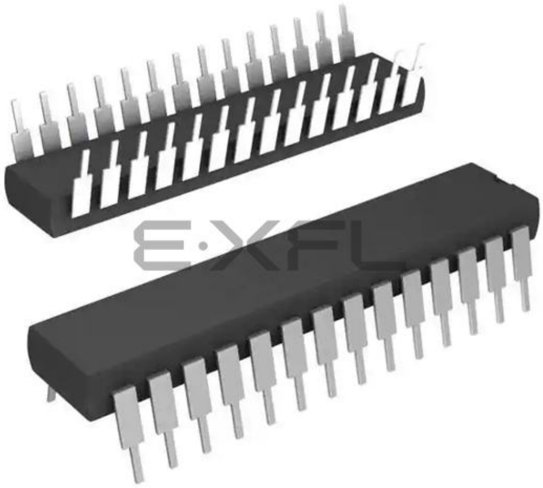


Welcome to E-XFL.COM

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	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LED, LVD, POR, PWM
Number of I/O	23
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 12x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908jl3empe

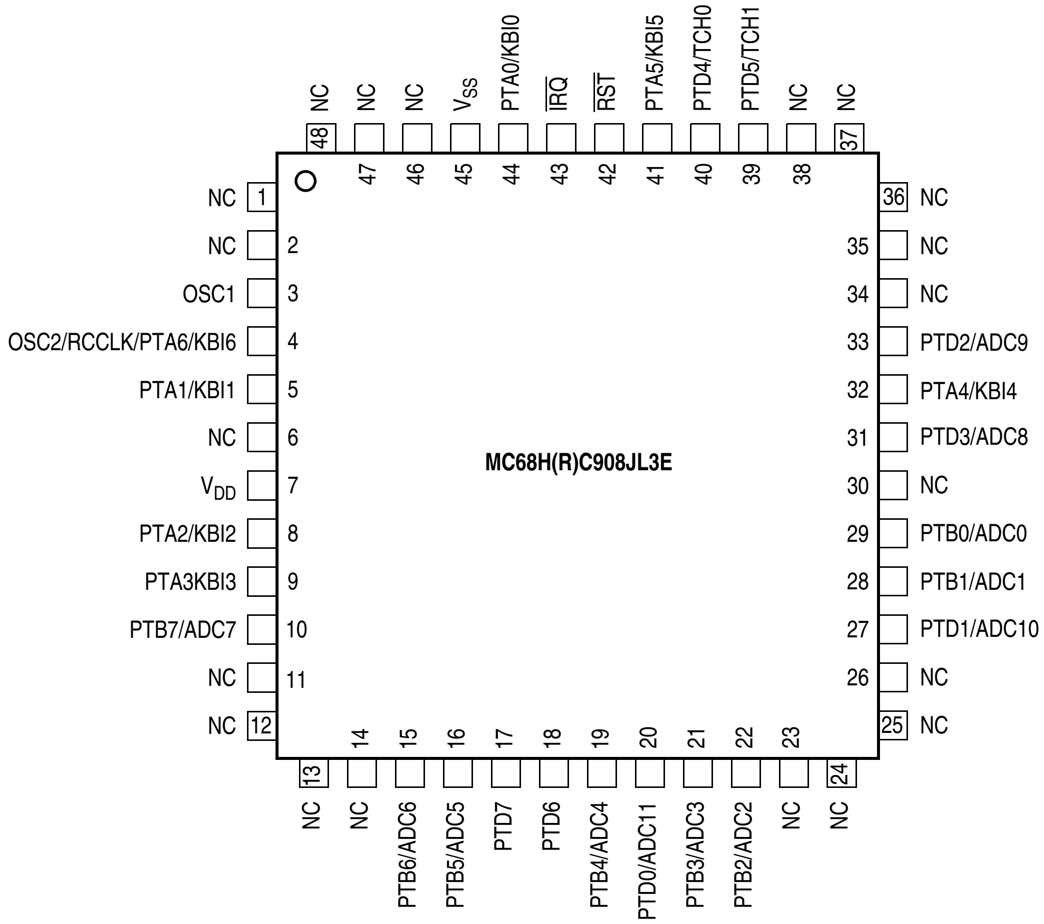
MC68HC908JL3/JK3E/JK1E
MC68HRC908JL3/JK3E/JK1E
MC68HLC908JL3/JK3E/JK1E
MC68HC908KL3E/KK3E
MC68HC08JL3E/JK3E
MC68HRC08JL3E/JK3E

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.freescale.com

Chapter 5
System Integration Module (SIM)



NC: No connection

Figure 1-4. 48-Pin LQFP Pin Assignment

2.10 Flash Program Operation

Programming of the Flash memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses $XX00, $XX20, $XX40, $XX60, $XX80, $XXA0, $XXC0 or $XXE0. Use this step-by-step procedure to program a row of Flash memory (Figure 2-5 shows a flowchart of the programming algorithm):

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any Flash location within the address range of the row to be programmed.
3. Wait for a time, t_{nvs} (10µs).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (5µs).
6. Write data to the byte being programmed.
7. Wait for time, t_{PROG} (30µs).
8. Repeat step 6 and 7 until all the bytes within the row are programmed.
9. Clear the PGM bit.
10. Wait for time, t_{nvh} (5µs).
11. Clear the HVEN bit.
12. After time, t_{rcv} (1µs), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

NOTE

The time between each Flash address change (step 6 to step 6), or the time between the last Flash addressed programmed to clearing the PGM bit (step 6 to step 10), must not exceed the maximum programming time, t_{PROG} max.

NOTE

Programming and erasing of Flash locations cannot be performed by code being executed from the Flash memory. While these operations must be performed in the order shown, other unrelated operations may occur between the steps.

2.11 Flash Protection

Due to the ability of the on-board charge pump to erase and program the Flash memory in the target application, provision is made to protect blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a Flash Block Protect Register (FLBPR). The FLBPR determines the range of the Flash memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the Flash memory ($FFFF). When the memory is protected, the HVEN bit cannot be set in either ERASE or PROGRAM operations.

LVID — Low Voltage Inhibit Disable Bit

1 = Low Voltage Inhibit disabled
0 = Low Voltage Inhibit enabled

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 × 2OSCOUT cycles instead of a 4096 × 2OSCOUT cycle delay.

1 = Stop mode recovery after 32 × 2OSCOUT cycles
0 = Stop mode recovery after 4096 × 2OSCOUT cycles

NOTE

Exiting stop mode by pulling reset will result in the long stop recovery.

If using an external crystal, do not set the SSREC bit.

STOP — STOP Instruction Enable

STOP enables the STOP instruction.

1 = STOP instruction enabled
0 = STOP instruction treated as illegal opcode

COPD — COP Disable Bit

COPD disables the COP module. (See Chapter 13 Computer Operating Properly (COP).)

1 = COP module disabled
0 = COP module enabled

3.4 Configuration Register 2 (CONFIG2)

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	IRQPUD	R	R	LVIT1	LVIT0	R	R	R
Reset:	0	0	0	Not affected	Not affected	0	0	0
POR:	0	0	0	0	0	0	0	0

R = Reserved

Figure 3-2. Configuration Register 2 (CONFIG2)

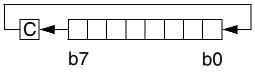
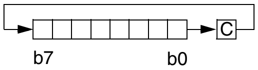
IRQPUD — $\overline{\text{IRQ}}$ Pin Pull-up control bit

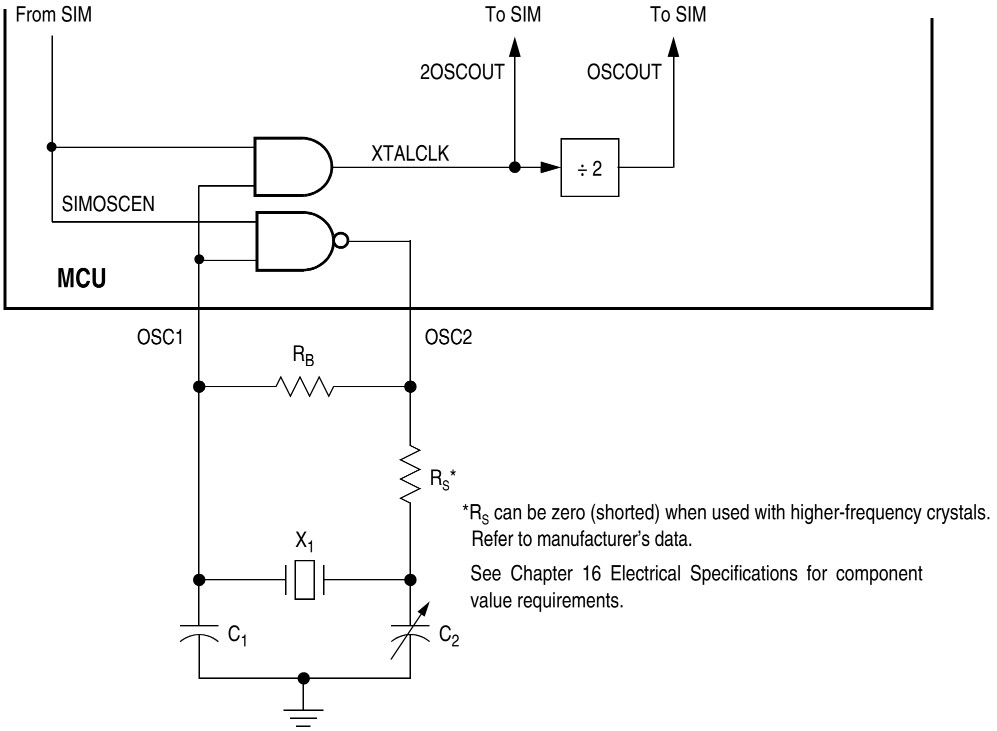
1 = Internal pull-up is disconnected
0 = Internal pull-up is connected between $\overline{\text{IRQ}}$ pin and V_{DD}

LVIT1, LVIT0 — Low Voltage Inhibit trip voltage selection bits

Detail description of the LVI control signals is given in Chapter 14 Low Voltage Inhibit (LVI)

Table 4-1. Instruction Set Summary (Sheet 5 of 6)

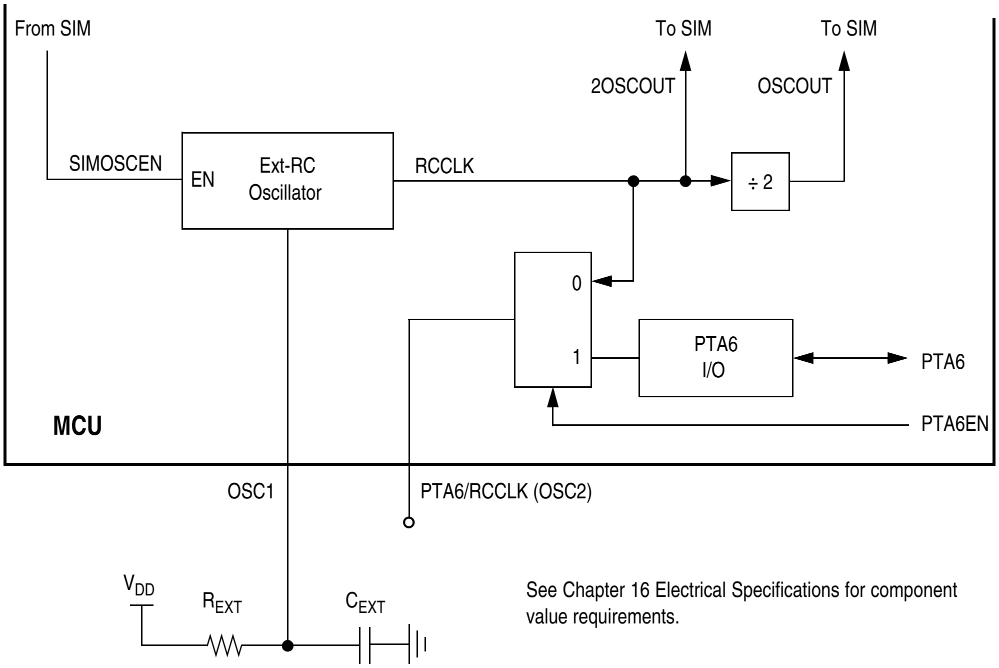
Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
PULA	Pull A from Stack	SP ← (SP + 1); Pull (A)	–	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	SP ← (SP + 1); Pull (H)	–	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	SP ← (SP + 1); Pull (X)	–	–	–	–	–	–	INH	88		2
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X ROL *opr*,SP	Rotate Left through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X ROR *opr*,SP	Rotate Right through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	SP ← $FF	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X SBC *opr*,SP SBC *opr*,SP	Subtract with Carry	A ← (A) – (M) – (C)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X STA *opr*,SP STA *opr*,SP	Store A in M	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
STHX *opr*	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↕	↕	–	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	I ← 0; Stop Processing	–	–	0	–	–	–	INH	8E		1
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X STX *opr*,SP STX *opr*,SP	Store X in M	M ← (X)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X SUB *opr*,SP SUB *opr*,SP	Subtract	A ← (A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5



*R_S can be zero (shorted) when used with higher-frequency crystals. Refer to manufacturer's data.

See Chapter 16 Electrical Specifications for component value requirements.

Figure 6-1. X-tal Oscillator External Connections



See Chapter 16 Electrical Specifications for component value requirements.

Figure 6-2. RC Oscillator External Connections

7.3.1 Entering Monitor Mode

Table 7-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a POR and will allow communication at 9600 baud provided one of the following sets of conditions is met:

1. If $\overline{\text{IRQ}} = V_{TST}$:
   - Clock on OSC1 is 4.9125 MHz (EXT OSC or XTAL)
   - PTB3 = low
2. If $\overline{\text{IRQ}} = V_{TST}$:
   - Clock on OSC1 is 9.8304 MHz (EXT OSC or XTAL)
   - PTB3 = high
3. If $FFFE & $FFFF is blank (contains $FF):
   - Clock on OSC1 is 9.8304 MHz (EXT OSC or XTAL or RC)
   - $\overline{\text{IRQ}} = V_{DD}$

Table 7-1. Monitor Mode Entry Requirements and Options

$\overline{\text{IRQ}}$	$FFFE and $FFFF	PTB3[1]	PTB2	PTB1	PTB0	OSC1 Frequency	Bus Frequency	Comments
V_{TST}[2]	X	0	0	1	1	4.9152 MHz	2.4576 MHz (OSC1 ÷ 2)	High-voltage entry to monitor mode.[3] 9600 baud communication on PTB0. COP disabled.
V_{TST}	X	1	0	1	1	9.8304 MHz	2.4576 MHz (OSC1 ÷ 4)	
V_{DD}	BLANK (contain $FF)	X	X	X	1	9.8304 MHz	2.4576 MHz (OSC1 ÷ 4)	Low-voltage entry to monitor mode.[4] 9600 baud communication on PTB0. COP disabled.
V_{DD}	NOT BLANK	X	X	X	X	At desired frequency	OSC1 ÷ 4	Enters User mode.

[1] PTB3 = 0: Bypasses the divide-by-two prescaler to SIM when using V_{TST} for monitor mode entry. The OSC1 clock must be 50% duty cycle for this condition.
[2] See Table 16-4. DC Electrical Characteristics (5 V) for V_{TST} voltage level requirements.
[3] For $\overline{\text{IRQ}} = V_{TST}$:
MC68HRC908JL3E/JK3E/JK1E — clock must be EXT OSC.
MC68HC908JL3E/JK3E/JK1E — clock can be EXT OSC or XTAL.
[4] For $\overline{\text{IRQ}} = V_{DD}$:
MC68HRC908JL3E/JK3E/JK1E — clock must be RC OSC.
MC68HC908JL3E/JK3E/JK1E — clock can be EXT OSC or XTAL.

If V_{TST} is applied to $\overline{\text{IRQ}}$ and PTB3 is low upon monitor mode entry (Table 7-1 condition set 1), the bus frequency is a divide-by-two of the clock input to OSC1. If PTB3 is high with V_{TST} applied to $\overline{\text{IRQ}}$ upon monitor mode entry (Table 7-1 condition set 2), the bus frequency is a divide-by-four of the clock input to OSC1. Holding the PTB3 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator *only if V_{TST} is applied to $\overline{\text{IRQ}}$*. In this event, the OSCOUT frequency is equal to the 2OSCOUT frequency, and OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50% duty cycle at maximum bus frequency.

8.4.1 TIM Counter Prescaler

The TIM clock source is one of the seven prescaler outputs. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM status and control register (TSC) select the TIM clock source.

8.4.2 Input Capture

With the input capture function, the TIM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM latches the contents of the TIM counter into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM CPU interrupt requests.

8.4.3 Output Compare

With the output compare function, the TIM can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM can set, clear, or toggle the channel pin. Output compares can generate TIM CPU interrupt requests.

8.4.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 8.4.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

8.4.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that

8.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 8.4.4 Pulse Width Modulation (PWM). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

8.4.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE

In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.

TSTOP — TIM Stop Bit

This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.

1 = TIM counter stopped
0 = TIM counter active

NOTE

Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode. When the TSTOP bit is set and the timer is configured for input capture operation, input captures are inhibited until the TSTOP bit is cleared.

When using TSTOP to stop the timer counter, see if any timer flags are set. If a timer flag is set, it must be cleared by clearing TSTOP, then clearing the flag, then setting TSTOP again.

TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. Setting TRST has no effect on any other registers. Counting resumes from $0000. TRST is cleared automatically after the TIM counter is reset and always reads as zero. Reset clears the TRST bit.

1 = Prescaler and TIM counter cleared
0 = No effect

NOTE

Setting the TSTOP and TRST bits simultaneously stops the TIM counter at a value of $0000.

PS[2:0] — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM counter as Table 8-2 shows. Reset clears the PS[2:0] bits.

Table 8-2. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal Bus Clock ÷ 1
0	0	1	Internal Bus Clock ÷ 2
0	1	0	Internal Bus Clock ÷ 4
0	1	1	Internal Bus Clock ÷ 8
1	0	0	Internal Bus Clock ÷ 16
1	0	1	Internal Bus Clock ÷ 32
1	1	0	Internal Bus Clock ÷ 64
1	1	1	Not available

9.3.2 Voltage Conversion

When the input voltage to the ADC equals V_{DD}, the ADC converts the signal to $FF (full scale). If the input voltage equals V_{SS}, the ADC converts it to $00. Input voltages between V_{DD} and V_{SS} are a straight-line linear conversion. All other input voltages will result in $FF if greater than V_{DD} and $00 if less than V_{SS}.

NOTE

Input voltage should not exceed the analog supply voltages.

9.3.3 Conversion Time

Fourteen ADC internal clocks are required to perform one conversion. The ADC starts a conversion on the first rising edge of the ADC internal clock immediately following a write to the ADSCR. If the ADC internal clock is selected to run at 1 MHz, then one conversion will take 14μs to complete. With a 1 MHz ADC internal clock the maximum sample rate is 71.43 kHz.

$$\text{Conversion Time} = \frac{\text{14 ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

$$\text{Number of Bus Cycles} = \text{Conversion Time} \times \text{Bus Frequency}$$

9.3.4 Continuous Conversion

In the continuous conversion mode, the ADC continuously converts the selected channel filling the ADC data register with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit (ADC status and control register, $003C) is set after each conversion and can be cleared by writing the ADC status and control register or reading of the ADC data register.

9.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.

9.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit is at 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

9.5 Low-Power Modes

The following subsections describe the ADC in low-power modes.

9.5.1 Wait Mode

The ADC continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the ADCH[4:0] bits in the ADC status and control register to 1’s before executing the WAIT instruction.

9.5.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before attempting a new ADC conversion after exiting stop mode.

9.6 I/O Signals

The ADC module has 12 channels that are shared with I/O port B and port D.

9.6.1 ADC Voltage In (ADCVIN)

ADCVIN is the input voltage signal from one of the 12 ADC channels to the ADC module.

9.7 I/O Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR)
- ADC data register (ADR)
- ADC clock register (ADICLK)

9.7.1 ADC Status and Control Register

The following paragraphs describe the function of the ADC status and control register.

Address: $003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:	▨							
Reset:	0	0	0	1	1	1	1	1

▨ = Unimplemented

Figure 9-3. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

When the AIEN bit is a 0, the COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the ADC status and control register is written or whenever the ADC data register is read. Reset clears this bit.

1 = Conversion completed (AIEN = 0)
0 = Conversion not completed (AIEN = 0)

When the AIEN bit is a 1 (CPU interrupt enabled), the COCO is a read-only bit, and will always be 0 when read.

AIEN — ADC Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of an ADC conversion. The interrupt signal is cleared when the data register is read or the status/control register is written. Reset clears the AIEN bit.

1 = ADC interrupt enabled
0 = ADC interrupt disabled

10.3 Port B

Port B is an 8-bit special function port that shares all eight of its port pins with the analog-to-digital converter (ADC) module, see Chapter 9 Analog-to-Digital Converter (ADC).

10.3.1 Port B Data Register (PTB)

The port B data register contains a data latch for each of the eight port B pins.

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							
Alternative Function:	ADC7	ADC6	AD4C5	ADC4	ADC3	ADC2	ADC2	ADC0

Figure 10-6. Port B Data Register (PTB)

PTB[7:0] — Port B Data Bits

These read/write bits are software programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

ADC[7:0] — ADC channels 7 to 0

ADC[7:0] are pins used for the input channels to the analog-to-digital converter module. The channel select bits, ADCH[4:0], in the ADC status and control register define which port pin will be used as an ADC input and overrides any control from the port I/O logic. See Chapter 9 Analog-to-Digital Converter (ADC).

10.3.2 Data Direction Register B (DDRB)

Data direction register B determines whether each port B pin is an input or an output. Writing a one to a DDRB bit enables the output buffer for the corresponding port B pin; a zero disables the output buffer.

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 10-7. Data Direction Register B (DDRB)

DDRB[7:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:0], configuring all port B pins as inputs.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1.

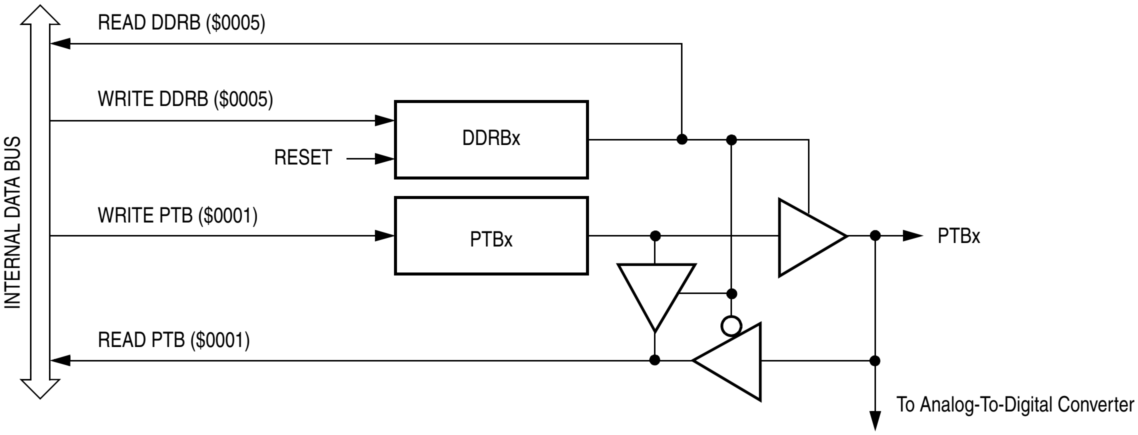


Figure 10-8. Port B I/O Circuit

When DDRBx is a 1, reading address $0001 reads the PTBx data latch. When DDRBx is a 0, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 10-3 summarizes the operation of the port B pins.

Table 10-3. Port B Pin Functions

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
			Read/Write	Read	Write
0	X[1]	Input, Hi-Z[2]	DDRB[7:0]	Pin	PTB[7:0][3]
1	X	Output	DDRB[7:0]	Pin	PTB[7:0]

1. X = don’t care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect the input.

11.3.1 $\overline{\text{IRQ}}$ Pin

A zero on the $\overline{\text{IRQ}}$ pin can latch an interrupt request into the IRQ latch. A vector fetch, software clear, or reset clears the IRQ latch.

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE set, both of the following actions must occur to clear IRQ:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic one to the ACK bit in the interrupt status and control register (INTSCR). The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to the ACK bit latches another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at locations $FFFA and $FFFB.
- Return of the $\overline{\text{IRQ}}$ pin to logic one — As long as the $\overline{\text{IRQ}}$ pin is at logic zero, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic one may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic zero. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the INTSCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ}}$ pin.

NOTE

When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.

NOTE

An internal pull-up resistor to V_{DD} is connected to the $\overline{\text{IRQ}}$ pin; this can be disabled by setting the IRQPUD bit in the CONFIG2 register ($001E).

11.4 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether the IRQ latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear the latches during the break state. (See Chapter 5 System Integration Module (SIM).)

To allow software to clear the IRQ latch during a break interrupt, write a one to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the latches during the break state, write a zero to the BCFE bit. With BCFE at zero (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ latch.

12.5.1 Keyboard Status and Control Register

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

Address: $001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:	▨	▨	▨	▨	▨	ACKK		
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 12-3. Keyboard Status and Control Register (KBSCR)

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port-A. Reset clears the KEYF bit.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the keyboard interrupt request on port-A. ACKK always reads as 0. Reset clears ACKK.

IMASKK— Keyboard Interrupt Mask Bit

Writing a 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests on port-A. Reset clears the IMASKK bit.

1 = Keyboard interrupt requests masked
0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins on port-A. Reset clears MODEK.

1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

15.4 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- Break status register (BSR)
- Break flag control register (BFCR)

15.4.1 Break Status and Control Register (BRKSCR)

The break status and control register contains break module enable and status bits.

Address: $FE0E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:			▨	▨	▨	▨	▨	▨
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 15-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a zero to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a one to BRKA generates a break interrupt. Clear BRKA by writing a zero to it before exiting the break routine. Reset clears the BRKA bit.

1 = Break address match
0 = No break address match

Table 16-4. DC Electrical Characteristics (5 V) (Continued)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
LVI reset voltage	V_{LVR5}	3.6	4.0	4.4	V

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 grounded; no port pins sourcing current. LVI is disabled.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{\text{RST}}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0 V.

16.6 5V Control Timing

Table 16-5. Control Timing (5 V)

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	8	MHz
$\overline{\text{RST}}$ input pulse width low[3]	t_{IRL}	750	—	ns

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{SS}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for this information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

A.5.6 Memory Characteristics

The Flash memory can only be read at an operating voltage of 2.2 to 5.5V. Program and erase are achieved at an operating voltage of 2.7 to 5.5V. The program and erase parameters in Table A-6 are for V_{DD} = 2.7 to 5.5V only.

Table A-6. Memory Characteristics

Characteristic	Symbol	Min	Max	Unit
RAM data retention voltage	V_{RDR}	1.3	—	V
Flash program bus clock frequency	—	1	—	MHz
Flash read bus clock frequency	f_{Read}[1]	32k	8M	Hz
Flash page erase time	t_{Erase}[2]	1	—	ms
Flash mass erase time	t_{MErase}[3]	4	—	ms
Flash PGM/ERASE to HVEN set up time	t_{nvs}	10	—	µs
Flash high-voltage hold time	t_{nvh}	5	—	µs
Flash high-voltage hold time (mass erase)	t_{nvhl}	100	—	µs
Flash program hold time	t_{pgs}	5	—	µs
Flash program time	t_{PROG}	30	40	µs
Flash return to read time	t_{rcv}[4]	1	—	µs
Flash cumulative program hv period	t_{HV}[5]	—	4	ms
Flash row erase endurance[6]	—	10k	—	cycles
Flash row program endurance[7]	—	10k	—	cycles
Flash data retention time[8]	—	10	—	years

1. f_{Read} is defined as the frequency range for which the Flash memory can be read.
2. If the page erase time is longer than t_{Erase} (Min), there is no erase-disturb, but it reduces the endurance of the Flash memory.
3. If the mass erase time is longer than t_{MErase} (Min), there is no erase-disturb, but it reduces the endurance of the Flash memory.
4. t_{rcv} is defined as the time it needs before the Flash can be read after turning off the high voltage charge pump, by clearing HVEN to 0.
5. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase.
   t_{HV} must satisfy this condition: $t_{nvs} + t_{nvh} + t_{pgs} + (t_{PROG} \times 32) \leq t_{HV}$ max.
6. The minimum row endurance value specifies each row of the Flash memory is guaranteed to work for at least this many erase / program cycles.
7. The minimum row endurance value specifies each row of the Flash memory is guaranteed to work for at least this many erase / program cycles.
8. The Flash is guaranteed to retain data over the entire operating temperature range for at least the minimum time specified.