芯港微

MCU008 32bit MCU

Revision

Version	Description	Date	
Preliminary	Initial draft	2019-10-20	
V0.1	First release	2019-12-03	
V0.12	Delete I2C SMB	2020-03-20	
V0.15	MCU008 with DES/TDES features	2020-07-07	
V0.17	Add electrical specification, pinout	2021-05-20	
V0.18	Update I2C registers	2022-01-03	

This Page Blank

Contents

Introduction

The MCU008 is a low-power CMOS 32-bit microcontroller based on the Arm Cortex-M0 core architecture, operation with wide range supply voltage from 2.2V ~ 5.5V.

MCU008 is built with compact, high energy efficient MCU processor operate at base frequency 24MHz, precise clock with external clock source. Integrated with 20KB Flash program memory, 2KB (4 x 512 byte) NVM (EEPROM), 4KB SRAM, three (3) 32-bit timers/counter, WDT, SysTick counter, 7 channels PWM, 16 channels 12-bit ADC (analog to digital converter), 28 bi-direction GPIOs, 2 UART, SPI, I2C and SWD debug interface. Peripherals can flexible configuration through different GPIO pins.

Features Overview

- CPU
  - 32-bit Cortex-M0 embedded core
  - base frequency at 24MHz
  - single cycle multiplier
  - AHB bus
  - APB bus
  - DMA AHB master

- MEMORY
  - 4K bytes SRAM
  - 20K bytes Flash
  - 2K bytes NVM (EEPROM) organize in 4 x 512 bytes
  - ICP programming
  - Sector Endurance: 100,000 cycles
  - 10 years Data Retention

- CLOCK
  - Internal 24MHz RC oscillator
  - Internal 32KHz RC oscillator
  - External 8MHz-40MHz oscillator
  - External 32.768KHz RTC oscillator

- INTERRUPT
  - NVIC
  - 16 peripheral function interrupt sources
  - 4 level interrupt priorities (low, high, rising, falling)

- SECURITY (built with PUF and HW encryption engine)
    - DES / TDES
    - SFR option selectable 1. DES, 2. TDES
    - User pass key register
    - Maximum 192 bits user pass key register
    - Enable / disable through SFR

- TIMER/COUNTER
  - Three (3) 32-bit incremental Timers / Counters
  - Configurable as one-stop mode or continuous mode
  - Configurable as wrapping mode or free-run mode
  - Configurable for interrupt generation
  - Configurable for input capture
  - 24-bit system tick timer

- WATCHDOG
  - 32-bit incremental counter
  - Configurable as one-stop mode or continuous mode
  - Configurable as wrapping mode or free-run mode
  - Configurable as normal Timer usage

- GPIO
  - 28 general purpose I/O pins (GPIO)
  - bi-directional
  - Pin individual pull high or pull down configurable
  - Hardware bit set, clear or toggle feature make software easier and faster
  - Capable serve as interrupt request
  - Open drain output configurable
  - drive current configurable
  - 2mA, 4mA, 6mA @3.3V driver capability

- PWM
  - 7 channels
  - SW Dead-zone control
  - Versatile operation configurable
  - Channel pair capable operation in complemental mode
  - configurable independent output

- COMMUNICATION
  - UART 1
  - UART 2
  - SPI
  - I2C

- ADC (Analog to Digital Converter)
  - 12 bits 1MSPS conversion
  - Maximum 16 channels
  - Internal VREF

- MISC SUPPORT
  - GPIO Peripherals functional mapping
  - Built in POR (Power-On Reset)
  - Built in brown out reset (BOR) register configurable
  - Low voltage detection (LVD) register configurable
  - Software control reset
  - Watchdog timer reset
  - SWD interface

- POWER MODE
  - active mode
  - Idle mode
  - sleep mode

- OPERATION ENVIRONMENT
  - 2.2V ~ 5.5V
  - -40℃ ~ +85℃

- PACKAGE
  - TSSOP20
  - LQFP 32/48

Description

Configuration Summary

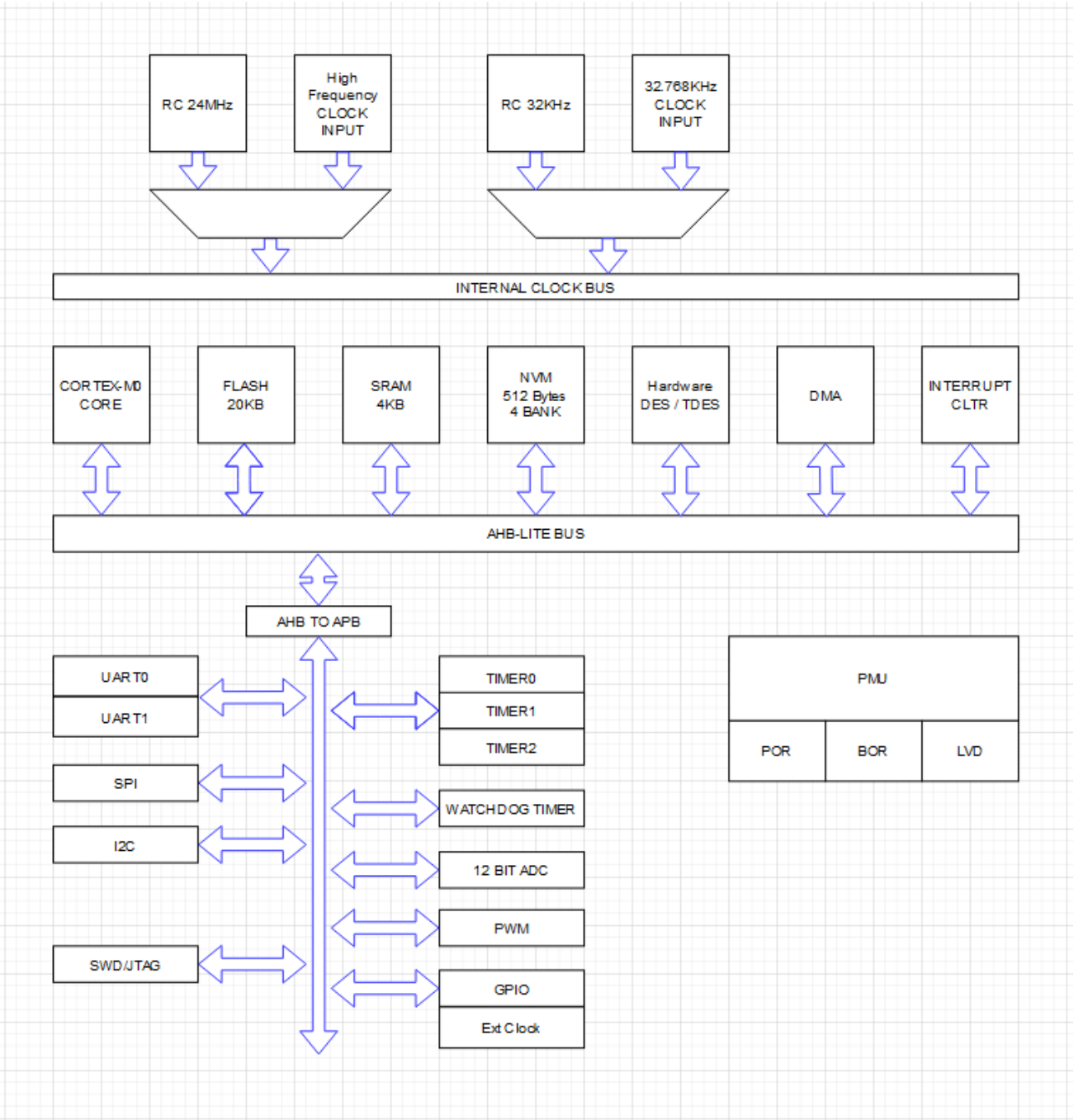
Features	MCU008
Package/Pin	TSSOP20, LQFP32, LQFP48
FLASH	20KB
SRAM	4KB
NVM	512 Byte x 4 bank
GPIO	Max. 28
Interrupt sources	16 peripherals external
Interrupt level	4
ADC	12bit, 500Ksps@3.3V, 1MSPS@5V, 16 channel
PWM	7 channel
UART	2
SPI	1, master
I2C	1, master
SWD debug port	1
32-bit Timer/Counter	3
DES / TDES	

Order Information

Speed	Supply Voltage	Package	Operation Range
24MHz	2.2V – 5.5V	TSSOP20/LQFP-48	-40℃ to +85℃

1. System

1.1 Block Diagram



2. Electrical Characteristics (Typical)

Absolute maximum ratings

Description	Symbol	Min	Typical	Max	unit
DC supply voltage	VDD-VSS	-0.3		5.25	V
GPIO voltage	V_I/V_O	VSS-0.3		VDD+0.3	
Current flow into VDD	I_{VDD}			150	mA
Current flow out from VSS	I_{VSS}			150	

Thermal characteristics

Description	Symbol	Min	Typical	Max	unit
Operating environment temperature	T_{OTG}	-40		85	℃
Storage temperature	T_{STG}	-55		125	

GPIO DC characteristics, VCC=3.3V

Description	**Condition**	**Symbol**	**Min**	**Typical**	**Max**	**Unit**
High level output	I_{OH}=-1.4/-2.9 /-6mA	V_{OH}	2.4			V
Low level output	I_{OH}=1.3 / 2.5 / 4.7mA	V_{OL}			0.4	
Input high voltage		V_{IH}	2.0		3.9	
Input low voltage		V_{IL}	-0.3		0.8	
Switch threshold	CMOS interface	V_{th}	1.25	1.46	1.71	
	Schmitt-falling-trigger		0.98	1.11	1.27	
	Schmitt-rising-trigger		1.53	1.82	2.02	
Schmitt-trigger hystersis			0.55	0.65	0.75	
Input pull-up resistance	V_{IN} = 0	R_{PU}	30	52	98	KΩ
	V_{IN} = VDD	R_{PD}	25	46	97	
Input leakage current	V_{IN} = 0 or VDD	I_{in}	-1		1	µA
Input leakage current with pull up	V_{IN} = 0		-38.2	-56.3	-136	
Input leakage current with pull down	V_{IN} = VDD		31.6	54.9	143	
Tri-state output leakage current		I_O	-1		1	

GPIO DC characteristics, VCC=5.0V

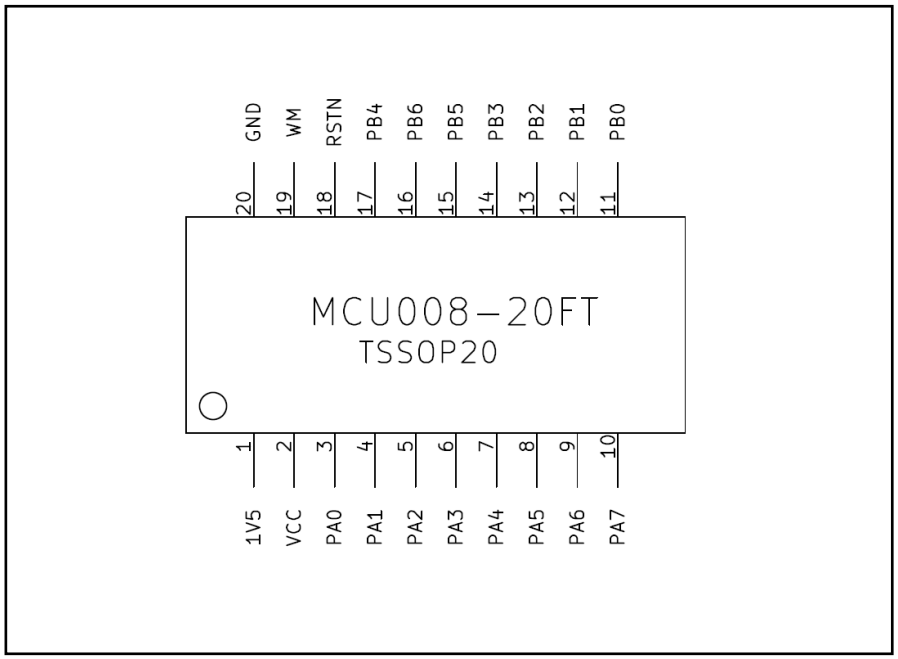
Description	Condition	Symbol	Min	Typical	Max	Unit
High level output	I_{OH}=-3/-6mA	V_{OH}	3.7			V
	I_{OH}=-5.8mA	V_{OH}	2.5			
Low level output	I_{OH}=2.3 / 4.7mA	V_{OL}			0.5	
Input high voltage		V_{IH}	3.15			
	TTL		2			
Input low voltage		V_{IL}			1.35	
	TTL				0.8	
Switch threshold	CMOS interface	V_{th}	2.11	2.3	2.53	
	Schmitt-falling-trigger		1.53	1.68	1.83	
	Schmitt-rising-trigger		2.43	2.63	2.83	
Schmitt-trigger hystersis			0.9	0.95	1	
Input pull-up resistance	V_{IN} = 0	R_{PU}	22K	36K	63K	Ω
	V_{IN} = VDD	R_{PD}	18K	31K	62K	
Input leakage current	V_{IN} = 0 or VDD	I_{in}	-1		1	µA
Input leakage current with pull up	V_{IN} = 0		-89	-169	-301	
Input leakage current with pull down	V_{IN} = VDD		67	159	308	
Tri-state output leakage current		I_O	-1		1	

General operation conditions

Parameter	Symbol	Min	Typical	Max	Unit
AHB clock frequency	f_{HCLK}	0	24	26	MHz
APB clock frequency	f_{PCLK}	0	24	26	
Clock start up time	T_{START}		50	100	µS
Clock jitter (p-p)	J_{P-P}		+/-300		pS
Standard operation voltage	V_{DD}	0	3.0	5.5	V
Core voltage	V_{DDC}	1.4	1.5	1.75	
GPIO input voltage	V_{IN}	-0.3		5.5	
Ambient temperature	T_A	-40		85	°C
POR reset delay time	T_D		300		µS
POR falling setup time	T_L		50		nS
BOR/VDT hystersis voltage			30		mV
BOR/VDT delay			100		nS
ADC SNR	SNR		62		dB
ADC THD	THD		66		
ADC INL (reference to A_{VDD})	INL		+/-3		LSB
ADC DNL (reference to A_{VDD})	DNL		+/-1.5		
ADC setup time from power down	T_{SETUP}		100		µS

3. Pin Out Description

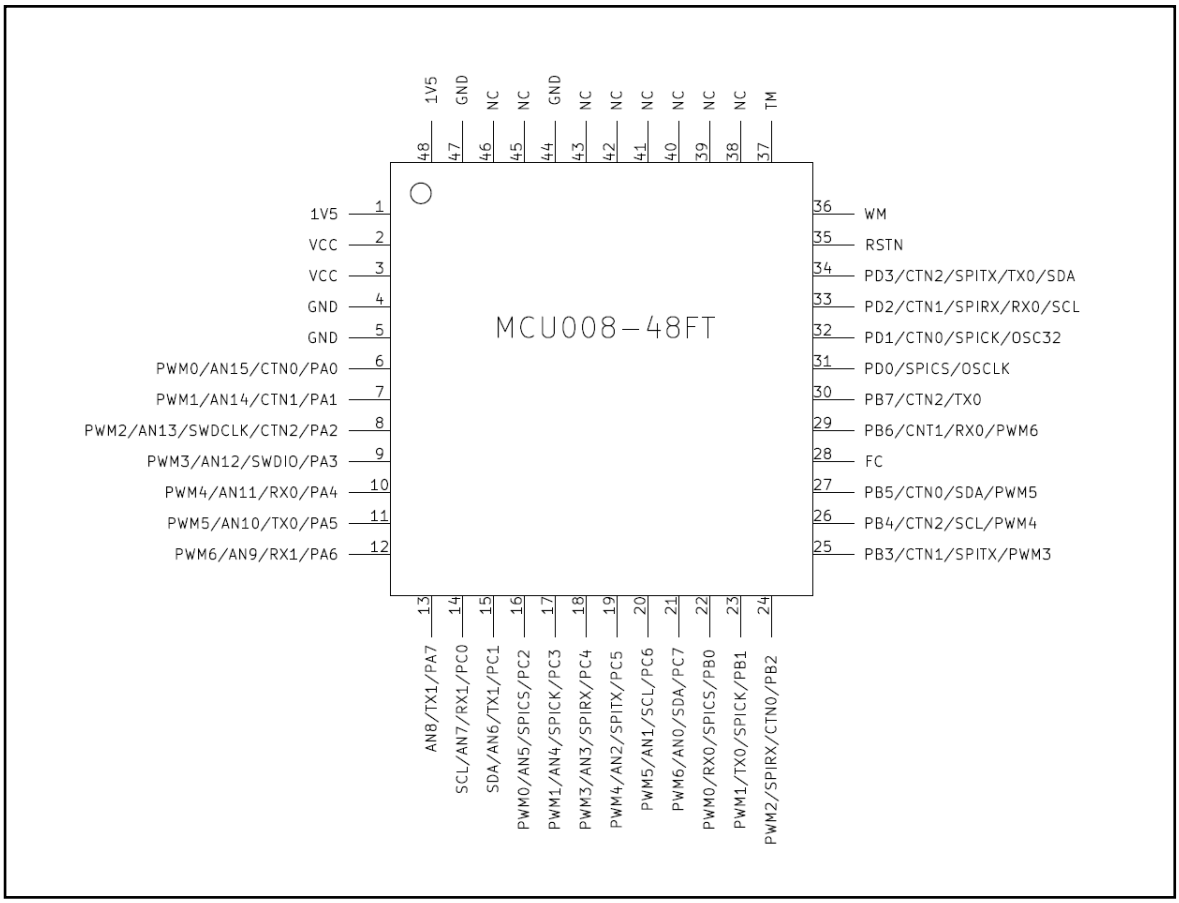
3.1 MCU008-20FT



MCU008-20FT Pinout Function Table
TSSOP20

Pin No.	Pin Name	Pin Type	Direction	Default (Reset) Function	Alternative Function
1	1V5	Power	-		
2	VCC	Power	Input	Vcc (+2.3V~5V)	Vcc (+2.3V~5V)
3	PA0	I/O	Bidirectional	PA0	CTN0/AN15/PWM0
4	PA1	I/O	Bidirectional	PA1	CTN1/AN14/PWM1
5	PA2	I/O	Bidirectional	PA2	SWD CLK /CTN2/AN13/PWM2
6	PA3	I/O	Bidirectional	PA3	SWD IO / AN12/PWM3
7	PA4	I/O	Bidirectional	PA4	USART Rx0/AN11/PWM4
8	PA5	I/O	Bidirectional	PA5	USART Tx0/AN10/PWM5
9	PA6	I/O	Bidirectional	PA6	USART Rx1/AN9/PWM6
10	PA7	I/O	Bidirectional	PA7	USART Tx1/AN8
11	PB0	I/O	Bidirectional	PB0	SPI CS/USART Rx0/PWM0
12	PB1	I/O	Bidirectional	PB1	SPI CLK/USART Tx0/PWM1
13	PB2	I/O	Bidirectional	PB2	CTN0/SPI RxD/PWM2
14	PB3	I/O	Bidirectional	PB3	CTN1/SPI TxD/PWM3
15	PB5	I/O	Bidirectional	PB5	CTN0/SDA/PWM5
16	PB6	I/O	Bidirectional	PB6	CTN1/USART Rx0/PWM6
17	PB4	I/O	Bidirectional	PB4	CTN2/SCL/PWM4
18	RSTN	System	Input	Reset	Reset
19	WM	System	Input	WM	WM
20	GND	Power	Input	GND, Vss	GND, Vss

3.2 MCU008-48FT



MCU008-48FT Pinout Function Table

Pin No.	Pin Name	Pin Type	Direction	Default (Reset) Function	Alternative Function
1	1V5	Power	-		
2	VCC	Power	Input	Vcc (+2.3V~5V)	Vcc (+2.3V~5V)
3	VCC	Power	Input	Vcc(+2.3V~5V)	Vcc(+2.3V~5V)
4	GND	Power	Input	GND, Vss	GND, Vss
5	GND	Power	Input	GND, Vss	GND, Vss
6	PA0	I/O	Bidirectional	PA0	CTN0/AN15/PWM0
7	PA1	I/O	Bidirectional	PA1	CTN1/AN14/PWM1
8	PA2	I/O	Bidirectional	PA2	SWD CLK /CTN2/AN13/PWM2
9	PA3	I/O	Bidirectional	PA3	SWD IO / AN12/PWM3
10	PA4	I/O	Bidirectional	PA4	USART Rx0/AN11/PWM4
11	PA5	I/O	Bidirectional	PA5	USART Tx0/AN10/PWM5
12	PA6	I/O	Bidirectional	PA6	USART Rx1/AN9/PWM6
13	PA7	I/O	Bidirectional	PA7	USART Tx1/AN8
14	PC0	I/O	Bidirectional	PC0	USART Rx1/AN7/SCL
15	PC1	I/O	Bidirectional	PC1	USART Tx1/AN6/SDA
16	PC2	I/O	Bidirectional	PC2	SPI CS/AN5/PWM0
17	PC3	I/O	Bidirectional	PC3	SPI CLK/AN4/PWM1

18	PC4	I/O	Bidirectional	PC4	SPI RxD/AN3/PWM3
19	PC5	I/O	Bidirectional	PC5	SPI TxD/AN2/PWM4
20	PC6	I/O	Bidirectional	PC6	SCL/AN1/PWM5
21	PC7	I/O	Bidirectional	PC7	SDA/AN0/PWM6
22	PB0	I/O	Bidirectional	PB0	SPI CS/USART Rx0/PWM0
23	PB1	I/O	Bidirectional	PB1	SPI CLK/USART Tx0/PWM1
24	PB2	I/O	Bidirectional	PB2	CTN0/SPI RxD/PWM2
25	PB3	I/O	Bidirectional	PB3	CTN1/SPI TxD/PWM3
26	PB4	I/O	Bidirectional	PB4	CTN2/SCL/PWM4
27	PB5	I/O	Bidirectional	PB5	CTN0/SDA/PWM5
28	FC	Power	Input	Filter Capacitor	Filter Capacitor
29	PB6	I/O	Bidirectional	PB6	CTN1/USART Rx0/PWM6
30	PB7	I/O	Bidirectional	PB7	CTN2/USART Tx0
31	PD0	I/O	Bidirectional	PD0	SPI CS/OSC CLK
32	PD1	I/O	Bidirectional	PD1	CTN0/SPI CLK/OSC32
33	PD2	I/O	Bidirectional	PD2	CTN1/SPI RxD/USART Rx0/SCL
34	PD3	I/O	Bidirectional	PD3	CNT2/SPI TxD/USART Tx0/SDA
35	RSTN	System	Input	Reset	Reset
36	WM	System	Input	WM	WM
37	TM	System	Input	TM	TM
38	NC				
39	NC				
40	NC				
41	NC				
42	NC				
43	NC				
44	GND	Power	Input	GND, Vss	GND, Vss
45	NC				
46	NC				
47	GND	Power	Input	GND, Vss	GND, Vss
48	1V5	Power	-		

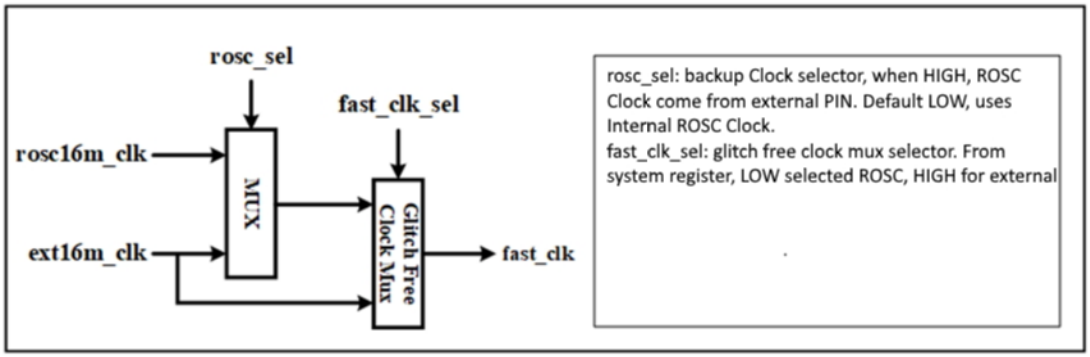
4. System Clock

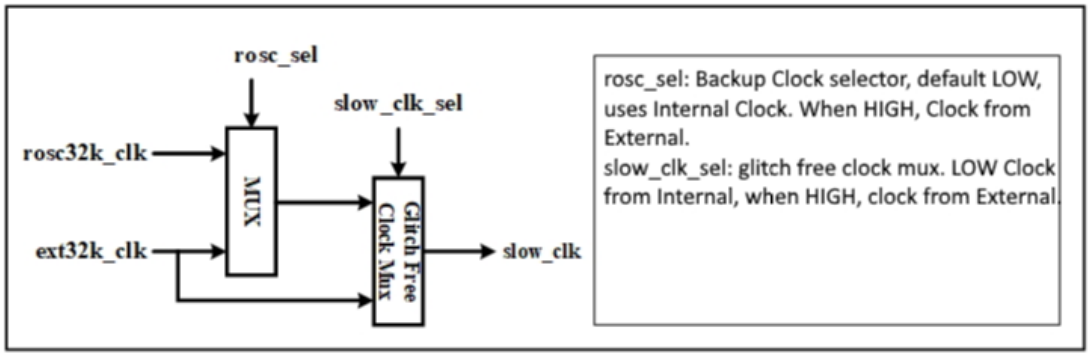
4.1 Description

MCU008 have 4 clock sources:

- 2 Internal clock sources with RC 24MHz high frequency and RC 32.0KHz low frequency clock.
- 2 optional configurable external clock sources with high frequency oscillator (4MHz~25MHz) and external low frequency 32.768KHz RTC oscillator.
- CPU designed to run highest frequency at 25MHz with external clock.,

4.2 Clock Source Selection





5. CPU Core

5.1 Description

- ARM M0 CPU
- AHB BUS (2 bus master)
  - M0 System Bus
  - DMA AHB Master
  - Polarity – M0 System Bus > DMA AHB Master
  - 8 slaves
- AHB to APB bridge
- APB BUS (15 slaves)
- Single bus DMA controller
  - memory to memory
  - memory to peripheral
  - peripheral to memory
  - peripheral support UART, SPI, I2C and ADC

5.2 DMA

- Single bus DMA controller
  - memory-to-memory
  - memory-to-peripheral
  - peripheral-to-memory
  - peripheral-to-peripheral

6. System Control and Reset

A set of Registers controlling the system functionally and behaviors, clocks, resets, peripherals and etc.

6.1 System Address Mapping

6.1.1 Memory Mapping

Address Space	Segment	Physical Slave
0x0000_0000 ~ 0x0000_4FFF	Code	20KB Flash ROM
0x0000_5000 ~ 0x0000_57FF	NVR	Flash NVR
0x0000_5800 ~ 0x0003_FFFF	-	Reserved
0x0004_0000 ~ 0x0004_0FFF	System Mirror	4KB System SRAM
0x2000_0000 ~ 0x2000_0FFF	System	4KB System SRAM
0x2000_1000 ~ 0x20FF_FFFF	-	Reserved
0x2100_0000 ~ 0x2100_4FFF	Code Mirror	20KB Flash ROM
0x2100_5000 ~ 0x3FFF_FFFF	-	Reserved
0x4000_0000 ~ 0x4005_FFFF	Peripheral	APB Peripherals
0x4006_0000 ~ 0x4006_FFFF	System Ctrl	AHB System Control Module
0x4007_0000 ~ 0x4007_FFFF	DMA	AHB DMA Module
0x4008_0000 ~ 0x4008_FFFF	Flash Ctrl	AHB Flash Control Module
0x4009_0000 ~ 0x4009_FFFF	DES	DES Slave Module
0x400A_0000 ~ 0xFFFF_FFFF	-	Reserved

6.1.2 APB Peripherals Mapping

Address Space	Segment	Physical Slave
0x4000_0000 ~ 0x4000_0FFF	APB Peripherals	GPIO
0x4000_1000 ~ 0x4000_7FFF		Reserved
0x4000_8000 ~ 0x4000_8FFF		WatchDog Timer
0x4000_9000 ~ 0x4000_9FFF		Timer0
0x4000_A000 ~ 0x4000_AFFF		Timer1
0x4000_B000 ~ 0x4000_BFFF		Timer2
0x4000_C000 ~ 0x4000_FFFF		Reserved
0x4001_0000 ~ 0x4001_0FFF		UART0
0x4001_1000 ~ 0x4001_1FFF		UART1
0x4001_2000 ~ 0x4001_7FFF		Reserved
0x4001_8000 ~ 0x4001_8FFF		SPI
0x4001_9000 ~ 0x4001_FFFF		Reserved
0x4002_0000 ~ 0x4002_0FFF		I2C
0x4002_1000 ~ 0x4002_7FFF		Reserved
0x4002_8000 ~ 0x4002_8FFF		LVD
0x4002_9000 ~ 0x4002_FFFF		Reserved
0x4003_0000 ~ 0x4003_0FFF		ADC
0x4003_1000 ~ 0x4003_7FFF		Reserved

0x4003_8000 ~ 0x4003_8FFF		PWM0
0x4003_9000 ~ 0x4003_9FFF		PWM1
0x4003_A000 ~ 0x4003_AFFF		PWM2
0x4003_B000 ~ 0x4003_BFFF		PWM3
0x4003_C000 ~ 0x4003_FFFF		Reserved
0x4004_0000 ~ 0x4004_0FFF		Pin Control
0x4004_1000 ~ 0x4005_FFFF		Reserved

6.2 Reset

- POR (Power On Reset)
- BOR (Brown Out Reset)
- LVD (low Voltage Detection)
- Watchdog Timeout Reset
- Software Reset
- External Reset

6.3 Register Description

Base Address 0x4006_0000

6.3.1 Clock Source Selection Register

Reg Intro:	Clock Source Selection						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iClkSrcSel	0x00+	31	2	RESERVED	0	rsvd	Reserved
		1	1	RWC	0	slow_clk_sel	0- ROSC32K 1-EXT32K
		0	0	RWC	0	fast_clk_sel	0- ROSC16M 1-EXT16M

6.3.2 Clock Control Selection Register

Reg Intro:	Clock Control Selection						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iClkCtrlSel	0x10+	31	3	RESERVED	0	rsvd	Reserved
		2	2	RWC	0	apb_clk_sel	0-sys clk; 1-sys div clk
		1	1	RWC	0	sys_clk_sel	0-osc clk; 1-osc div clk
		0	0	RWC	0	osc_clk_sel	0- fast clk; 1-slow clk

6.3.3 OSC Clock Divider Register

Reg Intro:	OSC Clock Divider(need tog)						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iOscClkDiv	0x20	31	5	RESERVED	0	rsvd	Reserved
		4	0	RW	0	value	OSC Clock Divider

6.3.4 System Clock Divider

Reg Intro:	System Clock Divider(need tog)						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iSysClkDiv	0x24	31	5	RESERVED	0	rsvd	Reserved
		4	0	RW	0	value	System Clock Divider

6.3.5 AHB Clock Gate Register

Reg Intro:	AHB Clock Gate						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iAhbClkGate	0x30+	31	1	RESERVED	0	rsvd	Reserved
		1	1	RWC	0	des	AHB DES Clock Gate
		0	0	RWC	0	dma	AHB DMA Clock Gate

6.3.6 APB Clock Gate Register

Reg Intro:	APB Clock Gate						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iApbClkGate	0x40+	31	16	RESERVED	0	rsvd	Reserved
		15	15	RWC	0	gio	APB GPIO Clock Gate
		14	14	RWC	0	wdg	APB WatchDog Clock Gate
		13	11	RWC	0	tmr	APB TIMER Clock Gate
		10	9	RWC	0	sci	APB SCI Clock Gate
		8	8	RWC	0	spi	APB SPI Clock Gate
		7	7	RWC	0	i2c	APB I2C Clock Gate
		6	3	RWC	0	pwm	APB PWM Clock Gate
		2	2	RWC	0	adc	APB ADC Clock Gate
		1	1	RWC	0	lvd	APB LVD Clock Gate
		0	0	RWC	0	pinctl	APB PinCtrl Clock Gate

6.3.7 Work clock Gate Register

Reg Intro:	Work Clock Gate						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iWorkClkGate	0x50+	31	1	RESERVED	0	rsvd	Reserved
		0	0	RWC	0	wdg	WatchDog Work Clock Gate

6.3.8 AHB Block Reset Register

Reg Intro:	AHB Block Reset						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iAhbBlkRst	0x60+	31	1	RESERVED	0	rsvd	Reserved
		1	1	RWC	0	des	AHB DES Clock Gate
		0	0	RWC	0	dma	AHB DMA Clock Gate

6.3.9 APB Block Reset Register

Reg Intro:	APB Block Reset						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iApbBlkRst	0x70+	31	16	RESERVED	0	rsvd	Reserved
		15	15	RWC	0	gio_rst_set	APB GPIO Clock Gate
		14	14	RWC	0	wdg_rst_set	APB WatchDog Clock Gate
		13	11	RWC	0	tmr_rst_set	APB TIMER Clock Gate
		10	9	RWC	0	sci_rst_set	APB SCI Clock Gate
		8	8	RWC	0	spi_rst_set	APB SPI Clock Gate
		7	7	RWC	0	i2c_rst_set	APB I2C Clock Gate
		6	3	RWC	0	pwm_rst_set	APB PWM Clock Gate
		2	2	RWC	0	adc_rst_set	APB ADC Clock Gate
		1	1	RWC	0	lvd_rst_set	APB LVD Clock Gate
		0	0	RWC	0	pinctl_rst_set	APB PinCtrl Clock Gate

6.3.10 Work Block Reset Register

Reg Intro:	Work Block Reset						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iWorkBlkRst	0x80+	31	1	RESERVED	0	rsvd	Reserved
		0	0	RWC	0	wdg_rst_set	WatchDog Work Clock Gate

6.3.11 Software Warm Reset Register

Reg Intro:	Software Warm Reset						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iSoftWarmReset	0x90	31	1	RESERVED	0	rsvd	Software Reset
		0	0	W1P	0	set	Software Warm Reset

6.3.12 Software Cold Reset Register

Reg Intro:	Software Cold Reset						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iSoftColdReset	0x94	31	1	RESERVED	0	rsvd	Software Reset
		0	0	W1P	0	set	Software Cold Reset

6.3.13 Reserved Register

6.3.14 RCOSC Control Register

Reg Intro:	RCOSC						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iRCOSC	0xA0+	31	0	RESERVED	0	rsvd	Reserved

6.3.15 Chip Mode Register

Reg Intro:	Chip Mode						
Reg Name	**Reg Offset**	**Reg Msb**	**Reg Lsb**	**Reg Type**	**Reg Rstval**	**Reg Function**	**Comment**
iChipMode	0xB0	31	3	RESERVED	0	rsvd	Reserved
		2	2	RO	0	awm	Aux Work Mode
		1	1	RO	0	wm	Work Mode
		0	0	RO	0	tm	Test Mode

7. DMA

Support single bus DMA

7.1 DMA Programmer's model

- Single bus DMA controller
- memory-to-memory
- memory-to-peripheral
- peripheral-to-memory
- peripheral-to-peripheral

7.2 Programming a DMA channel

To program a DMA channel:

1. Choose a free DMA channel with the necessary priority. DMA channel 0 has the highest priority and DMA channel 7 have the lowest priority.

2. Clear any pending interrupts on the channel you want to use by writing to the DMACIntTCClear and DMACIntErrClr Registers. See *Interrupt Terminal Count Clear Register* and *Interrupt Error Clear Register*.
   The previous channel operation might have left interrupts active.

3. Write the source address into the DMACCxSrcAddr Register. See *Channel Source Address Registers*.

4. Write the destination address into the DMACCxDestAddr Register. See *Channel Destination Address Registers*.

5. Write the address of the next LLI into the DMACCxLLI Register. See *Channel Linked List Item Registers*. If the transfer consists of a single packet of data, you must write 0 into this register.

6. Write the control information into the DMACCxControl Register. See *Channel Control Registers*.

7. Write the channel configuration information into the DMACCxConfiguration Register. See *Channel Configuration Registers*. If the Enable bit is set, then the DMA channel is automatically enabled.

7.3 Summary of DMA Register

Base Address 0x4007_0000

Name	Address (base+)	Type	Reset value	Description
DMACIntStatus	0x000	RO	0x00	Interrupt Status Register
DMACIntTCStatus	0x004	RO	0x00	Interrupt Terminal Count Status Register
DMACIntTCClear	0x008	WO	-	Interrupt Terminal Count Clear Register
DMACIntErrorStatus	0x00C	RO	0x00	Interrupt Error Status Register

DMACIntErrClr	0x010	WO	-	Interrupt Error Clear Register
DMACRawIntTCStatus	0x014	WO	-	Raw Interrupt Terminal Count Status Register
DMACRawIntErrorStatus	0x018	RO	-	Raw Error Interrupt Status Register
DMACEnbldChns	0x01C	RO	0x00	Enabled Channel Register
DMACSoftBReq	0x020	R/W	0x0000	Software Burst Request Register
DMACSoftSReq	0x024	R/W	0x0000	Software Single Request Register
DMACSoftLBReq	0x028	R/W	0x0000	Software Last Burst Request Register
DMACSoftLSReq	0x02C	R/W	0x0000	Software Last Single Request Register
DMACConfiguration	0x030	R/W	0x0000	Configuration Register
DMACSync	0x034	R/W	0x0000	Synchronization Register
DMACC0SrcAddr	0x100	R/W	0x00000000	Channel Source Address Registers
DMACC0DestAddr	0x104	R/W	0x00000000	Channel Destination Address Registers
DMACC0LLI	0x108	R/W	0x00000000	Channel Linked List Item Registers
DMACC0Control	0x10C	R/W	0x00000000	Channel Control Registers
DMACC0Configuration	0x110	R/W	0x00000	Channel Configuration Registers
DMACC1SrcAddr	0x120	R/W	0x00000000	Channel Source Address Registers
DMACC1DestAddr	0x124	R/W	0x00000000	Channel Destination Address Registers
DMACC1LLI	0c128	R/W	0x00000000	Channel Linked List Item Registers
DMACC1Control	0x12C	R/W	0x00000000	Channel Control Registers
DMACC1Configuration	0x130	R/W	0x00000	Channel Configuration Registers
DMACC2SrcAddr	0x140	R/W	0x00000000	Channel Source Address Registers
DMACC2DestAddr	0x144	R/W	0x00000000	Channel Destination Address Registers
DMACC2LLI	0x148	R/W	0x00000000	Channel Linked List Item Registers
DMACC2Control	0x14C	R/W	0x00000000	Channel Control Registers
DMACC2Configuration	0x150	R/W	0x00000	Channel Configuration Registers

DMACC3SrcAddr	0x160	R/W	0x00000000	Channel Source Address Registers
DMACC3DestAddr	0x164	R/W	0x00000000	Channel Destination Address Registers
DMACC3LLI	0x168	R/W	0x00000000	Channel Linked List Item Registers
DMACC3Control	0x16C	R/W	0x00000000	Channel Control Registers
DMACC3Configuration	0x170	R/W	0x00000	Channel Configuration Registers
DMACC4SrcAddr	0x180	R/W	0x00000000	Channel Source Address Registers
DMACC4DestAddr	0x184	R/W	0x00000000	Channel Destination Address Registers
DMACC4LLI	0x188	R/W	0x00000000	Channel Linked List Item Registers
DMACC4Control	0x18C	R/W	0x00000000	Channel Control Registers
DMACC4Configuration	0x190	R/W	0x00000	Channel Configuration Registers
DMACC5SrcAddr	0x1A0	R/W	0x00000000	Channel Source Address Registers
DMACC5DestAddr	0x1A4	R/W	0x00000000	Channel Destination Address Registers
DMACC5LLI	0x1A8	R/W	0x00000000	Channel Linked List Item Registers
DMACC5Control	0x1AC	R/W	0x00000000	Channel Control Registers
DMACC5Configuration	0x1B0	R/W	0x00000	Channel Configuration Registers
DMACC6SrcAddr	0x1C0	R/W	0x00000000	Channel Source Address Registers
DMACC6DestAddr	0x1C4	R/W	0x00000000	Channel Destination Address Registers
DMACC6LLI	0x1C8	R/W	0x00000000	Channel Linked List Item Registers
DMACC6Control	0x1CC	R/W	0x00000000	Channel Control Registers
DMACC6Configuration	0x1D0	R/W	0x00000	Channel Configuration Registers
DMACC7SrcAddr	0x1E0	R/W	0x00000000	Channel Source Address Registers
DMACC7DestAddr	0x1E4	R/W	0x00000000	Channel Destination Address Registers
DMACC7LLI	0x1E8	R/W	0x00000000	Channel Linked List Item Registers
DMACC7Control	0x1EC	R/W	0x00000000	Channel Control Registers

DMACC7Configuration	0x1F0	R/W	0x00000	Channel Configuration Registers
DMACPeriphID0	0xFE0	RO	0x80	DMACPeriphID0 Register
DMACPeriphID1	0xFE4	RO	0x10	DMACPeriphID1 Register
DMACPeriphID2	0xFE8	RO	0x04	DMACPeriphID2 Register
DMACPeriphID3	0xFEC	RO	0x0A	DMACPeriphID3 Register
DMACPCellID0	0xFF0	RO	0x0D	DMACPCellID0 Register
DMACPCellID1	0xFF4	RO	0xF0	DMACPCellID1 Register
DMACPCellID2	0xFF8	RO	0x05	DMACPCellID2 Register
DMACPCellID3	0xFFC	RO	0xB1	DMACPCellID3 Register
DMACITCR	0x500	R/W	0x0	Test Control Register
DMACITOP1	0x504	R/W	0x0000	Integration Test Output Register 1
DMACITOP2	0x508	R/W	0x0000	Integration Test Output Register 2
DMACITOP3	0x50C	R/W	0x0	Integration Test Output Register 3

8. MEMORY

8.1 Program (Flash) Memory

- 20K Bytes Flash
- 4K bytes SRAM
- 2K bytes NVR (EEPROM)
- NVR organized in 4 x512 Bytes
- 2KB block protection
- Support In-circuit programming (ICP)
- 100,000 sector erase cycles
- 10 years data retention

8.2 Flash Mapping

(RO = Read Only, RW = Read and Write, NA = Not Accessible)

Base Address	Size (byte)	Block	Utility	Used By	Authorization			
					Erased	ICP	IAP	TEST
0x4008_0038	476	CSR	Reserved					
0x4008_0034	4		Remap		RW	RW	RW	RW
0x4008_0030	4		Interrupt		RW	RW	RW	RW
0x4008_002C	4		Timing_Set4		RW	RW	RW	RW
0x4008_0028	4		Timing_Set3		RW	RW	RW	RW
0x4008_0024	4		Timing_Set2		RW	RW	RW	RW
0x4008_0020	4		Timing_Set1		RW	RW	RW	RW
0x4008_001C	4		Timing_Set0		RW	RW	RW	RW
0x4008_0018	4		Timer_Set		RW	RW	RW	RW
0x4008_0014	4		Pin_Set		RW	RW	RW	RW
0x4008_0010	4		Read_Delay		RW	RW	RW	RW
0x4008_000C	4		Write_Mode		RW	RW	RW	RW
0x4008_0008	4		Flash_Unlock		RW	RW	RW	RW
0x4008_0004	4		SWD_Unlock		RW	RW	RW	RW
0x4008_0000	4		Chip_Erase		RW	RW	RW	RW
0x0000_5800	512	Flash NVR 5	Factory Reserved	Factory	RW	RO	RO	RW
0x0000_57FC	4	Flash NVR 4	Chip ID	User	RW	RO	RO	RW
0x0000_57F8	4		Chip Info	User	RW	RO	RO	RW
0x0000_57F0	8		SWD Key	User	RW	NA	NA	RW
0x0000_57E8	8		Flash Key	User	RW	NA	NA	RW
0x0000_57E4	4		Flash Lock Bits Set1	User/SWD	RW	NA	NA	RW
0x0000_57E0	4		Flash Lock Bits Set0	User/M0	RW	NA	NA	RW
0x0000_57DC	4		SYS CFG0	User	RW	RW	RW	RW
0x0000_5600	480		Parameters	User	RW	RW	RW	RW
0x0000_5400	512	Flash NVR 3	Bootloader、ISP User Program	User	RW	RW	RW	RW
0x0000_5200	512	Flash NVR 2						
0x0000_5000	512	Flash NVR 1						
0x0000_0000	20K	Flash Main						

8.3 DAP Protection

	Size	Bit	Name	Description
SWD Key	8	63:33	Reserved	
		32	swd_key_disable	1'b1: SWD key is disabled 1'b0: SWD key is enabled
		31:0	swd_key_pattern	
Flash Key	8	63:33	Reserved	
		32	flash_key_disable	1'b1: flash key is disabled 1'b0: flash key is enabled
		31:0	flash_key_pattern	
Flash Lock Bits Set1/0	4	31:28	Reserved	
		27:26	Reserved	2'b11: Read and Write 2'b10: Write Only 2'b01: Read Only 2'b00: not accessible
		25:24	NVR3_lock_bits	
		23:22	NVR2_lock_bits	
		21:20	NVR1_lock_bits	
		19:18	block9_lock_bits	
		17:16	block8_lock_bits	
		15:14	block7_lock_bits	
		13:12	block6_lock_bits	
		11:10	block5_lock_bits	
		9:8	block4_lock_bits	
		7:6	block3_lock_bits	
		5:4	block2_lock_bits	
		3:2	block1_lock_bits	
		1:0	block0_lock_bits	

8.4 Register Description

CSR base address 0x4008_0000, refer to 4.2 Address Mapping table, following describe the register usage.

8.4.1 0x00 Chip_Erase

High	Low	Name	Access	Default	Function
31	1	Reserved			
0	0	csr_erase_req	RW	1'b0	Auto Clear 1'b1：Start To Erase 1'b0：Erase Done

8.4.2 0x04 SWD_Unlock

High	Low	Name	Access	Default	Function
31	4	Reserved			
3	3	por_done	RO		Power-on Reset of Flash IP is Done
2	1	cur_work_mode	RO		Value of Work Mode Pins
0	0	swd_key_verified	RO		

8.4.3 0x08 Flash_Unlock

High	Low	Name	Access	Default	Function
31	1	Reserved			
0	0	flash_key_verified	RO		

8.4.4 0x0C Write_Mode

High	Low	Name	Access	Default	Function
31	3	Reserved			
2	2	csr_bus_write_done	RO	1'b1	1'b1：write/erase is done 1'b0：write/erase is in process
1	1	csr_bus_write_erase	RW	1'b0	Auto Clear 1'b1：Erase when write to sector 1'b0：Not Erase
0	0	csr_bus_write_mode	RW	1'b0	1'b1：Hold HREADY of AHB Bus when write/erase 1'b0：Not Hold HREADY

9. SECURITY

9.1 DES / TDES algorithm

- Built-in DES / TDES encryption engine
- User provide pass key

9.2 Register Description

CSR base address 0x4009_0000

9.2.1 0x00 DES_Enable

High	Low	Name	Access	Default	Function
31	1	Reserved			
0	0	DES Enable	RW	1'b0	1'b1：Enable DES module 1'b0：Disable DES module

9.2.2 0x10 DES_Mode

High	Low	Name	Access	Default	Function
31	9	Reserved			
8	8	Quality	RW	1'b0	1'b1：QC mode 1'b0：normal mode
7	6	Algorithm Sel	RO	1'b1	2'b11：Reserved 2'b10：Reserved 2'b01：TDES 2'b00：DES
5	5	Reserved			Reserved
4	4	User Pass Key Register	RW	1'b0	1'b1：Enable 1'b0： Disable
3	1	Reserved			Reserved
0	0	DES function	RW	1'b0	1'b1：Encrypt Mode 1'b0：Decrypt Mode

9.2.3 0x20 DES_Status

High	Low	Name	Access	Default	Function
31	9	Reserved			
8	8	DES busy	RO	1'b0	1'b1: Busy 1'b0: Not Busy
7	5	Reserved			
4	4	Input Ready	RO	1'b0	1'b1: Ready 1'b0: Not Ready
3	1	Reserved			
0	0	Output Ready	RO	1'b0	1'b1: Ready 1'b0: Not Ready

9.2.4 0x30 DES_DinL

High	Low	Name	Access	Default	Function
31	0	Data input LSB32	RW	0x0	Input Data LSB32

9.2.5 0x40 DES_DinH

High	Low	Name	Access	Default	Function
31	0	Data input MSB32	RW	0x0	Input Data MSB32

9.2.6 0x50 DES_DoutL

High	Low	Name	Access	Default	Function
31	0	Output Data LSB32	RW	0x0	Output Data LSB32

9.2.7 0x60 DES_DoutH

High	Low	Name	Access	Default	Function
31	0	Output Data MSB32	RW	0x0	Output Data MSB32

9.2.8 0x70/0x80/0x90/0xA0/0xB0/0xC0 DES_key0~key5

High	Low	Name	Access	Default	Function
31	0	DES Key Registers	RW	0x0	User provide key 192bits

10. Interrupts

Level	Source
0	WatchDog
1	DMA
2	ADC
3	LVD
4	PWM0
5	PWM1
6	PWM2
7	PWM3
8	UART0
9	UART1
10	TIMER0
11	TIMER1
12	TIMER2
13	I2C
14	SPI
15	GPIO
16	FLASH

11. GPIOs

11.1 General Characteristics

- Maximum 28 bi-direction General Programmable Input Output ports
- Share peripherals functions configuration
- Input with pull-high pull-low configuration
- Output with pull-high pull-low configuration
- Support open drain (OD) configuration
- Support input debouncing with versatile configuration
- Support output driving current configuration (max 6mA@3.3V)
- Support interrupt configuration
- Support input capture for Timer

11.2 Address Base

0x4000_0000

11.3 GPIO Set/Clear/Toggle feature

Each GPIO Register occupied 4 word space.

Address offset	Type	Operations
0x 0	Common WRITE	GPIO = DATA_WRITE
0x 4	Bit SET	GPIO = GPIO | DATA_WRITE
0x 8	Bit CLR	GPIO = GPIO & ~DATA_WRITE
0x C	Bit Toggle	GPIO = GPIO ^ DATA_WRITE

By using bit set/clr/toggle fonction, software can efficiently change every GPIO pin individually, this avoids common read-modify-write operation.

11.4 Register Description

Name	Address	Reset value	Description
GPIO_DI	0x000	-	pin input
GPIO_DO	0x01x	0x0000_0000	pin output
GPIO_OE	0x02x	0x0000_0000	pin direction(1-output/0-input)
GPIO_IS	0x04x	0x0000_0000	interrupt status
GPIO_LV	0x05x	0x0000_0000	interrupt type(0-edge/1-level)
GPIO_PE	0x06x	0x0000_0000	enable Positive sense
GPIO_NE	0x07x	0x0000_0000	enable Negative sense
DB_EN	0x08x	0x0000_0000	debouncing start enable (trigger)
DB_LMT	0x09x	0x0000_0000	debouncing limit
CTN0_LV	0x0Ax	0x0000_0000	interrupt type(0-edge/1-level) for CTN0
CTN0_PE	0x0Bx	0x0000_0000	enable Positive sense for CTN0
CTN0_NE	0x0Cx	0x0000_0000	enable Negative sense for CTN0
CTN1_LV	0x0Dx	0x0000_0000	interrupt type(0-edge/1-level) for CTN1

CTN1_PE	0x0Ex	0x0000_0000	enable Positive sense for CTN1
CTN1_NE	0x0Fx	0x0000_0000	enable Negative sense for CTN1
CTN2_LV	0x10x	0x0000_0000	interrupt type(0-edge/1-level) for CTN2
CTN2_PE	0x11x	0x0000_0000	enable Positive sense for CTN2
CTN2_NE	0x12x	0x0000_0000	enable Negative sense for CTN2
DB_BEN	0x13x	0x0000_0000	pin debouncing enable

Data In Register, GPIO_DI

The GPIO_DI register reflect all the GPIO pins.

By read from it, software can determine the logic level of each pin.

Bit	Name	Access	Reset	Description
31:0	DI	RO	-	Input data

Data Out Register, GPIO_DO

The GPIO_DO register set the output value of each GPIO pin.

Only pin set with output direction can pass the value to the GPIO pin.

Bit	Name	Access	Reset	Description
31:0	DO	RW	All 0	Output data

Data Output Enable Register, GPIO_OE

The GPIO_OE register is the data direction control register. Bits set to 1 configure corresponding pin to be an output. Clearing a bit configures the pin to be input. All bits are cleared by a reset. Therefore, the GPIO pins are input by default.

Bit	Name	Access	Reset	Description
31:0	OE	RW	All 0	Bit set (1), pin output Bit cleared (0), pin input

Interrupt Status Register, GPIO_IS

The GPIO_IS register is the interrupt status register. Bits read HIGH in GPIO_IS reflect the status of interrupts trigger conditions detected.

Bit	Name	Access	Reset	Description
31:0	IS	RW	All 0	Reflect interrupt status of each GPIO pin Write 0, or use clear operation can reset IS to zero

Interrupt Type Register, GPIO_LV

The GPIO_LV register configure the sensitivity type of the GPIO interrupt. A bit of HIGH means level sensitivity, and LOW means edge sensitivity.

Bit	Name	Access	Reset	Description
31:0	LV	RW	All 0	Bit set, interrupt occur at certain logic level Bit clear, interrupt occur at certain signal edge

Positive Detect Enable Register, GPIO_PE

GPIO_PE register enables interrupt when positive condition detected.

Bit	Name	Access	Reset	Description
31:0	PE	RW	All 0	If set, Logic HIGH or Signal Rising condition will trigger an interrupt

Negative Detect Enable Register, GPIO_NE

GPIO_NE register enables interrupt when negative condition detected.

Bit	Name	Access	Reset	Description
31:0	NE	RW	All 0	Bit set, If Logic is LOW or signal falling edge detected, an interrupt will be triggered

Debouncing Enable Register, DB_EN

DB_EN register enables GPIO pin with debouncing function.

Bit	Name	Access	Reset	Description
31:1	-	-	-	-
0	DB_EN	RW	0	Bit set, the debouncing counter enabled

Debouncing Counter Limit Register, DB_LMT

DB_LMT register is used to set the limit of debouncing counter.

Bit	Name	Access	Reset	Description
31:20	-	-	-	-
19:0	DB_LMT	RW	All 0	DB_LMT register is used to set the limit of debouncing counter.

Input Capture Type for CTN0 Register, CTN0_LV

The CTN0_LV register configure the sensitivity type of the CTN0 input capture. A bit of HIGH means level sensitivity, and LOW means edge sensitivity.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN0_LV	RW	All 0	Bit set, input capture occurs at certain logic level Bit clear, input capture occurs at certain signal edge

Positive Detect Enable for CTN0 Register, CTN0_PE

CTN0_PE register enables input capture when positive condition detected.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN0_PE	RW	All 0	If set, Logic HIGH or Signal Rising condition will trigger an input capture

Negative Detect Enable for CTN0 Register, CTN0_NE

CTN0_NE register enables input capture when negative condition detected.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-

3:0	CTN0_NE	RW	All 0	Bit set, If Logic LOW or Signal Falling detected, an input capture will be triggered

Input Capture Type for CTN1 Register, CTN1_LV

The CTN1_LV register configure the sensitivity type of the CTN1 input capture. A bit of HIGH means level sensitivity, and LOW means edge sensitivity.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN1_LV	RW	All 0	Bit set, input capture occurs at certain logic level Bit clear, input capture occurs at certain signal edge

Positive Detect Enable for CTN1 Register, CTN1_PE

CTN1_PE register enables input capture when positive condition detected.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN1_PE	RW	All 0	If set, Logic HIGH or Signal Rising condition will trigger an input capture

Negative Detect Enable for CTN1 Register, CTN1_NE

CTN1_NE register enables input capture when negative condition detected.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN1_NE	RW	All 0	Bit set, If Logic LOW or Signal Falling detected, an input capture will be triggered

Input Capture Type for CTN2 Register, CTN2_LV

The CTN2_LV register configure the sensitivity type of the CTN2 input capture. A bit of HIGH means level sensitivity, and LOW means edge sensitivity.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN2_LV	RW	All 0	Bit set, input capture occurs at certain logic level Bit clear, input capture occurs at certain signal edge

Positive Detect Enable for CTN2 Register, CTN2_PE

CTN2_PE register enables input capture when positive condition detected.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-

3:0	CTN2_PE	RW	All 0	If set, Logic HIGH or Signal Rising condition will trigger an input capture

Negative Detect Enable for CTN2 Register, CTN2_NE

CTN2_NE register enables input capture when negative condition detected.

Bit	Name	Access	Reset	Description
31:4	-	-	-	-
3:0	CTN2_NE	RW	All 0	Bit set, If Logic LOW or Signal Falling detected, an input capture will be triggered

Debouncing Bit Enable Register, DB_BEN

DB_EN register enables debouncing function for every GPIO.

Bit	Name	Access	Reset	Description
31:28	-	-	-	-
27:0	DB_BEN	RW	All 0	Bit set, the debouncing function enabled for the corresponding pin

11.5 Input Capture for Timer

Combined with Timer, input capture can be realized both in level and edge mode.
In MCU008, there are three timers. Each timer can capture one of four GPIOs.
The relationship between timer and GPIO is illustrated in the following figure.

TIMER	GPIO
TIMER0	PAD_PA0、PAD_PB2、PAD_PB5、PAD_PD1
TIMER1	PAD_PA1、PAD_PB3、PAD_PB6、PAD_PD2
TIMER2	PAD_PA2、PAD_PB4、PAD_PB7、PAD_PD3

12. Timer/Counter

12.1 Programmable timer

- 32-bit incremental counter and 32-bit compare register
- Selectable one-stop mode, wrapping mode or free-run mode.
- Interrupt generation.
- Input capture.

12.2 Register Description

Base Address
Timer0 0x4000_9000
Timer1 0x4000_A000
Timer2 0x4000_B000

Name	Address	Reset	Description
TMR_CNT	0x000	Unknown	Timer count register
TMR_CMP	0x004	Unknown	Timer load register
TMR_CTL	0x008	0x0000_0000	Timer control register
TMR_LOCK	0x00C	0x0000_0000	Timer lock register

Timer Counter Register (TMR_CNT)

Bits	Name	Type	Description
31:0	Counter	RO	Timer Counter Register

Timer Compare Register (TMR_CMP)

Bits	Name	Type	Description
31:0	Compare	RW	Timer Compare Register

Timer Control Register (TMR_CTL)

Bits	Name	Type	Description
0	EN	RW	Timer Enable.
1	CC	WO	Clear Counter. Write 1 to this bit clears TMR_CNT, always read as 0
3:2	OM	RW	Operating Mode : 00 : Continuous wrapping mode 10 : Continuous free-run mode x1 : One-stop mode
4	IE	RW	Interrupt Enable
5	EE	RW	Watchdog Reset Enable (TIMER0 only)
6	IS	RW	Interrupt Status
7	ES	RW	Watchdog Reset Status (TIMER0 only)
8	FE	RW	Input Capture Enable
9	FM	RW	Input Capture Mode, 0- Edge Mode, 1 – Level Mode
31:10	-	-	RESERVED

Timer Lock Register (TMR_LOCK)

Bits	Name	Type	Description
31:0	Lock	RW	Timer Write Lock Register (TIMER0 only)

12.3 Function Description

Timer Operation

After a system reset, bits in the Control Register are reset, while the Counter Register and Compare Register is unknown.

The Timer is enabled when the EN bit in the Control Register is set and the counter starts running. The Control Register and Compare Register must be set to appropriate values first before enabling the Timer. The Counter Register can only be reset by writing 1 to CC bit in the Control Register.

The OM bits in the Control Register defines the counter's operation mode. See table below:

Mode	Description
Wrapping mode	The counter generates an interrupt at a constant interval, reset to 0 after reaching the value in compare register, and continues to count up.. This is the default mode.
Free-run mode	The counter reset to0 after reaching its maximum value, i.e. 0xFFFF_FFFF, and continues to count up.
One-stop mode	The counter generates an interrupt once. When the counter reaches the value in compare register, it automatically disabled until you re-enable it.

When the EN bit is cleared the counter value freezes. All other register values are retained.

When the value in counter matches the value in compare register, the interrupt status bit in control register is set and a interrupt request is generated to the system interrupt controller (NVIC) if interrupt status bit in control register is also set. The interrupt status bit is cleared only through writing 1 to this bit, or by system reset.

Input Capture Function

When FE bit in TMR_CTL is set, the input capture function is enabled. The timer begins to counter the input signal according to FM.

When FM is low, the input is a pulse, and the timer will count the number of the pulse. When FM is high, the input is a level，and the timer will count the width of the level.

When the counter reaches the limit, an interrupt can be generated.

13. WatchDog Timer

13.1 Overview

The WatchDog Timer uses low-speed clock to count，and can generate interrupt and reset according to the configuration.

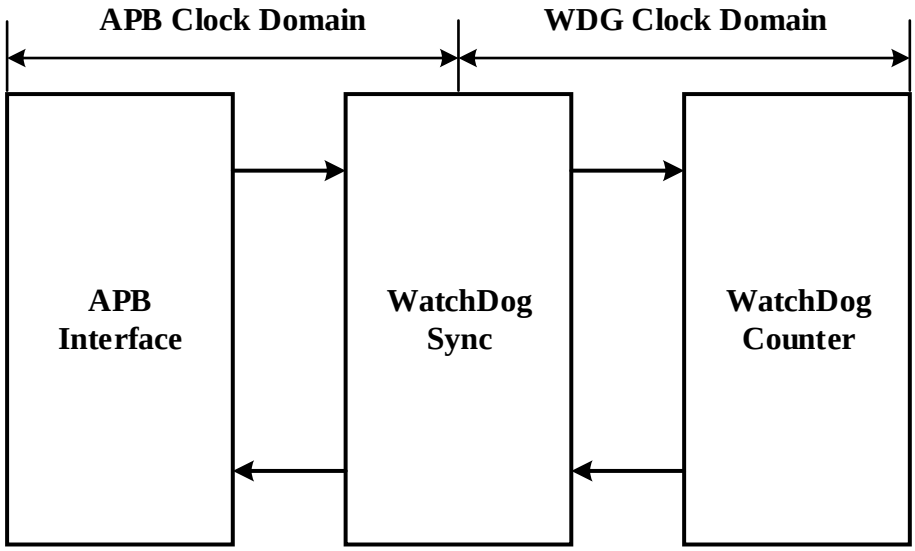
The structure of WatchDog Timer is illustrated in the following diagram, which includes three parts: APB Interface、WatchDog Sync Module and WatchDog Counter.

APB Interface is used to receive the command from Cortex-M0, and generate interrupt and reset signals.

WatchDog Sync Module is used for clock domain crossing between APB clock domain and WDG clock domain.

WatchDog Counter is the counter in WDG clock domain.

In addition, the WatchDog Timer can also be used as a normal timer.



13.2 Features

- 32-bit incremental counter and 32-bit compare register
- Selectable one-stop mode, wrapping mode or free-run mode.
- Interrupt generation.
- Watchdog function

13.3 Pin Description

The Timers have no external Pins.

13.4 Register Description

Base Address 0x4000_8000

Name	Address	Reset	Description
TMR_CNT	0x000	0x0000_0000	Timer count register
TMR_LMT	0x004	0xFFFF_FFFF	Timer compare register
TMR_CTL	0x008	0x0000_0000	Timer control register
TMR_LOCK	0x00C	0x0000_0000	Timer lock register

Timer Counter Register (TMR_CNT)

Bits	Name	Type	Description
31:0	Counter	RO	Timer Counter Register

Timer Compare Register (TMR_LMT)

Bits	Name	Type	Description
31:0	Compare	RW	Timer Compare Register for Upper Limit

Timer Control Register (TMR_CTL)

Bits	Name	Type	Description
0	EN	RW	Timer Enable.
1	LD	RW	Clear Counter. Write 1 to this bit clears TMR_CNT
3:2	OM	RW	Operating Mode : 00 : Continuons wrapping mode 10 : Continuons free-run mode x1 : One-stop mode
4	IE	RW	Interrupt Enable
5	EE	RW	Watchdog Reset Enable
6	IS	RW	Interrupt Status
7	ES	RW	Watchdog Reset Status
8	UM	RW	Watchdog Counter Update Mode
31:9	-	-	Reserved

Timer Lock Register (TMR_LOCK)

Bits	Name	Type	Description
31:0	Lock	RW	Timer Write Lock Register

13.5 Function Description

Timer Operation

After a system reset, bits in the Control Register are reset.

The Timer is enabled when the EN bit in the Control Register is set and the counter starts running. The Control Register and Compare Register must be set to appropriate values first before enabling the Timer. The Counter Register can only be reset by writing 1 to LD bit in the Control Register.

The OM bits in the Control Register defines the counter's operation mode. See table below:

Mode	Description
Wrapping mode	The counter generates an interrupt at a constant interval, reset to 0 after reaching the value in compare register, and continues to count up. This is the default mode.

Free-run mode	The counter reset to0 after reaching its maximum value, i.e. 0xFFFF_FFFF, and continues to count up.
One-stop mode	The counter generates an interrupt once. When the counter reaches the value in compare register, it automatically disabled until you re-enable it.

When the EN bit is cleared the counter value freezes. All other register values are retained.

When the value in counter matches the value in compare register, the interrupt status bit in control register is set and an interrupt request is generated to the system interrupt controller (NVIC) if interrupt status bit in control register is also set. The interrupt status bit is cleared only through writing 1 to this bit, or by system reset.

Watchdog Function

The watchdog function provides a way of recovering from software crashes. WDG have control bits to facilitate watchdog functionality, that is Watchdog Reset Enable (EE) and Watchdog Reset Status (ES) bits in Control Register and a single bit TMR_LOCK register.
When WDG is used to generate a regular interrupt, depending on a programmed value, the watchdog monitors the interrupt and set the ES bit if the interrupt remains no services for the entire programmed period. When ES bit is set, the WDG asserts a reset signal if EE bit is also set.
The Watchdog Reset Status (ES) is only cleared by Hardware reset (POR or RESETN pin), and will not be cleared under other reset condition. This allows boot up program to check the cause of most recent system reset.
Write access to the registers within the WDG can be disabled by the use of the watchdog lock register. Writing a value of 0xDEADFACE to this allows write access to all other registers. Writing any other value disables write access. This feature is included to allow some protection against software which might otherwise disable the watchdog functionality.

13.6 SysTick Counter

- Dedicated timer for fixed time interval SYSTICK (typical 1us interval)

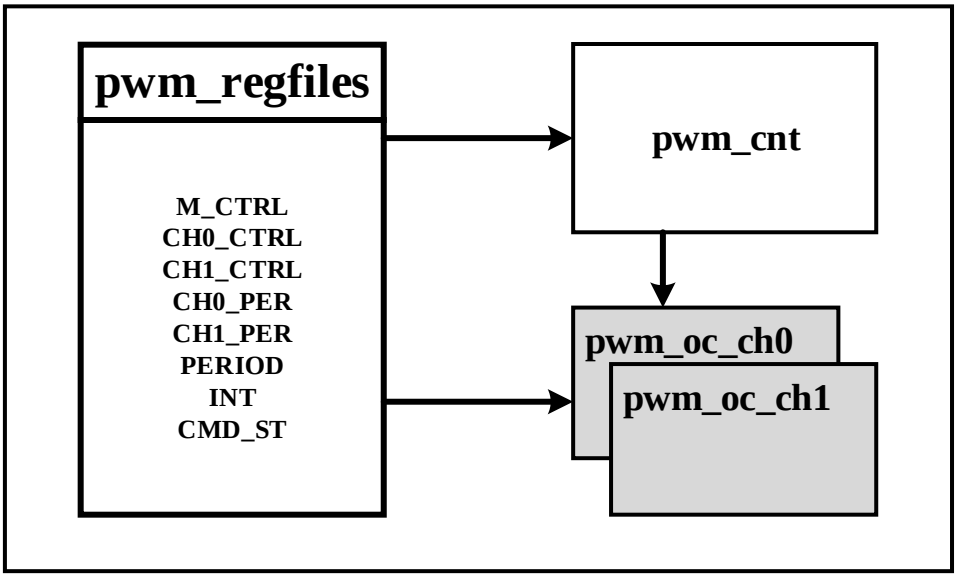
14. DPWM

DPWM are versatile dual channel PWM that can program as complementary PWM output, or program with different phase relationship within DPWM channel, or program as single PWM channel with shared clock. MCU008 have 3 DPWM pair, DPWM0、DPWM1 and DPWM2.
Below table are 3 DPWM clock, reset signals and base address

	Clock	Reset	Base Address
DPWM 0	pwm0_apb_g_clk	pwm0_apb_g_rstn	0x4003_8000
DPWM 1	pwm1_apb_g_clk	pwm1_apb_g_rstn	0x4003_9000
DPWM 2	pwm2_apb_g_clk	pwm2_apb_g_rstn	0x4003_A000

14.1 Overview

Below is dual channels PWM module block diagram, having Register Files、PWM Counter and PWM Out Control sub-modules.
Register Files inside are PWM control registers;
PWM Counter, 2 PWM channels shared one counter;
PWM Out Control, every PWM have their own output control module.



14.2 Registers Description

Base Address

PWM0 0x4003_8000
PWM1 0x4003_9000
PWM2 0x4003_A000
PWM3 0x4003_B000

Name	Address	Reset	Description
M_CTRL	0x00	0x0000_0000	PWM Main Control Register
CH0_CTRL	0x04	0x0000_0000	PWM Channel #0 Control Register
CH1_CTRL	0x08	0x0000_0000	PWM Channel #1 Control Register
CH0_PER	0x0C	0x0000_0000	PWM Channel #0 Period Register
CH1_PER	0x10	0x0000_0000	PWM Channel #1 Period Register

PERIOD	0x14	0x0000_0000	PWM Counter Period Register
INT	0x18	0x0000_0000	PWM Interrupt Register
CMD_STATUS	0x1C	0x0000_0000	PWM Command Status Register

PWM Main Control Register (M_CTRL)

Bits	Name	Type	Description
31:1	Reserved	-	Reserved
0	pwm_en	RW	PWM Counter Enable Register, 0-Disable, 1-Enable

PWM Channel #0 Control Register (CH0_CTRL)

Bits	Name	Type	Description
31:8	Reserved	-	Reserved
7	ch0_oen_n	RW	PWM Output enable of Inactive Period
6	ch0_oen_p	RW	PWM Output enable of Active Period
5	ch0_out_n	RW	PWM Polarity of Inactive Period
4	ch0_out_p	RW	PWM Polarity of Active Period
3:1	Reserved	-	Reserved
0	ch0_oc_en	RW	PWM Channel #0 Enable Register, 0-Disable, 1-Enable

PWM Channel #1 Control Register (CH1_CTRL)

Bits	Name	Type	Description
31:8	Reserved	-	Reserved
7	ch1_oen_n	RW	PWM Output enable of Inactive Period
6	ch1_oen_p	RW	PWM Output enable of Active Period
5	ch1_out_n	RW	PWM Polarity of Inactive Period
4	ch1_out_p	RW	PWM Polarity of Active Period
3:1	Reserved	-	Reserved
0	ch1_oc_en	RW	PWM Channel #1 Enable Register, 0-Disable, 1-Enable

PWM Channel #0 Period Register (CH0_PER)

Bits	Name	Type	Description
31:16	ch0_p_point	RW	PWM Channel #0 Active Point
15:0	ch0_n_point	RW	PWM Channel #0 Inactive Point

PWM Channel #1 Period Register (CH1_PER)

Bits	Name	Type	Description
31:16	ch1_p_point	RW	PWM Channel #1 Active Point
15:0	ch1_n_point	RW	PWM Channel #1 Inactive Point

PWM Counter Period Register (PERIOD)

Bits	Name	Type	Description
31:16	pwm_div	RW	PWM Counter Divider
15:0	pwm_lmt	RW	PWM Counter Limit

PWM Interrupt Register (INT)

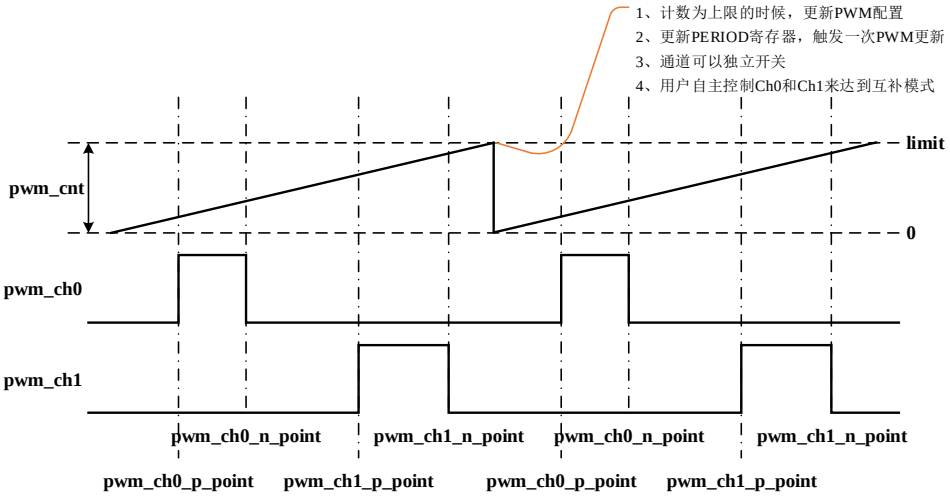
Bits	Name	Type	Description
31:2	Reserved	-	Reserved
1	Int_clear	W1C	PWM Interrupt Clear
0	Int_en	RW	PWM Interrupt Enable

PWM Command Status Register (CMD_ST)

Bits	Name	Type	Description
31:1	Reserved	-	Reserved
0	pwm_status	RO	0-IDLE, 1-BUSY

14.3 Function Description

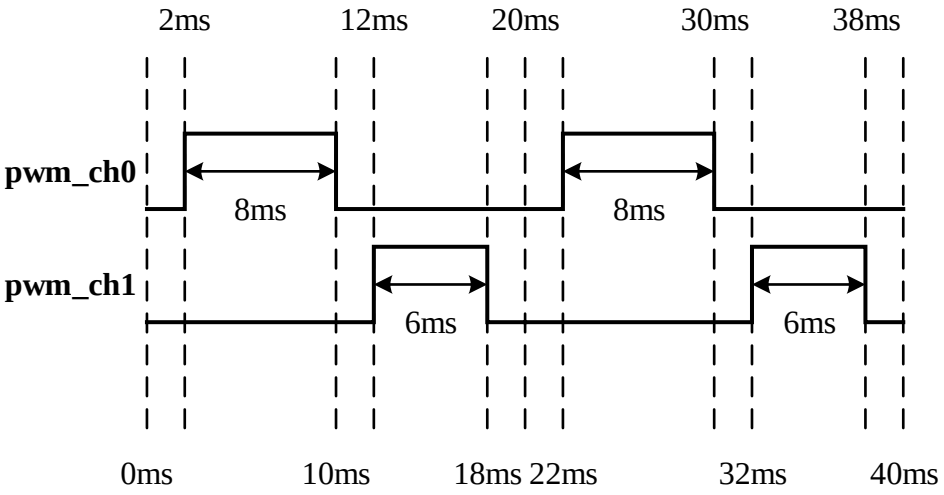
Below timing diagram showing a dual PWM function, each channel has their own control registers set but shared with same clock timer.
They generate PWM signals with same frequency, but can programming in different space make ratio.



14.4 Application Example

14.4.1 Target PWM wave form

Assume system clock is 24MHz，target PWM wave as below shown.



14.4.2 Task analysis

As diagram shown, the 2 PWM channels have same periodically which are 20ms，channel pwm_ch0 output high at 2ms~10ms, pwm_ch1 output high at 12ms~18ms.
This PWM are complementary wave form,
<1> divide 24MHz to generate 500KHz PWM clock.

<2> calculate the counter (pwm_cnt) range, here PWM cycle is 20ms, counter period 2us(500KHz), counter upper count is 9999(20000us/2us - 1).

<3> mapping PWM output is HIGH to counter (pwm_cnt), set value to 1000~5000(1000 = 2000us/2us；5000 = 10000us/2us) for pwm_ch0 (2ms~10ms), set value to 6000~9000(6000 = 12000us/2us；9000 = 18000us/2us) for pwm_ch1 (12ms~18ms).

14.4.3 configuration

<1> **CH0_CTRL** configuration

```
CH0_CTRL = 0xD1;
ch0_oen_n = 1;          // configure inactive (disabled)
ch0_oen_p = 1;          // configure active (enabled)
ch0_out_n = 0;          // configure inactive (output polarity)
ch0_out_p = 1;          // configure active(output polarity)
ch0_oc_en = 1;          // enable channel 0
```

<2> **CH0_PER** configuration

```
CH0_CTRL = (1000 <<16) | 5000;
ch0_p_point = 1000;          // configure active (start point)
ch0_n_point = 5000;          // configure active (end point)
```

<3> **CH1_CTRL** configuration

```
CH1_CTRL = 0xD1;
ch1_oen_n = 1;          // configure inactive
ch1_oen_p = 1;          // configure active
```

```
ch1_out_n = 0;          // configure inactive
ch1_out_p = 1;          // configure active
ch1_oc_en = 1;          // enable channel 1
```

<4> **CH1_PER** configuration

```
CH1_PER = (6000 <<16) | 9000;
ch1_p_point = 6000;     // configure active (start point)
ch1_n_point = 9000;     // configure active (end point)
```

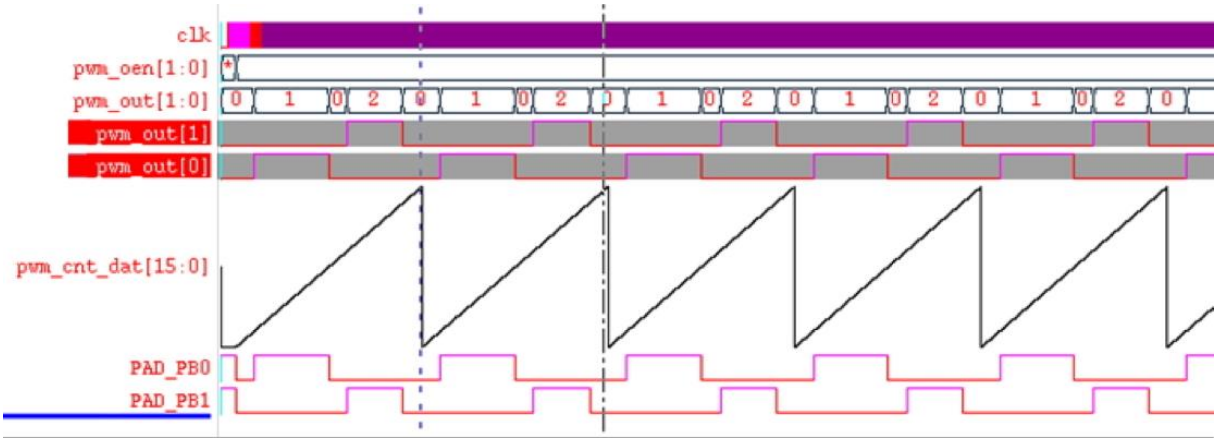
<5> **M_CTRL** configuration

```
CH0_CTRL = 1;
pwm_en = 1;             // enable PWM function
```

<6> **PERIOD** configuration

```
CH1_PER = (31 <<16) | 9999;
pwm_div = 32-1;         // divide by 32
pwm_lmt = 10000-1;      // upper limited (ceiling)
```

14.4.4 Output wave form



15. SPWM

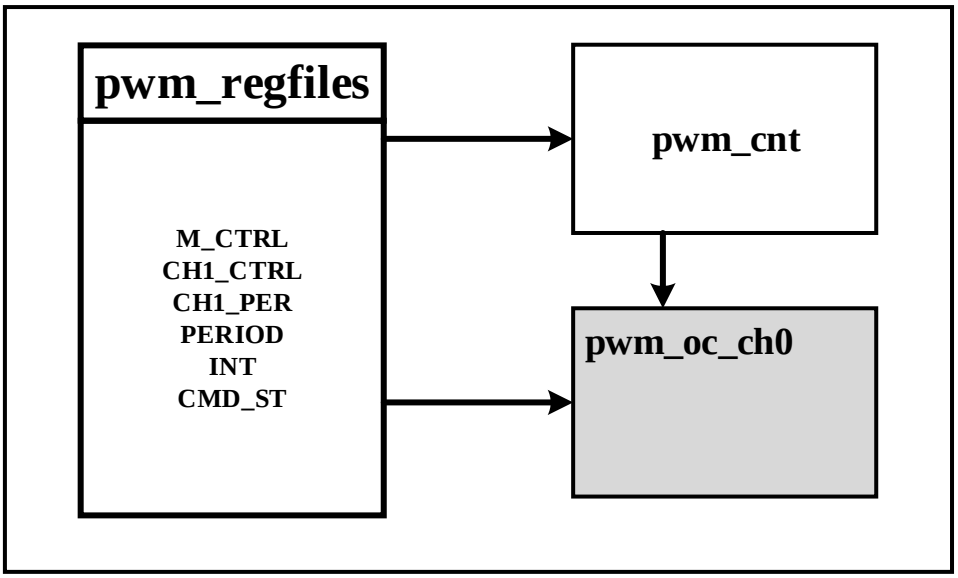
SPWM is a single channel PWM, use with DPWM could form total 7 channels PWM function.

Below table are SPWM clock, reset signals and base address

	Clock	Reset	Base Address
SPWM	pwm3_apb_g_clk	pwm3_apb_g_rstn	0x4003_B000

15.1 Overview

Below is single channel PWM module block diagram, having Register Files、PWM Counter and PWM Out Control sub-modules.
Register Files inside are PWM control registers;
PWM Counter, generate PWM control clock;
PWM Out Control, every PWM have their own output control module.



15.2 Registers Description

Base Address 0x4003_B000

Name	Address	Reset	Description
M_CTRL	0x00	0x0000_0000	PWM Main Control Register
CH1_CTRL	0x08	0x0000_0000	PWM Channel #1 Control Register
CH1_PER	0x10	0x0000_0000	PWM Channel #1 Period Register
PERIOD	0x14	0x0000_0000	PWM Counter Period Register
INT	0x18	0x0000_0000	PWM Interrupt Register
CMD_STATUS	0x1C	0x0000_0000	PWM Command Status Register

PWM Main Control Register (M_CTRL)

Bits	Name	Type	Description
31:1	Reserved	-	Reserved
0	pwm_en	RW	PWM Counter Enable Register, 0-Disable, 1-Enable

PWM Channel #1 Control Register (CH1_CTRL)

Bits	Name	Type	Description
31:8	Reserved	-	Reserved
7	ch1_oen_n	RW	PWM Output enable of Inactive Period
6	ch1_oen_p	RW	PWM Output enable of Active Period
5	ch1_out_n	RW	PWM Polarity of Inactive Period
4	ch1_out_p	RW	PWM Polarity of Active Period
3:1	Reserved	-	Reserved
0	ch1_oc_en	RW	PWM Channel #1 Enable Register, 0-Disable, 1-Enable

PWM Channel #1 Period Register (CH1_PER)

Bits	Name	Type	Description
31:16	ch1_p_point	RW	PWM Channel #1 Active Point
15:0	ch1_n_point	RW	PWM Channel #1 Inactive Point

PWM Counter Period Register (PERIOD)

Bits	Name	Type	Description
31:16	pwm_div	RW	PWM Counter Divider
15:0	pwm_lmt	RW	PWM Counter Limit

PWM Interrupt Register (INT)

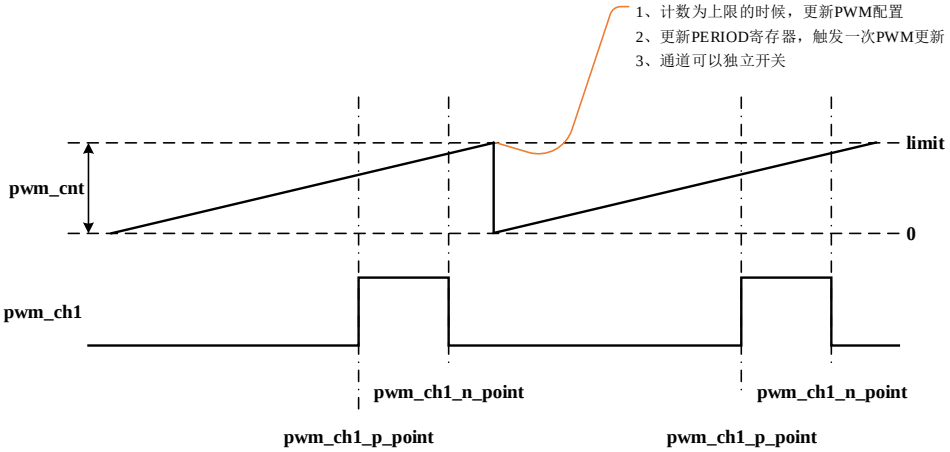
Bits	Name	Type	Description
31:2	Reserved	-	Reserved
1	Int_clear	W1C	PWM Interrupt Clear
0	Int_en	RW	PWM Interrupt Enable

PWM Command Status Register (CMD_ST)

Bits	Name	Type	Description
31:1	Reserved	-	Reserved
0	pwm_status	RO	0-IDLE, 1-BUSY

15.3 Function Description

Below timing diagram showing a typical single channel PWM function.



16. LVD

Below table are SPWM clock, reset signals and base address

	Clock	Reset	Base Address
LVD	lvd_apb_g_clk	lvd_apb_g_rstn	0x4002_8000

16.1 Overview

Below is LVD function diagram, having 5 main units;

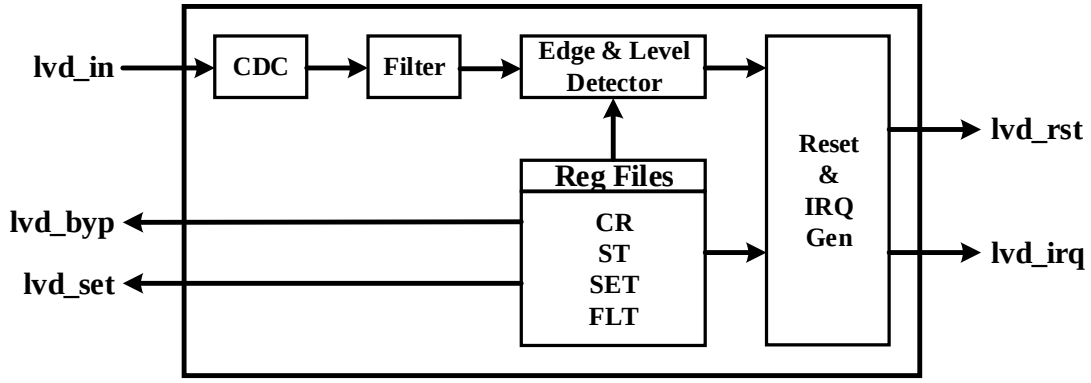
Registers File：control registers;

CDC：process lvd input cross clock domain;

Filter：filter synchronized lvd signal;

Edge & Level Detector：detect edge or level trigger according to register setting;

Reset & IRQ Gen：reset or IRQ generation;



16.2 Registers Description

Base Address 0x4002_8000

Name	Address	Reset	Description
CR	0x00	0x0000_0000	LVD Control Register
ST	0x04	0x0000_0000	LVD Status Register
SET	0x08	0x0000_0000	LVD Set Register
FLT	0x0c	0x0000_0000	LVD Filter Register

LVD Control Register (CR)

Bits	Name	Type	Description
31:5	Reserved	-	Reserved
4	lvd_ne	RW	Low level/Negative trigger
3	lvd_pe	RW	High level/Positive trigger
2	lvd_lv	RW	0-Edge Trigger, 1-Level Trigger
1	lvd_mode	RW	0-Interrupt Mode, 1-Reset Mode
0	lvd_en	RW	LVD enable

LVD Status Register (ST)

Bits	Name	Type	Description
31:1	Reserved	-	Reserved
0	lvd_st	RO	LVD Status

LVD Set Register (SET)

Bits	Name	Type	Description
31:7	Reserved	-	Reserved
6:4	Lvd_set	RW	LVD SET
3:1	Reserved	-	Reserved
0	Lvd_byp	RW	LVD Bypass

LVD SET [6:4]	Trigger Voltage
000	1.80
001	2.00
010	2.40
011	2.60
100	3.00
101	3.60
110	3.90
111	4.20

LVD Filter Register (FLT)

Bits	Name	Type	Description
31:16	flt_lmt	-	LVD filter counter limit
15:1	Reserved	-	Reserved
0	Flt_en	RW	LVD Filter enable

16.3 Function Description

Monitor input or battery voltage, generated Reset or Interrupt if voltage lower trigger level of register setting.

Avoid confusion, only support either LVD IRQ or LVD RESET in setting register.
Lvd_mode = 0x0 ; generate interrupt IRQ
Lvd_mode = 0x1 ; genertae RESET

17. Pin Control

Below table is pin control clock, reset signals and base address

	Clock	Reset	Base Address
Pin Control	pinctl_apb_g_clk	pinctl_apb_g_rstn	0x4004_0000

17.1 Overview

Chip designed with four (4) GPIO port, PA、PB、PC、PD。**

Port PA : 8 GPIO, PA0、PA1、PA2、PA3、PA4、PA5、PA6、PA7 ;

Port PB : 8 GPIO, PB0、PB1、PB2、PB3、PB4、PB5、PB6、PB7 ;

Port PC : 8 GPIO, PC0、PC1、PC2、PC3、PC4、PC5、PC6、PC7 ;

Port PD : 4 GPIO, PD0、PD1、PD2、PD3 ;

Pin Control : versatile setting GPIO functionality 、Driver strength。

**GPIO pin may not be available in all packages

17.2 Registers Description

Base Address 0x4004_0000

Name	Address	Description
PCTL_PA0	0x00	PA0 Pin Control Register
PCTL_PA1	0x04	PA1 Pin Control Register
PCTL_PA2	0x08	PA2 Pin Control Register
PCTL_PA3	0x0C	PA3 Pin Control Register
PCTL_PA4	0x10	PA4 Pin Control Register
PCTL_PA5	0x14	PA5 Pin Control Register
PCTL_PA6	0x18	PA6 Pin Control Register
PCTL_PA7	0x1C	PA7 Pin Control Register
PCTL_PB0	0x20	PB0 Pin Control Register
PCTL_PB1	0x24	PB1 Pin Control Register
PCTL_PB2	0x28	PB2 Pin Control Register
PCTL_PB3	0x2C	PB3 Pin Control Register
PCTL_PB4	0x30	PB4 Pin Control Register
PCTL_PB5	0x34	PB5 Pin Control Register
PCTL_PB6	0x38	PB6 Pin Control Register
PCTL_PB7	0x3C	PB7 Pin Control Register
PCTL_PC0	0x40	PC0 Pin Control Register
PCTL_PC1	0x44	PC1 Pin Control Register
PCTL_PC2	0x48	PC2 Pin Control Register
PCTL_PC3	0x4C	PC3 Pin Control Register
PCTL_PC4	0x50	PC4 Pin Control Register
PCTL_PC5	0x54	PC5 Pin Control Register
PCTL_PC6	0x58	PC6 Pin Control Register
PCTL_PC7	0x5C	PC7 Pin Control Register
PCTL_PD0	0x60	PD0 Pin Control Register
PCTL_PD1	0x64	PD1 Pin Control Register

PCTL_PD2	0x68	PD2 Pin Control Register
PCTL_PD3	0x6C	PD3 Pin Control Register

Pin Control Register (PCTL_PXn) [Remark : X = A/B/C/D，n = 0/1/2/3/4/5/6/7]

Bits	Name	Type	Description
31:16	Reserved	-	Reserved
15:12	FUN	RW	Function select
11:8	Reserved	-	Reserved
7:6	SEL	RW	Output power supply select
5:4	ES	RW	Driver Strength select
3	OD	RW	Open drain select enable, 1-Open Drain, 0-Output Buffer
2	SONOF	RW	Schmitt Trigger, 1-Schmitt Trigger, 0-CMOS Trigger
1	PD	RW	Pull down, valid HIGH
0	PU	RW	Pull up, valid HIGH

Remark: FUN has foolproof protect, PCTL_PXn[15:14] should be 2'b11 when FUN configuration.

Function selection bits:

FUN	Bits 15:12	4'b1100	4'b1101	4'b1110	4'b1111
PAD NAME	Group	Function 0	Function 1	Function 2	Function 3
PA0	PORT A	GIO_0/CTN0	Reserved	AN_15	PWM0
PA1		GIO_1/CTN1	Reserved	AN_14	PWM1
PA2		GIO_2/CTN2	Reserved	AN_13	PWM2
PA3		GIO_3	Reserved	AN_12	PWM3
PA4		GIO_4	SCI_RX_0	AN_11	PWM4
PA5		GIO_5	SCI_TX_0	AN_10	PWM5
PA6		GIO_6	SCI_RX_1	AN_9	PWM6
PA7		GIO_7	SCI_TX_1	AN_8	Reserved
PB0	PORT B	GIO_8	SPI_CS	SCI_RX_0	PWM0
PB1		GIO_9	SPI_CLK	SCI_TX_0	PWM1
PB2		GIO_10/CTN0	SPI_RXD	Reserved	PWM2
PB3		GIO_11/CTN1	SPI_TXD	Reserved	PWM3
PB4		GIO_12/CTN2	I2C_SCL	Reserved	PWM4
PB5		GIO_13/CTN0	I2C_SDA	Reserved	PWM5
PB6		GIO_14/CTN1	SCI_RX_0	Reserved	PWM6
PB7		GIO_15/CTN2	SCI_TX_0	Reserved	Reserved

PC0	PORT C	GIO_16	SCI_RX_1	AN_7	I2C_SCL
PC1		GIO_17	SCI_TX_1	AN_6	I2C_SDA
PC2		GIO_18	SPI_CS	AN_5	PWM0
PC3		GIO_19	SPI_CLK	AN_4	PWM1
PC4		GIO_20	SPI_RXD	AN_3	PWM3
PC5		GIO_21	SPI_TXD	AN_2	PWM4
PC6		GIO_22	I2C_SCL	AN_1	PWM5
PC7		GIO_23	I2C_SDA	AN_0	PWM6
PD0	PORT D	GIO_24	SPI_CS	Reserved	OSC_CLK
PD1		GIO_25/CTN0	SPI_CLK	Reserved	OSC32K_CLK
PD2		GIO_26/CTN1	SPI_RXD	SCI_RX_0	I2C_SCL
PD3		GIO_27/CTN2	SPI_TXD	SCI_TX_0	I2C_SDA

Driver Strength selection:

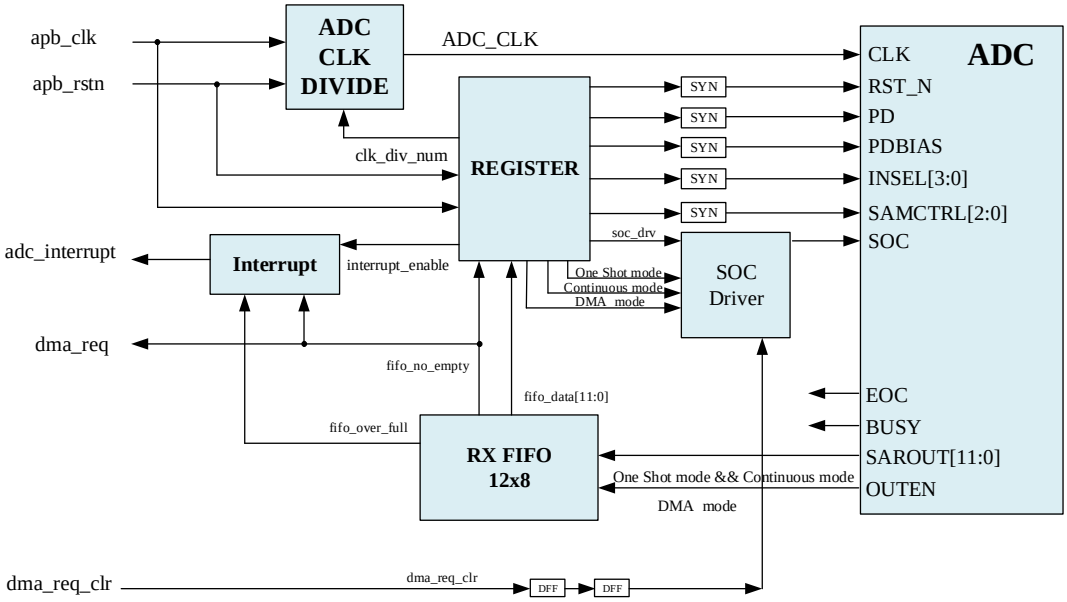
ES1	ES0	driving strength (5.0V)	driving strength (3.3V)	driving strength (2.2V)
0	0	2.3mA	1.3mA	0.5mA
0	1	4.5mA	2.5mA	1.0mA
1	0	8.1mA	4.8mA	1.7mA
1	1	9.0mA	6.1mA	2.3mA

18. ADC

18.1 Feature

- 12 bits +/- 1 LSB @1Msps
- 500Ksps @ 3.3V supply, 1Msps @ 5V supply
- 16 channel (IO pins)
- Internal vRef
- Configuration registers on APB bus, 5 bits address width, 16 bits data
- Support one shot, continuous or DMA mode
- Support ADC data valid and FIFO overflow interrupt

18.2 Block Diagram



18.3 Function

18.3.1 One Shot Mode

One Shot mode is one of ADC support mode, in this mode ADC will only convert once in each time which control by software/firmware. Program can decide when to start the conversion, how may samples. ADC SOC Control Register detect Rising Edge in this mode. (Continuous Mode uses Level Detect).

Typical configuration;

1) Clear *ADC Clock* Gate, reset *ADC Reset* register, set register *ADC PD* low. (*ADC PD* default is power down)
2) Polling the *ADC Reset* register *Ready* bit to wait ADC ready. (after ADC Reset，need about 100us for ADC Analog part start up)
3) Configure *ADC Mode* register to One Shot mode.
4) Configure *ADC Configuration* register *INSEL* and *SAMCTRL*.

5) Configure *ADC SOC* register, High is valid. (in this mode, ADC SOC is control by Register Write signal, recommend write 0 trigger SOC to HIGH)
6) Polling *Data register* to wait data valid or uses interrupt.
7) If need multiple sampling, repeat step 5
8) Set One Shot Mode to 0 to shutdown one shot mode. (note: FIFO will be cleared when Mode set to 0)

18.3.2 Continuous Mode

ADC support continuous mode, in this mode, ADC will continuously convert input signal. Program decide when and sampling timing. Set ADC SOC register 1 to start ADC and 0 to stop ADC.
Typical configuration;

1) Clear *ADC Clock* Gate, reset *ADC Reset*, set register *ADC PD* low. (*ADC PD* default is power down)
2) Polling the *ADC Reset* register *Ready* bit to wait ADC ready. (after ADC Reset，need about 100us for ADC Analog part start up)
3) Configure *ADC Mode* register to Continuous mode。
4) Configure *ADC Configuration* register *INSEL* and *SAMCTRL*.
5) Configure *ADC SOC* register, High is valid. (in this mode, *ADC SOC signal is control by* SOC register value, recommend Write 1 to pull *SOC* High）
6) Polling *Data register* to wait data valid or uses interrupt.
7) Set *SOC* register 0 to end ADC sampling.
9) Set *ADC Mode* Continuous Mode 0 to close ADC function. (note: FIFO will be cleared when Mode set to 0)

18.3.3 DMA Mode

DMA mode is using DMA transfer ADC sampled/converted data. In this mode program can control when to start the sampling. How much sampling of data is set in ADC sample NUM register, ADC will Stop when samples reach the counter value. Setting *SOC* register to 1 will start ADC sampling and convert input data, set SOC to 0 when reach decided number.
Typical configuration;

1) Clear *ADC Clock* Gate, reset *ADC Reset*, set register *ADC PD* low. (*ADC PD* default is power down)
2) Polling the *ADC Reset* register *Ready* bit to wait ADC ready. (after ADC Reset，need about 100us for ADC Analog part start up)
3) Configure *ADC Mode* register to Continuous mode。
4) Configure *ADC Configuration* register *INSEL* and *SAMCTRL*.
5) Configure *ADC sample NUM* register to decide samples data.
6) Configure *ADC Mode* to DMA mode。
7) Configure *SOC* register. Recommend Write 0. (in DMA mode, ADC SOC is control by Register Write signal. Same as One Shot Mode)
8) Polling *Data register* to check data is transferred by.
9) End *ADC DMA mode* (note: FIFO will be cleared when Mode set to 0)

18.3.4 Interrupt

ADC have 2 interrupt sources mux and control by register *ADC interrupt enable*.

- Source 1, Data Valid.

- Source 2, FIFO Overflow.

18.4 Register

18.4.1 Register Summary

BaseAddr 0x4003_0000

Table 4.1 Registers list

ADDRESS	Name	Default	Type	FUNCTION
BaseAddr + 0x00	ADC Reset	0x0	RW	Control ADC Reset, ADC Ready and Data Valid status
BaseAddr + 0x04	ADC Data Register	0x0	RO	ADC data
BaseAddr + 0x08	ADC Clock DIV	0x2	RW	ADC clock configuration
BaseAddr + 0x0c	ADC Mode	0x0	RW	MODE configuration
BaseAddr + 0x10	ADC SOC Control	0x0	RW	SOC configuration
BaseAddr + 0x14	ADC Configuration	0x0	RW	INST SAMCTRL
BaseAddr + 0x18	ADC PD	0x3	RW	ADC Power Down
BaseAddr + 0x1c	ADC sample NUM	0x1	RW	DMA MODE sample number
BaseAddr + 0x20	ADC interrupt enable	0x0	RW	ADC interrupt enable
BaseAddr + 0x24	ADC data valid intr	0x0	RW	ADC Data Valid Interrupt
BaseAddr + 0x28	ADC overflow intr	0x0	RW	ADC Data overflow Interrupt
BaseAddr + 0x2c	ADC busy	0x0	RO	ADC busy
BaseAddr + 0x30	ADC Clock Gate	0x1	RW	ADC Clock Gating

18.4.2 Register Description

1) *ADC reset* (BaseAddr + 0x00)
Control ADC reset register, status of ADC Ready and Data valid.

Bits	Name	Type	Function
31:3	-	-	Reserved
2	data_valid	RO	ADC Data 1：Data Valid 0：Data not Valid
1	adc_startup	RO	ADC startup ready signal. Need 100us to ready after ADC reset. 1：Ready 0：Not Ready

0	adc_reset	RW	ADC Reset 1：Not Reset 0：Reset

2) *ADC Data* (BaseAddr + 0x04)
Check data validation in *ADC reset* register.

Bits	Name	Type	Function
31:12	-	-	Reserved
11:0	data	RO	ADC Data, 12bit, data from FIFO。

3) *ADC Clock Divide* (BaseAddr + 0x08)
ADC Clock divided from APB

Bits	Name	Type	Function
31:6	-	-	Reserved
5:0	clk_div	RW	ADC Clock divider，6 bit。 Default 2, that is 8MHz (Only support APB_CLK=16MHz) 2, 8MHz 4, 4MHz 8, 2MHz 16, 1MHz

4) *ADC mode* (BaseAddr + 0x0C)
Support 3 modes, One shot mode, Continuous Mode and DMA mode。

Bits	Name	Type	Function
31:3	-	-	Reserved
2	dma_mode	RW	DMA mode 1：DMA mode Enable 0：DMA mode Disable
1	Continuous_mode	RW	Continuous mode 1：Continuous mode Enable 0：Continuous mode Disable
0	One_shot_mode	RW	One Shot mode 1：One Shot mode Enable 0：One Shot mode Disable

5) *ADC SOC* Control (BaseAddr + 0x10)
ADC Clock, divide from APB,

Bits	Name	Type	Function
31:1	-	-	Reserved
0	SOC	RW	ADC SOC, high level (1) valid

6) *ADC Configuration* Register (BaseAddr + 0x14)
ADC INSEL and SAMCTRL

Bits	Name	Type	Function
31:7	-	-	Reserved
6:4	SAMCTRL	RW	ADC SAMCTRL, control ADC sample timming
3:0	INSEL	RW	ADC INSEL, select ADC input channel

7) *ADC PD* (BaseAddr + 0x18)
ADC PD and PDBIAS control register

Bits	Name	Type	Function
31:2	-	-	Reserved
1	PDBIAS	RW	ADC PDBIAS, control ADC Bias power down 1:Power Down 0:Power On Default 1, Power Down mode
0	PD	RW	ADC P, control ADC Analog part power down 1:Power Down 0:Power On Default 1, Power Down mode

8) *ADC sample NUM* (BaseAddr + 0x1C)
For DMA mode, ADC sample number

Bits	Name	Type	Function
31:12	-	-	Reserved
11:0	sample number	RW	Only for DMA mode, sample number, 12bit Number of samples for DMA transfer, word, 12 bit valid data. Default is 1, should be greater than 0

9) *ADC interrupt enable* (BaseAddr + 0x20)
ADC interrupt control

Bits	Name	Type	Function
31:2	-	-	Reserved
1	FIFO overflow interrupt enable	RW	ADC FIFO overflow interrupt enable 1：enable 0：disable
0	data valid interrupt enable	RW	ADC data valid interrupt enable 1：enable 0：disable

10) *ADC Data Valid Interrupt* (BaseAddr + 0x24)
ADC data available/interrupt register

Bits	Name	Type	Function
31:1	-	-	Reserved
0	data valid interrupt	RO	Data Valid interrupt status 1：Interrupt Happened 0：Interrupt Not Happened

11) *ADC Overflow Interrupt* (BaseAddr + 0x28)
ADC overflow interrupt register

Bits	Name	Type	Function
31:1	-	-	Reserved
0	overflow interrupt	RO	Overflow interrupt status 1：Interrupt Happened

			0：Interrupt Not Happened

12) ADC busy (BaseAddr + 0x2C)
ADC busy status register

Bits	Name	Type	Function
31:1	-	-	Reserved
0	adc busy	RO	ADC is in conversion 1：ADC is Busy 0：ADC is Not Busy

13) ADC Clock Gate Register (BaseAddr + 0x30)
ADC Clock Gate register

Bits	Name	Type	Function
31:1	-	-	Reserved
0	adc clk gate	RW	ADC Clock Gate function. Default 1, default ADC clock in Gated mode 1：ADC Clock is gated 0：ADC Clock is not gated

19. UART Interface

19.1 Introduction

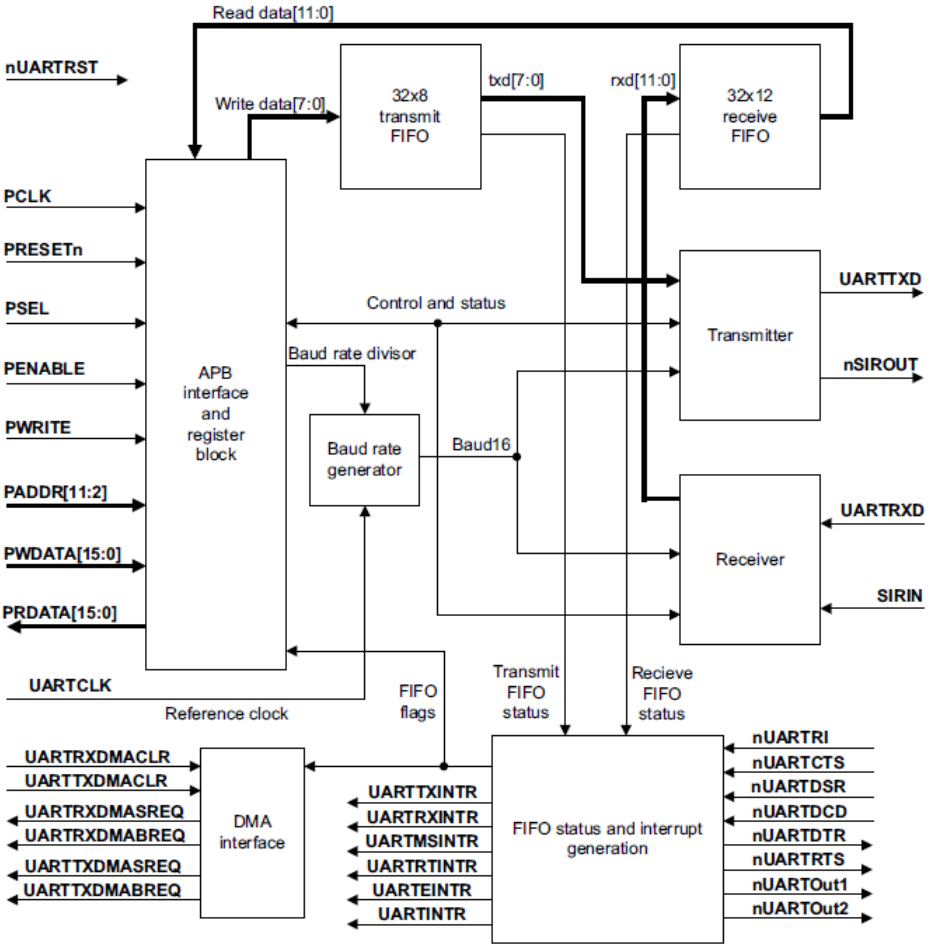
The UART is an AMBA slave module that connects to the Advanced Peripheral Bus (APB).

19.2 Features

- Support 2 UART
- Compliance to the AMBA Specification (Rev 2.0) onwards for easy integration into SoC implementation
- Separate 32×8 transmit and 32×12 receive First-In, First-Out (FIFO) memory buffers to reduce CPU interrupts.
- Programmable baud rate generator.
- Fully-programmable serial interface characteristics
- Interrupt generation with error condition interrupts
- Support DMA

19.3 Functional Description

Block Diagram



The functions of the UART are described in the following sections:

- AMBA APB interface
- Register block

- Baud rate generator
- Transmit FIFO
- Receive FIFO
- Transmit logic
- Receive logic
- Interrupt generation logic
- DMA interface
- Synchronizing registers and logic

19.3.1 AMBA APB interface

The AMBA APB interface generates read and write decodes for accesses to status/control registers, and the transmit and receive FIFOs.

19.3.2 Register block

The register block stores data written, or to be read across the AMBA APB interface.

19.3.3 Baud rate generator

The baud rate generator contains free-running counters that generate the internal ×16 clocks, Baud16 and IrLPBaud16 signals. Baud16 provides timing information for UART transmit and receive control. Baud16 is a stream of pulses with a width of one UARTCLK clock period and a frequency of 16 times the baud rate

19.3.4 Transmit FIFO

The transmit FIFO is an 8-bit wide, 32 location deep, FIFO memory buffer. CPU data written across the APB interface is stored in the FIFO until read out by the transmit logic. You can disable the transmit FIFO to act like a one-byte holding register.

19.3.5 Receive FIFO

The receive FIFO is a 12-bit wide, 32 location deep, FIFO memory buffer. Received data and corresponding error bits, are stored in the receive FIFO by the receive logic until read out by the CPU across the APB interface. The receive FIFO can be disabled to act like a one-byte holding register.

19.3.6 Transmit logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. Control logic outputs the serial bit stream beginning with a start bit, data bits with the Least Significant Bit (LSB) first, followed by the parity bit, and then the stop bits according to the programmed configuration in control registers.

19.3.7 Receive logic

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line break detection are also performed, and their status accompanies the data that is written to the receive FIFO.

19.3.8 Interrupt generation logic

Individual maskable active HIGH interrupts are generated by the UART. A combined interrupt output is also generated as an OR function of the individual interrupt requests.

You can use the single combined interrupt with a system interrupt controller that provides another level of masking on a per-peripheral basis. This enables you to use modular device drivers that always know where to find the interrupt source control register bits.

You can also use the individual interrupt requests with a system interrupt controller that provides masking for the outputs of each peripheral. In this way, a global interrupt service routine can read the entire set of sources from one wide register in the system interrupt controller. This is attractive where the time to read from the peripheral registers is significant compared to the CPU clock speed in a real-time system.

19.3.9 DMA interface

The UART provides an interface to connect to the DMA controller as UART DMA interface.

19.3.10 Synchronizing registers and logic

The UART supports both asynchronous and synchronous operation of the clocks, PCLK and UARTCLK. Synchronization registers and handshaking logic have been implemented, and are active at all times. This has a minimal impact on performance or area. Synchronization of control signals is performed on both directions of data flow, that is from the PCLK to the UARTCLK domain, and from the UARTCLK to the PCLK domain.

- during transmission, the UART data bit is used as the base for encoding
- during reception, the decoded bits are transferred to the UART receive logic.

The low-power divisor value is calculated as:

- Low-power divisor = (FUARTCLK / FIrLPBaud16)

where FIrLPBaud16 is nominally 1.8432MHz.

The divisor must be chosen so that 1.42MHz < FIrLPBaud16 < 2.12MHz.

19.4 Registers

19.4.1 Summary of registers

UART0 Base Address 0x4001_0000

UART1 Base Address 0x4001_1000

Offset	Name	Type	Reset	Width	Description
0x000	UARTDR	RW	0x---	12/8	Data Register, UARTDR
0x004	UARTRSR/ UARTECR	RW	0x0	4/0	Receive Status Register / Error Clear Register, UARTRSR/UARTECR
0x008-0x014	-	-	-	-	Reserved
0x018	UARTFR	RO	0b-10010---	9	Flag Register, UARTFR
0x01C	-	-	-	-	Reserved
0x020	UARTILPR	RW	0x00	8	Low-Power Counter Register, UARTILPR
0x024	UARTIBRD	RW	0x0000	16	Integer Baud Rate Register, UARTIBRD
0x028	UARTFBRD	RW	0x00	6	Fractional Baud Rate Register, UARTFBRD

0x02C	UARTLCR_H	RW	0x00	8	Line Control Register, UARTLCR_H
0x030	UARTCR	RW	0x0300	16	Control Register, UARTCR
0x034	UARTIFLS	RW	0x12	6	Interrupt FIFO Level Select Register, UARTIFLS
0x038	UARTIMSC	RW	0x000	11	Interrupt Mask Set/Clear Register, UARTIMSC
0x03C	UARTRIS	RO	0x00-	11	Raw Interrupt Status Register, UARTRIS
0x040	UARTMIS	RO	0x00-	11	Masked Interrupt Status Register, UARTMIS
0x044	UARTICR	WO	-	11	Interrupt Clear Register, UARTICR
0x048	UARTDMACR	RW	0x00	3	DMA Control Register, UARTDMACR
0x04C-0x07C	-	-	-	-	Reserved
0x080-0x08C	-	-	-	-	Reserved for test purposes
0x090-0xFCC	-	-	-	-	Reserved
0xFD0-0xFDC	-	-	-	-	Reserved for future ID expansion

19.4.2 Data Register, UARTDR

The UARTDR Register is the data register.
For words to be transmitted:

- if the FIFOs are enabled, data written to this location is pushed onto the transmit FIFO
- if the FIFOs are not enabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO).

The write operation initiates transmission from the UART. The data is prefixed with a start bit, appended with the appropriate parity bit (if parity is enabled), and a stop bit. The resultant word is then transmitted.

For received words:

- if the FIFOs are enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO
- if the FIFOs are not enabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO).

The received data byte is read by performing reads from the UARTDR Register along with the corresponding status information. The status information can also be read by a read of the UARTRSR/UARTECR Register.

Bits	Name	Function
15:12	-	Reserved.
11	OE	Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 once there is an empty space in the FIFO and a new character can be written to it.
10	BE	Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state), and the next valid start bit is received.
9	PE	Parity error. When set to 1, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the Line Control Register, UARTLCR_H select. In FIFO mode, this error is associated with the character at the top of the FIFO.
8	FE	Framing error. When set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is associated with the character at the top of the FIFO.

7:0	DATA	Receive (read) data character. Transmit (write) data character.

Note: You must disable the UART before any of the control registers are reprogrammed. When the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.

19.4.3 Receive Status Register / Error Clear Register, UARTRSR/UARTECR

The UARTRSR/UARTECR Register is the receive status register/error clear register. Receive status can also be read from the UARTRSR Register. If the status is read from this register, then the status information for break, framing and parity corresponds to the data character read from the Data Register, UARTDR prior to reading the UARTRSR Register. The status information for overrun is set immediately when an overrun condition occurs.

A write to the UARTECR Register clears the framing, parity, break, and overrun errors. All the bits are cleared to 0 on reset.

Bits	Name	Function
7:0	-	A write to this register clears the framing, parity, break, and overrun errors. The data value is not important.
7:4	-	Reserved, unpredictable when read.
3	OE	Overrun error. This bit is set to 1 if data is received and the FIFO is already full. This bit is cleared to 0 by a write to UARTECR. The FIFO contents remain valid because no more data is written when the FIFO is full, only the contents of the shift register are overwritten. The CPU must now read the data, to empty the FIFO.
2	BE	Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time (defined as start, data, parity, and stop bits). This bit is cleared to 0 after a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received.
1	PE	Parity error. When set to 1, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the Line Control Register, UARTLCR_H select. This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO.
0	FE	Framing error. When set to 1, it indicates that the received character did not have a valid stop bit (a valid stop bit is 1). This bit is cleared to 0 by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO.

Note: The received data character must be read first from the Data Register, UARTDR before reading the error status associated with that data character from the UARTRSR Register. This read sequence cannot be reversed, because the UARTRSR Register is updated only when a read occurs from the UARTDR Register. However, the status information can also be obtained by reading the UARTDR Register.

19.4.4 Flag Register, UARTFR

The UARTFR Register is the flag register. After reset TXFF, RXFF, and BUSY are 0, and TXFE and RXFE are 1.

Bits	Name	Function
15:9	-	Reserved, do not modify, read as zero.

8	RI	Ring indicator. This bit is the complement of the UART ring indicator, nUARTRI, modem status input. That is, the bit is 1 when nUARTRI is LOW.
7	TXFE	Transmit FIFO empty. The meaning of this bit depends on the state of the FEN bit in the Line Control Register, UARTLCR_H. If the FIFO is disabled, this bit is set when the transmit holding register is empty. If the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty. This bit does not indicate if there is data in the transmit shift register.
6	RXFF	Receive FIFO full. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H Register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when the receive FIFO is full.
5	TXFF	Transmit FIFO full. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H Register. If the FIFO is disabled, this bit is set when the transmit holding register is full. If the FIFO is enabled, the TXFF bit is set when the transmit FIFO is full.
4	RXFE	Receive FIFO empty. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H Register. If the FIFO is disabled, this bit is set when the receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.
3	BUSY	UART busy. If this bit is set to 1, the UART is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty, regardless of whether the UART is enabled or not.
2	DCD	Data carrier detect. This bit is the complement of the UART data carrier detect, nUARTDCD, modem status input. That is, the bit is 1 when nUARTDCD is LOW.
1	DSR	Data set ready. This bit is the complement of the UART data set ready, nUARTDSR, modem status input. That is, the bit is 1 when nUARTDSR is LOW.
0	CTS	Clear to send. This bit is the complement of the UART clear to send, nUARTCTS, modem status input. That is, the bit is 1 when nUARTCTS is LOW.

19.4.5 Integer Baud Rate Register, UARTIBRD

The UARTIBRD Register is the integer part of the baud rate divisor value.

Bits	Name	Function
15:0	BAUD DIVINT	The integer baud rate divisor. These bits are cleared to 0 on reset.

19.4.6 Fractional Baud Rate Register, UARTFBRD

The UARTFBRD Register is the fractional part of the baud rate divisor value.

Bits	Name	Function
5:0	BAUD DIVFRAC	The fractional baud rate divisor.. These bits are cleared to 0 on reset.

The baud rate divisor is calculated as follows:

- Baud rate divisor BAUDDIV = (FUARTCLK / (16 × Baud rate))
- where FUARTCLK is the UART reference clock frequency.

The BAUDDIV is comprised of the integer value (BAUD DIVINT) and the fractional value (BAUD DIVFRAC).

Note:

- The contents of the UARTIBRD and UARTFBRD registers are not updated until transmission or reception of the current character is complete.
- The minimum divide ratio possible is 1 and the maximum is 65535(216 - 1). That is, UARTIBRD =

0 is invalid and UARTFBRD is ignored when this is the case.

- Similarly, when UARTIBRD = 65535 (that is 0xFFFF), then UARTFBRD must not be greater than zero. If this is exceeded it results in an aborted transmission or reception.

19.4.7 Line Control Register, UARTLCR_H

The UARTLCR_H Register is the line control register. This register accesses bits 29 to 22 of the UART Line Control Register, UARTLCR.

All the bits are cleared to 0 when reset.

Bits	Name	Function
15:8	-	Reserved, do not modify, read as zero.
7	SPS	Stick parity select. 0 = stick parity is disabled 1 = either: ● if the EPS bit is 0 then the parity bit is transmitted and checked as a 1 ● if the EPS bit is 1 then the parity bit is transmitted and checked as a 0. This bit has no effect when the PEN bit disables parity checking and generation.
6:5	WLEN	Word length. These bits indicate the number of data bits transmitted or received in a frame as follows: b11 = 8 bits b10 = 7 bits b01 = 6 bits b00 = 5 bits.
4	FEN	Enable FIFOs: 0 = FIFOs are disabled (character mode) that is, the FIFOs become 1-byte-deep holding registers 1 = transmit and receive FIFO buffers are enabled (FIFO mode).
3	STP2	Two stop bits select. If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.
2	EPS	Even parity select. Controls the type of parity the UART uses during transmission and reception: 0 = odd parity. The UART generates or checks for an odd number of 1s in the data and parity bits. 1 = even parity. The UART generates or checks for an even number of 1s in the data and parity bits. This bit has no effect when the PEN bit disables parity checking and generation.
1	PEN	Parity enable: 0 = parity is disabled and no parity bit added to the data frame 1= parity checking and generation is enabled.
0	BRK	Send break. If this bit is set to 1, a low-level is continually output on the UARTTXD output, after completing transmission of the current character. For the proper execution of the break command, the software must set this bit for at least two complete frames. For normal use, this bit must be cleared to 0.

The UARTLCR_H, UARTIBRD, and UARTFBRD registers form the single 30-bit wide UARTLCR Register that is updated on a single write strobe generated by a UARTLCR_H write. So, to internally update the contents of UARTIBRD or UARTFBRD, a UARTLCR_H write must always be performed at the end.

Note:

- To update the three registers there are two possible sequences:
  - UARTIBRD write, UARTFBRD write, and UARTLCR_H write
  - UARTFBRD write, UARTIBRD write, and UARTLCR_H write.
- To update UARTIBRD or UARTFBRD only:
  - UARTIBRD write, or UARTFBRD write, and UARTLCR_H write.

Table is a truth table for the Stick Parity Select (SPS), Even Parity Select (EPS), and Parity ENable (PEN) bits of the Line Control Register, UARTLCR_H.

PEN	EPS	SPS	Parity bit (transmitted or checked)
0	x	x	Not transmitted or checked
1	1	0	Even parity
1	0	0	Odd parity
1	0	1	1
1	1	1	0

Note

- The UARTLCR_H, UARTIBRD, and UARTFBRD registers must not be changed:
  - when the UART is enabled
  - when completing a transmission or a reception when it has been programmed to become disabled.
- The FIFO integrity is not guaranteed under the following conditions:
  - after the BRK bit has been initiated
  - if the software disables the UART in the middle of a transmission with data in the FIFO, and then re-enables it.

19.4.8 Control Register, UARTCR

The UARTCR Register is the control register. All the bits are cleared to 0 on reset except for bits 9 and 8 that are set to 1.

Bits	Name	Function
15	CTSEn	CTS hardware flow control enable. If this bit is set to 1, CTS hardware flow control is enabled. Data is only transmitted when the nUARTCTS signal is asserted.
14	RTSEn	RTS hardware flow control enable. If this bit is set to 1, RTS hardware flow control is enabled. Data is only requested when there is space in the receive FIFO for it to be received.
13	Out2	This bit is the complement of the UART Out2 (nUARTOut2) modem status output. That is, when the bit is programmed to a 1, the output is 0. For DTE this can be used as Ring Indicator (RI).
12	Out1	This bit is the complement of the UART Out1 (nUARTOut1) modem status output. That is, when the bit is programmed to a 1 the output is 0. For DTE this can be used as Data Carrier Detect (DCD).
11	RTS	Request to send. This bit is the complement of the UART request to send, nUARTRTS, modem status output. That is, when the bit is programmed to a 1 then nUARTRTS is LOW.
10	DTR	Data transmit ready. This bit is the complement of the UART data transmit ready, nUARTDTR, modem status output. That is, when the bit is programmed to a 1 then nUARTDTR is LOW.
9	RXE	Receive enable. If this bit is set to 1, the receive section of the UART is enabled. Data reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit. When the UART is disabled in the middle of reception, it completes the current character before stopping.
8	TXE	Transmit enable. If this bit is set to 1, the transmit section of the UART is enabled. Data transmission occurs for either UART signals, or SIR signals depending on the setting of the SIREN bit. When the UART is disabled in the middle of transmission, it completes the current character before stopping.
7	LBE	Loopback enable. If this bit is set to 1 and the SIREN bit is set to 1 and the SIRTEST bit in the Test Control Register, UARTTCR is set to 1, then the nSIROUT path is inverted, and fed through to the SIRIN path. The SIRTEST bit in the test register must be set to 1 to override the normal half-duplex SIR operation. This must be the requirement for accessing the test registers during normal operation, and SIRTEST must be cleared to 0 when loopback testing is finished. This feature reduces the amount of external coupling required during system test. If this bit is set to 1, and the SIRTEST bit is set to 0, the UARTTXD path is fed through to the UARTRXD path. In either SIR mode or UART mode, when this bit is set, the modem outputs are also fed through to the modem inputs.

		This bit is cleared to 0 on reset, to disable loopback.
6:3	-	Reserved, do not modify, read as zero.
2	SIRLP	Reserved
1	SIREN	Reserved
0	UARTEN	UART enable: 0 = UART is disabled. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. 1 = the UART is enabled. Data transmission and reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit.

Note: To enable transmission, the TXE bit and UARTEN bit must be set to 1. Similarly, to enable reception, the RXE bit and UARTEN bit, must be set to 1.

Note: Program the control registers as follows:
1. Disable the UART.
2. Wait for the end of transmission or reception of the current character.
3. Flush the transmit FIFO by setting the FEN bit to 0 in the Line Control Register, UARTLCR_H.
4. Reprogram the UARTCR Register.
5. Enable the UART.

**MCU008 are NOT support hardware flow control.

19.4.9 Interrupt FIFO Level Select Register, UARTIFLS

The UARTIFLS Register is the interrupt FIFO level select register. You can use this register to define the FIFO level that triggers the assertion of UARTTXINTR and UARTRXINTR.

The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level.

The bits are reset so that the trigger level is when the FIFOs are at the half-way mark.

Bits	Name	Function
15:6	-	Reserved, do not modify, read as zero.
5:3	RXIFLSEL	Receive interrupt FIFO level select. The trigger points for the receive interrupt are as follows: b000 = Receive FIFO becomes ≥ 1/8 full b001 = Receive FIFO becomes ≥ 1/4 full b010 = Receive FIFO becomes ≥ 1/2 full b011 = Receive FIFO becomes ≥ 3/4 full b100 = Receive FIFO becomes ≥ 7/8 full b101-b111 = reserved.
2:0	TXIFLSEL	Transmit interrupt FIFO level select. The trigger points for the transmit interrupt are as follows: b000 = Transmit FIFO becomes ≤ 1/8 full b001 = Transmit FIFO becomes ≤ 1/4 full b010 = Transmit FIFO becomes ≤ 1/2 full b011 = Transmit FIFO becomes ≤ 3/4 full b100 = Transmit FIFO becomes ≤ 7/8 full b101-b111 = reserved.

19.4.10 Interrupt Mask Set/Clear Register, UARTIMSC

The UARTIMSC Register is the interrupt mask set/clear register. It is a read/write register.

On a read this register returns the current value of the mask on the relevant interrupt. On a write of 1 to the particular bit, it sets the corresponding mask of that interrupt. A write of 0 clears the corresponding mask.

Bits	Name	Function
15:11	-	Reserved, read as zero, do not modify.
10	OEIM	Overrun error interrupt mask. A read returns the current mask for the UARTOEINTR interrupt. On a write of 1, the mask of the UARTOEINTR interrupt is set. A write of 0 clears the mask.
9	BEIM	Break error interrupt mask. A read returns the current mask for the UARTBEINTR interrupt. On a write of 1, the mask of the UARTBEINTR interrupt is set. A write of 0 clears the mask.
8	PEIM	Parity error interrupt mask. A read returns the current mask for the UARTPEINTR interrupt. On a write of 1, the mask of the UARTPEINTR interrupt is set. A write of 0 clears the mask.
7	FEIM	Framing error interrupt mask. A read returns the current mask for the UARTFEINTR interrupt. On a write of 1, the mask of the UARTFEINTR interrupt is set. A write of 0 clears the mask.
6	RTIM	Receive timeout interrupt mask. A read returns the current mask for the UARTRTINTR interrupt. On a write of 1, the mask of the UARTRTINTR interrupt is set. A write of 0 clears the mask.
5	TXIM	Transmit interrupt mask. A read returns the current mask for the UARTTXINTR interrupt. On a write of 1, the mask of the UARTTXINTR interrupt is set. A write of 0 clears the mask.
4	RXIM	Receive interrupt mask. A read returns the current mask for the UARTRXINTR interrupt. On a write of 1, the mask of the UARTRXINTR interrupt is set. A write of 0 clears the mask.
3	DSRMIM	nUARTDSR modem interrupt mask. A read returns the current mask for the UARTDSRINTR interrupt. On a write of 1, the mask of the UARTDSRINTR interrupt is set. A write of 0 clears the mask.
2	DCDMIM	nUARTDCD modem interrupt mask. A read returns the current mask for the UARTDCDINTR interrupt. On a write of 1, the mask of the UARTDCDINTR interrupt is set. A write of 0 clears the mask.
1	CTSMIM	nUARTCTS modem interrupt mask. A read returns the current mask for the UARTCTSINTR interrupt. On a write of 1, the mask of the UARTCTSINTR interrupt is set. A write of 0 clears the mask.
0	RIMIM	nUARTRI modem interrupt mask. A read returns the current mask for the UARTRIINTR interrupt. On a write of 1, the mask of the UARTRIINTR interrupt is set. A write of 0 clears the mask.

All the bits are cleared to 0 when reset.

19.4.11 Raw Interrupt Status Register, UARTRIS

The UARTRIS Register is the raw interrupt status register. It is a read-only register. This register returns the current raw status value, prior to masking, of the corresponding interrupt. A write has no effect.

Caution: All the bits, except for the modem status interrupt bits (bits 3 to 0), are cleared to 0 when reset. The modem status interrupt bits are undefined after reset.

Bits	Name	Function
15:11	-	Reserved, read as zero, do not modify.

10	OERIS	Overrun error interrupt status. Returns the raw interrupt state of the UARTOEINTR interrupt.
9	BERIS	Break error interrupt status. Returns the raw interrupt state of the UARTBEINTR interrupt.
8	PERIS	Parity error interrupt status. Returns the raw interrupt state of the UARTPEINTR interrupt.
7	FERIS	Framing error interrupt status. Returns the raw interrupt state of the UARTFEINTR interrupt.
6	RTRIS	Receive timeout interrupt status. Returns the raw interrupt state of the UARTRTINTR interrupt.
5	TXRIS	Transmit interrupt status. Returns the raw interrupt state of the UARTTXINTR interrupt.
4	RXRIS	Receive interrupt status. Returns the raw interrupt state of the UARTRXINTR interrupt.
3	DSRRMIS	nUARTDSR modem interrupt status. Returns the raw interrupt state of the UARTDSRINTR interrupt.
2	DCDRMIS	nUARTDCD modem interrupt status. Returns the raw interrupt state of the UARTDCDINTR interrupt.
1	CTSRMIS	nUARTCTS modem interrupt status. Returns the raw interrupt state of the UARTCTSINTR interrupt.
0	RIRMIS	nUARTRI modem interrupt status. Returns the raw interrupt state of the UARTRIINTR interrupt.

19.4.12 Masked Interrupt Status Register, UARTMIS

The UARTMIS Register is the masked interrupt status register. It is a read-only register. This register returns the current masked status value of the corresponding interrupt. A write has no effect.

All the bits except for the modem status interrupt bits (bits 3 to 0) are cleared to 0 when reset. The modem status interrupt bits are undefined after reset.

Bits	Name	Function
15:11	-	Reserved, read as zero, do not modify
10	OEMIS	Overrun error masked interrupt status. Returns the masked interrupt state of the UARTOEINTR interrupt.
9	BEMIS	Break error masked interrupt status. Returns the masked interrupt state of the UARTBEINTR interrupt.
8	PEMIS	Parity error masked interrupt status. Returns the masked interrupt state of the UARTPEINTR interrupt.
7	FEMIS	Framing error masked interrupt status. Returns the masked interrupt state of the UARTFEINTR interrupt.
6	RTMIS	Receive timeout masked interrupt status. Returns the masked interrupt state of the UARTRTINTR interrupt.
5	TXMIS	Transmit masked interrupt status. Returns the masked interrupt state of the UARTTXINTR interrupt.
4	RXMIS	Receive masked interrupt status. Returns the masked interrupt state of the UARTRXINTR interrupt.
3	DSRMMIS	nUARTDSR modem masked interrupt status. Returns the masked interrupt state of the UARTDSRINTR interrupt.
2	DCDMMIS	nUARTDCD modem masked interrupt status. Returns the masked interrupt state of the UARTDCDINTR interrupt.
1	CTSMMIS	nUARTCTS modem masked interrupt status. Returns the masked interrupt state of the UARTCTSINTR interrupt.
0	RIMMIS	nUARTRI modem masked interrupt status. Returns the masked interrupt state of the UARTRIINTR interrupt.

19.4.13 Interrupt Clear Register, UARTICR

The UARTICR Register is the interrupt clear register and is write-only. On a write of 1, the corresponding interrupt is cleared. A write of 0 has no effect.

Bits	Name	Function
15:11	Reserved	Reserved, read as zero, do not modify.
10	OEIC	Overrun error interrupt clear. Clears the UARTOEINTR interrupt.
9	BEIC	Break error interrupt clear. Clears the UARTBEINTR interrupt.
8	PEIC	Parity error interrupt clear. Clears the UARTPEINTR interrupt.
7	FEIC	Framing error interrupt clear. Clears the UARTFEINTR interrupt.
6	RTIC	Receive timeout interrupt clear. Clears the UARTRTINTR interrupt.
5	TXIC	Transmit interrupt clear. Clears the UARTTXINTR interrupt.
4	RXIC	Receive interrupt clear. Clears the UARTRXINTR interrupt.
3	DSRMIC	nUARTDSR modem interrupt clear. Clears the UARTDSRINTR interrupt.
2	DCDMIC	nUARTDCD modem interrupt clear. Clears the UARTDCDINTR interrupt.
1	CTSMIC	nUARTCTS modem interrupt clear. Clears the UARTCTSINTR interrupt.
0	RIMIC	nUARTRI modem interrupt clear. Clears the UARTRIINTR interrupt.

19.4.14 DMA Control Register, UARTDMACR

The UARTDMACR Register is the DMA control register. It is a read/write register. All the bits are cleared to 0 on reset.

Bits	Name	Function
15:3	-	Reserved, read as zero, do not modify.
2	DMAONERR	DMA on error. If this bit is set to 1, the DMA receive request outputs, UARTRXDMASREQ or UARTRXDMABREQ, are disabled when the UART error interrupt is asserted.
1	TXDMAE	Transmit DMA enable. If this bit is set to 1, DMA for the transmit FIFO is enabled.
0	RXDMAE	Receive DMA enable. If this bit is set to 1, DMA for the receive FIFO is enabled.

20. SPI

20.1 Introduction

The SPI is a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:

- a SPI-compatible interface
- a Synchronous Serial interface
- a Microwire interface.

In both master and slave configurations, the SPI performs:

- parallel-to-serial conversion on data written to an internal 16-bit wide, 8-location deep transmit FIFO
- serial-to-parallel conversion on received data, buffering it in a similar 16-bit wide, 8-location deep receive FIFO.

Interrupts are generated to:

- request servicing of the transmit and receive FIFO
- inform the system that a receive FIFO over-run has occurred
- inform the system that data is present in the receive FIFO after an idle period has expired.

20.2 Features of the SPI

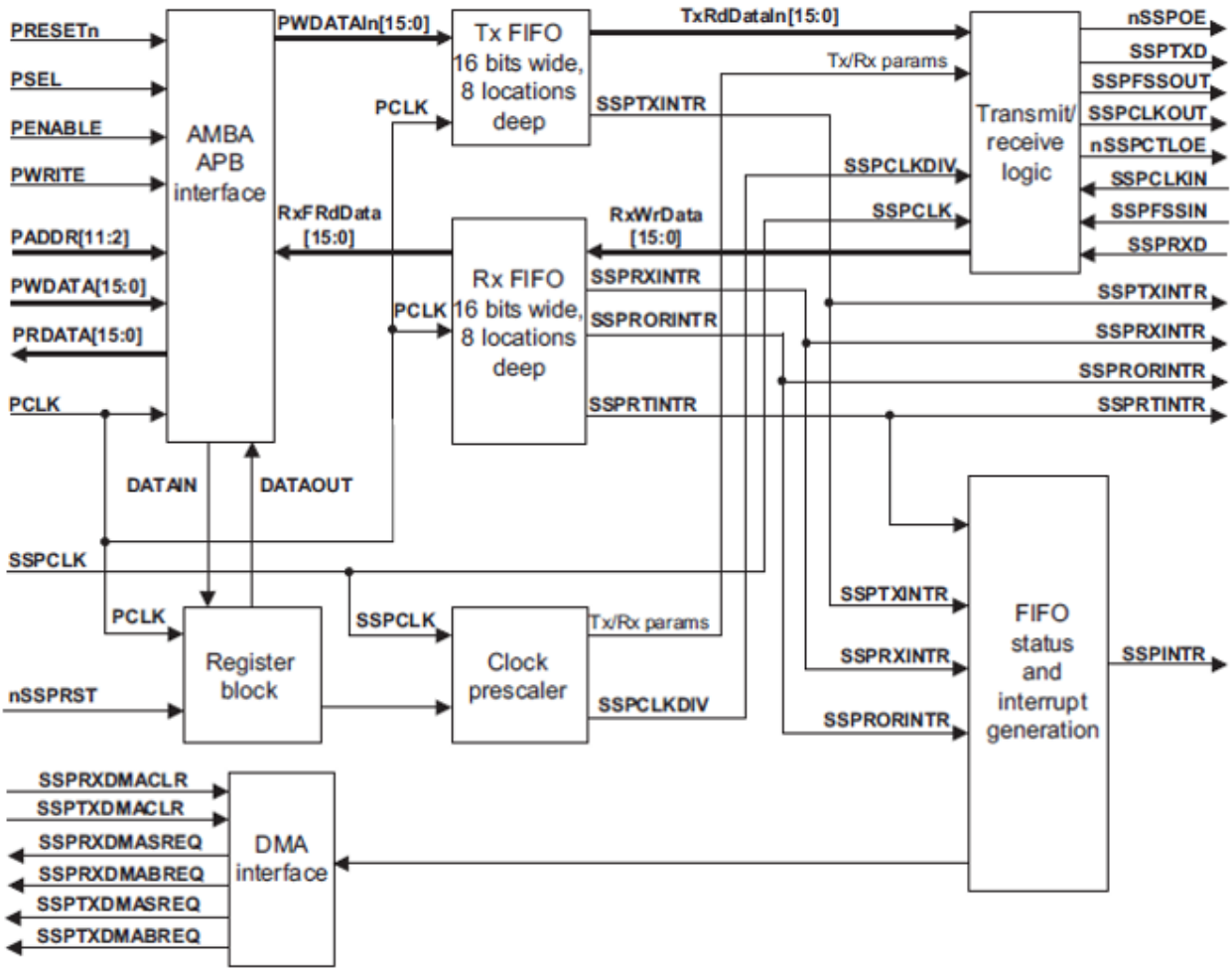
The SPI has the following features:

- Compliance to the AMBA Specification (Rev 2.0) for easy integration into System-on-Chip (SoC) implementation.
- Master or slave operation.
- Programmable clock bit rate and pre-scale.
- Separate transmit and receive first-in, first-out memory buffers, 16 bits wide, 8 locations deep.
- Programmable choice of interface operation, SPI, Microwire, or Synchronous Serial.
- Programmable data frame size from 4 to 16 bits.
- Independent masking of transmit FIFO, receive FIFO, and receive overrun interrupts.
- Internal loopback test mode available.
- Support for Direct Memory Access (DMA).
- Identification registers that uniquely identify the SPI. These can be used by an operating system to automatically configure itself.

20.3 Functional

The functions of the SPI are described in the following sections:

- AMBA APB interface
- Register block
- Clock pre-scaler
- Transmit FIFO
- Receive FIFO
- Transmit and receive logic
- Interrupt generation logic
- Synchronizing registers and logic
- DMA interface



SPI Block Diagram

20.4 Operation

The operation of the SPI is described in the following sections:

- Interface reset
- Configuring the SPI
- Enable SPI operation
- Clock ratios
- Programming the SSPCR0 Control Register
- Programming the SSPCR1 Control Register
- Frame format
- Synchronous serial frame format
- SPI frame format
- Microwire frame format
- DMA interface

20.4.1 Interface reset

The SPI is reset by the global reset signal PRESETn and a block-specific reset signal nSSPRST. An external reset controller must use PRESETn to assert nSSPRST asynchronously and negate it synchronously to SSPCLK. PRESETn must be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then taken HIGH again. The SPI requires PRESETn to be asserted LOW for at least one period of PCLK.

20.4.2 Configuring the SPI

Following reset, the SPI logic is disabled and must be configured when in this state.

Control registers SSPCR0 and SSPCR1 need to be programmed to configure the peripheral as a master or slave operating under one of the following protocols:

- SPI
- SSI
- Microwire

The bit rate, derived from the external SSPCLK, requires the programming of the clock pre-scale register SSPCPSR.

20.4.3 Enable SPI operation

You can either prime the transmit FIFO, by writing up to eight 16-bit values when the SPI is disabled, or allow the transmit FIFO service request to interrupt the CPU. Once enabled, transmission or reception of data begins on the transmit (SSPTXD) and receive (SSPRXD) pins.

20.4.4 Clock ratios

There is a constraint on the ratio of the frequencies of PCLK to SSPCLK. The frequency of SSPCLK must be less than or equal to that of PCLK. This ensures that control signals from the SSPCLK domain to the PCLK domain are certain to get synchronized before one frame duration:

FSSPCLK <= FPCLK.

In the slave mode of operation, the SSPCLKIN signal from the external master is double synchronized and then delayed to detect an edge. It takes three SSPCLKs to detect an edge on SSPCLKIN. SSPTXD has less setup time to the falling edge of SSPCLKIN on which the master is sampling the line. The setup and hold times on SSPRXD with reference to SSPCLKIN must be more conservative to ensure that it is at the right value when the actual sampling occurs within the SSPMS. To ensure correct device operation, SSPCLK must be at least 12 times faster than the maximum expected frequency of SSPCLKIN.

The frequency selected for SSPCLK must accommodate the desired range of bit clock rates. The ratio of minimum SSPCLK frequency to SSPCLKOUT maximum frequency in the case of the slave mode is 12 and for the master mode it is two.

To generate a maximum bit rate of 1.8432Mbps in the Master mode, the frequency of SSPCLK must be at least 3.6864MHz. With an SSPCLK frequency of 3.6864MHz, the SSPCPSR register has to be programmed with a value of two and the SCR[7:0] field in the SSPCR0 register needs to be programmed as zero.

To work with a maximum bit rate of 1.8432Mbps in the slave mode, the frequency of SSPCLK must be at least 22.12MHz. With an SSPCLK frequency of 22.12MHz, the SSPCPSR register can be programmed with a value of 12 and the SCR[7:0] field in the SSPCR0 register can be programmed as zero. Similarly the ratio of SSPCLK maximum frequency to SSPCLKOUT minimum frequency is 254 x 256.

The minimum frequency of SSPCLK is governed by the following equations, both of which have to be satisfied:

FSSPCLK(min) => 2 x FSSPCLKOUT(max) [for master mode]
FSSPCLK(min) => 12 x FSSPCLKIN(max) [for slave mode].

The maximum frequency of SSPCLK is governed by the following equations, both of which have to be satisfied:

FSSPCLK(max) <= 254 x 256 x FSSPCLKOUT(min) [for master mode]
FSSPCLK(max) <= 254 x 256 x FSSPCLKIN(min) [for slave mode].

20.4.5 Programming the SSPCR0 Control Register

The SSPCR0 register is used to:

- program the serial clock rate
- select one of the three protocols
- select the data word size (where applicable).

The Serial Clock Rate (SCR) value, in conjunction with the SSPCPSR clock pre-scale divisor value (CPSDVSR), is used to derive the SPI transmit and receive bit rate from the external SSPCLK.

The frame format is programmed through the FRF bits and the data word size through the DSS bits.

Bit phase and polarity, applicable to SPI format only, are programmed through the SPH and SPO bits.

20.4.6 Programming the SSPCR1 Control Register

The SSPCR1 register is used to:

- select master or slave mode
- enable a loop back test feature
- enable the SPI peripheral.

To configure the SPI as a master, clear the SSPCR1 register master or slave selection bit (MS) to 0, which is the default value on reset.

Setting the SSPCR1 register MS bit to 1 configures the SPI as a slave. When configured as a slave, enabling or disabling of the SPI SSPTXD signal is provided through the SSPCR1 slave mode SSPTXD output disable bit (SOD). This can be used in some multi-slave environments where masters might parallel broadcast.

To enable the operation of the SPI set the Synchronous Serial Port Enable (SSE) bit to 1.

Bit rate generation

The serial bit rate is derived by dividing down the input clock SSPCLK. The clock is first divided by an even pre-scale value CPSDVSR from 2 to 254, which is programmed in SSPCPSR. The clock is further divided by a value from 1 to 256, which is 1 + SCR, where SCR is the value programmed in SSPCR0.

The frequency of the output signal bit clock SSPCLKOUT is defined below:

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$$

For example, if SSPCLK is 3.6864MHz, and CPSDVSR = 2, then SSPCLKOUT has a frequency range from 7.2kHz to 1.8432MHz.

20.4.7 Frame format

Each data frame is between 4 and 16 bits long depending on the size of data programmed, and is transmitted starting with the MSB. There are three basic frame types that can be selected:

- Synchronous serial
- SPI
- Microwire.

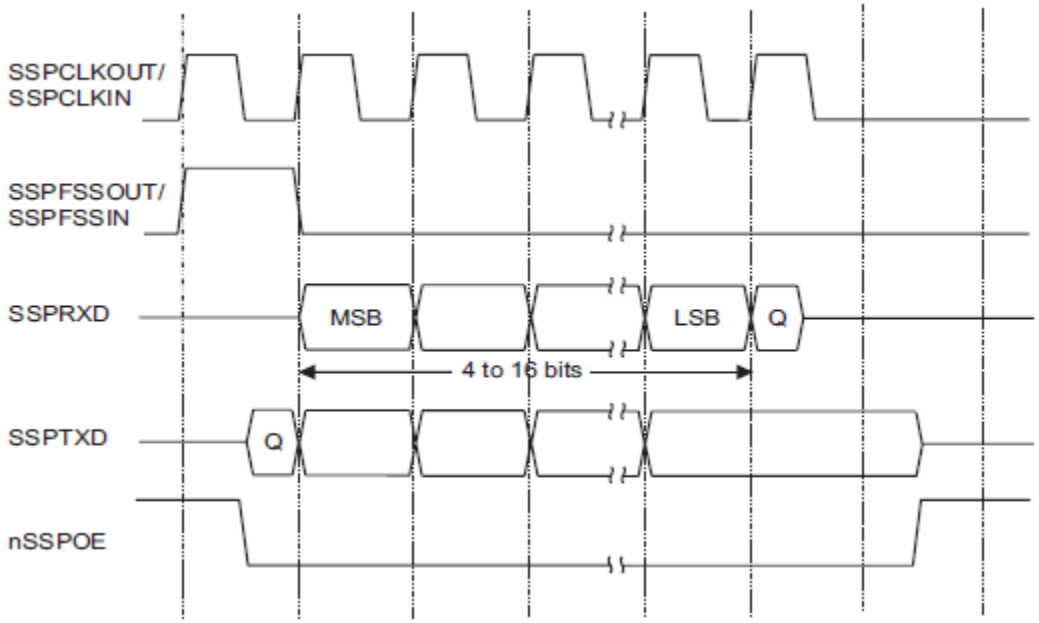
For all three formats, the serial clock (SSPCLKOUT) is held inactive while the SPI is idle, and transitions at the programmed frequency only during active transmission or reception of data. The idle state of SSPCLKOUT is utilized to provide a receive timeout indication that occurs when the receive FIFO still contains data after a timeout period.

For SPI and Microwire frame formats, the serial frame (SSPFSSOUT) pin is active LOW, and is asserted (pulled down) during the entire transmission of the frame.

For Synchronous serial frame format, the SSPFSSOUT pin is pulsed for one serial clock period starting at its rising edge, prior to the transmission of each frame. For this frame format, both the SPI and the off-chip slave device drive their output data on the rising edge of SSPCLKOUT, and latch data from the other device on the falling edge.

Unlike the full-duplex transmission of the other two frame formats, the Microwire format uses a special master-slave messaging technique, which operates at half-duplex. In this mode, when a frame begins, an 8-bit control message is transmitted to the off-chip slave. During this transmit, no incoming data is received by the SPI. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the requested data. The returned data can be 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

20.4.8 Synchronous serial frame format

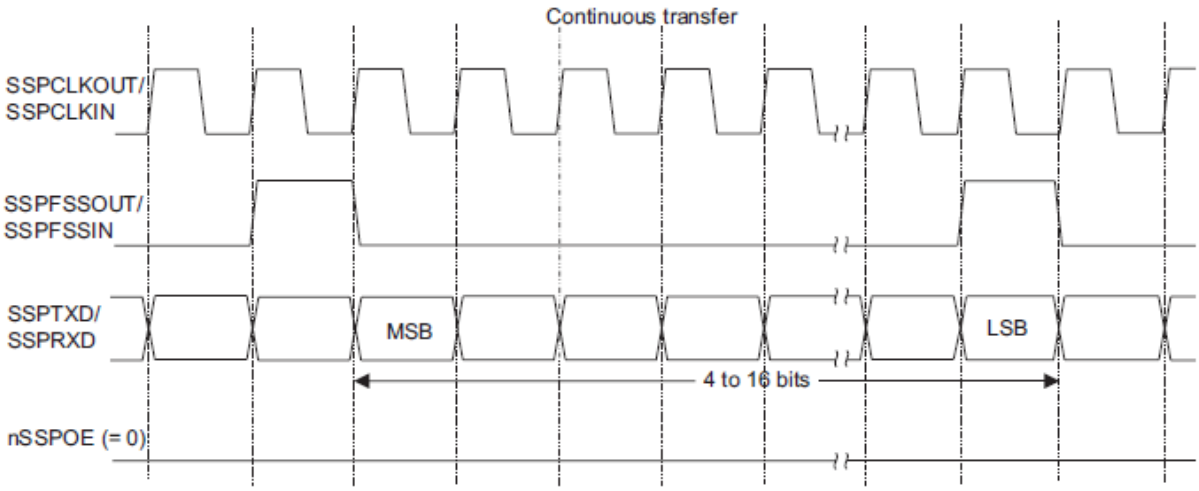


synchronous serial frame format (single transfer)

In this mode, SSPCLKOUT and SSPFSSOUT are forced LOW, and the transmit data line SSPTXD is tristated whenever the SPI is idle. Once the bottom entry of the transmit FIFO contains data, SSPFSSOUT is pulsed HIGH for one SSPCLKOUT period. The value to be transmitted is also

transferred from the transmit FIFO to the serial shift register of the transmit logic. On the next rising edge of SSPCLKOUT, the MSB of the 4 to 16-bit data frame is shifted out on the SSPTXD pin. Likewise, the MSB of the received data is shifted onto the SSPRXD pin by the off-chip serial slave device.

Both the SPI and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each SSPCLKOUT. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SSPCLKOUT after the LSB has been latched.



synchronous serial frame format (continues transfer)

20.4.9 SPI frame format

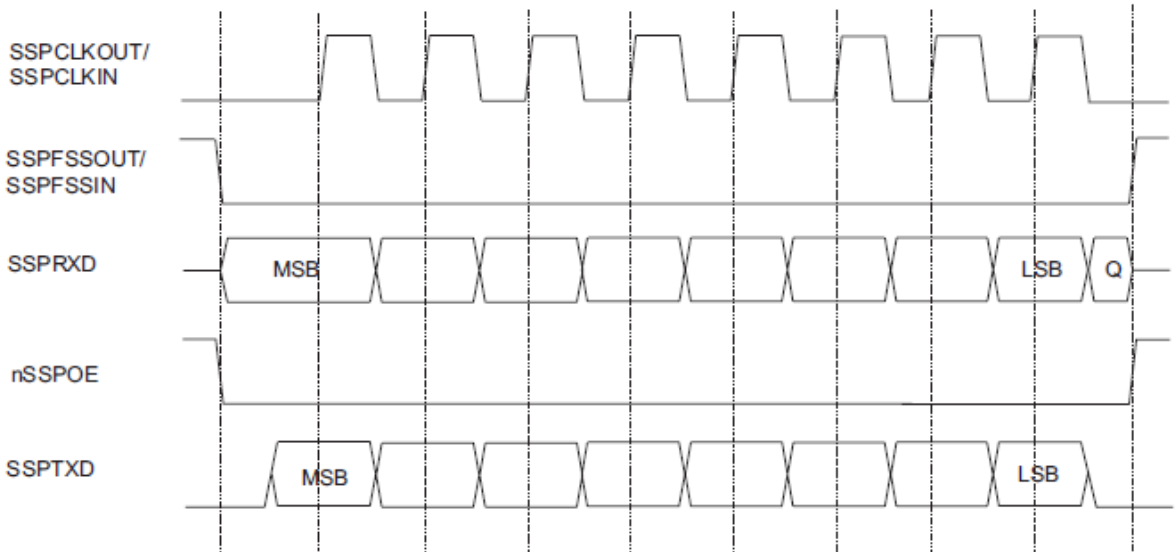
The SPI interface is a four-wire interface where the SSPFSSOUT signal behaves as a slave select. The main feature of the SPI format is that the inactive state and phase of the SSPCLKOUT signal are programmable through the SPO and SPH bits within the SSPSCR0 control register.

SPO, clock polarity
When the SPO clock polarity control bit is LOW, it produces a steady state low value on the SSPCLKOUT pin. If the SPO clock polarity control bit is HIGH, a steady state high value is placed on the SSPCLKOUT pin when data is not being transferred.

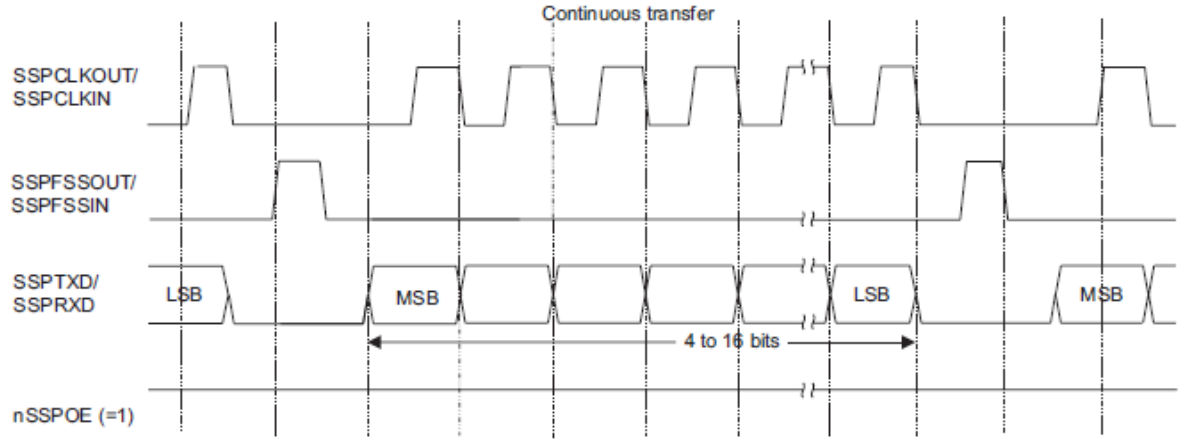
SPH, clock phase
The SPH control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge.

When the SPH phase control bit is LOW, data is captured on the first clock edge transition. If the SPH clock phase control bit is HIGH, data is captured on the second clock edge transition.

20.4.10 SPI Format with SPO=0, SPH=0



SPI frame format (single transfer) SPO=0, SPH=0



SPI frame format (continuous transfer) SPO=0, SPH=0

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the SPI is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- when the SPI is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable).

If the SPI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. This causes slave data to be enabled onto the SSPRXD input line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

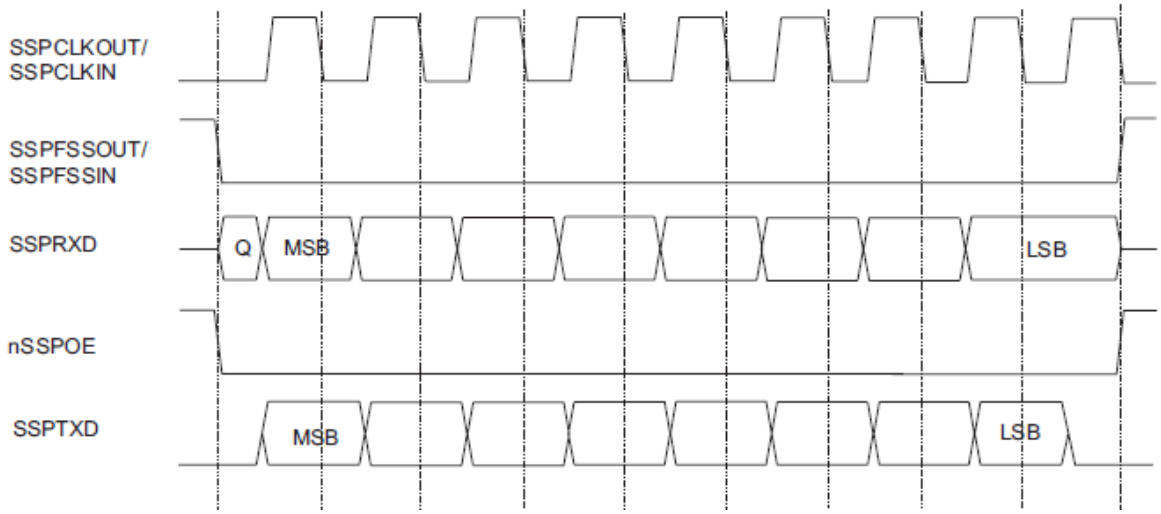
One half SSPCLKOUT period later, valid master data is transferred to the SSPTXD pin. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin goes HIGH after one further half SSPCLKOUT period.

The data is now captured on the rising and propagated on the falling edges of the SSPCLKOUT signal.

In the case of a single word transmission, after all bits of the data word have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

20.4.11 SPI Format with SPO=0, SPH=1



SPI frame format SPO=0, SPH=1

In this configuration, during idle periods:

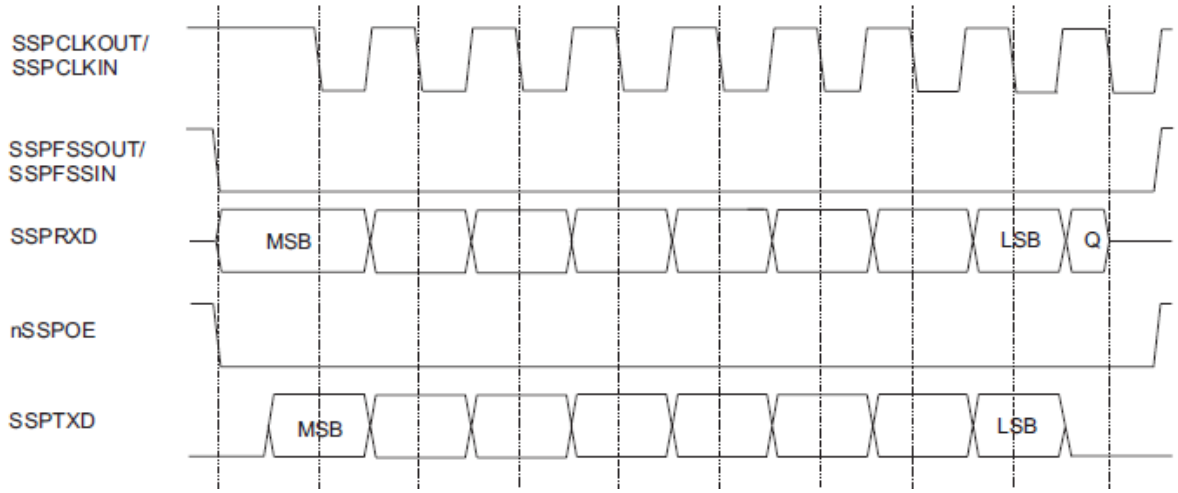
- the SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the SPI is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- when the SPI is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable).
-

If the SPI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After a further one half SSPCLKOUT period, both master and slave valid data is enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a rising edge transition.

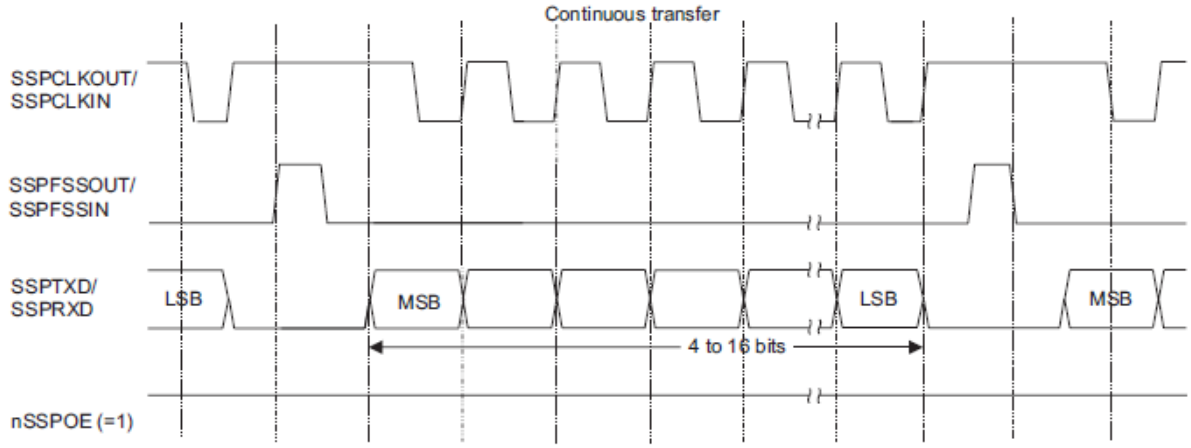
Data is then captured on the falling edges and propagated on the rising edges of the SSPCLKOUT signal.
In the case of a single word transfer, after all bits have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

20.4.12 SPI Format with SPO=1, SPH=0



SPI frame format (single transfer) SPO=1, SPH=0



SPI frame format (continuous transfer) SPO=1, SPH=0

In this configuration, during idle periods

- the SSPCLKOUT signal is forced HIGH
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the SPI is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- when the SPI is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable).

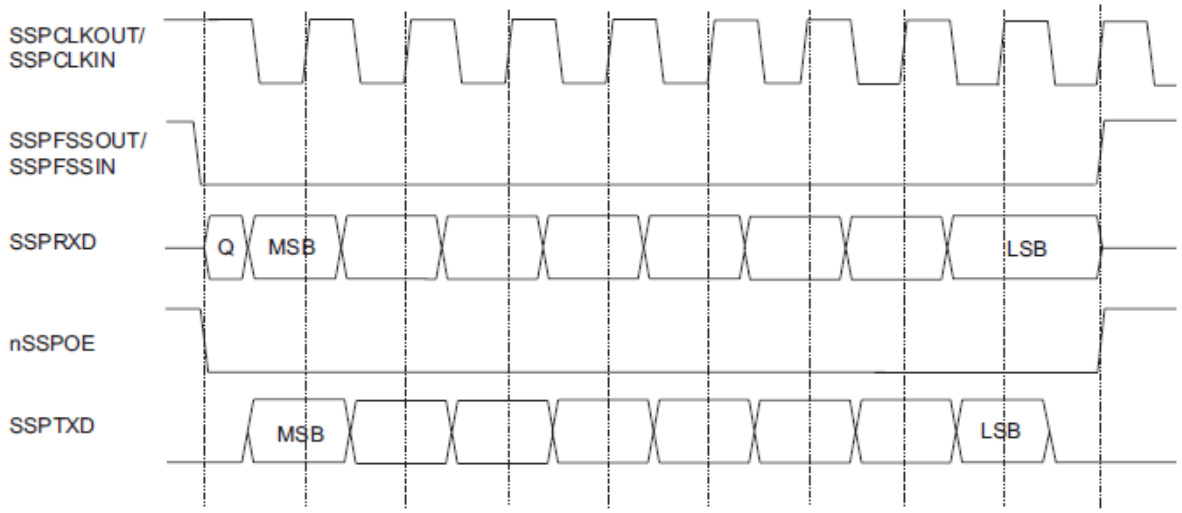
If the SPI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW, which causes slave data to be immediately transferred onto the SSPRXD line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half period later, valid master data is transferred to the SSPTXD line. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin becomes LOW after one further half

SSPCLKOUT period. This means that data is captured on the falling edges and be propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transmission, after all bits of the data word are transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.
However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

20.4.13 SPI Format with SPO=1, SPH=1



SPI frame format SPO=1, SPH=1

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced HIGH
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the SPI is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- when the SPI is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable).

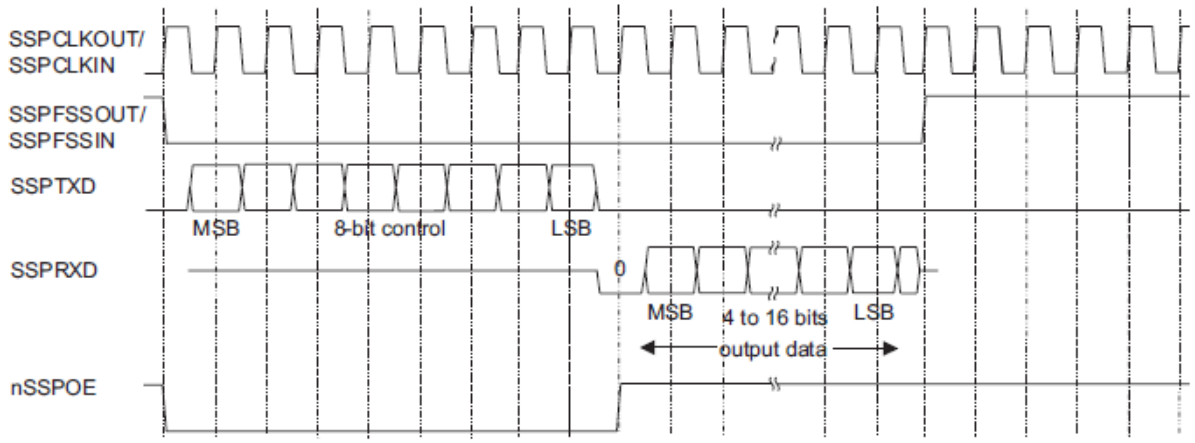
If the SPI is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After a further one half SSPCLKOUT period, both master and slave data are enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SSPCLKOUT signal.

After all bits have been transferred, in the case of a single word transmission, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transmissions, the SSPFSSOUT pins remains in its active LOW state, until the final bit of the last word has been captured, and then returns to its idle state as described above.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

20.4.14 Microwire frame format



Microwire frame format (singe transfer)

Microwire format is very similar to SPI format, except that transmission is half-duplex instead of full-duplex, using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the SPI to the off-chip slave device. During this transmission, no incoming data is received by the SPI. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the required data. The returned data is 4 to 16 bits in length, making the total frame length anywhere from 13 to 25 bits.

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance.

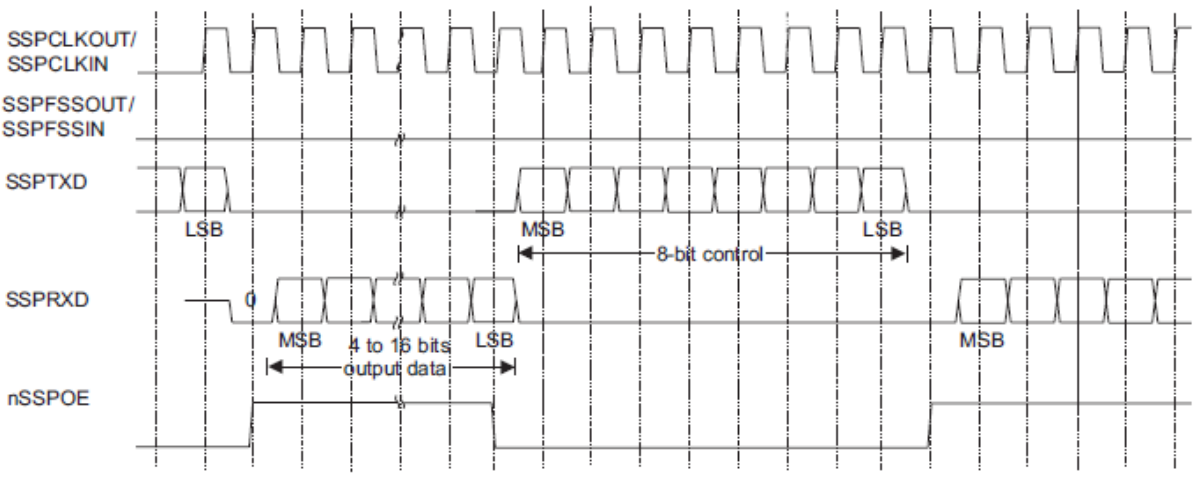
A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SSPFSSOUT causes the value contained in the bottom entry of the transmit FIFO to be transferred to the serial shift register of the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SSPTXD pin. SSPFSSOUT remains LOW for the duration of the frame transmission. The SSPRXD pin remains tristated during this transmission.

The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SSPCLKOUT. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SPI. Each bit is driven onto SSPRXD line on the falling edge of SSPCLKOUT. The SPI in turn latches each bit on the rising edge of SSPCLKOUT. At the end of the frame, for single transfers, the SSPFSSOUT signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter, that causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SSPCLKOUT after the LSB has been latched by the receive shifter, or when the SSPFSSOUT pin goes HIGH.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SSPFSSOUT line is continuously asserted (held LOW) and transmission of data occurs back to back. The control byte of the next frame follows directly after the LSB of the received data

from the current frame. Each of the received values is transferred from the receive shifter on the falling edge SSPCLKOUT, after the LSB of the frame has been latched into the SPI.

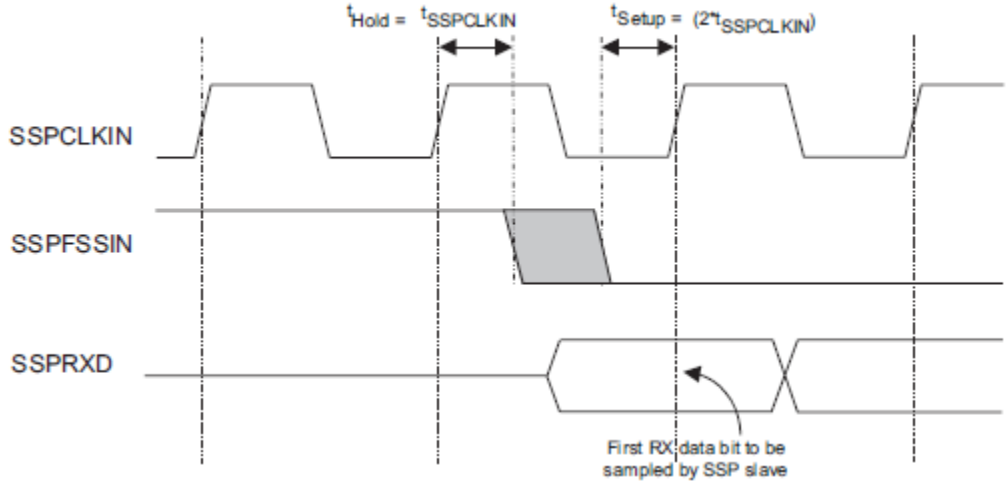


Microwire frame format (continuous transfer)

Setup and hold time requirements on SSPFSSIN with respect to SSPCLKIN in Microwire mode

In the Microwire mode, the SPI slave samples the first bit of receive data on the rising edge of SSPCLKIN after SSPFSSIN has gone LOW. Masters that drive a free-running SSPCKLIN must ensure that the SSPFSSIN signal has sufficient setup and hold margins with respect to the rising edge of SSPCLKIN.

Figure illustrates these setup and hold time requirements. With respect to the SSPCLKIN rising edge on which the first bit of receive data is to be sampled by the SPI slave, SSPFSSIN must have a setup of at least two times the period of SSPCLKIN on which the SPI operates. With respect to the SSPCLKIN rising edge previous to this edge, SSPFSSIN must have a hold of at least one SSPCLKIN period.



20.4.15 DMA interface

The SPI provides an interface to connect to the DMA controller. The DMA operation of the SPI is controlled through the SPI DMA control register, SSPDMACR. The DMA interface includes the following signals, for receive:

SSPRXDMASREQ
Single-character DMA transfer request, asserted by the SPI. This signal is asserted when the receive FIFO contains at least one character.

SSPRXDMABREQ
Burst DMA transfer request, asserted by the SPI. This signal is asserted when the receive FIFO contains four or more characters.

SSPRXDMACLR
DMA request clear, asserted by the DMA controller to clear the receive request signals. If DMA burst transfer is requested, the clear signal is asserted during the transfer of the last data in the burst.

The DMA interface includes the following signals, for transmit:

SSPTXDMASREQ
Single-character DMA transfer request, asserted by the SPI. This signal is asserted when there is at least one empty location in the transmit FIFO.

SSPTXDMABREQ
Burst DMA transfer request, asserted by the SPI. This signal is asserted when the transmit FIFO contains four or less characters.

SSPTXDMACLR
DMA request clear, asserted by the DMA controller to clear the transmit request signals. If DMA burst transfer is requested, the clear signal is asserted during the transfer of the last data in the burst.

The burst transfer and single transfer request signals are not mutually exclusive. They can both be asserted at the same time. For example, when there is more data than the watermark level of four in the receive FIFO, the burst transfer request and the single transfer request are asserted. When the amount of data left in the receive FIFO is less than the watermark level, the single request only is asserted. This is useful for situations where the number of characters left to be received in the stream is less than a burst.

For example, say 19 characters have to be received. The DMA controller then transfers four bursts of four characters and three single transfers to complete the stream.

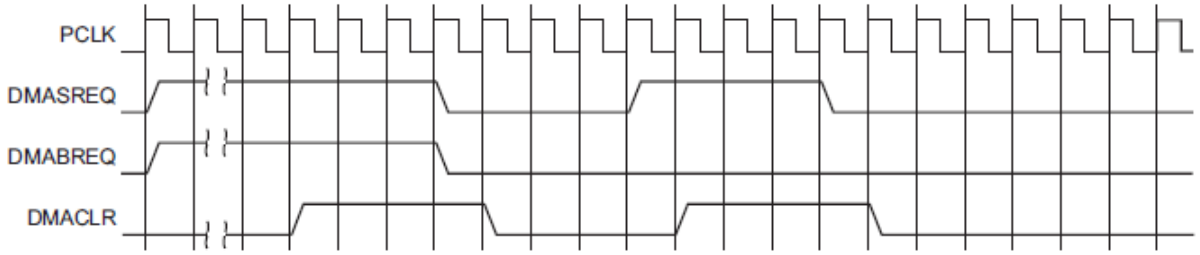
Note: For the remaining three characters the SPI does not assert the burst request.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is deasserted, a request signal can become active again, depending on the conditions described above. All request signals are deasserted if the SPI is disabled or the DMA enable signal is cleared.

Table shows the trigger points for DMABREQ, for both the transmit and receive FIFOs.

	Burst length	
Watermark level	Transmit (number of empty locations)	Receive (number of filled locations)
1/2	4	4

Figure shows the timing diagram for both a single transfer request and a burst transfer request with the appropriate DMA clear signal. The signals are all synchronous to PCLK.



20.5 Registers

The base address of the SPI is not fixed, and might be different for any particular system implementation. The offset of any particular register from the base address, however, is fixed.
The following locations are reserved, and must not be used during normal operation:

- locations at offsets +0x028 to +0x07C and +0xFD0 to +0xFDC are reserved for possible future extensions

Base Address 0x4001_8000

Address	Type	Width	Reset value	Name	Description
SPI Base + 0x00	Read/write	16	0x0000	SSPCR0	Control register 0
SPI Base + 0x04	Read/write	4	0x0	SSPCR1	Control register 1.
SPI Base + 0x08	Read/write	16	0x---	SSPDR	Receive FIFO (read) and transmit FIFO data register (write).
SPI Base + 0x0C	Read	5	0x03	SSPSR	Status register.
SPI Base + 0x10	Read/write	8	0x00	SSPCPSR	Clock pre-scale register.
SPI Base + 0x14	Read/write	4	0x0	SSPIMSC	Interrupt mask set and clear register.
SPI Base + 0x18	Read	4	0x8	SSPRIS	Raw interrupt status register.
SPI Base + 0x1C	Read	4	0x0	SSPMIS	Masked interrupt status register.
SPI Base + 0x20	Write	4	0x0	SSPICR	Interrupt clear register.
SPI Base + 0x24	Read/write	2	0x0	SSPDMACR	DMA control register.
SPI Base + 0x28 to 0x7C	-	-			Reserved
SPI Base + 0x80to 0x8C	-	-			Reserved
SPI Base + 0x90 to 0xFCC	-	-			Reserved
SPI Base + 0xFD0 to 0xFDC	-	-			Reserved for future expansion
SPI base +0xFEO	Read	8	0x22		

The following registers are described in this section:

- Control register 0, SSPCR0
- Control register 1, SSPCR1
- Data register, SSPDR
- Status register, SSPSR
- Clock pre-scale register, SSPCPSR
- Interrupt mask set or clear register, SSPIMSC0
- Raw interrupt status register, SSPRIS
- Masked interrupt status register, SSPMIS
- Interrupt clear register, SSPICR
- DMA control register, SSPDMACR

20.5.1 Control register 0, SSPCR0

SSPCR0 is control register 0 and contains five bit fields that control various functions within the SPI.

Bits	Name	Type	Function
15:8	SCR	Read/write	Serial clock rate. The value SCR is used to generate the transmit and receive bit rate of the SPI. The bit rate is: $\frac{F_{SSPCLK}}{CPSDVR \times (1+SCR)}$ where CPSDVSR is an even value from 2 to 254, programmed through the SSPCPSR register and SCR is a value from 0 to 255.
7	SPH	Read/write	SSPCLKOUT phase (applicable SPI frame format only).
6	SPO	Read/write	SSPCLKOUT polarity (applicable to SPI frame format only).
5:4	FRF	Read/write	Frame format: 00 SPI frame format 01 Synchronous Serial frame format 10 Microwire frame format 11 Reserved, undefined operation
3:0	DSS	Read/write	Data Size Select: 0000 Reserved, undefined operation 0001 Reserved, undefined operation 0010 Reserved, undefined operation 0011 4-bit data 0100 5-bit data 0101 6-bit data 0110 7-bit data 0111 8-bit data 1000 9-bit data 1001 10-bit data 1010 11-bit data 1011 12-bit data 1100 13-bit data 1101 14-bit data 1110 15-bit data 1111 16-bit data.

20.5.2 Control register 1, SSPCR1

SSPCR1 is the control register 1 and contains four different bit fields, which control various functions within the SPI.

Bits	Name	Type	Function
15:4	-	-	Reserved, read unpredictable, should be written as 0.
3	SOD	Read/write	Slave-mode output disable. This bit is relevant only in the slave mode (MS=1). In multiple-slave systems, it is possible for an SPI master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems the RXD lines from multiple slaves could be tied together. To operate in such systems, the SOD bit can be set if the SPI slave is not supposed to drive the SSPTXD line. 0 = SPI can drive the SSPTXD output in slave mode. 1 = SPI must not drive the SSPTXD output in slave mode.
2	MS	Read/write	Master or slave mode select. This bit can be modified only when the SPI is disabled (SSE=0): 0 = device configured as master (default) 1 = device configured as slave.
1	SSE	Read/write	Synchronous serial port enable: 0 = SPI operation disabled 1 = SPI operation enabled.
0	LBM	Read/write	Loop back mode: 0 = Normal serial port operation enabled 1 = Output of transmit serial shifter is connected to input of receive serial shifter internally.

20.5.3 Data register, SSPDR

SSPDR is the data register and is 16-bits wide. When SSPDR is read, the entry in the receive FIFO (pointed to by the current FIFO read pointer) is accessed. As data values are removed by the SPI

receive logic from the incoming data frame, they are placed into the entry in the receive FIFO (pointed to by the current FIFO write pointer).

When SSPDR is written to, the entry in the transmit FIFO (pointed to by the write pointer), is written to. Data values are removed from the transmit FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the SSPTXD pin at the programmed bit rate.

When a data size of less than 16 bits is selected, the user must right-justify data written to the transmit FIFO. The transmit logic ignores the unused bits. Received data less than 16 bits is automatically right-justified in the receive buffer.

When the SPI is programmed for Microwire frame format, the default size for transmit data is eight bits (the most significant byte is ignored). The receive data size is controlled by the programmer. The transmit FIFO and the receive FIFO are not cleared even when SSE is set to zero. This allows the software to fill the transmit FIFO before enabling the SPI.

Bits	Name	Type	Function
15:0	DATA	Read/write	Transmit/Receive FIFO: Read = Receive FIFO Write = Transmit FIFO. You must right-justify data when the SPI is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.

20.5.4 Status register, SSPSR

SSPSR is a read-only status register that contains bits that indicate the FIFO fill status and the SPI busy status.

Bits	Name	Type	Function
Bits	Name	Type	Function
15:5	-	-	Reserved, read unpredictable, should be written as 0.
4	BSY	Read	SPI busy flag (read-only): 0 = SPI is idle 1 = SPI is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.
3	RFF	Read	Receive FIFO full (read-only): 0 = Receive FIFO is not full 1 = Receive FIFO is full.
2	RNE	Read	Receive FIFO not empty (read-only): 0 = Receive FIFO is empty 1 = Receive FIFO is not empty.
1	TNF	Read	Transmit FIFO not full (read-only): 0 = Transmit FIFO is full 1 = Transmit FIFO is not full.
0	TFE	Read	Transmit FIFO empty (read-only): 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty.

20.5.5 Clock pre-scale register, SSPCPSR

SSPCPSR is the clock pre-scale register and specifies the division factor by which the input SSPCLK must be internally divided before further use.

The value programmed into this register must be an even number between 2 to 254. The least significant bit of the programmed number is hard-coded to zero. If an odd number is written to this register, data read back from this register has the least significant bit as zero.

Bits	Name	Type	Function

15:8	-	-	Reserved, read unpredictable, must be written as 0.
7:0	CPSDVSR	Read/write	Clock pre-scale divisor. Must be an even number from 2 to 254, depending on the frequency of SSPCLK. The least significant bit always returns zero on reads.

20.5.6 Interrupt mask set or clear register, SSPIMSC

The SSPIMSC register is the interrupt mask set or clear register. It is a read/write register.

On a read this register gives the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask.

All the bits are cleared to 0 when reset.

Bits	Name	Type	Function
15:4	Reserved	-	Reserved, read as zero, do not modify.
3	TXIM	Read/write	Transmit FIFO interrupt mask: 0 = Tx FIFO half empty or less condition interrupt is masked 1 = Tx FIFO half empty or less condition interrupt is not masked.
2	RXIM	Read/write	Receive FIFO interrupt mask: 0 = Rx FIFO half full or less condition interrupt is masked 1 = Rx FIFO half full or less condition interrupt is not masked.
1	RTIM	Read/write	Receive timeout interrupt mask: 0 = RxFIFO not empty and no read prior to timeout period interrupt is masked 1= RxFIFO not empty and no read prior to timeout period interrupt is not masked.
0	RORIM	Read/write	Receive overrun interrupt mask: 0 = RxFIFO written to while full condition interrupt is masked 1 = RxFIFO written to while full condition interrupt is not masked.

20.5.7 Raw interrupt status register, SSPRIS

The SSPRIS register is the raw interrupt status register. It is a read-only register. On a read this register gives the current raw status value of the corresponding interrupt prior to masking. A write has no effect.

Bits	Name	Type	Function
15:4	Reserved	-	Reserved, read as zero, do not modify
3	TXRIS	Read	Gives the raw interrupt state (prior to masking) of the SSPTXINTR interrupt
2	RXRIS	Read	Gives the raw interrupt state (prior to masking) of the SSPRXINTR interrupt
1	RTRIS	Read	Gives the raw interrupt state (prior to masking) of the SSPRTINTR interrupt
0	RORRIS	Read	Gives the raw interrupt state (prior to masking) of the SSPRORINTR interrupt

20.5.8 Masked interrupt status register, SSPMIS

Bits	Name	Type	Function
15:4	Reserved	-	Reserved, read as zero, do not modify
3	TXMIS	Read	Gives the transmit FIFO masked interrupt state (after masking) of the SSPTXINTR interrupt
2	RXMIS	Read	Gives the receive FIFO masked interrupt state (after masking) of the SSPRXINTR interrupt
1	RTMIS	Read	Gives the receive timeout masked interrupt state (after masking) of the SSPRTINTR interrupt
0	RORMIS	Read	Gives the receive over run masked interrupt status (after masking) of the SSPRORINTR interrupt

20.5.9 Interrupt clear register, SSPICR

The SSPICR register is the interrupt clear register and is write-only. On a write of 1, the corresponding interrupt is cleared. A write of 0 has no effect.

Bits	Name	Type	Function
15:2	Reserved	-	Reserved, read as zero, do not modify
1	RTIC	Write	Clears the SSPRTINTR interrupt
0	RORIC	Write	Clears the SSPRORINTR interrupt

20.5.10 DMA control register, SSPDMACR

The SSPDMACR register is the DMA control register. It is a read/write register. All the bits are cleared to 0 on reset.

Bits	Name	Type	Function
15:2	RESERVED	-	Reserved, read as zero, do not modify
1	Transmit DMA Enable (TXDMAE)	Read/ write	If this bit is set to 1, DMA for the transmit FIFO is enabled
0	Receive DMA Enable (RXDMAE)	Read/ write	If this bit is set to 1, DMA for the receive FIFO is enabled

21. I2C Interface

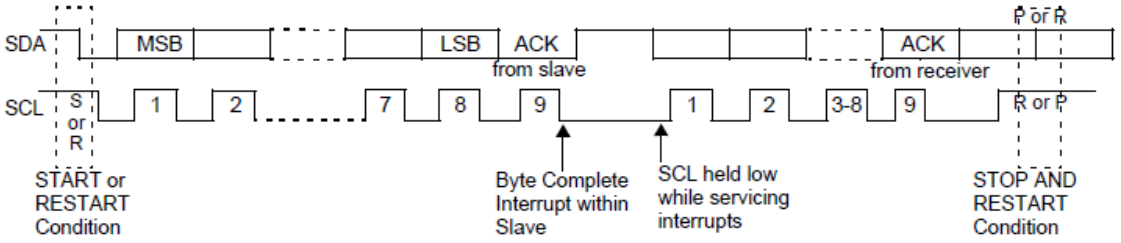
21.1 Introduction

The I2C bus is a two-wire serial interface, consisting of a serial data line (SDA) and a serial clock (SCL). These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. Devices can also be considered as masters or slaves when performing data transfers. A master is a device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

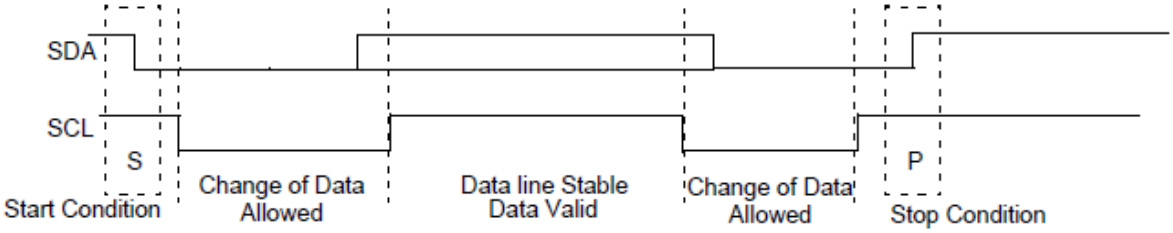
21.2 I2C Features

- I2C two-wire serial interface – consists of a serial data line (SDA) and a serial clock (SCL)
- Three speeds:
- Standard mode (0 to 100 Kb/s)
- High-speed mode (≤ 1.8 Mb/s)
- Clock synchronization
- Master OR slave operation
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Bulk transmit mode
- Ignores CBUS addresses (an older ancestor of I2C that used to share the I2C bus)
- Transmit and receive buffers
- Interrupt or polled-mode operation

21.3 Data format



START and STOP Condition



21.4 Registers

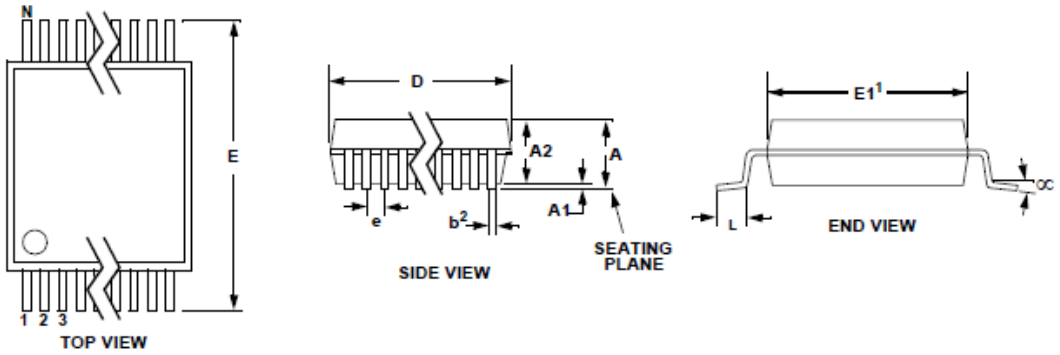
Base Address 0x4002_0000

Register	Offset	Description
IC_CON	0x00	Control Register. This register can be written only when the XX_apb_i2c is disabled, which corresponds...
IC_TAR	0x04	Target Address Register If the configuration parameter I2C_DYNAMIC_TAR_UPDATE is set to 'No'...
IC_SAR	0x08	Slave Address Register
IC_HS_MADDR	0x0C	High Speed Master Mode Code Address Register
IC_DATA_CMD	0x10	Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling...
IC_SS_SCL_HCNT	0x14	Standard Speed Clock SCL High Count Register
IC_SS_SCL_LCNT	0x18	Standard Speed Clock SCL Low Count Register
IC_FS_SCL_HCNT	0x1C	Fast Mode or Fast Mode Plus Clock SCL High Count Register
IC_FS_SCL_LCNT	0x20	Fast Mode or Fast Mode Plus Clock SCL Low Count Register
IC_HS_SCL_HCNT	0x24	High Speed Clock SCL High Count Register
IC_HS_SCL_LCNT	0x28	High Speed Clock SCL Low Count Register
IC_INTR_STAT	0x2C	Interrupt Status Register Each bit in this register has a corresponding mask bit in the IC_INTR_MASK...
IC_INTR_MASK	0x30	Interrupt Mask Register. These bits mask their corresponding interrupt status bits. This register...
IC_RAW_INTR_STAT	0x34	Raw Interrupt Status Register Unlike the IC_INTR_STAT register, these bits are not masked so...
IC_RX_TL	0x38	Receive FIFO Threshold Register
IC_TX_TL	0x3C	Transmit FIFO Threshold Register
IC_CLR_INTR	0x40	Clear Combined and Individual Interrupt Register
IC_CLR_RX_UNDER	0x44	Clear RX_UNDER Interrupt Register
IC_CLR_RX_OVER	0x48	Clear RX_OVER Interrupt Register
IC_CLR_TX_OVER	0x4C	Clear TX_OVER Interrupt Register
IC_CLR_RD_REQ	0x50	Clear RD_REQ Interrupt Register
IC_CLR_TX_ABRT	0x54	Clear TX_ABRT Interrupt Register
IC_CLR_RX_DONE	0x58	Clear RX_DONE Interrupt Register
IC_CLR_ACTIVITY	0x5C	Clear ACTIVITY Interrupt Register
IC_CLR_STOP_DET	0x60	Clear STOP_DET Interrupt Register
IC_CLR_START_DET	0x64	Clear START_DET Interrupt Register
IC_CLR_GEN_CALL	0x68	Clear GEN_CALL Interrupt Register
IC_ENABLE	0x6C	Enable Register

IC_STATUS	0x70	Status Register This is a read-only register used to indicate the current transfer status...
IC_TXFLR	0x74	Transmit FIFO Level Register This register contains the number of valid data entries in the...
IC_RXFLR	0x78	Receive FIFO Level Register This register contains the number of valid data entries in the receive...
IC_SDA_HOLD	0x7C	SDA Hold Time Length Register The bits [15:0] of this register are used to control the hold...
IC_TX_ABRT_SOURCE	0x80	Transmit Abort Source Register This register has 32-bit that indicate the source of the TX_ABRT...
IC_SLV_DATA_NACK_ONLY	0x84	Generate Slave Data NACK Register The register is used to generate a NACK for the data part of...
IC_DMA_CR	0x88	DMA Control Register This register is only valid when XX_apb_i2c is configured with a set of DMA...
IC_DMA_TDLR	0x8C	DMA Transmit Data Level Register This register is only valid when the XX_apb_i2c is configured...
IC_DMA_RDLR	0x90	Receive Data Level Register This register is only valid when XX_apb_i2c is configured with...
IC_SDA_SETUP	0x94	SDA Setup Register This register controls the amount of time delay (in terms of number of ic_clk...
IC_ACK_GENERAL_CALL	0x98	ACK General Call Register The register controls whether XX_apb_i2c responds with ACK or NACK...
STATUS	0x9C	Enable Status Register The register is used to report the XX_apb_i2c hardware status when the...
IC_FS_SPKLEN	0xA0	SS, FS or FM+ spike suppression limit. This register is used to store the duration, measured...
IC_CLR_RESTART_DET	0xA8	Clear RESTART_DET Interrupt Register
IC_SCL_STUCK_AT_LOW_TIMEOUT	0xAC	SCL Stuck at Low Timeout This register is used to store the duration, measured in ic_clk cycles, …
IC_SDA_STUCK_AT_LOW_TIMEOUT	0xB0	SDA Stuck at Low Timeout This register is used to store the duration, measured in ic_clk cycles, …
IC_CLR_SCL_STUCK_DET	0xB4	Clear SCL Stuck at Low Detect Interrupt Register
IC_DEVICE_ID	0xB8	Device-ID Register This Register contains the Device-ID of the component which includes 12-bits...

22. Package

22.1 TSSOP20L

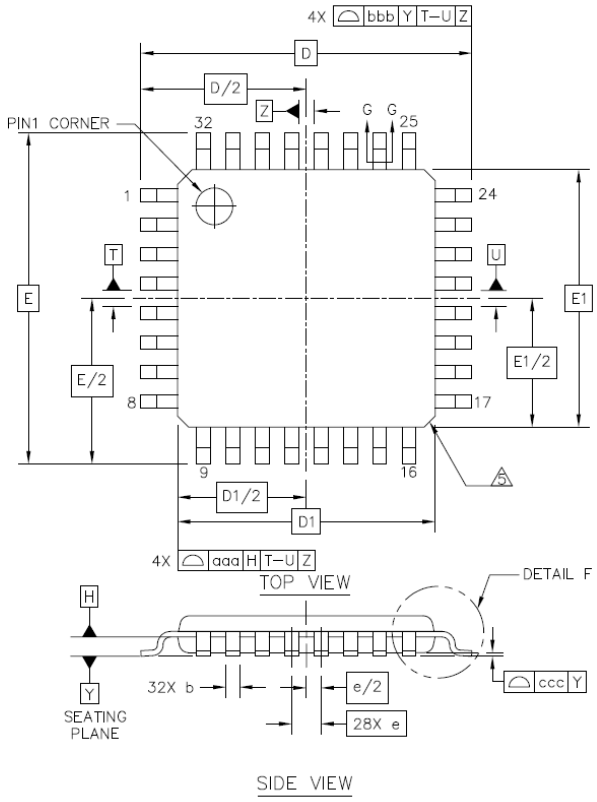


	INCHES			MILLIMETERS			NOTE
DIM	MIN	NOM	MAX	MIN	NOM	MAX	
A	--	--	0.043	--	--	1.10	
A1	0.002	0.004	0.006	0.05	--	0.15	
A2	0.03346	0.0354	0.037	0.85	0.90	0.95	
b	0.00748	0.0096	0.012	0.19	0.245	0.30	2,3
D	0.252	0.256	0.259	6.40	6.50	6.60	1
E	0.248	0.2519	0.256	6.30	6.40	6.50	
E1	0.169	0.1732	0.177	4.30	4.40	4.50	1
e	--	--	0.026	--	--	0.65	
L	0.020	0.024	0.028	0.50	0.60	0.70	
∝	0°	4°	8°	0°	4°	8°	

JEDEC #: MO-153

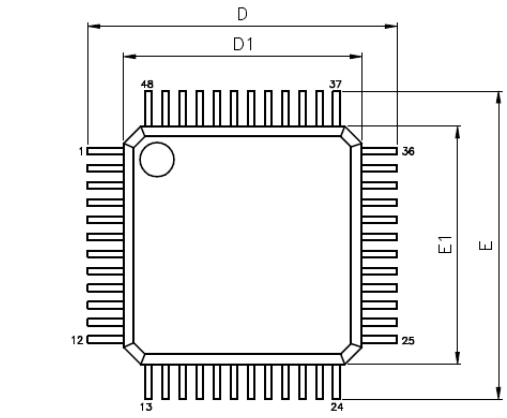
Controlling Dimension is Millimeters.

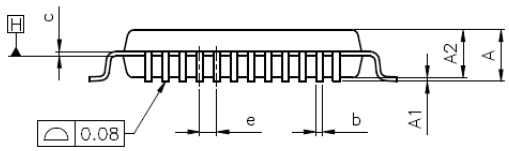
22.2 LPFQ32



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.2
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	0.95	---	1.05
LEAD WIDTH(PLATING)		b	0.32	0.37	0.42
LEAD WIDTH		b1	0.32	0.35	0.38
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D	9 BSC		
	Y	E	9 BSC		
	X	D1	7 BSC		
	Y	E1	7 BSC		
LEAD PITCH		e	0.8 BSC		
		L	0.5	0.6	0.7
FOOTPRINT		L1	1 REF		
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2	---	---
PACKAGE EDGE TOLERANCE		aaa	0.2		
LEAD EDGE TOLERANCE		bbb	0.2		
COPLANARITY		ccc	0.1		
LEAD OFFSET		ddd	0.2		
MOLD FLATNESS		eee	0.05		

22.3 LPFQ48





VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	NOM.	MAX.
A	--	--	1.60
A1	0.05	--	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c	0.09	--	0.20
D	9.00 BSC		
D1	7.00 BSC		
E	9.00 BSC		
E1	7.00 BSC		
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		
θ	0°	3.5°	7°

THERMALLY ENHANCED DIMENSIONS(SHOWN IN MM)

PAD SIZE	E2		D2	
	MIN.	MAX.	MIN.	MAX.
20*X20*	4.31	5.36	4.31	5.36