zilog

Embedded in Life

An IXYS Company

ZNEO32! Cortex-M3

Z32F0641 MCU

Product Specification

PS034404-0417

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concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded.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
Apr 2017	04	Updated part numbers to include the Cortex M identifier.	All
Aor 2016	03	Added timing information for peripherals; global edits for clarity.	All
Feb 2016	02	Updated Figure 18.2 LQFP-32 Package Dimension.	178
Nov 2015	01	Original issue.	

1. Overview

Introduction

Zilog's Z32F0641 MCU, a member of the ZNEO32! Family of microcontrollers, is a cost-effective and high-performance 32-bit microcontroller. The Z32F0641 MCU provides a 3-phase PWM generator unit which is suitable for inverter bridges, including motor drive systems.

Two 12-bit high speed ADC units with 16-channel analog multiplexed inputs support feedback retrieval from the inverter bridge. Multiple powerful external serial interfaces help communicate with on-board sensors and devices.

Figure 1.1 shows a block diagram of the Z32F0641 MCU.

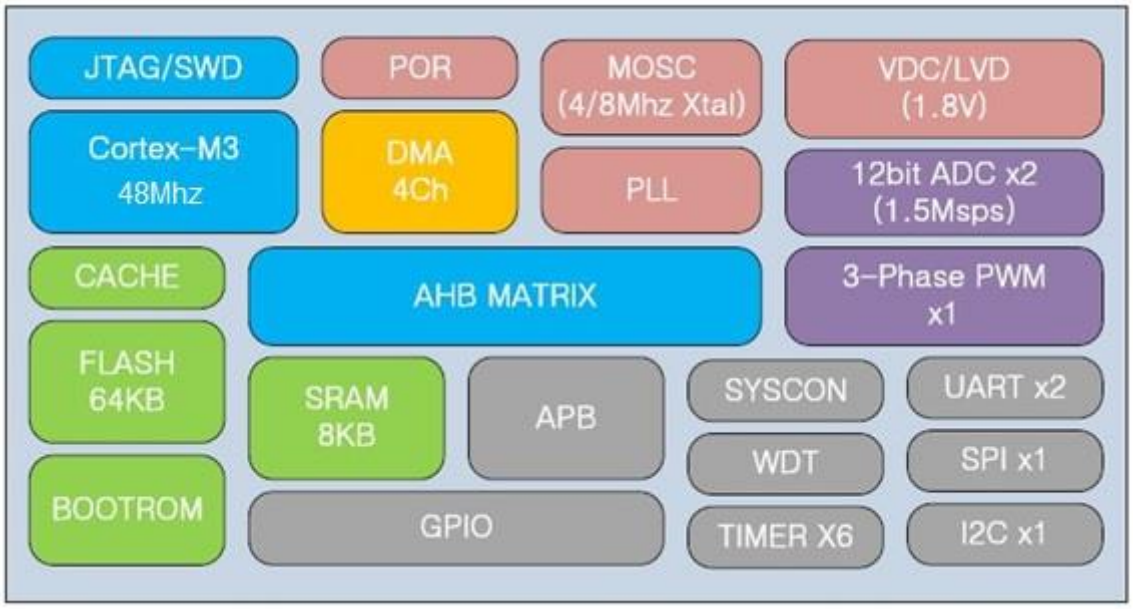


Figure 1.1 Block Diagram

Figure 1.2 and Figure 1.3 show the pin layouts.

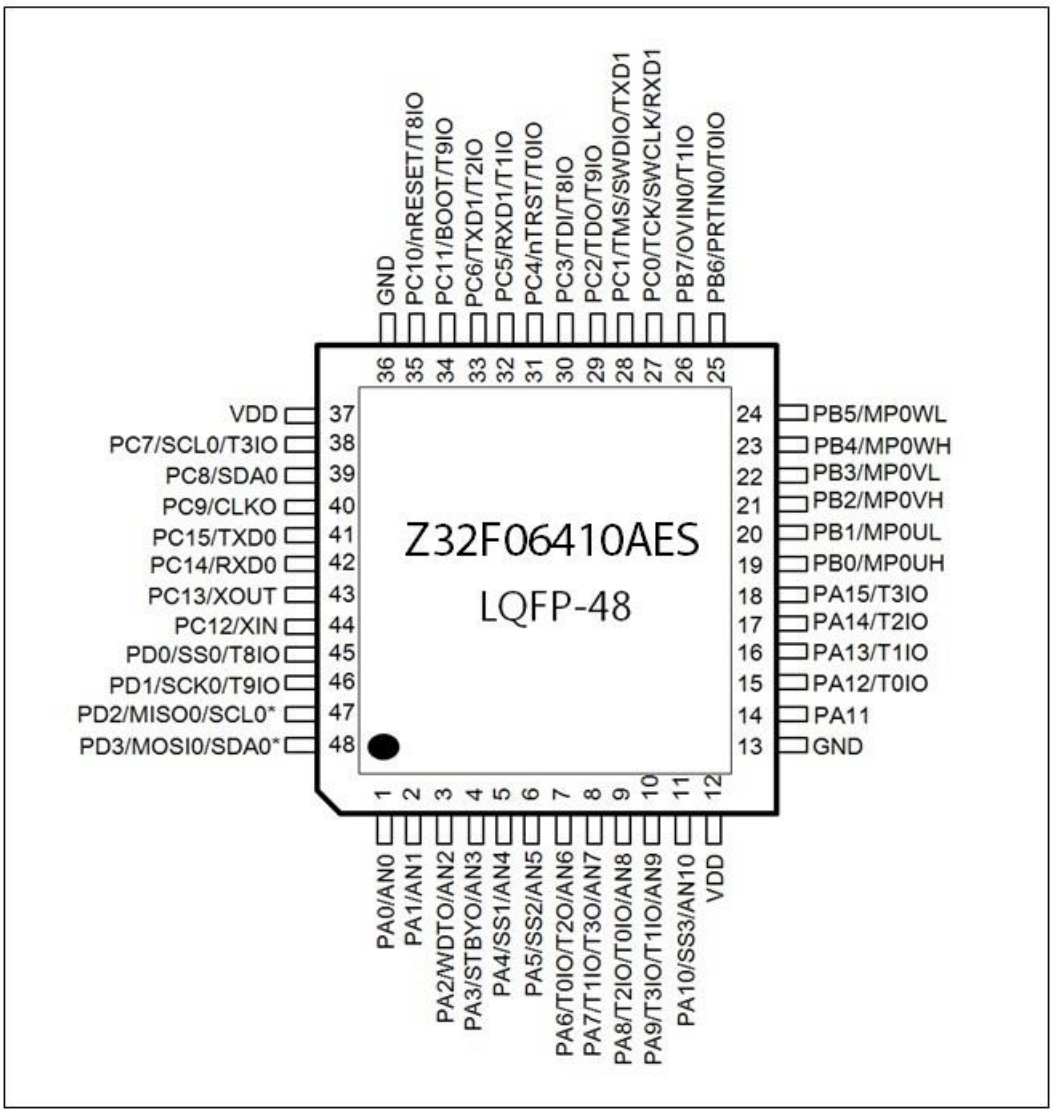


Figure 1.2 Pin Layout (LQFP-48)

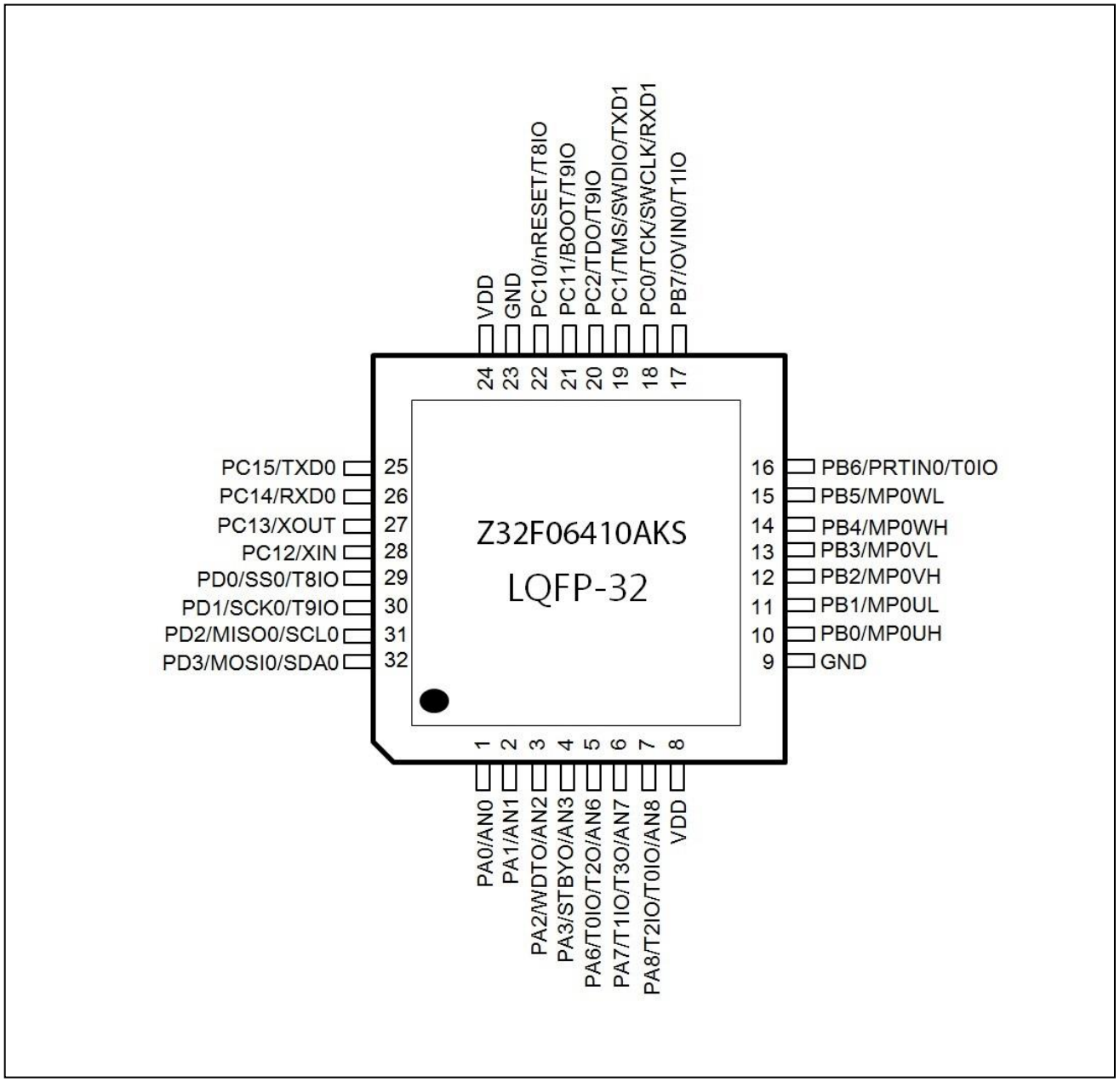


Figure 1.3 Pin Layout (LQFP-32)

Product Features

The Z32F0641 MCU offers the following features:

- High Performance low-power Cortex-M3 core
- 64 KB code Flash memory with cache function
- 8 KB SRAM
- 3-Phase PWM with ADC triggering function
- 1.5Msps high-speed ADC with sequential conversion function
  - 2 units with 11 channel Inputs
- Watchdog timer
- Six general purpose timers
  - Periodic, One-shot, PWM, Capture mode
  - Multi-timer synchronization option
- External communication ports:
  - 2 UARTs
  - 1 I^2C
  - 1 SPI
- Direct Memory Access (DMA) controller with 4 channels
- System fail-safe function by clock monitoring
- XTAL OSC fail monitoring
- Debug and emergency stop function
- Serial Wire Debug (SWD) and JTAG Debugger (JTAG is only for LQFP-48)
- Supports UART and SPI ISP
- Two types of package options
  - LQFP-48 (0.5mm pitch)
  - LQFP-32 (0.65mm pitch)
- Industrial grade operating temperature (-40 ~ +85℃)

Table 1.1 Device Type

Part Number	Flash	SRAM	UART	SPI	I2C	MPWM	ADC	I/O Ports	Package
Z32F06410AES	64KB	8KB	2	1	1	1	2-unit 11 ch	44	LQFP-48
Z32F06410AKS	64KB	8KB	2	1	1	1	2-unit 8 ch	30	LQFP-32

Architecture

Block Diagram

An internal block diagram of the Z32F0641 MCU is shown in Figure 1.4.

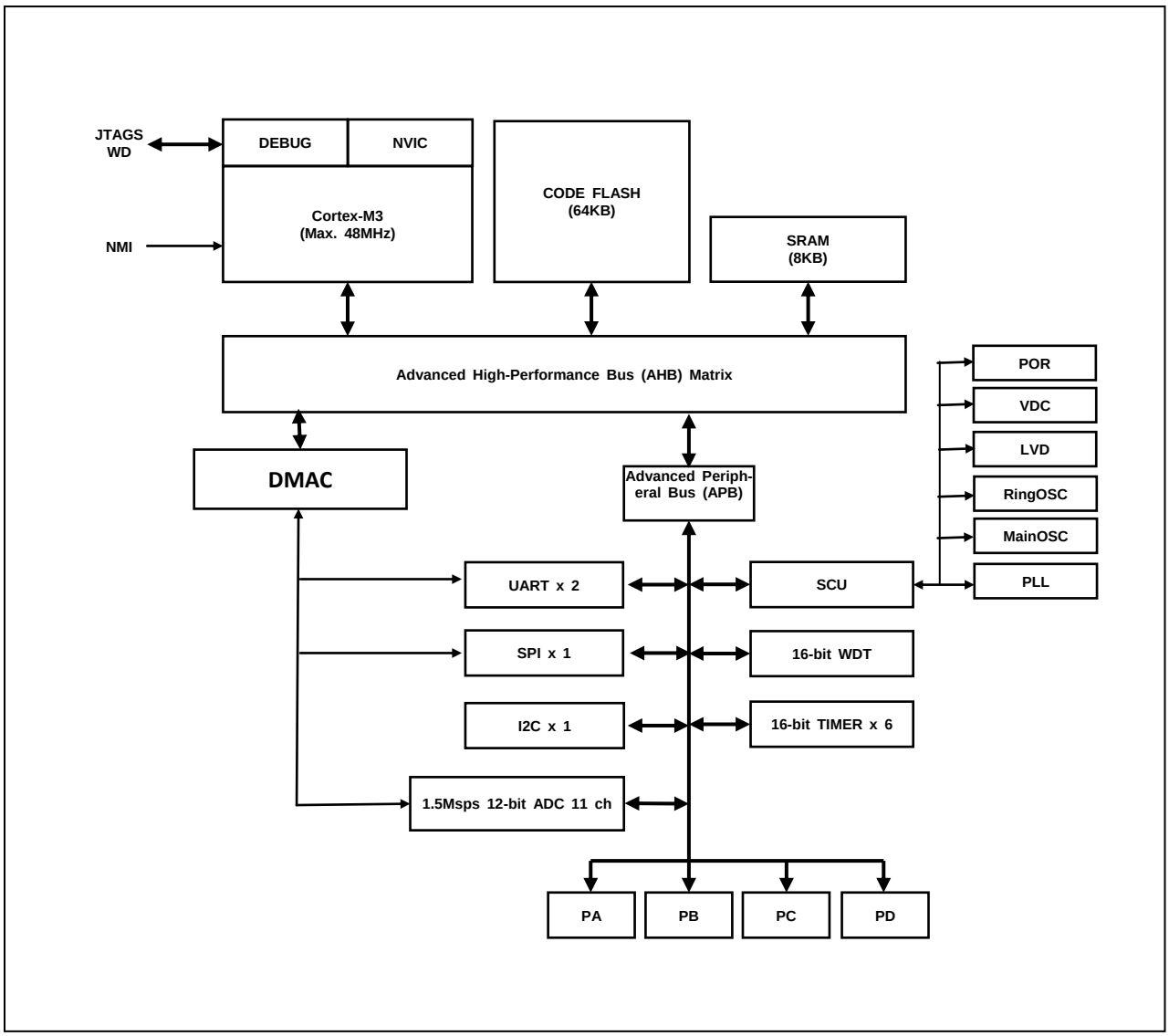


Figure 1.4 Internal Block Diagram

Functional Description

The following section provides an overview of the features of the Z32F0641 microcontroller.

ARM Cortex-M3

- ARM-powered Cortex-M3 core based on ARMv7M architecture, which is optimized for small-size and low-power systems. On core system timer (SYSTICK) provides a simple 24-bit timer that enables easy management of system operations
- 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
- JTAG and SWD debugging features
- Maximum 48 MHz operating frequency with zero wait execution

Nested Vector-Interrupt Controller (NVIC)

- The ARM Nested Vectored Interrupt Controller (NVIC) on the ARM Cortex-M3 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 interrupt service routine.
- The vector is fetched in parallel to the state saving, which enables efficient interrupt entry.
- The processor supports tail-chaining, which allows for back-to-back interrupts to be performed without the overhead of state saving and restoring

64 KB Internal Code Flash Memory

- The Z32F0641 MCU provides internal 64 KB code Flash memory and its controller, which is sufficient to program the motor algorithm and control the system. Self-programming is available and ISP and JTAG programming is also supported in boot or debugging mode.
- Instruction and data cache buffer are present and overcome the low-bandwidth Flash memory. The CPU can execute from Flash memory with zero wait state up to 48 MHz bus frequency.

8 KB 0-wait Internal SRAM

- On chip 8 KB 0-wait SRAM can be used for working memory space and program code can be loaded on this SRAM

Boot Logic

- Smart boot logic supports Flash programming. The Z32F0641 MCU can be accessed by an external boot pin; UART and SPI programming are available in Boot Mode

System Control Unit

- The System Control Unit (SCU) block manages internal power, clock, reset, and Operation Mode. The SCU also controls analog blocks (Oscillator Block, VDC and LVD)

32-bit Watchdog Timer

- The Watchdog Timer (WDT) performs the system monitoring function. The WDT generates an internal reset or interrupt if the system is in abnormal state

Multi-purpose 16-bit Timer

- Six-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 waveforms
- PWM has the ability to generate internal ADC trigger signals to measure the signal on time
- Dead time insertion and emergency stop functionality provide overcurrent protection for the chip and system

Serial Peripheral Interface (SPI)

- The Serial Peripheral Interface (SPI) block provides synchronous serial communication. The Z32F0641 MCU has 1 channel SPI module which includes the DMA function supported by a DMA controller. Transfer data is moved to/from the memory area without CPU operation
- Boot Mode uses this SPI block to download the Flash program

Inter-Integrated Circuit Interface

- The Z32F0641 MCU has 1 channel Inter-Integrated Circuit (I^2C) block which supports up to 400 kHz I^2C communication. Master and slave modes are supported

Universal Asynchronous Receiver/Transmitter

- The Z32F0641 MCU has 2 channels Universal Asynchronous Receiver/Transmitter (UART) block. For accurate baud rate control, the fractional baud rate generator is provided
- The UART features the DMA function, supported by a DMA controller. Transfer data is moved to/from the memory area without CPU operation

General PORT I/Os

- 16-bit PA, PB, PC, and PD ports are available and provide multiple 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)

- 2 built-in Analog-to-Digital Converters (ADC) can convert analog signals up to 1.5 Msps conversion rate. 11-channel analog MUX provides various combinations from external analog signals.
- The ADC features the DMA function, supported by a DMA controller. Transfer data is moved to/from the memory area without CPU operation.

Pin Description

Pin configurations are listed in Table 1.2.

Table 1.2 Pin Description

Pin Name		Pin Name	Type	Description	Remark
LQFP48	LQFP32				
1	1	PA0*	IOUS	PORT A Bit 0 Input/Output	
		AN0	IA	Analog Input 0	
2	2	PA1*	IOUS	PORT A Bit 1 Input/Output	
		AN1	IA	Analog Input1	
3	3	PA2*	IOUS	PORT A Bit 2 Input/Output	
		WDTO	O	Watchdog timer overflow output	
		AN2	IA	Comparator 2 Input	
4	4	PA3*	IOUS	PORT A Bit 3 Input/Output	
		AN3	IA	Analog Input 3	
5	-	PA4*	IOUS	PORT A Bit 4 Input/Output	
		SS1	I/O	Slave Select 1 for SPI0	
		AN4	IA	Analog Input 4	
6	-	PA5*	IOUS	PORT A Bit 5 Input/Output	
		SS2	I/O	Slave Select 2 for SPI0	
		AN5	IA	Analog Input 5	
7	5	PA6*	IOUS	PORT A Bit 6 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
		AN6	IA	Analog Input 6	
8	6	PA7*	IOUS	PORT A Bit 7 Input/Output	
		T1IO	I/O	Timer 1 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	
		AN7	IA	Analog Input 7	
9	7	PA8*	IOUS	PORT A Bit 8 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	
		AN8	IA	Analog Input 8	
10	-	PA9*	IOUS	PORT A Bit 9 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	
		T1IO	I/O	Timer 1 Input/Output	
		AN9	IA	Analog Input 9	
11	-	PA10*	IOUS	PORT A Bit 10 Input/Output	
		SS3	Output	ETM Trace Data 1	
		AN10	IA	Analog Input 10	
12	8	VDD	P	VDD	
13	9	GND	P	Ground	
14	-	PA11*	IOUS	PORT A Bit 11 Input/Output	
15	-	PA12*	IOUS	PORT A Bit 12 Input/Output	
		T0IO	I/O	Timer 0 Input/Output	

16	-	PA13*	IOUS	PORT A Bit 13 Input/Output	
		T1IO	I/O	Timer 1 Input/Output	
17	-	PA14*	IOUS	PORT A Bit 14 Input/Output	
		T2IO	I/O	Timer 2 Input/Output	
18	-	PA15*	IOUS	PORT A Bit 15 Input/Output	
		T3IO	I/O	Timer 3 Input/Output	
19	10	PB0	IOUS	PORT B Bit 0 Input/Output	
		PWM0UH	Output	PWM0 UH Output	
20	11	PB1	IOUS	PORT B Bit 1 Input/Output	
		PWM0UL	Output	PWM0 UL Output	
21	12	PB2	IOUS	PORT B Bit 0 Input/Output	
		PWM0VH	Output	PWM0 VH Output	
22	13	PB3	IOUS	PORT B Bit 1 Input/Output	
		PWM0VL	Output	PWM0 VL Output	
23	14	PB4	IOUS	PORT B Bit 4 Input/Output	
		PWM0WH	Output	PWM0 WH Output	
24	15	PB5	IOUS	PORT B Bit 5 Input/Output	
		PWM0WL	Output	PWM0 WL Output	
25	16	PB6	IOUS	PORT B Bit 6 Input/Output	
		PRTIN0	Input	PWM0 Protection Input signal 0	
		T0IO	I/O	Timer 0 Input/Output	
26	17	PB7	IOUS	PORT B Bit 7 Input/Output	
		OVIN0	Input	PWM0 Over-voltage input signal 0	
		T1IO	I/O	Timer 1 Input/Output	
27	18	PC0	IOUS	PORT C Bit 0 Input/Output	
		TCK/SWCK	Input	JTAG TCK, SWD Clock Input	
		RXD1	Input	UART0 Rx Data Input	
28	19	PC1	IOUS	PORT C Bit 1 Input/Output	
		TMS/SWDIO	I/O	JTAG TMS, SWD Data Input/Output	
		TXD1	Input	UART0 Tx Data Output	
29	20	PC2	IOUS	PORT C Bit 2 Input/Output	
		TDO/SWO	Output	JTAG TDO, SWO Output	
		T8IO	I/O	Timer 8 Input/Output	
30	-	PC3	IOUS	PORT C Bit 3 Input/Output	
		TDI	Input	JTAG TDI Input	
		T9IO	I/O	Timer 9 Input/Output	
31	-	PC4	IOUS	PORT C Bit 4 Input/Output	
		nTRST	Input	JTAG nTRST Input	
		T0IO	Input	Timer 0 input/Output	
32	-	PC5	IOUS	PORT C Bit 5 Input/Output	
		RXD1	Input	UART1 RXD Input	
		T1IO	I/O	Timer 1 input/Output	
33	-	PC6	IOUS	PORT C Bit 6 Input/Output	
		TXD1	Output	UART1 TXD Output	

		T2IO	I/O	Timer 2 input/Output	
34	21	PC11	IOUS	PORT C Bit 11 Input/Output	
		BOOT	Input	Boot mode Selection Input	
		T9IO	I/O	Timer 9 input/Output	
35	22	PC10	IOUS	PORT C Bit 10 Input/Output	
		nRESET	Input	External Reset Input	Pull-up
		T8IO	I/O	Timer 8 input/Output	
36	23	GND	P	Ground	
37	24	VDD	P	VDD	
38	-	PC7	IOUS	PORT C Bit 7 Input/Output	
		SCL0	Output	I^2C Channel 0 SCL In/Out	
		T3IO	I/O	Timer 3 input/Output	
39	-	PC8	IOUS	PORT C Bit 8Input/Output	
		SDA0	Output	I^2C Channel 0 SDA In/Out	
40	-	PC9	IOUS	PORT C Bit 9 Input/Output	
		CLKO	Output	System Clock Output	
41	25	PC15	IOUS	PORT C Bit 14 Input/Output	
		TXD0	Output	UART0 TXD Output	
		MISO0	I/O	SPI0 Master-Input/Slave-Output	
42	26	PC14	IOUS	PORT C Bit 14 Input/Output	
		RXD0	Input	UART0 RXD Input	
		MOSI0	I/O	SPI0 Master-Output/Slave-Input	
43	27	PC13	IOUS	PORT C Bit 13 Input/Output	
		XOUT	OA	External Crystal Oscillator Output	
44	28	PC12	IOUS	PORT C Bit 12 Input/Output	
		XIN	IA	External Crystal Oscillator Input	
45	29	PD0	IOUS	PORT D Bit 0 Input/Output	
		SS0	I/O	SPI1 Slave Select	
		T8IO	I/O	Timer 8 input/Output	
46	30	PD1	IOUS	PORT D Bit 1 Input/Output	
		SCK0	I/O	SPI0 Clock Input/Output	
		T9IO	I/O	Timer 9 input/Output	
47	31	PD2	IOUS	PORT D Bit 2 Input/Output	
		MISO0	I/O	SPI Channel 0 Master In / Slave Out	
		SCL0	Output	I^2C Channel 0 SCL In/Out	
48	32	PD3*	IOUS	PORT D Bit 3 Input/Output	
		MOSI0	I/O	SPI Channel 0 Master Out / Slave In	
		SDA0	Output	I^2C Channel 0 SDA In/Out	

***Notation:**

I=Input, O=Output, U=Pull-up, D=Pull-down,

S=Schmitt-Trigger Input Type, C=CMOS Input Type, A=Analog, P=Power

(*) Selected pin function after reset condition

Pin order may be changed with revision notice.

Memory Map

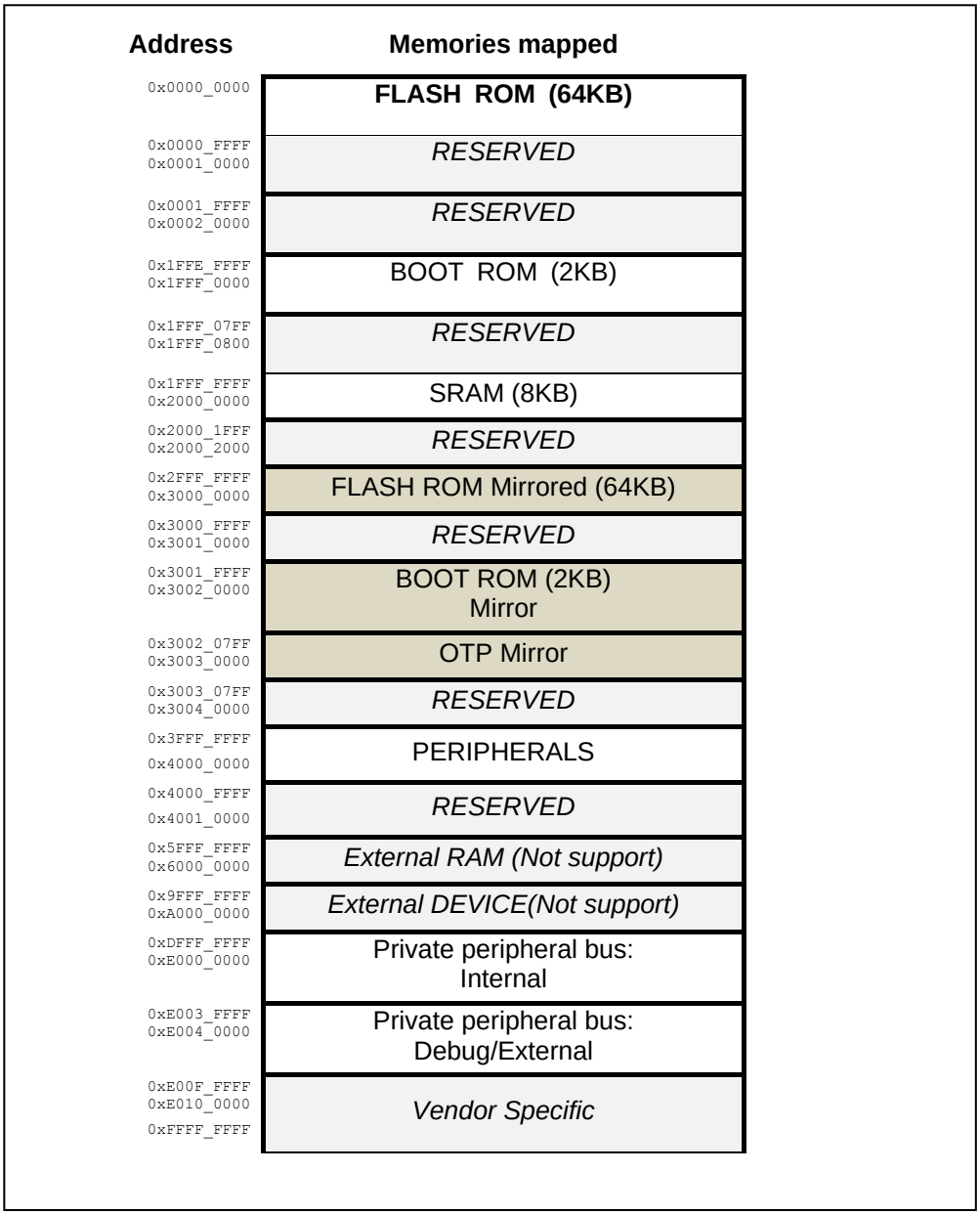


Figure 1.5 Main Memory Map

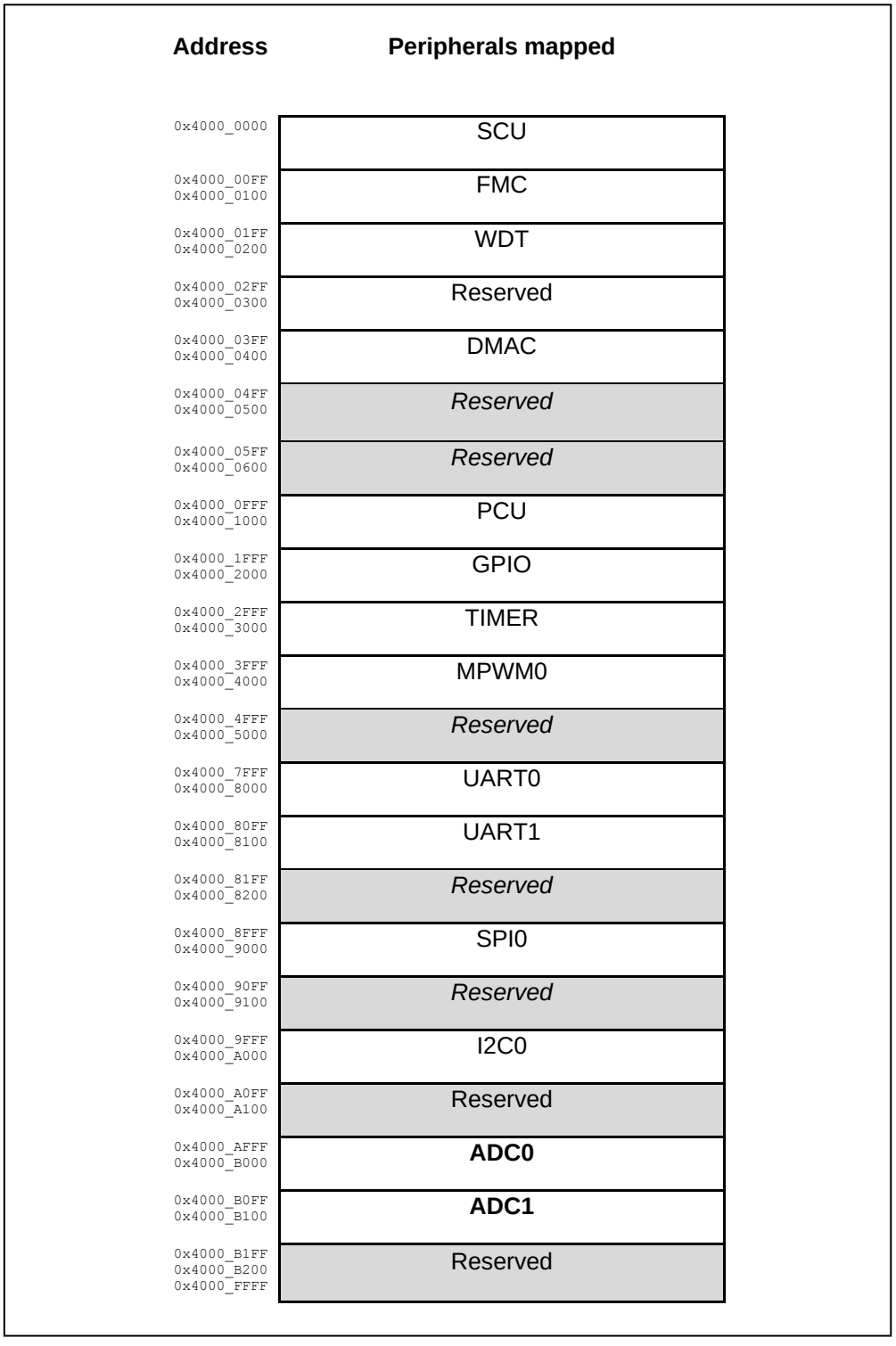


Figure 1.6 Peripheral Memory Map

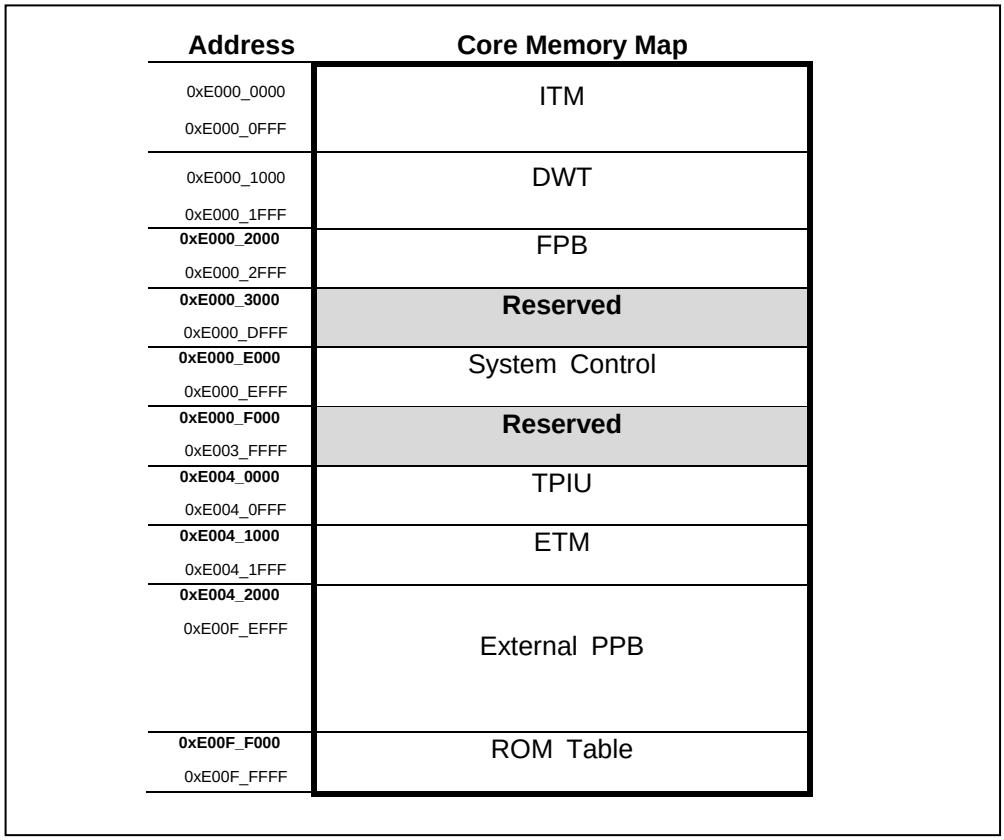


Figure 1.7 Cortex-M3 Private Memory Map

Note: Refer to document number DDI337 from ARM for more information about the memory maps.

2. CPU

Cortex-M3 Core

The CPU core is supported by the ARM Cortex-M3 processor which provides a high-performance, low-cost platform. Document number DDI337 from ARM provides more information about Cortex-M3.

Interrupt Controller

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	Reserved
-7	0x0000_0024	Reserved
-6	0x0000_0028	Reserved
-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	LVDDETECT
1	0x0000_0044	SYSCLKFAIL
2	0x0000_0048	XOSCFAIL
3	0x0000_004C	WDT
4	0x0000_0050	Reserved
5	0x0000_0054	TIMER0
6	0x0000_0058	TIMER1
7	0x0000_005C	TIMER2
8	0x0000_0060	TIMER3
9	0x0000_0064	Reserved
10	0x0000_0068	Reserved
11	0x0000_006C	Reserved
12	0x0000_0070	Reserved
13	0x0000_0074	TIMER8
14	0x0000_0078	TIMER9
15	0x0000_007C	Reserved
16	0x0000_0080	GPIOAE
17	0x0000_0084	GPIOAO
18	0x0000_0088	GPIOBE

19	0x0000_008C	GPIOBO
20	0x0000_0090	GPIOCE
21	0x0000_0094	GPIOCO
22	0x0000_0098	GPIODE
23	0x0000_009C	GPIODO
24	0x0000_00A0	MPWM0
25	0x0000_00A4	MPWM0PROT
26	0x0000_00A8	MPWM0OVV
27	0x0000_00AC	Reserved
28	0x0000_00B0	Reserved
29	0x0000_00B4	Reserved
30	0x0000_00B8	Reserved
31	0x0000_00BC	Reserved
32	0x0000_00C0	SPI0
33	0x0000_00C4	Reserved
34	0x0000_00C8	Reserved
35	0x0000_00CC	Reserved
36	0x0000_00D0	I2C0
37	0x0000_00D4	Reserved
38	0x0000_00D8	UART0
39	0x0000_00DC	UART1
40	0x0000_00E0	Reserved
41	0x0000_00E4	Reserved
42	0x0000_00E8	Reserved
43	0x0000_00EC	ADC0
44	0x0000_00F0	ADC1
45	0x0000_00F4	Reserved
46	0x0000_00F8	Reserved
47	0x0000_00FC	Reserved
48	0x0000_0100	Reserved
49	0x0000_0104	Reserved
50	0x0000_0108	Reserved
51	0x0000_010C	Reserved
52	0x0000_0110	Reserved
53	0x0000_0114	Reserved
54	0x0000_0118	Reserved
55	0x0000_011C	Reserved
56	0x0000_0120	Reserved
57	0x0000_0124	Reserved
58	0x0000_0128	Reserved
59	0x0000_012C	Reserved
60	0x0000_0130	Reserved
61	0x0000_0134	Reserved
62	0x0000_0138	Reserved
63	0x0000_013C	Reserved

3. Boot Mode

Boot Mode Pins

The Z32F0641 MCU includes a Boot Mode option to program internal Flash memory.To enter Boot Mode, set the BOOT pin to **Low** at reset timing.

Note: The Normal state of the BOOT pin is **High**.

Boot Mode supports UART boot and SPI boot.UART boot uses the UART0 port, and SPI boot uses SPI0.The pins used 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
SPI0	**SS0/PA12**	I	SPI Boot Slave Select
	SCK0/PA13	I	SPI Boot Clock Input
	MOSI0/PA14	I	SPI Boot Data Input
	MISO0/PA15	O	SPI Boot Data Output

Boot Mode Connections

Design the target board using either of the Boot Mode ports – UART or SPI. Figure 3.1 and Figure 3.2 display sample Boot Mode connections.

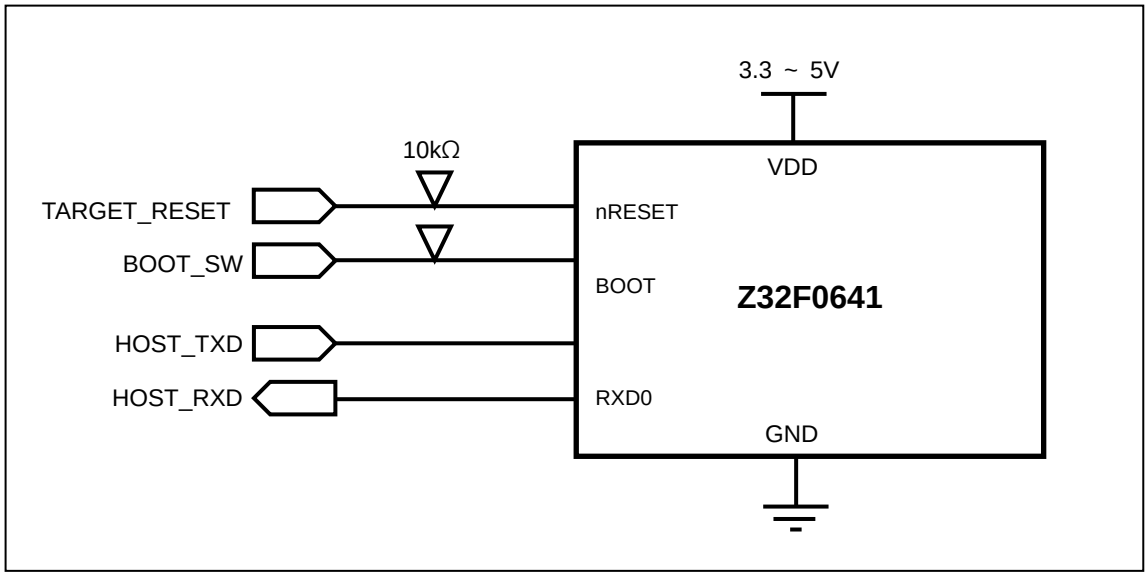


Figure 3.1 UART Boot Connection Diagram

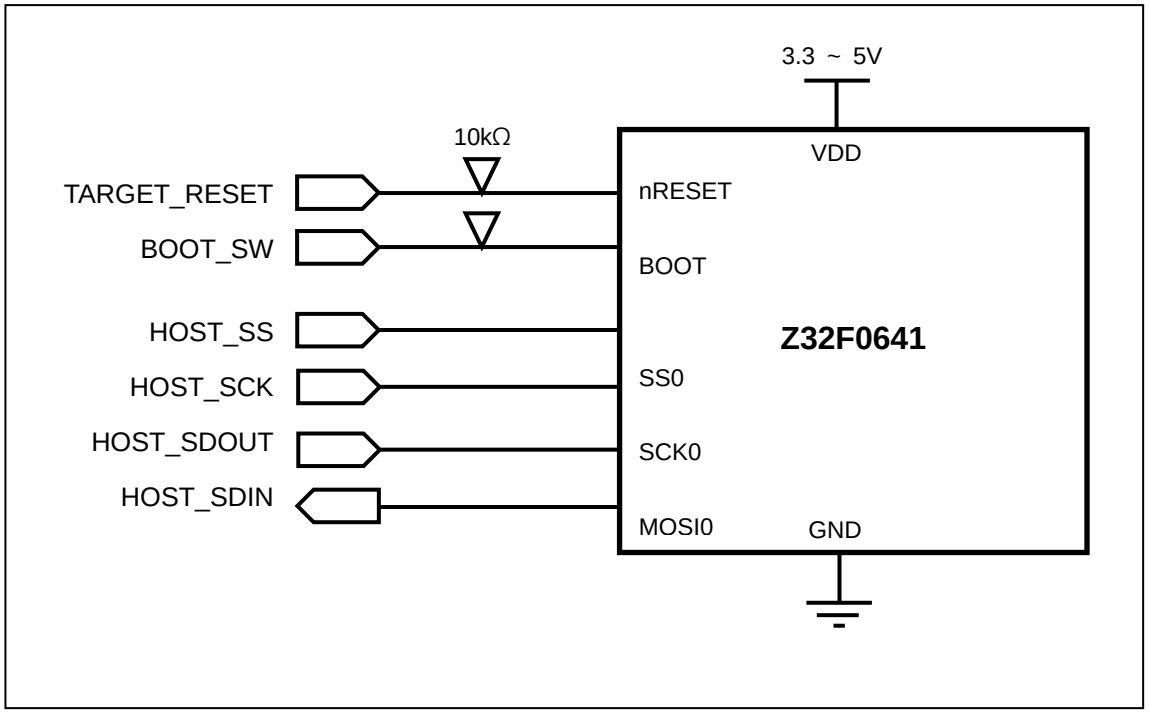


Figure 3.2 SPI Boot Connection Diagram

4. System Control Unit

Overview

The Z32F0641 microcontroller has an in-built intelligent power control block which manages the system analog blocks and operating modes. Internal reset and clock signals are controlled by the SCU block to maintain optimal system performance and power dissipation.

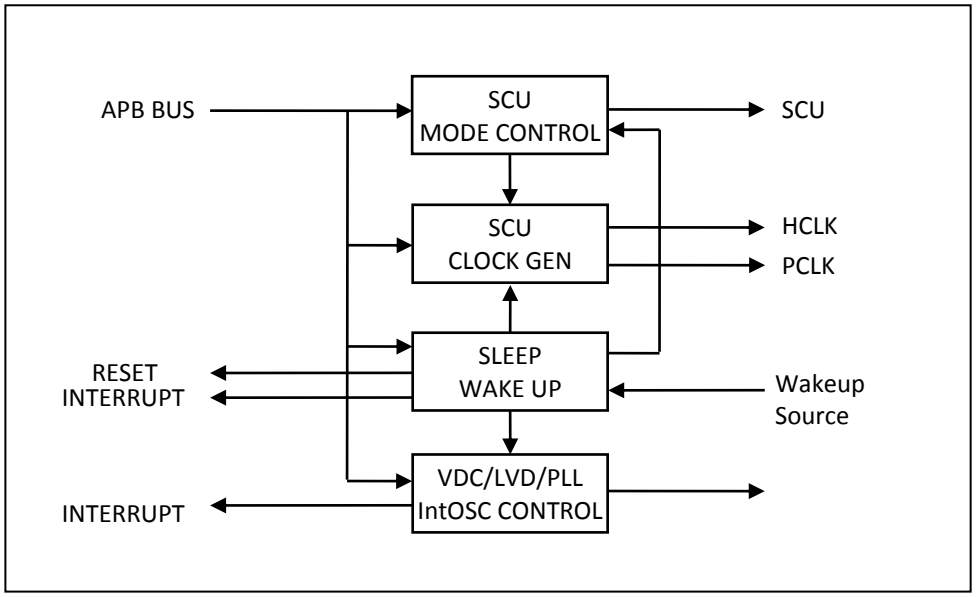


Figure 4.1 SCU Block Diagram

Clock System

The Z32F0641 MCU has the following two main operating clocks:

HCLK – Clock for the CPU and AHB bus system
PCLK – Clock for peripheral systems

Figure 4.2 and Figure 4.3 show the chip's clock system. Table 4.1 lists the clock source descriptions.

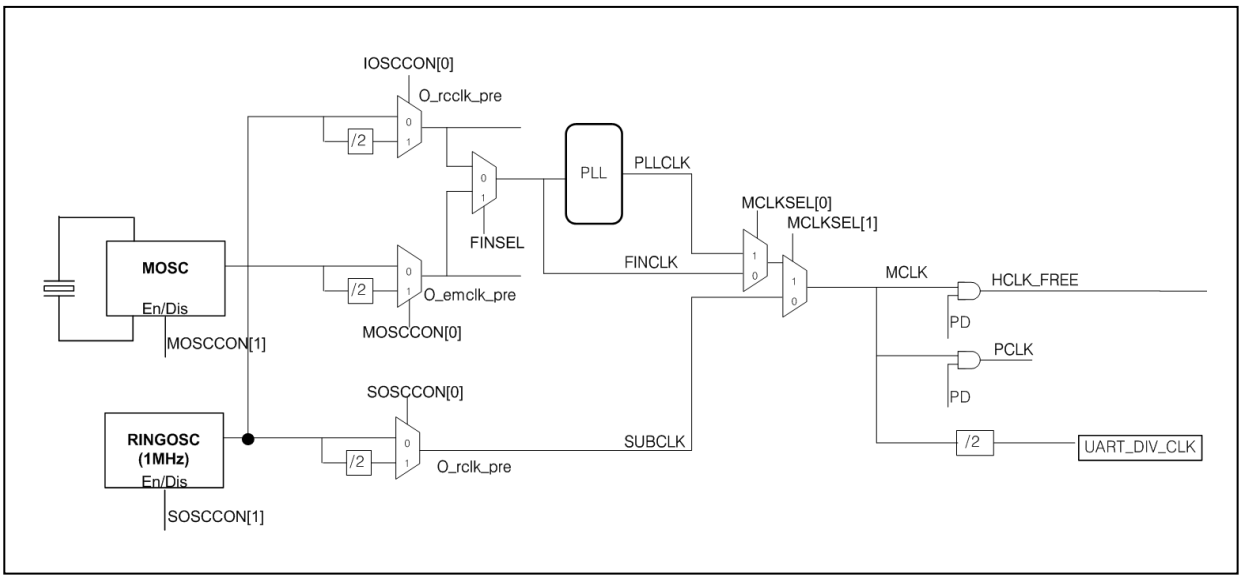


Figure 4.2 Clock Source Configuration

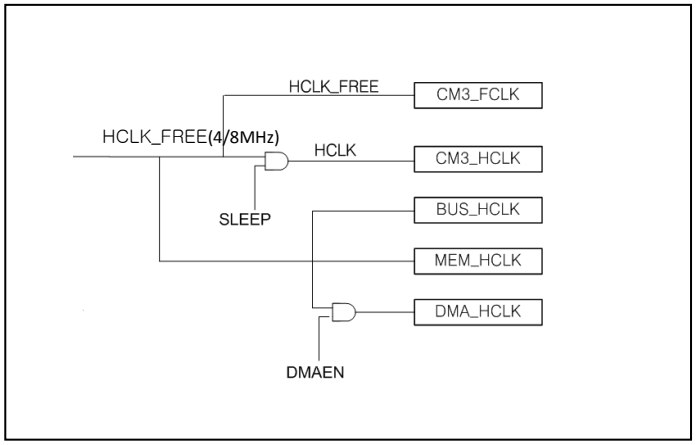


Figure 4.3 System Clock Configuration

Each of the multiplexers for switching the clock source contains a circuit which allows glitch-free switching between clock modes.

Table 4.1 Clock Sources

Clock name	Frequency	Description
MainOSC	XTAL(4MHz~8MHz)	External Crystal IOSC
PLL Clock	8MHz ~ 80MHz	On Chip PLL
ROSC	1MHz	Internal RING OSC

The PLL can synthesize the PLLCLK clock up to 80 MHz with the FIN reference clock. It also has an internal pre-divider and post-divider.

HCLK Clock Domain

The HCLK clock feeds the clock to the CPU and AHB bus. The Cortex-M3 CPU requires two clocks related with the HCLK clock:

FCLK – FCLK is a free-running clock which runs continuously except during Power-Down Mode
HCLK – HCLK can be stopped during Idle Mode

Miscellaneous Clock Domain for Cortex-M3

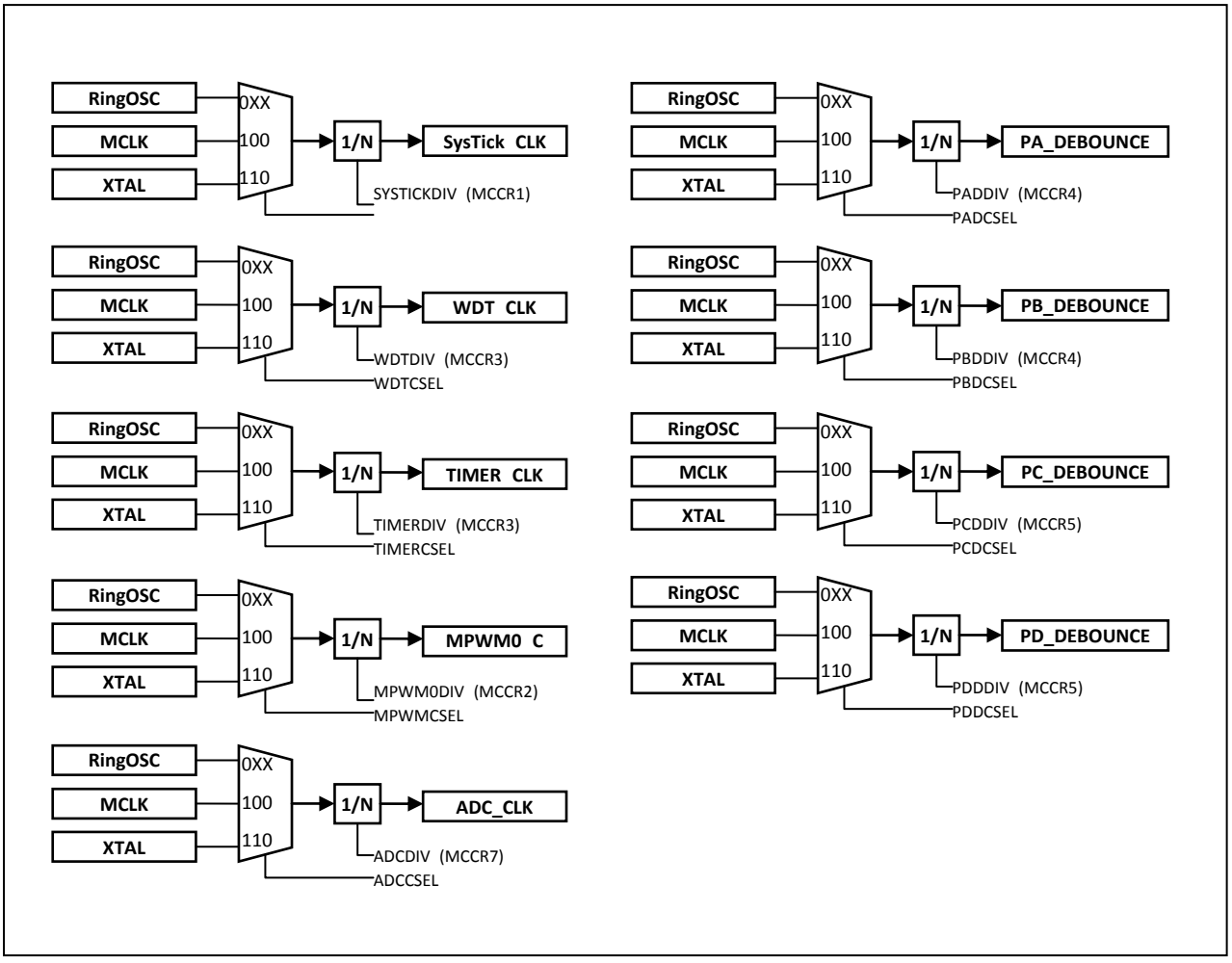


Figure 4.4 Miscellaneous Clock Configuration

PCLK Clock Domain

PCLK is the master clock of all the peripherals. It can be stopped in Power-Down Mode. Each peripheral clock is generated by the PCER register set.

Clock Configuration

After power up, the default system clock is fed by the RINGOSC (1 MHz) clock. RINGOSC is enabled by default at power up. The other clock sources are enabled by user controls with the RINGOSC system clock.

The MOSC clock can be enabled by the 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 - PCCMR and PCCCR registers should be correctly configured. After enabling the MOSC block, you must wait for more than 1 msec to ensure stable operation of crystal oscillation.

The PLL clock can be enabled by the PLLCON register. After enabling the PLL block, you must wait for the PLL lock flag. When the PLL output clock is stable; you can select MCLK for your system requirement. Before changing the system clock, Flash access wait should be set to the maximum value. After the system clock is changed, you will need to set the desired Flash access wait time.

An example flow chart outlining the steps to configure the system clock is shown in Figure 4.5.

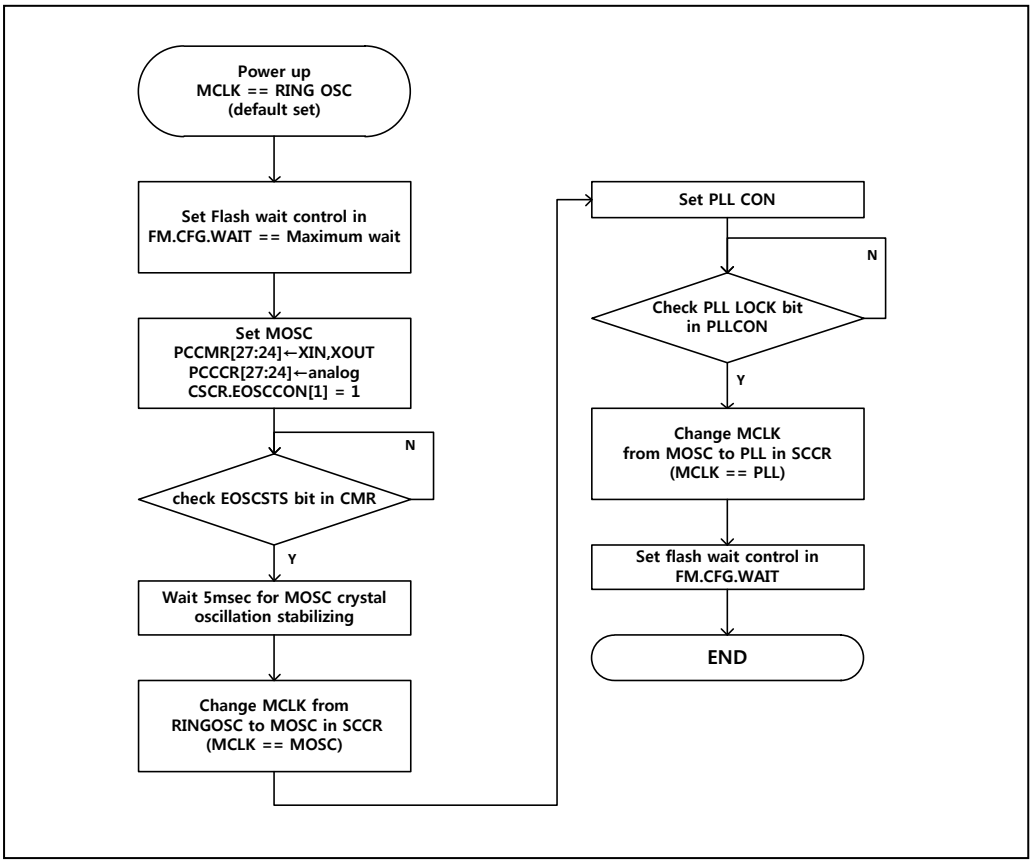


Figure 4.5. Clock Configuration Flow Chart

When you speed up the system clock up to maximum operating frequency, you should check the Flash wait control configuration. Flash read access time is one of the limiting factors in 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
000	0 clock wait	~16MHz
001	1 clock wait	~32MHz
010	2 clock wait	~48MHz
011	3 clock wait	~48MHz

Reset

The Z32F0641 MCU has two system resets:

- Cold reset by POR, which is effective during power up or down sequence, and
- Warm reset, which is generated by several reset sources. The reset event causes the chip to return to initial state.

The cold reset has only one reset source, 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

Cold Reset

Cold reset is an important feature of the chip when power is up. This characteristic globally affects the system boot. Internal VDC is enabled when VDD power is turned on. The internal VDD level slope is followed by the external VDD power slope. The internal PoR trigger level is 1.4 V of internal VDC voltage out level. At this time, boot operation is started. The RINGOSC clock is enabled and counts to 4 msec for internal VDC level stabilizing. During this time, the external VDD voltage level should be greater than the initial LVD level (2.3 V). After counting 4 msec, the CPU reset is released and the operation is started.

Figure 4.6 shows the power up sequence and internal reset waveform.

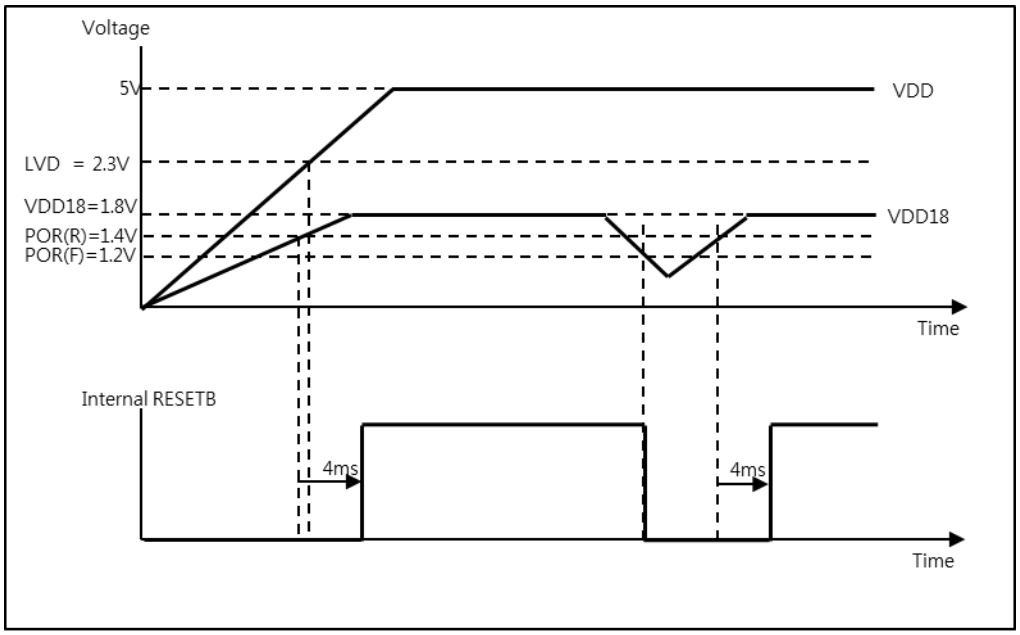


Figure 4.6. Power-up POR Sequence

The RSSR register shows the POR reset status. The last reset comes from POR; RSSR.PORST is set to “1”. After power up, this bit is always “1”. If an abnormal internal voltage drop occurs during normal operation, the system will be reset and this bit is also set to “1”.

When cold reset is applied, the chip returns to its initial state.

Warm Reset

The warm reset event has several reset sources. Some parts of the chip return to initial state when a warm reset condition occurs.
The warm reset source is controlled by the RSER register and the status appears in the RSSR register. The reset for each peripheral block is controlled by the PRER register. The reset can be masked independently.

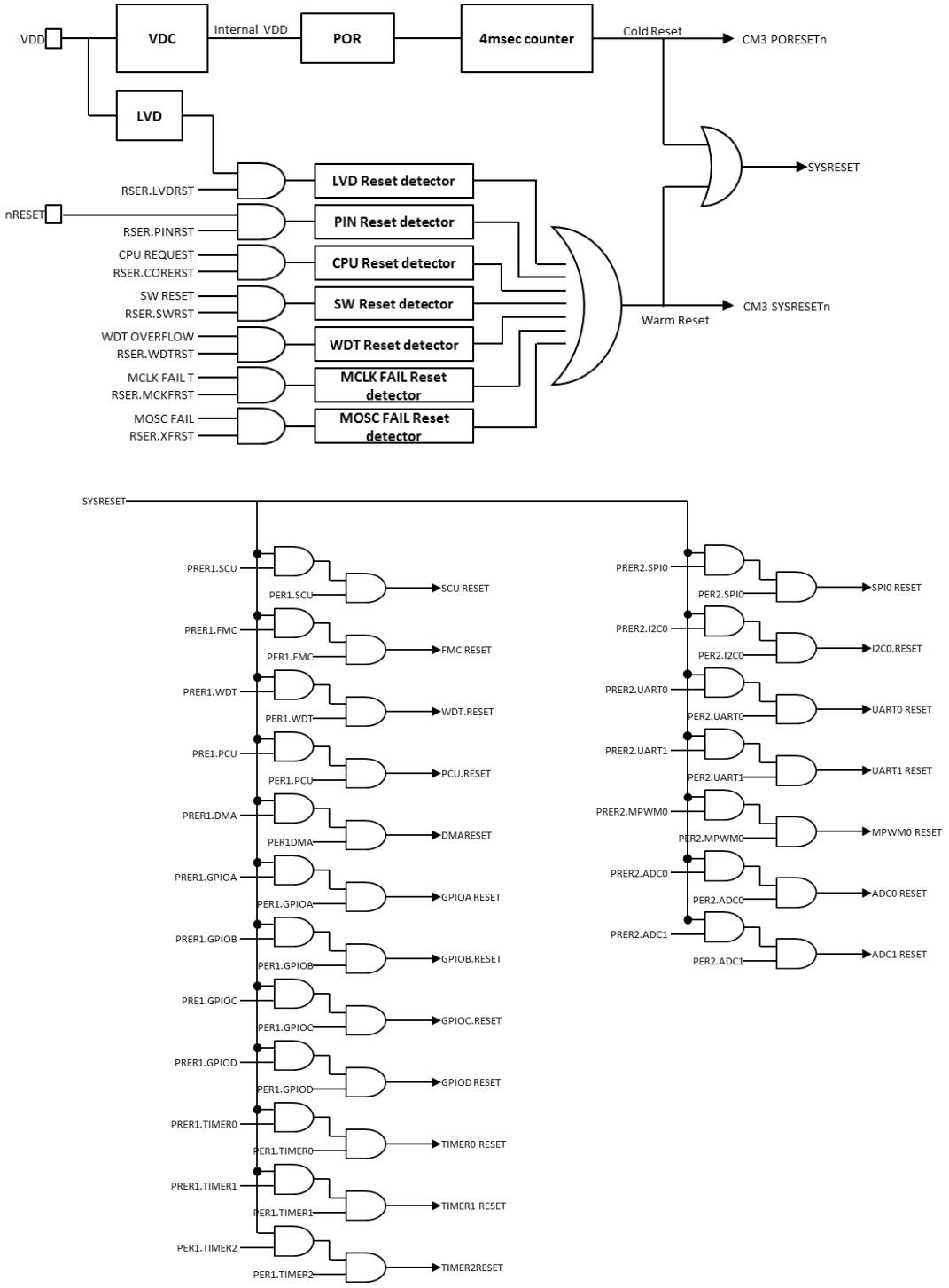


Figure 4.7. Reset Configuration

Operation Mode

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

Figure 4.8 shows the operating mode transition diagram.

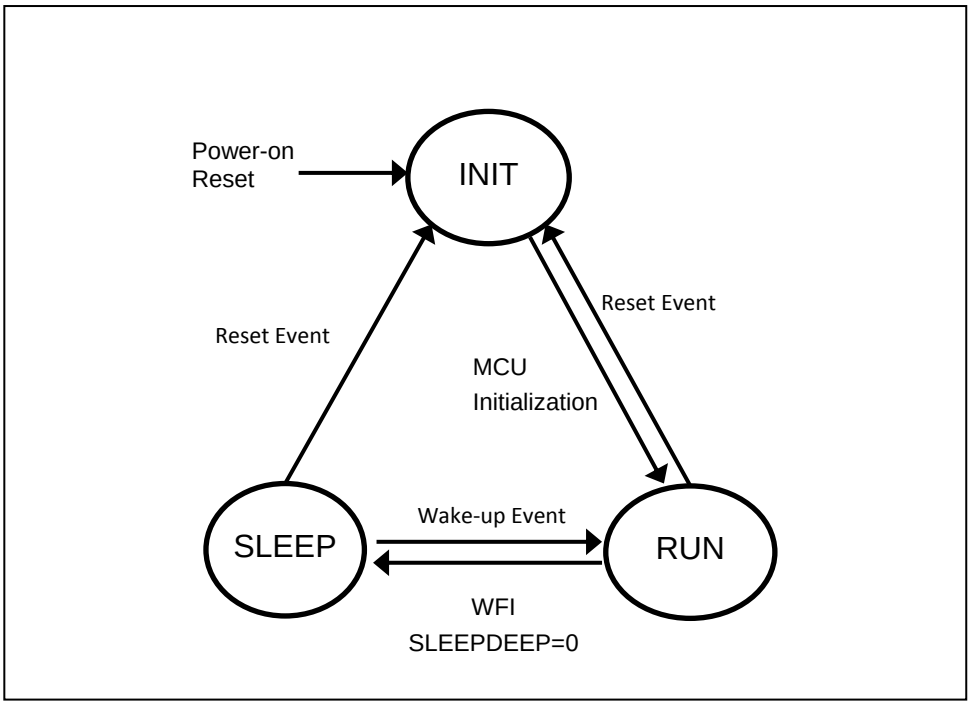


Figure 4.8. Operating Mode

RUN Mode

In RUN mode, the CPU core and the peripheral hardware is operated by using the high-speed clock. After reset, followed by the INIT state, the chip enters RUN mode.

SLEEP Mode

In SLEEP mode, only the CPU is stopped. Each peripheral function can be enabled by the function enable and clock enable bit in the PER and PCER register.

Pin Description

Table 4.3 SCU and PLL Pins

Pin Name	Type	Description
nRESET	I	External Reset Input
XIN/XOUT	OSC	External Crystal Oscillator
STBYO	O	Stand-by Output Signal
CLKO	O	Clock Output Monitoring Signal

Registers

The base address of the system control unit is `0x4000_0000` and the register map is described in Table 4.4.

Table 4.4 SCU Register Map

Name	Offset	Type	Description	Reset Value
CIDR	0x0000	R	CHIP ID Register	AC33_4064
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	RW	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	030F_0F3F*
PRER2	0x0024	RW	Peripheral reset enable register 2	0031_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
PLLCON	0x0060	RW	PLL Control register	0000_0000
VDCCON	0x0064	RW	VDC Control register	0000_000F
LVDCON	0x0068	RW	LVD Control register	0000_0001
EOSCR	0x0080	RW	External Oscillator control register	0000_0300
EMODR	0x0084	RW	External mode pin read register	0000_000X
DBCLK1	0x009C	RW	Debounce Clock Control register 1	0001_0001
DBCLK2	0x00A0	RW	Debounce Clock Control register 2	0001_0001
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

CIDR Chip ID Register

The CHIP ID Register shows chip identification information. This register is a 32-bit read-only register.

CIDR=0x4000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHIPID																															
0xAC33_4064																															
RO																															

31	CHIPID	Device ID
0		0xAC33_4064

CIDR=0x4000_000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHIPID2																															
0x0000_0000																															
RO																															

31	CHIPID2	Revision ID
0		0x0000_0000

SMR System Mode Register

The current operating mode is shown in this SCU mode register.The operating mode can be changed by writing a new mode in this register. The previous operating mode will be saved in this register after a reset event. 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
										PREVMODE					
0	0	0	0	0	0	0	0	0	0	00		0	0	0	0
										R					

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 PowerDown mode
		11 Previous operating mode was INIT mode

SRCR System Reset Control Register

The System Reset Control Register allows the software to initate a reset. This register also provides the polarity for the STBYOP pin.

SCR=0x4000_0008

7	6	5	4	3	2	1	0
							SWRST
0	0	0	0	0	0	0	0
							W

1	SWRST	Internal soft reset activation bit
		0 Normal operation
		1 Internal soft reset is applied and auto cleared

WUER Wakeup Source Enable Register

Enable the wakeup source when the chip is in Power-Down Mode. The source of chip wakeup should be enabled in each bit field for wakeup sources that will be used. 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**. The Wakeup Source Enable 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							WDTWUE	LVDWUE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				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
1	WDTWUE	Enable wakeup source of watchdog timer 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 any wakeup source, the wakeup source is identified by reading the Wakeup Source Status Register. When the bit is set to **1**, the related wakeup source issues the wake-up signal to the SCU. The bit is cleared when the event source is cleared by the software. These bits show the interrupt flag in each peripheral.

Examples:
A GPIO wakeup status is cleared by clearing the interrupt flag in the PCn.ISR register of the PCU block. A WDTWU interrupt is cleared by clearing the overflow interrupt flag in the WDT block. The LVD flag is cleared when the low voltage condition is resolved.

WUSR=0x4000_0014

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

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
1	WDTWU	Status of wakeup source of watchdog timer 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 is selected by the Reset Source Enable Register. When **1** is written in the bit field of each reset source, the reset source event is transferred to the reset generator. When **0** is written in the bit field of each reset source, the reset source event is masked and does not generate a reset event.

RSER=0x4000_0018

7	6	5	4	3	2	1	0
	PINRST	CORERST	SWRST	WDTRST	MCKFRST	XFRST	LVDRST
0	1	0	0	1	0	0	1
	RW	RW	RW	RW	RW	RW	RW

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	XFRST	External OSC 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 displays 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 a reset source is found, write **1** to the corresponding bit to clear the reset status. This register is an 8-bit register.

RSSR=0x4000_001C

7	6	5	4	3	2	1	0
PORST	**PINRST**	**CPURST**	**SWRST**	**WDTRST**	**MCKFRST**	**XFRST**	**LVDRST**
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
RC1	**RC1**	**RC1**	**RC1**	**RC1**	**RC1**	**RC1**	**RC1**

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

PRER1 Peripheral Reset Enable Register 1

The reset of each peripheral by Event Reset can be masked by this user setting. The PRER1/PRER2 register controls enablement 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
						TIMER9	TIMER8					TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA				DMA	PCU	WDT	FMC	SCU
0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
						RW	RW					RW	RW	RW	RW					RW	RW	RW	RW				RW	RW	RW	RW	RW

25	TIMER9	TIMER9 reset mask
24	TIMER8	TIMER8 reset mask
19	TIMER3	TIMER3 reset mask
18	TIMER2	TIMER2 reset mask
17	TIMER1	TIMER1 reset mask
16	TIMER0	TIMER0 reset mask
11	GPIOD	GPIOE reset mask
10	GPIOC	GPIOE reset mask
9	GPIOB	GPIOE reset mask
8	GPIOA	GPIOA reset mask
4	DMA	DMA reset mask
3	PCU	Port Control Unit reset mask
2	WDT	Watchdog Timer reset mask
1	FMC	Flash memory controller reset mask
0	SCU	System Control Unit reset mask

PRER2 Peripheral Reset Enable Register 2

The reset of each peripheral by Event Reset can be masked by this user setting. The PRER1/PRER2 register controls enablement 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.

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
										ADC1	ADC0				MWPM0							UART1	UART0				I2C0				SPI0
0	0	0	0	0	0	0	0	0	0	1	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				RW

21	ADC1	ADC1 reset enable
20	ADC0	ADC0 reset enable
16	MPWM0	MPWM0 reset enable
9	UART1	UART1 reset enable
8	UART0	UART0 reset enable
4	I2C0	I^2C0 reset enable
0	SPI0	SPI0 reset enable

PER1 Peripheral Enable Register 1

Prior to using a peripheral unit, it requires to be activated by writing **1** to the corresponding bit in the PER1/PER2 register. Until activation, the peripheral stays in Reset state.

To disable the peripheral unit, write **0** to the corresponding bit in the PER0/PER1 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
						TIMER9	TIMER8					TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA				DMA				
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	R	R	R	R

25	TIMER9	TIMER9 function enable
24	TIMER8	TIMER8 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
4	DMA	DMA function enable
3		Reserved
2		
1		
0		

PER2 Peripheral Enable Register 2

Prior to using a peripheral unit, it requires to be activated by writing **1** to the corresponding bit in the PER1/PER2 register. Until activation, the peripheral stays in Reset state.

To disable the peripheral unit, write **0** to the corresponding bit in the PER0/PER1 register, after which the peripheral enters the Reset state.

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
										ADC1	ADC0				MWPM0							UART1	UART0				I2C0				SPI0
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				RW

21	ADC1	ADC1 function enable
20	ADC0	ADC0 function enable
16	MPWM0	MPWM0 function enable
9	UART1	UART1 function enable
8	UART0	UART0 function enable
4	I2C0	I^2C0 function enable
0	SPI0	SPI0 function enable

PCER1 Peripheral Clock Enable Register 1

Prior to using a peripheral unit, its clock should be activated by writing **1** to the corresponding bit in the PCER1/PCER2 register. The peripheral will not operate correctly until its clock is 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
						TIMER9	TIMER8					TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA				DMA				
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	R	R	R	R

25	TIMER9	TIMER9 clock enable
24	TIMER8	TIMER8 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
4	DMA	DMA clock enable
3		Reserved
2		
1		
0		

PCER2 Peripheral Clock Enable Register 2

Prior to using a peripheral unit, its clock should be activated by writing **1** to the corresponding bit in the PCER1/PCER2 register. The peripheral will not operate correctly until its clock is enabled.

To stop the clock of the peripheral unit, write **0** 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
										ADC1	ADC0				MWPM0							UART1	UART0				I2C0				SPI0
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				RW

21	ADC1	ADC1 clock enable
20	ADC0	ADC0 clock enable
16	MPWM0	MPWM0clock enable
9	UART1	UART1 clock enable
8	UART0	UART0 clock enable
4	I2C0	I^2C0 clock enable
0	SPI0	SPI0 clock enable

CSCR Clock Source Control Register

The Z32F0641 MCU has multiple clock sources to generate internal operating clocks. Each clock source can be controlled by the Clock Source Control Register. This register is an 8-bit register.

CSCR=0x4000_0040

7	6	5	4	3	2	1	0
-		RINGOSCCON		-		EOSCCON	
00		10		00		00	
R		RW		R		RW	

5 4	RINGOSCCON	Internal ring oscillator control
		0 Stop internal sub oscillator X
		10 Enable internal sub oscillator
		11 Enable internal sub oscillator divide by 2
1 0	EOSCCON	External crystal oscillator control
		0 Stop External Ctystal oscillator X
		10 Enable External Ctystal oscillator
		11 Enable External Ctystal divide by 2

SCCR System Clock Control Register

The Z32F0641 MCU has multiple clock sources to generate internal operating clocks. Each clock source can be controlled by the System Clock Control Register. The MOSC MUST be running and stable before setting the FINSEL bit.

SCCR=0x4000_0044

7	6	5	4	3	2	1	0
-					FINSEL	MCLKSEL	
0000					0	00	
R					RW	RW	

2	FINSEL	PLL input source FIN select register
		0 IOSC clock is used as FIN clock
		1 MOSC clock is used as FIN clock
1 0	MCLKSEL	System clock select register
		0 Internal sub oscillator X
		10 PLL bypassed clock
		11 PLL output clock

Note: When changing FINSEL, both internal OSC and external OSC should be alive to prevent the chip from mal functioning.

CMR Clock Monitoring Register

To monitor the internal clock and external oscillator, the MCLKMNT/EOSCMNT bits must be set before the MCLK and EOSC bits are valid. The Clock Monitoring Register is a 16-bit register.

Note: The EOSC bit only checks for the EOSC oscillation, not its stability. When the system detects an MCLKFAIL interrupt, the MCLKREC bit determines if the system dies or will auto-recover using the ROSC. The system usually auto-recovers so that it can continue running.

CMR=0x4000_0048

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCLKREC								MCLKMNT	MCLKIE	MCLKFAIL	MCLKSTS	EOSCMNT	EOSCIE	EOSCFAIL	EOSCSTS
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R								RW	RW	RC1	RC1	RW	RW	RC1	RC1

15	MCLKREC	MCLK fail auto recovery
		0 MCLK is changed to RINGOSC by default when MCLKFAIL issued
		1 MCLK auto recovery is disabled
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	EOSCMNT	External oscillator monitoring enable
		0 External oscillator monitoring disabled
		1 External oscillator monitoring enabled
2	EOSCIE	External oscillator fail interrupt enable
		0 External oscillator fail interrupt disabled
		1 External oscillator fail interrupt enabled
1	EOSCFAIL	External oscillator fail interrupt
		0 External oscillator fail interrupt not occurred
		1 Read : External oscillator fail interrupt is pending Write : Clear pending interrupt
0	EOSCSTS	External oscillator status
		0 Not oscillate
		1 External oscillator is working normally

The clock monitoring function cannot cover all malfunction cases and is only used for reference. Figure 4.9 shows the operational diagram for clock monitoring function.

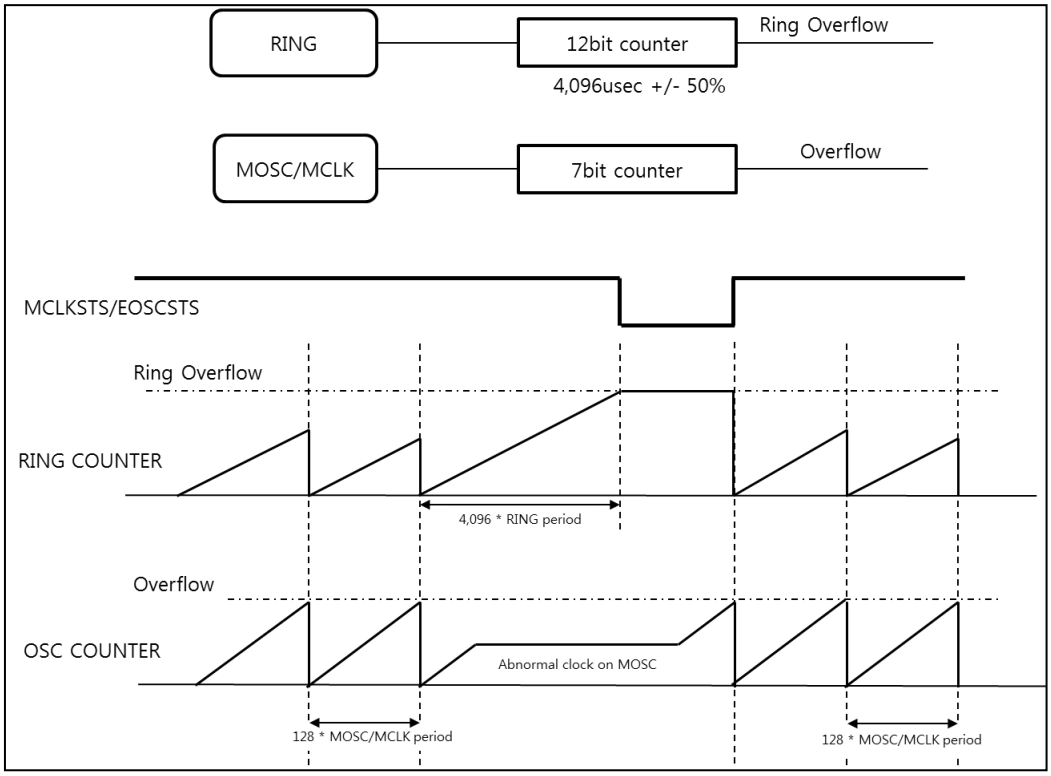


Figure 4.9. Clock Monitoring Function Diagram

NMIR Non-Maskable Interrupt Control Register

The Non-Maskable Interrupt Control Register can be set with software. There are five kinds of interrupt sources from MPWM, WDT, and SCU. Write access key `0xA32C` to NMR[31:16] is required before 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	WDTSTS	MCLKFAILSTS	LVDSTS				PROTEN	OVPEN	WDTEN	MCLKFAILEN	LVDEN
-		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
WO					R	R	R	R	R				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	WDTSTS	WDT Interrrupt 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 interru pt
		0 Disable
		1 Enable
2	WDTEN	WDT Interrrupt 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 Z32F0641 MCU can drive the clock from internal MCLK clock with a dedicated post divider. 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			
R			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) $\mathbf{CLKO} = \frac{\boldsymbol{MCLK}}{\mathbf{2} * (\mathbf{CLKODIV} + \mathbf{1})}$ $(\boldsymbol{CLKODIV} > \mathbf{0})$

PLLCON PLL Control Register

Integrated PLL can synthesize the high speed clock for extremely high performance of the CPU from either the internal oscillator (IOSC) or the external oscillator (MOSC). The PLL Control register provides the configuration for the PLL system. By default, the PLL system is in reset mode and disabled. You must negate the reset and enable the PLL to operate (bits 14 and 15 must be set). The Bypass bit must be set to output the PLL clock. The active clock is defined in SCCR bit 2 (FIN).

To calculate the PLL output:

PLL Out = ((Active clock / PREDIV) * FBCTRL) / POSTDIV

For example:

Using MOSC (assuming it is running at 8 MHz and selected):

PREDIV set to 1	(FIN / 2)
FBCTRL set to 0x04	(M=12)
POSTDIV set to 0x00	(N=1)

((8 MHz / 2) * 12) = 48 MHz

PLLCON=0x4000_0060

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLRSTB	PLLEN	BYPASS	LOCKSTS				PREDIV	FBCTRL				POSTDIV			
0	0	0	0	0	0	0	0	0000				0000			
RW	RW	RW	R				RW	RW				RW			

15	PLLRSTB	PLL reset	
		0 PLL reset is asserted	
		1 PLL reset is negated	
14	PLLEN	PLL enable	
		0 PLL is disabled	
		1 PLL is enabled	
13	BYPASS	FIN bypass	
		0 FOUT is bypassed as FIN	
		1 FOUT is PLL output	
12	LOCK	LOCK status	
		0 PLL is not locked	
		1 PLL is locked	
8	PREDIV	FIN predivider	
		0 FIN divided by 1	
		1 FIN divided by 2	
7 4	FBCTRL	Feedback control	
		0000 M = 4	1000 M = 20
		0001 M = 6	1001 M = 24
		0010 M = 8	1010 M = 26
		0011 M = 10	1011 M = 34
		0100 M = 12	1100 Not available
		0101 M = 14	1101
		0110 M = 16	1110
		0111 M = 18	1111
3 0	POSTDIV	Post divider control	
		000 N = 1	
		001 N = 2	
		010 N = 3	

011	N = 4
100	N = 6
101	N = 8
110	N = 3
111	N =16

VDCCON VDC Control Register

The on chip VDC control register, VDCTRIM, is used for the trim value of VDC output. To modify the VDCTRIM bit, **1** should be written to VDCTE simultaneously. The VDCWDLY value can be written by writing **1** to the VDCDE bit simultaneously.

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
RESERVE					RESERVED			VDCTE				VDCTRIM											VDCDE	VDCWDLY							
0	0	0	0	0	00			0	0	0	0	0000				0	0	0	0	0	0	0	0	0x7F							
W					RW			W				RW											W	RW							

23	VDCTE	VDCTRIM value write enable. Write only with VDCTRIM value.
		0 VDCTRIM field is not updated by writing
		1 VDCTRIM filed can be updated by writing
19 16	VDCTRIM	VDC output voltage trim value
8	VDCDE	VDCWDLY value write enable. Write only with VDCWDLY value
		0 FOUT is PLL output
		1 FOUT is bypassed as FIN
7 0	VDCWDLY	VDC warm-up delay count value. When SCU is waked up from powerdown mode, the warm-up delay is inserted for VDC output being stabilized. The amount of delay can be defined with this register value 7F : 2msec

LVDCON LVD Control Register

The LVD Control Register is an on chip brown-out detector control register. This 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
								LVDTE						LVDTRIM		SELEN						LVDSEL								LVDLVL	LVDEN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	00		0	0	0	0	0	0	00		0	0	0	0	0	0	0	1
								RW						RW		RO						RW								RO	RW

23	LVDTE	LVDTRIM value write enable. Write only with LVDTRIM value.
		0 LVDTRIM field is not updated by writing
		1 LVDTRIM filed can be updated by writing
17 16	LVDTRIM	LVD voltage level trim value It can writable when trim enable mode in FMC
15	SELEN	LVD level SEL value write enable. Write only.
		0 SEL field is not updated by writing
		1 SEL filed can be updated by writing
9 8	LVDSEL	LVD detect level select
		00 LVD detect level is 1.8V- 50mV
		01 LVD detect level is 2.2V – 50mV
		10 LVD detect level is 2.7V -50mV
		11 LVD detect level is 4.3V – 50mV
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

EOSCR External Oscillator Control Register

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

EOSCR=0x4000_0080

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISELEN						ISEL									
0	0	0	0	0	0	11		0	0	0	0	0	0	0	0
W						RW									

15	ISELEN	Write enable of bit field ISEL.
		0 Write access of ISEL field is masked
		1 Write access of ISEL field is accepted
9	ISEL	Select current.
8		00 Minimum current driving option
		01 Low current driving option
		10 High current driving option
		11 Maximum current driving option

EMODR External Mode Status Register

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
							BOOT
		0x0			0	0	-
		R			R	R	R

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

DBCLK1 Debounce Clock Control Register 1

The Debounce Clock Control register 1 controls the debounce timing configuration for Port A and Port B.

DBCLK1=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 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
23 16	PBDDIV	PORT B Debounce Clock N divider
10 8	PADCSEL	Debounce Clock for Port A source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
7 0	PADDIV	PORT A Debounce Clock N divider

DBCLK2 Debounce Clock Control Register 2

The Debounce Clock Control register 2 controls the debounce timing configuration for Port C and Port D.

DBCLK2=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 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
23 16	PDDDIV	PORT D Debounce Clock N divider
10 8	PCDCSEL	Debounce Clock for PORT C source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
7 0	PCDDIV	PORT C Debounce Clock N divider

MCCR1 Miscellaneous Clock Control Register 1

The Miscellaneous Clock Control register 1 controls the configuration for the System Tick clocks.

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
																					STCSEL			SYSTICKDIV							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000			0x01							
																					RW			RW							

10 8	STCSEL	SYSTIC Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
7 0	STDIV	SYSTIC Clock N divider

MCCR2 Miscellaneous Clock Control Register 2

The Miscellaneous Clock Control register 2 controls the optional configuration of MPWM0 clocks.

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
																					PWM0CSEL			PWM0DIV							
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	PWM0CSEL	PWM0 Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
7 0	PWM0DIV	PWM0 Clock N divider

MCCR3 Miscellaneous Clock Control Register 3

The Miscellaneous Clock Control register 3 controls the configuration for the Timer EXT0 and WDT clocks.

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
					TEXT0CSEL			TEXT0DIV													WDTCSEL			WDTDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW													RW			RW							

26 24	TEXT0CSEL	Timer EXT0 Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
23 16	TEXT0DIV	Timer EXT0 Clock N divider
10 8	WDTCSEL	WDT Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 Reserved 110 External Main OSC (XTAL) 111 Reserved
7 0	WDTDIV	WDT Clock N divider

MCCR4 Miscellaneous Clock Control Register 4

The Miscellaneous Clock Control Register 4 controls the clock setting for the ADC peripheral.

MCCR7=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																								
0	0	0	0	0	000			0x01																							
					RW			RW																								

26 24	ADCCSEL	ADC clock source select bit	
		0xx	RING OSC 1Mhz
		100	MCLK (bus clock)
		101	Reserved
		110	External Main OSC (XTAL)
		111	Reserved
23 16	ADCCDIV	ADC Clock N divider	

5. Port Control Unit

Overview

The Port Control Unit (PCU) controls the external I/O configuration to:

- Set the multiplex state of each pin (for alternative functions)
- Set external signal type (Analog / Push-Pull output /Open Drain output /Input)
- Set enable/monitor/trigger type for interrupts for each pin
- Set internal pull-up register control for each pin
- Set debounce for each pin

Note: You must enable both the Port Periphreal and the Port Periphreal CLOCK in PER1/PCER1/ to use the pins of the port.

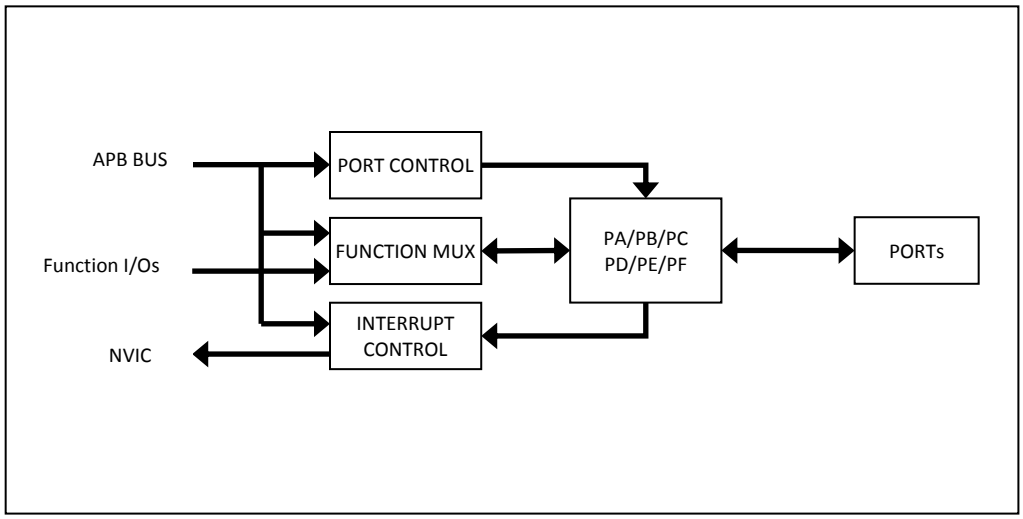


Figure 5.1 Block Diagram

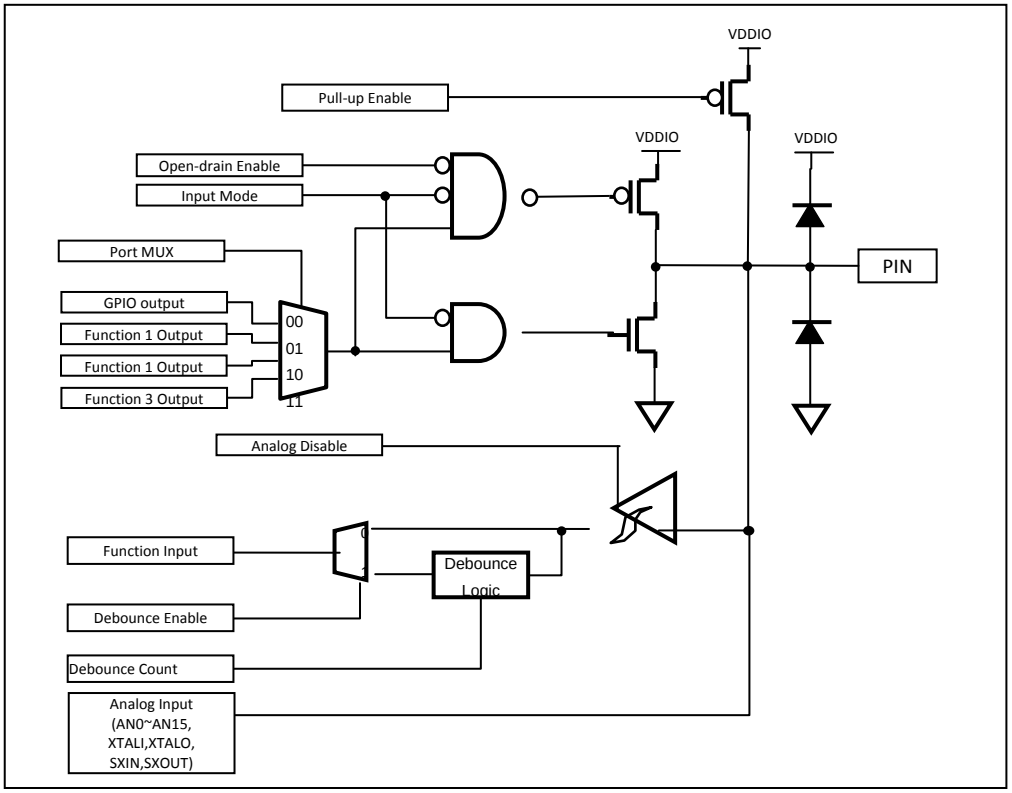


Figure 5.2 I/O Port Block Diagram (ADC and External Oscillator pins)

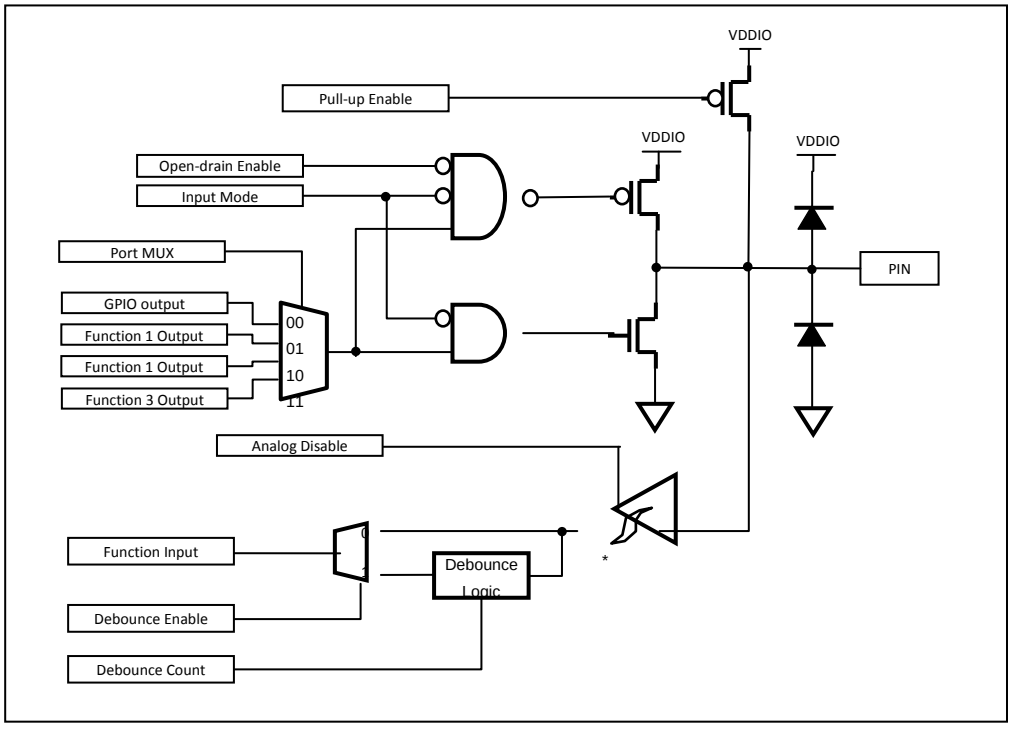


Figure 5.3 I/O Port Block Diagram (General I/O pins)

Pin Multiplexing

GPIO pins have alternative function pins. Table 5.1 Table 5.1 GPIO Alternative functionshows pin multiplexing information.

Table 5.1 GPIO Alternative function

Port	Pin	Function			
		00	01	10	11
PA	0	PA0*			AN0
	1	PA1*			AN1
	2	PA2*		WDTO	AN2
	3	PA3*			AN3
	4	PA4*	SS1		AN4
	5	PA5*	SS2		AN5
	6	PA6*	T0IO	T2IO	AN6
	7	PA7*	T1IO	T3IO	AN7
	8	PA8*	T2IO	T0IO	AN8
	9	PA9*	T3IO	T1IO	AN9
	10	PA10*	SS3		AN10
	11	PA11*			
	12	PA12*	T0IO		
	13	PA13*	T1IO		
	14	PA14*	T2IO		
	15	PA15*	T3IO		
PB	0	PB0*	MP0UH		
	1	PB1*	MP0UL		
	2	PB2*	MP0VH		
	3	PB3*	MP0VL		
	4	PB4*	MP0WH		
	5	PB5*	MP0WL		
	6	PB6*	PRTIN0	T0IO	
	7	PB7*	OVIN0	T1IO	
	8				
	9				
	10				
	11				
	12				
	13				
	14				
	15				

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

Table 5.2 GPIO Alternative Function

Port	Pin	Function			
		00	01	10	11
PC	0	PC0	TCK/SWCLK*	RXD1	
	1	PC1	TMS/SWDIO*	TXD1	
	2	PC2	TDO/SWO*	T8IO	
	3	PC3	TDI*	T9IO	
	4	PC4	nTRST*	T0IO	
	5	PC5*	RXD1	T1IO	
	6	PC6*	TXD1	T2IO	
	7	PC7*	SCL0	T3IO	
	8	PC8*	SDA0		
	9	PC9*	CLKO		
	10	PC10	nRESET*	T8IO	
	11	PC11/BOOT*		T9IO	
	12	PC12*			XIN
	13	PC13*			XOUT
	14	PC14*	RXD0	MOSI0(2)	
	15	PC15*	TXD0	MISO0(2)	
PD	0	PD0*	SS0	T8IO	
	1	PD1*	SCK0	T9IO	
	2	PD2*	MOSI0	SCL0	
	3	PD3*	MISO0	SDA0	
	4				
	5				
	6				
	7				
	8				
	9				
	10				
	11				
	12				
	13				
	14				
	15				

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

Registers

The base address of the PCU block is `0x4000_1000`.

Table 5.3 Base Address of Port

Port Name	Address
PCA	0x4000_1000
PCB	0x4000_1100
PCC	0x4000_1200
PCD	0x4000_1300

Table 5.4 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 register is the PA Port Mode select register, and must be set up correctly before using the port to ensure that the port functions as designed.

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*			AN0
PA1	PA1*			AN1
PA2	PA2*		WDTO	AN2
PA3	PA3*			AN3
PA4	PA4*	SS1		AN4
PA5	PA5*	SS2		AN5
PA6	PA6*	T0IO	T2IO	AN6
PA7	PA7*	T1IO	T3IO	AN7
PA8	PA8*	T2IO	T0IO	AN8
PA9	PA9*	T3IO	T1IO	AN9
PA10	PA10*	SS3		AN10
PA11	PA11*			
PA12	PA12*	T0IO		
PA13	PA13*	T1IO		
PA14	PA14*	T2IO		
PA15	PA15*	T3IO		

PCB.MR PORT B Pin MUX Register

This register is the PB Port Mode select register, and must be set up correctly before using the port to ensure that the port functions as designed.

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*	MP0UH		
PB1	PB1*	MP0UL		
PB2	PB2*	MP0VH		
PB3	PB3*	MP0VL		
PB4	PB4*	MP0WH		
PB5	PB5*	MP0WL		
PB6	PB6*	PRTIN0	T0IO	
PB7	PB7*	OVIN0	T1IO	

PCC.MR PORT C Pin MUX Register

This register is the PC Port Mode select register, and must be set up correctly before using the port to ensure that the port functions as designed.

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	01	01	01	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	TCK/SWCLK*	RXD1	
PC1	PC1	TMS/SWDIO*	TXD1	
PC2	PC2	TDO/SWO*	T8IO	
PC3	PC3	TDI*	T9IO	
PC4	PC4	nTRST*	T0IO	
PC5	PC5*	RXD1	T1IO	
PC6	PC6*	TXD1	T2IO	
PC7	PC7*	SCL0	T3IO	
PC8	PC8*	SDA0		
PC9	PC9*	CLKO		
PC10	PC10	nRESET*	T8IO	
PC11	PC11/BOOT*		T9IO	
PC12	PC12*			XIN
PC13	PC13*			XOUT
PC14	PC14*	RXD0	MOSI0[2]	
PC15	PC15*	TXD0	MISO0[2]	

PCD.MR PORT D Pin MUX Register

This register is the PD Port Mode select register, and must be set up correctly before using the port to ensure that the port functions as designed.

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*	SS0	T8IO	
PD1	PD1*	SCK0	T9IO	
PD2	PD2*	MOSI0	SCL0	
PD3	PD3*	MISO0	SDA0	

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

This register controls the input or output 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

PCCCR PORT C Pin Control Register

This register controls the input or output 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	10	10	00	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

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

PCA.PCR=0x4000_1008, PCB.PCR=0x4000_1108
PCC.PCR=0x4000_1208, 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

PCn.DER PORT n Debounce Enable Register

Every pin in the port has a digital debounce filter which can be configured by the PnDER registers. . The Debounce clock can be configured in the DBCLKx 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

Each individual pin can be an external interrupt source. The edge trigger interrupt and level trigger interrupt are both supported. The interrupt mode can be configured by setting the PnIER 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 PnISR register. PnISR register reports an interrupt source pin and an interrupt type.

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 of rising and falling edge interrupt mode. Not support for level trigger mode

PORTEN Port Access Enable

Port Access Enable provides register writing permission 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.

Functional Description

All the GPIO pins can be configured for different operations – inputs, outputs, and triggered interrupts (both level and edge) through the PDU. The system is also able to disable ports by setting the PER1 and PCER1 registers in the SCU. By default, all pins are disabled (except for UART0/SPI0) so the developer must enable these to operate.

All configuration parameters are protected by the Port Access Enable register. You must write the sequence in order (`0x15`, `0x51`) to the PORTEN register to configure any pin(s). Once the configuration is complete, write any other value to the PORTEN register to lock it.

Note: Do not read in between the sequence; it will prevent the configuration registers from being unlocked.

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

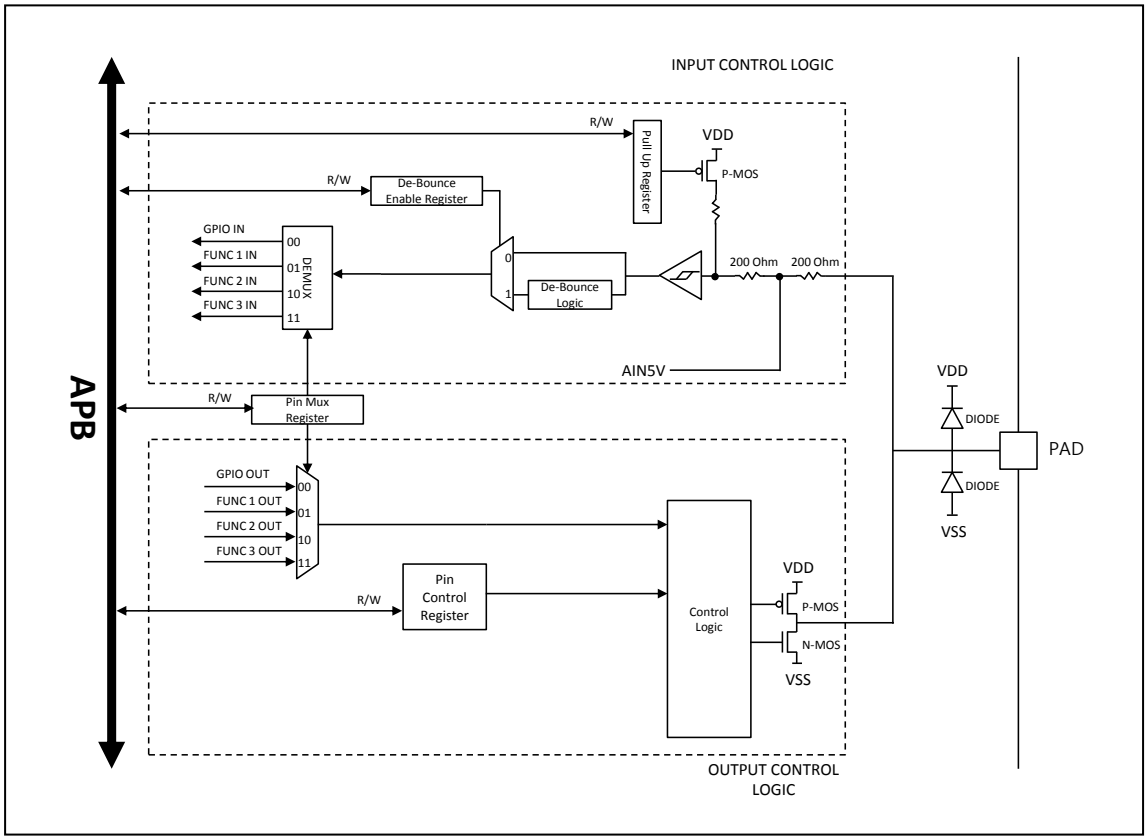


Figure 5.4. Port Diagram

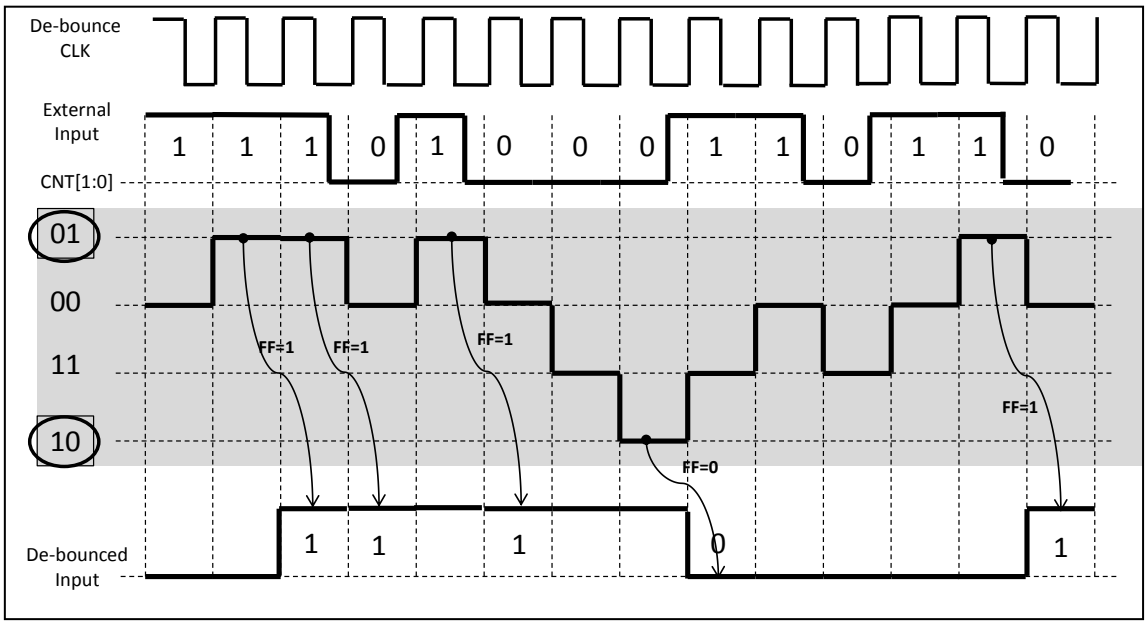


Figure 5.5. Debounce Function

When the Debounce function of input data is 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”

The Debounce CLK of each port group can be configured by the DBCLK Registers.

6. General Purpose I/O

Overview

Most pins, except the 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
- Input signal level
- Output Set/Clear pin by writing a **1**

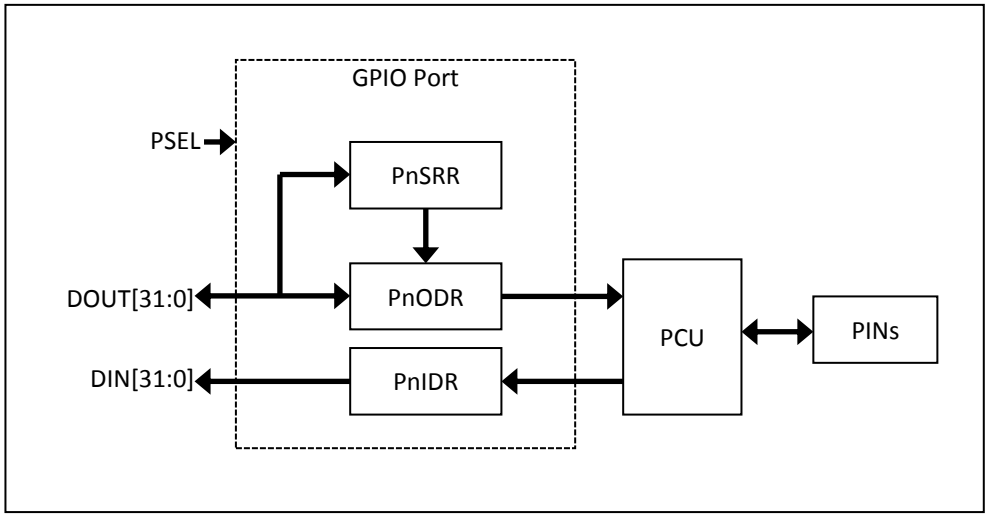


Figure 6.1 Block Diagram

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

Port	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 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 a mode other than Analog Mode, the pin level can be detected in the PnIDR 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
PnIDR															
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 PnODR register. Writing a **1** into the specific bit will set a corresponding bit of PnODR 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 PnODR register

Pn.BCR PORT n Bit Clear Register

Pn.BRR is a register for controlling each bit of the PnODR register. Writing a **1** into the specific bit will set a corresponding bit of PnODR 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
PnBCR															
0000															
WO															

BCR	Pin current level
0	Not effect
1	Clear correspondent bit in PnODR register

Functional Description

The GPIO registers provide the input/output condition of the GPIO pins. The input data registers give the states of the pins of the ports. The output data register is for setting the port pins. The Set and Clear registers control the pins at the individual level.

When configured as output, the value written to the GPIO Ouput 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.

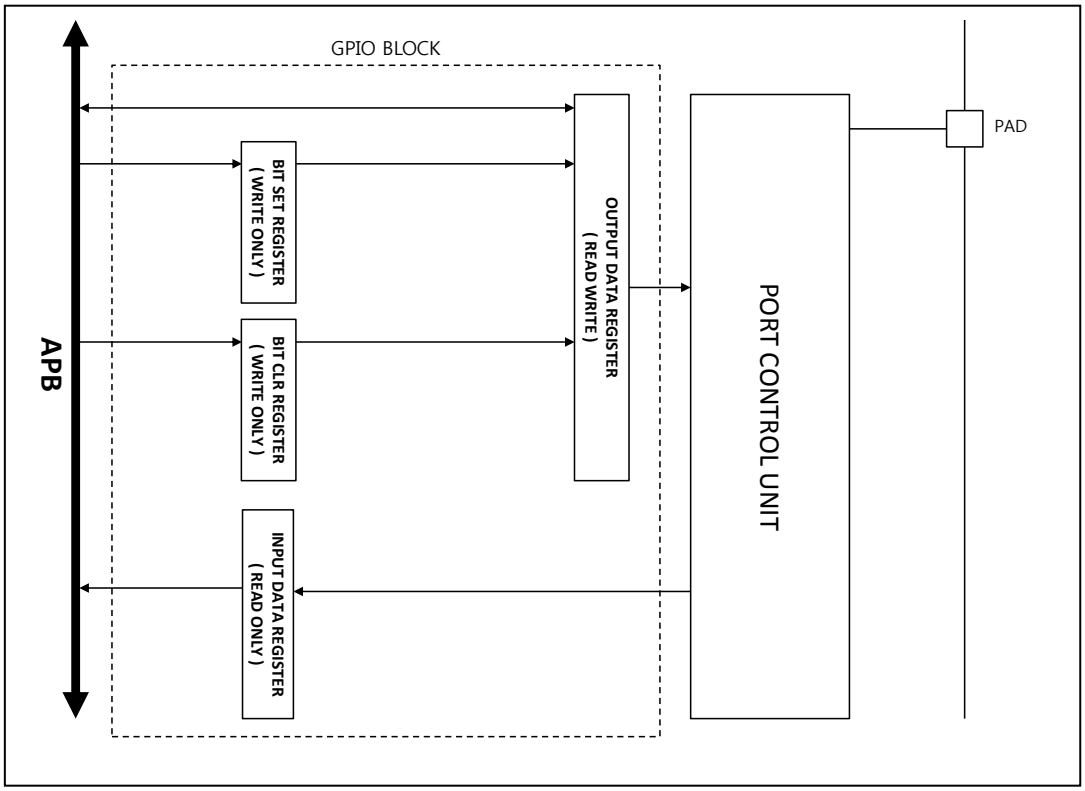


Figure 6.2. GPIO Diagram

7. Flash Memory Controller

Introduction

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

- 64KB Flash code memory
- 32-bit read data bus width
- Code cache block for fast access mode
- 128-byte page size
- Support page erase and macro erase
- 128-byte unit program

Note: Programming the Flash requires the execution to occur in RAM. Once the Program mode is selected, Flash is no longer able to be read for instructions.

Table 7.1 Internal Flash Specification

Item	Description
Size	64KB
Start Address	0x0000_0000
End Address	0x0000_FFFF
Page Size	128-byte
Total Page Count	512 pages
PGM Unit	128-byte
Erase Unit	128-byte

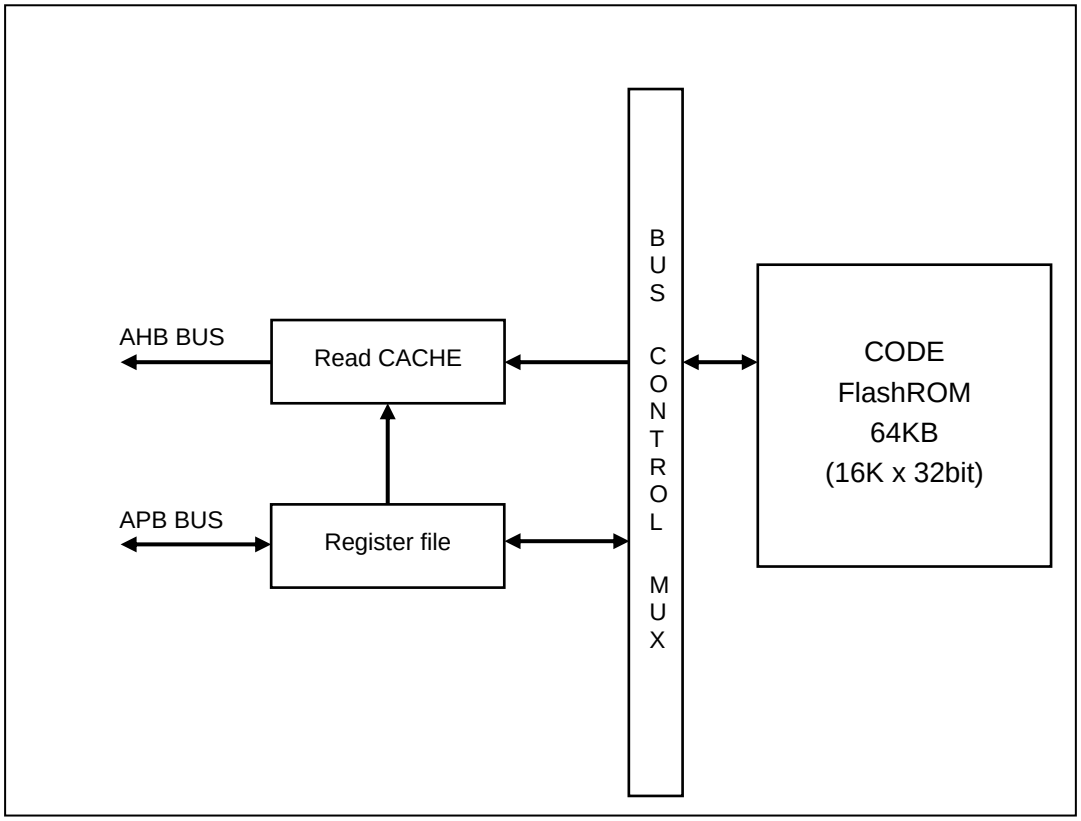


Figure 7.1 Block Diagram

Pin Description

The Flash Memory Controller has no external interface pins.

Registers

The base address of the Flash Memory Controller is `0x4000_0100`.

Table 7.2 shows the register memory map.

Table 7.2 Flash Memory Controller 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	0x00000000
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	0x000000bb
FM.DRTY	0x0018	RW	Flash Memory Dirty bit	
FM.TICK	0x001C	RO	Flash Memory Tick Timer	0x00000000
FM.CRC	0x0020	RO	Flash Memory Read CRC Value	
FM.BOOTCR	0x0074	RW	Boot ROM Remap Clear register	0x00000000
FM.PROT	0x0078	RW	Flash Page protection register	0x00000000
FM.JTAGEN	0x007C	RW	Jtag protection register	0x00000001

FM.MR Flash Memory Mode Register

The Flash Memory Mode Register is an internal Flash memory mode 32-bit 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
BOOT							IDLE	VERIFY	AMBAEN					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	RW	RW					RW	RW							R	R	RW						

31	BOOT	0	
		1	Boot mode enable status(read only)
24	IDLE	0	
		1	Boot mode enable status(read only)
23	VERIFY	0	
		1	Flash Verify mode enable status(read only)
22	AMBAEN	0	AMBA mode disable
		1	AMBA mode enable (can change wait state and etc)
17	TRMEN	0	
		1	Trim mode entry status(read only)
16	TRM	0	
		1	Trim mode status(read only)
9	FEMOD	0	
		1	Flash mode entry status(read only)
8	FMOD	0	
		1	Flash mode status(read only)
7	ACODE	5A → A5	Flash mode
0		A5 → 5A	Trim mode

FM.CR Flash Memory Control Register

The Flash Memory Control Register is an internal Flash memory control register.

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
											TIMER			TEST1	TEST0	VPPOUT	EVER	PVER		OTPBE	OTPAE	PPGM	AE			PMOD	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	R	RW	RW		RW	RW		RW			RW	RW	RW	RW	RW	RW

20	TIMER	0	Program/Erase timer enable
		1	(timer can be enable by PGM or ERS bit)
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		Enable charge-pump Vpp output
14	EVER		Set erase verify mode
13	PVER		Set program verify mode
11	OTPBE		OTP area B enable
10	OTPAE		OTP area A enable
9	PPGM		Pre PGM enable Page buffer set automatically
8	AE		All erase enable
5	PMODE		PMODE enable(Address path changing)
4	WE		Write enable
3	PBLD		Page buffer load(WE should be set)
2	PGM		Program enable
1	ERS	0	Program mode enable
		1	Erase mode enable
0	PBR		Page buffer reset

FM.AR Flash Memory Address Register

The Flash Memory Address Register 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		0x0000													
		RW													

13 0	FADDR	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

The Flash Memory Timer Register is an internal Flash memory Timer value register (16-bit). Erase/Program timer runs up to TMR[15:0]

FM.TMR=0x4000_0114

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR															
0x09C4															
RW															

7 0	TMR	Erase/PGM timer (default, 0x09C4) Timer counts up to TMR[15:0] by 1MHz int. OSC clock or External OSC clock. It can be selected in TMRCK bit.

FM.DRTY Flash Memory Dirty Bit Register

This is an internal Flash memory dirty bit clearing register.

FM.DRTY=0x4000_0118

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

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

Note: This device has a small internal cache. All cache lines are cleared when any data is written to this register.

FM.TICK Flash Memory Tick Timer Register

The Flash Memory Tick Timer Register is an internal Flash memory Burst Mode channel selection 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	0	0	0x00000															
								RW																							

17 0	FTICK	TICK goes to 0x3FFFF from written TICK value while TRM runs by PCLK clock

FM.CRC Flash Memory CRC Value Register

The Flash Memory CRC Value Register is the Cyclic Redundancy Check (CRC) value resulting from read access on internal Flash memory.

FM.CRC=0x4000_0120

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

15 0	CRC	CRC16 value

FM.CFG Flash Memory Config Value Register

The Flash Memory Config Value Register is the Flash configuration 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
WRITE KEY																HRESPD			TMRCK		WAIT			CRCINIT	CRCEN			TRIM			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1			0		0	1	1	0	0			0			
																												RW			

31 15	WRITE KEY		KEY Value : 0x7858
15	HRESPD		Disable HRESP(error response function) of Data or System bus (HRESP is AMBA AHB signal)
12	TMRCK	0	PGM/ERASE timer source is 1MHz RINGOSC PRM/ERASE timer source is External Clock
10 8	WAIT	000	No wait access for flash memory
		001	1-wait inserted for flash access
		010	2-wait inserted for flash access
		011	3-wait inserted for flash access
7	CRCINIT	0 1	CRC register winll be initialized. It should be reset again before read flash to generate CRC16 calculation (Initial value of FMCRC is 0xFFFF)
6	CRCEN	0 1	CRC16 enable CRC value will be calculated at every flash read timing
3 0	TRIM		FLASH TRIM Value (trim_mode_entry)

FM.BOOTCR Boot ROM Remap Clear Register

This is the Boot ROM remap clear register. This register is an 8-bit register.

FM.BOOTCR=0x4000_0174

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

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 (when BOOTROM low, external BOOT pin signal is masked)

FM.PROTECT Write Protection Control Register

The Write Protection Control Register is an internal Flash memory control register.

FM.PROTECT=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
WRITE_KEY																WP15	WP14	WP13	WP12	WP11	WP10	WP9	WP8	WP7	WP6	WP5	WP4	WP3	WP2	WP1	WP0
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

15	WP15	0xF000 ~ 0xFFFF, write_key is 0x87
14	WP14	0xE000 ~ 0xEFFF, write_key is 0x87
13	WP13	0xD000 ~ 0xDFFF, write_key is 0x87
12	WP12	0xC000 ~ 0xCFFF, write_key is 0x87
11	WP11	0xB000 ~ 0xBFFF, write_key is 0x87
10	WP10	0xA000 ~ 0xAFFF, write_key is 0x87
9	WP9	0x9000 ~ 0x9FFF, write_key is 0x87
8	WP8	0x8000 ~ 0x8FFF, write_key is 0x87
7	WP7	0x7000 ~ 0x7FFF, write_key is 0x87
6	WP6	0x6000 ~ 0x6FFF, write_key is 0x87
5	WP5	0x5000 ~ 0x5FFF, write_key is 0x87
4	WP4	0x4000 ~ 0x4FFF, write_key is 0x87
3	WP3	0x3000 ~ 0x3FFF, write_key is 0x87
2	WP2	0x2000 ~ 0x2FFF, write_key is 0x87
1	WP1	0x1000 ~ 0x1FFF, write_key is 0x98
0	WP0	0x0000 ~ 0x0FFF, write_key is 0x98

FM.JTAGEN JTAG Protection Control Register

The JTAG Protection Control Register is a debug access control register.

FM.JTAGEN=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
WRITE_KEY																															JTAGEN
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	1
																															RW

0	JTAGEN	0 1	Debug access port is disabled, write access code is 0xC7 Debug access port is enabled

Functional Description

Flash Memory Controller is an internal Flash memory interface controller. It mainly controls the program Flash memory operation and prepares read data for requesting from the bus.

Flash Organization

The 64 Kbytes code Flash memory consists of 1,024 pages which have a uniform 128-bytes page size. The Flash controller allows for reading or writing a data of the Flash memory. Read access can be performed by 8, 16, and 32 bits wide.

This momory is located at 0x0000_0000 address on the system memory map. The system boot address is 0x0000_0000, so this Flash memory is boot memory. The code data which is programmed in the Flash memory will boot up the device after the boot ROM sequence is completed.

Flash Read Operation

The Flash data read operation is requested from the bus. The Flash controller responds to the request by itself. The wait time should be correctly defined because the bus speed is usually faster than Flash data access time.

The normal read operation is not available in FLASH MODE in the ACODE.FM.MR field.

Flash Program Operation

The erase and program access of Flash memory is available only in FLASH MODE in the ACODE.FM.MR field. Therefore, self-programming is not supported. The Flash program/erase operation should be performed by the execution program on the SRAM memory.

The Flash program operation writes one page to the target address selected by the FM.AR register. At first, users should write the program data into the page buffer. Page buffer write is pefromed by word write access to the FM.DR register on the FM.AR address.

After filling the page buffer, users can start the Flash write operation and should wait for the IDLE bit to be set.

Figure 7.2 shows the page buffer loading operation.

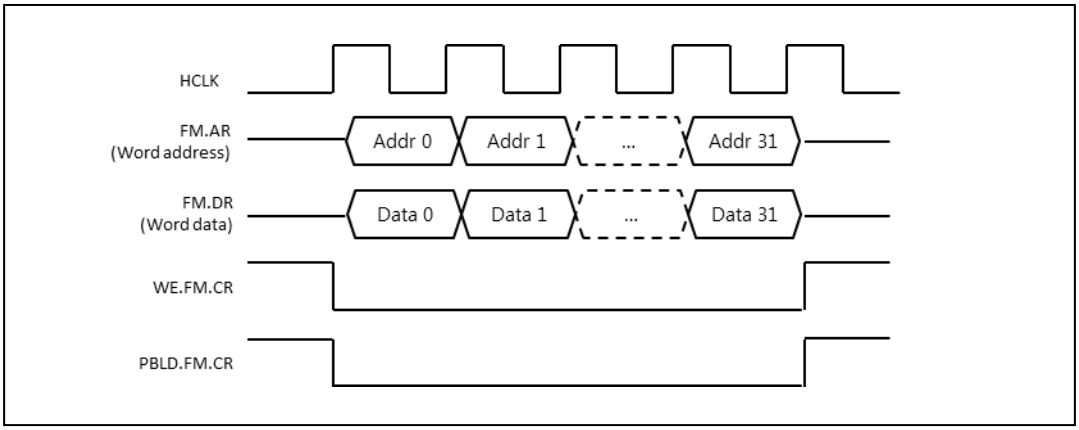


Figure 7.2. Page Buffer Load Timing Diagram

The Flash write of page buffer data is performed by the PRGM.FM.CR command. Safe writing operation requires correct program time. The program time tPGM is defined by the FM.TMR register. This timer counts the number of HCLK clock to the FM.TMR value. When the timer count starts, the IDLE.FM.MR register is cleared. When the timer count is completed, the IDLE.FM.MR register is set.

In this page write operation, the target page address should be written in the FM.AR register.

Figure 7.3 shows the page write operation.

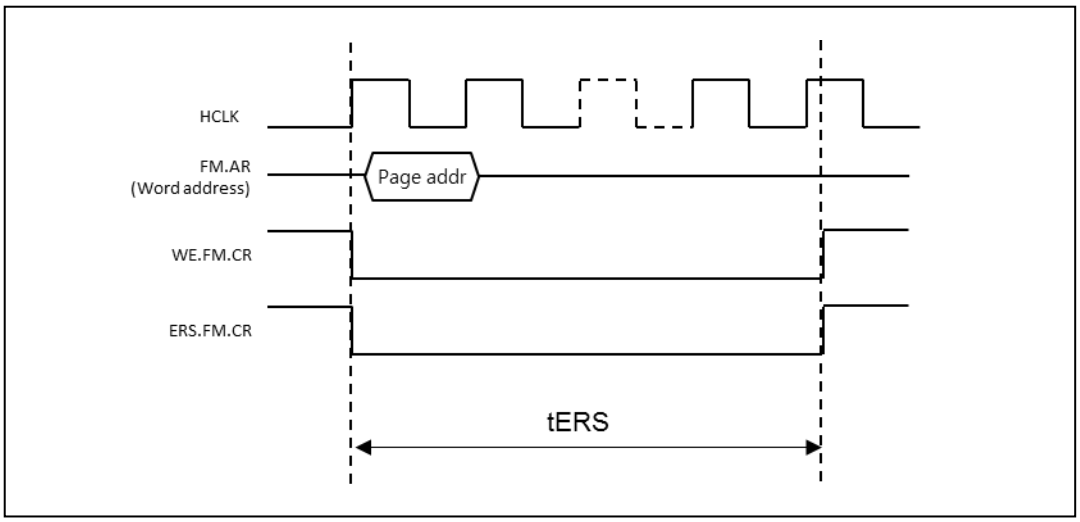


Figure 7.3.Page Erase Timing Diagram

The Flash erase of page data is done by the ERS.FM.CR command. Safe writing operation requires correct program time. The erase time tERS is defined by the FM.TMR register. This timer counts the number of HCLK clock to the FM.TMR value. When the timer count starts, the IDLE.FM.MR register is cleared. When the timer count is completed, the IDLE.FM.MR register is set.

Figure 7.4 shows the bulk erase operation.

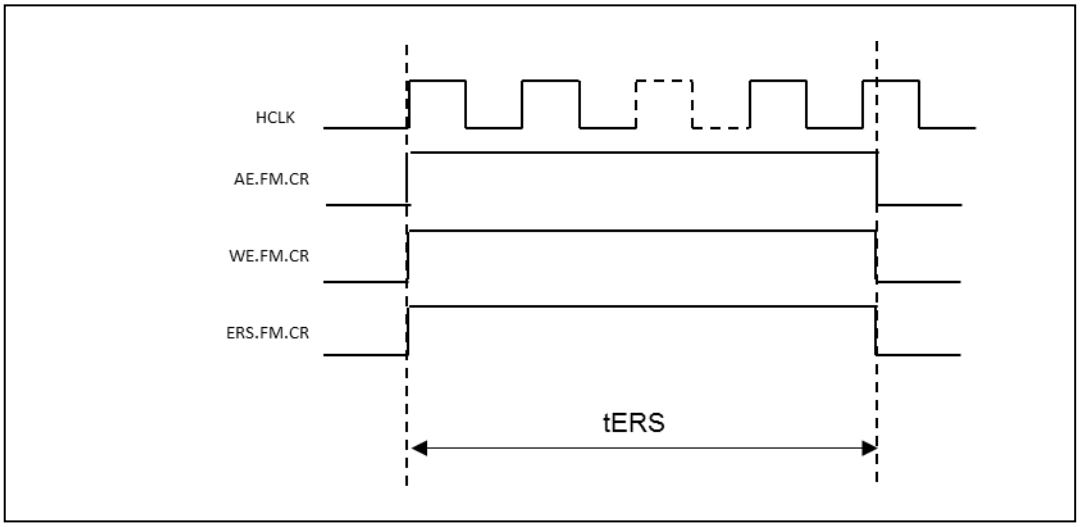


Figure 7.4. Bulk Erase Timing Diagram

8. Internal SRAM

Overview

The Z32F0641 MCU has a block of 0-wait on-chip SRAM. The size of SRAM is 8 KB. 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/PGM operation. This device does not support a memory remap strategy; therefore, jump and return is required to execute the code in the SRAM memory area.

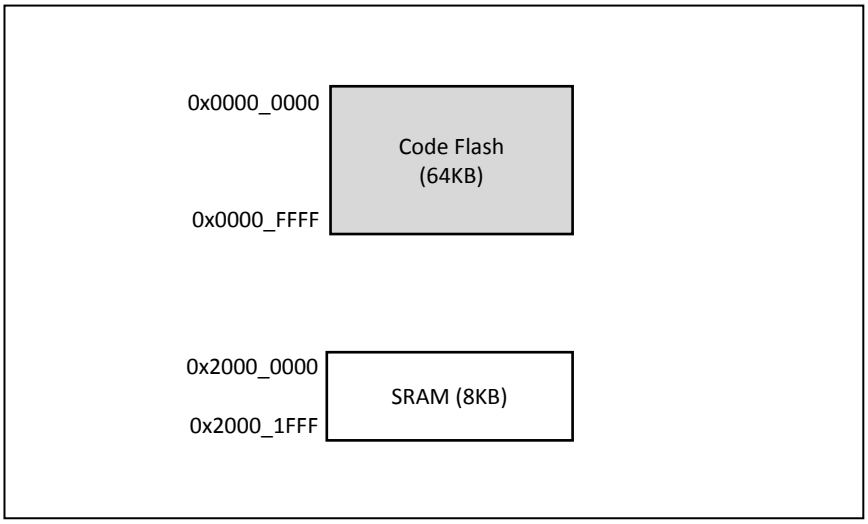


Figure 8.1 SRAM Block Diagram

9. Direct Memory Access Controller

Introduction

Features of the Direct Memory Access Controller (DMAC) include:

- Four channels
- Single transfer only
- Supports 8/16/32-bit data size
- Supports multiple buffers with same size
- Interrupt condition is transferred through peripheral interrupt

A block diagram of the DMAC is shown in Figure 9.1.

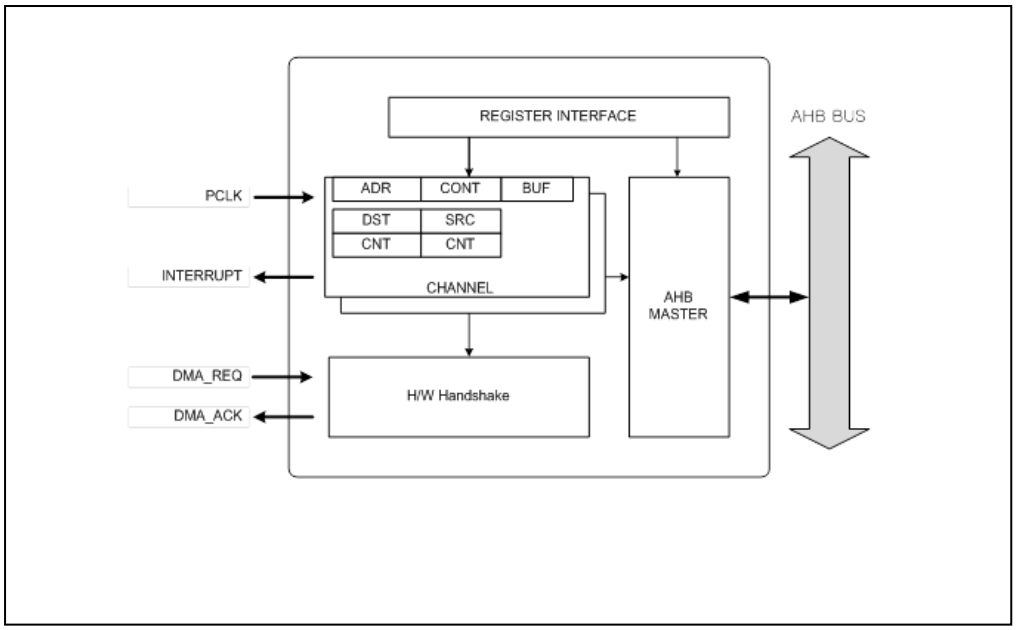


Figure 9.1 Block Diagram

Pin Description

The DMAC has no external interface pins.

Registers

The base address of the DMA controller is shown in Table 9.1.

Table 9.1 DMA Controller Base Address

Channel	Base Address
DMACH0	0x4000_0400
DMACH1	0x4000_0410
DMACH2	0x4000_0420
DMACH3	0x4000_0430

Table 9.2 shows the register map of the DMA controller.

Table 9.2 DMAC Register Map

Name	Offset	Type	Description	Reset Value
DC*n*.CR	0x0000	RW	DMA Channel n Control Register	0x0000_0000
DC*n*.SR	0x0004	RW	DMA Channel n Status Register	0x0000_0000
DC*n*.PAR	0x0008	R	DMA Channel n Peripheral Address	0x0000_0000
DC*n*.MAR	0x000C	RW	DMA Channel n Memory Address	0x2000_0000

DC*n*.CR DMA Controller Configuration Register

This DMA operation control register is a 32-bit register.

DC0.CR=0x4000_0400 , DC1.CR=0x4000_0410
DC2.CR=0x4000_0420 , DC3.CR=0x4000_0430

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
				TRANSCNT																PERISEL								SIZE		DIR	
0	0	0	0	0x000												0	0	0	0	0				0	0	0	0	00		0	0
RW																RW								RW						RW	

27 16	TRANSCNT	Number of DMA transfer remained Required transfer number should be written before ena ble DMA transfer.
		0 DMA transfer is done.
		N N transfers are remained
11 8	PERISEL	Peripheral selction
		N Associated peripheral selection. Refer to DMA Peripheral connection table
3 2	SIZE	Bus transfer size.
		00 DMA transfer is byte size transfer
		01 DMA transfer is half word size transfer
		10 DMA transfer is word size transfer
		11 Reserved
1	DIR	Select transfer direction.
		0 Transfer direction is from memory to peripheral. (T X)
		1 Transfer direction is from peripheral to memory (R X)

A DMA channel is connected to the selected peripheral. Table 9.3 shows the peripheral selections. This PERISEL field must be configured with the correct number of peripherals that will be connected to the DMA interface.

Table 9.3 DMAC PERISEL Selection

PERISEL[3:0]	Associatec Peipheral
0	CHANNEL IDLE
1	UART0 RX
2	UART0 TX
3	UART1 RX
4	UART1 TX
5	SPI0 RX
6	SPI0 TX
7	ADC0 RX
8	ADC1 RX
9 - 15	N.A.

Note: PERISEL cannot have the same value in different channels. If the same PERISEL value is wriiten in more than one channel, proper operation is not guaranteed. Unused channels must contain CHANNEL IDLE value in PERISEL bit postions.

DC*n*.SR DMA Controller Status Register

The DMA Controller Status Register is an 8-bit register. This register represents the current status of the DMA Controller and enables DMA function.

DC0.SR=0x4000_0404 , DC1.SR=0x4000_0414
DC2.SR=0x4000_0424 , DC3.SR=0x4000_0434

7	6	5	4	3	2	1	0
EOT							DMAEN
1	0	0	0	0	0	0	0
RO							RW

7	EOT	End of transfer.
		0 Data to be transferred is existing. TRANSCNT shows non zero value
		1 All data is transferred. TRANSCNT shows now 0
0	DMAEN	DMA Enable
		0 DMA is in stop or hold state
		1 DMA is running or enabled

DC*n*.PAR DMA Controller Peripheral Address Register

This register represents the peripheral addresses.

DC0.PAR=0x4000_0408 , DC1.PAR=0x4000_0418
DC2.PAR=0x4000_0428 , DC3.PAR=0x4000_0438

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
Peripheral BASE OFFSET																PAR															
0x4000																0x0000															
RO																RW															

31 0	PAR	Target Peripheral address of transmit buffer or receive buffer. User must set exact target peripheral buffer address in this field. If DIR is “0” this address is destination address of data transfer. If DIR is “1”, this address is source address of data transfer.

DC*n*.MAR DMA Controller Memory Address Register

This register represents the memory addresses.

DC0.MAR=0x4000_040C , DC1.MAR=0x4000_041C
DC2.MAR=0x4000_042C , DC3.MAR=0x4000_043C

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
																MAR															
0x2000																0x0000															
RO																RW															

31 0	MAR	Target memory address of data transfer. Address is automatically incremented according to SIZE bits when each transfer is done. If DIR is “0” this address is source address of data tra nsfer. If DIR is “1”, this address is destination address of dat a transfer.

Functional Description

The DMA controller performs direct memory transfer by sharing the system bus with the CPU core. The system bus is shared by 2 AHB masters following the round-robin priority strategy. Therefore, the DMA controller can share half of the system bandwidth.

The DMA controller is triggered only with a peripheral request. When a peripheral requests the transfer to the DMA controller, the associated channel is activated and accesses the bus to transfer the requested data from memory to the peripheral data buffer or from the peripheral data buffer to memory space.

1. User sets the peripheral address and memory address.
2. User configures DMA operation mode and transfer count.
3. User enables the DMA channel.
4. Peripheral sends a DMA request.
5. DMA activates the channel that was requested.
6. DMA reads data from the source address and saves it to the internal buffer.
7. DMA writes the buffered data to the destination address.
8. Transfer count number is decreased by 1.
9. When Transfer count is 0, the EOT flag is set and a notice sent to peripheral to issue the interrupt.
10. DMA does not have an interrupt source; the interrupt related DMA status can be shown from the assigned peripheral interrupt.

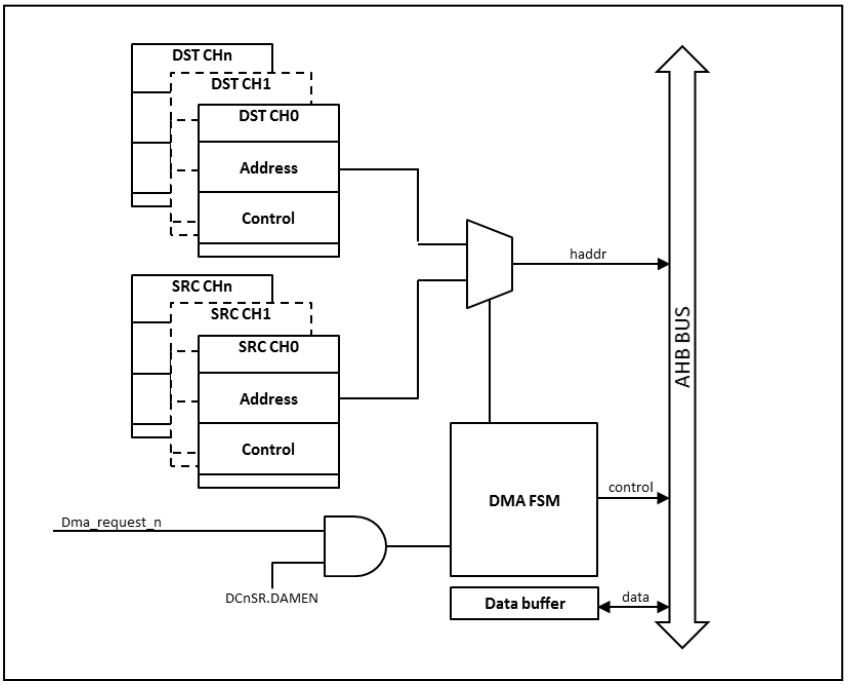


Figure 9.2. Block Diagram

Figure 9.2 shows the functional timing diagram of the DMA controller. The transfer request from the peripheral is pended internally and it invokes source data read transfer on the AHB bus. The read data from the source address is stored in the internal buffer. This data will then be transferred to the destination address when the AHB bus is available.

The timing diagram for a DMA transfer from the peripheral to memory is shown in Figure 9.3. A 4-clock cycle latency exists when accessing the peripheral. If the bus is occupied by a different bus master, the number of bus waiting cycles increase until the bus is available.

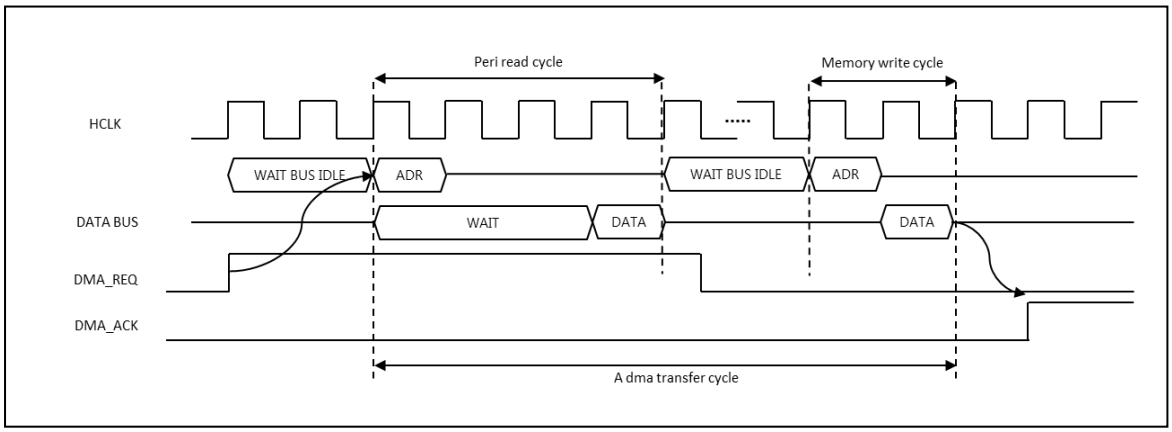


Figure 9.3. DMA Transfer from Peripheral to Memory

The timing diagram for a DMA transfer from memory to the peripheral is shown in Figure 9.4. 4-clock cycle latency exists during accessing the peripheral. If the bus is occupied by a different bus master, there are amount of bus waiting cycles.

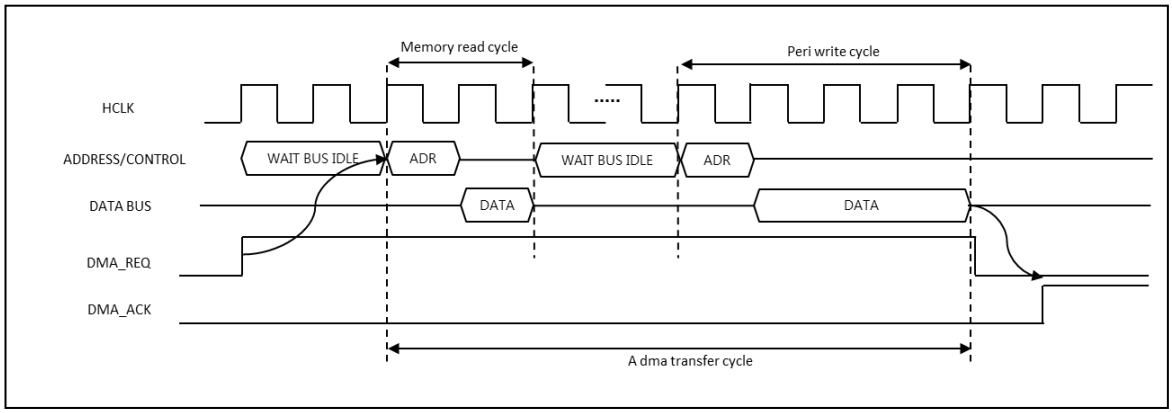


Figure 9.4. DMA Transfer from Memory to Peripheral

The figure is an example N data transfers with the DMA. The DMA transfer is started when DCnSR.DMAEN is set and will be cleared when all the number of transfer is completed.

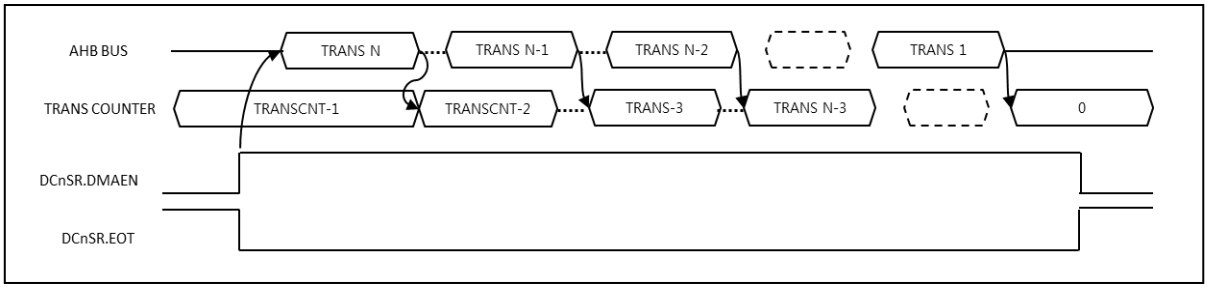


Figure 9.5. N DMA Transfer Example

10. Watchdog Timer

Overview

The Watchdog Timer can monitor the system and generate an interrupt or a reset. It has a 32-bit down-counter. The Miscellaneous Clock Control Register 3 provides base clock options with clock dividers to drive the WDT clock. This can be selected in the WDTCON register. To prevent the WDT from firing, reload the LR register with the appropriate value before the WDT times out. 32-bit down counter (WDTCVR)

Features include:

- Select reset or periodic interrupt
- Count clock selection
- Dedicated pre-scaler
- Watchdog overflow output signal

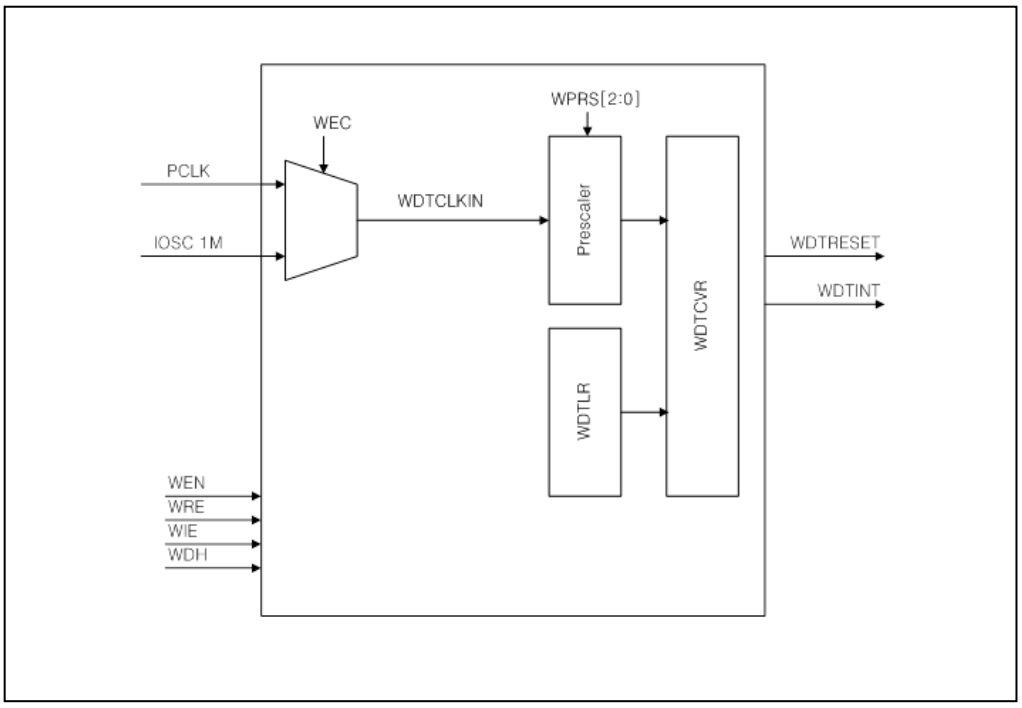


Figure 10.1 Block Diagram

Registers

The base address of the watchdog timer is 0x4000_0200 and the register map is described in Table 10.1. The initial watchdog time-out period is set to 2,000-miliseconds.

Table 10.1 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	0x0000FFFF
WDT.CON	0x0008	RW	WDT Control register	0x0000805C

WDT.LR Watchdog Timer Load Register

The WDTLR register is used to update the WDTCVR register. To update the WDTCVR register, the WEN bit of WDTCON should be set to **1** and written to the WDTLR register with a target value of WDTCVR.

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 WDTCVR value with written value

WDT.CNT Watchdog Timer Current Counter Register

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

WDT.LR=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_FFFF																															
RW																															

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

WDT.CON Watchdog Timer Control Register

WDT module should be configured properly before running. When target purpose is defined, the 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 (RINGOSC 1MHz)
2 0	WPRS	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. When 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 WDTLR value. Before WDT counter goes down to 0, the software can write a certain value to the WDTLR register to reload the WDT counter.

Timing Diagram

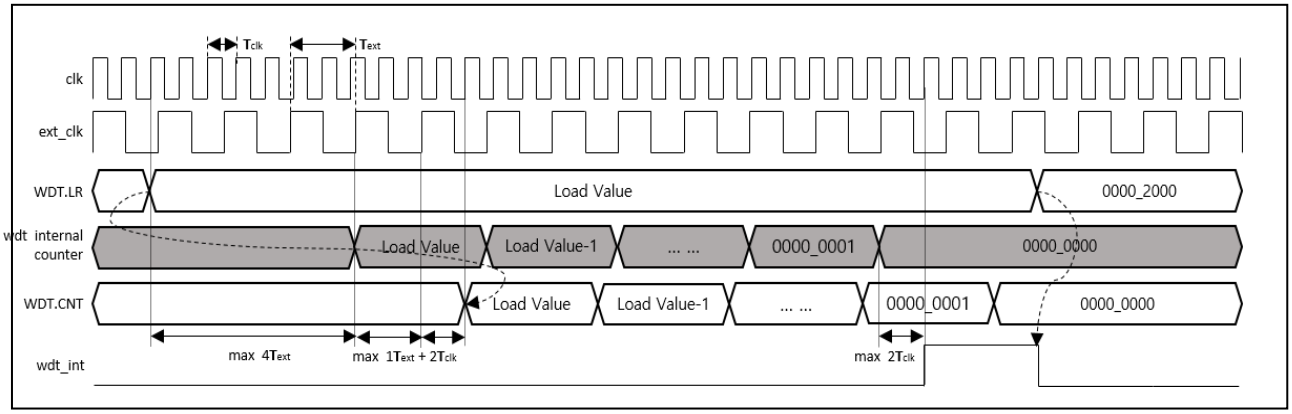


Figure 10.2. Timing Diagram in Interrupt Mode Operation when WDT Clock is the External Clock

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 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 the CNT value data might be delayed by a maximum of 2 system bus clocks in synchronous logic.

Prescale Table

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

The clock sources of watchdog timer can be peripheral clock (PCLK) or one of 3 external clock sources. The external clock source can be enabled by CKSEL (WDT.CON[3]) set to '1'. The external clock source is selected in the MCCR3 register of the System Control Unit block.

To make the WDT counter base clock, users can control the 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 Table 10.2.

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 $5T_{ext}$} + max $4T_{clk}$

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

Table 10.2. Pre-scaled WDT Counter Clock Frequency

Clock Source	WDTCLKIN	WDTCLKIN /4	WDTCLKIN/8	WDTCLKIN/16	WDTCLKIN/32	WDTCLKIN/64	WDTCLKIN/128	WDTCLKIN/256
Ring OSC	1Mhz	250khz	125khz	62.5khz	31.25khz	15.625khz	7.8125khz	3.90625khz
MCLK	MCLK (BUS CLK)	MCLK/4	MCLK/8	MCLK/16	MCLK/32	MCLK/64	MCLK/128	MCLK/256
EOSC	XTAL	XTAL/4	XTAL/8	XTAL/16	XTAL/32	XTAL/64	XTAL/128	XTAL/256

11. 16-Bit Timer

Overview

The timer block consists of six channels of 16-bit general-purpose timers. They support periodic timer, PWM pulse, one-shot timer, and capture mode. The 16-bit timer has the following features:

- 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 11.1 shows the block diagram of a unit timer block.

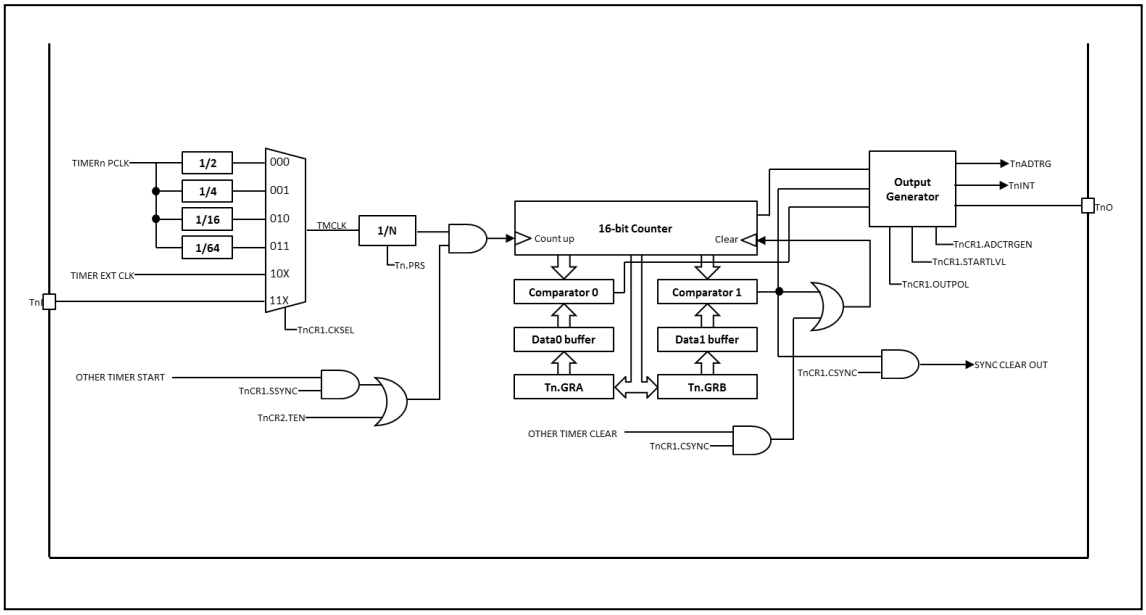


Figure 11.1 Block Diagram

Pin Description

Table 11.1 External Pin

Pin Name	Type	Description
TnIO	I	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 11.3.

Table 11.2 Base Address of Each Channel

Channel	Base Address
T0	0x4000_3000
T1	0x4000_3020
T2	0x4000_3040
T3	0x4000_3060
T8	0x4000_3100
T9	0x4000_3120

Table 11.3 Timer Register Map

Name	Offset	Type	Description	Reset Value
T*n*.CR1	0x--00	RW	Timer control register 1	0x00000000
T*n*.CR2	0x--04	RW	Timer control register 2	0x00000000
T*n*.PRS	0x--08	RW	Timer prescaler register	0x00000000
T*n*.GRA	0x--0C	RW	Timer general data register A	0x00000000
T*n*.GRB	0x--10	RW	Timer general data register B	0x00000000
T*n*.CNT	0x--14	RW	Timer counter register	0x00000000
T*n*.SR	0x--18	RW	Timer status register	0x00000000
T*n*.IER	0x--1C	RW	Timer interrupt enable register	0x00000000

Tn.CR1 Timer n Control Register 1

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

The timer module should be accurately configured prior to operating it. When a target purpose is defined, the timer can be configured in the TnCR1 register.

T0.CR1=0x4000_3000, T1.CR1=0x4000_3020
T2.CR1=0x4000_3040, T3.CR1=0x4000_3060
T8.CR1=0x4000_3100, T9.CR1=0x4000_3120

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSYNC	CSYNC	UAO	OUTPOL				ADCTRGEN	STARTLVL	CKSEL			CLRMOD		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
7	STARTLVL	Timer output polarity control
	0	Default output level is HIGH
	1	Defulat 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	CLRMOD	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	Timer operation mode control
	00	Normal periodic operation mode
	01	PWM mode
	10	One shot mode
	11	Capture mode

Tn.CR2 Timer Control Register 2

The 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
T8.CR2=0x4000_3104, T9.CR2=0x4000_3124

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

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 that the timer is started with TCLR bit setting at **1**.

Tn.PRS Timer n Prescaler Register

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

T0.PRS=0x4000_3008, T1.PRS=0x4000_3028
T2.PRS =0x4000_3048, T3.PRS=0x4000_3068
T8.PRS=0x4000_3108, T9.PRS=0x4000_3128

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 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
T8.GRA=0x4000_310C, T9.GRA=0x4000_312C

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 valu e 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
T8.GRB=0x4000_3110, T9.GRB=0x4000_3130

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 on e-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 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
T8.CNT=0x4000_3114, T9.CNT=0x4000_3134

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

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 timer module.

T0.SR=0x4000_3018, T1.SR=0x4000_3038
T2.SR=0x4000_3058, T3.SR=0x4000_3078
T8.SR=0x4000_3118, T9.SR=0x4000_3138

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

Note: The OVF flag occurs only when the counter rolls from `0xFFFF` to `0`.

Tn.IER Timer 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 TnIER register.

T0.IER=0x4000_301C, T1.IER=0x4000_303C
T2.IER=0x4000_305C, T3.IER=0x4000_307C
T8.IER=0x4000_311C, T9.IER=0x4000_313C

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

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

In Figure 11.2, TMCLK is a reference clock for operation of the timer. When this clock is divided by the prescaler setting, the counting clock will work.

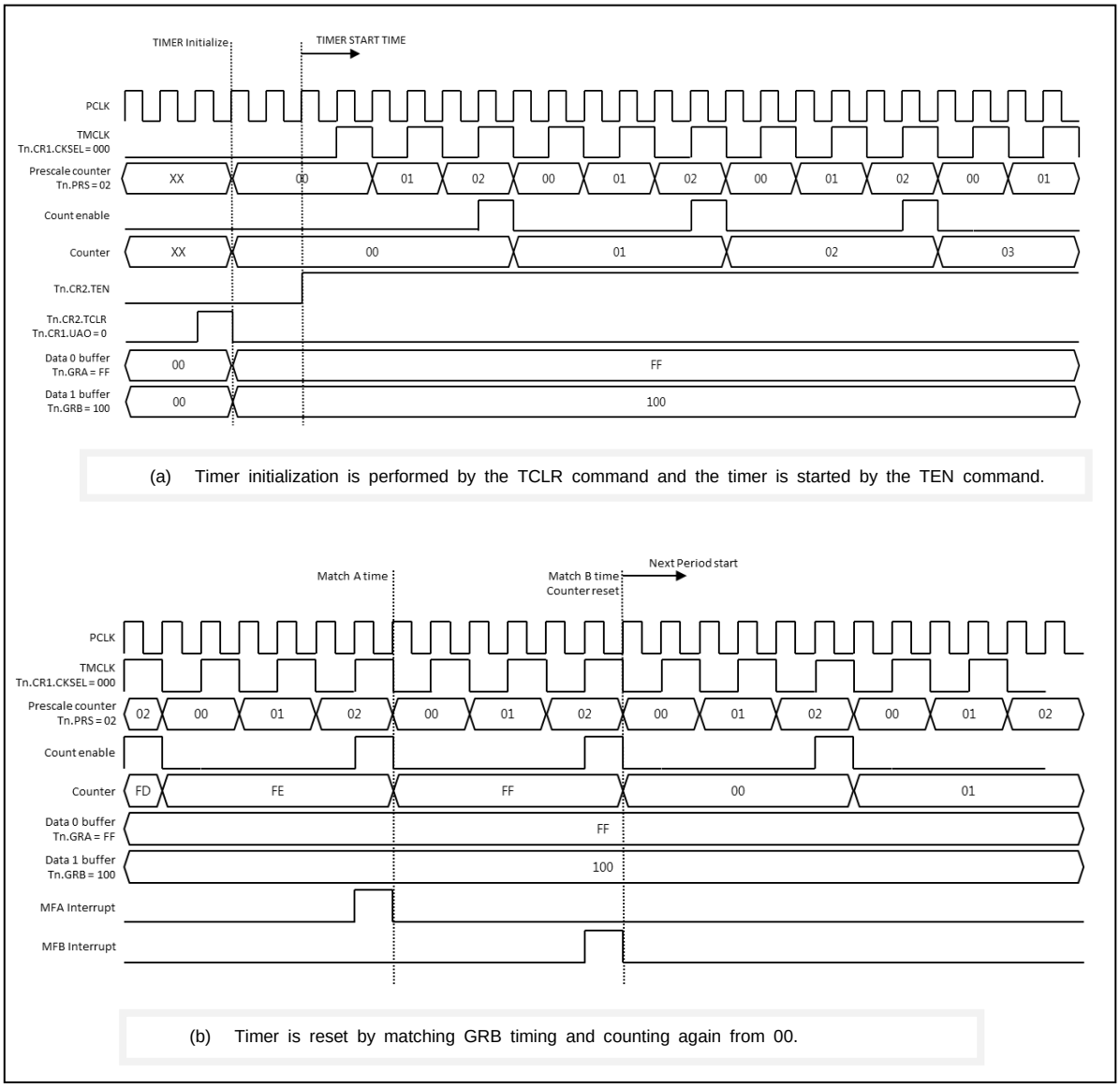


Figure 11.2. Basic Start and Match Operation

The period of timer count can be calculated as shown in 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 values 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 CR2.TCLR command before the CR2.TEN command.

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

Normal Periodic Mode

Figure 11.3 shows the timing diagram in normal periodic mode. Tn.GRB value decides the timer period. One more compare point is provided with Tn.GRA register value.

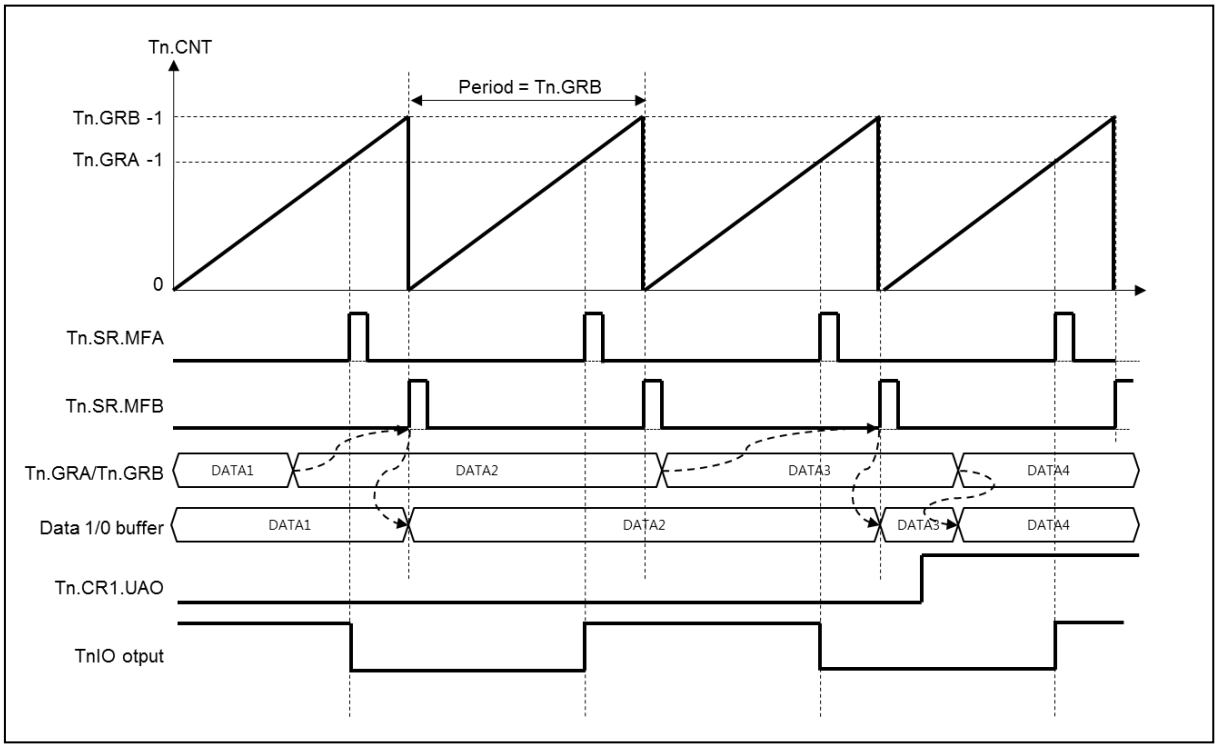


Figure 11.3. Normal Periodic Mode Operation

The period of timer count can be calculated as shown in 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 TnCR2.TEN is “1” because the period is “0”.

The value in Tn.GRA and Tn.GRB is loaded into the internal compare data buffers 0 and 1 when the loading condition occurs. In this periodic mode with TnCR1.UAO =0, the Tn.CR2.TCLR write operation and the GRB match event will load the compare data buffers.

When TnCR1.UAO is 1, the internal compare 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 time. If the value of TnGRA is 0, the TnIO output does not change its previous level. If TnGRA is the same as TnGRB, the TnIO ouput will toggle at same time as the counter start time. The initial level of the TnIO signal is decided by the TnCR1.STARTLVL value.

One Shot Mode

Figure 11.4 shows the timing diagram in one shot mode. Tn.GRB value decides the one shot period. One more compare point is provided with Tn.GRA register value.

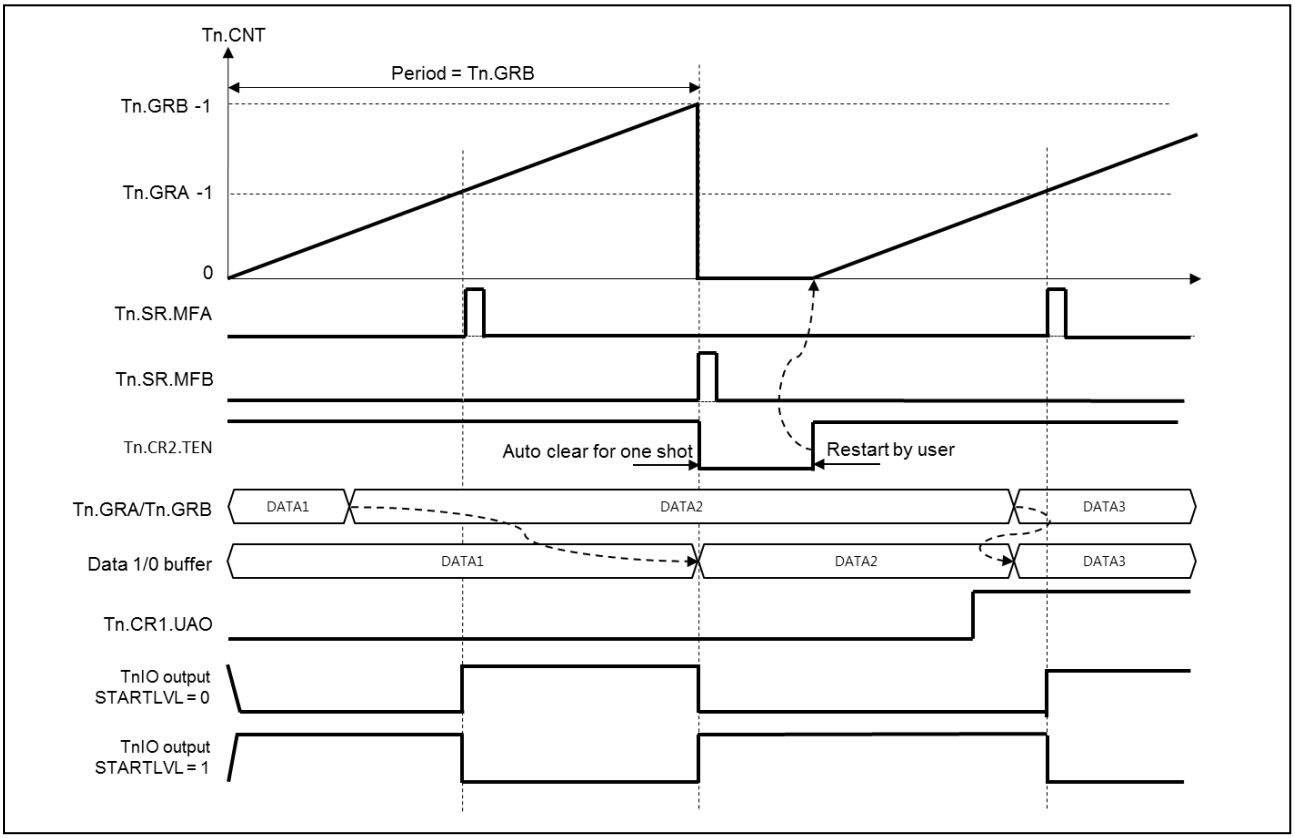


Figure 11.4. One Shot Mode Operation

The period of one shot count can be calculated as shown in 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 TnCR2.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 TnCR1.UAO =0, the Tn.CR2.TCLR write operation and the GRB match event will load the compare data buffers.

When TnCR1.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. Tn.GRB value defines the output pulse period and the Tn.GRA value defines the pulse width of one shot pulse.

PWM Timer Output

Figure 11.5 shows the timing diagram in PWM output mode. The Tn.GRB value decides the PWM pulse period. An additional comparison point is provided by the Tn.GRA register value which defines the pulse width of PWM output.

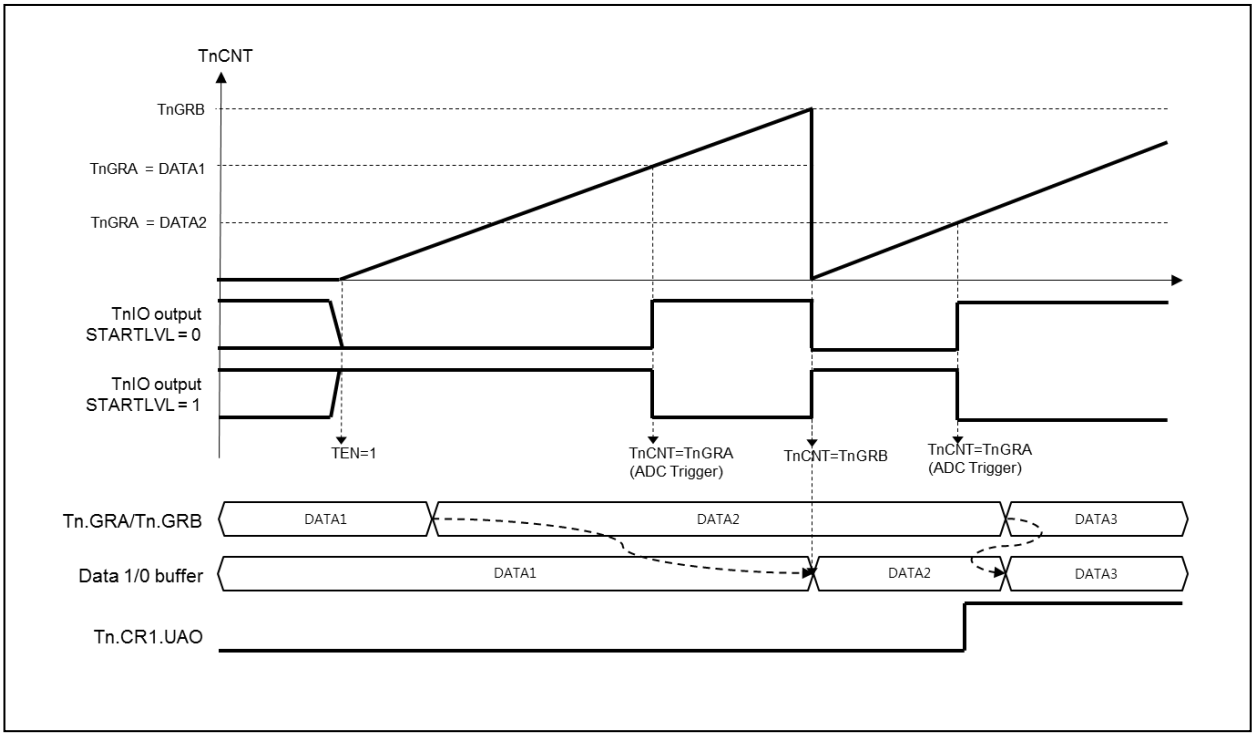


Figure 11.5. PWM Output Operation

The period of PWM pulse can be calculated as shown in 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 TnCR2.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 TnCR1.UAO =0, the Tn.CR2.TCLR write operation and the GRB match event will load the compare data buffers.

When TnCR1.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 the 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

2-PWM outputs are usually used as synchronous PWM signal control. This function is provided with a synchronous start function. Figure 8.6 shows the synchronous PWM generation function.

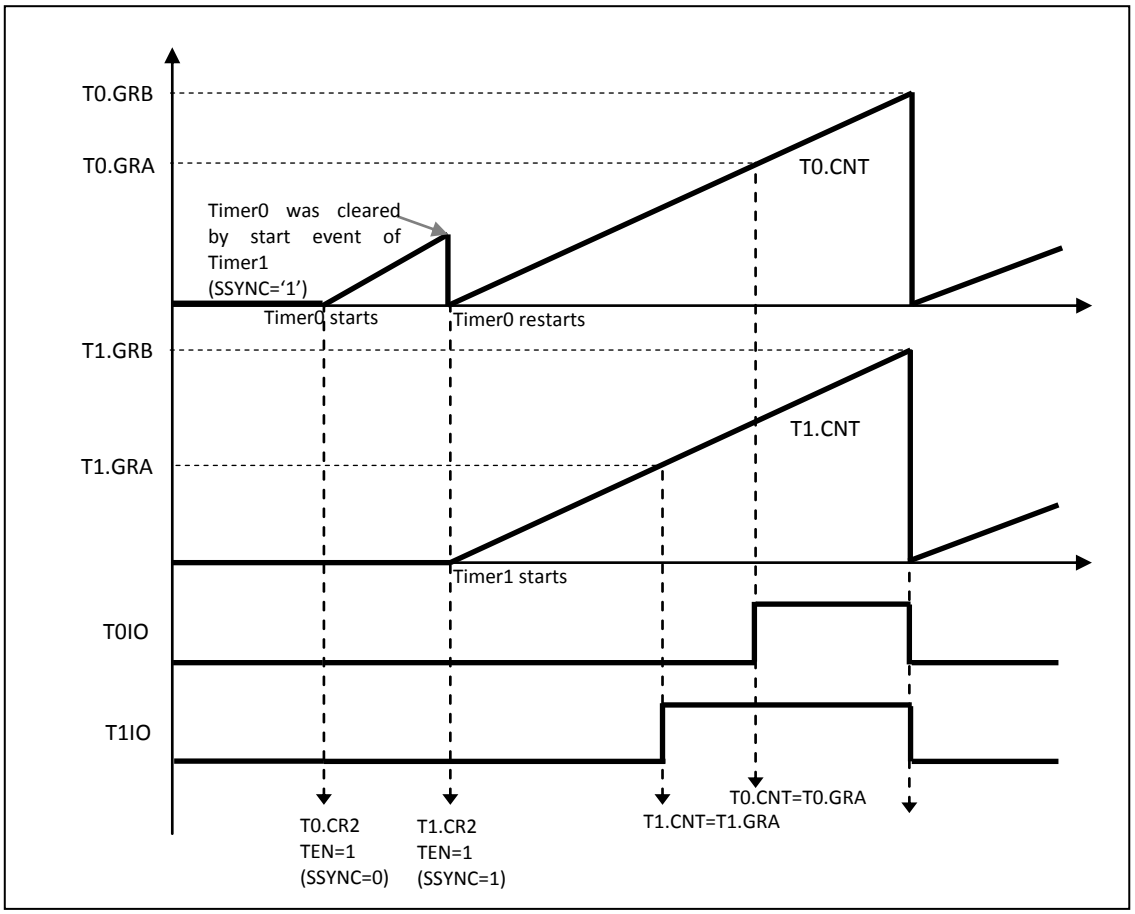


Figure 11.6. Example of Timer Synchronization Function (SSYNC='0')

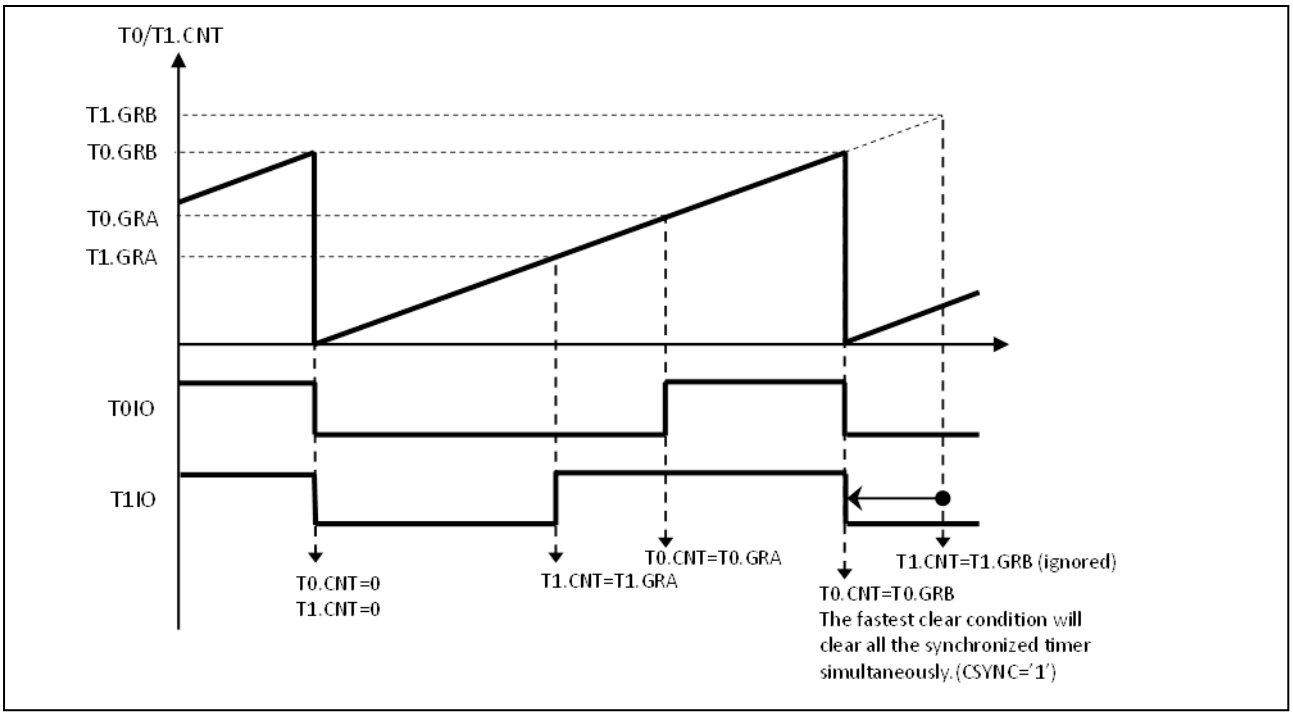


Figure 11.7. Example of Timer Synchronization Function (CSYNC='1')

The TnCR1.SSYNC bit controls start synchronization with other timer blocks. The TnCR1.CSYCN bit controls clear sync with other timer blocks. The SSYNC and CSYNC bits are only effective when used with two or more timers.

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

Capture Mode

Figure 11.8 shows the timing diagram in the Capture mode operation. The TnIO input signal is used for capturing the pulse.Rising and falling edges can capture the counter value in each capture codition.

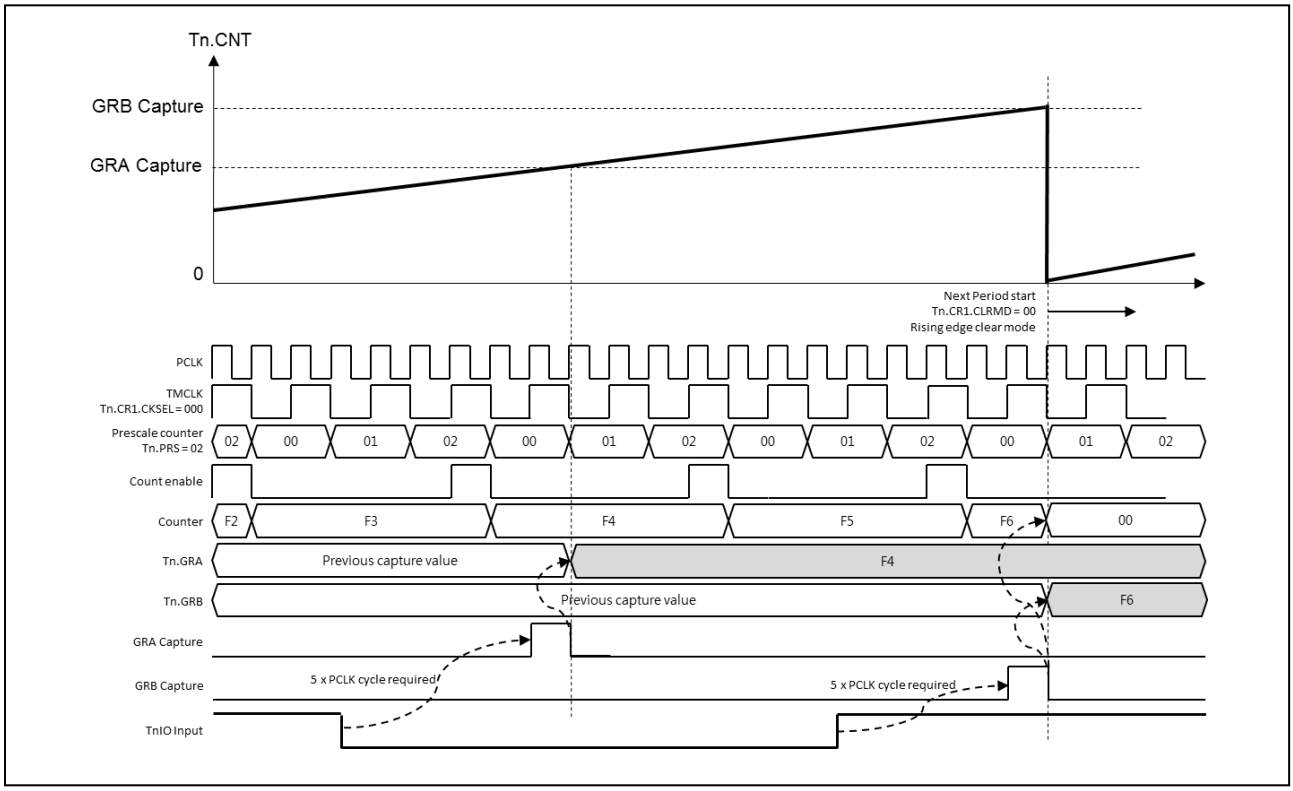


Figure 11.8. Capture Mode Operation

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 multiple modes. The TnCR1.CLRMD field controls the counter clear mode. The following clear modes are supported: Rising edge, Falling edge, Both edges, and None.

The example in Figure 11.8 is 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 performed by the ADC control register.

Figure 11.9 shows the ADC trigger function.

The conversion rate must be shorter than the timer period, else an overrun situation can occur. ADC acknowledge is not required because the trigger signal is automatically cleared after 3 PCLK clock pulses.

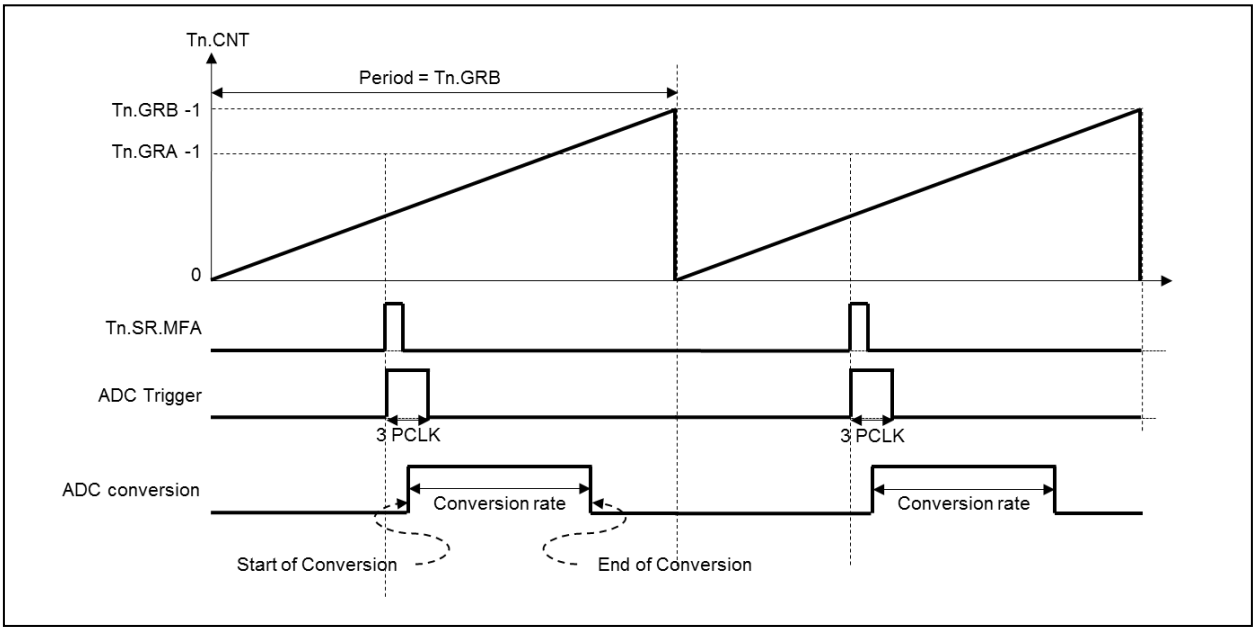


Figure 11.9. ADC Trigger Function Timing Diagram

12. UART

Overview

2-Channel Universal Asynchronous Receiver/Transmitter (UART) modules are included. Dedicated DMA support to transfer data between the Memory buffer and the Transmit/Receive buffer of the UART block is also provided.

The UART operation status, including error status, can be read from the status register. The prescaler, which generates proper baud rate, exists for each UART channel. This prescaler divides the UART clock source which is PCLK/2, from 1 to 65535. The baud rate is generated using the clock with a prescaler of 16, and an 8-bit precision clock tuning function.

Programmable interrupt generation function helps control communication via the UART channel. Features of the UART include:

- Compatible with 16450 UART
- Supports DMA transfer
- Standard asynchronous control bit (start, stop, and parity) configurable
- Programmable 16-bit fractional baud rate 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
- Loop-back control

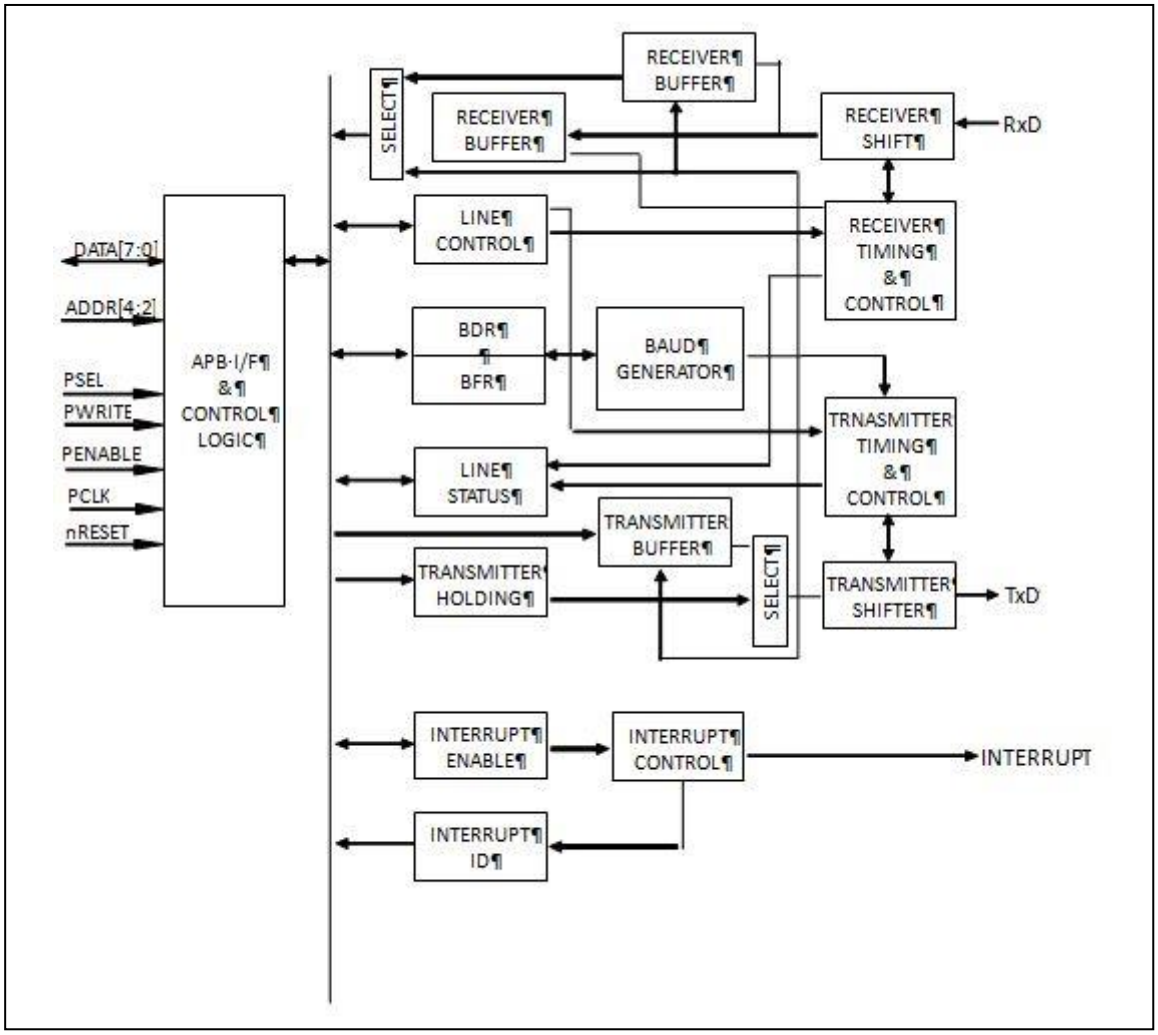


Figure 12.1 Block Diagram

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 shown in Table 12.2 and the register map is described in and Table 12.3.

Table 12.2 Base Address of Each Port

Name	Base Address
UART 0	0x4000_8000
UART 1	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	0x00
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	0x00

Un.RBR Receive Buffer Register

The UART Receive Buffer Register is an 8-bit Read-Only register.

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

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

Bit	Name	Description
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.

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

7	6	5	4	3	2	1	0
THR							
-							
WO							

Bit	Name	Description
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
-	-	DTXIEN	DRXIEN	TXIE	RLSIE	THREIE	DRIE
0	0	0	0	0	0	0	0
		RW	RW		RW	RW	RW

5	DTXIEN	DMA transmit done interrupt enable
		0 Receive line status interrupt is disabled
		1 Receive line status interrupt is enabled
4	DRXIEN	DMA receive done interrupt enable
		0 DMA receive done interrupt is disabled
		1 DMA receive done interrupt is enabled
3	TXIE	Transmit register empty interrupt enable
		0 Transmit register empty interrupt is disabled
		1 Transmit register empty interrupt is enabled
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

7	6	5	4	3	2	1	0
			TXE	IID			IPEN
0	0	0	0	000			0
			R	R			R

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. The Interrupt Source ID register shows one interrupt source which has the highest priority among pending interrupts. This priority is defined in the following order:

- Receive line status interrupt
- Receive data ready interrupt/Character timeout interrupt
- Transmit hold register empty interrupt
- Tx/Rx DMA complete interrupt

Table 12.4 Interrupt ID and Control

Priority	TXE	IID			IPEN	Interrupt Sources		
	Bit4	Bit3	Bit2	Bit1	Bit0	Interrupt	Interrupt Condition	Interrupt Clear
-	0	0	0	0	1	None	-	-
1	0	0	1	1	0	Receiver Line Status	Overrun, Parity, Framing or Break Error	Read LSR register
2	0	0	1	0	0	Receiver Data Available	Receive data is available.	Read receive register or read IIR register
3	0	0	0	1	0	Transmitter Holding Register Empty	Transmit buffer empty	Write transmit hold register or read IIR register
4	1	X	X	X	X	Transmitter Register Empty	Transmit registerr empty	Write transmit hold register or read IIR register
5	0	1	1	0	0	Rx DMA done	Rx DMA completed.	Read IIR register
6	0	1	0	1	0	Tx DMA done	Tx DMA completed.	Read IIR register
7	1	X	X	X	X	Transmitter register Empty and DMA done	Transmitter regiser Empty and Tx DMA completed.	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	
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 is generated according to bit 3, 4, 5 of UnLCR register. The following table shows the variation of parity bit generation.

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.

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		

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

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. No valid stop bit in receive charact ert
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 when the R HR is full
0	DR	Data received
		0 No data in receive holding register.
		1 Data received and saved in receive holding registe r

This register provides the status of data transfers between the transmitter and receiver. Users can get the line status information from this register and handle the next process. Bits 1,2,3, and 4 will cause the line status Interrupt when RLSIE bit in UnIEN register is set. Other bits generate an interrupt when their interrupt enable bit in UnIEN register is set.

Un.BDR Baud Rate Divisor Latch Register

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

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. The baud rate for the baud rate generator is determined using divider values from 1 to 65535 The 16 bit divider register (UnBDR) is written for desired baud rate.The baud rate calculation formula is shown below.

$$\mathrm{BDR} = \frac{UART_{PCLK}}{32 \times BaudRate}$$

For a speed of 48 MHz UART_PCLK, the divider value and error rate is described in table Table 12.5.

Table 12.5 Example of Baud Rate Calculation (without BFR)

UART_PCLK=48 MHz		
Baud Rate	**Divider**	**Error (%)**
1200	1250	0.00%
2400	625	0.00%
4800	312	0.16%
9600	156	0.16%
19200	78	0.16%
38400	39	0.16%
57600	26	0.16%
115200	13	0.16%

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 12.6 Example of Baud Rate Calculation

UART_PCLK=48 MHz			
Baud Rate	**Divider**	**FCNT**	**Error (%)**
1200	1250	0	0.0%
2400	625	0	0.0%
4800	312	128	0.0%
9600	156	64	0.0%
19200	78	32	0.0%
38400	39	16	0.0%
57600	26	10	0.01%
115200	13	5	0.01%

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

The FCNT value can be calculated using the equation above. For example, if the target baud rate is 4800 bps and UART_PCLK is 48 MHz, the BDR value is 312.5. Using the integer 312 as the BDR value and the floating number 0.5, the FNCT value will be 128, as shown in the following calculation:

FCNT = 0.5 * 256 = 128

The 8-bit fractional counter will count up by the BFR value every (baud rate)/16 periods and whenever the fractional counter overflows, the divisor value will increment 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
-					WAITVAL		
0	0	0	0	0	000		
					RW		

2:0	WAITVAL	Wait time is decided by this value $\text{Wait Time} = \frac{WAITVAL}{BAUDRATE}$

Functional Description

General Operation

The UART module is compatible with 16450 UART. Additionally, dedicated DMA channels and fractional baud rate compensation logic are provided. This UART module does not have an internal FIFO block. Therefore, data transfers are established either interactively or with DMA support. The DMA operation is described in this section.

Two DMA channels are provided for each UART module – one channel is for TX transfer and the other one is for RX transfer. Each channel has a 32-bit memory address register and a 16-bit transfer counter register.

Prior to DMA operation, the DMA Memory Address Register and the Transfer Count Register should be configured. For the RX operation, the memory address is the destination memory address and for the TX operation, the memory address is the source memory address.

The transfer counter register stores the number count of transfer data. Each time a single transfer is done, the counter is decremented by 1. When the counter reaches zero, the DMA done flag is delivered to the UART control block. If the interrupt is enabled, this flag generates the interrupt.

Receiver Sampling Timing

The UARTs operates with the following timing.

If the falling edge is on the receive line, the UART determines it to be 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.

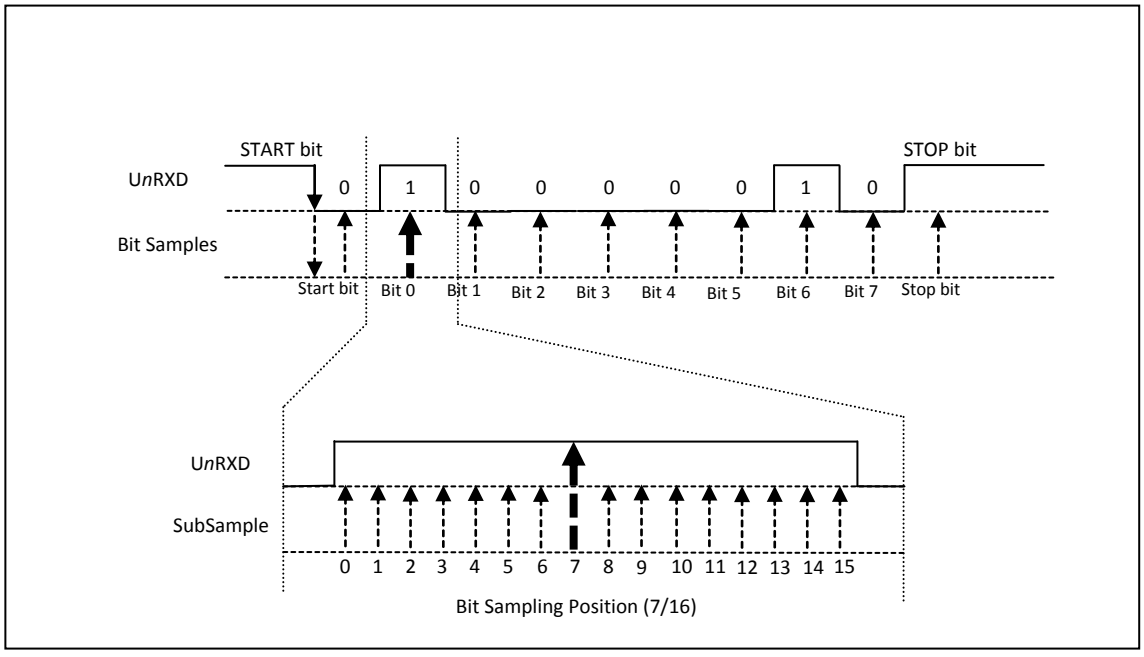


Figure 12.2. Sampling Timing of UART Receiver

Note: Enable the debounce settings in the PCU block to reinforce the immunity of external glitch noise.

Transmitter

The transmitter's function is to transmit data transmit. 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, 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

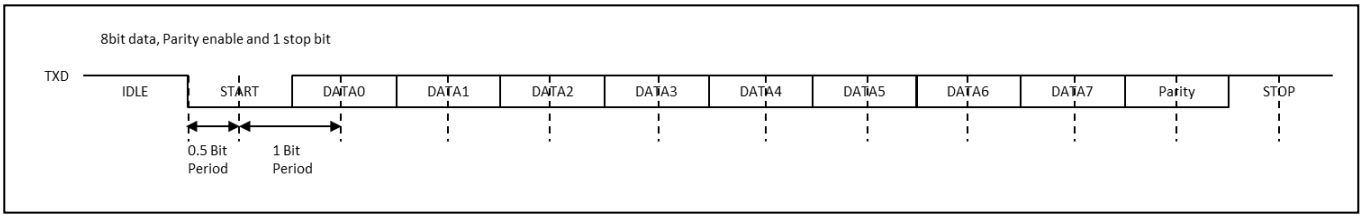


Figure 12.3. Transmit Data Format Example

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 of the Un.IDTR register. When this field is set to 0, zero 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.

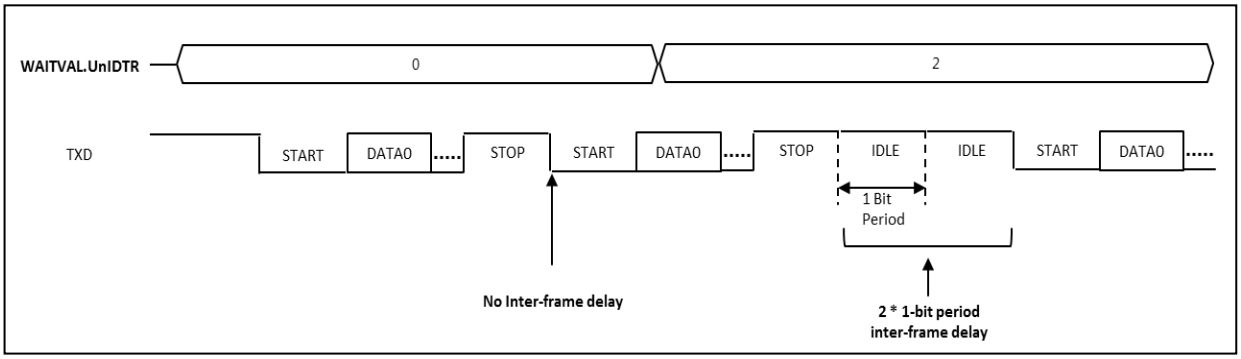


Figure 12.4. Inter-frame Delay Timing Diagram

Transmit Interrupt

The transmit operation generates 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 the interrupt timing that is best for the application.

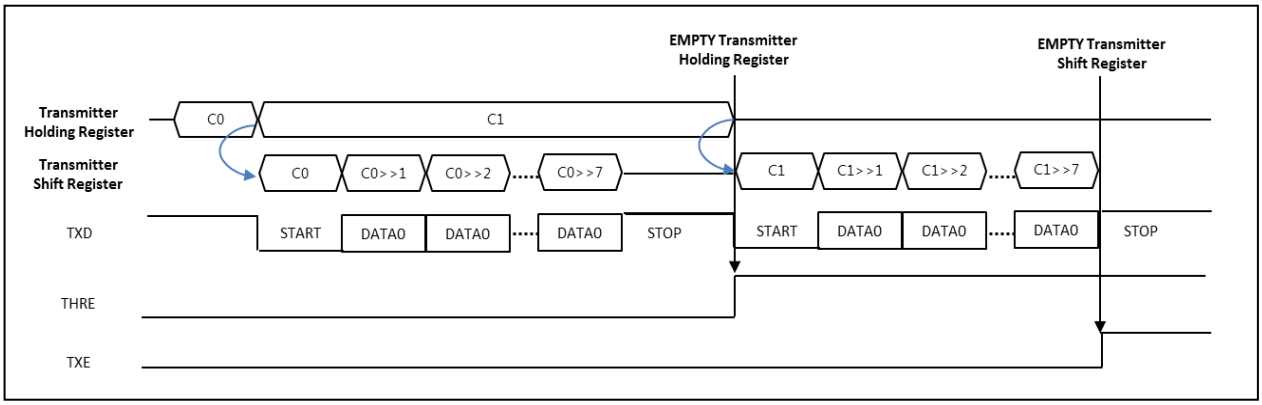


Figure 12.5. Transmit Interrupt Timing Diagram

DMA Transfers

The UART supports the DMA interface function. It is provided as an option, depending on the device. The start memory address for transfer data and the length of transfer data are programmed in the registers in the DMA block.

The end of transfer is notified via the related transfer done flag.

The Transmit with DMA operation invokes the DMA TX done flag DTX.UnIIR and sets the DMA TX done interrupt ID when all the transmit data are written to the transmit holding register.Two transmit data are remain in registers in the UART block after the DMA transfer done interrupt.

The Receive with DMA operation invokes the RXT.UnIIR DMA RX done flag and sets the DMA RX done interrupt ID when all the receive data are written to the destination memory. Therefore, the UART RXD signal is already in IDLE state when the DMA RX done interrupt is issued.

13. Serial Peripheral Interface

Overview

One-channel serial Interface is provided for synchronous serial communications with external peripherals. The Serial Peripheral Interface (SPI) block supports both 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.
- DMA transfer operation.

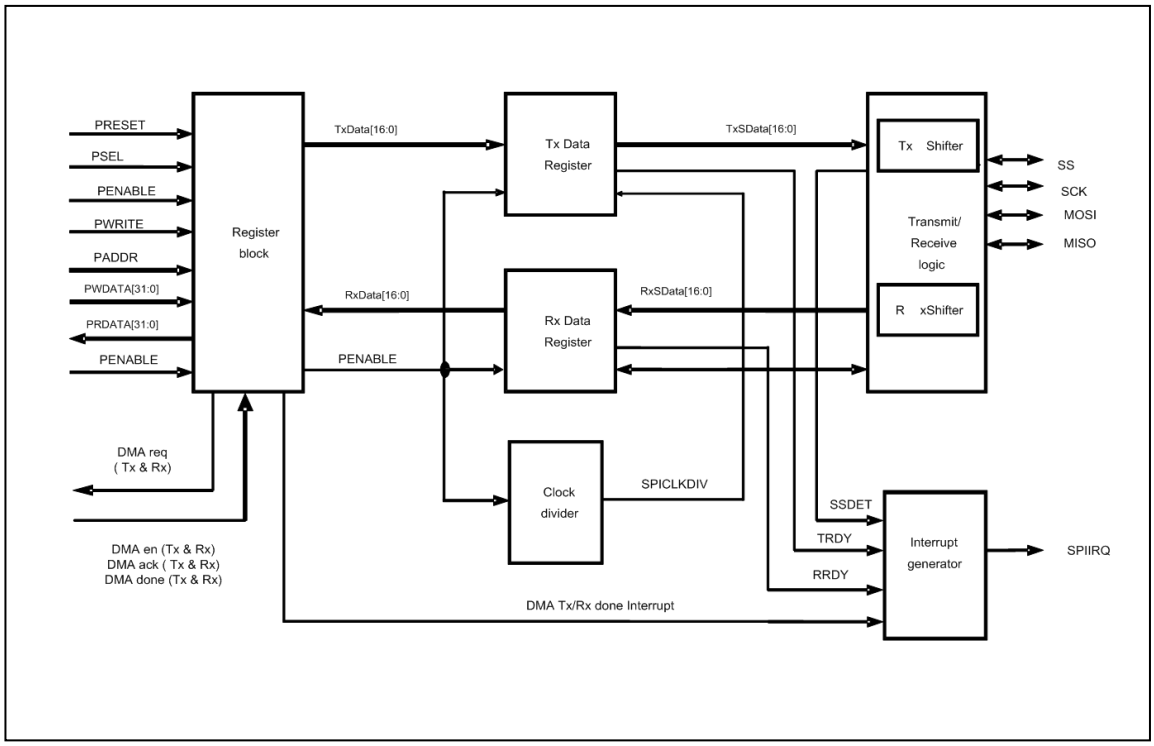


Figure 13.1 SPI Block Diagram

Pin Description

Table 13.1 External Pins

Pin Name	Type	Description
SS0	I/O	SPI0 Slave select input / output
SCK0	I/O	SPI0 Serial clock input / output
MOSI0	I/O	SPI0 Serial data (Master output, Slave input)
MISO0	I/O	SPI0 Serial data (Master input, Slave output)

Registers

The base address of SPI is `0x4000_9000` and the register map is described in Table 13.3.

Table 13.2 SPI Base Address

Name	Base Address
SPI0	0x4000_9000

Table 13.3 SPI Register Map

Name	Offset	Type	Description	Reset Value
SP0.TDR	0x00	W	SPI0 Transmit Data Register	-
SP0.RDR	0x00	R	SPI0 Receive Data Register	0x000000
SP0.CR	0x04	RW	SPI0 Control Register	0x001020
SP0.SR	0x08	RW	SPI0 Status Register	0x000006
SP0.BR	0x0C	RW	SPI0 Baud rate Register	0x0000FF
SP0.EN	0x10	RW	SPI0 Enable register	0x000000
SP0.LR	0x14	RW	SPI0 delay Length Register	0x010101

SP0.TDR SPI Transmit Data Register

SP0.TDR is a 17-bit read/write register. It contains serial transmit data.

SP0.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	TDR	Transmit Data Register	
0			

SP0.RDR SPI Receive Data Register

SP0.RDR is a 17-bit read/write register. It contains serial receive data.

SP0.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		0x00000															
																RW															

16	RDR	Receive Data Register
0		

SP0.CR SPI Control Register

SP0.CR is a 20-bit read/write register which can be set to configure SPI operation mode.

SP0.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	TXDIE	RXDIE	SSCIE	TXIE	RXIE	SSMOD	SSOUT	LBE	SSMASK	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	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
18	TXDIE	DMA Tx Done Interrupt Enable bit.
		0 DMA Tx Done Interrupt is disabled.
		1 DMA Tx Done Interrupt is enabled.
17	RXDIE	DMA Rx Done Interrupt Enable bit.
		0 DMA Rx Done Interrupt is disabled.
		1 DMA Rx Done Interrupt is enabled.
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 (SPnCR[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.
7 6		Reserved	
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

SP0.SR SPI Status Register

SP0.SR is a 10-bit read/write register. It contains the status of the SPI.

SP0.SR=0x4000_9008

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TXDMAF	RXDMAF		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	RC1	RC1	R	R	R

9	TXDMAF	DMA Transmit Operation Complete flag. (DMA to SPI)
	0	DMA Transmit Op is working or is disabled.
	1	DMA Transmit Op is done.
8	RXDMAF	DMA Receive Operation Complete flag. (SPI to DMA)
	0	DMA Receive Operation is working or is disabled.
	1	DMA Transmit Op is done.
7		Reserved
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 SPnRDR.
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 SPnTDR.
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 SPnTDR.
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 SPnRDR.

SP0.BR SPI Baud Rate Register

SP0.BR is a16-bit read/write register. Baud rate is set by writing the register.

SP0.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)

SP0.EN SPI Enable Register

SP0.EN is a bit read/write register. It contains the SPI enable bit.

SP0.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. - SPnSR is initialized by writing “0” to this bit but other registers ar en’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 SPTDR before this b it is active.

Note: When in SPI Slave mode, ensure that you disable the SPI prior to loading the TDR register, then enable it to prevent an extra byte from being sent.

SP0.LR SPI Delay Length Register

SP0.LR is a 24-bit read/write register. It contains start, burst, and stop length value.

SP0.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	StoPLength value
16		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (SPL ≥ 1)
15	BTL	BursTLength value
8		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (BTL ≥ 1)
7	STL	STart Length value
0		0x01 ~ 0xFF : 1 ~ 255 SCLKs. (STL ≥ 1)

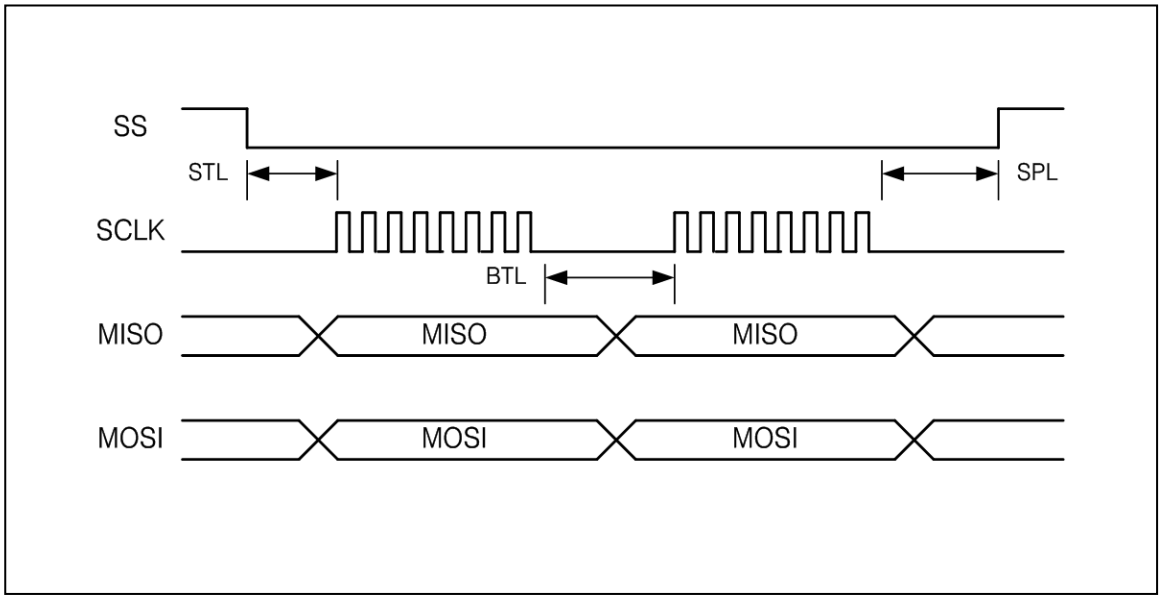


Figure 13.2 SPI Waveform (STL, BTL and SPL)

Functional Description

The SPI Transmit block and Receive block share the Clock Gen Block; however, they are independent of each other. The Transmit and Receive blocks contain 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 (SPnCR). 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 must run in the same mode. This may 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 the two different transfer timings, which are described in detail in the following 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. 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 a 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 (SPnTDR) is output on the MISO line. The actual transfer is started by a software write to the SPnTDR 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 its inactive to its active. 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.

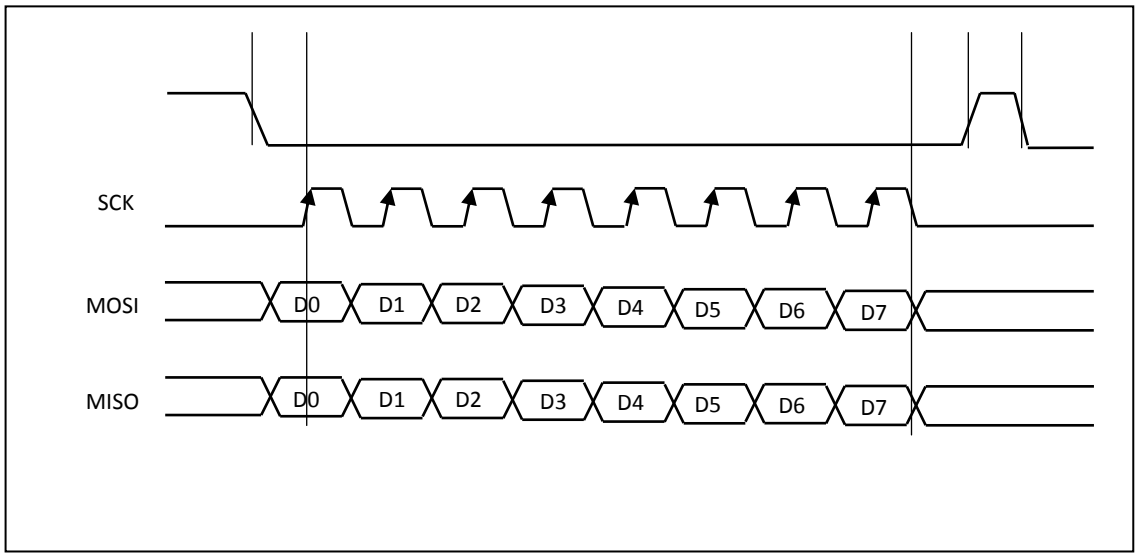


Figure 13.3 Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0)

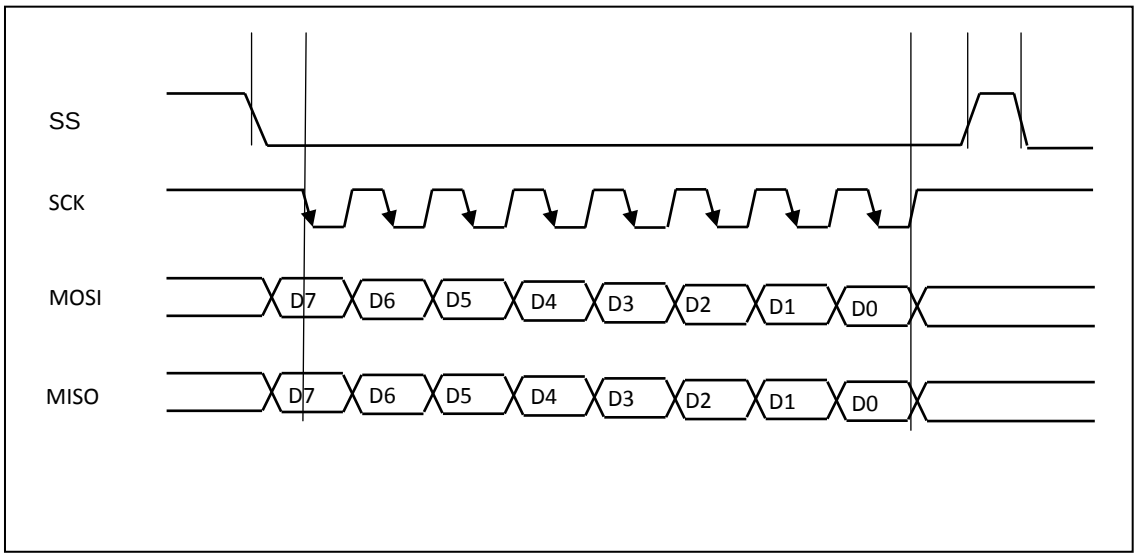


Figure 13.4 SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1)

The timing of a SPI transfer where CPHA is one 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 SPnTDR 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 SPnTDR.

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.

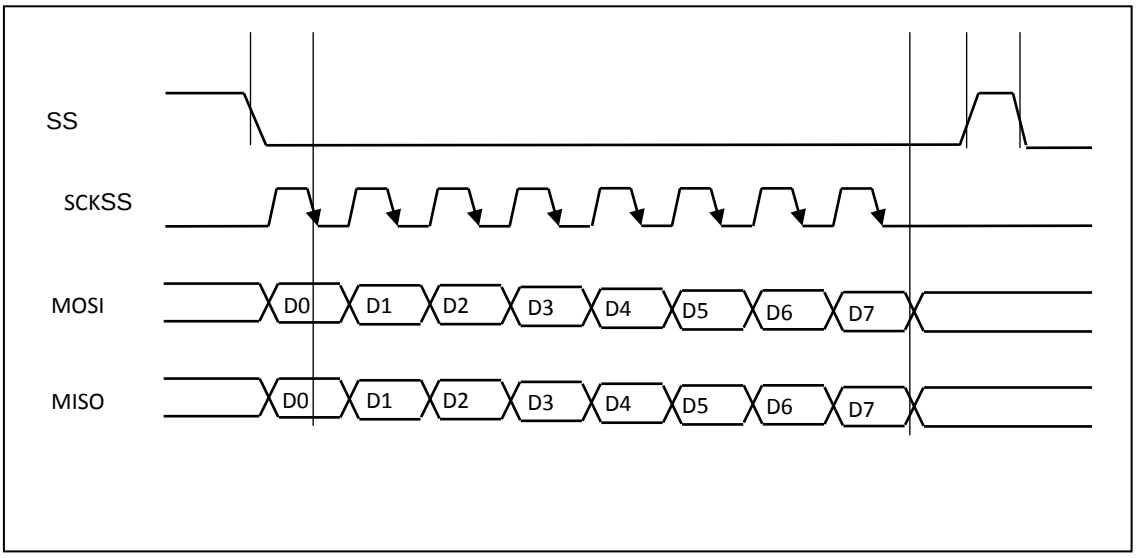


Figure 13.5 SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0)

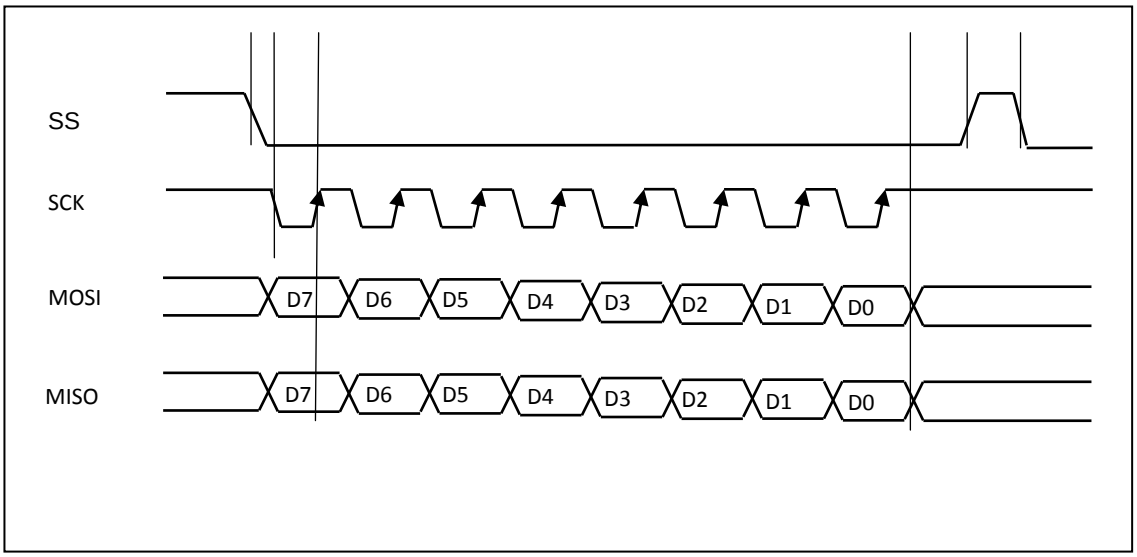


Figure 13.6 SPI transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1)

DMA Handshake

SPI supports the DMA handshaking operation. In order to operate a DMA handshake, DMA registers should first be set. (Refer to Chapter 6, Direct Memory Access Controller). As the transmitter and receiver are independent of each other, SPI can operate the two channels at the same time.

After the DMA channel for the receiver is enabled and the receive buffer is filled, SPI sends an Rx request to the DMA to empty the buffer and waits for an ACK signal from DMA. If the Receive buffer is filled again after the ACK signal, SPI sends an Rx request. If DMA Rx DONE becomes high, RXDMAF (SPnSR[8]) becomes **1** and an interrupt is serviced when RXDIE (SPnCR[17]) is set.

Similarly, if the transmit buffer is empty after the DMA channel for the transmitter is enabled, SPI sends a Tx request to the DMA to fill the buffer and waits for an ACK signal from DMA. If the transmit buffer is empty again after the ACK signal, SPI sends a Tx request. If DMA Tx DONE becomes high, TXDMAF(SPnSR[9]) becomes **1** and an interrupt is serviced when TXDIE(SPnCR[18]) is set.

The slave transmitter sends dummy data at the first transfer (8~17 SCLKs) in DMA handshake mode.

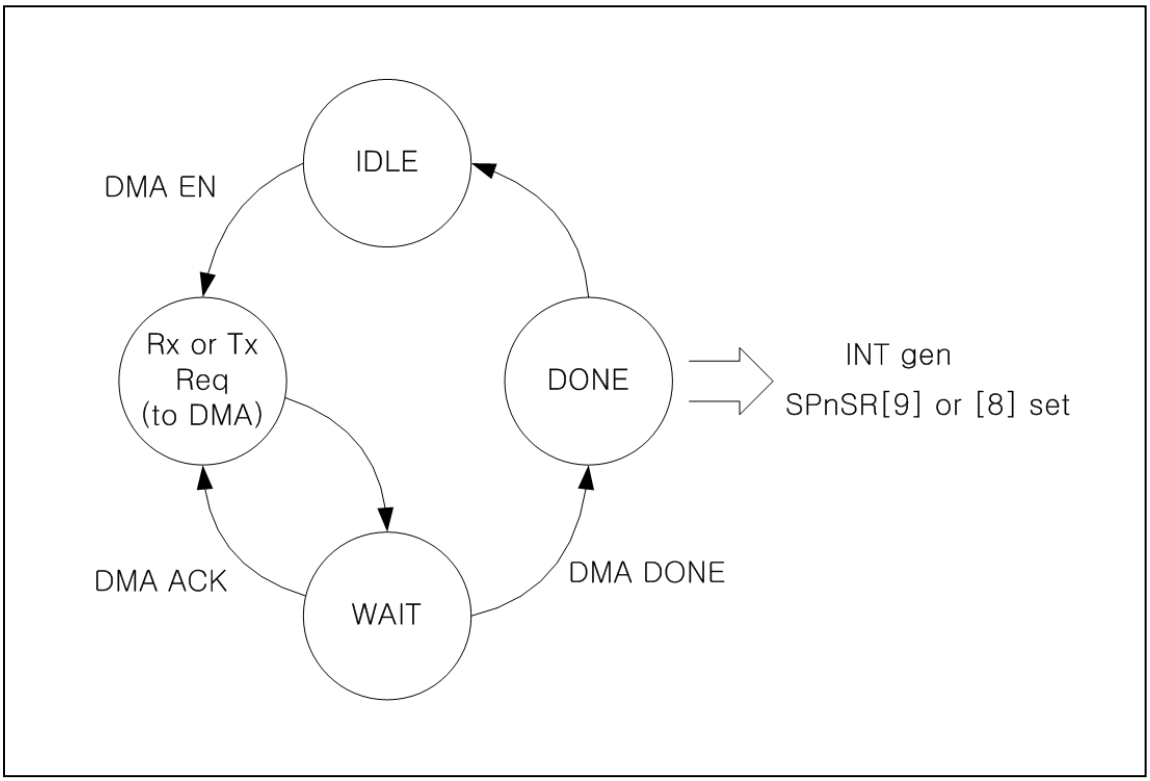


Figure 13.7 DMA Handshake Flow Chart

14. I²C Interface

Overview

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.

- 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

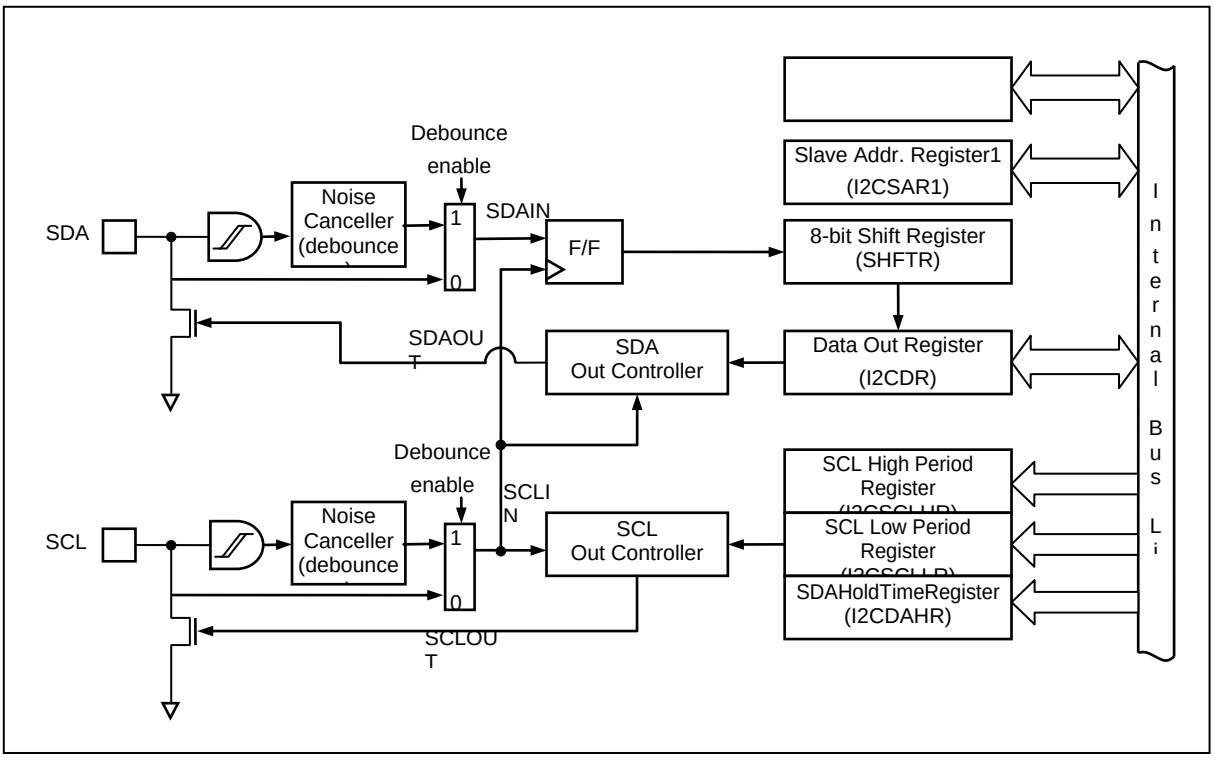


Figure 14.1 I²C Block Diagram

Pin Description

Table 14.1 I²C Interface External Pins

Pin Name	Type	Description
SCL0	I/O	I²C channel 0 Serial clock bus line (open-drain)
SDA0	I/O	I²C channel 0 Serial data bus line (open-drain)

Registers

The base address of I²C0 is `0x4000_A000`. The register map is described in Table 14.3.

Table 14.2 I²C Interface Base Address

Name	Base Address
I²C0	0x4000_A000

Table 14.3 I²C Register Map

Name	Offset	Type	Description	Reset Value
IC*n*.DR	0x00	RW	I²C0 Data Register	0xFF
IC*n*.SR	0x08	R, RW	I²C0 Status Register	0x00
IC*n*.SAR	0x0C	RW	I²C0 Slave Address Register	0x00
IC*n*.CR	0x14	RW	I²C0 Control Register	0x00
IC*n*.SCLL	0x18	RW	I²C0 SCL LOW duration Register	0xFFFF
IC*n*.SCLH	0x1C	RW	I²C0 SCL HIGH duration Register	0xFFFF
IC*n*.SDH	0x20	RW	I²C0 SDA Hold Register	0x7F

ICn.DR I²C Data Register

ICn.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.

IC0.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.

ICn.SR I²C Status Register

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

IC0.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.

ICn.SAR I²C Slave Address Register

ICn.SAR is an 8-bit read/write register. It shows the address in Slave Mode.

IC0.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.

ICn.CR I²C Control Register

ICn.CR is an 8-bits read/write register. The register can be set to configure I2C operation mode and simultaneously allowed for I2C transactions to be kicked off.

IC0.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.

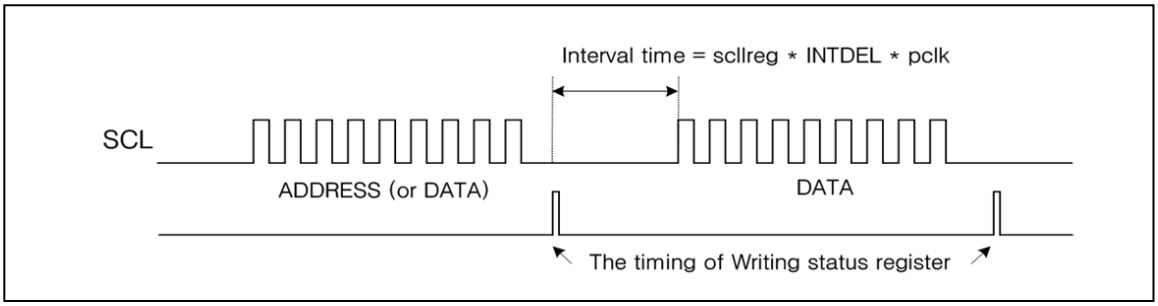


Figure 14.2 INTDEL in Master Mode

ICn.SCLL I²C SCL LOW Duration Register

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

IC0.SDLL=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.

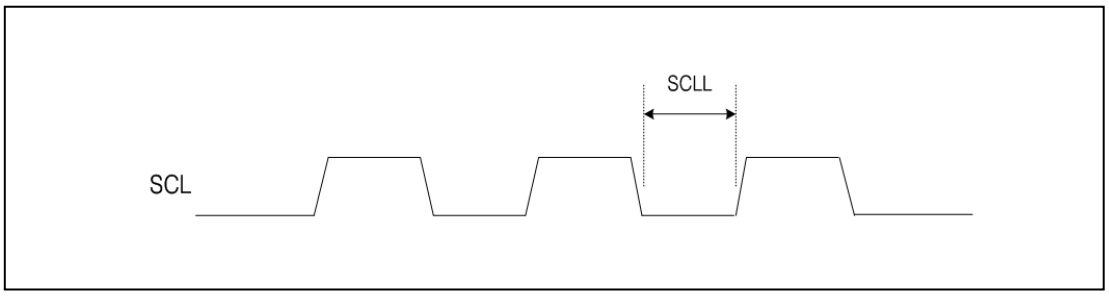


Figure 14.3 SCL LOW Timing

ICn.SCLH I²C SCL HIGH Duration Register

ICnSCLH is a 16-bit read/write register. SCL HIGH time will be set by writing this register in Master Mode.

IC0.SDLH=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.

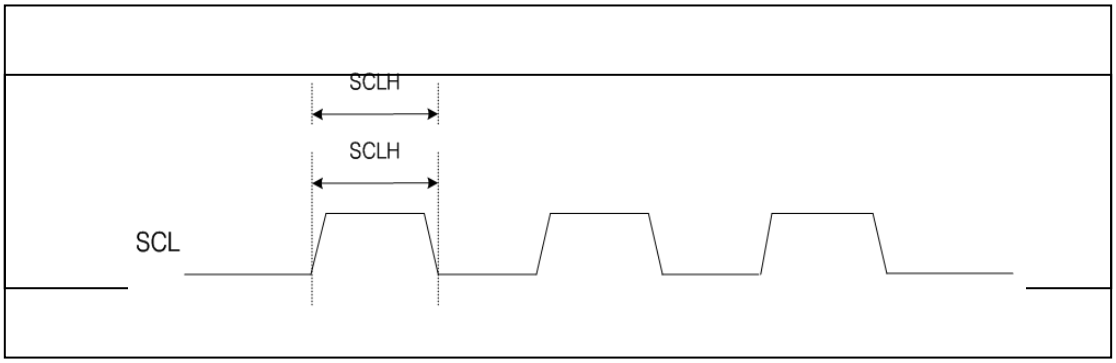


Figure 14.4 SCL LOW Timing

ICn.SDH SDA Hold Register

ICnSDH is a 15-bit read/write register. SDA HOLD time is set by writing this register in Master Mode.

IC0.SDH=0x4000_A020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDH														
	0x3FFF														
	RW														

14	SDH	SDA HOLD time setting value. SDH = (PCLK * SDH[14:0]) + 4 PCLKs
0		Default value is 0x3FFF.

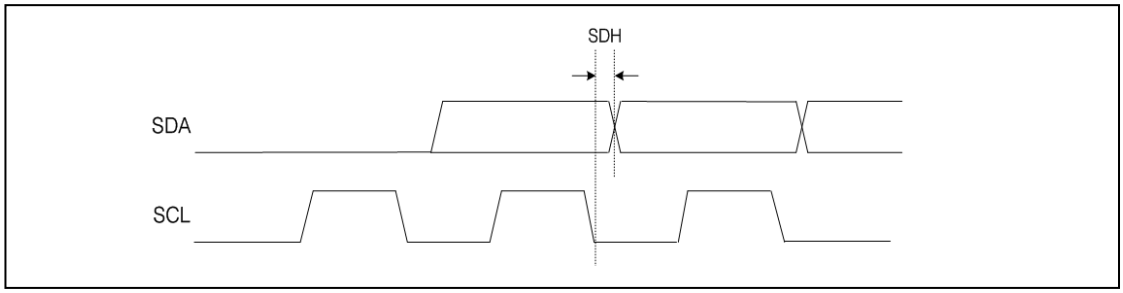


Figure 14.5 SDA HOLD Timing

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.

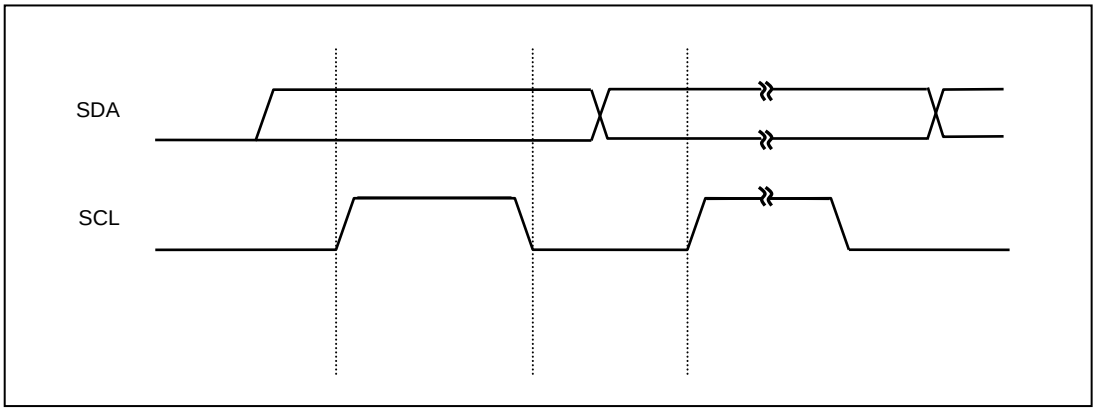


Figure 14.6 I²C Bus Bit Transfer

START/Repeated START/STOP

Within the procedure of the I^2C-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.

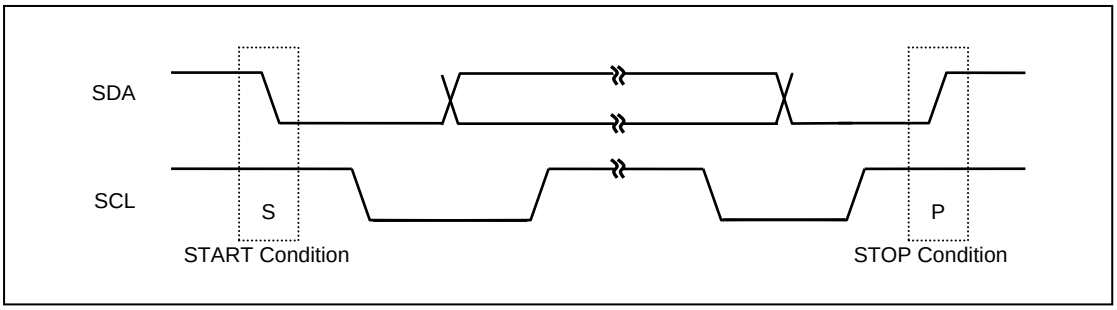


Figure 14.7 START and STOP Condition

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 must 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 acknowledgement is generated.

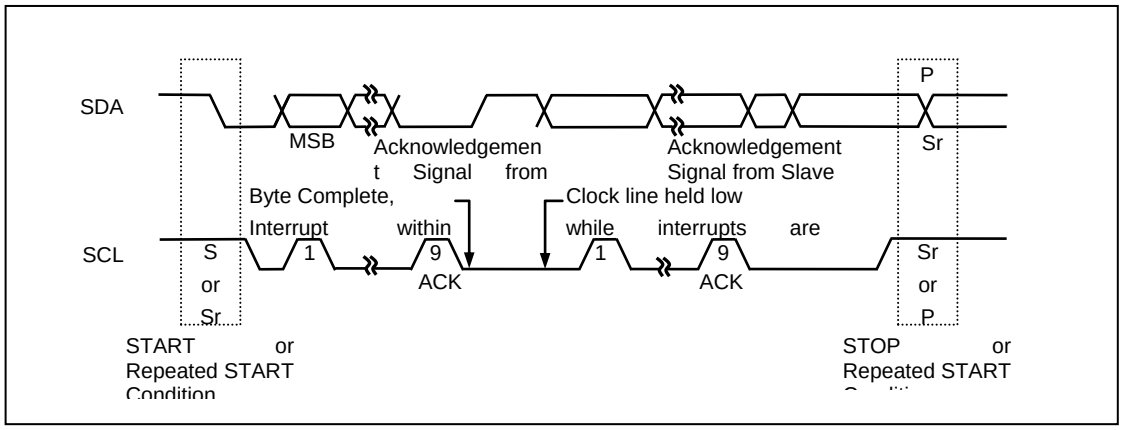


Figure 14.8 I²C Bus Data Transfer

Acknowledge

Data transfer with acknowledgement 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’s unable to receive or transmit because it’s 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 acknowledges the slave address but cannot receive any more data bytes later during the transfer, 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.

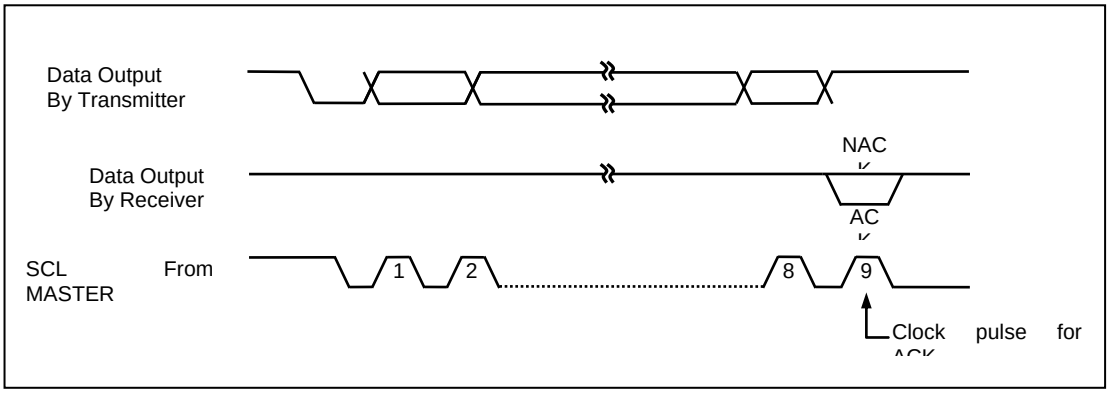


Figure 14.9 I²C Bus Acknowledgement

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. A defined clock is therefore needed 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 concerned 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”.

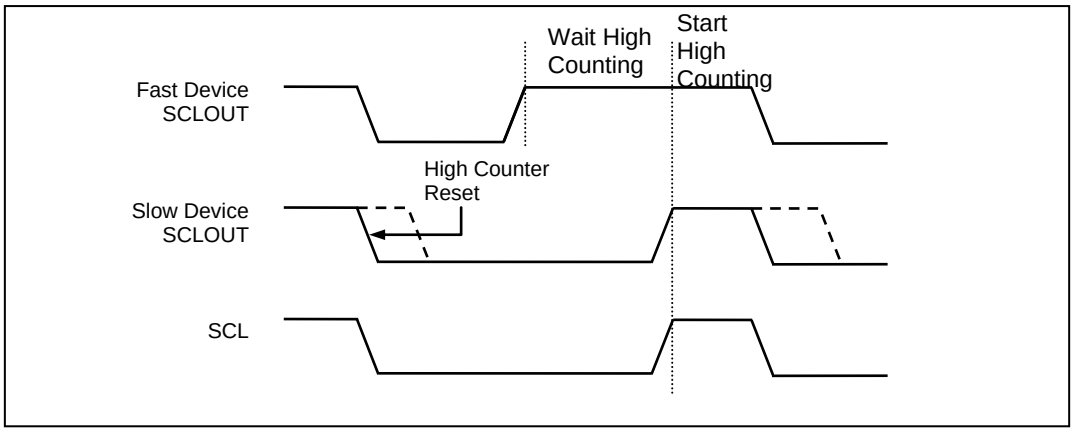


Figure 14.10 Clock Synchronization During the Arbitration Procedure

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 Dataout 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.

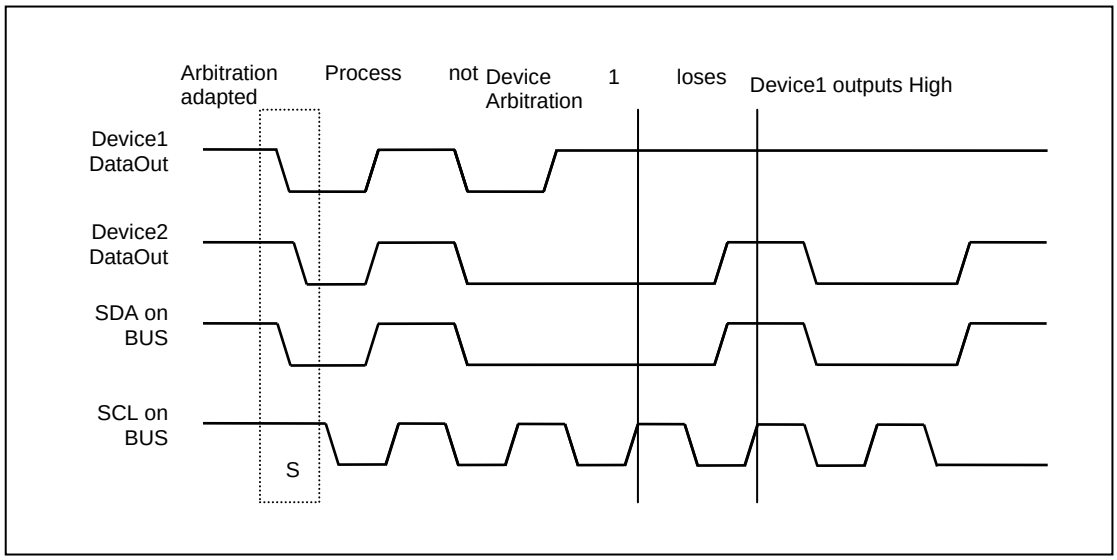


Figure 14.11 Arbitration Procedure Between Two Masters

I²C Operation

I²C supports the interrupt operation. After interrupt is serviced, IIF(ICnSR[10]) flag is set. ICnSR shows I²C-bus status information and SCL line stays “L” before the register is written as a certain value. The status register can be cleared by writing a zero.

Master Transmitter

The master transmitter shows the flow of the transmitter in Master Mode (see Figure 14.12).

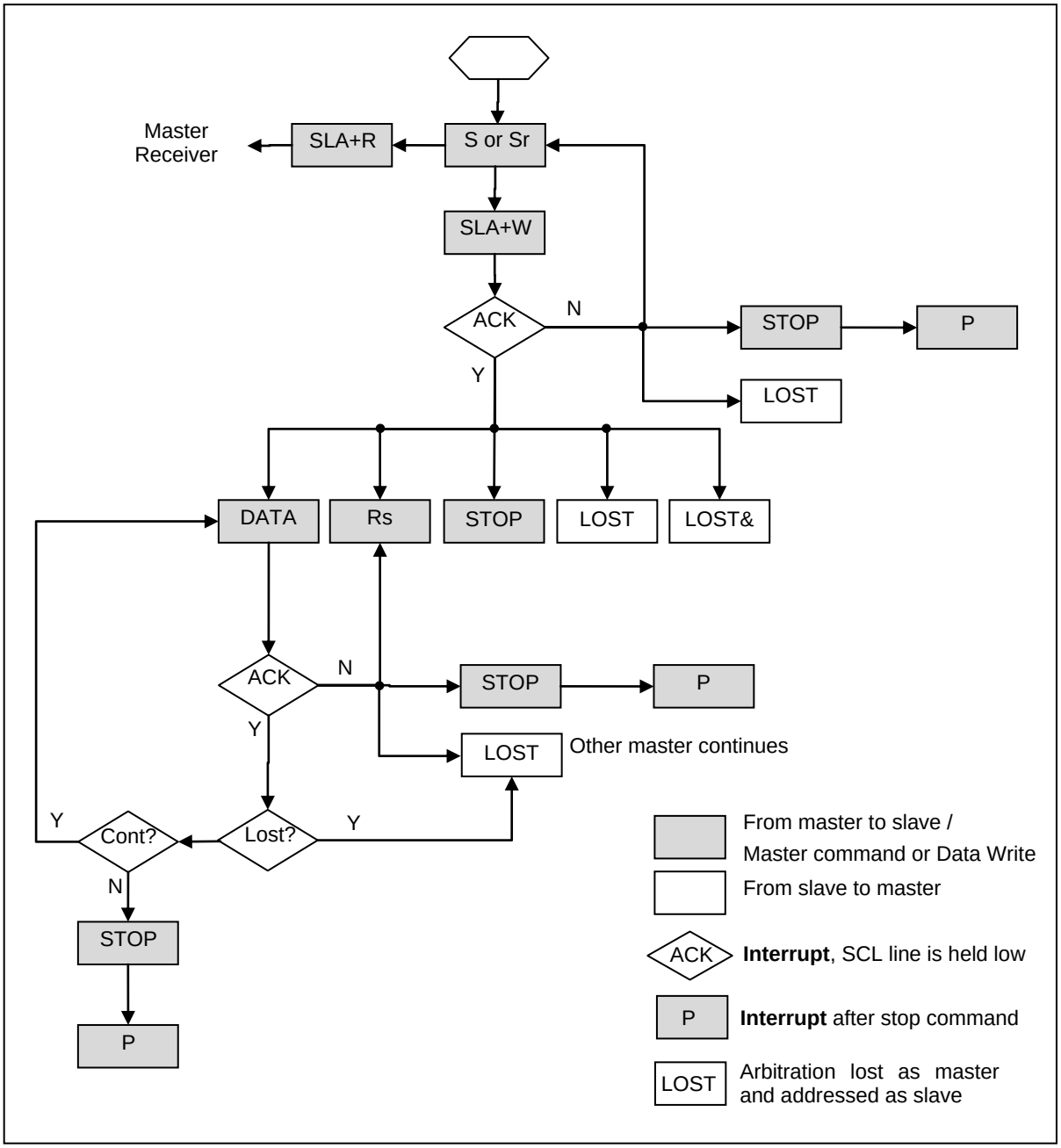


Figure 14.12 Transmitter Flowchart in Master Mode

Master Receiver

The master receiver shows the flow of the receiver in Master Mode (see Figure 14.13).

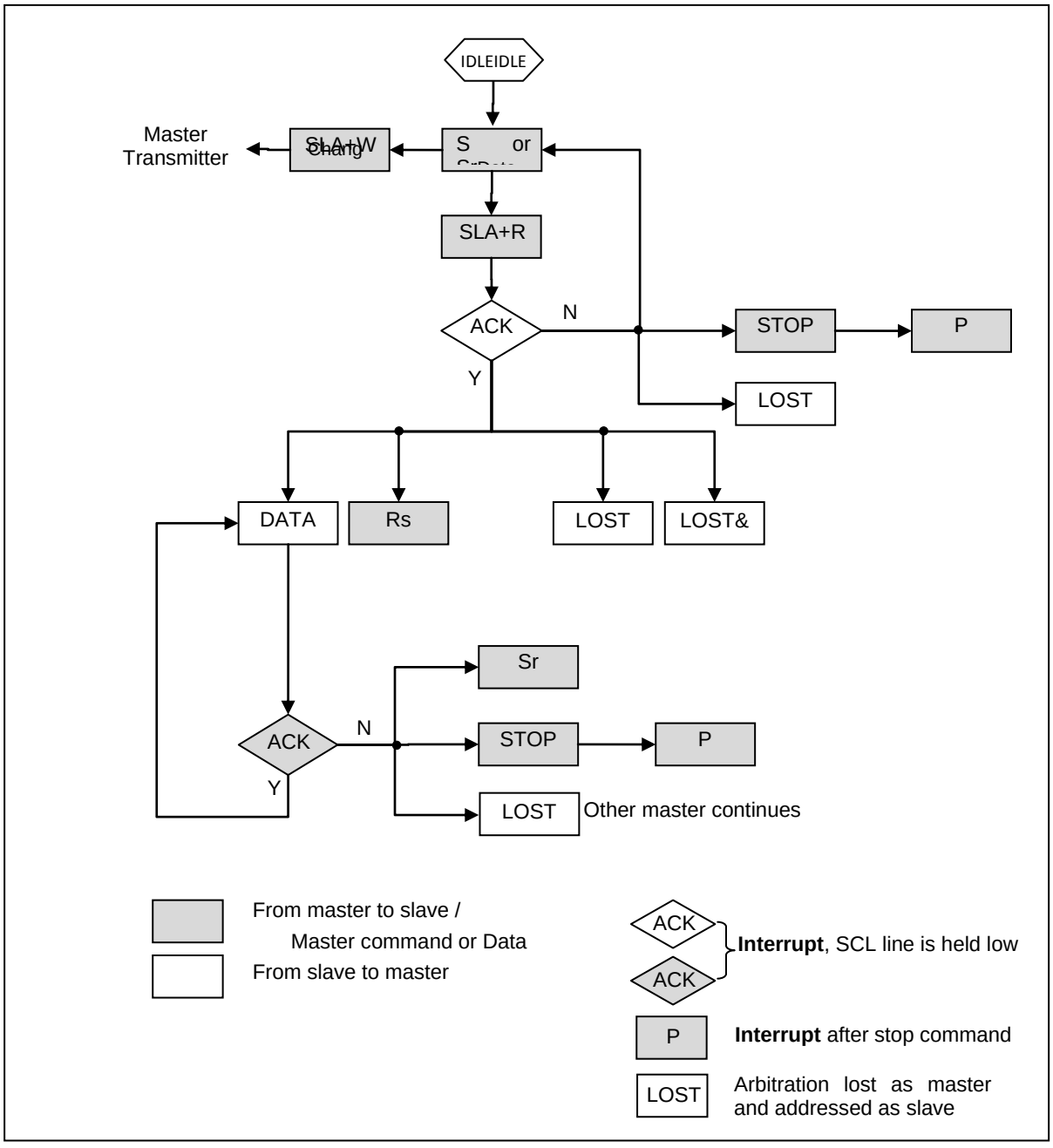


Figure 14.13 Receiver Flowchart in Master Mode

Slave Transmitter

The slave transmitter shows the flow of the transmitter in Slave Mode (see Figure 14.14).

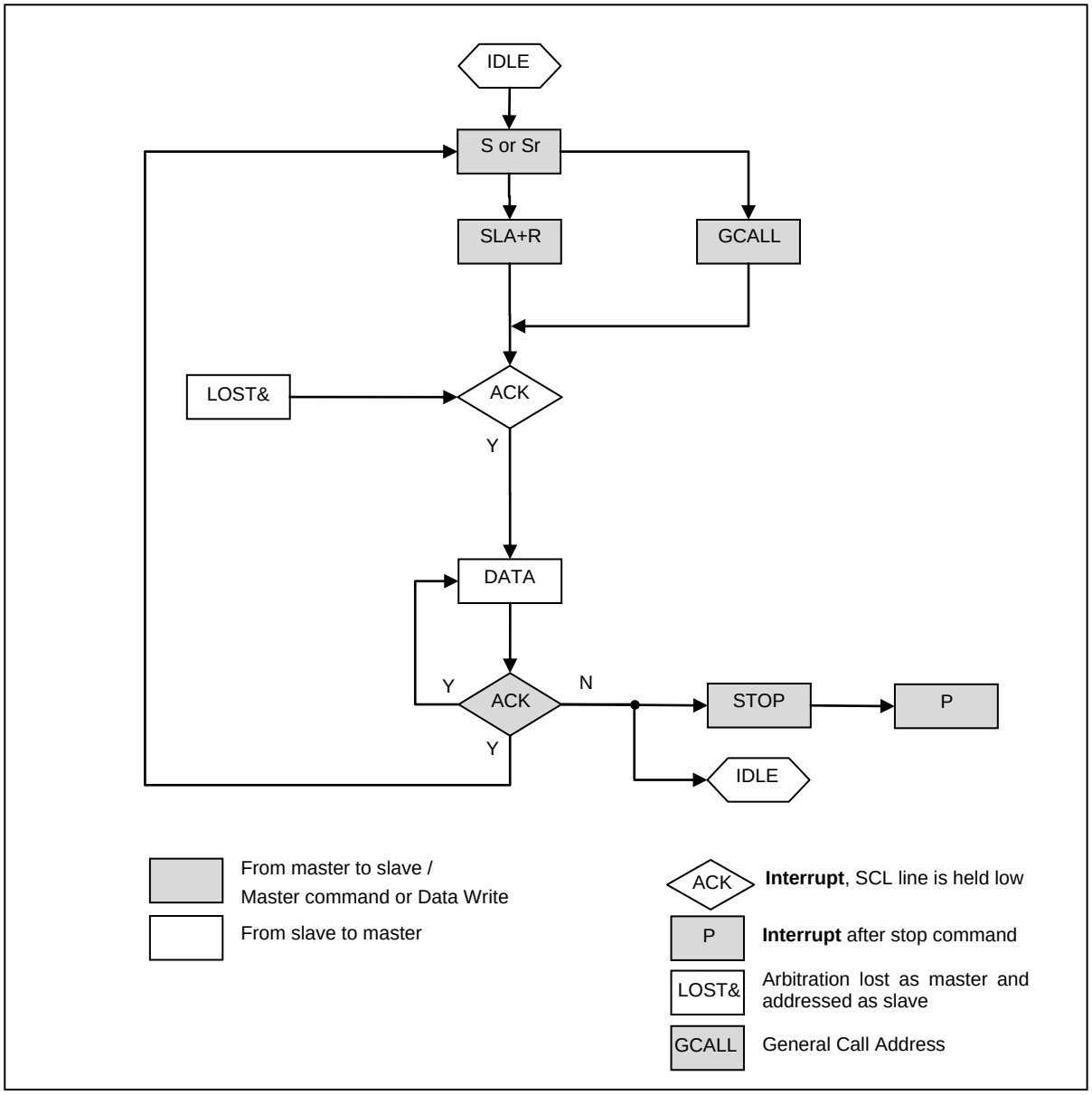


Figure 14.14 Transmitter Flowchart in Slave Mode

Slave Receiver

The slave receiver shows the flow of the receiver in Slave Mode (see Figure 14.15).

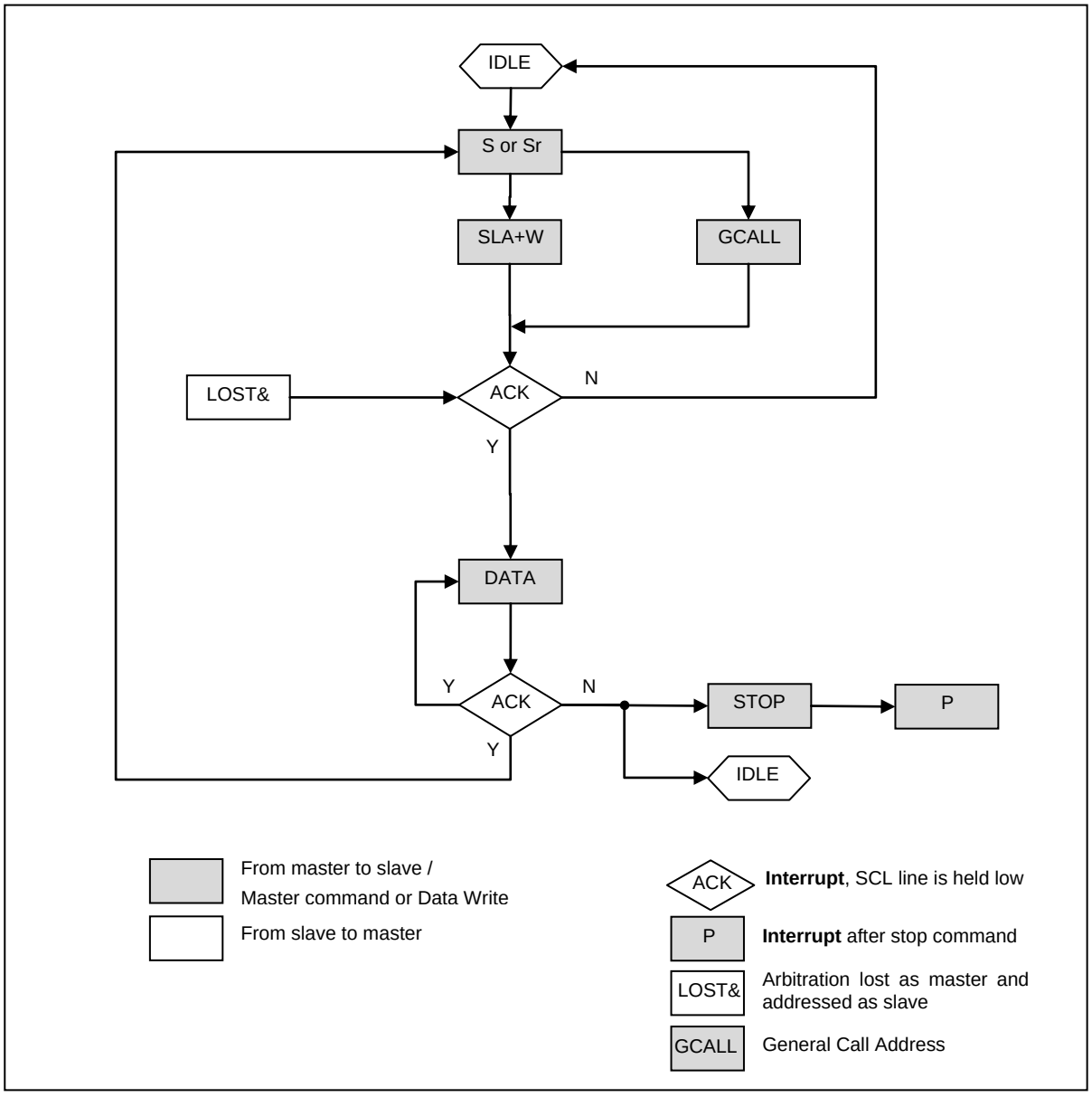


Figure 14.15 Receiver Flowchart in Slave Mode

15. Motor Pulse Width Modulator

Introduction

The Motor Pulse Width Modulator (MPWM) is a 16-bit programmable motor controller with the following features:

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

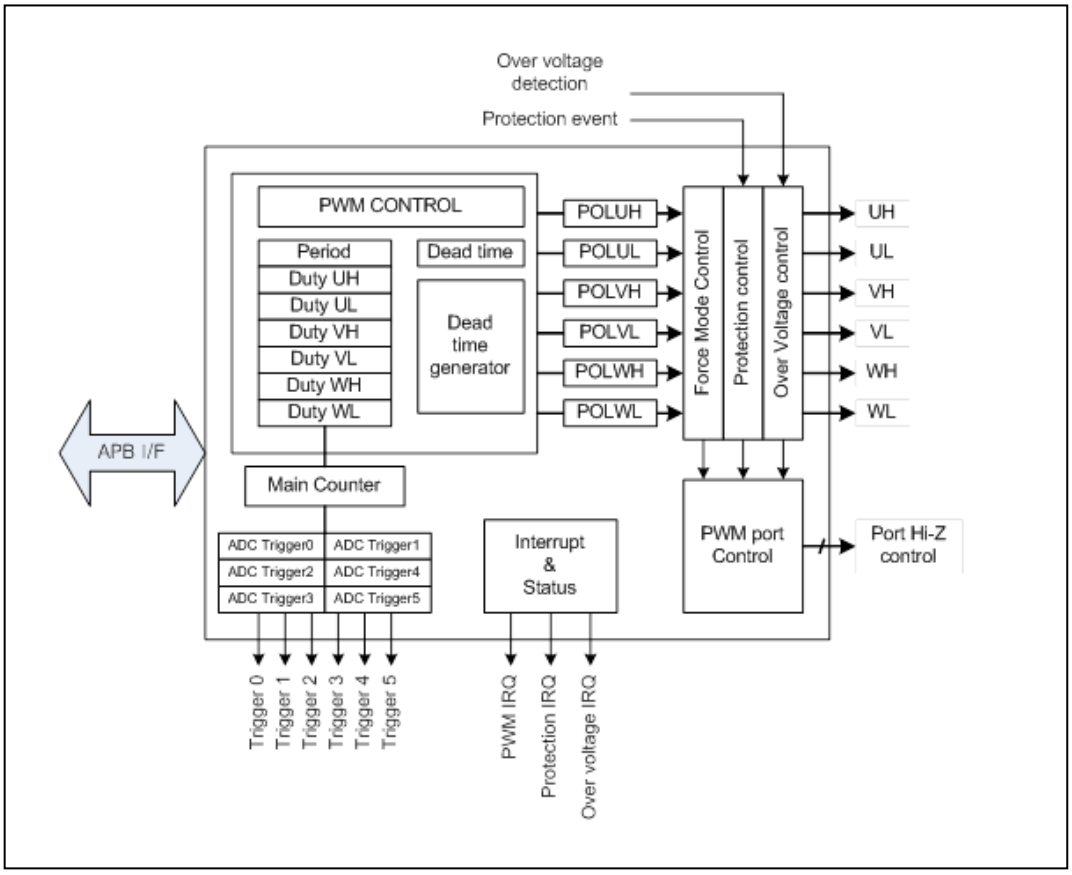


Figure 15.1 Block Diagram

Pin Description

Table 15.1 External Signals

Pin Name	Type	Description
MP0UH	O	MPWM 0 Phase-U H-side output
MP0UL	O	MPWM 0 Phase-U L-side output
MP0VH	O	MPWM 0 Phase-V H-side output
MP0VL	O	MPWM 0 Phase-V L-side output
MP0WH	O	MPWM 0 Phase-W H-side output
MP0WL	O	MPWM 0 Phase-W L-side output
PRTIN0	I	MPWM 0 Protection Input
OVIN0	I	MPWM 0 Over-voltage Input

Registers

The base address of MPWM is `0x4000_4000`. Table 15.2 shows the register memory map.

Table 15.2 MPWM Register Map

Name	Offset	Type	Description	Reset Value
MP0.MR	0x0000	RW	MPWM Mode register	0x0000_0000
MP0.OLR	0x0004	RW	MPWM Output Level register	0x0000_0000
MP0.FOLR	0x0008	RW	MPWM Force Output register	0x0000_0000
MP0.PRD	0x000C	RW	MPWM Period register	0x0000_0002
MP0.DUH	0x0010	RW	MPWM Duty UH register	0x0000_0001
MP0.DVH	0x0014	RW	MPWM Duty VH register	0x0000_0001
MP0.DWH	0x0018	RW	MPWM Duty WH register	0x0000_0001
MP0.DUL	0x001C	RW	MPWM Duty UL register	0x0000_0001
MP0.DVL	0x0020	RW	MPWM Duty VL register	0x0000_0001
MP0.DWL	0x0024	RW	MPWM Duty WL register	0x0000_0001
MP0.CR1	0x0028	RW	MPWM Control register 1	0x0000_0000
MP0.CR2	0x002C	RW	MPWM Control register 2	0x0000_0000
MP0.SR	0x0030	R	MPWM Status register	0x0000_0000
MP0.IER	0x0034	RW	MPWM Interrupt Enable	0x0000_0000
MP0.CNT	0x0038	R	MPWM counter register	0x0000_0001
MP0.DTR	0x003C	RW	MPWM dead time control	0x0000_0000
MP0.PCR0	0x0040	RW	MPWM protection 0 control register	0x0000_0000
MP0.PSR0	0x0044	RW	MPWM protection 0 status register	0x0000_0080
MP0.PCR1	0x0048	RW	MPWM protection 1 control register	0x0000_0000
MP0.PSR1	0x004C	RW	MPWM protection 1 status register	0x0000_0000
-	0x0054	-	Reserved	-
MP0.ATR1	0x0058	RW	MPWM ADC Trigger reg1	0x0000_0000
MP0.ATR2	0x005C	RW	MPWM ADC Trigger reg2	0x0000_0000
MP0.ATR3	0x0060	RW	MPWM ADC Trigger reg3	0x0000_0000
MP0.ATR4	0x0064	RW	MPWM ADC Trigger reg4	0x0000_0000
MP0.ATR5	0x0068	RW	MPWM ADC Trigger reg5	0x0000_0000
MP0.ATR6	0x006C	RW	MPWM ADC Trigger reg6	0x0000_0000

MP0.MR MPWM Mode Register

The PWM Operation Mode register is a 16-bit register.

MP0.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 initial PWM period and duty setting is completed, the UAO bit should be set once for updating the setting value into the 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 set state for at least 2-PWM clock periods. Otherwise, the update command can be missed and the internal registers will keep the previous data.

MCHMOD in the MP0.MR field is only effective when MOTORB in MP0.MR is a clear "0". Otherwise, the MCHMOD field value will be ignored internally and will retain a "00" value.

UPDOWN in the MP0.MR field is only effective when MOTORB in MP0.MR is set to "1". Otherwise, the UPDOWN field value will be ignored internally and will retain a "1" value. In the motor mode, the counter is always an up-down count operation.

MP0.OLR MPWM Output Level Register

The PWM Port Mode register is a 16-bit register.

MP0.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

5	WHL	0	Default Level
		1	Inverted Level
4	VHL	0	Default Level
		1	Inverted Level
3	UHL	0	Default Level
		1	Inverted Level
2	WLL	0	Default Level
		1	Inverted Level
1	VLL	0	Default Level
		1	Inverted Level
0	ULL	0	Default Level
		1	Inverted Level

The normal level is defined in each operating mode as shown in Table 15.3.

Table 15.3. MPWM Register Map

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. The example shown is for WH signal polarity control.

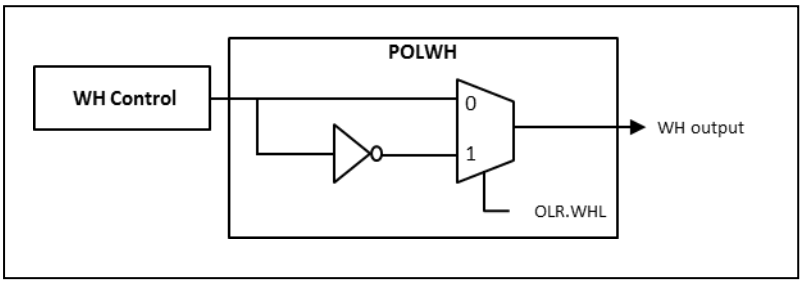


Figure 15.2. Polarity Control Block

MP0.FOLR MPWM Force Output Level Register

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

MP0.FOLR=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’

MP0.CR1 MPWM Control Register 1

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

MP0.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	IRQN	IRQ interval number
8		(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.

MP0.CR2 MPWM Control Register 2

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

MP0.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

MP0.PRD MPWM Period Register

The PWM Period Register is a 16-bit register.

MP0.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)

MP0.DUH MPWM Duty UH Register

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

MP0.DUH=0x4000_4010

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

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

MP0.DVH MPWM Duty VH Register

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

MP0.DVH=0x4000_4014

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

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

MP0.DWH MPWM Duty WH Register

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

MP0.DWH=0x4000_4018

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

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

MP0.DUL MPWM Duty UL Register

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

MP0.DUL=0x4000_401C

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

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

MP0.DVL MPWM Duty VL Register

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

MP0.DVL=0x4000_4020

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

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

MP0.DWL MPWM Duty WL Register

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

MP0.DWL=0x4000_4024

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

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

MP0.IER MPWM Interrupt Enable Register

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

MP0.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

MP0.IER[5:0] control bits are shared by 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 cases, 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.

MP0.SR MPWM Status Register

The PWM Status Register is a 16-bit register.

MP0.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	Current PWM Count mode is Up
		1	Current PWM Count mode is Down
14 12	IRQCNT		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

MP0.SR[5:0] status bits are shared by the duty match interrupt event and the ADC trigger match interrupt event. When ADC trigger mode is disabled, the interrupt is generated by the duty match condition. In other cases, 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.

MP0.CNT MPWM Counter Register

The PWM Counter register is a 16-bit Read-Only register.

MP0.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

MP0.DTR MPWM Dead Time Register

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

MP0.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.
		1	(Turn off both output when both H-side and L-side are active.) 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	

Protect short condtion 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 operates to force both H-side and L-side to low level.

MP0.PCR MPWM Protection 0,1 Control Register

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

MP0.PCR0=0x4000_4040,MP0.PCR1=0x4000_4048

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROT0EN	PROT0POL				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

MP0.PSR MPWM Protection 0,1 Status Register

The PWM Protection Status register is a 16-bit register. This register indicates which outputs are disabled. Users can set the output masks manually.

When writing a value, if PROTKEY is not written, the written values are ignored.

MP0.PSR0=0x4000_4044,MP0.PSR1=0x4000_404C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROTKEY								PROTIF		WHPROT	VHPROT	UHPROT	WLPROT	VLPROT	ULPROT
-								0		0	0	0	0	0	0
WO								RC		RW	RW	RW	RW	RW	RW

15	PROTKEY	Protection Clear Access Key
8		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 MP.PCR register is enabled, on any asserting signal on the external protection pins, the PWM output is prohibited with output values defined in MP.FOLR register. Additionally, users can prohibit the output manually by writing the designated value into the MP.PSR register.

MP0.ATRm MPWMn ADC Trigger Counter m Register

MP0.ATR1 MPWM ADC Trigger Counter 1 Register

MP0.ATR2 MPWM ADC Trigger Counter 2 Register

MP0.ATR3 MPWM ADC Trigger Counter 3 Register

MP0.ATR4 MPWM ADC Trigger Counter 4 Register

MP0.ATR5 MPWM ADC Trigger Counter 5 Register

MP0.ATR6 MPWM ADC Trigger Counter 6 Register

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

MP0.ATR1=0x4000_4058
MP0.ATR2=0x4000_405C
MP0.ATR3=0x4000_4060
MP0.ATR4=0x4000_4064
MP0.ATR5=0x4000_4068
MP0.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 16	ATMOD	ADC trigger Mode register
		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 0	ATCNT	ADC Trigger counter (it should be less than PWM period)

Functional Description

The MPWM includes 3 channels, each of which controls a pair of outputs. In normal PWM mode, each channel runs independently. Six PWM outputs can be generated.

Each PWM output is built with various settings. Figure 15.3 shows the diagram for generating PWM.

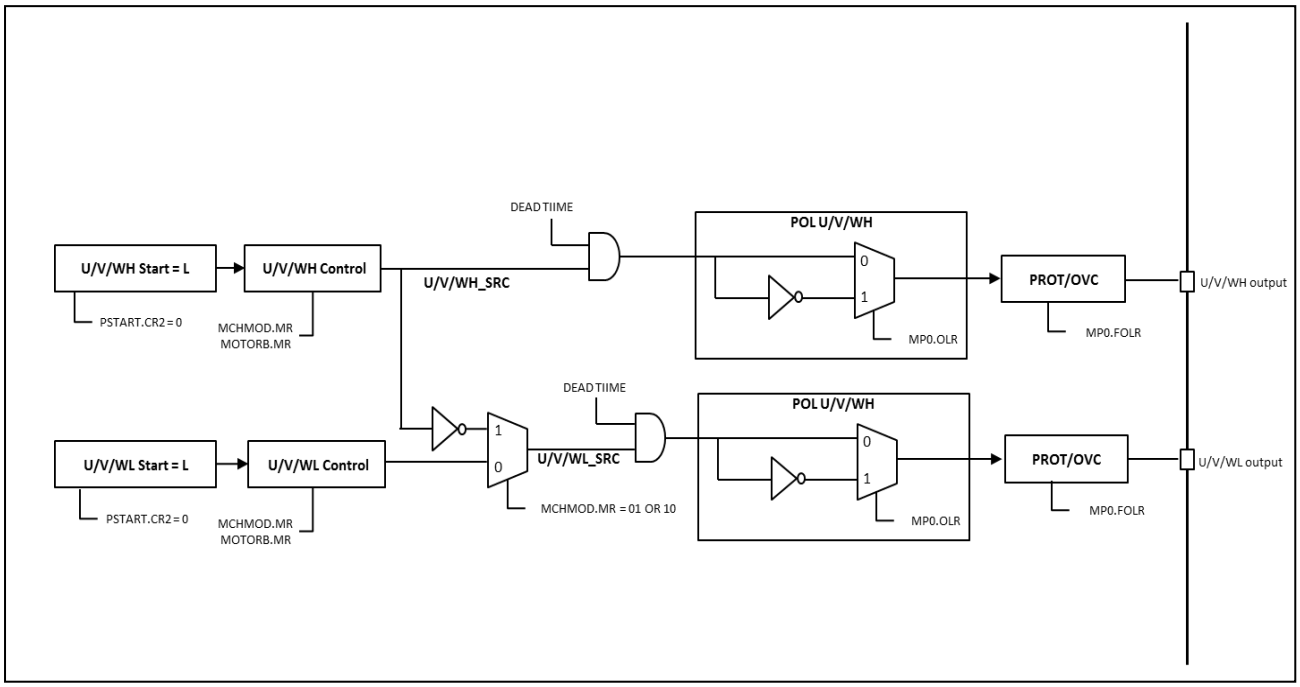


Figure 15.3. PWM Output Generation Chain

Normal PWM UP Count Mode Timing

In normal PWM mode, each channel runs independently. Six PWM outputs can be generated. An example of the waveform is shown in Figure 15.4. Before PSTART is activated, the PWM output stays at the default value L. When PSTART is enabled, the period counter starts up count up to the MP0.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 derived at the start of the counter value during duty value time.

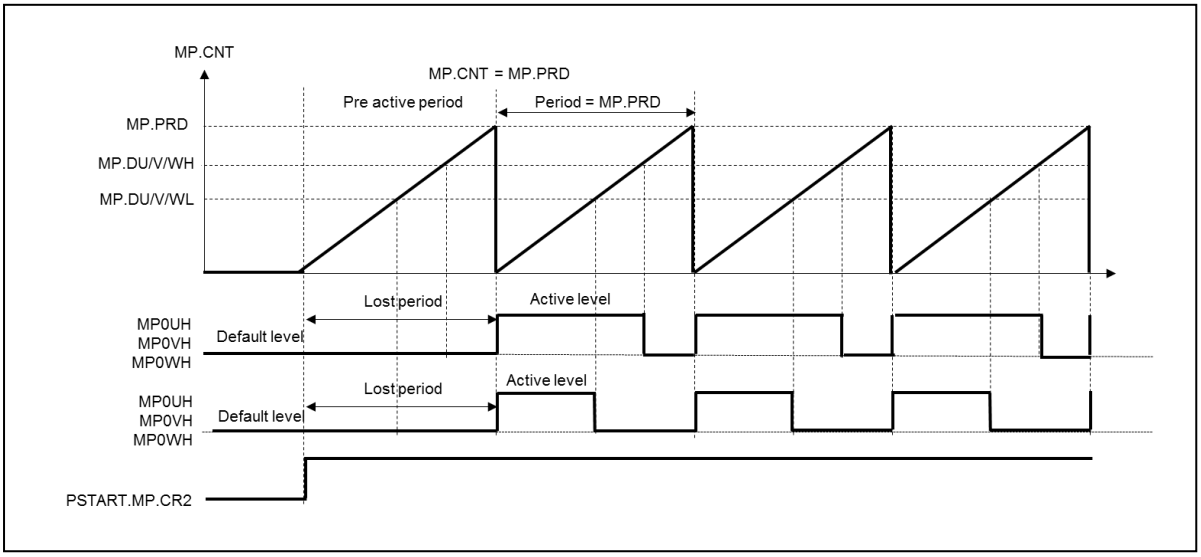


Figure 15.4. UP Count Mode Waveform (MOTORB=1, UPDOWN=0)

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 that in UP count mode. The default active level is opposite in a pair PWM output. This output polarity can be controlled by the MP0.OLR register.

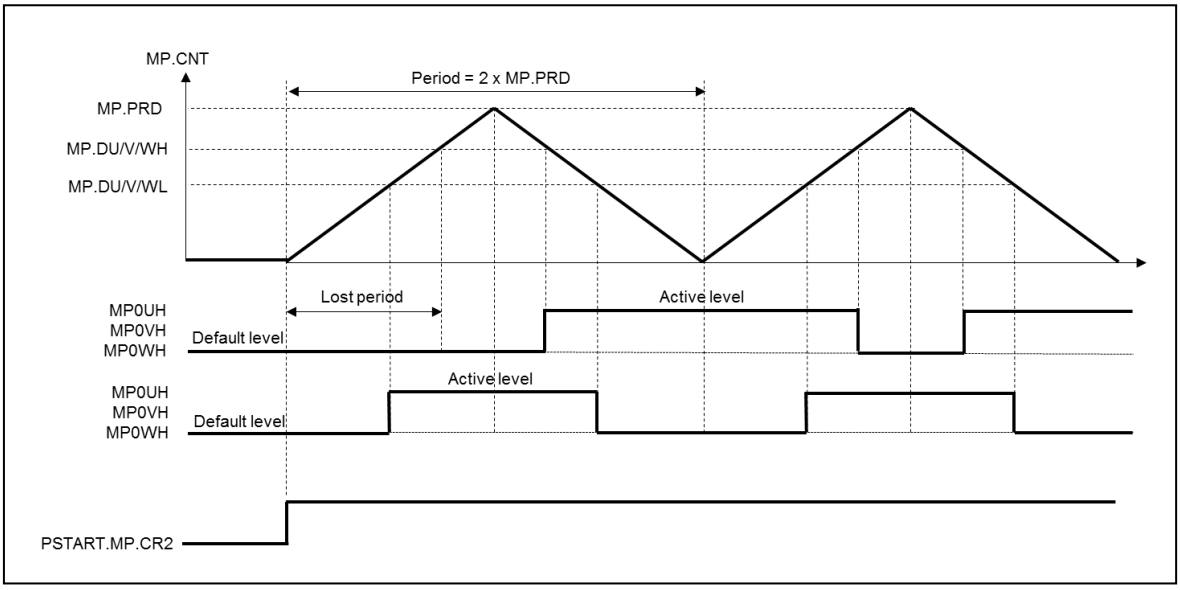


Figure 15.5. UP/DOWN Count Mode Waveform (MOTORB=0, MCHMOD=0, UPDOWN=1)

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 15.6 shows an example of a 2-channel symmetric mode waveform.

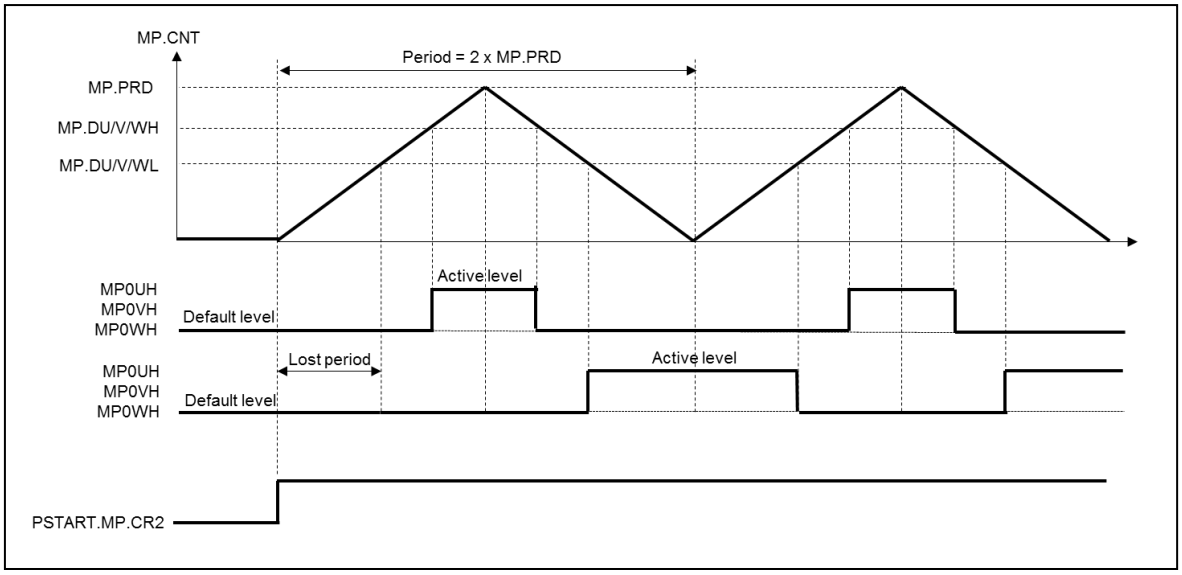


Figure 15.6. 2-Channel Symmetric Mode Waveform (MOTORB=0,MCHMOD=00)

The default start level of both H-side and L-side is low. For the H-side, PWM ouput 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 which 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 the H-side. During up count period, the H-side DUTY register matching condition generates the active level pulse and during down count period, the L-side DUTY register matiching condition generates the default level pulse.

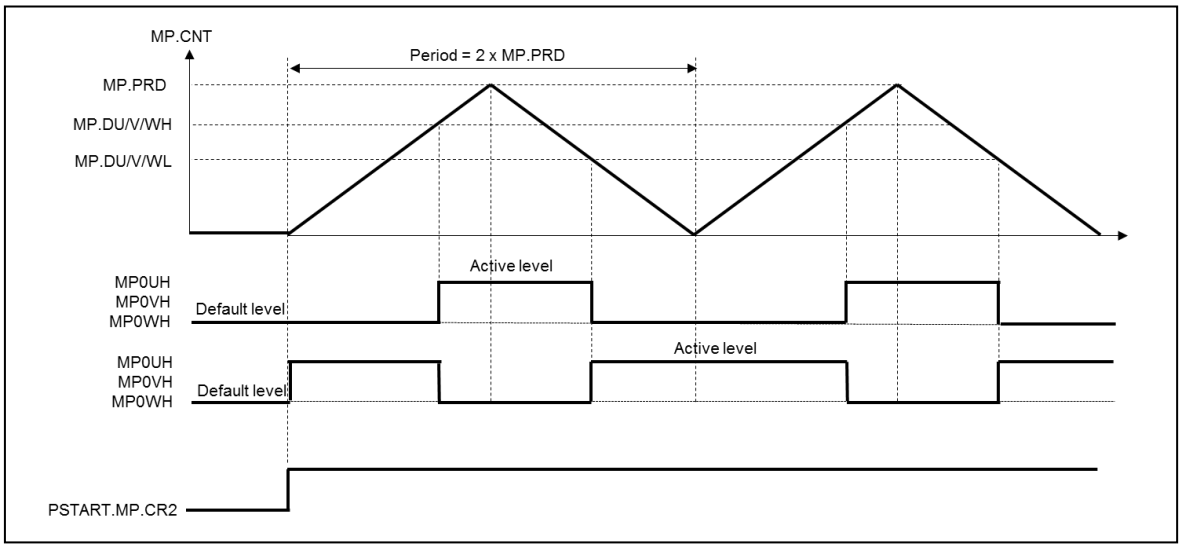


Figure 15.7. 1-Channel Asymmetric Mode Waveform (MOTORB=0,MCHMOD=01)

The default start level of both H-side and L-side is low. For the H-side, PWM ouput 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 the H-side output.

Motor PWM 1-Channel Symmetric Mode Timing

The 1-channel symmetric mode generates a symmetric duration pulse which is 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 generates the active level pulse and during down count period, the H-side DUTY register matiching condition also generates the default level pulse.

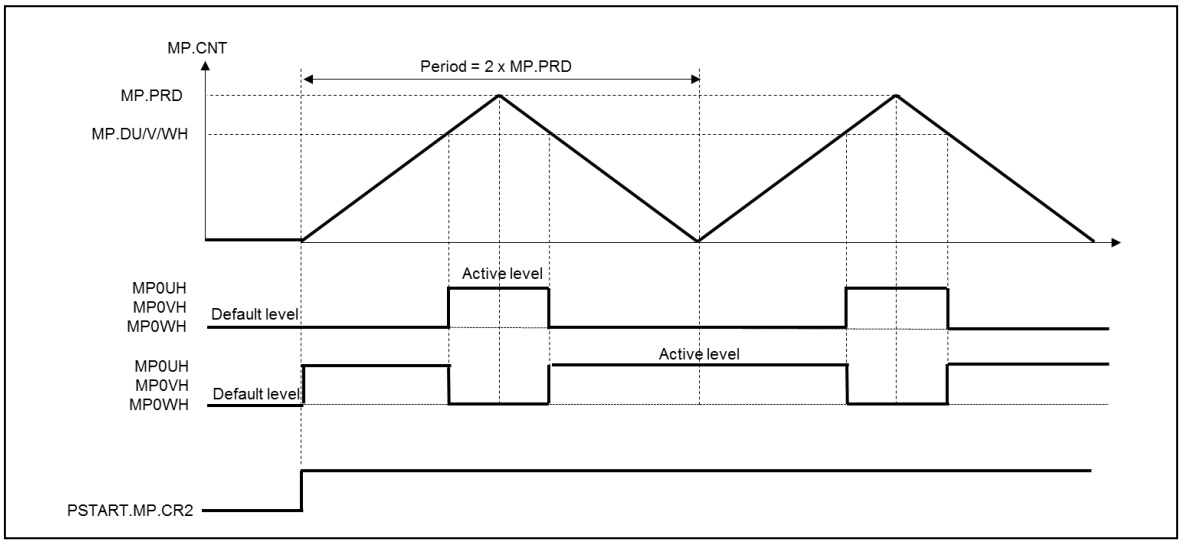


Figure 15.8. 1-Channel Symmetric Mode Waveform (MOTORB=0,MCHMOD=10)

The default start level of both H-side and L-side is low. For the H-side, PWM ouput 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 a dead-time function. This function is only available in the Motor PWM mode. When either theH-side or L-side output changes to active level, an amount of dead-time is inserted if the DTEN.MP.DTR bit is enabled.

The duration of dead-time is decided 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 the 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 15.9 is an example of dead-time operation in 1 channel symmetric mode.

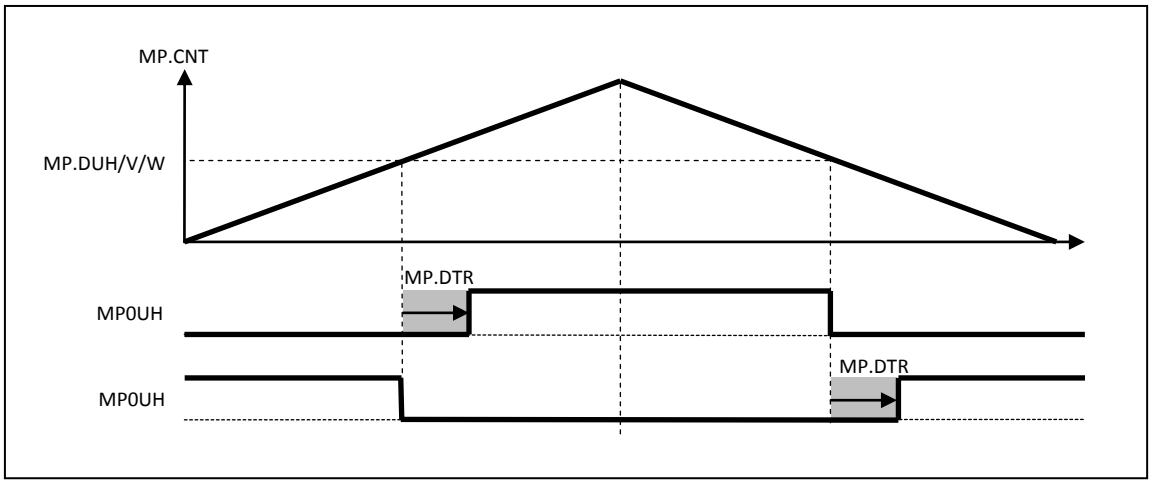


Figure 15.9. PWM Dead-time Operation Timing Diagram (Symmetric Mode)

Figure 15.10 displays an example of dead-time operation in 1-channel asymmetric mode.

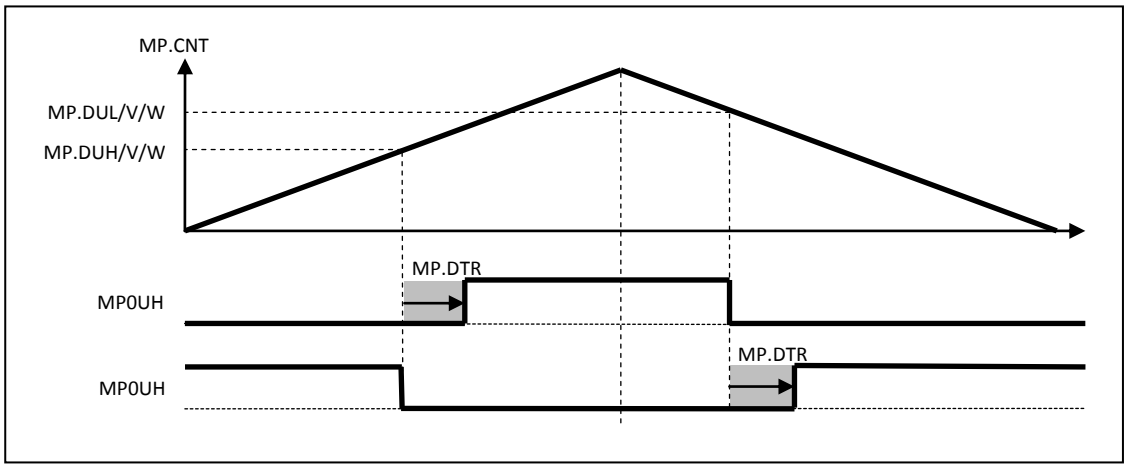


Figure 15.10. PWM Dead-Time Operation Timing Diagram (Asymmetric Mode)

The dead-time function is not available for 2-channel symmetric mode. Therefore, the dead condition is generated by each channel's duty control.

MPWM Dead-time Timing Examples for Special Conditions

Figure 15.11 shows the operation of dead-time.

In normal dead-time, 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.

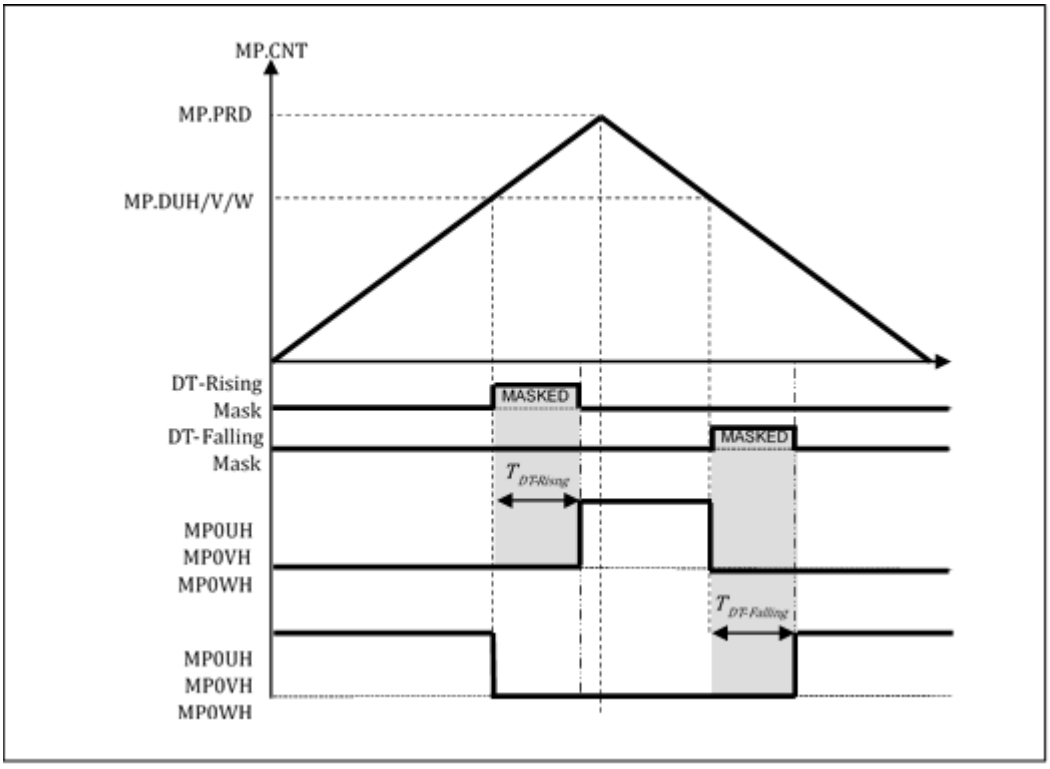


Figure 15.11. Normal Dead-Time Operation (T_{DUTY}>T_{DT})

The following figures display special-case scenarios of dead time configurations.

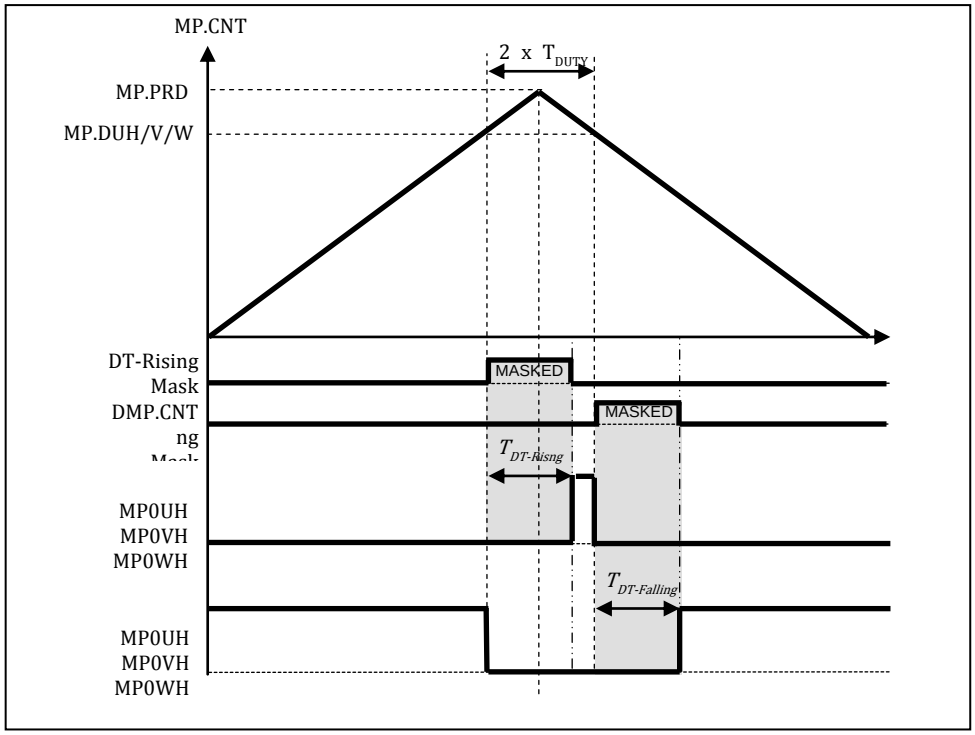


Figure 15.12. Minimum H-Side Pulse Timing (T_{DUTY}<T_{DT}<2xT_{DUTY})

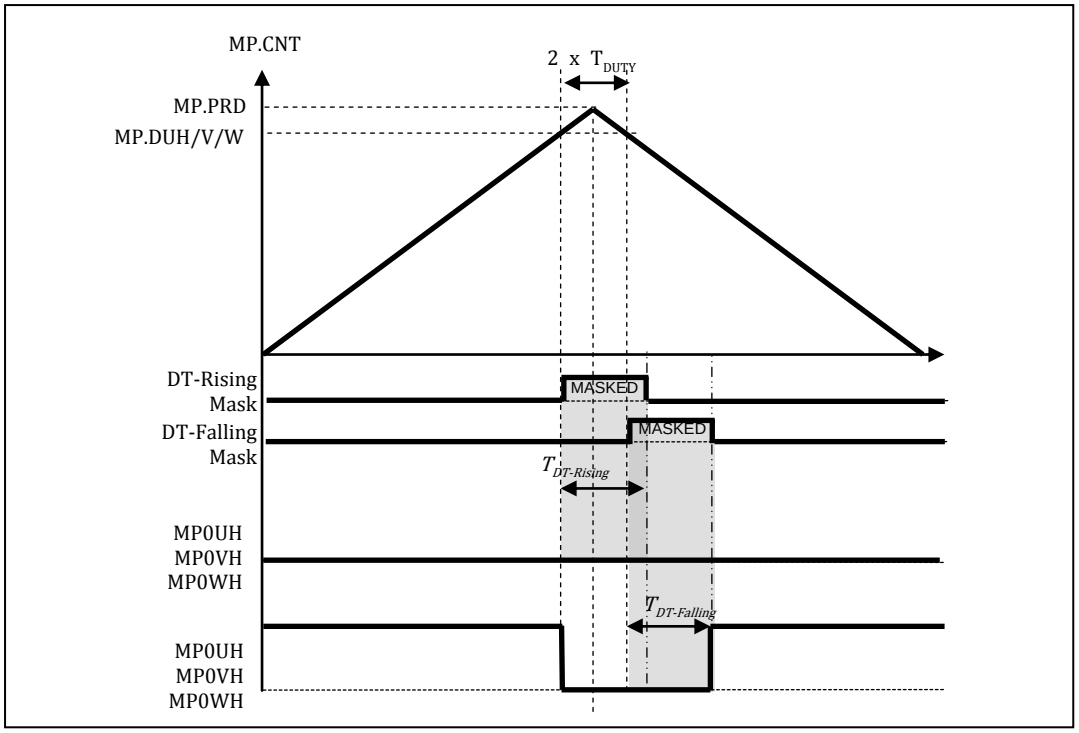


Figure 15.13. Zero H-Side Pulse Timing (T_{DT}>2xT_{DUTY})

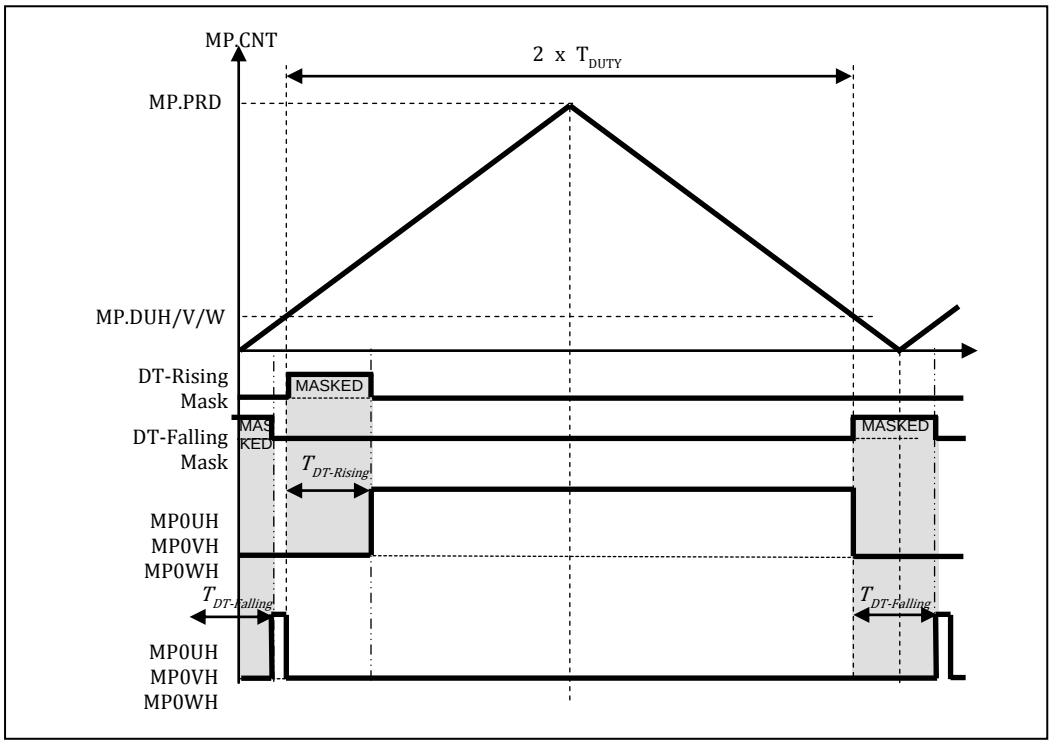


Figure 15.14. Minimum L-Side Pulse Timing (T_{DT}<Period-T_{DUTY})

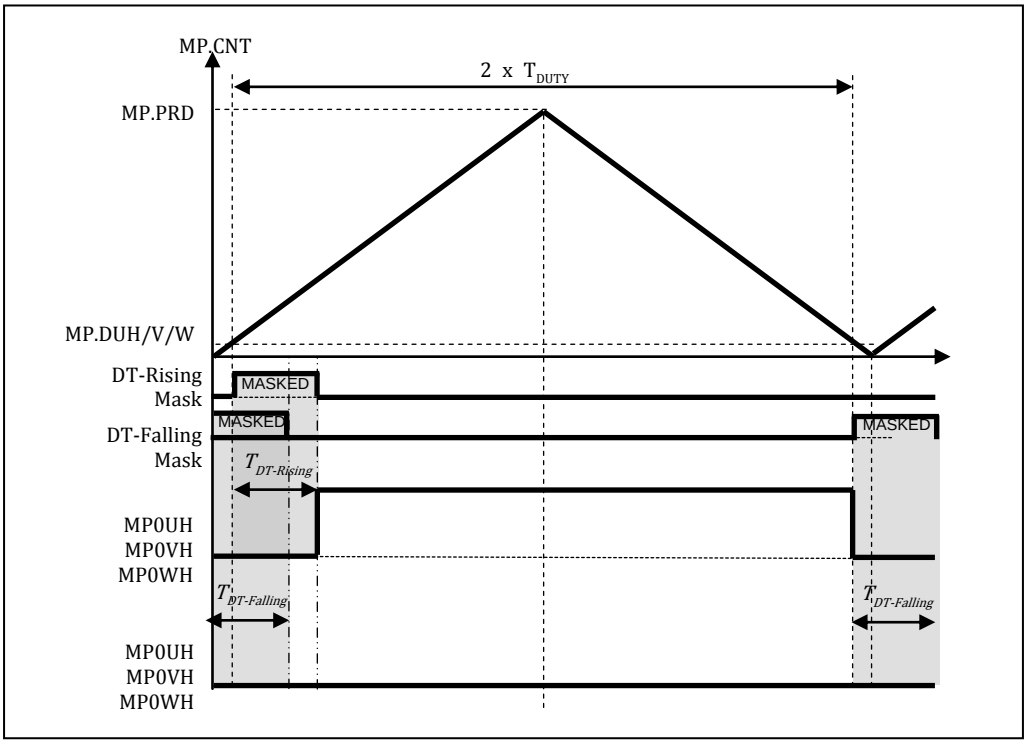


Figure 15.15. Zero L-Side Pulse Timing (T_{DT}>Period-T_{DUTY})

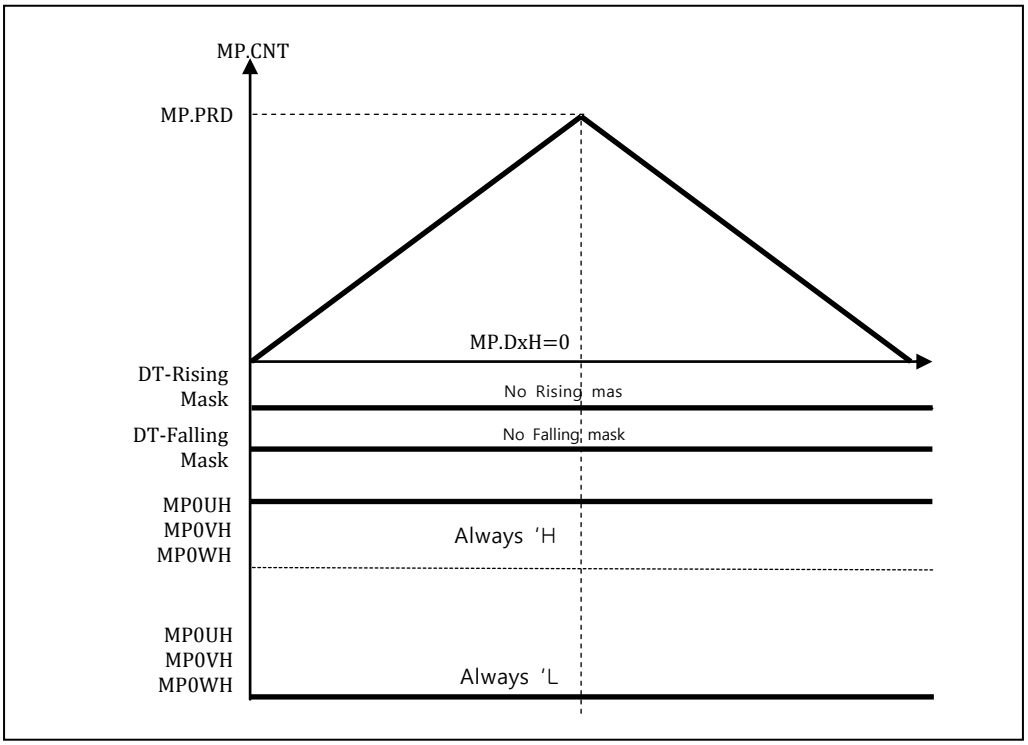


Figure 15.16. H-Side Always On (T_{DUTY}=Period: Dead-Time Disabled)

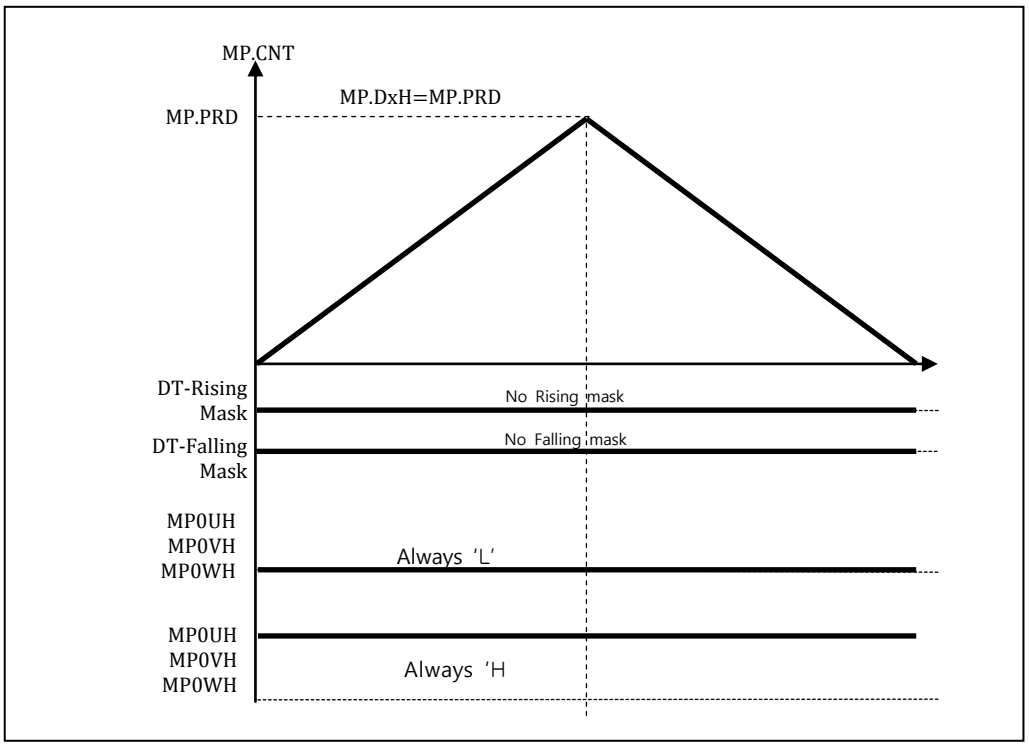


Figure 15.17. L-Side Always On (T_{DUTY}='0': Dead-Time Disabled)

Symmetrical Mode vs Asymmetrical Mode

In symmetrical mode, the waveform is symmetrical on both sides of the mid-point of the period. The duty comparison is performed twice in both up and down count period.

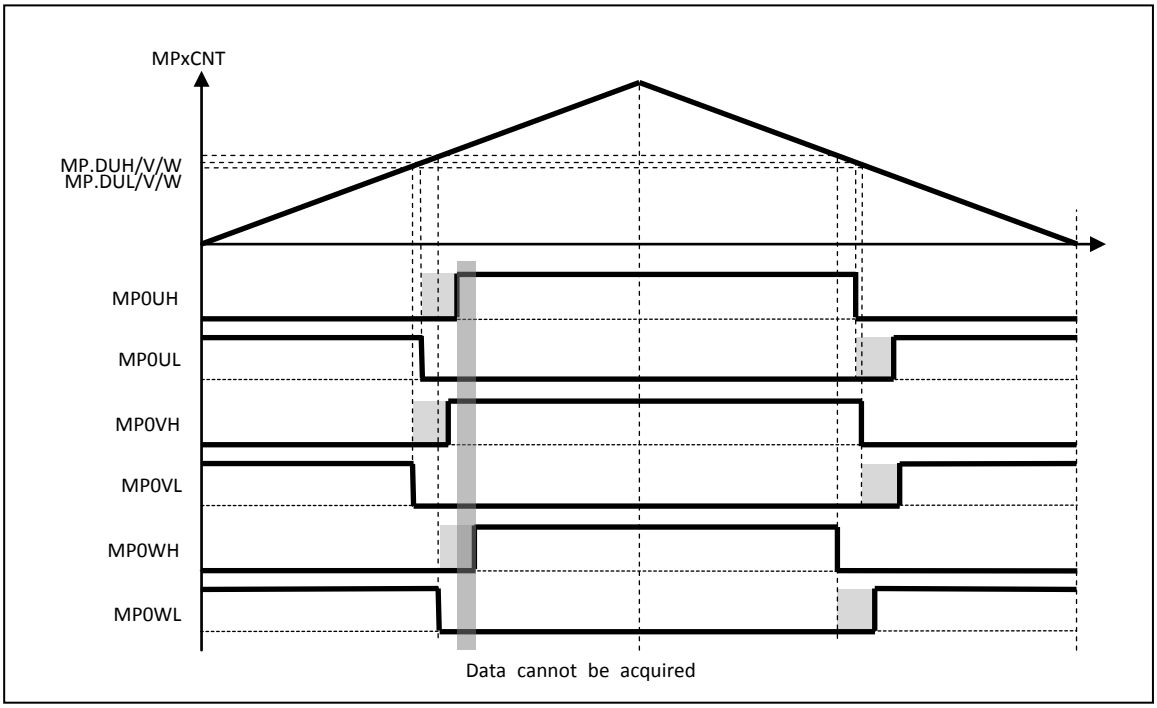


Figure 15.18. Symmetrical PWM Timing

In asymmetrical mode, the waveform is not symmetrical from the mid-point of the period. The duty comparison of H-side is performed in both up count period. The duty comparison of L-side is performed in both down count period.

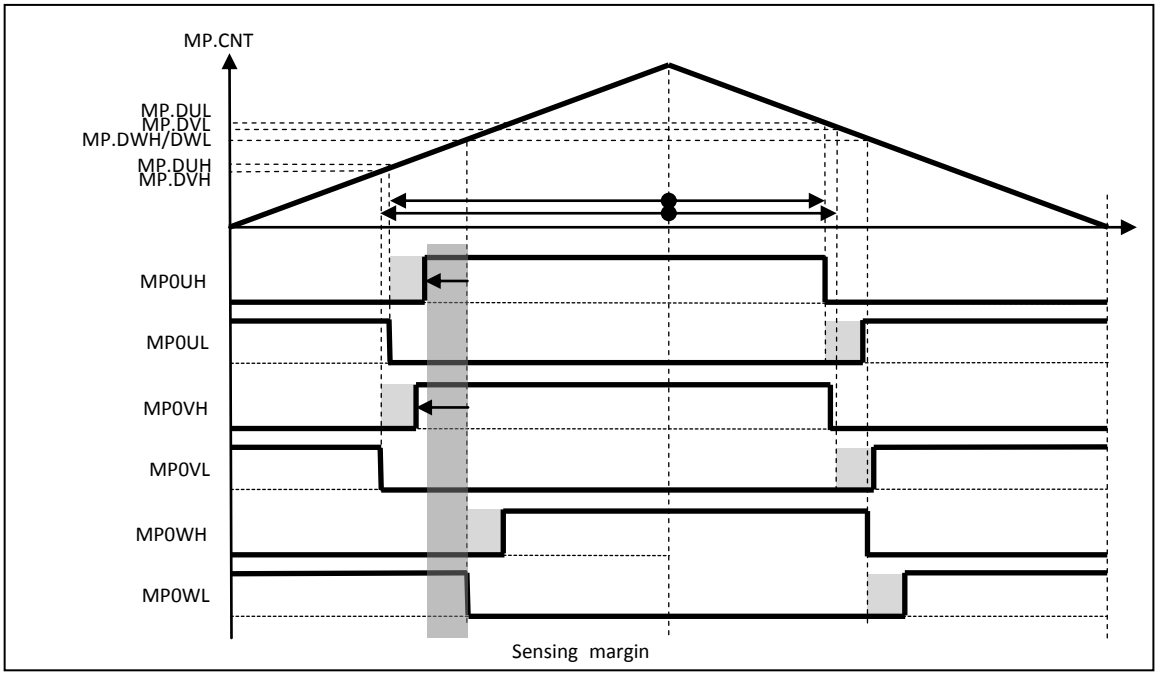


Figure 15.19. Asymmetrical PWM Timing and Sensing Margin

ADC Triggering Function

Six ADC trigger timing registers are provided. This dedicated register creates a trigger signal to start the ADC conversion. The conversion channel of ADC is defined in the ADC control register.

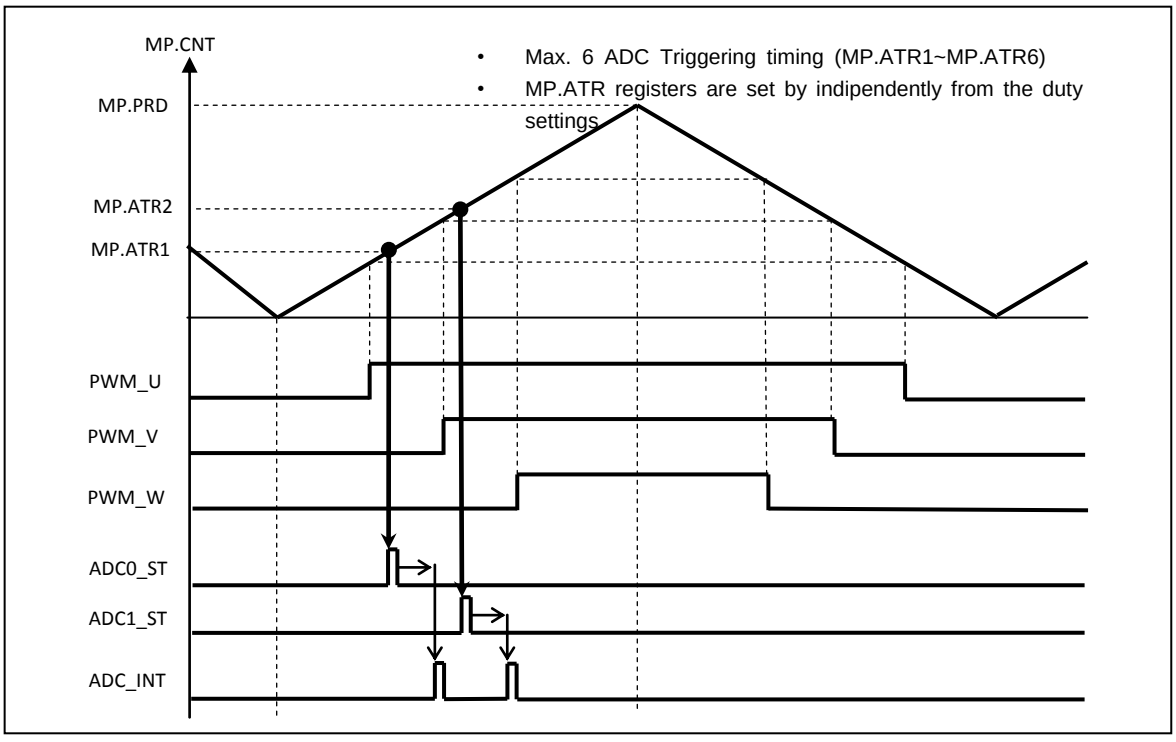


Figure 15.20. ADC Triggering Function Timing Diagram

Figure 15.21 shows an example of ADC data acquisition.

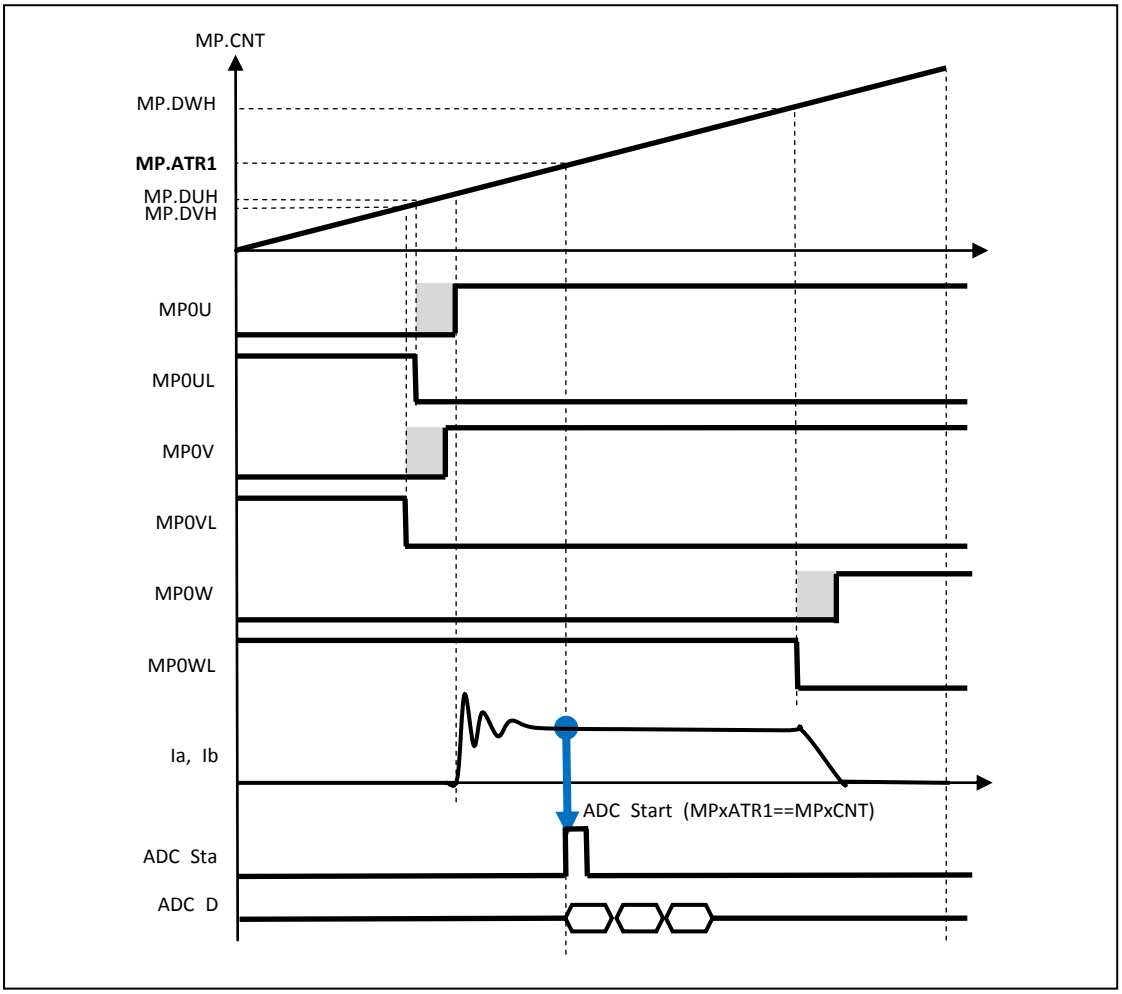


Figure 15.21. An Example of ADC Aquisition Timing by Event from MPWM

Interrupt Generation Timing

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

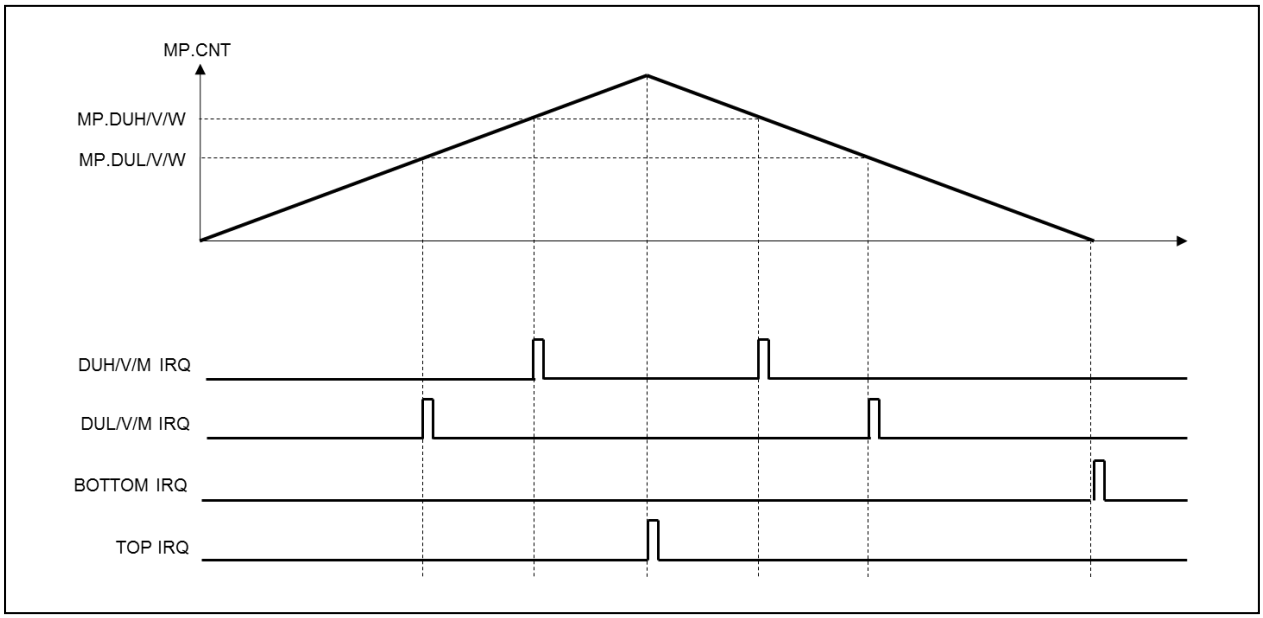


Figure 15.22. Interrupt Generation Timing

16. 12-Bit A/D Converter

Introduction

The A/D Converter (ADC) block consists of two independent ADC units. Features include:

- 11 channels of analog inputs (each ADC has 8 input channels)
- Single and Continuous conversion mode
- Up to 8 times sequential conversion supports
- Software trigger supports
- 4 internal trigger sources supports (PWMs, timers)
- Adjustable sample and hold time
- DMA transfers

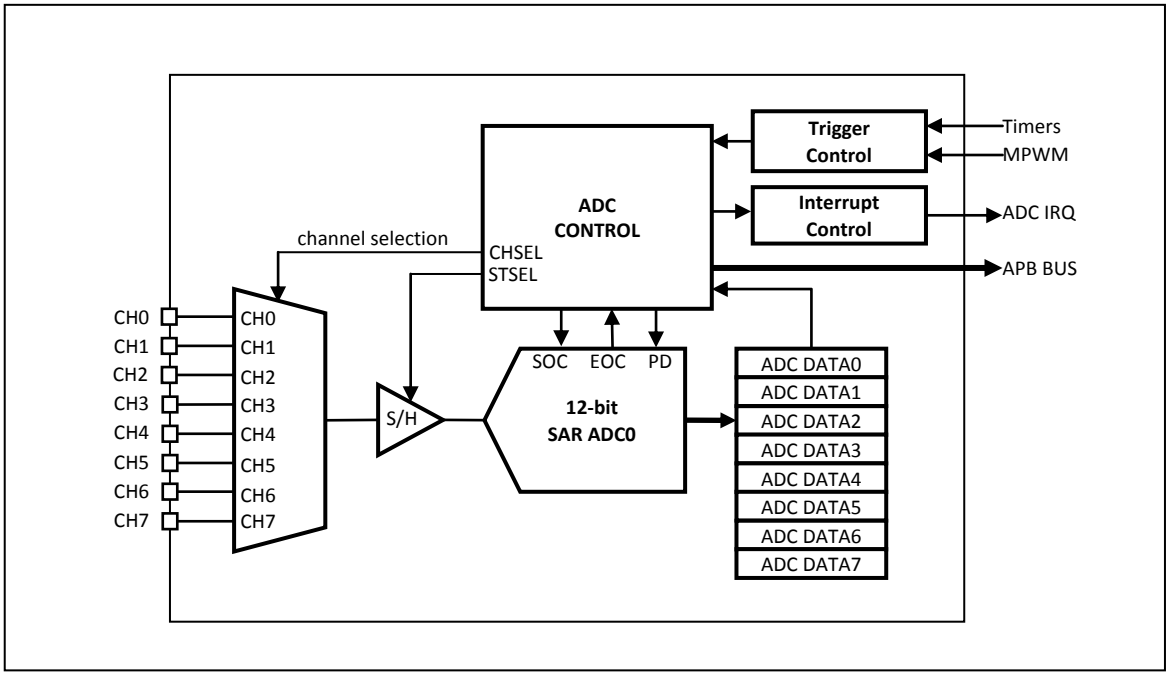


Figure 16.1 Block Diagram

Pin Description

Table 16.1 External Signal

Pin Name	Type	Description
VDD	P	Analog Power(3.0V~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

Registers

The base addresses of the ADC units are shown in Table 16.2.

Table 16.2 ADC Base Address

Name	Base Address
ADC0	0x4000_B000
ADC1	0x4000_B100

Table 16.3 ADC Register Map

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

AD*n*.MR ADC*n* Mode Register

The ADC Mode registers are 32-bit registers.

This register configures ADC Operation Mode. This register configures the ADC Operation Mode and should be written first before the other ADC registers are written.

AD0.MR=0x4000_B000, AD1.MR=0x4000_B100

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
														DMAEN	DMACH	STSEL					SEQCNT			ADEN	ARST	ADMOD				TRGSEL	
														0x0	0x0	0x0					0x0			0x0	0x0	0x0				0x0	
														RW	RW	RW					RW			RW	RW	RW				RW	

17	DMAEN	DMA enable bit – should be set to ‘1’ when ADCEN=’1’. When DMA function is enabled, DMA request at every end of conversion (also in sequential mode) and interrupt request only be generated when ADC receives DMA done from DMAC.			
16	DMACH	DMA channel option When DMACH is set, Channel information of DMA data will be located at ADn.DDR[3:0] for half word size transfer. Channel information is at ADn.DDR [19:16] in default.(DMACH is low)			
15 12	STSEL	Sampling Time Selection ADC Sample & Hold circuit sampling time become (2 + STSEL[3:0]) 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.			
		000	Single mode	100	5 sequence AD conversion
		001	2 sequence AD conversion	101	6 sequence AD conversion
		010	3 sequence AD conversion	110	7 sequence AD conversion
		011	4 sequence AD conversion	111	8 sequence AD conversion
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		
		01	Burst conversion mode		
		10	Reserved		
		11	Reserved		
1 0	TRGSEL	00	Event Trigger Disabled/Soft-Trigger Only		
		01	Timer Event Trigger		
		10	MPWM0 Event Trigger		
		11	Reserved		

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

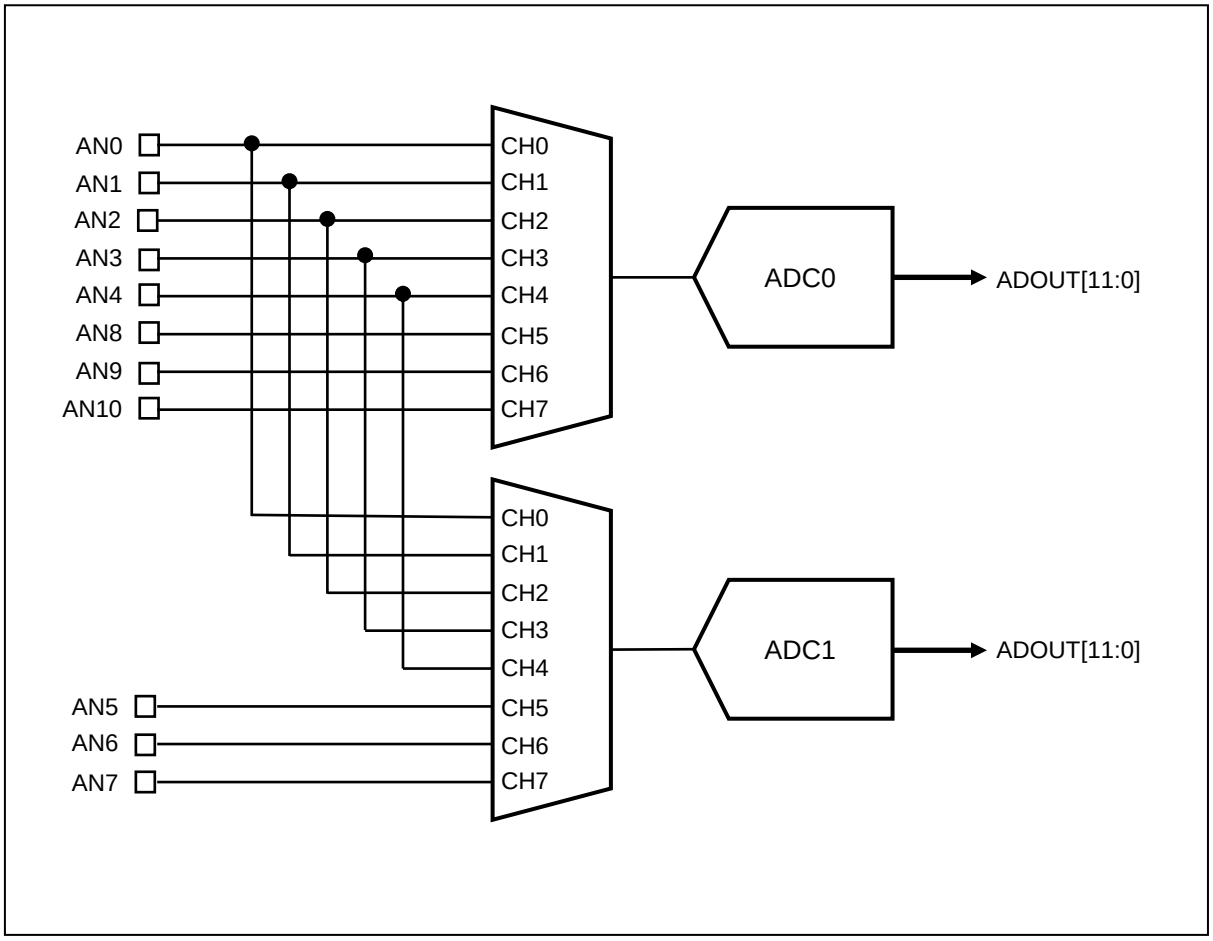


Figure 16.2 Analog Channel Block Diagram

AD*n*.CSCR ADC*n* Current Sequence/Channel Register

ADC Current Sequence/Channel registers are 7-bit registers. This register consists of Current Sequence Numbers (CSEQN) and Current Active Channel values. CSEQN can be written to set the current sequence number immediately. This register should be written first before ADn.SCSR is written.

AD0.CSCR=0x4000_B004, AD1.CSCR=0x4000_B104

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

7 4	CSEQN	Current Sequence Number , can write when not abusy	
		0000	Current Sequence is 0
		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	reserved
		1001	reserved
		1010	reserved
		1011	reserved
		1100	reserved
		1101	reserved
		1110	reserved
		1111	reserved

AD*n*.CCR ADC*n* Clock Control Register

ADC Control Registers are 16-bit registers.

AD0.CCR1=0x4000_B008, AD1.CCR1=0x4000_B108

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	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)
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*n*.TRG ADC Trigger Selection Register

The ADC Trigger registers are 32-bit registers.

In Single/Burst Mode, all the bit fields are used. In Burst Conversion mode, only the BSTTRG bit field (bit3~bit0) is used.

AD0.TRG=0x4000_B00C, AD1.TRG=0x4000_B10C

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

30 28	SEQTRG7	8th Sequence Trigger Source
26 24	SEQTRG6	7th Sequence Trigger Source
22 20	SEQTRG5	6th Sequence Trigger Source
18 16	SEQTRG4	5th Sequence Trigger Source
14 12	SEQTRG3	4th Sequence Trigger Source
10 8	SEQTRG2	3rd Sequence Trigger Source
6 4	SEQTRG1	2nd Sequence Trigger Source
2	SEQTRG0	1st Sequence Trigger Source
0	BSTTRG	Burst conversion Trigger Source

Value	Timer (TRGSEL ‘2’h1)	MPWM0 (TRGSEL ‘2’h2)
0	Timer 0	MP0ATR1
1	Timer 1	MP0ATR2
2	Timer 2	MP0ATR3
3	Timer 3	MP0ATR4
4	Timer 8	MP0ATR5
5	Timer 9	MP0ATR6
6	-	BOTTOM
7	-	PERIOD

AD*n*.SCSR ADC Sequence Channel Selection Register

The ADC Burst Mode Channel Select Register is a 32-bit register.

AD0.SCSR=0x4000_B018, AD1.SCSR=0x4000_B118

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*n*.CR ADCn Control Register

The ADCn Control Register controls start or stop ADC conversion operations. This register is an 8-bit register.

AD0.CR=0x4000_B020, AD1.CR=0x4000_B120

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

7	ASTOP	0	No
		1	ADC conversion stop (will be clear next @ADC clock) If ASTOP set after conversion cycle start, present conversion would be completed.
0	ASTART	0	No ADC conversion
		1	ADC conversion start (will be clear next @ADC clock) ADCEN should be “1” to start ADC If ASTART is set as 1’h1 when ARST is 1’h0 in trigger event mode, ADC conversion will start once as SEQCNT set.

AD*n*.SR ADC*n* Status Register

The ADC Status Register is a 32-bit register.

AD0.SR=0x4000_B024, AD1.SR=0x4000_B124

7	6	5	4	3	2	1	0
EOC	ABUSY	DOVRUN	DMAIRQ	TRGIRQ	EOSIRQ	-	EOCIRQ
0	0	0	0	0	0	-	0
RO	RO	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
5	DOVRUN		DMA overrun flag (not interrupt) (DMA ACK didn’t come until end of next conversion)
4	DMAIRQ		DMA done received (DMA transfer is completed)
3	TRGIRQ		ADC Trigger interrupt flag(Write “1” to clear flag) (0: no int / 1: int occurred)
2	EOSIRQ		This flag will be set upon final end of a sequence (Write “1” to clear flag)
		0	None.
		1	End-of-Sequence(burst) Interrupt occurred
0	EOCIRQ		This flag will be set upon each conversion when a sequence occurs. Use this bit when polling the ADC for completion in single conversion mode. (Write “1” to clear flag)
		0	None.
		1	End-of-Conversion Interrupt occurred

AD*n*.IER Interrupt Enable Register

Four interrupts are provided for ADC operations. Each interrupt can be enabled by writing **1** to the corresponding bit in the ADn.IER register.

AD0.IER=0x4000_B028, AD1.IER=0x4000_B128

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

4	DMAIRQE	DMA done interrupt enable 0: interrupt disable 1: interrupt enable
3	TRGIRQE	ADC trigger conversion interrupt enable
2	EOSIRQE	ADC sequence conversion interrupt enable
0	EOCIRQE	ADC single conversion interrupt enable

AD*n*.DDR ADC*n* DMA Data Register

The ADC DMA Data Register is a 16-bit register. This register is a temporary register only for DMA transfer (A/D conversion data of just completed conversion).

AD0.DDR=0x4000_B02C, AD1.DDR=0x4000_B12C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC DMA Temporary Data												ADMACH			
0x000												0x0			
R												R			

15 4	ADDMAR	ADC conversion result data (12-bit)
3 0	ADMACH	ADC data channel indicator

AD*n*.DR ADC*n* Sequence Data Register 0~7

ADC Data Registers are 16-bit registers holding the ADC conversion from the result register.

AD0.DR0=0x4000_B030, AD0.DR1=0x4000_B034, AD0.DR2=0x4000_B038, AD0.DR3=0x4000_B03C
AD0.DR4=0x4000_B040, AD0.DR5=0x4000_B044, AD0.DR6=0x4000_B048, AD0.DR7=0x4000_B04C
AD1.DR0=0x4000_B130, AD1.DR1=0x4000_B134, AD1.DR2=0x4000_B138, AD1.DR3=0x4000_B13C
AD1.DR4=0x4000_B140, AD1.DR5=0x4000_B144, AD1.DR6=0x4000_B148, AD1.DR7=0x4000_B14C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCDATA															
0x000															
R															

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

Functional Description

ADC Single Mode Timing

ADC conversion is started when ADCn.CR.ASTART is written as '1' in single conversion mode. After ADCnCR.ASTART is set, Start of Conversion (SOC) will be activated in 3 ADC clocks, ADCn.SR.EOCIRQ will be set in 2 ADC clocks, and 2 PCLKs after the End of Conversion.

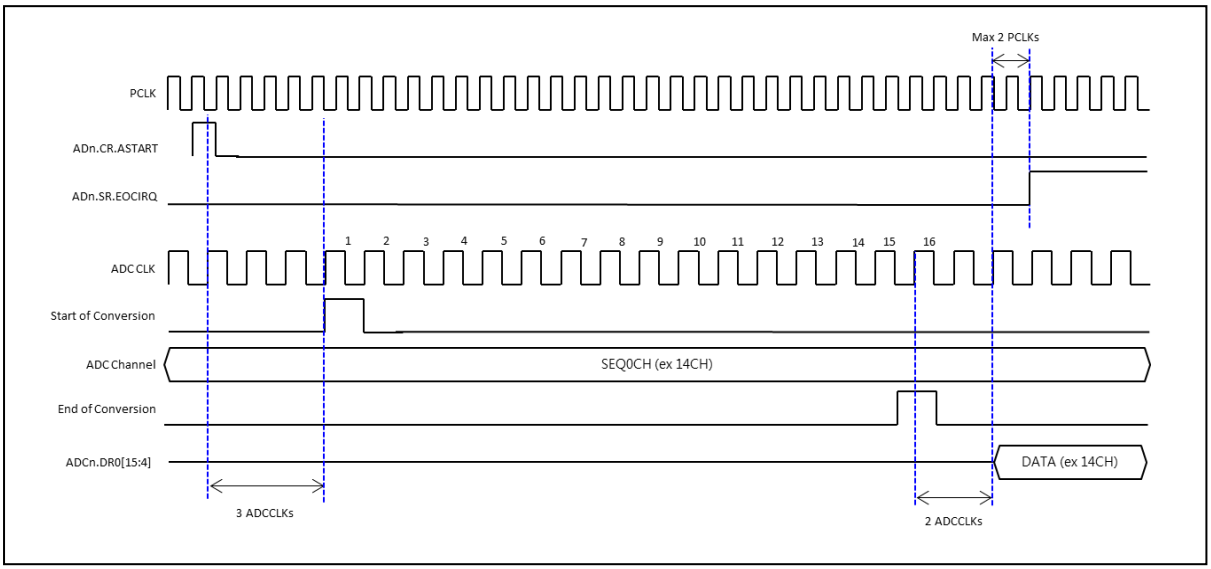


Figure 16.3. ADC Single Mode Timing (when ADCn.MR.AMOD = '0')

ADC Sequential Mode Timing Diagram

There are two sources to start conversion in burst mode: – TRG event (TIMER and MPWM) and ASTART. When TRGSEL is set as the timer event trigger or MPWM event trigger, the conversion is started by the trigger of ADn.TRG.BSTTRG (And.TRG[3:0]). For example, ADC conversion will be started by the trigger of TIMER9 if ADn.TRG.BSTTRG is set as TIMER9. Once the BSTTRG's trigger events occur, the ADC will convert all channels defined in sequence (And.MR.SEQCNT contains the number of channels to convert). See Figure 16.5.

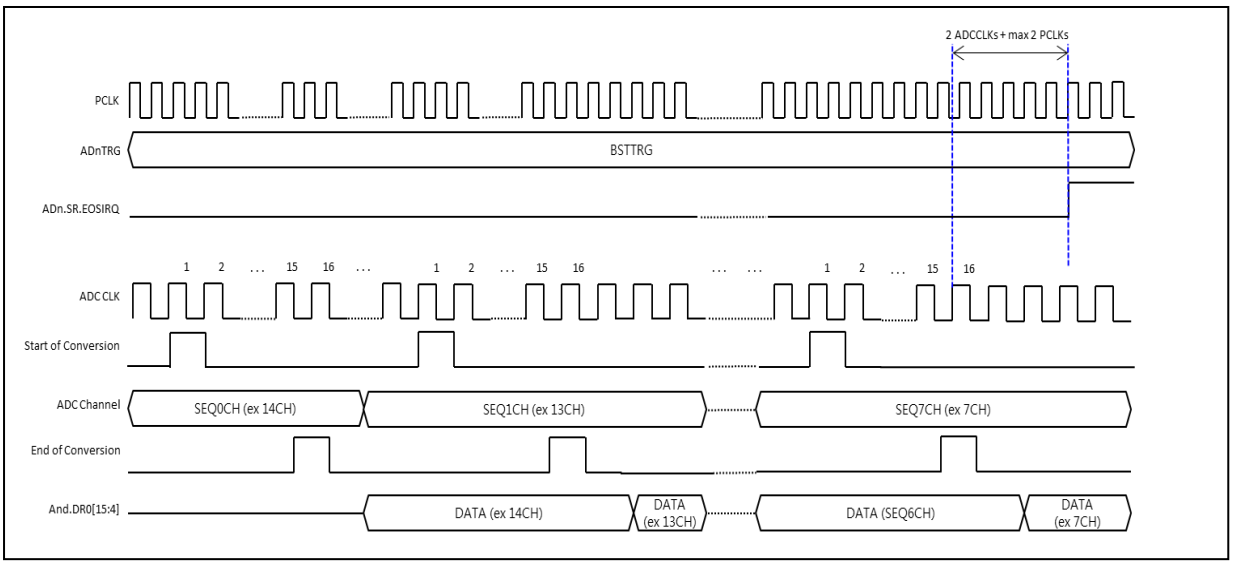


Figure 16.4. ADC Burst Mode Timing (When ADCn.MR.AMOD = '1')

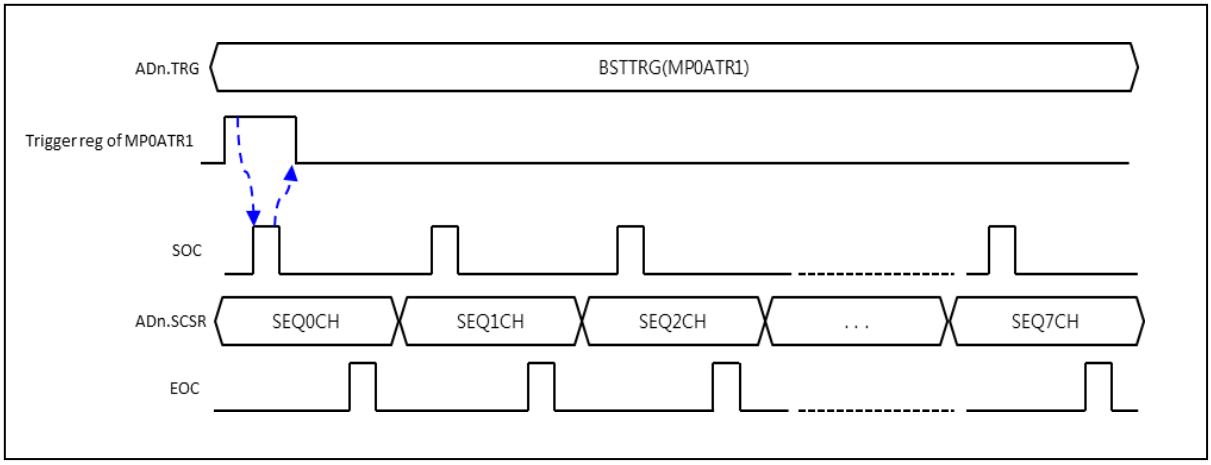


Figure 16.5. ADC Trigger Timing in Burst Mode (SEQCNT = 3'b111, 8 Sequence Conversion)

ADC Sequential Conversion Mode Timing Diagram

To set sequential conversion mode, ADn.MR.AMOD is 2'b00 and ADn.MR.SEQCNT is not 2'b00.

The operation of sequential mode is almost the same as burst mode. The difference is the source of SOC. Each SOC is made by the trigger of the SEQTRGx as each SEQCNT. See Figure 16.7.

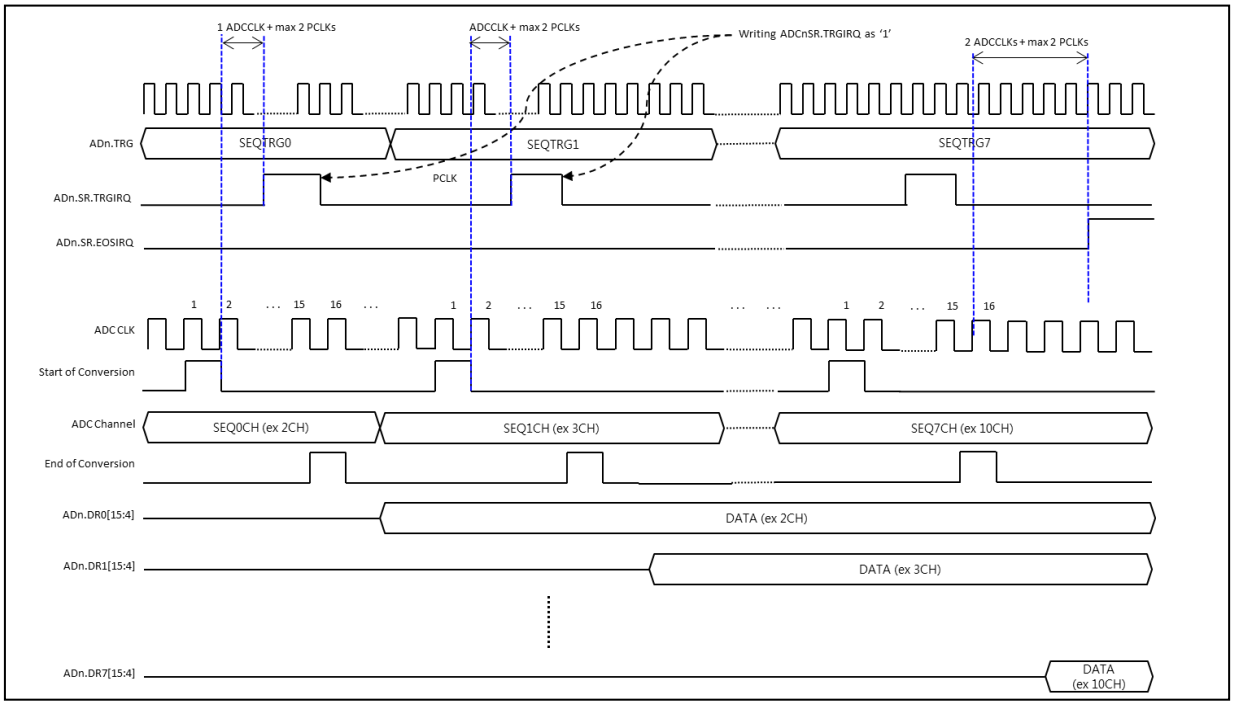


Figure 16.6. ADC Sequential Mode Timing (When ADn.MR.AMOD = 0 and ADn.MR.SEQCNT ≠ 0)

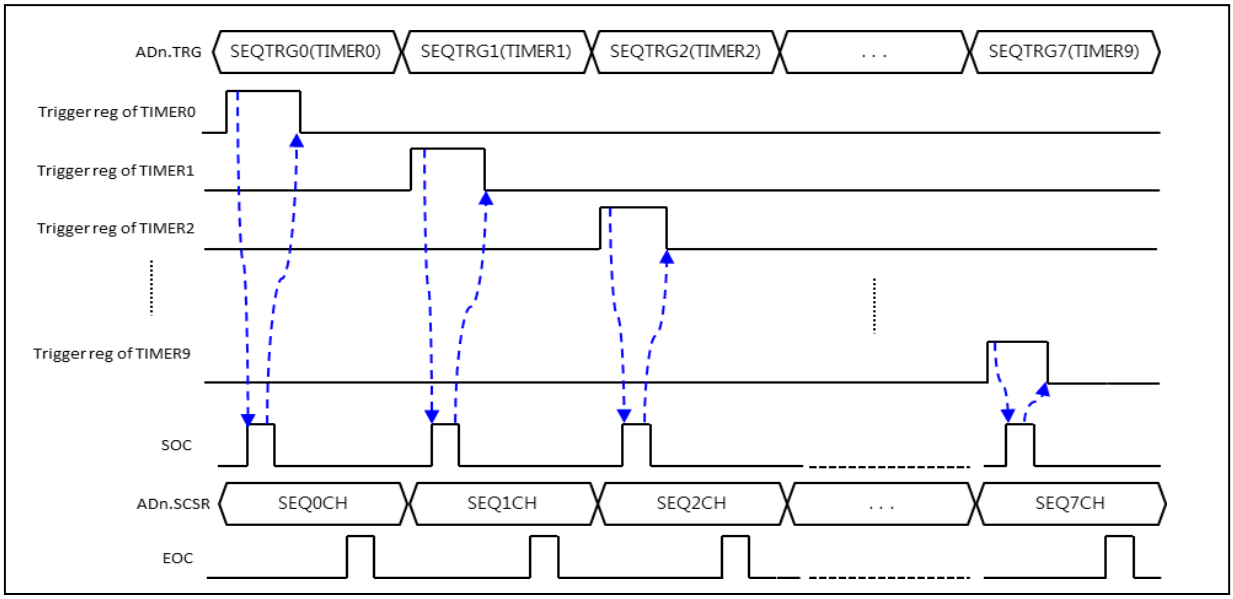


Figure 16.7. ADC Trigger Timing in Sequential Mode (SEQCNT = 3'b111, 8 Sequence Conversion)

17. 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 17.1 Absolute Maximum Ratings

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}		2.5	mA
Output Low Current Total	$\sum I_{OL}$		25	mA
Output High Current per pin	I_{OH}		– 2.5	mA
Output Low Current Total	$\sum I_{OH}$		25	mA
Input Main Clock Range		0.4	8	MHz
Operating Frequency		-	48	MHz
Storage Temperature	Tst	-55	+125	℃
Operating Temperature	Top	-40	+85	℃

DC Characteristics

Table 17.2 Recommended Operating Conditions

Parameter	Symbol	Condition	Min	Typical	Max	Unit
Supply Voltage	VDD		3.0		5.5	V
Supply Voltage	AVDD		3.0	5.0	5.5	V
Operating Frequency	FREQ	MOSC	4		8	MHz
		INTOSC		1		MHz
		PLL	4		80	MHz
Operating Temperature	Top	Top	-40		+85	℃

Table 17.3 DC Electrical Characteristics (VDD = +5V, Ta = 25℃)

Parameter	Symbol	Condition	Min	Typical	Max	Unit
Input Low Voltage	VIL	Schmitt input	-	-	0.2VDD	V
Input High Voltage	VIH	Schmitt input	0.8VDD	-	-	V
Output Low Voltage	VOL	IOL = 10mA	-	-	VSS+1	V
Output High Voltage	VOH	IOH = -3mA	VDD–1	-	-	V
Output Low Current	IOL		-	-	3	mA
Output High Current	IOH		-3	-		mA
Input High Leakage	IIH				4	uA
Input Low Leakage	IIL		-4			
Pull-up Resister	RPU	Rmax:VDD=3.0V Rmin:VDD=5V	30	-	70	kΩ

Current Consumption

Table 17.4. Current Consumption in Each Mode (Temperature: +25℃)

Parameter	Symbol	Condition	Min	Typical	Max	Unit
Normal Operation	IDD_{NORMAL}	ROSC=RUN MXOSC=8MHz HCLK=48MHz	20	20	-	mA
Sleep Mode	IDD_{SLEEP}	ROSC=RUN MXOSC=STOP HCLK =48MHz	-	8.3	-	mA

Note: UART en, 1 port toggle @5V

POR Electrical Characteristics

Table 17.5 POR Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typical	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 17.6. LVD Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typical	Max	Unit
Operating Voltage	VDD		1.7		5	V
Operating Current	IDD_{LVD}	Typ. <6uA when always on	-	1	-	mA
LVD Set Level 0	VLVD0	VDD falling (slow)	1.6	1.8	2.0	V
LVD Set Level 1	VLVD1	VDD falling (slow)	2.0	2.2	2.5	V
LVD Set Level 2	VLVD2	VDD falling (slow)	2.5	2.7	3.0	V
LVD Set Level 3	VLVD3	VDD falling (slow)	3.9	4.3	4.6	V

VDC Electrical Characteristics

Table 17.7 VDC Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typical	Max	Unit
Operating Voltage	VDD_{VDC}		3.0	-	5.5	V
VDC Output Voltage	$VOUT_{VDC}$	@RUN	1.62	1.8	1.98	V
		@STOP	1.4	1.8	2.0	V
Regulation Current	I_{OUT}				100	mA
Drop-out Voltage	$VDROP_{VDC}$	VDDVDC=3.0V IOUT=100mA	-	-	200	mV
Current Consumption	IDD_{NORM}	@RUN	-	100	150	uA
	IDD_{STOP}	@STOP	-	1	2	uA

External OSC Characteristics

Table 17.8 External OSC Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Voltage	VDD		3.0	-	5.5	V
IDD		@4MHz/5V	-	240		uA
Frequency	$OSCF_{req}$		4	8	10	MHz
Output Voltage	OSC_{VOUT}		1.2	2.4	-	V
Load Capacitance	$LOAD_{CAP}$		5	22	35	pF

PLL Electrical Characteristics

Table 17.9 PLL Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Operating Voltage	VDD_{PLL}		3.0		5.5	V
Output Frequency	FOUT		4		48	MHz
Operating Current	IDD_{PLL}	@50MHz		1.3		mA
Duty	$FOUT_{DUTY}$		40	-	60	%
P-P Jitter	JITTER	@Lock			500	Ps
VCO	VCO		20		80	MHz
Input Frequency	FIN		4		8	MHz
Locking time	LOCK				1	ms

ADC Electrical Characteristics

Table 17.10. ADC Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typical	Max	Unit
Operating Voltage	AVDD		3.0	5	5.5	V
Reference Voltage	AVREF		3.0	5	5.5	V
Resolution				12		Bit
Operating Current	IDDA				2.8	mA
Analog Input Range			0		AVDD	V
Conversion Rate				-	1.6	Msps
Operating Frequency	ACLK				25	MHz
DC Accuracy	INL			±2.5		LSB
	DNL			±1.0		LSB
Offset Error				±1.5		LSB
Full Scale Error				±1.5		LSB
SNDR	SNDR			68		dB
THD				-70		dB

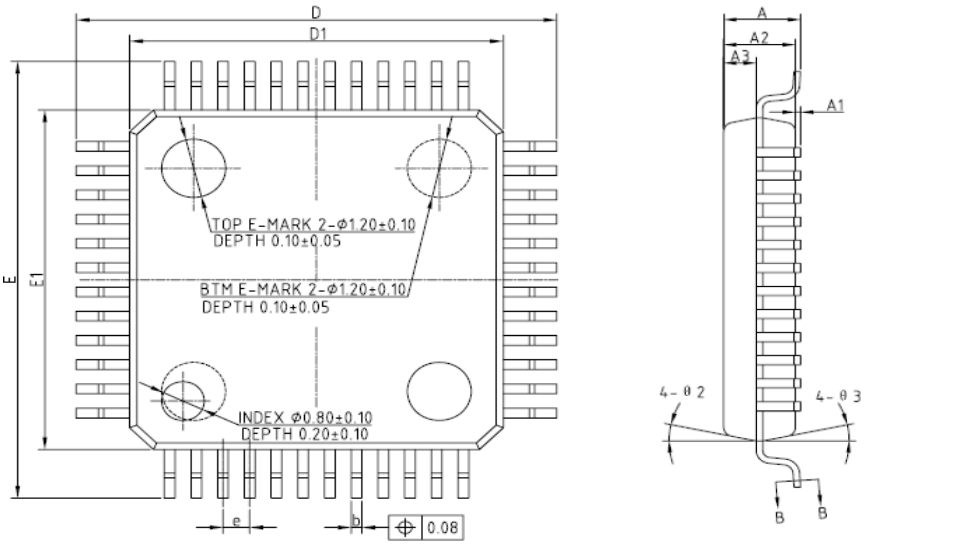
DNL: Maximum deviation between actual steps and the ideal one.

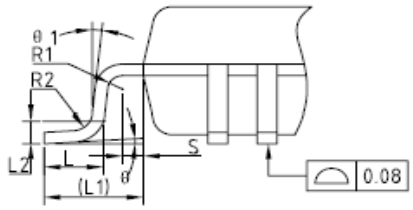
INL: Integral Linearity Error: maximum deviation between any actual transition and the end point

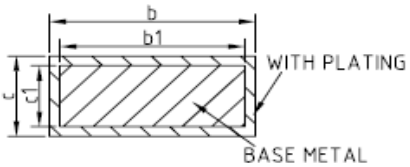
18. Package

Figure 18.1 displays the LQFP-48 package dimension and Figure 18.2 shows the LQFP-32 package dimension.

LQFP-48 Package







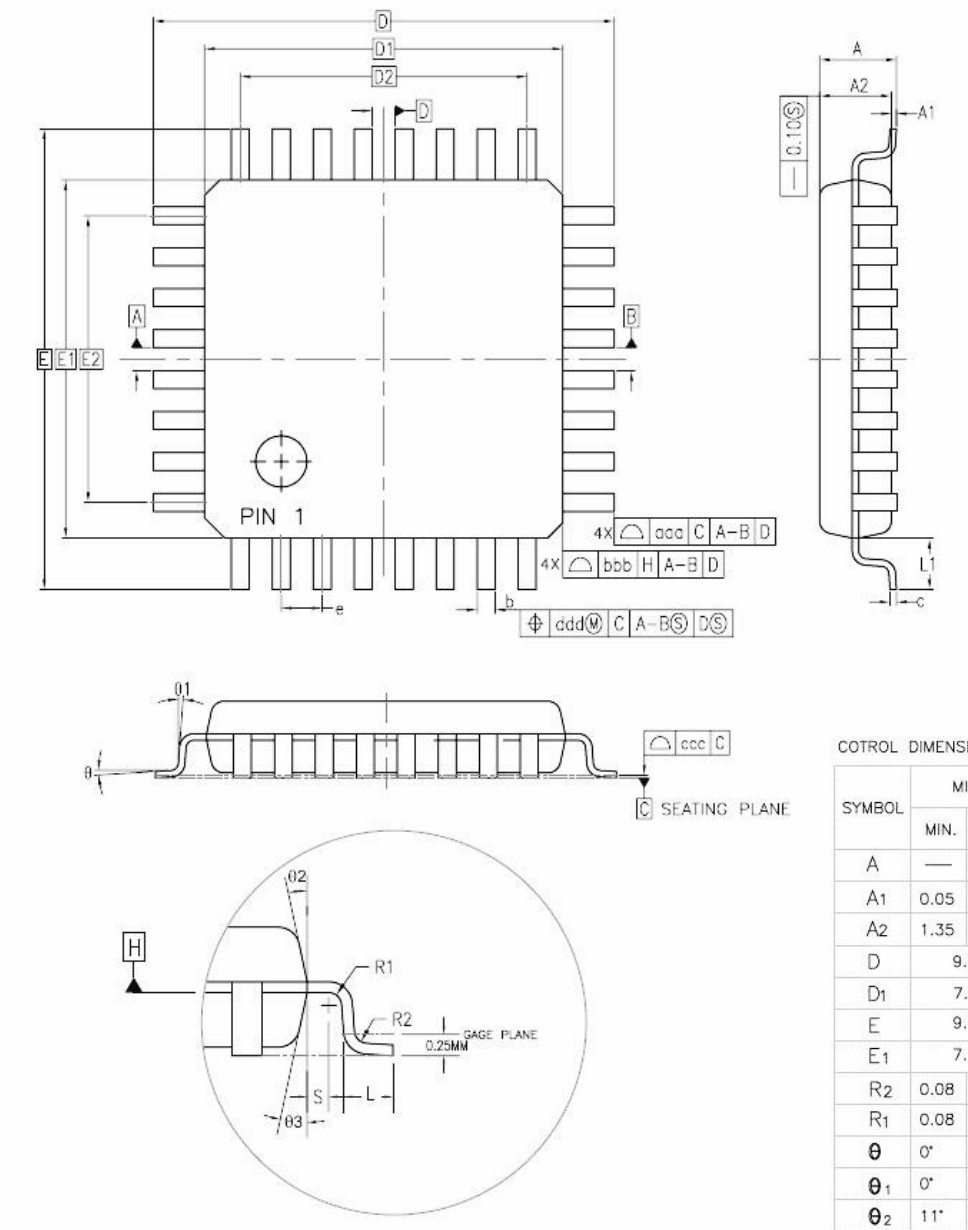
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
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
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

Figure 18.1 Package Dimension (LQFP-48 7X7)

LQFP-32 Package



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	0.30	0.35	0.45	0.012	0.014	0.018
e	0.80 BSC.			0.031 BSC.		
D2	5.60			0.220		
E2	5.60			0.220		

NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

Figure 18.2 Package Dimension (LQFP-32 7X7)

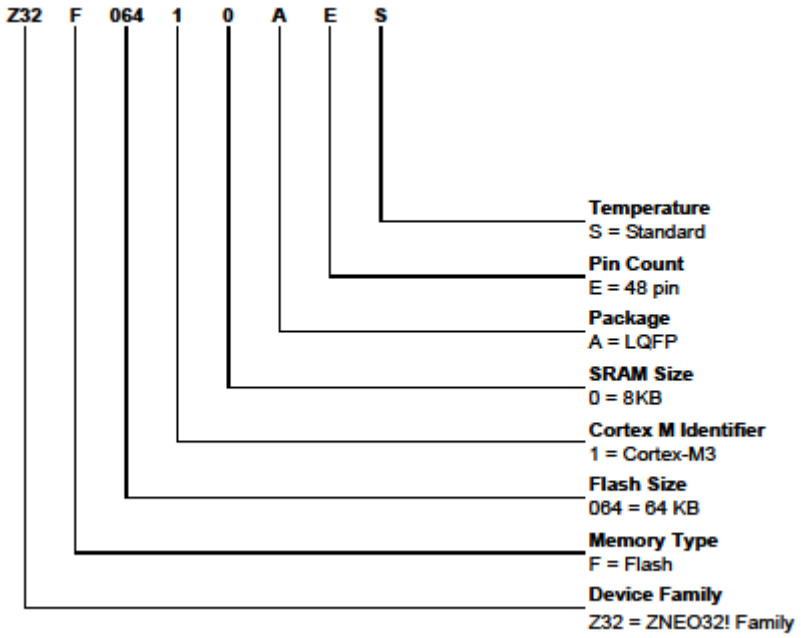
19. Ordering Information

Table 19.1 identifies the basic features and package styles available for the Z32F0641 MCU.

Table 19.1 Ordering Information for the Z32F0641 MCU

Device	Flash Size	SRAM	UART	SPI	I^2C	MPWM	ADC	I/O Ports	Package
Z32F06410AES	64 KB	8 KB	2	1	1	1	2-unit 11 channel	44	LQFP-48
Z32F06410AKS	64 KB	8 KB	2	1	1	1	2-unit 8 channel	30	LQFP-32

Zilog part numbers consist of a number of components, which are described below using part number Z32F06410AES as an example.



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

ZiLOG:
Z32F06410AKS