zilog

Embedded in Life

An IXYS Company

ZNEO32! Family of Microcontrollers

Z32F0642 MCU

Product Specification

PS039201-0217

PRELIMINARY

ZNEO32!

32 Bit Microcontrollers

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

ZNEO32! is a trademark or registered trademark of Zilog, Inc. All other product or service names are the property of their respective owners.

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Description	Page
Feb 2017	01	Original issue.	

1. Overview

Introduction

Zilog's Z32F0642 microcontroller, a member of the ZNEO32! Family of microcontrollers, is a cost-effective and high performance 32-bit microcontroller that is ideal for use in motor applications.

This Z32F0642 MCU offers 3-phase PWM generator units which are suitable for inverter motor drive systems. A built-in 3-phase PWM generator controls one inverter motor. One 12-bit high speed ADC unit with 12-channel analog multiplexed inputs is included to gather feedback from the motor. This MCU can control up to one inverter motor. Multiple powerful external serial interfaces help communicate with on-board sensors and devices.

Figure 1-1 shows a block diagram of the Z32F0642 MCU.

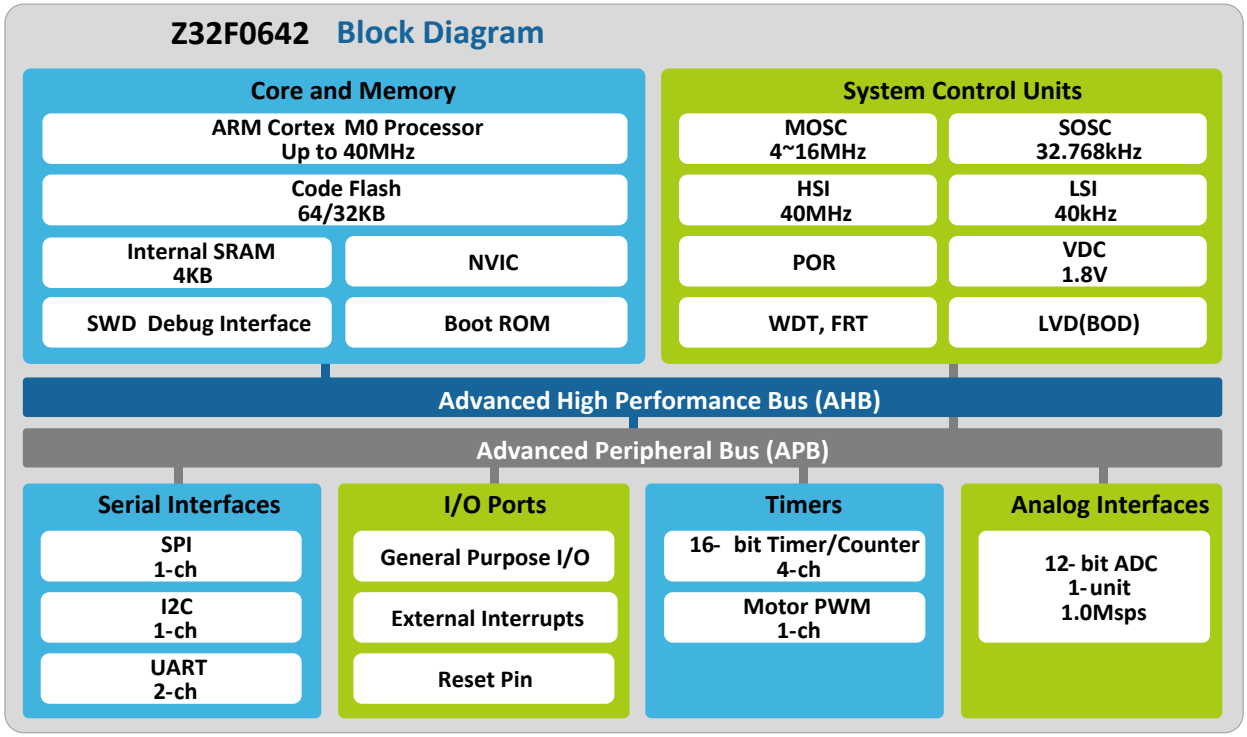


Figure 1-1 Block Diagram

Figure 1-2 displays the pin layout of the Z32F06423AKE MCU.

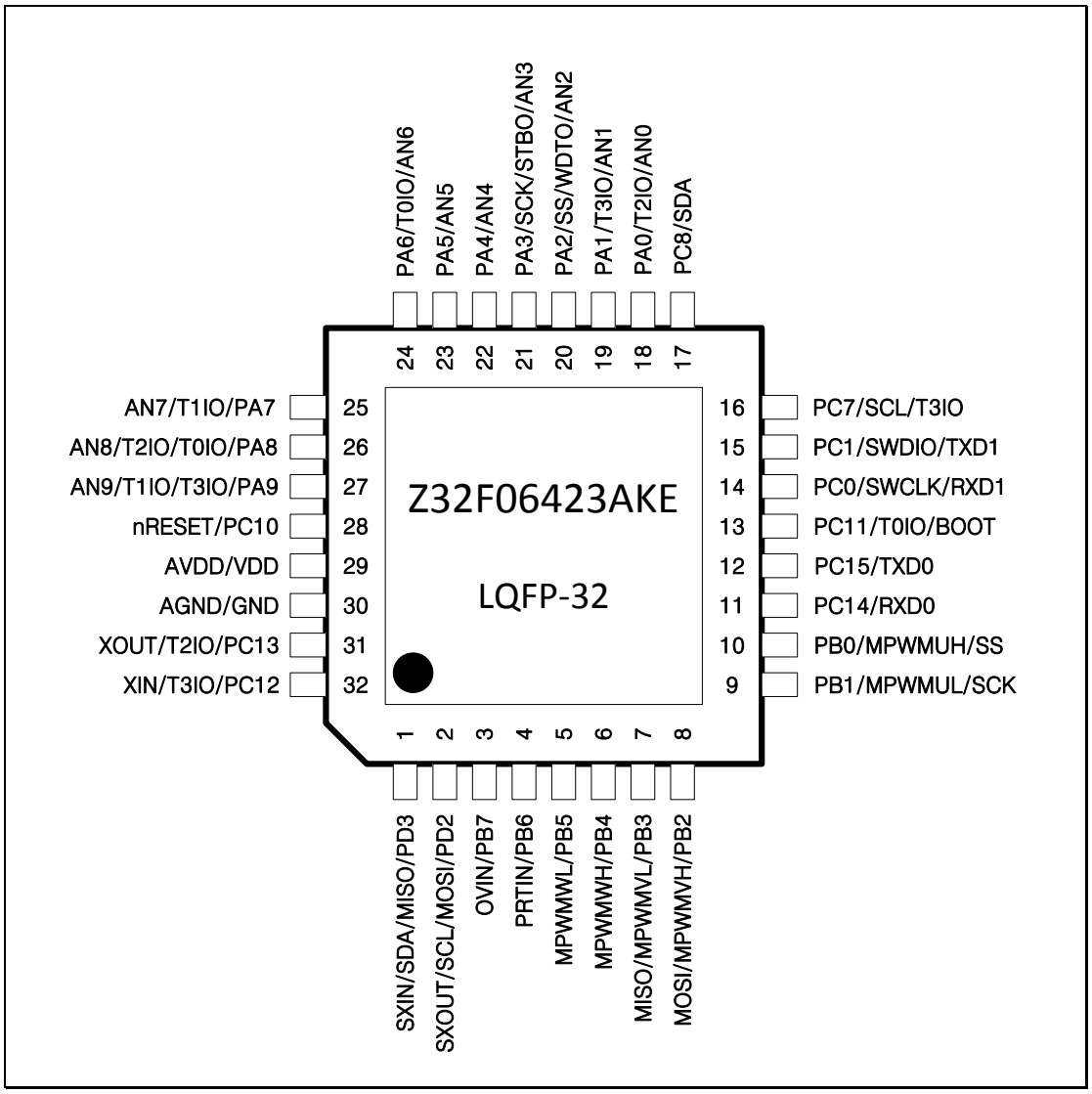


Figure 1-2 Pin Layout (LQFP-32)

Figure 1-3 displays the pin layout of the Z32F06423AEE MCU.

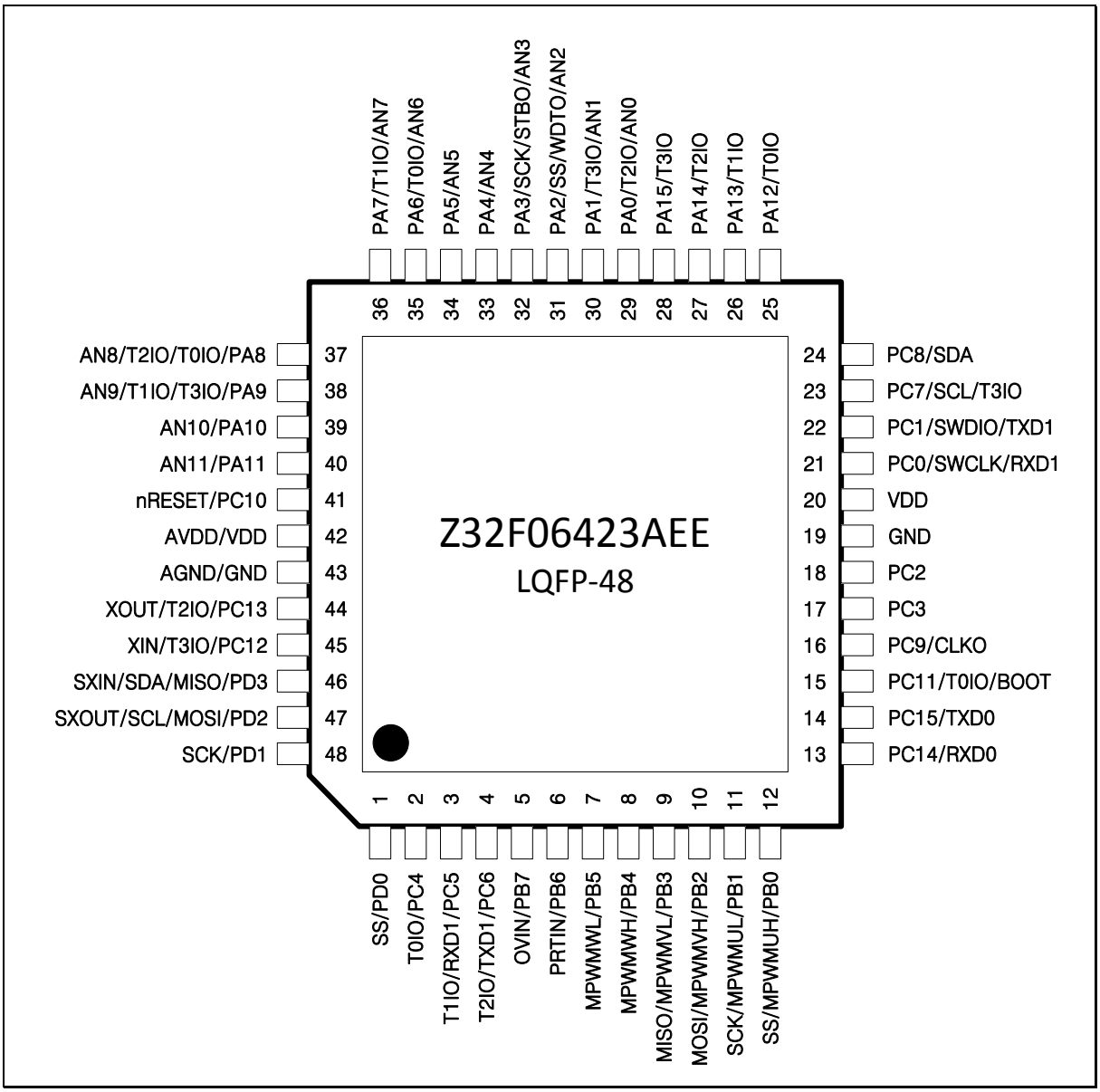


Figure 1-3 Pin Layout (LQFP-48)

Product Features

The Z32F0642 MCU offers the following features:

- High performance, low-power Cortex-M0 core
- 64 KB code Flash memory
  - Endurance : 10,000 times at room temperature
  - Retention : 10 years
- 4 KB SRAM
- General Purpose I/O (GPIO)
  - 44 ports (PA[15:0], PB[7:0], PC[15:0], PD[3:0]) : 48-Pin
  - 30Ports (PA[9:0], PB[7:0], PC[1:0], PC[8:7], PC[15:10], PD[3:2]) : 32-Pin
- 3-phase Motor PWM (MPWM) with ADC triggering function
  - 1-channel
- 1 MSPS high-speed 12-bit ADC with sequential conversion function
  - 12-channel : 48-Pin
  - 10-channel : 32-Pin
- Timer
  - 16-bit 4-channel
- Free Run Timer (FRT)
  - 32-bit 1-channel
- Watchdog Timer (WDT)
  - 32-bit 1-channel
- External communication ports:
  - 2-channel UARTs
  - 1-channel I^2C
  - 1-channel SPI
- Hardware Divider (DIV64)
- On-chip RC-oscillator
  - HSI : 40 MHz(±3% @-40 ~ +105℃)
  - LSI : 40 kHz(±20% @-40 ~ +105℃)
- System fail-safe function by clock monitoring
- XTAL OSC fail monitoring
- Power on reset
- Programmable low voltage detector (brown-out detector)
- Debug and emergency stop function
- SWD debugger
- Supports UART and SPI ISP
- Power down mode
  - IDLE, STOP1, STOP2 modes
- Sub-active mode
  - System used external 32.768 kHz crystal or system used internal 40 kHz LSI
- Operating frequency
  - 40 kHz ~ 40 MHz

  - External 32.768 kHz crystal
- Operating voltage
  - 2.2 V ~ 5.5 V
- Two package options:
  - LQFP-32
  - LQFP-48

Table 1-1 lists the device information.

Table 1-1 Device Type

Part Number	Flash	SRAM	UART	SPI	I2C	MPWM	ADC	I/O Ports	Package
Z32F06423AKE	64KB	4KB	2	1	1	1	1-unit 10 ch	30	LQFP-32
Z32F06423AEE	64KB	4KB	2	1	1	1	1-unit 12 ch	44	LQFP-48

Architecture

Figure 1-4 shows the block diagram of the Z32F0642 MCU.

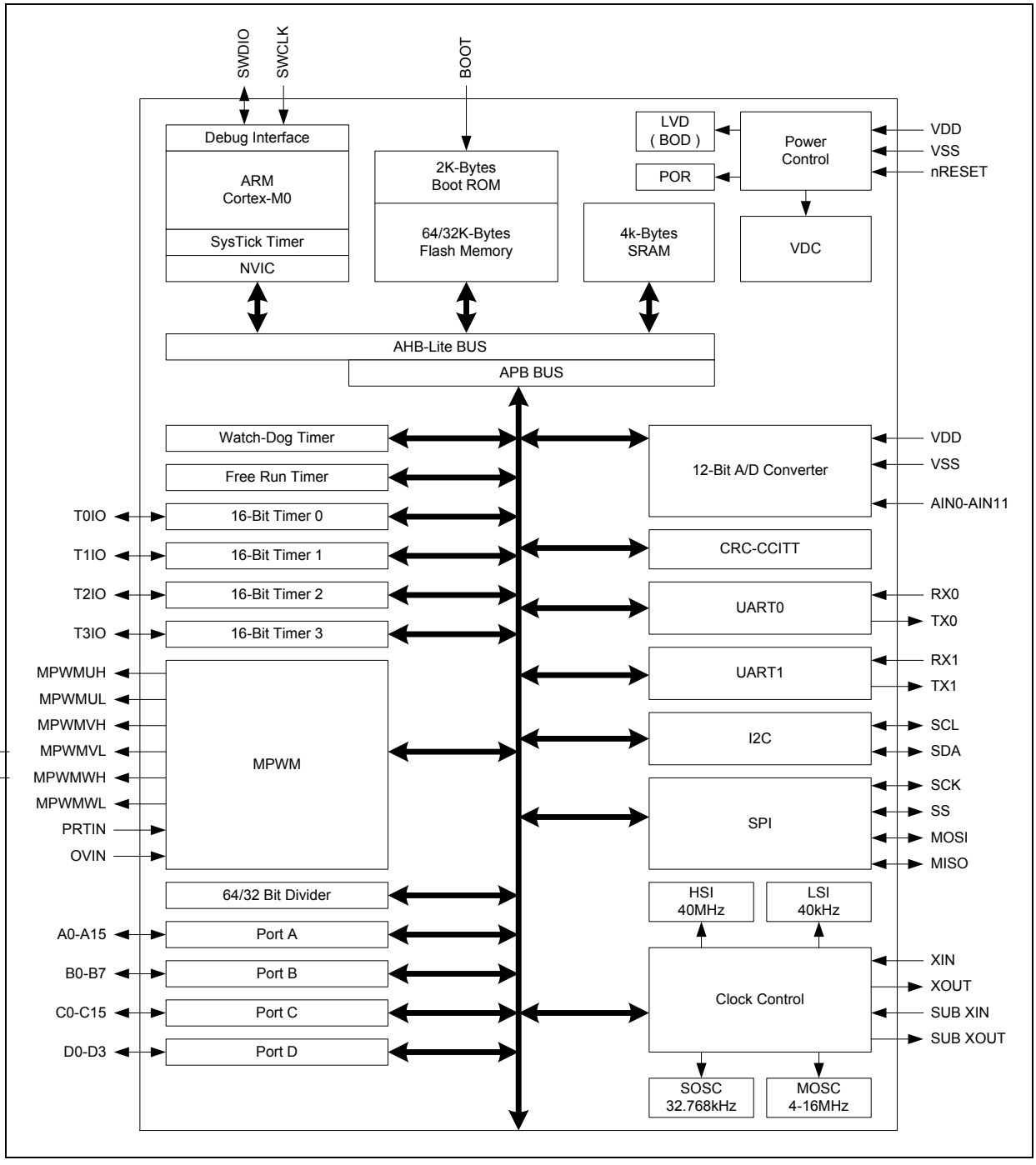


Figure 1-4 Block Diagram

Functional Description

The following section provides an overview of the features of the Z32F0642 microcontroller.

ARM Cortex-M0

- ARM powered Cortex-M0 core based on ARMv6M architecture which is optimized for small size and low power systems
- On-core system timer (SYSTICK) provides a simple 24-bit timer that makes it easy to manage the system operation
- Thumb-compatible Thumb-2 only instruction set processor core makes code high-density
- Hardware division and single-cycle multiplication
- Integrated Nested Vectored Interrupt Controller (NVIC) provides deterministic interrupt handling
- SWD debugging features
- Max 40 MHz operating frequency with one wait execution

Nested Vector-Interrupt Controller (NVIC)

- The ARM Nested Vectored Interrupt Controller (NVIC) on the ARM Cortex-M0 core handles all internal and external exceptions. When an interrupt condition is detected, the processor state is automatically stored to the stack and automatically restored from the stack at the end of the interrupt service routine
- The vector is fetched in parallel to the state saving, which enables efficient interrupt entry
- The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoring

64/32KB Internal Code Flash Memory

- The Z32F0642 MCU provides internal 64/32KB code Flash memory and its controller. This is sufficient to program motor algorithms and generally control the system. Self-programming is available and ISP and SWD programming is also supported in boot or debugging mode.
- Instruction and data cache buffers are available and overcome the low bandwidth Flash memory. The CPU can access Flash memory with one wait state up to 40 MHz bus frequency.

4KB 0-wait Internal SRAM

- On chip 4 KB 0-wait SRAM can be used for working memory space and program code can be loaded on this SRAM.

Boot Logic

- The smart boot logic supports Flash programming. The Z32F0642 MCU can be accessed by the external boot pin and UART and SPI programming are available in Boot mode. UART0 or SPI is used in boot mode communication.

System Control Unit (SCU)

- The SCU block manages internal power, clock, reset, and operation modes. It also controls analog blocks (Oscillator Block, VDC and BOD (LVD)).

32-bit Watchdog Timer (WDT)

- The watchdog timer performs the system monitoring function. It generates an internal reset or

interrupt to notice abnormal status of the system.

Multi-purpose 16-bit Timer

- Four-channel 16-bit general purpose timers support the following functions:
  - Periodic timer mode
  - Counter mode
  - PWM mode
  - Capture mode
- Built-in timer also supports counter-synchronization mode, which can generate synchronized waves and timing.

Motor PWM Generator

- 3-phase Motor PWM Generator is implemented. 16-bit up/down counter with prescaler supports triangular and saw tooth waveform
- The PWM has the ability to generate an internal ADC trigger signal to measure the signal on time
- Dead time insertion and emergency stop functionality help the chip and system maintain safety conditions

Serial Peripheral Interface (SPI)

- Synchronous serial communication is provided with the SPI block. The Z32F0642 MCU has a 1-channel SPI module. Boot mode uses this SPI block to download the Flash program.

Inter-Integrated Circuit Interface (I^2C)

- The Z32F0642 MCU has a 1-channel I^2C block and it supports up to 400 kHz I^2C communication. Master and the Slave modes are supported.

Universal Asynchronous Receiver/Transmitter (UART)

- The Z32F0642 MCU has a 2-channel UART block. For accurate baud rate control, a fractional baud rate generator is provided.

General PORT I/Os

- 16-bit PA, 8-bit PB, 16-bit PC, and 4-bit PD ports are available and provide the following functionality:
  - General I/O port
  - Independent bit set/clear function
  - External interrupt input port
    - Programmable pull-up and open-drain selection
    - On-chip input debounce filter

12-bit Analog-to-Digital Converter (ADC)

- One built-in ADC unit can convert analog signal up to 1 MSPS (sample per second) conversion rate. The 12-channel analog MUX provides various combinations from external analog signals.

Hardware Divider (DIV64)

- The divider module provides a hardware divider with the ability to accelerate complicated

calculations. This divider is a sequential 64-bit/32-bit divider that requires 32 clock cycles for one operation.

Pin Description

The pin configurations listed in Table 1-2 contain two pairs of power/ground pins and other dedicated pins. These multi-function pins provide four selections of functions including GPIO. The configuration, including pin ordering, can be changed without notice.

Table 1-2 Pin Description

Pin No		Pin Name	Type	Description	Remark
LQFP-48	LQFP-32				
1	-	PD0	IOUS	PORT D Bit 0 Input/Output	
		SS	I/O	SPI Channel Slave Select In/Out	
2	-	PC4	IOUS	PORT C Bit 4 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	
3	-	PC5	IOUS	PORT C Bit 5 Input/Output	
		RXD1	I	Uart RXD1 Input	
		T1IO	I/O	Timer 1 Input/Output	
4	-	PC6	IOUS	PORT C Bit 6 Input/Output	
		TXD1	O	Uart TXD1 Output	
		T2IO	I/O	Timer 2 Input/Output	
5	3	PB7	IOUS	PORT B Bit 7 Input/Output	
		OVIN	I	PWM Over-voltage input signal	
6	4	PB6	IOUS	PORT B Bit 6 Input/Output	
		PRTIN	I	PWM Protection Input signal	
7	5	PB5	IOUS	PORT B Bit 5 Input/Output	
		MPWMWL	O	MPWM WL Output	
8	6	PB4	IOUS	PORT B Bit 4 Input/Output	
		MPWMWH	O	MPWM WH Output	
9	7	PB3	IOUS	PORT B Bit 3 Input/Output	
		MPWMVL	O	MPWM VL Output	
		MISO	I/O	SPI Channel Master In / Slave Out	
10	8	PB2	IOUS	PORT B Bit 2 Input/Output	
		MPWMVH	O	MPWM VH Output	
		MOSI	I/O	SPI Channel Master Out / Slave In	
11	9	PB1	IOUS	PORT B Bit 1 Input/Output	
		MPWMUL	O	MPWM UL Output	
		SCK	I/O	SPI Channel CLK In / Out	
12	10	PB0	IOUS	PORT B Bit 0 Input/Output	
		MPWMUH	O	MPWM UH Output	
		SS	I/O	SPI Channel Slave Select In / Out	
13	11	PC14	IOUS	PORT C Bit 14 Input/Output	
		RXD0	I	Uart RXD0 Input	
14	12	PC15	IOUS	PORT C Bit 15 Input/Output	
		TXD0	O	Uart TXD0 Output	
15	13	PC11	IOUS	PORT C Bit 11 Input/Output	
		BOOT	IU	Boot mode Selection Input	Pull-up
		T0IO	I/O	Timer 0 Input/Output	
16	-	PC9	IOUS	PORT C Bit 9 Input/Output	
		CLKO	O	System Clock Output	

17	-	PC3	IOUS	PORT C Bit 3 Input/Output	
18	-	PC2	IOUS	PORT C Bit 2 Input/Output	
19	-	GND	P	GND	
20	-	VDD	P	VDD	
21	14	PC0	IOUS	PORT C Bit 0 Input/Output	
		SWCLK	I	SWD Clock Input	Pull-up
		RXD1	I	Uart1 RXD1 Input	
22	15	PC1	IOUS	PORT C Bit 1 Input/Output	
		SWDIO	I/O	SWD Data Input/Output	Pull-up
		TXD1	O	Uart1 TXD1 Output	
23	16	PC7	IOUS	PORT C Bit 7 Input/Output	
		SCL	I/O	I^2C Channel SCL In/Out	
		T3IO	I/O	Timer 3 Input/Output	
24	17	PC8	IOUS	PORT C Bit 8 Input/Output	
		SDA	I/O	I^2C Channel SDA In/Out	
25	-	PA12	IOUS	PORT A Bit 12 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	
26	-	PA13	IOUS	PORT A Bit 13 Input/Output	
		T1IO	I/O	Timer 1 Input/Output	
27	-	PA14	IOUS	PORT A Bit 14 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
28	-	PA15	IOUS	PORT A Bit 15 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	
29	18	PA0	IOUS	PORT A Bit 0 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
		AIN0	IA	Analog Input 0	
30	19	PA1	IOUS	PORT A Bit 1 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	
		AIN1	IA	Analog Input 1	
31	20	PA2	IOUS	PORT A Bit 2 Input/Output	
		SS	I/O	SPI Channel Slave Select In / Out	
		WDTO	O	Watchdog Timer Overflow Output	
		AIN2	IA	Analog Input 2	
32	21	PA3	IOUS	PORT A Bit 3 Input/Output	
		SCK	I/O	SPI Channel CLK In / Out	
		STBO	O	Power Down Mode Output	
		AIN3	IA	Analog Input 3	
33	22	PA4	IOUS	PORT A Bit 4 Input/Output	
		AIN4	IA	Analog Input 4	
34	23	PA5	IOUS	PORT A Bit 5 Input/Output	
		AIN5	IA	Analog Input 5	
35	24	PA6	IOUS	PORT A Bit 6 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	
		AIN6	IA	Analog Input 6	
36	25	PA7	IOUS	PORT A Bit 7 Input/Output	
		T1IO	I/O	Timer 1 Input/Output	
		AIN7	IA	Analog Input 7	
37	26	PA8	IOUS	PORT A Bit 8 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	
		AIN8	IA	Analog Input 8	
38	27	PA9	IOUS	PORT A Bit 9 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	

		T1IO	I/O	Timer 1 Input/Output	
		AIN9	IA	Analog Input 9	
39	-	PA10	IOUS	PORT A Bit 10 Input/Output	
		AIN10	IA	Analog Input 10	
40	-	PA11	IOUS	PORT A Bit 11 Input/Output	
		AIN11	IA	Analog Input 11	
41	28	PC10	IOUS	PORT C Bit 10 Input/Output	
		nRESET	IU	External Reset Input	Pull-up
42	29	VDD	P	VDD	
43	30	GND	P	GND	
44	31	PC13	IOUS	PORT C Bit 13 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
		XOUT	OA	External Crystal Oscillator Output	
45	32	PC12	IOUS	PORT C Bit 12 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	
		XIN	IA	External Crystal Oscillator Input	
46	1	PD3	IOUS	PORT D Bit 3 Input/Output	
		MISO	I/O	SPI Channel Master In / Slave Out	
		SDA	I/O	I^2C Channel SDA In/Out	
		SXIN	I	External Crystal Sub Oscillator Input	
47	2	PD2	IOUS	PORT D Bit 2 Input/Output	
		MOSI	I/O	SPI Channel Master Out / Slave In	
		SCL	I/O	I^2C Channel SCL In/Out	
		SXOUT	OA	External Crystal Sub Oscillator Output	
48	-	PD1	IOUS	PORT D Bit 1 Input/Output	
		SCK	I/O	SPI Clock Input/Output	

Notation: I=Input, O=Output, U=Pull-up, D=Pull-down,
S=Schmitt-Trigger Input Type, C=CMOS Input Type, A=Analog, P=Power
Pin order may be changed with revision notice

Memory Map

Figure 1-5 shows the memory map.

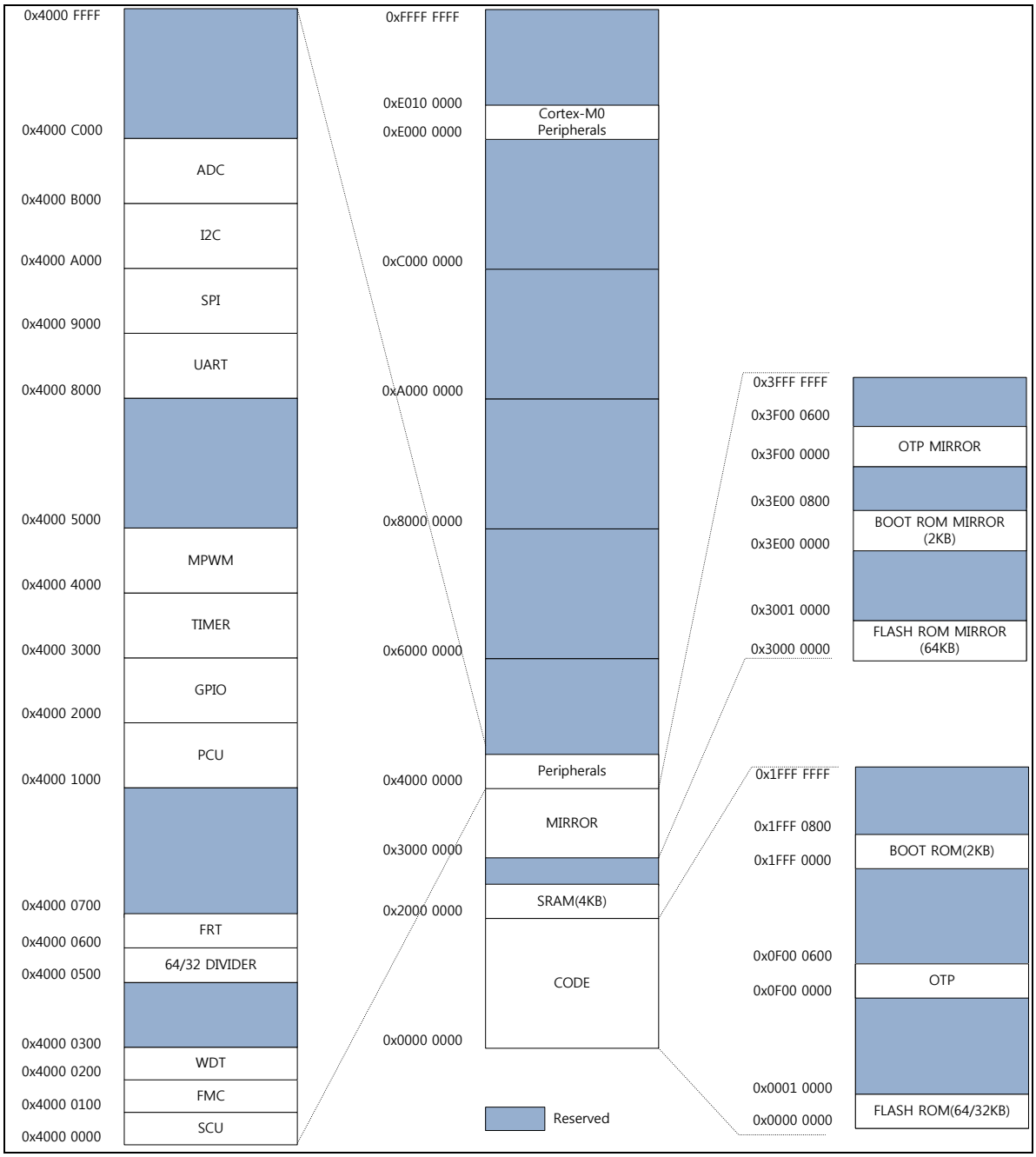


Figure 1-5 Memory Map

2. CPU

Cortex-M0 Core

The CPU core is supported by the ARM Cortex-M0 processor, which provides a high-performance, low-cost platform. To learn more about Cortex M0, refer to document number DDI0432C from ARM.

Interrupt Controller

Table 2-1 shows the Interrupt Vector Map.

Table 2-1 Interrupt Vector Map

Priority	Vector Address	Interrupt Source
-16	0x0000_0000	Stack Pointer
-15	0x0000_0004	Reset Address
-14	0x0000_0008	NMI Handler
-13	0x0000_000C	Hard Fault Handler
-12	0x0000_0010	MPU Fault Handler
-11	0x0000_0014	BUS Fault Handler
-10	0x0000_0018	Usage Fault Handler
-9	0x0000_001C	Reserved
-8	0x0000_0020	
-7	0x0000_0024	
-6	0x0000_0028	
-5	0x0000_002C	SVCall Handler
-4	0x0000_0030	Debug Monitor Handler
-3	0x0000_0034	Reserved
-2	0x0000_0038	PenSV Handler
-1	0x0000_003C	SysTick Handler
0	0x0000_0040	LVDFAIL
1	0x0000_0044	SYSCLKFAIL
2	0x0000_0048	MOSCFAIL
3	0x0000_004C	SOSCFAIL
4	0x0000_0050	WDT
5	0x0000_0054	TIMER0
6	0x0000_0058	TIMER1
7	0x0000_005C	TIMER2
8	0x0000_0060	TIMER3
9	0x0000_0064	FRT
10	0x0000_0068	GPIOAE

11	0x0000_006C	GPIOAO
12	0x0000_0070	GPIOBE
13	0x0000_0074	GPIOBO
14	0x0000_0078	GPIOCE
15	0x0000_007C	GPIOCO
16	0x0000_0080	GPIODE
17	0x0000_0084	GPIODO
18	0x0000_0088	MPWM
19	0x0000_008C	MPWMPROT
20	0x0000_0090	MPWMOVV
21	0x0000_0094	I2C
22	0x0000_0098	SPI
23	0x0000_009C	UART0
24	0x0000_00A0	UART1
25	0x0000_00A4	ADC
26	0x0000_00A8	Reserved
27	0x0000_00AC	
28	0x0000_00B0	
29	0x0000_00B4	
30	0x0000_00B8	
31	0x0000_00BC	

Note:

Each interrupt has an associated priority-level register. Each of them is 2 bits wide, occupying the two MSBs of the Interrupt Priority Level registers. Each Interrupt Priority Level register occupies 1 byte (8 bits). NVIC registers in the Cortex-M0 processor can only be accessed using word-size transfers, so for each access, four Interrupt Priority Level registers are accessed at the same time.

3. Boot Mode

Boot Mode Pins

The Z32F0642 MCU has a Boot mode option to program internal Flash memory. Enter Boot mode by setting the BOOT pin to ‘L’ at reset timing. (Normal state is ‘H’).

Boot mode supports UART boot and SPI boot. UART boot uses the UART0 port, and SPI boot uses SPI. The pins for Boot mode are listed in Table 3-1.

Table 3-1 Boot Mode Pins

Block	Pin Name	Dir	Description
SYSTEM	nRESET/PC10	I	Reset Input signal
	BOOT/PC11	I	‘0’ to enter Boot mode
UART0	RXD0/PC14	I	UART Boot Receive Data
	TXD0/PC15	O	UART Boot Transmit Data
SPI	SS/PA2	I	SPI Boot Slave Select
	SCK/PA3	I	SPI Boot Clock Input
	MOSI/PD2	I	SPI Boot Data Input
	MISO/PD3	O	SPI Boot Data Output

Boot Mode Connections

Users can design the target board using either of the Boot mode ports – UART or SPI.

Figures 3-1 through 3-3 show sample Boot mode connection diagrams.

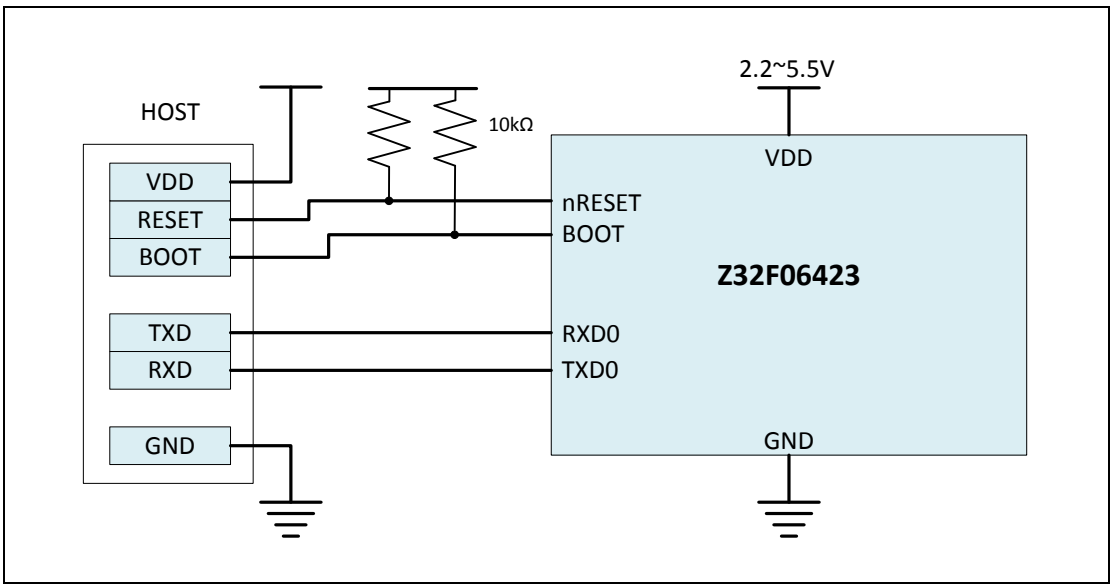


Figure 3-1 Boot Mode Connection Diagram

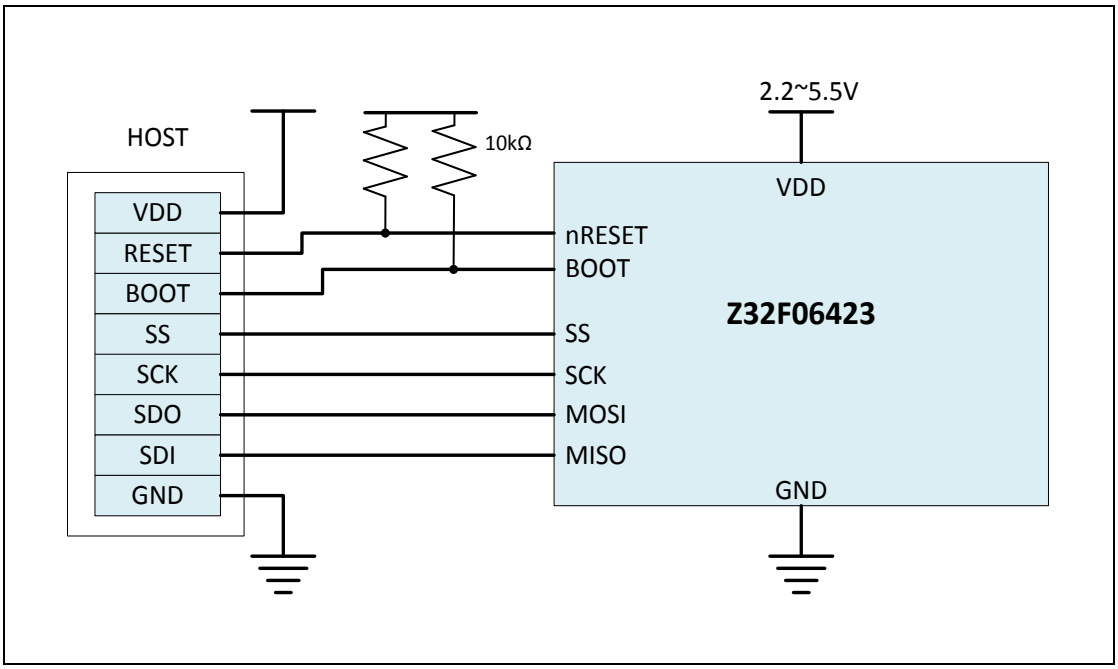


Figure 3-2 SPI Boot Connection Diagram

ISP Mode Connections

Users can design the target board using any ISP mode port.

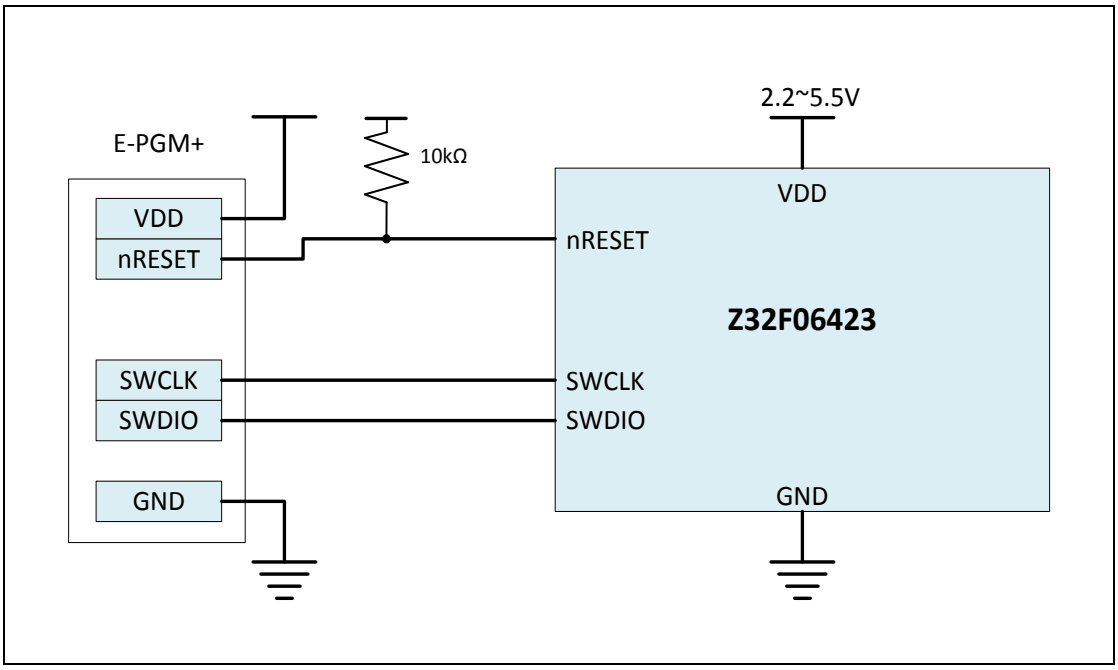


Figure 3-3 ISP and E-PGM+ Connection Diagram

4. System Control Unit (SCU)

Overview

The Z32F0642 MCU has a built-in intelligent power control block which manages system analog blocks and operating modes. Internal reset and clock signals are controlled by the SCU block to optimize system performance and power dissipation.

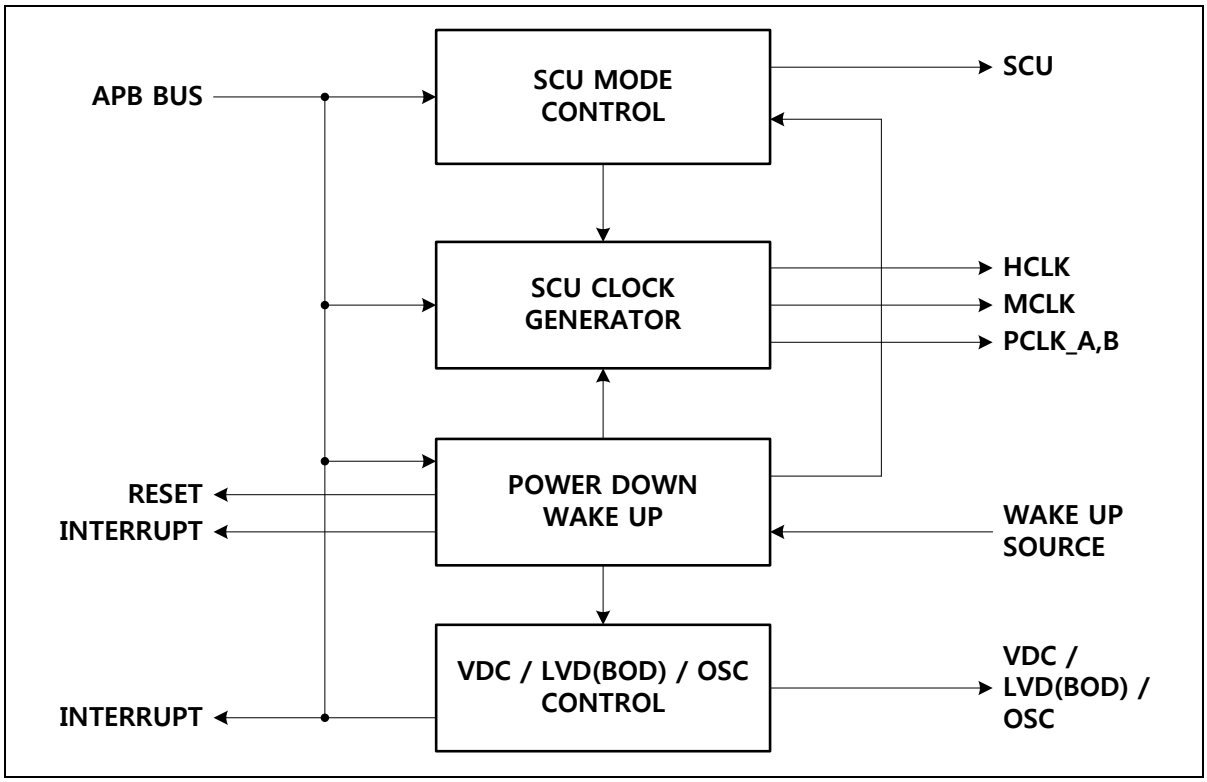


Figure 4-1 SCU Block Diagram

Clock System

The Z32F0642 MCU has two main operating clocks. One is HCLK which supplies the clock to the CPU and AHB bus system. The other clock is PCLK which supplies the clock to Peripheral systems.

Users can control the clock system variation with software. Figure 4-2 shows the clock system of the chip.

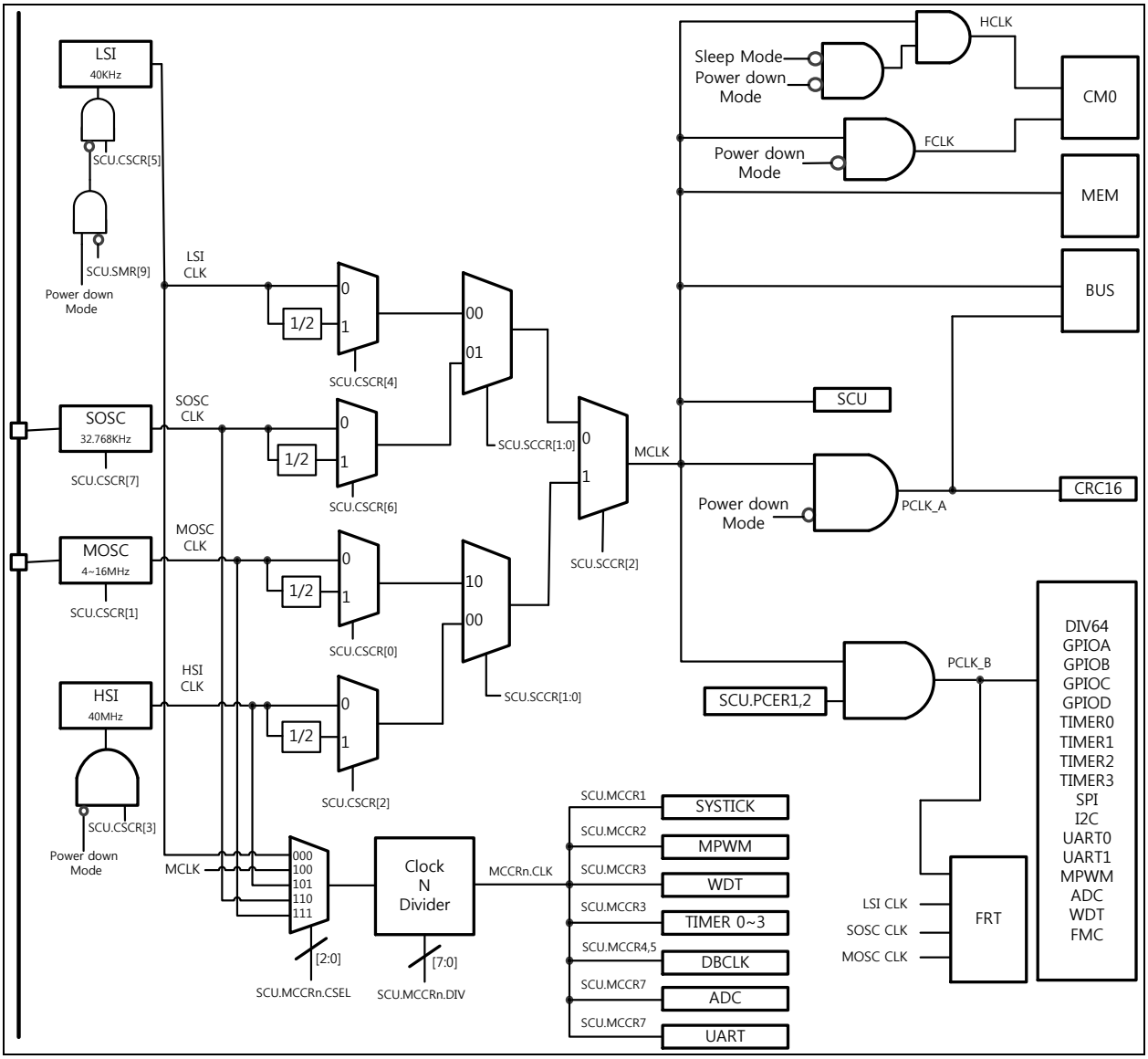


Figure 4-2 Clock Tree Configuration

Each of the mux to switch clock sources has a glitch-free circuit. Therefore, the clock can be switched without risk of glitches occurring. When you try to change the clock mux control, both clock sources should be alive. If one of them is not alive, the clock change operation is stopped and the system will be halted and not be recovered.

Table 4-1 lists the clock sources and their description.

Table 4-1 Clock Sources

Clock Name	Frequency	Description
MOSC	4-16 MHz	External Crystal OSC
SOSC	32.768 kHz	External Sub Crystal OSC
HSI	40 MHz	High Speed Internal OSC
LSI	40 kHz	Low Speed Internal OSC

HCLK Clock Domain

The HCLK clock feeds the clock to the CPU and AHB bus. Cortex-M0 CPU requires two clocks related to the HCLK clock, FCLK and HCLK. FCLK is the free running clock and is always running except in Power Down mode. HCLK can be stopped in Sleep mode and Power Down mode.

The bus system and memory systems are operated by the MCLK clock. The maximum bus operating clock speed is 40 MHz.

PCLK Clock Domain

PCLK_B is the master clock of all the peripherals. Each peripheral's clock is enabled in the SCU.PCER1 and SCU.PCER2 registers. Prior to enabling the PCLK_B input clock of each block, the peripheral is not accessible, even to read its registers. For FRT, various clocks can be used; however, CRC16 uses PCLK_A. This clock can be stopped in Power Down mode.

Clock Configuration Procedure

After power up, the default system clock is fed by the LSI (40 kHz) clock. LSI is enabled by default during the power up sequence. The other clock sources are enabled by user controls with the LSI system clock.

The HSI (40 MHz) clock can be enabled by the SCU.CSCR register.

The MOSC (4-16 MHz) clock can be enabled by the SCU.CSCR register. Before enabling the MOSC block, the pin mux configuration should be set for XIN, XOUT function. PC12 and PC13 pins are shared with MOSC's XIN and XOUT function – PCC.MR and PCC.CR registers should be correctly configured. After enabling the MOSC block, it is necessary to wait for more than 5 msec to ensure stable operation of crystal oscillation.

The SOSC (32.768 kHz) clock can be enabled by the SCU.CSCR register. Before enabling the SOSC block, the pin mux configuration should be set for SXIN, SXOUT function. PD3 and PD2 pins are shared with SOSC's SXIN and SXOUT function – PCD.MR and PCD.CR registers should be correctly configured. After enabling the SOSC block, it is necessary to wait for more than 10 msec to ensure stable operation of crystal oscillation.

You can change MCLK using the SCU.SCCR register. Figure 4-3 shows an example flow chart of the process to configure the system clock.

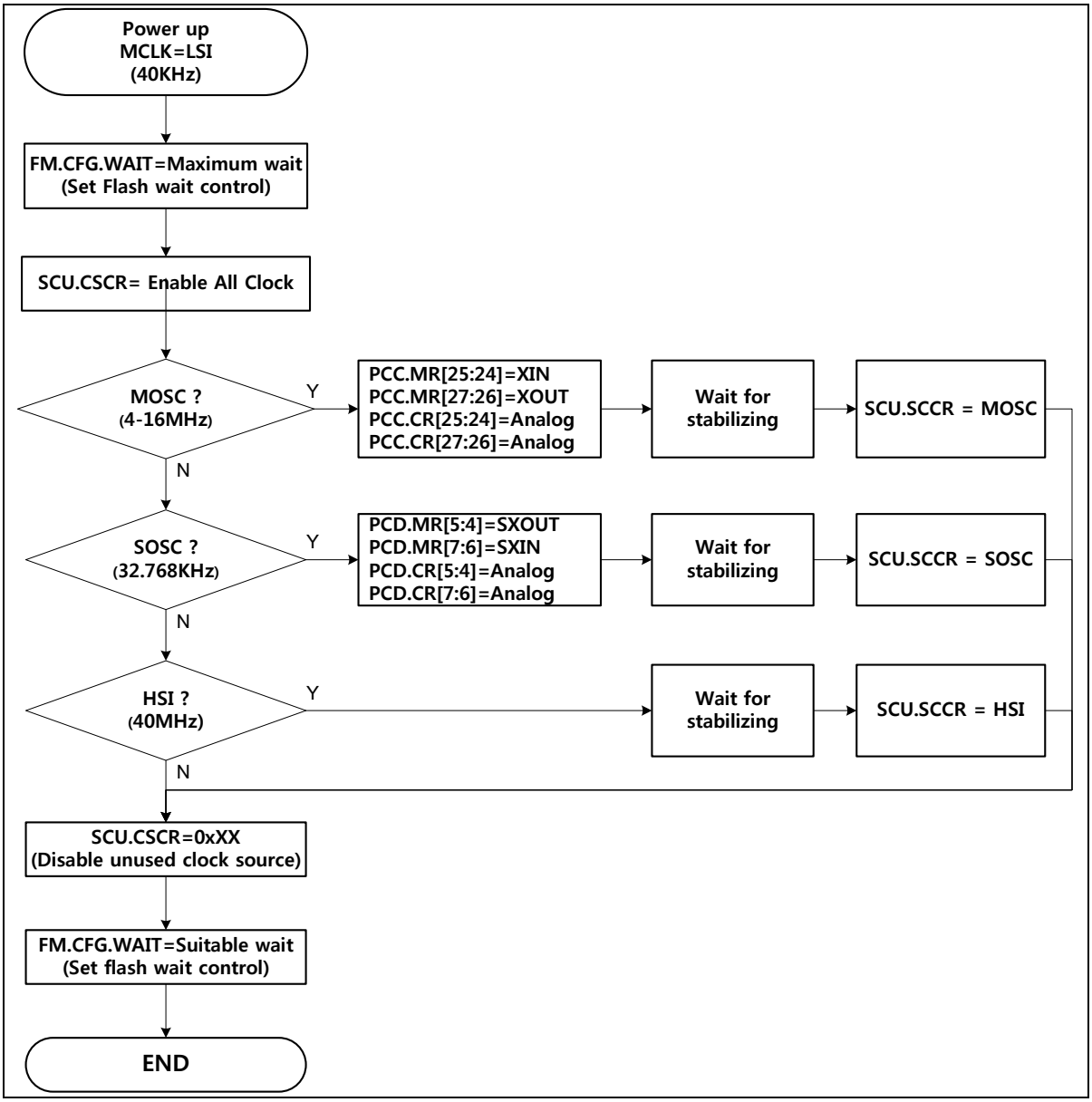


Figure 4-3 Clock Change Procedure

When you speed the system clock up to maximum operating frequency, check the configuration of Flash wait control. Flash read access time is a limiting factor for performance. The wait control recommendation is provided in Table 4-2.

Table 4-2 Flash Wait Control Recommendation

FM.CFG.WAIT	FLASH Access Wait	Available Max System Clock Frequency
00	0 clock wait	~20MHz
01	1 clock wait	~40MHz
10	2 clock wait	~40MHz

Reset

The Z32F0642 MCU has two system resets:

- Cold reset by POR, which is effective during power up or down sequence
- Warm reset, which is generated by several reset sources. The reset events cause the chip to turn on initial state.

The cold reset has only one reset source, which is POR. The warm reset has the following reset sources:

- nRESET pin
- WDT reset
- LVD reset
- MCLK Fail reset
- MOSC Fail reset
- S/W reset
- CPU request reset
- CPU Lockup reset

Cold Reset

Cold reset is an important feature of the chip when power is up. This characteristic affects the system boot globally. Internal VDC is enabled when VDDEXT power is turned on. The internal POR trigger level is 1.4 V of VDDEXT voltage out level, at which time the boot operation is started. The LSI clock is enabled and counts 4.25 msec for internal VDC level stabilizing. During this time, VDDEXT voltage level should be greater than the initial LVD level (1.65 V). After counting 4.25 mse, the cold reset is released and counts 0.4 msec for warm reset synchronizing. BOOTROM and CPU run after releasing cold and warm reset.

Figure 4-4 shows the power up sequence and internal reset waveform.

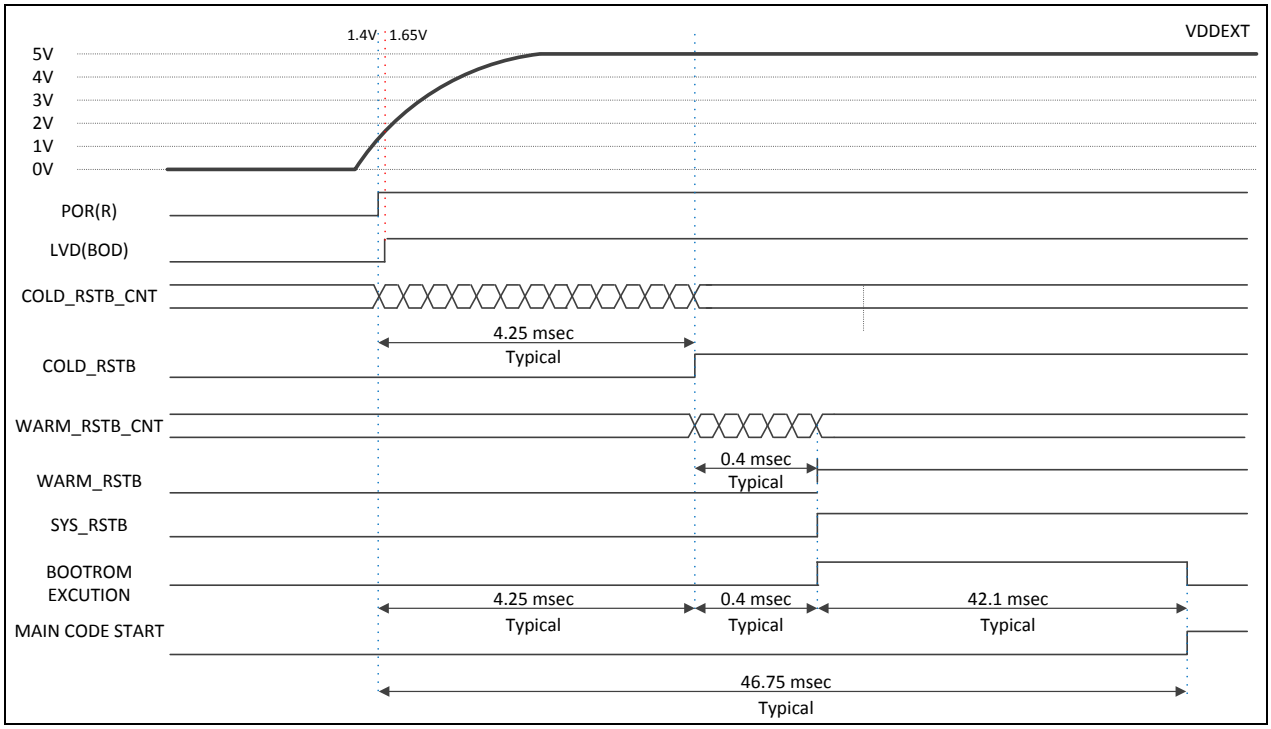


Figure 4-4 Power-up Procedure

Warm Reset

The warm reset event has several reset sources and some parts of the chip return to initial state when the warm reset condition occurs.

The warm reset source is controlled by the SCU.RSER register and the status appears in the SCU.RSSR register. The reset for each peripheral block is controlled by the SCU.PRER register. The reset can be masked independently.

Figure 4-5 shows a diagram of the Warm Reset.

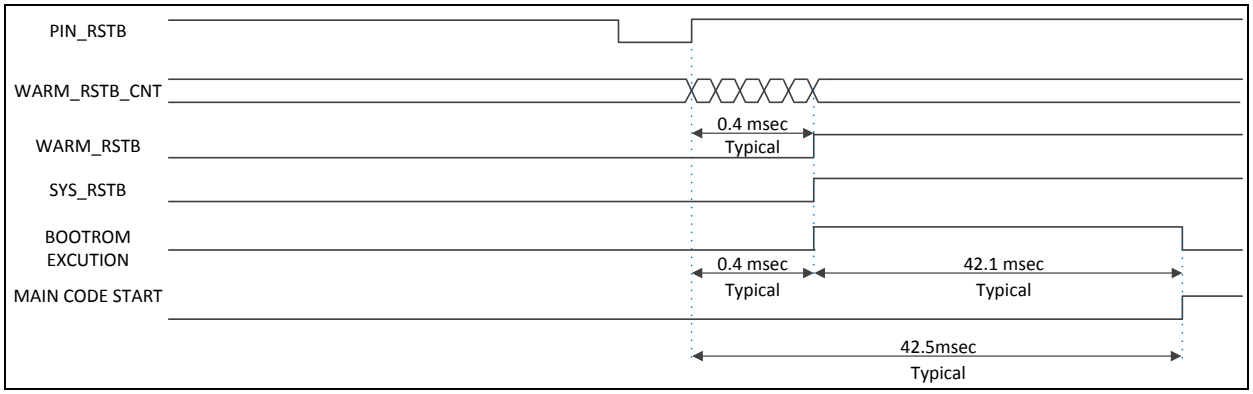


Figure 4-5 Warm Reset Diagram

Low Voltage Reset

A low voltage reset event occurs when the voltage drops below a certain level during operation. When an event occurs, you can select a reset or interrupt action. If a reset occurs, it will be reset to the warm reset state. For more information, refer to the Warm Reset section. Figure 4-6 shows a diagram of Low Voltage Reset.

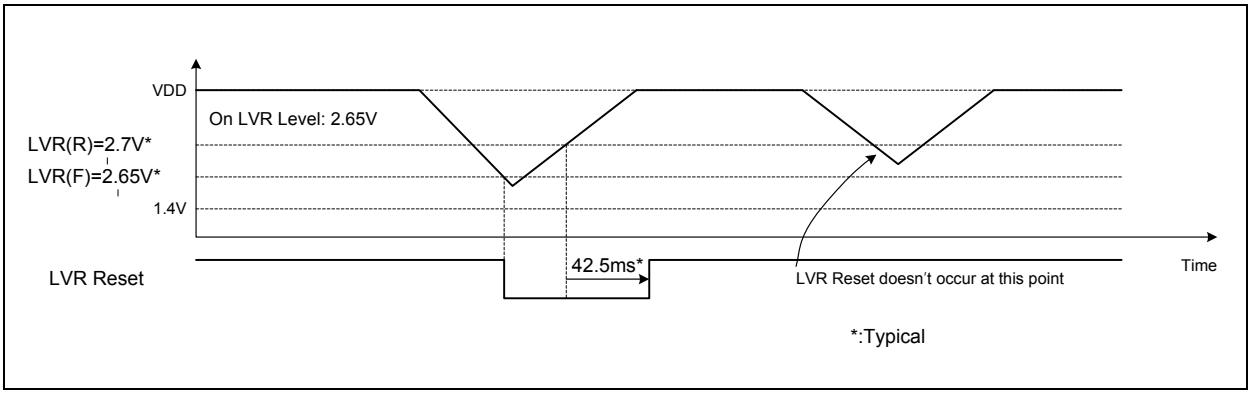


Figure 4-6 Low Voltage Reset Diagram

Reset Tree

Figure 4-7 shows the Reset Tree configuration.

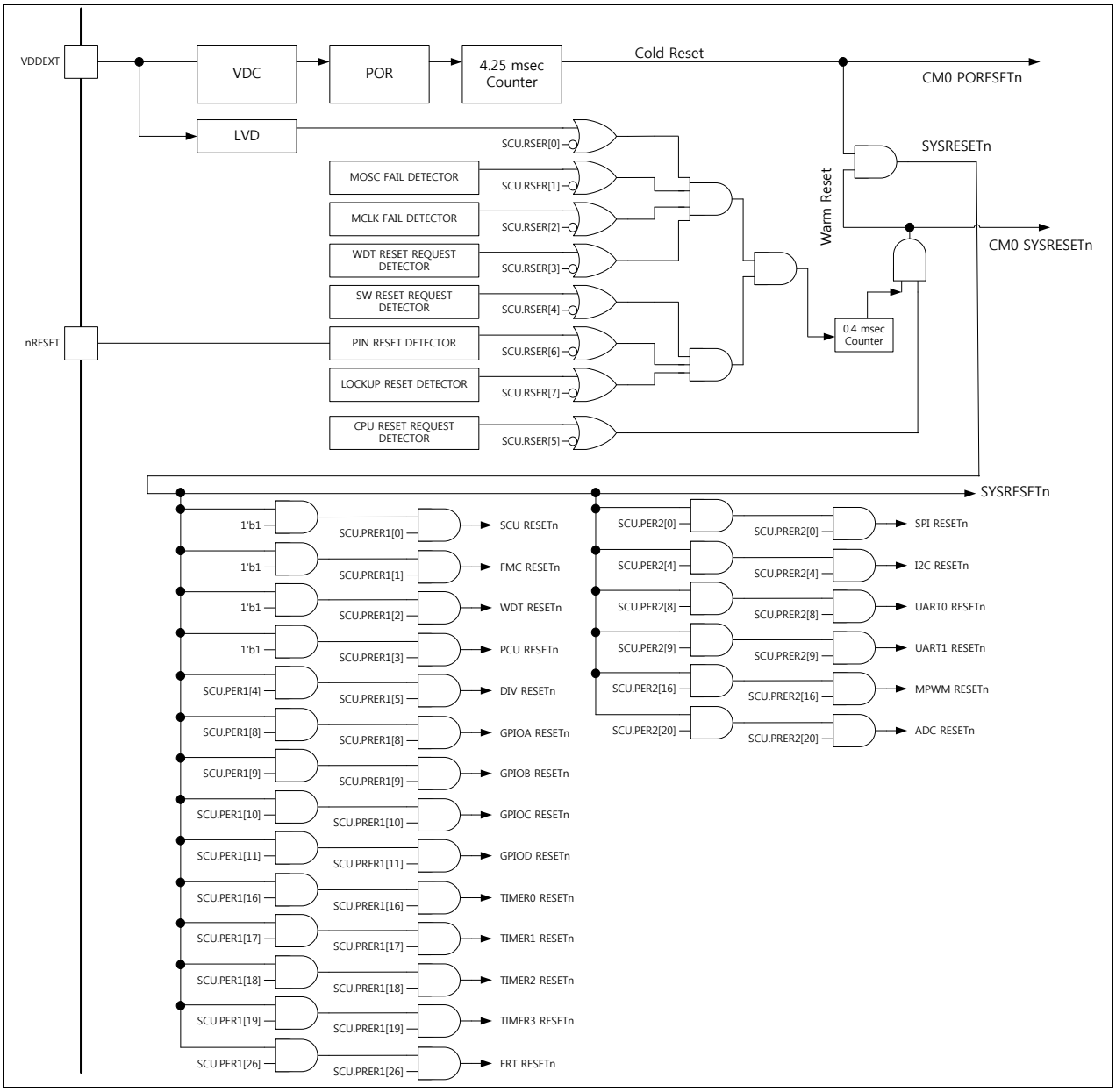


Figure 4-7 Reset Tree Configuration

Operation Mode

The INIT mode is the initial state of the chip when reset is asserted. The Run mode is maximum performance of the CPU with a high-speed clock system. The Sleep and the Power Down modes can be used as low power consumption modes. Low power consumption is achieved by halting the processor core and unused peripherals.

Figure 4-8 shows the Operation mode transition diagram.

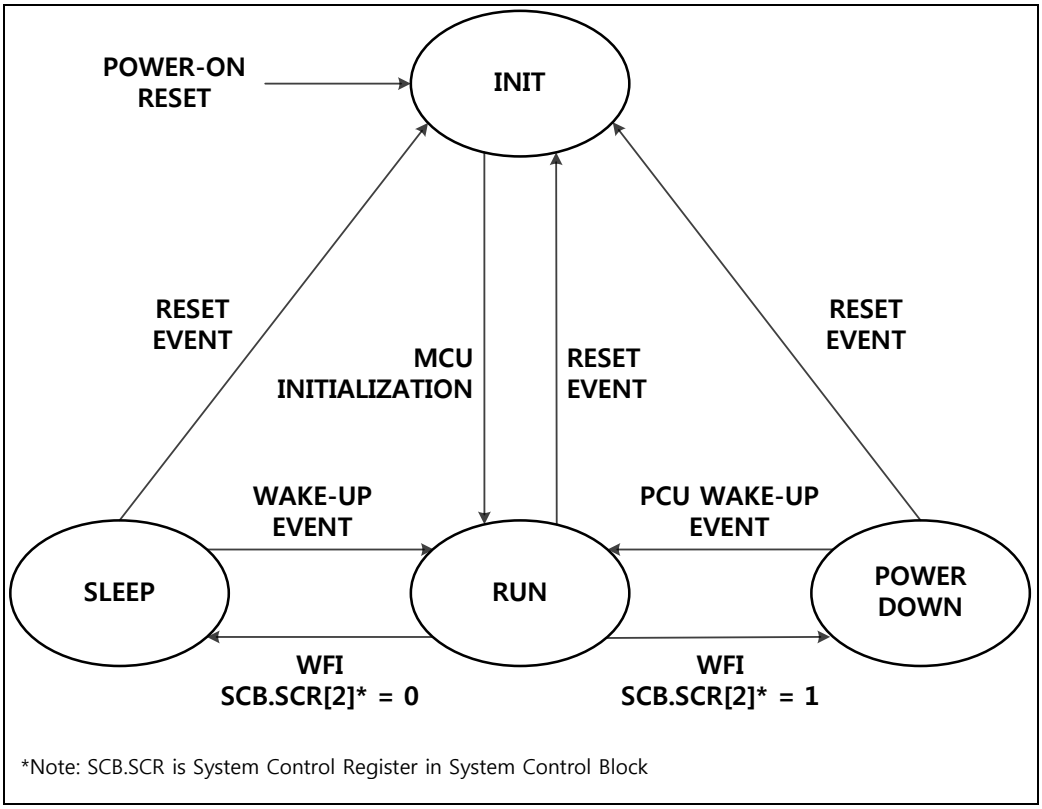


Figure 4-8 Operation Mode Block Diagram

Run Mode

In Run mode, the CPU and the peripheral hardware are operated by using the high-speed clock. Run mode is entered after reset followed by INIT state.

Sleep Mode

Only the CPU is stopped in Sleep mode. Each peripheral function can be enabled by the function enable and clock enable bit in the PER and PCER registers. Figure 4-9 shows the Sleep mode sequence.

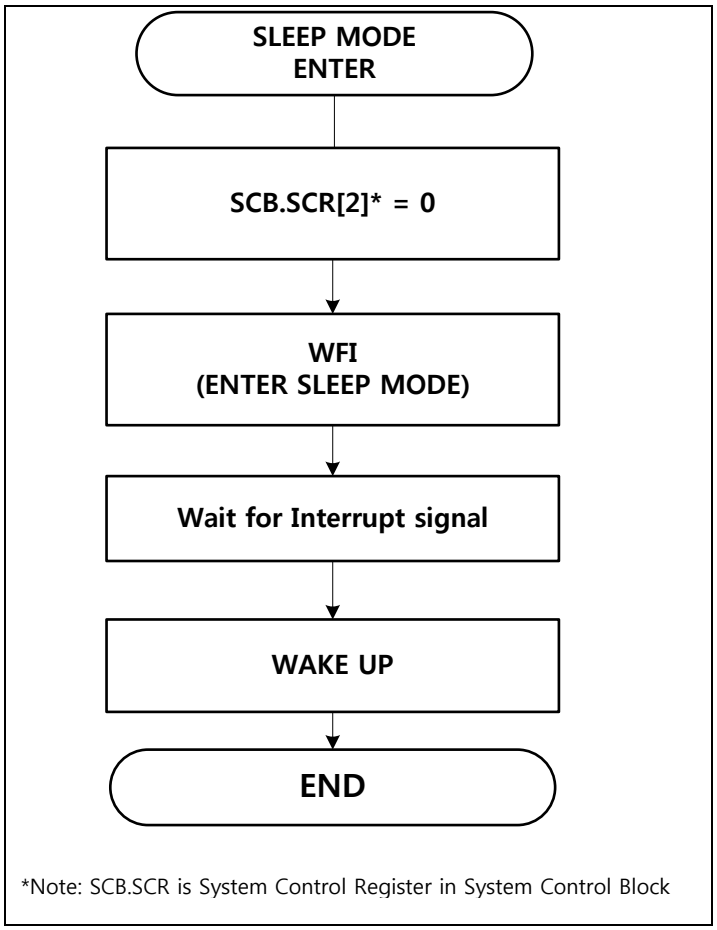


Figure 4-9 Sleep Mode Sequence

Power Down Mode

In Power Down mode, all internal circuits enter the Stop state. The power down operation includes a special power off sequence, as shown in Figure 4-10.

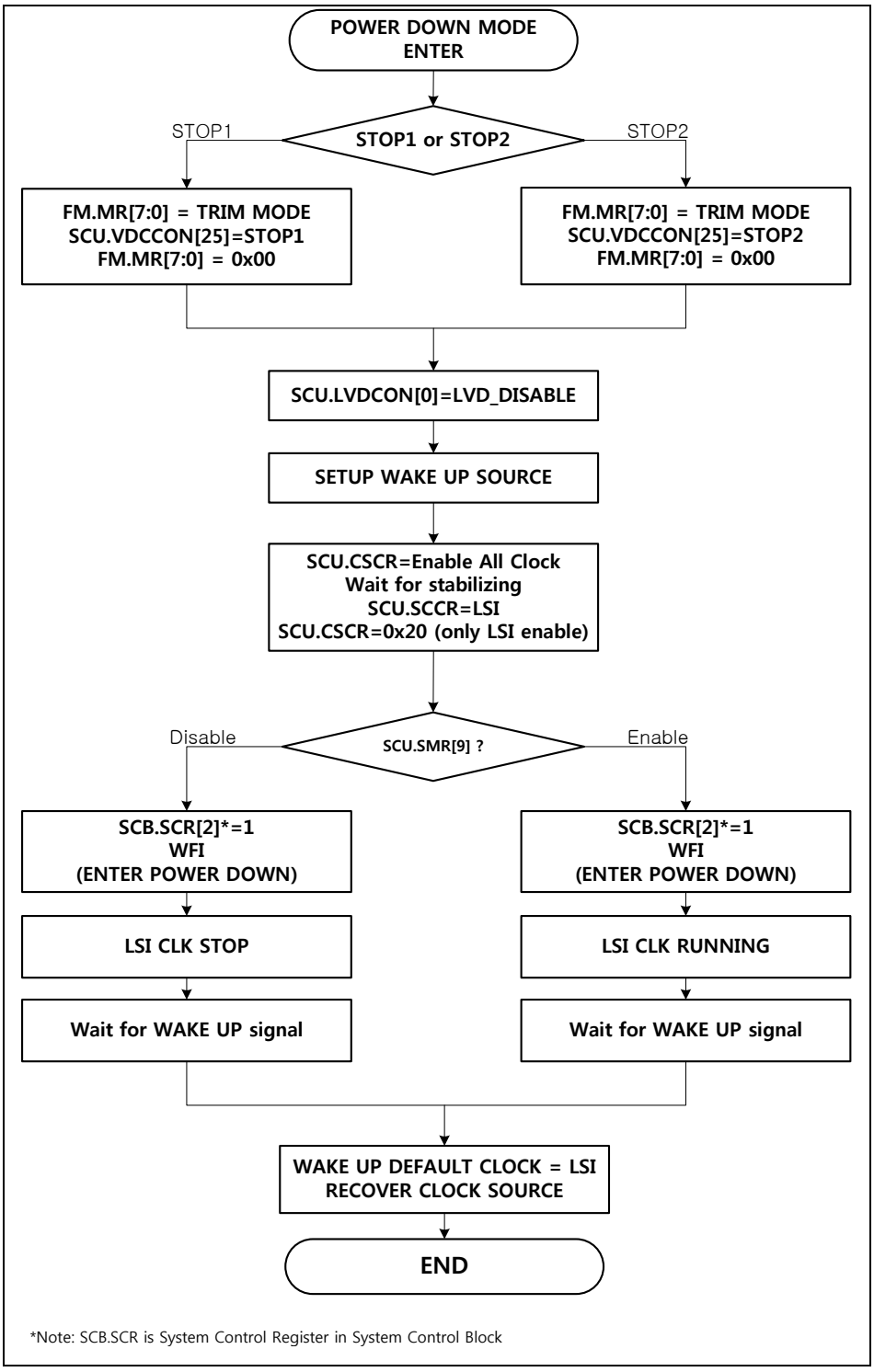


Figure 4-10 Power Down Mode Sequence

Pin Description

Table 4-3 SCU Pins

PIN NAME	TYPE	DESCRIPTION
nRESET	I	External Reset Input
XIN/XOUT	OSC	External Crystal Oscillator
SXIN/SXOUT	OSC	External sub-Crystal Oscillator
STBO	O	Stand-by Output Signal
CLKO	O	Clock Output Monitoring Signal

Registers

The base address of SCU is 0x4000_0000 and the register map is described in Table 4-5.

Table 4-4 Base Address of SCU

NAME	BASE ADDRESS
SCU	0x4000_0000

Table 4-5 SCU Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
SMR	0x0004	RW	System Mode Register	0000_0000
SRCR	0x0008	RW	System Reset Control Register	0000_0000
WUER	0x0010	RW	Wake up source enable register	0000_0000
WUSR	0x0014	RO	Wake up source status register	0000_0000
RSER	0x0018	RW	Reset source enable register	0000_0049
RSSR	0x001C	RW	Reset source status register	0000_0080*
PRER1	0x0020	RW	Peripheral reset enable register 1	040F_0F2F*
PRER2	0x0024	RW	Peripheral reset enable register 2	0011_0311*
PER1	0x0028	RW	Peripheral enable register 1	0000_000F*
PER2	0x002C	RW	Peripheral enable register 2	0000_0101*
PCER1	0x0030	RW	Peripheral clock enable register 1	0000_000F*
PCER2	0x0034	RW	Peripheral clock enable register 2	0000_0101*
CSCR	0x0040	RW	Clock Source Control register	0000_0020
SCCR	0x0044	RW	System Clock Control register	0000_0000
CMR	0x0048	RW	Clock Monitoring register	0000_0090
NMIR	0x004C	RW	NMI control register	0000_0000
COR	0x0050	RW	Clock Output Control register	0000_000F
VDCCON	0x0064	WO	VDC Control register	040F_007F
LVDCON	0x0068	RW	LVD Control register	0001_0101
HSIOSCTRIM	0x006C	RW	High Speed Internal OSC Trim Register	0XXX_XXXX
BISCCON	0x0070	RW	Built in self calibration control Register	0000_0000
MOSCR	0x0080	RW	External main Oscillator control register	0000_0301
EMODR	0x0084	RW	External mode pin read register	0000_0000
MCCR1	0x0090	RW	Misc Clock Control register 1	0000_0000
MCCR2	0x0094	RW	Misc Clock Control register 2	0000_0000
MCCR3	0x0098	RW	Misc Clock Control register 3	0000_0001
MCCR4	0X00A8	RW	Misc Clock Control register 4	0001_0000
DBCLK1	0x009C	RW	Debounce Clock Control register 1	0001_0001
DBCLK2	0x00A0	RW	Debounce Clock Control register 2	0001_0001

SMR System Mode Register

The previous operating mode is shown in this register. The previous operating mode is saved in this register after a reset event. There are two controllable bits in Power Down mode – LSI On/Off control and VDC On/Off control.

The System Mode register is a 16-bit register.

SMR=0x4000_0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						LSIAON	VDCAON			PREVMODE					
0	0	0	0	0	0	0	0	0	0	00		0	0	0	0
						RW	RW			RO					

9	LSIAON	LSI Always on select bit in power down mode	
		0	LSI is turned off when entering power down mode
		1	LSI runs when in power down mode
8	VDCAON	VDC Always on select bit in power down mode	
		0	VDC is turned off entering power down mode
		1	VDC runs when in power down mode
5	PREVMODE	Previous operating mode before current reset event	
4		00	Previous operating mode was RUN mode
		01	Previous operating mode was SLEEP mode
		10	Previous operating mode was Power Down mode
		11	Previous operating mode was INIT mode

SRCR System Reset Control Register

It is possible to check if the chip is in Power Down mode. To use the STBO output function, it should be set as STBO that has output mode in Pin Mux. It is possible to reset the MCU as SWRST bit set.

The System Reset Control register is an 8-bit register.

SCR=0x4000_0008

7	6	5	4	3	2	1	0
			STBOP				SWRST
0	0	0	0	0	0	0	0
			RW				RW

5	STBOP	STBO pin output polarity select bit
	0	Output Low when chip is in Power Down Output High when chip is in normal
	1	Output High when chip is in Power Down Output Low when chip is in normal
1	SWRST	Internal soft reset activation bit (check RSER[4] for reset)
	0	Normal operation
	1	Internal soft reset generated and auto cleared

WUER Wakeup Source Enable Register

Enable the wakeup source when the chip is in Power Down mode. Wakeup sources that are used as the source of chip wakeup should be enabled in each bit field. If the source is used as a wakeup source, the corresponding bit should be written as ‘1’. If the source is not used as a wakeup source, the bit should be written as ‘0’.

This register is a 16-bit register.

WUER=-0x4000_0010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				GPIODWUE	GPIOCWUE	GPIOBWUE	GPIOAWUE						FRTWUE	WDTWUE	LVDWUE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				RW	RW	RW	RW						RW	RW	RW

11	GPIODWUE	Enable wakeup source of GPIOD port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
10	GPIOCWUE	Enable wakeup source of GPIOC port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
9	GPIOBWUE	Enable wakeup source of GPIOB port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
8	GPIOAWUE	Enable wakeup source of GPIOA port pin change event
	0	Not used for wakeup source
	1	Enable the wakeup event generation
2	FRTWUE	Enable wakeup source of FRT event
	0	Not used for wakeup source

		1	Enable the wakeup event generation
1	WDTWUE	Enable wakeup source of WDT event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation
0	LVDWUE	Enable wakeup source of LVD event	
		0	Not used for wakeup source
		1	Enable the wakeup event generation

WUSR Wakeup Source Status Register

When the system is woken up by a wakeup source, the wakeup source is identified by reading this register. When the bit is set to 1, the related wakeup source issues the wakeup to the SCU. **The bit is cleared when the event source is cleared by the software.**

WUSR=0x4000_0014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				GPIODWU	GPIOCWU	GPIOBWU	GPIOAWU						FRTWU	WDTWU	LVDWU
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				RO	RO	RO	RO						RO	RO	RO

11	GPIODWU	Status of wakeup source of GPIOD port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
10	GPIOCWU	Status of wakeup source of GPIOC port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
9	GPIOBWU	Status of wakeup source of GPIOB port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
8	GPIOAWU	Status of wakeup source of GPIOA port pin change event	
		0	No wakeup event
		1	Wakeup event was generated
2	FRTWU	Status of wakeup source of FRT event	
		0	No wakeup event
		1	Wakeup event was generated
1	WDTWU	Status of wakeup source of WDT event	
		0	No wakeup event
		1	Wakeup event was generated
0	LVDWU	Status of wakeup source of LVD event	
		0	No wakeup event
		1	Wakeup event was generated

RSER Reset Source Enable Register

The reset source to the CPU can be selected using the RSER register. When writing ‘1’ in the bit field of each reset source, the reset source event is transferred to the reset generator. When writing ‘0’ in the bit field of each reset source, the reset source event is masked and does not generate the reset event.

RSER=0x4000_0018

7	6	5	4	3	2	1	0
LOCKUPRST	PINRST	CPURST	SWRST	WDTRST	MCKFRST	MOFRST	LVDRST
0	1	1	0	1	0	0	1
RW	RW	RW	RW	RW	RW	RW	RW

7	LOCKUPRST	CPU Lock up reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
6	PINRST	External pin reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
5	CPURST	CPU request reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
4	SWRST	Software reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
3	WDTRST	Watchdog Timer reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
2	MCKFRST	MCLK Clock fail reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
1	MOFRST	MOSC Clock fail reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
0	LVDRST	LVD reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled

RSSR Reset Source Status Register

The Reset Source Status register shows the reset source information when a reset event occurs. ‘1’ indicates that a reset event exists and ‘0’ indicates that a reset event does not exist for a given reset source.

When the reset source is found, writing ‘1’ to the corresponding bit clears the reset status. This register is an 8-bit register.

RSSR=0x4000_001C

8	7	6	5	4	3	2	1	0
LOCKUPRST	PORST	PINRST	CPURST	SWRST	WDTRST	MCKFRST	MOFRST	LVDRST
0	1	0	0	0	0	0	0	0
RC1	RC1	RC1	RC1	RC1	RC1	RC1	RC1	RC1

7	LOCKUPRST	CPU Lock up reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
7	PORST	Power on reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
6	PINRST	External pin reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
5	CPURST	CPU request reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
4	SWRST	Software reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
3	WDTRST	Watchdog Timer reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
2	MCLKFRST	MCLK fail reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
1	MOFRST	MOSC Clock fail reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status
0	LVDRST	LVD reset status bit
	0	Read : Reset from this event did not exist Write : no effect
	1	Read :Reset from this event occurred Write : Clear the status

PRER1 Peripheral Reset Enable Register 1

The reset of each peripheral by an event reset can be masked with the help of user settings. The PRER1/PRER2 register controls enabling of the event reset. If the corresponding bit is '1', the peripheral corresponding to this bit accepts the reset event. Otherwise, the peripheral is protected from the reset event and maintains its current operation.

PRER1=0x4000_0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					FRT							TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA			DIV64		PCU	WDT	FMC	SCU
0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	0	1	1	1	1
					RW							RW	RW	RW	RW					RW	RW	RW	RW			RW		RW	RW	RW	RW

26	FRT	FRT reset enable
19	TIMER3	TIMER3 reset enable
18	TIMER2	TIMER2 reset enable
17	TIMER1	TIMER1 reset enable
16	TIMER0	TIMER0 reset enable
11	GPIOD	GPIOD reset enable
10	GPIOC	GPIOC reset enable
9	GPIOB	GPIOB reset enable
8	GPIOA	GPIOA reset enable
5	DIV64	DIV64 reset enable
3	PCU	Port Control Unit reset enable
2	WDT	Watchdog Timer reset enable
1	FMC	Flash memory controller reset enable
0	SCU	System Control Unit reset enable

PRER2 Peripheral Reset Enable Register 2

Peripheral Reset Enable Register 2 is a 32-bit register.

PRER2=0x4000_0024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ADC				MPWM							UART1	UART0				I2C				SPI
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1
											RW				RW							RW	RW				RW				RW

20	ADC	ADC reset enable
16	MPWM0	MPWM reset enable
9	UART1	UART1 reset enable
8	UART0	UART0 reset enable
4	I2C	I^2C reset enable
0	SPI	SPI reset enable

PER1 Peripheral Enable Register 1

To use a peripheral unit, it should be activated by writing ‘1’ to the corresponding bit in the PER1/PER2 register. Prior to activation, the peripheral stays in reset state.

All the peripherals are enabled by default. To disable the peripheral unit, write ‘0’ to the corresponding bit in the PER1/PER2 register, after which the peripheral enters the reset state.

PER1=0x4000_0028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					FRT							TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA			DIV64		Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
					RW							RW	RW	RW	RW					RW	RW	RW	RW			RW		RO	RO	RO	RO

26	FRT	FRT function enable
19	TIMER3	TIMER3 function enable
18	TIMER2	TIMER2 function enable
17	TIMER1	TIMER1 function enable
16	TIMER0	TIMER0 function enable
11	GPIOD	GPIOD function enable
10	GPIOC	GPIOC function enable
9	GPIOB	GPIOB function enable
8	GPIOA	GPIOA function enable
5	DIV64	DIV64 function enable
3		Reserved
2		
1		
0		

PER2 Peripheral Enable Register 2

Peripheral Enable Register 2 is a 32-bit register.

PER2=0x4000_002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ADC				MPWM							UART1	UART0				I2C				SPI
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
											RW				RW							RW	RW				RW				RW

20	ADC	ADC function enable
16	MPWM	MPWM function enable
9	UART1	UART1 function enable
8	UART0	UART0 function enable
4	I2C	I^2C function enable
0	SPI	SPI function enable

PCER1 Peripheral Clock Enable Register 1

To use a peripheral unit, its clock should be activated by writing '1' to the corresponding bit in the PCER1/PCER2 register. The peripheral does not operate accurately if its clock is not enabled.

To stop the clock of the peripheral unit, write '0' to the corresponding bit in the PCER1/PCER2 register.

PCER1=0x4000_0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					FRT							TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA			DIV64		Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
					RW							RW	RW	RW	RW					RW	RW	RW	RW			RW	RW	RO	RO	RO	RO

26	FRT	FRT clock enable
19	TIMER3	TIMER3 clock enable
18	TIMER2	TIMER2 clock enable
17	TIMER1	TIMER1 clock enable
16	TIMER0	TIMER0 clock enable
11	GPIOD	GPIOD clock enable
10	GPIOC	GPIOC clock enable
9	GPIOB	GPIOB clock enable
8	GPIOA	GPIOA clock enable
5	DIV64	DIV64 clock enable
3		Reserved
2		
1		
0		

PCER2 Peripheral Clock Enable Register 2

To use a peripheral unit, its clock should be activated by writing '1' to the corresponding bit in the PCER1/PCER2 register.

PCER2=0x4000_0034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											ADC				MWPM							UART1	UART0				I2C				SPI
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
											RW				RW							RW	RW				RW				RW

20	ADC	ADC clock enable
16	MPWM	MPWM clock enable
9	UART1	UART1 clock enable
8	UART0	UART0 clock enable
4	I2C	I^2C clock enable
0	SPI	SPI clock enable

CSCR Clock Source Control Register

The Z32F0642 MCU has multiple clock sources to generate internal operating clocks. Each clock source can be controlled by the CSCR register. This register is an 8-bit register.

CSCR=0x4000_0040

7 6	5 4	3 2	1 0
SOSCCON	LSICON	HSICON	MOSCCON
00	10	00	00
RW	RW	RW	RW

7	SOSCCON	External crystal sub oscillator control	
6		0X	Disable external sub crystal oscillator
		10	Enable external sub crystal oscillator
		11	Enable external sub crystal oscillator divide by 2
5	LSICON	Low speed internal oscillator control	
4		0X	Disable low speed internal oscillator
		10	Enable low speed internal oscillator
		11	Enable low speed internal oscillator divide by 2
3	HSICON	High speed internal oscillator control	
2		0X	Disable high speed internal oscillator
		10	Enable high speed internal oscillator
		11	Enable high speed internal oscillator divide by 2
1	MOSCCON	External crystal main oscillator control	
0		0X	Disable external main crystal oscillator
		10	Enable external main crystal oscillator
		11	Enable external main crystal oscillator divide by 2

SCCR System Clock Control Register

Select the system clock source in SCCR and the selected clock source becomes MCLK. Before changing the clock, clock sources have to be enabled in the CSCR register and oscillating.

SCCR=0x4000_0044

7	6	5	4	3	2	1	0
					MCLKSEL		
					00		
					RW		

2	MCLKSEL	System clock select register	
0		000	LSI (40kHz)
		001	SOSC (32.768kHz)
		100	HSI (40MHz)
		110	MOSC (4MHz ~ 16MHz)

Note: When changing MCLKSEL, both clock sources should be enabled and stable.
For example, both HSI and MOSC should be enabled and stable, otherwise the chip will malfunction.

CMR Clock Monitoring Register

The clock can be monitored by LSI for security purposes. The Clock Monitoring register is a 16-bit register.

CMR=0x4000_0048

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCLKREC				SOSCMNT	SOSCIE	SOSCFAIL	SOSCSTS	MCLKMNT	MCLKIE	MCLKFAIL	MCLKSTS	MOSCMNT	MOSCIE	MOSCFAIL	MOSCSTS
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW				RW	RW	RC1	RC1	RW	RW	RC1	RC1	RW	RW	RC1	RC1

15	MCLKREC	MCLK fail auto recovery
		0 MCLK is changed to LSI by default when MCLKFAIL issued
		1 MCLK auto recovery is disabled
11	SOSCMNT	External sub oscillator monitoring enable
		0 External sub oscillator monitoring disabled
		1 External sub oscillator monitoring enabled
10	SOSCIE	External sub oscillator fail interrupt enable
		0 External sub oscillator fail interrupt disabled
		1 External sub oscillator fail interrupt enabled
9	SOSCFAIL	External sub oscillator fail interrupt
		0 External sub oscillator fail interrupt not occurred
		1 Read : External sub oscillator fail interrupt is pending Write : Clear pending interrupt
8	SOSCSTS	External sub oscillator status
		0 Not oscillate
		1 External sub oscillator is working normally
7	MCLKMNT	MCLK monitoring enable
		0 MCLK monitoring disabled
		1 MCLK monitoring enabled
6	MCLKIE	MCLK fail interrupt enable
		0 MCLK fail interrupt disabled
		1 MCLK fail interrupt enabled
5	MCLKFAIL	MCLK fail interrupt
		0 MCLK fail interrupt not occurred
		1 Read : MCLK fail interrupt is pending Write : Clear pending interrupt
4	MCLKSTS	MCLK clock status
		0 No clock is present on MCLK
		1 Clock is present on MCLK
3	MOSCMNT	External main oscillator monitoring enable
		0 External main oscillator monitoring disabled
		1 External main oscillator monitoring enabled
2	MOSCIE	External main oscillator fail interrupt enable
		0 External main oscillator fail interrupt disabled
		1 External main oscillator fail interrupt enabled
1	MOSCFAIL	External main oscillator fail interrupt
		0 External main oscillator fail interrupt not occurred
		1 Read : External main oscillator fail interrupt is pending Write : Clear pending interrupt
0	MOSCSTS	External main oscillator status
		0 Not oscillate
		1 External main oscillator is working normally

NMIR NMI Control Register

The NMI Control register s the non-maskable interrupt configuration register which can be set by software. There are five sources for the Non-maskable Interrupt events. This register provides the ability to enable and check the status of the source of the interrupt.

Write access key is required 0xA32C on NMIR [31:16] when writing to this register.

NMIR=0x4000_004C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESSCODE																			PROTSTS	OVPSTS	WDTINTSTS	MCLKFAILSTS	LVDSTS				PROTEN	OVPEN	WDTINTEN	MCLKFAILEN	LVDEN
-																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WO																			RO	RO	RO	RO	RO				RW	RW	RW	RW	RW

31 16	ACCESSCODE	This field enables writing access to this register. Writing 0xA32C is to enable writing.
12	PROTSTS	Protection condition status bit. This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
11	OVPSTS	Over Voltage Protection condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
10	WDTINTSTS	WDT Interrupt condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
9	MCLKFAILSTS	MCLK Fail condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
8	LVDSTS	LVD condition status bit This bit can't invoke NMI interrupt without enable bit
		0 Not occurred
		1 Event occurred
4	PROTEN	Protection condition enable for NMI interrupt
		0 Disable
		1 Enable
3	OVPEN	Over Voltage Protection condition enable for NMI interrupt
		0 Disable
		1 Enable
2	WDTINTEN	WDT Interrupt condition enable for NMI interrupt
		0 Disable
		1 Enable
1	MCLKFAILEN	MCLK Fail condition enable for NMI interrupt
		0 Disable
		1 Enable
0	LVDEN	LVD Fail condition enable for NMI interrupt
		0 Disable
		1 Enable

COR Clock Output Register

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. To use the CLKO output function, it should be set as CLKO that has output mode in Pin Mux. The Clock Output register is an 8-bit register.

COR=0x4000_0050

7	6	5	4	3	2	1	0
-			CLKOEN	CLKODIV			
000			0	1111			
RO			RW	RW			

4	CLKOEN	Clock output enable
		0 CLKO is disabled and stay “L” output
		1 CLKO Is enabled
3 0	CLKODIV	Clock output divider value CLKO = MCLK (CLKODIV = 0) $\text{CLKO} = \frac{MCLK}{2 * (\text{CLKODIV} + 1)} \quad (CLKODIV > 0)$

VDCCON VDC Control Register

The on-chip VDC Control register selects Stop mode operation for VDC and warm up count delay. The STOPSEL bit can be written when writing '1' to the VDCME bit simultaneously. The VDCWDLY value can be written by writing '1' to the VDCDE bit simultaneously. To change the VDCCON register value, it has to enter TRIM mode.

VDCCON=0x4000_0064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDCME	Reserved					STOPSEL	Reserved	Reserved															VDCDE					VDCWDLY			
0						0																	0					0x04			
WO						WO																	WO					WO			

31	VDCME	VDCMODE value write enable. Write only with VDCMODE value.
		0 VDCMODE field is not updated by writing
		1 VDCMODE filed can be updated by writing
25	STOPSEL	STOP MODE Select bit.
		0 VDC STOP MODE 1
		1 VDC STOP MODE 2
8	VDCDE	VDCWDLY value write enable. Write only with VDCWDLY value
		0 VDCWDLY Write disable
		1 VDCWDLY Write Enable
3 0	VDCWDLY	VDC warm-up delay count value. When SCU is woken up from power down mode, the warm-up delay is inserted for VDC output being stabilized. The amount of delay can be defined with this register value 4 : 2msec

CAUTION! You must not set the reserved bit fields.

Note: To enter TRIM mode to change the VDCCON value:

```
FM->MR=0xa5;
FM->MR=0x5a;        // TRIM mode enter
SCU->VDCCON = (1UL<<31) | (1UL<<25); // set VDC STOP MODE 2
FM->MR=0; // TRIM mode exit
```

LVDCON LVD Control Register

The on-chip Low Voltage Detector Control register is a 32-bit register.

LVDCON=0x4000_0068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SELEN						LVDSEL								LVDLVL	LVDEN
																0						00								0	1
																WO						RW								RO	RW

15	SELEN	LVD level SEL value write enable. Write only.	
		0	LVDSEL field is not updated by writing
		1	LVDSEL filed can be updated by writing
9	LVDSEL	LVD detect level select	
8		00	LVD detect level is 1.73V
		01	LVD detect level is 2.65V
		10	LVD detect level is 3.70V
		11	Reserved
1	LVDLVL	LVD Status	
		0	VDDEXT level is over than LVD level
		1	VDDEXT level is under than LVD level
0	LVDEN	LVD Function enable	
		0	LVD is not enabled
		1	LVD is enabled

HSIOSCTRIM High Speed Internal OSC Trim Register

The High Speed Internal Oscillator Trim register for enabling/disabling self-calibration is a 32-bit register.

HSIOSCTRIM=0x4000_006C

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

31	BISCON	Build in self calibration function enable.	
		0	BISC function disabled. IOSC supplies factory calibrated frequency.
		1	BISC function enabled. IOSC supplies self-calibrated frequency
30	REFSEL	Reference clock select for self-calibration	
		0	Main oscillator clock source is reference clock
		1	Sub oscillator clock source is reference clock

CAUTION! You must not set the reserved bit field.

Note: All trim bits are writable when trim mode is enabled

```
FM->MR=0xa5;
FM->MR=0x5a;        // TRIM mode enter
…                   // change HSIOSCTRIM value
FM->MR=0;           // TRIM mode exit
```

BISCCON Built-in Self Calibration Control Register

This register provides the comparison counts between the internal oscillator and the external oscillator for self calibration. The calculation for the value is:

INTOSC_COMP = (updateperiod / 1/desired clock frequency) - 1
XTAL_COMP = (updateperiod/1/XTAL frequency) - 1

In the above equations, *updateperiod* is the number of clocks of the internal oscillator to compare with XTAL clocks. Depending on the speed, this value is typically around 10 uS.

This register is a 32-bit register.

BISCCON=0x4000_0070

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

31 16	INTOSC_COMP[31:16]	INTOSC compare value
15 0	XTAL_COMP[15:0]	XTAL Compare value

Calibration supports the configurations in Table 4-6.

Table 4-6 BISC Count Value

XTAL FREQ	TARGET FREQ	UPDATE PERIOOD	XTAL_COMP	INTOSC_COMP
MHz	MHz	Nano Sec	Count Value	Count Value
10	40	10,000	99	399
8	40	1,000,000	7999	39999
6	40	10,000	59	399

EMOSCR External Main Oscillator Control Register

External main crystal oscillator has two characteristics. For the noise immunity, NMOS amp type is recommended and for the low power characteristic, INV amp type is recommended. This register is a 16-bit register.

EMOSCR=0x4000_0080

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FILSKIPWEN							FILSKIPEN	INVCLKWEN							INVCLKEN
0							0	0							0
WO							RW	WO							RW

15	FILSKIPWEN	Write enable of bit field FILSKIPEN.
		0 Write access of FILSKIPEN field is masked
		1 Write access of FILSKIPEN field is accepted
8	FILSKIPEN	Control External Main Oscillator Filter Skip bit
		0 External Main Oscillator Filter Skip Disable.
		1 External Main Oscillator Filter Skip Enable.
7	INVCLKWEN	Write enable of bit field FILSKIPEN.
		0 Write access of INVCLKEN field is masked
		1 Write access of INVCLKEN field is accepted
0	INVCLKEN	Control External Main Oscillator CLK Invert bit
		0 External Main Oscillator CLK Invert Disable.
		1 External Main Oscillator CLK Invert Enable.

EMODR External Mode Status Register

The External Mode Status register shows the external mode pin status while booting. This register is an 8-bit register.

EMODR=0x4000_0084

7	6	5	4	3	2	1	0
					Reserved	Reserved	BOOT
		0x0			-	-	-
		RO			-	-	RO

0	BOOT	BOOT pin level
		0 BOOT(PC11) pin is low
		1 BOOT(PC11) pin is high

DBCLK1Debounce Clock Control Register 1

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. PxDCSEL bits and PxDDIV bits of MCCR4 are used as PORT debounce clock sources. This register is a 32-bit register.

MCCR4=0x4000_009C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PBDCSEL			PBDDIV														PADCSEL			PADDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW														RW			RW							

26 24	PBDCSEL	Debounce Clock for Port B source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
23 16	PBDDIV	PORT B Debounce Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing PBDCSEL
10 8	PADCSEL	Debounce Clock for Port A source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
7 0	PADDIV	PORT A Debounce Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing PADCSEL

DBCLK2Debounce Clock Control Register 2

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. PxDCSEL bits and PxDDIV bits of MCCR5 are used as PORT debounce clock sources. This register is a 32-bit register.

MCCR5=0x4000_00A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PDDCSEL			PDDDIV														PCDCSEL			PCDDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW														RW			RW							

26 24	PDDCSEL	Debounce Clock for PORT D source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
23 16	PDDDIV	PORT D Debounce Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing PDDCSEL
10 8	PCDCSEL	Debounce Clock for PORT C source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
7 0	PCDDIV	PORT C Debounce Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing PCDCSEL

MCCR1 Miscellaneous Clock Control Register 1

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. STCSEL bits and STCDIV bits of MCCR1 are used as SYSTICK external clock sources. This register is a 32-bit register.

MCCR1=0x4000_0090

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
Reserved	Reserved	Reserved	STCSEL	STCDIV
-	-	-	000	0x00
-	-	-	RW	RW

10 8	STCSEL	SYSTICK Clock source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 Reserved
7 0	STCDIV	SYSTICK Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing STCSEL.

MCCR2 Miscellaneous Clock Control Register 2

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. PWMCSEL bits and PWMDIV bits of MCCR2 are used as MPWM clock sources. If it is used as MPWM, it must set this register. This register is a 32-bit register.

MCCR2=0x4000_0094

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
Reserved	Reserved		PWMCSEL	PWMDIV
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0	000	0x00
			RW	RW

10 8	PWMCSEL	PWM Clock source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 Reserved
7 0	PWMDIV	PWM Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing PWMCSEL

MCCR3 Miscellaneous Clock Control Register 3

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. TIMERCSEL bits and TIMERDIV bits of MCCR3 are used as TIMER external clock sources. WDTCSEL bits and WDTDIV bits of MCCR3 are used as WDT external clock sources. This register is a 32-bit register.

MCCR3=0x4000_0098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TIMERCSEL			TIMERDIV														WDTCSEL			WDTDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW													RW				RW							

26 24	TIMERCSEL	Timer Clock source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
23 16	TIMERDIV	Timer Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing TIMERCSEL
10 8	WDTCSEL	WDT Clock source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
7 0	WDTDIV	WDT Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing WDTCSEL

MCCR4 Miscellaneous Clock Control Register 4

The Z32F0642 MCU can drive the clock from an internal MCLK clock with a dedicated post divider. ADCCSEL bits and ADCDIV bits of MCCR4 are used as ADC external clock sources. UARTCSEL bits and UARTDIV bits of MCCR4 are used as UART clock sources. If it is used as UART, this register must be set.

MCCR4=0x4000_00A8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ADCCSEL			ADCCDIV														UARTCSEL			UARTCDIV							
0	0	0	0	0	000			0x01													000			0x01							
					RW			RW													RW			RW							

26 24	ADCCSEL	ADC clock source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 Reserved
23 16	ADCCDIV	ADC Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing ADCCSEL
10 8	UARTCSEL	UART clock source select bit 000 LSI 100 MCLK 101 HSI 110 MOSC 111 SOSC
7 0	UARTCDIV	UART Clock N divider 0x00 : disabled 0xN : (selected clock) / N To change the value, set 0x0 first without changing UARTCSEL

Functional Description

Clock Configuration

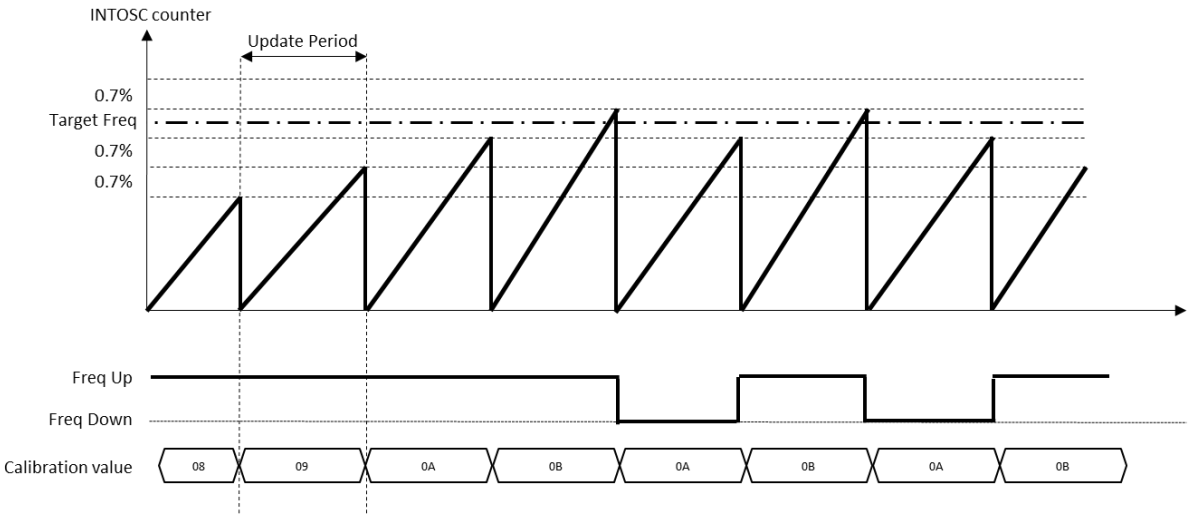
To configure the clock, see Clock Configuration Procedure.

Configure Clock Out for Monitoring Actual Clock Output

Use the following procedure to configure clock out for monitoring actual clock output:

1. Enable the Port C peripheral and clock in the SCU PER1 and PCER1 registers.
2. Unlock the Port Controller using the PORTEN register as defined in Port Control Unit (PCU).
3. Enable the Alternative function 01b for pin 9 on PORT C through the PCC_MR register.
4. Set the Pin type for pin 9 on Port C to output (00b).
5. Lock the Port Controller by writing any value to the PORTEN register.
6. Set bit 4 of the Clock Output Register (COR) register to enable the output.
7. Configure CLKODIV to the desired output divider.

Built-in Self Calibration



The self-calibration block has a 4-fine trim value which is configurable. The calibration value is changed until the frequency of INTOSC crosses the target frequency level. 8 steps up trim and 8 steps down trim are available with a 0.7% difference in each step.

The update period is decided by the reference clock counter value.

When the BISC function is enabled, the factory calibration value is replaced by the self-calibration value. A minimum of 8 times the update period is required before changing the system clock to the INTOSC clock.

5. Port Control Unit (PCU)

Overview

Port Control Unit (PCU) controls the external I/Os in the following manner:

- Sets pin function mux
- Sets external signal directions of each pin
- Sets interrupt trigger mode for each pin
- Sets internal pull-up register control and open drain control

Figure 5-1 shows a block diagram of the PCU.

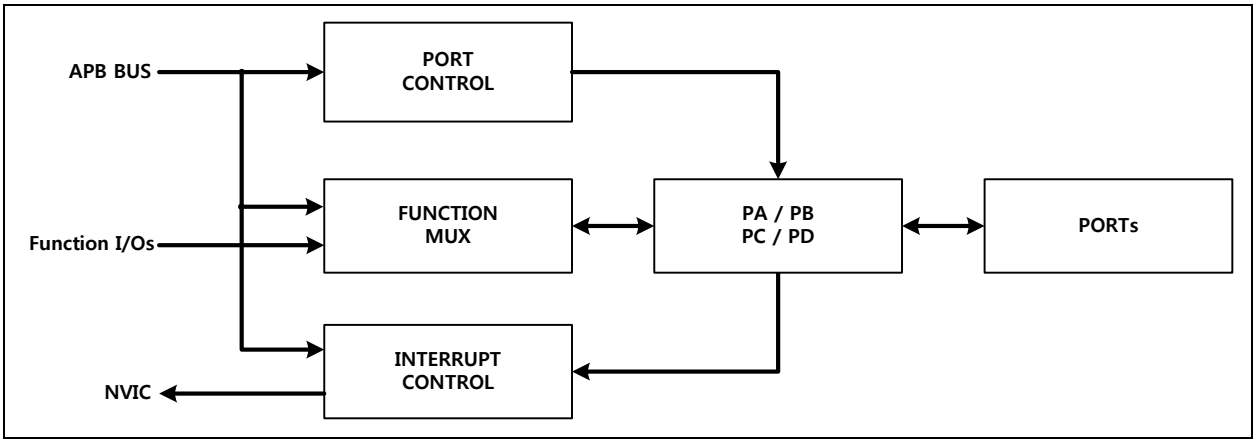


Figure 5-1 Block Diagram

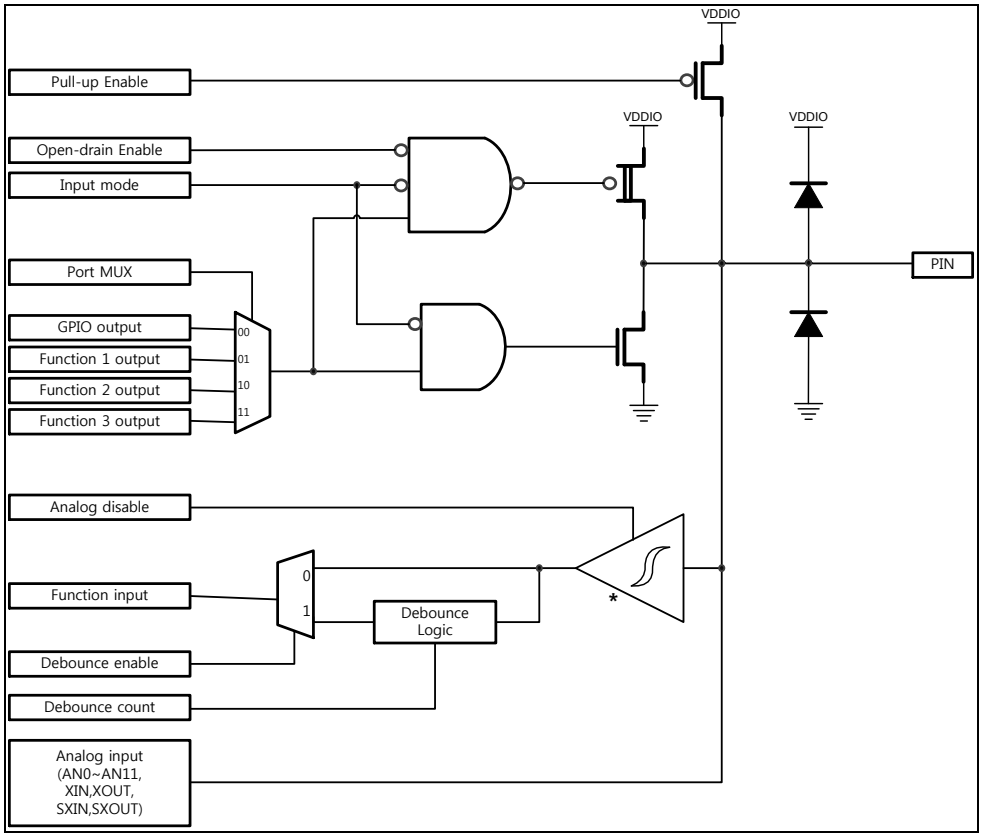


Figure 5-2 I/O Port Block Diagram (ADC and External Oscillator Pins)

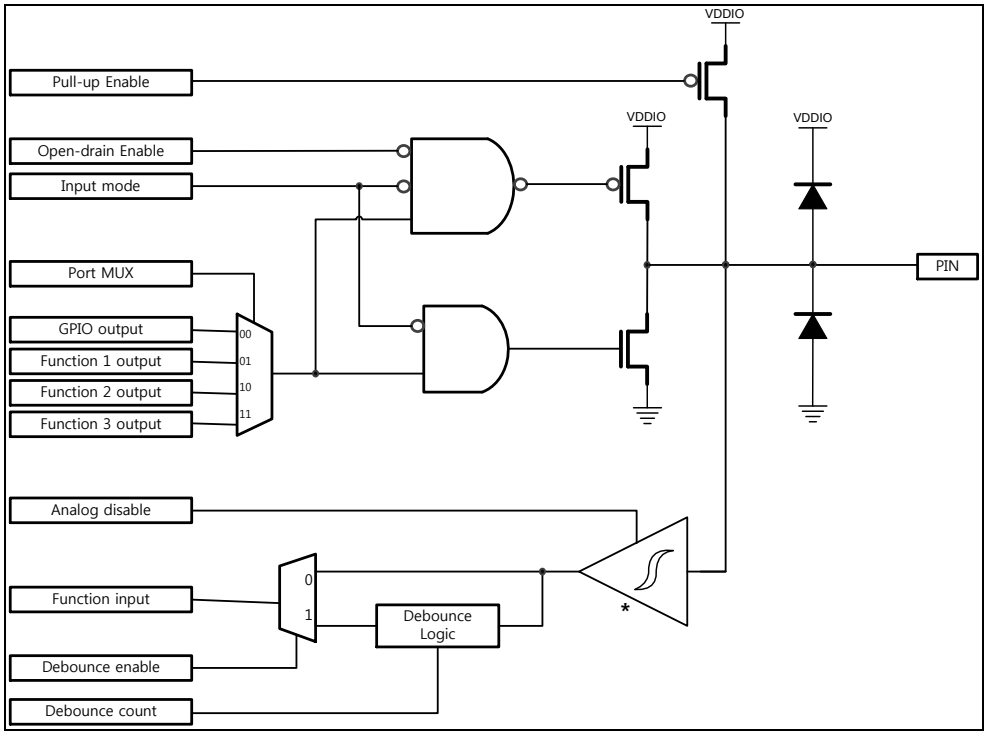


Figure 5-3 I/O Port Block Diagram (General I/O Pins)

Pin Multiplexing

GPIO pins have alternative function pins. Table 5-1 shows pin multiplexing information.

Table 5-1 GPIO Alternative Function

PORT	PIN	FUNCTION			
		00	01	10	11
PA	0	PA0*	T2IO		AIN0
	1	PA1*	T3IO		AIN1
	2	PA2*	SS	WDTO	AIN2
	3	PA3*	SCK	STBO	AIN3
	4	PA4*			AIN4
	5	PA5*			AIN5
	6	PA6*	T0IO		AIN6
	7	PA7*	T1IO		AIN7
	8	PA8*	T2IO	T0IO	AIN8
	9	PA9*	T3IO	T1IO	AIN9
	10	PA10*			AIN10
	11	PA11*			AIN11
	12	PA12*	T0IO		
	13	PA13*	T1IO		
	14	PA14*	T2IO		
	15	PA15*	T3IO		
PB	0	PB0*	MPWMUH	SS	
	1	PB1*	MPWMUL	SCK	
	2	PB2*	MPWMVH	MOSI	
	3	PB3*	MPWMVL	MISO	
	4	PB4*	MPWMWH		
	5	PB5*	MWMWL		
	6	PB6*	PRTIN		
	7	PB7*	OVIN		
PC	0	PC0	SWCLK*	RXD1	
	1	PC1	SWDIO*	TXD1	
	2	PC2*			
	3	PC3*			
	4	PC4*		T0IO	
	5	PC5*	RXD1	T1IO	
	6	PC6*	TXD1	T2IO	
	7	PC7*	SCL	T3IO	
	8	PC8*	SDA		VMRG
	9	PC9*	CLKO		
	10	PC10	nRESET*		
	11	PC11	BOOT*	T0IO	
	12	PC12*	T3IO		XIN
	13	PC13*	T2IO		XOUT
	14	PC14*	RXD0		
	15	PC15*	TXD0		
PD	0	PD0*	SS		
	1	PD1*	SCK		
	2	PD2*	MOSI	SCL	SXOUT
	3	PD3*	MISO	SDA	SXIN

(*) indicates default pin setting
(2) indicates secondary port

Registers

The base address of the PCU block is 0x4000_1000.

Register access is globally masked by the PORTEN register. To change register values except the PORTEN register, enable port access in advance.

Table 5-2 Base Address of Each Port Control

NAME	BASE ADDRESS
PCA	0x4000_1000
PCB	0x4000_1100
PCC	0x4000_1200
PCD	0x4000_1300

Table 5-3 PCU Register Map

NAME	OFFSET	TYPE	DESCRIPTION
PC*n*.MR	0x--00	RW	Port *n* pin mux select register
PC*n*.CR	0x--04	RW	Port *n* pin control register
PC*n*.PCR	0x--08	RW	Port *n* internal pull-up control register
PC*n*.DER	0x--0C	RW	Port *n* debounce control register
PC*n*.IER	0x--10	RW	Port *n* interrupt enable register
PC*n*.ISR	0x--14	RW	Port *n* interrupt status register
PC*n*.ICR	0x--18	RW	Port *n* interrupt control register
	0x--1C		Reserved
PORTEN	0x1FF0	RW	Port Access enable

PCA.MR PORT A Pin MUX Register

This is the PA Port mode selection register. This register must be set properly before using the port to ensure it functions correctly.

PCA.MR=0x4000_1000

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PA0	PA0*	T2IO		AIN0
PA1	PA1*	T3IO		AIN1
PA2	PA2*	SS	WDTO	AIN2
PA3	PA3*	SCK	STBO	AIN3
PA4	PA4*			AIN4
PA5	PA5*			AIN5
PA6	PA6*	T0IO		AIN6
PA7	PA7*	T1IO		AIN7
PA8	PA8*	T2IO	T0IO	AIN8
PA9	PA9*	T3IO	T1IO	AIN9
PA10	PA10*			AIN10
PA11	PA11*			AIN11
PA12	PA12*	T0IO		
PA13	PA13*	T1IO		
PA14	PA14*	T2IO		
PA15	PA15*	T3IO		

PCB.MR PORT B Pin MUX Register

This is the PB Port mode select register. This register must be set properly before using the port.to ensure it functions correctly.

PCB.MR=0x4000_1100

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PB0	PB0*	MPWMUH	SS	
PB1	PB1*	MPWMUL	SCK	
PB2	PB2*	MPWMVH	MOSI	
PB3	PB3*	MPWMVL	MISO	
PB4	PB4*	MPWMWH		
PB5	PB5*	MPWMWL		
PB6	PB6*	PRTIN		
PB7	PB7*	OVIN		

PCC.MR PORT C Pin MUX Register

This is the PC Port mode selection register. This register must be set properly before using the port to ensure it functions correctly.

PCC.MR=0x4000_1200

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
00	00	00	00	01	01	00	00	00	00	00	00	00	00	01	01
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PC0	PC0	SWCLK*	RXD1	
PC1	PC1	SWDIO*	TXD1	
PC2	PC2*			
PC3	PC3*			
PC4	PC4*		T0IO	
PC5	PC5*	RXD1	T1IO	
PC6	PC6*	TXD1	T2IO	
PC7	PC7*	SCL	T3IO	
PC8	PC8*	SDA		VMRG
PC9	PC9*	CLKO		
PC10	PC10	nRESET*		
PC11	PC11	BOOT*	T0IO	
PC12	PC12*	T3IO		XIN
PC13	PC13*	T2IO		XOUT
PC14	PC14*	RXD0		
PC15	PC15*	TXD0		

PCD.MR PORT D Pin MUX Register

This is the PD Port mode selection register. This register must be set properly before using the port to ensure it functions correctly.

PCD.MR=0x4000_1300

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PD0	PD0*	SS		
PD1	PD1*	SCK		
PD2	PD2*	MOSI	SCL	SXOUT
PD3	PD3*	MISO	SDA	SXIN

PCn.CR PORT n Pin Control Register (Except for PCC.CR)

This register is used for input or output control of each port pin. Each pin can be configured as input pin, output pin, or open-drain pin.

PCA.CR=0x4000_1004, PCB.CR=0x4000_1104, PCD.CR=0x4000_1304

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Pn	Port control
00	Push-pull output
01	Open-drain output
10	Input
11	Analog

PCC.CR PORT C Pin Control Register

This register is used for input or output control of each port pin. Each pin can be configured as input pin, output pin, or open-drain pin.

PCC.CR=0x4000_1204

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	10	10	11	11	11	11	11	11	11	11	10	10
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Pn	Port control
00	Push-pull output
01	Open-drain output
10	Input
11	Analog

PCn.PCR PORT n Pull-up Resistor Control Register (Except for PCC.PCR)

Every pin in the port has on-chip pull-up resistors which can be configured by the PCn.PCR registers.

PCA.PCR=0x4000_1008, PCB.PCR=0x4000_1108
PCD.PCR=0x4000_1308

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUE15	PUE14	PUE13	PUE12	PUE11	PUE10	PUE9	PUE8	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
0000															
RW															

n	PUEn	Port pull-up control
0		Disable pull-up resistor
1		Enable pull-up resister

PCC.PCR PORT C Pull-up Resistor Control Register

Every pin in the port has on-chip pull-up resistors which can be configured by the PCC.PCR registers.

PCC.PCR=0x4000_1208

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUE15	PUE14	PUE13	PUE12	PUE11	PUE10	PUE9	PUE8	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
0C03															
RW															

n	PUEn	Port pull-up control
		0 Disable pull-up resistor
		1 Enable pull-up resister

PCn.DER PORT n Debounce Enable Register

Every pin in the port has a digital debounce filter which can be configured by the PCn.DER registers.

PCA.DER=0x4000_100C, PCB.DER=0x4000_110C
PCC.DER=0x4000_120C, PCD.DER=0x4000_130C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDE15	PDE14	PDE13	PDE12	PDE11	PDE10	PDE9	PDE8	PDE7	PDE6	PDE5	PDE4	PDE3	PDE2	PDE1	PDE0
0000															
RW															

	PDEn	Pin debounce enable
		0 Disable debounce filter
		1 Enable debounce filter

PCn.IER PORT n Interrupt Enable Register

The entire pin can be an external interrupt source. The trigger interrupt and level trigger interrupt are supported. The Interrupt mode can be configured by setting the PCn.IER registers.

PCA.IER=0x4000_1010, PCB.IER=0x4000_1110
PCC.IER=0x4000_1210, PCD.IER=0x4000_1310

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIE15		PIE14		PIE13		PIE12		PIE11		PIE10		PIE9		PIE8		PIE7		PIE6		PIE5		PIE4		PIE3		PIE2		PIE1		PIE0	
00		00		00		00		00		00		00		00		00		00		00		00		00		00		00		00	
RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW	

	PIEn	Pin interrupt enable
		00 Interrupt disabled
		01 Enable interrupt as level trigger mode
		10 Reserved
		11 Enable interrupt as edge trigger mode

PCn.ISR PORT n Interrupt Status Register

When an interrupt is delivered to the CPU, the interrupt status can be detected by reading the PCn.ISR register. The PCn.ISR register reports the interrupt source pin and type of interrupt.

PCA.ISR=0x4000_1014, PCB.ISR=0x4000_1114
PCC.ISR=0x4000_1214, PCD.ISR=0x4000_1314

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PIS15	PIS14	PIS13	PIS12	PIS11	PIS10	PIS9	PIS8	PIS7	PIS6	PIS5	PIS4	PIS3	PIS2	PIS1	PIS0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PISn	Pin interrupt status
00	No interrupt event
01	Low level interrupt or Falling edge interrupt event is present
10	High level interrupt or rising edge interrupt event is present
11	Both of rising and falling edge interrupt event is present in edge trigger interrupt mode. Not available in level trigger interrupt mode

PCn.ICR PORT n Interrupt Control Register

This is the Interrupt Mode Control register.

PCA.ICR=0x4000_1018, PCB.ICR=0x4000_1118
PCC.ICR=0x4000_1218, PCD.ICR=0x4000_1318

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PIC15	PIC14	PIC13	PIC12	PIC11	PIC10	PIC9	PIC8	PIC7	PIC6	PIC5	PIC4	PIC3	PIC2	PIC1	PIC0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PICn	Pin interrupt mode
00	Prohibit external interrupt
01	Low level interrupt or Falling edge interrupt mode
10	High level interrupt or Rising edge interrupt mode
11	Both rising and falling edge interrupt mode. No support for level trigger mode

PORTEN Port Access Enable

The Port Access Enable (PORTEN) registers enable register-writing permissions for all PCU registers.

PORTEN=0x4000_1FF0

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

7 0	PORTEN	Writing the sequence of 0x15 and 0x51 in this register enables writing to PCU registers, and writing other values protects all PCU registers from writing.

Note: How to use PORTEN:

```
PORTEN=0x15; PORTEN=0x51;   // enable PORTEN
…                           // set PCn.MR, PCn.CR PCn.PCR and etc.
PORTEN=0;                   // disable PORTEN
```

Functional Description

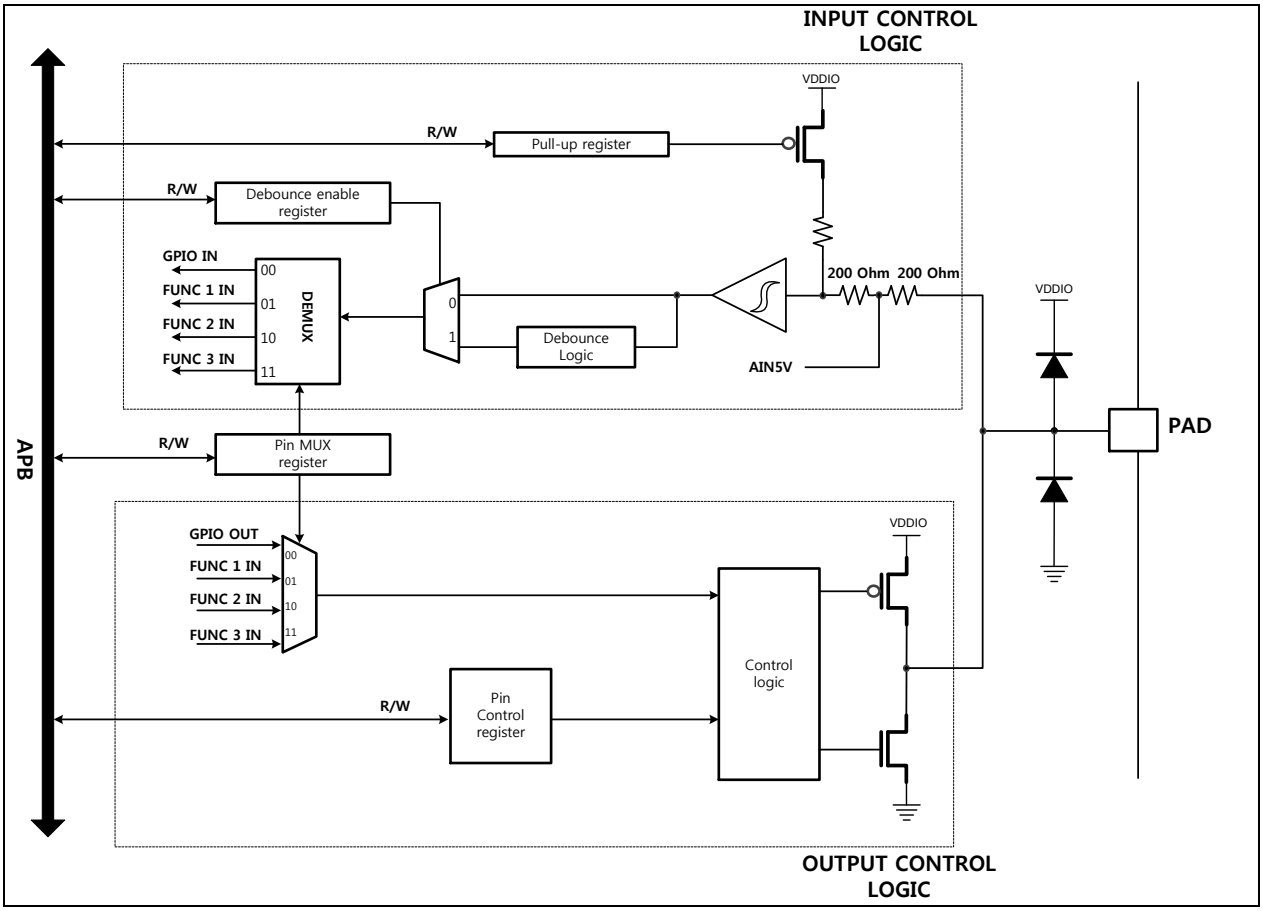


Figure 5-4 Functional Block Diagram

When the input functions of I/O port are used by the Pin Control register, the output function of I/O port is disabled. The Port function differs according to the the Pin Mux register.

The Input Data register captures the data present on the I/O pin or debounced input data at every GPIO clock cycle.

When the debounce functions of input data are used by the Debounce Enable register, the external input data is captured by the Debounce CLK.

- If CNT Value is “01”, Debounced Input Data is “1”
- If CNT Value is “10”, Debounced Input Data is “0”

It is possible to change the Debounce CLK of each port group used by the MCCR4~5 register.

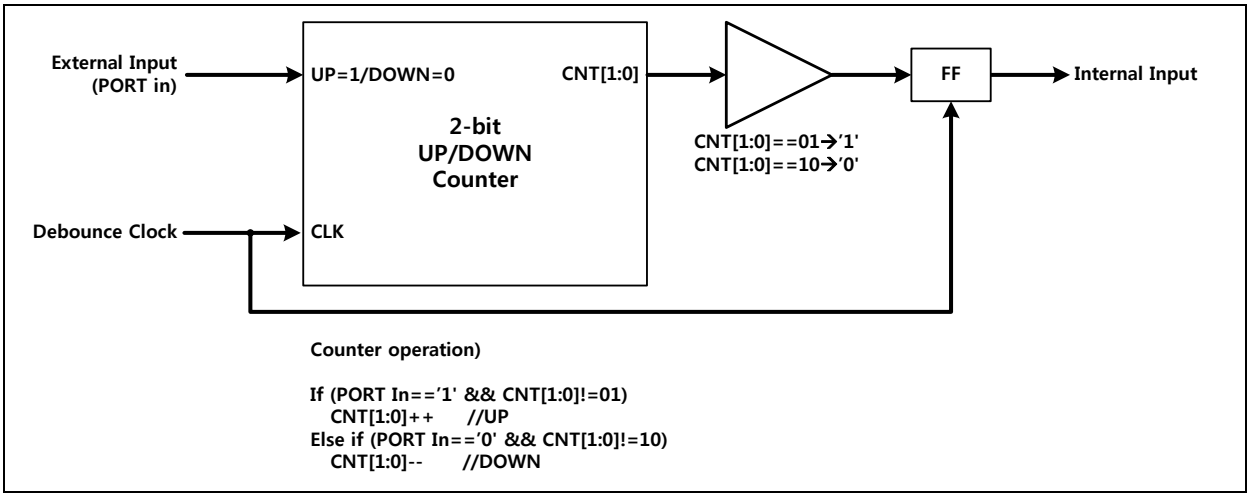


Figure 5-5. Debounce Logic

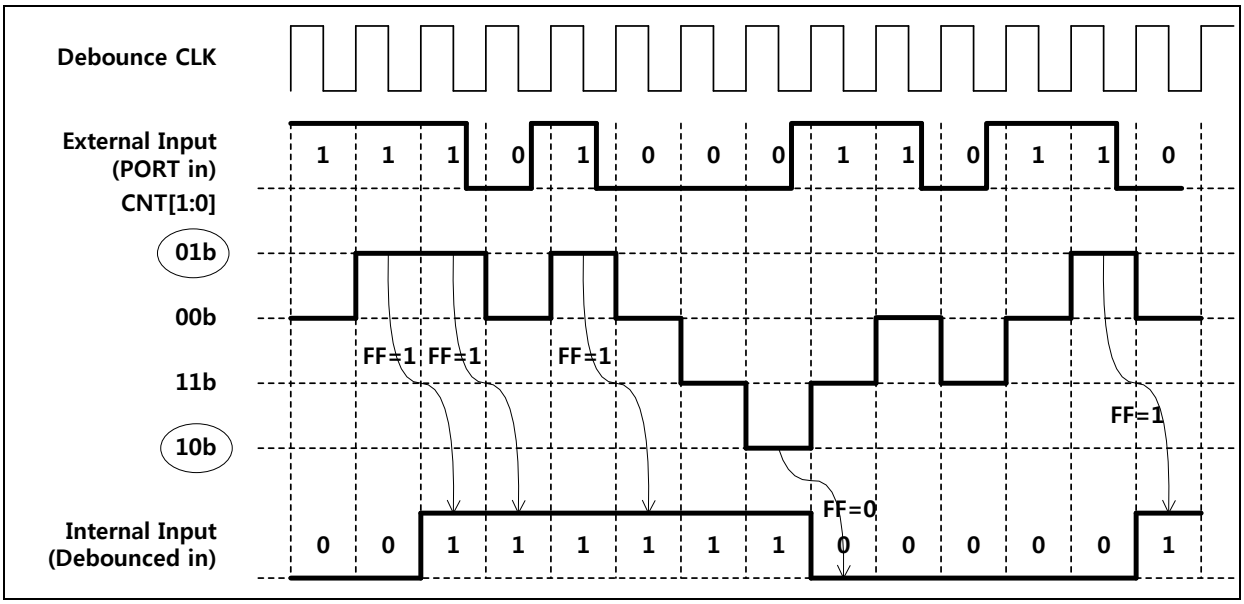


Figure 5-6. Port Debounce Example

6. General Purpose I/O (GPIO)

Overview

Most pins except dedicated function pins can be used as general I/O ports. General input/output ports are controlled by the GPIO block.

- Output signal level (H/L) select
- Read Input signal level

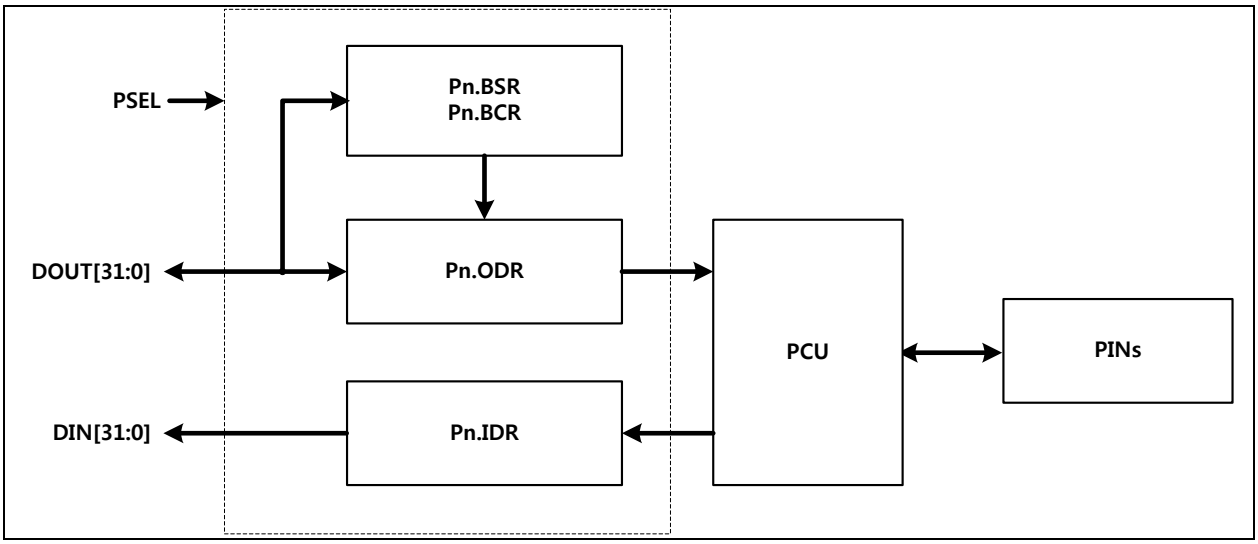


Figure 6-1 Block Diagram

Pin Description

Table 6-1 External Signal

PIN NAME	TYPE	DESCRIPTION
PA	IO	PA0 – PA15
PB	IO	PB0 – PB7
PC	IO	PC0 – PC15
PD	IO	PD0 – PD3

Registers

The base address of GPIO is 0x4000_2000 and the register map is described in Table 6-2 and Table 6-3.

Table 6-2 Base Address of Each Port

NAME	BASE ADDRESS
PA PORT	0x4000_2000
PB PORT	0x4000_2100
PC PORT	0x4000_2200
PD PORT	0x4000_2300

Table 6-3 GPIO Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
Pn.ODR	0x--00	RW	Port *n* Output data register	0x00000000
Pn.IDR	0x--04	RO	Port *n* Input data register	0x00000000
Pn.BSR	0x--08	WO	Port *n* Pin set register	0x00000000
Pn.BCR	0x—0C	WO	Port *n* Pin clear register	0x00000000

Pn.ODR PORT n Output Data Register

When the pin is set as output and GPIO mode, the pin output level is defined by the Pn.ODR registers.

PA.ODR=0x4000_2000, PB.ODR=0x4000_2100
PC.ODR=0x4000_2200, PD.ODR=0x4000_2300

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR															
0000															
RW															

ODR	Pin output level
	0 Output low level
	1 Output high level

Pn.IDR PORT n Input Data Register

Each pin level status can be read in the Pn.IDR register. Even if the pin is in alternative mode except analog mode, the pin level can be detected in the Pn.IDR register.

PA.IDR=0x4000_2004, PB.IDR=0x4000_2104
PC.IDR=0x4000_2204, PD.IDR=0x4000_2304

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR															
0000															
RO															

IDR	Pin current level
	0 The pin is low level
	1 The pin is high level

Pn.BSR PORT n Bit Set Register

Pn.BSR is a register for controlling each bit of the Pn.ODR register. Writing a ‘1’ into the specific bit will set a corresponding bit of Pn.ODR to ‘1’. Writing ‘0’ in this register has no effect.

PA.BSR=0x4000_2008, PB.BSR=0x4000_2108
PC.BSR=0x4000_2208, PD.BSR=0x4000_2308

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSR															
0000															
WO															

BSR	Pin current level
0	Not effect
1	Set correspondent bit in Pn.ODR register

Pn.BCR PORT n Bit Clear Register

Pn.BCR is a register for controlling each bit of the Pn.ODR register. Writing a ‘1’ into the specific bit will set a corresponding bit of Pn.ODR to ‘0’. Writing ‘0’ in this register has no effect.

PA.BCR=0x4000_200C, PB.BCR=0x4000_210C
PC.BCR=0x4000_220C, PD.BCR=0x4000_230C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCR															
0000															
WO															

BCR	Pin current level
0	Not effect
1	Clear correspondent bit in Pn.ODR register

Functional Description

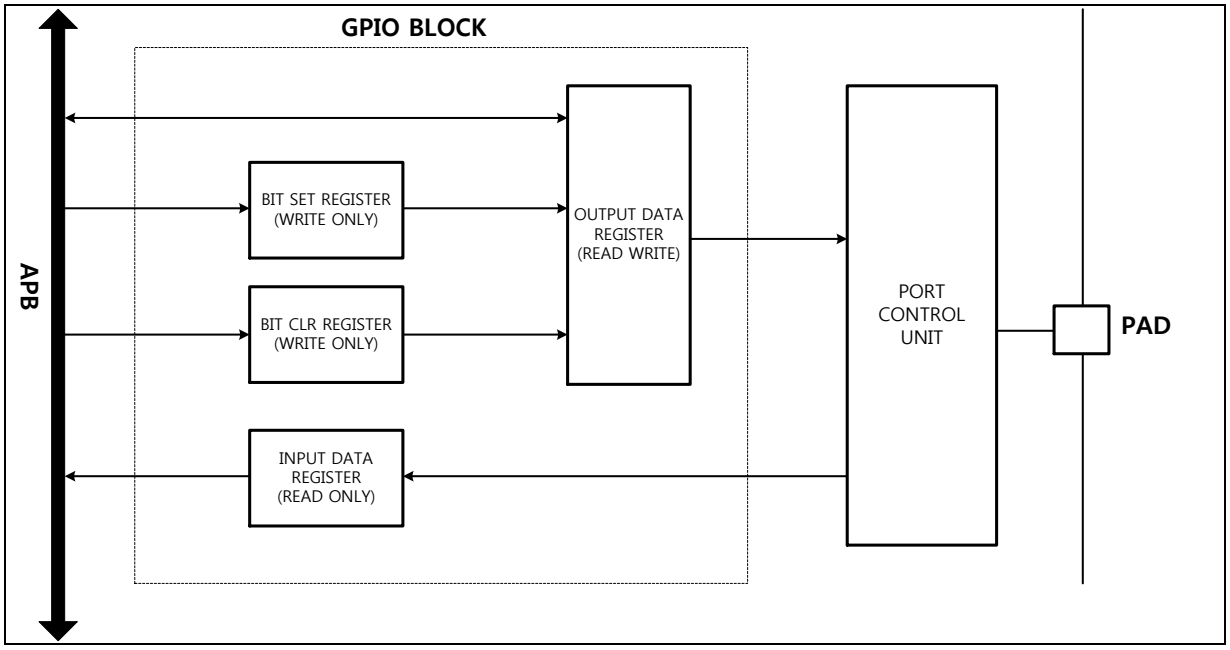


Figure 6-2 Functional Block Diagram

When configured as output, the value written to the GPIO Output Data register is output on the I/O Pin.

When setting the Bit Set register, the GPIO Output Data register sets the High. When setting the Bit Clr register, the GPIO Output Data register sets the Low.

The Input Data register captures the data present on the I/O pin or Debounced input data at every GPIO clock cycle.

7. Flash Memory Controller

Overview

The Flash Memory Controller is an internal Flash memory interface controller with the following features:

- 64/32 KB Flash memory with protection bits
- 32 word length program or erase at a time
- Bulk erase for 64/32 KB memory at a time
- 32 word size OTP area
- 50 ns Flash access read time
- 0-wait (under 20 MHz), 1-wait, 2-wait, and pre-fetch (read acceleration) access support
- Uses internal 40 MHz OSC clock for Erase/Program timing control

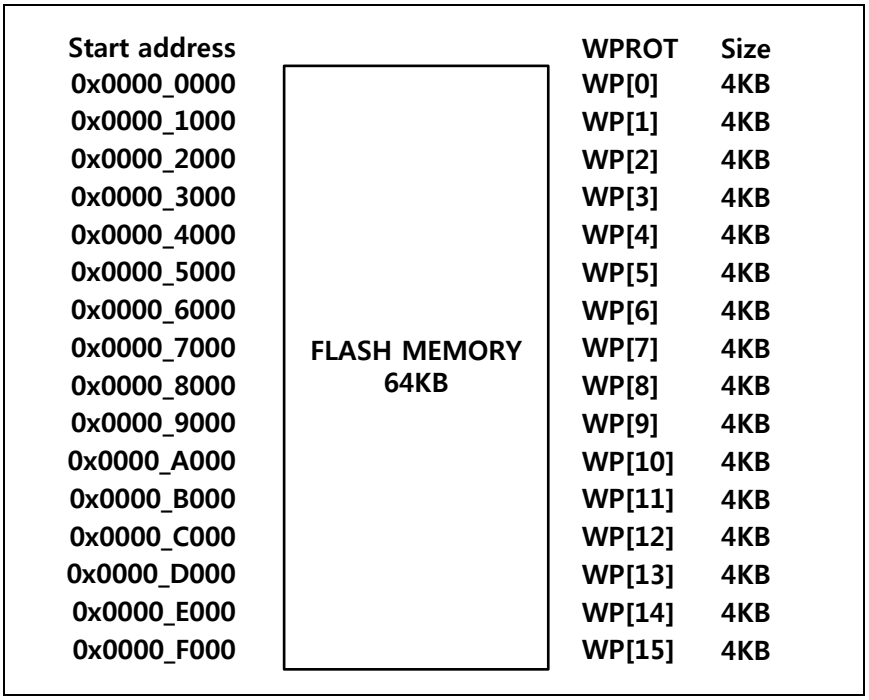


Figure 7-1 Block Diagram

Registers

The base address of the Flash Memory Controller is listed in Table 7-1.

Table 7-1 Flash Memory Controller Base Address

NAME	BASE ADDRESS
Flash Controller	0x4000_0100

Table 7-2 shows the Register memory map.

Table 7-2 FMC Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
FM.MR	0x0004	RW	Flash Memory Mode Select register	0x01000000
FM.CR	0x0008	RW	Flash Memory Control register	0x05000000
FM.AR	0x000C	RW	Flash Memory Address register	0x00000000
FM.DR	0x0010	RW	Flash Memory Data register	0x00000000
FM.TMR	0x0014	RW	Flash Memory Timer register	0x00018FFF
FM.TICK	0x001C	R	Flash Memory Tick Timer	0x00000000
FM.CRC	0x0020	R	Flash CRC16 check value	0x00000000
FM.CFG	0x0030	RW	Flash Memory Configuration value	0x00008200
FM.HWID	0x0040	R	Second HW ID for AC30M1x64/1x32	0x30146400
BOOTCR	0x0074	RW	Boot ROM clear, SRAM Remap register	0x00000000
FM.WPROT	0x0078	RW	Write Protection register	0x00FFFF00
FM.RPROT	0x007C	RW	Read Protection register	0x000000FF

FM.MR Flash Memory Mode Register

This is an internal 32-bit Flash memory mode register.

FM.MR=0x4000_0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							IDLE	TESTEN	AMBAEN	PROTEN				TRMEN	TRM							FEMOD	FMOD	ACODE							
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00							
							R	R	R					R	R							R	R	RW							

24	IDLE	0	Flash Idle state bit (“0” means flash busy for PGM or ERS)
		1	Flash Idle state bit (“1” means flash idle, free to read)
23	TESTEN (test only)	0	Flash test register disable (“0” means cannot set TEST reg)
		1	Flash test register enable (“0” means can set TEST reg)
22	AMBAEN	0	AMBA mode disabled status
		1	AMBA mode enable (can change wait state and etc)
21	PROTEN	0	Flash protection register disable (“0” means cannot access protection register)
		1	Flash protection register enable (“1” means can access protection register)
17	TRMEN	0	TRIM mode disabled status
		1	Trim mode entry status(read only)
16	TRM	0	TRIM mode disabled
		1	Trim mode status(read only) must be set with TRMEN
9	FEMOD	0	Flash (program/erase) mode disabled
		1	Flash mode entry status(read only)
8	FMOD	0	Flash (program/erase) mode disabled
		1	Flash mode status(read only) must be set with FEMOD
7 0	ACODE	5A → A5	Flash mode entry sequence
		A5 → 5A	Trim mode entry sequence
		81 → 28	AMBA mode entry sequence
		66 → 99	PROT mode entry sequence
		39 → 7D	TESTEN mode entry sequence (test only)

FM.CR Flash Memory Control Register

This is an internal Flash memory control register. FM.CR[17:0] bits can be accessed while Flash mode entry is activated. FMCR[31:28] bits can be accessed in Trim mode.

FM.CR=0x4000_0108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTP3	OTP2	OTP1	OTP0								TMREN			TEST		VPPOUT	EVER	PVER	BLKE	DMYE	OTPE	AEE	AEF	SUBACT	PPGM	PMODE	WE	PBLD	PGM	ERS	PBR
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RW	RW	RW	RW								RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31	OTP3	0	
		1	OTP area 3 access enable (user can access)
30	OTP2	0	
		1	OTP area 2 access enable (user can access)
29	OTP1	0	
		1	OTP area 1 access enable (user can access in a certain condition), OTP1 is used for read protection
28	OTP0	0	
		1	OTP area 0 access enable (user can not erase/program this area)
20	TMREN	0	Flash Tick timer enable
		1	Flash tick timer enable Tick timer runs by system clock while PGM or ERS undergoing
17	TEST	00	Normal operation
16		01	(read) Row voltage mode
		01	(write) ODD Row program
		10	Even Row program
		11	All Row program
15	VPPOUT	0	
		1	Charge pump Vpp output
14	EVER	0	
		1	Erase verify mode
13	PVER	0	
		1	Program verify mode
12	BLKE	0	
		1	128page write enable for full chip writing to save program time
11	DMYE	0	
		1	DUMMY area enable.
10	OTPE	0	
		1	OTP area A, B, C, D enable (user cannot access otp directly)
9	AEE	0	
		1	Pre PGM enable , Page buffer set automatically
8	AEF	0	
		1	All erase 64/32KB code area enable
7	SUBACT	0	
		1	SUB Active mode (System clock under 1MHz)
6	PPGM	0	
		1	Pre-PGM for Erase operation (pre-program before erase)
5	PMODE	0	
		1	PMODE enable(Address path changing)
4	WE	0	
		1	Write enable

3	PBLD	0	
		1	Page buffer load(WE should be set)
2	PGM	0	
		1	Program mode enable
1	ERS	0	
		1	Erase mode enable
0	PBR	0	
		1	Page buffer reset

FM.AR Flash Memory Address Register

This is an internal Flash memory program, erase address register.

FM.AR=0x4000_010C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		FADDR													
0	0	0x0000													
		RW													

13 0	FADDR	14-bit address covers 16K words address (one word = 4 bytes)

FM.DR Flash Memory Data Register

This is an internal Flash memory program data register.

FM.DR=0x4000_0110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDATA																															
0x0000_0000																															
RW																															

31 0	FDATA	Flash PGM data (32-bit)

FM.TMR Flash Memory Timer Register

This is an internal Flash memory timer value register (18-bit). The Erase/Program timer runs up to {TMR[17:0]}.

FM.TMR=0x4000_0114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														TMR																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x18FFF																	
								RW																							

	TMR	Erase/PGM timer (default, 0x18FFF) Timer counts up to {TMR[17:0]} by 40MHz HSI OSC clock

FM.DIRTY Flash Memory Dirty Bit Register

FMDRTY is the internal Flash memory dirty bit clear register.

FM.DR=0x4000_0110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDIRTY																															
0x0000_0000																															
WO																															

31 0	FDIRTY	Write any value here, cache line fill flag will be cleared

FM.TICKFlash Memory Tick Timer Register

This is an internal Flash memory tick timer register.

FM.TICK=0x4000_011C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														FTICK																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000																	
								RW																							

17 0	FTICK	TICK goes to 0x3FFFF from written TICK value while TMR runs by PCLK clock while Flash PGM or ERS (counts up only when IDLE bit of FMMR register is low)

FM.CRC Flash CRC Check Register

FMCRC is the CRC value resulting from read accesses on internal Flash memory.

FM.CRC=0x4000_012C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CRC16															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

15 0	CRC16	CRC16 check value read register polynomial: (1 + x5 + x12 + x16) data width: 32 (the first serial bit is D[31])

FM.CFG Flash Memory Configuration Register

This is an internal Flash memory Configuration register. This register has the same address as the FMTRIM0 register.

FM.CFG=0x4000_0130

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

31 16	WRITEKEY		Write key 0x7858
15	HRESPDIS	0	By default, when CPU try to write ROM area directly, flash interface will return ERROR response by AMBA protocol
		1	Disable HRESP(error response function) of Data or System bus (HRESP is AMBA AHB signal) This bit only be written in AMBA mode and MSB 16-bit (bit [31:16]) must be 0x7858
12	TESTCLK	0	TEST Clock selection (test purpose only) Set “1” to use system bus clock instead of internal 40MHz OSC This bits only be written in AMBA mode and MSB 16-bit (bit [31:16]) must be 0x7858
9 8	WAIT		This bits only be written in AMBA mode and MSB 16-bit (bit [31:16]) must be 0x7858
		00	WAIT is 00, flash access in 1 cycle (0-wait)
		01	WAIT is 01, flash access in 2 cycles (1-wait)
		10	WAIT is 10, flash access in 3 cycles (2-wait) – default
		11	WAIT is 11, flash access in pre-fetch mode Note) In pre-fetch mode, OTP (0x3F0000xx~0x3F0005xx) read and Program/Erase operation would not work correctly. User must exit from pre-fetch mode to read OTP or program/erase flash memory
7	CRCINIT	0	When this bit is set(‘1’), CRC register will be initialized It should be reset again before read flash to generate CRC16 calculation (Initial value of FMCRC is 0xFFFF)
6	CRCEN	0	CRC16 enable CRC value will be calculated at every flash read timing

FM.HWID Flash Hardware ID Register

The Flash Hardware ID register is a 32-bit read-only register for correct size information.

FM.HWID=0x4000_0140

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FHWID																															
0x3014_6400																															
R																															

31 0	FHWID	Flash HWID register It returns size option values 0x30146400 : 64KB flash product option 0x30143200 : 32KB flash product option 0x30FF0000 : wrong size option code, 64KB flash enable

BOOTCR Boot ROM Remap Clear Register

The Boot ROM remap clear register is an 8-bit register.

BOOTCR=0x4000_0174

7	6	5	4	3	2	1	0
			SREMAP				BOOTROM
0	0	0	0	0	0	0	1
							R

4	SREMAP	SRAM remap enable register When this bit is set, SRAM will be located at 0x0000_0000 address. This bit location can be accessed in AMBA mode Flash memory also can be read at 0x3000_0000 while SREAMP enable
0	BOOTROM	Boot Mode (only can be written in boot loader mode) This bit is used to clear boot loader mode at end of boot code, user cannot re-activate this bit. Always 0 in user mode.

Note) SREMAP bit can be writable when AMBA mode is enabled

```
FM->MR=0x81;
FM->MR=0x28;        // AMBA mode enter
…                   // change BOOTCR[4](SREMAP) value
FM->MR=0;           // AMBA mode exit
```

FM.WPROT Flash Memory Write Protection Register

This is an internal Flash memory write protection register. This register is updated from the OTP area of Flash during boot sequence; users cannot write to this register or clear any bit directly.

FM.WPROT=0x4000_0178

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

31 24	WPEN	Write Protect Access Enable Sectors 0-1: 0x98 Sectors 2-15: 0x87
15 0	WP	Sector(4KB block each) protect Each bit enable write protect corresponding 4K block when WP bit is set (‘1’). (Write protect enabled at boot)

FM.RPROT Flash Memory Read Protection Register

This is an internal Flash memory read protection register. This register is updated from the OTP area of Flash during boot sequence; therefore, users cannot write to or clear any bit directly.

FM.RPROT=0x4000_017C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCK2	LOCK1	JTAGDIS	-					-								-								RPEN							
0	0	0	-					-								-								0xFF							
RO	RO	RO														'								RW							

31	LOCK2	Read protection level 2 state flag
30	LOCK1	Read protection level 1 state flag
29	JTAGDIS	JTAG disable state flag
7 0	RPEN	Read Protection Enable/Disable By default, read protection disable (FM.RPROT = 0xFF)
	LOCK1	Read protection level 1 Code protection mode enable, debug can be connected Write 0x39 to activate LOCK1 (only can be written in Unlock state) Code in SRAM or debugger cannot read flash area When flash was read from SRAM or debugger, 0xA5A5A5A5 will be return as read data
	LOCK2	Read protection level 2 Code protection mode enable, debug cannot be connected Write any value except 0x39(include 0xFF) to activate LOCK2 (only can be written in Unlock state or LOCK1) When Flash was read from SRAM, 0xA5A5A5A5 will be return as read data

Functional Description

The Flash memory controller is an internal Flash memory interface controller which primarily controls the programming of Flash memory and preparing read data to be requested from the bus.

Flash Organization

The 64 Kbytes code Flash memory consists of 512 pages which have a uniform 128 byte page size. The Flash controller allows reading or writing of Flash memory data. This memory is located at 0x0000_0000 address on the system memory map. The system expects the code to be executed on boot to be located at address 0x0000_0000. There is no ability to change this address on the Cortex M0.

Flash Read Operation

The Flash data read operation is requested from the bus. The Flash controller responds to the request. The wait time should be correctly defined because the bus speed is usually faster than the Flash data access time. The Flash data access time is 20 Mhz on the Z32F0642 device.

Flash Program Operation

Erase and Program access of Flash memory is available only in Flash mode. Once in Flash mode, Flash cannot be read normally; therefore, self-programming is not supported. The Flash program erase operations must be performed by the execution program in SRAM memory.

For every erase operation, a pre-program operation MUST be performed first, to prevent over-erase of Flash memory cells. Programming and erase operations use the 40 Mhz internal oscillator, so the HSI internal oscillator must be enabled and selected.

Erase operations can be either a page (32 words) or the entire chip. Programming can be a single word or a page.

Flash Erase and Program Examples

To erase a sector:

A. Flash mode enable to write FM.CR register (write 0x5A and then write 0xA5 into FMMR)
B. Set PMODE bit first
C. Wait until IDLE bit of FM.MR register becomes “1”
D. Set target Page address in FM.AR
E. Set FM.TMR register to be 0.5ms operation (based on 40MHz Int OSC clock)
F. set PPGM, WE, PGM bits of FMCR
G. Wait until IDLE bit of FM.MR register become “1” after pre-program
H. Clear WE, PGM bits of FMCR
I. Wait 5us
J. Clear PPGM bit of FM.CR
K. Wait 30us before returning to normal operation
L. Clear PMODE bit of FM.CR
M. Clear Flash mode (write 0x00 into FM.MR)
N. Insert at least 2 NOPs, and return to normal operation
O. Flash mode enable to write FM.CR register (write 0x5A and then write 0xA5 into FM.MR)
P. Set PMODE bit first

Q. Wait until IDLE bit of FM.MR register becomes “1”
R. Set FM.TMR register to be 2.5ms operation (based on 40 MHz Int OSC clock)
S. Set target Page address in FM.AR
T. set WE, ERS bits of FM.CR
U. Wait until IDLE bit of FM.MR register become “1” after erase
V. Clear WE, ERS bits of FM.CR
W. Wait 30us before returning to normal operation
X. Clear PMODE bit of FM.CR
Y. Clear Flash mode (write 0x00 into FM.MR)
Z. Insert at least 2 NOPs, and return to normal operation

To Program a page (after erase):

A. Flash mode enable to write FM.CR register (write 0x5A and then write 0xA5 into FM.MR)
B. Set PMODE bit first
C. Wait until IDLE bit of FM.MR register becomes “1”
D. Set PBR bit of FM.CR and clear PBR bit of FM.CR(page buffer reset)
E. Set target Page address in FM.AR
F. Set PBLD bit of FM.CR to load data into page buffer
G. Write word(32-bit) data into FM.DR (max 32 words), address increased automatically based on word address
H. Clear PBLD bits of FM.CR
I. Set target Page address in FM.AR again
J. Set FM.TMR register to be 2.5ms operation (based on 40MHz Int OSC clock)
K. Set WE, PGM bits of FM.CR
L. Wait until IDLE bit of FM.MR register become “1” after program
M. Clear WE, PGM bits of FM.CR
N. Wait 30us before returning to normal operation
O. Clear PMODE bits of FM.CR
P. Clear Flash mode (write 0x00 into FM.MR)
Q. Insert at least 2 NOPs, and return to normal operation

8. Internal SRAM

Overview

The Z32F0642 MCU has a block of 0-wait on-chip SRAM. The size of SRAM is 4KB. The SRAM base address is 0x2000_0000.

The SRAM memory area is usually used for data memory and stack memory. Sometimes the code is dumped into the SRAM memory for fast operation or Flash erase/program operation.

This device does support memory remap strategy to remap memory to 0x00000000–0x00000FFF. Flash memory can be accessed at 0x30000000 when SRAM is remapped. To remap the SRAM, set the SREMAP bit in the FM->BOOTCR register.

9. Watch-Dog Timer (WDT)

Overview

The Watchdog timer can monitor the system and generate an interrupt or a reset. It has a 32-bit down-counter.

- 32-bit down counter (WDT.CNT)
- Select reset or periodic interrupt
- Count clock selection
- Dedicated pre-scaler
- Watchdog underflow output signal

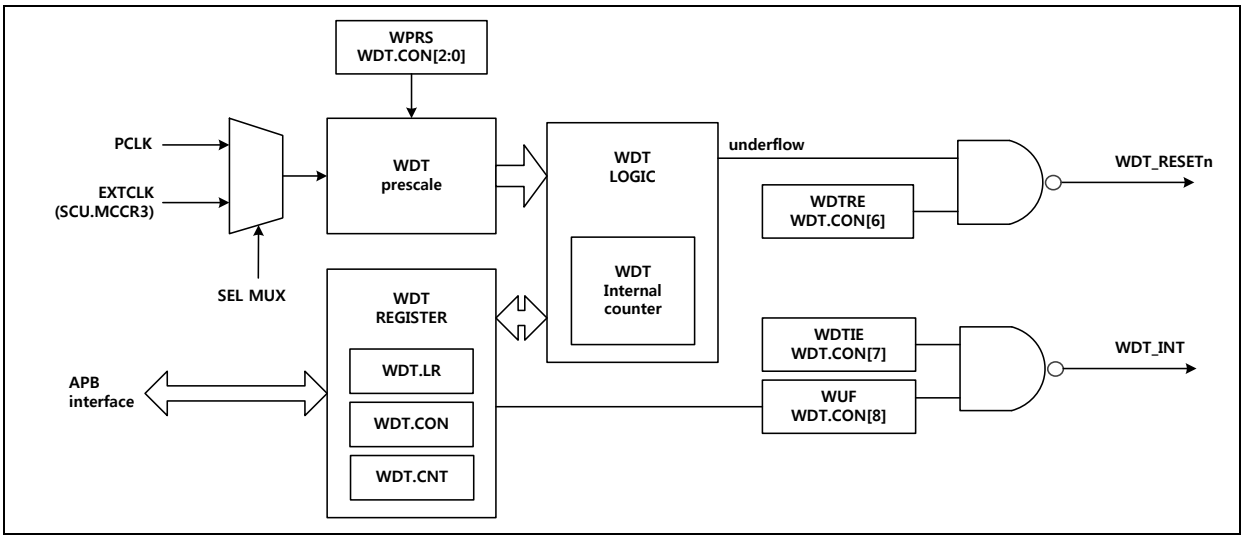


Figure 9-1 WDT Block Diagram

Registers

The base address of watchdog timer is 0x4000_0200 and the register map is described in Table 9-2. Initial watchdog time-out period is set to 2,000-miliseconds.

Table 9-1 Base Address of SCU

NAME	BASE ADDRESS
WDT	0x4000_0200

Table 9-2 Watchdog Timer Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
WDT.LR	0x0000	W	WDT Load register	0x00000000
WDT.CNT	0x0004	R	WDT Current counter register	0x00000000
WDT.CON	0x0008	RW	WDT Control register	0x0000805C

WDT.LR Watchdog Timer Load Register

The WDTLR register is used to update the WDTCNT register. To update the WDTCNT register, the WDTEN bit of WDTCON should be set to ‘1’ and written to the WDTLR register with a target value of WDTCNT. At least 5 WDT clocks are required to update WDTLR to WDTCNT. The WDT external clock source is controlled by WDTCSEL and WDTDIV in MCCR3.

WDT.LR=0x4000_0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTLR																															
0x0000_0000																															
RW																															

31 0	WDTLR	Watchdog timer load value register Keeping WEN bit as ‘1’, write WDTLR register will update WDTCNT value with written value

WDT.CNT Watchdog Timer Current Counter Register

The WDTCNT register represents the current count value of the 32-bit down counter .When the counter value reaches 0, the interrupt or reset is started.

WDT.CNT=0x4000_0204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTCNT																															
0x0000_0A3D																															
RO																															

31 0	WDTCNT	Watchdog timer current counter register 32-bit down counter will run from the written value.

WDT.CON Watchdog Timer Control Register

The WDT module should be configured properly before running. When the target purpose is defined, WDT can be configured in the WDTCON register.

WDT.CON=0x4000_0208

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDBG							WUF	WDTIE	WDTRE		WDTEN	CKSEL	WPRS		
1	0	0	0	0	0	0	0	0	1	0	1	1	100		
RW							RW	RW	RW		RW	RW	RW		

15	WDBG	Watchdog operation control in debug mode	
		0	Watchdog counter running when debug mode
		1	Watchdog counter stopped when debug mode
8	WUF	Watchdog timer underflow flag	
		0	No underflow
		1	Underflow is pending
7	WDTIE	Watchdog timer counter underflow interrupt enable	
		0	Disable interrupt
		1	Enable interrupt
6	WDTRE	Watchdog timer counter underflow interrupt enable	
		0	Disable reset
		1	Enable reset
4	WDTEN	Watchdog Counter enable	
		0	Watch dog counter disabled
		1	Watch dog counter enabled
3	CKSEL	WDTCLKIN clock source select	
		0	PCLK
		1	External clock
2 0	WPRS[2:0]	Counter clock prescaler WDTCLK = WDTCLKIN/WPRS	
		000	WDTCLKIN
		001	WDTCLKIN / 4
		010	WDTCLKIN / 8
		011	WDTCLKIN / 16
		100	WDTCLKIN / 32
		101	WDTCLKIN / 64
		110	WDTCLKIN / 128
		111	WDTCLKIN / 256

Functional Description

The watchdog timer count can be enabled by setting WDTEN (WDT.CON[4]) to '1'. As the watchdog timer is enabled, the down counter starts counting from the Load Value. If WDTRE (WDT.CON[6]) is set as '1', WDT reset will be asserted when the WDT counter value reaches '0' (underflow event) from the WDT.LR value. Before the WDT counter goes down to 0, the software can write a certain value to the WDT.LR register to reload the WDT counter.

Timing Diagram

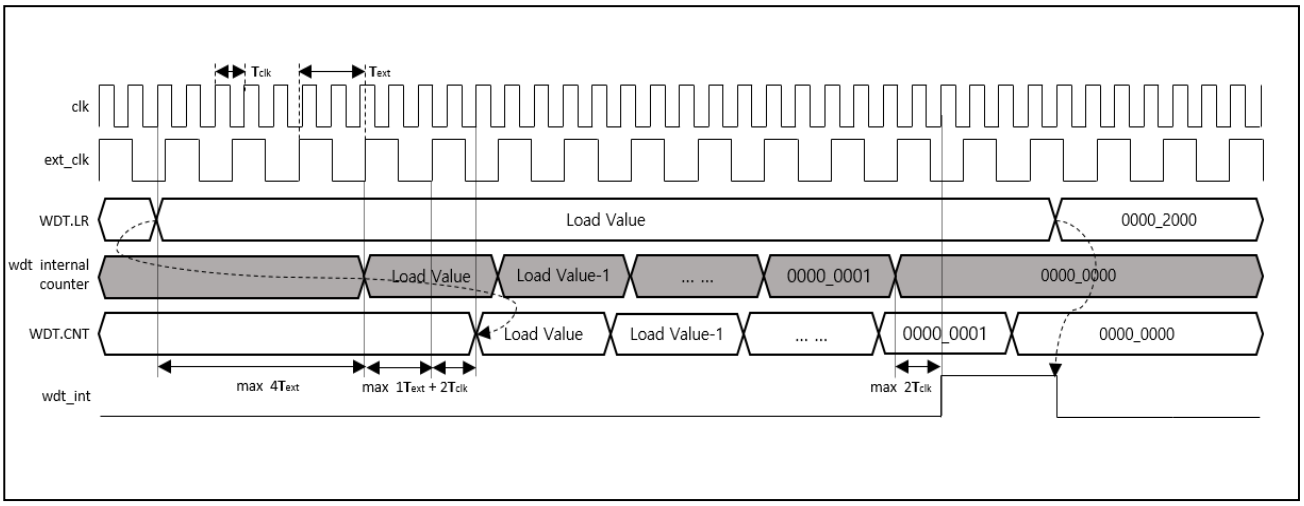


Figure 9-2 Timing Diagram in Interrupt Mode Operation when WDT Clock is External Clock

In WDT interrupt mode, after WDT underflow occurs, a certain count value is reloaded to prevent the next WDT interrupt in a short time period and this reloading action can only be activated when the watchdog timer counter is set to be in Interrupt mode (set WDTIE of WDT.CON). It takes up to 5 cycles from the Load value to the CNT value. The WDT interrupt signal and CNT value data might be delayed by a maximum of 2 system bus clocks in synchronous logic.

Prescale Table

The WDT includes a 32-bit down counter with programmable pre-scaler to define different time-out intervals.

The clock sources of the watchdog timer include the peripheral clock (PCLK) or one of five external clock sources. An external clock source can be enabled by setting CKSEL (WDT.CON[3]) to '1'. The external clock source is chosen in the MCCR3 register of the SCU (system control unit) block.

To make the WDT counter base clock, users can control 3-bit pre-scaler WPRS [2:0] in the WDT.CON register and the maximum pre-scaled value is "clock source frequency/256". The pre-scaled WDT counter clock frequency values are listed in following table.

Selectable clock source (40 kHz ~ 16 MHz) and the time out interval when 1 count

Time out period = {(Load Value) * (1/pre-scaled WDT counter clock frequency) + max 5Text} + max 4Tclk

*Time out period (time out period from load Value to interrupt set '1')

Table 9-3 Pre-scaled WDT Counter Clock Frequency

External Clock Source (WDTCLKIN)	WDTCLKIN	WDTCLKIN /4	WDTCLKIN /8	WDTCLKIN /16	WDTCLKIN /32	WDTCLKIN /64	WDTCLKIN/ 128	WDTCLKIN /256
LSI	40kHz	10kHz	5kHz	2.5kHz	1.25kHz	0.625kHz	0.3125kHz	0.15625kHz
MCLK	Bus clock	MCLK/4	MCLK/8	MCLK/16	MCLK/32	MCLK/32	MCLK/128	MCLK/256
HSI	40MHz	10MHz	5MHz	2.5MHz	1.25MHz	0.625MHz	0.3125MHz	0.15625MHz
MOSC	XTAL frequency (4MHz~ 16MHz)	XTAL/4	XTAL/8	XTAL/16	XTAL/32	XTAL/64	XTAL/128	XTAL/256
SOSC	32.768kHz	8.192kHz	4.096kHz	2.048kHz	1.024kHz	0.512kHz	0.256kHz	0.128kHz

10. 16-Bit Timer

Overview

The timer block consists of 4 channels of 16-bit general purpose timers. These timers have an independent 16-bit counter and dedicated prescaler feed counting clock. They can support periodic timer, PWM pulse, one-shot timer, and Capture mode. They can be synchronized together.

An additional optional free-run timer is provided. The main purpose of this timer is a periodical tick timer or a wake-up source.

- 16-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 10-bit prescaler
- Synchronous start and clear function

Figure 10-1 shows the block diagram of a unit timer block.

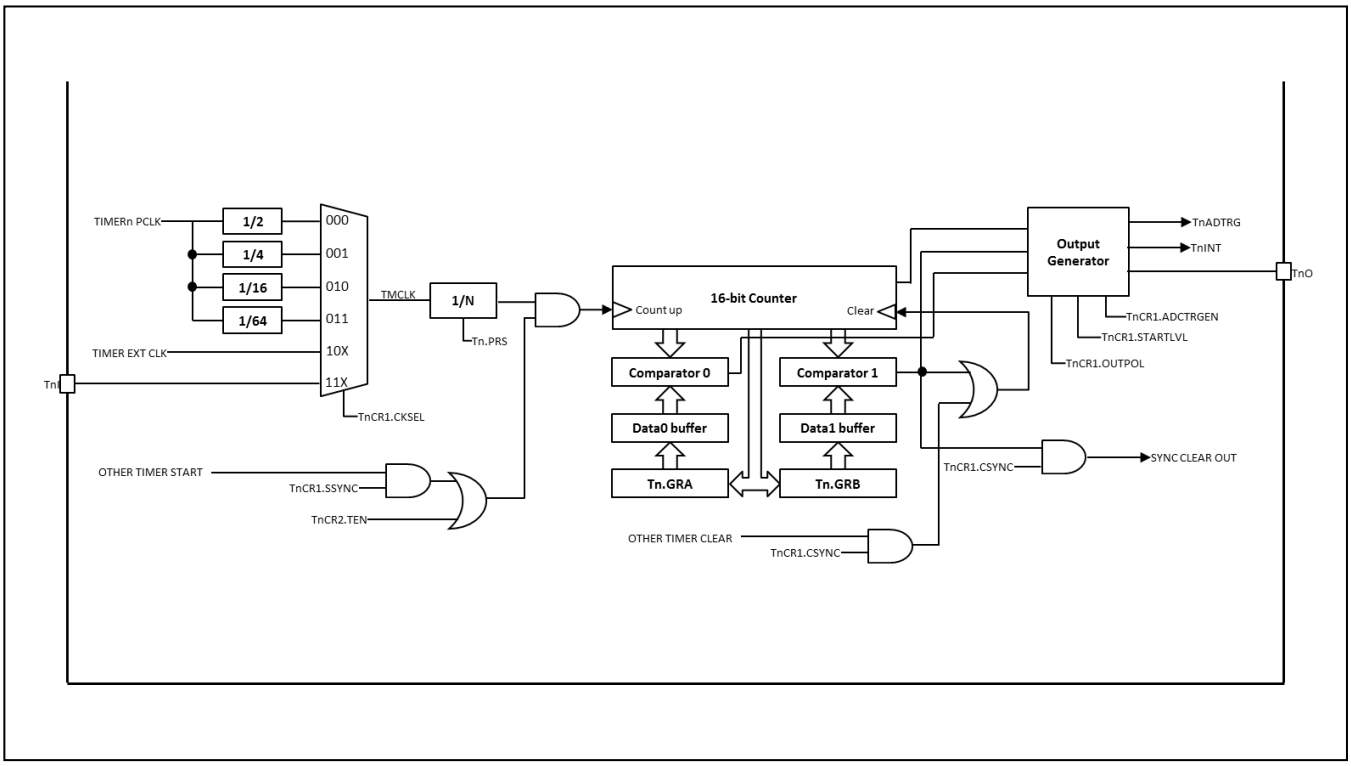


Figure 10-1 Block Diagram

Pin Description

Table 10-1 External Pin

PIN NAME	TYPE	DESCRIPTION
TnIO	I/O	External clock / capture input and PWM/one-shot output

Registers

The base address of the timer is 0x4000_3000 and the register map is described in Table 10-2 and Table 10-3.

Table 10-2 Base Address of Each Channel

NAME	BASE ADDRESS
T0	0x4000_3000
T1	0x4000_3020
T2	0x4000_3040
T3	0x4000_3060

Table 10-3 Timer Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
Tn.CR1	0x--00	RW	Timer control register 1	0x00000000
Tn.CR2	0x--04	RW	Timer control register 2	0x00000000
Tn.PRS	0x--08	RW	Timer prescaler register	0x00000000
Tn.GRA	0x--0C	RW	Timer general data register A	0x00000000
Tn.GRB	0x--10	RW	Timer general data register B	0x00000000
Tn.CNT	0x--14	RW	Timer counter register	0x00000000
Tn.SR	0x--18	RW	Timer status register	0x00000000
Tn.IER	0x--1C	RW	Timer interrupt enable register	0x00000000

Tn.CR1Timer n Control Register 1

The Timer Control register 1 is a16-bit register.

The Timer module should be correctly configured before running. When the target purpose is defined, the timer can be configured in the Tn.CR1 register. After configuring this register, you can start or stop the timer function using the Tn.CR2 register.

T0.CR1=0x4000_3000, T1.CR1=0x4000_3020
T2.CR1=0x4000_3040, T3.CR1=0x4000_3060

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSYNC	CSYNC	UAO	OUTPOL				ADCTRGEN	STARTLVL	CKSEL			CLRMD		MODE	
0	0	0	0	0	0	0	0	0	000			00		00	
RW	RW	RW	RW				RW		RW			RW		RW	

15	SSYNC	Synchronize start counter with other synchronized timers	
		0	Single counter mode
		1	Synchronized counter start mode
14	CSYNC	Synchronize clear counter with other synchronized timers	
		0	Single counter mode
		1	Synchronized counter clear mode
13	UAO	Select GRA, GRB update mode	
		0	Writing GRA or GRB takes effect after current period
		1	Writing GRA or GRB takes effect in current period
12	OUTPOL	Timer output polarity	
		0	Normal output
		1	Negated output
8	ADCTRGEN	ADC Trigger enable control	
		0	Disable adc trigger
		1	Enable adc trigger at same time of GRA match
7	STARTLVL	Timer output polarity control	
		0	Default output level is HIGH
		1	Default output level is LOW
6 4	CKSEL[2:0]	Counter clock source select	
		000	PCLK/2
		001	PCLK/4
		010	PCLK/16
		011	PCLK/64
		10X	MCCR3 clock setting
		11X	TnIO pin input (TnIO pin must be set as input mode)
3 2	CLRMD	Clear select when capture mode	
		00	Rising edge clear mode
		01	Falling edge clear mode
		10	Both edge clear mode
		11	None clear mode
1 0	MODE[1:0]	Timer operation mode control	
		00	Normal periodic operation mode
		01	PWM mode
		10	One shot mode
		11	Capture mode

Tn.CR2Timer n Control Register 2

Timer Control Register 2 is an 8-bit register.

T0.CR2=0x4000_3004, T1.CR2=0x4000_3024
T2.CR2=0x4000_3044, T3.CR2=0x4000_3064

7	6	5	4	3	2	1	0
PWMO						TCLR	TEN
0	0	0	0	0	0	0	0
R	R	R	R	R	R	WO	RW

7	PWMO	PWM output for read	
1	TCLR	Timer register clear	
		0	Normal operation
		1	Clear count register. (This bit will be cleared after next timer clock)
0	TEN	Timer enable bit	
		0	Stop timer counting
		1	Start timer counting

Note: It is recommended to start timer with TCLR bit set to ‘1’.

Tn.PRS Timer n Prescaler Register

The Timer Prescaler register is a 16-bit register to prescale the counter input clock.

T0.PRS=0x4000_3008, T1.PRS=0x4000_3028
T2.PRS =0x4000_3048, T3.PRS=0x4000_3068

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PRS									
0	0	0	0	0	0	000									
						RW									

9	PRS	Pre-scale value of count clock	
0		TCLK = CLOCK_IN/(PRS+1)	
		(*CLOCK_IN* is a selected timer input clock)	

Tn.GRA Timer n General Register A

The Timer General Register A is a 16-bit register.

T0.GRA=0x4000_300C, T1.GRA=0x4000_302C
T2.GRA =0x4000_304C, T3.GRA=0x4000_306C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GRA															
0x0000															
RW															

15 0	GRA	General Register A (Duty/Interrupt Register) Periodic mode / PWM / One-shot mode - In PWM mode this register is used as duty value. - When the counter value is matched with this value, GRA Match interrupt is requested Capture mode - Falling edge of TnIO port will capture the count value when rising edge clear mode - Rising edge of TnIO port will capture the count value when falling edge clear mode

Tn.GRB Timer n General Register B

The Timer General Register B is 16-bit register.

T0.GRB=0x4000_3010, T1.GRB=0x4000_3030
T2.GRB=0x4000_3050, T3.GRB=0x4000_3070

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GRB															
0x0000															
RW															

15 0	GRB	General Register B (Period Register) Periodic mode / PWM / One-shot mode - In periodic mode or PWM mode, this register is used as Period value. The counter will count up to (GRB-1) value. - When the counter value is matched with this value, GRB Match interrupt is requested only in PWM and one-shot modes. Capture mode - Rising edge of TnIO port will capture the count value when rising edge clear mode - Falling edge of TnIO port will capture the count value when falling edge clear mode

Tn.CNT Timer n Count Register

The Timer Count register is a 16-bit register.

T0.CNT=0x4000_3014, T1.CNT=0x4000_3034
T2.CNT=0x4000_3054, T3.CNT=0x4000_3074

31	30	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRECLR															
0x0000															
W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
0x0000															
RW															

31	PRECLR	Prescaler initialize when timer count value write operation	
16		0x00	Prescaler will be initialized when write timer count value on Tn.CNT[15:0]. After writing the count value, prescaler restarted from initial state to make accurate period for first count.
		0xFF	Prescaler will not be initialized and maintain current conditions even writing timer count value on Tn.CNT[15:0]. First count period is not accurate depends on its status when writing operation.
15	CNT	Timer count value register	
0		R	Read current timer count value
		W	Set count value

Tn.SR Timer n Status Register

The Timer Status register is an 8-bit register. This register indicates the current status of the timer module.

T0.SR=0x4000_3018, T1.SR=0x4000_3038
T2.SR=0x4000_3058, T3.SR=0x4000_3078

7	6	5	4	3	2	1	0
					MFA	MFB	OVF
0	0	0	0	0	0	0	0
					RW	RW	RW

2	MFA	GRA Match flag	
		0	No direction change
		1	Match flag with GRA
1	MFB	GRB Match flag	
		0	No direction change
		1	Match flag with GRB
0	OVF	Counter overflow flag	
		0	No direction change
		1	Counter overflow flag

Tn.IER Timer n Interrupt Enable Register

The Timer Interrupt Enable register is an 8-bit register.

Each status flag of the timer block can issue the interrupt. To enable the interrupt, write ‘1’ in the corresponding bit in the Tn.IER register.

T0.IER=0x4000_301C, T1.IER=0x4000_303C
T2.IER=0x4000_305C, T3.IER=0x4000_307C

7	6	5	4	3	2	1	0
					MAIE	MBIE	OVIE
0	0	0	0	0	0	0	0
					RW	RW	RW

2	MAIE	GRA Match interrupt enable	
		0	Not effect
		1	Enable match register A interrupt
1	MBIE	GRB Match interrupt enable	
		0	Not effect
		1	Enable match register B interrupt
0	OVIE	Counter overflow interrupt enable	
		0	Not effect
		1	Enable counter overflow interrupt

Functional Description

Timer Basic Operation

TMCLK in Figure 10-2 is a reference clock for operation of the timer. This clock is divided by the prescaler setting for the counting clock to work. Figure 10-2 shows the starting point of the counter and the ending of the period point of the counter in normal periodic mode.

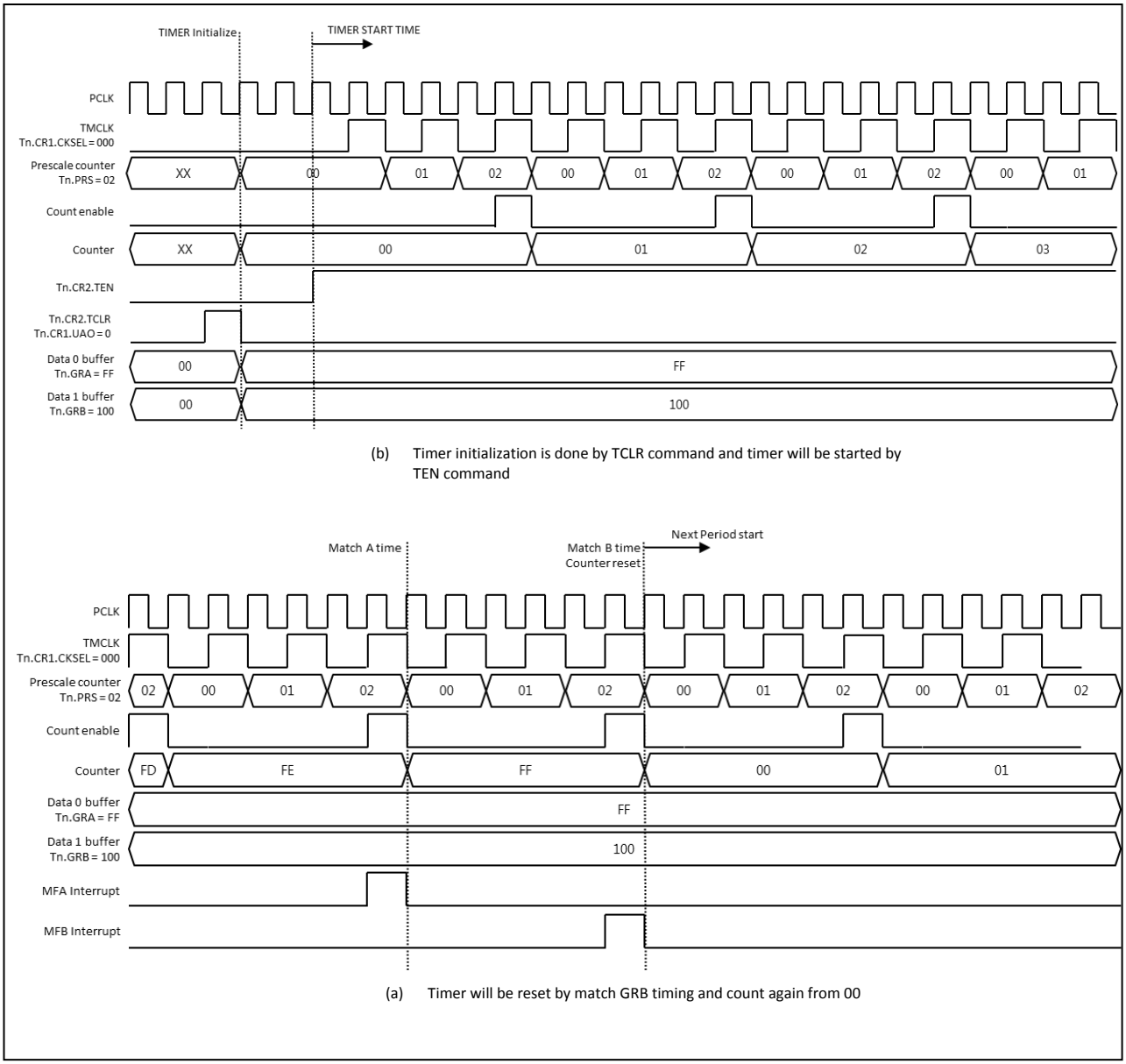


Figure 10-2 Basic Start and Match Operation

The period of timer count can be calculated using the following equation:

The period = TMCLK Period * Tn.GRB value.

Match A interrupt time = TMCLK Period * Tn.GRA value.

If the Tn.CR1.UAO bit is '0', the Tn.CR2.TCLR command will initialize all the registers in the timer block and load the GRA and GRB value into the Data0 and Data1 buffer. When you change the timer setting and restart the timer with the new setting, it is recommended that you write the Tn.CR2.TCLR command before the Tn.CR2.TEN command.

The update timing of the Data0 and Data1 buffers in dynamic operation is different in each operating mode and depends on the Tn.CR1.UAO bit.

Normal Periodic Mode

Figure 10-3 shows the timing diagram in normal periodic mode. The Tn.GRB value decides the timer period. One more comparison point is provided with the Tn.GRA register value.

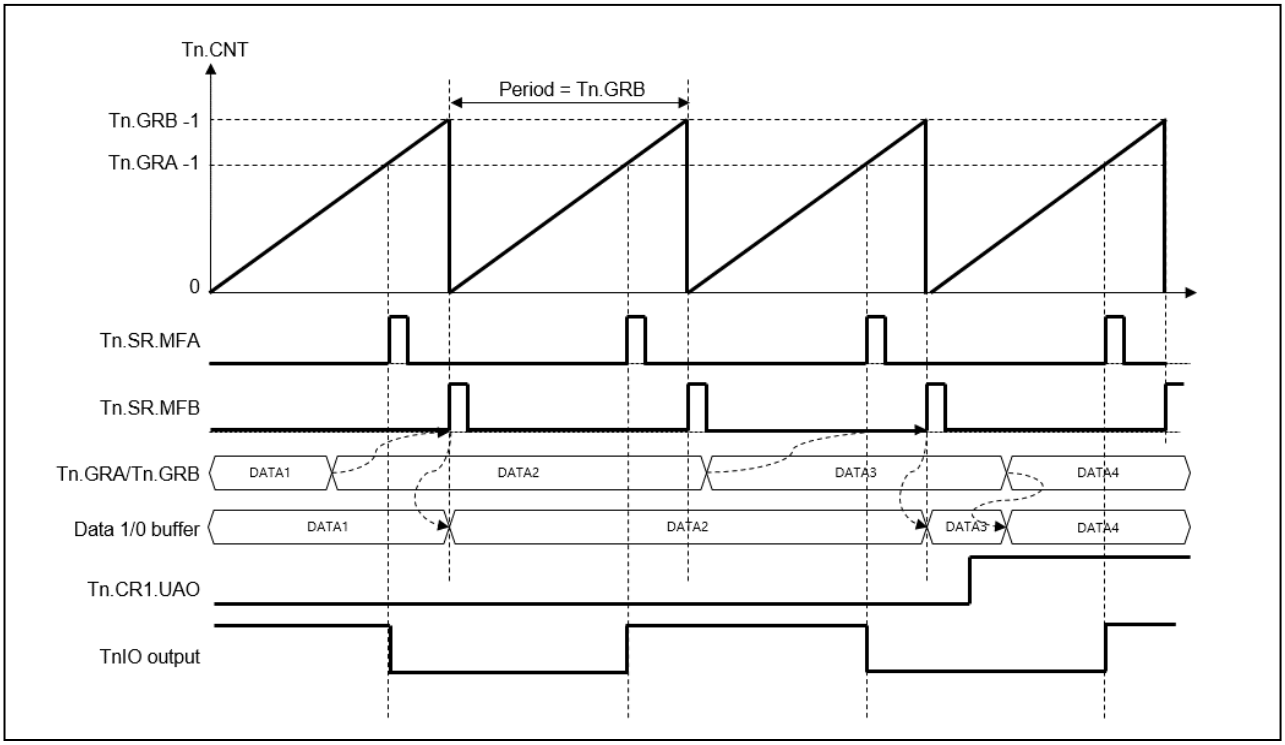


Figure 10-3 Normal Periodic Mode Operation

The period of timer count can be calculated using the following equation:

The period = TMCLK Period * Tn.GRB value.

Match A interrupt time = TMCLK Period * Tn.GRA value.

If Tn.GRB = 0, the timer cannot be started even if Tn.CR2.TEN is "1" because the period is "0".

The value in Tn.GRA and Tn.GRB is loaded into the internal compare data buffer 0 and 1 when the loading condition occurs. In this periodic mode with Tn.CR1.UAO =0, the Tn.CR2.TCLR write operation will load the data buffer and the next GRB match event will load the data buffer.

When Tn.CR1.UAO is 1, the internal comparison data buffer is updated whenever the Tn.GRA or Tn.GRB data is updated.

The TnIO output signal will be toggled at every Match A condition. If the Tn.GRA is 0 value, the TnIO output does not change its previous level. If Tn.GRA is the same as Tn.GRB, the TnIO output will toggle at the same time as the counter start time. The initial level of the TnIO signal is decided by the Tn.CR1.STARTLVL value.

One Shot Mode

Figure 10-4 shows the timing diagram in one shot mode. The Tn.GRB value decides the one shot period. An additional comparison point is provided with Tn.GRA register value.

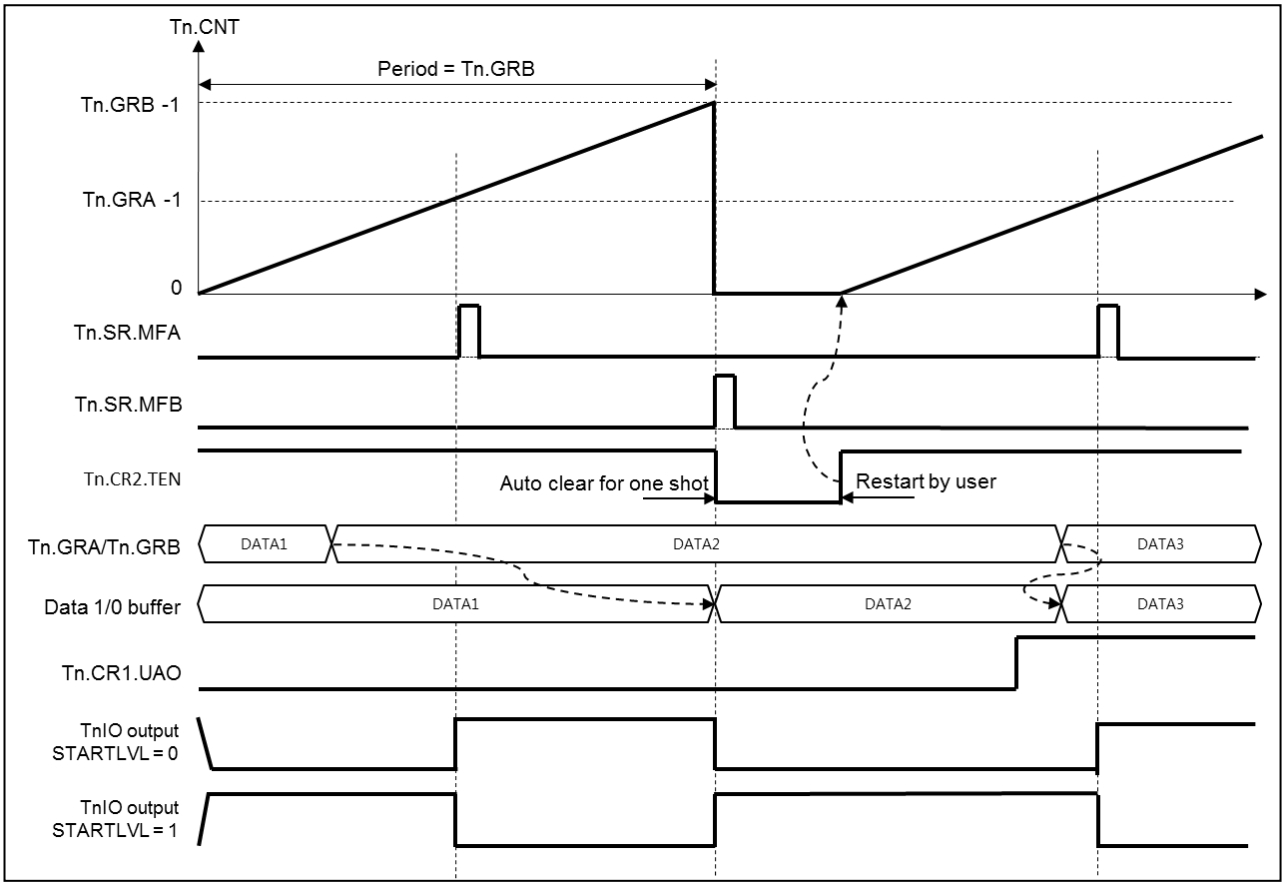


Figure 10-4 One Shot Mode Operation

The period of one shot count can be calculated using the following equation:

The period = TMCLK Period * Tn.GRB value

Match A interrupt time = TMCLK Period * Tn.GRA value

If Tn.GRB = 0, the timer cannot be started even if Tn.CR2.TEN is “1” because the period is “0”.

The value in Tn.GRA and Tn.GRB is loaded into internal compare data buffers 0 and 1 when the loading condition occurs. In this periodic mode with Tn.CR1.UAO =0, the Tn.CR2.TCLR write operation loads the data buffer and the next GRB match event will load the data buffer.

When Tn.CR1.UAO is 1, the internal compare data buffer is updated whenever the Tn.GRA or Tn.GRB data is updated. The TnIO output signal format is the same as PWM mode. The Tn.GRB value defines the output pulse period and the Tn.GRA value defines the pulse width of one shot pulse.

PWM Timer Output Examples

Figure 10-5 shows the timing diagram in PWM output mode. The Tn.GRB value decides the PWM pulse period. An additional comparison point is provided with the Tn.GRA register value which defines the pulse width of PWM output.

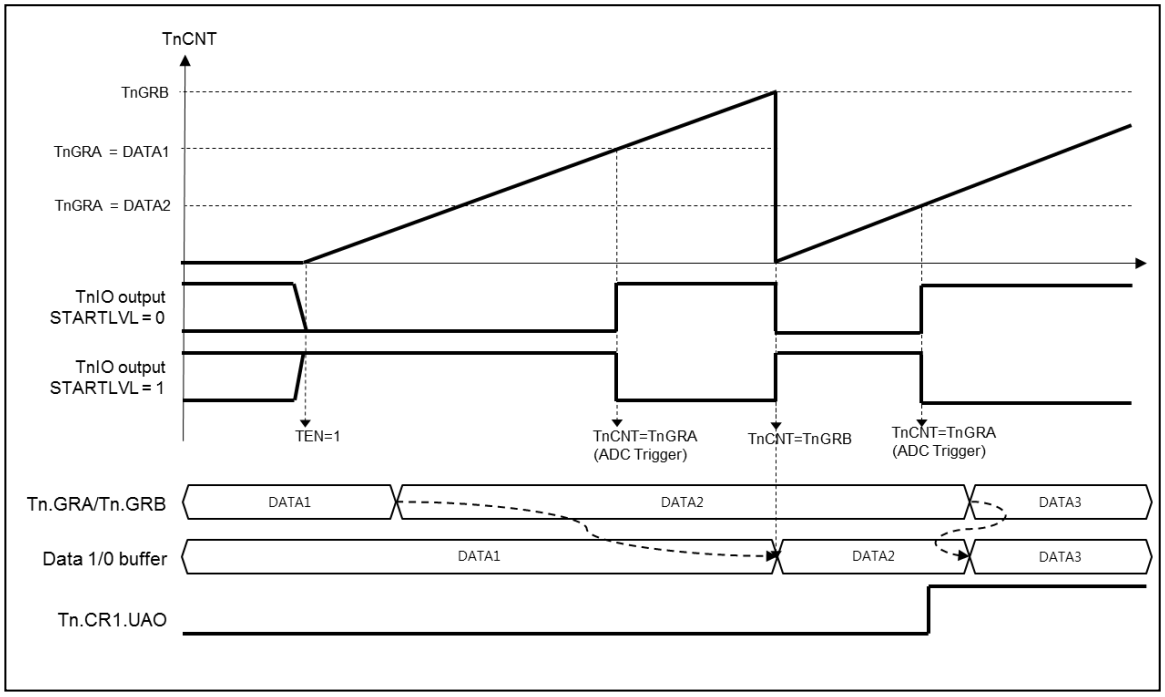


Figure 10-5 PWM Output Example

The period of PWM pulse can be calculated using the following equation:

The period = TMCLK Period * Tn.GRB value.

Match A interrupt time = TMCLK Period * Tn.GRA value.

If Tn.GRB = 0, the timer cannot be started even if Tn.CR2.TEN is “1” because the period is “0”.

The value in Tn.GRA and Tn.GRB is loaded into internal compare data buffers 0 and 1 when the loading condition occurs. In this periodic mode with Tn.CR1.UAO =0, the Tn.CR2.TCLR write operation loads the data buffer and the next GRB match event will load the data buffer.

When Tn.CR1.UAO is 1, the internal compare data buffer is updated whenever the Tn.GRA or Tn.GRB data is updated. The TnIO output signal generates a PWM pulse. The Tn.GRB value defines the output pulse period and the Tn.GRA value defines the pulse width of one shot pulse. The active level of PWM pulse can be controlled by the Tn.CR1.STARTLVL bit value.

ADC Trigger generation is available at Match A interrupt time.

PWM Synchronization Function

Two PWM outputs are usually used as synchronous PWM signal control. This function is provided with synchronous start function. Figure 10-6 shows the synchronous PWM generation function.

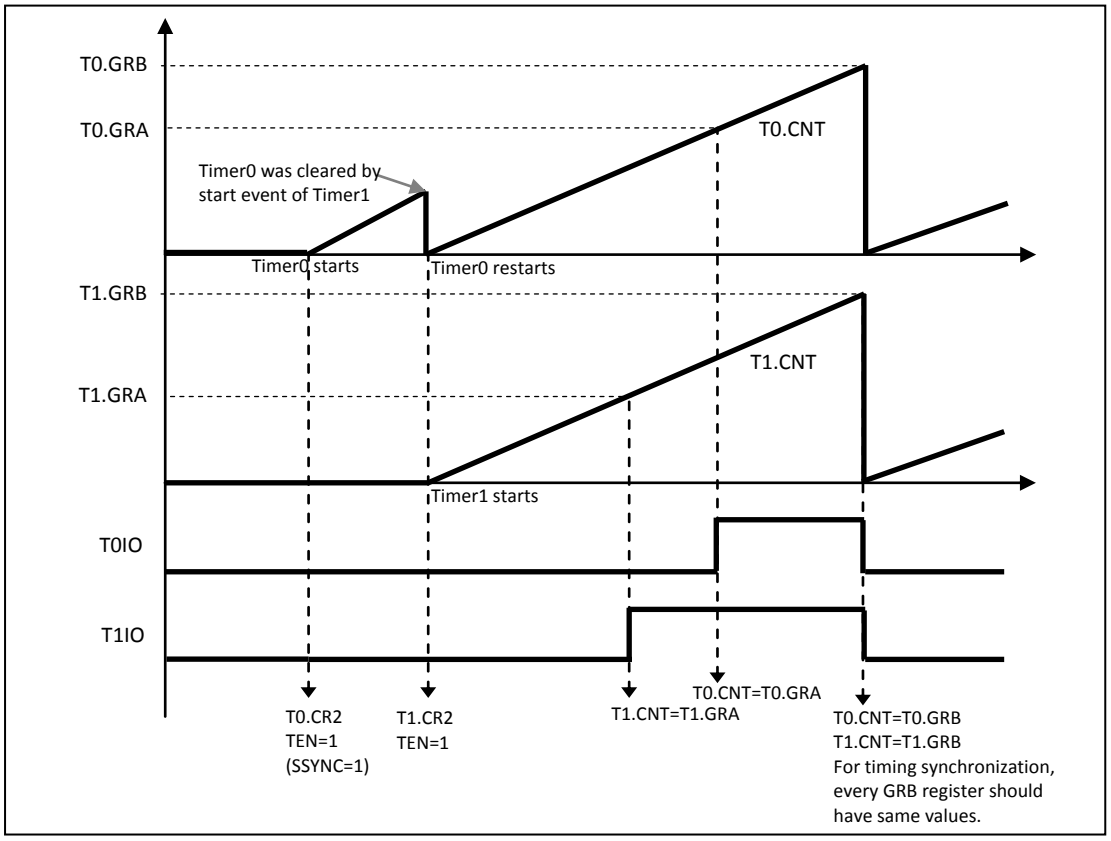


Figure 10-6 A Example of Timer Synchronization Function (SSYNC=1)

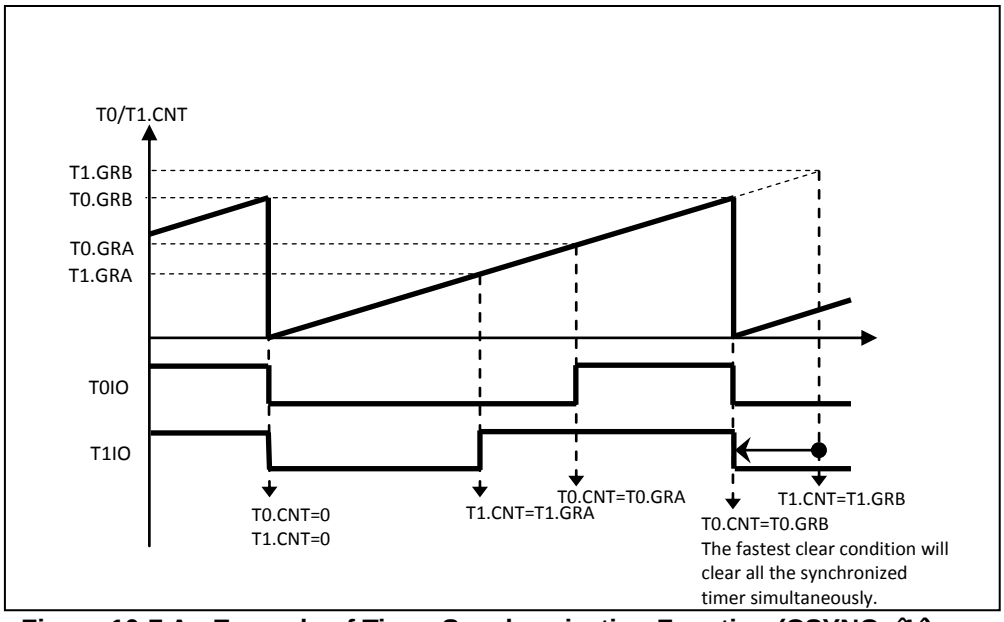


Figure 10-7 An Example of Timer Synchronization Function (CSYNC=1)

The Tn.CR1.SSYNC bit controls start synchronization with other timer blocks. The Tn.CR1.CSYCN bit controls clear synchronization with other timer blocks. This bit is only effective if there are at least 2 additional timers with the sync control bits set.

For example, timer0 and timer1 set the SSYNC and CCSYNC bits in each CR1 register; both timers start when one of them is enabled. Both timers will be cleared with a short period match value. However, others are not affected by these 2 timers, and they can be operated independently because their SYNC control bit is 0.

Capture Mode

Figure 10-8 shows the timing diagram in Capture mode operation. The TnIO input signal is used for capture pulse. The rising and falling edges can capture the counter value in each capture condition.

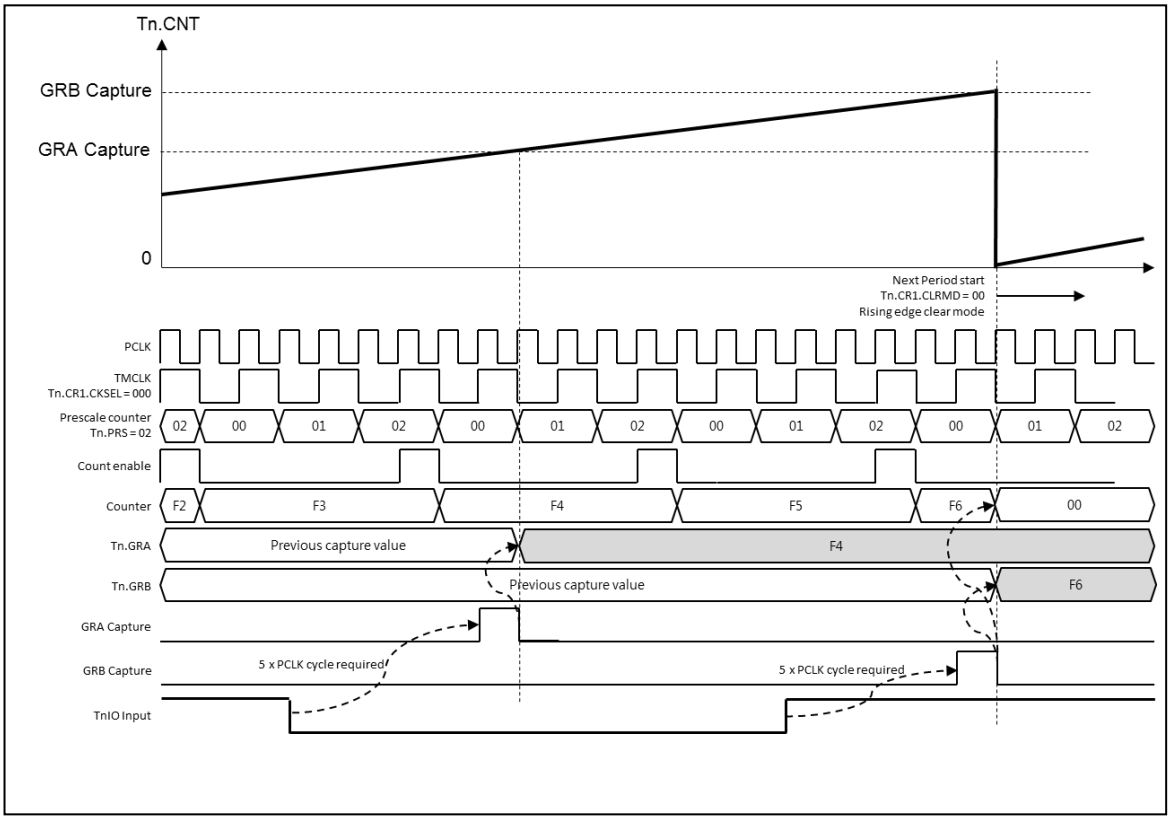


Figure 10-8 Capture Mode Timing Diagram

A 5 PCLK clock cycle is required internally. Therefore, the actual capture point is after 5 PCLK clock cycles from the rising or falling edge of the TnIO input signal.

The internal counter can be cleared in various modes. The Tn.CR1.CLRMD field controls the counter clear mode. Rising edge clear mode, falling edge clear mode, both edges clear mode and no clear mode are supported.

Figure 10-8 shows an instance of rising edge clear mode.

ADC Trigger Function

The Timer module can generate ADC start trigger signals. One timer can be one trigger source of the ADC block. Trigger source control is accomplished by the ADC control register. Figure 10-9 shows the ADC trigger function.

The conversion rate must be shorter than the timer period to prevent occurrence of an overrun situation. ADC acknowledge is not required because the trigger signal will be cleared automatically after 3 PCLK clock pulses.

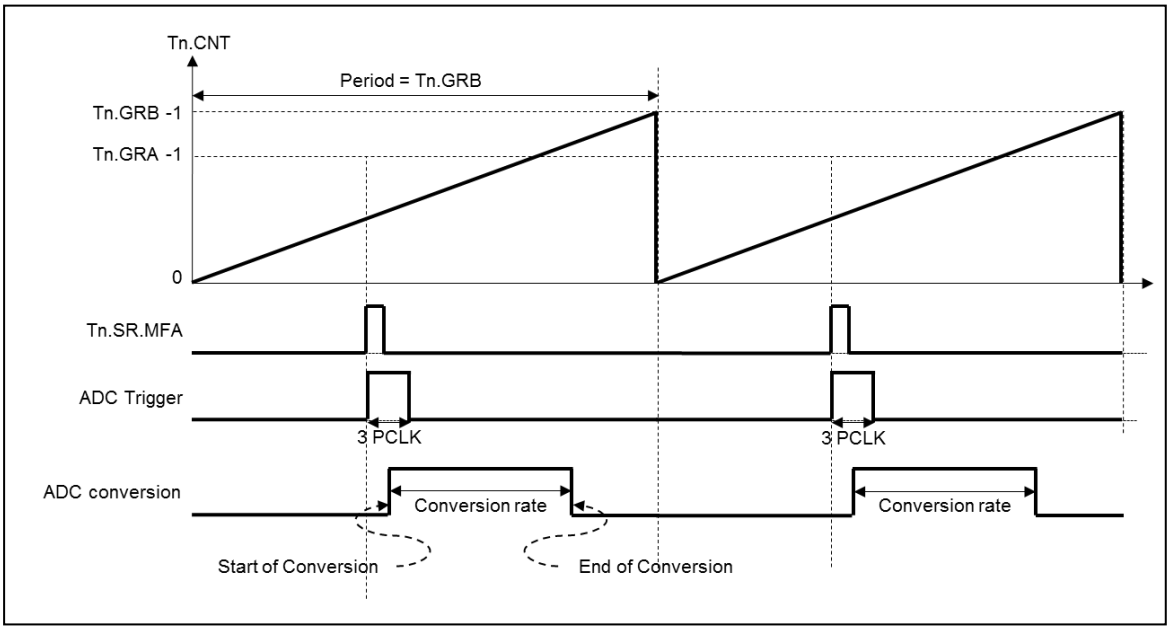


Figure 10-9 ADC Trigger Function Timing Diagram

11. Free Run Timer (FRT)

Overview

The FRT block is a 32-bit Free Run Timer. It can be used in Power-down Mode.

- 32-bit up-counter with SOSC, MOSC, LSI
- Matched Interrupt

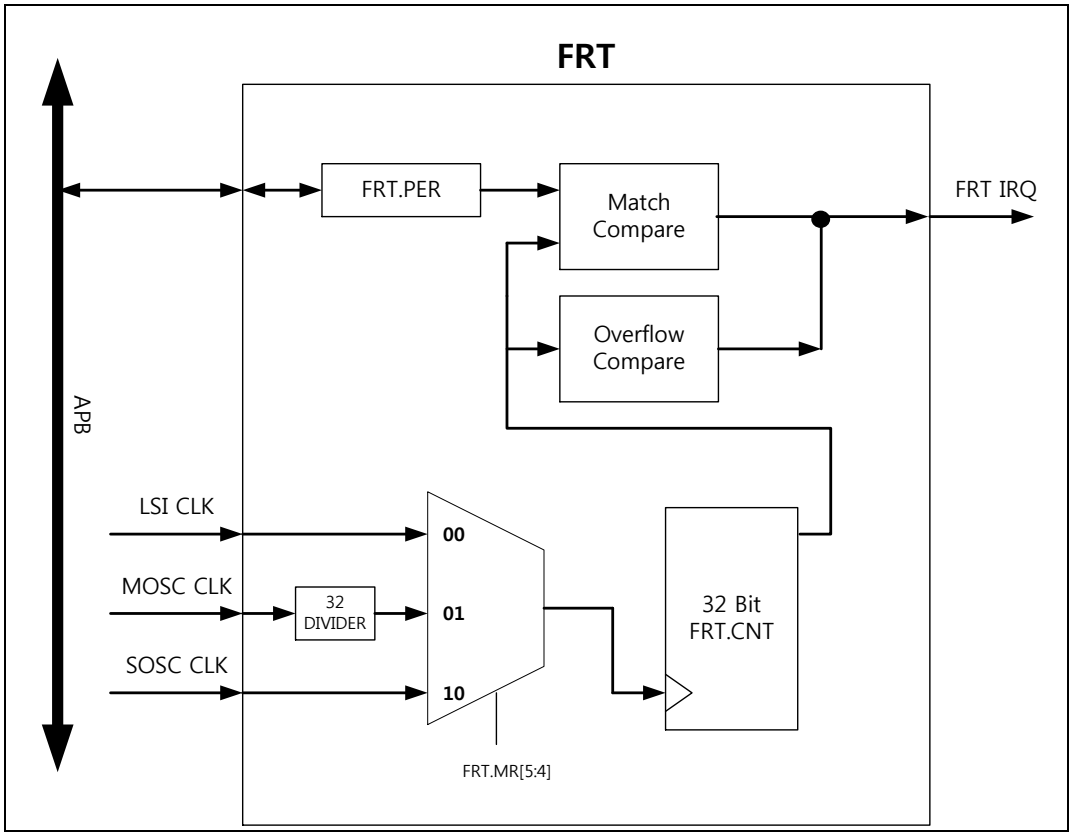


Figure 11-1 FRT Block Diagram

Registers

The base address of FRT is 0x4000_0600 and the register map is described in Table 11-1 and Table 11-2.

Table 11-1 Base Address of Channel

NAME	BASE ADDRESS
FRT	0x4000_0600

Table 11-2 FRT Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
FRT.MR	0x0000	RW	FRT mode register	0x00000000
FRT.CR	0x0004	RW	FRT control register	0x00000000
FRT.PER	0x0008	RW	FRT period match register	0x00000000
FRT.CNT	0x000C	RO	FRT counter register	0x00000000
FRT.SR	0x0010	RW	FRT status register	0x00000000

FRT.MR FRT Mode Register

FRT is a 32-bit up counter. It can be used in Power Down mode when using SUB OSC. The SUB OSC clock is directly connected to FRT. This is an 8-bit register.

FRT.MR=0x4000_0600

7	6	5	4	3	2	1	0
		CLKSEL			MCD	OVIE	MIE
0	0	0	0		0	0	0
		RW	RW		RW	RW	RW

5 4	CLKSEL	FRT counter clock source control	
		0	Low Speed Internal Oscillator clock (40kHz)
		1	External Oscillator clock divided by 32
		2	Sub Oscillator clock
		3	Reserved
2	MCD	Counter Match Clear Disable bit	
		0	Counter Match Clear function is enabled. Whenever the counter matches FRT.PER, the counter will be set zero and waiting for MF to be cleared.
		1	Counter Match Clear function is disabled. The counter will keep countering without set zero
1	OVIE	Over Flow Interrupt Enable bit	
		0	Not effect
		1	Interrupt enabled
0	MIE	Match Interrupt Enable bit	
		0	Not effect
		1	Interrupt enabled

FRT.CR FRT Control Register

The FRT Control Register is an 8-bit register.

FRT.CR=0x4000_0604

7	6	5	4	3	2	1	0
				RREQ	CLR	HOLD	EN
0	0	0	0	0	0	0	0
				RW	WO	RW	RW

3	RREQ	FRT Counter read request bit
		0 No action
		1 Request to read FRTn.CNT (cleared when CNTACK(FSR[1]) is high)
2	CLR	FRT Counter register clear bit
		0 No action
		1 Clear the counter
1	HOLD	FRT Counter register hold bit
		0 No action
		1 Hold the counter
0	EN	FRT enable bit
		0 FRT Disabled
		1 FRT Enabled

FRT.PER FRT Period Match Register

The FRT Period Match Register is a 32-bit register.

FRT.PER=0x4000_0608

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
PERIOD			
0x0000_0000			
RW			

32 0	PERIOD	FRT Period Match Data

FRT.CNT FRT Counter Register

The FRT Counter Register is a 32-bit register.

FRT.CNT=0x4000_060C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT																															
0x0000_0000																															
RO																															

32 0	CNT		FRT Counter

FRT.SRFRT Status Register

The FRT Status Register is an 8-bit register.

FRT.SR=0x4000_0610

7	6	5	4	3	2	1	0
					RACK	OVF	MF
0	0	0	0	0	0	0	0
					WC1	WC1	WC1

2	RACK	Read Counter Acknowledge bit	
		0	Not ready to read CNT value
		1	Ready to read CNT value
1	OVF	OverFlow Interrupt flag bit	
		0	Overflow interrupt did not occur
		1	Overflow interrupt occurred
0	MF	Interrupt flag bit	
		0	Match interrupt did not occur.
		1	Match Interrupt occurred In Counter Match Clear mode, this bit should be cleared for restarting the counter.

Functional Description

The Free Run Timer has two types of interrupts – overflow and match interrupts.

Match Interrupt Operation

The match interrupt timing diagram is shown in Figure 11-2. FRT.MR.MIE should be set as '1' for using the match-interrupt.

The FRT clock starts the FRT counter after FRT.CR.EN is '1'. Interrupt and wakeup signals occur when the counter is matched with the value of FRT.PER. The 'interrupt' signal might be delayed by a maximum of 2 system clocks and the 'wakeup' signal might be delayed by a maximum of (1 clk + 2 frt_clk).

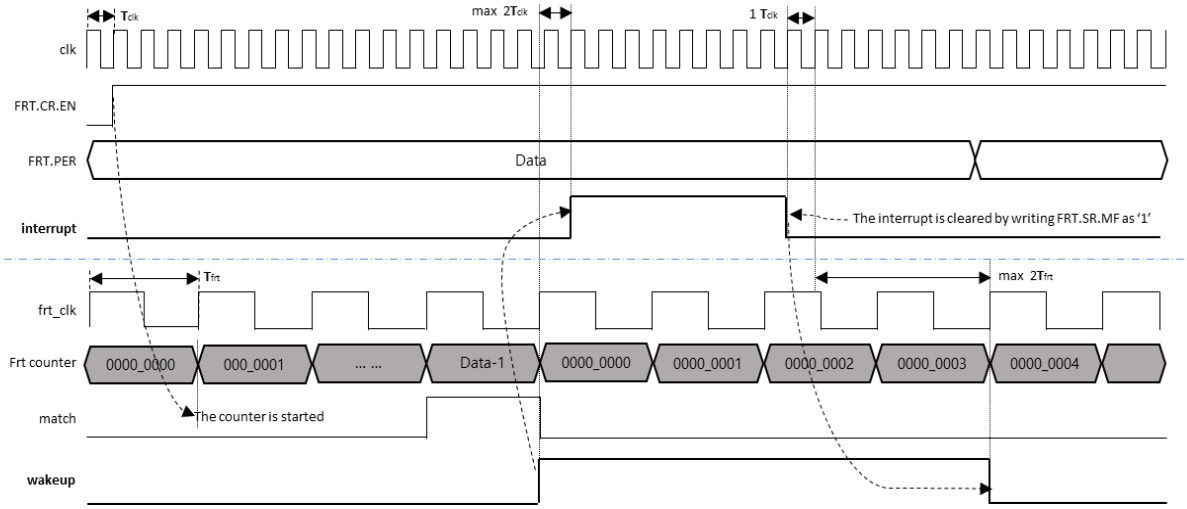


Figure 11-2 Match Interrupt Operation Timing Diagram

Overflow Interrupt Operation

The overflow interrupt timing diagram is shown in Figure 11-3. The overflow-interrupt operation is similar to the match interrupt operation. The overflow interrupt is started to set when the FRT counter matches 0xFFFFFFFF.

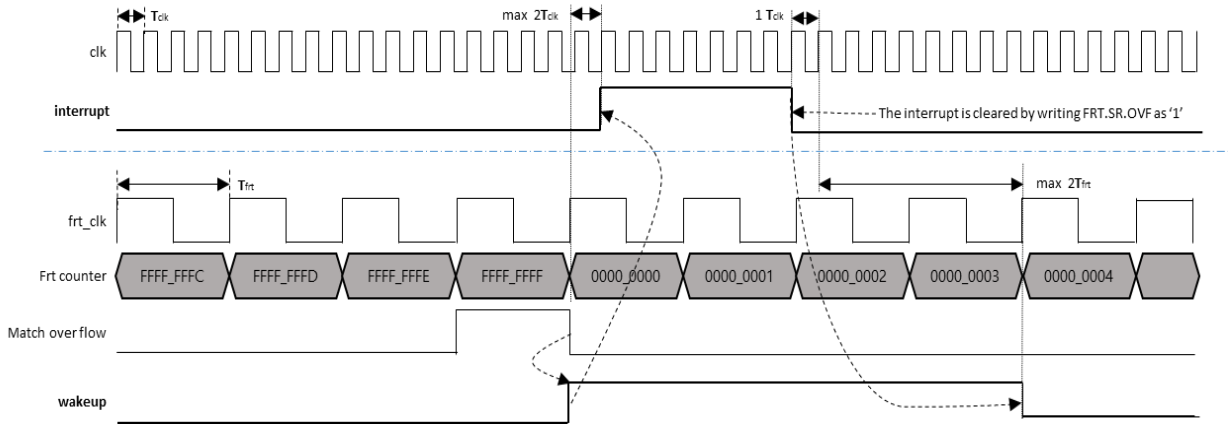


Figure 11-3 Timing Diagram in Overflow Interrupt Operation

12. Universal Asynchronous Receiver/Transmitter (UART)

Overview

2-channel Universal Asynchronous Receiver/Transmitter (UART) modules are provided. The UART operation status including error status can be read from status register. The prescaler, which generates proper baud rate, exists for each UART channel. The prescaler can divide the UART clock source which is PCLK, from 1 to 65535. The baud rate is generated by the clock which is internally divided by 16 of the prescaled clock and 8-bit precision clock tuning function.

Programmable interrupt generation function helps control communication via the UART channel.

- Compatible with 16450
- Standard asynchronous control bit (start, stop, and parity) configurable
- Programmable 16-bit fractional baud generator
- Programmable serial communication
  - 5-, 6-, 7,- or 8- bit data transfer
  - Even, odd, or no-parity bit insertion and detection
  - 1-, 1.5,- or 2-stop bit-insertion and detection
- 16-bit baud rate generation with 8-bit fraction control
- Hardware inter-frame delay function
- Stop bit error detection
- Detail status register

Note: You must set the MCCR4 Register in the SCU before using the UART!

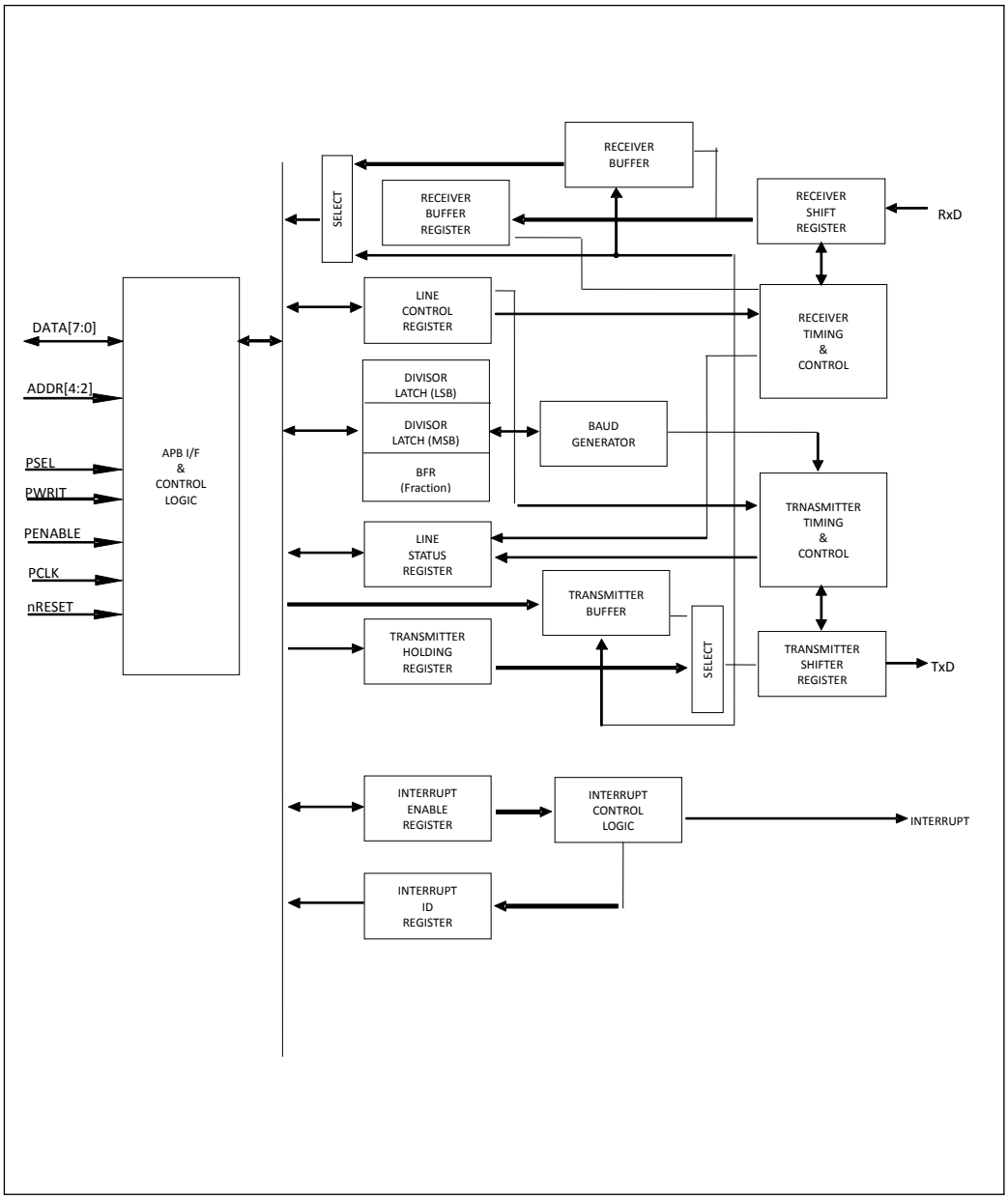


Figure 12-1 Block Diagram

Pin Description

Table 12-1 External Signal

PIN NAME	TYPE	DESCRIPTION
TXD0	O	UART Channel 0 transmit output
RXD0	I	UART Channel 0 receive input
TXD1	O	UART Channel 1 transmit output
RXD1	I	UART Channel 1 receive input

Registers

The base address of UART is 0x4000_8000 and the register map is described in Table 12-2 and Table 12-3.

Table 12-2 Base Address of Each Port

NAME	BASE ADDRESS
U0	0x4000_8000
U1	0x4000_8100

Table 12-3 UART Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
Un.RBR	0x00	R	Receive Data Buffer Register	0x00
Un.THR	0x00	W	Transmit Data Hold Register	0x00
Un.IER	0x04	RW	Interrupt Enable Register	0x00
Un.IIR	0x08	R	Interrupt ID Register	0x01
-	0x08	-	Reserved	-
Un.LCR	0x0C	RW	Line Control Register	0x00
Un.DCR	0x10	RW	Data Control Register	
Un.LSR	0x14	R	Line Status Register	0x00
-	0x18	-	Reserved	-
Un.SCR	0x1C	RW	Scratch Pad Register	0x00
Un.BDR	0x20	RW	Baud rate Divisor Latch Register	0x0000
Un.BFR	0x24	RW	Baud rate Fractional Counter Value	0x00
Un.IDTR	0x30	RW	Inter-frame Delay Time Register	0x80

Un.RBR Receive Buffer Register

The UART Receive Buffer register is an 8-bit read-only register. Received data is read out from this register. The maximum length of data is 8 bits. The last data received will be maintained in this register until a new byte is received.

U0.RBR=0x4000_8000, U1.RBR=0x4000_8100

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

7 0	RBR	Receive Buffer Register

Un.THR Transmit Data Hold Register

The UART Transmit Data Hold register is an 8-bit write-only register. The data for transmit can be stored in this register. But the write data cannot be read from this register. The data which is written in the Un.THR register will be transferred into the Transmit Shifter register whenever the Transmit Shifter register is empty.

U0.THR=0x4000_8000, U1.THR=0x4000_8100

7	6	5	4	3	2	1	0
THR							
-							
WO							

7 0	THR	Transmit Data Hold Register

Un.IER UART Interrupt Enable Register

The UART Interrupt Enable register is an 8-bit register.

U0.IER=0x4000_8004, U1.IER=0x4000_8104

7	6	5	4	3	2	1	0
-	-			-	RLSIE	THREIE	DRIE
0	0	0	0	0	0	0	0
					RW	RW	RW

2	RLSIE	Receiver line status interrupt enable	
		0	Receive line status interrupt is disabled.
		1	Receive line status interrupt is enabled
1	THREIE	Transmit holding register empty interrupt enable	
		0	Transmit holding register empty interrupt is disabled
		1	Transmit holding register empty interrupt is enabled
0	DRIE	Data receive interrupt enable	
		0	Data receive interrupt is disabled
		1	Data receive interrupt is enabled

Un.IIR UART Interrupt ID Register

The UART Interrupt ID register is an 8-bit register.

U0.IIR=0x4000_8008, U1.IIR=0x4000_8108

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

18	RLS	Receiver line status flag (Error)	
17	THRE	Transmit holding register empty flag	
16	DR	Data receive interrupt flag	
4	TXE	Interrupt source ID See interrupt source ID table	
3 1	IID	Interrupt source ID See interrupt source ID table	
0	IPEN	Interrupt pending bit	
		0	Interrupt is pending
		1	No interrupt is pending.

The UART supports 3-priority interrupt generation and the Interrupt Source ID register shows one interrupt source which has the highest priority among pending interrupts. The priority is defined as:

- Receive line status interrupt
- Receive data ready interrupt/ Character timeout interrupt
- Transmit hold register empty interrupt

Table 12-4 Interrupt ID and Control

Priority	TXE	IID		IPEN	Interrupt Sources		
	Bit 4	Bit 2	Bit 1	Bit 0	Interrupt	Interrupt Condition	Interrupt Clear
-	0	0	0	1	None	-	-
1	0	1	1	0	Receiver Line Status	Overrun, Parity, Framing or Break Error	Read LSR register
2	0	1	0	0	Receiver Data Available	Receive data is available.	Read receive register or read IIR register
3	0	0	1	0	Transmitter Holding Register Empty	Transmit buffer empty	Write transmit hold register or read IIR register
4	1	X	X	X	Transmitter Register Empty	Transmit register empty	Write transmit hold register or read IIR register

Un.LCR UART Line Control Register

The UART Line Control register is an 8-bit register.

U0.LCR=0x4000_800C, U1.LCR=0x4000_810C

7	6	5	4	3	2	1	0
	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN[1:0]	
0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW

6	BREAK	When this bit is set, TxD pin will be driven at low state in order to notice the alert to the receiver.
	0	Normal transfer mode
	1	Break transmit mode
5	STICKP	Force parity and it will be effective when PEN bit is set.
	0	Parity stuck is disabled
	1	Parity stuck is enabled and parity always the bit of PARITY.
4	PARITY	Parity mode selection bit and stuck parity select bit
	0	Odd parity mode
	1	Even parity mode
3	PEN	Parity bit transfer enable
	0	The parity bit disabled
	1	The parity bit enabled
2	STOPBIT	The number of stop bit followed by data bits.
	0	1 stop bit
	1	1.5 / 2 stop bit In case of 5 bit data case, 1.5 stop bit is added. In case of 6,7 or 8 bit data, 2 stop bit is added
1 0	DLEN	The data length in one transfer word.
	00	5 bit data
	01	6 bit data
	10	7 bit data
	11	8 bit data

Parity bit will be generated according to bit 3,4,5 of Un.LCR register. Table 12-5 shows the variation of parity bit generation.

Table 12-5 Interrupt ID and Control

STICKP	PARITY	PEN	Parity
X	X	0	No Parity
0	0	1	Odd Parity
0	1	1	Even Parity
1	0	1	Force parity as “1”
1	1	1	Force parity as “0”

Un.DCR UART Data Control Register

The UART Data Control register is an 8-bit register. The inversion function of Tx or Rx data line, is controlled by this Un.DCR register. When the corresponding bit is set to 1, the data line of Tx or RX signal will be inverted.

U0.DCR=0x4000_8010, U1.DCR=0x4000_8110

7	6	5	4	3	2	1	0
			LBON	RXINV	TXINV		
0	0	0	0	0	0	0	0
			RW	RW	RW		

4	LBON	Local loopback test mode enable	
		0	Normal mode
		1	Local loopback mode (TxD connected to RxD internally)
3	RXINV	Rx Data Inversion Selection	
		0	Normal RxData Input
		1	Inverted RxData Input
2	TXINV	Tx Data Inversion Selection	
		0	Normal TxData Output
		1	Inverted TxData Output

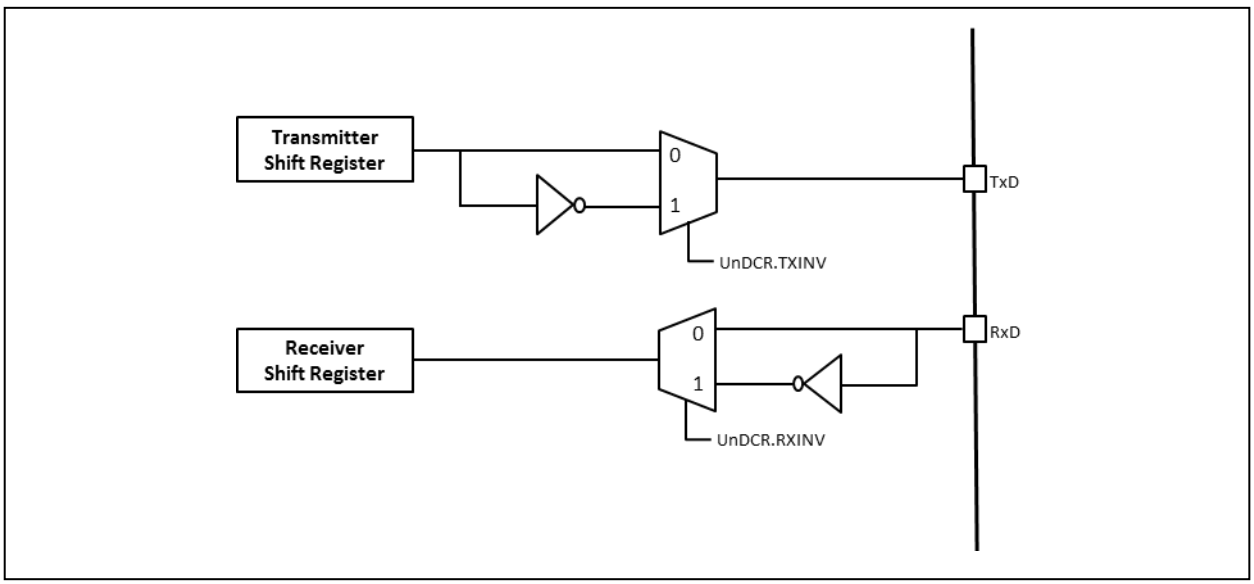


Figure 12-2 Data Inversion Control Diagram

Un.LSR UART Line Status Register

The UART Line Status register is an 8-bit register.

U0.LSR=0x4000_8014, U1.LSR=0x4000_8114

7	6	5	4	3	2	1	0
-	TEMT	THRE	BI	FE	PE	OE	DR
0	1	1	0	0	0	0	0
	R	R	R	R	R	R	R

6	TEMT	Transmit empty.
	0	Transmit register has the data is now transferring
	1	Transmit register is empty.
5	THRE	Transmit holding empty.
	0	Transmit holding register is not empty.
	1	Transmit holding register empty
4	BI	Break condition indication bit
	0	Normal status
	1	Break condition is detected
3	FE	Frame Error.
	0	No framing error.
	1	Framing error. The receive character did not have a valid stop bit
2	PE	Parity Error
	0	No parity error
	1	Parity error. The receive character does not have correct parity information.
1	OE	Overrun error
	0	No overrun error
	1	Overrun error. Additional data arrives while the RHR is full
0	DR	Data received
	0	No data in receive holding register.
	1	Data has been received and is saved in the receive holding register

This register provides the status of data transfers between Transmitter and Receiver. Users can get the line status information from this register and can handle the next process. Bits 1,2,3,4 will raise the line status interrupt when the RLSIE bit in the Un.IEN register is set. Other bits can generate its interrupt when its interrupt enable bit in the Un.IEN register is set.

Un.BDR Baud Rate Divisor Latch Register

The UART Baud Rate Divisor Latch register is a 16-bit register.

Note: Make sure the UART clock is set in MCCR4.

U0.BDR=0x4000_8020, U1.BDR=0x4000_8120

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDR															
0x0000															
RW															

15 0	BDR	Baud rate Divider latch value

To establish communication with the UART channel, the baud rate should be set properly. The programmable baud rate generator is provided to give from 1 to 65535 divider number. The 16-bit divider register (UnBDR) should be written for the expected baud rate $UART_{clock}$ gets from MCCR4.

The baud rate calculation formula is shown in the following equation:

$$\text{BDR} = \frac{UART_{clock}}{16 \times BaudRate}$$

For a $UART_{clock}$ speed of 40 MHz, the divider value and error rate is listed in Table 12-6.

Table 12-6 Example of Baud Rate Calculation (without BFR)

$UART_{clock}$=40 MHz		
Baud rate	**Divider**	**Error (%)**
1200	2083	0.02%
2400	1041	0.06%
4800	520	0.16%
9600	260	0.16%
19200	130	0.16%
38400	65	0.16%
57600	43	0.94%
115200	21	3.34%

Un.BFR Baud Rate Fraction Counter Register

The Baud Rate Fraction Counter register is an 8-bit register.

U0.BFR=0x4000_8024, U1.BFR=0x4000_8124

7	6	5	4	3	2	1	0
BFR							
0x00							
RW							

7 0	BFR	Fractions counter value.	
		0	Fraction counter is disabled
		N	Fraction counter enabled. Fraction compensation mode is operating. Fraction counter is incremented by FCNT.

Table 12-7 Example of Baud Rate Calculation

$UART_{clock}$ =40 MHz			
Baud rate	**Divider**	**FCNT**	**Error (%)**
1200	2083	85	0.00%
2400	1041	170	0.00%
4800	520	213	0.00%
9600	260	106	0.00%
19200	130	53	0.00%
38400	65	262	0.00%
57600	43	103	0.00%
115200	21	179	0.01%

$$\mathrm{FCNT} = \mathrm{Float} * 256$$

The FCNT value is calculated using the equation above. For example, when the target baud rate is 4800 bps and $UART_{clock}$ is 40MHz, the BDR value is 520.8333. The integer number 520 should be the BDR value and the floating number 0.8333 will result in the FCNT value of 213, as shown below:

FCNT = 0.8333 * 256 = 213.3333, so the FCNT value is 213.

The 8-bit fractional counter will count up by FCNT value every (baud rate)/16 periods and when the fractional counter overflows, the divisor value increments by 1. Therefore, this period will be compensated. In the next period, the divisor value will return to the original set value.

Un.IDTR Inter-frame Delay Time Register

The UART Inter-frame Time register is an 8-bit register. A dummy delay can be inserted between two continuous transmits.

U0.IDTR=0x4000_8030, U1.IDTR=0x4000_8130

7	6	5	4	3	2	1	0
SMS	DMS				WAITVAL		
1	0	0	0	0	000		
RW	RW				RW		

7	SMS	Start Bit Multi sampling enable
	0	Multi sampling is disable for start bit, Single sample will be done at 8/16 baud rate for the start bit
	1	Multi sampling is enabled for start bit. Sampling is done 3 times at 7/16, 8/16 and 9/16 baud rate. Dominant value of 3 samples will be selected for the start bit
6	DMS	Data Bit Multi sampling enable
	0	Multi sampling is disable for data bit, Single sample will be done at 8/16 baud rate for the data bit
	1	Multi sampling is enabled for data bit. Sampling is done 3 times at 7/16, 8/16 and 9/16 baud rate. Dominant value of 3 samples will be selected for the data bit
2 0	WAITVAL	Wait time is decided by this value $\text{Wait Time} = \frac{WAITVAL}{BAUDRATE}$

Functional Description

The UART module is compatible with 16450 UART. Additionally, fractional baud rate compensation logic is provided. This module does not have an internal FIFO block. Therefore, data transfer will establish interactive support.

Receiver Sampling Timing

The UART operates per the following timing:

If the falling edge is on the receive line, UART judges it as the start bit. From the start timing, UART oversamples 16 times of 1-bit and detects the bit value at the 7th sample of 16 samples.

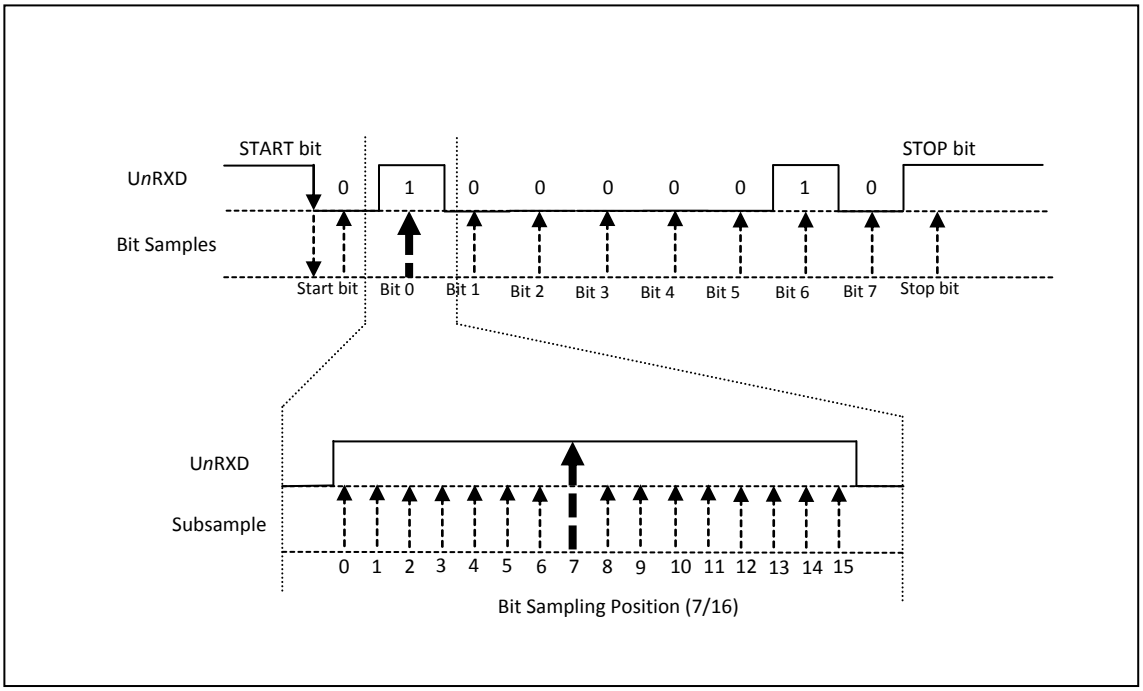


Figure 12-3 Sampling Timing of UART Receiver

Note: It is recommended to enable debounce settings in the PCU block to reinforce the immunity of external glitch noise.

Transmitter

The transmitter's function is to transmit data. The start bit, data bits, optional parity bit, and stop bit are serially shifted, with the least significant bit first. The number of data bits is selected in the DLAN[1:0] field in the Un.LCR register.

The parity bit is set according to the PARITY and PEN bit field in the Un.LCR register. If the parity type is even, then the parity bit depends on the one bit sum of all data bits. For odd parity, the parity bit is the inverted sum of all data bits.

The number of stop bits is selected in the STOPBIT field in the Un.LCR register.

An example of transmit data format is shown in Figure 12-4.

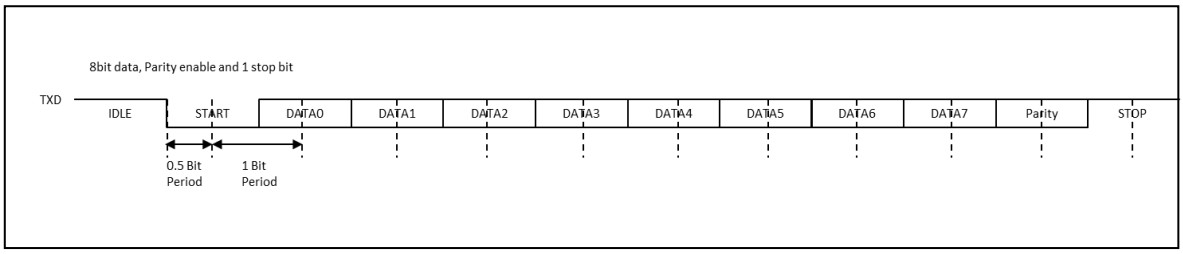


Figure 12-4 Transmit Data Format Example

Inter-frame Delay Transmission

The inter-frame delay function allows the transmitter to insert an Idle state on the TXD line between two characters. The width of the Idle state is defined in the WAITVAL field in the Un.IDTR register. When this field is set to 0, no time-delay is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted character during the number of bit periods defined in the WATIVAL field.

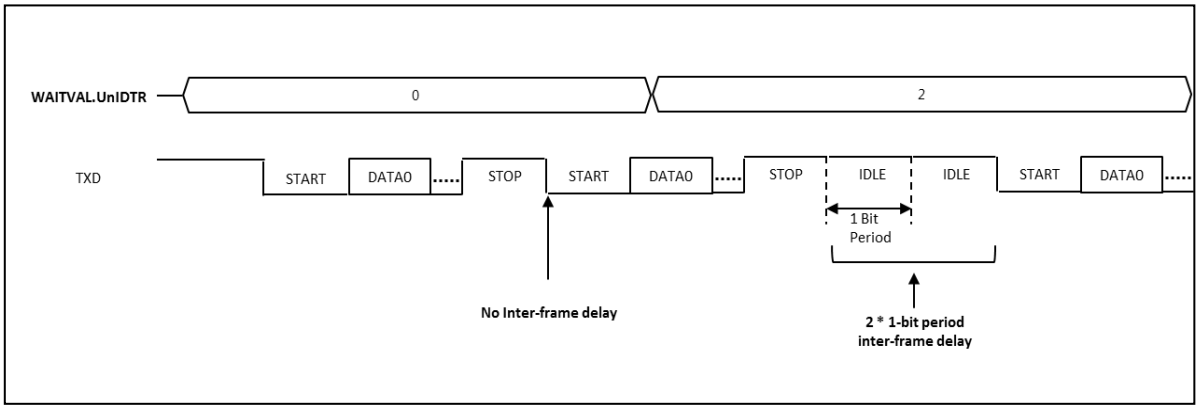


Figure 12-5 Inter-frame Delay Timing Diagram

Transmit Interrupt

The transmit operation creates interrupt flags. When the Transmitter Holding register is empty, the THRE interrupt flag will be set. When the Transmitter Shifter register is empty, the TXE interrupt flag will be set. Users can select which interrupt timing is best for the application.

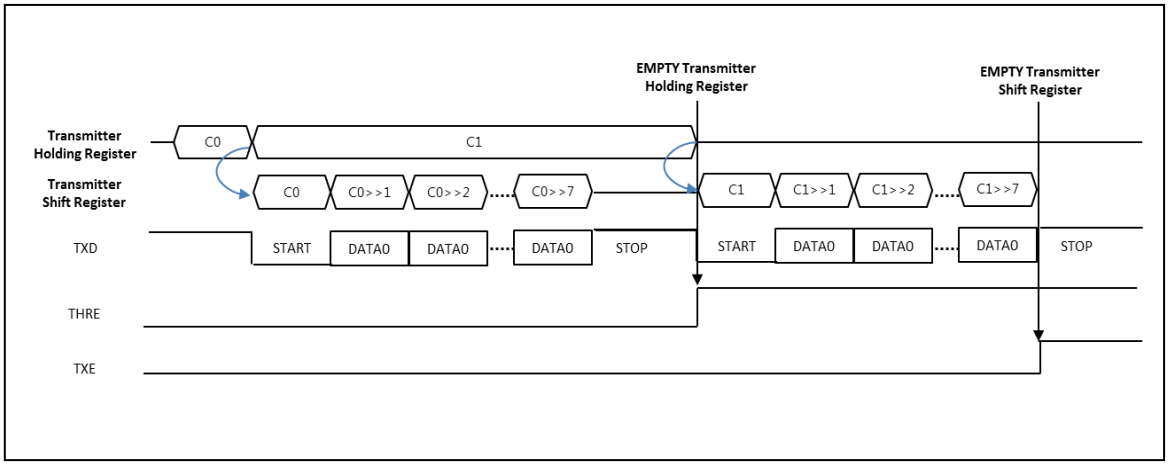


Figure 12-6 Transmit Interrupt Timing Diagram

13. Serial Peripheral Interface (SPI)

Overview

One-channel serial interface is provided for synchronous serial communications with external peripherals. The SPI block supports Master and Slave modes. Four signals are used for SPI communication – SS, SCK, MOSI, and MISO.

- Master or Slave operation
- Programmable clock polarity and phase
- 8, 9, 16, 17-bit wide transmit/receive register
- 8, 9, 16, 17-bit wide data frame
- Loop-back mode
- Programmable start, burst, and stop delay time

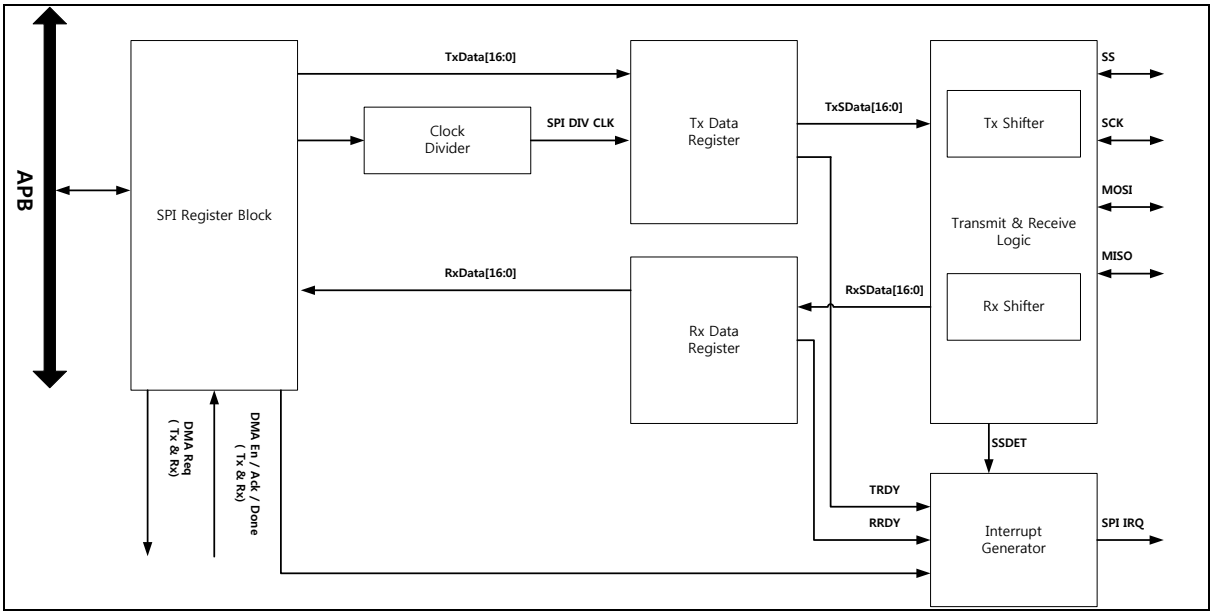


Figure 13-1 Block Diagram

Pin Description

Table 13-1 External Pins

PIN NAME	TYPE	DESCRIPTION
SS	I/O	SPI Slave select input / output
SCK	I/O	SPI Serial clock input / output
MOSI	I/O	SPI Serial data (Master output, Slave input)
MISO	I/O	SPI Serial data (Master input, Slave output)

Registers

The base address of SPI is 0x4000_9000 and the register map is described in Table 13-2 and Table 13-3.

Table 13-2 SPI Base Address

NAME	BASE ADDRESS
SPI	0x4000_9000

Table 13-3 SPI Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
SP.TDR	0x00	W	SPI Transmit Data Register	-
SP.RDR	0x00	R	SPI Receive Data Register	0x000000
SP.CR	0x04	RW	SPI Control Register	0x001020
SP.SR	0x08	RW	SPI Status Register	0x000006
SP.BR	0x0C	RW	SPI Baud rate Register	0x0000FF
SP.EN	0x10	RW	SPI Enable register	0x000000
SP.LR	0x14	RW	SPI delay Length Register	0x010101

SP.TDR SPI Transmit Data Register

SP.TDR is a 17-bit read/write register. It contains serial transmit data.

SP.TDR=0x4000_9000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TDR															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

16 0	TDR	Transmit Data Register

SP.RDR SPI Receive Data Register

SP.RDR is a 17-bit read/write register. It contains serial receive data.

SP.RDR=0x4000_9000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RDR															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

16 0	RDR	Receive Data Register

SP.CR SPI Control Register

SP.CR is a 20-bit read/write register and can be set to configure SPI operation mode.

SP.CR=0x4000_9004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											TXBC	RXBC			SSCIE	TXIE	RXIE	SSMOD	SSOUT	LBE	SSMARK	SSMO	SSPOL			MS	MSBF	CPHA	CPOL	BITSZ	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	00	
											RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	

20	TXBC	Tx buffer clear bit.
		0 No action
		1 Clear Tx buffer
19	RXBC	Rx buffer clear bit
		0 No action
		1 Clear Rx buffer
16	SSCIE	SS Edge Change Interrupt Enable bit.
		0 nSS interrupt is disabled.
		1 nSS interrupt is enabled for both edges (L→H, H→L)
15	TXIE	Transmit Interrupt Enable bit.
		0 Transmit Interrupt is disabled.
		1 Transmit Interrupt is enabled.
14	RXIE	Receive Interrupt Enable bit..
		0 Receive Interrupt is disabled.
		1 Receive Interrupt is enabled.
13	SSMOD	SS Auto/Manual output select bit.
		0 SS output is not set by SSOUT (SP.CR[12]). - SS signal is in normal operation mode.
		1 SS output signal is set by SSOUT.
12	SSOUT	SS output signal select bit.
		0 SS output is ‘L.’
		1 SS output is ‘H’.
11	LBE	Loop-back mode select bit in master mode.
		0 Loop-back mode is disabled.
		1 Loop-back mode is enabled.
10	SSMASK	SS signal masking bit in slave mode.
		0 SS signal masking is disabled. - Receive data when SS signal is active.
		1 SS signal masking is enabled. - Receive data at SCLK edges. SS signal is ignored.
9	SSMO	SS output signal select bit.
		0 SS output signal is disabled.
		1 SS output signal is enabled.
8	SSPOL	SS signal Polarity select bit.
		0 SS signal is Active-Low.
		1 SS signal is Active-High.
5	MS	Master/Slave select bit.
		0 SPI is in Slave mode.
		1 SPI is in Master mode.
4	MSBF	MSB/LSB Transmit select bit.
		0 LSB is transferred first.
		1 MSB is transferred first.
3	CPHA	SPI Clock Phase bit.

		0	Sampling of data occurs at odd edges (1,3,5,…,15).
		1	Sampling of data occurs at even edges (2,4,6,…,16).
2	CPOL	SPI Clock Polarity bit.	
		0	Active-high clocks selected.
		1	Active-low clocks selected.
1	BITSZ	Transmit/Receive Data Bits select bit.	
		00	8 bits
		01	9 bits
		10	16 bits
0		11	17 bits

CPOL=0, CPHA=0 : data sampling at rising edge, data changing at falling edge
CPOL=0, CPHA=1 : data sampling at falling edge, data changing at rising edge
CPOL=1, CPHA=0 : data sampling at falling edge, data changing at rising edge
CPOL=1, CPHA=1 : data sampling at rising edge, data changing at falling edge

SP.SR SPI Status Register

SP.SR is a 10-bit read/write register. It contains the status of SPI interface.

SP.SR=0x4000_9008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SSDET	SSON	OVRF	UDRF	TXIDLE	TRDY	RRDY
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
									RC1	RC1	RC1	RC1	R	R	R

6	SSDET	The rising or falling edge of SS signal Detect flag.
		0 SS edge is not detected.
		1 SS edge is detected.
		- The bit is cleared when it is written as “0”.
5	SSON	SS signal Status flag.
		0 SS signal is inactive.
		1 SS signal is active.
4	OVRF	Receive Overrun Error flag.
		0 Receive Overrun error is not detected.
		1 Receive Overrun error is detected.
		- This bit is cleared by writing or reading SP.RDR.
3	UDRF	Transmit Underrun Error flag.
		0 Transmit Underrun is not occurred.
		1 Transmit Underrun is occurred.
		- This bit is cleared by writing or reading SP.TDR.
2	TXIDLE	Transmit/Receive Operation flag.
		0 SPI is transmitting data
		1 SPI is in IDLE state.
1	TRDY	Transmit buffer Empty flag.
		0 Transmit buffer is busy.
		1 Transmit buffer is ready.
		- This bit is cleared by writing data to SP.TDR.
0	RRDY	Receive buffer Ready flag.
		0 Receive buffer has no data.
		1 Receive buffer has data.
		- This bit is cleared by writing data to SP.RDR.

SP.BR SPI Baud Rate Register

SP.BR is a 16-bit read/write register. The baud rate can be set by writing to the register.

SP.BR=0x4000_900C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR															
0x00FF															
RW															

15	BR	Baud rate setting bits
		Baud Rate = PCLK / (BR + 1)
0		(BR must be bigger than “0”, BR >= 2)

SP.EN SPI Enable Register

SP.EN is a bit read/write register. It contains the SPI enable bit.

SP.EN=0x4000_9010

7	6	5	4	3	2	1	0
							ENABLE
0	0	0	0	0	0	0	0
							RW

0	ENABLE	SPI Enable bit
		0 SPI is disabled. - SP.SR is initialized by writing "0" to this bit but other registers aren't initialized.
		1 SPI is enabled. - When this bit is written as "1", the dummy data of transmit buffer will be shifted. To prevent this, write data to SP.TDR before this bit is active.

SP.LR SPI Delay Length Register

SP.LR is a 24-bit read/write register. It contains start, burst, and stop length values.

SP.LR=0x4000_9014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SPL								BTL								STL							
0	0	0	0	0	0	0	0	0x01								0x01								0x01							
								RW								RW								RW							

23	SPL	StoP Length value
16		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (SPL ≥ 1)
15	BTL	BursT Length value
8		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (BTL ≥ 1)
7	STL	STart Length value
0		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (STL ≥ 1)

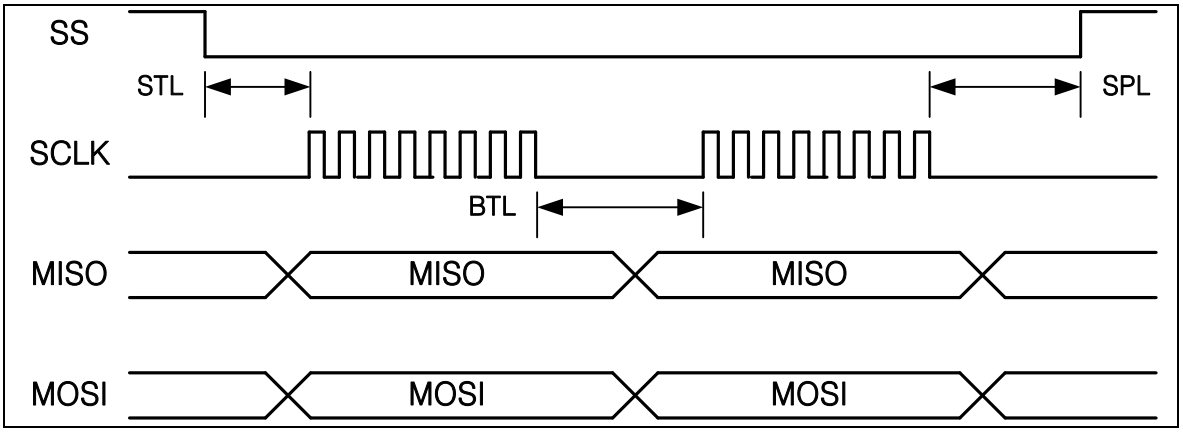


Figure 13-2 SPI Wave form (STL, BTL and SPL)

Functional Description

The SPI Transmit block and Receive block share the Clock Gen block but they are independent of each other. The Transmit and Receive blocks have double buffers and SPI is available for back to back transfer operation.

SPI Timing

The SPI has four modes of operation. These modes essentially control the way data is clocked in or out of an SPI device. The configuration is done by two bits in the SPI control register (SP.CR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock. The clock phase (CPHA) control bit selects one of the two fundamentally different transfer formats. To ensure proper communication between master and slave, both devices have to run in the same mode. This can require a reconfiguration of the master to match the requirements of different peripheral slaves.

The clock polarity has no significant effect on the transfer format. Switching this bit causes the clock signal to be inverted (active high becomes active low and idle low becomes idle high). The settings of the clock phase, however, select one of two different transfer timings, which are described in further detail in the next two chapters. Because the MOSI and MISO lines of the master and the slave are directly connected to each other, the diagrams show the timing of both devices – master and slave. The nSS line is the slave select input of the slave. The nSS pin of the master is not shown in the diagrams. It has to be inactive by a high level on this pin (if configured as input pin) or by configuring it as an output pin.

The timing of an SPI transfer where CPHA is zero is shown in Figure 13-3 and Figure 13-4. Two wave forms are shown for the SCK signal – one for CPOL equals zero and another for CPOL equals one.

When the SPI is configured as a slave, the transmission starts with the falling edge of the /SS line. This activates the SPI of the slave and the MSB of the byte stored in its data register (SP.TDR) is output on the MISO line. The actual transfer is started by a software write to the SP.TDR of the master. This causes the clock signal to be generated. In cases where the CPHA equals zero, the SCLK signal remains zero for the first half of the first SCLK cycle. This ensures that the data is stable on the input lines of both the master and the slave. The data on the input lines is read with the edge of the SCLK line from inactive to active state. The edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one) causes the data to be shifted one bit further so that the next bit is output on the MOSI and MISO lines.

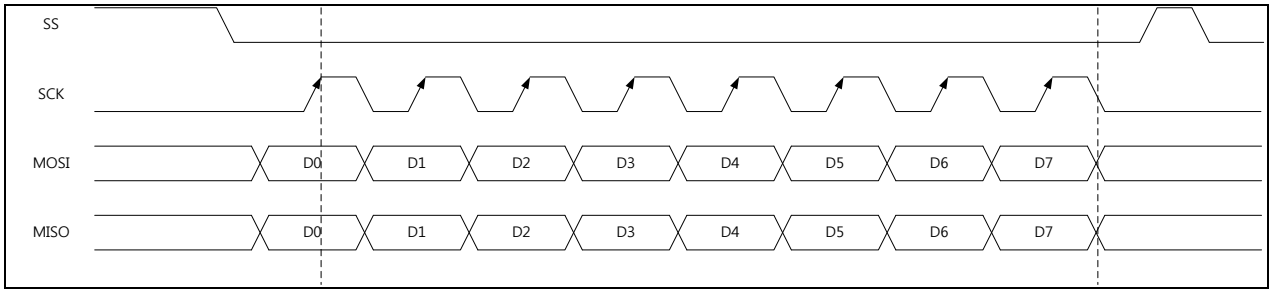


Figure 13-3 SPI Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0)

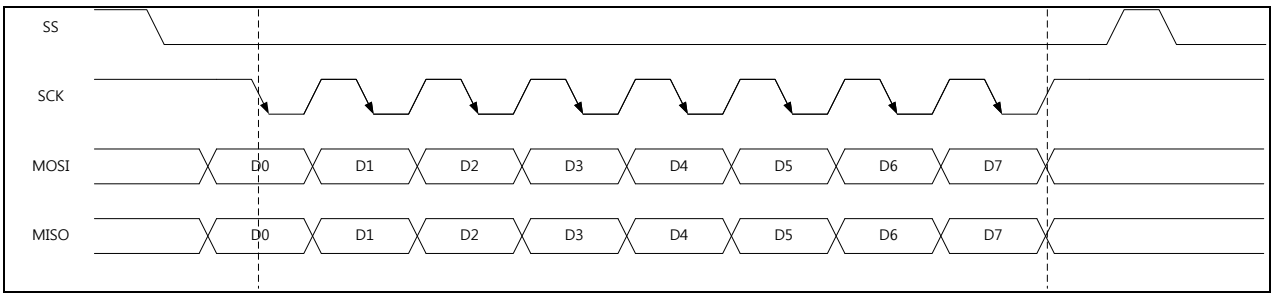


Figure 13-4 SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1)

The timing of an SPI transfer where CPHA is 1, is shown in Figure 13-5 and Figure 13-6.Two wave forms are shown for the SCLK signal – one for CPOL equals zero and another for CPOL equals one.

Similar to the previous cases, the falling edge of the nSS lines selects and activates the slave. Compared to the previous cases, where CPHA equals zero, the transmission is not started and the MSB is not output by the slave at this stage. The actual transfer is started by a software write to the SP.TDR of the master which causes the clock signal to be generated. The first edge of the SCLK signal from its inactive to its active state (rising edge if CPOL equals zero and falling edge if CPOL equals one) causes both the master and the slave to output the MSB of the byte in the SP.TDR.

As shown in Figure 13-3 and Figure 13-4, there is no delay of half a SCLK-cycle. The SCLK line changes its level immediately at the beginning of the first SCLK-cycle. The data on the input lines is read with the edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one). After eight clock pulses, the transmission is completed.

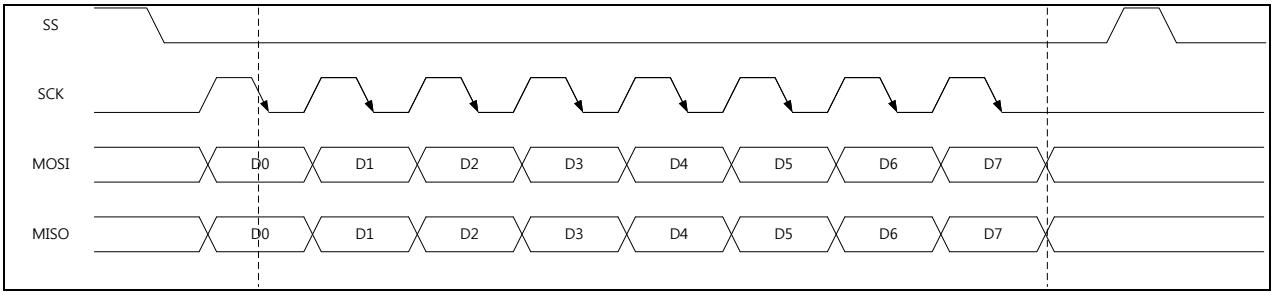


Figure 13-5 SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0)

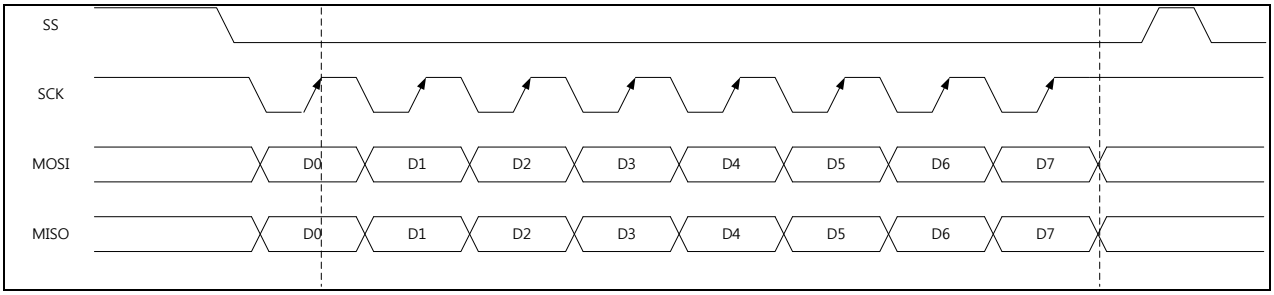


Figure 13-6 SPI Transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1)

14. I²C Interface

Overview

The Inter-Integrated Circuit (I²C) bus serves as an interface between the microcontroller and the serial I²C bus. It provides two wires, serial bus interface to a large number of popular devices and allows parallel-bus systems to communicate bidirectionally with the I²C-bus. Features include:

- Master and slave operation
- Programmable communication speed
- Multi-master bus configuration
- 7-bit addressing mode
- Standard data rate of 100/400 KBps
- STOP signal generation and detection
- START signal generation
- ACK bit generation and detection

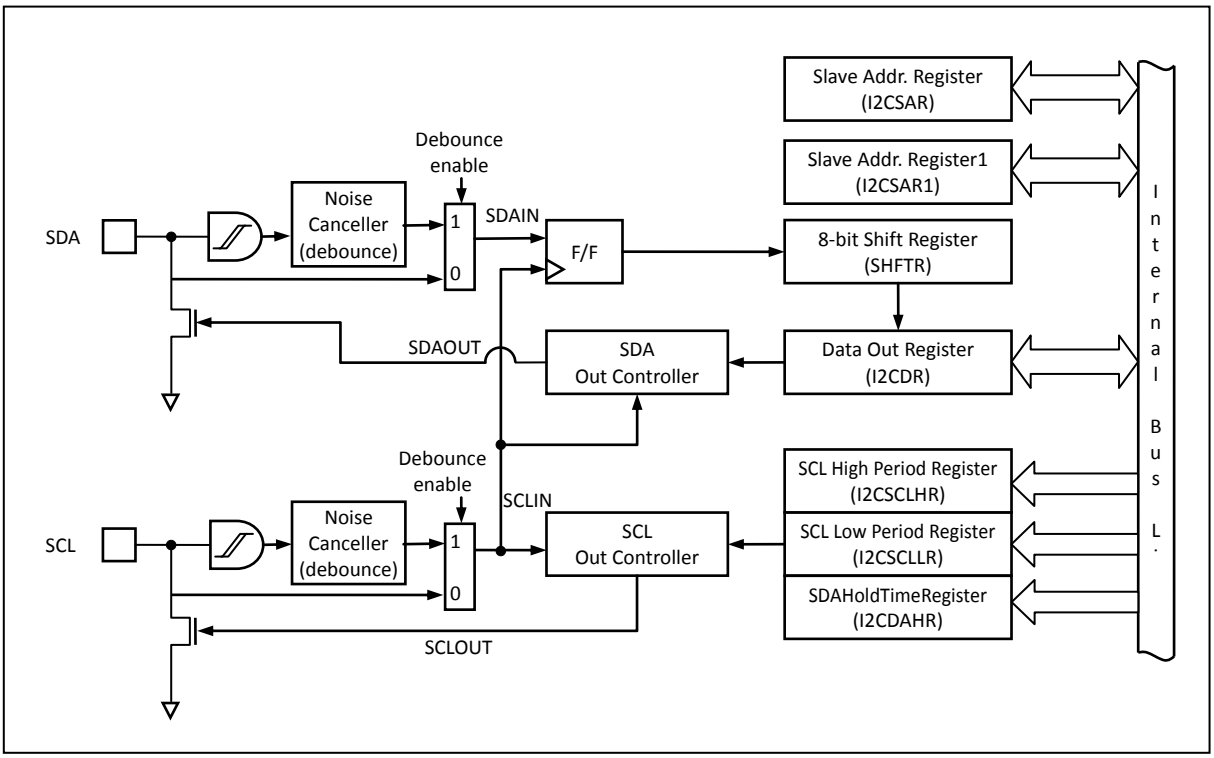


Figure 14-1. I²C Block Diagram

Pin Description

Table 14-1 I²C Interface External Pins

PIN NAME	TYPE	DESCRIPTION
SCL	I/O	I²C channel Serial clock bus line (open-drain)
SDA	I/O	I²C channel Serial data bus line (open-drain)

Registers

The base address of I²C is 0x4000_A000. The register map is described in Table 14-2 and Table 14-3.

Table 14-2 I²C Interface Base Address

NAME	BASE ADDRESS
I²C	0x4000_A000

Table 14-3 I²C Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
IC.DR	0x00	RW	I²C Data Register	0xFF
IC.SR	0x08	R, RW	I²C Status Register	0x00
IC.SAR	0x0C	RW	I²C Slave Address Register	0x00
IC.CR	0x14	RW	I²C Control Register	0x00
IC.SCLL	0x18	RW	I²C SCL LOW duration Register	0xFFFF
IC.SCLH	0x1C	RW	I²C SCL HIGH duration Register	0xFFFF
IC.SDH	0x20	RW	I²C SDA Hold Register	0x7F

IC.DR I²C Data Register

IC.DR is an 8-bit read/write register. It contains a byte of serial data to be transmitted or a byte which has just been received.

IC.DR=0x4000_A000

7	6	5	4	3	2	1	0
DR							
0xFF							
RW							

7 0	DR	The most recently received data or data to be transmitted.

IC.SR I²C Status Register

IC.SR is an 8-bit read/write register. It contains the status of I²C bus interface. Writing to the register clears the status bits.

IC.SR=0x4000_A008

7	6	5	4	3	2	1	0
GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMODE	RXACK
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

7	GCALL	General call flag	
		0	General call is not detected.
		1	General call detected or slave address (ID byte) was sent.
6	TEND	1 Byte transmission complete flag	
		0	The transmission is working or not completed.
		1	The transmission is completed.
5	STOP	STOP flag	
		0	STOP is not detected.
		1	STOP is detected.
4	SSEL	Slave flag	
		0	Slave is not selected.
		1	Slave is selected.
3	MLOST	Mastership lost flag	
		0	Mastership is not lost.
		1	Mastership is lost.
2	BUSY	BUSY flag	
		0	I²C bus is in IDLE state.
		1	I²C bus is busy.
1	TMODE	Transmitter/Receiver mode flag	
		0	Receiver mode.
		1	Transmitter mode.
0	RXACK	Rx ACK flag	
		0	Rx ACK is not received.
		1	Rx ACK is received.

IC.SAR I²C Slave Address Register

IC.SAR is an 8-bits read/write register. It shows the address in Slave mode.

IC.SAR=0x4000_A00C

7	6	5	4	3	2	1	0
SVAD							GCEN
0x00							0
RW							RW

7 1	SVAD	7-bit Slave Address
0	GCEN	General call enable bit
	0	General call is disabled.
	1	General call is enabled.

IC.CR I²C Control Register

IC.CR is a 16-bit read/write register. This register can be set to configure I²C operation mode and simultaneously allowed for I²C transactions to be kicked off.

IC.CR=0x4000_A014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						INTDEL		IIF		SOFTRST	INTEN	ACKEN		STOP	START
0	0	0	0	0	0	00		0	0	0	0	0	0	0	0
						RW		R		RW	RW	RW		RW	RW

9 8	INTDEL	Interval delay value between address and data transfer (or DATA and DATA)
	0	1 * ICnSCLL
	1	2 * ICnSCLL
	2	4 * ICnSCLL
	3	8 * ICnSCLL
7	IIF	Interrupt status bit
	0	Interrupt is inactive
	1	Interrupt is active
5	SOFTRST	Soft Reset enable bit.
	0	Soft Reset is disabled.
	1	Soft Reset is enabled..
4	INTEN	Interrupt enabled bit.
	0	Interrupt is disabled.
	1	Interrupt is enabled.
3	ACKEN	ACK enable bit in Receiver mode.
	0	ACK is not sent after receiving data.
	1	ACK is sent after receiving data.
1	STOP	Stop enable bit. When this bit is set as “1” in transmitter mode, next transmission will be stopped even though ACK signal has been received.
	0	Stop is disabled.
	1	Stop is enabled. When this bit is set, transmission will be stopped.
0	START	Transmission start bit in master mode.
	0	Waits in slave mode.
	1	Starts transmission in master mode.

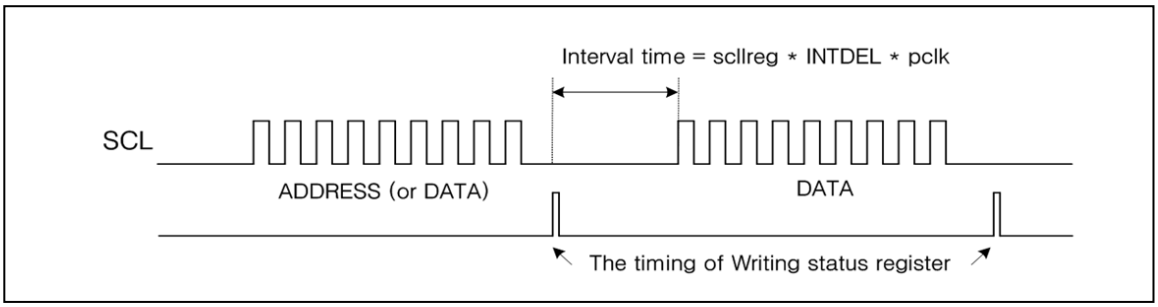


Figure 14-2 INTDEL in Master Mode

IC.SCLL I²C SCL LOW Duration Register

IC.SCLL is a 16-bit read/write register. SCL LOW time can be set by writing this register in Master mode.

IC.SCLL=0x4000_A018

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

15	SCLL	SCL LOW duration value. SCLL = (PCLK * SCLL[15:0]) + 2*PCLKs
0		Default value is 0xFFFF.

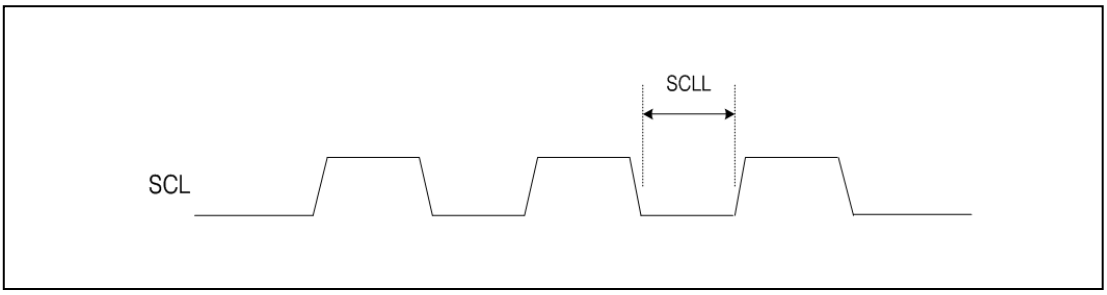


Figure 14-3 SCL LOW Timing

IC.SCLHI²C SCL HIGH Duration Register

IC.SCLH is a 16-bit read/write register. SCL HIGH time can be set by writing this register in Master mode.

IC.SCLH=0x4000_A01C

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

15	SCLH	SCL HIGH duration value. SCLH = (PCLK * SCLH[15:0]) + 3 PCLKs
0		Default value is 0xFFFF.

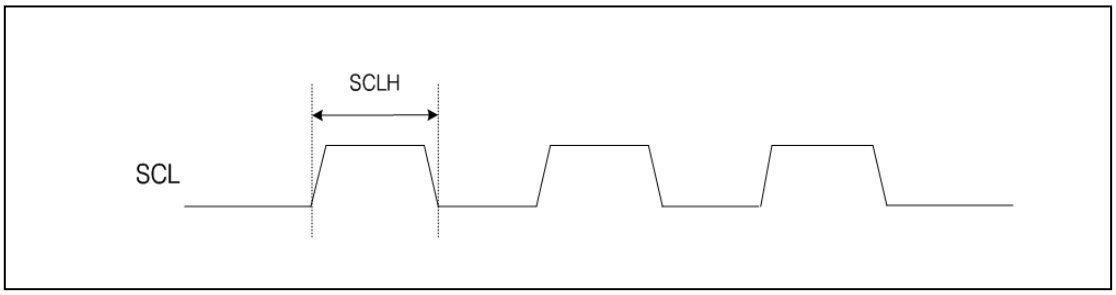


Figure 14-4 SCL HIGH Timing

IC.SDH SDA Hold Register

IC.SDH is a 15-bit read/write register. SDA HOLD time can be set by writing this register in Master mode.

IC.SDH=0x4000_A020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDH														
0x7FFF															
RW															

14	SDH	SDA HOLD time setting value. SDH = (PCLK * SDH[14:0]) + 4 PCLKs
0		Default value is 0x7FFF.

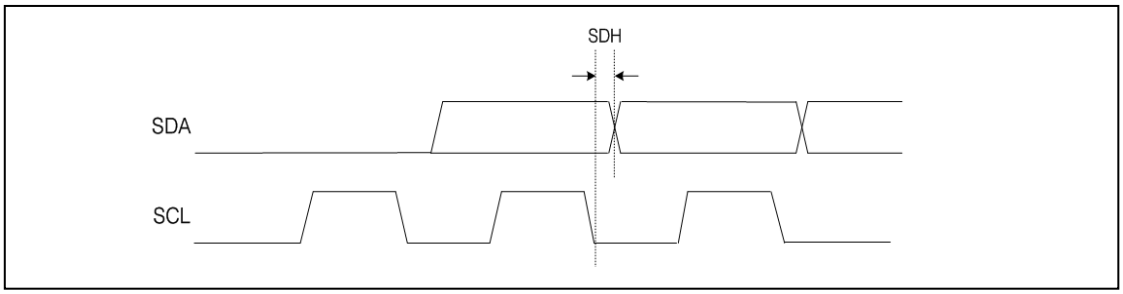


Figure 14-5 SDA HOLD Timing

Functional Description

I²C Bit Transfer

The data on the SDA line must be stable during the “H” period of the clock. The “H” or “L” state of the data line can only change when the clock signal on the SCL line is “L” (see Figure 14-6).

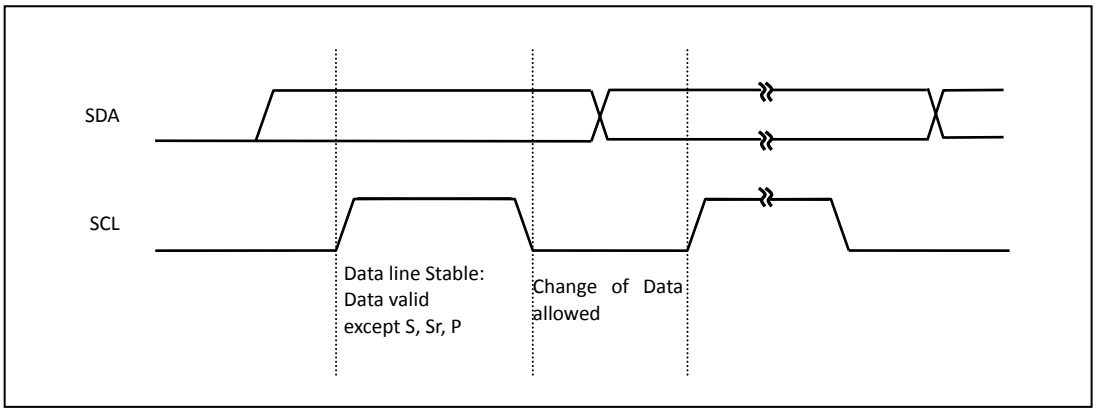


Figure 14-6 I²C Bus Bit Transfer

START/Repeated START/STOP

Within the procedure of the I²C-bus, unique situations arise which are defined as START(S) and STOP(P) conditions (see Figure 14-7).

An “H” to “L” transition on the SDA line while SCL is “H” is one such unique case. This situation indicates a START condition. An “L” to “H” transition on the SDA line while SCL is “H” defines a STOP condition.

START and STOP conditions are always generated by the Master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

The bus is busy if a repeated START(Sr) is generated instead of a STOP condition. In this respect, the START(S) and repeated START(Sr) conditions are functionally identical. For the remainder of this document therefore, the S symbol will be used as a generic term to represent both the START and repeated START conditions, unless Sr is particularly relevant.

Detection of START and STOP conditions by devices connected to the bus is easy if they incorporate the necessary interfacing hardware. However, microcontrollers with no such interface have to sample the SDA line at least twice per clock period to sense the transition.

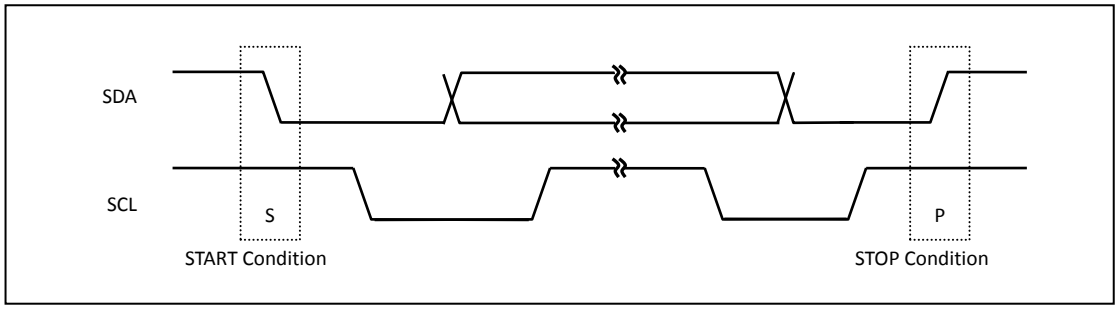


Figure 14-7 START and STOP Condition

Data Transfer

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (see Figure 14-8). If a slave can't receive or transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL "L" to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

A message which starts with such an address can be terminated by generation of a STOP condition, even during the transmission of a byte. In this case, no acknowledge is generated.

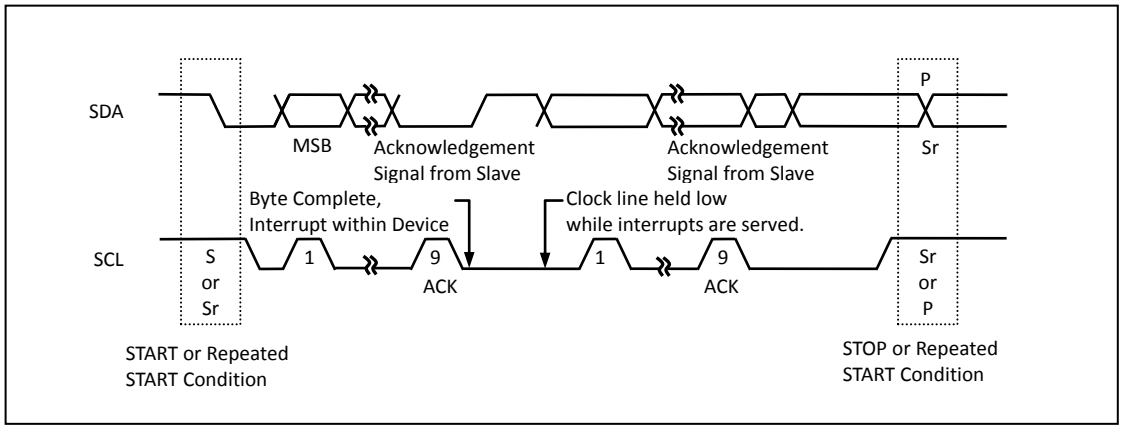


Figure 14-8 I^2C Bus Data Transfer

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse.

The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable “L” during the “H” period of this clock pulse (see Figure 14-9). Set-up and hold times must also be taken into account.

When a slave doesn’t acknowledge the slave address (for example, it is unable to receive or transmit because it is performing some real-time function), the data line must be left “H” by the slave. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a slave-receiver does acknowledge the slave address but, sometime later in the transfer cannot receive any more data bytes, the master must again abort the transfer. This is indicated by the slave generating the not-acknowledge on the first byte to follow. The slave leaves the data line “H” and the master generates a STOP or a repeated START condition.

If a master-receiver is involved in a transfer, it must signal the end of data to the slave-transmitter by not generating acknowledge on the last byte that was clocked out of the slave. The slave-transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

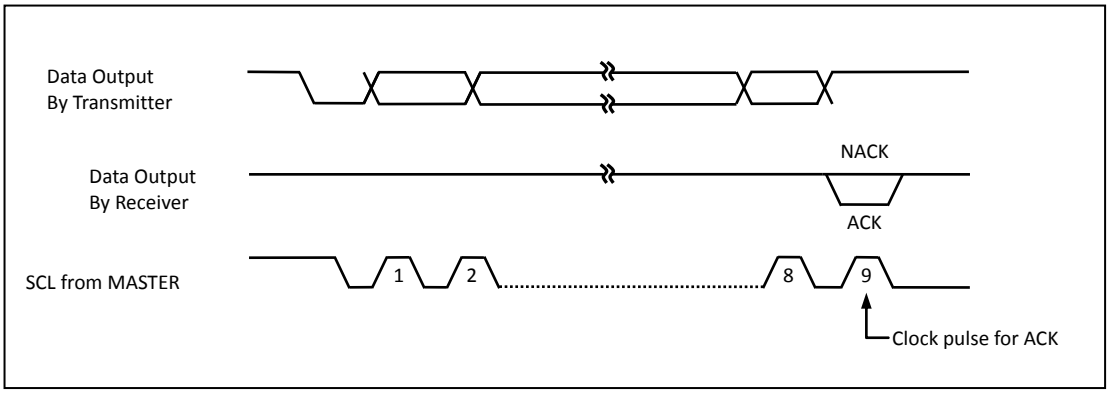


Figure 14-9 I²C Bus Acknowledge

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I^2C-bus. Data is only valid during the “H” period of the clock. Therefore, a defined clock is required for the bit-by-bit arbitration procedure to take place.

Clock synchronization is performed using the wired-AND connection of I^2C interfaces to the SCL line. This means that an “H” to “L” transition on the SCL line will cause the devices to start counting off their “L” period and, once a device clock has gone “L”, it will hold the SCL line in that state until the clock “H” state is reached (see Figure 14-10). However, the “L” to “H” transition of this clock may not change the state of the SCL line if another clock is still within its “L” by the device with the longest “L” period. Devices with shorter “L” periods enter an “H” wait-state during this time.

When all devices concerned have counted off their “L” period, the clock line will be released and go “H”. There will then be no difference between the device clocks and the state of the SCL line, and the devices will start counting their “H” periods. The first device to complete its “H” period will again pull the SCL line “L”.

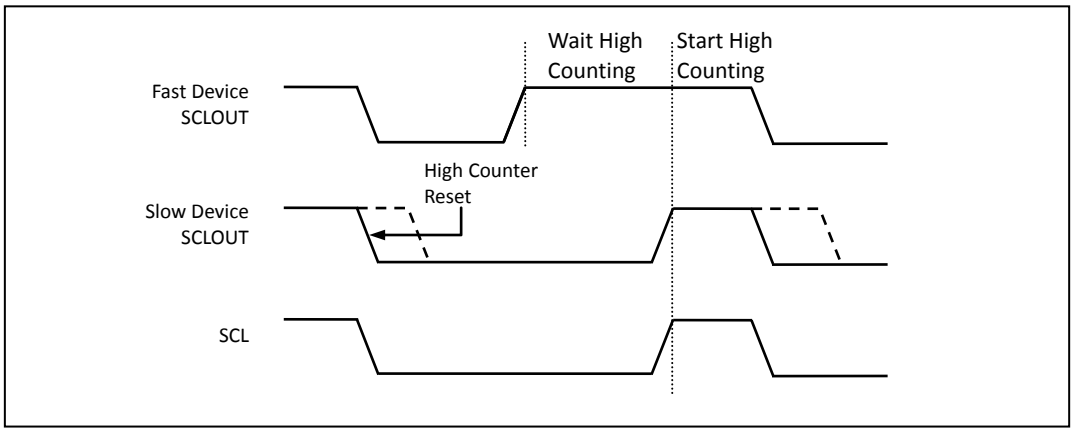


Figure 14-10 Clock Synchronization During the Arbitration Procedure

Arbitration

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition within the minimum hold time of the START condition which results in a defined START condition to the bus.

Arbitration takes place on the SDA line, while the SCL line is at the “H” level, in such a way that the master which transmits “H” level, while another master is transmitting “L” level will switch off its DATA output stage because the level on the bus doesn’t correspond to its own level.

Arbitration can continue for many bits. Its first stage is comparison of the address bits. If the masters are each trying to address the same device, arbitration continues with comparison of the data-bits if they are master-transmitter or acknowledge-bits if they are master-receiver. Because address and data information on the I^2C-bus is determined by the winning master, no information is lost during the arbitration process.

A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

If a master also incorporates a slave function and it loses arbitration during the addressing stage, it’s possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave mode.

Figure 14-11 shows the arbitration procedure for two masters. Of course, more may be involved (depending on how many masters are connected to the bus). As soon as there is a difference between the internal data level of the master generating Device1 data out and the actual level on the SDA line, its data output is switched off, which means that a “H” output level is then connected to the bus. This will not affect the data transfer initiated by the winning master.

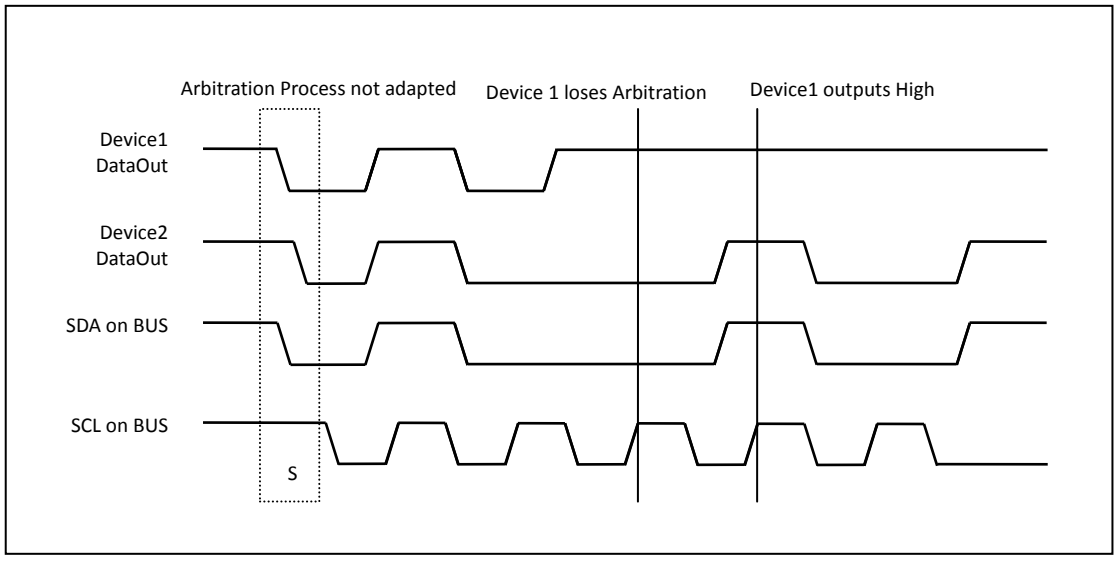


Figure 14-11 Arbitration Procedure Between Two Masters

I^2C Operation

I^2C supports the interrupt operation. Once interrupt is serviced, the IIF (IC.CR[7]) flag is set. ICnSR shows I^2C-bus status information and the SCL line stays “L” before the register is written as a certain value. The status register can be cleared by writing to the status register.

Master Transmitter

The master transmitter shows the flow of transmitter in Master mode (see Figure 14-12).

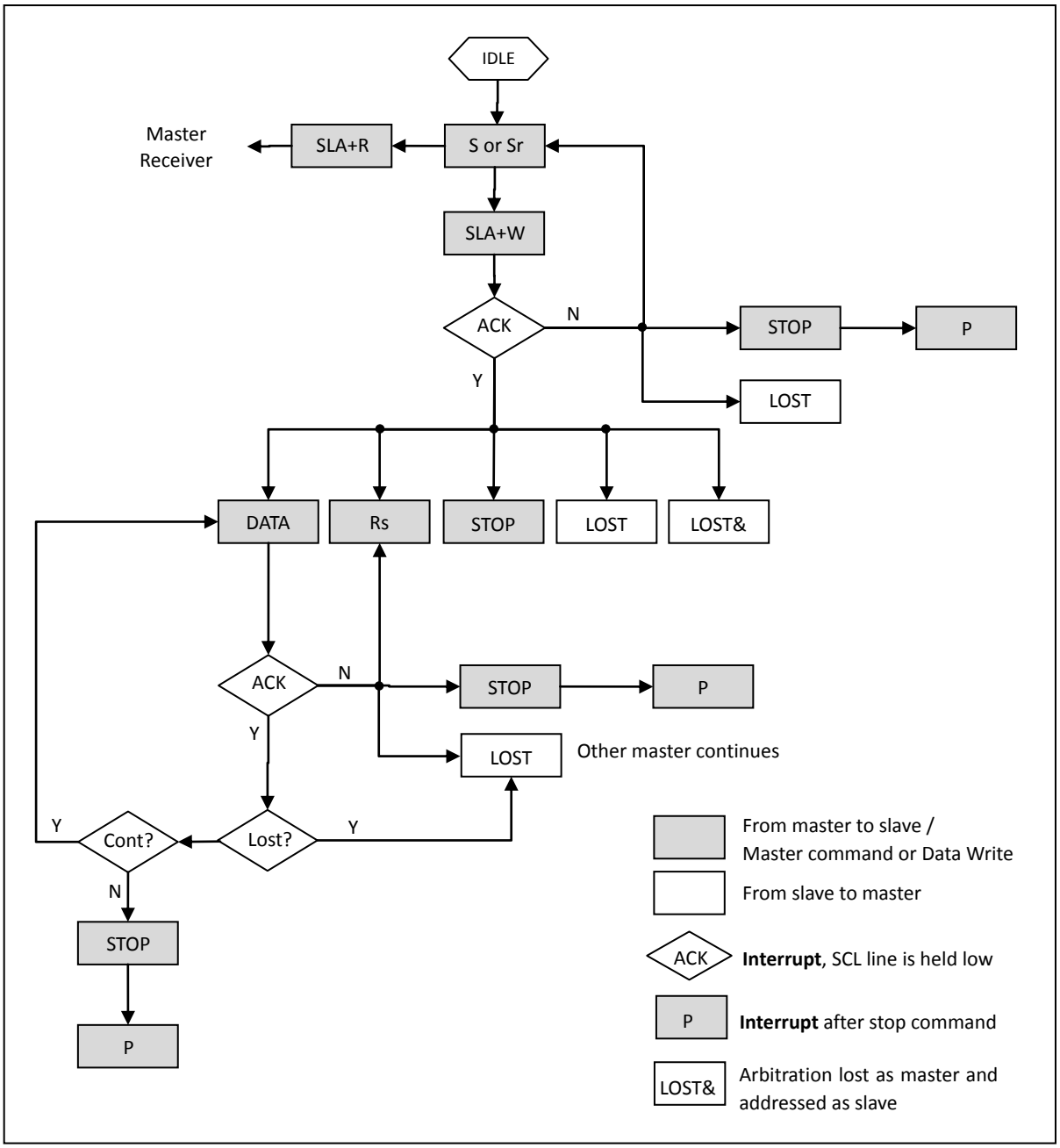


Figure 14-12 Transmitter Flowchart in Master Mode

Master Receiver

The master receiver shows the flow of receiver in Master mode (see Figure 14-13).

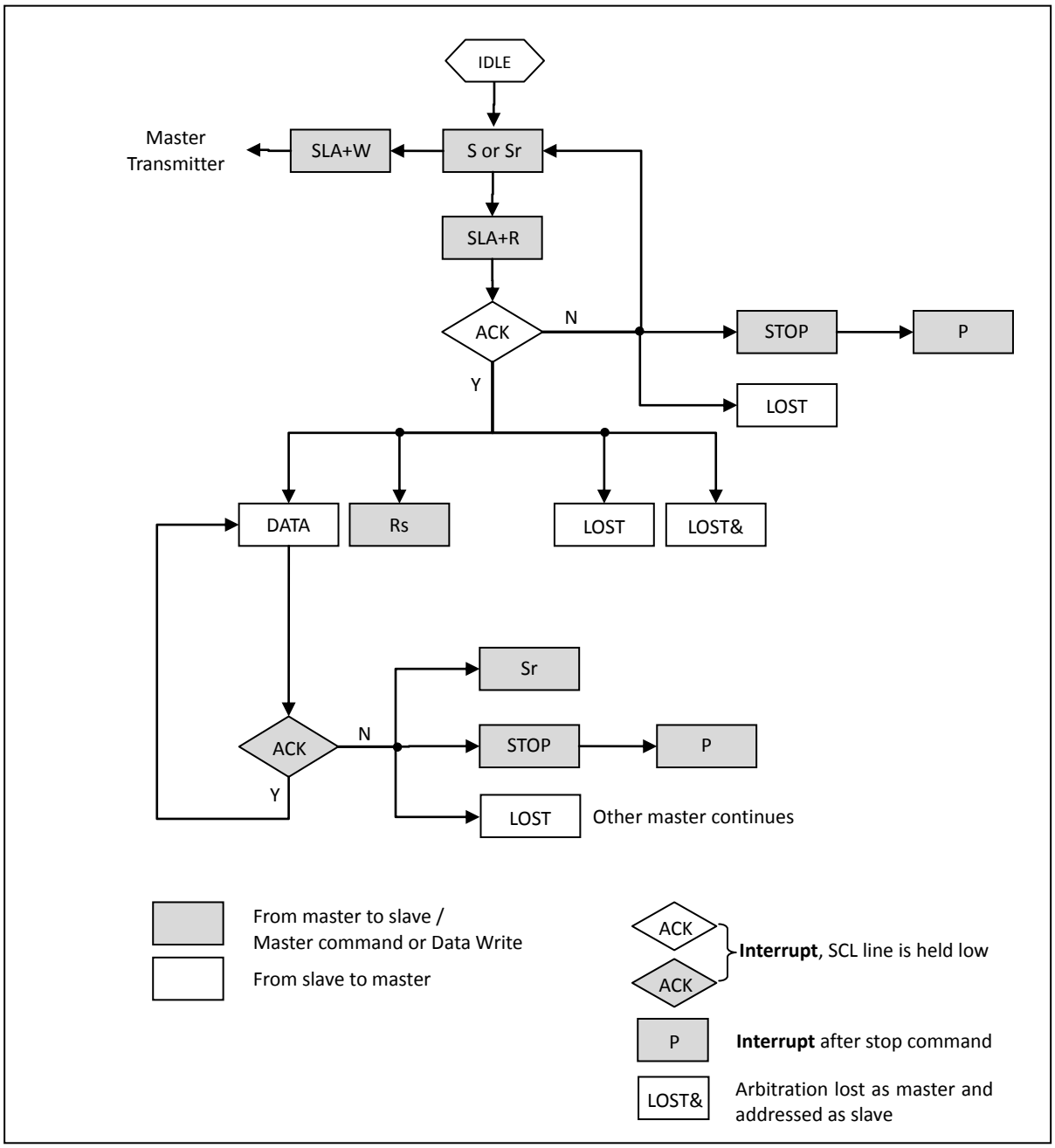


Figure 14-13 Receiver Flowchart in Master Mode

Slave Transmitter

The slave transmitter shows the flow of transmitter in Slave mode (see Figure 14-14).

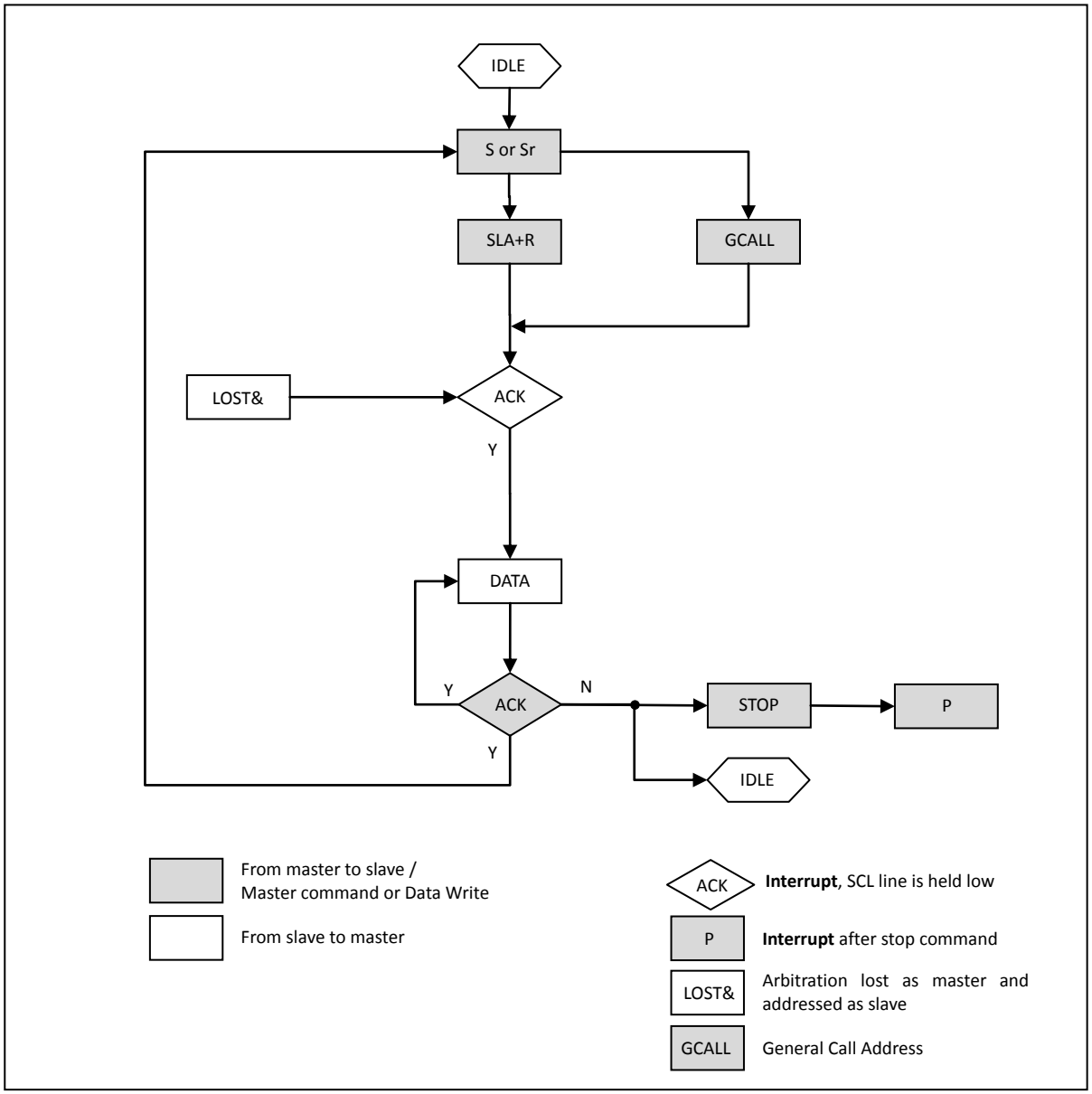


Figure 14-14 Transmitter Flowchart in Slave Mode

Slave Receiver

The slave receiver shows the flow of receiver in Slave mode (see Figure 14-15).

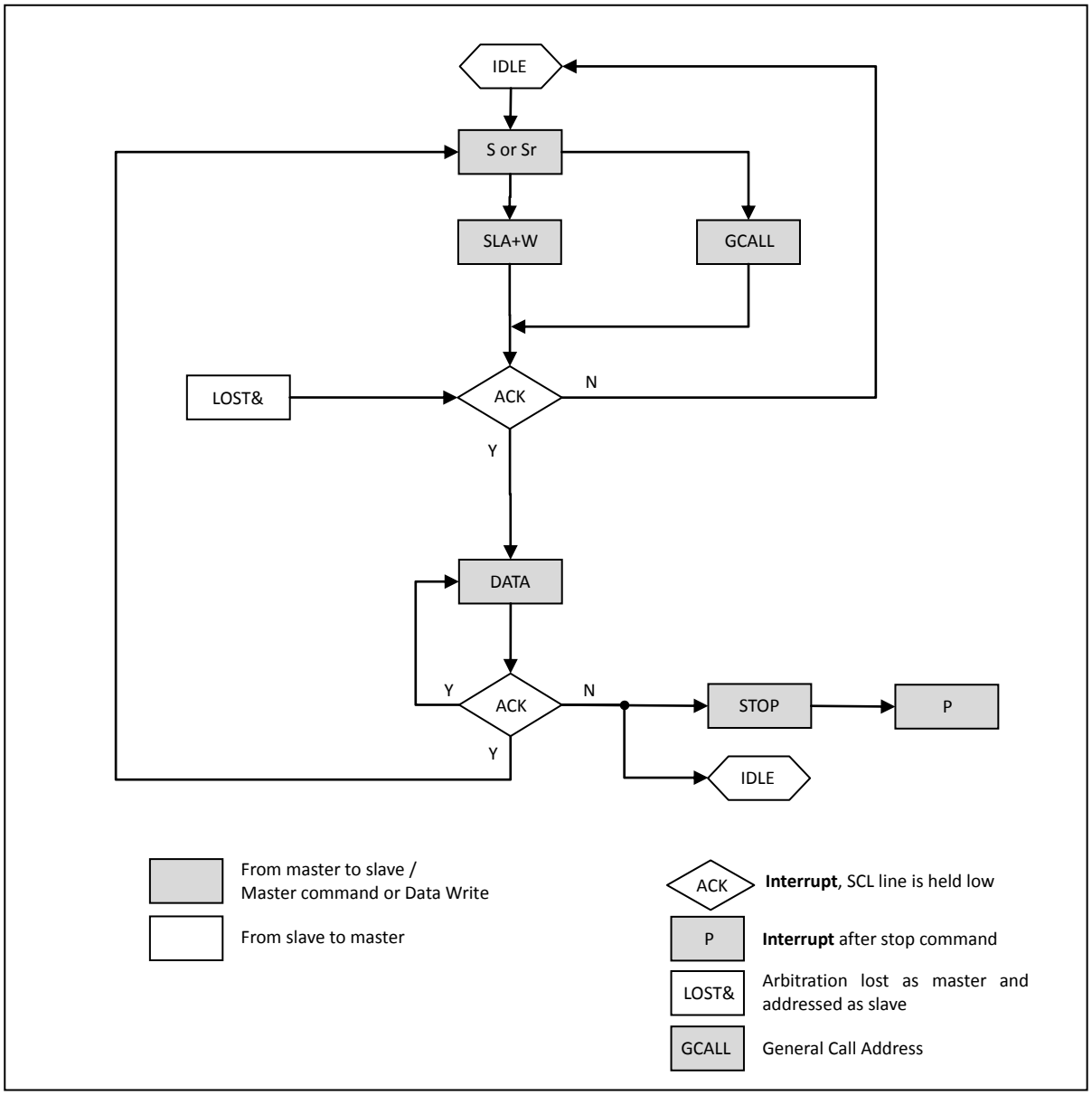


Figure 14-15 Receiver Flowchart in Slave Mode

15. Motor Pulse Width Modulator (MPWM)

Overview

Motor Pulse Width Modulator (MPWM) is a programmable motor controller which is optimized for 3-phase AC and DC motor control applications. It can be used in many other applications that require timing, counting, and comparison features.

MPWM includes 3 channels, each of which controls a pair of outputs that can control a motor.

- 16-bit counter
- 6-channel outputs for motor control
- Dead-time support
- Protection event and over voltage event handling
- 6 ADC trigger outputs
- Interval interrupt mode (period interrupt only)
- Up-down count mode

The MPWM clock source which is MPWM counter clock source will be provided from the SCU block. The MPWM resolution and period will be defined by this MPWM clock configuration. The default MPWM clock is the same as the RINGOSC clock. Prior to enabling the MPWM module, proper MPWM clock selection is required.

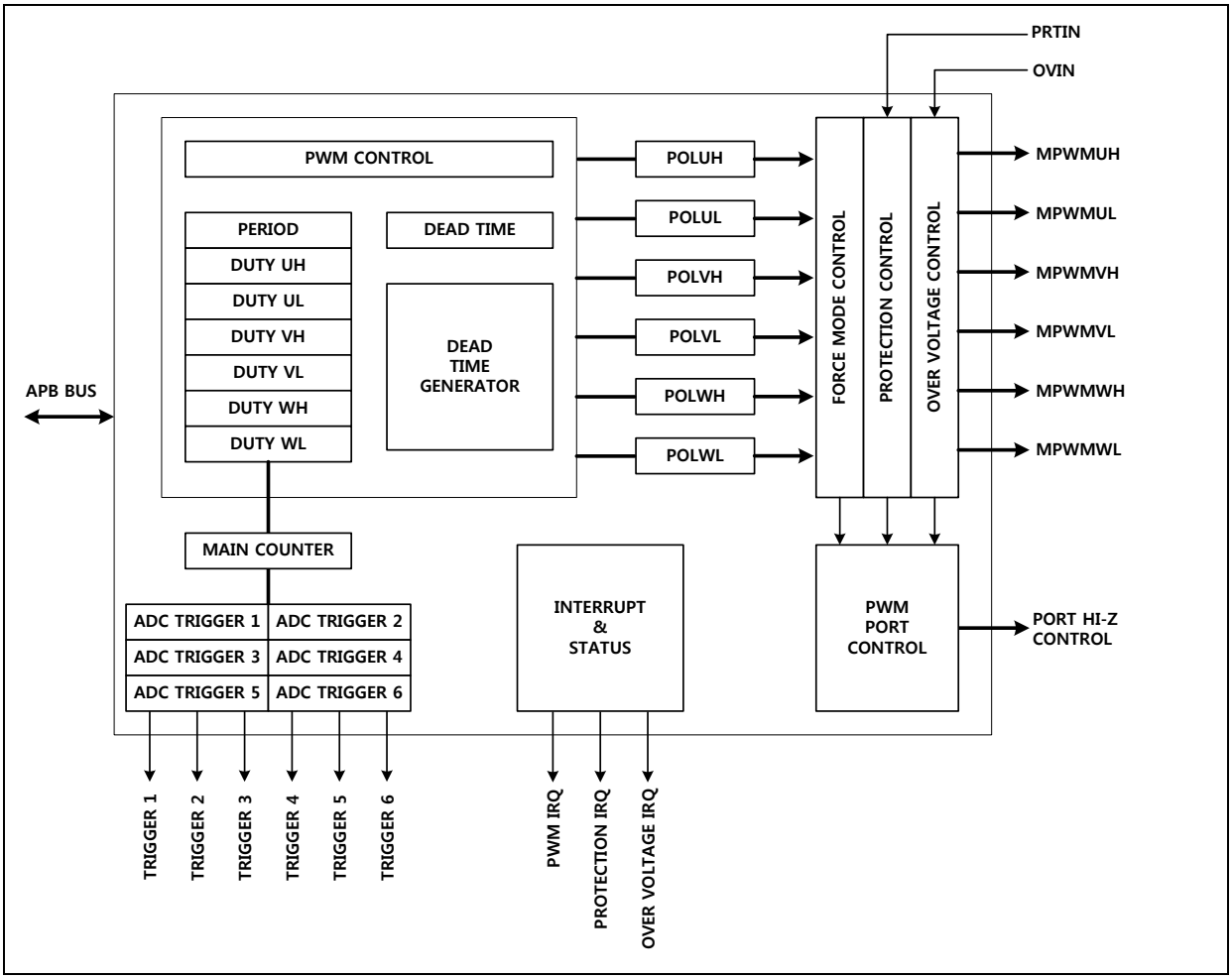


Figure 15-1 Block Diagram

Pin Description

Table 15-1 External Signals

PIN NAME	TYPE	DESCRIPTION
MPWMUH	O	MPWM Phase-U H-side output
MPWMUL	O	MPWM Phase-U L-side output
MPWMVH	O	MPWM Phase-V H-side output
MPWMVL	O	MPWM Phase-V L-side output
MPWMWH	O	MPWM Phase-W H-side output
MPWMWL	O	MPWM Phase-W L-side output
PRTIN	I	MPWM Protection Input
OVIN	I	MPWM Over-voltage Input

Registers

The base address of MPWM is shown in Table 15-2.

Table 15-2 MPWM Base Address

NAME	BASE ADDRESS
MPWM	0x4000_4000

Table 15-3 shows the register memory map.

Table 15-3 MPWM Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
MP.MR	0x0000	RW	MPWM Mode register	0x0000_0000
MP.OLR	0x0004	RW	MPWM Output Level register	0x0000_0000
MP.FOR	0x0008	RW	MPWM Force Output register	0x0000_0000
MP.PRD	0x000C	RW	MPWM Period register	0x0000_0002
MP.DUH	0x0010	RW	MPWM Duty UH register	0x0000_0001
MP.DVH	0x0014	RW	MPWM Duty VH register	0x0000_0001
MP.DWH	0x0018	RW	MPWM Duty WH register	0x0000_0001
MP.DUL	0x001C	RW	MPWM Duty UL register	0x0000_0001
MP.DVL	0x0020	RW	MPWM Duty VL register	0x0000_0001
MP.DWL	0x0024	RW	MPWM Duty WL register	0x0000_0001
MP.CR1	0x0028	RW	MPWM Control register 1	0x0000_0000
MP.CR2	0x002C	RW	MPWM Control register 2	0x0000_0000
MP.SR	0x0030	R	MPWM Status register	0x0000_0000
MP.IER	0x0034	RW	MPWM Interrupt Enable	0x0000_0000
MP.CNT	0x0038	R	MPWM counter register	0x0000_0001
MP.DTR	0x003C	RW	MPWM dead time control	0x0000_0000
MP.PCR0	0x0040	RW	MPWM protection 0 control register	0x0000_0000
MP.PSR0	0x0044	RW	MPWM protection 0 status register	0x0000_0080
MP.PCR1	0x0048	RW	MPWM protection 1 control register	0x0000_0000
MP.PSR1	0x004C	RW	MPWM protection 1 status register	0x0000_0000
-	0x0054	-	Reserved	-
MP.ATR1	0x0058	RW	MPWM ADC Trigger reg1	0x0000_0000
MP.ATR2	0x005C	RW	MPWM ADC Trigger reg2	0x0000_0000
MP.ATR3	0x0060	RW	MPWM ADC Trigger reg3	0x0000_0000
MP.ATR4	0x0064	RW	MPWM ADC Trigger reg4	0x0000_0000
MP.ATR5	0x0068	RW	MPWM ADC Trigger reg5	0x0000_0000
MP.ATR6	0x006C	RW	MPWM ADC Trigger reg6	0x0000_0000

MP.MR MPWM Mode Register

The Motor PWM operation mode register is a 16-bit register.

MP.MR=0x4000_4000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOTORB								UAO		TUP	BUP		MCHMOD		UPDOWN
0								0		0	0		00		0
RW								RW		RW	RW		RW		RW

15	MOTORB	0	Motor mode
		1	Normal mode
7	UAO	0	Update will be executed at designated timing.
		1	Update all duty, period register at once. When UPDATE set, Duty and Period registers are updated after two PWM clocks
5	TUP	0	Period, duty values are not updated at every period match.
		1	Period, duty values are updated at every period match.
4	BUP	0	Period, duty values are not updated at every bottom match
		1	Period, duty values are updated at every bottom match
2 1	MCHMOD	00	2 channels symmetric mode Duty H decides toggle high/low time of H-ch Duty L decides toggle high/low time of L-ch
		01	1 channel asymmetric mode Duty H decides toggle high time of H-ch Duty L decides toggle low time of H-ch L channel become the inversion of H channel
		10	1 channel symmetric mode Duty H decides toggle high/low time of H-ch L channel become the inversion of H channel
		11	Not valid (same with 00)
0	UPDOWN	0	PWM Up count mode (only available when MOTORB='1')
		1	PWM Up/Down count mode (This bit should be '1' if MOTORB='0')

After the initial PWM period and duty is set, the UAO bit should be set once for updating the setting value into internal operating registers. This action will help to transfer the setting data from the user interface register to the internal operating register. The UAO bit should stay at the set state for at least 2 PWM clock periods. If this does not occur, the update command can be missed and internal registers will retain the previous data.

The MCHMOD in the MP.MR field is only effective when MOTORB in MP.MR is a clear "0". Otherwise, the MCHMOD field value will be ignored internally and will retain the "00" value.

The UPDOWN in the MP.MR field is only effective when MOTORB in MP.MR is set to "1". Otherwise, the UPDOWN field value will be ignored internally and will retain the "1" value. In the Motor mode, the counter is always updown count operation.

MP.OLR MPWM Output Level Register

The PWM output level register is an 8-bit register. This register controls the active level of each PWM output port. The default active level is negated when the corresponding bit is set.

The normal level is defined in each operating mode.

MP.OLR=0x4000_4004

7	6	5	4	3	2	1	0
		WHL	VHL	UHL	WLL	VLL	ULL
0	0	0	0	0	0	0	0
		RW	RW	RW	RW	RW	RW

WHL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
VHL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
UHL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
WLL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
VLL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L
ULL	0	Normal Output = L / Active Output = H
	1	Normal Output = H / Active Output = L

The normal level is defined in each operating mode as shown in Table 15-4.

Table 15-4 MPWM Output Level Setting

PWM Output	Level	NORMAL mode		MOTOR Mode
		UP Mode	UPDOWN Mode	
WH	Default level	LOW	HIGH	LOW
	Active level	HIGH	LOW	HIGH
WL	Default level	LOW	LOW	HIGH
	Active level	HIGH	HIGH	LOW
VH	Default level	LOW	HIGH	LOW
	Active level	HIGH	LOW	HIGH
VL	Default level	LOW	LOW	HIGH
	Active level	HIGH	HIGH	LOW
UH	Default level	LOW	HIGH	LOW
	Active level	HIGH	LOW	HIGH
UL	Default level	LOW	LOW	HIGH
	Active level	HIGH	HIGH	LOW

The Polarity Control block is shown in Figure 15-2 using the WH signal polarity control example.

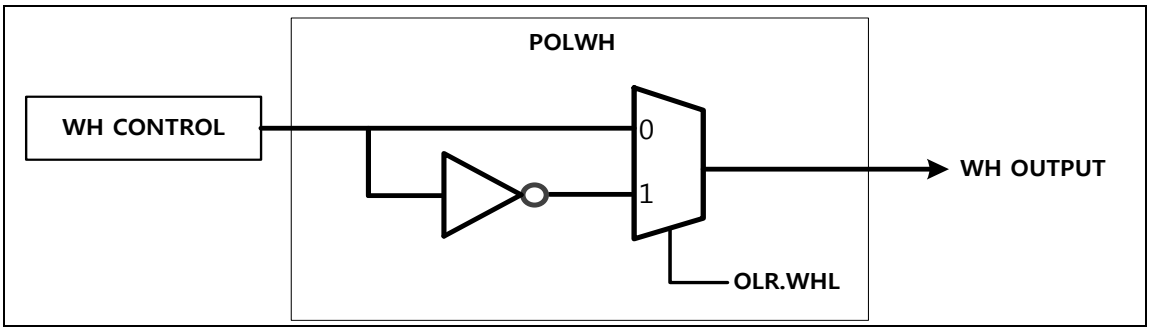


Figure 15-2 Polarity Control Block

MP.FOR MPWM Force Output Register

The PWM force output register is an 8-bit register. The PWM output level can be forced by an abnormal event externally or user-intended condition. When the forced condition occurs, each PWM output level which is programmed in the Force Output register will be forced.

MP.FOR=0x4000_4008

7	6	5	4	3	2	1	0
		WHFL	VHFL	UHFL	WLFL	VLFL	ULFL
0	0	0	0	0	0	0	0
		RW	RW	RW	RW	RW	RW

5	WHFL	Select WH Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
4	VHFL	Select VH Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
3	UHFL	Select UH Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
2	WLFL	Select WL Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
1	VLFL	Select VL Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’
0	ULFL	Select UL Output Force Level
		0 Output Force Level is ‘L’
		1 Output Force Level is ‘H’

MP.CR1 MPWM Control Register 1

The PWM Control Register 1 is a 16-bit register.

MP.CR1=0x4000_4028

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					IRQN										PWMEN
					000				0	0	0	0	0	0	0
					RW										RW

10 8	IRQN	IRQ interval number (Every 1~8th PRDIRQ,BOTIRQ,ATRn)
0	PWMEN	PWM enable When this bit set 0, the PWM block stay in reset state but user interface can be accessed. To operate the PWM block, this bit should be set 1.

Basically, PRDIRQ and BOTIRQ are generated every period. However, the interrupt interval can be controlled from 0 to 8 periods. When IRQN.CR1 = 0, the interrupt is requested every period, otherwise the interrupt is requested every (IRQN+1) times of period.

MP.CR2 MPWM Control Register 2

The PWM Control Register 2 is an 8-bit register.

MP.CR2=0x4000_402C

7	6	5	4	3	2	1	0
HALT							PSTART
0	0	0	0	0	0	0	0
RW							RW

7	HALT	PWM HALT (PWM counter stop but not reset) PWM outputs keep previous state	
0	PSTART	0	PWM counter stop and clear
		1	PWM counter start (will be resynced @PWM clock twice) PWMEN should be “1” to start PWM counter

MP.PRD MPWM Period Register

The PWM Period register is a 16-bit register.

MP.PRD=0x4000400C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERIOD															
0x0002															
RW															

15 0	PERIOD	16-bit PWM period. It should be larger than 0x0010 (if Duty is 0x0000, PWM will not work)

MP.DUH MPWM Duty UH Register

The PWM UH channel duty register is a 16-bit register.

MP.DUH=0x4000_4010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15 0	DUTY	16-bit PWM Duty for UH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MP.DVH MPWM Duty VH Register

The PWM VH channel duty register is a 16-bit register.

MP.DVH=0x4000_4014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15 0	DUTY	16-bit PWM Duty for VH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MP.DWHMPWM Duty WH Register

The PWM WH channel duty register is a 16-bit register.

MP.DWH=0x4000_4018

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for WH output.
0		It should be larger than 0x0001
		(if Duty is 0x0000, PWM will not work)

MP.DUL MPWM Duty UL Register

The PWM UL channel duty register is a 16-bit register.

MP.DUL=0x4000_401C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for UL output.
		It should be larger than 0x0001
0		(if Duty is 0x0000, PWM will not work)

MP.DVL MPWM Duty VL Register

The PWM VL channel duty register is a 16-bit register.

MP.DVL=0x4000_4020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for VL output.
		It should be larger than 0x0001
0		(if Duty is 0x0000, PWM will not work)

MP.DWL MPWM Duty WL Register

The PWM WL channel duty register is a 16-bit register.

MP.DWL=0x4000_4024

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUTY															
0x0001															
RW															

15	DUTY	16-bit PWM Duty for WL output. It should be larger than 0x0001
0		(if Duty is 0x0000, PWM will not work)

MP.IER MPWM Interrupt Enable Register

The PWM Interrupt Enable Register is an 8-bit register.

MP.IER=0x4000_4034

7	6	5	4	3	2	1	0
PRDIEN	BOTIEN	WHIE	VHIE	UHIE	WLIE	VLIE	ULIE
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

7	PRDIEN	PWM Counter Period Interrupt enable
		0 interrupt disable
		1 interrupt enable
6	BOTIEN	PWM Counter Bottom Interrupt enable
		0 interrupt disable
		1 interrupt enable
5	WHIE ATR6IE	WH Duty or ATR6 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
4	VHIE ATR5IE	VH Duty or ATR5 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
3	UHIE ATR4IE	UH Duty or ATR4 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
2	WLIE ATR3IE	WL Duty or ATR3 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
1	VLIE ATR2IE	VL Duty or ATR2 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable
0	ULIE ATR1IE	UL Duty or ATR1 Match Interrupt enable
		0 interrupt disable
		1 interrupt enable

MP.IER[5:0] control bits are shared by the duty match interrupt event and ADC trigger match interrupt event. When ADC trigger mode is disabled, the interrupt is generated by the duty match condition. In other instances, the interrupt is generated by the ADC trigger counter match condition. The ADC trigger mode is selected by the ATMOD bit field in the ATRm register.

MP.SR MPWM Status Register

The PWM Status Register is a 16-bit register.

MP.SR=0x4000_4030

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOWN	IRQCNT							PRDIF	BOTIF	DWHIF ATR6F	DVHIF ATR5F	DUHIF ATR4F	DWLIF ATR3F	DVLIF ATR2F	DULIF ATR1F
0	000			0	0	0	0	0	0	0	0	0	0	0	0
RW	RW							RW	RW	RW	RW	RW	RW	RW	RW

15	DOWN	0	PWM Count Up
		1	PWM Count Down
14 12	IRQCNT[2:0]		Interrupt count number of period match (Interval PRDIRQ mode)
7	PRDIF		PWM Period Interrupt flag(write “1” to clear flag)
		0	No interrupt occurred
		1	Interrupt occurred
6	BOTIF		PWM Bottom Interrupt flag(write “1” to clear flag)
		0	No interrupt occurred
		1	Interrupt occurred
5	DWHIF ATR6F		PWM duty WH interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR6 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
4	DVHIF ATR5F		PWM duty VH interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR5 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
3	DUHIF ATR4F		PWM duty UH interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR4 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
2	DWLIF ATR3F		PWM duty WL interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR3 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
1	DVLIF ATR2F		PWM duty VL interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR2 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred
0	DULIF ATR1F		PWM duty UL interrupt flag(write “1” to clear flag) (Duty interrupt is enabled if ATR1 was disabled)
		0	No interrupt occurred
		1	Interrupt occurred

MP.SR[5:0] status bits are shared by the duty match interrupt event and ADC trigger match interrupt event. When the ADC trigger mode is disabled, the interrupt is generated by the duty match condition. In other instances, the interrupt is generated by the ADC trigger counter match condition. The ADC trigger mode is selected by the ATMOD bit field in the ATRm register.

MP.CNT MPWM Counter Register

The PWM Counter Register is a 16-bit read-only register.

MP.CNT=0x4000_4038

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
0x0000															
RW															

	CNT	PWM Counter Value

MP.DTR MPWM Dead Time Register

The PWM Dead Time register is a 16-bit register.

MP.DTR=0x4000_403C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEN	PSHRT						DTCLK	DT							
0	0	0	0	0	0	0	0	0x00							
RW							RW	RW							

15	DTEN	Dead-time function enable 2 channel symmetric mode does not support dead time function. It should be disabled in 2 channel symmetric mode. 0 Disable Dead-time function 1 Enable Dead-time function
14	PSHRT	Protect short condition This function is effective only for 2 channel symmetric mode. For 1 channel mode, never activated on both H-side and L-side at same time. L-side is always opposite of H-side. 0 Enable output short protection function. (Turn off both output when both H-side and L-side are active.) 1 Disable output short protection function.
8	DTCLK	Dead-time prescaler 0 Dead time counter uses PWM CLK/4 1 Dead time counter uses PWM CLK/16
7 0	DT	Dead Time value (Dead time setting makes output delay of 'low to high transition' in normal polarity) 0x01 ~0xFF : Dead time

The Protect Short condition is only for internal PWM level, not for external PWM level. When the internal signal of H-side and L-side are the same high level, the protection short function works to force both H-side and L-side to low level.

MP.PCRn MPWM Protection 0,1 Control Register

The PWM Protection Control register is a 16-bit register.

MP.PCR0=0x4000_4040, MP.PCR1=0x4000_4048

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROTEN	PROTPOL				PROTD			PROTIE		WHPROTM	VHPROTM	UHPROTM	WLPROTM	VLPROTM	ULPROTM
0	0				000			0		0	0	0	0	0	0
RW	RW				RW			RW		RW	RW	RW	RW	RW	RW

15	PROT0EN	Enable Protection Input 0
14	PROT0POL	Select Protection Input Polarity 0: Low-Active 1: High-Active
10 8	PROTD	Protection Input debounce 0 – no debounce 1~7 – debounce by (MPWMCLK * PROTD[2:0])
7	PROTIE	Protection Interrupt enable 0 Disable protection interrupt 1 Enable protection interrupt
5	WHPROTM	Activate W-phase H-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
4	VHPROTM	Activate V-phase H-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
3	UHPROTM	Activate U-phase H-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
2	WLPROTM	Activate W-phase L-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
1	VLPROTM	Activate V-phase L-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value
0	ULPROTM	Activate U-phase L-side protection output 0 Disable Protection Output 1 Enable Protection Output with FOR value

Note: MP.PCR0 is related to the PRTIN pin and MP.PCR1 is related to OVIN.

MP.PSRn MPWM Protection 0,1 Status Register

The PWM Protection Status Register is a 16-bit register.

This register indicates which outputs are disabled. Users have the ability to set the output masks manually.

If PROTKEY is not written when writing any value, the written values are ignored.

MP.PSR0=0x4000_4044, MP.PSR1=0x4000_404C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROTKEY								PROTIF		WHPROTF	VHPROTF	UHPROTF	WLPROTF	VLPROTF	ULPROTF
-								0		0	0	0	0	0	0
WO								RC		RW	RW	RW	RW	RW	RW

15 8	PROTKEY	Protection Clear Access Key To clear flags, write the key with protection flag (PSR0 key is 0xCA and PSR1 key is 0xAC) Writing without PROTKEY prohibited.
7	PROTIF	Protection Interrupt status 0 No Protection Interrupt 1 Protection Interrupt occurred
5	WHPROT	Activate W-phase H-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
4	VHPROT	Activate V-phase H-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
3	UHPROT	Activate U-phase H-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
2	WLPROT	Activate W-phase L-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
1	VLPROT	Activate V-phase L-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled
0	ULPROT	Activate U-phase L-side protection flag 0 Protection not occurred. 1 Protection occurred or protection output enabled

If the PROTEN bit in the MP.PCRn register is enabled, on any asserting signal on the external protection pins, the PWM output will be prohibited when output values are defined in the MP.FOLR register.

Users can prohibit the output manually by writing the designated value into the MP.PSRn register.

Note: MP.PSR0 is related to the PRTIN pin and MP.PSR1 is related to OVIN.

MP.ATRm MPWM ADC Trigger Counter m Register

MP.ATR1	MPWM ADC Trigger Counter 1 Register
MP.ATR2	MPWM ADC Trigger Counter 2 Register
MP.ATR3	MPWM ADC Trigger Counter 3 Register
MP.ATR4	MPWM ADC Trigger Counter 4 Register
MP.ATR5	MPWM ADC Trigger Counter 5 Register
MP.ATR6	MPWM ADC Trigger Counter 6 Register

The PWM ADC Trigger Counter Register is a 32-bit register.

MP.ATR1=0x4000_4058
MP.ATR2=0x4000_405C
MP.ATR3=0x4000_4060
MP.ATR4=0x4000_4064
MP.ATR5=0x4000_4068
MP.ATR6=0x4000_406C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ATUDT		ATMOD		ATCNT																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0x0000															
												RW		RW		RW															

19	ATUDT	Trigger register update mode
		0 ADC trigger value applied at period match event (at the same time with period and duty registers update)
		1 Trigger register update mode When this bit set, written Trigger register values are sent to trigger compare block after two PWM clocks (through synchronization logic)
17	ATMOD	ADC trigger Mode register
16		00 ADC trigger Disable
		01 Trigger out when up count match
		10 Trigger out when down count match
		00 Trigger out when up-down count match
15	ATCNT	ADC Trigger counter
0		(it should be less than PWM period)

Functional Description

The MPWM includes three channels, each of which controls a pair of outputs that in turn can control an off-chip component. In normal PWM mode, each channel runs independently. 6 PWM outputs can be generated.

Each PWM output is built with various settings. Figure 15-3 shows the flow for generating PWM output signal.

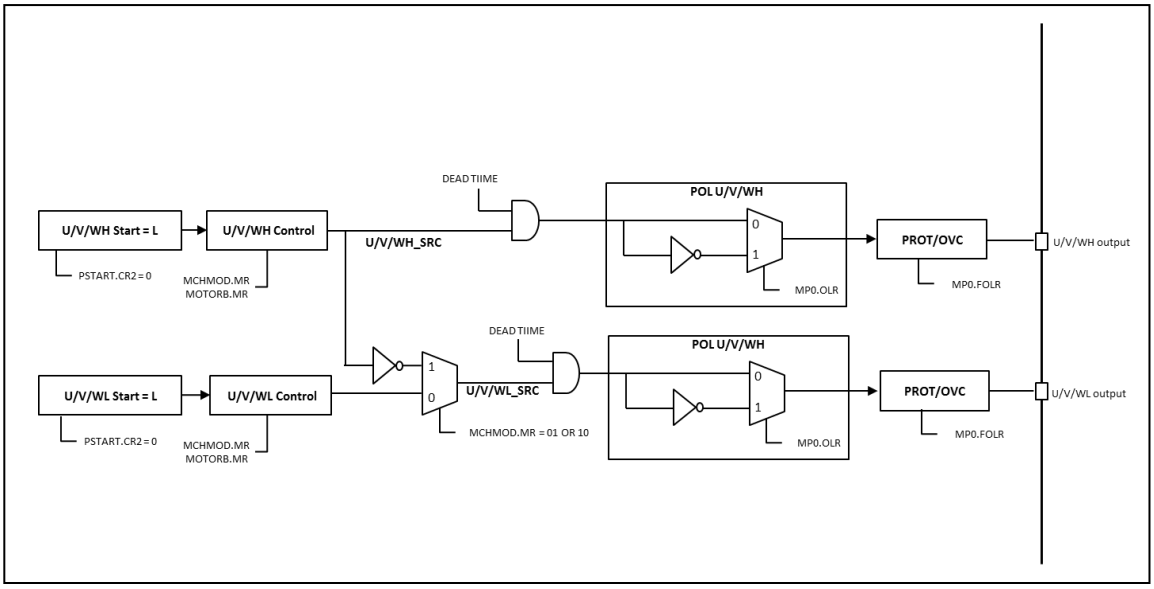


Figure 15-3 PWM Output Generation Chain

Normal PWM UP Count Mode Timing

In normal PWM mode, each channel runs independently. Six PWM outputs can be generated. The example waveform is shown in Figure 15-4. Before PSTART is activated, the PWM output will stay at default value L. When PSTART is enabled, the period counter starts up count until the MP.PRD count value. In the first period, the MPWM does not generate a PWM pulse.

The PWM pulse is generated from the second period. The active level is driven at the start of the counter value during duty value time.

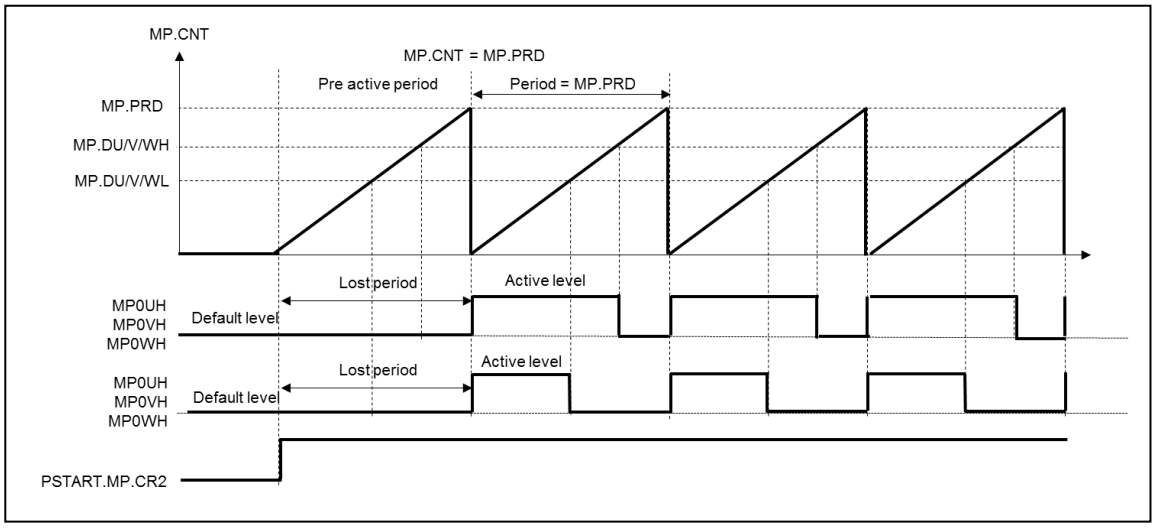


Figure 15-4 UP Count Mode Waveform (MOTORB=1, UPDOWN=0)

Normal PWM UP/DOWN Count Mode Timing

The basic operation of UP/DOWN count mode is the same as UP count mode except the one period is twice the UP count mode. The default active level is opposite in a pair PWM output. This output polarity can be controlled by the MP.OLR register.

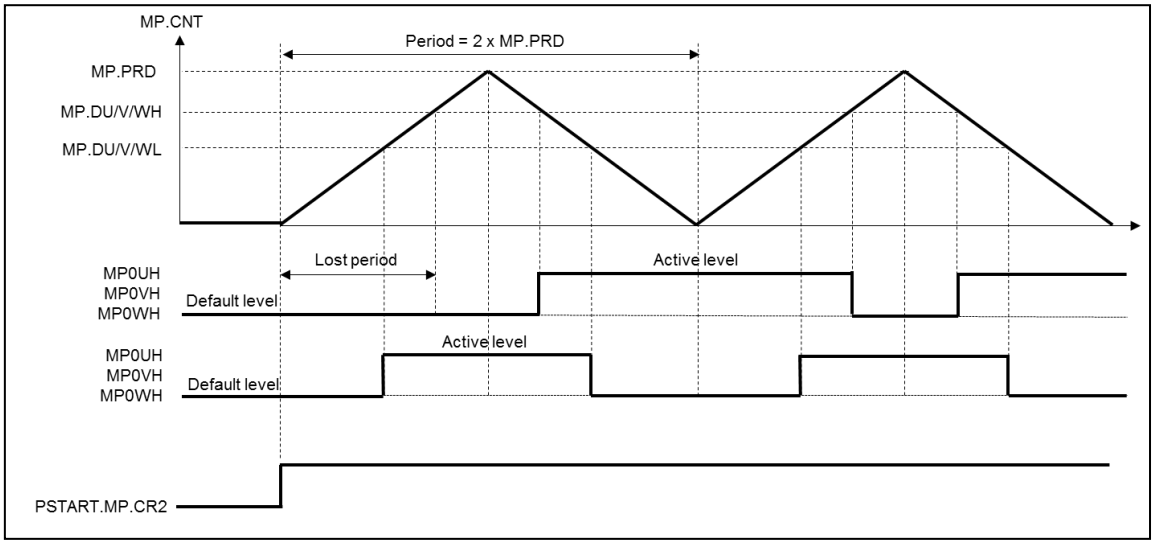


Figure 15.1 UP/DOWN Count Mode Waveform (MOTORB=0, MCHMOD=0, UPDOWN=1)

Motor PWM 2-Channel Symmetric Mode Timing

The motor PWM operation has three types of operating modes: 2-Channel Symmetric mode, 1-Channel Symmetric mode, and 1-Channel Asymmetric mode.

Figure 15-5 is for 2 channel symmetric mode waveform.

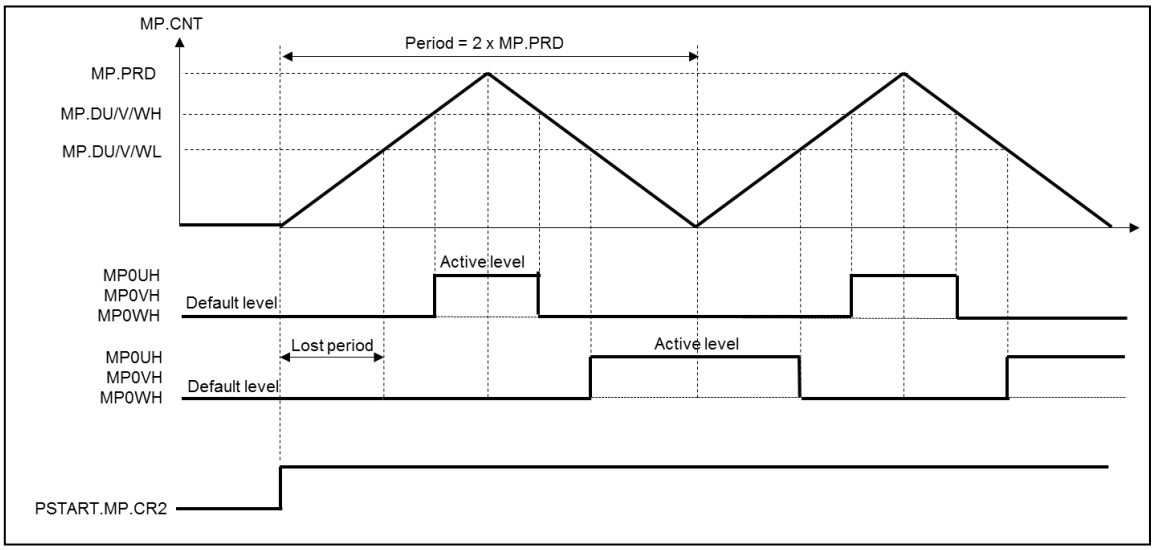


Figure 15-5 2-Channel Symmetric Mode Waveform (MOTORB=0, MCHMOD=00)

The default start level of both H-side and L-side is low. For the H-side, the PWM output level is changed to active level when the duty level is matched in up count period and is returned to the default level when the duty level is matched in down count period.

The symmetrical feature appears in each channel that is controlled by the corresponding duty register value.

Motor PWM 1-Channel Asymmetric Mode Timing

The 1-Channel Asymmetric mode makes asymmetric duration pulses which are defined by the H-side and L-side duty register. Therefore, the L-side signal is always the negative signal of H-side. During up count period, the H-side duty register matching condition makes the active level pulse and during down count period, the L-side duty register matching condition makes the default level pulse.

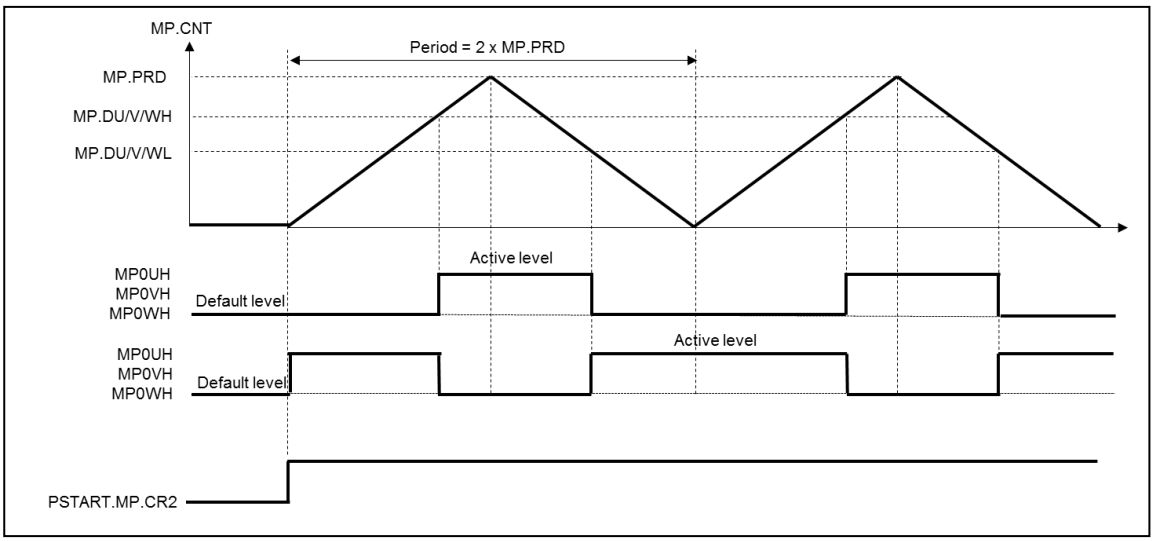


Figure 15.2 1-Channel Asymmetric Mode Waveform (MOTORB=0, MCHMOD=01)

The default start level of both H-side and L-side is low. For the H-side, the PWM output level is changed to active level when the H-side duty level is matched in up count period and is returned to the default level when the L-side duty level is matched in down count period.

When the PSTART is set, the L-side PWM output is changed to the active level, then the L-side PWM output is the inverse output of H-side output.

Motor PWM 1-Channel Symmetric Mode Timing

The 1-channel symmetric mode makes symmetric duration pulses which are defined by the H-side DUTY register. Therefore, the L-side signal is always the negative signal of H-side. During up count period, the H-side DUTY register matching condition makes the active level pulse and during down count period, the H-side DUTY register matching condition also makes the default level pulse.

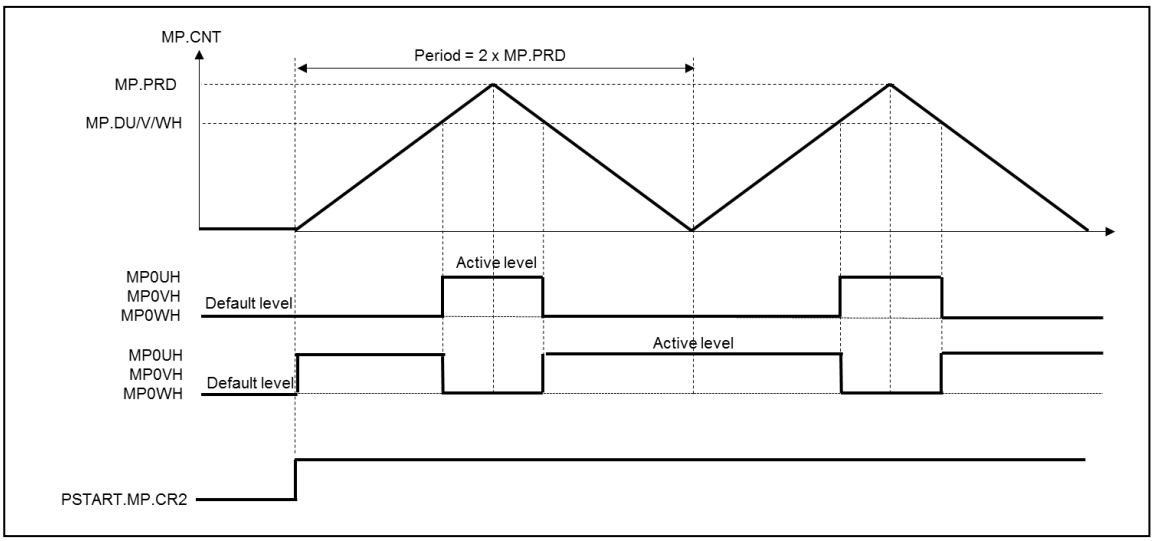


Figure 15.3 1-Channel Symmetric Mode Waveform (MOTORB=0, MCHMOD=10)

The default start level of both H-side and L-side is low. For the H-side, the PWM output level is changed to active level when the H-side duty level is matched in up count period and is returned to the default level when the H-side duty level is matched again in down count period.

When the PSTART is set, the L-side PWM output is changed to the active level, then the L-side PWM output is the inverse output of H-side output.

PWM Dead-time Operation

To prevent an external short condition, the MPWM provides dead time functionality. This function is only available for Motor PWM mode. When either H-side or L-side output changes to active level, dead time will be inserted if the DTEN.MP.DTR bit is enabled.

The duration of dead time is determined by the value in the DT.MP.DTR[7:0] field.

When DTCLK = 0, the dead time duration = DT[7:0] * (PWM clock period * 4)

When DTCLK = 1, the dead time duration = DT[7:0] * (PWM clock period * 16)

When the PWM counter reaches duty value, the PWM output is masked and the dead time counter starts to run. When the dead time counter reaches the value in the DT[7:0] register, the output mask is disabled.

Figure 15-6 is an example of dead time operation in 1-Channel Symmetric mode.

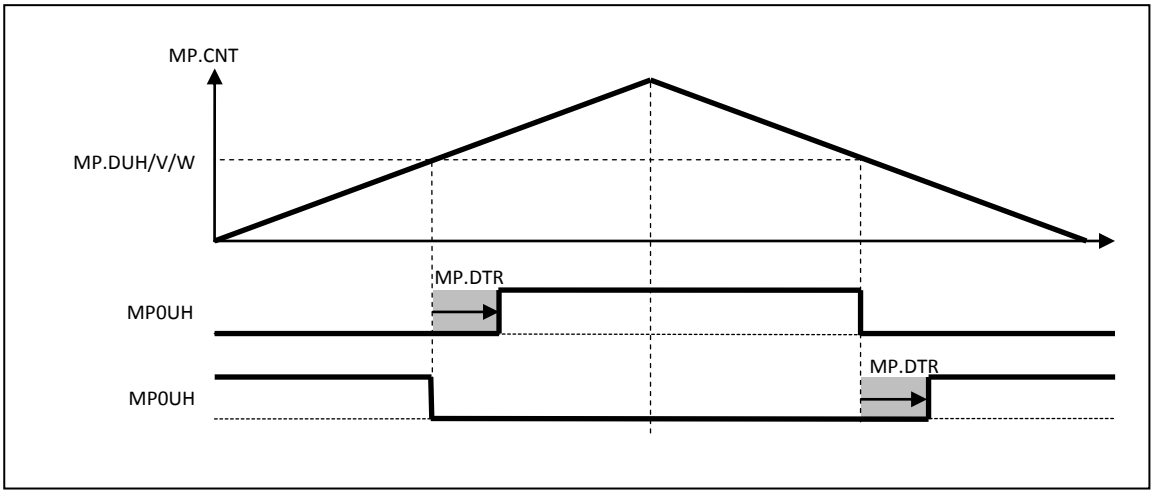


Figure 15-6 PWM Dead-time Operation Timing Diagram (Symmetric Mode)

Figure 15-7 shows an example of 1-Channel Asymmetric mode operation.

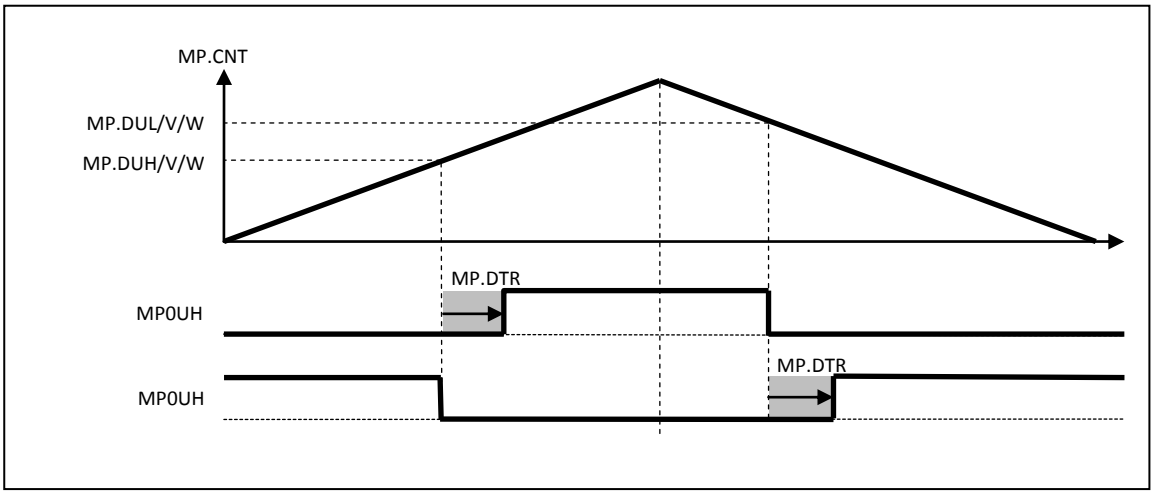


Figure 15-7 PWM Dead-time Operation Timing Diagram (Asymmetric Mode)

For 2-Channel Symmetric mode, the dead time function is not available. Therefore, the dead condition is generated by each channel's duty control.

MPWM Dead-time Timing Examples in Special Case

The following figures show how dead-time operates.

An example of normal dead time is explained. Dead time masking is activated at duty match time and the dead time counter runs. When the dead time counter reaches the dead time value, the mask is disabled.

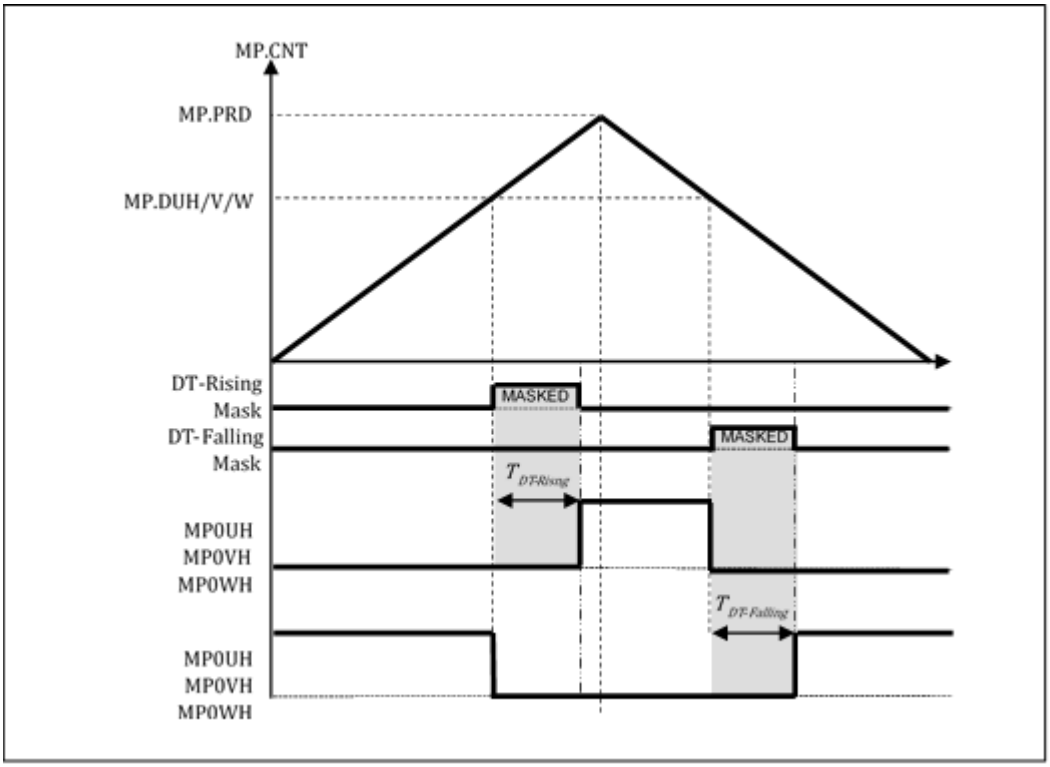


Figure 15.4 Normal Dead-time Operation (T_{DUTY}>T_{DT})

The following images show special instances of dead time configuration.

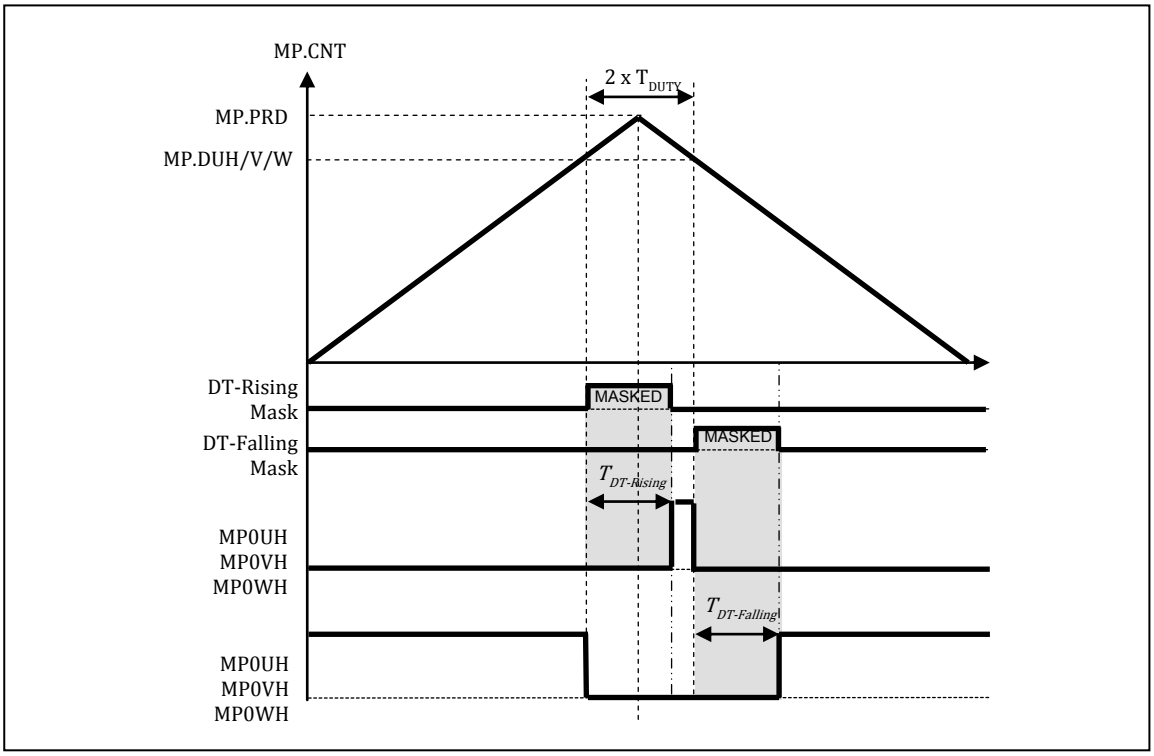


Figure 15-8 Minimum H-side Pulse Timing (T_{DUTY}<T_{DT}<2xT_{DUTY})

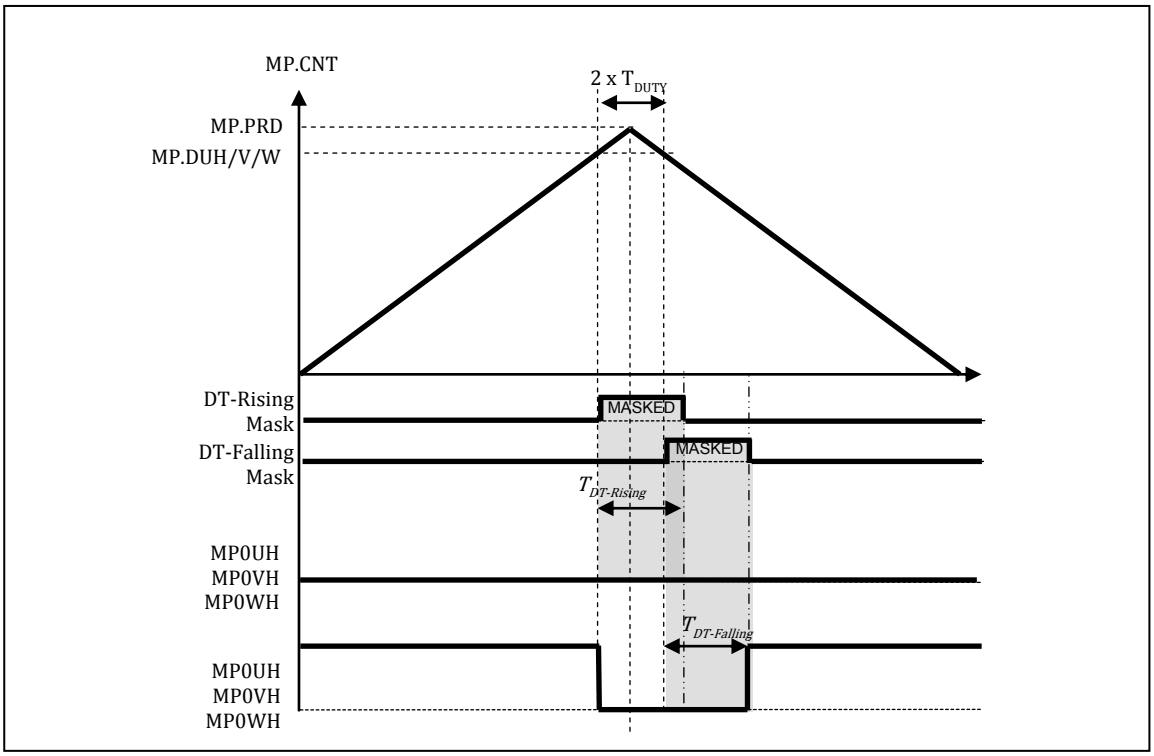


Figure 15-9 Zero H-side Pulse Timing (T_{DT}>2xT_{DUTY})

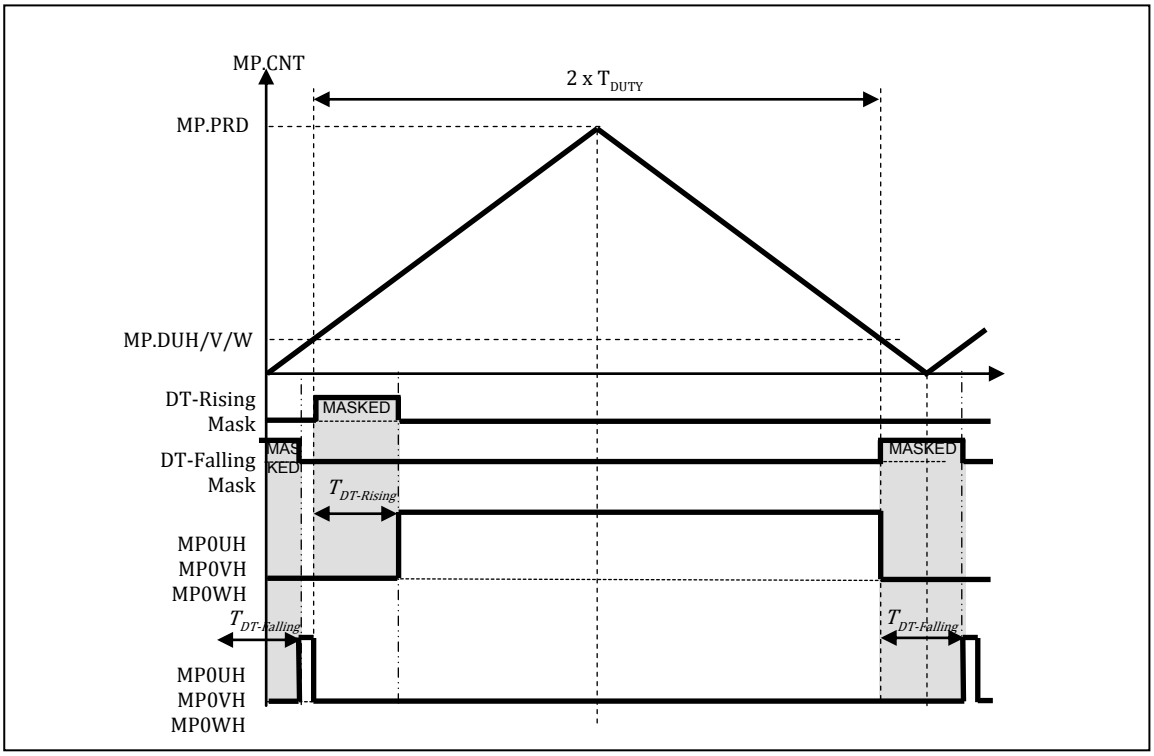


Figure 15-10 Minimum L-side Pulse Timing (T_{DT}<Period-T_{DUTY})

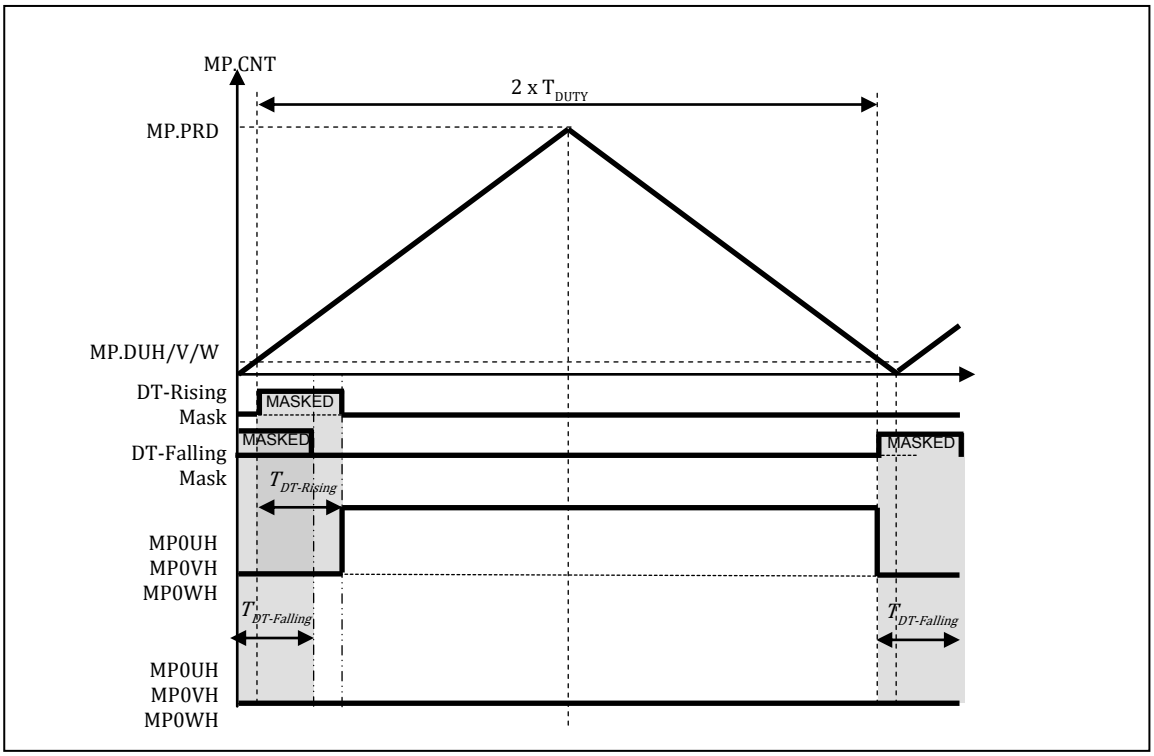


Figure 15-11 Zero L-side Pulse Timing (T_{DT}>Period-T_{DUTY})

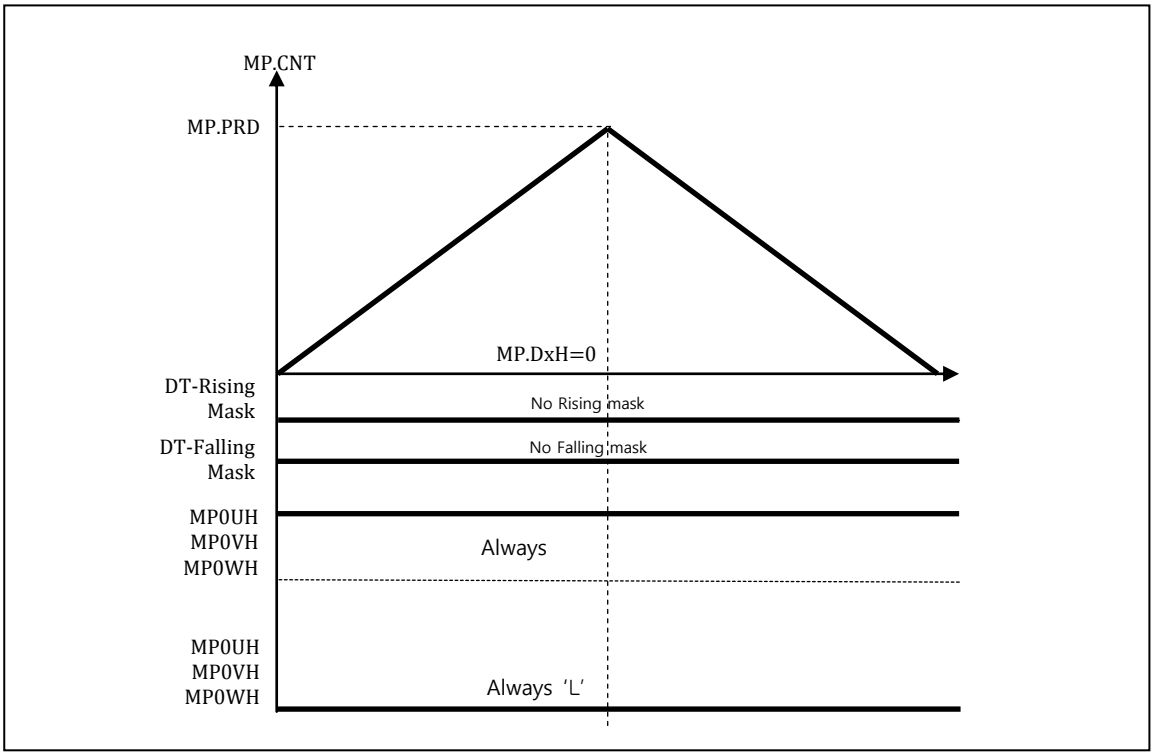


Figure 15-12 H-side Always On (T_{DUTY}=Period: Dead-time Disabled)

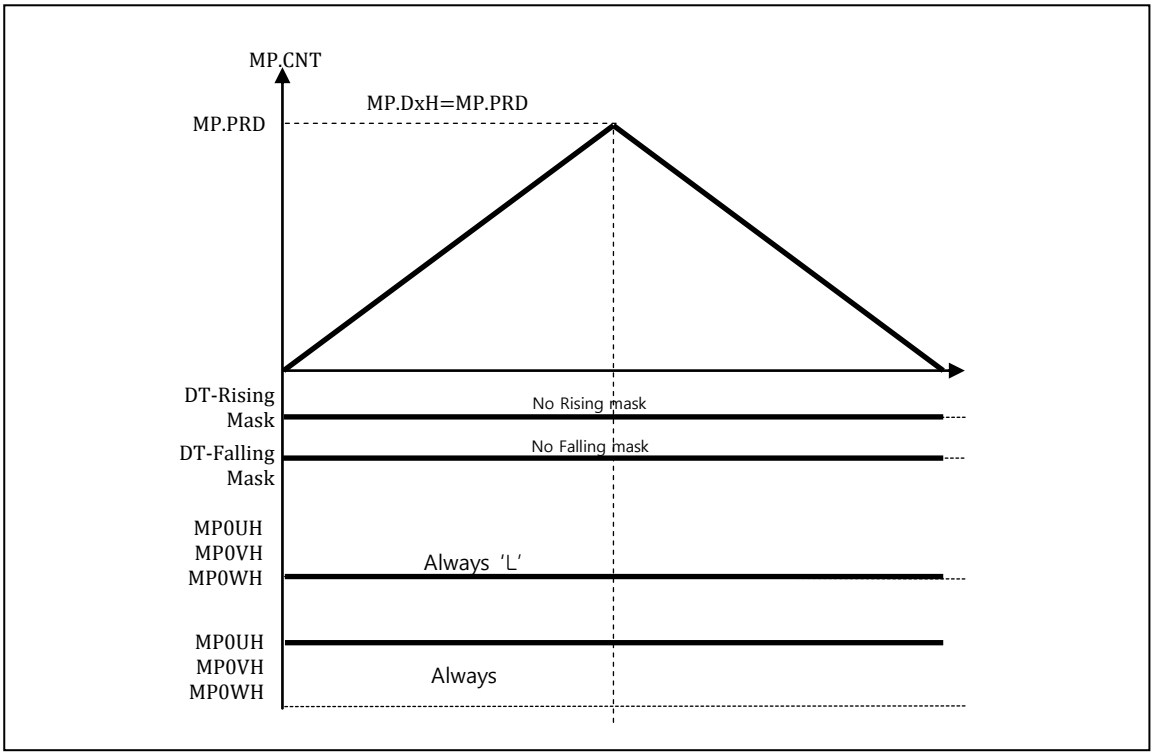


Figure 15-13 L-side Always On (T_{DUTY}=0ô Dead-time Disabled)

Symmetrical Mode vs Asymmetrical Mode

In Symmetrical mode, the wave form is between the up and down counters. The same duty value is used for both the up and down counter matches. The on time and off time is the same between the up and down counters. The end result is that in a period, the duty time is centered in the period.

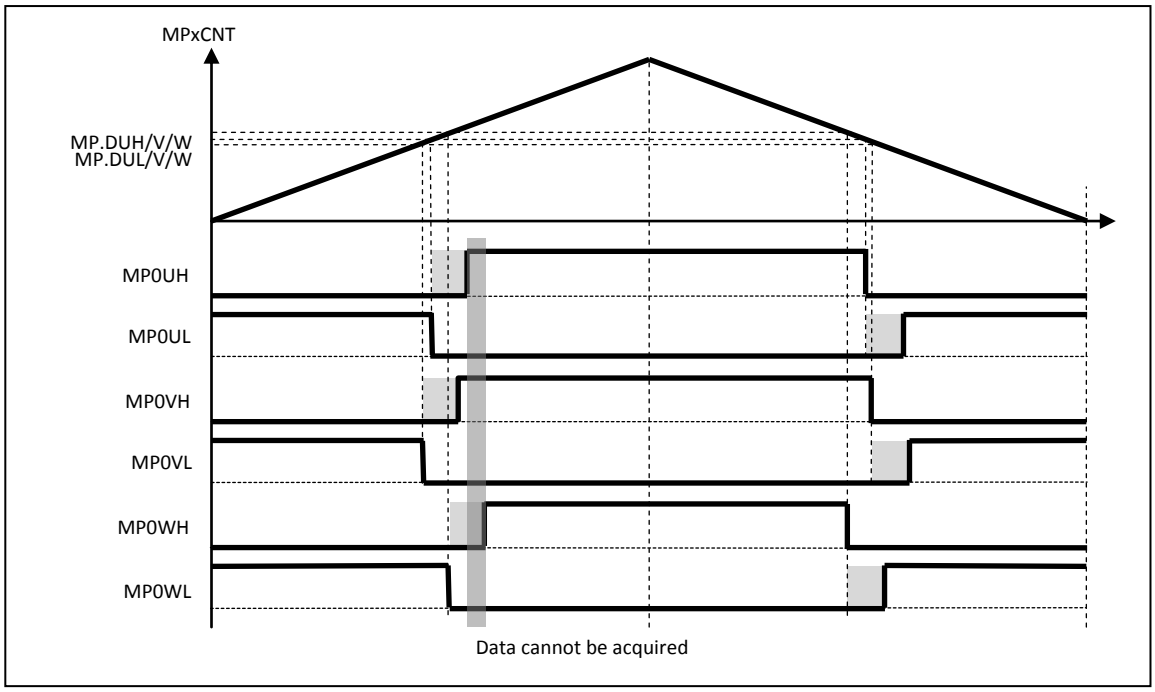


Figure 15-14 Symmetrical PWM Timing

In Asymmetrical mode, the wave from is not symmetric between the up and down counters. The Duty High is used to match on the up counter and the Duty Low is used to match on the down counter.

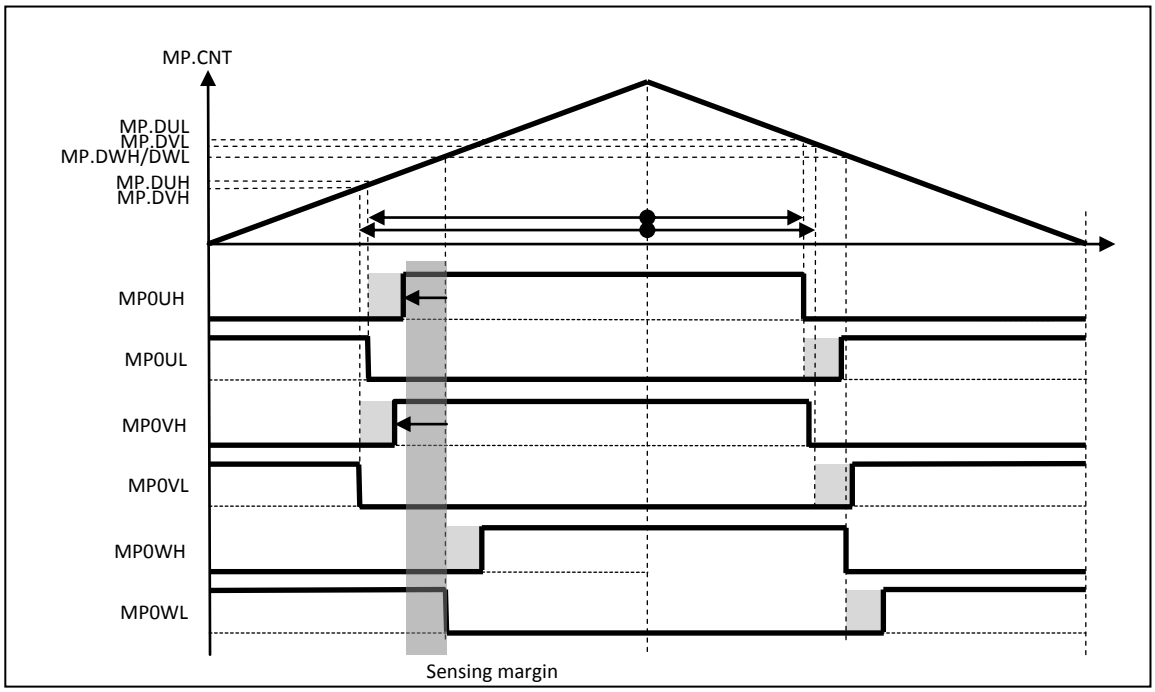


Figure 15-15 Asymmetrical PWM Timing and Sensing Margin

Description of ADC Triggering Function

A total of six ADC trigger timing registers are provided. This dedicated register triggers a signal to start ADC conversion. The conversion channel of ADC is defined in the ADC Control register.

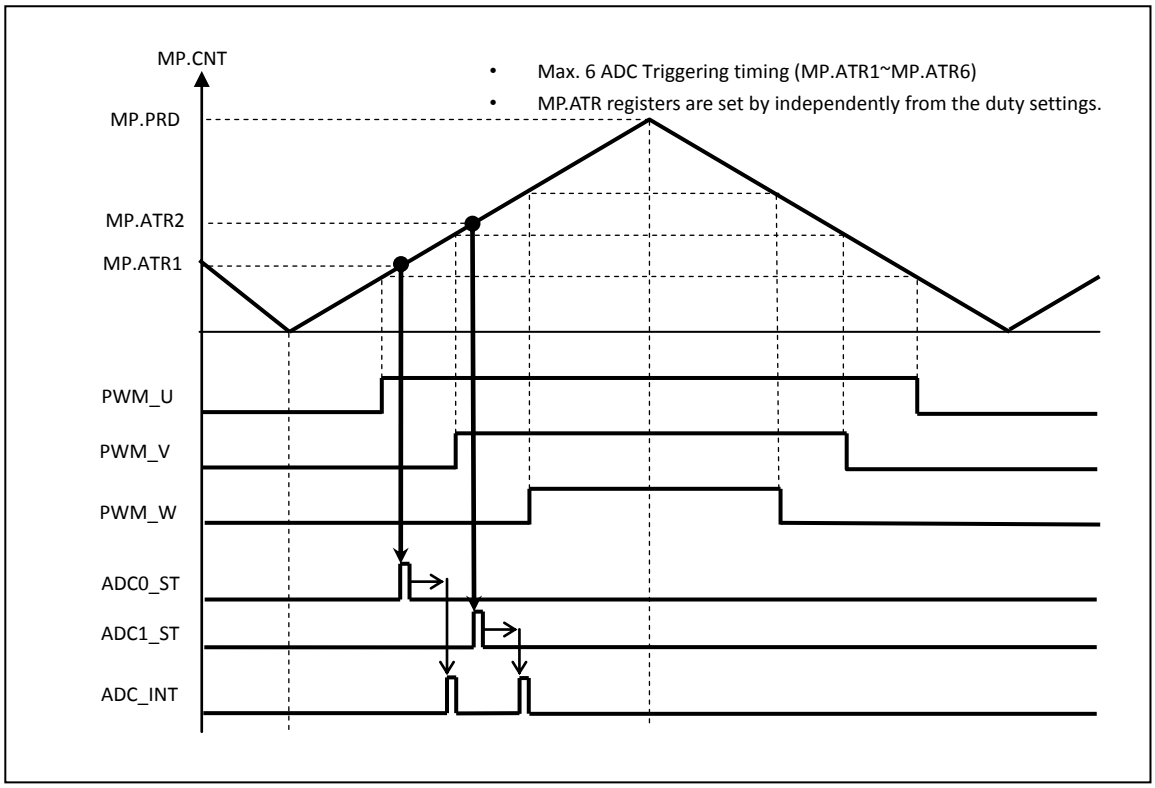


Figure 15-16 ADC Triggering Function Timing Diagram

Figure 15-17 shows an example of ADC Data acquisition.

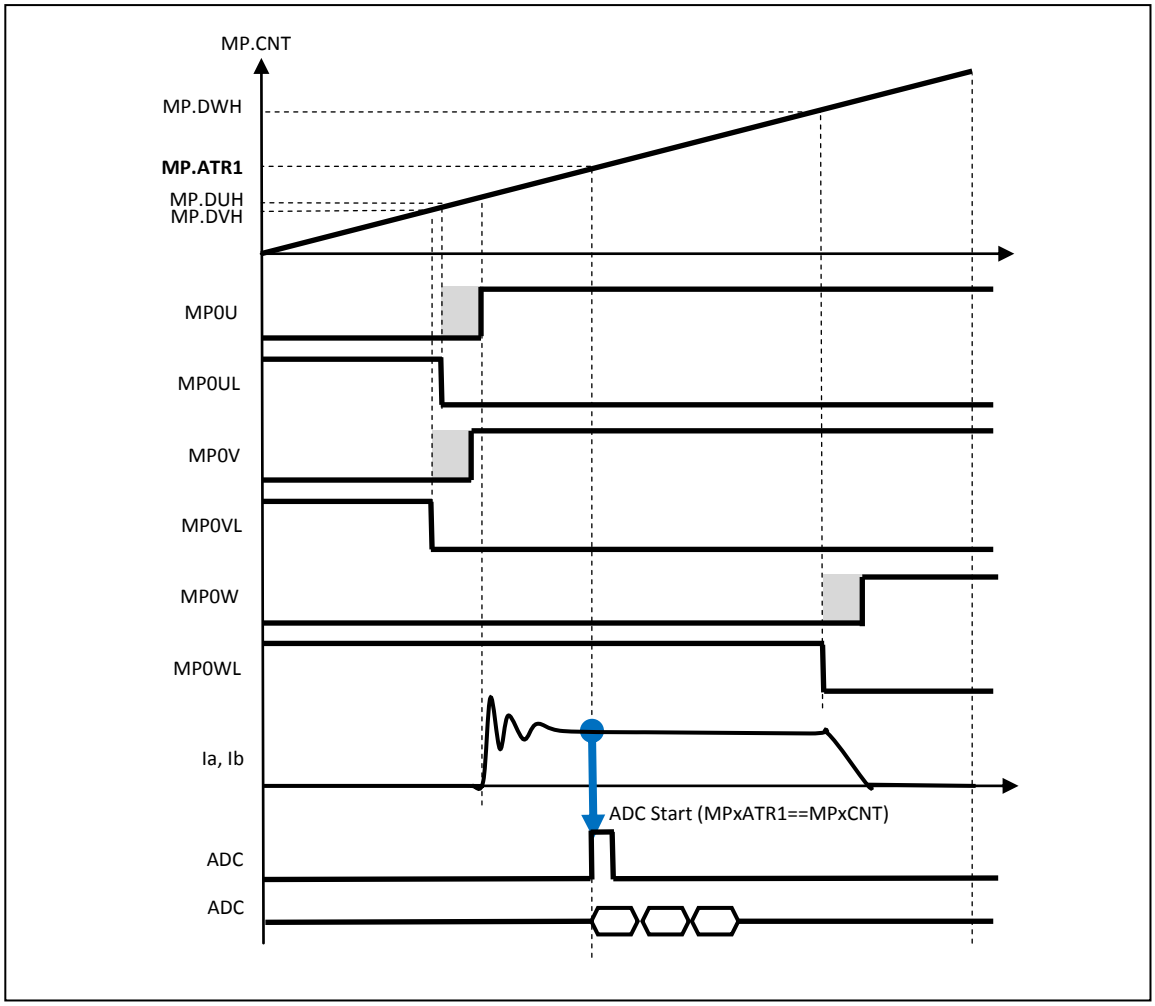


Figure 15-17 An Example of ADC Acquisition Timing by Event from MPWM

Interrupt Generation Timing

Each timing event can make an interrupt request to the CPU.

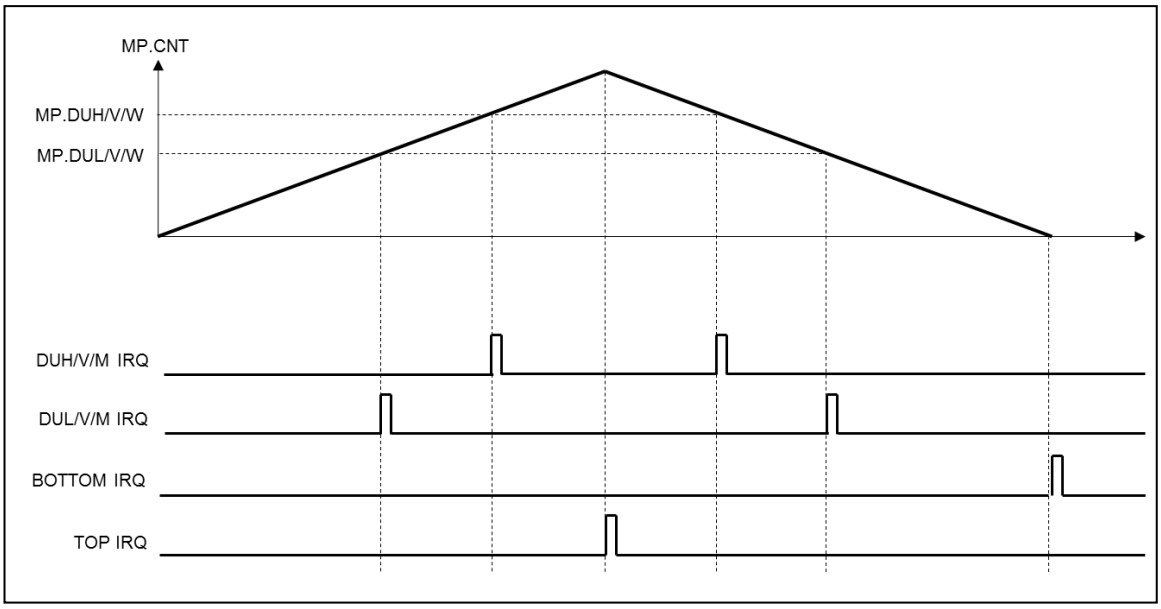


Figure 15-18 Interrupt Generation Timing

16. Divider (DIV64)

Overview

The divider module provides the hardware divider the ability to accelerate complicated calculations. This divider is a sequential 64-bit/32-bit divider and requires 32 clock cycles for one operation.

The equation for this operation is:

(AREGH,AREGL)/BREG = (QREGH,QREGL)

- Unsigned 64-bit dividend
- Unsigned 32-bit divisor
- Unsigned 64-bit quotient
- Unsigned 32-bit remainder
- Unsigned 32-cycle operating time

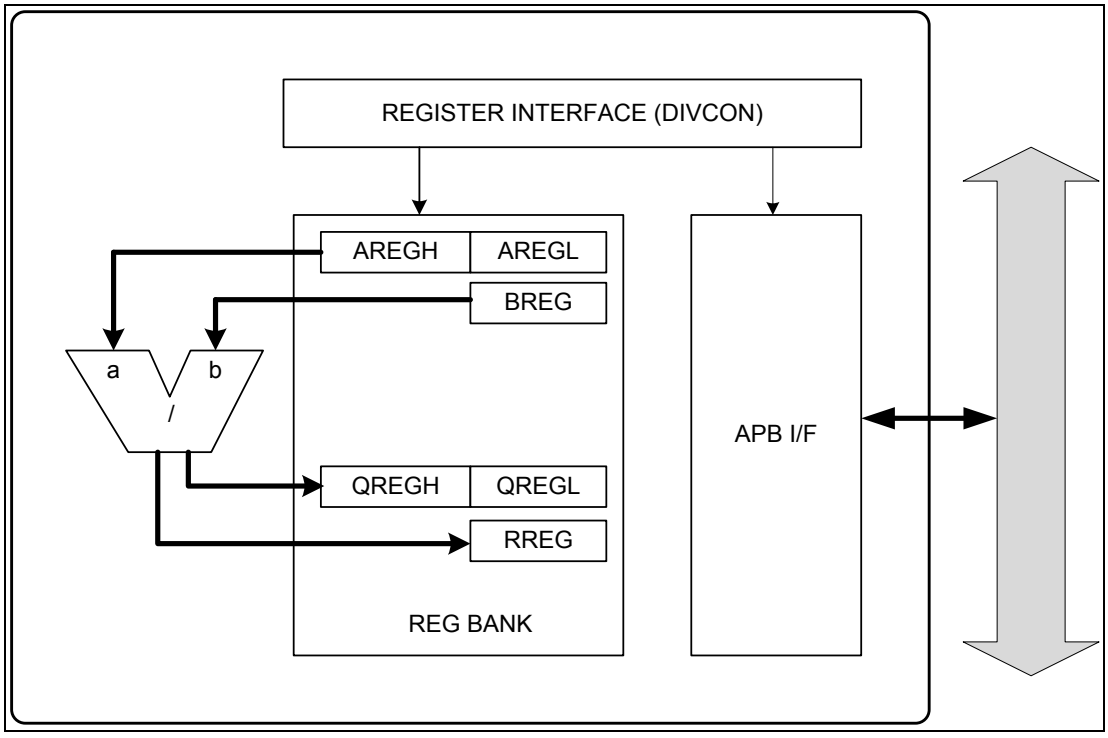


Figure 16-1 Block Diagram

Registers

The base address of the divider is 0x4000_0500 and the register map is described in Table 16-1.

Table 16-1 DIV64 Base Address

NAME	BASE ADDRESS
DIV64	0x4000_0500

Table 16-2 DIV64 Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
CR	0x0000	RW	DIV control register	0x00000000
AREGL	0x0004	RW	Most 32bit data register for dividend	0x00000000
AREGH	0x0008	RW	Least 32bit data register for dividend	0x00000000
BREG	0x000C	RW	32bit data register for divisor	0x00000000
QREGL	0x0010	R	Most 32bit data register for quotient	0x00000000
QREGH	0x0014	R	Least 32bit data register for quotient	0x00000000
RREG	0x0018	R	32bit data register for remainder	0x00000000

CR Divider Control Register

The DIVCON register controls the hardware divider module.

CR=0x4000_0500

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					I_ERROR	BUSY	DONE				MODE				START
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
					RO	RO	RO				RW				RW

10	I_ERROR	Divide by zero flag	
		0	Not divide by zero
		1	Divide by zero
9	BUSY	Divider is now under operating	
		0	Divider is not busy
		1	Divider is busy
8	DONE	Divider operation done flag	
		0	Divider is now operating
		1	Divider operation is done
4	MODE	Start operation mode	
		0	START bit write operation will trigger the divide operation
		1	BREG register write operation will trigger the divide operation
0	START	Divide operation start command. This bit is effective when MODE bit is 0	
		0	No effect
		1	Start divider

AREGL AREG (Dividend) Lower 32-bit Register

The lower 32-bit value of dividend should be written to this register.

AREGL=0x4000_0504

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AREGL[31:0]																															
0x0000_0000																															
RW																															

31 0	AREGL	Lower 32 bit value for dividend A.

AREGH AREG (Dividend) High 32-bit Register

The high 32-bit value of dividend should be written to this register.

AREGH=0x4000_0508

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AREGH[63:32]																															
0x0000_0000																															
RW																															

31 0	AREGH	High 32 bit value for dividend A.

BREG BREG (Divisor) Register

The 32-bit value of the divisor should be written to this register.

When the MODE bit is set to 1, the divide operation is started automatically as soon as the value is written to this register.

BREG=0x4000_050C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BREG[31:0]																															
0x0000_0000																															
RW																															

31 0	BREG	32 bit value for divisor B.

QREGL QREG (Quotient) Lower 32-bit Register

The divider stores the lower 32-bit value of the quotient in this register.

QREGL=0x4000_0510

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QREGL[31:0]																															
0x0000_0000																															
R																															

31 0	QREGL	Lower 32 bit value for quotient.

QREGH QREG (Quotient) High 32-bit Register

The divider stores the high 32-bit value of the quotient in this register.

QREGH=0x4000_0514

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QREGH[63:32]																															
0x0000_0000																															
R																															

31 0	QREGH	High 32 bit value for quotient.

RREG RREG (Remainter) Register

The divider stores the 32-bit value of the remainder in this register.

RREG=0x4000_0518

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RREG[31:0]																															
0x0000_0000																															
R																															

31 0	RREG	32 bit value for remainder.

17. 12-Bit A/D Converter

Introduction

The ADC block consists of 1 ADC unit, with the following features:

- 12 channels of analog inputs
- Single and Continuous conversion mode
- Up to 8 times sequential conversion support
- Software trigger support
- 3 internal trigger sources support (Soft-trig, MPWM, Timers)
- Adjustable sample and hold time

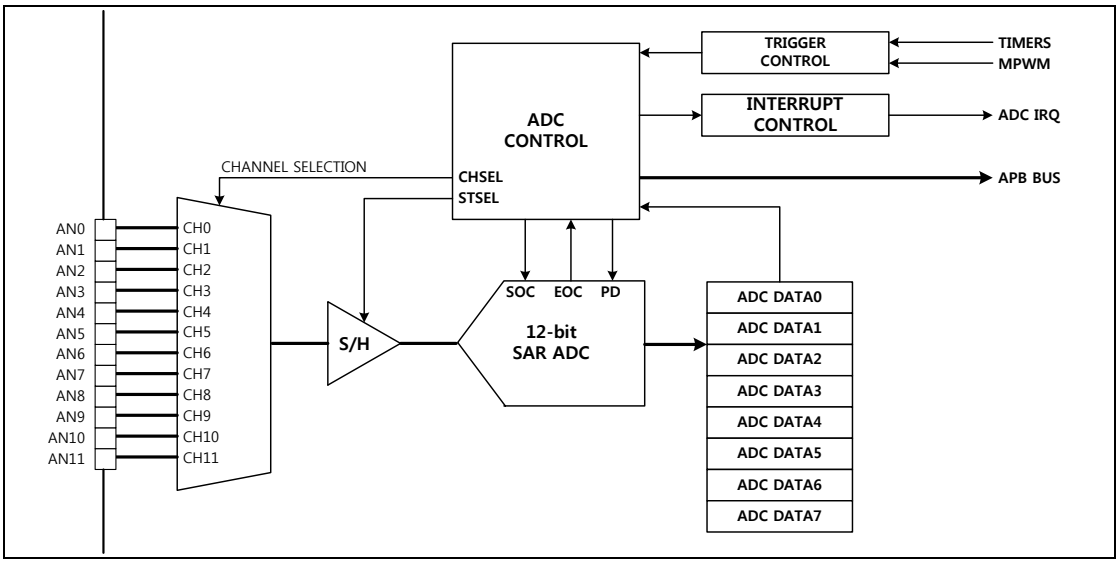


Figure 17-1 Block Diagram

Pin Description

Table 17-1 External Signal

PIN NAME	TYPE	DESCRIPTION
VDD	P	Analog Power(2.4V~5V)
VSS	P	Analog GND
AN0	A	ADC Input 0
AN1	A	ADC Input 1
AN2	A	ADC Input 2
AN3	A	ADC Input 3
AN4	A	ADC Input 4
AN5	A	ADC Input 5
AN6	A	ADC Input 6
AN7	A	ADC Input 7
AN8	A	ADC Input 8
AN9	A	ADC Input 9
AN10	A	ADC Input 10
AN11	A	ADC Input 11

Registers

The base address of the ADC unit is shown in Table 17-2.

Table 17-2 ADC Base Address

NAME	BASE ADDRESS
ADC	0x4000_B000

Table 17-3 ADC Register Map

NAME	OFFSET	TYPE	DESCRIPTION	RESET VALUE
AD.MR	0x0000	RW	ADC Mode register	0x00
AD.CSCR	0x0004	RW	ADC Current Sequence/Channel register	0x00
AD.CCR	0x0008	RW	ADC Clock Control register	0x80
AD.TRG	0x000C	RW	ADC Trigger Selection register	0x00
-	0x0010	-	Reserved	-
-	0x0014	-	Reserved	-
AD.SCSR	0x0018	RW	ADC Burst mode channel select	0x00
AD.CR	0x0020	RW	ADC Control register	0x00
AD.SR	0x0024	RW	ADC Status register	0x00
AD.IER	0x0028	RW	ADC Interrupt Enable register	0x00
-	0x002C	-	Reserved	-
AD.DR0	0x0030	R	ADCn Sequence 0 Data register	0x00
AD.DR1	0x0034	R	ADCn Sequence 1 Data register	0x00
AD.DR2	0x0038	R	ADCn Sequence 2 Data register	0x00
AD.DR3	0x003C	R	ADCn Sequence 3 Data register	0x00
AD.DR4	0x0040	R	ADCn Sequence 4 Data register	0x00
AD.DR5	0x0044	R	ADCn Sequence 5 Data register	0x00
AD.DR6	0x0048	R	ADCn Sequence 6 Data register	0x00
AD.DR7	0x004C	R	ADCn Sequence 7 Data register	0x00

AD.MR ADC Mode Register

The ADC Mode registers are 32-bit registers.

This register configures the ADC operation mode. This register should be writen first before the other registers.

AD.MR=0x4000_B000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																STSEL					SEQCNT			ADEN	ARST	ADMOD				TRGSEL	
																0x0					0x0			0x0	0x0	0x0				0x0	
																RW					RW			RW	RW	RW				RW	

16 12	STSEL	Sampling Time Selection ADC Sample & Hold circuit sampling time become (2 + STSEL[4:0]) MCLK cycles Minimum sampling time is 2 MCLK cycles			
10 8	SEQCNT	Number of coversion in a sequence If ADMOD is 2'h0 and SEQCNT is not 3'h0, CSEQN will be increased up to SEQCNT by trigger event. SEQCNT is burst count In burst mode. See 14.4.1~3 to know between Triggers and SEQCNT in burst conversion mode and in single sequential convertion mode			
		000	1st single sequential conversion or 1 burst count	100	5st single sequential conversion or 5 burst count
		001	2nd single sequential conversion or 2 burst counts	101	6st single sequential conversion or 6 burst counts
		010	3rd single sequential conversion or 3 burst counts	110	7st single sequential conversion or 7 burst counts
		011	4st single sequential conversion or 4 burst counts	111	8st single sequential conversion or 8 burst counts
7	ADEN	0	ADC disable		
		1	ADC enable		
6	ARST	0	Stop at the end of sequence. Should set ASTART as 1 to restart again		
		1	Restart at the end of sequence.		
5 4	ADMOD	00	Single conversion mode (single sequential conversion mode when SEQCNT is not 0x0)		
		01	Burst conversion mode		
		10	Reserved		
		11	Reserved		
1 0	TRGSEL	00	Event Trigger Disabled/Soft-Trigger Only		
		01	Timer Event Trigger		
		10	MPWM Event Trigger		
		11	Reserved		

If ADCMOD was set for Burst Mode, ADC channels are controlled by SEQ0CH ~ SEQ7CH. Sequential mode always start from SEQ0CH. (In 3 sequential mode, Analog inputs of channels which assigned at SEQ0CH, SEQ1CH and SEQ2CH are converted sequentially).

AD.CSCR ADC Current Sequence/Channel Register

ADC Current Sequence/Channel registers are 7-bit registers. This register consists of Current Sequence Numbers and Current Active Channel values. A Current Sequence Number (CSEQN) can be written to change the next sequence number. When you write CSEQN as 0x7 when CSEQN is 0x3 and AD.MR.SEQCNT is 0x7, the next sequence number is 0x7. AD converts the AD.SCSR.SEQ7CH channel and the 4,5,6 sequences are skipped. This register should be written first, before AD.SCSR.

AD.CSCR=0x4000_B004

7	6	5	4	3	2	1	0
-	CSEQN			CACH			
-	0x0			0x0			
-	RW			RO			

6 4	CSEQN	Current Sequence Number, can write when not abusy AD starts conversion the AD.SCSR.SEQ*CH's channel by AD.TRG.SEQTRG* in Single sequential mode. AD starts conversion the AD.SCSR.SEQ*CH's channel by AD.TRG.BSTTRG in Burst mode
	0000	Current Sequence is 0 the AD.SCSR.SEQ0CH's channel is converted by AD.TRG.SEQTRG0 in Single sequential mode or by AD.TRG.BSTTRG in Burst mode
	0001	Current Sequence is 1
	0010	Current Sequence is 2
	0011	Current Sequence is 3
	0100	Current Sequence is 4
	0101	Current Sequence is 5
	0110	Current Sequence is 6
	0111	Current Sequence is 7
3 0	CACH	Current Active Channel
	0000	ADC channel 0 is active
	0001	ADC channel 1 is active
	0010	ADC channel 2 is active
	0011	ADC channel 3 is active
	0100	ADC channel 4 is active
	0101	ADC channel 5 is active
	0110	ADC channel 6 is active
	0111	ADC channel 7 is active
	1000	ADC channel 8 is active
	1001	ADC channel 9 is active
	1010	ADC channel 10 is active
	1011	ADC channel 11 is active
	1100	reserved
	1101	reserved
	1110	reserved
	1111	reserved

AD.CCR ADC Clock Control Register

The ADC Control registers are 16-bit registers. The ADC Clock Control Register sets the ADC clock for determining the period to execute a conversion.

AD.CCR=0x4000_B008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPDA	CLKDIV							ADCPD	EXTCLK	CLKINVT	-				
0	0x00							1	0	0					
RW	RW							RW	RW	RW					

15	ADCPDA	ADC R-DAC disable to save power Don't set "1" here(it's optional bit)
14 8	CLKDIV[6:0]	ADC clock divider when EXTCLK is '0'. ADC clock = system clock/CLKDIV CKDIV=0 : ADC clock=system clock CKDIV=1 : ADC clock=stop
7	ADCPD	ADC Power Down 0 – ADC normal mode 1 – ADC Power Down mode
6	EXTCLK	Select if ADC uses external clock. 0 – internal clock(CKDIV enabled) 1 – external clock(SCU clock-MCCR4)
5	CLKINVT	Divided clock inversion(optional bit) 0 – duty ratio of divided clock is larger than 50% 1 – duty ratio of divided clock is less than 50%

AD.TRG ADC Trigger Selection Register

ADC Trigger registers are 32-bit registers.

For the ADC Trigger channel register, in Single/Burst mode, all the bit fields are used.

In Burst Conversion mode, only the BSTTRG bit field (bit3~bit0) is used.

AD.TRG=0x4000_B00C

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
SEQTRG7	SEQTRG6	SEQTRG5	SEQTRG4	SEQTRG3	SEQTRG2	SEQTRG1	SEQTRG0 BSTTRG
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
RW RW	RW RW	RW RW	RW RW	RW RW	RW RW	RW RW	RW RW

31 28	SEQTRG7	8th Sequence Trigger Source
27 24	SEQTRG6	7th Sequence Trigger Source
23 20	SEQTRG5	6th Sequence Trigger Source
19 16	SEQTRG4	5th Sequence Trigger Source
15 12	SEQTRG3	4th Sequence Trigger Source
11 8	SEQTRG2	3rd Sequence Trigger Source
7 4	SEQTRG1	2nd Sequence Trigger Source
3	SEQTRG0	1st Sequence Trigger Source
0	BSTTRG	Burst conversion Trigger Source

Value	Timer (TRGSEL '2'h1)	MPWM (TRGSEL '2'h2)
0	Timer 0	MP.ATR1
1	Timer 1	MP.ATR2
2	Timer 2	MP.ATR3
3	Timer 3	MP.ATR4
4		MP.ATR5
5		MP.ATR6
6	-	BOTTOM
7	-	PERIOD

AD.SCSR ADC Sequence Channel Selection Register

The ADC Burst Mode Channel Select register is a 32-bit register. For ADC single mode, it uses SEQ0CH to select the channel.

AD.SCSR=0x4000_B018

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
SEQ7CH	SEQ6CH	SEQ5CH	SEQ4CH	SEQ3CH	SEQ2CH	SEQ1CH	SEQ0CH
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
RW	RW	RW	RW	RW	RW	RW	RW

31 28	SEQ7CH	8th conversion sequence channel selection
27 24	SEQ6CH	7th conversion sequence channel selection
23 20	SEQ5CH	6th conversion sequence channel selection
19 16	SEQ4CH	5th conversion sequence channel selection
15 12	SEQ3CH	4th conversion sequence channel selection
11 8	SEQ2CH	3rd conversion sequence channel selection
7 4	SEQ1CH	2nd conversion sequence channel selection
3 0	SEQ0CH	1st conversion sequence channel selection This channel should be used for Single mode

AD.CR ADC Control Register

The ADC Control register is an 8-bit register.

AD.CR=0x4000_B020

7	6	5	4	3	2	1	0
ASTOP							ASTART
0							0
WO							RW

7	ASTOP	0	No operation
		1	ADC conversion stop (will be clear next @ADC clock) If ASTOP is set after a conversion starts, the conversion is completed and AD stops.
0	ASTART	0	No ADC conversion
		1	ADC conversion start when single mode (AD.MR.ADMOD and AD.MR.SEQCNT are 0x0. this bit will be cleared by coming @AD clock. If ASTART is set as 0 when ARST is 0 in Timer/MPWM trigger event mode, AD converts to AD.MR.SEQCNT once and AD stops. ASTART should be written to start the conversion sequence again

AD.SR ADC Status Register

The ADC Status register is an 8-bit register.

AD.SR=0x4000_B024

7	6	5	4	3	2	1	0
EOC	ABUSY	-	-	TRGIRQ	EOSIRQ	-	EOCIRQ
0	0	-	-	0	0	-	0
RO	RO	-	-	RC	RC	-	RC

7	EOC	ADC End-of-Conversion flag (Start-of-Conversion made by ADC_CLK clears this bit, not ASTART)
6	ABUSY	ADC conversion busy flag
-	-	Reserved.
-	-	Reserved.
3	TRGIRQ	ADC Trigger interrupt flag (Write “1” to clear flag) (0: no int / 1: int occurred)
2	EOSIRQ	This flag will be set at the end of a burst conversion or a sequence convrersion set (Write “1” to clear flag). *Sequence conversion set is the operation that AD converts to AD.MR.SEQCNT.
		0 None.
		1 End-of-Sequence Interrupt occurred in burst or single sequential mode
0	EOCIRQ	This flag will be set upon each conversion in a single is occurred (Write “1” to clear flag)
		0 None.
		1 End-of-Conversion Interrupt occurred

AD.IER Interrupt Enable Register

AD.IER=0x4000_B028

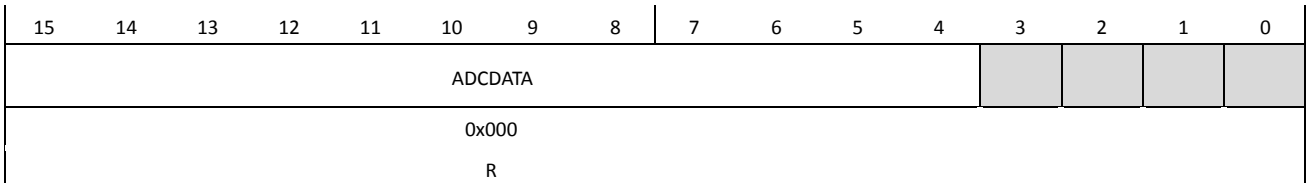
7	6	5	4	3	2	1	0
				TRGIRQE	EOSIRQE		EOCIRQE
0	0	0	0	0	0		0
				RW	RW		RW

3	TRGIRQE	ADC trigger conversion interrupt enable
2	EOSIRQE	ADC sequence conversion interrupt enable
1	-	Reserved.
0	EOCIRQE	ADC single conversion interrupt enable

AD.DRmADC Sequence Data Register 0~7

The ADC Data registers are 16-bit registers. The ADC Data registers contain the latest conversion results for each of the 8 sequence conversions.

AD.DR0=0x4000_B030, AD.DR1=0x4000_B034, AD.DR2=0x4000_B038, AD.DR3=0x4000_B03C
AD.DR4=0x4000_B040, AD.DR5=0x4000_B044, AD.DR6=0x4000_B048, AD.DR7=0x4000_B04C



15 4	ADCDATA	ADC channel 0~7 data (12-bit)

Functional Description

AD Conversion Timing Diagram

When AD.MR.ADMOD is 0x0 and AD.MR.SEQCNT is 0x0, ADC conversion is started by writing AD.CR.ASTART as ‘1’. After AD.CR.ASTART is set, Start of Conversion (SOC) is activated in 3 ADC clocks and AD.SR.EOCIRQ is set in 2 ADC clocks and 2 PCLKs after End of Conversion.

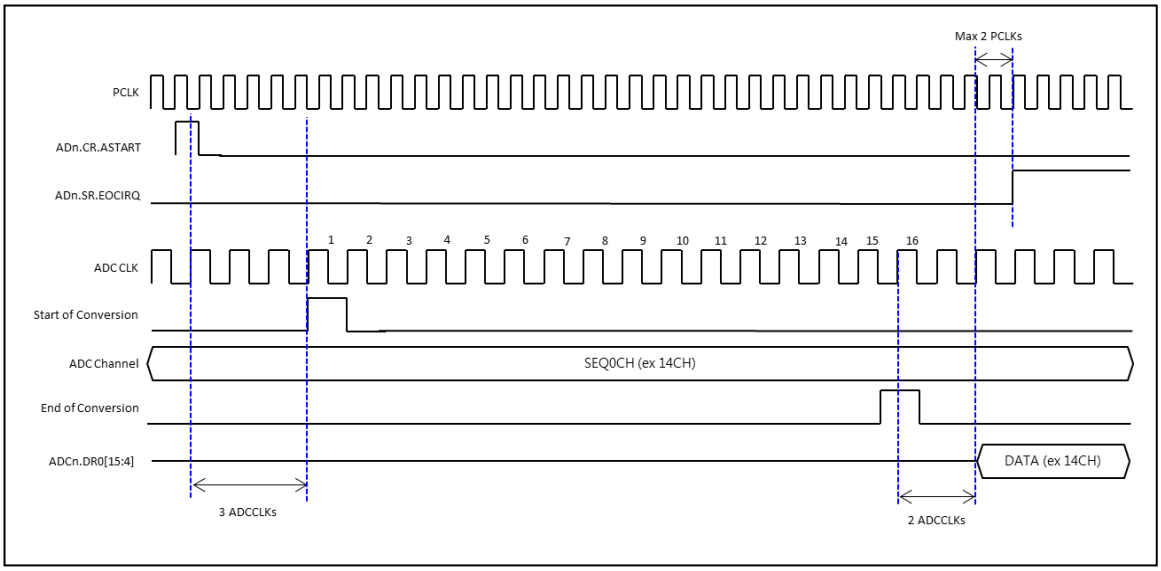


Figure 17-2 ADC Single Mode Timing (When ADCn.MR.AMOD = 0x0)

ADC Burst Conversion Mode Timing Diagram

The Burst Conversion mode (Burst mode) occurs when AD.MR.ADMOD is 0x1. When there are two sources to make SOC in Burst mode, one is the TRG event (TIMER and MPWM) and the other is AD.CR.ASTART. When AD.MR.TRGSEL is set as timer event trigger or MPWM event trigger, SOC is made by the trigger of AD.TRG.BSTTRG (AD.TRG[3:0]). For example, ADC conversion is started by the trigger of TIMER3 if AD.TRG.BSTTRG is set as TIMER3. Once the BSTTRG triggers events, ADC converts ADC channels per the values set in AD.MR.SEQCNT. See Figure 17-3.

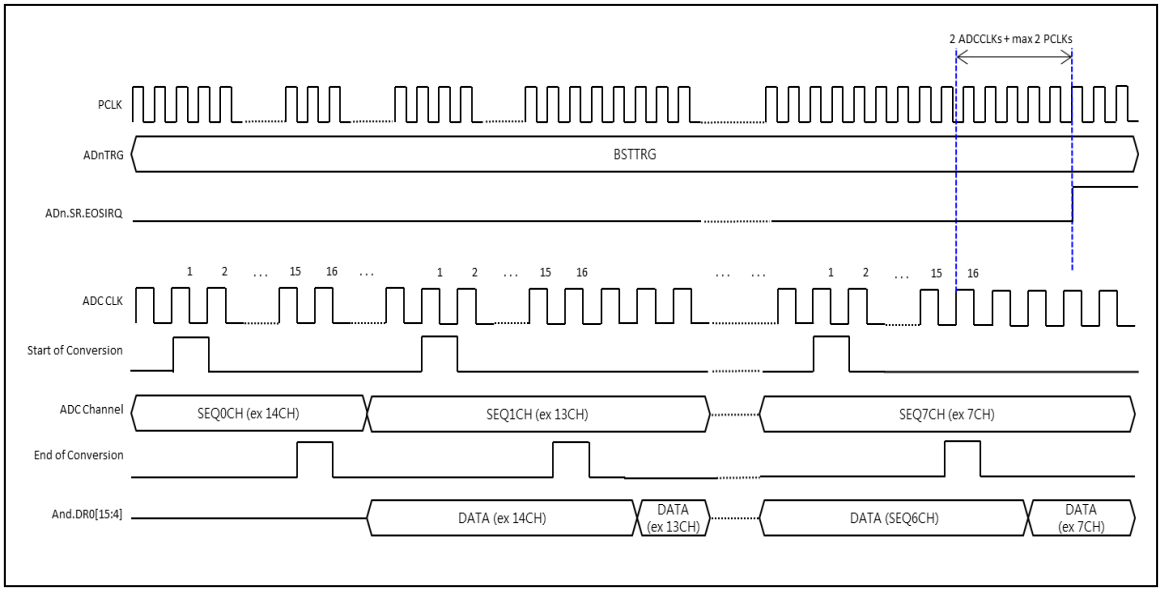


Figure 17-3 ADC Burst Mode Timing (When AD.MR.AMOD = 2'b01)

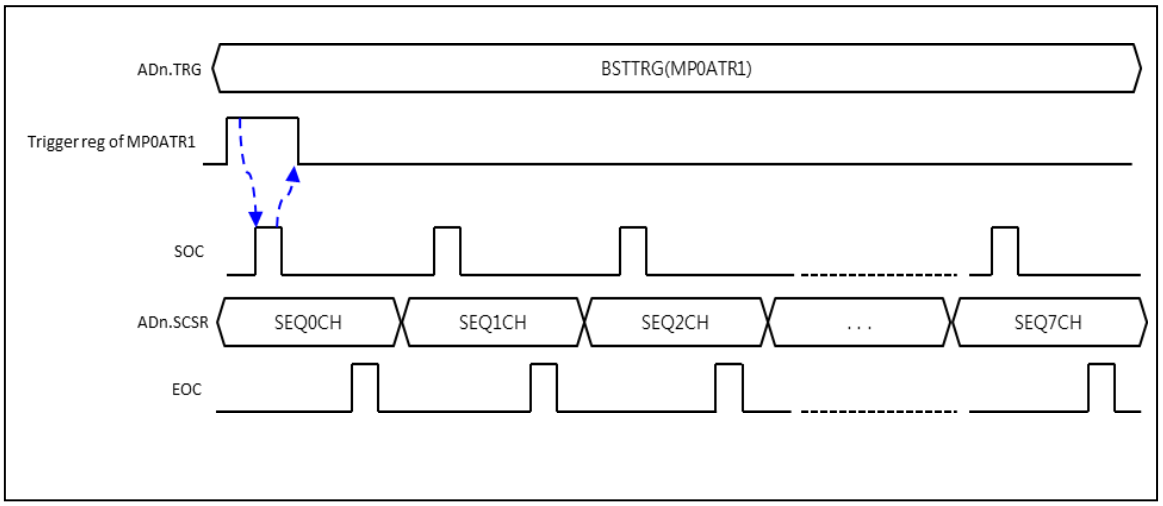


Figure 17-4 ADC Trigger Timing in Burst Mode (SEQCNT = 3'b111, 8 Sequence Coversion)

ADC Sequential Conversion Mode Timing Diagram

Single Sequential Conversion mode (Single Sequential mode) occurs when AD.MR.ADMOD is 0x0 and AD.MR.SEQCNT is not 0x0. To set Sequential Conversion mode, AD.MR.AMOD is 2'b00 and AD.MR.SEQCNT is not 2'b00.

The operation of Sequential mode is almost the same as the Burst mode. The difference is the source of SOC. Each SOC is made by the trigger of SEQTRGx as each SEQCNT. See Figure 17-5.

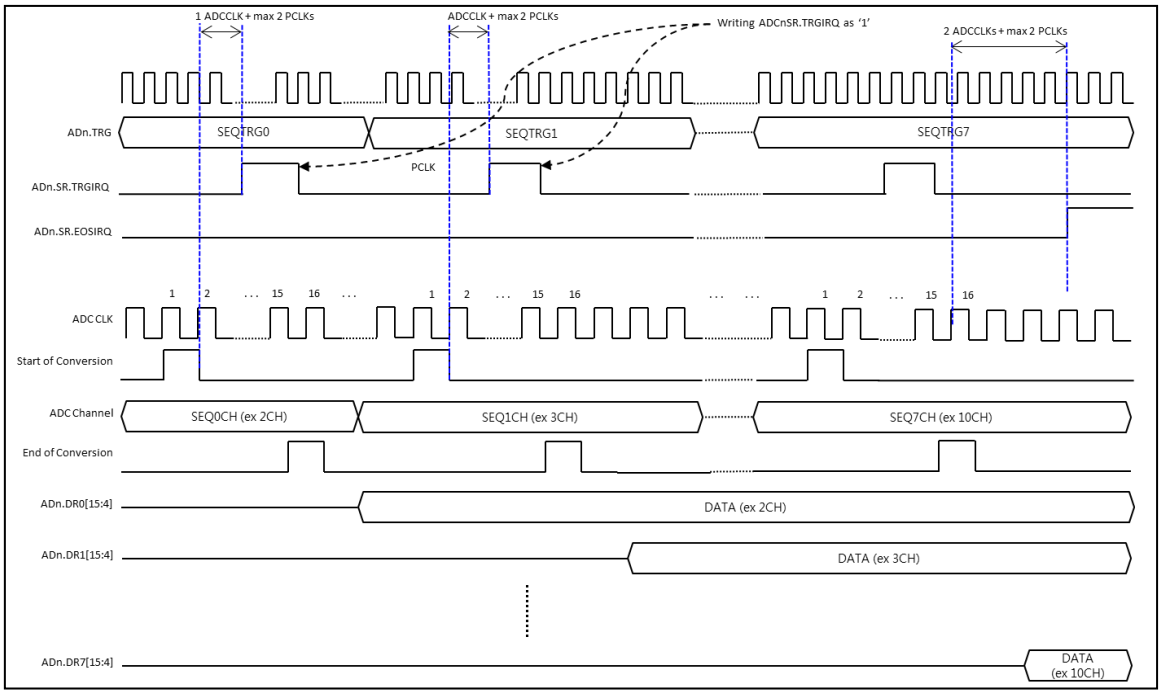


Figure 17-5 ADC Sequential Mode Timing (When AD.MR.AMOD = ó0 and AD.MR.SEQCNT Í ó0ô

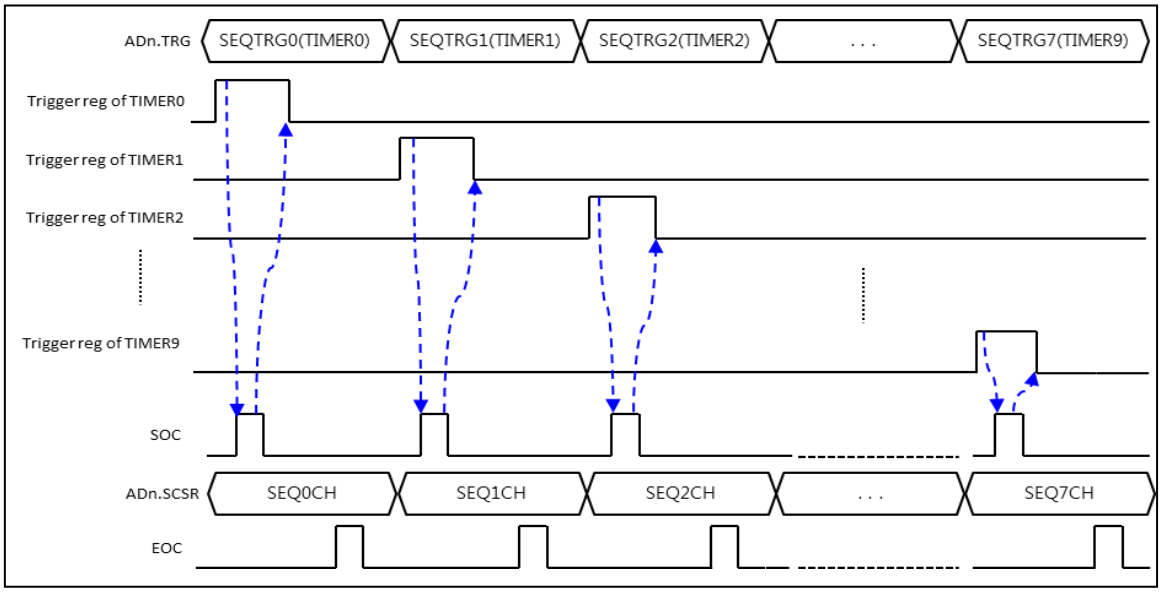


Figure 17-6 ADC Trigger Timing in Sequential Mode (SEQCNT = 3ô111, 8 Sequence Coversion)

18. Electrical Characteristics

DC Characteristics

Absolute Maximum Ratings

Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions.

Table 18-1 Absolute Maximum Rating

Parameter	Symbol	Min	Max	Unit
Power Supply (VDD)	VDD	-0.5	+6	V
Analog Power Supply (AVDD)	AVDD	-0.5	+6	V
VDC Output Voltage	VDD18			V
Input High Voltage		-	VDD+0.5	V
Input Low Voltage		VSS – 0.5	-	V
Output Low Current per pin	I_{OL}		5	mA
Output Low Current Total	$\sum I_{OL}$		40	mA
Output High Current per pin	I_{OH}		5	mA
Output Low Current Total	$\sum I_{OH}$		40	mA
Power consumption				mW
Input Main Clock Range		4	16	MHz
Operating Frequency		-	40	MHz
Storage Temperature	Tst	-55	+125	℃
Operating Temperature	Top	-40	+105	℃

DC Characteristics

Table 18-2 Recommended Operating Condition

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply Voltage	VDD		2.2	-	5.5	V
Supply Voltage	AVDD		2.2	-	5.5	V
Operating Frequency	FREQ	MOSC	4	-	16	MHz
		SOSC	-	32.768	-	kHz
		HSI	38.8	40	41.2	MHz
		LSI	32	40	48	kHz
Operating Temperature	Top	Top	-40	-	+105	℃

Table 18-3 DC Electrical Characteristics (VDD = +5V, Ta = 25ÁC)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Input Low Voltage	V_{IL}	Schmitt input	-	-	0.2VDD	V
Input High Voltage	V_{IH}	Schmitt input	0.8VDD	-	-	V
Output Low Voltage	V_{OL}	I_{OL} = 3mA	-	-	VSS+1.0	V
Output High Voltage	V_{OH}	I_{OH} = -3mA	VDD–1.0	-	-	V
Input High Leakage	I_{IH}				4	uA
Input Low Leakage	I_{IL}		-4			
Pull-up Resister	R_{PU}	VDD=5V	30	-	90	kΩ

Current Consumption

Table 18-4 describes the current consumption in Normal, Sleep, and Power Down modes under various conditions.

Table 18-4 Current Consumption in Each Mode (Temperature: +25ÁC Only)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Normal Operation	IDD_{NORMAL}	LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK= LSIOSC	–	2.6	–	mA
		LSIOSC=RUN HSIOSC=OFF MXOSC=OFF SXOSC=OFF HCLK=LSIOSC	–	0.7	–	mA
		LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK=HSIOSC	–	10.3	–	mA
		LSIOSC=OFF HSIOSC=RUN MXOSC=OFF SXOSC=OFF HCLK=HSIOSC	–	9.4	–	mA
		LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK=MXOSC	–	4.2	–	mA
		LSIOSC=OFF HSIOSC=OFF MXOSC=RUN SXOSC=OFF HCLK=MXOSC	–	3.2	–	mA
		LSIOSC=RUN HSIOSC=RUN MXOSC=RUN SXOSC=RUN HCLK=SXOSC	–	2.6	–	mA
		LSIOSC=OFF HSIOSC=OFF MXOSC=OFF SXOSC=RUN HCLK=SXOSC	–	0.7	–	mA

Sleep Mode	IDD_{SLEEP}	LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=LSIOSC	–	2.5	–	mA
		LSIOSC=RUN HSIOSC=OFF SXOSC=OFF MXOSC=OFF HCLK=LSIOSC	–	0.6	–	mA
		LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=HSIOSC		7.6		mA
		LSIOSC=OFF HSIOSC=RUN SXOSC=OFF MXOSC=OFF HCLK=HSIOSC	–	6.8	–	mA
		LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=MXOSC	–	3.5	–	mA
		LSIOSC=OFF HSIOSC=OFF SXOSC=OFF MXOSC=RUN HCLK=MXOSC	–	2.5	–	mA
		LSIOSC=RUN HSIOSC=RUN SXOSC=RUN MXOSC=RUN HCLK=SXOSC	–	2.5	–	mA
		LSIOSC=OFF HSIOSC=OFF SXOSC=RUN MXOSC=OFF HCLK=SXOSC	–	0.6	–	mA
PowerDown Mode	IDD_{STOP}	LSIOSC=STOP HSIOSC=STOP SXOSC=STOP MXOSC=STOP HCLK=STOP	–	5	10	uA

Note:

UART en, 1 port toggle @5V

LSIOSC (40KHz), HSIOSC (40MHz), MXOSC (8MHz), SXOSC (32.768KHz)

POR Electrical Characteristics

Table 18-5 POR Electrical Characteristics (Temperature: -40 ~ +105ÁC)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD18		1.6	1.8	2.0	V
Operating Current	IDD_{PoR}	Typ. <6uA If always on	-	60	-	nA
POR Set Level	VR_{PoR}	VDD rising (slow)	1.3	1.4	1.55	V
POR Reset Level	VF_{PoR}	VDD falling (slow)	1.1	1.2	1.4	V

LVD Electrical Characteristics

Table 18-6 LVD Electrical Characteristics (Temperature: -40 ~ +105ÁC)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD		1.7		5	V
Operating Current	IDDLVD	Typ. <6uA when always on	-	1	-	mA
LVD Set Level 0	VLVD0	VDD falling (slow)	1.58	1.73	2.2	V
LVD Set Level 1	VLVD1	VDD falling (slow)	2.4	2.65	3.1	V
LVD Set Level 2	VLVD2	VDD falling (slow)	3.55	3.7	4.15	V
LVD Set Level 3(1)	VLVD3	VDD falling (slow)	4.2	4.35	4.8	V

Caution: (1) This LVD Voltage level is not recommended, because it sometimes can change LVD detection level at high temperature.

VDC Electrical Characteristics

Table 18-7 VDC Electrical Characteristics (Temperature: -40 ~ +105ÁC)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD_{VDC}		2.2	-	5.5	V
Current Consumption	IDD_{NORM}	@RUN	-	100	150	uA
	IDD_{STOP}	@STOP	-	1	2	uA

External OSC Characteristics

Table 18-8 External OSC Characteristics (Temperature: -40 ~ +105ÁC)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Voltage	VDD		2.2	-	5.5	V
IDD		@4MHz/5V	-	240		uA
Frequency	$OSCF_{req}$		4	-	16	MHz
Output Voltage	OSC_{VOUT}		1.2	2.4	-	V
Load Capacitance	$LOAD_{CAP}$		5	22	35	pF

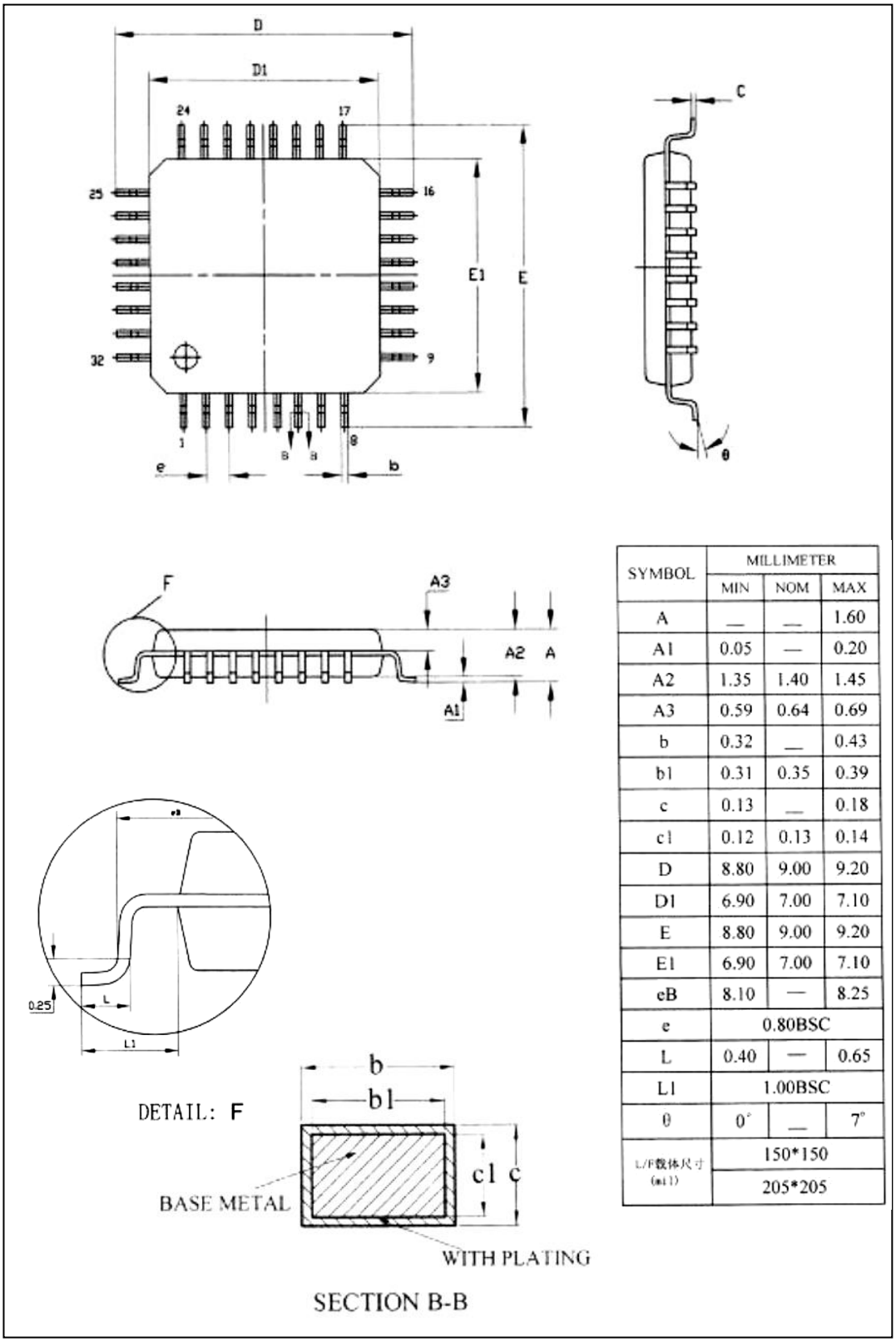
ADC Electrical Characteristics

Table 18-9. ADC Electrical Characteristics (Temperature: -40 ~ +105ÁC)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	AVDD		2.4	5	5.5	V
Resolution				12		Bit
Operating Current	IDDA				2.8	mA
Analog Input Range			0		AVDD	V
Conversion Rate				-	1.0	MSPS
Operating Frequency	ACLK				16	MHz
DC Accuracy	INL			±3.5		LSB
	DNL			±2.5		LSB
Offset Error				±1.5		LSB
Full Scale Error				±1.5		LSB
SNDR	SNDR			68		dB
THD				-70		dB

19. Package

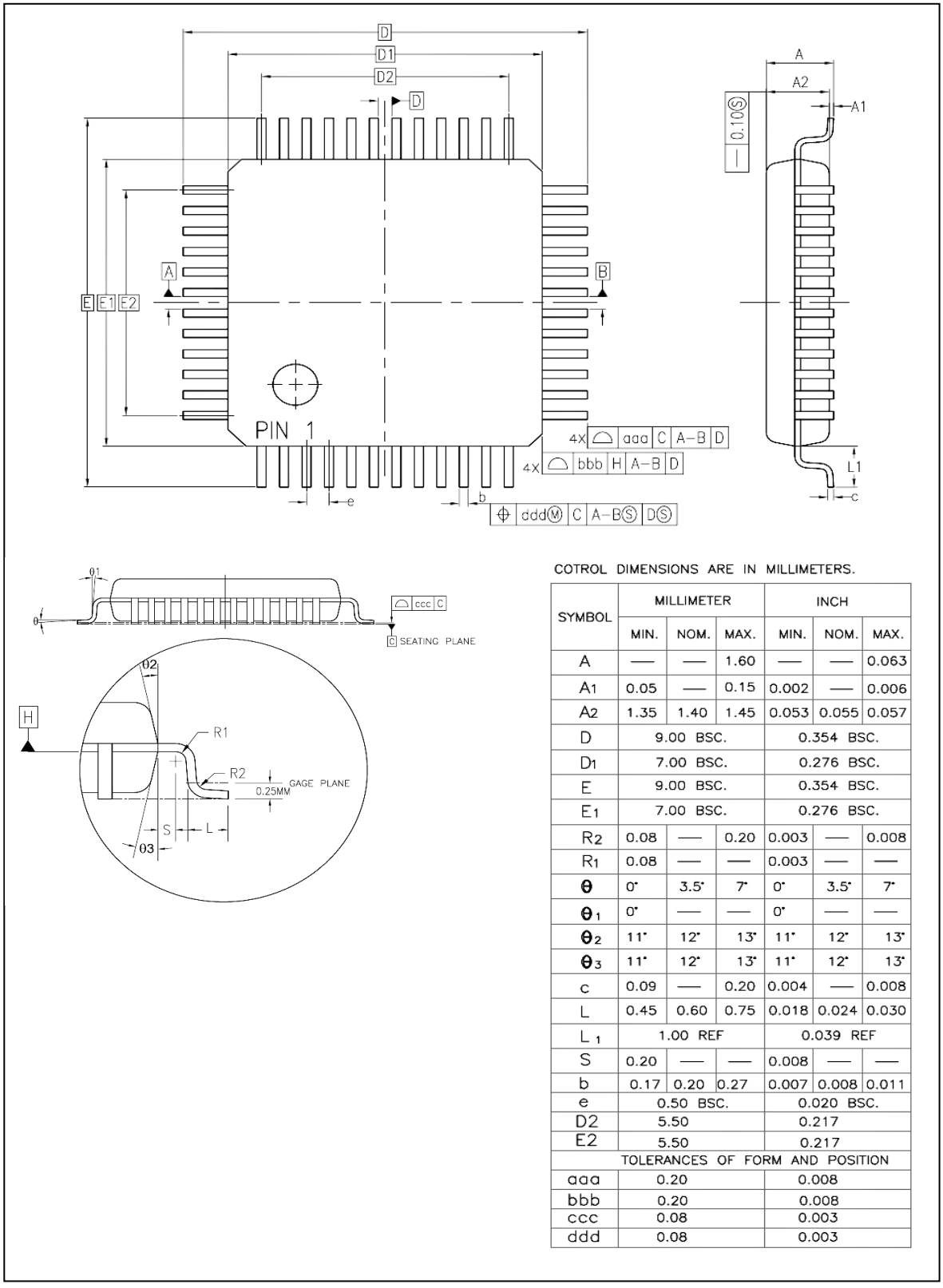
LQFP-32 Package Dimension



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.32	—	0.43
b1	0.31	0.35	0.39
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.80BSC		
L	0.40	—	0.65
L1	1.00BSC		
θ	0°	—	7°
L/F载体尺寸 (mil)	150*150		
	205*205		

Figure 19-1 Package Dimension (LQFP-32)

LQFP-48 Package Dimension



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	—	0.15	0.002	—	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D	9.00 BSC.			0.354 BSC.		
D1	7.00 BSC.			0.276 BSC.		
E	9.00 BSC.			0.354 BSC.		
E1	7.00 BSC.			0.276 BSC.		
R2	0.08	—	0.20	0.003	—	0.008
R1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	—	—	0°	—	—
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	5.50			0.217		
E2	5.50			0.217		
TOLERANCES OF FORM AND POSITION						
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

Figure 19-2 Package Dimension (LQFP-48)

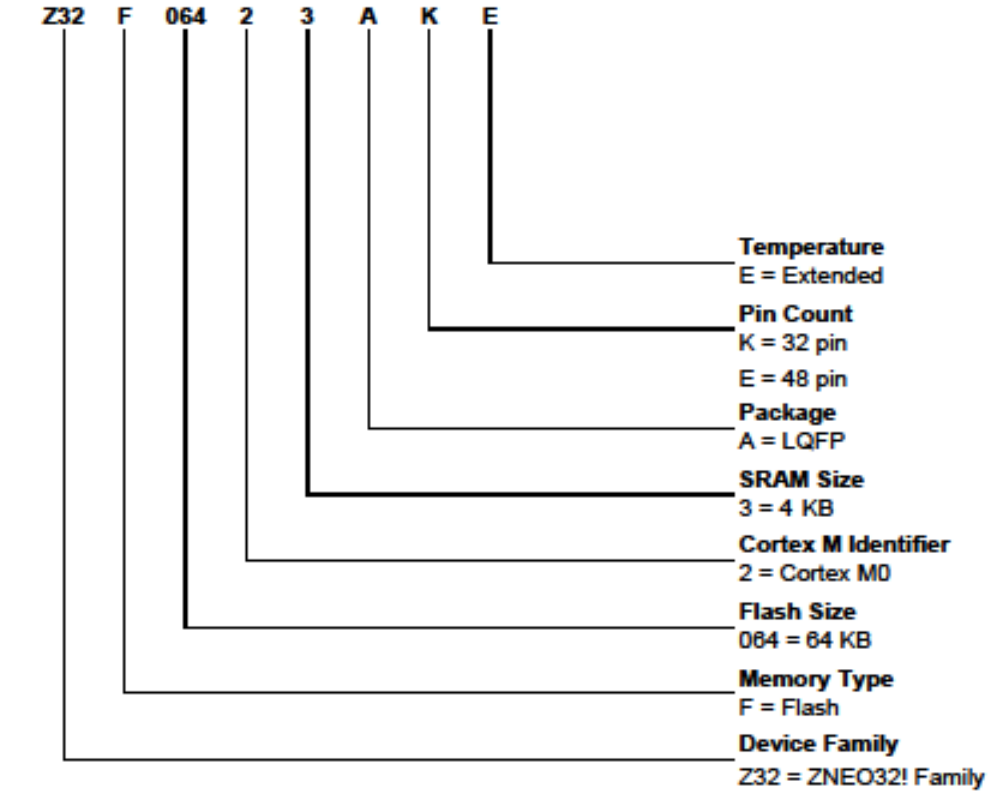
20. Ordering Information

Table 20-1 identifies the basic features and package styles available for the Z32F0642 MCU.

Table 20-1 Ordering Information

Part Number	Flash	SRAM	UART	SPI	I2C	MPWM	ADC	I/O Ports	Package
Z32F06423AKE	64KB	4KB	2	1	1	1	1-unit 10 ch	30	LQFP-32
Z32F06423AEE	64KB	4KB	2	1	1	1	1-unit 12 ch	44	LQFP-48

Zilog part numbers consist of a number of components, which are described below using part number Z32F06423AKE as an example.



Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.