Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ARM® Cortex®-M0
Core Size	32-Bit Single-Core
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, Motor Control PWM, POR, PWM, WDT
Number of I/O	44
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/z32f06423aee

3. Boot Mode

Boot Mode Pins

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

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

Table 3-1 Boot Mode Pins

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

ISP Mode Connections

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

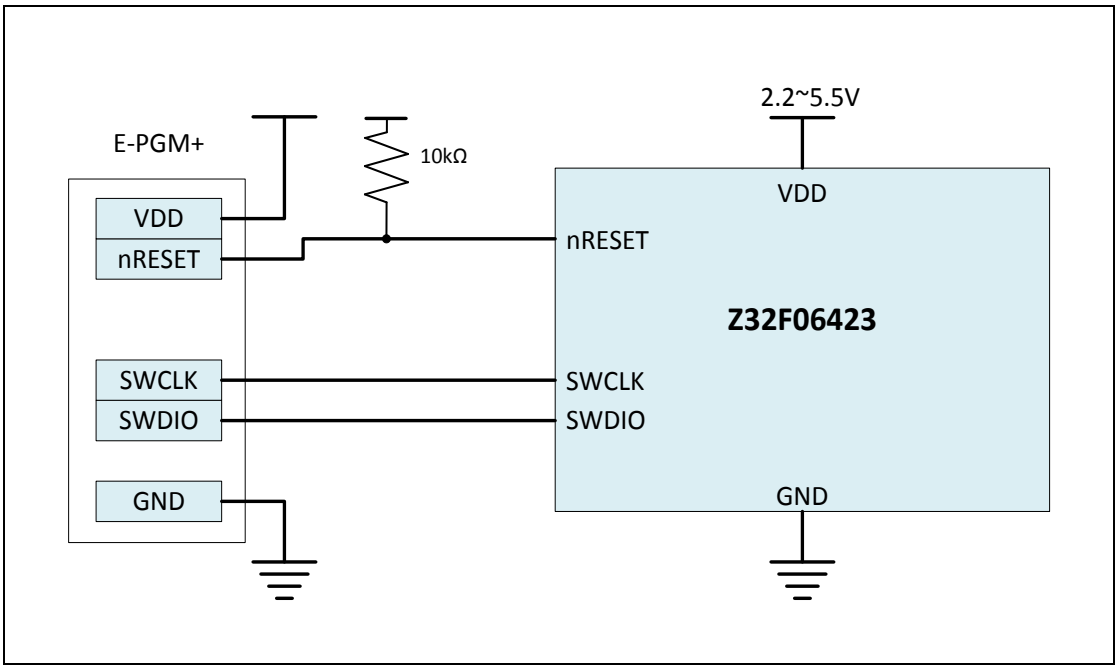

Figure 3-3 ISP and E-PGM+ Connection Diagram

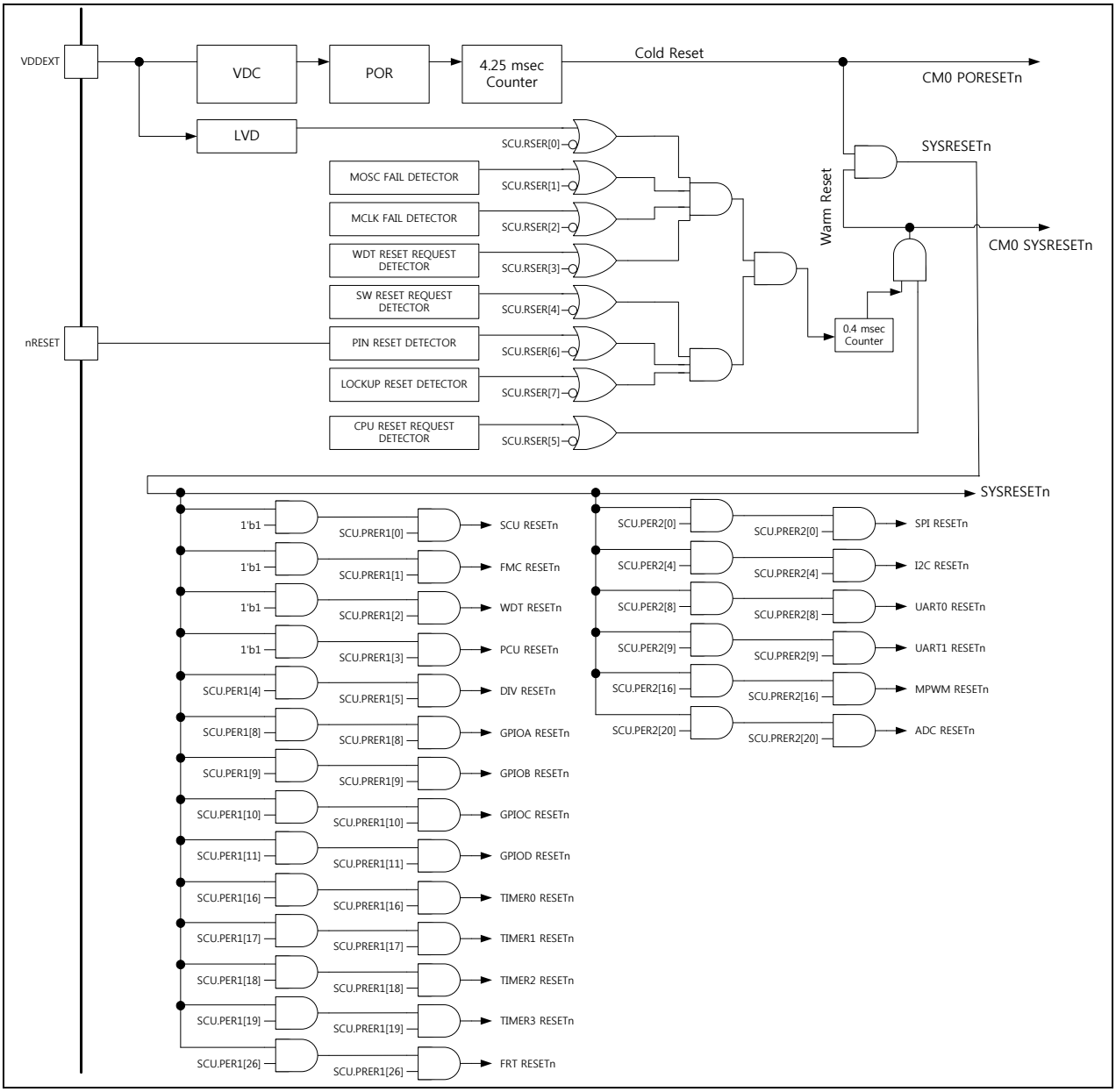

Figure 4-7 Reset Tree Configuration

RSER Reset Source Enable Register

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

RSER=0x4000_0018

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

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

MCCR1 Miscellaneous Clock Control Register 1

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

MCCR1=0x4000_0090

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

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

MCCR2 Miscellaneous Clock Control Register 2

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

MCCR2=0x4000_0094

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

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

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

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

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

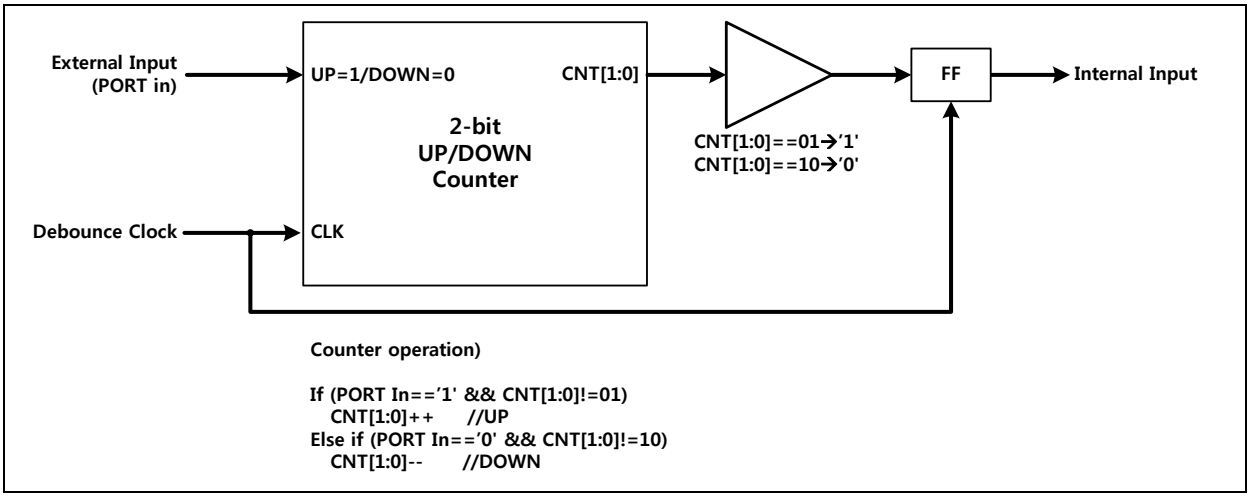

Figure 5-5. Debounce Logic

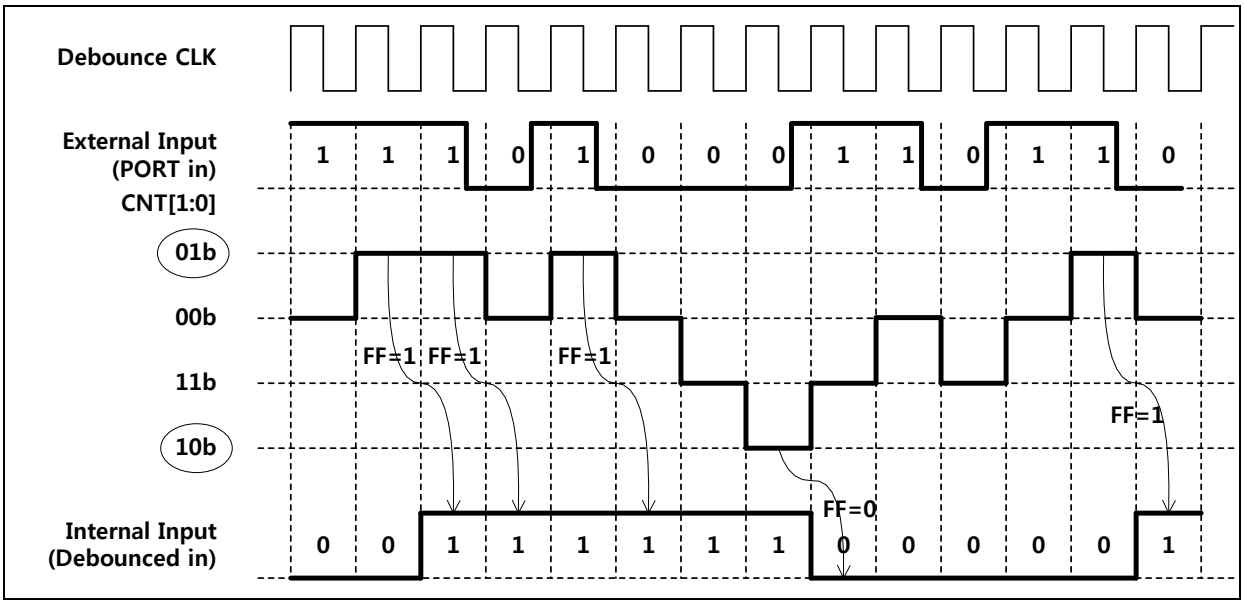

Figure 5-6. Port Debounce Example

Pn.BSR PORT n Bit Set Register

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

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

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

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

Pn.BCR PORT n Bit Clear Register

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

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

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

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

Registers

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

Table 7-1 Flash Memory Controller Base Address

NAME	BASE ADDRESS
Flash Controller	0x4000_0100

Table 7-2 shows the Register memory map.

Table 7-2 FMC Register Map

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

9. Watch-Dog Timer (WDT)

Overview

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

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

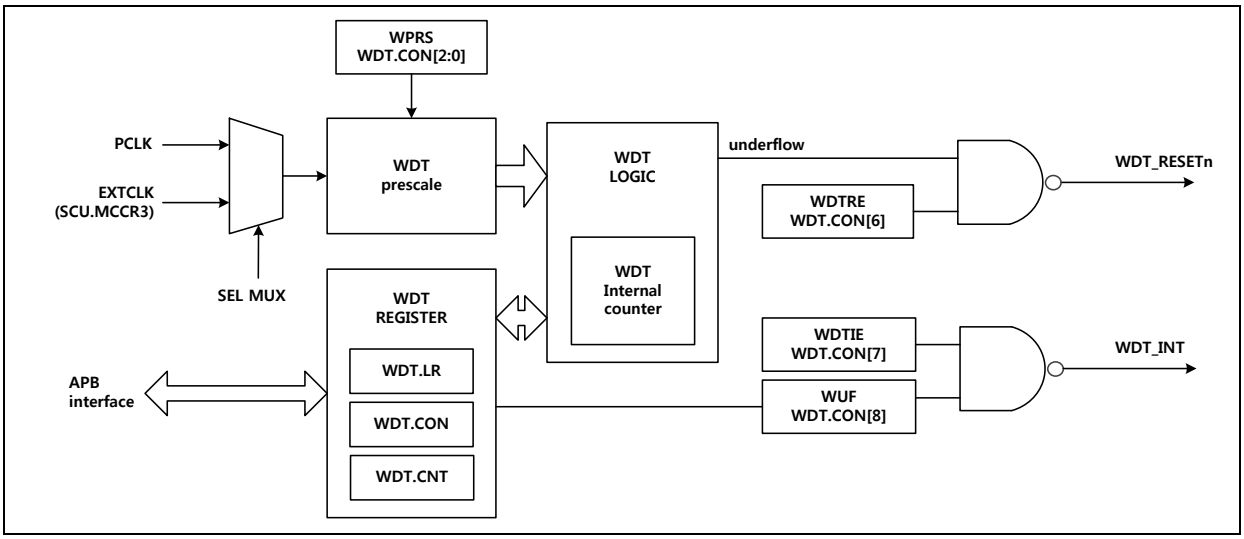

Figure 9-1 WDT Block Diagram

Registers

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

Table 9-1 Base Address of SCU

NAME	BASE ADDRESS
WDT	0x4000_0200

Table 9-2 Watchdog Timer Register Map

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

Functional Description

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

Timing Diagram

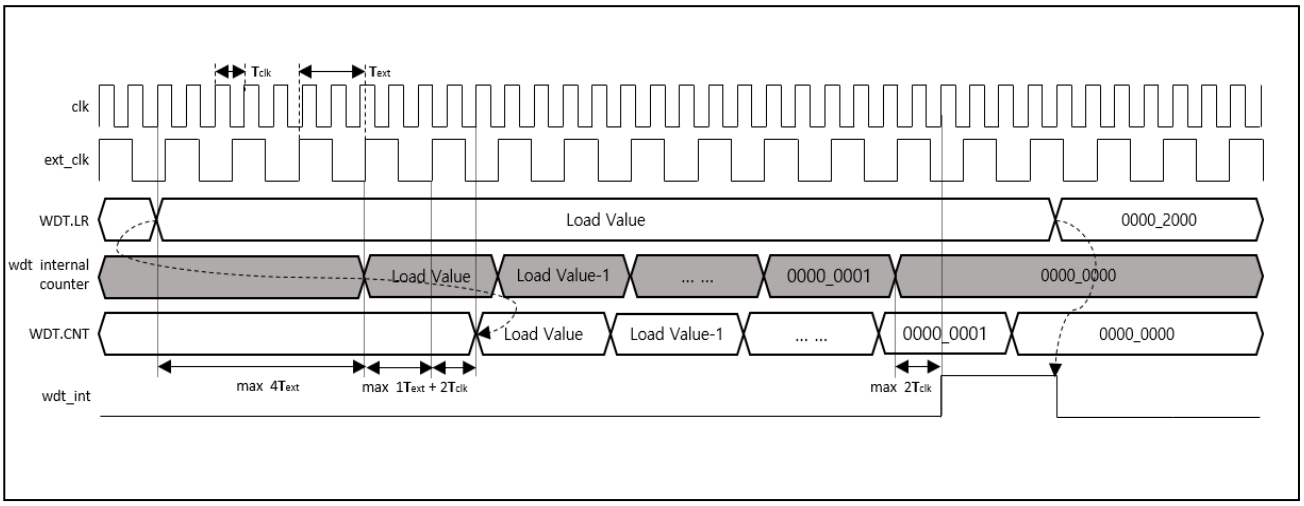

Figure 9-2 Timing Diagram in Interrupt Mode Operation when WDT Clock is External Clock

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

Prescale Table

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

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

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

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

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

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

Registers

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

Table 12-2 Base Address of Each Port

NAME	BASE ADDRESS
U0	0x4000_8000
U1	0x4000_8100

Table 12-3 UART Register Map

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

Un.RBR Receive Buffer Register

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

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

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

7 0	RBR	Receive Buffer Register

Un.BFR Baud Rate Fraction Counter Register

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

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

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

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

Table 12-7 Example of Baud Rate Calculation

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

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

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

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

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

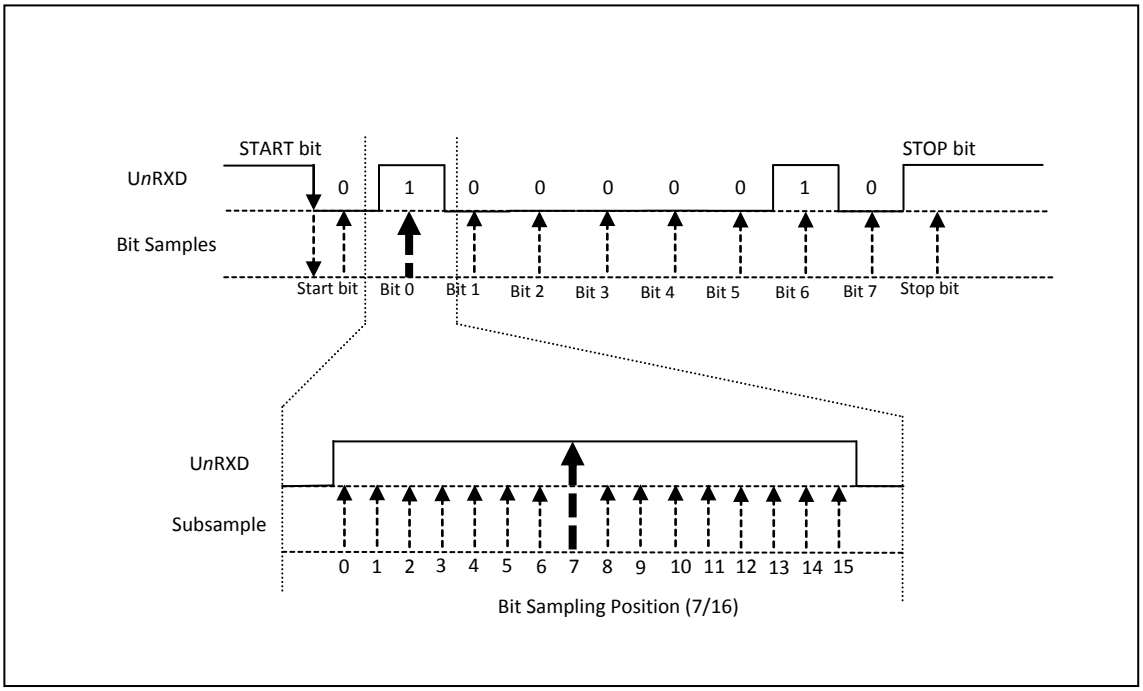

Figure 12-3 Sampling Timing of UART Receiver

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

Transmitter

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

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

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

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

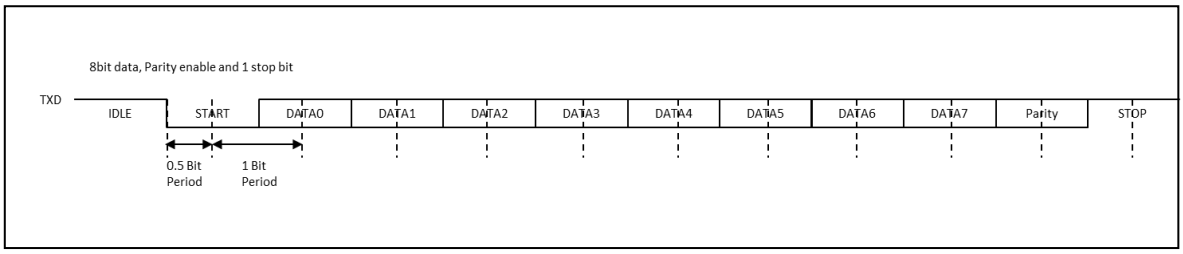

Figure 12-4 Transmit Data Format Example

Inter-frame Delay Transmission

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

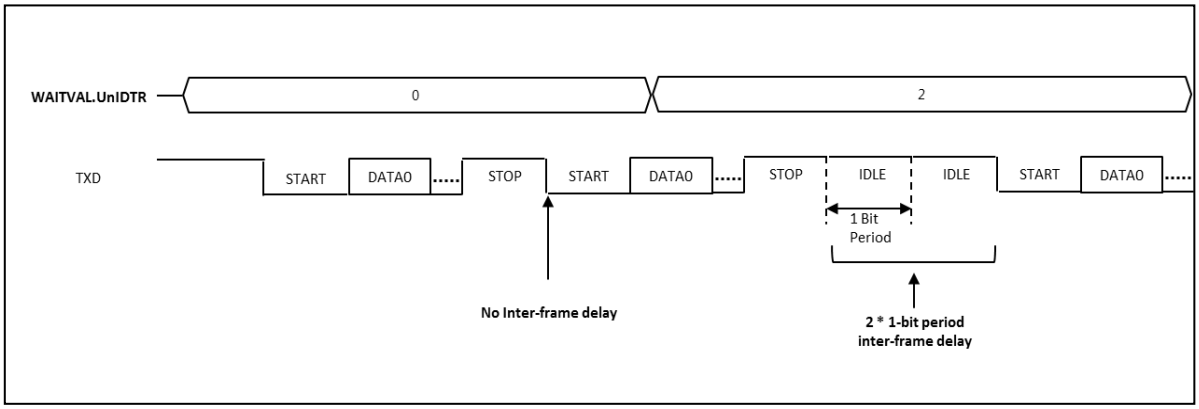

Figure 12-5 Inter-frame Delay Timing Diagram

Transmit Interrupt

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

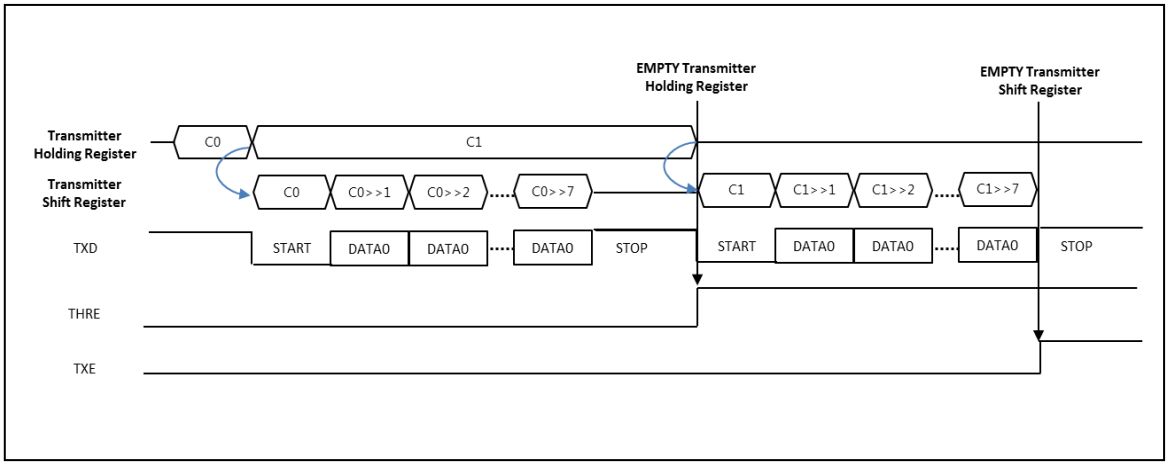

Figure 12-6 Transmit Interrupt Timing Diagram

Pin Description

Table 14-1 I²C Interface External Pins

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

Registers

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

Table 14-2 I²C Interface Base Address

NAME	BASE ADDRESS
I²C	0x4000_A000

Table 14-3 I²C Register Map

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

MP.DWL MPWM Duty WL Register

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

MP.DWL=0x4000_4024

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

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

MP.IER MPWM Interrupt Enable Register

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

MP.IER=0x4000_4034

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

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

Motor PWM 1-Channel Asymmetric Mode Timing

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

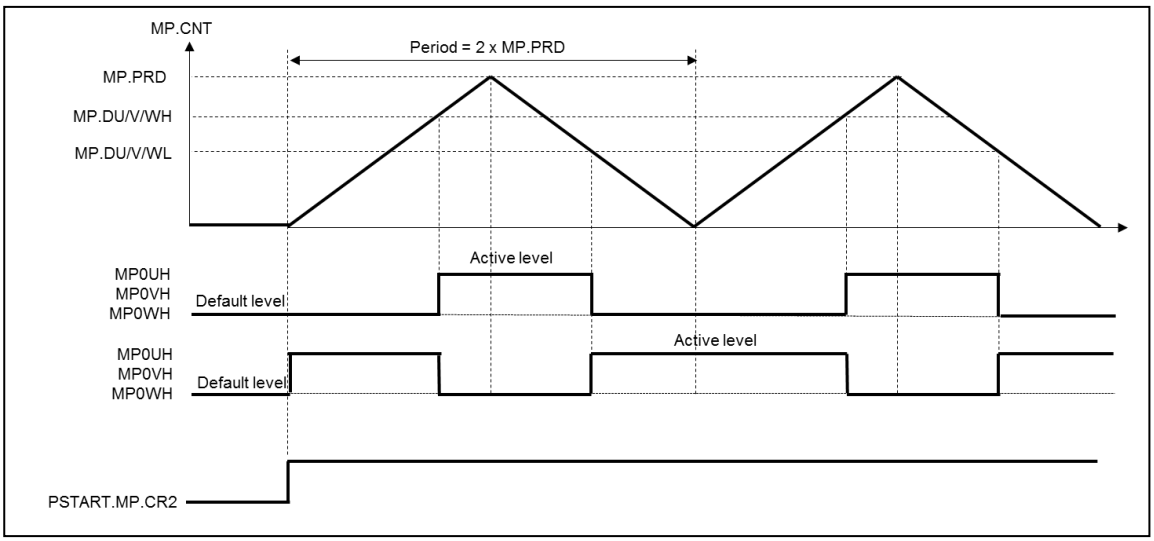

Figure 15.2 1-Channel Asymmetric Mode Waveform (MOTORB=0, MCHMOD=01)

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

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

Symmetrical Mode vs Asymmetrical Mode

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

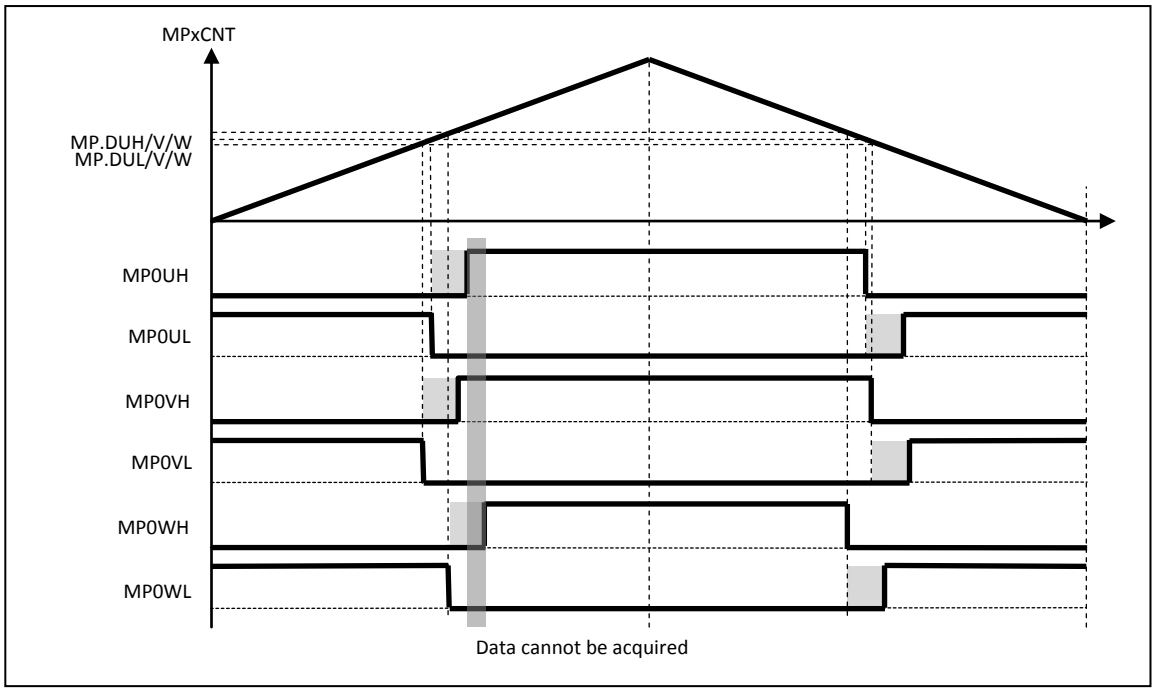

Figure 15-14 Symmetrical PWM Timing

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

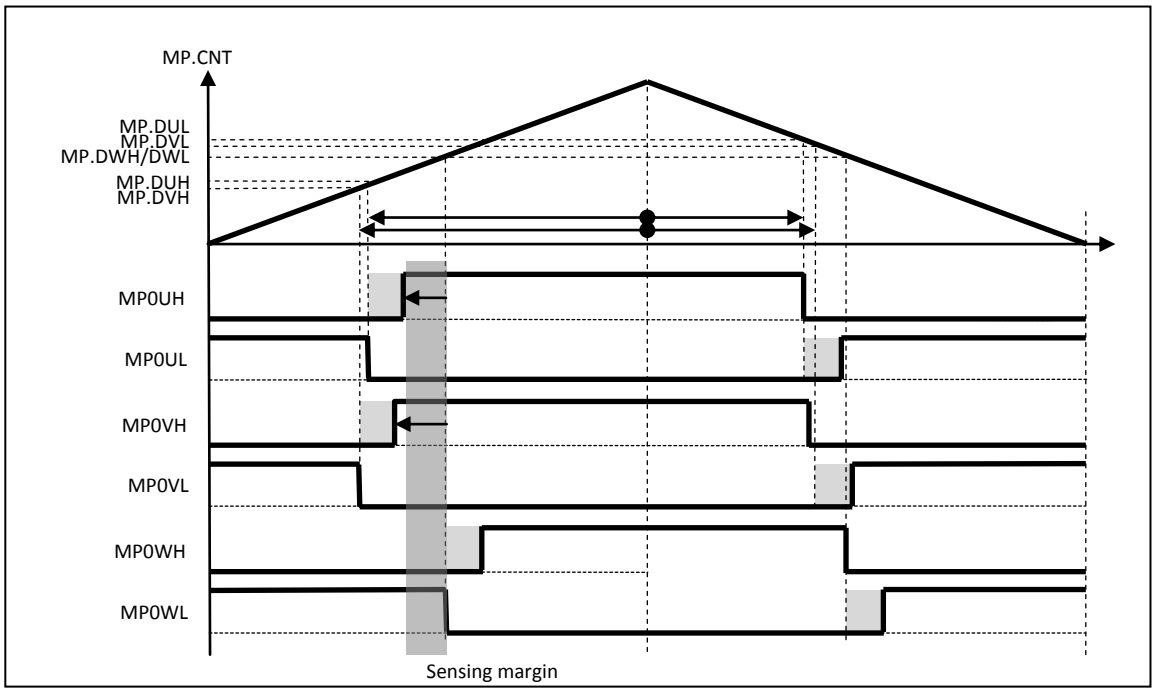

Figure 15-15 Asymmetrical PWM Timing and Sensing Margin

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

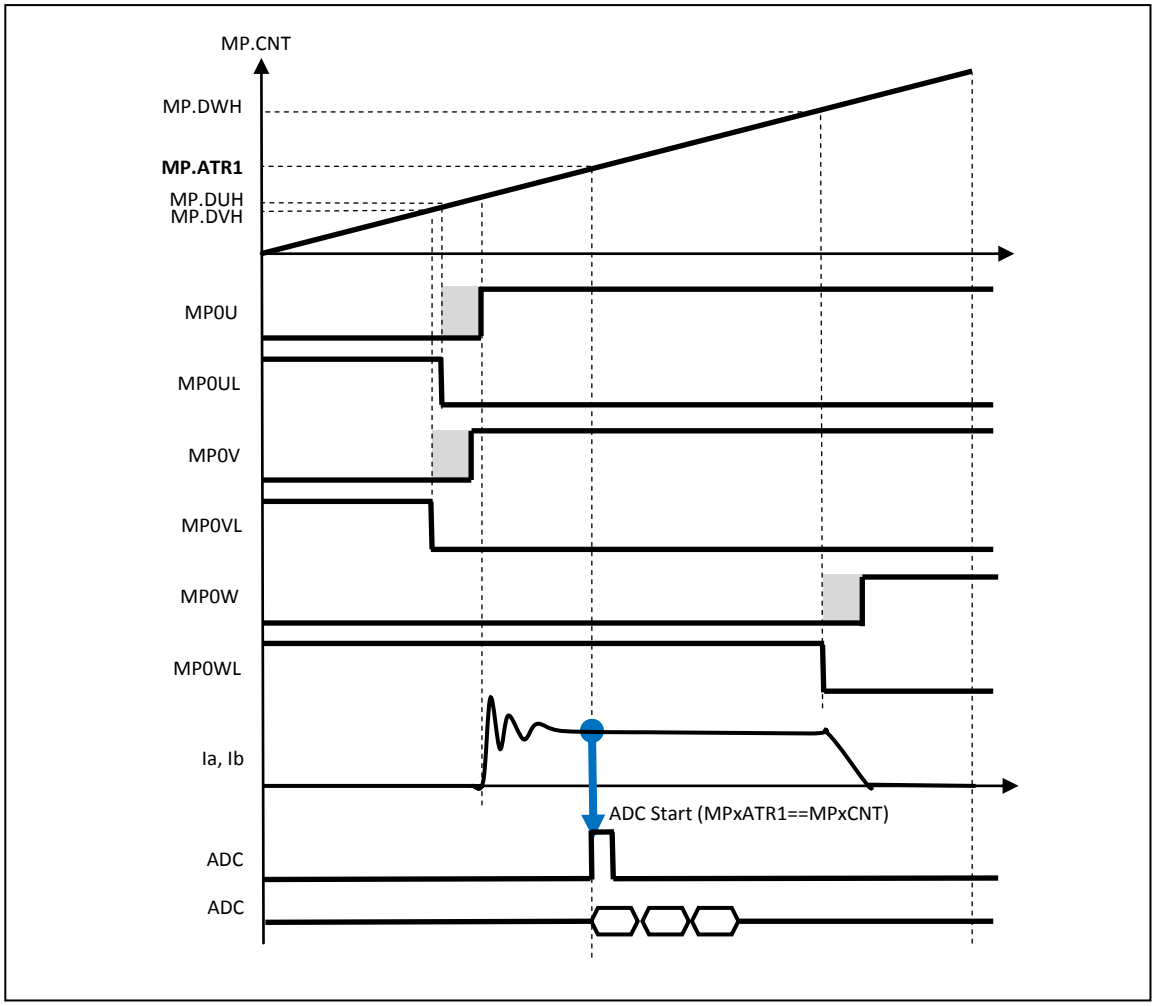

Figure 15-17 An Example of ADC Acquisition Timing by Event from MPWM

ADC Sequential Conversion Mode Timing Diagram

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

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

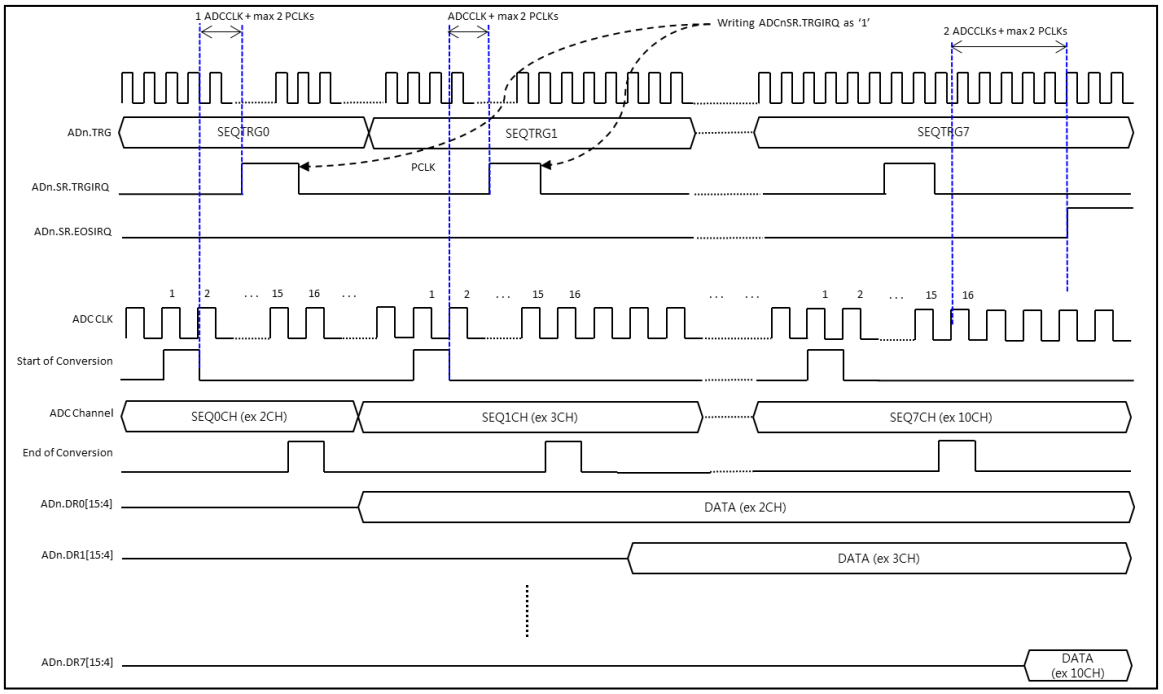

Figure 17-5 ADC Sequential Mode Timing (When AD.MR.AMOD = ó0 and AD.MR.SEQCNT Í ó0ô

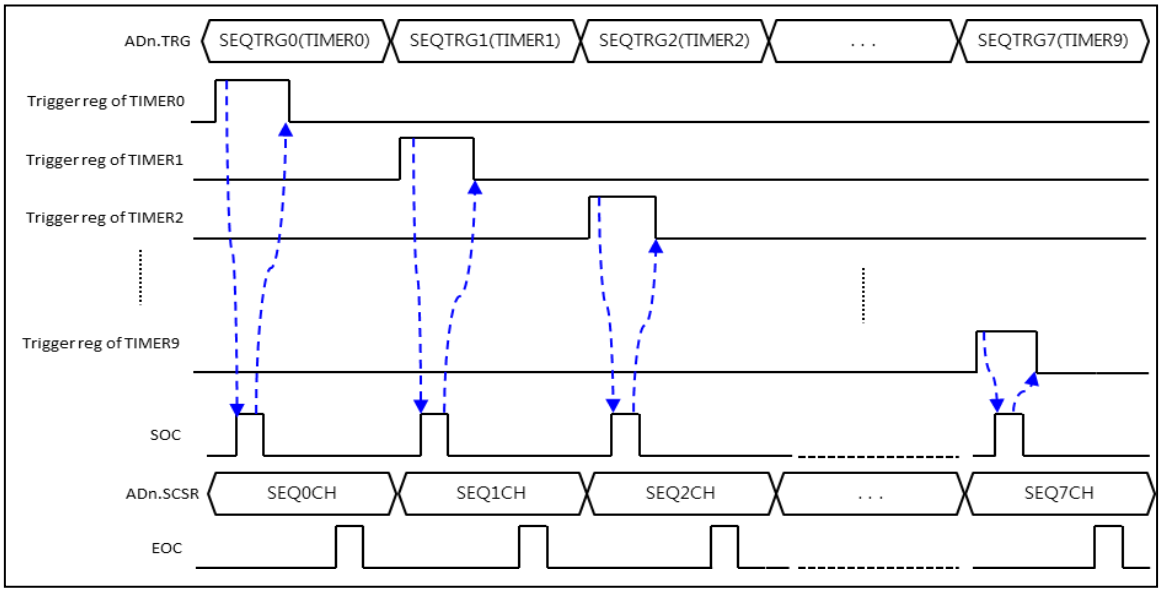

Figure 17-6 ADC Trigger Timing in Sequential Mode (SEQCNT = 3ô111, 8 Sequence Coversion)