



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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	37
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68908gz8cfae

4.4	I/O Signals	67
4.4.1	Crystal Amplifier Input Pin (OSC1)	67
4.4.2	Crystal Amplifier Output Pin (OSC2)	67
4.4.3	External Filter Capacitor Pin (CGMXFC)	67
4.4.4	PLL Analog Power Pin (V_{DDA})	67
4.4.5	PLL Analog Ground Pin (V_{SSA})	67
4.4.6	Oscillator Enable Signal (SIMOSCEN)	67
4.4.7	Oscillator Stop Mode Enable Bit (OSCSTOPENB)	67
4.4.8	Crystal Output Frequency Signal (CGMXCLK)	68
4.4.9	CGM Base Clock Output (CGMOUT)	68
4.4.10	CGM CPU Interrupt (CGMINT)	68
4.5	CGM Registers	68
4.5.1	PLL Control Register	69
4.5.2	PLL Bandwidth Control Register	71
4.5.3	PLL Multiplier Select Register High	72
4.5.4	PLL Multiplier Select Register Low	73
4.5.5	PLL VCO Range Select Register	73
4.6	Interrupts	74
4.7	Special Modes	75
4.7.1	Wait Mode	75
4.7.2	Stop Mode	75
4.7.3	CGM During Break Interrupts	75
4.8	Acquisition/Lock Time Specifications	75
4.8.1	Acquisition/Lock Time Definitions	75
4.8.2	Parametric Influences on Reaction Time	76
4.8.3	Choosing a Filter	76

Chapter 5

Configuration Register (CONFIG)

5.1	Introduction	79
5.2	Functional Description	79

Chapter 6

Computer Operating Properly (COP) Module

6.1	Introduction	83
6.2	Functional Description	83
6.3	I/O Signals	84
6.3.1	CGMXCLK	84
6.3.2	STOP Instruction	84
6.3.3	COPCTL Write	84
6.3.4	Power-On Reset	84
6.3.5	Internal Reset	84
6.3.6	Reset Vector Fetch	85
6.3.7	COPD (COP Disable)	85
6.3.8	COPRS (COP Rate Select)	85
6.4	COP Control Register	85
6.5	Interrupts	85
6.6	Monitor Mode	85

Chapter 10

Low-Power Modes

10.1	Introduction	109
10.1.1	Wait Mode	109
10.1.2	Stop Mode	109
10.2	Analog-to-Digital Converter (ADC)	109
10.2.1	Wait Mode	109
10.2.2	Stop Mode	109
10.3	Break Module (BRK)	110
10.3.1	Wait Mode	110
10.3.2	Stop Mode	110
10.4	Central Processor Unit (CPU)	110
10.4.1	Wait Mode	110
10.4.2	Stop Mode	110
10.5	Clock Generator Module (CGM)	110
10.5.1	Wait Mode	110
10.5.2	Stop Mode	110
10.6	Computer Operating Properly Module (COP)	111
10.6.1	Wait Mode	111
10.6.2	Stop Mode	111
10.7	External Interrupt Module (IRQ)	111
10.7.1	Wait Mode	111
10.7.2	Stop Mode	111
10.8	Keyboard Interrupt Module (KBI)	111
10.8.1	Wait Mode	111
10.8.2	Stop Mode	111
10.9	Low-Voltage Inhibit Module (LVI)	112
10.9.1	Wait Mode	112
10.9.2	Stop Mode	112
10.10	Enhanced Serial Communications Interface Module (ESCI)	112
10.10.1	Wait Mode	112
10.10.2	Stop Mode	112
10.11	Serial Peripheral Interface Module (SPI)	112
10.11.1	Wait Mode	112
10.11.2	Stop Mode	112
10.12	Timer Interface Module (TIM1 and TIM2)	113
10.12.1	Wait Mode	113
10.12.2	Stop Mode	113
10.13	Timebase Module (TBM)	113
10.13.1	Wait Mode	113
10.13.2	Stop Mode	113
10.14	MSCAN	113
10.14.1	Wait Mode	113
10.14.2	Stop Mode	113
10.15	Exiting Wait Mode	114
10.16	Exiting Stop Mode	115

15.7	I/O Signals	195
15.7.1	PTE0/TxD (Transmit Data)	195
15.7.2	PTE1/RxD (Receive Data)	195
15.8	I/O Registers	196
15.8.1	ESCI Control Register 1	196
15.8.2	ESCI Control Register 2	198
15.8.3	ESCI Control Register 3	200
15.8.4	ESCI Status Register 1	201
15.8.5	ESCI Status Register 2	203
15.8.6	ESCI Data Register	204
15.8.7	ESCI Baud Rate Register	204
15.8.8	ESCI Prescaler Register	206
15.9	ESCI Arbiter	209
15.9.1	ESCI Arbiter Control Register	209
15.9.2	ESCI Arbiter Data Register	210
15.9.3	Bit Time Measurement	210
15.9.4	Arbitration Mode	212

Chapter 16

System Integration Module (SIM)

16.1	Introduction	213
16.2	SIM Bus Clock Control and Generation	215
16.2.1	Bus Timing	215
16.2.2	Clock Startup from POR or LVI Reset	215
16.2.3	Clocks in Stop Mode and Wait Mode	215
16.3	Reset and System Initialization	216
16.3.1	External Pin Reset	216
16.3.2	Active Resets from Internal Sources	217
16.3.2.1	Power-On Reset	217
16.3.2.2	Computer Operating Properly (COP) Reset	218
16.3.2.3	Illegal Opcode Reset	218
16.3.2.4	Illegal Address Reset	218
16.3.2.5	Low-Voltage Inhibit (LVI) Reset	218
16.3.2.6	Monitor Mode Entry Module Reset (MODRST)	219
16.4	SIM Counter	219
16.4.1	SIM Counter During Power-On Reset	219
16.4.2	SIM Counter During Stop Mode Recovery	219
16.4.3	SIM Counter and Reset States	219
16.5	Exception Control	219
16.5.1	Interrupts	219
16.5.1.1	Hardware Interrupts	222
16.5.1.2	SWI Instruction	223
16.5.1.3	Interrupt Status Registers	223
16.5.2	Reset	225
16.5.3	Break Interrupts	225
16.5.4	Status Flag Protection in Break Mode	225

- Specific features of the MC68HC908GZ16 in 32-pin LQFP are:
 - Port A is only 4 bits: PTA0–PTA3; 4-pin keyboard interrupt (KBI) module
 - Port B is only 6 bits: PTB0–PTB5; 6-channel ADC module
 - Port C is only 2 bits: PTC0–PTC1; shared with MSCAN08 module
 - Port D is only 7 bits: PTD0–PTD6; shared with SPI, TIM1, and TIM2 modules
 - Port E is only 2 bits: PTE0–PTE1; shared with ESCI module
- Specific features of the MC68HC908GZ16 in 48-pin LQFP are:
 - Port A is 8 bits: PTA0–PTA7; 8-pin KBI module
 - Port B is 8 bits: PTB0–PTB7; 8-channel ADC module
 - Port C is only 7 bits: PTC0–PTC6; shared with MSCAN08 module
 - Port D is 8 bits: PTD0–PTD7; shared with SPI, TIM1, and TIM2 modules
 - Port E is only 6 bits: PTE0–PTE5; shared with ESCI module

1.2.2 Features of the CPU08

Features of the CPU08 include:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit index register and stack pointer
- Memory-to-memory data transfers
- Fast 8×8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.3 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908GZ16.

1.4 Pin Assignments

Figure 1-2 and Figure 1-3 illustrate the pin assignments for the 32-pin LQFP and 48-pin LQFP respectively.

4.3.9 CGM External Connections

In its typical configuration, the CGM requires external components. Five of these are for the crystal oscillator and two or four are for the PLL.

The crystal oscillator is normally connected in a Pierce oscillator configuration, as shown in Figure 4-2. Figure 4-2 shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

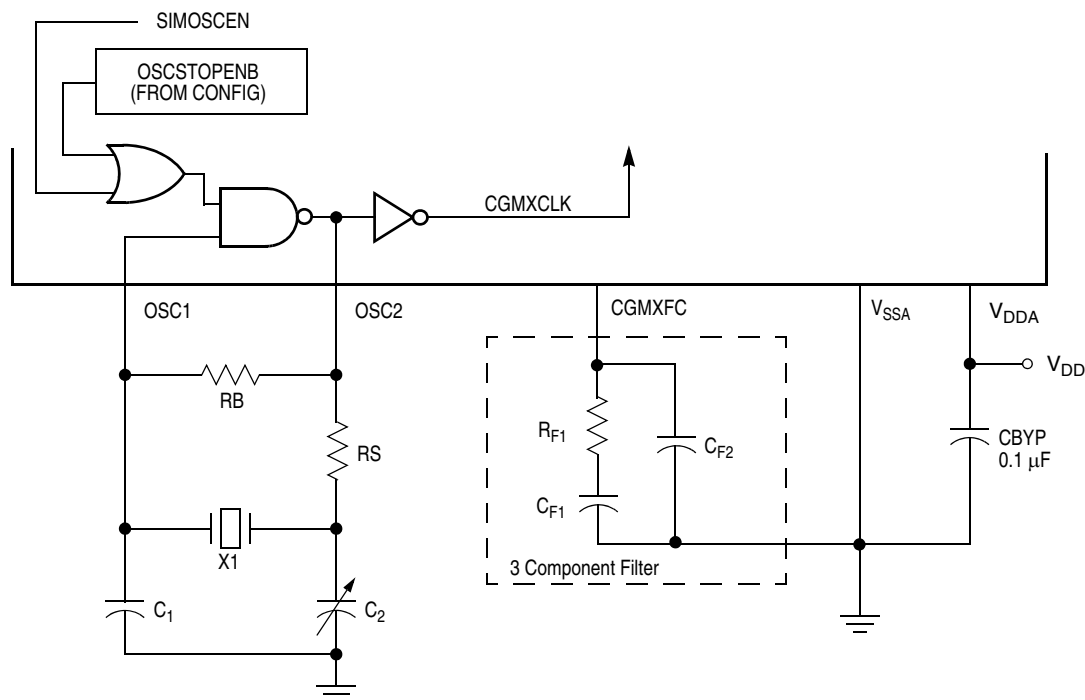
- Crystal, X_1
- Fixed capacitor, C_1
- Tuning capacitor, C_2 (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_S

The series resistor (R_S) is included in the diagram to follow strict Pierce oscillator guidelines. Refer to the crystal manufacturer's data for more information regarding values for C_1 and C_2 .

Figure 4-2 also shows the external components for the PLL:

- Bypass capacitor, C_{BYP}
- Filter network

Routing should be done with great care to minimize signal cross talk and noise.



Note: Filter network in box can be replaced with a single capacitor, but will degrade stability.

Figure 4-2. CGM External Connections

Chapter 9

Keyboard Interrupt Module (KBI)

9.1 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA7. When a port pin is enabled for keyboard interrupt function, an internal pullup device is also enabled on the pin.

9.2 Features

Features include:

- Eight keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level- interrupt sensitivity
- Exit from low-power modes
- I/O (input/output) port bit(s) software configurable with pullup device(s) if configured as input port bit(s)

9.3 Functional Description

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin also enables its internal pullup device. A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard pins goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard pin does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one pin because another pin is still low, software can disable the latter pin while it is low.
- If the keyboard interrupt is falling edge- and low-level sensitive, an interrupt request is present as long as any keyboard interrupt pin is low and the pin is keyboard interrupt enabled.

Chapter 11

Low-Voltage Inhibit (LVI)

11.1 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls below the LVI trip falling voltage, V_{TRIPF} .

11.2 Features

Features of the LVI module include:

- Programmable LVI reset
- Selectable LVI trip voltage
- Programmable stop mode operation

11.3 Functional Description

Figure 11-1 shows the structure of the LVI module. The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit, $LVIPWRD$, enables the LVI to monitor V_{DD} voltage. Clearing the LVI reset disable bit, $LVIRSTD$, enables the LVI module to generate a reset when V_{DD} falls below a voltage, V_{TRIPF} . Setting the LVI enable in stop mode bit, $LVISTOP$, enables the LVI to operate in stop mode. Setting the LVI 5-V or 3-V trip point bit, $LVI5OR3$, enables the trip point voltage, V_{TRIPF} , to be configured for 5-V operation. Clearing the $LVI5OR3$ bit enables the trip point voltage, V_{TRIPF} , to be configured for 3-V operation. The actual trip points are shown in Chapter 21 Electrical Specifications.

NOTE

After a power-on reset (POR) the LVI's default mode of operation is 3 V. If a 5-V system is used, the user must set the $LVI5OR3$ bit to raise the trip point to 5-V operation. Note that this must be done after every power-on reset since the default will revert back to 3-V mode after each power-on reset. If the V_{DD} supply is below the 5-V mode trip voltage but above the 3-V mode trip voltage when POR is released, the part will operate because V_{TRIPF} defaults to 3-V mode after a POR. So, in a 5-V system care must be taken to ensure that V_{DD} is above the 5-V mode trip voltage after POR is released.

If the user requires 5-V mode and sets the $LVI5OR3$ bit after a power-on reset while the V_{DD} supply is not above the V_{TRIPR} for 5-V mode, the microcontroller unit (MCU) will immediately go into reset. The LVI in this case will hold the part in reset until either V_{DD} goes above the rising 5-V trip point, V_{TRIPR} , which will release reset or V_{DD} decreases to approximately 0 V which will re-trigger the power-on reset and reset the trip point to 3-V operation.

Address:	\$0005							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-7. Data Direction Register B (DDRB)

DDRB7–DDRB0 — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB7–DDRB0, 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 13-8 shows the port B I/O logic.

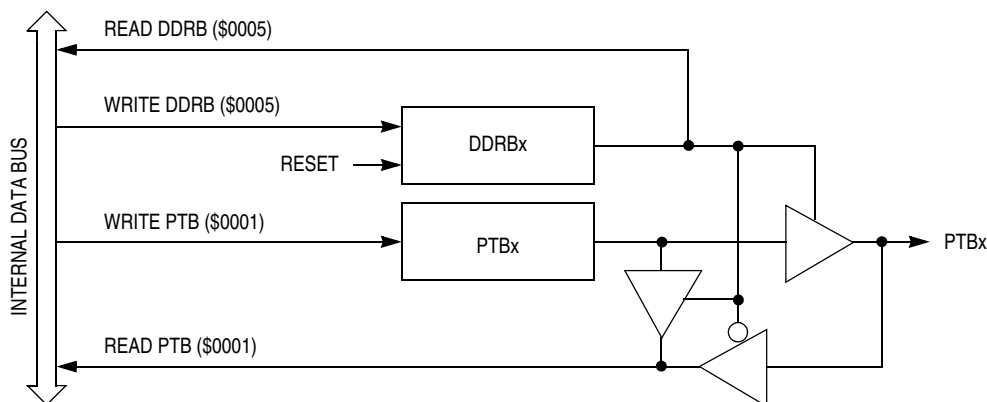


Figure 13-8. Port B I/O Circuit

When bit DDRBx is a logic 1, reading address \$0001 reads the PTBx data latch. When bit DDRBx is a logic 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 13-3 summarizes the operation of the port B pins.

Table 13-3. 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⁾	DDRB7–DDRB0	Pin	PTB7–PTB0 ⁽³⁾
1	X	Output	DDRB7–DDRB0	PTB7–PTB0	PTB7–PTB0

1. X = Don't care

2. Hi-Z = High impedance

3. Writing affects data register, but does not affect input.

T1CH1 and T1CH0 — Timer 1 Channel I/O Bits

The PTD7/T1CH1–PTD6/T1CH0 pins are the TIM1 input capture/output compare pins. The edge/level select bits, ELSxB and ELSxA, determine whether the PTD7/T1CH1–PTD6/T1CH0 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 19 Timer Interface Module (TIM).

SPSCK — SPI Serial Clock

The PTD3/SPSCK pin is the serial clock input of the SPI module. When the SPE bit is clear, the PTD3/SPSCK pin is available for general-purpose I/O.

MOSI — Master Out/Slave In

The PTD2/MOSI pin is the master out/slave in terminal of the SPI module. When the SPE bit is clear, the PTD2/MOSI pin is available for general-purpose I/O.

MISO — Master In/Slave Out

The PTD1/MISO pin is the master in/slave out terminal of the SPI module. When the SPI enable bit, SPE, is clear, the SPI module is disabled, and the PTD0/ \overline{SS} pin is available for general-purpose I/O.

Data direction register D (DDRD) does not affect the data direction of port D pins that are being used by the SPI module. However, the DDRD bits always determine whether reading port D returns the states of the latches or the states of the pins. See Table 13-5.

\overline{SS} — Slave Select

The PTD0/ \overline{SS} pin is the slave select input of the SPI module. When the SPE bit is clear, or when the SPI master bit, SPMSTR, is set, the PTD0/ \overline{SS} pin is available for general-purpose I/O. When the SPI is enabled, the DDRB0 bit in data direction register B (DDRB) has no effect on the PTD0/ \overline{SS} pin.

13.6.2 Data Direction Register D

Data direction register D (DDRD) determines whether each port D pin is an input or an output. Writing a logic 1 to a DDRD bit enables the output buffer for the corresponding port D pin; a logic 0 disables the output buffer.

Address:	\$0007							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-14. Data Direction Register D (DDRD)

DDRD7–DDRD0 — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD7–DDRD0, configuring all port D pins as inputs.

- 1 = Corresponding port D pin configured as output
- 0 = Corresponding port D pin configured as input

NOTE

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

Figure 13-15 shows the port D I/O logic.

After every instruction, the CPU checks all pending interrupts if the I bit is not set. If more than one interrupt is pending when an instruction is done, the highest priority interrupt is serviced first. In the example shown in Figure 14-4, if an interrupt is pending upon exit from the interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

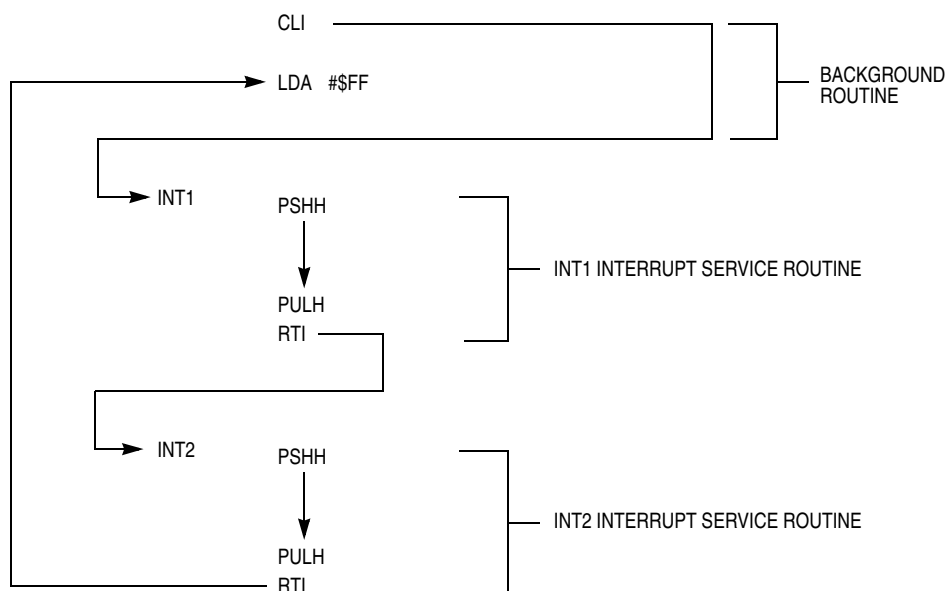


Figure 14-4. Interrupt Recognition Example

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE

To maintain compatibility with the M6805 Family, the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, save the H register and then restore it prior to exiting the routine.

See Figure 14-5 for a flowchart depicting interrupt processing.

14.3.2 Sources

The sources in Table 14-1 can generate CPU interrupt requests.

14.3.2.1 Software Interrupt (SWI) Instruction

The software interrupt (SWI) instruction causes a non-maskable interrupt.

NOTE

*A software interrupt pushes PC onto the stack. An SWI does **not** push PC – 1, as a hardware interrupt does.*

14.3.2.2 Break Interrupt

The break module causes the CPU to execute an SWI instruction at a software-programmable break point.

Enhanced Serial Communications Interface (ESCI) Module

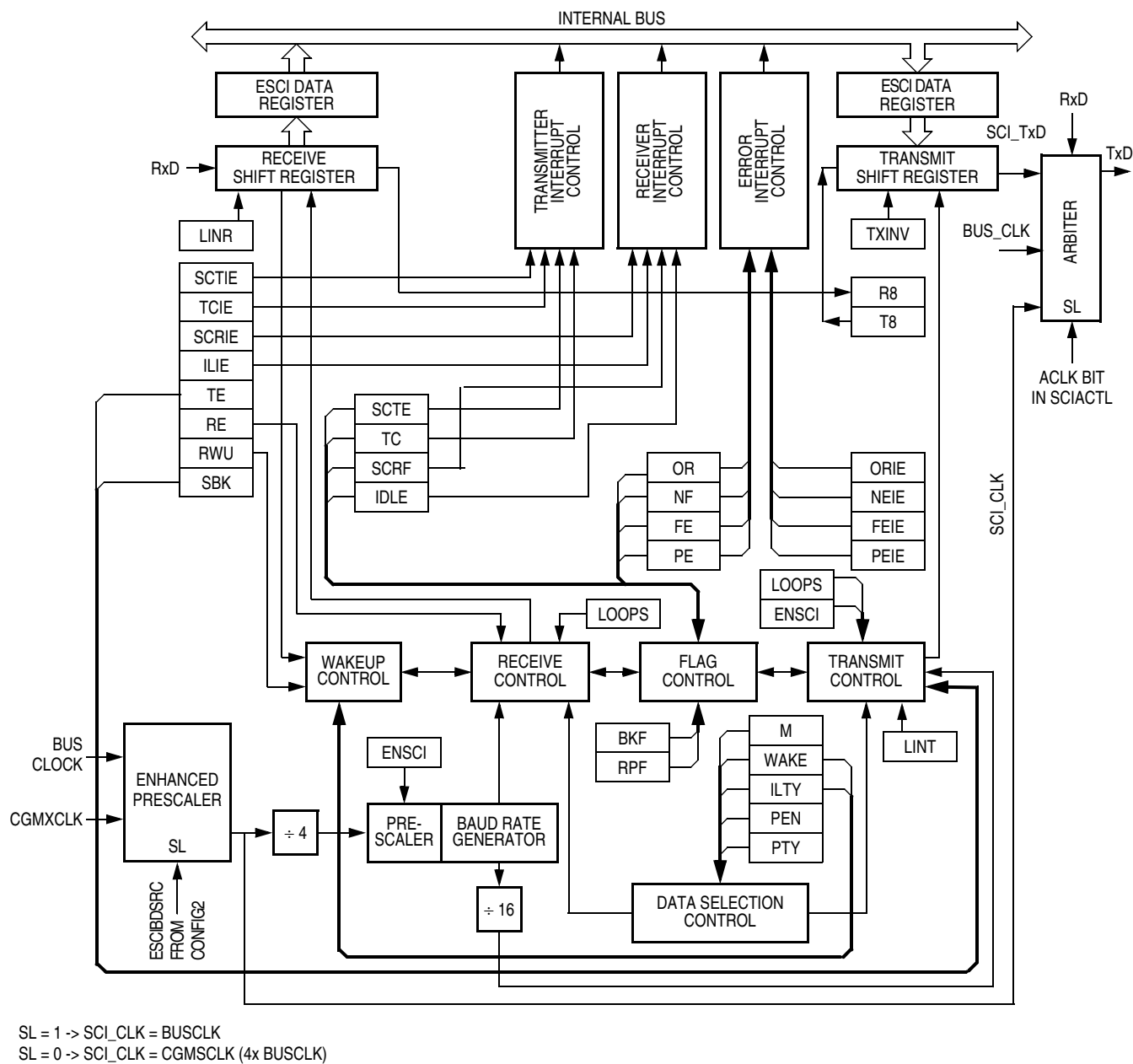


Figure 15-2. ESCI Module Block Diagram

Depending on the state of the WAKE bit in SCC1, either of two conditions on the RxD pin can bring the receiver out of the standby state:

1. Address mark — An address mark is a logic 1 in the MSB position of a received character. When the WAKE bit is set, an address mark wakes the receiver from the standby state by clearing the RWU bit. The address mark also sets the ESCI receiver full bit, SCRF. Software can then compare the character containing the address mark to the user-defined address of the receiver. If they are the same, the receiver remains awake and processes the characters that follow. If they are not the same, software can set the RWU bit and put the receiver back into the standby state.
2. Idle input line condition — When the WAKE bit is clear, an idle character on the RxD pin wakes the receiver from the standby state by clearing the RWU bit. The idle character that wakes the receiver does not set the receiver idle bit, IDLE, or the ESCI receiver full bit, SCRF. The idle line type bit, ILTY, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit.

NOTE

With the WAKE bit clear, setting the RWU bit after the RxD pin has been idle will cause the receiver to wake up.

15.4.3.7 Receiver Interrupts

These sources can generate CPU interrupt requests from the ESCI receiver:

- ESCI receiver full (SCRF) — The SCRF bit in SCS1 indicates that the receive shift register has transferred a character to the SCDR. SCRF can generate a receiver CPU interrupt request. Setting the ESCI receive interrupt enable bit, SCRIE, in SCC2 enables the SCRF bit to generate receiver CPU interrupts.
- Idle input (IDLE) — The IDLE bit in SCS1 indicates that 10 or 11 consecutive logic 1s shifted in from the RxD pin. The idle line interrupt enable bit, ILIE, in SCC2 enables the IDLE bit to generate CPU interrupt requests.

15.4.3.8 Error Interrupts

These receiver error flags in SCS1 can generate CPU interrupt requests:

- Receiver overrun (OR) — The OR bit indicates that the receive shift register shifted in a new character before the previous character was read from the SCDR. The previous character remains in the SCDR, and the new character is lost. The overrun interrupt enable bit, ORIE, in SCC3 enables OR to generate ESCI error CPU interrupt requests.
- Noise flag (NF) — The NF bit is set when the ESCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, in SCC3 enables NF to generate ESCI error CPU interrupt requests.
- Framing error (FE) — The FE bit in SCS1 is set when a logic 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate ESCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in SCS1 is set when the ESCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate ESCI error CPU interrupt requests.

Table 15-11 shows the ESCI baud rates that can be generated with a 4.9152-MHz bus frequency.

Table 15-10. ESCI Prescaler Divisor Fine Adjust

PSSB[4:3:2:1:0]	Prescaler Divisor Fine Adjust (PDFA)
0 0 0 0 0	0/32 = 0
0 0 0 0 1	1/32 = 0.03125
0 0 0 1 0	2/32 = 0.0625
0 0 0 1 1	3/32 = 0.09375
0 0 1 0 0	4/32 = 0.125
0 0 1 0 1	5/32 = 0.15625
0 0 1 1 0	6/32 = 0.1875
0 0 1 1 1	7/32 = 0.21875
0 1 0 0 0	8/32 = 0.25
0 1 0 0 1	9/32 = 0.28125
0 1 0 1 0	10/32 = 0.3125
0 1 0 1 1	11/32 = 0.34375
0 1 1 0 0	12/32 = 0.375
0 1 1 0 1	13/32 = 0.40625
0 1 1 1 0	14/32 = 0.4375
0 1 1 1 1	15/32 = 0.46875
1 0 0 0 0	16/32 = 0.5
1 0 0 0 1	17/32 = 0.53125
1 0 0 1 0	18/32 = 0.5625
1 0 0 1 1	19/32 = 0.59375
1 0 1 0 0	20/32 = 0.625
1 0 1 0 1	21/32 = 0.65625
1 0 1 1 0	22/32 = 0.6875
1 0 1 1 1	23/32 = 0.71875
1 1 0 0 0	24/32 = 0.75
1 1 0 0 1	25/32 = 0.78125
1 1 0 1 0	26/32 = 0.8125
1 1 0 1 1	27/32 = 0.84375
1 1 1 0 0	28/32 = 0.875
1 1 1 0 1	29/32 = 0.90625
1 1 1 1 0	30/32 = 0.9375
1 1 1 1 1	31/32 = 0.96875

ARUN— Arbiter Counter Running Flag

This read-only bit indicates the arbiter counter is running. Reset clears ARUN.

- 1 = Arbiter counter running
- 0 = Arbiter counter stopped

AROVFL— Arbiter Counter Overflow Bit

This read-only bit indicates an arbiter counter overflow. Clear AROVFL by writing any value to SCIACTL. Writing logic 0s to AM1 and AM0 resets the counter keeps it in this idle state. Reset clears AROVFL.

- 1 = Arbiter counter overflow has occurred
- 0 = No arbiter counter overflow has occurred

ARD8— Arbiter Counter MSB

This read-only bit is the MSB of the 9-bit arbiter counter. Clear ARD8 by writing any value to SCIACTL. Reset clears ARD8.

15.9.2 ESCI Arbiter Data Register

Address: \$000B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 15-20. ESCI Arbiter Data Register (SCIADAT)

ARD7–ARD0 — Arbiter Least Significant Counter Bits

These read-only bits are the eight LSBs of the 9-bit arbiter counter. Clear ARD7–ARD0 by writing any value to SCIACTL. Writing logic 0s to AM1 and AM0 permanently resets the counter and keeps it in this idle state. Reset clears ARD7–ARD0.

15.9.3 Bit Time Measurement

Two bit time measurement modes, described here, are available according to the state of ACLK.

1. **ACLK = 0** — The counter is clocked with the bus clock divided by four. The counter is started when a falling edge on the RxD pin is detected. The counter will be stopped on the next falling edge. ARUN is set while the counter is running, AFIN is set on the second falling edge on RxD (for instance, the counter is stopped). This mode is used to recover the received baud rate. See Figure 15-21.
2. **ACLK = 1** — The counter is clocked with one quarter of the ESCI input clock generated by the ESCI prescaler. The counter is started when a logic 0 is detected on RxD (see Figure 15-22). A logic 0 on RxD on enabling the bit time measurement with ACLK = 1 leads to immediate start of the counter (see Figure 15-23). The counter will be stopped on the next rising edge of RxD. This mode is used to measure the length of a received break.

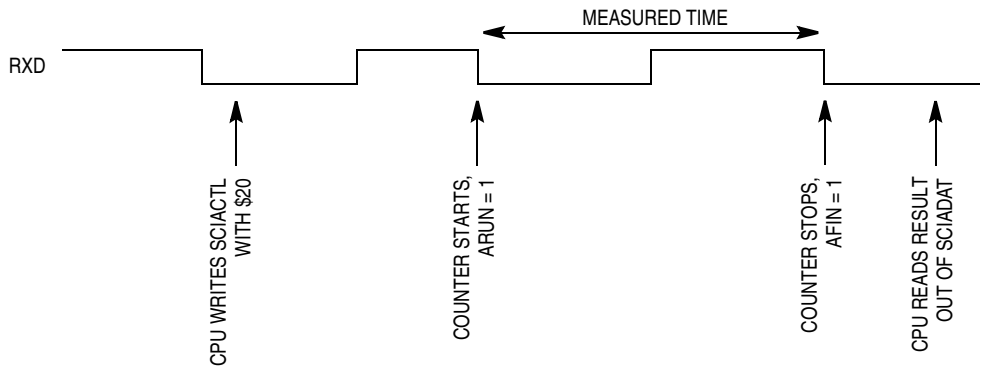


Figure 15-21. Bit Time Measurement with ACLK = 0

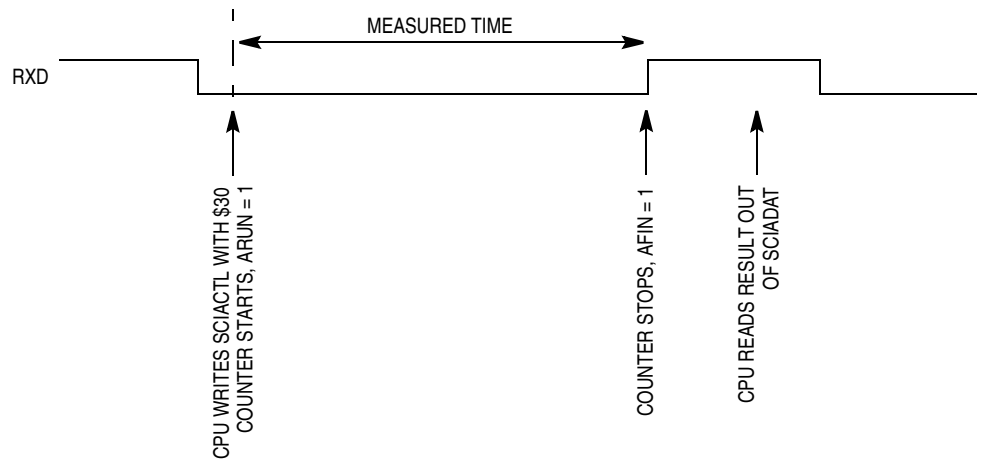


Figure 15-22. Bit Time Measurement with ACLK = 1, Scenario A

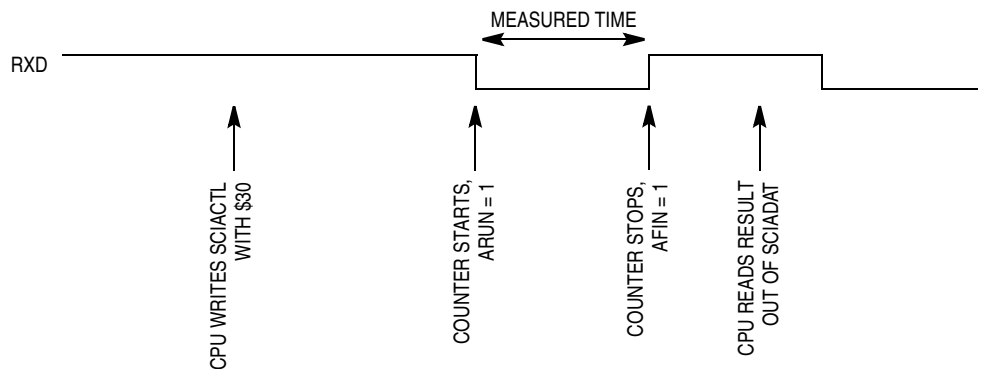


Figure 15-23. Bit Time Measurement with ACLK = 1, Scenario B

ILAD — Illegal Address Reset Bit (opcode fetches only)

1 = Last reset caused by an opcode fetch from an illegal address
0 = POR or read of SRSR

MODRST — Monitor Mode Entry Module Reset Bit

1 = Last reset caused by monitor mode entry when vector locations \$FFFE and \$FFFF are \$FF after POR while $\overline{\text{IRQ}} = V_{\text{DD}}$
0 = POR or read of SRSR

LVI — Low-Voltage Inhibit Reset Bit

1 = Last reset caused by the LVI circuit
0 = POR or read of SRSR

16.7.3 Break Flag Control Register

The break flag control register (BFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

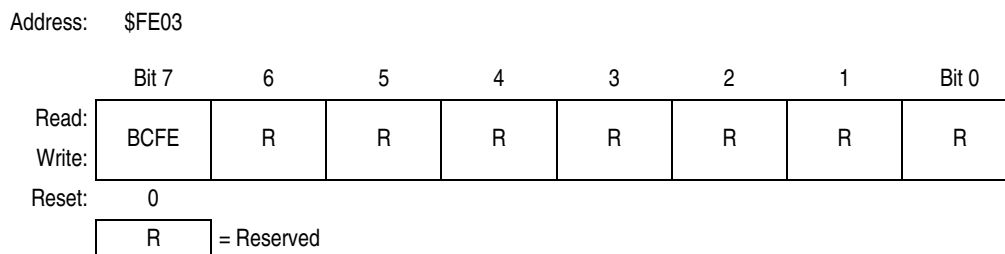


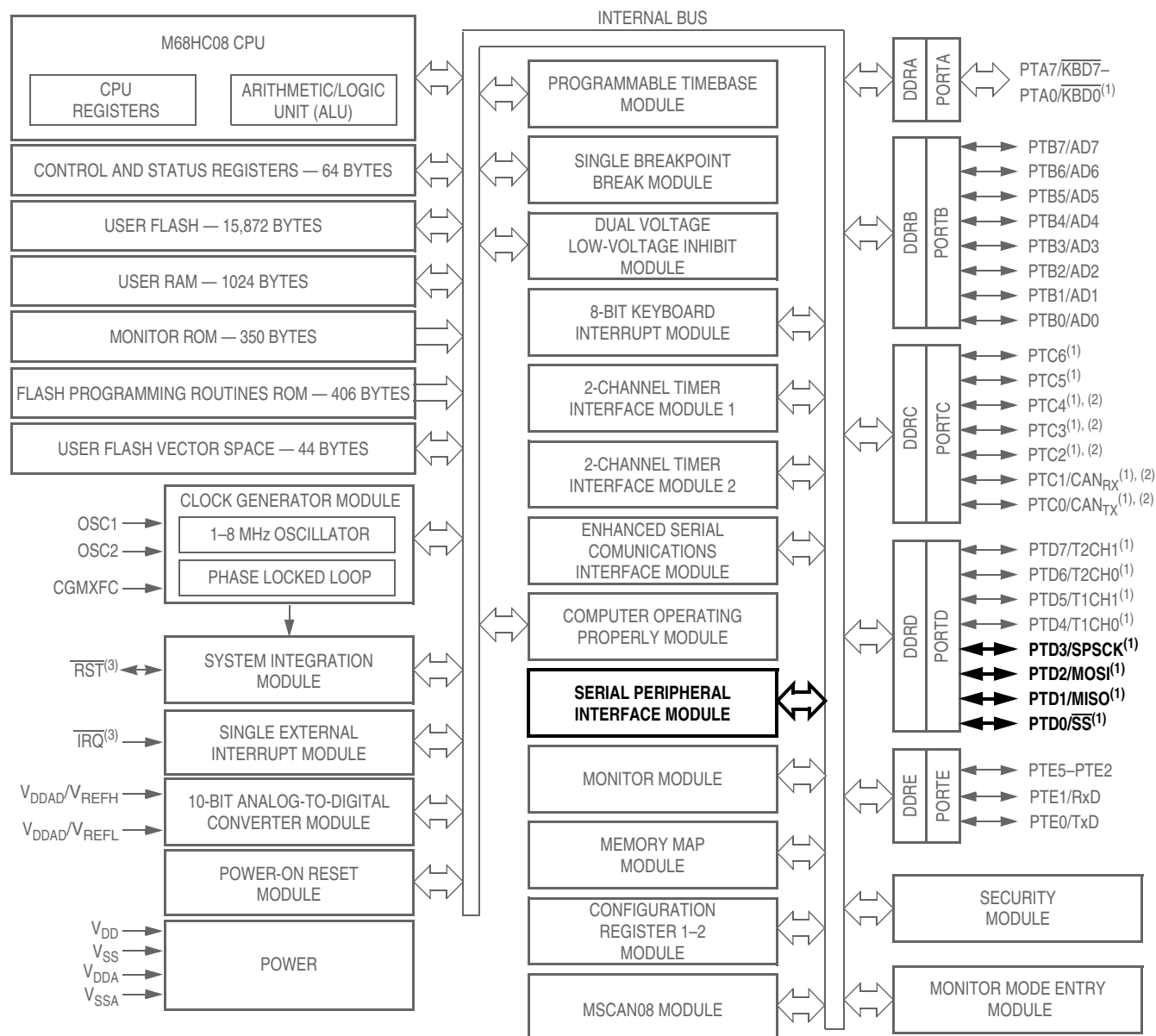
Figure 16-22. Break Flag Control Register (BFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break
0 = Status bits not clearable during break

Serial Peripheral Interface (SPI) Module



1. Ports are software configurable with pullup device if input port.
2. Higher current drive port pins
3. Pin contains integrated pullup device

Figure 17-1. Block Diagram Highlighting SPI Block and Pins

Enter monitor mode with pin configuration shown in Table 20-1 by pulling $\overline{\text{RST}}$ low and then high. The rising edge of $\overline{\text{RST}}$ latches monitor mode. Once monitor mode is latched, the values on the specified pins can change.

Once out of reset, the MCU waits for the host to send eight security bytes (see 20.3.2 Security). After the security bytes, the MCU sends a break signal (10 consecutive logic 0s) to the host, indicating that it is ready to receive a command.

20.3.1.1 Normal Monitor Mode

If V_{TST} is applied to $\overline{\text{IRQ}}$ and PTB4 is low upon monitor mode entry, the bus frequency is a divide-by-two of the input clock. If PTB4 is high with V_{TST} applied to $\overline{\text{IRQ}}$ upon monitor mode entry, the bus frequency will be a divide-by-four of the input clock. Holding the PTB4 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 CGMOUT frequency is equal to the CGMXCLK frequency, and the OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50% duty cycle at maximum bus frequency.

When monitor mode was entered with V_{TST} on $\overline{\text{IRQ}}$, the computer operating properly (COP) is disabled as long as V_{TST} is applied to either $\overline{\text{IRQ}}$ or $\overline{\text{RST}}$.

This condition states that as long as V_{TST} is maintained on the $\overline{\text{IRQ}}$ pin after entering monitor mode, or if V_{TST} is applied to $\overline{\text{RST}}$ after the initial reset to get into monitor mode (when V_{TST} was applied to $\overline{\text{IRQ}}$), then the COP will be disabled. In the latter situation, after V_{TST} is applied to the $\overline{\text{RST}}$ pin, V_{TST} can be removed from the $\overline{\text{IRQ}}$ pin in the interest of freeing the $\overline{\text{IRQ}}$ for normal functionality in monitor mode.

20.3.1.2 Forced Monitor Mode

If entering monitor mode without high voltage on $\overline{\text{IRQ}}$, then all port B pin requirements and conditions, including the PTB4 frequency divisor selection, are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

NOTE

If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial power-on reset (POR). Once the reset vector has been programmed, the traditional method of applying a voltage, V_{TST} , to $\overline{\text{IRQ}}$ must be used to enter monitor mode.

An external oscillator of 8 MHz is required for a baud rate of 7200, as the internal bus frequency is automatically set to the external frequency divided by four.

When the forced monitor mode is entered the COP is always disabled regardless of the state of $\overline{\text{IRQ}}$ or $\overline{\text{RST}}$.

20.3.1.3 Monitor Vectors

In monitor mode, the MCU uses different vectors for reset, SWI (software interrupt), and break interrupt than those for user mode. The alternate vectors are in the \$FE page instead of the \$FF page and allow code execution from the internal monitor firmware instead of user code.

Table 20-2 summarizes the differences between user mode and monitor mode.

Table 20-7. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high-byte:low-byte order
Opcode	\$0C
<p style="text-align: center;">Command Sequence</p>	

Table 20-8. RUN (Run User Program) Command

Description	Executes PULH and RTI instructions
Operand	None
Data Returned	None
Opcode	\$28
<p style="text-align: center;">Command Sequence</p>	

The MCU executes the SWI and PSHH instructions when it enters monitor mode. The RUN command tells the MCU to execute the PULH and RTI instructions. Before sending the RUN command, the host can modify the stacked CPU registers to prepare to run the host program. The READSP command returns the incremented stack pointer value, SP + 1. The high and low bytes of the program counter are at addresses SP + 5 and SP + 6.

	SP
HIGH BYTE OF INDEX REGISTER	SP + 1
CONDITION CODE REGISTER	SP + 2
ACCUMULATOR	SP + 3
LOW BYTE OF INDEX REGISTER	SP + 4
HIGH BYTE OF PROGRAM COUNTER	SP + 5
LOW BYTE OF PROGRAM COUNTER	SP + 6
	SP + 7

Figure 20-15. Stack Pointer at Monitor Mode Entry

21.14 Timer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Timer input capture pulse width	t_{TH}, t_{TL}	2	—	t_{CYC}
Timer Input capture period	t_{TLTL}	Note ⁽¹⁾	—	t_{CYC}

1. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1 t_{CYC} .

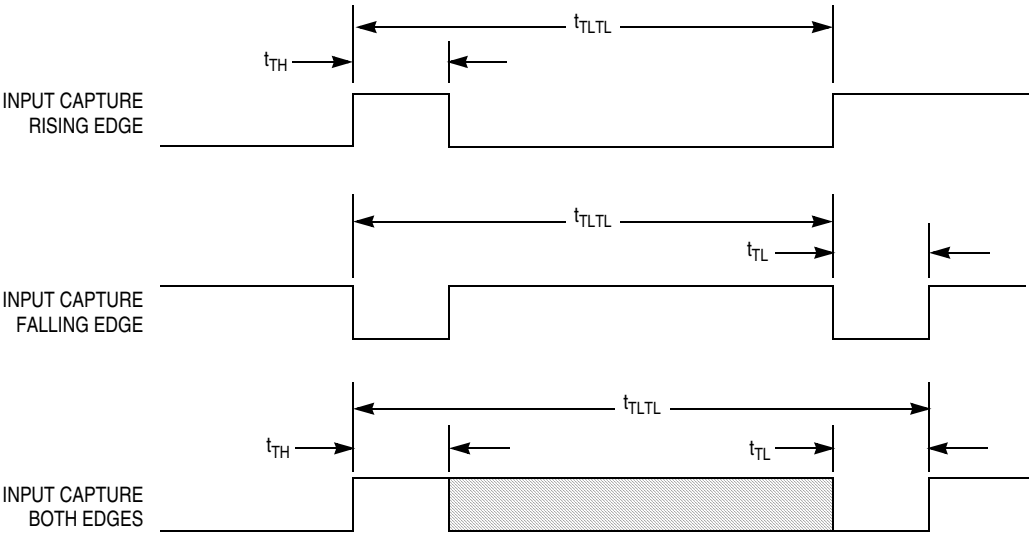


Figure 21-4. Input Capture Timing