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	Active
Core Processor	ARM® Cortex®-M0
Core Size	32-Bit Single-Core
Speed	72MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVR, I²S, PWM
Number of I/O	53
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 15x12b
Oscillator Type	External, Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/m0564sg4ae

List of Tables

2 FEATURES

2.1 NuMicro® M0564 Features

- Core
  - ARM® Cortex®-M0 core running up to 72 MHz
  - One 24-bit system timer
  - Supports low power sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Supports programmable mask-able interrupts
  - Serial Wire Debug supports with 2 watch-points/4 breakpoints
- Built-in LDO for wide operating voltage ranged from 2.5V to 5.5V
- Flash Memory
  - Supports 256/128 KB application ROM (APROM)
  - Supports 4 KB Flash for loader (LDROM)
  - Supports 2 KB Security Protection Rom (SPROM)
  - Supports 12 bytes User Configuration block to control system initiation
  - Supports Data Flash with configurable memory size
  - Supports 2 KB page erase for all embedded flash
  - Supports In-System-Programming (ISP), In-Application-Programming (IAP) update embedded flash memory
  - Supports CRC-32 checksum calculation function
  - Supports flash all one verification function
  - Hardware external read protection of whole flash memory by Security Lock Bit
  - Supports 2-wired ICP update through SWD/ICE interface
- SRAM Memory
  - 20 KB embedded SRAM
  - Supports byte-, half-word- and word-access
  - Supports PDMA mode
- Hardware Divider
  - Signed (two's complement) integer calculation
  - 32-bit dividend with 16-bit divisor calculation capacity
  - 32-bit quotient and 32-bit remainder outputs (16-bit remainder with sign extends to 32-bit)
  - Divided by zero warning flag
  - 6 HCLK clocks taken for one cycle calculation
  - Write divisor to trigger calculation
  - Waiting for calculation ready automatically when reading quotient and remainder
- PDMA (Peripheral DMA)
  - Supports 5 independent configurable channels for automatic data transfer between memories and peripherals
  - Supports single and burst transfer type
  - Supports Normal and Scatter-Gather Transfer modes
  - Supports two types of priorities modes: Fixed-priority and Round-robin modes
  - Supports byte-, half-word- and word-access
  - Supports incrementing mode for the source and destination address for each channel
  - Supports time-out function for channel 0 and channel 1
  - Supports software and SPI/I2S, UART, USCI, ADC, PWM and TIMER request
- Clock Control

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
			EBI_AD13	I/O	MFP7	EBI address/data bus bit 13.
26	34	51	PC.6	I/O	MFP0	General purpose digital I/O pin.
			USCI0_DAT1	I/O	MFP4	USCI0 data 1 pin.
			ACMP1_O	O	MFP5	Analog comparator 1 output pin.
			PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
			EBI_AD14	I/O	MFP7	EBI address/data bus bit 14.
27	35	52	PC.7	I/O	MFP0	General purpose digital I/O pin.
			USCI0_CTL1	I/O	MFP4	USCI0 control 1 pin.
			PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
			EBI_AD15	I/O	MFP7	EBI address/data bus bit 15.
28		53	PE.4	I/O	MFP0	General purpose digital I/O pin.
			I2C0_SCL	I/O	MFP2	I2C0 clock pin.
			I2C1_SCL	I/O	MFP3	I2C1 clock pin.
			USCI0_CTL0	I/O	MFP4	USCI0 control 0 pin.
			SC0_PWR	O	MFP5	Smart Card 0 power pin.
			PWM1_BRAKE0	I	MFP6	PWM1 Brake 0 input pin.
			EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
			INT0	I	MFP8	External interrupt 0 input pin.
29		54	PE.5	I/O	MFP0	General purpose digital I/O pin.
			I2C0_SDA	I/O	MFP2	I2C0 data input/output pin.
			I2C1_SDA	I/O	MFP3	I2C1 data input/output pin.
			USCI0_CLK	I/O	MFP4	USCI0 clock pin.
			SC0_RST	O	MFP5	Smart Card 0 reset pin.
			PWM1_BRAKE1	I	MFP6	PWM1 Brake 1 input pin.
			EBI_ALE	O	MFP7	EBI address latch enable output pin.
			INT1	I	MFP8	External interrupt 1 input pin.
30	36	55	PE.6	I/O	MFP0	General purpose digital I/O pin.
			ICE_CLK	I	MFP1	Serial wired debugger clock pin.
			I2C0_SCL	I/O	MFP2	I2C0 clock pin.
			UART0_RXD	I	MFP3	UART0 data receiver input pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			EBI_nWRL	O	MFP7	EBI low byte write enable output pin.
			INT1	I	MFP8	External interrupt 1 input pin.
44	58	92	PB.1	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH1	A	MFP1	ADC0 channel 1 analog input.
			VDET_P1	A	MFP2	Voltage detector positive input 1 pin.
			UART2_TXD	O	MFP3	UART2 data transmitter output pin.
			TM3	I/O	MFP4	Timer3 event counter input/toggle output pin.
			SC0_RST	O	MFP5	Smart Card 0 reset pin.
			PWM0_SYNC_OUT	O	MFP6	PWM0 counter synchronous trigger output pin.
			EBI_nWRH	O	MFP7	EBI high byte write enable output pin
			USCI1_DAT1	I/O	MFP8	USCI1 data 1 pin.
45	59	93	PB.2	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH2	A	MFP1	ADC0 channel 2 analog input.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
			SPI1_CLK	I/O	MFP3	SPI1 serial clock pin.
			UART1_RXD	I	MFP4	UART1 data receiver input pin.
			SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
			TM_BRAKE0	I	MFP6	Timer Brake 0 input pin.
			EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
			USCI0_DAT0	I/O	MFP8	USCI0 data 0 pin.
46	60	94	PB.3	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH3	A	MFP1	ADC0 channel 3 analog input.
			SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
			SPI1_MISO	I/O	MFP3	SPI1 MISO (Master In, Slave Out) pin.
			UART1_TXD	O	MFP4	UART1 data transmitter output pin.
			TM_BRAKE1	I	MFP6	Timer Brake 1 input pin.
			EBI_ALE	O	MFP7	EBI address latch enable output pin.
			USCI0_DAT1	I/O	MFP8	USCI0 data 1 pin.
47	61	95	PB.4	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH4	A	MFP1	ADC0 channel 4 analog input.
			SPI0_SS	I/O	MFP2	SPI0 slave select pin.

6.2.2.2 Power-on Reset (POR)

The Power-on reset (POR) is used to generate a stable system reset signal and forces the system to be reset when power-on to avoid unexpected behavior of MCU. When applying the power to MCU, the POR module will detect the rising voltage and generate reset signal to system until the voltage is ready for MCU operation. At POR reset, the PORF(SYS_RSTSTS[0]) will be set to 1 to indicate there is a POR reset event. The PORF(SYS_RSTSTS[0]) bit can be cleared by writing 1 to it. Figure 6.2-3 shows the power-on reset waveform.

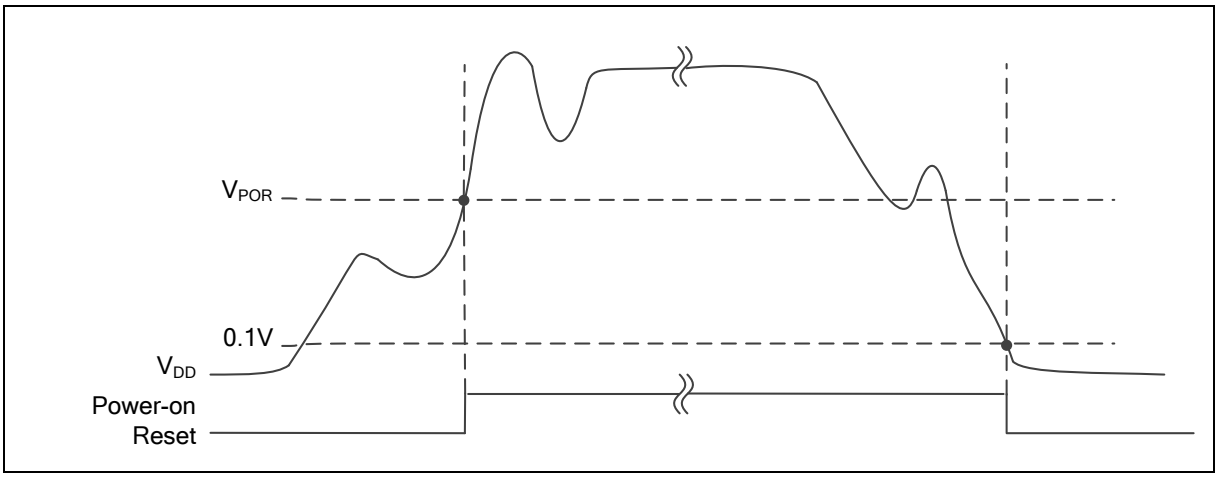


Figure 6.2-3 Power-on Reset (POR) Waveform

6.2.2.3 Low Voltage Reset (LVR)

If the Low Voltage Reset function is enabled by setting the Low Voltage Reset Enable Bit LVREN (SYS_BODCTL[7]) to 1, after 200us delay, LVR detection circuit will be stable and the LVR function will be active. Then LVR function will detect AV_{DD} during system operation. When the AV_{DD} voltage is lower than V_{LVR} and the state keeps longer than De-glitch time set by LVRDGSEL (SYS_BODCTL[14:12]), chip will be reset. The LVR reset will control the chip in reset state until the AV_{DD} voltage rises above V_{LVR} and the state keeps longer than De-glitch time set by LVRDGSEL (SYS_BODCTL[14:12]). The default setting of Low Voltage Reset is enabled without De-glitch function. Figure 6.2-4 shows the Low Voltage Reset waveform.

The MCU Reset is similar with CHIP Reset. The difference is that BS(FMC_ISPCTL[1]) will not be reloaded from CONFIG0 setting and keep its original software setting for booting from APROM or LDROM. User can set the SYSRESETREQ(AIRCR[2]) to 1 to assert the MCU Reset.

6.2.3 Power Modes and Wake-up Sources

There are several wake-up sources in Idle mode and Power-down mode. Table 6.2-2 lists the available clocks for each power mode.

Power Mode	Normal Mode	Idle Mode	Power-Down Mode
Definition	CPU is in active state	CPU is in sleep state	CPU is in sleep state and all clocks stop except LXT and LIRC. SRAM content retended.
Entry Condition	Chip is in normal mode after system reset released	CPU executes WFI instruction.	CPU sets sleep mode enable and power down enable and executes WFI instruction.
Wake-up Sources	N/A	All interrupts	RTC, WDT, I²C, Timer, UART, BOD, GPIO, EINT, USCI, ACMP and VDET.
Available Clocks	All	All except CPU clock	LXT and LIRC
After Wake-up	N/A	CPU back to normal mode	CPU back to normal mode

Table 6.2-2 Power Mode Difference Table

WWDT	ON	ON	Halt
RTC	ON	ON	ON/OFF[5]
UART	ON	ON	ON/OFF[6]
SC	ON	ON	Halt
USCI	ON	ON	Halt
I^2C	ON	ON	Halt
SPI	ON	ON	Halt
ADC	ON	ON	Halt
ACMP	ON	ON	Halt

Table 6.2-3 Clocks in Power Modes

Wake-up sources in Power-down mode:

RTC, WDT, I²C, Timer, UART, USCI, BOD, VDET, GPIO, and ACMP.

After chip enters power down, the following wake-up sources can wake chip up to normal mode. Table 6.2-4 lists the condition about how to enter Power-down mode again for each peripheral.

*User needs to wait this condition before setting PDEN(CLK_PWRCTL[7]) and execute WFI to enter Power-down mode.

Wake-Up Source	Wake-Up Condition	System Can Enter Power-Down Mode Again Condition*
BOD	Brown-Out Detector Interrupt	After software writes 1 to clear BODIF (SYS_BODCTL[4]).
VDET	Voltage Detector Interrupt	After software writes 1 to clear VDETIF (SYS_BODCTL[19]).
GPIO	GPIO Interrupt	After software write 1 to clear the Px_INTSRC[n] bit.
TIMER	Timer Interrupt	After software writes 1 to clear TWKF (TIMERx_INTSTS[1]) and TIF (TIMERx_INTSTS[0]).
WDT	WDT Interrupt	After software writes 1 to clear WKF (WDT_CTL[5]) (Write Protect).
RTC	Alarm Interrupt	After software writes 1 to clear ALMIF (RTC_INTSTS[0]).
	Time Tick Interrupt	After software writes 1 to clear TICKIF (RTC_INTSTS[1]).
UART	nCTS wake-up	After software writes 1 to clear CTSWKF (UARTx_WKSTS[0]).
	RX Data wake-up	After software writes 1 to clear DATWKF (UARTx_WKSTS[1]).
	Received FIFO Threshold Wake-up	After software writes 1 to clear RFRTWKF (UARTx_WKSTS[2]).
	RS-485 AAD Mode Wake-up	After software writes 1 to clear RS485WKF (UARTx_WKSTS[3]).
	Received FIFO Threshold Time-out Wake-up	After software writes 1 to clear TOUTWKF (UARTx_WKSTS[4]).
USCI UART	CTS Toggle	After software writes 1 to clear WKF (UUART_WKSTS[0]).
	Data Toggle	After software writes 1 to clear WKF (UUART_WKSTS[0]).
USCI I^2C	Data toggle	After software writes 1 to clear WKF (UI2C_WKSTS[0]).

6.3 Clock Controller

6.3.1 Overview

The clock controller generates clocks for the whole chip, including system clocks and all peripheral clocks. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and a clock divider. The chip will not enter Power-down mode until CPU sets the Power-down enable bit PDEN(CLK_PWRCTL[7]) and Cortex®-M0 core executes the WFI instruction. After that, chip enters Power-down mode and wait for wake-up interrupt source triggered to leave Power-down mode. In Power-down mode, the clock controller turns off the 4~24 MHz external high speed crystal (HXT), internal 22.1184 MHz internal high speed RC oscillator (HIRC) and 48 MHz internal high speed RC oscillator (HIRC48) to reduce the overall system power consumption. Figure 6.3-1 shows the clock generator and the overview of the clock source control.

The clock generator consists of 6 clock sources, which are listed below:

- 32.768 kHz external low-speed crystal oscillator (LXT)
- 4~24 MHz external high speed crystal oscillator (HXT)
- Programmable PLL output clock frequency (PLLFOUT), PLL source can be selected from external 4~24 MHz external high speed crystal (HXT) or 22.1184 MHz internal high speed oscillator (HIRC)
- 22.1184 MHz internal high speed RC oscillator (HIRC)
- 48 MHz internal high speed RC oscillator (HIRC48)
- 10 kHz internal low speed RC oscillator (LIRC)

Each of these clock sources has certain stable time to wait for clock operating at stable frequency. When clock source is enabled, a stable counter start counting and correlated clock stable index (HIRCSTB(CLK_STATUS[4]), LIRCSTB(CLK_STATUS[3]), PLLSTB(CLK_STATUS[2]), HXTSTB(CLK_STATUS[0]), LXTSTB(CLK_STATUS[1]) and HIRC48STB(CLK_STATUS[5])) are set to 1 after stable counter value reach a define value as shown in Table 6.3-8. System and peripheral can use the clock as its operating clock only when correlate clock stable index is set to 1. The clock stable index will auto clear when user disables the clock source (LIRCEN(CLK_PWRCTL[3]), HIRCEN(CLK_PWRCTL[2]), HXTEN(CLK_PWRCTL[0]), PD(CLK_PLLCTL[16]), LXTEN(CLK_PWRCTL[1]) and HIRC48EN(CLK_PWRCTL[13])). Besides, the clock stable index of HXT, HIRC and PLL will auto clear when chip enter power-down and clock stable counter will re-counting after chip wake-up if correlate clock is enabled.

6.10 Hardware Divider (HDIV)

6.10.1 Overview

The hardware divider (HDIV) is useful to the high performance application. The hardware divider is a signed, integer divider with both quotient and remainder outputs.

6.10.2 Features

- Signed (two's complement) integer calculation
- 32-bit dividend with 16-bit divisor calculation capacity
- 32-bit quotient and 32-bit remainder outputs (16-bit remainder with sign extends to 32-bit)
- Divided by zero warning flag
- 6 HCLK clocks taken for one cycle calculation
- Write divisor to trigger calculation
- Waiting for calculation ready automatically when reading quotient and remainder

6.10.3 Blcok Diagram

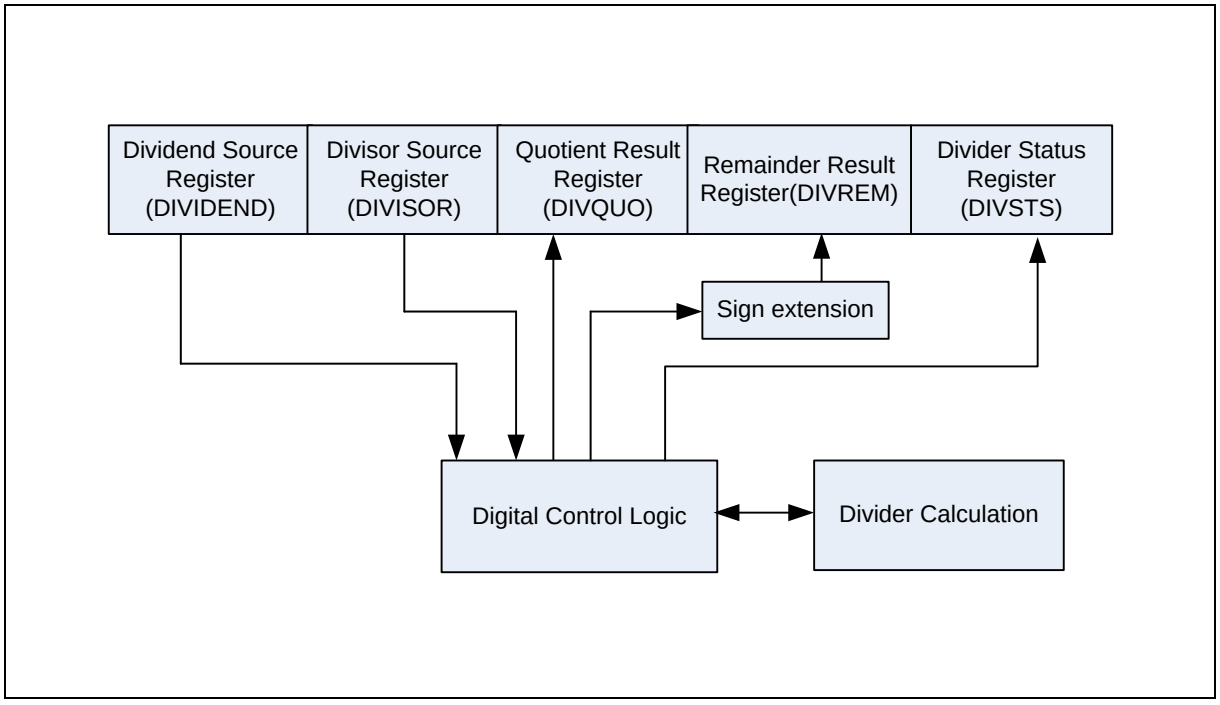


Figure 6.10-1 Hardware Divider Block Diagram

6.12 PDMA Controller (PDMA)

6.12.1 Overview

The peripheral direct memory access (PDMA) controller is used to provide high-speed data transfer. The PDMA controller can transfer data from one address to another without CPU intervention. This has the benefit of reducing the workload of CPU and keeps CPU resources free for other applications. The PDMA controller has a total of 5 channels and each channel can perform transfer between memory and peripherals or between memory and memory. The PDMA supports time-out function for channel 0 and channel 1.

6.12.2 Features

- Supports 5 independently configurable channels
- Supports selectable 2 level of priority (fixed priority or round-robin priority)
- Supports transfer data width of 8, 16, and 32 bits
- Supports source and destination address increment size can be byte, half-word, word or no increment
- Supports software and SPI, UART, I^2S, I^2C, ADC, PWM and TIMER request
- Supports Scatter-Gather mode to perform sophisticated transfer through the use of the descriptor link list table
- Supports single and burst transfer type
- Supports time-out function for channel0 and channel 1

6.16.4 Programming Examples

Example 1: The SPI controller is set as a full-duplex master to access an off-chip slave device with the following specifications:

- Data bit is latched on positive edge of SPI bus clock.
- Data bit is driven on negative edge of SPI bus clock.
- Data is transferred from MSB first.
- SPI bus clock is idle at low state.
- Only one byte of data to be transmitted/received in a transaction.
- Use the first SPI slave select pin to connect with an off-chip slave device. The slave selection signal is active low.

The operation flow is as follows:

1) Set DIVIDER (SPIx_CLKDIV [7:0]) to determine the output frequency of SPI clock.
2) Write the SPIx_SSCTL register a proper value for the related settings of Master mode:
    1. Clear AUTOSS (SPIx_SSCTL[3]) to 0 to disable the Automatic Slave Selection function.
    2. Configure slave selection signal as active low by clearing SSACTPOL (SPIx_SSCTL[2]) to 0.
    3. Enable slave selection signal by setting SS (SPIx_SSCTL[0]) to 1 to activate the off-chip slave device.
3) Write the related settings into the SPIx_CTL register to control the SPI master actions.
    1. Configure this SPI controller as master device by setting SLAVE (SPIx_CTL[18]) to 0.
    2. Force the SPI clock idle state at low by clearing CLKPOL (SPIx_CTL[3]) to 0.
    3. Select data transmitted on negative edge of SPI bus clock by setting TXNEG (SPIx_CTL[2]) to 1.
    4. Select data latched on positive edge of SPI bus clock by clearing RXNEG (SPIx_CTL[1]) to 0.
    5. Set the bit length of a transaction as 8-bit in DWIDTH bit field (SPIx_CTL[12:8] = 0x08).
    6. Set MSB transfer first by clearing LSB (SPIx_CTL[13]) to 0.
4) Set SPIEN (SPIx_CTL[0]) to 1 to enable the data transfer with the SPI interface.
5) If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPIx_TX register.
6) Waiting for SPI interrupt if the UNITIEN (SPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]).
7) Read out the received one byte data from SPIx_RX register.
8) Go to 5) to continue another data transfer or set SS (SPIx_SSCTL[0]) to 0 to inactivate the off-chip slave device.

Example 2: The SPI controller is set as a full-duplex slave device and connects with an off-chip master device. The off-chip master device communicates with the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is latched on positive edge of SPI bus clock.
- Data bit is driven on negative edge of SPI bus clock.
- Data is transferred from LSB first.
- SPI bus clock is idle at high state.
- Only one byte of data to be transmitted/received in a transaction.
- Slave selection signal is active high.

The operation flow is as follows:

1) Write the SPIx_SSCTL register a proper value for the related settings of Slave mode.

   Select high level for the input of slave selection signal by setting SSACTPOL (SPIx_SSCTL[2]) to 1.

2) Write the related settings into the SPIx_CTL register to control this SPI slave actions
   1. Set the SPI controller as slave device by setting SLAVE (SPIx_CTL[18]) to 1.
   2. Select the SPI clock idle state at high by setting CLKPOL (SPIx_CTL[3]) to 1.
   3. Select data transmitted on negative edge of SPI bus clock by setting TXNEG (SPIx_CTL[2]) to 1.
   4. Select data latched on positive edge of SPI bus clock by clearing RXNEG (SPIx_CTL[1]) to 0.
   5. Set the bit length of a transaction as 8-bit in DWIDTH bit field (SPIx_CTL[12:8] = 0x08).
   6. Set LSB transfer first by setting LSB (SPIx_CTL[13]) to 1.
3) Set the SPIEN (SPIx_CTL[0]) to 1. Wait for the slave select trigger input and SPI clock input from the off-chip master device to start the data transfer.
4) If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPIx_TX register.
5) If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the SPIx_TX register does not need to be updated by software.
6) Waiting for SPI interrupt if the UNITIEN (SPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]).
7) Read out the received one byte data from SPIx_RX register.
8) Go to 4) to continue another data transfer or stop data transfer.

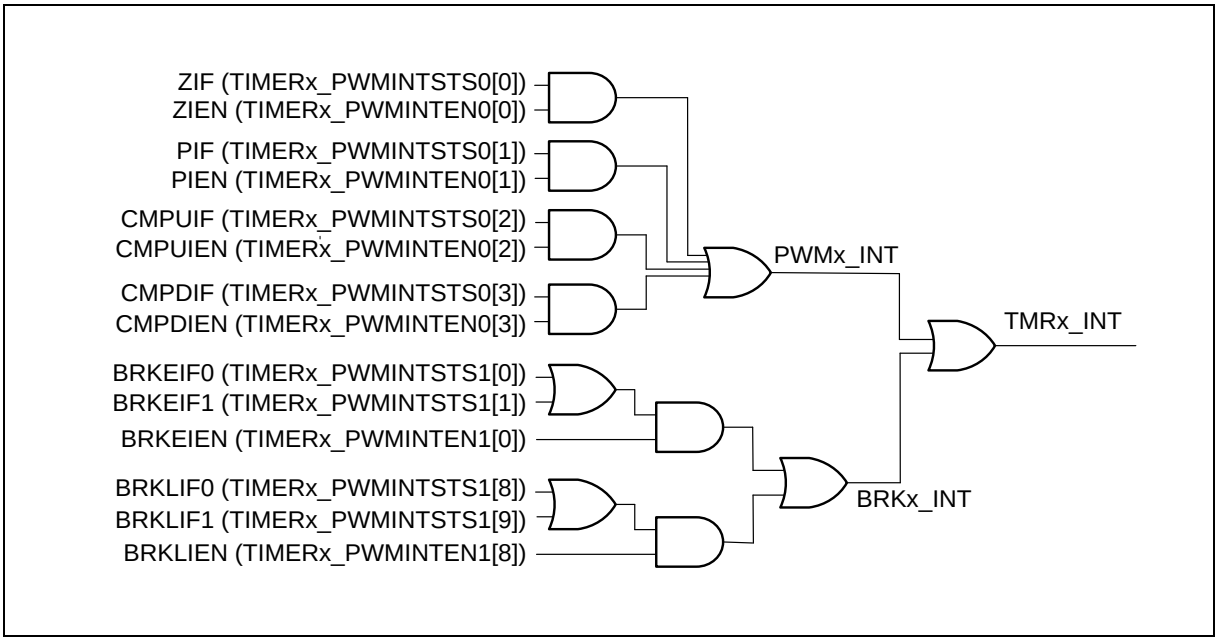


Figure 6.17-23 PWM Interrupt Architecture Diagram

6.17.3.20 PWM Trigger ADC Generator

The PWM counter event can be one of the ADC conversion trigger source. User sets TRGSEL (TIMERx_PWMADCTS[3:0]) to select which PWM counter event can trigger ADC conversion after TRGEN (TIMERx_PWMADCTS [7]) is enabled.

There are five PWM counter events can be selected as the trigger source to start ADC conversion which shown in Figure 6.17-24.

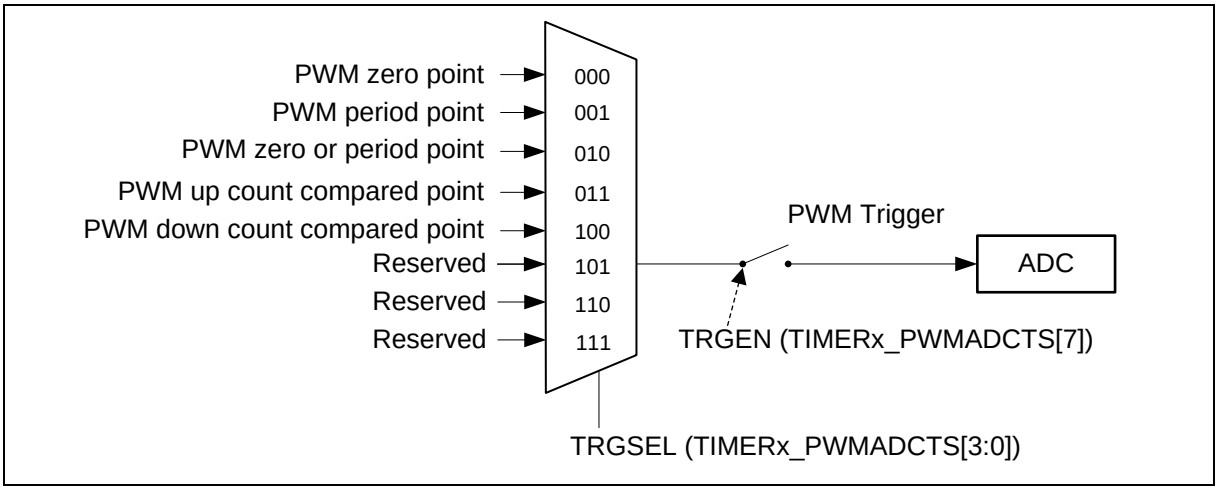


Figure 6.17-24 PWM Trigger ADC Block Diagram

6.19 USCI – UART Mode

6.19.1 Overview

The asynchronous serial channel UART covers the reception and the transmission of asynchronous data frames. It performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the controller. The receiver and transmitter being independent, frames can start at different points in time for transmission and reception.

The UART controller also provides auto flow control. There are two conditions to wake up the system.

6.19.2 Features

- Supports one transmit buffer and two receive buffer for data payload
- Supports hardware auto flow control function
- Supports programmable baud-rate generator
- Support 9-Bit Data Transfer (Support 9-Bit RS-485)
- Baud rate detection possible by built-in capture event of baud rate generator
- Supports Wake-up function (Data and nCTS Wakeup Only)

- Supports Word Suspend function
- Supports 3-wire, no slave select signal, bi-direction interface
- Supports wake-up function by slave select signal in Slave mode
- Supports one data channel half-duplex transfer

6.23 Watchdog Timer (WDT)

6.23.1 Overview

The Watchdog Timer (WDT) is used to perform a system reset when system runs into an unknown state. This prevents system from hanging for an infinite period of time. Besides, the Watchdog Timer supports the function to wake up system from Idle/Power-down mode.

6.23.2 Features

- Supports 18-bit free running up counter
- Selectable time-out interval (2^4 ~ 2^{18}) and the time-out interval is 1.6 ms ~ 26.214s if WDT_CLK is 10 kHzSupports selectable WDT reset delay period between WDT time-out event to WDT reset system event, and it includes 1026、130、18 or 3 * WDT_CLK delay period
- System kept in reset state about 63 * WDT_CLK period time after system reset event occurred
- Supports to force WDT function enabled after chip powered on or reset by setting CWDTEN[2:0] in Config0 register
- Supports WDT time-out wake-up function only if WDT clock source is selected as LIRC or LXT

6.23.3 Clock Control

The WDT clock control is shown in Figure 6.23-1.

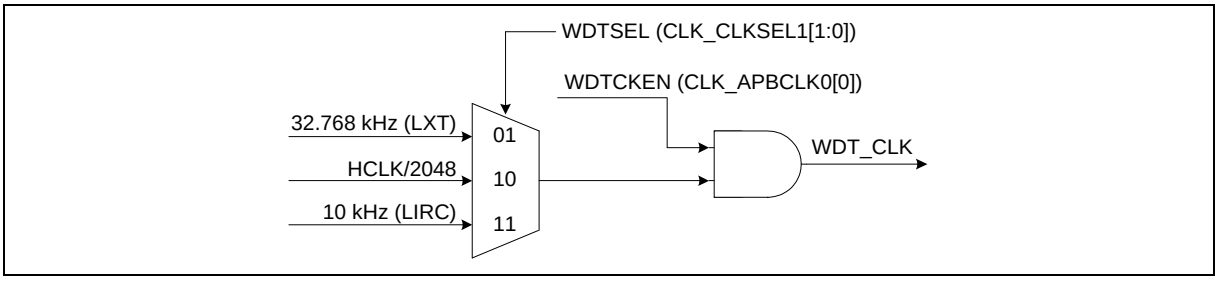


Figure 6.23-1 Watchdog Timer Clock Control

6.24 Window Watchdog Timer (WWDT)

6.24.1 Overview

The Window Watchdog Timer (WWDT) is used to perform a system reset while WWDT counter is not reload within a specified window period when application program run to uncontrollable status by any unpredictable condition.

6.24.2 Features

- Supports 6-bit down counter value CNTDAT (WWDT_CNT[5:0]) and maximum 6-bit compare value CMPDAT (WWDT_CTL[21:16]) to adjust the WWDT compare time-out window period flexible
- Supports PSCSEL (WWDT_CTL[11:8]) to programmable maximum 11-bit prescale counter period of WWDT counter
- WWDT counter suspends in Idle/Power-down mode
- WWDT counter only can be reloaded within in valid window period to prevent system reset

6.24.3 Clock Control

The WWDT clock control and block diagram are shown as follows.

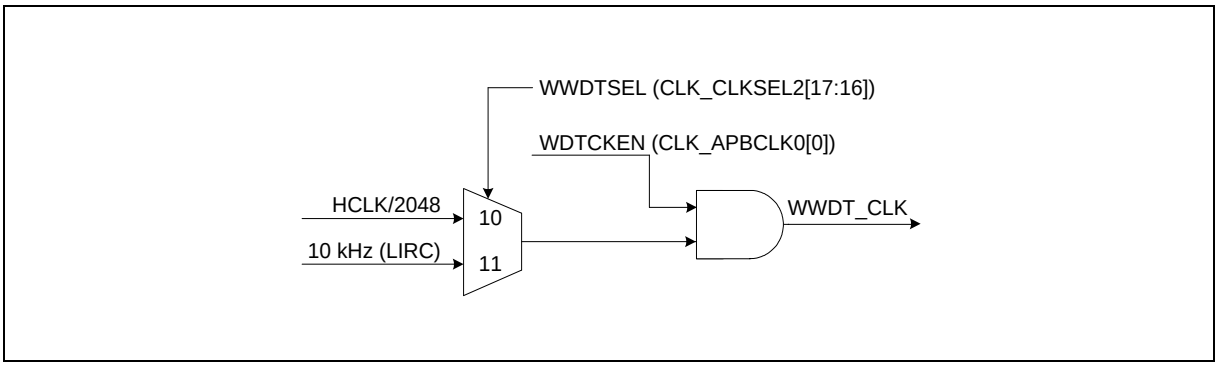


Figure 6.24-1 WWDT Clock Control

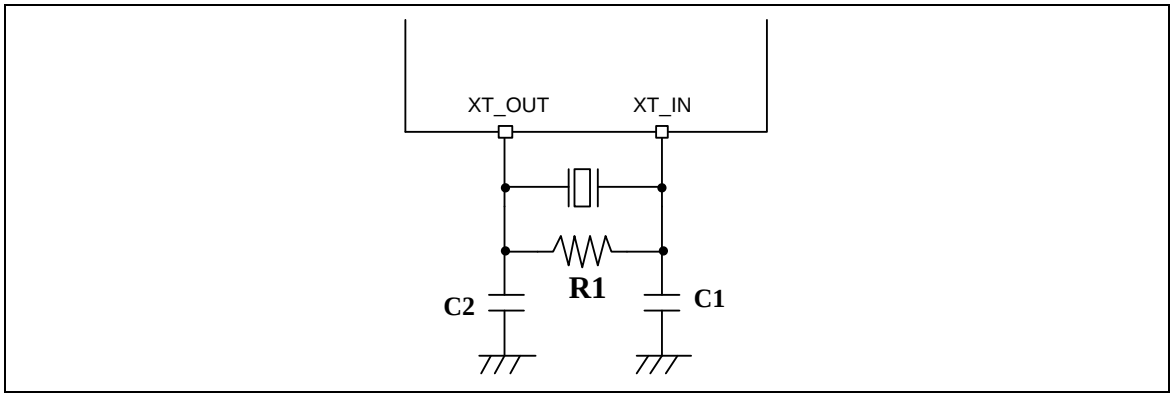
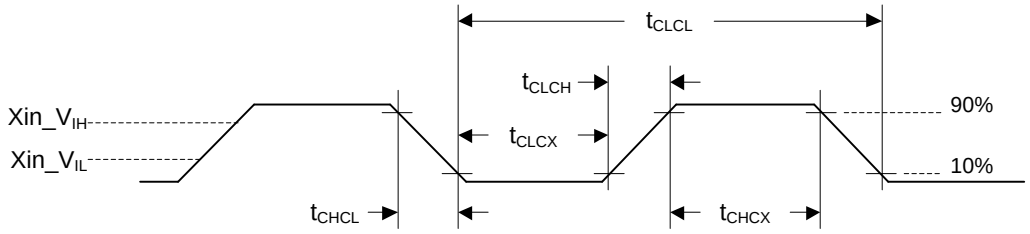


Figure 8.3-1 Typical Crystal Application Circuit

8.3.3 External 32.768 kHz Low Speed Crystal (LXT) Input Clock

PARAMETER	SYM.	SPECIFICATIONS				TEST CONDITION
		MIN.	TYP.	MAX.	UNIT	
Clock High Time	t_{CHCX}	TBD	-	-	nS	
Clock Low Time	t_{CLCX}	TBD	-	-	nS	
Clock Rise Time	t_{CLCH}	TBD	-	TBD	nS	
Clock Fall Time	t_{CHCL}	TBD	-	TBD	nS	
LXT Input Pin Input High Voltage	Xin_V_{IH}	0.7V_{DD}	-	V_{DD}	V	
LXT Input Pin Input Low Voltage	Xin_V_{IL}	0	-	0.3V_{DD}	V	



Note: Duty cycle is 50%.

8.7 SPI Dynamic Characteristics

8.7.1 Dynamic Characteristics of Data Input and Output Pin

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
SPI MASTER MODE (VDD = 4.5 V~5.5V, 30 PF LOADING CAPACITOR)					
t_{DS}	Data setup time	4	2	-	ns
t_{DH}	Data hold time	0	-	-	ns
t_V	Data output valid time	-	7	11	ns
SPI MASTER MODE (VDD = 3.0~3.6 V, 30 PF LOADING CAPACITOR)					
t_{DS}	Data setup time	5	3	-	ns
t_{DH}	Data hold time	0	-	-	ns
t_V	Data output valid time	-	13	18	ns

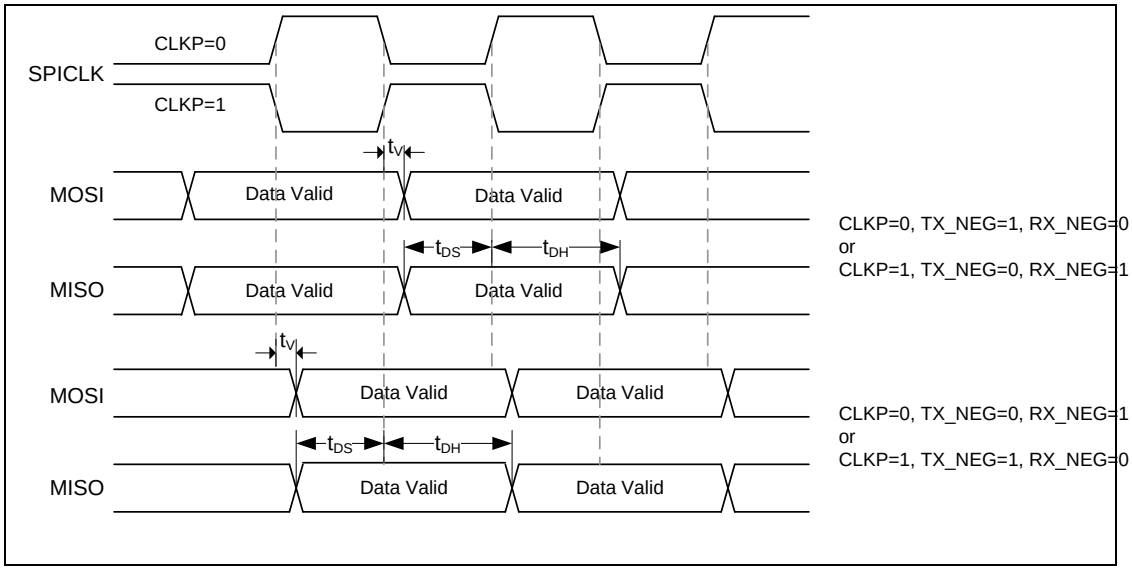


Figure 8.7-1 SPI Master Mode Timing Diagram

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
SPI SLAVE MODE (VDD = 4.5 V~5.5V, 30 PF LOADING CAPACITOR)					
t_{DS}	Data setup time	0	-	-	ns
t_{DH}	Data hold time	2*PCLK+4	-	-	ns
t_V	Data output valid time	-	2*PCLK+11	2*PCLK+19	ns
SPI SLAVE MODE (VDD = 3.0 V ~ 3.6 V, 30 PF LOADING CAPACITOR)					
t_{DS}	Data setup time	0	-	-	ns