM0 & 2) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, suppose PLOAD is 63(64-1=0~63, total of 64-level). Thus, the time unit of DTIME is 1/64.</p>	R/W	0x00

CCP1_PWM_DTIME CCP1 PWM Dead-Time Register								Address : 0x5004_1030
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
PWM_DTIME_1								
7	6	5	4	3	2	1	0	
PWM_DTIME_0								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserve	R	0x0000
[15:8]	PWM_DTIME_1	<p>PWM delay time register for PWM 1 & PWM3 channel These bits are available when PWM operating in complementary mode. Complementary Mode: In same CCP, it allows settings the signals in PWM0 & PWM1 or PWM2 & PWM3 to be complementary or independent from each other. CCP1's PWM0 & PWM2 are master pins, and PWM1 & PWM3 are slave pins. [15:8]: settings of the delay time at slave pin's (PWM1 & 3) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, suppose PLOAD is 63(64-1=0~63, total of 64-level). Thus, the time unit of DTIME is 1/64.</p>	R/W	0x00
[7:0]	PWM_DTIME_0	<p>PWM delay time register for PWM0 & PWM2 channel These bits are available when PWM operating in complementary mode. [7:0]: settings of the delay time at master pin's (PWM0 & 2) lower edge.</p> <p>Note: The time unit of DTIME is relevant to Pload's value. For example, suppose PLOAD is 63(64-1=0~63, total of 64-level). Thus, the time unit of DTIME is 1/64.</p>	R/W	0x00

10. Embedded Memory FLASH Controller

10.1. Introduction

The GPCM1F SERIALS has an embedded user-programmable non-volatile memory (NVM), also called E-FLASH or FLASH, for storage of user code and data. E-FLASH is GPCM1F SERIALS Program Memory. Program Memory is where user application code stored.

10.2. Feature

- Reading by word(4 Byte)
- Erasing by page (1K Bytes)
- Embedded (64KB-64B) Program Memory
- Supports configurable user option for chip initialization
- Supports In-Application-Programming (IAP) to update embedded FLASH memory

10.3. Block Diagram

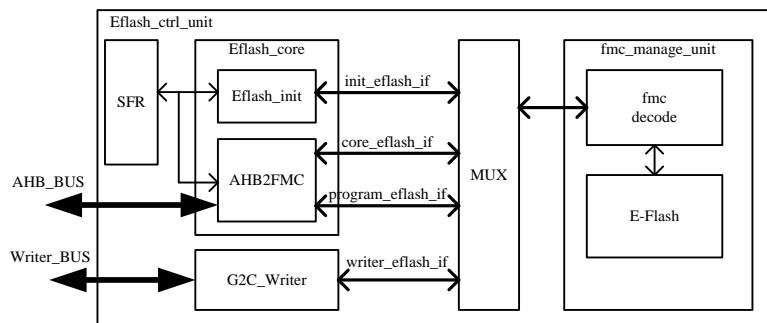


Fig9.1 Block Diagram

10.4. Function

10.4.1. Device

EFLASH controller main functions are program and erases EFLASH. User achieves the goal by using In-System-Programming (ISP), In-Application-Programming (IAP) device.

In System Programming(ISP)

In-System Programming (ISP) is a process whereby a blank device mounted to a circuit board can be programmed with the end-user code without the need to remove the device from the circuit board. Also, a previously programmed device can be erased and reprogrammed without removal from the circuit board. In order to perform ISP operations, the GPCM1F SERIALS is powered up in a special "SWD mode". SWD mode allows GPCM1F SERIALS to communicate with an external host device through the serial port, such as a PC or terminal. The GPCM1F SERIALS receives commands and data from the host to control the FLASH control register, base address is 0x5001_0000, to erase and reprogram code memory, etc. Once the ISP operations have been completed, the device is reconfigured so that it will operate normally next time it is either reset or power removed and reapplied.

In Application Programming (IAP)

Some applications may have a need to be able to erase and program code memory under the control for the application. For example, an application may have a need to store calibration information or perhaps need to be able to download new code portions. The ability to erase and program code memory in the end-user application is "In-Application Programming" (IAP).

10.4.2. Memory Organization

For GPCM1F Series, E-FLASH memory organization consists of the Program memory and User option memory. The map address and organization as follows figure 9.2.

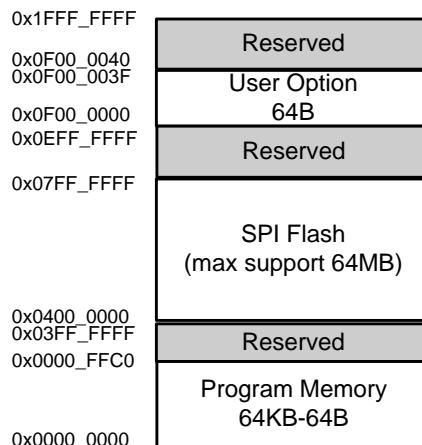


Fig.9.2 Memory Organization

Program Memory (PM)

Program Memory is where user application code stored and the size is (64KB-64B). The base address is from 0x0000_0000. After reset, the CPU M0 all will start boot form Address 0x0000_0000.

10.4.3. E-FLASH Erase and Program

The FLASH Program and Erase Controller block handles the program and erase operations of the embedded user-programmable non-volatile memory (FLASH). An ongoing E-FLASH operation will not block the CPU as long as the CPU does not access the FLASH memory. Two types of keycode need to set for start to unlock E-FLASH/UPT program and erase function.

Unlock E-FLASH

After reset, E-FLASH block is protected (FLASH_LOCK=1). The FLASH_CTRL register is not accessible in write mode. An unlocking sequence should be written to the FLASH_FSHKEY register to open up the block. This sequence consists of two write cycles, where two key values (KEY1 = 0xABCD5678 and KEY2 = 0x1234FEDC) are written to the FLASH_FSHKEY address. By this process, FLASH_LOCK will set 0. It will accept user to programing and erase E-FLASH. Any wrong sequence locks up the FLASH Ctrl block and FLASH_CTRL register. The FLASH Ctrl block and FLASH_CTRL register can be locked by the user's software by writing then FLASH_LOCK bit of the FLASH_CTRL register to 1.

Note: When using erase and program operations to Internal FLASH memory or switching to Wait state, Cache OFF must be issued first. Then, wait for all erase and program operations are finished before issuing Cache ON command. Otherwise, system may mistakenly trigger the WDT reset.

10.4.3.1. Program Memory Programming

The program operation starts when the CPU writes a word into a program memory address with the FLASH_TYPE of the FLASH_CTRL register set in Program. During the programming, FLASH_BUSY bit is set, the CPU will keep wait until the ongoing FLASH memory programming is completed.

To program memory, the procedure as below should be followed:

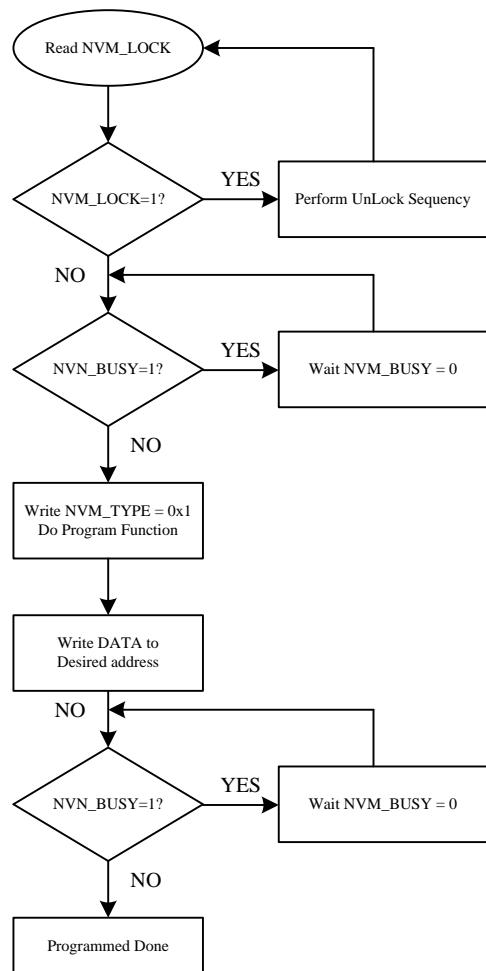


Fig.9.3 PM Programming Procedure

10.4.3.2. Programming Page Erase

A page of the E-FLASH can be erased using the Page Erase feature. To erase a page, the procedure as below should be followed. The page size of GPCM1F Serial is 1K Bytes.

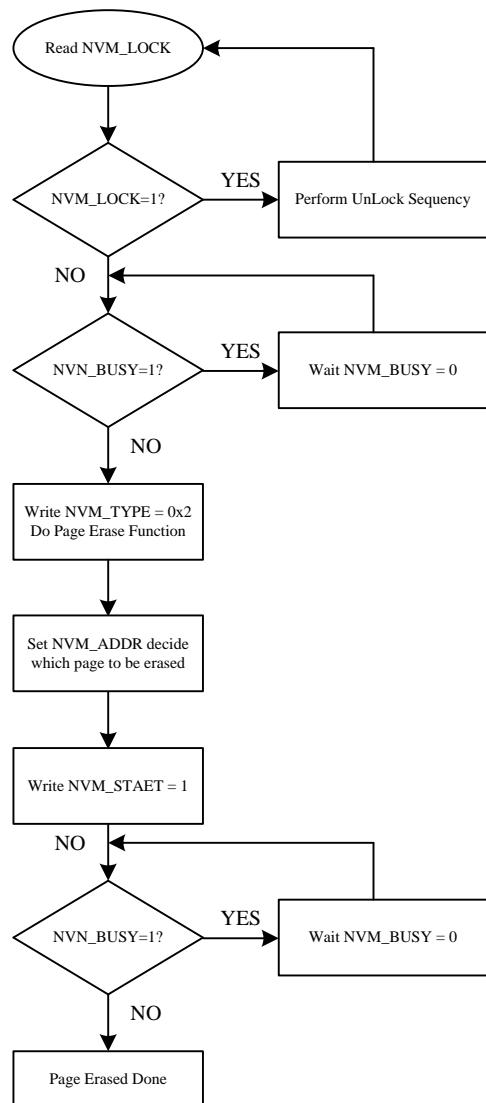


Fig.9.4 E-FLASH Erase Procedure

10.4.3.3. E-FLASH Chip Erase

E-FLASH Chip erased function will erase PM and UPT memory region, the procedure below should be followed:

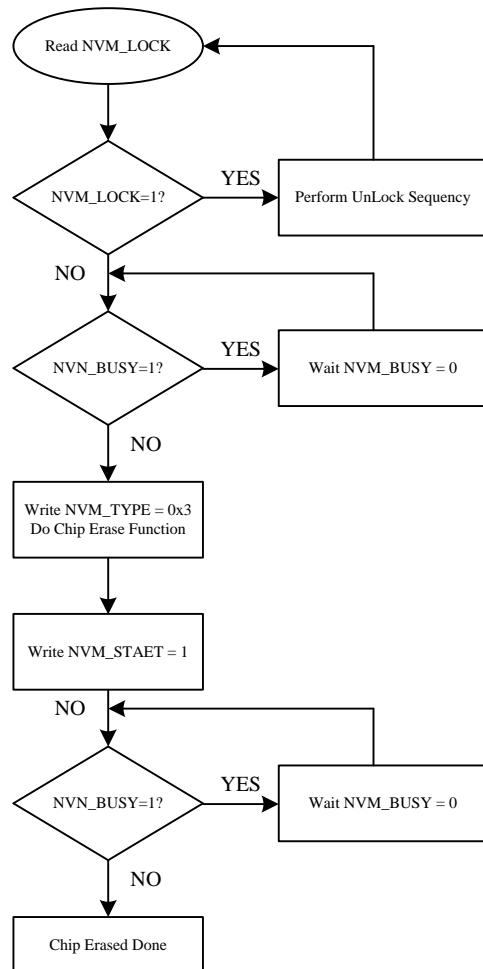


Fig.9.5 E-FLASH Chip Erase Procedure

10.5. Register Description

Registers Map

Base Address : 0x5001_0000				
Name	Description	Address	Access	Reset value
FLASH_CTRL_UNLOCK	FLASH Lock/ Unlock Control Register	0x5001_0000	R/W	0x0000_0000
FLASH_CTRL	FLASH Operation Control Register	0x5001_0008	R/W	0x0000_0001
FLASH_ERASE_ADDR	FLASH Erase Address Register	0x5001_000C	R/W	0xFFFF_FE00
FLASH_STS	FLASH Status Register	0x5001_0010	R	0x0000_0000
FLASH_OPTION0_STS	FLASH Operation 0 Status Register	0x5001_0014	R/W	0x0000_0004
FLASH_ERASE_CTRL	FLASH Erase Control Register	0x5001_0020	R/W	0x0000_0000

Registers Function

FLASH_CTRL_UNLOCK FLASH Lock/ Unlock Control Register Address : 0x5001_0000

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



GPCM1F SERIES PROGRAMMING GUIDE

FLASH_FSHKEY							
7	6	5	4	3	2	1	0
FLASH_FSHKEY							

Bit	Name	Description	Access	Reset value
[31:0]	FLASH_FSHKEY	FLASH Control Register Unlock Key Write 0xABCD5678 and 0x1234FEDC to FLASH_FSHKEY in sequence, the FLASH_CTRL registers will be unlocked; all bit fields can be accessed. Note: When write other words to FLASH_FSHKEY, the FLASH_CTRL register will be reset and locked.	R/W	0

FLASH_CTRL FLASH Control Register								Address : 0x5001_0008
31	30	29	28	27	26	25	24	
--	--	--	--	--	--	--	--	
23	22	21	20	19	18	17	16	
--	--	--	--	--	--	--	--	
15	14	13	12	11	10	9	8	
--	--	--	--	--	--	--	--	TYPE
7	6	5	4	3	2	1	0	
--	--	--	--	--	--	--	--	LOCK

Bit	Name	Description	Access	Reset value
[31:12]	--	Reserved	R	0
[11:8]	TYPE	FLASH Operation Type: 0000: Idle or Read 0001: Program 0010: Page Erase 0011: Chip Erase Others: Reserved Note1: FLASH_TYPE writes access is available on writing the correct key sequence to the FLASH_FSHKEY. Note2: FLASH Program Type only support 32-bit program, other data size will not affect occurs.	R/W	0
[7:1]	--	Reserved	R	0
[0]	LOCK	FLASH Lock / Unlock Status 0 = Indicate the FLASH_CTRL are unlocked. 1 = Indicate the FLASH_CTRL are locked. Write 1 ,will Lock Note: This bit is reset by hardware after detecting the unlock sequence of FLASH_FSHKEY.	R/W	1

Note. This FLASH_Ctrl is updated only when IC is running EFLASH initial Function. We can write when FLASH_FSHKEY pass.

FLASH_ERASE_ADDR FLASH Erase Address Register								Address : 0x5001_000C
31	30	29	28	27	26	25	24	
FLASH_ADDR								
23	22	21	20	19	18	17	16	

FLASH_ADDR							
15	14	13	12	11	10	9	8
FLASH_ADDR							
7	6	5	4	3	2	1	0
FLASH_ADDR							

Bit	Name	Description	Access	Reset value
[31:9]	FLASH_ADDR[31:9]	FLASH Address for FLASH Erase Chooses the address to a page to erase when Page Erase is selected Page size is 1K bytes. Note: Write access to this register is blocked when the FLASH_BUSY bit is set.	R/W	0x7FFF_FF
[8:0]	FLASH_ADDR[8:0]	FLASH Address for FLASH Erase Chooses the address to a page to erase when Page Erase is selected The FLASH_ADDR[8:0] always keeps in zero for page size 512Bytes. Note: Write access to this register is blocked when the FLASH_BUSY bit is set.	R	0x00

FLASH_STS FLASH Status Register								Address : 0x5001_0010
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
								FLASH_BUSY

Bit	Name	Description	Access	Reset value
[31:1]	--	Reserved	R	0
[0]	STS_FLAG	FLASH Operation Busy Flag This indicates that a FLASH operation is in progress. This is set on the beginning of a FLASH operation and reset when the operation finishes or when an error occurs.	R	0

FLASH_OPTION0_STS FLASH Operation 0 Status Register								Address : 0x5001_0014
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
								WAITSTATE

Bit	Name	Description				Access	Reset value																			
[31:3]	--	Reserved				R	0																			
		Latency Wait State Configuration When Cortex-M0 access FLASH, FLASH will response wait state numbers as :																								
		<table border="1"> <thead> <tr> <th>FLASH_LATEN CY</th> <th>Wait Cycle</th> <th>LATENCY</th> <th>Wait Cycle</th> </tr> </thead> <tbody> <tr> <td>3'b000</td> <td>1T</td> <td>3'b100</td> <td>4T</td> </tr> <tr> <td>3'b001</td> <td>1T</td> <td>3'b101</td> <td>5T</td> </tr> <tr> <td>3'b010</td> <td>2T</td> <td>3'b110</td> <td>6T</td> </tr> <tr> <td>3'b011</td> <td>3T</td> <td>3'b111</td> <td>7T</td> </tr> </tbody> </table>				FLASH_LATEN CY	Wait Cycle	LATENCY	Wait Cycle	3'b000	1T	3'b100	4T	3'b001	1T	3'b101	5T	3'b010	2T	3'b110	6T	3'b011	3T	3'b111	7T	
FLASH_LATEN CY	Wait Cycle	LATENCY	Wait Cycle																							
3'b000	1T	3'b100	4T																							
3'b001	1T	3'b101	5T																							
3'b010	2T	3'b110	6T																							
3'b011	3T	3'b111	7T																							
		Note: For GPCM1F SERIALS E-FLASH read cycle >= 40ns																								

FLASH_ERASE_CTRL FLASH Erase Control Register
Address : 0x5001_0020

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

Bit	Name	Description	Access	Reset value
[0]	ERASE_CTRL_NVM_ST ART (ERASE_START)	FLASH Operation Start. When Write 1 to this register bit 0, it will trigger the operation of page erase or chip erase.	R/W	0

11. MAC Controller

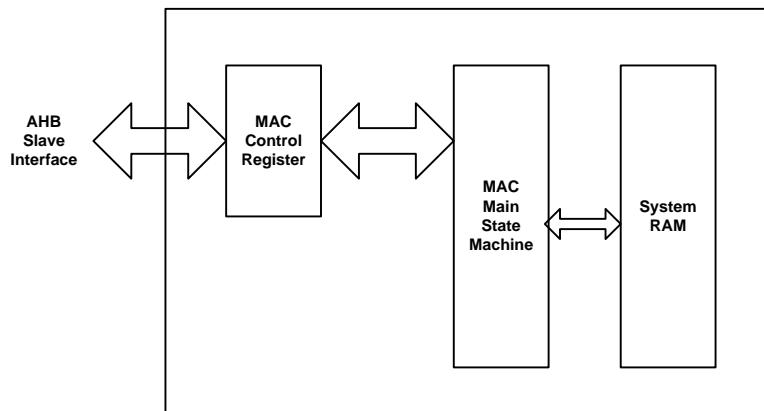
11.1. Introduction

The MAC controller is designed for accelerate the audio MAC process of 16b system.

11.2. Features

- Supports maximum 16x16 MAC function
- Supports programmable address increment or decrement per MAC operation.
- Supports 8/16/32b saturation detection
- Supports 40b accumulator.
- Supports programmable start address without align limitation.
- Supports output auto shift function.

11.3. Block Diagram



11.4. Register Description

Registers Map

Base Address : 0x5003_0000				
Name	Description	Address	Access	Reset value
MAC_CTRL	MAC Activation Register	0x5003_0000	R/W	0x0000_0000
MAC_LENGTH	MAC Length Register	0x5003_0004	R/W	0x0000_0000
MAC_ADDR_X	MAC Start address X Register	0x5003_0008	R/W	0x0000_0000
MAC_ADDR_Y	MAC Start address Y Register	0x5003_000C	R/W	0x0000_0000
MAC_STEP_XY	MAC Step X/Y Register	0x5003_0010	R/W	0x0000_0000
MAC_OUTPUT	MAC Output Register	0x5003_0014	R	0x0000_0000
MAC_OUTPUT_EXT8	MAC Output Extent Register	0x5003_0018	R	0x0000_0000
MAC_SATURATION	MAC OUT Saturation Register	0x5003_001C	R	0x0000_0000
MAC_INTSTS	MAC Interrupt Status Register	0x5003_0020	R/W	0x0000_0000

Registers Function

MAC_CTRL MAC Control Register Address : 0x5003_0000							
31	30	29	28	27	26	25	24
			--				
23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8
--				OUT_SFT_MODE[4:0]			
7	6	5	4	3	2	1	0
INT_EN	MAC_SIGN_EN	SATMODE	-	-	RESET	START	

Bit	Name	Description	Access	Reset value
[31:13]	--	Reserved	R	0
[12:8]	OUTPUT_SHIFT (OUT_SFT_MODE)	MAC output shift control. Can shift left side 1 or shift right side 0~16. 0x00: MAC_OUT right shift 0 step 0x01: MAC_OUT right shift 1 steps 0x02: MAC_OUT right shift 2 steps 0x03: MAC_OUT right shift 3 steps 0x04: MAC_OUT right shift 4 steps 0x05: MAC_OUT right shift 5 steps 0x06: MAC_OUT right shift 6 steps 0x07: MAC_OUT right shift 7 steps 0x08: MAC_OUT right shift 8 steps 0x09: MAC_OUT right shift 9 steps 0x0A: MAC_OUT right shift 10 steps 0x0B: MAC_OUT right shift 11 steps 0x0C: MAC_OUT right shift 12 steps 0x0D: MAC_OUT right shift 13 steps 0x0E: MAC_OUT right shift 14 steps 0x0F: MAC_OUT right shift 15 steps 0x10: MAC_OUT right shift 16 steps 0x1F: MAC_OUT left shift one step Other: no action	R/W	0
[7]	INT_ENABLE (INT_EN)	Interrupt Enable When this bit is set, MAC will issue interrupt to confirm CPU that MAC completes calculation. Else CPU must polling "START/BUSY" to check MAC completes calculation or not.	R/W	0
[6]	OPERATION_MODE_SEL (MAC_SIGN_EN)	MAC calculate by signed or unsigned mode 0: MAC+=signed(X) * signed(Y) 1: MAC+= signed(X) * unsigned(Y)	R/W	0
[5:4]	SATURATION_MODE_SE L (SATMODE)	Saturation mode control register. 0: 16-bit Saturation mode 1: 32-bit Saturation mode 2: 8-bit Saturation mode. 3: Reserved.	R/W	0
[3:2]	--	Reserved	R	0
[1]	RESET	MAC RESET Control MAC Controller reset register. When this bit is write 1, the MAC engine will be reset to default value immediately. This bit will be clear immediately after the MAC controller is reset.	R/W	0
[0]	START	MAC START/BUSY Control MAC Controller start to do MAC calculation with length defined in	R/W	0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description						Access	Reset value
	P_MAC_LENGTH register.								

MAC_LENGTH MAC Length Register								Address : 0x5003_0004	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	Length[7:0]	

Bit	Name	Description						Access	Reset value
[31:8]	--	Reserved						R	0
[7:0]	LENGTH	MAC length This register is indicates MAC operation times. 0: 1 MAC operation 1: 2 MAC operation ... 255: 256 MAC operation						R/W	0

MAC_ADDR_X MAC Addr X Register								Address : 0x5003_0008	
31	30	29	28	27	26	25	24	MAC_ADDRx[31:24]	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	MAC_ADDRx[15:8]	
7	6	5	4	3	2	1	0	MAC_ADDRx [7:0]	

Bit	Name	Description						Access	Reset value
[31:24]	MAC_ADDRx[31:24]	MAC Address X register The value in this register is the current address in SRAM with byte unit. The value in register will be update automatically after user read the result from the result port when MODE is set to 0.						R/W	0
[23:16]	--	Reserved						R	0
[15:0]	MAC_ADDRx[15:0]	MAC Address X register The value in this register is the current address in SRAM with byte unit.MAC_ADDRx reserved the address[23:16].						R/W	0

MAC_ADDR_Y MAC Addr Y Register								Address : 0x5003_000C	
31	30	29	28	27	26	25	24	MAC_ADDRy[31:24]	



GPCM1F SERIES PROGRAMMING GUIDE

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

Bit	Name	Description	Access	Reset value
[31:24]	MAC_ADDRy[31:24]	MAC Address Y register The value in this register is the current address in SRAM with byte unit.	R/W	0
[23:16]	--	Reserved	R	0
[15:0]	MAC_ADDRy[15:0]	MAC Address Y register The value in this register is the current address in SRAM with byte unit. MAC_ADDRy reserved the address[23:16].	R/W	0

MAC_STEP_XY MAC Address Step X/Y Register								Address : 0x5003_0010
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								STEPY[4:0]
7	6	5	4	3	2	1	0	
--								STEPX[4:0]

Bit	Name	Description	Access	Reset value
[31:13]	--	Reserved	R	0
[12:8]	ADDRY_STEP (STEPY[4:0])	MAC Address StepY register This register is used to control the address increase or decrease of AddrY after every MAC operation. This is 8b signed register. 0x10: -16 0x00: 0 0x0F: +15	R/W	0
[7:5]	--	Reserved	R	0
[4:0]	ADDRX_STEP (STEPX[4:0])	MAC Address StepX register This register is used to control the address increase or decrease of AddrX after every MAC operation. This is 8b signed register. 0x10: -16 0x00: 0 0x0F: +15	R/W	0

MAC_OUTPUT MAC Output[31:0] Register								Address : 0x5003_0014
31	30	29	28	27	26	25	24	
MAC_OUT[31:24]								
23	22	21	20	19	18	17	16	



GPCM1F SERIES PROGRAMMING GUIDE

MAC_OUT[23:16]							
15	14	13	12	11	10	9	8
MAC_OUT[15:8]							
7	6	5	4	3	2	1	0
MAC_OUT[7:0]							

Bit	Name	Description	Access	Reset value
[31:0]	MAC_OUT[31:0]	MAC Output Result Port. The value in this register is bit 31~0 of MAC 40 bits result. This register will be updated when user writes 1 to START bit and the MAC operation is done.	R	0

MAC_OUTPUT_EXT8 MAC Output Extent Register								Address : 0x5003_0018
31	30	29	28	27	26	25	24	
--	--	--	--	--	--	--	--	
23	22	21	20	19	18	17	16	
--	--	--	--	--	--	--	--	
15	14	13	12	11	10	9	8	
--	--	--	--	--	--	--	--	
7	6	5	4	3	2	1	0	
MAC_OUT_EXT8[7:0]								

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7:0]	MAC_OUT_EXT8[7:0]	MAC Output Result Port The value in this register is bit 39~32 of MAC 40 bits result. This register will be updated when user writes 1 to START bit and the MAC operation is done.	R	0

MAC_SATURATION MAC Saturation Output Register								Address : 0x5003_001C
31	30	29	28	27	26	25	24	
MAC_SAT_OUT[31:24]								
23	22	21	20	19	18	17	16	
MAC_SAT_OUT[23:16]								
15	14	13	12	11	10	9	8	
MAC_SAT_OUT[15:8]								
7	6	5	4	3	2	1	0	
MAC_SAT_OUT[7:0]								

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



GPCM1F SERIES PROGRAMMING GUIDE

MAC_INTSTS MAC Interrupt Status Register								Address : 0x5003_0020
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
--								MAC_INT

Bit	Name	Description	Access	Reset value
[31:1]	--	Reserved	R	0
[0]	MAC_INT	MAC Interrupt status flag 1: interrupt occurred Note: Write 1 to clear.	R/W	0

12. Timer

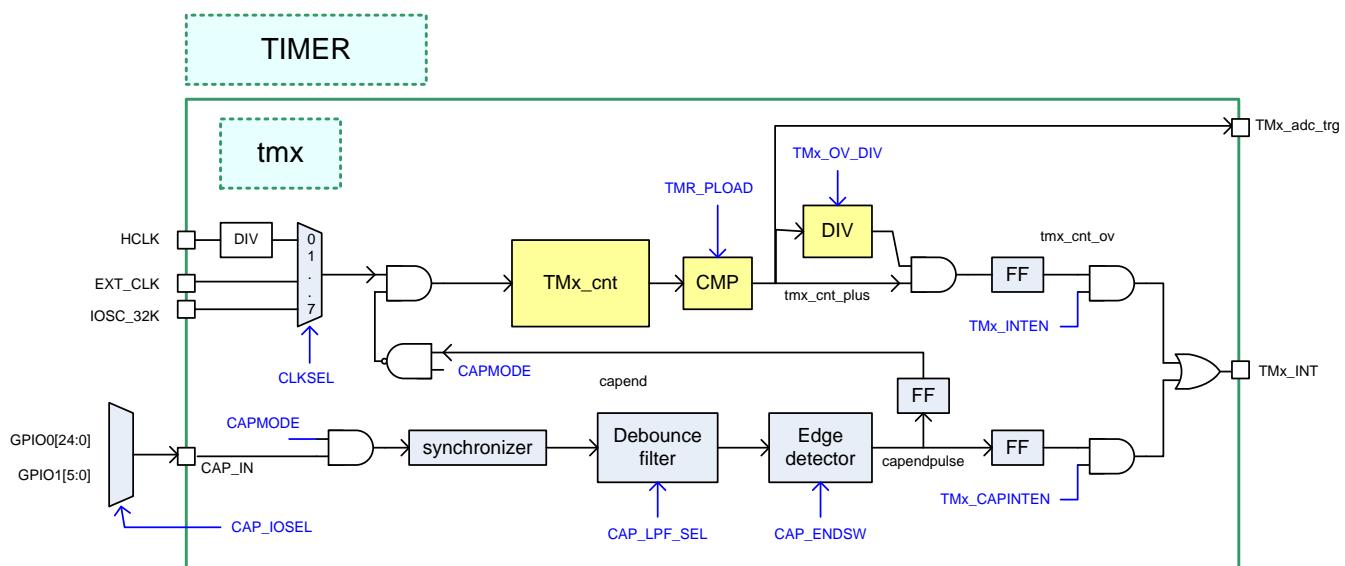
12.1. Introduction

The GPCM1F SERIALS Provide 3 set timers, provide counter mode and capture mode function to user

12.2. Feature

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

12.3. Block Diagram



12.4. Function

All timerx support two modes: counter mode and capture mode.

Counter Mode:

When the TMR_EN be set, The TMR will reload 0 and start up counter. When the timer counter is equal the TMR_PLOAD value, it will send out timer overflow event. It also has an interrupt enable and interrupt flag and 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 function on and off. In addition, it also has an interrupt enable and interrupt flag.

12.5. Register Description

Registers Map

Base Address : 0x5005_0000				
Name	Description	Address	Access	Reset value
TM0_PLOAD	Timer0 Pre-Load Data Register	0x5005_0000	R/W	0x0000_1000
TM0_COUNT	Timer0 Actual Value Register	0x5005_0004	R	0x0000_0000
TM0_CTRL	Timer0 Control Register	0x5005_0008	R/W	0x0000_0000

TM0_CAPSRC	Timer0 Capture Mode Source Selection Register	0x5005_000C	R/W	0x0000_0000
TM1_PLOAD	Timer1 Pre-Load Data Register	0x5005_0010	R/W	0x0000_1000
TM1_COUNT	Timer1 Actual Value Register	0x5005_0014	R	0x0000_0000
TM1_CTRL	Timer1 Control Register	0x5005_0018	R/W	0x0000_0000
TM1_CAPSRC	Timer1 Capture Mode Source Selection Register	0x5005_001C	R/W	0x0000_0000
TM2_PLOAD	Timer2 Pre-Load Data Register	0x5005_0020	R/W	0x0000_1000
TM2_COUNT	Timer2 Actual Value Register	0x5005_0024	R	0x0000_0000
TM2_CTRL	Timer2 Control Register	0x5005_0028	R/W	0x0000_0000
TM2_CAPSRC	Timer2 Capture Mode Source Selection Register	0x5005_002C	R/W	0x0000_0000
TM_INT_INTEN	Timer interrupt enable Register	0x5005_0030	R/W	0x0000_0000
TM_INT_INTSTS	Timer interrupt Status Register	0x5005_0034	R/W	0x0000_0000

Registers Function

TM0_TMR_PLOAD	Timer Pre-Load Data Register	Address : 0x5005 0000					
TM1_TMR_PLOAD	Timer Pre-Load Data Register	Address : 0x5005 0010					
TM2_TMR_PLOAD	Timer Pre-Load Data Register	Address : 0x5005 0020					
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
PLOAD[15:8]							
7	6	5	4	3	2	1	0
PLOAD[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	PRELOAD_DATA (PLOAD)	Timer pre-load data register The way it counts starts from 0 till TMR_PLOAD value; it then triggers the timer overflow, and TMR will return zero. Note: Because Timer counter starts counting from 0, in order to count for N timers requires PLOAD to be filled a value of (N -1).	R/W	0x1000

TM0_COUNT	Timer Actual Value Register	Address : 0x5005 0004					
TM1_COUNT	Timer Actual Value Register	Address : 0x5005 0014					
TM2_COUNT	Timer Actual Value Register	Address : 0x5005 0024					
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
TMR[15:8]							
7	6	5	4	3	2	1	0
TMR[7:0]							



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	TMR	Timer value register. Count mode is used to play the current Timer count value. Capture mode is used to store Capture value and when CAP_START = 1, it will be cleared to 0.	R/W	0x0000

TM0_CTRL Timer Control Register Address : 0x5005 0008							
TM1_CTRL Timer Control Register Address : 0x5005 0018							
TM2_CTRL Timer Control Register Address : 0x5005 0028							
31	30	29	28	27	26	25	24
--	--	--	--	--	--	--	--
23	22	21	20	19	18	17	16
--	--	--	--	--	--	--	--
15	14	13	12	11	10	9	8
TMINT_DIV				CAPENDSW	CAPEND	CAPSTART	CAPMODE
7	6	5	4	3	2	1	0
CLK_SEL			LPF_SEL			TM_RELOAD	TM_ENABLE

Bit	Name	Description	Access	Reset value																
[31:16]	--	Reserved	R	0																
[15:12]	TMINT_DIV	TIMER X INTERRUPT divider. 0: original interrupt (DAC/ADC trigger pulse), 1: divide 2 and assert interrupt, n: divide (n+1) and assert interrupt, Max 15 : divide 16	R/W	0x0																
[11]	CAP_ENDSW	Capture end edge switch. 1: Fall Edge 0: Rise Edge	R/W	0x0																
[10]	CAP_END	Capture End In Capture mode, when capture end edge happen, this bit will tie 1 and will stop TMR counter. It can be clear by CPU	R/W	0x0																
[9]	CAP_START	Capture Start Only works in capmode, Writing 1, TMR will be cleared to 0. It will auto clear, when TMR clear is finished.	R/W	0x0																
[8]	CAPMODE	Capture mode enable bits. 0 = disabled ,1 = enabled	R/W	0x0																
[7:5]	CLK_SEL	Timer input clock source selection bits <table border="1"><tr><td>TMR_CK_SRC[2:0]</td><td>Clock Source</td></tr><tr><td>000</td><td>HCLK</td></tr><tr><td>001</td><td>HCLK / 2</td></tr><tr><td>010</td><td>HCLK / 4</td></tr><tr><td>011</td><td>HCLK / 8</td></tr><tr><td>100</td><td>HCLK / 16</td></tr><tr><td>101</td><td>HCLK / 32</td></tr><tr><td>110</td><td>External Clock</td></tr></table>	TMR_CK_SRC[2:0]	Clock Source	000	HCLK	001	HCLK / 2	010	HCLK / 4	011	HCLK / 8	100	HCLK / 16	101	HCLK / 32	110	External Clock	R/W	0x0
TMR_CK_SRC[2:0]	Clock Source																			
000	HCLK																			
001	HCLK / 2																			
010	HCLK / 4																			
011	HCLK / 8																			
100	HCLK / 16																			
101	HCLK / 32																			
110	External Clock																			

Bit	Name	Description		Access	Reset value	
		111 IOSC32KHz				
[4:2]	LPF_SEL	Low pass filter selection bits		R/W	0x00	
		TMR_LPF_SEL[1:0]	Filter Cycles (HCLK)			
		000	LPF is disable			
		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	Timer Reload. Writing 1, TMR will clear 0. It will be auto-cleared when reload is done.			R/W 0x0	
[0]	TM_ENABLE	Timer enable Bit. 0 = disabled ,1 = enabled			R/W 0x0	

TM0_CAPSRC	Timer0 Capture Mode Source Selection Register								Address : 0x5005 000C
TM1_CAPSRC	Timer1 Capture Mode Source Selection Register								Address : 0x5005 001C
TM2_CAPSRC	Timer2 Capture Mode Source Selection Register								Address : 0x5005 002C
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	--	IOSEL

Bit	Name	Description								Access	Reset value
[31:5]	--	Reserved								R	0
[4:0]	IOSEL	TIMER X capture mode source GPIO selection								R/W	0x0
		SEL	KEY	SEL	KEY	SEL	KEY	SEL	KEY		
		5'h0	IOA0	5'h8	IOA8	5'h10	IOA16	5'h18	IOA24		
		5'h1	IOA1	5'h9	IOA9	5'h11	IOA17	5'h19	IOB0		
		5'h2	IOA2	5'hA	IOA10	5'h12	IOA18	5'h1A	IOB1		
		5'h3	IOA3	5'hB	IOA11	5'h13	IOA19	5'h1B	IOB2		
		5'h4	IOA4	5'hC	IOA12	5'h14	IOA20	5'h1C	IOB3		
		5'h5	IOA5	5'hD	IOA13	5'h15	IOA21	5'h1D	IOB4		
		5'h6	IOA6	5'hE	IOA14	5'h16	IOA22	5'h1E	IOB5		
		5'h7	IOA7	5'hF	IOA15	5'h17	IOA23	5'h1F	Disable		



GPCM1F SERIES PROGRAMMING GUIDE

TM_INT_INTEN Timer Interrupt Enable Register								Address : 0x5005 0030	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	--	
-	-	TM2_CAPINTE N	TM2_INTEN	TM1_CAPINTE N	TM1_INTEN	TM0_CAPINTE N	TM0_INTEN		

Bit	Name	Description	Access	Reset value
[31:6]	--	Reserved	R	0
[5]	TM2_CAPINT_ENABLE (TM2_CAPINTEN)	TM2 Capture mode Interrupt enable	R/W	0x0
[4]	TM2_INT_ENABLE (TM2_INTEN)	TM2 Interrupt enable	R/W	0x0
[3]	TM1_CAPINT_ENABLE (TM2_CAPINTEN)	TM1 Capture mode Interrupt enable	R/W	0x0
[2]	TM1_INT_ENABLE (TM2_INTEN)	TM1 Interrupt enable	R/W	0x0
[1]	TM0_CAPINT_ENABLE (TM2_CAPINTEN)	TM0 Capture mode Interrupt enable	R/W	0x0
[0]	TM0_INT_ENABLE (TM2_INTEN)	TM0 Interrupt enable	R/W	0x0

TM_INT_INTSTS Timer Interrupt Status Register								Address : 0x5005 0034	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	--	
-	-	TM2_CAPINT	TM2_INT	TM1_CAPINT	TM1_INT	TM0_CAPINT	TM0_INT		

Bit	Name	Description	Access	Reset value
[31:6]	--	Reserved	R	0
[5]	TM2_CAPINT_FLAG (TM2_CAPINT)	TM2 Capture mode Interrupt flag. It will set when capture end edge is detected. Writing 1, it will be cleared.	WOC	0x0
[4]	TM2_INT_FLAG (TM2_INT)	TM0 Interrupt flag. It will set 1 when TMR overflow. Writing 1, it will be cleared	WOC	0x0
[3]	TM1_CAPINT_FLAG (TM1_CAPINT)	TM1 Capture mode Interrupt flag. It will set when capture end edge be detection. Writing 1, it will be cleared.	WOC	0x0
[2]	TM1_INT_FLAG	TM0 Interrupt flag. It will set 1 when TMR overflow. Writing 1, it will be	WOC	0x0

Bit	Name	Description	Access	Reset value
	(TM1_INT)	cleared.		
[1]	TM0_CAPINT_FLAG (TM_CAPINT)	TM0 Capture mode Interrupt flag. It will set when capture end edge be detection. Writing 1, it will be cleared.	WOC	0x0
[0]	TM0_INT_FLAG (TM0_INT)	TM0 Interrupt flag. It will set 1 when TMR overflow. Writing 1, it will be cleared	WOC	0x0

13. General-purpose I/O (GPIOs)

13.1. Introduction

The purpose of input port and output port is to communicate with other devices. Two programmable I/O ports are available in GPCM1F series, Port IOA & Port IOB. There are 25 IOs in Port IOA and 6 IOs in Port IOB. All of 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.

13.2. Feature

- Supports 25(IOA[24:0]) + 6(IOB[5:0]) GPIO for Uart,I2C, ... special function
- All IO Support all can be programing input, output, driving and pull High/Low.
- SPIFC (Serial Peripheral Interface controller for FLASH device access) for IOB[5:0]
- Supports Touch IO Function for IOA[9:0]
- Supports SAR ADC Line for IOA[24:17]
- Supports X32K Crystal PAD for IOA[14:13]
- Supports IO Wakeup and Bit Operation

13.3. Function

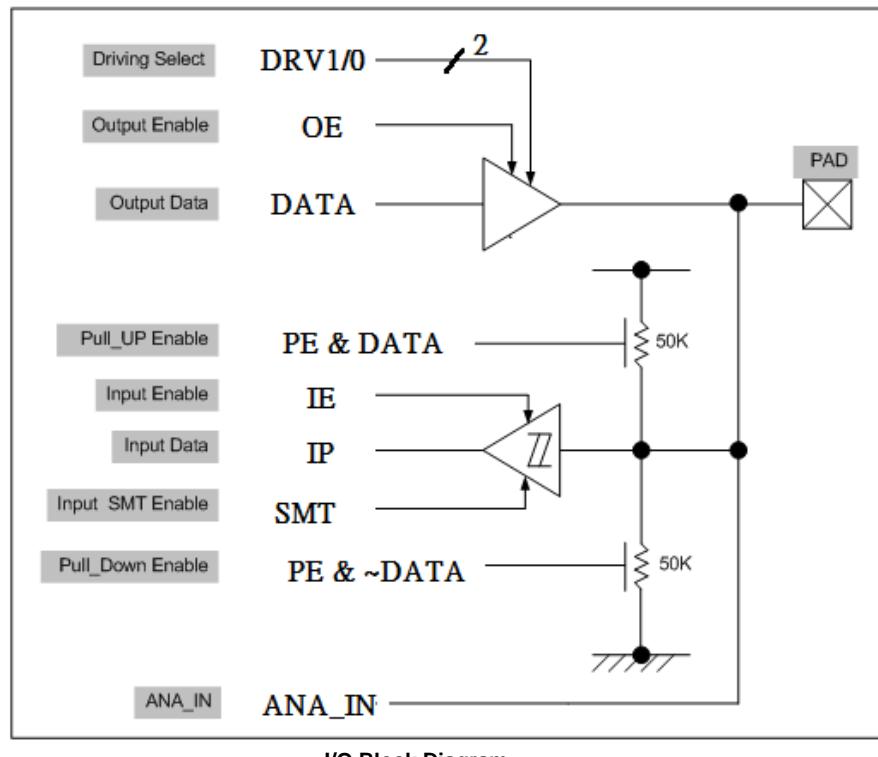
The GPCM1F series provides a bit-to-bit I/O configuration; every I/O bit can be defined individually. Each bit's configuration has four types, they are input, input floating, open—drain and output DRV. Each IO pull-high and Schmitt trigger function can be enabled individually. Some IOs is shared with digital function and analog function. The driving and sinking capability on each IO can be set up individually. User can set IOs driving and sinking capability base on actually 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**
0	0	1	Pull High	Yes**
0	1	x	Float	Yes**
0	1	x	Float	Yes**
1	0	0	Output open	No
1	0	1	Output sink mode	No
1	1	0	Output Low	No
1	1	1	Output High	No

*Default: Input with pull low.

**All of IOA/B 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 P_KEY_CHG_EN



Bit Operation

User can independently set each IO output buffer data. Ex: To set gpio0_6 output buffer data, user needs to read out **GPIOA_OBUF** and rewrite original value except bit 6 to **GPIOA_OBUF**. In GPCM1F series it supports IO bit operation. User directly writes at **GPIOA_OBIT_OBIT06** for gpio0_6 output buffer data setting.

KEY Change Detection

All pads support KEY-Change detection function. It is also as one of Wakeup System sources. User can decide which pad enable KEY-Change detection by **IOFUNC_WAKEEN[30:0]**. First, read **GPIOA_IDATA** to get PAD status (include IOA and IOB), and all pad status will be latched. After reading **GPIOA_IDATA**, for every enable KEY-Change detection function PAD, if its status change, it will be shown at **IOFUNC_STS[30:0]**. Key change function is also supporting interrupt to control.

IR Function

GPIO Function supports IR Function; it supports 7 trigger sources by **IR_CLK_SEL**. The counter will add by detection trigger source rise plus. When counter==1 & trigger source = 1, the **IR_OUT** = 1, else **IR_OUT** = 0. IR Function all support mask, pole and duty switch by **IOFUNC_CTRL1**.

13.4. Register Function

Registers Map

Base Address : 0x5007_0000				
Name	Description	Address	Access	Reset value
IOA Setting				
GPIOA_CFG0	IOA mode control register0	0x5007_0000	R/W	0x0000_0000
GPIOA_CFG1	IOA mode control register1	0x5007_0004	R/W	0x0000_0000
GPIOA_DRV0	IOA driving control register	0x5007_0008	R/W	0x0000_0000
GPIOA_DRV1	IOA driving control register	0x5007_000C	R/W	0x0000_0000



GPCM1F SERIES PROGRAMMING GUIDE

Base Address : 0x5007_0000				
Name	Description	Address	Access	Reset value
GPIOA_SMT	IOA Schmitt trigger control register	0x5007_0010	R/W	0x01FF_FFFF
GPIOA_OBUF	IOA output data buffer register	0x5007_0014	R/W	0x0000_0000
GPIOA_IDATA	IOA input data status register	0x5007_0018	R	0x0000_0000
IOA Bit Operation				
GPIOA_OBIT_OBIT00	IOA0 bit operation	0x5007_0020	R/W	0x0000_0000
GPIOA_OBIT_OBIT01	IOA1 bit operation	0x5007_0024	R/W	0x0000_0000
GPIOA_OBIT_OBIT02	IOA2 bit operation	0x5007_0028	R/W	0x0000_0000
GPIOA_OBIT_OBIT03	IOA3 bit operation	0x5007_002C	R/W	0x0000_0000
GPIOA_OBIT_OBIT04	IOA4 bit operation	0x5007_0030	R/W	0x0000_0000
GPIOA_OBIT_OBIT05	IOA5 bit operation	0x5007_0034	R/W	0x0000_0000
GPIOA_OBIT_OBIT06	IOA6 bit operation	0x5007_0038	R/W	0x0000_0000
GPIOA_OBIT_OBIT07	IOA7 bit operation	0x5007_003C	R/W	0x0000_0000
GPIOA_OBIT_OBIT08	IOA8 bit operation	0x5007_0040	R/W	0x0000_0000
GPIOA_OBIT_OBIT09	IOA9 bit operation	0x5007_0044	R/W	0x0000_0000
GPIOA_OBIT_OBIT10	IOA10 bit operation	0x5007_0048	R/W	0x0000_0000
GPIOA_OBIT_OBIT11	IOA11 bit operation	0x5007_004C	R/W	0x0000_0000
GPIOA_OBIT_OBIT12	IOA12 bit operation	0x5007_0050	R/W	0x0000_0000
GPIOA_OBIT_OBIT13	IOA13 bit operation	0x5007_0054	R/W	0x0000_0000
GPIOA_OBIT_OBIT14	IOA14 bit operation	0x5007_0058	R/W	0x0000_0000
GPIOA_OBIT_OBIT15	IOA15 bit operation	0x5007_005C	R/W	0x0000_0000
GPIOA_OBIT_OBIT16	IOA16 bit operation	0x5007_0060	R/W	0x0000_0000
GPIOA_OBIT_OBIT17	IOA17 bit operation	0x5007_0064	R/W	0x0000_0000
GPIOA_OBIT_OBIT18	IOA18 bit operation	0x5007_0068	R/W	0x0000_0000
GPIOA_OBIT_OBIT19	IOA19 bit operation	0x5007_006C	R/W	0x0000_0000
GPIOA_OBIT_OBIT20	IOA20 bit operation	0x5007_007_0	R/W	0x0000_0000
GPIOA_OBIT_OBIT21	IOA21 bit operation	0x5007_0074	R/W	0x0000_0000
GPIOA_OBIT_OBIT22	IOA22 bit operation	0x5007_0078	R/W	0x0000_0000
GPIOA_OBIT_OBIT23	IOA23 bit operation	0x5007_007C	R/W	0x0000_0000
GPIOA_OBIT_OBIT24	IOA24 bit operation	0x5007_0080	R/W	0x0000_0000
IOB Setting				
GPIOB_CFG0	IOB mode control register	0x5007_0100	R/W	0x0000_0000
GPIOB_DRV0	IOB driving control register	0x5007_0108	R/W	0x0000_0000
GPIOB_SMT	IOB Schmitt trigger control register	0x5007_0110	R/W	0x0000_003F
GPIOB_OBUF	IOB output data buffer register	0x5007_0114	R/W	0x0000_0000
GPIOB_IDATA	IOB input data status register	0x5007_0118	R	0x0000_0000
IOB Bit Operation				
GPIOB_OBIT_OBIT00	IOB0 bit operation	0x5007_0120	R/W	0x0000_0000
GPIOB_OBIT_OBIT01	IOB1 bit operation	0x5007_0124	R/W	0x0000_0000
GPIOB_OBIT_OBIT02	IOB2 bit operation	0x5007_0128	R/W	0x0000_0000
GPIOB_OBIT_OBIT03	IOB3 bit operation	0x5007_012C	R/W	0x0000_0000
GPIOB_OBIT_OBIT04	IOB4 bit operation	0x5007_0130	R/W	0x0000_0000
GPIOB_OBIT_OBIT05	IOB5 bit operation	0x5007_0134	R/W	0x0000_0000
Special Function Setting				
IOFUNC_CTRL0	GPIO function selection control register	0x5007_0200	R/W	0x0000_0431

Base Address : 0x5007_0000						
Name	Description			Address	Access	Reset value
IOFUNC_CTRL1	GPIO IR function control register			0x5007_0204	R/W	0x0000_0040
IOFUNC_CTRL2	GPIO EXT 0~3 Input pins select control register			0x5007_0208	R/W	0x1F1F_1F1F
IOFUNC_WAKEEN	GPIO key-change wake up enable register			0x5007_0210	R/W	0x0000_0000
IOFUNC_STS	GPIO key-change interrupt status register			0x5007_0214	R/W	0x0000_0000

Registers Function

GPIOA_CFG0 IOA MODE control register								Address : 0x5007_0000	
31	30	29	28	27	26	25	24		
IOA[15]_MODE			IOA[14]_MODE			IOA[13]_MODE			IOA[12]_MODE
23	22	21	20	19	18	17	16		
IOA[11]_MODE			IOA[10]_MODE			IOA[9]_MODE			IOA[8]_MODE
15	14	13	12	11	10	9	8		
IOA[7]_MODE			IOA[6]_MODE			IOA[5]_MODE			IOA[4]_MODE
7	6	5	4	3	2	1	0		
IOA[3]_MODE			IOA[2]_MODE			IOA[1]_MODE			IOA[0]_MODE

Bit	Name	Description				Access	Reset value
[31:30]	IOA[15]_MODE	IOA[15] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[29:28]	IOA[14]_MODE	IOA[14] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[27:26]	IOA[13]_MODE	IOA[13] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode E		
[25:24]	IOA[21]_MODE	IOA[21] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[23:22]	IOA[11]_MODE	IOA[11] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[21:20]	IOA[10]_MODE	IOA[10] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		

Bit	Name	Description				Access	Reset value
		2'b01	input floating	2'b11	output mode		
[19:18]	IOA[9]_MODE	IOA[9] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[17:16]	IOA[8]_MODE	IOA[8] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[15:14]	IOA[7]_MODE	IOA[7] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[13:12]	IOA[6]_MODE	IOA[6] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[11:10]	IOA[5]_MODE	IOA[5] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[9:8]	IOA[4]_MODE	IOA[4] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[7:6]	IOA[3]_MODE	IOA[3] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[5:4]	IOA[2]_MODE	IOA[2] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[3:2]	IOA[1]_MODE	IOA[1] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		

Bit	Name	Description				Access	Reset value
[1:0]	IOA[0]_MODE	IOA[0] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		

GPIOA_CFG1 IOA MODE control register								Address : 0x5007_0004	
31	30	29	28	27	26	25	24		
23	22	21	20	19	18	17	16		
			--			IOA[24]_MODE			
15	14	13	12	11	10	9	8		
IOA[23]_MODE		IOA[22]_MODE		IOA[21]_MODE		IOA[20]_MODE			
7	6	5	4	3	2	1	0		
IOA[19]_MODE		IOA[18]_MODE		IOA[17]_MODE		IOA[16]_MODE			

Bit	Name	Description				Access	Reset value
[31:18]	--	Reserved				R	0
[17:16]	IOA[24]_MODE	IOA[24] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[15:14]	IOA[23]_MODE	IOA[23] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[13:12]	IOA[22]_MODE	IOA[22] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[11:10]	IOA[21]_MODE	IOA[21] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[9:8]	IOA[20]_MODE	IOA[20] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		

Bit	Name	Description				Access	Reset value
[7:6]	IOA[19]_MODE	IOA[19] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain	R/W	0
		2'b01	input floating	2'b11	output mode		
[5:4]	IOA[18]_MODE	IOA[18] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain	R/W	0
		2'b01	input floating	2'b11	output mode		
[3:2]	IOA[17]_MODE	IOA[17] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain	R/W	0
		2'b01	input floating	2'b11	output mode		
[1:0]	IOA[16]_MODE	IOA[16] MODE control register					
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain	R/W	0
		2'b01	input floating	2'b11	output mode		

GPIOA_DRV0 IOA driving strength control register								Address : 0x5007_0008
31	30	29	28	27	26	25	24	
	IOA[15]_DRV		IOA[14]_DRV		IOA[13]_DRV		IOA[12]_DRV	
23	22	21	20	19	18	17	16	
	IOA[11]_DRV		IOA[10]_DRV		IOA[9]_DRV		IOA[8]_DRV	
15	14	13	12	11	10	9	8	
	IOA[7]_DRV		IOA[6]_DRV		IOA[5]_DRV		IOA[4]_DRV	
7	6	5	4	3	2	1	0	
	IOA[3]_DRV		IOA[2]_DRV		IOA[1]_DRV		IOA[0]_DRV	

Bit	Name	Description				Access	Reset value
[31:30]	IO15_DRV_SEL (IOA[15]_DRV)	IOA[15] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA	R/W	0
		2'b01	8mA	2'b11	16mA		
[29:28]	IO14_DRV_SEL (IOA[14]_DRV)	IOA[14] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA	R/W	0
		2'b01	8mA	2'b11	16mA		
[27:26]	IO13_DRV_SEL (IOA[13]_DRV)	IOA[13] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA	R/W	0
		2'b01	8mA	2'b11	16mA		
[25:24]	IO12_DRV_SEL	IOA[12] Driving strength control register				R/W	0

Bit	Name	Description				Access	Reset value
	(IOA[12]_DRV)	DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[23:22]	IO11_DRV_SEL (IOA[11]_DRV)	IOA[11] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[21:20]	IO10_DRV_SEL (IOA[10]_DRV)	IOA[10] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[19:18]	IO9_DRV_SEL (IOA[9]_DRV)	IOA[9] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[17:16]	IO8_DRV_SEL (IOA[8]_DRV)	IOA[8] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[15:14]	IO7_DRV_SEL (IOA[7]_DRV)	IOA[7] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[13:12]	IO6_DRV_SEL (IOA[6]_DRV)	IOA[6] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[11:10]	IO5_DRV_SEL (IOA[5]_DRV)	IOA[5] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[9:8]	IO4_DRV_SEL (IOA[4]_DRV)	IOA[4] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[7:6]	IO3_DRV_SEL (IOA[3]_DRV)	IOA[3] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[5:4]	IO2_DRV_SEL (IOA[2]_DRV)	IOA[2] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[3:2]	IO1_DRV_SEL	IOA[1] Driving strength control register				R/W	0

Bit	Name	Description				Access	Reset value
	(IOA[1]_DRV)	DRV[1:0]					
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[1:0]	IO0_DRV_SEL (IOA[0]_DRV)	IOA[0] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		

GPIOA_DRV1 IOA driving strength control register								Address : 0x5007_000C
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								IOA[24]_DRV
15	14	13	12	11	10	9	8	
IOA[23]_DRV		IOA[22]_DRV		IOA[21]_DRV		IOA[20]_DRV		
7	6	5	4	3	2	1	0	
IOA[19]_DRV		IOA[18]_DRV		IOA[17]_DRV		IOA[16]_DRV		

Bit	Name	Description				Access	Reset value
[31:18]	--	Reserved				R	0
[17:16]	IO24_DRV_SEL (IOA[24]_DRV)	IOA[24] Driving strength control register		DRV[1:0]			
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[15:14]	IO23_DRV_SEL (IOA[23]_DRV)	IOA[23] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[13:12]	IO22_DRV_SEL (IOA[22]_DRV)	IOA[22] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[11:10]	IO21_DRV_SEL (IOA[21]_DRV)	IOA[21] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[9:8]	IO20_DRV_SEL (IOA[20]_DRV)	IOA[20] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[7:6]	IO19_DRV_SEL (IOA[19]_DRV)	IOA[19] Driving strength control register					
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		

Bit	Name	Description				Access	Reset value
[5:4]	IO18_DRV_SEL (IOA[18]_DRV)	IOA[18] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[3:2]	IO17_DRV_SEL (IOA[17]_DRV)	IOA[17] Driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[1:0]	IO16_DRV_SEL (IOA[16]_DRV)	IOA[16] driving strength control register				R/W	0
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		

GPIOA_SMT								IOA Schmitt trigger control register		Address : 0x5007_0010	
31	30	29	28	27	26	25	24				
--								IO24_SMT			
23	22	21	20	19	18	17	16	23	22	21	20
IO23_SMT	IO22_SMT	IO21_SMT	IO20_SMT	IO19_SMT	IO18_SMT	IO17_SMT	IO16_SMT	IO23_SMT	IO22_SMT	IO21_SMT	IO20_SMT
15	14	13	12	11	10	9	8	15	14	13	12
IO15_SMT	IO14_SMT	IO13_SMT	IO12_SMT	IO11_SMT	IO10_SMT	IO9_SMT	IO8_SMT	IO15_SMT	IO14_SMT	IO13_SMT	IO12_SMT
7	6	5	4	3	2	1	0	7	6	5	4
IO7_SMT	IO6_SMT	IO5_SMT	IO4_SMT	IO3_SMT	IO2_SMT	IO1_SMT	IO0_SMT	IO7_SMT	IO6_SMT	IO5_SMT	IO4_SMT

Bit	Name	Description				Access	Reset value
[31:25]	--	Reserved				R	0
[24]	IO24_SMT	IOA.24 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[23]	IO23_SMT	IOA.23 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[22]	IO22_SMT	IOA.22 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[21]	IO21_SMT	IOA.21 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[20]	IO20_SMT	IOA.20 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[19]	IO19_SMT	IOA.19 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[18]	IO18_SMT	IOA.18 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[17]	IO17_SMT	IOA.17 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[16]	IO16_SMT	IOA.16 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[15]	IO15_SMT	IOA.15 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[14]	IO14_SMT	IOA.14 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[13]	IO13_SMT	IOA.13 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[12]	IO12_SMT	IOA.12 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[11]	IO11_SMT	IOA.11 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[10]	IO10_SMT	IOA.10 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[9]	IO9_SMT	IOA.9 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[8]	IO8_SMT	IOA.8 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[7]	IO7_SMT	IOA.7 schmitt control register. 1:Enabled 0:Disabled				R/W	1
[6]	IO6_SMT	IOA.6 schmitt control register. 1:Enabled 0:Disabled				R/W	1

Bit	Name	Description	Access	Reset value
[5]	IO5_SMT	IOA.5 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[4]	IO4_SMT	IOA.4 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[3]	IO3_SMT	IOA.3 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[2]	IO2_SMT	IOA.2 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[1]	IO1_SMT	IOA.1 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[0]	IO0_SMT	IOA.0 schmitt control register. 1:Enabled 0:Disabled	R/W	1

GPIOA_OBUF IOA output data buffer register								Address : 0x5007_0014
31	30	29	28	27	26	25	24	
--								IOA_OBUF [24]
23	22	21	20	19	18	17	16	
IOA_OBUF [23:16]								
15	14	13	12	11	10	9	8	
IOA_OBUF [14:8]								
7	6	5	4	3	2	1	0	
IOA_OBUF [7:0]								

Bit	Name	Description	Access	Reset value
[31:25]	--	Reserved	R	0
[24:0]	IOA_OBUF	IOA.x output data buffer register. According to the Input/ Output mode set in GPIOA_CFG0 and GPIOA_CFG1, the settings of IOA_OBUF may vary. Output mode: 0: Output Lo 1: Output Hi Output open-drain mode: 0: Output open 1: Output sink Input mode: 0: Input pull-Lo 1: Input pull-Hi Note: Writing data into GPIOA_OBUF will synchronously change the GPIOA_OBUF and GPIOA_IDATA.	R/W	0

GPIOA_IDATA IOA input data status register								Address : 0x5007_0018
31	30	29	28	27	26	25	24	
--								IOA_IDATA [24]
23	22	21	20	19	18	17	16	
IOA_IDATA [23:16]								
15	14	13	12	11	10	9	8	
IOA_IDATA [14:8]								
7	6	5	4	3	2	1	0	
IOA_IDATA [7:0]								

Bit	Name	Description						Access	Reset value
[31:25]	--	Reserved						R	0
[24:0]	IOA_IDATA	Read data from the IOAx pad. This port is read only						R	0

GPIOA0_OBIT IOA0 Bit operation register								Address : 0x5007_0020	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
--								IOA0_OBIT	

Bit	Name	Description						Access	Reset value
[31:1]	--	Reserved						R	0
[0]	IOA0_OBIT	IOA0 bit operation. SW can write this register to change GPIOA_OBUF[0] value.						R/W	0

GPIOA1_OBIT IOA1 Bit operation register								Address : 0x500_0024	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
--								IOA1_OBIT	

Bit	Name	Description						Access	Reset value
[31:2]	--	Reserved						R	0
[1]	IOA1_OBIT	IOA1 bit operation. SW can write this register to change GPIOA_OBUF[1] value.						R/W	0
[0]	--	Reserved						R	0

GPIOA2_OBIT IOA2 Bit operation register								Address : 0x5007_0028	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
--								IOA2_OBIT	

Bit	Name	Description	Access	Reset value
[31:3]	--	Reserved	R	0
[2]	IOA2_OBIT	IOA2 bit operation. SW can write this register to change GPIOA_OBUF[2] value.	R/W	0
[1:0]	--	Reserved	R	0

GPIOA3_OBIT IOA3 Bit operation register								Address : 0x5007_002C
31	30	29	28	27	26	25	24	
			--					
23	22	21	20	19	18	17	16	
			--					
15	14	13	12	11	10	9	8	
			--					
7	6	5	4	3	2	1	0	
		--		IOA3_OBIT		--		

Bit	Name	Description	Access	Reset value
[31:4]	--	Reserved	R	0
[3]	IOA3_OBIT	IOA3 bit operation. SW can write this register to change GPIOA_OBUF[3] value.	R/W	0
[2:0]	--	Reserved	R	0

GPIOA4_OBIT IOA4 Bit operation register								Address : 0x5007_0030
31	30	29	28	27	26	25	24	
			--					
23	22	21	20	19	18	17	16	
			--					
15	14	13	12	11	10	9	8	
			--					
7	6	5	4	3	2	1	0	
		--		IOA4_OBIT		--		

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0
[4]	IOA4_OBIT	IOA4 bit operation. SW can write this register to change GPIOA_OBUF[4] value.	R/W	0
[3:0]	--	Reserved	R	0

GPIOA5_OBIT IOA5 Bit operation register								Address : 0x5007_0034
31	30	29	28	27	26	25	24	
			--					
23	22	21	20	19	18	17	16	
			--					
15	14	13	12	11	10	9	8	

7	6	5	4	--	3	2	1	0
--		IOA5_OBIT		--				

Bit	Name	Description	Access	Reset value
[31:6]	--	Reserved	R	0
[5]	IOA5_OBIT	IOA5 bit operation. SW can write this register to change GPIOA_OBUF[5] value.	R/W	0
[4:0]	--	Reserved	R	0

GPIOA6_OBIT IOA6 Bit operation register								Address : 0x5007_0038
31	30	29	28	27	26	25	24	
--			--					
23	22	21	20	19	18	17	16	
--			--					
15	14	13	12	11	10	9	8	
--			--					
7	6	5	4	3	2	1	0	
--	IOA6_OBIT		--					

Bit	Name	Description	Access	Reset value
[31:7]	--	Reserved	R	0
[6]	IOA6_OBIT	IOA6 bit operation. SW can write this register to change GPIOA_OBUF[6] value.	R/W	0
[5:0]	--	Reserved	R	0

GPIOA7_OBIT IOA7 Bit operation register								Address : 0x5007_003C
31	30	29	28	27	26	25	24	
--			--					
23	22	21	20	19	18	17	16	
--			--					
15	14	13	12	11	10	9	8	
--			--					
7	6	5	4	3	2	1	0	
IOA7_OBIT			--					

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7]	IOA7_OBIT	IOA7 bit operation. SW can write this register to change GPIOA_OBUF[7] value.	R/W	0
[6:0]	--	Reserved	R	0

GPIOA8_OBIT IOA8 Bit operation register								Address : 0x5007_0040
31	30	29	28	27	26	25	24	
--			--					

23	22	21	20	19	18	17	16
				--			
15	14	13	12	11	10	9	8
			--				IOA8_OBIT
7	6	5	4	3	2	1	0
				--			

Bit	Name	Description						Access	Reset value
[31:9]	--	Reserved						R	0
[8]	IOA8_OBIT	IOA8 bit operation. SW can write this register to change GPIOA_OBUF[8] value.						R/W	0
[7:0]	--	Reserved						R	0

GPIOA9_OBIT IOA9 Bit operation register								Address : 0x5007_0044	
31	30	29	28	27	26	25	24		
				--					
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
			--			IOA9_OBIT	--		
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description						Access	Reset value
[31:10]	--	Reserved						R	0
[9]	IOA9_OBIT	IOA9 bit operation. SW can write this register to change GPIOA_OBUF[9] value.						R/W	0
[8:0]	--	Reserved						R	0

GPIOA10_OBIT IOA10 Bit operation register								Address : 0x5007_0048	
31	30	29	28	27	26	25	24		
				--					
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
			--			IOA10_OBIT	--		
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description						Access	Reset value
[31:11]	--	Reserved						R	0
[10]	IOA10_OBIT	IOA10 bit operation. SW can write this register to change GPIOA_OBUF[10] value.						R/W	0
[9:0]	--	Reserved						R	0

GPIOA11_OBIT IOA11 Bit operation register								Address : 0x5007_004C	
31	30	29	28	27	26	25	24		
				--					
23	22	21	20	19	18	17	16		
				--					
15	14	13	12	11	10	9	8		
			--	IOA11_OBIT		--			
7	6	5	4	3	2	1	0		
				--					

Bit	Name	Description							Access	Reset value
[31:12]	--	Reserved							R	0
[11]	IOA11_OBIT	IOA11 bit operation. SW can write this register to change GPIOA_OBUF[11] value.							R/W	0
[10:0]	--	Reserved							R	0

GPIOA12_OBIT IOA12 Bit operation register								Address : 0x5007_0050	
31	30	29	28	27	26	25	24		
				--					
23	22	21	20	19	18	17	16		
				--					
15	14	13	12	11	10	9	8		
			--	IOA12_OBIT		--			
7	6	5	4	3	2	1	0		
				--					

Bit	Name	Description							Access	Reset value
[31:13]	--	Reserved							R	0
[12]	IOA12_OBIT	IOA12 bit operation. SW can write this register to change GPIOA_OBUF[12] value.							R/W	0
[11:0]	--	Reserved							R	0

GPIOA13_OBIT IOA13 Bit operation register								Address : 0x5007_0054	
31	30	29	28	27	26	25	24		
				--					
23	22	21	20	19	18	17	16		
				--					
15	14	13	12	11	10	9	8		
			--	IOA13_OBIT		--			
7	6	5	4	3	2	1	0		
				--					

Bit	Name	Description							Access	Reset value
[31:14]	--	Reserved							R	0
[13]	IOA13_OBIT	IOA13 bit operation. SW can write this register to change GPIOA_OBUF[13]							R/W	0

Bit	Name	Description						Access	Reset value
		value.							
[12:0]	--	Reserved						R	0

GPIOA14_OBIT IOA14 Bit operation register								Address : 0x5007_0058	
31	30	29	28	27	26	25	24		
			--						
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
--	IOA14_OBIT			--					
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description						Access	Reset value
[31:15]	--	Reserved						R	0
[14]	IOA14_OBIT	IOA14 bit operation. SW can write this register to change GPIOA_OBUF[14] value.						R/W	0
[13:0]	--	Reserved						R	0

GPIOA15_OBIT IOA15 Bit operation register								Address : 0x5007_005C	
31	30	29	28	27	26	25	24		
			--						
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
IOA15_OBIT			--						
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description						Access	Reset value
[31:16]	--	Reserved						R	0
[15]	IOA15_OBIT	IOA15 bit operation. SW can write this register to change GPIOA_OBUF[15] value.						R/W	0
[14:0]	--	Reserved						R	0

GPIOA16_OBIT IOA16 Bit operation register								Address : 0x500_0060	
31	30	29	28	27	26	25	24		
			--						
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
			--						
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description	Access	Reset value
[31:17]	--	Reserved	R	0
[16]	IOA16_OBIT	IOA16 bit operation. SW can write this register to change GPIOA_OBUF[16] value.	R/W	0
[15:0]	--	Reserved	R	0

GPIOA17_OBIT IOA17 Bit operation register								Address : 0x5007_0064
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	IOA17_OBIT
7	6	5	4	3	2	1	0	--
								--

Bit	Name	Description	Access	Reset value
[31:18]	--	Reserved	R	0
[17]	IOA17_OBIT	IOA17 bit operation. SW can write this register to change GPIOA_OBUF[17] value.	R/W	0
[16:0]	--	Reserved	R	0

GPIOA18_OBIT IOA18 Bit operation register								Address : 0x5007_0068
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	IOA18_OBIT
7	6	5	4	3	2	1	0	--
								--

Bit	Name	Description	Access	Reset value
[31:19]	--	Reserved	R	0
[18]	IOA18_OBIT	IOA18 bit operation. SW can write this register to change GPIOA_OBUF[18] value.	R/W	0
[17:0]	--	Reserved	R	0

GPIOA19_OBIT IOA19 Bit operation register								Address : 0x5007_006C
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	IOA19_OBIT
								--

7	6	5	4	3	2	1	0
--							

Bit	Name	Description						Access	Reset value
[31:20]	--	Reserved						R	0
[19]	IOA19_OBIT	IOA19 bit operation. SW can write this register to change GPIOA_OBUF[19] value.						R/W	0
[18:0]	--	Reserved						R	0

GPIOA20_OBIT IOA20 Bit operation register								Address : 0x5007_0070	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
--									

Bit	Name	Description						Access	Reset value
[31:21]	--	Reserved						R	0
[20]	IOA20_OBIT	IOA20 bit operation. SW can write this register to change GPIOA_OBUF[20] value.						R/W	0
[19:0]	--	Reserved						R	0

GPIOA21_OBIT IOA21 Bit operation register								Address : 0x5007_0074	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
--									

Bit	Name	Description						Access	Reset value
[31:22]	--	Reserved						R	0
[21]	IOA21_OBIT	IOA21 bit operation. SW can write this register to change GPIOA_OBUF[21] value.						R/W	0
[20:0]	--	Reserved						R	0

GPIOA22_OBIT IOA22 Bit operation register								Address : 0x5007_0078	
31	30	29	28	27	26	25	24		
--									

23	22	21	20	19	18	17	16
--	IOA22_OBIT			--			
15	14	13	12	11	10	9	8
			--				
7	6	5	4	3	2	1	0
			--				

Bit	Name	Description						Access	Reset value
[31:23]	--	Reserved						R	0
[22]	IOA22_OBIT	IOA22 bit operation. SW can write this register to change GPIOA_OBUF[22] value.						R/W	0
[21:0]	--	Reserved						R	0

GPIOA23_OBIT IOA23 Bit operation register								Address : 0x5007_007C	
31	30	29	28	27	26	25	24		
			--						
23	23	21	20	19	18	17	16		
IOA23_OBIT			--						
15	14	13	12	11	10	9	8		
			--						
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description						Access	Reset value
[31:24]	--	Reserved						R	0
[23]	IOA23_OBIT	IOA23 bit operation. SW can write this register to change GPIOA_OBUF[23] value.						R/W	0
[22:0]	--	Reserved						R	0

GPIOA24_OBIT IOA24 Bit operation register								Address : 0x5007_0080	
31	30	19	28	27	26	25	24		
			--						IOA24_OBIT
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
			--						
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description						Access	Reset value
[31:25]	--	Reserved						R	0
[24]	IOA24_OBIT	IOA24 bit operation. SW can write this register to change GPIOA_OBUF[24] value.						R/W	0
[23:0]	--	Reserved						R	0



GPCM1F SERIES PROGRAMMING GUIDE

GPIOB_CFG0 IOB MODE control register								Address : 0x5007_0100	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	IOB[5]_MODE	IOB[4]_MODE
IOB[3]_MODE		IOB[2]_MODE		IOB[1]_MODE		IOB[0]_MODE			

Bit	Name	Description				Access	Reset value
[31:12]	-	Reserved				R/W	0
[11:10]	IOB[5]_MODE	IOB[5] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[9:8]	IOB[4]_MODE	IOB[4] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[7:6]	IOB[3]_MODE	IOB[3] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[5:4]	IOB[2]_MODE	IOB[2] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[3:2]	IOB[1]_MODE	IOB[1] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		
[1:0]	IOB[0]_MODE	IOB[0] MODE control register				R/W	0
		MODE[1:0]		MODE[1:0]			
		2'b00	input mode	2'b10	output open-drain		
		2'b01	input floating	2'b11	output mode		

GPIOB_DRV0 IOB driving strength control register								Address : 0x5007_0108	
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	



GPCM1F SERIES PROGRAMMING GUIDE

		--			IOB[5]_DRV		IOB[4]_DRV
7	6	5	4	3	2	1	0
IOB[3]_DRV		IOB[2]_DRV		IOB[1]_DRV		IOB[0]_DRV	

Bit	Name	Description				Access	Reset value
[31:12]	--	Reserved				R	0
[11:10]	IO5_DRV_SEL (IOB[5]_DRV)	IOB[5] Driving strength control register				R/W	01
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[9:8]	IO4_DRV_SEL (IOB[4]_DRV)	IOB[4] Driving strength control register				R/W	01
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[7:6]	IO3_DRV_SEL (IOB[3]_DRV)	IOB[3] Driving strength control register				R/W	01
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[5:4]	IO2_DRV_SEL (IOB[2]_DRV)	IOB[2] Driving strength control register				R/W	01
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[3:2]	IO1_DRV_SEL (IOB[1]_DRV)	IOB[1] Driving strength control register				R/W	01
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		
[1:0]	IO0_DRV_SEL (IOB[0]_DRV)	IOB[0] Driving strength control register				R/W	01
		DRV[1:0]		DRV[1:0]			
		2'b00	4mA	2'b10	12mA		
		2'b01	8mA	2'b11	16mA		

GPIOB_SMT								IOB Schmitt trigger control register		Address : 0x5007_0110	
31	30	29	28	27	26	25	24	--			
23	22	21	20	19	18	17	16	--			
15	14	13	12	11	10	9	8	--			
7	6	5	4	3	2	1	0	--			
--		IO5_SMT	IO4_SMT	IO3_SMT	IO2_SMT	IO1_SMT	IO0_SMT				

Bit	Name	Description				Access	Reset value
[31:6]	--	Reserved				R	0

Bit	Name	Description	Access	Reset value
[5]	IO5_SMT	IOB.5 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[4]	IO4_SMT	IOB.4 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[3]	IO3_SMT	IOB.3 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[2]	IO2_SMT	IOB.2 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[1]	IO1_SMT	IOB.1 schmitt control register. 1:Enabled 0:Disabled	R/W	1
[0]	IO0_SMT	IOB.0 schmitt control register. 1:Enabled 0:Disabled	R/W	1

GPIOB_OBUF IOB output data buffer register								Address : 0x5007_0114
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
IOB_OBUF[5:0]								

Bit	Name	Description	Access	Reset value
[31:6]	--	Reserved	R	0
[5:0]	IOB_OBUF	IOB.x output data buffer register. According to the input or output mode defined in GPIOB_CFG0, the settings of IOB_OBUF may vary. Output mode: 0: Output Lo 1: Output Hi Output open-drain mode: 0: Output open 1: Output sink Input mode: 0: Input pull-Lo 1: Input pull-Hi Note: Writing data into GPIOB_OBUF will synchronously change the GPIOB_OBUF and GPIOB_IDATA.	R/W	0

GPIOB_IDATA IOB input data status register								Address : 0x5007_0118
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
IOB_IDATA[5:0]								

Bit	Name	Description	Access	Reset value
[31:6]	--	Reserved	R	0

Bit	Name	Description						Access	Reset value
[5:0]	IOB_IDATA	Read data from the IOBx pad. This port is read only.						R	0

GPIOB0_OBIT IOB0 Bit operation register								Address : 0x5007_0120
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
								IOB0_OBIT

Bit	Name	Description						Access	Reset value
[31:1]	--	Reserved						R	0
[0]	IOB0_OBIT	IOB0 bit operation. SW writes this register to change GPIOB_OBUF[0] value.						R/W	0

GPIOB1_OBIT IOB1 Bit operation register								Address : 0x5007_0124
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
								IOB1_OBIT
								--

Bit	Name	Description						Access	Reset value
[31:2]	--	Reserved						R	0
[1]	IOB1_OBIT	IOB1 bit operation. SW writes this register to change GPIOB_OBUF[1] value.						R/W	0
[0]	Reverse	--						--	--

GPIOB2_OBIT IOB2 Bit operation register								Address : 0x5007_0128
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	--
								IOB2_OBIT
								--

Bit	Name	Description	Access	Reset value
[31:3]	--	Reserved	R	0
[2]	IOB2_OBIT	IOB2 bit operation. SW writes this register to change GPIOB_OBUF[2] value.	R/W	0
[1:0]	Reverse	--	--	--

GPIOB3_OBIT IOB3 Bit operation register								Address : 0x5007_012C
7	6	5	4	3	2	1	0	
--								IOB3_OBIT

Bit	Name	Description	Access	Reset value
[31:4]	--	Reserved	R	0
[3]	IOB3_OBIT	IOB3 bit operation. SW writes this register to change GPIOB_OBUF[3] value.	R/W	0
[2:0]	Reverse	--	--	--

GPIOB4_OBIT IOB4 Bit operation register								Address : 0x5007_0130
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
--								
IOB4_OBIT								

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0
[4]	IOB4_OBIT	IOB4 bit operation. SW writes this register to change GPIOB_OBUF[4] value.	R/W	0
[3:0]	Reverse	--	--	--

GPIOB5_OBIT IOB5 Bit operation register								Address : 0x5007_0134
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
--								
IOB5_OBIT								

Bit	Name	Description	Access	Reset value
[31:6]	--	Reserved	R	0
[5]	IOB5_OBIT	IOB5 bit operation. SW writes this register to change GPIOB_OBUF[5]	R/W	0

Bit	Name	Description	Access	Reset value
		value.		
[4:0]	Reverse	--	--	--

IOFUNC_CTRL0 GPIO Function Control register								Address : 0x5007_0200
31	30	29	28	27	26	25	24	
23	22	21	20	19	18	17	16	
			--					I2C_IOSEL
15	14	13	12	11	10	9	8	
--	SPI0_IOSEL	CCP0_IOSEL	SPIFC_IOSEL	UART_IOSEL		SPI1_IOSEL	IR_IOSEL	
7	6	5	4	3	2	1	0	
--	SPI1_CS_EN	SPI0_CS_EN		--				ICE_EN

Bit	Name	Description				Access	Reset value
[31:18]	--	Reserved				R	0
[17:16]	I2C_IOSEL	I2C IO select when I2C enabled				R/W	0
		I2C_SCK	I2C_IO_SEL=0	I2C_IO_SEL=1	I2C_IO_SEL=2		
		I2C_SDA	PAD IOA0	PAD IOA15	PAD IOA13		
[15]	--	Reserved				R	0
[14]	SPI0_IOSEL	SPI0 IO select when SPI0 enabled.				R/W	0
		SPI0_SCLK	SPI1_IO_SEL =0	SPI1_IO_SEL =1			
		SPI0_SI(MOSI)	PAD IOA9	PAD IOB1			
		SPI0_CS	PAD IOA10	PAD IOB2			
		SPI0_SO(MISO)	PAD IOA11	PAD IOB3			
[13]	CCP0 (CCP0_IOSEL)	CCP0 Supports 4 x Capture IN When CCP_CAP0_EN				R/W	0
		CCP_CAP0_IN	CCGPIO_SEL =0	CCGPIO_SEL =1			
		When CCP_CAP1_EN	CCP_CAP1_IN	CCGPIO_SEL =0	CCGPIO_SEL =1		
		When CCP_CAP2_EN	CCP_CAP2_IN	CCGPIO_SEL =0	CCGPIO_SEL =1		
		When CCP_CAP3_EN	CCP_CAP3_IN	CCGPIO_SEL =0	CCGPIO_SEL =1		

Bit	Name	Description			Access	Reset value
[12]	SPIFC_IOSEL	When SPIFC Enabled:	SPIFC_IO_SEL =0	SPIFC_IO_SEL =1		
		SPI_CS	PAD IOB3	PAD IOB5		
		SPI_CLK	PAD IOB1	PAD IOB1		
		SPI_IO[0] = SI (MOSI)	PAD IOB2	PAD IOB0		
		SPI_IO[1] = SO (MISO)	PAD IOB4	PAD IOB4		
		SPI_IO[2] = WP	PAD IOB5	PAD IOB3		
		SPI_IO[3]= HOLD	PAD IOB0	PAD IOB2		
[11:10]	UART_IOSEL	When UART Enabled:	UART_IO_SEL=0	UART_IO_SEL=1		
		UART_TX	PAD IOA4	PAD IOA13		
		UART_RX	PAD IOA5	PAD IOA14		
			UART_IO_SEL=2	UART_IO_SEL=3		
		UART_TX	PAD IOA22	PAD IOA0		
		UART_RX	PAD IOA23	PAD IOA9		
[9]	SPI1_IOSEL	When SPI1 Enabled:	SPI1_IO_SEL =0	SPI1_IO_SEL =1		
		SPI1_SCLK	PAD IOA0	PAD IOA18		
		SPI1_SI (MOSI)	PAD IOA1	PAD IOA19		
		SPI1_CS	PAD IOA2	PAD IOA20		
		SPI1_SO (MISO)	PAD IOA3	PAD IOA21		
[8]	IR_IOSEL	When IR Enabled: If IR_IO_SEL =0, PAD IOA4 as IR IO Else PAD IOA17as IR IO			R/W	0
[7:6]	-	Reserved			R	0
[5]	SPI1_HWCS_ENA BLE (SPI1_CS_EN)	SPI1_CS_EN 1: SPI1 CS pin control by H/W. 0: By S/W control Note.1: If SPI1_IO_SEL =0, PAD IOA2 is as SPI1_CSN, else PAD IOA20 is as SPI1_CSN. Note.2: Regardless of SPI1_CS_EN being controlled by hardware or software, CS pin (IOA.2 or IOA.20) will be set to output mode; it cannot be configured as input mode.			R/W	1
[4]	SPI0_HWCS_ENA BLE (SPI0_CS_EN)	SPI0_CS_EN 1: SPI0 CS pin control by H/W. 0: By S/W control Note.1: If SPI0_IO_SEL =0, PAD IOA11 is as SPI0_CSN, else PAD IOB3 is as SPI1_CSN. Note.2: Regardless of SPI0_CS_EN being controlled by hardware or software, CS pin (IOA.11 or IOB.3) will be set to output mode; it cannot be configured as input mode.			R/W	1
[3:1]	--	Reserved			R	0
[0]	ICE_ENABLE (ICE_EN)	ICE Enabling bit. When this bit is "1", PAD RESETn as ICE clock and PAD IOA24 as ICE data.			R/W	1



GPCM1F SERIES PROGRAMMING GUIDE

IOFUNC_CTRL1 GPIO IR Control register								Address : 0x5007_0204	
31	30	29	28	27	26	25	24		
				--					
23	22	21	20	19	18	17	16		
				--					
15	14	13	12	11	10	9	8		
				--					
7	6	5	4	3	2	1	0		
IR_POL	IR_MASK	IR_CLK_SEL				IR_DUTY		IR_EN	

Bit	Name	Description				Access	Reset value
[31:8]	--	Reserved				R	0
[7]	IR_POL	IR TX polarity control bit. 0: positive 1: negative				R/W	0
[6]	IR_MAK_ENABLE (IR_MASK)	IR TX mask bit. This bit is used to mask IR TX value. IR TX value is "1" when this bit is set. 0 : disabled, 1: enabled				R/W	1
[5:3]	IR_CLK_SEL	IR TX CLK source selection				R/W	0
		IR_CLK_SEL		IR_CLK_SEL			
		3'b000	timer0	3'b100	ccp1		
		3'b001	timer1	3'b101	cts0		
		3'b010	timer2	3'b110	cts1		
		3'b011	ccp0	3'b111	--		
[2:1]	IR_DUTY	IR TX duty control register				R/W	0
		IR_DUTY	function	IR_DUTY	function		
		2'b00	1/2 duty	2'b10	1/4 duty		
		2'b01	1/3 duty	2'b11	1/5 duty		
[0]	IR_TX_ENABLE (IR_EN)	IR TX enable bit. 0 : disabled, 1: enabled				R/W	0

IOFUNC_CTRL2 GPIO EXT 0~3 Input pins select control register								Address : 0x5007_0208	
31	30	29	28	27	26	25	24		
					EXT3_PINSEL[4:0]				
23	22	21	20	19	18	17	16		
					EXT2_PINSEL[4:0]				
15	14	13	12	11	10	9	8		
					EXT1_PINSEL[4:0]				
7	6	5	4	3	2	1	0		
					EXT0_PINSEL[4:0]				

Bit	Name	Description								Access	Reset value
[31:29]	--	Reserved								R	0
[28:24]	EXT3_PINSEL	For EXTINT3/ CCP EXTCLK3 Key selection								R/W	'h1F
		SEL	KEY	SEL	KEY	SEL	KEY	SEL	KEY		
		5'h0	IOA0	5'h8	IOA8	5'h10	IOA16	5'h18	IOA24		
		5'h1	IOA1	5'h9	IOA9	5'h11	IOA17	5'h19	IOB0		
		5'h2	IOA2	5'hA	IOA10	5'h12	IOA18	5'h1A	IOB1		

Bit	Name	Description								Access	Reset value
		5'h3	IOA3	5'hB	IOA11	5'h13	IOA19	5'h1B	IOB2		
		5'h4	IOA4	5'hC	IOA12	5'h14	IOA20	5'h1C	IOB3		
		5'h5	IOA5	5'hD	IOA13	5'h15	IOA21	5'h1D	IOB4		
		5'h6	IOA6	5'hE	IOA14	5'h16	IOA22	5'h1E	IOB5		
		5'h7	IOA7	5'hF	IOA15	5'h17	IOA23	5'h1F	Disable		
[23:21]	--	Reserved								R	0
[20:16]	EXT2_PINSEL	For EXTINT2/Timer2 ExtCLK/CCP EXTCLK2 Key selection								R/W	'h1F
		SEL	KEY	SEL	KEY	SEL	KEY	SEL	KEY		
		5'h0	IOA0	5'h8	IOA8	5'h10	IOA16	5'h18	IOA24		
		5'h1	IOA1	5'h9	IOA9	5'h11	IOA17	5'h19	IOB0		
		5'h2	IOA2	5'hA	IOA10	5'h12	IOA18	5'h1A	IOB1		
		5'h3	IOA3	5'hB	IOA11	5'h13	IOA19	5'h1B	IOB2		
		5'h4	IOA4	5'hC	IOA12	5'h14	IOA20	5'h1C	IOB3		
		5'h5	IOA5	5'hD	IOA13	5'h15	IOA21	5'h1D	IOB4		
		5'h6	IOA6	5'hE	IOA14	5'h16	IOA22	5'h1E	IOB5		
		5'h7	IOA7	5'hF	IOA15	5'h17	IOA23	5'h1F	Disable		
[15:13]	--	Reserved								R	0
[12:8]	EXT1_PINSEL	For EXTINT1/Timer1 ExtCLK/CCP EXTCLK1 Key selection								R/W	'h1F
		SEL	KEY	SEL	KEY	SEL	KEY	SEL	KEY		
		5'h0	IOA0	5'h8	IOA8	5'h10	IOA16	5'h18	IOA24		
		5'h1	IOA1	5'h9	IOA9	5'h11	IOA17	5'h19	IOB0		
		5'h2	IOA2	5'hA	IOA10	5'h12	IOA18	5'h1A	IOB1		
		5'h3	IOA3	5'hB	IOA11	5'h13	IOA19	5'h1B	IOB2		
		5'h4	IOA4	5'hC	IOA12	5'h14	IOA20	5'h1C	IOB3		
		5'h5	IOA5	5'hD	IOA13	5'h15	IOA21	5'h1D	IOB4		
		5'h6	IOA6	5'hE	IOA14	5'h16	IOA22	5'h1E	IOB5		
		5'h7	IOA7	5'hF	IOA15	5'h17	IOA23	5'h1F	Disable		
[7:5]	--	Reserved								R	0
[3:0]	EXT0_PINSEL	For EXTINT0/Timer0 ExtCLK/CCP0 EXTCLK Key selection								R/W	'h1F
		SEL	KEY	SEL	KEY	SEL	KEY	SEL	KEY		
		5'h0	IOA0	5'h8	IOA8	5'h10	IOA16	5'h18	IOA24		
		5'h1	IOA1	5'h9	IOA9	5'h11	IOA17	5'h19	IOB0		
		5'h2	IOA2	5'hA	IOA10	5'h12	IOA18	5'h1A	IOB1		
		5'h3	IOA3	5'hB	IOA11	5'h13	IOA19	5'h1B	IOB2		
		5'h4	IOA4	5'hC	IOA12	5'h14	IOA20	5'h1C	IOB3		
		5'h5	IOA5	5'hD	IOA13	5'h15	IOA21	5'h1D	IOB4		
		5'h6	IOA6	5'hE	IOA14	5'h16	IOA22	5'h1E	IOB5		
		5'h7	IOA7	5'hF	IOA15	5'h17	IOA23	5'h1F	Disable		

GPIO key-change wake up enable register								Address : 0x5007_0210	
31	30	29	28	27	26	25	24		
KEY_CHG_INT_EN	KEY_CHG_EN[30:24]								
23	22	21	20	19	18	17	16		

KEY_CHG_EN[23:16]							
15	14	13	12	11	10	9	8
KEY_CHG_EN[15:8]							
7	6	5	4	3	2	1	0
KEY_CHG_EN[7:0]							

Bit	Name	Description								Access	Reset value
[31]	INT_ENABLE (KEY_CHG_INT_EN)	key change interrupt Enabling bit.								R/W	0
[30:0]	IOA[24:0] & IOB[5:0] _ENABLE (KEY_CHG_EN)	key change Enabling bit. This register is used to enable PAD key change function KEY_CHG_EN[x] 1: enable key change function								R/W	0
		EN	KEY	EN	KEY	EN	KEY	EN	KEY		
		[0]	IOA0	[8]	IOA8	[16]	IOA16	[24]	IOA24		
		[1]	IOA1	[9]	IOA9	[17]	IOA17	[25]	IOB0		
		[2]	IOA2	[10]	IOA10	[18]	IOA18	[26]	IOB1		
		[3]	IOA3	[11]	IOA11	[19]	IOA19	[27]	IOB2		
		[4]	IOA4	[12]	IOA12	[20]	IOA20	[28]	IOB3		
		[5]	IOA5	[13]	IOA13	[21]	IOA21	[29]	IOB4		
		[6]	IOA6	[14]	IOA14	[22]	IOA22	[30]	IOB5		
		[7]	IOA7	[15]	IOA15	[23]	IOA23				

IOFUNC_STS	GPIO key-change interrupt status register								Address : 0x5007_0214
31	30	29	28	27	26	25	24		
INTF_FLAG				--					
23	22	21	20	19	18	17	16		
				--					
15	14	13	12	11	10	9	8		
				--					
7	6	5	4	3	2	1	0		
				--					

Bit	Name	Description								Access	Reset value
[31]	INTF_FLAG	Key change interrupt status Flag. When a pad has any key change occurred, it will set 1. Write "1" to clear this bit.								R/W	0
[30:0]	-	Reserved								R	0

14. Line-IN ADC (SAR ADC) Controller

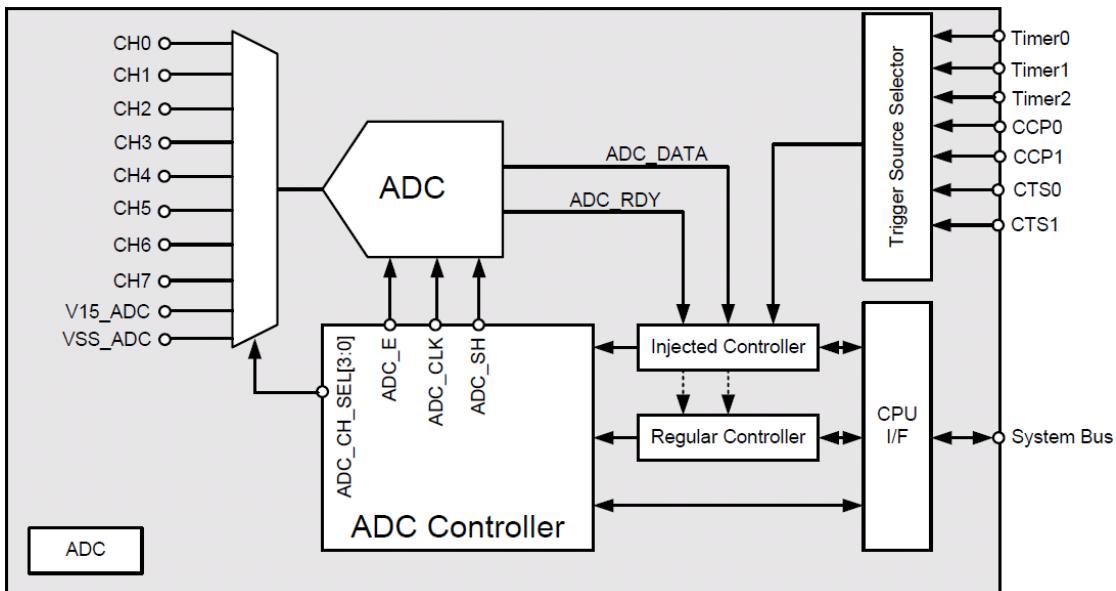
14.1. Introduction

The GPCM1F series has one 12-bit SAR ADC controller. It has up to 8 multiplexed channels allowing it measure signals from eight external and two internal sources. The functionality of 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 with left-aligned or right-aligned. A function of analog watchdog feature allows the application to detect input-voltage. It is used to detect whether the input voltage is outside the user-defined range. The input clock of SAR ADC generated from the PCLK clock that divided by a pre-scaler must not exceed 3MHz.

14.2. Features

- 12-bit resolution
- SAR ADC has up to 8 channels in regular mode and 4 channels in injected mode
- Interrupt generation at end of conversion, end of Injected conversion, and 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: 2.4 V to 5.5 V
- SAR ADC input range: $0V \leq VIN \leq VREF(VREG_IN)$
- DMA request generation during regular channel conversion

14.3. Block Diagram



14.4. Function

14.4.1. SAR ADC on-off control

The SAR ADC controller can be turn-on by setting the SAR_ADC_GCTRL.SAR_ADC_EN bit. After SAR_ADC_GCTRL.SAR_ADC_EN set to 1, SAR_ADC_RDY will go low to indicate SAR ADC is initializing. Until SAR_ADC_RDY becomes 1, then, the user can start using the ADC. In regular mode, conversion starts when SAR_ADC_CTRL.SFT_STR bit is set to 1. In injected mode, conversion starts when one of seven trigger source is triggered. User can stop conversion and putting the

SAR ADC in power down mode by disabling the SAR_ADC_GCTRL.SAR_ADC_EN bit.

14.4.2. Channel Selection

SAR ADC has up to 10 multiplexed channels allowing it 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. The regular channels and their order in the conversion sequence must be selected 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 four conversions. The injected channels and their order in the conversion sequence must be selected 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=0~3) bits.

If the SAR_ADC_REG_SEQ.SEQ0TH_SEL or SAR_ADC_CTRL.INJy_EN (y=0~3) registers are modified during a conversion, the current conversion data will be abnormal.

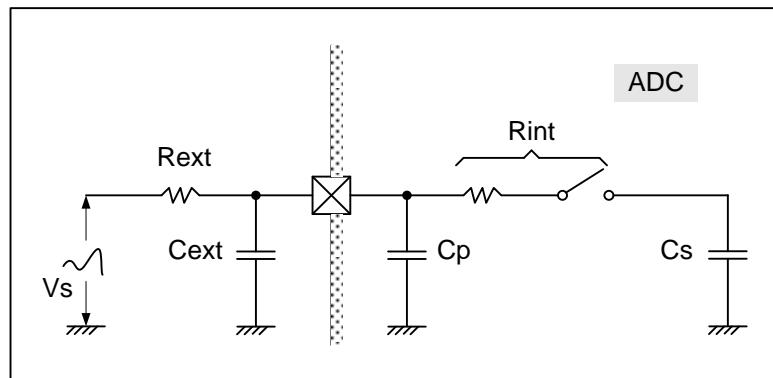
Table shows the channel mapping of SAR ADC channel mapping table

	ADC
Channel #	Channel Mapping
0	CH0 (IOA.17)
1	CH1 (IOA.18)
2	CH2 (IOA.19)
3	CH3 (IOA.20)
4	CH4 (IOA.21)
5	CH5 (IOA.22)
6	CH6 (IOA.23)
7	CH7 (IOA.24)
8	V15_ADC
9	VSS_ADC

14.4.3. Sampling path

The SAR ADC structure of the GPCM1F 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 internal switch will be molding to Rint. An analog voltage source Vs that should be converted. The value of Rext, Cext, Rint and Cs are strongly defining the required length of the sample cycles.

The capacitance value of switched capacitor Cs is approximately 10pf. The parasitic resistance value of Rint is range in $200\Omega \sim 14K\Omega$ 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. Figure shows a single channel model of ADC.



A single channel model of ADC.

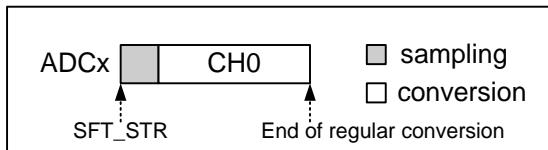
14.4.4. Regulation conversion mode

Regular conversion with loop function disabled

In regular conversion mode, the SAR ADC does one conversion. This mode is started by setting the SAR_ADC_CTRL.SFT_STR only.

Once the selected channel completes the conversion, there are two situations that will be considered.

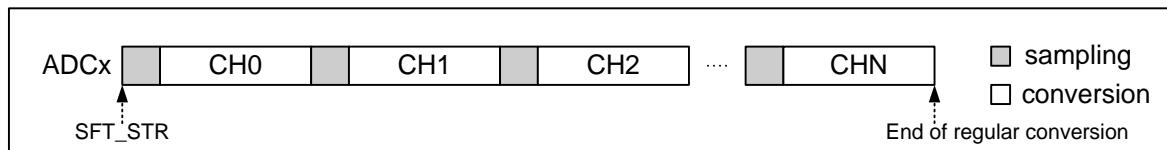
- If a regular channel was converted:
 1. The converted data is stored in the 16-bit SAR_ADC_REG_DATA register
 2. The SAR_ADC_CTRL.REG_INTF flag is set
 3. An interrupt is generated if the SAR_ADC_CTRL.REG_INTE is set.
 4. User clear SAR_ADC_CTRL.REG_INTF flag by write SAR_ADC_CTRL.REG_INTF to 1.



A regular conversion

- If N regular channels were 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
3. An interrupt is generated if the SAR_ADC_CTRL.REG_INTE is set.
4. User clear SAR_ADC_CTRL.REG_INTF flag by writing SAR_ADC_CTRL.REG_INTF to 1.
5. Wait some SAR_ADC_CLK cycles that were set in SAR_ADC_CTRL.REG_SEQ_GAP, executing next conversion until all of the regular channels are finished.



N regular conversion

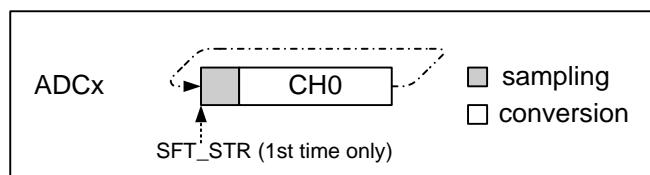
Regular conversion with loop function enabled

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. Before start SAR ADC conversion, user must set SAR_ADC_CTRL.LOOP_EN to 1. The data conversion will be triggered by hardware automatically until software disable SAR ADC or set SAR_ADC_CTRL.LOOP_EN to 0.

Once the selected channel completes the conversion, there are two situations that will be considered.

- If only one regular channel was selected:

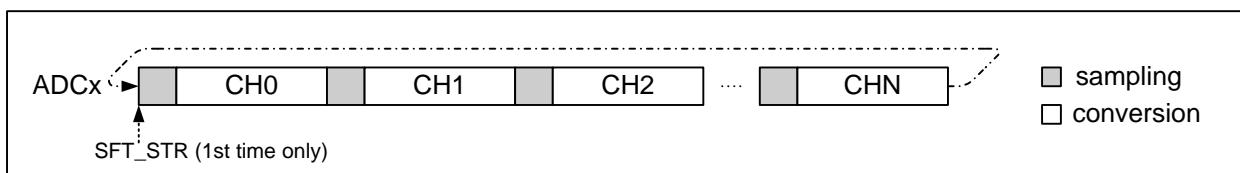
1. The converted data is stored in the 16-bit SAR_ADC_REG_DATA register
2. The SAR_ADC_CTRL.REG_INTF flag is set
3. An interrupt is generated if the SAR_ADC_CTRL.REG_INTE is set.
4. User clear SAR_ADC_CTRL.REG_INTF flag by writing SAR_ADC_CTRL.REG_INTF to 1.
5. Wait some SAR_ADC_CLK cycles that were set in SAR_ADC_CTRL.REG_SEQ_GAP, executing next iteration until software disable SAR ADC or set SAR_ADC_CTRL.LOOP_EN to 0.



A regular conversion with loop function enable

- If N regular channel were 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
3. An interrupt is generated if the SAR_ADC_CTRL.REG_INTE is set.
4. User clear SAR_ADC_CTRL.REG_INTF flag by writing SAR_ADC_CTRL.REG_INTF to 1.
5. Wait some SAR_ADC_CLK cycles that were set in SAR_ADC_CTRL.REG_SEQ_GAP, executing next conversion until all of the regular channel are finished.
6. Wait some SAR_ADC_CLK cycles that were set in SAR_ADC_CTRL.REG_SEQ_GAP, executing next iteration until software disable SAR ADC or set SAR_ADC_CTRL.LOOP_EN to 0.

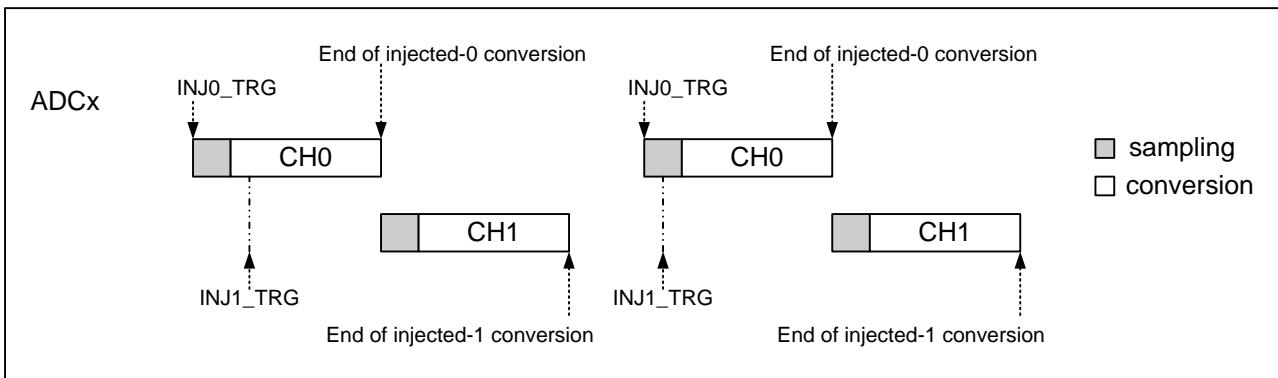


N regular conversion with loop function enable

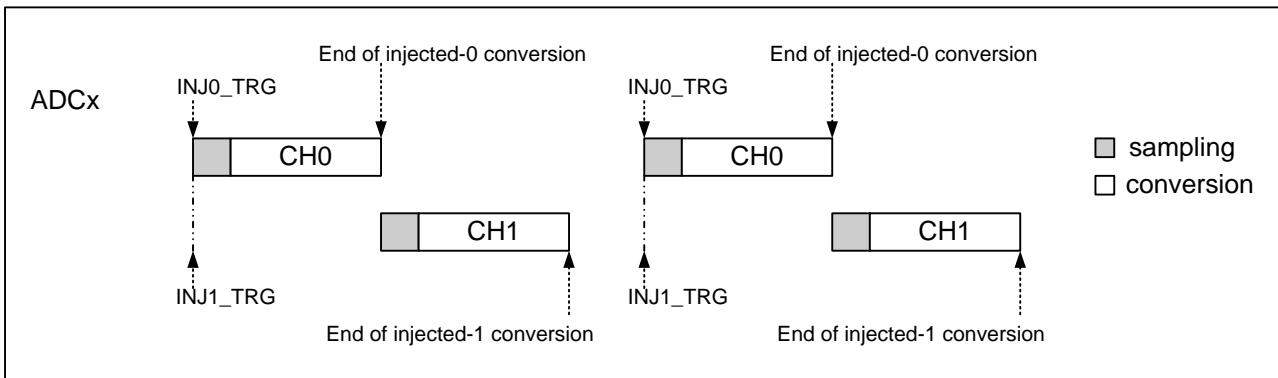
14.4.5. Injected conversion mode

To use triggered injection, one of SAR_ADC_CTRL.INJy_EN(y=0 ~ 3) must be set. Each injected channel supports 8 trigger sources. User can select a trigger source by setting SAR_ADC_CTRL.INJy_TRG_SEL(y=0~3).The priority of injected channel is INJ0 > INJ1 > INJ2 > INJ3. In addition, the priority of any injection channel is always higher than the regular channel. Below figures explain how injected mode works.

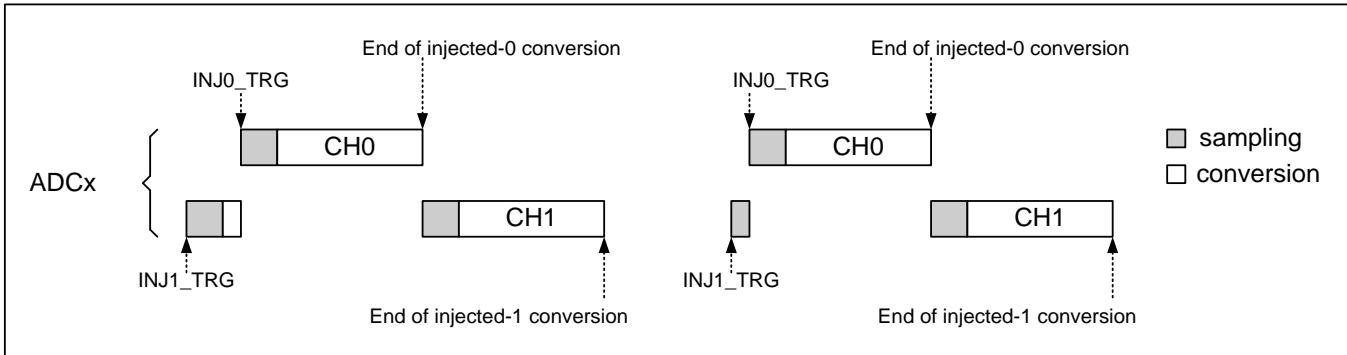
1. Start conversion of an injected channel by a trigger source.
2. The converted data is stored in the 16-bit SAR_ADC_INJy_DATA register
3. The SAR_ADC_CTRL.INJy_INTF flag is set
4. User clear SAR_ADC_CTRL.INJy_INTF flag by writing SAR_ADC_CTRL.INJy_INTF to 1.



Injected conversion mode (INJ0 leads INJ1)



Injected conversion mode (INJ0 and INJ1 at the same time)



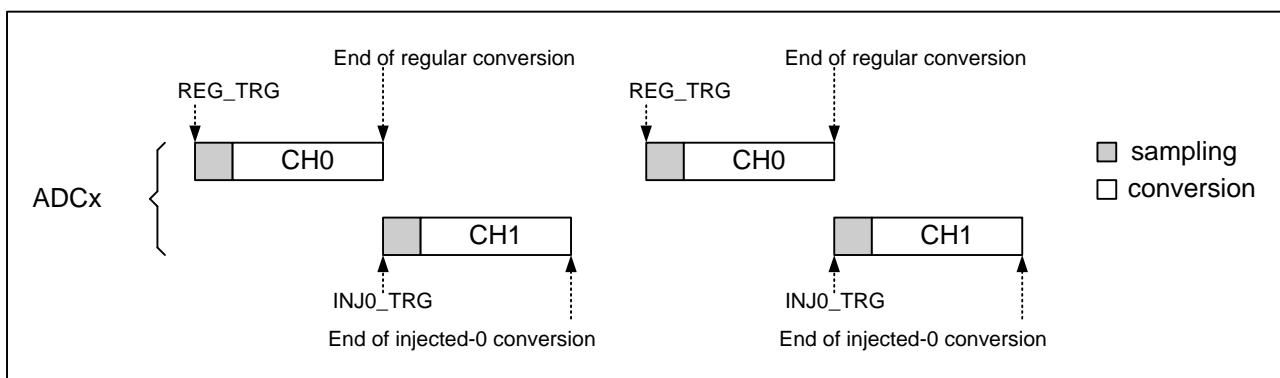
Injected conversion mode (INJ1 leads INJ0)

14.4.6. Combined regular/injected simultaneous mode

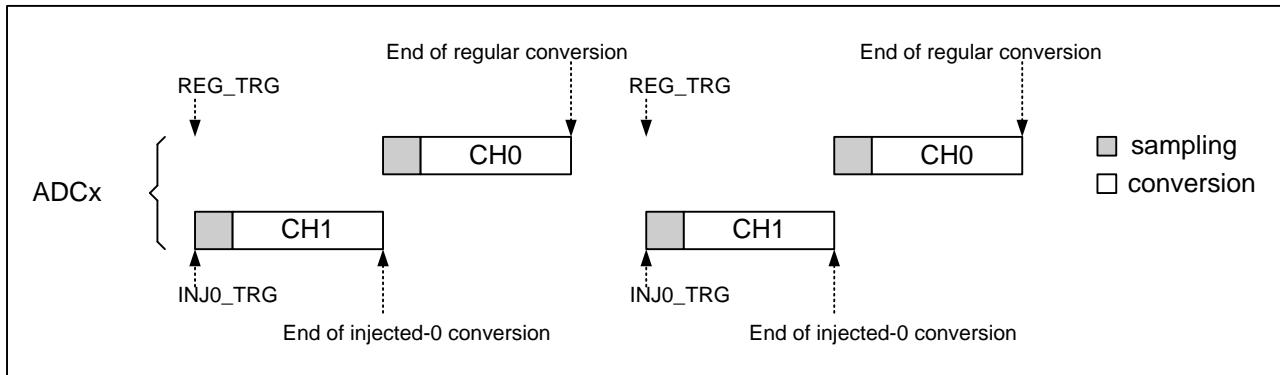
In combined regular and injected simultaneous mode, it is possible to interrupt regular channel simultaneous conversion to start injected trigger conversion of an injected channel. Below Figures show the behavior of an injected channel trigger interrupts a regular simultaneous conversion. The injected channel conversion is immediately started after the injected event triggers. If regular conversion is already running, it will be suspended. And regular channel resumed synchronously at the end of the injected conversion. Below figures how regular and injected mode works.

- If regular channel first converted:
 1. Regular channel is running.
 2. If an injected trigger occurs during the regular channel conversion, the regular conversion is suspended and the injected channel is converted.
 3. Then, the regular channel conversion is resumed at the end of the injected conversion.
- If injected channel first converted:

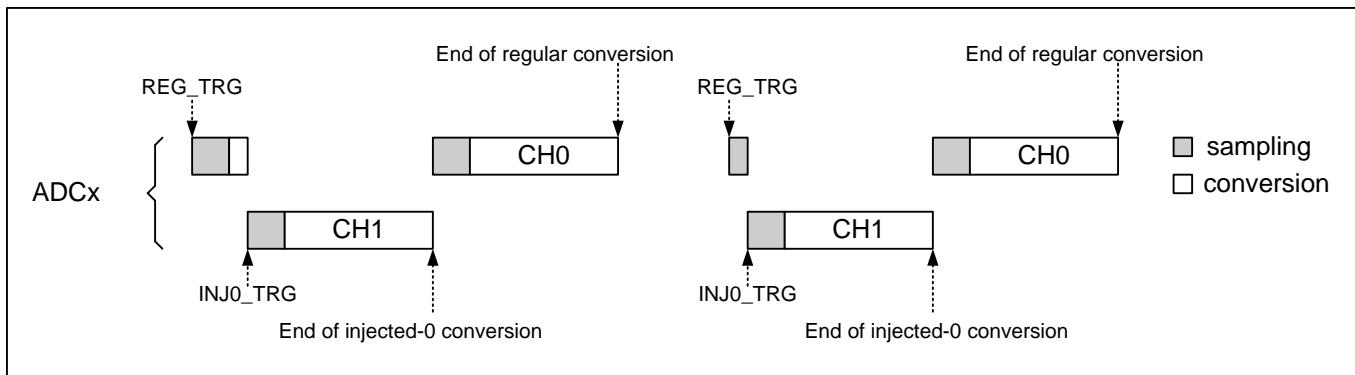
1. Injected channel is running.
2. If a regular trigger occurs during the injected channel conversion, the regular conversion is started at the end of the injected conversion.



Combined regular/injected conversion mode (regular leads INJ0)



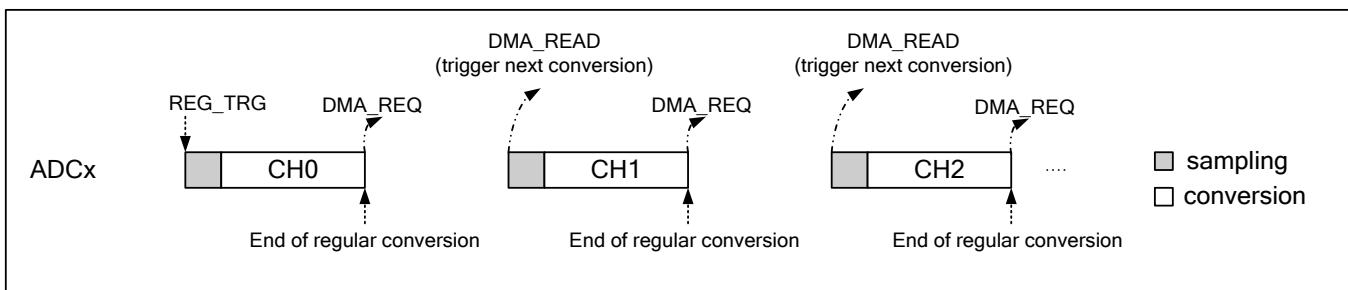
Combined regular/injected conversion mode (regular and INJ0 arrive at the same time)



Combined regular/injected conversion mode (INJ0 interrupts regular)

14.4.7. Regular conversion with DMA

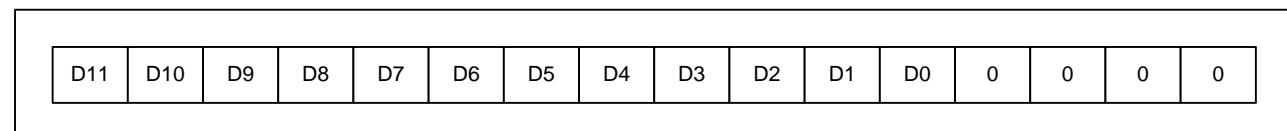
In regular mode, since converted data are stored in SAR_ADC_REG_DATA, user can read data out by CPU access. In addition, DMA access is another access path. The DMA access is applied for conversion of more than one regular channel. This prevents the data stored in SAR_ADC_REG_DATA from being lost or overwritten. Please note that the DMA function does not support injected mode. The figure 10-13 shows an example of the regular conversion with DMA function.



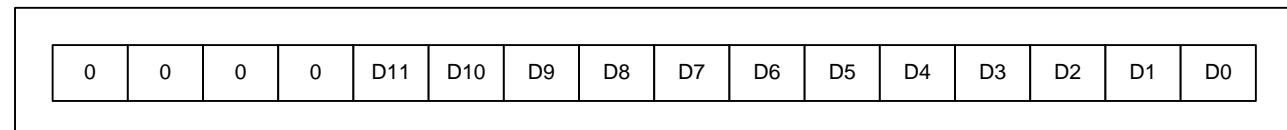
The regular conversion with DMA function

14.4.8. Data alignment

In GPCM1F series, SAR ADC provides two data alignment type; user can set data alignment with left-aligned or right-aligned. **SAR_ADC_CTRL.DAT_ALIGN** bit selects the alignment of data stored after conversion. The following figures show some examples of the different data alignment.



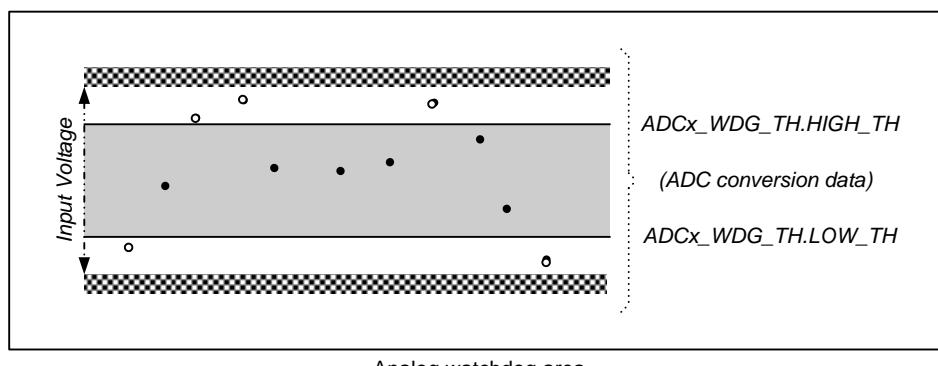
Left alignment of SAR ADC data (**SAR_ADC_CTRL.DAT_ALIGN=0**)



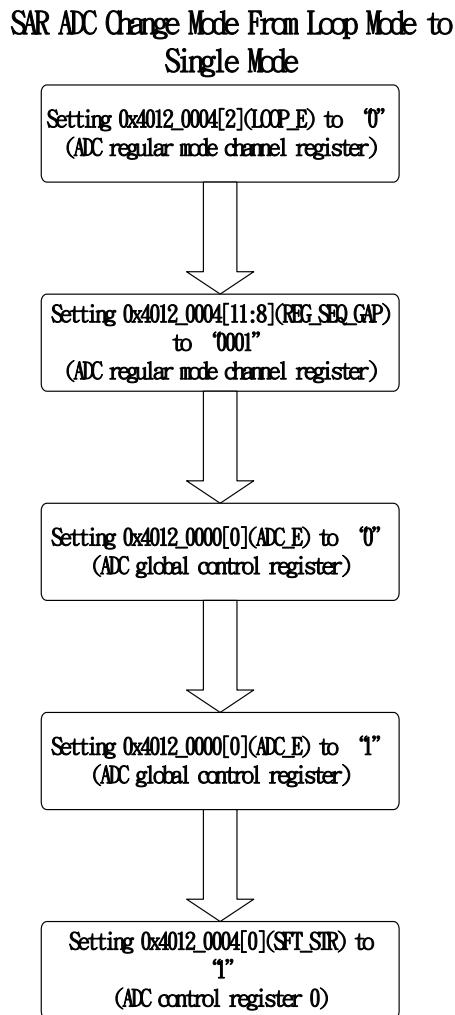
Right alignment of SAR ADC data (**SAR_ADC_CTRL.DAT_ALIGN=1**)

14.4.9. Analog watchdog

In GPCM1F 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** are set if the analog voltage converted by the SAR ADC is below **SAR_ADC_WDG_TH.LOW_TH** or above **SAR_ADC_WDG_TH.HIGH_TH**. 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_TH** setting. The comparison is done before the data alignment.



14.4.10. SAR ADC Program Example



14.5. Register Description

Register map

Base Address : 0x4012_0000				
Name	Description	Address	Access	Reset value
SAR_ADC_GCTRL	SAR ADC Global Control Register	0x4012_0000	R/W	0x0000_0F02
SAR_ADC_CTRL	SAR ADC Control Register	0x4012_0004	R/W	0x8000_0000
SAR_ADC_STS	SAR ADC Status Register	0x4012_0008	R/W	0x0000_0000
SAR_ADC_SMP_SEL0	SAR ADC Sample Time Control Register 0	0x4012_000C	R/W	0x0000_0000
SAR_ADC_SMP_SEL1	SAR ADC Sample Time Control Register 1	0x4012_0010	R/W	0x0000_0000
SAR_ADC_REG_COV	SAR ADC Regular Mode Conversion Channel Register	0x4012_0014	R/W	0x0000_0000
SAR_ADC_REG_SEQ	SAR ADC Regular Sequence Register	0x4012_0018	R/W	0x0000_0000
SAR_ADC_INJ_SEQ	SAR ADC Injected Sequence Register	0x4012_001C	R/W	0x0000_0000
SAR_ADC_WDG_TH	SAR ADC Watchdog Threshold Register	0x4012_0020	R/W	0x0000_0000
SAR_ADC_REG_DATA	SAR ADC Regular Data Register	0x4012_0024	R/W	0x0000_0000
SAR_ADC_INJ0_DATA	SAR ADC Injected0 Data Register	0x4012_0028	R/W	0x0000_0000
SAR_ADC_INJ1_DATA	SAR ADC Injected1 Data Register	0x4012_002C	R/W	0x0000_0000

Base Address : 0x4012_0000						
Name	Description			Address	Access	Reset value
SAR_ADC_INJ2_DATA	SAR ADC Injected2 Data Register			0x4012_0030	R/W	0x0000_0000
SAR_ADC_INJ3_DATA	SAR ADC Injected3 Data Register			0x4012_0034	R/W	0x0000_0000

Registers Function

SAR_ADC_GCTRL SAR ADC Global Control Register Address : 0x4012_0000

31	30	29	28	27	26	25	24
--	A_WDGJ_INT_E	A_WDGR_INT_E	INJ3INTE	INJ2_INTE	INJ1_INTE	INJ0_INTE	REG_INTE
23	22	21	20	19	18	17	16
				--			
15	14	13	12	11	10	9	8
--	--	--		ADC_CLK_SEL[4:0]			
7	6	5	4	3	2	1	0
--	--	--	DMA_REQE	--	DAT_ALIGN	ADC_FMT_SEL	SAR_ADC_EN

Bit	Name	Description	Access	Reset value
[31]	--	Reserved	R	0
[30]	ANALOG_WDGJ_INT_E (A_WDGJ_INTEN)	Analog Watchdog interrupt enable on Injected channel 0 = disabled 1 = enabled	R/W	0x0
[29]	ANALOG_WDGR_INT_E (A_WDGR_INTEN)	Analog Watchdog interrupt enable on regular channel 0 = disabled 1 = enabled	R/W	0x0
[28]	INJ3_INT_ENABLE (INJ3_INTEN)	Injected conversion interrupt enable of channel 3 0 = disable 1 = enable	R/W	0x0
[27]	INJ2_INT_ENABLE (INJ2_INTEN)	Injected conversion interrupt enable of channel 2 0 = disabled 1 = enabled	R/W	0x0
[26]	INJ1_INT_ENABLE (INJ1_INTEN)	Injected conversion interrupt enable of channel 1 0 = disabled 1 = enabled	R/W	0x0
[25]	INJ0_INT_ENABLE (INJ0_INTEN)	Injected conversion interrupt enable of channel 0 0 = disabled 1 = enabled	R/W	0x0
[24]	REG_MODE_INT_ENAB LE (REG_INTEN)	Regular mode interrupt enable 0 = disabled 1 = enabled	R/W	0x0
[23:13]	--	Reserved	R	0
[12:8]	ADC_CLK_SEL	SAR ADC clock select bits 0 : PCLK/2 1 : PCLK/4 2 : PCLK/6 3 : PCLK/8	R/W	0xF

Bit	Name	Description	Access	Reset value
		4 : PCLK/10 . . . 31:PCLK/((SAR_ADC_CLK_SEL+1)*2)		
[7:5]	--	Reserved	R	0
[4]	DMA_ENABLE (DMA_REQE)	DMA request enable 0 = DMA mode disabled 1 = DMA mode enabled	R/W	0x0
[3]	--	Reserved	R	0
[2]	DAT_ALIGN	SAR ADC output data alignment 0 = left aligned 1 = right aligned	R/W	0x0
[1]	ADC_FMT_SEL	SAR ADC data sign enable register 0: SAR ADC data is unsigned data 1: SAR ADC data is signed data	R/W	0x1
[0]	ADC_ENABLE (SAR_ADC_EN)	SAR ADC analog block enable 0 = disabled 1 = enabled Note: User should set this bit enable before running SAR ADC function.	R/W	0x0

SAR_ADC_CTRL SAR ADC Control Register Address : 0x4012_0004

31	30	29	28	27	26	25	24
INJ3_TRG_SEL			INJ3_EN	INJ2_TRG_SEL			INJ2_EN
23	22	21	20	19	18	17	16
INJ1_TRG_SEL			INJ1_EN	INJ0_TRG_SEL			INJ0_EN
15	14	13	12	11	10	9	8
A_WDGJ_EN	--	--	--	REG_SEQ_GAP			
7	6	5	4	3	2	1	0
REG_CH_NUM				A_WDGR_E	LOOP_EN	REG_EN	SFT_STR

Bit	Name	Description	Access	Reset value																
[31:29]	INJ3_TRG_SEL	Injection channel3 trigger source select bits <table border="1"> <tr> <td>INJ3_TRG_SEL[2:0]</td> <td>Injection channel3 trigger source</td> </tr> <tr> <td>000</td> <td>Timer0 interrupt</td> </tr> <tr> <td>001</td> <td>Timer1 interrupt</td> </tr> <tr> <td>010</td> <td>Timer2 interrupt</td> </tr> <tr> <td>011</td> <td>CCP0 timer interrupt</td> </tr> <tr> <td>100</td> <td>CCP1 timer interrupt</td> </tr> <tr> <td>101</td> <td>CTS timer0 interrupt</td> </tr> <tr> <td>110</td> <td>CTS timer1 interrupt</td> </tr> </table>	INJ3_TRG_SEL[2:0]	Injection channel3 trigger source	000	Timer0 interrupt	001	Timer1 interrupt	010	Timer2 interrupt	011	CCP0 timer interrupt	100	CCP1 timer interrupt	101	CTS timer0 interrupt	110	CTS timer1 interrupt	R/W	0x0
INJ3_TRG_SEL[2:0]	Injection channel3 trigger source																			
000	Timer0 interrupt																			
001	Timer1 interrupt																			
010	Timer2 interrupt																			
011	CCP0 timer interrupt																			
100	CCP1 timer interrupt																			
101	CTS timer0 interrupt																			
110	CTS timer1 interrupt																			
[28]	INJ3_ENABLE (INJ3_EN)	Injection channel 3 enable bit 0 = disabled 1 = enabled	R/W	0x00																

Bit	Name	Description	Access	Reset value																
		Note: Before using injection sequence, user must set SAR_ADC_CTRL.INJ3_EN to 1 first. Then, set ADCx and wait hardware trigger to start injection sequence.																		
[27:25]	INJ2_TRG_SEL	<p>Injection channel2 trigger source select bits</p> <table border="1"> <tr><td>INJ2_TRG_SEL[2:0]</td><td>Injection channel2 trigger source</td></tr> <tr><td>000</td><td>Timer0 interrupt</td></tr> <tr><td>001</td><td>Timer1 interrupt</td></tr> <tr><td>010</td><td>Timer2 interrupt</td></tr> <tr><td>011</td><td>CCP0 timer interrupt</td></tr> <tr><td>100</td><td>CCP1 timer interrupt</td></tr> <tr><td>101</td><td>CTS timer0 interrupt</td></tr> <tr><td>110</td><td>CTS timer1 interrupt</td></tr> </table>	INJ2_TRG_SEL[2:0]	Injection channel2 trigger source	000	Timer0 interrupt	001	Timer1 interrupt	010	Timer2 interrupt	011	CCP0 timer interrupt	100	CCP1 timer interrupt	101	CTS timer0 interrupt	110	CTS timer1 interrupt	R/W	0x0
INJ2_TRG_SEL[2:0]	Injection channel2 trigger source																			
000	Timer0 interrupt																			
001	Timer1 interrupt																			
010	Timer2 interrupt																			
011	CCP0 timer interrupt																			
100	CCP1 timer interrupt																			
101	CTS timer0 interrupt																			
110	CTS timer1 interrupt																			
[24]	INJ2_ENABLE (INJ2_EN)	<p>Injection channel 2 enable bit</p> <p>0 = disabled 1 = enabled</p> <p>Note: Before using injection sequence, user must set SAR_ADC_CTRL.INJ2_EN to 1 first. Then, set ADCx and wait hardware trigger to start injection sequence.</p>	R/W	0x00																
[23:21]	INJ1_TRG_SEL	<p>Injection channel1 trigger source select bits</p> <table border="1"> <tr><td>INJ1_TRG_SEL[2:0]</td><td>Injection channel1 trigger source</td></tr> <tr><td>000</td><td>Timer0 interrupt</td></tr> <tr><td>001</td><td>Timer1 interrupt</td></tr> <tr><td>010</td><td>Timer2 interrupt</td></tr> <tr><td>011</td><td>CCP0 timer interrupt</td></tr> <tr><td>100</td><td>CCP1 timer interrupt</td></tr> <tr><td>101</td><td>CTS timer0 interrupt</td></tr> <tr><td>110</td><td>CTS timer1 interrupt</td></tr> </table>	INJ1_TRG_SEL[2:0]	Injection channel1 trigger source	000	Timer0 interrupt	001	Timer1 interrupt	010	Timer2 interrupt	011	CCP0 timer interrupt	100	CCP1 timer interrupt	101	CTS timer0 interrupt	110	CTS timer1 interrupt	R/W	0x0
INJ1_TRG_SEL[2:0]	Injection channel1 trigger source																			
000	Timer0 interrupt																			
001	Timer1 interrupt																			
010	Timer2 interrupt																			
011	CCP0 timer interrupt																			
100	CCP1 timer interrupt																			
101	CTS timer0 interrupt																			
110	CTS timer1 interrupt																			
[20]	INJ1_ENABLE (INJ1_EN)	<p>Injection channel 1 enable bit</p> <p>0 = disabled 1 = enabled</p> <p>Note: Before using injection sequence, user must set SAR_ADC_CTRL.INJ1_EN to 1 first. Then, set ADCx and wait hardware trigger to start injection sequence.</p>	R/W	0x00																
[19:17]	INJ0_TRG_SEL	<p>Injection channel0 trigger source select bits</p> <table border="1"> <tr><td>INJ0_TRG_SEL[2:0]</td><td>Injection channel0 trigger source</td></tr> <tr><td>000</td><td>Timer0 interrupt</td></tr> <tr><td>001</td><td>Timer1 interrupt</td></tr> <tr><td>010</td><td>Timer2 interrupt</td></tr> <tr><td>011</td><td>CCP0 timer interrupt</td></tr> <tr><td>100</td><td>CCP1 timer interrupt</td></tr> <tr><td>101</td><td>CTS timer0 interrupt</td></tr> <tr><td>110</td><td>CTS timer1 interrupt</td></tr> </table>	INJ0_TRG_SEL[2:0]	Injection channel0 trigger source	000	Timer0 interrupt	001	Timer1 interrupt	010	Timer2 interrupt	011	CCP0 timer interrupt	100	CCP1 timer interrupt	101	CTS timer0 interrupt	110	CTS timer1 interrupt	R/W	0x0
INJ0_TRG_SEL[2:0]	Injection channel0 trigger source																			
000	Timer0 interrupt																			
001	Timer1 interrupt																			
010	Timer2 interrupt																			
011	CCP0 timer interrupt																			
100	CCP1 timer interrupt																			
101	CTS timer0 interrupt																			
110	CTS timer1 interrupt																			
[16]	INJ0_ENABLE	Injection channel 0 enable bit	R/W	0x00																

Bit	Name	Description	Access	Reset value																																				
	(INJ0_EN)	0 = disabled 1 = enabled Note: Before using injection sequence, user must set SAR_ADC_CTRL.INJ0_EN to 1 first. Then, set ADCx and wait hardware trigger to start injection sequence.																																						
[15]	ANALOG_WDGJ_ENAB LE (A_WDGJ_EN)	Analog Watchdog enable on Injected channel 0 = disabled 1 = enabled																																						
[14:12]	--	Reserved	R	0x0																																				
[11:8]	REG_SEQ_GAP_SEL (REG_SEQ_GAP)	Regular sequence gap select bits REG_SEQ_GAP[3] is set to "0", regular sequence gap interval comes from internal counter. REG_SEQ_GAP[3] is set to "1", regular sequence gap interval comes from timer. <table border="1"> <tr> <td>REG_SEQ_GAP[3:0]</td> <td>Cycles of ADCx clock/Trigger source</td> </tr> <tr> <td>0000</td> <td>0</td> </tr> <tr> <td>0001</td> <td>1</td> </tr> <tr> <td>0010</td> <td>2</td> </tr> <tr> <td>0011</td> <td>3</td> </tr> <tr> <td>0100</td> <td>4</td> </tr> <tr> <td>0101</td> <td>5</td> </tr> <tr> <td>0110</td> <td>6</td> </tr> <tr> <td>0111</td> <td>7</td> </tr> </table> <table border="1"> <tr> <td>REG_SEQ_GAP[3:0]</td> <td>Trigger source</td> </tr> <tr> <td>1000</td> <td>Timer 0</td> </tr> <tr> <td>1001</td> <td>Timer 1</td> </tr> <tr> <td>1010</td> <td>Timer 2</td> </tr> <tr> <td>1011</td> <td>CCP0 timer Trigger</td> </tr> <tr> <td>1100</td> <td>CCP1 timer Trigger</td> </tr> <tr> <td>1101</td> <td>CTS timer0 interrupt</td> </tr> <tr> <td>1110</td> <td>CTS timer1 interrupt</td> </tr> <tr> <td>1111</td> <td>Manual Trigger</td> </tr> </table>	REG_SEQ_GAP[3:0]	Cycles of ADCx clock/Trigger source	0000	0	0001	1	0010	2	0011	3	0100	4	0101	5	0110	6	0111	7	REG_SEQ_GAP[3:0]	Trigger source	1000	Timer 0	1001	Timer 1	1010	Timer 2	1011	CCP0 timer Trigger	1100	CCP1 timer Trigger	1101	CTS timer0 interrupt	1110	CTS timer1 interrupt	1111	Manual Trigger	R/W	0x0
REG_SEQ_GAP[3:0]	Cycles of ADCx clock/Trigger source																																							
0000	0																																							
0001	1																																							
0010	2																																							
0011	3																																							
0100	4																																							
0101	5																																							
0110	6																																							
0111	7																																							
REG_SEQ_GAP[3:0]	Trigger source																																							
1000	Timer 0																																							
1001	Timer 1																																							
1010	Timer 2																																							
1011	CCP0 timer Trigger																																							
1100	CCP1 timer Trigger																																							
1101	CTS timer0 interrupt																																							
1110	CTS timer1 interrupt																																							
1111	Manual Trigger																																							
[7:4]	REG_CH_NUM	Channel number during regular sequence Note: The physical channel number is SAR_ADC_CTRL.REG_CH_NUM + 1. In addition, the scan sequence is always being started from sequence-0.	R/W	0x0																																				
[3]	ANALOG_WDGR_ENAB LE (A_WDGR_E)	Analog Watchdog enable on regular channel 0 = disabled 1 = enabled	R/W	0x0																																				
[2]	LOOP_ENABLE (LOOP_EN)	SAR ADC loop scan enable bit 0 = single conversion mode 1 = loop scan conversion mode Note: This bit is available in regular mode only	R/W	0x0																																				
[1]	REG_ENABLE	Regular mode enable bit	R/W	0x0																																				

Bit	Name	Description	Access	Reset value
	(REG_EN)	0 = disabled 1 = enabled Note: Before using regular sequence, user must set SAR_ADC_CTRL.REG_EN to 1 first. Then, set SAR_ADC_CTRL.SFT_STR to 1 to start regular sequence.		
[0]	SFT_START (SFT_STR)	Software start bit 0 = disabled 1 = enabled (start conversion of regular channel) Note: This bit is available in regular mode only. This bit will be cleared by hardware, when LOOP_EN is set to 0 and all of regular scan were be finished. If LOOP_EN is set to 1, then signal be cleared by software.	R/W	0x0

SAR_ADC_STS SAR ADC Status Register								Address : 0x4012_0008
31	30	29	28	27	26	25	24	
--	A_WDGJ_INTF	A_WDGR_INTF	INJ3_INTF	INJ2_INTF	INJ1_INTF	INJ0_INTF	REG_INTF	
23	22	21	20	19	18	17	16	
--	--	--	--	--	--	--	--	
15	14	13	12	11	10	9	8	
--	--	--	--	--	--	--	--	
7	6	5	4	3	2	1	0	
--	--	--	--	--	--	--	--	SAR_ADC_R DY

Bit	Name	Description	Access	Reset value
[31]	--	Reserved	R	0x0
[30]	ANALOG_WDGJ_INT_F LAG (A_WDGJ_INTF)	Analog Watchdog interrupt flag on Injected channel Read : 0 = no analog watchdog occurred 1 = analog watchdog occurred Write : 0 = no effect 1 = clear this flag	R/W	0x0
[29]	ANALOG_WDGR_INT_F LAG (A_WDGR_INTF)	Analog Watchdog interrupt flag on regular channel Read : 0 = no analog watchdog occurred 1 = analog watchdog occurred Write : 0 = no effect 1 = clear this flag	R/W	0x0
[28]	INJ3_INT_FLAG (INJ3_INTF)	Injected conversion interrupt flag of channel 3 Read : 0 = no injected conversion has been finished 1 = injected conversion has finished Write : 0 = no effect	R/W	0x0

Bit	Name	Description	Access	Reset value
		1 = clear this flag		
[27]	INJ2_INT_FLAG (INJ2_INTF)	Injected conversion interrupt flag of channel 2 Read : 0 = no injected conversion has been finished 1 = injected conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[26]	INJ1_INT_FLAG (INJ1_INTF)	Injected conversion interrupt flag of channel 1 Read : 0 = no injected conversion has been finished 1 = injected conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[25]	INJ0_INT_FLAG (INJ0_INTF)	Injected conversion interrupt flag of channel 0 Read : 0 = no injected conversion has been finished 1 = injected conversion has been finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[24]	REG_MODE_INT_FLAG (REG_INTF)	Regular conversion interrupt flag Read : 0 = no regular conversion has been finished 1 = regular conversion has been finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[23:1]	--	Reserved	R	0x0
[0]	ADC_RDY_FLAG (SAR_ADC_RDY)	SAR ADC data ready flag 0: Conversion has not been completed by ADC. 1: Conversion has been completed by ADC.	R	0x0

SAR_ADC_SMP_SEL0 SAR ADC Sample Time Control Register 0
Address : 0x4012_000C

31	30	29	28	27	26	25	24
--	CH7_SMP_SEL				--	CH6_SMP_SEL	
23	22	21	20	19	18	17	16
--	CH5_SMP_SEL				--	CH4_SMP_SEL	
15	14	13	12	11	10	9	8
--	CH3_SMP_SEL				--	CH2_SMP_SEL	
7	6	5	4	3	2	1	0
--	CH1_SMP_SEL				--	CH0_SMP_SEL	

Bit	Name	Description	Access	Reset value
[31]	--	Reserved	R	0
[30:28]	CH7_SMP_SEL	Channel7 sample time selection	R/W	0x0
		CH7_SMP_SEL[2:0] Cycles of ADCx clock		
		000 1		
		001 2		
		010 4		
		011 8		
		100 16		
		101 32		
		110 48		
		111 64		
[27]	--	Reserved	R	0x0
[26:24]	CH6_SMP_SEL	Channel6 sample time selection	R/W	0x0
		CH6_SMP_SEL[2:0] Cycles of ADCx clock		
		000 1		
		001 2		
		010 4		
		011 8		
		100 16		
		101 32		
		110 48		
		111 64		
[23]	--	Reserved	R	0x0
[22:20]	CH5_SMP_SEL	Channel5 sample time selection	R/W	0x0
		CH5_SMP_SEL[2:0] Cycles of ADCx clock		
		000 1		
		001 2		
		010 4		
		011 8		
		100 16		
		101 32		
		110 48		
		111 64		
[19]	--	Reserved	R	0x0
[18:16]	CH4_SMP_SEL	Channel4 sample time selection	R/W	0x0
		CH4_SMP_SEL[2:0] Cycles of ADCx clock		
		000 1		
		001 2		
		010 4		
		011 8		
		100 16		
		101 32		
		110 48		
		111 64		

Bit	Name	Description		Access	Reset value
[15]	--	Reserved		R	0x0
[14:12]	CH3_SMP_SEL	Channel3 sample time selection	CH3_SMP_SEL[2:0] Cycles of ADCx clock	R/W	0x0
		000	1		
		001	2		
		010	4		
		011	8		
		100	16		
		101	32		
		110	48		
		111	64		
		Reserved			
[10:8]	CH2_SMP_SEL	Channel2 sample time selection	CH2_SMP_SEL[2:0] Cycles of ADCx clock	R/W	0x0
		000	1		
		001	2		
		010	4		
		011	8		
		100	16		
		101	32		
		110	48		
		111	64		
		Reserved			
[6:4]	CH1_SMP_SEL	Channel1 sample time selection	CH1_SMP_SEL[2:0] Cycles of ADCx clock	R/W	0x0
		000	1		
		001	2		
		010	4		
		011	8		
		100	16		
		101	32		
		110	48		
		111	64		
		Reserved			
[3]	CH0_SMP_SEL	Channel0 sample time selection	CH0_SMP_SEL[2:0] Cycles of ADCx clock	R/W	0x0
		000	1		
		001	2		
		010	4		
		011	8		
		100	16		
		101	32		
		110	48		
		111	64		
		Reserved			



GPCM1F SERIES PROGRAMMING GUIDE

SAR_ADC_SMP_SEL1 SAR ADC Sample Time Control Register 1

Address : 0x4012_0010

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--	CH9_SMP_SEL			--	CH8_SMP_SEL		

Bit	Name	Description		Access	Reset value
[31:7]	--	Reserved		R	0x0
[6:4]	CH9_SMP_SEL	Channel9 sample time selection	CH9_SMP_SEL[2:0] Cycles of ADCx clock	R/W	0x0
		000	1		
		001	2		
		010	4		
		011	8		
		100	16		
		101	32		
		110	48		
		111	64		
[3]	--	Reserved		R	0x0
[2:0]	CH8_SMP_SEL	Channel8 sample time selection	CH8_SMP_SEL[2:0] Cycles of ADCx clock	R/W	0x0
		000	1		
		001	2		
		010	4		
		011	8		
		100	16		
		101	32		
		110	48		
		111	64		

SAR_ADC_REG_COV SAR ADC Regular Mode Conversion Channel Register

Address : 0x4012_0014

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

Bit	Name	Description	Access	Reset value
[31:4]		Reserved	R	0x0
[3:0]	REG_CHA	Regular Mode Conversion Channel This register contains SAR ADC channel information when SAR ADC in regular mode. This register will be update after SAR ADC conversion is done.	R	0x0

SAR_ADC_REG_SEQ SAR ADC Regular Sequence Register
Address : 0x4012_0018

431	30	29	28	27	26	25	24
SEQ7TH_SEL				SEQ6TH_SEL			
23	22	21	20	19	18	17	16
SEQ5TH_SEL				SEQ4TH_SEL			
15	14	13	12	11	10	9	8
SEQ3TH_SEL				SEQ2TH_SEL			
7	6	5	4	3	2	1	0
SEQ1TH_SEL				SEQ0TH_SEL			

Bit	Name	Description	Access	Reset value
[31:28]	SEQ7TH_SEL	The 7 th conversion in regular sequence This indicates which channel in 7 th regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[27:24]	SEQ6TH_SEL	The 6 th conversion in regular sequence This indicates which channel in 6 th regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[23:20]	SEQ5TH_SEL	The 5 th conversion in regular sequence This indicates which channel in 5 th regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[19:16]	SEQ4TH_SEL	The 4 th conversion in regular sequence This indicates which channel in 4 th regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[15:12]	SEQ3TH_SEL	The 3 rd conversion in regular sequence This indicates which channel in 3 rd regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[11:8]	SEQ2TH_SEL	The 2 nd conversion in regular sequence This indicates which channel in 2 nd regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[7:4]	SEQ1TH_SEL	The 1 st conversion in regular sequence This indicates which channel in 1 st regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[3:0]	SEQ0TH_SEL	The 0 th conversion in regular sequence	R/W	0x0

Bit	Name	Description	Access	Reset value
		This indicates which channel in 0 th regular sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table		

SAR_ADC_INJ_SEQ SAR ADC Injected Sequence Register								Address : 0x4012_001C
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
SEQ3RD_SEL				SEQ2RD_SEL				
7	6	5	4	3	2	1	0	
SEQ1RD_SEL				SEQ0RD_SEL				

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:12]	SEQ3RD_SEL	The 3 rd conversion in injected sequence This indicates which channel in 3 rd injected sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[11:8]	SEQ2RD_SEL	The 2 nd conversion in injected sequence This indicates which channel in 2 nd injected sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[7:4]	SEQ1RD_SEL	The 1 st conversion in injected sequence This indicates which channel in 1 st injected sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0
[3:0]	SEQ0RD_SEL	The 0 th conversion in injected sequence This indicates which channel in 0 th injected sequence Note: The channel mapping of ADC, please refer to SAR ADC channel mapping table	R/W	0x0

SAR_ADC_WDG_TH SAR ADC Watchdog Threshold Register								Address : 0x4012_0020
31	30	29	28	27	26	25	24	
HIGH_TH[11:4]								
23	22	21	20	19	18	17	16	
HIGH_TH[3:0]				--				
15	14	13	12	11	10	9	8	
LOW_TH[11:4]								
7	6	5	4	3	2	1	0	
LOW_TH[3:0]				--				



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
[31:20]	HIGH_THRESHOLD (HIGH_TH)	Analog watchdog high threshold	R/W	0x000
[19:16]	--	Reserved	R	0x0
[15:4]	LOW_THRESHOLD (LOW_TH)	Analog watchdog low threshold	R/W	0x000
[3:0]		Reserved	R	0x0

SAR_ADC_REG_DATA SAR ADC Regular Data Register

Address : 0x4012_0024

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

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	SAR_ADC_REG_DATA	SAR ADC regular data/Manual Trigger source SAR ADC in regular mode and REG_SEQ_GAP is set to "1101", SAR ADC completes data conversion and wait SW to issue trigger signal to inform SAR ADC controller start next sample conversion. SW must write "1" to start next SAR ADC sample operation.	R/W	0x0000

SAR_ADC_INJ0_DATA SAR ADC Injected0 Data Register

Address : 0x4012_0028

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADC0_INJ0_DATA[15:8]							
7	6	5	4	3	2	1	0
ADC0_INJ0_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	ADC0_INJ0_DATA	ADC0 injected0 data	R/W	0x0000

SAR_ADC_INJ1_DATA SAR ADC Injected1 Data Register

Address : 0x4012_002C

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

ADC0_INJ1_DATA[15:8]							
7	6	5	4	3	2	1	0
ADC0_INJ1_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	ADC0_INJ1_DATA	ADC0 injected1 data	R/W	0x0000

SAR_ADC_INJ2_DATA SAR ADC Injected2 Data Register Address : 0x4012_0030

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADC0_INJ2_DATA[15:8]							
7	6	5	4	3	2	1	0
ADC0_INJ2_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	ADC0_INJ2_DATA	ADC0 injected2 data	R/W	0x0000

SAR_ADC_INJ3_DATA SAR ADC Injected3 Data Register Address : 0x4012_0034

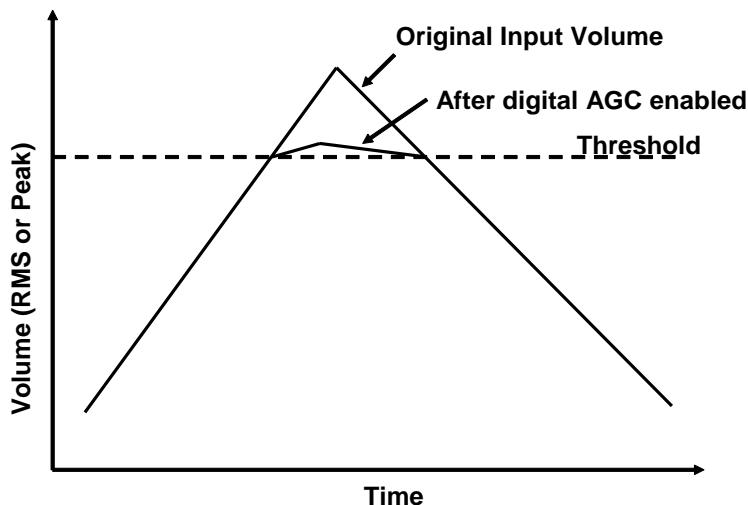
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADC0_INJ3_DATA[15:8]							
7	6	5	4	3	2	1	0
ADC0_INJ3_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	ADC0_INJ3_DATA	ADC0 injected3 data	R/W	0x0000

15. Audio Delta-Sigma ADC

15.1. Introduction

A 16-bit resolution SDM ADC (Delta-Sigma Modulation ADC) with DAGC (Digital Audio Gain Control) and auto mute function are embedded in GPCM1F series. When peak or RMS of record sound is larger than the value, {threshold, 0x80}, the digital AGC will start to activate then lower down the PGA gain. The definition of attack time is how fast the digital AGC responses to the sound data peak or RMS over the threshold. The fast attack time will result in fast response to the sound 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 sound data, meaning the compress will start after a while. This is to preserve the transient part in the wave, but it's inappropriate to avoid 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. $1 \times T_{sample}$. Real attack time of "100" means the attack time is approx. $128 \times T_{sample}$, etc. Where T_{sample} time is the recording internal of ADC, for example, when 8 kHz sample rate is applied, T_{sample} is $1/8K = 0.125\text{ms}$. The definition of release time is how fast the compressor responses to the sound data peak or RMS 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. We may try both settings and find the best one for the application. Real release time of "1" means the release time is approximately $1 \times T_{sample}$ ms. Real release time of "100" means the release time is approximately $128 \times T_{sample}$, etc.



Auto Mute:

- (a) Function: It is capable of reducing background noise after running AGC.
- (b) Threshold (0x4012100C Bit[31:16]:16-bit): It determines whether to enter Mute Mode or the threshold value of return from mute mode back to Normal Mode. The larger Threshold value is given, the easier to enter Mute Mode.
- (c) Normal & Mute debounce (16-bit):
 - (1) When the current MIC volume reaches the user's given Threshold value, Debounce value will determine for how long it turns the normal mode into mute mode, or return from mute mode back to normal mode.
 - (2) Mute debounce(0x40121010 Bit[15:0]): During Normal Mode and the current MIC volume is lower than Threshold, Mute debounce counter+1. When the total count exceeds the setup value of Mute debounce, it enters Mute Mode.
 - (3) Normal debounce(0x40121010 Bit[31:16]): During Mute Mode and the current MIC volume is higher than Threshold, Normal debounce counter+1. When the total count exceeds the setup value of Normal debounce, it returns back to Normal Mode.
- (d) RAMP Counter:
 - (1) It is used to adjust the slope of entering Mute mode or returning to Normal mode, total of eight-level adjustment available.
 - (2) When Normal mode is gradually entering into Mute mode, for each Ramp Count -1, it will reduce by $1/8 \times \text{ADC data}$.

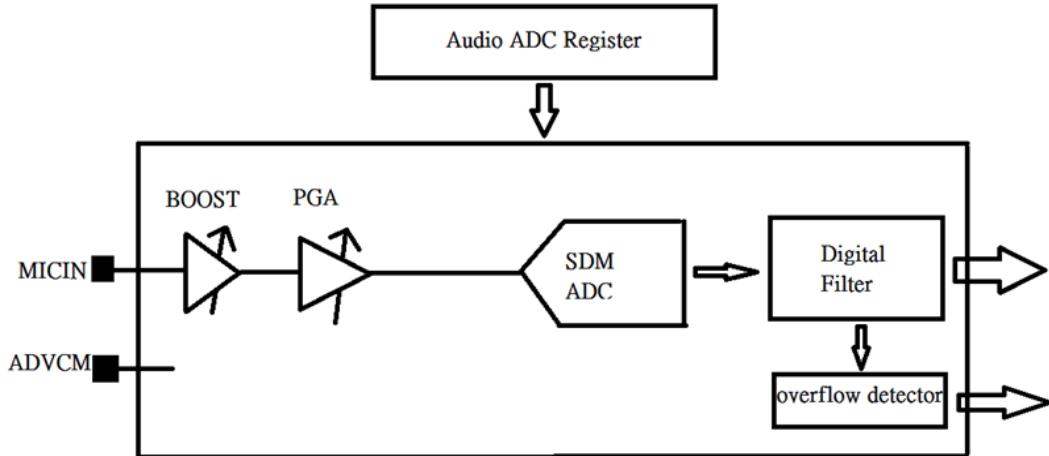
- (3) When Mute mode is gradually returning to Normal mode, for each Ramp Count +1, it will increase by 1/8*ADC data.
 (4) RAMP Counter Step(0x4012100C Bit[3:2]):2-bit. It is used to control the number of count to allow Ramp Counter +1 or -1. For example, if it is set to 03, it will count for three times before allowing Counter +1 or -1.

15.2. Feature

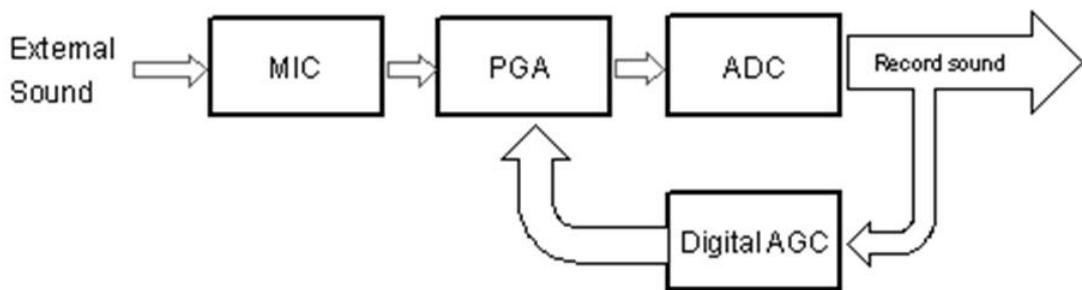
- 16-bit Delta-Sigma ADC
- ADC high pass filter
- Boost gain control(4-level)
- PGA (Programmable Gain Amplifier) control (32-level)
- AGC (Auto Gain Control)
- Auto Mute function.

15.3. Block Diagram

MIC_IN Analog Input signal will pass through Booster gain to amplify the analog signal and then, re-amplify again via PGA. After that, through ADC to convert analog to digital signal before perform DAGC and Auto Mute processing via Digital Filter.



The compressor detects the output level before the main volume multiplier, and it will control the main volume dynamically. See the following diagram for more details.



15.4. Register Description

Register Map

Base Address : 0x4012_1000				
Name	Description	Address	Access	Reset value
DS_ADC_CTRL	Delta-Sigma ADC Control Register	0x4012_1000	R/W	0x0000_8000
DS_ADC_AGC_CTRL0	Delta-Sigma ADC AGC Control Register 0	0x4012_1004	R/W	0x1000_2070

Base Address : 0x4012_1000						
Name	Description			Address	Access	Reset value
DS_ADC_AGC_CTRL1	Delta-Sigma ADC AGC Control Register 1			0x4012_1008	R/W	0x0003_FC80
DS_ADC_MUTE_CTRL0	Delta-Sigma ADC MIC Auto-mute Control Register 0			0x4012_100c	R/W	0x0500_0002
DS_ADC_MUTE_CTRL1	Delta-Sigma ADC MIC Auto-mute Control Register 1			0x4012_1010	R/W	0x0001_0001
DS_ADC_STS	Delta-Sigma ADC Status Register			0x4012_1018	R/W	0x0000_0000
DS_ADC_DATA	Delta-Sigma ADC DATA Port			0x4012_101C	R	0x0000_0000

Register Function

DS_ADC_CTRL Delta-Sigma ADC Control Register								Address : 0x4012_1000
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
--	MIC_FIFO_LEVEL[3:0]			--	FIFO_INT_EN	FIFO_OVWR_EN	FIFO_EN	--
15	14	13	12	11	10	9	8	--
FMT_SEL	--	PGA_GAIN[4:0]						--
7	6	5	4	3	2	1	0	--
IN_LIMIT	BOOST_GAIN		DSADC_EN	DSADC_RST	HPF_EN	HPF_EN	MIC_EN	--

Bit	Name	Description	Access	Reset value
[31:23]	--	Reserved	R	0
[22:20]	FIFO_LVL (MIC_FIFO_LEVEL)	MIC FIFO_LEVEL When data numbers in MIC FIFO greater than "MIC_FIFO_LEVEL", MIC_FIFO_FLAG will be set. 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	R/W	0x0
[19]	--	Reserved	R	0
[18]	INT_ENABLE (FIFO_INT_EN)	MIC FIFO_INT_EN When MIC FIFO data numbers greater than "MIC_FIFO_LEVEL" or full, MIC FIFO can issue interrupt to CPU. If this bit is disabled, CPU must polling "MIC_FIFO_FULL" flag to confirm MIC FIFO status. 0 : MIC FIFO interrupt is disabled 1 : MIC FIFO interrupt is enabled	R/W	0x0
[17]	FIFO_OVWR_ENABLE (FIFO_OVWR_EN)	MIC FIFO_OVERWRITE This register is used to control newest data will be overwrite by MIC FIFO or not When MIC FIFO is full. 0 : Newest MIC data won't be overwritten by MIC FIFO when MIC FIFO is full.	R/W	0x0

Bit	Name	Description			Access	Reset value																																																																																														
		1 : Newest MIC data will be overwritten by MIC FIFO when MIC FIFO is full.																																																																																																		
[16]	FIFO_ENABLE (FIFO_EN)	MIC FIFO_EN This register is used to control mic FIFO. If this bit is disabled, MIC data must be read by DMA engine real time. 0 : disable 1: Enable			R/W	0x0																																																																																														
[15]	FMT_SEL	Delta-Sigma ADC data sign/ unsigned control 0: Unsigned data 1: Sign data			R/W	1																																																																																														
[14:12]	--	Reserved			R	0																																																																																														
[12:8]	PGA_GAIN	Delta-Sigma ADC PGA GAIN <table border="1"> <thead> <tr> <th>ADC_PGA_GAIN</th> <th>PGA gain (dB)</th> <th>Total gain (db)@Boost=21dB</th> </tr> </thead> <tbody> <tr><td>00000</td><td>33</td><td>54</td></tr> <tr><td>00001</td><td>31.5</td><td>52.5</td></tr> <tr><td>00010</td><td>30</td><td>51</td></tr> <tr><td>00011</td><td>28.5</td><td>49.5</td></tr> <tr><td>00100</td><td>27</td><td>48</td></tr> <tr><td>00101</td><td>25.5</td><td>46.5</td></tr> <tr><td>00110</td><td>24</td><td>45</td></tr> <tr><td>00111</td><td>22.5</td><td>43.5</td></tr> <tr><td>01000</td><td>21</td><td>42</td></tr> <tr><td>01001</td><td>19.5</td><td>40.5</td></tr> <tr><td>01010</td><td>18</td><td>39</td></tr> <tr><td>01011</td><td>16.5</td><td>37.5</td></tr> <tr><td>01100</td><td>15</td><td>36</td></tr> <tr><td>01101</td><td>13.5</td><td>34.5</td></tr> <tr><td>01110</td><td>12</td><td>33</td></tr> <tr><td>01111</td><td>10.5</td><td>31.5</td></tr> <tr><td>10000</td><td>9</td><td>30</td></tr> <tr><td>10001</td><td>7.5</td><td>28.5</td></tr> <tr><td>10010</td><td>6</td><td>27</td></tr> <tr><td>10011</td><td>4.5</td><td>25.5</td></tr> <tr><td>10100</td><td>3</td><td>24</td></tr> <tr><td>10101</td><td>1.5</td><td>22.5</td></tr> <tr><td>10110</td><td>0</td><td>21</td></tr> <tr><td>10111</td><td>-1.5</td><td>19.5</td></tr> <tr><td>11000</td><td>-3</td><td>18</td></tr> <tr><td>11001</td><td>-4.5</td><td>16.5</td></tr> <tr><td>11010</td><td>-6</td><td>15</td></tr> <tr><td>11011</td><td>-7.5</td><td>13.5</td></tr> <tr><td>11100</td><td>-9</td><td>12</td></tr> <tr><td>11101</td><td>-10.5</td><td>10.5</td></tr> <tr><td>11110</td><td>-12</td><td>9</td></tr> </tbody> </table>	ADC_PGA_GAIN	PGA gain (dB)	Total gain (db)@Boost=21dB	00000	33	54	00001	31.5	52.5	00010	30	51	00011	28.5	49.5	00100	27	48	00101	25.5	46.5	00110	24	45	00111	22.5	43.5	01000	21	42	01001	19.5	40.5	01010	18	39	01011	16.5	37.5	01100	15	36	01101	13.5	34.5	01110	12	33	01111	10.5	31.5	10000	9	30	10001	7.5	28.5	10010	6	27	10011	4.5	25.5	10100	3	24	10101	1.5	22.5	10110	0	21	10111	-1.5	19.5	11000	-3	18	11001	-4.5	16.5	11010	-6	15	11011	-7.5	13.5	11100	-9	12	11101	-10.5	10.5	11110	-12	9	R/W	0x0
ADC_PGA_GAIN	PGA gain (dB)	Total gain (db)@Boost=21dB																																																																																																		
00000	33	54																																																																																																		
00001	31.5	52.5																																																																																																		
00010	30	51																																																																																																		
00011	28.5	49.5																																																																																																		
00100	27	48																																																																																																		
00101	25.5	46.5																																																																																																		
00110	24	45																																																																																																		
00111	22.5	43.5																																																																																																		
01000	21	42																																																																																																		
01001	19.5	40.5																																																																																																		
01010	18	39																																																																																																		
01011	16.5	37.5																																																																																																		
01100	15	36																																																																																																		
01101	13.5	34.5																																																																																																		
01110	12	33																																																																																																		
01111	10.5	31.5																																																																																																		
10000	9	30																																																																																																		
10001	7.5	28.5																																																																																																		
10010	6	27																																																																																																		
10011	4.5	25.5																																																																																																		
10100	3	24																																																																																																		
10101	1.5	22.5																																																																																																		
10110	0	21																																																																																																		
10111	-1.5	19.5																																																																																																		
11000	-3	18																																																																																																		
11001	-4.5	16.5																																																																																																		
11010	-6	15																																																																																																		
11011	-7.5	13.5																																																																																																		
11100	-9	12																																																																																																		
11101	-10.5	10.5																																																																																																		
11110	-12	9																																																																																																		



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description			Access	Reset value
		11111 -∞ -∞				
[7:6]	IN_LIMIT	Select ADC input limit range. 00: 0.84*Full range 01: 0.71*Full range 10: 0.60*Full range 11: 0.50*Full range			R/W	0x0
[5:4]	BOOST_GAIN	Boost Amplifier Gain Control. 00 : 0db 01 : 7db 10 : 14db 11 : 21db			R/W	0x0
[3]	DSADC_ENABLE (DSADC_EN)	Delta-Sigma ADC Control 0: disabled 1:enabled			R/W	0x0
[2]	DSADC_RESET (DSADC_RST)	Delta-Sigma ADC reset. 0: Reset 1: normal operation Note: This bit should be set to "1" when ADC is enabled.			R/W	0x0
[1]	HPF_ENABLE (HPF_EN)	ADC high pass filter control. 0= Disabled 1= Enabled			R/W	0x0
[0]	MIC_ENABLE (MIC_EN)	Enable microphone amplifier. 0: Disabled 1: Enabled			R/W	0x0

DS_ADC_AGC_CTRL0 DS_ADC AGC Control Register Address 0x4012_1004

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

Bit	Name	Description	Access	Reset value
[31]	--	Reserved	R	0
[30:16]	TOGGLE_THRESHOLD	TOGGLETHRESHOLD Register This register is used to control 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.	R/W	0x1000
[15]	--	Reserved	R	0
[14:8]	THRESHOLD	Digital AGC threshold control. minimum value is 0x01 maximum value is 0x7F Note: 0x00 is not allowed.	R/W	0x20



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
[7:3]	--	Reserved	R	0x07
[2]	DAGC_ENABLE (DAGC_EN)	Digital AGC enable control 0: Disable DAGC (default) 1: Enable DAGC 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.	R/W	0x0
[1]	ZERO_CROSS_DISABLE (ZERO_CROSS)	Disable Zero Cross Function of Compressor. As described before, the compressor will adjust the main volume automatically. This bit is used to control whether or not the volume change wait to zero cross. 0: Enable zero cross 1: Disable zero cross.	R	0x0
[0]	MODE_SEL	Peak Mode Control register. 0: RMS mode 1: Peak mode	R	0x0

DS_ADC_AGC_CTRL1 DS_ADC AGC Control Register1 Address 0x4012_1008

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

Bit	Name	Description	Access	Reset value
[31:20]	UPDATE_FREQ	Update frequency control. (12-bit resolution) Note: It is used to set up the AGC refresh rate (adjust volume gain once per number of samples).	R/W	0x0
[19:18]	RELEASE_SCALE	Release Time Scale Control 00= Release Time*1 01= Release Time*4 10= Release Time*16 11= Release Time*64	R/W	0x0
[17:10]	RELEASE_TIME	Release time control The definition of release time is how fast the compressor responses to the wave data peak or RMS lower than the threshold.	R/W	0x0FF
[9:8]	ATTACK_SCALE	Attack Time Scale Control 00= Attack Time*1 01= Attack Time*4 10= Attack Time*16 11= Attack Time*64	R/W	0x0
[7:0]	ATTACK_TIME	Attack time control. The definition of attack time is how fast the compressor responses to the	R/W	0x80



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description						Access	Reset value
		wave data peak or RMS over the threshold.							

DS_ADC_MUTE_CTRL0 Delta-Sigma ADC MIC Auto-Mute Control Register0 Address:0x4012_100C

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

Bit	Name	Description	Access	Reset value
[31:16]	SILENCE_TH	Threshold value In normal mode, wave in data smaller than silence threshold value and keep more than silence debounce counts, it will change to silence mode. Or it will keep in normal mode. In silence mode, if wave in data is bigger than silence threshold value and keep more than normal debounce counts, it will change back to normal mode. Or it will keep in silence mode.	R/W	0x0500
[15:5]	--	Reserved	R	0
4	ECHO_CANCEL_ENAB LE (ECHO_CANCEL_EN)	Echo canceling control bit This bit is used to control "DS_ADC_DATA" register bit [31:16] containing DAC conversion data or not. 1: "DS_ADC_DATA"[31:16] containing DAC conversion data. 0: "DS_ADC_DATA"[31:16] is "0x0000"	R/W	0
[3:2]	RAMPCNT_STEP	RAMP counter step control bit 00: 0 step 01: 1 step 10: 2 steps 11: 3 steps Decreasing step in normal mode, increasing step in silence mode. This value means sample counts in each step. For example, rampcnt_step=0x3 means through 3 sample data = 1 step	R/W	0
[1]	FMT_SEL	MIC AGC signed/unsigned data select This bit is used to control the MSB bit of Delta-Sigma conversion data is signed or unsigned. 1: the MSB bit of Delta-Sigma data is signed. 0: the MSB bit of Delta-Sigma data is unsigned.	R/W	1
[0]	MIC_AUTOMUTE_EN	Auto mute function control bit 0: Disabled 1: Enabled	R/W	0

DS_ADC_MUTE_CTRL1 Delta-Sigma ADC MIC Auto-Mute Control Register1 Address : 0x4012_1010

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

Bit	Name	Description	Access	Reset value
[31:16]	NORMAL_DEBOUNCE	Normal debounce value. Debounce count to enter the normal mode	R/W	1
[15:0]	SILENCE_DEBOUNCE	Mute debounce value. Debounce count to enter the mute mode	R/W	1

DS_ADC_STS Delta-Sigma ADC Status Register								Address : 0x4012_1018
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
FIFO_FULL_FLAG AG INT_FLAG --								
15	14	13	12	11	10	9	8	
-- CURPGA_GAIN								
7	6	5	4	3	2	1	0	
--								

Bit	Name	Description	Access	Reset value
[31:20]	--	Reserved	R	0
[19]	FIFO_FULL_FLAG	MIC FIFO full flag 0: Not full. 1: MIC FIFO is full or greater than MIC_FIFO_LEVEL. Note: This bit will be cleared to "0" automatically when data read from MIC FIFO.	R/W	0
[18]	INT_FLAG	Delta-Sigma ADC interrupt flag 0: No DS_ADC INT occurred. 1: DS_ADC INT occurred. Write '1' to clear this flag.	R/W	0
[17:13]	--	Reserved	R	0
[12:8]	CURPGA_GAIN	Current PGA Gain value. Indicates the actually PGA_GAIN gain to Delta-Sigma ADC.	R	0
[7:0]	--	Reserved	R	0

DS_ADC_DATA Delta-Sigma Data Register								Address : 0x4012_101C
31	30	29	28	27	26	25	24	
DS_ADC_DATA[31:24]								
23	22	21	20	19	18	17	16	
DS_ADC_DATA[23:16]								
15	14	13	12		10	9	8	

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

Bit	Name	Description	Access	Reset value
[31:0]	DS_ADC_DATA	DS_ADC data register. [15:0]: Delta-Sigma ADC Data. [31:16]: Determined by ECHO_CANCEL_EN bit, whether to store DAC conversion data. Note: If MIC_MUTE_EN is enabled, data is processed by MIC_MUTE module else is the Delta-Sigma ADC conversion data.	R	0x0000

DS_ADC_WAKEUP_CTRL0								Delta-Sigma ADC Wakeup Control Register0		Address : 0x4012_1020			
31	30	29	28	27	26	25	24	SMP_NUM [7:0]					
<hr/>													
23	22	21	20	19	18	17	16	<hr/>					
<hr/>													
15	14	13	12	11	10	9	8	<hr/>					
<hr/>													
7	6	5	4	3	2	1	0	<hr/>					
<hr/>													
								ACC_MOD	WAKEUP_INTF	WAKEUP_EN			

Bit	Name	Description	Access	Reset value
[31:24]	SMP_NUM	Delta-Sigma ADC wake up sample number This register is used to control ADC sample numbers during ADC detect region. When ADC exceeds this register setting value, it finish ADC detection and compare accumulation value in detect region and "SD_ADC_WAKEUP_THRESHOLD". If accumulation value greater than "SD_ADC_WAKEUP_THRESHOLD", ADC will issue interrupt to confirm CPU or ADC wait for next detection. 0x00: 4 samples 0x01: 8 samples 0x02 :12 samples 0x03:16 samples . . . 0xFF:1024 samples	R/W	0x0000
[23:3]	--	Reserved	R	0
[2]	ACC_MOD	DS-ADC data accumulation method select. This bit is used to control Delta-Sigma ADC data accumulation method. 0: Delta-Sigma ADC data is positive 1: Delta-Sigma ADC data is negative	R/W	0x0
[1]	WAKEUP_INTF	DS-ADC wake up status flag. This bit is assert to "1" When Delta-Sigma ADC detect sound from outside and accumulation of detection value greater than	R/W	0x0

Bit	Name	Description	Access	Reset value
		"SD_ADC_WAKEUP_CTRL1_THRESHOLD", CPU must write "1" to clear it.		
[0]	WAKEUP_ENABLE (WAKEUP_EN)	Delta-Sigma ADC Wakeup Enable Control If system enters deep-sleep mode, user can enable this bit to turn on Delta-Sigma ADC detects sound from outside. If ADC detects sounds from outside, it issues interrupt to confirm CPU to wakeup whole system. When CPU receives interrupt, it must clear this bit to "0". 0:Disabled 1:Enabled	R/W	0x0

DS_ADC_WAKEUP_CTRL1
Delta-Sigma ADC Wakeup Control Register1
Address : 0x4012_1024

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

Bit	Name	Description	Access	Reset value
[31:26]	--	Reserved	R	0
[25:16]	STABLE_TIME	DS-ADC wakeup stable time This register defines Delta-Sigma ADC stable time. ADC starts to accumulate ADC value when stable time counter exceeds register setting value. 0x00:1/32768 sec. 0x01:2/32768 sec. .. 0x3FF:1024/32768 sec.	R/W	0x0
[15:0]	THRESHOLD	Delta-Sigma ADC Wakeup Threshold This register defines ADC wakeup threshold value.	R/W	0x0

16. DAC Up-sampling / Audio PWM control

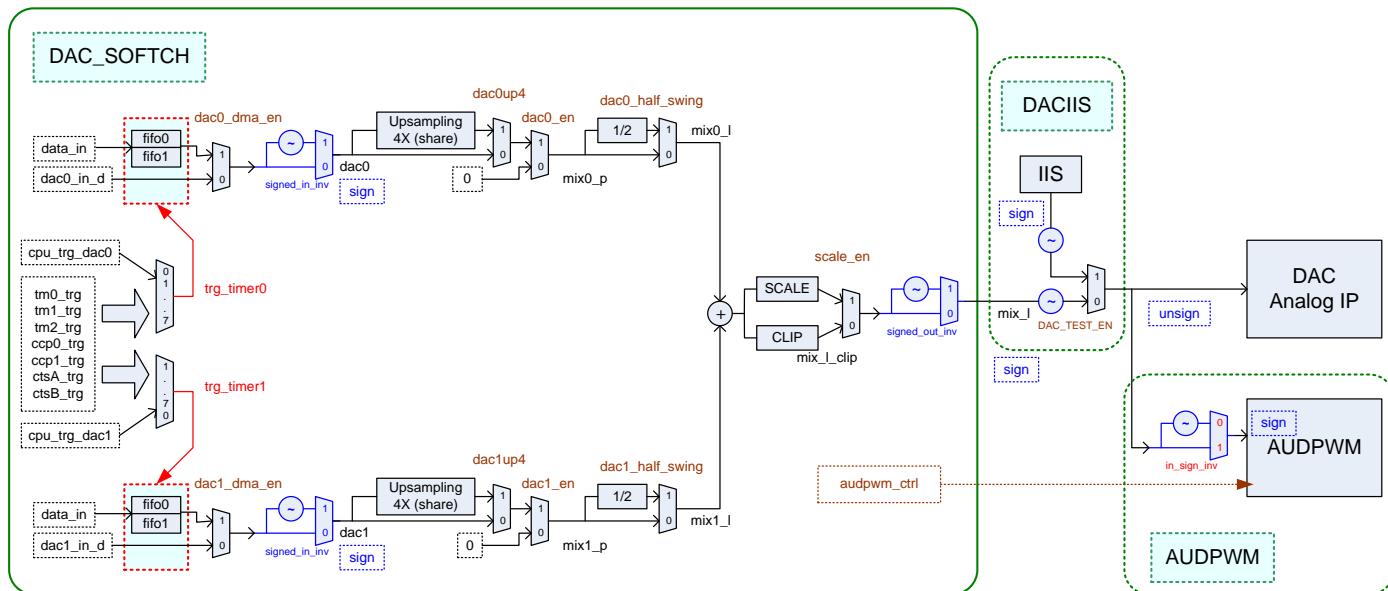
16.1. Introduction

The audio of GPCM1F SERIALS can be exported to the DAC or AUDPWM. The DAC data ranges from 0x0000 to 0xFFFF (12-bit DAC). To AUDPWM data ranges from 0x0000 to 0xFFFF. If the DAC data will be processed as PCM data, user needs to know that the DC level for the DAC Data is 0x0000. The data of DAC should be delivered to DAC_CH0_IN (W) (0x4013_0010) and DAC_CH1_IN (W) (0x4013_0014). GPCM1F SERIALS also supports DMA data in. GPCM1F SERIALS supports two input channels data port and new 4x up-sampling function. User is allowed to send the various data into different channels and thus, IC will perform the mix operation before DAC conversion.

16.2. Feature

- Supports 4x up-sampling
- Supports 2 input channels
- Supports DMA input mode
- Supports scale / truncation mode when data saturation
- Audio PWM support digital gain control
- Audio PWM support auto mute control

16.3. Block Diagram



16.4. Function

DAC SOFTCH module can enable DAC Channel0 or Channel1 separately (setting by DAC_CHx_EN). Hardware samples DAC0/DAC1 input data when trigger pulse happened (setting by CHx_TRG_SEL). And user can set CHx_UPSAMPLING_MODE to select up-sampling mode respectively.

Hardware will mix two channels data before output to current DAC IP or Audio PWM IP

Output Mode:

Current DAC mode:

After power on reset, the DAC analogy IP is default as turned off. So if users need to use the DAC, users have to turn on it by setting AIP_DAC_EN =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 set AUDPWM_GAIN to re-scale audio data from 1/32x to 2x.

Input Mode:
Manual mode:

User need to deliver new update DAC data to DAC_CH0_IN/ DAC_CH1_IN.

DMA mode:

User can also use DMA control unit to deliver DAC data. Hardware will auto fill DAC FIFO when FIFO not full.(setting by CHx_DMA_EN)

16.5. Register Description

Register Map

Base Address : 0x4013_1000						
Name	Description			Address	Access	Reset value
DAC_CTRL	DAC Control Register			0x4013_0000	R/W	0x0000_0000
DAC_STS	DAC Status register			0x4013_0004	R/W	0x0000_0000
AUDPWM_CTRL	AUDIO PWM Control Register			0x4013_000C	R/W	0x0000_0000
DAC_CH0_DATA	DAC Channel0 input data			0x4013_0010	R/W	0x0000_0000
DAC_CH1_DATA	DAC Channel1 input data			0x4013_0014	R/W	0x0000_0000
MIX_DATA_OUT	DAC Mixed output data			0x4013_0018	R	0x0000_0000
DAC_CH0_DMA_DATA0	DAC Channel0 DMA FIFO data0 value			0x4013_0020	R	0x0000_0000
DAC_CH0_DMA_DATA1	DAC Channel0 DMA FIFO data1 value			0x4013_0024	R	0x0000_0000
DAC_CH1_DMA_DATA0	DAC Channel1 DMA FIFO data0 value			0x4013_0030	R	0x0000_0000
DAC_CH1_DMA_DATA1	DAC Channel1 DMA FIFO data1 value			0x4013_0034	R	0x0000_0000

Register Function

DAC_CTRL DAC Control Register								Address : 0x4013_0000
31	30	29	28	27	26	25	24	
CH1_ERR_EN	CH0_ERR_EN	CH1_INT_EN	CH0_INT_EN	--	--	UPSMP_SW_RST	SW_RST	
23	22	21	20	19	18	17	16	
DAC_EN	--	SIGN_OUT_IN_V	SIGN_IN_INV		--		SCALE_EN	
15	14	13	12	11	10	9	8	
CH1_HALF_EN		CH1_TRG_SEL		DAC_CH1_UPSMP_MODE	CH1_DMA_EN	DAC_CH1_EN		
7	6	5	4	3	2	1	0	
CH0_HALF_EN		CH0_TRG_SEL		DAC_CH0_UPSMP_MODE	CH0_DMA_EN	DAC_CH0_EN		

Bit	Name	Description	Access	Reset value
[31]	DAC_CH1_DMAERR_IN T_ENABLE (CH1_ERR_INT_EN)	DAC Channel1 DMA FIFO Error interrupt enable 1: interrupt enabled 0: interrupt disabled	R/W	0x0
[30]	DAC_CH0_DMAERR_IN T_ENABLE (CH0_ERR_INT_EN)	DAC Channel0 DMA FIFO Error interrupt enable 1: interrupt enabled 0: interrupt disabled	R/W	0x0
[29]	DAC_CH1_INT_ENABL	DAC Channel1 interrupt enable	R/W	0x0

Bit	Name	Description	Access	Reset value																		
	E (CH1_INT_EN)	1: interrupt enabled 0: interrupt disabled																				
[28]	DAC_CH0_INT_ENABL E (CH0_INT_EN)	DAC Channel0 interrupt enable 1: interrupt enabled 0: interrupt disabled	R/W	0x0																		
[27:26]	--	Reserved	R	0																		
[25]	UPSMP_SW_RESET (UPSMP_SW_RST)	Up sample function Software reset 1: reset assert, and auto de-assert when reset done	R/W	0x0																		
[24]	DAC_SW_RESET (SW_RST)	DAC control Software reset 1: reset assert, and auto de-assert when reset done	R/W	0x0																		
[23]	CURRENT_DAC_ENAB LE (DAC_EN)	Current DAC enable 1: enabled, 0: disabled	R/W	0x0																		
[22]	--	Reserved	R	0																		
[21]	DACOUT_FMT_SEL (SIGN_OUT_INV)	Mixed data signed to unsigned enable 1: output unsigned data 0: output signed data (default output to DACiis signed data)	R/W	0x0																		
[20]	DACIN_FMT_SEL (SIGN_IN_INV)	Input data signed/unsigned select 1: input unsigned data 0: input signed data(default)	R/W	0x0																		
[19:17]	--	Reserved	R	0																		
[16]	SCALE_ENABLE (SCALE_EN)	Saturation mode enable 1: scale enable 0 : truncation	R/W	0x0																		
[15]	CH1_HALF_ENABLE (CH1_HALF_EN)	DAC Channel1 volume control 1: Half swing, signed data 0 : Full swing, signed data	R/W	0x0																		
[14:12]	DAC_CH1_TRG_SEL (CH1_TRG_SEL)	DAC channel1 Trigger source selection. <table border="1"> <tr><td>CH0_TRG_SEL[2:0]</td><td>Compare Mechanism</td></tr> <tr><td>000</td><td>Manual (new channel1 data in)</td></tr> <tr><td>001</td><td>Timer0</td></tr> <tr><td>010</td><td>Timer1</td></tr> <tr><td>011</td><td>Timer2</td></tr> <tr><td>100</td><td>CCP0</td></tr> <tr><td>101</td><td>CCP1</td></tr> <tr><td>110</td><td>CTS timer0</td></tr> <tr><td>111</td><td>CTS timer1</td></tr> </table>	CH0_TRG_SEL[2:0]	Compare Mechanism	000	Manual (new channel1 data in)	001	Timer0	010	Timer1	011	Timer2	100	CCP0	101	CCP1	110	CTS timer0	111	CTS timer1	R/W	0x0
CH0_TRG_SEL[2:0]	Compare Mechanism																					
000	Manual (new channel1 data in)																					
001	Timer0																					
010	Timer1																					
011	Timer2																					
100	CCP0																					
101	CCP1																					
110	CTS timer0																					
111	CTS timer1																					
[11:10]	DAC_CH1_UPSMP_MO DE	DAC channel1 UpSampling selection 00 : Bypass up sampling block (Disable) 1x : 4x up sampling	R/W	0x0																		
[9]	DAC_CH1_DMA_ENABL E (CH1_DMA_EN)	DAC Channel1 DMA enable 1: enabled 0: disabled Need to set DMA control unit first	R/W	0x0																		
[8]	DAC_CH1_ENABLE	DAC channel1 enable	R/W	0x0																		

Bit	Name	Description	Access	Reset value																		
	(DAC_CH1_EN)	1: enabled 0: disabled																				
[7]	DAC_CH0_HALF_ENAB LE (CH0_HALF_EN)	DAC Channel0 volume control 1: Half swing, signed data 0 : Full swing, signed data	R/W	0x0																		
[6:4]	DAC_CH0_TRG_SEL (CH0_TRG_SEL)	DAC channel0 Trigger source selection. <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>CH0_TRG_SEL[2:0]</td><td>Compare Mechanism</td></tr> <tr><td>000</td><td>Manual (new channel0 data in)</td></tr> <tr><td>001</td><td>Timer0</td></tr> <tr><td>010</td><td>Timer1</td></tr> <tr><td>011</td><td>Timer2</td></tr> <tr><td>100</td><td>CCP0</td></tr> <tr><td>101</td><td>CCP1</td></tr> <tr><td>110</td><td>CTS timer0</td></tr> <tr><td>111</td><td>CTS timer1</td></tr> </table>	CH0_TRG_SEL[2:0]	Compare Mechanism	000	Manual (new channel0 data in)	001	Timer0	010	Timer1	011	Timer2	100	CCP0	101	CCP1	110	CTS timer0	111	CTS timer1	R/W	0x0
CH0_TRG_SEL[2:0]	Compare Mechanism																					
000	Manual (new channel0 data in)																					
001	Timer0																					
010	Timer1																					
011	Timer2																					
100	CCP0																					
101	CCP1																					
110	CTS timer0																					
111	CTS timer1																					
[3:2]	DAC_CH0_UPSMP_MO DE	DAC channel0 UpSampling selection [1 : 0] = 00 : Bypass up sampling block (with 1x timer) [1 : 0] = 01 : 1x sampling (with 4x timer) [1 : 0] = 1x : 4x up sampling (with 4x timer)	R/W	0x0																		
[1]	DAC_CH0_DMA_ENABL E (CH0_DMA_EN)	DAC Channel0 DMA enable 1: enabled 0: disabled Need to set DMA control unit first	R/W	0x0																		
[0]	DAC_CH0_ENABLE (DAC_CH0_EN)	DAC channel0 enable 1: enabled 0: disabled	R/W	0x0																		

DAC_STS								DAC Status Register		Address : 0x4013_0004	
31	30	29	28	27	26	25	24				
CH1_ERR_FLA G	CH0_ERR_FLA G	CH1_INT_FLA G	CH0_INT_FLA G								
23	22	21	20	19	18	17	16	--			
15	14	13	12	11	10	9	8	--			
7	6	5	4	3	2	1	0	--			

Bit	Name	Description	Access	Reset value
[31]	DAC_CH1_ERR_FLAG (CH1_ERR_FLAG)	DAC Channel1 DMA FIFO error flag When flag asserted, write one clear this bit	R/W	0x0
[30]	DAC_CH0_ERR_FLAG (CH0_ERR_FLAG)	DAC Channel0 DMA FIFO error flag When flag asserted, write one clear this bit	R/W	0x0
[29]	DAC_CH1_INT_FLAG (CH1_INT_FLAG)	DAC Channel1 interrupt flag When flag asserted, write one clear this bit	R/W	0x0

Bit	Name	Description	Access	Reset value
[28]	DAC_CH0_INT_FLAG (CH0_INT_FLAG)	DAC Channel0 interrupt flag When flag asserted, write one clear this bit	R/W	0x0
[27:0]	--	Reserved	R	0

AUDPWM_CTRL Audio PWM Control Register								Address : 0x4013_000C
31	30	29	28	27	26	25	24	
--	--	AUDFLAG	--	--	--			AUDDETECT
23	22	21	20	19	18	17	16	
			--					AUDPWMEN
15	14	13	12	11	10	9	8	
			--	AUDCLK_SEL		AUDPWM_GAIN[5:4]		
7	6	5	4	3	2	1	0	
			AUDPWM_GAIN[3:0]	MUTE_SEL		DATAIN_FMT_SEL		AUDPWM_EN

Bit	Name	Description	Access	Reset value
[31:29]	--	Reserved	R	0
[28]	AUD_DETECT_FLAG (AUDFLAG)	Detect current DAC or Audio PWM mode flag 1: Audio PWM mode 0: Current DAC mode	R	0x0
[27:25]	--	Reserved	R	0
[24]	AUD_DETECT_START (AUDDETECT)	Auto Detect DAC mode or Audio PWM mode. 1:Enable , 0:Disable Note: Because the H/W AUDP and DACO output pin is shared, if this function is enabled, it will automatically detect and determine the external circuit, from the shared pin, a current DAC mode or Audio PWM mode is currently used. Use AUD_DETECT_FLAG (Bit.28) to display.	R/W	0x0
[23:17]	--	Reserved	R	0
[16]	AUDPWM_IP_ENABLE (AUDPWMEN)	Analog PWM enable control ; 1:enabled , 0:disabled	R/W	0x0
[15:12]	--	Reserved	R	0
[11:10]	MUTE_STATE_SEL (AUDCLK_SEL)	Audio PWM Clock mute state selection(generate 1us pulse to count) 00: 16Mhz(div16), 01: 8Mhz(div8), 10: 4Mhz(div4), 11: 2Mhz(div2)	R/W	0x0
[9:4]	AUDPWM_GAIN	Audio PWM Gain register 6'h00: 1/32 6'h01: 2/32 6'h1f: 32/32 (1x gain) 6'h20: 33/32 6'h3F: 64/32 (2x gain)	R/W	0x0
[3:2]	AUTOMUTE_SETTING (MUTE_SEL)	Auto Mute control setting 00: data=0(~1ms) or data keep ~16ms, mute (if data = 0 and time retained at 1ms, or all data are kept the same for 16ms, it will enter Mute mode)	R/W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
		01: data=0(~1ms) or data keep ~2ms, mute (same as above, except difference in time.) 10: data=0(~1ms) , mute 11: auto mute off		
[1]	DATAIN_FMT_SEL	Audio PWM input data signed/unsigned select 1: unsigned 0: signed	R/W	0x0
[0]	AUDPWM_ENABLE (AUDPWM_EN)	Audio PWM enable 1: enabled 0: disabled	R/W	0x0

DAC_CH0_DATA DAC Channel0 input data								Address : 0x4013_0010
15	14	13	12	11	10	9	8	
DAC_CH0_IN[15:8]								
7	6	5	4	3	2	1	0	
DAC_CH0_IN[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DAC_CH0_IN	DAC channel0 input Data value	R/W	0x0

DAC_CH1_DATA DAC Channel1 input data								Address : 0x4013_0014
15	14	13	12	11	10	9	8	
DAC_CH1_IN[15:8]								
7	6	5	4	3	2	1	0	
DAC_CH1_IN[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DAC_CH1_IN	DAC channel1 input Data value	R/W	0x0

MIX_DATA_OUT DAC mixed output data								Address : 0x4013_0018
15	14	13	12	11	10	9	8	
MIX_DATA_OUT[15:8]								
7	6	5	4	3	2	1	0	
MIX_DATA_OUT[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	MIX_DATA_OUT	DAC mixed output Data value	R	0x0



GPCM1F SERIES PROGRAMMING GUIDE

DAC_CH0_DMA_DATA0 DAC0 dma FIFO data0

Address : 0x4013_0020

15	14	13	12	11	10	9	8
DAC0_DMA_DATA0[15:8]							
7	6	5	4	3	2	1	0
DAC0_DMA_DATA0 [7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DAC0_DMA_DATA0	DAC Channel0 DMA FIFO Data0 value Note: FIFO Address Auto Change mechanism: Ch0 has 2-level FIFO, located at 0x40130020 & 0x40130024; DMA destination Address is fixed to 0x40130020; H/W will auto-switch the address. (For DMA only)	R	0x0

DAC_CH0_DMA_DATA1 DAC0 dma FIFO data1

Address : 0x4013_0024

15	14	13	12	11	10	9	8
DAC0_DMA_DATA1[15:8]							
7	6	5	4	3	2	1	0
DAC0_DMA_DATA1 [7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DAC0_DMA_DATA1	DAC Channel0 DMA FIFO Data1 value	R	0x0

DAC_CH1_DMA_DATA0 DAC1 dma FIFO data0

Address : 0x4013_0030

15	14	13	12	11	10	9	8
DAC1_DMA_DATA0[15:8]							
7	6	5	4	3	2	1	0
DAC1_DMA_DATA0 [7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DAC1_DMA_DATA0	DAC Channel1 DMA FIFO Data0 value Note: FIFO Address Auto Change mechanism: Ch1 has 2-level FIFO, located at 0x40130030 & 0x40130034; DMA destination Address is fixed to 0x40130030; H/W will auto-switch the address. (For DMA only)	R	0x0

DAC_CH1_DMA_DATA1 DAC1 dma FIFO data1

Address : 0x4013_0034

15	14	13	12	11	10	9	8
DAC1_DMA_DATA1[15:8]							
7	6	5	4	3	2	1	0
DAC1_DMA_DATA1 [7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	DAC1_DMA_DATA1	DAC Channel1 DMA FIFO Data1 value	R	0x0

17. Serial peripheral interface (SPI)

17.1. Introduction

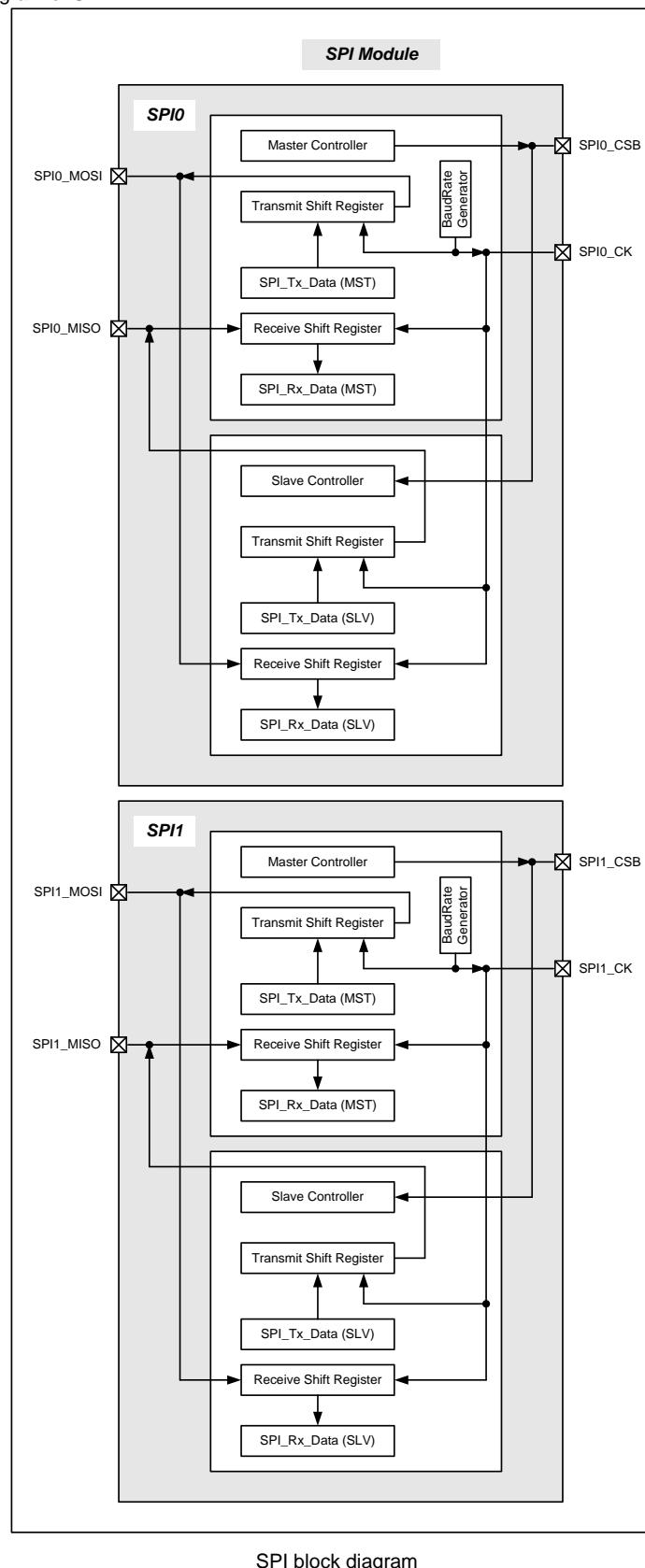
The GPCM1F incorporates two Serial Peripheral Interface (SPI). The SPI is a synchronous serial communication interface specification used for short distance communication. It allows half/ full-duplex serial communication with external devices. The master device originates the frame for reading and writing. For multiple slave applications, devices are supported through selection with individual chip select (CS) lines.

17.2. Features

- Supports half/ full-duplex synchronous transfers
- 8-bit transfer frame format
- Master or slave operation
- Support Multi-master mode
- Programmable clock polarity and phase
- Master mode fault and overrun flags with interrupt capability
- 1-byte transmission and reception with DMA

17.3. Block Diagram

The Figure is the block diagram of SPI.



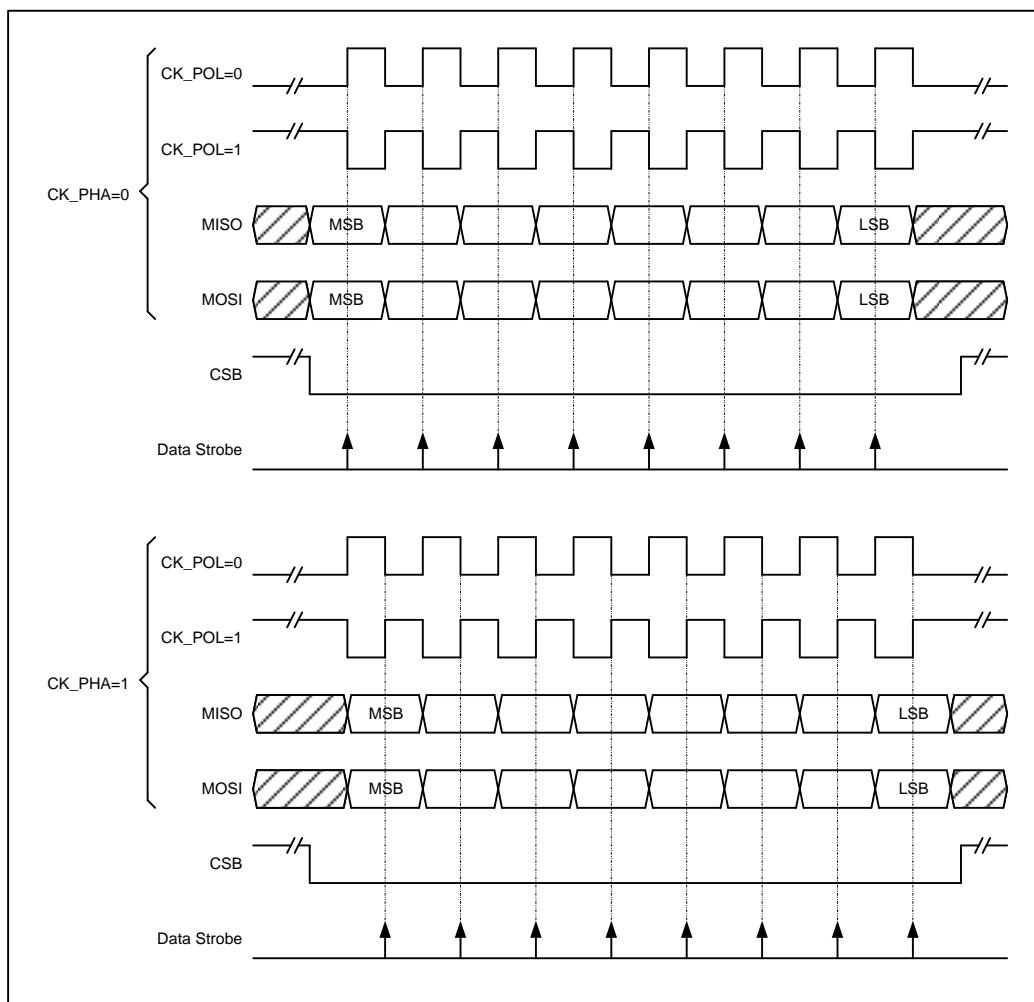
SPI block diagram

17.4. Function

17.4.1. SPI clock phase and clock polarity

A standard SPI frame consists of four possible timing relationships. That may be chosen by setting **SPIx_CTRL.CK_PHA** and **SPIx_CTRL.CK_POL**. At **SPIx_CTRL.CK_POL=0** the idle state of the clock is 0 and active state is 1. Oppositely, the idle state of the clock is 1 and active state is 0.

For clock phase choice, data are captured on the clock's rising edge and data are changed at the falling edge when **SPIx_CTRL.CK_PHA=0**. Oppositely, data are captured on the clock's falling edge and data are changed at the rising edge when **SPIx_CTRL.CK_PHA=1**. 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**

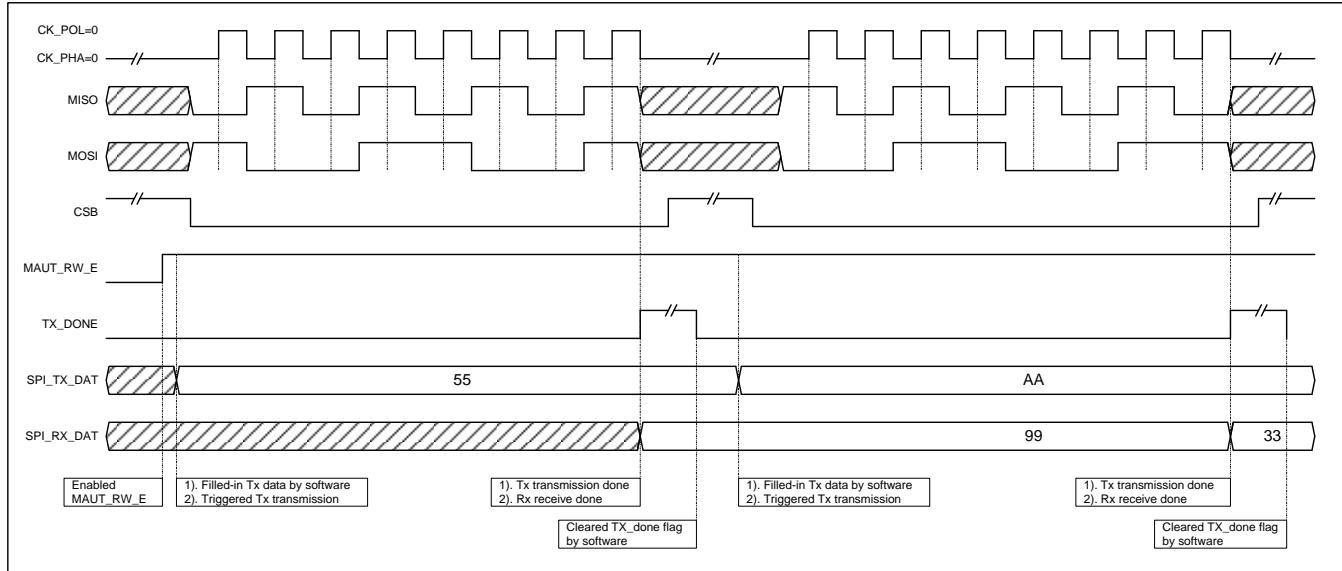
17.4.2. Master mode

17.4.2.1. Signe byte transmission with manual mode

In master mode, before using SPI transmitter function, user must set **SPIx_CTRL.MAUT_E** to 1. Then, SPI will send out data on MOSI pin immediately when CPU filled in data to **SPIx_TX_DAT**. At the same time, **SPIx_RX_DAT** were received data that from MISO pin.

During a SPI transmission, data shift out most significant bit (MSB) on the MOSI pin and shifts in data from the MISO pin. In this mode, the **SPIx_TX_DAT** register is a buffer between the system bus and the transmit shift register. And **SPIx_RX_DAT** register is

a buffer between the system bus and the receive shift register. Every frame consists of 8 bits data bit. The **SPIx_STS.TX_DONE** will be set to 1 when 8 bits data were shifted out by MOSI pin. An interrupt is generated if the **SPIx_CTRL.TX_INT_EN** bit is set. The **SPIx_STS.TX_DONE** flag is set by hardware and cleared by software. After clearing this bit, fill in the next data into **SPIx_TX_DAT** to make the next transfer. Figure shows a single byte communication with transmitting

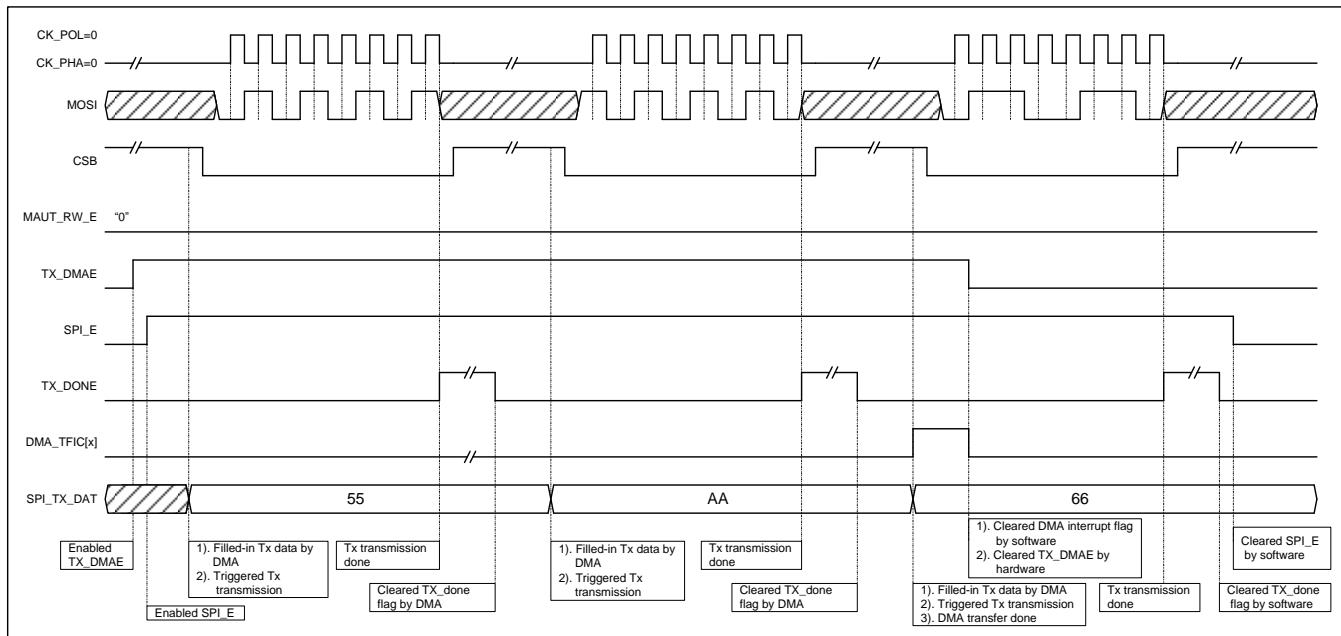


Single byte communication with manual mode (master mode)

17.4.2.2. Transmit with DMA function

The transmitter of SPI master supports DMA mode that can be enabled by setting **SPIx_CTRL.TX_DMAE** to 1. Data are loaded from SRAM and configured by DMA peripheral.

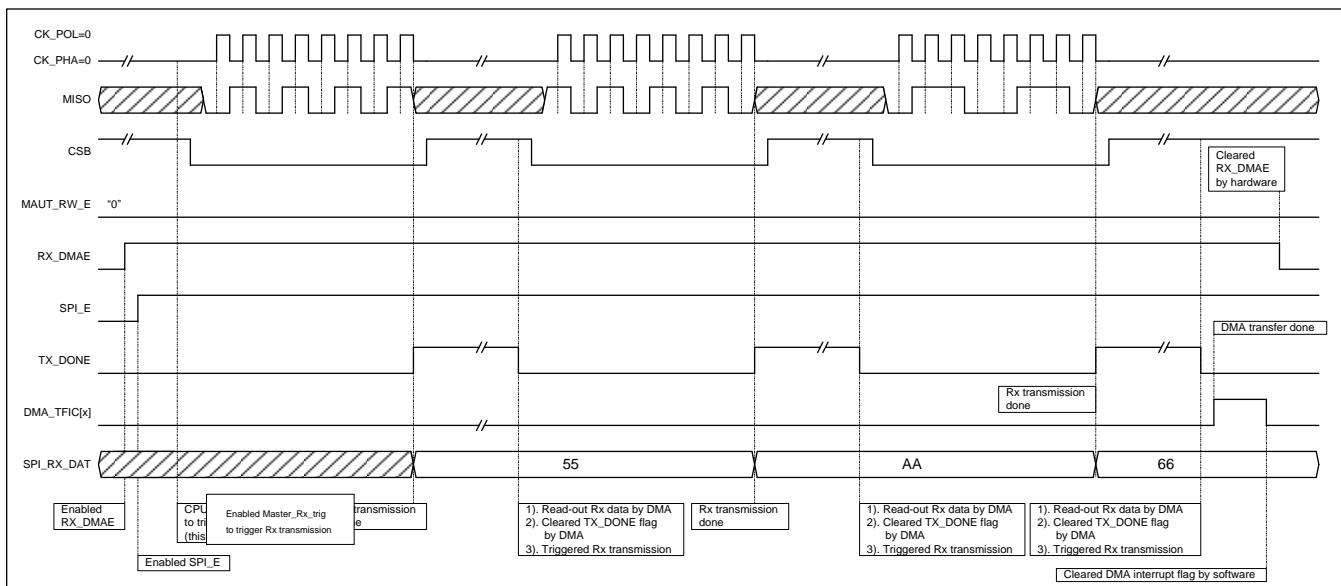
When the number of data transfers are reached, the DMA controller generates an interrupt. Once all of the transfer data have been written to **SPIx_TX_DAT** by DMA, user can monitor **SPIx_STS.TX_DONE** flag to make sure that the SPI communication is complete or not. This avoids incomplete transmission of the last data before disabling the SPI. The software must wait until **SPIx_STS.TX_DONE=1**.



SPI transmitting with DMA (master mode)

17.4.2.3. Receive with DMA function

The receiver of SPI master supports DMA function that can be enabled by setting **SPIx_CTRL.RX_DMAE** to 1. Then, controller triggers SPI interface to receive data by Enabling Master_Rx_trig. Data are stored in SRAM and configured by DMA peripheral. When the number of data transfers are reached, the DMA controller generates an interrupt. Then, user disables the SPI controller by set **SPIx_CTRL.SPI_E** to 0.



SPI receiving with DMA (master mode)

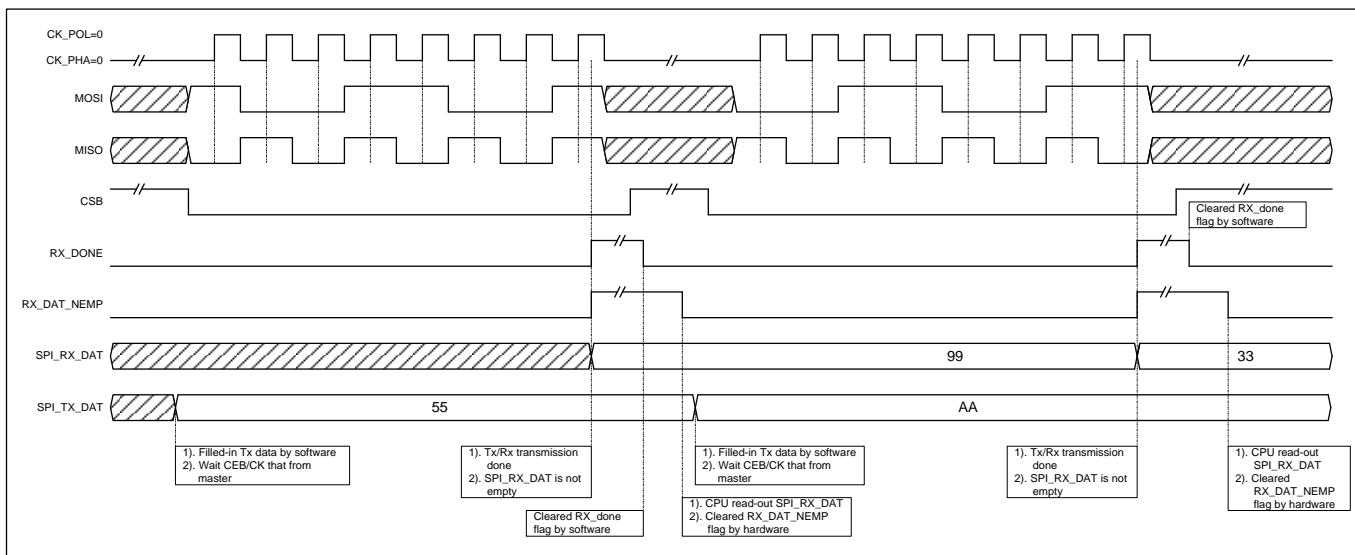
17.4.3. Slave mode

17.4.3.1. Single byte transmission with manual mode

In slave mode, before using SPI function, user must set **SPIx_CTRL.MAUT_RW_E** to 1 and filled in data to **SPIx_TX_DAT**. Then,

SPI will send out data on MISO pin immediately when controller received SPI clock input that from external master. In the same time, **SPIx_RX_DAT** were received data that from MOSI pin.

During a SPI transmission, data shift out most significant bit(MSB) on the MISO pin and shifts in data from the MOSI pin. In this mode, the **SPIx_TX_DAT** register is a buffer between the system bus and the transmit shift register. And **SPIx_RX_DAT** register is a buffer between the system bus and the receive shift register. Every frame consists of 8 bits data bit. The **SPIx_STS.RX_DONE** will be set to 1 when 8 bits data were shifted in/out by MOSI/MISO pin. An interrupt is generated if the **SPIx_CTRL.RX_INT_EN** bit is set. The **SPIx_STS.RX_DONE** flag is set by hardware and cleared by software. After clearing this bit, fill in the next data into **SPIx_TX_DAT** to make the next transfer. Figure shows a single byte communication with transmitting

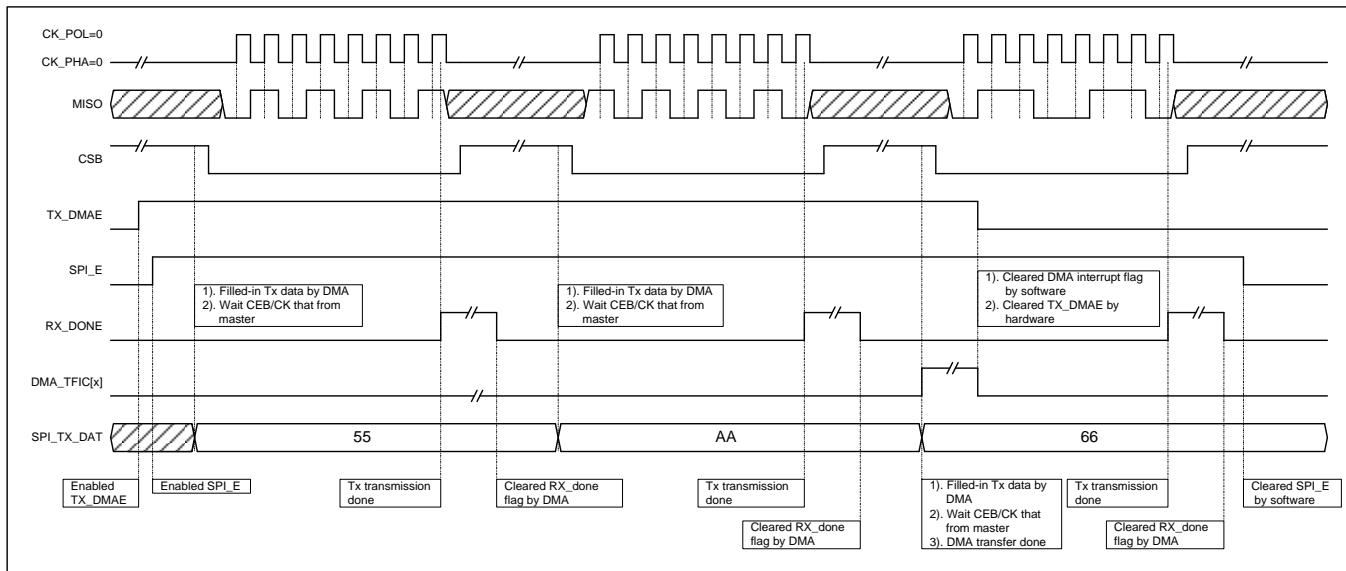


Single byte communication with manual mode (slave mode)

17.4.3.2. Transmit with DMA function

The transmitter of SPI slave supports DMA mode that can be enabled by setting **SPIx_CTRL.TX_DMAE** to 1. Transmission data are loaded from SRAM and configured by DMA peripheral.

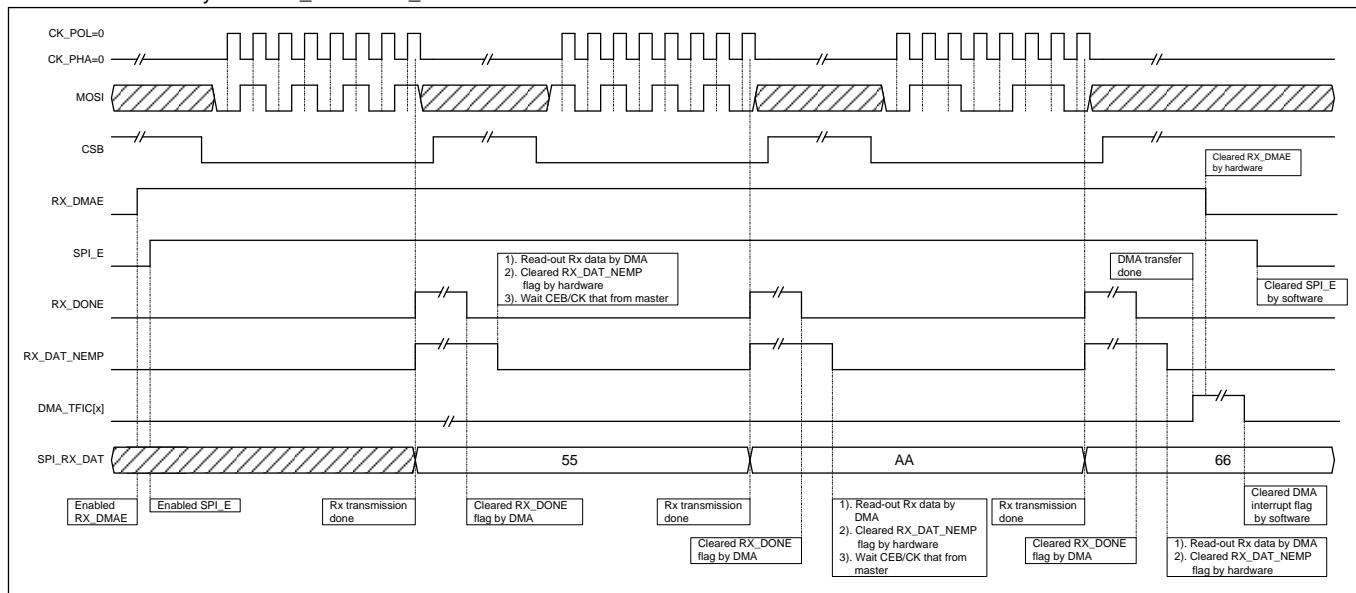
When the number of data transfers are reached, the DMA controller generates an interrupt. Once all of the transfer data have been written to **SPIx_TX_DAT** by DMA, user can monitor **SPIx_STS.RX_DONE** flag to make sure that the SPI communication is complete or not. This avoids incomplete transmission of the last data before disabling the SPI. The software must wait until **SPIx_STS.RX_DONE=1**.



SPI transmitting with DMA (slave mode)

17.4.3.3. Receive with DMA function

The receiver of SPI slave supports DMA mode that can be enabled by setting **SPIx_CTRL.RX_DMAE** to 1. **SPIx_RX_DAT** starts to receive data that from MOSI pin after set **SPIx_CTRL.SPI_E** to 1. The received data are stored in SRAM and configured by DMA peripheral. When the number of data transfers are reached, the DMA controller generates an interrupt. Then, user disables the SPI controller by set **SPIx_CTRL.SPI_E** to 0.



SPI receiving with DMA (slave mode)

17.4.3.4. Overrun error

In slave mode, when a character is received and **SPIx_STS.RX_DAT_NEMP** has not been cleared; then, an overrun error will occur. The old data that is stored in **SPIx_RX_DAT** will be replaced by new data.

17.5. Register Description

Register map

Base Address : 0x4009_0000				
Name	Description	Address	Access	Reset value
SPI0_CTRL	SPI 0 Control Register	0x4009_0000	R/W	0x3000_0800
SPI0_STS	SPI 0 Status Register	0x4009_0004	R/W	0x0000_0002
SPI0_TX_DAT	SPI 0 Transmit Data Register	0x4009_0008	R/W	0x0000_0000
SPI0_RX_DAT	SPI 0 Receive Data Register	0x4009_000C	R/W	0x0000_0000

Base Address : 0x4009_1000				
Name	Description	Address	Access	Reset value
SPI1_CTRL	SPI 1 Control Register	0x4009_1000	R/W	0x3000_0800
SPI1_STS	SPI 1 Status Register	0x4009_1004	R/W	0x0000_0002
SPI1_TX_DAT	SPI 1 Transmit Data Register	0x4009_1008	R/W	0x0000_0000
SPI1_RX_DAT	SPI 1 Receive Data Register	0x4009_100C	R/W	0x0000_0000

Register Function

SPI0_CTRL	SPI 0 Control Register	Address : 0x4009_0000
SPI1_CTRL	SPI 1 Control Register	Address : 0x4009_1000
31	30	29
MASTER_RX_T RIG	TX_DISABLE	--
23	22	21
		20
		19
		18
		17
		16
		SPI_CLK_SEL
15	14	13
		12
		11
		10
CSB_SW_EN	--	CSB_GPIO
		ERR_INT_EN
		RX_INT_EN
7	6	5
		4
		3
		2
		1
		0
--	LOOPBACK_E N	CLK_POL
		CLK_PHA
		CSB_KEEPPL
		MODE_SEL
		SPI_EN

Bit	Name	Description	Access	Reset value										
[31]	MASTER_RX_TRIG	SPI Master Rx normal/DMA trigger 1=enabled(auto clear) Note: In DMA mode, this bit must be written "1" so that DMA Rx will be enabled. (Triggered once to move one data). In Normal mode 下, this bit must also be written "1" in order to read one Rx data.	R/W	0x00										
[30]	TX_DISABLE	Tx mode disable (For SPI Master Rx to disable Tx function) 1= Tx disabled 0= Tx enabled	R/W	0x00										
[29:24]	--	Reserved	R	0										
[23:20]	SPI_CLK_SEL (CLK_SEL)	SPI Clock output selection <table border="1" data-bbox="524 1830 1191 2021"> <tr> <td>SPI_CLK_SEL[3:0]</td> <td>Frequency of SPI clock</td> </tr> <tr> <td>0000</td> <td>PCLK / 2</td> </tr> <tr> <td>0001</td> <td>PCLK / 4</td> </tr> <tr> <td>0010</td> <td>PCLK / 6</td> </tr> <tr> <td>0011</td> <td>PCLK / 8</td> </tr> </table>	SPI_CLK_SEL[3:0]	Frequency of SPI clock	0000	PCLK / 2	0001	PCLK / 4	0010	PCLK / 6	0011	PCLK / 8	R/W	0x0
SPI_CLK_SEL[3:0]	Frequency of SPI clock													
0000	PCLK / 2													
0001	PCLK / 4													
0010	PCLK / 6													
0011	PCLK / 8													

Bit	Name	Description		Access	Reset value
		0100	PCLK / 10		
		0101	PCLK / 16		
		0110	PCLK / 32		
		0111	PCLK / 64		
		1000	PCLK / 128		
		1001	PCLK / 256		
		1010	PCLK / 512		
		1011	PCLK / 1024		
[19]	--	Reserved		R	0
[18]	MAUT_RW_EN (MAUT_EN)	Manual mode Enable 0 = disabled, 1 = enabled Note: This bit is for master mode only.		R/W	0x0
[17]	RX_DMA_EN	Rx buffer DMA Enabling bit. 0 = disabled, 1 = enabled		R/W	0x0
[16]	TX_DMA_EN	Tx buffer DMA Enabling bit. 0 = disabled, 1 = enabled		R/W	0x0
[15]	CSB_SW_EN	SPI CSB control by SW(Bit.11 CSB_GPIO) enable 0=disabled 1=enabled		R/W	0x00
[14:12]	-	Reserved		R	0x0
[11]	CSB_GPIO	SPI CSB control bit 0=low 1=high		R/W	1
[10]	ERR_INT_EN	SPI Received overrun interrupt enable. 0 = disabled, 1 = enabled		R/W	0x0
[9]	RX_INT_EN	SPI Received complete interrupt enable. 0 = disabled, 1 = enabled		R/W	0x0
[8]	TX_INT_EN	SPI transmitted complete interrupt enable. 0 = disabled, 1 = enabled		R/W	0x0
[7:6]	--	Reserved		R	0
[5]	LOOPBACK_EN	SPI loop-back enable 0 = disabled 1 = enabled Note: The SPI Rx will connect SPI Tx automatically when this bit is set to 1.This can be used with self-test.		R/W	0x0
[4]	CLK_POL	SPI clock polarity 0 = Data capture on SPI clock's rising edge if CK_PHA = 0 1 = Data capture on SPI clock's falling edge if CK_PHA = 0		R/W	0x0
[3]	CLK_PHA	SPI clock phase 0 = 1 st clock transition is the 1 st data capture edge 1 = 2 nd clock transition is the 1 st data capture edge		R/W	0x0
[2]	CSB_KEEP_L	SPI CSB keeping low control 0 = SPI CSB is controlled by SPI controller 1 = SPI CSB is keeping low state		R/W	0x0
[1]	MODE_SEL	SPI operating mode select bit.0 = master mode 1 = slave mode		R/W	0x0
[0]	SPI_EN	SPI Enabling bit. 0 = peripheral disable 1 = peripheral enable		R/W	0x00

SPI0_STS	SPI 0 Status Register								Address : 0x4009_0004
SPI1_STS	SPI 1 Status Register								Address : 0x4009_1004
31	30	29	28	27	26	25	24	--	
23	22	21	20	19	18	17	16	--	
15	14	13	12	11	10	9	8	--	
7	6	5	4	3	2	1	0	--	TX_DONE_FLAG
			RX_BUF_FLAG	RX_OV_FLAG	RX_DONE_FLAG	--			

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0
[4]	RX_BUF_NEMPTY_FLA G (RX_BUF_FLAG)	Receive buffer not empty flag 0 = Rx buffer empty 1 = Rx buffer not empty	R/W	0x0
[3]	RX_OVER_RUN_FLAG (RX_OV_FLAG)	SPI received overrun flag read : 0 = no overrun occurred 1 = overrun occurred write : 0 = on effect 1 = clear this bit Note: This flag is set when data are received and the previous data have not yet been read from SPI_TXD.	R/W	0x0
[2]	RX_DONE_FLAG	SPI receive complete flag read : 0 = receiving is not complete 1 = receiving is complete write : 0 = on effect 1 = clear this bit	R/W	0x0
[1]	--	Reserved	R	0x1
[0]	TX_DONE_FLAG	SPI transmission complete flag read: 0 = transmission is not complete 1 = transmission is complete write: 0 = on effect 1 = clear this bit	R/W	0x0

SPI0_TX_DAT	SPI 0 Transmit Data Register								Address : 0x4009_0008
SPI1_TX_DAT	SPI 1 Transmit Data Register								Address : 0x4009_1008
31	30	29	28	27	26	25	24	--	

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

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7:0]	SPI_TX_DAT	The transmit data register	R/W	0x0

SPI0_RX_DAT SPI 0 Receive Data Register Address : 0x4009_000C

SPI1_RX_DAT SPI 1 Receive Data Register Address : 0x4009_100C

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

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7:0]	SPI_RX_DAT	The receive data register	R	0x0

18. SPI controller for FLASH device access (SPIFC)

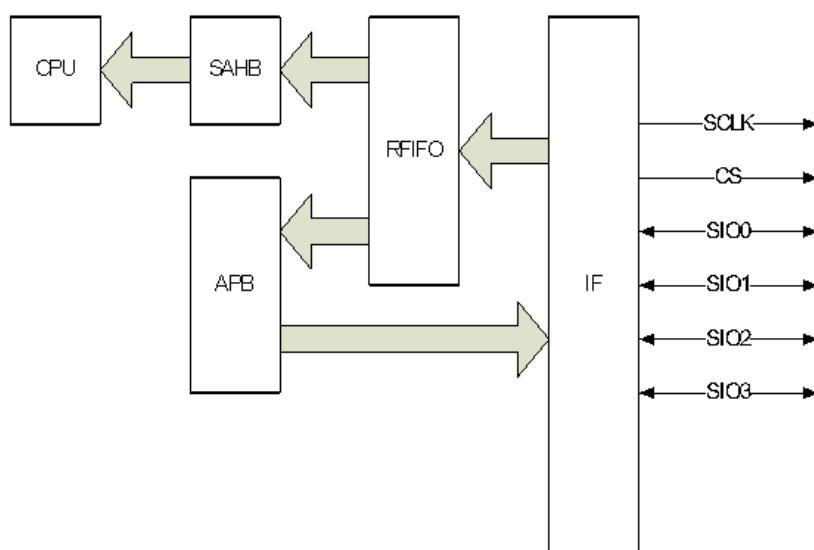
18.1. Introduction

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.

18.2. Features

- Supports 1-bit, 2-bit and 4-bit data bus
- Provides internal FIFO with 32-bit width and depth 8
- Configurable received data timing
- Configurable variable packets types

18.3. Block Diagram

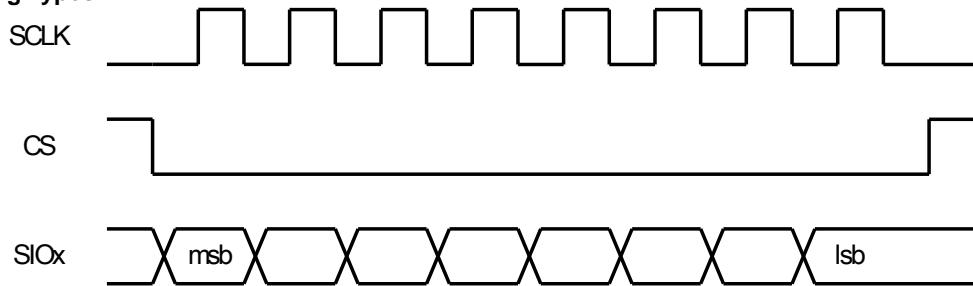


18.4. Function

Pin Descriptions

Pin name	IO	*SPIFC_IO_SEL =0	*SPIFC_IO_SEL =1	Descriptions
SCLK	O	IOB1	IOB1	Serial Clock.
CS	O	IOB3	IOB5	Chip Select for slave device.
SIO0	I/O	IOB2	IOB0	Serial data output or serial data input/output for 2 x IO mode and 4 x IO mode.
SIO1	I/O	IOB4	IOB4	Serial data input or serial data input/output for 2 x IO mode and 4 x IO mode.
SIO2	I/O	IOB5	IOB3	Serial data input/output for 4 x IO mode.
SIO3	I/O	IOB0	IOB2	Serial data input/output for 4 x IO mode.

*Note: Please refer to the P_GPIO_FUN_CTR(0x5007_0200) Bit.12 for the SPIFC_IO_SEL definition in the GPIO chapter.

Timing Types**Packet Types**

Some packet types are listed below, and these packet types could be composed by appropriate settings.

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 for packet types are listed below.

COMMAND

```
man_mode = 1
spif_cmd = xxx

wo_cmd = 0
cmd_only = 1
one_cmd = 0
wo_enhan = don't care
wo_addr = don't care
to_addr = don't care

enhan_by = don't care
rx_bc = don't care
tx_bc = don't care

Write REG_CMD to initiate transaction
Wait back2idle is asserted, and clear it.
```

COMMAND**ADDRESS**

```
man_mode = 1
spif_cmd = xxx

wo_cmd = 0
cmd_only = 0
one_cmd = 0
wo_enhan = don't care
wo_addr = 0
to_addr = 1

enhan_by = don't care
rx_bc = don't care
tx_bc = don't care

Write REG_CMD to initiate transaction
Wait back2idle is asserted, and clear it.
```

COMMAND	ADDRESS	TX
---------	---------	----

```
man_mode = 0 (DMA mode)
spif_cmd = xxx

wo_cmd = 0
cmd_only = 0
one_cmd = 0
wo_enhan = 1
wo_addr = 0
to_addr = 0
saddr = start address of transmitted data.

enhan_by = don't care
rx_bc = 0
tx_bc = transmitted byte counts

Write REG_CMD to initiate transaction
Wait back2idle is asserted, and clear it.
(For burst programming, STATUS register indicates the actual status)
```

COMMAND	ADDRESS	RX
---------	---------	----

```
man_mode = 1
spif_cmd = xxx

wo_cmd = 0
cmd_only = 0
one_cmd = 0
wo_enhan = 1
wo_addr = 0
to_addr = 0

enhan_by = don't care
rx_bc = desired byte counts (=32 bytes)
tx_bc = 0

Write REG_CMD to initiate transaction
Wait back2idle is asserted and Read desired byte counts.
Clear back2idle.
```

COMMAND	ADDRESS	ENHAN	DUMMY	RX
---------	---------	-------	-------	----

ADDRESS	ENHAN	DUMMY	RX
---------	-------	-------	----

ADDRESS	ENHAN	DUMMY	RX
---------	-------	-------	----

Wait idle = 1 for further operation.

man_mode = 0
spif_cmd = xxx

wo_cmd = 0
cmd_only = 0
one_cmd = 1
wo_enhan = 0
wo_addr = 0
to_addr = 0

enhan_by = xxx
rx_bc = non-zero
tx_bc = 0

COMMAND	TX
---------	----

man_mode = 1
spif_cmd = xxx

wo_cmd = 0
cmd_only = 0
one_cmd = 0
wo_enhan = 1
wo_addr = 1
to_addr = 0

enhan_by = don't care
rx_bc = 0
tx_bc = non-zero

Write REG_CMD to initiate transaction

Wait txb_done=1 for finishing transmission of one-byte data.
Write REG_TX_WD can clear this bit.

COMMAND	RX
----------------	-----------

```

man_mode = 1
spif_cmd = xxx

wo_cmd = 0
cmd_only = 0
one_cmd = 0
wo_enhan = 1
wo_addr = 1
to_addr = 0

enhan_by = don't care
rx_bc = desired byte counts (=32 bytes)
tx_bc = 0

Write REG_CMD to initiate transaction

Wait back2idle is asserted and Read desired byte counts.
Clear back2idle.

```

18.5. Register Description

Register Map

Base Address : 0x5006_0000				
Name	Description	Address	Access	Reset value
SPIFC_CTRL0	SPIFC control register	0x5006_0000	R/W	0x0000_0000
SPIFC_CMD	SPIFC command register	0x5006_0004	R/W	0x0000_0003
SPIFC_PARA	SPIFC parameter register	0x5006_0008	R/W	0x0000_4000
SPIFC_ADDRL	SPIFC address low register	0x5006_000c	R/W	0x0000_0000
SPIFC_ADDRH	SPIFC address high register	0x5006_0010	R/W	0x0000_0000
SPIFC_TX_DATA	SPIFC write data register	0x5006_0014	R/W	0x0000_0000
SPIFC_RX_DATA	SPIFC read data register	0x5006_0018	R	0x0000_0000
SPIFC_TX_BC	SPIFC write data count register	0x5006_001c	R/W	0x0000_0000
SPIFC_RX_BC	SPIFC read data count register	0x5006_0020	R/W	0x0000_0020
SPIFC_TIMING	SPIFC timing control register	0x5006_0024	R/W	0x0000_1700
SPIFC_CTRL1	SPIFC control register1	0x5006_002c	R/W	0x0000_0000
SPIFC_CTRL2	SPIFC control register2	0x5006_0034	R/W	0x0000_0080
SPIFC_TX_DATA32	SPIFC 32-bit Tx data register	0x5006_0038	R/W	0x0000_0000
SPIFC_USER_KEY0	SPIFC scramble User Key0 register	0x5006_004c	R/W	0x0000_0000
SPIFC_USER_KEY1	SPIFC scramble User Key1 register	0x5006_0050	R/W	0x0000_0000

Register Function

SPIFC_CTRL0 SPI Control Register								Address : 0x5006_0000	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
TX_DONE_FLAG	RX_EMP_FLAG	--				IGNORE_LCLK	OP_MODE		
7	6	5	4	3	2	1	0		
CMD_WIDTH		ADDR_WIDTH		DATA_WIDTH		--	PENDING_FLAG		

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15]	TX_DONE_FLAG	Indicate the byte transmission is done. 1: done, 0: note yet.	R/W	0
[14]	RX_FIFO_EMP_FLAG (RX_EMP_FLAG)	Indicator of RX FIFO empty. 1: FIFO is empty. 0: FIFO is not empty.	R/W	0
[13:10]	--	Reserved	R	0
[9]	IGNORE_LAST_CLK (IGNORE_LCLK)	Ignore Last clock. 0: not ignore clock, 1: ignore clock. The feature is only available for CLK_STATE = 0.	R/W	0
[8]	OPERATION_MODE (OP_MODE)	SPIFC operation mode. 0: Auto mode 1: Manual mode	R/W	0
[7:6]	CMD_WIDTH (CMIO)	IO bit width for command. See MIO for bit width definition. 00: 1-bit mode 01: 2-bit mode 10: 4-bit mode 11: Reserved	R/W	0
[5:4]	ADDR_WIDTH (AMIO)	IO bit width definition for Address, enhancement and dummy data. See MIO for bit width definition. 00: 1-bit mode 01: 2-bit mode 10: 4-bit mode 11: Reserved	R/W	0
[3:2]	DATA_WIDTH (MIO)	IO bit width definition for TX & RX data. 00: 1-bit mode 01: 2-bit mode 10: 4-bit mode 11: Reserved	R/W	0
[1]	CLK_STATE	SPI clock state. 0: low, 1: high. Note: when SPI CLK pin is not used, keep it at low or high.	R/W	0
[0]	PENDING_FLAG	Indicator of SPI operation is done. 1: Done, 0: Not Yet. Write SPIFC_SPI can clear the entity.	R	0

SPIFC_CMD SPI FLASH Command Register								Address : 0x5006_0004	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--		ONE_CMD	--	--		ONLYCMD_EN	WO_CMD		
7	6	5	4	3	2	1	0		
SPICMD									

Bit	Name	Description	Access	Reset value
[31:14]	--	Reserved	R	0
[13]	1ST_TRANS_WITHCMD _EN (ONE_CMD)	Only the first transaction has command bits. 0: Each transaction has command bits. 1: Only the first transaction has command bits.	R/W	0
[12:10]	--	Reserved	R	0
[9]	ONLYCMD_EN	A Transaction contains only command information. 0: Disabled 1: Only command information	R/W	0
[8]	WITHOUTCMD_EN (WO_CMD)	Transaction without command bits enable. 0: Disabled 1: Without command bits	R/W	0
[7:0]	SPICMD	Command bits in each transaction	R/W	0x3

SPIFC_PARA SPI FLASH Parameter Register								Address : 0x5006_0008	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
	WO_ENHAN	ADDR_ONLY	WO_ADDR	DUMMY_CLK					
7	6	5	4	3	2	1	0		
ENHANCE_BIT									

Bit	Name	Description	Access	Reset value
[31:15]	--	Reserved	R	0
[14]	ENHAN_DISABLE (WO_ENHAN)	Transaction doesn't contain enhancement bits. 1= Transaction doesn't contain enhance bits. 0= Disabled	R/W	1
[13]	ONLY_ADDR_EN (ADDR_ONLY)	Transaction only contains address bits. 1= Transaction only contains address bits. 0= Disabled	R/W	0
[12]	WITHOUT_ADDR_E N (WO_ADDR)	Transaction doesn't contain address bits. 1= Transaction doesn't contain address bits. 0= Disabled	R/W	0

Bit	Name	Description	Access	Reset value
[11:8]	DUMMY_CLK	Dummy clock cycle. 0x0= 0 dummy cycle. 0x1= 1 dummy cycles 0x2= 2 dummy cycles ... 0x9= 9 dummy cycles 0xA~F= Reserved.	R/W	0
[7:0]	ENHANCE_BIT	Enhancement bits in a transaction. Please see vendor's spec. for more descriptions.	R/W	0

SPIFC_ADDR SPI FLASH Low Address Register								Address : 0x5006_000C
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
SPIFC_ADDR_L[15:8]								
7	6	5	4	3	2	1	0	
SPIFC_ADDR_L[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	SPIFC_ADDR_L	The first 16 bits of 32 SPI FLASH address. The entity is valid if man_mode (Manual mode) is set	R/W	0

SPIFC_ADDRH SPI FLASH High Address Register								Address : 0x5006_0010
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
SPIFC_ADDR_H[31:24]								
7	6	5	4	3	2	1	0	
SPIFC_ADDR_H[23:16]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:0]	SPIFC_ADDR_H	The last 16 bits of 32 SPI FLASH address. The entity is valid if man_mode (Manual mode) is set	R/W	0

SPIFC_TX_DATA SPI FLASH Write Data Register								Address : 0x5006_0014
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	

7	6	5	4	--	3	2	1	0
TX_DATA[7:0]								

Bit	Name	Description						Access	Reset value
[31:8]	--	Reserved						R	0
[7:0]	TX_DATA	8-bit write data. Make sure the controller is idle before writing the registers						R/W	0

SPIFC_RX_DATA SPI FLASH Read Data Register								Address : 0x5006_0018	
31	30	29	28	27	26	25	24		
RX_DATA [31:24]									
23	22	21	20	19	18	17	16		
RX_DATA [23:16]									
15	14	13	12	11	10	9	8		
RX_DATA [15:8]									
7	6	5	4	3	2	1	0		
RX_DATA [7:0]									

Bit	Name	Description						Access	Reset value
[31:0]	RX_DATA	Get data from SPI FLASH.						R	0

SPIFC_TX_BC SPI FLASH Write Data Count Register								Address : 0x5006_001c	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
TX_BC[7:0]									

Bit	Name	Description						Access	Reset value
[31:9]	--	Reserved						R	0
[8:0]	TX_BC	Transmitted byte counts of a transaction						R/W	0

SPIFC_RX_BC SPI FLASH Read Data Count Register								Address : 0x5006_0020	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8		
--									
7	6	5	4	3	2	1	0		
RX_BC[7:0]									

Bit	Name	Description	Access	Reset value
[31:9]	--	Reserved	R	0
[8:0]	RX_BC	Received byte counts of a transaction. If man_mode (Manual mode) is clear, the bit 0 & 1 should be zero (Note: Auto mode is in a unit of Word(1 word=4-byte)so, Bit[1:0] = 00)	R/W	0x020

SPIFC_TIMING SPI FLASH Timing Control Register								Address : 0x5006_0024
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	CS_HI_CNT
SMP_DELAY				--	SMP_CLK_EDGE_SEL			

Bit	Name	Description	Access	Reset value
[31:11]	--	Reserved	R	0x2
[10:8]	CS_HI_CNT	CS keeps high for duration = (CS_HI_CNT +1)*System clock. Note: This is the time to be kept between the interval of when the entire data being read and after CS pin becoming high (SPI FLASH disabled), and the next data to be read (CS pin keep high for power saving issues).	R/W	0x7
[7:4]	SMP_DELAY	Sample clock delay for Received data. 0: sample clock without delay 1: sampled clock delay 1 delay_cell ... N: sampled clock delay N delay_cell Note: 1 cell time = 1ns	R/W	0x0
[3]	--	Reserved	R	0
[2:0]	SMP_CLK_EDGE_SEL	Sample timing for Received data. 0: sampled at 1 st clock edge. 1: sampled at 2 nd clock edge. ... N: sampled at (N+1)th clock edge. nsaction. If man_mode is clear, the bit 0 & 1 should be zero	R/W	0x0

SPIFC_CTRL1 SPIFC Control Register 1								Address : 0x5000_002c
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	--
15	14	13	12	11	10	9	8	--
7	6	5	4	3	2	1	0	4BYTES_ADDR_EN
				--				SPIFC_EN

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15:8]	4BYTES_ADDR_EN	4 bytes address mode enable Bit 0: SPIf_addr[24] enable signal. 1: enabled, 0: disabled. Bit 1: SPIf_addr[25] enable signal. 1: enabled, 0: disabled. Bit 2: SPIf_addr[26] enable signal. 1: enabled, 0: disabled. Bit 3: SPIf_addr[27] enable signal. 1: enabled, 0: disabled. Bit 4: SPIf_addr[28] enable signal. 1: enabled, 0: disabled. Bit 5: SPIf_addr[29] enable signal. 1: enabled, 0: disabled. Bit 6: SPIf_addr[30] enable signal. 1: enabled, 0: disabled. Bit 7: SPIf_addr[31] enable signal. 1: enabled, 0: disabled. Note: If SPI FLASH size exceeds 128 Mb, it is necessary to issue a 4-byte address mode to make all bit[15:8] to "1". Once the entity is not zero, the 32-bit addressing is activated.	R/W	0x0
[7:1]	--	Reserved	R	0
[0]	SPIFC_EN	SPF FLASH controller Enabling bit. 0: Disabled 1: Enabled	R/W	0x0

SPIFC_CTRL2 **SPIFC Control Register 2**
Address : 0x5006_0034

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

Bit	Name	Description	Access	Reset value
[31:10]	--	Reserved	R	0
[9:8]	CLK_SEL	SPI-FLASH clock divide. The SPI_FLASH clock is divided by the controller SPI source clock, 00: divide by 1. (SPI_FLASH clock = source clock) 01: divide by 2. 10: divide by 3. 11: divide by 4	R/W	0x0
[7:0]	CS_HOLD_CNT	SPI-FLASH CS# hold cycle. After the SPI_FLASH read transaction done, the controller will keep the CS# in low state till the CS_HOLD_CNT count done or next non-continuous address access. If the CS_HOLD_CNT = 0xff, the CS# will keep low till the next non-continuous address access	R/W	0x80



GPCM1F SERIES PROGRAMMING GUIDE

SPIFC_TX_DATA32 SPI FLASH 32-bit Tx data Register

Address : 0x5006_0038

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

Bit	Name	Description	Access	Reset value
[31:0]	TX_DATA	32-bit write data. Make sure the controller is idle before writing the registers	R/W	0

SPIFC_USER_KEY0 Scramble USER Key0 Register

Address : 0x5006_004C

31	30	29	28	27	26	25	24
user_key0[31:24]							
23	22	21	20	19	18	17	16
user_key0[23:16]							
15	14	13	12	11	10	9	8
user_key0[15:10]							
7	6	5	4	3	2	1	0
user_key0[7:0]							

Bit	Name	Description	Access	Reset value
[31:0]	user_key0	Scramble user_key0. This register set the SPI-FLASH controller scramble key0. – This key combine with user_key1 and system-scramble-key to generate scramble key. Note : To enable the user key function, both user_key0 and user_key1 must not to be "0"	R/W	0x0000_0000

SPIFC_USER_KEY1 Scramble USER Key1 Register

Address : 0x5006_0050

31	30	29	28	27	26	25	24
user_key1[31:24]							
23	22	21	20	19	18	17	16
user_key1[23:16]							
15	14	13	12	11	10	9	8
user_key1[15:10]							
7	6	5	4	3	2	1	0
user_key1[7:0]							

Bit	Name	Description	Access	Reset value
[31:0]	user_key1	Scramble user_key0. This register set the SPI-FLASH controller scramble key1. – This key	R/W	0x0000_0000



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
		combine with user_key0 and system-scramble-key to generate scramble key. Note : To enable the user key function, both user_key0 and user_key1 must not to be "0"		

19. Universal asynchronous receiver transmitter (UART)

19.1. Introduction

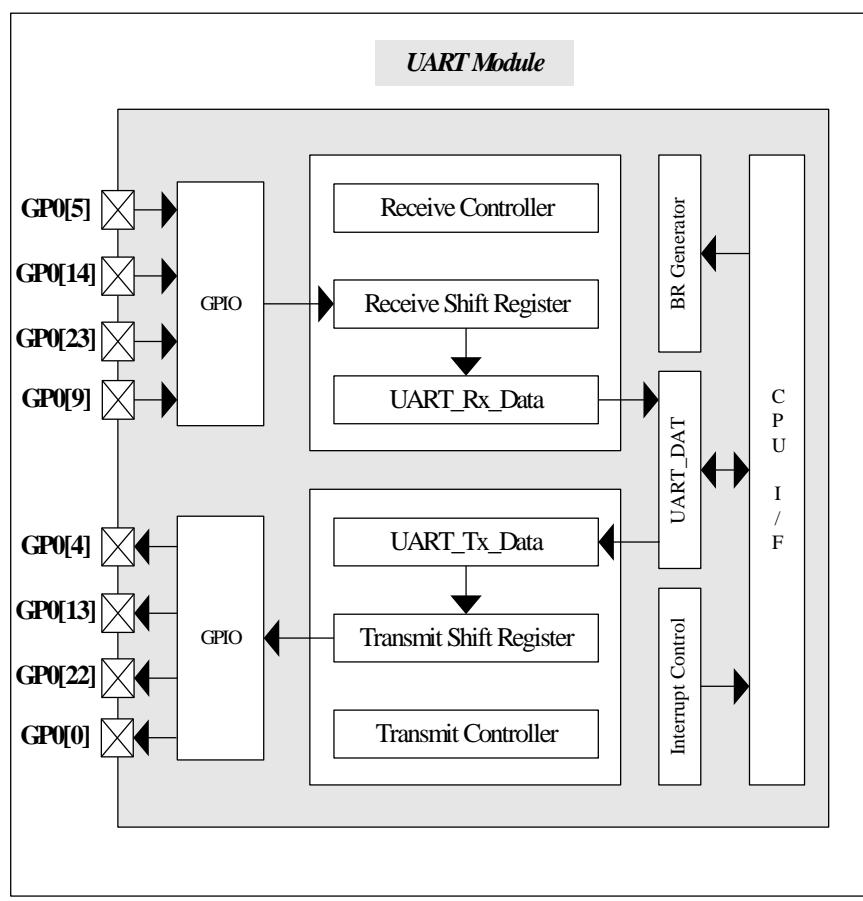
The asynchronous serial channel includes the reception and the transmission frames. The receiver and transmitter are independent, frames can start at different points in time for transmission and reception. The UART provides a wide range of baud rates by baud rate generator.

It also supports half-duplex single wire communication.

19.2. Features

- Half duplex, Tx & 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

19.3. Block Diagram



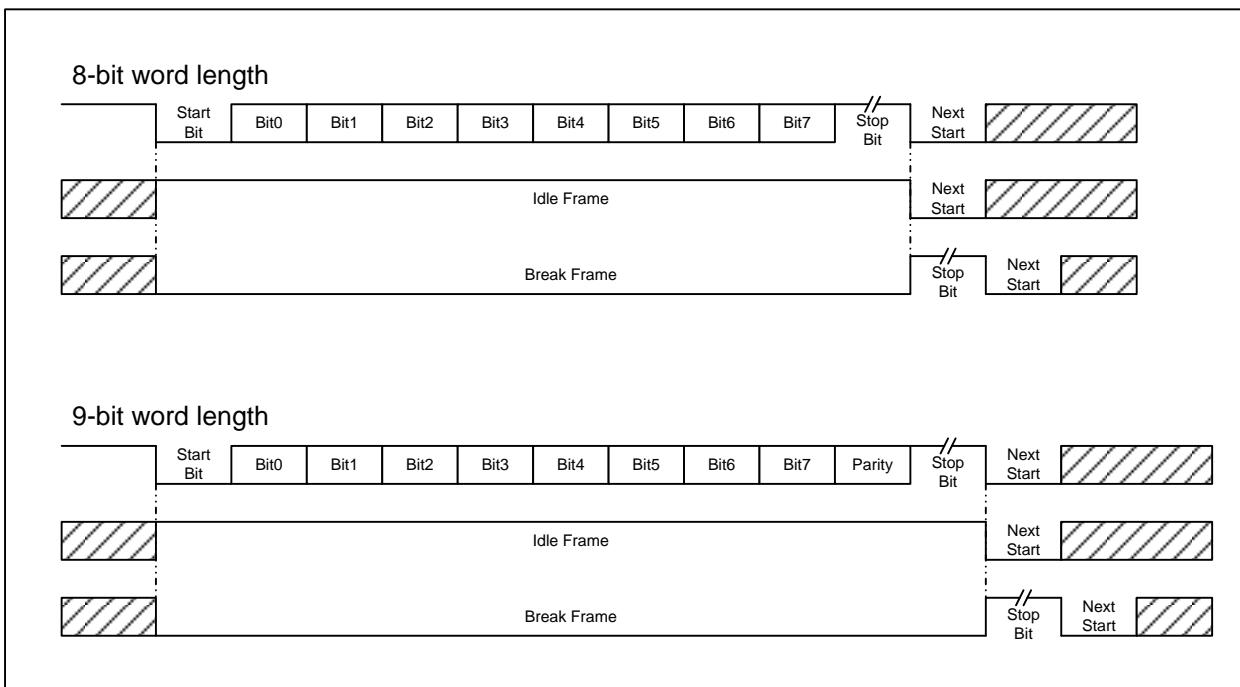
UART block diagram

19.4. Function

19.4.1. Frame Format

A standard UART frame is shown in below figure. It consists of :

- An idle character with the signal level 1.
- One start bit with the signal level 0.
- A data frame.
- A parity bit, it 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.



The frame format of UART

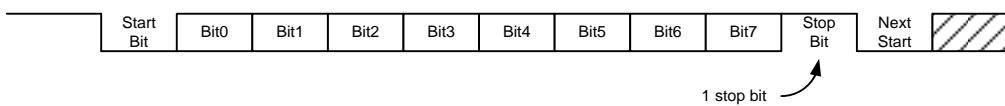
19.4.2. Transmitter

The transmitter can send data words of either 8 or 9 bits depending on 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** to 1. Then fill in data to

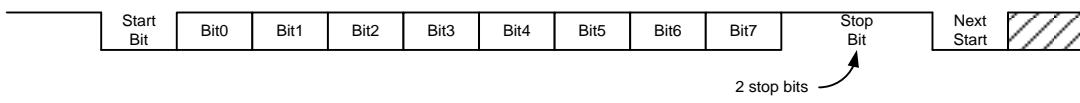
UART_DATA. When the baud-rate count is overflow, the data in the transmit shift register is output on the **UART_TX** pin.

During a UART transmission, data shifts out least significant bit (LSB) on the **UART_TX** pin. 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 a logic level low for one bit period. The character is terminated by a configurable number of stop bit. The number of stop bit is programmable by **UART_CTRL.STOP_SEL** and new start bit can be transferred directly after the last stop bit.

8-bit word length with 1 stop bit



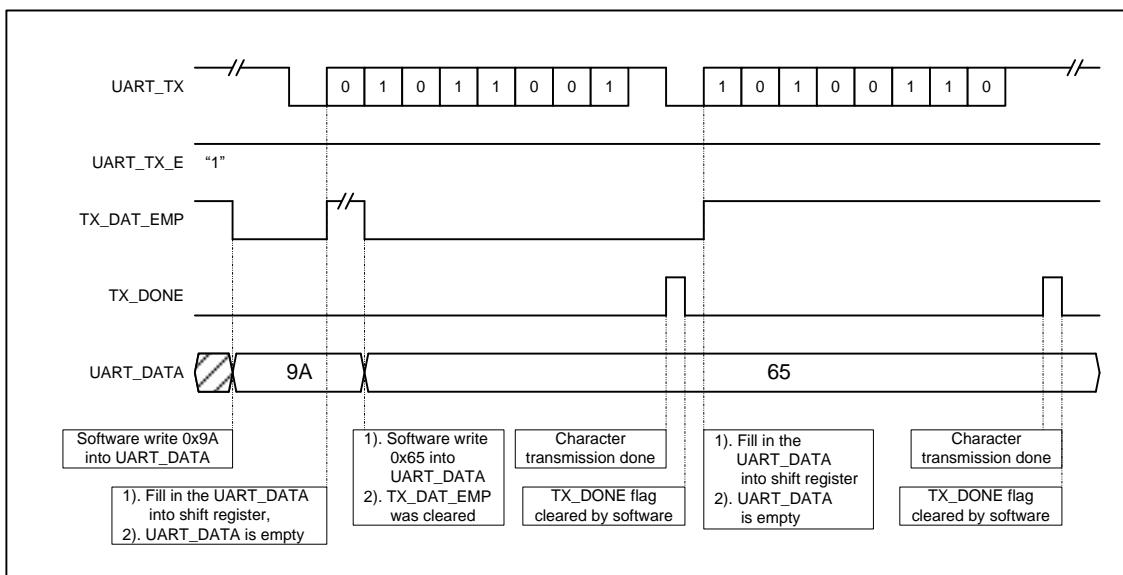
8-bit word length with 2 stop bits



Configurable stop bit

19.4.3. Single byte transmission

After **UART_CTRL.UART_EN** and **UART_CTRL.UART_TX_EN** were set to 1. UART will send out data immediately when host (CPU or DMA) fill in data to **UART_DATA**. The **UART_STS.TX_DAT_EMP_FLAG** bit is always cleared by a write to the **UART_DATA** register. This bit is set by hardware; it indicates the data have been moved from **UARTx_DATA** to the shift register and the data transmission has started. Then, the **UART_DATA** register is empty. It allows user to write in next data to **UART_DATA** register. In addition, **UART_STS.TX_DAT_EMP_FLAG** can trigger an interrupt to CPU if **UART_CTRL.INTE** is enabled.

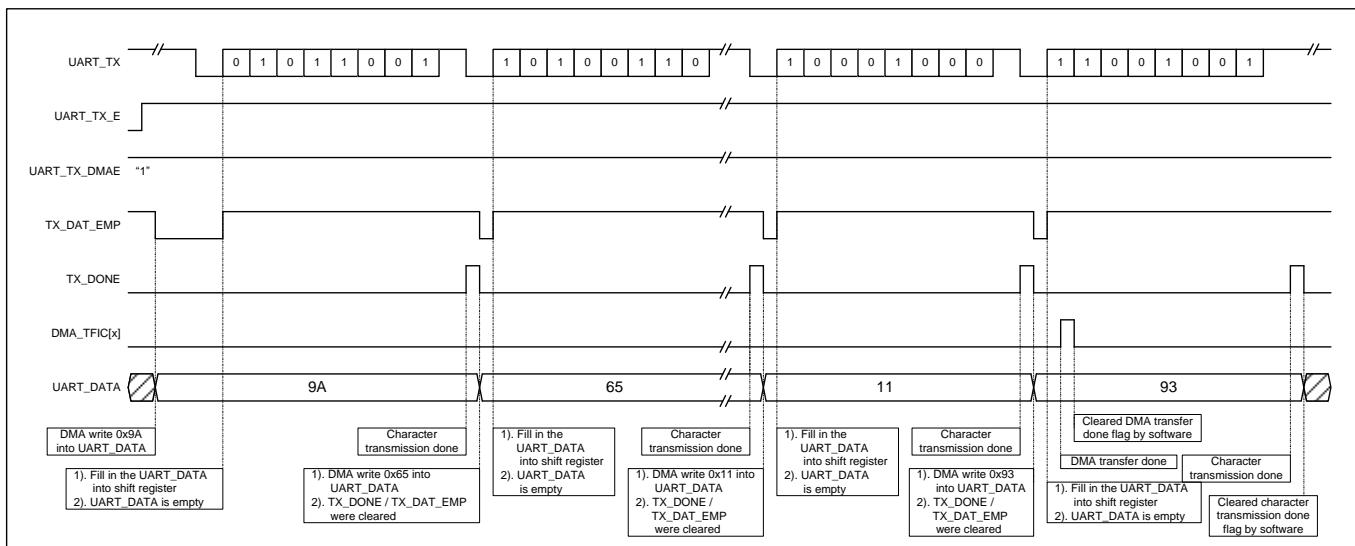


Single byte communication with transmitting

19.4.4. Transmit with DMA function

In GPCM1Fxxx, UART supports DMA mode that can be enabled by setting **UART_CTRL.TX_DMAE**. Data is loaded from 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 have been written to **UART_DATA** by DMA, user can monitor **UART_STS.TX_DONE_FLAG** flag to make sure that the UART communication is complete or not. This avoids incomplete transmission of the last data before disabling the UART. The software must wait until **UART_STS.TX_DONE_FLAG=1**.



UART transmitting with DMA

19.4.5. Receiver

In GPCM1F, The receiver can receive data words of either 8 or 9 bits depending on the **UART_CTRL.PARITY_BIT_EN** bit status.

19.4.6. Baud Rate Generation

In GPCM1Fxxx, 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, hardware supports a technique for clock compensation.

For example:

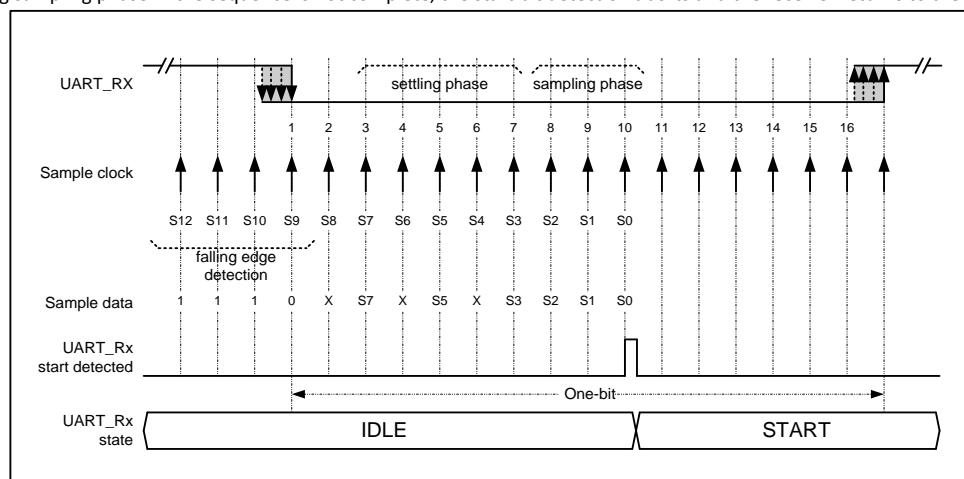
PCLK = 16MHz, Baud rate = 115200

So, **UART_BAUD_RATE.BR_DIV** = $16M / (16 * 115200) = 8.6805555$

User choice BR_DIV = 9, that means the baud rate of receiver is slower than 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.

19.4.7. Start Bit Detection

In GPCM1F, the start bit of UART is detected when a specific sequence of samples is recognized. In settling phase, 3-bits sample data (S5, S7, and S3) at least 2 out of the 3 sampled bits are at 0. At the same time, 3-bits sample data (S2, S1, and S0) at least 2 out of the 3 sampled bits are at 0 during sampling phase. If the sequence is not complete, the start bit detection aborts and the receiver returns to the idle state.

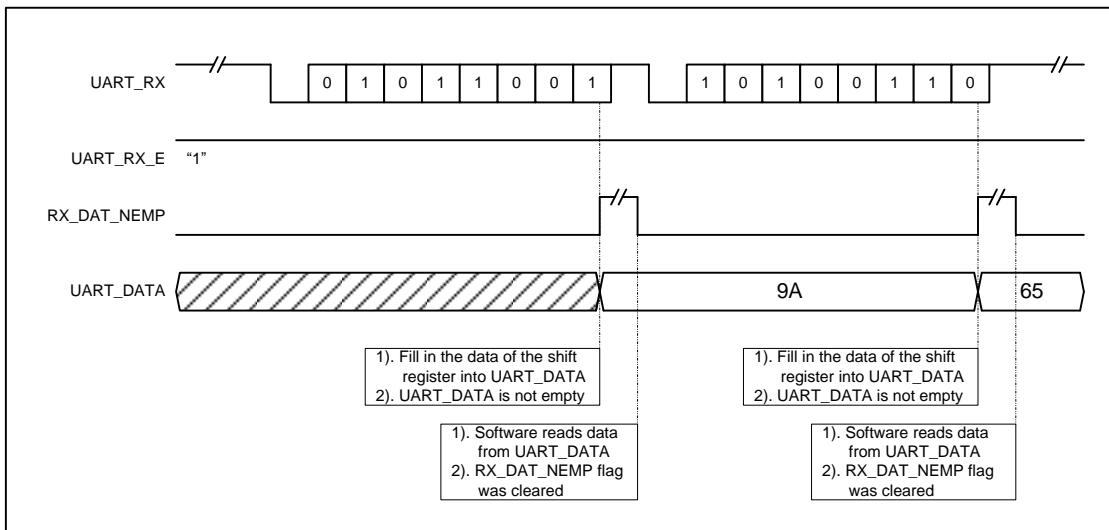


Start bit detection

19.4.8. Sign byte receive

After **UART_CTRL.UART_EN** and **UART_CTRL.UART_RX_EN** are set to 1, data shifts in through the **UART_RX** pin. The **UART_STS.RX_DAT_NEMP_FLAG** bit is always cleared by a read from the **UART_DATA** register. This bit is set by hardware; it indicates the data has been moved from shift register to **UART_DATA** and the data transmission has finished. Then, the **UART_DATA** register is not empty. It allows user 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.INTE** is enabled.

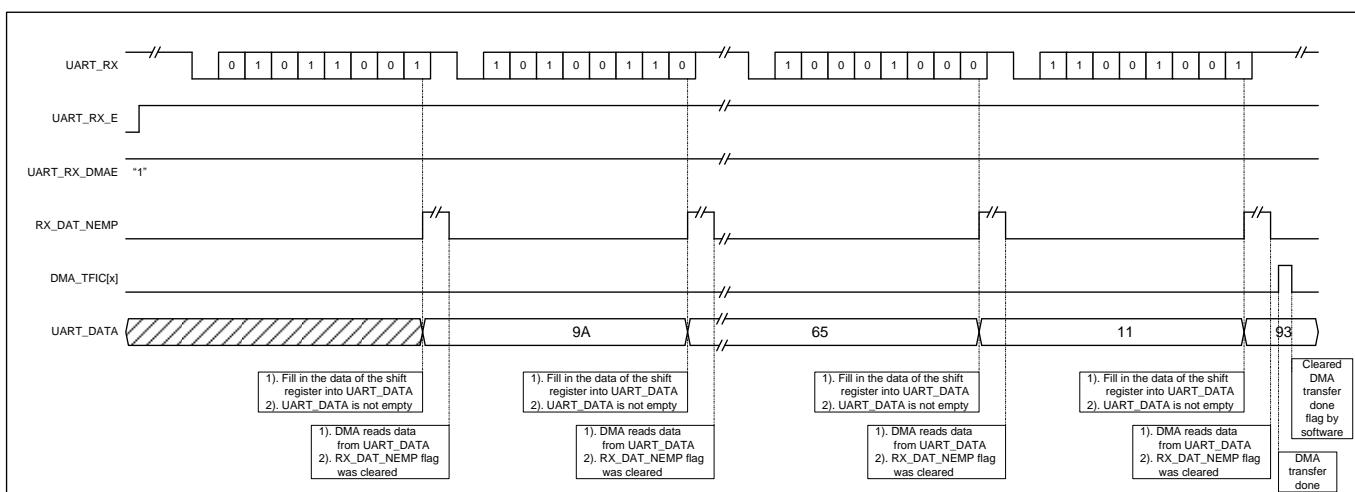


Single byte communication with receiving

19.4.9. Receive with DMA function

In GPCM1Fxxx, UART supports DMA mode that can be enabled by setting **UART_CTRL.TX_DMAE**. Data are stored in SRAM and configured by DMA peripheral.

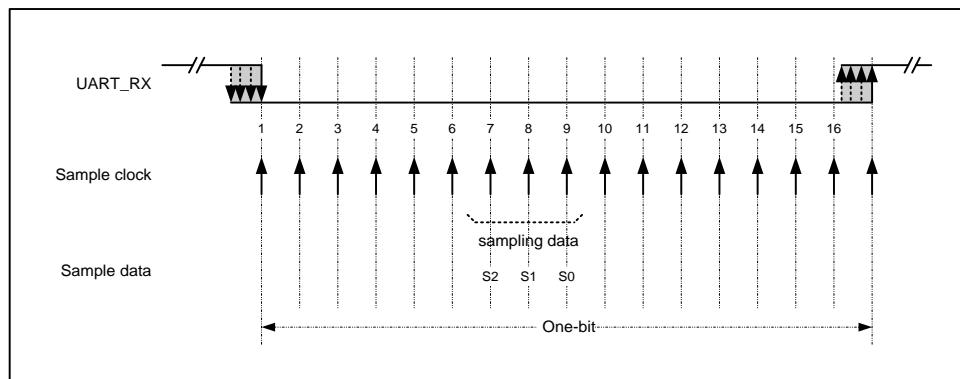
When the number of data transfers is reached, the DMA controller generates an interrupt. Once all of the transfer data has written to SRAM by DMA, user can monitor **UART_STS.TX_DONE_FLAG** flag to make sure that the UART communication is complete or not.



UART receiving with DMA

19.4.10. Noise error reduction

For data recovery, over-sampling techniques are used to distinguish valid data or noise. The hardware selects the bit[9:7] of the sampled data for inspection. The valid data that must meet 3-bits sample data at least 2 out of the 3 sampled bits are at 1. Otherwise, the data is 0.



Noise error reduction

19.4.11. 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 get the attention of a mismatched receiver. When a break frame is detected, receiver can generate an interrupt if the **UART_CTRL.RX_BRK_FLAG_INTEN** bit is set to 1.

19.4.12. 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, receiver can generate an interrupt if the **UART_CTRL.RX_IDLE_FLAG_INTEN** bit is set to 1.

19.4.13. Overrun error

When a character is received and **UART_STS.RX_DAT_NEMP_FLAG** has not been cleared. An overrun error will occur. The old data that is stored in **UART_DATA** will be replaced by new data.

19.4.14. Parity check

Parity check can be enabled by setting the **UART_CTRL.PARITY_CHK_EN** to 1. The even/odd parity type is selected by setting **UART_CTRL.PARITY_SEL**. For even parity, the parity is calculated to obtain an even number of 1 inside a frame. For odd parity, the parity is calculated to obtain an odd number of 1 inside a frame.

19.5. Register Description

Register map

Base Address : 0x400A_0000				
Name	Description	Address	Access	Reset value
UART_CTRL	UART Control Register	0x400A_0000	R/W	0x0000_0000
UART_STS	UART Status Register	0x400A_0004	R/W	0x0000_0002
UART_BAUD_RATE	UART Baud Rate Register	0x400A_0008	R/W	0x0000_0000
UART_DATA	UART Data Shadow Register	0x400A_000C	R/W	0x0000_0000

Register Function

UART_CTRL JART Control Register								Address : 0x400A_0000	
31	30	29	28	27	26	25	24	--	
--	--	--	--	--	--	--	--	--	
23	22	21	20	19	18	17	16	--	
--	--	RX_DMAE	TX_DMAE	LOOPBACK_EN	STOP_SEL	SEND_IDLE	--	--	
15	14	13	12	11	10	9	8	--	
RX_BRK_FLAG_IN_T_EN	RX_IDLE_FLAG_IN_TEN	RX_INT_EN	TX_INT_EN	PARITY_SEL	PARITY_CHK_EN	PARITY_BIT_EN	UART_EN	--	
7	6	5	4	3	2	1	0	UART_RX_EN	
--								UART_TX_EN	

Bit	Name	Description	Access	Reset value
[31:22]	--	Reserved	R	0
[21]	RX_DMA_ENABLE (RX_DMAE)	Rx buffer DMA enable 0 = disabled, 1 = enabled	R/W	0x0
[20]	TX_DMA_ENABLE (TX_DMAE)	Tx buffer DMA enable 0 = disabled, 1 = enabled	R/W	0x0
[19]	LOOPBACK_ENABLE (LOOPBACK_EN)	UART loop-back enable 0 = disabled, 1 = enabled Note: The UART Rx will connect UART Tx automatically when this bit is set to 1. This can be used with self-test.	R/W	0x0
[18]	STOP_SEL	STOP bits select 0 = 1STOP bit 1 = 2 STOP bit	R/W	0x0
[17]	SEND_IDLE	UART send idle enable 0 = disabled, 1 = enabled	R/W	0x0
[16]	--	Reserved	R	0
[15]	RX_BRK_FLAG_INT_EN	UART received break interrupt enable 0 = disabled, 1 = enabled	R/W	0x0
[14]	RX_IDLE_FLAG_INT_EN	UART detected idle interrupt enable 0 = disabled, 1 = enabled	R/W	0x0
[13]	RX_INT_EN	UART received a word interrupt enable 0 = disabled, 1 = enabled	R/W	0x0
[12]	TX_INT_EN	UART transmission done interrupt enable 0 = disabled, 1 = enabled	R/W	0x0
[11]	PARITY_SEL	Parity selection 0 = even parity 1 = odd parity	R/W	0x0
[10]	PARITY_CHK_EN	Parity check enable (For Rx) 0 = disabled, 1 = enabled Note: The parity bit will be sent or check when PARITY_BIT_EN and PARITY_CHK_EN are set to 1.	R/W	0x0
[9]	PARITY_BIT_ENABLE (PARITY_BIT_EN)	Parity bit enable (For Tx) 0 = Disabled(1 Start bit, 8 Data bits, n Stop bit) 1 = Enabled(1 Start bit, 8 Data bits, 1 Parity bit , n Stop bit)	R/W	0x0
[8]	UART_EN	UART enable bit 0 = disabled, 1 = enabled Note: This bit is set and cleared by software	R/W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
[7:2]	--	Reserved	R	0
[1]	UART_RX_EN	UART receiver enable 0 = disabled, 1 = enabled	R/W	0x00
[0]	UART_TX_EN	UART transmitter enable 0 = disabled, 1 = enabled	R/W	0x00

UART_STS UART Status Register								Address : 0x400A_0004
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
RX_BRK_FLAG	RX_IDLE_FLAG	RX_DAT_NEMP_FLAG	TX_DONE_FLAG	--	--	--	--	RX_STP_ERR_FLAG
7	6	5	4	3	2	1	0	
RX_PARITY_ERR_FLAG	RX_PRITY	RX_OV_RUN_FLAG	--	--	--	TX_DAT_EMP_FLAG	--	

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0
[15]	RX_BRK_FLAG	UART break detected flag read: 0 = no break character is detected 1 = break character is detected write: 0 = on effect 1 = clear this bit Note: This flag is used to receiver mode only.	R/W	0x0
[14]	RX_IDLE_FLAG	UART idle line detected flag read: 0 = No Idle Line is detected 1 = Idle Line is detected write: 0 = on effect 1 = clear this bit Note: This flag is used to receiver mode only.	R/W	0x0
[13]	RX_DAT_NEMP_FLAG	UART Rx data not empty flag 0 = Not ready 1 = Ready (This bit is set to 1 when received data is ready to be read.) Note: This bit automatically cleared to 0 when the CPU or DMA reads UART_DATA.	R/W	0x0
[12]	TX_DONE_FLAG	UART transmission complete flag read: 0 = Transmission is not complete 1 = Transmission is complete write: 0 = on effect	R/W	0x0

Bit	Name	Description	Access	Reset value
	--	1 = clear this bit		
[11:9]	--	Reserved	R	0
[8]	RX_STP_ERR_FLAG	UART receiver stop bit error flag read: 0 = No stop bit error 1 = stop bit error write: 0 = on effect 1 = clear this bit	R/W	0x0
[7]	RX_PARITY_ERR_FLAG	UART receiver parity error flag read: 0 = No parity error 1 = Parity error write: 0 = on effect 1 = clear this bit	R/W	0x0
[6]	RX_PRITY	UART receiver parity bit flag Note: This bit is available when UART_CTRL.PARITY_BIT_EN and UART_CTRL.PARITY_CHK_EN are set to 1.	R	0x0
[5]	RX_OV_RUN_FLAG	UART over run error flag read: 0 = No overrun error 1 = Overrun error is detected write: 0 = on effect 1 = clear this bit Note: This flag is used to receiver mode only. The content of UART_DATA will be overwritten by new data when this bit is set to 1.	R/W	0x0
[4:2]	--	Reserved	R	0
[1]	TX_DAT_EMP_FLAG	UART transmit data register empty flag This bit is set to 1 when 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 to 0 when the CPU or DMA writes UART_DATA.	R/W	0x1
[0]	--	Reserved	R	0

UART_BAUD_RATE UART Baud Rate Register
Address : 0x400A_0008

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

Bit	Name	Description	Access	Reset value
[31:21]	--	Reserved	R	0
[20]	BR_CMP_SEL	Baud rate compensate selection 0 = add counting cycle 1 = sub counting cycle	R/W	0x0
[19:16]	BR_CMP_CYCLE	Baud rate compensate cycles Example: PCLK = 16MHz , Baud rate = 115200 , So, BR_DIV = 16M / (16 * 115200) = 8.6805555 User choice BR_DIV = 9, that means the baud rate of receiver is slower than 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.	R/W	0x0
[15:12]	--	Reserved	R	0
[11:0]	BR_DIV	Baud rate divider Tx / Rx Baud Rate = PCLK / (16 * BR_DIV[11:0]) BR_DIV[11:0] = PCLK / (16 * Baud Rate)	R/W	0x0

UART_DATA UART Data Shadow Register Address : 0x400A_000C

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

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7:0]	UART_DAT	The Received or Transmitted data register	R/W	0x00

20. Inter integrated circuit interface (I2C)

20.1. Introduction

The I²C is used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance.

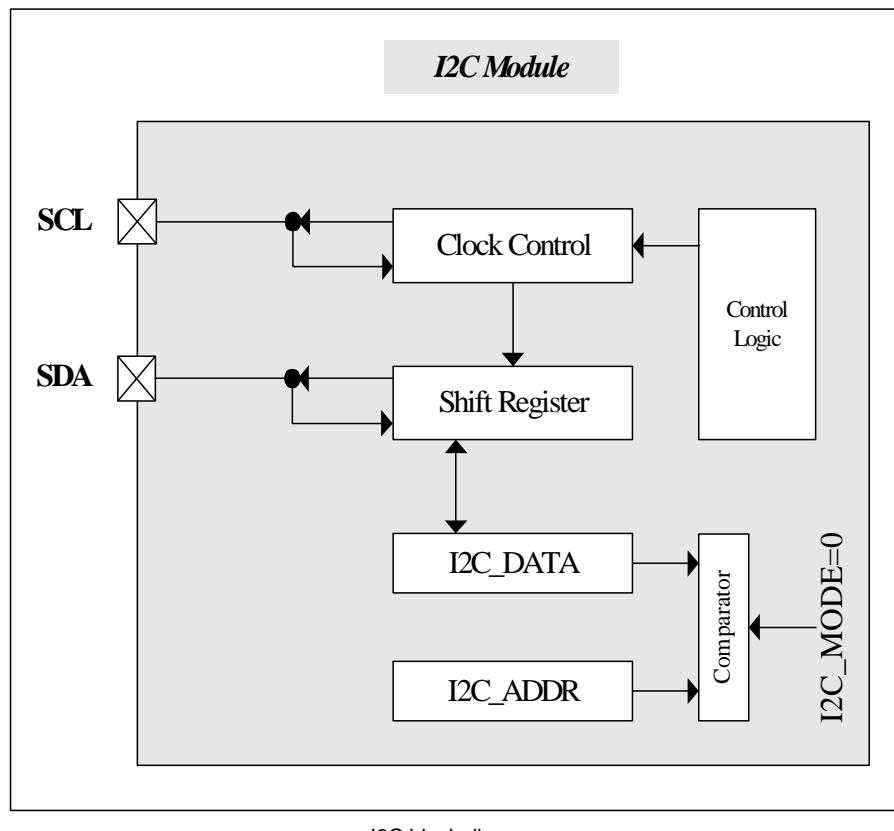
The I²C uses only two bidirectional open-drain lines, that are serial data line (SDA) and serial clock line (SCL). Logic 0 is output by sinking the bus to ground, and logic 1 is output by letting the bus to floating state, and via the pull-up resistor pulls it to high.

20.2. Features

- Multi-master capability
- Supports I²C master and slave mode
- Generation and detection of 7-bit/10-bit addressing and General Call
- Optional clock
- Supports DMA capability

20.3. Block Diagram

The Figure is the block diagram of I²C.



I2C block diagram

20.4. Function

I2C supports 7-bit or 10-bit (depending on the device used) 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 bus owner. In I2C controller, four transfer modes are supported:

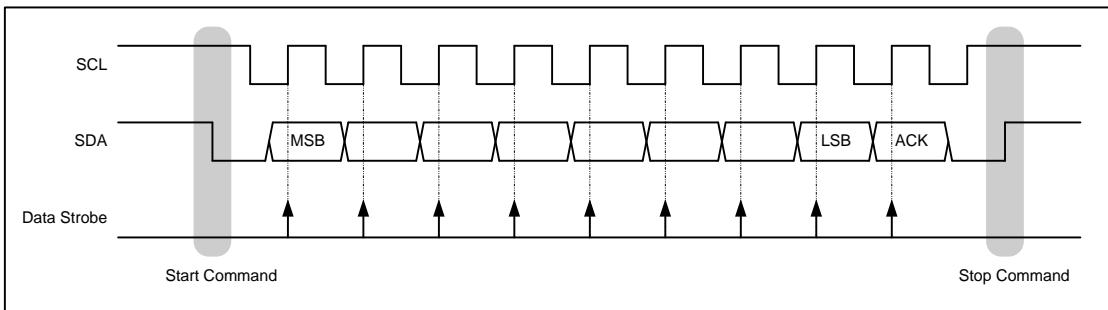
- master transmit – master is sending data to a slave
- master receive – master is receiving data from a slave
- slave transmit – slave is sending data to the master
- slave receive – slave is receiving data from the master

• GPCM1F has three IO pin options for I2C0 I/F:

- (a) IOA[0] (CLK) & IOA[1] (Data) pins.
- (b) IOA[13] (CLK) & IOA[17] (Data) pins.
- (c) IOA[15] (CLK) & IOA[16] (Data) pins.

Formats of I2C frame

Figure shows the frame format of I2C.

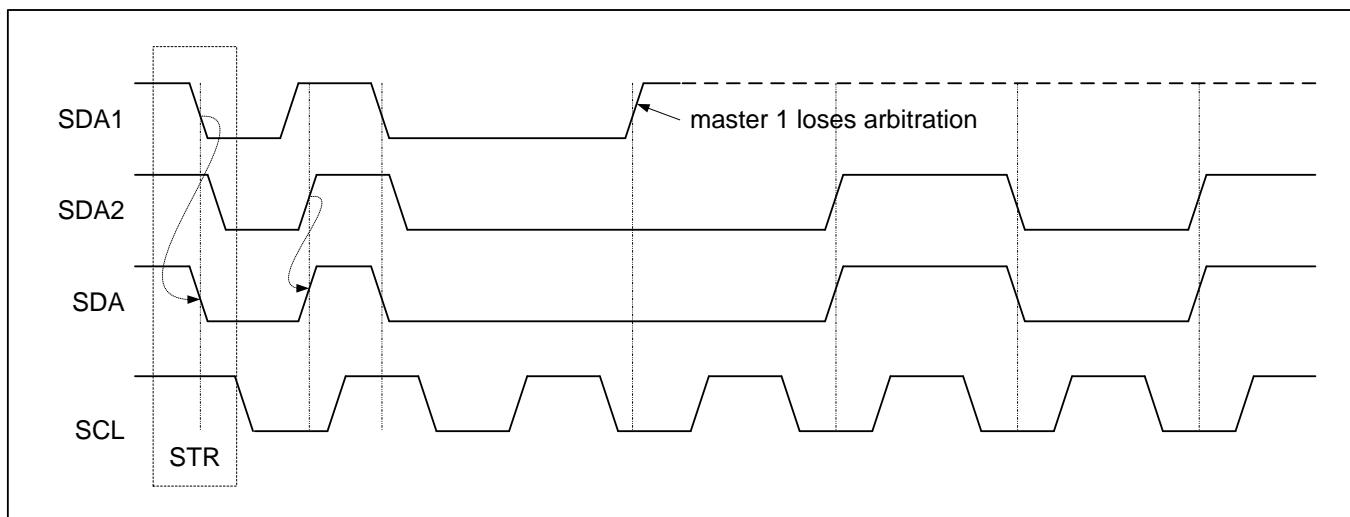


The frame format of I2C

20.4.1. Master mode

20.4.1.1. Arbitration lost

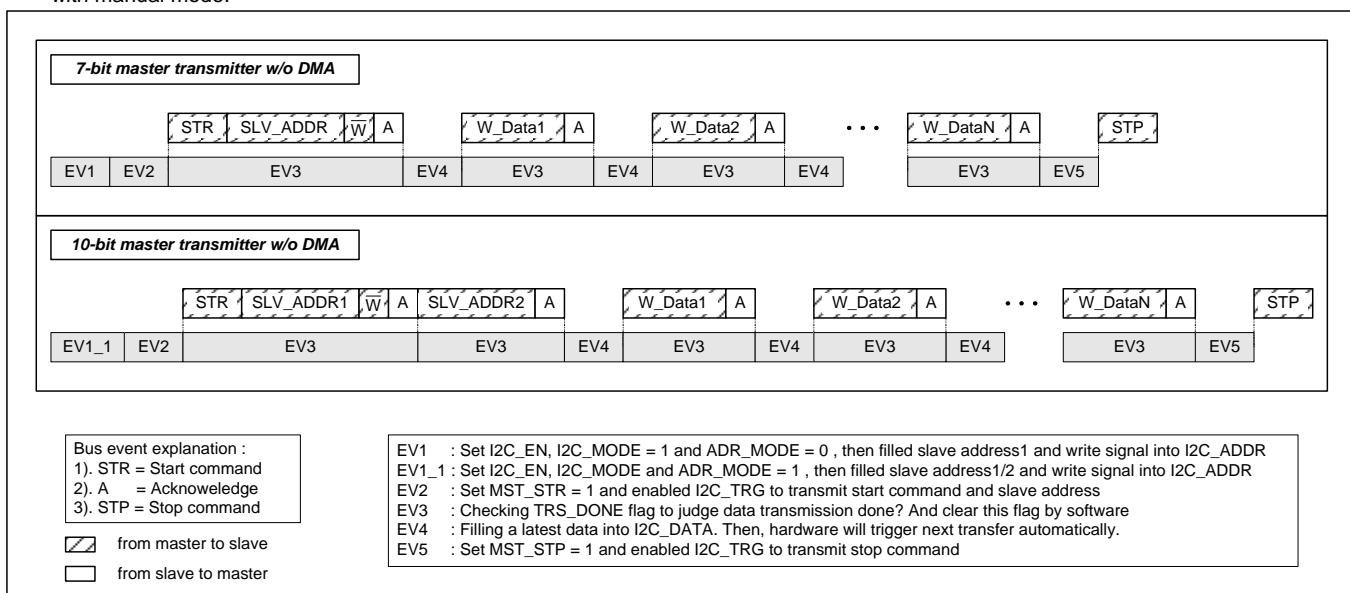
A master may start a transfer only if the bus is free. Arbitration takes place on the SDA line, while the SCL line is at the HIGH level. In such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output stage because the level on the bus doesn't correspond to its own level. The **I2C_STS.ARB_LOST_FLAG** is set by hardware when the I2C interface detects an arbitration lost. Then, I2C controller switches from master to slave mode automatically. A master that still generate clock pulses until the end of the packet when it loses the arbitration. Figure shows an arbitration mechanism of two masters.



Arbitration mechanism of two masters.

20.4.1.2. 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** to 1. In addition, master can inform slave this transfer is a write transfer by set **I2C_ADDR.RW_SEL** to 0. Then, I2C will send out data immediately when set **I2C_CTRL.I2C_STR**. During an I2C transmission, data shifts out most significant bit first on the SDA pin. The **I2C_STS.TRS_DONE_FLAG** flag will be set to 1 when acknowledge bit was finished. Before next transmission, user must clear this bit by software. Next, fill in latest data into **I2C_DATA** that will trigger next transfer. If user wants to terminate I2C transmission, set **I2C_CTRL.MST_STP** and **I2C_CTRL.I2C_STR** to 1 that will induce a stop command transfer. Figure shows a master transmit with manual mode.

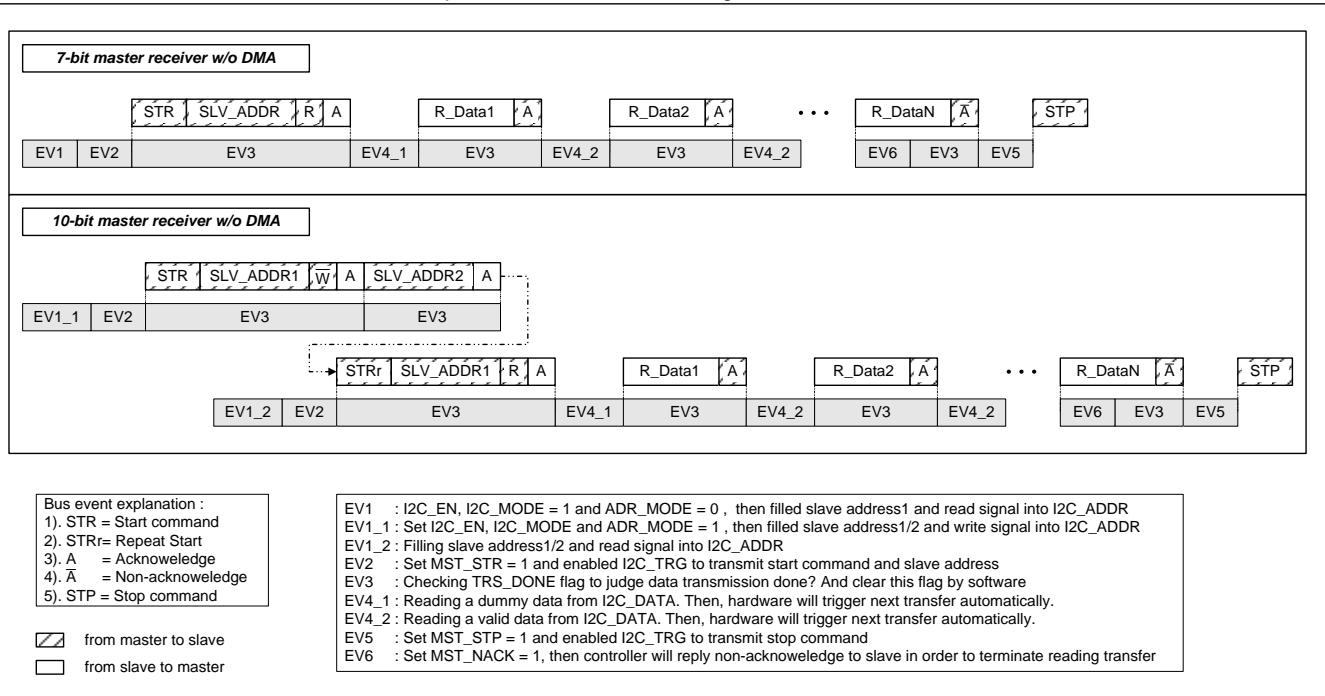


I2C master transmit with manual mode

20.4.1.3. 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** to 1. In addition, master can inform slave this transfer is a read transfer by set **I2C_ADDR.RW_SEL** to 1. Then, I2C will send out clock when set **I2C_CTRL.I2C_STR**. During an I2C transmission, data shifts in most significant bit first from the SDA pin. The **I2C_STS.TRS_DONE_FLAG** will be set to 1 when acknowledge bit was finished. Before next transmission, user must clear this bit

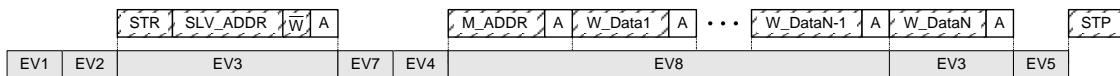
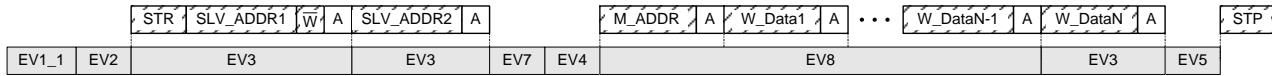
by software. Next, read data from **I2C_DATA** that will trigger next transfer. If user wants to terminate I2C transmission, set **I2C_CTRL.MST_NACK** to 1 during final read transfer. After final transfer done, set **I2C_CTRL.MST_STP** and **I2C_CTRL.I2C_STR** to 1 that will induce a stop command transfer. Below figure shows a master receive with manual mode.



I2C master receive with manual mode

20.4.1.4. 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** to 1. In addition, master can inform slave this transfer is a write transfer by set **I2C_ADDR.RW_SEL** to 0. Then, I2C will send out data immediately when set **I2C_CTRL.I2C_STR**. After finished slave-address transmission, set **I2C_CTRL.TX_DMAE** to 1 and filling a data into **I2Cx_DATA**, then hardware will trigger next transfer automatically. Next, checks DMA transfer done flag to estimate whether the transmission is finished or not. If DMA transfer done flag is set to 1, clearing this flag by software and waits final transfer done by checking **I2C_STSTRS_DONE_FLAG** flag. If user wants to terminate I2C transmission, set **I2C_CTRL.MST_STP** and **I2C_CTRL.I2C_STR** to 1 that will induce a stop command transfer. Figure shows a master transmit with DMA mode.

7-bit master transmitter w/i DMA

10-bit master transmitter w/i DMA

Bus event explanation :

- 1). STR = Start command
- 2). A = Acknowledge
- 3). STP = Stop command

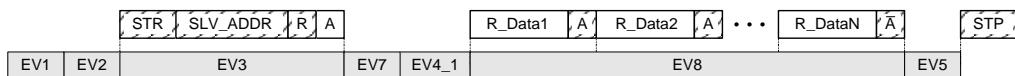
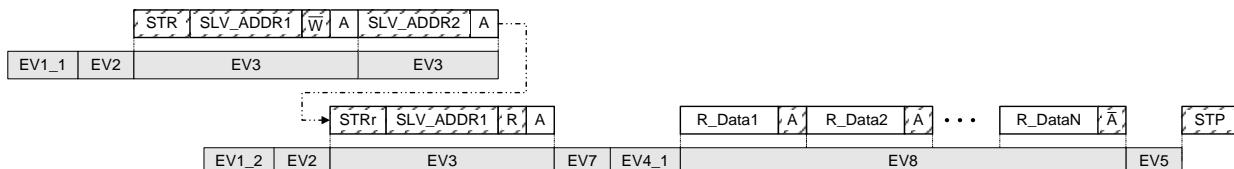
 from master to slave

 from slave to master

- EV1 : Set I2C_EN, I2C_MODE = 1 and ADR_MODE = 0 , then filled slave address1 and write signal into I2C_ADDR
 EV1_1 : Set I2C_EN, I2C_MODE and ADR_MODE = 1 , then filled slave address1/2 and write signal into I2C_ADDR
 EV2 : Set MST_STR = 1 and enabled I2C_TRG to transmit start command and slave address
 EV3 : Checking TRS_DONE flag to judge data transmission done? And clear this flag by software
 EV4 : Filling a latest data into I2C_DATA. Then, hardware will trigger next transfer automatically.
 EV5 : Set MST_STP = 1 and enabled I2C_TRG to transmit stop command
 EV7 : Set I2C_TX_DMA_EN = 1 to enable I2C DMA function
 EV8 : Checking DMA transfer done flag , and cleared flag by software

I2C master transmitter with DMA
20.4.1.5. 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** to 1. In addition, master can inform slave this transfer is a read transfer by set **I2C_ADDR.RW_SEL** to 1. Then, I2C will send out clock when set **I2C_CTRL.I2C_STR**. After finished slave-address transmission, set **I2C_CTRL.RX_DMAE** to 1 and reading a dummy data from **I2C_DATA**, then hardware will trigger next transfer automatically. Next, checks DMA transfer done flag to estimate whether the transmission is finished or not. If DMA transfer done flag is set to 1, clearing this flag by software. After DMA transfer done, set **I2C_CTRL.MST_STP** and **I2C_CTRL.I2C_STR** to 1 that will induce a stop command transfer. Figure shows a master receive with DMA mode.

7-bit master receiver w/i DMA

10-bit master receiver w/i DMA

Bus event explanation :

- 1). STR = Start command
- 2). STRr= Repeat Start
- 3). A = Acknowledge
- 4). \bar{A} = Non-acknowledge
- 5). STP = Stop command

 from master to slave

 from slave to master

- EV1 : I2C_EN, I2C_MODE = 1 and ADR_MODE = 0 , then filled slave address1 and write signal into I2C_ADDR
 EV1_1 : Set I2C_EN, I2C_MODE and ADR_MODE = 1 , then filled slave address1/2 and write signal into I2C_ADDR
 EV1_2 : Filling slave address1/2 and read signal into I2C_ADDR
 EV2 : Set MST_STR = 1 and enabled I2C_TRG to transmit start command and slave address
 EV3 : Checking TRS_DONE flag to judge data transmission done? And clear this flag by software
 EV4_1 : Reading a dummy data from I2C_DATA. Then, hardware will trigger next transfer automatically.
 EV4_2 : Reading a valid data from I2C_DATA. Then, hardware will trigger next transfer automatically.
 EV5 : Set MST_STP = 1 and enabled I2C_TRG to transmit stop command
 EV6 : Set MST_NACK = 1, then controller will reply non-acknowledge to slave in order to terminate reading transfer
 EV7 : Set I2C_RX_DMA_EN = 1 to enable I2C DMA function
 EV8 : Checking DMA transfer done flag , and cleared flag by software

I2C master receiver with DMA

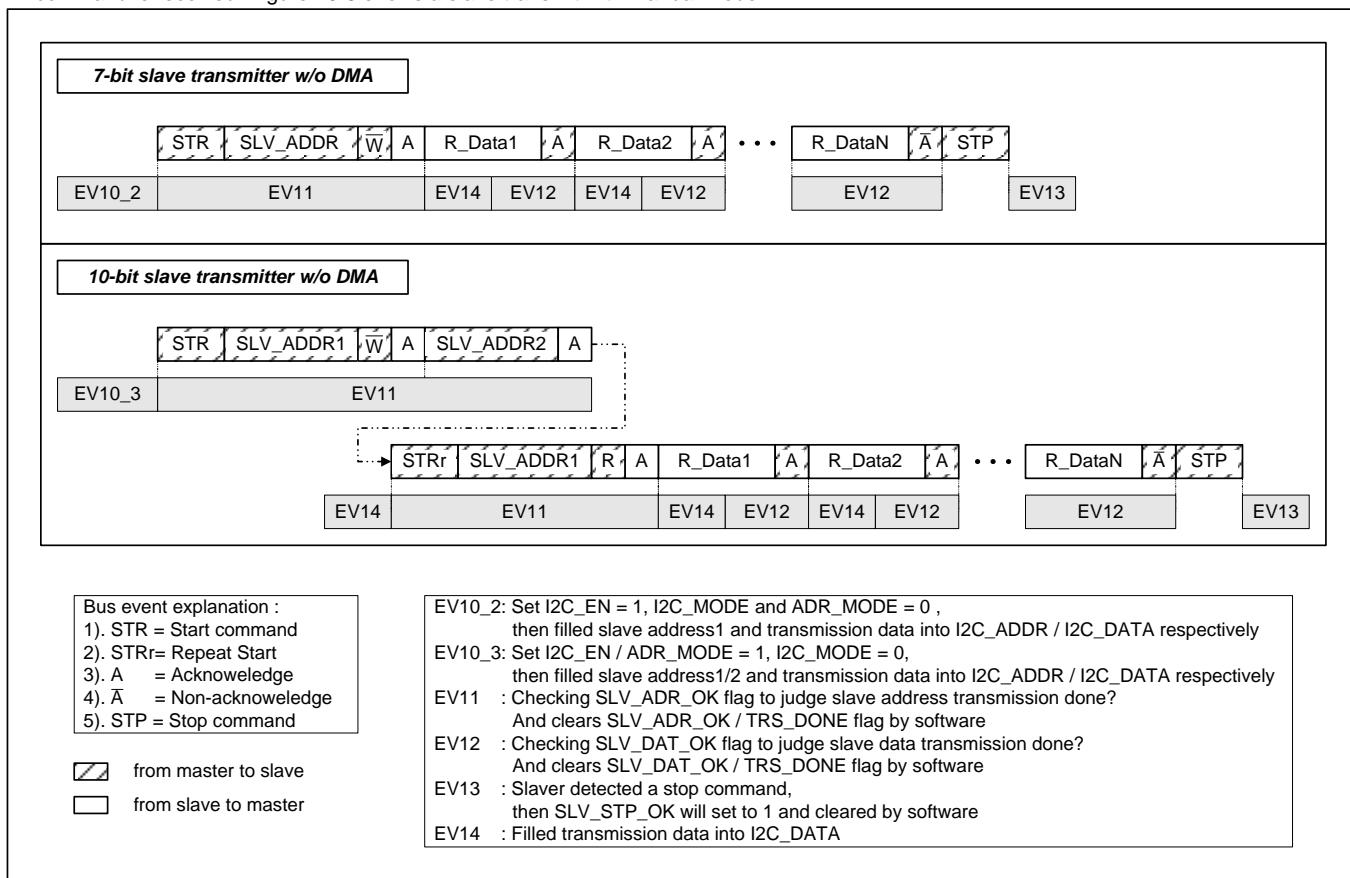
20.4.2. Slave mode

20.4.2.1. General call

In I2C bus, a 'general call' address which 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_FLAG** will be set when slave received a general call. CPU can receive an interrupt request if **I2C_CTRL.I2C_INT_EN** is set to 1.

20.4.2.2. I2C slave transmit with manual mode

In slave mode, before receiving data from master, user must fill in slave-address and set **I2C_CTRL.I2C_EN** to 1. Slave controller will set **I2C_STS.SLV_DATA_DONE_FLAG** flag to 1 if the received data of slave address was matched **I2C_ADDR**. Otherwise **I2C_STS.I2C_ERR_FLAG** will be set to 1. After slave-address checking, slave controller starts to receive input data from the master. During an I2C transmission, the **I2C_STS.TRS_DONE_FLAG** and **I2C_STS.DATA_DONE_FLAG** flag will be set to 1. User must clear these flags by software. Next, read the receiving data from **I2C_DATA**. The transfer will execute until a stop command is received. Figure 10-8 shows a slave transmit with manual mode.

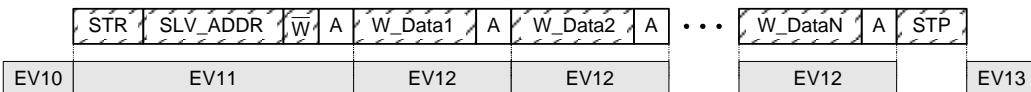
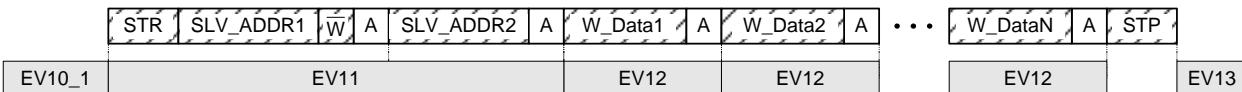


I2C slave transmit with manual mode

20.4.2.3. I2C slave receive 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** to 1. Slave controller will set **I2C_STS.SLV_DATA_DONE_FLAG** flag to 1 if the received data of slave address is matched **I2C_ADDR**. Otherwise **I2C_STS.SLV_ADR_ERR_FLAG** flag will be set to 1. After slave-address checking, filled-in data into **I2C_DATA** and waits **I2C_STS.DATA_DONE_FLAG** flag to be set to 1. The transfer will execute until a stop command is received.

Figure 10-9 shows a slave receive with manual mode.

7-bit slave receiver w/o DMA

10-bit slave receiver w/o DMA

Bus event explanation :

- 1). STR = Start command
- 2). A = Acknowledge
- 3). STP = Stop command

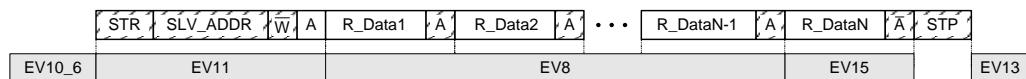
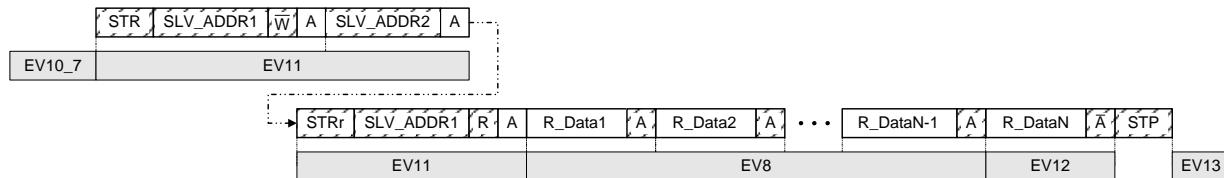
- from master to slave
 from slave to master

- EV10 : Set I2C_EN = 1, I2C_MODE and ADR_MODE = 0 , then filled slave address1 into I2C_ADDR
 EV10_1: Set I2C_EN / ADR_MODE = 1, I2C_MODE = 0,
 then filled slave address1/2 into I2C_ADDR
 EV11 : Checking SLV_ADR_OK flag to judge slave address transmission done?
 And clears SLV_ADR_OK / TRS_DONE flag by software
 EV12 : Checking SLV_DAT_OK flag to judge slave data transmission done?
 And clears SLV_DAT_OK / TRS_DONE flag by software
 EV13 : Slaver detected a stop command,
 then SLV_STP_OK will set to 1 and cleared by software

I2C slave receive with manual mode

20.4.2.4. I2C slave transmit with DMA

In slave mode, before using I2C transmits function, user must fill in slave-address and set **I2C_CTRL.TX_DMAE**, **I2C_CTRL.I2C_EN** to 1. After **I2C_STS.SLV_DATA_DONE_FLAG** flag was set to 1, checks DMA transfer done flag to estimate whether the transmission is finished or not. If DMA finished data transfer, then slave waits final data transfer done by checking **I2C_STS.TRS_DONE_FLAG**. Next, a stop command must be received in order to terminate transfer. Figure shows a slave transmit with DMA mode.

7-bit slave transmitter w/i DMA

10-bit slave transmitter w/i DMA

Bus event explanation :

- 1). STR = Start command
- 2). STRr= Repeat Start
- 3). A = Acknowledge
- 4). \bar{A} = Non-acknowledge
- 5). STP = Stop command

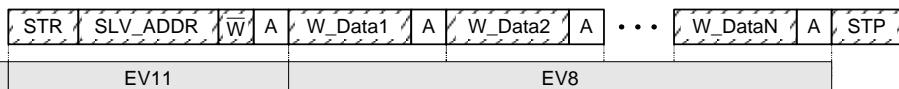
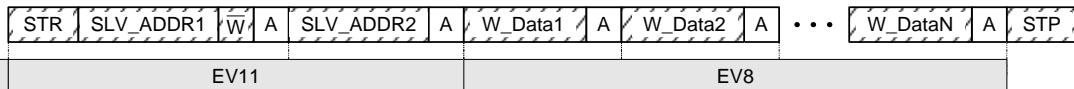
- from master to slave
 from slave to master

EV10_6: Set I2C_EN and TX_DMAE = 1, I2C_MODE and ADR_MODE = 0 , then filled slave address1 into I2C_ADDR
 EV10_7: Set I2C_EN , TX_DMAE , ADR_MODE = 1 and I2C_MODE = 0, then filled slave address1/2 into I2C_ADDR
 EV11 : Checking SLV_ADR_OK flag to judge slave address transmission done?
 And clears SLV_ADR_OK / TRS_DONE flag by software
 EV8 : Checking DMA transfer done flag , and cleared flag by software
 EV15 : Checking TRS_DONE flag
 EV13 : Slaver detected a stop command, then SLV_STP_OK will set to 1 and cleared by software

I2C slave transmit with DMA

20.4.2.5. I2C slave receive with DMA

In slave mode, before using I2C receives function, user must fill in slave-address and set **I2C_CTRL.RX_DMAE**, **I2C_CTRL.I2C_EN** to 1. After **I2C_STS.SLV_DATA_DONE_FLAG** flag was set to 1, checks DMA transfer done flag to estimate whether the transmission is finished or not. If DMA finished data transfer, then slave waits final data transfer done by checking **I2C_STS.TRS_DONE_FLAG**. These bits can be cleared by software. Next, a stop command must be received in order to terminate transfer. Figure shows a slave receive with DMA mode.

7-bit slave receiver w/i DMA

10-bit slave receiver w/i DMA


Bus event explanation :

- 1). STR = Start command
- 2). A = Acknowledge
- 3). STP = Stop command

from master to slave
 from slave to master

EV10_4: Set I2C_EN and RX_DMAE = 1, I2C_MODE and ADR_MODE = 0, then filled slave address1 into I2C_ADDR

EV10_5: Set I2C_EN / ADR_MODE / RX_DMAE = 1, I2C_MODE = 0, then filled slave address1/2 into I2C_ADDR

EV8 : Checking DMA transfer done flag , and cleared flag by software

EV11 : Checking SLV_ADR_OK flag to judge slave address transmission done?
And clears SLV_ADR_OK / TRS_DONE flag by software

EV13 : Slaver detected a stop command,
then SLV_STP_OK will set to 1 and cleared by software

I2C slave receive with DMA

20.5. Register Description

Register map

Base Address : 0x400B_0000						
Name	Description			Address	Access	Reset value
I2C0_CTRL	I2C Control Register			0x400B_0000	R/W	0x0000 0000
I2C0_STS	I2C Status Register			0x400B_0004	R/W	0x0000 0002
I2C0_ADDR	I2C Transmit Data Register			0x400B_0008	R/W	0x0000 0000
I2C0_DATA	I2C Receive Data Register			0x400B_000C	R/W	0x0000 0000

Register Function

I2C_CTRL I2C Control Register Address : 0x400B_0000							
31	30	29	28	27	26	25	24
		--			RX_DMAE	TX_DMAE	MODE_SEL
23	22	21	20	19	18	17	16
ERR_SADR_IN	I2C_INT_EN						
T_EN							
15	14	13	12	11	10	9	8
MST_STR	MST_STP	MST_NACK	--			I2C_CLK_SEL	I2C_EN
7	6	5	4	3	2	1	0
			--				I2C_STR

Bit	Name	Description	Access	Reset value
[31:27]	--	Reserved	R	0x00
[26]	RX_DMA_EN	DMA enable for I2C Rx mode 0 = disabled	R/W	0x0

Bit	Name	Description	Access	Reset value																
		1 = enabled																		
[25]	TX_DMA_EN	DMA enable for I2C Tx mode 0 = disabled 1 = enabled	R/W	0x0																
[24]	MODE_SEL	I2C controller operating mode select bits 0 = slaver mode 1 = master mode	R/W	0x0																
[23]	ERR_SADR_INT_EN	Slaver address error interrupt enable bit 0 = disabled 1 = enabled Note : This bit is available when I2C controller operating in slaver mode.	R/W	0x0																
[22]	I2C_INT_EN	I2C interrupt enable bit 0 = disabled 1 = enabled	R/W	0x0																
[21:16]	I2C_DB_TIME	SCL / SDA input de-bounce time select bits	R	0x0																
[15]	MST_STR	I2C controller issued start command enable bit 0 = disabled 1 = enabled Note : This bit will be cleared automatically when this transfer is finished.	R/W	0x0																
[14]	MST_STP	I2C controller issued stop command enable bit 0 = disabled 1 = enabled Note : This bit will be cleared automatically when this transfer is finished.	R/W	0x0																
[13]	MST_NACK	I2C controller issued non-acknowledge enable bit 0 = disabled 1 = enabled Note : This bit will be cleared automatically when this transfer is finished.	R/W	0x0																
[12]	--	Reserved	R	0x0																
[11:9]	CLK_SEL	I2C controller serial clock select bits <table border="1" data-bbox="516 1426 1191 1729"> <tr> <td>CLK_SEL[1:0]</td><td>Clock Source</td></tr> <tr> <td>000</td><td>I2C clock is system clock / 16</td></tr> <tr> <td>001</td><td>I2C clock is system clock / 32</td></tr> <tr> <td>010</td><td>I2C clock is system clock / 64</td></tr> <tr> <td>011</td><td>I2C clock is system clock / 128</td></tr> <tr> <td>100</td><td>I2C clock is system clock / 256</td></tr> <tr> <td>101</td><td>I2C clock is system clock / 768</td></tr> <tr> <td>110</td><td>I2C clock is system clock / 1024</td></tr> </table>	CLK_SEL[1:0]	Clock Source	000	I2C clock is system clock / 16	001	I2C clock is system clock / 32	010	I2C clock is system clock / 64	011	I2C clock is system clock / 128	100	I2C clock is system clock / 256	101	I2C clock is system clock / 768	110	I2C clock is system clock / 1024	R/W	0x0
CLK_SEL[1:0]	Clock Source																			
000	I2C clock is system clock / 16																			
001	I2C clock is system clock / 32																			
010	I2C clock is system clock / 64																			
011	I2C clock is system clock / 128																			
100	I2C clock is system clock / 256																			
101	I2C clock is system clock / 768																			
110	I2C clock is system clock / 1024																			
[8]	I2C_EN	I2C controller enable bit 0 = disabled 1 = enabled	R/W	0x0																
[7:1]	--	Reserved	R	0x00																
[0]	I2C_STR	The start transmission trigger bit This bit is for master mode only. The I2C master will begin to transmit or receive data when I2C_EN is set to 1. This bit will be cleared by H/W automatically.	R/W	0x0																

Bit	Name	Description	Access	Reset value
		0 = disabled 1 = enabled Note: This bit is available when I2C_CTRL.I2C_EN is enabled.		

I2C_STS I2C Status Register								Address : 0x400B_0004
31	30	29	28	27	26	25	24	--
								--
23	22	21	20	19	18	17	16	--
								--
15	14	13	12	11	10	9	8	--
				SLV_ADR_ERR_FLAG	BUSY_FLAG	GEN_CALL_FL AG		--
7	6	5	4	3	2	1	0	--
--	ARB_LOST_FL AG	--	SLV_DATA_DO NE_FLAG	DATA_DONE_F LAG	STOP_CMD_F LAG	NO_ACK_FLAG	TRS_DONE_F LAG	--

Bit	Name	Description	Access	Reset value
[31:12]	--	Reserved	R	0x00 0000
[11]	SLV_ADR_ERR_FLAG	Slaved address error flag read: 0 = slaver address is correct address 1 = slaver address is wrong address write: 0 = no effect 1 = clear this bit Note: This bit is only used in slave mode.	R/W	0x0
[10]	BUSY_FLAG	I2C controller BUSY_FLAG flag read: 0 = No communication on the bus 1 = Communication ongoing on the bus write: 0 = no effect 1 = clear this bit	R/W	0x0
[9]	GEN_CALL_FLAG	I2C general call flag read: 0 = I2C master has issued general call 1 = I2C master has not issued general call write: 0 = no effect 1 = clear this bit	R/W	0x0
[8:7]	-	Reserved	R	0x0
[6]	ARB_LOST_FLAG	I2C bus arbitration lost flag read: 0 = I2C bus arbitration lost is not occurred 1 = I2C bus arbitration lost is occurred	R/W	0x0

Bit	Name	Description	Access	Reset value
		<p>write:</p> <p>0 = no effect 1 = clear this bit</p> <p>Note: This error occurs when the I2C interface detects an arbitration lost condition.</p>		
[5]	-	Reserved	R	0x0
[4]	SLV_DATA_DONE_FLAG	<p>Slaved address had been received</p> <p>read:</p> <p>0 = slaver address is not asserted 1 = slaver address is asserted and match to the setting</p> <p>write:</p> <p>0 = no effect 1 = clear this bit</p> <p>Note: This bit is only used in slave mode.</p>	R	0x0
[3]	DATA_DONE_FLAG	<p>Data transmit or receive done flag</p> <p>read:</p> <p>0 = data is transmitting or idle now 1 = data is transmission complete</p> <p>write:</p> <p>0 = no effect 1 = clear this bit</p> <p>Note: This bit is only used in slave mode.</p>	R/W	0x0
[2]	STOP_CMD_FLAG	<p>Stop command received done flag</p> <p>read:</p> <p>0 = stop command is not be received 1 = stop command had received</p> <p>write:</p> <p>0 = no effect 1 = clear this bit</p> <p>Note: This bit is only used in slave mode.</p>	R/W	0x0
[1]	NO_ACK_FLAG	<p>I2C have not received acknowledging signal.</p> <p>read :</p> <p>0 = acknowledge 1 = no acknowledge</p> <p>write :</p> <p>0 = clear this bit 1 = on effect</p>	R/W	0x1
[0]	TRS_DONE_FLAG	<p>I2C controller transmission complete flag</p> <p>read:</p> <p>0 = I2C is idle or on going 1 = I2C finished data transmission</p> <p>write:</p> <p>0 = no effect 1 = clear this bit</p>	R/W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

I2C_ADDR								I2C Transmit Data Register	Address : 0x400B_0008	
31	30	29	28	27	26	25	24	--		
23	22	21	20	19	18	17	16	--	ADDR_MODE	
15	14	13	12	11	10	9	8	ADDR2		
7	6	5	4	3	2	1	0	ADDR1	RW_SEL	

Bit	Name	Description	Access	Reset value
[31:17]	--	Reserved	R	0x0000
[16]	ADDR_MODE	Address mode control bit 0 = 7-bit mode 1 = 10-bit mode Note: This bit is available in slaver mode only.	R/W	0x0
[15:8]	ADDR2	Address 2 nd Byte 7-bit addressing mode : don't care 10-bit addressing mode : slave address[7:0]	R/W	0x0
[7:3]	ADDR1[6:2]	Address 1 st 7Bits 7-bit addressing mode : slave address[6:2] 10-bit addressing mode : 0x1E	R/W	0x00
[2:1]	ADDR1[1:0]	Address 1 st 7Bits 7-bit addressing mode : slave address[1:0] 10-bit addressing mode : slave address[9:8]	R/W	0x0
[0]	RW_SEL	I2C read / write control signal 0 = write 1 = read Note: This bit is shared between master and slaver mode.	R/W	0x0

I2C_DATA								I2C Receive Data Register	Address : 0x400B_000C	
31	30	29	28	27	26	25	24	--		
23	22	21	20	19	18	17	16	--		
15	14	13	12	11	10	9	8	--		
7	6	5	4	3	2	1	0	I2C_DATA		

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x00 0000
[7:0]	I2C_DATA	I2C controller read / write data register Note: The bit is shared by the master and slave modes.	R/W	0x00

21. I2S

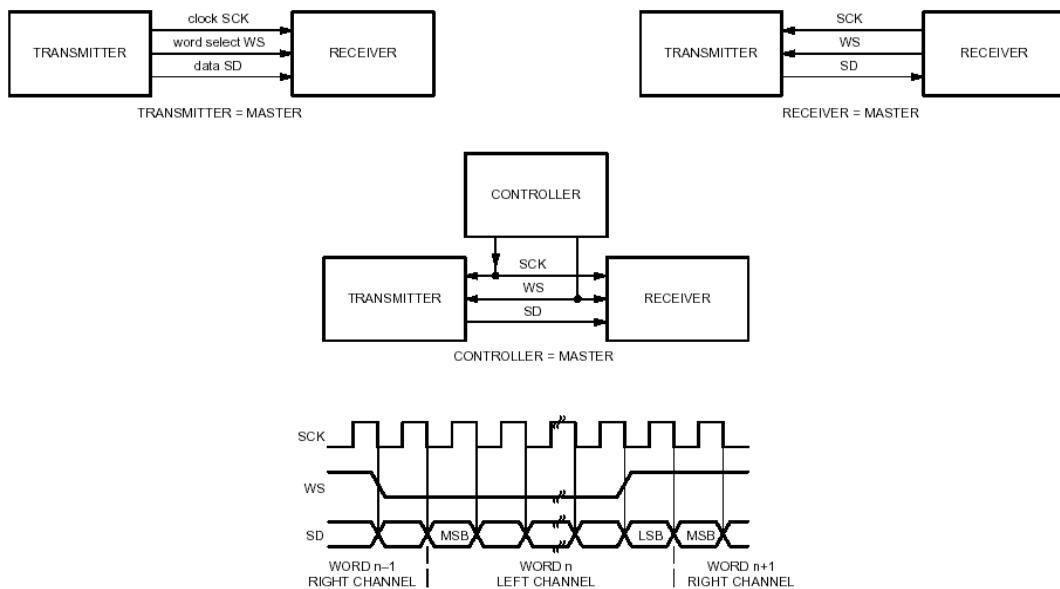
21.1. Introduction

The bus only needs to handle audio data while 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 them difficult to define the master. In such system, there is usually a system master controlling digital audio data-flow between each IC. Transmitters then, have to generate data under the control of an external clock, and so act as a slave. 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.

21.2. Features

- One RX channels supported (MCLK: IOA[18]/ BCLK: IOA[19]/ SLR: IOA[20]/ DATA: IOA[21])
- 2 Words RX FIFO
- Supports configurable settings of each 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.

21.3. Block Diagram



21.4. Function

21.4.1. 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 to '0') for data transmission. If the receiver receives more bits than its word length, the bits after the LSB are ignored. On the other hand, if the receiver receives fewer bits than its word length, the missing bits are given zero instead. And so, the MSB has a fixed position, where 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 when transmitting data that 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).

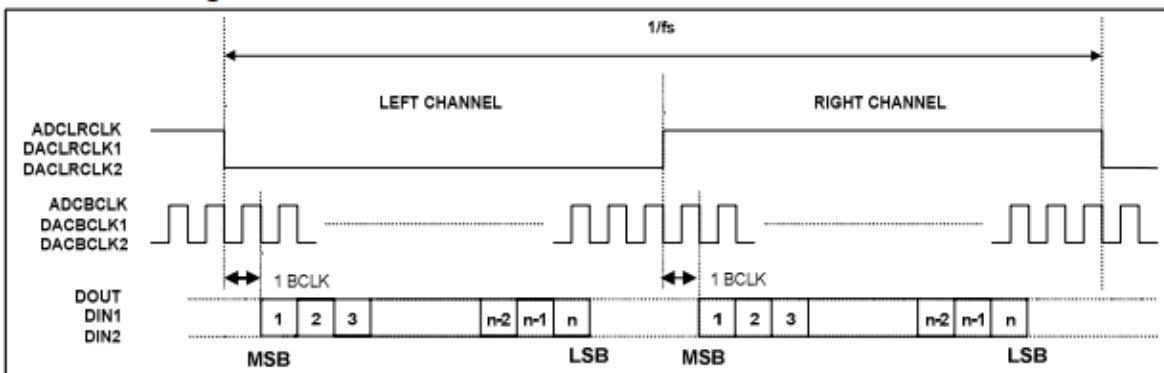
WS may change either on a trailing or leading edge of the serial clock, but it doesn't need to be symmetrical. In slave, this signal is latched on the leading edge of the clock signal. The WS line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clears the input for the next word.

21.4.2. Normal mode & I2S mode

I2S mode:

I2S mode is a subset of network mode and used in audio application, one frame only includes 2 slots, one slot is left audio channel, and another slot is right audio channel. Generally, the I2S' frame sync asserts one bit before the first bit of the frame as the following figure.

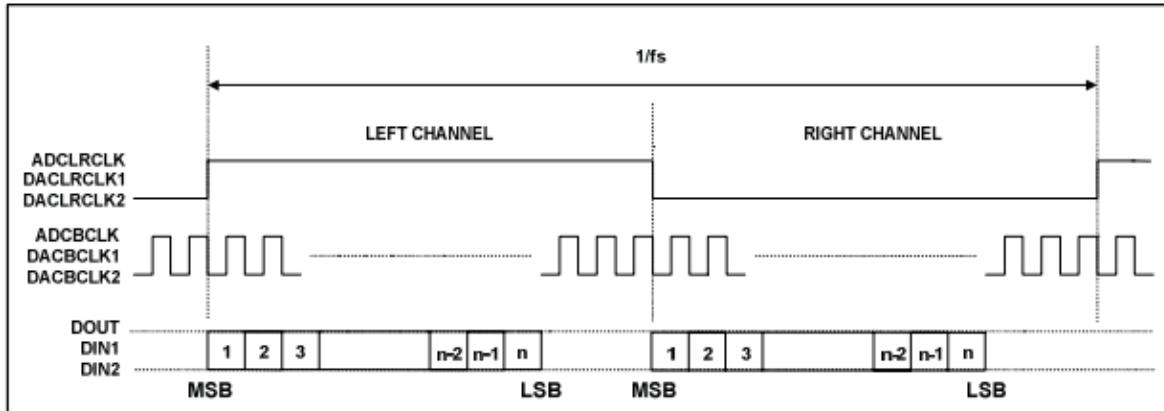
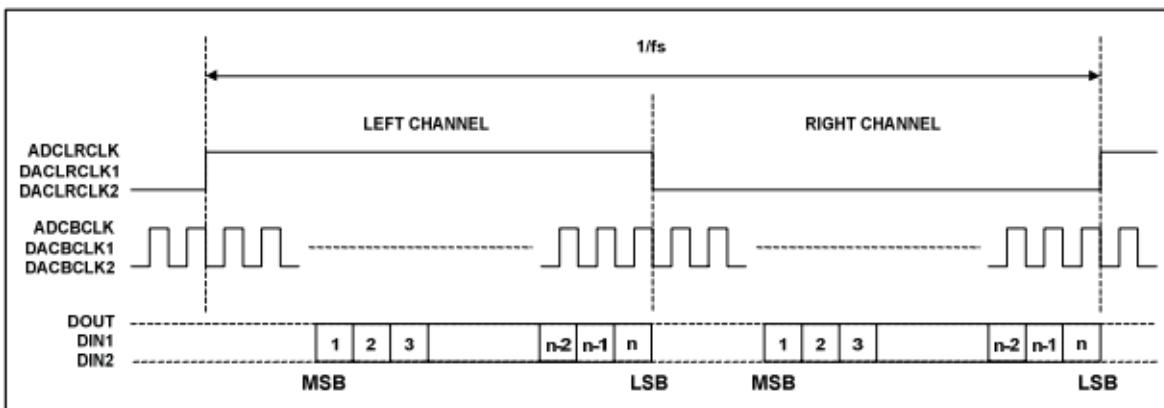
I2S Mode Timing



Normal mode:

The major difference between Normal mode and I2S mode is in the first bit of frame data, which will be aligned with Left/ Right channel rising/falling edge in normal mode, but I2S mode will be shifted 1-bit backward.

As the example below, the Normal mode has Left justified & Right justified modes.

Normal_LJ(Left Justified)

Normal_RJ(Right Justified)


21.4.3. I2S RX Interrupt

When data stored in RX FIFO is more than half of its size, "interrupt" will be activated. The function can be enabled or disabled by "EN_HALF_FULL_IRT" of I2S_CTRL.

21.4.4. I2S RX FIFO

RX FIFO is a 2-port FIFO and can read and write data at the same time. The I2S RX FIFO is 2 Words (2 * 32).

21.5. Register Description

Register map

Base Address : 0x400C_0000				
Name	Description	Address	Access	Reset value
I2S_CTRL	I2S RX Control Register	0x400C_0000	R/W	0x0000_4A20
I2S_DATA	I2S RX Data Register	0x400C_0004	R	0x0000 0000
I2S_STS	I2S RX Status Register	0x400C_0008	R/W	0x0000 0000
I2S_CTRL2	I2S RX Control Register2	0x400C_000C	R/W	0x4000 0003

Register Function

I2S_CTRL I2S RX Control Register Address : 0x400C_0000							
31	30	29	28	27	26	25	24
--							

23	22	21	20	19	18	17	16
--	--	MONO_MODE	DATA_ODR	MERGE_EN	--	--	HALF_INT_EN
15	14	13	12	11	10	9	8
--	INT_POLARITY	MODE_SEL	FRAMING_MODE_SEL	FRAME_SIZE_SEL	FRAME_SIZE_SEL	DATA LENG SEL[2]	
7	6	5	4	3	2	1	0
DATA LENG SEL[1:0]	ALIGN_SEL	RX_MODE	RX_LATCH_SEL	RX_FRAME_POL	FRAME_POL	FRAME_POL	RX_EN

Bit	Name	Description	Access	Reset value
[31:22]	--	Reserved	R	0x00
[21]	MONO_MODE_ENABLE (MONO_MODE)	Mono mode enable 0: Stereo mode 1: Mono mode 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.	R/W	0x0
[20]	RIGHT_LSB_ENABLE (DATA_ODR)	Data Order in 16-bit Merge mode 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. Note: Whether Right channel data is in LSB of RX_DATA, it is valid only when word length is 16-bit and the MERGE function is on.	R/W	0x0
[19]	MERGE_ENABLE (MERGE_EN)	16-bit Merge mode 0: Disable 1: Enable Note: For 24-bit or 32-bit frame length, this bit must be "0". Note: In Merge mode, it will merge two16-bit data into one 32-bit data.	R/W	0x0
[18:17]	--	Reserved	R	0x00
[16]	HALF_FULL_INT_ENAB LE (HALF_INT_EN)	I2S FIFO Half full Interrupt Enable 0: Disabled 1: Enabled	R/W	0x0
[15]	--	Reserved	R	0x00
[14]	INT_SEL (INT_POLARITY)	Interrupt Polarity select 1: Rising-edge triggered 0: Falling- edge triggered	R/W	0x1
[13]	MODE_SEL	Master/ Slave mode select When 0, the I2S RX controller acts as a slave, received LRCK, and BLCK clock from transmitter. 0: Slave, receive LRCK_IN (fixed) 1: Master	R/W	0x0
[12:11]	FRAMING_MODE_SEL	Framing Mode 00: I2S mode 01: Normal Mode 1x: Reserved	R/W	0x1
[10:9]	FRAME_SIZE_SEL	Frame size If frame size of codec cannot be adjusted equal to frame size, set it to I2S controller; we suggest using 11(frame size is not predictable), I2S controller will determine it automatically. 00: Each frame (right/left) is 16-bit length 01: Each frame (right/left) is 24-bit length	R/W	0x1

Bit	Name	Description	Access	Reset value
		10: Each frame (right/left) is 32-bit length 11: Frame size is unpredictable, only used in slave mode		
[8:6]	DATA_LENGTH_SEL (DATA LENG_SEL)	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, where bit[19:0] is audio data and bit[31:20] is not audio data. Note that the data word length must not be greater than frame size. 000: Data word length is 16-bit length 001: Data word length is 18-bit length 010: Data word length is 20-bit length 011: Data word length is 22-bit length 100: Data word length is 24-bit length 101: Data word length is 32-bit length 110: Data word length is 32-bit length 111: Data word length is 32-bit length	R/W	0x0
[5]	ALIGN_SEL	Received Data Right/Left Alignment. When frame size is greater than valid data length, left alignment informs I2S controller that the valid data bit will be received first and next, the received dummy bits. Right alignment means dummy bits will be received first and then, valid data bits will be received. 0: Right alignment 1: Left alignment (Default)	R/W	0x1
[4]	RX_DATA_ENDIANNES S_SEL (RX_MODE)	Bit Priority Receiving Mode 0: MSB bit receiving first (Default) 1: LSB bit receiving first	R/W	0x0
[3]	RX_DATA_LATCH_SEL (RX_LATCH_SEL)	Receiving Data Latch Mode Select 0: Receiving data bit on serial clock falling edge (Default) 1: Receiving data bit on serial clock rising edge	R/W	0x0
[2]	RX_FRAME_POL	Receiving Frame Polarity Selection User should assure both RX and Transmitter are set to the same frame polarity definition. 0: LRCK=0 is the right frame (Default) 1: LRCK=0 is the left frame	R/W	0x0
[1]	FRAME_POL	First Frame Polarity User should assure both RX and Transmitter set to the same as the first frame LR polarity definition. 0: Left frame (Default) 1: Right frame	R/W	0x0
[0]	RX_EN	Enable I2S RX RX_EN is high active enable signal for receive data. It should be asserted to receiver after setting all the required configurations. 0: Disabled 1: Enabled	R/W	0x0

I2S DATA								I2S RX Data Register								Address : 0x400C_0004							
31	30	29	28	27	26	25	24	RX_DAT[31:24]															
23	22	21	20	19	18	17	16	RX_DAT[23:16]															
15	14	13	12	11	10	9	8	RX_DAT[15:8]															
7	6	5	4	3	2	1	0	RX_DAT[7:0]															

Bit	Name	Description																Access	Reset value
[31:0]	RX_DATA	RX_DATA port Reading data from this port by CPU or DMA will get data from RX FIFO.																R	0x0

I2S_STS								I2S RX Status								Address : 0x400C_0008							
31	30	29	28	27	26	25	24	--															
23	22	21	20	19	18	17	16	--								CLR_FIFO HALF_FULL_FLAG							
15	14	13	12	11	10	9	8	OVERFLOW_FLAG								--							
7	6	5	4	3	2	1	0	--								FIFO_NUM							

Bit	Name	Description																Access	Reset value
[31:19]	--	Reserved																R	0x00
[18]	CLEAR_FIFO_ENABLE (CLR_FIFO)	Clear RX FIFO Clears RX FIFO bit automatically and clears it to 0 after the FIFO pointer is cleared. Write 1 to clear RX FIFO and polling the bit to become 0 to make sure clean RX FIFO action done.																R/W	0x0
[17]	HALF_FULL_FLAG	I2S FIFO Half Full INT Status Flag 1 = INT occurred (Write 1 to Clear) 0 = No Interrupt																R/W	0x0
[16]	--	Reserved																R	0x00
[15]	OVERFLOW_FLAG	Overflow Flag After RX FIFO become full, if there still receiving data to RX FIFO, then OVERFLOW_FLAG will become 1. Note that when OVERFLOW_FLAG occurs, it means the receiving audio samples are lost and broken sounds occur. Software should avoid overflow occurrence. 0: No overflow occurs. 1: Overflow occurs. Write 1 to clear the bit (clears the flag).																R/W	0x0
[14:4]	--	Reserved																R	0x00
[3:2]	FIFO_INT_LEVEL (FIFO_NUM)	RX FIFO Number The number of word stored in the RX FIFO																R	0x0

Bit	Name	Description	Access	Reset value
[1]	FIFO_FULL_FLAG (FULL_FLAG)	RX FIFO Full Flag When FIFO becomes full, WORD_NO is 32; this flag is 1. 0: RX FIFO is not full 1: RX FIFO is full	R	0x0
[0]	FIFO_EMPTY_FLAG (EMPTY_FLAG)	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	R	0x00

I2S_CTRL2								I2S RX Control Register2								Address : 0x400C_000C	
31	30	29	28	27	26	25	24										
BCLK_DIV_SEL								--									
23	22	21	20	19	18	17	16										
								--									
15	14	13	12	11	10	9	8										
								--									
7	6	5	4	3	2	1	0										
--		--		MCLK_DIV													

Bit	Name	Description	Access	Reset value
[31:29]	BCLK_DIV_SEL	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	R/W	0x2
[28:6]	--	Reserved	R	0x00
[5:0]	MCLK_DIV_SEL (MCLK_DIV)	I2S master clock : 0: MCLK= Pclk / 2 1~63: MCLK = Pclk/ (MCLK_DIV+1)	R/W	0x3

22. PWM I/O

22.1. Introduction

There are 8 PWM I/Os presented in GPCM1F SERIALS series. The purpose of PWM is useful in LCD and motor-relevant applications. PWM output can be acquired via setting up clock source and duty. 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 to output; the buffer control bit of I/O is able to control output's polarity.

22.2. Features

- Supports 8 PWM I/Os (IOA[12:5])
- Supports different clock sources (Timer0/1/2, CCP Timer0/1 & CPU CLK DIV 8 ~ 4096)
- Separates PWM duty control (8-bit Resolution, its valid bit will be effected by the given value of Period)
- Separates PWM Period control (32/64/128/256 period)

22.3. Function

PWM Output Diagram with I/O Configuration

Before PWM outputs, set the output I/O to output mode; the buffer control bit of the I/O will determine the output polarity. See the following diagram for reference.

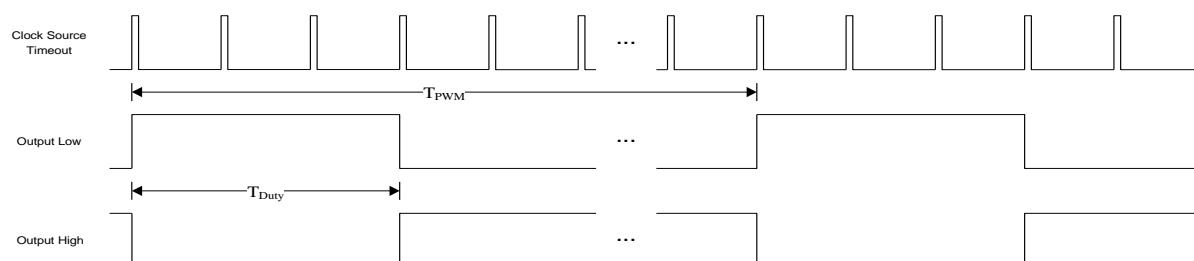
Set register CLK_SEL[3:0] for different clock source timeout

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 reference PWMIO_CTRL1 Bit[23:16] Inverse setting. If the IOx_INV = 0, PWMIOx will keep in 'Lo'. If the IOx_INV = 1, PWMIOx will keep in 'Hi'



Note: $T_{PWM} = \text{Clock Source Timeout} * \text{period}$

22.4. Register Description

Register Map

Base Address : 0x4016_0000				
Name	Description	Address	Access	Reset value
PWMIO_CTRL0	PWM I/O Control Register 0	0x4016_0000	R/W	0x0000_0000
PWMIO_CTRL1	PWM I/O Control Register 1	0x4016_0004	R/W	0x0000_0000
PWMIO_TOGGLE_CTRL	PWMIO IO Toggle Control	0x4016_0008	R/W	0x0000_0000
PWMIO_PERIOD_CTRL	PWM IO Period Control	0x4016_000C	R/W	0x0000_0000
PWMIO_PWMIO0_Duty	PWM I/O0 Duty Control	0x4016_0010	R/W	0x0000_0000

Base Address : 0x4016_0000				
Name	Description	Address	Access	Reset value
PWMIO_PWMIO1_Duty	PWM I/O1 Duty Control	0x4016_0014	R/W	0x0000_0000
PWMIO_PWMIO2_Duty	PWM I/O2 Duty Control	0x4016_0018	R/W	0x0000_0000
PWMIO_PWMIO3_Duty	PWM I/O3 Duty Control	0x4016_001C	R/W	0x0000_0000
PWMIO_PWMIO4_Duty	PWM I/O4 Duty Control	0x4016_0020	R/W	0x0000_0000
PWMIO_PWMIO5_Duty	PWM I/O5 Duty Control	0x4016_0024	R/W	0x0000_0000
PWMIO_PWMIO6_Duty	PWM I/O6 Duty Control	0x4016_0028	R/W	0x0000_0000
PWMIO_PWMIO7_Duty	PWM I/O7 Duty Control	0x4016_002C	R/W	0x0000_0000

Register Function

PWMIO_CTRL0								PWM I/O Control Register 0		Address : 0x4016_0000	
31	30	29	28	27	26	25	24	--	--	--	--
23	22	21	20	19	18	17	16	--	--	--	--
15	14	13	12	11	10	9	8	--	--	--	SYNC_EN
7	6	5	4	3	2	1	0	--	--	--	--
CLK_SEL				--							

Bit	Name	Description	Access	Reset value
[31:9]	--	Reserved	R	0x0
[8]	SYNC_EN	Disables/Enables whether PWM I/O has synchronization function. When it is enabled and duty is changed, PWM IO will change its duty after 256 cycles 0 = Disabled 1 = Enabled	R/W	0x0
[7:4]	CLK_SEL	Set the clock source of PWM I/O 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 = F _{CPU} /8 1001 = F _{CPU} /16 1010 = F _{CPU} /32 1011 = F _{CPU} /64 1100 = F _{CPU} /256 1101 = F _{CPU} /1024	R/W	0x0

Bit	Name	Description	Access	Reset value
		1110 = $F_{CPU}/4096$ 1111 = not define yet		
[3:0]	--	Reserved	R	0x0

PWMIO_CTRL1 PWM I/O Control Register 1								Address : 0x4016_0004
31	30	29	28	27	26	25	24	--
23	22	21	20	19	18	17	16	
PWM7_INV	PWM6_INV	PWM5_INV	PWM4_INV	PWM3_INV	PWM2_INV	PWM1_INV	PWM0_INV	
15	14	13	12	11	10	9	8	
PWM7_MSK	PWM6_MSK	PWM5_MSK	PWM4_MSK	PWM3_MSK	PWM2_MSK	PWM1_MSK	PWM0_MSK	
7	6	5	4	3	2	1	0	
PWM7_EN	PWM6_EN	PWM5_EN	PWM4_EN	PWM3_EN	PWM2_EN	PWM1_EN	PWM0_EN	

Bit	Name	Description	Access	Reset value
[31:24]	--	Reserved	R	0x0
23	PWM7_INV	Inverse PWM I/O7	R/W	0x0
22	PWM6_INV	Inverse PWM I/O6	R/W	0x0
21	PWM5_INV	Inverse PWM I/O5	R/W	0x0
20	PWM4_INV	Inverse PWM I/O4	R/W	0x0
19	PWM3_INV	Inverse PWM I/O3	R/W	0x0
18	PWM2_INV	Inverse PWM I/O2	R/W	0x0
17	PWM1_INV	Inverse PWM I/O1	R/W	0x0
16	PWM0_INV	Inverse PWM I/O0	R/W	0x0
15	PWM7_MSK	MASK PWM I/O7 (keep PWM7_INV value)	R/W	0x0
14	PWM6_MSK	MASK PWM I/O6 (keep PWM6_INV value)	R/W	0x0
13	PWM5_MSK	MASK PWM I/O5 (keep PWM5_INV value)	R/W	0x0
12	PWM4_MSK	MASK PWM I/O4 (keep PWM4_INV value)	R/W	0x0
11	PWM3_MSK	MASK PWM I/O3 (keep PWM3_INV value)	R/W	0x0
10	PWM2_MSK	MASK PWM I/O2 (keep PWM2_INV value)	R/W	0x0
9	PWM1_MSK	MASK PWM I/O1 (keep PWM1_INV value)	R/W	0x0
8	PWM0_MSK	MASK PWM I/O0 (keep PWM0_INV value)	R/W	0x0
7	PWM7_EN	Enables/Disables PWM I/O7(IOA12)	R/W	0x0
6	PWM6_EN	Enables/Disables PWM I/O6(IOA11)	R/W	0x0
5	PWM5_EN	Enables/Disables PWM I/O5(IOA10)	R/W	0x0
4	PWM4_EN	Enables/Disables PWM I/O4(IOA9)	R/W	0x0
3	PWM3_EN	Enables/Disables PWM I/O3(IOA8)	R/W	0x0
2	PWM2_EN	Enables/Disables PWM I/O2(IOA7)	R/W	0x0
1	PWM1_EN	Enables/Disables PWM I/O1(IOA6)	R/W	0x0
0	PWM0_EN	Enables/Disables PWM I/O0(IOA5)	R/W	0x0

Note: When 0x0004[7:0] is enabled, PWM IO counter will be reset and re-count again. In addition, when PWM IO is required to be output mode and when it is in input mode, the PWM will not output, but counter will keep counting.



GPCM1F SERIES PROGRAMMING GUIDE

P_PWMIO_TOGGLE_CTRL PWM I/O Toggle Control								Address : 0x4016_0008
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
TOGGLE7_EN	TOGGLE6_EN	TOGGLE5_EN	TOGGLE4_EN	TOGGLE3_EN	TOGGLE2_EN	TOGGLE1_EN	TOGGLE0_EN	

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x0
7	TOGGLE7_EN	Disables/Enables PWM IO7 with 50% of duty cycle output	R/W	0x0
6	TOGGLE6_EN	Disables/Enables PWM IO6 with 50% of duty cycle output	R/W	0x0
5	TOGGLE5_EN	Disables/Enables PWM IO5 with 50% of duty cycle output	R/W	0x0
4	TOGGLE4_EN	Disables/Enables PWM IO4 with 50% of duty cycle output	R/W	0x0
3	TOGGLE3_EN	Disables/Enables PWM IO3 with 50% of duty cycle output	R/W	0x0
2	TOGGLE2_EN	Disables/Enables PWM IO2 with 50% of duty cycle output	R/W	0x0
1	TOGGLE1_EN	Disables/Enables PWM IO1 with 50% of duty cycle output	R/W	0x0
0	TOGGLE0_EN	Disables/Enables PWM IO0 with 50% of duty cycle output	R/W	0x0

PWMIO_PERIOD_CTRL PWM I/O Period Control Register								Address : 0x4016_000C
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
PERIOD7		PERIOD6		PERIOD5		PERIOD4		
7	6	5	4	3	2	1	0	
PERIOD3		PERIOD2		PERIOD1		PERIOD0		

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:14]	PERIOD7	Setting PWM I/O7 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0
[13:12]	PERIOD6	Setting PWM I/O6 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0
[11:10]	PERIOD5	Setting PWM I/O5 period 00 = 256	R/W	0x0

Bit	Name	Description	Access	Reset value
		01 = 128 10 = 64 11 = 32		
[9:8]	PERIOD4	Setting PWM I/O4 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0
[7:6]	PERIOD3	Setting PWM I/O3 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0
[5:4]	PERIOD2	Setting PWM I/O2 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0
[3:2]	PERIOD1	Setting PWM I/O1 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0
[1:0]	PERIOD0	Setting PWM I/O0 period 00 = 256 01 = 128 10 = 64 11 = 32	R/W	0x0

PWMIO_PWMIO0_Duty								PWM I/O0 Duty Control		Address : 0x4016_0010	
31	30	29	28	27	26	25	24	--	--	--	--
23	22	21	20	19	18	17	16	--	--	--	--
15	14	13	12	11	10	9	8	--	--	--	--
7	6	5	4	3	2	1	0	PWM0_Duty			

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x0
[7:0]	PWM0_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty	R/W	0x00



GPCM1F SERIES PROGRAMMING GUIDE

PWMIO_PWMIO1_Duty								PWM I/O1 Duty Control								Address : 0x4016_0014							
31	30	29	28	27	26	25	24	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
23	22	21	20	19	18	17	16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	14	13	12	11	10	9	8	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
7	6	5	4	3	2	1	0	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty	PWM1_Duty

Bit	Name	Description								Access	Reset value
[31:8]	--	Reserved								R	0x0
[7:0]	PWM1_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty								R/W	0x00

PWMIO_PWMIO2_Duty								PWM I/O2 Duty Control								Address : 0x4016_0018							
31	30	29	28	27	26	25	24	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
23	22	21	20	19	18	17	16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	14	13	12	11	10	9	8	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
7	6	5	4	3	2	1	0	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty	PWM2_Duty

Bit	Name	Description								Access	Reset value
[31:8]	--	Reserved								R	0x0
[7:0]	PWM2_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty								R/W	0x00

PWMIO_PWMIO3_Duty								PWM I/O3 Duty Control								Address : 0x4016_001C							
31	30	29	28	27	26	25	24	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
23	22	21	20	19	18	17	16	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
15	14	13	12	11	10	9	8	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
7	6	5	4	3	2	1	0	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty	PWM3_Duty

Bit	Name	Description								Access	Reset value
[31:8]	--	Reserved								R	0x0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description	Access	Reset value
[7:0]	PWM3_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty	R/W	0x00

PWMIO_PWMIO4_Duty PWM I/O4 Duty Control Address : 0x4016_0020							
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
PWM4_Duty							

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x0
[7:0]	PWM4_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty	R/W	0x00

PWMIO_PWMIO5_Duty PWM I/O Duty Control Address : 0x4016_0024							
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
PWM5_Duty							

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x0
[7:0]	PWM5_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty	R/W	0x00

PWMIO_PWMIO6_Duty PWM I/O6 Duty Control Address : 0x4016_0028							
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8



GPCM1F SERIES PROGRAMMING GUIDE

7	6	5	4	3	2	1	0
PWM6_Duty							

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x0
[7:0]	PWM6_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty	R/W	0x00

PWMIO_PWMIO7_Duty								PWM I/O7 Duty Control		Address : 0x4016_002C	
31	30	29	28	27	26	25	24				
--											
23	22	21	20	19	18	17	16				
--											
15	14	13	12	11	10	9	8				
7	6	5	4	3	2	1	0	PWM7_Duty			

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0x0
[7:0]	PWM7_Duty	Duty Control for PWM I/O0 Write = Duty control Read = Represent duty	R/W	0x00

23. Quadrature Decoder

23.1. Introduction

The purpose of quadrature decoder (QD) is to detect the phase differences between two square waves, normally used for rotating devices to detect the rotation velocity and position, etc. There are two sets of quadrature decoders in GPCM1F SERIALS series, each comprising of two data inputs, control bits, counter and a clear port to reset the counter. User only needs to input two square waves into quadrature decoder through two input pins and starts running function. If one of the waves changes, quadrature decoder will determine forward or backward based on the phase differences and change the value in the counter. QD counter has 16-bit resolution. The following example depicts the operation of QD. Suppose the forward threshold is set to 15 and backward threshold is set to -16. If forward, counter is added by 1, up to +15 (0FH) at maximum threshold. If the maximum is reached, forward will keep counter at +15. On the other hand, if backward, counter is subtracted by 1, down to -16 (F0H) at minimum threshold. If minimum is reached, backward will keep counter at -16. In other words, counter's range is between -16 and +15. More information about the determination on forward and backward, please refer to the following section.

When counter's value changes, quadrature decoder will determine to issue an interrupt according to user's settings. The interrupt modes includes two types: all interrupt mode and flow interrupt mode. When all interrupt mode is set, whenever quadrature decoder detects any forward/backward signal, interrupt is issued. That is, whenever counter changes, interrupt is issued. If flow interrupts mode is set, interrupt is issued in the following two conditions:

- In forward and counter is greater than +12 (0DH ~ 0FH).
- In backward and counter is less than -13 (F2H ~ F0H).

Also, there is a clear port in each quadrature decoder to reset counter. Writing any data into the clear port will reset the counter to 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.

23.2. Features

- Supports two sets of quadrature decoders
- Supports 16-bit forward/backward counter
- Supports input de-bounce filter

23.3. Function

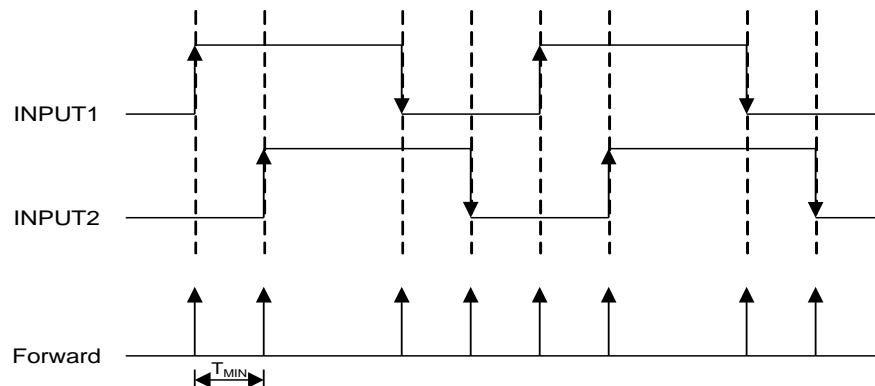
Quadrature Decoder Diagram

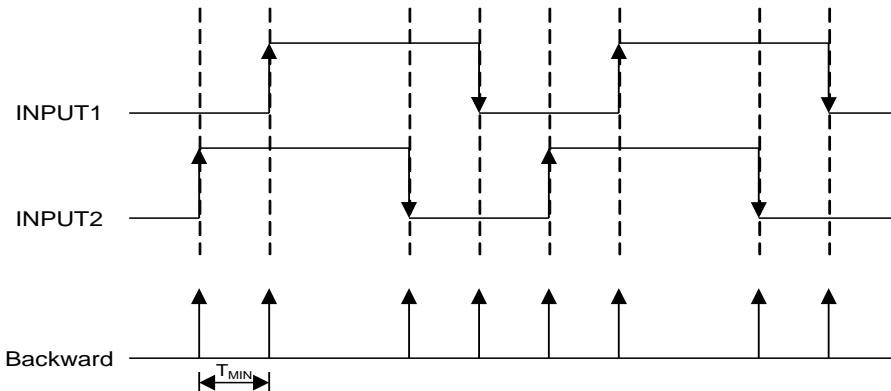
The following diagrams describe the forward and backward conditions quadrature decoder holds. Each quadrature decoder has two inputs. We summarize as follows:

Quadrature decoder 1 – INPUT1: IOA.18, INPUT2: IOA.19

Quadrature decoder 2 – INPUT1: IOA.20, INPUT2: IOA.21

Note that when quadrature decoder starts, its corresponding INPUT I/Os must be configured as input mode.





Note: T_{MIN} is the minimum interval between two events; it should be greater than $1 / (F_{CPU}/2)$.

23.4. Register Description

Register Map

Base Address : 0x4016_0040					
Name	Description		Address	Access	Reset value
QD_QD0_CNT	Counter of Quadrature Decoder 0		0x4016_0040	R	0x0000_0000
QD_QD1_CNT	Counter of Quadrature Decoder 1		0x4016_0044	R	0x0000_0000
QD_QD0_CLR	Clear Port of QD0 Counter		0x4016_0048	W	0x0000_0000
QD_QD1_CLR	Clear Port of QD1 Counter		0x4016_004C	W	0x0000_0000
QD_CTRL	Control Register of Quadrature Decoders		0x4016_0050	R/W	0x0000_0000
QD_STS	Status Register of Quadrature Decoders		0x4016_0054	R/W	0x0000_0000
QD_FW_TH	Forward counter Threshold		0x4016_0058	R/W	0x0000_000F
QD_BW_TH	Backward counter Threshold		0x4016_005C	R/W	0x0000_FFF0

Register Function

QD_QD0_CNT Counter of Quadrature Decoder 0 Address : 0x4016_0040							
31	30	29	28	27	26	25	24
--	--	--	--	--	--	--	--
23	22	21	20	19	18	17	16
--	--	--	--	--	--	--	--
15	14	13	12	11	10	9	8
QD0_CNT[15:8]							
7	6	5	4	3	2	1	0
QD0_CNT[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:0]	QD0_CNT	Counter of quadrature decoder 0	R	0x0000

Note: Write is ineffective.

QD_QD1_CNT Counter of Quadrature Decoder 1 Address : 0x4016_0044							
31	30	29	28	27	26	25	24
--	--	--	--	--	--	--	--
23	22	21	20	19	18	17	16
--	--	--	--	--	--	--	--



GPCM1F SERIES PROGRAMMING GUIDE

15	14	13	12	11	10	9	8
QD1_CNT[15:8]							
7	6	5	4	3	2	1	0
QD1_CNT[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:0]	QD1_CNT	Counter of quadrature decoder 1	R	0x0000

Note: Write is ineffective.

QD_QD0_CLR Clear Port of QD0 Counter								Address : 0x4016_0048
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
QD0_CLR[15:8]								
7	6	5	4	3	2	1	0	
QD0_CLR[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:0]	QD0_CLR	Write any data to clear QD0 counter Write = Clear QD0 counter	W	0x0000

QD_QD1_CLR Clear Port of QD1 Counter								Address : 0x4016_004C
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
QD1_CLR[15:8]								
7	6	5	4	3	2	1	0	
QD1_CLR[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:0]	QD1_CLR	Write any data to clear QD1 counter Write = Clear QD1 counter	W	0x0000

QD_CTRL Quadrature Decoder Control Register								Address : 0x4016_0050
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								

15	14	13	12	11	10	9	8
--	--	--	--	QD1_REV_EN	QD1_POS_EN	QD0_REV_EN	QD0_POS_EN
7	6	5	4	3	2	1	0
OVER_MODE_SEL	DEBOUNCE_SEL			QD1_EN	QD1_INT_MODE DE	QD0_EN	QD0_INT

Bit	Name	Description	Access	Reset value
[31:12]	--	Reserved	R	0x0
[11]	QD1_REV_EN	QD1 motor reserve interrupt enable 1: Enabled 0: Disabled	R/W	0x0
[10]	QD1_POS_EN	QD1 motor positive interrupt enable 1: Enabled 0: Disabled	R/W	0x0
[9]	QD0_REV_EN	QD0 motor reserve interrupt enable 1: Enabled 0: Disabled	R/W	0x0
[8]	QD0_POS_EN	QD0 motor positive interrupt enable 1: Enabled 0: Disabled	R/W	0x0
[7]	OVERFLOW_MODE2_EN ABLE (OVER_MODE_SEL)	QD Overflow mode select: 1: Original overflow mode 0: Default mode	R/W	0x0
[6:4]	DEBOUNCE_SEL	Quadrature input debounce time select 0 = no debounce 1 = 4T 2 = 8T 3 = 16T 4 = 32T 5 = 40T 6 = 80T 7 = 128T Note: 'T' Time unit is the system clock.	R/W	0x0
[3]	QD1_EN	Quadrature decoder 1 control bit 1: Enable 0: Disable	R/W	0x0
[2]	QD1_INT_MODE	Quadrature decoder 1 interrupt mode 0 = All interrupts mode 1 = Overflow interrupts mode	R/W	0x0
[1]	QD0_EN	Quadrature decoder 0 control bit 1: Enable 0: Disable	R/W	0x0
[0]	QD0_INT_MODE	Quadrature decoder 0 interrupt mode 0 = All interrupts mode 1 = overflow interrupts mode	R/W	0x0

QD_STS Quadrature Decoder Status Register								Address : 0x4016_0054	
31	30	29	28	27	26	25	24		
			--						
23	22	21	20	19	18	17	16		
			--						
15	14	13	12	11	10	9	8		
	--		QD1_REV_INT_ FLAG	QD1_POS_IN T_FLAG	QD0_REV_INT _FLAG	QD0_POS_IN T_FLAG			
7	6	5	4	3	2	1	0		
			--						

Bit	Name	Description	Access	Reset value
[31:12]	--	Reserved	R	0x0
[11]	QD1_REV_INT_FLAG	QD1 motor reverse interrupt flag read: 0 = QD1motor is not reversed 1 = QD1 motor reverse write: 0 = no effect 1 = clear this bit	R/W	0x0
[10]	QD1_POS_INT_FLAG	QD1 motor positive interrupt flag read: 0 = QD1motor is not positive 1 = QD1 motor positive write: 0 = no effect 1 = clear this bit	R/W	0x0
[9]	QD0_REV_INT_FLAG	QD0 motor reverse interrupt flag read: 0 = QD0 motor is not reversed 1 = QD0 motor reverse write: 0 = no effect 1 = clear this bit	R/W	0x0
[8]	QD0_POS_INT_FLAG	QD0 motor positive interrupt flag read: 0 = QD0 motor is not positive 1 = QD0 motor positive write: 0 = no effect 1 = clear this bit	R/W	0x0
[7:0]	--	Reserved	R	0x0

Quadrature Decoder Forward Threshold Register								Address : 0x4016_0058
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
QD_FW_TH[15:8]								
7	6	5	4	3	2	1	0	
QD_FW_TH[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:0]	QD_FW_TH	<p>Quadrature counter Forward(Positive) threshold Note: It's divided into default mode & original mode, controlled by QD_CTRL Bit.7 OVER_MODE.</p> <p>Default Mode: When counter[15:0] == QD_FW_TH [15:0], interrupt will assert. And the forward counter add to QD_FW_TH[15:0] and stop counting</p> <p>Default : 0x000F the same as original design</p> <p>Original mode: When counter[15:2] == QD_FW_TH [15:2], interrupt will assert. And the forward counter add to QD_FW_TH[15:0] and stop counting Ex: default qd counter = 0x0c ~0x0F</p>	R/W	0x000F

Quadrature Decoder Backward Threshold Register								Address : 0x4016_005C
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
QD_BW_Thr[15:8]								
7	6	5	4	3	2	1	0	
QD_BW_Thr[7:0]								

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15:0]	QD_BW_TH	<p>Quadrature counter Backward(Reserve) threshold Note: It's divided into default mode & original mode, controlled by QD_CTRL Bit.7 OVER_MODE.</p> <p>Default Mode: Default : 0xFFFF the same as original design</p> <p>When counter[15:0] == QD_BW_TH [15:0], interrupt will assert. And the backward counter sub to QD_BW_TH[15:0] and stop counting</p>	R/W	0xFFFF

Bit	Name	Description	Access	Reset value
		<p>Original mode:</p> <p>When counter[15:2] == QD_BW_TH [15:2], interrupt will assert. And the backward counter sub to QD_BW_TH[15:0] and stop counting</p> <p>Ex: default qd counter = 0xFFFF ~0xFFFF3</p> <p>[15]: sign bit</p>		

24. Capacitive Touch Sensor (CTS)

24.1. Introduction

GPCM1F SERIALS contains Capacitive Touch Sensor (CTS) module for related application. User can increase its precision by adjusting charge/discharge current of pull up/down.

24.2. Feature

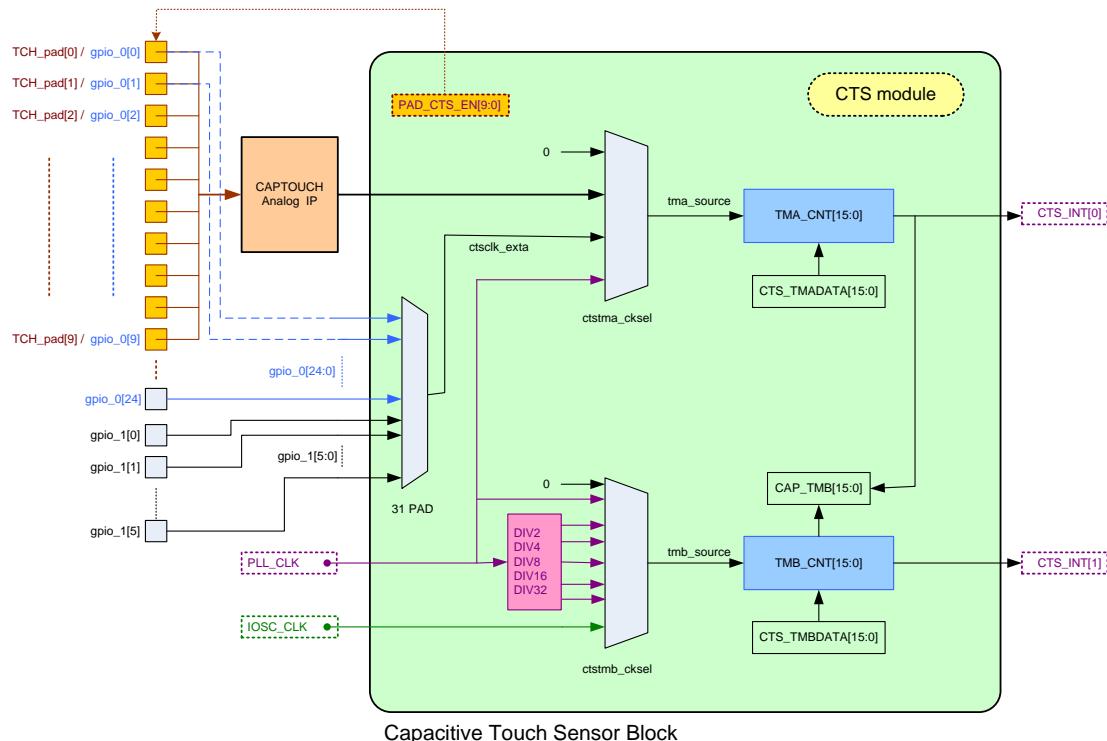
- Self-capacitance Touch pads, supports IOA[9:0], total of 10 IO pins for Touch pads
- Supports Normal mode or Deep Sleep mode wakeup function
- Two sets of 16-bit timers with preload function, can be dedicated for CTS or general timer.
 - (a) TMA CLK Source can be the CTS Module; or can be input through IO pin.
 - (b) TMB CLK Source can be selected via losc_CLK(16MHz) or via frequency division from Sys. For the frequency of CLK, if using CTS wakeup function in Low Power Mode, CLK Src must select losc16M.
 - (c) Note: In Low Power Mode, it requires extra 32KHz for Multi-IO sequence scanning CLK Src. The frequency (8Hz ~ 128Hz) can be determined by CTS_CTRL[5:4](0x400E0000) for Multi-IO sequence scanning.
- Low power mode for reducing current consumption in deep sleep mode.
- Multi-IO Mode: In Low Power mode, CTS_CTRL (0x400E_0000)Bit[7] can determine whether to use sequence scan or fixed scan on a specific pad, up to 10 Touch pads can be scanned.
- Auto vs. Manual Stop Mode: Determines if TMA Counter reaches user's setup value for number of Touch charge/discharge time, whether to stop counting.
 - (a) Auto Stop Mode will stop CTS_TMA & TMB counter. For re-count, user needs to write "1" to CTS_CTRL (0x400E_0000)Bit[1].
 - (b) Manual Stop Mode will not automatically stop & re-load TMB Counter Data.

24.3. Notes

If an adjacent pad, located after the Touch PAD, is used for key-change wakeup source, please set IO mode to Input pull-low. It cannot be set as Input pull-high; otherwise, system will be mistakenly awakened. Reasons are as follows:

- (a) When CTS enters sleep mode, it will scan the touch pads in sequence. When the last touch pad is scanned, due to hardware design, scan index will be incremented by 1 which will cause the adjacent I/O becoming low status for a short period (approx. 70ns). If the adjacent IO is set input pull-high, it will turn the IO status from high to low instantly. That will mistakenly trigger key-change wakeup INT to wake system up.
- (b) For example, suppose IOA[3:2] are used for touch pads. The next adjacent IOA.4 cannot be set as Input pull-high, but the IOA.1 does not affected by this limitation.
- (c) Suppose an IO, which is adjacent to the touch pad, is set output mode. Under sleep mode, it will turn to floating state temporarily (approx. 70ns). In most applications, it is not notable.

24.4. Block Diagram



24.5. Function

Auto stop mode:

Write START_ENABLE=1, then CTS_TMA/TMB will start counting. Interrupt asserted when CTS_TMA overflow happened. Hardware capture CTS_TMB data to CAP_TMB register and stop counting CTS_TMA/TMB counter. User need to Write START_ENABLE=1 again to re-start counting.

Manual stop mode:

Write START_ENABLE=1, then CTS_TMA/TMB will start counting. Interrupt asserted when CTS_TMA overflow happened. Hardware capture CTS_TMB data to CAP_TMB register and load CTS_TMBDATA. The Counter won't stop in this mode, so hardware will have the same reaction when CTS_TMA overflow next time. User need to write START_ENABLE=0 to stop counting. If CTS_TMA clock frequency doesn't change, CAP_TMB register data should be the same

24.6. Register Description

Register Map

Base Address : 0x400E_0000				
Name	Description	Address	Access	Reset value
CTS_CTRL	CTS control register	0x400E_0000	R/W	0x0000_0A00
CTS_STS	CTS interrupt status	0x400E_0004	R/W	0x0000_0000
CTS_TMAPLOAD	CTS preload register for timer A	0x400E_0008	R/W	0x0000_FFFF
CTS_TMACOUNT	CTS counter for timer A	0x400E_000C	R	0x0000_0000
CTS_TMB0PLOAD	CTS preload register for timer B0	0x400E_0010	R/W	0x0000_FFFF
CTS_TMBCOUNT	CTS counter for timer B	0x400E_0014	R	0x0000_0000
CTS_TMBCAP	Capture register for timer B	0x400E_0018	R/W	0x0000_0000
CTS_PADSEL	CTS TMA clock source select (From IO PAD)	0x400E_001C	R/W	0x0000_0000
CTS_TMB1PLOAD	CTS preload register for timer B1	0x400E_0020	R/W	0x0000_FFFF
CTS_TMB2PLOAD	CTS preload register for timer B2	0x400E_0024	R/W	0x0000_FFFF

CTS_TMB3PLOAD	CTS preload register for timer B3	0x400E_0028	R/W	0x0000_FFFF
CTS_TMB4PLOAD	CTS preload register for timer B4	0x400E_002C	R/W	0x0000_FFFF
CTS_TMB5PLOAD	CTS preload register for timer B5	0x400E_0030	R/W	0x0000_FFFF
CTS_TMB6PLOAD	CTS preload register for timer B6	0x400E_0034	R/W	0x0000_FFFF
CTS_TMB7PLOAD	CTS preload register for timer B7	0x400E_0038	R/W	0x0000_FFFF
CTS_TMB8PLOAD	CTS preload register for timer B8	0x400E_003C	R/W	0x0000_FFFF
CTS_TMB9PLOAD	CTS preload register for timer B9	0x400E_0040	R/W	0x0000_FFFF

Register Function

CTS_CTRL CTS Control Register0								Address : 0x400E_0000
31	30	29	28	27	26	25	24	
--		TMB_INTEN_SEL	TMA_INTEN_SEL	--	--	--	--	
23	22	21	20	19	18	17	16	
TMB_CLK_SEL			TMB_MODE	TMA_CLK_SEL			TMA_MODE	
15	14	13	12	11	10	9	8	
--			CHARGE_CUR RENT	MULTI_IO_INDEX				
7	6	5	4	3	2	1	0	
SLEEP_MULTII O_EN	SLEEP_EN	SLEEP_CLK_SEL		--	AUTOSTOP_EN	START_ENABL E	CTSEN_ENA BLE	

Bit	Name	Description	Access	Reset value
[31:30]	--	Reserved	R	0x0
[29]	TMB_INTEN_SEL	Timer B interrupt enable 1: Enabled 0: Disabled	R/W	0x0
[28]	TMA_INTEN_SEL	Timer A interrupt enable 1: Enabled 0: Disabled	R/W	0x0
[27:24]	--	Reserved	R	0x0
[23:21]	TMB_CLK_SEL	TMB clock source select: 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 (16Mhz for sleep mode)	R/W	0x0
[20]	TMB_MODE	CTS TMB dedicated for CTS timer or General timer 0: CTS TMB is dedicated for touch sensor timer 1: CTS TMB is general timer	R/W	0x0
[19:17]	TMA_CLK_SEL	CTS TMA Clock Source Select 000: CTS TMA clock OFF(default) 001: CTS TMA clock source is from IO pad 010: CTS TMA clock source is from touch sensor module	R/W	0x0

Bit	Name	Description	Access	Reset value
		011: CTS TMA clock source is SYS clock 100: CTS TMA clock source is SYS clock / 2 101: CTS TMA clock source is SYS clock / 4 110: CTS TMA clock source is SYS clock / 8 111: CTS TMA clock source is SYS clock / 16		
[16]	TMA_MODE	CTS TMA dedicated for CTS timer or General timer 0: CTS TMA is dedicated for touch sensor 1: CTS TMA is general timer	R/W	0x0
[15:13]	--	Reserved	R	0x0
[12]	CHARGE_CURRENT	Charge/Discharge current selection 0 : 25uA, 1 : 50uA	R/W	0x0
[11:8]	MULTI_IO_INDEX	It displays which touch pad is scanned currently. Index 0 = IOA0, Index1 = IOA1..., Index9 = IOA9 Note: SCAN from IOA0 to IOA9 and re-scan it.	R	0xA
[7]	SLEEP_MULTIIO_EN	Sleep mode multi IO function enable 1: Enabled 0: Disabled	R/W	0x0
[6]	SLEEP_EN	Low power mode enable for saving current consumption in sleep mode 1: Enabled 0: Disabled	R/W	0x0
[5:4]	SLEEP_CLK_SEL	Low power mode in sleep, trigger clock source select 00: 8 Hz 01: 16 Hz 10: 32 Hz 11: 128 Hz	R/W	0x0
[3]	--	Reserved	R	0x0
[2]	AUTOSTOP_ENABLE (AUTOSTOP_EN)	CTS Auto Stop control bit 0: manual stop mode, stop CTS_TMA/TMB counter by write \$20D1 b0=0 1: auto stop mode, when CTS_TMA overflow will stop CTS_TMA/TMB counter automatically, write \$20D1 b0=1 for next start	R/W	0x0
[1]	START_ENABLE	CTS Start control bit 0: stop CTS_TMA/TMB counter in manual stop mode(0x400E0000[2]=0) 1: start CTS_TMA/TMB counter and auto stop counting when CTS_TMA overflow Notice: Don't set 1 in Low power mode.(If set "1", it cannot make system enter sleep mode normally.)	R/W	0x0
[0]	CTSEN_ENABLE	CTS enable control bit 0: Disabled 1: Enabled	R/W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

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

Bit	Name	Description	Access	Reset Value
[31:30]	--	Reserved	R	0x0
[29]	TMB_INTF	CTS Timer B overflow/underflow interrupts flag read : 0 : idle / busy 1 : timer interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[28]	TMA_INTF	CTS Timer A overflow/underflow interrupts flag read : 0 : idle / busy 1 : timer interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[27:0]	--	Reserved	R	0x0

CTS_TMAPLOAD CTS TMA data								Address : 0x400E_0008
TMAPLOAD [15:8]								
TMAPLOAD [7:0]								

Bit	Name	Description	Access	Reset Value
[15:0]	TMAPLOAD	Capacitive touch sensor TMA data register(TMB counter value when TMA = TMAPLOAD)	R/W	0xFFFF

CTS_TMACOUNT CTS TMA CNT data								Address : 0x400E_000C
TMACOUNT[15:8]								
TMACOUNT[7:0]								

Bit	Name	Description	Access	Reset Value
[15:0]	TMACOUNT	Capacitive touch sensor TMA count register	R	0x0

CTS_TMB0PLOAD CTS preload register for timer B0 Address : 0x400E_0010

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

Bit	Name	Description	Access	Reset Value
[15:0]	TMBDATA	Capacitive touch sensor TMB data register	R/W	0xFFFF

CTS_TMBCOUNT CTS TMB CNT data Address : 0x400E_0014

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

Bit	Name	Description	Access	Reset Value
[15:0]	TMBCOUNT	Capacitive touch sensor TMB count register	R	0x0

CTS_TMBCAP CTS TMB CAP data Address : 0x400E_0018

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

Bit	Name	Description	Access	Reset Value
[15:0]	TMBCAP	TMB counter's value will be captured into TMBCAP while TMA = TMAPLOAD	R	0x0

CTS_PADSEL CTS TMA clock source select (From IO PAD or touch module) Address : 0x400E_001C

15	14	13	12	11	10	9	8
--						PADSEL[9:8]	
7	6	5	4	3	2	1	0
PADSEL[7:0]							

Bit	Name	Description	Access	Reset Value
[31:10]	--	Reserved	R	0x0
[9:0]	1.PADSEL 2.TMA_CLKSRC	1.When CTS_CTRL[19:17] TMA_CLK_SEL = 010, CTS TMA clock source from touch sensor module [9:0]: PAD_IO bit select for touch, can be multiple. 2. When CTS_CTRL[19:17] TMA_CLK_SEL = 001, CTS TMA clock source from GPIO [4:0]: select external GPIO index [4:0] = 0 ~ 24 : IOA[0] ~ IOA[24] [4:0] = 25~30 : IOB[0] ~IOB[5] Note that When TMA_CLK_SEL = 001, source from GPIO,	R/W	0x0



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description					Access	Reset Value
		Need to take care GPIO setting						

CTS_TMB1PLOAD CTS TMB data1								Address : 0x400E_0020
15	14	13	12	11	10	9	8	
TMB1PLOAD[15:8]								
7	6	5	4	3	2	1	0	
TMB1PLOAD[7:0]								

Bit	Name	Description					Access	Reset Value
[15:0]	TMB1PLOAD	Capacitive touch sensor TMB data1 register					R/W	0xFFFF

CTS_TMB2PLOAD CTS TMB data2								Address : 0x400E_0024
15	14	13	12	11	10	9	8	
TMB2PLOAD[15:8]								
7	6	5	4	3	2	1	0	
TMB2PLOAD[7:0]								

Bit	Name	Description					Access	Reset Value
[15:0]	TMB2PLOAD	Capacitive touch sensor TMB data2 register					R/W	0xFFFF

CTS_TMB3PLOAD CTS TMB data3								Address : 0x400E_0028
15	14	13	12	11	10	9	8	
TMB3PLOAD[15:8]								
7	6	5	4	3	2	1	0	
TMB3PLOAD[7:0]								

Bit	Name	Description					Access	Reset Value
[15:0]	TMB3PLOAD	Capacitive touch sensor TMB data3 register					R/W	0xFFFF

CTS_TMB4PLOAD CTS TMB data4								Address : 0x400E_002C
15	14	13	12	11	10	9	8	
TMB4PLOAD[15:8]								
7	6	5	4	3	2	1	0	
TMB4PLOAD[7:0]								

Bit	Name	Description					Access	Reset Value
[15:0]	TMB4PLOAD	Capacitive touch sensor TMB data4 register					R/W	0xFFFF

CTS_TMB5PLOAD CTS TMB data5								Address : 0x400E_0030
15	14	13	12	11	10	9	8	
TMB5PLOAD[15:8]								
7	6	5	4	3	2	1	0	
TMB5PLOAD[7:0]								



GPCM1F SERIES PROGRAMMING GUIDE

Bit	Name	Description				Access	Reset Value
[15:0]	TMB5PLOAD	Capacitive touch sensor TMB data5 register				R/W	0xFFFF

CTS_TMB6PLOAD CTS TMB data6								Address : 0x400E_0034
15	14	13	12	11	10	9	8	TMB6PLOAD[15:8]
7	6	5	4	3	2	1	0	TMB6PLOAD[7:0]

Bit	Name	Description				Access	Reset Value
[15:0]	TMB6PLOAD	Capacitive touch sensor TMB data6 register				R/W	0xFFFF

CTS_TMB7PLOAD CTS TMB data7								Address : 0x400E_0038
15	14	13	12	11	10	9	8	TMB7PLOAD[15:8]
7	6	5	4	3	2	1	0	TMB7PLOAD[7:0]

Bit	Name	Description				Access	Reset Value
[15:0]	TMB7PLOAD	Capacitive touch sensor TMB data7 register				R/W	0xFFFF

CTS_TMB8PLOAD CTS TMB data8								Address : 0x400E_003C
15	14	13	12	11	10	9	8	TMB8PLOAD[15:8]
7	6	5	4	3	2	1	0	TMB8PLOAD[7:0]

Bit	Name	Description				Access	Reset Value
[15:0]	TMB8PLOAD	Capacitive touch sensor TMB data8 register				R/W	0xFFFF

CTS_TMB9PLOAD CTS TMB data9								Address : 0x400E_0040
15	14	13	12	11	10	9	8	TMB9PLOAD[15:8]
7	6	5	4	3	2	1	0	TMB9PLOAD[7:0]

Bit	Name	Description				Access	Reset Value
[15:0]	TMB9PLOAD	Capacitive touch sensor TMB data9 register				R/W	0xFFFF

25. TIMEBASE Control

25.1. Introduction

GPCM1F SERIALS contains TimeBase module for related application. The TimeBase provides a programmable periodic interrupts for sleep or deep sleep modes

25.2. Feature

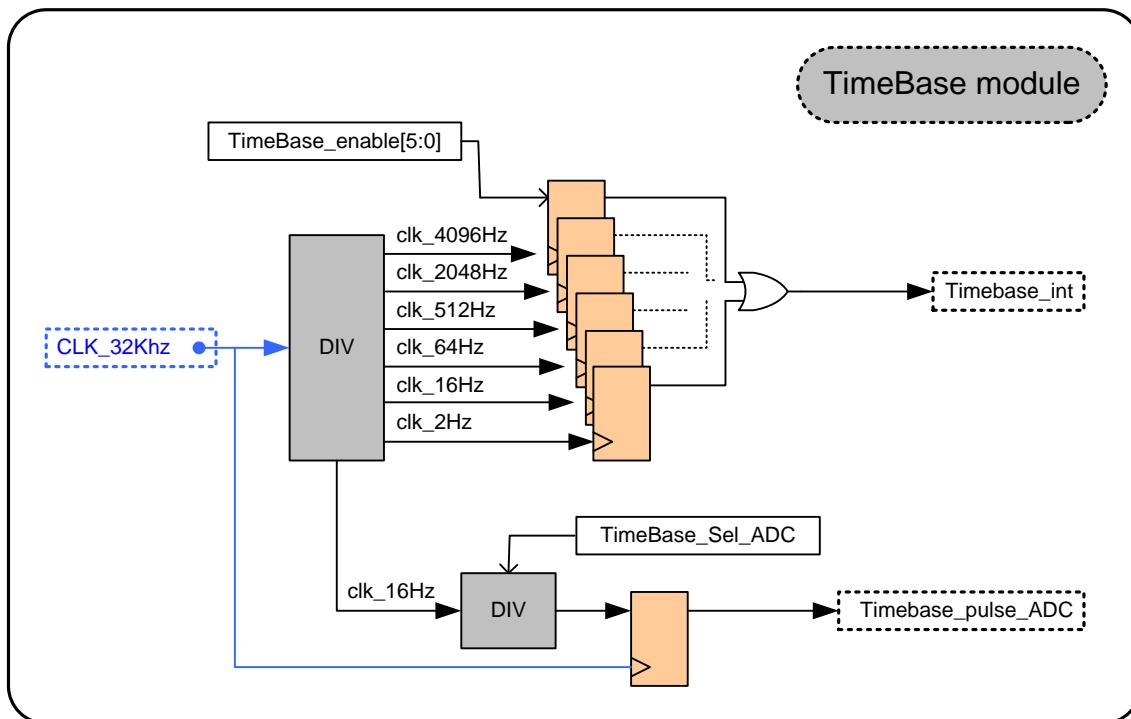
TimeBase module offers the following clock functions:

- Programmable periodic interrupts with respect to 2Hz / 16Hz / 64Hz / 512Hz / 2048Hz / 4096Hz
- Sleep mode wakeup source for ADC wakeup scan from 16Hz, 16/2Hz, ...16/15Hz,16/16Hz

25.3. Note

- 2768Hz CLK Source must be enabled for Timebase clock. The CLK source can be Internal 32KHz or External XTAL 32768Hz.
- Must set CTS_TBCKEN (0x50001010)Bit[10] =1 (enable the respective CLK Control Register) ; otherwise, Register cannot be written any value.
- **GPCM1F has two levels of interrupt control-** M0's internal NVIC interrupt control and GPCM1F's INT Control. Both NVIC & GPCM1F 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) in order to assure each interrupt frequency stays the same.
- When GPCM1F Timebase CLK source is switched from Iosc32K to XTAL32K, we can check the status of XTAL32K by checking CLK_STS(0x50001050)'s bit[3]; however, when X32K STS Flag = 1, we can only assure CLK Src has been switched from Iosc32K to XTAL32K, but XTAL may not be stable at this time. It is recommended to wait for additional one or two 32KHz clocks before turning off Iosc 32K to assure system operates normally. For more information about example codes, please refer to the settings in TimeBase_Initial at pack\Timebase example code.

25.4. Block Diagram



25.5. Function

Normal TimeBase mode

Write TimeBase_enable=1, then 14bit timebase counter will start counting. The counter is clocked by its own dedicated low-speed clock (LSIRC32K), so it can continue counting when system enter sleep/deep sleep mode. Interrupt asserted when counter reach related setting (TimeBase_INTENn[5:0]). The Counter won't stop, hardware will assert interrupt periodically when timebase counter overflow next time. User need to write TimeBase_enable=0 to stop counting.

ADC wakeup source mode

The TimeBase module can also create periodic clock_pulse for ADC wakeup scan purpose. User can set TIMEBASE_CTRL Bit[19:16] TIMEBASE_SEL register for different time interval.(from 16Hz, 8Hz to 1Hz)

25.6. Register Description

Register Map

Base Address : 0x400E_0060					
Name	Description			Address	Access
TIMEBASE_CTRL	TimeBase control register		0x400E_0060		R/W 0x0000_0100
TIMEBASE_STS	TimeBase interrupt status		0x400E_0064		R/W 0x0000_0000

Register

TIMEBASE_CTRL TIMEBASE Control Register								Address : 0x400E_0060	
31	30	29	28	27	26	25	24		
--									
23	22	21	20	19	18	17	16		
--									
15	14	13	12	11	10	9	8	CLRCNT_EN	TB_EN
--									



GPCM1F SERIES PROGRAMMING GUIDE

7	6	5	4	3	2	1	0
--		4KHZ_EN	2KHZ_EN	512HZ_EN	64HZ_EN	16HZ_EN	2HZ_EN

Bit	Name	Description	Access	Reset Value
[31:20]	--	reserved	R	0x0
[19:16]	TIMEBASE_SEL	TimeBase source select for Mic ADC Pulse interval = 62.5ms *(TimeBase_Sel +1) 0: disabled 1: 125ms interval (8Hz) 2: 187.5ms interval 3: 250ms interval (4Hz) 4: 312.5ms interval 14: 937.5ms interval 15: 1s interval (1Hz)	R/W	0x0
[15:10]	--	reserved	R	0x0
[9]	CLRCNT_ENABLE (CLRCNT_EN)	TimeBase clear counter bit 1: write one reset counter and auto set to zero	WOC	0x0
[8]	TB_EN	1: timebase counter enabled 0: timebase disabled	R/W	0x1
[7:6]	--	reserved	R	0x0
[5]	4KHZ_ENABLE (4KHZ_EN)	Timebase 4096Hz interrupt enable bit 1: Enabled 0: Disabled	R/W	0x0
[4]	2KHZ_ENABLE (2KHZ_EN)	Timebase 2048Hz interrupt enable bit 1: Enabled 0: Disabled	R/W	0x0
[3]	512HZ_ENABLE (512HZ_EN)	Timebase 512Hz interrupt enable bit 1: Enabled 0: Disabled	R/W	0x0
[2]	64HZ_ENABLE (64HZ_EN)	Timebase 64Hz interrupt enable bit 1: Enabled 0: Disabled	R/W	0x0
[1]	16HZ_ENABLE (16HZ_EN)	Timebase 16Hz interrupt enable bit 1: Enabled 0: Disabled	R/W	0x0
[0]	2HZ_ENABLE (2HZ_EN)	Timebase 2Hz interrupt enable bit 1: Enabled 0: Disabled	R/W	0x0

TIMEBASE_STS TIMEBASE INT STATUS Address : 0x400E_0064							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--		4KHZ_INTF	2KHZ_INTF	512HZ_INTF	64HZ_INTF	16HZ_INTF	2HZ_INTF

Bit	Name	Description	Access	Reset Value
[31:6]	--	reserved	R	0x0
[5]	4KHZ_INTF_FLAG (4KHZ_INTF)	Timebase 4096Hz interrupt flag read : 0 : 1 : timebase interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[4]	2KHZ_INTF_FLAG (2KHZ_INTF)	Timebase 2048Hz interrupt flag read : 0 : 1 : timebase interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[3]	512HZ_INTF_FLAG (512HZ_INTF)	Timebase 512Hz interrupt flag read : 0 : 1 : timebase interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[2]	64HZ_INTF_FLAG (64HZ_INTF)	Timebase 64Hz interrupt flag read : 0 : 1 : timebase interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[1]	16HZ_INTF_FLAG (16HZ_INTF)	Timebase 16Hz interrupt flag read : 0 : 1 : timebase interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0
[0]	2HZ_INTF_FLAG (2HZ_INTF)	Timebase 2Hz interrupt flag read : 0 : 1 : timebase interrupt triggered write : 0 : no effect 1 : clear these bits	R/W	0x0

26. Watchdog Timer

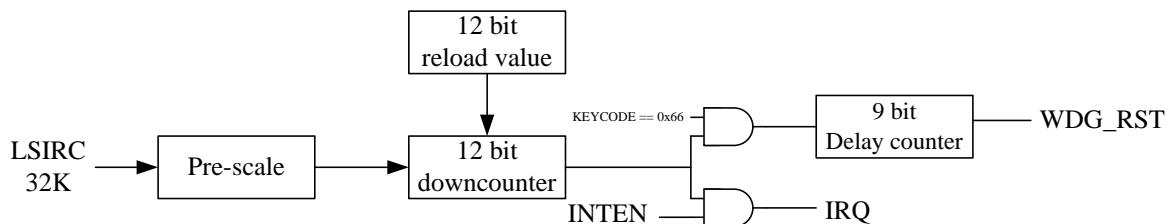
26.1. Introduction

The independent watchdog (WDG) is clocked by its own dedicated low-speed clock (LSIRC32K) and thus stays active even if the main clock fails. The WDG is best suited to applications which require the watchdog to run as a totally independent process outside the main application but has lower timing accuracy.

26.2. Features

- Free-running down-counter
- Clocked from an independent low speed RC oscillator
- Reset (if watchdog activated) when the down-counter value of 0x000 is reached

26.3. Block Diagram



26.4. Function

When the independent watchdog is started by writing the value 0x99 or 0x66 in the Key register (**WDG_KEYCODE.KEYCODE**), the counter starts counting down from the reset value of 0xFFFF. When it reaches the end of count value (0x000) a reset signal is generated (WDG reset).

Whenever the key value 0xAA is written in the **KEYCODE** register, the **WDG_CTRL.RCNT** value is reloaded in the counter and the watchdog reset is prevented.

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;
}
  
```

This implies that it is the case of the reload operation (writing 0xAA).

26.5. Register Description

Register Map

Base Address : 0x400D_0000				
Name	Description	Address	Access	Reset value
WDG_KEYCODE	Watchdog Key code Register	0x400D_0000	R/W	0x0000_0000
WDG_CTRL	Watchdog Control Register	0x400D_0004	R/W	0xFFFF_1210
WDG_STS	Watchdog Status Register	0x400D_0008	R/W	0x0000_0000



GPCM1F SERIES PROGRAMMING GUIDE

Register Function

WDG_KEYCODE Watchdog Key Code Register								Address : 0x400D_0000
31	30	29	28	27	26	25	24	
--								
23	22	21	20	19	18	17	16	
--								
15	14	13	12	11	10	9	8	
--								
7	6	5	4	3	2	1	0	
KEYCODE								

Bit	Name	Description	Access	Reset value
[31:8]	--	Reserved	R	0
[7:0]	KEYCODE	Watchdog Key Code AA = Reloads the watchdog down counter value from WDG_CTRL.RCNT 55 = Enables access to WDG_KEYCODE/WDG_CTRL 44 = Disables access to WDG_KEYCODE/WDG_CTRL 99 = Starts the Watchdog counting without reset function 66 = Starts the Watchdog counting with reset function 00 = Disables Watchdog function Note: If illegal command reset function(WDG_CTRL Bit.12 = 1) is enabled, when an illegal CMD is issued, Watchdog illegal CMD reset is triggered immediately.	R/W	0

WDG_CTRL Watchdog Control Register								Address : 0x400D_0004
31	30	29	28	27	26	25	24	
--				CNT[11:8]				
23	22	21	20	19	18	17	16	
--			CNT[7:0]					
15	14	13	12	11	10	9	8	
--		ILL_RST_EN	--		CLK_SEL			
--								
7	6	5	4	3	2	1	0	
--					TIMEOUT_FLA		--	
					G			

Bit	Name	Description	Access	Reset value
[31:28]	--	Reserved	R	0
[27:16]	CNT	Watchdog counter reload value	R/W	0xFFFF
[15:13]	--	Reserved	R	0
[12]	ILL_RST_EN	illegal command reset function 0 = Disabled 1 = Enabled	R/W	1
[11]	--	Reserved	R	0
[10:8]	CLK_SEL	Watchdog clock pre-scaler 000 = LSIRC/1 001 = LSIRC/4	R/W	010

Bit	Name	Description	Access	Reset value
		010 = LSIRC/8 011 = LSIRC/16 100 = LSIRC/32 101 = LSIRC/64 110 = LSIRC/128 111 = LSIRC/256		
[7:2]	--	Reserved	R	0
[1]	TIMEOUT_FLAG	Watchdog timer timeout Flag 0 = RCNT not down count to zero. 1 = RCNT down count to zero. Note: It's cleared by software writing 1.	R/W	0
[0]	--	Reserved	R	0

WDG_STS Watchdog Status Register								Address : 0x400D_0008
31	30	29	28	27	26	25	24	
		--			--			
23	22	21	20	19	18	17	16	
		--			--			
15	14	13	12	11	10	9	8	
		--			--			
7	6	5	4	3	2	1	0	ILL_RESET_FLAG
		--			--			

Bit	Name	Description	Access	Reset value
[31:2]	--	Reserved	R	0
[1]	ILL_RESET_FLAG	Watchdog illegal access reset flag 1-> illegal cmd wdt rst. Write 1 to clear	R/W	0
[0]	--	Reserved	R	0