WT58F2C8/WT58F2C9
Flash Memory Type
32-bit Microcontroller

Data Sheet

Rev. 1.04

February 2015

Disclaimers

Right to make change –

This document provides technical information for user. Weltrend Semiconductor, Inc. reserves the right to make change without further notice to any products herein.

Table of Contents

1. General Description

The WT58F2C8/WT58F2C9 is a high-performance 32-Bit Microcontroller. It incorporates the 32-bit RISC CPU operating up to 48 MHz, Flash memory up to 192Kbytes and SRAM to 12K bytes, and rich peripherals/interfaces such as high-speed ADC, PWM, I2C, SPI, and UART.

Part No.	Flash	SRAM
WT58F2C8	128 KB	12 KB
WT58F2C9	192 KB	12 KB

2. Features

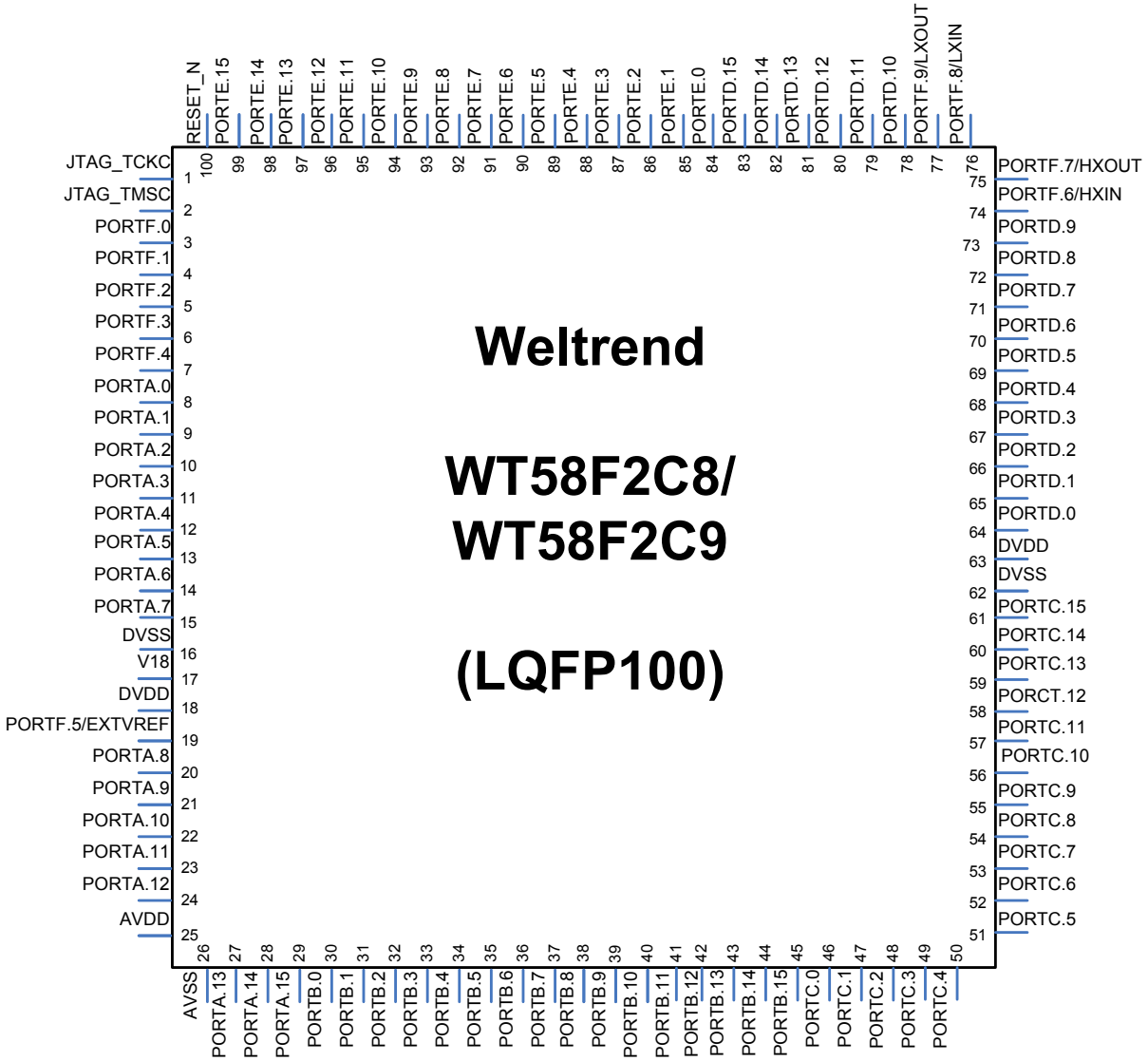
- Embedded 32-bit RISC CPU core
  - 1.28 DMIPS/MHz (Dhrystone 2.1) performance
  - Normal operating mode: 48 MHz, 36 MHz, 24 MHz, 12 MHz.
- Memory:
  - Flash memory: 192K bytes (48K x 32)
  - SRAM: 12K bytes (3K x 32)
- CPU clock source
  - External crystal oscillator 1 MHz ~ 48 MHz
  - PLL output up to 48 MHz
  - Internal RC oscillator 12 / 24 / 36 / 48 MHz
  - External crystal oscillator 32 kHz for RTC counter
  - Low speed RC 32 kHz
- Clock Management
  - Internal RC oscillator can be calibrated ± 1% with 32 kHz crystal and ± 4% without 32 kHz crystal
  - Programmable PLLs and oscillator clock sources
  - Independent 32 kHz RC Watchdog Timer (IWDT)
  - start-up via High Speed RC oscillator.
  - Window Watchdog timer with system clock
- Power Management
  - Low-power management modes (Key Wake up)
  - Integrated Power on reset, Low voltage Reset and 8 level low DVDD detector(LVD)
- High-Speed ADC Module (3.3V~2.4V)
  - 10-bit resolution with 1 Successive Approximation Register (SAR) converters (1M SPS)
    - ±1 LSB accuracy at 3.0V
  - Up to 12 input sources (one dedicated for temperature sensor)
  - Vref = 1.25V / 2.5V
  - Temperature sensor (±3℃)
- Up to 12 16-bit timer with IC/OC/OCN/PWM
- DMA: ADC, I2C(x8), UART(x4), TIMER(x12), SPI(x4)
- 5V-input-torlence Tri-state I/O structure. 16 External Interrupt Input (IRQ), 16 Powerdown IO Wake up.
- Communication interface
  - Up to 7 I2C bus included 1 multi-master I2C bus up to 400 kHz (1 MHz)

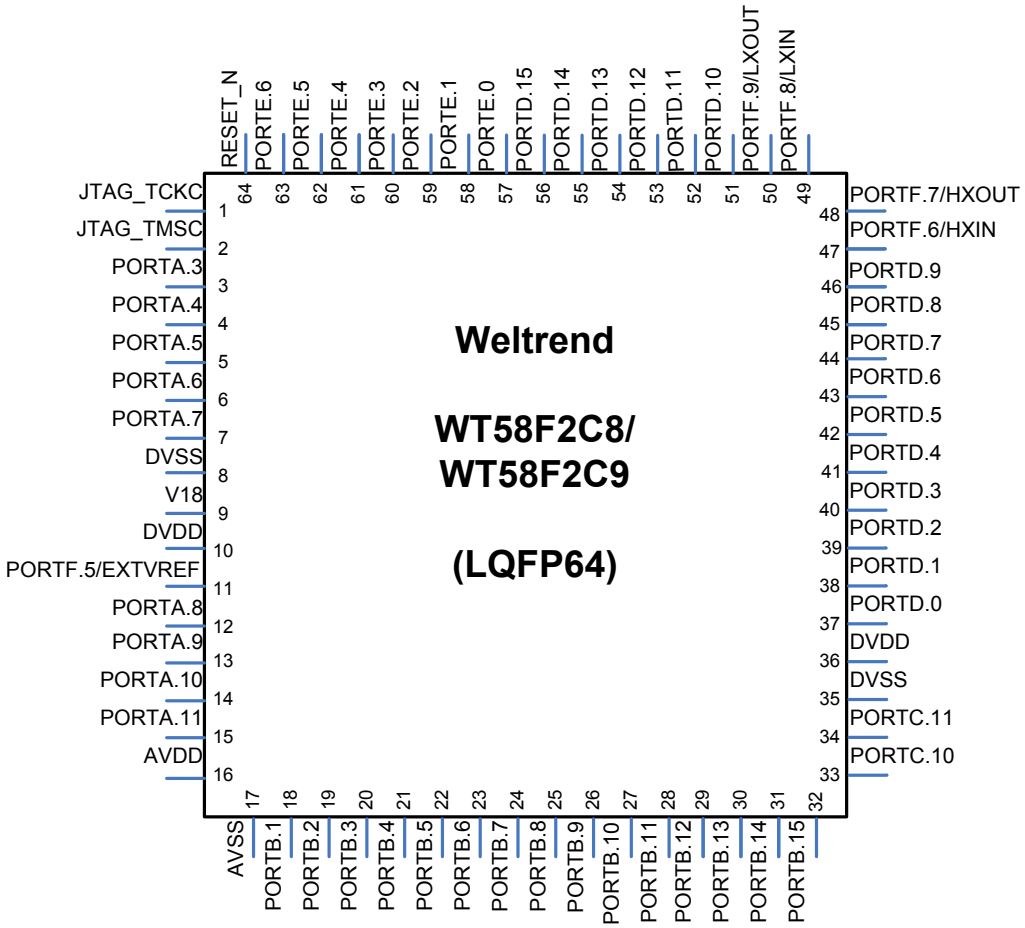
- Up to 4 master/slave SPI up to 24 MHz
- Up to 4 UART up to 1.5Mbps

- PWM up to 12 channels.
- Support build in ISP & ICE mode (2-wire JTAG mode)
- Flash memory read out protection
- Power Consumption
  - Active Mode: 650uA/MHz.
  - Standby Mode: less than 6uA
- Operating Conditions:
  - 2.0V to 3.6V, -40ºC to +85ºC.
- ESD protection HBM > 4KV, MM > 400V
- Package type: LQFP64, LQFP100 (0.5mm)

3. Pin Configuration

3.1 Package





3.2 Pin Description

LQFP 100	LQFP 64	Pin Name	I/O	Function Description
1	1	JTAG_TCKC		
2	2	JTAG_TMSC		
3		PORTF.0	I/O	GPIO / TMR5_capi0
4		PORTF.1	I/O	GPIO / TMR5_capi1
5		PORTF.2	I/O	GPIO / TMR5_mato0
6		PORTF.3	I/O	GPIO / TMR5_mato1
7		PORTF.4	I/O	GPIO / IR
8		PORTA.0	I/O	GPIO / SPI3_MISO
9		PORTA.1	I/O	GPIO / SPI3_MOSI
10		PORTA.2	I/O	GPIO / SPI3_SCLK
11	3	PORTA.3	I/O	GPIO / SPI3_NSS
12	4	PORTA.4	I/O	GPIO / TMR9_capi0
13	5	PORTA.5	I/O	GPIO / TMR9_capi1
14	6	PORTA.6	I/O	GPIO / TMR9_mato0
15	7	PORTA.7	I/O	GPIO / TMR9_mato1
16	8	DVSS	P	3.3V VSS
17	9	V18	P	1.8V Power
18	10	DVDD	P	3.3V Power
19	11	PORTF.5/EXTVREF	I/O	GPIO / ADC reference voltage input
20	12	PORTA.8	I/O	GPIO / AIN0 / TMR2_capi0
21	13	PORTA.9	I/O	GPIO / AIN1 / TMR2_capi1
22	14	PORTA.10	I/O	GPIO / AIN2 / TMR2_mato0
23	15	PORTA.11	I/O	GPIO / AIN3 / TMR2_mato1
24		PORTA.12	I/O	GPIO / AIN4 / TMR7_capi0
25	16	AVDD	P	Analog 3V power
26	17	AVSS	P	Analog GND
27		PORTA.13	I/O	GPIO / AIN5 / TMR7_capi1
28		PORTA.14	I/O	GPIO / AIN6 / TMR7_mato0
29		PORTA.15	I/O	GPIO / AIN7 / TMR7_mato1
30		PORTB.0	I/O	GPIO / AIN8 / TMR4_capi0
31	18	PORTB.1	I/O	GPIO / AIN9 / TMR4_capi1
32	19	PORTB.2	I/O	GPIO / AIN10 / TMR4_mato0
33	20	PORTB.3	I/O	GPIO / AIIN11 / TMR4_mato1
34	21	PORTB.4	I/O	GPIO / I2C5_SCL / TMR6_capi0
35	22	PORTB.5	I/O	GPIO / I2C5_SDA / TMR6_capi1
36	23	PORTB.6	I/O	GPIO / I2C7_SCL / TMR6_mato0
37	24	PORTB.7	I/O	GPIO / I2C7_SDA / TMR6_mato1
38	25	PORTB.8	I/O	GPIO / I2C2_SCL / TMR8_capi0
39	26	PORTB.9	I/O	GPIO / I2C2_SDA / TMR8_capi1
40	27	PORTB.10	I/O	GPIO / I2C3_SCL / TMR8_mato0
41	28	PORTB.11	I/O	GPIO / I2C3_SDA / TMR8_mato1
42	29	PORTB.12	I/O	GPIO / I2C4_SCL
43	30	PORTB.13	I/O	GPIO / I2C4_SDA / IR

LQFP 100	LQFP 64	Pin Name	I/O	Function Description
44	31	PORTB.14	I/O	GPIO / I2C0_SCL / PLL48_EXT_REFIN
45	32	PORTB.15	I/O	GPIO / I2C0_SDA
46		PORTC.0	I/O	GPIO / I2C6_SCL
47		PORTC.1	I/O	GPIO / I2C6_SDA
48		PORTC.2	I/O	GPIO
49		PORTC.3	I/O	GPIO
50		PORTC.4	I/O	GPIO / TMRB_capi0
51		PORTC.5	I/O	GPIO / TMRB_capi1
52		PORTC.6	I/O	GPIO / TMRB_mato0
53		PORTC.7	I/O	GPIO / TMRB_mato1
54		PORTC.8	I/O	GPIO / UART3_TX
55		PORTC.9	I/O	GPIO / UART3_RX
56	33	PORTC.10	I/O	GPIO / PWM6
57	34	PORTC.11	I/O	GPIO / PWM7
58		PORTC.12	I/O	GPIO / PWM8
59		PORTC.13	I/O	GPIO / PWM9
60		PORTC.14	I/O	GPIO / PWM10
61		PORTC.15	I/O	GPIO / PWM11
62	35	DVSS	P	
63	36	DVDD	P	
64	37	PORTD.0	I/O	GPIO / TMR0_capi0
65	38	PORTD.1	I/O	GPIO / TMR0_capi1
66	39	PORTD.2	I/O	GPIO / TMR0_mato0
67	40	PORTD.3	I/O	GPIO / TMR0_mato1
68	41	PORTD.4	I/O	GPIO / SPI0_MISO
69	42	PORTD.5	I/O	GPIO / SPI0_MOSI
70	43	PORTD.6	I/O	GPIO / SPI0_SCLK
71	44	PORTD.7	I/O	GPIO / SPI0_NSS
72	45	PORTD.8	I/O	GPIO / SPI1_MISO
73	46	PORTD.9	I/O	GPIO / SPI1_MOSI
74	47	PORTF.6/HXIN	I/O	GPIO / 1M ~ 48 MHz XTAL oscillator input
75	48	PORTF.7/HXOUT	I/O	GPIO / 1M ~ 48 MHz XTAL oscillator output
76	49	PORTF.8/LXIN	I/O	GPIO / 32768Hz XTAL oscillator input
77	50	PORTF.9/LXOUT	I/O	GPIO / 32768Hz XTAL oscillator output
78	51	PORTD.10	I/O	GPIO / SPI1_SCLK
79	52	PORTD.11	I/O	GPIO / SPI1_NSS
80	53	PORTD.12	I/O	GPIO / SPI2_MISO / TMRA_capi0
81	54	PORTD.13	I/O	GPIO / SPI2_MOSI / TMRA_capi1
82	55	PORTD.14	I/O	GPIO / SPI2_SCLK / TMRA_mato0
83	56	PORTD.15	I/O	GPIO / SPI2_NSS / TMRA_mato1
84	57	PORTE.0	I/O	GPIO / UART0_TX / TMR1_capi0
85	58	PORTE.1	I/O	GPIO / UART0_RX / TMR1_capi1
86	59	PORTE.2	I/O	GPIO / UART1_TX / TMR1_mato0
87	60	PORTE.3	I/O	GPIO / UART1_RX / TMR1_mato1
88	61	PORTE.4	I/O	GPIO / UART2_TX

LQFP 100	LQFP 64	Pin Name	I/O	Function Description
89	62	PORTE.5	I/O	GPIO / UART2_RX
90	63	PORTE.6	I/O	GPIO / PWM0
91		PORTE.7	I/O	GPIO / PWM1
92		PORTE.8	I/O	GPIO / PWM2
93		PORTE.9	I/O	GPIO / PWM3
94		PORTE.10	I/O	GPIO / PWM4
95		PORTE.11	I/O	GPIO / PWM5
96		PORTE.12	I/O	GPIO / TMR3_capi0
97		PORTE.13	I/O	GPIO / TMR3_capi1
98		PORTE.14	I/O	GPIO / TMR3_mato0
99		PORTE.15	I/O	GPIO / TMR3_mato1
100	64	RESET_N	I	External reset input, active Low

3.3 Package Type Reference

	LQFP64	LQFP100
I2C	6	7
SPI	3	4
UART	3	4
PWM	3	12
TMR Capture/Match	7	12
ADC	7 inputs	12 inputs
Temperature Sensor	1	1
GPIO	54 (note1)	90 (note1)

Note 1: The number specified in the above table shows the maximum number of modules can be used.

3.4 Multi-Function I/O Port Priority List

Following is the priority list when multiple functions exist on one port pin.

Pin Name	Default	1st Priority	2nd Priority	3rd Priority	Remap
PortA.0	PTA_GPIO[0]	PTA_WAKE[0]	SPI3_MISO		
PortA.1	PTA_GPIO[1]	PTA_WAKE[1]	SPI3_MOSI		
PortA.2	PTA_GPIO[2]	PTA_WAKE[2]	SPI3_SCK		
PortA.3	PTA_GPIO[3]	PTA_WAKE[3]	SPI3_NSS		
PortA.4	PTA_GPIO[4]	PTA_WAKE[4]	TMR9		
PortA.5	PTA_GPIO[5]	PTA_WAKE[5]	TMR9		
PortA.6	PTA_GPIO[6]	PTA_WAKE[6]	TMR9		
PortA.7	PTA_GPIO[7]	PTA_WAKE[7]	TMR9		
PortA.8	PTA_GPIO[8]	AIN[0]	TMR2		
PortA.9	PTA_GPIO[9]	AIN[1]	TMR2		
PortA.10	PTA_GPIO[10]	AIN[2]	TMR2		
PortA.11	PTA_GPIO[11]	AIN[3]	TMR2		
PortA.12	PTA_GPIO[12]	AIN[4]	TMR7		
PortA.13	PTA_GPIO[13]	AIN[5]	TMR7		
PortA.14	PTA_GPIO[14]	AIN[6]	TMR7		
PortA.15	PTA_GPIO[15]	AIN[7]	TMR7		
PortB.0	PTB_GPIO[0]	AIN[8]	PTB_WAKE[0]	TMR4	
PortB.1	PTB_GPIO[1]	AIN[9]	PTB_WAKE[1]	TMR4	
PortB.2	PTB_GPIO[2]	AIN]10]	PTB_WAKE[2]	TMR4	
PortB.3	PTB_GPIO[3]	AIN[11]	PTB_WAKE[3]	TMR4	
PortB.4	PTB_GPIO[4]	PTB_WAKE[4]	I2C5_SCL	TMR6	
PortB.5	PTB_GPIO[5]	PTB_WAKE[5]	I2C5_SDA	TMR6	
PortB.6	PTB_GPIO[6]	PTB_WAKE[6]	I2C7_SCL	TMR6	
PortB.7	PTB_GPIO[7]	PTB_WAKE[7]	I2C7_SDA	TMR6	
PortB.8	PTB_GPIO[8]	I2C2_SCL	TMR8		
PortB.9	PTB_GPIO[9]	I2C2_SDA	TMR8		
PortB.10	PTB_GPIO[10]	I2C3_SCL	TMR8		
PortB.11	PTB_GPIO[11]	I2C3_SDA	TMR8		
PortB.12	PTB_GPIO[12]	I2C4_SCL			
PortB.13	PTB_GPIO[13]	I2C4_SDA			
PortB.14	PTB_GPIO[14]	I2C0_SCL			
PortB.15	PTB_GPIO[15]	I2C0_SDA			
PortC.0	PTC_GPIO[0]	I2C6_SCL			
PortC.1	PTC_GPIO[1]	I2C6_SDA			
PortC.2	PTC_GPIO[2]				
PortC.3	PTC_GPIO[3]				
PortC.4	PTC_GPIO[4]	TMRB			
PortC.5	PTC_GPIO[5]	TMRB			
PortC.6	PTC_GPIO[6]	TMRB			
PortC.7	PTC_GPIO[7]	TMRB			
PortC.8	PTC_GPIO[8]	UART3_TX			
PortC.9	PTC_GPIO[9]	UART3_RX			
PortC.10	PTC_GPIO[10]	PWM6			
PortC.11	PTC_GPIO[11]	PWM7			
PortC.12	PTC_GPIO[12]	PWM8			
PortC.13	PTC_GPIO[13]	PWM9			
PortC.14	PTC_GPIO[14]	PWM10			

Pin Name	Default	1st Priority	2nd Priority	3rd Priority	Remap
PortC.15	PTC_GPIO[15]	PWM11			
PortD.0	PTD_GPIO[0]	PTD_WAKE[0]	TMR0		
PortD.1	PTD_GPIO[1]	PTD_WAKE[1]	TMR0		
PortD.2	PTD_GPIO[2]	PTD_WAKE[2]	TMR0		
PortD.3	PTD_GPIO[3]	PTD_WAKE[3]	TMR0		
PortD.4	PTD_GPIO[4]	PTD_WAKE[4]	SPI0_MISO		
PortD.5	PTD_GPIO[5]	PTD_WAKE[5]	SPI0_MOSI		
PortD.6	PTD_GPIO[6]	PTD_WAKE[6]	SPI0_SCK		
PortD.7	PTD_GPIO[7]	PTD_WAKE[7]	SPI0_NSS		
PortD.8	PTD_GPIO[8]	SPI1_MISO			
PortD.9	PTD_GPIO[9]	SPI1_MOSI			
PortD.10	PTD_GPIO[10]	SPI1_SCK			
PortD.11	PTD_GPIO[11]	SPI1_NSS			
PortD.12	PTD_GPIO[12]	SPI2_MISO	TMRA		
PortD.13	PTD_GPIO[13]	SPI2_MOSI	TMRA		
PortD.14	PTD_GPIO[14]	SPI2_SCK	TMRA		
PortD.15	PTD_GPIO[15]	SPI2_NSS	TMRA		
PortE.0	PTE_GPIO[0]	PTE_WAKE[0]	UART0_TX	TMR1	
PortE.1	PTE_GPIO[1]	PTE_WAKE[1]	UART0_RX	TMR1	
PortE.2	PTE_GPIO[2]	PTE_WAKE[2]	UART1_TX	TMR1	
PortE.3	PTE_GPIO[3]	PTE_WAKE[3]	UART1_RX	TMR1	
PortE.4	PTE_GPIO[4]	PTE_WAKE[4]	UART2_TX		
PortE.5	PTE_GPIO[5]	PTE_WAKE[5]	UART2_RX		
PortE.6	PTE_GPIO[6]	PTE_WAKE[6]	PWM0		
PortE.7	PTE_GPIO[7]	PTE_WAKE[7]	PWM1		
PortE.8	PTE_GPIO[8]	PWM2			
PortE.9	PTE_GPIO[9]	PWM3			
PortE.10	PTE_GPIO[10]	PWM4			
PortE.11	PTE_GPIO[11]	PWM5			
PortE.12	PTE_GPIO[12]	TMR3_mati0			
PortE.13	PTE_GPIO[13]	TMR3_mati1			
PortE.14	PTE_GPIO[14]	TMR3_capi0			
PortE.15	PTE_GPIO[15]	TMR3_capi1			
PortF.0	PTF_GPIO[0]	TMR5_mati0			
PortF.1	PTF_GPIO[1]	TMR5_mati1			
PortF.2	PTF_GPIO[2]	TMR5_capi0			
PortF.3	PTF_GPIO[3]	TMR5_capi1			
PortF.4	PTF_GPIO[4]				
PortF.5	PTF_GPIO[5]	EXT VREF			
PortF.6	LXIN	PTF_GPIO[9]			
PortF.7	LXOUT	PTF_GPIO[10]			
PortF.8	HXIN	PTF_GPIO[11]			
PortF.9	HXOUT	PTF_GPIO[12]			

Note:
(1) For those pins which are not used as analog input, be sure not to configure as analog input in the GPIO control registers.
(2) To utilize the functions in the remap column, the respective remap bit in the GPIO control registers must be set.

3.5 GPIO Configurations for Peripherals

TMR0/1/2/3/4/5/6/7/8/9/A/B

Timer Pinout	Configuration	GPIO Configuration
TMRx_capi0	Input capture channel x	Input floating
	Output compare channel x	Output push-pull
TMRx_capi1	Input capture channel x	Input floating
	Output compare channel x	Output push-pull
TMRx_mato0	Input capture channel x	Input floating
	Output compare channel x	Output push-pull
TMRx_mato1	Input capture channel x	Input floating
	Output compare channel x	Output push-pull

UART0/1/2/3

UART Pinout	Configuration	GPIO Configuration
UARTx_TX	Output open-drain	Output open-drain with pull-up
	Output push-pull	Output push-pull
UARTx_RX	Input channel x	Input with pull-up

SPI0/1/2/3

SPI Pinout	Configuration	GPIO Configuration
SPI0_NSS		Input floating / pull-up
SPI0_SCK	Master	Output push-pull
	Slave	Input floating / pull-up
SPI0_MISO	Master normal	Input floating
	Master bidirectional output enable	Input floating
	Master bidirectional output disable	Input floating
	Slave normal	Output push-pull
	Slave bidirectional output enable	Output push-pull
	Slave bidirectional output disable	Input floating
SPI0_MOSI	Master normal	Output push-pull
	Master bidirectional output enable	Output push-pull
	Master bidirectional output disable	Input floating
	Slave normal	Input floating
	Slave bidirectional output enable	Input floating
	Slave bidirectional output disable	Input floating

I2C0/2/3/4/5/6/7

I2C Pinout	Configuration	GPIO Configuration
I2C0_SCL	I2C clock	Open drain with pull-up function
I2C0_SDA	I2C data I/O	Open drain with pull-up function

PWM0/1/2/3/4/5/6/7/8/9/10/11/12

PWM Pinout	Configuration	GPIO Configuration
PWMx	PWM	Output push-pull
		Open-drain

3.6 Memory Mapping

Index	Function	Description
0x0000_0000~0x0002_FFFF	192K Flash Memory	ILM Bus
0x0003_0000~0x000F_FFFF	Reserved	
0x0010_0000~0x0010_0BFF	12K SRAM	DLM Bus
0x0010_0C00~0x001F_67FF	Reserved	
0x001F_6800~0x001F_6BFF	GPIO.A~F	
0x001F_6C00~0x001F_FFFF	Reserved	
0x0020_0000~0x0020_03FF	System Control	APB 0 Bus
0x0020_0400~0x0020_07FF	RTC	
0x0020_0800~0x0020_0BFF	WWDT	
0x0020_0C00~0x0020_0FFF	Wake Up & Interrupt	
0x0020_1000~0x0020_13FF	Timer0	
0x0020_1400~0x0020_17FF	Timer2	
0x0020_1800~0x0020_1BFF	Timer4	
0x0020_1C00~0x0020_1FFF	Timer6	
0x0020_2000~0x0020_23FF	Timer8	
0x0020_2400~0x0020_27FF	TimerA	
0x0020_2800~0x0020_2BFF	Reserved	
0x0020_2C00~0x0020_2FFF	Reserved	
0x0020_3000~0x0020_33FF	UART0	
0x0020_3400~0x0020_37FF	UART2	
0x0020_3800~0x0020_3BFF	SPI0	
0x0020_3C00~0x0020_3FFF	SPI2	
0x0020_4000~0x0020_43FF	I2C0	
0x0020_4400~0x0020_47FF	I2C2	
0x0020_4800~0x0020_4BFF	I2C4	
0x0020_4C00~0x0020_4FFF	I2C6	
0x0020_5000~0x0020_53FF	IR	
0x0020_5400~0x0020_57FF	Reserved	
0x0020_5800~0x0020_5BFF	Reserved	
0x0020_5C00~0x0020_63FF	Reserved	
0x0020_6000~0x0020_63FF	Reserved	
0x0020_6400~0x0020_67FF	Reserved	
0x0020_6800~0x0020_6BFF	Reserved	
0x0020_6C00~0x0020_6FFF	Reserved	
0x0020_7000~0x0020_73FF	Reserved	
0x0020_7400~0x0020_77FF	Reserved	
0x0020_7800~0x0020_7BFF	Reserved	
0x0020_7C00~0x0020_7FFF	Reserved	
0x0020_8000~0x0020_83FF	ADC	APB 1 Bus
0x0020_8400~0x0020_87FF	IWDT	
0x0020_8800~0x0020_8BFF	EEPROM Emulator	
0x0020_8C00~0x0020_8FFF	PWM	
0x0020_9000~0x0020_93FF	Timer1	
0x0020_9400~0x0020_97FF	Timer3	
0x0020_9800~0x0020_9BFF	Timer5	
0x0020_9C00~0x0020_9FFF	Timer7	
0x0020_A000~0x0020_A3FF	Timer9	

Index	Function	Description
0x0020_A400~0x0020_A7FF	TimerB	
0x0020_A800~0x0020_ABFF	Reserved	
0x0020_AC00~0x0020_AFFF	Reserved	
0x0020_B000~0x0020_B3FF	UART1	
0x0020_B400~0x0020_B7FF	UART3	
0x0020_B800~0x0020_BBFF	SPI1	
0x0020_BC00~0x0020_BFFF	SPI3	
0x0020_C000~0x0020_C3FF	I2C1	
0x0020_C400~0x0020_C7FF	I2C3	
0x0020_C800~0x0020_CBFF	I2C5	
0x0020_CC00~0x0020_CFFF	I2C7	
	Reserved	
0x0030_0000~0x0030_03FF	DMA	AHB-Lite Bus
0x0030_0400~0x003F_FFFF	Reserved	
0x0040_0000~0x00FF_FFFF	Reserved	

4. Functional Description

4.1 Block Diagram

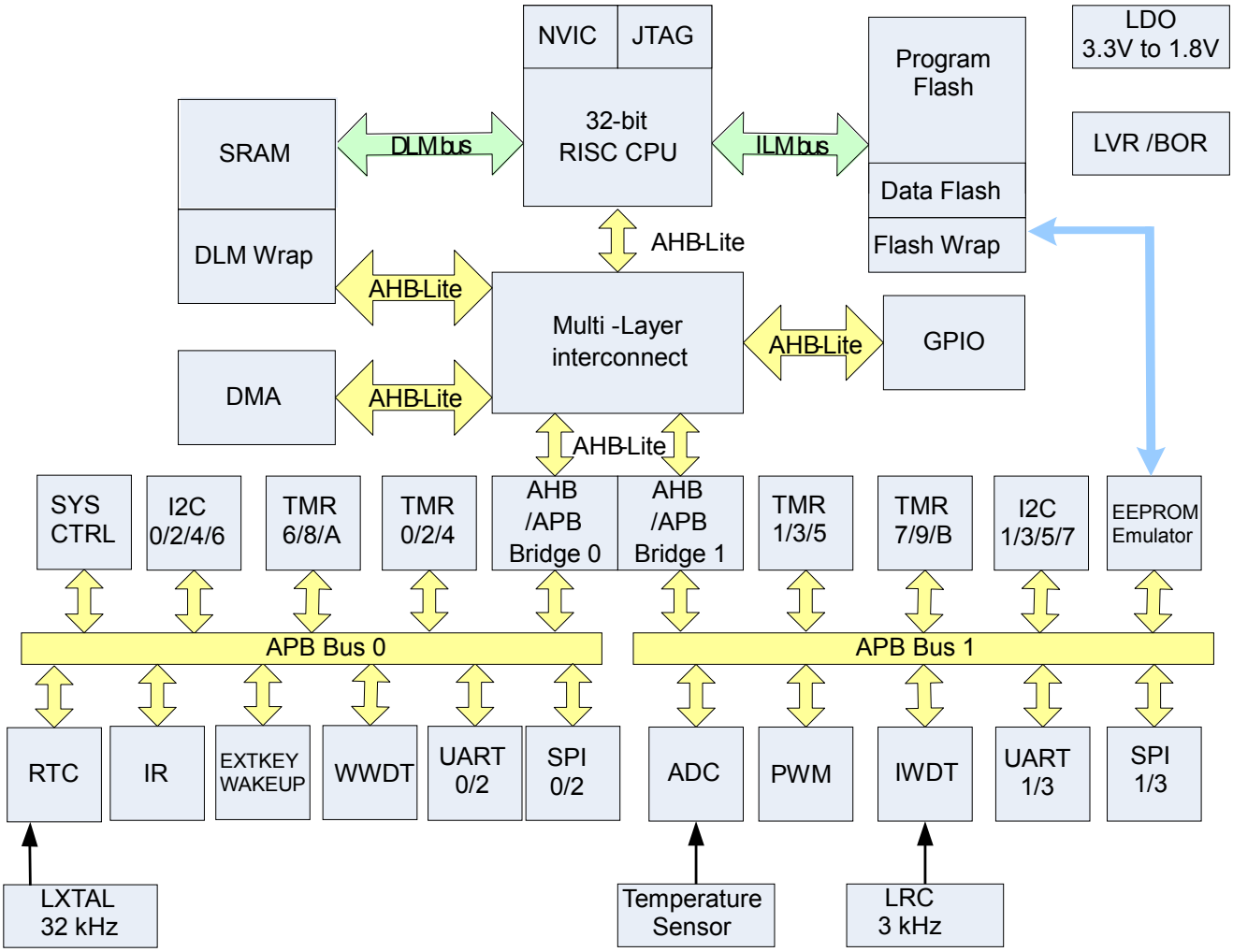


Figure 1: System block diagram

4.2 MCU

4.2.1 Internal MCU

Embedded 32-bit RISC core with 32 bit address and 32 bit data bus operates at up to 48 MHz.

4.2.2 Embedded Flash Memory

192K bytes of embedded flash memory located at 0x0000_0000~0x0002_FFFF is available for storing program.

Flash data read protection is implemented to protect customer code.

4.2.3 Embedded SRAM

12K bytes of embedded SRAM accessed at CPU clock speed with no wait states.
Its location is from 0x0010_0000 to 0x0010_2FFF.

4.3 System Reset

There are five reset sources for this controller. Fig.1 shows the block diagram of reset logic.

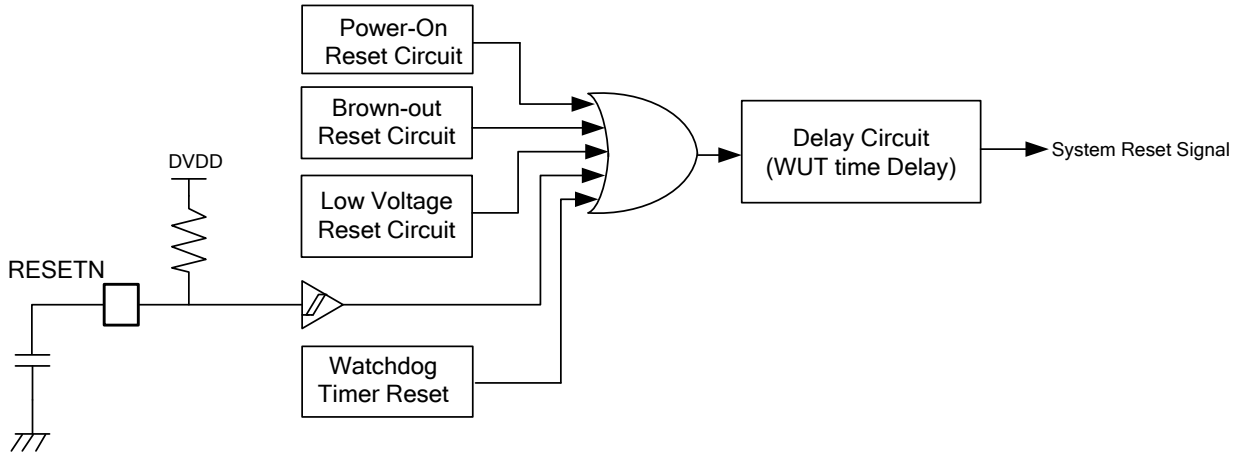


Figure 2: System reset block diagram

4.3.1 NRST

The NRST Reset happens when there is a low level on the RESET_N pin.

4.3.2 Low VDD Reset

The Low-VDD-Reset is generated when DVDD is below 1.5V.

4.3.3 Independent Watch Dog Timer Reset

The independent Watchdog-Timer-Reset happens when the independent watchdog timer is time out. Please refer to the independent watchdog timer section for more details.

4.3.4 Window Watch Dog Timer Reset

The window Watchdog-Timer-Reset happens when the window watchdog timer is time out. Please refer to the window watchdog timer section for more details.

All reset signals will last warm-up time (count delay via MCU clock), awaiting system stable.
Reset sources:
(1) RESETN pin = low (option (a) digital filter 16 clock (b) no digital filter)
(2) DVDD18 < 1. 5V reset (disabled by register LVR_ON)
(3) Independent Watchdog timer reset (disabled when operating at internal test modes)
(4) Window Watchdog timer reset (disabled when operating at internal test modes)

4.3.5 Power On Reset

A power on reset is generated when power on/power down reset.

4.4 Low Voltage Detection

The Low Voltage Detection monitors the DVDD power supply voltage. The detection voltage can be set from 2.0V~3V (8 stages) through software setting.

System Control Register: address range 0x0020_0000 ~ 0x0020_03FF

Index	Default	R/W	Bit	Name	Description
14_H	0	R	31:5		Reserved
	3'b0	R/W	4:2	LVD_VS	DVDD low voltage detection voltage select 000: 2.0V 001: 2.175V 010: 2.35V 011: 2.525V 100: 2.7V 101: 2.875V 110: 3.05V 011: 3.225V
	0	R/W	1	LVD_ON	LVD power control bit: 0: disable 1: enable
	1	R/W	0	LVR_ON	LVR power control bit: 1.7V active 0: disable 1: enable

4.5 Oscillator System Overview

The WT59F2C9 oscillator system has the following modules and features:

- Various external and internal oscillator options as clock sources
- An on-chip PLL to boost internal operating frequency
- A clock switching mechanism between various clock sources
- Programmable clock postscaler for system power savings.
- Configuration bits for main oscillator selection.

After initialization, default HSI is selected and clock configuration registers determine the clock source upon Power-on Reset (POR) and Low-voltage Reset (LVR). Thereafter, the clock source can be changed between permissible clock sources. The clock configuration registers controls the clock switching and reflects system clock related status bits.

4.5.1 Clock System

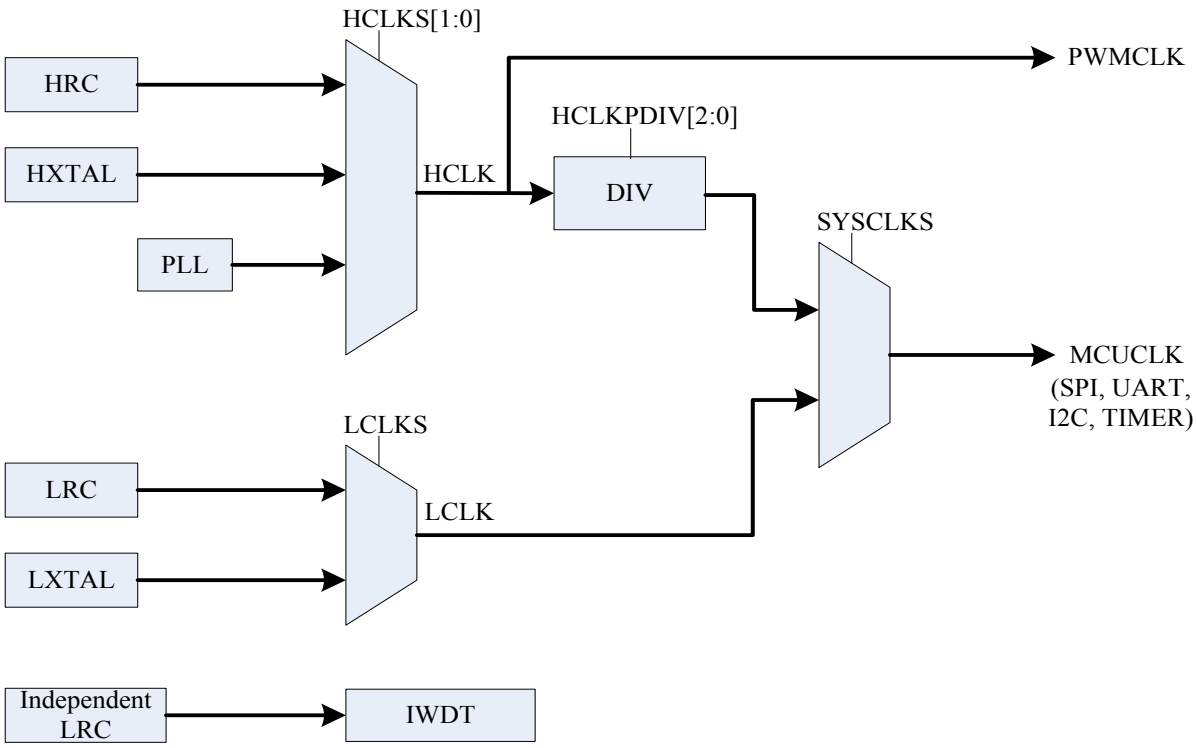


Figure 3: Clock system block diagram

The clock control circuit accepts clock selection setting of system page to select HRC, HXTAL or PLL clock source and clock source is as source of internal clock post divider to divide clock by M/N for flash clock and system clock respectively.

Note 1: PLL has to be turned off before change PLL configuration. Turn on PLL after finish PLL configuration, Then wait for PLL lock to switch to PLL clock.
Note 2: To get better performance, it's suggested to set HCLKPDIV[[6:4] in 0x00200004[10:8] to 0'b010 when switching system clock source from PLL to non-PLL or non-PLL to PLL output. User can change back to the intended system clock frequency setting after clock switching.

4.5.2 PLL

WT58F2C8/WT58F2C9 PLL consists of Phase Frequency Detector, Charge Pump, Loop Filter, Ring-Oscillator VCO and Programmable Frequency Divider.

Block Diagram

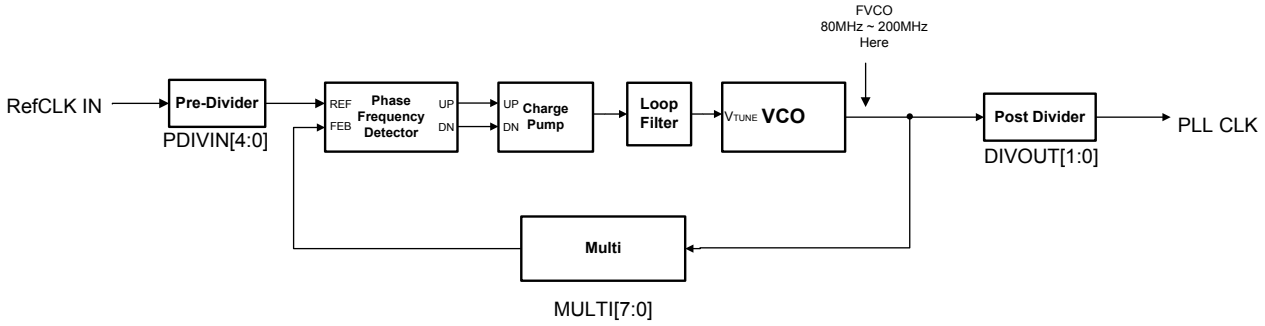


Figure 5: PLL Block Diagram

$$\text{PLL CLK Frequency} = \frac{\frac{F_{RefCLK}}{PDIVIN[4:0] + 1} \times (MULTI[7:0] + 1)}{2^{(\sim DIVOUT[1:0]+1)}}$$

$$\text{VCO Frequency } F_{VCO} = \frac{F_{RefCLK}}{PDIVIN[4:0] + 1} \times (MULTI[7:0] + 1)$$

VCO Frequency Range: 80MHz < F_{RefCLK} < 200 MHz

REGISTER

PLL register is included in the system register and quoted as below:

System Control Register: address range 0x0020_0000 ~ 0x0020_03FF

Index	Default	R/W	Bit	Name	Description
34	-	-	31:18	-	reserved
	'b00	R/W	17:16	PLL_REFCLKSEL	PLL Reference Clock select 00: HRCClk 01: HXTALClk 10: HClk 11: PADPORTB[0]

Index	Default	R/W	Bit	Name	Description
	‘b0000_0000	R/W	15:8	PLL_MULTI	PLL Multipler 0000_0000: x 1 0000_0001: x 2 0000_0010: x 3 . . 1111_1111: x 256
	‘b00	R/W	7:6	PLL_DIVOUT	PLL output clock pre-divider 00: F_{VCO} / 16 01: F_{VCO} / 8 10: F_{VCO} / 4 11: F_{VCO} / 2
	‘b0_0000	R/W	5:1	PLL_PDIVIN	PLL input clock pre-divider 0_0000: F_{RefCLK} / 1 0_0001: F_{RefCLK} / 2 0_0010: F_{RefCLK} / 3 . . 1_1111: F_{RefCLK} / 32
	‘b0	R/W	0	PLL_ON	PLL power control 0: disable PLL 1: enable PLL

4.5.3 High Speed External Crystal (HXTAL)

The range of high speed external crystal oscillates is from 1 MHz to 48 MHz. To fit different oscillating frequency, the “HXTAL_TYPE” register in system control register has to be properly set as below:
HXTAL Range Selection “HXTAL_TYPE” for HXTAL
00: 1M~12M
01: 12M ~ 24M
10: 12M ~ 36M
11: 12M ~ 48M

4.6 System Register

System Control Register: address range 0x0020_0000 ~ 0x0020_03FF

Index	Default	R/W	Bit	Name	Description
00			31:0		Reserved
04			31:24		Reserved
	‘b011	R/W	23:21	SYSCLK_Freq	System clock frequency indicator. To help hardware generate correct timing reference, software must set these registers based on the system clock. 000: 1 MHz 001: 3 MHz 010: 6 MHz 011: 12 MHz 100: 24 MHz 101: 36 MHz 110: reserved 111: 48 MHz
			20:17		
	0	R/W	16	FSCM_EN	Must be set as 0
	-	-	15:11	-	reserved
	‘b000	R/W	10:8	WARMUPTIME	Clock warm up time after Power-down wake up: 000: 2^{10} / F_{main} 001: 2^{9} / F_{main} 010: 2^{8} / F_{main} 011: 2^{6} / F_{main} 100: 2^{3} / F_{main} 101: 2^{11} / F_{main} 110: 2^{12} / F_{main} 111: 2^{13} / F_{main}
	-	-	7	-	reserved
	‘b000	R/W	6:4	HCLKPDIV	High Speed Clock Pre-divide 000: /1 001: /2 010: /4 011: /6 100: /8 101: /12 110: /16 111: /24
	0	R/W	3	LCLKS	Low Speed Clock Source Select: 0: LRC 1: LXTAL
	‘b00	R/W	2:1	HCLKS	High Speed Clock Source Select: 00: HRC 01: HXTAL 10: PLL 11: none Turn on the target clock source before change clock to target clock source.
	0	R/W	0	SYSCLKS	Main system Clock Select: 0: High speed clock 1: Low Speed clock

Index	Default	R/W	Bit	Name	Description
08	-	-	31	-	reserved
	0	R/W	30:24	HRC_TUNE48M	HRC tune for 48 MHz
	-	-	23	-	reserved
	0	R/W	22:16	HRC_TUNE24M	HRC tune for 36 MHz
	-	-	15	-	reserved
	0	R/W	14:8	HRC_TUNE12M	HRC tune for 24 MHz
	-	-	7		reserved
	0	R/W	6:0	HRC_TUNE12M	HRC tune for 12 MHz
0C	-	-	31:28	-	reserved
	‘b00	R/W	27:26	HRC_CLKSEL	HRC central clock frequency domain select: 00: 12 MHz 01: 24 MHz 10: 36 MHz 11: 48 MHz
	1	R/W	25	HRC_OUT	HRC output enable control 0: disable output clock 1: enable output clock
	1	R/W	24	HRC_ON	HRC enable power control 0: disable HRC 1: enable HRC
			23:17		
	1	R/W	16	LRC_ON	LRC power control: 0: disable LXTAL 1: enable LXTAL
	-	-	15:11	-	reserved
	‘b01	R/W	10:9	LXTAL_GAIN	LXTAL circuit current 00: 3uA 01: 6uA (default) 10: 9uA 11: 12uA
	1	R/W	8	LXTAL_ON	LXTAL power control: 0: disable LXTAL 1: enable LXTAL
	-	-	7:4	-	reserved
	‘b0	R/W	3	HXTAL_ENRES	HXTAL external apply circuit with internal resistor 0: disable internal resistor 1: enable internal resistor
	‘b00	R/W	2:1	HXTAL_TYPE	HXTAL central frequency domain select: 00: 1M~12M 01: 12M ~ 24M 10: 12M ~ 36M 11: 12M ~ 48M
	1	R/W	0	HXTAL_ON	HXTAL power control bit: 0: disable HXTAL 1: enable HXTAL
10	-	-	31:4	-	reserved
	0	R/W	3	WAIT_EN	Flash access wait enable, (don’t set FLASH_PS_EN[1]) for 24 MHz < sys_clk < 48 MHz, set this bit to “1”

Index	Default	R/W	Bit	Name	Description
	0	R/W	2	FLASH_PS_EN[2]	Flash Power Saving Enable[2] (for high Speed) 0: disable flash power saving 1: enable flash power saving
	0	R/W	1	FLASH_PS_EN[1]	Flash Power Saving Enable[1] (for Low Speed)(WAIT_EN = 0) 0: disable flash power saving 1: enable flash power saving
	0	R/W	0	FLASH_PS_EN[0]	Flash Power Saving Enable[0] (for low Speed) 0: disable flash power saving 1: enable flash power saving This function only applied for system clock is 12 MHz or under 12 MHz only. For system clock faster than 12 MHz, don't enable this function.
18	0	W	31	CLR_RST_FLAG	Clear all reset flag (clear once after startup) 0: no action 1: clear all reset flag
	'h0	R	30:16		Reserved
	0	R	15	WWDTRSTF	WWDT reset flag
	0	R	14	IWDTRSTF	IWDT reset flag
	X	R	13	EXTPINRSTF	External pin reset flag
	X	R	12	LVRRSTF	LVR reset flag
	X	R	11	BOR33RSTF	BOR33 reset flag
	X	R	10	POR33RSTF	POR33 reset flag
	0	R/W	9	RST_FIL_ON[2]	LVR 0: "LVR" no digital filter 1: "LVR" havedigital filter Reset by POR
	0	R/W	8	RST_FIL_ON[1]	LVD 0: "LVD" no digital filter 1: "LVD" havedigital filter Reset by POR
	0	R/W	7	RST_FIL_ON[0]	External reset filter control 0: "PADRESET_n" pin no digital filter 1: "PADRESET_n" pin havedigital filter Reset by POR
	-	-	6	-	reserved
		R	5	LVDF	LVD output flag
	3'b000	R/W	4:2	LVD_VS	DVDD low voltage detection voltage select 000: 2.0V 001: 2.175V 010: 2.35V 011: 2.525V 100: 2.7V 101: 2.875V 110: 3.05V 111: 3.225V
	0	R/W	1	LVD_ON	LVD power control bit: 0: disable 1: enable
	1	R/W	0	LVR_ON	LVR power control bit: 0: disable 1: enable
1C	-	-	31:12	-	Reserved

Index	Default	R/W	Bit	Name	Description
	'h7FF	R	13:3	RC_CAL_CONT	RC counter calibrate with external clock
	0	R	2	RC_CAL_OK	1: RC Calibration done. Write "0" to "RC_CAL_EN" will clear this flag
	0	R/W	1		Calculation Clock Select 0: MCU Clock 1: HRCClk
	0	R/W	0	RC_CAL_EN	RC calibration enable 1: Enable(Start) RC Calibration 0: Disable RC Calibration Write "1" to start RC Calibration. Write "0" to reset the RC Calibration
34	-	-	31:18	-	reserved
	'b00	R/W	17:16		PLL Reference Clock select 00: HRCClk 01: HXTALClk 10: HClk 11: PADPORTB[0]
	'b0	R/W	15:8		PLL Multipler 0000_0000
	'b0	R/W	7:6		PLL output clock pre-divider
	'b0	R/W	5:1		PLL input clock pre-divider 0000:
	'b0	R/W	0	PLL_ON	PLL power control 0: disable PLL 1: enable PLL

Note-1: Index and default are heximal, and bits are decimal.

Note-2: Instruction "standby 1" should be appended after setting register bit to activate all clocks off function.

Note-3: Target clock source should be stable before clock switch; it needs 5ms for HXTAL, 20us for HRC after clock source is turned on.

4.7 Independent Watchdog Timer

4.7.1 Introduction

The independent watchdog (IWDT) is clocked by its own dedicated low-speed clock (LRC 32 kHz) and thus stays active even if the main clock fails. The IWDT is best suited to applications which require the watchdog to run as a totally independent process outside the main application, but have lower timing accuracy constraints.

4.7.2 Main Features

- Free-running Down counter
- Clocked from an independent RC oscillator (RC 32 KHz)
- Reset (if watchdog activated) when the down counter value of 0x000 is reached

4.7.3 Functional Description

When IWDT is started by writing the value 0xCCCC in the KEY register (IWDT_KEY), the counter starts counting down from the reset value of 0xFFF. When it reached the end of count value (0x000) a reset signal is generated (IWDT Reset). Whenever the key value 0xAAAA is written in the IWDT_KEY register, the RELOAD_CNT value is reloaded in the counter and watchdog reset is prevented. Write access to the PRE_SCALER and RELOAD_CNT registers is protected. To modify them, the code 0x5555 in the IWDT_KEY must first be written. A write access to this register with a different value will break the sequence and register access will be protected again. This implies that it is the case of the reload operation (writing 0xAAAA). Status registers (RELOAD_RVU and PRESCL_PVU) are available to indicate that an update of the pre-scaler or the down-counter reload value is on going. Updating the pre-scaler and down counter value can only work when the RELOAD_WORK bit is “0”; writing new value to PRE_SCALER and RELOAD_CNT when RELOAD_WORK is “1” may cause unexpected result.

4.7.4 Register Definition

IWDT Register: address range 0x0020_8400 ~ 0x0020_87FF

Index	Default	R/W	Bit	Name	Description
00	00	R/W	31:16	Reserved	
		W	15:0	IWDT_KEY[15:0]	16’hCCCC: Enable IWDT (Reset Counter Start) 16’hAAAA: Reset counter reload and pre-scaler reset 16’h5555: Enable assess pre-scaler and reset counter registers (lock open)
04	00	R/W	31:8	Reserved	
		R	7	EN_IWDT	Flag to indicate IWDT is activated. 0: IWDT disable. 1: IWDT enable.
		R/W	6:5	Reserved	
		R	4	RELOAD_WORK	1: Previous reload process is working, invalid for next reloading reset counter 0: Ready for next reloading reset counter
		W	3	RST_CLR	IWDT Reset event clear, write “1” to clear RESET_FLAG
		R	2	RESET_FLAG	IWDT Reset event flag
		R	1	RELOAD_RVU	Reload reset counter value program access
		R	0	PRESCL_PVU	Pre-scaler counter value program access

Index	Default	R/W	Bit	Name	Description
08	00	R/W	31:3	Reserved	
			2:0	PRE_SCALER[2:0]	000: divider / 4 100: divider / 64 001: divider / 8 101: divider / 128 010: divider / 16 110: divider / 256 011: divider / 32 111: divider / 256
0C	FFF	R/W	31:14	Reserved	
			13:0	RELOAD_CNT[13:0]	Down counter reload value

4.7.5 Block Diagram

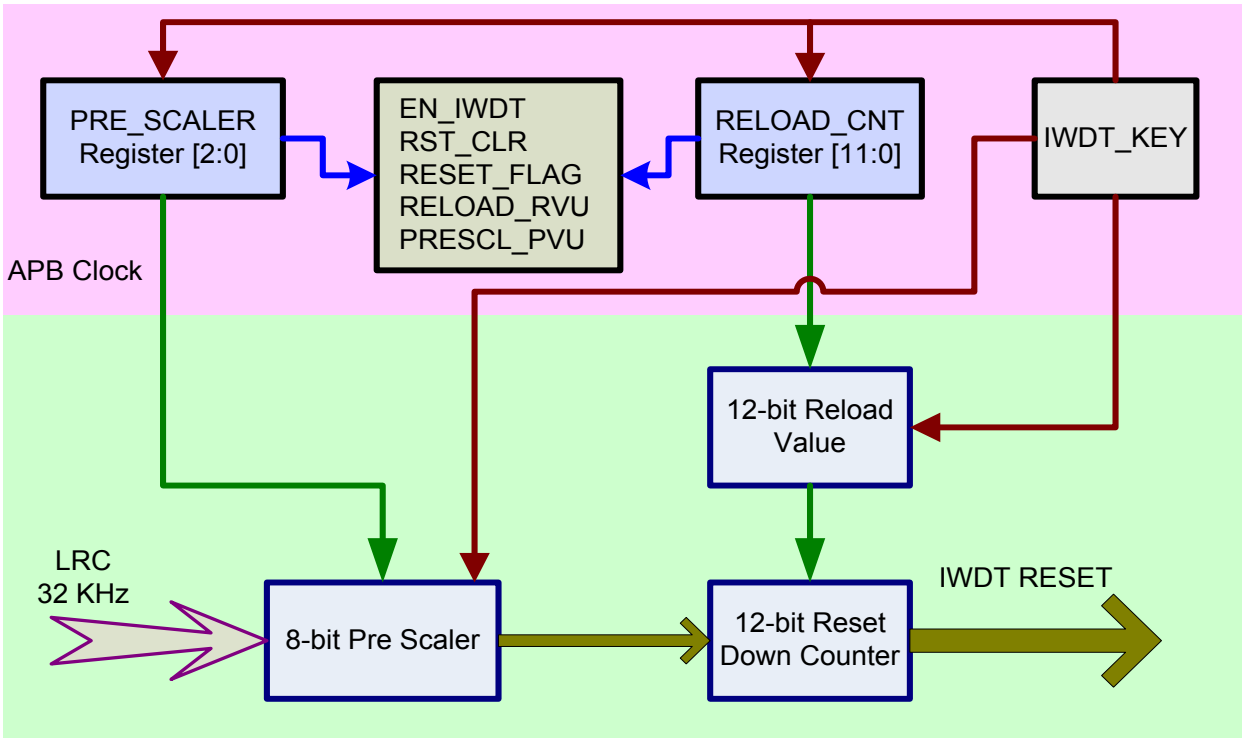


Figure 4: IWDT block diagram

Min./Max. Timeout period at 32 KHz (LRC)

Prescaler Divider	PRE_SCALER[2:0]	Minimum Timeout (ms) RELOAD_CNT[13:0] = 0x0000	Maximum Timeout (ms) RELOAD_CNT[13:0] = 0x3FFF
/ 4	0	0.122	500
/ 8	1	0.244	1000
/ 16	2	0.488	2000
/ 32	3	0.976	4000
/ 64	4	1.953	8000
/ 128	5	3.906	16000
/ 256	6 or 7	7.813	32000

IWDT_KEY: This write-only register must be written by software at regular intervals with the key Value AAAAh, otherwise the watchdog generates a reset when the reset counter reaches 0. To reload reset counter, it takes 5 LSI clock cycles. If the previous reload access does not complete then reload again, it takes two more LSI clock cycle for next valid reload. After system wake up, two more LSI clock cycle is also requirement for next valid reload access. Writing the key value 5555h to enable access to the PRE_SCALER and RELOAD_CNT registers. Writing the key value CCCCh starts the watchdog.

PRE_SCALER: This register is write-access protected by IWDT_KEY = 5555h lock open.
It is written by software to select the pre-scaler divider feeding the counter clock. Writing this register takes 3 LSI clock cycles to reset pre-scaler counter. Therefore, this value may not be up to date or valid if a write operation to this register is ongoing. For this reason the value read from this register is valid only when the PRESCL_PVU bit is low. If PRESCL_PVU bit is high, pre-scaler writing access is not completed. Reading this register for a previous writing value.

RELOAD_CNT: This register is write-access protected by IWDT_KEY = 5555h lock open.
Writing this register takes 3 LSI clock cycles to reload reset counter. Therefore, this value may not be up to date or valid if a write operation to this register is ongoing. For this reason the value read from this register is valid only when the RELOAD_RVU bit is low. If RELOAD_RVU bit is high, reload counter writing access is not completed. Reading this register for a previous writing value.

For normal register setting, first write IWDT_KEY = 5555h register lock open. Then writing PRE_SCALER and RELOAD_CNT for the requirement of regular reset timing interval. A reset counter reload access is necessary IWDT_KEY = AAAAh for counter up-date to reload value and start new reset timing interval.

4.8 Window Watchdog Timer

The window watchdog timer (WWDT) is used to detect the occurrence of a software fault which causes the application program to abandon its normal sequence. The WWDT generates an MCU reset, unless the program refreshes the down counter before its bit-6 cleared. An MCU reset is also generated if the down counter is refreshed before it has reached the window register value. This means that the down counter must be refreshed in a limited window.

4.8.1 Features

- Programmable free-running down counter
- Conditional reset:
  - Reset (if WWDT enable) when the down counter value becomes less than 40h
  - Reset (if WWDT enable) if the down counter is reloaded outside the window
- Early wakeup interrupt (EWI): triggered (if WWDT enable) when the down counter is equal to 40h. This function can be used to reload the counter and prevent WWDT reset.

4.8.2 Functions

If WWDT is enabled and when the down counter rolls over 0x40h to 0x3Fh, it generates a MCU reset. If the software reloads the counter while the value is greater than the value stored in the window register, then a reset is generated. The application program must write in the DOWN_CNT_TIME register at regular intervals during normal operation to prevent an MCU reset. This operation must occur only when the down counter value is lower than window register value (WINDOW_VALUE). Therefore, the value to be stored in the 0x00 (WWDT_EN and DOWN_CNT_TIME) register must be between 0xFFh and 0xC0h.

- Enable WWDT: The Window watch dog timer is always disabled after reset. It is enabled by setting the (WWDT_EN = 1) register. It will not be disabled again once it's enabled except by a reset. Down counter control: When WWDT is enabled, the T6 bit (DOWN_CNT_TIME [6]) must be set to prevent generating an immediate reset. The T[5:0] bits contain the number of increments which represents the time delay before the WWDT produces a reset. The timing varies between a minimum and a maximum value due to the unknown status of pre-scaler when writing to the TIME_BASE register.

The WINDOW_VALUE register contains the high limit of the window. To prevent a reset, the down counter must be reloaded when its value is lower than the window register value and greater than 0x3Fh after setting WWDT enable (WWDT_EN = 1).

Another way to reload the counter is to use the early interrupt (EARLY_INT_EN). When the down counter reached the value 0x40h, this interrupt is generated and the corresponding interrupt service routine can be used to reload the down counter to prevent WWDT reset. The interrupt is cleared by setting (EARLY_INT_CLR) register. A software reset also can be generated by clearing T6 bit (DOWN_CNT_TIME [6] = 0) when WWDT is active (WWDT_EN = High).

4.8.3 Block Diagram

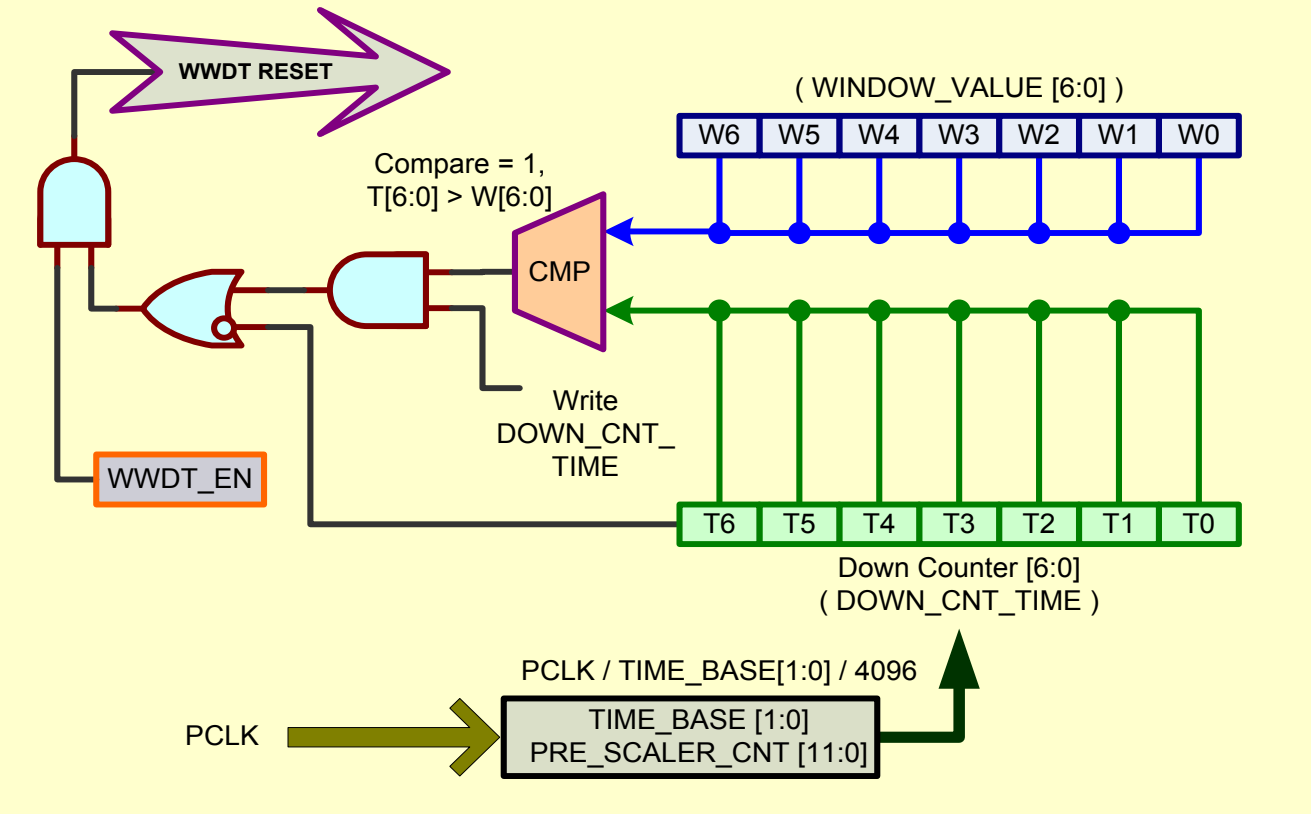


Figure 5: WWDT block diagram

4.8.4 Program the watchdog timeout

The timing diagram of WWDT is depicted as Fig. 8. Below is the formula to calculate the time out value:

$$T_{WWDT} = T_{PCLK} \times 4096 \times 2^{TIME\_BASE[1:0]} \times (T[5:0] + 1) \quad (ms)$$

Min./Max. Timeout value at 48.00 MHz (PCLK)

TIME_BASE[1:0]	Minimum Timeout (us) DOWN_CNT_TIME[6:0] = 0x40h	Maximum Timeout (ms) DOWN_CNT_TIME[6:0] = 0x7Fh
00 (/ 1)	85.33	5.46
01 (/ 2)	170.67	10.92
10 (/ 4)	341.33	21.85
11 (/ 8)	682.67	43.69

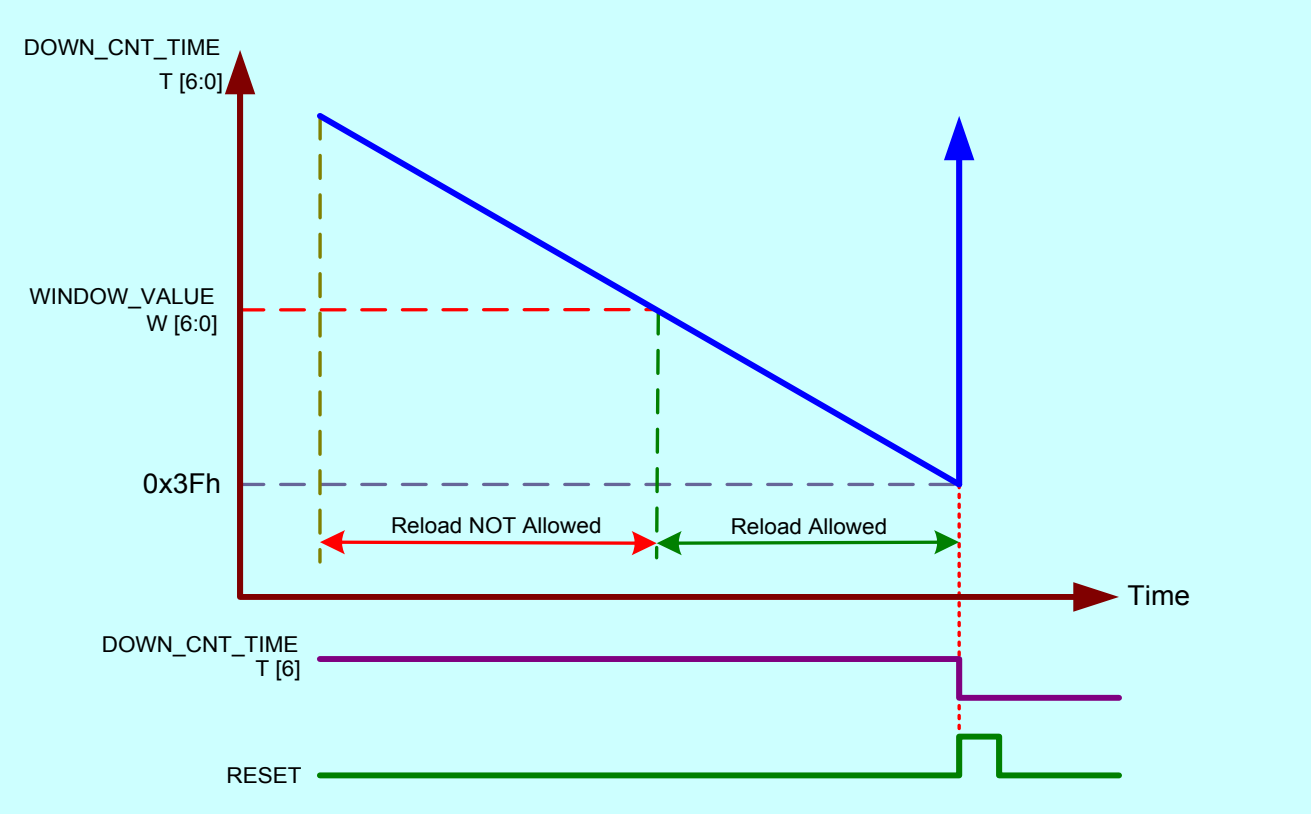


Figure 6: WWDT timing diagram

4.8.5 Register Definition

WWDT Register: address range 0x0020_0800 ~ 0x0020_0BFF

Index	Default	R/W	Bit	Name	Description
00	h'0	R/W	31:8	Reserved	
	0	R/W	7	WWDT_EN	Write "1" only, can only be cleared by reset 1: Enable WWDT (When Reset, WWDT disable) 0: Disable WWDT
	h'7F	R/W	6:0	DOWN_CNT_TIME	Reset down counter Reset active (DOWN_CNT_TIME = 3Fh)
04	h'0	R/W	31:10	Reserved	
		R/W	9	EARLY_INT_EN	Early interrupt enable 1: enable early wwdt interrupt 0: disable early wwdt interrupt
	0	R/W	8:7	TIME_BASE	PCLK divided by 4096 and divided by following number once 00: divided by 1, 10: divided by 4 01: divided by 2, 11: divided by 8
	h'7F	R/W	6:0	WINDOW_VALUE	Window value to be compared to reset down counter
08	h'0	R/W	31:2	Reserved	
		W	1	EARLY_INT_CLR	Clear early interrupt flag Write "1" to clear EARLY_INT_FLAG
		R	0	EARLY_INT_FLAG	Early interrupt flag (Reset down counter = 40h)

4.9 Flash Programmer

4.9.1 Register Definition

Flash Programmer: address range 0x0020_8800 ~ 0x0020_8BFF

Index	Default	R/W	Bit	Name	Description
00			31-0		Reserved
04			31-0		Reserved
08			31-0		Reserved
0c			31-0		Reserved
10			31-0		Reserved
14			31-0		Reserved
18			31-0		Reserved
1c			31-0		Reserved
40		R	31:4		Reserved
	4'h0	W	3-0	FP_EN0	FP function enable data1 (0x0A)
44		R	31:4		Reserved
	4'h0	W	3-0	FP_EN1	FP function enable data2 (0x05)
48		R	31:16		Reserved
	16'h0000	R/W	15-0	FP_ADDR	FP address
4c		R	31:10		Reserved
			9		Reserved : This bit must be kept as "0" for normal operation
		W	8	FP_MAS1	FP mass erase (auto clear)
		W	7	FP_READ	FP read (auto clear)
					Reserved
		W	5	FP_ERASE	FP erase (auto clear)
		W	4	FP_PROG	FP program (auto clear)
			3:0		Reserved
50	32'h00000000	W	31-0	FP_DATA	FP_DATA[31:0], single mode program data
54	26'h0		31:6		Reserved
	1'b0	R/W	5		Enable MCU INT broken Flash Program / Erase 0: disable (default) 1: enable
	1'b1	-	4	Reserved	1: must be set to "1"
	1'b0	W	3	prog_done	This bit will be set 1 after each erase, program, read, verify or mass erase. Write 1 to clear it.
			2		Reserved
58			1		Reserved: This bit must be kept as "0" for normal operation
	1'b1	R/W	0	FP_wakeup_en	0: mcu will stop to fetch instruction before flash is programmed and start to fetch instruction after programming is finished.
	25'b0	R	31-7		reserved
	1'b0	R/W	6	FP_IFREN	Must be set as 0
	2'b0	R	5-4		reserved

Index	Default	R/W	Bit	Name	Description
5c 60	4'h8	R/W	3-0	FP_TCTL[3:0]	FP Erase/Program Timing When erase flash, FP_TCTL = 4'h5 When program flash, FP_TCTL = 4'h7
	32'h00000000	R	31:0	FP_RDATA	DATA[31:0] read from FP
	x		31:4		reserved
	1'b1	W	3	Erase_breaken	1: enable the erase_break_flag be generated when MCU interrupt in flash erase process

Note 1: When data protection mode, only address range 0x7C00~0xBFFF is allowed to R/W by program.
Note 2: Erase_breaken and Prog_breaken is read only and they will be read back 0.

4.9.2 Data Protection

	no mass erase and data is protected	mass erase or data is not protected
Main block 0~ EERSTRADD-1	X	R/W
Main block EERSTRADD ~0x1fff	R/W	R/W

4.10 DM A Controller

4.10.1 Features

- Compliant with AMBA v2.0
  - AHB slave interface for DMA controller configuration
  - AHB master interface for data transfer
  - Transfer type – Single mode
  - 32-bits (word), 16-bits (half-word), 8-bits (byte) wide data transaction
- 7 configurable DMA channels
  - Support memory-to-peripheral transfer
  - Support peripheral-to-memory transfer
  - Support peripheral-to-peripheral transfer
- Peripherals supported
  - 45 sets requests/acknowledges hardware handshake
  - Timers (dma_req/ack[11:0])
  - UARTs(RX/TX) (dma_req/ack[19:12] for UART),
  - SPI(Tx/Rx) (dma_req/ack[27:20])
  - IICs(RX/TX) (dma_req/ack[43:28])
  - ADC (dma_req/ack[44])
- Arbitration scheme
  - Round-robin arbitration
  - Configurable 4 level priority
- Circular mode

4.10.2 Functional Description

The Direct Memory Access Controller is designed to enhance the system performance and reduce the processor-interrupt generation frequency. There are 7 configurable channels for memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral transfers. Each channel is connected to dedicated hardware handshake signal.

Block Diagram

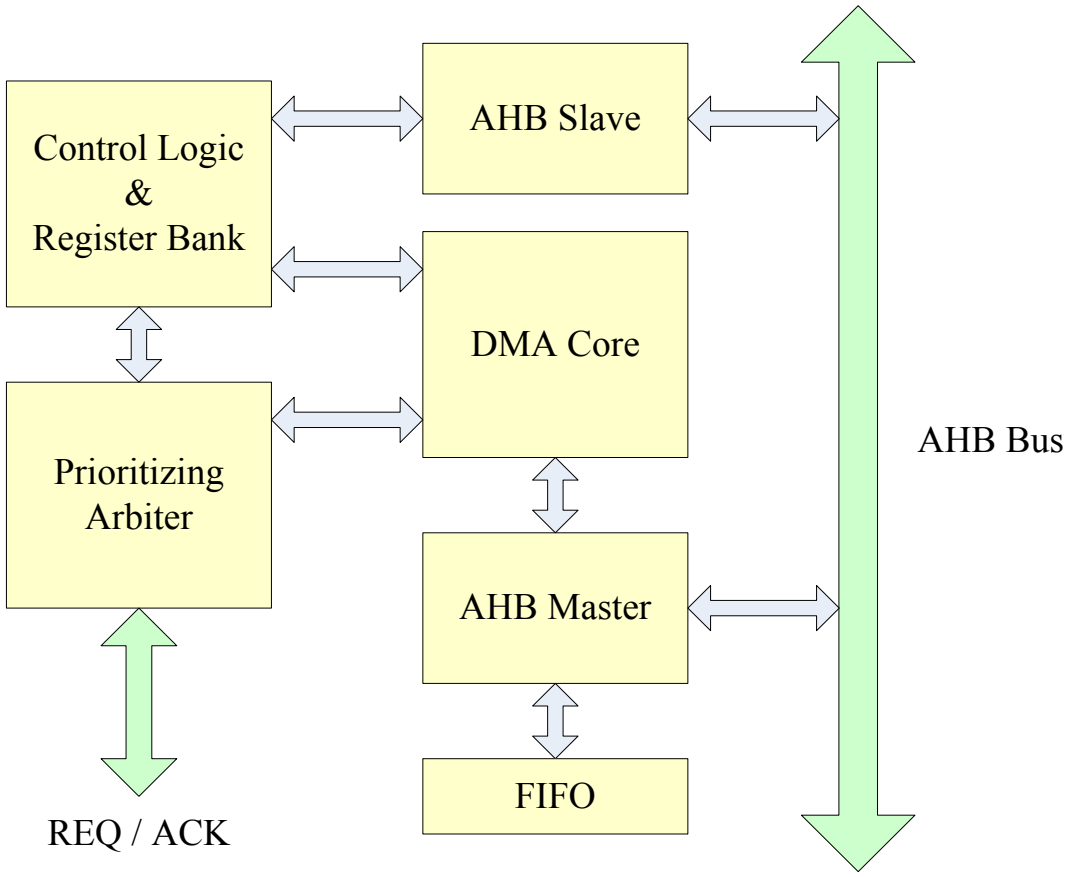


Figure 7: DMA block diagram

AHB Master Interface

The system can transfer data on AHB bus through this AHB master interface.

AHB Slave Interface

The system can configure the DMA controller or access the devices on AHB bus through this AHB slave interface.

FIFO Buffer

The FIFO buffer provides the data transfer buffer between the source and the destination.

DMA Core

The DMA data transfer engine.

Prioritizing Arbiter

To handle hardware handshake signals to start DMA transfer and configure up to 7-channels. They can group the round-robin arbitration scheme into 4 priority levels.

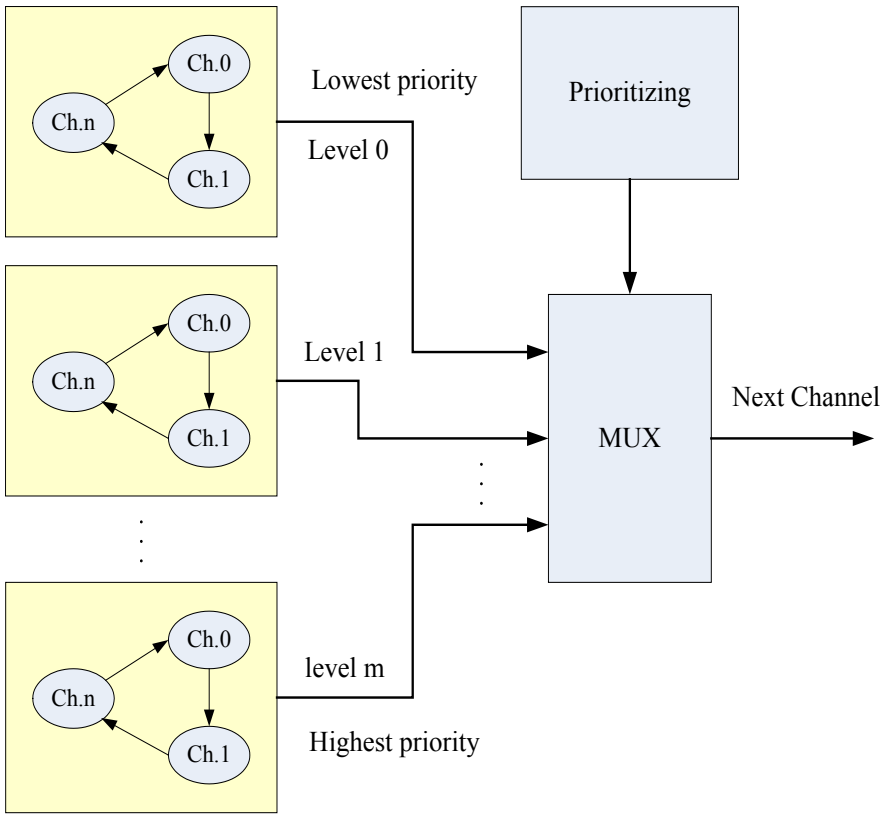


Figure 8: DMA Arbiter block diagram

Control Logic & Register Bank

Register set from AHB slave interface and generate some control logic for DMA transfer.

4.10.3 Register Definition

Global Setting for Interrupt Status/Clear & Channel Busy

DMA: address range 0x0030_0000 ~ 0x0030_03FF

Index	Default	R/W	Bit	Name	Description
04			31:7		Reserved
	0	RO	6	dma_busy6	1: Channel 6 busy 0: Channel 6 available
	0	RO	5	dma_busy5	1: Channel 5 busy 0: Channel 5 available
	0	RO	4	dma_busy4	1: Channel 4 busy 0: Channel 4 available
	0	RO	3	dma_busy3	1: Channel 3 busy 0: Channel 3 available
	0	RO	2	dma_busy2	1: Channel 2 busy 0: Channel 2 available
	0	RO	1	dma_busy1	1: Channel 1 busy 0: Channel 1 available
	0	RO	0	dma_busy0	1: Channel 0 busy 0: Channel 0 available
08			31:28		Reserved
	0	RO	27	err_int6	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	26	half_int6	0: No half_int event on channel 6 1: A half_int event on channel 6
	0	RO	25	total_int6	0: No total_int event on channel 6 1: A total_int event on channel 6
	0	RO	24	dma_int6	0: No half_int or total_int event on channel 6 1: An err_int6 or half_int6 or total_int6 event on channel 6
	0	RO	23	err_int5	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	22	half_int5	0: No half_int event on channel 5 1: A half_int event on channel 5
	0	RO	21	total_int5	0: No total_int event on channel 5 1: A total_int event on channel 5
	0	RO	20	dma_int5	0: No half_int or total_int event on channel 5 1: An err_int5 or half_int5 or total_int5 event on channel 5
	0	RO	19	err_int4	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	18	half_int4	0: No half_int event on channel 4 1: A half_int event on channel 4
	0	RO	17	total_int4	0: No total_int event on channel 4 1: A total_int event on channel 4
	0	RO	16	dma_Int4	0: No half_int or total_int event on channel 4 1: An err_int4 or half_int4 or total_int4 event on channel 4

Index	Default	R/W	Bit	Name	Description
	0	RO	15	err_int3	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	14	half_int3	0: No half_int event on channel 3 1: A half_int event on channel 3
	0	RO	13	total_int3	0: No total_int event on channel 3 1: A total_int event on channel 3
	0	RO	12	dma_int3	0: No half_int or total_int event on channel 3 1: An err_int3 or half_int3 or total_int3 event on channel 3
	0	RO	11	err_int2	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	10	half_int2	0: No half_int event on channel 2 1: A half_int event on channel 2
	0	RO	9	total_int2	0: No total_int event on channel 2 1: A total_int event on channel 2
	0	RO	8	dma_int2	0: No half_int or total_int event on channel 2 1: An err_int2 or half_int2 or total_int2 event on channel 2
	0	RO	7	err_int1	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	6	half_int1	0: No half_int event on channel 1 1: A half_int event on channel 1
	0	RO	5	total_int1	0: No total_int event on channel 1 1: A total_int event on channel 1
	0	RO	4	dma_int1	0: No half_int or total_int event on channel 1 1: An err_int1 or half_int1 or total_int1 event on channel 1
	0	RO	3	err_int0	Set by hardware, clear by software 0: No transfer error event 1: Transfer error
	0	RO	2	half_int0	0: No half_int event on channel 0 1: A half_int event on channel 0
	0	RO	1	total_int0	0: No total_int event on channel 0 1: A total_int event on channel 0
	0	RO	0	dma_int0	0: No half_int or total_int event on channel 0 1: An err_int0 or half_int0 or total_int0 event on channel 0
0C			31:28		Reserved
	0	WO	27	clr_err_int6	0: No effect 1: Clear err_int6 flag
	0	WO	26	clr_half_int6	0: No effect 1: Clear half_int6 flag
	0	WO	25	clr_total_int6	0: No effect 1: Clear total_int6 flag
	0	WO	24	clr_int6	0: No effect 1: Clear dma_int6, err_int6, total_int6 and half_int6 flags
	0	WO	23	clr_err_int5	0: No effect 1: Clear err_int5 flag
	0	WO	22	clr_half_int5	0: No effect 1: Clear half_int5 flag

Index	Default	R/W	Bit	Name	Description
	0	WO	21	clr_total_int5	0: No effect 1: Clear total_int5 flag
	0	WO	20	clr_int5	0: No effect 1: Clear dma_int5, err_int5, total_int5 and half_int5 flags
	0	WO	19	clr_err_int4	0: No effect 1: Clear err_int4 flag
	0	WO	18	clr_half_int4	0: No effect 1: Clear half_int4 flag
	0	WO	17	clr_total_int4	0: No effect 1: Clear total_int4 flag
	0	WO	16	clr_int4	0: No effect 1: Clear dma_int4, err_int4, total_int4 and half_int4 flags
	0	WO	15	clr_err_int3	0: No effect 1: Clear err_int3 flag
	0	WO	14	clr_half_int3	0: No effect 1: Clear half_int3 flag
	0	WO	13	clr_total_int3	0: No effect 1: Clear total_int3 flag
	0	WO	12	clr_int3	0: No effect 1: Clear dma_int3, err_int3, total_int3 and half_int3 flags
	0	WO	11	clr_err_int2	0: No effect 1: Clear err_int2 flag
	0	WO	10	clr_half_int2	0: No effect 1: Clear half_int2 flag
	0	WO	9	clr_total_int2	0: No effect 1: Clear total_int2 flag
	0	WO	8	clr_int2	0: No effect 1: Clear dma_int2, err_int2, total_int2 and half_int2 flags
	0	WO	7	clr_err_int1	0: No effect 1: Clear err_int1 flag
	0	WO	6	clr_half_int1	0: No effect 1: Clear half_int1 flag
	0	WO	5	clr_total_int1	0: No effect 1: Clear total_int1 flag
	0	WO	4	clr_int1	0: No effect 1: Clear dma_int1, err_int1, total_int1 and half_int1 flags
	0	WO	3	clr_err_int0	0: No effect 1: Clear err_int0 flag
	0	WO	2	clr_half_int0	0: No effect 1: Clear half_int0 flag
	0	WO	1	clr_total_int0	0: No effect 1: Clear total_int0 flag
	0	WO	0	clr_int0	0: No effect 1: Clears dma_int0, err_int0, total_int0 and half_int0 flags

DMA Channel X Source Register

Base Address: 0x0030_0000
X = 0 ~ 6, where x is channel number
Channel X Address index: 0x10 + 0x10 * (channel number X)

Index	Default	R/W	Bit	Name	Description
10	00	R/W	23:0	dma_saddr0	Channel 0 source address
20	00	R/W	23:0	dma_saddr1	Channel 1 source address
30	00	R/W	23:0	dma_saddr2	Channel 2 source address
40	00	R/W	23:0	dma_saddr3	Channel 3 source address
50	00	R/W	23:0	dma_saddr4	Channel 4 source address
60	00	R/W	23:0	dma_saddr5	Channel 5 source address
70	00	R/W	23:0	dma_saddr6	Channel 6 source address

DMA Channel X Destination Register

Base Address: 0x0030_0000
X = 0 ~ 6, where x is channel number
Channel X Address index: 0x14 + 0x10 * (channel number X)

Index	Default	R/W	Bit	Name	Description
14	00	R/W	23:0	dma_daddr0	Channel 0 destination address
24	00	R/W	23:0	dma_daddr1	Channel 1 destination address
34	00	R/W	23:0	dma_daddr2	Channel 2 destination address
44	00	R/W	23:0	dma_daddr3	Channel 3 destination address
54	00	R/W	23:0	dma_daddr4	Channel 4 destination address
64	00	R/W	23:0	dma_daddr5	Channel 5 destination address
74	00	R/W	23:0	dma_daddr6	Channel 6 destination address

DMA Channel X Number of Data Register

Base Address: 0x0030_0000
X = 0 ~ 6, where x is channel number
Channel X Address index: 0x18 + 0x10 * (channel number X)

Index	Default	R/W	Bit	Name	Description
18			31:9		Reserved
	0	R/W	8:0	dma_lenth0	Channel 0 total size transfer length: 1~256 0: DMA transfer stop
28			31:9		Reserved
	0	R/W	8:0	dma_lenth1	Channel 1 total size transfer length: 1~256 0: DMA transfer stop
38			31:9		Reserved
	0	R/W	8:0	dma_lenth2	Channel 2 total size transfer length: 1~256 0: DMA transfer stop
48			31:9		Reserved
	0	R/W	8:0	dma_lenth3	Channel 3 total size transfer length: 1~256 0: DMA transfer stop

Index	Default	R/W	Bit	Name	Description
58			31:9		Reserved
	0	R/W	8:0	dma_lenth4	Channel 4 total size transfer length: 1~256 0: DMA transfer stop
68			31:9		Reserved
	0	R/W	8:0	dma_lenth5	Channel 5 total size transfer length: 1~256 0: DMA transfer stop
78			31:9		Reserved
	0	R/W	8:0	dma_lenth6	Channel 6 total size transfer length: 1~256 0: DMA transfer stop

DMA Channel X Configuration Register

Base Address: 0x0030_0000
X = 0 ~ 6, where x is channel number
Channel X Address index: 0x1C + 0x10 * (channel number X)

Index	Default	R/W	Bit	Name	Description
1C			31:16		Reserved
	0	RW	15:14	src_width0	Channel 0 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width0	Channel 0 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved
	0	RW	10	circ_mode0	Channel 0 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled
	0	R/W	9:8	pri_ch0	Channel 0 priority level: 11: Highest priority 10: High priority 01: Medium priority 00: Low priority (Default)
	0	R/W	7:6	src_adr0_ctl	Channel 0 source address control: 11: Reserved 10: Fixed source address 01: Reserved 00: Increment source address (Default)
	0	R/W	5:4	dst_adr0_ctl	Channel 0 destination address control 11: Reserved 10: Fixed destination address 01: Reserved 00: Increment destination address (Default)
	0	R/W	3	en_err_int0	Channel 0 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int0	Channel 0 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable

Index	Default	R/W	Bit	Name	Description
	0	R/W	1	en_total_int0	Channel 0 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen0	Channel 0 write 1 enable.
2C			31:16		Reserved
	0	RW	15:14	src_width1	Channel 1 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width1	Channel 1 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved
	0	RW	10	circ_mode1	Channel 1 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled
	0	R/W	9:8	pri_ch1	Channel 1 priority level: 3: Highest priority 2: High priority 1: Medium priority 0: Low priority (Default)
	0	R/W	7:6	src_adr1_ctl	Channel 1 source address control: 3: Reserved 2: Fixed source address 1: Reserved 0: Increment source address (Default)
	0	R/W	5:4	dst_adr1_ctl	Channel 1 destination address control: 3: Reserved 2: Fixed destination address 1: Reserved 0: Increment destination address (Default)
	0	R/W	3	en_err_int1	Channel 1 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int1	Channel 1 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable
	0	R/W	1	en_total_int1	Channel 1 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen1	Channel 1 write 1 enable
3C			31:16		Reserved
	0	RW	15:14	src_width2	Channel 2 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width2	Channel 2 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved

Index	Default	R/W	Bit	Name	Description
	0	RW	10	circ_mode2	Channel 2 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled
	0	R/W	9:8	pri_ch2	Channel 2 priority level: 3: Highest priority 2: High priority 1: Medium priority 0: Low priority (Default)
	0	R/W	7:6	src_adr2_ctl	Channel 2 source address control: 3: Reserved 2: Fixed source address 1: Reserved 0: Increment source address (Default)
	0	R/W	5:4	dst_adr2_ctl	Channel 2 destination address control: 3: Reserved 2: Fixed destination address 1: Reserved 0: Increment destination address (Default)
	0	R/W	3	en_err_int2	Channel 2 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int2	Channel 2 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable
	0	R/W	1	en_total_int2	Channel 2 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen2	Channel 2 write 1 enable
4C			31:16		Reserved
	0	RW	15:14	src_width3	Channel 3 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width3	Channel 3 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved
	0	RW	10	circ_mode3	Channel 3 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled
	0	R/W	9:8	pri_ch3	Channel 3 priority level: 3: Highest priority 2: High priority 1: Medium priority 0: Low priority (Default)
	0	R/W	7:6	src_adr3_ctl	Channel 3 source address control: 3: Reserved 2: Fixed source address 1: Reserved 0: Increment source address (Default)

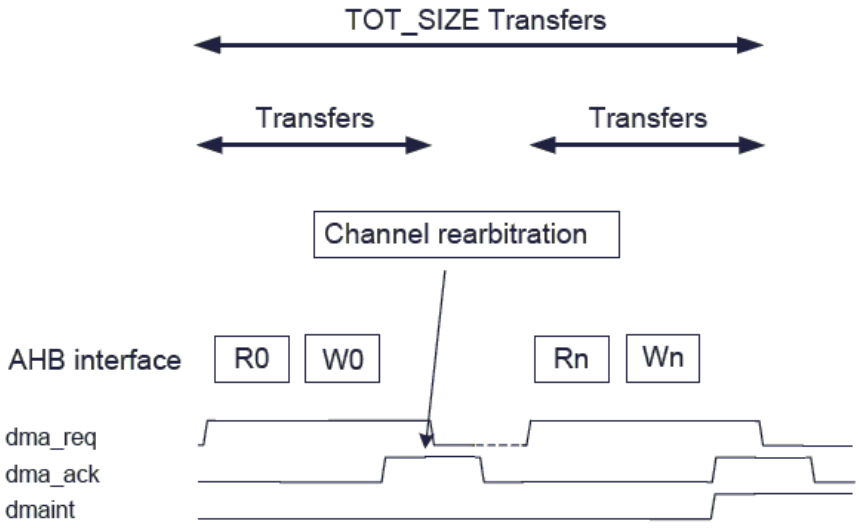
Index	Default	R/W	Bit	Name	Description
	0	R/W	5:4	dst_adr3_ctl	Channel 3 destination address control: 3: Reserved 2: Fixed destination address 1: Reserved 0: Increment destination address (Default)
	0	R/W	3	en_err_int3	Channel 3 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int3	Channel 3 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable
	0	R/W	1	en_total_int3	Channel 3 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen3	Channel 3 write 1 enable
5C			31:16		Reserved
	0	RW	15:14	src_width4	Channel 4 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width4	Channel 4 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved
	0	RW	10	circ_mode4	Channel 4 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled
	0	R/W	9:8	pri_ch4	Channel 4 priority level: 3: Highest priority 2: High priority 1: Medium priority 0: Low priority (Default)
	0	R/W	7:6	src_adr4_ctl	Channel 4 source address control: 3: Reserved 2: Fixed source address 1: Reserved 0: Increment source address (Default)
	0	R/W	5:4	dst_adr4_ctl	Channel 4 destination address control: 3: Reserved 2: Fixed destination address 1: Reserved 0: Increment destination address (Default)
	0	R/W	3	en_err_int4	Channel 4 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int4	Channel 4 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable
	0	R/W	1	en_total_int4	Channel 4 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen4	Channel 4 write 1 enable
6C			31:16		Reserved

Index	Default	R/W	Bit	Name	Description
	0	RW	15:14	src_width5	Channel 5 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width5	Channel 5 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved
	0	RW	10	circ_mode5	Channel 5 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled
	0	R/W	9:8	pri_ch5	Channel 5 priority level: 3: Highest priority 2: High priority 1: Medium priority 0: Low priority (Default)
	0	R/W	7:6	src_adr5_ctl	Channel 5 source address control: 3: Reserved 2: Fixed source address 1: Reserved 0: Increment source address (Default)
	0	R/W	5:4	dst_adr5_ctl	Channel 5 destination address control: 3: Reserved 2: Fixed destination address 1: Reserved 0: Increment destination address (Default)
	0	R/W	3	en_err_int5	Channel 5 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int5	Channel 5 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable
	0	R/W	1	en_total_int5	Channel 5 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen5	Channel 5 write 1 enable
7C			31:16		Reserved
	0	RW	15:14	src_width6	Channel 6 source transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
	0	RW	13:12	dest_width6	Channel 6 destination transfer data width 00: 8bits 01: 16bits 10: 32bits 11: Reserved
			11		Reserved
	0	RW	10	circ_mode6	Channel 6 Circular Buffer Mode 1: Circular buffer mode enabled 0: Circular buffer mode disabled

Index	Default	R/W	Bit	Name	Description
	0	R/W	9:8	pri_ch6	Channel 6 priority level: 3: Highest priority 2: High priority 1: Medium priority 0: Low priority (Default)
	0	R/W	7:6	src_adr6_ctl	Channel 6 source address control: 3: Reserved 2: Fixed source address 1: Reserved 0: Increment source address (Default)
	0	R/W	5:4	dst_adr6_ctl	Channel 6 destination address control: 3: Reserved 2: Fixed destination address 1: Reserved 0: Increment destination address (Default)
	0	R/W	3	en_err_int6	Channel 6 error interrupt enable 1: Error Interrupt enable 0: Error Interrupt disable
	0	R/W	2	en_half_int6	Channel 6 half interrupt enable: 1: half Interrupt enable 0: half Interrupt disable
	0	R/W	1	en_total_int6	Channel 6 total interrupt enable: 1: total Interrupt enable 0: total Interrupt disable
	0	R/W	0	dma_chen6	Channel 6 write 1 enable

4.10.4 DMA Transfer Flow

Each time you enable DMA channel for transfer data. You must configure four registers first, the source address register, destination address register, total transfer length register and configuration register. The configuration register must configure at last.



4.10.5 DMA Transfer Error

There are two kinds of transfer error below,
1. DMA address error: DMA read/write a reserved address.
2. DMA timeout error: When DMA finish a transfer and set act high to device. But device request is not set low after receive DMA ack high.

If DMA transfer error occurs, the transfer channel will disable by hardware and release the channel.

Normal Mode Flow Chart

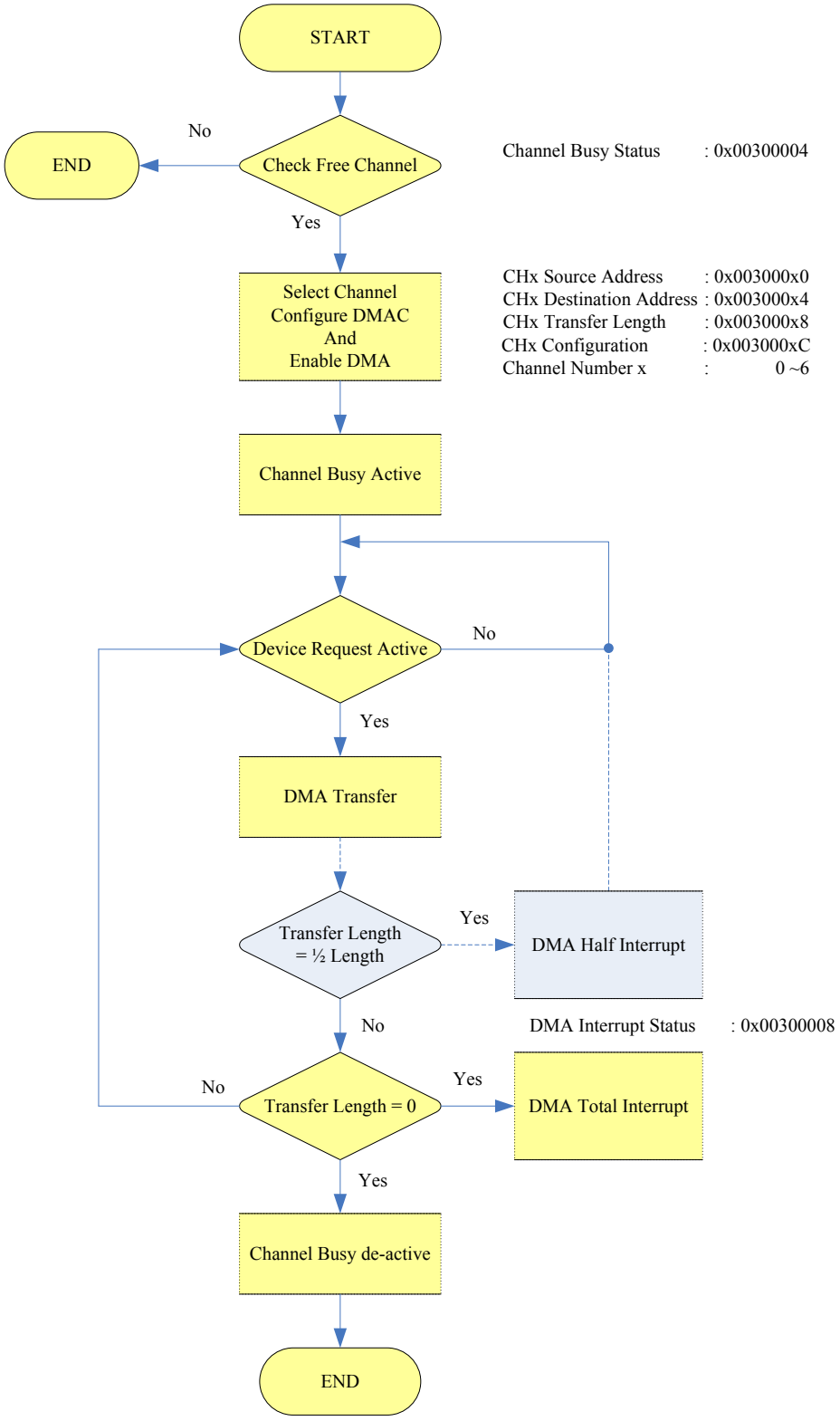


Figure 9: DMA Normal Mode Transfer Flow Chart

Circular Mode Flow Chart

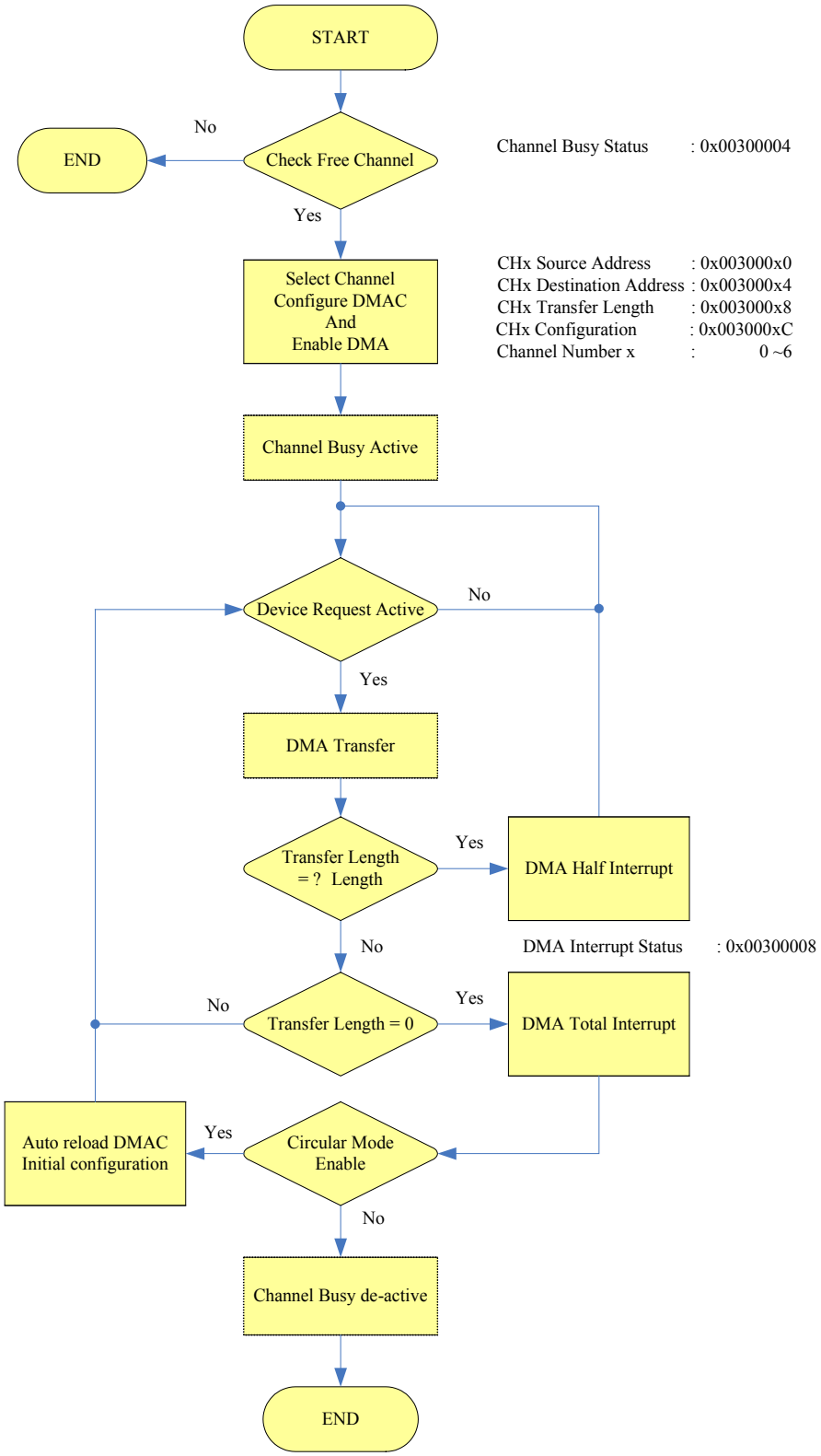


Figure 10: DMA Circular Mode Transfer Flow Chart

Circular Mode

The circular mode is available to handle circular buffer by SRAM.

1. When DMA half length data transfer done, the DMA interrupt sent to MCU and MCU move the first half data from SRAM. At the same time, DMA transfer the last half data from the device to SRAM.
2. When DMA total length data transfer done, the DMA interrupt send to MCU and MCU move the last half data from SRAM. At the same time, DMA reload the initial DMA configuration automatically and transfer the first half data from the device to SRAM.
3. If circular buffer mode is disabled, the DMA will stop after all of the total length data transfer done.

Peripheral to Memory

1. Check which channel available
2. MCU set the DMA channel register
   a. SrcAddr is device address
   b. DstAddr is memory write start address
   c. Length is total number in DMA transaction
   d. Config is set priority, address control, interrupt enable and channel enable
3. DMA hardware handshake
   a. After channel wins the arbitration, DMAC will wait the device req_i signal to be asserted before starting DMA transfer
   b. When data transfer completed, DMAC asserts the ack_o signal to device
   c. When the device receive the ack_o, The device de-asserts the req_i
   d. DMAC de-assert ack_o, update Length, DstAddr and re-arbitrates among all DMA requests
4. DMA transfer complete
   a. The channel Length is 0
   b. The DMAC generates an interrupt signal to MCU
   c. MCU read DMAC interrupt status register to know which channel the interrupt bit was asserted
   d. The MCU write one to the interrupt clear register to clear DMA channel interrupt
   e. Release channel

Memory to Peripheral

1. Check which channel available
2. MCU set the DMA channel register
   a. SrcAddr is device address
   b. DstAddr is memory write start address
   c. Length is total number in DMA transaction
   d. Config is set priority, address control, interrupt enable and channel enable
3. DMA hardware handshake
   a. After channel wins the arbitration, DMAC will wait the device req_i signal to be asserted before starting DMA transfer
   b. When data transfer completed, DMAC asserts the ack_o signal to device
   c. When the device receive the ack_o, The device de-asserts the req_i
   d. DMAC de-assert ack_o, update Length, DstAddr and re-arbitrates among all DMA requests
4. DMA transfer complete
   a. The channel Length is 0
   b. The DMAC generates an interrupt signal to MCU
   c. MCU read DMAC interrupt status register to know which channel the interrupt bit was asserted
   d. The MCU write one to the interrupt clear register to clear DMA channel interrupt
   e. Release channel

Peripheral to Peripheral

1. Check which channel available
2. MCU set the DMA channel register
    a. SrcAddr is device address
    b. DstAddr is memory write start address
    c. Length is total number in DMA transaction
    d. Config is set priority, address control, interrupt enable and channel enable
3. DMA hardware handshake
    a. After channel wins the arbitration, DMAC will wait both of the source and destination the device req_i signal to be asserted before starting DMA transfer
    b. When data transfer completed, DMAC asserts the ack_o signal to device
    c. When the device receive the ack_o, The both of the device de-assert the req_i
    d. DMAC de-assert ack_o, update Length, DstAddr and re-arbitrates among all DMA requests
4. DMA transfer complete
    a. The channel Length is 0
    b. The DMAC generates an interrupt signal to MCU
    c. MCU read DMAC interrupt status register to know which channel the interrupt bit was asserted
    d. The MCU write one to the interrupt clear register to clear DMA channel interrupt
    e. Release channel

DMA Support Peripherals

Device Name	=	Base Address
TIMER0	=	0x0020_1000
TIMER1	=	0x0020_9000
TIMER2	=	0x0020_1400
TIMER3	=	0x0020_9400
TIMER4	=	0x0020_1800
TIMER5	=	0x0020_9800
TIMER6	=	0x0020_1C00
TIMER7	=	0x0020_9C00
TIMER8	=	0x0020_2000
TIMER9	=	0x0020_A000
TIMERA	=	0x0020_2400
TIMERB	=	0x0020_A400
UART0 tx	=	0x0020_300C
UART0 rx	=	0x0020_3010
UART1 tx	=	0x0020_B00C
UART1 rx	=	0x0020_B010
UART2 tx	=	0x0020_340C
UART2 rx	=	0x0020_3410
UART3 tx	=	0x0020_B40C
UART3 rx	=	0x0020_B410
SPI0	=	0x0020_3810
SPI1	=	0x0020_B810
SPI2	=	0x0020_3C10
SPI3	=	0x0020_BC10
I2C0 tx	=	0x0020_400C
I2C0 rx	=	0x0020_4010

I2C2 tx = 0x0020_440C
I2C2 rx = 0x0020_4410
I2C3 tx = 0x0020_C40C
I2C3 rx = 0x0020_C410
I2C4 tx = 0x0020_480C
I2C4 rx = 0x0020_4810
I2C5 tx = 0x0020_C80C
I2C5 rx = 0x0020_C810
I2C6 tx = 0x0020_4C0C
I2C6 rx = 0x0020_4C10
I2C7 tx = 0x0020_CC0C
I2C7 rx = 0x0020_CC10

ADC = 0x0020_8000

4.11 GPIO and AFIO

4.11.1 Functional Description

Subject to the specific hardware characteristics of each I/O port listed in the datasheet, each port bit of the General Purpose IO (GPIO) Ports, can be individually configured by software in several modes:

- Analog input
- Input floating
- Input pull-up (avoid PAD floating, not for external use)
- Output open-drain
- Output push-pull

General-purpose I/O (GPIO)
During and just after reset, the alternate functions are not active and the I/O ports are configured in input floating mode.

When configured as output, the value written to the Output Data register (PT***X***_GPIO) is output on the I/O pin. It is possible to use the output driver in Push-Pull mode or Open-Drain mode (only the N-MOS is activated when outputting 0).

All GPIO pins have an internal weak pull-up which can be activated or not when configured as input.

Atomic bit set or reset:
There is no need for the software to disable interrupts when programming the PT***X***_GPIO at bit level: it is possible to modify only one or several bits in a single atomic APB write access. This is achieved by programming to '1' the Bit Set/Reset Register (PT***A***_BR / PT***X***_BS) to select the bits you want to modify. The unselected bits will not be modified.

Note1: PT***X***_BS has higher priority than PT***X***_BR.
Note2: For port a and port b, when configured as analog input, the pin is needed to be configured as input floating and the internal Schmitt Trigger buffer input have to be turned off.

4.11.2 Register Definition 想成c的指標

GPIO Register: address range 0x001F_6800 ~ 0x001F_6BFF

Index	Default	R/W	Bit	Name	Description
00		-	31:16		Reserved
	'h0	R/W	15:0	PTA_GPIO	Output data to PortA[15:0] 0: output low 1: output high
04		-	31:16		Reserved
		R	15:0	PTA_PADIN	Input data from PortA[15:0]
08		-	31:16		Reserved
	16'hFFFF	R/W	15:0	PTA_DIR	PortA[15:0] Direction 0: output mode 1: input mode
0C	'h0	R/W	31:16 15:0	PTA_CFG	Reserved PortA[15:0] Configuration
10		-	31:16		Reserved
	'h0	R/W	15:0	PTA_BR	PortA output data reset to low 0: no effect 1: PortA[x] clear to low
14		-	31:16		Reserved
	'h0	R/W	15:0	PTA_BS	PortA output data set to high 0: no effect 1: PortA[x] set to high
18		-	31:16		Reserved
	'h0	R/W	15:0	PTA_PADINSEL	Digital Input Buffer Off 0: set PortA[x] as digital I/O 1: set PortA[x] as analog input (AIN)
1C		-	31:16		Reserved
	'h0	R/W	15:0	PTA_FS	PortA digital special function selection
20		-	31:16		Reserved

PTA_CFG:

	PTA_DIR=0	PTA_DIR=1
0	CMOS ouput	Input without pull-up
1	Open-Drain output	Input with pull-up

PTA_FS:

bit	PTA_FS=0	PTA_FS=1
15	GPIO	TMR7
14	GPIO	TMR7
13	GPIO	TMR7
12	GPIO	TMR7
11	GPIO	TMR2
10	GPIO	TMR2
9	GPIO	TMR2
8	GPIO	TMR2
7	GPIO	TMR9
6	GPIO	TMR9
5	GPIO	TMR9
4	GPIO	TMR9
3	GPIO	SPI3_NSS
2	GPIO	SPI3_SCK
1	GPIO	SPI3_MOSI
0	GPIO	SPI3_MISO

Index	Default	R/W	Bit	Name	Description
	'h0	R/W	15:0	PTB_GPIO	Output data to PortB[15:0] 0: output low 1: output high
24		-	31:16		Reserved
		R	15:0	PTB_PADIN	Input data from PortB[15:0]
28		-	31:16		Reserved
	16'hFFFF	R/W	15:0	PTB_DIR	PortB[15:0] Direction 0: output mode 1: input mode
2C			31:16		Reserved
	'h0	R/W	15:0	PTB_CFG	PortB[15:0] Configuration

	PTB_DIR=0	PTB_DIR=1
0	CMOS ouput	Input without pull-up
1	Open-Drain output	Input with pull-up

30		-	31:16		Reserved
	'h0	R/W	15:0	PTB_BR	PortB output data reset to low 0: no effect 1: PortB[x] clear to low
34		-	31:16		Reserved
	'h0	R/W	15:0	PTB_BS	PortB output data set to high 0: no effect 1: PortB[x] set to high
38		-	31:16		Reserved
	'h0	R/W	15:0	PTB_PADINSEL	Digital Input Buffer Off 0: set PortB[x] as digital I/O 1: set PortB[x] as analog input (AIN)
3C		-	31:16		Reserved
	'h0	R/W	15:0	PTB_FS	PortA function selection

	PTB_FS=0	PTB_FS=1
15	GPIO	I2C0_SDA
14	GPIO	I2C0_SCL
13	GPIO	I2C4_SDA
12	GPIO	I2C4_SCL
11	GPIO	I2C3_SDA / TMR8
10	GPIO	I2C3_SCL / TMR8
9	GPIO	I2C2_SDA / TMR8
8	GPIO	I2C2_SCL / TMR8
7	GPIO	I2C7_SDA / TMR6
6	GPIO	I2C7_SCL / TMR6
5	GPIO	I2C5_SDA / TMR6
4	GPIO	I2C5_SCL / TMR6
3	GPIO	TMR4
2	GPIO	TMR4
1	GPIO	TMR4
0	GPIO	TMR4

40		-	31:16		Reserved
	'h0	R/W	15:0	PTC_GPIO	Output data to PortC[15:0] 0: output low 1: output high
44		-	31:16		Reserved

<table>
<tr><th>Index</th><th>Default</th><th>R/W</th><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td>R</td><td>15:0</td><td>PTC_PADIN</td><td>Input data from PortC[15:0]</td></tr>
<tr><td rowspan="2">48</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>16’hFFFF</td><td>R/W</td><td>15:0</td><td>PTC_DIR</td><td>PortC[15:0] Direction
0: output mode
1: input mode</td></tr>
<tr><td>4C</td><td></td><td></td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTC_CFG</td><td>PortC[15:0] Configuration
<table>
<tr><th></th><th>PTC_DIR=0</th><th>PTC_DIR=1</th></tr>
<tr><td>0</td><td>CMOS ouput</td><td>Input without pull-up</td></tr>
<tr><td>1</td><td>Open-Drain output</td><td>Input with pull-up</td></tr>
</table></td></tr>
<tr><td rowspan="2">50</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>’h0</td><td>R/W</td><td>15:0</td><td>PTC_BR</td><td>PortC output data reset to low
0: no effect
1: PortC[x] clear to low</td></tr>
<tr><td rowspan="2">54</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>’h0</td><td>R/W</td><td>15:0</td><td>PTC_BS</td><td>PortC output data set to high
0: no effect
1: PortC[x] set to high</td></tr>
<tr><td rowspan="2">58</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>’h0</td><td>R/W</td><td>15:0</td><td>PTC_PADINSEL</td><td>Digital Input Buffer Off
0: set PortC[x] as digital I/O
1: set PortC[x] as analog input</td></tr>
<tr><td rowspan="2">5C</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>’h0</td><td>R/W</td><td>15:0</td><td>PTC_FS</td><td>PortC digital special function selection
<table>
<tr><th>bit</th><th>PTC_FS=0</th><th>PTC_FS=1</th></tr>
<tr><td>15</td><td>GPIO</td><td>PWM11</td></tr>
<tr><td>14</td><td>GPIO</td><td>PWM10</td></tr>
<tr><td>13</td><td>GPIO</td><td>PWM9</td></tr>
<tr><td>12</td><td>GPIO</td><td>PWM8</td></tr>
<tr><td>11</td><td>GPIO</td><td>PWM7</td></tr>
<tr><td>10</td><td>GPIO</td><td>PWM6</td></tr>
<tr><td>9</td><td>GPIO</td><td>UART3_RX</td></tr>
<tr><td>8</td><td>GPIO</td><td>UART3_TX</td></tr>
<tr><td>7</td><td>GPIO</td><td>TMRB</td></tr>
<tr><td>6</td><td>GPIO</td><td>TMRB</td></tr>
<tr><td>5</td><td>GPIO</td><td>TMRB</td></tr>
<tr><td>4</td><td>GPIO</td><td>TMRB</td></tr>
<tr><td>3</td><td>GPIO</td><td></td></tr>
<tr><td>2</td><td>GPIO</td><td></td></tr>
<tr><td>1</td><td>GPIO</td><td>I2C6_SDA</td></tr>
<tr><td>0</td><td>GPIO</td><td>I2C6_SCL</td></tr>
</table></td></tr>
<tr><td rowspan="2">60</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>’h0</td><td>R/W</td><td>15:0</td><td>PTD_GPIO</td><td>Output data to PortD[15:0]
0: output low
1: output high</td></tr>
<tr><td rowspan="2">64</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>R</td><td>15:0</td><td>PTD_PADIN</td><td>Input data from PortD[15:0]</td></tr>
<tr><td rowspan="2">68</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>16’hFFFF</td><td>R/W</td><td>15:0</td><td>PTD_DIR</td><td>PortD[15:0] Direction
0: output mode</td></tr>
</table>

<table>
<tr><th>Index</th><th>Default</th><th>R/W</th><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>1: input mode</td></tr>
<tr><td>6C</td><td>’h0</td><td>R/W</td><td>31:16
15:0</td><td>PTD_CFG</td><td>Reserved
PortD[15:0] Configuration
<table>
<tr><td></td><td>PTD_DIR=0</td><td>PTD_DIR=1</td></tr>
<tr><td>0</td><td>CMOS ouput</td><td>Input without pull-up</td></tr>
<tr><td>1</td><td>Open-Drain output</td><td>Input with pull-up</td></tr>
</table></td></tr>
<tr><td>70</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTD_BR</td><td>PortD output data reset to low
0: no effect
1: PortD[x] clear to low</td></tr>
<tr><td>74</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTD_BS</td><td>PortD output data set to high
0: no effect
1: PortD[x] set to high</td></tr>
<tr><td>78</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTD_PADINSEL</td><td>Digital Input Buffer Off
0: set PortD[x] as digital I/O
1: set PortD[x] as analog input</td></tr>
<tr><td>7C</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTD_FS</td><td>PortD digital special function selection
<table>
<tr><td>bit</td><td>PTD_FS=0</td><td>PTD_FS=1</td></tr>
<tr><td>15</td><td>GPIO</td><td>SPI2_NSS / TMRA</td></tr>
<tr><td>14</td><td>GPIO</td><td>SPI2_SCK / TMRA</td></tr>
<tr><td>13</td><td>GPIO</td><td>SPI2_MOSI / TMRA</td></tr>
<tr><td>12</td><td>GPIO</td><td>SPI2_MISO / TMRA</td></tr>
<tr><td>11</td><td>GPIO</td><td>SPI1_NSS</td></tr>
<tr><td>10</td><td>GPIO</td><td>SPI1_SCK</td></tr>
<tr><td>9</td><td>GPIO</td><td>SPI1_MOSI</td></tr>
<tr><td>8</td><td>GPIO</td><td>SPI1_MISO</td></tr>
<tr><td>7</td><td>GPIO</td><td>SPI0_NSS</td></tr>
<tr><td>6</td><td>GPIO</td><td>SPI0_SCK</td></tr>
<tr><td>5</td><td>GPIO</td><td>SPI0_MOSI</td></tr>
<tr><td>4</td><td>GPIO</td><td>SPI0_MISO</td></tr>
<tr><td>3</td><td>GPIO</td><td>TMR0</td></tr>
<tr><td>2</td><td>GPIO</td><td>TMR0</td></tr>
<tr><td>1</td><td>GPIO</td><td>TMR0</td></tr>
<tr><td>0</td><td>GPIO</td><td>TMR0</td></tr>
</table></td></tr>
<tr><td>80</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTE_GPIO</td><td>Output data to PortE[15:0]
0: output low
1: output high</td></tr>
<tr><td>84</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td></td><td>R</td><td>15:0</td><td>PTE_PADIN</td><td>Input data from PortE[15:0]</td></tr>
<tr><td>88</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>16’hFFFF</td><td>R/W</td><td>15:0</td><td>PTE_DIR</td><td>PortE[15:0] Direction
0: output mode
1: input mode</td></tr>
<tr><td>8C</td><td></td><td></td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td></td><td>’h0</td><td>R/W</td><td>15:0</td><td>PTE_CFG</td><td>PortE[15:0] Configuration</td></tr>
</table>

Index	Default	R/W	Bit	Name	Description
					<table><tr><td></td><td>PTE_DIR=0</td><td>PTE_DIR=1</td></tr><tr><td>0</td><td>CMOS ouput</td><td>Input without pull-up</td></tr><tr><td>1</td><td>Open-Drain output</td><td>Input with pull-up</td></tr></table>
90		-	31:16		Reserved
	'h0	R/W	15:0	PTE_BR	PortE output data reset to low 0: no effect 1: PortE[x] clear to low
94		-	31:16		Reserved
	'h0	R/W	15:0	PTE_BS	PortE output data set to high 0: no effect 1: PortE[x] set to high
98		-	31:16		Reserved
	'h0	R/W	15:0	PTE_PADINSEL	Digital Input Buffer Off 0: set PortE[x] as digital I/O 1: set PortE[x] as analog input
9C		-	31:16		Reserved
	'h0	R/W	15:0	PTE_FS	PortE digital special function selection <table><tr><td>bit</td><td>PTE_FS=0</td><td>PTE_FS=1</td></tr><tr><td>15</td><td>GPIO</td><td>TMR3_capi1</td></tr><tr><td>14</td><td>GPIO</td><td>TMR3_capi0</td></tr><tr><td>13</td><td>GPIO</td><td>TMR3_mati1</td></tr><tr><td>12</td><td>GPIO</td><td>TMR3_mati0</td></tr><tr><td>11</td><td>GPIO</td><td>PWM5</td></tr><tr><td>10</td><td>GPIO</td><td>PWM4</td></tr><tr><td>9</td><td>GPIO</td><td>PWM3</td></tr><tr><td>8</td><td>GPIO</td><td>PWM2</td></tr><tr><td>7</td><td>GPIO</td><td>PWM1</td></tr><tr><td>6</td><td>GPIO</td><td>PWM0</td></tr><tr><td>5</td><td>GPIO</td><td>UART2_RX</td></tr><tr><td>4</td><td>GPIO</td><td>UART2_TX</td></tr><tr><td>3</td><td>GPIO</td><td>UART1_RX / TMR1</td></tr><tr><td>2</td><td>GPIO</td><td>UART1_TX / TMR1</td></tr><tr><td>1</td><td>GPIO</td><td>UART0_RX / TMR1</td></tr><tr><td>0</td><td>GPIO</td><td>UART0_TX / TMR1</td></tr></table>
A0		-	31:16		Reserved
	'h0	R/W	15:0	PTF_GPIO	Output data to PortF[15:0] 0: output low 1: output high
A4		-	31:16		Reserved
		R	15:0	PTF_PADIN	Input data from PortF[15:0]
A8		-	31:16		Reserved
	16'hFFFF	R/W	15:0	PTF_DIR	PortE[15:0] Direction 0: output mode 1: input mode
AC			31:16		Reserved
	'h0	R/W	15:0	PTF_CFG	PortF[15:0] Configuration <table><tr><td></td><td>PTF_DIR=0</td><td>PTF_DIR=1</td></tr><tr><td>0</td><td>CMOS ouput</td><td>Input without pull-up</td></tr><tr><td>1</td><td>Open-Drain output</td><td>Input with pull-up</td></tr></table>

<table>
<tr><th>Index</th><th>Default</th><th>R/W</th><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td rowspan="2">B0</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>'h0</td><td>R/W</td><td>15:0</td><td>PTF_BR</td><td>PortF output data reset to low
0: no effect
1: PortF[x] clear to low</td></tr>
<tr><td rowspan="2">B4</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>'h0</td><td>R/W</td><td>15:0</td><td>PTF_BS</td><td>PortF output data set to high
0: no effect
1: PortF[x] set to high</td></tr>
<tr><td rowspan="2">B8</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>'h0</td><td>R/W</td><td>15:0</td><td>PTF_PADINSEL</td><td>Digital Input Buffer Off
0: set PortF[x] as digital I/O
1: set PortF[x] as analog input</td></tr>
<tr><td rowspan="2">BC</td><td></td><td>-</td><td>31:16</td><td></td><td>Reserved</td></tr>
<tr><td>'h3C0</td><td>R/W</td><td>15:0</td><td>PTF_FS</td><td>PortF digital special function selection
<table>
<tr><th>bit</th><th>PTF_FS=0</th><th>PTF_FS=1</th></tr>
<tr><td>9</td><td>GPIO</td><td>LXOUT</td></tr>
<tr><td>8</td><td>GPIO</td><td>LXIN</td></tr>
<tr><td>7</td><td>GPIO</td><td>HXOUT</td></tr>
<tr><td>6</td><td>GPIO</td><td>HXIN</td></tr>
<tr><td>5</td><td>GPIO</td><td>VERF</td></tr>
<tr><td>4</td><td>GPIO</td><td></td></tr>
<tr><td>3</td><td>GPIO</td><td>TMR5_capi1</td></tr>
<tr><td>2</td><td>GPIO</td><td>TMR5_capi0</td></tr>
<tr><td>1</td><td>GPIO</td><td>TMR5_mati1</td></tr>
<tr><td>0</td><td>GPIO</td><td>TMR5_mati0</td></tr>
</table></td></tr>
</table>

4.12 Wake Up & Interrupt

4.12.1 Wake Up & Toggle Register

Wake-up source includes RTC 1s, LVD, and GPIO
Wake-up & Interrupt: address range 0x0020_0C00 ~ 0x0020_0FFF

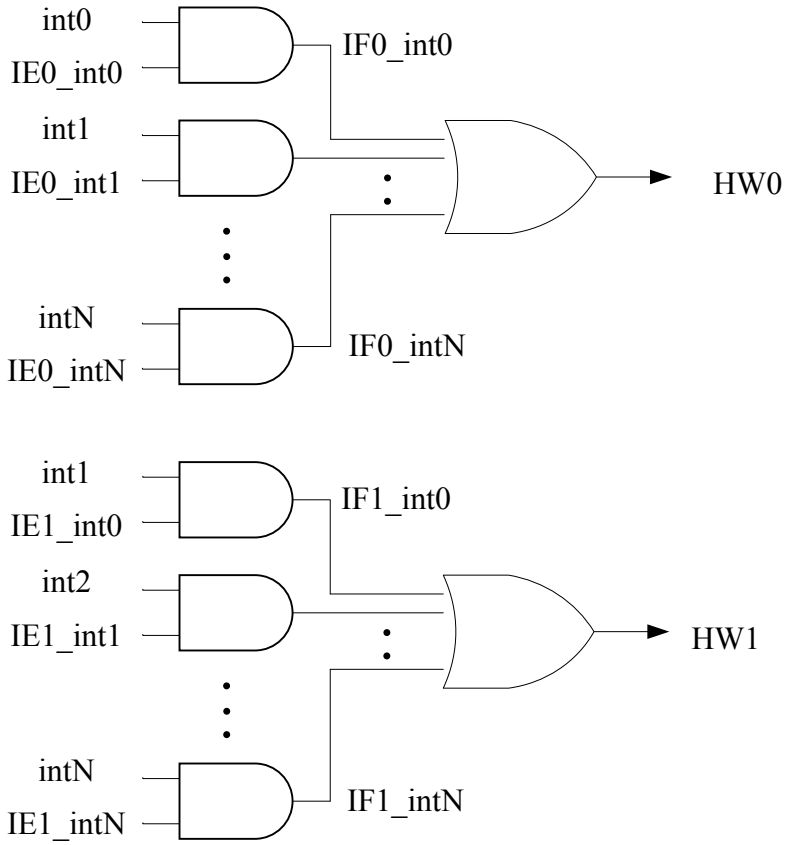
Index	Default	R/W	Bit	Name	Description
04			31:6		Reserved
	0	R	5	IN_TOG	0: No input toggle 1: OR all input toggle
			4		Reserved
	0	R	3	RTC_1S_TOG	0: No RTC 1s trigger 1: RTC 1s trigger
	0	R	2	LVD_EVT	0: No LVD event trigger 1: LVD event trigger
			1:0		Reserved
10			31:8		Reserved
	0	R/W	7:0	PTA_WAKE	PortA[x] Wake up enable 0: Disable MCU wake up via PortA[x] 1: Enable MCU wake-up via PortA[x] state change, The I/O wake-up trigger type is defined by PTA_TRIGEDGE, and MCU ISR enable is defined by PTA_IE.
14			31:8		Reserved
	0	R/W	7:0	PTA_IE	When MCU wake up by PortA[x] trigger, enable ISR of MCU 0: MCU wake up and no enter ISR. 1: MCU wake up and enter ISR.
18			31:8		Reserved
	0	R/W	7:0	PTA_TRIGEDGE	PortA[x] wake up trigger state 0: rising edge trigger. 1: falling edge trigger.
1C			31:8		Reserved
	0	R/W	7:0	PTA_IF	PortA[x] wake up trigger flag 0: no event. 1: event of state change. Write “1” to clear trigger flag.
20			31:8		Reserved
	0	R/W	7:0	PTB_WAKE	PortB[x] Wake up enable 0: Disable MCU wake up via PortB[x] 1: Enable MCU wake-up via PortB[x] state change, The I/O wake-up trigger type is defined by PTB_TRIGEDGE, and MCU ISR enable is defined by PTB_IE.
24			31:8		Reserved
	0	R/W	7:0	PTB_IE	When MCU wake up by PortB[x] trigger, enable ISR of MCU 0: MCU wake up and no enter ISR. 1: MCU wake up and enter ISR.
28			31:8		Reserved

Index	Default	R/W	Bit	Name	Description
	0	R/W	7:0	PTB_TRIGEDGE	PortB[x] wake up trigger state 0: rising edge trigger. 1: falling edge trigger.
2C			31:8		Reserved
	0	R/W	7:0	PTB_IF	PortB[x] wake up trigger flag 0: no event. 1: event of state change. Write “1” to clear trigger flag.
40			31:8		Reserved
	0	R/W	7:0	PTD_WAKE	PortD[x] Wake up enable 0: Disable MCU wake up via PortD[x] 1: Enable MCU wake-up via PortD[x] state change, The I/O wake-up trigger type is defined by PTD_TRIGEDGE, and MCU ISR enable is defined by PTD_IE.
44			31:8		Reserved
	0	R/W	7:0	PTD_IE	When MCU wake up by PortD[x] trigger, enable ISR of MCU 0: MCU wake up and no enter ISR. 1: MCU wake up and enter ISR.
48			31:8		Reserved
	0	R/W	7:0	PTD_TRIGEDGE	PortD[x] wake up trigger state 0: rising edge trigger. 1: falling edge trigger.
4C			31:8		Reserved
	0	R/W	7:0	PTD_IF	PortD[x] wake up trigger flag 0: no event. 1: event of state change. Write “1” to clear trigger flag.
50			31:8		Reserved
	0	R/W	7:0	PTE_WAKE	PortE[x] Wake up enable 0: Disable MCU wake up via PortE[x] 1: Enable MCU wake-up via PortE[x] state change, The I/O wake-up trigger type is defined by PTE_TRIGEDGE, and MCU ISR enable is defined by PTE_IE.
54			31:8		Reserved
	0	R/W	7:0	PTE_IE	When MCU wake up by PortE[x] trigger, enable ISR of MCU 0: MCU wake up and no enter ISR. 1: MCU wake up and enter ISR.
58			31:8		Reserved
	0	R/W	7:0	PTE_TRIGEDGE	PortE[x] wake up trigger state 0: rising edge trigger. 1: falling edge trigger.
5C			31:8		Reserved
	0	R/W	7:0	PTE_IF	PortE[x] wake up trigger flag 0: no event. 1: event of state change. Write “1” to clear trigger flag.

4.12.2 MCU Interrupt

INT0 (HW0) is caused by LVD, WWDT, SPI, I2C, ADC, RTC 1s, UART, QEI, PWM, DMA, Timer and input toggle interruption. Each interrupt can be enabled/disabled independently by programming IE0_XXX register and identified by IF0_XXX register.

INT1 (HW1) is caused by LVD, WWDT, SPI, I2C, ADC, RTC 1s, UART, QEI, PWM, DMA, Timer and input toggle interruption. Each interrupt can be enabled/disabled independently by programming IE1_XXX register and identified by IF1_XXX register.



Wake-up & Interrupt: address range 0x0020_0C00 ~ 0x0020_0FFF

Index	Default	R/W	Bit	Name	Description
100	0		31:16		Reserved
	0		15		Reserved
	0	R/W	14	IE0_DMA6	0: Disable DMA6 interrupt 1: Enable DMA6 interrupt
	0	R/W	13	IE0_DMA5	0: Disable DMA5 interrupt 1: Enable DMA5 interrupt
	0	R/W	12	IE0_DMA4	0: Disable DMA4 interrupt 1: Enable DMA4 interrupt
	0	R/W	11	IE0_DMA3	0: Disable DMA3 interrupt 1: Enable DMA3 interrupt
	0	R/W	10	IE0_DMA2	0: Disable DMA2 interrupt

Index	Default	R/W	Bit	Name	Description
					1: Enable DMA2 interrupt
	0	R/W	9	IE0_DMA1	0: Disable DMA1 interrupt 1: Enable DMA1 interrupt
	0	R/W	8	IE0_DMA0	0: Disable DMA0 interrupt 1: Enable DMA0 interrupt
	0		7		Reserved
	0	R/W	6	IE0_ALARM	0: Disable RTC alarm interrupt 1: Enable RTC alarm interrupt
	0	R/W	5	IE0_ADC	0: Disable ADC interrupt 1: Enable ADC interrupt
	0	R/W	4	IE0_FSCM	Must be set as 0
	0	R/W	3	IE0_WWDT	0: Disable WWDT interrupt 1: Enable WWDT interrupt
	0	R/W	2	IE0_IN_TOG	0: Disable all-input toggle interrupt 1: Enable all-input toggle interrupt
	0	R/W	1	IE0_RTC1S	0: Disable RTC 1s interrupt 1: Enable RTC 1s interrupt
	0	R/W	0	IE0_LVD	0: Disable LVD interrupt 1: Enable LVD interrupt
104			31:20		Reserved
	0	R/W	15	IE0_I2C7	0: Disable I2C7 interrupt 1: Enable I2C7 interrupt
	0	R/W	14	IE0_I2C6	0: Disable I2C6 interrupt 1: Enable I2C6 interrupt
	0	R/W	13	IE0_I2C5	0: Disable I2C5 interrupt 1: Enable I2C5 interrupt
	0	R/W	12	IE0_I2C4	0: Disable I2C4 interrupt 1: Enable I2C4 interrupt
	0	R/W	11	IE0_I2C3	0: Disable I2C3 interrupt 1: Enable I2C3 interrupt
	0	R/W	10	IE0_I2C2	0: Disable I2C2 interrupt 1: Enable I2C2 interrupt
	0	R/W	9	IE0_IR	0: Disable IR interrupt 1: Enable IR interrupt
	0	R/W	8	IE0_I2C0	0: Disable I2C0 interrupt 1: Enable I2C0 interrupt
	0	R/W	7	IE0_SPI3	0: Disable SPI3 interrupt 1: Enable SPI3 interrupt
	0	R/W	6	IE0_SPI2	0: Disable SPI2 interrupt 1: Enable SPI2 interrupt
	0	R/W	5	IE0_SPI1	0: Disable SPI1 interrupt 1: Enable SPI1 interrupt
	0	R/W	4	IE0_SPI0	0: Disable SPI0 interrupt 1: Enable SPI0 interrupt
	0	R/W	3	IE0_UART3	0: Disable UART3 interrupt 1: Enable UART3 interrupt
	0	R/W	2	IE0_UART2	0: Disable UART2 interrupt 1: Enable UART2 interrupt
	0	R/W	1	IE0_UART1	0: Disable UART1 interrupt

Index	Default	R/W	Bit	Name	Description
					1: Enable UART1 interrupt
	0	R/W	0	IE0_UART0	0: Disable UART0 interrupt 1: Enable UART0 interrupt
108			31:12		Reserved
	0	R/W	11	IE0_TIMERB	0: Disable TIMERB interrupt 1: Enable TIMERB interrupt
	0	R/W	10	IE0_TIMERA	0: Disable TIMERA interrupt 1: Enable TIMERA interrupt
	0	R/W	9	IE0_TIMER9	0: Disable TIMER9 interrupt 1: Enable TIMER9 interrupt
	0	R/W	8	IE0_TIMER8	0: Disable TIMER8 interrupt 1: Enable TIMER8 interrupt
	0	R/W	7	IE0_TIMER7	0: Disable TIMER7 interrupt 1: Enable TIMER7 interrupt
	0	R/W	6	IE0_TIMER6	0: Disable TIMER6 interrupt 1: Enable TIMER6 interrupt
	0	R/W	5	IE0_TIMER5	0: Disable TIMER5 interrupt 1: Enable TIMER5 interrupt
	0	R/W	4	IE0_TIMER4	0: Disable TIMER4 interrupt 1: Enable TIMER4 interrupt
	0	R/W	3	IE0_TIMER3	0: Disable TIMER3 interrupt 1: Enable TIMER3 interrupt
	0	R/W	2	IE0_TIMER2	0: Disable TIMER2 interrupt 1: Enable TIMER2 interrupt
	0	R/W	1	IE0_TIMER1	0: Disable TIMER1 interrupt 1: Enable TIMER1 interrupt
	0	R/W	0	IE0_TIMER0	0: Disable TIMER0 interrupt 1: Enable TIMER0 interrupt
110			31:16		Reserved
	0		15		Reserved
	0	R/W	14	IE1_DMA6	0: Disable DMA6 interrupt 1: Enable DMA6 interrupt
	0	R/W	13	IE1_DMA5	0: Disable DMA5 interrupt 1: Enable DMA5 interrupt
	0	R/W	12	IE1_DMA4	0: Disable DMA4 interrupt 1: Enable DMA4 interrupt
	0	R/W	11	IE1_DMA3	0: Disable DMA3 interrupt 1: Enable DMA3 interrupt
	0	R/W	10	IE1_DMA2	0: Disable DMA2 interrupt 1: Enable DMA2 interrupt
	0	R/W	9	IE1_DMA1	0: Disable DMA1 interrupt 1: Enable DMA1 interrupt
	0	R/W	8	IE1_DMA0	0: Disable DMA0 interrupt 1: Enable DMA0 interrupt
	0		7		Reserved
	0	R/W	6	IE1_ALARM	0: Disable RTC alarm interrupt 1: Enable RTC alarm interrupt
	0	R/W	5	IE1_ADC	0: Disable ADC interrupt 1: Enable ADC interrupt
	0	R/W	4	IE1_FSCM	0: Disable FSCM interrupt 1: Enable FSCM interrupt
	0	R/W	3	IE1_WWDT	0: Disable WWDT interrupt 1: Enable WWDT interrupt

Index	Default	R/W	Bit	Name	Description
	0	R/W	2	IE1_IN_TOG	0: Disable all-input toggle interrupt 1: Enable all-input toggle interrupt
	0	R/W	1	IE1_RTC1S	0: Disable RTC 1s interrupt 1: Enable RTC 1s interrupt
	0	R/W	0	IE1_LVD	0: Disable LVD interrupt 1: Enable LVD interrupt
114			31:20		Reserved
	0	R/W	15	IE1_I2C7	0: Disable I2C7 interrupt 1: Enable I2C7 interrupt
	0	R/W	14	IE1_I2C6	0: Disable I2C6 interrupt 1: Enable I2C6 interrupt
	0	R/W	13	IE1_I2C5	0: Disable I2C5 interrupt 1: Enable I2C5 interrupt
	0	R/W	12	IE1_I2C4	0: Disable I2C4 interrupt 1: Enable I2C4 interrupt
	0	R/W	11	IE1_I2C3	0: Disable I2C3 interrupt 1: Enable I2C3 interrupt
	0	R/W	10	IE1_I2C2	0: Disable I2C2 interrupt 1: Enable I2C2 interrupt
	0	R/W	9	IE1_IR	0: Disable IR interrupt 1: Enable IR interrupt
	0	R/W	8	IE1_I2C0	0: Disable I2C0 interrupt 1: Enable I2C0 interrupt
	0	R/W	7	IE1_SPI3	0: Disable SPI3 interrupt 1: Enable SPI3 interrupt
	0	R/W	6	IE1_SPI2	0: Disable SPI2 interrupt 1: Enable SPI2 interrupt
	0	R/W	5	IE1_SPI1	0: Disable SPI1 interrupt 1: Enable SPI1 interrupt
	0	R/W	4	IE1_SPI0	0: Disable SPI0 interrupt 1: Enable SPI0 interrupt
	0	R/W	3	IE1_UART3	0: Disable UART3 interrupt 1: Enable UART3 interrupt
	0	R/W	2	IE1_UART2	0: Disable UART2 interrupt 1: Enable UART2 interrupt
	0	R/W	1	IE1_UART1	0: Disable UART1 interrupt 1: Enable UART1 interrupt
	0	R/W	0	IE1_UART0	0: Disable UART0 interrupt 1: Enable UART0 interrupt
118			31:12		Reserved
	0	R/W	11	IE1_TIMERB	0: Disable TIMERB interrupt 1: Enable TIMERB interrupt
	0	R/W	10	IE1_TIMERA	0: Disable TIMERA interrupt 1: Enable TIMERA interrupt
	0	R/W	9	IE1_TIMER9	0: Disable TIMER9 interrupt 1: Enable TIMER9 interrupt
	0	R/W	8	IE1_TIMER8	0: Disable TIMER8 interrupt 1: Enable TIMER8 interrupt
	0	R/W	7	IE1_TIMER7	0: Disable TIMER7 interrupt

Index	Default	R/W	Bit	Name	Description
					1: Enable TIMER7 interrupt
	0	R/W	6	IE1_TIMER6	0: Disable TIMER6 interrupt 1: Enable TIMER6 interrupt
	0	R/W	5	IE1_TIMER5	0: Disable TIMER5 interrupt 1: Enable TIMER5 interrupt
	0	R/W	4	IE1_TIMER4	0: Disable TIMER4 interrupt 1: Enable TIMER4 interrupt
	0	R/W	3	IE1_TIMER3	0: Disable TIMER3 interrupt 1: Enable TIMER3 interrupt
	0	R/W	2	IE1_TIMER2	0: Disable TIMER2 interrupt 1: Enable TIMER2 interrupt
	0	R/W	1	IE1_TIMER1	0: Disable TIMER1 interrupt 1: Enable TIMER1 interrupt
	0	R/W	0	IE1_TIMER0	0: Disable TIMER0 interrupt 1: Enable TIMER0 interrupt
120			31:16		Reserved
	0	R	15		Reserved
	0	R	14	IF0_DMA6	0: No event of DMA6 interrupt 1: Event of DMA6 interrupt
	0	R	13	IF0_DMA5	0: No event of DMA5 interrupt 1: Event DMA5 interrupt
	0	R	12	IF0_DMA4	0: No event of DMA4 interrupt 1: Event DMA4 interrupt
	0	R	11	IF0_DMA3	0: No event of DMA3 interrupt 1: Event DMA3 interrupt
	0	R	10	IF0_DMA2	0: No event of DMA2 interrupt 1: Event DMA2 interrupt
	0	R	9	IF0_DMA1	0: No event of DMA1 interrupt 1: Event DMA1 interrupt
	0	R	8	IF0_DMA0	0: No event of DMA0 interrupt 1: Event DMA0 interrupt
	0		7		Reserved
	0	R	6	IF0_ALARM	0: No event of RTC alarm interrupt 1: Event ETC alarm interrupt
	0	R	5	IF0_ADC	0: No event of ADC interrupt 1: Event ADC interrupt
	0	R	4	IF0_FSCM	Must be set as 0
	0	R	3	IF0_WWDT	0: No event of WWDT interrupt 1: Event WWDT interrupt
	0	R	2	IF0_IN_TOG	0: No event of all-input toggle interrupt 1: Event all-input toggle interrupt
	0	R	1	IF0_RTC1S	0: No event of RTC 1s interrupt 1: Event RTC 1s interrupt
	0	R	0	IF0_LVD	0: No event of LVD interrupt 1: Event LVD interrupt
124			31:20		Reserved
	0	R	15	IF0_I2C7	0: No event of I2C7 interrupt 1: Event I2C7 interrupt
	0	R	14	IF0_I2C6	0: No event of I2C6 interrupt

Index	Default	R/W	Bit	Name	Description
					1: Event I2C6 interrupt
	0	R	13	IF0_I2C5	0: No event of I2C5 interrupt 1: Event I2C5 interrupt
	0	R	12	IF0_I2C4	0: No event of I2C4 interrupt 1: Event I2C4 interrupt
	0	R	11	IF0_I2C3	0: No event of I2C3 interrupt 1: Event I2C3 interrupt
	0	R	10	IF0_I2C2	0: No event of I2C2 interrupt 1: Event I2C2 interrupt
	0	R	9	IF0_IR	0: No event of IR interrupt 1: Event IR interrupt
	0	R	8	IF0_I2C0	0: No event of I2C0 interrupt 1: Event I2C0 interrupt
	0	R	7	IF0_SPI3	0: No event of SPI3 interrupt 1: Event SPI3 interrupt
	0	R	6	IF0_SPI2	0: No event of SPI2 interrupt 1: Event SPI2 interrupt
	0	R	5	IF0_SPI1	0: No event of SPI1 interrupt 1: Event SPI1 interrupt
	0	R	4	IF0_SPI0	0: No event of SPI0 interrupt 1: Event SPI0 interrupt
	0	R	3	IF0_UART3	0: No event of UART3 interrupt 1: Event UART3 interrupt
	0	R	2	IF0_UART2	0: No event of UART2 interrupt 1: Event UART2 interrupt
	0	R	1	IF0_UART1	0: No event of UART1 interrupt 1: Event UART1 interrupt
	0	R	0	IF0_UART0	0: No event of UART0 interrupt 1: Event UART0 interrupt
128			31:12		Reserved
	0	R	11	IF0_TIMERB	0: No event of TIMERB interrupt 1: Event TIMERB interrupt
	0	R	10	IF0_TIMERA	0: No event of TIMERA interrupt 1: Event TIMERA interrupt
	0	R	9	IF0_TIMER9	0: No event of TIMER9 interrupt 1: Event TIMER9 interrupt
	0	R	8	IF0_TIMER8	0: No event of TIMER8 interrupt 1: Event TIMER8 interrupt
	0	R	7	IF0_TIMER7	0: No event of TIMER7 interrupt 1: Event TIMER7 interrupt
	0	R	6	IF0_TIMER6	0: No event of TIMER6 interrupt 1: Event TIMER6 interrupt
	0	R	5	IF0_TIMER5	0: No event of TIMER5 interrupt 1: Event TIMER5 interrupt
	0	R	4	IF0_TIMER4	0: No event of TIMER4 interrupt 1: Event TIMER4 interrupt
	0	R	3	IF0_TIMER3	0: No event of TIMER3 interrupt 1: Event TIMER3 interrupt
	0	R	2	IF0_TIMER2	0: No event of TIMER2 interrupt 1: Event TIMER2 interrupt
	0	R	1	IF0_TIMER1	0: No event of TIMER1 interrupt 1: Event TIMER1 interrupt
	0	R	0	IF0_TIMER0	0: No event of TIMER0 interrupt

Index	Default	R/W	Bit	Name	Description
					1: Event TIMER0 interrupt
130			31:16		Reserved
	0	R	15		Reserved
	0	R	14	IF1_DMA6	0: No event of DMA6 interrupt 1: Event of DMA6 interrupt
	0	R	13	IF1_DMA5	0: No event of DMA5 interrupt 1: Event DMA5 interrupt
	0	R	12	IF1_DMA4	0: No event of DMA4 interrupt 1: Event DMA4 interrupt
	0	R	11	IF1_DMA3	0: No event of DMA3 interrupt 1: Event DMA3 interrupt
	0	R	10	IF1_DMA2	0: No event of DMA2 interrupt 1: Event DMA2 interrupt
	0	R	9	IF1_DMA1	0: No event of DMA1 interrupt 1: Event DMA1 interrupt
	0	R	8	IF1_DMA0	0: No event of DMA0 interrupt 1: Event DMA0 interrupt
	0		7		Reserved
	0	R	6	IF1_ALARM	0: No event of RTC alarm interrupt 1: Event ETC alarm interrupt
	0	R	5	IF1_ADC	0: No event of ADC interrupt 1: Event ADC interrupt
	0	R	4	IF1_FSCM	Must be set as 0
	0	R	3	IF1_WWDT	0: No event of WWDT interrupt 1: Event WWDT interrupt
	0	R	2	IF1_IN_TOG	0: No event of all-input toggle interrupt 1: Event all-input toggle interrupt
	0	R	1	IF1_RTC1S	0: No event of RTC 1s interrupt 1: Event RTC 1s interrupt
	0	R	0	IF1_LVD	0: No event of LVD interrupt 1: Event LVD interrupt
134			31:20		Reserved
	0	R	15	IF1_I2C7	0: No event of I2C7 interrupt 1: Event I2C7 interrupt
	0	R	14	IF1_I2C6	0: No event of I2C6 interrupt 1: Event I2C6 interrupt
	0	R	13	IF1_I2C5	0: No event of I2C5 interrupt 1: Event I2C5 interrupt
	0	R	12	IF1_I2C4	0: No event of I2C4 interrupt 1: Event I2C4 interrupt
	0	R	11	IF1_I2C3	0: No event of I2C3 interrupt 1: Event I2C3 interrupt
	0	R	10	IF1_I2C2	0: No event of I2C2 interrupt 1: Event I2C2 interrupt
	0	R	9	IF1_IR	0: No event of IR interrupt 1: Event IR interrupt
	0	R	8	IF1_I2C0	0: No event of I2C0 interrupt 1: Event I2C0 interrupt
	0	R	7	IF1_SPI3	0: No event of SPI3 interrupt

Index	Default	R/W	Bit	Name	Description
					1: Event SPI3 interrupt
	0	R	6	IF1_SPI2	0: No event of SPI2 interrupt 1: Event SPI2 interrupt
	0	R	5	IF1_SPI1	0: No event of SPI1 interrupt 1: Event SPI1 interrupt
	0	R	4	IF1_SPI0	0: No event of SPI0 interrupt 1: Event SPI0 interrupt
	0	R	3	IF1_UART3	0: No event of UART3 interrupt 1: Event UART3 interrupt
	0	R	2	IF1_UART2	0: No event of UART2 interrupt 1: Event UART2 interrupt
	0	R	1	IF1_UART1	0: No event of UART1 interrupt 1: Event UART1 interrupt
	0	R	0	IF1_UART0	0: No event of UART0 interrupt 1: Event UART0 interrupt
138			31:12		Reserved
	0	R	11	IF1_TIMERB	0: No event of TIMERB interrupt 1: Event TIMERB interrupt
	0	R	10	IF1_TIMERA	0: No event of TIMERA interrupt 1: Event TIMERA interrupt
	0	R	9	IF1_TIMER9	0: No event of TIMER9 interrupt 1: Event TIMER9 interrupt
	0	R	8	IF1_TIMER8	0: No event of TIMER8 interrupt 1: Event TIMER8 interrupt
	0	R	7	IF1_TIMER7	0: No event of TIMER7 interrupt 1: Event TIMER7 interrupt
	0	R	6	IF1_TIMER6	0: No event of TIMER6 interrupt 1: Event TIMER6 interrupt
	0	R	5	IF1_TIMER5	0: No event of TIMER5 interrupt 1: Event TIMER5 interrupt
	0	R	4	IF1_TIMER4	0: No event of TIMER4 interrupt 1: Event TIMER4 interrupt
	0	R	3	IF1_TIMER3	0: No event of TIMER3 interrupt 1: Event TIMER3 interrupt
	0	R	2	IF1_TIMER2	0: No event of TIMER2 interrupt 1: Event TIMER2 interrupt
	0	R	1	IF1_TIMER1	0: No event of TIMER1 interrupt 1: Event TIMER1 interrupt
	0	R	0	IF1_TIMER0	0: No event of TIMER0 interrupt 1: Event TIMER0 interrupt

4.13 RC Oscillator Calibration

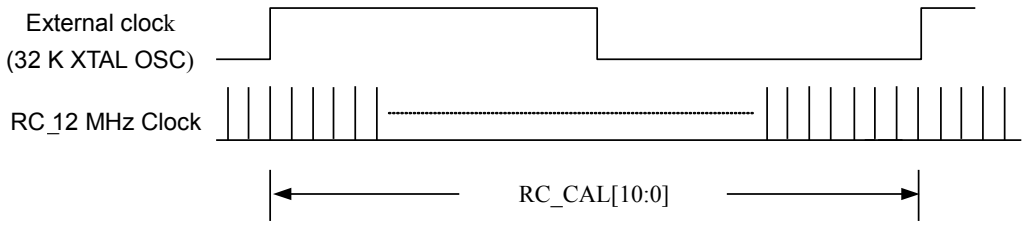
The module uses 32.768 kHz crystal oscillator to calibrate High Speed RC oscillator.
The RC oscillator calibration related registers are included in the system control register and abstracted as below for quick reference:
Firmware can write “1” to “RC_CAL_EN” to start the calibration. It will take 1~2 32.768 kHz clock period to finish the calibration process. Check “RC_Calibrate_Done” flag to make sure calibration process done to make sure RC calibration process finished before read the “RC_CALIBRATE” value.

System control register: address range 0x0020_0000 ~ 0x0020_03FF

Index	Default	R/W	Bit	Name	Description
10			31:14		Reserved
	0	R	13:3	RC_CALIBRATE	RC counter calibrate with external clock
	0	R	2	RC_Calibrate_Done	1: RC Calibration done. Write “0” to “RC_CAL_EN” will clear this flag
	0	R/W	1	CAL_CLK_SEL	Clock calibration source select 0: MCU Clock 1: High Speed RCOSC Clock
	0	R/W	0	RC_CAL_EN	RC calibration enable 1: Enable(Start) RC Calibration 0: Disable RC Calibration Write “1” to start RC Calibration. Write “0” to reset the RC Calibration

RCOSC=12 MHz

	32 kHz	± 0.27%	50 kHz	± 0.42%	100 kHz	± 0.83%
RC_CAL[10:0]	366	11.993 MHz	239	11.950 MHz	119	11.900 MHz
RC_CAL[10:0]	367	12.026 MHz	240	12.000 MHz	120	12.000 MHz
RC_CAL[10:0]	368	12.059 MHz	241	12.050 MHz	121	12.100 MHz



Count Table for 50 kHz XTAL clk

CNT	12 MHz	%	CNT	12 MHz	%	CNT	12 MHz	%
236	11.80 MHz	-1.67	239	11.95 MHz	-0.42	242	12.10 MHz	+0.83
237	11.85 MHz	-1.25	240	12.00 MHz	0	243	12.00 MHz	+1.25
238	11.90 MHz	-0.83	241	12.05 MHz	+0.42	244	12.05 MHz	+1.67

Input 50 kHz (20us) precision < ± 2% counter table
The flash information block stores RC oscillator calibration data, check “initial_load” for detailed information.

4.14 ADC

4.14.1 Features

- Based on a Successive Approximation Register(SAR) architecture
- Maximum conversion rate 1M SPS.
- ±1 LSB accuracy at 3V
- 13 input sources (AN0 ~ AN12)
- Input AN12 is internally connected to temperature sensor output
- Support DMA
- The Operating voltage is 2.4V ~ 3.3V

4.14.2 Block Diagram

Figure 15 shows the block diagram of the 10-bit A/D. The 10-bit A/D converter have 13 analog input pins, designated AN0-AN12. In addition, there is one analog input pin for temperature sensor which designated AN12.

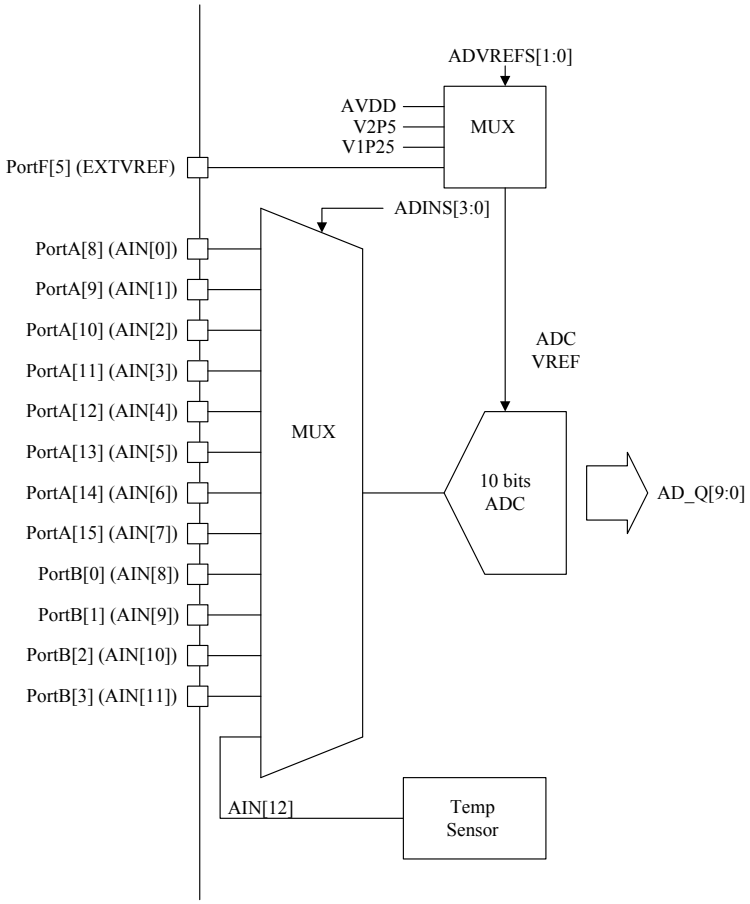


Figure 11: ADC Block Diagram

4.14.3 Register Definition

ADC Control & Data Register: address range 0x0020_8000 ~ 0x0020_83FF

Index	Default	R/W	Bit	Name	Description
00			31:11		Reserved
	0	R/W	10	TMPSON	Temp Sensor Enable Bit 0: Temp Sensor Off 1: Temp Sensor On Measure Temp Sensor, ADCLKS='b111.
			9:7		Reserved
	0	R/W	6	DMA_ADE	DMAEN: DMA_AD enable bit 1 = enable DMA AD request 0 = disable DMA AD request
	0	R/W	5	ADON	ADON: AD Power Control 0: AD Power Down (Default) 1: AD Enable
	0	R/W	4	ADSTR	ADCSTR: 1: AD Start converting, clear when conversion stop.
	0	R/W	3	ADIOVE	ADCIOVE: ADC buffer overflow interrupt enable bit 0: Disable ADC buffer overflow interrupt request 1: Enable ADC buffer overflow interrupt request
	0	R(W1)	2	ADIOV	ADCIOV: ADC interrupt overflow flag 0: ADC buffer not overflow 1: ADC buffer overflow Write 1 to clear the flag
	0	R/W	1	ADIE	ADCIE: ADC interrupt enable bit 0: Disable ADC interrupt request 1: Enable ADC interrupt request
	0	R(W1)	0	ADIF	ADCIF: ADC interrupt flag 0: ADC has no interrupt request 1: ADC issue interrupt request Write 1 to clear the flag
04			31:13		Reserved
	00	R/W	12:11	ADSHSEL	AD Sample and Hold Width 00: 2 Clock 01: 4 Clock 10: 6 Clock 11: 8 Clock
	0	R/W	10:9	ADVREFS	AD VREF Source Select Bit 00: V1P25 01: V2P5 10: AVDD 11: External Vref Input
	0	R/W	8	ADQFORM	FORM: Data Output Format bits 0: (DOUT = 0000 00dd dddd dddd) 1: (DOUT = dddd dddd dd00 0000)
	0	R/W	7:4	ADINS	ADINS<3:0>: Selects Input Channel 0000: AIN[0] 0001: AIN[1] 0010: AIN[2]

Index	Default	R/W	Bit	Name	Description
					0011: AIN[3] 0100: AIN[4] 0101: AIN[5] 0110: AIN[6] 0111: AIN[7] 1000: AIN[8] 1001: AIN[9] 1010: AIN[10] 1011: AIN[11] 1100: AIN[12] 1101: AIN[13] 1110: reserved 1111: reserved
	0	R/W	3:1	ADCLKS	ADCLKS: AD Conversion Clock Select bits 111 = CLK/256 110 = CLK/128 101 = CLK/64 100 = CLK/32 011 = CLK/16 010 = CLK/8 001 = CLK/4 000 = CLK/2
08	-	-	31:16	-	Reserved
	-	R	15:0	ADQ	AD output buffer

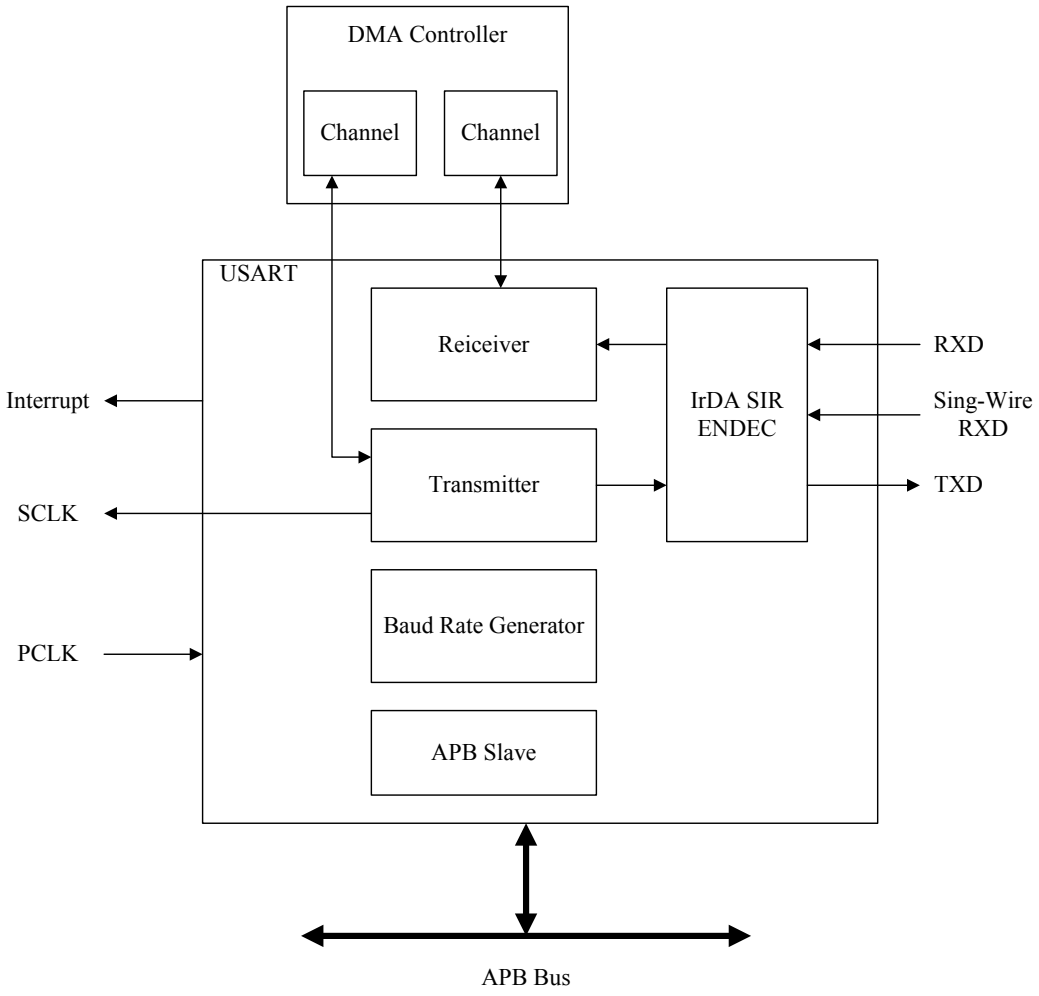
4.15 UART

4.15.1 Main Features

- Full duplex, asynchronous communications
- NRZ standard format
- Configurable oversampling method by 16 or by 8
- A baud rate generator
- 8 bits or 9 bits data word length
- 1 or 2 stop bits
- Separate enable bits for Transmitter or Receiver
- Configurable multibuffer communication using DMA
  - Buffering of received/transmitted bytes in reserved SRAM
- Transmitter clock output for synchronous transmission
- IrDA SIR ENDEC
  - Support for 3/16 bit duration for normal mode
- Single-Wire half-duplex communication
- Transfer detection flags:
  - Receive buffer full
  - Transmit buffer empty
  - End of Transmission
- Parity control
  - Even, odd or no-parity generation and detection
- Four error detection flags
  - Overrun error
  - Noise error
  - Frame error
  - Parity error
- Interrupts sources with flags
  - Transmit data register empty
  - Transmit complete
  - Receive data register full
  - Break Detection
  - Overrun error
  - Frame error
  - Noise error
  - Parity error
- Multi-processor communication mode
  - USART can enter Mute mode
  - Two receiver wakeup form Mute mode (by idle line detection or address mark detection)

4.15.2 Functional Description

Block Diagram



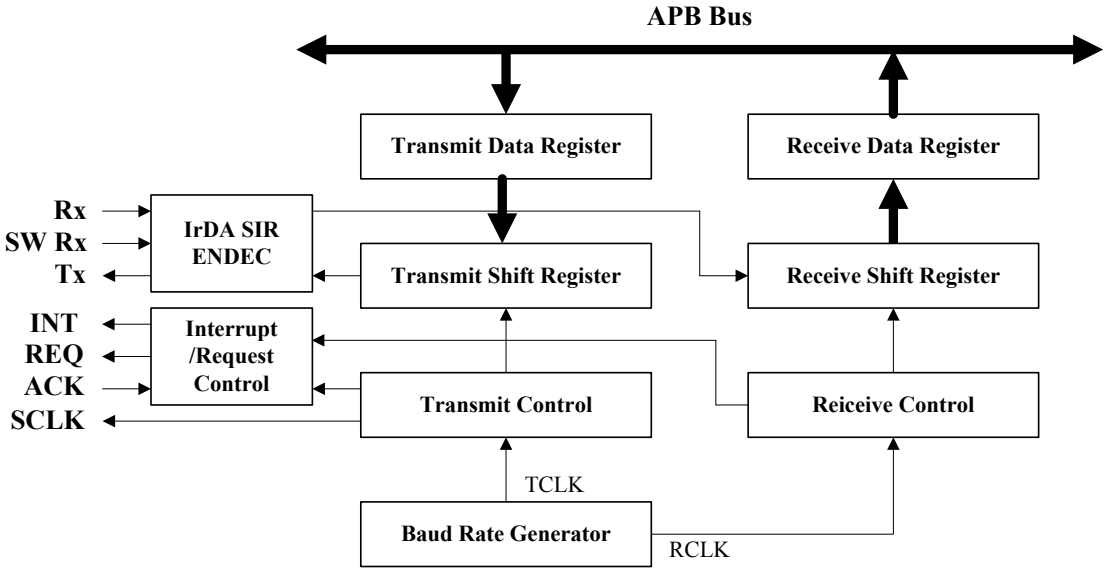
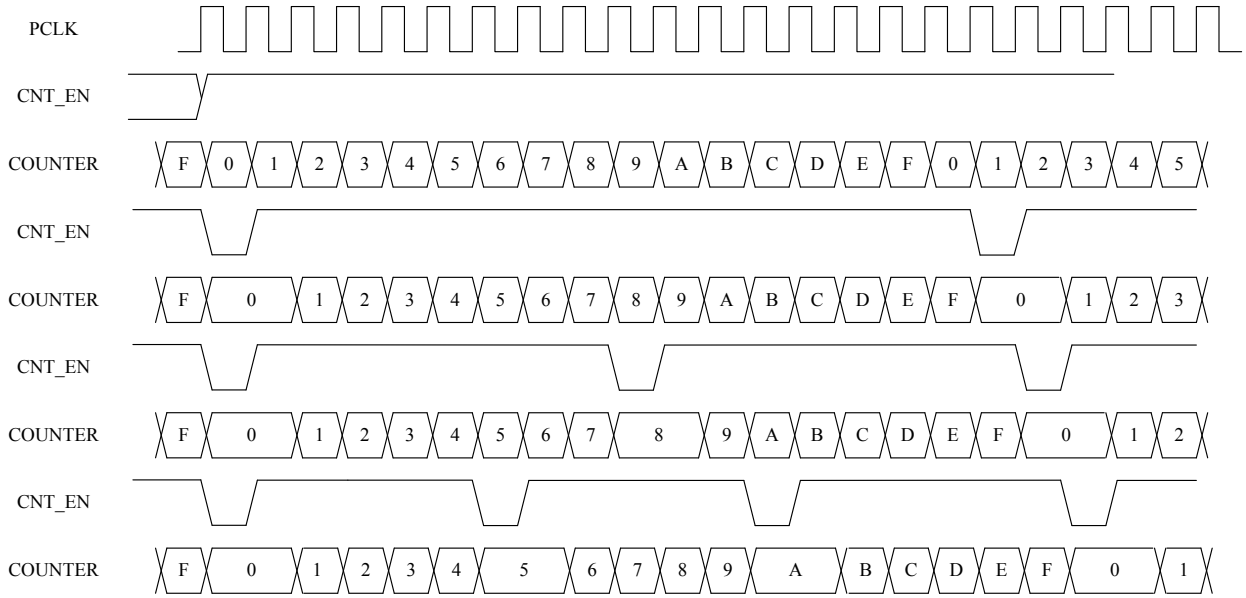


Figure 12: UART Block Diagram

Fractional Baud Rate Generator

$$\text{Baud rate} = \frac{Pclk}{8*(2-OVER8)*(Mantissa+\frac{Franction}{16})}$$

Ex: $\frac{Pclk}{16*(1+\frac{1}{16})}$, $\frac{Pclk}{16*(1+\frac{2}{16})}$, $\frac{Pclk}{16*(1+\frac{3}{16})}$



Ex: $\frac{Pclk}{16*(3+\frac{3}{16})}$

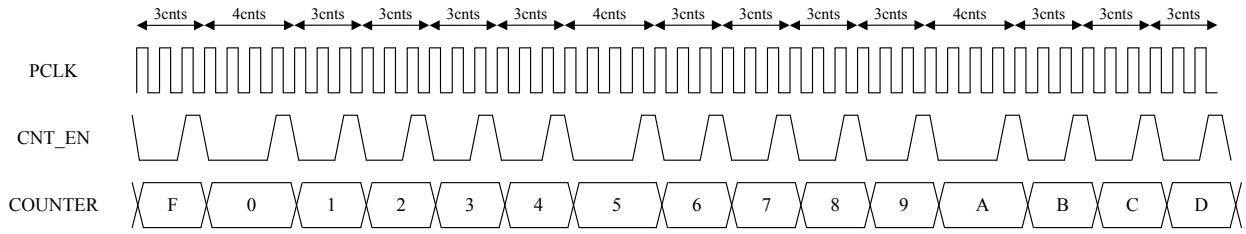


Figure 13: Example for Baud Rate Generator

4.15.3 Register Definition

UART0: address range 0x0020_3000 ~ 0x0020_33FF
UART1: address range 0x0020_B000 ~ 0x0020_B3FF
UART2: address range 0x0020_3400 ~ 0x0020_37FF
UART3: address range 0x0020_B400 ~ 0x0020_B7FF

Control Register 1

Index	Default	R/W	Bit	Name	Description
00			31:18	Reserved	
	0	R/W	17	UE	UART enable 0: UART prescaler and outputs disabled 1: UART enabled
	0		16	reserved	
	0	R/W	15	TE	Transmitter enable

Index	Default	R/W	Bit	Name	Description
					0: Transmitter is disabled 1: Transmitter is enabled
	0	R/W	14	RE	Receiver enable 0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit
	0	R/W	13	OVER8	Over sampling mode 0: over sampling by 16 1: over sampling by 8 Note: Over sampling by 8 is not available in IrDA, LIN mode, when IREN.
	0	R/W	12	M	Word length 0: 1 Start bit, 8 Data bits, n Stop bit 1: 1 Start bit, 9 Data bits, n Stop bit
	0	R/W	11	DMAT	DMA enable transmitter 1: DMA mode is enabled for transmission 0: DMA mode is disabled for transmission
	0	R/W	10	DMAR	DMA enable receiver 1: DMA mode is enabled for reception 0: DMA mode is disabled for reception
	0	R/W	9	RWU	Receiver wakeup 0: Receiver in active mode 1: Receiver in mute mode
	0	R/W	8	WAKE	Wakeup method 0: Idle Line 1: Address Mark
	0	R/W	7:4	ADDR[3:0]	Address of the UART node This bit-field gives the address of the USART node.
	0	R/W	3		Reserved
	0	R/W	2	PCE	Parity control enable 0: disable 1: enable
	0	R/W	1	PS	Parity selection 0: Even 1: Odd
	0	R/W	0	STOP	0: STOP = 1bit 1: STOP = 2bit

Control Register 2

Index	Default	R/W	Bit	Name	Description
04			31:9	Reserved	
	0	R/W	8	IDLEIE	IDLE interrupt enable 0: Interrupt is inhibited 1: An UART interrupt is generated whenever IDLE=1
	0	R/W	7	BDIE	BD interrupt enable 0: Interrupt is inhibited 1: An UART interrupt is generated whenever BD=1
	0	R/W	6	TXEIE	TXE interrupt enable 0: Interrupt is inhibited 1: An UART interrupt is generated whenever TXE=1
	0	R/W	5	TCIE	Transmission complete interrupt enable 0: Interrupt is inhibited 1: An UART interrupt is generated whenever TC=1
	0	R/W	4	RXNEIE	RXNE interrupt enable

Index	Default	R/W	Bit	Name	Description
					0: Interrupt is inhibited 1: An UART interrupt is generated whenever ORE=1 or RXNE=1
	0	R/W	3	EIE	Error Interrupt enable USART: 0: Interrupt is inhibited 1: An interrupt is generated whenever FER=1 or ORE=1 or NE=1.
			2:1	Reserved	
	0	R/W	0	PEIE	PE interrupt enable 0: Interrupt is inhibited 1: An UART interrupt is generated whenever PE=1

Status Register 1

Index	Default	R/W	Bit	Name	Description
08			31:9	Reserved	
	0	R	8	IDLE	IDLE line detected 0: No Idle Line is detected 1: Idle Line is detected Clear by S/W write to 0
	0	R	7	BD	Break Detect 0: Break not detected 1: Break detected Clear by S/W write to 0
	1	R	6	TXE	Transmit data register empty 0: Data is not transferred to the shift register 1: Data is transferred to the shift register)
	0	R	5	TC	USART: Transmission complete 0: Transmission is not complete 1: Transmission is complete Clear by S/W write to 0 or Clear by H/W when transmit data register is not empty.
	0	R	4	RXNE	USART: Read data register not empty 0: Data is not received 1: Received data is ready to be read. Clear by S/W read Receive DATA Register or S/W write to 0
			3:0	Reserved	

Transmit Data Register

Index	Default	R/W	Bit	Name	Description
0C			31:9	Reserved	
	0	W	8	TDR[8]	USART: Transmit USART Data value
	0	W	7:6	TDR[7:6]	USART: Transmit USART Data value.
	0	W	5:0	TDR[5:0]]	USART: Transmit USART Data value

Receive Data Register

Index	Default	R/W	Bit	Name	Description
10			31:9	Reserved	
	0	R	8:0	RDR[8:0]	Receive Data value

Baud Rate Register

Index	Default	R/W	Bit	Name	Description
14			31:16	Reserved	
	0	R/W	15:4	Mantissa[11:0]	mantissa of Baud Rate Generator
	0	R/W	3:0	Fraction[3:0]	fraction of Baud Rate Generator

Control Register 3

Index	Default	R/W	Bit	Name	Description
18			31:15		
	0	R/W	14	IREN	IrDA mode enable 0: IrDA disable 1: IrDA enable
	0	R/W	13	HDSEL	Half-duplex selection 0: Half-duplex disable 1: Half-duplex enable
	0	R/W	12	CLKEN	Synchronous mode enable 0: SCLK pin disable. 1: SCLK pin enable.
	0	R/W	11	CPOL	Clock polarity
	0	R/W	10	CPHA	Clock phase
	0	R/W	9	LBCL	Last bit clock pulse 0: The last bit(MSB) clock pulse is not output to SCLK pin. 1: The last bit(MSB) clock pulse is output to SCLK pin.
	'h0100	R/W	8:0		reserved

4.15.4 Fractional baud rate generation

$$\text{Baud rate} = \frac{Sysclk}{8*(2-OVER8)*(Mantissa+\frac{Franction}{16})}$$

Baud Rate Support Table (Bps):

	Sysclk 24 MHz (OVER8=0)			Sysclk 24 MHz (OVER8=1)		
	Baud Rate Register	Actual	Error	Baud Rate Register	Actual	Error
2.4 K	625	2400	0.0%	1250	2400	0.0%
9.6 K	156.25	9600	0.0%	312.5	9600	0.0%
19.2 K	78.125	19200	0.0%	156.25	19210	0.0%
38.4 K	39.0625	38400	0.0%	78.125	38400	0.0%
57.6 K	26.0625	57554	0.08%	52.125	57554	0.08%
115.2 K	13	115384	0.16%	26	115384	0.16%

	Sysclk 24 MHz (OVER8=0)			Sysclk 24 MHz (OVER8=1)		
	Baud Rate Register	Actual	Error	Baud Rate Register	Actual	Error
230.4 K	6.5	230769	0.16%	13	230769	0.16%
921.6K	1.625	923077	0.16%	3.25	923077	0.16%
2 M	NA	NA	NA	1.5	2M	0.0%
3 M	NA	NA	NA	1	3M	0.0%

4.15.5 Receiver

Overrun error
An overrun error occurs when a character is received when RXNE has not been reset. Data cannot be transferred from the shift register to the RDR register until the RXNE bit is cleared.

Noise error
Data sampling when oversampling by 16, sample value is bit 8, 9, 10.
Data sampling when oversampling by 8, sample value is bit 4, 5, 6.
The Noise error flag bit is set at the rising edge of the RXNE bit.

Sample value	Noise error	Received bit
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

Framing error
A framing error is detected when the stop bit of each data byte is low in reception process.

4.15.6 Multi-processor communication mode

USART enter mute mode if RWU bit is 1.
USART enter normal mode if RWU bit is 0.
RWU bit cannot program if RXNE flag is 1.

Idle line detection: WAKE = 0

1. USART enter mute mode when the RWU bit is written to 1
2. If Idle frame is detected. USART wakes up to normal mode and RWU bit is set to 0 by hardware but IDLE bit does not set in status register.
3. In mute mode, RXNE flag does not set

Address mark detection: WAKE = 1

1. The USART is received address byte (MSB is a '1') and data byte (MSB is a '0')

2. If address byte is match ADDR register(Control Register 1), RWU will set 0 by hardware. If address byte is mismatch, RWU will set 1 by hardware
3. The RXNE flag does not set for address byte. In mute mode, RXNE flag does not set for data byte

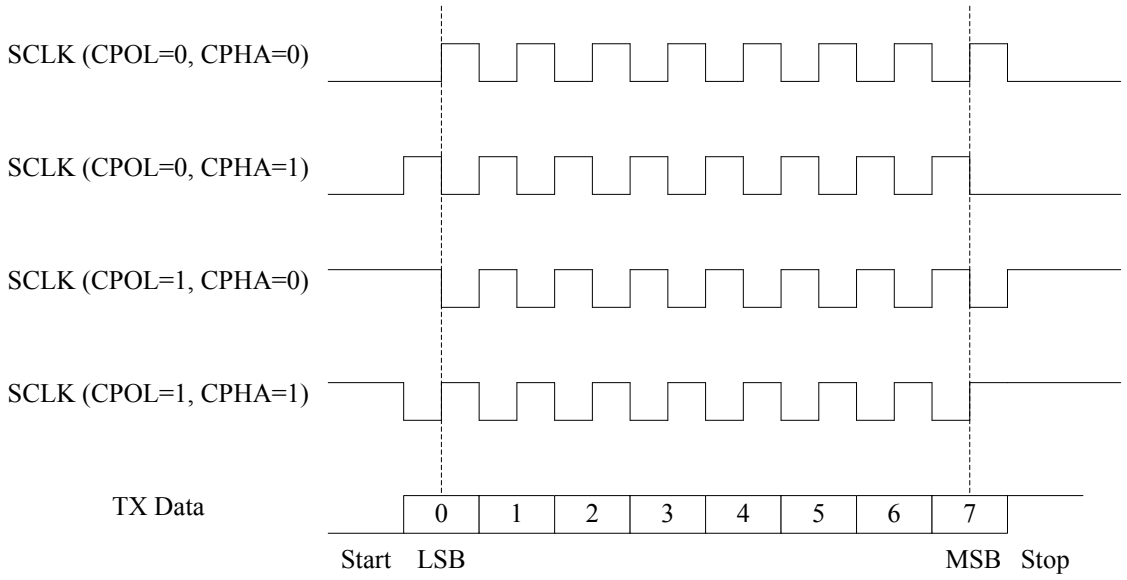
4.15.7 Parity Control

M bit	PCE bit	USART Data Frame
0	0	Start + 8 bit data + Stop
0	1	Start + 7 bit data + parity bit + Stop
1	0	Start + 9 bit data + Stop
1	1	Start + 8 bit data + parity bit + Stop

4.15.8 USART synchronous mode

The synchronous mode is selected by writing the CLKEN bit.
In synchronous mode, HDSEL and IREN bits must be kept cleared.

The USART allows the user to control a serial communications in master mode.
The SCLK pin is output of the USART transmitter clock. In synchronous mode the USART transmitter works exactly like in asynchronous mode. But as SCLK is synchronized with TX.



USART sync mode clock timing diagram (M=0)

Figure 14: USART sync mode clock timing

4.15.9 Single-wire half-duplex communication

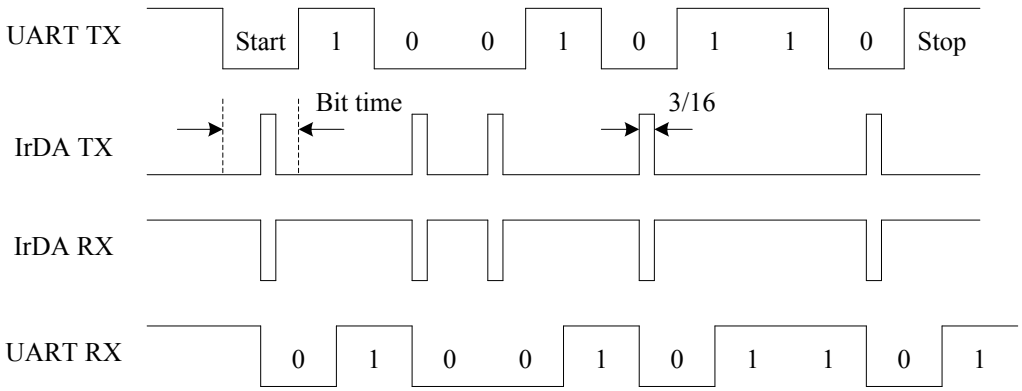
The single-wire half-duplex mode is selected by setting the HDSEL bit in the Control Register 3. In this mode, CLKEN and IREN bit must kept cleared.

When HDSEL is written to 1:
1. the TX and RX lines are internally connected.
2. the RX pin is no longer used.
3. the TX pin is floating input when no data is transmitted

4.15.10 IrDA SIR ENDEC block

The IrDA mode is selected by setting the IREN bit. STOP, CLKEN and HDSEL bits must be kept cleared.

IrDA is a half duplex communication protocol. If the Transmitter is busy, any data on the IrDA receive line will be ignored by the IrDA decoder. If the Receive is busy, data on the TX from the USART to IrDA will not be encoded by IrDA.



IrDA 3/16 Data Modulation

Figure 15: IrDA Data Modulation

4.15.11 Communication by DMA

Transmission using DMA

DMA mode can be enabled for transmission by setting DMAT bit. Data is loaded from a source device (ex. SRAM) to the Transmit Data Register (0x0C). The source device data must ready before DMA start transmission.

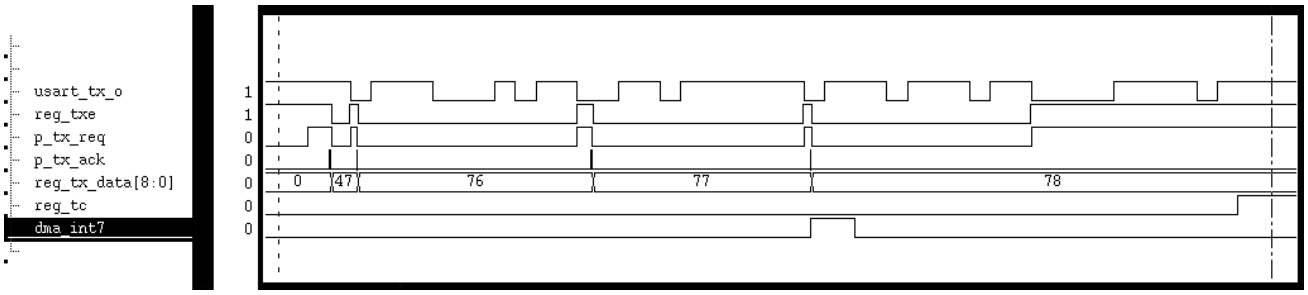


Figure 16: Illustration of Transmission using DMA

Reception using DMA

DMA mode can be enabled for reception by setting the DMAR bit. Data is loaded from the Receive Data Register (0x10) to a destination device (ex. SRAM).

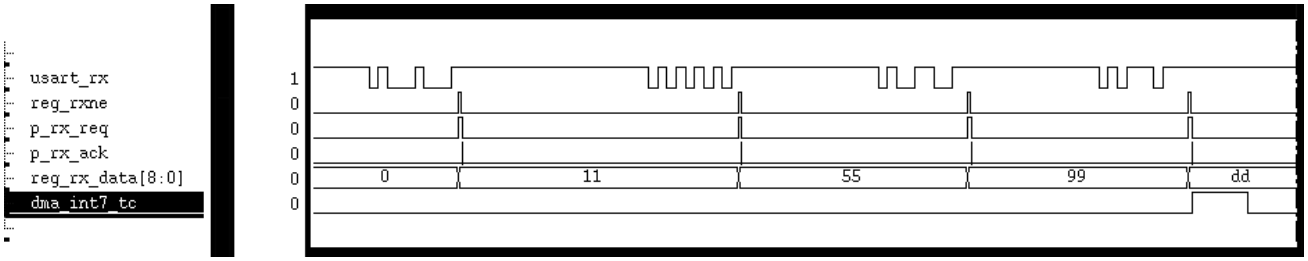


Figure 17: Illustration of Reception using DMA

4.16 SPI

The Serial Peripheral Interface (SPI) module is a half/ full-duplex, synchronous serial interface useful for communicating with other peripheral or microcontroller devices.

4.16.1 Main Features

- Master or slave operation
- Programmable clock polarity and phase
- Programmable data order with MSB-first or LSB-first shifting
- Dedicated transmission and reception flags with interrupt capability
- Full-duplex synchronous transfers on three lines
- Simplex synchronous transfers on two lines with or without a bidirectional data line
- 8 or 16 bit transfer frame format selection
- Multimaster mode capability with bus busy status flag
- Master or slave mode baud rate up to fCLK / 2
- NSS management by hardware or software for both master and slave
- Hardware CRC feature for reliable communication
- Miscellaneous error flags with interrupt capability
- Transmission and reception buffer with DMA capability
- Capable for next access delay configuration
- Capable for master RX-only pseudo count

4.16.2 Block Diagram

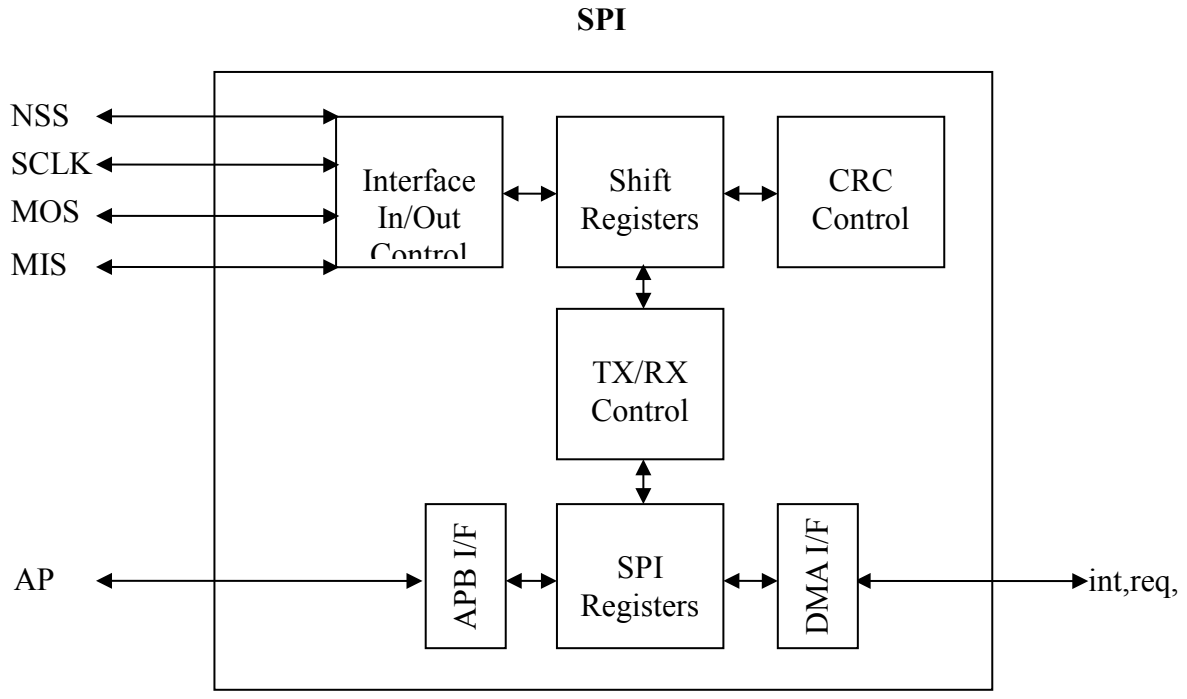


Figure 18: SPI Block Diagram

4.16.3 Control Register Definition

SPI0: address range 0x0020_3800 ~ 0x0020_3BFF
SPI1: address range 0x0020_B800 ~ 0x0020_BBFF
SPI2: address range 0x0020_3C00 ~ 0x0020_3FFF
SPI3: address range 0x0020_BC00 ~ 0x0020_BFFF

Index	Default	R/W	Bit	Name	Description
00h (spi_cr0)	0	R/W	15	spe	SPI enable, if modfif occurs, this bit will be cleared note: sclk should be kept idle state while spe is turned off
	0	R/W	14	hscc	high-speed clock compensation
	0	R/W	13	bidimode	bidirectional data mode enable (simplex)
	0	R/W	12	bidioe	bidirectional mode output enable 1: TX, 0: RX
	0	R/W	11	crce	hardware CRC enable
	0	R/W	10	crcnext	CRC transfer next write '1' after last TX data is written or after second last data is read note: it is automatically set under DMA mode and this bit is automatically cleared after CRC is generated/checked
	-	-	9	-	reserved
			8	nssctl	slave select (NSS) output control (master mode) 0: NSS = 1 1: NSS = 0
	0	R/W	7	mstr	master selection
	0	R/W	6	ssoe	slave select output enable (master mode), if ssoe is set, nss input is ignored and modfif should not be set in master mode
	0	R/W	5	ssm	software slave mode
	0	R/W	4	ssi	software slave input
	0	R/W	3	mode16	frame length setting (8/16 bits) 1: 16, 0: 8
	0	R/W	2	lsbfe	LSB-first enable
	0	R/W	1	cpol	clock polarity 1: sclk = '1' when bus is idle 0: sclk = '0' when bus is idle
	0	R/W	0	cpha	clock phase 1: second clock transition is the data capture edge 0: first clock transition is the data capture edge
04h (spi_cr1)	-	-	15:12	-	reserved
	0h	R/W	11:8	nad	next access delay (word to word delay, unit: 0.5bit) includes NSS to SCLK to NSS delay (master mode only) note: the busy flag is inactive during NAD period
	00h	R/W	7:0	brs	bit rate selection clock speed = Fsc / ((brs+1)*2)
08h (spi_cr2)	00h	R/W	15:8	pscnt	pseudo master rx-only counter (for sclk generation)
	0	R/W	7	txnfie	TX not full interrupt enable
	0	R/W	6	rxneie	RX not empty interrupt enable
	0	R/W	5	modfie	mode fault interrupt enable
	0	R/W	4	crcfie	CRC fault interrupt enable
	0	R/W	3	txovrie	TX overrun interrupt enable
	0	R/W	2	rxovrie	RX overrun interrupt enable
	0	R/W	1	txdmaen	TX DMA enable
	0	R/W	0	rxdmaen	RX DMA enable
0ch	-	-	15:8	-	reserved

Index	Default	R/W	Bit	Name	Description
(spi_cr3)	1	R	7	txnfif	TX not full flag, write buf_dat to clear
	0	R	6	rxneif	RX not empty flag, read buf_dat to clear
	-	R/W	5	modfif	mode fault flag, write '1' to clear fault condition: master mode: NSS = input-0 slave mode: NSS = high but frame is incomplete
	-	R/W	4	crcfif	CRC fault flag, write '1' to clear
	-	R/W	3	txovrif	TX overrun flag, write '1' to clear
	-	R/W	2	rxovrif	RX overrun flag, write '1' to clear
	-	R	1	busy	busy status under TX/RX
	-	-	0	-	reserved
10h	00h	R/W	15:0	buf_dat	buffer data for TX/RX
14h	-	-	-	-	Reserved
18h	ffffh	R	15:0	crc_tx	CRC TX data
1ch	ffffh	R	15:0	crc_rx	CRC RX data

Status Flags
Three status flags are provided for the application to completely monitor the state of the SPI bus.

TX buffer not full flag (TXNF)
When it is set, this flag indicates that the TX buffer is not full and the next data to be transmitted can be loaded into the buffer. The TXNF flag is cleared when writing to the BUF_DAT register.

RX buffer not empty (RXNE)
When set, this flag indicates that there are valid received data in the RX buffer. It is cleared when BUF_DAT is read.

BUSY flag
This BUSY flag is set and cleared by hardware (writing to this flag has no effect). The BUSY flag indicates the state of the communication layer of the SPI.
When BUSY is set, it indicates that the SPI is busy communicating.

The BUSY flag is useful to detect the end of a transfer if the software wants to disable the SPI and enter Halt mode (or disable the peripheral clock). This avoids corrupting the last transfer.

The BUSY flag is also useful to avoid write collisions in a multimaster system.

The BUSY flag is set when a transfer starts.

It is cleared:
● when a transfer is finished
● when the SPI is disabled
● when a master mode fault occurs (MODFIF=1)

When communication is not continuous, the BUSY flag is low between each communication.

When communication is continuous:
● in master mode, the BUSY flag is kept high during all the transfers
● in slave mode, the BUSY flag goes low for half SPI clock cycle between each transfer

Error Flags

Master mode fault (MODFIF)
Master mode fault occurs when the master device has its NSS pin pulled low (in NSS hardware mode) or SSI bit low (in NSS software mode), this automatically sets the MODFIF bit. Master mode fault affects the SPI peripheral in the following ways:
● The MODFIF bit is set and an SPI interrupt is generated if the ERRIE bit is set.
● The SPE bit is cleared. This blocks all output from the device and disables the SPI interface.

Write '1' to MODFIF bit to clear the mode fault status.
To avoid any multiple slave conflicts in a system comprising several MCUs, the NSS pin must be pulled high during the MODFIF bit clearing sequence. The interrupt caused by MODFIF is not controlled by SPE setting.

As a security, hardware does not allow the setting of the SPE bit while the MODFIF bit is set.

In a slave device, if an incomplete frame is processing and NSS is deactivated from low to high, the MODFIF bit will be set. An interrupt routine can be used to recover cleanly from this state by performing a reset or returning to a default state.

CRC error
This flag is used to verify the validity of the value received when the CRCE bit in the control register. The CRCERR flag register is set if the CRC value received is incorrect. The CRCFIF responses CRC check result after crcnext is cleared by hardware.

Overrun condition

TXOVRIF
An overrun condition occurs when the device has not fill data bytes before frame transmission starts. When an overrun condition occurs:
● the TXOVRIF bit is set and an interrupt is generated if the TXOVRIE bit is set.
In this case, the transmitter buffer has no contents to transmit from the device, in this condition, transmitted data is meaningless if TXOVRIF is occurred.
Clearing the TXOVRIF bit is done by write '1' to the TXOVRIF register.

RXOVRIF
An overrun condition occurs when the TX device has sent data bytes and the RX device has not cleared the RXNE bit resulting from the previous data byte transmitted. When an overrun condition occurs:
● the RXOVRIF bit is set and an interrupt is generated if the RXOVRIE bit is set.
In this case, the receiver buffer contents will not be updated with the newly received data from the master device. A read from the BUF_DAT register returns this byte. All other subsequently transmitted bytes are lost.
Clearing the RXOVRIF bit is done by write '1' to the RXOVRIF register.

SPI Interrupts

Interrupt event	Event flag	Enable Control bit
Transmit buffer not full flag	TXNFIF	TXNFIE
Receive buffer not empty flag	RXNEIF	RXNEIE
Mode fault flag	MODFIF	MODFIE
CRC error flag	CRCFIF	CRCFIE
TX overrun error flag	TXOVRIF	TXOVRIE
RX overrun error flag	RXOVRIF	RXOVRIE

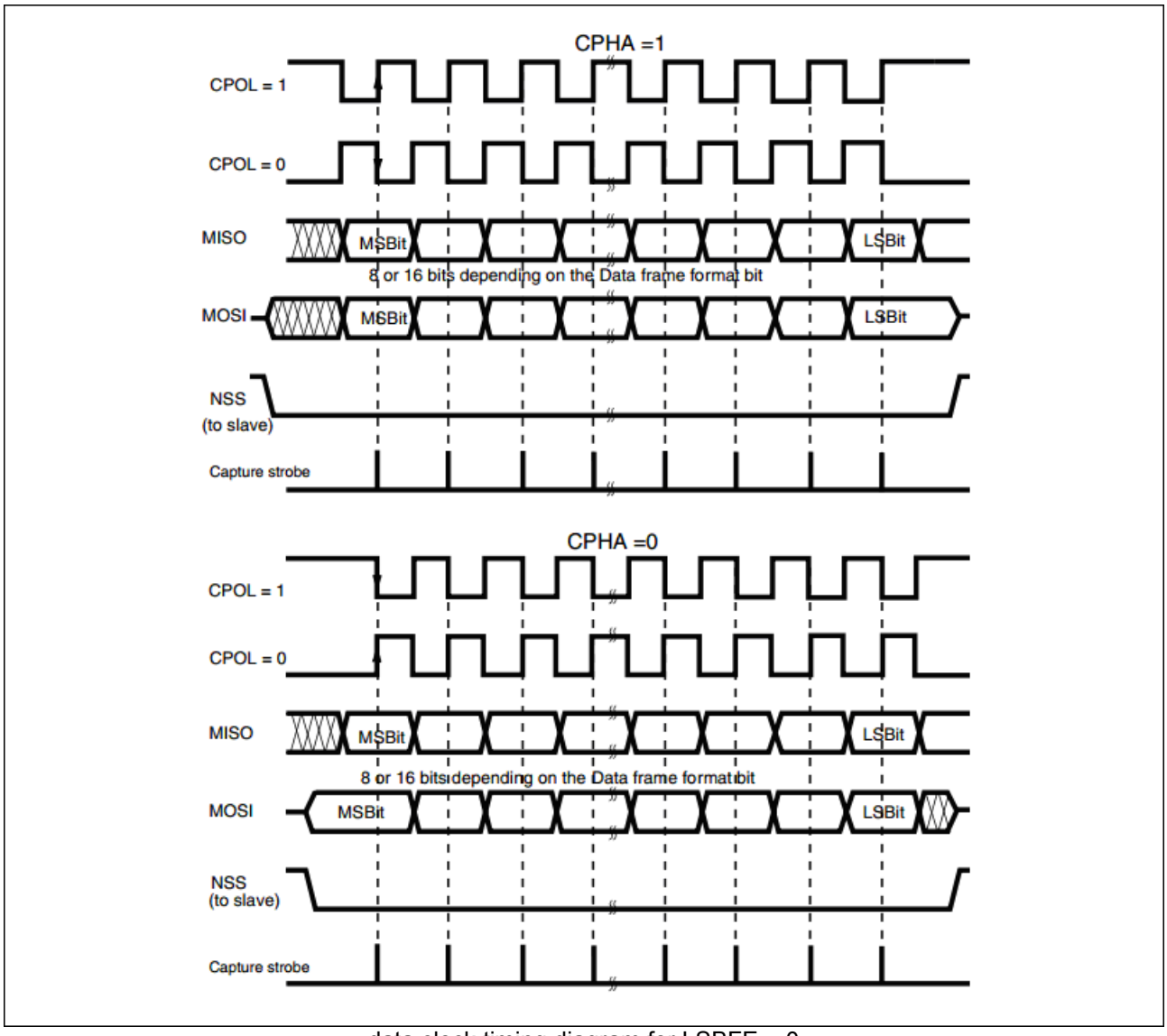
SPI interrupt is controlled by SPE setting, interrupt enable and interrupt flag, except "MODFIF", which is an interrupt source independent of SPE setting.

SPI communication using DMA

To operate at its maximum speed, the SPI needs to be fed with the data for transmission and the data received on the Rx buffer should be read to avoid overrun. To facilitate the transfers, the SPI features a DMA capability implementing a simple request/acknowledge protocol.
A DMA access is requested when the enable bit is turned on. Separate requests must be issued to the TX and RX buffers:
● In transmission, a DMA request is issued each time TXNF is set to 1. The DMA then writes to the BUF_DAT register (this clears the TXNF flag).
● In reception, a DMA request is issued each time RXNE is set to 1. The DMA then reads the BUF_DAT register (this clears the RXNE flag).

When the SPI is used only to transmit data, it is possible to enable only the SPI TX DMA channel. In this case, the RXOVRIF flag is set because the data received are not read.
When the SPI is used only to receive data, it is possible to enable only the SPI RX DMA channel by setting pscnt register.
In transmission mode, when the DMA has written all the data to be transmitted (flag TCIF is set in the DMA register), the BUSY flag can be monitored to ensure that the SPI communication is complete. This is required to avoid corrupting the last transmission before disabling the SPI or entering the Stop mode. The software must first wait until TXNF=1 and then until BUSY=0.

4.16.4 Timing Diagram



data clock timing diagram for LSBFE = 0

Figure 19: SPI data clock timing diagram for LSBFE = 0

4.16.5 Configuration Flow Chart

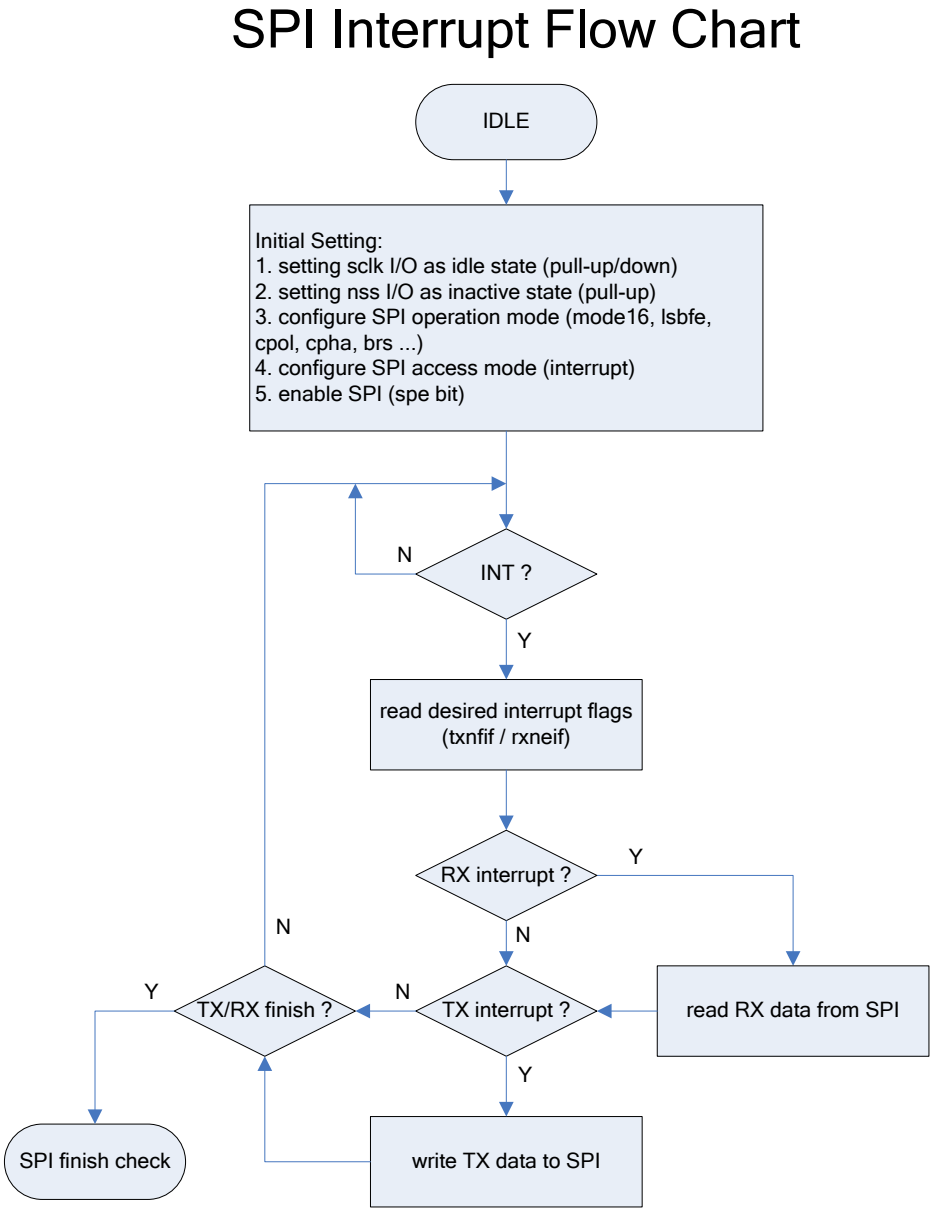


Figure 20: SPI interrupt flow chart

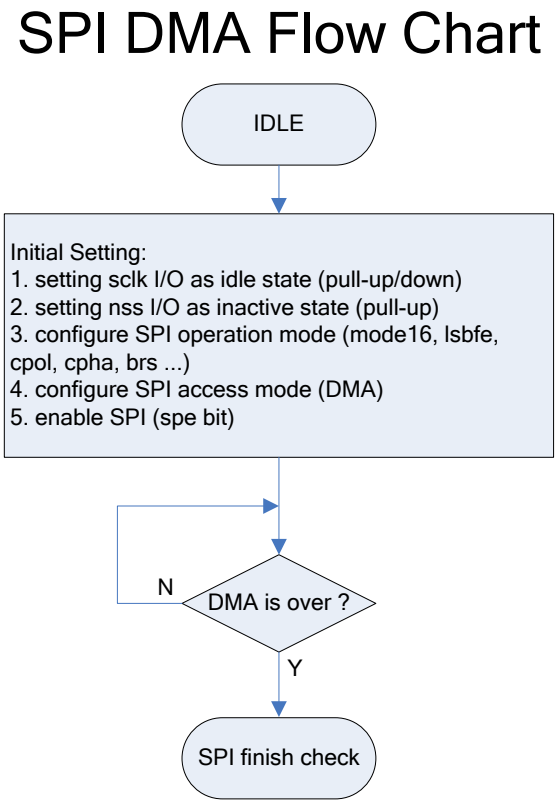


Figure 21: SPI DMA flow chart

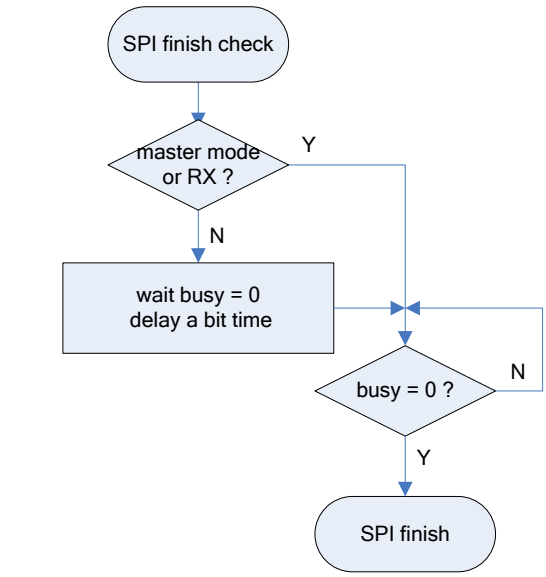


Figure 22: SPI finish check

4.16.6 Setting Examples

A. Simplex Transfer with Interrupts

A.1 slave to master (16-bit)

master setting
W 04h 0001h // bit-time = pclk*4
W 08h 0440h // rxneif interrupt enable (for RX), pseudo 4 half-word
W 00h 00cch // master mode, NSS output enable, mode16 & lsbfe enable
W 00h 80cch // SPI enable

master int
R 10h xxxxh // read received data and clear rxneif flag

slave setting
W 04h 0001h // bit-time = pclk*4
W 08h 0080h // txnfif interrupt enable (for TX)
W 00h 000ch // slave mode, mode16 & lsbfe enable
W 00h 800ch // SPI enable

slave int
W 10h xxxxh // write next data and clear txnfif flag

A.2 master to slave (8-bit)

master setting
W 04h 0003h // bit-time = pclk*8
W 08h 0080h // txnfif interrupt enable (for TX)
W 00h 00c3h // master mode, NSS output enable, cpol & cpha enable
W 00h 80c3h // SPI enable

master int
W 10h xxxxh // write next data and clear txnfif flag

slave setting
W 04h 0003h // bit-time = pclk*8
W 08h 0040h // rxneif interrupt enable (for RX)
W 00h 0003h // slave mode, cpol & cpha enable
W 00h 8003h // SPI enable

slave int
R 10h xxxxh // read received data and clear rxneif flag

B. Full Duplex Transfer with Interrupts

master setting (8-bit)
W 04h 0003h // bit-time = pclk*8
W 08h 00c0h // txnfif & rxneif interrupt enable (for TX & RX)
W 00h 00c7h // master mode, NSS output enable, lsbfe & cpol & cpha enable
W 00h 80c7h // SPI enable

master int
W 10h xxxxh // write next data and clear txnfif flag
R 10h xxxxh // read received data and clear rxneif flag

slave setting
W 04h 0003h // bit-time = pclk*8
W 08h 00c0h // txnfif & rxneif interrupt enable (for TX & RX)
W 00h 0007h // slave mode, lsbfe & cpol & cpha enable
W 00h 8007h // SPI enable

slave int
W 10h xxxxh // write next data and clear txnfif flag
R 10h xxxxh // read received data and clear rxneif flag

C. Half Duplex Transfer with Interrupts

note: connect master-mosi to slave-miso

C.1 slave to master (8-bit)

master setting
W 04h 0003h // bit-time = pclk*8
W 08h 0440h // rxneif interrupt enable (for RX), pseudo 4 bytes
W 00h 20b6h // master bidi-RX mode, S/W NSS inactive, lsbfe & cpol enable
W 00h 20a6h // S/W NSS active
W 00h a0a6h // SPI enable

master int
R 10h xxxxh // read received data and clear rxneif flag

slave setting
W 04h 0003h // bit-time = pclk*8
W 08h 0080h // txnfif interrupt enable (for TX)
W 00h 3036h // slave bidi-TX mode, S/W NSS inactive, lsbfe & cpol enable
W 00h 3026h // S/W NSS active
W 00h b026h // SPI enable

slave int
W 10h xxxxh // write next data and clear txnfif flag

C.2 master to slave (16-bit)

master setting
W 04h 0001h // bit-time = pclk*4
W 08h 0080h // txnfif interrupt enable (for TX)
W 00h 30b9h // master bidi-TX mode, S/W NSS inactive, mode16 & cpha enable
W 00h 30a9h // S/W NSS active
W 00h b0a9h // SPI enable

master int
W 10h xxxxh // write next data and clear txnfif flag

slave setting
W 04h 0001h // bit-time = pclk*4
W 08h 0040h // rxneif interrupt enable (for RX)
W 00h 2039h // slave bidi-RX mode, S/W NSS inactive, mode16 & cpha enable
W 00h 2029h // S/W NSS active
W 00h a029h // SPI enable

slave int
R 10h xxxxh // read received data and clear rxneif flag

D. Full Duplex Transfer with DMA

master setting (8-bit)
W 04h 0003h // bit-time = pclk*8
W 08h 0003h // txnfif & rxneif DMA enable (for TX & RX)
W 00h 00c4h // master mode, NSS output enable, lsbfe enable
W 00h 80c4h // SPI enable

master DMA
W 10h xxxxh // write next data and clear txnfif flag
R 10h xxxxh // read received data and clear rxneif flag

slave setting
W 04h 0003h // bit-time = pclk*8
W 08h 0003h // txnfif & rxneif DMA enable (for TX & RX)
W 00h 0004h // slave mode, lsbfe enable
W 00h 8004h // SPI enable

slave DMA
W 10h xxxxh // write next data and clear txnfif flag
R 10h xxxxh // read received data and clear rxneif flag

E. Full Duplex Transfer with DMA & CRC

master setting (8-bit)
W 04h 0003h // bit-time = pclk*8
W 08h 0003h // txnfif & rxneif DMA enable (for TX & RX)
W 00h 08c4h // master mode, CRC enable, NSS output enable, lsbfe enable
W 00h 88c4h // SPI enable

master DMA
W 10h xxxxh // write next data and clear txnfif flag
R 10h xxxxh // read received data and clear rxneif flag

slave setting
W 04h 0003h // bit-time = pclk*8
W 08h 0003h // txnfif & rxneif DMA enable (for TX & RX)
W 00h 0804h // slave mode, CRC enable, lsbfe enable
W 00h 8804h // SPI enable

slave DMA
W 10h xxxxh // write next data and clear txnfif flag
R 10h xxxxh // read received data and clear rxneif flag

4.17 Timer

The general-purpose timers consist of a 16-bit counter driven by a programmable prescaler, they are designed to count cycles of the peripheral clock (PCLK) or an external clock, and may be used for a variety of purposes, including measuring the pulse lengths of input signals (input capture) or generating output waveforms (output compare and PWM) based on match registers. The timers can also generate interrupts or DMA requests as system requirements.

4.17.1 Main Features

- A 16 bit timer/counter with a programmable 16 bit prescaler
- Counter or timer operation
- Two 16 bit capture channels per timer
- Synchronization circuit to control the timer with external signals or other timers
- Interrupt/DMA generation on the following events:
  - Input capture
  - Output match
- Four 16 bit match registers that allow:
  - Continuous operation with optional interrupt generation on match
  - Stop timer on match with optional interrupt generation
  - Reset timer on match with optional interrupt generation
- Two external outputs corresponding to match registers, with the following capabilities:
  - Set low on match
  - Set high on match
  - Toggle on match
- The combination of paired match registers generates PWM output

4.17.2 Block Diagram

TIMER

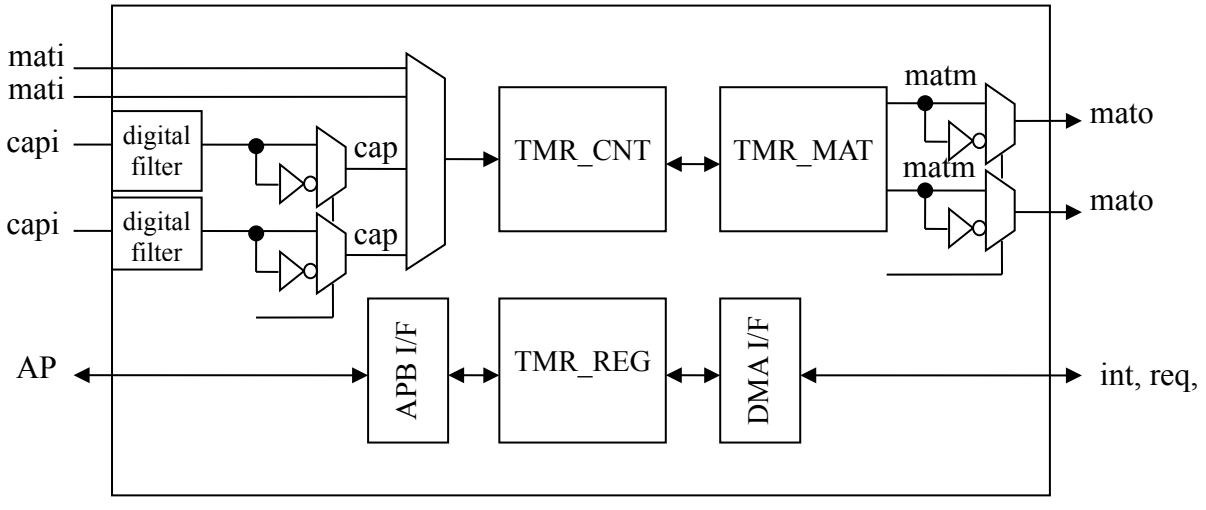


Figure 23: Timer Block Diagram

4.17.3 Register Definition

Timer0: address range 0x0020_1000 ~ 0x0020_13FF
Timer1: address range 0x0020_9000 ~ 0x0020_93FF
Timer2: address range 0x0020_1400 ~ 0x0020_17FF
Timer3: address range 0x0020_9400 ~ 0x0020_97FF
Timer4: address range 0x0020_1800 ~ 0x0020_1BFF
Timer5: address range 0x0020_9800 ~ 0x0020_9BFF
Timer6: address range 0x0020_1C00 ~ 0x0020_1FFF
Timer7: address range 0x0020_9C00 ~ 0x0020_9FFF
Timer8: address range 0x0020_2000 ~ 0x0020_23FF
Timer9: address range 0x0020_A000 ~ 0x0020_A3FF
TimerA: address range 0x0020_2400 ~ 0x0020_27FF
TimerB: address range 0x0020_A400 ~ 0x0020_A7FF

Index	Default	R/W	Bit	Name	Description
00H	0	R/W	15	tctl_en	timer enable
	-	-	14:9	-	reserved
	0	R/W	8	tctl_matco	mat0/mat1 control off 1: write mctl doesn't affect mat1/mat0 value 0: write mctl changes mat1/mat0 value accordingly
	0	R/W	7	tctl_de	debug enable 1: timer operates normally during debug mode 0: timer hold during debug mode
	0	R/W	6	tctl_iosw	I/O swap 1: swap capix/matox I/O definition and direction 0: no operation
	0	R/W	5:4	tctl_sel	counter input select 00: cap0 01: cap1 10: mati0 11: mati1
	0	R/W	3:2	tctl_mode	counter/timer mode 00: timer 01: rising edge counter 10: falling edge counter 11: both edge counter
	0	R/W	1	tctl_rst	counter reset
	0	R/W	0	tctl_st	timer start/stop control
04H	0	R/W	15:0	tcnt	timer counter, incremented every (pscl + 1) of trigger source
08H	0	R/W	15:0	pscl	prescale setting
0CH	0	R/W	15:0	pcnt	prescale counter
10H	-	-	15:12	-	reserved
	0	R/W	11:10	cctl_capdf	digital filter setting for capi0 & capi1 00: no operation 01: 2 clock 10: 4 clock 11: 8 clock
	0	R/W	9	cctl_c1sw	invert capi1 for cap1
	0	R/W	8	cctl_c0sw	invert capi0 for cap0

Index	Default	R/W	Bit	Name	Description
	-	-	7:4	-	reserved
	0	R/W	3	cctl_cap1f	capture enable on cap1 falling edge (stored in cap1f)
	0	R/W	2	cctl_cap1r	capture enable on cap1 rising edge (stored in cap1r)
	0	R/W	1	cctl_cap0f	capture enable on cap0 falling edge (stored in cap0f)
	0	R/W	0	cctl_cap0r	capture enable on cap0 rising edge (stored in cap0r)
14H	-	-	15:12	-	reserved
	0	R/W	11:10	cictl_c1actx	cap1 extended operation on counter 00: no operation 01: hold counter on cap1 low level 10: reset counter on cap1 falling edge 11: reserved
	0	R/W	9:8	cictl_c0actx	cap0 extended operation on counter
	0	R/W	7:6	cictl_m1act	mati1 operation on counter 00: no operation 01: hold counter on mati1 high level 10: reset counter on mati1 rising edge 11: trigger counter on mati1 rising edge (set tctl_st)
	0	R/W	5:4	cictl_m0act	mati0 operation on counter
	0	R/W	3:2	cictl_c1act	cap1 operation on counter 00: no operation 01: hold counter on cap1 high level 10: reset counter on cap1 rising edge 11: trigger counter on cap1 rising edge (set tctl_st)
	0	R/W	1:0	cictl_c0act	cap0 operation on counter
18H	0	R/W	15:14	mctl_m1mk	matm1 output mask 00: same as mat1 01: reserved 10: clear matm1 if cap0 = 1 11: clear matm1 if cap1 = 1
	0	R/W	13:12	mctl_m0mk	matm0 output mask
	0	R/W	11	mctl_mat1	mat1 value
	0	R/W	10	mctl_mat0	mat0 value
	0	R/W	9	mctl_m1sw	invert mat1 for mato1
	0	R/W	8	mctl_m0sw	invert mat0 for mato0
	0	R/W	7:6	mctl_mat1b	match control for mat1b 00: no operation 01: clear the corresponding external match bit to 0 10: set the corresponding external match bit to 1 11: toggle the corresponding external match bit
	0	R/W	5:4	mctl_mat1a	match control for mat1a
	0	R/W	3:2	mctl_mat0b	match control for mat0b
	0	R/W	1:0	mctl_mat0a	match control for mat0a
1CH	-	-	15:8	-	reserved
	0	R/W	7	moctl_m1bs	mat1b stop counter (clear tctl_st)
	0	R/W	6	moctl_m1br	mat1b reset counter
	0	R/W	5	moctl_m1as	mat1a stop counter
	0	R/W	4	moctl_m1ar	mat1a reset counter
	0	R/W	3	moctl_m0bs	mat0b stop counter
	0	R/W	2	moctl_m0br	mat0b reset counter
	0	R/W	1	moctl_m0as	mat0a stop counter
	0	R/W	0	moctl_m0ar	mat0a reset counter
30H	0	R/W	15:0	mat0a (cap0r)	match mat0a register for output match mode capture cap0r register for input capture mode

Index	Default	R/W	Bit	Name	Description
34H	0	R/W	15:0	mat0b (cap0f)	match mat0b register
38H	0	R/W	15:0	mat1a (cap1r)	match mat1a register
3CH	0	R/W	15:0	mat1b (cap1f)	match mat1b register
40H	0	R/W	15	of_tcnt	overflow flag for tcnt (tcnt loop from ffffh to 0000h)
	-	-	14:12	-	reserved
	0	R/W	11	of_cap1f	overflow flag for cap1f (twice or more interrupts occur)
	0	R/W	10	of_cap1r	overflow flag for cap1r
	0	R/W	9	of_cap0f	overflow flag for cap0f
	0	R/W	8	of_cap0r	overflow flag for cap0r
	0	R/W	7	if_mat1b	interrupt flag for mat1b (note-1)
	0	R/W	6	if_mat1a	interrupt flag for mat1a
	0	R/W	5	if_mat0b	interrupt flag for mat0b
	0	R/W	4	if_mat0a	interrupt flag for mat0a
	0	R/W	3	if_cap1f	interrupt flag for cap1f
	0	R/W	2	if_cap1r	interrupt flag for cap1r
	0	R/W	1	if_cap0f	interrupt flag for cap0f
	0	R/W	0	if_cap0r	interrupt flag for cap0r
44H	-	-	15:8	-	reserved
	0	R/W	7	ie_mat1b	interrupt enable for mat1b
	0	R/W	6	ie_mat1a	interrupt enable for mat1a
	0	R/W	5	ie_mat0b	interrupt enable for mat0b
	0	R/W	4	ie_mat0a	interrupt enable for mat0a
	0	R/W	3	ie_cap1f	interrupt enable for cap1f
	0	R/W	2	ie_cap1r	interrupt enable for cap1r
	0	R/W	1	ie_cap0f	interrupt enable for cap0f
	0	R/W	0	ie_cap0r	interrupt enable for cap0r
48H	-	-	15:8	-	reserved
	0	R/W	7	rf_mat1b	DMA request flag for mat1b (note-2)
	0	R/W	6	rf_mat1a	DMA request flag for mat1a
	0	R/W	5	rf_mat0b	DMA request flag for mat0b
	0	R/W	4	rf_mat0a	DMA request flag for mat0a
	0	R/W	3	rf_cap1f	DMA request flag for cap1f
	0	R/W	2	rf_cap1r	DMA request flag for cap1r
	0	R/W	1	rf_cap0f	DMA request flag for cap0f
	0	R/W	0	rf_cap0r	DMA request flag for cap0r
4CH	-	-	15:8	-	reserved
	0	R/W	7	re_mat1b	DMA request enable for mat1b
	0	R/W	6	re_mat1a	DMA request enable for mat1a
	0	R/W	5	re_mat0b	DMA request enable for mat0b
	0	R/W	4	re_mat0a	DMA request enable for mat0a
	0	R/W	3	re_cap1f	DMA request enable for cap1f
	0	R/W	2	re_cap1r	DMA request enable for cap1r
	0	R/W	1	re_cap0f	DMA request enable for cap0f
	0	R/W	0	re_cap0r	DMA request enable for cap0r

note-1:
all if_xxx need to write '1' to clear
all if_xxx/of_xxx are cleared if tctl_en=0
note-2:
all rf_xxx need to write '1' to clear
all rf_xxx are cleared if tctl_en=0

4.18 Multi-master I2C Interface

4.18.1 Introduction

I2C (inter-integrated circuit) bus Interface serves as an interface between the microcontroller and the serial I2C bus. It provides multi-master capability, and controls all I2C bus-specific sequencing, protocol, arbitration and timing. It supports standard and fast speed modes.

4.18.2 Features

- Multi-master capability: the same interface can act as Master or Slave
- I2C Master features:
  - Clock generation
  - Start and Stop generation
- I2C Slave features:
  - Programmable I2C Address detection
  - Dual Addressing Capability to acknowledge 2 slave addresses
  - Stop bit detection
- Generation and detection of 7-bit/10-bit addressing and General Call
- Supports different communication speed:
  - Standard Speed Mode (up to 100 kHz)
  - Fast Mode (up to 400 kHz)
  - High Mode Plus (up to 1 MHz)
- Status flags:
  - Transmitter/Receiver mode flag
  - End-of-Byte transmission flag
  - I2C busy flag
  - Master receiver return ACK(NACK) flag.
- Error flags:
  - Arbitration lost condition for master mode
  - Time out if SCL remained LOW for 25 ms (Timeout) or master cumulative clock low extend time more than 10 ms in SMbus mode.
  - Detection of misplaced start or stop condition
  - Overrun/Underrun if clock stretching is disabled
- 1 Interrupt vector with various sources:
  - 1 Interrupt for address/ data communication
  - 1 Interrupt for error/alert condition
- Optional Clock Stretching
- 1-byte buffer with DMA capability

4.18.3 Block Diagram

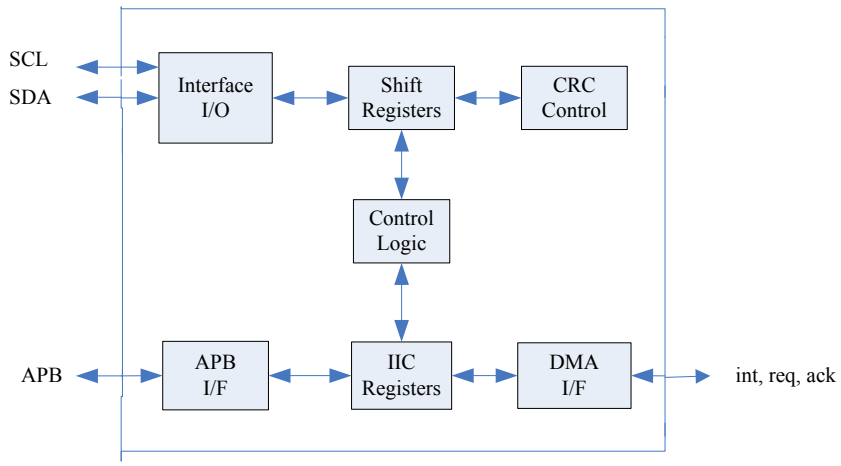


Figure 24: I2C Block Diagram

4.18.4 Functional Description

In addition to receiving and transmitting data, this interface converts it from serial to parallel Format and vice versa. The interrupts are enabled or disabled by software. The interface is connected to the I2C bus by a data pin (SDA) and by a clock pin (SCL). It can be connected with a standard (up to 100 kHz), Fast Mode (up to 400kHz).

Mode Selection

The interface can operate in one of the following modes:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

Communication Flow

In Master mode, the I2C interface initiates a data transfer and generates the clock signal. A serial data transfer always begins with a START condition and ends with a STOP condition. Both START and STOP conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognizing its own addresses (7 or 10-bit), and the General Call address. The General Call address detection may be enabled or disabled by software. The reserved SMBus addresses can also be enabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the start condition contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to the following figure.

Acknowledge may be enabled or disabled in master mode by software. The master RX return NACK in last transfer. The I2C interface addresses (dual addressing 7-bit/ 10-bit and/or general call address) can be selected by software.

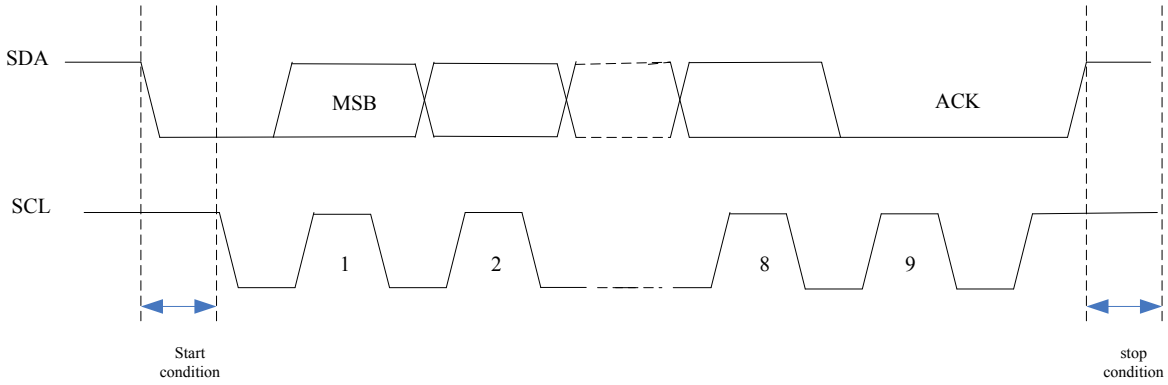


Figure 25: I2C Bus Protocol

I2C initialization

Noise filter

Before enabling the I2C peripheral, you must configure the digital noise filter, if needed. When the digital filter is enabled, the level of the SCL or the SDA line is internally changed only if it remains stable for more than the programming length set by the FILTER in I2C_TIMINGR1 register.

Timing

The timings must be configured in order to guarantee a correct data hold and setup time, used in master and slave modes. This is done by programming the SCLDEL[7:0] and SDADEL[7:0] bits in the I2C_TIMINGR1 register.

In addition, in master mode, the SCL clock high and low levels must be configured by programming the SCLH[11:0] and SCLL[11:0] bits in the I2C_TIMINGR1 register.

Software reset

A software reset can be performed by setting the SWRST bit in the I2C_CR2 register.
Internal states machines are reset and communication control bits, as well as status bits come back to their reset value. The configuration registers are not impacted.

Data transfer

The data transfer is managed through transmit and receive data registers and a shift register.

Reception

The SDA input fills the shift register. When the complete data byte is received, the shift register is copied into I2C_RXDR register if it is empty (RXNE=0). If RXNE=1, meaning that the previous received data byte has not yet been read, the SCL line is stretched low until I2C_RXDR is read if the clock stretching is enabled (NOSTRETCH=0). The stretch is inserted between the 8th and 9th SCL pulse (before the Acknowledge pulse).

Transmission

If the I2C_TXDR register is not empty (TXE=0), its content is copied into the shift register after the 9th SCL pulse (the Acknowledge pulse). Then the shift register content is shifted out on SDA line. If TXE=1, meaning that no data is written yet in I2C_TXDR, SCL line is stretched low until I2C_TXDR is written. The stretch is done after the 9th SCL pulse.

Hardware transfer management

The I2C has a byte counter embedded in hardware in order to manage byte transfer and to close the communication in various modes such as:
– NACK, STOP and ReSTART generation in master mode
– ACK control in slave receiver mode
– PEC generation/checking when SMBus feature is supported
The byte counter is always used in master mode. By default it is disabled in slave mode, but it can be enabled by software by setting the SBC (Slave Byte Control) bit in the I2C_CR1 register. The number of bytes to be transferred is programmed in the NBYTES[7:0] bit field in the I2C_BCNTR register. If the number of bytes to be transferred (NBYTES) is greater than 255, or if a receiver wants to control the acknowledge value of a received data byte, the reload mode must be selected by setting the RELOAD bit in the I2C_BCNTR register. In this mode, TCR flag in the I2C_ISCR is set when the number of bytes programmed in NBYTES has been transferred, and an interrupt is generated if TCIE is set. SCL is stretched as long as TCR flag is set. TCR is cleared by software when NBYTES is written to a non-zero value. When the NBYTES counter is reloaded with the last number of bytes, RELOAD bit must be cleared.

When RELOAD=0 in master mode, the counter can be used in 2 modes:
– **Automatic end mode** (AUTOEND = '1' in the I2C_BCNTR register). In this mode, the master automatically sends a STOP condition once the number of bytes programmed in the NBYTES[7:0] bit field has been transferred.
– **Software end mode** (AUTOEND = '0' in the I2C_ BCNTR register). In this mode, software action is expected once the number of bytes programmed in the NBYTES[7:0] bit field has been transferred; the TC flag is set and an interrupt is generated if the TCIE bit is set. The SCL signal is stretched as long as the TC flag is set. The TC flag is cleared by software when the START or STOP bit is set in the I2Cx_CR2 register. This mode must be used when the master wants to send a RESTART condition.

I2C Slave Mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register. Then it is compared with the address of register I2C_OAR1 (bit1~bit10 for 10 bit mode or bit1~7 for 7 bit mode) and with the address of register I2C_OAR2 (bit1~bit7,if OA2EN bit=1) or the General Call address (if register I2C_CR1 bit0 (GCEN) = 1). Up to 7 OA2 LSB can be masked by configuring the OA2MSK[2:0] bits in the T2C_OAR2 register. As soon as OA2MSK is not equal to 0, the address comparator for OA2 excludes the I2C reserved addresses (0000 XXX and 1111 XXX), which are not acknowledged. If OA2MSK=7, all received 7-bit addresses are acknowledged (except reserved addresses). OA2 is always a 7-bit address.

Note: In 10-bit addressing mode, the comparison includes the header sequence (11110xx0), where xx denotes the two most significant bits of the address.

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the interface generates in sequence:
- an interrupt is generated if the ADDRIE in the I2C_CR1 register is set.
- if several addresses are enabled you must read the ADDCODE[6:0] bits in the I2C_SR2 register in order to check which address matched. DIR flag must also be checked in order to know the transfer direction.

In 10-bit mode, after receiving the address sequence the slave is always in Receiver mode. It will enter Transmitter mode on receiving a repeated Start condition followed by the header sequence with matching address bits and the least significant bit set (11110xx1).

DIR in I2C_SR2 indicates whether the slave is in Receiver or Transmitter mode.

Slave Byte Control Mode:
In order to allow byte ACK control in slave reception mode, Slave Byte Control mode must be enabled by setting the SBC bit in the I2C_CR1 register. This is required to be compliant with SMBus standards.
Reload mode must be selected in order to allow byte ACK control in slave reception mode (RELOAD=1). To get control of each byte, NBYTES must be initialized to 0x1 in the ADDR interrupt subroutine, and reloaded to 0x1 after each received byte. When the byte is received, the TCR bit in is set, stretching the SCL signal low between the 8th and 9th SCL pulses. You can read the data from the I2C_RXDR register, and then decide to acknowledge it or not by configuring the NACK bit in the I2C_CR2 register. The SCL stretch is released by programming NBYTES to a non-zero value: the acknowledge or not acknowledge is sent and next byte can be received. NBYTES can be loaded with a value greater than 0x1, and in this case, the reception flow is continuous during NBYTES data reception.

Note 1: The SBC bit must be configured when the I2C is disabled, or when the slave is not addressed, or when ADDR=1. The RELOAD bit value can be changed when ADDR=1, or when TCR=1.

Note 2: Slave Byte Control mode is not compatible with NOSTRETCH mode. Setting SBC when NOSTRETCH=1 is not allowed.

Slave transmitter
A transmit interrupt status (TXIS) is generated when the I2C_TXDR register becomes empty. An interrupt is generated if the TXIE bit is set in the I2C_CR1 register.
The TXIS bit is cleared when the I2C_TXDR register is written with the next data byte to be transmitted.
When a NACK is received, the NACKF bit is set in the I2C_ISCR register and an interrupt is generated if the NACKIE bit in the I2C_CR1 register is set. The slave automatically releases the SCL and SDA lines in order to let the master perform a STOP or a RESTART condition. The TXIS bit is not set when a NACK is received.
When a STOP is received and the STOPIE bit is set in the I2C_CR1 register, the STOPF flag is set in the I2C_ISCR register and an interrupt is generated. In most applications, the SBC bit is usually programmed to '0'. In this case, If TXE = 0 when the slave address is received (ADDR=1), you can choose either to send the content of the I2C_TXDR register as the first data byte, or to flush the I2C_TXDR register by setting the TXE bit in order to program a new data byte.
In Slave Byte Control mode (SBC=1), the number of bytes to be transmitted must be programmed in NBYTES in the address match interrupt subroutine (ADDR=1). In this case, the number of TXIS events during the transfer corresponds to the value programmed in NBYTES.

Slave receiver
RXNE is set in I2C_ISCR when the I2C_RXDR is full, and generates an interrupt if RXIE in I2C_CR1 is set.
RXNE is cleared when I2C_RXDR is read.
When a STOP is received and STOPIE is set in I2C_CR1, STOPF is set in I2C_ISCR and an interrupt is generated.

I2C Master Mode

Before enabling the I2C, software must configure the SCCH and SCLL in the I2C_TIMINGR1 register to set the master clock frequency.
In order to support multi-master environment and slave clock stretching, a clock synchronization mechanism is implemented:

- The low level of the clock is counted using the SCLL counter, starting from the SCL low level internal detection.
- The high level of the clock is counted using the SCLH counter, starting from the SCL high level internal detection.

Communication Initialization

To initiate the communication, following parameters must be programmed for the addressed slave in the I2C_SADDR register:

- Addressing mode (7-bit or 10-bit): ADD10
- Slave address to be sent: SADD[9:0]
- Transfer direction: RD_WRN
- The number of bytes to be transferred: NBYTES[7:0]. If the number of bytes is equal to or greater than 255 bytes, NBYTES[7:0] must initially be filled with 0xFF.

You must then set the START bit in I2C_CR2 register. Changing all the above bits is not allowed when START bit is set.
The master will automatically send the START condition followed by the slave address as soon as it detects that the bus is free (BUSY = 0).
In case of an arbitration loss, the master automatically switches back to slave mode and can acknowledge its own address if it is addressed as a slave.

Master Transmitter

For a write transfer, the TXIS flag is set after each byte transmission, after the 9th SCL pulse when an ACK is received. An interrupt will be generated if the TXIE bit in the I2C_CR1 is set. Written data to the I2C_TXDR will clear the flag.
The number of TXIS events during the transfer corresponds to the value programmed in NBYTES[7:0].
If the total number of data bytes to be sent is greater than 255, reload mode must be selected by setting the RELOAD bit in the I2C_BCNTR register. In this case, when NBYTES data have been transferred, the TCR flag in register I2C_ISCR is set and the SCL line is stretched low until NBYTES[7:0] is written to a non-zero value.

The TXIS flag is not set when a NACK is received. A STOP condition is automatically sent after the NACK reception. the NACKF flag is set in the I2C_ISCR register, and an interrupt is generated if the NACKIE bit is set.

Master Receiver

For a read transfer, the RXNE flag is set after each byte reception, after the 8th SCL pulse. An RXNE event generates an interrupt if the RXIE bit is set in the I2C_CR1 register. The flag is cleared when I2C_RXDR is read.
If the total number of data bytes to be received is greater than 255, reload mode must be selected by setting the RELOAD bit in the I2C_BCNTR register. In this case, when NBYTES[7:0] data have been transferred, the TCR flag is set and the SCL line is stretched low until NBYTES[7:0] is written to a non-zero value.

- When RELOAD=0 and NBYTES[7:0] data have been transferred:
  - In automatic end mode (AUTOEND=1), a NACK and a STOP are automatically sent after the last received byte.
  - In software end mode (AUTOEND=0), a NACK is automatically sent after the last received byte, the TC flag is set and the SCL line is stretched low in order to allow software actions

  A RESTART condition can be requested by setting the START bit in the I2C_CR2 register with the proper slave address configuration, and number of bytes to be transferred. Setting the START bit clears the TC flag and the START condition, followed by slave address, are sent on the bus.

  A STOP condition can be requested by setting the STOP bit in the I2C_CR2 register. Setting the STOP bit clears the TC flag and the STOP condition is sent on the bus.

Error conditions

Bus error

A bus error is detected when a START or a STOP condition is detected at the wrong location. The bus error flag BERR will be set when bus error occurs.

Arbitration lost
An arbitration loss is detected when the sampled SDA does not match the sent SDA. The arbitration loss flag ARLO will be set and an interrupt will be generated if the ERRIE bit in the I2C_CR1 is set.

- In master mode, arbitration loss is detected during the address phase, data phase and data acknowledge phase. In this case, the SDA and SCL lines are released, the START control bit is cleared by hardware and the master switches automatically to slave mode.
- In slave mode, arbitration loss is detected during data phase and data acknowledge phase. In this case, the transfer is stopped, and the SCL and SDA lines are released.

Overrun/underrun
An overrun or underrun error is detected in slave mode when NOSTRETCH=1 and:

- When a new byte is received and the RXDR register has not been read yet. The new received byte is lost, and a NACK is automatically sent.
- When a new byte should be sent and the I2C_TXDR register has not been written yet, 0xFF is sent.

The OVR flag in the I2C_ISCR register will be set, and an interrupt will be generated if the ERRIE bit in the I2C_CR1 register is set.

Packet Error Check
A PEC error is detected when the received PEC byte does not match with the I2C_PECR register content.
A NACK is automatically sent after the wrong PEC reception.
The PECERR flag in the I2C_ISCR register will be set and an interrupt will be generated if the ERRIE bit in the I2C_CR1 register is set.

Timeout
A timeout error occurs when the SCL remains low or cumulative SCL extend low time exceed the reached time defined in the TIMEOUTA or TIMEOUTB, or when both SCL and SDA remained high for the time defined in the TIMEOUTA when TIDLE=1. The TIMEOUT flag in the I2C_ISCR will be set and an interrupt will be generated if the ERRIE bit in the I2C_CR1 register is set.

When a timeout violation is detected in master mode, a STOP condition is automatically sent.
When a timeout violation is detected in slave mode, SDA and SCL lines are automatically released.

Alert
The ALERT flag will be set when a falling edge is detected on the SMBA pin and the SMBHEN bit and ALERTEN bit are set. An interrupt will be generated if the ERRIE bit in the I2C_CR1 register is set.

DMA requests

Transmission using DMA

DMA mode can be enabled for transmission by setting the TXDMAEN bit in the I2C_CR1 register.
Data will be loaded from a Memory area configured using the DMA to the I2C_TXDR register whenever the TXIS bit is set.

Only the data is transferred with DMA.

- In master mode: the initialization, the slave address, direction, number of bytes and START bit are programmed by software. When all data are transferred using DMA, the DMA must be initialized before setting the START bit. The end of transfer is managed with the NBYTES counter.
- In slave mode:
  – With NOSTRETCH=0, when all data are transferred using DMA, the DMA must be initialized before the address match event, or in ADDR interrupt subroutine, before clearing ADDR.
  – With NOSTRETCH=1, the DMA must be initialized before the address match event.
- For SMBus: the PEC transfer is managed with NBYTES counter.

Reception using DMA

DMA mode can be enabled for reception by setting the RXDMAEN bit in the I2C_CR1 register. Data will be loaded from the I2C_RXDR register to a Memory area configured using the DMA whenever the RXNE bit is set. Only the data(including PEC) are transferred with DMA.

- In master mode, the initialization, the slave address, direction, number of bytes and START bit are programmed by software. When all data are transferred using DMA, the DMA must be initialized before setting the START bit. The end of transfer is managed with the NBYTES counter.
- In slave mode with NOSTRETCH=0, when all data are transferred using DMA, the DMA must be initialized before the address match event, or in the ADDR interrupt subroutine, before clearing the ADDR flag.

4.18.5 Register Definition

I2C0: address range 0x0020_4000 ~ 0x0020_43FF

Control Register 1 (I2C_CR1)

Index	Default	R/W	Bit	Name	Description
00h			31:23		Reserved
	0	R/W	22	MNACKOPT	Master mode received NACK option 0: auto stop. 1: continue to TX.
	0	R/W	21	SNACKOPT	Slave mode transmitted data-NACK option 0: stop RX. 1: continue to RX.
	0	R/W	20	NOSARB	Slave mode arbitration disable 0: with arbitration. 1: no arbitration.
			19:13		Reserved
	0	R/W	12	GCEN	General call address (00000000) enable 0: disabled. 1: enabled.
	0	R/W	11	NOSTRETCH	Clock stretching disable (in slave mode) 0: Clock stretching enabled. 1: Clock stretching disabled.
	0	R/W	10	SBC	Slave bye control 0: disabled. 1: enabled.
	0	R/W	9	RXDMAEN	DMA reception requests enable 0: disabled. 1: enabled.
	0	R/W	8	TXDMAEN	DMA transmission requests enable 0: disabled. 1: enabled.
	0	R/W	7	ERRIE	Error interrupts enable 0: disabled. 1: enabled. Note: Any of these errors will generate an interrupt: ARLO, BERR, OVR, TIMEOUT, PECERR, ALERT
	0	R/W	6	TCIE	Transfer Complete interrupt enable 0: disabled. 1: enabled. Note: Any of these events will generate an interrupt: TC, TCR
	0	R/W	5	STOPIE	STOP detection interrupt enable 0: disabled. 1: enabled.
	0	R/W	4	NACKIE	Not acknowledge received interrupt enable 0: disabled. 1: enabled.
	0	R/W	3	ADDRIE	Address match interrupt enable 0: disabled. 1: enabled.
	0	R/W	2	RXIE	RX interrupt (RXNE) enable 0: disabled. 1: enabled.

Index	Default	R/W	Bit	Name	Description
	0	R/W	1	TXIE	TX interrupt (TXIS) enable 0: disabled. 1: enabled.
	0	R/W	0	I2CEN	I2C peripheral enable 0: disabled. 1: enabled.

Note: when SMBUS is set, the related IIC pins will be configured as TTL level.

Own Address 1 Register (I2C_OAR1)

Index	Default	R/W	Bit	Name	Description
04h			31:12		Reserved
	0	R/W	11	ADDMODE	1: 10-bit address mode 0: 7-bit address mode
	0h	R/W	10:1	OA1	[10:1] for 10-bit address [7:1] for 7-bit address
	0	R/W	0	OA1EN	OA1 enable/disable

Own Address 2 Register (I2C_OAR2)

Index	Default	R/W	Bit	Name	Description
08h			31:11		Reserved
	0h	R/W	10:8	OA2MSK	OA2 masks 000: no mask, all OA2[7:1] are compared 001: only OA2[7:2] are compared 010: only OA2[7:3] are compared 011: only OA2[7:4] are compared 100: only OA2[7:5] are compared 101: only OA2[7:6] are compared 110: only OA2[7] is compared 111: all (except reserved) 7-bit received address are acknowledged.
	0h	R/W	7:1	OA2	[7:1] of 7-bit address
	0	R/W	0	OA2EN	OA2 enable/disable

Transmit Data Register (I2C_TXDR)

Index	Default	R/W	Bit	Name	Description
0Ch			31:8		Reserved
	00h	R/W	7:0	TXDATA	8-bit transmit data

Receive Data Register (I2C_RXDR)

Index	Default	R/W	Bit	Name	Description
10h			31:8		Reserved
		R	7:0	RXDATA	8-bit receive data

Slave Address Register (I2C_SADDR)

Index	Default	R/W	Bit	Name	Description
14h			31:12		Reserved
	0	R/W	11	ADD10	1: 10-bit address mode 0: 7-bit address mode
	0h	R/W	10:1	SADD	[10:1] for 10-bit address [7:1] for 7-bit address
	0	R/W	0	RD_WRN	0: for write 1: for read

Byte Count Register (I2C_BCNTR)

Index	Default	R/W	Bit	Name	Description
18h			31:10		Reserved
	0	R/W	9	AUTOEND	Automatic end mode (master mode) 0: software end mode: TC flag is set when NBYTES data are transferred, stretching SCL low. 1: automatic end mode: a STOP condition is automatically sent when NBYTES data are transferred. Note: this bit has no effect in slave mode or when the RELOAD bit is set.
	0	R/W	8	RELOAD	NBYTES reload mode 0: the transfer is completed after the NBYTES data transfer (STOP or RESTART will fellow). 1: the transfer is not completed after the NBYTES data transfer (NBYTES will be reloaded). TCR flag wis set when NBYTES data are transferred, stretching SCL low.
	0h	R/W	7:0	NBYTES	Number of bytes The number of bytes to be transmitted/received is programmed there. This field is don't care in slave mode with SBC=0.

Control Register 2 (I2C_CR2)

Index	Default	R/W	Bit	Name	Description
1Ch			31:4		Reserved
	0	R/W	3	SWRST	Software Reset the I2C module
	0	R/W	2	NACK	NACK generation (slave mode) This bit is set by software, clear by hardware when the NACK is sent, or when a STOP condition or an Address Matched is received. 0: an ACK is sent after current received byte. 1: an NACK is sent after current received byte. Note: Writing '0' to this bit has no effect. ...
	0	R/W	1	STOP	Stop generation (master mode) This bit is set by software, clear by hardware when a Stop condition is detected. 0: No Stop generation. 1: Stop generation after current byte transfer.
	0	R/W	0	START	Start generation This bit is set by software, clear by hardware after the Start is sent. 0: No Start generation. 1: Start/Restart generation.

Interrupt Status/Clear Register (I2C_ISCR)

Index	Default	R/W	Bit	Name	Description
20h			31:16		Reserved
	0	R	15	BUSY	Bus Busy This flag indicates that a communication is in progress on the bus. It is set by hardware when a START condition is detected. It is cleared by hardware when a Stop condition is detected.
			14:13		Reserved
	0	R/W1C	12	TIMEOUT	Timeout or t_{LOW} detection flag This flag is set by hardware when a timeout or extended clock timeout occurred. When timeout occurs, SDA/SCL will be released. User has to write 1 to this bit to clear the timeout flag.
			11		Reserved
	0	R/W1C	10	OVR	Overrun/Underrun (slave mode) 1: Overrun or underrun 0: No overrun/underrun
	0	R/W1C	9	ARLO	Arbitration lost 1: Arbitration Lost detected 0: No Arbitration Lost detected
	0	R/W1C	8	BERR	Bus error 1: Misplaced start or stop condition 0: No misplaced start or stop condition
	0	R	7	TCR	Transfer Complete Reload This flag is set by hardware when RELOAD=1 and NBYTES data have been transferred. It is cleared by software when NBYTES is written to a non-zero value.
	0	R	6	TC	Transfer Complete (master mode) This flag is set by hardware when RELOAD=0, AUTOEND=0 and NBYTES data have been transferred. It is cleared by software when START bit or STOP bit is set.
	0	R/W1C	5	STOPF	Stop condition flag This flag is set by hardware when a Stop condition is detected on the bus and the peripheral is involved in this transfer: - either as a master, provided that the STOP condition is generated by the peripheral. - or as a slave, provided that the peripheral has been addressed previously during this transfer.
	0	R/W1C	4	NACKF	Not Acknowledge received flag This flag is set by hardware when a NACK is received after a byte transmission.
	0	R/W1C	3	ADDR	Address matched (slave mode) This bit is set by hardware as soon as the received slave address matched with one of the enabled slave addresses.
	0	R	2	RXNE	Receive data register not empty This bit is set by hardware when the received data is copied into the ICR_RXDR register, and is ready to be read. It's cleared when I2C_RXDR is read.
	0	R/W1S	1	TXIS	Transmit interrupt status This bit is set by hardware when the I2C_TXDR register is empty and the data to be transmitted must be written in the I2C_TXDR register. It is cleared when the next data to be sent is written in the I2C_TXDR register. This bit can be written to '1' by software when NOSTRETCH=1

Index	Default	R/W	Bit	Name	Description
					only, in order to generate a TXIS event (interrupt if TXIE=1 or DMA request if TXDMAEN=1).
	1	R/W1S	0	TXE	Transmit data register empty This bit is set by hardware when the I2C_TXDR register is empty. It is cleared when the next data to be sent is written in the I2C_TXDR register. This bit can be written to '1' by software in order to flush the transmit data register I2C_TXDR.
			31:16		Reserved

Status Register 2 (I2C_SR2)

Index	Default	R/W	Bit	Name	Description
24h			31:16		Reserved
	0	R	15	BUSY	Bus busy
			14		Reserved
		R	13	SDAP	Filtered SDA pin input status
		R	12	SCLP	Filtered SCL pin input status
	1	R	11	NOTACTIVE	The peripheral is not in active state
	0	R	10	SLVACTIVED	Slave address matched, auto cleared after STOP condition
	0	R	9	MASTER	Master is active
	0	R	8	MRW	Master R/W (Receive/Transmit) status
	0h	R	7:1	ADDCODE	Address match code (slave mode) These bit are updated with the received address when an address match event occurred (ADDR=1). In the case of a 10-bit address, ADDCODE provides the 10-bit header followed by the 2 MSBs of the address.
	0	R	0	DIR	Transfer direction (slave mode) 0: Write transfer, slave enters receiver mode. 1: Read transfer, slave enters transmitter mode.

PEC Register (I2C_PECR)

Index	Default	R/W	Bit	Name	Description
28h			31:8		Reserved
	0h	R	7:0	PEC	Packet error checking register This field contains the internal PEC.

Timing Register 1 (I2C_TIMINGR1)

Index	Default	R/W	Bit	Name	Description
2Ch			31:20		Reserved
	3h	R/W	19:16	FILTER	Digital noise filter for SCL/SDA inputs. The filter will filter spike with a length of up to FILTER[3:0]*t_{CLK}
	1Eh	R/W	15:8	SCLDEL	Data setup time t_{SCLDEL} = (SCLDEL [7:0]+1)*t_{CLK}
	14h	R/W	7:0	SDADEL	Data hold time t_{SCLDEL} = (SDADEL [7:0]+1)*t_{CLK}

Timing Register 2 (I2C_TIMINGR2)

Index	Default	R/W	Bit	Name	Description
30h			31:24		Reserved
	21h	R/W	23:12	SCLH	SCL high period (master mode) t_{SCLH} = (SCLH [11:0]+1)*t_{CLK}
	4Dh	R/W	11:0	SCLL	SCL low period (master mode) t_{SCLL} = (SCLL [11:0]+1)*t_{CLK}

Timeout Register 1 (I2C_TIMEOUTR1)

Index	Default	R/W	Bit	Name	Description
34h			31:16		Reserved
	0	R/W	15	TIMOUTEN	Clock timeout enable 0: SCL timeout detection is disabled. 1: SCL timeout detection is enabled.
			14:13		Reserved
	0	R/W	12	TIDLE	Idle clock timeout detection 0: TIMEOUTA is used to detect SCL low timeout 1: TIMEOUTA is used to detect bot SCL and SDA high timeout (bus idle condition)
	0h	R/W	11:0	TIMEOUTA	Bus Timeout A This field is used to configure: - The SCL low timeout condition $t_{TIMEOUT}$ when TIDLE=0 $t_{TIMEOUT}$ = (TIMEOUTA +1)*2048*t_{CLK} - The bus idle condition (both SCL and SDA high) when TIDLE=1 t_{IDLE} = (TIMEOUTA +1)*4*t_{CLK}

Timeout Register 2 (I2C_TIMEOUTR2)

Index	Default	R/W	Bit	Name	Description
38h			31:16		Reserved
	0	R/W	15	TEXTEN	Extended clock timeout enable 0: disabled. 1: enabled.
			14:12		Reserved
	0h	R/W	11:0	TIMEOUTB	Bus Timeout B $t_{LOW:EXT}$ = (TIMEOUTB +1)*2048*t_{CLK}

4.19 Master/Slave I2C Interface

4.19.1 Register Definition

I2C2: address range 0x0020_4400 ~ 0x0020_47FF
I2C3: address range 0x0020_C400 ~ 0x0020_C7FF
I2C4: address range 0x0020_4800 ~ 0x0020_4BFF
I2C5: address range 0x0020_C800 ~ 0x0020_CBFF
I2C6: address range 0x0020_4C00 ~ 0x0020_4FFF
I2C7: address range 0x0020_CC00 ~ 0x0020_CFFF

Control Register

Index	Default	R/W	Bit	Name	Description
00	0	R/W	7	MI2C_EN	I2C function enable (can used for reset I2C)
	1		6:5	MI2C_CLK	Select MI2C clock 00: SCL clock 400KHz 01: SCL clock 250KHz (default) 10: SCL clock 125KHz 11: SCL clock 62.5KHz
		W	4	MI2C_START	Set Master I2C START phase
		W	3	MI2C_STOP	Set Master I2C STOP phase
	0	R/W	2	MI2C_TXNAK	Master I2C transmit ACK bit after next Rx state 0: ACK 1: NACK
		W	1	MI2C_CLR_RT	Clear transmit/receive interrupt (both mode)
		W	0	MI2C_CLR_STP	Clear slave mode STOP interrupt

Status Register

Index	Default	R/W	Bit	Name	Description
04		R	7	MI2C_RDY	INT when I2C receive/transmit 9 bits or SLAVE STOP phase
		R	6	MI2C_INT_RT	INT when I2C receive/transmit 9 bits
		R	5	MI2C_INT_STOP	INT when I2C slave mode STOP phase
		R	4	MI2C_BB	Slave mode bus busy
		R	3	MI2C_FIRST	Master/Slave mode FIRST phase This is first byte from master I2C with specific slave address
		R	2	MI2C_RW	Slave mode READ/WRITE phase (8th bit of first byte) 0: Master/Slave I2C as receiver 1: Master/Slave I2C as transmitter
		R	1	MI2C_RXNAK	ACK bit indicator when I2C in slave Tx mode =1: Master mode return NACK (Slave will pull SAD high) =0: Master mode return ACK
		R	0	MI2C_DMAFAIL	DMA speed not enough

Note: Clear MI2C_DMAFAIL flag by MI2C_CLR_RT

Receive & Transmit Buffer Register

Index	Default	R/W	Bit	Name	Description
08	00	R/W	7:0	MI2C_DSLV	Master I2C transmit slave address buffer
0C	FF	R/W	7:0	MI2C_DTX	I2C Tx buffer
10		R	7:0	MI2C_DRX	I2C Rx buffer

Note: Separation Tx/Rx buffer for DMA address define use

Master/Salve I2C Slave Address Register

Index	Default	R/W	Bit	Name	Description
14	00	R/W	7:1	MI2C_SADR	I2C slave address
	0	R/W	0	MI2C_SLVE	Enable I2C slave mode

Master/Slave I2C Extend Control Register

Index	Default	R/W	Bit	Name	Description
18			7:3		Reserved
	0	R/W	2	MI2C_DMAEN	Enable master I2C DMA mode (master mode only)
	0	R/W	1	MI2C_AUTOSTP	Enable master I2C auto STOP when receive NACK bit
	0	R/W	0	MI2C_WAIT	Enable slave I2C pull SCL low after 9th bit

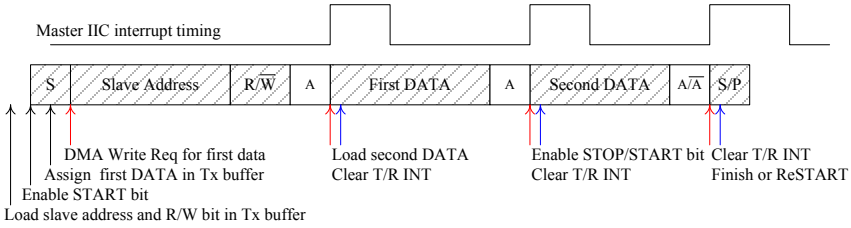
Note-1: Master/Salve I2C SCL & SDA input pin with digital filter

Note-2: When DMA enable, the master mode Tx/Rx data source from internal RAM and the DMA requests are generated only for data transfers

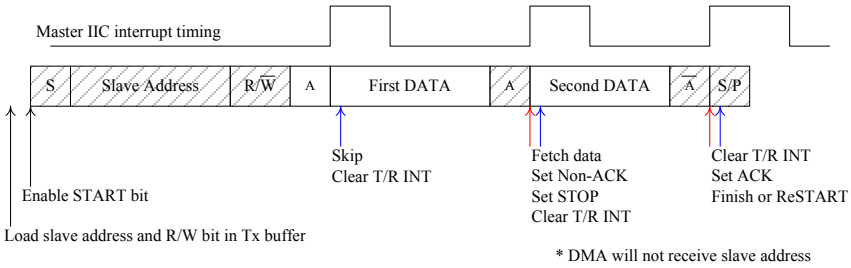
4.19.2 I2C Command Timing Flow

I²C Command Timing Flow

● Master-Transmitter mode

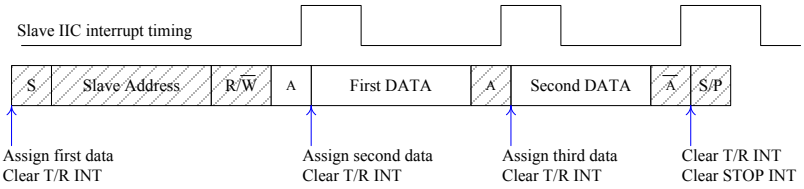


● Master-Receiver mode

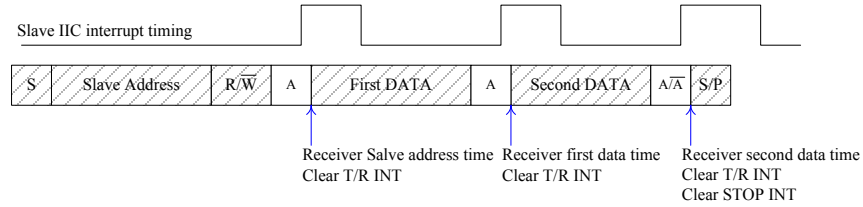


* DMA will not receive slave address

● Slave-Transmitter mode



● Slave-Receiver mode



: From Slave to Master

: From Master to Slave

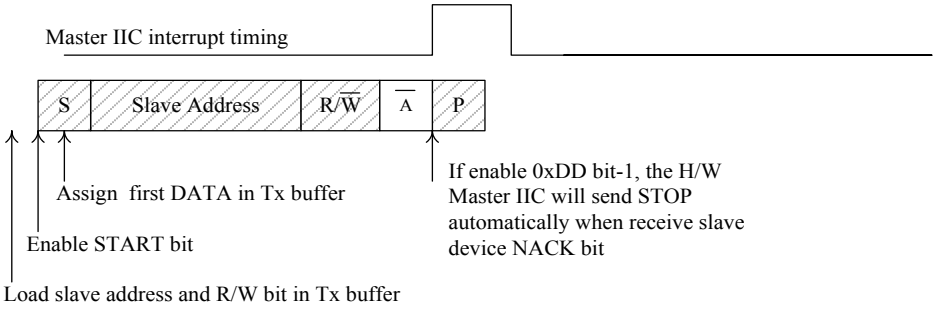
: Command Action during interrupt period

: DMA Request Time

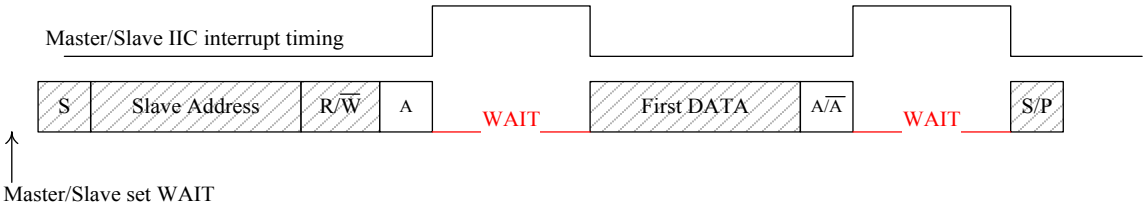
Command Limitation
Master Mode
1. Each command must-action in previous interrupt.
2. After set return Non-ACK bit, remember to recover it or it will return Non-ACK forever until set return ACK bit.
3. START and STOP bit will not be latched, so we dose not recover it.
Slave Mode
1. Each command must-action in previous interrupt (include WAIT command).
2. In Slave-Tx mode, the first byte will send initial Tx buffer data or you can load needed data as initial data before Slave I2C be selected.

Master I²C Command Timing Flow 2

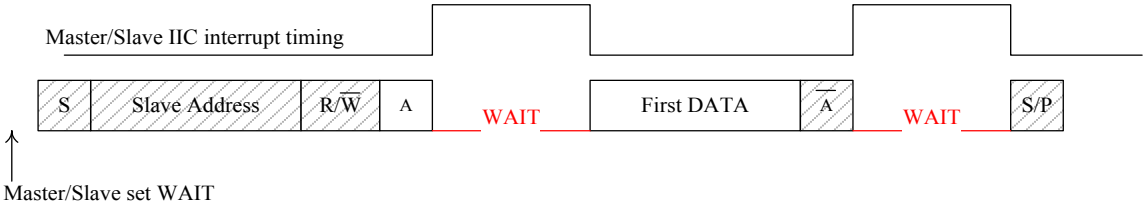
● Master I²C Auto-STOP



● Master Tx/Slave Rx Wait



● Master Rx/Slave Tx Wait



: From Slave to Master

: From Master to Slave

: Command Action during interrupt period

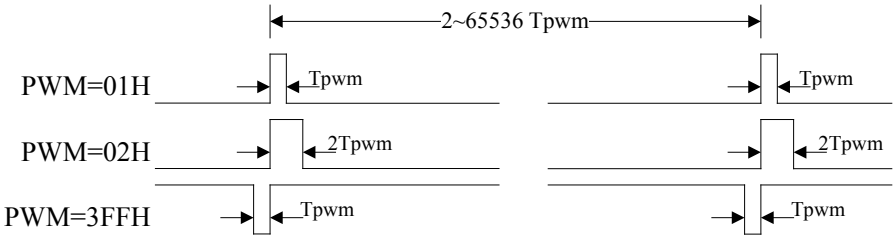
Command Limitation
Master Mode
1. Each command must-action in previous interrupt. **2. After set return Non-ACK bit, remember to recover it or it will return Non-ACK forever until set return ACK bit.** **3. START and STOP bit will not be latched, so we dose not recover it.**
Slave Mode
1. Each command must-action in previous interrupt (include WAIT command). **2. In Slave-Tx mode, the first byte will send initial Tx buffer data or you can load needed data as initial data before Slave I2C be selected.**

4.20 PWM

The PWM module supports a wide variety of PWM modes and is ideal for power conversion applications.

4.20.1 Functional Description

The PWM module generates 12 16-bit PWM output at frequency range:
The corresponding PWM register controls the PWM width. PWM width is up to 65535*Tpwm.



4.20.2 Register Definition

PWM register: address range 0x0020_8C00 ~ 0x0020_8CFF

Index	Default	R/W	Bit	Name	Description
00	0000 0000		31:12		Reserved
		R/W	11:0	PWM_EN[11:0]	1: Enable PWM[x] function 0: Disable PWM[x] function
04	0000 0000		31:2		Reserved
		R/W	1:0	PWM_BAS_CLK	PWM base clock select 00: system clock 01: symtem clock / 2 10: system clock / 3 11: system clock / 12
08	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK0[6:0]	Select PWM0 clock
0C	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK1[6:0]	Select PWM1 clock
10	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK2[6:0]	Select PWM2 clock
14	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK3[6:0]	Select PWM3 clock
18	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK4[6:0]	Select PWM4 clock
1C	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK5[6:0]	Select PWM5 clock
20	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK6[6:0]	Select PWM6 clock
24	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK7[6:0]	Select PWM7 clock
28	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK8[6:0]	Select PWM8 clock
2C	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK9[6:0]	Select PWM9 clock

Index	Default	R/W	Bit	Name	Description
30	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK10[6:0]	Select PWM10 clock
34	0000 0000		31:7		Reserved
		R/W	6:0	PWM_CLK11[6:0]	Select PWM11 clock
38	0000 2000		31:16		Reserved
		R/W	15:0	PWM0[15:0]	Select duty cycle of PWM 0 output 0: PWM width = 0 1: PWM width= Tpwm 2: PWM width= 2*Tpwm : FFFF: PWM width= 65535*Tpwm
3C	0000 2000		31:16		Reserved
		R/W	15:0	PWM1[15:0]	Select the width of PWM 1 output [15:0]:
40	0000 2000		31:16		Reserved
		R/W	15:0	PWM2[15:0]	Select the width of PWM 2 output [15:0]
44	0000 2000		31:16		Reserved
		R/W	15:0	PWM3[15:0]	Select the width of PWM 3 output [15:0]
48	0000 2000		31:16		
		R/W	15:0	PWM4[15:0]	Select the width of PWM 4 output [15:0]
4C	0000 2000		31:16		
		R/W	15:0	PWM5[15:0]	Select the width of PWM 5 output [15:0]
50	0000 2000		31:16		
		R/W	15:0	PWM6[15:0]	Select the width of PWM 6 output [15:0]
54	0000 2000		31:16		
		R/W	15:0	PWM7[15:0]	Select the width of PWM 7 output [15:0]
58	0000 2000		31:16		
		R/W	15:0	PWM8[15:0]	Select the width of PWM 8 output [15:0]
5C	0000 2000		31:16		
		R/W	15:0	PWM9[15:0]	Select the width of PWM 9 output [15:0]
60	0000 2000		31:16		
		R/W	15:0	PWM10[15:0]	Select the width of PWM 10 output [15:0]
64	0000 2000		31:16		
		R/W	15:0	PWM11[15:0]	Select the width of PWM 11 output [15:0]
68	3FFF		31:16		
		R/W	15:0	PERIOD0[15:0]	Select period of the PWM0 output[15:0]
6C	3FFF		31:16		
		R/W	15:0	PERIOD1[15:0]	Select period of the PWM1 output[15:0]
70	3FFF		31:16		
		R/W	15:0	PERIOD2[15:0]	Select period of the PWM2 output[15:0]
74	3FFF		31:16		
		R/W	15:0	PERIOD3[15:0]	Select period of the PWM3 output[15:0]
78	3FFF		31:16		
		R/W	15:0	PERIOD4[15:0]	Select period of the PWM4 output[15:0]
7C	3FFF		31:16		
		R/W	15:0	PERIOD5[15:0]	Select period of the PWM5 output[15:0]
80	3FFF		31:16		
		R/W	15:0	PERIOD6[15:0]	Select period of the PWM6 output[15:0]
84	3FFF		31:16		
		R/W	15:0	PERIOD7[15:0]	Select period of the PWM7 output[15:0]
88	3FFF		31:16		
		R/W	15:0	PERIOD8[15:0]	Select period of the PWM8 output[15:0]

Index	Default	R/W	Bit	Name	Description
8C	3FFF		31:16		
		R/W	15:0	PERIOD9[15:0]	Select period of the PWM9 output[15:0]
90	3FFF		31:16		
		R/W	15:0	PERIOD10[15:0]	Select period of the PWM10 output[15:0]
94	3FFF		31:16		
		R/W	15:0	PERIOD11[15:0]	Select period of the PWM11 output[15:0]

(a) If PWM_EN[11:0] = 12'h000, turn off PWM output
(b) PWM input Clock = 12 MHz
If PWM base clock = 1 MHz = 1 us
PWM output clock = 1 us * (PERIOD+1) * (PWM_CLKx +1))
MAX clock = 2 us * 1 = 2us = 500KHz
MIN clock = 65536 us * 128 = 8388608us =0.1192 Hz

PWM output clock (Min / Max)	12 MHz	24 MHz
PWM_BAS_CLK = 00 (12/24 MHz)	500KHz / 0.1192 Hz	1 MHz / 0.2384Hz
PWM_BAS_CLK = 01 (6/12 MHz)	250KHz / 0.0596 Hz	500KHz / 0.1192Hz
PWM_BAS_CLK = 10 (4/8 MHz)	166.6KHz / 0.03Hz	333.3KHz / 0.06Hz
PWM_BAS_CLK = 11 (1/2 MHz)	41.67KHz / 0.0099Hz	83.3KHz / 0.0198Hz

4.21 RTC

4.21.1 Features

- Time in hours, minutes, and seconds
- 24-hour format
- Calendar for weekday, date, month, and year
- Alarm configurable for minute, hour, day, and month
- Leap year correction
- Requires external 32.768 kHz clock crystal

4.21.2 Register Definition

RTC Register: address range 0x0020_0400 ~ 0x0020_07FF

Index	Default	R/W	Bit	Name	Description
00	00	R/W	31:8	Reserved	
		R/W	7	EN_RTC	1: Enable RTC 0: Disable RTC
		R/W	6	CHG_RTC_LCLK	Enable RTC clock = 32 KHz low frequency clock 1: can't not read/write RTC register (Power Down Mode) 0: can read/write RTC register
		R	5	RTC_ALARM	1: Event of alarm 0: No event of alarm
		R	4	RTC_1S	1: Event of RTC 1s 0: No event of RTC 1s
		R/W	3	ALARM_EN	1: Enable alarm wake-up and interrupt 0: Disable alarm
		R/W	2	1S_EN	1: Enable one second wake-up and interrupt 0: Disable one second
		W	1	CLR_ALARM	1: Clear event alarm interrupt 0: No clear event alarm interrupt
		W	0	CLR_RTC_1S	1: Clear event RTC 1s interrupt 0: No clear event RTC 1s interrupt
04	00	R/W	31:7	Reserved	
			6:0	RTC_SEC[6:0]	Second coded in BCD, range is 0~59. SEC [6:4] represents 10 seconds. SEC[3:0] represents seconds.
08	00	R/W	31:7	Reserved	
			6:0	RTC_MIN[6:0]	Minute coded in BCD, range is 0~59. MIN[6:4] represents 10 minutes. MIN[3:0] represents minutes.
0C	00	R/W	31:6	Reserved	
			5:0	RTC_HOUR[5:0]	Hour coded in BCD, range is 0~23. HOUR[5:4] represents 10 hours. HOUR[3:0] represents hours.
10	01	R/W	31:6	Reserved	
			5:0	RTC_DAY[5:0]	Day of month coded in BCD, range is 1~31. DAY[5:4] represents 10 days. DAY[3:0] represents days.
14	00	R/W	31:3	Reserved	
			2:0	RTC_WEEK[2:0]	Day of week. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday

Index	Default	R/W	Bit	Name	Description
					110: Saturday
18	01	R/W	31:4	Reserved	
			3:0	RTC_MONTH[3:0]	Month. 0001: January 0010: February 0011: March 0100: April 0101: May 0110: June 0111: July 1000: August 1001: September 1010: October 1011: November 1100: December
1C	00	R/W	31:8	Reserved	
			7:0	RTC_YEAR[7:0]	Year coded in BCD, range is 0~99. YEAR[7:4] represents 10 years. YEAR [3:0] represents years.
24	00	R/W	31:8	Reserved	
			7	RTC_MAE	Minute alarm enable. 0: Disable. 1: Enable.
			6:0	RTC_AMIN[6:0]	Alarm minute coded in BCD. Range is 0 ~ 59.
28	00	R/W	31:8	Reserved	
			7	RTC_HAE	Hour alarm enable. 0: Disable. 1: Enable.
			6	Reserved	
			5:0	RTC_AHOUR[5:0]	Alarm hour coded in BCD. Range is 0 ~ 23.
2C	01	R/W	31:8	Reserved	
			7	RTC_DAE	Day alarm enable. 0: Disable. 1: Enable.
			6	Reserved	
			5:0	RTC_ADAY[5:0]	Alarm day coded in BCD. Range is 1 ~ 31.
30	00	R/W	31:8	Reserved	
			7	RTC_WAE	Day of week alarm enable. 0: Disable. 1: Enable.
			6:3	Reserved	
			2:0	RTC_AWEEK[2:0]	Alarm day of week. Range is 0 ~ 6.
34	01	R/W	31:8	Reserved	
			7	RTC_TAE	Month alarm enable. 0: Disable. 1: Enable.
			6:4	Reserved	
			3:0	RTC_AMONTH[3:0]	Alarm month. Range is 1 ~ 12.

Note 1: CHG_RTC_LCLK= 1 ⇔ 0: must wait 32khz rising pulse to synchronize
MCU write CHG_RTC_LCLK=1, and need check CHG_RTC_LCLK=1 or wait 30us(1/32khz) MCU write CHG_RTC_LCLK=0, and need check CHG_RTC_LCLK=0 or wait 30us(1/32khz)
CHG_RTC_LCLK=0: RTC block clock=12 MHz, and catch crystal 32khz rising pulse. So MCU can directly read/write all RTC register
CHG_RTC_LCLK=1: RTC block clock=32khz, MCU can NOT write/read MCU directly

Note 2: OSC off procedure
(1) set CHG_RTC_LCLK= 1
(2) wait(check) read CHG_RTC_LCLK= 1 (max delay 1/32khz=30us)
(3) set OSC_OFF=1, clear OSC_OFF=0

4.22 IR

4.22.1 Register Definition

IR register: address range 0x0020_1800 ~ 0x0020_1BFF

Index	Default	R/W	Bit	Name	Description
00			31:8		Reserved
	0	R/W	7	EN_IR	1: Enable IR 0: Disable IR
	0	R/W	6	IR_SEDG	1: single edge trigger 0: both edge trigger
	0	R/W	5	IR_RF	1: rising edge trigger 0: falling edge trigger
	0	R/W	4	EN_OV_INT	1: Enable over flow interrupt 0: Disable over flow interrupt
	0	R/W	3:2	PRE_SCAL	IR Pre scaler time 00: 64us 01: 32us 10: 128us 11: 1024us
		W	1	CLR_IR_INT	1: Clear interrupt “IR_INT 0: No clear interrupt “IR_INT
	0	W	0	IR_PTSEL	Input Port Sel: 0: PortB[13] 1: PortF[4]
04			31:3		Reserved
		R	2	IR_HL	Read IR input H/L
		R	1	IR_OVFLW	1: IR over flow interrupt 0: No IR over flow interrupt
		R	0	IR_INT	1: IR interrupt = edge trigger + over flow 0: No IR interrupt
08			31:8		Reserved
		R	7:0	IR_CNT	IR counter
0C			31:4		Reserved
	0	R/W	3:0	IR_FILTER	IR digital filter =0H: 2*84ns = 168ns digital filter =1H: 1*32us = 32us digital filter =FH: 15*32us = 480us digital filter

Note-1: IR input: digital filter
Note-2: If IR_CNT[7:0] =8’hFF: H/W generate overflow interrupt & keep count = 8’hFF
Note-3: IR pulse: 440us –9ms, digital filter: 168ns – 480us
Note-4: IR_FILTER[4:0] clock base is “mcu_clk”

5. Electrical Characteristics

5.1 Absolute Maximum Ratings

Parameter	Min.	Typ.	Max.	Unit
Power supply voltage (3.0V supply)			3.6	V
Storage temperature	-60	-	150	ºC

5.2 Recommended Operating Condition

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
DVDD	DVDD Power supply		2.4		3.6	V
T_{opr}	Operation temperature		-40		85	ºC

5.3 Power Consumption

(DVDD=3.0V)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
I_{VDD48M} (Note 1)	Normal operating current @48 MHz			**30**		mA
I_{VDD24M} (Note 1)	Normal operating current @24 MHz			**21**		mA
I_{VDD12M} (Note 1)	Normal operating current @12 MHz			**16**		mA
$I_{VDDSUSP}$	OSC/XTAL off	No load on output		**6**		uA

Note 1: Power consumption is for reference only. The actual power consumption depends on chip setting, operation frequency, external loading, etc.

5.4 Digital I/O

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage		0.8 x DVDD		DVDD	V
V_{IL}	Input low voltage		0		0.2 x DVDD	V
V_{OH}	Output high voltage	DVDD=3.0V, I_{OH} = 4mA	2.4			V
V_{OL}	Output low voltage	DVDD=3.0V, I_{OL} = 4mA			0.4	V
R_{PU}	Pull up resistor			32		KΩ

5.5 LVD / LVR

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
I_{LVD}	LVD active current	DVDD=3.0V				V
V_{LVR}	LVR active voltage	VDD=1.8V	1.3	1.4	1.5	V
I_{LVR}	LVR active current	VDD=1.8V				μV

5.6 POR / BOR

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
V_{POR}	POR action voltage	DVDD=3.0V with 100ms ramp-up		1.85		V
V_{BOR}	BOR action voltage	DVDD=3.0V		1.85		V

5.7 ADC Converter

Symbol	Characteristic	Condition	Min.	Typ.	Max.	Unit
AVDD	ADC VDD Supply		2.4	3	3.6	V
V_{REF}	ADC External Input Reference		1.0		AVDD	V
V_{ADCI}	ADC input voltage		VSS		V_{REF}	V
F_s	ADC Sampling rate		0.05		12	MHz
F_{conv}	ADC Conversion Rate (SPS)				1	MHz
$R_{ADC\ (Note\ 1)}$	Sampling switch resistance				1	KΩ
$C_{ADC\ (Note\ 1)}$	Internal sample and hold capacitor				2	pF
ENOB	ADC accuracy			9		bit
I_{ADC}	ADC current	AVDD = 3.0V		7.5		mA

Note 1: All parameters are based on characterization, not tested in production.

Internal Reference and Temp Sensor

Symbol	Characteristic	Condition	Min.	Typ.	Max.	Unit
V1P25	Internal 1.25V Reference			1.25		V
V2P5	Internal 2.5V Reference			2.5		V
V_{TMPS}	Temp Sensor output voltage		0.65		1	V
F_{VTMPS}	Temp coefficient	Temp Sensor that depends on the temperature		6.7		mV/℃

Note 1: All parameters are based on characterization, not tested in production.

5.8 LDO

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
I_{LDO}	LDO 1.8V power current	VDD=3.0V with I_{load} = 0 uA		1.5		μA
I_{LDO}	LDO 1.8V power current	VDD=3.0V with I_{load} = 10 mA		500		μA

5.9 Internal 12/24/48 MHz RC Oscillator

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
F_{HRC}	RC frequency			48		MHz
$\Delta F_{RCH1}/F_{RC}$	Frequency tolerance(Note1)	without external crystal oscillator calibration (25℃)		±1		%
		without external crystal oscillator calibration (0℃~70℃)		±2		%
		without external crystal oscillator calibration (-20℃~85℃)		±3		%
		without external crystal oscillator calibration (-40℃~125℃)		±4		%
		with external crystal oscillator calibration (-40℃~85℃)			±1	%
I_{12MRC}	RCOSC operating current	F_{HRC} = 12 MHz		0.25		mA
		F_{HRC} = 24 MHz		0.5		mA
		F_{HRC} = 48 MHz		1		mA

Note 1: All parameters are based on characterization, not tested in production.

5.10 Internal 32 kHz RC Oscillator

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
F_{LRC}	LRC frequency	VDD = 3V, 25℃		32		kHz
$\Delta F_{RC}/F_{RC}$	Frequency tolerance	VDD = 2V ~5V, -40℃ ~ 85℃		±30		%
I_{LRC}	RCOSC operating current	F_{LRC} = 32 kHz			5	uA

5.11 PLL

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
F_{PLLI}	PLL input reference frequency		0.5		5	MHz
F_{VCO}	PLL VCO range		100		200	MHz
t_{LOC}	PLL lock time(Note 1)	F_{PLLI} = 0.5 MHz, F_{VCO} = 200 MHz		400		μS
I_{PLL}	PLL operation current			1.3		mA

Note 1: Parameter is based on characterization, not tested in production. For normal application, customer has to check PLL lock flag to switch to PLL clock

Note 2: To change PLL configuration, PLL has to be power down first. Power on PLL again after finish PLL configuration and wait for PLL lock flag set to switch to PLL clock.

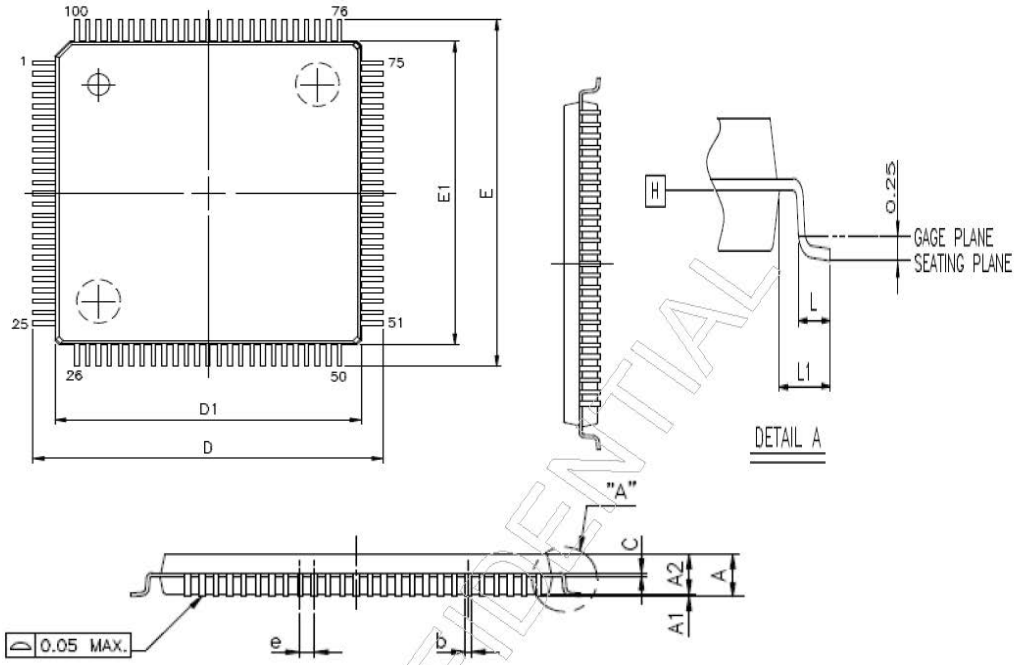
5.12 ESD & Latch Up

Symbol	Parameter	Condition	Class	Max.	Unit
V_{ESDHBM}	Electrostatic discharge voltage (human body model)	T_A = +25 °C conforming to MIL-STD-883G Method 3015.7	3A	4	KV
V_{ESDMM}	Electrostatic discharge voltage (machine model)	T_A = +25 °C conforming to JESD22-A115	C	400	V
LU	Static latch-up class	T_A = +25 °C conforming to JESD78	I	400	mA

6. Package Characteristics

6.1 LQFP100 Outline Drawing

Low-Profile Quad Flat Package **LQFP-100 PIN**



SYMBOLS	MIN	NOR	MAX
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
b	0.17	0.20	0.27
c	0.09	0.127	0.20
D	16.00 BSC		
D1	14.00 BSC		
E	16.00 BSC		
E1	14.00 BSC		
e	0.50 BSC		
L	0.45	0.60	0.75
L1	1.00 REF		

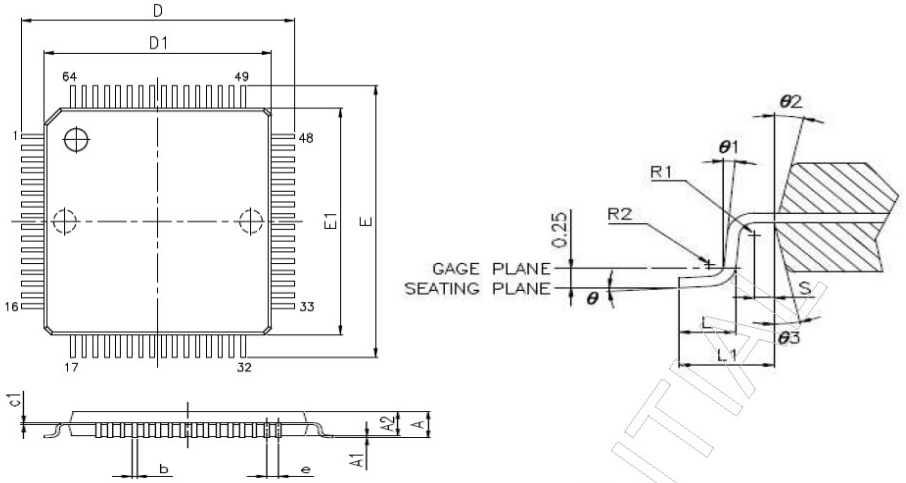
UNIT: mm

NOTES:
1. JEDEC outline : MS-026 BED
2. Datum plane H is located at the bottom of the mold parting line coincident with where the lead exits the body.
3. Dimensions "D1" and "E1" do not include mold protrusion. Allowable protrusion is 0.25mm per side. "D1" and "E1" are maximum plastic body size dimensions including mold mismatch and are determined at datum plane H.
4. Dimension "b" does not include dambar protrusion.

PREPARE	Cynthia	DATE: 2012/7/27
CHECK	Lawrence	DATE: 2012/7/27
APPROVE	Eric	DATE: 2012/7/27

6.2 LQFP64 Outline Drawing

Low-Profile Quad Flat Package **LQFP-64 PIN**



SYMBOLS	MIN	NOR	MAX
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
b	0.17	0.22	0.27
c1	0.09	-	0.16
D	11.75	12.00	12.25
D1	9.90	10.00	10.10
E	11.75	12.00	12.25
E1	9.90	10.00	10.10
e		0.50 BSC	
L	0.45	0.60	0.75
L1		1.00 REF	
S		0.20 REF	
θ°		3.5 REF	
θ1°		5.0 REF	
θ2°		12 REF	
θ3°		12 REF	
R1		0.16 REF	
R2		0.15 REF	

UNIT: mm

NOTES:
1. JEDEC outline : MS-026 BCD
2. Dimensions “D1” and “E1” do not include mold protrusion. Allowable protrusion is 0.25mm per side. “D1” and “E1” are maximum plastic body size dimensions including mold mismatch.
3. Dimension “b” does not include dambar protrusion. Allowable dambar protrusion shall not cause the lead width to exceed the maximum “b” dimension by more than 0.08mm.

PREPARE	Cynthia	DATE: 2012/7/27
CHECK	Lawrence	DATE: 2012/7/27
APPROVE	Eric	DATE: 2012/7/27

7. Revision History

Version	History	Date
1.0	Initial issue	August 2014
1.01	Revision	September 2014
1.02	1. Pin description update 2. PLL update 3. GPIO special function update 4. IR Interrupt vector update 5. UART description update	January 2015
1.03	1. Pin description update 2. TYPO correction	January 2015
1.031	TYPO correction (p. 64)	January 2015
1.032	TYPO correction	January 2015
1.04	Feature update	February 2015