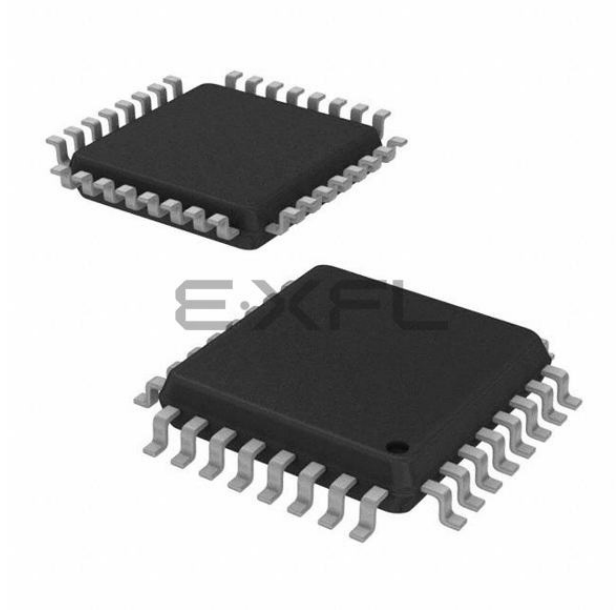


Welcome to E-XFL.COM

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	LINbus, SCI, SPI
Peripherals	POR, PWM
Number of I/O	24
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908ey16cfaer

Chapter 19
Development Support

Chapter 2
Memory

2.1 Introduction

The M68HC08 central processor unit (CPU08) can address 64 Kbytes of memory space. The memory map, shown in Figure 2-1, includes:

- 16 Kbytes of FLASH memory, 15, 872 bytes of user space
- 512 bytes of random-access memory (RAM)
- 36 bytes of user-defined vectors
- 310 bytes of monitor routines in read-only memory (ROM)
- 1024 bytes of integrated FLASH burn-in routines in ROM

2.2 Unimplemented Memory Locations

Accessing an unimplemented location can cause an illegal address reset. In the memory map (Figure 2-1) and in register figures in this document, unimplemented locations are shaded.

2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on microcontroller unit (MCU) operation. In the Figure 2-1 and in register figures in this document, reserved locations are marked with the word reserved or with the letter R.

2.4 Input/Output (I/O) Section

Most of the control, status, and data registers are in the zero page area of $0000–$003F. Additional I/O registers have these addresses:

- $FE00; SIM break status register, SBSR
- $FE01; SIM reset status register, SRSR
- $FE03; SIM break flag control register, SBFCR
- $FE08; FLASH control register, FLCR
- $FE09; break address register high, BRKH
- $FE0A; break address register low, BRKL
- $FE0B; break status and control register, BRKSCR
- $FE0C; LVI status register, LVISR
- $FF7E; FLASH block protect register, FLBPR
- $FF80; ICG trim value (optional), ICGT

Data registers are shown in Figure 2-2. and Table 2-1 is a list of vector locations.

Chapter 4
BEMF Counter Module (BEMF)

4.1 Introduction

This section describes the BEMF module. The BEMF counter integrates over time, while the PTD0/TACH0 pin is active. This function is useful for measuring recirculation currents in motors occurring on switching of inductive loads.

BEMF is the abbreviation for **B**ack **E**lectro**M**agnetic **F**orce.

4.2 Functional Description

The 8-bit BEMF counter runs at the internal bus frequency divided by 64. Whenever PTD0/TACH0 is a logic 1, the counter increments by 1 with each period.

4.3 BEMF Register

The BEMF register contains the eight read-only bits of the BEMF counter, showing its actual value. A read access to the BEMF register resets all counter bits to 0.

Address: $000B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BEMF7	BEMF6	BEMF5	BEMF4	BEMF3	BEMF2	BEMF1	BEMF0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 4-1. BEMF Register (BEMF)

4.4 Input Signal

Port D shares the PTD0/TACH0 pin with the BEMF module. To measure an external signal with the BEMF module, PTD0/TACH0 must be configured as an input (DDRD0 = 0).

4.5 Low Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

4.5.1 Wait Mode

The BEMF module remains active after execution of the WAIT instruction. In WAIT mode the BEMF register is not accessible by the CPU.

EXTSLOW — Slow External Crystal Enable Bit

The EXTSLOW bit has two functions. It configures the ICG module for a fast (1 MHz to 8 MHz) or slow (30 kHz to 100 kHz) speed crystal. The option also configures the clock monitor operation in the ICG module to expect an external frequency higher (307.2 kHz to 32 MHz) or lower (60 Hz to 307.2 kHz) than the base frequency of the internal oscillator. See Chapter 8 Internal Clock Generator (ICG) Module.

1 = ICG set for slow external crystal operation
0 = ICG set for fast external crystal operation

EXTCLKEN — External Clock Enable Bit

EXTCLKEN enables an external clock source or crystal/ceramic resonator to be used as a clock input. Setting this bit enables PTC4/OSC1 pin to be a clock input pin. Clearing this bit (default setting) allows the PTC4/OSC1 and PTC3/OSC2 pins to function as general-purpose input/output (I/O) pins. Refer to Table 5-1 for configuration options for the external source. See Chapter 8 Internal Clock Generator (ICG) Module for a more detailed description of the external clock operation.

1 = Allows PTC4/OSC1 to be an external clock connection
0 = PTC4/OSC1 and PTC3/OSC2 function as I/O port pins (default).

TMBCLKSEL — Timebase Clock Select Bit

TMBCLKSEL enables an enable the extra divide by 128 prescaler in the timebase module. Setting this bit enables the extra prescaler and clearing this bit disables it. Refer to Table 16-1 for timebase divider selection details.

1 = Enables extra divide by 128 prescaler in timebase module.
0 = Disables extra divide by 128 prescaler in timebase module.

OSCENINSTOP — Oscillator Enable In Stop Mode Bit

OSCENINSTOP, when set, will enable the internal clock generator module to continue to generate clocks (either internal, ICLK, or external, ECLK) in stop mode. See Chapter 8 Internal Clock Generator (ICG) Module. This function is used to keep the timebase running while the rest of the microcontroller stops. When clear, all clock generation will cease and both ICLK and ECLK will be forced low during stop mode. The default state for this option is clear, disabling the ICG in stop mode.

1 = Oscillator enabled to operate during stop mode
0 = Oscillator disabled during stop mode (default)

NOTE

This bit has the same functionality as the OSCSTOPENB CONFIG bit in MC68HC908GP20 and MC68HC908GR8 parts.

SSBPUENB — $\overline{SS}$ Pullup Enable Bit

Clearing SSBPUENB enables the $\overline{SS}$ pullup resistor.

1 = Disables $\overline{SS}$ pullup resistor.
0 = Enables $\overline{SS}$ pullup resistor.

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS. See Chapter 6 Computer Operating Properly (COP) Module.

1 = COP timeout period = 8176 CGMXCLK cycles
0 = COP timeout period = 262,128 CGMXCLK cycles

LVISTOP — LVI Enable in Stop Mode Bit

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP.

1 = LVI enabled during stop mode
0 = LVI disabled during stop mode

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	V	1	1	H	I	N	Z	C
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

Figure 7-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE

To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

8.3.3.2 External Clock Input Path

The external clock input path is the means by which the microcontroller uses an external clock source. The input to the path is the PTC4/OSC1 pin and the output is the external clock (ECLK). The path, which contains input buffering, is enabled when the external clock generator enable signal (ECGEN) is set. When not enabled, the PTC4/OSC1 pin reverts to its port function.

8.3.4 Clock Monitor Circuit

The ICG contains a clock monitor circuit which, when enabled, will continuously monitor both the external clock (ECLK) and the internal clock (ICLK) to determine if either clock source has been corrupted. The clock monitor circuit, shown in Figure 8-5, contains these blocks:

- Clock monitor reference generator
- Internal clock activity detector
- External clock activity detector

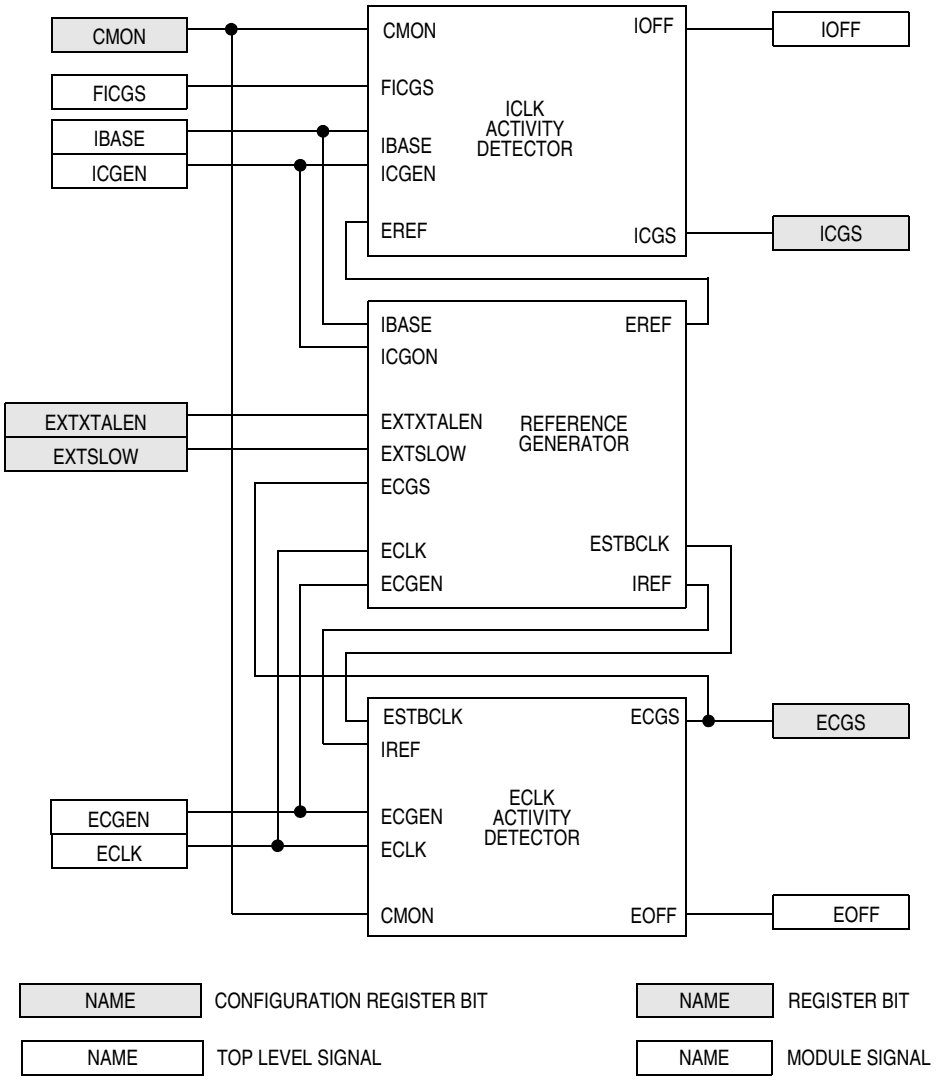


Figure 8-5. Clock Monitor Block Diagram

Chapter 12
Input/Output (I/O) Ports (PORTS)

12.1 Introduction

Twenty-four bidirectional input/output (I/O) pins form five parallel ports. All I/O pins are programmable as inputs or outputs.

NOTE

Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS}. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.

12.2 Port A

Port A is a 7-bit general-purpose bidirectional I/O port that shares pin functions with the serial peripheral interface (SPI) and keyboard (KBD) modules.

12.2.1 Port A Data Register

The port A data register contains a data latch for each of the seven port A pins.

Address:	$0000							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Write:								
Reset:	Unaffected by reset							
Alternative Function:		$\overline{SS}$	SPSCK	$\overline{KBD4}$	$\overline{KBD3}$	$\overline{KBD2}$	$\overline{KBD1}$	$\overline{KBD0}$

= Unimplemented

Figure 12-1. Port A Data Register (PTA)

PTA[6:0] — Port A Data Bits

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

12.2.2 Data Direction Register A

Data direction register A determines whether each port A pin is an input or an output. Writing a 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a 0 disables the output buffer.

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 consecutive 1s from the transmit shift register to the TxD pin. If software clears the TE bit, the transmitter completes any transmission in progress before the TxD returns to the idle condition (1). Clearing and then setting TE during a transmission queues an idle character to be sent after the character currently being transmitted. Reset clears the TE bit.

1 = Transmitter enabled
0 = Transmitter disabled

NOTE

Writing to the TE bit is not allowed when the enable ESCI bit (ENSCI) is clear. ENSCI is in ESCI control register 1.

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

1 = Receiver enabled
0 = Receiver disabled

NOTE

Writing to the RE bit is not allowed when the enable ESCI bit (ENSCI) is clear. ENSCI is in ESCI control register 1.

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a 1. The 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no 1s between them. Reset clears the SBK bit.

1 = Transmit break characters
0 = No break characters being transmitted

NOTE

Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the ESCI to send a break character instead of a preamble.

13.9 ESCI Arbiter

The ESCI module comprises an arbiter module designed to support software for communication tasks as bus arbitration, baud rate recovery and break time detection. The arbiter module consists of an 9-bit counter with 1-bit overflow and control logic. The CPU can control operation mode via the ESCI arbiter control register (SCIACTL).

13.9.1 ESCI Arbiter Control Register

Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AM1	ALOST	AM0	ACLK	AFIN	ARUN	AROVFL	ARD8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-18. ESCI Arbiter Control Register (SCIACTL)

AM1 and AM0 — Arbiter Mode Select Bits

As shown in Table 13-12, these read/write bits select the mode of the arbiter module. Reset clears AM1 and AM0.

Table 13-12. ESCI Arbiter Selectable Modes

AM[1:0]	ESCI Arbiter Mode
0 0	Idle / counter reset
0 1	Bit time measurement
1 0	Bus arbitration
1 1	Reserved / do not use

ALOST — Arbitration Lost Flag

This read-only bit indicates loss of arbitration. Clear ALOST by writing a 0 to AM1. Reset clears ALOST.

ACLK — Arbiter Counter Clock Select Bit

This read/write bit selects the arbiter counter clock source. Reset clears ACLK.

1 = Arbiter counter is clocked with one quarter of the ESCI input clock generated by the ESCI prescaler.

0 = Arbiter counter is clocked with the bus clock divided by four

NOTE

For ACLK=1, the Arbiter input clock is driven from the ESCI prescaler. The prescaler can be clocked by either the bus clock or CGMXCLK depending on the state of the ESCIBDSRC bit in CONFIG2.

AFIN— Arbiter Bit Time Measurement Finish Flag

This read-only bit indicates bit time measurement has finished. Clear AFIN by writing any value to SCIACTL. Reset clears AFIN.

1 = Bit time measurement has finished

0 = Bit time measurement not yet finished

Chapter 14
System Integration Module (SIM)

14.1 Introduction

This section describes the system integration module (SIM), which supports up to 24 external and/or internal interrupts. The SIM is a system state controller that coordinates the central processor unit (CPU) and exception timing. Together with the CPU, the SIM controls all microcontroller unit (MCU) activities. A block diagram of the SIM is shown in Figure 14-1.

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals:
  - Stop/wait/reset entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing
- Modular architecture expandable to 128 interrupt sources

Table 14-1 shows the internal signal names used in this section.

Table 14-1. Signal Name Conventions

Signal Name	Description
CGMXCLK	Selected clock source from internal clock generator module (ICG)
CGMOUT	Clock output from ICG module (bus clock = CGMOUT divided by two)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset (POR) module to the SIM
IRST	Internal reset signal
R/$\overline{W}$	Read/write signal

14.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in Figure 14-2. This clock originates from either an external oscillator or from the internal clock generator.

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. Figure 14-10 demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the load-accumulator- from-memory (LDA) instruction is executed.

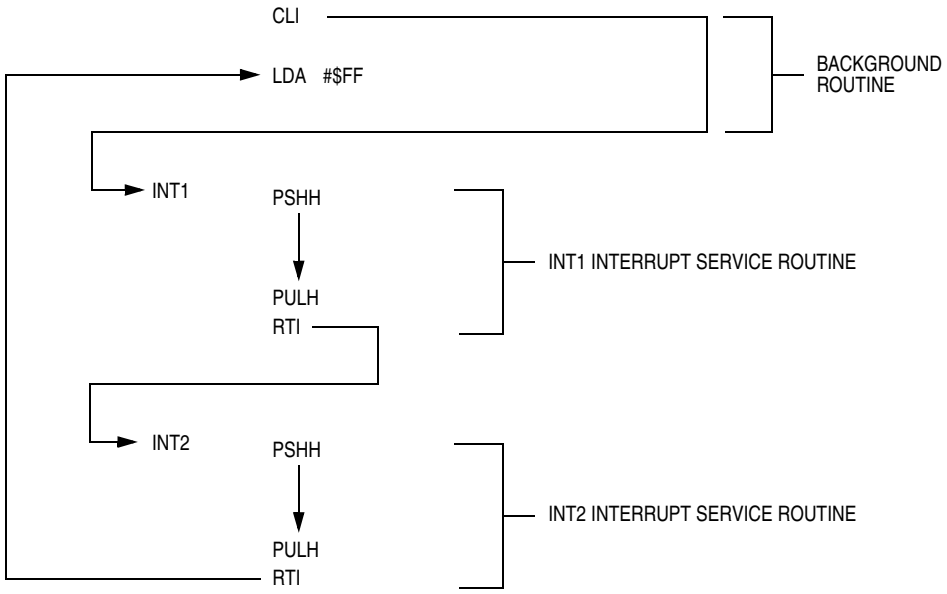


Figure 14-10. Interrupt Recognition Example

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, M146805, and MC68HC05 Families 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, software should save the H register and then restore it prior to exiting the routine.

14.5.1.2 SWI Instruction

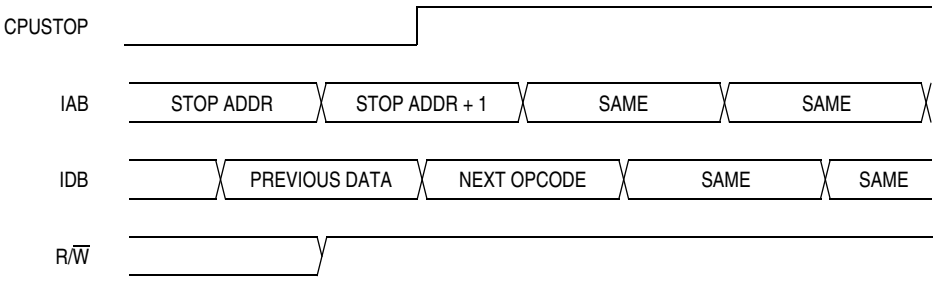
The SWI instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask (I bit) in the condition code register.

NOTE

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

14.5.2 Reset

All reset sources always have higher priority than interrupts and cannot be arbitrated.



Note: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 14-14. Stop Mode Entry Timing

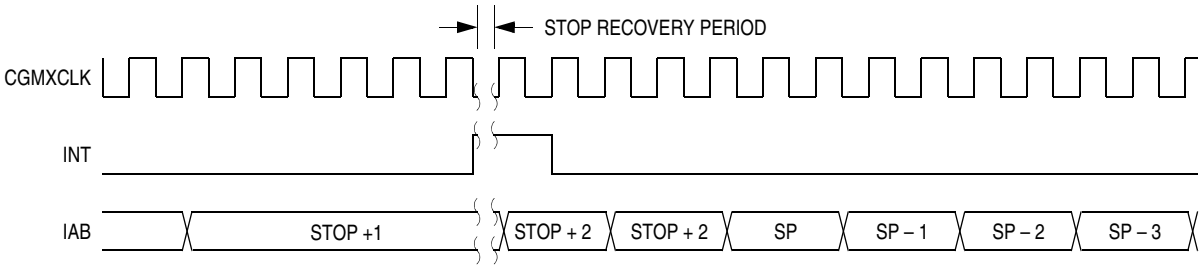


Figure 14-15. Stop Mode Recovery from Interrupt

14.7 SIM Registers

The SIM has three memory mapped registers. Table 14-3 shows the mapping of these registers.

Table 14-3. SIM Registers

Address	Register	Access Mode
$FE00	SBSR	User
$FE01	SRSR	User
$FE03	SBFCR	User

14.7.1 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from stop or wait mode.

Address: $FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note(1)	
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Note: 1. Writing a 0 clears SBSW

Figure 14-16. SIM Break Status Register (SBSR)

15.5.2 Transmission Format When CPHA = 0

Figure 15-4 shows an SPI transmission in which CPHA (SPCR) is 0. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ($\overline{SS}$) is at logic 0, so that only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI (see 15.6.2 Mode Fault Error). When CPHA = 0, the first SPSCK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SPSCK edge, and a falling edge on the $\overline{SS}$ pin is used to start the transmission. The $\overline{SS}$ pin must be toggled high and then low again between each byte transmitted.

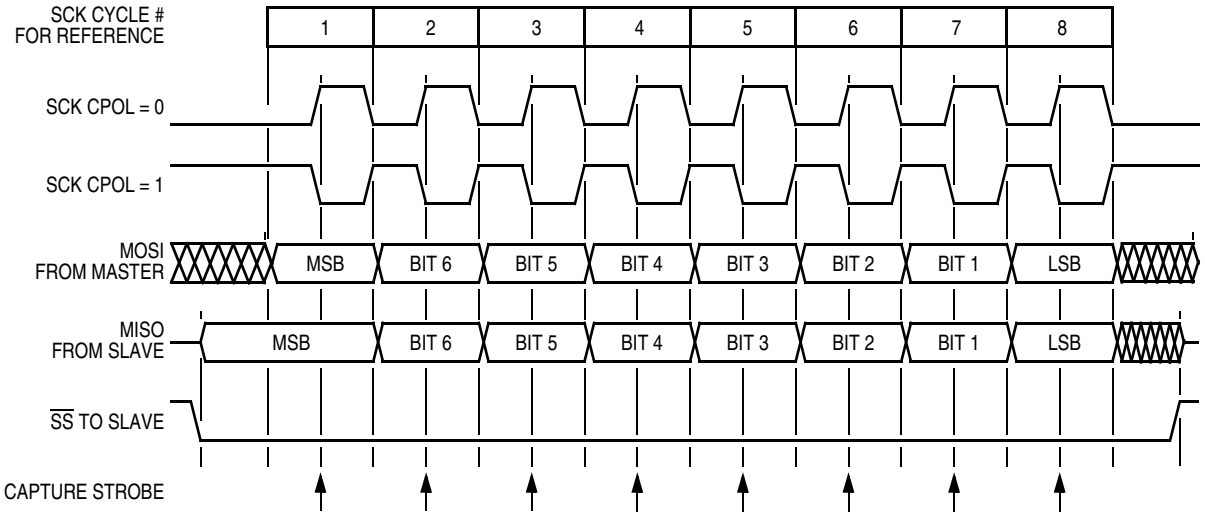


Figure 15-4. Transmission Format (CPHA = 0)

15.5.3 Transmission Format When CPHA = 1

Figure 15-5 shows an SPI transmission in which CPHA (SPCR) is 1. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ($\overline{SS}$) is at logic 0, so that only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 15.6.2 Mode Fault Error.) When CPHA = 1, the master begins driving its MOSI pin on the first SPSCK edge. Therefore, the slave uses the first SPSCK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions. This format may be preferable in systems having only one master and only one slave driving the MISO data line.

16.5 TBM Interrupt Rate

The interrupt rate is determined by the equation:

$$t_{TBMRATE} = \frac{1}{f_{TBMRATE}} = \frac{Divider}{f_{TBMCLK}}$$

where:

f_{TBMCLK} = Frequency supplied from the internal clock generator (ICG) module

Divider = Divider value as determined by TBR2–TBR0 settings. See Table 16-1

As an example, a clock source of 4.9152 MHz and the TBR2–TBR0 set to {011}, the divider tap is 128 and the interrupt rate calculates to 128/4.9152 x 10^6 = 26 µs.

Table 16-1. Timebase Divider Selection

TBR2[1]	TBR1[1]	TBR0[1]	Divider Tap TMBCLKSEL	
			0	1
0	0	0	32,768	4,194,304
0	0	1	8192	1,048,576
0	1	0	2048	262144
0	1	1	128	16,384
1	0	0	64	8192
1	0	1	32	4096
1	1	0	16	2048
1	1	1	8	1024

1. Do not change TBR2–TBR0 bits while the timebase is enabled (TBON = 1).

16.6 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

16.6.1 Wait Mode

The timebase module remains active after execution of the WAIT instruction. In wait mode the timebase register is not accessible by the CPU.

If the timebase functions are not required during wait mode, reduce the power consumption by stopping the timebase before executing the WAIT instruction.

16.6.2 Stop Mode

The timebase module may remain active after execution of the STOP instruction if the internal clock generator has been enabled to operate during stop mode through the OSCENINSTOP bit in the configuration register. The timebase module can be used in this mode to generate a periodic wake up from stop mode.

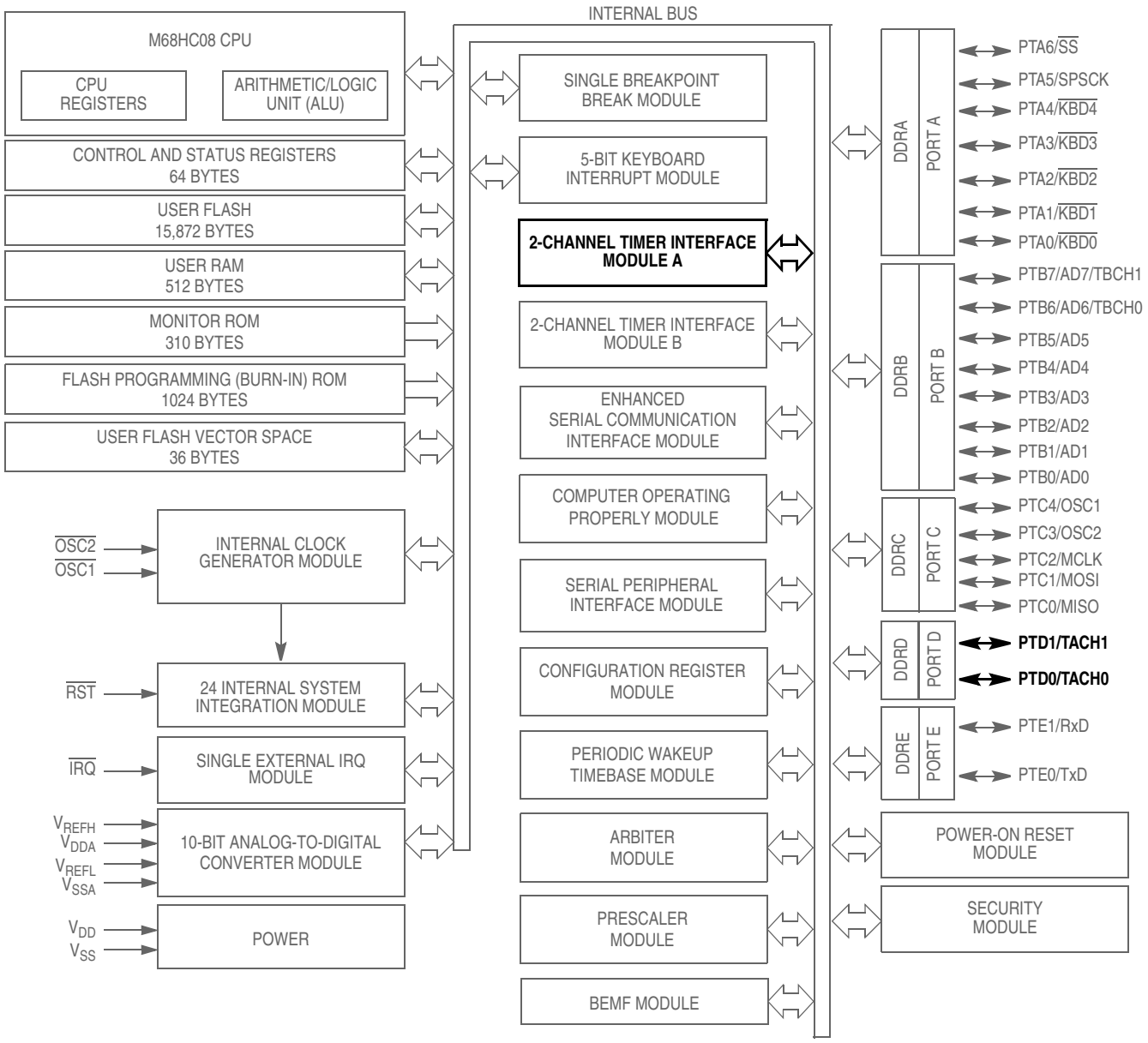


Figure 17-1. Block Diagram Highlighting TIMA Block and Pins

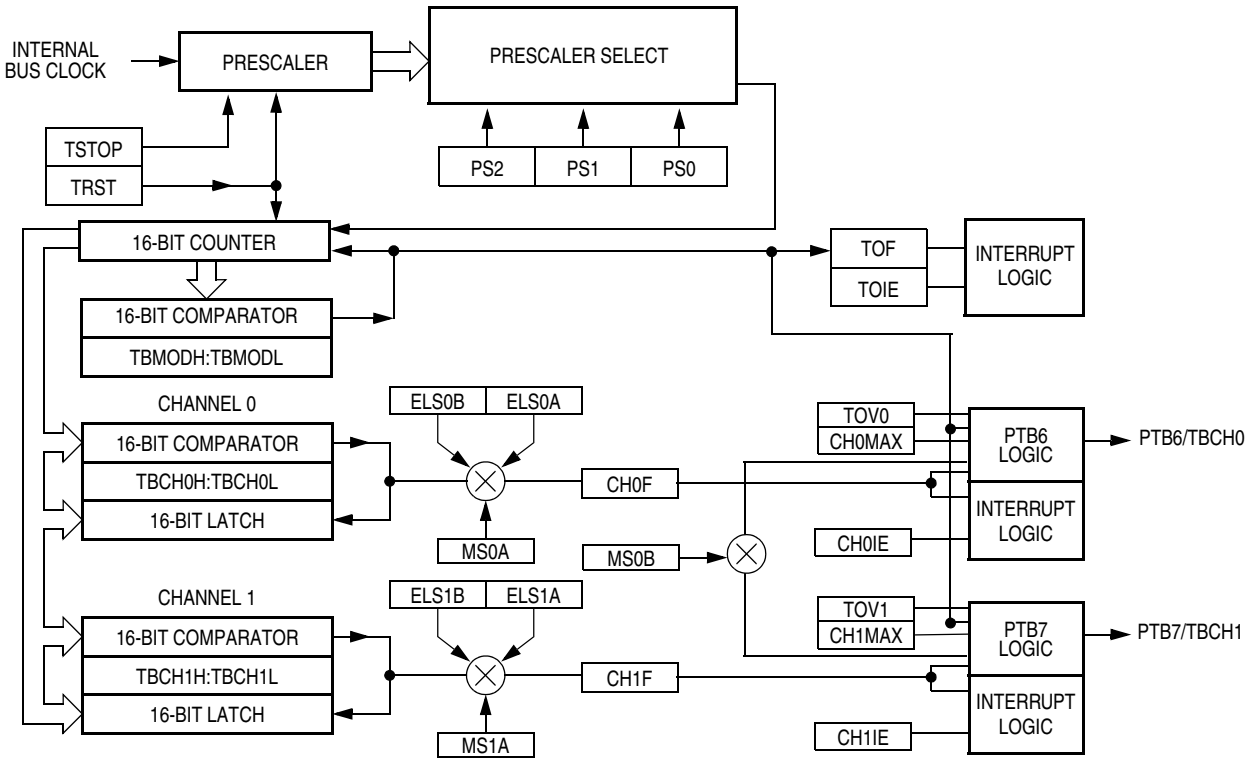


Figure 18-2. TIMB Block Diagram

18.3.1 TIMB Counter Prescaler

The TIMB clock source can be 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 TIMB status and control register select the TIMB clock source.

18.3.2 Input Capture

An input capture function has three basic parts: edge select logic, an input capture latch, and a 16-bit counter. Two 8-bit registers, which make up the 16-bit input capture register, are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The polarity of the active edge is programmable. The level transition which triggers the counter transfer is defined by the corresponding input edge bits (ELSxB and ELSxA in TBSC0 through TBSC1 control registers with x referring to the active channel number). When an active edge occurs on the pin of an input capture channel, the TIMB latches the contents of the TIMB counter into the TIMB channel registers, TBCHxH–TBCHxL. Input captures can generate TIMB CPU interrupt requests. Software can determine that an input capture event has occurred by enabling input capture interrupts or by polling the status flag bit.

The free-running counter contents are transferred to the TIMB channel status and control register (TBCHxH–TBCHxL, see 18.8.5 TIMB Channel Registers) on each proper signal transition regardless of whether the TIMB channel flag (CH0F–CH1F in TBSC0–TBSC1 registers) is set or clear. When the status flag is set, a CPU interrupt is generated if enabled. The value of the count latched or “captured” is the time of the event. Because this value is stored in the input capture register 2 bus cycles after the actual event

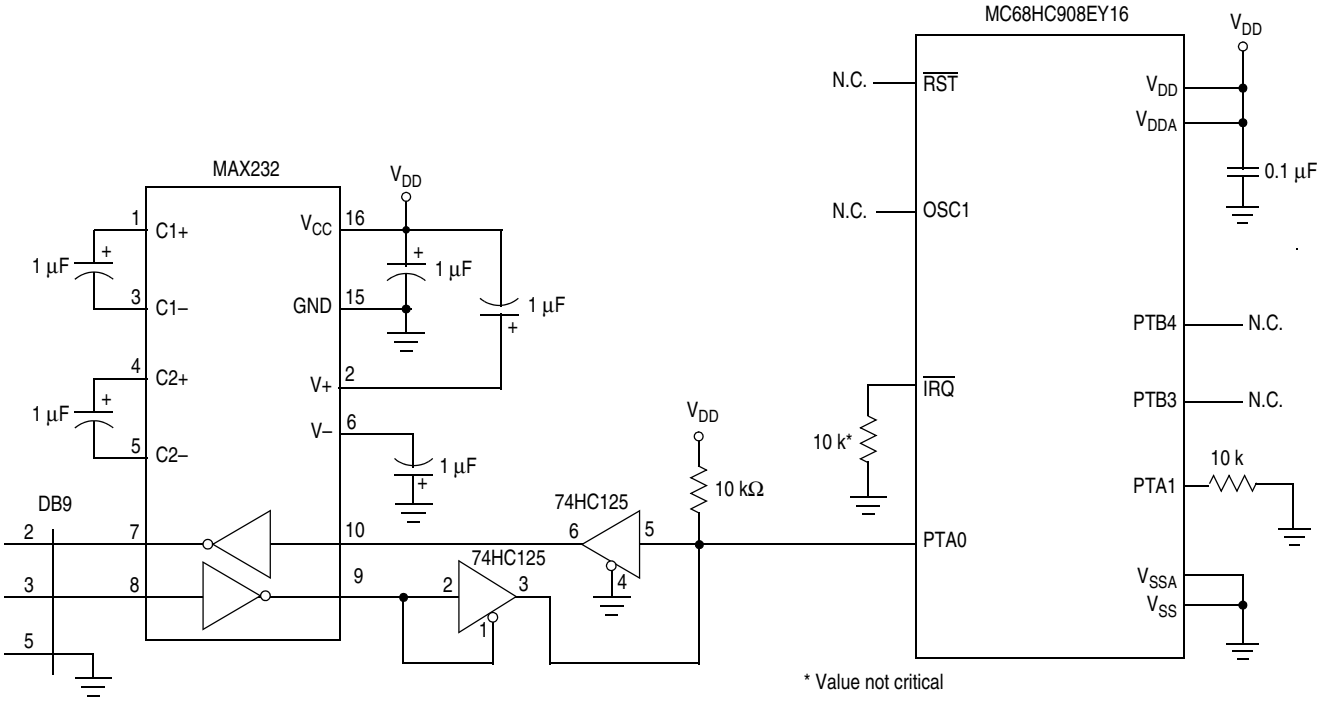


Figure 19-11. Forced Monitor Mode ($\overline{IRQ}$ = V_{SS})

Enter monitor mode with pin configuration shown in Table 19-1 by pulling $\overline{RST}$ low and then high. The rising edge of $\overline{RST}$ latches monitor mode. Once monitor mode is latched, the levels on the port pins except PTA0 can change.

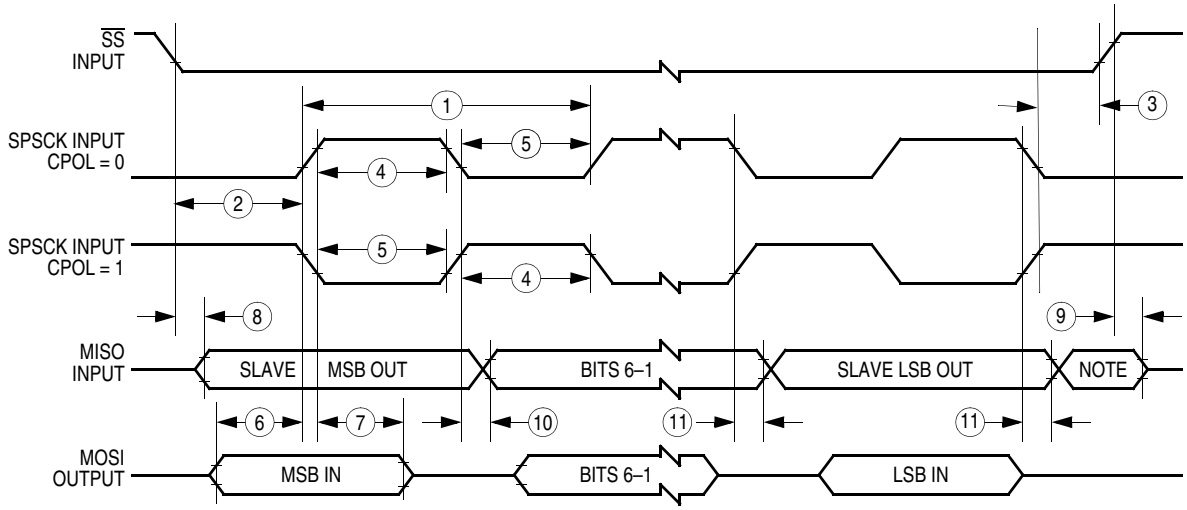
Once out of reset, the MCU waits for the host to send eight security bytes (see 19.3.2 Security). After the security bytes, the MCU sends a break signal (10 consecutive 0s) to the host, indicating that it is ready to receive a command.

19.3.1.1 Normal Monitor Mode

If V_{TST} is applied to $\overline{IRQ}$ upon monitor mode entry, the bus frequency is a divide-by-four of the input clock.

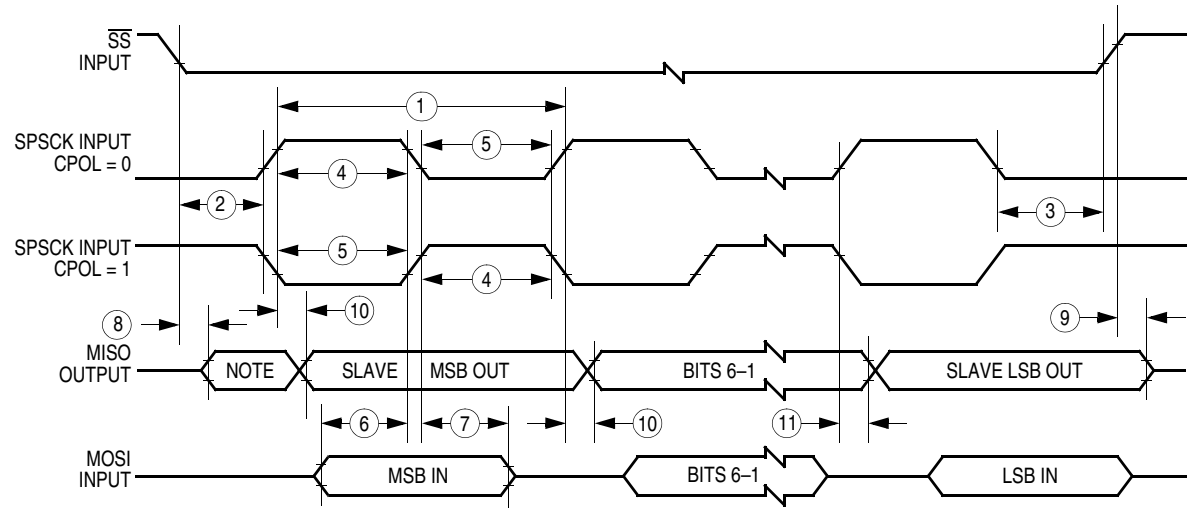
When monitor mode was entered with V_{TST} on $\overline{IRQ}$, the computer operating properly (COP) is disabled as long as V_{TST} is applied to either $\overline{IRQ}$ or $\overline{RST}$.

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



Note: Not defined but normally MSB of character just received

a) SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted

b) SPI Slave Timing (CPHA = 1)

Figure 20-2. SPI Slave Timing

SPI — See "serial peripheral interface module (SPI)."

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU08 containing the address of the next available storage location on the stack.

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

TIM — See "timer interface module (TIM)."

timer interface module (TIM) — A module used to relate events in a system to a point in time.

timer — A module used to relate events in a system to a point in time.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — Mode of low-jitter PLL operation during which the PLL is locked on a frequency. Also see "acquisition mode."

two’s complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two’s complement number indicates the sign of the number (1 indicates negative). The two’s complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value. Executing an opcode at an unimplemented location causes an illegal address reset.

V —The overflow bit in the condition code register of the CPU08. The CPU08 sets the V bit when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow bit.

variable — A value that changes during the course of program execution.

VCO — See "voltage-controlled oscillator."