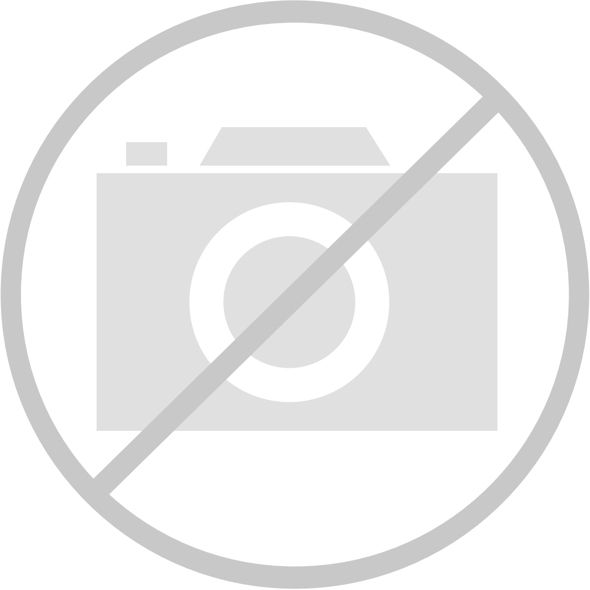
NXP Semiconductors - MC9S08PT60VQH Datasheet



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	HCS08
Core Size	8-Bit
Speed	20MHz
Connectivity	I^{2}C, LINbus, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	57
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b SAR
Oscillator Type	External, Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08pt60vqh

Table 5-1. Vector summary (from lowest to highest priority) (continued)

Vector number	Address (high/ low)	Vector name	Module	Source	Local enable	Description
				TC	TCIE	
24	0xFFCE:FFCF	Vsci0rx	SCI0	IDLE RDRF LBKDIF RXEDGIF	ILIE RIE LBKDIE RXEDGIE	SCI0 receive
23	0xFFD0:FFD1	Vsci0err	SCI0	OR NF FE PF	ORIE NEIE FEIE PEIE	SCI0 error
22	0xFFD2:FFD3	Vadc	ADC	COCO	AIEN	ADC conversion complete interrupt
21	0xFFD4:FFD5	Vacmp	ACMP	ACF	ACIE	Analog comparator interrupt
20	0xFFD6:FFD7	Vmtim1	MTIM1	TOF	TOIE	MTIM1 overflow interrupt
19	0xFFD8:FFD9	Vmtim0	MTIM0	TOF	TOIE	MTIM0 overflow interrupt
18	0xFFDA:FFDB	Vftm0ovf	FTM0	TOF	TOIE	FTM0 overflow
17	0xFFDC:FFDD	Vftm0ch1	FTM0CH1	CH1F	CH1IE	FTM0 channel 1
16	0xFFDE:FFDF	Vftm0ch0	FTM0CH0	CH0F	CH0IE	FTM0 channel 0
15	0xFFE0:FFE1	Vftm1ovf	FTM1	TOF	TOIE	FTM1 overflow
14	0xFFE2:FFE3	Vftm1ch1	FTM1CH1	CH1F	CH1IE	FTM1 channel 1
13	0xFFE4:FFE5	Vftm1ch0	FTM1CH0	CH0F	CH0IE	FTM1 channel 0
12	0xFFE6:FFE7	Vftm2ovf	FTM2	TOF	TOIE	FTM2 overflow
11	0xFFE8:FFE9	Vftm2ch5	FTM2CH5	CH5F	CH5IE	FTM2 channel 5
10	0xFFEA:FFEB	Vftm2ch4	FTM2CH4	CH4F	CH4IE	FTM2 channel 4
9	0xFFEC:FFED	Vftm2ch3	FTM2CH3	CH3F	CH3IE	FTM2 channel 3
8	0xFFEE:FFEF	Vftm2ch2	FTM2CH2	CH2F	CH2IE	FTM2 channel 2
7	0xFFF0:FFF1	Vftm2ch1	FTM2CH1	CH1F	CH1IE	FTM2 channel 1
6	0xFFF2:FFF3	Vftm2ch0	FTM2CH0	CH0F	CH0IE	FTM2 channel 0
5	0xFFF4:FFF5	Vftm2fault	FTM2	FAULTF	FAULTIE	FTM2 fault
4	0xFFF6:FFF7	Vclk	ICS	LOLS	LOLIE	Clock loss of lock
3	0xFFF8:FFF9	Vlvw	System control	LVWF	LVWIE	Low-voltage warning
2	0xFFFA:FFFB	Vwdog Virq	WDOG IRQ	WDOGF IRQF	WDOGI IRQIE	WDOG timeout IRQ interrupt
1	0xFFFC:FFFD	Vswi	Core	SWI Instruction	—	Software interrupt
0	0xFFFE:FFFF	Vreset	System control	WDOG LVD RESET pin	WDOGE LVDRE RSTPE	Watchdog timer Low-voltage detect External pin

5.2.1 Features

Features of the IRQ module include:

- A Dedicated External Interrupt Pin
- IRQ Interrupt Control Bits
- Programmable Edge-only or Edge and Level Interrupt Sensitivity
- Automatic Interrupt Acknowledge
- Internal pullup device

A low level applied to the external interrupt request (IRQ) pin can latch a CPU interrupt request. The following figure shows the structure of the IRQ module:

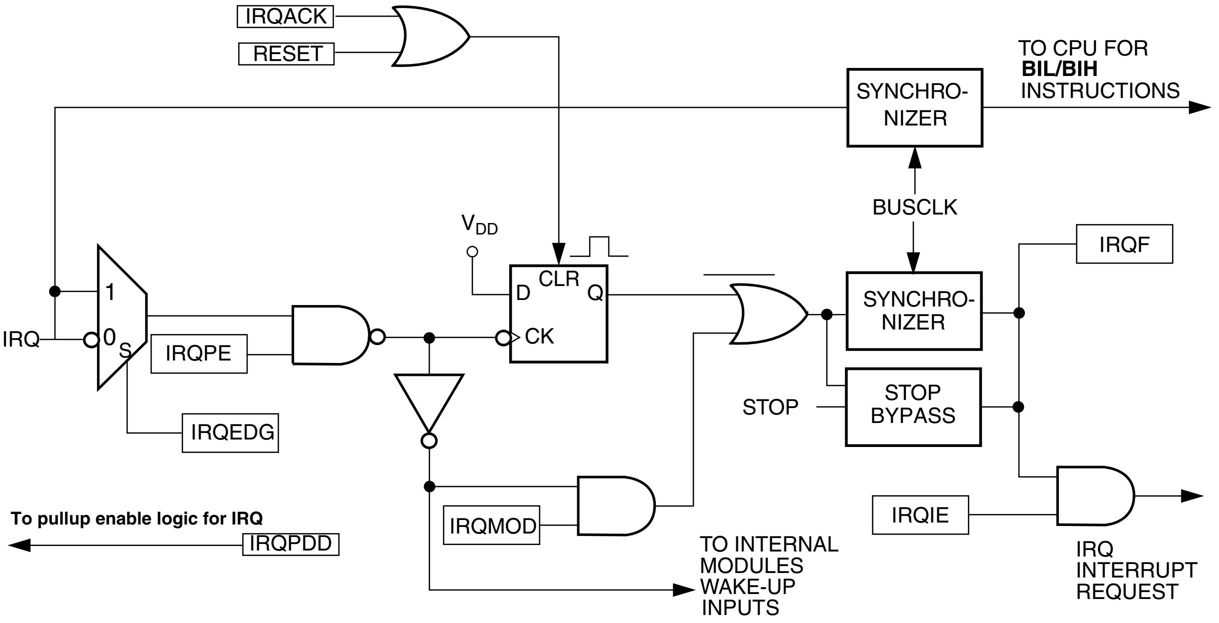


Figure 5-3. IRQ module block diagram

External interrupts are managed by the IRQSC status and control register. When the IRQ function is enabled, synchronous logic monitors the pin for edge-only or edge-and-level events. When the MCU is in stop mode and system clocks are shut down, a separate asynchronous path is used so that the IRQ, if enabled, can wake the MCU.

SYS_SOPT2 field descriptions (continued)

Field	Description
	0 TxD0 output is connected to pinout directly. 1 TxD0 output is modulated by FTM0 channel 0 before mapped to pinout.
6 FTMSYNC	FTM2 Synchronization Select Writing a 1 to this bit generates a PWM synchronization trigger to the FTM modules. 0 No synchronization triggered. 1 Generate a PWM synchronization trigger to the FTM2 modules.
5 RXDFE	SCI0 RxD Filter Select This bit enables the SCI0 RxD input filtered by ACMP. When this function is enabled, any signal tagged with ACMP inputs can be regarded SCI0. 0 RXD0 input signal is connected to SCI0 module directly. 1 RXD0 input signal is filtered by ACMP, then injected to SCI0.
4 RXDCE	SCI0 RxD Capture Select This bit enables the SCI0 RxD is captured by FTM0 channel 1. 0 RXD0 input signal is connected to SCI0 module only. 1 RXD0 input signal is connected to SCI0 module and FTM0 channel 1.
3 ACIC	Analog Comparator to Input Capture Enable This bit connects the output of ACMP to FTM1 input channel 0. 0 ACMP output not connected to FTM1 input channel 0. 1 ACMP output connected to FTM1 input channel 0.
2 RTCC	Real-Time Counter Capture This bit allows the Real-time Counter (RTC) overflow to be captured by FTM1 channel 1. 0 RTC overflow is not connected to FTM1 input channel 1. 1 RTC overflow is connected to FTM1 input channel 1.
ADHWTS	ADC Hardware Trigger Source These bits select the ADC hardware trigger source. All trigger sources start ADC conversion on rising edge. 00 RTC overflow as the ADC hardware trigger. 01 MTIM0 overflow as the ADC hardware trigger. 10 FTM2 init trigger with 8-bit programmable delay. 11 FTM2 match trigger with 8-bit programmable delay.

7.5 High current drive

Output high sink/source current drive can be enabled by setting the corresponding bit in the HDRVE register for PTH1, PTH0, PTE1, PTE0, PTD1, PTD0, PTB5 and PTB4. Output high sink/source current when they are operated as output. High current drive function is disabled if the pin is configured as an input by the parallel I/O control logic. When configured as any shared peripheral function, high current drive function still works on these pins, but only when they are configured as outputs.

7.6 Pin behavior in stop mode

In stop3 mode, all I/O is maintained because internal logic circuitry stays powered up. Upon recovery, normal I/O function is available to the user.

7.7 Port data registers

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Port A Data Register (PORT_PTAD)	8	R/W	00h	7.7.1/162
1	Port B Data Register (PORT_PTBD)	8	R/W	00h	7.7.2/163
2	Port C Data Register (PORT_PTCD)	8	R/W	00h	7.7.3/163
3	Port D Data Register (PORT_PTDD)	8	R/W	00h	7.7.4/164
4	Port E Data Register (PORT_PTED)	8	R/W	00h	7.7.5/164
5	Port F Data Register (PORT_PTFD)	8	R/W	00h	7.7.6/165
6	Port G Data Register (PORT_PTGD)	8	R/W	00h	7.7.7/165
7	Port H Data Register (PORT_PTHD)	8	R/W	00h	7.7.8/166
30AF	Port High Drive Enable Register (PORT_HDRVE)	8	R/W	00h	7.7.9/167
30B0	Port A Output Enable Register (PORT_PTAOE)	8	R/W	00h	7.7.10/168
30B1	Port B Output Enable Register (PORT_PTBOE)	8	R/W	00h	7.7.11/169
30B2	Port C Output Enable Register (PORT_PTCOE)	8	R/W	00h	7.7.12/170
30B3	Port D Output Enable Register (PORT_PTDOE)	8	R/W	00h	7.7.13/172
30B4	Port E Output Enable Register (PORT_PTEOE)	8	R/W	00h	7.7.14/173
30B5	Port F Output Enable Register (PORT_PTFOE)	8	R/W	00h	7.7.15/174
30B6	Port G Output Enable Register (PORT_PTGOE)	8	R/W	00h	7.7.16/175
30B7	Port H Output Enable Register (PORT_PTHOE)	8	R/W	00h	7.7.17/176

Table continues on the next page...

PORT_PTEIE field descriptions (continued)

Field	Description
	0 Input disabled for port E bit 3. 1 Input enabled for port E bit 3.
2 PTEIE2	Input Enable for Port E Bit 2 This read/write bit enables the port E pin as an input. 0 Input disabled for port E bit 2. 1 Input enabled for port E bit 2.
1 PTEIE1	Input Enable for Port E Bit 1 This read/write bit enables the port E pin as an input. 0 Input disabled for port E bit 1. 1 Input enabled for port E bit 1.
0 PTEIE0	Input Enable for Port E Bit 0 This read/write bit enables the port E pin as an input. 0 Input disabled for port E bit 0. 1 Input enabled for port E bit 0.

7.7.23 Port F Input Enable Register (PORT_PTFIE)

Address: 0h base + 30BDh offset = 30BDh

Bit	7	6	5	4	3	2	1	0
Read Write	PTFIE7	PTFIE6	PTFIE5	PTFIE4	PTFIE3	PTFIE2	PTFIE1	PTFIE0
Reset	0	0	0	0	0	0	0	0

PORT_PTFIE field descriptions

Field	Description
7 PTFIE7	Input Enable for Port F Bit 7 This read/write bit enables the port F pin as an input. 0 Input disabled for port F bit 7. 1 Input enabled for port F bit 7.
6 PTFIE6	Input Enable for Port F Bit 6 This read/write bit enables the port F pin as an input. 0 Input disabled for port F bit 6. 1 Input enabled for port F bit 6.
5 PTFIE5	Input Enable for Port F Bit 5 This read/write bit enables the port F pin as an input.

Table continues on the next page...

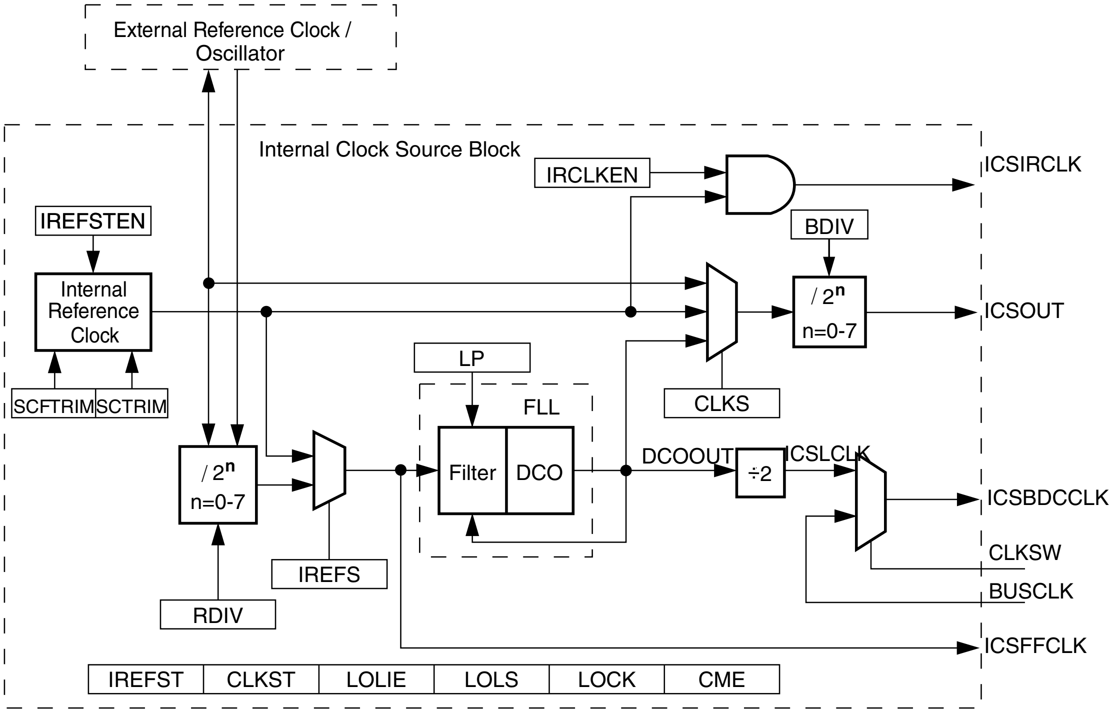


Figure 8-2. Internal clock source (ICS)

8.2.1.1 Bus frequency divider

The ICS_C2[BDIV] bits can be changed at anytime and the actual switch to the new frequency occurs immediately.

8.2.1.2 Low power bit usage

The low power bit (ICS_C2[LP]) is provided to allow the FLL to be disabled and thus conserve power when it is not used.

However, in some applications it may be desirable to allow the FLL to be enabled and to lock for maximum accuracy before switching to an FLL engaged mode. To do this, write the ICS_C2[LP] bit to 0.

8.2.1.3 Internal reference clock (ICSIRCLK)

When ICS_C1[IRCLKEN] is set the internal reference clock signal is presented as ICSIRCLK, which can be used as an additional clock source. To re-target the ICSIRCLK frequency, write a new value to the ICS_C3[SCTRIM] and ICS_C4[SCFTRIM] bits to trim the period of the internal reference clock:

- In output compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of triggers (match trigger)
- Software control of PWM outputs
- Up to four fault inputs for global fault control
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- The generation of an interrupt when the fault condition is detected
- Synchronized loading of write buffered FTM registers
- Write protection for critical registers
- Backwards compatible with TPM
- Testing of input captures for a stuck at zero and one conditions
- Dual edge capture for pulse and period width measurement

12.1.3 Modes of operation

When the MCU is in active BDM background or BDM foreground mode, the FTM temporarily suspends all counting until the MCU returns to normal user operating mode. During stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During wait mode, the FTM continues to operate normally. If the FTM does not need to produce a real time reference or provide the interrupt sources needed to wake the MCU from wait mode, the power can then be saved by disabling FTM functions before entering wait mode.

12.3.20 External Trigger (FTMx_EXTTRIG)

This register indicates when a channel trigger was generated, enables the generation of a trigger when the FTM counter is equal to its initial value, and selects which channels are used in the generation of the channel triggers. Several FTM channels can be selected to generate multiple triggers in one PWM period.

Channels 6 and 7 are not used to generate channel triggers.

Address: Base address + 23h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TRIGF	INITTRIGEN	CH1TRIG	CH0TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG
Reset	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
7 TRIGF	Channel Trigger Flag Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to its initial value. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.
5 CH1TRIG	Channel 1 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

Table continues on the next page...

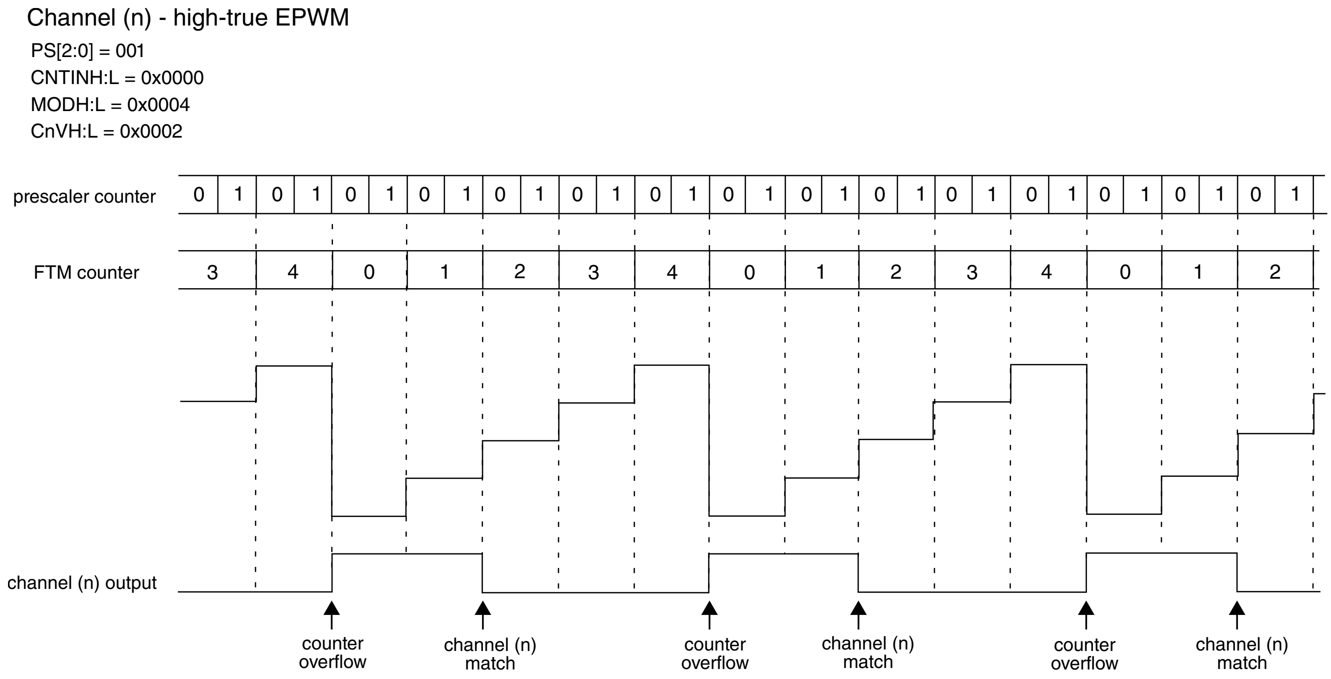


Figure 12-186. Notation used

12.4.1 Clock Source

FTM module has only one clock domain that is the system clock.

12.4.1.1 Counter Clock Source

The CLKS[1:0] bits in the SC register select one of three possible clock sources for the FTM counter or disable the FTM counter. After any MCU reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The fixed frequency clock is an alternative clock source for the FTM counter that allows the selection of a clock other than the system clock or an external clock. This clock input is defined by chip integration. Refer to chip specific documentation for further information. Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed the system clock frequency.

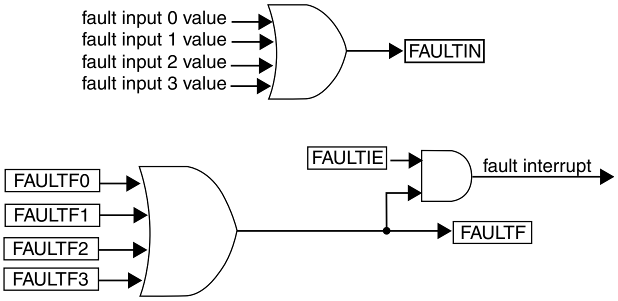


Figure 12-243. FAULTF and FAULTIN bits and fault interrupt

If the fault control is enabled (FAULTM[1:0] ≠ 0:0), a fault condition has occurred (rising edge at the logic OR of the enabled fault input) and (FAULTEN = 1), then channel (n) and (n+1) outputs are forced to their safe value (that is, the channel (n) output is forced to the value of POL(n) and the channel (n+1) is forced to the value of POL(n +1)).

The fault interrupt is generated when (FAULTF = 1) and (FAULTIE = 1). This interrupt request remains set until:

- Software clears the FAULTF bit (by reading FAULTF bit as 1 and writing 0 to it)
- Software clears the FAULTIE bit
- A reset occurs

Note

Fault control is available only in combine mode.

12.4.14.1 Automatic fault clearing

If the automatic fault clearing is selected (FAULTM[1:0] = 1:1), then the disabled channel outputs are enabled when the fault input signal (FAULTIN) returns to zero and a new PWM cycle begins. See the following figure.

14.2.1.1 Wait mode

The RTC continues to run in Wait mode if enabled before executing the WAIT instruction. Therefore, the RTC can be used to bring the MCU out of Wait mode if the real-time interrupt is enabled. For lowest possible current consumption, the RTC must be stopped by software if not needed as an interrupt source during Wait mode.

14.2.1.2 Stop modes

The RTC continues to run in Stop3 mode if the RTC is enabled before executing the STOP instruction. Therefore, the RTC can be used to bring the MCU out of stop modes with no external components, if the real-time interrupt is enabled.

14.2.2 Block diagram

The block diagram for the RTC module is shown in the following figure.

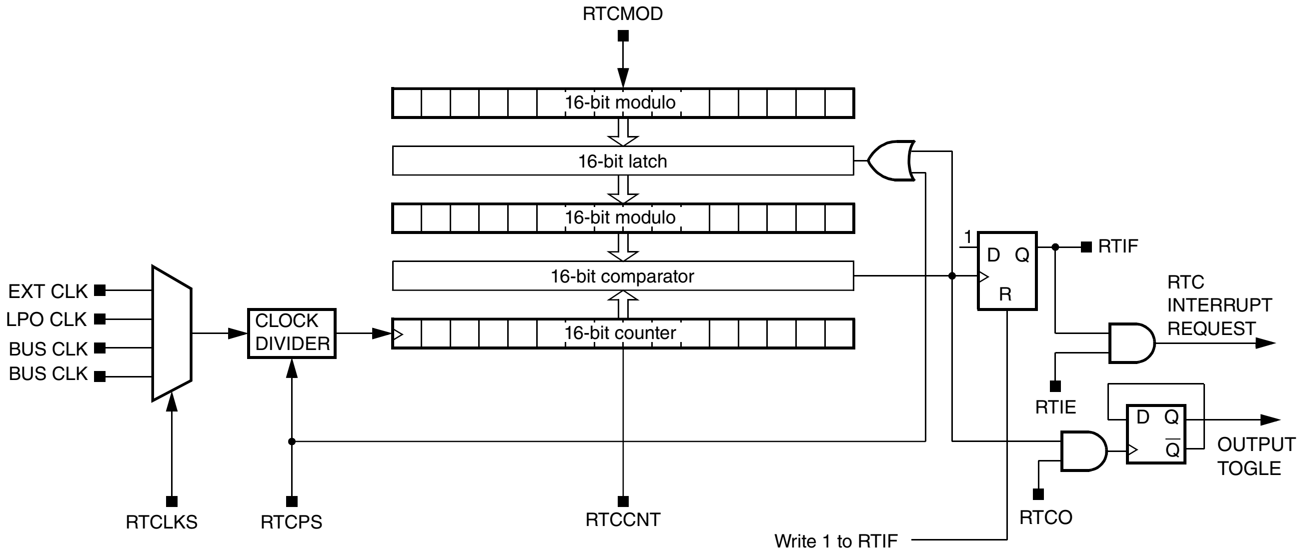


Figure 14-1. Real-time counter (RTC) block diagram

14.3 External signal description

RTCO is the output of RTC. After MCU reset, the RTC_SC1[RTCO] is set to high. When the counter overflows, the output is toggled.

15.4.2.1 Send break and queued idle

SCI_C2[SBK] sends break characters originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 10 bit times including the start and stop bits. A longer break of 13 bit times can be enabled by setting SCI_S2[BRK13]. Normally, a program would wait for SCI_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 1, and then write 0 to SCI_C2[SBK]. This action queues a break character to be sent as soon as the shifter is available. If SCI_C2[SBK] remains 1 when the queued break moves into the shifter, synchronized to the baud rate clock, an additional break character is queued. If the receiving device is another Freescale Semiconductor SCI, the break characters are received as 0s in all eight data bits and a framing error (SCI_S1[FE] = 1) occurs.

When idle-line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for SCI_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the SCI_C2[TE] bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while SCI_C2[TE] is cleared, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while SCI_C2[TE] is cleared, set the general-purpose I/O controls so the pin shared with TxD is an output driving a logic 1. This ensures that the TxD line looks like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to SCI_C2[TE].

The length of the break character is affected by the SCI_S2[BRK13] and SCI_C1[M] as shown below.

Table 15-39. Break character length

BRK13	M	SBNS	Break character length
0	0	0	10 bit times
0	0	1	11 bit times
0	1	0	11 bit times
0	1	1	12 bit times
1	0	0	13 bit times
1	0	1	14 bit times
1	1	0	14 bit times
1	1	1	15 bit times

When CPHA = 0, the slave begins to drive its MISO output with the first data bit value (MSB or LSB depending on LSBFE) when SS goes to active low. The first SPSCK edge causes both the master and the slave to sample the data bit values on their MISO and MOSI inputs, respectively. At the second SPSCK edge, the SPI shifter shifts one bit position which shifts in the bit value that was just sampled and shifts the second data bit value out the other end of the shifter to the MOSI and MISO outputs of the master and slave, respectively. When CPHA = 0, the slave's SS input must go to its inactive high level between transfers.

16.4.5 SPI Baud Rate Generation

As shown in the following figure, the clock source for the SPI baud rate generator is the bus clock. The three prescale bits (SPPR2:SPPR1:SPPR0) choose a prescale divisor of 1, 2, 3, 4, 5, 6, 7, or 8. The three rate select bits (SPR3:SPR2:SPR1:SPR0) divide the output of the prescaler stage by 2, 4, 8, 16, 32, 64, 128, 256, or 512 to get the internal SPI master mode bit-rate clock.

The baud rate generator is activated only when the SPI is in the master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease I_{DD} current.

The baud rate divisor equation is as follows (except those reserved combinations in the SPI Baud Rate Divisor table).

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \times 2^{(\text{SPR} + 1)}$$

The baud rate can be calculated with the following equation:

$$\text{BaudRate} = \text{BusClock} / \text{BaudRateDivisor}$$

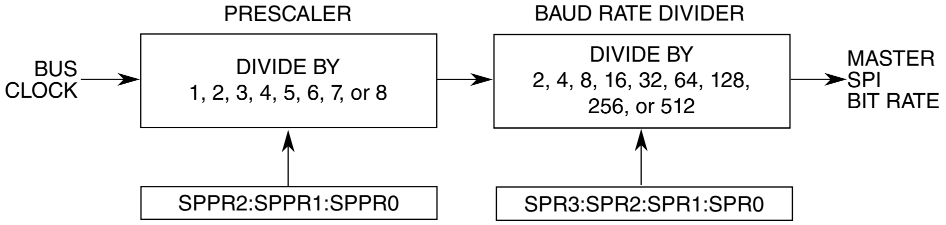


Figure 16-17. SPI Baud Rate Generation

16.4.6 Special Features

The following section shows the module special features.

17.3 Memory map/register definition

The SPI has 8-bit registers to select SPI options, to control baud rate, to report SPI status, to hold an SPI data match value, and for transmit/receive data.

SPI memory map

Address offset (hex)	Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	30A0	SPI Control Register 1 (SPI1_C1)	8	R/W	04h	17.3.1/475
1	30A1	SPI Control Register 2 (SPI1_C2)	8	R/W	00h	17.3.2/477
2	30A2	SPI Baud Rate Register (SPI1_BR)	8	R/W	00h	17.3.3/478
3	30A3	SPI Status Register (SPI1_S)	8	R	20h	17.3.4/479
4	30A4	SPI data register high (SPI1_DH)	8	R/W	00h	17.3.5/482
5	30A5	SPI Data Register low (SPI1_DL)	8	R/W	00h	17.3.6/483
6	30A6	SPI match register high (SPI1_MH)	8	R/W	00h	17.3.7/484
7	30A7	SPI Match Register low (SPI1_ML)	8	R/W	00h	17.3.8/484
8	30A8	SPI control register 3 (SPI1_C3)	8	R/W	00h	17.3.9/485
9	30A9	SPI clear interrupt register (SPI1_CI)	8	R/W	00h	17.3.10/486

17.3.1 SPI Control Register 1 (SPIx_C1)

This read/write register includes the SPI enable control, interrupt enables, and configuration options.

Address: 30A0h base + 0h offset = 30A0h

Bit	7	6	5	4	3	2	1	0
Read Write	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

SPI1_C1 field descriptions

Field	Description
7 SPIE	SPI Interrupt Enable: for SPRF and MODF (when FIFO is not supported or not enabled) or for read FIFO (when FIFO is supported and enabled) When the FIFO is not supported or not enabled (FIFOMODE is not present or is 0): Enables the interrupt for SPI receive buffer full (SPRF) and mode fault (MODF) events.

Table continues on the next page...

Chapter 18
Inter-Integrated Circuit (I2C)

18.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

The inter-integrated circuit (I^2C, I2C, or IIC) module provides a method of communication between a number of devices.

The interface is designed to operate up to 100 kbit/s with maximum bus loading and timing. The I2C device is capable of operating at higher baud rates, up to a maximum of clock/20, with reduced bus loading. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF. The I2C module also complies with the *System Management Bus (SMBus) Specification, version 2*.

18.1.1 Features

The I2C module has the following features:

- Compatible with *The I^2C-Bus Specification*
- Multimaster operation
- Software programmable for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- START and STOP signal generation and detection
- Repeated START signal generation and detection
- Acknowledge bit generation and detection

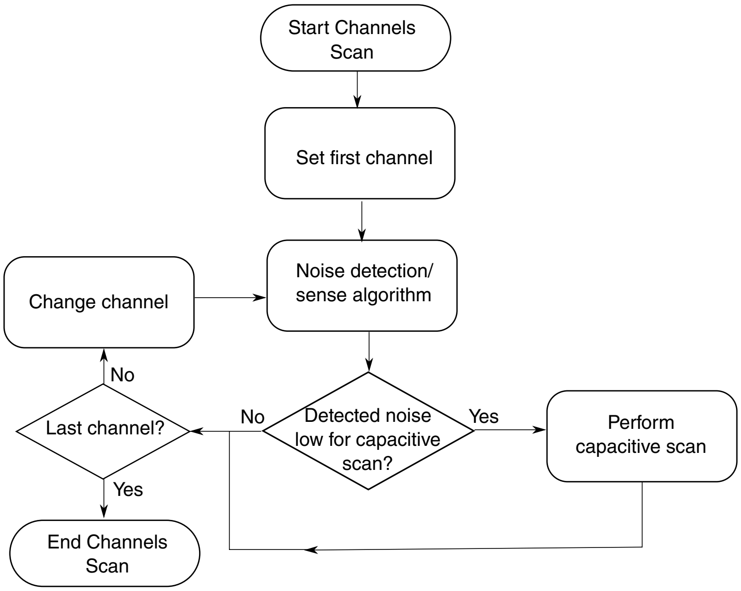


Figure 21-16. Detection of touch with noise sense and normal capacitive sense flow chart

One example of noise detection mode is shown in the following figure. in this figure the TSI is working in capacitive mode until 28 µs (T1) when it is changed to noise detection mode. In noise detection mode the selected pad is biased with 0.815V and all AC waveform in this pad is caused by a noise source external to the MCU.

It is possible to observe in the following figure that, in noise detection mode, the clkref output has the peak detection and the number of detected peaks can be counted or used by digital block. The clkext output has the internal oscillator output and can be used to set the maximum noise detection time window.

The waveform of the following figure shows two operations during noise detection mode:

- The V(vp) and V(vm) thresholds are changed in 34.4 µs (T2).
- The Rs series resistance value is changed between 184 kΩ (TSI_CS2[EXTCHRG]=011b) and 32 kΩ (TSI_CS2[EXTCHRG]=101). Because of this Rs change the amplitude of noise waveform change also.

22.5.4 CRC Data 3 Register (CRC_D3)

D3 is one of the CRC data registers (D0:D3). The set of CRC data registers contains the value of seed, data, and checksum. When CRC_CTRL[WAS] bit is set, any write to the data registers is regarded as seed for CRC module. When CRC_CTRL[WAS] bit is clear, any write to the data registers is regarded as data for general CRC computation, in which D0:D2 does not accept any data and D3 accept 8-bit write upon the polynomial configuration. When final data are written, the final result can be read from the data register. The registers of D0:D1 contain the MSB 16-bit of CRC data, which is used only in CRC 32-bit mode. Only D3 is used to dummy data to CRC. Writing D2 will be ignored when WAS = 0.

Address: 3060h base + 3h offset = 3063h

Bit	7	6	5	4	3	2	1	0
Read Write	D3							
Reset	1	1	1	1	1	1	1	1

CRC_D3 field descriptions

Field	Description
D3	CRC Data Bit 7:0

22.5.5 CRC Polynomial 0 Register (CRC_P0)

P0 is one of the CRC polynomial registers (P0:P3). The set of CRC polynominal registers contains the value of polynomial. The registers of P0:P1 contain the MSB 16-bit of CRC polynomial, which is used only in CRC 32-bit mode. The registers of P2:P3 contain the LSB 16-bit of CRC polynomial, which is used in both CRC 16- and 32-bit modes.

Address: 3060h base + 4h offset = 3064h

Bit	7	6	5	4	3	2	1	0
Read Write	P0							
Reset	0	0	0	0	0	0	0	0

CRC_P0 field descriptions

Field	Description
P0	CRC Polynominal Bit 31:24

22.5.8 CRC Polynomial 3 Register (CRC_P3)

P3 is one of the CRC polynomial registers (P0:P3). The set of CRC polynominal registers contains the value of polynomial. The registers of P0:P1 contain the MSB 16-bit of CRC polynomial, which is used only in CRC 32-bit mode. The registers of P2:P3 contain the LSB 16-bit of CRC polynomial, which is used in both CRC 16- and 32-bit modes.

Address: 3060h base + 7h offset = 3067h

Bit	7	6	5	4	3	2	1	0
Read Write	P3							
Reset	0	0	1	0	0	0	0	1

CRC_P3 field descriptions

Field	Description
P3	CRC Polynominal Bit 7:0

22.5.9 CRC Control Register (CRC_CTRL)

Address: 3060h base + 8h offset = 3068h

Bit	7	6	5	4	3	2	1	0
Read	TOT		TOTR		0	FXOR	WAS	TCRC
Write					▨			
Reset	0	0	0	0	0	0	0	0

CRC_CTRL field descriptions

Field	Description
7–6 TOT	Reverse of Write These bits identify the reverse of the input data. 00 No reverse. 01 Bit is reversed in byte; No byte is reversed. 10 Reserved. 11 Reserved.
5–4 TOTR	Reverse of Read These bits identify the reverse of the output data. 00 No reverse. 01 Bit is reversed in byte; No byte is reversed. 10 Reserved. 11 Reserved.

Table continues on the next page...

23.2.2 Watchdog Control and Status Register 2 (WDOG_CS2)

This section describes the function of the watchdog control and status register 2.

Address: 3030h base + 1h offset = 3031h

Bit	7	6	5	4	3	2	1	0
Read	WIN	FLG	0	PRES	0		CLK	
Write		w1c						
Reset	0	0	0	0	0	0	0	1

WDOG_CS2 field descriptions

Field	Description
7 WIN	Watchdog Window This write-once bit enables window mode. See the Window mode section. 0 Window mode disabled. 1 Window mode enabled.
6 FLG	Watchdog Interrupt Flag This bit is an interrupt indicator when INT is set in control and status register 1. Write 1 to clear it. 0 No interrupt occurred. 1 An interrupt occurred.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PRES	Watchdog Prescalar This write-once bit enables a fixed 256 pre-scaling of watchdog counter reference clock. (The block diagram shows this clock divider option.) 0 256 prescalar disabled. 1 256 prescalar enabled.
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLK	Watchdog Clock This write-once field indicates the clock source that feeds the watchdog counter. See the Clock source section. 00 Bus clock. 01 1 kHz internal low-power oscillator (LPOCLK). 10 32 kHz internal oscillator (ICSIRCLK). 11 External clock source.

23.3 Functional description

The WDOG module provides a fail safe mechanism to ensure the system can be reset to a known state of operation in case of system failure, such as the CPU clock stopping or there being a run away condition in the software code. The watchdog counter runs continuously off a selectable clock source and expects to be serviced (refreshed) periodically. If it is not, it resets the system.

The timeout period, window mode, and clock source are all programmable but must be configured within 128 bus clocks after a reset.

23.3.1 Watchdog refresh mechanism

The watchdog resets the MCU if the watchdog counter is not refreshed. A robust refresh mechanism makes it very unlikely that the watchdog can be refreshed by runaway code.

To refresh the watchdog counter, software must execute a refresh write sequence before the timeout period expires. In addition, if window mode is used, software must not start the refresh sequence until after the time value set in the WDOG_WINH and WDOG_WINL registers. See the following figure.