

Welcome to [E-XFL.COM](https://www.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	8KB (8K 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/mc908ey8cfaer

MC68HC908EY16

MC68HC908EY8

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>

Refer to the [Revision History](#) for a summary of changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
This product incorporates SuperFlash® technology licensed from SST.

© Freescale Semiconductor, Inc., 2005. All rights reserved.

Table of Contents

18.3.1	TIMB Counter Prescaler	211
18.3.2	Input Capture	211
18.3.3	Output Compare	212
18.3.3.1	Unbuffered Output Compare	212
18.3.3.2	Buffered Output Compare	213
18.3.4	Pulse Width Modulation (PWM)	213
18.3.4.1	Unbuffered PWM Signal Generation	214
18.3.4.2	Buffered PWM Signal Generation	214
18.3.4.3	PWM Initialization	215
18.4	Interrupts	215
18.5	Low-Power Modes	216
18.5.1	Wait Mode	216
18.5.2	Stop Mode	216
18.6	TIMB During Break Interrupts	216
18.7	I/O Signals	216
18.7.1	TIMB Channel I/O Pins (PTB7/TBCH1–PTB6/TBCH0)	216
18.8	I/O Registers	217
18.8.1	TIMB Status and Control Register	217
18.8.2	TIMB Counter Registers	219
18.8.3	TIMB Counter Modulo Registers	219
18.8.4	TIMB Channel Status and Control Registers	220
18.8.5	TIMB Channel Registers	223

Chapter 19 Development Support

19.1	Introduction	225
19.2	Break Module (BRK)	225
19.2.1	Functional Description	225
19.2.1.1	Flag Protection During Break Interrupts	227
19.2.1.2	TIM During Break Interrupts	227
19.2.1.3	COP During Break Interrupts	227
19.2.2	Break Module Registers	227
19.2.2.1	Break Status and Control Register	228
19.2.2.2	Break Address Registers	228
19.2.2.3	Break Status Register	229
19.2.2.4	Break Flag Control Register	229
19.2.3	Low-Power Modes	229
19.3	Monitor Module (MON)	230
19.3.1	Functional Description	230
19.3.1.1	Normal Monitor Mode	233
19.3.1.2	Forced Monitor Mode	234
19.3.1.3	Monitor Vectors	235
19.3.1.4	Data Format	235
19.3.1.5	Break Signal	235
19.3.1.6	Baud Rate	235
19.3.1.7	Commands	236
19.3.2	Security	239

2.6.2 FLASH Page Erase Operation

Use this step-by-step procedure to erase a page (64 bytes) of FLASH memory to read as logic 1:

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH location within the address range of the block to be erased.
4. Wait for a time, t_{NVS} (minimum 10 μ s).
5. Set the HVEN bit.
6. Wait for a time, t_{Erase} (minimum 1 ms or 4 ms).
7. Clear the ERASE bit.
8. Wait for a time, t_{NVH} (minimum 5 μ s).
9. Clear the HVEN bit.
10. After time, t_{RCV} (typical 1 μ s), the memory can be accessed in read mode again.

NOTE

While these operations must be performed in the order shown, other unrelated operations may occur between the steps.

NOTE

Due to the security feature (see [19.3 Monitor Module \(MON\)](#)) the last page of the FLASH (0xFFDC–0xFFFF), which contains the security bytes, cannot be erased by Page Erase Operation. It can only be erased with the Mass Erase Operation.

In applications that require more than 1000 program/erase cycles, use the 4 ms page erase specification to get improved long-term reliability. Any application can use this 4 ms page erase specification. However, in applications where a FLASH location will be erased and reprogrammed less than 1000 times, and speed is important, use the 1 ms page erase specification to get a shorter cycle time.

External Interrupt (IRQ)

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of these actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (ISCR). Writing a 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears both interrupt latches.

The external interrupt pin is falling-edge triggered and is software-configurable to be both falling-edge and low-level triggered. The MODE bit in the ISCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

When an interrupt pin is edge-triggered only, the interrupt latch remains set until a vector fetch, software clear, or reset occurs.

When an interrupt pin is both falling-edge and low-level-triggered, the interrupt latch remains set until both of these occur:

- Vector fetch or software clear
- Return of the interrupt pin to logic 1

The vector fetch or software clear may occur before or after the interrupt pin returns to logic 1. As long as the pin is low, the interrupt request remains pending. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

When set, the IMASK bit in the ISCR masks all external interrupt requests. A latched interrupt request is not presented to the interrupt priority logic unless the corresponding IMASK bit is clear.

NOTE

The interrupt mask (I) in the condition code register (CCR) masks all interrupt requests, including external interrupt requests. See [Figure 9-2](#).

9.4 $\overline{\text{IRQ}}$ Pin

A logic 0 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 these actions must occur to clear the IRQ latch:

- 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 1 to the ACK bit in the interrupt status and control register (ISCR). 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 can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge on $\overline{\text{IRQ}}$ 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 1 — As long as the $\overline{\text{IRQ}}$ pin is at logic 0, the IRQ latch remains set.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic 1 can occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic 0. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

Keyboard Interrupt (KBD) Module

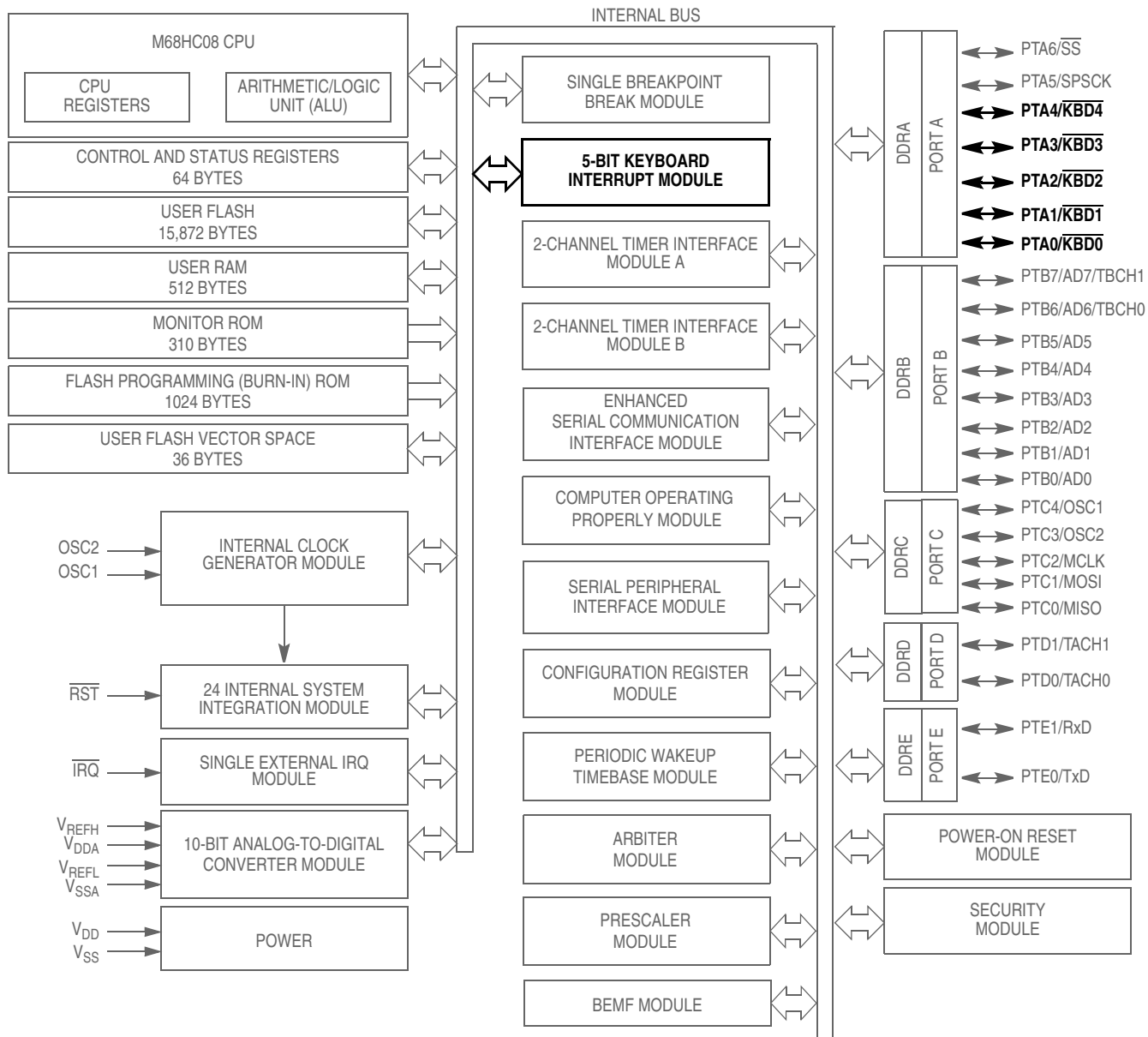


Figure 10-1. Block Diagram Highlighting Keyboard Block and Pins

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.

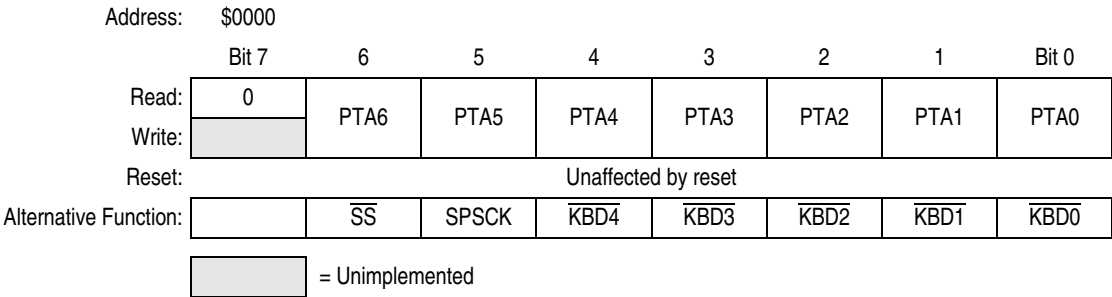


Figure 12-1. Port A Data Register (PTA)

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.

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

DDRC[4:0] — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC[4:0] and MCLKEN, configuring all port C pins as inputs.

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

NOTE

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

Figure 12-9 shows the port C I/O logic.

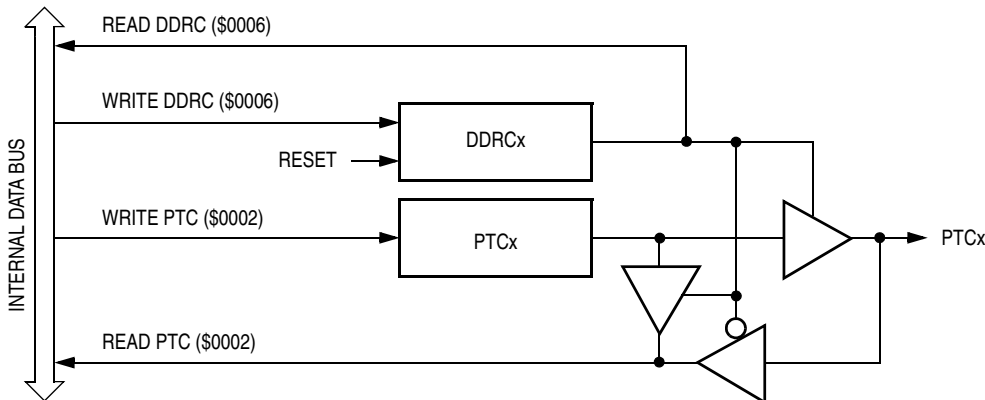


Figure 12-9. Port C I/O Circuit

When bit DDRCx is a 1, reading address \$0002 reads the PTCx data latch. When bit DDRCx is a 0, reading address \$0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-3 summarizes the operation of the port C pins.

Table 12-3. Port C Pin Functions

DDRC Bit	PTC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PTC	
			Read/Write	Read	Write
0	2	Input, Hi-Z	DDRC[7]	Pin	PTC2
1	2	Output	DDRC[7]	0	—
0	X	Input, Hi-Z	DDRC[4:0]	Pin	PTC[4:0] ⁽¹⁾
1	X	Output	DDRC[4:0]	PTC[4:0]	PTC[4:0]

X = don't care

Hi-Z = high impedance

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

12.5 Port D

Port D is a 2-bit special function port that shares its pins with the timer interface module (TIMA).

12.5.1 Port D Data Register

The port D data register contains a data latch for each of the two port D pins.

Fast Data Tolerance

Figure 13-8 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

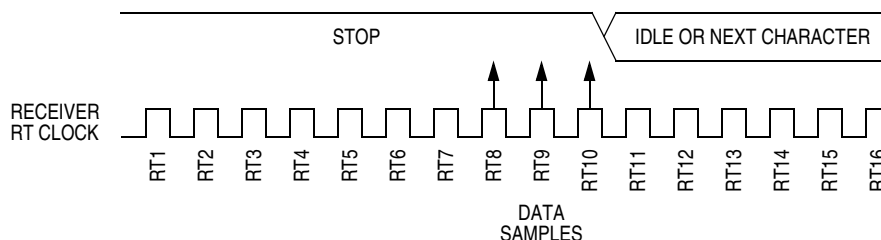


Figure 13-8. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver
 $9 \text{ bit times} \times 16 \text{ RT cycles} + 10 \text{ RT cycles} = 154 \text{ RT cycles}$.

With the misaligned character shown in Figure 13-8, the receiver counts 154 RT cycles at the point when the count of the transmitting device is $10 \text{ bit times} \times 16 \text{ RT cycles} = 160 \text{ RT cycles}$.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left| \frac{154 - 160}{154} \right| \times 100 = 3.90\%.$$

For a 9-bit character, data sampling of the stop bit takes the receiver
 $10 \text{ bit times} \times 16 \text{ RT cycles} + 10 \text{ RT cycles} = 170 \text{ RT cycles}$.

With the misaligned character shown in Figure 13-8, the receiver counts 170 RT cycles at the point when the count of the transmitting device is $11 \text{ bit times} \times 16 \text{ RT cycles} = 176 \text{ RT cycles}$.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$\left| \frac{170 - 176}{170} \right| \times 100 = 3.53\%.$$

13.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

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 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

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.

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the ESCI detects noise on the RxD pin. NF generates an NF CPU interrupt request if the NEIE bit in SCC3 is also set. Clear the NF bit by reading SCS1 and then reading the SCDR. Reset clears the NF bit.

- 1 = Noise detected
- 0 = No noise detected

FE — Receiver Framing Error Bit

This clearable, read-only bit is set when a 0 is accepted as the stop bit. FE generates an ESCI error CPU interrupt request if the FEIE bit in SCC3 also is set. Clear the FE bit by reading SCS1 with FE set and then reading the SCDR. Reset clears the FE bit.

- 1 = Framing error detected
- 0 = No framing error detected

PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the ESCI detects a parity error in incoming data. PE generates a PE CPU interrupt request if the PEIE bit in SCC3 is also set. Clear the PE bit by reading SCS1 with PE set and then reading the SCDR. Reset clears the PE bit.

- 1 = Parity error detected
- 0 = No parity error detected

13.8.5 ESCI Status Register 2

ESCI status register 2 (SCS2) contains flags to signal these conditions:

- Break character detected
- Incoming data

Address:	\$0014							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	BKF	RPF
Write:								
Reset:	0	0	0	0	0	0	0	0
	<div style="display: inline-block; width: 20px; height: 10px; background-color: #cccccc; border: 1px solid black;"></div> = Unimplemented							

Figure 13-14. ESCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the ESCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading the SCDR. Once cleared, BKF can become set again only after 1s again appear on the RxD pin followed by another break character. Reset clears the BKF bit.

- 1 = Break character detected
- 0 = No break character detected

RPF — Reception in Progress Flag Bit

This read-only bit is set when the receiver detects a 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch), or when the receiver detects an idle character. Polling RPF before disabling the ESCI module or entering stop mode can show whether a reception is in progress.

- 1 = Reception in progress
- 0 = No reception in progress

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 (SCICTL).

13.9.1 ESCI Arbiter Control Register

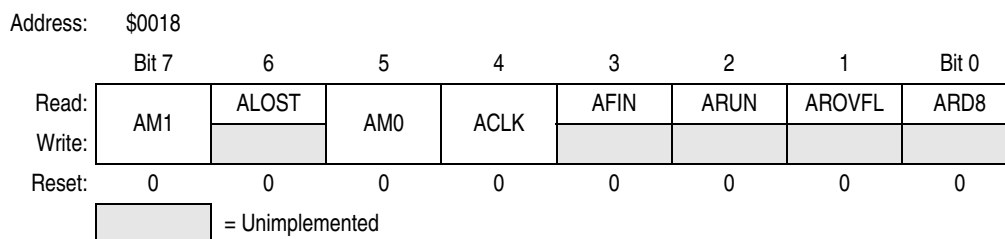


Figure 13-18. ESCI Arbiter Control Register (SCICTL)

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 SCICTL. Reset clears AFIN.

1 = Bit time measurement has finished

0 = Bit time measurement not yet finished

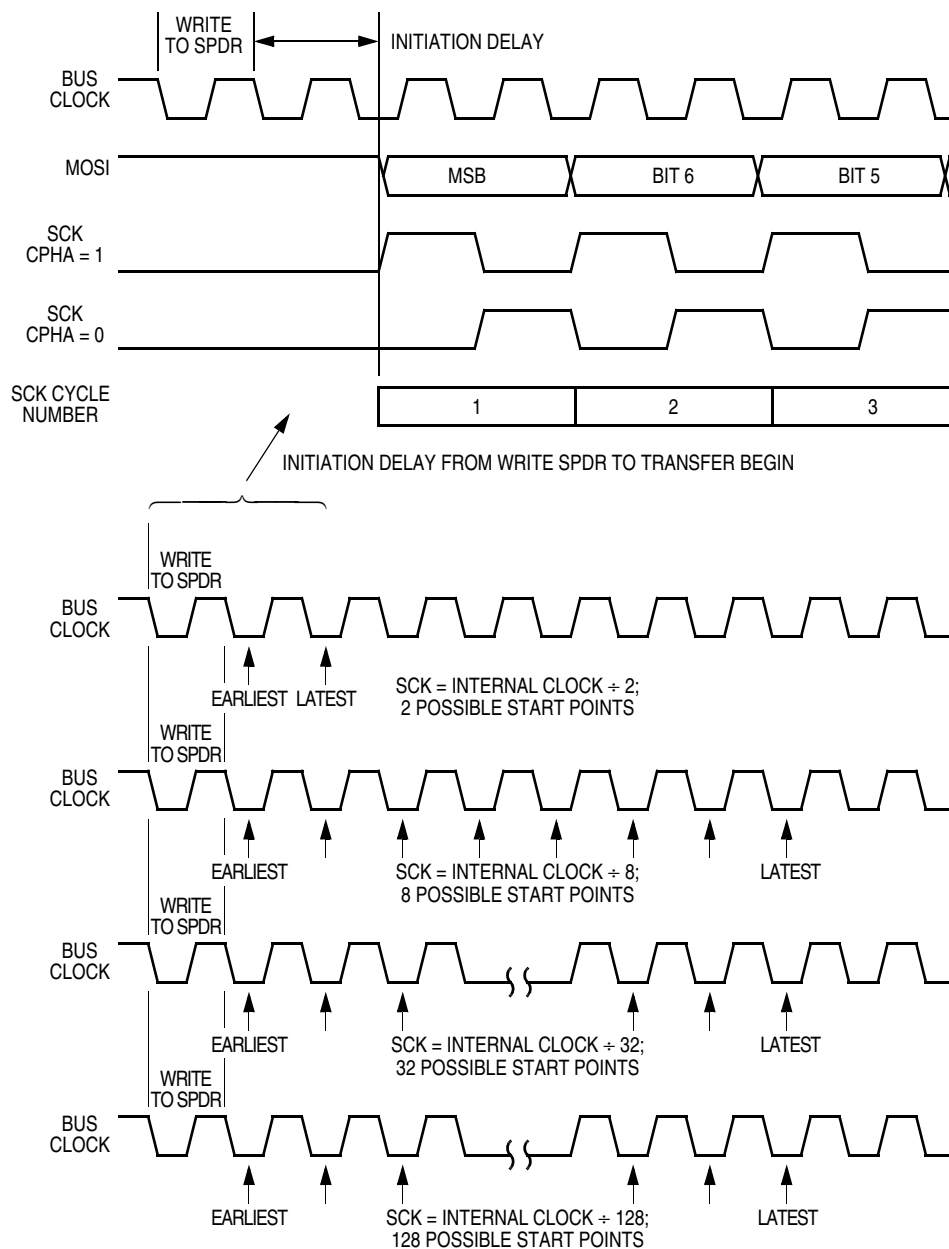


Figure 15-6. Transmission Start Delay (Master)

15.6 Error Conditions

Two flags signal SPI error conditions:

1. **Overflow (OVRF in SPSCR)** — Failing to read the SPI data register before the next byte enters the shift register sets the OVRF bit. The new byte does not transfer to the receive data register, and the unread byte still can be read by accessing the SPI data register. OVRF is in the SPI status and control register.
2. **Mode fault error (MODF in SPSCR)** — The MODF bit indicates that the voltage on the slave select pin (\overline{SS}) is inconsistent with the mode of the SPI. MODF is in the SPI status and control register.

17.4 Interrupts

These TIMA sources can generate interrupt requests:

- TIMA overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIMA counter modulo registers. The TIMA overflow interrupt enable bit, TOIE, enables TIMA overflow CPU interrupt requests. TOF and TOIE are in the TIMA status and control register.
- TIMA channel flags (CH1F–CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIMA CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE.

17.5 Low-Power Modes

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

17.5.1 Wait Mode

The TIMA remains active after the execution of a WAIT instruction. In wait mode, the TIMA registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIMA can bring the MCU out of wait mode.

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

17.5.2 Stop Mode

The TIMA is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions or the state of the TIMA counter. TIMA operation resumes when the MCU exits stop mode.

17.6 TIMA During Break Interrupts

A break interrupt stops the TIMA counter and inhibits input captures.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state.

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

17.8.4 TIMA Channel Status and Control Registers

Each of the TIMA channel status and control registers:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIMA overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Register Name and Address

TASC0 — \$0025

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Register Name and Address

TASC1 — \$0028

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0		R					
Reset:	0	0	0	0	0	0	0	0

R

R = Reserved

Figure 17-7. TIMA Channel Status and Control Register (TASC0–TASC1)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIMA counter registers matches the value in the TIMA channel x registers.

When CHxIE = 1, clear CHxF by reading TIMA channel x status and control register with CHxF set, and then writing a 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a 1 to CHxF has no effect.

- 1 = Input capture or output compare on channel x
- 0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIMA CPU interrupts on channel x.

Reset clears the CHxIE bit.

- 1 = Channel x CPU interrupt requests enabled
- 0 = Channel x CPU interrupt requests disabled

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the TIMA channel 0.

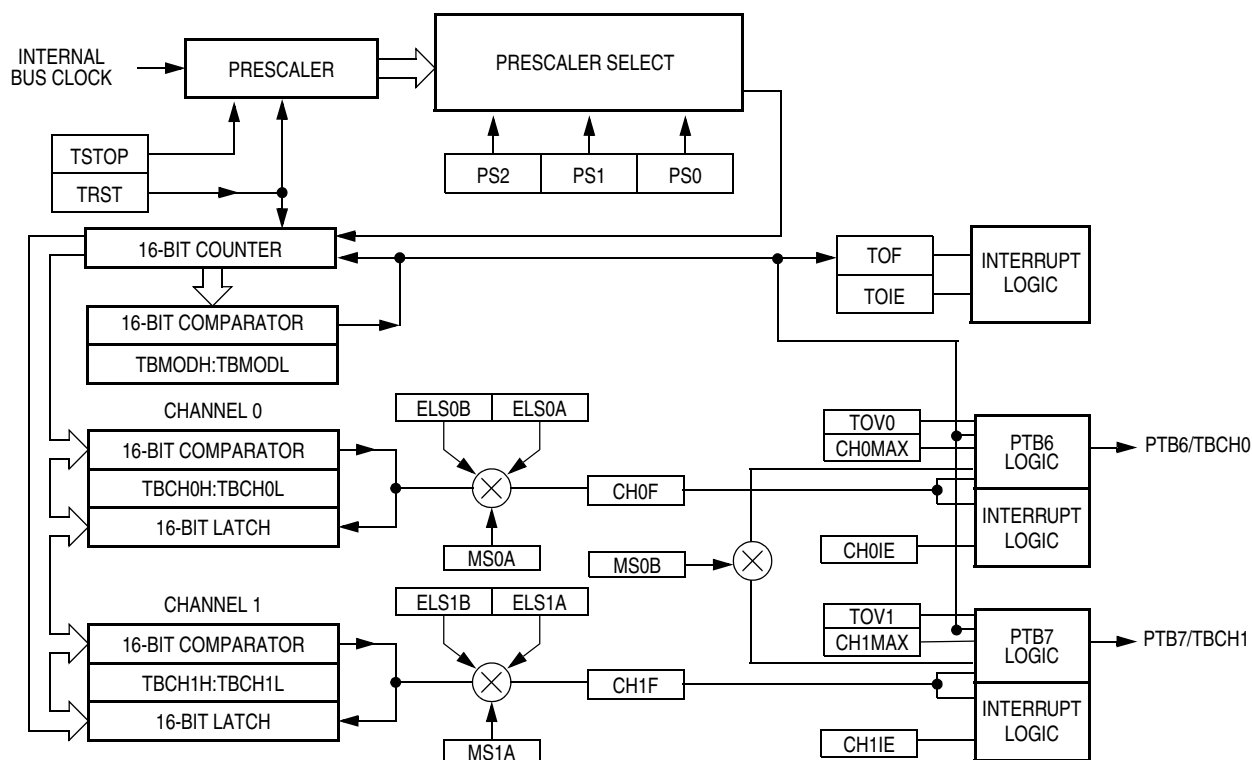


Figure 18-2. TIMB Block Diagram

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

Setting MS0B disables the channel 1 status and control register and reverts TBCH1 to general-purpose I/O.

Reset clears the MSxB bit.

- 1 = Buffered output compare/PWM operation enabled
- 0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See [Table 18-2](#).

- 1 = Unbuffered output compare/PWM operation
- 0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin once PWM, input capture, or output compare operation is enabled. See [Table 18-2](#). Reset clears the MSxA bit.

- 1 = Initial output level low
- 0 = Initial output level high

NOTE

Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIMB status and control register (TBSC).

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to port B, and pin PTBx/TBCHx is available as a general-purpose I/O pin. However, channel x is at a state determined by these bits and becomes transparent to the respective pin when PWM, input capture, or output compare mode is enabled. [Table 18-2](#) shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

Table 18-2. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

NOTE

Before enabling a TIMB channel register for input capture operation, make sure that the PTBx/TBCHx pin is stable for at least two bus clocks.

19.3 Monitor Module (MON)

The monitor module allows debugging and programming of the microcontroller unit (MCU) through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST} , as long as vector addresses \$FFFE and \$FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features of the monitor module include:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between MCU and host computer
- Standard non-return-to-zero (NRZ) communication with host computer
- Standard communication baud rate (9600 @ 2.4576-MHz bus frequency)
- Execution of code in random-access memory (RAM) or FLASH
- FLASH memory security feature⁽¹⁾
- FLASH memory programming interface
- Monitor mode entry without high voltage, V_{TST} , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Normal monitor mode entry if V_{TST} is applied to \overline{IRQ}

19.3.1 Functional Description

Figure 19-8 shows a simplified diagram of the monitor mode.

The monitor module receives and executes commands from a host computer.

Figure 19-9 and Figure 19-11 show example circuits used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

Table 19-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If \$FFFE and \$FFFF do not contain \$FF (programmed state):
 - The external clock is 9.8304 MHz (9600 baud)
 - $IRQ = V_{TST}$
- If \$FFFE and \$FFFF contain \$FF (erased state):
 - The external clock is 9.8304 MHz (9600 baud)
 - $\overline{IRQ} = V_{DD}$ (this can be implemented through the internal \overline{IRQ} pullup)
- If \$FFFE and \$FFFF contain \$FF (erased state):
 - The ICG clock is nominal 1.6 MHz (nominal 6300 baud)
 - $\overline{IRQ} = V_{SS}$

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

20.9 Trimmed Accuracy of the Internal Clock Generator

The unadjusted frequency of the low-frequency base clock (IBASE), when the comparators in the frequency comparator indicate zero error, can vary as much as $\pm 25\%$ due to process, temperature, and voltage. The trimming capability exists to compensate for process affects. The remaining variation in frequency is due to temperature, voltage, and change in target frequency (multiply register setting). These affects are designed to be minimal, however variation does occur. Better performance is seen with lower settings of N.

20.9.1 Trimmed Internal Clock Generator Characteristics

Characteristic ⁽¹⁾	Symbol	Min	Typ	Max	Unit
Absolute trimmed internal oscillator tolerance ^{(2),(3)} –40°C to 85°C –40°C to 135°C	$F_{\text{abs_tol}}$	— —	± 2.0 ± 2.5	± 3.5 ± 5.0	%
Variation over temperature ^{(3), (4)}	$V_{\text{ar_temp}}$	—	0.05	0.08	%/°C
Variation over voltage ^{(3), (5)} 25°C –40°C to 85°C –40°C to 135°C	$V_{\text{ar_volt}}$	— — —	1.0 1.0 1.0	2.0 2.0 2.0	%/V

1. These specifications concern long-term frequency variation. Each measurement is taken over a 1-ms period.
2. Absolute value of variation in ICG output frequency, trimmed at nominal V_{DD} and temperature, as temperature and V_{DD} are allowed to vary for a single given setting of N.
3. Specification is characterized but not tested.
4. Variation in ICG output frequency for a fixed N and voltage
5. Variation in ICG output frequency for a fixed N

20.11 SPI Characteristics

Diagram Number ⁽¹⁾	Characteristic ⁽²⁾	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	$f_{OP}/128$ DC	$f_{OP}/2$ f_{OP}	MHz MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2 1	128 —	t_{cyc} t_{cyc}
2	Enable lead time	$t_{Lead(S)}$	1	—	t_{cyc}
3	Enable lag time	$t_{Lag(S)}$	1	—	t_{cyc}
4	Clock (SPSCK) high time Master Slave	$t_{SCKH(M)}$ $t_{SCKH(S)}$	$t_{cyc} - 25$ $1/2 t_{cyc} - 25$	$64 t_{cyc}$ —	ns ns
5	Clock (SPSCK) low time Master Slave	$t_{SCKL(M)}$ $t_{SCKL(S)}$	$t_{cyc} - 25$ $1/2 t_{cyc} - 25$	$64 t_{cyc}$ —	ns ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	30 30	— —	ns ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	30 30	— —	ns ns
8	Access time, slave ⁽³⁾ CPHA = 0 CPHA = 1	$t_{A(CP0)}$ $t_{A(CP1)}$	0 0	40 40	ns ns
9	Disable time, slave ⁽⁴⁾	$t_{DIS(S)}$	—	40	ns
10	Data valid time, after enable edge Master Slave ⁽⁵⁾	$t_{V(M)}$ $t_{V(S)}$	— —	50 50	ns ns
11	Data hold time, outputs, after enable edge Master Slave	$t_{HO(M)}$ $t_{HO(S)}$	0 0	— —	ns ns

1. Numbers refer to dimensions in [Figure 20-1](#) and [Figure 20-2](#).

2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} , unless noted; 100 pF load on all SPI pins.

3. Time to data active from high-impedance state

4. Hold time to high-impedance state

5. With 100 pF on all SPI pins

Glossary

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."